2003

A university wide smart card system

Philip Cummins
.Edith Cowan University

Recommended Citation

This Thesis is posted at Research Online.
https://ro.ecu.edu.au/theses_hons/135
A UNIVERSITY WIDE SMART CARD SYSTEM

A dissertation to be submitted in partial fulfilment of the requirements for the degree of:

Bachelor of Science Honours (Computer Science)

By: Philip Cummins, 0953848

Supervisor: Associate Professor Wojciech Kuczborski

Date: 12th of December, 2003
Edith Cowan University

Copyright Warning

You may print or download ONE copy of this document for the purpose of your own research or study.

The University does not authorize you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site.

You are reminded of the following:

• Copyright owners are entitled to take legal action against persons who infringe their copyright.

• A reproduction of material that is protected by copyright may be a copyright infringement. Where the reproduction of such material is done without attribution of authorship, with false attribution of authorship or the authorship is treated in a derogatory manner, this may be a breach of the author’s moral rights contained in Part IX of the Copyright Act 1968 (Cth).

• Courts have the power to impose a wide range of civil and criminal sanctions for infringement of copyright, infringement of moral rights and other offences under the Copyright Act 1968 (Cth). Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.
USE OF THESIS

The Use of Thesis statement is not included in this version of the thesis.
ABSTRACT

Presently, many tertiary educational institutions utilise card technologies for staff and student identification and the support of other related services within the institution. A new type of card technology has been developed known as a Smart Card, where an existing plastic card has an embedded integrated circuit and is capable of expanding the range of applications and services capable within an institution.

This thesis will detail the advantages that these new Smart Cards have over existing institution cards, detail other institution’s attempts to use Smart Cards and finally, this thesis proposes a suitable system for integrating Smart Cards into an existing tertiary educational institution.
COPYRIGHT AND ACCESS DECLARATION

I certify that this thesis does not, to the best of my knowledge and belief:

(i) incorporate without acknowledgment any material previously submitted for a degree or diploma in any institution of higher education;

(ii) contain any material previously published or written by another person exception where due reference is made in the text; or

(iii) contain any defamatory material

Philip Cummins

Date: 6/01/2004
ACKNOWLEDGMENTS

I thank the following people who helped with this thesis:

Associate Professor Wojciech Kuczborski for supervising this thesis.

My immediate family for supporting me.

The School of Computer Science & Software Engineering, the University of Western Australia for allowing me to continue my pursuit of higher education.

Doctor Dominic Voon and Michael Verini, who were always encouraging me to finish.
TABLE OF CONTENTS

Use of Thesis ........................................................................................................... 1
A University Wide Smart Card System ................................................................... i
Abstract ................................................................................................................ ii
Copyright and Access Declaration ......................................................................... iii
Acknowledgments .................................................................................................... iv
Table of Contents .................................................................................................... v
  Table of Figures ....................................................................................................... vii
1.0 Introduction ........................................................................................................ 1
  1.1 Background ....................................................................................................... 1
  1.2 Significance ....................................................................................................... 1
  1.3 Statement of Problem ....................................................................................... 2
  1.4 Research Questions .......................................................................................... 2
    1.4.1 Benefits of moving to Smart Cards ............................................................ 3
    1.4.2 Comparative Analysis of the Tertiary Sector .............................................. 3
    1.4.3 The Proposed University Wide Smart Card System ................................ 3
  1.5 Summary .......................................................................................................... 3
2.0 Literature review ................................................................................................. 4
  2.1 Introduction ....................................................................................................... 4
  2.2 Benefits of Smart Cards .................................................................................... 4
  2.3 Smart Cards in Tertiary Educational Institutions .......................................... 5
  2.4 The Proposed University Wide Smart Card System ....................................... 6
  2.5 Summary .......................................................................................................... 8
3.0 Benefits of moving to smart cards ..................................................................... 9
  3.1 Advantages of Smart Cards ............................................................................. 9
    3.1.1 Compatible with Existing Card Technologies .......................................... 9
    3.1.2 Advanced Security Against Duplication ............................................... 10
    3.1.3 Higher Data Storage Capacity ................................................................. 11
    3.1.4 Available in Smaller Form Factors ........................................................... 11
    3.1.5 Less Affected by Wear and Tear .............................................................. 11
    3.1.6 Less Subject to Data Degradation ........................................................... 12
    3.1.7 Access Hardware is More Compact ......................................................... 12
    3.1.8 Capable of Independent Computation ..................................................... 13
    3.1.9 Multi-Application Capable ...................................................................... 13
    3.1.10 Support Vendor and Application Specific Functionality ....................... 14
    3.1.11 Available in Contactless Form Factors .................................................. 14
  3.2 Disadvantages of Smart Cards ....................................................................... 14
    3.2.1 Access Hardware Can Be Expensive ....................................................... 14
    3.2.2 Smart Cards are More Expensive ............................................................ 15
    3.2.3 Access Hardware Can Wear Out From Continual Use ......................... 15
  3.3 Benefits for the Educational Tertiary Sector .................................................. 16
4.0 A comparative Analysis of the tertiary sector ................................................... 17
  4.1 Institutions Utilising Card Based Systems ....................................................... 17
  4.2 Facilities Provided From Card Based Systems .............................................. 17
  4.3 Australian Institutions Utilising Smart Cards ............................................... 18
4.3.1 La Trobe University .............................................................. 19
4.3.2 Technical and Further Education, New South Wales (TAFE NSW) ................................................................ 19
4.3.3 The University of Western Australia ........................................ 20
4.3.4 Edith Cowan University ........................................................ 20

4.4 International Institutions Utilising Smart Cards ...................................... 21
4.4.1 Florida State University ........................................................ 22
4.4.2 Michigan Technological University ........................................ 22
4.4.3 Stanford University .............................................................. 22
4.4.4 The University of Arizona ....................................................... 22
4.4.5 University of Central Florida ................................................ 22
4.4.6 University of Toronto ........................................................... 23
4.4.7 Waterford Institute of Technology ........................................... 23

4.5 Institutions That Previously Utilised Smart Cards ................................... 23
4.5.1 University of Michigan ......................................................... 23
4.5.2 Washington University in St. Louis .......................................... 24

4.6 Institution Smart Card Privacy Concerns ........................................... 25

4.7 Conclusions ............................................................................. 25

5.0 The Proposed university wide smart card system .................................. 26
5.1 Introduction .............................................................................. 26

5.2 Smart Card Access Hardware ................................................................ 28
5.2.1 Introduction ........................................................................ 28
5.2.2 Smart Card Access Hardware ................................................. 28
5.2.3 Smart Cards ......................................................................... 30
5.2.4 APIs, Smart Card Protocols and Software .................................. 31
5.2.5 Summary ............................................................................. 32

5.3 The Client Interface ...................................................................... 32
5.3.1 Introduction ........................................................................ 32
5.3.2 Requirements ...................................................................... 32
5.3.3 Options for Client Front Ends ............................................... 33
5.3.4 The Client Front End Proposal – Utilise HTML Web Browsers .... 33
5.3.5 Why HTML Web Browsers? ................................................... 34
5.3.6 Interactivity with HTML and Web Sites ................................... 34
5.3.7 Why not Java and Java Applets? ............................................ 36
5.3.8 Security and Access to Smart Cards ....................................... 36
5.3.9 The Client User Experience .................................................. 37
5.3.10 Summary ........................................................................... 38

5.4 Server Front End ........................................................................ 38
5.4.1 Introduction ........................................................................ 38
5.4.2 The Proposed HTML Web Server and Operating System ......... 38
5.4.3 Why Apache? ....................................................................... 39
5.4.4 Secure HTTP with SSL ......................................................... 39
5.4.5 Client Interaction with Session Based Handling ....................... 40
5.4.6 PHP for Session Management and Dynamic HTML Web Pages 42
5.4.7 Access to Smart Card Access Hardware and Smart Cards ....... 44
5.4.8 Summary ............................................................................. 45

5.5 Server Back End .......................................................................... 45
5.5.1 Introduction ........................................................................ 45
5.5.2 Requirements ...................................................................... 45
5.5.3 Integrating with Existing Institution Systems ............................ 45
5.5.4 The Solution – Directory Services ......................................... 46
1.0 INTRODUCTION

1.1 Background

Within tertiary educational institutions, the use of plastic cards based on technologies such as magnetic stripes or barcodes for identification purposes is well known and appreciated among staff and students. These institution specific cards closely resemble the familiar bank credit and automatic teller machine (ATM) cards, or government issued identification cards, such as a driver’s license.

These cards are primarily utilised for identification of an institution’s staff and students. However, they have numerous other possible purposes, such as enabling the owner to borrow library materials or allowing access to certain secure areas.

Advances in card technologies have enabled integrated circuits (IC’s) to be embedded into existing plastic cards. Cards such as these are known as Smart Cards. The rising popularity of Smart Cards has enabled a potential new and exciting range of possible applications to be developed and utilised by institutions that were previously not possible on existing card based technologies.

1.2 Significance

As Smart Card technology is still a relatively new and untested technology within the tertiary sector, any institution considering adopting Smart Card technology over their existing card based technologies should investigate this course of action carefully.

In particular, close attention to the possible advantages to Smart Card technologies will determine if the institution will require them in the first place, or whether it is still sufficient to rely on existing card technologies.
If the institution can see itself beginning to utilise Smart Cards, it should closely analyse other institutions that have successfully adopted these new card technologies. The institution should also closely analyse those that did use Smart Cards, but failed to deploy them successfully.

Finally, the introduction of Smart Cards should be handled as a slow migration process across the institution. To this end, possible strategies for handling this migration should be looked into in order to maximise the efficiency of adoption of the new cards.

Each of these areas should be closely researched and analysed in order to determine if indeed Smart Cards should be used within the institution, and if so, how best to implement a successful migration towards using them.

1.3 Statement of Problem

The aims of this thesis are to provide tertiary educational institutions that are considering moving to Smart Cards suitable advantages as to why they should use Smart Cards; details of other institutions that are using Smart Cards; and details of a proposed system that can help facilitate the adoption of Smart Cards within the institution.

1.4 Research Questions

This thesis will attempt to answer three primary questions that tertiary educational institutions are likely to encounter when considering the adoption of Smart Cards as their primary identification card. These questions are as follows and are described briefly in the following sections of this thesis.

1. What are the benefits of moving to Smart Cards?
2. Perform a comparative analysis of the tertiary sector.
3. Propose a suitable University wide Smart Card system.
1.4.1 Benefits of moving to Smart Cards.

This chapter of the thesis will attempt to detail possible advantages that an institution would obtain if it were to adopt Smart Card technology over existing card based technologies. In addition, this chapter details some of the possible disadvantages an institution would face if it did indeed adopt Smart Cards.

1.4.2 Comparative Analysis of the Tertiary Sector.

To facilitate an institution's decision on Smart Card technologies, this chapter details other institutions that have successfully migrated to using Smart Cards as their primary identification card and their current applications. In particular, this chapter also details some of the unsuccessful attempts and major incidents that occurred so that previous mistakes are not repeated and can be avoided.

1.4.3 The Proposed University Wide Smart Card System.

This chapter details a proposal for building a Smart Card system that would successfully migrate and integrate with existing institution systems. This proposal considers backward compatibility with existing institution systems, as well as the development of an institution wide data directory that would help facilitate the use of Smart Cards.

1.5 Summary

This thesis will attempt to provide pertinent information on available Smart Card and related technology, information on other tertiary educational institutional efforts on utilising Smart Card technology and a possible proposal on how best to integrate Smart Card technologies with existing institutional systems.
2.0 LITERATURE REVIEW

2.1 Introduction

As an essential part of researching the three main questions posed by this thesis, it was necessary to examine and review available literature material for suitability for inclusion into this thesis. There are three sections to the literature review corresponding to each of the questions posed.

2.2 Benefits of Smart Cards

This section details the books, organisations and companies that were of use in determining the benefits of Smart Cards over existing card based technologies. While researching Smart Card capabilities and possible applications, there were several books available that I was able to reference from.

"Smart Cards" by José Luis Zoreda and José Manuel Otón (Zoreda & Otón, 1994) provides a good comparison of emerging Smart Card technologies, including existing card technologies such as magnetic stripe cards and optical cards. In addition to this, it also provides in depth information on how to access and program Smart Cards. Of interest is the in depth research of Smart Card applications and their uses at the time of the book being written in 1994.

