
Service-Oriented Architecture

Software Engineering

—

Cyril ANAK STELL
Linn MJELSTAD
Clovis OUEDRAOGO

Josué ALVAREZ
Aminata DIOP
Cécile DUTHOIT

—

Table of contents

- I. SOAP and REST: advantages and context of use
- II. Integration of our SOA project in our integrative project
- III. Our implementation
 - A. Our BPEL module: import data from a CSV file to post in our app database
 - B. Our Android application: REST implementation
 - C. Our app server: REST implementation
 - D. Deployment
- IV. Project management
- V. Conclusion

I. SOAP and REST: advantages and context of use



	SOAP	REST
Meaning	Simple Object Access Protocol	Representational State Transfer
Preferred format	XML	JSON
Can be cached	NO	YES
Supports SSL	YES	YES

**Supports
point to point
identity only**

I. SOAP and REST: advantages and context of use



Supports
identity through
intermediaries

Reliable delivering
between distributed
applications (even if
software component,
system, or network
failures)

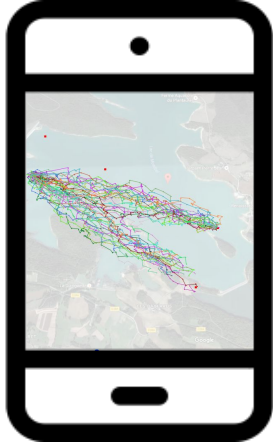
Atomiticy
Consistency
Isolation
Durability

	SOAP	REST
Supports identity through intermediaries	YES (WS-Security)	NO (only SSL)
Standard messaging system	YES (WS-ReliableMessaging)	NO
ACID compliant	YES (WS-AtomicTransaction)	NO (limited by HTTP)
Can be used for banking transactions	YES (3 features above are required)	Should not

II. Integration of our SOA project in our integrative project

IoTTracking *GPS tracking for boat regattas*

Live trajectory



Web client

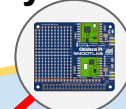
Web
Server



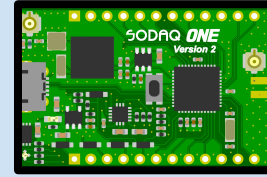
Lora
Server



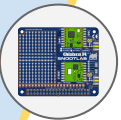
Gateway



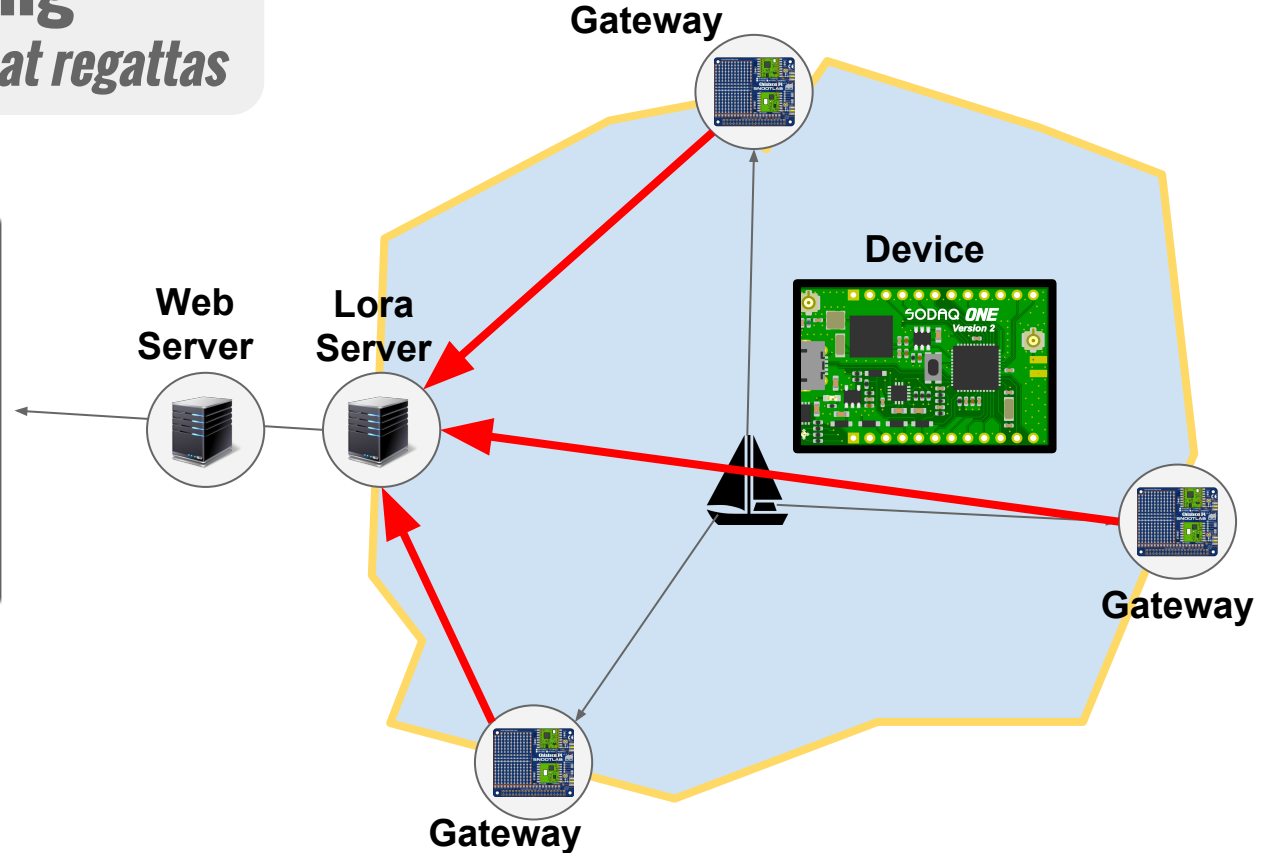
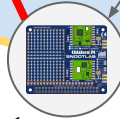
Device



