

# **A Conceptual Model for Compliant Management Systems**

by

**Leigh Howard de la Motte,  
B.Comp. (Hons.), Grad. Cert. Commercialisation**

A dissertation submitted in fulfilment  
of the requirements of the Degree of

**Doctor of Philosophy**

**University of Tasmania**

**November 2012**

## Declaration

---

I, Leigh de la Motte, declare that this dissertation contains no material which has been accepted for the award of any other degree or diploma in any tertiary institution. To my knowledge and belief, this dissertation contains no material previously published or written by another person except where due reference is made in the text of the dissertation.

This dissertation may be made available for loan and limited copying and communication in accordance with the Copyright Act 1968.

Signed,

Leigh de la Motte

Date:

## Abstract

---

The dissertation specifies a Set of Concepts and a Conceptual Model for Compliant Management Systems (CMSs). The Set of Concepts is drawn from existing methodologies and the CMS Model, which is based on the Concepts, is evaluated against a compiled Set of Standard Requirements. The dissertation is analytical, conceptual and argumentative in nature.

The Model is designed to enable the future development of compliant, efficient and globally functional organisational Information Technology (IT) Systems. The *Three Central Requirements* of the Model – *Compliance* (essentially a Legal requirement), *Efficiency* (essentially a Business Management requirement) and *Global Functionality* (essentially an IT requirement) – necessitated a multi-disciplinary investigation in the Legal, Business Management and IT areas.

Compliance was the focus of the research as initial investigations showed no significant solutions existed that provided specific guidelines for the design of inherently compliant IT systems. As compliance relates to meeting regulatory requirements on a local or jurisdictional basis, whereas efficiency and global functionality favour global standards for IT systems (that enable system interoperability), the means for delivering “globally compliant” IT systems is problematic. The research therefore took the approach of “thinking globally and acting locally” by designing a Model appropriate for individual local organisations that can be used, in association with the development of appropriate regulation, as a template for globally compliant IT systems.

It was found that the issues of *Privacy*, *Consent*, and *Accountability* were at the core of compliance requirements and that Digital Service Contracts with systematic Authorisation mechanisms for contract parties (represented as IT System “Groups”) provided the means for assuring compliance. The Model proposes software based on *Service*, *Group* and *Service Contract* components – the Service Contract components fulfilling Legal requirements, the Service components fulfilling Business Management requirements and the Group components fulfilling IT requirements.

The Model description in the dissertation focuses on the IT System design. Technical (software design) issues are not explored in detail, making the dissertation readable to persons with minimal IT background. The aim was to draw together existing knowledge in the Legal, Business Management and IT disciplines and to explore relationships between the fields at a high level rather than to delve in detail into any of the three disciplines. The research spans many fields within the three research disciplines, including Contract Law, Privacy Law, Company Law, Legal Consent, Franchising Systems, Business Process Management, Service Oriented Architectures, Cloud Computing, Access Control, Authorisation, Operating Systems, Data Management and Object Oriented Programming.

## Acknowledgements

---

The initial part of the research (about 75%) was supervised by the School of Computing and Information Systems. The remainder was supervised by the School of Law.

Special thanks go to my Supervisors – Jacky Hartnett, for her patience, time and advice on security and IT concepts, Prof Gino Dal Pont for his guidance in producing this dissertation and his advice on legal matters and Prof Young J. Choi for his oversight.

I would also like to thank all the health practitioners and administrators that have contributed ideas. Special thanks go to my wife, Heather, for her insights into hospital practices.

Christian McGee and Andrew Spilling were very helpful in assisting me with IT issues. My thanks also go to Daniel de la Motte for proof reading this document.

Finally, thank you to all my PhD colleagues, who have always made themselves available to assist when asked. Particular thanks go to Barry Pearn and Dean Steer for their help.

# Table of Contents

---

## Chapter 1

<b>Introduction .....</b>	<b>1</b>
1.1 Research Goals and Central Requirements .....	1
1.1.1 Limits of the Research.....	2
1.1.2 Ultimate Control and Compliance.....	3
1.2 The Research Disciplines .....	4
1.2.1 Disciplinary Dissertation Approaches .....	5
1.2.2 Use of Terminology.....	6
1.3 Background .....	7
1.3.1 Regulatory (Legal) Background.....	8
1.3.2 IT Background.....	9
1.3.3 Business Management Background .....	11
1.3.4 Existing Research Methodologies .....	12
1.4 CMS Definitions.....	15
1.4.1 Major CMS Terms.....	16
1.4.2 Definitions of Major CMS Terms .....	16
1.4.3 Organisational CMS Terms.....	18
1.4.4 Introducing the Seven General Concepts .....	19
1.5 Services .....	23
1.5.1 All-Encompassing Services.....	23
1.5.2 Standardized Service Parties .....	24
1.5.3 Standardized Service Processes.....	25
1.5.4 Commercial Service Aspects.....	26
1.5.5 Compliant Management System .....	27
1.5.6 Enabling Service Automation and Reuse.....	28
1.6 Groups .....	30
1.6.1 Groups used to Define and Manage Resources .....	30
1.6.2 Hierarchical Groups .....	32
1.6.3 Globally Addressable Groups .....	33

1.6.4	Group Templates .....	34
1.6.5	Group Types and Sub-Types .....	34
1.6.6	Groups Associated with Services .....	35
1.7	Service Contracts .....	35
1.7.1	Organisation Based Contracts .....	36
1.7.2	Single-Service Contracts .....	36
1.7.3	Sub-Service Contracts .....	37
1.7.4	Generic Service Contracts .....	37
1.7.5	Digitally Managed Contracts.....	38
1.7.6	Digital Service Contracts Enabling Compliance.....	39
1.8	Compliance Concepts .....	40
1.8.1	Privacy .....	40
1.8.2	Consent .....	41
1.8.3	Accountability .....	42
1.8.4	Authorisation .....	43
1.9	Dissertation Structure .....	47

## Chapter 2

<b>Methodology .....</b>	<b>49</b>
2.1 Introduction .....	49
2.1.1 Initial Aim of the Research .....	49
2.1.2 Background .....	50
2.1.3 The Aim of Developing a Comprehensive and Practical Model..	50
2.1.4 A Research Voyage .....	51
2.2 Research Methodology .....	51
2.2.1 Examining Research Fields .....	53
2.2.2 Extracting Design Requirements .....	54
2.2.3 Finding Model Concepts .....	54
2.2.4 Developing the Model .....	55
2.2.5 The Validation Approach .....	56
2.3 Linking the General Concepts .....	58
2.3.1 Using Groups to Manage Authorisation and Consent.....	59
2.3.2 From Authorisation and Consent to Privacy .....	59

2.3.3	Linking Accountability and Privacy.....	60
2.3.4	Using Services to Manage Business Processes .....	60
2.3.5	Unifying the General Concepts with Service Contracts.....	61
2.4	Research Philosophy .....	61
2.4.1	Use Inductive Reasoning.....	61
2.4.2	Develop a Simple General Purpose Solution .....	62
2.4.3	Use Best Practice from Existing Solutions.....	63
2.4.4	Achieve Efficiency through Standardization .....	64
2.4.5	Incorporate Change Management .....	64
2.4.6	Adopt User-friendly Components .....	65
2.4.7	Automate Administration .....	66
2.5	Conclusion.....	67

## Chapter 3

### **Literature Review .....68**

3.1	Introduction .....	68
3.2	Computer Security.....	69
3.2.1	Information Security Issues.....	69
3.2.2	Data Security Qualities.....	70
3.2.3	Information Ownership and Control .....	70
3.2.4	Authentication and Authorisation .....	72
3.2.5	Access Control Basics .....	73
3.2.6	The Principle of Least Privilege.....	74
3.2.7	Granularity.....	75
3.2.8	Computer Security Conclusion .....	75
3.3	Role-Based Access Control.....	75
3.3.1	Group Concept .....	77
3.3.2	Context Concept.....	78
3.3.3	Multiple Contexts .....	79
3.3.4	Constraint Concept.....	80
3.3.5	RBAC Conclusion.....	81
3.4	Set Theory .....	82
3.4.1	Set Based Access.....	82



3.5	Clark-Wilson Model.....	83
3.5.1	Well Formed Transactions .....	83
3.5.2	The Clark-Wilson Model.....	84
3.6	Health Informatics .....	86
3.6.1	Cooperative Practices .....	86
3.6.2	Multiple Authorisations.....	87
3.6.3	Guaranteed Access .....	87
3.7	Professional Access Control.....	88
3.7.1	Professional Management .....	88
3.7.2	Service Teams .....	89
3.7.3	Restricted Access .....	89
3.7.4	Authorisation Timing .....	90
3.7.5	Authorisation Ordering.....	92
3.8	Purpose Based Access Control.....	92
3.8.1	Purpose Concept.....	92
3.9	Service Oriented Architecture .....	93
3.10	Cloud Computing .....	95
3.10.1	Inter-Domain Access .....	95
3.10.2	Inter-Domain Roles .....	96
3.10.3	Inter-Domain Rule Processing .....	97
3.11	Generic Programming .....	98
3.11.1	The Generic Nature of Services and Groups .....	98
3.11.2	User Perceptions of Service and Group Concepts.....	99
3.11.3	Generic Services and Groups .....	100
3.12	Business Process Management.....	101
3.12.1	Workflow Processes .....	101
3.12.2	Organisational Systems .....	102
3.12.3	Business Constraints and Workflow Management .....	103
3.12.4	Task Based Authorisation .....	104
3.12.5	Task Based Access .....	104
3.12.6	Standard Business Processes .....	105
3.13	Legal Compliance.....	106
3.13.1	Organisational Compliance .....	106

3.13.2 Privacy Requirements.....	107
3.13.3 Client Consent .....	108
3.13.4 Legal Accountability .....	109
3.13.5 Service Contracts.....	111
3.14 Design Requirements Table .....	113
3.15 Conclusion.....	115

## Chapter 4

<b>Model Concepts .....</b>	<b>116</b>
4.1 Introduction .....	116
4.2 Model Concepts Relationship Diagram .....	116
4.3 Services .....	118
4.3.1 C#1: Global Services.....	118
4.3.2 C#2: Service Based Organisation.....	119
4.3.3 C#3: Serviceflow Dependencies .....	119
4.3.4 C#4: Service Cells .....	120
4.3.5 C#5: Client-Services.....	121
4.3.6 C#6: Service Profiles.....	121
4.3.7 C#7: Service Based Information .....	122
4.3.8 C#8: Client-Service Management .....	122
4.3.9 C#9: Service Based Security .....	122
4.4 Groups .....	123
4.4.1 C#10: Service Teams.....	123
4.4.2 C#11: Contextual Groups.....	124
4.4.3 C#12: Group Based Organisation.....	125
4.4.4 C#13: Group Based Access .....	126
4.4.5 C#14: Global Groups.....	127
4.4.6 C#15: Global Group Addressing .....	128
4.4.7 C#16: Group Based Services.....	128
4.4.8 C#17: Group Based Information Storage.....	129
4.4.9 C#18: Set Based Groups.....	130
4.4.10 C#19: Partial Permissions .....	131
4.4.11 C#20: Group Based Management .....	132

4.5	Service Contracts .....	133
4.5.1	C#21: Compulsory Compliance .....	133
4.5.2	C#22: Compliant System.....	133
4.5.3	C#23: Worker-Service Contracts .....	134
4.5.4	C#24: Client-Service Contracts.....	135
4.5.5	C#25: Contract Based Services .....	135
4.5.6	C#26: Validation by Authorisation .....	136
4.6	Privacy.....	137
4.6.1	C#27: The “5 Ws” of Access Control .....	137
4.6.2	C#28: Service Based Purpose.....	137
4.6.3	C#29: Service Based Access Principle.....	138
4.6.4	C#30: Service Based Privacy .....	138
4.6.5	C#31: Access by Authorisation .....	139
4.7	Consent.....	140
4.7.1	C#32: Consent Type Specification.....	140
4.7.2	C#33: Service Based Consent .....	141
4.7.3	C#34: Bundled Client Consent.....	141
4.7.4	C#35: Implied Permissions .....	141
4.7.5	C#36: Consent by Authorisation .....	142
4.7.6	C#37: Service Based Information Provision .....	142
4.8	Accountability .....	143
4.8.1	C#38: Service Based Accountability.....	143
4.8.2	C#39: Client-Service Accounting .....	144
4.8.3	C#40: Required Auditing .....	144
4.8.4	C#41: Client-Service Records .....	144
4.8.5	C#42: Service Based Auditing .....	145
4.9	Authorisation .....	145
4.9.1	C#43: Mandatory Authorisation.....	145
4.9.2	C#44: Authorisation Type Specification.....	146
4.9.3	C#45: Managerial Authorisation .....	146
4.9.4	C#46: Client-Service Allocation .....	147
4.9.5	C#47: Worker Acceptance .....	148
4.9.6	C#48: Worker Authorisation .....	148

4.9.7	C#49: Client Authorisation .....	150
4.9.8	C#50: Service Based Authorisation .....	151
4.9.9	C#51: Sub-Service Authorisation.....	151
4.9.10	C#52: Authorisation Management .....	152
4.10	Model Concepts Table.....	152
4.11	Conclusion.....	154

## Chapter 5

### **Model Description..... 155**

5.1	Introduction .....	155
5.2	Incorporating Model Concepts .....	155
5.2.1	Foundational Model Concepts.....	156
5.2.2	Model Design Concepts .....	157
5.3	Model Components .....	158
5.3.1	Component Relationships.....	158
5.3.2	Organisation Composition.....	159
5.3.3	Component Origin .....	160
5.3.4	Component Instances .....	162
5.4	The Compliant Management System .....	163
5.4.1	CMS Programs and Data .....	165
5.4.2	Global Data Storage .....	169
5.4.3	A Global Access System example.....	170
5.4.4	Managing Components.....	172
5.5	Service Management .....	173
5.5.1	The Service Management Process.....	175
5.5.2	High Level Service Management .....	176
5.5.3	Management of Personal Work.....	177
5.6	Group Management .....	178
5.6.1	Group Management Services .....	179
5.7	Contract Management .....	180
5.7.1	Service Authorisations .....	181
5.7.2	Contract Management Services.....	182
5.8	Client-Service Example.....	183

5.9 Table of Model Concept Mechanisms.....	186
5.10 Conclusion.....	189

## Chapter 6

<b>Analytical Evaluation .....</b>	<b>190</b>
6.1 Introduction .....	190
6.1.1 The Standard Requirement Set.....	190
6.1.2 Model Evaluation .....	190
6.2 Purpose Related Standards .....	192
6.2.1 SR#1: Purpose & Correctness .....	192
6.2.2 SR#2: Need to Know Access .....	192
6.2.3 SR#3: Intellectual Property .....	193
6.3 Service/Task Related Standards .....	193
6.3.1 SR#4: Critical Protection.....	193
6.3.2 SR#5: Security Integration .....	193
6.3.3 SR#6: Service Based Access .....	194
6.4 Consent Practice Standards .....	194
6.4.1 SR#7: Data Collection.....	194
6.4.2 SR#8: Consent for Purpose .....	195
6.4.3 SR#9: Consent Judgment .....	195
6.4.4 SR#10: Consent Collection .....	196
6.5 Client Rights Standards .....	196
6.5.1 SR#11: Client Access .....	196
6.5.2 SR#12: Accessor Information .....	197
6.5.3 SR#13: Disclosure Account .....	197
6.5.4 SR#14: Restriction .....	197
6.5.5 SR#15: Complaint & Appeal .....	198
6.5.6 SR#16: Information Provision Process .....	198
6.6 Organisation Rights Standards .....	199
6.6.1 SR#17: Directory Data .....	199
6.6.2 SR#18: De-Identified Data .....	199
6.6.3 SR#19: Restriction Refusal .....	200
6.6.4 SR#20: Access Refusal .....	200

6.7	Access Rules Standards .....	200
6.7.1	SR#21: Sensitive Information .....	200
6.7.2	SR#22: Relevant Disclosures .....	201
6.7.3	SR#23: Basic Disclosures .....	201
6.7.4	SR#24: Emergency Access .....	202
6.7.5	SR#25: Event Based Disclosure.....	202
6.7.6	SR#26: Victim Disclosure.....	202
6.7.7	SR#27: Regulated Disclosure.....	203
6.8	Authorisation Standards .....	204
6.8.1	SR#28: Authorisation .....	204
6.8.2	SR#29: Role Based.....	204
6.8.3	SR#30: Validation .....	205
6.8.4	SR#31: Written Authorisation.....	205
6.8.5	SR#32: Information Flow.....	206
6.8.6	SR#33: Transaction .....	206
6.8.7	SR#34: Inter-Domain Authorisation .....	207
6.9	Compliance Practices Standards .....	207
6.9.1	SR#35: Openness & Accountability.....	207
6.9.2	SR#36: Compliance.....	208
6.9.3	SR#37: Risk Management.....	208
6.9.4	SR#38: Security Responsibility.....	209
6.9.5	SR#39: Skills & Qualifications .....	209
6.9.6	SR#40: User Account.....	210
6.9.7	SR#41: Privacy Personnel .....	210
6.9.8	SR#42: Data Correction .....	210
6.10	Information Provision Standards.....	211
6.10.1	SR#43: Information Provision.....	211
6.10.2	SR#44: Disclosure Account Data.....	211
6.10.3	SR#45: Staff Reminder .....	212
6.11	Monitoring/Auditing/Reporting Standards.....	212
6.11.1	SR#46: Activity Monitoring.....	212
6.11.2	SR#47: Security Auditing .....	213
6.11.3	SR#48: Management Auditing.....	213

6.11.4 SR#49: Incident Management .....	213
6.11.5 SR#50: Required Documentation.....	214
6.12 Model Rating Table .....	214
6.13 Conclusion.....	216

## Chapter 7

### **Results and Discussion .....218**

7.1 Introduction .....	218
7.1.1 Incorporating the Old and the New .....	219
7.1.2 Tradition and Future Possibilities.....	220
7.1.3 Towards Automated Business Management .....	221
7.1.4 Results and Discussion Topics .....	222
7.2 Contributions of the Research .....	222
7.2.1 Model Concept Set and New Concepts .....	222
7.2.2 The Model .....	230
7.2.3 The Standard Requirements .....	231
7.3 Analysis of the Model .....	231
7.3.1 Model Validation.....	231
7.3.2 Comparison with Current Methodologies .....	232
7.3.3 Limitations of the Model.....	239
7.4 Application of the Model .....	241
7.4.1 Regulatory Compliance .....	241
7.4.2 Business Efficiency .....	250
7.4.3 Global Functionality.....	255
7.5 Future Work .....	259
7.5.1 Compliant Management System .....	259
7.5.2 Service Based Regulation.....	260
7.5.3 Privacy of Personal Records.....	260
7.5.4 Business Web .....	260
7.5.5 Web of Identities .....	260
7.5.6 Supply Chain Management .....	261
7.5.7 Component Repositories .....	261
7.6 Conclusion.....	262

**References .....264**

**Glossary .....268**



## List of Figures

---

Figure 1: Aims of the Research .....	1
Figure 2: CMS Operations .....	18
Figure 3: Model Concepts .....	20
Figure 4: A Service Based Organisation .....	23
Figure 5: Basic Organisational Concept.....	24
Figure 6: A Service and its Sub-Components .....	26
Figure 7: A Personnel Group and its Sub-Components .....	32
Figure 8: Service Contract Types .....	36
Figure 9: Research Fields and Methodologies .....	51
Figure 10: Basic Research Methodology .....	52
Figure 11: Research Methodology in Practice .....	52
Figure 12: Initial General Concepts .....	58
Figure 13: Utilising Groups.....	59
Figure 14: Linking Privacy.....	59
Figure 15: Linking Accountability .....	60
Figure 16: Utilising Services .....	60
Figure 17: Unifying the General Concepts with Service Contracts .....	61
Figure 18: Access Control Matrix, ACLs and C-Lists.....	74
Figure 19: Role-Based Access Control .....	76
Figure 20: Critical Factors Analysis of the OASIS SOA Reference Architecture....	94
Figure 21: Components of a Workflow (Russell et al. 2005) .....	101
Figure 22: Model Concepts Relationship Diagram.....	117
Figure 23: Group Based Organisation Example.....	126
Figure 24: Group Based Information Storage Example.....	130
Figure 25: Set Operations.....	131
Figure 26: Service Based Privacy .....	139
Figure 27: Key Terms for Managerial Service Offers .....	148
Figure 28: Key Terms for Worker Acceptance of Services .....	149
Figure 29: Key Terms for Client Acceptance of Services.....	150
Figure 30: Service Based Authorisation.....	151
Figure 31: Foundational Model Concepts .....	156

Figure 32: Basic Model Component Relationships.....	158
Figure 33: Organisation Components.....	159
Figure 34: Component Hierarchies .....	161
Figure 35: CMS Operating System (CMSOS) .....	164
Figure 36: CMS Service Process.....	166
Figure 37: CMS Contents.....	167
Figure 38: Data and Component Relationships.....	168
Figure 39: Global Search and Authorisation Example.....	171
Figure 40: Service Management Process .....	174
Figure 41: Sub-Group and Management Group Relationships .....	179
Figure 42: Client-Service Example .....	184
Figure 43: Webs and Clouds .....	249
Figure 44: Access Control Model Comparisons .....	252

## List of Tables

---

Table 1: Design Requirements .....	114
Table 2: Model Concepts .....	153
Table 3: Model Mechanisms .....	188
Table 4: Model Ratings .....	215

## Chapter 1

# Introduction

---

### 1.1 Research Goals and Central Requirements

The primary aim of this dissertation is to describe **specific concepts** that enable the development of organisational IT systems that are compliant in legal terms, efficient in management terms and globally functional in IT terms. Its secondary aim is to define and examine a **conceptual model** built on these concepts. A future aim is to develop a software system based on the conceptual model.

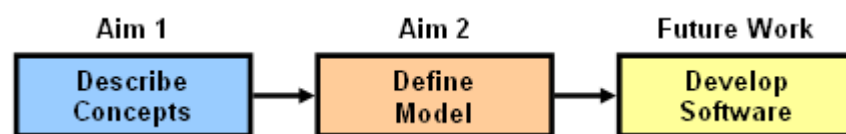


Figure 1: Aims of the Research

A total of 52 specific concepts, hereafter termed *Model Concepts* or *Concepts*, are described in the dissertation. The Model Concepts are sub-divided into seven groups, each based on one of the *Seven General Concepts* (the *General Concepts*) of *Services*, *Groups*, *Service Contracts* (*Digital Service Contracts* or *Contracts*), *Privacy*, *Consent*, *Authorisation*, and *Accountability*. A major part of the research involved defining these General Concepts and their interrelationships.<sup>1</sup>

The conceptual model is called the *Compliant Management System Model* (the *CMS Model* or the *Model*). The purpose of the Model is to describe the components and mechanisms that are necessary in a software system that meets *Compliance*, *Efficiency* and *Global Functionality* requirements. These three requirements (Compliance, Efficiency and Global Functionality) are hereafter termed the *Three*

---

<sup>1</sup> The first occurrence of CMS Terms defined in the dissertation appear in *italics* (see also 1.2.2).

*Central Requirements* (or the *Central Requirements*) of the research. A software system that is designed to meet the requirements of the Model is hereafter termed a *Compliant Management System (CMS)*, *IT System* or *System* (note the capitalised ‘S’).

**Compliance** (the *First Central Requirement*), the most basic requirement, refers to ensuring that an *Organisation* employing a CMS functions in a compliant fashion, that is, in accordance with relevant *Regulations*. **Efficiency** (the *Second Central Requirement*) refers to the CMS enabling the Organisation to run in an efficient fashion. Since the system to be developed is an IT system that must be interoperable with other systems, the CMS requires **Global Functionality** (the *Third Central Requirement*). While Efficiency and Global Functionality are considered equally important, the research focusses primarily on Compliance as an existing scalable solution in the area was lacking. The Central Requirements are defined more fully at 1.4.2.

The Model fundamentally proposes that *Digital Service Contracts (DSCs)* be used as the instrument for achieving Organisational Compliance, and that systematic Authorisation mechanisms for Contract parties (*Managers, Workers* and *Clients*) provide the means for assuring that the Contractual obligations of Organisations are met.

### 1.1.1 Limits of the Research

It was initially intended that a live CMS system would be tested in a hospital environment, but this proved problematic due to the scale of the test required and the available resources. This, together with the need for efficiency and global functionality, prompted the search for a broader, more generic solution. The research therefore does not seek to prove itself directly applicable to any specific set of organisational situations.

The work focuses on authorisation, that is, what authenticated persons are allowed to do within a system. While the issue of authentication is discussed, the work does not specify any particular authentication system.

Neither does the research specify any specific form for Digital Service Contracts. It only provides the IT System mechanisms necessary for their development and management. It avoids reference to specific laws, rules and regulations, as compliance within specific jurisdictions or organisational environments does not lie at the core of the significance of the Model.

The dissertation is conceptual in that it seeks to highlight, associate and draw together Concepts from the three research disciplines in preference to detailing solutions to specific problems in any one of the disciplines. Consequently, some existing methodologies may have been rejected or not considered due to their incompatibility with other aspects of the Model rather than their individual merit.

The dissertation is analytical and argumentative in its approach to dealing with the relevance of existing methodologies and comparisons with the functionalities of the proposed CMS Model. The analysis, particularly with regard to evaluation of the Model, is qualitative rather than quantitative, as scientific testing of a particular solution was not the aim. Mathematical analysis of aspects of the Model was not considered feasible due to the available expertise or of sufficient value compared to the analytical evaluation method used.

Finally, it was not the aim to specify particular Business Process Management or Service Oriented Architecture methodologies within the Model, as this is considered a part of future work in the design of CMS software.

### **1.1.2 Ultimate Control and Compliance**

One of the reasons for using IT is to automate manual tasks with a view to making life easier and/or safer. For example, IT can be used to control the temperature in a room. The nature of such automation is that decisions previously controlled by humans are controlled, or at least potentially controlled, by IT Systems.

While some may view such automated decision making as best suited to non-essential actions, such as switching on a heater at the moment the temperature falls below a prescribed minima, any IT-controlled system can feasibly be programmed to make any decision automatically.

Consider the issue of controlling an airliner. Flight crew have the option of engaging or disengaging the auto-pilot. However, the auto-pilot could potentially decide whether or not it should be in control and only allow the crew to pilot the plane when it decides it is warranted. This may indeed be preferable if and when it becomes evident that auto-pilots are safer than human pilots, especially considering that a human pilot can be incapacitated or be a suicidal terrorist.

An ultimate example of automation is a domestic robot that is capable of killing a human being. The question this dissertation asks is whether or not such automation should be designed in compliance with appropriate regulation. The dissertation does not argue whether or not ultimate control should be given to IT Systems; it argues that automated systems should be proven to be compliant *before* ultimate control is surrendered to them.

This research focuses on IT-controlled business systems where **the IT system runs the business in a compliant manner**. This is an extension of Business Management expert Michael Gerber's (Gerber 1995, pp. 91-92) observation that in franchised businesses "[t]he system runs the business. The people run the system...". Such an IT system is not merely a tool used by persons, it exerts a high level of automated control where manual overriding is the exception rather than the rule.

## 1.2 The Research Disciplines

Each of the Three Central Requirements has a particular dimension. The Model must enable a CMS that is compliant in a legal sense, efficient (and thus commercially attractive) in a business management sense and globally functional in an IT sense. To build globally functional software that is efficient and compliant it is necessary to not only understand Information Technology (IT) design requirements but Legal and Business Management requirements as well. The research cannot therefore avoid some multi-disciplinary slant.

The dissertation therefore relates to the *Three Research Disciplines (Research Disciplines)*: IT, Business Management and Law. Within each of these disciplines a

number of sub-disciplines were studied. Details of these studies are found in Chapters 2 and 3.

The first three General Concepts each have their basis in one of the Three Research Disciplines. Services relate to Business Management, Groups relate to Information Technology Management and Service Contracts relate to Law. The Model therefore is strongly influenced by each of the Research Disciplines.

### **1.2.1      Disciplinary Dissertation Approaches**

In each of the three Research Disciplines there are “standard” approaches to writing and structuring dissertations. There are, unsurprisingly, some inconsistencies between these standard approaches.

IT dissertations tend to take either an “Information Systems Approach” that uses case studies or a “Computer Science Approach” where software is written and tested. Each tends to proffer a hypothesis that is subject to some quantitative test and/or analysis.

A scientific approach places no credence on commercial viability or success. This conflicts with “Business Management Approaches” that *are* concerned with methodologies that have been shown to be efficient or successful.

Similarly, a “Legal Approach” tends to view organisational compliance as the responsibility of organisational executives (managers). It is not generally, or at least overly, concerned about system designs or standard business practices.

As this dissertation crosses the three Research Disciplines, the approach adopted needs to be somewhat unique. The chosen approach is *analytical* in that it studies existing methodologies and issues in each of the Research Disciplines, *conceptual* in that it focuses on developing ideas that are applicable in each discipline, and *argumentative* in that justification for the proposed Concepts and the Model are based on argument as opposed to quantitative testing.

### 1.2.2 Use of Terminology

As the research is multi-disciplinary there is a need to clearly define the terminology used in this dissertation. *CMS Terminology* is the term used to describe the complete collection of terms defined in the dissertation. The major *CMS Terms* are defined in 1.4.2 and in the Model Concepts Chapter (Chapter 4), while others are defined as they appear in the text. The complete collection is contained in the Glossary.<sup>2</sup>

#### 1.2.2.1 The Use of Italics and Capitalisation for CMS Terminology

*Italics* are used for two purposes in this dissertation – for CMS Terminology and for general emphasis. The initial occurrence(s) of each CMS Term in the dissertation is highlighted in first letter capitalised *italics*. Some non-CMS Terms also appear in *italics* to highlight the particular words (as in this paragraph). These terms can be distinguished by their being in lower case and by their absence in the Glossary.

After their first appearance, most CMS Terms appear first letter capitalised (for example, CMS Terms). This distinguishes them from the general usage of words. For example, the words “service” and “system” are used to describe any service or system whereas the words Service and System refer to a CMS Service and a CMS System respectively.

#### 1.2.2.2 References to System Functionalities and Mechanisms

Any number of different CMSs may be developed from the CMS Model. However, when the functionalities or mechanisms of a CMS are discussed, the dissertation targets the singular terms; *the CMS*, *the IT System* or *the System* to distinguish what the System does from what other non-CMS systems do or what the users do. For example, *the System* uses Digital Service Contracts or *the Manager* sends an Authorisation Request to *the System*.

#### 1.2.2.3 References to Components

The term “component” is used generally in the dissertation to mean “a part of”. The capitalised CMS Term *Component*, however, refers to software Components in the

---

<sup>2</sup> The names of the Model Concepts are considered CMS Terms.



Model. Furthermore, Model Components can be described at different levels, as a “bowl of chicken soup” can be described at different levels in a restaurant. The soup is an “entrée” (a type), a “recipe” (a template) and a particular “dish” (an instance) served to a particular customer.

At the highest level, there are three types of Components in the Model – Services, Groups and Service Contracts – hence the dissertation speaks of Service Components, Group Components and Contract Components. At the second level each particular Service, Group or Contract Component is a template for Component instances (the third Component level) – hence the dissertation speaks of Services in terms of the items offered by an organisation and *Client-Services* as individual instances of particular Services.

As the software Components in the Model mirror real world Services, Groups and Contracts, there are three levels of software Components. There are Service, Group and Contract software *Component Types*; Service, Group and Contract software *Component Templates*; and Service, Group and Contract software *Component Instances*.

Components are all hierarchical in that Components can be *Sub-Components* of higher level Components. There are three types of Sub-Components – *Sub-Services*, *Sub-Groups* and *Sub-Contracts*.

#### **1.2.2.4 References to Contracts**

As mentioned previously, CMS Systems use Contracts called Digitally Service Contracts (DSCs). The CMS Terms, *Digital Service Contract*, *Service Contract*, *Contract*, *Generic Service Contract* and *Digitally Managed Contract* all refer to a “CMS Service Contract” and are used interchangeably throughout the dissertation according to the context.

### **1.3 Background**

The Model facilitates CMSs that are globally connected IT systems that efficiently automate business management and seamlessly incorporate compliance into

management processes. An organisational CMS is an IT system. It could also be described as a “service-based system”, a “business system”, a “business management system”, or a “generic business system”, but for consistency it will normally be termed the “CMS” or the “System”.<sup>3</sup>

The following subsections give some basic background information in each of the Three Research Disciplines. The main purpose is to provide background to readers who are unfamiliar with aspects of these Disciplines.

### **1.3.1 Regulatory (Legal) Background**

The Central Requirement of Compliance directed the research into the study of relevant regulation. This background information relates to ways to attain regulatory compliance.

Regulations are developed in order to set levels of practice and to provide some form of standardisation. The issue of compliance relates to how well an entity conforms to regulation. Two fundamental approaches are used to encourage compliance – “penalisation” and “compulsion”. Penalisation has to do with checking for non-compliance and applying penalties to those not complying. Compulsion has to do with requiring that only technology or practices that inherently comply can be used. For example, there are two ways of govern compliance with speed limits. A penalisation based approach involves issuing fines to drivers who do not comply with speed limits. A compulsion based approach could involve only registering vehicles that are fitted with speed limiters.

Penalisation and compulsion based approaches can also be applied to business practices. Fines can be imposed for businesses that breach regulations. Alternatively, a compulsion based approach could mandate the use of business systems which inherently comply with regulation. This dissertation argues that for business compliance a regulatory approach based on compulsion is preferable to one based on penalisation. The compulsion-based approach requires a business system

---

<sup>3</sup> *The CMS or the System* refers to an Organisation’s CMS which is one instance of *a* CMS.

that is inherently compliant. The Model proposes such a system, one based on *Generic Service Contracts* (see 1.7.4).

Regulation may be considered effective if rates of non-compliance are low. Non-compliance can often stem from the “Cost of Compliance”, namely the need for responses from business, which carry both a time and financial cost. There may also be a “Cost of Non-Compliance”, which can be estimated according to the “Probability of Penalisation” multiplied by the “Cost of Penalisation”. Where the Cost of Non-Compliance is similar to or less than the Cost of Compliance, there is little incentive for businesses to comply in practice. Levels of compliance will, in these instances, tend to be lower. This is analogous with drivers choosing to speed in places where they perceive there is little risk of being penalised.

To achieve high Rates of Compliance, it is necessary to minimise the Cost of Compliance. One of the fundamental ways of reducing compliance costs is to provide systems that are inherently compliant. Such systems are suitable for a diversity of organisations thereby reducing costs due to economies of scale. The Model proposes a System that is a generic business system.

### 1.3.2 IT Background

This background information relates to the ability of computers (and IT systems)<sup>4</sup> to provide generic business systems that *automate* tasks and *reuse* stored information.

Computers fundamentally operate by processing data. They contain two fundamental electronic components: a processor and a memory. The processor performs a series of logical operations which are stored in memory as computer programs. In other words, the program provides the processor with a series of logical operations for it to do. Each operation generally involves the input of data from the memory and the output of resultant data to the memory. If a computer program can be likened to a cake recipe, then a single computer *process* is one execution of the program as the making of a single cake is one execution of the cake recipe. Each execution of a program thus spawns a new process in the processor.

---

<sup>4</sup> Computers are actually components of IT systems although the two terms are often interchangeable.

The reason computers are useful to humans is that they can perform complex processes quickly and accurately, and can store and transmit large amounts of information. The challenge for IT system designers has always been how to utilise these “machine world” characteristics to gain “real world” advantages. To interact with the real world, computers are connected to input and output devices. Input devices either receive direct user input or automatically monitor real world conditions. Output devices can provide feedback to users or be automatically controlled to perform real world actions.

IT systems (utilising computers) are useful to business because they can provide efficiency gains. These gains are fundamentally due to the *automation* of repetitive business processes and the *reuse* of stored information. For example, an IT system may utilise a program that issues a reminder notice to clients when their accounts become overdue. Given the necessary information, the IT system can automatically check the accounts and issue the reminders. The reminders can simply reuse a standard letter with customised client information. IT systems can thus make quantitative decisions based on stored data values. They can also generate new data by processing existing stored information. The Model proposes using a service-based IT system that utilises *automation* and *reuse* benefits.

Computers (and IT systems) not only reuse information, they reuse programs. Reuse occurs where the same programs are reused multiple times in the same computer. Programs can also be reused by copying them to other computers. Essentially, once a program is designed for one computer, it can be used by any computer. As the cost to copy a program from one computer to another is negligible, the more computers that use a program, the cheaper the cost per use. The major cost, then, is in initially writing the program. “Generic programs” that can be reused in many different places for many different purposes provide major cost benefits over programs tailored for the particular needs of individual organisations. It was once the norm for programs to be tailored for individual organisations. Now generic programs are simply configured or customised for each organisation. For example, in the past various word processing programs were used. Now Microsoft Word is used by most organisations for myriad purposes. Furthermore, if organisations use the same programs, they easily can share information with each other. For instance,

organisations that each use Microsoft Word can share Microsoft Word documents. The Model proposes an IT-based generic business system composed of generic programs.

Advances in technology have enabled computers to be linked together to form computer networks. Networks vary in size from local area networks in homes, through to organisational networks, up to the internet, which is a global network. Many nonetheless view computers in terms of their own experience with Personal Computers (PCs). With PCs all the essential components are in front of the user; everything is located *locally*. There is a box containing the processor and the memory with a few input devices (for example, keyboard, mouse and dvd player) and output devices (for example, monitor and printer) attached. The software is all stored in the memory and the processor performs all the computing. This view is very limited, as networking technology enables the use both hardware and software components which can be located *remotely* (as well as *locally*). The Model advocates the use of both *local* and *remote* components.<sup>5</sup>

### 1.3.3 Business Management Background

This background information relates to the ability of an IT-based business management system to provide efficiency gains.

Business Management expert Michael Gerber (Gerber 1995, pp. 91-92) makes the point that in franchised businesses “[t]he system runs the business. The people run the system... The system integrates all the elements required to make a business work”. This dissertation argues that “the system” which runs the business<sup>6</sup> can be an IT system and that this ultimately means that “the IT system runs the business”. It is analogous to an IT system piloting a plane or controlling a spacecraft. Persons still

---

<sup>5</sup> The Model assumes that any component can be remote which means it deals with all possibilities.

<sup>6</sup> While the dissertation uses the terms “Business” and “Business Management”, the research also applies to non-profit and government entities. The terms “Organisation” and “Business” are largely interchangeable in the dissertation. “Organisation” is generally used when referring to the Model, while “Business” tends to be used when the context is more commercial.

run the IT system and can override it where necessary. The IT system simply performs the mundane management tasks in a prescribed manner.

An IT-based business management system offers two main efficiency benefits. First, internal business management tasks can be automated and require fewer personnel for management tasks. Secondly, external e-business methods can be utilised more directly to increase online enquiries and purchases. This dissertation aims to provide insights into how to leverage internal and external benefits by utilising the proposed IT-based business management system.

Compliance costs are also a business management issue. The proposed IT-based business management system offers the opportunity to build compliance into the System. The System uses normal business process actions to automatically trigger compliance-related actions. For example, the allocation of a patient to a nurse in a hospital System triggers the provision of access permissions for the nurse to access the patient's private medical record. The automation of such compliance-related actions provides efficiency gains that effectively reduce compliance costs.

### **1.3.4 Existing Research Methodologies**

The following provides a brief overview of the major existing research methodologies that have influenced the development of the Model, namely Business Process management, Service Oriented Architecture, Role Based Access Control, Generic Programming and Cloud Computing.

#### **1.3.4.1 Business Process Management and Service Oriented Architecture**

Business Process Management (BPM) and the Service Oriented Architecture (SOA) are two fields of research that focus on using IT systems to manage businesses. BPM stems from Business Management research where the focus is on examining existing business processes and then representing them in the IT system as "tasks". SOA stems from IT research where the focus is to get functional software components, "services" to work together to perform business processes. While the methodologies of the two fields differ, the basic aims are similar.

Both fields are having an impact in the commercial world. In particular, the largest business software company in the world, SAP, bases its technology on SOA. However, SAP solutions are mainly aimed at large enterprises and do not at present scale down well for Small to Medium Enterprises (SMEs).

The Model utilises a service-based approach that is similar to that used in SOA. BPM functionalities are incorporated in the proposed System.

#### **1.3.4.2 Role Based Access Control**

Another relevant field of research, within the broader field of Computer Security, is that of *Access Control*. Access Control is fundamentally concerned with “who has access to what” in an IT system – the “who” being a user (or “subject”) and the “what” a file (or “object”). Role Based Access Control (RBAC) is the arguably the most relevant Access Control Model employed in current IT systems. The fundamental RBAC characteristic is to grant access to users on the basis of group membership where groups represent organisational “roles”. The major drawback to RBAC is inefficiency, as systems using it properly require considerable work to administer where, as is the norm, there are large numbers of users and roles.

The Model utilises a “Group” concept (see 1.6) to manage resources. By incorporating Groups into the service-based approach the objective is to overcome the inefficiencies of RBAC.

#### **1.3.4.3 Generic Programming**

Generic programs are written to be useful to many different users. Their major drawback, vis-à-vis programs tailored to individual needs, is that users may have to alter the way they do things in order to use these programs. However, this disadvantage is usually outweighed by various advantages. The most obvious advantage is that the cost of generic programs is usually a fraction of the cost of tailored programs. A second advantage is that of being able to exchange files with other users of the generic program. A third advantage is that users who have learned to use generic programs do not ordinarily require training to the same extent as users of tailored programs.

There are other, more hidden, advantages that generic programs tend to possess. These have to do with the ability to build security features into the programs and the ability to ensure regulatory compliance. It is generally only when a program becomes broadly used that it is economically feasible to build these complex features into programs.

The Model proposes that generic programs be utilised. These programs form the proposed IT-based business management System.

#### **1.3.4.4 Cloud Computing**

The “Cloud Computing” paradigm is based on the idea that the components in the “cloud” are unknown or not visible to the user. Accepting that the term “cloud computing” is somewhat vague, the following definition, from the US National Institute of Standards and Technology (NIST), is useful:

*Cloud Computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models.*

The three service models mentioned in the definition are “Cloud Software as a Service” (SaaS), “Cloud Platform as a Service” (PaaS) and “Cloud Infrastructure as a Service” (IaaS). The idea of services being located within the cloud is evident. Indeed, there are manifold commonalities between the Cloud Computing and Service Oriented Architecture paradigms.

The Model incorporates the Cloud Computing paradigm, in that it assumes that System Components can be located *remotely* as well as *locally*. In fact, the requirement of *remote access* is a key requirement upon which the Model is designed. Nor are the Service and Group Components in the Model location specific. Services (that is, Service Templates or Service Components) are designed



to be imported from remote locations. Groups are designed to represent global entities.

Considerable technical work has been done to enable Cloud Computing systems to be functional. Indeed, the proportion of computing devices<sup>7</sup> that are connected to networks is increasing; most are now designed to be “connected”.

While this increased connectivity has obvious benefits, it has generated many business management, security and legal issues. For example, when a customer uses a service that is offered by one organisation and components of the service are provided by any number of unknown (at least to the customer) remote organisations, legal concerns arise. The organisation offering the service may itself be remote from the customer. In a Cloud Computing environment it will often be the case that the only sense in which a service belongs to an organisation is in the legal sense. The Model seeks to address these legal concerns by incorporating a generic legal contract, which it terms the Service Contract (see 1.7), with every Service. And the Model utilises generic Services (see 1.5.3) and global Groups (see 1.6.3) to deal with the business management and security issues.

## 1.4 CMS Definitions

Different terms are used in the Research Disciplines. As the dissertation is written for readers in each discipline, it is important to be clear regarding terminology. Jargon from a particular discipline may confuse those outside the discipline. CMS Terminology is the set of CMS Terms used in the dissertation. It seeks to be understandable to persons in all three disciplines.

The main purpose of CMS Terminology is to describe the Model. While most of the terms were developed for the sake of the Model, they seek to represent the nature of the relationships in a practical way. In other words, they are designed to be understandable for persons in the workplace. This aligns with the philosophy that

---

<sup>7</sup> In this context, fixed (desktop computers) as well as mobile computers (including smart-phones).

the Model will be easier to administer if users can understand it, at least to the degree needed to perform their jobs efficiently.

#### 1.4.1 Major CMS Terms

Major CMS Terms include Organisation, Regulation (fundamental CMS Terms) Compliance, Efficiency, Global Functionality (the three Central Requirements), Service, Group, Service Contract (the three Component Concepts), Privacy, Consent, Accountability, Authorisation (the four Compliance Concepts), Manager, Worker, Client (the three Service Contract parties) and the names of the Model Concepts.

#### 1.4.2 Definitions of Major CMS Terms

*An **Organisation** is an organisation that employs a CMS.*

***Regulation** refers to any rule that is externally imposed on an Organisation with which it is required to comply.*

***Compliance** means the fulfilment by Organisations of all the requirements of applicable Regulations.*

***Efficiency** refers to Organisations functioning with lower administrative overheads and/or greater commercial viability than current methods or systems generally allow.*

***Global Functionality** refers to CMSs being functional in processing terms (for example, processing speed) and interoperable with other relevant systems on the Internet.*

***Services** are templates for standard business process tasks in Organisations, with each Client-Service instance being managed by the IT System.*

*A **Group** is a resource set with related membership data and Management Groups.*

*A **Service Contract** is a digital Service agreement that enables Organisational Compliance by fulfilling Privacy, Consent and Accountability Regulations and CMS Authorisation requirements.*

*Client **Privacy** is enabled by restricting access to information to that which is necessary for the Client-Service being provided.*

*Client **Consent** is incorporated into Service Contracts and managed in Service processes.*

***Accountability** relates to meeting accounting-related Regulations through the use of appropriate Service Contracts and Service Delivery processes.*

***Authorisation** mechanisms are the means by which Service Contract offers and acceptances are entered into the IT System during Service allocation and delivery processes.*

*A **Manager** is the Organisational representative having the authority to make Service-based decisions on behalf of the Organisation.*

*A **Worker** is an Organisational representative or sub-contractor who delivers Organisational Services to Clients on behalf of the Organisation.*

*A **Client** is the recipient of Organisational Services and is external to the Organisation unless the Service is an Internal Service.*

The Model Concepts are defined in Chapter 4.

### 1.4.3 Organisational CMS Terms

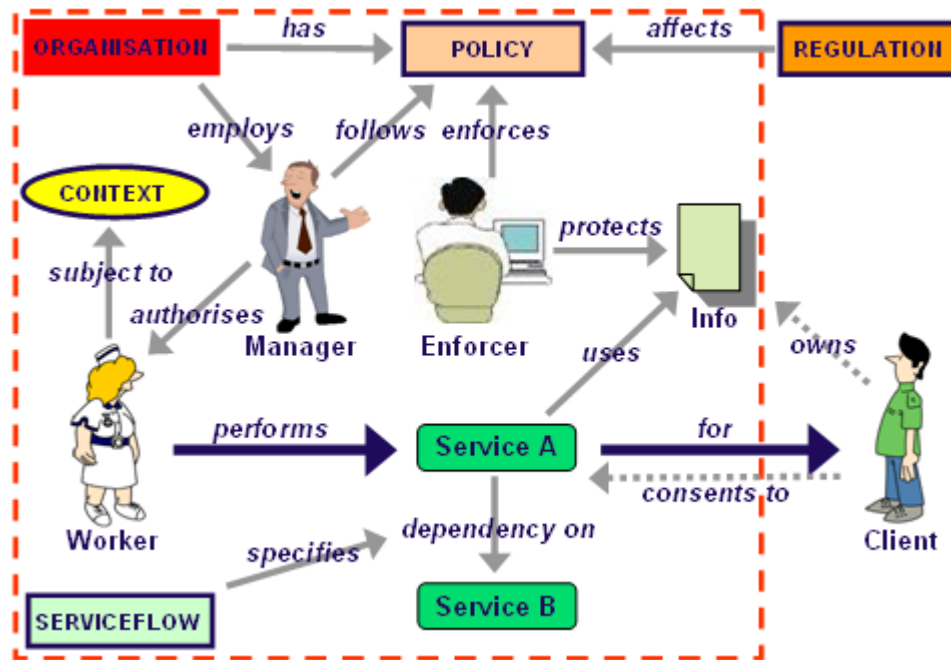


Figure 2: CMS Operations

Figure 2 displays the “Organisational CMS Terms” pictorially. These terms are used throughout the dissertation to describe the operational aspects of the CMS System. The diagram is important because it summarises the operational relationships between the major Organisational entities and resources. It is “Service oriented” since Services are “exposed” and represent the operational work of the Organisation – Groups and Service Contracts come into play as part of *Service Delivery*. An explanation of the diagram follows, with CMS Terms first letter capitalised and “relational” terms in bold.

The *Organisation* is represented by the dashed red rectangle. The Organisation **has** a *Policy* it maintains and follows. *Regulations* specified by external bodies **affect** the *Policy*. The Organisation has *Serviceflows* that **specify** business processes that are consistent with the *Policy*.

Four fundamental players are represented in the diagram: the *Enforcer*, the *Manager*, the *Worker* and the *Client*. The *Enforcer* is the *Manager* in charge of the System, who is responsible for establishing, configuring, monitoring and reporting on the System’s performance. The *Enforcer*’s job is to ensure that policy is **enforced** and that *Info* (Information or data) is **protected**. The *Manager* is **employed** by the

Organisation and is charged with making decisions on behalf of the Organisation. He or she must **follow** the Organisation's Policy. The decisions the Manager makes relate to Authorising the work that the Organisation does. More specifically, the Manager **Authorises** a Worker to **perform** a Service for a Client. When a Service is performed for a Client, this is one instance of Service Delivery known as a *Client-Service*. The details of Service Delivery are contained in the *Service Description* that forms part of the Service Contract.

The Authorisation is **subject** to *Context*. Context specifies conditions such as role, time or location. Groups are used in the Model to specify Context. Performance of the Service also requires the **Consent** of the Client. In order to perform the Service, information relating to the client is required and/or generated. In this sense, Client-Services **use** Info (*Service Information*). Clients have a form of **ownership** of their Info, that is, their Personally Identifiable Data (PID). The Organisation is required to **protect** the Privacy of this Info and to maintain accurate Service records for Accountability purposes. The performance of one Service is often related to the performance of a related Service. The Serviceflows **specify** the **dependencies** between Services. The Authorisation of the Service Contracts can be viewed as a dependant *Authorisation Service*.

#### 1.4.4 Introducing the Seven General Concepts

As mentioned at 1.1, the Seven General Concepts integral to the dissertation are Services, Groups, Service Contracts, Privacy, Consent, Accountability and Authorisation. Figure 3 locates the *Seven General Concepts* at the zenith. Each of the General Concepts has a number of *Specific Concepts* listed below it; these are *Sub-Concepts* of the General Concept. The first Concepts in each list (those with thick borders) together form the seven *Foundational Model Concepts*. The Seven General Concepts are subdivided into two groups: the three *Component Concepts* (Services, Groups and Service Contracts) and the four *Compliance Concepts* (Privacy, Consent, Accountability and Authorisation). The Component Concepts are defined in each of the three following sections while the Compliance Concepts are defined in the subsequent section.

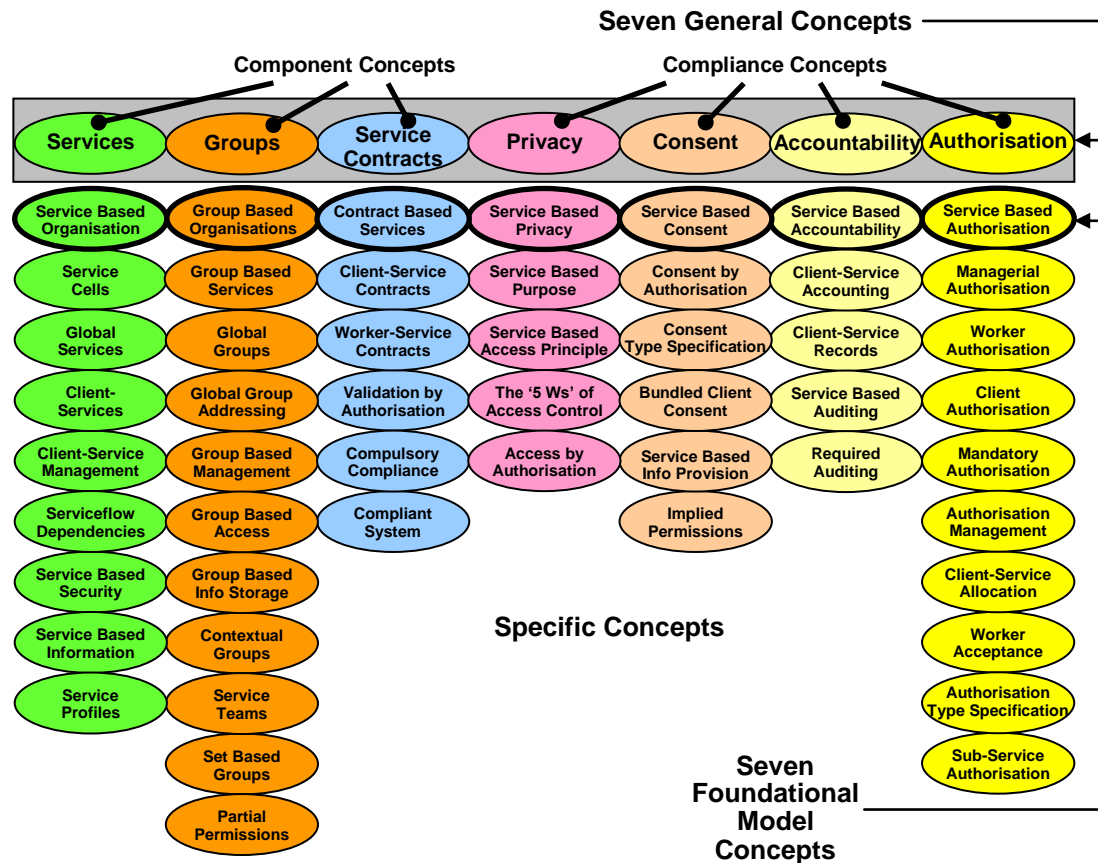


Figure 3: Model Concepts

Components that are *abstractions* of the real world<sup>8</sup> are used in software programming to *model* real world entities and interactions. The real world is thus modelled in the machine world. The Model defines Service, Group and Service Contract Components. This means that Services, Groups and Service Contracts exist both as real world entities and corresponding software entities in the Model.

In Object Oriented Programming (OOP) the world is viewed in terms of Objects that interact with each other. In the Model, Organisational work is viewed as Services that are performed by Groups within the Organisation and managed through Service Contracts. CMS software is based on three basic Components (the *Model Components*): one for Services, one for Groups and one for Service Contracts. In

<sup>8</sup> The term “real world” is used in the dissertation in the sense that the IT system needs to model entities (with “abstractions”) and processes that occur in organisations in particular, and in the world more generally.

OOP Services, Groups and Service Contracts would each be represented by an Object Class, that is, each would be a type of Object.

**The Service, Group and Service Contract Components embody the way that the IT System relates to the real world.** The IT System only understands work in terms of “Services”, people in terms of what “Groups” they belong to, and dealings in terms of “Service Contract” offers and acceptances. For example, the Hospital IT System has information about Alice’s admission last Tuesday because the “Patient Admission” Service was offered, accepted and performed for Alice. It knows that Bob is a doctor because he is a member of the “Doctor” Group and it knows that Bob is caring for Alice because he is in “Alice’s Care Team” Group.

The Model proposes Generic Service Contracts, which can be managed by the IT System and have the potential to be applied globally. Service Contracts encompass all the legal agreements that are necessary between the parties involved in service provision. Their primary purpose is to ensure Organisational Compliance.

From the above it is apparent that the IT System must be capable of relating to users through its “view” of the world. Yet an IT system, not being human, can only hold a “logical view” and utilise “logical processes”. **The reason why the Services, Groups and Contracts Components are useful is that persons understand what Services, Groups and Contracts are (that is, they can adopt this logical view), and in turn can communicate with the IT System by using Service/Group/Contract-based processes (that is, they can adopt the logical processes).** This is in contrast to Objects in OOP, where computer programmers are the only persons that understand the “Object” concept. Currently many IT tasks can only be performed by IT system administrators because only they know how to perform them. By utilising Component Types that Managers, Workers and Clients understand, the Model aims to enable these persons to perform IT tasks (albeit in a new way and often automatically) currently only performed by IT system administrators.

The Compliance Concepts — Privacy, Consent, Accountability and Authorisation — each represent an area of Organisational compliance requirements. In the Model each has a particular relationship to Services and Service Contracts.

The primary purpose of each of the Model's General Concepts is that:

1. **Services** are templates for standard business process tasks in Organisations with each Client-Service instance being managed by the IT System,
2. **Groups** represent Organisational Structure by defining collections of resources and are used to control who performs each Client-Service,
3. **Service Contracts** enable Compliance by fulfilling Privacy, Consent, Accountability and Authorisation requirements in a single digital Service agreement,
4. Client **Privacy** is enabled by restricting access to information to that which is necessary for the Client-Service being provided,
5. Client **Consent** is incorporated into Service Contracts and managed in Service processes,
6. **Accountability** requirements are addressed in Service Contracts and functionalities in Service processes, and
7. **Authorisation**<sup>9</sup> mechanisms are the means by which Service Contract offers and acceptances are input into the IT System during the Service process.

The following describes the General Concepts, albeit still in an introductory sense. Greater detail follows, particularly in Chapters 4 and 5. The reasons why these concepts are integral are not discussed at this early stage. Introducing the General Concepts here serves to introduce the terminology and to specify definitions that may vary from those made in other research methodologies.

---

<sup>9</sup> An "Authorisation" in this sense *is* an "offer" or "acceptance" received from a Service Contract party in the form of an "Authorisation Request" that is "input" (or entered) into the IT management system.



## 1.5 Services

*Services are templates for standard business process tasks in Organisations with each Client-Service instance being managed by the IT System.*

Services represent distinct units of work (also known as jobs or tasks) performed by Organisations. The Model defines Services with the following basic *Service Properties*:

1. They are **All-Encompassing**,
2. They apply to **Standardized Parties**,
3. They consist of **Standardized Processes**,
4. They have **Commercial Aspects**, and
5. They are managed by a **Compliant Management System (CMS)**.

The following subsections deal with each of these Service Properties.

