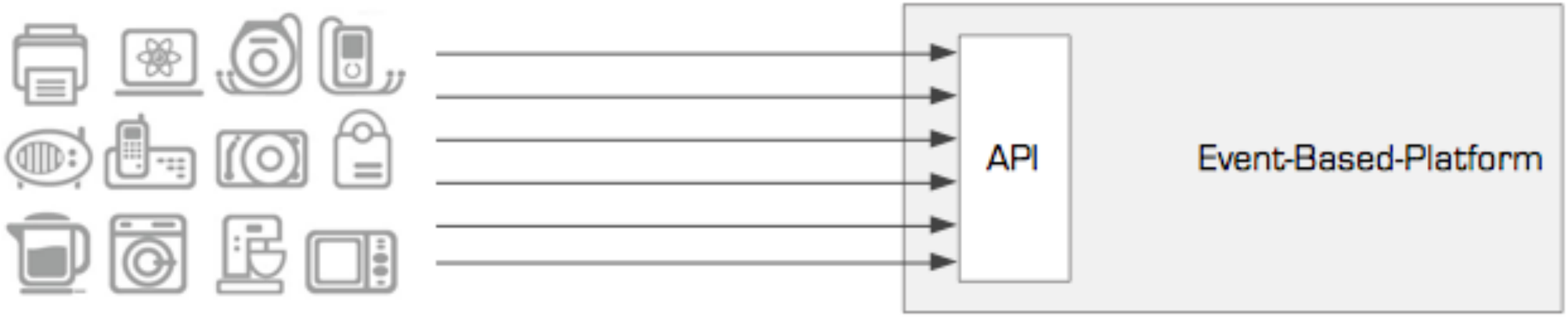
(Cognitive) Event-Handling for OpenWhisk

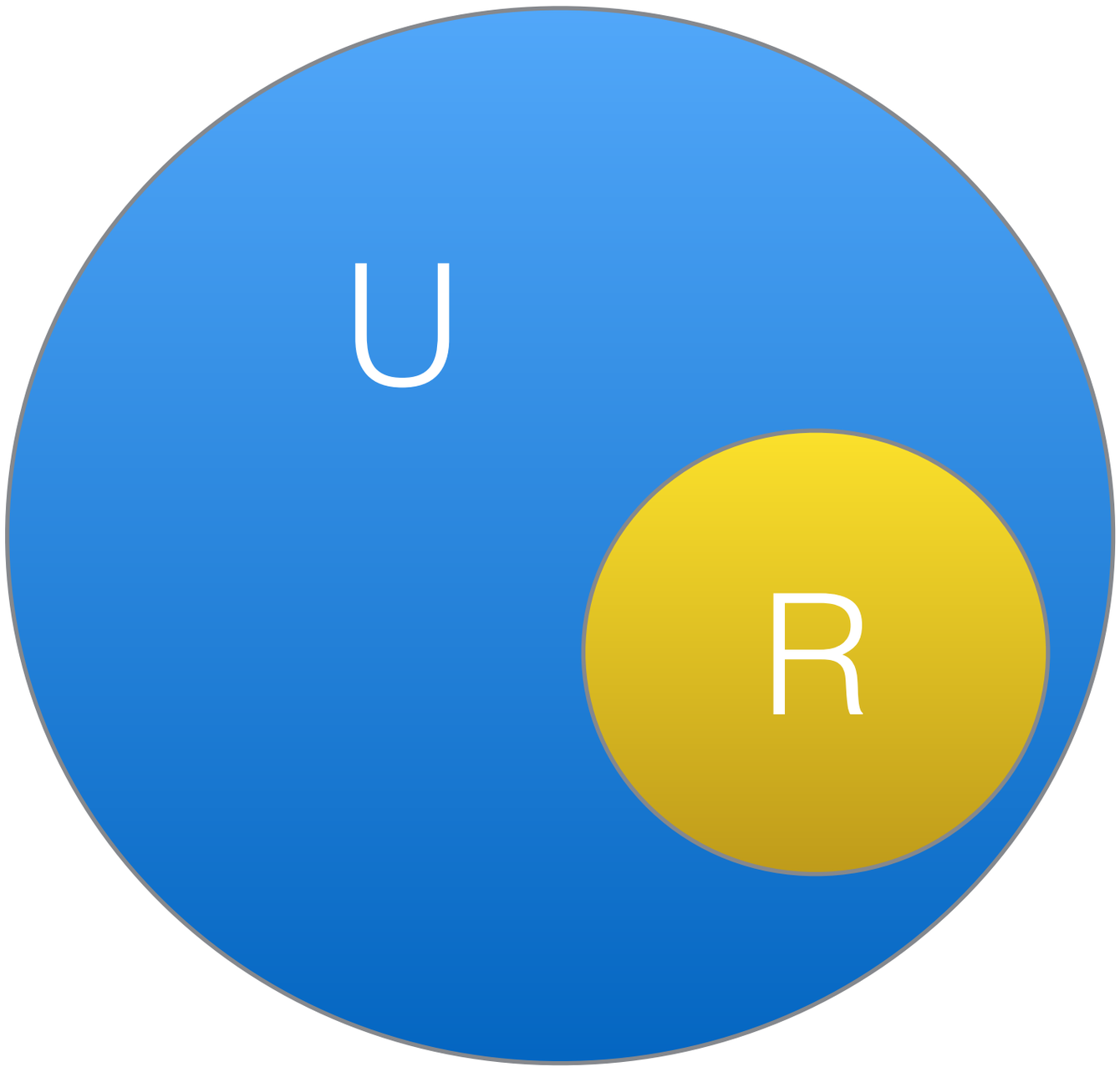
Julian Skupnjak (aka Julz)
Charalampos Georgias (aka Babis)
Öznur Öner (aka Özi)

Project

Problem Statement



Huge Amount of Events
Each Event Triggers an Action!

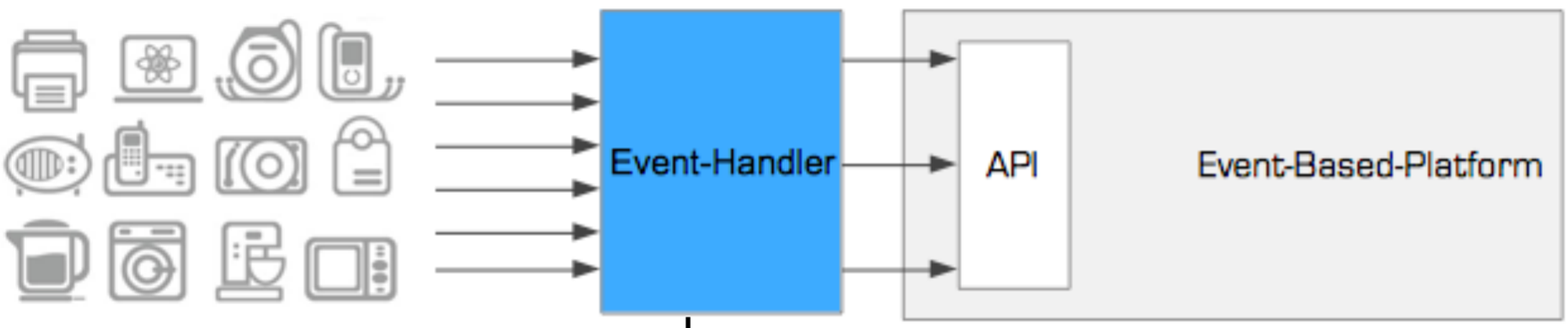


U={All Events}

R={Relevant Events}

R⊆U

Proposed Solution



Cognitive

- Event-Filtering
- Event-Aggregation
- Event-Masking
- Event-De-Duplication
- Event-Correlation

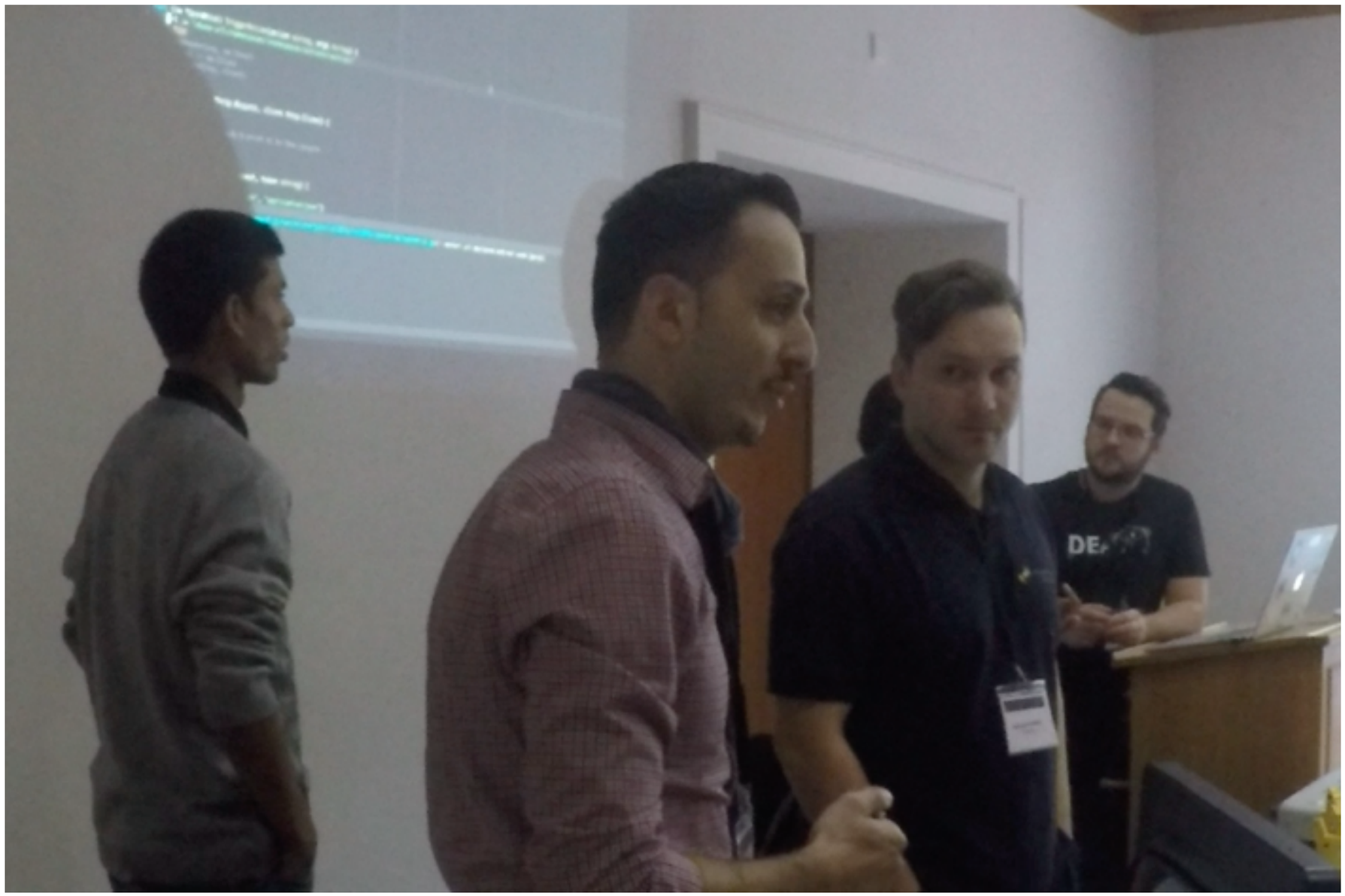
Benefits & Technicity

- **More efficient** (i.e. higher scalability) and intelligent handling of events
- **Reduced network load and code activations** on event-aased platforms (e.g. OpenWhisk)
- **Improved performance** by not overloading the event-action platform with neglectable events
- **Higher cost-efficiency** as neglectable events do not cause the execution of actions anymore
- **Enhancement of the programming model**, e.g. higher level of expressive power

