



# Architecture of SEMPER

Michael Steiner

IBM Zurich Research Laboratory

CH8803 Rüschlikon, Switzerland

Ph. +41 1 724 8286 / Fax +41 1 724 8955

<sti@zurich.ibm.com> / <<http://www.zurich.ibm.com/~sti>>

*Second Public SEMPER Workshop,  
November 4, 1998*

ACTS



# Outline

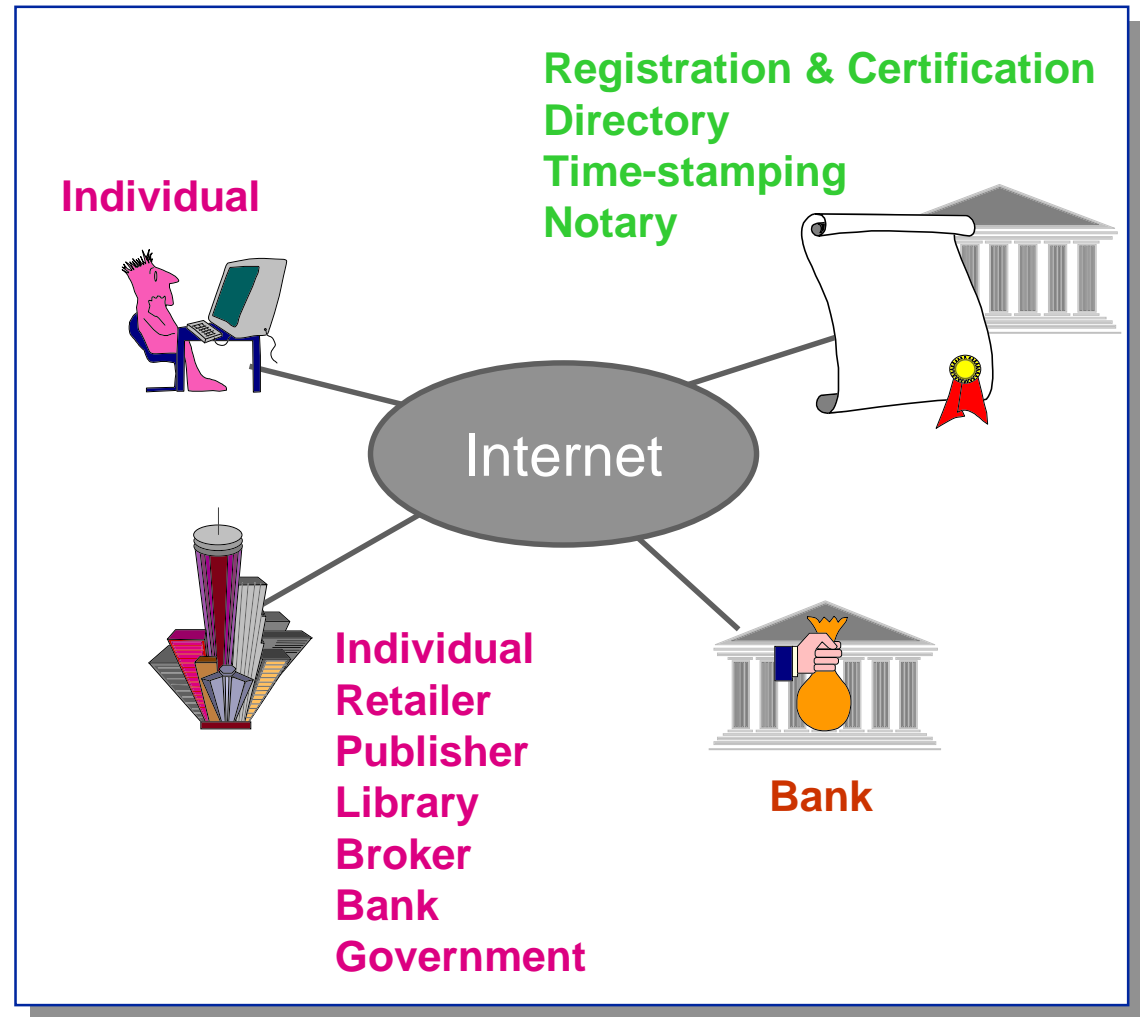
---

- ◆ **Introduction**
- ◆ **Objectives**
- ◆ **Model**
- ◆ **Architecture**
- ◆ **Outlook**
- ◆ **Conclusions**