However, due to the rapid advancements in Smart Card technology there existed no authoritative book on new emerging Smart Card technologies for some years after 1994. The "Smart Card Handbook" by Rankl Wolfgang (Wolfgang & Wolfgang, 1997) is one of the most authoritative books on Smart Cards presently and attempts to fill this gap. This book focuses primarily on defining the Smart Card itself; from the physical form factor sizes to the protocols used for communicating to them and the various instruction sets used to program processor based Smart Cards.
The “Smart Card Handbook” is not fully authoritative on Smart Card security systems nor does it provide any adequate example applications for utilising Smart Cards. For this, it is recommended to read “Smart Card Security and Applications” by Mike Hendry (Hendry, 1997) that provides a comprehensive in-depth study into Smart Card security. This includes techniques such as public/private key encryption and biometrics that are not fully documented elsewhere. In addition, this book also contains useful example applications and documentation on how Smart Cards can be utilised within the industry.

To find companies manufacturing and utilising Smart Cards, it is possible to obtain up to date details by searching for forums or groups of companies that specialise in Smart Card technologies. As an example, the Asia Pacific region has a strong Smart Card initiative and has several groups such as the Asia Pacific Smart Card Forum (smartcardforum.asn.au, 2003) and the Asia Pacific Smart Card Association (apsca.org, 2003). Internationally, there are Smart Cards Online (smartex.com, 2003) and the Smart Card Group (smartcardgroup.com, 2003) that have membership details of Smart Card related companies.

By utilising the on-line membership databases it is possible to discover numerous companies, businesses and manufacturers of Smart Cards and associated hardware, some of who are referenced in this thesis.

2.3 Smart Cards in Tertiary Educational Institutions

While researching for the comparative analysis of tertiary educational institutions utilising Smart Cards, it was necessary to discover which institutions were in fact using Smart Cards. This proved to be rather difficult as there are no easy means of discovering who is using Smart Cards or not, because of the numerous institutions utilising card based technologies.

To assist in discovering possible tertiary institutions utilising Smart Cards, it was possible to use the members links from the National Association of Campus Card Users (NACCU, 2003c) to provide a list of tertiary institutions that use an ID card of one form or another, however this may not necessarily be a Smart Card.
Fortunately, research into Australian tertiary institutions and their use of Smart Cards uncovered a paper by Janaka Welikala, Danielle Fowler and Paula Swatman called “Introducing Multi-purpose, Multi-function Smart Cards to Australian Universities” (Welikala, Fowler, & Swatman, 1997). Of most benefit were numerous references to International and Australian tertiary institutions that were using Smart Cards. In addition to this paper, Queensland University of Technology’s Smart Card home page (qut.edu.au, 2000) also provided many references and links to related institutions. Using these sources, it is possible to obtain a considerably more concise list of tertiary institutions that have some connection to using Smart Cards, but it is by no means a comprehensive one.

Using this more accurate list of tertiary institutions, it was then possible to investigate each one and determine whether they were indeed using Smart Cards. It was also possible to research what sort of applications were in use, as well as any other information concerning the success and failures they experienced while implementing Smart Card technologies. Information of this nature was documented in various institution journals or by the public media.

2.4 The Proposed University Wide Smart Card System

While researching options for the proposed University wide Smart Card system, the primary concern was to discover and find ways to integrate the existing institution systems into the new proposed system. After some research, there appeared to be a successful migration at the University of Michigan (umich.edu, 2003b) to a new directory protocol called LDAP, or the Lightweight Directory Access Protocol.

Following up on research into LDAP, initially there were few books or documentation available. However, the growing popularity of LDAP has produced several excellent books that can assist in designing and deploying LDAP services and provide ideas for providing new opportunities and applications while maintaining backwards compatibility with existing systems.

The primary book for obtaining the necessary prerequisite information about LDAP, in particular, what LDAP can be used for and how to deploy it are well documented in “LDAP: Understanding and Deploying LDAP Directory Services” by
Timothy Howes, Mark Smith and Gordon Good (Howes, Smith, & Good, 1999). This book has detailed information and case studies for designing, deploying and using LDAP, as well as in-depth discussions on related issues such as security, namespace and schema design as well as replication and backup issues.

This is a very in depth book that has recently been brought out in a 2nd edition. Unfortunately, there is very little programmer or systems administration specific information in this book. This requires that other sources of material are needed to fully comprehend how LDAP can be fully utilised.

The following two books are specifically written for programmers, and are “LDAP: Programming Directory-Enabled Applications with Lightweight Directory Access Protocol” by Timothy Howes and Mark Smith (Howes & Smith, 1997), and “LDAP Programming with Java™” by Rob Weltman and Tony Dahbura (Weltman & Dahbura, 2000).

“LDAP: Programming Directory-Enabled Applications with Lightweight Directory Access Protocol” targets C programmers who wish to get straight into accessing LDAP servers and writing directory enabled applications that can take advantage of LDAP.

“LDAP Programming with Java™” focuses on accessing the Netscape Directory Server (netscape.com, 2003b) utilising Java to build LDAP enabled applications. It also provides insight into using Java servlets to build web sites that can integrate with LDAP services.

Unfortunately, neither book provides actual details of how to administrate your own LDAP server, or how to build a suitable system that includes a LDAP server. Previously, systems administrators were required to reference numerous web sites and online documentation to learn how to set up and use LDAP servers. Fortunately, recently a new book has arrived to fill in the missing gaps.

“LDAP Systems Administration” by Gerald Carter (Carter, 2003) provides systems administrators with concise information on how to set up and configure the free OpenLDAP server (openldap.org, 2003). Previously, information in this book was available, however it was not readily accessible unless collected individually.

This book lacks the required information on how to design and deploy LDAP services, so it is necessary to obtain “Understanding and Deploying LDAP Directory
Services” in order to obtain a more complete understanding of LDAP and how it can be used.

2.5 Summary

The above review of available literature clarifies some of the research accomplished that otherwise would have been not as well documented as originally hoped. It should provide an ample base of knowledge for people interested in implementing the proposed University Wide Smart Card System to continue their research and avoid some of the mistakes and failures from previous attempts.
3.0 BENEFITS OF MOVING TO SMART CARDS

3.1 Advantages of Smart Cards

Having reviewed the most common literature and manufacturer documentation on Smart Cards, there are some very specific advantages that Smart Cards have over existing card based technologies. The following is a list of most of the advantages that Smart Cards would have over existing card based technologies.

3.1.1 Compatible with Existing Card Technologies

Smart Cards in general have the same ISO 7810 form factor (Wolfgang & Wolfgang, 1997) as existing card technologies, such as credit or ATM cards. It is entirely possible to create Smart Cards that also support existing card applications with the inclusion of visual information, such as pictures and barcodes, or magnetic stripes on the back of the Smart Card.

ID-1 format, as specified by the ISO 7810 standard is the most widely used form factor and is the same one used by banks for ATM and credit cards. ID-00 is an intermediate form factor, while ID-000 is the form factor favoured by mobile phone manufacturers due to its small size.

<table>
<thead>
<tr>
<th>Card</th>
<th>Width</th>
<th>Height</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID-1</td>
<td>85.6 mm</td>
<td>54 mm</td>
</tr>
<tr>
<td>ID-00</td>
<td>66 mm</td>
<td>33 mm</td>
</tr>
<tr>
<td>ID-000</td>
<td>25 mm</td>
<td>15 mm</td>
</tr>
<tr>
<td>ID-000 relative to ID-00</td>
<td>6.25 mm</td>
<td>16.4 mm</td>
</tr>
</tbody>
</table>

Note that ID-1 and ID-00 have a bevel of radius 3.18 mm, and that ID-000 has a 3 mm cut to provide correct alignment.

Figure 1 ISO 7810 card form factor sizes.
As an example, cards at institutions typically utilise a barcode printed onto the card to provide library facilities and may also utilise a magnetic stripe to provide security access, or to provide other facilities such as ATM access or a simple magnetic stripe based debit card.

It is a relatively simple matter to print the required visual information required for existing technologies or to purchase Smart Cards that already have an existing magnetic stripe ready for use. This would add at least one to two compatible systems to the Smart Card. More could be added, depending on how existing systems operate and whether it can be accommodated onto the Smart Card itself.

3.1.2 Advanced Security Against Duplication

Recently, it has been possible to duplicate visual based cards easily with relatively low cost equipment. In particular, a barcode on a card could be photocopied and misused quite easily, especially with unattended photocopying machines found in libraries. Magnetic stripe cards fare no better, and can be read and copied verbatim with any relatively sophisticated equipment that can read or write the magnetic stripe, with blanks being cheap and easily obtained.

By contrast, Smart Cards have to be read and written using strict electronic protocols that can effectively isolate the in-built integrated circuit (IC) from trivial duplication, and are capable of using protection such as PIN codes, on-board processing of security to sections of memory, encryption and various other application and card specific security measures.

Smart Cards can also have volatile memory and physical security so that attempting to reverse engineer the IC on the card can result in the memory contents being irretrievably lost in the process. As reverse engineering attempts improve, so does the manufacturing of Smart Cards to make it more and more difficult to bypass the Smart Card’s in-built security measures.

As an example, the card used in mobile phones known as the SIM or GSM Smart Card has a PIN code that is required before data can be read or written to the card itself. This provides reasonable security against casual use or duplication if the card is lost or stolen.
3.1.3 Higher Data Storage Capacity

A Smart Card can effectively store more data than is possible on existing card technologies, such as name, address, visual identification and other related information. Sizes of 1 KB to 32 KB or more data on Smart Cards are commonplace. This amount of data if physically printed on the card would consume more surface area than the card could support. By comparison, magnetic stripe cards can at most store about 1 KB of data. However, there are exceptions – some optical cards are capable of storing up to 6 MB of information (Wolfgang & Wolfgang, 1997).

As an example, the high end microprocessor cards such as GemPlus GemXplore ‘Xpresso (GemPlus, 2003e) can store 32 to 64 KB of data as a high end card in addition to being able to run Java byte code programs. More commonplace memory cards such as the GemPlus GPM8K (GemPlus, 2003g) store 8 KB of data, however 16 KB memory cards are not uncommon either.

3.1.4 Available in Smaller Form Factors

In addition to higher storage capacity, Smart Cards are also capable of being reduced to almost the size of the electrical contact itself, known as the minicard form factor or ID-000 as detailed above. This facility has proven to be most successful in the mobile phone market where the card size is considerably constrained. Manufacturers of contactless Smart Cards also have been successful, producing even smaller, more portable and durable versions of Smart Cards such as the GemPlus KeyFob8000 (GemPlus, 2003d) and Dallas iButton (Dallas, 2003a).

3.1.5 Less Affected by Wear and Tear

Visually dependent cards require anywhere from low to high integrity of the image on the card, depending on the amount of data stored visually. Naturally, damage to the card itself on any of the visual portions of the card will affect read-back of data and may require the card to be replaced.

Magnetic stripe cards are subject to wear and tear by constant swiping of the card through access devices. Eventually these cards can have their magnetic stripe wear away or the data can gradually degrade and become error-prone or unreadable.
Smart Cards are unaffected by visual degradation of the card or by normal wear and tear on the card’s printed area with the exception being damage to the electrical contact, which can cause the card to malfunction.

Contactless Smart Cards are even more durable; many are protected in a non-card form factor or have no electrical contact at all, with the IC embedded completely inside the card. In general, Smart Cards are much less likely to stop working from wear and tear than other current card technologies.

3.1.6 Less Subject to Data Degradation

Magnetic stripe cards are prone to erasure or modification of stored data if the card is near a powerful electro-magnetic source, in particular if they are low coercivity type cards which are created with a relatively weak magnetic recording process. High coercivity magnetic stripe cards (Hendry, 1997) are less likely to be erased or modified because of a considerably stronger recording process. Magnetic stripe cards are also subject to a slow degradation of the magnetic recording, so over time data may become error prone as the signal becomes unreadable.

Smart Cards are not based on simple magnetic data storage and instead utilise non-volatile memory. As a result, they are considerably less likely to be affected by electro-magnetic interference and are generally well protected against adverse environments and static electricity due to the specially designed contact area.

3.1.7 Access Hardware is More Compact

For visual based cards, the equipment required to read a card can be quite bulky. This can require a video camera and a processor computer to a simple barcode scanner with a dedicated microcontroller. This however does take up space and requires visual synchronisation to read the data on the card correctly.

Magnetic stripe card readers fare no better and are generally quite large compared to Smart Card access devices. The card reader also provides alignment and ensures that the card is read properly. Cards must be swiped uniformly through the device - too fast or too slow may cause data reading errors to occur.

