Technical Proof-of-Concept Projects

Olivier Liechti
Open Source Frameworks
Master Of Science in Engineering (MSE)

MSE | MASTER OF SCIENCE IN ENGINEERING

Planning

Date	Java EE Frameworks	Gamification Project
23.09.13	Intro, Java EE Overview, EJBs	Environment setup 1
30.09.13	REST APIs & JAX-RS	Environment setup 2 (automation)
07.10.13	Design and document a REST API for your gamification engine	
14.10.13	Persistence with JPA	Test and implement your REST API
21.10.13	Break	
28.10.13	Test and implement your REST API	
04.11.13	Test and implement your REST API	
11.11.13	Presentations & Demos	
18.11.13	**Spring Framework**	
25.11.13	Technical POC Project: Define the scope & plan the activities	
02.12.13	Technical POC Project: Build the reference system	
09.12.13	Technical POC Project: Build the test infrastructure	
16.12.13	Technical POC Project: Build the test infrastructure	
23.12.13	Break	
30.12.13		
06.01.14	Presentations & Demos	
13.01.14	Java Message Service	Q&A

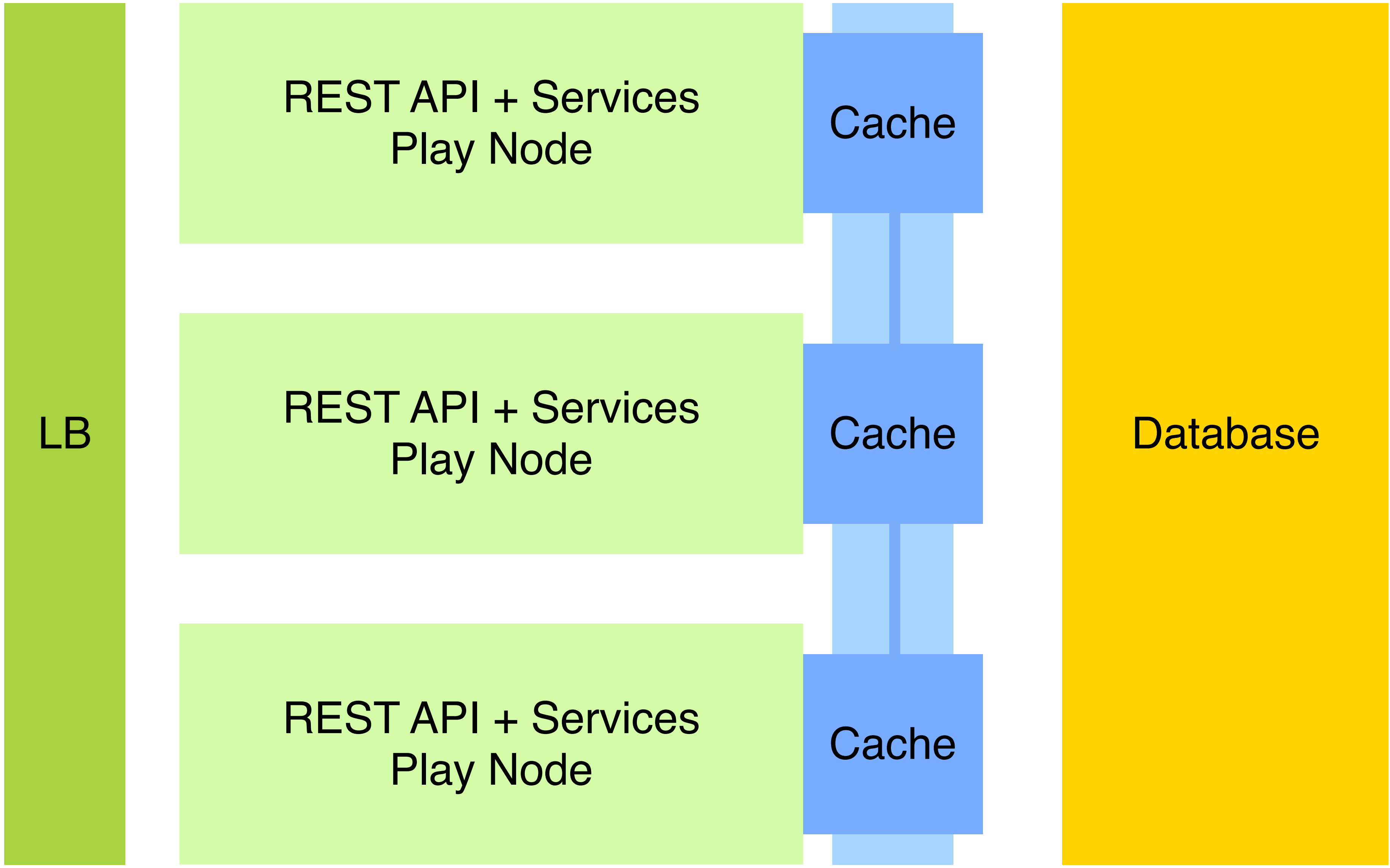
Introduction

- **Objectives**
    - Answer specific technical questions, based on facts.
    - Define a clear strategy to answer these questions (implementation of “system-under-test”, generation of test data, simulation of traffic, analysis of collected metrics).
