StreamCom: Business Analysis and System Design Specification

(Deliverables: M2, M7)

Technical Report #
IC/2002/037

Pavel Balabko
Hien Dat TRAN
Alain Wegmann

Institute for computer Communications and Applications
Swiss Federal Institute of Technology – Lausanne
EPFL-DSC-ICA
CH-1015 Lausanne, Switzerland

{pavel.palabko@epfl.ch,
alain.wegmann@epfl.ch,
hiendat.tran@epfl.ch}
September 2001
Table Of Contents

STREAMCOM: BUSINESS ANALYSIS AND SYSTEM DESIGN SPECIFICATION .......................... 1
TECHNICAL REPORT V.2 (DELIVERABLES: M2, M7) ................................................................. 1

1. INTRODUCTION ....................................................................................................................... 5

2. BUSINESS ANALYSIS SPEC .................................................................................................. 7

2.1. COLLABORATION MODEL .................................................................................................. 7
2.2. COLLABORATION SPEC ....................................................................................................... 9
2.2.1. Sell and Get a Service ....................................................................................................... 9
       Sell and Get a Service ....................................................................................................... 11
       Sell and Get a Service Activity Diagram ........................................................................... 12
       Sell a Service ..................................................................................................................... 13
       Actors in Sell a Service ...................................................................................................... 15
       Get a Service ...................................................................................................................... 15
       Actors in Get a Service ...................................................................................................... 17
       “StreamCom General Architecture”: an Example of the Business Scenario ..................... 19

3. IT SYSTEM DESIGN ............................................................................................................... 21

   Activity Diagram : Choose Service ......................................................................................... 22
   Activity Diagram: Buy Online Stream and Ticket Distribution ............................................. 23
   Conceptual Model : ClientSystem View-Point ........................................................................ 24
   Conceptual Model : RetailerSystem View-Point ..................................................................... 26
   Conceptual Model : BBSystem View-Point ............................................................................. 28
   Conceptual Model : KeyServer View-Point ............................................................................. 29
   Conceptual Model : Ticket ..................................................................................................... 30
   3.1.1. Get a Service ............................................................................................................... 31
   Activity Diagram : Get a Service in particular ...................................................................... 31
   Pattern of Get a Service ......................................................................................................... 32
   Conceptual Model : ClientSystem View-Point ........................................................................ 34
   Conceptual Model : ContentSystem View-Point .................................................................... 36
   Conceptual Model : KeyServer View-Point ............................................................................. 38
   Conceptual Model : BBSystem View-Point ............................................................................. 40
   3.1.2. Payment ....................................................................................................................... 42
   Payment ................................................................................................................................... 43

4. IT SYSTEM IMPLEMENTATION .............................................................................................. 44

4.1. SELL AND GET A SERVICE .............................................................................................. 44
4.1.1. Sell a Service .................................................................................................................. 44
       Sequence Diagram : Choose a Service ................................................................................. 44
       Sequence Diagram : Buy a Service ....................................................................................... 45
       Logical Design Class Diagram : Client System View-Point .................................................. 46
       Logical Design Class Diagram : RetailerSystem View-Point .............................................. 48
       Logical Design Class Diagram : BBSystem View-Point ....................................................... 50
       Logical Design Class Diagram : KeyServer View-Point ...................................................... 51
       Interfaces in Get a Service .................................................................................................. 52
4.1.2. Get a Service .................................................................................................................. 53
       Sequence Diagram : Get a Service ....................................................................................... 53
       Logical Design Class Diagram : ClientSystem View-Point .................................................. 54
       Logical Design Class Diagram : ContentServer View-Point ............................................. 56
       Logical Design Class Diagram : KeyServer View-Point ...................................................... 58
       Interfaces in Get a Service .................................................................................................. 60
4.1.3. Redeem ................................................................. 61
Redeem ........................................................................ 61
1. Introduction

This report represents the work of several people within the framework of the StreamCom project. This work, done together with our colleges from the University of St. Gallen\(^1\), covers the modeling part of StreamCom. The aim of the modeling part of the StreamCom project includes the following goals (see Figure 1):

- Specify the main business actors, business activities and business environment for the StreamCom project (done by ICA).
- Specify the Generic Components model for streamed information distribution (done by MCM).
- Provide a system design that specifies one of the possible system’s behaviors that is based on the StreamCom Business Analysis and use the StreamCom Generic Component model (done by ICA).

