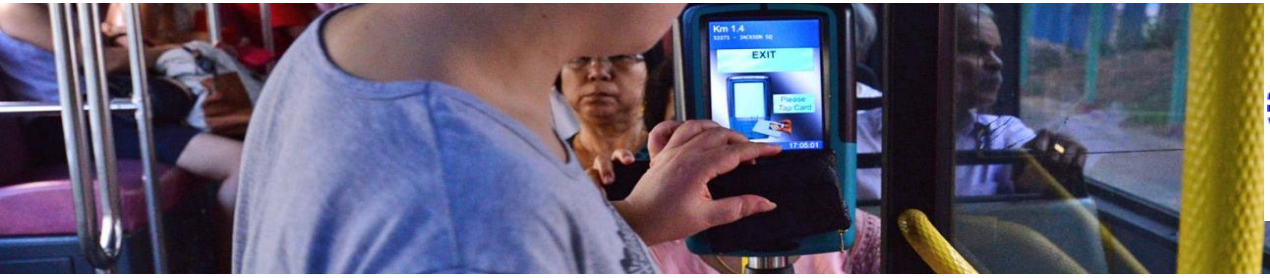


Planeta Virtualization Platform is a simple way to add a CIPURSE CARD into an existing implementation.

✓ OBJECTIVE:

- This article intends to present a simple, secure and scalable solution to help system integrators and/or public transport operators to add ***CIPURSE*** into existing implementation without redesign your solution and keeping the same memory mapping; that means the same ***APPLICATION*** software but with an enhanced security and specially with the possibilities to increase incrementally the security as well as the possibility at the latest stage to have a **SAM CENTRIC** like solution. This is where the SAM can verify the operations and can deny non-allowed operations. In this article, the ***WORD LEGACY CARD*** will be linked mainly to ***MIFARE CLASSIC*** technology, despite the solution can be applied to other legacy technologies like **Calypso** and **MIFARE ULTRA LITE/C**.

✓ CURRENT STATUS:

The big issue with a public transportation project is the way the technology was introduced in the market in the past 2 decades.
There was only 1 cost-effective technology that dominates the market, the leading companies provided many reference designs, APIs, methodologies and etc.
The MIFARE technology was considered by the customers as a secure; and that supposition was confirmed by all vendors along the time, despite the fact specialist had a different opinion.
MIFARE was released and due to back compatibility remains the same all the time, let us explore a basic description of MIFARE CLASSIC:

- *ISO14443A 106kb/s = GOOD.*
- *3 pass mutual authentication= GOOD*
- *48 bits secret key = BAD, indeed very bad.*
- *Obscure crypto engine (or proprietary) = BAD*
- *Confidential set of commands, we just send a reader command based on a READER API provide by the vendor, no information about internal details = BAD.*
- *Proprietary set of commands, non T=CL protocol=BAD.*

Anyway the MIFARE CLASSIC technology dominates the market as the main contactless card for PT applications for decades until specialized people from Universities and research centers start to publish and reveal the it's weakness. At certain time even the technology owner published few articles giving some hints to enhance the security of MIFARE implementations.
Indeed with the best technics is not possible to reach higher levels of security using MIFARE. Some hints already implemented by PLANETA since 2003 are:

- *Use of unidirectional counters to help avoid card content cloning*
- *Use of keys stored in the READER to help improve card authenticity verification (just 1 sector).*
- *Use of DATA signature using keys outside the CARD for another level of security*
- *Use of key derivation based on secure algorithms like CMAC using AES128.*
- *Use of manufacturer identification to detect clone cards and emulators*
- *Use of speed validation*
- *Adding of black lists*
- *Adoption of a SAM CENTRIC solution, allowing to the SAM to do the balance calculations*

✓ The BIG OBSTACLES:

Many customers looked to an alternative products like MIFARE PLUS, DESFIRE, CALYPSO and now CIPURSE, since it is an OPEN STANDARD and has few different chip manufacturers.
Almost all of the customers faced the below barriers to implement new technologies:

