New White Card Schemes and Java Card™ Technology

Eric Vétillard

CTO
Trusted Labs
www.trusted-labs.com

TS-3814

White Card Opportunities

Where this talk is going

Learn about the new opportunities related to white card schemes, and how application developers can seize them

Agenda

The White Card Concept
Case Studies: 2 New Architectures
Responsibility and Trust
Constraints for Developers
Practical Impact on an Application
Interfacing with Java ME
Opportunities: What? Where? When?

Agenda

The White Card Concept

Case Studies: 2 New Architectures

Responsibility and Trust

Constraints for Developers

Practical Impact on an Application

Interfacing with Java ME

Opportunities: What? Where? When?

White Card Scheme

- A white card is…
  - …a card with no default application
  - …a card with no issuer
  - …owned by the cardholder
- In a traditional card scheme, a card...
  - …comes preloaded with at least one application
  - …is associated to a card issuer
  - …is owned by its issuer

The Ultimate Flexibility

The promise of Java Card technology

- Proprietary cards
  - One issuer, one application provider
  - Typically, one main application, possibly two or three
- City cards
  - One issuer, several application providers
  - A few applications, often related to public service
- White cards
  - No issuer, several application providers
  - Many possible applications of different kinds

Why It Hasn't Worked

- Money: Who pays for the card?
  - The cardholder won't spend any money
  - Why spend money without owning the card?
- Responsibility: Who is responsible for the card?
  - Who will bring guarantees to the application providers?
- Marketing: Whose logo is on the card?
  - Why do you have 50 cards in your wallet?
  - The single sign-on card remains a dream

What Has Changed Recently

- Somebody is willing to pay for the card
  - The card is embedded in another device
  - You get it “free” when you buy this device
- Some people don’t need their logo on the card
  - In particular, public service providers
  - For them, card issuance is a cost, and no advertising
- Certification technologies are available
  - Card certification schemes are converging
  - Automated application certification is available

What We Will See Next

- How some issues have been addressed
- Which issues remain to be addressed
- How to take advantage of it

Agenda

The White Card Concept

Case Studies: 2 New Architectures

Responsibility and Trust

Constraints for Developers

Practical Impact on an Application

Interfacing with Java ME

Opportunities: What? Where? When?

Embedding Smart Cards in Devices

- Useful as a security element
  - For instance, to protect content
  - Or just to hold a few sensitive data
    - Authentication data
    - Certificates and other credentials
- Most useful if it is open
  - Embedding Java Card technology is useful
    - It is even a strong enabler
- Java Card technology is now cheap enough
  - It is available on small chips

From NFC to Smart NFC (1)

Case study #1

- NFC is a standard for contactless communication
  - Targeting in particular mobile phones
  - Allowing phones to be used as contactless cards
  - Originally promoted by Philips, Sony, and Nokia
- Similar technology is already in use
  - Mobile Felica is successful in Japan
    - Used for mobile payment and transport applications
  - Experiments are performed in Europe and USA
    - A NFC experiment (payment, ticketing) in Caen (France)
    - A multi-application NFC experiment in Atlanta's stadium (USA)

From NFC to Smart NFC (2)

Case study #1

- NFC is a simple communication chip
  - The intelligence in in the mobile phone
  - There is a problem for managing sensitive data
    - A secure element of some kind must be used (e.g., a SIM)
  - Chipmakers are adopting it beyond Philips and Sony
    - For instance, Inside Contactless has introduced eNFC
- Smart NFC includes a smart card chip
  - The technology is promoted by Philips
    - NFC communication is proposed on high-end SmartMX

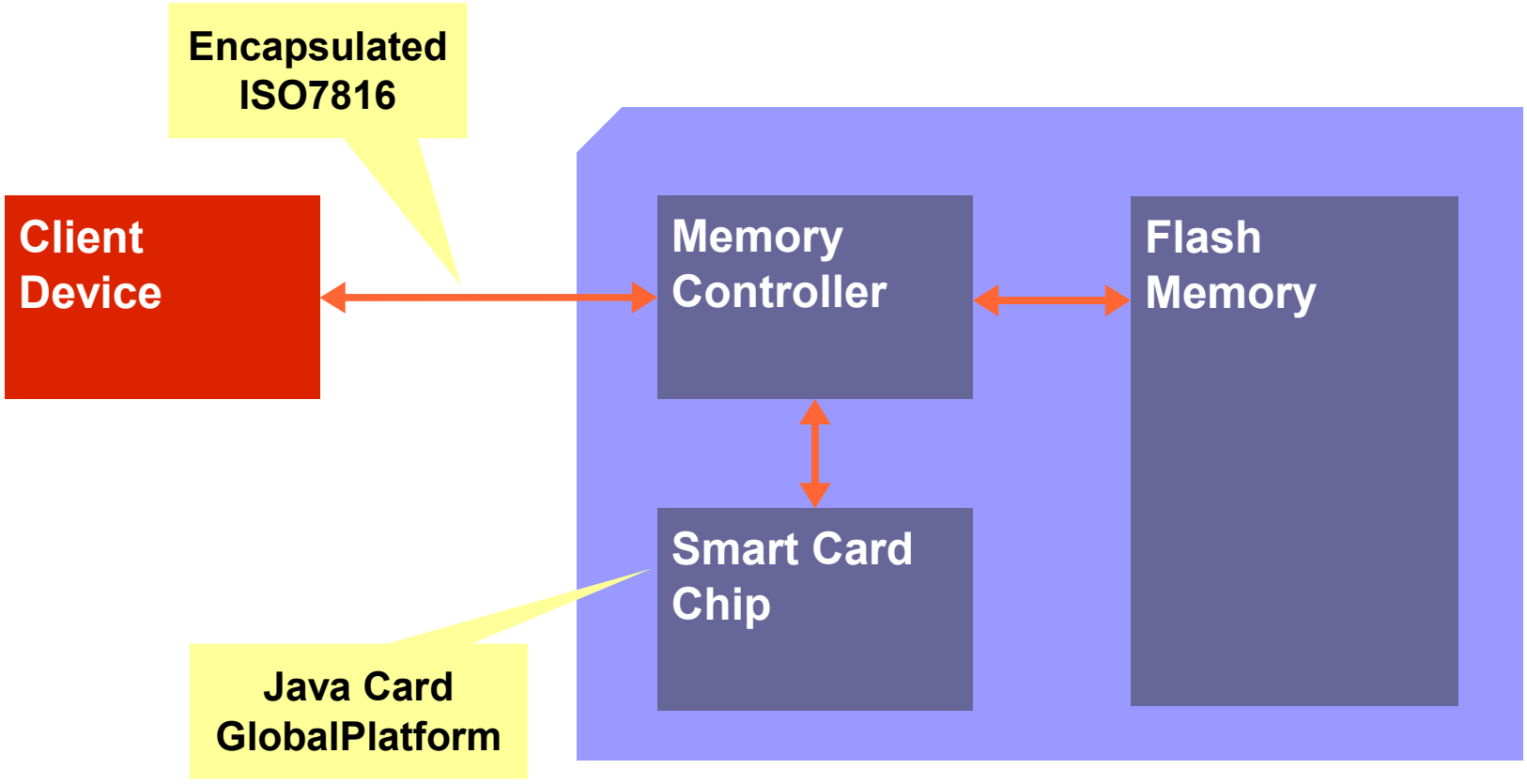
Smart Memory Cards (1)

Case study #2

- Memory cards only include data
  - Very often, sensitive or protected data
  - Most attempts to include security have been flawed
    - Lack of experience in security software and hardware
- New kinds of cards include a smart card chip
  - For instance, the X-Mobile Card from Renesas
  - Companies like SanDisk also consider it
  - In both cases, they include Java Card technology

Smart Memory Cards (2)

Case study #2



Agenda

The White Card Concept

Case Studies: 2 New Architectures

Responsibility and Trust

Constraints for Developers

Practical Impact on an Application

Interfacing with Java ME

Opportunities: What? Where? When?

Responsibility and Trust Issues

Several different layers

- On the card architecture
  - Allow several actors to manage their applications
  - Keep some level of control on the card
- About card and application certification
  - Some certification is required by sensitive applications
  - Who can manage this certification?
- About production and distribution
  - Smart object producers can’t handle sensitive data
  - The initialization of the chip must be minimal
    - In particular, the security requirements must be minimal

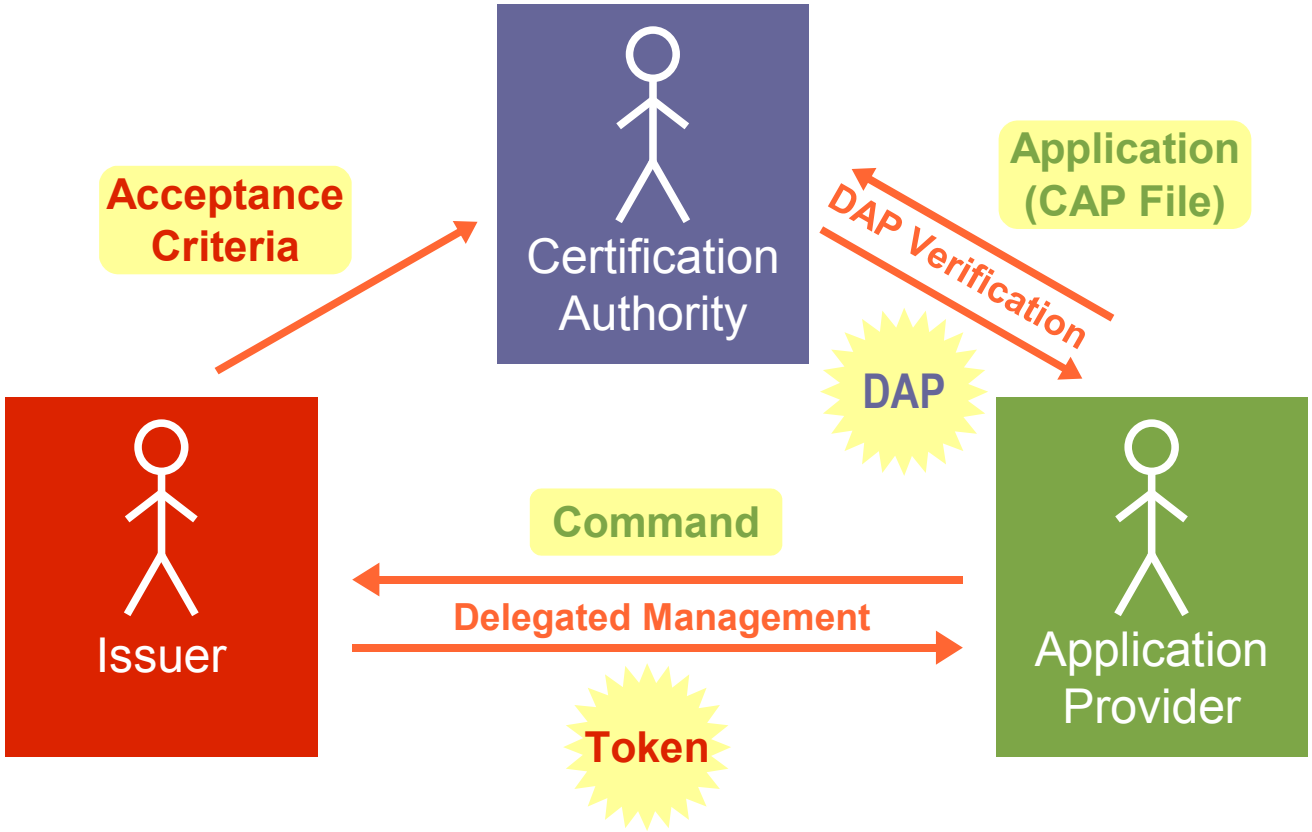
Trust Issues at the Card Level

The GlobalPlatform solution

- GlobalPlatform defines several roles
  - The card issuer is responsible for the card
  - Application providers can be represented
  - Certification authorities as well
- GlobalPlatform defines relationships
  - Application providers have security domains
  - Certification authorities have DAP verification
- Open issue: Replace the issuer