---

\(^1\) See: Markus Greunz, Katarina Stanoevski-Slabeve « Generic Components that enable Business Models for Content Streaming», Universität St. Gallen – Hochschule für Wirtschafts-, Rechts- und Sozialwissenschaften (HSG)
The StreamCom Business Analysis Model done by ICA uses generic actors and generic business actions (or collaborations), represented in the StreamCom Generic Components Model. According to the General System Theory\(^2\), any “system is inseparable from its environment”. This means that the consideration of the environment of a system is important in understanding the system itself. That is why one of the main goals of the business analysis is to consider the environment of systems that can be build on the base of the StreamCom platform. In this work we have considered the following business actors in the StreamCom environment: Ads Provider, Stream Owner, Stream Publisher, Monitor Company (see the Section 2.1).

The StreamCom Generic Components part is done by MCM. “Components of a generic business model can be viewed as a template for specific business models which factors out a set of assumptions that all specific business models derived from it will have in common while leaving open some aspects that are to be decided on a case by case basis”

The StreamCom System Design Specification\(^3\) describes one of several possible design choices that can be done based accordingly to StreamCom Business Analysis model. The presented here design corresponds to the implementation of the demonstrator done by programming partners. The StreamCom System Design Specification complements the Generic Component Model. While the Generic Components model specifies components for the StreamCom platform, the System Design model specifies the example of system behavior that can be built using such components.

Let’s also note that this report as well as the report provided by MCM does not specify all the technical details of implementations provided by StreamCom programming partners. Specification artifacts allow to understand the overall project ideas and to see how technical solutions from different partners can be integrated into one system.

This report was automatically was build from the UML model done in the Rational Rose case tool\(^4\). The structure of this report is the following: In section 2 of this report we show the specification of the StreamCom Business Analysis Model (Deliverable M.2 accordingly to the plan of the project). In section 3 of this report we show the implementation model (Deliverable M.7 accordingly to the plan of the project).

\(^3\) The System Design specification still can have some minor changes till the end of the project.
\(^4\) To get the Rational Rose model please send a request to Pavel Balabko: pavel.balabko@epfl.ch
2. **Business Analysis Spec**

The Business Analysis Specification represents the result of the analysis of several existing systems dealing with streaming of the information over the Internet and the environment of these systems. Here we show just only the resulting model that gives the business context for systems based on the StreamCom framework. For details on the analysis see [http://in3www.epfl.ch/~pbalabko/Projects/StreamCom/AsIsToBeModels.pdf](http://in3www.epfl.ch/~pbalabko/Projects/StreamCom/AsIsToBeModels.pdf)

2.1. **Collaboration Model**

![Collaboration Model Diagram](image-url)
Glossary

Ads Provider
An Ads Provider provides advertisement to Retailer in order to give it to Customer.

Stream Owner
A Stream Owner owns streams he wants to sell on the market.

Stream Publisher
A Stream Publisher prepares streams for reselling.

Monitor Company
A Monitoring Company observes all the traffic activity of selling streams.

Customer
A Customer is anyone who is interested in getting or previewing a Stream.

KeyServer
A Key Server works like a Digital Right Manager. It is responsible for the digital keys distribution used for encrypting and decrypting online streams.

Bandwidth Broker
A BandWidth Broker BB is responsible for the bandwidth reservation.

Content Server
A Content Server disseminates service to Customer by broadcasting encrypted message which is a part of a stream.

Retailer
A Retailer sells streams (provided by the Content Provider) to Customer.

Retailer Proxy
Retailer Proxy goal is to collects offers from different customers and matching them in an optimal way. This optimization results in SLA/OLAs messages send to a retailer.
2.2. **Collaboration Spec**

2.2.1. **Sell and Get a Service**

**Documentation:**

**Purpose:**
The Purpose of this collaboration is to sell a Service from a Retailer to a Member. The service can be provided in the form of CDs, DVDs or on-line streams.

**Participants:**
- One Member (Customer) who buys the stream
- One Retailer who sells the Stream
- Retailer Proxy who collects and optimizes customer requests.
- One Content Server who broadcasts the stream (in case of Online Stream)
- One KeyServer who sells keys for security (in case of Online Stream)
- One Bandwidth Broker who reserves for bandwidth requirement (in case of Online Stream)

**Pre-Conditions:**
- ServiceCatalog exists
- Credit card number exists

**Policies:**
- The Member should be provided with an on-line stream version of the service when he buys it on CD or DVD.
- Customer can get on-line service just on the specified computer (due to reservation)
- A Stream can include ads from the Ads provider
- The Bandwidth Broker should reserve required bandwidth for on-line stream.
- The Retailer has the right to start a bandwidth reservation
- Relationships between Retailer, Content Server, Key Server, Bandwidth Broker exist.

