Smart Card Tutorial - Part 1

First Published in December 1998

Smart Cards and the Future

Now that we are fast approaching the millennium it is interesting to look at what has been happening in the world of Smart Cards and what we think is likely to happen in the future.

We will look at several aspects of the industry to try and identify the core concepts that will shape our future. There are four concepts that we believe will be at the heart of these matters:

- Applications
- Technology
- Infrastructure
- Security

Let us look at each of these issues in turn.

Smart Card Applications

Everyone is looking for the killer application, the one that returns the business case ten times over. We will propose here that we can classify applications into two types:

- Prepayment
- Authentication

Prepayment as an application area falls into two types, closed systems and open systems. Today the largest use of Smart Cards is for prepaid telephone cards. This is a closed system in as much as that the cards are issued by a particular service provider for use in his infrastructure. The same would be true for mass transit applications, which are likely to increase over the next few years. The chips used in these cards are very simple memory chips or chips with limited security functionality. The chips are small and can be produced at a relatively low cost, typically 50 cents or less. Because of the closed nature of this application a simpler security architecture is possible which significantly reduces the complexity of the chip.

The prepaid electronic purse is another major application area that has yet to come to the fore. The greatest interest here is for open payment systems where the issuer and acceptor may be commercially unrelated. Because you are effectively dealing with a cash alternative the risks make this significantly different and the requirements for security are much higher.

At this time there are a small number of major players, VisaCash, Mondex, Proton and the Geltkarte being the most obvious. Amongst these schemes Mondex stands out as being different since the underlying model is a direct asset transfer system. The other schemes actually effect a remote asset transfer system. This also affects the risk model in as much as that, in the case of Mondex, were there a fraud problem it is the Mondex value issuers that lose value. In all the other schemes it is the recently held assets of the cardholder that is on risk. However Mondex offers significant advantages in that it most closely maps the use of cash and avoids the interest overheads of processing individual transactions. For low value payments this is an important consideration.

It is our opinion that the electronic purse has its greatest advantage over cash in three application areas:

- Electronic ticketing
- Low value remote payments
- Meters (e.g parking)

At the current time the penetration into all these three areas is still relatively limited largely because the necessary infrastructure is still in its infancy. There is much talk of micro payments on the Internet but again we are still in the horse and cart game. Information providers cannot accept such payments until the infrastructure is in place and consumers will not apply for such payment mechanisms until there are enough services in place to justify their use.

Many observers have commented that there is no need for the electronic purse in Internet payments, it can be handled by existing debit/credit card instruments either directly or through the use of accounts held by the service providers. In our view the truth of this statement depends on the size of the payments and the multiplicity of service providers. For small payments you are forced into a secondary accounting system and nobody would want to set up hundreds of individual accounts.

So in our forecast for the future we see an increase in the use of closed prepayment schemes and a substantial increase in the use of open prepayment systems. A particular but so far underplayed advantage of the electronic purse is that they can be held and used by people who do not qualify for the use of other payment instruments. In particular minors represent a large sector of the population with an important spending power. It should also be noted that many of these prepayment applications are essentially anonymous

The simplest authentication application is really the classic ID card where the scope of use seems almost unbounded. There are really two underlying mechanisms:

- Cardholder Authentication
- Card Authentication
- Transaction Authentication

Card authentication is relatively straightforward in that the card can contain a unique cryptographic identity that can be built into local and remote authentication processes. Limiting the cardholder to the card has always been the more elusive problem. The use of a PIN (Personal Identification Number) is well known but there still remains the problem of secure terminals that can pass the PIN into the card for checking. The only one you really trust is the terminal that you totally control. The use of biometrics is often touted as the saviour to this problem but an untrusted terminal could just as easily replay a biometric measurement as a PIN. It does however help the problem of impersonation if you can overcome the error problem. If you make the biometric control too rigid then you end up rejecting the true owner. If you are too generous on the controls then you make impersonation too easy.

There are two new application areas where the Smart Card is viewed to play a major role, access to computer workstations and LANs (Local Area Networks), and access to the Internet. Whilst these might still appear to be relatively small, being measured in a few tens of millions, there is no doubt that it will become far larger. It is in this area that Microsoft focused when they announced their Windows Card earlier this year.

In our classification we can differentiate between ID cards with local storage and ID cards with remote data storage. You could of course have a combination of both, but it seems to us that applications will fall in general into one of these two classifications. Looking at the basic application areas we can see driving licence and passports storing data on the card for local use whereas the access card for the Internet, as referred to previously, will be largely concerned with the management of remote data resources.

