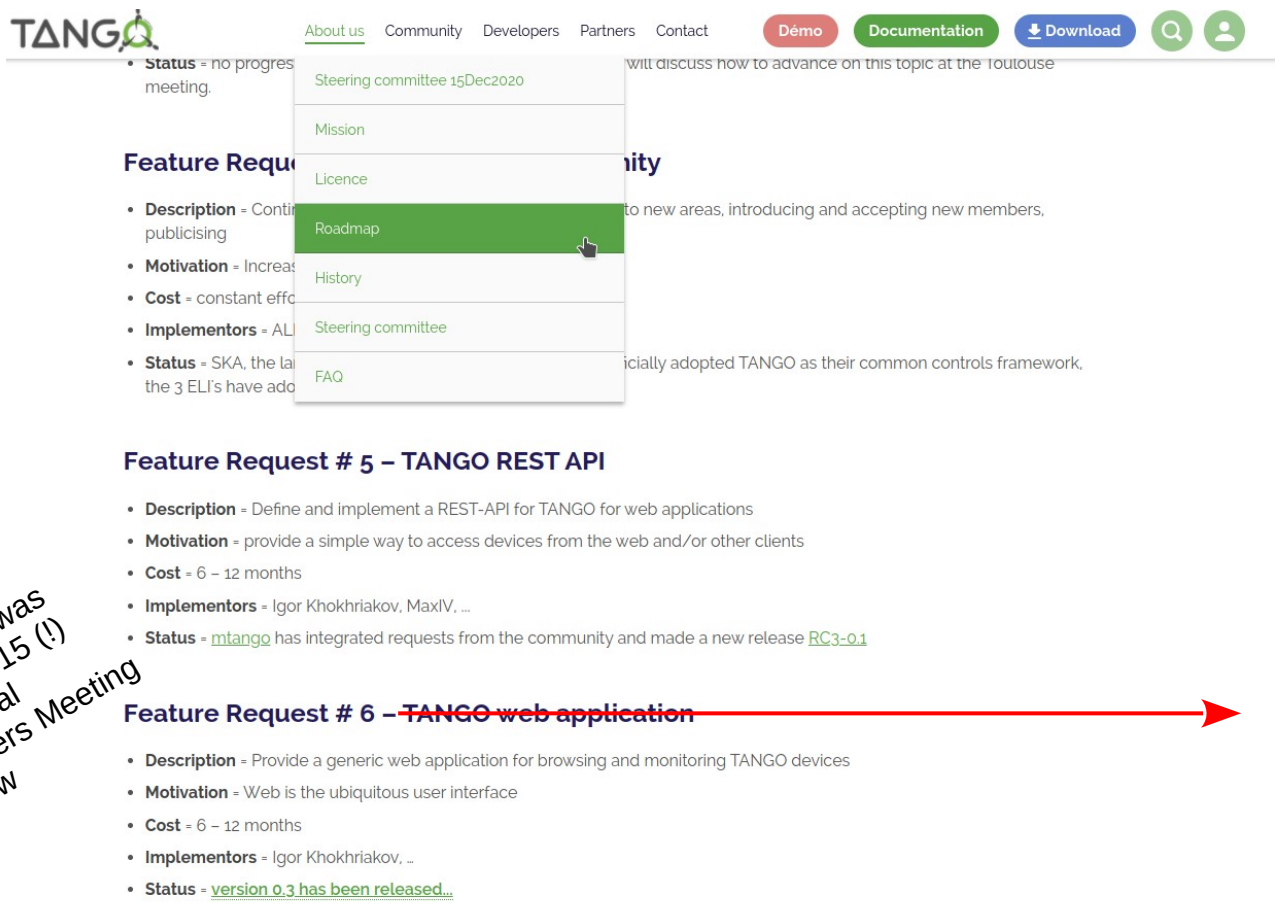
Tango Controls satellite workshop

ICALEPCS 2021

Olga Merkulova

ik-company.com

How did Waltz appear?