### 1.5.1 All-Encompassing Services

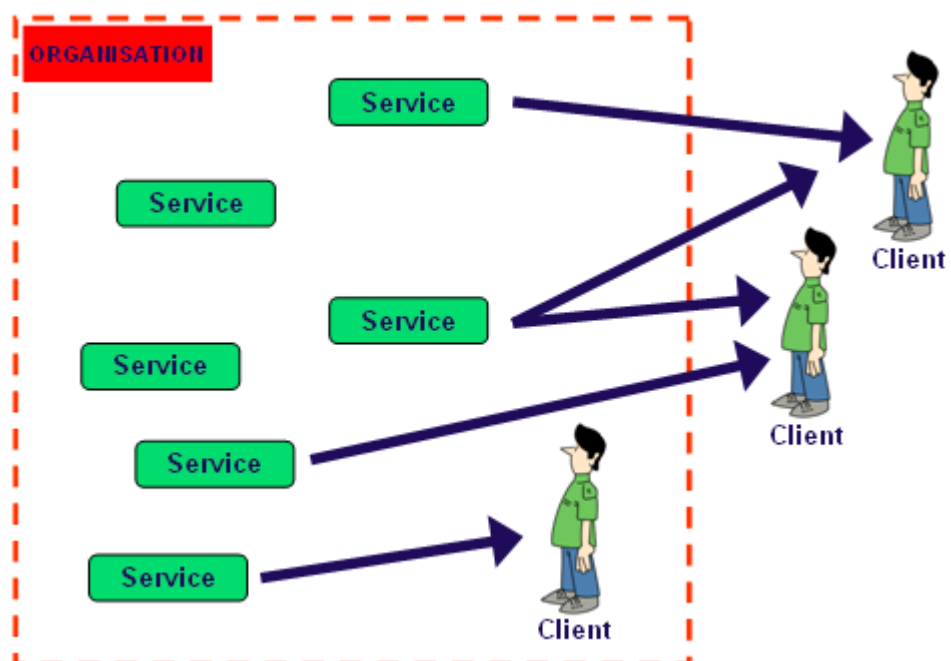
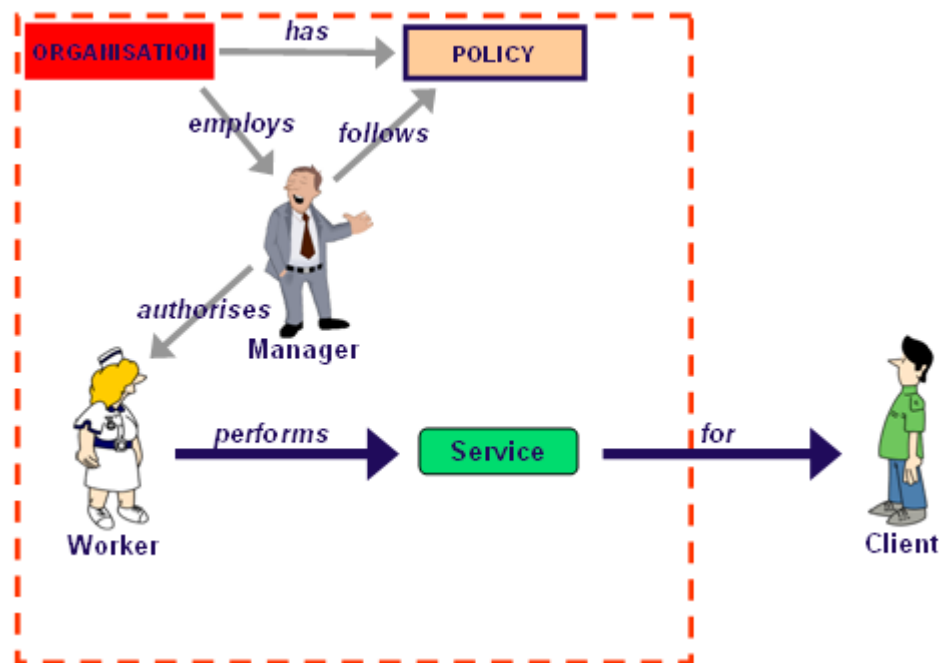


Figure 4: A Service Based Organisation

The Model uses the premise that Organisations essentially exist to provide Services to Clients who will usually pay for the Service. However, some of the work will be for Clients who are within the Organisation. All the work that is performed by the Organisation is defined by Services. In this sense Services are “units of work”.

The Model considers the provision of goods (products) as a Service. For example, the purchase of a book from a retailer is considered to be a “Retail Service”. The purchaser of the book is paying not just for the book itself, but also for the retailer’s advice, payment processing, delivery, accounting and any other associated work. There is no distinction therefore between “goods” and “services”.

### 1.5.2 Standardized Service Parties



**Figure 5: Basic Organisational Concept**

Figure 5 shows the basic parties and relationships that exist in the provision of a Service. The dashed red square represents the Organisation. The Organisation has a Policy outlining the way work is to be performed within the Organisation. Managers employed by the organisation must follow the Organisation’s policy. A Manager Authorises a Worker to perform a Service for a Client, who typically is outside the Organisation. In reality the individual items and relationships may vary on the surface but the basic concepts remain constant. Examples of variations could be that

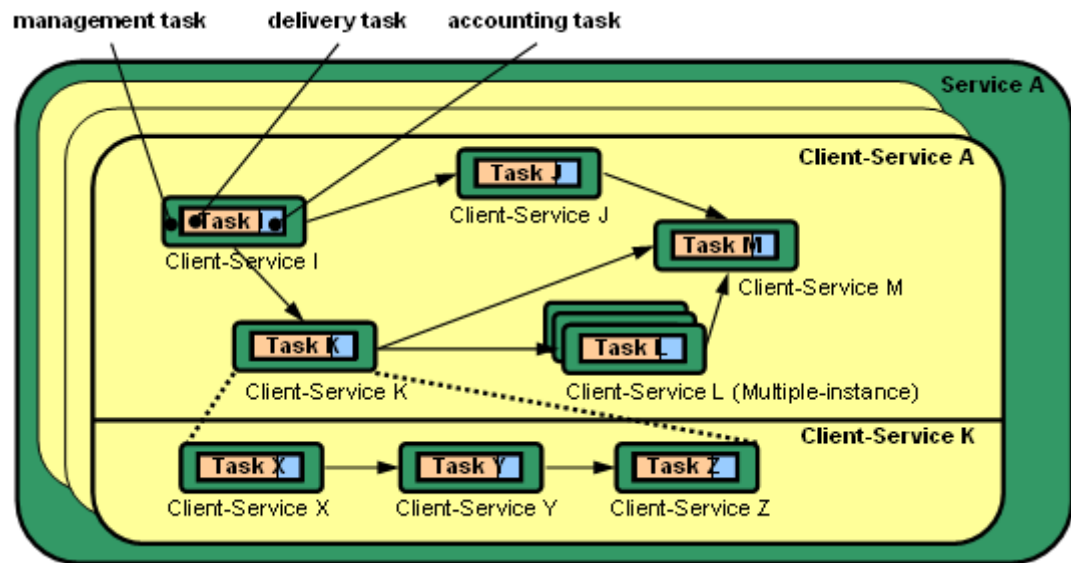
the Client is within the Organisation (for *Internal Services*), the Policy may be verbal rather than written, a Manager may also perform the duties of a Worker, the Worker could be a computer program, or the Service could be the sale of a product. Despite potential variations, the items, parties and relationships always exist in some definite form.

The Model centres on the *Organisational Premise* that “**a Manager Authorises a Worker to perform a Service for a Client**”. This phrase concisely summarises the relationships between the three fundamental parties involved in Service Contracts and Service Delivery.

### 1.5.3 Standardized Service Processes

Consider the work done in an Organisation in BPM terms where there are business processes and associated business tasks. There are essentially three types of business tasks: *delivery tasks*, *management tasks* and *accounting tasks*. A delivery task is the actual work that is done by workers in providing the Service; for example, a nurse administering prescribed medication to a patient. A management task is the work that is done by managers and workers in making decisions about the Services that are delivered; for example, deciding which nurse will administer the medication. An accounting task is the work that is done in recording information about the task; say, that the medication has been administered.

Consistent with the foregoing, in the Model a Service is composed of a management task, a delivery task and an accounting task. In Figure 6 the tasks are shown as Task I, Task J etc. The management tasks are represented by the dark green areas surrounding the delivery (tan) and accounting (light blue) tasks. The delivery and accounting tasks are shown alongside one another because they are usually performed by the Worker, whereas the management task is performed by the Manager.



**Figure 6: A Service and its Sub-Components**

Individual Services can be components of higher level Services. In this sense the higher level Service is the “Service” and the individual component Services are Sub-Services. The timing of Sub-Services and the dependencies between Sub-Services are defined by the *Serviceflow*. This is synonymous with the “Workflow” (BPM terminology) for the Service. There is no limit to the number of levels; individual Sub-Services can in turn have component Sub-Services. In Figure 6 Services I, J, K, L, and M are Sub-Services of Service A, while Services X, Y, and Z are Sub-Services of Service K.

In the Model all work is performed for a Client. Clients can be *External Clients* (who are outside the organisation) or *Internal Clients* (within the organisation). Work is performed by invoking the appropriate Service. This creates an instance of the Service called a Client-Service. Services can be performed for different Clients. Each time a new Client-Service instance is created. A Service can also be performed multiple times for the same client. This is a *Multiple-Instance Client-Service*. In this sense the Service is like a template and each Client-Service is one copy made from the template. This is akin to individual cakes being baked using the same recipe.

#### 1.5.4 Commercial Service Aspects

In both BPM and SOA only the work-related aspects of Tasks and Services are normally considered. In other words, BPM and SOA Services are primarily thought

of as units of work that must be done. The Model adds a “commercial dimension” by considering that Services have value, can be advertised, and can be bought and sold. The Model therefore takes a *Client-Centric Approach* to Services (where all Service instances have a Client) as well as a *Service Management* approach (where all Service Instances are managed by the IT System).

For example, consider the Service of purchasing a new motor vehicle. The manufacturer can design a “Vehicle X” Service for a motor vehicle model that contains all the necessary information including specifications and photographs. Retailers (suppliers) that stock the vehicle can use the Service information directly in their advertising and catalogues, attaching their own details and pricing information. A customer is then able to search online for the vehicle, to locate all nearby suppliers, to compare details and costs, and even purchase the vehicle.

Once the Service has been purchased by the customer, the Service Management related Components of the IT System deal with the processing of the *order* and *supply* of the vehicle.

### **1.5.5 Compliant Management System**

The Model advocated in the dissertation, while having unique features, proposes a CMS that incorporates BPM, SOA and RBAC functionalities. The CMS employs Groups to control access to Services where the Services are essentially business process tasks. The inefficiencies of RBAC are overcome by incorporating BPM and SOA functionalities in the CMS. This is achieved by automating access control decisions. Instead of IT system administrators having to make these decisions by manually granting access permissions, the permissions are automatically triggered by managerial decisions made at the coalface. For example, when a Manager allocates a client’s job to a Worker, the Worker is automatically given access to the relevant Client information.

The CMS provides efficiency gains through the *automation* of manual tasks and through the *reuse* of generic tasks and information. In this context a task or “Business Task” is a specific item/piece of work.

### **1.5.6 Enabling Service Automation and Reuse**

Computers can only handle systematic processes and can only make decisions based on logic. Therefore, to attain the automation and reuse benefits of an IT management system there is a fundamental need to standardise and quantify business tasks. In other words, to gain time saving IT benefits, persons need to work in ways that computers can relate to. They have to work systematically and logically.

Delivery tasks, management tasks, and accounting tasks can all be automated. For delivery tasks to be automated the task must be broken down into basic components (sub-tasks) which are carried out in a logical and repeatable order. In other words, a standard procedure for doing the task must be found. For example, if the task is to bake a cake, the work is divided into sub-tasks like “beat the eggs” or “mix the ingredients in a bowl”. To program a robot to bake the cake, the robot would be programmed to perform each sub-task and then programmed to perform the different sub-tasks in the appropriate sequence. In a sense it is a similar process to teaching a child to bake. The child is taught how to do each sub-task and then how to do them in sequence.

Another important aspect of automation is representing the state of the tasks by measureable values. This amounts to quantifying the state. This needs to be done so that the robot can decide when a task is complete. For example, for the egg-beating sub-task the “beating time” or the mixture’s “colour variation” could be measured. The robot could then be programmed to “beat the eggs for 60 seconds” or “beat the eggs until the colour variation is zero”.

Once the sub-task programs are written, the robot can easily be programmed to bake other types of cakes by varying individual sub-task operations or by varying the sequence of the sub-tasks. This involves the reuse of the sub-task programs.

Management decisions can also be automated. This likewise requires a systematic and quantified approach. Automated decisions are made when a particular state is reached or the decision is made according to the particular state. Consider the management decision of deciding how many staff to allocate for a day’s work in a shop. A human manager could make a qualified decision like “I reckon that four

staff will be adequate tomorrow”. A computer cannot make a qualified decision; it can only make a quantified decision. For example, based on previous turnover data, it could calculate an anticipated turnover from the day of the week and the predicted weather, and then estimate the number of staff required from the anticipated turnover.

Another factor in management decisions is that often decisions relate to more than one specific task. For example, in the case of the nurse administering patient medication, there are a number of management decisions that are made. First, at the time of recruitment the worker (the nurse) is assigned a nurse role in the organisation. Secondly, the nurse is assigned to a particular hospital ward. Thirdly, the nurse is assigned to do a particular shift. Fourthly, the nurse is assigned to look after the particular patient. Fifthly, the medication is prescribed to be given at a particular time. Automating the management of such tasks requires systematic procedures and quantified data about personnel allocation and the state of the tasks.

Given the necessary information, accounting tasks can also be automated. As accounting practices are mathematically based and systematic, computers are well suited to handling them.

The core notion here is that business tasks must be systematic and quantified. The Model handles this in a simple way. Services represent sequenced sets of sub-tasks, Groups represent organisational components, and Service Contracts represent agreements about Service provision. Data relating to specific Services, Groups and Service Contracts represent the state of the organisation and its activities.

While automation and a systematic approach are necessary, it should also be noted that flexibility is required in local systems to meet specific requirements. This means that systems and automation need to have a degree of customisability.

## 1.6 Groups

***Groups** represent Organisational Structure by defining collections of resources and are used to control who performs each Client-Service.*

Groups are primarily used to define and manage the roles of personnel in organisations. However, they can also define and manage collections of non-human resources (for example, computers or rooms). The Model defines Groups with the following basic *Group Properties*:

1. They are used to **Define and Manage Resources**,
2. They are **Hierarchical**,
3. They are **Globally Addressable**,
4. They are instances of a **Group Template**,
5. They have **Types, Instances and Sub-Types**, and
6. They are **Associated with Services**.

The following subsections deal with each of these Group Properties.

### 1.6.1 Groups used to Define and Manage Resources

Most commonly Groups can be used to represent persons with something in common (for instance, members of a “Doctors” Group have a common medical qualification). More generally, they are used to represent *sets* of any *resource*. *Sets* in a mathematical/IT sense are collections of objects of the same class. *Resources* in a Business Management sense are Organisational assets, and include “human resources”.

In the Model Groups have two functions: they define sets of resources and they define how the resource set is managed. Resource sets are defined by specifying the “members” of the set. Often just knowing the members is sufficient, say, with a “personnel set” to enable the choice of a member for a task. There are occasions however, when other data about the individual members is needed. For example, it



may be useful to know the birth date of members, so that they can be ordered or so that a member can be chosen within a desired age range. In this sense a Group is “a set of member resources with group related membership data”.

The second Group function defined in the Model relates to the management of the resource set. In practice this means that there are some personnel assigned to manage the group. The main function of Group Managers is to determine when to add and remove members of the group. Group Managers may also have access to relevant membership data. Managers may have different management roles. For example, if the Group represented a committee, there may be separate management groups for a chairman, a secretary and a treasurer. In this sense a Group is “a set of resources with associated management groups”.

Combining these two Group functions a Group can be defined simply as:

*A **Group** is a resource set with related membership data and Management Groups.*

This definition can be referred to as the *Simple Group Definition*.

The following diagram, Figure 7, shows a Personnel Group and its components. While some of the items in the diagram have been described in the foregoing paragraphs, other items will be described later. The light blue circles in the diagram represent the membership components of the Groups while the surrounding dark blue areas represent the management components of the Groups.

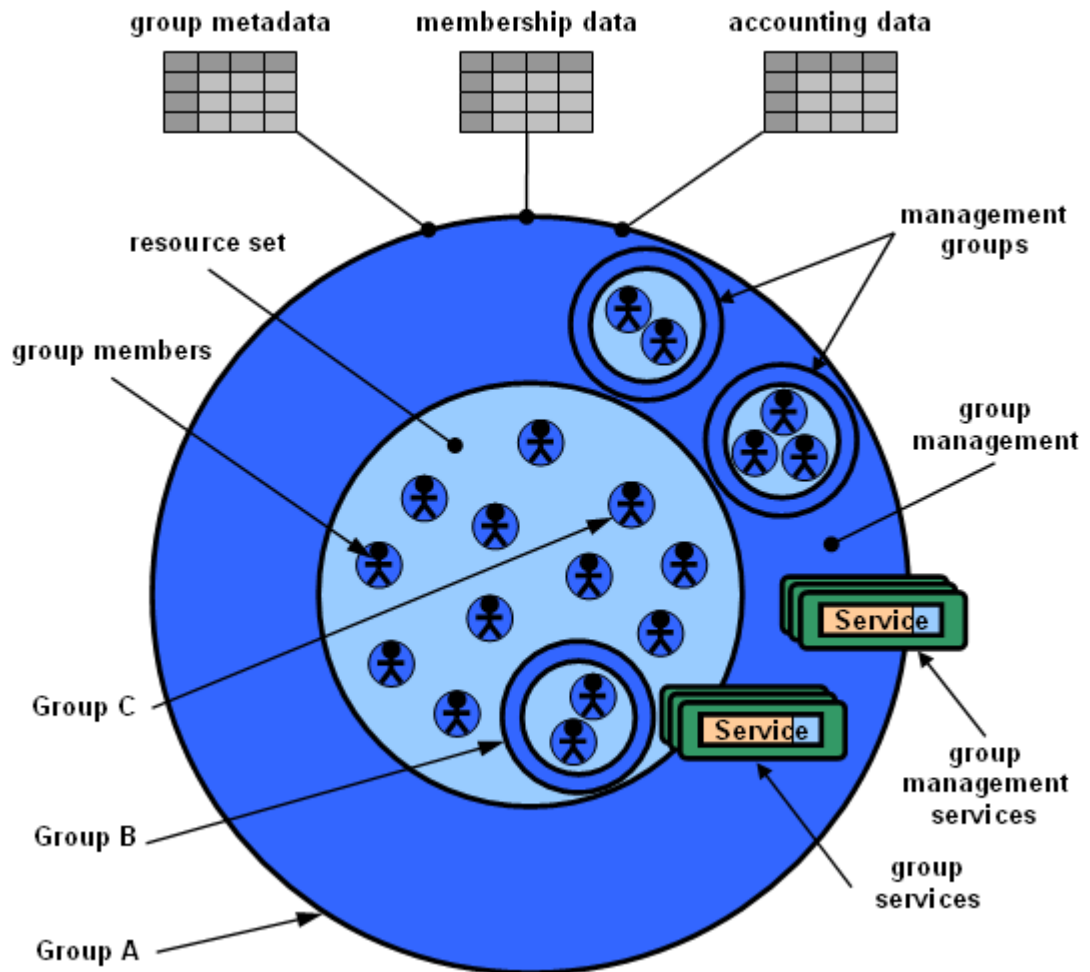


Figure 7: A Personnel Group and its Sub-Components

### 1.6.2 Hierarchical Groups

Any Group can be a Sub-Group of another Group. Accordingly, Groups may form hierarchical Group structures. The membership data for a group records the Sub-Groups of the Group. The group metadata records a Group's Super-Groups (that is, the Groups of which it is a member). In Figure 7, Group A is the Personnel Group. Groups B and C are Sub-Groups of Group A.

Group C is a special kind of Group called a *Base Level Group* (BLG). The Model uses a Set Based Group definition where Groups have '0' or more members and BLGs have exactly '0' members. In this strict sense, BLGs have no members because they merely hold information (stored as group data) about the single resource they represent. In Figure 7 the BLGs, like Group C, are depicted with a

stick figure to symbolise a single human resource and they have no light blue region within them for group members.<sup>10</sup>

The foregoing can be summarised into two *Group Rules*:

1. **All resources have their own Base Level Group (BLG), and**
2. **The members of all Groups are other Groups.**

This means, particularly from an IT perspective, that the System sees all Organisational resources, both human and non-human, whether singular or collective, as Groups. A key quality of Groups is that they have ‘members’ and that ‘rules of membership’ can be defined functionally.

### 1.6.3 Globally Addressable Groups

Groups in the Model differ from “groups” in existing methodologies in that they have a *Global Group Address (GGA)*. This is similar to a web-page’s IP (Internet Protocol) address which is a number of the form specified in the protocol (for example, 218.76.155.6). As with an IP address, the GGA locates a specific set of memory bits (‘0’s or ‘1’s stored in individual electronic components) in a specific computer. This can be likened to a house address that enables a particular house to be found anywhere in the world. In the case of a web-page, the memory bits store the information that generates the web-page. In the case of a Group, the memory bits store the information that represents the Group.

The basic purpose for making Groups globally addressable is to enable Groups to be accessed globally. This enables remote access to Services and associated data. For example, a doctor treating a tourist can search the tourist’s remote medical record because their legitimacy to access the information can be proven by their relevant Group membership. This is possible because the Group Rules enable a Globally

---

<sup>10</sup> It is envisaged that in the IT design, Group membership will be stored as a list (or similar data structure) of pointers to lower level groups. Group hierarchies will be represented as trees (strictly graphs) where the leaf nodes have null pointers. So that all groups are functionally the same and only one Group “class” is required, data in BLGs will also be stored as Group data. The concept of BLGs is used to portray the (management) idea that at some level Groups represent single resources.

Addressable “Doctor” Group in one hospital (or state/country) to be a member of the “Remote Doctors” Group in the facility where the patient records are held. This demonstrates how the Model enables credentials to be recognised in a simple way across domain and jurisdictional boundaries.

#### **1.6.4 Group Templates**

As with Services, Groups are generated from *Group Templates*. This means that a New Group can be formed by copying an existing Group Template. While this can be effected on the level of individual Groups, it is possible to copy whole Group Hierarchies. The latter would represent entire Organisations or part(s) of larger Organisations. This simplifies the task of setting up IT Systems for new enterprises.

Group Templates contain the *Group Metadata* and *Group Functionalities*. Group Metadata describes the data types of Group members. Group Functionalities essentially describe the means for granting and revoking Group membership, and are expressed as *Group Management Services*.

Once a Group has been copied from a Group Template, it can be customised (if necessary) to meet individual Organisational needs. New Group Templates can also easily be created if the new Customised Groups need to be replicated.

An example of a Group Template is a Committee Group Template, containing all the necessary Sub-Groups and Management Groups for the various committee executives.

#### **1.6.5 Group Types and Sub-Types**

Resources can be classified into different categories, for example, human resources, venues and IT resources. The Model represents each category as a *Group Type*. Categories can be divided into sub-categories, for example, human resources could be divided into different Organisational roles. The Model represents these sub-categories as *Group Sub-Types*. These sub-divisions can be replicated any number of times. This means that Group Types, like Groups themselves are defined in hierarchical structures. For example, a type of human resource Group may be a

Project Team, and there may be different types of Project Teams (with different Group Metadata and Group Functionalities).

The main purpose of Group Types is to enable the classification of Group Templates. Labelling Group Templates with their Group Type allows appropriate templates to be found when a New Group is required.

### 1.6.6 Groups Associated with Services

Groups and Services are the first two Component Types used in the Model. The state of Groups and Services (and Service Contracts) defines the state of the Organisation. In the Model Services are associated with Groups, the primary purpose being to control access to Services. This functions in a fairly basic way: a user who is a member of a Group is allowed to use the Services that are associated with that Group. For example, a hospital Worker who is in the “Nurse” Group can perform the “Administer Medication” Service. Organisation Management is therefore enabled to restrict Workers who perform Services to those who are appropriately qualified.

## 1.7 Service Contracts

*A **Service Contract** is a digital Service agreement that enables Organisational Compliance by fulfilling Privacy, Consent and Accountability Regulations and CMS Authorisation requirements.*

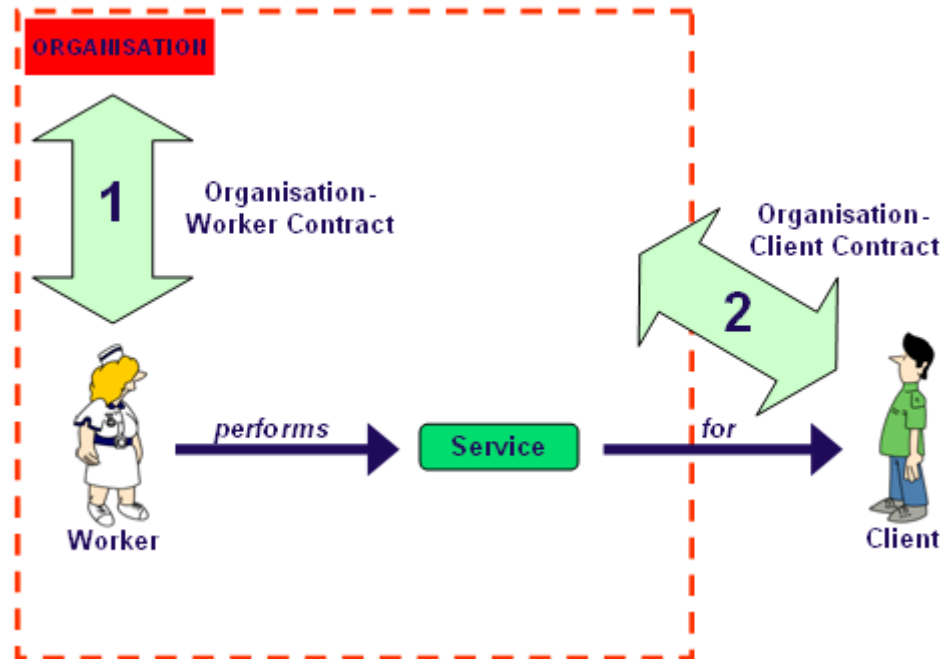
Service Contracts are contracts associated with the provision of Organisational Services. The Model defines Service Contracts (Digital Service Contracts or Contracts) with the following basic *Service Contract Properties*:

1. They are **Organisation Based**,
2. They are **Single-Service**,
3. They recognize **Sub-Services**,
4. They are **Generic** in nature, and

## 5. They are **Digitally Managed**.

The following subsections deal with each of these Service Contract Properties.

### 1.7.1 Organisation Based Contracts



**Figure 8: Service Contract Types**

On the organisational level, the Model defines two contract types relating to the delivery of Services (see Figure 8). The first is between the Organisation and the Worker and involves a contract to perform work (Client-Services) for the Organisation. The second is between the Organisation and the Client and involves a contract for the delivery of Organisational Services to the Client.

### 1.7.2 Single-Service Contracts

As mentioned previously (see 1.3.4.4), the Model is designed to utilize both *local* and *remote* services. One of the problems with remote services in particular is the “one-off nature” of the service provision. For example, if a person purchases a book online through a brokerage service such as Amazon, the relationship with the actual supplier of the book will often be solely for the supply of that book. To deal with this problem the Model proposes that all Services utilize “one-off contracts”. This means that each Service has its own *Single-Service Contract*.

### 1.7.3 Sub-Service Contracts

There are two types of Sub-Services: *Unassociated* and *Associated*. *Unassociated Sub-Services* relate to Sub-Services that are independent of each other whereas *Associated Sub-Services* relate to Sub-Services that are components of a Service (see Figure 6).

An employment contract between an Organisation and a Worker can be considered as a Service Contract composed of Unassociated Sub-Services. Each job that the Worker undertakes is a separate Sub-Service. In a contractual sense each job is independent of the other jobs, each having its own Single-Service Contract. Any long term contracts, such as employment contracts, that involve participants in Single-Service Contracts must recognize the existence of these Single-Service Contracts. In particular, it is envisaged that *Service Based Employment Contracts* recognise each Single-Service Contract as a separate Sub-Contract.

Where Services are comprised of Associated Sub-Services, the Single-Service Contract associated with the Service must recognize the existence of Sub-Contracts for each Sub-Service.

It should be understood that both Unassociated and Associated Sub-Services do not necessarily involve the same persons or entities. For example, a Travel Service provided by a Travel Agent may be comprised of Sub-Services provided by other organisations, or a Treatment Service provided to a hospital patient may be comprised of Sub-Services that are performed by different Workers.

### 1.7.4 Generic Service Contracts

The Model envisages Service Contracts that are generic in nature. An example of a generic service contract is the standard Contract for the Sale of Land produced by the respective Real Estate Institutes in each Australian jurisdiction. These are used by many real estate agents as contracts between vendors and purchasers, subject to modification in individual instances.

Generic Service Contracts offer multiple advantages. First, they are the product of expertise and experience, and their interpretation will likely have been the subject of

judicial observations. Consequently, they can be perceived as being reliable in that unforeseen consequences should be minimised. Secondly, they are likely to prove easier to administer, as those administering the contracts gain experience in their use. Thirdly, reuse of the same contract reduces the cost per use due to economies of scale. Fourthly, advice and information regarding the contracts is more likely to be readily available to those not experienced in their use.

### 1.7.5 Digitally Managed Contracts

In practice contracts take a number of forms. They can be entered into by verbal, written or digital means. “Digital” in this sense means “in a form that is understandable by a computer” (computers are digital devices where all information is stored as ‘1’s and ‘0’s).

In order for compliance to be achieved, the Model requires that all Service Contracts be managed and Authorised by the IT System. In other words, all contracts must be “digitally managed”. A *Digitally Managed Contract (DMC)* is defined here as:

*A **Digitally Managed Contract** is a contract where all the evidence that is required to enforce the contract is either stored digitally or the location where it can be readily retrieved is stored digitally.*

The ultimate goal from an IT sense is that all aspects of a contract be stored and accessed digitally through the IT System, with no need for paper-based forms and information. This would aid accessibility. Such a contract is a *Purely Digital Contract (PDC)*. A PDC is a form of DMC where all the *information* and the *authorisations* (offers and acceptances) are recorded digitally. In other words, all the information necessary for the contract to be legitimate is available to the contract parties in digital form and offers and acceptances are recorded digitally (say, by a party clicking an “I agree” button on a computer screen).

Reiterating, the Model only requires DMCs in order for compliance to be fulfilled, not PDCs. While it would make management easier if all Service Contracts were PDCs, by only requiring DMCs the Model is more flexible as it can deal with other



forms of evidence, including paper-based documents. The Model can therefore be applied to systems that rely on paper-based documentation.

### 1.7.6 Digital Service Contracts Enabling Compliance

**The main purpose of having Service Contracts is to enable compliance.** If an IT system utilises DMCs, it can be designed to be compliant. **CMS Systems use DMCs called *Digitally Service Contracts (DSCs)*.**

Regulations are rarely static, and regulation differs between jurisdictions. Consequently, the Model does not directly embody individual regulations. Rather, it seeks to embody key regulatory concepts within Digital Service Contracts. These target the General Concepts of Privacy, Consent, Accountability and Authorisation.

On an organisational level the goal is for the IT System to ensure that an organisation is compliant with regulation. This is achieved by the IT System managing and monitoring Service provision in a way that ensures that the organisation fulfils its part of each Service Contract. Compliance does not mean that no mistakes or unlawful activities will be committed by individuals in the Organisation, but that adequate checks and balances are in place. In practice the goal is not only to minimise mistakes and unlawful activities but to provide evidence to protect the Organisation against false accusations. For example, the IT System would seek to minimise medical errors such as drug overdoses by ensuring staff do appropriate checks and protect the Organisation from false accusations by providing Service related data as evidence that correct procedure was followed. On a jurisdictional (national or state) level, Service Contracts need to be accepted as legally binding. And in the long term, once it can be shown that compliant and efficient business management systems are widely available, their use could be mandated in some way.

Globally, it remains to individual jurisdictions (whether at national or state level) to recognise these Service Contracts as being legal within their jurisdiction. Where multiple jurisdictions recognise their legality, Services performed between the jurisdictions can be legally protected. For example, if an individual in one country pays for a product from a business in another country and it is not received, the

individual may have legal redress if both countries recognise the legitimacy of the Service Contract.

The Compliance Concepts outlined in the following section are all enabled in the Model through the use of Digital Service Contracts. DSCs manage Privacy, Consent and Accountability and are controlled through Authorisations. This is highlighted later in Figure 31.

## 1.8 Compliance Concepts

### 1.8.1 Privacy

*Client **Privacy** is enabled by restricting access to information to that which is necessary for the Client-Service being provided.*

The dissertation deals with “information privacy”, particularly the responsibilities of organisations to keep client information private. Information privacy relates to the *rights* and *abilities* of an individual or organisation to control the collection, storage, sharing and dissemination of personal information. The *rights to control* are dependent on legislation and the *ability to control* on the system used to manage the information. The advent of the Internet and the explosion in the amount of personal data that is stored digitally has heightened the need to develop laws and systems that protect Personally Identifiable Data (PID).

Although organisations have adopted privacy policies, and these may indeed be of value, there remain challenges in ensuring that these policies are effective in the timely detection of privacy breaches, and in then ensuring appropriate sanctions are imposed.

Privacy legislation has been enacted across the Australian legal landscape, but there remain challenges in ensuring that the legislation maintains currency with developments in IT technology and offers effective mechanisms to deal with evolving threats to IT based systems.

The Model aims to enable the development of IT systems that are compliant with privacy law. The main method proposed for achieving this is based on the idea that Managers and Workers should only have access to client information on a “need-to-know” basis. This method relies on the notion that only the PID necessary to perform a Service should be accessed; that Client-Services be performed by appropriate personnel; and that comprehensive Client-Service records be kept and monitored to detect and deter privacy breaches.

### 1.8.2 Consent

*Client **Consent** is incorporated into Service Contracts and managed in Service processes.*

The Model deals with “Client Consent” and encapsulates the idea that Client Consent is required for all Services received by a client and for all accesses to a client’s PID.<sup>11</sup> It simplifies consent management by exploiting the fact that each Service has an inherent purpose that defines required PID disclosures, and by incorporating the act of granting consent to access PID into the act of consenting to receive a Service.

Client consent is of different types (Clarke 2002, ; HealthConnect 2002, ; Coiera & Clarke 2004), and encompasses written, verbal and implied consent. For consent to be legitimate the client must be informed of the potential consequences of their acceptance, including disclosures of their PID.

The Model requires that:

1. The type(s) of consent required for each Service be specified,
2. Consent be obtained for all Services,
3. Consent for a Service implies consent to access Service-related PID, and
4. The Client be informed of the implications of their consent.

---

<sup>11</sup> Excepting instances where specific overriding laws permit authorities to access information, for example, as part of a criminal investigation.

In many situations it is not practical to record that consent has been given. For example, in a hospital a patient may consent to (verbal or implied) having their temperature taken (a Service). The nurse will not ask the patient if they can enter the reading into their medical record. This is a part of the Service of temperature taking. The Model proposes to deal with consent in cases like this by assuming that consent is given if a temperature reading is entered, or that consent is refused if a “consent refused” item is checked or selected.

It is envisaged that policy experts and legislators will play a role in specifying what constitutes effective consent for particular Services. The Model does not specify these domain specific requirements; it simply proposes that the requirements be specified for all Services. In addition, it proposes that Workers be informed of the form of consent required for a Service and that the Client/patient be informed of their rights with regard to consent. Consent refusals should be recorded so that if the patient later dies or suffers loss as a result of refusing a Service the System has a record of their refusal. The legal advantages of having an auditable consent record are obvious.

Services can be designed to specify where written consent is required and prompt workers to obtain it. Also, it may also be possible in the future to have systems that digitally record verbal consent, accept digital signatures or use some biometric sign, such as a fingerprint scan, to signify consent.

### **1.8.3 Accountability**

*Accountability relates to meeting accounting-related Regulations through the use of appropriate Service Contracts and Service Delivery processes.*

Accountability has to do with being answerable and responsible to someone for actions taken (or not taken). To be accountable, an organisation must maintain adequate records. Essentially, an organisation should be able to show *who* did *what* and *when*, *where* and *why* they did it (defined here as the “5Ws”). For example, if a patient has a procedure in a hospital, apart from the obvious need for financial records, the hospital should be able to prove from their records:

1. *who* performed the procedure,
2. *what* the procedure entailed,
3. *when* the procedure was performed,
4. *where* the procedure was performed, and
5. *why* the procedure was performed.

While there are a variety of accountability requirements applicable in different situations and at different times, the 5Ws provide a generic idea of what constitutes accountability with regard to Services. In a service-based IT system the challenge is to capture all the information required for accountability at the time that decisions are made and tasks are performed. There is great advantage if the majority of this information can be captured automatically without the need for Managers and Workers to spend time doing purely administrative tasks. The Model proposes that this information can be collected through recording necessary data associated with Services and Groups. As well as to enable compliance, the Model is also concerned with providing accountability in an efficient manner by utilising automation.

#### **1.8.4 Authorisation**

*Authorisation mechanisms are the means by which Service Contract offers and acceptances are input into the IT System during Service allocation and delivery processes.*

Authorisation in a general sense means to permit someone to do something with official sanction. In an IT system authorisation takes the form of permitting access to resources (by employing Access Control techniques). For example, a nurse is authorised to access their patients' records.

The challenge is to input real world *Authorisation Requests* into the IT System. In the case of the nurse being authorised, a Manager/supervisor makes a decision to request authorisation, which request is entered into the IT system. In the Model such a request is “captured” when the Manager allocates (usually at the start of a shift) the nurse to care for particular patients. For the IT System to be informed of the

Authorisation Request, the allocation must be recorded by the IT System. It is not sufficient for the nurse to be verbally told that they are looking after particular patients; this alone will not permit access to the relevant patient records.

From an IT and Business Management perspective an Authorisation is not made by humans; it is made by the computer in response to a request from a human. The input act therefore amounts to a request for Authorisation. In the Model, **Authorisation is deemed legal only when the IT System has accepted the input Authorisation Request.** The IT System may deny the Authorisation or make it conditional in some way. The IT System is the business management system and it is in charge unless and until it is overridden. This has a parallel with a computer being in charge of flying a plane unless the pilot takes over. Reiterating Michael Gerber's (Gerber 1995, pp. 91-92) statement, "[t]he system runs the business. The people run the system...".

It should be noted that the aim here is to gain the advantages of IT automation. In practice, in allocating a patient to a nurse the Manager is not necessarily aware that the IT System is making access control decisions; he or she is simply performing a management task. It is important to understand that a central focus of the Model is the capturing of Service based decisions like this and that the decision inputs form part of the Service Contract. All such decisions are "translated" to Authorisation Requests because the IT System understands what an Authorisation Request is.

#### **1.8.4.1 Service Contracts and Authorisation**

Authorisation has a contractual perspective. In traditional contract negotiations between two parties, agreement is evidenced by one party (the *offeror*) making an offer to the other party (the *offeree*), who accepts that offer according to its terms. In practice, the process of negotiation dictates that there may be multiple offers and counter-offers before acceptance is attained. For example, in negotiations for the sale of a property the parties may make several price offers and counter-offers before agreement is reached. Terms other than the price may also form part of the negotiations. For example, agreement to repair a defect may be a part of the contract, or the contract may be subject to finance being approved. When all

decisions are “informed”, **the Model views Service Contract offers and acceptances as being equivalent** (the “equivalence” factor).

Another issue is the use of *Agents* who act for the parties. An Agent may be given the power to enter into an agreement on behalf of a party. In contract negotiations with Organisations, Managers act as Agents for the Organisation, as Organisations are merely legal entities with no ability to act for themselves. When Agents are engaged they are ordinarily “authorised” by their principal to act in contract negotiations.

Other parties may act for themselves in contract negotiations. For example, Workers may negotiate employment contracts for themselves, and Clients may act for themselves in contract negotiations with Organisations. The Model views parties who act for themselves as “authorising” themselves to make contract decisions. This means that **both the parties to a Service Contract have *Authorisers* who act for them** (the “authoriser” factor).

By combining the equivalence and authoriser factors, **the Model views both parties as Authorising the Service Contract**. From an IT perspective, Service Contracts require an Authorisation Request from both parties before the Authorisation can be accepted and acted upon by the IT System. To enable Compliance it is necessary for the IT System to perform its acceptance (management) role. In practice, this means that the IT System is ensuring that a valid Service Contract is always in place.

Finally, there are two instances where Authorisation Requests are implicit rather than explicit. These involve *Automatic Sub-Service Authorisation Requests* and *Implied Authorisation Requests*. Where Authorisation of a Service Contract is deemed as Authorisation for a Sub-Service Contract, the relevant Service Contract Authorisation Requests will automatically generate the necessary Sub-Service Authorisation Requests. For example, when a patient consents to an operation requiring anaesthesia, the Sub-Services provided by the anaesthetist will also be Authorised. Where implied consent from a Client for a Service is sufficient there will be an Implied Authorisation Request. For example, when a patient self-administers pain killers provided in a hospital, consent is implied and an Implied Authorisation Request will be used when the patient’s record is updated.

In summary, the Model utilises *Digital Service Contracts* that construe the Authorisation Request from each party as the party's offer or acceptance of the Service. As well as Service Delivery terms, Service Contracts also include terms covering Consent for the Service (where this is required for Compliance) and *permission* for associated information use (to fulfil Privacy requirements). The Model therefore aims to unify Service Delivery, consent and privacy agreements in an overriding Service Contract. This means that Service Contracts in the Model are "multi-functional". This distinguishes them from service contracts used in other methodologies, such as SOA, that simply utilise basic service delivery contracts.

#### 1.8.4.2 Authorisation Terminology

The terms "Authorisation" and "Authorise" are used in several senses in the dissertation. Each sense correlates to one of the three main research disciplines.

##### 1. Authorisation in the Sense of a Human-Computer Interaction

For example, "The Manager must Authorise the Client-Service" OR "The Client must click on the Accept button to Authorise the Service". Here the Authorisation is **the act of entering the Authorisation Request into the IT System**. In this sense, the "Authorisation Action" is the focus; the fact that the other party may not Authorise the Service or that the IT System may refuse the Authorisation Request is immaterial. Unless another sense is obvious, this sense is the one used.

##### 2. Authorisation in the Sense of a Business Decision

For example, "The Manager Authorises the Worker to perform the Client-Service by allocating the Worker to the Client". Here the Authorisation relates to **the business decision to allocate organisational (human and physical) resources**. From a Business Management perspective the Manager is performing a management task. This business decision is seen as an Authorisation Request by the IT System. In this sense business management decisions are captured as Authorisation Requests.

##### 3. Authorisation in the Legal/Contractual Sense

For example, "A Worker is Authorised when the IT System sanctions the Authorisation Request". Here **the IT System stores some information that proves**



**the Worker is Authorised.** In other words, the Worker's state is "Authorised" in the IT System with regard to the particular Client-Service. While the vast majority of such Authorisation Requests will be sanctioned, it is possible that the IT System may refuse the Authorisation Request.

## 1.9 Dissertation Structure

The dissertation is divided into seven chapters plus a **References** section and a **Glossary** of CMS Terms and Acronyms. The **Table of Contents** at the beginning is also a useful reference as it contains the major headings in each chapter and may clarify the nature of the dissertation. The following paragraphs outline what each chapter contains and its basic purpose.

The **Introduction** (Chapter 1) outlines the research and describes its nature. It explains some basic concepts and sets the scene for the details covered in the following chapters.

The **Methodology** chapter (Chapter 2) goes through the steps that were used in the development of the Model. It essentially details the research process.

The **Literature Review** (Chapter 3) examines each of the Research Fields that contributed to the development of the Model. The advantages and drawbacks of aspects in each field are discussed. The Design Requirements uncovered in each Research Field are also listed. These requirements form the basis for the design of the Model. The Design Requirement Set can be seen as a preliminary research result.

The **Model Concepts** (Chapter 4) are one of the results of the research. The Model incorporates many different and often independent concepts. The concepts behind the Model are explained in this chapter. Some of these concepts are ones that have been taken from existing research. These are mentioned in Chapter 3. The remaining concepts are new concepts.

The next chapter (Chapter 5) provides the **Model Description**. The focus here is to describe the *functionality* of the model rather than to explain the concepts. The

conceptual aspects were dealt with in the preceding chapter so that the model description remains concise and easy to understand.

In the **Analytical Evaluation** chapter (Chapter 6) the Model is tested against three well known regulatory standards. The first standard is an IT standard, the second is a Business Management standard, and the third is a Legal/Regulatory standard. The process detailed in this chapter involves listing all the relevant requirements from each of the standards. The requirements are then incorporated into a single composite list. Finally the model is tested against each requirement in the composite list. The composite list of standard requirements can be seen as a research result in itself.

The **Results and Discussion** chapter (Chapter 7) highlights the significance of the research and the results obtained. The chapter also summarises the research and the results and gives conclusions regarding various issues.

## Chapter 2

# Methodology

---

## 2.1 Introduction

### 2.1.1 Initial Aim of the Research

The aim of the research is to develop a model rather than a working prototype. The reason for this is that it is better to have a general solution than a specific one simply designed to meet local needs, programming languages and system constraints. There has been a plethora of IT applications and software systems designed for complex domains, such as hospitals. While some of them may have proven useful in local situations, no single general solution to the problems of administering a compliant solution has proven generally useful.

Authorisation in the context of Access Control and Consent Management was the initial focus of the research. However, it became evident that a workable authorisation model had to incorporate Business Management and Legal concepts. This was primarily because it was seen that IT system designers needed to build software in a way that authorisation is captured in normal work processes and that legal issues had to be considered in order to achieve compliance.

The principal characteristics of the Model are that it should be simple and that the Components it employs should be easy to understand. This assists in overcoming the problem that programmers and system designers, though technically proficient, rarely possess a sound understanding of business and workplace practices. It is also a step in seeking to encourage technical people to speak the same language as the users of the software that they write or manage.

### **2.1.2 Background**

The research began as an extension to a previous research project (de la Motte 2004). In that project an Access Control Model called Professional Access Control (PAC), which provided administrative advantages in professional environments, was developed. The project included the discovery and analysis of hospital scenarios in the local public hospital system. It dealt with a system with stored records in an Oracle database. While the PAC model provided some useful insights, it revealed the need for a more comprehensive solution. This research aims to provide that comprehensive solution.

### **2.1.3 The Aim of Developing a Comprehensive and Practical Model**

Practical experience in the PAC Project made it clear that in order to be useful the model needed to utilise the entry of day-to-day procedural decisions to trigger access control settings. Access Control decisions needed to be automated in order to gain the necessary administrative efficiency.

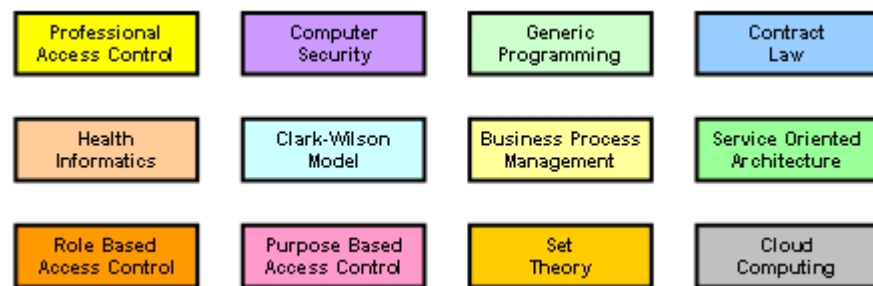
The main problem seemed to be to develop a method for capturing these day-to-day procedural decisions in an efficient and useful way. It was clear that the model needed to be comprehensive and deal with every circumstance where access to stored information was needed by organisational staff. The nature of the problem was to enable efficient access control administration while maintaining the Principle of Least Privilege, which essentially specifies that access should be on a “need-to-know” basis.

Traditionally system administrators administer access control settings. To enable efficiency, the Model requires a System which only required their intervention when absolutely necessary. This meant that ways for getting staff with no access control knowledge to trigger access control procedures had to be found. It is clear that the System needs to be styled around how general staff members perform their work and how they interact with their organisation’s IT system.

The research therefore initially focussed on developing the Model so that the System automatically captured all the required access control triggering information from the standard procedural inputs of general organisational staff.

#### 2.1.4 A Research Voyage

The research involved a preliminary literature analysis and consideration of past work. Neither of these provided any obvious direction for the research to take. It was therefore necessary to set out somewhat blindly and to look for clues to a solution along the way. This can be likened to the ancient navigators setting sail in a general direction to discover new lands that could provide useful resources. In this case the “land masses” represent existing Research Fields (and Methodologies) and the “resources” represent Design Requirements.



**Figure 9: Research Fields and Methodologies**

Studies of the Research Fields often uncovered Design Requirements that could not be satisfied by techniques in that field. The need to find satisfactory techniques prompted the search to move to new fields. For example, studies in the field of Role Based Access Control (RBAC) uncovered the requirement to “Model Constraints”. As no suitable solution existed in RBAC research, the search progressed until satisfactory techniques were found in the Business Process Management (BPM) field.

## 2.2 Research Methodology

The Basic Research Methodology was to:

1. Examine an Existing Research Field or Methodology,

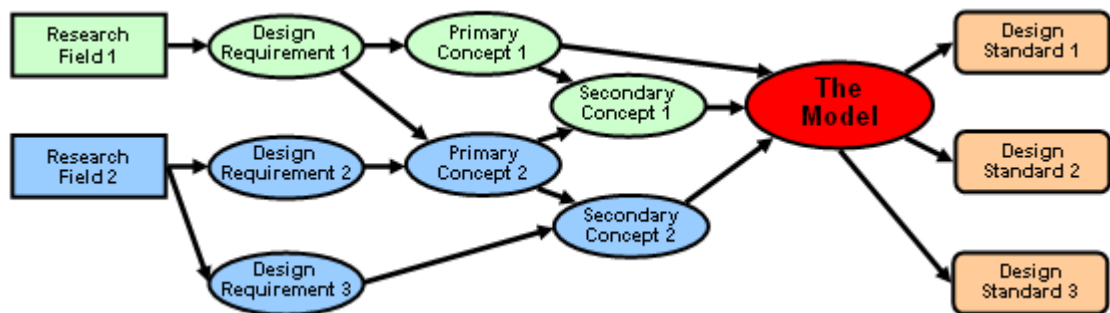
2. Extract Design Requirements from the existing knowledge,
3. Find a Model Concept that met the Design Requirement,
4. Develop the Model by incorporating the Model Concepts, and
5. Validate the Model against Design Standards.

The foregoing is shown graphically in Figure 10.



**Figure 10: Basic Research Methodology**

This Methodology is a linear view of the five steps involved in the research process. In practice, though, an additional dimension surfaced. This was due to the fact that Research Fields normally yielded more than one Design Requirement and that Design Requirements in turn could lead to more than one Model Concept. The complexity is increased by multiple levels of Model Concepts. Figure 11 shows Primary and a Secondary Concept levels, although in reality additional levels existed.



**Figure 11: Research Methodology in Practice**

The diagram shows that Model Concepts were derived in a number of ways. Primary Concept 1 is simply derived from a single Design Requirement whereas Primary Concept 2 is derived from multiple Design Requirements. Secondary Concept 1 is derived from multiple lower level Model Concepts while Secondary Concept 2 is derived from both a Design Requirement and a lower level Model Concept.

Figure 11 serves to highlight the multi-dimensional nature of the research process. The full complexity of the Component relationships is apparent from Figure 22. These relationships are detailed in Chapter 4.

While the research process was multi-dimensional, the five steps of the Basic Research Methodology (see Figure 10) remain useful in understanding and explaining the research methodology. The following subsections explain each of these steps in greater detail.

### **2.2.1 Examining Research Fields**

This Methodology chapter is located before the Literature Review chapter because the purpose of the Literature Review in the research is atypical. The reason for doing this is to introduce the strategy used in the Literature Review process and to show its relevance in the research process. The Literature Review process was not simply an examination of relevant research as is often the case; it also entailed a search for Design Requirements and Model Concepts. The relevance of this process needs to be understood before the Literature Review chapter is read.

The Literature Review chapter addresses each of the Research Fields outlined in Figure 9. The advantages and drawbacks of aspects in each field are discussed.

Research fields were initially examined when they showed potential for providing relevant insights, Design Requirements or Model Concepts. Many additional Research Fields to those shown in Figure 9 were examined in the actual process. These are not elaborated because they did not directly furnish any Design Requirements or Model Concepts. For example, the Sociology of Groups was examined. While the examination confirmed that Groups were a standard and practical way of representing collections of persons, it did not directly contribute any Design Requirements or Model Concepts.

### 2.2.2 Extracting Design Requirements

As well as discussing aspects in each Research Field in the Literature Review, the Design Requirements for the field are listed. The Design Requirements then appear in total in Table 1.

While several criteria were used in selecting relevant Design Requirements, the process was essentially qualitative rather than quantitative. The main criterion was whether the requirement represented best practice in the research field. Other criteria related to the significance, relevance and generality of the requirement.

The development of the Model was progressive; *Intermediate Model Concepts* were developed before the final Model was formed. On completion of the Model these became Model Concepts. The *Requirement Extraction Process* involving the extraction of Design Requirements was coincident to a degree with finding Intermediate Model Concepts.

In some cases the need for an Intermediate Model Concept was first determined and the requirements needed to develop the Concept were then found. For example, the need for Service Contracts was established, following which the requirements for them were found. This merely highlights the multi-dimensional complexity of the practical process.

**The fundamental research method could be described as “Concept Driven Requirements Extraction”.** The overall goal is to provide a collection of Model Concepts that facilitate all the Design Requirements.

### 2.2.3 Finding Model Concepts

The primary purpose of the Model Concepts was to provide the basis for the Design Requirements to be met by the Model. For example, the Concept of Service Based Purpose met the “Purpose Binding” requirement. While this one-to-one relationship between requirement and concept is not unusual, more complex relationships exist (see Figure 11).



The Model Concepts can be categorised as “Original”, “Existing” or “Derived”. An Original Model Concept is one that was conceived as part of the research. An Existing Model Concept is one that comes without alteration from a source within one of the Research Fields. A Derived Model Concept has been modified from an Existing concept in some original way.

A major output of this research is the collection of Model Concepts. Collectively they give insight into how to develop a compliant, efficient and functional business management system.

Many of the Original and the Derived Model Concepts can also be seen individually as results of the research. These Concepts have relevance in their own right.

#### **2.2.4 Developing the Model**

The Model defines the functional IT Components of the proposed business management system. Each of the functional Components is either directly or indirectly based on one or more Model Concepts. As a whole the functionality of the combined Components incorporates all the Model Concepts. The relationship between Model Concepts and Model Components is detailed in Chapter 5.

The basic aim in the Model Development was to create a model that efficiently and effectively dealt with all the Design Requirements by incorporating all the Model Concepts.

The “Model Development Methodology” was to:

1. Define the Model Components,
2. Define the Component Processes needed for each Model Concept,
3. Define the Component Data required by each Component Process, and
4. Analyse the Component Processes to ensure they are efficient and effective.

The first step in the Model Development Methodology was to define the Model Components. The first two Component Types – Groups and Services – took basic shape early in the research timeframe. As the initial Research Fields were studied, key Concepts that formed the bases of the Model Components were uncovered. The

Service Contract Component Type was added later as a means for incorporating Compliance.

The details of the Components were fleshed out as additional Concepts and required functionalities emerged. These additional Concepts and required functionalities were built into the Model as *Component Processes* and *Component Data*. This “fleshing out” process encompassed the second and third steps in the Model Development Methodology.

The last step in the Model Development Methodology involved the analysis of the Component Processes. This analysis was qualitative in nature and took the form of reviewing each process to ensure that no more efficient or effective method was available. The object here was chiefly to ensure that the human and automated processes could be performed quickly in practice, that is, with no obvious choke points unnecessarily impeding their completion.

### **2.2.5 The Validation Approach**

Many computer science research projects result in a software application that is used in some way to validate the research. The validation approach used is a “see, look, it works” one. Typically, while the application may work in the particular environment in which it is tested, there may be no proof that it works elsewhere. This approach is not useful when seeking to test a general solution because proving that it works in a given situation is of limited value.

There is also a problem in attempts to validate concepts that are not currently employed in practical systems. While the concepts of Groups and Services are understood by the average worker, current systems do not incorporate these concepts in the way that is envisaged here.

With these problems in mind, an alternative evaluation approach was chosen. This approach is best described as an “Analytical Evaluation”. Due to administrative complexity it is generally impracticable to quantify the efficiency of business processes. Within the scope of this research, with the large number of business processes covered, it would have been impossible. A qualitative approach was

therefore required. Analytical Evaluation is such an approach. Advice received from Professor Shirley Gregor (Gregor 2007) confirmed the Analytical Evaluation method as the best approach to use for this research.<sup>12</sup>

The Analytical Evaluation takes a *standard* and evaluates whether or not the model meets each requirement present in the standard. In this case three standards were used. The first is a Regulatory standard, the second, an IT Business Management standard, and the third, an IT industry standard. The standards reflect current requirements from different perspectives. Validation against the three standards is therefore a thorough test for the Model.

It is common practise to evaluate business and security systems against recognised industry standards. This is true of the US Department of Defence's Orange Book and the EU's Common Criteria for security and is the reason why the three standards used here were developed. In this case a model is being evaluated against the standards before the system is developed in order to ensure that systems based on the model are Compliant.

The "Analytical Evaluation Methodology" involved:

1. Requirement Extraction from three standards,
2. Requirement Integration to produce a Standard Requirements set,
3. Matching Standard Model Services to the Standard Requirements, and
4. Determining if all Standard Requirements were met.

The three Standards used in the Requirement Extraction stage were the OECD Privacy Principles (Organisation for Economic Co-operation and Development

---

<sup>12</sup> Shirley Gregor is the foundation Professor of Information Systems at the Australian National University, Canberra, where she is a Director of the National Centre for Information Systems Research. Among her many honours Professor Gregor was made an Officer of the Order of Australia in the Queen's Birthday Honour's list in June 2005 for services as an educator and researcher in the field of information systems and in the development of applications for electronic commerce in the agribusiness sector.

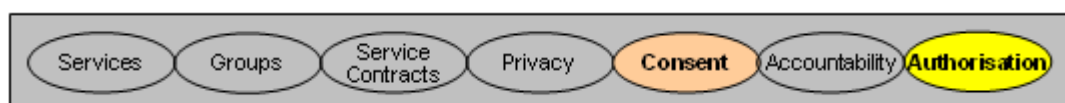
2006), the Control Objectives for Information and related Technology (COBIT) framework (ITGI (IT Governance Institute) 2007), and the United States Health Insurance Portability and Accountability Act (HIPAA) 1996. These Standards were chosen as the best examples of Regulatory, Business Management, and IT (industry based) standards respectively. A set of requirements was generated from each of these Standards, totalling around 330 in number.

In the Requirement Integration stage the three sets of requirements were integrated into one set of Standard Requirements. Many of the requirements were duplicates, as similar requirements existed in the different standards. Other requirements related to different aspects of a single requirement and by expressing the requirement in a more general form several similar requirements could be encapsulated. In a process that removed duplicates and aggregated similar requirements, the total number of requirements was reduced to 50. These 50 requirements formed the set of *Standard Requirements*.

The third and fourth stages involved functional comparisons between the Model and the Standard Requirements. The purpose of the comparisons was primarily to assess the level to which the Model achieves its goal of being a general purpose model. The third stage involved finding Model Services whose functionality matched that required by each of the Standard Requirements. In the fourth stage the matches were examined to determine whether or not they were conditional. If they were conditional, the conditions upon which the match depended were specified.

## 2.3 Linking the General Concepts

The nature of the research voyage meant the scope of the research broadened from its genesis. Its initial aim was to find a general access control solution that incorporated client consent. This essentially entailed establishing appropriate authorisation and consent mechanisms.



**Figure 12: Initial General Concepts**

The following subsections supply an overview of how the remaining General Concepts were incorporated. They can be thought of as a brief description of the entire voyage.

### 2.3.1 Using Groups to Manage Authorisation and Consent

Research into Access Control and workplace considerations in the Health Domain indicated that the Group Component was a possible choice to enable a functional Access Control Mechanism in the proposed management System.



Figure 13: Utilising Groups

### 2.3.2 From Authorisation and Consent to Privacy

In searching for the above solution it became evident that the issue of privacy of client data was central. When most of the foundational access control works were done in the 1970s and 1980s it was considered that data belonged to an organisation and that the organisation should protect its data. While this remains relevant to a degree, the world has radically changed in that personal information is now stored *en masse* by organisations, which have a responsibility to use and protect that information correctly. In other words, the issue of client privacy is now central.

It was therefore clear that the issue of privacy needed to be incorporated. To gain an understanding of privacy issues there was a need to take a broad high level approach. It was necessary to understand how legal requirements and business practices dealt with the privacy issue.



Figure 14: Linking Privacy

### 2.3.3 Linking Accountability and Privacy

In the course of this study it was noticed that proper privacy practices required the same sort of accountability as the maintenance of business records such as accounting records. Both essentially required that it be known “who had done what”. It therefore seemed logical that accountability and privacy be incorporated into one system. This made sense because the fundamental aim was to automatically trigger access control settings in the normal course of work.

Hence, though the scope of the research was limited at the outset to the access control area of research, to solve the access control problem it proved necessary to broaden its scope and to look at all aspects that impinge on access control activities within an organisation.

The aim shifted from finding a general purpose access control solution to finding a general privacy and accountability solution. The goal moved from defining a technical model to defining a Business Management Model that was compliant, efficient and functional.



Figure 15: Linking Accountability

### 2.3.4 Using Services to Manage Business Processes

The need to find an authorisation mechanism that could be incorporated into normal work processes led to the Business Process Management (BPM) and Service Oriented Architecture (SOA) research methodologies. This prompted the decision to use the Service Component to describe organisational work.



Figure 16: Utilising Services

### 2.3.5 Unifying the General Concepts with Service Contracts

At this point a simple Authorisation mechanism that could incorporate Privacy, Consent and Accountability into Service Management was required. The Concept of using Service Contracts as both the focus of Authorisation decisions and the mechanism to unify Privacy, Consent and Accountability decisions was born. In effect, Service Contracts unify the other Component Concepts (Services and Groups) and the Compliance Concepts (Privacy, Consent, Accountability and Authorisation).



