Embark Software Architecture Description

April 28, 2021

Explanation of Technology Stack

- Docker:
  Docker is chosen to deploy the embark system independent from customers ecosystem. Also the overhead is relatively low since the emba service already uses a docker to provide the crucial Kali Linux distribution Additionally to make the development faster we have included docker-compose as well which abstracts out most long Docker commands with simpler ones.

- OS: Kali Linux
  As mentioned previously the emba service, as tool dedicated to penetration testing, depends on the Kali ecosystem. So the choice of another OS would just complicate the development without offering a real benefit.

- Backend Framework: Django
  Django is the most comprehensive web development framework in python and provides basically everything developers want right out-of-the-box; such as ORM, uWSGI and aSGI support. It works on the MVC(Model View Controller) system abstracting out a lot of other configuration details. As a matter of fact the SD Team has already worked with Django, which eases development even further.

- Database: MongoDB
  A database is needed for two different reasons. First of all to provide a proper handling of uploaded firmwares although only references and metadata about the firmware is committed into the DB. Secondly the database will store (at least a subset of) the data generated by emba for further processing and aggregation. The choice of MongoDB once again depends on an already existing Database shipped with emba. The team agrees, there is no need to differ from the emba ecosystem.

- Frontend:
  - HTML-5
  - Javascript
  - CSS:
    Bootstrap is a free and open-source CSS framework directed at responsive front-end web development. It contains CSS and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

Diagrams

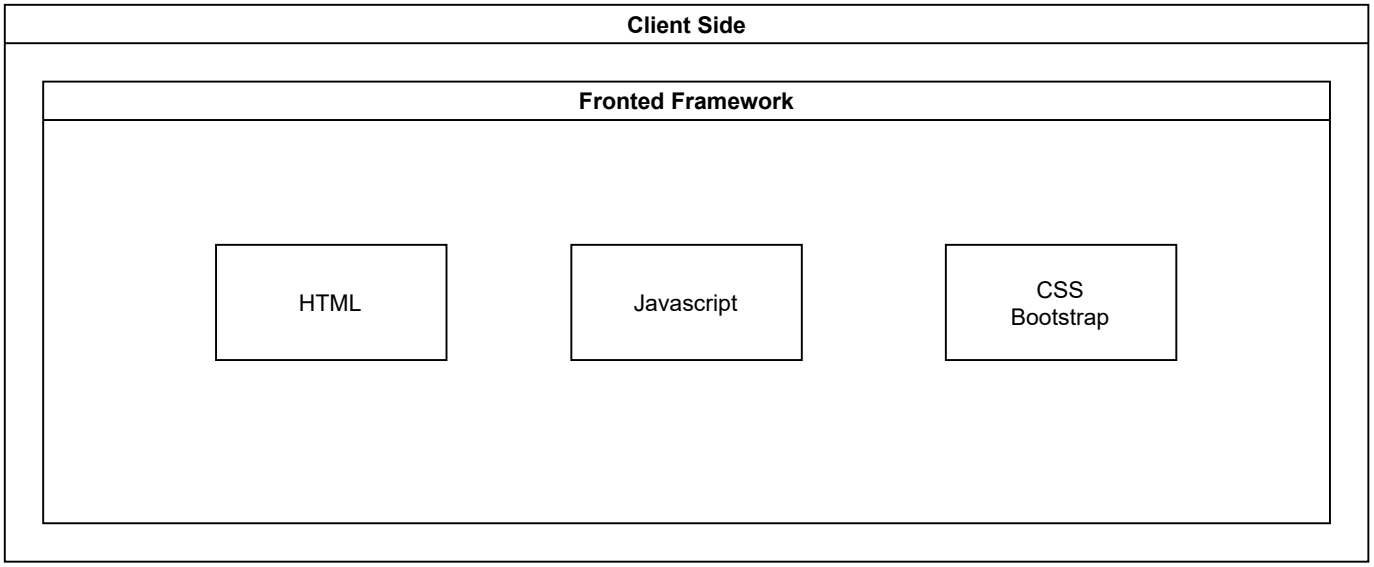
Class Diagram

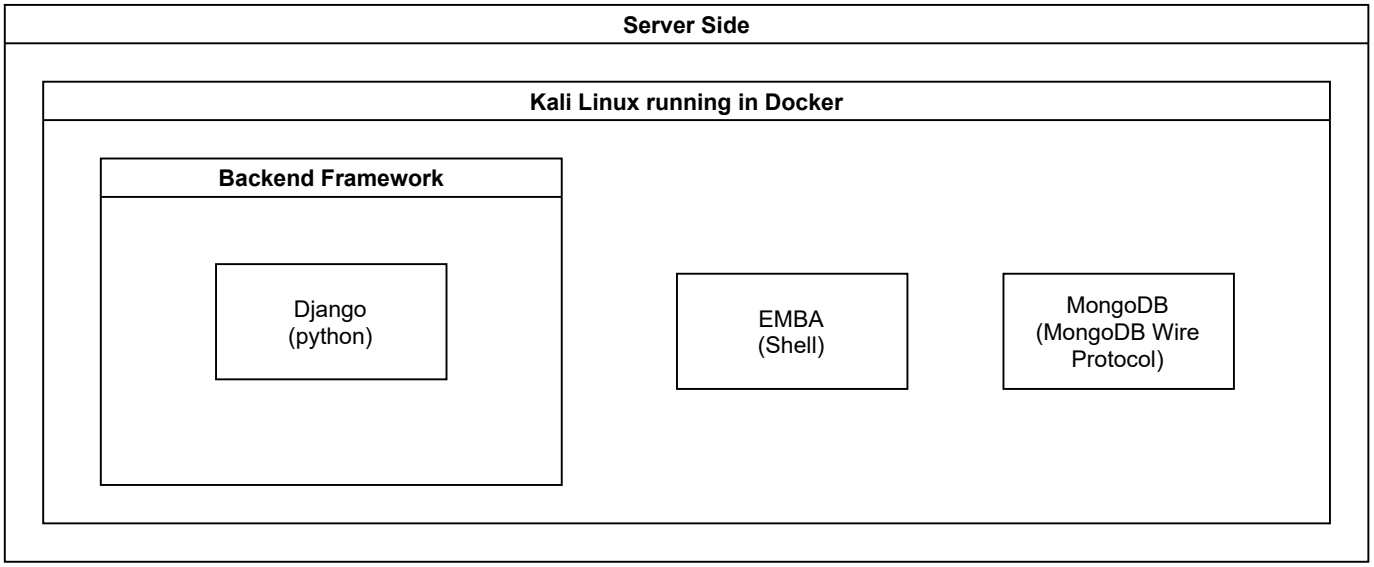
The UML diagram displays the dependencies between our software components. There it is not necessary to describe how a feature is to be implemented, but rather to describe a rough framework of the software. Our UML diagram devides the software into frontend and backend. In the frontend, we use a template-based website design that is refined by Bootstrap later on. The backend consists of a runner to provide user interaction, an uploader to upload firmware images, a control dashboard to display the list of analyzed firmware images and the control dashboard for a report on the currently analyzed image.

