DISTRIBUTED E-LEARNING USING THE RTF MIDDLEWARE

Hendrik Roreger

06.06.2011

Overview

- **Topic of the diploma thesis:**
  Design and implementation of a C++ middleware
  - Collaboration Centric Application Middleware (CCAM)
  - Based on the Real-time Framework (RTF)

- Requirements analysis of an E-Learning application
  - Work in conjunction with BMT Cordah
  - Analysed the requirements of BMT's edutain@grid demonstrator

- Generalization of specific requirements
  - Definition Collaboration Centric Application (CCA)

- Design of CCAM based on:
  - BMT's requirements
  - Definition of CCA
  - General conditions of the edutain@grid project

- Implementation based on RTF
  - Demonstrated by a basic E-Learning application

Real-time Framework

- Framework that enables high speed communication between clients
  - Automated serialization mechanism
  - Highly efficient communication protocol implementation over TCP/UDP optimized with respect to the low-latency and low-overhead requirements
  - Transparently redirect communication endpoints to a new resource

- API for using different parallelization approaches: zoning, instancing, replication for a scalable multi-server implementation of ROIA.
  - Automated distribution management, synchronization and parallelization of the update processing.
  - Transparent monitoring of common metrics that is used by the management and business layer of the edutain@grid system.

- Part of the EU funded edutain@grid middleware
  - Grid (Cloud) environment for Real-Time Online Interactive Applications (ROIA)
  - Business logic is layer between RTF and framework to enforce service level agreements

Motivation – E-Learning application

- Requirements Analysis at an edutain@grid Project Partner

- BMT Cordah Limited
  - Environmental consultancy, in particular marine safety consultancy
  - Development of Search and Rescue(SAR) planning tool
  - Offers SAR training courses to coastguards
    - General SAR courses
    - Courses about the SAR planning tool of BMT

- Demand for E-Learning on the Internet
  - Avoid travel expenses and increase staff availability
    - Global market
    - Courses are held at BMT's site or at the customer's site

Definition: Collaboration Centric Application (CCA)

- Application class
  - Enables a large group of users to collaborate on a certain topic
  - Users work with client applications to take part
  - Users Types
    - Session Creator
    - Session Attendee

- Session Creator
  - Create, alter or delete users
  - Create, prepare, start and delete sessions
  - Delegate interaction between users in a session

- Session Attendee
  - Join session
  - Experience session
  - Manipulate Session

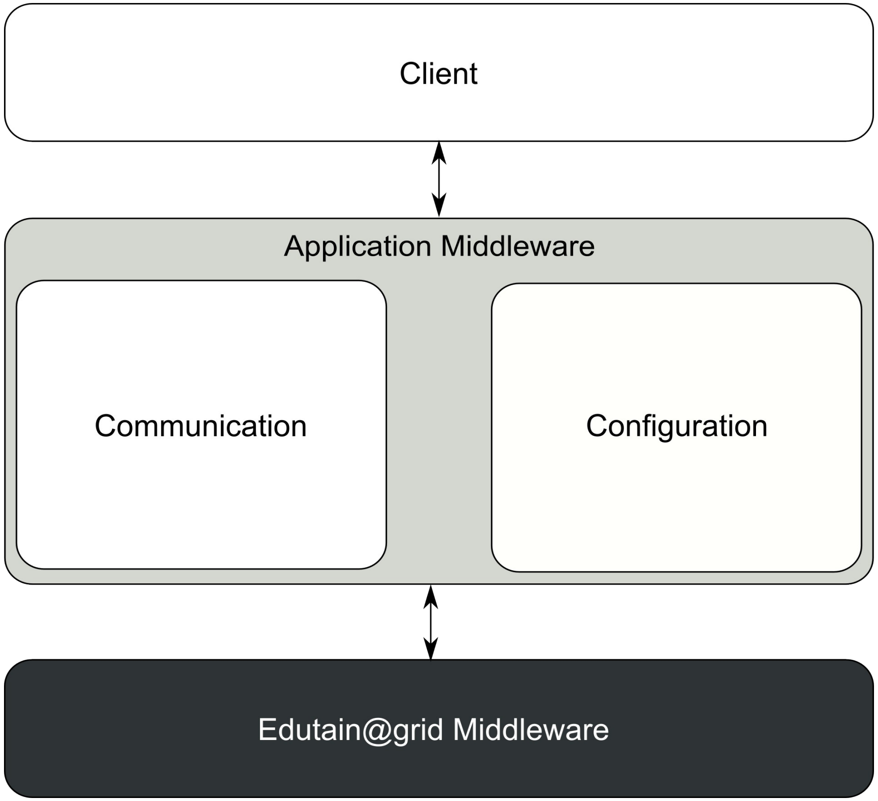
Requirements on CCAM

- Middleware which supports development of CCAs based on
  - Characteristics of CCAs
  - Technical platform: Real-time Framework (RTF)
- Comfortable API for CCA developer made possible by
  - Black-box principle
  - Division between configuration and communication functions
- Black-box principle
  - Collaboration Centric – ROIA Process (CC-ROIA-Process)
    - RTF requires Server process for each application build upon it
    - Therefore CC-ROIA-Process is generic process which is used for every CCAM based application
  - Distributed Attribute
    - Middleware API is mainly used by instantiating datatypes