Figure 17: Unifying the General Concepts with Service Contracts

## 2.4 Research Philosophy

In this section the philosophies on which the research is based are explained. The philosophies can be seen as principles that have been applied in the development of the Model. In a sense, they are sub-goals of the research or general rules upon which the Model is founded. Each of the following subsections is devoted to one of these philosophies, in no particular order of importance.

### 2.4.1 Use Inductive Reasoning

Inductive reasoning, sometimes called inductive logic or just induction, is the process of reasoning in which the premises of an argument are believed to support the conclusion but do not entail it. In other words the premises support the conclusion but do not ensure its truth. Induction is a form of reasoning that makes generalizations based on individual instances (Popper & Miller 1983).

Induction is used to formulate laws or to ascribe properties or relations to types based on a number of observations or experiences (that is, one or more). For example, induction can be used to conclude that “all ice is cold” based on the observation that

“this ice is cold” or the experience that “all ice I have ever touched was cold” (Wikipedia 2008b).<sup>13</sup>

Induction is a commonly used technique. While there is argument as to whether it is scientific, it is a useful technique in many circumstances. The greater the number of observations that support the conclusion, the stronger the case is that the conclusion is true. Another example of inductive reasoning is the widely supported theory of global warming. Many scientific observations support the conclusion that global warming is occurring, yet no single observation conclusively proves the theory because each observation can be explained by other occurrences.

The design of the Model is based on the set of requirements extracted during the review process. Most of these were drawn from “observations” of previous research. The reasoning behind the formulation of these requirements is thus inductive. The strength or validity of each requirement is related to the number of observations upon which it is based. For example, the requirement to use Groups is based on a large number of observations whereas the requirement to facilitate Authorisation Timing (that is, *uno tempore* and *ex post* Authorisations) is only based on a small number of observations.

#### **2.4.2 Develop a Simple General Purpose Solution**

Perhaps the most basic principle applied is that the Model had to be general in nature; that is, it should be usable in a variety of different domains. While the main domain used throughout the research was the health domain, the Model was not designed for use solely in this domain. The health domain was chosen due to its very complex nature and large variety of accountability and security requirements on multiple levels. Developing a model in such a domain therefore lends itself to a simpler extension to cover other domains.

While the specific requirements of individual domains are largely covered in the general approach, it is natural that individual domains will have their unique requirements. It is hoped that the flexibility of the Model will allow these unique

---

<sup>13</sup> Wikipedia is referenced for explanatory purposes, not as an authoritative source.



requirements to be modelled even though the Model has not been designed specifically to meet them. To some extent the Model is a “one size fits all” model. And it is hoped that its flexibility allows that one size to stretch or shrink to meet the individual requirements of specific domains.

The scope of the research was large and it touched on many Research Fields. By necessity it was not possible to dig too deeply into any particular Research Field. Rather, it was a matter of uncovering the relevant Design Requirements in each Research Field. The goal was to find a simple solution to a complex problem.

It is the case that a level of standardisation is required for a generic solution that enables developments in associated research areas to be effectively integrated. **The Model aims to facilitate interoperability rather than specifically solving existing localised problems.** For example, it is designed to enable an accounting program to reference a government website to access the current tax rates and to enable a hospital personnel program to reference a medical registration site to access the current state of a practitioner’s personal qualifications.

Ultimately, the real value of the Model lies in its simplicity. It centres on people having a natural understanding of Groups, Services and Service Contracts. The fact that these concepts are easy to understand enables interoperability. Regulations can be drafted in terms of Groups, Services and Service Contracts. Businesses can be managed in terms of Groups, Services and Service Contracts. Access can be controlled in the course of managing Groups, Services and Service Contracts.

### **2.4.3 Use Best Practice from Existing Solutions**

The “one size fits all” philosophy works because the Model seeks to encapsulate what can be described as “best practice”. The nature of the Review Process was essentially to find best practice requirements. The reasoning is that if best practice is encapsulated, it can be applied in a variety of domains. This is similar to the principle that standard accounting practices, based on double entry bookkeeping, are applied to all organisations. The model therefore seeks to reflect best practice in regard to authorisation and privacy practices.

In order to achieve best practice in one research field it was sometimes necessary to consult other research fields for a solution. For example, by modelling the contractual nature of authorisations, legal practices set the best practice standard for the technical authorisation mechanism. In this way the Model links different research fields together.

This philosophy draws on existing solutions. It takes the best aspects of existing solutions and attempts to mould them into a standard solution that is general in its application.

#### **2.4.4 Achieve Efficiency through Standardization**

There are inherent strengths and weaknesses associated with the standardisation of anything. The strengths or benefits are in such things as economies of scale associated with commercialisation, ease of use because of familiarity, and in the ability to get diverse systems to interoperate. The weaknesses relate to such things as the need to change existing or localized procedures to conform to new standards, the need to retrain users, and the inability to exploit efficient localized systems.

The main advantage of a standardized solution is that it is inherently cheaper to acquire. The main disadvantage is that standardization requires efficient and effective localized change management. In practice, the cost advantages of standardization can be nullified by complex change management issues. It is therefore of paramount importance that the change management required to enable standardization provides real cost savings over non-standard solutions. A balance between standardization and change management is required.

#### **2.4.5 Incorporate Change Management**

To provide the necessary cost savings associated with standardization, it is necessary to take change management issues into account when setting the standards. In other words, change management needs to be incorporated into the standardization process.

How can change management be incorporated into the standardization process? To answer this question in the context of computer systems it is necessary to understand that not all business procedures are digitally based or even designed with computerisation in mind. Rather, many procedures are based at best on written business policy or professional and industry standards or at worst on some unwritten localised practice.

Computer systems by their mathematical nature require a rule-based approach. They require the conversion of “paper based policies” to “digital policies” which can be enforced by the computer system. If no policies (rules) exist they must be created, which is likely to be a daunting and time consuming task.

A general model, to be practically feasible, must be able to deal with paper-based policies. In other words, in situations where there is a paper-based policy but not a digital policy users need to be directed to the paper-based policy. Where no paper-based policy exists, the model must allow one to be developed over time and/or refer the user to someone who knows what to do. There must, accordingly, be a well defined process for changing to digital policies. This process must by its nature be modular in that individual policies will be dealt with at different times. The process that is required could be described as “evolutionary”.

The fundamental driver to inspire change in procedures must be associated with the desire to model best practice. This is because best practice can be seen as a worthy and understandable goal. The philosophy is therefore that the goal of generalization is to model best practice and the reason for standardization is, at least in part, to enable best practice to be achieved.

#### **2.4.6 Adopt User-friendly Components**

One of the key things that a model such as the CMS Model specifies is software Components. As examples, Object Oriented Design (OOD) uses the “Object” component to model things in the real world; Business Process Management (BPM) uses the “task” component to model the jobs that workers perform; and Role Based Access Control (RBAC) use the “role” component to specify access privileges that can be applied to groups of users. It can be argued that the success of these models

lies largely in the comprehensibility of the component and in the simplicity with which the component models reality. In other words, designers and users can relate to the ideas of objects, tasks and roles.

Perhaps the most important philosophy behind the Model is that the Components must be simple and comprehensible to users who are not computer literate. If the Components meet this requirement, it should be possible for lay system users to make decisions requiring a basic understanding of the concepts involved. The consequence is that administrative tasks that would otherwise need to be performed by IT personnel can be performed by workers and organisational administrators. For example, if everyone understands the concept of a “group” of people, then system users ought to be capable of moving people in and out of groups.

#### **2.4.7 Automate Administration**

It was clear from the Technical Review that existing models lacked the capacity to be administered at low cost. To reduce the administrative burden associated with the model it seemed obvious that, as well as ordinary users taking part in administration, appropriate work related user decisions needed to be used to automatically trigger administrative procedures. For example, assigning a worker to look after a client should trigger the necessary access control procedure needed to enable the worker to access the appropriate client records.

In terms of business processes it was necessary to understand that some of the requirements involved manual processes (conducted by humans), some involved automated processes (conducted by the computer system), and others required a combination of manual and automated processes. It was therefore necessary to determine how both manual and automated processes could be transformed into software processes in a way that enabled them to work seamlessly together. Another factor to consider in this regard was how manual processes could be automated in an evolutionary way (that is, over time).

## 2.5 Conclusion

The basic methodology is to base the proposed Model on a broad ranging set of Model Concepts. These Concepts are drawn from general principles evident in existing research and address the set of Design Requirements extracted in the Literature Review process, which forms the next Chapter.

The research reviewed a broad range of existing Research Fields and Methodologies in the search for concepts and requirements (principles). The Basic Research Methodology involved examining each Research Field/Methodology to extract Design Requirements and to find Model Concepts that met the Design Requirements. The Model was developed by incorporating the Model Concepts, and validated against Design Standards.

The primary aim of the search for concepts was to find the General Concepts for the Model. The initial General Concepts were Authorisation and Consent. Groups, Privacy, Accountability, Services and Service Contracts were added in turn to generate the set of seven General Concepts.

The aim was to use Inductive Reasoning to develop a Simple General Purpose Solution based on Best Practice from Existing Solutions. The Central Requirement of Efficiency would be achieved through Standardization, by incorporating Change Management principles, User-friendly Components and Automated Administration processes.

The first stage of the research was the Literature Review. The next Chapter details this.

## Chapter 3

# Literature Review

---

### 3.1 Introduction

This Literature Review serves two main purposes: it describes the research topics that relate to the research, and it facilitates the Design Requirement Extraction process.

The twelve main sections of this chapter are devoted to the twelve Research Fields shown in Figure 9 (see 2.1.4). Each Research Field is sub-divided into Research Topics, and each Research Topic is reviewed in up to four stages.<sup>14</sup> In the First Stage the topic and research are described. The “Useful (or Important) Aspects” (the pros) and the “Problem Aspects” (the cons) within the research topic may be described in the Second and Third Stages. In the Fourth Stage Design Requirements (DRs) arising from the Research Topic are noted. The DRs take the following form and are included immediately after the related discussion.

#### ***DR#0: Example DR***

In section 14 the Design Requirements are collated into a single table. The final section contains a Critical Analysis of the preceding sections.

It was not the purpose to look for every possible detailed requirement that existed in each Research Field. Rather the idea was to capture the essence of the functionalities that are generally required. As the Model is a general purpose model, the Design Requirements are expressed in general terms. They can be thought of as “general principles”.

---

<sup>14</sup> The first stage applies to all topics but the remaining three stages are optional.

The Design Requirements are expressed from a managerial point of view, rather than from a technical or regulatory point of view, because the proposed System is a “management system”. The focus is the users’ perspective, the users predominantly being Managers and Workers.

While several criteria were used in selecting relevant Design Requirements, the process was essentially qualitative rather than quantitative. The main criterion was whether the requirement represented best practice in the research field. Other criteria related to the significance, relevance and generality of the requirement.

## **3.2 Computer Security**

Computer Security is a sub-field of IT research. While it deals with infrastructure (hardware), software and communication security, the dissertation is primarily concerned with *Information Security*.

### **3.2.1 Information Security Issues**

Organisations store a plethora of information that relates to Business Management and clients. There are obvious needs and requirements for them to keep this information secure. *Information Security* is concerned with verbal, written as well as digital information, while *Computer Security* is concerned primarily with the digital information.

Organisations, depending on their size and nature, each have some form of written or unwritten *policy* relating to Information Security. One or more persons in an organisation will be charged with the responsibility of managing Information Security. This involves *protecting information* from a wide variety of both internal and external threats that can be due to either intentional or accidental actions.

#### ***DR#1: Information Protection***

A common method of providing protection is to perform a *Risk Analysis* (Bishop 2003, pp. 17-18, ; Pfleeger & Pleegeer 2003, p. 494) and then to devise a Security Plan based on the analysis. The aim of the Security Plan is to introduce measures

that *reduce security risks*. The cost of providing elevated levels of protection is generally high, so economics usually dictate that only cost effective solutions are employed. Costs can be categorized as *upfront* (for such things as systems, software and training) and *ongoing* (for such things as upgrades and administration).

### ***DR#2: Reduce Security Risks***

The Security Plan will prescribe the management requirements that are necessary to maintain security. Even in organisations lacking a written plan, some form of *Information Security Management* will be required.

### ***DR#3: Information Security Management***

#### **3.2.2 Data Security Qualities**

The research is concerned with data<sup>15</sup> security. Pfleeger (2000, p. 9) identified the three primary data security qualities as *confidentiality*, *integrity* and *availability*. *Confidentiality* relates to preventing unauthorised disclosure. For example, persons generally do not want their personal information revealed to strangers or used for marketing purposes. *Integrity* relates to preventing unauthorised modification. For example, banks do not want account balance information to be inappropriately changed. *Availability* relates to preventing denial of authorised access. For example, a Worker doing a task wants access to necessary information.

#### **3.2.3 Information Ownership and Control**

“Information ownership” of client information held by organisations is not a clearly defined or uniform concept as it depends both on statute law that varies between jurisdictions and case laws that affect various circumstances. As there are different ways to specify how information is *controlled*, who *owns* the information may be of little significance, unless specific laws make it so.

---

<sup>15</sup> *Data*, as an IT term, is often equated with the term *information*, though *information* normally relates to some entity and as such has a context.



There are some parallels to the idea of who *owns* a client's money deposited in a bank. While legally the bank clearly *owns* the money (it could be possible to dictate otherwise), often clients are given *control* over it, for example, by enabling clients to make deposits and withdrawals using Internet or Phone Banking. This is not always the case though; for instance, with Term Deposits the client usually forfeits *control* for the specified period.

So there are two issues, *ownership* and *control*. While the dissertation does support a position on *ownership* (one of "client ownership" of their own PID), it argues *control* is more important because IT systems can enable effective *control*. However, before discussing *control* the "effects of ownership" need consideration.

The effect of an organisation owning client information ("organisational ownership") infers some right to use that information for the organisation's benefit, with little or no regard for the client's wishes. Consider the example of an organisation selling the phone numbers of its clients to marketing companies for profit.

The effect of "client ownership" is that an organisation must ask permission to use client information or to pass it on. Consider the example of a hospital passing on information regarding a patient's condition to persons that may inquire.

Regardless of actual ownership, an organisation could treat client information as if it were owned by the client even if it did in fact own the information itself. In other words, while it controls the client's information (termed here as "organisational control"), it could ask the client's permission for using the information even if not required to do so. This could be considered as good business practice, where the wishes of the client are paramount.

#### ***DR#4: Client Based Permission***

Another factor that needs to be addressed is whether clients should be able to access information about them that is held by an organisation and have the right to modify incorrect information. Consider the case of bills sent to an incorrect mailing address.

In some jurisdictions clients do have these “client reviewal” rights. And, even if they do not, it could be argued that common sense dictates that errors should be correctable.

#### ***DR#5: Client Information Reviewal***

Online facilities that allow patients to manage their own digital medical records have been available, for example, Google Health (Google 2008). With Google Health and its alternatives, patients’ records are held in a central location where they can be accessed and added to by the patients or health professionals. Patients are able to access their record and potentially correct any errors themselves.

Alternatively, the records could be held by organisations or their agents (for example, “information banks”) and the patient could be given the ability to access and correct errors. Regardless of where the information resides, patients have *client control*.<sup>16</sup> Even when *client control* is in place, an organisation would have some recourse to reject changes where it could prove that they were illegitimate.

#### ***DR#6: Client Control***

### **3.2.4 Authentication and Authorisation**

Bishop (2003, p. 309) defines *Authentication* as ‘the binding of an identity to a subject’.

Typically, the *identity* is a person using the IT system (that is, a *user*) and the *subject* is their representation in the system. For example, a “person” (*identity*) has a “username” (*subject*). A person may have multiple subjects in one or more systems. As far as the IT system is concerned the *person* does not exist; only the *subject*. The *binding* of the identity to the subject is done when the *person* is required to provide some information (for example, a password) to the system to prove that they are a legitimate (*authorised*) *user*.

---

<sup>16</sup> The dissertation advocates Client Control of Personal Data (for example, a confirmed medical diagnosis) and Organisational Control of related Management Data (for example, email discussions containing professional opinions regarding the diagnosis or professional notes such as reminders relating to required future actions).

This research is not concerned with *authentication* except to say that authentication mechanisms are to be employed in the proposed IT System. The main reason for raising the issue is that *authentication* and *authorisation*, while related, can be confused. Authentication involves proving the legitimacy of a *subject*, while authorisation involves controlling what system resources are available to the *subject*.

### 3.2.5 Access Control Basics

Access Control is the primary sub-field of Computer Security dealt with in the dissertation. A basic background follows.

In Access Control terms, system resources are termed *objects*. The entities that access (use) the objects are termed *subjects*. The system controls access to *objects* by storing “Access Rules” that allocate *rights*<sup>17</sup> (such as, *read*, *write*, *modify* and *append*) to *subjects*. Access Rules can be expressed as “Basic Access Triples” of the form (*subject*, *right*, *object*). For example, the Rule (John, read, File A) would allow the identity with username “John” read-only access to “File A”.

Lampson in his seminal paper (1971) showed how Access Rules could be stored in an “Access Control Matrix”, in “Access Control Lists (ACLs)”<sup>18</sup> or in “Capability Lists (C-Lists)”<sup>19</sup> (see Figure 18). An Access Control Matrix is a table with *subjects* listed down the side, *objects* listed on the top and *rights* entered into appropriate table cells. ACLs are composed of a list of (*subject*, *right*) pairs and are associated/attached to each *object*. Conversely, C-Lists are composed of a list of (*right*, *object*) pairs and are associated/attached to each *subject*. Once the Access Rules are stored in the system, the computer will perform the appropriate access control checks whenever a *subject* seeks to access an *object*.

The methods used to enter and check the Access Rules are a key factor in determining the effectiveness of Access Control Models. Ideally the entry process, which usually involves manual inputs by system administrators, should involve

---

<sup>17</sup> *Rights* are also known as *privileges* and *permissions*.

<sup>18</sup> Access Control Lists store a list of Subject-Right pairs with each Object (see Figure 18).

<sup>19</sup> Capability lists store a list of Right-Object pairs with each Subject (see Figure 18).

minimal work and the checking process, which is automatically performed by the system, should be fast in terms of processing time.

### Basic Access Triple:

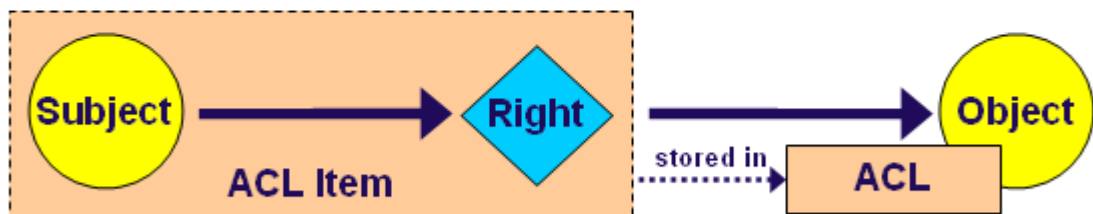


### Access Control Matrix:

	Objects			
Subjects				

Rights

### Access Control Lists (ACLs):



### Capability Lists (C-Lists):

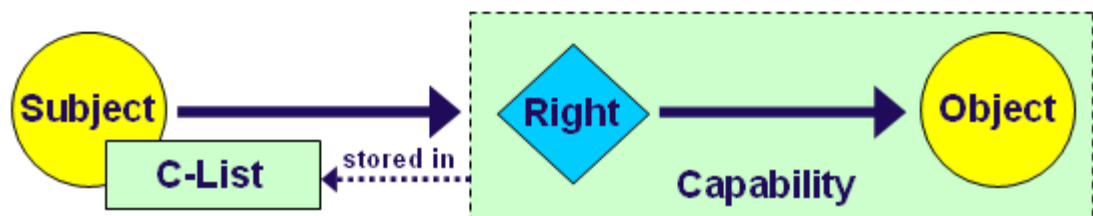


Figure 18: Access Control Matrix, ACLs and C-Lists

### 3.2.6 The Principle of Least Privilege

The Principle of Least Privilege (Pfleeger & Pleeger 2003, p. 265) (Gollmann 1999, p. 43) seeks to maintain the confidentiality of information. It essentially requires that accesses be allowed on a “need-to-know” basis. For this to be the case users must only be given the privileges that are necessary to access the information they need for the task they are performing. Beresnevichiene (2003, pp. 16-17) expresses this

idea in terms that "...permissions [rights] should not persist beyond the time that they are required for performance of a task".

#### **Useful Aspects:**

The Principle of Least Privilege can be used as a security goal for access control. It is one of the prime criteria for judging the worth of an access control model.

### **3.2.7 Granularity**

The concept of *granularity* (Pfleeger & Pleeger 2003, p. 183) is used to describe the extent to which an Access Control Model is able to model complex access control requirements. A system that allows a user access to a large object (such as a file) when they only require access to a part of the object (such as a single record in the file) is *coarse-grained*, while a system that only allows access to the required object (the single record) is *fine-grained*. While a **fine granularity provides better security**, it can be more difficult to implement.

#### **Useful Aspects:**

The ultimate goal is to make the granularity as fine as possible while minimising the amount of administration that is required.

### **3.2.8 Computer Security Conclusion**

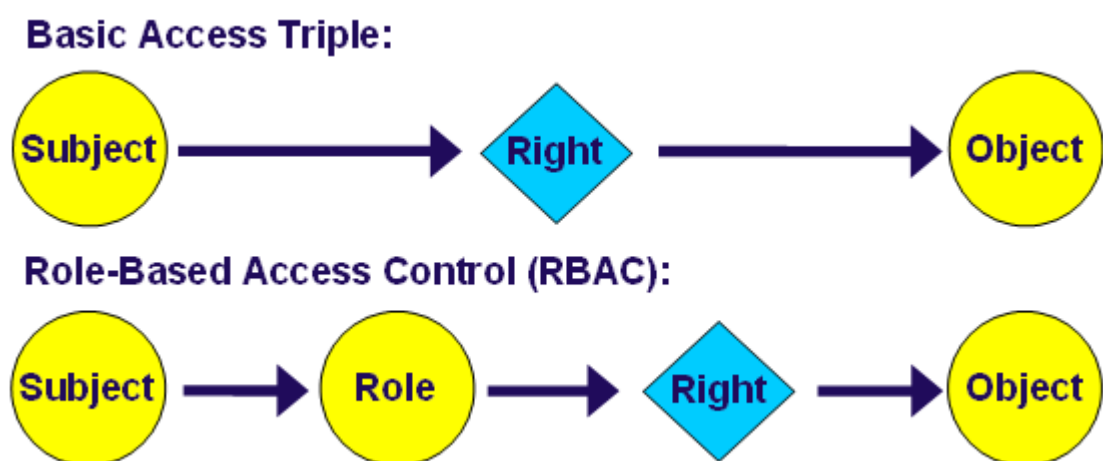
Six Design Requirements were extracted from this Research topic. The first three, regarding information protection, risk reduction and information security management are fairly typical and obvious requirements for computer systems. The last three relate to information ownership. Since the Model is client-centric, the principle of client ownership was deemed to be appropriate.

## **3.3 Role-Based Access Control**

Role-Based Access Control (RBAC) was introduced at 1.3.4.2.

RBAC, pioneered by Ferraiolo & Kuhn (1992), is an important and well established Access Control Model. RBAC is based on the classic Mandatory Access Control (MAC) model where the system controls access to resources based on classification levels assigned to both the objects (like “Confidential” and “Top Secret”) and the users (their security clearance level). Sandhu et al. (1996) introduced a family of reference models for RBAC which have been the basis for further developments. RBAC has evolved to a stage where it has been incorporated in many commercial systems, including Informix, Oracle and Sybase (Ramaswamy & Sandhu 1998). In February 2004, the National Institute of Standards and Technology (NIST) RBAC Model was adopted by the American National Standards Institute, International Committee for Information Technology Standards (ANSI/INCITS) as ANSI INCITS 359-2004 (NIST 2004).

By allowing administrators to group privileges together in role components (see Figure 19) and to allocate roles to subjects (users), the administrative burdens imposed by previous models is reduced. Instead of having to specify access triples for every subject-right-object interaction, right-object permissions could simply be given to a role. The role could in turn be applied to any number of subjects. This reduces the number of rules that are necessary. Also, the concept of assigning users to organisational roles is one that is intuitive and need not be performed by system administrators. This can further reduce the administrative burden.



**Figure 19: Role-Based Access Control**

While RBAC offers substantial administrative advantages over previous models, its use is still problematic in large organisations where there are thousands of roles and

millions of objects. To alleviate the administrative management burden in such cases a number of role management mechanisms have been proposed. Ferraiolo et al. (2003) and Ferraiolo (2003) describe a number of techniques including the Enterprise RBAC (ERBAC) model. Kern & Walhorn (2005) use rules to automate role allocation. This approach stems from the Rule Based RBAC (RB-RBAC) model proposed by Al-Kahtani & Sandhu (2002). Other models by Caelli & Rhodes (1999), Crook et al. (2002), and Fernandez (2005) also seek to make role management in large enterprises easier.

RBAC has spawned many other derivatives. Some of these are mentioned in later sections.

### **3.3.1 Group Concept**

Roles essentially provide a way of grouping subjects together. This idea is now almost universally applied in organisational computer systems. Most operating systems, including the current versions of both Windows and MacIntosh systems, utilise hierarchical group structures. Groups in such systems are very similar to, if not identical to, RBAC roles.

#### **Useful Aspects:**

The concept of roles and groups are now central to access control systems. The idea of placing users into groups is a task that is well understood by systems administrators. The existence of role/group hierarchies has also assisted in simplifying administration. Systems have evolved which routinely facilitate role/group management.

#### **Problem Aspects:**

While the possibilities of utilising roles on a large scale are attractive, applying RBAC to large organisations has proven difficult. Kern & Walhorn (2005) give some insights into role management in a number of large organisations which vary from having 150,000 users with 50 roles at one extreme to 11,500 users with 2800 roles at the other. They state that “the administration of roles in large organisations can become quite cumbersome and needs to be automated”. If roles are to be used

on a more global scale, the task becomes even more complex. Rhodes & Caelli (1999, p. 1) state that "...emerging networked systems, which have greater numbers of users, roles, and program components, challenge the expressive power of these classical RBAC models".

### ***DR#7: Hierarchical Groups***

#### **3.3.2 Context Concept**

Many RBAC based models have taken contextual conditions into account in order to model complex access control needs. The basic idea is that as well as using role-based controls, certain context variables are checked at runtime to determine whether accesses should be permitted. Neumann & Strembeck (2003) define a *conditional permission* as an RBAC permission which is constrained by one or more context constraints. Attributes representing context conditions are often stored for each user. For example, each user could be given an attribute which described their current location. Certain types of accesses could then be restricted to users in specified locations. Other researchers that use this approach include Bacon et al. (2002), Beresnevichiene (2003), and Georgiadis et al. (2001).

#### **Useful Aspects:**

The approach extends RBAC and preserves its advantages. It offers an additional means for the definition and enforcement of fine-grained context-dependent access control policies (Neumann & Strembeck 2003, p. 1).

#### **Problem Aspects:**

There is a fundamental problem with the approach because all users need to be given each attribute, say location, even if it is totally irrelevant to their access needs. To find which users have a particular attribute requires all users be searched. It seems likely, therefore, that there are more efficient ways to achieve this functionality.

Also, the attributes are checked at the time that an access is requested. If access is given at this point and an attribute changes so that access should be denied, there is no obvious mechanism to stop the access. The proponents of these systems refer to them as "dynamic". They may be more flexible than standard RBAC but they are



best described as “static” because changes in context do not trigger any access review mechanism.

In this dissertation a context check is a *static process* and the word *context* implies a one-time condition checking process. In contrast, 3.3.4 deals with *constraints*, which imply a *dynamic process*.

#### ***DR#8: Represent Contexts***

##### **3.3.3 Multiple Contexts**

Fernandez (2005, p. 2) points out that current methods using Access Control Lists (ACLs) and group based policies have had problems modelling multiple contexts. He states that “...static listings usually determine resource access by evaluating an object’s name or unique identifier. Groups determine resource access by evaluating a single object characteristic. However, resource access is usually based on the evaluation of multiple object characteristics contained in an object profile”.

Another aspect that requires modelling are “negative permissions” (Gollmann 1999, p. 38), where the onus is on *denying* a particular access instead of *allowing* all the other possible accesses. Allowing all possible accesses is much more complicated than making a general rule with limited exceptions (denials). Negative permissions are also necessary because there are circumstances where access needs to be specifically denied to a particular user, say, to a hospital worker who is an ex-partner of a patient.

In order for fine-grained control to be achieved in an access control system, the system must be able to take more than one context into account. For example, an access request may need to take a worker’s role, the current time and their location into account before granting access. It must be possible to form complex rules. The problem is that complex rules normally incur inordinate administrative overheads. Simple administrative techniques that have a degree of automation are required to ensure that overheads are acceptable.

#### ***DR#9: Multiple Contexts***

Two related issues are how to store the rules and how to process inter-domain access requests. These are addressed at 3.10.

### 3.3.4 Constraint Concept

Beresnevichiene (2003, p. 16) states that “[a]uthorisation constraints are an important aspect of access control and are a powerful mechanism for laying out higher level organisational policy. With the help of constraints the designers can lay out a broad scope of what is acceptable, and make it across administration domains”.

In RBAC security policies are expressed as constraints (or rules) on users and roles; separation of duties is a well-known constraint (Bertino et al. 1999, p. 1). RBAC based models use the concepts of *sessions* and *mutually exclusive roles* to deal with separation of duties.

#### Useful Aspects:

The need to be able to model constraints in access control has been accepted. Botha & Eloff (2001, p. 1) give a clue to the solution for modelling constraints when they point out that “[t]he workflow environment, being primarily concerned with the facilitation of complex work processes, provides a context in which the specification of separation of duty requirements can be studied”. One of the fundamental arguments presented in this dissertation is that Workflow Managements Systems should be used to enforce constraints. It was the study of this topic that led the research into the Business Managerial field.

#### Problem Aspects:

RBAC based models fundamentally don’t deal well with constraints. Beresnevichiene (2003, p. 16) states that “[u]nfortunately, the analysis and enforcement of constraints within role-based security systems has been only preliminary and tentative” Bertino et al. (1999, p. 1) concur in observing that “[u]nfortunately, current role-based access control models are not adequate to model such constraints”. El Kalam et al. (2003, p. 1) go further in arguing that “[n]one of the classical access control models such as DAC, MAC, RBAC, TBAC or TMAC is fully satisfactory to model security policies that are not restricted to static

permissions but also include contextual rules related to permissions, prohibitions, obligations and recommendations”. Pappas & Hailes studied traditional access control models and found that “[a]ccess control mechanisms currently employed in various applications lack the power to provide (express and enforce) complex, dynamic relationships between users and resources in a scalable and consistent way” (Pappas & Hailes 2003, p. 1).

This dissertation argues that the fundamental reason why these models do not deal with constraints is that constraints represent relationships between processes (or tasks/services). The relationship is between the states of different processes. In other words, there is dependence between processes. As the states of the processes change dynamically, static attributes and static components like sessions and mutually exclusive roles cannot represent these relationships in a simple way, or indeed at all. In contrast to constraints, contexts are independent in nature in that one context does not affect another. For example, a Worker’s role does not affect their location, though both may be relevant to determine if an Authorisation is to be given. The dissertation argues that the fundamental difference between “context” and “constraints” is that contexts model static independent conditions whereas constraints model dynamic dependent conditions.

#### ***DR#10: Represent Constraints***

The requirement to model constraints is evident in Access Control Research but there is no clear solution within the Research Field. Solutions are, however, evident in IT Business Management research (see 3.12.3).

#### **3.3.5 RBAC Conclusion**

While the study of RBAC (and its derivatives) revealed some key Design Requirements and other important issues, it essentially revealed that there were considerable problems relating to practical RBAC implementations. It was therefore deemed insufficient to base the Model on RBAC. As this is a substantial issue, a detailed argument as to why RBAC was not preferred is contained in the Results and Discussion chapter (Chapter 7).

While the Model does draw on the RBAC Design Requirements listed, it incorporates its own unique access control system. This system draws on the Business Management and Legal fields as well as other access control and IT research.

### **3.4 Set Theory**

#### **3.4.1 Set Based Access**

Set Theory is a foundational mathematical model and is well known and understood. It has very broad application, indeed Lehen (date unknown) states that “the goal of Set Theory was to provide a common axiomatic basis for all of mathematics”.

The thought here is to model groups using set theory. In set theory a set is essentially a collection of things of any kind. A *group* could also model a collection of things of any kind; for example, a group of users, a group of objects, a group of tasks, or potentially any other collection of things of the same kind. The benefit would be that set operations could be used to specify rules involving groups. For example, the Rules of Set Theory can be used to describe interactions between associated roles and associated object groups.

Set theory has been used as the basis for a number of computing mechanisms. For example, Dovier et al. (2000) incorporate set theory in their studies into handling constraints in logic programming. More specifically, there are a number of examples where set theory has proved useful in access control. Chen & Sandhu (1996) use a language based on set theory to specify constraints in a role-based model. Set theory is also the basis of the Role-based Trust Management Framework proposed by Li & Mitchell (2002, ; 2003). They define new operators to facilitate authorisations from multiple individuals in different roles.

#### **Useful Aspect:**

The benefit of basing groups on sets is that set operations can be performed on combinations of groups. For example, users can be grouped into sets and set operations can be used to allow privileges to be given to set unions, intersections and

relative complements. This is significant because fine-grained access control rules can be specified without the need to add additional groups to represent fine-grained roles. Group/role management is a much simpler task with because there are fewer groups to manage. For example, the intersection of “Doctors” and “Patient A Care Team” groups can provide what previously needed to be defined as a separate “Doctors for Patient A” group.

#### ***DR#11: Set Operations***

### **3.5 Clark-Wilson Model**

#### **3.5.1 Well Formed Transactions**

The Clark-Wilson model is one of the key methodologies upon which the Model is based. The significance of the issues surrounding well formed transactions must be understood because well formed transactions are a computer security concept that applies to business procedures. **The essence of the Clark-Wilson model is that persons do not access information directly, but access procedures<sup>20</sup> that in turn access the information.** This ensures that existing information is only accessed and modified in prescribed ways. For example, rather than a bank worker altering a client’s account balance directly, the worker uses deposit and withdrawal procedures that in turn change the balance.

In their seminal paper Clark & Wilson (1987) present a policy for data integrity based on commercial data processing practices. They compare the mechanisms needed for this policy with the mechanisms needed to enforce the lattice model for information security. They argue that a lattice model is not sufficient to characterize integrity policies, and that distinct mechanisms are needed to control disclosure and to provide integrity.

Clark & Wilson point out that while mandatory confidentiality protection (non-disclosure) mechanisms contain tight rules about writing data these rules are not

---

<sup>20</sup> Programs (that is, pieces of computer code) that perform prescribed tasks.

designed to protect integrity. The “non write-down” rules exist to stop classified data being copied to a non-classified state.

In contrast, they emphasise that while confidentiality is important in commercial data processing, the most important goal is usually “to ensure integrity of data to prevent fraud and errors”. The authors note that the two mechanisms at the heart of fraud and error control, *well-formed transactions* and *separation of duties*, were derived long before computer systems came into existence, and add that:

*The concept of the well-formed transaction is that a user should not manipulate data arbitrarily, but only in constrained ways that preserve or ensure the integrity of the data ... [and] separation of duty ... is ensured indirectly by separating all operations into several subparts and requiring that each subpart be executed by a different person. (Clark & Wilson 1987, pp. 186-187)*

*To ensure that data items are manipulated only by means of well-formed transactions, it is first necessary to ensure that a data item can be manipulated only by a specific set of programs ... [and] to ensure separation of duties, each user must be permitted to use only certain sets of programs. (Clark & Wilson 1987, p. 187)*

### 3.5.2 The Clark-Wilson Model

The Clark-Wilson model defines data items within the system to which the integrity model must be applied as *Constrained Data Items (CDIs)*. It then defines two classes of procedures: *Transformation Procedures (TPs)* and *Integrity Verification Procedures (IVPs)*. The TP corresponds to the concept of the well-formed transaction. The purpose of the TPs is to change the set of CDIs from one valid state to another. The purpose of an IVP is to confirm that all of the CDIs in the system conform to the integrity specification at the time the IVP is executed.

The model assures integrity with a two-part process: *certification*, which is done by the security officer, system owner, and system custodian with respect to an integrity

policy; and *enforcement*, which is done by the system. The model is summarised by five *certification rules C1 – C5* and four *enforcement rules E1 – E4*:

- C1: All IVPs must properly ensure that all CDIs are in a valid state.
- C2: All TPs must be certified to be valid. For each TP, and each set of CDIs that it may manipulate, the *security officer* must specify a “relation”.
- C3: The list of relations in E2 must be certified to meet the separation of duty requirement.
- C4: All TPs must be certified to write to an append-only CDI (the log) all information necessary to permit the nature of the operation to be reconstructed.
- C5: Any TP that takes an Unconstrained Data Item (UDI) as an input value should take the input from a UDI to a CDI.
- E1: The system must maintain the list of relations specified in rule C2, and must ensure that the only manipulation of any CDI is by a TP.
- E2: The system must maintain a list of relations which relates a user, a TP, and the data objects that TP may reference on behalf of that user.
- E3: The system must authenticate the identity of each user attempting to execute a TP.
- E4: Only the agent permitted to certify entities may change the list of such entities. An agent that can certify an entity may not have any execute rights with respect to that entity.

#### **Useful Aspect:**

The aspects of the Clark-Wilson model that specify the requirements for TPs are of particular interest. These are covered in the certification and enforcement rules. By ensuring that programs conform to these rules integrity can be assured.

***DR#12: TP User Authentication***

***DR#13: Certified User List Management***

***DR#14: Certified TPs***

***DR#15: User-TP Relation Specification***

***DR#16: TP-Data Relation Specification***

***DR#17: Separation of Duties*****Problem Aspect:**

Clark & Wilsons' work indicates that access by users to programs should be controlled. The concept of a program has changed over the decades since their model was formulated. Programs now are often very complex and monolithic in nature. They usually perform myriad functions. For their idea to work and for the Principle of Least Privilege to be adhered to, smaller components than programs must normally be used.

**3.6 Health Informatics**

Health Informatics deals with the storage and retrieval of health information. This section focuses on Design Requirements related to practices in the Health Domain, particularly in hospitals.

**3.6.1 Cooperative Practices**

“Cooperative Practices” have to do with work situations where responsibilities or tasks are shared in some way. “Collaboration” is another term which refers to the same type of thing.

Collaboration came to the attention of researchers in the field of Access Control in the late 90s. Access control models before this were not able to adequately deal with the issue. Cohen et al. (2002, p. 98) sum this up in their observation that “...attempts to apply RBAC in collaborative environments revealed some of RBAC's limitations. In particular, RBAC does not provide a component that captures a set of collaborating users, operating in, potentially different roles”.

A solution to the problem was found in using a component for “teams” (see also 3.7.2). Thomas (1997) introduced the notion of Team-based Access Control (TMAC) in 1997. TMAC is a significant MAC based model that attempts to deal with volatile environments. He described how the advantages of RBAC could be coupled with the concept of restricting access only to relevant clients. The idea of



integrating teams into RBAC was also pursued by Wang (1999), where it was applied to cooperative hypermedia environments.

Work on teams has been applied in the health domain. Georgiadis et al (2001) developed a model called Context-based Team Based Access Control (C-TMAC) and programmed a “view-based active access-control system”, using the model, for use in healthcare environments (Georgiadis et al. 2002). The most recent incarnation of TMAC, TMAC 2004, by Alotaiby & Chen (2004) uses organisational roles rather than specific team roles.

#### ***DR#18: Cooperative Practices***

### **3.6.2 Multiple Authorisations**

In a number of cases policy and/or regulation require that multiple persons authorise a particular task. Often this will be the person performing the task and one or more others. For example, before a nurse administers a Schedule 8 Medication (that is a “Dangerous Drug”) to a patient, the nurse and a witness must sign a drug register. If an IT system is used for such registrations then it must facilitate the input of multiple authorisations for a single task.

#### ***DR#19: Multiple Authorisations***

### **3.6.3 Guaranteed Access**

In clinical situations, such as in hospitals, emergencies arise where immediate access to patient records can prove the difference between life and death. Health professionals therefore demand that access to records be guaranteed. Systems must be able to meet this demand.

#### ***DR#20: Guaranteed Access***

### **3.7 Professional Access Control**

As mentioned at 2.1.2, the Professional Access Control (PAC) model (de la Motte 2004, ; de la Motte & Hartnett 2005b) was developed as a prelude to this research. Here it is examined for the purpose of extracting Design Requirements for the Model in this Dissertation.

#### **3.7.1 Professional Management**

PAC was developed for the professional hospital environment. In this environment health professionals routinely take responsibility for the privacy of medical records and patient consent. In fact, with paper-based systems *all* the responsibility is placed on the health professionals. In other words, the professionals managed privacy and consent in paper-based systems. Due to the inconvenience in retrieving paper records and the presence of administrators who looked after them, there was limited opportunity or incentive for inappropriate action to be taken by the health professionals.

With the advent of IT systems in hospitals and digital health records a problem arose because in IT systems the IT system administrators traditionally control access to records. Health professionals were now required to obtain the appropriate permissions from the IT personnel. Now as the culture dictated that the health professionals should not be impeded in doing their work, access control rules tended to be relaxed. For example, all health professionals in a hospital could be given access to all patient records.

So with digital health records any health professional can potentially get access to any number of records instantaneously and without any scrutiny. This situation poses many security concerns including the stealing and divulging of private patient information. There are concerns, for example, that health insurers could gain access to digital medical records and use the information to refuse cover to particular individuals.

The main aim of PAC is to restrict access to hospital records on a “need-to-know” basis and give the health professionals the ability to manage access control themselves without the need for involving IT personnel (except in an auditing role). By providing simple access control mechanisms that could be employed at the coalface, PAC provides both increased security and reduced administration costs.

***DR#21: Need-to-Know Access***

***DR#22: Reduce Administration***

### **3.7.2 Service Teams**

In order to facilitate Need-to-Know access, PAC employs an individual “Patient Care Team” for each patient. Team Members are allowed to access parts of the Patient’s record according to their organisational role, for example, an Administrator on the team can access the Patient’s administrative records. Other persons in the hospital are allowed “Restricted Access” to the Patient’s record depending on the circumstances and their relationship to the Team.

**Useful Aspects:**

The concept of Need-to-Know access is appropriate for general use in Organisations. The concept of a Patient Care Team in a hospital can also be generalised to “Client Service Teams” in any Organisation.

***DR#23: Service Teams***

### **3.7.3 Restricted Access**

For persons not in the Patient Care Team, PAC provides three types of Restricted Access through “Restricted Authorisation” mechanisms. *Associate Authorisation* allows a colleague of a Team member with the same role to assist the Team Member with their work. *Emergency Authorisation* allows access to any patients’ record but limits access according to the Worker’s role. *Critical Authorisation* allows access to any part of any record.

***DR#24: Associate Authorisation***

***DR#25: Emergency Authorisation***

**DR#26: Critical Authorisation**

These Restricted Authorisation types all provide “provisional access” where the authorisation must be later confirmed. The confirmations are provided through mechanisms that employ *Authorisation Timing* and *Authorisation Ordering* techniques.<sup>21</sup> These techniques are described below.

**3.7.4 Authorisation Timing**

If a Worker performs a service at a particular time, Authorisation Timing has to do with *when*, relative to this time, the Worker received the required authorisation.

The reason that Authorisation Timing is an issue is that in order to fulfil the Principle of Least Privilege an authorisation should only persist for as long as is necessary to perform the required work. Many systems grant access for indefinite periods simply to cut down the amount of administration required. If authorisations are made more restrictive, mechanisms must exist to facilitate occasions, like emergencies, where an authorisation is not already in place.

As Bretan (2004) points out, most access control and authorisation models follow the principle that “[a] user makes an access request of a system in some context, and the system either authorises the access request or denies it”. However, there are other possibilities. Stevens and Wulf (2002) categorised authorisations as *ex ante* when given prior to access, *uno tempore* when given at the time access is required, and *ex post* when given retrospectively.<sup>22</sup>

A number of systems have been proposed which deal with *ex post* authorisations. In fact, it is common to use the processes of logging and auditing as a mechanism to check whether accesses are properly authorised. However, logging and auditing can be seen as unsatisfactory because the procedures involved are not necessarily well defined, extensive or reliable.

---

<sup>21</sup> *Authorisation Timing* and *Authorisation Ordering* are terms defined in the dissertation.

<sup>22</sup> The terms *ex ante*, *uno tempore* and *ex post* are use in the dissertation to denote *Authorisation Times*.

Optimistic Security (Povey 1999) introduced the idea that all accesses can initially be allowed. It allows for integrity to be maintained by providing mechanisms for rolling back data to previous states. Actions can be taken against users who abuse their access rights.

The concept of *provisional authorisations* (Kudo 2002, p. 1) (Bretan 2004) tells the user that his request will be authorised provided he (and/or the system) takes certain security actions such as signing a statement prior to authorisation of his request. If the actions are mandatory then the mechanism can be used for *uno tempore* (or *ex post*) authorisations. If they are discretionary, they can be used only for *ex post* authorisations.

Rissanen et al. (2004, p. 1) proposed a system which routinely allows access, under specified constraints, even when it is not explicitly permitted. Their system depends on audit and sanctioning for enforcement. They maintain that this mirrors procedures in “manual organisations” and that established organisation theory recognizes such dependency on “rule bending” (where rules are treated as discretionary guidelines rather than mandatory specifications).

#### **Useful Aspect:**

The *ex ante*, *uno tempore* and *ex post* concept and terminology is very useful. *Ex ante authorisations* are the norm and are given before an access is requested. *Uno tempore authorisations* are sought when an access is requested and must be received before the access proceeds. *Ex post authorisations* are retrospective in that the access is made and then its legitimacy is confirmed (or denied) through an auditing process.

The concept of provisional authorisations being given subject to specified security actions being performed is also useful and can be applied in the mechanisms that deal with *uno tempore* authorisations.

#### ***DR#27: Authorisation Timing***

### 3.7.5 Authorisation Ordering

An access or a task may be authorised by a number of different persons. The concept of “Authorisation Order” recognizes that the authorisations given by one authoriser can override those of another authoriser. It also recognizes that there is logical order of who to seek an authorisation from. Rules can dictate the priority for choosing between multiple authorisers. For example, round robin or first-come-first-served methods could be used.

Various methods of seeking an authorisation can also exist, say, verbal, written or email methods may be used.

Rissanen et al. (2004, ; 2005) introduced the idea of “Discretionary Overriding of Access Control”, where *ex post* authorisations are available in certain situations when *ex ante* authorisations are not. They also propose that procedures for obtaining authorisations be part of the system. This is in contrast to most Access Control models, which specify *what* authorisations a user requires but do not provide the user with a mechanism (other than to contact a system administrator) to obtain the required authorisation. The procedures for obtaining authorisations essentially automatically direct an Authorisation Request to an appropriate authoriser(s).

#### ***DR#28: Authorisation Order***

## 3.8 Purpose Based Access Control

### 3.8.1 Purpose Concept

“Purpose Based Access Control”<sup>23</sup> methodologies are based on Consent. Consent is required when data is collected. For Consent to be valid it must be Informed Consent. Informed consent requires that the client be informed of the *purpose* for which the data will be used.

---

<sup>23</sup> “Purpose Based Access Control” is a term used in the dissertation to refer to Access Control methodologies that use *purpose* to limit access.

“Purpose Binding” refers to the concept of providing metadata or related data which shows the purpose for which data is to be used. It has taken the form of attaching labels to data (Fischer-Hubner & Ott 1998, ; Byun et al. 2005) to indicate for what it can be used. While the labelling approach is very complex and has the problem of defining and matching appropriate labels, the research has established the case that purpose binding is necessary to facilitate privacy compliance. For example, Fischer-Hubner & Ott state:

*“In order to enforce privacy, it should be checked whether the purpose of the task, currently performed by the user who wants to access personal data, corresponds to the purpose for which that personal data were obtained (requirement of purpose binding).”*

Byun et al. detail related work which deals with purpose binding and privacy protection. They point to the proposed IBM Enterprise Privacy Authorisation Language (EPAL) (IBM) for writing enterprise privacy policies to govern data handling practices in IT systems, Agrawal et al.’s (2002) concept of Hippocratic databases which incorporates privacy protection within relational database systems, and Lefevre et al.’s (2004) approach to enforcing privacy policy in database environments.

#### **Important Aspects:**

The research leads to the conclusion that for a system to be privacy compliant it must utilise appropriate purpose binding and consent management mechanisms.

#### ***DR#29: Purpose Binding***

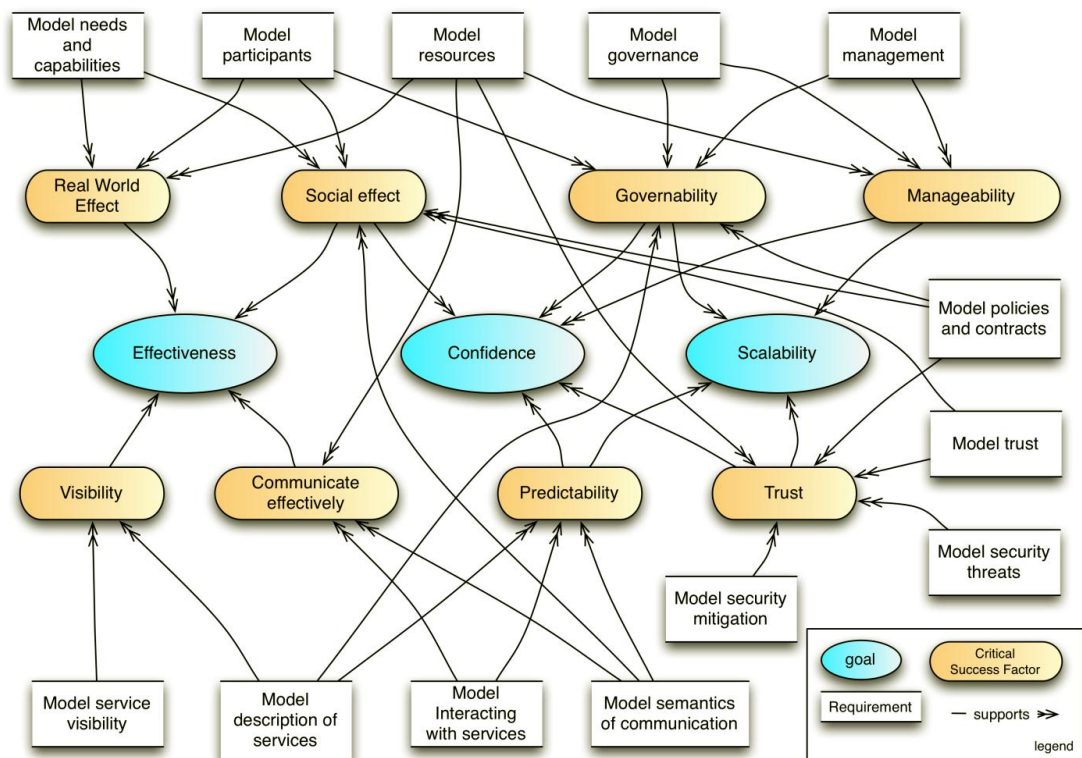
### **3.9 Service Oriented Architecture**

SOA was introduced at 1.3.4.1.

The SOA methodology has been progressively developed over the past two decades. It essentially seeks to model organisations in a modular way in order to facilitate the design of efficient IT-based business management systems.

The US-based Organization for the Advancement of Structured Information Standards (OASIS) is a not-for-profit consortium that provides global information system standards that had its beginnings in 1993. One of the many standards it has promoted is a Reference Model for SOA (OASIS 2006). There are several other groups, most notably the Open Group, that proffer similar standards.

Figure 20 shows a Critical Factor Analysis diagram from the OASIS Reference Architecture for Service Oriented Architecture document (OASIS 2009). The purpose of including this diagram is to highlight the complexity of the issues that the OASIS model seeks to incorporate.



**Figure 20: Critical Factors Analysis of the OASIS SOA Reference Architecture**

### Useful Aspect:

SOA research and development shows that using the concept of Services can lead to commercially viable business solutions. For example, SAP is the largest provider of IT business software in the world. SAP software is based on SOA. Services are the central concept in SOA.



***DR#30: Represent Services*****Problem Aspect:**

SAP provides IT solutions for large enterprises, including mining companies such as BHP, banks such as the Commonwealth Bank, and airlines such as Lufthansa. SOA software to date has not proved cost-effective or popular for SMEs. Arguably, this could be attributed to the fact that SOA standards and offerings to date have been overly complex and not suitable for use by untrained non-IT personnel.

**3.10 Cloud Computing**

Cloud Computing was introduced at 1.3.4.4.

**3.10.1 Inter-Domain Access**

Inter-domain access allows for information to be accessed from outside a domain. Networks in many organisations are now accessed from outside the organisation.

It is common for many copies of information to exist in diverse places. When the original information is updated at the source, the copies do not reflect these changes. A system that allows inter-domain access can mean that the copies can be automatically updated. Alternatively, the source can be accessed when the information is needed meaning that copying is not required.

There can also be a need to perform tasks remotely. For example, it is common for doctors to authorise procedures remotely. The concept of Web Services also requires inter-domain access. Services in one domain are provided for users in other domains. Koshutanski & Massacci (2003, p. 1) state that the issue has to do with business processes that cross organisational boundaries and list the following differences with traditional aspects of access control architectures:

1. credential vs classical user-based access control,
2. interactive and partner-based vs one-server-gathers-all requests of credentials from clients,

3. controlled disclosure of information vs all-or-nothing access control decisions,
4. abducting missing credentials for fulfilling requests vs deducing entailment of valid requests from credentials in formal models, and
5. “source-code” authorisation processes vs data describing policies for communicating policies or for orchestrating the work of authorisation servers.

### 3.10.2 Inter-Domain Roles

To facilitate inter-domain access it is necessary that credentials in one domain are recognised in another domain. Role-Based Trust Management (“RT”) (Li & Mitchell 2002, ; Li & Mitchell 2003) is an example where this occurs. The term “Trust Management” relates to the fundamental idea of a “trusted system” (Pfleeger & Pleeger 2003, p. 229),<sup>24</sup> specifically as to whether or not (or to what degree) an outside party is “trusted” and how to represent the level of trust in the system. Essentially, in RT roles in one domain can facilitate access in another domain where those roles are recognised. Other methods are described in Attribute Based Access Control (ABAC) (Wang et al. 2004) and Coalition Based Access Control (CBAC) (Cohen et al. 2002).

In dealing with inter-domain access the concept of intrusion detection needs to be considered (Pfleeger 2000, pp. 468-473). This is because it is possible that intrusion detection systems may hamper legitimate inter-domain accesses and inter-domain authorisations.

#### Useful Aspects:

Work in the area of Trust Management (Li & Mitchell 2003) has shown that the possession of an organisation role can be used to facilitate authorisations for accesses to systems of other organisations that recognise the role. Roles therefore can be used to enable remote authorisations, potentially up to a global scale.

#### ***DR#31: Inter-Domain Roles***

---

<sup>24</sup> The concept of “trust” is a fundamental Access Control concept that relates to how *trustworthy* a system is. It can be thought of as an indication as to how *secure* the system is.

Two other interesting issues, trust negotiation and inter-domain privacy, were raised by Lee et al. (2006) and Cautis (2007) respectively. Lee et al. produced a service called Traust which is a “Trust Negotiation-Based Authorisation Service for Open Systems”. Cautis considered the privacy aspects of inter-domain access. Both of these methodologies point to the need for a form of global protocol for inter-domain access that is more substantial than in peer-to-peer relationships where two organisations enter into bilateral access agreements.

### ***DR#32: Global Access Protocols***

#### **Problem Aspects:**

RT, the system proposed by Li & Mitchell (2003), combines RBAC and trust-management (TM) systems. It uses RBAC role, session and role activation concepts and TM methods for managing distributed authority through credentials and notation for denoting relationships between parties. While the TM methods may be useful, the RBAC session and role activation concepts limit the modelling of constraints (see 3.3.4).

RBAC models also struggle with the complexities inherent in large networks. Rhodes & Caelli (1999, p. 1) concur, saying:

*Basic RBAC models have been successfully applied since the mainframe era, but emerging networked systems, which have greater numbers of users, roles, and program components, challenge the expressive power of these classical RBAC models. This is particularly true for cross-enterprise distributed networks for electronic commerce applications. The development of new modelling concepts and techniques is required to support large-scale, enterprise-wide, distributed systems.*

#### **3.10.3 Inter-Domain Rule Processing**

Two further issues are how to store access rules and process access requests. This is complicated when inter-domain access is required. The questions concern which domain's rules are used and how the rules take other domains into account.

A number of methodologies target these issues. Rule-Based RBAC approaches (Kern & Walhorn 2005, ; Al-Kahtani & Sandhu 2002, ; Desmond 2003) deal with rule specification in RBAC systems. Organisation Based Access Control (ORBAC) (El Kalam et al. 2003) uses a context-based approach for rule specification in organisations. Attribute Based Access Control (Wang et al. 2004) targets the inter-domain problem.

Rules need to be stored and processed in a way that maximises both security and processing speed. These two needs can be conflicting as adding security features, such as encryption,<sup>25</sup> increases the amount of processing required.

### ***DR#33: Efficient Rule Processing***

## **3.11 Generic Programming**

The advantages of Generic Programming were outlined at 1.3.4.3

### **3.11.1 The Generic Nature of Services and Groups**

Services designed for one organisation may be reusable in other organisations. In fact, there are sets of services common to all organisations, say basic accounting services. Other sets of services are relevant to specific industry sectors (for example, health) and for particular business types (for example, florists). At the lowest level there will be sets relevant to particular organisations (for example, McDonalds or a government department).

The idea of reuse of services is clearly evident in both the SOA and Web Service methodologies. In both cases Services provided by one organisation can be used by other organisations, persons and even other Services (that is, computer programs) that are outside the organisation. These methodologies both employ systems that facilitate interoperability between remote systems.

As with Services, Group Structures designed for one organisation may also be reusable in other organisations. Again, there will be sets of groups, particularly those

---

<sup>25</sup> Encryption is used to turn plain text messages into code so unauthorised users cannot read them.

representing roles, common to all organisations; to business types, and to industry sectors.

In 3.10.2 it was evident that roles (that is, role groups) in an organisation could be recognised outside the organisation. There is no reason why this concept cannot be extrapolated to include any type of group. Recognising Groups on a global level would enable systems to interact on a global scale.

#### ***DR#34: System Interoperability***

### **3.11.2 User Perceptions of Service and Group Concepts**

This subsection deals with how users perceive the concepts of Services and Groups. User perception is important because the Model seeks to gain efficiencies by enabling untrained or minimally trained users (Clients, Workers and Managers) to perform actions whose current equivalents require IT expertise.

In relation to Clients, most internet users are familiar with the idea of finding goods and services online; this is, after all, one of the fundamental reasons for using the internet. Persons are also familiar with the idea that organisations provide services to consumers, both online and in traditional ways.

Similarly, most Workers and Managers would find it easy to understand that their work involved providing Services to Clients, their Organisation or fellow Workers.

Consequently, it is not difficult to envisage that the idea that both the offering of a Service to clients and the provision of the Service work can be related to the central idea of a “Unit of Service”. In other words, **persons can relate to the idea that a Service can represent a unit of organisational work.**

Sociology is the prime research area which looks at *groups*. In fact, sociology has been defined as “the study of social groups, and how social groups influence – and are influenced by – the people who live within them” (Gallaudet University Dept of Sociology 2008). Social psychologists are also interested in the study of groups.

Within sociology, “a *group* is usually defined as a collection of humans who share certain characteristics, interact with one another, accept expectations and obligations

as members of the group, and share a common identity” (Wikipedia 2008a). *Social groups* are defined as having two or more people interact and identify with one another (Bussiere 1999). Hogg (2005) speaks in terms of social categorisation in stating that “[a] core premise is that human groups are social categories ... [and] ... Social categorization transforms perception ... to the prescriptions of a contextually relevant ingroup or outgroup prototype”.

