

Manual

1. How to use the code?

a. Import the code in Eclipse

- i. File > New > Java project
 1. Choose name "DS" > Finish
- ii. Drag the files of the "src" folder to "src" in the Eclipse IDE
- iii. Right click on "DS" > Build Path > Configure Build Path
 1. Go to Libraries > "Add external JARs..."
 - a. Select all .jar files from the folder lib (except the "avro-tools-1.7.7.jar" file)
 - b. Apply > Ok

b. Single computer functionality

- i. Note that you always run the controller before any client. Otherwise you will get the error message "Cannot establish connection with the controller!".
- ii. Go to controller/Controller.java in the package explorer
 1. Click the green arrow
 - a. The server/controller is now running!

Note: If new controller is elected and you want to restore old

controller

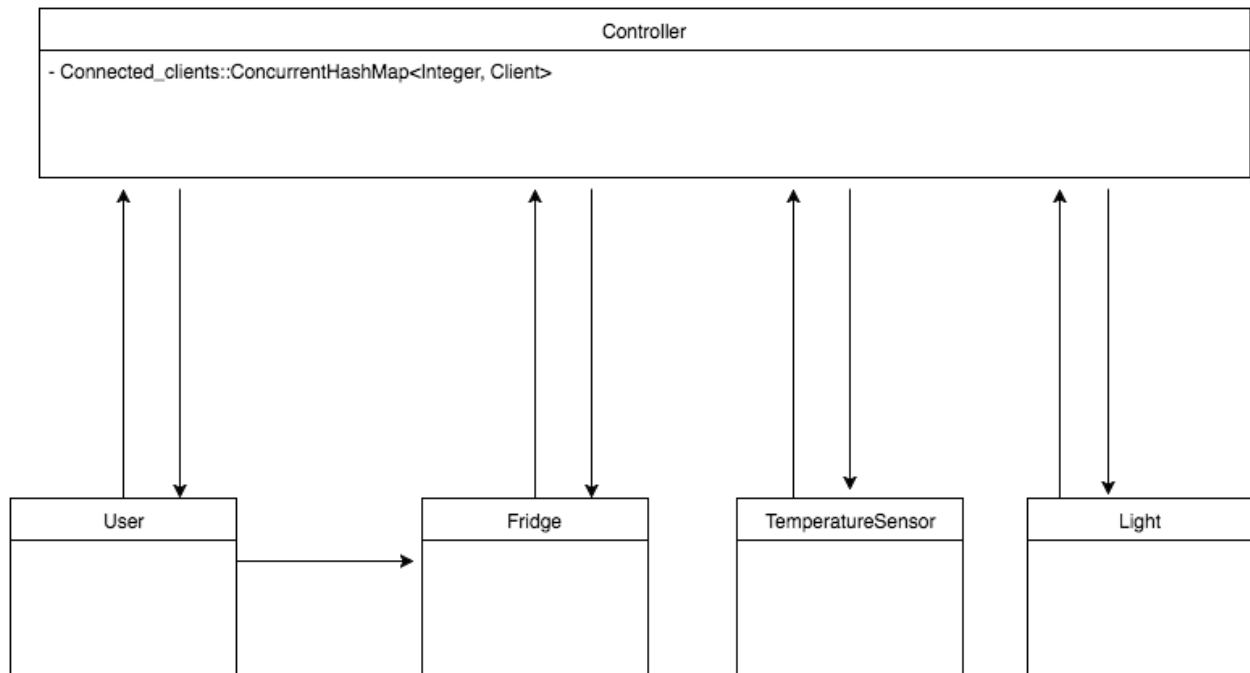
⇒ give IP address of new controller as parameter.

- iii. Go to clients
 1. Select the client you want to run
 - a. Click the green arrow
 - i. "The client you specified is now connected with the server and running"
 - ii. Type "list" in the Terminal to see which commands you can execute with this client

c. Distributed functionality

- i. Type "ifconfig" in the Terminal (@ server side pc) to determine its IP-address.
- ii. Type the IP-address you found in the previous step in the Build configuration of the clients (@ client side pc) + potential port number (there is a default number provided in the code)
- iii. Now follow the steps as described in the single computer functionality paragraph.

2. Description of the architecture

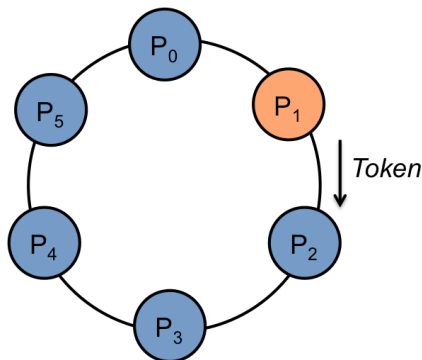


Pinging:

Controller pings to every client to see if it is still responding. The client also pings but only to the controller. Dit is hoe het in de werkelijkheid ook vaak gebeurt. Op deze manier kunnen we de lijst van geconnecteerde devices up-to-date houden.

Fault tolerance:

When the controller fails, the clients arrange themselves into a ring and start the election process (according to Chang-Roberts).



Threads:

The user interface runs on its own thread (in client and controller) to allow for true fault tolerance.

3. Used technologies and libraries

- **Eclipse** (Open source IDE)
- **Apache Avro** (Communication between server and clients and the mutual communication between the clients)
- **Imports:**
 - `Java.util` (Map, concurrent hashmap, list, random etc.)
 - `Java.io` (IOException, InputStreamReader, BufferedReader etc.)
 - `Java.net` (InetAddress, InetSocketAddress etc.)
 - `Java.lang.reflect.UncheckedIOException`