If we consider conventional credit/debit card instruments we are really providing an authentication mechanism for accessing remote accounts. The card itself provides the account identification but the main accounting database is held remotely. If you look at Visa and MasterCard there are today something approaching 1bn cards in circulation worldwide.

The use of Smart Cards for these payment instruments does, however, offer an additional security service and that is the ability to authenticate instructions, in this case a particular payment transaction.

The fact that we have included all these applications in one area represents our view that in most cases an identity card has little value if it cannot authenticate instructions whatever the particular application may be. In other words a passive identity card that just stores identification data has little commercial value, it is nearly always necessary to challenge the card and owner to prove their identity and to be able to show convincingly, after the event, that the card was truly used.

Dr. David B Everett

Smart Card Tutorial - Part 2

First Published in February 1999

Smart Cards and the Future

Sometimes we wonder where the technology will ever end, the pace of development seems unbounded. In the early days of the microprocessor many experts seemed convinced that there would never be a need for a microprocessor with more than 8 bits yet in just a few years 32 bits has become the norm. Then the much quoted albeit inaccurate report of Bill Gates that there would never be a need for more than 640K of memory. Today on the humble PC few would set out with less than 32Mbytes of memory. So what about the Smart Card? In the early 90s we boasted about Smart Card chips with 1K byte of E2 and 256 bytes of RAM. In the mid 90s 8K bytes of E2 and 512 bytes of RAM became the norm. Where are we now at the end of the millennium?

The major manufacturers are offering 16K bytes of E2 with between 1K byte – 2K byte of RAM complete with a 1024 bit crypto co-processor. Over the same period the ROM has increased from 6K bytes to 32K bytes whilst the ubiquitous 8 bit micro-processor has moved to 16 bits with 32 bit RISC processors gently emerging. The semiconductor process technology has moved dramatically to enable these extensive resources for a single chip microprocessor. In the early 90s 1.2uM feature size was common but now we see 0.65uM with every sign that it will move to 0.35uM over the next few years. Even though the resources are getting bigger the chips themselves are getting smaller. The early Smart Card chips struggled to keep within the form factor of the contact plate which limited the size to about 25mm2, now we are looking at half that size.

So what is going on behind the scenes? Simple ID/authentication applications clearly do not need these extensive resources and the associated higher cost of the chip. However when you look at the multi-application operating systems then the larger the memory the more you can do with them. The early driver of large E2 was the GSM application, which still today is the main consumer of the modern Smart Card chips. It seems clear that for the short term this will continue and the most likely development is for the multi-application operating systems (Multos, Javacard, Windows card) to pick up GSM as a primary application allowing a more general extension to the infrastructure provided by the GSM networks. Anything less than 16K bytes of E2 would be unacceptable and to allow adequate working space for applications running under these virtual machine architectures 1K bytes of RAM would seem to be a minimun.

Having set the scene for the picture let us now conjecture what may happen over the next few years. First we will consider the applications. As discussed in our last article the underlying mechanism for most applications is ID with an associated authentication process. This in itself requires little memory either for the application code or its associated data. In fact the larger consumer of E2 memory is the public key certificates that underlie the emerging PKI (Public Key Infrastructure) development. The use of RSA tends to result in certificates that occupy about 1 K bytes of memory for each certificate. The problem here is how many certificates do you need to store. The current fragmented structure of the existing PKI schemes requires a number of certificates to be stored, at least one for each scheme in which the card needs to participate. However we expect a consolidation of these schemes so that over the next few years a small number of global schemes will dominate.

However we do expect the multi-application cards to become a significant part of the overall market and this will lead to a requirement for ever increasing memory size. But is there a practical limit? In terms of the basic technology no, but there are limitations in the ISO 7816 standards that potentially will cause problems. Most Smart Cards operate at very low clock speeds, typically at 3.58MHZ with a simple half duplex serial port. Serial communications using the ISO T=0 and T=1 protocols invariably operate at 9600 baud. This means that getting data in and out of the card is a pretty slow process. Now there is no inherent reason why we shouldn't use higher clock speeds and hardware serial port controllers. Witness the popularity of the USB (Universal Serial Bus) now becoming common place on the PC. The overhead here will be higher

power consumption and larger chip area to incorporate the hardware controllers and of course we will need new standards. What you might wonder here is are we re-inventing the PCMCIA specification.