Gateway



Gateway

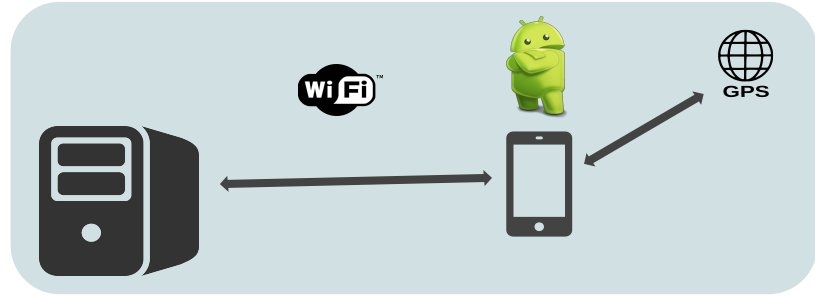


II. Integration of our SOA project in our integrative project

Our needs:

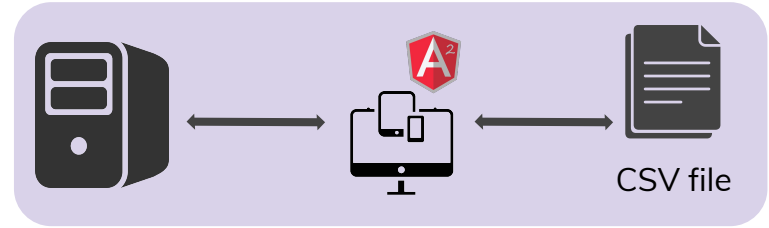
1. Import GPS positions

⇒ set races trajectory



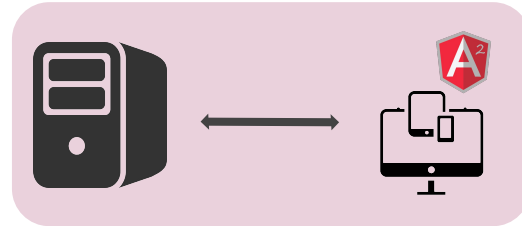
