



# SEMPER Generic Payment Service Framework

N. Asokan

IBM Zurich Research Laboratory

CH 8803 Rüschlikon, Switzerland

Ph. +41 1 724 8608 / Fax +41 1 724 8953

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

*Second Public SEMPER Workshop*

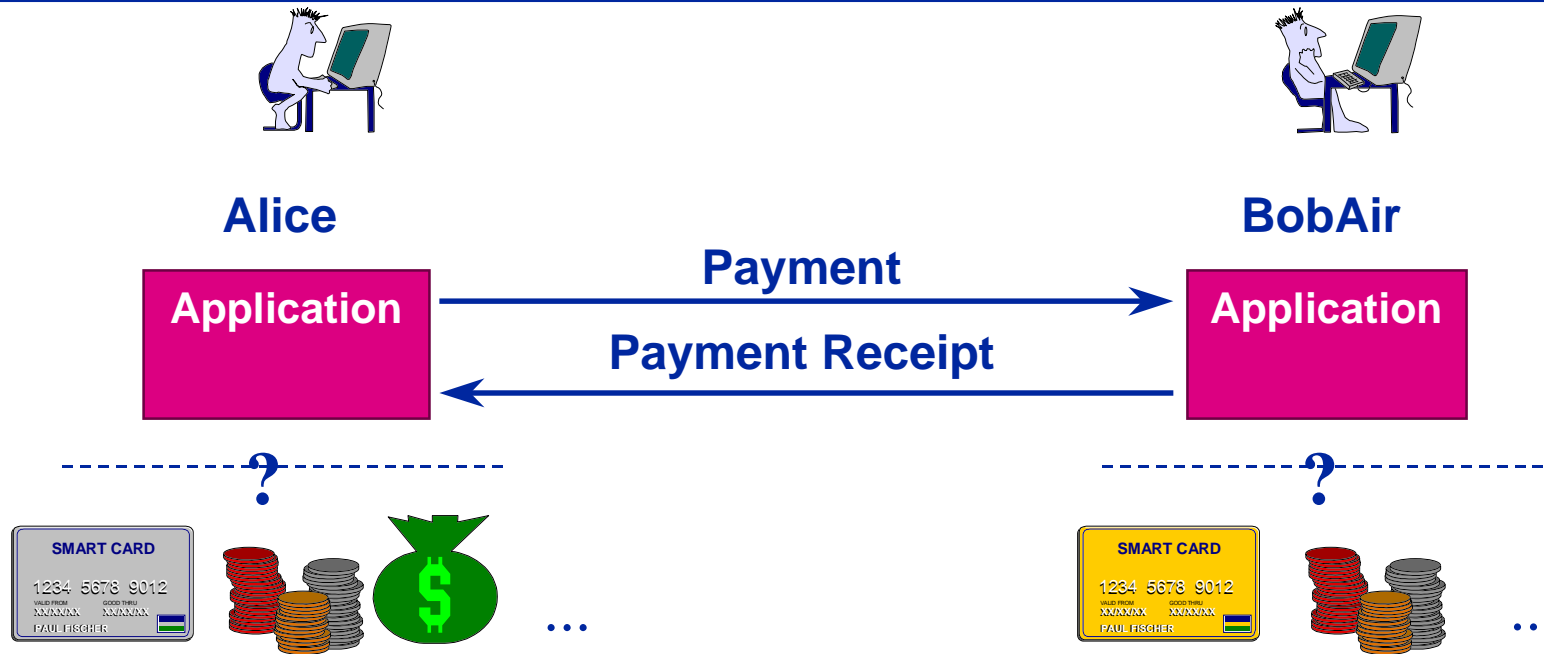
*November 4, 1998*

ACTS



# 1. Introduction

## Using Electronic Payment Systems



- ◆ **How to allow applications to**
  - ◆ use any available payment system?
  - ◆ choose a suitable payment instrument for a transaction?