- *The design was made bottom up, that means the customer designed a MIFARE MEMORY MAPPING then try to fit the PT application.*
- *The equipment has no free memory to add a big software to completely manage a new implementation based on CIPURSE.*
- *Since the API for MIFARE has direct commands to manage the CARD, sometimes, the available API don't have a function for T=CL cards.*
- *Since MIFARE and CIPURSE are completely different products, the customer considers a completely new design, but they are afraid about that, they are afraid about big changes.*
- *The good news is the work made to promote MIFARE PLUS as an alternative to MIFARE enabled the availability of a RAW ISO14443A function, since PLUS use it but not using a T=CL just sending ISO CL frames. This function is useful for T=CL implementation as part of adaptation library.*

✓ Trying to find a solution:

At the end, the majority of the current implementations are based on a MIFARE MEMORY MAPPING, the definitions are like below defined :

- DIRECTORY
  - **Normally BLOCK 1,2.**
- USER_ID_INFO:
  - **Few blocks normally in the same SECTOR, READING and WRITING protected. 3 blocks are normally enough, no back up, since the data is not updated in field.**
- LOG_TRANSACTIONS_INFO:
  - **Few blocks (3) normally in the same directory, READING and WRITING protected, 3 blocks are normally enough, another copy to implement a BACK-UP.**
- PURSE_INFO:
  - **Few blocks also in the same SECTOR and with BACK-UP.**
- VALUE_BLOCK:
  - **Normally requires 2 blocks for unicity control (when exist) but you need to allocate 3 blocks in the same SECTOR to allow RESTORE and TRANSFER between back-up.**

Since the application requires many different fields yet the memory capacity of a MIFARE CARD is not big, the data fields normally use bit fields then the allocation of each data field is not based on bytes offsets and sizes but in bits offsets and sizes.