2. Import data from CSV file

⇒ get racers id and boat id from Freg

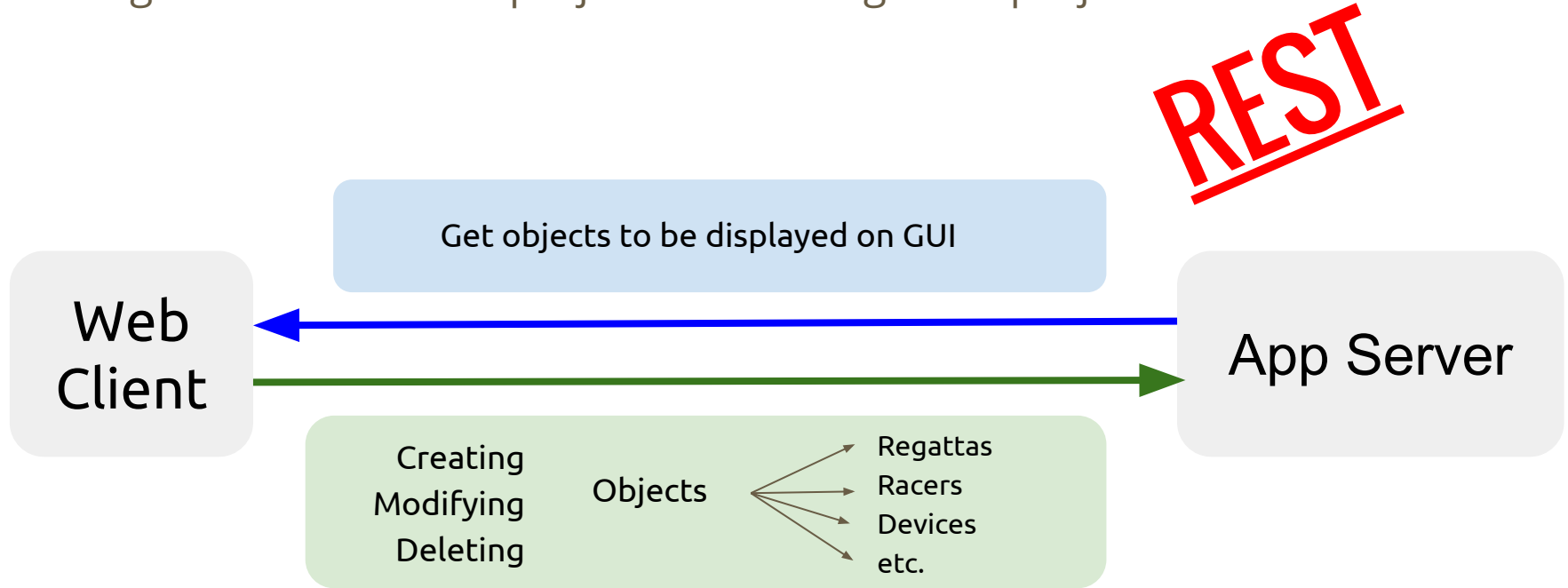


3. Communicate between app and server

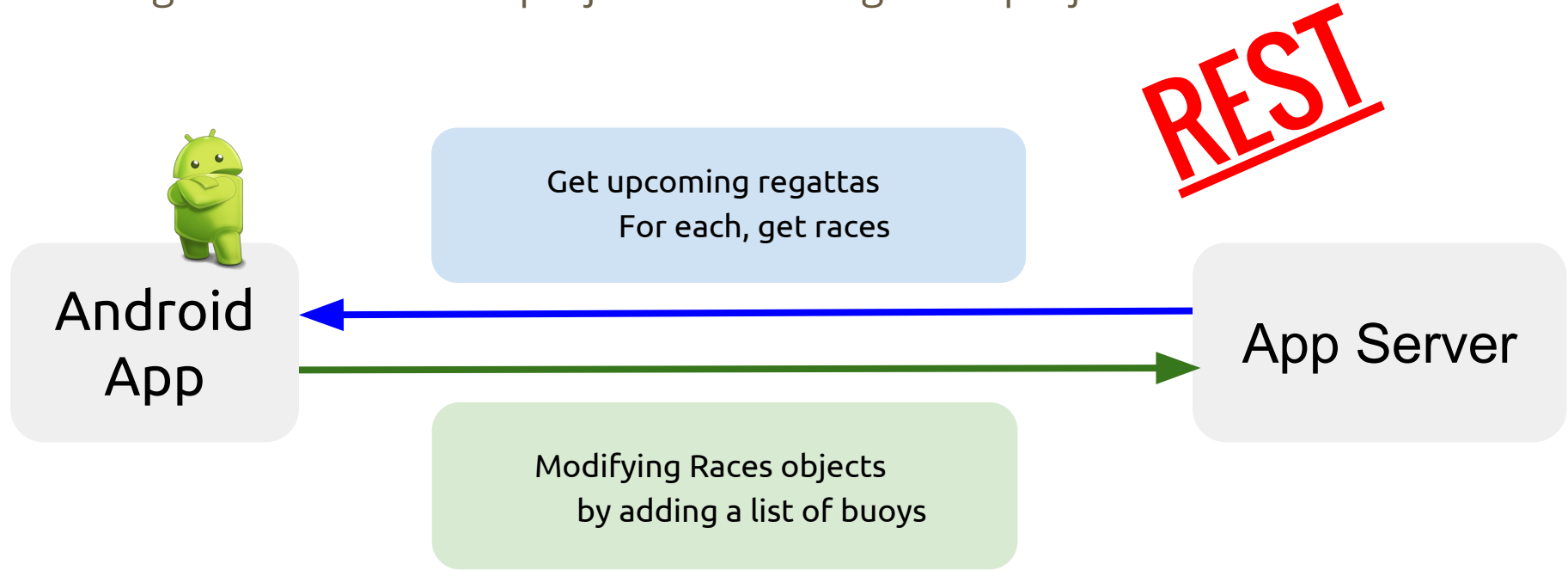
⇒ modify objects in the database
and get them from the database