**Non-functional Requirements (Security Requirements):**
- Any exchange of information between Participants should be done in a secure way.
- Customer and Retailer should exchange money against ticket using fair-exchange protocol.
- Retailer and Content Server should be able to identify clients that using provided services illegally.

**Post-condition:**
- PC1. A Customer became a Member (for the first time only)
PC3. A Retailer got money from the Customer.
PC4. A Content Server, A Key Server and A Bandwidth Broker got money from the Retailer for co-operatively providing stream content to a customer (by means of redeeming tickets).
PC4. A Client got Service Level Agreement (SLA)
PC5. A Retailer Proxy got Operational Level Agreement (OLA)
PC6. A KeyServer got a ticket and micro-payments from the client.
PC7. A Retailer got confirmation of QoS with OLA.

Note: The stream can be copied to the user device like CD, or MP3 player (if the customer does not have any copy restrictions).
Sell and Get a Service Activity Diagram

Main Activities of Sell and Get a Service

- Start
  - [register for the first time]
  - [already a member]
  - Sign Up as a Member
    - Sell a Service
      - [modify]
      - [get service]
    - Modify Sell a Service
      - [get service]
      - [cancel to get service]
    - Get a Service
      - [get service]
      - [pay for service]
  - Payment
  - End
**Sell a Service**

**Documentation:**

**Purpose:**

The purpose of this collaboration is that a Member gets a Service (on-line stream Ticket or hard copy) of his choice from a Retailer and pays for it with a credit card. In the case of an on-line stream, the Member also gets a Ticket and a key that allows him to start the Video.

**Parameters:**

- Identification information (MemberId and Password, RetailerId, ContentServerId ...)
- Name of the service (video or TV channel)
- Service information
- Bandwidth require
- Payment information (ex: credit card number).

**Pre-Conditions:**

- ServiceCatalog exists
- Relationships between Retailer, Content Server, Key Server, Bandwidth Broker exist

**Post-Conditions:**

- A Customer became a member of Retailer (for the first time only)
- A Member got a Ticket and key
- A keyServer got a TicketID (or Ticket)
- A bandwidth reservation is done by Bandwidth Broker
- A Retailer got a reference to the Customer’s paying facilities (credit card, or address for issuing a bill)
- A customer got hard copy (not mandatory)

**Non-functional Requirements (Security Requirements):**

Any exchange of information between Participants should be done in a secure way. Customer and Retailer should exchange money against ticket using fair-exchange protocol.

**Basic Course of Events:**

2. [Client] Registers in the system.
3. [Retailer] Sends information (Mainly it is some advertisement information, some bonuses, special offers, based on the UserProfile).
4. [Member] Chooses a service title and type (on-line stream, DVD, CD) and specifies the number of minutes/views/copies. The Member can see the service description and preview if necessary.
5. [Client] sends requests SLA message to RetailerProxy.
6. [RetailerProxy] sends a OLR message to Retailer.
7. [Retailer] sends request for QoS accordingly to OLR to BB
9. [Retailer] sends the OLA message to RetailerProxy and price information to Client
10. [RetailerProxy] sends the SLA message to Client
11. [Client] Enters payment information (credit card number or address for issuing a bill).
12. [Retailer] create a Ticket and sends a Ticket to Customer.


**Actors in Sell a Service**

---

**Get a Service**

**Documentation:**

**Purpose:**

The Purpose of this UC is to get a Service bought in the "Buy a Service" UC and specified by the TicketId. A customer can get the service in several pieces that are broadcasted by the Content Server.

Note: A customer can get his Service (stream) in the "hard copy" (in form of CD, DVD etc). The ticket is not used in that case.

**Parameters:**

TicketId  
Ticket, Key

**Pre-Conditions:**

Customer had a Ticket and key  
Content server had an Online-Service  
Bandwidth between Content server and Customer is ensured

**Post-Conditions:**

Customer got/viewed an On-line Service  
KeyServer got Ticket and Micro-payments  
Bandwidth reservation is deleted
Non-functional Requirements (Security Requirements):
Any exchange of information between Participants should be done in a secure way. Retailer and Content Server should be able to identify clients that using provided services illegally.