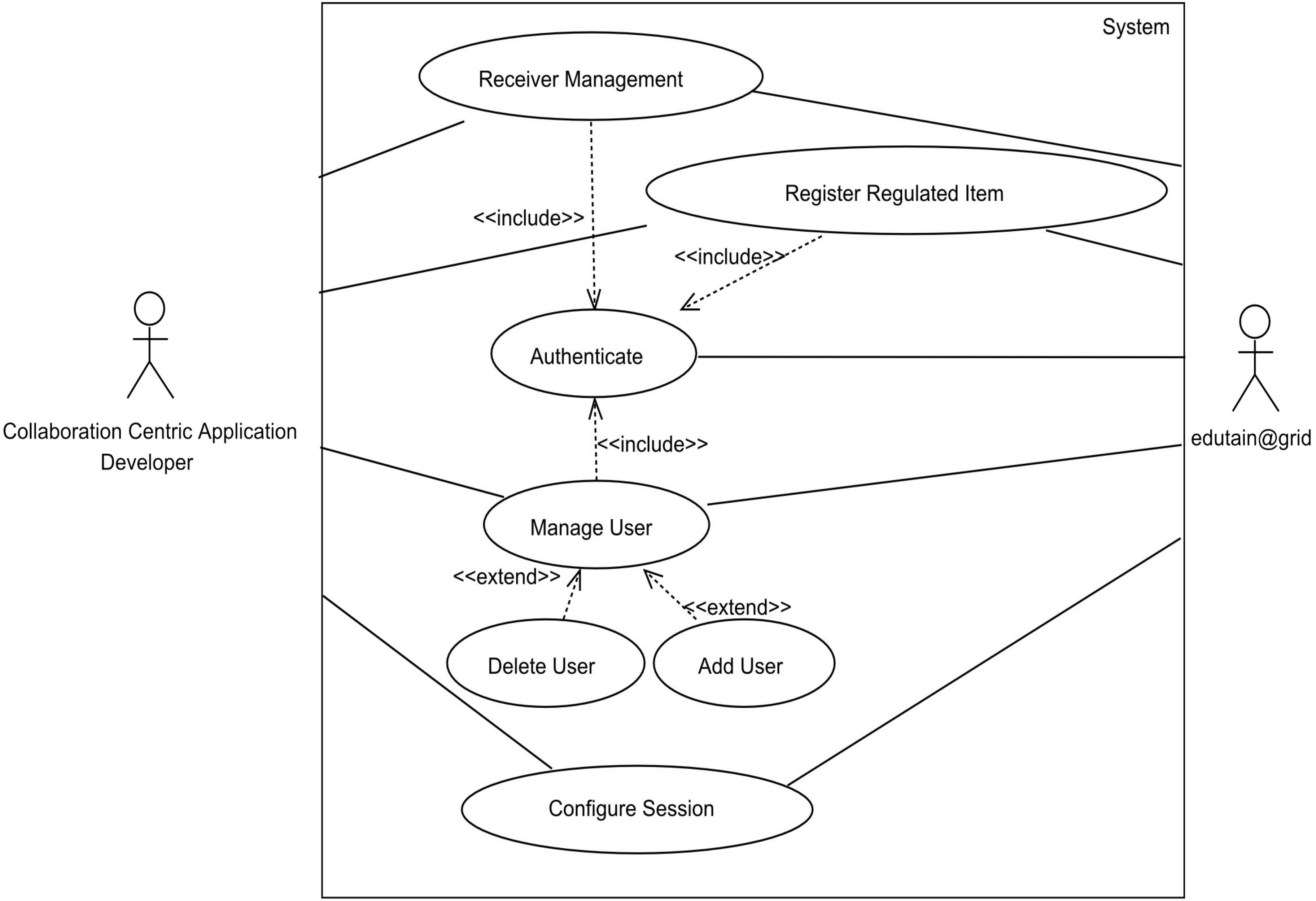
Division between configuration and communication

- Challenge: A developer has to ensure that required configuration is given in advance of using communication functions
  - Example: Start an E-Learning session
    - Client connected?
    - Authorised to connect?