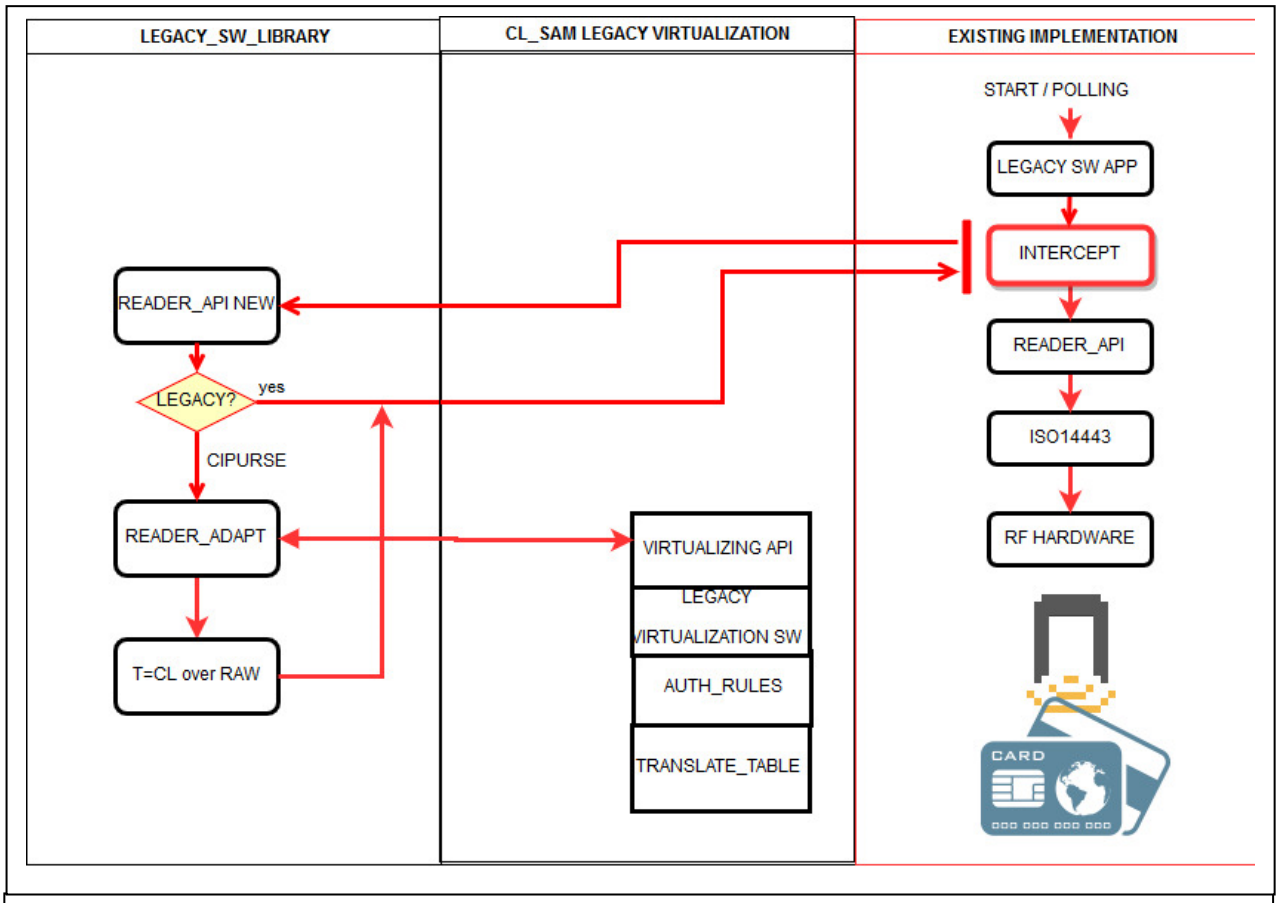


FIG 1: General SW implementation view using a VIRTUALIZATION LIBRARY by SW..

You will find fields like (just to mention few):

- Balance field = 20 bits size
- CTC (card transaction counter) = 14 bits size
- LTC (load Transaction counter)= 14 bits size
- CARD_BLOCK_FLAG= 1 bit size
- LAST_BUS_ID= 14bits.
- Issuing date= year,month,day= 7,4,5 bits.
- Issuing ID= 30 bits.
- Etc..

✓ The BASIC IDEA:

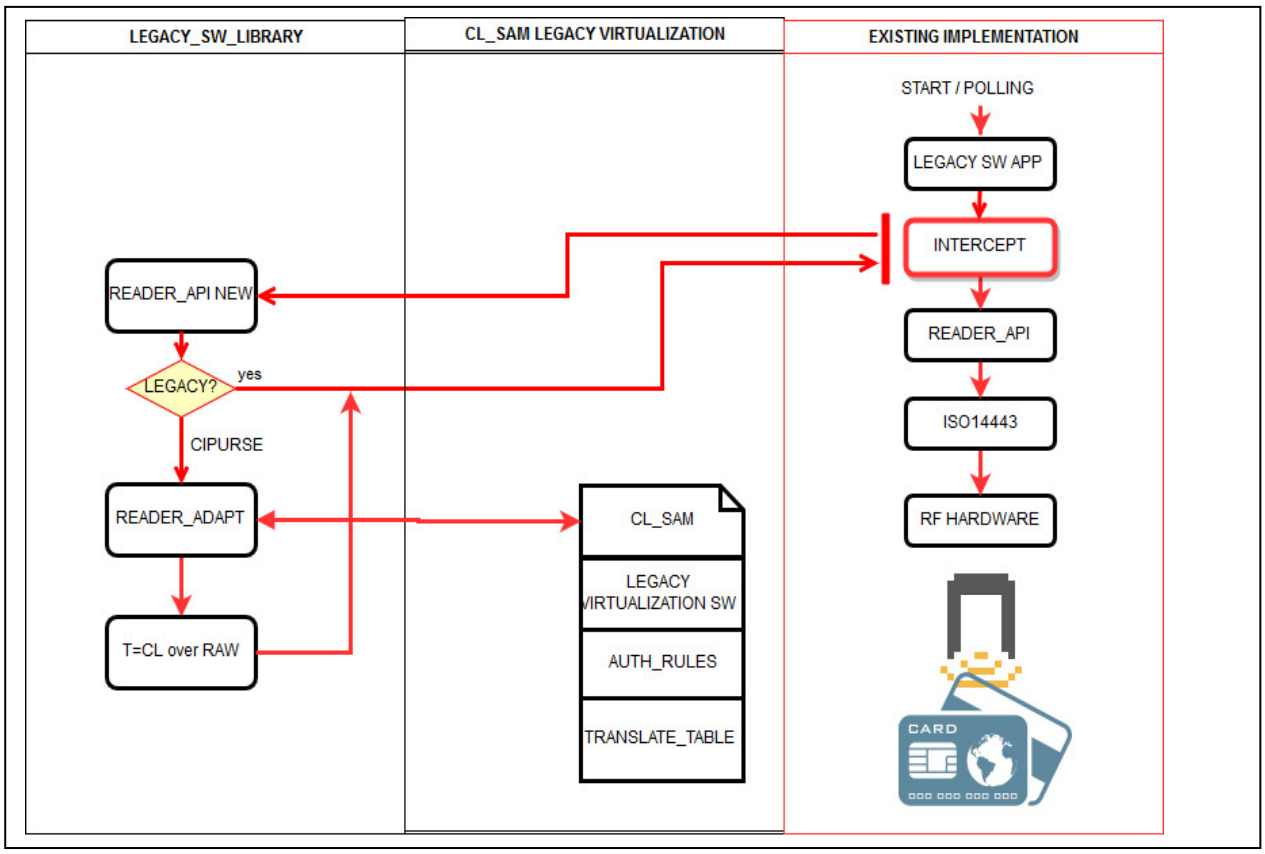
Based on the above information we are providing one solution to keep the ***CUSTOMER APPLICATION*** exactly the ***SAME, NO CHANGES***, adding ***CIPURSE*** by a ***LEGACY MEMORY MAPPING VIRTUALIZATION*** through the ***ORIGINAL MIFARE API*** intercepted by our ***LIBRARY*** and using a ***VIRTUALIZING*** Library by ***SOFTWARE*** or based on an ***ADVANCED SAM*** (CL_SAM which run Virtualizing Library) and can protect the ***MUTUAL AUTHENTICATION, TRANSLATE TABLES*** and ***KEY DERIVATION*** based on ***LEGACY (MIFARE) KEYS***, anyway the original card KEYS are not used as KEYS but ***as DIVERSIFICATION DATA*** to derive the ***KEY*** for each ***CIPURSE FILE*** which is MAPPED to hold a set of ***LEGACY MEMORY BLOCKS***. The operations are:

- We intercept the READER API, specifically the commands related to LEGACAY CARD operations (PRESENCE, LOAD_KEY,AUTHENTICATE,READ,WRITE,INC,DEC,TRANSFER,RESTORE), this is made

in C language just by DEFINES, just including a .h file with a MACRO REDEFINITION for the API changing the API MIF_FUCNTION() by PVP_MIF_FUCNTION().