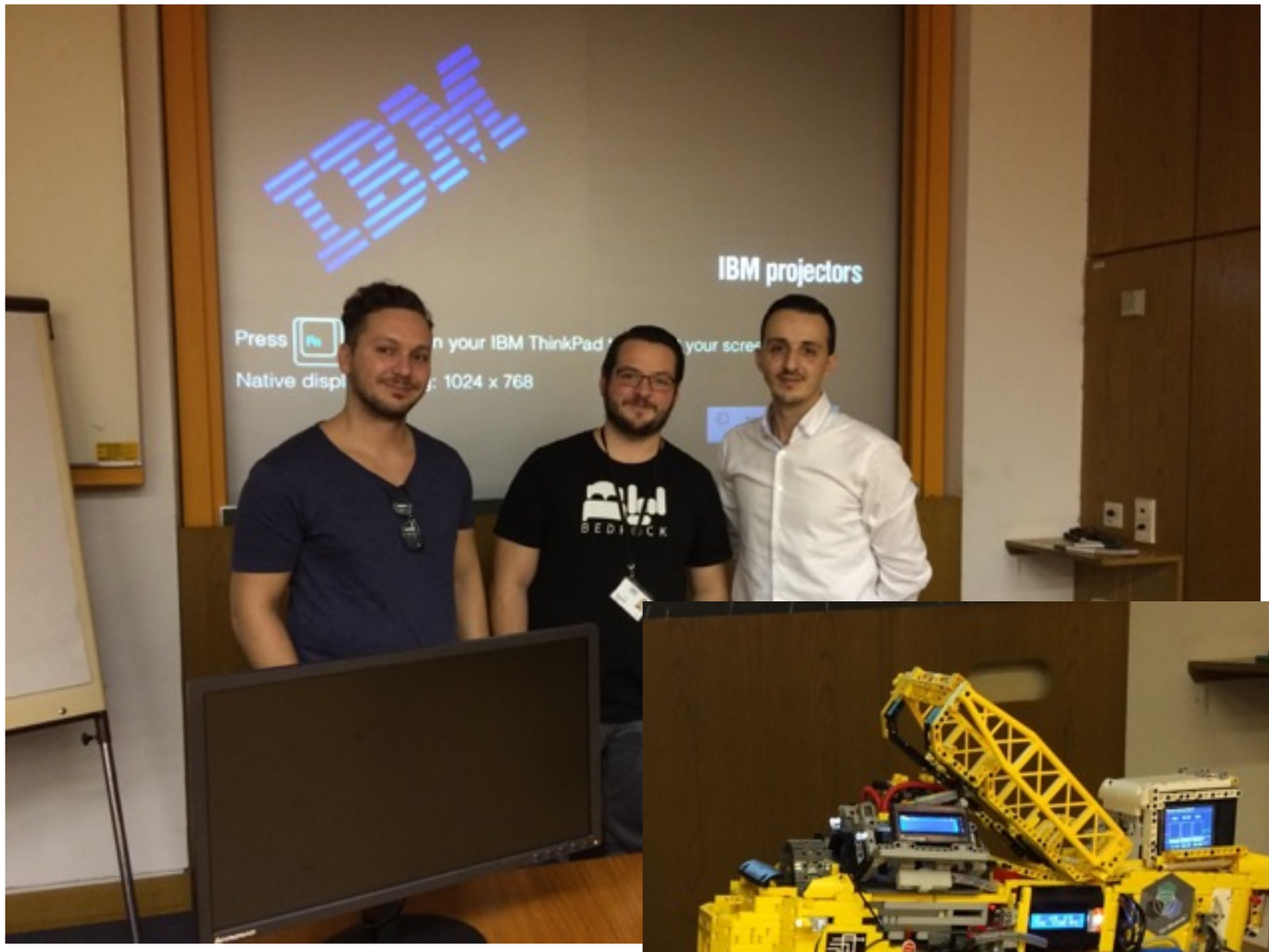
Presentations

IoT2016 Conference

OW Tutorial in Stuttgart

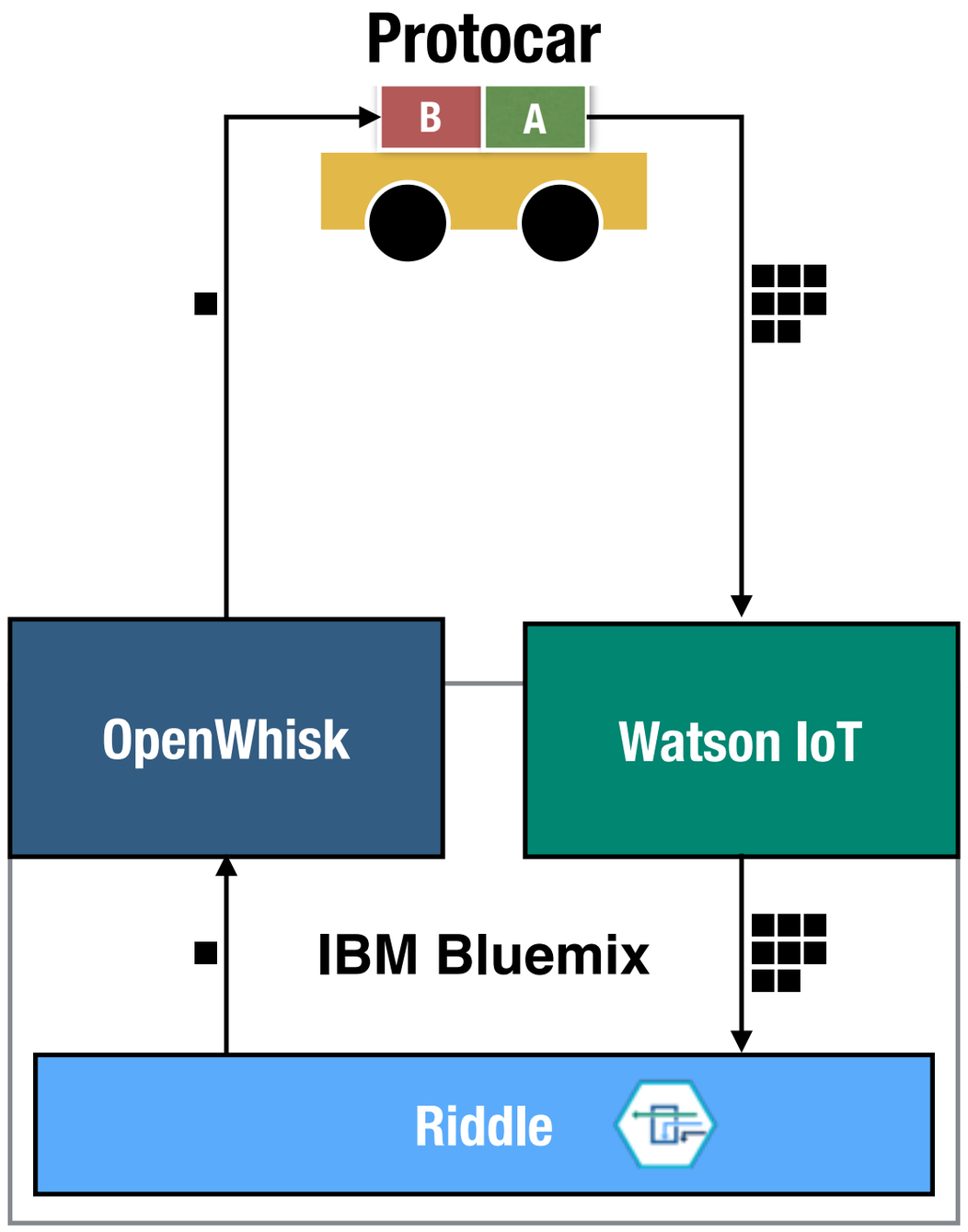


Presentation @IBM R&D



Scenario

Scenario



Riddle

User Interface

IoT-Gateway (Watson IoT)

Controller

Subscribe

Controller

EBP-Trigger

API

DB

Event-Handling Chain

Filter (eg Omnibus)

Correlator (eg Omnibus)

OW

Message Bus

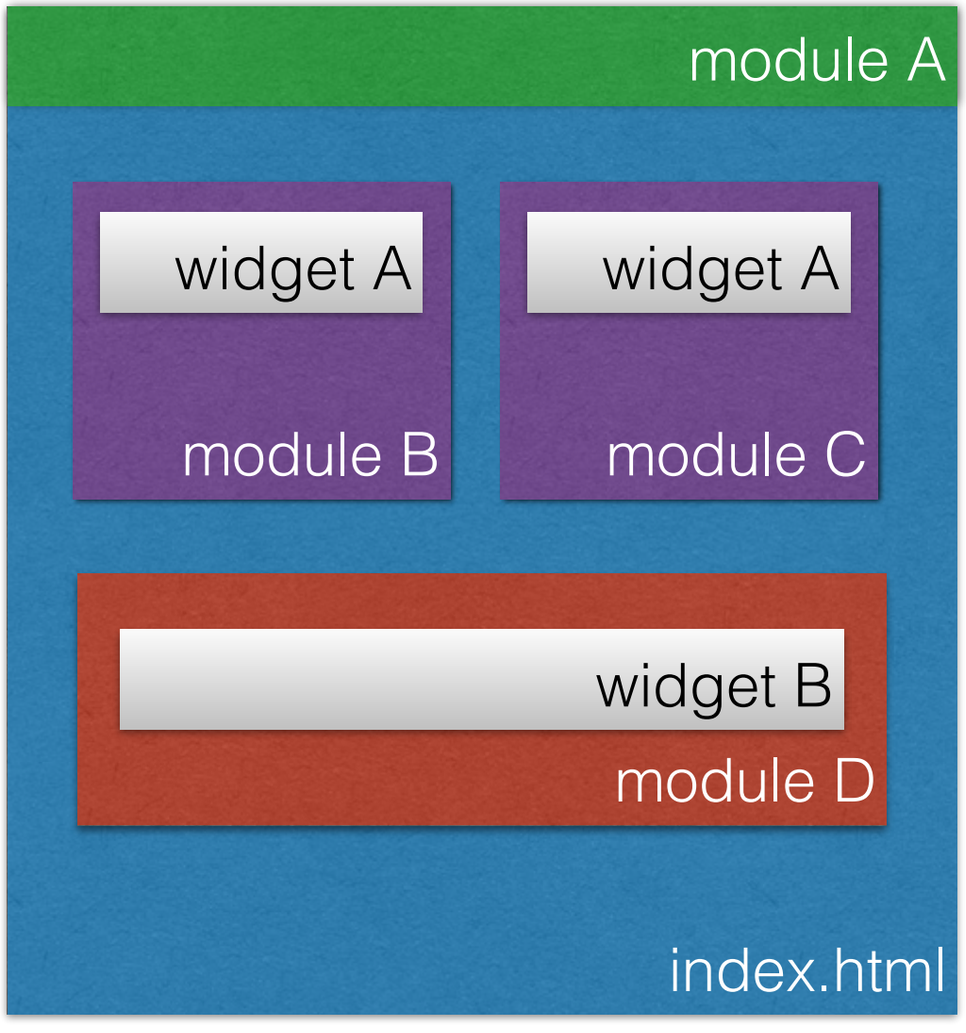
filter queue

correlator queue

EBP-Queue

Components

AngularJs




```
|-public
        |-components/dashboard
        |         |-dashboard.ctr.js
        |         |-dashboard.router.js
        |         |-dashboard.module.js
        |         |-dashboard.html
|-core
        |         |-core.router.js
        |         |-core.module.js
|-lib
        |         |-angularJs
        |         |-bootstrap
|-app.module.js
|-index.html
```

