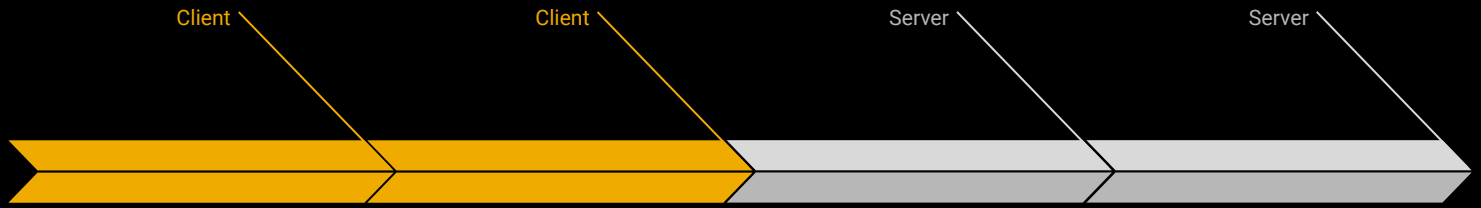
SAP Hybris

Reactive Programming
Efficient Server Applications

Piotr Wittchen, Arkadiusz Galwas, SAP Hybris

Let’s plan an application

Basic application flow



Data Acquisition

Reading data from sensors.

Sending Data

Sending data from the Client to the Server over HTTP protocol.