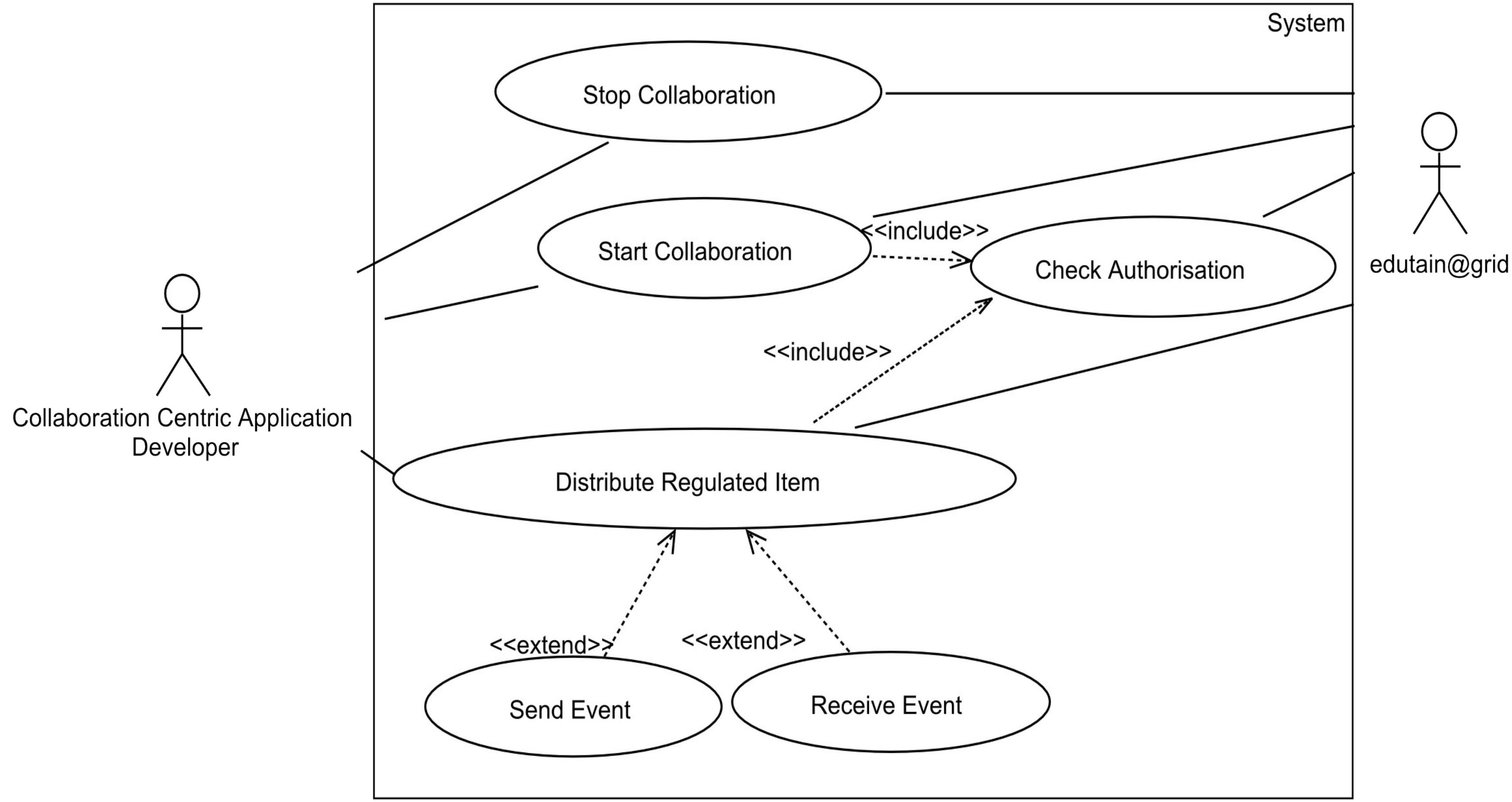
- Solution:
  - Configuration functions
    - XML configuration given in advance of execution
  - Communication functions
    - Runtime functions