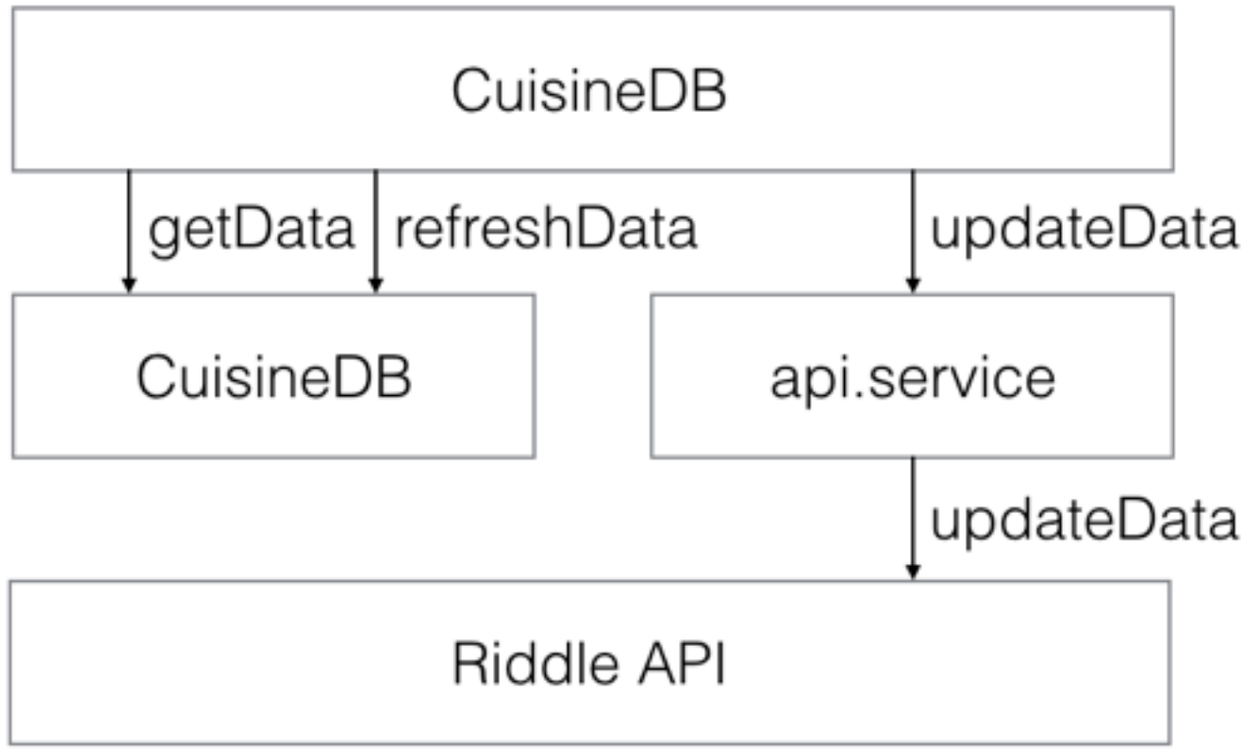
Ui Services & DB

- Configuration of Event-Handler

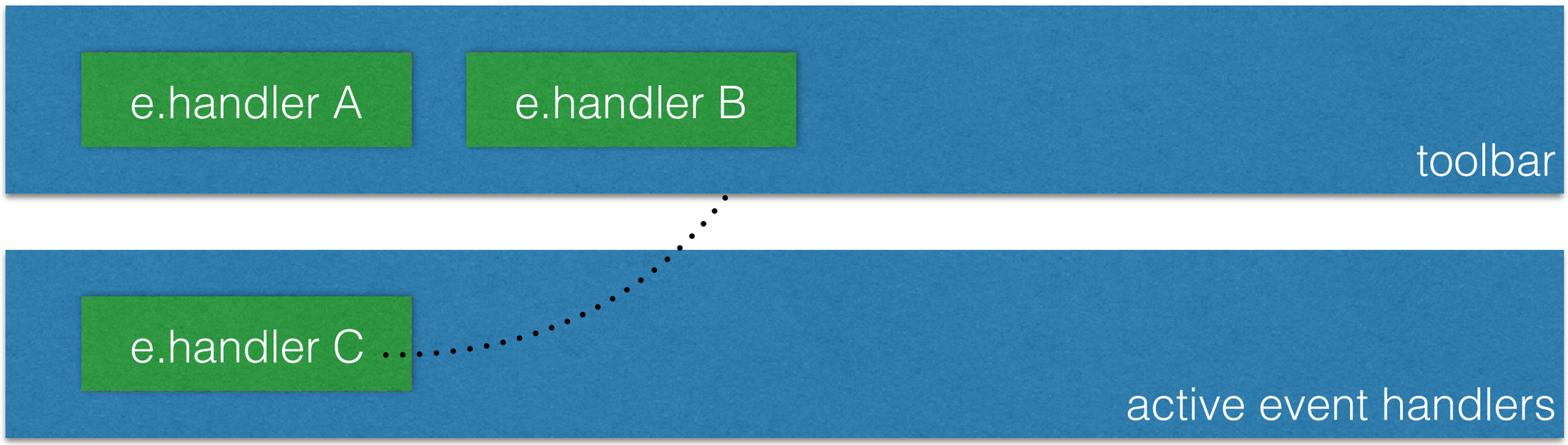
  - save/delete updates the api.service

- Cuisine DB updates the intern UI-Database

- faster processing through database



Other API calls

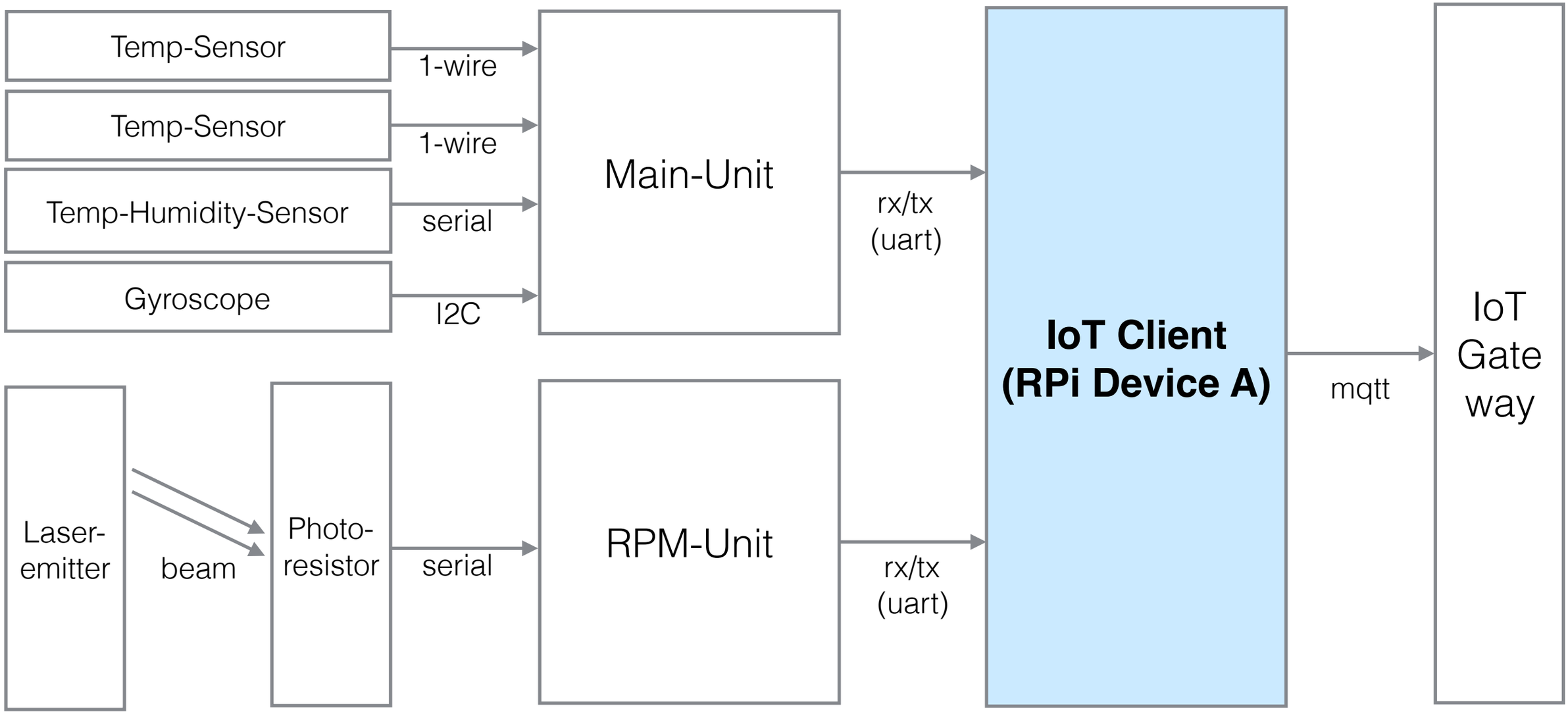


- drag and drop
- by activating/deactivating the handler chain
- web-sockets for incoming and outgoing events

