PictuRAS - Phase 2

Soft. Arch. Req. - Master in Computer Engineering
University of Minho

October 29, 2024

In Phase 2 of the project, teams must design an architectural solution for the **PictuRAS** application. The application must function in a *web* environment (possibly also in a *mobile* environment) and its architecture must adequately consider the following types of non-functional requirements:

Non-functional Requirements

1. **Usability**:
   - A *web* application for general use that competes with many similar applications will only succeed if its use is simple, intuitive, and user-friendly.
   - The interface must be designed to provide a fluid user experience, with clear and accessible navigation, minimizing the need for complex instructions.
2. **Scalability** and **elasticity**:
   - A *web* application for general use, if successful, may experience significant usage spikes. Thus, for the application to continue evolving despite its own success, it is crucial that it be prepared to scale elastically and automatically.
   - The architecture must support a significant increase in the number of simultaneous users without experiencing service failures or interruptions.
   - At the same time, unnecessary costs should be avoided during periods of low usage.
   - Techniques that allow horizontal scalability should be preferred (e.g., load balancing, *message queuing*, *stateless* services, *caching*).
3. **Extensibility**:
   - The number of possible image processing tools that can be integrated into the application is almost unlimited. Thus, the application must be easy to extend.
   - The extension of the application's scope should not be limited to a single technology. The application should promote a development context that coexists with a high level of technological heterogeneity. This means that different parts of the application can be developed with different technologies, languages, and patterns.

Template for the Architectural Solution

This template is based on arc42, with some modifications introduced by the teaching team. It should be seen as a reference, with each team evaluating which parts make sense to include, or even add.

1. **Introduction and Goals**
   - Short description of the requirements, driving forces, extract (or abstract) of requirements (present a summary of the requirements document).
   - Top five quality goals (non-functional requirements) for the architecture with highest priority.

2. **Constraints**
    - Anything that constrains teams in design and implementation decisions or decision about related processes.
    - Content: describe the requirements that are mandatory to be included in the solution (e.g., app must be accessible by a web browser).
3. **Context and Scope**
    - Delimits your system from its external communication partners (neighboring systems and users).
    - Specifies the external interfaces.
    - Shown from both business/domain perspective and technical perspective.
    - Content: context diagram and with the indication of interfaces of the system with the main actors and/or the external components.
4. **Solution Strategy**
    - Summary of the fundamental decisions and solution strategies that shape the architecture. Can include technology, top-level decomposition, approaches to achieve top quality goals and relevant organizational decisions.
    - Content: identify and describe the architectural patterns included/adopted by the solution (e.g., microservices, MVC, client-server).
5. **Building Block View**
    - Static decomposition of the system, abstractions of source-code, shown as hierarchy of white boxes (containing black boxes), up to the appropriate level of detail.
    - Content: components and class diagrams for the solution.
6. **Runtime View**
    - Behaviour of building blocks as scenarios, covering important use cases or features, interactions at critical external interfaces, operation and administration plus error and exception behaviour.
    - Content: sequence diagrams (solution space) for the main use cases of the system.
7. **Deployment View**
    - Technical infrastructure with environments, computers, processors, topologies. Mapping of (software) building blocks to infrastructure elements.
    - Content: deployment diagrams that describe where the system components will be deployed.

Delivery Deadlines

- ~~App requirements document delivery (LaTeX/Typst and PDF versions): **Oct.18**.~~
- ~~Requirements document defense: **Oct.21-25**.~~
- App architecture delivery (LaTeX/Typst and PDF versions): **Nov.15**.
- Architecture defense: **Nov.18-22**.
- App implementation delivery: **Jan.17**.
- Implementation defense: **Jan.27-31**.