Configuration Use Cases



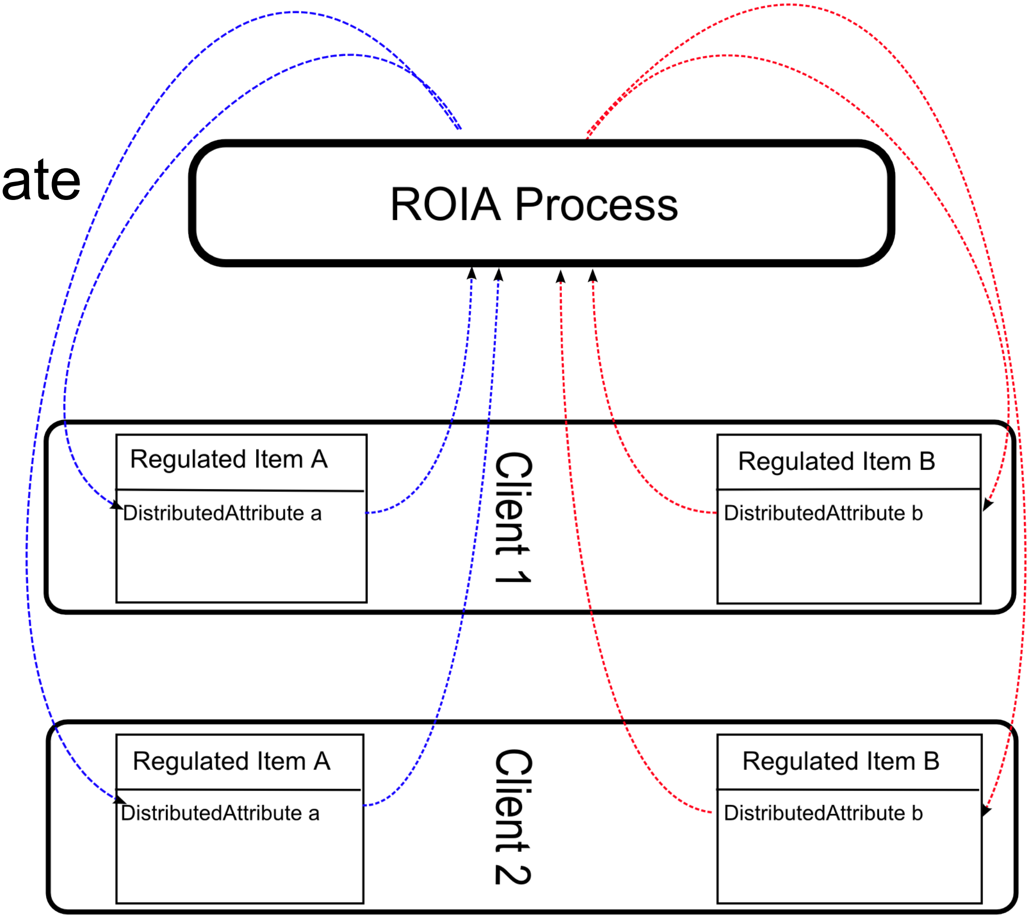
Communication Use Cases



- Subject of collaboration: Regulated Item
  - Abstract objects which are the subject of a collaboration session
- Challenge: Developer has to integrate Regulated Items into the application logic of the CCA

Solution: Distributed Attributes

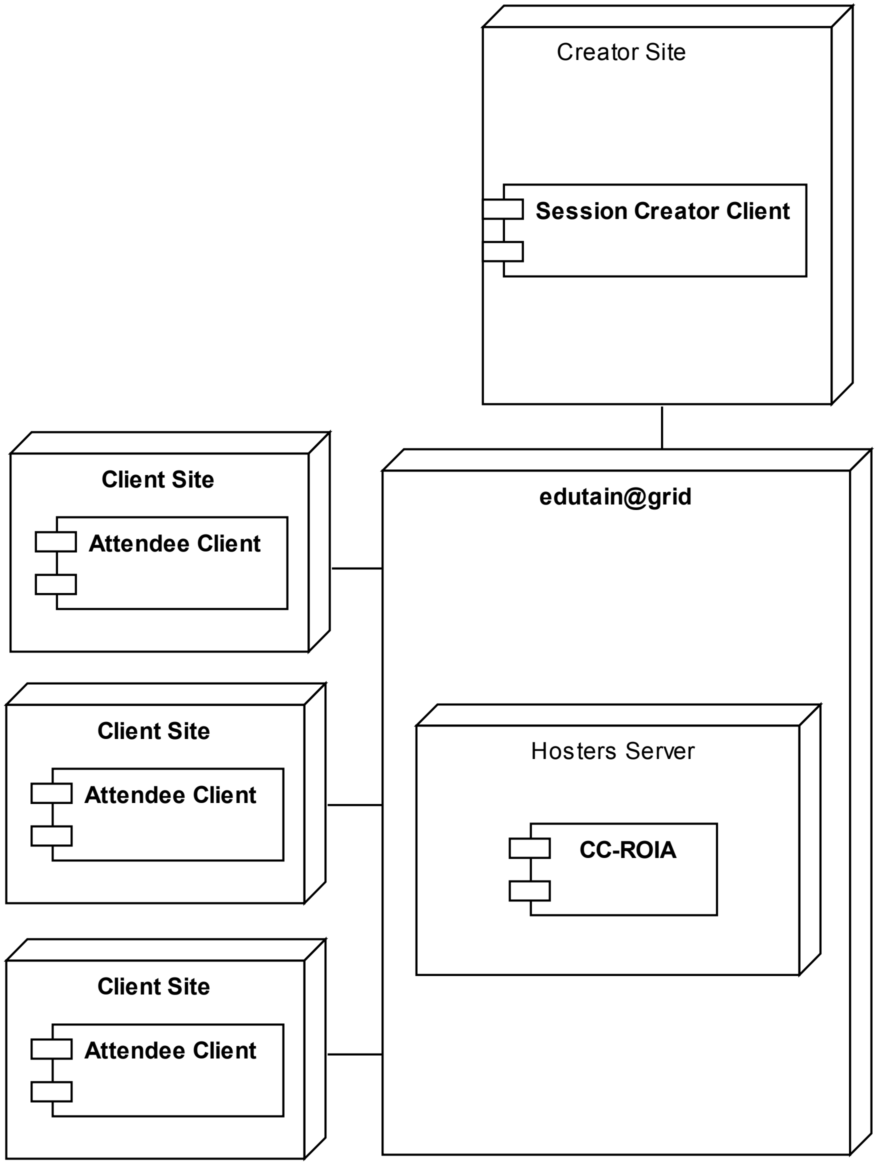
- Concrete Regulated Items
  - Ordinary C++ objects
  - Members having a distributed state
    - **Distributed Attribute**



- Simple datatypes
  - Distribution implemented by using RTF
- Global State
  - ID is assigned to each Distributed Attribute
  - Each Distributed Attribute on each client identified by the same ID is in the same state
- State transmission
  - RTF state and message transmission mechanism