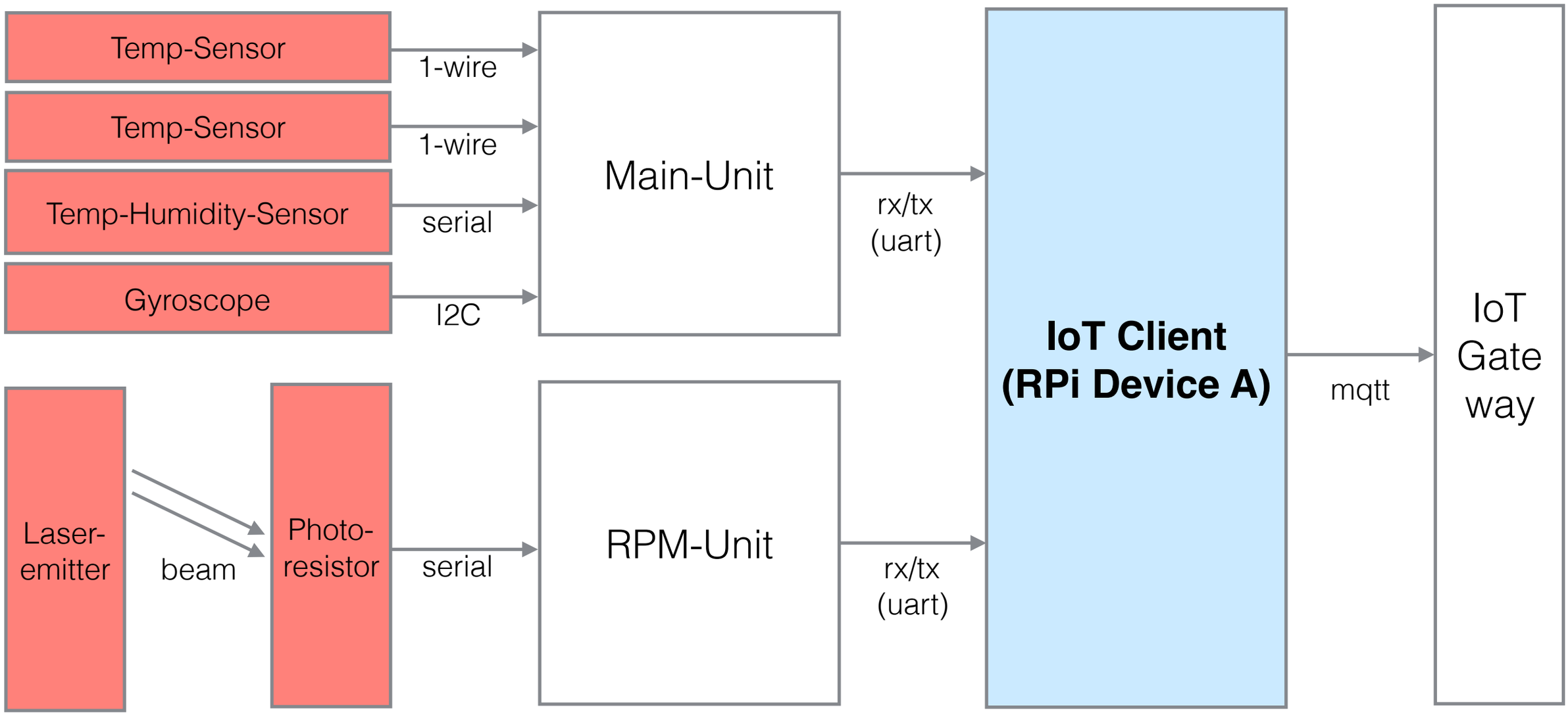
activate of event-handlers

deactivate of event-handlers

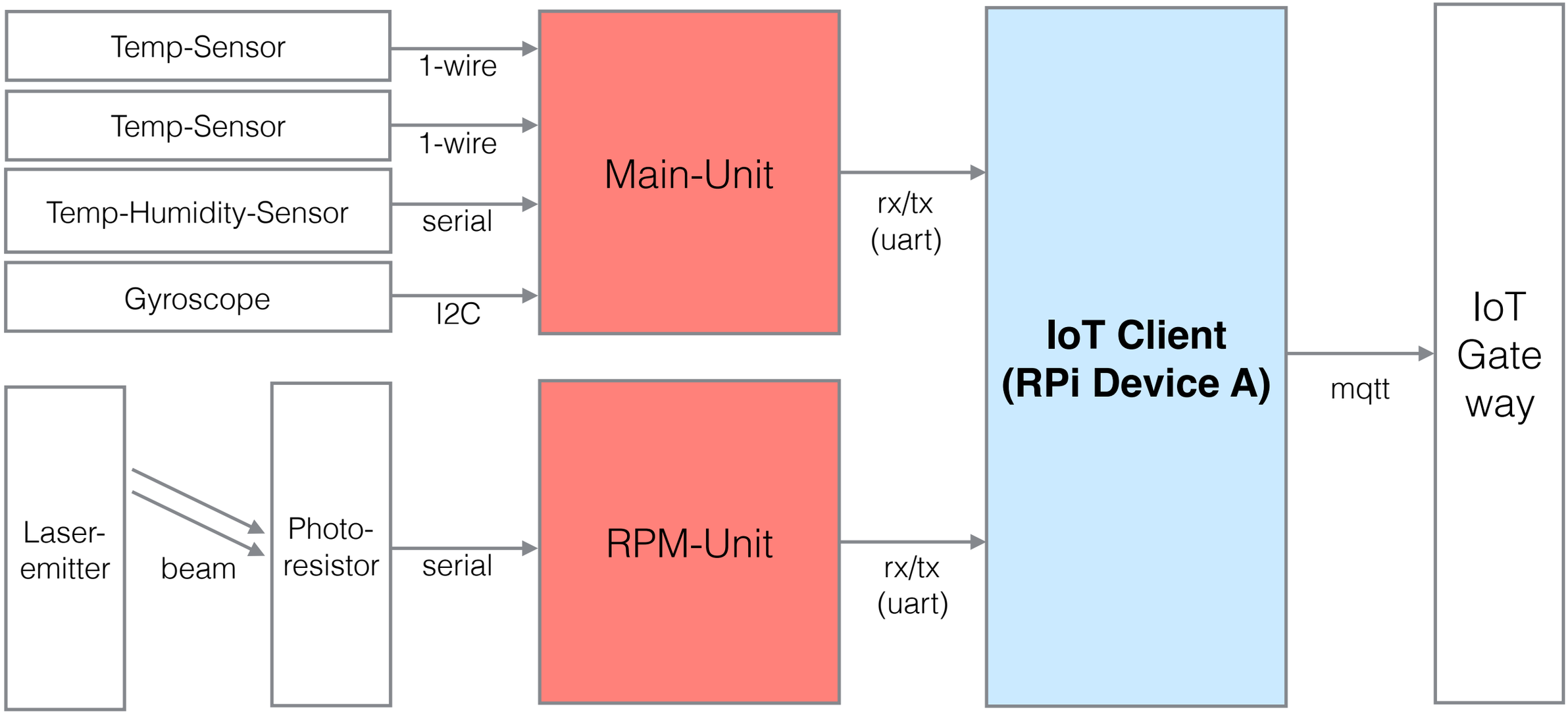
Producer - „Protocar“



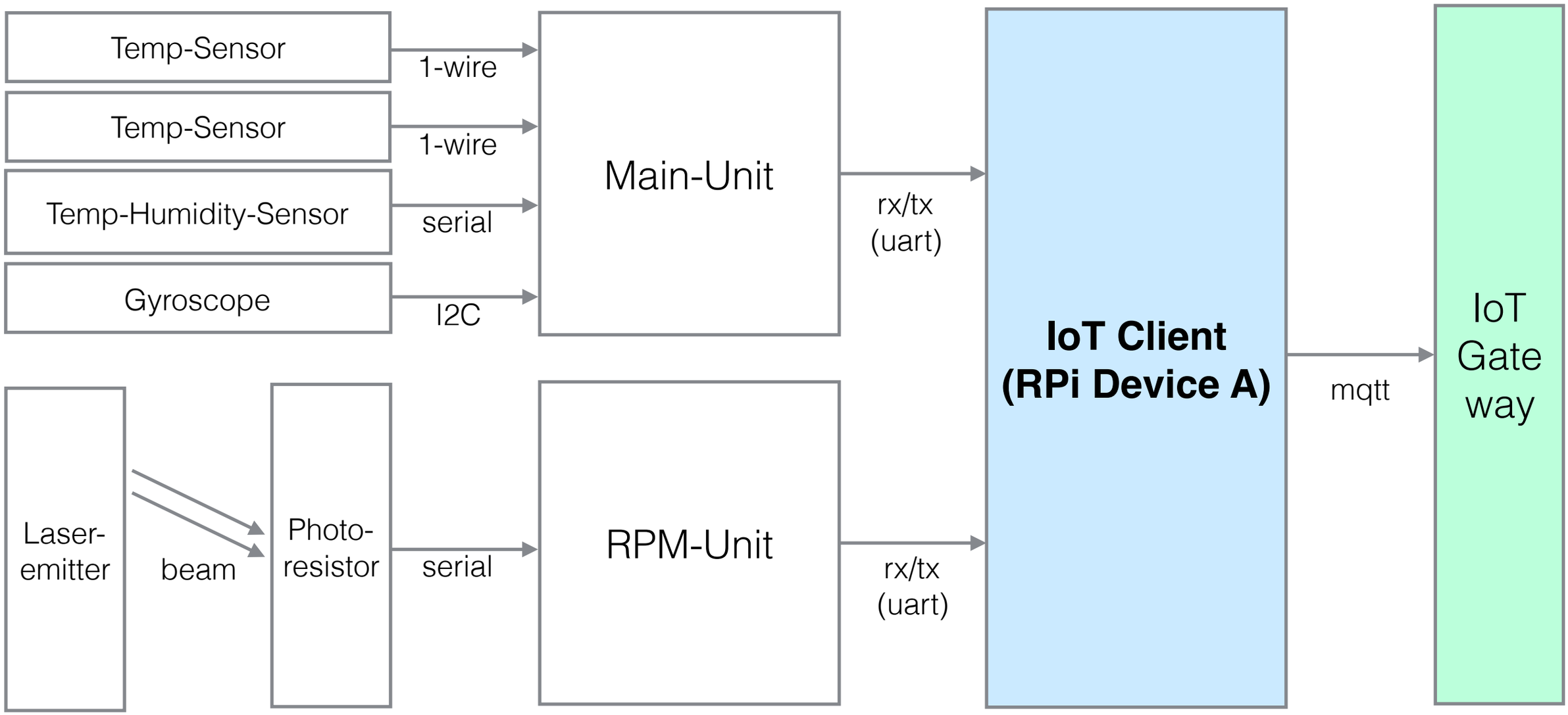
Producer - „Protocar“



