ABC4TRUST

Attribute-based Credentials for Trust

Smart cards and how they fit into ABC4Trust

Reference Group – 1st Meeting

Rueschlikon, 2012-02-13/14

Jakob I. Pagter (jakob.i.pagter@alexandra.dk)
Head of Research and Innovation, Security Lab
The Alexandra Institute Ltd, Denmark
www.alexandra.dk

Slides based on input from Pascal Paillier, CryptoExperts

A research project funded by the European Commission's 7th Framework Programme

SEVENTH FRAMEWORK PROGRAMME

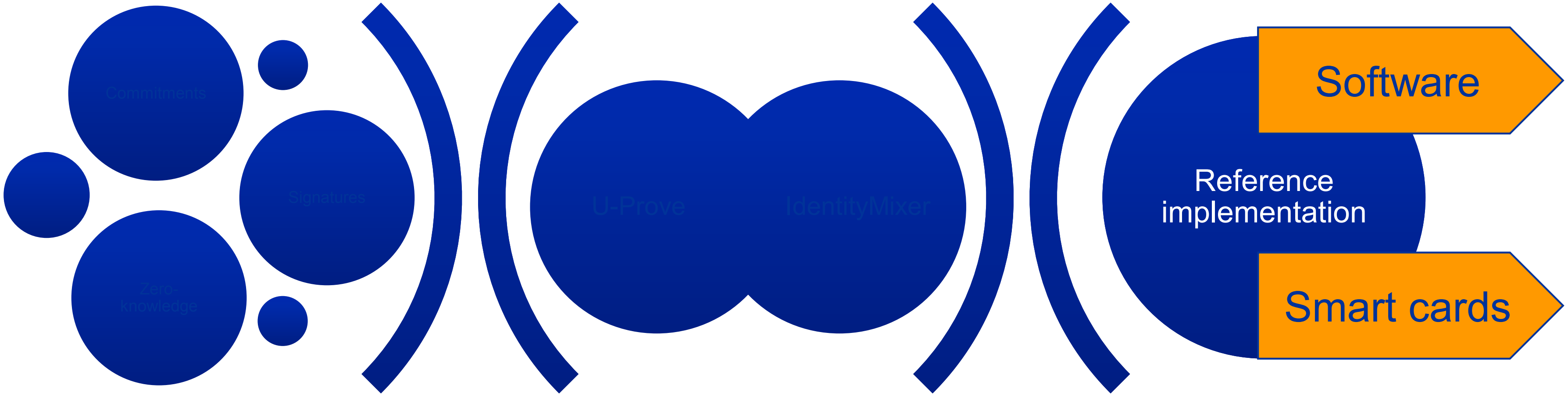
High-level goals

WP3 – comparison

- Bridge “chasm” between theoretical and practical security

WP4 – reference implementation

- Provide reference components for practical use (pilots)



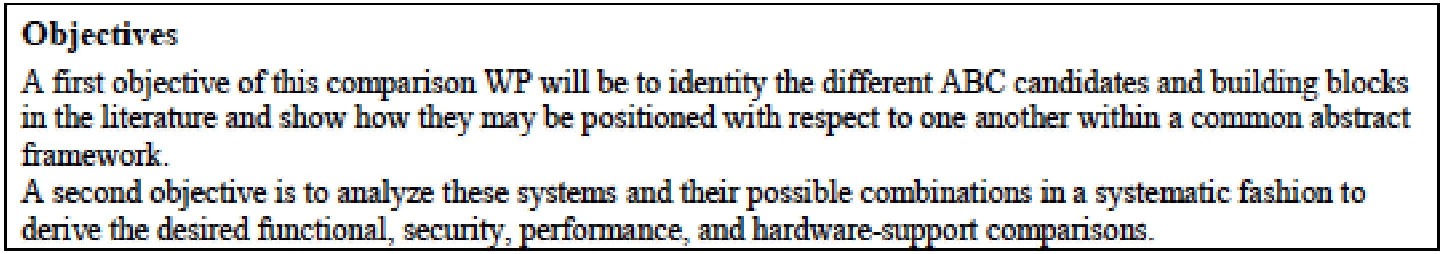
Theoretical crypto

ABC systems

Systems architecture

Comparison – objectives

- Common abstract framework for ABC technologies
- Systematic analysis
  - Functionality
  - Security
  - Performance
  - Hardware-support



Objectives

A first objective of this comparison WP will be to identity the different ABC candidates and building blocks in the literature and show how they may be positioned with respect to one another within a common abstract framework.