II. Integration of our SOA project in our integrative project

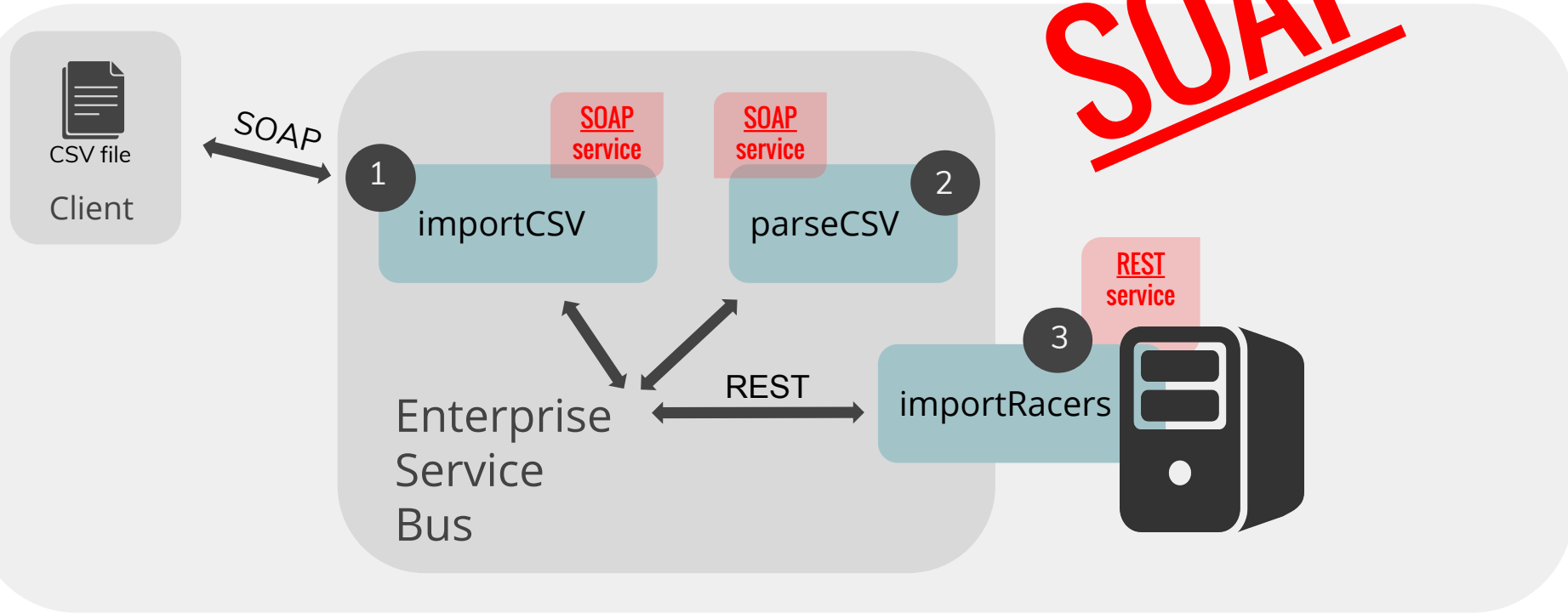


II. Integration of our SOA project in our integrative project



II. Integration of our SOA project in our integrative project

What we planned to do...