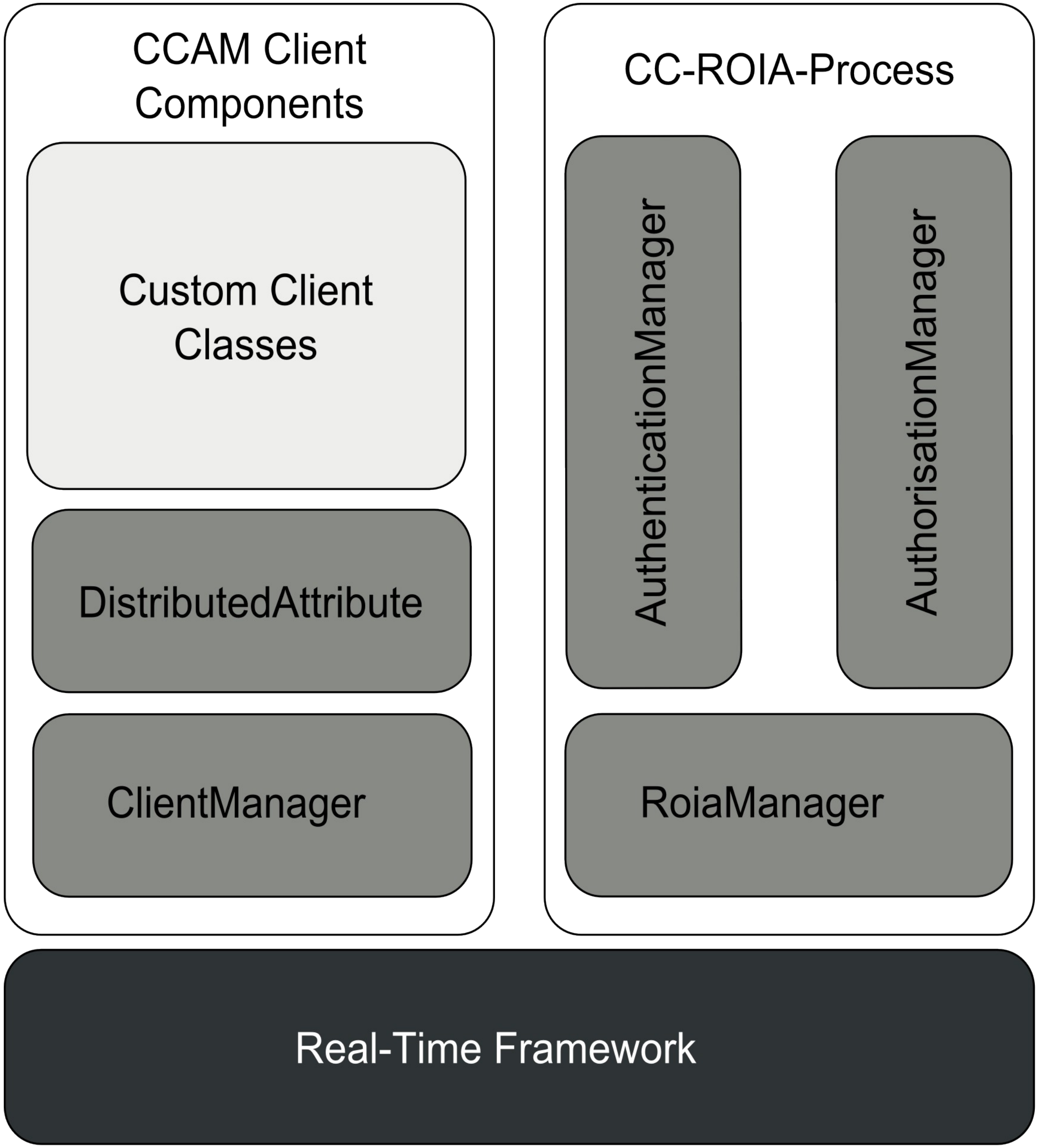
CCA Deployment

- Client Application(s)
  - Implementation based on CCAM
  - Session Creator and Attendee Client can be the same binary
    - Configuration through XML
- CC–ROIA-Process
  - Client applications connection through the Internet
  - Controls and Distributes the state of each Distributed Attribute



CCAM Components

- Client Components
  - DistributedAttributes
  - ClientManager
    - Client connection and management tasks
- CC-ROIA-Process
  - RoiaManager
    - Handles client connections
    - Distributes states
  - AuthenticationManager
    - Decides, whether clients are allowed to connect to the CC-ROIA-Process or not
  - AuthorisationManager
    - Controls, whether message sent by client is granted to be received by RoiaManager or not



Distributed Attributes

- Abstract Superclass: DistributedAttribute

```
class DistributedAttribute {
public:
void registerUpdateListener
    (DistributedAttributeUpdateListener* listener);
void allowAccess(std::string username);
void denyAccess(std::string username);
```

- Authorisation Interface
  - allow- or denyAccess grants or revokes the right of a user to alter a DistributedAttribute