A second objective is to analyze these systems and their possible combinations in a systematic fashion to derive the desired functional, security, performance, and hardware-support comparisons.

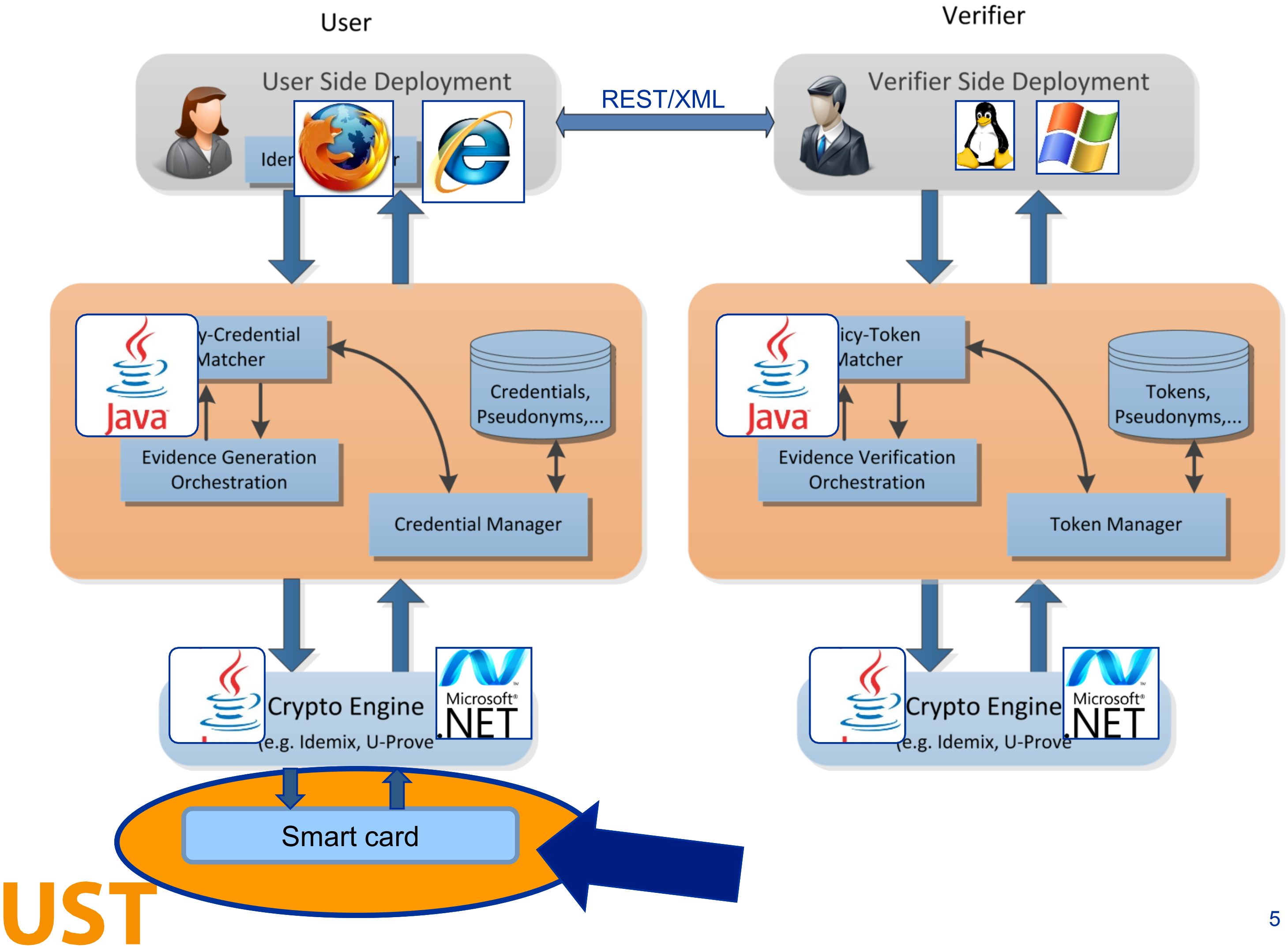
Reference implementation – objectives

- Provide a reference platform
  - Demonstrate the federated architecture (WP2)
  - Support any ABC system