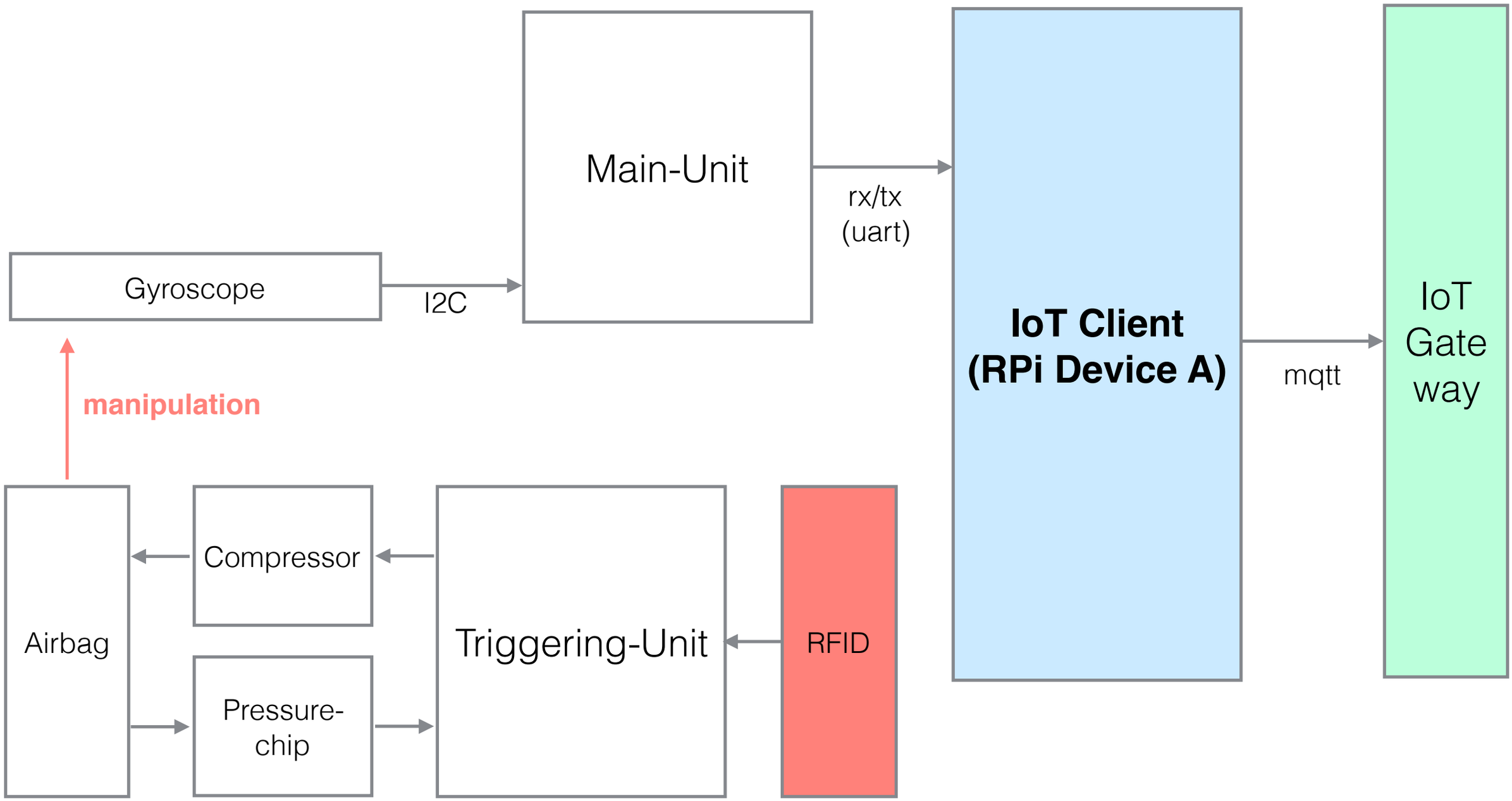
Producer - „Protocar“



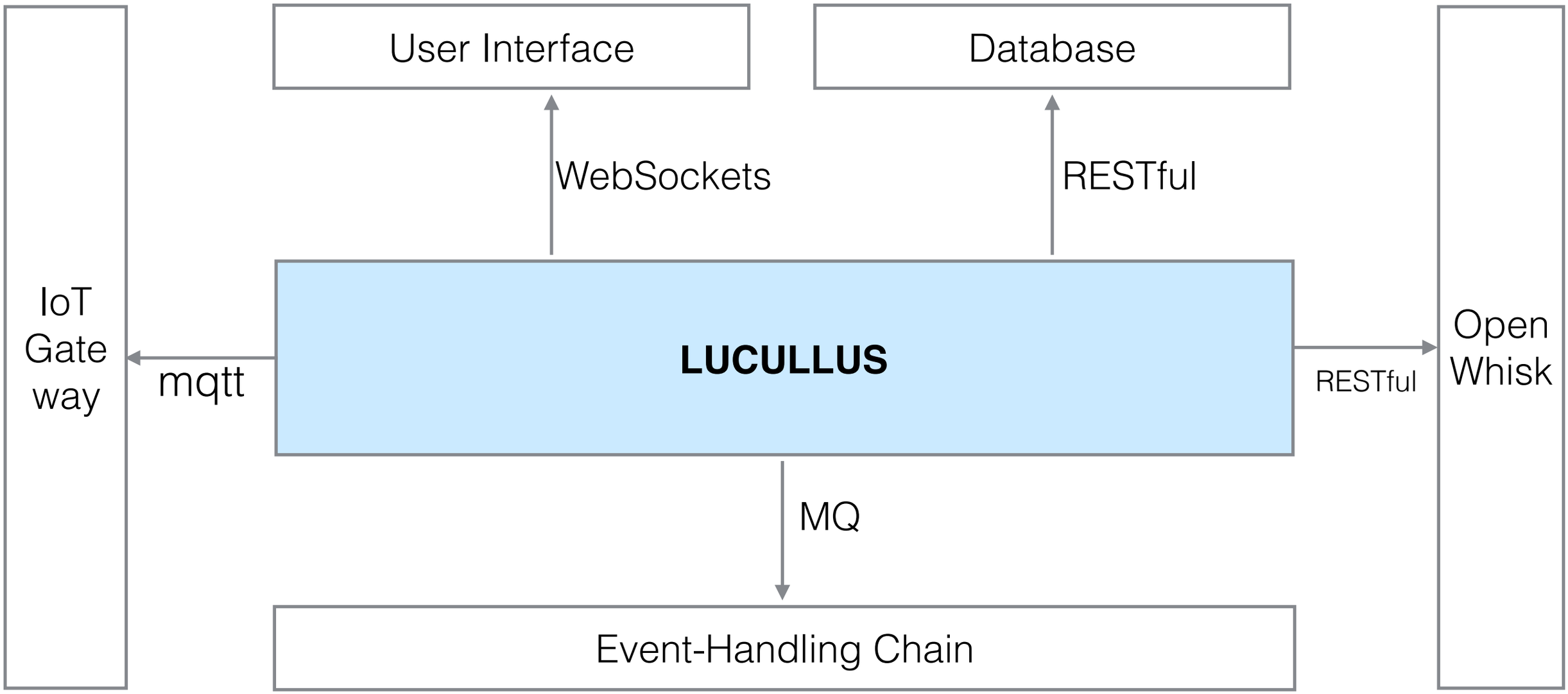
Producer - „Protocar“



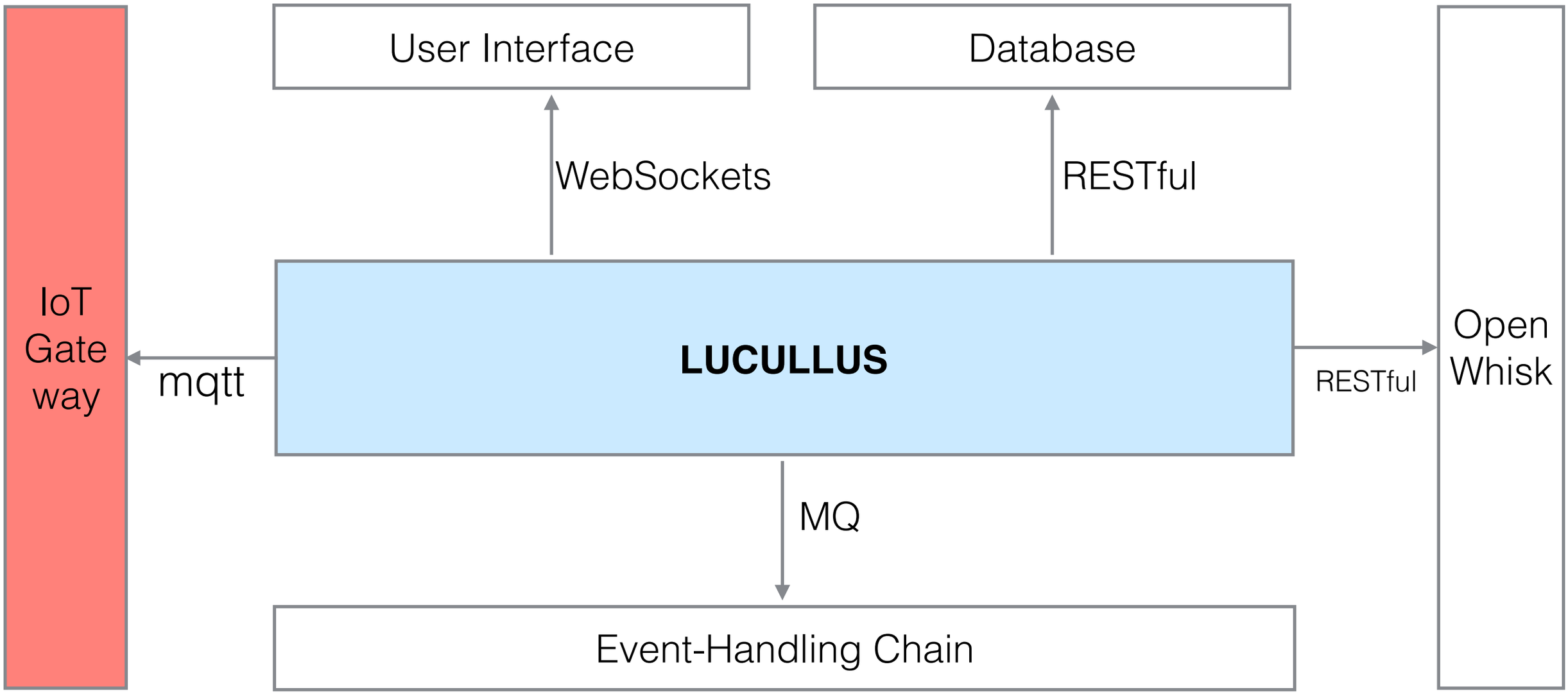
Producer - „Protocar“



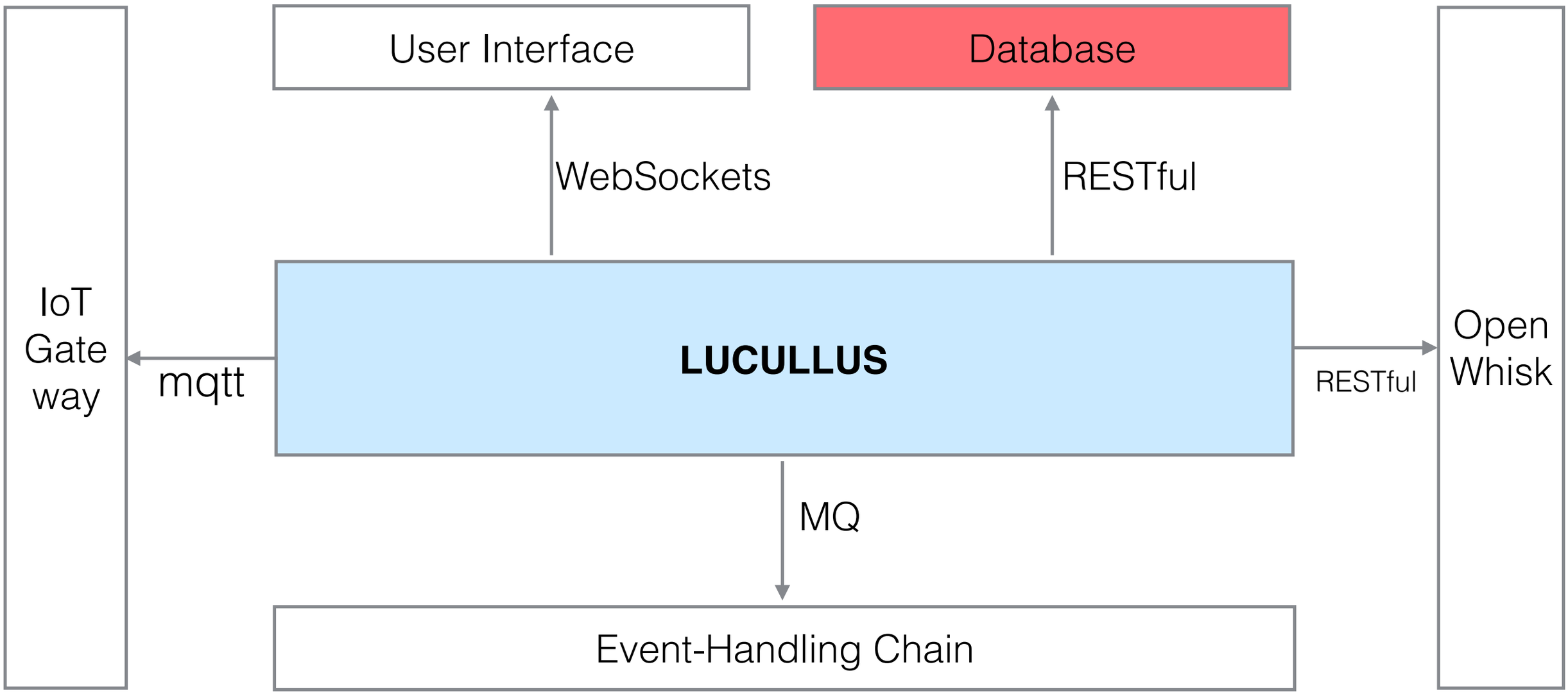