# Introduction

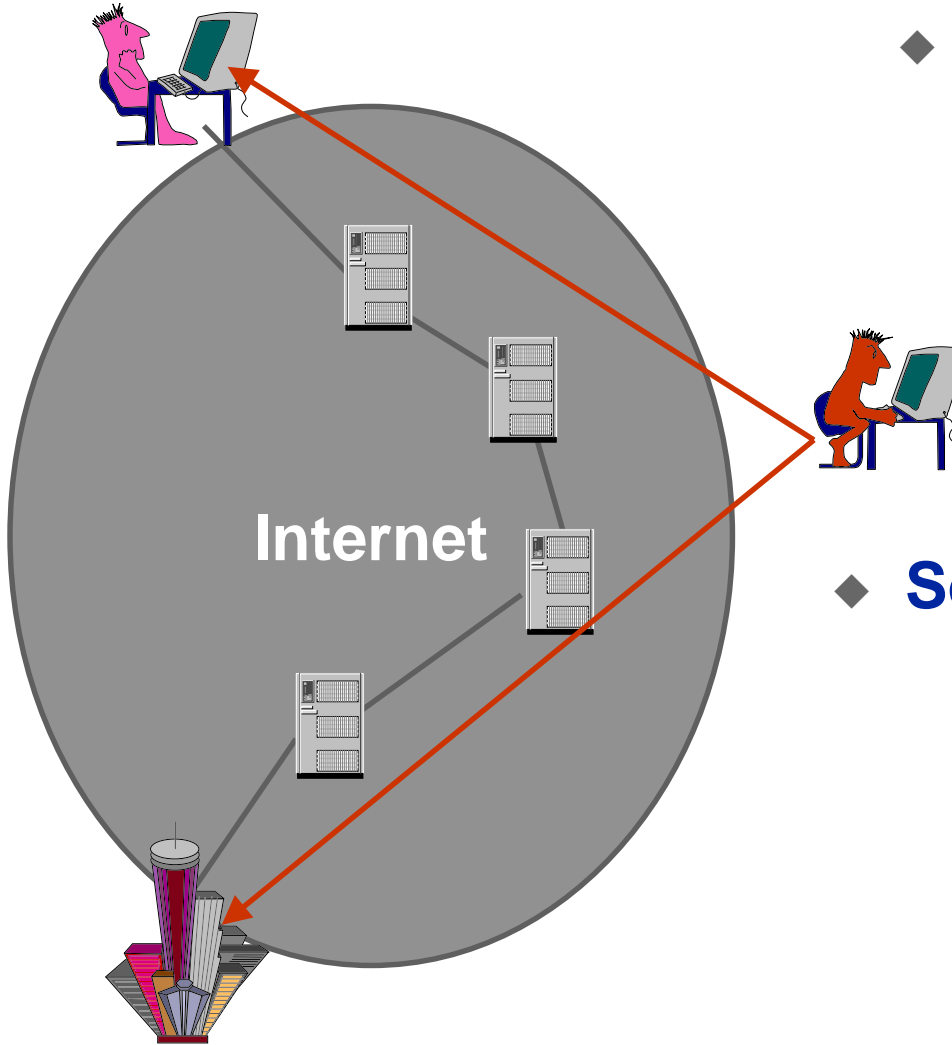
- ◆ Document Exchange
- ◆ Mail Order Retailing
- ◆ Electronic Publishing
- ◆ Ticketing
- ◆ Subscriptions
- ◆ Information Brokerage
- ◆ Auctioning
- ◆ ...

## Electronic Marketplace “Internet”



# Introduction

## Current Situation



### ◆ Requirements

- ◆ Integrity
- ◆ Authentication
- ◆ Confidentiality

### ◆ Some issues

- ◆ Weak cryptography
- ◆ Trust in CA
- ◆ Trust in Peer

# Secure Electronic Commerce

---

- ◆ **More than secure communication ...**
  - ◆ **Multi-party problems:** payments, notarized contract signing, auctioning, copy protection, ...
  - ◆ **Multi-party Security:** limit trust in others, make trust explicit, verifiability of trusted parties
- ◆ **More than payments ...**
  - ◆ **Processes:** Systems must be securely linked, e.g. contract with payment with delivery
- ◆ **More than technical security...**
  - ◆ **Legal and technical foundations:** Digital signatures; registration & certification; secure hardware; unified user interfaces; Dispute handling...