Sequence Diagram

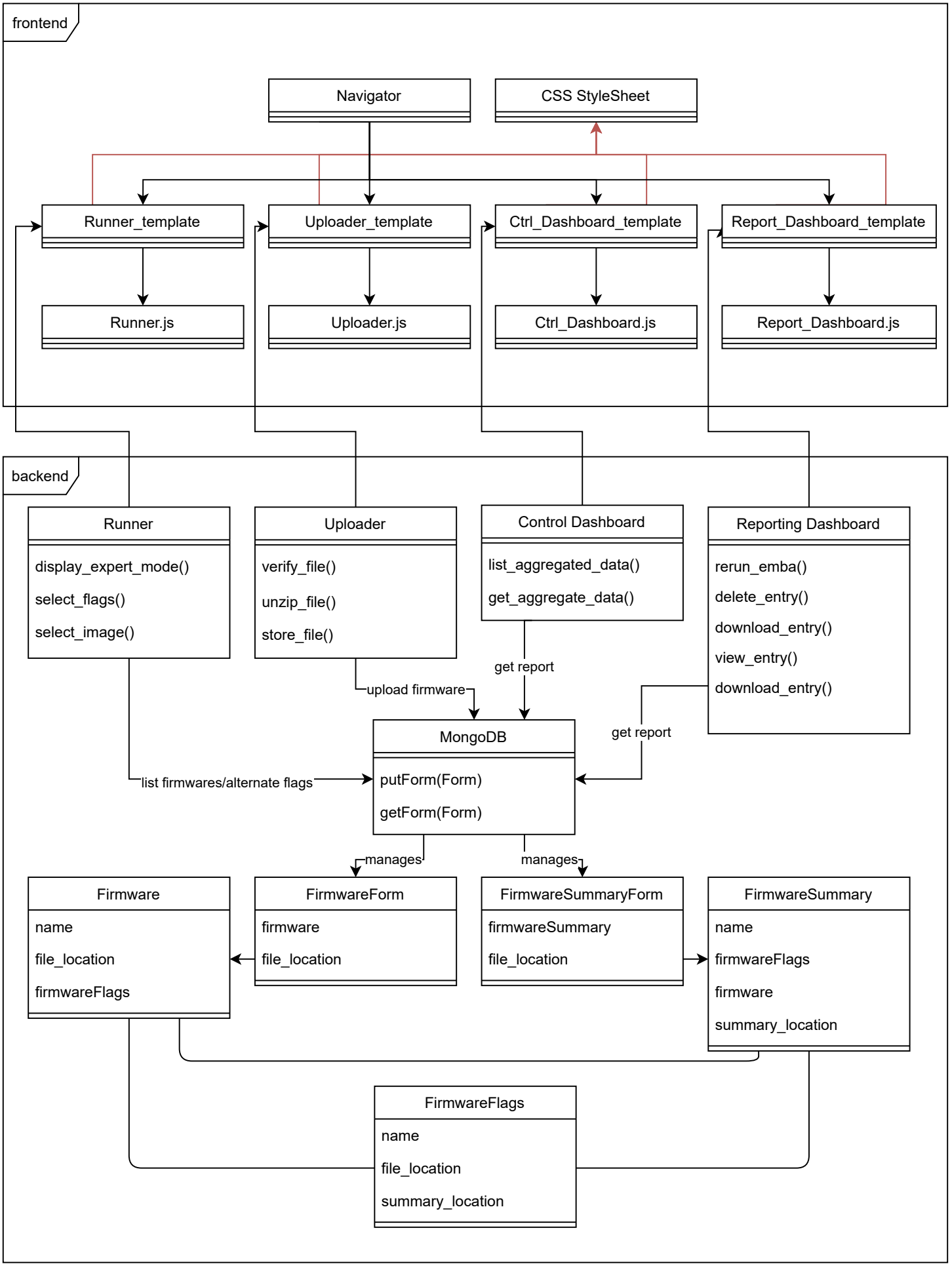
The Sequence diagram shows the runtime-behaviour of the application. With this type of representation, the individual communication streams between the software components can be illustrated. By abstracting the user layer, the places where the user interacts with the software can be identified. Loop frames are used to combine activity flows that can run more than once.

Embark: Technology Stack





Embark: UML-Diagram



Embark: Sequence Diagram