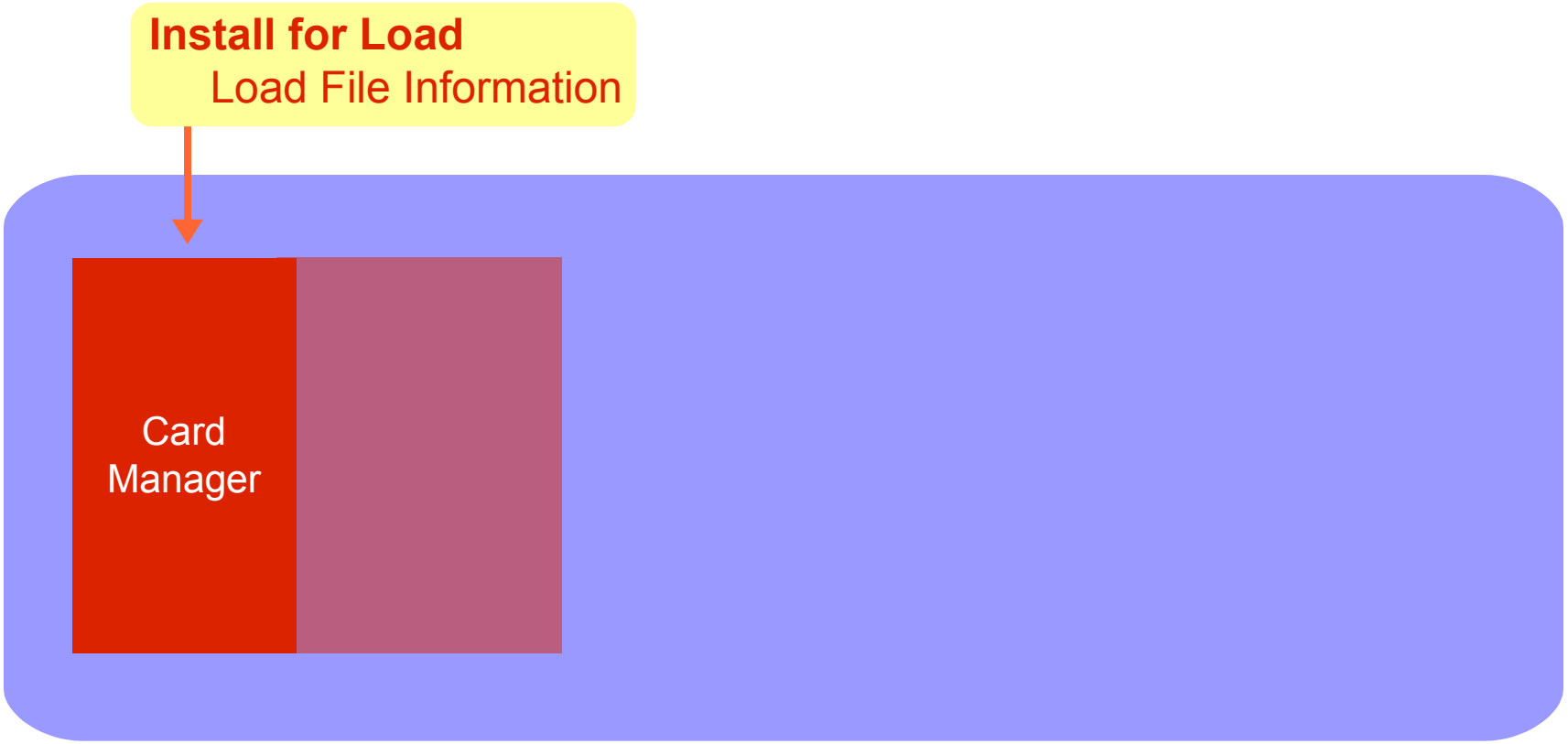
GlobalPlatform Roles

Relationships between actors



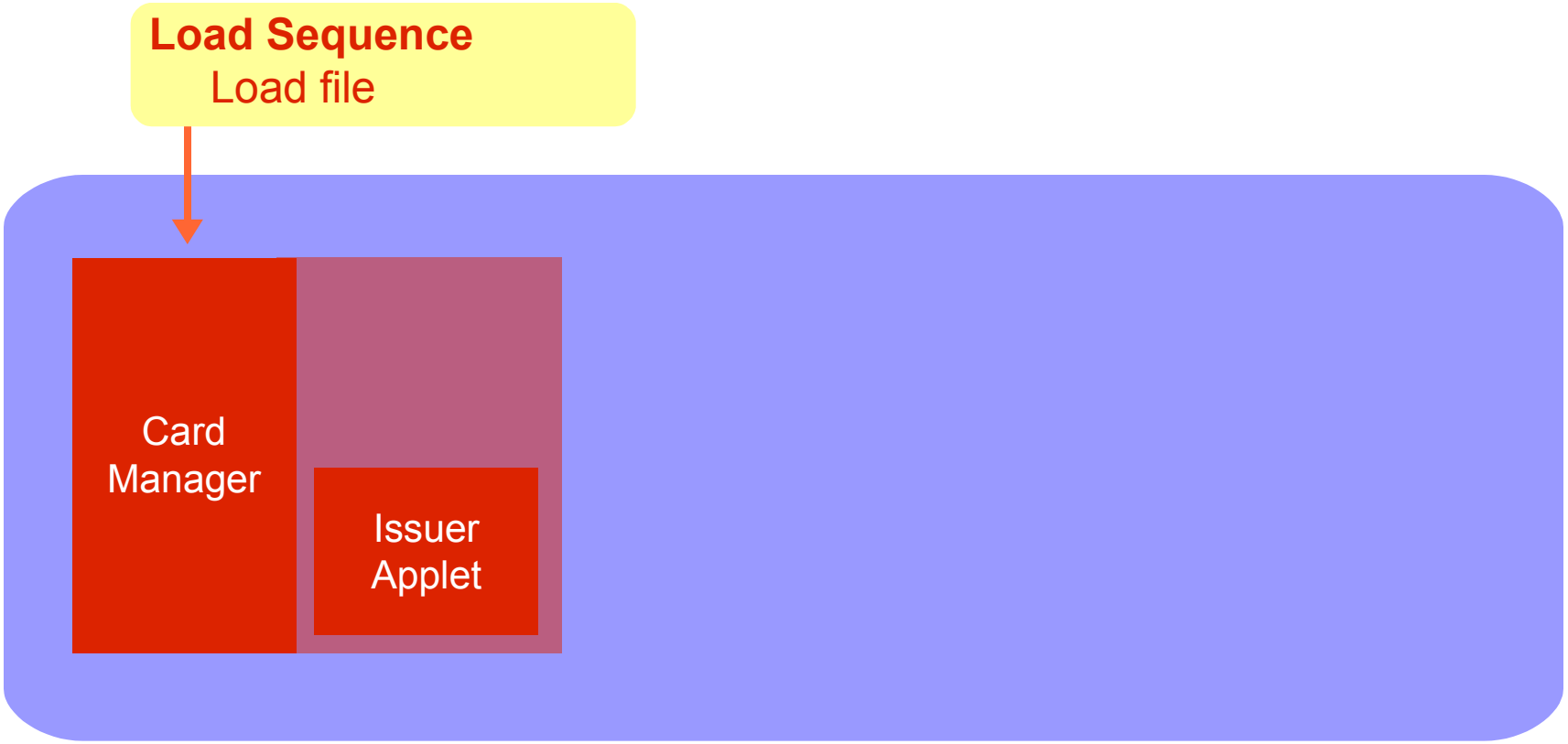
GlobalPlatform Procedures

Loading an issuer application (1/3)