By comparison, Smart Card access devices can be very small, notable mechanisms are the SIM or GSM card readers in mobile phones that can be as small
as the contact area of the Smart Card itself. Other Smart Card access devices are small enough to fit comfortably onto a key ring and can accept full sized Smart Cards. (Hendry, 1997)

3.1.8 Capable of Independent Computation

Advances in the design and manufacture of Smart Card IC's mean that it is now possible to embed complex microprocessors into a Smart Card. Examples of these high end Smart Cards include cards with an on-board Java implementation (GemPlus, 2003e), encryption cards with on-board key exchange and encryption and decryption support (GemPlus, 2003f), memory cards with password protection (GemPlus, 2003b), auto decrementing charge cards (GemPlus, 2003a) and other related products.

By contrast, visual and magnetic stripe cards have no on-card processing capabilities at all and rely on microcontrollers or processors attached to the card device to handle all processing facilities.

3.1.9 Multi-Application Capable

As Smart Cards have higher data capacities and greater security in addition to the ability to segment available memory so that access to certain memory sections can be tightly controlled, it is possible for more than one application to be applied to a card at any one time. This has advantages for vendors endorsing their particular Smart Card as it enables them to secure their data while allowing affiliated companies to be able to utilise their Smart Card as well.

With memory segmentation and separate access mechanisms it is possible to have two or more separately controlled memory sections for different companies or applications on one Smart Card. For example, a vendor specific supported cash card can have several secure separate areas of memory. This enables the Smart Card to store information specific to several vendors separately, making the card more secure than existing card technologies. The main difference is that the security is handled on the card, and not in the access device.
3.1.10 Support Vendor and Application Specific Functionality

Smart Cards also support application specific functionality on the card itself. This is of benefit where advanced security is required, or where vendors require a specific function such as a disposable auto-decrementing card (for a prepaid telephone card) where a simple memory based card would be able to be duplicated trivially and subsequently be prone to abuse and fraud.

3.1.11 Available in Contactless Form Factors

Recent advances in contactless Smart Cards (GemPlus, 2003d) have enabled new cards to be used with no physical contact to an access device. This is of benefit where card wear and tear can cause mechanical difficulties with access devices. The contactless nature of Smart Cards means that frequently used Smart Card access devices such as Point of Sale (POS) terminals and security access terminals can be migrated or designed as a contactless based system. This provides greater reliability and enables the access device to be hermetically sealed to avoid wear and tear, deliberate vandalism or sabotage.

3.2 Disadvantages of Smart Cards

Although Smart Cards bring many advantages to tertiary educational institutions, there are some disadvantages that should be listed. This is in order to provide a balanced view and should be taken into consideration if they should be considered for adoption by any institution.

3.2.1 Access Hardware Can Be Expensive

Access hardware is expensive, depending on the type of Smart Card and whether it is contact based or contactless. Many contact based Smart Card access hardware devices typically cost $50 to $100, or even up to $250. Economies of scale and widespread use mean that existing card readers are cheaper for barcode and magnetic stripe cards. For contactless Smart Cards, the costs are even higher, both for the access hardware and card itself.
Fortunately, iButtons and other different form factor Smart Cards use low cost parallel, serial or USB port readers that can utilise simpler access protocols. This enables access hardware to leverage on generic in-built hardware of the PC rather than including extra hardware in the access device itself. This allows considerably more cost effective designs for access hardware. (Dallas, 2003b)

Recent development and interest in Smart Cards has meant that the cost of contact based Smart Card access devices has dropped in price considerably. For example, the product “SIM Manager” from Smart Silicon Systems is only now $40, complete with software and Smart Card access hardware (simmanager.com.au, 2003). This enables the user to load and save details off their mobile phone SIM/GSM card.

3.2.2 Smart Cards are More Expensive

Smart Cards can be considerably more expensive than simple barcode or magnetic stripe cards, depending on configuration and requirements. Simple memory based cards with no security can cost about $1, while more advanced memory cards cost up to $5 or more and microprocessor based cards can cost up to $100.

Factoring this in among the number of users, this can discourage adoption of the card or cost considerably more to get their card widely spread among the users. In particular, the replacement cost of even the most basic Smart Cards was about $20 - $50 Australian in tertiary educational institutions, which is a fair replacement value for their ID cards.

3.2.3 Access Hardware Can Wear Out From Continual Use

For contact based Smart Cards, the constant insertion and removal of the Smart Card for simple contact based access devices eventually wears the contacts out, forcing the access device to be replaced frequently depending on use. This can cause issues if the device wears out at an inconvenient time or if it begins to damage the cards itself.

A solution to this is to switch to contactless Smart Cards, or to mechanised Smart Card access devices that have more complicated contact mechanisms that
avoid this sort of wear and tear. Unfortunately, the more advanced Smart Card access devices do cost more to purchase and obtain.

3.3 Benefits for the Educational Tertiary Sector

Having reviewed several advantages of utilising Smart Cards, the primary advantage is that the tertiary educational institution would be able to utilise the Smart Cards as true multi-application ID cards.

If a Smart Card with sufficient memory, processing capability and memory security access has been chosen, it is possible for many different applications to be developed. For independent organisations within the institution, this enables their own applications to be developed and utilised, ideally with a minimum of modifications to the Smart Card infrastructure that has been put in place.
4.0 A COMPARATIVE ANALYSIS OF THE TERTIARY SECTOR

4.1 Institutions Utilising Card Based Systems

With the considerably widespread use of card based technologies, it is a natural expectation that Universities and other tertiary education institutions would adopt the use of similar cards with similar functionality. At the moment, use of cards as visual identification, in particular student and staff ID cards, is prevalent among such institutions. In addition, many institutions use visual barcodes or stickers to add services such as borrowing library materials and to obtain discounts from institution affiliated stores and service outlets in addition to other possible benefits.

Within the United States of America (USA) alone, a group called the National Association of Campus Card Users (NACCU) (NACCU, 2003c) was formed in 1993 to help co-ordinate and collect information about institutions using card technologies. NACCU shows that there are over two hundred listed institutions (NACCU, 2003a) that use card technologies of one form or another to facilitate institutions working with numerous students and staff. However, NACCU claims (NACCU, 2003b) that they have over four hundred and thirty members, this demonstrates that card technology is well and truly integrated into educational institutions.

4.2 Facilities Provided From Card Based Systems

Today many educational institutions utilise card systems for the following facilities, which may require one or more cards depending on the institution’s level of integration with the following facilities:

1. Staff and student identification cards (visual photograph and/or signature);
2. Library borrowing facilities;
3. Access to buildings or secure areas;
4. Discounts for public transportation;
5. Discounts to on or off-campus stores and affiliated merchants;
6. Discounts for campus related societies such as the student guild and gym;
7. Access to restricted information, such as student records;
8. Generic electronic cash or electronic purse facilities;
9. ATM or bank linked facilities;
10. Meal plans;
11. Photocopying, printing and faxing facilities;
12. Internet access payments;
13. Vending facilities for food, beverages and other small items;
14. Telephone access facilities via calling card numbers;
15. Discounted special events and activities.

Depending on the level of sophistication of the technology used or required, staff and students of institutions may carry one or more cards as required. For instance, as a student at Edith Cowan University in 1995 it was possible to carry one student ID card that also doubled as a library borrowing card and discount public transport card. There was also a separate magnetic stripe card for photocopying and printing facilities in local University libraries in addition to another magnetic stripe card used for security access (now replaced with a contactless Smart Card).

Clearly, there is room for improvement, as Smart Cards can provide all the previously listed facilities above on a single card. Many institutions are currently looking at replacing their existing card technologies with a card based on a combination of technologies such as the magnetic stripe, visual identification and Smart Card. These technologies all fit on one card to reduce the number of cards in circulation and for the user's convenience.

4.3 Australian Institutions Utilising Smart Cards

Currently within Australia the most well known institutions actively using Smart Cards are La Trobe University, the University of Western Australia and NSW
TAFE. Considering the number of educational institutions in Australia compared to the USA, Australia compares favourably as being on the leading edge of technology in the adoption of Smart Cards for tertiary institutions. In addition to these institutions, Edith Cowan University is also shown to currently use Smart Cards.

4.3.1 La Trobe University

La Trobe University, in Melbourne, Australia (latrobe.edu.au, 2003b) introduced Smart Cards in 1999 for the start of the year 2000 (latrobe.edu.au, 2000) (ERG, 1999) to staff and students of the University through Australia’s prominent Smart Card provider ERG (ERG, 2003) and their affiliated company ECard (ECard, 2003).

Currently, the La Trobe Card (latrobe.edu.au, 2003a) provides a contactless Smart Card that is also a photographic ID card for students and staff. Library borrowing facilities are provided by a barcode printed onto the card. Security access is controlled by contactless proximity sensors linked to doors and communicate to the card wirelessly.

The La Trobe Card also supports e-purse facilities for photocopying and printing as well as small purchases through related merchants located on campus. Facilities such as Cash-to-Card machines are provided to reload monetary values onto cards as required. Replacement cards can be purchased for $20 Australian.

Currently, the La Trobe Card does not support advanced functionality such as ATM facilities or telephone calling card facilities. However, at the moment the La Trobe Card is the most advanced Smart Card in use at any education institution inside Australia.

4.3.2 Technical and Further Education, New South Wales (TAFE NSW)

The Technical And Further Education, New South Wales (tafensw.edu.au, 2003b) has campuses located throughout NSW, Australia for assisting people with further education outside of the traditional University environment. Introduced in 1999, the TAFECard (tafensw.edu.au, 2003a) is essentially a highly commercialised Smart Card with strong links to industry such as Yalo, Telstra, ECard and Getronics Australia (dialelectronics.com.au, 2001).
The TAFECard is used as a traditional visual ID card in addition to being a secure access card and discount card at selected merchants. In addition to this, the TAFECard supports e-purse facilities and supports Telstra phone booths as a calling card for student and staff calls.

Currently, the TAFECard is widely supported on and off TAFE NSW campuses thanks to the ERG initiative program called Yalo (Yalo, 2003a). Yalo supports numerous merchants (Yalo, 2003b) who are able to support the TAFECard and provide additional benefits and incentives to students.

4.3.3 The University of Western Australia

The University of Western Australia, in Perth, Western Australia (uwa.edu.au, 2003a) recently introduced Smart Cards in 2001 (uwa.edu.au, 2000) as the primary ID card for staff and students. The cards and related technology were obtained from UniCard (UniCard, 2003) who have a strong history of providing card based facilities to educational institutions within Australia.

At present, the UWA Campus Card (uwa.edu.au, 2003b) provides similar facilities to the La Trobe Card, however it is based on a simpler contact based Smart Card and has no magnetic stripe facility. Currently, the card provides photographic ID and barcode based library borrowing facilities. In addition, it provides security access using the same contact based Smart Card. Lost cards cost $15 Australian to replace.

The UWA Campus Card also provides e-purse facilities for photocopying, printing and Internet access. UWA provides several Cash-to-Card machines located in publicly accessible areas. At the moment, the UWA Campus Card is rather limited, however we expect to see new applications developed for it over time.

4.3.4 Edith Cowan University

Edith Cowan University in Perth, Western Australia (cowan.edu.au, 2003) presently does not utilise Smart Cards as their primary ID card for staff and students, however it does utilise contactless card technologies for its primary security access control system (ecu.edu.au, 2003a).
At present, Edith Cowan University has several research groups that investigate Smart Card technologies in the School of Computer & Information Science (ecu.edu.au, 2003b) including the Information Technology Systems group (ecu.edu.au, 2003b) as well as the Internet and Computer Security Lab group (ecu.edu.au, 2003b).

The School of Computer & Information Science also has strong interests in Smart Cards as the current Executive Dean for the Faculty of Computing, Health and Science, Professor Tony Watson (ecu.edu.au, 2003b) is well known for his interests in Smart Cards and related technologies.

It is to be expected that at some point in time Edith Cowan University would possibly migrate their staff and student ID card system to merge with their current security access system to provide a single Smart Card solution.

4.4 International Institutions Utilising Smart Cards

Internationally, there are other high profile tertiary institutions that have implemented Smart Cards over existing visual and magnetic stripe based card technologies. Most implementations are located in USA, Canada or Ireland, which is not surprising given their high level of technological development during the information technology boom of the last 1990’s.

All the following tertiary institutions support the accepted standard applications of visual ID, security access, library borrowing, e-purse facilities such as photocopying, printing and small purchases and discounts through approved on and off campus merchants. To support e-purse facilities, the institutions also support Cash-to-Card machines on campus.