Consequently, as the understanding of groups is fairly universal, it is not difficult to envisage that users would understand the concept of needing to belong to a Group in order to receive Services provided to the Group. In other words, **persons can relate to the idea of Group membership**.

With widespread understanding of both Service and Group concepts by persons, it is not a large step to **enable persons to communicate with IT systems in terms of Services and Groups**, as long as the Service and Group definitions are represented in a user-friendly manner.

#### ***DR#35: User-Friendly Groups & Services***

##### **3.11.3 Generic Services and Groups**

Most persons are familiar with generic brands, such as those offered by supermarket chains. The major incentive for consumers to purchase these brands is that they are usually among the most affordable. While it may not be as obvious, a generic Service-based IT System is essentially an affordable “off-the-shelf” business management system that can potentially be used by any organisation.

While well-established large enterprises (organisations) may have developed their own management systems or even employed one of the large expensive generic solutions available (for example, SAP); the same is not usually true for SMEs. There are two key differences in this regard between large enterprises and SMEs. The first is that most SMEs are relatively new and often do not have established systems in place. The second is that they tend to lack expertise in IT and other management areas. A generic Service-based approach like that envisaged in the Model proposed can automate IT and management tasks. This is likely to be of benefit to many

SMEs. In particular, new SMEs with no existing systems would likely be prime candidates for such systems as the process for setting up a new business management system could itself be automated.

A generic Service-based System is affordable because its components are designed to be reused in manifold circumstances. The needs of individual organisations are met by providing an appropriate set of standard modular software components that are easily configured for the organisation's particular needs.

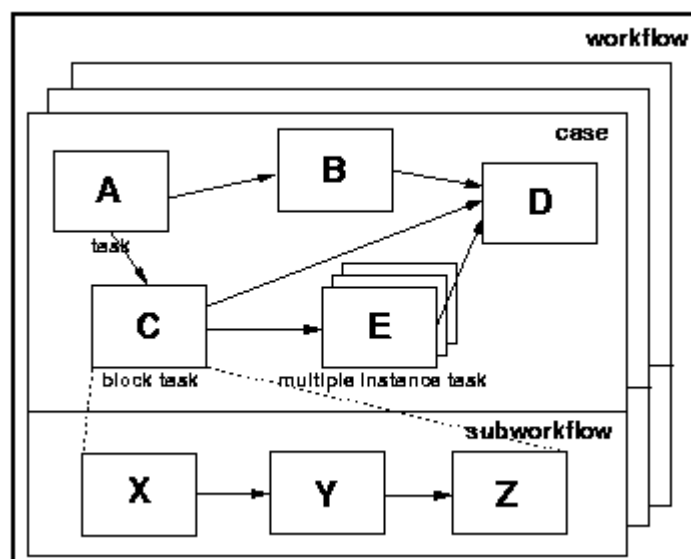
### ***DR#36: Modular IT Design***

## **3.12 Business Process Management**

BPM was introduced at 1.3.4.1.

### **3.12.1 Workflow Processes**

BPM systems, that is, Workflow Management Systems (WMSs), provide Business Management functionalities by enabling Managers to control Business Processes. Each Business Process (or “Workflow”) is defined within the BPM system and consists of a collection of related Workflow Tasks (see Figure 21).



**Figure 21: Components of a Workflow (Russell et al. 2005)**

The WMS manages each Task. For each Task the first step is for the Task to be “allocated” to a person (or other “resource”). This can be achieved in a number of defined ways including allocating the task to a particular person or making it available for the next “free” person.

#### ***DR#37: Task Allocation***

After allocation, the WMS manages the Task to its completion.

The BPM methodology was discovered in the search for a way to model organisational system and access control constraints.

### **3.12.2 Organisational Systems**

Much research, such as in Systems Theory (Maani & Cavana 2002, pp. 6-7) and Information Systems (Oz 2004, pp. 11-17), has conceptualized organisations as systems. Business Management expert Michael Gerber (Gerber 1995, pp. 91-92) points out that in franchised businesses “[t]he system runs the business. The people run the system... The system integrates all the elements required to make a business work”. The point is that IT is tending to move from a position where it simply provided tools for workers to use to a point where it plays is an integral part in the management of organisations.

An *Organisational Concept* that facilitates this management role while meeting IT and regulatory requirements was therefore sought. By incorporating the SOA idea of describing work by Services and the basic business concept that organisations exist to supply societal demands, the following Organisational Concept was chosen:

*The CMS **Organisational Concept** is that “An Organisation exists to provide Services to Clients”.*

Services include the provision of goods. In other words, the provision of goods is just a type of Service.



### 3.12.3 Business Constraints and Workflow Management

BPM models “Workflow Tasks”. BPM Workflow Tasks (“BPM Tasks”) are modular units of work not dissimilar to SOA Services. Furthermore, BPM research by Russell et al. (2005) detailed over 40 constraints that applied to workflow processes and one in particular related to workflow authorisations. Warner & Atluri (2006) also looked at authorisation in their work focussing on detecting collusion between workers. It is evident that authorisations have two different aspects: an access control aspect which has to do with accessing information, and a workflow aspect which has to do with allocating workflow tasks. Access control constraints can therefore be modelled in a BPM based system.

Business constraints in BPM are modelled as dependency rules between workflow tasks.<sup>26</sup> For example, to uphold the Separation of Duty requirement that different persons perform two related tasks, a dependency rule will be applied by the system to ensure that the second task is not allocated to the person who performed the first task. In another example, where it is necessary that one task be completed before another is commenced, the system will check the dependency rule relating to the second task (to ensure that the first task is complete) before it allows its commencement. The constraints detailed by Russell et al. (2005) relate to such things as task allocation, delegation, automatic execution as well as the Separation of Duties requirement.

#### **Useful Aspects:**

BPM research is much more advanced and sophisticated in handling constraints than Access Control methodologies, which have struggled with the issue. Using BPM, an authorisation can be treated as a workflow task (see 3.12.4). The significance of this is that Access Control administration can be managed through workflow tasks that are administered by a Workflow Management System (WMS). In other words, **Workflow Management Systems can be used to administer access control.**

---

<sup>26</sup> A “Workflow” (by definition) defines the relationships (dependencies) between individual tasks. A Workflow can be represented on a “Flow Diagram” like in Figure 21.

WMSs have the capacity to handle SOA type Services if the Services are defined in a similar way to BPM Tasks. They can also conceivably handle Authorisation Timing (see 3.7.4) and Authorisation Ordering (see 3.7.5) requirements, amongst others.

#### **3.12.4 Task Based Authorisation**

The idea of using a task-based approach to manage access control authorisations is not new. Thomas & Sandhu (1997) outlined a family of “Task-based Authorisation Control (TBAC)” models. They detailed many of the advantages that such an approach has over traditional subject-object view of access control. Possibly the most important advantage is apparent in the following quote:

*TBAC can form the basis of self-administering security models as security administration can be coupled and automated with task activation and termination events.*

Bertino et al. (1999), Botha & Eloff (2001), Atluri & Warner (2005), and Hung & Karlapalem (2003) are other researchers who consider authorisation as part of workflow.

##### **Useful Aspects:**

There is evidence that workflow management techniques can be used to handle access control authorisations. Thomas & Sandhu (1997) indicate that a major advantage over traditional methods is that self-administration through automation with task based workflow events is possible.

##### **Problem Aspect:**

Most current task-based authorisation mechanisms have focussed on standard *ex ante* authorisations that do not cater for events such as emergencies (see 3.7.4).

#### **3.12.5 Task Based Access**

This subsection ties together two preceding ideas. In 3.12.3 the idea of using workflow tasks to enable the modelling of business constraints was introduced; in 3.5.1 the concept of using programs in well formed transactions was introduced. For

the Clark-Wilson model to be applied, a “smaller component” than a program is needed. Workflow Tasks are such smaller components.

At the data access level<sup>27</sup> Workflow Tasks can be the TPs that manipulate the data. In other words, **Workflow Tasks that are designed to perform Well Formed Transactions on stored data can be used as the mechanism to access all data.** Access by users to data can then be controlled by controlling access to Workflow Tasks rather than the actual data (as required by Clark-Wilson). If Workflow Tasks are designed to meet the Clark-Wilson certification and enforcement rules, access can be managed by a WMS in a way that assures integrity.

### 3.12.6 Standard Business Processes

Workflows are designed to meet all the standard Business Management processes. These include Personnel Management. A large part of Personnel Management is to categorise persons into organisational positions. Tasks are then allocated to the persons according to their organisational position, workload and other constraints.

#### ***DR#38: Personnel Management***

The accounting and auditing processes can also be managed by the WMS. When tasks are performed, the necessary information can either be extracted from standard business data or be input as part of the Task.

#### ***DR#39: Business Accountability***

#### ***DR#40: Business Auditing***

A WMS can also keep track of business resources. When Tasks are performed they use resources. Tasks can trigger other Workflows that deal with resupplying the used resources.

#### ***DR#41: Inventory Management***

A WMS stores information about the goods and services offered by the business. This information can be used for marketing purposes. For example, an online catalogue can be made available to the business website to allow customers to peruse

---

<sup>27</sup> Where the Access Control System determines if the user has the *right* to access the data.

the business's products. Pictures and other product information can be made available.

***DR#42: Goods/Services Cataloguing***

***DR#43: Information for Advertising***

### **3.13 Legal Compliance**

There are two main types of laws that affect organisations: Public Laws and Private Laws. Public Laws are rules that regulators have put in place that directly govern the operations of business. Non-compliance with these laws can result in fines and imprisonment. Private Laws govern contracts that citizens make with one-another and with businesses. Non-compliance with these laws can result in compensation payouts through litigation.

Rather than focussing directly on specific laws and regulations, this section looks at a number of legal principles that are relevant to organisations and IT-based business management systems.

#### **3.13.1 Organisational Compliance**

Regulation at a legal, industry or professional level (collectively “regulations”) will mandate or drive organisational practices and non-compliance may attract governmental and/or legally sanctioned penalties. Generally regulations are designed to ensure that the interests of citizens, or more specifically organisational clients, are protected.

From an organisation's point of view, regulations form part of the business environment. While professionally sound organisations will normally maintain compliance with the regulations that affect them, other organisations may fail to comply due to oversight; unfamiliarity; lack of knowledge or understanding; or simply to save compliance costs (see 1.3.1). A particular concern is where organisations adopt (or are guided by) risk management approaches (see 3.2.1) that target ‘risk factors’ rather than holistic attempts to secure full compliance.

In the first instance, compliance is usually effected through manipulation of the organisation's written policies and procedures. However, for compliance to be real (that is, effective), this must be translated into actual practice which is dependent on the organisation's management system (in particular, process management). In other words, the organisation must *practice* what its policies and procedures preach. If the management system is IT-based, the computer system must be compliant. A fundamental research question therefore arises: "Can an organisation's computer system be compliant with specified regulations?"

There are many laws and regulations that affect organisations. Some affect all organisations while others are specific to industry sectors (for example, manufacturing) or organisational type (for example, company or non-profit organisation).

This research primarily deals with laws and regulations that directly affect IT-based business management systems. However, it also considers non-IT related laws and regulations that can be managed by IT-based business management systems.

### **3.13.2 Privacy Requirements**

Privacy was introduced at 1.8.1.

The fundamental IT Design Requirements of *Information Protection* and *Need-to-Know Access* were enunciated at 3.2.1 and 3.7.1, respectively. These two requirements form the basis of privacy protection for the Model, and each represent long established Computer Security principles.

The approach taken in the dissertation is to base the Model on these Computer Security principles and to show that in doing this it is compliant with Privacy Laws. Compliance with Privacy Law is thus one of the *tests* for the Model. The issue is therefore dealt with in Chapter 6.

Another aspect to Privacy is its relationship with Consent.

### 3.13.3 Client Consent

Consent was introduced at 1.8.2.

With regard to a hospital patient, as well as the obvious Consent that is required before a treatment is given, there is the aspect of Consent required to use and pass on Personally Identified Data (PID) provided by the patient. Consent to access PID is central to maintaining Privacy.

There are two basic types of Consent (Clarke 2002, ; HealthConnect 2002, ; Coiera & Clarke 2004): *Express Consent* and *Implied Consent*. Express Consent requires that an explicit indication of Consent (such as in writing, electronically or verbally) be given. Implied Consent is Consent that can be gleaned from the circumstances or a person's actions.

#### ***DR#44: Written Client Consent***

#### ***DR#45: Digital Client Consent***

#### ***DR#46: Verbal Client Consent***

#### ***DR#47: Implied Client Consent***

There is more to Consent than just the securing of Consent. For Consent to be legitimate it must be given *voluntarily* and not under any coercion. The client must be properly informed of the implications of their Consent. The client must also have the *capacity* to give Consent. That is, the client's mental and physical state must be sufficient for him or her to be able to give legitimate *Informed Consent*.

#### ***DR#48: Informed Client Consent***

Where clients are *incapable* of giving Consent, for example, if they are underage or unconscious, an *agent* (often the next-of-kin) may be asked to Consent on their behalf. On occasions where this is not practicable, for example, in a medical emergency, Consent may be implied on the basis that the services being provided are in the clients' best interests. In this case it could be considered that the practitioner (Manager/Worker) is acting as the clients' *default agent*.

#### ***DR#49: Agent Client Consent***

In cases where an *agent* is acting on behalf of a client, the *agent* must be given the information necessary to make an informed decision (that is, to give an Informed Consent).

In other cases there may be policies or client directives/permissions in place to inform “supporting agents” about private client information. For example, a hospital patient may give permission for their condition to be reported to specified carers, friends and relatives.

#### ***DR#50: Informing Agents***

##### **Useful Aspects:**

Consent to access PID is necessary. For Consent to be legitimate it must be Informed Consent. The adequate provision of information to the client is therefore required.

In the health domain Consent is more often associated with the provision of treatment to the client than access to the client’s PID.

The role of client agents is an important consideration.

##### **Problem Aspects:**

If Consent is required, how can an IT system handle this? There is a need to incorporate a “consent management system” into the IT system. While paper-based systems are currently the norm in handling Consent, it ultimately needs to be integrated into the IT system. More specifically, the question that needs to be answered is: “Can consent management be an integral part of workflow management?”

#### **3.13.4 Legal Accountability**

A fundamental expectation of businesses is that they are to be accountable for the goods and services that they provide and for their internal business processes. While there are myriad regulations that apply in practice, the overall need for legal accountability should not be lost. The following subsection is an example of an Act,

albeit from the United States (US), that relates to the primary issue of accountability in IT-based management systems.

#### **3.13.4.1 Sarbanes-Oxley (SOX)**

The Sarbanes-Oxley Act of 2002, commonly called SOX or Sarbox, legislation establishes new or enhanced standards for all US public company boards, management and public accounting firms. It was enacted in response to a number of major corporate and accounting scandals (for example, the Enron scandal) (Wikipedia 2008c).

The Act contains eleven titles that describe specific mandates and requirements for financial reporting. It covers issues such as auditor independence, corporate governance, internal control assessment and enhanced financial disclosure. The Act also establishes a new quasi-public agency, the Public Company Accounting Oversight Board, or PCAOB, charged with overseeing, regulating, inspecting, and disciplining accounting firms in their roles as auditors of public companies (Wikipedia 2008c).

Severe non-compliance penalties apply. These include financial penalties and the provision for offenders to be imprisoned for up to 20 years. SOX has relevance to organisational IT systems.

*The most contentious aspect of SOX is section 404, which requires management and the external auditor to report on the adequacy of the company's internal control over financial reporting (ICFR). The financial reporting processes of many companies depend to some extent on IT systems. Therefore, Information technology controls that specifically address financial risks may be within the scope of a SOX 404 assessment. Chief information officers are typically responsible for the IT organisation and IT personnel may be directly involved in SOX compliance efforts. The SOX 404 guidance requires the usage of an internal control framework, such as the COSO framework. The IT Governance Institute's "COBIT: Control Objectives of Information*



*and Related Technology” is also used by many companies.*  
(Wikipedia 2008c)

**Important Aspects:**

Panko & Ordway (2005) describe the nature of the controls. They point out that the controls are to help organisations achieve their goals, such as producing accurate financial reports, despite the presence of threats. The purpose of the controls is to give *reasonable assurance* that the organisation will meet its goals.

Kaarst-Brown & Kelly (2005) state that the principal accountability of the Chief Information Officers (CIO) is “to ensure that every step of a company’s business process is documented and audited, and that all systems are in agreement and enforcing appropriate internal controls”.

SOX is concerned with accountability compliance. The fact that each step in the business process needs to be documented and audited aligns to that required to ensure privacy compliance. There is thus the potential to tie privacy and accountability compliance together in some way.

Kaarst-Brown & Kelly also point out that there is a trend towards even more legislation which involves IT. They name the Anti-Spam Act 2004, and Privacy Acts – PIPEDA 2004 in Canada and the European Directive on Data Privacy 2000 in the EU.

**3.13.5 Service Contracts**

Service Contracts were introduced at 1.7.

In 1.7.1 two contract types relating to the delivery of Services (see Figure 8) are defined. The first is between the Organisation and the Worker and involves a contract to perform work (Client-Services) for the Organisation (that is, an employment contract). The second is between the Organisation and the Client and involves a contract for the delivery of Organisational Services to the Client.

***DR#51: Employment Contracts***

With regard to Organisation/Client contracts, as previously defined, Organisations exist to provide Services to Clients. This involves the Client contacting the organisation (or vice versa) with the intention of utilising one of the Organisation's Services. Some sort of arrangement or agreement is then reached regarding the provision of the Service. The Service is provided and due payment is made by the Client.

Normally both the Organisation and the Client will be happy about the Service provision, and that will be the end of the matter. However, on occasions there will be some disagreement regarding the provision of the Service. Often this will be resolved by the parties through negotiation. Where a dispute is unable to be resolved by the parties, it may ultimately be adjudicated by a court, which will engage in a close analysis of the terms of the contract, against the backdrop of any relevant statute and case law, to render a judgment in the matter.

When a case is made against an Organisation, it is generally on the basis that the Organisation failed to fulfil its contractual responsibilities. To defend its position it is vital that the Organisation has verifiable and adequate records regarding the provision of the Service. For example, can they prove that the Client gave Informed Consent for the Service to be provided, and can they prove exactly what Services were provided?

#### **3.13.5.1 Contract Law Basics**

Contract Law is a complex topic, but the following points to the relevant aspects of contracts.

Willmott et al. (2005, pp. 27,28) describe the five elements of a contract as agreement (*offer* and *acceptance*), *certainty*, *intention* to create legal relations, and *consideration*.

Generally, an agreement is evidenced by a meeting of the minds of the contracting parties, stemming from the making of an *offer* (by the *offeror*) and its *acceptance* (by the *offeree*). The terms contained in the agreement must be sufficiently *certain* to create clear legal obligations and entitlements. *Intention* to create legal relations relates to whether the parties wish their agreement to have legal force. *Consideration*

is seen as the price paid, or value given (whether or not monetary) for the promise(s) contained in the contract.

**Important Aspect:**

In the management of Services it ought to be possible to ensure that service contracts exist and that the elements of *offer*, *acceptance*, *certainty*, *intention* and *consideration* are clearly defined and documented where practicable.

### **3.13.5.2 Service Contract Requirements**

Design Requirements for Service Contracts must be written from the point of the Organisation, as the Model defines the functionality of the Organisation's IT-based business management System. Managers are charged with the responsibility of offering Service Contracts on the Organisation's behalf. The Client may accept the Service Contract that is offered.

***DR#52: Managerial Offer***

***DR#53: Client Acceptance***

Service Contracts will be of various forms, depending on the nature of the Service. Both Managerial Offers and Client Acceptances will also take various forms. These forms mirror the various forms of Consent described at 3.13.3. The Model requires that Client Acceptance be incorporated in Service Consent.

***DR#54: Service Consent***

## **3.14 Design Requirements Table**

Table 1 lists the Design Requirements that were extracted in the preceding Literature Review process. The purpose of the table is to collate the Design Requirements for reference purposes. The background colours used for each Research Field correspond to those in the Model Concepts Relationship Diagram (Figure 22).

Research Field	No.	Design Requirement
Computer Security	1	Information Protection
	2	Reduce Security Risks
	3	Information Security Management
	4	Client Based Permission
	5	Client Information Reviewal
	6	Client Control
Role-Based Access Control	7	Hierarchical Groups
	8	Represent Contexts
	9	Multiple Contexts
	10	Represent Constraints
Set Theory	11	Set Operations
Clark-Wilson Model	12	TP User Authentication
	13	Certified User List Management
	14	Certified TPs
	15	User-TP Relation Specification
	16	TP-Data Relation Specification
	17	Separation of Duties
Health Informatics	18	Cooperative Practices
	19	Multiple Authorisations
	20	Guaranteed Access
Professional Access Control	21	Need-to-Know Access
	22	Reduce Administration
	23	Service Teams
	24	Associate Authorisation
	25	Emergency Authorisation
	26	Critical Authorisation
	27	Authorisation Timing
	28	Authorisation Order
Purpose Based Access Control	29	Purpose Binding
Service Oriented Architecture	30	Represent Services
Cloud Computing	31	Inter-Domain Roles
	32	Global Access Protocols
	33	Efficient Rule Processing
Generic Programming	34	System Interoperability
	35	User-Friendly Groups & Services
	36	Modular IT Design
Business Process Management	37	Task Allocation
	38	Personnel Management
	39	Business Accountability
	40	Business Auditing
	41	Inventory Management
	42	Goods/Services Cataloguing
	43	Information for Advertising
Legal Compliance	44	Written Client Consent
	45	Digital Client Consent
	46	Verbal Client Consent
	47	Implied Client Consent
	48	Informed Client Consent
	49	Agent Client Consent
	50	Informing Agents
	51	Employment Contracts
	52	Managerial Offer
	53	Client Acceptance
	54	Service Consent

Table 1: Design Requirements

### 3.15 Conclusion

The Literature Review gives an overview of the significant Research Fields (and Research Methodologies) that were investigated. It highlights the pros and cons associated with the main Research Topics. These will be followed up in the Results and Discussion (Chapter 7).

The Design Requirements extracted through the Literature Review process are noted in the relevant Research Topic subsection. The entire set of Design Requirements is collated in Table 1 at 3.14.

The early sections of the chapter investigated IT Research Fields with a focus on Access Control Models and techniques. This revealed that while existing models contained many useful concepts, such as *groups* and *well formed transactions*, none provided a general purpose access control solution. It also highlighted that existing solutions lacked effective techniques for *constraint modelling*, *efficient administration* and *purpose binding*.

The investigation of Business Management methodologies revealed that access control functionalities could be incorporated into WMSs to provide *constraint modelling* and *efficient administration* and that SOA type Services could incorporate *purpose binding*. It also highlighted the need for compliant systems.

The Review of Legal Compliance revealed connections between Privacy, Consent and Accountability, and the potential of Service Contracts to facilitate Compliance. It also showed the various Accountability and Authorisation functionalities that were required in a consent-based service management system.

The Design Requirements essentially define the research problem. The following two chapters explore the proposed solution. Chapter 4 details the Model Concepts that form the basis of the Model while the Model itself is described in Chapter 5.

## Chapter 4

# Model Concepts

---

### 4.1 Introduction

The Model Concepts Relationship (MCR) Diagram (Figure 22) is central to the dissertation as it summarises the research. It shows the relationships between the Design Requirements and Model Concepts as well as the relationships between Model Concepts. Figure 11 (Research Methodology in Practice) explains the multi-dimensional nature of the relationships.

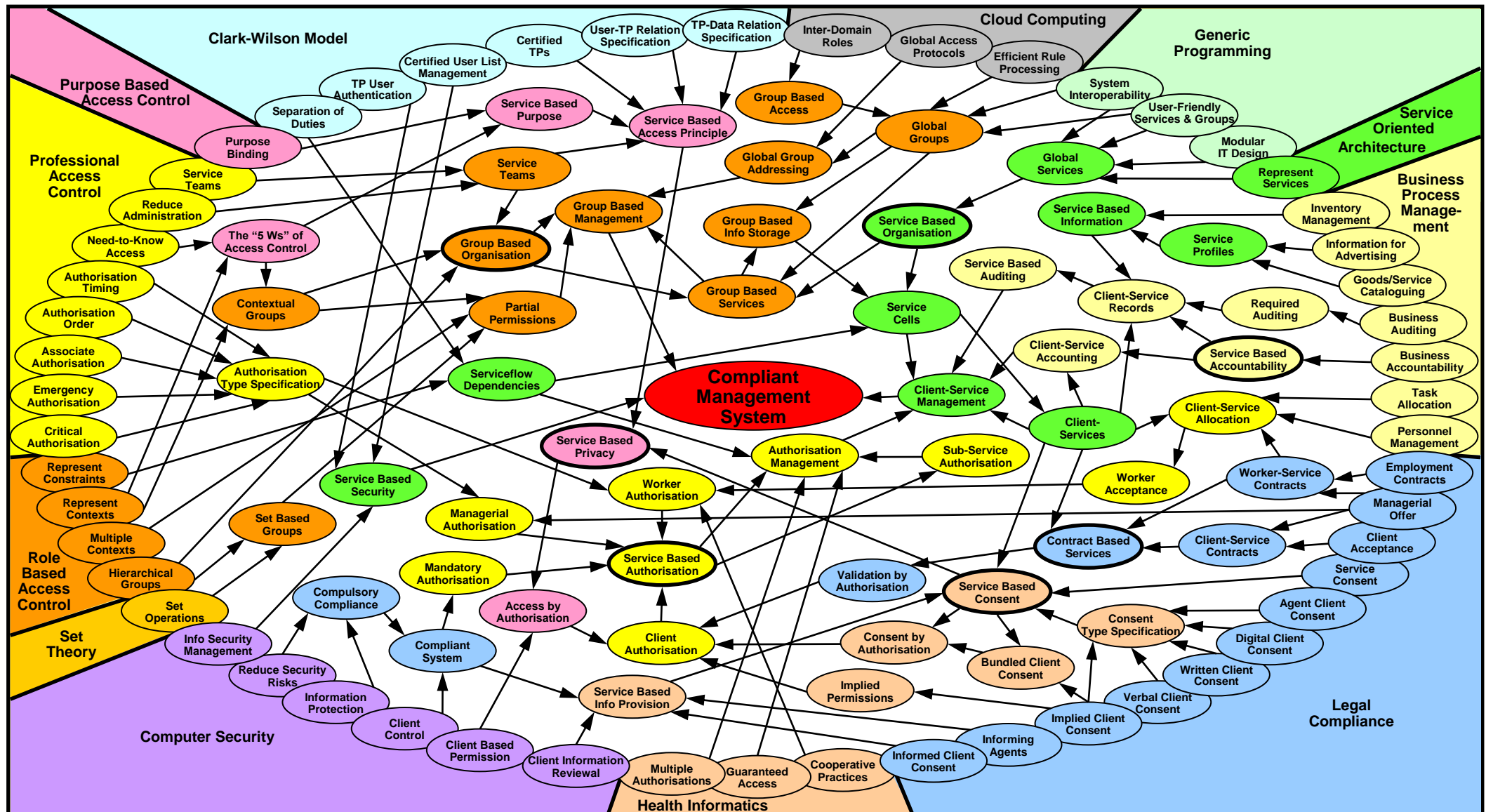
The following section (4.2) explains the MCR Diagram. The bulk of the chapter contains a section for each General Concept detailing each of the related Model Concepts (see Figure 3). The Model Concepts are denoted in the following manner:

*C#0: Example Concept*

At 4.10 the Model Concepts are collated into a single table. A Conclusion follows.

### 4.2 Model Concepts Relationship Diagram

The MCR Diagram consists of an outer Research Field area and three sets of ellipses – the “outer”, “inner” and “mid” sets. The outer ellipses (that form a larger ellipse) each represent a Design Requirement; the inner ellipse represents the proposed CMS; and the mid ellipses (in the adjoining white space) represent the Model Concepts. The arrows show the relationships between the ellipses. The outer Research Field area categorises the Design Requirement according to their field, reflecting Table 1 (Design Requirements). The colours of the Model Concepts match that of their corresponding General Concept in Figure 3 (Model Concepts).



**Figure 22: Model Concepts Relationship Diagram**

## 4.3 Services

### 4.3.1 C#1: Global Services

From: DR#30: Represent Services, DR#34: System Interoperability, DR#35: User-Friendly Groups & Services, DR#36: Modular IT Design

To: C#2: Service Based Organisation

*A **Global Service** is a simple standardised unit of commerce.*<sup>28</sup>

A Global Service is “commercial” in that Organisations provides these Services and customers seek them.

From a customer (Client) perspective a Global Service is a Service (or product) that the customer wishes to find and perhaps purchase. The customer may search for the Service locally, nationally or globally. Often the customer will wish to compare services provided by multiple organisations before selecting an alternative. Global Service mechanisms must be user-friendly to encourage customers to use them.

From an Organisation perspective a Global Service is a unit of production or work that the Organisation undertakes for some gain, usually financial. The Organisation will advertise its Services locally, nationally or globally, and manage production and Service Delivery. Service Management mechanisms must also be user-friendly for Managers and Workers to promote workplace efficiency.

The purpose of standardisation in this commercial sense is to facilitate the reuse of Service descriptions and Service processes, and to facilitate cost effective advertising and customer searching. This enables system interoperability and modular IT design.

---

<sup>28</sup> A unit in this sense is a standard measure (as an OOP Object is now a “standard”). The dissertation proposes and defines such a “Standard Service”.



### 4.3.2 C#2: Service Based Organisation

From: C#1: Global Services

To: C#4: Service Cells, C#16: Group Based Services

*A **Service Based Organisation** is one that defines all its work in terms of Global Services.*

From an external perspective the Organisation reveals its Services to customers (often by advertisement) and to commercial partners, such as suppliers (through information exchanges).

From an internal perspective the Organisation manages its business by managing Service Delivery, which includes the production and/or warehousing of goods.

In BPM business processes are defined by Workflows that in turn define the dependencies between tasks. In the Model business processes are defined by *Serviceflows* that in turn define the dependencies between Services.

### 4.3.3 C#3: Serviceflow Dependencies

From: DR#10: Represent Constraints, DR#17: Separation of Duties

To: C#4: Service Cells, C#52: Authorisation Management

***Serviceflow Dependencies** specify the dependencies between Services and component Sub-Services.<sup>29</sup>*

*Serviceflows* are the method by which business processes are defined. They are to Business Management what a *recipe* is to a cake. They show such things as the order in which related Services must be performed and any other dependencies between Services. By enabling dependencies between Services (or Sub-Services)<sup>30</sup> to be

---

<sup>29</sup> The term *Serviceflow* is defined in this dissertation. The concept is identical to the BPM concept of having *Workflows* composed of *Tasks* – here *Serviceflows* are composed of Services.

<sup>30</sup> A *Sub-Service* IS itself a Service – the “*Sub*” prefix merely refers to the hierarchical nature of Services where any Service can be a component of a higher level Service.

specified, Serviceflows can represent constraints such as the Separation of Duties requirement (where two related Services must be performed by different persons).

The term *constraint* (an Access Control term) generally relates to a limitation on a single Service (or action), for example, “a constraint on Service B is that it must start after Service A is completed”. The term *dependency* between two Services (a Business Process term) essentially is a different way of saying the same thing, for example, “Service B is dependant on Service A’s completion”. When speaking of Serviceflows as Business Processes the term *dependency* is perhaps more appropriate. So, **Serviceflow Dependencies model and/or specify constraints.**

#### 4.3.4 C#4: Service Cells

From: C#2: Service Based Organisation, C#3: Serviceflow Dependencies, C#17: Group Based Information Storage

To: C#5: Client-Services, C#8: Client-Service Management

*A **Service Cell** is the technical (IT) definition and specification of a standardised Global Service.*

In relation to franchised business systems Gerber (Gerber 1995, pp. 91-92) states that “[t]he system integrates all the elements required to make a business work. It transforms the business into an *organism*, driven by the integrity and orchestration of its parts”.

Organisms consist of cells. The cell terminology defined here is referred to as the *Service Cell Paradigm*. It is used to portray the idea of Service Based Organisations being composed of Service Components (as organisms are composed of cells); these components having a basic design that can be adapted and replicated.

Moreover, diverse organisms are composed of similar cell components in the same way that diverse organisations are composed of similar Service Components. Some cells are on the outside of organisms and are visible to the world, while others are internal and provide some internal function to the organism; as some Organisational Services are visible to the world while others provide internal management functions.

An organism's DNA defines its cellular designs; as a Service Based Organisation's *Service Templates* define its Services. Each Organisational Service is represented in the IT System by a Service Template from which each instance of the Service is generated. Each Service Template is an adaptation/derivation of the standard Service Cell.<sup>31</sup>

The challenge is thus to specify the design of the standard Service Cell in such a way as to allow all Service Templates to be definable by the standard.

#### **4.3.5 C#5: Client-Services**

From: C#4: Service Cells

To: C#8: Client-Service Management, C#25: Contract Based Services, C#33: Service Based Consent, C#39: Client-Service Accounting, C#41: Client-Service Records, C#46: Client-Service Allocation

*A **Client-Service** is one instance of the delivery of an Organisational Service to a Client.*

Each Client-Service (instance) is generated within the IT System from the Service Template for the particular Service.

#### **4.3.6 C#6: Service Profiles**

From: DR#42: Goods/Services Cataloguing, DR#43: Information for Advertising

To: C#7: Service Based Information

*A **Service Profile** is the set of standard information that describes a Service.*

The Service Profile contains information for potential Customers, to Clients receiving the Service, to Workers performing the Service, and to Managers managing the Service. It describes what is being offered to the Client.

---

<sup>31</sup> As each *Object Class* is an adaptation/derivation of the standard *Object* in OOP.

#### **4.3.7 C#7: Service Based Information**

From: DR#41: Inventory Management, C#6: Service Profiles

To: C#41: Client-Service Records

*Service Based Information encompasses all the information associated with a Client-Service, including the Service Profile and Client-Service Management data.*

The Service Template defines the items of Service Based Information for a particular Service. Some of these information items are mandatory (such as the Client's name and the date and time) while others are optional (such as pictures of the Service).

#### **4.3.8 C#8: Client-Service Management**

From: C#4: Service Cells, C#5: Client-Services, C#39: Client-Service Accounting, C#42: Service Based Auditing, C#52: Authorisation Management

To: CMS

*Client-Service Management incorporates the management of all Organisational Services.*

Client-Service Management is the overarching management methodology used in the CMS. It equates to the Business Management methodology for Service Based Organisations.

#### **4.3.9 C#9: Service Based Security**

From: DR#3: Information Security Management, DR#12: TP User Authentication, DR#13: Certified User List Management

To: CMS

*Service Based Security is the methodology whereby Information Security is maintained in the course of the management of Organisational Services.*

Essentially, users can only access and manipulate data in prescribed ways in the course of performing Client-Services, and can only perform Client-Services for which they are authorised. This deals with the issue of Access Control.

However, there are many other functions required to facilitate Information Security. These include auditing, back-ups, encryption of sensitive data and virus protection, to name a few. Service Based Security deals with all these issues by providing Services for each required function. Thus all security management is Service-based.

## 4.4 Groups

### 4.4.1 C#10: Service Teams

From: DR#22: Reduce Administration, DR#23: Service Teams

To: C#12: Group Based Organisation, C#29: Service Based Access Principle

*A **Service Team** is an Organisational group containing members (Managers and Workers) who are Authorised to perform Services for a specified Client.*

By definition, each Client of the Organisation has his or her own personal Service Team. While it is possible that multiple Clients have the same Service Team, Service Teams must be able to be tailored for individual Clients. This requires a mechanism whereby a Service Team is provided to each Client rather than one where each Client is assigned to an existing Service Team (though it must be possible that the provision of a Service Team to a Client entails assigning an existing Service Team to them).

In short, the relationship is that the Service Team *belongs* to the Client, not the Client *belongs* to the Service Team. This is required because the primary question that must be answered by the IT System is “who is allowed to deal with the Client”; the answer is “those on the Client’s Service Team”.

Service Teams are represented by Groups in the IT System, and Managers and Workers on the Service Team are recognised by the IT System as such by their “membership” in the “Service Team” Group. In order for the IT System to allow a

Service to be performed for the Client, it is **mandatory** that the Manager/Worker be a *member* of the Client's Service Team Group.

When a person initially becomes a Client of the Organisation, a Service Team Group is automatically created for the Client in the IT System and the group persists while they remain a Client of the Organisation. When a Service Team is created, one or more "default members" will automatically be added to the Group.

If a Manager/Worker wishes to perform a Service for a Client, and is not currently in the Client's Service Team, the System must provide mechanisms to allow the Manager/Worker to become a member. This is necessary to deal with cooperative work, emergencies and other unforeseen occurrences where membership is legitimate.

All group memberships require legitimate Authorisations. In current systems the authorisations are normally provided by system administrators or their delegates. These persons require considerable system knowledge in order to perform these functions in a secure manner.

However, in practice system administrators must be guided by management when making such authorisations. This is because it is the Managers (and fellow Workers) who know "who should be doing what" or, more particularly, "which Worker should perform which Service for which Client". By allowing appropriate Managers and Workers to automatically<sup>32</sup> control membership of Service Teams, system administrators are relieved of the task and administration is reduced.

#### **4.4.2 C#11: Contextual Groups**

From: DR#8: Represent Contexts, C#27: The "5 Ws" of Access Control

To: C#12: Group Based Organisation, C#19: Partial Permissions

---

<sup>32</sup> Service requests to non-group members *automatically* add that person to the Service Team. For example, in referring a patient to a Specialist, the Doctor *automatically* authorises the Specialist to join the Patient Care Team.

***Contextual Groups** are Organisational Groups that each represent a specific Context, where membership of the Group confers the context on the Group member.*

“Contexts” represent some aspect or attribute that is used to determine whether a person is Authorised to perform a Service. In effect they specify Organisational rules. Common contexts include role, location and time. A Worker can be required to have an appropriate role (for example, “Nurse”) to perform a Service or may be restricted to performing the Service in a particular location (for example, on “Ward A”) or at a particular time (for example, during the “Morning Shift”). Each context is represented by a corresponding Group in the IT System.

To enforce the “contextual rules” the IT System reviews group memberships to verify that a Worker is Authorised to perform a particular Client-Service. For example, to Authorise a Worker to perform the Service of giving a medication to a Ward A Patient at 11am, the System may check that the Worker is a member of the Patient’s Service Team, the “Nurse” role Group, the “Ward A” Group and the “Morning Shift” Group.

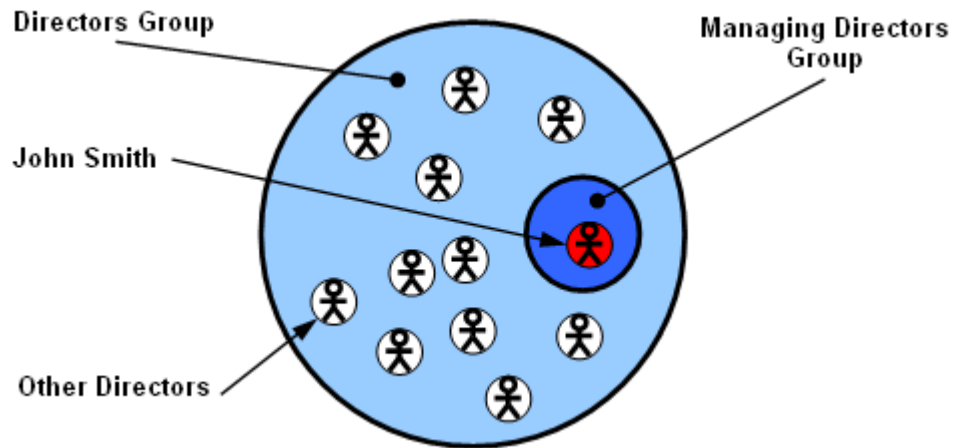
#### **4.4.3 C#12: Group Based Organisation**

From: DR#7: Hierarchical Groups, C#10: Service Team, C#11: Contextual Groups

To: C#16: Group Based Services, C#20: Group Based Management

*A **Group Based Organisation** uses Organisational Groups to define its management structure, its Staff and to specify authorisation rules for Client-Service provision.*

Individuals, teams and positions within the Organisation are defined by Groups. The Groups form hierarchical structures. Figure 23 contains an example where an Organisation’s management structure is represented by hierarchical Groups. John Smith is the Managing Director of the Group Based Organisation. In the diagram he is represented by a Group, the Board of Directors is represented by another Group and the position of Managing Director by yet another Group.



**Figure 23: Group Based Organisation Example**

As mentioned in the previous section, contextual groups are used to specify Organisational rules. This is so because appropriate Group memberships are required by staff before Client-Services can be performed.

Staff members in a Group Based Organisation are managed by placing them in appropriate groups according to their position, role or any other relevant contextual attribute. While individual group memberships will change over time, the overall group structure of the Organisation should remain relatively static except for the creation and deletion of time-based groups such as Project Teams or Short Term Committees.

Furthermore, many Group structures will be common among Organisations, particularly where Organisations use standard management structures and practices. This means that entire organisations can be based on existing “group templates”. This point is discussed in greater detail in Chapter 5.

#### **4.4.4 C#13: Group Based Access**

From: DR#31: Inter-Domain Roles

To: C#14: Global Groups



***Group Based Access** enables an Organisation to allow External Group members to perform Services within the Organisation by extending internal Group membership to the members' External Group.*<sup>33</sup>

Group Based Access enables controlled external access to Services and Service related information. This means that External Groups, particularly role-based Groups in other Organisations, can be allowed to perform internal Services, particularly "Data Access Services".<sup>34</sup>

For example, if a General Practitioner (GP) is a member of the Doctor Group in a General Practice and a hospital makes the Group a member (that is, a Sub-Group) of its own Doctor Group and the GP a member of the Relevant Patient Service Teams (when the patients specify their GP), the GP can then access the medical hospital records of his or her own patients.

#### **4.4.5 C#14: Global Groups**

From: DR#33: Efficient Rule Processing, DR#34: System Interoperability, DR#35: User-Friendly Groups & Services, C#13: Group Based Access

To: C#15: Global Group Addressing, C#16: Group Based Services, C#17: Group Based Information Storage

***Global Groups** are Organisational Groups that are open to memberships from External Groups or are themselves enabled to be members of External Groups.*

The purpose of Global Groups is to allow Organisations to work together and to share information in a secure controlled manner. The essential characteristic of Global Groups is that Groups in other Organisations can be members of Groups within an Organisation. This enables an Organisation to effectively provide any Service to external users and allows Services in one Organisation to automatically use Services in collaborating Organisations.

---

<sup>33</sup> An *External Group* is any Group from another Group Based Organisation.

<sup>34</sup> Information is not accessed directly in the Model but through Data Access Services.

This functionality enables various benefits. Organisations can more easily provide online searching and sales mechanisms to customers, both Business to Consumer (B2C) and Business to Business (B2B). There is potential to automate procurement in supply chains, to advertise and broker Services, and scope to check credentials (such as qualifications and licences) by checking Group Memberships in appropriate Organisational Groups.

From a functional IT perspective, making Access Control decisions through checking Group memberships is efficient, as the processing time for checks is small. This is especially true when compared to the time required to check separate sets of Access Control rules in each domain. Global Groups effectively store the rules as Group memberships.

#### **4.4.6 C#15: Global Group Addressing**

From: DR#32: Global Access Protocols, C#14: Global Groups

To: C#20: Group Based Management

*Global Group Addressing provides all Global Groups with a global internet address.*

The Global Group Address (GGA) could be a Universal Resource Locator (URL) or the like. The purpose of GGAs is to allow Groups to be found from any internet connected computer or phone.

Giving Groups a GGA is an original concept that opens the door to numerous possibilities for online system interactivity. When each individual, team and Organisation is represented and identified with a global address, access rights to information and Services can be managed on a global basis. GGAs effectively enable a “Global Access System” where everyone is effectively using one worldwide computer. This is discussed further in Chapter 7.

#### **4.4.7 C#16: Group Based Services**

From: C#2: Service Based Organisation, C#12: Group Based Organisation

To: C#17: Group Based Information Storage, C#20: Group Based Management

***Group Based Services** are Services that are associated with Groups, can only be activated through that association, and can only be performed by Group members.*

An “Inactive Service” is a Service that has been configured (from a Service Template) but is not released to be used, while an “Active Service” is one that has been configured and released for use. A Service is activated when it is associated with one or more Groups. When a Service is associated with a Group, then the Group Members are able to perform that Service and Clients are able to receive the Service.

#### **4.4.8 C#17: Group Based Information Storage**

From: C#14: Global Groups, C#16: Group Based Services

To: C#4: Service Cells

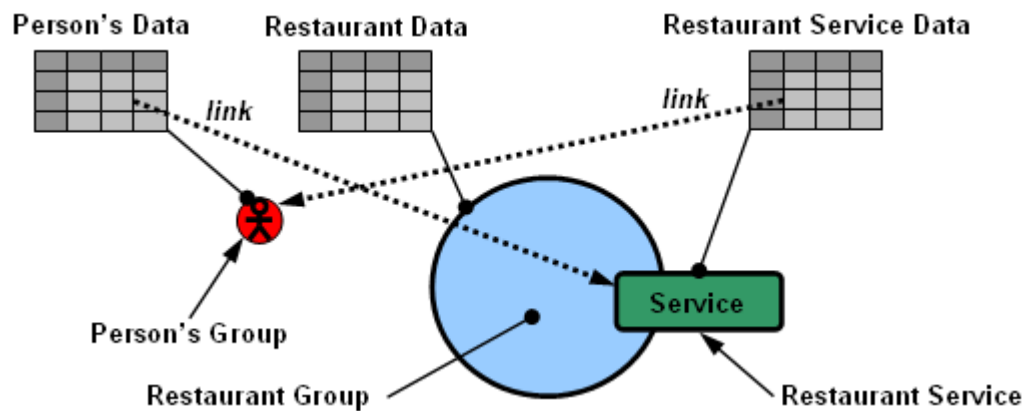
***Group Based Information Storage** is a paradigm for systematic data storage where information describing a global entity is stored in memory locations that are directly associated with the Global Group that represents the entity.*

In the Group Based Information Storage paradigm data/information is of two types: that relating to entities, represented by “Global Groups”, and that relating to what entities do, represented by “Global Services”.<sup>35</sup> For example, in Figure 24 a person books a restaurant meal. The person’s name and contact information are stored with a Global Group that represents them, while data relating to the restaurant booking is stored with the Restaurant Service (including a link to the person’s Global Group). If the restaurant wishes to contact the person, it is directed to the person’s Global Group and the information stored with it. A customer who wishes to retrieve

---

<sup>35</sup> In IT programming terms this is fundamentally different from the way data storage in the OO paradigm where data is stored as Object attributes that are modified through OO Methods (Functions).

information about the booking is directed to the information stored with the Restaurant Service.<sup>36</sup>



**Figure 24: Group Based Information Storage Example**

Since Services are associated with Groups, as they are Group Based Services, Global Service information is indirectly linked to the Global Group that is performing the Service. In the example, the Restaurant Service is performed by the Global Group representing the Restaurant, which means that information about the restaurant booking is indirectly associated with the Restaurant Group. In non-digital terms, if persons desire information about their restaurant booking (maybe they forgot the time) they simply telephone the restaurant. In digital terms, they can use their computer/smartphone to get the information and it will automatically contact the (digital/virtual) restaurant and retrieve the information from the Restaurant Service Data.

#### 4.4.9 C#18: Set Based Groups

From: DR#7: Hierarchical Groups, DR#11: Set Operations

To: C#19: Partial Permissions

***Set Based Groups** are Groups whose membership is defined by a mathematical set.*

---

<sup>36</sup> To access information “Data Access Services” are activated and only relevant information is released.

Set based Groups each have a *Group Type* which defines the common attribute that all Group Members share, for example, all members of the Nurse (the Group Type) Group are nurses and all members of the Ward A Group are working on Ward A.

The important characteristic of Set Based Groups is that associations between and rules pertaining to multiple Groups can be defined using *Set Operations*. For example (see Figure 25), for groups A and B, the *Union* operation  $A \cup B$  can specify where membership of *either* group is required (that is, membership of A OR B), the *Intersection* operation  $A \cap B$  can specify where membership of *both* groups is required (that is, membership of A AND B) and the *Relative Complement* operations  $(A - B)$  can specify where one membership but not another is required (that is, A AND NOT B).

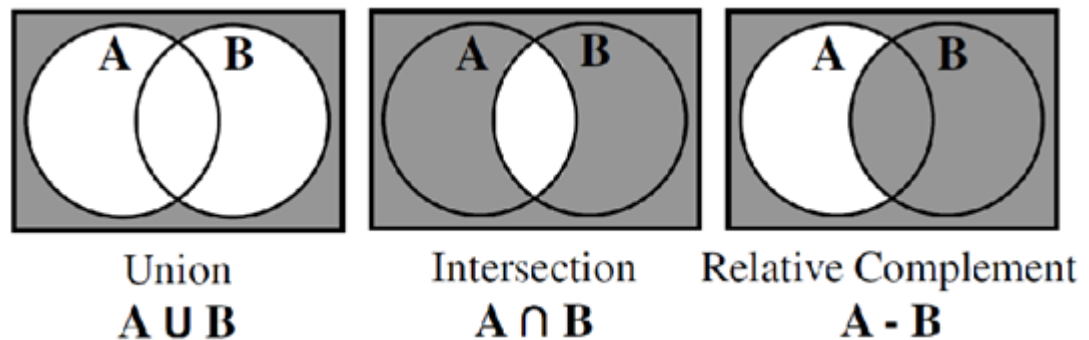


Figure 25: Set Operations

#### 4.4.10 C#19: Partial Permissions

From: DR#9: Multiple Contexts, C#11: Contextual Groups, C#18: Set Based Groups

To: C#20: Group Based Management

***Partial Permissions** are permissions which apply to multiple Groups that are specified through rules stored with and applied to each of the Groups involved.*

Partial Permissions allow Global Access Rules to be specified and applied without the need for separate and perhaps conflicting domain specific Access Control mechanisms. Partial Permissions were defined (de la Motte & Hartnett 2005a) as a general method of applying rules to Groups where the information specifying the

rules is stored as Group Based Information rather than in a separate “Rule Set”. The aim is to ensure that rules can be found simply when inter-domain access is involved; the rules are stored with the entity being queried (that is, the Group), not in some domain specific and/or inaccessible Rule Set.

For Group Based Services the rules relate to Services and are stored with the relevant Groups.<sup>37</sup> For example, to perform an operation on a patient a Doctor must be in both the Patient’s “Service Team” Group and in the “Surgeons” Group. To enable this rule a Partial Permission to perform the “Operation” Service is given to the “Surgeons” Group. This must match with a general Partial Permission given to the “Service Team” Group that requires “Service Team” Group membership. The operation can only be performed if both Partial Permissions exist. The permissions are linked when the Partial Permission is given to the “Surgeons” Group – that Partial Permission being made subject to the existence of the general Partial Permission.

#### **4.4.11 C#20: Group Based Management**

From: C#12: Group Based Organisation, C#15: Global Group Addressing, C#16: Group Based Services, C#19: Partial Permissions

To: CMS

*Group Based Management is the fundamental Business Management methodology employed in Group Based Organisations.*

The organisational structure in a Group Based Organisation is defined by hierarchical Groups. The Group structure includes relatively static Groups like Staff Position Groups, through to more dynamic Groups like Project Teams, to very dynamic Groups like Client Service Teams. The primary Business Management mechanism for controlling what staff members are able to do is Group Based Management because Group Membership is required to perform Services. Group Based

---

<sup>37</sup> Partial Permissions do not only apply to Services, they can be used to specify other rule types that relate to Groups such as Access Control Rules (where the permissions are access *rights*).

Management essentially involves controlling Group Memberships by both manual and automated means.

## 4.5 Service Contracts

### 4.5.1 C#21: Compulsory Compliance

From: DR#1: Information Protection, DR#2: Reduce Security Risks

To: C#22: Compliant System

*Compulsory Compliance describes the imposition that various regulations and legal requirements place on Organisations.*

Compulsory Compliance (“Compliance” for short) is the general concept that covers all the relevant regulatory and legal requirements imposed on any particular Organisation.

Organisations are required to comply with regulations placed on them by governing bodies. Regulations can be imposed by international governing bodies, by different levels of government (whether administratively or by statute), and by industry or professional bodies. Furthermore, Organisations have to comply with the common law and their contractual obligations to Clients and partner/controlling Organisations.

### 4.5.2 C#22: Compliant System

From: DR#6: Client Control, C#21: Compulsory Compliance

To: C#37: Service Based Information Provision, C#43: Mandatory Authorisation

*A Compliant System is a business management system that inherently maintains Compliance.*

The aim of a Compliant System is to provide some level of assurance or guarantee that the Organisation employing the system is Compliant. Essentially there are two types of Compliant Systems: *Reactionary Systems* and *Mandated Systems*.

Reactionary Systems are either configured by the Organisation or their third party agent (usually the system supplier). They deal with imposed requirements by continually being updated or modified in order to maintain their currency.

Mandated Systems are directly or indirectly configured by regulatory bodies. Indirect configuration is where a regulatory body, or its agent, test and officially rate or sanction the system's compliance. Direct configuration occurs where procedures and/or data supplied by the regulatory body are used to control the system.<sup>38</sup>

A Compliant System may contain a combination of Reactionary and Mandated components. The nature of software upgrading and modification and the practicalities of changing compliance requirements will normally necessitate at least some Reactionary components.

#### **4.5.3 C#23: Worker-Service Contracts**

From: DR#51: Employment Contracts, DR#52: Managerial Offer

To: C#25: Contract Based Services, C#46: Client-Service Allocation

***Worker-Service Contracts** are Employment Contracts or Sub-Contracts that are expressed in terms of Organisational Services and specify the rights and responsibilities of the Organisation and the Worker.*

Worker-Service Contracts exist between the Organisation and each of its employees and sub-contractors. Their purpose from a legal perspective is to ensure that Workers are informed of their employment and contractual rights, and of the requirements on them to conform to Organisation policies and practices, particularly with regard to dealing with Clients. Each of the parties must be clear on their respective rights and responsibilities, and know what each is expected to do, and the parameters of their obligations. Thereby any 'expectation gap' can be reduced.

---

<sup>38</sup> A *taxation calculator* is an example of a supplied procedure. *Taxation thresholds and rates* are examples of supplied data.



#### 4.5.4 C#24: Client-Service Contracts

From: DR#52: Managerial Offer, DR#53: Client Acceptance

To: C#25: Contract Based Services

*Client-Service Contracts exist for each Client-Service performed by the Organisation and specify the rights and responsibilities of the Organisation and the Client regarding the Service.*

Managers representing the Organisation decide which particular Services the Organisation will offer to each Client.

Except in circumstances where the Organisation is legally enabled to do otherwise, all Client-Services must be accepted by the Client before they are performed by the Organisation. The types of Client-Service Contract and the forms of acceptance (for example, written, digital, verbal or implied) applicable to a Service are specified in Service Management Information and enacted in Service Management Procedures.

#### 4.5.5 C#25: Contract Based Services

From: C#5: Client-Services, C#23: Worker-Service Contracts, C#24: Client-Service Contracts

To: C#26: Validation by Authorisation

*The Contract Based Services principle specifies that all Client-Services offered by the Organisation are Contract-based and that both Worker-Service Contracts and Client-Service Contracts are in place to cover each Client-Service, before it is performed.*

The Authorisation process for each Client-Service ensures that each Service is “Contract Based”. The process requires that a Manager, on behalf of the Organisation Authorises a Worker to perform a Service for the Client. The Worker can only be Authorised if their Worker-Service Contract covers the delivery of the Service and the Client-Service can only be performed if a Client Service Contract has been offered by the Organisation and accepted by the Client. This means that: 1)

Managers representing the Organisation decide which particular Services the Organisation will offer to each Client and which Worker will perform each Service; and 2) except in circumstances where the Organisation is legally enabled to do otherwise, all Client-Services must be accepted by the Client before they are performed.

The types of Client-Service Contract and the forms of acceptance (for example, written, digital, verbal or implied) applicable to each Service are specified in Service Management Information and documented in Service Management Procedures. It is the CMS's task to ensure that all contractual requirements are met for each Client-Service.

#### **4.5.6 C#26: Validation by Authorisation**

From: C#25: Contract Based Services

To: C#49: Client Authorisation

*The **Validation by Authorisation** principle requires the CMS to ensure compliance by requiring that Authorisation Requests from each party to a Service Contract are received before the Service is validated.*

Parties to a Service Contract register their offer or acceptance to the contract by inputting an Authorisation Request to the CMS. In other words, the CMS communicates with the parties through Authorisation Requests. The role of the CMS, as a Compliant System, is to ensure Compliance with Service Contract requirements. It performs this role by ensuring that it receives all the necessary Authorisation Requests before validating<sup>39</sup> the Service provision.

---

<sup>39</sup> *Validation* in this sense means ensuring that the Service is legitimately offered and accepted, that is, in a Compliant fashion.

## 4.6 Privacy

### 4.6.1 C#27: The “5 Ws” of Access Control

From: DR#8: Represent Contexts, DR#21: Need-to-Know Access

To: C#11: Contextual Groups, C#28: Service Based Purpose

*The 5 Ws of Access Control (Who, What, When, Where and Why) define the five primary Access Control questions that must to be answered in order to grant or deny access requests.*

More specifically, the 5 Ws are:

1. **Who** requires access?
2. **What** is to be accessed?
3. **When** is the access to occur?
4. **Where** is the location of the access? and
5. **Why** is the access required?

The 5 Ws are general in nature and define the essence of Access Control. While there are situations where more or less information is needed according to the level of security required, they are useful in describing the fundamental design requirements for access control systems.

### 4.6.2 C#28: Service Based Purpose

From: DR#29: Purpose Binding, C#27: The “5 Ws” of Access Control

To: C#29: Service Based Access Principle

*Service Based Purpose describes the fundamental assumption that the purpose for accessing a Client’s information in an Organisational environment is to perform a Service for a Client.*

Of the 5 Ws, the first four (who, what, when and where) are all clearly evident and easy to define. The last (why) is open to speculation. How does one explain the “reason” for an access request to the (access control) system? By writing a comment or ticking a box? How inconvenient is this to do? And how does the system automatically analyse the reason and decide if it is legitimate? These questions raise potentially difficult challenges.

To answer the *why* question, the system is looking to confirm whether the “Purpose of Access” matches the Services that the *accessor* (the *who*) is Authorised to perform. Service Based Purpose makes the requested connection between “Access” and “Service” by making a simple assumption – the Purpose of Access is to perform a Service for the Client.

#### **4.6.3 C#29: Service Based Access Principle**

From: DR#14: Certified TPs, DR#15: User-TP Relation Specification, DR#16: TP-Data Relation Specification, C#10: Service Team, C#28: Service Based Purpose

To: C#30: Service Based Privacy

*The **Service Based Access Principle** is that Services access the information they require, and access to who performs Services is restricted to Authorised personnel.*

The Service Based Access Principle comes directly from Clark and Wilson’s (1987) concept that users should not access data directly but only through certified Transformation Procedures (TPs). It makes Services the TPs (that is, the Service tasks or procedures) and controls user access to them. This means that users do not access information, particularly Client PID, directly, but only through certified Services.

#### **4.6.4 C#30: Service Based Privacy**

From: C#29: Service Based Access Principle, C#33: Service Based Consent

To: C#31: Access by Authorisation

*Service Based Privacy uses Service Based Consent to restrict access to Services and the Service Based Access Principle to restrict access to Client Information.*<sup>40 41</sup>

Service Based Consent incorporates consent to access relevant service related information with the Client's consent to receive the Service. Expressed another way, consent to access related data is part of the Service Contract that must be accepted by the Client prior to Service Delivery.

The Service Based Access Principle only allows access to Client information in the performance of accepted Services.

Service Based Privacy maintains Privacy by using Service Based Consent to Authorise the membership of the Worker in the Client Service Team and Service Based Access to Authorise the Service to access related Client information (see Figure 26).