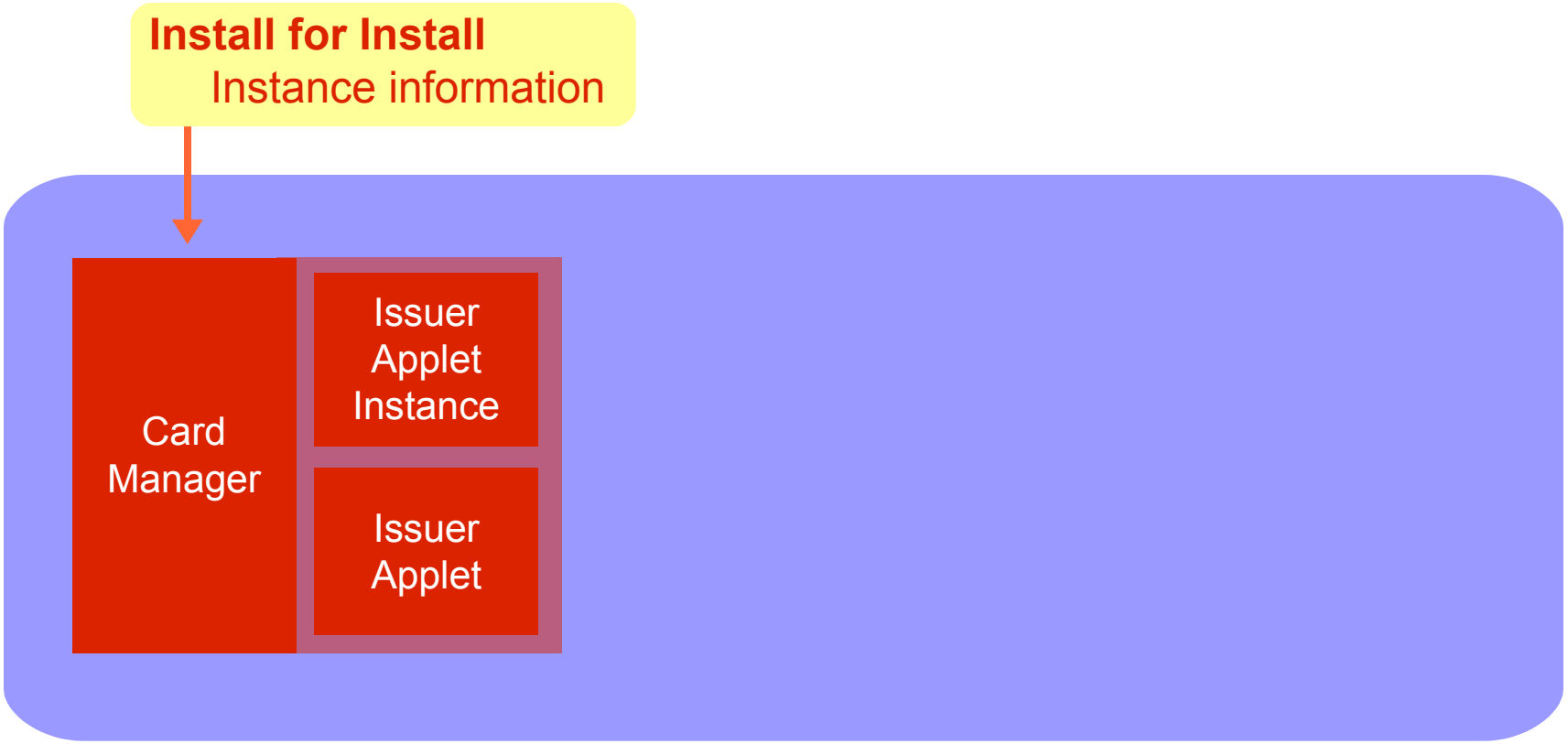
GlobalPlatform Procedures

Loading an issuer application (2/3)



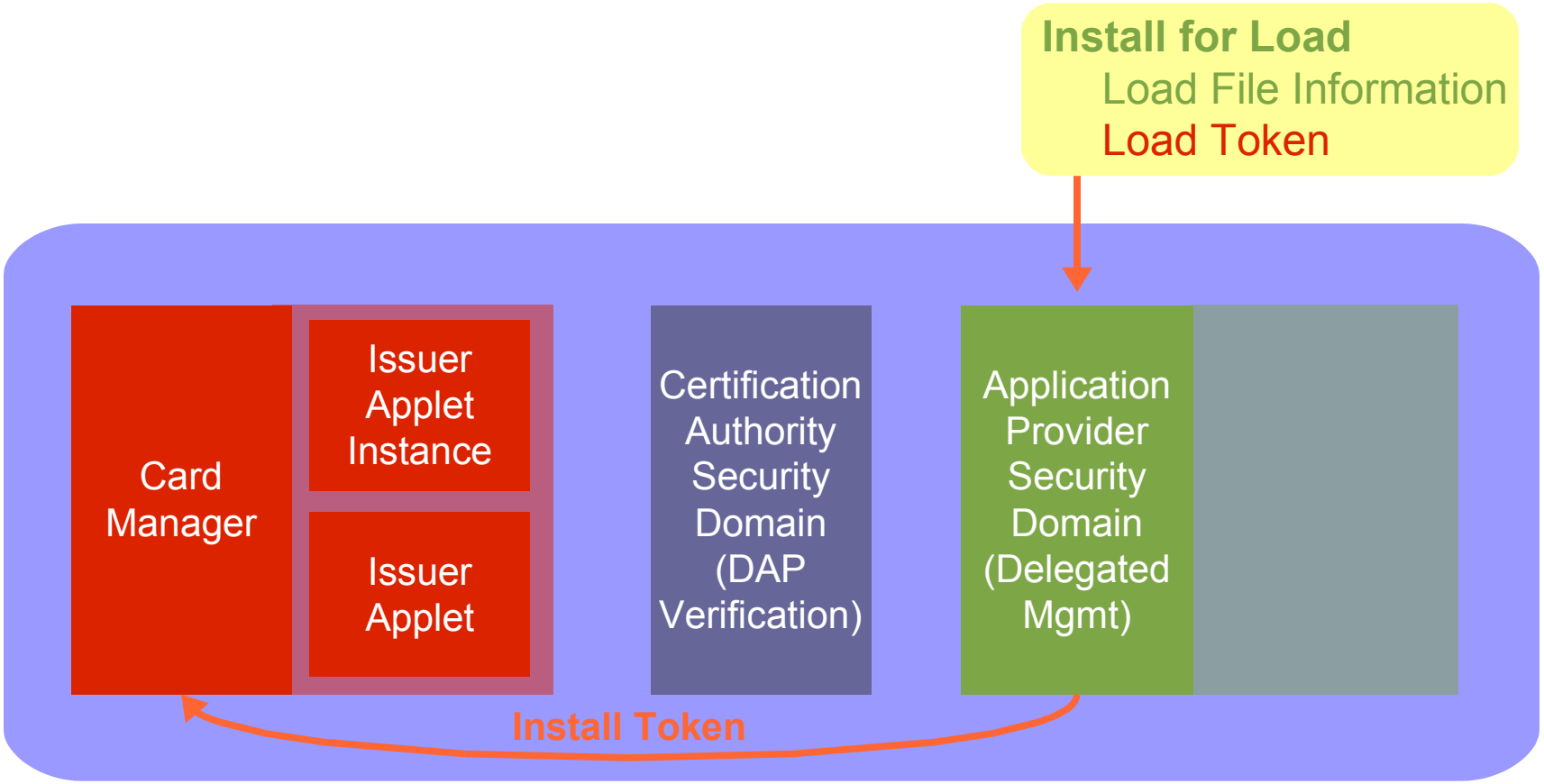
GlobalPlatform Procedures

Loading an issuer application (3/3)



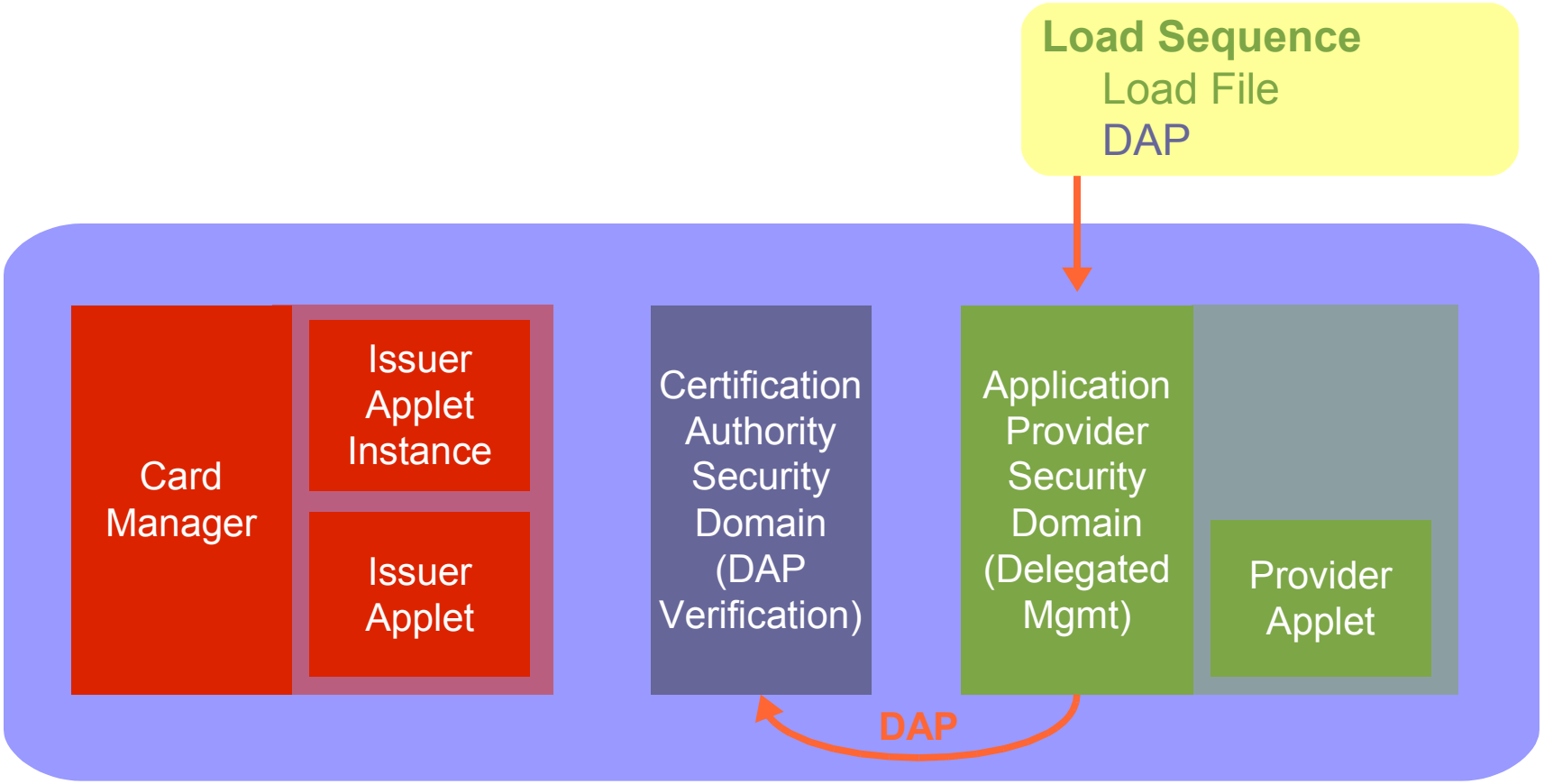
GlobalPlatform Procedures

Loading a third-party application (1/3)