The following list is not conclusive however, as research is rather difficult because some tertiary institutions have either dropped the use of Smart Cards in favour of visual and magnetic stripe cards only, or have ambiguously (or mistakenly) labelled their own cards as “smart” cards.
4.4.1 Florida State University

Florida State University (fsu.edu, 2003a) first began its Smart Card program in 1994, developing the FSUCard (fsu.edu, 2003b). In addition to providing all of the standard applications, the FSUCard also supports telephone calling card numbers, printed onto the card itself. Services for making telephone calls are provided by The Seminole Circuit, a popular telephone service provider. In addition to this, the FSUCard also supports direct ATM access by virtue of its magnetic stripe, and is linked to SunTrust Banks Inc.

4.4.2 Michigan Technological University

Michigan Technological University (mtu.edu, 2003a) also supports a contactless Smart Card solution they call the Tech Express Card (mtu.edu, 2003b). The Tech Express Card supports the standard services provided by existing card technologies.

4.4.3 Stanford University

Stanford University (stanford.edu, 2003b) has replaced existing visual and magnetic card based technologies with a Smart Card based system called the Stanford Campus Card (stanford.edu, 2003a). Since June 2002, the Stanford Campus Card has been converted to a proximity Smart Card system (stanford.edu, 2002).

4.4.4 The University of Arizona

The University of Arizona (arizona.edu, 2003a) introduced the CatCard (arizona.edu, 2003b) in March 1998. Currently, it supports all major card services and applications in addition to car parking e-purse facilities. In addition to this, at the start of 2003 the CatCard was linked to Wells Fargo Bank to provide ATM facilities.

4.4.5 University of Central Florida

The University of Central Florida (ucf.edu, 2003a) supports the UCFCard (ucf.edu, 2003b) which is primarily a staff and student ID card, library borrowing card and an e-purse facility card. It does not support other applications at the moment, such as secure building access.
4.4.6 University of Toronto

The University of Toronto (utoronto.ca, 2003a) in Canada has its own TCard (utoronto.ca, 2003b) that supports all expected card applications and services. As the University is rather large, not all applications are supported at all locations, presumably due to difficulty in co-ordinating services and facilities for fully supporting the TCard.

4.4.7 Waterford Institute of Technology

The Waterford Institute of Technology (wit.ie, 2003a) in Ireland has replaced its existing cards with the new Smart Card they call the WITCard (wit.ie, 2003b) that was introduced in September 1999. In addition to standard applications the WITCard also supports independent vending machines and voicemail telephone support, so that students can access voicemail from any phone on campus or access e-purse facilities for food and beverages.

4.5 Institutions That Previously Utilised Smart Cards

Some tertiary institutions have used or tested Smart Cards and for various reasons decided against continuing to use them. This section details some of these institutions unsuccessful attempts to adapt to Smart Card technology.

4.5.1 University of Michigan

The University of Michigan (umich.edu, 2003b) was one of the first Universities to begin using Smart Cards. The MCard (umich.edu, 2003a) is the University of Michigan's own Smart Card implementation that they researched and introduced in 1994 and was one of the largest users of Smart Cards at that time (umich.edu, 1996b),

However, the University of Michigan decided to end the use of Smart Cards and officially withdrew all support in the middle of 2001 (umich.edu, 2003c) and began replacing existing cards with their new yellow MCard that is magnetic stripe based only.
Various reasons were discussed for the failure of the MCard Smart Cards, but it appears that strong student refusal to exclusively use the MCard due to perceived commercialisation of the MCard as well as implementation problems stopped the MCard from being as successful as it could have been. (Keating, 1995) (umich.edu, 1996a) (Weissert, 1996) (Schillaci, 1997)

Some of the implementation issues that were noted in local publications included student dissatisfaction at the implementation of the cash chip feature that meant it was difficult for a student to extract cash from the chip if necessary.

Other problems with the cash chip feature meant that it was not as universally accepted as first hoped, causing students to prefer to use cash over the MCard as it struggled to overcome the inherent merchant dominance of existing bank ATM cards, credit cards and cash.

Students also expressed dissatisfaction at the commercial monopolisation of the MCard by the First Bank of America and AT&T, meaning many students with alternative banks and telephone services were unhappy with how the MCard eliminated their options for better services and lower fees.

4.5.2 Washington University in St. Louis

Washington University in St. Louis (wustl.edu, 2003) also investigated using Smart Cards beginning in 1995 and ending four years later, in the middle of 1999. The University decided against continuing to use Smart Cards due to the lack of acceptance by the staff and students in addition to implementation issues. However, there were some specific areas where the Smart Card was very successful, in particular the on-chip electronic cash facility for fixed-fee services such as laundry payments. (Everding, 1995) (Everding, 1999)

Again, the lack of universal acceptance and reliance on more popular forms of payments such as ATM cards, credit cards and cash hindered the use of the Smart Cards and failed to catch on with the students with a few exceptions as detailed above. Since existing services such as security access and meal plans were still operating on magnetic stripe technology, it was a simple matter to remove the Smart Cards and revert back to the magnetic stripe technology alone.
4.6 Institution Smart Card Privacy Concerns

In addition to some failures to successfully promote Smart Card technologies to tertiary institutions, there were also some privacy concerns that were discovered among the University of Arizona’s (arizona.edu, 2003a) Smart Card implementation called the CatCard (arizona.edu, 2003b).

Terrence Bressi, an employee of the University of Arizona upon obtaining the CatCard discovered after some investigation that the University of Arizona had released confidential information such as names, addresses and Social Security Numbers (SSN) in direct violation of the University of Arizona’s policies. The original document concerning this was emailed on 18th of March 1998 (Bressi, 1998a).

Eventually the University of Arizona was forced to make drastic changes to the implementation of the CatCard as Terrence Bressi escalated the issue and refused to accept the CatCard in its original implementation. For further information on this particular incident, it is recommended to read Terrence Bressi’s web site concerning the CatCard (Bressi, 1998b).

This raises important issues concerning privacy that can be easily overlooked when implementing Smart Cards at a tertiary institution. It is highly recommended that any institution considering the use of Smart Card and related technologies fully assess the privacy concerns of staff and students of the institution to avoid possible costly legal and implementation issues.

4.7 Conclusions

The use of card technologies in tertiary institutions has been shown to be extremely popular. There is a growing recognition that it is possible to utilise Smart Card technologies to consolidate multiple cards into one card for each student and staff member of the institution. In addition to this, Smart Cards are capable of providing new and improved facilities and applications previously unobtainable on existing card technologies.
5.0 THE PROPOSED UNIVERSITY WIDE SMART CARD SYSTEM

5.1 Introduction

As the final research question for this thesis, this chapter proposes a possible system that could be used in a University or other tertiary educational institution. Particular emphasis is placed on the system’s ability to be backwards compatible to the institution’s existing systems, and to be of low cost, expandable and scalable to fit the requirements of the institution.

As every institution is different, the proposed system detailed in this chapter should be taken as an outline of how a system could be developed and used successfully within the institution. Institutions looking to develop such a system should consider their own requirements carefully.

As an overview, the proposed system is a HTML or web based client server solution. The clients are primarily expected to be Personal Computers (PCs) with or without Smart Card access hardware. These PCs would then utilise a web browser to communicate to the proposed system’s servers. These would then be able to provide the client with their required information.

On the server side of the proposed system, the servers primarily run web servers to handle client requests. Software on the server would handle access to the client’s Smart Card. This software also handles information requests to the server back ends, which are primarily databases or other information storage systems.
This chapter has sections detailing each of the main parts of the proposed section, which are the Smart Card Access hardware, which details possible hardware that the institution would be recommended to purchase to support Smart Cards.

The Client Interface, which details the client and communications to the server, as well as communications to the Smart Card access hardware. The Server Front End handles communication to the clients, while the Server Back End handles data storage and information requests from the server front end. Each of the above sections of the proposed system will be detailed later in this chapter.
5.2 Smart Card Access Hardware

5.2.1 Introduction

In the proposed system, there are currently three types of clients that can connect into the system utilising Smart Cards. The first are computers that have Smart Card access devices, such as those used by the staff and students of the University for general use, administration and security of the system.

The second type are embedded systems utilised in the University that are expected to use Smart Cards, such as photocopying and printing machines, security access and vending facilities. These types of clients generally have strict requirements for what form of Smart Card access hardware can be connected to it, and will only be briefly discussed.

The third type of client is one that has no Smart Card access at all, and must operate with a reduced level of access and services while connected to the proposed system. This will be discussed further as part of the server section of the proposed system.

This section primarily aims at discussing possible Smart Card access hardware that can be utilised by the first type of client, where a computer is used in conjunction with a Smart Card access device to provide enhanced access to the proposed system.

5.2.2 Smart Card Access Hardware

To successfully utilise Smart Cards in the proposed system, each of the client computers will have a Smart Card access device connected to it. This is required to read, write and process data stored on the Smart Card. Smart Card access devices usually are capable of connecting to a standard Personal Computer (PC), so the various connection interfaces listed below may be used:

1. Serial;
2. PS/2;
3. Universal Serial Bus (USB);
4. PCMCIA;

Serial

A PC typically comes standard with one or two serial ports. These are the traditional interfaces for numerous low cost, low speed devices. Adding additional ports is a simple task with many standard ISA or PCI expansion cards available at low cost.

An example of a serial access device would be the SIM Manager (simmanager.com.au, 2003) Smart Card reader for mobile phone SIM/GSM cards. This device connects to a standard PC serial port and PS/2 port for power, and can handle reading and writing mobile phone numbers and SMS messages to a mobile phone SIM/GSM card.

PS/2

This interface is primarily used for keyboard and mouse communications on newer ATX style PCs. Although this is usually specifically for the keyboard and mouse, many integrated Smart Card access devices that are embedded in keyboards or mice can utilise the available PS/2 connection. A good example is the Cherry G83-6700 SmartBoard (Cherry, 2003a), a keyboard with an embedded Smart Card access device. Both devices operate over a common PS/2 connection.

Universal Serial Bus (USB)

This interface was designed by Intel to become the new serial device standard to avoid using serial or parallel ports. This new interface standard enables easier connection of devices via daisy chainning and can provide power to several devices through the USB connection itself. Several Smart Card access devices are USB based for the user’s convenience. As an example, Cherry provides a standalone access device called the Smartcard Reader USB ST-1000 U (Cherry, 2003b), however there are several other USB Smart Card access devices available.

PCMCIA

PCMCIA is used primarily in laptops as the slimmest form factor devices available. Several Smart Card access devices connect via PCMCIA and use a slot in mechanism for the Smart Card itself. This is of benefit to users of Smart Cards who wish to avoid bulky external mechanisms. SafeNet is one such company that sells PCMCIA Smart Card access hardware (SafeNet, 2003).
For simplicity, it is recommended that the proposed system utilise a serial or USB based Smart Card access device, as these interfaces are the most widely supported and easiest to utilise by both the users and developers of the proposed system.

### 5.2.3 Smart Cards

Due to the high number of different application specific Smart Cards with varying capabilities, any implementation of the proposed system must seriously consider what Smart Card to use in order to be successful.

Naturally, unsophisticated memory cards are fairly inexpensive, while more sophisticated Smart Cards with protected memory segmentation are fairly expensive and must be investigated appropriately. In particular, implementers of the proposed system must see how much it would cost to distribute the Smart Card to all possible users. In addition, questions arise such as whether the card should be contact based or contactless - a difficult issue if the card is to be used in a high use area, such as security access where access devices can malfunction through excessive use.

These questions can only be answered after a thorough investigation and plan of how the proposed system will operate and what applications it will support. A simple ID card may suffice with a simple write-once memory card and utilise server connectivity to link the user of the ID card to any number of facilities. More complex cards may require segmented memory sections and separate codes to access each - these would be determined by what applications the institution requires.

Currently, the proposed system primarily concentrates on utilising the Smart Card as an identification device only. However, institutions should feel free to investigate which Smart Card is best from them to support possible current and future applications.

A recommended card that is capable of handling most applications is the GemClub Memo Smart Card (GemPlus, 2003c). This Smart Card supports numerous applications and has many features that are designed to assist any institution to successfully implement the proposed system.
5.2.4 APIs, Smart Card Protocols and Software

To support reading and writing to Smart Cards, the access hardware device must support the Smart Card’s native protocol for communications. The most common protocols are defined in the ISO/IEC 7816-3 standard and are called T=0 and T=1. There are also some less widely used T series protocols available but it is recommended to utilise only the T=0 or T=1 protocols for maximum compatibility.