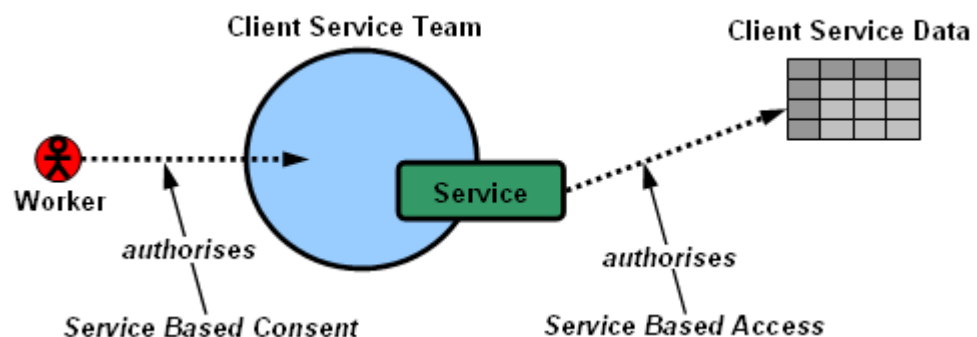


Figure 26: Service Based Privacy

#### 4.6.5 C#31: Access by Authorisation

From: DR#4: Client Based Permission, C#30: Service Based Privacy

<sup>40</sup> The paper “Using a Client-Task Based Approach to Achieve a Privacy Compliant Access Control System” (de la Motte & Hartnett 2008) discusses the issue of using workflow tasks to achieve Privacy Compliance.

<sup>41</sup> The connection of Service Based Privacy with privacy laws is discussed in the paper “Implementation Considerations in the Framing of Privacy Laws” (de la Motte 2007).

To: C#49: Client Authorisation

*The **Access by Authorisation** principle requires that Client permission to access Service Information is communicated to the CMS as an Authorisation Request from or on behalf of the Client.*<sup>42</sup>

The inputting of the Authorisation Request to consent to a Service (that is, accept the Service Contract) is the action that the CMS deems as “permission to access related Service Information”.

## 4.7 Consent

### 4.7.1 C#32: Consent Type Specification

From: DR#44: Written Client Consent, DR#45: Digital Client Consent, DR#46: Verbal Client Consent, DR#47: Implied Client Consent, DR#49: Agent Client Consent

To: C#33: Service Based Consent

***Consent Type Specification** requires that the consent type(s) for each Service must be specified together with any conditions, so that what constitutes legitimate consent is clear.*

Consent types include but are not limited to written, digital, verbal, implied and agent consent. Conditions may include specifying if a form needs to be used, the kinds of digital Consent that are appropriate and what constitutes implied consent in the circumstances.

Consent specifications are made so that Managers and Workers are clear on the requirements for legitimate Consent in their work environment. Making Consent Management part of standard work processes aims to minimise litigation against the Organisation for breaches of Consent requirements and increase Client satisfaction by including them more in decision making processes.

---

<sup>42</sup> Service Information is contained in the Service Description.

#### 4.7.2 C#33: Service Based Consent

From: DR#54: Service Consent, C#5: Client-Services, C#32: Consent Type Specification, C#37: Service Based Information Provision

To: C#30: Service Based Privacy, C#34: Bundled Client Consent, C#36: Consent by Authorisation

*The **Service Based Consent** principle requires that all Client Consent in a Service Based Organisation applies directly to the Services that the Organisation performs for the Client.*

Except in specified circumstances, such as when a patient is unconscious, each Client-Service must be performed with the prior Consent of the Client. All Client Consent required in the Organisation is based on the Services performed; no other Consent is needed.

#### 4.7.3 C#34: Bundled Client Consent

From: DR#47: Implied Client Consent, C#33: Service Based Consent

To: C#36: Consent by Authorisation

***Bundled Client Consent** occurs when a single Consent acceptance is given by the Client to cover more than one Client-Service.*

Client Consent may be for a single Service or a group of related Services; and consent for a particular Service may imply Consent for other Services. Whenever the Consent given covers more than one Client-Service, Bundled Client Consent describes the multiple nature of the Consent.

#### 4.7.4 C#35: Implied Permissions

From: DR#47: Implied Client Consent

To: C#49: Client Authorisation

***Implied Permissions** are the permissions to perform actions that are given when the Client gives Consent to receive a Service.*

Implied Permissions are applied when a single Consent triggers a number of actions. The single Consent construes a permission to perform each of these actions. Implied Permissions for a Service include permissions to perform the Service and any Sub-Services and *bundled* Services, to access and write/modify any related Client information, and to charge the agreed fee for the Service.

#### **4.7.5 C#36: Consent by Authorisation**

From: C#33: Service Based Consent, C#34: Bundled Client Consent

To: C#49: Client Authorisation

*The **Consent by Authorisation** principle requires that Consent for a Service is communicated to the CMS as an Authorisation Request from or on behalf of the Client.*

The inputting of the Authorisation Request to Consent to a Service (that is, accept the Service Contract) is the action that the CMS deems as “permission to perform the Service”.

#### **4.7.6 C#37: Service Based Information Provision**

From: DR#5: Client Information Reviewal, DR#48: Informed Client Consent, DR#50: Informing Agents, C#22: Compliant System

To: C#33: Service Based Consent

***Service Based Information Provision** requires that Information relating to the performance of a Service is provided to the Client to fulfil informed consent requirements and allow Client review.*

There are two sets of Service Based Information that must be provided to the Client: *Descriptive Service Information* and *Personal Service Information*.



Descriptive Service Information is the information that details the Service and its possible implications for, and effect on, the Client. This information must be provided to the Client as part of ensuring that consent is *informed*.<sup>43</sup>

Personal Service Information is the information that is entered or modified as part of Service Delivery. The Client, under this definition, has the right to review this information in order to be informed and to have any errors rectified. The Client Review process is facilitated by an optional “Review Service” that is available to the Client. This optional Service is associated with all Client-Services.

## 4.8 Accountability

### 4.8.1 C#38: Service Based Accountability

From: DR#39: Business Accountability

To: C#39: Client-Service Accounting, C#41: Client-Service Records

*Service Based Accountability requires that all accounting be done in terms of Services.*

Accountability deals with the recording of required work related information. In a Service Based Organisation, as all work is defined as Services, all work related information can be attributed to specific Service instances. Service Based Accountability records required information for each Client-Service.

The work related information required for a particular Service will vary according to the applicable regulations and Organisational policies. Generally, it will detail “*who* did *what* and *when*, *where* and *why* they did it” and the financial accounting information associated with the Client-Service.

---

<sup>43</sup> There is an onus on the Service provider to ensure that the information given and the possible personal implications have been understood by the Client.

#### 4.8.2 C#39: Client-Service Accounting

From: C#5: Client-Services, C#38: Service Based Accountability

To: C#8: Client-Service Management

*In Client-Service Accounting each Client-Service is one accounting item.*

Client-Service Accounting is one part of Service Based Accountability. It deals with the financial transactions (essentially *income* and *expenditure*) that take place in an Organisation. *Income* is from the Client-Services provided in the form of payments received from Clients for the Services they receive. *Expenditure* is the cost of the Services that are paid to others, and includes wages and resources costs.

In Client-Service Accounting, income and expenditure amounts are attributed to specific Client-Services. Client-Services are the units of accounting, that is, all items appearing in the accounts are considered to be Client-Services.<sup>44</sup>

#### 4.8.3 C#40: Required Auditing

From: DR#40: Business Auditing

To: C#41: Client-Service Records

*Required Auditing meets Compliance requirements by ensuring that Client-Services are adequately audited.*

Auditing is a standard business process that assures that accounting and Services meet Compliance requirements.

#### 4.8.4 C#41: Client-Service Records

From: C#5: Client-Services, C#7: Service Based Information, C#38: Service Based Accountability, C#40: Required Auditing

---

<sup>44</sup> *Products, resources and wages* can all be described in terms as Client-Services, remembering that the Client can also be within the Organisation.

To: C#42: Service Based Auditing

*Client-Service Records for all Client-Services are to be collected and retained for a least the required time.*

Interactions with all Client-Services are monitored and saved so that particulars of past Service Authorisations and Service Deliveries can be recovered if contentions arise, and required auditing procedures can be facilitated.

#### **4.8.5 C#42: Service Based Auditing**

From: C#41: Client-Service Records

To: C#8: Client-Service Management

*In Service Based Auditing Client-Services are the items that are audited with the aim of assuring that all Organisational Services are Compliant.*

Service Based Audits investigate a number of aspects of Services, including allocation of roles and Client-Services to Workers and Managers, Client-Service charges, access auditing to assure privacy and monitoring of usage patterns.

### **4.9 Authorisation**

#### **4.9.1 C#43: Mandatory Authorisation**

From: C#22: Compliant System

To: C#50: Service Based Authorisation

*Mandatory Authorisation dictates that all Client-Services must be Authorised by the CMS.*

It is the CMS's task to ensure that all Client-Services are properly Authorised. This entails certifying that all types of Authorisation Requests are processed through to a satisfactory completion.

Mandatory Authorisation is one of the main principles that ensure CMS Compliance.

#### 4.9.2 C#44: Authorisation Type Specification

From: DR#24: Associate Authorisation, DR#25: Emergency Authorisation, DR#26: Critical Authorisation, DR#27: Authorisation Timing, DR#28: Authorisation Order

To: C#45: Managerial Authorisation, C#48: Worker Authorisation

*Authorisation Type Specification* requires that the valid Authorisation Type(s) (ex ante, uno tempore or ex post)<sup>45</sup> of all Authorisation Requests associated with a Client-Service be specified in the Service Description.<sup>46</sup>

The specification of Authorisation Type enables the CMS to manage and certify each Authorisation. This ultimately contributes to CMS Compliance.

#### 4.9.3 C#45: Managerial Authorisation

From: DR#52: Managerial Offer, C#44: Authorisation Type Specification

To: C#50: Service Based Authorisation

*Managerial Authorisations* require that Managers Authorise each Client-Services on behalf of the Organisation.

The Managers in the Organisation are responsible for deciding which Organisational Services are to be offered to which Clients. This means that at some point in the Organisation decision making process every Client-Service it offers must be approved (Authorised). It could be decided that a particular Service be offered to anyone that desires the Service, while another Service should only be offered on a case-by-case basis or anywhere between these two extremes.

---

<sup>45</sup> Ex ante authorisations are given prior to Service Delivery, uno tempore authorisations are given at the time the Service is delivered, and ex post authorisations are given retrospectively.

<sup>46</sup> Service Descriptions exist in the SOA methodology and the same terminology is used in the Model. CMS Service Descriptions are stored digitally and constitute a part of Service Contracts.

Accordingly, for a particular Service, the Managerial Authorisation may cover anywhere from a single to every Client-Service instance. Where a Managerial Authorisation covers more than one Client-Service instance, the CMS will automatically recognise the Managerial Authorisation for each Client-Service covered by the given Authorisation.

#### 4.9.4 C#46: Client-Service Allocation

From: DR#37: Task Allocation, DR#38: Personnel Management, C#5: Client-Services, C#23: Worker-Service Contracts

To: C#47: Worker Acceptance

*Client-Service Allocation occurs when a Manager Authorises a Worker(s) to perform a Service(s) for a Client(s).*

Allocation of work is a primary managerial function. In a Service Based Organisation there are two common types of work allocation: allocating Worker(s) to Client(s), or allocating Client-Services (that is, particular tasks) to Worker(s). In practice, therefore, a single allocation may involve multiple Workers, multiple Services and/or multiple Clients. For example, a Team of Workers may be allocated to perform a particular Service to a particular Client, a Worker may be allocated to service (multiple Services) a Group of Clients, or a Team may be allocated to service a Group of Clients.

Whatever the allocation, the CMS will always break these allocations down to multiple Worker-Client-Service Allocations in order to manage the work.

Client-Service Allocation is the method used to facilitate Managerial Authorisation for Client-Services (see Figure 27). In Legal terms a Client-Service Allocation involves two contractual offers/requests – one offering the Service to the Client *and* another requesting the Worker to perform the Service as part of their contract of employment. From a Business Management perspective the Client-Service Allocation is the central mechanism for managing business tasks (Services). From an IT perspective a Client-Service Allocation enables the required access permissions to automatically be put in place.

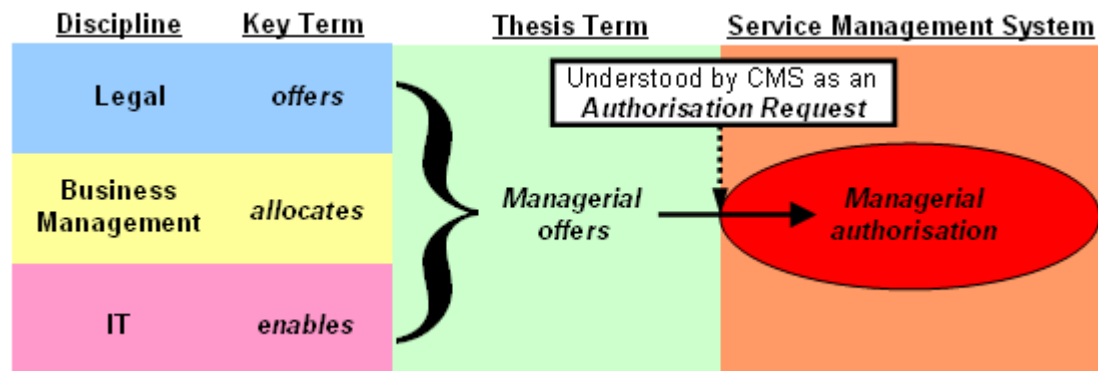


Figure 27: Key Terms for Managerial Service Offers

#### 4.9.5 C#47: Worker Acceptance

From: C#46: Client-Service Allocation

To: C#48: Worker Authorisation

*Worker Acceptance* requires that Workers must accept the Client-Services that they are allocated.

Most of the time a Worker will unquestioningly perform the Client-Services that are allocated to them. However, there are both legitimate and illegitimate (from the Organisation's point of view) reasons why a Worker may refuse to perform an allocated Client-Service, for example, they may be too busy, they may not wish to deal with a particular Client, or they may morally object to performing the Service. In any case, even if a reason is considered illegitimate, a Worker cannot (or should not) be forced to perform a job if they refuse to do it. Provision must therefore be made for Workers to provide their acceptance or refusal.<sup>47</sup>

#### 4.9.6 C#48: Worker Authorisation

From: DR#18: Cooperative Practices, C#44: Authorisation Type Specification, C#47: Worker Acceptance

To: C#50: Service Based Authorisation

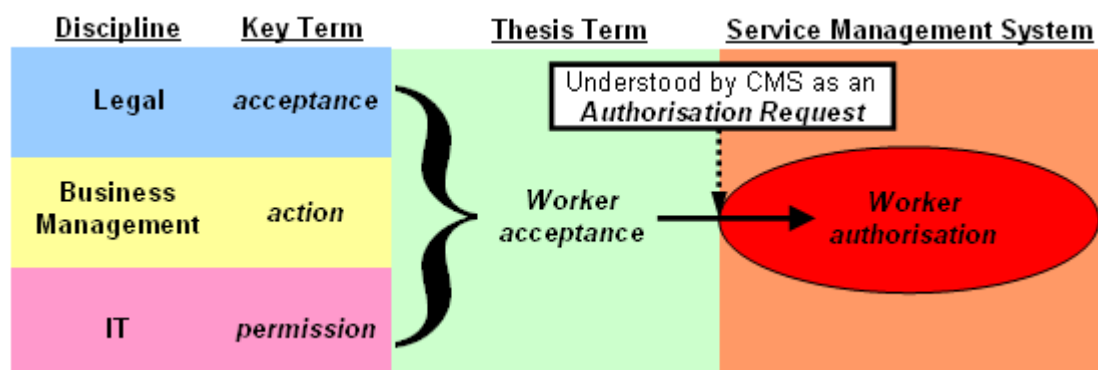
<sup>47</sup> Refusals may affect the Organisation's willingness to employ a Worker but that is a separate issue to the management of particular Client-Services.

***Worker Authorisation*** requires that *Workers must Authorise each Client-Service in order to indicate their acceptance to deliver the Service on the Organisation's behalf.*

An Authorisation Request must be received by the CMS to confirm that the Worker will perform the specified Client-Service. The generation of the Authorisation Request will normally be automated; it will be triggered by a Service related *action* that the Worker performs, for example, when the Worker interacts with the Service interface to begin delivering the Service.

Alternatively, the System may assume that the Worker accepts the Service unless they perform an *opt-out action*, such as making themselves unavailable to perform the Service. It is conceivable that different approaches would be appropriate in different circumstances, depending on factors such as the availability of alternative Workers and the frequency of refusals.

Worker Acceptance is facilitated through Worker Authorisation for Client-Services (see Figure 28). In Legal terms a Worker Acceptance involves *acceptance* of the managerial request to perform the Service as part of the contract of employment. From a Business Management perspective the Worker Acceptance involves an *action* to proceed with performing the Client-Service (or refusing). From an IT perspective Worker Acceptance utilises the required access *permissions* that were put in place through Client-Service Allocation.



**Figure 28: Key Terms for Worker Acceptance of Services**

#### 4.9.7 C#49: Client Authorisation

From: C#26: Validation by Authorisation, C#31: Access by Authorisation, C#35: Implied Permissions, C#36: Consent by Authorisation

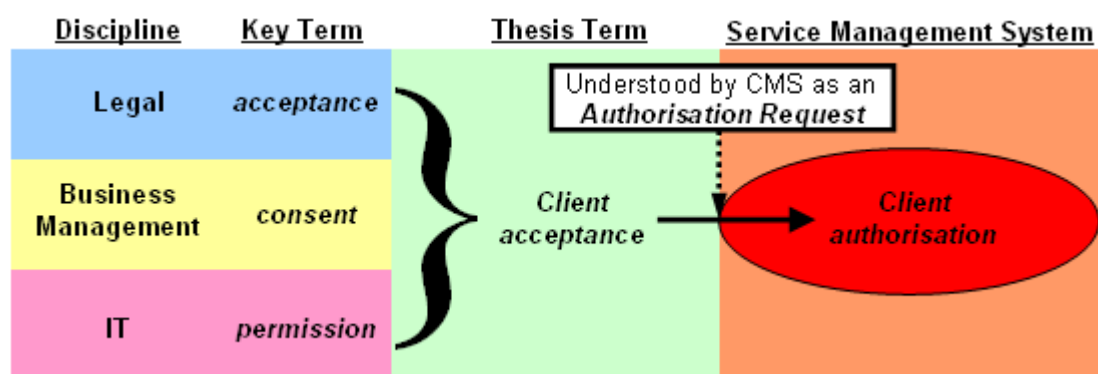
To: C#50: Service Based Authorisation

***Client Authorisation** requires that Clients must Authorise each Client-Service they are to receive to indicate their consent/acceptance of the Service.*

An Authorisation Request must be received by the CMS to confirm that the Client has *accepted* the specified Client-Service.

While *acceptance* is a legal contract related term, it is used here in a broader sense to include *acceptance* of the Service Contract, *consent* to receive the Service and *permission* to access and write/modify Client Information.

This interpretation of *acceptance* is a key research outcome of the dissertation because it is the concept that ties together the three research disciplines of Law, Business Management and IT (see Figure 29). In this regard, **obtaining acceptance (of a contract) is primarily a Legal issue**, **obtaining consent (to receive the Service) is primarily a Business Management issue** and **obtaining permission (to access private information) is primarily an IT issue**.<sup>48</sup>



**Figure 29: Key Terms for Client Acceptance of Services**

<sup>48</sup> While *consent* and *permission* have legal implications, they are the terms primarily used in the *Business Management* and *IT* disciplines respectively. The *Legal* term *acceptance* best encapsulates the three related concepts.



#### 4.9.8 C#50: Service Based Authorisation

From: C#43: Mandatory Authorisation, C#45: Managerial Authorisation, C#48: Worker Authorisation, C#49: Client Authorisation

To: C#51: Sub-Service Authorisation, C#52: Authorisation Management

***Service Based Authorisation** mandates that Managerial, Worker and Client Authorisations are received before a Client-Service is performed.*

Authorisation Requests must be received by the CMS to confirm that a Manager has *offered* the Client-Service, and the Worker and Client have both *accepted* the Client-Service (see Figure 30).

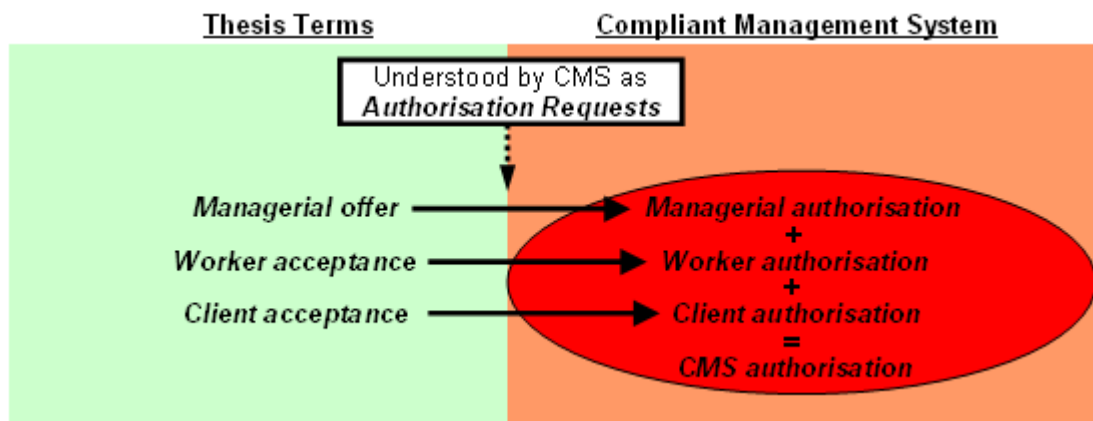


Figure 30: Service Based Authorisation

#### 4.9.9 C#51: Sub-Service Authorisation

From: C#50: Service Based Authorisation

To: C#52: Authorisation Management

***Sub-Service Authorisation** implies that when a Service is Authorised all the Sub-Services are automatically Authorised.*

Services are hierarchical in that any Service can be a Sub-Service of another Service. By definition, the Authorisation of a Service implies Authorisation for all Sub-Services.

#### 4.9.10 C#52: Authorisation Management

From: DR#19: Multiple Authorisations, DR#20: Guaranteed Access, C#3: Serviceflow Dependencies, C#50: Service Based Authorisation, C#51: Sub-Service Authorisation

To: C#8: Client-Service Management

*Authorisation Management requires the CMS to manage Service Authorisations to ensure that all Services are performed with proper Authorisations in place.*

Ensuring proper Service Authorisations is a primary task for the CMS, as it is a key to facilitating System Compliance. Authorisation Management involves the CMS both *passively* receiving Authorisation Requests as and *actively* pursuing other Authorisation Requests.

### 4.10 Model Concepts Table

Table 2 lists the Model Concepts that were developed to meet the Design Requirements. The table serves to collate the Concepts for reference purposes. The background colours used for each General Concept correspond to those in the Model Concepts Diagram (Figure 3) and the Model Concepts Relationship Diagram (Figure 22).

General Concept	No.	Model Concept
Services	1	Global Services
	2	Service Based Organisation
	3	Serviceflow Dependencies
	4	Service Cells
	5	Client Services
	6	Service Profiles
	7	Service Based Information
	8	Client-Service Management
	9	Service Based Security
Groups	10	Service Teams
	11	Contextual Groups
	12	Group Based Organisation
	13	Group Based Access
	14	Global Groups
	15	Global Group Addressing
	16	Group Based Services
	17	Group Based Information Storage
	18	Set Based Groups
	19	Partial Permissions
	20	Group Based Management
Service Contracts	21	Compulsory Compliance
	22	Compliant System
	23	Worker-Service Contracts
	24	Client-Service Contracts
	25	Contract Based Services
	26	Validation by Authorisation
Privacy	27	The “5W’s” of Access Control
	28	Service Based Purpose
	29	Service Based Access Principle
	30	Service Based Privacy
	31	Access by Authorisation
Consent	32	Consent Type Specification
	33	Service Based Consent
	34	Bundled Client Consent
	35	Implied Permissions
	36	Consent by Authorisation
	37	Service Based Information Provision
Accountability	38	Service Based Accountability
	39	Client-Service Accounting
	40	Required Auditing
	41	Client-Service Records
	42	Service Based Auditing
Authorisation	43	Mandatory Authorisation
	44	Authorisation Type Specification
	45	Managerial Authorisation
	46	Client-Service Allocation
	47	Worker Acceptance
	48	Worker Authorisation
	49	Client Authorisation
	50	Service Based Authorisation
	51	Sub-Service Authorisation
	52	Authorisation Management

Table 2: Model Concepts

## 4.11 Conclusion

The Model Concepts Relationship Diagram (Figure 22) was introduced at the outset of the chapter. This is the main diagram in the dissertation as it shows all the relationships between Research Fields, Design Requirements and Concepts.

The 52 Model Concepts were then described in turn, categorised according to their related General Concept. They were collated in Table 2.

The following chapter details the Model and focuses on the *mechanisms* that allow Services to be managed through Groups and Service Contracts and controlled by Authorisations from Managers, Workers and Clients.

## Chapter 5

# Model Description

---

### 5.1 Introduction

This Chapter provides a high level description of the Model that is aimed at a broad audience. It is built on the Model Concepts outlined in the previous Chapter. While the chapter *is* technical in nature – the Model, after all, describes a technical (IT) system – the focus is on describing the necessary mechanisms in a comprehensible way rather than a specific solution (that is, a computer program and related data model).

The Chapter begins by discussing the incorporation of the Model Concepts and defining the main software Components of the Model. The CMS and methods employed for managing Services are then described. This is followed by sections on Group Management and Contract Management. A Client-Service example is then provided which ties together the previous topics. A final section contains a table that details how each Model Concept is incorporated into the Model.

### 5.2 Incorporating Model Concepts

52 Model Concepts were described in the previous chapter. In order to explain the Model a smaller set of seven Model Concepts, called *Foundational Model Concepts* (FMCs), is used. The FMCs can be seen as the seven most important Model Concepts. As a group they embody the core functionalities of the Model.

The seven FMCs are further broken down into two new Concepts called *Model Design Concepts* (see 5.2.2). These two Concepts embody the essence of the FMCs

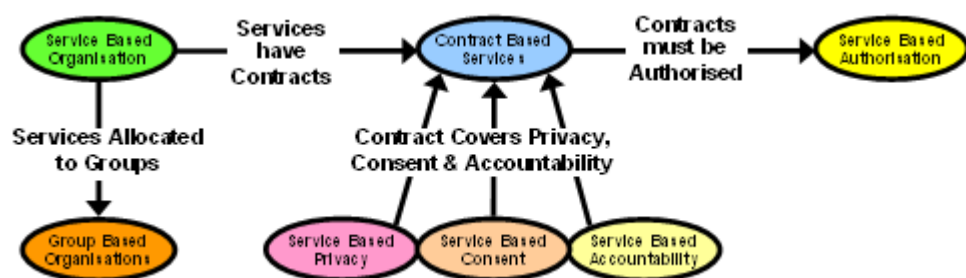
and provide the link between the “theoretical” Model Concepts and the “practical” IT Components that form the basis of the Model.

### 5.2.1 Foundational Model Concepts

Each of the Seven General Concepts has an associated Foundational Model Concept (see Figure 3). The Foundational Model Concepts (FMCs) that form the basis of the Model are:

1. Service Based Organisation (C#2),
2. Group Based Organisation (C#12),
3. Contract Based Services (C#25),
4. Service Based Privacy (C#30),
5. Service Based Consent (C#33),
6. Service Based Accountability (C#38), and
7. Service Based Authorisation (C#50).

Understanding the relationships between the FMCs is the key to understanding the Model and how real Organisations are represented (modelled) in the CMS. Figure 31 shows the relationships and flows from left to right.



**Figure 31: Foundational Model Concepts**

First, Organisations are described in terms of Services (which define “what they do”) and Human Resource Groups (which define “who can do”). Personnel Managers specify (generally) “who *can* do what” by allocating Services to Groups, and

Operational Managers specify (specifically) “who *will* do what” by allocating Client-Services to Workers.

Secondly, each Client-Service has its own Service Contract that specifies Privacy, Consent and Accountability requirements for Service Delivery. The Service Delivery process for each Service (that is, the prescribed Work Tasks) includes “Contractual Tasks” which constitute the Offering (by the Organisation) and Acceptance (by the Client) of the Service Contract.

Thirdly, parties communicate their performance of *Contractual Tasks* to the CMS through Authorisation Requests. The CMS monitors and records reception of the Authorisation Requests and ensures that all necessary Requests are received before Service Delivery tasks are commenced or completed (depending on the requirements).

### 5.2.2 Model Design Concepts

The Model utilises two *Model Design Concepts*:

1. The *Service Based Organisation Design Concept* which specifies that:

*A **Service Based Organisation** is composed of Services, Groups and Service Contracts*

2. The *Service Based Work Design Concept* which specifies that:

***Service Based Work** is composed of Client-Services where Groups do Contracted Services for Clients.*

A Service is what a Service Based Organisation *offers*; a Client-Service is what it *does*. A Client-Service is one instance of the delivery of a Service. The sum of all the Client-Services performed constitutes all the work that a Service Based Organisation does.

The two Design Concepts provide the basis for implementing the seven Foundational Model Concepts (FMCs) outlined at the end of the previous chapter. The FMCs in turn provide the basis for implementing all the other Model Concepts.

### 5.3 Model Components

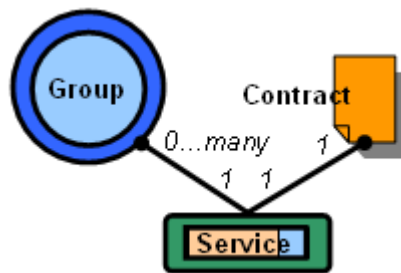
Services, Groups and Service Contracts are the three *Model Component Types* (or Component Types). Collections of Components of the three types make up an Organisation (as shown in Figures 32 and 33, below). The Component Types are defined as follows:

**Services are the units of work** performed by the Organisation.

**Groups represent Organisation entities/resources.**

**Service Contracts define work expectations** for the Organisation, its personnel and the Client.

#### 5.3.1 Component Relationships



**Figure 32: Basic Model Component Relationships**

Figure 32 shows the relationships between the three Model Component Types. The *relationships* between Component Types are shown by the lines between them, with the dotted end indicating that that Component Type is associated with the Component Type at the non-dotted end. This means that Services are associated with Groups and Service Contracts, not the converse.

The relationship between Services and Groups is *1 to 0...many*, meaning that each Service is allocated to 0, 1 or more Groups. The case where a Service is allocated to “0” Groups is where the Service exists but is not being offered or used. Services that are being offered are allocated to “1 or more” Groups.

The relationship between Services and Service Contracts (“Contracts”) is *1 to 1*, meaning that each Service is associated with exactly “1” Contract.



The association between Services and Groups embodies the Group Based Services FMC. Group members (including Sub-Group members) can perform the Services that are associated with the Group. This association relationship controls *who* (which Groups) is allowed to do *what* (which Services). In other words, it controls which Groups do which Services.

The Contract Based Services FMC (see Figure 31) is embodied in the association relationship between Services and Service Contracts. Services can only be performed in accordance with the associated Service Contract. This association relationship defines the expectations of Service Delivery.

The combination of the Service-to-Group and Service-to-Contract associations defines how **Groups do Contracted Services**.

### 5.3.2 Organisation Composition

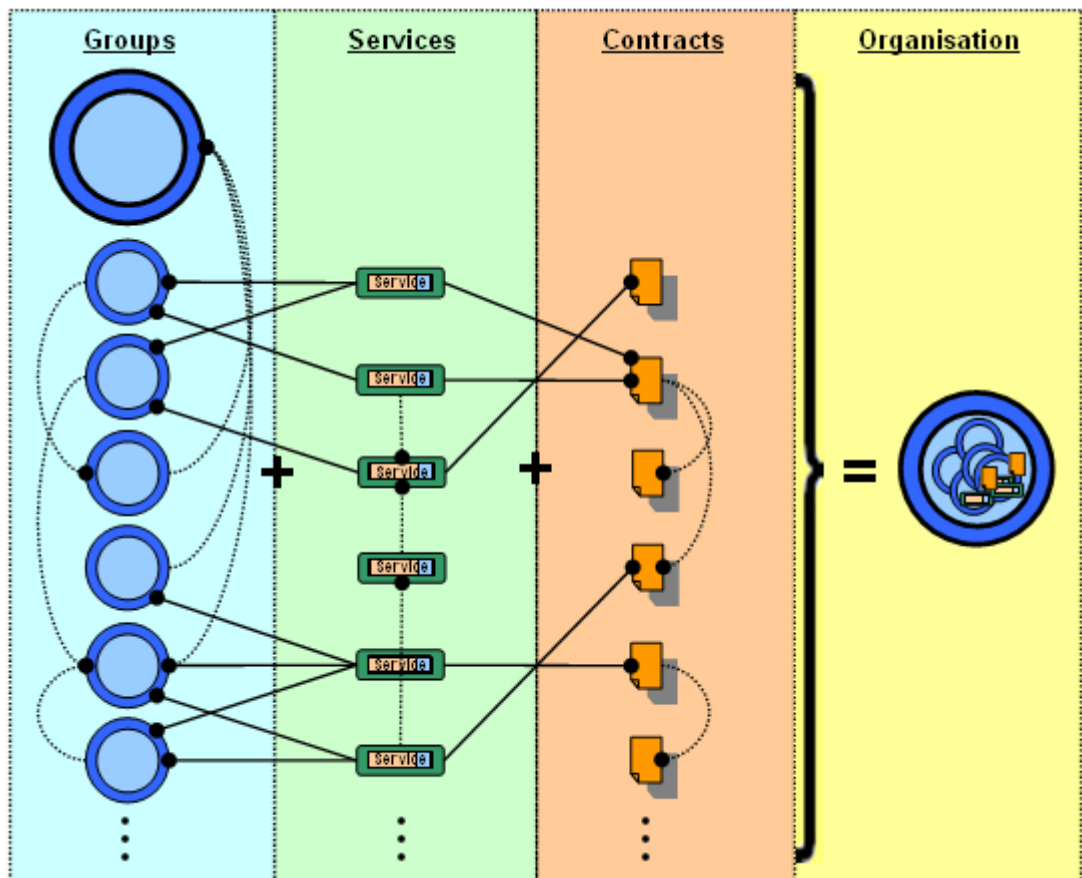


Figure 33: Organisation Components

Figure 33 shows how an Organisation is composed of many Groups, Services and Contracts, and the relationships between them. The solid lines show the Basic Component Relationships between Services and Groups, and Services and Contracts (the type shown in Figure 32). The dotted lines reveal the relationships between Components of the same type: Group/Sub-Group relationships, Service/Sub-Service relationships and Contract/Sub-Contract relationships.

The “Organisation Icon” in the “Organisation” column represents the entire Organisation. The Organisation itself is defined by a single Global Group (the Organisation is a global entity), the *Organisation Group*, represented by the large icon at the top of the “Group” column. The Organisation Icon shows the Sub-Groups of the Organisation inside the Organisation Group together with the associated Services and Contracts.

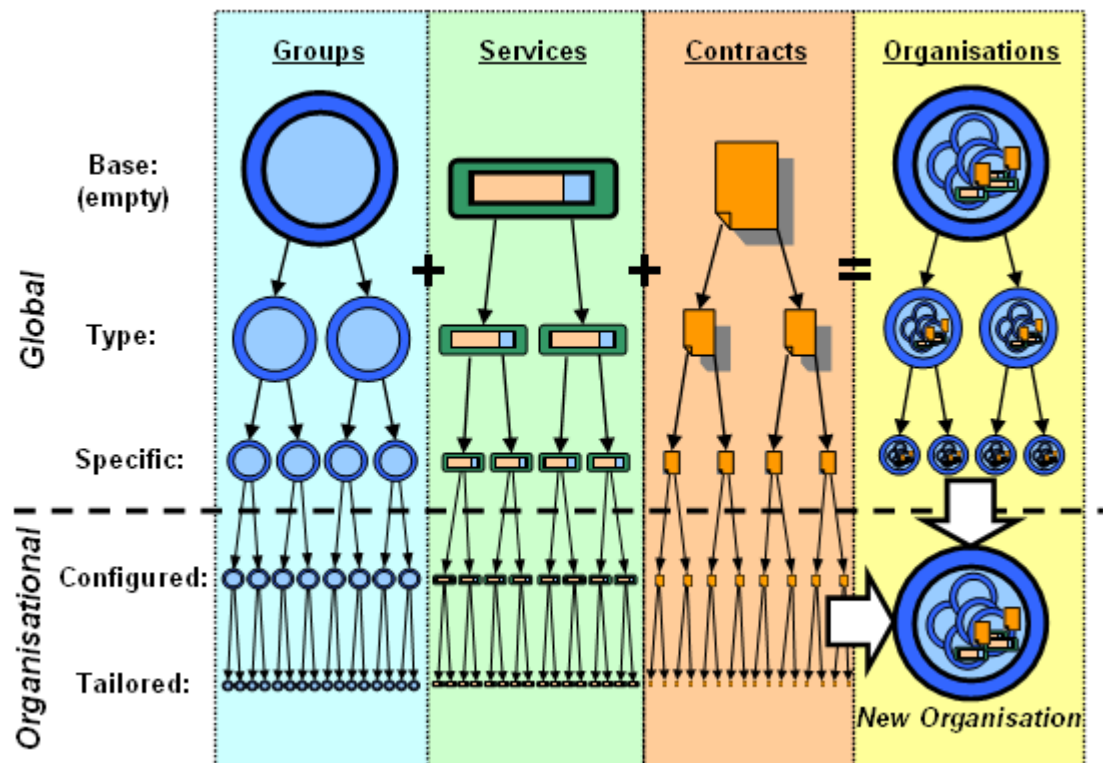
### 5.3.3 Component Origin

The Model is based on the idea of reusing Components. This means having *Global Components* that can be copied and modified as necessary to meet local requirements. The concept of Global Groups was discussed at 1.6.3 and 4.4.5, but Services and Contracts can also be global; for instance, consider the need for global standards for online businesses.

In view of this, the Model employs three hierarchical structures, one for Services, one for Groups and one for Service Contracts (as shown in Figure 34, below). At the top level of each hierarchy are *Base Components* for a *Base Service*, a *Base Group* and a *Base Service Contract*. These Base Components are the Model Components which would be at the level below the “Object” Class in Object Oriented Programming (OOP) terms. There is one Base Component for each Component Type.

Each of the three *Component Hierarchies* has five Level Types (“Base”, “Type”, “Specific”, “Configured” and “Tailored”). The top three levels contain Components that are stored globally, while the bottom two levels represent Components for individual Organisations. The diagram is symbolic only as its purpose is to portray

multiple levels. In practice, each Component can have *0...many* children and each level can have *0...many* sub-levels.



**Figure 34: Component Hierarchies**

The fundamental idea is to have repositories of Components stored on the “Global” level that can be used (copied) by individual Organisations and configured and tailored to meet their individual requirements. It is also possible for Organisations to *upload* their Components into the repositories so that they can be used by others. This facilitates the evolution of templates, organisational structures and business management techniques.

As well as individual Group, Service and Contract Components, whole *Organisational Templates* can be created and stored, as shown in Figure 34. These can be copied to provide New Organisations with ready-made “Start-up Systems”. For example, a new florist business could copy an Organisational Template for a typical florist containing appropriate Services, Groups and Contracts. The Organisational Template can then be modified (configured and/or tailored) to meet individual business needs.

The collection of imported and modified Components and the relationships between them form the foundation of an Organisation's System. The running of the Organisation, which involves the creation and management of Client-Services, is built on this foundation.

#### 5.3.4 Component Instances

The System that implements the Model, the CMS, is essentially composed of two types of elements, Components and Component Instances.<sup>49</sup> The Components (and the relationships between them) represent the management structure of the Organisation, and the state of the Component Instances represents the state of the Organisation.

The Components are imported and modified as necessary when the Organisation is created. They can be added to and modified over time as the management structure evolves. Their state, however, is relatively "static".

Component Instances represent the state of the work that the Organisation is performing. This is in constant flux and is thus "dynamic" in nature.

An example of a Component is a "House Painting Service". An Organisation can offer this Service to customers, who when they accept the Service become *Clients* of the Organisation. Now for John to get his house painted by the Organisation he must firstly be offered and then accept a House Painting Service (that is, accepting a contract for the provision of the Service by the Organisation). On acceptance, the CMS will generate a new instance of a House Painting Service. This is a Component Instance. Every time a customer accepts a House Painting Service a new Component Instance is generated. The CMS manages each of these instances and, at any particular time, each instance will be at a particular stage of completion. So, in terms of Components and Component Instances, there is exactly one House Painting Service Component and many Component Instances. This is typical.

---

<sup>49</sup> In this context the first level of Component Type is represented by the Base Components of each type.

The CMS uses a Component as a *template* for the generation of Component Instances.<sup>50</sup> More specifically, a Service is a template for making Client-Services, a Group is a template for making Client-Service Groups and a Service Contract is a template for making Client-Service Contracts.

In technical IT (OOP) terms, Components are like Object Classes, Component Instances like Objects (that are instances of an Object Classes), *Component Instance Data* like Object Attribute values and Service Tasks like Object Methods. As Object Classes are hierarchical, so too are Components. A CMS can therefore easily be programmed using an OOP language like Java.

## 5.4 The Compliant Management System

The CMS is the heart of the Business Management software that an Organisation uses.

Users of the System interact with it through interfaces which, in most cases, are contained in devices with screens displaying Graphical User Interfaces (GUIs) and keyboard or touch-screen input functionalities (for example, desktop computers, laptop computers, tablet computers, PDAs and mobile phones).<sup>51</sup>

The CMS includes all the Services, Groups and Contracts Components that are needed for the Organisation. Some of these are *System Components*. The System Components together comprise the *Compliant Management System Operating System (CMSOS)*. The CMSOS is comprised of three sets of Service Components: *Model Services*, *View Services* and *Controller Services*.<sup>52</sup> Essentially the Model Services are the data storage components, the View Services are the user interface components and the Controller Services are the Service functionality components.

---

<sup>50</sup> A *template* is used to make multiple copies of an item, like a cake recipe is used to make multiple cakes.

<sup>51</sup> There is no limit to the types of input and output devices that the system can employ. Services can be used that are designed to interact with any digital device.

<sup>52</sup> “Model-View-Controller” (MVC) is a commonly used methodology for designing software. Here it is used to *describe* the system rather than being the *prescribed* method of software design.

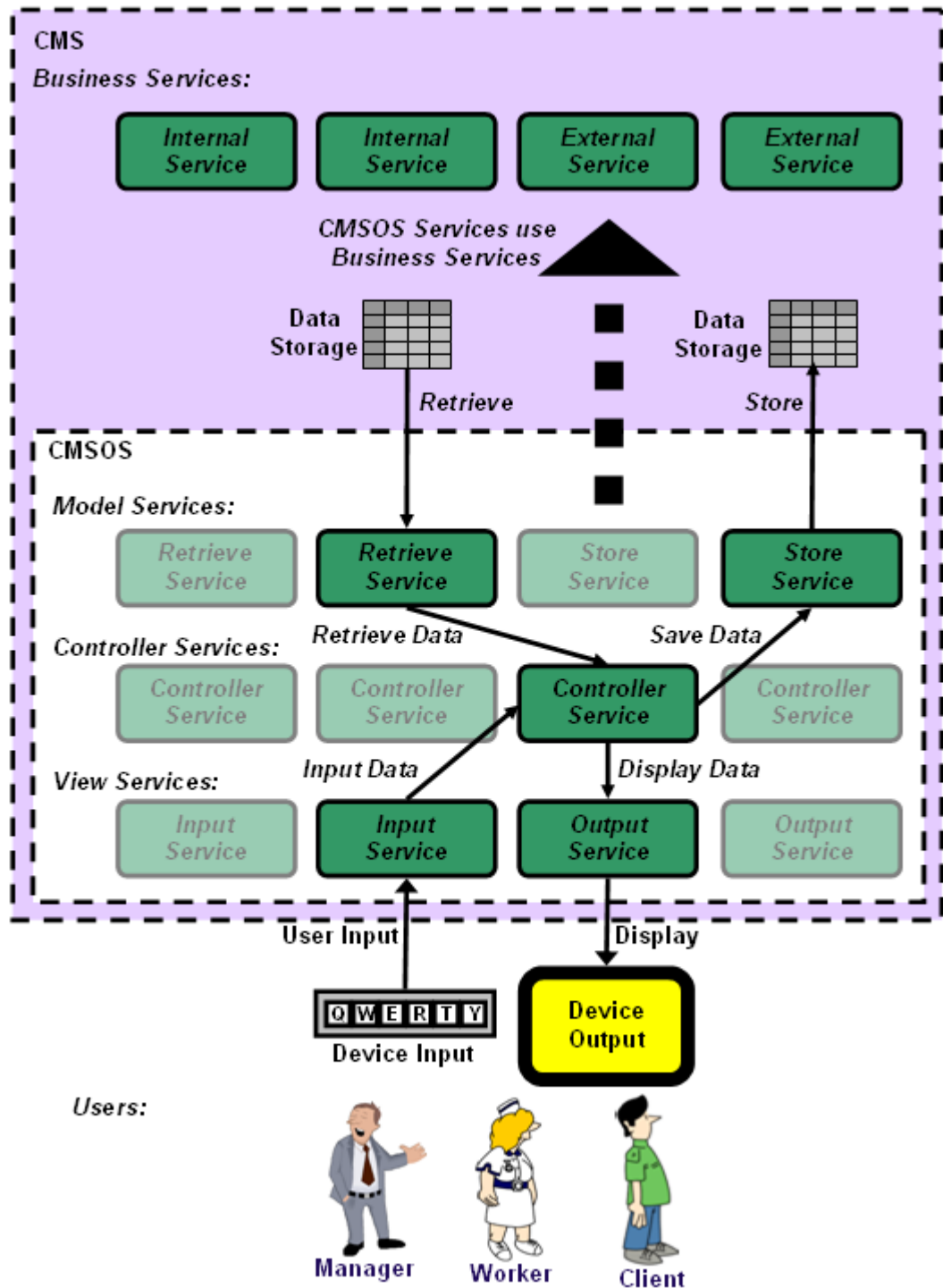


Figure 35: CMS Operating System (CMSOS)

Figure 35 shows the CMS from a Services perspective. The three sets of CMSOS Service Components are shown inside the inner dashed rectangle. The highlighted Services show how one combination of Service Components can function together. The *Controller Service* does operations on data received from the *Retrieve Service* and/or *Input Service* and sends resultant data for storage to the *Store Service* or for

display to the *Output Service*. The CMSOS employs appropriate Input and Output Services for each device and appropriate Retrieve and Store Services for each data storage type. Retrieve and Store Services maintain data Privacy by employing Service Based Privacy.

Services in the CMS, other than System Components, are *Business Services*. These represent *Internal Services* that are used to manage the Organisation as well as *External Services* which represent the Services that the Organisation offers to its customers. The main difference between Internal and External Services is that the Client is within the Organisation for Internal Services and outside the Organisation for External Services.

The diagram also shows that CMSOS Services *use* Business Services. In fact, Business Services are run as Sub-Services to the CMSOS Services. This is akin to computer operating systems (such as Microsoft Windows) running applications (such as Microsoft Word).

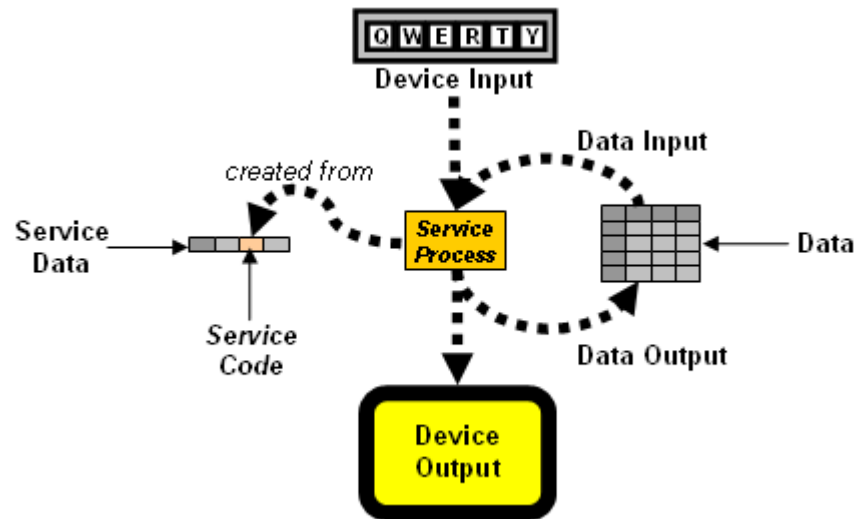
#### 5.4.1 CMS Programs and Data

It was explained at 1.3.2 that computers store *data* that includes *program code*. To perform work a *process* is created from the *program code*. The process loads *input data* from memory or input device, processes it and outputs the *resultant data* to memory or output device.

Here *program code* is termed *Service Code* and process termed *Service Process* because, of the three Component Types, only Services have *program code* and generate *processes*.<sup>53</sup> The Service Code is stored as part of *Service Data*. Figure 36 shows how a Service Process is created from the Service Code of a particular Service, and how it uses and produces *data*.

---

<sup>53</sup> Services are “dynamic” in that they represent things (business processes) that are happening and changing, whereas Groups and Service Contracts are (relatively) “static” in that their state rarely change (in terms of computer time). In other words, of the three components, Services *do* all the changes while Groups and Service Contracts just sit there and wait to be changed (by a Service).



**Figure 36: CMS Service Process**

Figure 37, below, shows the entire content of the CMS. All that is stored is *CMS Data*. The CMS Data is made up of six *data collections*, with a *Component Data* collection and a *Component Instance Data* collection for each of the three Component Types. Each row (termed *Data Set*) of the tables shown represents a single *Component* or *Component Instance*; each Component and Component Instance, therefore, has its own Data Set.<sup>54</sup> The only program code that the CMS contains is stored in each Service Data item.<sup>55</sup>

A Component Instance is created when an Authorisation Request for its creation is accepted by the CMS. The Component Instance Data is created and initialised (usually with some default values from the Component Data) at this point. The Component Instance Data is the only thing that the System has that represents the Component Instance, so from the System's point of view, the Component Instance Data **is** the Component Instance. This is analogous to a system seeing its users as their aliases (for example, their *usernames*).<sup>56</sup>

An example of a Client-Service Contract in the form of a Standard Paper Contract clarifies the difference between Component Data and Component Instance Data. The paper contract is the *Contract Data* (which is Component Data) and the

<sup>54</sup> While storing data in tables in a database is common, it is not the only storage method available.

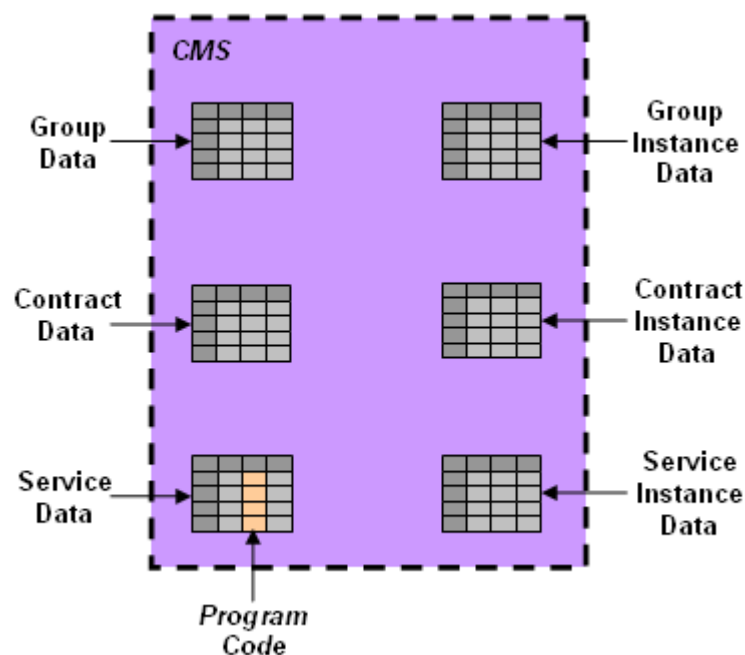
<sup>55</sup> Alternatively, pointers to a separate Program Code storage repository can also be employed.

<sup>56</sup> In the Model the user aliases are Base-level User Groups (BUGs).



information that is supplied by the participants is the *Contract Instance Data* (which is Component Instance Data).<sup>57</sup> The Contract Instance Data represents the state of the particular contract, while the Contract Data is the same for all contracts of the type.

In logical terms, each Component has Component Data associated with it and each Component Instance has Component Instance Data associated with it. But in physical terms the *data* is normally stored in some form of collective data repository (for example, computer databases). A logical address is an address related to a model whereas a physical address is a real physical location. For example, the street address of a house is a logical address whereas its physical address is its global location (perhaps expressed by global coordinates). The physical address of data is given by its location in computer memory.



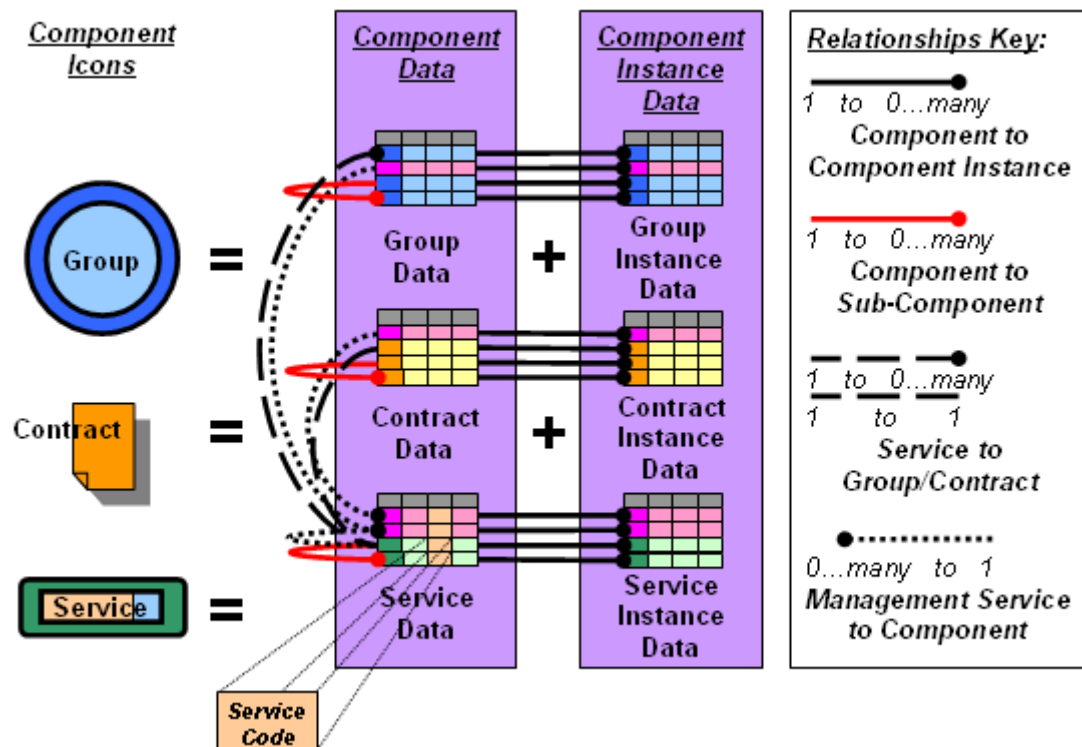
**Figure 37: CMS Contents**

The CMS Data contains all the information that represents the business structure of the Organisation and all the data that represents the current (and past) state of the Organisation. The data includes the Service Code and information that defines all the relationships between Components, other Components and Component Instances.

<sup>57</sup> Normally paper contracts would be printed from a digital version, even if the digital version was scanned from an original paper document.

Figure 38 shows the relationships (expressed as *pointers* to related *data sets*) that are stored in each Component and Component Instance Data Set.<sup>58</sup> The *Component Icons* appear in the left column. They are simply a graphical representation of the Components that are used for explanation purposes. The actual Components and Component Instances are the Data Sets.

The relationships are depicted by the lines connecting the various Data Sets.<sup>59</sup> The “Relationship Key” on the right briefly summarises the meanings of the lines.



**Figure 38: Data and Component Relationships**

There are four types of relationships: 1) Component to Component Instance; 2) Component to Sub-Component; 3) Service to Group/Contract; and 4) Management Service to Component.

Component to Component Instance relationships are those between Components and the Component Instances that are generated from them, for example, Service to

<sup>58</sup> A *pointer* is a data item which is an address – the memory location of related data. Where necessary, data associated with *pointers* (*pointer data*) is used to detail the nature of relationships.

<sup>59</sup> The lines connecting the Component Data collections (in the left mauve box) are analogous to the relationships shown between Organisation Components in Figure 33.

Client-Service. The relationships are *1 to 0...many* as, for each Component, zero to many Component Instances can be generated.

Component to Sub-Component relationships are the hierarchical relationships between Components and their own Sub-Components, for example, Service to Sub-Service and Group to Sub-Group. Again the relationships are *1 to 0...many* as each Component can have zero to many Sub-Components.

Service to Group and Service to Contract relationships represent the allocation of Services to Groups (a *1 to 0...many* relationship) and the association of a Service to its Service Contract (a *1 to 1* relationship).<sup>60</sup>

Management Service to Component relationships define how Services, Groups and Contracts are managed. Each Service, Group and Service Contract can have zero to many Management Services acting on them. Management Services have the effect of changing the state of Components and Component Instances.

#### 5.4.2 Global Data Storage

Cloud Computing (introduced at 1.3.4.4) is one of the key methodologies that the Model incorporates. In IT terms this essentially means that, subject to restrictions, data and programs can be *stored anywhere* and *accessed from anywhere* in the global network.

Data and program code are both stored in computer memory. As a house has a global street address, locations in computer memory can have a *Global Memory Address (GMA)*. When a process requires data, it can obtain it from any computer memory in the world as long as it has access through a network. Similarly, output data can be sent to any accessible global location. This means that the processor (hardware in a computer chip) that runs a process does not have to be in the same location as the data. The processor only needs to know the GMA of the data (or input/output device). This occurs when persons interact with a webpage; they are accessing remote processes and data.

---

<sup>60</sup> Services can be attached to more than one group or can be partially attached to a set of Groups, where multiple Group memberships are required for Service performance.

The main challenges in using remote processes and data are finding them and being authorised to access them. While some processes and data may be made available to all, many are prescribed some form of protection for security and privacy reasons.

In the Model, Groups have Global Group Addresses (GGAs) and data is accessed through Services that are associated with Groups and Service Contracts. In other words, all data (Component Data and Component Instance Data) is logically connected to at least one Group which has a GGA. This enables a *Global Access System (GAS)* based on the Model.

### 5.4.3 A Global Access System example

Consider the problem of a tourist presenting unconscious at a hospital, with a surgical scar on his head and his name and home address, and the doctor needing to find and be authorised to access the patient's latest brain scan, which is stored in an unknown foreign hospital's database. It is an emergency. Finding the information is vital and problematic, and accessing it requires proper authorisation.