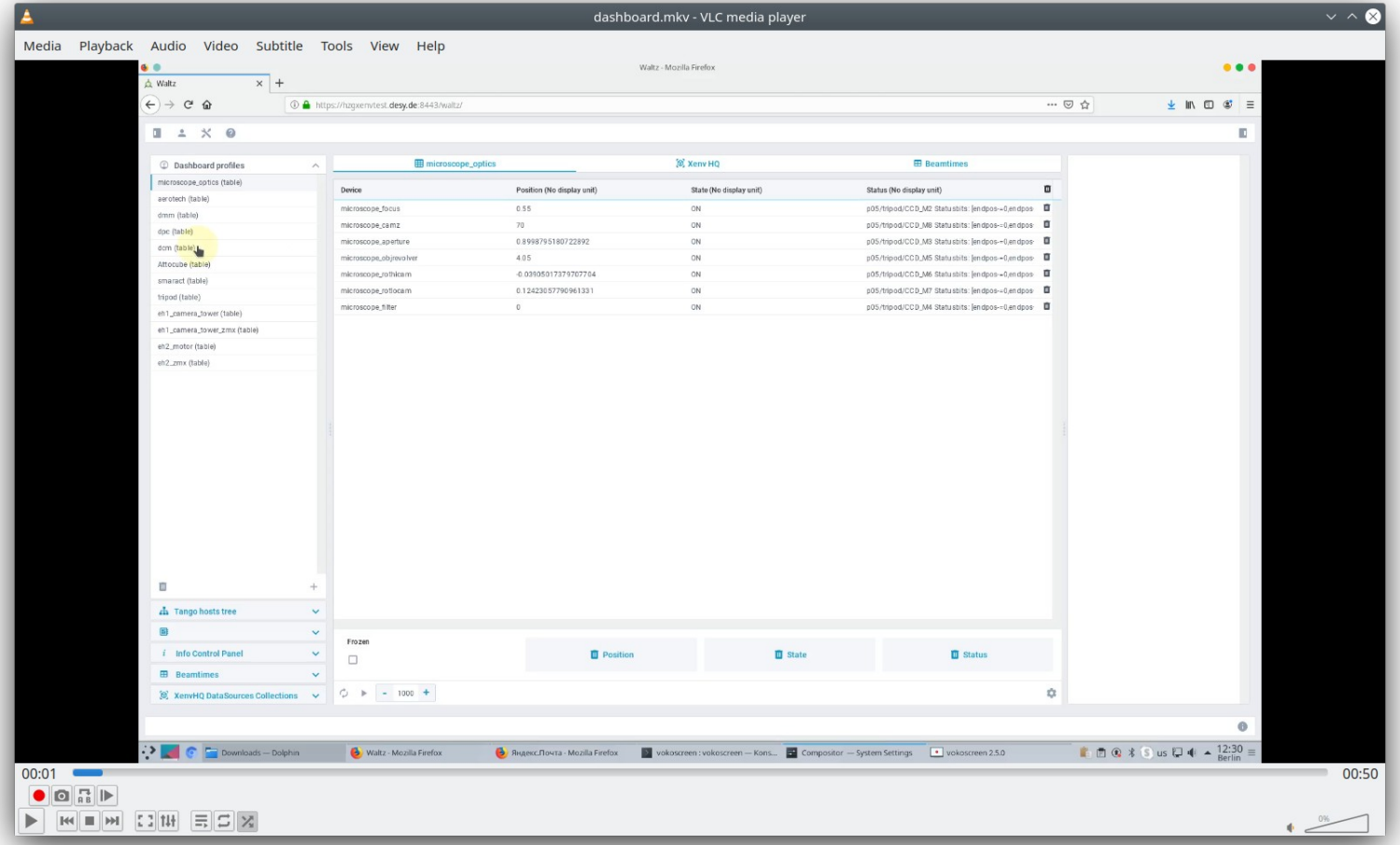
The roadmap was defined in 2015 (!) at the annual Tango Users Meeting In Krakow