T=0 was the first widely used protocol, in particular on mobile phone SIM/GSM cards and is a byte or character based protocol. T=1 is another widely used protocol however it is used for block based data transfers only. There are other manufacturer based custom protocols that exist however they are considerably less commonplace than T=0 or T=1. (Wolfgang & Wolfgang, 1997)

Because of the proliferation of different Smart Card hardware and protocols, the PC/Sleep Card (PC/SC) Workgroup (pcscworkgroup.com, 2003a) was formed to create a new standard of interoperability between Smart Card hardware and PCs. They released their first PC/SC Specification (pcscworkgroup.com, 1997) in December 1997. The most common operating systems (OS) such as Microsoft Windows and Apple MacOS X support Smart Card hardware operating to the PC/SC specification.

For example, Microsoft provides suitable Application Programmer’s Interfaces (APIs) based on the PC/SC specification as part of every Windows release since Windows 95 (microsoft.com, 1999). By programming to these APIs it is possible to operate with a wide range of Smart Card access devices without necessarily being knowledgeable of the low level implementation details.

Apple also provides Smart Card API’s that are based on the PC/SC specification and are currently heavily tied into the security aspects (apple.com, 2003b) of MacOS X. For other operating systems such as GNU/Linux it is recommended that you obtain a Smart Card developer’s kit and supported hardware that can be programmed specifically with or without the use of the PC/SC specification or API’s.

Smart Card manufacturers also design their access hardware to be compliant to the PC/SC specification. A listing of all the compatible hardware is available from
the PC/SC Workgroup's web site (pcscworkgroup.com, 2003b) and it is recommended to consider only PC/SC compatible Smart Card access hardware.

When deciding on a particular Smart Card access device and Smart Card for use in the proposed system, it is recommended that only PC/SC devices and Smart Cards that support the T=0 or T=1 protocols be used for maximum flexibility and compatibility.

5.2.5 Summary

From these details it is possible to make informed decisions about what are the capabilities of various Smart Card access devices, as well as determining to a degree what Smart Card to choose. Specifically, by choosing only Smart Card hardware that operates to the PC/SC Specification and utilises the T=0 or T=1 protocols, it is guaranteed to have less implementation issues over non-compliant Smart Card access hardware.

5.3 The Client Interface

5.3.1 Introduction

To utilise Smart Cards successfully, it is proposed to have a common front end client interface for staff and students of the institution. This is to enable staff and students to interact with various institution based applications. However, this section is only for advanced levels of interaction, as certain systems such as simple photocopying or vending machines require very little user interaction, whereas administration, printing, checking card balances and other advanced applications would benefit greatly from a more advanced system as detailed below.

5.3.2 Requirements

With any system, it is important to have a common front end client interface that the user can interact with to perform tasks. However, for the University Wide Smart Card system there are particular requirements that can be identified easily:

1. Must be of low cost, preferably free;
2. Must be preferably identical across different operating systems;
3. Must attempt to avoid installations of software on client computers;
4. Must be accessible in the institution or remotely;
5. Must be able to access Smart Cards if available.

These requirements reduce considerably the range of products that could assist in building the system. In general this not a problem as the developers of the proposed system will be able to generate a client interface for minimal cost.

5.3.3 Options for Client Front Ends

There are two options for the client front end system: programs that execute on the client side and communicate to a server, known as client-server applications and; programs that only provide an interface to a server which then does all the processing, these are known as thin client application systems.

Examples of client side programs that communicate to a server include HTML web browsers, various network aware programs such as Oracle (oracle.com, 2003), FileMaker Pro (filemaker.com, 2003), or custom written programs in Java (sun.com, 2003c), Visual Basic (microsoft.com, 2003d) or other programming languages. The main distinction is that client programs are installed onto the client and does most of the processing, while the server is used to access and store data only.

Server side programs that support thin clients operate by installing a screen sharing program on the client to connect to a remote server that has the necessary software installed. Many programs exist to do this such as VNC (att.com, 2003), pcAnywhere (symantec.com, 2003), Timbuktu Pro (netopia.com, 2003), X Windows (xfree86.org, 2003) and Citrix (citrix.com, 2003). The primary difference to client-server applications is that the client must be pre-installed and only handles user interaction and screen sharing, while all processing is handled on the server.

5.3.4 The Client Front End Proposal – Utilise HTML Web Browsers

The advent of the Internet and the World Wide Web (WWW) however provides a common interface for users to access the system via web browsers such as
Internet Explorer (microsoft.com, 2003b), Netscape (netscape.com, 2003c) and Mozilla (mozilla.org, 2003).

As all popular operating systems provide Internet access capabilities now in addition to a web browser, it is proposed that the best way to provide a nearly identical front end for the users is to present HTML (Hypertext Markup Language) for clients to view and access as the primary interface.

5.3.5 Why HTML Web Browsers?

Using HTML browsers, it is possible to remove most of the restrictions that would prevent the proposed system from being adopted. Other solutions require installation of server and client software, or were too processor intensive for either the client or the server and failed to scale appropriately, or were very costly to support numerous users.

The most common browsers are currently free for downloading or are bundled with the OS already. Since the advent of the Internet, many people are already accustomed to using HTML browsers to access files, and many would have the client web browser software already installed.

In addition, the rapid adoption of online commerce through the use of interactive web sites such as Amazon.com (amazon.com, 2003) and many similar web commerce sites provides a useful model on which to base the proposed system. These existing web sites provide a similar environment for clients of the proposed system to adopt.

5.3.6 Interactivity with HTML and Web Sites

As web sites typically serve static web pages, many different techniques have been developed in order to provide dynamic user specific web pages to the client. Some of the different client based techniques are listed below.

Attempts at providing interactivity through web pages include JavaScript, Java and Flash. However, differences in how each OS and browser treats these competing efforts mean they are not viable as an essential service. Below is a list of various problems involved with each form of interactivity with HTML browsers.
**Netscape JavaScript** (netscape.com, 2003a)

Not all operating systems and browsers support JavaScript or equally well. Differences exist between versions of browsers that can cause unnecessary problems. JavaScript is viable as a means of sanity checking user inputs and providing some level of user interaction that is client based.

**Sun Java** (sun.com, 2003c)

Most operating systems support Java 1.1 applets, however the more sophisticated Java 1.2 applets cannot be used effectively due to the majority of operating system developers being unable to support a common standard for Java. Sun however has released a Java Runtime Environment (JRE) which enables support of Java greater than 1.1, but requires a lengthy download and installation before it can be utilised. In addition to this, HTML pages must be run through a converter before the newer Java applets can be utilised.

**Macromedia Flash** (macromedia.com, 2003)

Macromedia Flash is a proprietary file format that downloads and executes through the Macromedia Flash plug in. It is not available on all platforms, and requires the user to download the plug in to access the Flash media. It also requires licensing fees and costly development software in order to generate Flash based sites.

Fortunately, the Hypertext Transfer Protocol (HTTP) includes methods for simple interaction with the user via HTML forms that all browsers support consistently. Many of the most popular interactive web sites utilize complex HTML with forms to provide their user interface, which can be considered to say the technique is workable and relatively easy to make cross platform and cross browser compatible.

It is recommended that developers of the proposed system utilise only HTML forms to generate the client front end interface to reduce difficulties from incompatible operating systems and browsers. It is recommended that JavaScript be investigated to provide some level of immediate interaction with the user to provide error checking for the HTML forms in addition to simple dialogs to warn the user of possible errors.
5.3.7 Why not Java and Java Applets?

There is a strong feeling among the general Internet community that Java is the logical choice for interactive web sessions, and in a sense many developers can agree. Unfortunately, there are currently three main competing Java implementations: Microsoft Java, Netscape Java and Sun Java. It is difficult to determine exactly which one of these or which version of Java is going to run on any given client computer.

Currently, the HTML standard supports Java using the <APPLET> tag. However, due to HTML processing in Internet Explorer and Netscape, this will always invoke the older Java 1.1 implementation rather than the later implementations.

In order to use later versions of Java, a programmer must run a HTML converter on their HTML source code to convert Java <APPLET> tags to embedded object tags in order to override the in-built Java implementation.

Furthermore, a user must have a newer Java implementation installed in order for the Java applets to execute. Unfortunately, the default is still firmly wedged at Java 1.1. To enable later Java support, a client must typically download the Java Runtime Environment (JRE), which can take considerable time to accomplish.

Essentially, the downside to this is that most of the tools to create sophisticated Java sites are held back by the requirement that you require a version of Java greater than 1.1. It is very difficult to determine how many clients would have the newer version of Java installed, so it is better off being avoided.

5.3.8 Security and Access to Smart Cards

Unfortunately, the use of HTML and HTML forms does not enable complex interaction with client side devices, in particular a Smart Card access device. To access Smart Cards, more complex software is required and a corresponding higher level of access is needed.

In order to access the Smart Card, a client program would need to be executing on the user's computer that could be contacted by the server whenever it was required to access the Smart Card.
This sort of interaction is acceptable for the proposed system as this would be limited to a small number of computers and would be generally not available on most student or staff computers unless expressly installed.

For convenience, it is proposed to utilise Java to code the client program. This would be an application that would start up with the computer and solely handle requests from the server. Alternatively, a custom written program could also be written in C, C++ or Visual Basic.

In general, the client program would utilise the operating system API’s for access to the Smart Card hardware, preferably conforming to the PC/SC specification. For Java, this would require Java’s own API’s or the Java Native Method Invocation (JNMI) system for access.

Alternatively, depending on the type of Smart Card access device, it is also possible to code Java to access the device directly if it is a serial device using the JComm Java libraries, or other API’s if available with different programming languages such as C/C++.

Security for the client program is directly dependant on the client program. It is possible to integrate a variety of authentication mechanisms, verification of the server as well as encryption to secure data being read or written by the Smart Card, utilising Secure Sockets Layer (SSL) and OpenSSL (openssl.org, 2003). This method of providing encryption over the network is highly recommended.

5.3.9 The Client User Experience

Essentially, the final user experience of the system should be closely modelled to popular electronic commerce web sites. These web sites enable the user to log in, check the status of various topics, and in general allow the user to browse around and seek information.

With an attached Smart Card reader, it is then possible to enable further enhancements and perform functions not available to the user without a Smart Card by using the card to update details or provide further authentication.

This is an important distinction, as it is expected that many users would be accessing the site without a Smart Card initially except from specialised client computers. Therefore, it is important to provide limited functionality to the initial
users of the system with full functionality provided to those users with access to the appropriate hardware.

5.3.10 Summary

To promote the proposed University wide Smart Card system and encourage rapid adoption, it is recommended that the use of HTML web browsers will considerably help the users of the system by providing a free client interface that they are competent and familiar with. With the addition of supported Smart Card access hardware, it is then possible to provide full access to the proposed system for staff and students as required with additional installable client software.

5.4 Server Front End

5.4.1 Introduction

Because the client front end of the proposed system is web based, the server front end is a HTML web server that is capable of handling the client requests for web pages. The server front end will also be able to communicate to the client’s computer in order to access any Smart Card hardware if available.

However, current web servers merely provide data from programs or files and return it to the user. In order to provide dynamic web pages as a unique session per user, additional software is required to modify or create HTML dynamically according to what the user is currently doing. This will be explained in detail in the following section.

5.4.2 The Proposed HTML Web Server and Operating System

In order to choose a suitable web server, flexibility, cost and scalability were considered. The primary suitable platform was GNU/Linux (kernel.org, 2003) along with the superb Apache (apache.org, 2003c) web server, however other Unix operating systems such as FreeBSD could also be used.

Apache is a free web server for Unix derivatives that powers many well known web sites. Apache supports many different file formats and handles third
party modules admirably, enabling dynamic HTML to be generated by almost any program that can output HTML. The other major web server that could be considered for use is Microsoft’s Internet Information Services (IIS) (microsoft.com, 2003c).

Consideration was given to Microsoft’s IIS, however the high cost of implementation that requires the purchase of Microsoft Windows Server 2003 (microsoft.com, 2003f), the lack of extra programming facilities and inability to scale cost-wise was not adequate to make it the primary choice.

5.4.3 Why Apache?

Apache was clearly the best choice for the proposed University Wide Smart Card system. A variety of reasons are explained below:

1. **Cost** - Apache currently costs nothing, which enables and encourages scalability. If using GNU/Linux, the Operating System also costs nothing.

2. **Flexibility** - Since Apache is open source, there is technically nothing you cannot do with Apache. Closed source web servers do not provide the same flexibility. In addition, Apache supports modules and external programs, which cannot be done on other OS’s as easily.

3. **Scripting** - Apache in conjunction with GNU/Linux has exceptionally powerful scripting and programming capabilities, something that GUI OS’s traditionally cannot compete with.

   In addition to these reasons, Apache currently is the most widely used web server on the Internet today, with over 60% of the Internet web servers being run by Apache according to Netcraft (netcraft.com, 2003a) through their regular web surveys (netcraft.com, 2003b) as of June 2003.

