Dear Students,
This is to provide you with instructions about the project each group must realize for the final test.

The final test will be about engineering and implementing a **(micro-)SOA application** by

- following the SOA engineering principles,
- applying the SOA best practices,
- adopting the technologies explained in the classroom, **the way they have been taught** (**please, do not download and "adapt" examples from the Internet: realize an "original" project yourself ;**).

Although you are free to choose the application domain and the specific purpose of the application, some constraints follow.

1. The application must comprise **REST** and **SOAP** services (developed using Apache CXF and Spring), and **microservices** (as per the lectures by doc. Filippone).

2. The application must be composed of (at least) **three service providers**, (at least) **two service prosumers**, and (at least) **one client application** (e.g., java application, web application, mobile app). As for the microservices, the interaction client-to-service must pass through an API gateway.

3. The client(s) must interact with (at least) the prosumer(s), and the prosumer(s) must interact with (at least) two providers.

4. Develop at least one **asynchronous** service (polling and/or callback approaches). At least two prosumers **must execute their job "in parallel" and then synchronize/coordinate each other** before responding to the client(s), e.g., run in parallel for collecting data, then synchronize/coordinate each other for polishing/aggregating the collected data, and only after responding to the client(s)... **Importantly, the need to use asynchrony must be clearly motivated/justified, not in general, but by considering your system's logic/interactions/...**
5.

6. Concerning the client(s), the goal is to show **at least three different interactions with the prosumer(s)**. In addition to the interactions with the prosumer(s), the client(s) can also interact with the providers directly, if needed.

7. From an engineering point of view, you must create an architectural diagram using a service/component diagram or the classical blocks (for services/components) and lines (for connections) diagram. The **interaction scenarios** must be shown and documented. For each interaction scenario, the corresponding **sequence diagrams** must also be produced.

8. The application, its goals, expected outputs, how it works, etc., must be introduced with a clear **textual description** (a .docx or .pptx would be ok).

9. The **code must be documented** with clear and verbose comments within the source code, WSDL files, etc. Use **Open API** and **Swagger** for all REST services.

10. A **README.txt** file must be provided to clearly and unambiguously explain all the steps to be followed to set up the application.

11. Use **Spring Boot**, and **Docker** for microservices.

12. The project(s) must be **Mavenized** (**DO NOT USE OTHER PROJECT MANAGEMENT TOOLS**)

13. **[Optionally] Maven archetype(s)** can be realized for the project(s) structure.

14. Microservices must be enabled to be **deployed in multiple instances, and their interaction must be load-balanced**. Note that load balancers, API gateways, and discovery services are "additional services" with respect to the business services to be realized as described above. **Importantly**, **the need for scaling must be motivated/justified, not in general, but by considering your system's logic/load/...**

During the project defense, **each member** of the group **must** clearly demonstrate his/her contribution to the project realization. I am assuming that the groups are the same as the ones of the mid-term HOMEWORK. Please, send me an email only if something changed.

HOMEWORK ASSESSMENT METHOD

- Completeness of contents (i.e., **quantity** in terms of having all the necessary or appropriate parts)
- Quality of contents (e.g., **completeness and soundness**)
- Quality of presentation (e.g., **clarity**)
- Contents organization (e.g., **gradualness**)
- Presentation format (e.g., slides arrangement -> **no hasty preparation**)
- **Originality of the project** with respect to existing examples from the Internet.
- **Ability to build on and elaborate on** what has been explained in the classroom.