# 1. Introduction

## Outline

---

### 1. Introduction

- context, objectives

### 2. Design

- service interfaces, architecture

### 3. Usage

- writing adapters, usage from applications

### 4. Extensions

- token-based protocol interfaces, dispute handling, authorisation policy framework

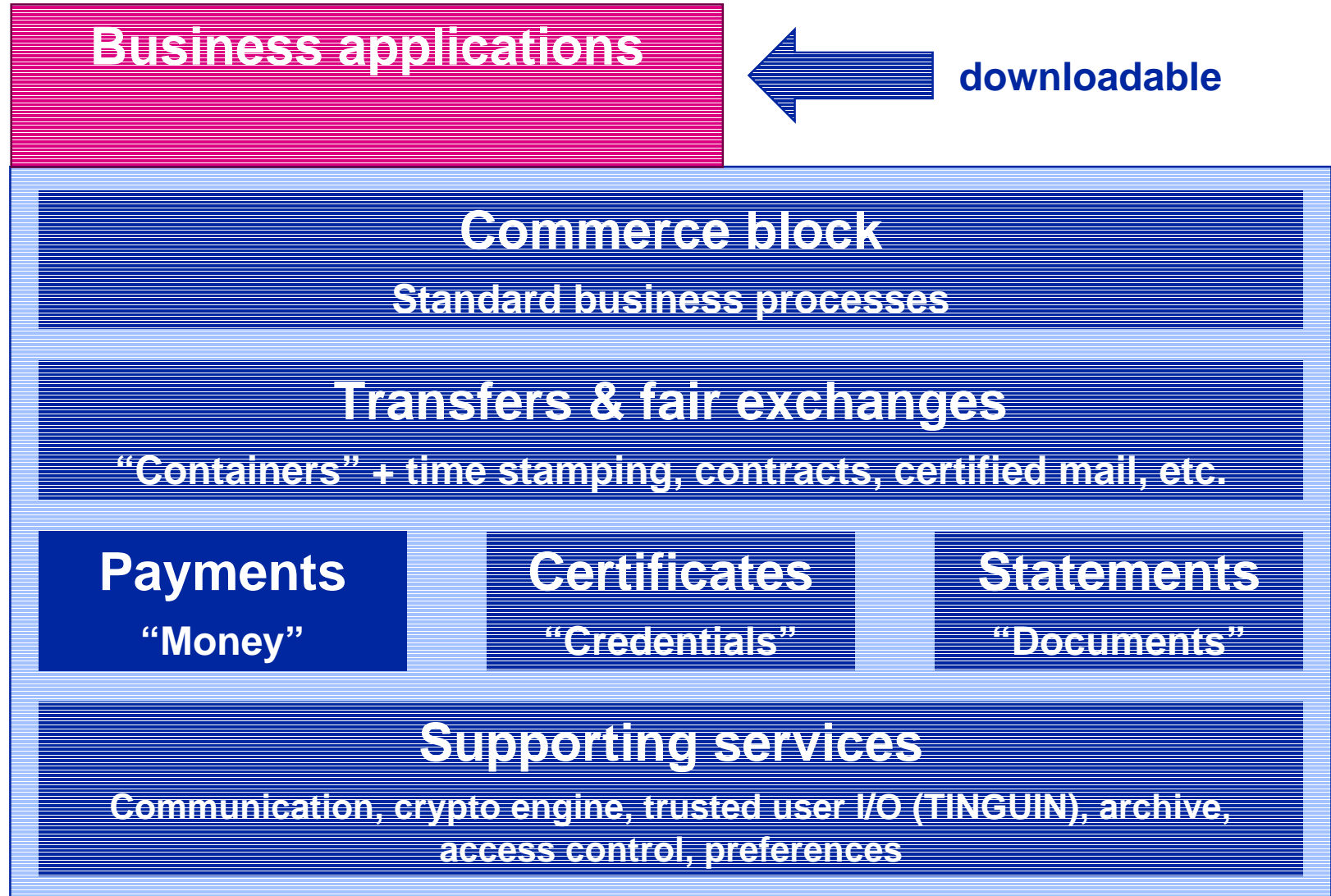
### 5. Conclusion

- summary, credits



# 1. Introduction

## SEMPER Architecture



# 1. Introduction

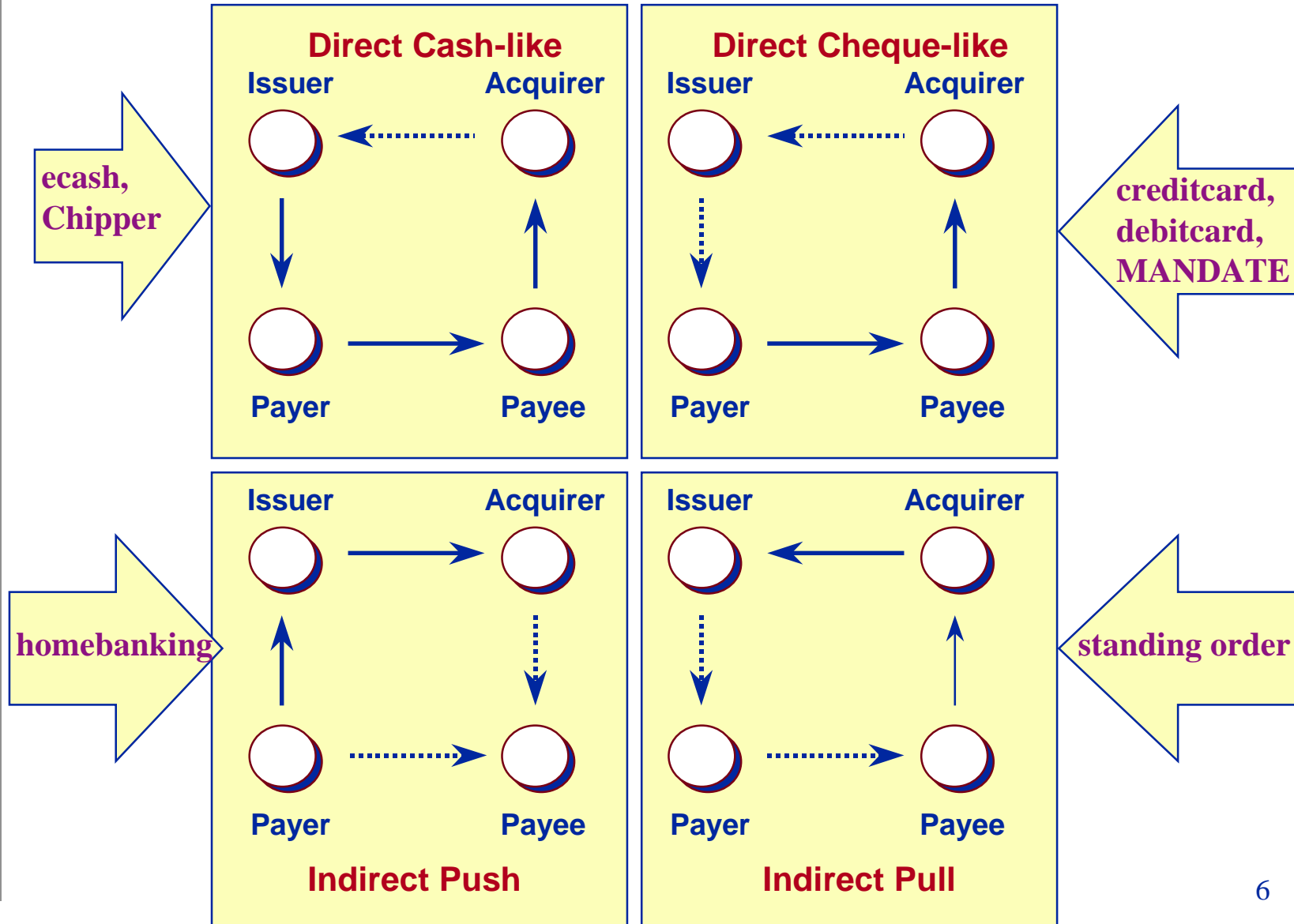
## Design Objectives

- ◆ **Generic Payment Service Framework (GPSF)**
  - ◆ Unified interface for services
    - ◆ enable development of applications independent of payment systems
    - ◆ support adapting new payment systems
  - ◆ Management of multiple means of payment
    - ◆ payment instrument selection, negotiation
- ◆ **Framework for handling disputes**
- ◆ **Symmetric design for payer and payee**
  - ◆ the same player may play both roles over time