5.4.4 Secure HTTP with SSL

In addition to supporting modules, Apache also supports the Secure HTTP protocol. This enables HTML and other information to be transferred securely from the client to the server and vice versa over a SSL (Secure Sockets Layer) connection using OpenSSL (openssl.org, 2003). This has been used consistently among the most
popular web sites to protect user information such as credit card numbers, addresses and passwords.

SSL has become the standard for providing secure communications due to the widespread acceptance of SSL for secure HTTP sessions. The actual installation and configuration of SSL is relatively simple and requires very little modification to the Apache web server configuration. Clients fortunately support SSL by default in the most common browsers and require no configurational changes in order to support secure web communications.

For Secure HTTP to be supported correctly, the server must have a valid certificate from an authorised Certifying Authority (CA) such as VeriSign (verisign.com, 2003) that must be submitted and paid for. Details for handling this are documented with the OpenSSL software, and details for integrating OpenSSL with Apache are documented on the Apache web site (apache.org, 2003a).

5.4.5 Client Interaction with Session Based Handling

For Apache to interact uniquely with connecting clients, it must serve client specific pages of HTML, that is, pages of HTML that utilise state information from the client to custom generate web pages. Typically, the HTML is stored on the server and then copied via the HTTP protocol to the client, which then displays the HTML. This sort of interaction is strictly passive.

However, it is possible to use the modular architecture of Apache to provide true interactive sessions using various different techniques and software. In order to handle a unique session for each client with HTML pages, it is required to have some form of state information included with the HTML requests. Several different techniques were developed such as the Common Gateway Interface (CGI), cookies, appended URLs and hidden HTML forms. How each of these operates will be described further down.

To provide interaction with the various modules, the client must specifically access a web page that invokes the module. The various means of doing this are:
HTML Forms Processing

The user typically fills in a form and then specifically sends the information to the specified module via the HTTP POST command. The module can then process it via information received. This is the primary mechanism that users interact with to update the session details, but does not handle how the HTML pages appear.

Special Uniform Resource Locators (URLs)

The user typically accesses a specialised URL that the server recognizes and then calls the module responsible for handling that section of the URL. This can be something like:


This executes the CGI program ‘mycgi’ with parameters of ‘index.html’. Various interactions can then append data to the URL which passes parameters to the CGI for further processing, such as:


The addition of parameters on the URL such as the login=true enables information to be passed and processed to the CGI program. However, this means of data transfer is not particularly secure.

Hidden HTML Forms

Since specialised URLs are not secure, some web sites attach extra information into hidden HTML forms in order to append state to the web page by modifying the HTML sent to the user with extra hidden information for this purpose. A hidden HTML form is simply a standard HTML form with no visible elements except for the submission button.

Cookies

Other web developers saw that most of the other solutions weren’t ideal, so they implemented the concept of cookies. Cookies are small (4K or smaller) packets of information that can be read and written to by the server via the browser.
When a browser sends a request for a particular HTML page, it can be requested by the server that it requires a number of cookies be transferred from the client in order to process the page correctly. The server can then read and update the cookie’s information to reflect changes in the state of the session with the user.

Pre-processors

As the Apache web server is capable of processing files requested from a URL, usually a HTML page, it is possible to embed commands directly into the HTML page and then have a program called a pre-processor read and execute these commands.

Out of these different techniques, pre-processing with HTML forms has proven the most popular so far, with special URLs and hidden HTML forms proving the less popular due to the visibility to the user and the potential for abuse. Pre-processing with only HTML forms is more secure and prevents unauthorised access.

5.4.6 PHP for Session Management and Dynamic HTML Web Pages

After investigating several different systems for handling server side dynamic HTML, including CGI (apache.org, 2003b), Java servlets (sun.com, 2003a), Java Server Pages (JSP) (sun.com, 2003b), Active Server Pages.NET (ASP.NET) (asp.net, 2003), Perl (perl.org, 2003) and PHP (php.net, 2003), it is proposed to utilise PHP: Hypertext Pre-processor (PHP). PHP is a relatively new scripting language, however it is specifically designed to handle dynamic HTML web pages as well as HTTP sessions. It is fast and easy to program in, and has excellent documentation and a friendly user base.

ASP.NET is Microsoft’s latest offering for web developers to generate dynamic HTML web pages from languages that support Microsoft’s .NET development environments. At the moment, ASP.NET operates primarily under Microsoft IIS and requires Windows 2000 or Windows 2003 Server. To develop for ASP.NET, it is recommended to use Visual Studio .NET (microsoft.com, 2003e) although there are several other .NET compatible development tools available.

The high cost of Microsoft Windows Server and developer tools such as Visual Studio .NET mean that although interesting, ASP.NET is not able to scale as cost effectively as other software solutions. Fortunately, the Mono project
(ximian.com, 2003) attempts to provide a free implementation of the .NET development platform and links into the Apache web server with the mod_mono module (go-mono.org, 2003).

This open source solution for the .NET development platform will prove interesting in the future once it is proven stable by providing a low cost, scalable solution which will work well with the advanced development tools from Microsoft provided by Visual Studio .NET or from the Mono project.

Other solutions such as CGI, Java servlets and Perl are adequate systems in their own right, they are not particularly programmer or web designer friendly. All of them require a programmer to endure a lengthy edit/compile/install cycle for each modification made to the system, and the use of special URLs to activate them was prone to abuse from insightful users. In addition to this, they also must process all data for the HTML page, meaning that a simple web page would require considerable amounts of code to copy blocks of HTML to and from existing files.

However, PHP avoids this by embedding PHP scripts directly into the HTML page itself. This considerably shortens development time by enabling a web designer or programmer to design the HTML first, and then add in short PHP scripts to handle the dynamic portions of the web site. In addition to this, PHP does not require any special URLs in order to operate. It is possible to simply rename HTML files from the .html suffix to PHP with a .php suffix to enable PHP scripting, or have all HTML files pre-processed by PHP with trivial configuration changes to the Apache web server.

PHP also supports programmer friendly sessions. As there are two different ways to support user sessions, namely cookies and appended URLs (where the session parameters are stored in the URL) this could prove difficult for programmers to support both. Fortunately, PHP supports both techniques through a single API, making support for sessions transparent and considerably easier to implement.

After exploring the various options available, it was decided that PHP would be the ideal solution to providing a dynamic HTML front end to the client. PHP handles all the required software for the server back end interface with an extensively supported API.
5.4.7 Access to Smart Card Access Hardware and Smart Cards

The system detailed above can read and write to the client’s Smart Card by utilising PHP to communicate to the client’s Smart Card access program. It is possible to add in other features to the access program as required such as performing calculations on the card, including advanced encryption and other such facilities.

As an example of how this could operate, it is convenient to use PHP pre-processing to include checking the client for the existence of a Smart Card access program. Such a system could operate as follows:

1. Client accesses the login web page;
2. The login web page has a PHP script that communicates to the client’s computer to determine if there is a Smart Card access program that it is able to communicate with successfully;
3. Depending on the client’s preferences, this may automatically redirect and log the client in automatically (single sign on), or require extra authentication to log in, such as a password or PIN number;
4. End result is that the client is logged in via the Smart Card;
5. However, if the Smart Card is unavailable, PHP can provide an alternative HTML form to enable the client to log in normally via username and password.

This method provides a transparent login process for clients if they so request. Additionally, the process can be extended to include other forms of logging in, such as biometric, video or voiceprint login authentication mechanisms.

As explained in the Client Front End section, it is possible to extend the client and server Smart Card access programs to provide encrypted secure communications via SSL that can ensure that Smart Card information is secured from tampering and is guaranteed to go to the server only.
5.4.8 Summary

As described, the server front end is essentially the Apache web server with PHP. This is fully capable of handling users and session based connections as well as interacting with the server back end. It also provides connectivity between the server and the client for access to Smart Cards, and includes all the prerequisite API's for communicating to the server back end.

5.5 Server Back End

5.5.1 Introduction

In the proposed system, the server back end is the most important. In order to successfully integrate with the institution’s existing systems, it is required to handle numerous different data sources and be able to connect each one under a common communications protocol. This feature of the proposed system is not to be taken lightly, as it provides a clear mechanism and path for adapting existing systems into the new proposed system.

5.5.2 Requirements

The requirements of the server back end are to provide a common interface to dissimilar data sources in addition to providing a common communications protocol. This is required to join the many and varied systems already in use at the institution together to form a seamless whole. In addition to these requirements, the server back end must also provide the data storage mechanisms used to store required data for the proposed system.

5.5.3 Integrating with Existing Institution Systems

It is anticipated that any new system proposed would have to deal with numerous existing dissimilar data sources that are currently in use at the institution. For instance, it is not unusual to find that there are separate servers to provide information on security access, student marks, student details and library borrowing...
facilities. These are known as application specific databases, and generally it is difficult to import or export data to and from each database.

Each of these data sources if not managed correctly will force unnecessary duplication of data and delays in getting up to date data from one location to another. For example, if student enrolment information is not up to date and easily accessible, delays in security access or computer account generation may result as it is difficult or impossible to confirm student details accurately.

To resolve this difficulty, what is required is a common protocol that is able to adapt to each of these dissimilar data sources. This protocol will be able to connect each data source to another with a minimum of trouble so that data can be distributed considerably more efficiently than before.

5.5.4 The Solution – Directory Services

As the number of application specific databases grew, a number of companies realised that this was becoming a difficult problem. Their systems administrators were finding it difficult to keep all the individual databases up to date with what was going on in the company. In particular, many companies have a telephone directory that is essential that it is up to date; however they were finding it difficult to do so when a simple change of location of an employee could result in updating numerous different databases.

Based on the telephone directory, several companies researched into providing comprehensive directory services, where all information in a company could be stored in a directory based structure and would then become the centralised means of obtaining key information.

Several successful attempts at hierarchically organized data in the field of information management were first attempted by the major operating systems to handle the vast amounts of information that were required to properly manage the system, or grew out of efforts to provide general purpose directory information systems. Some of the better known types of directory services are described:
Novell eDirectory (formerly Novell Directory Services)

Novell’s eDirectory (novell.com, 2003a) server is the new name of their older product Novell Directory Services (NDS). NDS was heavily used in Novell’s Network Operating System (NOS) Netware (novell.com, 2003b), and was pioneered by Novell to help organize and control the vast amount of administrational information required by Netware.

Microsoft Windows Active Directory

Active Directory (microsoft.com, 2003a) is Microsoft’s competing directory product to Novell’s eDirectory and NDS, and was released with Windows 2000 Server and is included in Windows 2003 Server. At present, Active Directory usually handles Windows domains and enables easy administration of multiple Windows based computers.

Open Directory and NetInfo

NetInfo (apple.com, 2001) was one of the earlier directory based systems information databases and was developed by NeXT Computer Inc. It contained information about various aspects of the NeXTStep and OpenStep operating systems such as user and group information. At the time (and currently) it simplifies and removes the heavy Unix dependencies on loosely formatted text files for configuration of the OS.

Today, NetInfo remains in use today as NeXT was purchased by Apple and has become the main information directory system of MacOS X. It has now been improved and is now part of Open Directory (apple.com, 2003a).

X.500

This was the first standards based directory protocol, developed by the International Telecommunications Union (ITU) and the International Organization for Standardization (ISO). X.500 was based on the Open Systems Interconnect (OSI) network protocols. Eventually the protocol was simplified via gateway servers to work over TCP/IP, which then became LDAP, and eventually X.500 was made redundant in favour of LDAP only servers.
Lightweight Directory Access Protocol (LDAP) and OpenLDAP

LDAP grew out of programmer's attempts to simplify access to existing X.500 directories. Eventually, due to the popularity of LDAP, it was decided to use only LDAP servers rather than X.500 servers with an LDAP gateway or front end. OpenLDAP (openldap.org, 2003) is the current Open Source implementation of the original University of Michigan's standalone LDAP server. OpenLDAP will be described further in this thesis.

5.5.5 Lightweight Directory Access Protocol (LDAP)

Fortunately, a suitable protocol already exists known as the Light Weight Directory Access Protocol or LDAP. LDAP is a non-proprietary extensible directory based protocol that can be highly effective in most applications if used correctly. It also has recently obtained considerable support from commercial companies and is supported as the de facto standard of protocol communication in directory based applications and servers.

LDAP itself was specified in the Request for Comments (RFC) 1487 (Yeong, Howes, & Kille, 1993) in July 1993 and was a simplified version of the X.500 protocol. The main differences between X.500 and LDAP are that it removes redundant functionality and uses TCP/IP as a communications protocol instead of the much more difficult to implement OSI model. LDAP version 2 was specified in RFC 1777 (Yeong, Howes, & Kille, 1995) and was the first popular release of the LDAP protocol.