    - Implement advanced mechanisms in the context of Java EE & REST APIs.
- **Guidelines**
    - 3 themes
    - 6 teams of 3 students
    - Every theme will be treated by two teams
- **General references**
    - http://jmeter.apache.org/
    - http://jmeter-plugins.org/

Theme 1: Play Framework & Distributed Caching

- **Introduction**
  - Play is a web framework that is becoming very popular. Scalability was one of its main design goals.
  - Play can be used both by Java and Scala developers.
- **Questions**
  - **Question 1**: What is the performance impact of using a caching layer when implementing a REST API with Play?
  - **Question 2**: How is it possible to use a caching layer in a cluster environment, when several Play “nodes” are setup to serve HTTP requests?
- **References**
  - http://www.playframework.com/
  - http://www.playframework.com/documentation/2.2.x/JavaSessionFlash
  - http://www.playframework.com/documentation/2.2.x/JavaCache
  - http://www.reactivemanifesto.org/

Theme 1: Play Framework & Distributed Caching



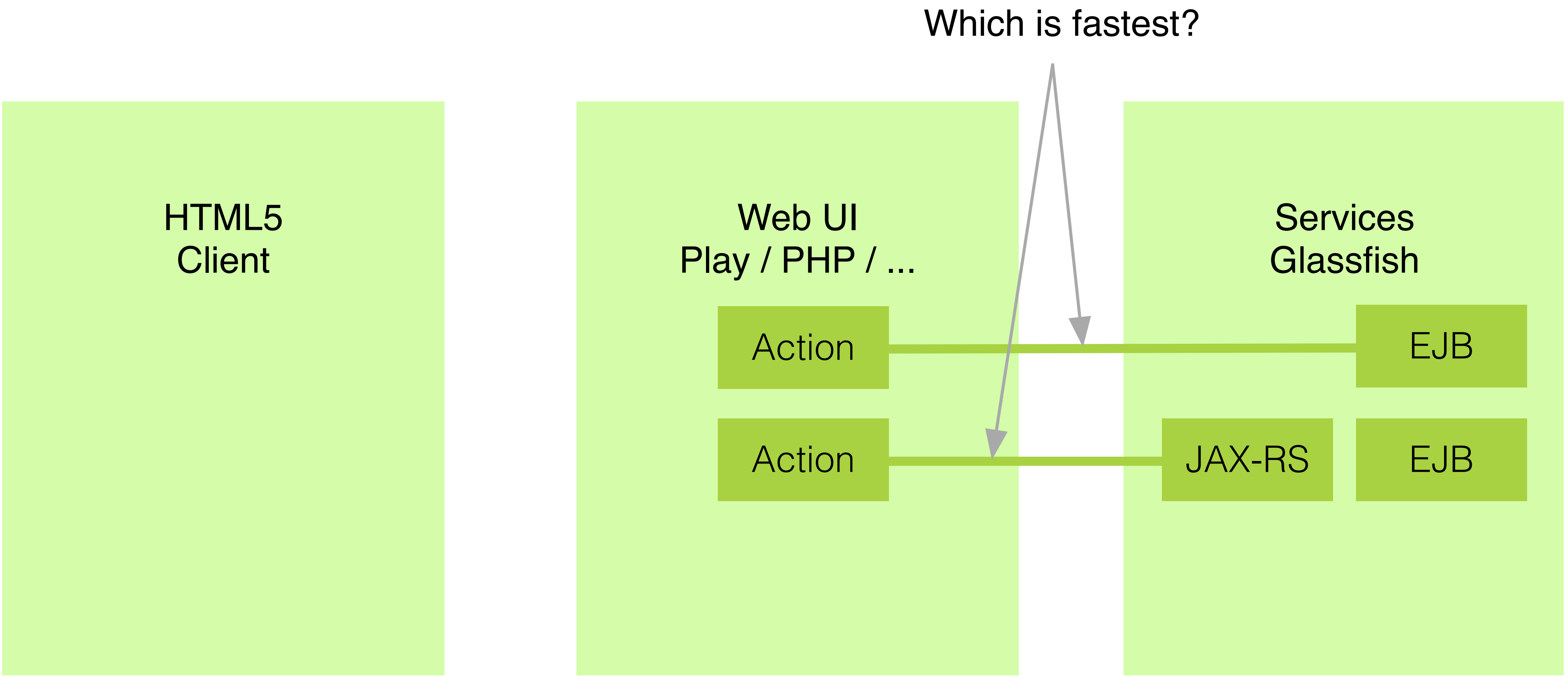
Theme 1: Play Framework & Distributed Caching