- Then we add a SOURCE CODE to be compiled together with the original SW with the NEW interface and the CALL_BACK of the original function. The selection is made based on the last CARD DETECTED, if the CARD is MIFARE we just call back the same original function, if the card is CIPURSE we build a APDU and send it to a VIRTUALIZING LIBRARY (see fig 1) or to the SAM (executing the Library internally see fig 2).
- The VIRTUALIZING LIBRARY internally fetch the data block from a TRANSLATE_TABLE where we have a list of correspondence like:
    - MIFARE_BLOCK_NUMBER=> MODE, KEY_NUMBER, CIPURSE_SFID, CIPURSE_OFSET. We can define what DATA BLOCK inside a CIPURSE FILE SYSTEM correspond to a MIFARE BLOCK. The MODE defines the AUTHENTICATION ENHANCED METHOD.
    - The default AUTHENTICATION METHOD translates the MIFARE MIF_AUTHENTICATION into a CIPURSE MUTUAL_AUTHENTICATION using the MIFARE KEY as a DIVERSIFICATION DATA to derive the AES_KEY. We can do some optimizations not doing again a MUTUAL_AUTHENTICATION when we have a SECURE CHANNEL ON and the SECTORS READ use the SAME MIFARE KEY. The result will be a better performance.
    - By configuration is it possible to force a NEW AUTHENTICATION CYCLE for each FILE accessed defined by a TRANSLATION TABLE linking a BLCOK NUMBER and KEYSET/RULE to a CIPURSE FILE + OFFSET / AUTHENTICATION rule.
    - When the INTERPCEPT code receives the LEGACY CARD COMMAND it translate to a generic set of COMMANDS in a APDU format and submit to the VIRTUALIZING LIBRARY, the command then is executed leading to a possible sequence of operations like MUTUAL AUTHENTICATION, READ DATA at an OFSET or RECORD number , WRITE DATA at an OFSET/RECORD number, etc.. The result will be returned to the function call only when the VIRTUALIZING function is completed.
    - In terms of security we have the following increments in relation to the original LEGACY CARD:
        - The LEGACY CARD keys exchanged in clear with from the APP until the READER and through the API is only used as a DIVERSIFICATION DATA not as a KEY, allowing to link the APP with it, but protecting it against spying, considering the derivation is done securely like SAM based.
        - The LEGACY CARD is not protected against CARD EMULATORS because the READ after WRITE to verify the UPDATE is not secure, then an emulator can provide the DATA in a READ after WRITE cycle tricking the APP. In the VIRTUALING LIBRARY all READS are MACed then we always do the READ VERIFICATION to proof the READ is legitimate NEW, then a READ after WRITE can provide a proof of a CARD UPDATE.
        - The MASTER KEYS for CIPURSE ADFs are never passed in clear in the API.
        - Using an ADVANCED SAM applying a SAM CENTRIC methods will be possible to manage and control some forbidden operations like incrementing balance without a secure authorization or unlock the card or changing back the card transaction counter etc.
    - The solution can help in areas such as a shared VIRTUALIZATION, where multiple LEGACY CARDS can be managed by a single CIPURSE card with different ADF for each LEGACY VIRTUALIZATION.
    - The files created to VIRTUALIZE a LEGACY CARD MEMORY BLOCK can be:
        - LINEAR FILE with N x 16 bytes size to virtualize an N LEGACY MEMORY blocks, these blocks don’t need to be contiguous in the LEGACY MEMORY. We can put all blocks that uses the same KEY or a group of MEMORY blocks, allowing for an optional optimization during READ and WRITE.

- RECORD FILE with n records of 16bytes to virtualize n blocks.
- RECORD VALUE FILE with n x 12 bytes to virtualize n blocks of MIFARE VALUE BLOCK. The translation between CIPURSE 12 bytes block to LEGACY CARD 16bytes VALUE BLOCK is made automatically by the VIRTUALIZATION LIBRARY.



- FIG 2: General SW implementation view using a VIRTUALIZATION LIBRARY Managed By an ADVANCED SAM (CL_SAM).

NOTE: ***All CIPURSE files will be protected by a SM_RULES and depending on the PROFILE we will be able to activate ENCed communication as mandatory, otherwise we can use MACed communication. The security rules can be modeled by a toll.***