# 2. Design

## Existing Payment Models



## 2. Design

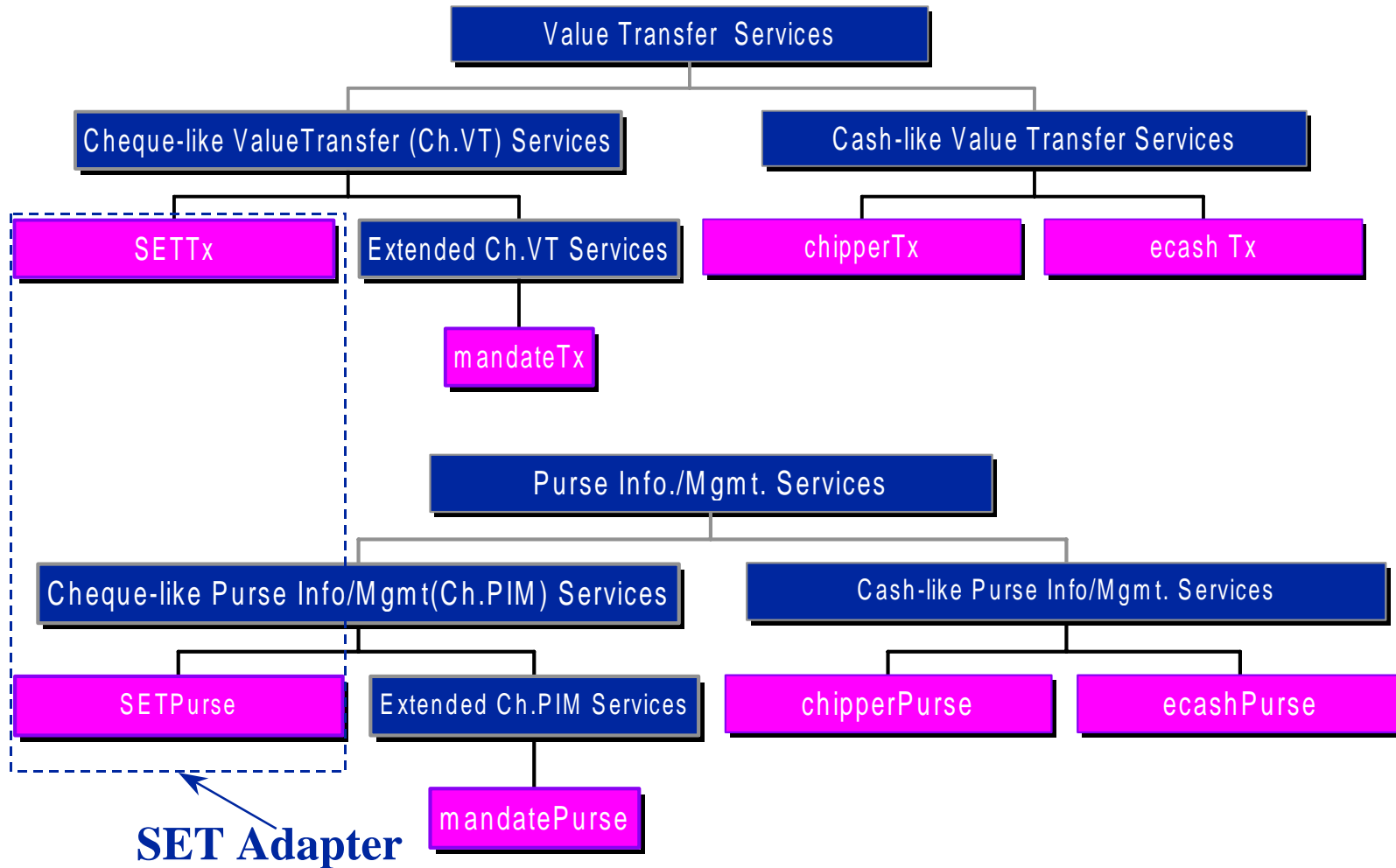
# Services

- ◆ **Value transfer**
  - ◆ move electronic value from one player to another
- ◆ **Purse management**
  - ◆ manage payment instruments in GPSF
  - ◆ policy/preferences management, access control
- ◆ **Purse selection**
  - ◆ negotiations with peer, local preferences
- ◆ **Transaction management**
  - ◆ status, cancellation, and recovery of transactions
- ◆ **Information services**
  - ◆ information about purses and transactions
- ◆ **Dispute management**



