

OMWeb - Virtual Web-based Remote Library for Modelica in Engineering Courses

Presenter: Mohsen Torabzadeh-Tari

Slides: Zoheb Muhhamed Hossain

PELAB, Linköping University, Sweden

OMWeb - Introduction

OMWeb is a Web interface for OpenModelica where the students can "right away" start coding from a web browser; sparing them from downloading and installing the OMC compiler on their system.

Benefits

- Code from any where
- Students' focus more on learning than setting up the system
- Gets to learn the language even if they have a very low-end system

OMWeb Architecture

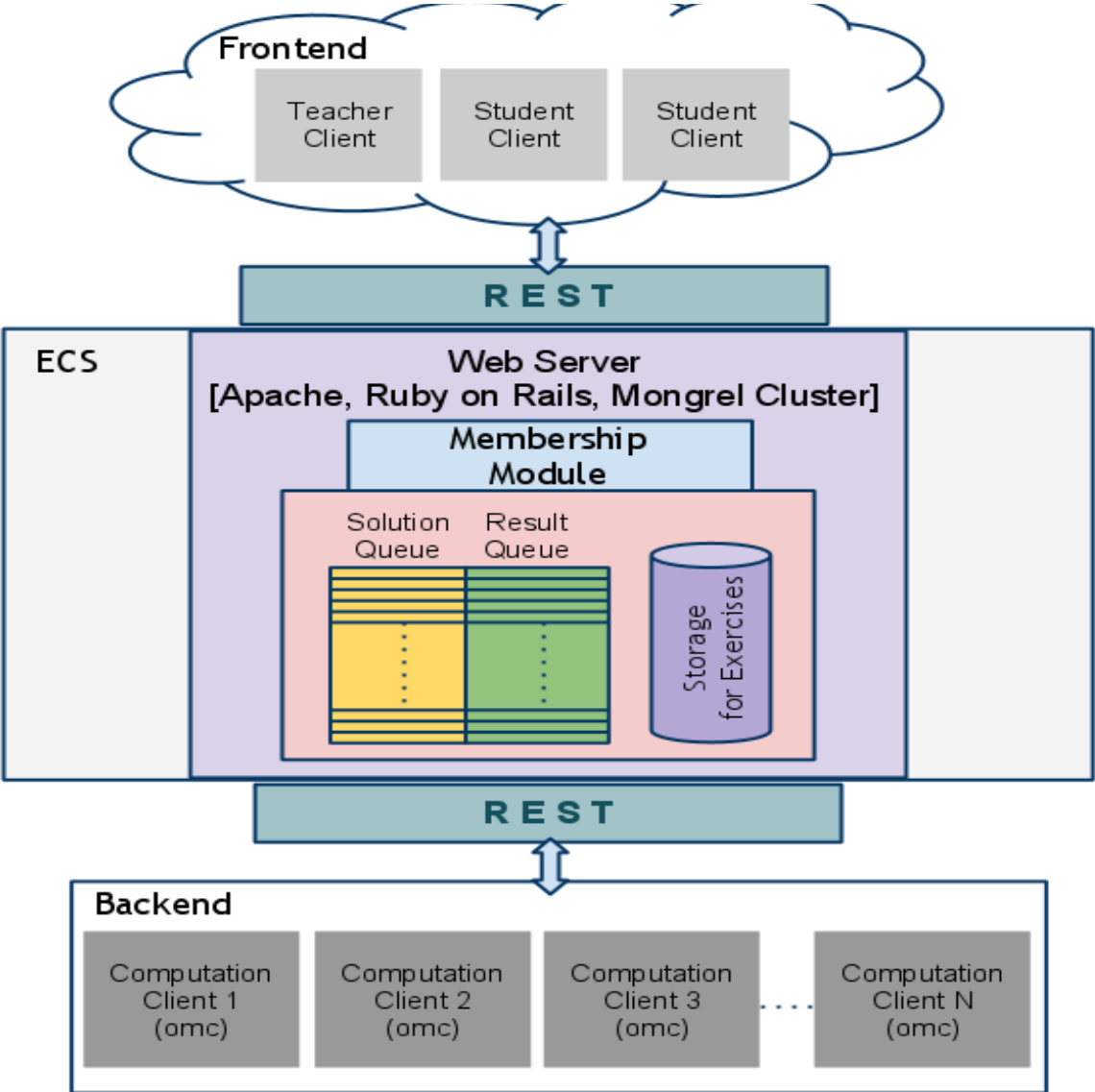
The architecture consists of 3-layers

- **The Student Clients and/or Teacher Clients**
 - Java applet based GUI interface
- **The E-learning Community Server middle-ware**
 - Developed in Ruby on Rails
- **The Computation Client(s)**
 - Developed in C++

Interfacing between Clients and ECS

- Representational State Transfer, REST standard over HTTP
 - Communication Methods
 - GET, POST, UPDATE and DELETE
 - Carrier data type
 - JavaScript Object Notation, JSON string

OMWeb Architecture - Illustration



OMWeb - Teacher Client

Applet

new Exercise Generator Delete selected Tab

Numlab Exercise Generator

← 🖨️ 📁 🖨️ 🌐 ▶️ 📧 📦 →

How many visible elements? Links to Resources?