- The VIRTUALIZATION commands managed by VIRTULAIZATION LIBRARY are:
  - LOAD_KEY(KEY_ID[1], KEY_MODE[1], KEY_DATA[6]) : Stores the KEY internally.
  - AUTHENTICATION(KEY_ID[1],KEY_MODE[1],KEY_DATA[6})
  - READ_BLOCKS(BLOCK_NUM[1],SIZE_READ)
  - WRITE_BLOCK((BLOCK_NUM[1],DATA[nx16],SIZE_WRITE)
  - INC_BLOCK(BLOCK_NUM[1],VALUE[4])
  - DEC_BLOCK(BLOCK_NUM[1],VALUE[4])
  - RESTORE_BLOCK(BLOCK_NUM_O[1],BLOCK_NUM_D[1])
  - TRANSFER_BLCOK(BLOCK_NUM_O[1],BLOCK_NUM_D[1])
- When the VIRTUALIZATION LIBRARY find an ADVANCED SAM (CL_SAM) it will send the COMMAND to it and manage the sequence of responses with a 91XX status, which informs the need to get a command from the SAM and send to CIPURSE until the answer is not 91XX, completing the command and responding to the API function.

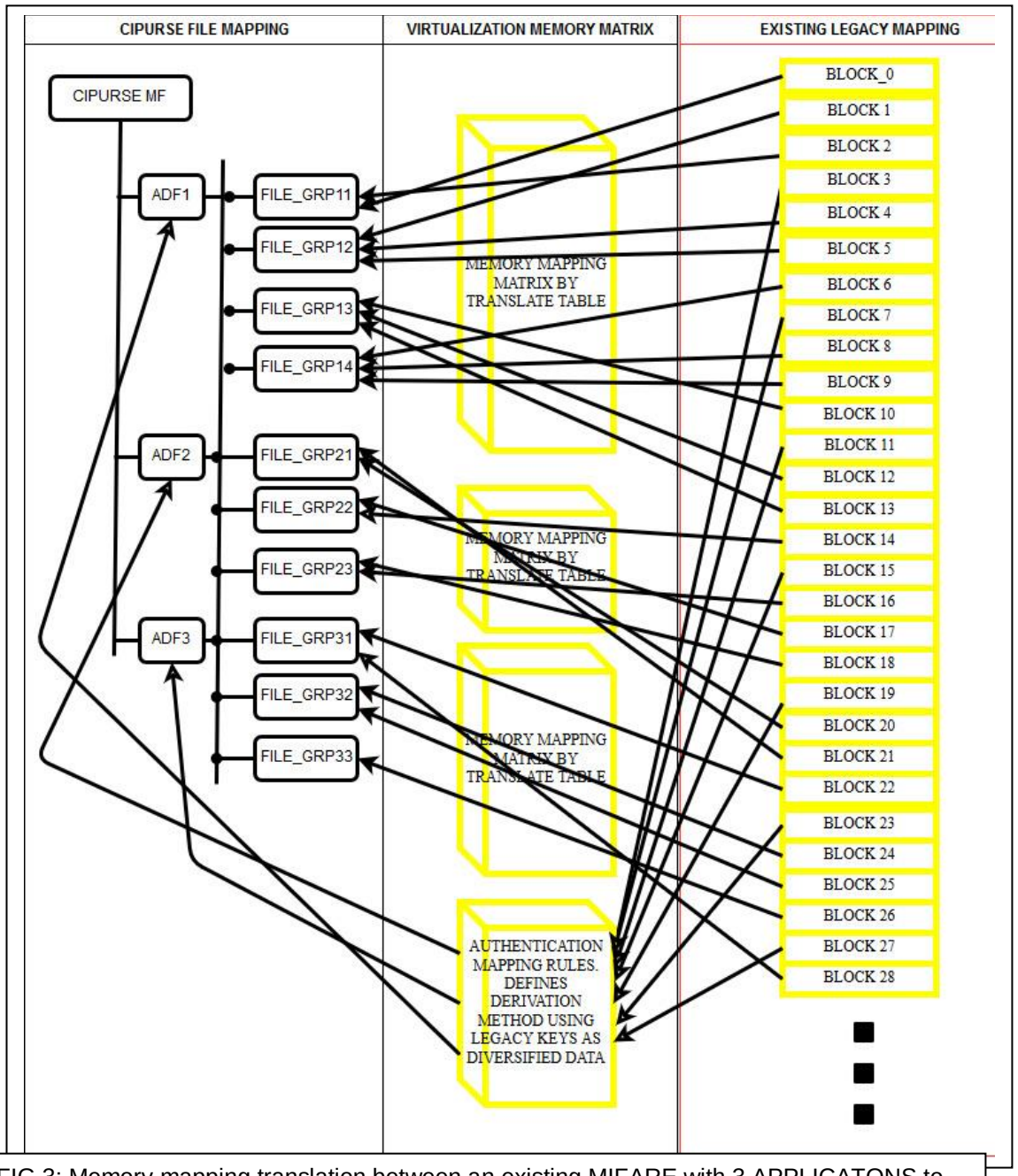


FIG 3: Memory mapping translation between an existing MIFARE with 3 APPLICATONS to an organized 3 ADFs with the same memory mapping.

✓ Conclusions and advantages:

- The existing LEGACY APPLICATION will continue exactly the SAME, no changes then NO RISK.
- We ADD CIPURSE to the current implementation as an additional option.
- Using CIPURSE we can solve all the weakness related to MIFARE technology almost immediate, like:
  - **Authenticity check, now we can do a SECURE MUTUAL ATHENTICATION using AES128 with a strong derivation method. I can use the original MIFARE KEY not as key but as a DIVERSIFIED DATA, then we can link the CIPURSE DERIVED KEY to the LEGACY original WEAK KEY, now used as DATA not as KEY.**
  - **We may solve the problem related to UID size, translating 4,7,10 PHYSICAL to a 4,7,10 LOGICAL. We can have a field inside the CIPURSE card to store a LOGICAL UID to be used as a physical in the current system.**
  - **We can protect the CARD against content cloning.**
  - **We can protect the CARD against CARD CLONING.**