# Objectives

## Objectives for Architecture

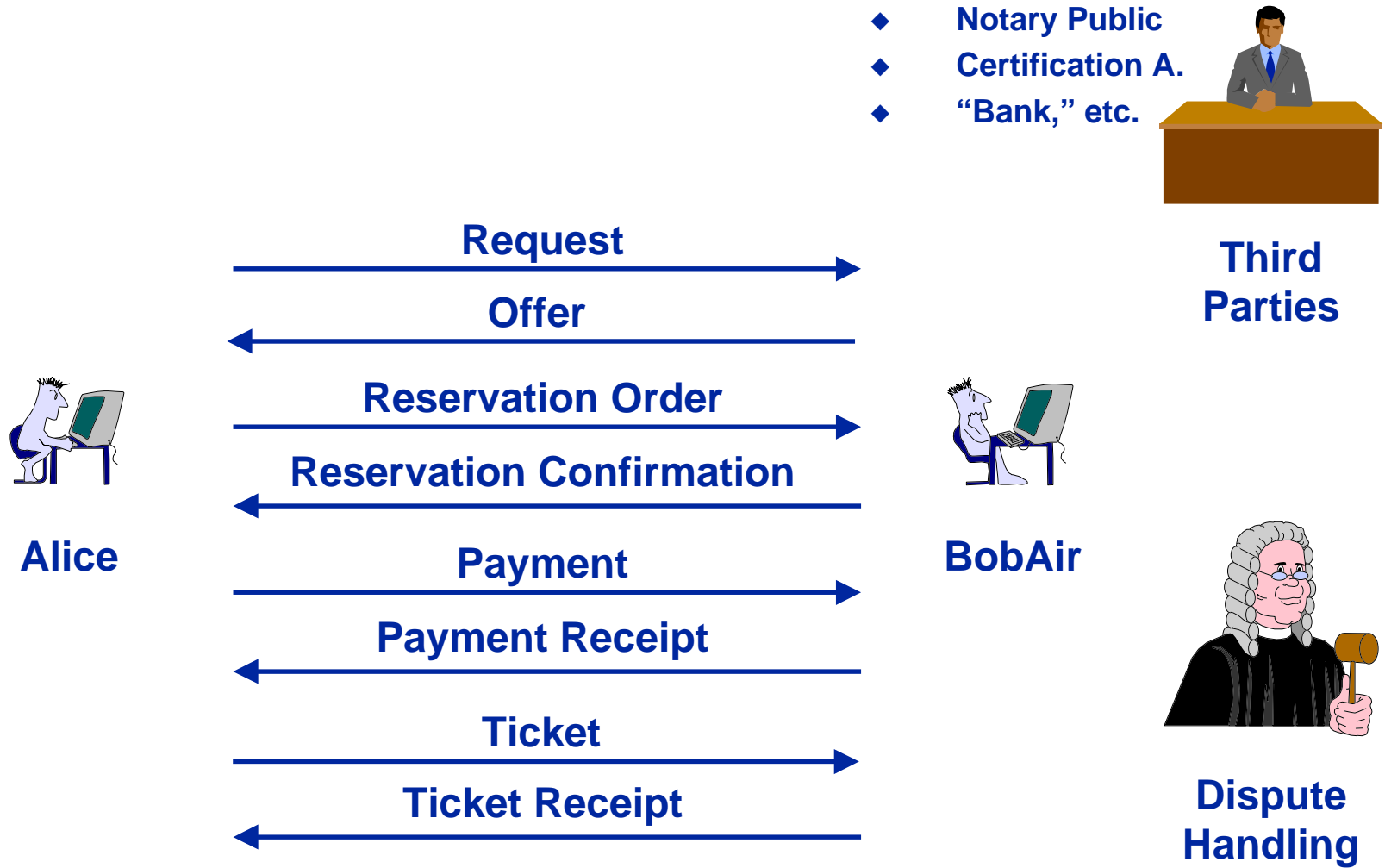
---

- ◆ **Coherent model as basis**
  - ◆ should be easy to understand ..
- ◆ **Security as driving factor**
  - ◆ security cannot be added later ...
  - ◆ addresses multi-party security requirements
  - ◆ supports dispute handling
- ◆ **Openness**
  - ◆ Extensibility
  - ◆ Uniformity
  - ◆ Generality



Model

# Scenario: Buying Airline Tickets ...



- ◆ Notary Public
- ◆ Certification A.
- ◆ "Bank," etc.