Main Course of Events:
1. [Customer] Sends the Ticket to the KeyServer of Content Server
2. [Customer] Sends micro-payments to Keyserver
3. [KeyServer] Sends decryption key to Customer
4. [Customer] Receives and decrypts messages (content)
5. Repeats event 2 – 4

[1-5] Content Server broadcasts the content
[1-5] We ISP supports QoS for the Customer
**Actors in Get a Service**

Documentation:

```
+buyer
Member

Client

ClientSystem

ContentSystem

ContentClerk

KeyServerClerk

KeyServerSystem

BBClerk

BBSystem

+service provider

+BankwidthReservation

Get a Service

Pre-Conditions:

Key Server got Ticket and micro-payment tokens

Post-Conditions:

KeyServer got money from Retailer
ContentServer got money from Retailer
Bandwidth Broker got money from Retailer

Main Course of Events:

1. [KeyServer] Sends the Ticket identifier and the last received micro-payment token to Retailer.
2. [Retailer] Verifies the Ticket and the last micro-payment token

Payment

Purpose:

The Purpose of this UC is that the Retailer redeem money to KeyServer, Content Server and Bandwidth Broker

Parameters;

The last second half of micropayment token
3. [Retailer] Calculates redeem money correspondent to Ticket and the last half token
4. [Retailer] Pays for KeyServer, ContentServer, RetailerProxy and Bandwidth Broker.

**Actors in Payment**
“StreamCom General Architecture”: an Example of the Business Scenario

The following collaboration diagram (instance level) represents a possible scenarios based on “Sell a service”, “Get a Service” and “Payment” Collaborations. This typical scenario gives the general understanding of the main StreamCom Systems and the way how these system collaborate together.
The following collaboration diagram (specification level) specifies the main business systems, their relations and cardinalities.
3. **IT System Design**

The System Design in the StreamCom project (see Figure 2) represents the composition of 5 subsystems implemented by different programming partners. The specification of each subsystem includes the behavior (done with activity diagrams here) and state (done with class diagrams here) specifications.

![Figure 2 System in StreamCom Project](image-url)
Activity Diagram : Choose Service

1. Enter MemberID and Password
   - Send to Retailer
   - Receive MemberID and Password
   - Verify MemberID and Password

2. Enter Search Criteria
   - Send Searching request
   - Receive
   - Send result
   - Build Search Page (Information and Ads)

3. Choose a Service to view Details or to buy
   - Send Request
   - Receive Request
   - Receive
   - Send Request
   - Receive
   - Send

4. Choose type of Stream (Online/VCD/CD) and other information
   - Send Buy Request
   - Receive
   - Receive1
   - Start Bandwidth Reservation
   - Set Up Bandwidth Reservation
   - Receive info
   - Send info

5. In the first interaction we have
   - Start Payment and get Ticket
   - Receive and show
   - Send
   - Yes
   - Choose another Service
   - Described in Another Diagram

6. Choose and Buy Service
Activity Diagram: Buy Online Stream and Ticket Distribution

1. ClientSystem
   - Start
   - generateMicroPayment\[\{w[0],...,w[n]\}\]
   - sendMicroPayment\[w[0]\]
   - This ticket is not usable until User pay for the Service
   - saveTicket T va key k
   - start payment and enter credit card number
   - send
   - receive
   - receive and show Confirmation

2. RetailerSystem
   - receive
   - generateSessionKey k
   - generateTicket T
   - sendTicket and Key \{(T,k)\}
   - send Confirmation
   - end

RetailerSystem: ClientSystem
Conceptual Model : ClientSystem View-Point

Conceptual Model :
ClientSystem View-Point > Buy a Service

Glossary

BuyServiceParameters
Concept representing some information when buy a Service

On-line Service
Concept representing a Service in form of an Online Service

MySelf

User
Concept representing one User of the System, i.e. a Member

Retailer
A Retailer sells streams (provided by the Content Provider) to Customer

Ticket
Concept representing a ticket that is used to get an online Service

LoginTxn
Concept representing information regarding the action of a Customer login on the System
Service
  Concept representing any kind of Service
ExchangeTicketAgainstMoneyTxn
  Concept representing information regarding the action of buying Ticket for an online service
BuyServiceTxn
  Concept representing information regarding the action of buying Services
Conceptual Model : RetailerSystem View-Point

SellServiceParameter
Concept representing the different way a Service can be bought, i.e. payment per views, payment per time and payment per copy

ServiceCatalog
Concept representing all ServiceSpecs

SellServiceHistory
Concept representing all past Sell Service Transactions

Service
Concept representing any kind of Service