In the Model a person has a *Global Base-level User Group (G-BUG)* that is associated all their existing aliases (*O-BUGs*) in the Organisations they deal with, and Organisations belong to Group hierarchies based of their type (in this case medical/hospital). As shown in Figure 39, below, each O-BUG is connected to their Service Team (in this case called a Patient Care Team). All the Services the client received are associated with their Service Team and all the Service Instance Data (including the Brain Scan) is associated with the relevant Service. The steps to access the Brain Scan are:

1. Locate the patient's G-BUG by searching the "Registered Patient" Group using the name and address,
2. Find his medical practice and hospital O-BUGs using the G-BUG associations,
3. Find all his relevant "Patient Care Teams" using the O-BUG associations,

4. Locate all his Client-Services relating to “Brain Scan”, and
5. Gain Authorisation to access the Brain Scan.

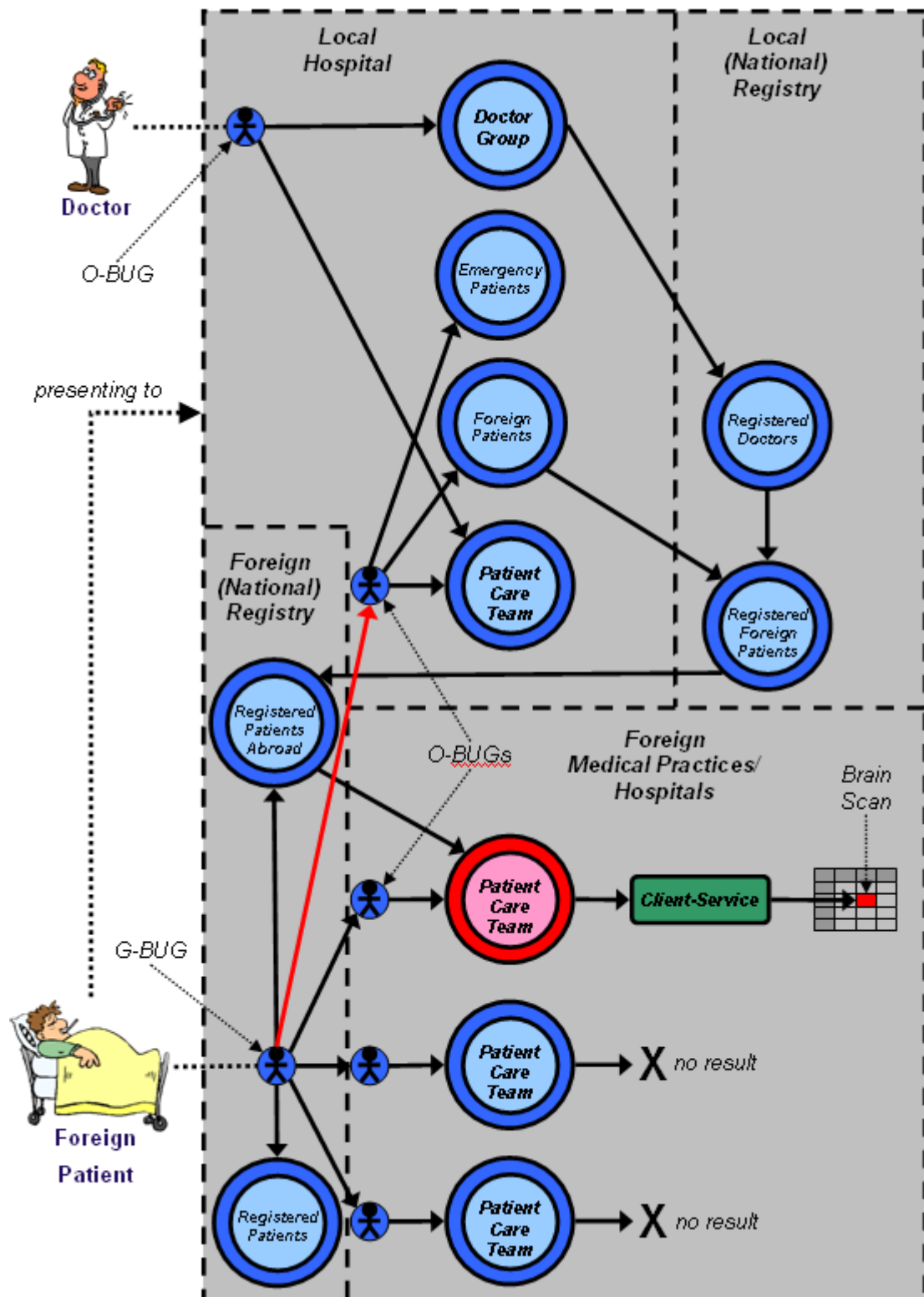


Figure 39: Global Search and Authorisation Example

To gain Authorisation to access the Brain Scan, the Doctor must be a member of the patient's Foreign Hospital Patient Care Team (the red one in the diagram). To become a member of this team the Doctor must achieve the status of being a registered foreign doctor who is approved to treat the patient without their consent. This is achieved through the Registries (for Practitioners and Patients) in each country. A patient placed in the "Foreign Patients" Group in the Local Hospital is automatically placed in the "Registered Foreign Patients" Group in the Local Registry. This Group membership includes "treatment information" confirming the local hospital, treating Doctor and emergency status. This information is automatically transferred to the "Registered Patients Abroad" Group in the Foreign Registry.<sup>61</sup> The Patient's membership of this Group, together with the treatment information that establishes the Patient-Doctor relationship, enables the Doctor to become a temporary member of the patient's Foreign Hospital Patient Care Team, thus enabling access to the Brain Scan.

The point here is that all the necessary steps can be done *automatically* and *within seconds* with the input of the patient's *name* and *address* (by an administrator or other personnel) and his *unconscious* and *emergency* state (by the Doctor). In order to ensure the legitimacy of the access (with regard to privacy and security), the Authorisation will be automatically reviewed and confirmed as legitimate (or otherwise) by an appropriate person. This is an instance of an *ex post* Authorisation (with *separation of duties*) being used in an emergency where information provision takes priority over information privacy.

#### 5.4.4 Managing Components

The CMS does its work by managing individual Components and Component Instances on a lower level and managing all the Organisational work on a higher level. The following three sections detail the basic management mechanisms, one section for each Model Component.

---

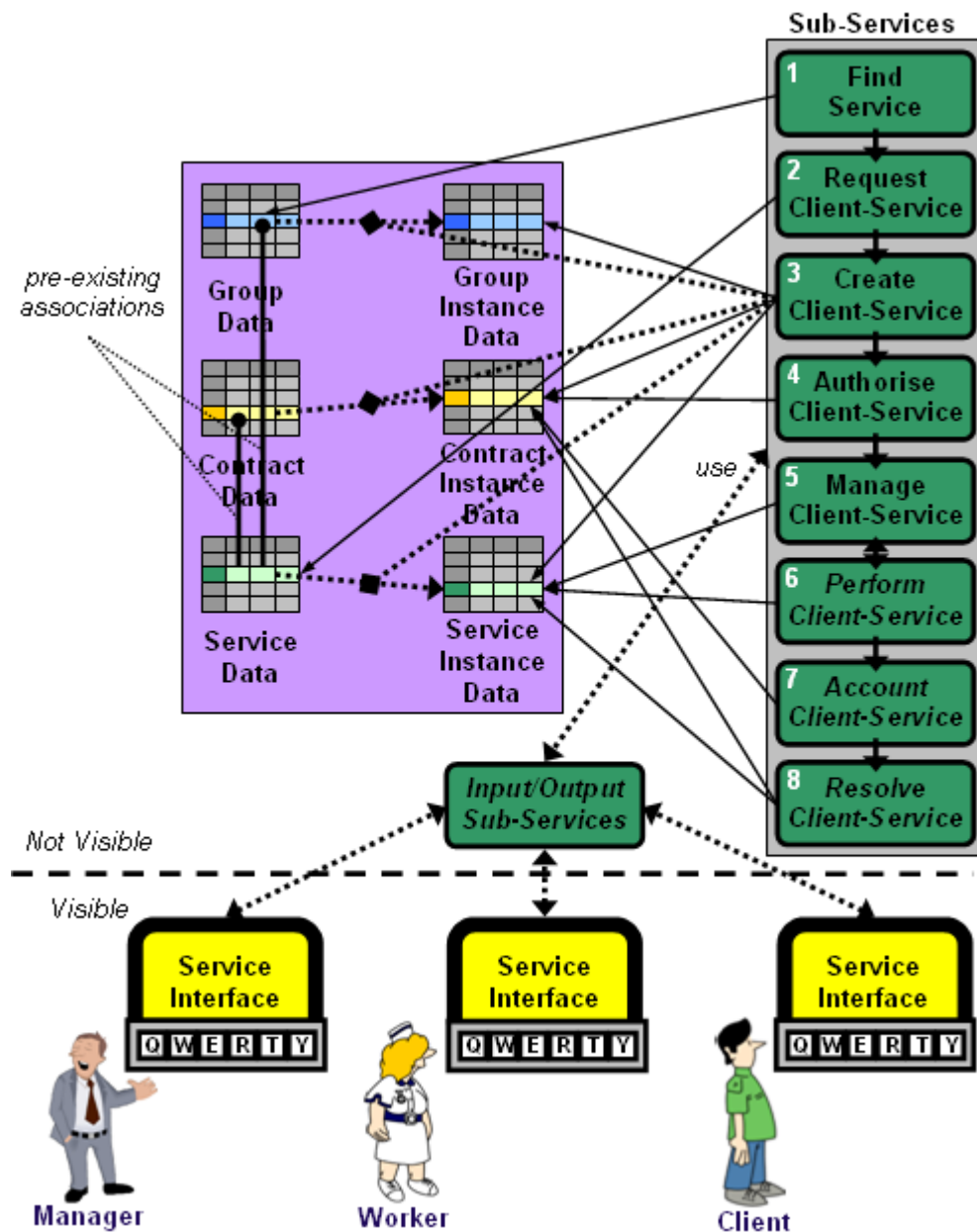
<sup>61</sup> The patient's G-BUG is found in the Foreign Patient Registry from the name and address and then the patient is automatically placed in the "Registered Patients Abroad" Group and the G-BUG linked to the Local Hospital O-BUG (the red arrow in Figure 39). This link facilitates later access in the foreign country to the current local treatment information.

At this stage, it is useful to reiterate that Service Management is essentially a Business Management issue that focuses on what Managers do; that Group Management is essentially an IT issue that focuses on what System Administrators do (or have in the past done); and that Contract Management is essentially a Legal issue that focuses on what personnel must do to ensure Compliance.

## 5.5 Service Management

Service basics were covered at 1.5, where Figures 4, 5 and 6 were used to explain Service Components, relationships and terminology.

The CMSOS is responsible for the management of *Service Instances*, also called Client-Services. Depending on the circumstances, a Manager, a Worker or the Client can request that a particular Service be performed. When the Service is requested, a Client-Service is created. Once the Service has been Authorised, it can be performed and finalised. Figure 40 demonstrates the *Service Management Process*.



**Figure 40: Service Management Process**

The Service Management Process is itself a Service with its own Service Data and Service Instance Data (not shown). It has eight sequential Sub-Services (shown on the right in Figure 40). The Component Data collections and Component Instance Data collections are shown in the mauve box. The arrows between the Sub-Services and the data collections show the data sets that are used for each Sub-Service. When the Client-Service is created in Step #3, new data sets (rows) are created for it in the *Service Instance*, *Group Instance* and *Contract Instance Data* collections. The subsequent steps use these data sets.



The participants (the Manager, Worker and Client) interact with *Service Interfaces* if and when necessary. They may use the same or separate interfaces depending on the circumstances. Often the Client's interaction may be performed by the Worker in communication with the Client, for example, the Worker may verbally tell the Client some information, or may (when permitted) verbally ask the Client for consent and then enter the consent themselves.

### 5.5.1 The Service Management Process

The Service Management Process is shown in Figure 40. The eight steps in the process are defined by the eight Sub-Services in the diagram. Details of each step are as follows:

1. **Find Service:** A participant uses a search process, for example, a keyword search or a hierarchical category search, to find the Service.
2. **Request Client-Service:** The participant (requestor) requests the Client-Service.
3. **Create Client-Service:** The CMS creates a new Client-Service instance, sets all the default values and obtains any other necessary details from the requestor.
4. **Authorise Client-Service:** The CMS determines the participants and obtains the necessary Authorisation Requests (as specified in the Service Description) from them.
5. **Manage Client-Service:** The CMS keeps track of the Service Delivery process and instigates specified actions to progress the Service Delivery as necessary.
6. **Perform Client-Service:** The participants perform the specified actions (that is, *deliver* the Client-Service) and

completion of actions (or non-completion) is communicated to the CMS.

7. **Account Client-Service:** The CMS records the necessary accounting information and manages the Client billing and fee collection processes.
8. **Resolve Client-Service:** Any dispute is managed by the CMS to resolution.

The Authorisation Process in step #4 is critical to ensure compliance and involves the establishment of the Client-Service Contract. Contract Management is discussed at 5.7.

Steps #5 and #6 involve the *management* and *delivery* of the Client-Service. These may involve a number of actions and iterations and occur in tandem.

Some actions are “standard actions” in that they are common to all Client-Services (for example, accounting for the Service charge), while some are “specific actions” that depend on the particular Service (for example, the administration of a specific drug) and yet others are “individual actions” that are unique to the Client-Service (for example, John’s temperature must be taken at 10.30am).

The delivery of most Client-Services will generally proceed according to plan and no unusual steps will be required. However, on occasions when there are mechanical breakdowns, personnel absences, workload issues and the like, alternative arrangements need to be found. In these cases the CMS, in its Service Management role, must methodically (and logically) go through alternatives or get someone (a human) to make a decision when required or when alternatives are exhausted. This is akin to the autopilot on an aircraft disengaging or being overridden when a dangerous or unforeseen situation arises.

### 5.5.2 High Level Service Management

The Service Management Process described so far relates to the process that is undertaken for each individual Client-Service, that is, it relates to how particular

Client-Services are controlled. A higher level of Service Management exists where decisions about multiple Client-Services must be considered and delivery priorities need to be established. On this level, the delivery of a Client-Service depends on other Client-Services, for example, a Doctor has many Clients to attend to and some priority must be established. Another example is patients presenting at an Accident and Emergency Department in a hospital. Priority here must take into account the severity of the patient's condition as well as the time the patient has been waiting.

The CMS manages such situations, and indeed all situations, by utilizing appropriate Management Services. The Management Service to be used in a particular situation will usually be a standard Management Service that employs a standard algorithm (for example, "first-come-first-served") to determine priority. Such a Service can be copied, configured and tailored in the same way as any Service.

More complex algorithms, which take into account multiple factors, can of course be employed. Emergency Departments may function best with a complex algorithm, but once the algorithm is proven it should be able to be reused by other Emergency Departments. Such reuse is the type of standardization envisaged here.

As well as managing Services and their delivery, the CMS also manages Groups and Contracts. The succeeding two sections deal with these in turn. Managing Groups and Contracts has to do with controlling the *state* of individual Groups and Contracts. As *state* is represented by *data*, state changes involve data changes. In the Model *data* is not altered directly but only through the operation of Services. *Group Management Services* and *Contract Management Services* exist for the purpose of changing Group and Contract states.

### **5.5.3 Management of Personal Work**

Managers and Workers ("personnel") interact with the System through Service Interfaces. The interface shows the personnel the Client-Services that they are personally required to perform and information about the Client-Services, and allows them to manage what they do. Personnel must also make Authorisation decisions about Client-Services. The interface provides prompts (for example, a pop-up box

on the screen) where Authorisation decisions can be made (for example, by choosing from a list of options and clicking “OK”).

## 5.6 Group Management

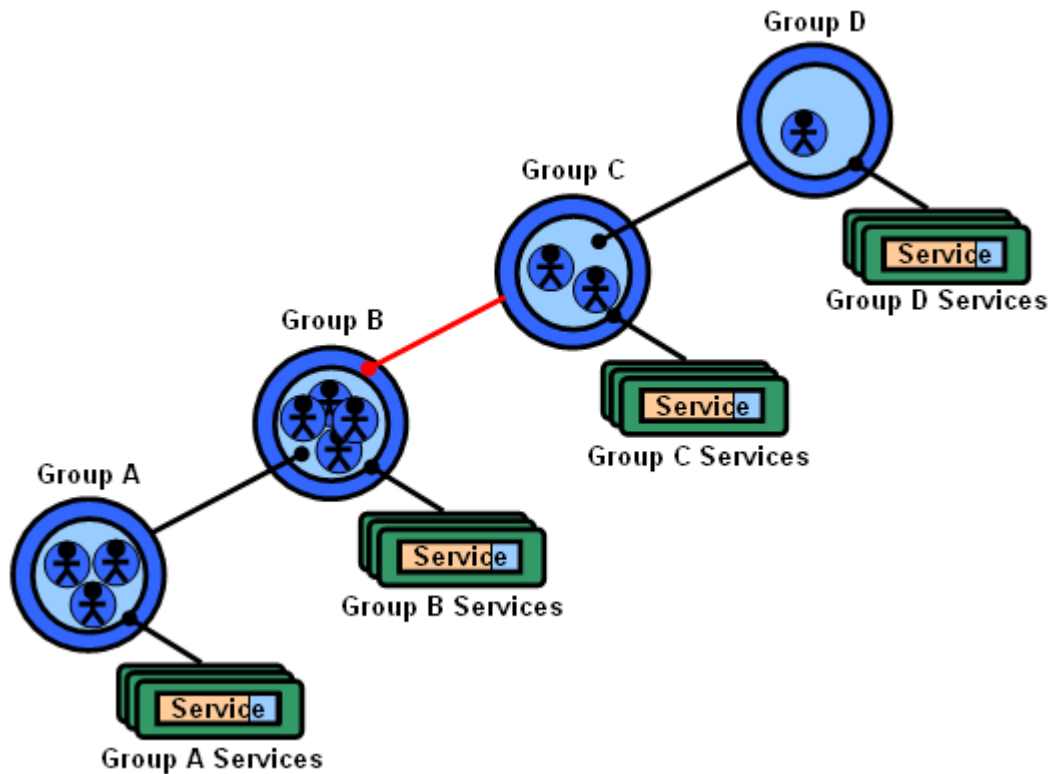
In order to perform a Service, personnel (Managers and Workers) must belong to a Group that is Authorised to perform the Service; in other words the Service must be associated with the Group. *Group Management* is therefore the key to managing *who* is Authorised to do *what*. Group Management is enabled through Group Management Services.

Group basics were covered at 1.6, where Figure 7 was used to define Group Structure and terminology.

Groups represent collections of things in an Organisation, for example, collections of Workers or collections of computers, or collections of drugs in a medicine cabinet. Groups are either “active”, where Group members can perform the Group’s Services, or “passive”, where Group members (being resources) are merely used or consumed in the performance of a Service. For example, a Group of Workers is “active” while a Group of drugs is “passive”.

The Model defines two types of relationships between Groups: the “member of” (or Sub-Group) relationship that allows the members of a Group to also be the members of another Group, and the “manager of” (or Management Group) relationship that allows the members of a Group to perform Management Services on another Group.

The ability for higher level Groups to perform lower level duties in an Organisation are represented by Sub-Group relationships, while supervisory responsibilities are represented by Management Group relationships. These relationships, in connection with appropriate Services, enable the Group Structure to fully represent the managerial structure of an Organisation.



**Figure 41: Sub-Group and Management Group Relationships**

Figure 41 shows the relationships between role Groups and associated Services. The members of each Group can perform the Services attached to the Group. Group A members can perform Group A Services, Group B members can perform Group B Services, etc. The Sub-Group relationship between the higher level Group A (which could be a “Doctor” Group) and the lower level Group B (which could be a “Nurse” Group) means that Group A members can also perform Group B Services. The Management Group relationship between Group C (which could be the “In-Charge Nurse” Group) and Group B enables Group C members to perform Group C Services (which would be Group Management Services) on Group B. The Sub-Group relationship between Group D (which could be the “Nursing Manager” Group) and Group C enables the members of Group D to also perform Group C Services.

### 5.6.1 Group Management Services

*Group Management Services* (GM Services) are Services that are responsible for changing the state of a Group – that is their purpose. In fact, they are the only Services that are permitted to change the state of a Group. If other Services need to change the state of a Group, they must use GM Services, as Sub-Services, to do the

job. GM Services therefore act as Transformation Procedures (TPs) in the Clark-Wilson Model (see 3.5), enabling data integrity to be preserved.

By definition, every Group has its own *Management Group Set (MGS)* that contains at least one Management Group. The Management Groups in the MGS are said to *belong* to the Group. The sole purpose of Management Groups is to *manage* the Group they belong to. The only Services associated with Management Groups are therefore GM Services. The GM Services associated with a Management Group only act on the Group that the Management Group belongs to.

Having said that every Group has its own MGS, it needs to be clarified that an Organisation can have Management Teams (that is, Groups of Managers) that can be responsible for managing Groups (that is, managing zero or more Groups). In this case the mechanism employed is that the Management Teams are made Sub-Groups of Management Groups belonging to the Groups being managed by the Team. These relationships are depicted in Groups C and D in Figure 41.

## 5.7 Contract Management

*Contract Management* has to do with ensuring Compliance. This is achieved by using standard compliant practices and by checking to ensure that human error and neglect is minimised, chiefly through building in a monitoring process and by ensuring participants are informed of Service and contractual requirements.

Standardisation and simplicity are important goals of the Model. This particularly applies to Contracts because the personnel delivering an Organisation's Services are parties (or at least agents of the Organisation) in these Contracts. The easier it is for them to understand their contractual responsibilities, the more likely it is that they perform them in a compliant manner. Personnel are trained in their own specialities and in the heat of a busy work environment can tend to focus on Service Delivery tasks without much consideration of detailed contractual obligations. By providing standardised contracts and Contract Management procedures rather than a plethora of different contracts and procedures, personnel can more effectively meet contractual obligations.

Each Service has a Contract. This fulfils the Contract Based Service FMC. Service Contracts fulfil the Service Based Privacy, Service Based Consent and Service Based Accountability FMCs.

Standard Contracts that can be copied, configured and tailored are utilised in the Model. This leads to there being some “standard obligations” that are common to all Client-Service Contracts (for example, explaining the implications of Client Consent), while there are “specific obligations” that depend on the particular Service (for example, the need to explain the possible side effects of medication) and “individual obligations” that are unique to the Client-Service (for example, explaining the relationship of a medication to the other medications the patient is taking).

### **5.7.1 Service Authorisations**

The System manages each Client-Service Contract. It does this by requiring Authorisations. This fulfils the Service Based Authorisation FMC.

Authorisations are received as Authorisation Requests by the CMS. The CMS processes the requests and tags them as *confirmed*, *denied* or *conditional* (requiring some further action being undertaken). The System *manages* Authorisations and ensures that Informed Consent is obtained from each participant before (usually) the Client-Service is commenced.

To facilitate emergencies and other unpredictable events, some flexibility is built into the System. The amount of flexibility depends on the service-based factors that determine how much security is necessary. Some trust is placed on individuals but the System looks for evidence of systematic misuse and provides warnings and alerts where necessary. There is a fundamental balance between flexibility and security where risks need to be considered. The basic security premise is that where flexibility is given, monitoring is in place. *Uno tempore* and *ex post* Authorisations are the mechanisms for giving flexibility.

Before an Authorisation is requested, the System will seek to ensure that the participants are all properly informed about what Consent entails and their

Contractual obligations. The aim is to make sure that the Contract is enforceable and that the Organisation fulfils its Contractual obligations through its personnel's actions.

The System manages the Organisation's Contractual obligations by providing necessary information and by requiring personnel to sign off on contract-related tasks (that is, to tell the System that they have fulfilled particular obligations). It does this by either directly making information available to the participants, by reminding personnel to provide the information to the Client on the System's behalf (that is, the Organisation's behalf), by reminding personnel of Service Delivery requirements, and by checking where practicable that tasks has been carried out.

### **5.7.2 Contract Management Services**

*Contract Management Services* (CM Services) are provided by the System to do the required tasks. As with GM Services, CM Services are responsible for changing Contract Data and are used as Sub-Services by other Services to do so.

CM Services can be classed as *Primary CM Services* and *Secondary CM Services*. Primary CM Services are *Authorisation Management Services* (AM Services) that are utilized for every Client-Service, bearing in mind that Authorisation is the mechanism that the Model uses to manage Service Contracts, and that Service Contracts encompass Consent, Privacy and Accountability requirements.

AM Services are responsible for providing, for each Client-Service, the *Contract Offer* to the Client and for gaining the *Contract Acceptance* from the Client. Managers, in their role as work allocators, are responsible for making the Contract Offer to the Client on behalf of the Organisation. In most cases, however, it will be a Worker who presents the Contract Offer to the Client and obtains their Contract Acceptance. Managers and Workers, in the process of doing their jobs, provide the required Authorisations to the System through AM Services.

What constitutes an appropriate Contract Offer and Contract Acceptance depends on the nature of the Client-Service. Contract Data specifies the appropriate requirements for each Client-Service and the System accordingly puts appropriate



AM Services in place to meet the specified requirements. An example of specified requirements is that, for a surgical procedure, a Contract Offer requires provision of information on the procedure, possible effects, privacy implications, and cost to the Patient; Contract Acceptance requires that the System be informed that the Patient has received and understood the required information and signed a Written Contract (in the medical context, commonly a Consent Form) to indicate their acceptance.

Secondary CM Services are essentially *Administration Services* that are triggered when Authorisations are received by the System. For example, when a Contract is accepted by the Client, the necessary *Accounting Services* are triggered. These Services update the Organisation's financial records and deal with Patient billing, payment and the like.

## 5.8 Client-Service Example

A *Service Request* for a Client-Service triggers the creation of a new Client-Service instance. Figure 42, below, demonstrates the Components involved in the System's management of a Client-Service and the main processes involved.

The Group structure, except for the Client-Service Team, is in place before the Client-Service is requested. The personnel on the right have been assigned to the Patient Care Team. The Group Membership relationships show the interactions between different Groups.

The Service in the *ONE SERVICE* box represents a particular Service that has previously been assigned to the "Nurse" Group. It is this Service that has been requested. Though it is not specified in the diagram, either the Manager (who could be a Nurse Supervisor), the Nurse or the Patient could have requested the Service. Once the request is received by the System, the Client-Service, Client-Service Team and the Client-Service Contract are created. These are each Component Instances that are generated from Components (of which only the Service Component is shown). These actions are covered in the first three steps (Sub-Services) of the Service Management Process (see 5.5.1 and Figure 40).

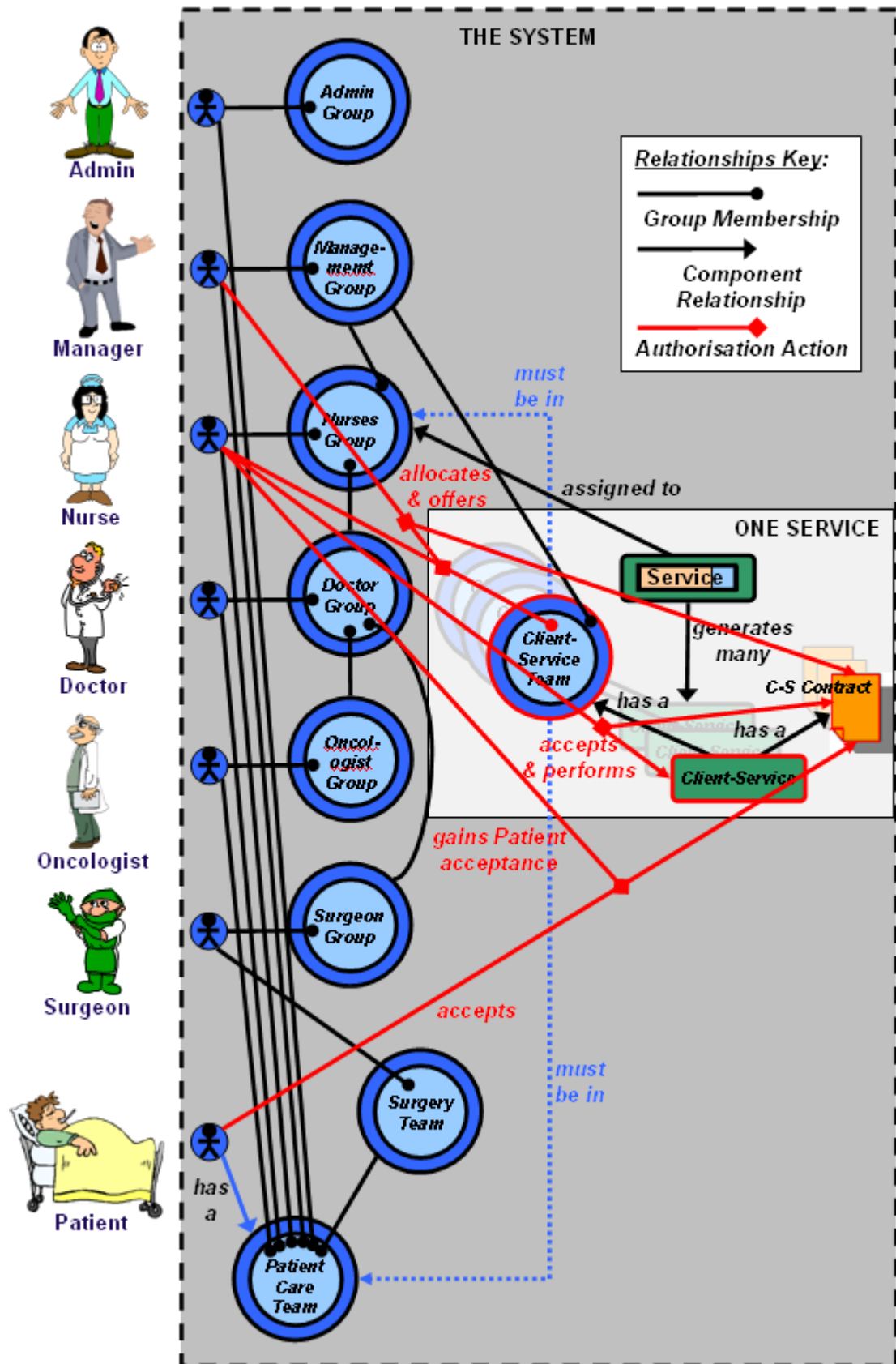


Figure 42: Client-Service Example

One of the main purposes of this example is to detail the *Authorisation Client-Service Process* (the *Authorisation Process*), being Step #4 in the Service Management Process. In the example the Manager initiates the Authorisation Process, which is typical, but in other cases the Worker or the Client (Patient) can initiate the process. The Authorisation Process (denoted by the red lines) goes as follows:

1. **Allocation and Offer:** The Manager allocates the Client-Service to the Nurse (Worker) and in so doing offers the work (defined in the Client-Service Contract) to the Nurse and the Service to the Patient. The System places the Nurse in the Client-Service Team (and, if necessary, in the Patient Care Team) after it does the “must be in” checks shown.
2. **Worker Acceptance:** The Nurse accepts the work and is confirmed in the Client-Service Team, which allows the Nurse to proceed with delivering the Service to the Patient.
3. **Patient Acceptance:** In this case the first task (Sub-Service) that the Nurse must do is gain the Patient’s Acceptance to the Contract. The Contract Data (information for the Nurse) will specify the valid forms of acceptance and the Nurse chooses/uses one of these forms.

It is important to understand that the System manages the Authorisation Process and that no overt contractual relation is necessarily perceived by the participants in the ordinary course of Service Provision. For example, when purchasing a product from a shop, the shop assistant and the customer will usually go through the purchasing process without necessarily being mindful of a contract being entered into.

In the Client-Service example, the Manager will likely be thinking “who can I get to do this job”, and the Nurse will begin delivering the Service and merely check that the Patient is receptive (if a more stringent *acceptance*, such as *written consent*, is not required). In this case the Nurse only needs to inform the System if the Patient refuses the offer of the Service.<sup>62</sup> This needs to be done to prove that, if faced with a subsequent complaint or legal action, the Service (say a medication) was in fact offered and the Nurse’s “duty of care” upheld.

The remaining four steps in the Service Management Process have to do with the management and delivery of the Client-Service, and the accounting and resolution phases. Apart from the nature of the Components and Component Instances involved and the *process definition*, there is nothing particularly unique about these steps compared to existing Process Management methodologies. They are therefore not detailed here.

## 5.9 Table of Model Concept Mechanisms

Table 3 lists all the Model Concepts (in the left column) and shows the “Model Mechanism” that is employed to incorporate each of them. The Model Mechanisms are only described briefly in order to keep the table concise and because different IT design techniques (for example, OO Design, design patterns and database design techniques) can be used to programme them in software development processes.

Concept	Model Mechanism
Global Services	Service Components (templates) are stored in global repositories and can be copied and uploaded.
Service Based Organisation	All work is defined as Services. When the System is implemented the required Services are copied, configured and/or modified. Other Services can be added later.
Serviceflow Dependencies	Serviceflow Dependencies are specified in Service Data
Service Cells	All the information necessary to create and manage a Service is specified in the Service Data. This means that no other data is needed which fulfils the concept.
Client-Services	Client-Services are used.
Service Profiles	This is stored as Standard Service Data entered during the Service Copying and Service Configuration processes.
Service Based Information	Information about Services is stored as Service Data and Client-Service Data.

<sup>62</sup> This would entail the Nurse classifying the Client-Service as “Refused” rather than the normal “Completed”.

Client-Service Management	The CMS is responsible for this – the Service Management Process deals with it.
Service Based Security	Service Data is accessed only through Storage and Retrieval Services as required.
<b>Concept</b>	<b>Model Mechanism</b>
Service Teams	This is fulfilled on 2 levels. Firstly, each Client-Service has its own Client-Service Team that comprises the personnel delivering the Client-Service. Secondly, a Service Team is automatically generated for each Client that comprises the personnel that perform any Service for the Client.
Contextual Groups	The System allows for any context to be represented by a Group or Group Hierarchy.
Group Based Organisation	The structure of Organisations is defined by their Group structure.
Group Based Access	All data (Group, Service and Contract) can be located and accessed through the Global Group Address.
Global Groups	Group Components (templates) are stored in global repositories and can be copied and uploaded.
Global Group Addressing	Each Group has a Global Group Address that is stored in Group Data and Group Instance Data.
Group Based Services	In order to be performed Services must first be associated with one or more Groups. They can then be performed by Group Members.
Group Based Info Storage	This is facilitated through Group-Service and Service-Contract associations that enable info to be accessed through from Global Group Addresses.
Set Based Groups	All Groups are “Set-based” in that all Group Members have a least one common attribute that helps define the “Group Type”. For example, all members of the “Staff Group” are Organisation staff.
Partial Permissions	“Partial Service Permissions” are employed when the right to perform a Service depends on the membership of more than one Group.
Group Based Management	Each Group has at least one Management Group whose sole task is to manage the Group.
Compulsory Compliance	The Model provides a System which, if mandated by authorities, is capable of facilitating Compulsory Compliance.
Compliant System	The use of Compliant Services enables System Compliance.
Worker-Service Contracts	The System requires that each Worker’s Contract details terms relating to the provision of Organisational Services by the Worker to Clients on behalf of the Organisation.
Client-Service Contracts	Each Client-Service has a Client-Service Contract associated with it.
Contract Based Services	Consent, Privacy and Accountability are managed through the provision and management of Service Contracts and Workers’ Contracts are also based of Service Provision.
Validation by Authorisation	Client-Service Contracts are managed and validated through the Model’s Authorisation Process.
The “5W’s” of Access Control	“Who” is the allocated Worker, “what” is the Service, “when” can be defined by time-based Groups such as Shift Groups, “where” can be defined by location Groups and “why” by the “Service Based Purpose” Concept.
Service Based Purpose	Information collection is performed as part of a Service and only information necessary for the provision of the Service is collected or accessed.
Service Based Access Principle	Access to information is only enabled in the performance of Services and the information accessed by each Service is restricted to that which is necessary for the performance of the Service.
Service Based Privacy	Consent to receive a Service is equated to the right to access information necessary for Service Provision.
Access by Authorisation	Only personnel Authorised to perform a Service can access the information and only that information which is necessary to perform the Service.
Consent Type Specification	The type of consent required for each Service, including specific detail, is included in the Service Data.

Service Based Consent	This is a central principle of the Model. Client Consent is on a per Service basis. Clients consent to the Service as a whole which includes consent for Service Delivery, permission to access and update relevant information and agreement to pay for the Service.
<b>Concept</b>	<b>Model Mechanism</b>
Bundled Client Consent	Clients can be asked to Consent to a number of Services at the same time. In all cases the details of what Services are included will be made clear to the Client.
Implied Permissions	The Client will be informed through Organisational policy and the basic Service Contract of the Organisation that implied permissions and consent are applied in specified circumstances. Most commonly this involves the Client being informally offered Services that they are considered to accept unless they take some action to refuse them.
Consent by Authorisation	Consent is communicated to the System through Authorisation Requests. The System manages Consent according to the Authorisation Requests it requires and receives.
Service Based Info Provision	In order to ensure that Consent received is always Informed Consent, information about each Service is directly or indirectly provided to participants before consent is validated.
Service Based Accountability	Client-Services are units of commerce in the Model. They have costs and prices, and profits and losses associated with them.
Client-Service Accounting	Each Client-Service is accounted for separately.
Required Auditing	Auditing Services are used to routinely determine whether policies and procedures are being properly enforced.
Client-Service Records	As part of Client-Service Accounting records of any interactions with the Client are kept on a per Client-Service basis.
Service Based Auditing	Auditing Services review the System by routinely examining Client-Service Records and outcomes.
Mandatory Authorisation	Authorisations from Managers, Workers and the Client are required for each Client-Service.
Authorisation Type Specification	Authorisation types are specified on a per Service basis by a 2-dimensional matrix. The 'format' dimension specifies the "Consent Type" (verbal, written, implied,...) while the "Timing Type" ( <i>ex ante</i> , <i>uno tempore</i> & <i>ex post</i> ) specified when Authorisation must be received.
Managerial Authorisation	Managers provide Authorisations on behalf of the Organisation. These Authorisations are equivalent to the Organisation <i>offering</i> the Service to the Client.
Client-Service Allocation	Managerial Authorisations also take place when a Client-Service is <i>allocated</i> to a Worker. The <i>allocation</i> is considered the <i>offer</i> to the Worker.
Worker Acceptance	Workers, by definition in the Model, have the right to refuse to provide a Service. By accepting to perform a Client-Service a Worker waives this right and Service Delivery can take place without the need for a Managerial re-allocation.
Worker Authorisation	Worker Acceptance is conveyed to the System as an Authorisation Request.
Client Authorisation	Client Acceptance is conveyed to the System as an Authorisation Request.
Service Based Authorisation	Authorisations take place and are required on a per Client-Service basis.
Sub-Service Authorisation	Authorisation for a Service implies by definition Authorisation of all Sub-Services.
Authorisation Management	The System manages Authorisations by ensuring that all the necessary Authorisation Requests are received before the state of a Client-Service is deemed "Authorised". Ex post Authorisations are considered as "Conditional Authorisations" by the System and are automatically monitored to ensure that conditions are met.

Table 3: Model Mechanisms

## 5.10 Conclusion

The Foundational Model Concepts are the seven most important Model Concepts. Each is associated with a corresponding General Concept and collectively they embody the core functionalities of the Model. At a deeper level, the Specific Concepts are also incorporated in the Model.

The Service Based Organisation and Service-Based Work Model Design Concepts embody the essence of the FMCs and provide the basis for the Model. The three Model Components – Services, Groups and Contracts – emerge from them. The Service-Group and Service-Contract associations are a key to understanding Model functionality as they facilitate the basic idea that “Groups do Contracted Services”.

The global nature of the Components enables Organisational Systems to be established in a simple manner. The logical connection between data (Service, Group and Contract Data) and Global Group Addresses facilitates a Global Access System (GAS) which provides interoperability benefits that are demonstrated in the Global Authorisation example.

The CMS contains its own operating system Services (CMSOS Services) together with all the Business Services necessary for Organisational operations. The entire System is contained in six Data collections, a Component Data collection and a Component Instance Data collection for each of the three Model Components.

Client-Services manipulate all data and a Service Management Process controls each Client-Service. Higher level Management Services are used to prioritise and allocate Client-Services and to provide Group Management and Contract Management functions.

System Compliance is enabled through the management of Client-Service Contracts. Authorisations from Managers, Workers and the Client for each Client-Service constitute contract *offers* and *acceptances* and are required by the CMS.

The following Analytical Evaluation chapter examines how the Model copes with the requirements outlined in a number of Standards: a *privacy standard*, a *health industry standard* and a *business security standard*.

## Chapter 6

# Analytical Evaluation

---

### 6.1 Introduction

#### 6.1.1 The Standard Requirement Set

The Analytical Evaluation takes three standards and evaluates whether or not the Model meets each requirement of the standards. The first standard is a Regulatory standard – the OECD Privacy Principles (Organisation for Economic Co-operation and Development 2006); the second is an IT Business Management standard – the Control Objectives for Information and related Technology (COBIT) framework (ITGI (IT Governance Institute) 2007); and the third is an IT industry standard – the United States Health Insurance Portability and Accountability Act (HIPAA) 1996. These standards were chosen because they were the most comprehensive found in each of the three research fields. Validation against the three standards is therefore a thorough test for the model.

330 relevant requirements in total were extracted from the three standards. These requirements were aggregated by removing duplicates and merging similar requirements to give a *Standard Requirement Set* of 50 requirements.

#### 6.1.2 Model Evaluation

The Standard Requirement Set is used in the following section to evaluate the Model. The evaluation considers the 50 requirements in turn. The requirements are divided into ten categories, with a section devoted to each category and a subsection devoted to each requirement.



In each subsection, the requirement is firstly numbered and named. This takes the form:

***SR#0: Example SR***

A “Description” of the requirement, the “Model Functionality”, a “Model Rating” and “Rating Conditions” are then detailed.

In most cases Model Functionality is described in terms of *Standard Model Services*. Standard Model Services are the CMSOS Services (Compliant Management System Operating System Services) that provide the Model’s functionality.

The Model Rating consists of a term (word) that describes the degree to which the Model complies with the requirement. The terms include:

1. **Compliant:** The requirement is met.<sup>63</sup>
2. **Achievable:** The requirement can be met if the Rating Condition is met.
3. **Non-Compliant:** The requirement is not met.

An “Achievable” rating requires the Organisation to utilise specific Services and have specified personnel and/or policies in place to meet the Rating Condition.

The ratings for all 50 Standard Requirements are summarised in a table in the final section of the chapter.

An Overall Evaluation of the Model is given in the Conclusion. The following Results and Discussion chapter explores this further.

---

<sup>63</sup> It is assumed that personnel comply with Standard Service Management Policies, receive adequate training about these policies and that their compliance is appropriately monitored.

## 6.2 Purpose Related Standards

### 6.2.1 SR#1: Purpose & Correctness

**Description:** Purpose of use must be specified before Personally Identified Data (PID) collection and the PID only used for the specified or a compatible purpose. The data must be correct (that is, relevant, accurate, complete & current) when used and retained for any required period.

**Model Functionality:** The *purpose* of collecting PID is for the performance of a Service. Before accepting a Service the Client is informed of the details of the Service and the PID that is required. This is a part of the Informed Consent process. *Correctness* is facilitated by employing Service processes that ensure that PID is double-checked where necessary and appropriate.

**Model Rating:** Achievable

**Rating Condition:** Employment of double-checking processes.

### 6.2.2 SR#2: Need to Know Access

**Description:** Only the PID necessary for the purpose must be accessed.

**Model Functionality:** Services only use the data necessary for their completion.

**Model Rating:** Compliant

**Rating Condition:** None

### 6.2.3 SR#3: Intellectual Property

<b>Description:</b>	Intellectual Property must only be used for specified purposes.
<b>Model Functionality:</b>	All Services that are associated with the use of intellectual property must be appropriately Authorised (on a per Client-Service, a per Service, or another basis).
<b>Model Rating:</b>	Achievable
<b>Rating Condition:</b>	Appropriate Authorisation processes.

## 6.3 Service/Task Related Standards

### 6.3.1 SR#4: Critical Protection

<b>Description:</b>	Critical business functions, services & data must be protected.
<b>Model Functionality:</b>	Services handle critical business functions and also protect data from being altered directly by users. Services can be classified as <i>Critical Services</i> where all data changes are recorded for future recovery. Services can also be protected by monitoring processes.
<b>Model Rating:</b>	Achievable
<b>Rating Condition:</b>	Appropriate use of Critical Services and monitoring.

### 6.3.2 SR#5: Security Integration

<b>Description:</b>	Security should be integrated into day-to-day procedures.
---------------------	---

**Model Functionality:** The CMS incorporates the Authorisation process into day-to-day procedures.

**Model Rating:** Compliant

**Rating Condition:** None

### **6.3.3 SR#6: Service Based Access**

**Description:** Need-to-Know Access must be applied to Services.

**Model Functionality:** Workers can only perform a Client-Service if they a) are on the Client-Service Team, b) are members of a Group (or combination of Groups) whose members are qualified to perform the Service, and c) have been allocated by Management to perform the Client-Service.

**Model Rating:** Compliant

**Rating Condition:** None

## **6.4 Consent Practice Standards**

### **6.4.1 SR#7: Data Collection**

**Description:** PID must be obtained lawfully & fairly with the client's knowledge.

**Model Functionality:** Before accepting a Service the Client is informed of the details of the Service and the PID that is required. This forms part of the Informed Consent process.

**Model Rating:** Compliant

**Rating Condition:** None

**6.4.2 SR#8: Consent for Purpose**

**Description:** PID must be obtained with consent, and consent must be again obtained if the PID is to be reused for a non-compatible secondary use.

**Model Functionality:** Client-Services and associated PID collection are only performed after Client consent is obtained. Secondary use of data (apart from compliant use of de-identified data) is restricted to compatible secondary Services. Clients are informed upon request of secondary Services that may be provided.

**Model Rating:** Compliant

**Rating Condition:** None

**6.4.3 SR#9: Consent Judgment**

**Description:** Professional judgment of the client's best interest or their previous expressed preference can be used in emergencies, when the client is not present or when goods are given to the client's agent on their behalf.

**Model Functionality:** In specified circumstances where consent judgments made without the client's knowledge are permitted, CMS Consent mechanisms require personnel to give reasons for their judgment. These reasons are audited. The right for individuals to make these judgments can be revoked or require supervision. Clients are informed of the circumstances where Consent judgments can be made without their knowledge and are given the opportunity to revoke such rights.

**Model Rating:** Compliant

**Rating Condition:** None

#### **6.4.4 SR#10: Consent Collection**

**Description:** Direct consent can be given orally, and indirect consent can be inferred by the client's refusal to object when given the opportunity or by professional judgment based on the circumstances.

**Model Functionality:** Where it is specified that oral or inferred consent can be used for a Service, the Worker is required to confirm to the CMS either directly or by implication (for example, in noting that the Service has been completed without complications), that Consent was secured from the client. Professional judgement is covered by SR#9.

**Model Rating:** Compliant

**Rating Condition:** None

### **6.5 Client Rights Standards**

#### **6.5.1 SR#11: Client Access**

**Description:** Clients must be given the opportunity to establish the existence, nature and purpose of their PID, and to gain access to it or obtain a copy of it (excluding restricted data).

**Model Functionality:** A record of Client-Services is maintained. Knowing which Services have been utilized enables a) a PID "Reporting Service" to list the Services and their nature and purpose, and b) a PID "Recovery Service" to enable Client to access their PID and/or copy it.

**Model Rating:** Compliant

**Rating Condition:** None

#### **6.5.2 SR#12: Accessor Information**

**Description:** Clients have the right to be informed of who may access their PID.

**Model Functionality:** Knowing which Services a Client has accepted allows the Groups and individuals able to perform a Service for the Client to be specified. A “Reporting Service” can compile an accessor’s list of Groups and/or individuals for any Client.

**Model Rating:** Compliant

**Rating Condition:** None

#### **6.5.3 SR#13: Disclosure Account**

**Description:** Clients have the right to receive a disclosure account.

**Model Functionality:** Disclosure accounts can be compiled from Client-Service Records (logs). These can detail who has accessed what information.

**Model Rating:** Compliant

**Rating Condition:** None

#### **6.5.4 SR#14: Restriction**

**Description:** Clients must be given the opportunity to restrict access to their PID and to restrict the purposes for which it is used.

**Model Functionality:** Workers can be removed from Service Teams at a Client’s request. “Restricted Service Teams” that can

require Client approval can be utilized where access needs to be restricted. Clients can also refuse the provision of any Service and thus the use of their PID.

**Model Rating:** Compliant

**Rating Condition:** None

#### **6.5.5 SR#15: Complaint & Appeal**

**Description:** Clients must be provided with appropriate complaint and appeal mechanisms, and be able to have incorrect PID rectified.

**Model Functionality:** Basic information about complaint, appeal and PID amendment processes is to be provided to Clients, including responsible personnel. “Complaint Services”, “Appeal Services” and “Amendment Services” are available for Clients.

**Model Rating:** Compliant

**Rating Condition:** None

#### **6.5.6 SR#16: Information Provision Process**

**Description:** Information provided to clients must be given in a timely manner and at a reasonable cost.

**Model Functionality:** The CMS ensures that “Information Provision Services” are delivered in a timely manner. All Services by definition have a cost associated with them. While the CMS provides efficient charging and accounting mechanisms to facilitate low administrative costs, it remains for Organisations to ensure that charges for Services are reasonable and reflect the nature and amount of work involved.



**Model Rating:** Achievable

**Rating Condition:** Reasonable fees are charged.

## 6.6 Organisation Rights Standards

### 6.6.1 SR#17: Directory Data

**Description:** With oral consent the client's name, location, general condition and religious affiliation may be stored in a directory.

**Model Functionality:** The Service that collects directory information will require oral or written consent. A specific "Directory Retrieval Service" is used for retrieving directory information.

**Model Rating:** Compliant

**Rating Condition:** None

### 6.6.2 SR#18: De-Identified Data

**Description:** Appropriately de-identified data may be used without client consent.

**Model Functionality:** Separate Services are used to extract de-identified data. These Services are designed to meet appropriate standards for de-identified data use.

**Model Rating:** Compliant

**Rating Condition:** None

### 6.6.3 SR#19: Restriction Refusal

**Description:** Organisations are not required to comply with client restriction requests.

**Model Functionality:** Policy determines when restrictions cannot be met. “Restriction Services” notify workers when Client requests are being transcended.

**Model Rating:** Compliant

**Rating Condition:** None

### 6.6.4 SR#20: Access Refusal

**Description:** Access and amendment rights can be denied in specified circumstances, which include the lawful refusal to disclose specified information about a minor to a parent or guardian, and to disclose information to persons suspected of domestic violence, abuse or neglect.

**Model Functionality:** Clients, Agents and others can be moved into specific “Denied Access”, “Restricted Access”, “Denied Disclosure” and “Restricted Disclosure” Groups.

**Model Rating:** Achievable

**Rating Condition:** Adequate personnel training.

## 6.7 Access Rules Standards

### 6.7.1 SR#21: Sensitive Information

**Description:** Sensitive information such as deceased PID and psychiatric notes must be protected.

**Model Functionality:** Sensitive information is protected by encryption and by changing Service Team access levels from “Standard” to “Restricted” (to specified persons), “Personal” (to the Client and Owner), or “Owner”.

**Model Rating:** Achievable

**Rating Condition:** Use of appropriate encryption standards

### 6.7.2 SR#22: Relevant Disclosures

**Description:** Relevant PID may be disclosed to a client’s family members, other relatives, close friends and other persons or agents identified by the client.

**Model Functionality:** Policy determines what is relevant to each group. “Disclosure Services” inform personnel of what information can be passed on to requesting persons.

**Model Rating:** Achievable

**Rating Condition:** Adequate personnel training.

### 6.7.3 SR#23: Basic Disclosures

**Description:** Client name, location and general condition may be disclosed to a person who names the Client. In addition, the Client’s religious affiliation may be disclosed to clergy and the Client’s death to family members, Client representatives and carers.

**Model Functionality:** Policy determines what is relevant to each group. “Disclosure Services” inform personnel of what information can be passed on to requesting persons.

**Model Rating:** Achievable

**Rating Condition:** Adequate personnel training.

#### **6.7.4 SR#24: Emergency Access**

**Description:** PID may be accessed for treatment purposes, operational purposes and in emergencies.

**Model Functionality:** Services defined as *Critical Services* or *Emergency Services* (which includes specified “Treatment Services” and “Operational Services”) enable access in emergencies.

**Model Rating:** Compliant

**Rating Condition:** None

#### **6.7.5 SR#25: Event Based Disclosure**

**Description:** PID must be disclosed to the Privacy Commissioner, to investigators of fraud or abuse and to support disaster relief efforts. With client consent PID can be used for marketing purposes.

**Model Functionality:** The Data Controller can add individuals to the Client Service Team to facilitate access to appropriate Critical Services and Emergency Services that can in turn access the required PID. Information for marketing purposes is obtained through “Disclosure Services” which require Client consent.

**Model Rating:** Compliant

**Rating Condition:** None

#### **6.7.6 SR#26: Victim Disclosure**

**Description:** Whistleblowers and crime victims may disclose relevant PID.

**Model Functionality:** The Data Controller can add individuals to the Client Service Team to facilitate access to appropriate Critical Services and Emergency Services that can in turn access the required PID, in line with official requests from a representative such as an ombudsman.

**Model Rating:** Achievable

**Rating Condition:** Subject to appropriate approvals/orders being gained.

#### **6.7.7 SR#27: Regulated Disclosure**

**Description:** PID must be disclosed (with appropriate assurances) when required a) by law, b) for public health activities, c) to government authorities dealing with domestic violence, abuse or neglect victims, d) for authorised health oversight activities, e) for judicial and administrative proceedings, f) for law enforcement purposes, g) to specified authorities regarding decedents, h) for organ and tissue donation purposes, i) for authorised research purposes, j) for serious health and safety threats, k) for specialized government functions, and l) for specified workers' compensation purposes.

**Model Functionality:** Special "Disclosure Services" can be designed to accommodate each case, with the Data Controller processing each request. Alternatively, the Data Controller can provide appropriate access to Critical Services and Emergency Services to the user, in line with specific official requests from the user.

**Model Rating:** Achievable

**Rating Condition:** Subject to appropriate Disclosure Services being available and approvals/orders being gained.

## 6.8 Authorisation Standards

### 6.8.1 SR#28: Authorisation

**Description:** Authorisation policies and procedures must deal with granting and modifying authorisations, supervision of staff (where appropriate) and staff termination. The need to protect sensitive information, to authorise access to programs, and to classify groups of computers need to be considered.

**Model Functionality:** The granting and modification of Authorisations are handled by *Authorisation Services*. *Management Services* for Groups and Client-Services allow managers to control Group memberships and supervise personnel. Terminations are handled by staff “Termination Services”. Sensitive information is protected by encryption, by setting appropriate Service Team Access Levels and by disallowing direct access to personnel. Programs are accessed through Services and computers belong to “Network Location Groups” where “Login Services” control computer access.

**Model Rating:** Compliant

**Rating Condition:** None

### 6.8.2 SR#29: Role Based

**Description:** Facility and system access should be based on staff roles and functions.

**Model Functionality:** User roles are modelled by membership of “Role Groups”. Access to facilities and Systems is based on Group memberships.

**Model Rating:** Compliant

**Rating Condition:** None

### 6.8.3 SR#30: Validation

**Description:** Authorisations must be checked for validity (that is, expiry, non-revocation and correctness). They must be non-conditional but they may be compound.

**Model Functionality:** *Authorisation Services* check the validity of each Authorisation Request, which includes checking for expiry, non-revocation and correctness. Authorisations are generally non-conditional, but Authorisations for Critical Services and Emergency Services can be conditional upon the Manager or Worker providing a valid reason for their actions.

**Model Rating:** Non-Compliant

**Rating Condition:** There is scope for conditional Authorisations.

### 6.8.4 SR#31: Written Authorisation

**Description:** Written authorisations must be in plain text and include the authoriser’s name and/or role, the accessor’s name and/or role, a purpose description, an expiry date or event, and the client’s signature and date.

**Model Functionality:** All written Authorisations contain the Authoriser’s name and role, the accessor’s name and role, a purpose description, and an expiry date or event, and the Client’s signature and date.

**Model Rating:** Compliant

**Rating Condition:** None

#### 6.8.5 SR#32: Information Flow

**Description:** The information flow in and out of the organisation must be defined.

**Model Functionality:** Information is always related to a Service, and the source of input information and the destination of output information are defined by the Service. Where Client-Service participants are outside the Organisation, information flow is controlled by the Service and monitored appropriately.

**Model Rating:** Compliant

**Rating Condition:** None

#### 6.8.6 SR#33: Transaction

**Description:** Trust is required for all e-transactions and critical transactions, and specified transactions must be authorised and/or verified.

**Model Functionality:** All participants in Client-Services must be authenticated and Authorised. The Level of Trust depends on the authentication process used. The CMS can require Client-Service participants to have specified Levels of Trust by requiring that specified “Authentication Services” are used. *Authorisation Services* ensure that relevant Authorisations are received by all Client-Service participants. Verifications are required for specified transaction Services.



**Model Rating:** Achievable

**Rating Condition:** Depends on Authentication processes used.

### 6.8.7 SR#34: Inter-Domain Authorisation

**Description:** Authorisation policies for inter-domain access need to take into account the requirements of service providers, suppliers, customers, agents, subcontractors, and staff and management in controlling organisations.

**Model Functionality:** Inter-domain access is facilitated by recognizing Group memberships from other domains or by placing outsiders in Groups within the organisational domain. This controls the Services available to outsiders. The participants' requirements are catered for in the design of Services and Groups and in the allocation of Services to Groups. Preferably, inter-domain access is given to outsiders rather than copies of information being sent out.

**Model Rating:** Compliant

**Rating Condition:** None

## 6.9 Compliance Practices Standards

### 6.9.1 SR#35: Openness & Accountability

**Description:** Policies and practices must be open and the Data Controller accountable.

**Model Functionality:** Openness is facilitated by using standard and defined Services. Accountability is maintained through appropriate record keeping, auditing, monitoring and reporting Services.

**Model Rating:** Compliant

**Rating Condition:** None

### 6.9.2 SR#36: Compliance

**Description:** Compliance by the organisation and its staff with security management and privacy requirements must be ensured.

**Model Functionality:** Organisational compliance is achieved through using standard approved Services. Staff compliance is facilitated by providing necessary guidance information within Service processes, appropriate staff training and proper activity monitoring.

**Model Rating:** Compliant

**Rating Condition:** None

### 6.9.3 SR#37: Risk Management

**Description:** Cost-effective risk management techniques should be utilized in order to ensure the confidentiality, integrity and availability of all PID.

**Model Functionality:** Risk management relates to all security issues. With regard to access control and Authorisation procedures, risks are managed through monitoring Authorisations for errors and potential violations. Confidentiality and integrity of PID is maintained through proper Authorisation of data *Access Services*. Availability is achieved by allowing emergency and critical accesses. Standardisation provides cost-effectiveness.

**Model Rating:** Achievable

**Rating Condition:** System redundancy to facilitate appropriate availability protection.

#### 6.9.4 SR#38: Security Responsibility

**Description:** Minimum security responsibilities for staff, including the enforcement of separation of duties requirements must be established.

**Model Functionality:** Staff members are informed by the System of steps that they need to follow. However, System simplicity is the key to Compliance. Staff members are not assumed to know anything about security. Requirements such as the Separation of Duties are handled by appropriate *Authorisation Services*.

**Model Rating:** Compliant

**Rating Condition:** None

#### 6.9.5 SR#39: Skills & Qualifications

**Description:** The skills and qualifications of security personnel must remain current.

**Model Functionality:** Skills and qualifications are represented by Group memberships. Skill and qualification checks are managed by “Quality Assurance Services”. The CMS allows SMEs to function without security personnel either by not requiring them or by outsourcing the functions.

**Model Rating:** Achievable

**Rating Condition:** Automated security features and outsourcing being acceptable.

**6.9.6 SR#40: User Account**

**Description:** User accounts must employ unique user IDs and be managed appropriately.

**Model Functionality:** It is standard practice to use unique user IDs. User ID is equivalent to “BUG ID number”. BUGS are managed by “Registration Services” and staff “Termination Services”.

**Model Rating:** Compliant

**Rating Condition:** None

**6.9.7 SR#41: Privacy Personnel**

**Description:** The organisation must designate a Data Controller (privacy officer) and provide appropriately trained request and complaint handling officers.

**Model Functionality:** Policy defines the role of the Data Controller, and the request and complaint handling officers. The CMS allows SMEs to function without security personnel either by not requiring them or by outsourcing the functions.

**Model Rating:** Achievable

**Rating Condition:** Automated security features and outsourcing being acceptable.

**6.9.8 SR#42: Data Correction**

**Description:** Data must be amended when substantiated errors are notified.

**Model Functionality:** “Amendment Services” handle amendment requests.

**Model Rating:** Compliant

**Rating Condition:** None

## 6.10 Information Provision Standards

### 6.10.1 SR#43: Information Provision

**Description:** Clients must be informed of a) PID uses, b) their revocation rights (including after emergency accesses), c) the non-conditional authorisation requirement, d) ramifications of their failure to consent, e) disclosure possibilities, f) the availability of disclosure accounts, access mechanisms and PID copies, g) reasons for withholding information, and h) the provision of signed written authorisation copies.

**Model Functionality:** “Client Information Services” handle the provision of information to Clients. All Client-Services specify the information that must be provided to Clients and the way(s) it can be delivered. General policy information is provided to Clients directly in writing and may be given routinely or upon request. Clients are notified of the sources of specific Service-based information and it is given to them at their request or routinely, depending on the nature of the Service.