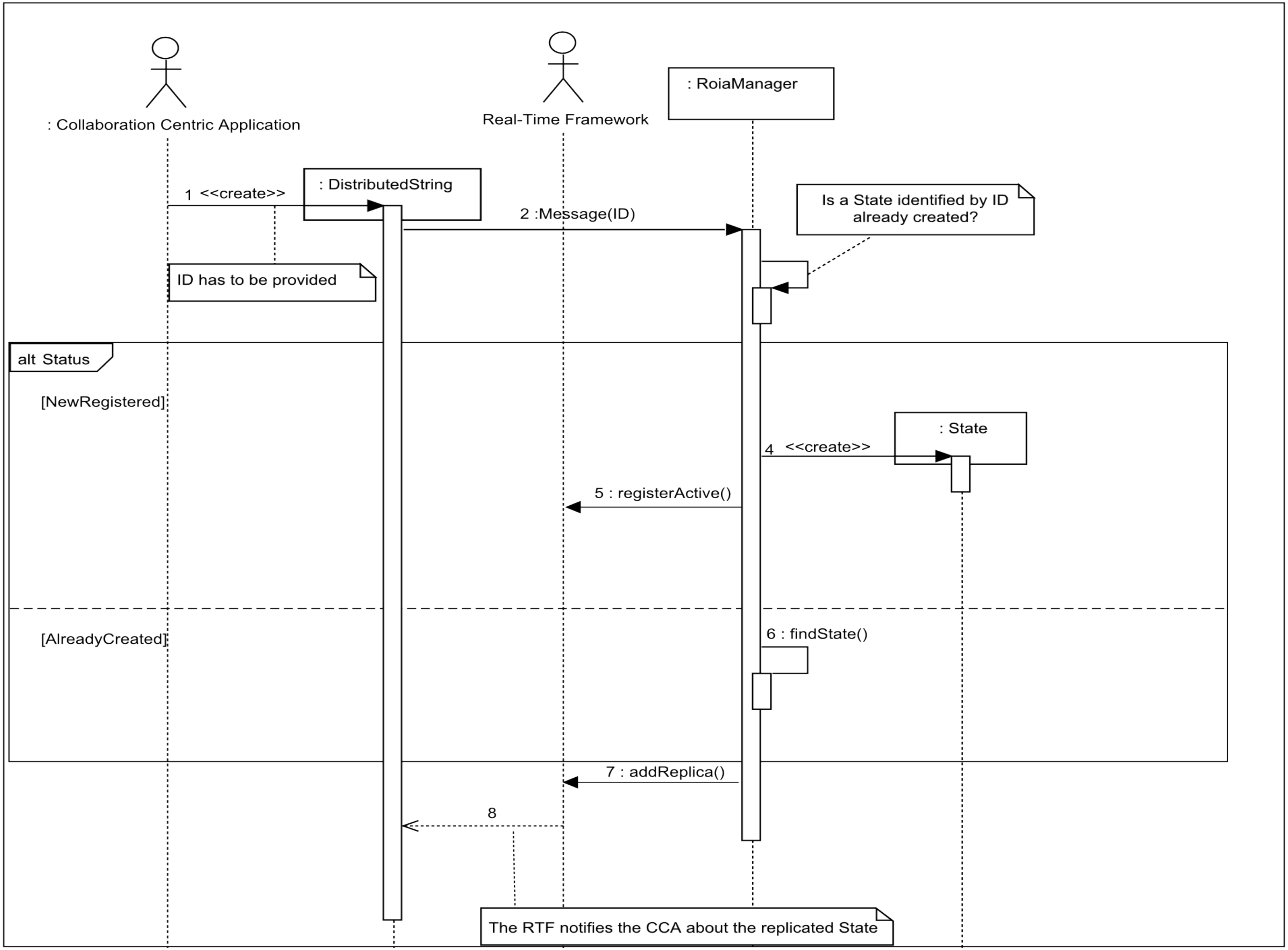
- Update Listener
  - registerUpdateListener enables to register an update listener object
  - A onUpdate method is called each time the state of a DistributedAttribute changes

Distributed String

- Distributed character string similar to an STL string
- Derived from DistributedAttribute
- Interface of a DistributedString

```
class DistributedString: public DistributedAttribute
{
public:
DistributedString(gcf_uint32 id, std::string init ="" );
DistributedString(std::string newval = "");
virtual ~DistributedString();
std::string get();
void set(std::string newval);
```

Distributed String



Distributed Vector

- DistributedVector
  - Dynamic datatype
    - Each element contained is contained at each instance identified by the same ID
  - Usage similar to a C++ std::vector

- Interface of a DistributedVector

```
class DistributedVector : public DistributedAttribute
{
public:
DistributedVector(gcf_uint32 id);
virtual ~DistributedVector();
void push_back(DistributedAttribute& newElement);
std::vector<DistributedAttribute*>::iterator begin();
std::vector<DistributedAttribute*>::iterator end();
```

- Contains DistributedAttributes
  - Therefore the state of the elements are automatically distributed

XML Configuration

- Example – Session Creator configuration file

```
<ccam:client serverhost="localhost" serverport="25000">
<ccam:credential name="instructor"
   password="instructorpassword" />
<ccam:deployment>
<ccam:credential name="student"
   password="studentpassword" />
</ccam:deployment>
</ccam:client>
```