Partially funded by Tango Collaboration contract in 2018

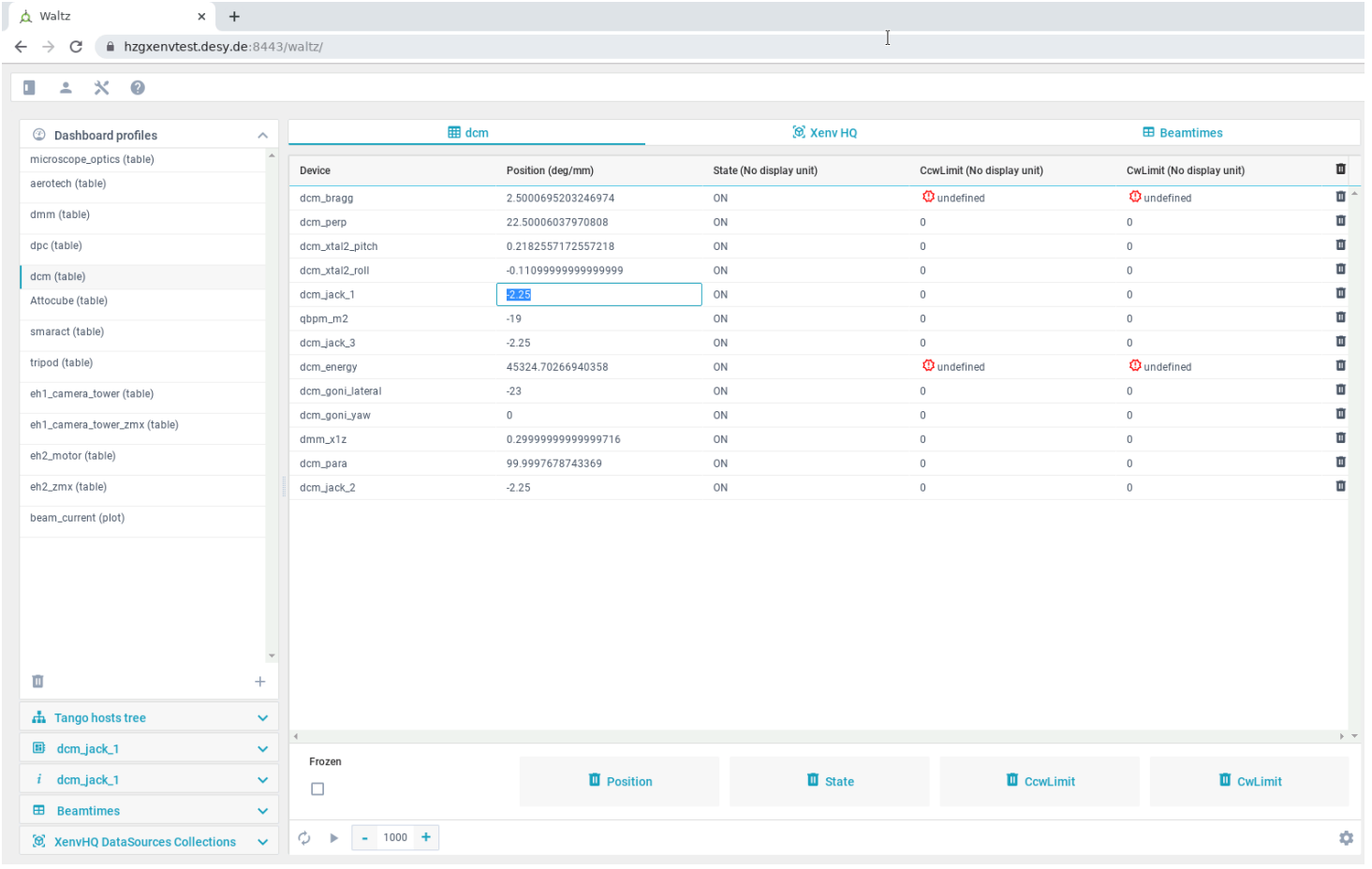
WALTZ CONTROLS