Controller - „Lucullus“



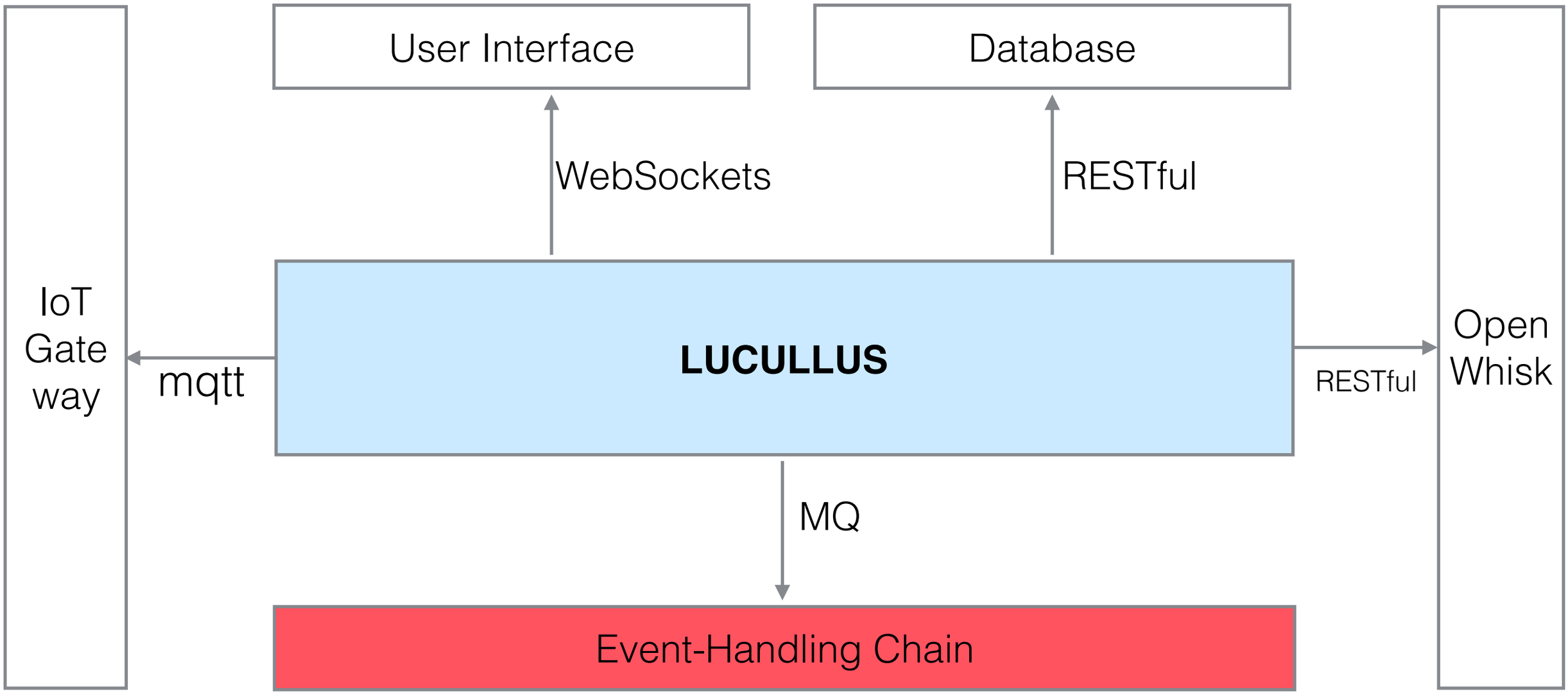
Controller - „Lucullus“



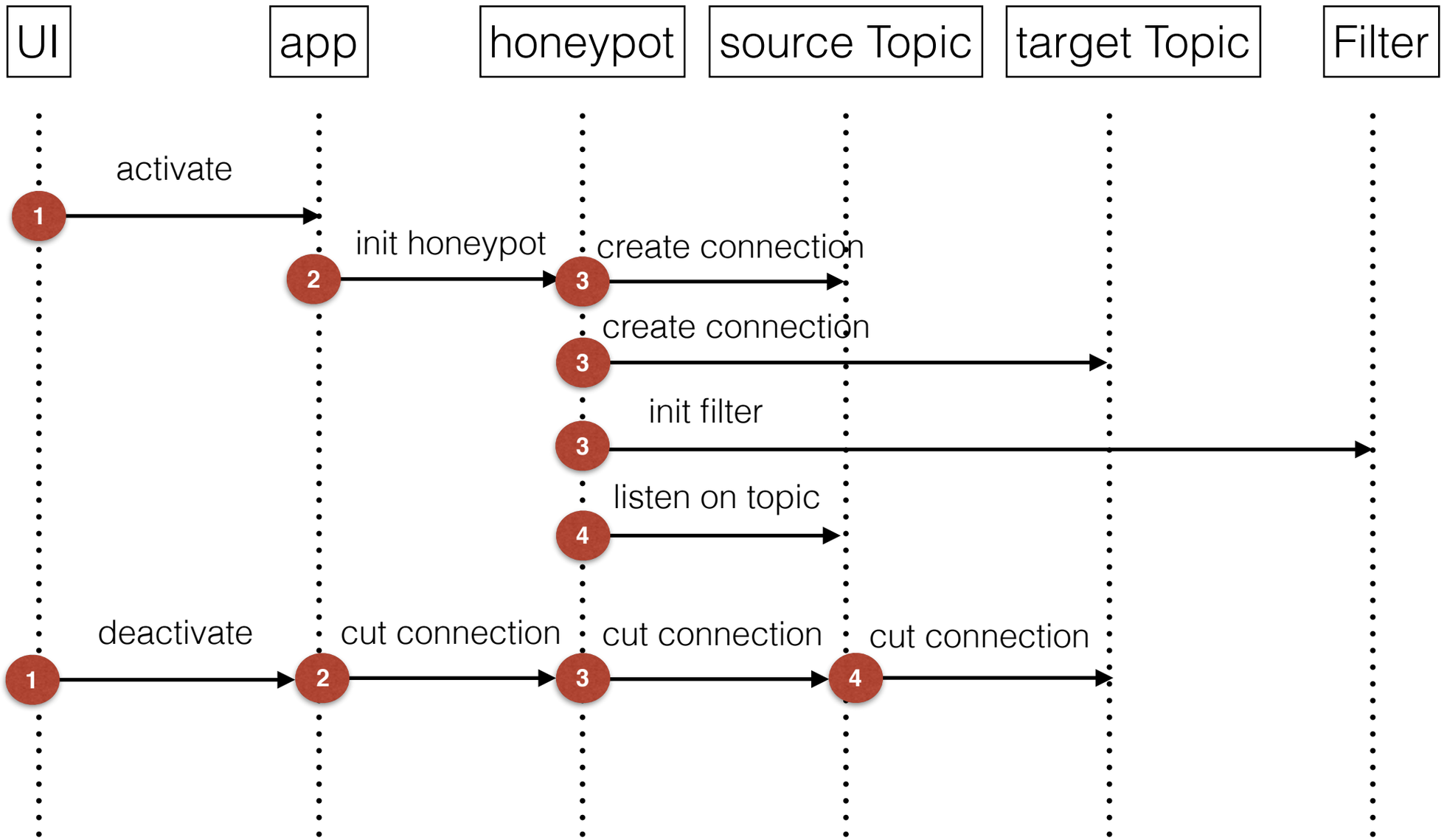
Controller - „Lucullus“



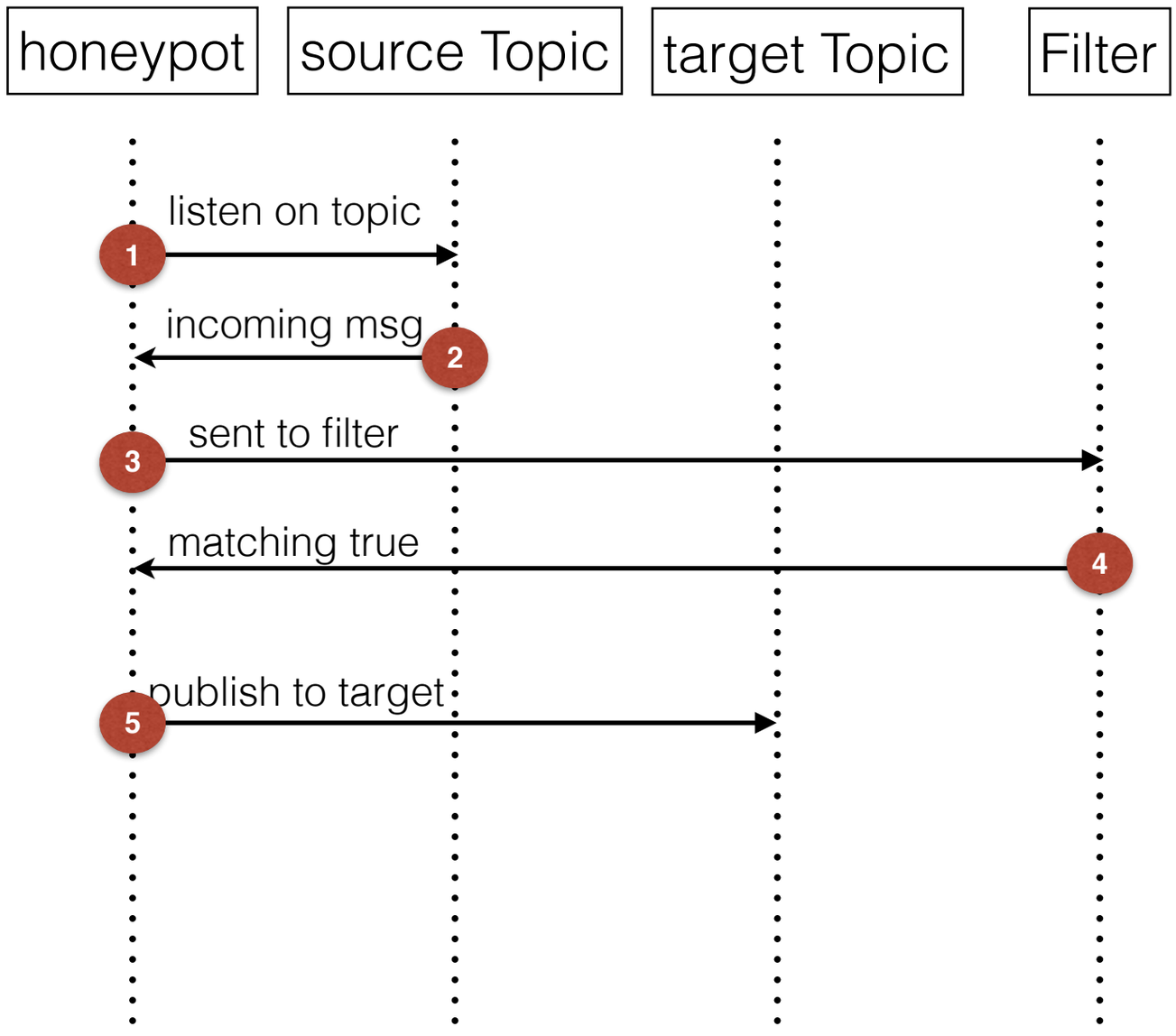
Controller - „Lucullus“