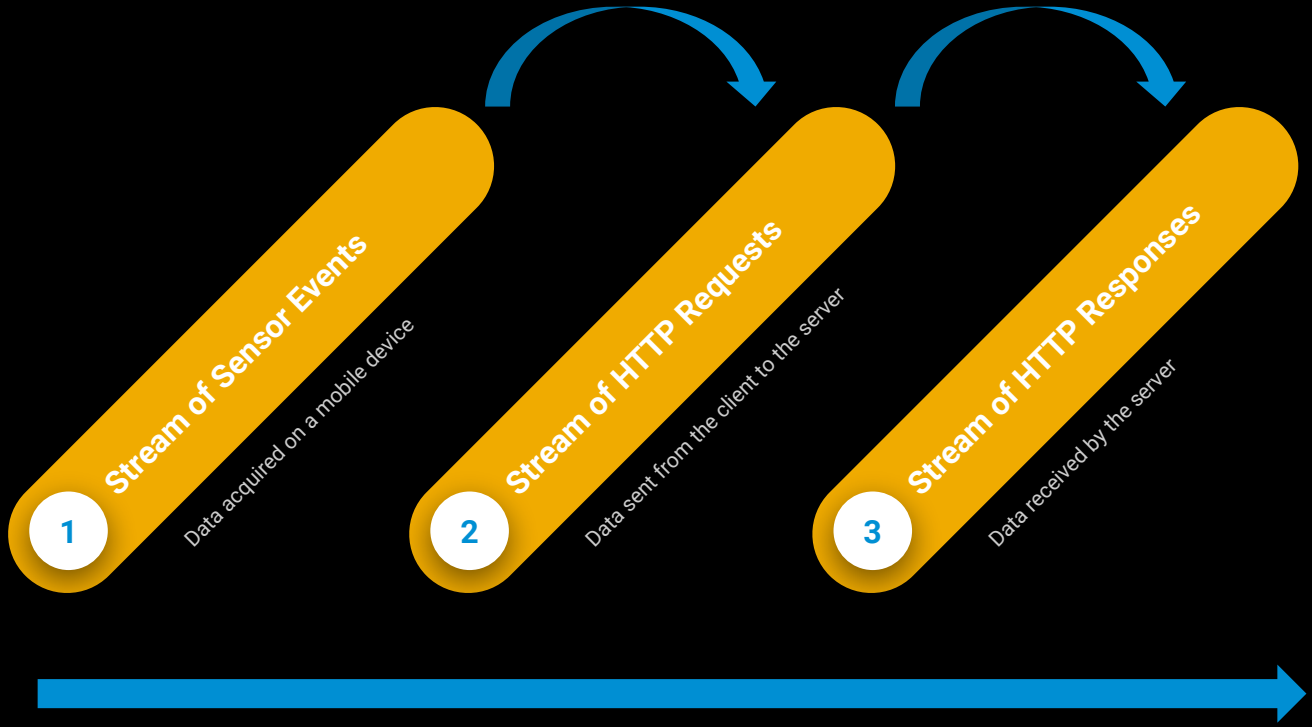
Data Handling

Handling incoming data received from the Client.

Data Processing

Processing handled data and logging them.

Flow of the data streams inside the application



Continuous stream of the data without breaking a chain

Technologies we are going to use

- **Server**
  - Gradle
  - Java 9
  - RxJava 2
  - Vert.x
    - Core
    - Rx
  - Logback
- **Client**
  - Gradle
  - Android
  - Java 7
  - Retrolambda
  - RxJava 2
  - RxAndroid
  - OkHttp 3
  - ReactiveSensors

Side note

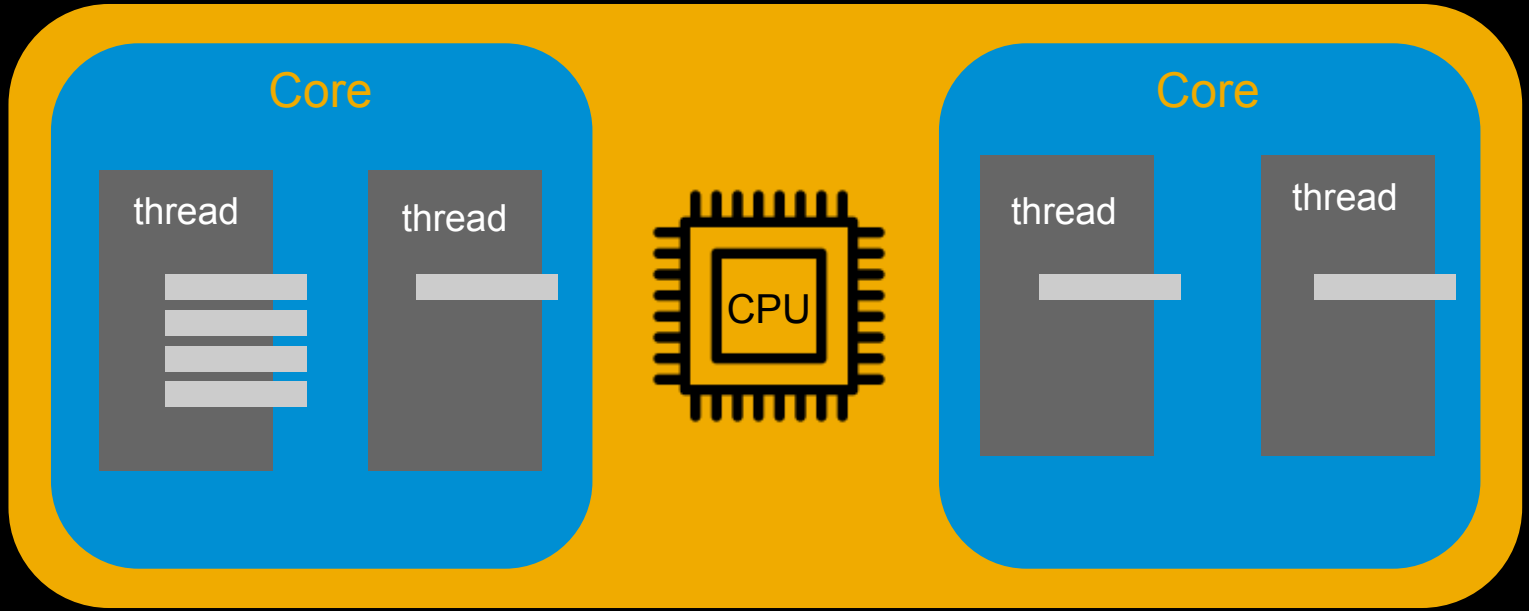
This is just a simple demo with the only one endpoint.
For more endpoints it's better to use
Vertx-web on the server and Retrofit on the client.