## Model

# Secure Transfer & Exchanges

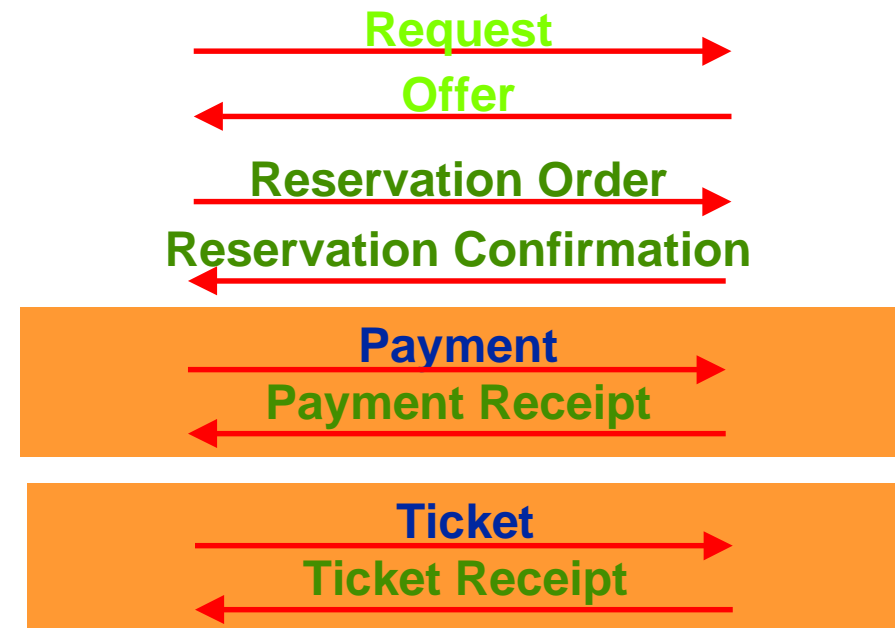
- ◆ **Transfer & Exchanges**
  - ◆ **Data**
  - ◆ **Statements**
  - ◆ **Payments**
  - ◆ **Credentials**
- ◆ **Security Requirements**
  - ◆ **Authentication**
  - ◆ **Confidentiality**
  - ◆ **Non-repudiation**
  - ◆ **Fairness**