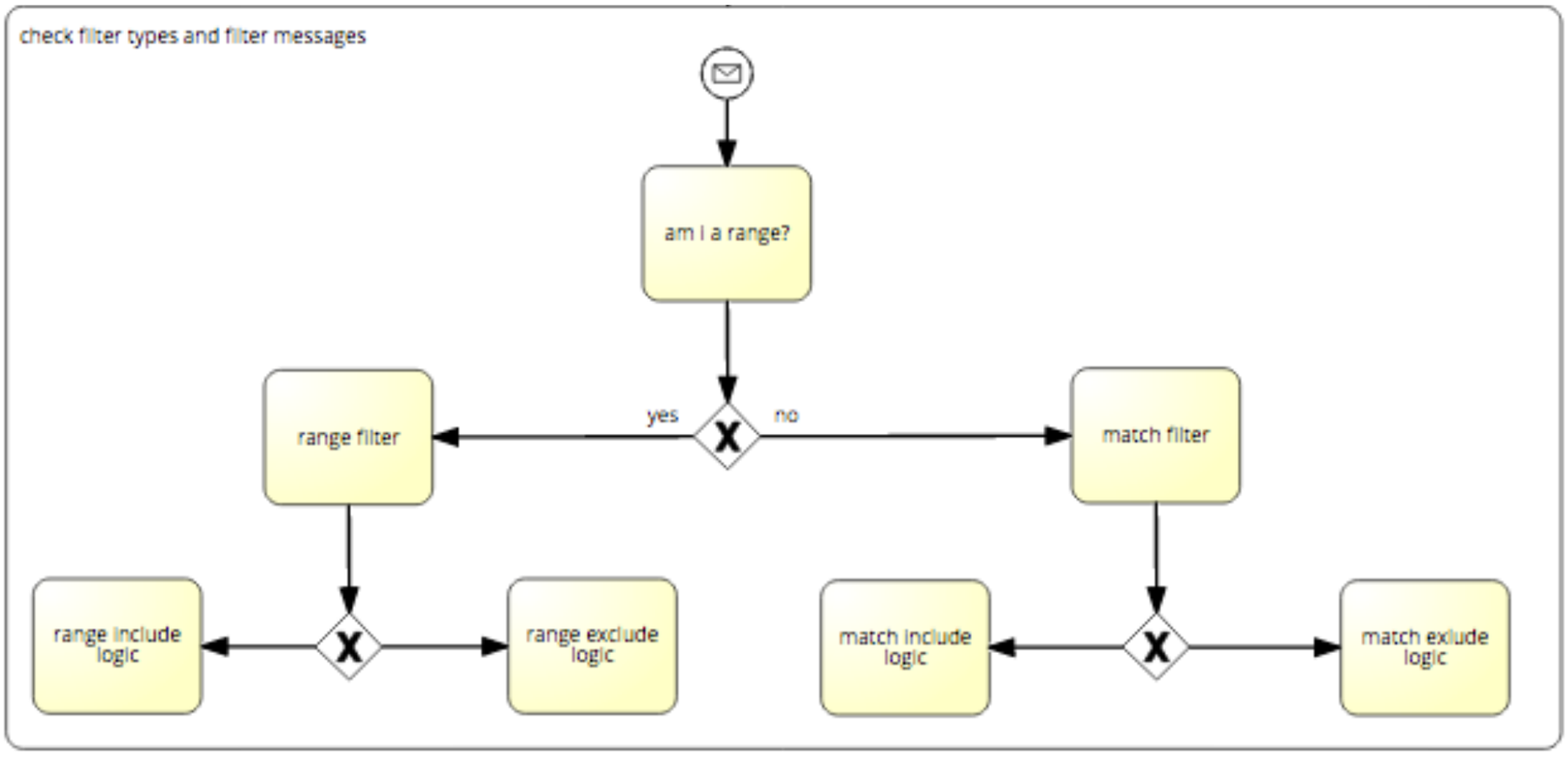
Riddle-Screen



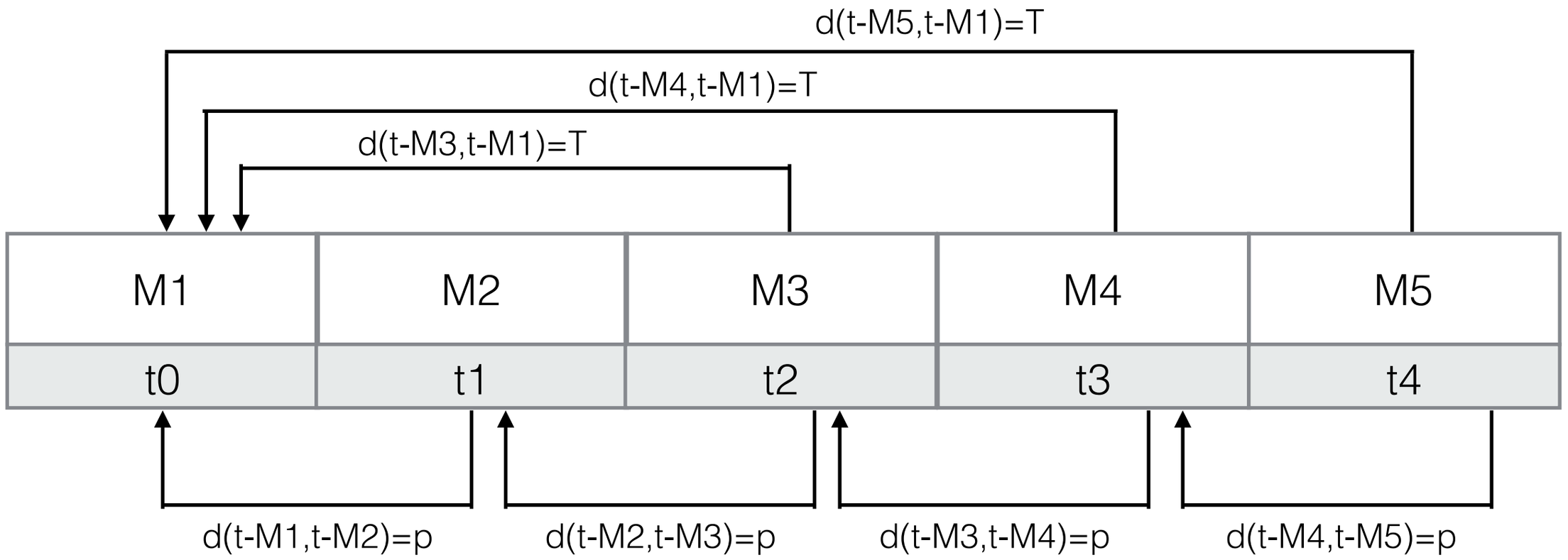
Filter


honeypot
source Topic
target Topic
Filter
listen on topic
1
incoming msg
2
sent to filter
3
matching true
4
publish to target
5

