Legend - colors

- ▢ Partly or fully programmed block
- ▢ Functional block
- ▢ DB-Server
- ▢ Block still in designing phase
- ▢ Nice-to-have function

Legend - numbers

- 1 – GUI: fully implemented in the frontend. Partly supporting the REST-API version 1.0. CEFRIEL is in charge of the development.
  Relevant files (.js, .css ...) located on the production server.
- 2 – Server to frontend communication interface:
  The REST-server interface is fully implemented using Jersey (Java) and supports the REST-API version 1.0. An instance is currently running on port 3000 of the production server using Apache Tomcat 8.
  The REST-client block is just an idea, hasn't been designed yet (see discussion with Daniel, CEFRIEL).
  The event manager exists and is sofar only fetching existing user projects and simulation results, without any database in the backend. This block can be enriched as both the REST-API and the Database functionalities are extended.
  The ORM framework Hibernate has been embedded in the Jersey project using Maven and is ready-to-use.
- 3 – SimObject generator:
  This functional block is an Eclispe project on ist own. Ist functions are to be triggered by the block 2, in reaction to Client events. The SimObject is built up using the Spring framework, using XMLs as inputs.
  A rough draft of the core SimObject components has been coded in Eclipse, together with a JSON parser (on side).
  The REST-Client for interfacing the computational backend has still to be designed, together with the load-balancing / simlation management block.
- 4 – Account management system:
  Responsible is all4ip, Timo.
- 5 – Simulation-thread management:
  A Java Wrapper around the (unofficial) library JMI has been developed, in order to start, kill and retrieve information from different Matlab instances. These instances are linked to different Java threads, whose management and interaction is carried out within a Java runtime. A first functioning version has been coded in a separate Eclipse project and exported as .jar using Maven. The interaction with the .jar should be reported in a JavaDoc API (already generated by Maven).
- 6 – Simulator:
  Responsible: Christian. An object-structure has still to be fixed. The Matlab scripts to be executed by Java are not known yet.