If we were only interested in using our card in conjunction with a PC then arguably a PCMCIA card would be fine but we pick up a number of disadvantages. The most obvious is cost, the PCMCIA interface connector is relatively expensive compared with the simple Smart Card contact plate and the overall fabrication is more costly than sticking a micro-module into a piece of plastic.

In round figures a PCMCIA card is always likely to be an order of magnitude more expensive than a Smart Card. Then there is the form factor for which even a type 1 PCMCIA card is far thicker than the ISO Smart Card format.

In the same way that PCMCIA card is more costly than the Smart Card the reader is also more expensive. While today there are more PCMCIA readers on computers than Smart Card readers this situation is about to change as we are already seeing computers shipped with a Smart card reader.

In terms of read only memory the current ROM ion implanted technology is very efficient but clearly it has the enormous disadvantage that the chip has to be manufactured against a previously constructed ROM image. Some manufacturers have provided chips with flash memory, which can be re-written about a hundred times. The read time of this memory is fairy fast at the expense of a slow write time but arguably if this memory is used for loading applications units it doesn't matter.

The static RAM memory is of course the most expensive real estate on the chip needing six transistors for a single memory cell. For larger RAM arrays dynamic RAM is far more efficient but this certainly doesn't apply with the 1-2K bytes of memory provided with the current technology.

The non-volatile memory is the next most expensive real estate on the chip. Today we are used to the ubiquitous EEPROM memory. There have, over recent years, been some moves towards using Ferro Electric RAM (FRAM) but there are still significant problems with this technology. Although in principle it is possible to construct FRAM with a single storage cell in practice two cells operating in a differential mode have been necessary to get adequate reliability. This negates the potential simplicity of the FRAM memory size. Since all reads of an FRAM cell require a read/write cycle the endurance is limited by all access to the memory regardless of whether it is a read or write. Current E2 memory has a write endurance of 105 -106 cycles whilst the read cycle is practically unlimited. The FRAM cell has an endurance of about 1010 cycles, for most applications this is probably not a problem whilst the faster write time is a significant advantage.

It is the properties of the ferro electric material which cause the greatest problem. These materials are heat sensitive and cannot withstand many of the high temperatures processes used in the fabrication of the overall chip. As a result the FRAM cells have to be fabricated as one of the last stages in the chip fabrication process. There are concerns that this is potentially a security vulnerability since any sensitive data in the chip is most likely to be stored in the non volatile memory which in this case would be near the surface of the chip.

Over recent years there have been many advances in the construction of contactless cards. The single chip Combi card, which has both a contact and contactless interface is typical. More recently we have seen contactless interface chips appear which can be interfaced directly to the existing Smart Card microcontroller. The interest here is what is the application for contactless cards, clearly in the area of mass transit the contact card is totally unacceptable. But do we see GSM cards or financial cards using a contactless interface, probably not because the contact interface in the terminal is easier and cheaper to realise.

At the end of the day for a Smart card application to succeed the business case has to be justified. The GSM card neatly circumvents the cost problem because it can be viewed as part of the mobile phone. Today this is still the best business case for an application that uses a relatively large and therefore more expensive chip.

However the application of the latest technology to the Smart Card chip results in smaller chips and cheaper cards. The contactless cards being used for payphone applications are now being manufactured for less than 50 cents (US).

Ten years ago people searched for the $1 chip with large memory and a crypto coprocessor. Today this sort of figure is becoming a reality so that the overall card price can be well under $2.

So looking into the next millennium we can see telephone cards still occupying top place in the volume stakes. But the real added value is going to come from the more sophisticated chips with large memories and multi-application operating systems. It is these multi-application operating systems that we will look at in more detail next month.

David Everett

Smart Card Tutorial - Part 3

First Published in March 1999

Smart Cards and the Future

In looking at the future of Smart Cards there is really only one question we would like to answer "who is going to make money, where"?

The business of Smart Cards has changed so much over the last few years that we need to have a look at the then and now. In the early days of the Industry card companies such as Gemplus would produce the total card as a solution to a problem. The Card Company would design and develop the software programs to be manufactured into the ROM by the silicon manufacturer. On receiving the silicon wafer the manufacturer would then produce the micro modules for embedding into the card plastic. The same company would then print the cards and personalise the application with the relevant data. Such cards might typically be sold for about $5 of which up to 50% would be recovered to pay for the development process and the necessary profit stream. A production run of 1 million cards would return several million dollars.