Alice



BobAir



# Architecture

## Bird's View



Alice

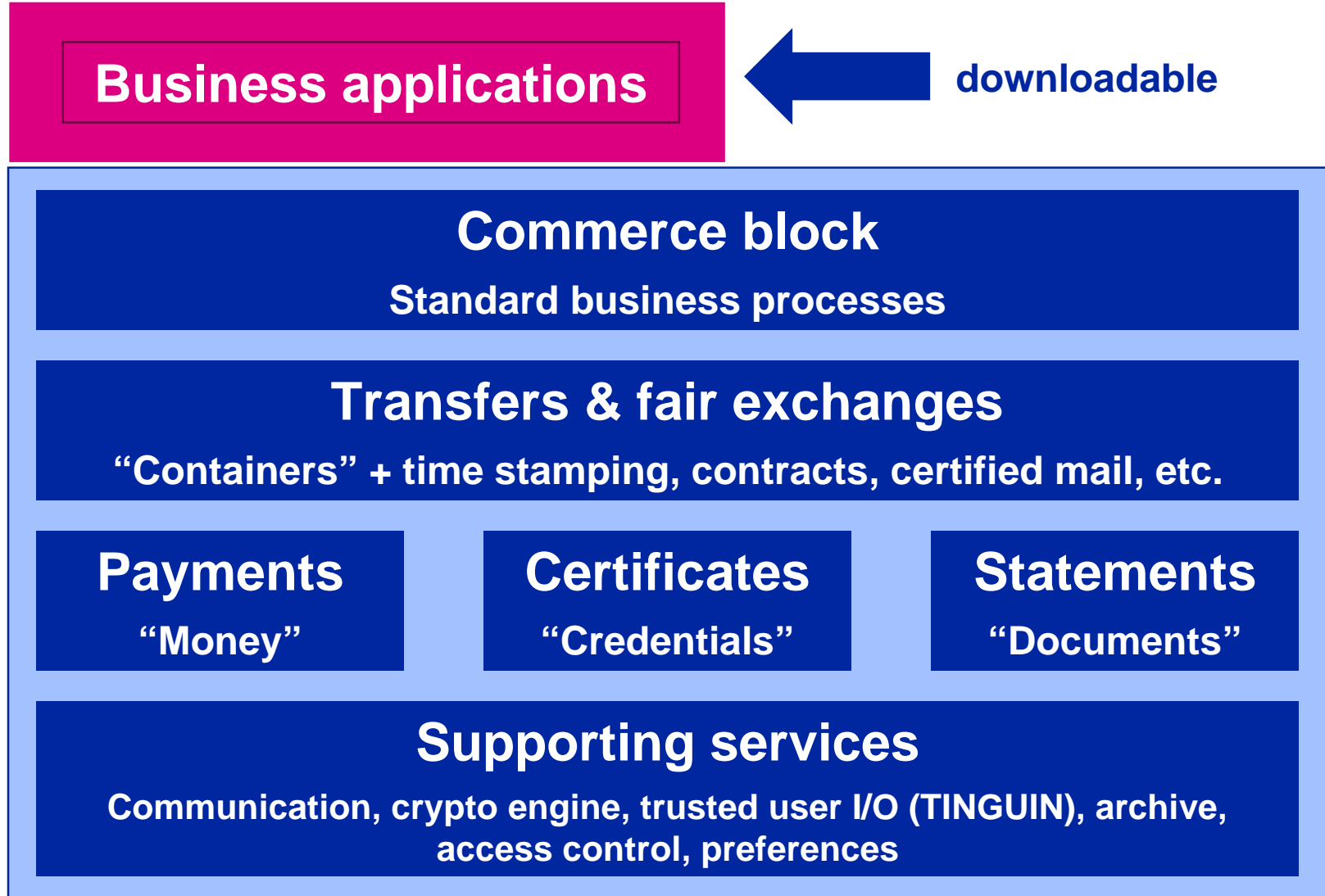