Renamed by the community vote @Tango Controls Users meeting in 2018

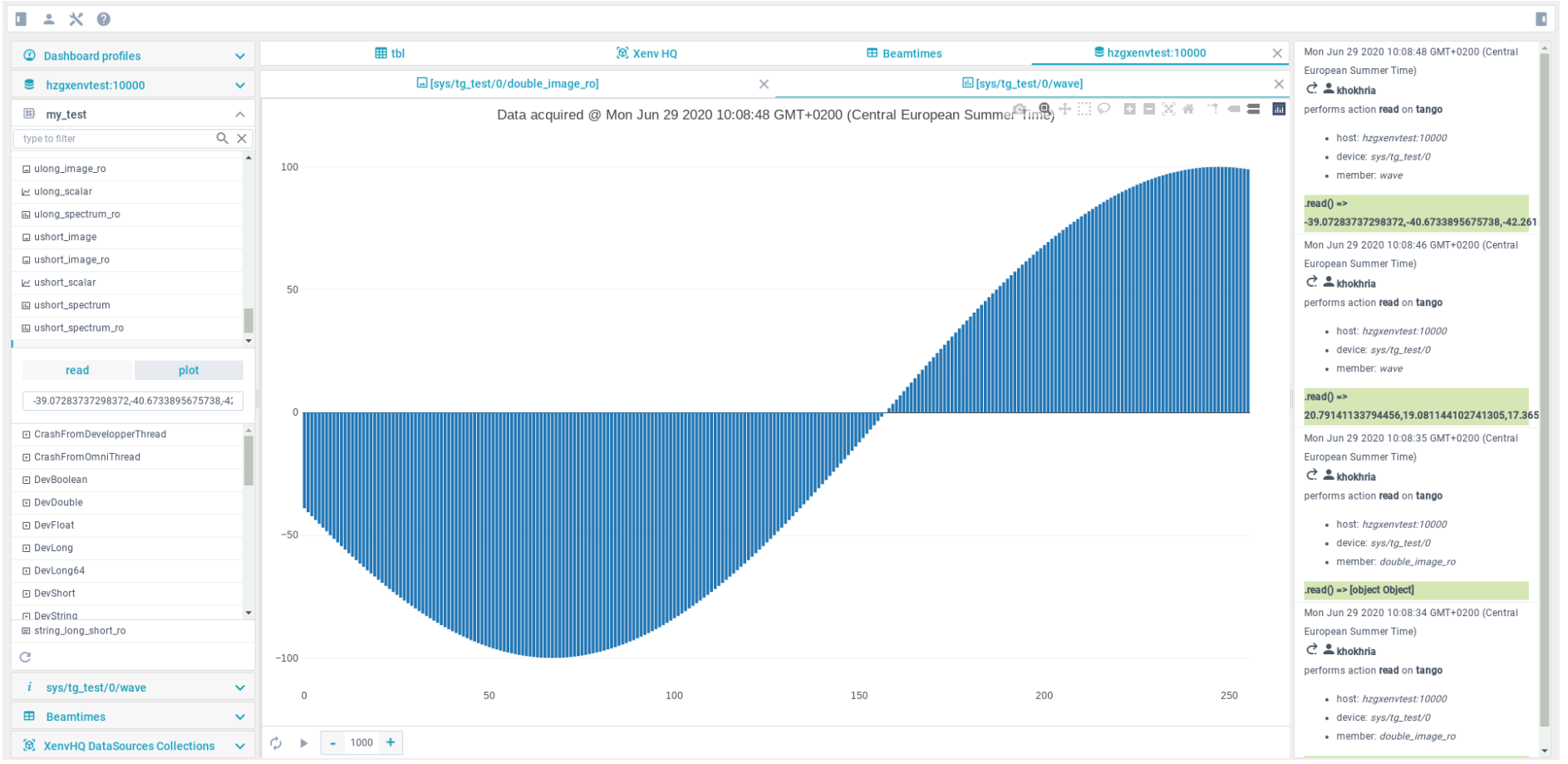
