JCOP

A CROSS-INDUSTRY VALUE PROPOSITION

EXECUTIVE SUMMARY

JCOP is the open IBM smart card operating system. This paper shows how JCOP adds substantially to the value proposition of smart card-based solutions for card issuers, primarily by software technology-induced improvements in hardware costs, operational efficiencies gained in card manufacturing, and finally, card-lifetime cost savings by adherence to independent standards.

Therefore, this paper highlights the dedication of IBM to smart card industry-standards, the use of an impartial licensing strategy, and the technology by which low-end smart card microcontrollers can be brought in line with JavaCard, GlobalPlatform, ISO, and ETSI standards to be of use in all existing and emerging markets using smart card technology, such as banking, telecommunications, identification, transport, health, etc.

On one hand, JCOP combines the disparate smart card technology approaches in various industries by a cost-effective technology that improves upon proprietary approaches in performance and functionality. On the other hand, with a highly configurable technology available, JCOP is impartially licensed to all partners in the smart card value chain in any industry to ensure a synergistic growth in business value.

MICHAEL BAENTSCH

JCOP

A CROSS-INDUSTRY VALUE PROPOSITION

INTRODUCTION

JCOP is the open IBM smart card operating system based on open, third-party standards, such as Java, GlobalPlatform, ISO, PKCS, and others. JCOP relies on three pillars for customer success: One is utmost technological superiority, i.e., the ability to provide the smallest, fastest, and functionally richest implementation of open standards. The second is the possibility to tailor JCOP according to customer demands. The third is IBM's servicing model, putting the card issuer at the center of attention – total standards adherence, cross-industry interoperability, increasing operational efficiencies, and lowered card lifetime costs are the cornerstones of this model.

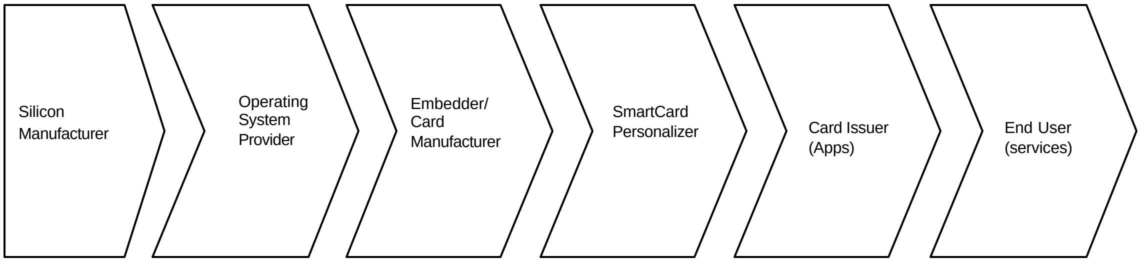
Therefore, this document first revisits the market setting, discusses the generic benefits of open over proprietary systems, before the IBM technology and business attitude are explained in more detail.

MARKET SETTING

SMART CARD INDUSTRY

The smart card industry's top 5 players currently still draw a significant amount of their $3billion revenue from raw manufacturing processes around plastic cards, and stored-value memory cards. However, this business is a total commodity business without much opportunity for differentiation by card manufacturers and card issuers alike. Thus –and because of the urge for higher security services– cards with embedded microcontrollers have become more and more important. Accordingly, in 2002 alone, it is expected that over a billion of such cards are shipped.

Two main strands of embedded software are found in these cards: One is proprietary to the respective card manufacturer; the other is based on an open, third-party standard. In order to explain the benefits of either approach to the manufacturers and issuers of smart cards respectively, the card value chain is highlighted below.



Example companies in each section are:

Silicon Manufacturer: Infineon, Philips, ST Microelectronics, Samsung, Hitachi, …

Operating System Provider: IBM, Gemplus, Schlumberger, Datacard, Oberthur, …

Embedder/Card Manufacturer: Schlumberger, Gemplus, Oberthur, G&D, Trueb, …

SmartCard Personalizer: TSYS, Gemplus, Schlumberger, Oberthur, Trueb, …

Card Issuer: Visa banks, Vodafone, Mastercard issuers, Orange, Governments, …

End User: Bank customer, Mobile Phone user (SIM), health care customer, citizen, …

This value chain is very much like any other commodity value chain in which the highest amount of value is added only in the very last stage, namely the service provisioning to the end user. All manufacturing stages are subject to fierce competition.

Historically, only the card manufacturers have developed –proprietary– card operating systems, thus fully controlling the center piece of the value chain. The net result of this was card issuer lock-in, followed by technological stagnation. Standards-compliance was claimed, e.g., to ISO or PKCS, but as those standards leave implementation gaps open, a technology lock-in was possible, and indeed occurred.

