Presentation of the module
CSC7321 Middleware for
distributed systems

Chantal Taconet

CCN/CSC7321
octobre 2019

Revision : 513
1 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:
  ◆ moodle access (http://moodle.imtbs-tsp.eu/) TSP : CSC7321 : Middleware for distributed systems
  ◆ public access (http://www-inf.telecom-sudparis.eu/COURS/CSC7321/)
2 Objectives of this teaching unit

- Be aware of different software techniques for designing **distributed applications**
  - Name and describe the main **interaction patterns** (synchronous call, callbacks, orchestration, asynchronous calls, 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
3 Prerequisites for this Teaching Unit

- Labs on Unix OS
- Object oriented programming and modeling (with UML diagrams)
- Implementation in the Java language
- Eclipse IDE (integrated development environment)
- Relational databases
4 Organisation of this teaching unit

- Component-based middleware with JavaEE (persistent components)

- Middleware for synchronous requests (illustrated with REST Web Services)

- Introduction to software architecture and quality attributes (scalability, interoperability, security)

- Publish subscribe pattern and Distributed Event Based Systems (illustrated with AMQP RabbitMQ)
5 Big Picture

Structural Compositions

- SCA

Activity Orchestrations

- BPEL

Application servers

- Life cycle (instantiate)
- Persistency

JavaEE

Publish/Subscribe

- RabbitMQ

WebServices/JavaRMI

Synchronous Call

sockets

TCP/UDP

| Télécom SudParis — INF — octobre 2019 — CCN/CSC7321 | 6/15 |
6 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 oral presentation
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
# 7.1 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.

A travel agency that acts as an operator of the system can prepare some tours, etc. on behalf of future clients.

### Use Case Diagram

- **Tourist**
  - get a POI
  - list the set of tours
  - get a POI
  - add a POI to a tour
  - move a POI in the sequence of a tour
  - remove a POI from a tour

- **Operator**
  - list the set of tours
  - get a tour
  - get a POI
  - create a tour
  - add a POI to a tour
  - move a POI in the sequence of a tour
  - remove a POI from a tour
  - create a POI
  - modify the description of a POI
  - remove a POI
### 7.2 Use Case Diagram — management of group of participants

**Tourist**

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

**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 lefted the group

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

- **Tourist**
  - subscribe to location information
  - remove subscription to location information
  - publish location
  - notify the location of a participant
  - high priority use case are in green

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

**Notes:**
- High priority use case are in green.
### 7.4 Use Case Diagram — management of visits

**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

**VLibTour**

*High priority use cases are in green*
7.5 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).
7.6 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)
8 Questions