Waltz GUI video demo



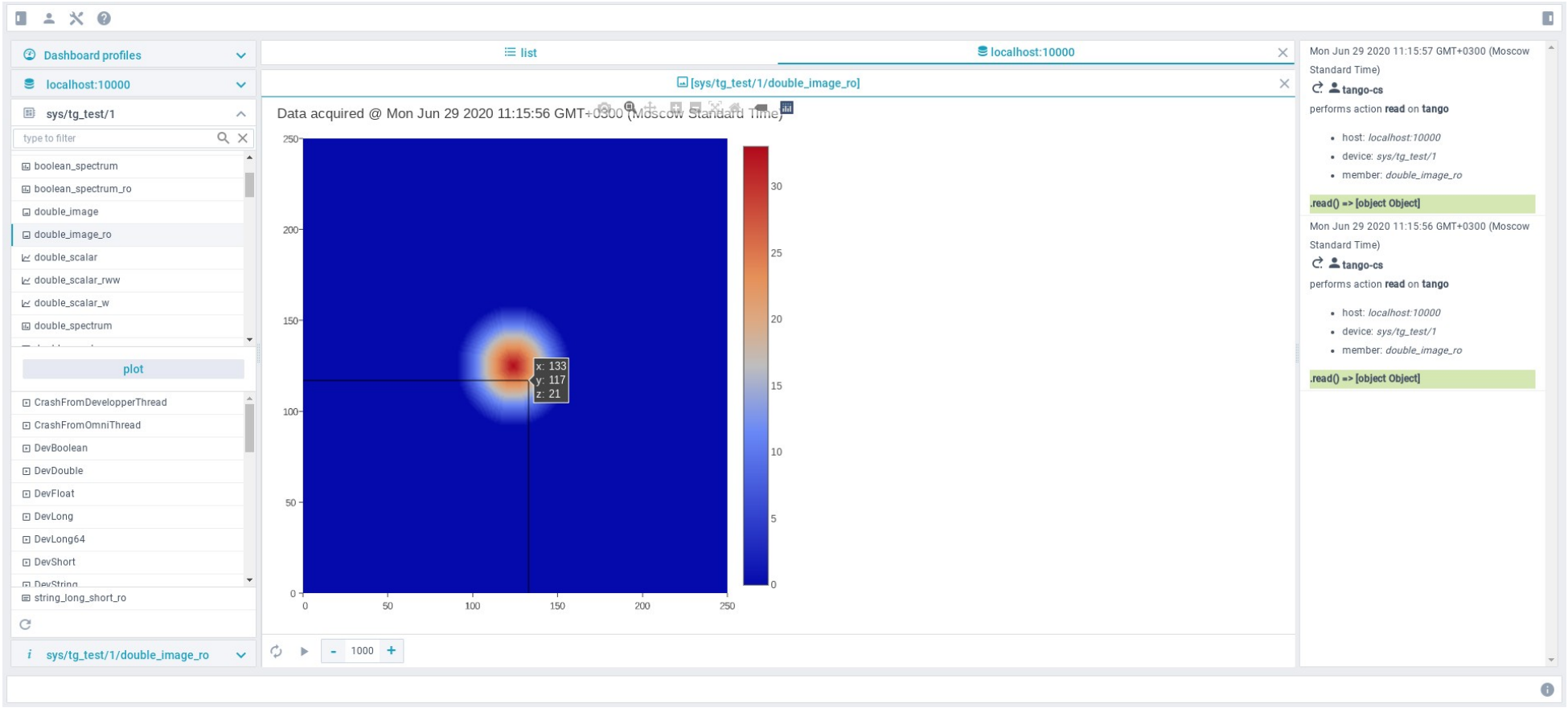
Waltz's GUI – several dashboards