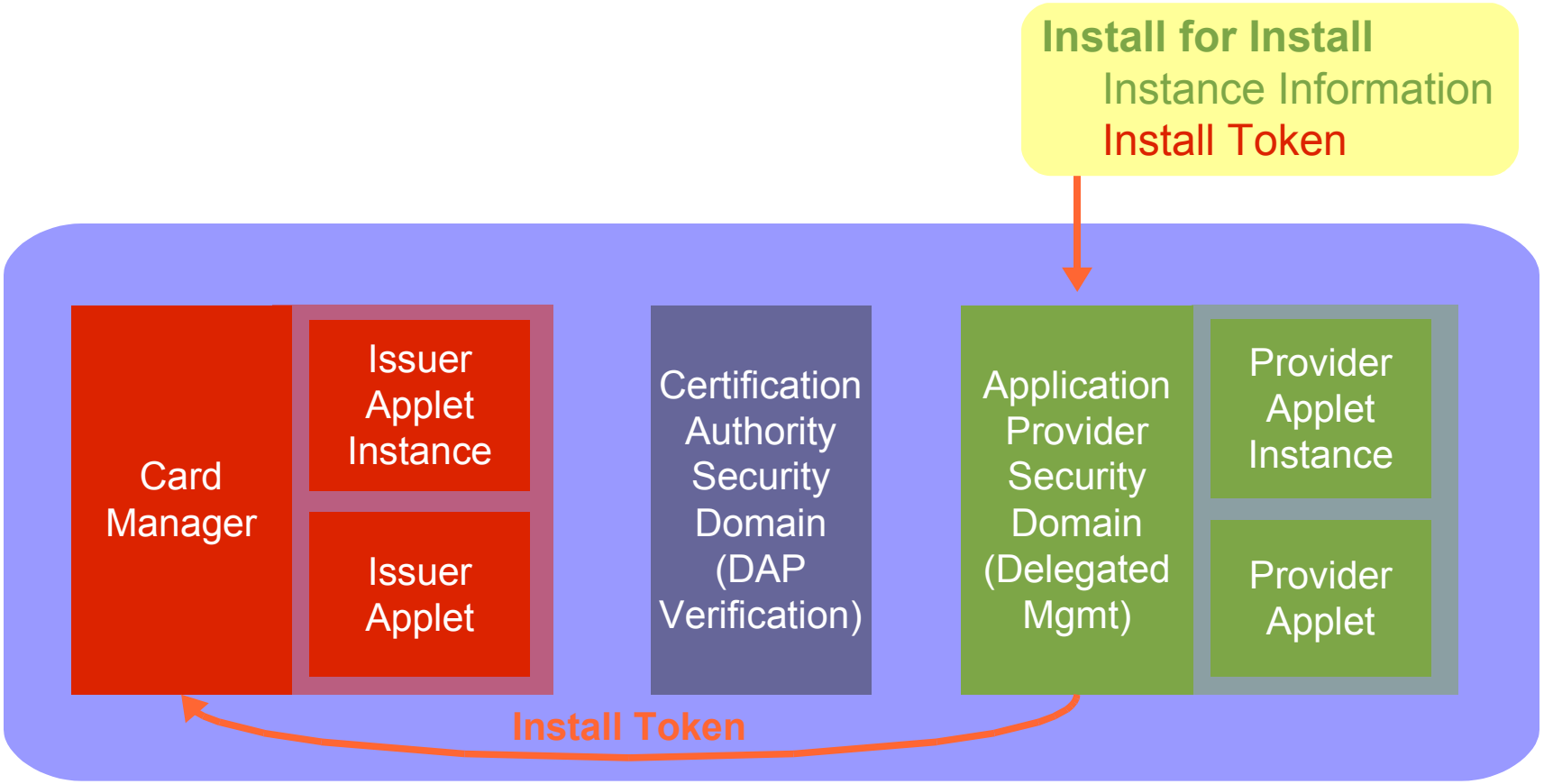
GlobalPlatform Procedures

Loading a third-party application (2/3)



GlobalPlatform Procedures

Loading a third-party application (3/3)



Trust Issues at the Application Level

Card and application certification

- Cards and application can be certified
  - By mandating external certifications (CC?)
  - By using specific procedures and evaluations
    - OK for applications, difficult for cards because of cost
- Criteria needs to be defined
  - Criteria for the card (for security and interoperability)
  - Criteria for applications (for security and portability)
- Open issue: What level of criteria
  - Depends of the target applications
  - For instance, Pay-TV is much stronger than GSM

Card Certification

Why certify a card?

- Guaranteeing interoperability
  - By providing a complete specification
    - Listing all required features
  - By enforcing interoperability guidelines
    - In particular for proprietary specifications
- Guaranteeing a level of security
  - Level required by an industry
    - Low for GSM — Always on-line
    - Medium for banking, transport — Often on-line
    - High for Pay-TV — Never on-line
  - For white cards, most likely medium-level security

Card Certification

How to certify a card?

- Reusing other certification results
  - The card has a Common Criteria certificate
    - With all the desired properties
  - The card has been certified by Visa, MasterCard
    - OK if their process suits your specific needs
- Using your own certification
  - Could be efficient for interoperability
    - Develop a test suite and methodology
    - Then, have a low-cost certification procedure
  - For security, the prices are quite high
    - Who will assume the costs?

Application Certification

Why certify an application?

- Checking for portability
  - Making sure it uses only available features
    - With all the desired properties
- Checking for security issues
  - Making sure the application contains no malware
    - Checking for its use of resources
    - Checking how it shares data and code with others
  - Verifying that the application is safe for the others
    - Making sure it cannot harm other applications
  - Verifying the security of an application
    - Depends on its specification, may be too costly

Application Certification

How to certify an application?

- Black-box automated certification
  - Suited for portability checks
  - Suited for generic security checks
    - Use of sharing, use of sensitive APIs
  - Inexpensive and highly scalable
- White-box code review
  - Verifying the security of an application
  - Takes into account the specificity of the application
  - Expensive and not scalable

Trust Issues at the Production Level

Producing cards embedded in objects

- Solution 1: The object is a “big smart card”
  - The smart card is initialized in a secure factory
  - The process is traditional, by a card manufacturer
- Solution 2: The object is a standard object
  - The smart card hardware includes additional security
  - The initialization process is very simple
    - For instance, writing a public key in memory
  - The real initialization happens later
    - Over-the-air in the case of phones
    - Over Internet (with a reader) in other cases

Where Are We Today?

Trust remains a major issue

- Java Card technology and GlobalPlatform are commodities
  - Java Card technology is now reaching all markets
  - Issuers start to see the interest of GlobalPlatform
    - Including its more advanced features
- Certification is evolving rapidly
  - EMVCo is replacing Visa and MasterCard
  - Automated application validation is available
- Standardization is not yet there
  - Business models need to clarify first

Agenda

The White Card Concept

Case Studies: 2 New Architectures

Responsibility and Trust

Constraints for Developers

Practical Impact on an Application

Interfacing with Java ME

Opportunities: What? Where? When?

Applications for White Cards

Opportunities for developers

- Open schemes represent an opportunity
  - Any open scheme allow a developer to piggyback a large issuer
  - City cards can significantly lower the cost of issuance
  - White cards can make cards widely available
- Openness comes at a price
  - The target cards are restricted in some ways
  - Developer becomes more responsible

Constraints for Application

White card specificities

- Dynamic application management
  - The application is not the card
  - Be ready for updates and versioning
- Certification
  - Be very cautious in the design of external interfaces
  - Make your application easily provable
- End-user management
  - Be ready for deletion, possibly backup and restoration
  - Be ready to interface with other applications

The Application Is Not the Card

Manage their security separately

- Common confusion in many specs
  - “If the counter overflows, terminate the card”
- Security issues need to be sorted out
  - Some threats apply only to the application
    - The PIN try counter has reached 0
  - Some threats are global to the card
    - There are repeated memory integrity errors
  - Some threats are difficult to sort
    - There is a memory integrity on a crucial application data
- Finally, beware of required privileges
  - Terminating a card requires special privileges

The Application Is Not the Card

Manage their lifecycles separately

- Common confusion in many specs
  - “Populate the data before issuing the card”
- Typical mistakes
  - “Installation is performed in a secure environment”
    - Keys and PINs get default values
  - “The application is installed just after being loaded”
    - What is the process is interrupted?
- “I completely control my application’s lifecycle”
    - What if the card is locked or terminated?
- Most existing specs are incorrect
  - Be careful when you get inspiration

Updates and Versioning

Java Card technology's little problem

- Why updates are important
  - Changing an application without changing the card
  - More applications, more bugs, more updates
- There is no provision for updates
  - In Java Card platform, code cannot be updated
    - Data has to be destroyed first
  - In GlobalPlatform, update is not considered
- The application may include update functions
  - These functions can be used for evolution
  - They are more difficult to use for bug fixing

Programming for Updates

Use Java Card platform and GlobalPlatform

- Use Java Card platform's service framework
  - Services can be shared quite easily
  - Services can be added dynamically
- Use shareable containers
  - This is the most difficult part
  - How to make the sensitive data accessible to updates
- Use GlobalPlatform for security
  - In particular, rely on security domains if possible
    - For the application's secure sessions
    - For loading, installing, and personalizing the updates

Programming for Certification

Be obviously portable and secure

- Only use sensitive features if required
  - Accessing a system feature (or another application's)
    - For instance, terminating the card
- Follow guidelines if there are any
  - Do not try to bend the rules
    - Automated provers simply hate it
  - Make your application easily provable
    - For instance, make sure to remove all dead code
  - Be particular careful with shared data
    - Use of another's application data puts you under scrutiny

Programming for an End-User

Prepare for the worst

- Only applies to real white cards
  - Applications are managed by the user
  - A smart card only has finite resources
- Some applications will have to be deleted
  - A backup/restore feature may be available
    - At the platform level (with some constraints)
    - At the application level (designed in the application)
  - Backup and restore may also be useful for updates

Programming with an End-User

Prepare to share some data

- Some data is global to the card
  - Basic data about the user (name, etc.)
  - Possibly some authentication (global PIN, biometry)
- If you are lucky, you may get useful identification
  - Unique card serial number
  - Unique end user identifier
- This data will be managed by a global application
  - Accessible through a specific sharing mechanism
  - You need to use this mechanism
    - And you need to use it correctly (remember the certification)

Legacy vs. New Applications

A cruel dilemma

- Legacy applications are simple to use
  - Their security has often been assessed
  - The infrastructure often already exists
    - Terminal software, servers, personalization,
  - But they are difficult to adapt to new situations
- New applications are more flexible
  - They can be specifically designed for open systems
  - But they are new
    - Their security is hard to design from scratch
    - The entire infrastructure has to be designed