MySelf

On-line Service
Concept representing a Service in form of an Online Service

Bandwidth Broker
A BandWidth Broker BB is responsible for the bandwidth reservation

BandWidthReservation
Concept representing the bandwidth reserve for an online service

UserCatalog
Concept representing all User concepts

SellServiceTxn
Concept representing information regarding the action of selling one or several Services

Ticket
Concept representing a ticket that is used to get an online Service

KeyServer
A Key Server works like a Digital Right Manager. It is responsible for the digital keys distribution used for encrypting and decrypting online streams

BandWidthReservationTxn
Concept representing information regarding the action of reserving the bandwidth

User
Concept representing one User of the System, i.e. a Member

TicketDistributionTxn
Concept representing information regarding the action of distributing the Ticket
Conceptual Model: BBSystem View-Point

Conceptual Model Sell a service:
BandWidth Broker View-Point > Buy a Service > BandWidth Reservation

Glossary

Policy(RetailerList)
Concept representing all Retailer that has administrative permission to make reservation

ReservationList
Concept representing all BandWidth Reservation

Myself(BBSystem)

Retailer
A Retailer sells streams (provided by the Content Provider) to Customer

BandWidthReservation
Concept representing the bandwidth reserve for an on line service

Admission Control
Concept representing the action of setup Bandwidth Reservation
Conceptual Model : KeyServer View-Point

Conceptual Model :
KeyServer View-Point > Buy a Service > Get Ticket

Glossary

Retailer
A Retailer sells streams (provided by the Content Provider) to Customer

On-line Service
Concept representing a Service in form of an Online Service

GetTicketTxn
Concept representing information regarding the action of getting Ticket that is stand for money

MySelf

Ticket
Concept representing a ticket that is used to get an online Service

TicketCatalog
Concept representing all current Ticket of KeyServer
**Conceptual Model : Ticket**

**Conceptual Model for Ticket**

![Diagram showing relationships between KeyServer, Ticket, On-line Service, and Retailer]

**Glossary**

**KeyServer**
A Key Server works like a Digital Right Manager. It is responsible for the digital keys distribution used for encrypting and decrypting online streams.

**Retailer**
A Retailer sells streams (provided by the Content Provider) to Customer.

**Ticket**
Concept representing a ticket that is used to get an online Service.

**On-line Service**
Concept representing a Service in form of an Online Service.
3.1.1. Get a Service

Activity Diagram: Get a Service in particular

Get an Online Service in particular

- Client
- Keyserver
- Contentserver

1. **Send Ticket** T
2. **Receive** Ticket T
3. **Verify Ticket** T
   - **Store Ticket** T

4. **Receive**
5. **Decrypt** the session key $k[i]$
6. **Display**
7. **Receive**
8. **Decrypt** and show confirmation
9. **Receive**
10. **Send** confirmation encrypted by key $k[i]$
11. **Decrypt** the session key $k[i]$
12. **Receive**
13. **Send** $k[i]$ encrypted with $k[i]$
14. **Receive**
15. **Save** lastest token $w[i]$
16. **Encrypt** message with $k[i]$
17. **Broadcast** encrypted message to IP multicast
18. **Receive and start Redeem**

- **Yes**
- **No**

- **Get an Online Service in particular**
- **Request DecryptionKey** (keyID, Token w[i])
- **Verify Micropayment w[i]**
- **Generate SessionKey $k[i]$**
- **Save lastest token w[i]**
- **Receive**
- **Decrypt** to get key $k[i]$
- **Receive**
- **Send** $k[i]$ encrypted with $k[i]$
- **Receive**
- **Save** latest token $w[i]$
- **Encrypt** message with $k[i]$
- **Broadcast** encrypted message to IP multicast
Pattern of Get a Service

Glossary

Ticket
Concept representing a ticket that is used to get an online Service

MessageKey
Concept representing key to decrypt or encrypt a message that is a part of an Online Service

Message
Concept representing a message that is a part of an Online Service

ChangeTicketForKeyTxn
Concept representing information regarding the action of change Ticket for Key that is used to decrypt the online Service

Member
GetServiceTxn
    Concept representing information regarding the action of getting a message that is a part of an Online Service

Content Server
    A Content Server disseminates service to Customer by broadcasting encrypted message which is a part of a stream

GetKeyTxn
    Concept representing information regarding the action of getting key that is used to encrypt message before broadcasting

Bandwidth Broker
    A BandWidth Broker BB is responsible for the bandwidth reservation

BroadcastServiceTxn
    Concept represent information regarding the action of encrypting and broadcasting a message of an Online Service