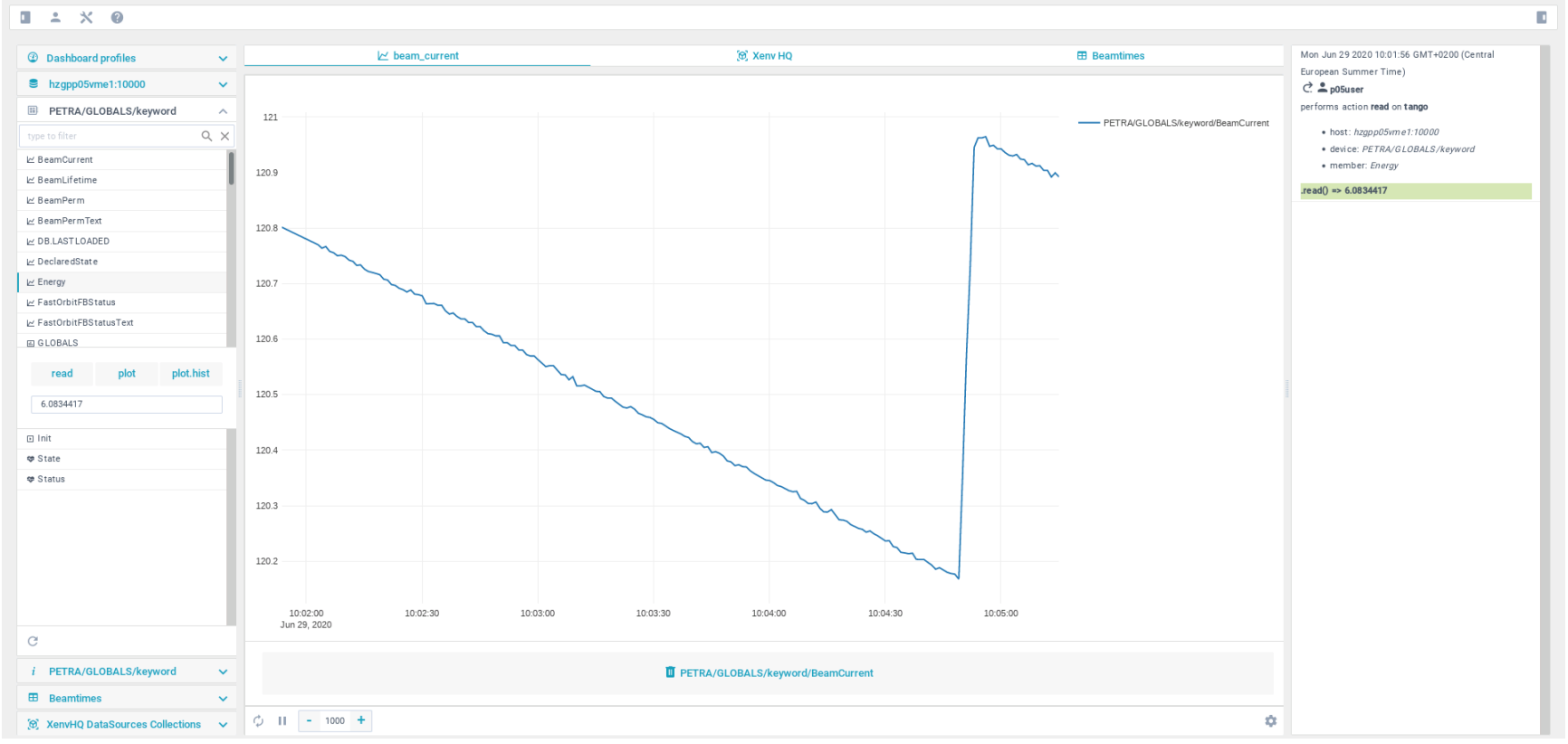
Waltz's GUI – plots, logs …



Waltz's GUI – images, information...