Agenda

The White Card Concept

Case Studies: 2 New Architectures

Responsibility and Trust

Constraints for Developers

Practical Impact on an Application

Interfacing with Java ME

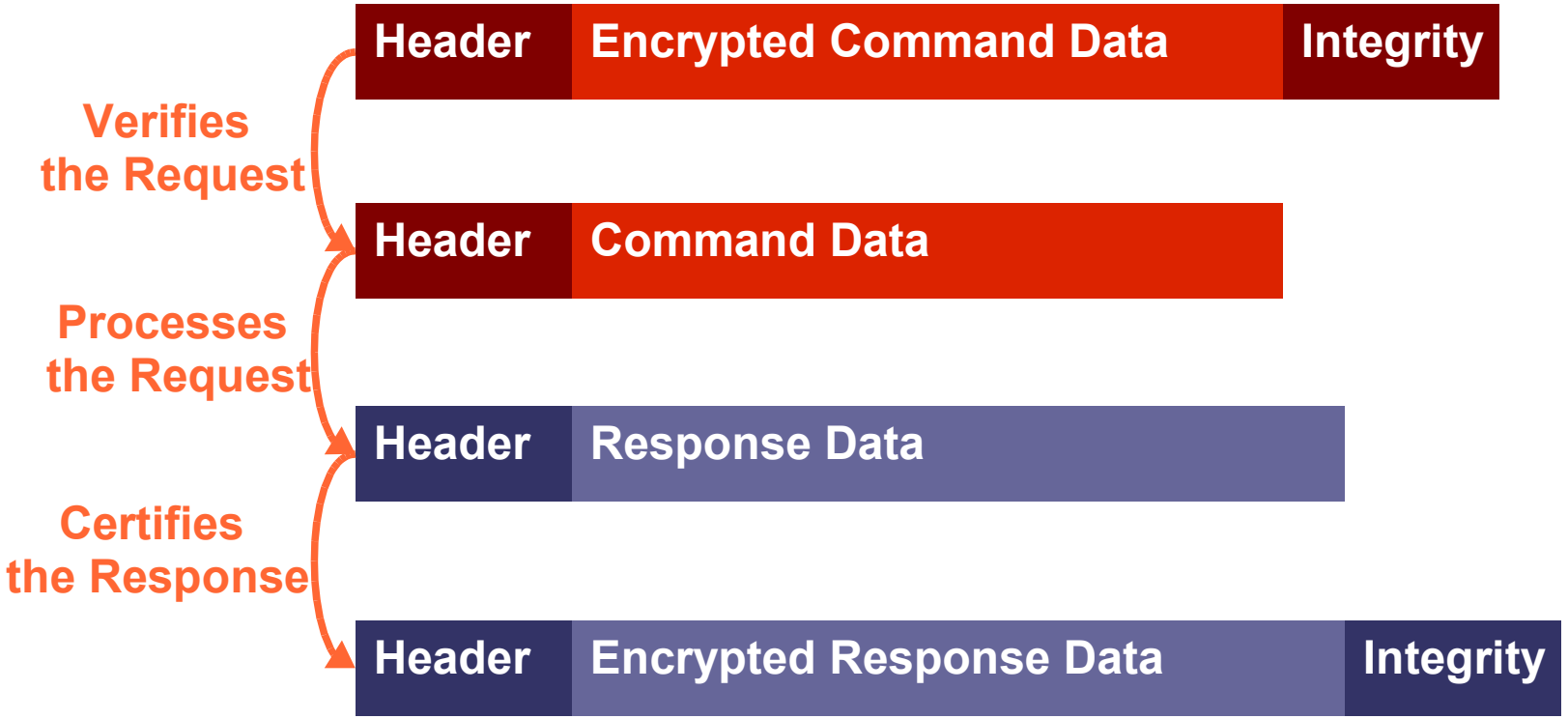
Opportunities: What? Where? When?

Careful Design of an Application

Preparing for open deployments

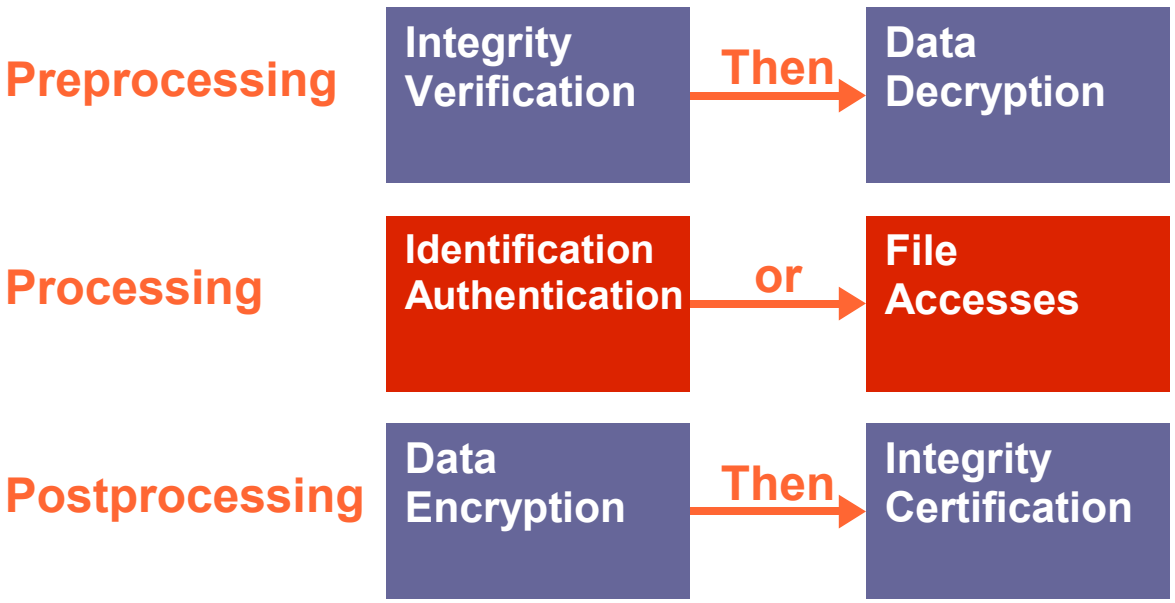
- Step 1: Be ready for updates
  - Design the application to be modifiable
- Step 2: Be ready for backup and restore
  - Allow the application to be temporarily deleted
- Step 3: Include security
  - Make sure the process cannot be abused
- Step 4: Prepare the infrastructure
  - Consider all the other elements