Now lets move to the end of the millennium where life in the Smart Card world has become a lot more competitive. Major applications such as GSM are produced against open specifications such that manufacturers have to compete aggressively. In such a market we can see the total card price in the region of $2 - $3. The contribution for the software is estimated to be in the region of 10% - 20%. A production run of 1 million cards might now only return $200,000 to pay for the software development.

Just before passing on we need to agree that such a return is totally unacceptable in that it would not cover costs. What can be done in a back bedroom for $20,000 ends up costing at least $400,000 as a production item. The point that we are making so far is that the providers of Smart Card software either have to generate large volumes, which incidentally is why the GSM application is so successful (65 million cards in 1998), or the application provider has to enter a market that is not price sensitive. Such markets are often by their nature short lived because they either fade out or they expand into competitive volume markets.

It is the emergence of the Smart Card as an essential token in our electronic world that is causing these new market pressures. We have mentioned GSM as the current success story but popping out of the wings we can see financial cards (credit, debit and electronic purse) and probably even more important the ID cards that will become the backbone of electronic commerce.

Interoperability and cost go hand in hand. The PC market has developed on a common infrastructure that allows interoperability between applications on different hardware implementations. The operating system in this case Windows xx hides the details of the particular hardware from the application software. In principle at least we can run our application software on any PC providing say Windows 95. Well that's what it says on the box, and in general it works. In the case of the PC we have the advantage of using a common processor base from Intel.

In the case of the Smart Card life is somewhat different. We have about six major manufacturers of chips which to all intents and purposes have little in common. There is no way that a native application written for a Siemens chip will run on a Hitachi chip, or any other chip come to that. Both the software developer and the consumer lose by this approach. The software developer by the previous argument needs volume to pay for his development and to give a return on his investment. There are far more developers working on the PC than the Macintosh and here we really only have a choice of three (UNIX machines being the other alternative). It is this very problem that has given rise to Java, which gives you the ability to program with platform interoperability. The advantage to the end consumer is equally clear.

Apart from platform interoperability Java is also intended to manage another important requirement, security! Much is spoken about the security of the Java sandbox but the basic principles are quite simple. If we invoke a common abstract machine on top of the real machine then all applications are written to

interface with the abstract machine, the real machine is hidden. Now it is only necessary to write the applet once, in the language of the abstract machine. Whenever we meet the same abstract machine regardless of the underlying platform (assuming it has sufficient available resources) our application will run. But the abstract machine is controlling all access to the resources of the real machine, the memory, I/O, disc drives etc.

This means that we can in a well-designed system stop one application from interfering with another. *Figure 1* shows the difference between these two approaches. It is important to note that in the real machine environment that the applications can interface with the hardware resources direct. This means that one application can interfere with another application's memory storage. One application could read the secret cryptographic keys of another application. The only way to stop this is by having special hardware that controls the allocation of memory to applications, in other words a hardware memory management unit (MMU). Looking at current Smart card chips only the new Philips Smart XA offers this sort of facility.