CUSTOMER VALUE

CUSTOMER DESIRES

If it were not for technological lock-in, the card issuers would want to make the ultimate purchase decisions on which technology is to be used. This decision is largely driven by multi-sourcing concerns: Card issuers are clearly aware of the difference in value open, third-party systems are bringing them over proprietary systems. All card issuers have experienced the drawbacks of being technically locked in to a specific proprietary technology, which in the case of smart cards always had its root cause in the smart card operating system implementation.

Therefore, card issuers now demand the same they do for the rest of their operations, namely the capability to source all external components from *different* providers. The goal is clearly to be able to benefit from competitive pricing, but also from an independence of single sources.

In practice, for example, no mobile phone operator relies on smart cards based on one silicon provider alone, nor does it accept its applications being tied to one operating system alone. Instead, application development should be handled by a totally different entity than the provisioning of the operating system; yet other competing entities are to be able to provide the personalization services and the card-lifetime managing services.

LIFETIME COSTS

For the card issuer, the costs of a smart card consist of all work and material elements incurred during production, distribution and use of a card. Therefore, all prices matter, most prominently those of the silicon, the operating system, the applications, the plastic printing, the card personalization, the card shipment, and finally, the card lifetime management. Also negligible is the card printing and embedding as this is a highly standardized process in which strong competitive forces act to keep prices at a minimum. Likewise is silicon a commodity on offer from various companies that vie for market share with various strategies, most of which are price-oriented, though.

The one major cost driver to the customer is thus the software involved: One alternative is the decision for a proprietary operating system, which is –due to smaller intellectual property and brand licenses– typically less expensive than open, third-party standards based ones. Proprietary systems, however, automatically incur higher application costs, as proprietary knowledge has to be build or –if external support is used– paid for. Similarly, the personalization of proprietary cards and applications consequently incurs higher costs due to the need to set up customized procedures. The same is valid for any card-based post-issuance service, e.g., those for card management systems: Proprietary technology-based solutions have to be more expensive due to the necessity to involve the owner of the proprietary technology.

The alternative is the use of card operating systems based on open standards. The obvious drawback of higher brand costs is more than offset by the costs saved in further card life cycle stages: Applications can be developed either cheaply by the card issuers themselves, or any of thousands of independent software shops capable of coding in Java. Personalization is also trivial as both the operating system as well as the applications are standardized to the degree that no premium prices for the card initialization can be charged. Similarly, if high-end multi-application cards are considered, the costs of a card management system handling standardized cards will surely be smaller than one where continuous proprietary customizations have to be paid for.

The only remaining –perceived– drawback of cards based on open standards is their often quoted –but untested– sluggish performance, or their –rumored– requirement for high-end card hardware. While these properties may be true for some implementations, JCOP is the proof to the contrary:

- Any version of JCOP runs on low-end 8 bit smart card controllers with little resources, which directly translates to savings in silicon size and thus hardware costs.
- JCOP implements a full 2.5G JavaCard/GlobalPlatform/ETSI/WIM compliant card in less than 64kBytes of code on a low-end smart card chip
- Applications running on JCOP can show –if the applets have been properly programmed– performance characteristics of proprietary cards; independent tests have shown JCOP to be the fastest overall Java/GlobalPlatform card on the market today [1].
- JCOP implements all functional capabilities required in smart cards today, be they contact or contact-less operation, DES or RSA cryptography, PIN or biometry authentication, to name just a few dimensions.
- JCOP is a family of products running on a diverse range of controllers, thus ensuring a one-time only learning curve for customers regardless of the growth path of solutions; near-zero customer switching costs between members of the JCOP family for maximum flexibility of smart card-based solution deployment.

RESULTS

Most card issuers are well aware of the issues outlined above, and are increasingly demanding full standards compliance from their card manufacturers. Over the course of the past five years, there has been a clear winner determined by the market place, namely the combination of JavaCard and GlobalPlatform on top of the pre-existing ISO communication standards. This is clearly exemplified by the rapidly growing market shares of such cards. The role of Java in the telecommunications industry is already widely touted; what is new is its prevalence in the banking industry: In just one year, one Java-based product alone was issued nearly three times as often as all MultOS cards over all the years: This product was IBM JCOP.

IBM JCOP implementations have shown that open, standards-based operating systems have no inherent technological disadvantage over proprietary systems: Functionality and speed of proprietary systems have been matched and exceeded [2]. Consequently, all business advantages of open systems as outlined above can be garnered by flexibly using members of the JCOP family as needed.