## 2. Design

# Classes in GPSF

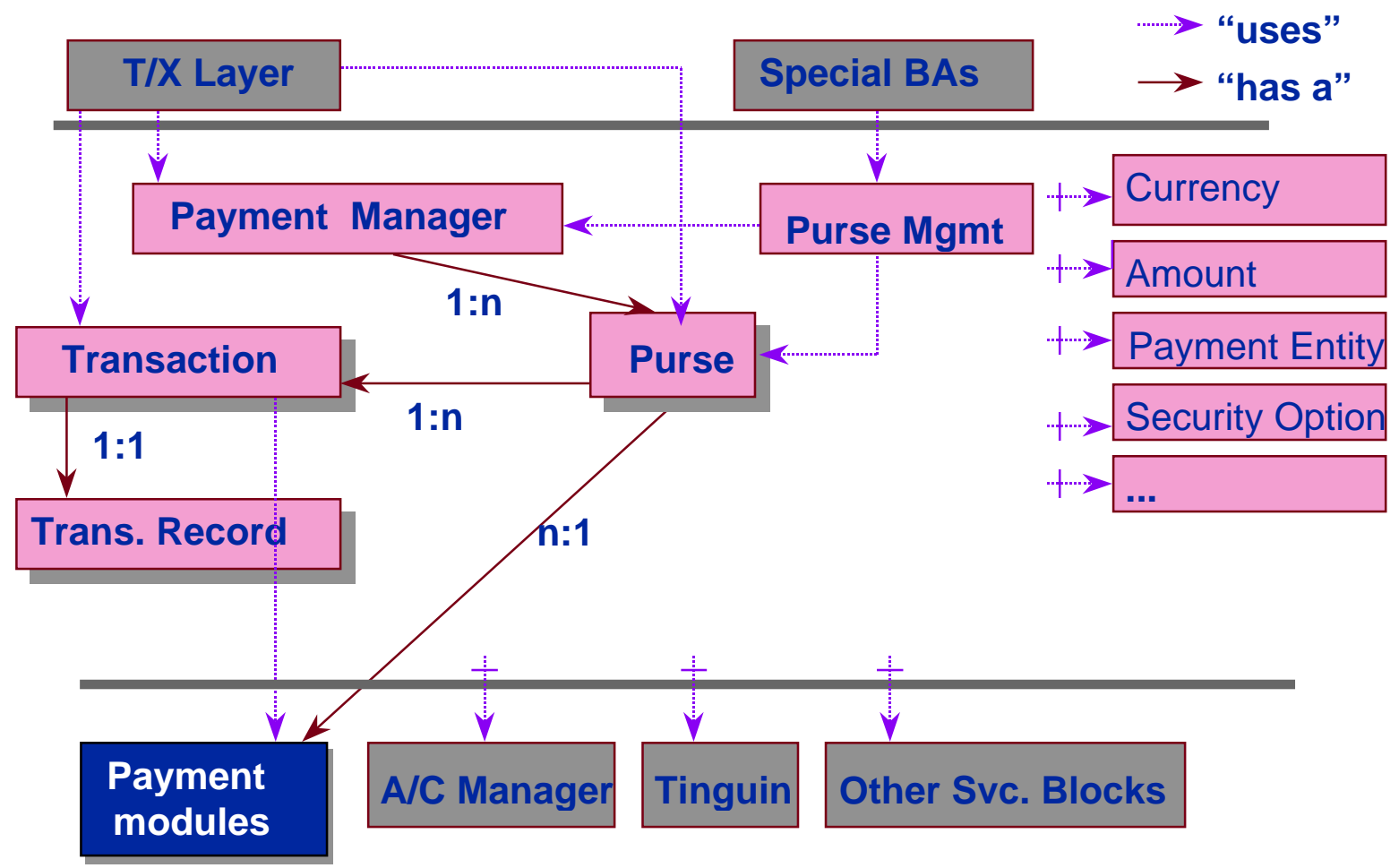






# 2. Design

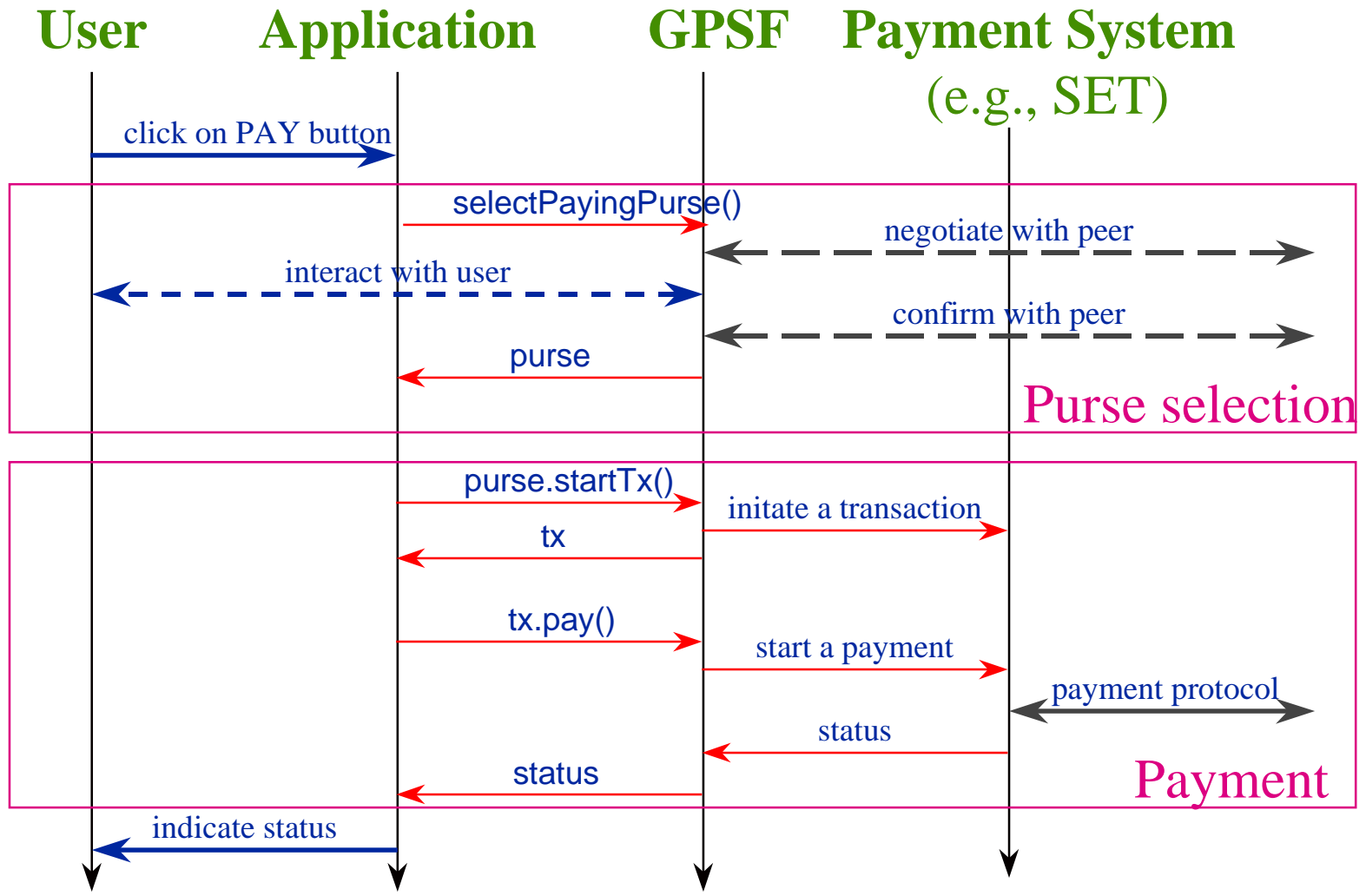
## Objects in GPSF





# 3. Usage

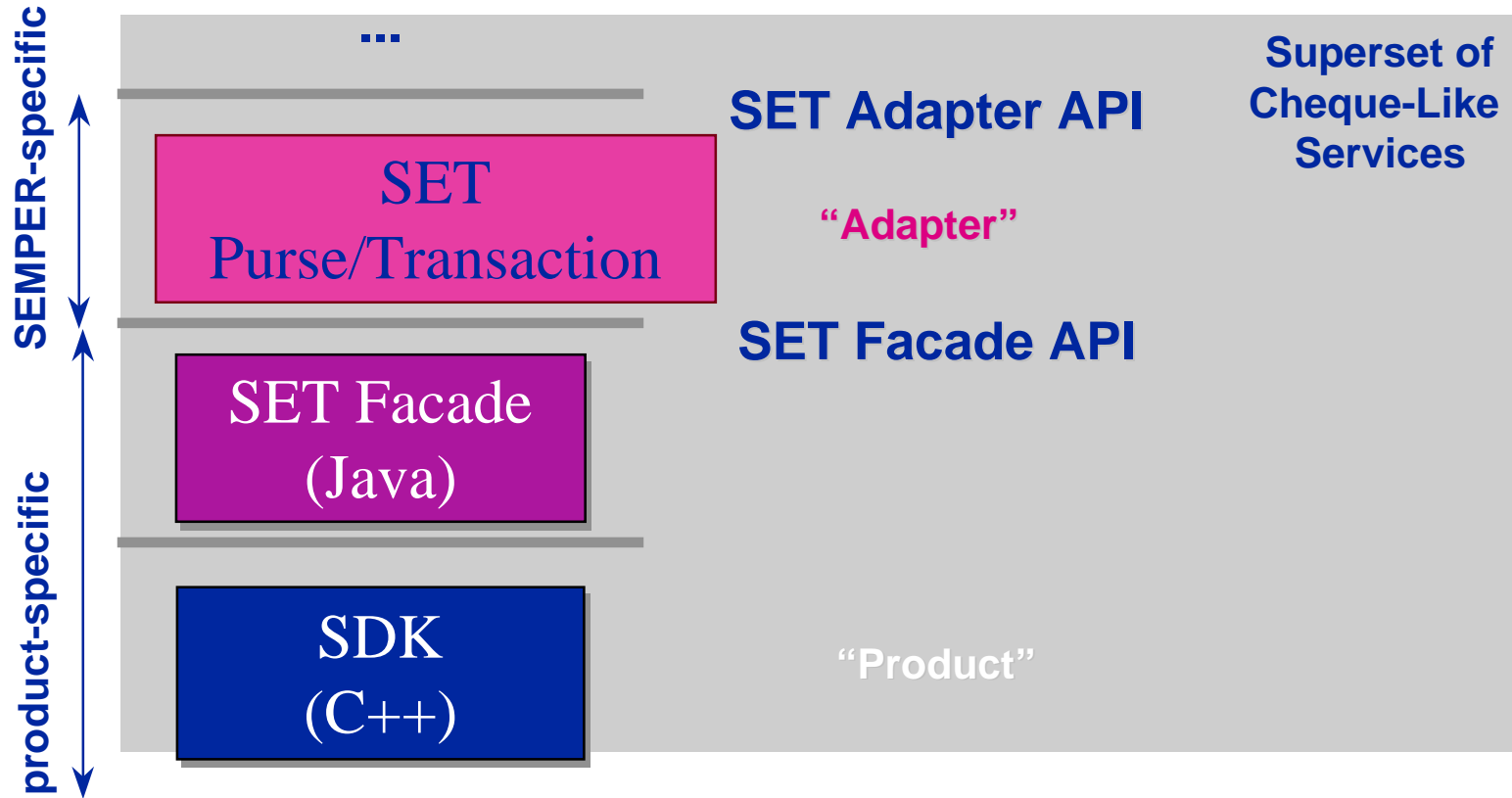
## Making Payment Transactions



### 3. Usage



# Adapting a Payment System



## 4. Extensions

# Extensions

---

- ◆ **Token-based interface definition**
  - ◆ Use existing channel between callers
    - ◆ protocol messages given back to caller
    - ◆ callers responsible for transporting tokens
  - ◆ Allows asynchronous operation
  - ◆ Allows compensation of security services
- ◆ **Authorisation policy framework**
  - ◆ access control as well as policy decisions
- ◆ **Dispute handling framework**