BobAir



# Architecture

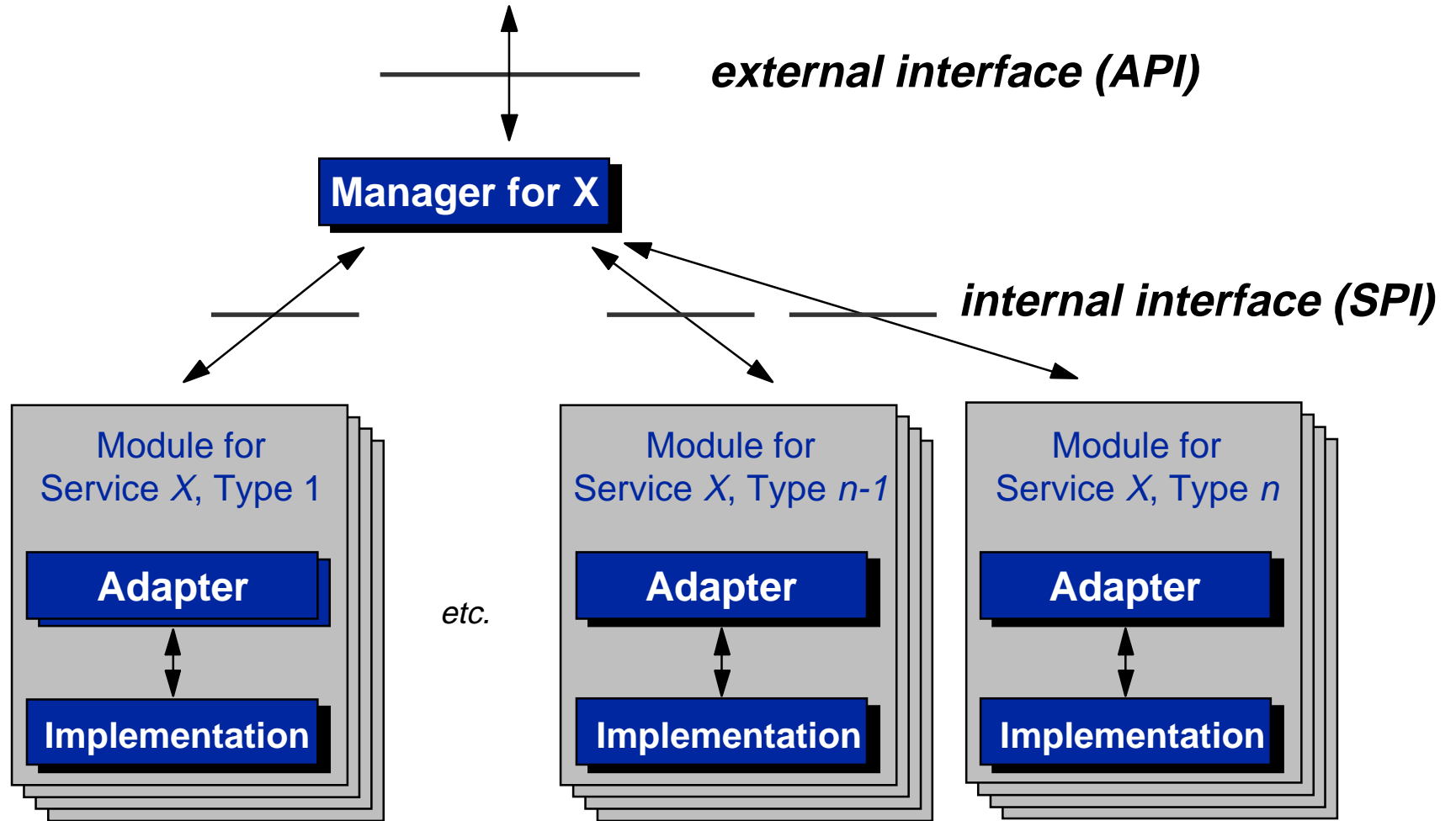
## A closer look ...