Let’s make an application!

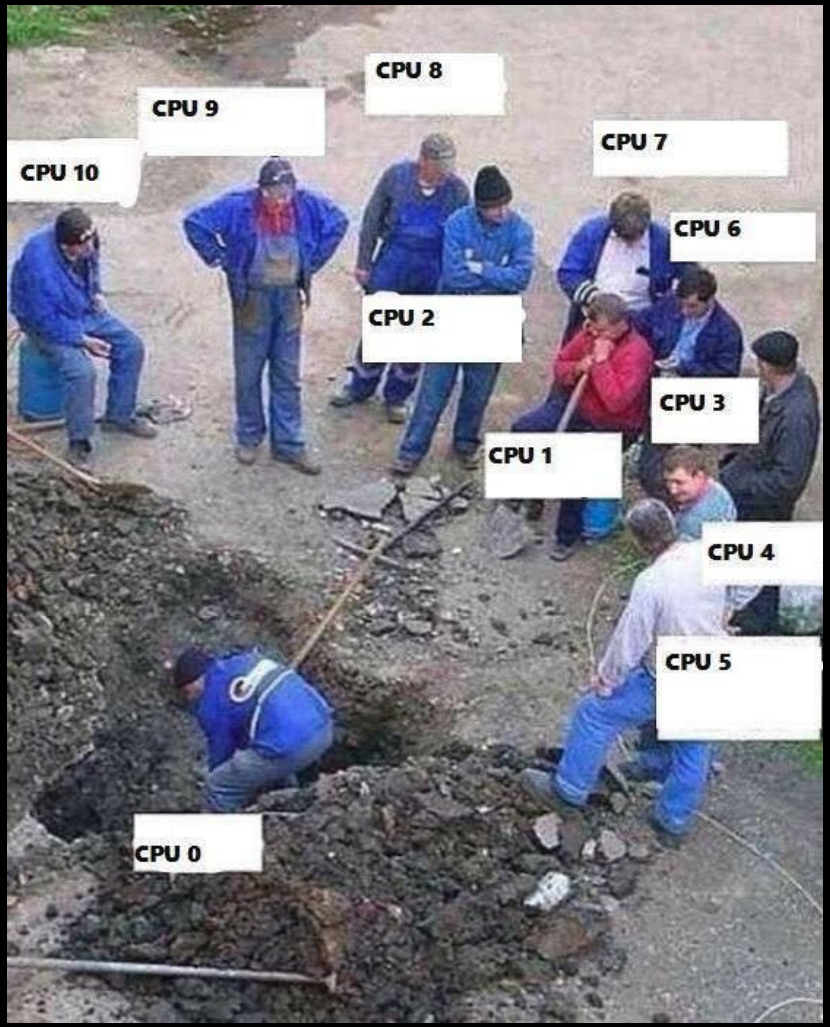
Source code of this project can be found at:
https://github.com/pwittchen/reactive-client-server/

Optimizing CPU usage with RxJava

Optimizing CPU usage



If we are not using all available cores efficiently, we're wasting resources and applications are getting slow.


CPU 8
CPU 9
CPU 7
CPU 10
CPU 6
CPU 2
CPU 3
CPU 1
CPU 4
CPU 5
CPU 0

Source code of this project can be found at:
https://github.com/pwittchen/java-flow-experiments

Thank you

Contact information:

Piotr Wittchen, Arkadiusz Galwas

piotr.wittchen@sap.com, arkadiusz.galwas@sap.com

Software Developers

SAP Hybris