IBM'S CUSTOMER COMMITMENTS

TECHNOLOGY

- *100% standards compliance*

  JCOP must always be able to prove better adherence to standards than any alternative, be it proprietary or based on some open standards, as this is the core property sought by customers looking for multi-sourcing, and vendor independence.

- *Best industry performance of products*

  In the face of increasing competition, JCOP will continue to beat the competition on functionality and performance, as these are the prime customer-tested properties of standards-based systems.

- *Continued improvements on operational efficiency*

  Creating one software version for use in many industries, together with a flexible customization (see Appendix A) are the foundation on which further technology-based improvements for operational cost savings are implemented.

BUSINESS

- *Open Systems*

  IBM is committed to the successful deployment of open systems for customer benefit.

- *Cross-industry application*

  JCOP technology can be applied in solutions in various markets, possibly bridging technology-induced gaps enabling new uses and business relationships.

- *Independence*

  JCOP is made available impartially to all IBM business partners, neither favoring nor putting at a disadvantage suppliers, distribution partners, or end customers of smart card solution elements.

APPENDIX A: JCOP CUSTOMIZATION TECHNOLOGY

JCOP CONFIGURATIONS

The JCOP base system consists of various independent components, such as a high-performance virtual machine, transaction subsystem, contact- and contactless communications modules, various JavaCard and GlobalPlatform APIs, and a plethora of cryptographic libraries.

Upon key customer demands, these components can be joined together to fill different needs: Thus, it has been possible to create a very inexpensive DES-only mask with standard post-issuance loading capabilities (JCOP10), as well as high-end public key systems featuring on-card RSA key generation capabilities with up to 2048bit key lengths using standard 8-bit contact-only smart card controllers (JCOP20, JCOP21sim), or even dual-interface controllers adding contactless operations (JCOP30).

None of these base configurations are tailored in any way for a particular application. The base system only implements what is the standards requirement for the financial sector (ISO, EMV, JavaCard, GlobalPlatform compliance), the telecom sector (ISO, ETSI, JavaCard, GlobalPlatform compliance), or any other industry targeted (ID/Health: PKCS; transport: Mifare; etc.). The actual tailoring for specific solutions is done independently by the card issuer by adding specific third party applications (applets) to the base system.

This flexibility of design of the JCOP family ensures that customers are getting a maximum degree of choice for their smart card solution needs. When switching between members of the JCOP family, no additional learning costs are incurred: All members of the JCOP family behave identically where none of the differential features, e.g., RSA vs. DES, are involved.

THE CUSTOM MASK PROCESS

In an effort to reduce the manufacturing cost of multi-application cards, IBM has developed a secure procedure for the inclusion of application code into ROM sections of a *Custom Mask*, i.e., a piece of software tailored for one specific smart card solution scenario consisting of the applications the card has to cater for.

Provided the card issuer has (had) the different applets developed using the JavaCard and/or GlobalPlatform APIs, the option to load such applications into EEPROM is flexible, but is relatively expensive, as it costs

- EEPROM space (silicon cost; opportunity cost)
- Card production time (initialization/personalization cost)

The former relates directly to cost of silicon, as more EEPROM always means bigger, and hence, more expensive chips. The latter relates to personalization cost, as each second that the card has to be programmed longer, high-volume production runs are slowed down, and are accordingly priced higher. Also, the smaller the free EEPROM is because of loaded applets, the higher is the opportunity cost of smart card related services that could have loaded applets developed later post-issuance into otherwise free memory.

The answer to this problem is a process that IBM has established and makes available via the silicon manufacturer: Without involvement of IBM, tested applets intended for large-volume rollouts can be delivered by a customer to the silicon manufacturer directly for integration with the desired version of JCOP. Due to the design of JCOP, as well as the established security procedures, it is ensured that the applet runs from ROM in the same way it would run from EEPROM, and that it cannot influence the rest of the system in any other way than if it were loaded post-issuance into EEPROM.