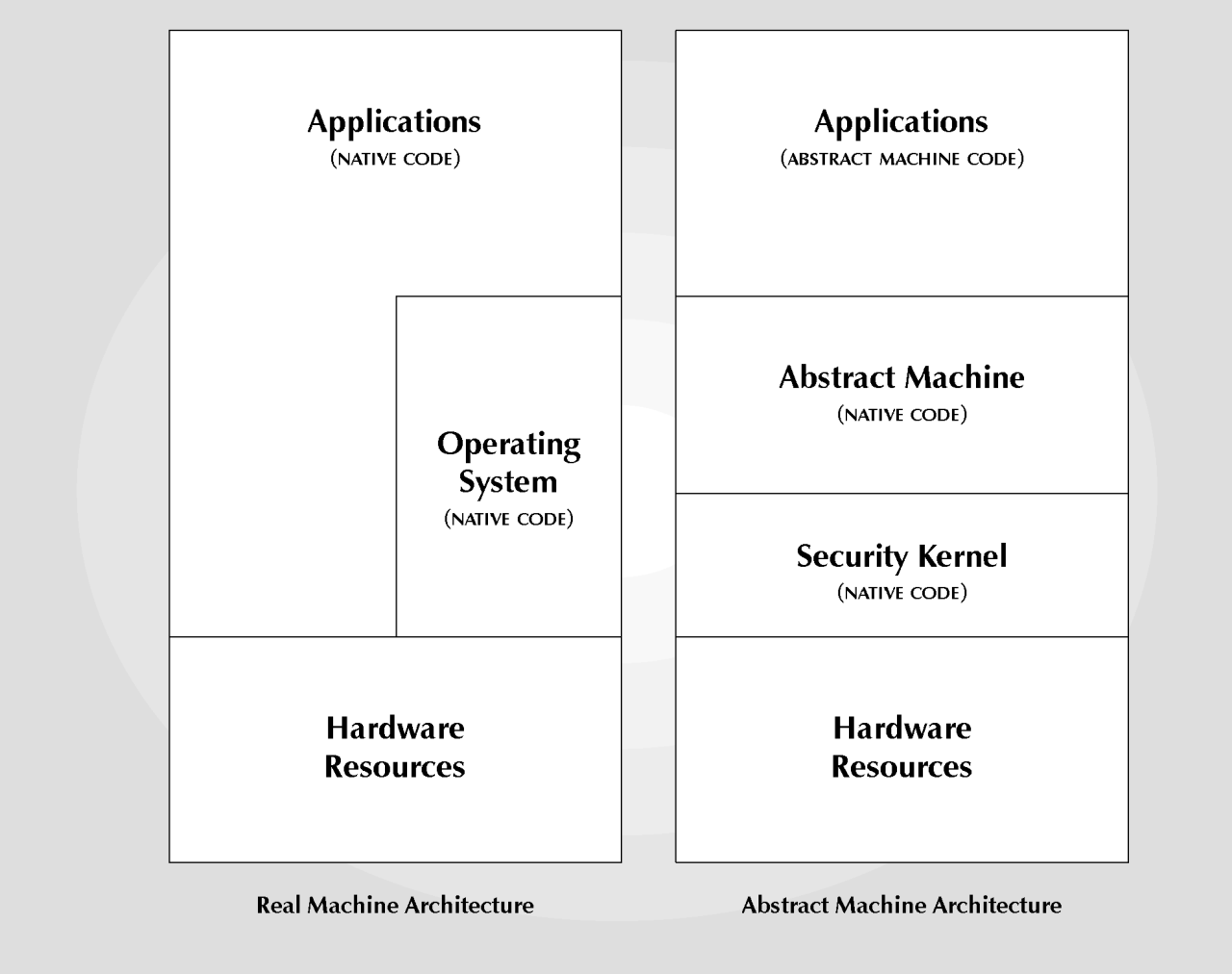

Figure 1: Abstract and Real Machines

This approach to Smart Cards has now been adopted by three organisations:

- Maosco Multos
- Sun JavaCard
- Microsoft Smart Cards for Windows

In each case the operating system is composed of two parts, a virtual machine which is a particular specification of an abstract machine and a security kernel which forms the interface between the virtual machine and the hardware resources.

We can see immediately that the business model for the traditional card companies has changed. Because the definition of the virtual machines are public it opens up a New World for software developers to enter the Smart Card arena. The card companies are also directly competing with each other on a standard product so their differentiation has to be based on implementing the same thing better or cheaper than their competitors. In any event the margins for software are much reduced but the volumes are potentially much higher.

There is also the secondary advantage for application developers of being able to obtain off the shelf products because the ROM mask is fixed and can be manufactured with no prior knowledge of the applications.

So is this a win-win situation for all the players? The semiconductor companies end up with manufacturing standard products. The card companies can use a standard chip which optimises the cost of the card but their traditional margins are much reduced so they need to move higher up the food chain as system integrators which will produce a much higher added value. The consumers also gain because of the lower cost of the core component.

But there must be a snag and of course there is. The virtual machines put in an extra software layer that is at the core of the application execution. Every abstract machine instruction has to be converted to one or more real machine instructions in real time. This results in a performance degradation of about 1:20 in the conversion from the abstract machine to the real machine. However in general applications will not run 20 times slower. The designers of the security kernel will provide libraries written in native code that executes the core routines of the Smart Card applications. The cryptographic algorithms for instance will be executed in native code accessed from the virtual machine. Typically we might expect Smart Card applications to run some 10% - 20% slower but given the improvement in performance of the chips this is likely to be perfectly acceptable.

What we are really saying here is that the Smart Card is much more likely in the future to become a commodity where scale of volume is necessary to produce the necessary revenue returns. The higher added value will accrue to the system integrators and this is where the battleground of the future will be.

David B Everett