## 4. Extensions

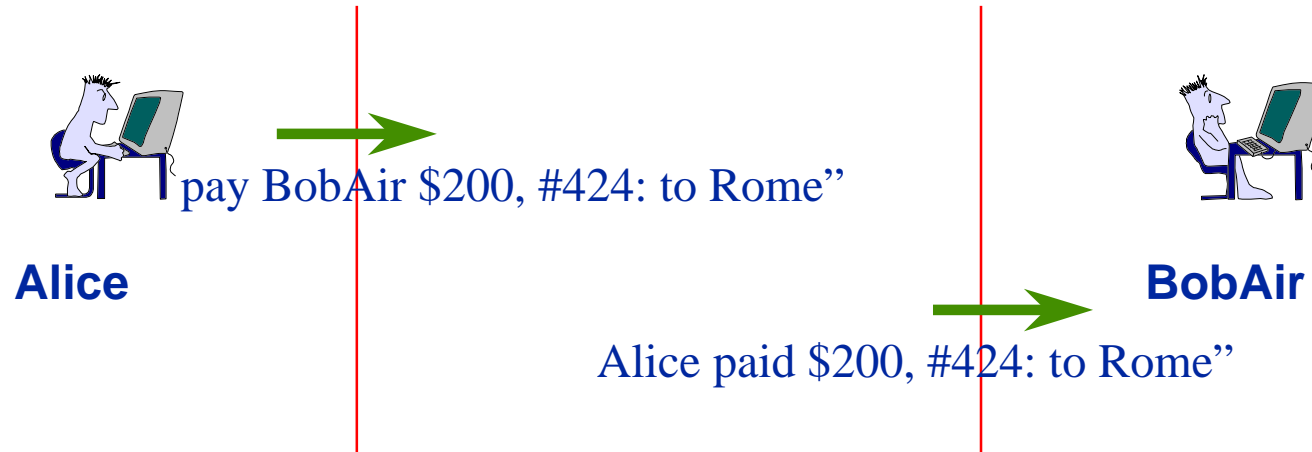


# Handling Disputes

- ◆ **No support in existing payment systems**
  - ◆ some systems collect evidence; but do not specify how to use it
- ◆ **Support more than adversarial disputes**
  - ◆ local verification, customer care, proof to third parties (e.g. income tax authorities)
- ◆ **Generality implies need for generic dispute service**
  - ◆ keep evidence inside the system; expose an interface to specify *how to use it*
- ◆ **Other issues**
  - ◆ legal significance
  - ◆ many levels (e.g., contract, payment, signature)

## 4. Extensions

# Expressing Dispute Claims



### ◆ What sort of disputes?

- ◆ Alice paid (did not pay) \$200 to Bob
- ◆ Alice paid Bob before March 19, 1995, 12:00CET
- ◆ Bob could effect a payment with no further action from Alice
- ◆ Alice once paid \$200 to Bob (but Bob made a refund)
- ◆ Alice did not approve a debit of \$300

## 5. Conclusions



# Conclusions

- ◆ **GPSF is an extensible framework**
  - ◆ generality, transparency, abstraction
- ◆ **Prototype implementation**
  - ◆ basis for IBM CommercePOINT e-till
- ◆ **Adapters for several payment systems**
  - ◆ SET, ecash, chipper, Mandate
  - ◆ homebrew systems (generic, MarisCC, OTV bill payment)
  - ◆ MOMENTS smartcard payment system
- ◆ **Several directions for future work**
  - ◆ dispute handling
  - ◆ authorisation policies