# Impact of Smartcard constraints on DBMS

Laxmi Joshi, Nisreen Alzhrani

**Abstract**— Smartcards are the most secure portable computing device today. The first smartcard was developed by BULL for the French banking system. They have been used successfully in applications involving money, proprietary and personal data (such as banking, healthcare, insurance, etc.). Electronic ticketing in public transportation based on smart cards is gaining momentum worldwide. It is widely recognized that a smart card system can deliver benefits to both passengers and operators, but due to its complexity, implementation can come at a considerable cost. As smartcards get more powerful (with 32 bit CPU and more than 1 MB of stable memory in the next versions) and become multi-application, the need for database management arises. However, smartcards have severe hardware limitations (very slow write, very little RAM, constrained stable memory, no autonomy, etc.) which make traditional database technology irrelevant. The major problem is scaling down database techniques so they perform well under these limitations. In this paper, we give an in-depth analysis of this problem and propose a PicoDBMS solution based on highly compact data structures and query execution without RAM. We show the effectiveness of our techniques through performance evaluation.

**Index Terms**— Smartcard –Applications – PicoDBMS, RAM, DBA.

## 1 INTRODUCTION

Smartcards are the most secure portable computing device today. The first smartcard was developed by Bull for the French banking system in the 80s to significantly reduce the losses associated with magnetic stripe credit card fraud. Since then, smartcards have been used successfully around the world in various applications involving money, proprietary data and personal data (such as banking, pay comparable to RAM but very slow write time (10 ms/word). Following Moore's law for processor and memory capacities, smartcards will get rapidly more powerful. A database is an organized collection of data and the database data collection with DBMS is called a database system. A numerous database systems have been developed. PicoDBMS has been designed to provide an effective solution to secured portable folders on smart cards. This section shows that database components (not limited to PicoDBMS) embedded on secured chips (not limited to smart cards) can be exploited in other important contexts and can open very exciting research perspectives. A smart card is a type of credit card that contains a built-in microprocessor, or central processing unit that handles digital information, and memory provided by an embedded integrated circuit. Smart cards are commonly used for financial transactions and for authentication and identification purposes.


A smart card's microprocessor communicates with external services, or computer hosts, via card reading services such as ATMs and ticket readers. The smart card uses a serial interface and draws power from the card reader. The microprocessor in the card is able to process a limited set of instructions for applications such as cryptography, or converting digital data into an encrypted code for security purposes.

### 2. Related Works

From the early 90's, the need for data management appears in several applications running on small devices, from industrial controllers and cable-television set-top boxes to medical-imaging systems and airplanes flight controls. By considering smart cards as traditional small devices, one could envision using embedded DBMSs or Light DBMSs technology in this environment. However the primary objective of smart cards being security, smart cards has very specific hardware architecture.

Block diagram of the Hitachi H8/3113 smartcard chip

ram :

A recent study proposes specific storage techniques to manage data in flash memory on a smart card. The design is also limited to mono-relation queries and is strongly impacted by the physical characteristics of the target smart card architecture. Indeed, they propose to store the data in NOR type of FLASH memory generally dedicated to store programs as ROM replacement. Since updates in NOR flash memory are very costly (updating a single data induces a large and costly bloc erasure), the techniques are driven by update cost minimization (using dummy records and deleted bits). While this study shows the impact of hardware characteristics on the DBMS internals, it does not comply with the memory constraints of the smart card nor addresses complex query processing, mandatory to extract the authorized part of the data.

**3. Impact on the PicoDBMS architecture**

We now analyze the impact of the smartcard constraints on the PicoDBMS architecture, thus justifying why traditional database techniques, and even lightweight DBMS techniques, are irrelevant. The smartcard's properties and their impact are:

**3.1 Highly Secure**: smartcard's hardware security makes it the ideal storage support for private data. The PicoDBMS must contribute to the data security by providing access right management and a view mechanism that allows complex view definitions (i.e., supporting data composition and aggregation). The PicoDBMS code must not present security holes due to the use of sophisticated algorithms2.

**3.2 Highly portable**: the smartcard is undoubtedly the most portable personal computer (the wallet computer). The data located on the smartcard are thus highly available. They are also highly vulnerable since the smartcard can be lost, stolen or accidentally destroyed. The main consequence is that durability cannot be enforced locally.

**3.3 Limited Storage Resources**: despite the foreseen increase in storage capacity, the smartcard will remain the lightest representative of personal computers for a long time. This means that specific storage models and execution techniques must be devised to minimize the volume of persistent data (i.e., the database) and the memory consumption during execution. In addition, the functionalities of the PicoDBMS must be carefully selected and their implementation must be as light as possible. The lightest the PicoDBMS, the biggest them on board database.