Waltz can gather info from 2 different SCADA systems – TINE and TANGO

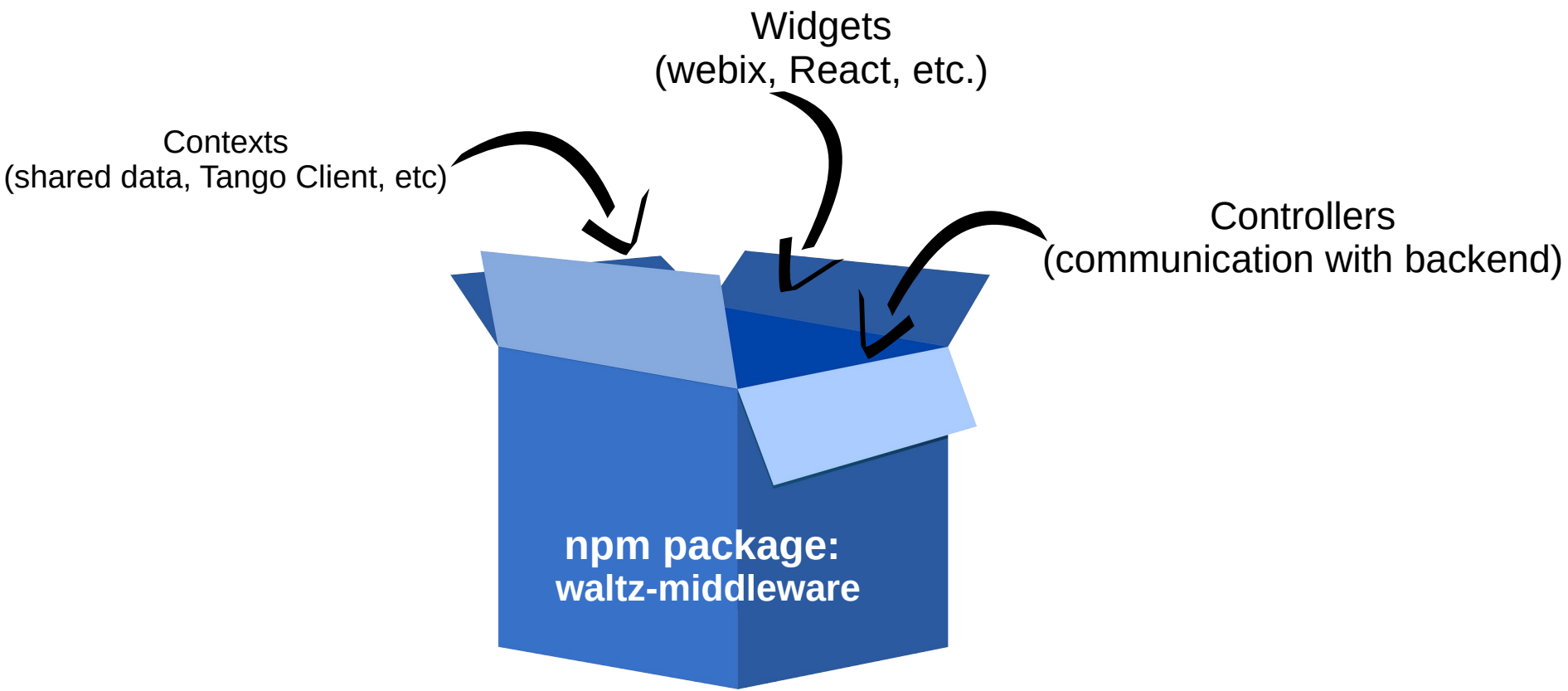


Waltz's tutorials

https://github.com/waltz-controls/waltz/wiki

- Developer Guide TangoWebapp
- Workshop@DESY User Guide Waltz
- Workshop@ESRF Waltz
- Workshop@ESRF_development
- Waltz@ESRF Programme