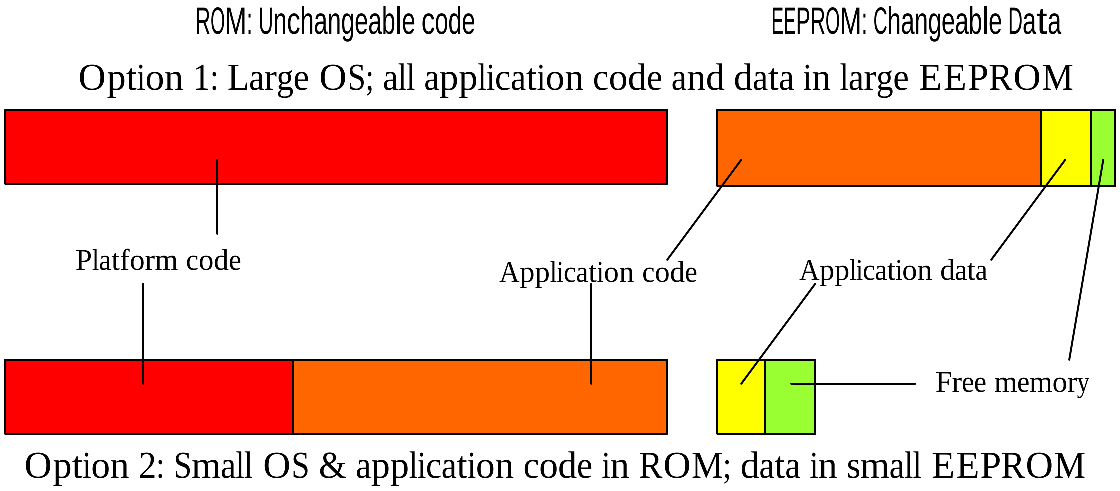
Sample Calculation: 5 applets with a combined (non-rewritable code) size of 27 kB and a (rewritable) data requirement of 4 kB are to be run on a JavaCard.

Option 1: The card issuer purchases a 32kB EEPROM card, and has the applets (code and data) be loaded into EEPROM, leaving 32kB - 27kB (code) - 4kB (data) = 1kB EEPROM free.

Option 2: The card issuer submits the applets' code for inclusion into a custom ROM mask on an 8kB EEPROM card leaving 8kB – 4kB (data) = 4kB EEPROM free.

Savings of option 2 over option 1:

- A smaller and less expensive card can be used (8k EEPROM vs. 32k EEPROM)
- Card initialization time is reduced (to zero)
- More EEPROM remains free for future applications (4kB vs. 1kB)


ROM: Unchangeable code
EEPROM: Changeable Data
Option 1: Large OS; all application code and data in large EEPROM
Platform code
Application code
Application data
Free memory
Option 2: Small OS & application code in ROM; data in small EEPROM

APPENDIX B: Q&A

In order to address typical questions popping up periodically when reviewing IBM's offerings in the smart card space, the below question and answer section serves as a guide to reconciliation.

Q: Proprietary Card operating systems (COS) are sometimes less expensive than open COSs. How can open COSs add value?

A: A proprietary COS causes significantly higher card-lifetime costs (proprietary application development, proprietary personalization and card handling, more expensive solutions) than an open COS, by far offsetting the lower brand and licensing royalties of the proprietary COS that are due to the involvement of fewer copyright and patent holders as compared to open COSs. Also, proprietary systems are slower to market as the tight coupling between COS and applications does not permit separate development, testing, and approval/certification cycles as possible for open COS-based systems.

Q: Is a proprietary COS faster and functionally richer than an open COS?

A: No; this is only an implementation question: Unlike many competitor's products using the same standards, JCOP has proven in performance benchmarks to be at least as fast as proprietary systems. Functionality is also just a question of implementation capabilities: JCOP is available in versions fully supporting all capabilities sought after in the financial, telecom, and identification market: With proper applications, one JCOP version can replace any proprietary COS incumbent.

Q: Is JCOP currently certified?

A: The three initial versions of JCOP are Visa approved at level 3. The product JCOP21id will attain FIPS-140/2 Level 3 certification by the end of 2002. A Common Criteria EAL4+ certification will be done for the platform in 2003.

REFERENCES

[1] Don Davis: *Battering Down a Barrier to Entry*; Card Technology, August 7, 2002.

[2] M. Baentsch, et.al.: *JavaCard—From Hype to Reality*; IEEE Concurrency, Oct.-Dec. 1999.

[3] Sun Microsystems: *JavaCard 2.1.1* http://java.sun.com/products/javacard

[4] Global Platform Consortium: *OpenPlatform 2.0.1'* http://www.globalplatform.org/

[5] GSM 11.11: *Specification of the Subscriber Identity Module* 8.0.3

[6] 3GPP TS 11.14: *Specification of the SIM Application Toolkit* 8.5.0

[7] 3GPP TS 03.19: *Subscriber Identity Module API for JavaCard* 8.3.0

[8] GSM 03.40: *Technical Realization of the Short Message Service (SMS)* 7.4.0

[9] 3GPP TS 03.48: *Security Mechanisms for SIM Application Toolkit* 8.8.0

[10] WAP-260-WIM-20010712-a: *Wireless Application Protocol, Wireless Identity Module*

[11] WAP-261-WTLS-20010406-a: *Wireless Application Protocol, Wireless Transport Layer Security*