CSC7321 Middleware and software architecture for distributed applications

Revision : 513

Chantal Taconet

September 2020
Presentation of CSC7321

1. Administrative information
2. Objectives
3. Prerequisites
4. Plan of this TU
5. Evaluation
6. MicroProject
7. Questions
Administrative Information

- TU Coordinator: Chantal Taconet
  mailto:chantal.taconet@telecom-sudparis.eu

- TU teachers:
  - Chantal Taconet mailto:chantal.taconet@telecom-sudparis.eu
  - Sophie Chabridon mailto:sophie.chabridon@telecom-sudparis.eu
  - Denis Conan mailto:denis.conan@telecom-sudparis.eu

- TU resources:

- Remote conditions
  - The classes will be accessible remotely via BigBlueButton
  - Software should be installed in your own computer
Objectives of this teaching unit

- Be aware of different software techniques for designing *distributed applications*
  - Name and describe the main *interaction patterns* (synchronous request, publish/subscribe) between distributed software components
  - Learn master technologies for producing enterprise distributed applications: Web Services (REST), JavaEE, RabbitMQ
  - Design the *architecture of a multi-component distributed application* made of several functional modules with computing components, persistent components, client components.
  - Learn responses to architectural concerns (*scalability, interoperability, security*)
- Design and implement one distributed applications through one micro-project
Prerequisites for this Teaching Unit

- Labs on Unix OS
- Object oriented programming and modeling (with UML diagrams)
- Implementation in the Java language
- Integrated development environment such as Eclipse IDE
- Relational databases
Middleware for synchronous requests (illustrated with REST Web Services)

Introduction to software architecture and quality attributes (scalability, interoperability, security)
Organisation of this teaching unit II

- Component-based middleware with JavaEE (persistent components)
Organisation of this teaching unit III

- Publish subscribe pattern and Distributed Event Based Systems (illustrated with AMQP RabbitMQ)
Plan of this TU

Big Picture

- Structural Compositions
  - SCA
  - Activity Orchestrations
    - BPEL

- Application servers
  - Life cycle (instantiate)
  - Persistency

- Publish/Subscribe
  - RabbitMQ

- WebServices/JavaRMI
  - Synchronous Call

- sockets
  - TCP/UDP
Evaluation

- Study and presentation of an article (3/10)
  - Slides and oral presentation

- Labs and intermediary deliverables (1/10)

- Micro Project (6/10)
  - Design and architectural choices
  - Implementation in java
  - Slides and final defense
Subject: realize a bike tourism application (for olympics 2024)

- Administrators define bike tours (e.g. From Musée Grévin to Les Catacombes)
- Group of tourists select a tour among available ones
- Group of tourists exchange and visualize their positions
- The system verify bike availabilities all around the tour
Use Case Diagram — management of tours and POIs

A travel agency that acts as an operator of the system can prepare some tours, etc. on behalf of future clients.
Use Case Diagram — management of group of participants

- create a group and join it
- join a group
- leave a group
- remove a group

**Tourist**

**VLibTour**

- The creation of group is performed by one participant, he becomes the first member to join the group.
- The action is performed automatically in these cases:
  - after a timeout (e.g. 1h) with no action from the participants
  - all participants arrived to the last POI
  - all the participants have left the group

High priority use case are in green
Use Case Diagram — management of locations

- Tourist
  - publish location
  - notify the location of a participant
  - remove subscription to location information
  - subscribe to location information

- VLibTour
  - To receive the location of the other participants, the actor agree to give their location periodically

high priority use case are in green
Use Case Diagram — management of visits

VLibTour

Tourist

- get current position
- get the position of the next POI
- search for the arrival bike station
- step to the next position in current path towards the next POI
- step to next POI in current visit

high priority use case are in green
Use Case Diagram — statistics of visits

VLibTour

- get the total number of groups on a period
- get the average group size
- get the number of visits
- get the most popular tour

VlibTour

Operator
Software Architecture of the system

These two components are co-located if they use the same RabbitMQ broker. If so, the lobby room system creates the group communication systems on demand (one per group).
Micro Project modalities

■ Important dates
- Subject of the project: today
- Implement parts of the microproject during the labs
- Project defense: Exam week (mid November)

■ Results
- Original implementation
- Report (6-10 pages)
- Defense: slides and demo (1/2 hour)
Questions