Waltz's architecture



Waltz has quite simple code API

You just list the widgets you want to have

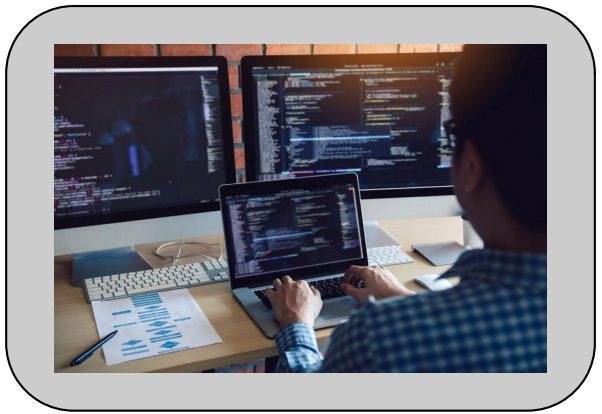
```
import {Application} from "@waltz-controls/middleware";
import {MainWindow} from "./layout.widgets";
import {Login} from "./login.widget";
import {interval, throwError,  timer} from "rxjs";
import {mergeMap, throttleTime} from "rxjs/operators";

const app = new Application({name:'waltz', version:'1.0.0'})
    .registerErrorHandler(err => {console.error(err)})
    .registerContext('tango-test', Promise.resolve("some context"))
    .registerObservable(1234, () => interval(100).pipe(throttleTime(1000)), 'numbers', "numbers")
    .registerWidget(app => new Login(app))
    .registerWidget(app => new MainWindow(app))
    .run();
```

Waltz under the hood

- Tech stack (development):
  - JavaScript 6
  - RxJS – reactive extension to JS (reactive streams)
  - *NPM** - dependency management
  - Webpack – for building
  - Waltz-Middleware
  - Webix library (for Waltz GUI, can be replaced with React or others)
- In production:
  - SPA – single page application
  - Static bundle – after build → a number a static files
  - *[Waltz GUI Requires Tango REST]*

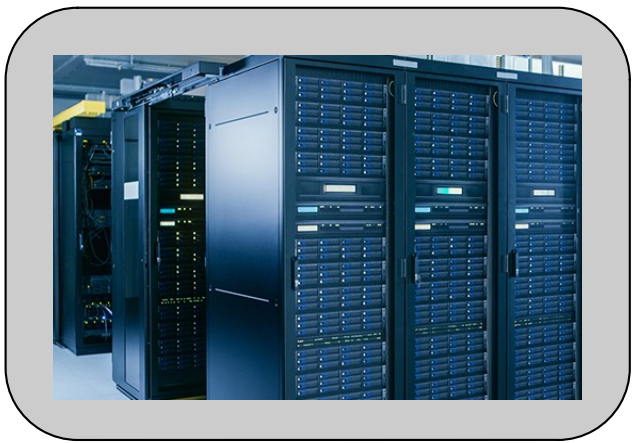
* GitHub NPM registry, requires authorization

Development → Production



- Setup GitHub packages authorization
  https://docs.github.com/en/packages/working-with-a-github-packages-registry/working-with-the-npm-registry
- Setup npm command line utility, i.e. Node development environment
- Use Waltz plugin template repo
  https://github.com/waltz-controls/waltz-webpack-plugin
- Execute: `npm install && npm run build`

- Production environment to serve static bundles, e.g.
  - *Apache Tomcat** OR
  - NodeJS OR
  - Nginx OR
  - etc.



* Used at Hereon for historical reasons

Waltz based/related projects

- Waltz GUI
  - XenvHQ – https://github.com/IK-Company/xenvhq-waltz-plugin
  - BeamtimeDB (no Tango) – https://github.com/IK-Company/beamtimedb-waltz-plugin
- React integration – https://github.com/waltz-controls/waltz/pull/257
- AXSIS GUI (no Tango) – https://github.com/Ingvord/axsis-xes-gui
- EPICS integration – https://github.com/waltz-controls/waltz/pull/210

Thank you!

Questions?

Web site: https://ik-company.com/

E-mail to: info@ik-company.com