After a review in 1995 it was discovered that X.500 was almost redundant as programmers preferred to use the LDAP protocol instead. Programmers then built an LDAP only server called slapd at the University of Michigan in December 1995 and has been a cornerstone of the LDAP community ever since, eventually becoming the industry standard implementation called OpenLDAP (openldap.org, 2003).

5.5.6 LDAP Advantages

With LDAP, it is possible to organize data into coherent directories that can separate data into hierarchically organized sections compared to a flat database system or multiple separate databases. A short example would be in an existing
University system, several departments may have student accounts for computer access based off a central Student Records system, as shown:

![Diagram showing Student Records with Arts, Engineering, and Mathematics Accounts]

Denotes a weakly defined or ambiguous protocol used to communicate, usually one way only

**Figure 3 A sample University system setup**

In this particular setup, each of the separate organisations of the University would be forced to generate their own accounts and store student details based off the master Student Records. Depending on complexity, each organisation may have accounts for various systems such as GNU/Linux, Windows, MacOS, WebCT, security access and more.

Naturally, keeping information up to date becomes a problem depending on how the institution is structured and how frequently data is kept synchronised with other sources, in particular, the master Student Record sources. In general, data flow is one way only and is of a protocol that is weakly defined or ambiguous at best.

Questions arise such as to what form of data is passed from server to server? How much access does another server get, and who gets access? These questions can be resolved by using LDAP to control access and provide a common protocol for communicating essential data from location to location.

Under LDAP, it is possible to hierarchically organise data and servers into a much more useful system:
Figure 4 A LDAP Server Setup

With LDAP, it is now possible to organise servers hierarchically so that there is a strong protocol connection between different servers. Using the LDAP protocol, there is a clear means of communicating between any of the divisions of the University and the Student Records system, and there are clear means of determining details from each of the divisions of the University.

Using LDAP’s referral mechanism it is possible for a query to one server to automatically initiate queries onto the other servers, or provide details to the LDAP clients about the referral for them to query themselves.

For further details on LDAP and namespace design, it is highly recommended the implementers of the proposed system thoroughly read and understand “Understanding and Deploying LDAP Directory Services” (Howes et al., 1999). It is important that the LDAP namespace is properly designed from the start to avoid problems later on.
5.5.7 LDAP and PHP

In order to communicate between the web server front end and LDAP, programmers are required to use the LDAP protocol from PHP based off user web page requests. Fortunately, PHP has an entire API that enables it to connect and communicate to LDAP servers which is well supported and documented.

Using PHP, it is a relatively simple task to open a connection to the LDAP server and communicate information to and from the LDAP server, process it and then display results to the user using PHP to generate dynamic HTML web pages.

Utilising this technique it is possible to generate dynamic HTML web pages for each user logging into the proposed system by using PHP to first authenticate the user (either via a Smart Card or other authentication mechanism) and then use queries to the LDAP servers to retrieve the required user’s information as required.

5.5.8 Open Source LDAP – OpenLDAP

To use LDAP as the back end of the proposed system it is necessary to have an LDAP server capable of communicating with PHP (or Java) in order to service user requests. It is possible to use directory servers such as Novell’s eDirectory or Windows Active Directory, but in order to keep costs low, it is recommended for the proposed system to use OpenLDAP as the primary back end system.

OpenLDAP is a free Open Source implementation of the LDAP protocol and is based off the original University of Michigan implementation of a standalone LDAP server and strongly resembles the original distribution.

OpenLDAP has two main parts. One part is the front end LDAP protocol handler, which handles LDAP requests and queries. The other part is a back end gateway information interface, which enables OpenLDAP to communicate to a wide variety of data sources, such as Structured Query Language (SQL) databases, simple indexed databases as well as plain text files, shell scripts and more which will be described.
5.5.9 LDAP Data Sources

For LDAP to operate it requires some form of data source to access and store data. Using slapd, the standalone LDAP server from OpenLDAP, it is possible to utilize slapd’s in-built gateway interfaces to connect to simple indexed databases such as Berkeley DB (BDB) (sleepycat.com, 2003), GNU DB (GDBM) (gnu.org, 2003), or SQL databases such as PostgreSQL (postgresql.org, 2003) and MySQL (mysql.com, 2003). Last but not least, gateway interfaces can be written using scripting languages such as PHP, Perl and Unix shell scripts, or programming languages such as C/C++ and Java in order to connect LDAP to entirely new forms of data altogether.

BDB, GDBM, MySQL and Postgres

The most popular forms of databases used in OpenLDAP are simple hashed index style databases, the one recommended for use in OpenLDAP is the Berkeley DB, or BDB, however it is possible to use GNU’s DBM, or GDBM. OpenLDAP by default will use BDB as a simple indexing database and store the attributes into it.

However, indexed databases are not optimised for frequent writing, therefore the option to use an SQL database as an alternative is highly recommended for larger installations that would support frequent updates sufficiently.

PostgreSQL is one of the most popular free SQL databases available. Its advanced capabilities over GDBM include sophisticated caching, rollback and commit facilities as well as rapid relational table linking when it is clear that GDBM is not sufficient. In addition to PostgreSQL, MySQL can also be used as an excellent database server in conjunction with OpenLDAP and provides similar functionality.

LDAP

OpenLDAP also supports the ability to connect to other LDAP servers. This enables multiple LDAP servers to be connected together and encourages scalability and distributed data to be utilised. In particular, OpenLDAP can be connected to Novell’s eDirectory or Microsoft’s Active Directory in order to obtain critical user information and details.
Scripting

OpenLDAP also supports a scriptable backend for connecting to the many programming environments or shells under Unix for processing data. In addition, it also supports scripting systems such as Perl and PHP that many Unix users would use for general text processing.

Custom Written HTML, XML or Text Data Links

If the above solutions are insufficient, it is possible to always code a custom backend to directly link into OpenLDAP. This enables programmers to access data over alternative forms, such as HTML, XML or simple data connections and files either by coding your own backend or using the scriptable backend to provide data as required to OpenLDAP. These are known as information gateways.

With custom written gateway information interfaces, it is possible for programmers to adapt the existing institution systems into the new proposed system without having to discard the old system or isolate it. This is an important feature of LDAP as it enables a gradual adoption of LDAP throughout the institution without requiring sweeping changes that usually cause many problems initially.

As the gateway interfaces become more advanced, it is possible to support writing and updating of the data as required. The ideal end result is that each of the existing systems at the institution is supported by a gateway interface to LDAP or supports LDAP directly.

5.5.10 LDAP Schemas

One of the differences between LDAP and traditional databases is that LDAP directories can be unconstrained. Therefore almost any directory can have any number of additional attributes. To constrain the LDAP database, developers use schemas, these are the equivalent of table descriptions in SQL. Schemas are essentially rules about what sort of information can be stored in the LDAP directory.

The use of schemas enables mapping to more efficient indexed and relational databases possible from within LDAP, and is in general good practice to reduce the amount of information that could be inadvertently added on over time.

Fortunately, the OpenLDAP implementation of LDAP provides numerous schemas for use which can be adapted and implemented into the new proposed
system, most of which are proposed and standardised on by the main LDAP protocol developers.

Developers of the proposed system can use and adapt the predefined schemas, or it is possible to develop new schemas once a suitable Object Identifier (OID) number has been obtained from the Internet Assigned Numbers Authority (IANA) (iana.org, 2003), also known as an Enterprise number and is closely tied with the Simple Network Management Protocol (SNMP).

Once an OID number range has been obtained, it is possible for the developers of the proposed system to allocate OID numbers for the schemas they develop for the system without fear of a conflict with other users of schemas in LDAP around the world.

5.5.11 LDAP Namespace Design

One of the most important parts of working with LDAP is the design of the LDAP namespace. The LDAP namespace determines directly how the LDAP servers are connected, how data is accessed as well as how security is implemented. Therefore, developers of the proposed system should consider carefully how best to implement the system’s LDAP namespace in order to reduce difficulties later on from poor design or forced redesign. The best book to refer to considering these design issues is “Understanding and Deploying LDAP Directory Services” by T. Howes (Howes et al., 1999).

5.5.12 LDAP Security

LDAP security was formerly one of the more weakly defined areas of the LDAP protocol. Since LDAP was designed originally without much security in mind, companies decided to strengthen the security in the latest version of LDAP, or LDAP version 3. Several solutions exist to give LDAP security features that are becoming more common as the LDAP v3 protocol becomes more widely used.

There are currently five forms of security with LDAP databases, these are:

1. Access Control Lists;
2. User Based Password Security;
3. Predefined Server Security;
4. TCP/IP Security and Firewalling;
5. Secure Sockets Layer (SSL).

Access Control Lists (ACLs)

Access Control Lists (ACLs) are prominently utilised in Netscape’s LDAP Directory Server. This embeds access control information for each attribute and directory into the LDAP database itself, with strict controls over who can edit the fields. This form of security is extremely flexible and quite intuitive to use.

User Based Password Security

Most LDAP implementations support a binding security, that is, a password is required in order to connect to the LDAP database to read or write information. Without a password, it is impossible to connect. OpenLDAP supports this form of security.

Predefined Server Security

This works in conjunction with the user binding security. It defines security access levels according to the user connected about which directories and attributes can be read, compared or written. This is the most popular form of security for LDAP databases, and is well supported by OpenLDAP.

TCP/IP Security and Firewalling

Security via TCP/IP can be accomplished using several different techniques. The most common security mechanism is tcpwrappers, which intercept all accesses to ports and optionally allows or denies access to services such as telnet and ftp, and in particular, OpenLDAP.

In addition to this, the actual plug-ins for the web server may optionally deny access in a similar fashion to tcpwrappers, or the developers may use other forms of security access built in to the proposed system to restrict access. Lastly, TCP/IP firewalls may be put in place to control the flow of data to the server transparently, ensuring yet more security.
Secure Sockets Layer (SSL)

Security with SSL is easily accomplished with the installation of OpenSSL. SSL is the de facto standard for communications for secure HTTP. A simple plug-in addition to Apache enables SSL to operate with HTTP to communicate the https protocol. In addition to just securing HTTP, it is possible to use SSL and OpenSSL to secure other forms of communications, such as the LDAP protocol, forcing all LDAP communications to be handled over a secure connection only.

Security these days is still a difficult issue to handle, it is recommended that developers of the proposed system carefully consider the overall security of the system and decide on this from a data access point of view.

5.5.13 Summary

The advent of directory protocols and services such as LDAP has opened a new direction for data handling and sharing between different applications and systems. In the proposed University Wide Smart Card System, LDAP provides a key mechanism to enable backwards compatibility with the existing institution’s data systems while also providing a means of implementing future applications that can fit into the LDAP directory system.

5.6 Scalability and Maintenance

5.6.1 Introduction

This section details various issues that are expected during the development of the proposed system. The most important parts of the system that should be handled correctly are the ability of the proposed system to scale to support numerous users. In addition to this, the proposed system should also be easily able to synchronise to other servers as well as provide a suitable backup mechanism.

5.6.2 Scalability

Scalability is an important issue for any major network service. Too few servers can frustrate the systems users as the servers are slow to return required information or worse still, fail to respond at all. It is therefore important that during
the development and maintenance of the system that a close eye is kept on resources being used and to make sure that they are adequate. This tries to minimize problems by taking advantage of multiple servers and scalability.

Fortunately, LDAP and OpenLDAP have in-built mechanisms for handling scalability. Firstly, as the directory namespace is hierarchically organized, it is possible to separate parts of the directory automatically to different servers by utilising LDAP references or chaining techniques. These techniques allow different parts of the directory namespace to automatically be split over multiple servers, allowing the load to be spread over several servers.

Secondly the nature of the Internet allows the resolution of names to automatically be resolved to separate servers. A computer may refer to a server by name, however this may go to different actual IP addresses in reality on different requests from different computers. This will be explained further.

Thirdly OpenLDAP also includes a replication server called slurpd that can automatically update several LDAP servers simultaneously with new information as it is entered into the LDAP directory, enabling several servers to mirror each other.

With the ability to direct LDAP requests to different servers depending on which section of the directory namespace that is being accessed as well as the ability to mirror critical servers, LDAP provides solid abilities to scale to support any number of users that may be required.

5.6.3 Domain Name System (DNS) and BIND

The Domain Name System (DNS) and the DNS resolving daemon BIND (isc.org, 2003) are a very good example of scalability in action. Typically, each computer on the Internet has at least one or more IP addresses registered in a freely accessible DNS server. These servers are specially designed to transform text addresses into IP addresses, and are used to allow us to easily remember names of servers over the more difficult to remember IP address.