    - **We can add incrementally security over DATA FIELDS by using an ADVANCED SAM like CL_SAM with a MIRROR protection and a SAM CENTRIC implementation. We can start without this kind of protection and add protection until all data FIELDS are protected and the implementation will become SAM CENTRIC.**
    - **We can add easily a new application, for example we can add MOBILE OFF_LINE HCE implementation.**
    - **New implementation for ABT (account Based Ticketing) can be added with few commands just relying on SAM security and can be based on a completely new design, for example a credential with ECC SIGNATURE verification.**
- *We may use this implementation for many different applications not only for **Public Transportation**, like*:
    - Mobile Payment
    - Vending Machines
    - Loyalty Programs
    - Private Label
    - Payment Applications

PVP (*Planeta Virtualization Platform*)

The PVP solution transforms legacy terminals, such as found in public transport and payment applications, into cutting edge acquiring nodes.

The final result is capable of handling cards and mobile secure transaction for the complete spectrum of available technologies, such as legacy and new proprietary brands (e.g. MiFARE) as well as recent open standards (e.g. NFC, HCE, and CIPURSE).

The PVP solution achieves virtualization through two components:

- **An advanced SAM acting as a distributed trusted execution environment and running the VIRTUALIZATION LIBRARY inside;**
- **VIRTUALIZING LIBRARY with 2 modules:**
  - **VIRTUALIZING LIBRARY by SW module**
  - **VIRTUALIZING LIBRARY based on an ADVANCED SAM implementation, calling the SAM and managing all VIRTUALIZATION (memory translation / authentication linkage) inside the SAM automatically, if the SAM is present and correctly configured the SAM will be used otherwise if the SW is ready and configured with a correct files it can be used.**

The TERMINAL Intercept and API emulation to call VIRTUALING Library.

All code are provided as source code, allowing the customer to adapt any LEGACY implementation, thus avoiding big reinvestments.