  - Simple application case included to ease adoption and deployment (anonymous hotel booking)
  - Easily adaptable to create pilots
  - 2 versions:
    - Purely software version
    - Hybrid, hardware assisted version (smart cards integrated)
- Inputs from WP2/3, interactions with WP5/6/7

Reference implementation: overview

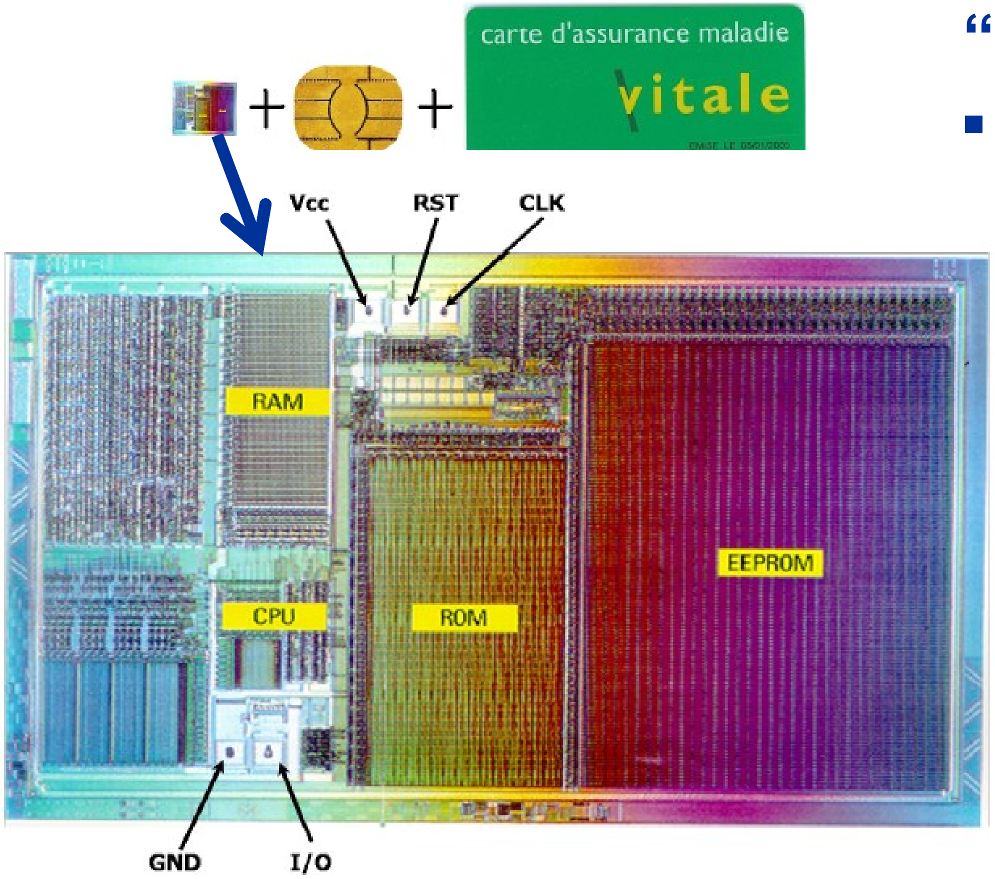
Also:

- Issuer
- RA
- Inspector



ABC4TRUST

Smart cards



“Smart” = chip on-board

- CPU a.k.a MCU (microcontroller unit)
  - 8-bit CISC architectures (Intel 8051, Motorola 68HC05, etc)
  - 16-bit e.g. ARM CalmRisc
  - 32-bit RISCs (SmartMips, Sparc, ARM SCxxx, etc)

Memory

- RAM (a few kilobytes), working memory
- ROM (cheap and dense, 256 KB), operating system
- EEPROM, Flash (non-volative memory, up to 1GB), static application code or data storage
- EPROM (fuses, small amount), irreversible card state

Control registers to address specific hardware modules e.g. TRNG, clock generator (3.57 MHz), timers, serial IO interface (9600-230000 bit/s)

Cryptoprocessors ([3]DES, AES, modular arithmetics for RSA)

- 25 mm2 max

Smart card privacy

- No low-level identifiers need to be revealed

Smart card security

Attack vectors

- Bad or poor crypto
- OS flaws and bugs
- Fault-induced file reading
- Invasive attacks
- Aggressive attacks

Protection

- Intrusion detection
- User behavior
- State-of-the-art engineering

Smart cards - programming

“Open cards”

- Download your own applet in card
  - JavaCard ≠ Java, see www.javacard.org
  - Full .Net implementation
  - Davlik (VM Android) also virtualized recently
- Pros
  - Reusable across smart card products
  - Standard APIs
  - Standard development methods e.g. Javacard
- Cons
  - No access to hardware and crypto processing
  - Increased code size
  - Lack of performances in contactless products