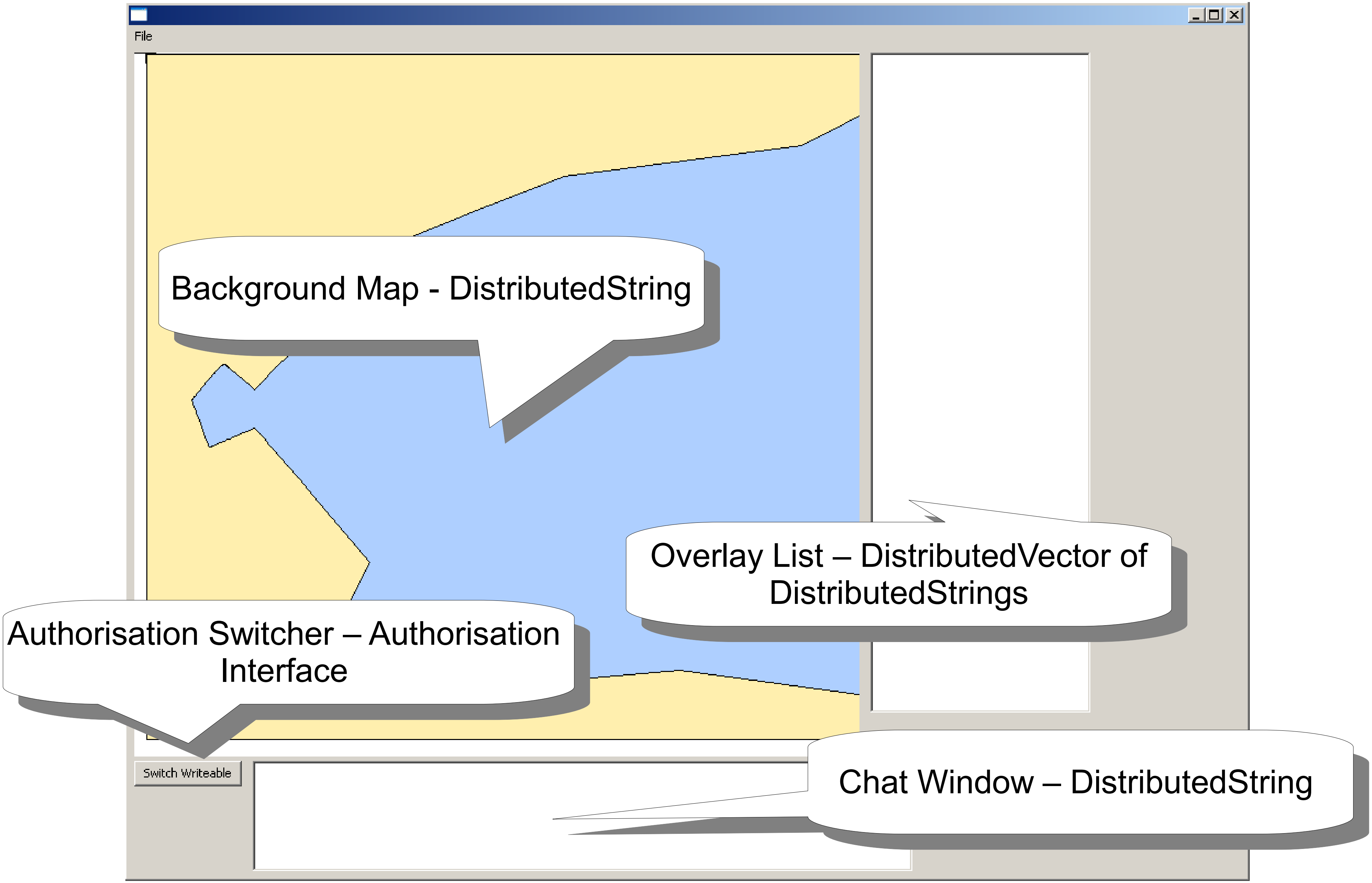
- Example – Attendee configuration file

```
<ccam:client serverhost="localhost" serverport="25000">
<ccam:credential name="student"
   password="studentpassword" />
</ccam:client>
```

- Example – CC-ROIA-Process configuration file

```
<ccam:config username="instructor"password="instructorpassword" >
```