Typical Command Processing



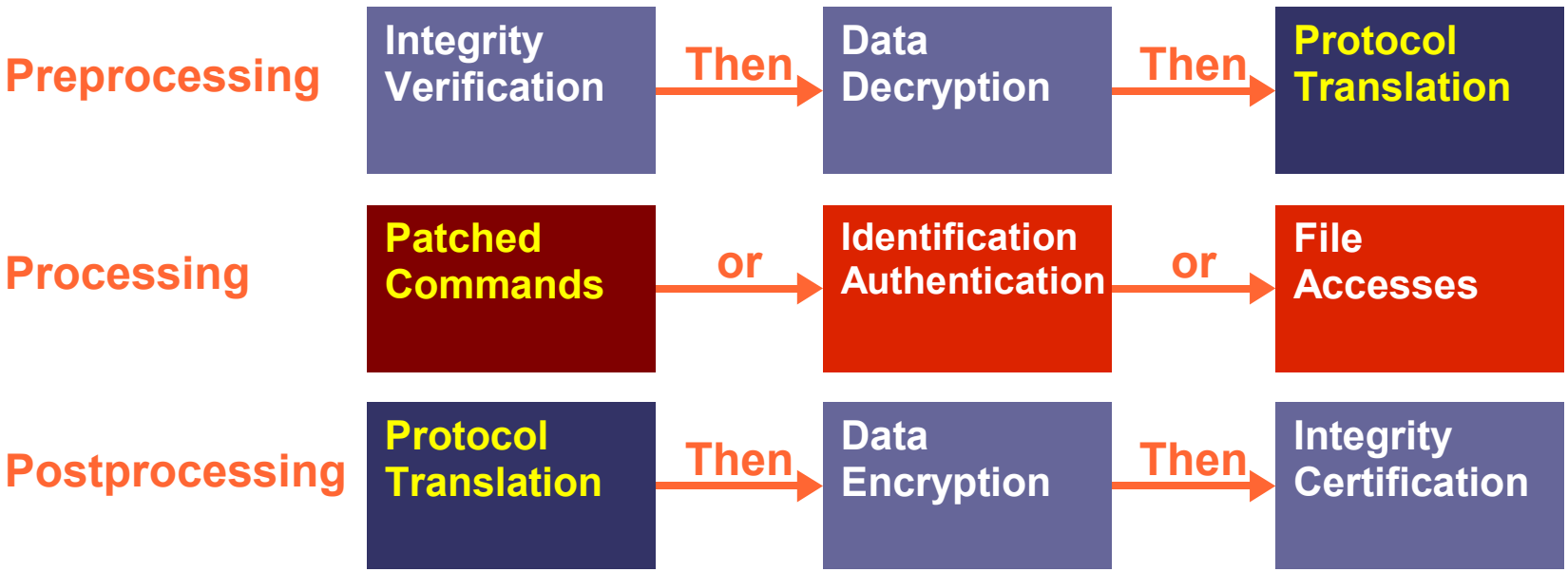
Typical Processing Sequence

With standard security

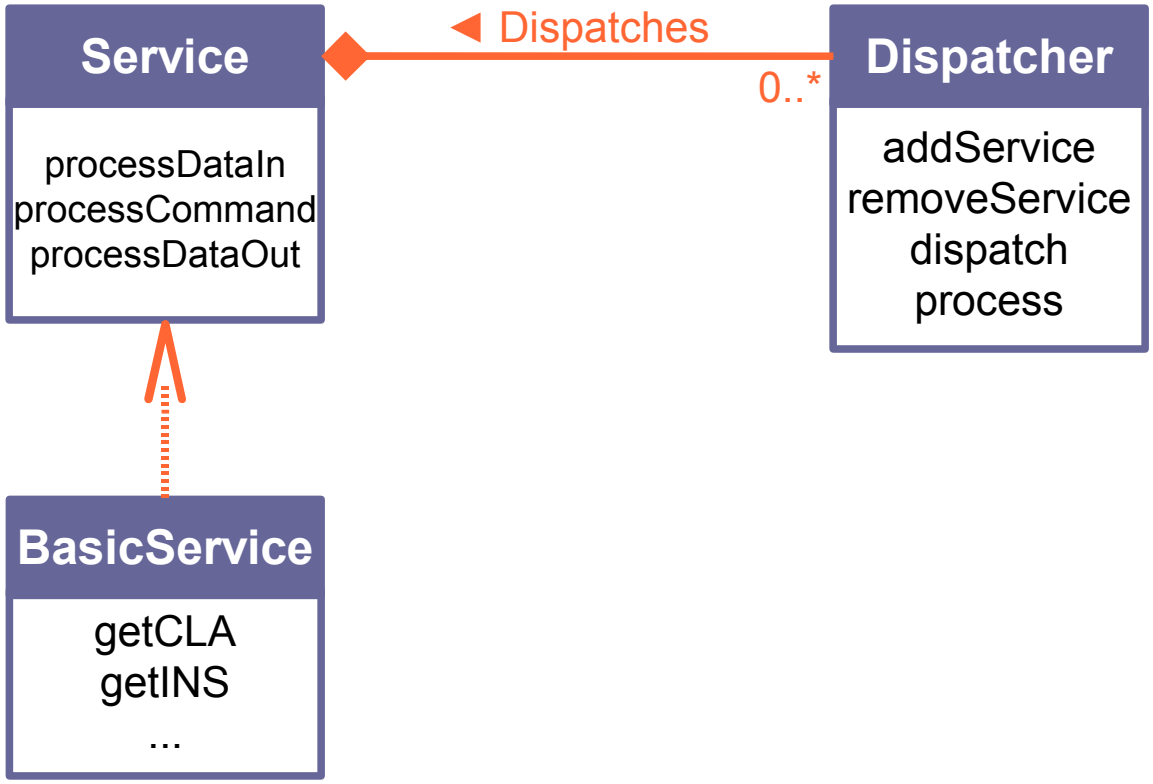


Typical Processing Sequence

With updates



The Java Card Service Framework



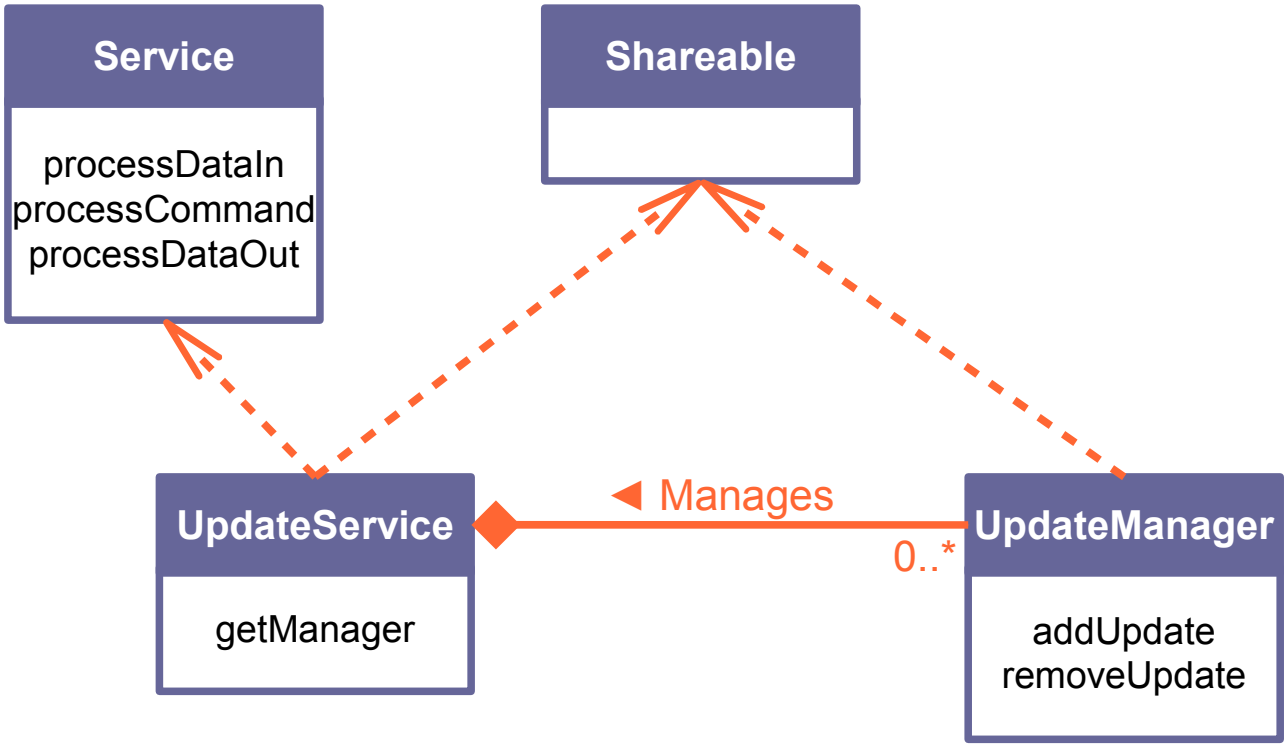
An Update Service

Required features

- Dynamic management
  - Updates can be added at any time
  - Updates can be added for any phase
  - An update may be replaced by another one
- Code extension
  - Updates are allocated in another context
  - Updates may need to access some data
- Simplicity
  - Update management must be simple
  - The number/size of updates must remain under control

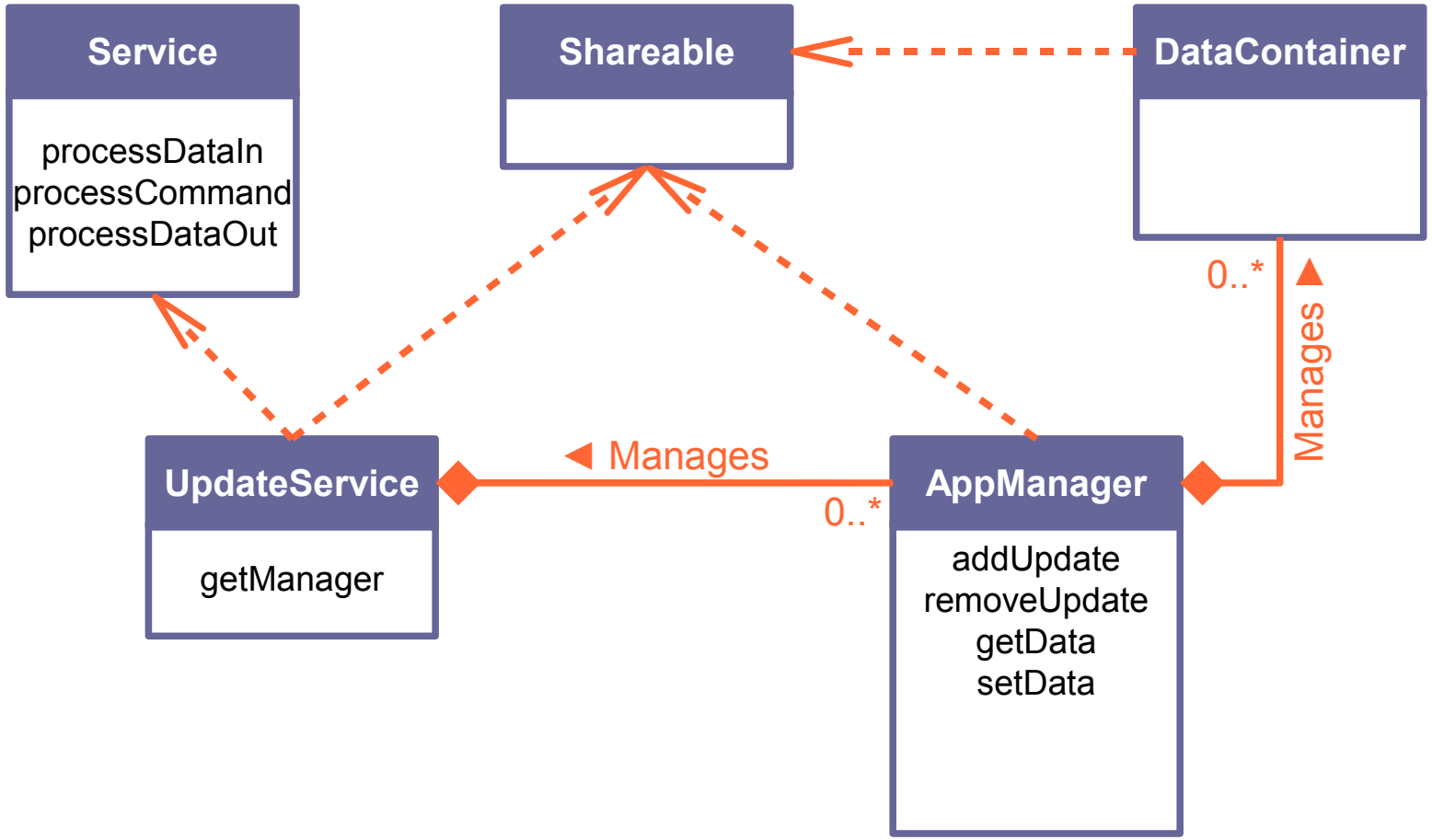
An Update Service

Minimal features



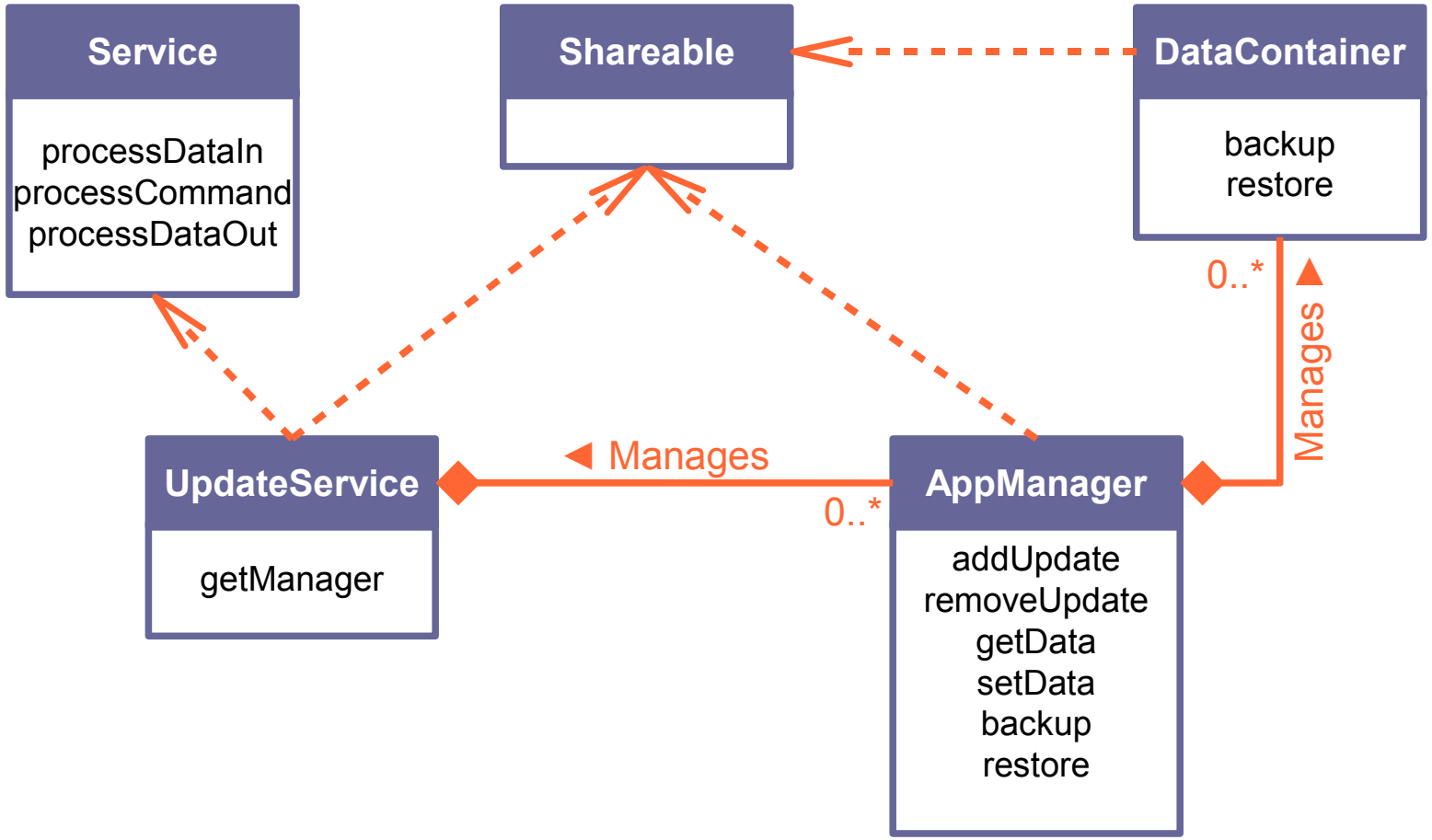
An Update Service

Adding access to data



An Update Service

Adding backup and restore



A Few Trivial Details

- How to identify an update
  - `void AppManager.addUpdate (byte phase, short id, UpdateService service)`
- How to identify a data container
  - `void AppManager.getData(short id)`
- How to perform a data backup
  - `byte DataContainer.backup (byte seq, APDU apdu)`

Is It That Simple?