**Model Rating:** Compliant

**Rating Condition:** None

### 6.10.2 SR#44: Disclosure Account Data

**Description:** Disclosure accounts for clients must contain the disclosure date, the receiver’s name and address, a brief data description and a brief purpose statement.

**Model Functionality:** Disclosure accounts are generated by “Disclosure Services” and contain the disclosure date, the receiver’s name and address, a brief data description and a brief purpose statement.

**Model Rating:** Compliant

**Rating Condition:** None

### 6.10.3 SR#45: Staff Reminder

**Description:** Staff must be reminded of security risks and their responsibilities.

**Model Functionality:** “Worker Information Services” remind workers of their responsibilities and of security threats. Service-based information is provided as part of the Service Delivery process.

**Model Rating:** Compliant

**Rating Condition:** None

## 6.11 Monitoring/Auditing/Reporting Standards

### 6.11.1 SR#46: Activity Monitoring

**Description:** Procedures to review, record and examine activity on PID and to check automated security functions must be implemented.

**Model Functionality:** *Access Services* are used to access PID and all *Access Services* activities are recorded and monitored by *Monitoring Services*. *Monitoring Services* look for unusual accesses and access patterns and report any suspicious activities. Examination of Service Records

reveals when particular PID has been accessed. Automated Security functions and Monitoring Services are themselves monitored by “Security Monitoring Services”, tested by “Security Testing Services”, and audited by “Auditing Services” to ensure that they are functioning correctly.

**Model Rating:** Compliant

**Rating Condition:** None

#### **6.11.2 SR#47: Security Auditing**

**Description:** All changes to security settings and authorisations must be recorded.

**Model Functionality:** *Monitoring Services* record all changes to security settings and Authorisations. These Services are monitored, tested and audited.

**Model Rating:** Compliant

**Rating Condition:** None

#### **6.11.3 SR#48: Management Auditing**

**Description:** The management of user accounts must be audited.

**Model Functionality:** *Auditing Services* perform auditing functions which include the management of user accounts.

**Model Rating:** Compliant

**Rating Condition:** None

#### **6.11.4 SR#49: Incident Management**

**Description:** Policies and procedures for the timely resolution of security and information flow incidents must be

established in order that violations and exceptions are effectively prevented, detected, reported, contained or corrected.

**Model Functionality:** *Monitoring Services* report suspect incidents. Each of these incidents is then dealt with by “Incident Management Services”. As part of the process, “Incident Reports” are produced by *Reporting Services* for management to make security corrections and the corrections are documented.

**Model Rating:** Compliant

**Rating Condition:** None

#### 6.11.5 SR#50: Required Documentation

**Description:** The following must be documented: a) security incidents and outcomes; b) specified actions, activities and accesses; c) authorisations; d) compliance with notice requirements; e) client accessible PID sets; and f) the provision of disclosure accounts.

**Model Functionality:** Wherever Service documentation is required, *Reporting Services* are employed to collect, store and manage the information.

**Model Rating:** Compliant

**Rating Condition:** None

### 6.12 Model Rating Table

The following Table (Table 4) summarises the ratings of the Model against each of the requirements in the Standard Requirements Set. The “Score” in the last column supplies an indication of the level of Compliance in each Category. The Score is



Category	No	Requirement Name	Rating	Score
Purpose Related	1	Purpose & Correctness	Achievable	4 / 6
	2	Need to Know Access	Compliant	
	3	Intellectual Property	Achievable	
Service/Task Related	4	Critical Protection	Achievable	5 / 6
	5	Security Integration	Compliant	
	6	Service Based Access	Compliant	
Consent Practices	7	Data Collection	Compliant	8 / 8
	8	Consent for Purpose	Compliant	
	9	Consent Adjudgement	Compliant	
	10	Consent Collection	Compliant	
Client Rights	11	Client Access	Compliant	11 / 12
	12	Accessor Information	Compliant	
	13	Disclosure Account	Compliant	
	14	Restriction	Compliant	
	15	Complaint & Appeal	Compliant	
	16	Information Provision	Achievable	
Organisation Rights	17	Directory Data	Compliant	7 / 8
	18	De-Identified Data	Compliant	
	19	Restriction Refusal	Compliant	
	20	Access Refusal	Achievable	
Access Rules	21	Sensitive Information	Achievable	9 / 14
	22	Relevant Disclosures	Achievable	
	23	Basic Disclosures	Achievable	
	24	Emergency Access	Compliant	
	25	Event Based Disclosure	Compliant	
	26	Victim Disclosure	Achievable	
	27	Regulated Disclosure	Achievable	
Authorisation	28	Authorisation	Compliant	11 / 14
	29	Role Based	Compliant	
	30	Validation	Non-Compliant	
	31	Written Authorisation	Compliant	
	32	Information Flow	Compliant	
	33	Transaction	Achievable	
	34	Inter-Domain Authorisation	Compliant	
Compliance Practices	35	Openness & Accountability	Compliant	13 / 16
	36	Compliance	Compliant	
	37	Risk Management	Achievable	
	38	Security Responsibility	Compliant	
	39	Skills & Qualifications	Achievable	
	40	User Account	Compliant	
	41	Privacy Personnel	Achievable	
	42	Data Correction	Compliant	
Information Provision	43	Information Provision	Compliant	6 / 6
	44	Disclosure Account Data	Compliant	
	45	Staff Reminder	Compliant	
Monitoring/ Auditing/ Reporting	46	Activity Monitoring	Compliant	10 / 10
	47	Security Auditing	Compliant	
	48	Management Auditing	Compliant	
	49	Incident Management	Compliant	
	50	Required Documentation	Compliant	

Table 4: Model Ratings

obtained by attributing a Score to each Rating (Compliant = 2, Achievable = 1, Non-Compliant = 0) and adding the Scores for the Category. The Total Score for all the 50 Requirements is 84/100. The Score is purely a qualitative measure used to highlight the areas where compliance is good as opposed to not so good. It does not take into account the relative significance of each requirement. Neither is it meant to be a formal/mathematical evaluation as this was beyond the scope of the research in terms of available timeframe and local expertise.

### **6.13 Conclusion**

Overall the Model was shown to be able to comply with the Standard Requirement Set in 49 of the 50 requirements. The Model rated “Achievable” for 14 of the 50 requirements, the majority of these being in the “Access Rules” and “Compliance Practices” Categories.

The one point of “Non-Compliance” relates to part of SR#30 (Validation) that requires authorisations to be non-conditional. The requirement for non-conditional authorisations arises from traditional access control methodologies such as Mandatory Access Control (see 3.3), where System administrators strictly control all accesses. This is not applied in the Model because the Model is designed to give control to Business Managers and Professional personnel and to be useful in SMEs where there are no designated System administrators. Furthermore, conditional authorisations are necessary to deal with emergency and critical situations where the need to provide a Service to a Client outweighs the need for strict privacy controls.

The “Access Rules” requirements pose some implementation challenges as the rules for disclosure of information are fairly complex and relate to various classes of persons. While the CMS can greatly assist personnel in dealing with these issues, there is a need for the personnel to have a level of expertise in the operation of the Services involved.

A number of the “Compliance Practices” requirements place specific responsibilities on Security related personnel. The Model is designed for the automation and/or

outsourcing of some of these responsibilities, and accordingly additional checks and balances need to be employed to achieve Compliance.

Most of the remaining “Achievable” ratings are related to System requirements that are specific in nature and need special Services to achieve Compliance. While some Organisations may see these special Services as unnecessary or irrelevant, the CMS can certainly deliver them.

On the positive side, the Model rated “Compliant” for 35 of the 50 Standard Requirements, that is, for 70% of the requirements. This is an encouraging result as many of the requirements are demanding and require specific functionalities which either do not exist or rarely exist in current Organisational systems. This result shows that the standardised nature of the Model effectively deals with a good majority of the Standard requirements. In particular, it does extremely well in the Categories that are Client-related (“Consent Practices”, “Client rights” and “Information Provision”), scoring 25 out of 26.

As the Model is designed to be used by personnel without specific security or system administration skills, the 70% compliancy proportion can be considered an excellent result, particularly as the remaining requirements generally relate to less critical and uncommon situations.

In summary, this essentially means that 70% compliance can be achieved at little cost using standard CMS Services and that compliance can be increased to near 100% by employing Services that incur additional, but not exorbitant, cost. The extra costs would be chiefly for outsourcing and/or personnel training and supervision.

The following Results and Discussion chapter takes a broader view of the Model’s advantages and drawbacks, and gives further insight into the issue of Organisational Compliance and its relevance.

## Chapter 7

# Results and Discussion

---

### 7.1 Introduction

This dissertation has described a Model, the CMS Model, together with the required design concepts, for producing a Compliant, Efficient and Globally Functional IT-based business management System.

Legislation such as the Sarbanes-Oxley Act has highlighted the need for business management systems to meet compliance requirements. This introduces a problem because regulations are usually written in terms of general principles that are difficult to translate into the logical terms necessary in a computer system. For example, a privacy principle that states that data can only be used for the purpose for which it was collected is difficult for an IT system to implement because there is no efficient way of representing a purpose in a logical way.

The research therefore sought to find a way to frame regulations in a logical way that could be translated into systems to facilitate their compliance. The aim was to find a way to enable built-in compliance in an IT-based business management System.

To be practically useful, such an IT-based business management System must not only be compliant, it must be commercially viable. This means that the IT System must be efficient in terms of its business management role and globally functional in terms of its connectedness.

The proposed System, accordingly, needed to meet three Central Requirements – Compliance in a regulatory sense, Efficiency in a business management sense, and Global Functionality in an IT sense. The three Central Requirements bring together the Legal, Business Management and IT research fields.

### 7.1.1 Incorporating the Old and the New

The primary purpose of this chapter is to discuss the nature of the advances achieved in the research and to place these in context with existing methodologies. While the research adds a number of new concepts, it fundamentally takes many existing concepts, modifies them where necessary and combines them to achieve the aim of a Compliant, Efficient and Globally Functional model. In order to better comprehend this methodology, comparison with the invention of the first successful airplane by the Wright Brothers (Wilbur and Orville) is useful (Wikipedia 2011).

The Wright Brothers tested various existing wing designs in a purpose built six foot wind tunnel to determine the most efficient design, and tested their control ideas with kites and gliders for several years before building the powered Wright Flyer 1 in 1903. This aircraft, constructed with a spruce frame covered with muslin, and incorporating a lightweight gasoline engine (fabricated in the brothers' bicycle shop), was the first to fly successfully. Its wing, construction and engine were based on existing ideas and techniques.

From the outset of their experiments the brothers regarded *control* as the unsolved third part of “the flying problem”. They believed sufficiently promising knowledge of the other two issues — *wings* and *engines* — already existed. Their fundamental breakthrough was their invention of “three-axis control”, which enabled the pilot to steer the aircraft effectively and to maintain its equilibrium. The first axis of control was the “elevator”, used to vary altitude. The second axis involved “banking” to turn (they observed this in birds and people riding bicycles), which was initially achieved using “wing warping” to twist the wings and later with ailerons. The third aspect of control was their discovery of the true purpose of a movable “vertical rudder” — to aim or align the aircraft correctly during banking turns and when leveling off from turns and wind disturbances (as opposed to the existing idea that it was for changing the direction of flight as with a ship's rudder).

In summary, the Wright brothers looked at existing methods and theories, adapted existing inventions, brought ideas together and added some of their own. The ideas they added focused on *control*, which enabled the development of relatively stable and safe aircraft.

In similar vein this research takes existing methods and theories, adapts existing inventions, brings ideas together and adds new ways to *control* IT systems. The existing methodologies adopted have to do with Services, Groups, Privacy, Consent, and Accountability. The new method of *control* relates to using Authorisation mechanisms to enable enforceable Digital Service Contracts. This method of control enables compliant IT management Systems to be developed.

In line with the Wright Brothers analogy, Compliance can be viewed as the Central Requirement that needs to be solved; after all, promising knowledge about Efficiency and Global Functionality already exists and needs little more than to be brought together and adapted. Compliance is therefore the main focus of both the research and the validation process. In contrast, the Efficiency and Global Functionality of the Model are not explicitly tested; arguments about how the Model meets these requirements are instead made.

This chapter, accordingly, focuses more on discussing the issue of Compliance, with secondary attention being devoted to Efficiency and Global Functionality. It also seeks to separate out the Concepts that are new or significantly changed from existing methodologies.

As the Wright Brothers introduced a “safe” (stable) flying system, the Model introduces “safe” IT systems – in the sense that such systems will not violate persons’ privacy, right to consent and security of their possessions. With safe IT systems in place, the potential for automation is substantially increased. As airline passengers are comfortable with airplanes that utilise auto-pilots, organisations and their customers can become accustomed to automated business systems which “do the right thing”.

### **7.1.2 Tradition and Future Possibilities**

The practice so far as law is concerned has been to frame business and IT legislation and regulation in a manner that can be administered by the court system. The future is to build legal requirements into business management systems so that they are inherently compliant. This means that future legislation and regulation needs to relate directly to the functionalities that are built into IT based business management

systems. In other words, legislation and regulation needs to be framed in a logical manner and written in terms of Services, Groups, Service Contracts and Authorisation mechanisms.

Business tradition has been to write organisational policies to implement legislative and regulatory requirements and to use custom designed IT systems as tools that aid business management. The future is to use IT systems to manage business (that is, “the system runs the business”) and to build legislative and regulatory requirements into the system (that is, policy is implemented in the system and is a part of the system).

IT tradition has been to custom design business software to meet the business needs of individual organisations and to largely ignore regulatory requirements unless they are expressed as design requirements. Future IT systems will manage organisations in a more standardised way that is inherently compliant with regulatory requirements.

### **7.1.3 Towards Automated Business Management**

The main purpose of business IT systems is to provide benefits through the automation of business processes. Adding the need for the business processes to be Compliant with Regulations, the proposed System needed to **automate business management in a compliant manner**. It is argued that a Compliant System aids automation because users have more confidence in automated features that are proven to be Compliant than those that are unproven. This is analogous to automobile owners having confidence in vehicles that conform to design standards.

It should be noted that the research is not about how to modify current systems so that they meet the central requirements, but rather to specify how future systems can be designed to meet them. Proven principles from current methodologies nonetheless underpin the proposed Model.

With a view to the future, the research makes some assumptions about future directions in IT-based business management. These broadly include:

1. Compliance requirements will become more stringent;

2. Business-to-business (B2B) systems will have to be interoperable;<sup>64</sup>
3. Online transactions will become common; and
4. Automated system management will become the norm.<sup>65</sup>

The aim of the research was not to develop a specific software solution that meets the central requirements, but rather to develop a Model upon which specific Systems can be based. The Model is based on a set of Model Requirements extracted from proven methodologies and design principles. The set of Model Concepts, itself a result of the research, is derived from the Model Requirement set.

#### **7.1.4 Results and Discussion Topics**

The following section details the contributions of the research that include the Set of Model Concepts, the Model itself, and the Standards Requirements. The Model is then analysed by exploring its validity, comparing it with current methodologies and highlighting its limitations.

The real world application of the Model is then discussed in terms of the three Central Requirements. This leads into the final two sections that detail Future Work and provide a Conclusion.

### **7.2 Contributions of the Research**

#### **7.2.1 Model Concept Set and New Concepts**

The Model Concepts are described in Chapter 4 and are summarised in Table 2. Figure 22 shows the relationships between the Model Requirements and Model Concepts.

The Model Concepts can be viewed as a collection of design principles for IT-based business management systems. They can be used collectively as a set of principles to

---

<sup>64</sup> For example, to facilitate automated supply chain management.

<sup>65</sup> Leading to such things as a reduction in IT administrators and outsourcing of IT management.



guide design. The names and definitions for each Concept can also be used for reference purposes. They can therefore be seen as a resource in themselves, independent of the Model.

While the Set of Model Concepts embodies the ideas contained in the Model and is useful as a collection of design principles, some of the concepts can individually be considered as research results. These concepts are new concepts or at least contain new adaptations of existing concepts. The following subsections detail twelve such concepts. The first seven are “**Major New Concepts**” as they contain significant new ideas. The remaining five are “**Significant Concept Adaptations**” as they contain significant new variations of existing ideas.

#### 7.2.1.1 Contract Based Services

Contract Based Services utilise Digital Service Contracts (DSCs). DSCs provide the focus point for the automated management of Services. The System manages Services in a Compliant fashion by controlling Service Delivery according to the contractual decisions it receives by way of Authorisation Requests. For each Client-Service the System directs two types of Service Contracts: a Manager to Worker Contract (part of a *Service Based Employment Contract*) and a Manager to Client Contract (the *Client-Service Delivery Contract* covering Privacy, Consent and Accountability).<sup>66</sup> The key advantage of these Digital Service Contracts is that they amalgamate the collection of Privacy permissions, Consent decisions and agreed consideration (that is, the cost of the Client-Service) into one “Acceptance Authorisation” made by or on behalf of the Client. This is of significance because it provides the mechanism necessary to guarantee Compliance, significantly streamlines Service administration (Efficiency) and provides the standardisation necessary for online contracts to be negotiated from anywhere (Global Functionality).

---

<sup>66</sup> The Manager to Client Contract may be presented to the Client by a Manager, a Worker or by the System and, where the Service is a Sub-Services of an already accepted Service, acceptance may be implied without explicit presentation.

### **7.2.1.2 Validation by Authorisation**

Using digital (System) Authorisations as a standard mechanism for managing Contracts is new and significant. Contract offers and acceptances are entered into the System as Authorisation Requests that the System then checks. Of particular importance is the fact that all Client acceptances are recorded and are necessary before Services are provided (except in specified emergencies).

The systematic and open approach to collecting legitimate acceptances through Authorisations arguably reduces the likelihood of Service Delivery errors and fraudulent activities. Client acceptances of clearly enunciated contracts, where Informed Consent is clearly received, should also reduce the likelihood of false claims and litigation by Organisational Clients. Consider the case of a medical procedure that has negative results. By checking Authorisation Requests the System is able to ensure that the medical practitioner is properly qualified, trained and experienced in providing the procedure (Service) in question, thereby reducing the likelihood of errors.<sup>67</sup> By being able to prove that the Client gave Informed Consent and thus understood the procedural risks, the likelihood of false claims and litigation is reduced.

The Service Based Authorisation concept ensures that Authorisation records exist in the System and can be used to validate access permissions, Consent decisions and Contract acceptances. This facilitates Compliance with record keeping regulations and provides evidence to show that the Organisation has legitimately performed Client-Services. Automatic monitoring of Authorisation Requests and dynamic analysis of request patterns also provides a means for detecting fraudulent and erroneous actions by Organisation personnel.

### **7.2.1.3 Service Based Purpose**

Service Based Purpose may seem like an innocuous, inconsequential Concept but it is actually the linchpin for the entire research because it enables the amalgamation of

---

<sup>67</sup> The systematic nature required in the definition of Service tasks and the procedural Information available to practitioners also helps ensure that best practice is achieved and practitioners do not solely have to rely on their memory.

Privacy, Consent and Accountability. It essentially takes what have previously been qualitative decisions about Privacy and Consent, and transforms the decision making process into a logical, quantitative one, thus allowing amalgamation with Accountability decisions that have always been logical in nature. With all three decision types now being quantitative and logical, IT Systems are able to both receive Authorisation decisions as input from users and automatically make specified logical decisions based on predefined parameters.

This idea can also be seen when the 5 Ws of Access Control (who, what, when, where and why) are considered. The first four Ws are logical in that the user, requested resources, time and location are all quantitatively defined for any access. In contrast the last “why” is not. In the past the answer to the question “why does the user want access to the resource” often required a decision based on qualitative reasoning. Service Based Purpose, by definition, makes the answer to that question the same – “in order to perform the requested Client-Service”. The IT system can then answer the ensuing question “have all the required Authorisation Requests been received?” with a logical “yes” or “no”. The decision making process is now wholly quantitative, lending it to automation through the use of IT.

#### **7.2.1.4 Service Cells**

The idea of a Service Cell unites Legal, Business Management and IT concepts into a single modular “unit of commerce”. This means that, from a technical viewpoint, all the information required to enforce Authorisation, Privacy, Consent and Accountability requirements are contained in the *Service Metadata* (that includes the Service Description). This enables Services to be copied. It enables Business Management mechanisms that allow new Services to be easily added and old Services to be removed. And it enables standard Service Management Processes to be implemented, providing simple and consistent administrative mechanisms. Organisations are viewed as collections of Service Cells – from a legal perspective, a management perspective and an IT System perspective.

The Service Cell Concept is the catalyst for the implementation of many of the other important Model Concepts. These include Service Based Authorisation, Mandatory Authorisation, Authorisation Type Specification, Service Based Consent, Consent

Type Specification (where Consent is specified on a per Service basis), Service Based Privacy (where information is only accessed through Services that act as Transformation Procedures), Service Based Accountability, and Service Based Security (where security is focused around Services).

Of particular note are **Service Based Consent** and **Service Based Privacy**, which in themselves are new and potentially significant concepts in the Access Control research field.

#### **7.2.1.5 Group Based Services**

Group Based Services has to do with attaching Services to Groups. While this is chiefly a new Access Control concept (that is, a Computer Security concept), it also has implications in the management of Services because it is a management mechanism. Attaching Services to Groups is the way that “who can do what” is specified in the Organisational System – the “who” being the Group members and the “what” being the Services.

In the past Access Control mechanisms have largely focused on the subject-right-object idea where someone (the system administrator or object creator) explicitly specifies rules relating to users and information. This has been very problematic and has resulted in systems where it is difficult to efficiently implement and guarantee adequate privacy and security requirements.

Attaching Services to Groups provides a simple yet powerful way of defining access rules that does not require system knowledge to implement at the coalface. It enables managers to administer Access Control invisibly when Service based work decisions are made. **Standard management decisions automatically trigger correct Access Control specifications – a breakthrough and central goal of the research.** The mechanism has substantial benefits in relation to Compliance (enabling fine-grained access control) and Efficiency (reducing/alleviating the administrative access control burden).

#### **7.2.1.6 Global Group Addressing**

Global Group Addressing is the mechanism that enables Organisational Groups to be recognised on a global scale. It allows both private users and users in other Organisations to interact with an Organisation through its Groups. It opens membership in specified groups to users outside the Organisation, enabling the users to have limited and controlled access to Organisational resources, for example, GPs accessing the medical records of their patients held in Organisations like hospitals.

Global Group Addressing has a further great commercial benefit that will be addressed in future work. This has to do with linking Global Groups to “Identities”. Groups represent individuals (as Base Level User Groups), Organisations and Organisational units in the system and can be used as a mean to do business, particularly remote online business. A registration process that links Identities to Groups can be used to prove that a user legitimately belongs to a particular Group or that a Group legitimately represents an Organisation or Organisational unit. The Group can then be used to represent the party in communications and transactions. For example, when a customer buys a product online, the customer can know they are dealing with a legitimate business and the business can know that the customer is real.

#### **7.2.1.7 Partial Permissions**

Partial Permissions provide a key mechanism that facilitates Service Based Access and thus Service Based Privacy. Incorporated with Global Groups, they also facilitate Access Control on a global scale by enabling access permissions (that is, rights to use Services) to be stored as Group metadata. This makes it possible to require memberships of multiple Global Groups (including remote Groups in other Organisations) before access to Group Based Services are permitted, without the need to check separate Organisational or inter-Organisational rule-sets. For example, a GP who is a member of the “Doctor” Group in a recognised medical practice and “Patient A’s Service Team” in a hospital could use a Hospital Service to access Patient A’s medical records held by the hospital.

#### **7.2.1.8 Compliant System**

While the ideas of constructing IT systems that comply with particular legislation (such as the Sarbanes-Oxley Act) and of rating software against standards (such as Computer Security standards like the Common Criteria) are not new, the idea of using a Service-based system that utilises Digital Service Contracts to achieve Compliance is original.

Extending the concept of IT-based compliance into the regulatory domain and suggesting that regulations also be Service based are also new ideas. The concept of a Compliant system in this research is unique in that it incorporates the idea of specifying Regulations in a digital/logical way that enables them to be translated directly into software. This is in contrast to current methodologies that have software being written in order to comply with written regulations, so that when the regulations change the software often must be rewritten.

#### **7.2.1.9 Group Based Organisation**

The structure of a Group Based Organisation is fundamentally defined by Group hierarchies. Groups can represent many different collections of personnel, from role Groups whose members share the same qualification or job type; to Service Teams whose members all deal with the same Client or Group of Clients; to Project Teams and Committee Groups whose members work on the same set of tasks.

While the ideas of Groups and Group hierarchies are not new, the Group management mechanism the research has developed is new. Each Group has by definition one or Management Groups associated with it. The Management Groups perform Group Management Services for the Group. The Management Group to Group relationships define the supervisory structures within the Organisation and the Management Services define the tasks that Organisational Managers perform.

#### **7.2.1.10 Service Teams**

The idea of Service Teams in Organisations is not new, but the concept that each Client has their own distinct Service Team by definition is original. Having distinct

(or unique) Service Teams does not preclude a standard Service Team being assigned to a number of Clients, as is the case in other methodologies.

There are a number of benefits of distinct Service Teams. These include having fine-grained control of Service Team membership and enabling Clients to deny membership of their Service Team to unwanted individuals. They also provide a means for keeping track of which personnel are currently looking after a Client (or who have looked after the Client in the past) and who Authorised each membership request.

#### **7.2.1.11 Contextual Groups**

Context relates to variables that are used to determine who may access particular Services or information. For example, a user may be required to be performing a particular role or be in a specified location in order to perform a task, or may be restricted as to what times the task can be performed. While role-based Groups are not a new idea, the idea of being able to represent all contextual variables through Groups breaks new ground.

While representing the location of all Workers by mapping coordinates may be useful, placing all Workers in a location (say Ward A) in a location based Group is of greater use from a management perspective when, for instance, a Manager wants to know which nurses have been allocated to Ward A. Using contextual Groups also enables Partial Permissions to be used to specify allowable Services; for example, only Nurses in the Intensive Care Ward can perform a specified Intensive Care Service.

#### **7.2.1.12 Service Based Information Provision**

Service Based Information Provision proposes a standard way of ensuring that a Client receives the necessary information and assistance in giving Informed Consent. This serves to provide standard information that can be easily copied and enforceable Contracts where Consent can be proven to be informed.

When Services are designed, expert opinion is used to determine the information that is required for a particular Service as well as the means that must be used to deliver

the information to the Client and to check that the Client has understood the information. What is new in this research is the standardised process used for all Services to achieve the required outcomes rather than the broader ideas of information provision and Informed Consent.

### **7.2.2 The Model**

Seven Foundational Concepts were extracted from Model Concept set. Each of these corresponds to one of the seven General Concepts identified in Chapter 1. Figure 3 shows the relationships.

The Model is based on the seven Foundational Concepts. The Foundational Concepts reflect the relationships between the seven General Concepts, for example, Service Based Privacy implies the Privacy is managed through Services.

Figure 31 shows the relationships between the Foundational Concepts and provides a key to understanding the Model. It essentially shows that Services are performed by Group members and controlled through the Authorisation of Service Contracts that are used to fulfil Consent, Privacy and Accountability requirements.

Consent, Privacy and Accountability are the areas where Compliance is required, while Services, Groups and Service Contracts are the Model Components. Authorisations are the tool by which the Model Components are controlled – Services require Authorisation before being performed, Group membership is gained through Authorisation, and Service Contracts are offered and accepted through Authorisations. The proper Authorisation of Service Contracts ensures Consent, Privacy and Accountability Compliance. In this sense Authorisations are the glue that binds all the General Concepts together.

Considering the complexity of the issues involved in the Compliant management of Business Services, Systems based on the Model will only require (most) users to have a basic working knowledge of what constitutes a Service, of Group membership, Contract management and Authorisations. The Model is designed with user friendliness in mind and uses simple Concepts of Services, Groups, Contracts and Authorisations to enable this.



By enabling users (Managers and Workers) to make Authorisation decisions at the coalface, the Model takes away the need for System Administrators to perform access control functions. As the decisions are made as a part of normal work processes, like allocating a Client or a Service to a Worker, no additional administrative burden is imposed on Workers. Administration is triggered by normal work processes and is managed by the System. This automation of administration provides Efficiency benefits by removing the need for manual administration and Compliance benefits through building in Consent, Privacy and Accountability functionality.

### **7.2.3 The Standard Requirements**

The Standard Requirements are the set of requirements formed by combining a privacy standard, a business security standard and a health industry standard. The Standard Requirement Set was used to evaluate the Model.

While the set was generated for the purpose of evaluating the Model, it can be of further use. The set could be useful as a tool to evaluate management systems in the health sector and, as the health sector is one of the most complex sectors in management terms, the set may well be useful in evaluating a broad range of business systems.

## **7.3 Analysis of the Model**

### **7.3.1 Model Validation**

The Model was evaluated against the Standard Requirement Set, a combined set of Legal, Business and Industry Standards – the OECD Privacy Principles, COBIT and HIPAA.

The evaluation focused on the Compliance aspects of the Model. The test for each requirement was whether or not the requirement could be implemented in a System based on the Model.

The results of the evaluation showed that the Model Complied with 35 of the 50 requirements and was conditionally Compliant with a further 14 of the requirements. *Non-Compliance* only occurred for one requirement, and then only for one aspect of that requirement – the requirement that conditional authorisations not be allowed.

It is argued that the requirement for non-conditional authorisations is not valid in environments where the *availability* of information (such as health records) takes precedence over protecting the *privacy* of the information in specified circumstances like emergencies. While the Model does allow conditional Authorisations, they are optional, and Systems based on the Model can make all Authorisations non-conditional if necessary. When this is done the System is able to Comply with all 50 of the Standard Requirements.

**With the design of appropriate Service processes and proper personnel training, a System based on the Model can comply completely with all the Standard Requirements. That System can therefore be described or rated as “Compliant”.**

The first Central Requirement of the research was to show that a Compliant system could be designed. The validation affirms that this is possible.

In practice, as with other rating systems such as the US Department of Defence’s Orange Book and the EU’s Common Criteria, business systems could be given a “compliance rating” that indicates the degree of compliance. Appropriate Systems for different industries and particular business types could then be specified through regulation or business advisories.

### **7.3.2 Comparison with Current Methodologies**

Many of the Model Concepts were derived from existing methodologies. However, the Model differs from all the existing methodologies essentially because it takes a broader multidisciplinary approach to solving the problem. Legal, Business Management and IT methodologies were considered throughout the research.

This section looks at comparing the Model with the major existing methodologies. A subsection is devoted to each of the methodologies or to a group of similar methodologies.

#### **7.3.2.1 SOA, Web Services and BPM**

Service Oriented Architecture (SOA) can be seen as providing the rationale for using Services as one of the three Component Concepts. The Services Concept was adapted in the Model definition of Service Components. SOA is the basis for a number of commercial business systems, the most successful in terms of uptake being SAP systems.

While the Model is interested in providing Concepts for use in commercial products, its focus is on providing an open source standard where global Services can be “used” by any Service-based System. Global interoperability is therefore a paramount consideration.

The idea of an open standard is to facilitate cost effective systems that can be used by organisations large and small alike. This meant that System automation was a key goal in the Model, because Small to Medium Enterprises (SMEs) do not usually possess IT expertise or the funds necessary to use current SOA systems. This is in contrast to systems like SAP that are designed for Large Enterprises (LEs).

Developments in Web Services provided the rationale for the cloud-based automated management system envisaged in the Model design. The Compliant Management System (CMS) proposed by the Model is designed to not need local IT expertise and to be cost effective by utilising standard Components rather than ones that are custom designed.

Web Services currently offer IT Services on the Internet. By contrast, the Model is designed to deliver any Service. It copes with manual and semi-automated Services like trade Services and over-the-counter Sales, as well as automated Services like Web Services. The Model is thus designed for use at a local, “offline” level, as well as for online business.

Business Process Management (BPM) offers the task-based management functionalities required for the internal business systems. The Model utilises these functionalities by treating Services as Business Tasks for the purpose of internal management. A Service is viewed and structured as a Business Task. The difference is that a Service in the Model has broader functionality than a Task. A Service has both a legal function and a commercial function, in addition to its internal IT and Business Management functions.

While security features can be built onto SOA, Web Service and BPM systems, security is not a strong consideration in their design. Their primary function is to efficiently deliver Services. It is these Efficiency principles that the Model utilises. The Model differs from all these methodologies in that Compliance is a central focus. Compliance essentially entails building the required security features into the core of the System rather than considering them an optional extra.

#### **7.3.2.2 RBAC**

Role-Based Access Control (RBAC) and its many derivatives form the basis for the adoption of Groups as the primary access control mechanism in the Model. The Model sought to find ways to overcome three nagging deficiencies in RBAC:

1. Inability to scale well to large systems due to administrative overheads,
2. Difficulties recognising Groups globally, and
3. Need for local Systems Administrators (IT Staff).

While some RBAC derivatives have found ways to address the first two deficiencies, none have to date addressed more than one.

The Model addresses all three deficiencies by enabling standard (business process) management decisions to trigger access control operations (reducing administrative overheads and alleviating the need for local System Administrators) and by giving Groups a global address.

These benefits are further explained at 7.4.2.2.

### **7.3.2.3 Clark-Wilson Model**

The Clark-Wilson Model provides the basis for building access control into Services. This is achieved by considering the task-based functions within Services as Transformation Procedures (TPs). This means that users do not access data directly but through using TP-based Services. In other words, what is controlled at the coalface is access to Services as opposed to access to data. The Model facilitates this by placing users into Groups, allocating Services to Groups and requiring individual Client-Services to be Authorised. This functionality is a key to enabling the Model to achieve Compliance.

### **7.3.2.4 Compliant Systems**

In the field of Computer Security various standards have been set (such as TCSEC and the Common Criteria) and systems have been designed to meet security requirements (such as Multics (Bishop 2003, p. 139)). Systems are also being designed to meet the requirements of design frameworks like COBIT, and legal “standards” like the OECD Privacy Principles, HIPAA and the Sarbanes-Oxley Act.

All these standards relate to how the system should perform and specify little about the methods the system should adopt in order to be compliant. The end result is that compliant system design is typically ad hoc in nature with solutions generally tailored for individual organisations.

The approach to Compliance used in the Model is entirely different. The Model defines a set of Components – Services, Groups and Service Contracts – that can be “directly regulated”. This means that rules can be expressed in terms of Services, Groups and Service Contracts. Rules can specify who can belong to a particular Group, what Services the Group is allowed to perform, and the type of Service Contract that must be used. For example, a rule could specify that a particular surgical qualification is required in order to belong to a “Surgeons” Group, that the “Surgeons” Group is allowed to perform the “Brain Surgery” Service, and that a specified written Service Contract be used.

The System can directly implement such rules without any interpretation being required. This is essentially because the rules are expressed in logical terms that the

System can understand. The dissertation proposes that Regulations be framed in a logical way relating to Services, Groups and Service Contracts. This opens the way for Regulation not only to be specified in written terms but ultimately in machine readable form where Systems can be automatically updated when Regulations are put in place.

#### **7.3.2.5 Service Contracts**

While Service Contracts have long been recognised by the law, the idea of them applying to Services provided on or by computers is more recent. Some SOA systems do use Service Contracts, and so the idea of Digital Service Contracts is not new. However, this research takes the idea of Service Contracts to a step further, at least from an IT and Business Management perspective if not from a Legal one.

The Model places the Service Contract at its centre in functional terms. That is, Service performance is guided and controlled by Service (Contract) Based Authorisations. This means that business decisions – expressed through Authorisations – and the Services that are performed are subject to the requirements of Service Contracts. As the system runs the business, Organisational Compliance is attained by ensuring Service Contracts are mandatorily enforced by the management system. It could be said that **the Model achieves compliance through mandatorily enforced Digital Service Contracts.**

#### **7.3.2.6 Organisational Security Policies**

If organisations have explicit Security Policies, they are in written form. It is then a matter of training and monitoring personnel to ensure that the policies are followed. This is problematic as security depends on the degree of diligence of organisational personnel, individually and collectively.

The Model sees security policies in terms of Services that must be performed by the Organisation. As with all Services, the CMS is charged with managing all *Security Services*. In designing a Security Service, Security Policy is essentially being built into the System. Once this is done and the System controls the Security Services, written Security Policies become somewhat redundant, and are essentially only needed as a source of information or a training tool for personnel.

Security Services can cover all areas of security, from the security of the IT system itself to the security of all the resources of the organisation. The scope of coverage depends on the level of CMS control. For example, if there was a Security Policy to “lock all access doors after hours”, the CMS may require security guards to verify that they have indeed locked the doors, it may automatically check that the doors are locked using remote sensors, or it may itself lock the doors using a remote locking system. What is important to note is that in all cases the System is managing and/or controlling the security operation (that is, the performance of the Security Service); the Security Policies are thus built into the System.

#### **7.3.2.7 Privacy Policies**

Privacy Policies are a type of Security Policy that relate to Information, in particular, Personally Identified information (PID) and sensitive organisational information. As with other Security Policies, they are traditionally kept in written form, whether stored on paper or online.

Again the Model aims to build these policies into the System, making them enforceable. It does so by restricting access to information using Groups and Services – requiring Group membership and proper Authorisation to access Services and designing Services so they only access the information that is necessary for their performance. In comparison with written Privacy Policies, which are notoriously difficult to enforce due to information availability and accessibility, built in Privacy Policies can be enforced automatically.

It is one thing to have robust processes to restrict access; it is another to ensure that individuals do not gain illegitimate access. Breaching a System that uses Service Based Privacy requires individuals to either obtain illegitimate Authorisations from Managers or misuse their Organisational role to perform Services that they can themselves Authorise (where Separation of Duties is not enforced). To be totally effective the System therefore requires monitoring and auditing processes to detect possible breaches. These include checking access profiles in order to find possible anomalies.

Effective monitoring and auditing processes that are transparent to users act as a strong disincentive to improper use. This is further enhanced if retribution, including prosecution in extreme cases, is forthcoming. At present it appears that many privacy breaches remain undetected, and that those that are detected rarely result in prosecution due to lack of traceability or regulatory ineffectiveness.

#### **7.3.2.8 Accounting Systems**

The Model was not originally designed with accounting in mind, but when it became evident that a Service-based system was envisaged, it likewise became obvious that an accounting system based on Services was appropriate. Automated accounting systems already exist, so adding accounting Services to the Model was elementary. Since the Model views all work done in an Organisation in terms of Services, *Accounting Services* associated with each business Service could be used to track all transactions.

Accounting systems are logical in nature and have their basis in mathematics. This is the same functionality that exists in IT systems, so adding accounting to an IT system is a relatively simple and well established process.

It is noteworthy that compliance with regard to accounting is a standard organisational requirement. There was accordingly little difficulty fitting Service Based Accountability into the System.

#### **7.3.2.9 IT Standardisation**

In order to facilitate interoperability between systems, especially on a global level, some form of standardisation is necessary. There are three general methods by which standardisation occurs: by “planning” in an administrative sense, by “evolution” in a commercial sense and by “design” in a scientific sense.

“Administrative Standardisation” involves getting interested parties to discuss and agree on what the standards should be, usually before any substantial development begins. Where the issues at hand are complex or there are many parties with different ideas, agreement can be difficult to reach, be very time consuming, and often compromises can make the solution less than ideal.



“Commercial Standardisation” involves market forces making a product so popular that it becomes a de facto standard. It is often the case that a number of products will set such standards. An example is the dominance of the PC and Mac computers. Other companies design software and hardware products to work with the established “standards”. Compared to Administrative Standardisation, Commercial Standardisation usually occurs more quickly and lasts for a shorter period (until something better comes along). This is because there is far less need for negotiation and compromise in the design of commercial products.

“Scientific Standardisation” involves a standard being proposed that is freely available to anyone that wishes to take advantage of it. It involves a model, system or product being established as an “open source” resource.

The Model seeks to provide the framework for an open standard that can be utilised both for non-profit and commercial purposes. In this sense both the Scientific Standardisation and the Commercial Standardisation may be appropriate. A variety of Compliant Management Systems can be created based on the Model and, through the use of standardised Services, Groups and Service Contracts, the Systems should be interoperable.

### **7.3.3 Limitations of the Model**

#### **7.3.3.1 Generality**

The Model ultimately seeks to set up a global framework for doing e-business or, more specifically, facilitate the trading of Services. It seeks to incorporate all types of Services from over-the-counter sales to online transactions. To cover such a broad area and range of activities the Model needs to be general in nature. That is, it needs to provide a general framework upon which specific local, industry and organisational functionalities can easily be included.

The central focus is to apply a universal system based on Digital Service Contracts (DSCs) that are globally accepted. This requires enough simplicity and flexibility to allow national jurisdictions to recognise the legitimacy of DSCs.

While the Model aims at achieving Global Functionality, *that* is not its sole purpose. It also seeks to provide Compliant Management Systems for industry sectors and individual organisations. At an Industry level the Model provides the basis for interoperable systems, for example, it could be the basic for a State or National Health Records Management System. At an Organisational level the Model provides both Compliance and Efficiency benefits that, on their own, can justify its use.

Usefulness to individual Organisations is seen as the fundamental means of achieving interoperability on both an Industry and a Global level. While it is possible for jurisdictions to ultimately mandate the use of compliant systems for organisations, there remains a need that such systems be efficient. The idea is to create Compliant Management Systems that are popular for their inherent usefulness (like Microsoft Word) and then facilitate interoperability as an “added benefit” (like the sharing of Word files).

#### **7.3.3.2 Model Maturity**

Due to the generality required, the Model has been developed after researching a broad range of methodologies in diverse fields primarily including the IT, Business Management and Legal fields. While the research was broad, there may be some other useful methodologies that were not investigated. Also, the Model will no doubt have to address future developments to remain relevant over time.

While the Model is designed to facilitate the development of one or more Compliant Management Systems, this is Future Work (see Figure 1) that has not yet been done. The development of CMSs will provide useful feedback and highlight instances where the Model can be modified or extended.

With these considerations in mind, the Model could be described as in its infancy. That is not to say that it lacks any particular thing. Considering that the verification results were encouraging, it merely means that additional work on the Model will likely be useful.

### **7.3.3.3 Authorisation not Authentication**

As mentioned previously, Authorisations are the tool by which the Model Components are controlled and the means for ensuring Consent, Privacy and Accountability compliance. Authorisations are the glue that binds all the General Concepts together.

At its core, Authorisation has to do with controlling what system users are allowed to do. In the case of the CMS this means controlling access to Services. Put simply, it is about controlling “who can do what”.

Authorisation is not concerned about identifying the user (the “who”), only what the (legitimate) user is allowed to do. Proving “who the user is” is another matter. “Authentication” is the term used to describe the process of establishing a user’s true Identity. This research is only about Authorisation, not Authentication. However, a global Authentication system is envisaged, and is described in the Future Work section (see 7.5.5).

## **7.4 Application of the Model**

The following three subsections discuss the real world implications of the Model in terms of the three Central Requirements, Compliance, Efficiency and Global Functionality.

### **7.4.1 Regulatory Compliance**

The aim of this research was to provide a Model that provides the foundation for the design of compliant organisational Business Management Systems. This means that the core issue is the regulation of IT.

Before discussing the issue of regulating IT, the meaning of Compliance needs to be elaborated. This is achieved by defining *Levels of Compliance*.

#### 7.4.1.1 Defining Levels of Compliance

Compliance can be considered to have three aspects relating to *Levels of Standardisation*, *Levels of Scope* and *Levels of Coverage*.

*Levels of Standardization* relate to the type of system, with respect to compliance, that is employed by an organisation. They can be described as follows, from simplest to the most complex:

1. Tailored System,
2. Un-Rated Standard System,
3. Rated Standard System,
4. Mandatory System.

A Tailored System is one that is purpose designed for the organisation. It is simply a one-off system that is designed to be compliant. The design is based on the Model and incorporates the Model Concepts.

An Un-rated Standard System is one that is based on a standard compliant design and has been configured for the organisation in question. Such a system would most likely be based on either an open source compliant system or a commercial compliant system (see 7.4.2.1).

A Rated Standard System is the same as above, except that the system has been tested in relation to meeting an accepted regulatory standard and been given a compliance rating. Part of the reason for an organisation to adopt such a system would be that it inherently complies with regulation. The onus rests on the system suppliers to ensure that it continues to remain compliant as regulations change.

A Mandatory System is one that has been proven to be compliant, to the satisfaction of applicable regulators. In such cases regulation would be framed to mandate that organisational systems meet compliance requirements within the jurisdiction. In other words, system requirements are explicitly specified by regulation.

*Levels of Scope* relate to the size of the business collection to which compliance is applied or regulated. They can be described as follows, from a single local entity to a sector of related industries:

- A. Entity/Division,
- B. Organisation,
- C. Industry,
- D. Sector.

An entity or division is a single locally managed unit. Such a unit directs the work that it performs and has some responsibility to maintain a compliant system of its own. Units include locally based SMEs as well as autonomous divisions of larger organisations.

An Organisation in this sense is one that is made up of more than one local division.

An Industry is a collection of Organisations that perform similar functions, that is, ones that are in the same Industry.

A Sector is a collection of related Industries.

*Levels of Coverage* relate to the jurisdiction to which compliance is applied or regulated. They can be described as follows, from local to global:

- i. Local
- ii. Regional,
- iii. National,
- iv. Global.

These terms are self-explanatory and, while they generally refer to governmental or jurisdictional divisions, the Local and Regional terms may just be geographical.

The overall Level of Compliance can be expressed in terms of the *Levels of Standardisation, Scope and Coverage*. For example, compliance can be specified to be Mandatory for Organisations Nationally, meaning that National regulations would specify that Organisations must use specified compliant systems. If required, this could be denoted as “4Biii” compliance (taking the appropriate label for each of the three levels).

#### **7.4.1.2 Applicability of the Model**

The Model aims to be applicable to any Level of Compliance. First of all it enables organisations to be compliant themselves. It then seeks to enable other organisations to meet the same standards because, when two organisations wish to do business with one another, problems can be caused if they do not meet the same compliance standards. To address this problem the Model provides a mechanism for standardization through the acceptance of Digital Service Contracts. This supplies regulators and commercial software companies with the means to effect standardization on a larger scale.

Compliance at an organisational level is the first priority of the Model, and achieving this is arguably a valid goal in itself. However, achieving compliance on a broader scale is of obvious benefit. This is synonymous to the environmental idea “think globally, act locally”.

#### **7.4.1.3 Legal and Logical Reasoning**

Regulation has a tendency to trail behind technological advances in the IT field, particularly in the area of online IT. Regulatory requirements like the Sarbanes-Oxley Act in the US and various Privacy Laws have sought to bring some accountability to the use of IT in organisational environments. Such Laws are typically about (digital) information and specify how information should be handled and how records should be kept.

IT-based business systems tend to be designed according to business requirements where the focus is on providing the appropriate functionality rather than on appropriate information use and security. In IT terms information is just *data* that is

input to or output from functional programs. Any thought of information security and appropriate information use is generally an afterthought.

The managers that make compliance-related decisions in organisations are faced with the task of having to add functionality to their IT systems in order to keep them compliant with written regulations. This can impose significant costs on organisations because adding functionality to existing systems is usually complex.

This dissertation argues that much of the complexity is caused by the fact that regulations are framed as legal principles whereas software is written according to logical principles, and that this fundamental difference in design methodologies causes many problems. For example, Privacy Laws state that information should be used for an appropriate *purpose*, but purpose is not inherently a logical thing and computers do not ask *why* a particular process is to be performed.

Problems of this kind, the dissertation maintains, can be addressed by finding the common ground between legal and logical principles and then designing both regulation and software based on common (ground) principles. The common ground chosen is based on the idea that “Organisational work entails Groups of persons providing Services to Clients according to Service Contracts.” The idea is to design regulation and software around the 5 Ws – *who* (which Group) can do *what* (which Service) and *when* and *where* they can do it. The question of *why* it is being done is answered by the idea that “the Client (inherently) *requests* the Service when they Consent to receive it”.<sup>68</sup> In other words, the reason *why* an Organisation performs a Service for a Client is that the Client *requested* it. The *purpose* for information use is then (inherently) “to perform the requested Service”.

In summary, **this dissertation is about how to regulate the public use of IT and how to design IT systems that can be regulated.**

---

<sup>68</sup> The Consent (which must be Informed Consent) implies that the Client accepts the terms of the Service Contract offered by the Organisation.

#### **7.4.1.4 Why do IT Systems Need to be Regulated?**

An IT system can be viewed as a piece of technology that performs certain tasks – as a tool that can be used by humans. An automobile can be viewed in the same way. When the automobile was invented, the thought of regulating its use was probably far from anyone’s mind. However, with the increase in usage of automobiles and the inherent dangers they posed, there emerged a need to regulate their use. Nowadays thousands of regulations cover road design, traffic rules, vehicle design, licensing and multiple other areas. These regulations serve at least two purposes: to maintain public safety and to enable reasonable travel times.

Computers have only been in common use by consumers for a couple of decades. Regulating their use is now becoming a necessity because of the damage that can be caused to people. As with automobiles, the perceived need to regulate comes when usage poses a public threat and computer use is now prevalent. This dissertation argues that it is now time to properly regulate the use of IT so that IT systems are safe and fast (that is, compliant and efficient). It argues that IT systems need to conform to certain standards if they are to be widely used for important actions that have real world consequences. This means that certain fundamentals must be built into systems in order to enable regulation; that IT regulations need to be expressed in logical terms; and that persons should be “licensed” to use computers for certain activities.

This necessarily raises the question of whether all IT use be regulated. The answer is no. Automobile usage again provides a useful analogy. The public use of automobiles is regulated closely, but their private use is largely unregulated. Persons can build their own vehicle and drive it on their own property, largely unregulated (except for noise etc). This dissertation proposes a similar regulation of IT. Public use should be regulated where there is potential for personal damage to citizens.

The “public IT domain” largely means the business world. The “private IT world” largely means home systems and online communications between home system users. There is a line between the two where private activities become public activities.



One way of distinguishing private from public is in terms of online services and whether persons are profiting from providing them. Think of a person making a batch of jam at their home from their own fruit trees. It is a private activity to give the jam away. It can be a private activity to be recompensed in some way for the cost of making the jam. The jam could be sold at a School Fete to raise money for the School. This can still be considered a non-profit activity. However, it all changes when the jam is sold for profit; regulations now apply. These regulations have to do with business registration, taxation, food standards and various other things.

Another way on distinguishing private from public is whether the online services involve persons known to each other in real life. It is argued that where services, such as social networks, are open to public membership then they should be regulated in some way.

As with the imposition of regulation in general, the regulation of IT requires a balanced approach. The internet has provided a platform for individuals to freely express their views. These freedoms should be rigorously protected. However, there is also a need to regulate in order to protect at-risk individuals. The dissertation only advocates regulation of organisational management systems. This could be described as supporting “responsible behavior in the public IT domain” while allowing “freedom in the private IT domain”. Such a distinction is required in order to understand the scope of IT based regulation.

#### **7.4.1.5 Regulating Business on the Internet**

The public use of automobiles is regulated in two basic ways, by:

1. Requiring them to conform to design and usage standards, and
2. Licensing users through registering their Identities.

This dissertation proposes a similar approach to regulating “public use” of the Internet. “Private use” would remain unregulated. First, a “Business Web” that utilized Compliant Systems (such as the CMS) would provide conformity to design

and usage standards. Secondly, a “Web of Identities” would enable users to be “licensed”.

While the development of this Web-based approach is a part of Future Work (see 7.5.4 and 7.5.5), the description serves to illustrate how Compliance may be achieved on a global scale.

The Internet is the global network of computers connected together by wires and wireless means. The World Wide Web (the Web) is a system of interlinked documents that are stored and accessed through the Internet. The Web uses the Internet in much the same way as an application uses a computer. A computer can host many applications and the Internet can host many “Webs”.

The Web can be described as a “Web of Information” (or Data). Concepts for a “Web of Services” (W3C 2011), a “Web of Things” (or Devices) (Guinard & Trifa 2009) and a “Web of Identities” (Korth 2009) have also been proposed. Together, these four Web concepts embody the entities that are involved in the provision of Internet (Cloud-based) Services.

For persons (or “Users”) to interact with an IT system, including a Web-based system, they are given an “IT Identity”.<sup>69</sup> The term “Web Identity” will be used hereafter to describe an IT Identity in a Cloud-based (or Web-based) system.

To do business on a regulated Business Web Organisations (as Providers) must have a registered Identity. Clients would also have registered Identities but, depending on the nature of their transactions, they may be able to use “Anonymous Identities”.<sup>70</sup>

The dissertation advocates a Service-based approach to facilitate the design of Compliant Systems and a Group-based approach to license users (by enabling the registration of their Identities). A regulatory framework is created by describing all Organisational work in terms of Services; by mandating the use of Compliant

---

<sup>69</sup> A Username is an example of an IT Identity. A Business System IT Identity may be an Account Name or Number.

<sup>70</sup> An Anonymous Identity is a user specified Identifier that is not formally linked to an actual person or their PID. This is a form of “self-registration”.

Service Contracts; and by representing the parties involved in Service Delivery by Groups that have registered Identities.

As the Model utilizes a Cloud-based approach, these issues need to be considered in the context of a Cloud-based system. While the following can be considered as Future Work, it sheds light on the issue of global IT regulation.

To *provide* Business Services, a *Provider* with a Web Identity uses a Device to host a Service that interacts with Information. To *use* a Business Service, a Client with a Web Identity accesses the Device to receive the Service that interacts with Information. Figure 43 shows a Provider (using Device P-Device) with registered Web Identity A and a Client (using Device C-Device) with registered Web Identity B accessing a Cloud-based system.<sup>71</sup>

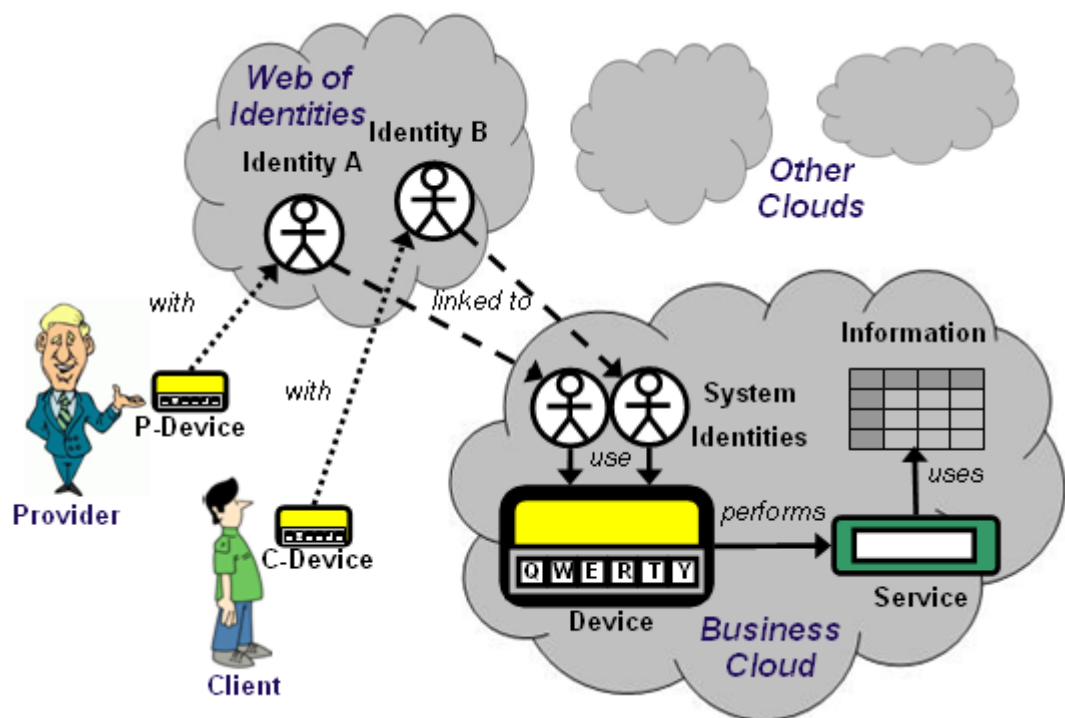


Figure 43: Webs and Clouds

<sup>71</sup> Any or all of the items shown in the Business Cloud could be located within the Providers own Organisational Network – the processes remain the same.

### **7.4.2 Business Efficiency**

The Central Requirement for Efficiency relates to the usability, effectiveness and cost of business management systems based on the Model. As no System was developed for the research (it is Future Work) there was no testing for Efficiency. In order to justify the claim that the Model provides Efficiency benefits the following arguments should be considered.

#### **7.4.2.1 Compliance Costs**

The Model Validation showed that Systems based on the Model (CMSs) can be designed to be Compliant. It is argued that an Organisation that utilises a Compliant System will reap tangible ongoing benefits, including reduced Compliance costs, reduction in procedural breaches and errors, and decreased legal costs.

Compliance costs come in two main forms: costs associated with implementing Compliant business processes, and costs associated with proving that the Organisation is Compliant. When a Compliant CMS is utilised there are no implementation costs as CMS processes are already Compliant. If the System has been rated as Compliant, it only needs to be demonstrated that the CMS was used for all Organisational Services.

As the CMS utilises generic Services that can incorporate “best practice”, and checks and balances are built into the System, the likelihood of procedural breaches and errors is reduced (compared to non-compliant systems). Procedural breaches and errors take time to rectify and can lead to litigation against the Organisation resulting in legal costs and compensation payouts.

While reducing procedural breaches and errors leads to Efficiency gains and reduction in legal costs, utilising a Compliant system where Compliance can be proven produces a secondary benefit – the reduction in the incidence of erroneous claims. The Service Records produced by a CMS can provide proof that proper procedures were followed (such as that legal Consent was obtained) and that alleged errors did not occur. Litigation may be avoided by revealing such proof to a potential litigant before they proceed.

#### 7.4.2.2 Reduced Administration

Traditional Access Control Models based on Mandatory Access Control (MAC) place the responsibility for deciding *who* (which user) accesses *what* (which data) in the hands of system administrators. This can work in situations when the system administrators have a good knowledge of all the users and the nature of the data. This is, however, rarely the case since the numbers of users and the amount of data has increased exponentially. In most cases, due to their many other responsibilities, system administrators have little time to attain the knowledge of users and data that is necessary for properly informed decisions to be made. This is evident through the many attempts to reformulate MAC based methodologies through the likes of Role Based Access Control (RBAC) (see 3.3.5), the obvious increases in remoteness of users from system administrators, and the increased amount of data that needs to be protected.

The problem of trying to place access control decision making into the hands of personnel who possess the appropriate knowledge of users and data has been addressed in some current methodologies by the technique of delegating administrative privileges to personnel closer to coalface. There remains a problem with this course, however, namely the fact that the delegated personnel require an intimate knowledge of how the access control system works. The training and supervision requirements for doing this properly rarely make it practicable.

**The Model overcomes these problems by changing the nature of access control decision making and by automating the process.** Figure 44 compares MAC, RBAC and the Model. In MAC based methodologies, access to data (objects) is controlled by allocating privileges (access control “rights”) to individual users (subjects). This is a “one step process”. In RBAC based methodologies, the privileges are given to role based groups (that is, groups of individual users) and users are allocated to Groups.<sup>72</sup> This is a “two step process”.

---

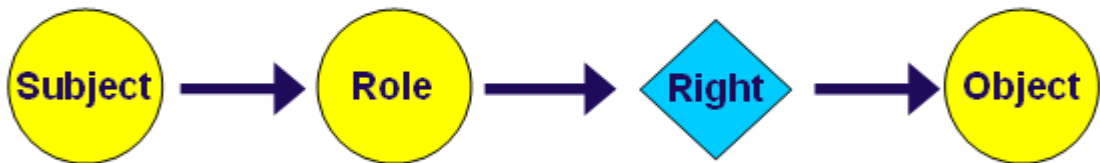
<sup>72</sup> In this sense RBAC based methodologies are a subset of MAC based methodologies.

In the Model, access to data is controlled by allocating privileges to Services, allocating Services to Groups, and then allocating users to Groups. This is a “three step process”.

**Basic Access Triple (MAC):**



**Role-Based Access Control (RBAC):**



**The Model:**



**Figure 44: Access Control Model Comparisons**

The one step MAC process is simple but requires an unwieldy amount of administration because privileges must be (manually) assigned for every subject/object pair where access is required. RBAC simplifies this by limiting the assignment of privileges to roles, there normally being fewer roles than users. However, when fine-grained access control is necessary, the number of roles can blow out and even approximate or exceed the number of users (as users can have multiple roles). For example, in Oracle database systems a system administrator can be assigned many different database management roles, each to facilitate a particular function. This means that RBAC systems, too, often require inordinate amounts of (manual) administration to facilitate proper (fine-grained) access control.