KeyServer
    A Key Server works like a Digital Rigth Manager. It is responsible for the digital keys distribution used for encrypting and decrypting online streams

SellKeyTransaction
    Concept representing information regarding the action of selling key that is used to encrypt or decrypt a message of an online service
Conceptual Model : Client System View-Point

Conceptual Model : Get a Service
Client System View-Point > Get an Online Service

Glossary

Content Server
A Content Server disseminates service to Customer by broadcasting encrypted message which is a part of a stream

Message
Concept representing a message that is a part of an Online Service

On-line Service
Concept representing a Service in form of an Online Service

Key Server
A Key Server works like a Digital Right Manager. It is responsible for the digital keys distribution used for encrypting and decrypting online streams

Get Service Txn
Concept representing information regarding the action of getting a message that is a part of an Online Service
MessageKey
   Concept representing key to decrypt or encrypt a message that is a part of an Online Service

Ticket
   Concept representing a ticket that is used to get an online Service

ChangeTicketForKeyTxn
   Concept representing information regarding the action of change Ticket for Key that is used to decrypt the online Service

MySelf
Conceptual Model : ContentSystem View-Point

Conceptual Model : Broadcast Online Service
Content View-Point > Broadcast Online Service

MySelf
(from Actors)

On-line Service
(from Classes)

KeyServer
(from Classes)

GetKeyTxn
(from Classes)

BroadcastServiceTxn
(from Classes)

MessageKey
(from Classes)

Message
(from Classes)

User
(from Classes)

Glossary

MySelf

On-line Service
   Concept representing a Service in form of an Online Service

KeyServer
   A Key Server works like a Digital Right Manager. It is responsible for the digital
   keys distribution used for encrypting and decrypting online streams

User
   Concept representing one User of the System, i.e. a Member

GetKeyTxn
   Concept representing information regarding the action of getting key that is used
   to encrypt message before broadcasting

Message
   Concept representing a message that is a part of an Online Service

MessageKey
Concept representing key to decrypt or encrypt a message that is a part of an Online Service

**BroadcastServiceTxn**

Concept represent information regarding the action of encrypting and broadcasting a message of an Online Service
**Conceptual Model : KeyServer View-Point**

**Conceptual Model : Get a Service**
KeyServer View-Point > Get Online Service > Sell Keys

---

**Glossary**

**PublicKey**
Concept representing public key that is used to encrypt the message

**PrivateKey**
Concept representing public key that is used to encrypt the message

**User**
Concept representing one User of the System, i.e. a Member

**MessageKey**
Concept representing key to decrypt or encrypt a message that is a part of an Online Service

**Content Server**
A Content Server disseminates service to Customer by broadcasting encrypted message which is a part of a stream

**Ticket**
Concept representing a ticket that is used to get an online Service

**TicketCatalog**
Concept representing all current Ticket of KeyServer

**SellKeyTransaction**
Concept representing information regarding the action of selling key that is used to encrypt or decrypt a message of an online service

MySelf
Conceptual Model : BBSystem View-Point

Conceptual Model Get a Service : BBSystem View-Point

Glossary

**BandWidthReservation**

Concept representing the bandwidth reserve for an on line service

**ReservationList**

Concept representing all BandWidth Reservation

**Policy(RetailerList)**

Concept representing all Retailer that has administrative permission to make reservation
On-line Service
Concept representing a Service in form of an Online Service

Routing Protocol
Concept representing routing protocol using in a router

Entry

Myself(BBSystem)

Message
Concept representing a message that is a part of an Online Service

Routing Table

Router
3.1.2. Payment

Payment Collaboration

\[
M = Mr + Mc + Mk + Mb
\]

- **Member**
  - Charge money
  - Get money
  - Pay

- **Content Server**
  - Payment for Content
  - Get money

- **Retailer**
  - Payment for Key
  - Payment for Bandwidth

- **Key Server**
  - Payment for Key
  - Get money

- **Bandwidth Broker**
  - Payment for Bandwidth
  - Get money

\[
M = Mr + Mc + Mk + Mb
\]

- **Mr** = Money for Retailer
- **Mc** = Money for Content Server
- **Mk** = Money for Key Server
- **Mb** = Money for Bandwidth Broker
Payment

- Start
  - require redeem T, w[last]
    - send T, w[last]
      - receive
        - verify (T, w[last])