II. Integration of our SOA project in our integrative project

What we actually did

REST

CSV file



Upload
CSV file

Parse
CSV file

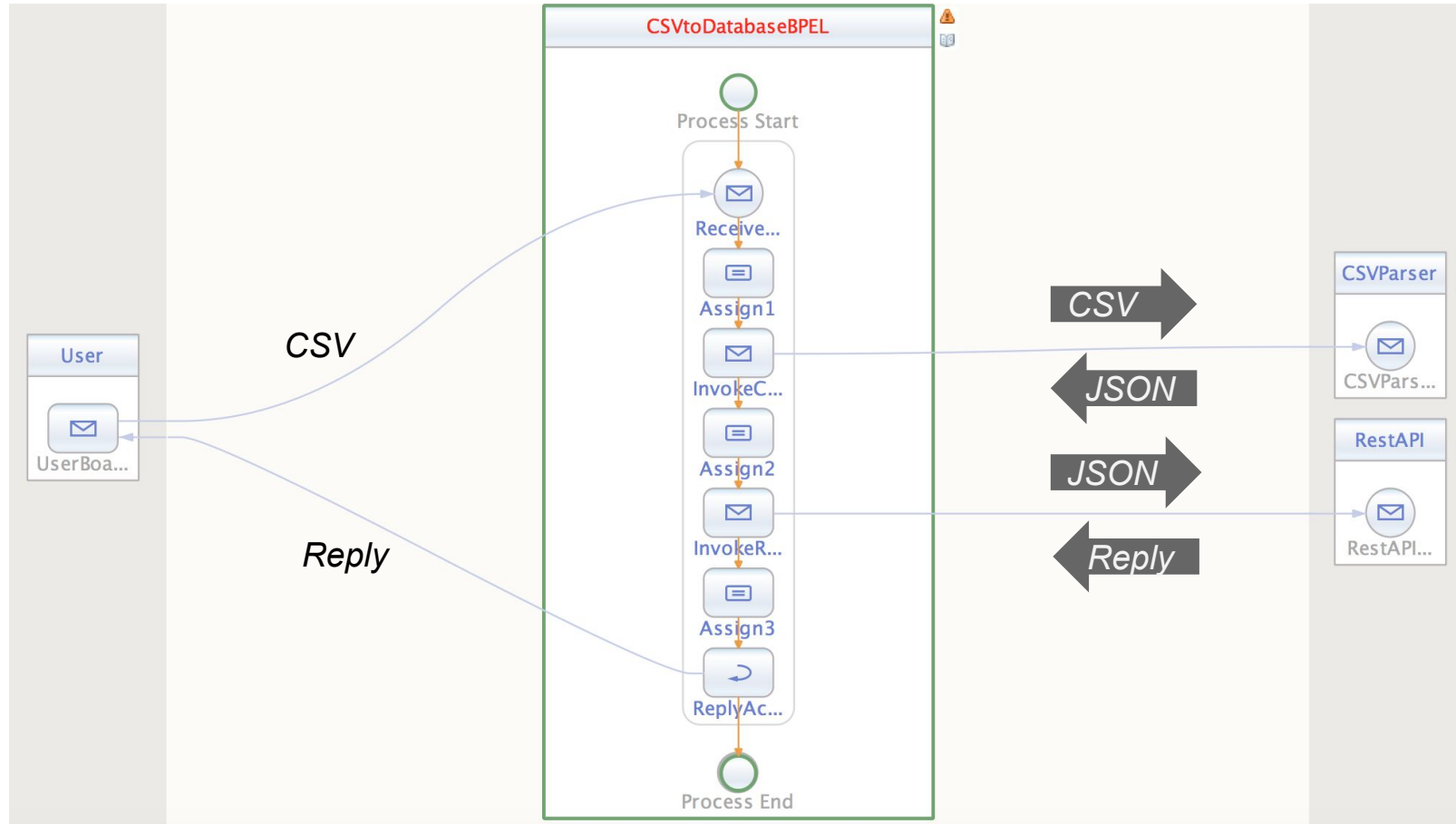
Application



Set new
entries in
database

App Server

III.A Our implementation: a BPEL module to import data from a CSV file



III.A Our implementation: a BPEL module to import data from a CSV file

Assign 1

The screenshot displays a BPEL editor interface with the 'Mapper' tab selected. The top toolbar includes tabs for 'Source', 'Design', 'Mapper', and 'Logging', along with icons for 'Operator', 'Boolean', 'String', 'Node', 'Number', 'Date & Time', and 'BPEL'. Below the toolbar, there are two side panels. The left panel, titled 'All' and 'Output', shows a tree view of variables: 'Variables' (expanded), 'CSVParserWSDL', 'AckHourraToUser', 'AckFromRestAPI', 'XMLParsedObjectsToRestAPI', 'XMLParsedObjectsFromParser', 'CSVToParser', 'CSVFromUser' (expanded), 'firstCSVUser' (expanded), 'firstCSVFromUser' (selected), 'Properties', and 'Partner Links'. The right panel, titled 'Input' and 'All', shows a tree view of variables: 'Variables' (expanded), 'CSVParserWSDL', 'AckHourraToUser', 'AckFromRestAPI', 'XMLParsedObjectsToRestAPI', 'XMLParsedObjectsFromParser', 'CSVToParser' (expanded), 'CSVfromUser' (expanded), 'CSV_parserRequest' (selected), 'Properties', 'CSVFromUser', and 'Partner Links'. The central workspace is empty.

III.B Our implementation: our Android application - REST implementation

Collecting GPS coordinates



Regatta

Collections de données GPS

ACQUÉRIR UN POINT

1.4706070805145646:43.56519232
703216
1.4706070805145646:43.56519232
703216
1.4706070805145646:43.56519232
703216

Nom de la collection

collection1

TERMINER

Write



Local file

III.B Our implementation: our Android application - REST implementation

Exporting a collection



REST

GET(Regattas, Races)



PUT(Collection)

App Server

III.C Our implementation: our app server - REST implementation

Service Oriented Architecture

Data-oriented REST API : Auto-generated services

```
public static schema : Schema = new Schema({  
  "startDate"      : new properties.DateProperty(),  
  "endDate"        : new properties.DateProperty(),  
  "races"           : new properties.ArrayProperty(new properties.ObjectProperty<Race>(Race.schema)),  
  "location"        : new properties.StringProperty(),  
  "name"            : new properties.StringProperty()  
})
```



