

## Architectural design Brainstorm sessie:

### Algemene Opmerkingen

We mogen Javascript gebruiken.

Java kent iedereen goed. Dus dat kan je makkelijk gebruiken en er zijn goed te gebruiken testing tools.

We hebben besloten dat ons programma maar een client ondersteund. Dit staat nog niet in ons requirements document als constraint.

Topics zijn verdeeld qua verslag en om 4 uur kijken we hoe het er bij elkaar uit ziet. Dan vullen we elkaar aan.

Mogelijke CouldHave is auto-backups van de BioBricks als je bezig bent.

### Opbouw systeem

#### Browser

In Javascript. Dit word de GUI. Communicatie gebeurd via AJAX (JSON files).

#### Server

In Java. Moet SBML files kunnen lezen en doet import export van BioBricks. Hier doen we veel checks of bepaalde berekeningen wel mogen.

We moeten kijken hoeveel we opslaan. (BioBricks, lijst van files etc....)

SBML vertalen naar JSON en dat weer opsturen naar Browser

### Het verslag

1. Introduction	Thomas
1.1. Purpose of the System	Thomas
1.2. Design Goals	Thomas
1.3. Definitions, acronyms and abbreviations	Thomas
1.4. References	Thomas
1.5. Overview	Thomas
3. Proposed software architecture	Thomas
3.1. Overview	Thomas
3.2. Subsystem Decomposition (which sub-systems and dependencies are there between the sub-systems?)	Niels & JanPieter
3.2.1. Interface (API) of each sub-system	Niels & JanPieter
3.7. Boundary Conditions (how is the system started and stopped, what happens in case of a system crash)	Niels & JanPieter

3.3. Hardware/Software Mapping (mapping of sub-systems to processes and computers, communication between computers),	JanPieter
3.4. Persistent Data Management (file/ database, database design) (JSON SBML research)	Albert
3.5. Global Resource Handling and Access Control for the different actors	Felix
3.6. Concurrency (which processes run in parallel, how do they communicate, how are deadlocks prevented?)	Felix

### Opmerkingen voor verslag

3.5 Access control laten we wachten.. Dit is niet erg interessant voor ons programma.

3.6 Concurrency, We ondersteunen maar een client. Wel aparte threading voor simuleren. Het afhandelen van signalen word een aparte thread.