Of course not

- The sequencing of updates is important
  - A patch has to be inserted in the correct spot
  - This is very much application-dependent
- Exchanging actual data is not obvious
  - Firewall constraints are quite strong
- Backup and restore is difficult
  - There is no reflection in Java Card platform
  - Reallocation on restore is a difficult issue

Security Issues

Applications and updates

- An update must be authenticated
  - A patch is always sensitive
  - An application can only accept authenticated patches
- The application may need to be authenticated
  - Some updates may be sensitive
- Data containers must be protected
  - Making data accessible through sharing is sensitive
  - The distribution of references must be controlled

Security Solutions

Using the available mechanisms

- Personalizing updates addresses the issues
  - Updates can have knowledge of a shared secret
  - Authentication is made easier
- Use the GlobalPlatform mechanisms
- Use the security domain's secure channels
    - Use all available options (including output protection)
  - Use the personalization mechanisms
- Think system, not only card application
  - Many security mechanisms can involve a server

Final Recommendations

Evolution, not revolution

- Simple is beautiful
  - More complexity, more bugs
- Refactor, don't rewrite
  - Don't throw away code that works
- Structure your code
  - Partial, localized updates are simpler
- Specify the security
  - Adding security later creates holes

Agenda

The White Card Concept

Case Studies: 2 New Architectures

Responsibility and Trust

Constraints for Developers

Practical Impact on an Application

Interfacing with Java ME

Opportunities: What? Where? When?

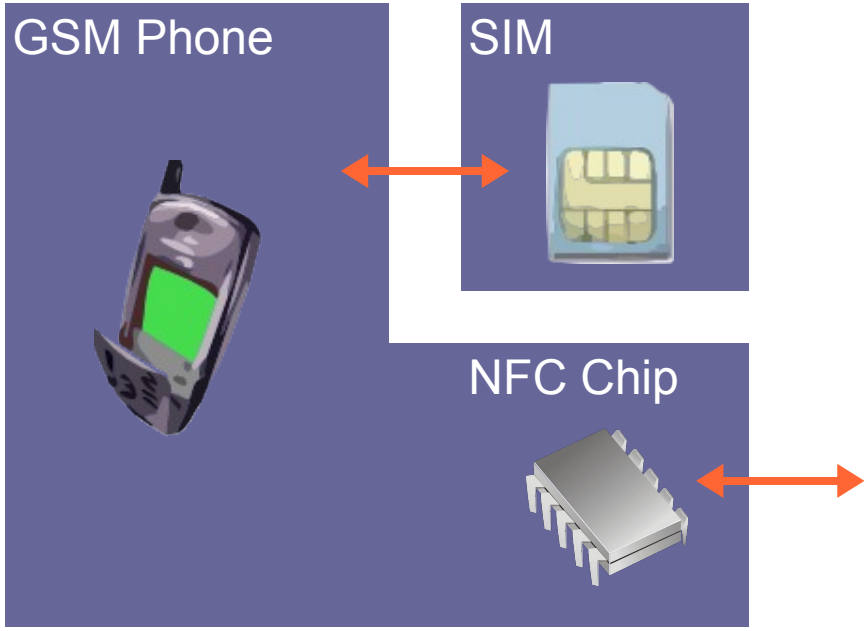
An Open System

Mobile Felica

- Introduced by DoCoMo, Sony, and partners
  - A contactless chip embedded in a phone
  - The chip is connected to the phone
- APIs are available on the phones
  - Allowing exchanges with remote servers
  - Using the phone's user interface
- Two levels of API are available
  - A basic level for simple applications
  - An advanced level for sensitive applications

Possible Architectures

Smart card, NFC, and mobile phone

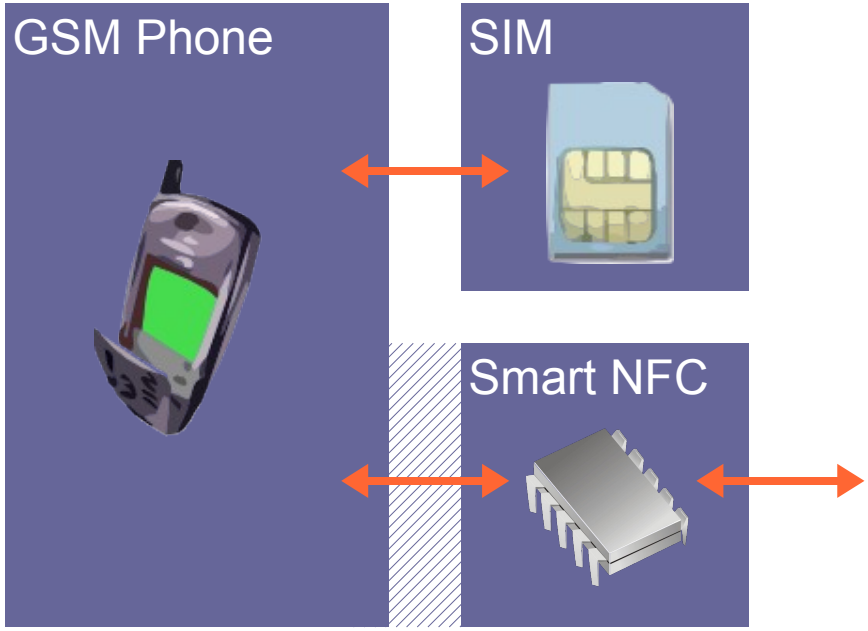


- Contactless interface
  - NFC for communication
  - Applications on the phone
- Security in the SIM card
  - Sensitive apps on the SIM
  - Accessed only through the phone/NFC

Simple NFC Chip Integrated in the Phone

Possible Architectures

Smart card, NFC, and mobile phone



- Contactless interface
  - NFC for communication
  - Applications on the phone
- Security in the Smart NFC
  - Sensitive apps on the chip
  - Communication link with the phone
  - Chip can be independent

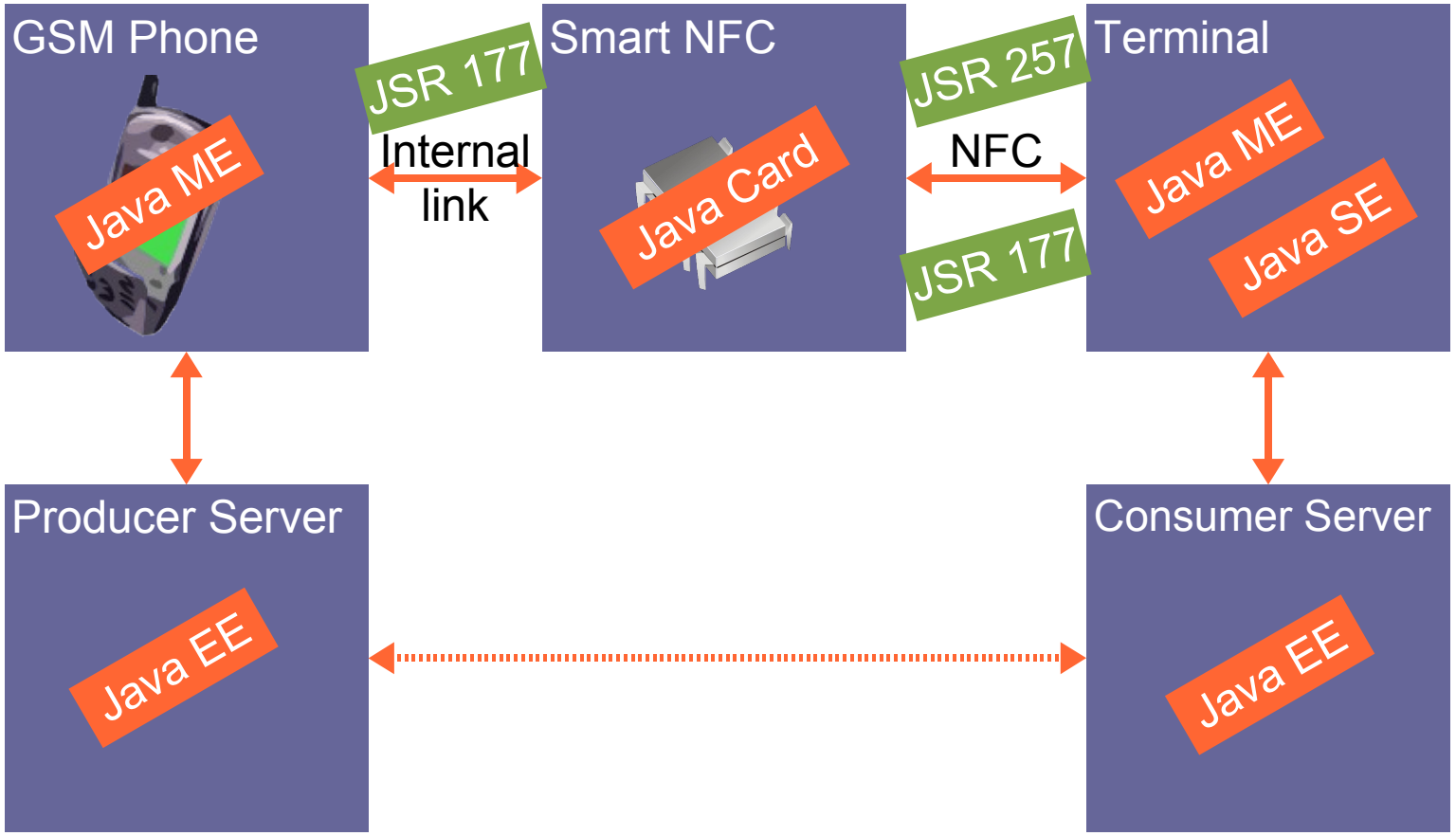
Complex NFC Chip Integrated in the Phone

The Smart NFC Architecture

Pros and cons

- Main advantage: An independent chip
  - Works independently of the phone
    - Does not require power from the phone
  - Phone software can be secured
- Potential problem: New security element
  - Could be useful for phone manufacturers
  - The operators are wary of this new development

Complete Application Architecture



Smart NFC and Mobile Phones

Typical usage patterns

- The Smart NFC is used as a smart card
  - Independently of the phone's functions
  - For instance, as a transport token
- The Smart NFC is used through the phone
  - For instance, to access a reload server
    - The phone performs the user interface
    - The phone then acts as a proxy with the server
  - In many cases, the phone is transparent
    - The communication is secured between the Smart NFC and the server

Agenda

The White Card Concept
Case Studies: 2 New Architectures
Responsibility and Trust
Constraints for Developers
Practical Impact on an Application
Interfacing with Java ME
Opportunities: What? Where? When?