- **What do I expect in the documentation & presentation?**
  - **Explain** how Play approaches HTTP session state management and how it is different from the traditional Java EE approach. Explain the benefits of this approach.
  - **Explain** what is means to use a caching layer in a cluster environment and what are the issues to consider. Explain how Ehcache addresses these issues.
  - Describe the **strategy** for answering the two questions. Describe the test scenarios you have defined and the “System-Under-Test” you have implemented.
  - Describe what kind of **test data** you have generated (and how) and what kind of **traffic** you have simulated. Describe the **conditions** of the experiment.
  - Present the **collected metrics** and **analyze** them to answer the questions.
  - Explain how someone can **do another run** of the experiment (how to setup a test environment, how to use tools to generate the data and simulate the traffic, etc.).

Theme 2: How to Expose the Business Services?

- **Introduction**
  - There are different technologies for implementing the Presentation Tier: PHP frameworks, Java Frameworks such as Play, etc.
  - Unbundling the Web Tier and the Business Tier and deploying them independently offers advantages. In such a scenario, we need to define the communication protocol between the two tiers. In other words, to decide how the business services are exposed and invoked.
  - Putting a REST API just in front of the business services is a popular approach. But what is the overhead and is it worth using alternative protocols (RMI/IIOP, PHP-Java Bridge, etc.)?
- **Questions**
  - **Question 1**: Given the Presentation Tier technology of your choice (PHP, Play, etc.), how can you implement two ways of invoking Business Services from your controllers. One being via a REST API, the other being an optimized protocol.
  - **Question 2**: What is the performance overhead of using the REST API as a communication protocol?
- **References**
  - http://docs.oracle.com/javaee/7/tutorial/doc/ejb-intro004.htm#GIPIZ
  - http://docs.spring.io/spring/docs/3.0.x/spring-framework-reference/html/remoting.html
  - http://www.jcp.org/en/jsr/detail?id=223
  - http://php-java-bridge.sourceforge.net/pjb/index.php
  - http://threecrickets.com/scripturian/
  - http://quercus.caucho.com/

Theme 2: How to Expose the Business Services?