<http://server.xyz/api/regatas/>

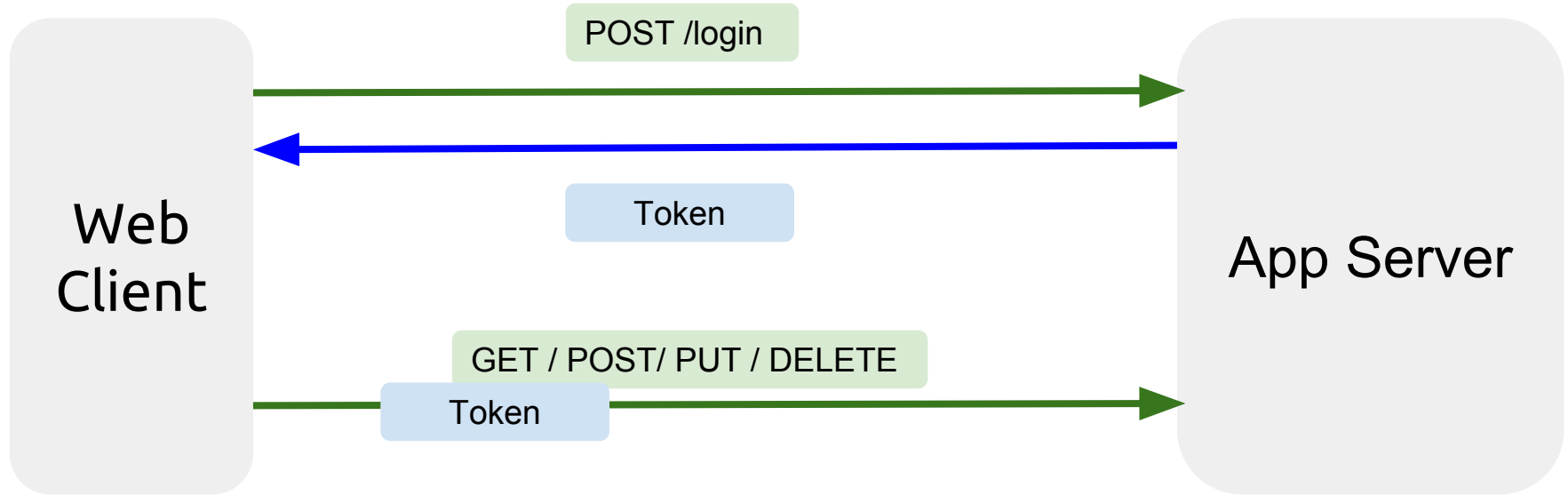
GET	/
GET	/
DELETE	/
POST	/
PUT	/

```
{  
  "startDate": 1472797167854,  
  "endDate": 1472800767854,  
  "concurrents": [  
    {  
      "boatIdentifier": "cdb1c2be-defa-495a-9696-f29f66b1fd6a",  
      "user": "587ab1f447a7ee51ce4b5780",  
      "skipperName": "Skipper_0",  
      "device": "587ab1f447a7ee51ce4b576c"  
    }  
  ],  
  "map": "587ab1f447a7ee51ce4b5794",  
  "data": "587ab1f547a7ee51ce4b5795",  
  "buoys": [  
    {  
      "x": 635,  
      "y": 432  
    }  
  ],  
  "name": "Race_0"  
}
```