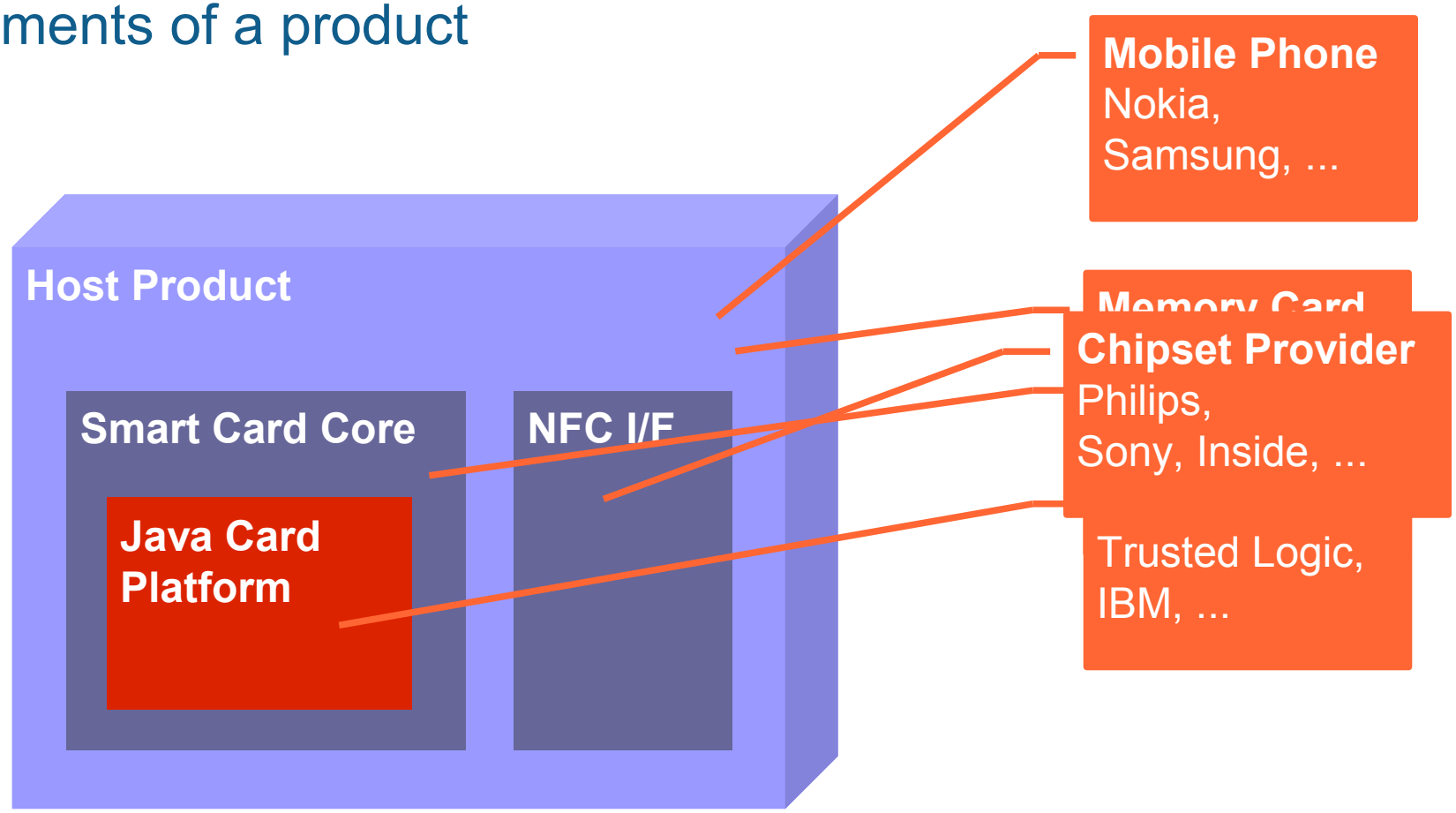
Where Are We Now?

The situation in early 2006

- People are busy preparing their offers
  - Silicon vendors are preparing their chips
  - Hardware vendors are integrating products
  - Service providers are preparing their offers
- Many prototypes and pilots are scheduled
  - Following the 2005 pilots
- Good time to prepare applications
  - Get the ideas straight, study the feasibility
  - Remain flexible in terms of architecture

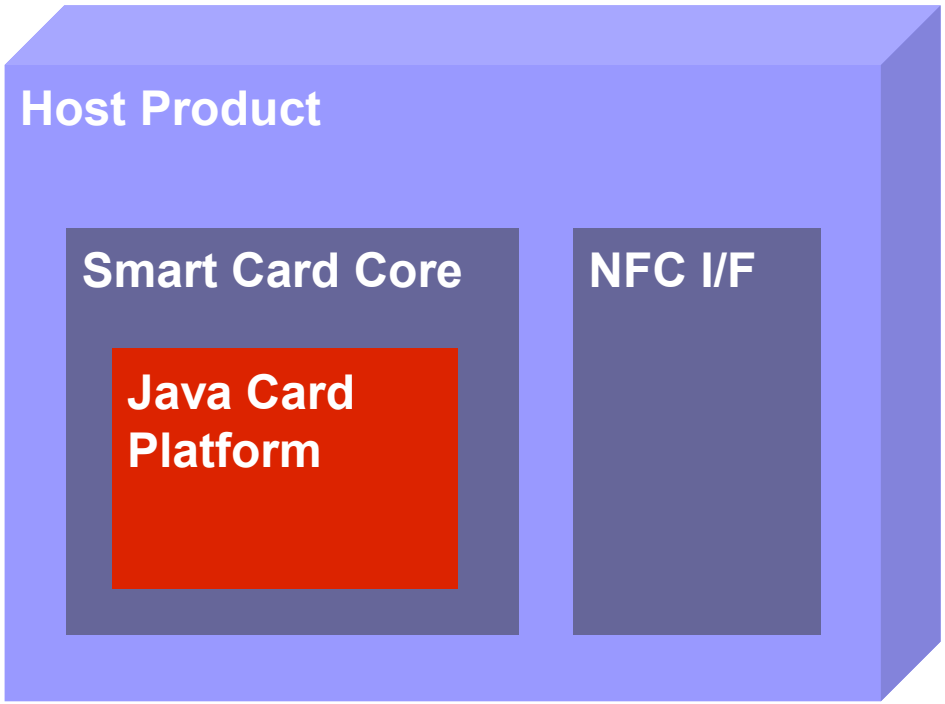
What Is a Product?

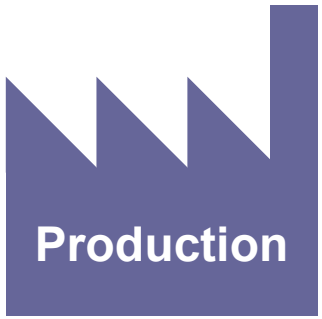
Elements of a product



What Is a Product?

Elements of a product





Smart Card Maker
Axalto, Gemplus, G&D

Other Manufacturer
Nokia, Renesas, ...

Marketing (Non) Issues

Why some people don't need their logo on the card

- Some issuers don't care about image
  - For instance, public services (transport)
- Some issuers put their logo elsewhere
  - For instance, on the mobile application (banking)
- Integration with a mobile is a driving force
  - For instance, for mobile payment
- Interaction with a memory card is a driving force
  - For instance, for digital rights management

Business Opportunities

Example 1: Transport, with RATP

- RATP uses a contactless application
  - The NaviGo pass, based on the Calypso specification
- They have shown interest in Java Card technology
  - Objective: Stop issuing cards themselves
  - For them, cards are a pure cost
  - They see Java Card platform as an enabler for
    - Dynamic management of the application
    - Hosting of the application on other cards
- Some experiments are under way

Business Opportunities

Example 2: From mobile payment to home banking

- Mobile payment is growing everywhere
  - Many experiments occur in Europe
  - In North America, it builds on contactless payment
  - It has been commercially introduced in Asia
- The context is quite specific
  - Mobile operators are strongly involved
  - Banks are not the simplest kind of issuer
- Home banking also is an opportunity
  - Phishing is an increasing concern for banks
  - Mobile phones are a way to secure home banking

Business Opportunities

Example 3: Personal applications

- End-user chooses the applications to load
  - Standalone applications: password safes, etc.
  - Parts of other applications: best scores, etc.
- Only possible with really open cards
  - More forward-looking than other schemes
  - No existing trials for now
  - Also requires NFC on terminals (like home PCs)

Summary

- More open card systems are becoming common
  - Some companies are ready to share cards
- NFC devices may be strong enablers
  - They address the cost issue for white cards
- Applications need to be ready
  - User control implies some new constraints
  - Applications need to be ready for certification
- Many prototypes are happening now
  - It is time to design applications

For More Information

- Java-based standards
  - Java Card 2.2.2
    - java.sun.com/products/javacard
  - Security and Trust Services API for J2ME™ (JSR 177), Contactless Communication API (JSR 257)
    - www.jcp.org/en/jsr/detail?id=177
    - www.jcp.org/en/jsr/detail?id=257
- Standard organizations
  - GlobalPlatform
    - www.globalplatform.org
  - NFC Forum
    - www.nfcforum.org

Q&A

Eric Vétillard
eric.vetillard@trusted-labs.com

New White Card Schemes and Java Card™ Technology

Eric Vétillard

CTO
Trusted Labs
www.trusted-labs.com

TS-3814