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
Using Electronic Payment Systems

1. Introduction

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

- **Business applications**
- **Commerce block**
  - Standard business processes
- **Transfers & fair exchanges**
  - “Containers” + time stamping, contracts, certified mail, etc.
- **Payments**
  - “Money”
- **Certificates**
  - “Credentials”
- **Statements**
  - “Documents”
- **Supporting services**
  - Communication, crypto engine, trusted user I/O (TINGUIN), archive, access control, preferences
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
Existing Payment Models

- **Direct Cash-like**
  - Issuer
  - Acquirer
  - Payer
  - Payee

- **Direct Cheque-like**
  - Issuer
  - Acquirer
  - Payer
  - Payee

- **Indirect Push**
  - Issuer
  - Acquirer
  - Payer
  - Payee

- **Indirect Pull**
  - Issuer
  - Acquirer
  - Payer
  - Payee

Existing Payment Models:
- creditcard, debitcard, MANDATE
- ecash, Chipper
- homebanking
- standing order

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**
Classes in GPSF

2. Design

Value Transfer Services

Cheque-like Value Transfer (Ch.VT) Services
- SETTx
- Extended Ch.VT Services
  - mandateTx

Cash-like Value Transfer Services
- chipperTx
- ecashTx

Purse Info./Mgmt. Services

Cheque-like Purse Info/Mgmt (Ch.PIM) Services
- SETPurse
- Extended Ch.PIM Services
  - mandatePurse

Cash-like Purse Info/Mgmt. Services
- chipperPurse
- ecashPurse

SET Adapter
Objects in GPSF

2. Design

- T/X Layer
- Special BAs
- Payment Manager
- Purse Mgmt
- Transaction
- Purse
- Trans. Record
- Payment Entity
- Currency
- Amount
- Security Option
- Payment modules
  - A/C Manager
  - Tinguin
  - Other Svc. Blocks
Making Payment Transactions

3. Usage

User
- click on PAY button

Application
- selectPayingPurse()
- interact with user
- purse

GPSF
- initiate a transaction
- tx
- tx.pay()
- status

Payment System
(e.g., SET)
- negotiate with peer
- confirm with peer

Purse selection
- Purse selection

Payment
- payment protocol
- status
- start a payment
Adapting a Payment System

3. Usage

- SET Purse/Transaction
- SET Facade (Java)
- SDK (C++)
- SET Adapter API
- Superset of Cheque-Like Services

“Adapter”

“Product”

SEMPER-specific

SEMPER-specific
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**
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)
Expressing Dispute Claims

What sort of disputes?

- Alice paid (did not pay) $200 to Bob
- Alice paid Bob before March 19, 1995, 12:00 CET
- 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
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