          - calculate money redeem
            - send redeem for KeyServer
            - send redeem for ContentServer
              - receive money
            - receive money
            - send redeem for Bandwidth Broker
              - receive money
            - receive money
4. IT System Implementation

4.1. Sell and Get a Service

4.1.1. Sell a Service

Sequence Diagram: Choose a Service

- Enter MemberID and password
- Verify
- Build Search page
- Enter Search Criteria
- Lookup and build preview
- View Details a service
- Lookup and build details
- Buy
- Reserve bandwidth
- Setup bandwidth reservation
- Calculate Price
Sequence Diagram: Buy a Service

Payment and Ticket distribution

ClientSystem

(ExchangeTicketAgainstMoneyTxn)

TicketDistributionTxn

Retailer

KeyServer

payment information

Client

BankSystem

(k,T)

check solvency(payment info)

save(ticketID)

confirmation

confirmation

n,w[0]

k=generateKey()

(k,T)

n,w[0]= generateMicroPayments()

Payment and Ticket distribution
Logical Design Class Diagram: Client System View-Point

Glossary

BuyServiceParameters
Concept representing some information when buy a Service

On-line Service
Concept representing a Service in form of an Online Service

MySelf

User
Concept representing one User of the System, i.e. a Member

Retailer
A Retailer sells streams (provided by the Content Provider) to Customer

Ticket
Concept representing a ticket that is used to get an online Service

**LoginTxn**
Concept representing information regarding the action of a Customer login on the System

**Service**
Concept representing any kind of Service

**ExchangeTicketAgainstMoneyTxn**
Concept representing information regarding the action of buying Ticket for an online service

**BuyServiceTxn**
Concept representing information regarding the action of buying Services
Logical Design Class Diagram: Retailer System View-Point

Glossary

SellServiceParameter
Concept representing the different way a Service can be bought, i.e. payment per views, payment per time and payment per copy

Ticket
Concept representing a ticket that is used to get an online Service

KeyServer
A Key Server works like a Digital Right Manager. It is responsible for the digital keys distribution used for encrypting and decrypting online streams.

**On-line Service**
Concept representing a Service in form of an Online Service

**Bandwidth Broker**
A BandWidth Broker BB is responsible for the bandwidth reservation

**ServiceCatalog**
Concept representing all ServiceSpecs

**UserCatalog**
Concept representing all User concepts

**TicketDistributionTxn**
Concept representing information regarding the action of distributing the Ticket

**SellServiceHistory**
Concept representing all past Sell Service Transactions

**Service**
Concept representing any kind of Service

**BandWidthReservationTxn**
Concept representing information regarding the action of reserving the bandWidth

**MySelf**

**User**
Concept representing one User of the System, i.e. a Member

**SellServiceTxn**
Concept representing information regarding the action of selling one or several Services
Logical Design Class Diagram : BBSystem View-Point

**Glossary**

**Policy(RetailerList)**
Concept representing all Retailer that has administrative permission to make reservation

**ReservationList**
Concept representing all BandWidth Reservation

**Myself(BBSystem)**

**Retailer**
A Retailer sells streams (provided by the Content Provider) to Customer

**BandWidthReservation**
Concept representing the bandwidth reserve for an on line service

**Admission Control**
Concept representing the action of setup Bandwidth Reservation
Logical Design Class Diagram: KeyServer View-Point

Glossary

Retailer
A Retailer sells streams (provided by the Content Provider) to Customer

On-line Service
Concept representing a Service in form of an Online Service

GetTicketTxn
Concept representing information regarding the action of getting Ticket that is stand for money

MySelf

Ticket
Concept representing a ticket that is used to get an online Service

TicketCatalog
Concept representing all current Ticket of KeyServer
Interfaces in Get a Service

Interfaces in Sell Service

Sell Service <Interface>
- getChosenServiceIds()
- addService()
- removeService()
- calculateTotalPrice()
- startExchangeTicket()
- addTransaction()
- addServiceId()

Choose Service (from Sell and Get a Service)
- getChosenIds()
- sendSearchingRequest()
- receiveAndShowSearchingRequest()
- sendViewDetails()
- receiveAndShowDetails()
- sendBuyingRequest()
- receiveAndShowPrice()
- startExchangTicketAgainstMoney()

buy Service <Interface>
- getChosenIds()
- sendSearchingRequest()
- receiveAndShowSearchingRequest()
- sendingViewDetails()
- receiveAndShowDetails()
- sendBuyingRequest()
- receiveAndShowPrice()