**3.4 Stable storage is main memory**: smartcard stable memory provides the read speed and direct access granularity of a main memory. Thus, a PicoDBMS can be considered as a main memory DBMS (MMDBMS). However the dramatic cost of writes distinguishes a PicoDBMS from a traditional MMDBMS. This impacts the storage and access methods of the PicoDBMS as well as the way transaction atomicity is achieved.

**3.5 Non autonomous**: compared to other computers, the smartcard has no independent power supply, thereby precluding disconnected and asynchronous processing. Thus, all transactions must be completed while the card is inserted in a terminal (unlike PDA, write operations cannot be cached in RAM and reported on stable storage asynchronously).

Databases and database systems have become an essential component of everyday life in modern society. In the course of a day, most of us encounter several activities that involve some interaction with a database. For example, if we go to the bank to deposit or withdraw funds; if we make a hotel or airline reservation; if we access a computerized library catalog to search for a bibliographic item; or if we order a magazine subscription from a publisher, chances are that our activities will involve someone accessing a database. Even purchasing items from a supermarket nowadays in many cases involves an automatic update of the database that keeps the inventory of supermarket items.

**4. Restricting Unauthorized Access**

When multiple users share a database, it is likely that some users will not be authorized to access all information in the database. For example, financial data is often considered confidential, and hence only authorized persons are allowed to access such data. In addition, some users may be permitted only to retrieve data, whereas others are allowed both to retrieve and to update. Hence, the type of access operation—retrieval or update—must also be controlled. Typically, users or user groups are given account numbers protected by passwords, which they can use to gain access to the database. A DBMS should provide a security and authorization subsystem, which the DBA uses to create accounts and to specify account restrictions. The DBMS should then enforce these restrictions automatically. Notice that we can apply similar controls to the DBMS software. For example, only the DBA's staff may be allowed to use certain privileged software, such as the software for creating new accounts. Similarly, parametric users may be allowed to access the database only through the canned transactions developed for their use.

**5. Providing Backup and Recovery**

A DBMS must provide facilities for recovering from hardware or software failures. The backup and recovery subsystem of the DBMS is responsible for recovery. For example, if the computer system fails in the middle of a complex update program, the recovery subsystem is responsible for making sure that the database is restored to the state it was in before the program started executing. Alternatively, the recovery subsystem could ensure that the program is resumed from the point at which it was interrupted so that its full effect is recorded in the database.

**6. Flexibility**

It may be necessary to change the structure of a database as requirements change. For example, a new user group may emerge that needs information not currently in the database. In response, it may be necessary to add a file to the database or to extend the data elements in an existing file. Modern DBMSs allow certain types of changes to the structure of the database without affecting the stored data and the existing application programs.

**7. Availability of Up-to-Date Information**

A DBMS makes the database available to all users. As soon as one user's update is applied to the database, all other users can immediately see this update. This availability of up-to-date information is

essential for many transaction-processing applications, such as reservation systems or banking databases, and it is made possible by the concurrency control and recovery subsystems of a DBMS

## Conclusion

As smartcards become more and more versatile, multiapplications and powerful, the need for database techniques arises. However, smartcards have severe hardware limitations which make traditional database technology irrelevant. The major problem is scaling down database techniques so they perform well under these limitations. In this paper, we addressed this problem and finally, we proposed query processing techniques which handle complex query plans with no RAM consumption. This is achieved by considering extreme right-deep trees which can pipeline all operators of the plan including aggregates. We measured the performance of our execution model with an implementation of our query engine on two old-fashion computers which we configured to be similar to forthcoming smartcard architectures. We showed that the resulting performance matches the smartcard application's requirements. This work is done in the context of a new project with Bull Smart Cards and Terminals. The next step is to implement our PicoDBMS on Bull's smartcard new technology, called Over Soft, and to assess its functionality and performance on real world applications.

[1] L. Bouganim, O. Kapitskaia, P. Valduriez. MemoryAdaptive Scheduling for Large Query Execution. Int. Conf. on Information and Knowledge Management (CIKM), 1998.

[2] C. Bobineau, L. Bouganim, P. Pucheral, P. Valduriez. PicoDBMS: Scaling down Database Techniques for the Smartcard. PRiSM Technical Report n°2000/05, 2000.

[3] Blythe, P. T. 2004. Improving public transport ticketing through smart cards. Municipal Engineer (1) 157: 47-54.

[4] Boardman, A. E., D. H. Greenberg, A. R. Vining, and D. L. Weimer. 2006. Cost-Benefit Analysis. Concepts and Practice. New Jersey: Pearson Prentice Hall.

[5]Date, C. [1983] An Introduction to Database Systems, Vol. 2, Addison-Wesley, 1983.

[6] Fernandez, E., summers, R., and Wood, C. [1981] Database Security and Integrity, AddisonWesley, 1981.

[7] Chen, Z. 2000. Java Card Technology for Smart Cards: Architecture and Programmer's guide. Addison-Wesley 1.