# Architecture

## Open Service Block Architecture



# Accountability: Implications for Design



Did supposed signer accept liability for such signatures?

Was supposed signer aware of signed contents?

Was supposed signer the creator of a signature?

- ◆ Certificate cannot bind a user *per se*
- ◆ Ideal registration:
  - ◆ CA and user sign a contract
    - ◆ Certification policy
    - ◆ Accepted liabilities
    - ◆ Contents of certificate
  - ◆ User interface
    - ◆ Standard GUIs
    - ◆ Standard presentation SW
    - ◆ “Point of no return”
  - ◆ Secure key handling and signature generation
    - ◆ secrets generated by user
    - ◆ secrets never leave **trusted user device** (e.g., electronic wallet)
  - ◆ Notarization of signature

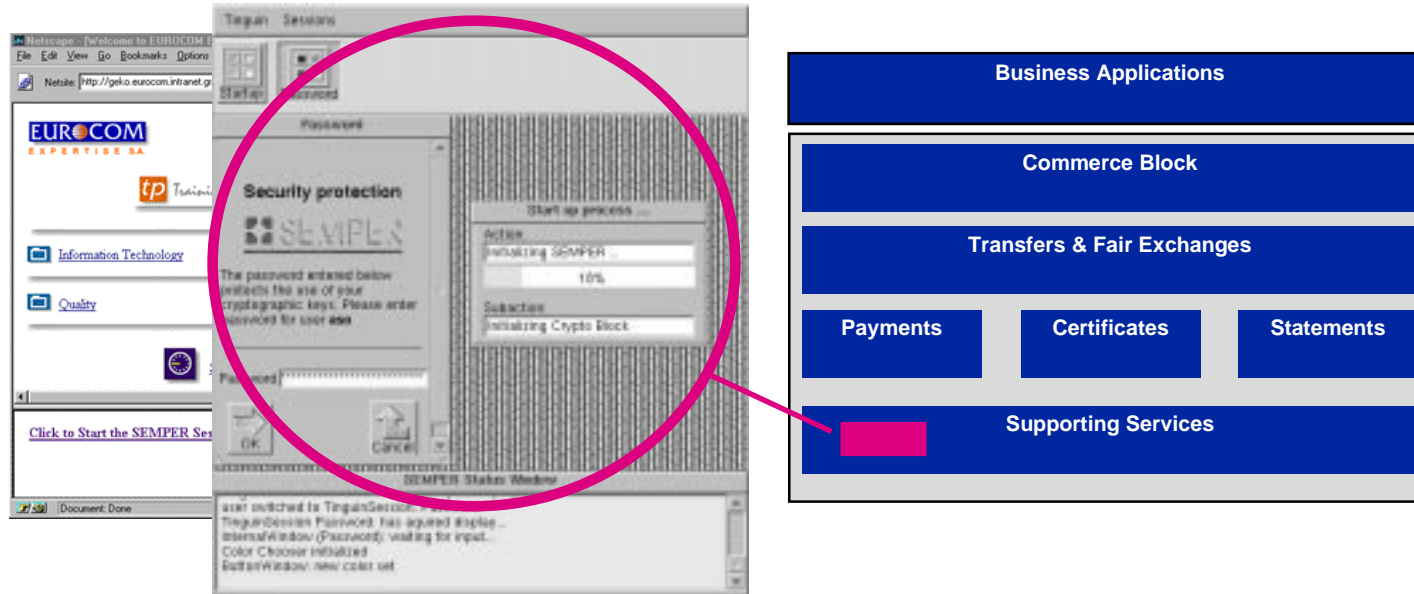
# Commerce Layer

---

- ◆ **Deals**
  - ◆ business context with negotiated quality of service
  - ◆ secure linkage of commerce transactions
  - ◆ collection of evidence, deal browser & dispute handler.
- ◆ **Commerce transactions**
  - ◆ Extensible class hierarchy of primitive transactions (payments, offers, orders, ..)
  - ◆ Core classes enforce standardized presentation of information to the user and proper authorization
- ◆ **Downloadable extensions**
  - ◆ certification

