The current system consists of a set of applications. Some of the system functions have not been implemented yet; some solutions are too old.

The new system is meant to be web-oriented. It should replace the most inconvenient UI solutions and implement the new features being waited.

One important purpose of the new system is to provide hierarchical database configuration editing. The current system requires user's knowing relational database systems and/or C++ while editing the configuration.

### Homogeneity of values

```javascript
module.exports = function() {
  const ValueStream = require('valuestream');
  const webpack = require('webpack');
  const path = require('path');
  const test = require('./test');

  function Test() {
    this.name = null;
    this.counter = 0;
  }

  Test.prototype = {
    name: function() {
      return this._name || (this._name = this.constructor.name);
    },
    increment: function() {
      this.counter++;
    },
    decrement: function() {
      if (this.counter > 0) this.counter--;
    },
    equals: function(other) {
      return this.counter === other.counter;
    },
    plus: function(other) {
      return this.counter + other.counter;
    },
    times: function(other) {
      return this.counter * other.counter;
    },
    divide: function(other) {
      return this.counter / other.counter;
    },
    modulo: function(other) {
      return this.counter % other.counter;
    },
    max: function(other) {
      return Math.max(this.counter, other.counter);
    },
    min: function(other) {
      return Math.min(this.counter, other.counter);
    },
    compareTo: function(other) {
      return this.counter - other.counter;
    },
    toString: function() {
      return this.name + ' ' + this.counter;
    },
  }

  exports.Test = Test;
}
```

There is no difference between synchronous/asynchronous calls for template editors.

### Object configuration

Node.js

- mysql (npm: mysql)
- node-static (npm: node-static)
- crypto (npm: crypto)
- sessions (npm: sessions)
- PEG.js (npm: pegjs)
- Formidable (npm: formidable)
- Formidable (npm: formidable)

A simple web-server has been created using Node.js framework.

- A template engine is designed. The main feature is homogeneity of synchronous and asynchronous values.
- A hierarchical objects (mapped on relational DB) editor has been implemented.
- The server allows user to read and modify experiment configuration, conditions and metadata.
- This is the first part of management system refining. Some new changes are coming soon.