EXERCISE

PostTime

Name

Description

Student's code content


```
model Pendulum "Planar Pendulum"  
  
    constant Real PI=3.141592653589793;  
    parameter Real m=1, g=9.81, L=0.5;  
    Real F;  
    output Real x(start=0.5),y(start=0);  
    output Real vx,vy;
```











Choose an exercise now

OMWeb - Student Client

Applet

Numlab

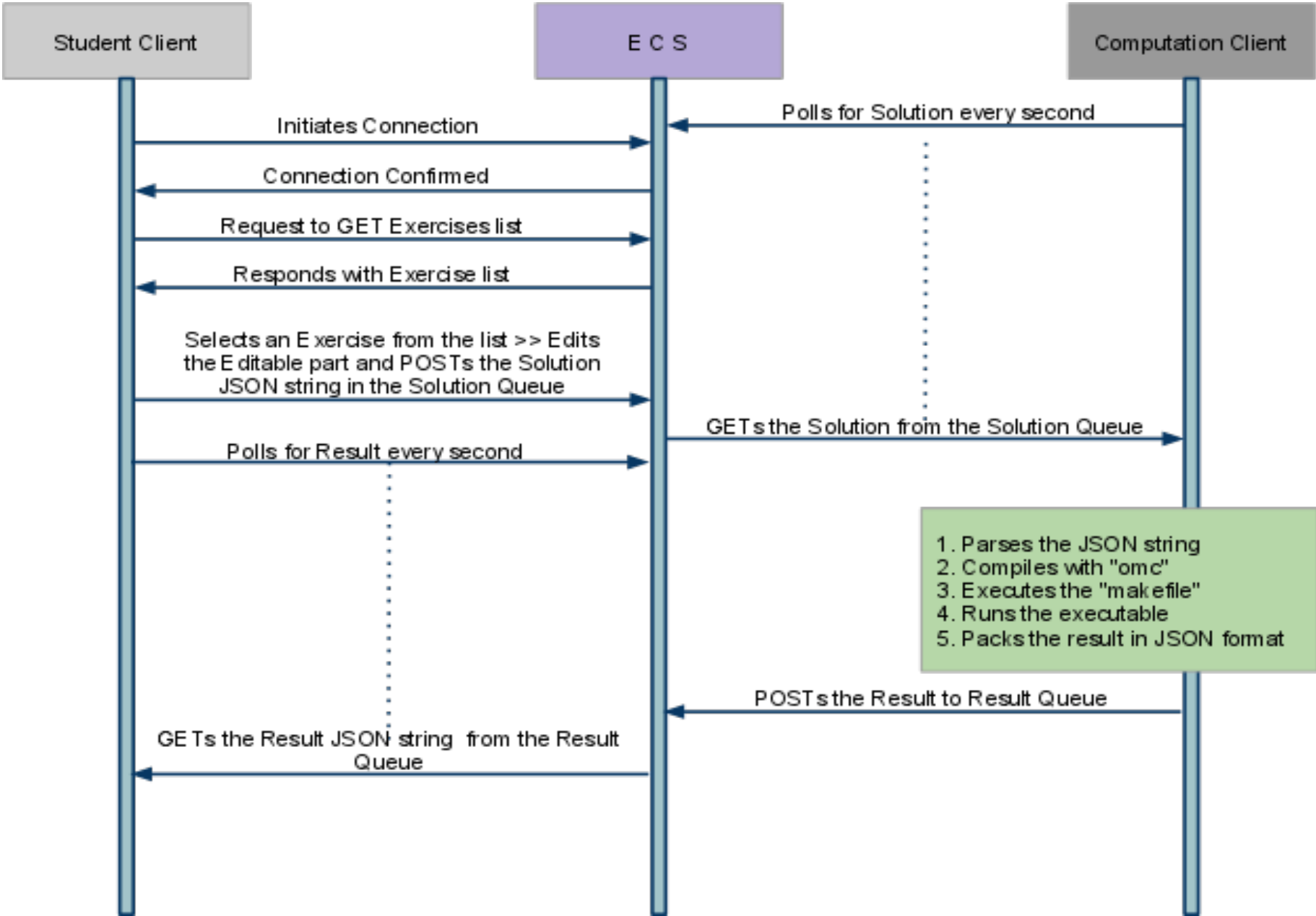
Exercises 

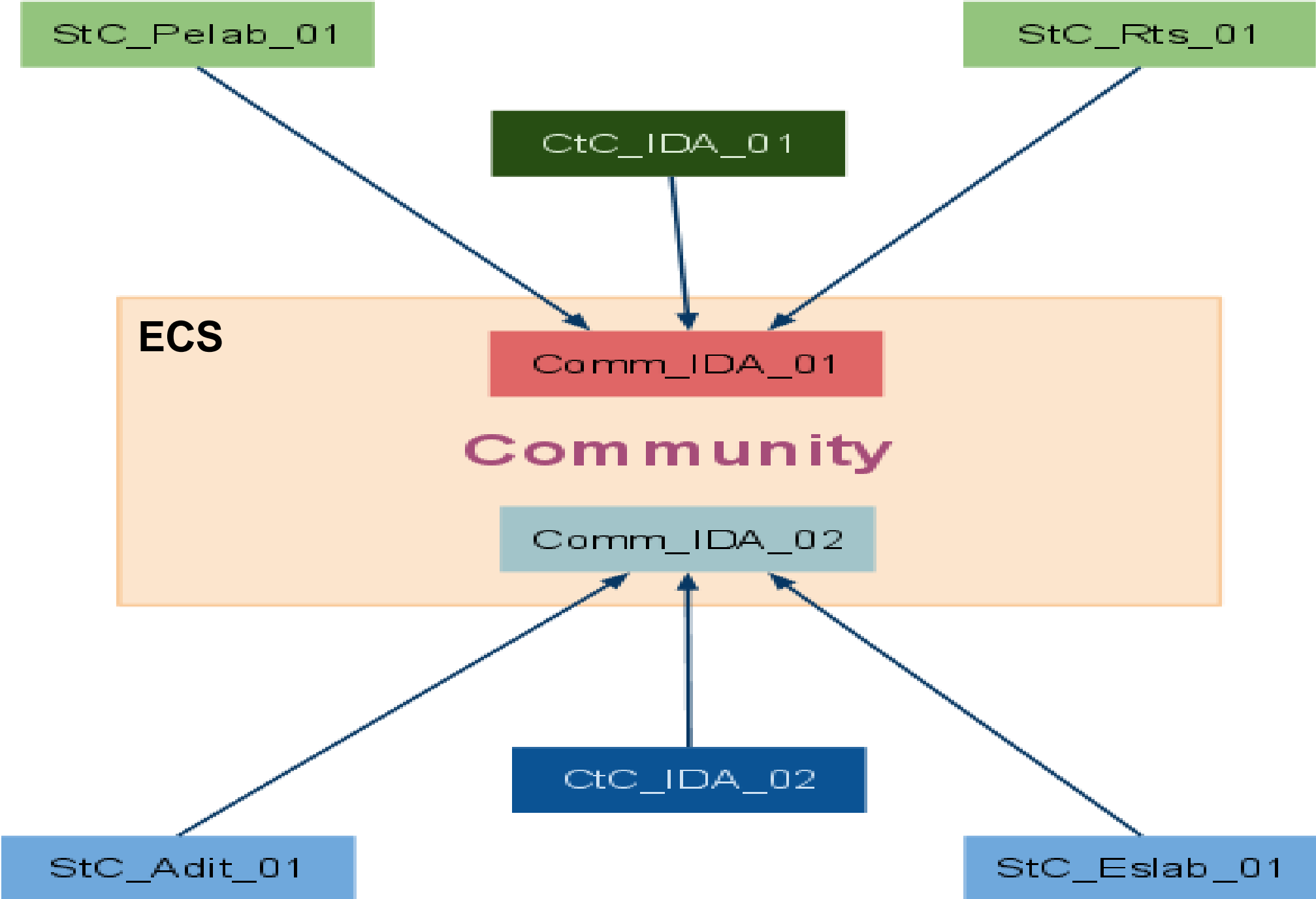
```
model Pendulum "Planar Pendulum"  
  
    constant Real PI=3.141592653589793;  
    parameter Real m=1, g=9.81, L=0.5;  
    Real F;  
    output Real x(start=0.5),y(start=0);  
    output Real vx,vy;  
  
equation  
  
    m*der(vx)=-x/L*F;  
    m*der(vy)=-y/L*F-m*g;  
  
    der(x)=vx;  
    der(y)=vy;  
    x^2+y^2=L^2;  
  
end Pendulum;
```

Modelic:

OMWeb - Message Flow



OMWeb - ECS



OMWeb - Analysis

Why ECS?

- No typical Client-Server connection
 - No Socket creation
 - No State saving
- Computation Client is dedicated for **computation** tasks only
 - No need to maintain any connected session with ECS
- Both the Student and Computation Clients are independent of each other, in terms of the languages they were developed with
 - Portability
- Event driven FIFO Queues for storing and processing JSON strings(Solution Queue and Result Queue)
- Simple RESTful interface eases the method of communication

OMWeb - Analysis

Computation Client (courtesy of the Stuttgart University)

- Sandbox mechanism
 - Secure
 - Limits the user accessibility in the system
- Stuttgart University's implementation
 - Matlab
- Linköping University's implementation
 - OpenModelica
 - Functional Languages (future work)
- Interfacing with the ECS
 - REST standard (GET, POST, UPDATE, DELETE)

OMWeb - Analysis

Student Client

- Support for textual GUI
- Future work
 - Drag and Drop model icons and/or diagram
 - Implement Modelica Annotations
- Interfacing with the ECS
 - REST standard (GET, POST, UPDATE, DELETE)

OMWeb - Future Work

Student Client

- 3D Plotting
- Modularize to support other programming languages

ECS

- Increase the data transfer limit, >2MB per message

Computation Client

- Extend it to support more programming languages, eg. Schema