✓ **CONTENT:**

- **Supply of PVP source code in C for the TERMINAL.**
- **Supply of a SAM with preloaded PVP implementation**
- **Development of an adaptation layer to run PVP in the embedded hardware of choice and providing:**
  - **Definition of the LEGACY READER API**
  - **Definition of a RAW or T=CL interface (or the HW INFO)**
  - **Definition of a LEGACY MEMORY MAPPING providing:**
    - **LIST of SECTOR / BLOCKS**
    - **LIST of SECTORS KEY ATTRIBUTES for each block or the CONFIG BLOCK content.**
    - **NOTE1: We don't need to have all the information about MIFARE MEMORY MAPPING definition, just the blocks in use and desirable the combination of the blocks, in other words the group of blocks that belongs to the same DATA structure.**
    - **NOTE2: We don't need to have the MIFARE KYES, we just need to know the SECTORS that use the same KEYs and the ACCESS CONDITIONS. The customer will generate it our NEW CIPURSE M ASTER KEY for the project and that key will be derived using CMAC algorithm and should provide in the personalization toll the MIFARE keys to be used as DIVERSIFIED DATA.**
- **The LIBRARY can support a management of UID SIZE allowing 4bytes, 7 bytes and 10 bytes PHYSICAL UID with a translation to 4,7 or 10 bytes LOGICAL UID.**
- **Support random UID.**
- **Support NFC emulated by mobile phones (Host Card Emulation) for both online and offline acceptance.**

✓ ADVANTAGES:

- The adaptation of the secure technology(CIPURSE, MiFARE,) will be done exactly as the be customized as per the requirement of the client, thus providing the same DATA modeling and just INCREASING the SECURITY and PERFORMANCE.

- Does not require a new development of the APP SW but support new secure technologies (e.g. side-by-side support of a same APP for both MiFARE and CIPURSE; thus allowing flexible procurement and migration strategies.

- All codes provide for LIBRARY implementation are provided as a source code, allowing easy adaptation for new HW and or APIs. The VIRTUALIZING Library has only about 2.500 LOC (lines of code) and use less then 10Kbytes of RAM + 4K STORAGE.

- The risk for MIGRATION / ADITION of CIPURSE is ZERO, since the original APP is not changed.

- The regular repeated AUTHENTICATION for different SECTORS in LEGACY cards can be avoided using an optimized way where if the KEY is the SAME the AUTHENTICATION is skipped, or even if the DATA BLOCK belongs to the same DATA GROUP defined in the config.

- The same method can be adapted for other card types, then Cards like DESFIRE, NTAG, ULTRA LITE/C, MIFARE PLUS, PICO PASS, CRYPTO, RF, etc…, can all coexist without conflict.

- The implementation cost is made of the cost of SAMs, which are anyhow considered mandatory for SECURE solution. The SAM may be a SAM CENTRIC or NOT with obviously different results in terms of security.

- The READ after WRITE is SAFE, meaning when the APP verify matching of the READ after WRITE the card was UPDATED due to the internal check of the READING REPSONSE with MAC, then is not possible to trick the answer.

- Depending on the CIPURSE profile we can chose different SECURE CHANNEL methods, but the minimum is CMD= PLAIN / ANSWER=MACed.

✓ ADITIONAL ADVANTAGES DUE TO ADOPTION OF AN ADVANCED SAM:

- The implementation based on an ADVANCED SAM can also use:
  - Pre-encripted DATA, allowing to have another level of security.
  - Sign DATA structure, by adding an extra 8 bytes to each FILE holding an SIGNATURE generated and updated by the SAM.
  - Requires less than 400 LOC (lines of code) added to the current SOFTWARE to implement the VIRTUALIZATION, since the VIRTUALIZING Library is inside the SAM.

- Despite the presence of a SAM between the API and the CARD/Mobile, the performance is enhanced compared with a LEGACY implementation.

- Thanks to the SAM distributed Trusted Execution Environment features, controlling the operation of CARDS security is strongly enhanced, and the acceptance of HCE Tokens in offline mode is enabled.

- Eventually the MIFARE implementations can see their security increased because of the additional security mechanisms enabled by the Distributed Execution Environment features of the SAM.
- The ADOPTION of an ADVANCED SAM can enable additional security over the CARD DATA, protecting DATA fields like BALANCE, CTC, LTC, LOCK, etc.