- startExchangTicketAgainstMoney()

Member (from Actors)

<<Actor>>

ExchangticketAgainstMoney <Interface>
- sendTicket()
- getTicket()
- createMicroPayment()

KeyServer
- keyServerID
- name

GetTicket (from Actors)

<<Actor>>

Ticket Distribution
- receiveMicroPayment()
- createKey()
- sendKey()
- receivePayment()
- receiveConfirm()
- sendConfirm()

Retailer
- retailerID
- name

<<Actor>>

Bandwidth Reservation
- flowDesc()
- checkStatus()
- setSrc()
- setDisc()
4.1.2. Get a Service

Sequence Diagram: Get a Service

Verifies:
- Identifier of Content Server
- Expiration time of the ticket
- Signature of the Retailer
- Ticket is not used

Client

ClientSystem

KeyServer

ContentServer

IPMulticast

StartVideo(ticketID)

sendTicket(T)

verifyTicket(T)

verifyMicroPayment(w[j])

getBroadCast[i]

createSessionKey()

encryptionKey

encrypt B[i]

decryptBroadcast(B[i])

display(B[i])

the last second-half

send last second-half

repeat many times

encrypted by key k

DecryptionKey

decrypt to get key k[j]

getBroadCast[i]

broadCast B[i]

broadCast B[i]

encrypted by key k

getDecryptionKey(keyId,w[j])

save(w[j])

createSessionKey()

encryptionKey

encrypt B[i]

decryptBroadcast(B[i])

display(B[i])

the last second-half

send last second-half

Verifies:
- Identifier of Content Server
- Expiration time of the ticket
- Signature of the Retailer
- Ticket is not used

Client

ClientSystem

KeyServer

ContentServer

IPMulticast
Logical Design Class Diagram: ClientSystem View-Point

Glossary

Content Server
A Content Server disseminates service to Customer by broadcasting encrypted message which is a part of a stream

Message
Concept representing a message that is a part of an Online Service

On-line Service
Concept representing a Service in form of an Online Service

KeyServer
A Key Server works like a Digital Right Manager. It is responsible for the digital keys distribution used for encrypting and decrypting online streams

GetServiceTxn
Concept representing information regarding the action of getting a message that is a part of an Online Service

**MessageKey**
Concept representing key to decrypt or encrypt a message that is a part of an Online Service

**Ticket**
Concept representing a ticket that is used to get an online Service

**ChangeTicketForKeyTxn**
Concept representing information regarding the action of change Ticket for Key that is used to decrypt the online Service

**MySelf**
Logical Design Class Diagram : ContentServer View-Point

Glossary

MySelf

On-line Service
Concept representing a Service in form of an Online Service

KeyServer
A Key Server works like a Digital Right Manager. It is responsible for the digital keys distribution used for encrypting and decrypting online streams

User
Concept representing one User of the System, i.e. a Member

GetKeyTxn
Concept representing information regarding the action of getting key that is used to encrypt message before broadcasting

Message
Concept representing a message that is a part of an Online Service
MessageKey
   Concept representing key to decrypt or encrypt a message that is a part of an Online Service

BroadcastServiceTxn
   Concept represent information regarding the action of encrypting and broadcasting a message of an Online Service
Logical Design Class Diagram : KeyServer View-Point

Logical Design Class Diagram : Sell Keys
KeyServer View-Point > Get Online Service > Sell Keys

Glossary

**PublicKey**
Concept representing public key that is used to encrypt the message

**PrivateKey**
Concept representing public key that is used to encrypt the message

**User**
Concept representing one User of the System, i.e. a Member

**MessageKey**
Concept representing key to decrypt or encrypt a message that is a part of an Online Service

**Content Server**
A Content Server disseminates service to Customer by broadcasting encrypted message which is a part of a stream

**Ticket**
Concept representing a ticket that is used to get an online Service

**TicketCatalog**
Concept representing all current Ticket of KeyServer

**SellKeyTransaction**
Concept representing information regarding the action of selling key that is used to encrypt or decrypt a message of an online service

MySelf
Interfaces in Get a Service

KeyServer

- keyServerID
- name

Distribute Keys
- getTicket()
- verifyTicket()
- saveTicket()
- getMicroPayment()
- verifyMicroPayment()
- saveLastestMicroPayment()
- sendSessionKey()
- getKey()
- decryptKey()

KeyDistribution

- changTiketfor Key
- sendTicket()
- receiveAndDecryptConfirm()
- requestSessionKey()
- sendSessionKey()

Content Server

- disseminate Service
- encryptMessage()
- broadCastMessage()

Member

- get Online
- requestService()
- getService()
- decryptService()
- displayService()
4.1.3. **Redeem**

**Redeem**

Sequence Diagram: Redeem process