filter logic

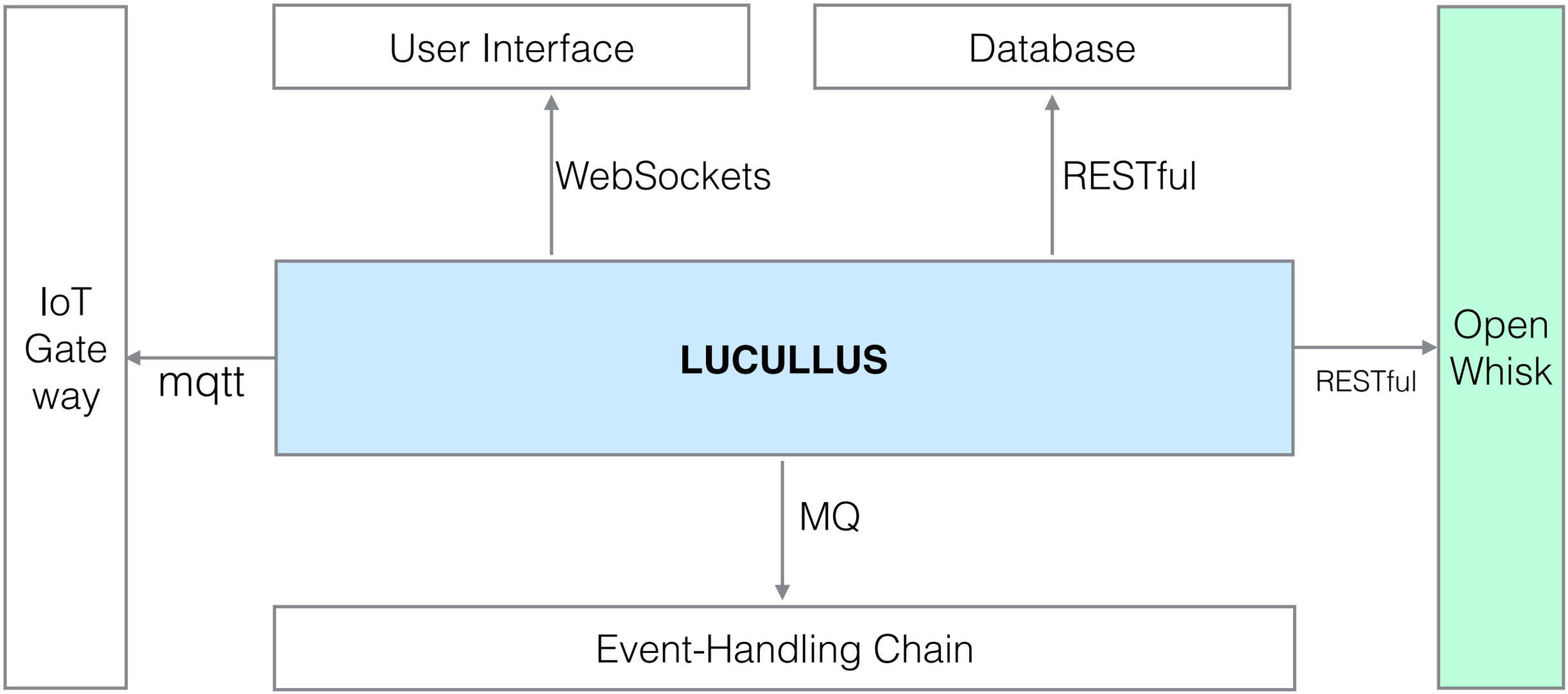


Riddle-Correlator

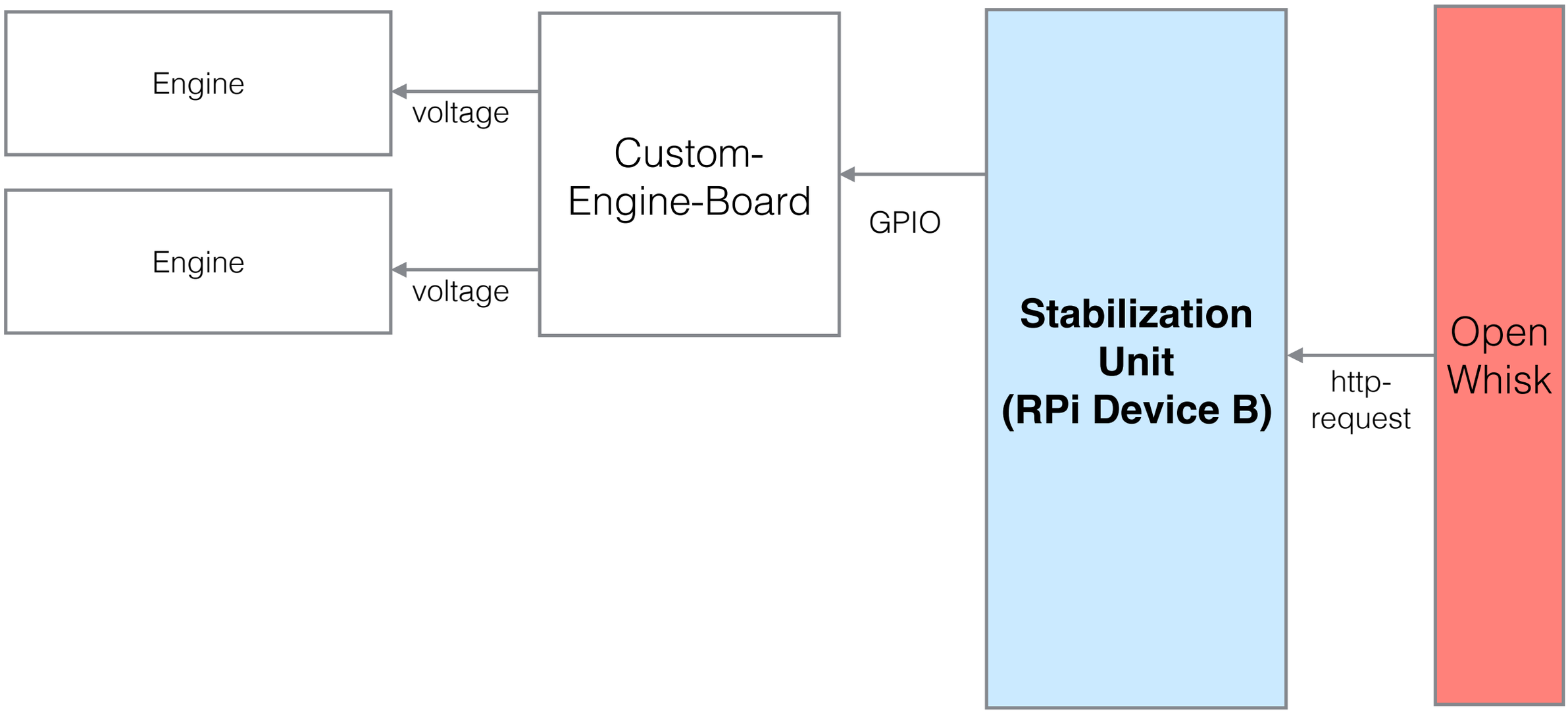


M = Message
t = time
d = time difference
p = pitch
T = Treshold

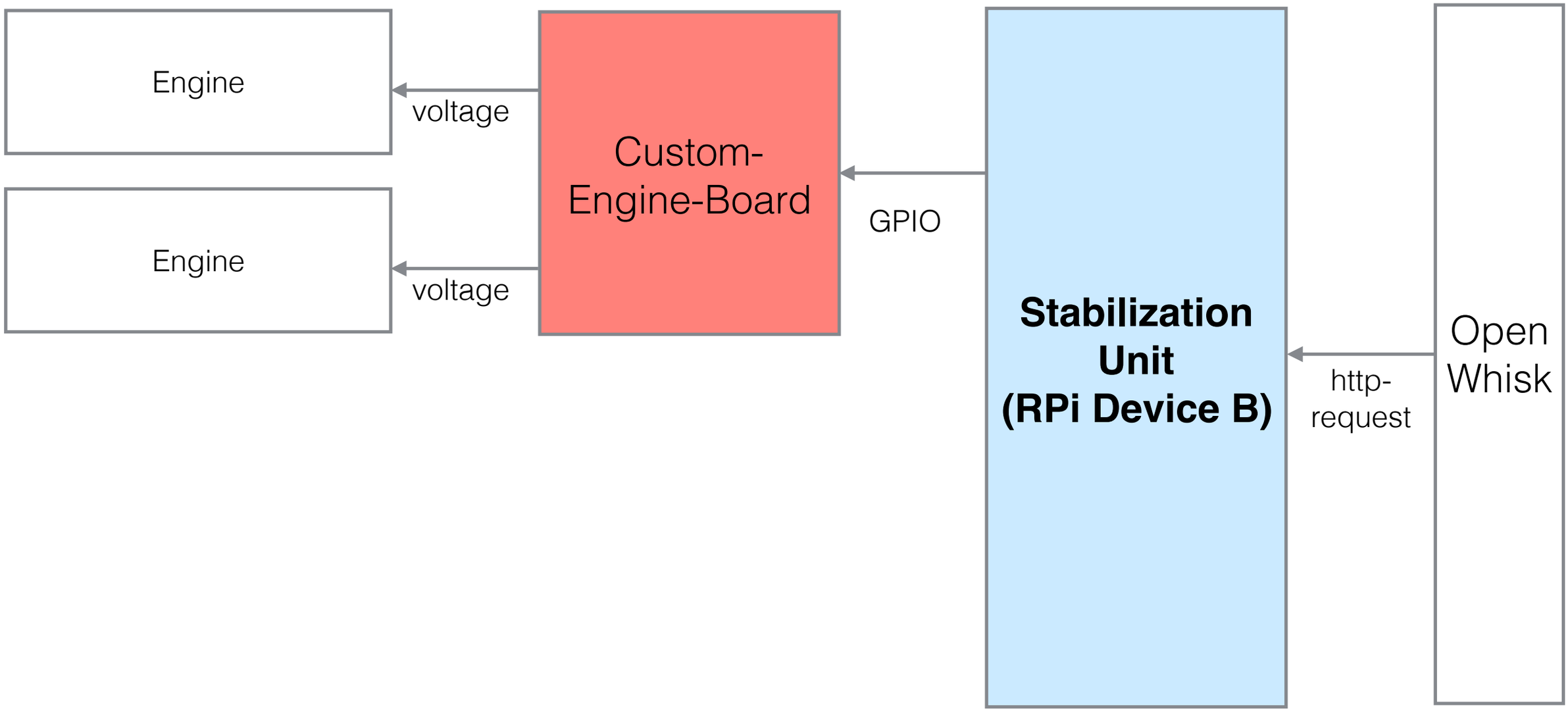
Controller - „Lucullus“



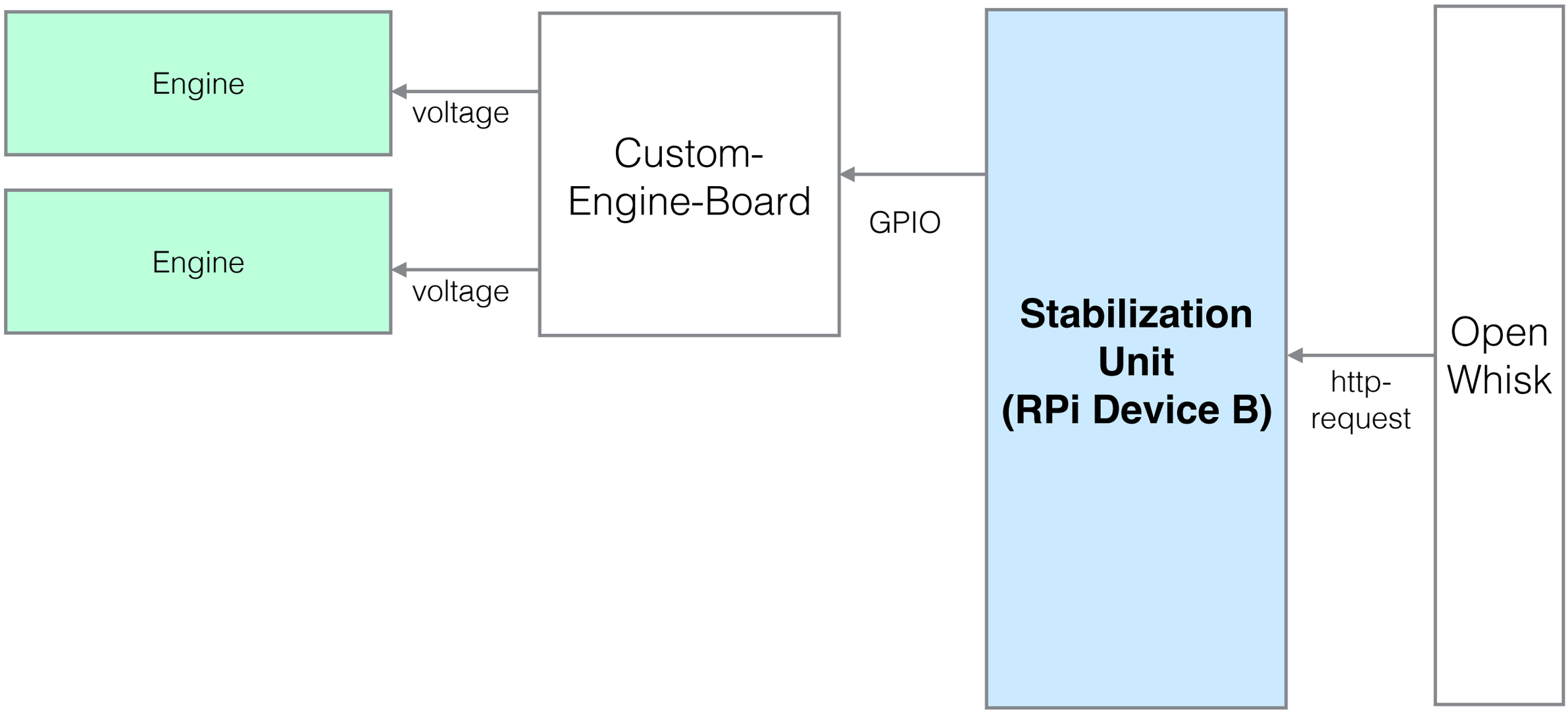
Consumer - „Protocar“



Consumer - „Protocar“

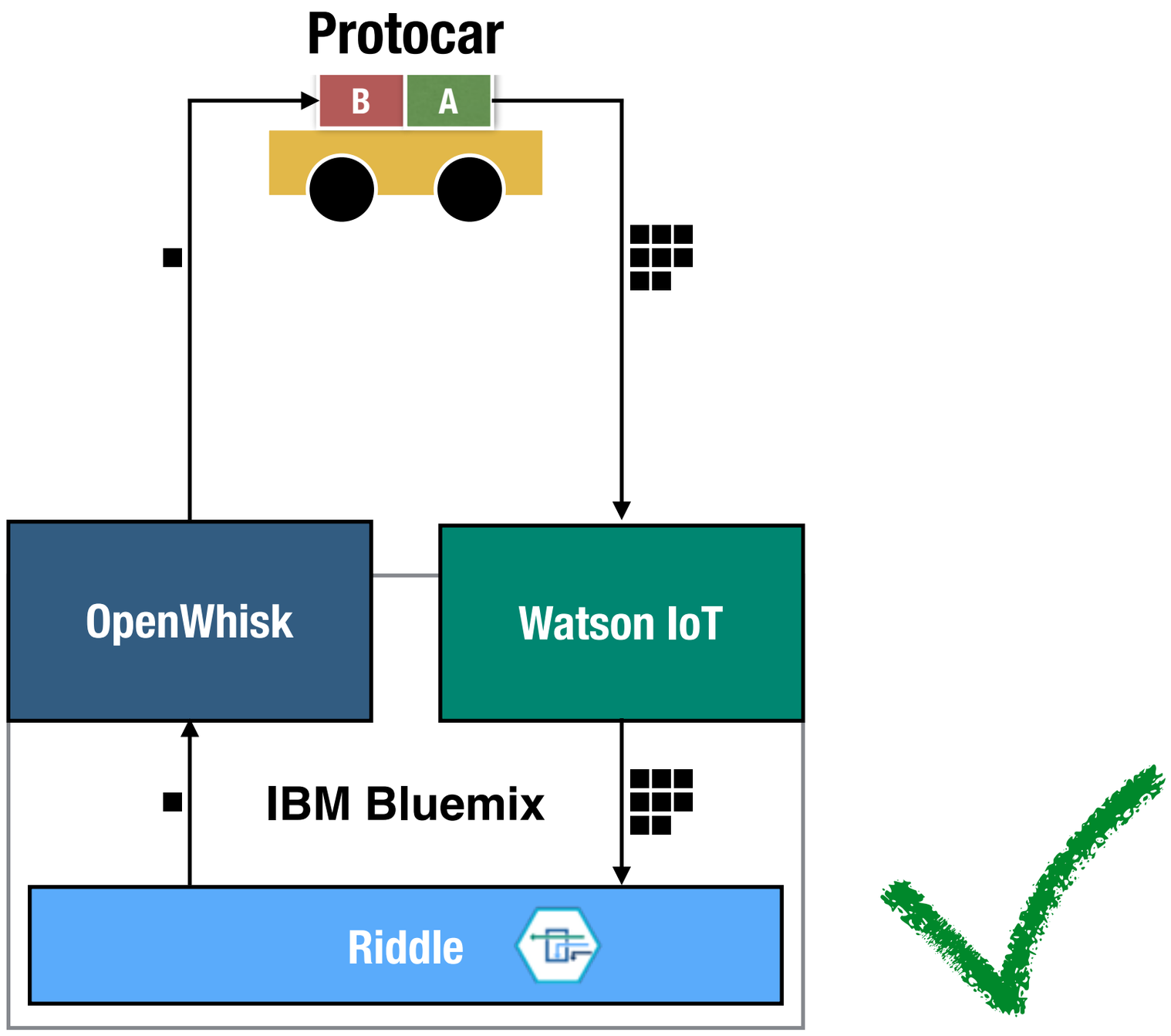


Consumer - „Protocar“



Vision

Conclusion



https://github.com/Event-Riddle/MenuCard/wiki

Demo