In both MAC and RBAC system administrators (or their delegates) require knowledge of the system, user roles and data objects. The Model changes the nature of access control because it allows management by personnel other than trained system administrators. Each of the three steps in the Model process can be performed by non-IT personnel.

The first step is the assignment of users to Groups.<sup>73</sup> Personnel Managers can assign users to Groups under their responsibility (relatively static Groups), such as a Group representing qualifications or the Organisational department that the user is in. Managers closer to the coalface (that is, ‘operational managers’) can assign users to Groups such as Project Teams, Client Service Teams and Shift Teams. These assignments are part of the normal work of Managers. The Model captures these assignments and automatically sets up the access control privileges without requiring the Managers to perform any Access Control Tasks.

The second step is the assignment of Services to Groups. In this case “professional knowledge” of what a particular Group is allowed to do is only required where non-generic Services exist. The majority of these assignments can therefore be done automatically by utilising generic Groups and Services. Where professional knowledge is required, the assignment task can be passed to the appropriate Manager, who does not require system knowledge, only knowledge of the Group’s function and the nature of the Services in question.

The third step is the assignment of privileges to Services. Again most of this can be done automatically through the use of generic Services and Data Storage techniques. Where professional knowledge is required, it relates to policy knowledge of the appropriate data necessary to perform a particular Service.

The net result of utilising the professional decisions of non-IT personnel to automate access control actions is that the cost of access control administration is greatly reduced. This amounts to substantial Efficiency gains.

#### **7.4.2.3 Generic Components and Best Practice**

The use of generic Services, Groups and Contracts reduces the need to tailor solutions for organisations. Generic components are more cost effective than tailored solutions because multiple users effectively share in the cost of producing and maintaining generic components.

---

<sup>73</sup> Groups in the Model are generic in nature (for example, they represent personnel qualifications) and much fewer in number compared to RBAC, thus making them comprehensible by personnel without system knowledge.

Some may argue that using generic components makes organisations the same, and that diversity should be maintained by opposing such standardising practices. However, the use of generic components should be seen as a pursuit of best practice. The pursuit of best practice has a lot to do with Efficiency and employing standards that facilitate Efficiency gains for all organisations. Using generic components does not necessarily mean that businesses lose market differentiation and become less distinct. Rather, repetitive and menial tasks are automated, which can in turn enable resources to be freed for creative purposes. Businesses by nature tend to be creators of trends, followers of trends, or a mixture of both. Market differentiation and pursuing best practice are related but separate issues – both are required to some degree to ensure commercial success.

#### **7.4.2.4 Access Control Granularity**

The primary benefit of the Model over current Access Control Models like RBAC in access control terms is the fine granularity that is enabled. In fact, the granularity can be as fine as in MAC because the users are allocated to single Client-Services. This means that only a user performing a Service for a Client has access to the Client's data and then only to that data necessary for the performance of the Client-Service.

The granularity gain over RBAC based models is two-fold. The Model's three-step access control process inherently allows for finer granularity because Services only access the data that they require, whereas RBAC gives access to all data that *may* be required.

The second granularity gain is substantial and potentially game-changing because the Model restricts access to only the personnel who are in the process of performing a required Service for a Client. RBAC, in contrast, normally allows access to all client records relevant to the role involved. Users can therefore examine the records of any client, not just the clients they are directly servicing. For example, a Nurse could access the records of a neighbour, friend or relative even if they are not currently being treated at the hospital.



Restricting access to Clients being Serviced is enabled through Partial Permissions (de la Motte & Hartnett 2005a). Partial Permissions mean that memberships of multiple Groups can be required before access to Services is allowed. For example, to perform a Client-Service, a Nurse must be a member of both the “Nurse” Group (to perform a Nursing Service assigned to the Group) and the Client Service Team.

Partial Permissions enable two distinct advantages: complex rule formulation and more efficient Group Management. Partial Permissions enable complex rules to be formulated using “AND”, “OR” and “NOT” relationships to be described. For example, access may require membership of Group A AND (Group B OR Group C) AND NOT Group D.<sup>74</sup> As complex rules involving multiple Group memberships can be applied, the need for specific RBAC style role Groups for special purposes is allayed, meaning that far fewer Organisational Groups need to be defined.

As rules tend to be relatively static and Group memberships relatively dynamic, Group Management takes more time than *Rule Management*. Group Management is simplified and made more efficient by having fewer Groups to consider. While the Model requires more complex rules than RBAC based methodologies, requiring greater Rule Management, this is far outweighed by the Group Management efficiencies gained by having fewer Groups. In any case, generic Rules can be utilised when generic Groups and Services are utilised, meaning little or no Rule Management is required.

### 7.4.3 Global Functionality

The Model treats each Organisation as a single entity that is able to deal both locally or online. In order to facilitate this, a cloud based approach to system functionality is primarily considered. However, this in no way precludes the Model from being applied to Systems that only use local IT resources. The Model is designed to produce Compliance and Efficiency benefits as well as Global Functionality benefits.

---

<sup>74</sup> The “NOT” relationship enables negative permissions to be enforced. This means that members of a specified Group (for example, a “Sex Offenders” Group) can be used to excluded persons from performing specified Services or becoming members of specified Groups.

The Global Functionality of the Model can therefore be considered as optional or ancillary to the other aspects of the Model.

#### **7.4.3.1 Unified Approach**

In the past there has been a tendency for local businesses to be different entities to online businesses. For example, a consumer can purchase a book from a local bookseller or from an online bookstore. While some businesses trade both locally and online, often the local and online parts of the business are seen as separate.

It is argued here that the “separation approach” is not the most efficient way to do business. It should be more efficient if both parts of the business use the same systems – the same ordering, accounting and personnel management systems, for example. This is termed the “unified approach”.

While there are some differences in payment, customer service and delivery options between local and online businesses, the core business processes are the same and should be integrated. In any case, as technology advances, the divide between local and online Services will tend to blur. For example, a consumer may order a product online and then pick it up and pay for it at a local store, order and pay for a restaurant meal online while in the restaurant, or purchase a local residential property online.

#### **7.4.3.2 Futuristic Approach**

Since the advent of the Internet there have been steady and substantial increases in connectivity in term of the number of users and computers connected and the speed and capacity of the data links in the network. In the past security on the Internet has often been a low priority, mainly due to the costs, network overheads and general inconvenience. While the percentage of online business transactions (as opposed to local transactions) has increased dramatically, there is a long way to go to get to the point where all transactions can potentially be done online if desired. The hurdle to getting to this point is the *security* and efficiency of online transactions.

Interfaces on modern smart-phones and the like are enabling new efficient ways of performing online transactions. For example, consider the technology that allows users to swipe their smart-phones at terminals to make small payments.

The automation of B2B transactions is also a consideration. This involves automated interactions between business management systems to enable such things as the automated ordering, payment and accounting for stock.

In order to retain its relevance, the Model is designed to facilitate all forms of transactions, including automated online transactions. While the Model allows for the development of efficient transaction related Services, it focuses on the problem of efficiently securing online transactions. This is because the security problem is the most intractable. While some online transactions do not require much security to be attractive to consumers, this is not the usual case. It is argued here that a high level of security is needed before customers will be happy to perform all transactions online.

#### **7.4.3.3 Digital Service Contracts – the Means for Online Regulation**

The Model aims to achieve this high level of security through the mechanism of globally enforceable Digital Service Contracts (DSCs). The proposed mechanism uses a two-fold approach: by defining methodologies for universal DSCs, and a globally regulatory system for enforcing them.

DSCs allow an Organisation's system to be Compliant with Regulation. The Authorisation mechanisms defined in Systems based on the Model ensure that the Organisation fulfils its Contractual obligations and can show that it has done so. By utilising Compliant Systems customers can be assured that Services provided by the Organisation will be of a high standard. Not only do DSCs provide a way for Organisations to ensure they are Compliant and for customers to be assured regarding Service standards, they provide a mechanism for enabling the regulation and enforcement of online Service Contracts.

Regulators can enhance the use of DSCs in three ways: they can *Recognise DSC Legitimacy* in their jurisdiction, *Provide DSC Enforcement Mechanisms* (including for Contracts involving foreign parties), and *Mandate Compliant Systems* use in organisations.

To *Recognise DSC Legitimacy* regulators must take steps to ensure that DSCs can be used as evidence in Contract Litigation. Ideally this would require legislation that

would specify DSC requirements or, more generally, what constitutes acceptable digital evidence.

To *Provide DSC Enforcement Mechanisms* regulators must find an appropriate and cost effective way for parties to seek legal redress and receive compensation. While current mechanisms may be able to deal with local disputes where parties can attend court or mediation sittings, these may not be suitable for parties that are located outside the jurisdiction. Such mechanisms could include accepting representations by video conferencing or providing/accepting local legal representatives for foreign parties. To some degree these mechanisms already exist within the legal process. As DSCs can cover small as well as large claims, a number of different mechanisms may be appropriate.

To *Mandate Compliant Systems* regulators must first be provided with evidence that such systems exist and work satisfactorily. This means that such systems must be developed and proved. The Model provides the methodology for their development.

#### **7.4.3.4 Assisting Online Regulation**

The Model potentially provides three things to help regulators establish standards for Online Services: Compliant Management Systems that use and provide DSCs, *Online Component Repositories* that can be used to quickly establish these systems, and Global Groups that can be used to register Identities.

Compliant Management Systems can be developed that are based on the Model. The next step in Future Work (see 7.5.1) is to develop such a System (that is, a CMS).

Online Component Repositories can be provided to enable Organisations to establish Compliant Systems or, if not entire business management systems, partial systems that can be used for online Services. Regulators could potentially provide all registered Organisations in their jurisdiction with basic online entities that could be accessed by consumers. This could involve providing business registration details in the form of an information page to the public for each Organisation. Details could include the type of business, the owner's name, the address of the business, and its contact details. By incorporating some form of complaint mechanism, authorities

and the public could be alerted to businesses and individuals engaging in unlawful or unprofessional practices.

The Model utilises Global Groups, each with a Global Group Address. The Model envisages that Global Groups can be used as Identities, namely that individuals, Organisations and Organisational groups are viewed as Identities by users. Technically they are User Groups as defined in the Model, but they can be seen as Identities by users because they represent persons and Groups of persons. Groups Hierarchies are formed through Groups being members of other Groups. As each Group has a Global Address, the global hierarchy of Groups form a “Web of Groups”. As the term Web of Groups has little meaning to users, it can be replaced by the term “Web of Identities” where the Identities are the Groups.

The reasons for establishing a *Web of Identities* are twofold: to provide persons with online Identities that can be used for doing business, and to enable these Identities to be registered to actual persons.

## **7.5 Future Work**

The following subsections summarise the main possibilities for future work that stem from this research.

### **7.5.1 Compliant Management System**

The Model provides the basis for a Compliant Management System (CMS). As explained in the dissertation Introduction (see 1.1), the development of a CMS is Future Work. The complexity of this development depends on the level of Compliance that is required. It is envisaged that a Local Un-rated Entity-based system that could be adapted to enable standardization and system interoperability would initially be developed.

### **7.5.2 Service Based Regulation**

One of the primary outcomes of this research from a legal perspective is the idea that regulations should be written in terms of Services. How this should be done in order to achieve workable legal outcomes needs to be further explored.

What must first be investigated in this regard is the scope and suitable wording of written Service Based Regulations. The possibilities for having “digital regulation” could also be explored. The idea here is that IT systems could automatically interact with such regulations so that they were immediately enforced. For example, rather than specifying tax thresholds and rates in written form, they could be stored digitally in an accepted form and automatically downloaded to Business and Accounting IT systems.

### **7.5.3 Privacy of Personal Records**

The initial inspiration for this research was to explore ways of efficiently maintaining the privacy of medical records by taking patient consent into account. While the Model is a general solution that incorporates privacy and consent solutions, it still meets the medical record requirements that were initially evident. There remains work to be done in developing software that applies the Model to medical records and, more generally, to other situations where privacy and consent are important organisational concerns.

### **7.5.4 Business Web**

The development of a CMS that can be used by multiple organisations is the springboard for the development of a Business Web. The Business Web would provide standard features that would streamline things like Service Discovery, Service Management, Supply Chain Management and Business Transactions.

### **7.5.5 Web of Identities**

The Web of Identities is a collection of Web Identities, each of which is authenticated, has a Web address, and is attached to a Global Group.

Authentication processes would be developed to link Web Identities to real persons, groups of persons and organisations. These processes provide mechanisms for registering users so that parties to online transactions are known and traceable. This provides a higher level of security in online transactions since parties are dissuaded from fraudulent behavior because they can no longer remain anonymous or easily fake identities. A variety of processes could be used, from self-registration through to strict governmental registration that would provide stringent Identity checking procedures.

Registered Web Identities would be given a Web Address that is linked to a Global Group on the Business Web. This ties real Identities to Organisational Groups on the Web and facilitates transactions security.

#### **7.5.6 Supply Chain Management**

While this was referred to in the Business Web subsection (see 7.5.4), Supply Chain Management using CMSs is a research area in itself that warrants further investigation. The availability of standardized CMSs to organisations in supply chains opens up numerous possibilities for the automation of Supply Chain Management.

#### **7.5.7 Component Repositories**

Investigation of how to efficiently establish and manage component repositories relates closely to the development of the Business Web, although the two research areas are fairly independent. While the Business Web can operate without component reuse by using supplied components, its functionality could be considerably expanded by incorporating open component repositories. Another reason why component repositories should be considered separately is the existence of considerable existing research in the area, including research on ontologies, that needs to be considered.

## 7.6 Conclusion

The research produced a Model and set of Model Concepts that provide the basis for the design of Compliant Business Systems. It took a broad view covering Legal, Business Management and IT fields. The Model proposes three basic design components: Services, which represent commercial units of work, Groups, which represent collections of IT Identities, and Service Contracts, that embody the legal aspects of Service Delivery (Privacy, Consent and Accountability). Compliance is achieved by systematically acquiring appropriate Authorisations from each of the Service Contract parties – Managers, Workers and Clients.

A number of new ideas (Concepts) are proposed in the dissertation. They include Contract Based Services, Validation by Authorisation, Service Based Purpose, Service Cells, Group Based Services, Global Group Addressing, Partial Permissions, Compliant Systems, Group Based Organisations, (Individual) Service Teams, Contextual Groups and Service Based Information Provision.

The Model was validated against three standards – the OECD Privacy Principles, COBIT and HIPAA. The Model complied with 35 of the 50 requirements and was conditionally compliant with a further 14 of the requirements. It was *non-compliant* with the single remaining requirement only because conditional authorisations were considered to be appropriate in emergencies. It is argued that in these cases *availability* of information should take precedence over *privacy* protection. If emergency access is disabled, comprehensive management Services incorporated and appropriate personnel training is employed, a CMS can be totally Compliant. This fulfils the major requirement of the research, the First Central Requirement of Compliance.

Arguments were made to show that the Model was also Efficient (reducing compliancy, administration and software costs while providing better access control granularity) and Globally Functional (enabling interoperability and online regulation through the use of Digital Service Contracts), the second and third Central Requirements. It was argued that Compliance was the key breakthrough in the research, as most of the necessary concepts for Efficiency and Global Functionality were already known and required little more than appropriate adaptation.



With the establishment of the fact that the Model enabled Compliance to be built-in to IT-based Business Management systems, the dissertation proceeded to describe regulatory mechanisms for establishing Compliance on a larger scale. This required the availability of Compliant Systems on a large scale and acceptance of Digital Service Contracts.

It also argued that there would be major benefits in drafting regulations in terms of Services. This effectively expresses regulations “logically”, which allows them to be easily translated into Service-based rules for IT systems.

The Model can be viewed as one of the first steps towards mandating the use of compliant business IT systems that provide certainty and security for online transactions. The aim is not to restrict online business by imposing cumbersome regulation, but to encourage it by providing automated security without inconvenience or significant cost to users.

## References

---

- Agrawal, R., Kiernan, J., Srikant, R. and Xu, Y. 2002, 'Hippocratic databases', The 28th International Conference on Very Large Databases (VLDB)
- Al-Kahtani, M. A. and Sandhu, R. 2002, 'A Model for Attribute-Based User-Role Assignment', 18th Annual Computer Security Applications Conference, IEEE, Las Vegas, Nevada, USA, p. 353
- Alotaiby, F. T. and Chen, J. X. 2004, 'A Model for Team-based Access Control (TMAC 2004)', International Conference on Information Technology: Coding and Computing (ITCC'04), IEEE, Las Vegas, Nevada, USA
- Atluri, V. and Warner, J. 2005, 'Supporting Conditional Delegation in Secure Workflow Management Systems', Symposium on Access Control Models and Technologies 2005, ACM Press, New York, NY, USA, Stockholm, Sweden, pp. 59-66
- Bacon, J., Moody, K. and Yao, W. 2002, 'A Model of OASIS Role-Based Access Control and Its Support for Active Security.' ACM Transactions on Information and System Security Vol. 5 No. 4: 492–540.
- Beresnevichene, Y. 2003, A role and context based security model, University of Cambridge Computer Laboratory, Cambridge.
- Bertino, E., Ferrari, E. and Atluri, V. 1999, 'The Specification and Enforcement of Authorization Constraints in Workflow Management Systems.' ACM Transactions on Information and System Security Vol. 2No. 1: 65–104.
- Bishop, M. 2003, Computer Security Art and Science, Pearson Education, Inc., Boston, MA, US.
- Botha, R. A. and Eloff, J. H. P. 2001, 'Separation of duties for access control enforcement in workflow environments.' IBM Systems Journal 403: 666-682.
- Bretan, A. 2004, Authorization Models in Medical Information Systems. Viewed 8<sup>th</sup> November 2011, <<http://www.cse.fau.edu/~security/public/AuthModelsMedInfSystems.ppt>>
- Bussiere, K. 1999, Social Groups. viewed 13th September 2005, <<http://www.usi.edu/libarts/socio/chapter/groups/test.html>>
- Byun, J.-W., Bertino, E. and Li, N. 2005, 'Purpose based access control of complex data for privacy protection', Symposium on Access Control Models and Technologies 2005, ACM Press, New York, NY, USA, Stockholm, Sweden, pp. 102 - 110
- Caelli, W. and Rhodes, A. 1999, RBACManager: Implementing a Minimal Role Based Access Control Scheme (RBACM) Under the Windows NT 4.0 Workstation® Operating System. Viewed 8<sup>th</sup> November 2011, <<http://www.isrc.qut.edu.au/resource/techreport/qut-isrc-tr-1999-003.pdf>>
- Cautis, B. 2007, 'Distributed access control: a privacy-conscious approach', Symposium on Access Control Models and Technologies, ACM, Sophia Antipolis, France, pp. 61 - 70
- Chen, F. and Sandhu, R. S. 1996, 'Constraints for role-based access control', Symposium on Access Control Models and Technologies, ACM Press, New York, NY, USA, Gaithersburg, Maryland, US

- Clark, D. D. and Wilson, D. H. 1987, 'A Comparison of Commercial and Military Computer Security Policies', IEEE Computer Society Symposium on Security and Privacy, Oakland, USA
- Clarke, R. 2002, 'e-Consent: A Critical Element of Trust in e-Business', 15th Bled Electronic Commerce Conference, Bled, Slovenia
- Cohen, E., Thomas, R. K., Winsborough, W. and Shands, D. 2002, 'Models for Coalition Based Access Control (CBAC)', Seventh ACM symposium on Access control models and technologies, ACM Press, Monterey, California, USA, pp. 97-106
- Coiera, E. and Clarke, R. 2004, 'e-Consent: The Design and Implementation of Consumer Consent Mechanisms in an Electronic Environment.' Journal of the American Medical Informatics Association 112: 129-140.
- Crook, R., Ince, D. and Nuseibeh, B. 2002, 'Towards an Analytical Role Modelling Framework for Security Requirements', 8th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ-02), Essen, Germany
- de la Motte, L. 2004, Professional Access Control, Honours Dissertation, School of Computing, University of Tasmania, Launceston, viewed 8<sup>th</sup> November 2011, <<http://eprints.utas.edu.au/118/>>.
- de la Motte, L. 2007, 'Implementation Considerations in the Framing of Privacy Laws.' Privacy Law Bulletin Vol 4 No 2: 14-18.
- de la Motte, L. and Hartnett, J. 2005a, Partial Access Control Permissions and Rights. Viewed 8<sup>th</sup> November 2011, < <http://eprints.utas.edu.au/236/>>
- de la Motte, L. and Hartnett, J. 2005b, 'Professional Access Control', 13th Health Informatics Conference HIC2005, Melbourne, Australia
- de la Motte, L. and Hartnett, J. 2008, 'Using a Client-Task Based Approach to Achieve a Privacy Compliant Access Control System.' electronic Journal of Health Informatics Vol 3 No 1.
- Desmond, J. 2003, Roles or Rules: The Access Control Debate, esecurityplanet. Viewed 8<sup>th</sup> November 2011, <<http://www.esecurityplanet.com/views/article.php/2241671>>
- Dovier, A., Piazza, C., Pontelli, E. and Rossi, G. 2000, Sets and constraint logic programming. Viewed 8<sup>th</sup> November 2011, < <http://dl.acm.org/citation.cfm?doid=365151.365169> >
- El Kalam, A. A., Baida, R. E., Balbiani, P., Benferhat, S., Cuppens, F., Deswarte, Y., Mieke, A., Saurel, C. and Trouessin, G. 2003, 'Organisation based access control', 4th International IEEE Workshop on Policies for Distributed Systems and Networks, IEEE, Lake Como, Italy, pp. 120-131
- Fernandez, R. 2005, Enterprise Dynamic Access Control (EDAC) Overview, SSC San Diego.
- Ferraiolo, D. 2003, Evolution of Access Control in Commercial Products. Viewed 8<sup>th</sup> November 2011, <[http://www.cs.purdue.edu/homes/ninghui/courses/Fall03/lectures/ferraiolo\\_6.pdf](http://www.cs.purdue.edu/homes/ninghui/courses/Fall03/lectures/ferraiolo_6.pdf)>
- Ferraiolo, D. and Kuhn, R. 1992, 'Role-Based Access Control', 15th National Computer Security Conference, Baltimore, MD
- Ferraiolo, D. F., Ahn, G.-J., R.Chandramouli and Gavrila, S. I. 2003, 'The Role Control Center: Features and Case Studies', 8th ACM Symposium on Access Control Models And Technologies, ACM Press New York, NY, USA, Como, Italy, pp. 12 - 20
- Fischer-Hubner, S. and Ott, A. 1998, 'From a Formal Privacy Model to its Implementation', 21st National Information Systems Security Conference, Arlington, VA

- Gallaudet University Dept of Sociology 2008, Definition of Sociology. Viewed 13th September 2005, <<http://depts.gallaudet.edu/sociology/>>
- Georgiadis, C. K., Mavridis, I., Pangalos, G. and Thomas, R. K. 2001, 'Flexible Team-Based Access Control Using Contexts', SACMAT '01, ACM, Chantilly, Virginia, USA, pp. 21-27
- Georgiadis, C. K., Mavridis, I. K. and Pangalos, G. I. 2002, 'Programming a view-based active access-control system for healthcare environments.' Health Informatics Journal (2002): 191-198.
- Gerber, M. E. 1995, The E Myth Revisited, HarperCollins.
- Gollmann, D. 1999, Computer Security, John Wiley & Sons Ltd, Chichester, West Sussex, England.
- Google 2008, Google Health. Viewed 8<sup>th</sup> November 2011, <<http://www.google.com/intl/en-AU/health/about/index.html>>
- Gregor, P. S. 2007, Personal Discussion at EII Winter School Session titled "Techniques for the Evaluation Phase in Information Technology Research", Brisbane.
- Guinard, D. and Trifa, V. 2009, 'Towards the Web of Things: Web Mashups for Embedded Devices', International World Wide Web Conferences, Madrid, Spain
- HealthConnect 2002, Consent and Electronic Health Records - A Discussion Paper.
- Hogg, M. a. 2005, Introduction to Sociology - Groups. Viewed 8<sup>th</sup> November 2011, <[http://en.wikibooks.org/wiki/Introduction\\_to\\_Sociology/Groups](http://en.wikibooks.org/wiki/Introduction_to_Sociology/Groups)>
- Hung, P. C. K. and Karlapalem, K. 2003, 'A Secure Workflow Model', Australasian Information Security Workshop (AISW2003), Australian Computer Society, Inc. - Conferences in Research and Practice in Information Technology, Adelaide, Australia
- IBM The Enterprise Privacy Authorization Language (EPAL). Viewed 8<sup>th</sup> November 2011, <[www.zurich.ibm.com/security/enterprise-privacy/epal](http://www.zurich.ibm.com/security/enterprise-privacy/epal)>
- ITGI (IT Governance Institute) 2007, COBIT 4.1.
- Kaarst-Brown, M. L. and Kelly, S. 2005, 'IT Governance and Sarbanes-Oxley: The latest sales pitch or real challenge for the IT Function?' Hawii International Conference on System Sciences, IEEE, Hawii
- Kern, A. and Walhorn, C. 2005, 'Rule Support for RoleBased Access Control', Symposium on Access Control Models and Technologies 2005, ACM Press, New York, NY, USA, Stockholm, Sweden, pp. 130-138
- Korth, A. 2009, The Web of Identities: Making Machine-Accessible People Data. viewed 2nd August 2011, <[http://www.readwriteweb.com/archives/web\\_of\\_identities\\_making\\_machine-accessible\\_people\\_data.php](http://www.readwriteweb.com/archives/web_of_identities_making_machine-accessible_people_data.php)>
- Koshutanski, H. and Massacci, F. 2003, An Access Control Framework for Business Processes for Web Services. Viewed 8<sup>th</sup> November 2011, <<http://delivery.acm.org/10.1145/970000/968562/p15-koshutanski.pdf?key1=968562&key2=3364201111&coll=GUIDE&dl=ACM&CFID=40901577&CFTOKEN=92407136>>
- Kudo, M. 2002, 'PBAC: Provision-based access control model.' International Journal of Information Security 12: 116-130.
- Lampson, B. W. 1971, 'Protection', 5th Princeton Conf. on Information Sciences and Systems, Princeton NJ US

- Lee, A. J., Winslett, M., Basney, J. and Welch, V. 2006, 'Traust: A Trust Negotiation-Based Authorization Service for Open Systems', 11th ACM Symposium on Access Control Models and Technologies, ACM, Lake Tahoe, California, USA, pp. 39 - 48
- LeFevre, K., Agrawal, R., Ercegovac, V., Ramakrishnan, R., Xu, Y. and DeWitt, D. 2004, 'Disclosure in hippocratic databases', The 30th International Conference on Very Large Databases (VLDB)
- Lehnen, A. date unknown, An Elementary Introduction to Logic and Set Theory. Viewed 9th August 2005, <<http://matchmadison.edu/alehnen/weblogic/logcont.htm>>
- Li, N. and Mitchell, J. C. 2002, 'Design of a Role-based Trust-management Framework', IEEE Symposium on Security and Privacy, 2002, IEEE
- Li, N. and Mitchell, J. C. 2003, 'RT: A Role-based Trust-management Framework', Third DARPA Information Survivability Conference
- Maani, K. e. and Cavana, R. Y. 2002, Systems Thinking and Modelling - Understanding Change and Complexity, Pearson Education New Zealand.
- Neumann, G. and Strembeck, M. 2003, 'An Approach to Engineer and Enforce Context Constraints in an RBAC Environment', SACMAT '03, ACM, Como, Italy, pp. 65-79
- NIST 2004, Homepage - Role Based Access Control. Viewed 8<sup>th</sup> November 2011, <<http://csrc.nist.gov/rbac/>>
- OASIS 2006, Reference Model for Service Oriented Architecture 1.0.
- OASIS 2009, Reference Architecture Foundation for Service Oriented Architecture Version 1.0 Committee Draft 02.
- Organisation for Economic Co-operation and Development 2006, OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data. Viewed 8<sup>th</sup> November 2011, <[http://www.oecd.org/document/18/0,2340,en\\_2649\\_34255\\_1815186\\_1\\_1\\_1\\_1,00.html](http://www.oecd.org/document/18/0,2340,en_2649_34255_1815186_1_1_1_1,00.html)>  
<[http://www.oecd.org/document/18/0,2340,en\\_2649\\_201185\\_1815186\\_1\\_1\\_1\\_1,00.html](http://www.oecd.org/document/18/0,2340,en_2649_201185_1815186_1_1_1_1,00.html)>
- Oz, E. 2004, Management Information Systems, Thomson Course Technology.
- Panko, R. R. and Ordway, N. 2005, 'Sarbanes-Oxley: What about All the Spreadsheets?' EuSpRIG 2005, University of Greenwich, London, UK
- Pappas, A. and Hailes, S. 2003, A Comparison of Traditional Access Control Models and Digital Rights Management. Viewed 8<sup>th</sup> November 2011, <<http://www.ee.ucl.ac.uk/~apappas/drafts/LCS%202003%20paper.pdf>>
- Pfleeger, C. P. 2000, Security in Computing, Prentice Hall PTR, Upper Saddle River, New Jersey.
- Pfleeger, C. P. and Pleeger, S. L. 2003, Security in Computing, Prentice Hall PTR, Upper Saddle River, New Jersey.
- Popper, K. and Miller, D. 1983, A proof of the impossibility of inductive probability, Nature: 687-688.
- Povey, D. 1999, Optimistic Security: A New Access Control Paradigm. Viewed 8<sup>th</sup> November 2011, <<http://delivery.acm.org/10.1145/340000/335188/p40-povey.pdf?key1=335188&key2=3036482111&coll=GUIDE&dl=ACM&CFID=41485954&CFTOKEN=74504619>>

- Ramaswamy, C. and Sandhu, R. 1998, 'Role-Based Access Control Features in Commercial Database Management Systems', 21st National Information Systems Security Conference, Crystal City, Virginia, USA
- Rhodes, A. and Caelli, W. 1999, A Review Paper Role Based Access Control, University of Queensland, Brisbane Australia.
- Rissanen, E., Firozabadi, B. S. and Sergot, M. 2004, Towards A Mechanism for Discretionary Overriding of Access Control, Viewed 8<sup>th</sup> November 2011, <<http://www.sics.se/isl/pbr/papers/overridepos.pdf>>.
- Rissanen, E., Firozabadi, B. S. and Sergot, M. 2005, Discretionary Overriding of Access Control in the Privilege Calculus. Viewed 8<sup>th</sup> November 2011, <<http://www.doc.ic.ac.uk/~mjs/publications/override-fast2004.pdf>>
- Russell, N., Hofstede, A. H. M. t., Edmond, D. and Aalst, W. M. P. v. d. 2005, Workflow Resource Patterns. Viewed 8<sup>th</sup> November 2011, <<http://www.wis.win.tue.nl/~wvdaalst/BPMcenter/reports/.../BPM-04-07.pdf>>
- Sandhu, R. S., Coynek, E. J., Feinsteink, H. L. and Youmank, C. E. 1996, 'Role-Based Access Control Models.' IEEE Computer 292: 38-47.
- Stevens, G. and Wulf, V. 2002, A New Dimension in Access Control: Studying Maintenance Engineering across Organizational Boundaries. Viewed 8<sup>th</sup> November 2011, <<http://dl.acm.org/citation.cfm?id=587106>>
- Thomas, R. K. 1997, 'Team-based Access Control (TMAC): A Primitive for Applying Role-based Access Controls in Collaborative Environments', RBAC '97, ACM, Fairfax Va USA, pp. 13-19
- Thomas, R. K. and Sandhu, R. S. 1997, 'Task-based Authorisation Controls (TBAC): A Family of Models for Active and Enterprise-oriented Authorisation Management', IFIP WG11.3 Workshop on Database Security, Chapman & Hall, Lake Tahoe, California, USA
- W3C 2011, Web of Services. Viewed 8<sup>th</sup> November 2011, <<http://www.w3.org/standards/webofservices/>>
- Wang, L., Wijesekera, D. and Jajodia, S. 2004, 'A Logic-based Framework for Attribute based Access Control', 2004 ACM workshop on Formal methods in security engineering
- Wang, W. 1999, 'Team-and-Role-Based Organizational Context and Access Control for Cooperative Hypermedia Environments', ACM Hypertext '99, ACM, Darmstadt, Germany, p. 37-46
- Warner, J. and Atluri, V. 2006, 'Inter-instance authorization constraints for secure workflow management', 11th ACM Symposium on Access Control Models and Technologies, ACM, Lake Tahoe, California, USA, pp. 190 - 199
- Wikipedia 2008a, Group (Sociology). Viewed 8<sup>th</sup> November 2011, <[http://en.wikipedia.org/wiki/Group\\_%28sociology%29](http://en.wikipedia.org/wiki/Group_%28sociology%29)>
- Wikipedia 2008b, Inductive Reasoning. Viewed 8<sup>th</sup> November 2011, <[http://en.wikipedia.org/wiki/Inductive\\_reasoning](http://en.wikipedia.org/wiki/Inductive_reasoning)>
- Wikipedia 2008c, Sarbanes-Oxley Act. Viewed 8<sup>th</sup> November 2011, <<http://en.wikipedia.org/wiki/Sarbanes-Oxley>>
- Wikipedia 2011, Wright Brothers. Viewed 13<sup>th</sup> September 2011, <[http://en.wikipedia.org/wiki/Wright\\_brothers](http://en.wikipedia.org/wiki/Wright_brothers)>
- Willmott, L., Christensen, S. and Butler, D. 2005, Contract Law, Oxford University Press.

## Glossary

---

### A

**ABAC** – Attribute Based Access Control.

**Access by Authorisation** – The principle that requires that Client permission to access Service Information is communicated to the CMS as an Authorisation Request from or on behalf of the Client.

**Access Control** – A Research Field within Computer Security that is fundamentally concerned with “who has access to what” in an IT system.

**Access Services** – Model Services that access PID.

**Accountability** – Relating to meeting accounting-related Regulatory requirements through the use of appropriate Service Contracts and Service Delivery processes.

**Accounting Services** – Services associated with each business Service that administer accountability related functions.

**ACL** – Access Control List.

**Agent** – A person/party who is legally authorised to act on behalf of a contracting party (the principal).

**AM Services** – Authorisation Management Services.

**Associated Sub-Services** – Sub-Services that are each components of the same Service.

**Associate Authorisation** – Allows a colleague of a Team member with the same role to assist the Team Member with their work.

**Auditing Services** – Services that perform auditing functions which include the management of user accounts.

**Authentication** (Bishop (2003, p. 309)) – The binding of an identity to a subject.

**Authorisation** (General) – The mechanism providing the means by which Service Contract offers and acceptances are input into the IT System during Service allocation and delivery processes.

**Authorisation** (Human-Computer Interaction) – The *act* of entering an Authorisation Request into the IT System.

**Authorisation** (Business Decision) – The business *decision* to allocate organisational (human and physical) resources.

**Authorisation** (Legal/Contractual) – Where the IT System stores some information that *proves* an Authorisation Request has been granted, for example, that a Worker is Authorised to perform a Client-Service.

**Authorisations** (*ex ante*) – Authorisations given prior to information access.

**Authorisations** (*ex post*) – Authorisations given retrospectively.

**Authorisations** (*uno tempore*) – Authorisations given at the time access is required.

**Authorisation Management** – The principle that requires the CMS to manage all Service Authorisations to ensure that all Services are performed with proper Authorisations in place.

**Authorisation Order** – The priority for choosing between multiple authorisers where authorisations given by one authoriser can override those of another authoriser or there is a logical order of who to seek an authorisation from.

**Authorisation Process** – The Authorisation Client-Service Process.

**Authorisation Client-Service Process** – Step #4 in the Service Management Process that includes Allocation and Offer, Worker Acceptance and Client Acceptance.

**Authorisation Request** – A request to the CMS that an Authorisation be granted, that is, that a Client-Service be allowed to be performed.

**Authorisation Service** – A Sub-Service of the Service to be delivered that is responsible for ensuring that the Service is properly Authorised.

**Authorisation Services** – Services that handle the granting and modification of Authorisations.

**Authorisation Timing** – Has to do with *when*, relative to the time of Service Delivery, the Worker received the required authorisation.

**Authorisation Type Specification** – The principle requiring that the valid Authorisation Type(s) (*ex ante*, *uno tempore* or *ex post*) of all Authorisation Requests associated with a Client-Service be specified in the Service Description.

**Authoriser** – A person who is authorised by the CMS to input an Authorisation Request.



**Automatic Sub-Service Authorisation Request** – An Authorisation Request that is automatically generated when a Service Authorisation Request is made that covers the Sub-Service.

## B

**Base Components** – Base Services, Base Groups and Base Service Contracts.

**Base Group** – The top level Group Template upon which all Groups are based.

**Base Level Group** – A Group that has no members and represents a single Organisational resource.

**Base Service** – The top level Service Template upon which all Services are based.

**Base Service Contract** – The top level Service Contract Template upon which all Service Contracts are based.

**BLG** – Base Level Group.

**BPM** – Business Process Management.

**BUG** – Base-level User Group.

**Bundled Client Consent** – Where a single Consent acceptance is given by the Client to cover more than one Client-Service.

**Business Management** – One of the Three Research Disciplines.

**Business Services** – The Internal Services that are used to manage the Organisation as well as External Services which represent the Services that the Organisation offers to its customers.

**B2B** – Business to Business.

**B2C** – Business to Consumer.

## C

**CBAC** – Coalition Based Access Control.

**CDIs** – Constrained Data Items.

**Central Requirements** – The Three Central Requirements.

**CIO** – Chief Information Officers.

**Client** – The recipient of Organisational Services who is external to the Organisation unless the Service is an Internal Service.

**Client Authorisation** – The principle requiring that Clients must Authorise each Client-Service they are to receive to indicate their consent/acceptance of the Service.

**Client-Centric Approach** – Where all Service instances have a specified Client.

**Client-Service** – A Service Instance, that is, one instance of the delivery of an Organisational Service to a Client.

**Client-Service Accounting** – Each Client-Service is one accounting item.

**Client-Service Allocation** – Occurs when a Manager Authorises a Worker(s) to perform a Service(s) for a Client(s).

**Client-Service Contracts** – Contracts that exist for each Client-Service performed by the Organisation and specify the rights and responsibilities of the Organisation and the Client regarding the Service.

**Client-Service Delivery Contract** – A Service Contract Instance, that is, a Manager to Client Service Contract covering privacy, consent and accountability.

**Client-Service Management** – Incorporates the management of all Organisational Services.

**Client-Service Records** – Records for all Client-Services.

**Cloud Computing** – A model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (eg, networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

**CMS** – Compliant Management System.

**CMSOS** – Compliant Management System Operating System.

**CMS Data** – The entire collection of Component Data and Component Instance Data that constitutes the CMS.

**CMS Model** – The Compliant Management System Model.

**CMS Term** – Individual terms defined in this dissertation.

**CMS Terminology** – The entire collection of terms defined in the dissertation.

**CM Services** – Contract Management Services.

**COBIT** – Control Objectives for Information and related Technology.

**Compliance** – The state where an Organisation meets all the requirements of applicable Regulations.

**Compliance Concepts** – Privacy, Consent, Accountability and Authorisation.

**Compliant Management System** – An IT System that is based on the Model.

**Compliant Management System Operating System** – The collection of System Components at the core of the CMS.

**Compliant System** – A business management system that inherently maintains Compliance.

**Component** – Either a Service, a Group or a Service Contract.

**Component** – A template for Component Instances.

**Component** – Software abstractions for Services, Groups and Service Contracts used in the Model.

**Component Concepts** – Services, Groups and Service Contracts.

**Component Data** – Data pertaining to a Component.

**Component Hierarchies** – The hierarchies of Services, Groups and Service Templates.

**Component Icons** – Graphical representation of Components that are used for explanation purposes.

**Component Instance** – A Service Instance, a Group Instance or a Service Contract Instance.

**Component Instance Data** – Data pertaining to a Component Instance.

**Component Processes** – Processes defined within a Component.

**Component Template** – A Service Template, a Group Template or a Service Contract Template.

**Component Types** – Services, Groups and Service Contracts.

**Compulsory Compliance** – Describes the imposition that various regulations and legal requirements place on Organisations.

**Computer Security** – Security concerned primarily with the digital information.

**Concepts** – The Model Concepts.

**Consent** – Where Client consent is incorporated into Service Contracts and managed in Service processes.

**Consent by Authorisation** – The principle requiring that Consent for a Service is communicated to the CMS as an Authorisation Request from or on behalf of the Client.

**Consent Type Specification** – The requirement that the consent type(s) for each Service be specified together with any conditions, so that what constitutes legitimate consent is clear.

**Constrained Data Items** – Data items within the system to which the (Clark-Wilson) integrity model must be applied.

**Context** – Conditions upon which an Authorisation is given, that is, a specification of who, what, when, where or why can perform an action, particularly a Service.

**Contextual Groups** – Organisational Groups that each represent a specific Context, where membership of the Group confers the context on the Group member.

**Contract** – A Service Contract.

**Contract Acceptance** – The legal acceptance of an offer by a person to whom the offer is made (the offeree).

**Contract Based Services** – A principle specifying that all Client-Services offered by the Organisation are Contract-based and that both Worker-Service Contracts and Client-Service Contracts are in place to cover each Client-Service, before it is performed.

**Contract Management** – The administration of Contracts.

**Contract Management Services** – Service provided by the System to change Contract Data and Contract Instance Data that are used as Sub-Services by other Services.

**Contract Offer** – The indication of a willingness to legally contract with another person (the offeree) on certain terms.

**Controller Services** – CMSOS Services that perform operations (processes) on data.

**Critical Authorisation** – Allows access to any part of any record.

**Critical Services** – Services that can be performed by authorised personell anytime they see fit.

**C-List** – Capability List.

**C-TMAC** – Context-based Team Based Access Control.

## D

**Descriptive Service Information** – The information that details the Service and its possible implications for, and effect on, the Client.

**Digitally Managed Contract** – A contract where all the evidence that is required to enforce the contract is either stored digitally or the location where it can be readily retrieved is stored digitally.

**Digital Service Contract** – A Service Contract.

**Digital Service Contract** – The Model instrument that is central to achieving Organisational Compliance.

**Digital Service Contract** – A Service Contract where Authorisation Requests from each party are construed as the party's offer or acceptance of the Service.

**DMC** – Digitally Managed Contract.

**DR** – Design Requirements.

**DSC** – Digital Service Contract.

## E

**Efficiency** – The state where an Organisation functions with lower administrative overheads and/or greater commercial viability than current methods or systems generally allow.

**Emergency Authorisation** – Allows access to any Clients' record but limits access.

**Emergency Services** – Services that can be performed by authorised personnel for Clients that have not been assigned to them.

**Enforcer** – The Manager in charge of the System, who is responsible for Compliance, that is, establishing, configuring, monitoring and reporting on the System's performance and ensuring that Organisational policy is enforced.

**ERBAC** – Enterprise RBAC.

**Express Consent** – Express Consent requires that an explicit indication of Consent (such as in writing, electronically or verbally) be given.

**External Clients** – Clients who are outside the Organisation.

**External Group** – Any Group from another Group Based Organisation.

**External Services** – Services provided to Clients outside the Organisation.

## F

**First Central Requirement** – Compliance.

**Foundational Model Concepts** – The seven main Model Concepts that form the basis of the Model.

## G

**GAS** – Global Access System.

**General Concepts** – The Seven General Concepts.

**Generic Service Contract** – A standard Service Contract that is used repeatedly for the same type of Service.

**GGA** – Global Group Address.

**Global Access System** – A proposed System based on the Model that utilises Global Groups and registered Identities for Individuals, Organisations and Organisational Groups.

**Global Base-level User Group** – A BUG that represents a user's global identity.

**Global Components** – Components that are accessible globally and can be copied and modified as necessary to meet local requirements.

**Global Functionality** – The state where a CMS is functional in processing terms (for example, processing speed) and interoperable with other relevant systems on the Internet.

**Global Groups** – Organisational Groups that are open to memberships from External Groups or are themselves enabled to be members of External Groups.

**Global Group Address** – A network address that represents a Group.

**Global Group Addressing** – Addressing that provides all Global Groups with a global internet address.

**Global Memory Address** – A globally accessible address in computer memory where specific data is stored.

**Global Service** – A simple standardised unit of commerce.

**GMA** – Global Memory Address.

**GM Services** – Group Management Services.

**Group** – One of the Component Types.

**Group** – A resource set with related membership data and Management Groups.

**Groups** – Groups represent Organisation entities/resources.

**Group Based Access** – A method that enables an Organisation to allow External Group members to perform Services within the Organisation by extending internal Group membership to the members' External Group.

**Group Based Information Storage** – A paradigm for systematic data storage where information describing a global entity is stored in memory locations that are directly associated with the Global Group that represents the entity.

**Group Based Management** – The fundamental Business Management methodology employed in Group Based Organisations.

**Group Based Organisation** – An Organisation that uses Organisational Groups to define its management structure, its Staff and to specify authorisation rules for Client-Service provision.

**Group Based Services** – Services that are associated with Groups, can only be activated through that association, and can only be performed by Group members.

**Group Data** – Data pertaining to a Group.

**Group Functionalities** – The means for granting and revoking Group membership.

**Group Instance** – One instance of a Group.

**Group Instance Data** – Data pertaining to a Group Instance.

**Group Management** – The administration of Groups.

**Group Management Services** – The Services used to manage Groups.

**Group Metadata** – The data that describes the Group and its members.

**Group Properties** – That Groups are used to define and manage resources, are hierarchical, globally addressable, are instances of a Group Template, have Types, Instances and Sub-Types, and are associated with Services.

**Group Rules** – That members of all Groups are other Groups AND all resources have their own Base Level Group.

**Group Sub-Type** – A category of Sub-Groups in the Group Hierarchy.

**Group Template** – A pattern from which Group Instances are created.

**Group Type** – One of the Component Types.

**Group Type** – A category of Groups in the Group Hierarchy that enable the classification of Group Templates.

**Group Type** – A classification of a Group based on a common attribute that all Group Members share.

**G-BUG** – Global Base-level User Group.

## H

**HIPAA** – (United States) Health Insurance Portability and Accountability Act.

## I

**IaaS** – Infrastructure as a Service.

**Implied Authorisation Request** – An Authorisation Request that is based on Implied Consent.

**Implied Consent** – Consent that can be inferred from the circumstances or a person's actions.

**Implied Permissions** – Permissions to perform actions that are given when the Client gives Consent to receive a Service.

**Info** – Service Information.

**Information Protection** – Protecting the Privacy, confidentiality, integrity and availability of information.

**Information Security** – Security pertaining to verbal, written as well as digital information.

**Information Security Management** – The management of Information Security.

**Informed Consent** – Consent that is given voluntarily (not under any coercion) by a capable (that is, mentally and physically) Client after being properly informed of the implications of consent.

**Input Service** – A CMSOS Service that receives data from input devices.

**Integrity Verification Procedure** – A procedure that confirms that all of the CDIs in the system conform to the integrity specification at the time the IVP is executed.

**Intermediate Model Concept** – Model Concepts that were developed before the final Model was formed.

**Internal Clients** – Clients who are within the Organisation.

**Internal Services** – Services provided to Clients within the Organisation.

**Intersection** – The Set Operation ( $A \cap B$ ) where membership of *both* sets is required (that is, membership of A AND B).

**IT** – Information Technology.

**IT** – One of the Three Research Disciplines.

**ITGI** – IT Governance Institute.



**IT System** – Compliant Management System.

**IVPs** – Integrity Verification Procedures.

## L

**Law** – One of the Three Research Disciplines.

**LEs** – Large Enterprises.

**Levels of Compliance** – A measure of the degree that a system complies with regulation.

**Levels of Coverage** – Relate to the jurisdiction to which compliance is applied.

**Levels of Scope** – Relate to the size of the business collection to which compliance is applied or regulated.

**Levels of Standardization** – Relate to the type of system, with respect to compliance, that is employed by an organisation.

## M

**Management Group Set** – The set of Groups responsible for managing each specific Group.

**Manager** – An Organisational representative with the authority to make Service-based decisions on behalf of the Organisation.

**Managerial Authorisations** – The principle requiring that Managers Authorise each Client-Services on behalf of the Organisation.

**Mandated Systems** – Compliant Systems that are directly or indirectly configured by regulatory bodies (Indirect configuration is where a regulatory body, or its agent, test and officially rate or sanction the system's compliance, and Direct configuration is where procedures and/or data supplied by the regulatory body are used to control the system).

**Mandate Compliant Systems** – The legal steps regulators must take to make the use of Compliant systems mandatory for organisations.

**Mandatory Authorisation** – The principle dictating that all Client-Services must be Authorised by the CMS.

**MCR Diagram** – Model Concepts Relationship Diagram.

**MGS** – Management Group Set.

**Model** – The Compliant Management System Model.

**Model Components** – Components.

**Model Component Types** – Component Types.

**Model Concepts** – The set of 52 Concepts from which the Model is formulated.

**Model Design Concepts** – The Service Based Organisation Design Concept and the Service Based Work Design Concept.

**Model Services** – CMSOS Services that interact with stored Component Data and Component Instance Data.

**Monitoring Services** – Services that look for unusual accesses and access patterns and report any suspicious activities.

**Multiple-Instance Client-Service** – A Service can also be performed multiple times for the same Client.

## N

**Need-to-Know Access** – Where access is restricted to persons who need information in order to perform authorised tasks.

**NIST** – (United States) National Institute of Standards and Technology.

## O

**OASIS** – Organization for the Advancement of Structured Information Standards.

**OECD** – Organisation for Economic Co-operation and Development.

**Online Component Repositories** – Online storage facilities for standard for Component Templates.

**OOP** – Object Oriented Programming.

**ORBAC** – Organisation Based Access Control.

**Organisation** – An organisation that employs a CMS.

**Organisational Base-level User Group** – An identity/alias that represents a user within an organisational IT system.

**Organisational Concept** – “Organisations exist to provide Services to Clients”.

**Organisational Premise** – That “a Manager Authorises a Worker to perform a Service for a Client”.

**Organisational Template** – A collection of Service, Group and Service Contract templates that can be used as the basis for an Organisational CMS.

**Organisation Group** – A single Global Group that represents the Organisation as a global entity.

**Output Service** – A CMSOS Service that sends data to output devices.

**O-BUGs** – Organisational Base-level User Group.

**O-BUGs** – An identity/alias that represents a user within an organisational IT system.

## P

**PaaS** – Platform As a Service.

**PAC** – Professional Access Control.

**Partial Permissions** – Permissions which apply to multiple Groups that are specified through rules stored with and applied to each of the Groups involved.

**PC** – Personal Computer.

**PCAOB** – Public Company Accounting Oversight Board.

**PDAs** – Personal Digital Assistants.

**PDC** – Purely Digital Contract.

**Personal Service Information** – The information that is entered or modified as part of Service Delivery.

**PID** – Personally Identified (or Identifiable) Data.

**Primary CM Services** – Authorisation Management Services that are utilized for every Client-Service.

**Privacy** – Where Client privacy is enabled by restricting access to information to that which is necessary for the Client-Service being provided.

**Provider** – An organisation that provides a service, particularly a Global Service.

**Provide DSC Enforcement Mechanisms** – The legal steps regulators must take to ensure that appropriate and cost effective means exist for parties to seek legal redress and receive compensation.

**Purely Digital Contract** – A form of DMC where all the information and the authorisations (offers and acceptances) are recorded digitally.

## R

**RBAC** – Role Based Access Control.

**RB-RBAC** – Rule Based RBAC.

**Reactionary Systems** – Compliant Systems that are either configured by the Organisation or its agent (usually the system supplier) that deal with imposed requirements by continually being updated or modified in order to maintain their currency.

**Recognise DSC Legitimacy** – The legal steps regulators must take to ensure that DSCs can be used as evidence in Contract Litigation.

**Regulation** – Any rule that is externally imposed on an Organisation to which it is required to comply.

**Relative Complement** – The Set Operation ( $A - B$ ) where one membership *but not* another is required (that is,  $A \text{ AND NOT } B$ ).

**Reporting Services** – Services that provide specified reports.

**Required Auditing** – Where Compliance requirements are met by ensuring that Client-Services are adequately audited.

**Requirement Extraction Process** – The extraction of Design Requirements.

**Research Disciplines** – The Three Research Disciplines.

**Retrieve Service** – A CMSOS Service that accesses stored Component Data and Component Instance Data.

**RT** – Role-Based Trust Management.

**Rule Management** – The administration required to set up relationships between Services, Groups and Service Contracts.

## S

**SaaS** – Software as a Service.

**Secondary CM Services** – Administration Services that are triggered when Authorisations are received by the System.

**Second Central Requirement** – Efficiency.

**Service** – One of the Component Types.

**Serviceflows** – The method which defines business processes and the relationships between dependent Services.

**Serviceflow Dependencies** – Specify the dependencies between Services and component Sub-Services.

**Services** – The units of work performed by the Organisation.

**Services** – Templates for standard business process tasks in Organisations with each Client-Service instance being managed by the IT System.

**Service Based Access Principle** – The principle that Services access the information they require, and access to who performs Services is restricted to Authorised personnel.

**Service Based Accountability** – The principle requiring that all accounting be done in terms of Services.

**Service Based Auditing** – Where Client-Services are the items that are audited with the aim of assuring that all Organisational Services are compliant.

**Service Based Authorisation** – The principle mandating that Managerial, Worker and Client Authorisations are received before a Client-Service is performed.

**Service Based Consent** – The principle requiring that all Client Consent in a Service Based Organisation applies directly to the Services that the Organisation performs for the Client.

**Service Based Employment Contract** – Employment contracts that recognise each Single-Service Contract as a separate employment Sub-Contract.

**Service Based Information** – Encompasses all the information associated with a Client-Service, including the Service Profile and Client-Service Management data.

**Service Based Information Provision** – The principle requiring that Information relating the performance of a Service is provided to the Client to fulfil informed consent requirements and allow Client review.

**Service Based Organisation** – An Organisation composed of Services, Groups and Service Contracts.

**Service Based Organisation** – An Organisation that defines all its work in terms of Global Services.

**Service Based Organisation Design Concept** – A Service Based Organisation is composed of Services, Groups and Service Contracts.

**Service Based Privacy** – The principle that uses Service Based Consent to restrict access to Services and the Service Based Access Principle to restrict access to Client Information.

**Service Based Purpose** – Describes the fundamental assumption that the purpose for accessing a Client's information in an Organisational environment is to perform a Service for a Client.

**Service Based Security** – The methodology whereby Information Security is maintained in the course of the management of Organisational Services.

**Service Based Work** – Work that is composed of Client-Services where Groups do Contracted Services for Clients.

**Service Based Work Design Concept** – Service Based Work is composed of Client-Services where Groups do Contracted Services for Clients.

**Service Cell** – A Service Cell is the technical (IT) definition and specification of a standardised Global Service.

**Service Cell Paradigm** – A paradigm that portrays the idea of Service Based Organisations being composed of Service Components (as organisms are composed of cells); these components having a basic design that can be adapted and replicated.

**Service Code** – The program code that constitutes a Service Component.

**Service Contract** – One of the Component Types.

**Service Contract** – A digital Service agreement that enables Organisational Compliance by fulfilling Privacy, Consent and Accountability Regulations and CMS Authorisation requirements.

**Service Contract** – A Digital contract that defines work expectations for the Organisation, its personnel and the Client.

**Service Contract Data** – Data pertaining to a Service Contract.

**Service Contract Instance** – One instance of a Service Contract.

**Service Contract Instance Data** – Data pertaining to a Service Contract Instance.

**Service Contract Properties** – That Service Contracts are Organisation based, are Single-Service, recognize Sub-Services, are generic in nature, and are digitally managed.

**Service Contract Template** – A pattern from which Service Instances are created.

**Service Data** – Data pertaining to a Service.

**Service Delivery** – The work involved in providing a Client-Service.

**Service Description** – A description of the details of Service Delivery and the contractual requirements.

**Service Information** – Information used by, provided by or generated by a Service Instance during Service Delivery, particularly PID.

**Service Instance** – One instance of a Service.

**Service Instance Data** – Data pertaining to a Service Instance, that is, a Client-Service.

**Service Interfaces** – The user interfaces utilised when a Client-Service is performed.

**Service Management** – The management of each Client-Service by the IT System.

**Service Management Process** – The set of Sub-Services used to manage each Client-Service.

**Service Metadata** – The data that describes the Service and its operation.

**Service Process** – The computer process that runs when a Client-Service is performed.

**Service Profile** – The set of standard information that describes a Service.

**Service Properties** – That Services are all-encompassing, apply to standardized parties, consist of standardized processes, have commercial aspects, and are managed by a CMS.

**Service Request** – A request to the CMS that triggers the creation of a new Client-Service.

**Service Team** – An Organisational group containing members (Managers and Workers) who are Authorised to perform Services for a specified Client.

**Service Template** – A pattern from which Service Instances are created.

**Set Based Groups** – Groups whose membership is defined by a mathematical set.

**Set Operations** – Mathematical operation involving sets.

**Seven General Concepts** – Services, Groups, Service Contracts, Privacy, Consent, Accountability and Authorisation.

**SGD** – Simple Group Definition.

**Simple Group Definition** – A Group is a resource set with related membership data and Management Groups.

**Single-Service Contract** – All Services have their own contract.

**SME** – Small to Medium Enterprise.

**SOA** – Service Oriented Architecture.

**SOX** – Sarbanes-Oxley Act of 2002.

**Specific Concepts** – A Model Concept.

**Standard Model Services** – The CMSOS Services that provide the Model's functionality.

**Standard Requirements** – The set of 50 requirements derived from the three standards and used for the Analytical Evaluation of the Model.

**Standard Requirement Set** – The Standard Requirements.

**Store Service** – A CMSOS Service that stores Component Data and Component Instance Data.

**Sub-Component** – A Component that is part of a higher level Component in the Component Hierarchy.

**Sub-Concept** – A Model Concept that is associated with one of the Seven General Concepts.

**Sub-Contract** – A Service Contract that is part of a higher level Service Contract in the Service Contract Hierarchy.

**Sub-Group** – A Group that is a member of a higher level Group in the Group Hierarchy.

**Sub-Service** – A Service that is part of a higher level Service in the Service Hierarchy.

**Sub-Service Authorisation** – The principle implying that when a Service is Authorised all the Sub-Services are automatically Authorised.

**System** – Compliant Management System.

**System Components** – The Model Services, View Services and Controller Services that make up the CMS Operating System.

## T

**TBAC** – Task-Based Authorisation Control.

**TCSEC** – Trusted Computer System Evaluation Criteria.

**Third Central Requirement** – Global Functionality.

**Three Central Requirements** – Compliance, Efficiency and Global Functionality.

**Three Research Disciplines** – IT, Law and Business Management disciplines.

**TMAC** – TeaM-based Access Control.

**TPs** – Transformation Procedures.



**Transformation Procedures** – Well formed transactions that change the set of CDIs from one valid state to another according to the Worker's role.

## U

**Unassociated Sub-Services** – Sub-Services that are independent of one another.

**Union** – The Set Operation ( $A \cup B$ ) where membership of *either* set is required (that is, membership of  $A \text{ OR } B$ ).

**URL** – Universal Resource Locator.

## V

**Validation by Authorisation** – A principle requiring the CMS to ensure compliance by requiring that Authorisation Requests from each party to a Service Contract are received before the Service is validated.

**View Services** – CMSOS Services that interact with input and output devices.

## W

**Web of Identities** – A proposed system based on the Model that provides a mechanism for registering Identities for Individuals, Organisations and Organisational Groups.

**WMS** – Workflow Management System.

**Worker** – An Organisational representative or sub-contractor who delivers Organisational Services to Clients on behalf of the Organisation.

**Worker Acceptance** – The principle requiring that Workers must choose to accept the Client-Services that they are allocated.

**Worker Authorisation** – The principle requiring that Workers must Authorise each Client-Service in order to indicate their acceptance to deliver the Service on the Organisation's behalf.

**Worker-Service Contracts** – Employment Contracts or Sub-Contracts that are expressed in terms of Organisational Services and specify the rights and responsibilities of the Organisation and the Worker.

**5 Ws of Access Control** – (Who, What, When, Where and Why) define the five primary Access Control questions that must to be answered in order to grant or deny access requests.