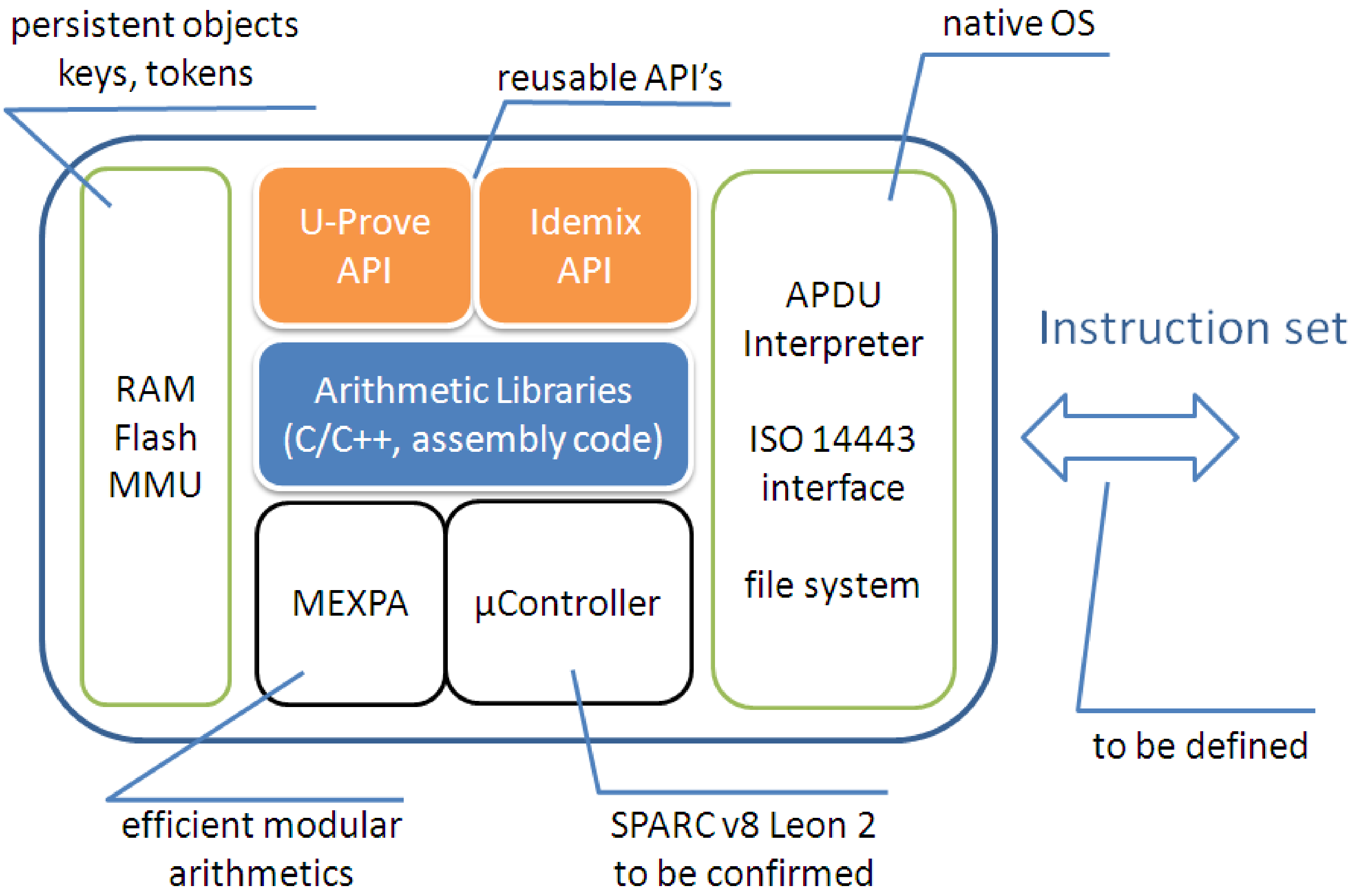
Dedicated cards

- You have to program your own
  - file system and memory management
  - fixed set of APDUs (read/write/exeles, access to crypto APIs)
  - but you may rely on ISO standardized, reusable guidelines and software
- Pros
  - dedicated and closed product but complete control of what you are doing
  - more efficient using native code (C/C++ and assembly for entire OS)
- Cons
  - hardly extensible
  - access emulators, tools and docs from chip manufacturers prone to bugs unless advanced validation methods
  - high level of expertise

Smart cards in ABC4Trust

- We want to show that ABCs are practical on smart cards
- We selected a contactless smart card chip with crypto processor
- We found that, using pre-computations (coupons):
  - U-Prove can be made efficient (issuance and presentation)
  - Idemix can be made efficient for issuance (less clear for presentation)
- Specification and development of the ABC4Trust card are now underway
- CryptoExperts working on implementation for pilots

ABC4Trust smart card architecture



Comparison – smart cards

- Mathematica-based simulation of the issuance and presentation phases of ECC-based U-Prove
- Conducting a feasibility study for Idemix and estimated performances for the issuing phase and the presentation phase
  - Next step: Mathematica simulation
- Comparing different low-level implementation strategies

Comparison – smart card numbers

- Boost U-Prove issuance
  - Using pre-computed coupons
  - Optimized partitioning of exponentiations
  - Estimations for 33MHz Leon 2 + MEXPA
    - 259ms to issue a token
    - 39ms x #{undisclosed attributes} + 50ms
  - Issues
    - When to compute coupons
    - Select specific curves to speed-up group operations
- Idemix: issuance ~231ms with coupons
- Continue hardware benchmarking
  - Continue and improve understanding of U-Prove and Identity Mixer (Attempt to construct for security proof of U-Prove)
  - Choosing curves (ECC) for U-Prove

Conclusion

- Focus on comparing and integrating different technologies
- Smart cards as user trusted module
  - Based on dedicated approach
- Reference implementation on mainstream platforms

Thank you!

Open questions…

- Should smart cards (and application) support both U-Prove and Idemix in a concurrent manner or must there be cards supporting U-Prove and cards supporting Idemix?
- Is implementing a third ABC system necessary to demonstrate that the architecture is interoperable/technology-agnostic?