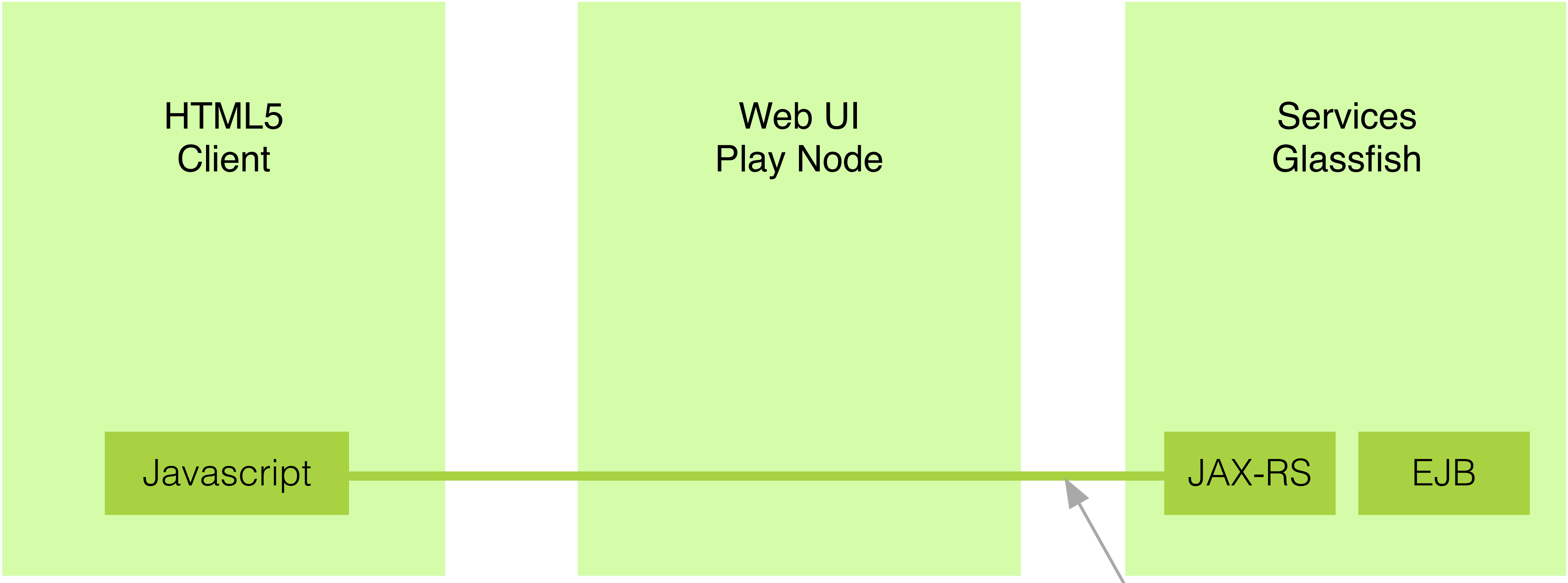
Theme 2: How to Expose the Business Services?

- **What do I expect in the documentation & presentation?**
  - **Explain** the foreseen benefits and drawbacks associated with the different communication protocols, describe the trade-offs you anticipate.
  - Describe the **strategy** for answering the two questions. Describe the test scenarios you have defined and the “System-Under-Test” you have implemented.
  - Describe what kind of **test data** you have generated (and how) and what kind of **traffic** you have simulated. Describe the **conditions** of the experiment.
  - **Describe** how you can implement the alternative communication protocols and what technologies/libraries you have used.
  - Present the **collected metrics** and **analyze** them to answer the questions.
  - Explain how someone can **do another run** of the experiment (how to setup a test environment, how to use tools to generate the data and simulate the traffic, etc.).

Theme 3: Optimized Serialization in REST APIs

- **Introduction**
  - Assuming that the back-end is exposing a REST API and that an HTML5 client is using this API, there are different ways to handle the serialization of the payload.
  - The first question is what format to use for the payload (xml vs json vs something else). The second question is how to handle the serialization of that format in an optimal way.
- **References**
  - https://github.com/eishay/jvm-serializers/wiki
  - http://www.cowtowncoder.com/blog/archives/2012/04/entry_470.html
  - http://mobilebit.wordpress.com/2013/09/25/protobuf-mqtt-yummy-fast/
  - http://techtraits.com/noproto/
- **Questions**
  - **Question 1**: Is it worth using an alternative to json (such as protocol buffers)? What is the gain and what are the drawbacks?
  - **Question 2**: Assuming that we want to use json, what is the most efficient way to handle the serialization/ deserialization of the payloads?

Theme 3: Optimized Serialization in REST APIs



Is there an alternative to json?
How can we optimize the performance of json processing?

Theme 3: Optimized Serialization in REST APIs

- **What do I expect in the documentation & presentation?**
  - **Explain** the foreseen benefits and drawbacks associated with the different communication protocols, describe the trade-offs you anticipate.
  - Describe the **strategy** for answering the two questions. Describe the test scenarios you have defined and the “System-Under-Test” you have implemented.
  - Describe what kind of **test data** you have generated (and how) and what kind of **traffic** you have simulated. Describe the **conditions** of the experiment.
  - **Describe** how you can implement the alternative communication protocols and what technologies/libraries you have used.
  - Present the **collected metrics** and **analyze** them to answer the questions.
  - Explain how someone can **do another run** of the experiment (how to setup a test environment, how to use tools to generate the data and simulate the traffic, etc.).

Teams

Team 1
Caching

Team 2
Caching

Team 3
Call EJBs from PHP

Team 4
Call EJBs from Play

Team 5
Alternatives to Json

Team 6
Optimized Json