However, most DNS servers can only resolve the local domain of DNS names to IP addresses or a small section of DNS entries. When a DNS server encounters a name they do not recognize, they pass the request onto a server they
know is able to handle it, which then either resolves the name or passes it on to the next server and so on.

Eventually the name will be resolved and cached by the DNS servers along the way. In this fashion, no DNS server has to store all the translations of names to IP addresses, yet is able to find out the IP address of any particular computer as required.

However, in order to support multiple servers in LDAP, the LDAP directory namespace must be split into several distinct sections. A completely separate LDAP server would handle each section. Chaining or referral techniques would enable the LDAP servers to communicate to each other as required.

This unfortunately does not help when a critical server is required to handle a commonly accessed section of the LDAP directory namespace, such as the root of the directory namespace. In this situation, multiple servers with the same information must be used along with a load balancing DNS server.

A load balancing DNS server will return different IP addresses to the same domain name in order to balance client requests to the servers available on an equal basis. Two of the main techniques are detailed below.

**Round Robin BIND Load Balancing**

Fortunately, the designers of BIND realized that some form of simple load balancing would be required. Currently, in BIND 9 a simple Round Robin load balancing system is available for use and is documented in the “BIND 9 Administrator Reference Manual” (isc.org, 2001). This simply has a list of IP addresses that relate to a single hostname, and the BIND daemon simply iterates through these to get a simple but effective form of load balancing.

**Dynamic Load Balancing**

Realizing that the Round Robin approach could be improved, several programmers developed alternatives to BIND such as Eddie (eddie.sourceforge.net, 2003). Using alternative DNS servers such as Eddie enable the IP address of the least loaded server to be sent to the client, which is a vast improvement over the existing BIND Round Robin approach.
Using BIND or specialised DNS alternatives it is possible to effectively load balance among many servers automatically without too much difficulty. When new servers are required, administrators simply need to update the DNS information to include the new server and it will automatically be put into use.

There are other techniques available for dynamic load balancing that utilise more sophisticated techniques, such as firewalls and network address translation (NAT) to redirect requests to a single IP address to a collection of identical servers, or even specialised hardware. To investigate these, it is recommended to read “Server Load Balancing” (Bourke, 2001) that gives a good overview of available techniques.

5.6.4 Synchronisation

However, with the advantages of scalability there also comes the difficult problem of synchronization among the servers. If data is not synchronized, incorrect values may be returned between different servers that can be confusing or alarming to users, such as incorrect passwords or out of date information.

OpenLDAP fortunately supports a daemon called slurpd, which enables replication among connected LDAP servers. slurpd works with slapd to handle updates among the different servers by designating one server to be the master repository of information, where all changes occur first. The master server then passes the updates to the slave slapd servers utilising slurpd. Using this with BIND’s scalable DNS resolution system, it is possible to have multiple servers working effectively together.

Unfortunately, this does not extend to other parts of the proposed system such as the HTML files that form the client interface or applications that exist on the servers such as Smart Card server programs. Two possible solutions are proposed.

The first solution is that a separate file synchronisation program could be utilised. This means that when changes are made to any of the HTML files, these would then be synchronised across the other servers. rsync (samba.anu.edu.au, 2003) is one such program that is widely used for this purpose.

Another solution would be to host the HTML server inside the LDAP database, mapping the HTML files and folders in the directory and then using slurpd
and slapd to automatically update the files using a custom written LDAP backend that specifically reads and writes files. Other files required by the proposed system could be handled in this way as well.

5.6.5 Backup Procedures

Backup procedures for GNU/Linux still are quite difficult to accomplish, due to the number of different possible solutions and servers that are available for GNU/Linux. This section will focus on the possible backup procedures for LDAP servers only.

As OpenLDAP supports replication through the slurpd program that sends a transaction log of changes to each of the replicated servers, it is possible to use this to backup OpenLDAP servers. Ideally, to minimise downtime it is best to have at least two servers. One server handles requests while the other is used to backup the data.

Firstly, the OpenLDAP slapd server and corresponding backend databases are stopped while slurpd is left running in order to obtain any transaction changes. The administrator would then be free to backup each of the backend databases. For instance, MySQL and BDB are simply stopped and their database files copied to a backup location.

Secondly, the servers are restarted and slurpd is allowed to reconnect to the OpenLDAP slapd server. This then executes the backlogged transactions, which then brings the backend servers up to date with the logged transactions.

If there were only one server available, the entire system would need to be stopped momentarily while the backup is performed. GNU/Linux supports scripting facilities to handle these tasks at a convenient time, such as early in the morning.

For actual backup to alternative media such as magnetic tape, it is recommended that a professional backup software package be considered. Various backup server software packages include Arkeia (arkeia.com, 2003), Dantz Retrospect (dantz.com) or BRU (tolisgroup.com, 2003).
5.6.6 Summary

As shown above, OpenLDAP provides adequate means of scaling to support numerous users over the Internet utilising the slurpd replication server and the ability to separate the LDAP directory namespace. This facility is considered essential in order to support small to large educational institutions.

5.7 Conclusions

As detailed in this chapter, the proposed system provides institutions with a clear migratory path towards implementing Smart Cards and related applications. Of particular interest, the proposed system has the potential to be fully backwards compatible with the existing systems in place, as well as providing a scalable solution that is of low cost.

Naturally, as each institution will have differing requirements, the above proposal should be taken as only a guideline or possible implementation for how such a system could be built for an institution. The institution should investigate other options or solutions that may be more suitable for their existing systems.
6.0 CONCLUSIONS

6.1 Benefits of Moving to Smart Cards

It is clear that tertiary educational institutions considering migrating to Smart Cards obtain advantages in that sufficiently advanced Smart Cards provide better security, advanced card based applications and backwards compatibility with existing institution card based systems.

For an institution, the promise of new and interesting possible applications utilising Smart Cards encourages new incentives and programs for the staff and students of the institution to investigate. In particular, the electronic purse facilities have proven to be quite popular among institutions for small purchases and fees, and may prove to be a popular use of Smart Cards depending on how it is handled.

The improved security of Smart Cards may also make it possible to use Smart Cards as a sign sign-on system for computer and security access, while also being more difficult to duplicate and be able to be exploited fraudulently.

Finally, the backwards compatibility with existing card systems is of great benefit to the institution as it would be able to phase in Smart Cards and have more than one system operating in parallel in order to minimise disruption to existing systems.

6.2 Comparative Analysis of the Tertiary Sector

Having analysed several tertiary institutions that currently are or were utilising Smart Cards, many Smart Card implementations are primarily designed to replace the existing card technologies and provide a suitable environment and opportunity to gradually phase in new applications and services.

For example, all Smart Cards supported the traditional services such as an identification card, library borrowing card and security access card. The cards also
provided confirmation of discounts for items and services at specially selected stores such as cafeterias, bookshops and related institution stores, in addition to discount public transport and institution services.

More advanced services such as electronic purse facilities for printing, photocopying, Internet access fees, food and beverages were also not uncommon among Smart Card implementations and is one of the primary new applications available to institutions migrating to Smart Cards.

The institution's level of industry links to the banking and telecommunication companies determined whether their Smart Card could also be used as a telephone calling card or ATM/credit card. At the moment, not many institutions were able to implement these advanced features.

Of particular interest were the failed attempts to migrate to Smart Cards. Following up on institution attempts it appears that there is a critical point where at least one primary application must be supported by the Smart Card utilising the embedded IC, such as security access. Failure to migrate at least security access appears to enable an institution to revert back to existing card technologies despite any possible advantages it has brought the institution.

Unfortunately, the use of the electronic purse facilities by itself an insufficient advantage that Smart Cards have over existing card technologies and it appears as though supporting these facilities can be removed easily. With security access firmly in place utilising Smart Card technologies, an institution would face considerable costs in replacing the security access infrastructure to revert to existing card technologies.

Of final interest, there are also possible privacy concerns when an institution migrates to utilising Smart Cards. These concerns should be carefully considered to avoid any future problems. There are a growing number of people who find technology such as Smart Cards as an intrusion on their privacy if information is handled incorrectly.

This also has problems when institutions seek industrial links to provide advanced services such as calling card numbers and bank connections to enable the Smart Card to operate as an ATM or credit card. Several people may view this as an attempt to commercialise the institution and outsource basic services, and so it may
not be entirely welcome. Therefore, care must be taken that any proposed changes do not upset or antagonise the staff and students of the institution.

6.3 The Proposed University Wide Smart Card System

As detailed, the proposed University wide Smart Card system primarily aims to provide a suitable backwards compatible infrastructure for Smart Card applications. This is important as the institution can then successfully migrate towards implementing Smart Card applications and services without disrupting existing card based services already in use at the institution.

Within the proposed system, the Smart Card is primarily used as an identification device with a unique ID encoded onto the Smart Card. Depending on the type of Smart Card, other application specific data may be stored onto the Smart Card itself as required by the institution.

As it is expected initially that there would be low numbers of Smart Card enabled client terminals, it is recommended that the institution support clients without Smart Card capabilities as much as it can do so. Fortunately, the web based client front end system with Apache and PHP is fully capable of determining the type of client and providing the appropriate interface automatically so it is more a institutional policy issue to deal with.

The combination of PHP with the Apache web server provides a potent combination of Open Source software that enables true dynamic web sessions for staff and students logged into the proposed system. In order to communicate to the existing institution systems, a common protocol is required to control and simplify data connections between the different existing systems.

To accomplish this, LDAP and the Open Source OpenLDAP server is utilised in order to provide a common communications protocol between the different systems and in order to provide a suitable scalable directory namespace design to accommodate each of the existing and newly developed systems. By utilising LDAP the complicated decisions in connectivity are simplified and controlled to a manageable level.
Finally, the proposed system is capable with dynamic load balancing techniques and a suitable namespace design to scale the system to handle as many users as required. This will prove to be a critical issue towards providing key services to the institution.

### 6.4 Conclusions

This thesis I believe has provided adequate and relevant information for an institution to begin research and development on their own University wide Smart Card system. Due to the number of different possible ways that the system could be implemented, I have preferred to avoid detailing any specific ways to handle such a system and have preferred to outline general ideas on how a system could be designed instead.

In particular, I believe the use of LDAP will significantly rise in the future as a means of resolving difficulties in data communication from one system to another. The proposed system is an example of how LDAP can be used to enable connectivity between existing systems and unify them into a single common interface.

Also included in this thesis is relevant information on other institution’s attempts to integrate Smart Card technology, and provides an adequate set of examples of how and what they have accomplished to date. More importantly, this thesis provides details on failed attempts to integrate Smart Card technology, providing essential information on how to avoid mistakes.
REFERENCES


70


APPENDIX A: ABBREVIATIONS

ACL  Access Control List
API  Application Programmer’s Interface
AT  AT, the older standard form factor for PCs
ATM  Automatic Teller Machine
ATX  ATX, the current standard form factor for PCs
BDB  Berkeley Database
BIND  Berkeley Internet Name Domain
CA  Certifying Authority
CGI  Common Gateway Interface
GDBM  GNU Database
GNU  GNU’s Not Unix
HTML  HyperText Markup Language
HTTP  HyperText Transfer Protocol
IANA  Internet Assigned Numbers Authority
IC  Integrated Circuit
ID  Identification
IP  Internet Protocol (Address)
ISA  Industry-Standard Architecture
ISO  International Organization for Standardization
IT  Information Technology
ITU  International Telecommunications Union
JNDI  Java Naming Directory Interface
JNMI  Java Native Method Invocation
JRE  Java Runtime Environment
JSP  Java Server Pages
LDAP  Lightweight Directory Access Protocol
NAT  Network Address Translation
NOS  Network Operating System
NSW  New South Wales, Australia
OID  Object Identifier
OS  Operating System
OSI  Open Systems Interconnect
PC  Personal Computer
PCI  Peripheral Connection Interface
PHP  PHP: HyperText PreProcessor
POS  Point of Sale
PS/2  The standard mouse and keyboard connection for ATX computers
RFC  Request for Comments
SQL  Structured Query Language
SNMP  Simple Network Management Protocol
SSL  Secure Sockets Layer
SSN  Social Security Number
TAFE  Technical And Further Education
TCP/IP  Transmission Control Protocol/Internet Protocol
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA</td>
<td>United States of America</td>
</tr>
<tr>
<td>USB</td>
<td>Universal Serial Bus</td>
</tr>
<tr>
<td>WWW</td>
<td>World Wide Web</td>
</tr>
</tbody>
</table>