III.C Our implementation: our app server - REST implementation

Service Oriented Architecture

Authentication scheme : JSON Web Token



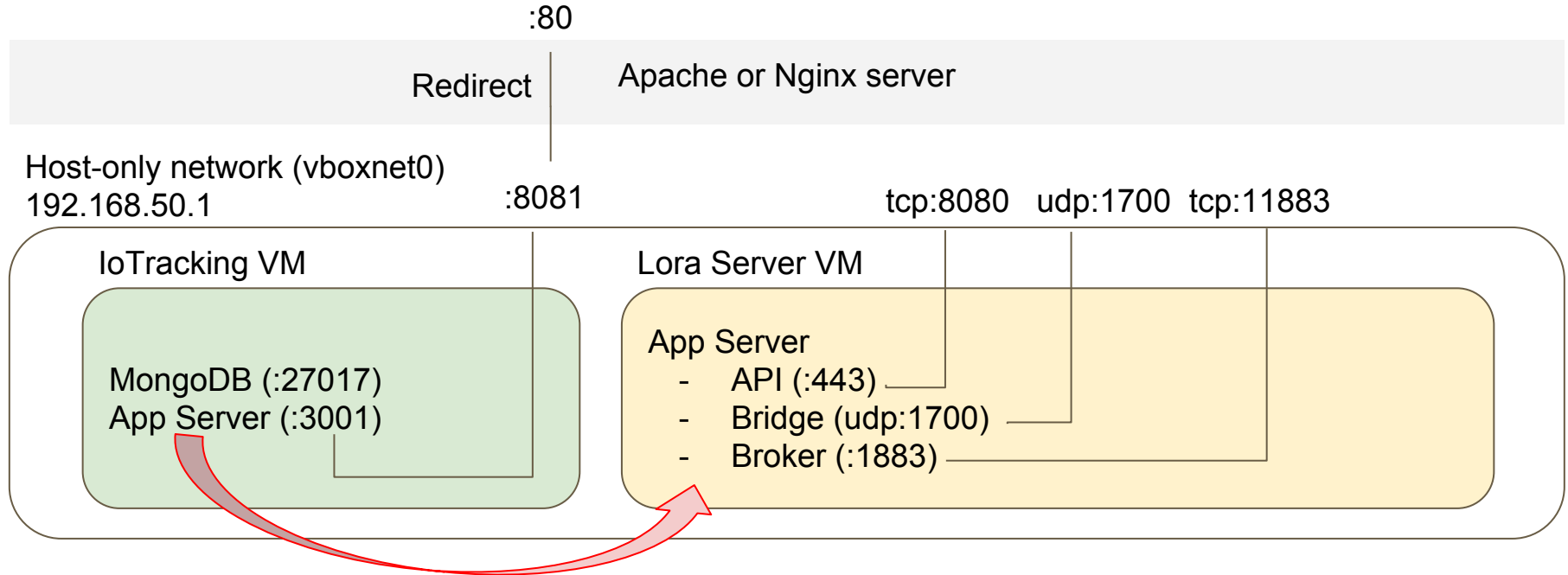
III.D Our implementation: Deployment

Keywords :

FLEXIBLE and EASY

Constraints :

- Only one physical machine with **limited resources** (no cloud)
- Deployment process must be **easy**.



IV. Project management - Planning

October

November

December

January

IceScrum

Agile Framework - Scrum
Online software IceScrum

Sprint meeting every week







Planning of tasks for each work session


Task list shared with all group members




*Found IceScrum too time
consuming to keep up to date*

IV. Project management - Planning






  IoTracking  Backlogs  Planning  Task board  Features



 82  

Sprint 6 - In progress
09/12 → 15/12

 All tasks 

0 / 19 \$
0 ⌚  [New task](#)

TODO 33


IN PROGRESS

DONE

☆ 32 Mise en place des Web Services SOAP
1 \$


48 ? ⌚
Création service
écriture sur la base de
données

New task


47  ? ⌚
Creation service
import fichiers CSV

0/2 In progress

☆ 40 [Client Web] Gestion des devices
1 \$

72  ? ⌚
Server communication

New task

71 
Client only page
design

1/2 In progress

IV. Project management - Planning

1.

Meeting at the beginning of each work session:

- What needs to be done
- Who will do what

2.

Set an agenda before the work session begins

3.

Global ToDo list for the week



Efficient!

Everybody knows what they have to work on

Exemple

Thursday 19/01
15h

Planning:

- Communication LoRa server to Gateway
- Android application
- Work on the final report

Exemple

Week 3 - 2017


ToDo:

- Finish presentation for english
- Prepare demonstration of project
- Finish documentation for the project

IV. Project management - Software Engineering




Project Management Plan

 *guide both project execution and project control*

- Scope
- Schedule
- Cost
- Quality
- Human Resources
- Communications
- Risk and Procurement
- etc.




Software Requirement Specifications

 *give a detailed description of the requirements of our project*

- Scope
- Constraints
- Functional requirements
- External requirements
- Availability
- Security
- Maintainability
- etc.



Software Design Document

 *expands and describe the functionalities presented in the SRS*

- Scope
- System description
- System Architecture
- Design constraints
- Components description
- Interfaces description
- etc.

V. Conclusion

Service-Oriented Architecture

Need to make a
relevant choice
about **SOAP or REST**

SOAP if

- Architecture already in SOAP
- Banking-like transactions

REST if

- All other cases

Project Management & Software Document

Necessary to **organise a project**
&
Keep the **right direction**

Use Agile Scrum method to

- **Emphasis** on the more urgent tasks.
- Less loss of time

IceScrum

- Not so convenient...
- We all experienced Agile methods before



Thanks for your attention !

— Any questions ? —