# Architecture

## Trusted Interactive Graphical User Interface

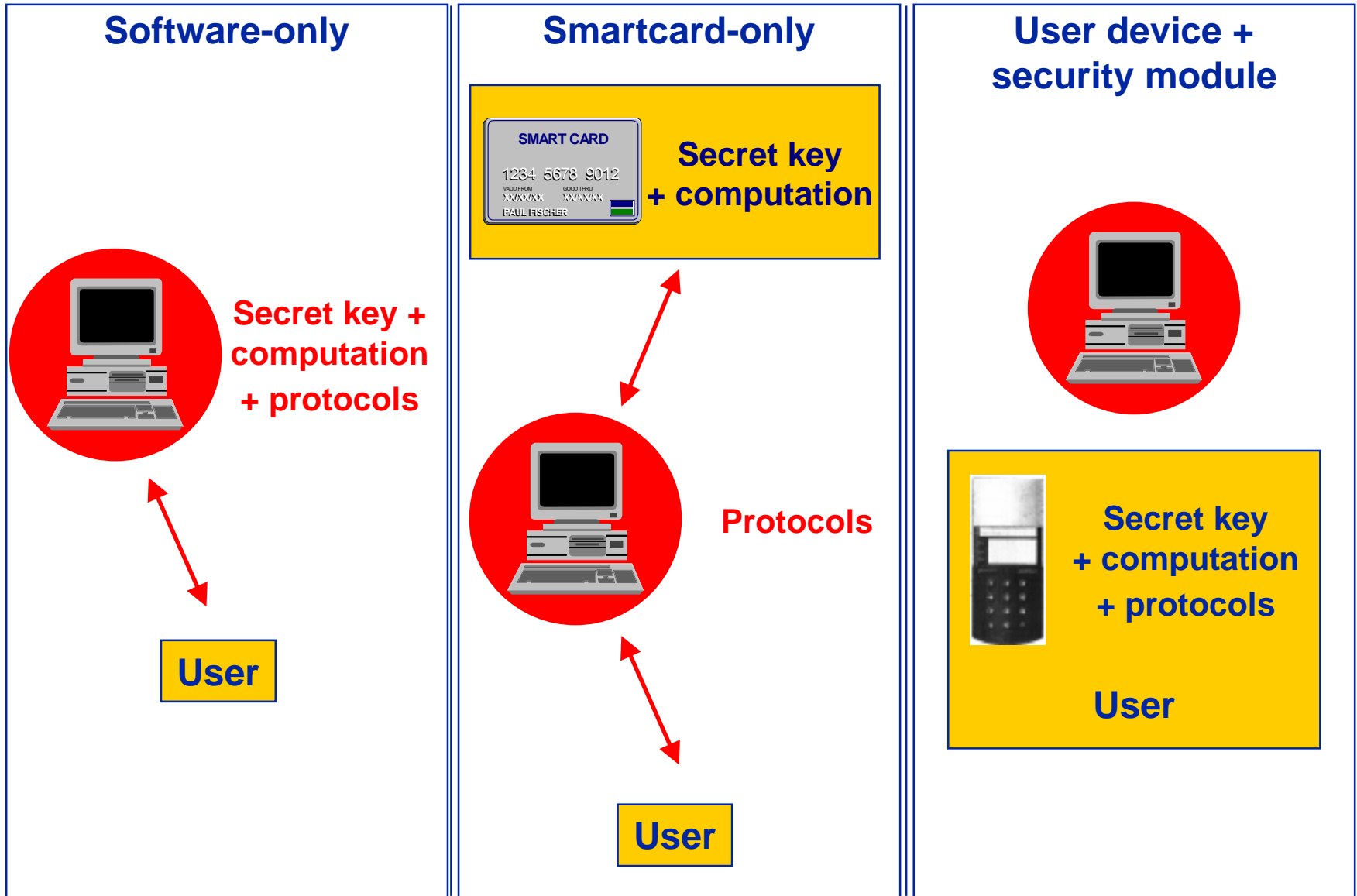


- ◆ **Why?**
  - ◆ Never trust a window on an untrusted PC or in a browser ...
- ◆ **How?**
  - ◆ Approximation in software: Dedicated window
  - ◆ Ideal solution: “Electronic wallets” with keypad & display
- ◆ **More general problem: Untrusted Hardware**



Outlook

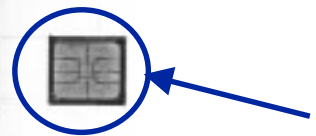
# Need for Trusted User Devices





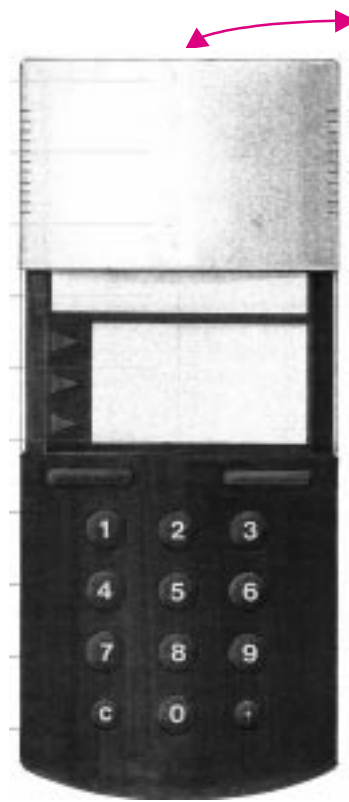
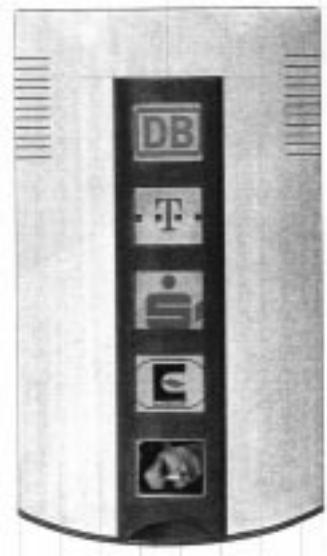
# Outlook

## Electronic Wallet: *Design Example*



**Plug-in Security Modules**

*Back*



**Infrared**

**Display**

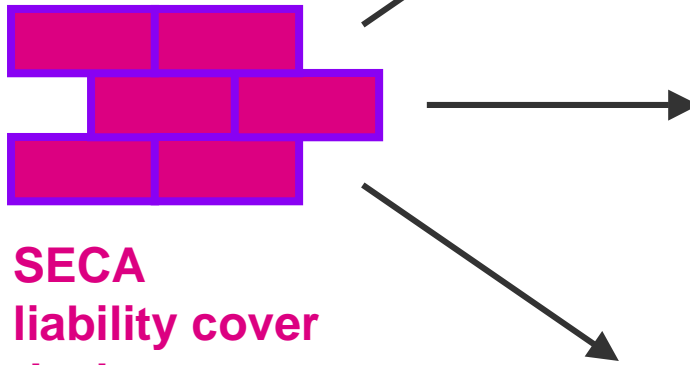
**Keypad**

*Front*

# Conclusions

## Conclusions

### Sound architecture, solid bricks



- SECA
- liability cover
- deal concept
- fair exchanges
- signed offers/orders
- trusted user interface
- pervasive anonymity
- dispute handling

### Supports multiple business models:

- business-to-business, business-to-consumer, private-to-private
- symmetric design

### Comprehensive:

- multi-party security at its core
- processes, not just steps
- large set of supported services

### Extensible:

- service framework
- generic interfaces
- downloading of modules, BAs