CCA Demo



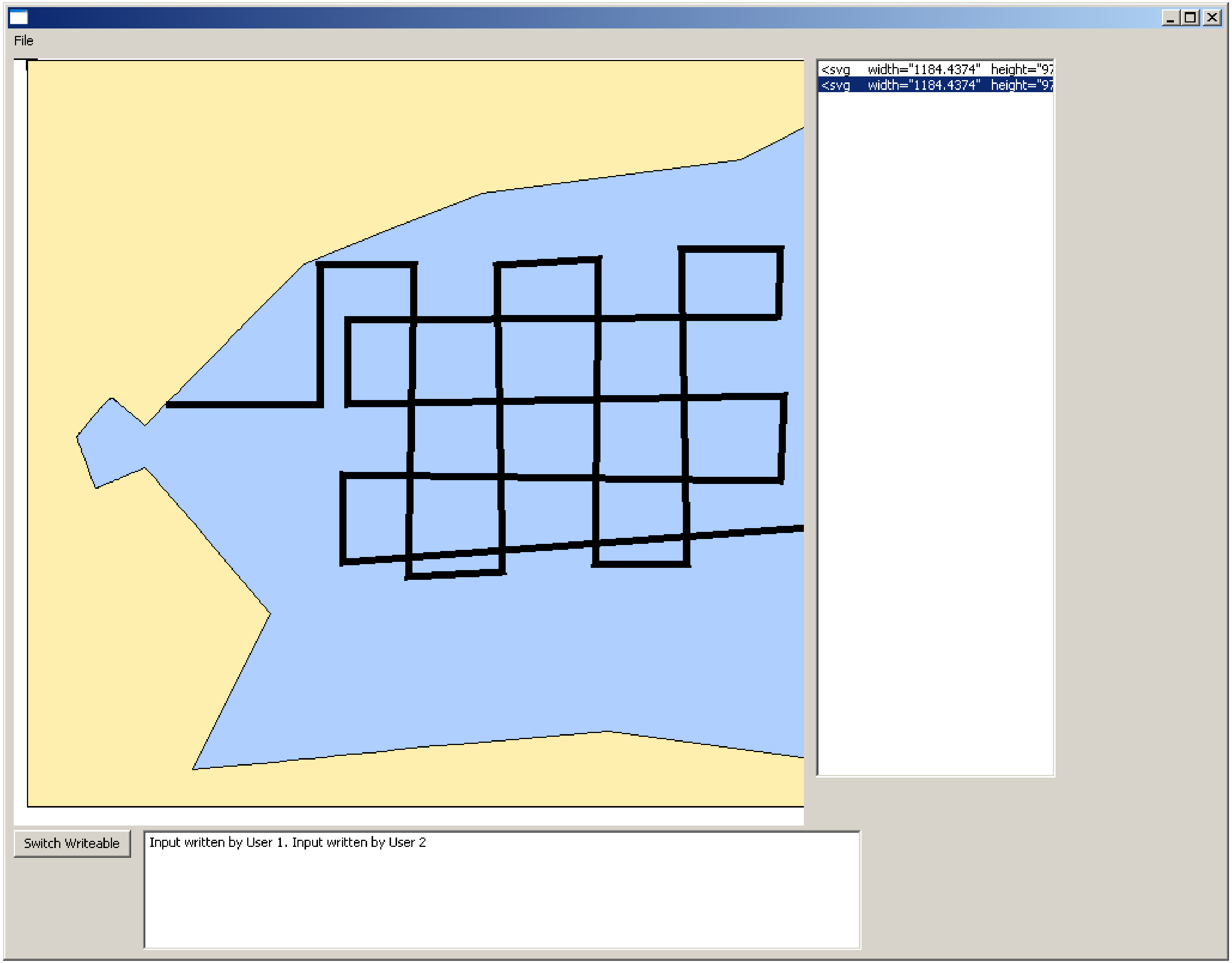
Background Map - DistributedString

Overlay List – DistributedVector of DistributedStrings

Authorisation Switcher – Authorisation Interface

Chat Window – DistributedString

CCA Demo



```
<svg    width="1184.4374"   height="97
<svg    width="1184.4374"   height="97
```

Input written by User 1. Input written by User 2

Summary

- Middleware which supports implementation of CCAs
- Implementation based on Real-time Framework
- Demonstrator shows implementation of an E-Learning system with CCAM
- BMT Cordah Limited uses CCAM in its edutain@grid E-Learning application
  - Positive feedback from BMT's developers

Comparison to Related Work

- Judith Rodríguez-Estévez, Manuel Caeiro-Rodríguez, and Juan M. Santos-Gago:Standardization in Computer Based Learning
  - *Runtime Environment*: “The basic tasks of runtime environments are to deliver contents to the learner, to support the interaction between content and learner, and the decision of the next educational resource ‘to deliver’ depending on the static and dynamic structure of the course and on the prior actions of the student”
- RTF enables implementation of a Runtime System
  - Comparison with similar Runtime Systems is senseful

Comparison AVANTE

- Runtime System implemented via CORBA: (Víctor Theoktisto, Adelaide Bianchini, Edna Ruckhaus, and Lee Lima: AVANTE: A Web Based Instruction Architecture based on XML/XSL Standards, Free Software and Distributed CORBA Components)
  - Contains complete environment for course management, user authentication, collaborative work
  - Predefined Services are provided for clients
  - Clients are Web sites accessing services via JSP

- Comparison with RTF and edutain@grid
  - AVANTE defines classes from that courses have to be derived: Powerful predefined components <-> RTF's DistributedAttributes generic: Less powerful, more flexible
  - CORBA: ORB has to be set up, resources have to be locally available <-> edutain@grid: infrastructure can be accessed on demand, ressources are rented, therefore more cost efficient

Comparison E-Learning Grids

- “Towards E-Learning Grids: Using Computing in Electronic Learning” (Pankratius and Vossen)
  - “Proposes Grid computing in the context of e-learning”
  - For example, compute photo-realistic visualizations in real-time and display the computing result
  - Approach: E-Learning Grid that allows this kind of computing by accessing a *grid learning object* (GLOB) via Web-Services

- Comparison with RTF and edutain@grid
  - RTF e-learning addition is focused on distributing states in an easy manner <-> E-Learning Grids is focused on “resource hungry” e-learning scenarios
  - E-Learning Grid is proposed for organizations like Universities, which have a lot of computing power that is not used (e.g cip pool) <-> focuses on e-learning provider that don't have a powerful infrastructure

Vielen Dank für Ihre Aufmerksamkeit! Fragen?