

I. Khokhriakov*, F. Wilde, Helmholtz-Zentrum Geesthacht, O. Merkulova, IK Company, Kontakt: igor.khokhriakov@hzg.de

"Waltz is now used as a standard commissioning tool at P05 and thus replaced other tools like Astor, Jive and ATK panel."

- F. Wilde, Beamline scientist, P05, DESY, Hamburg, Germany

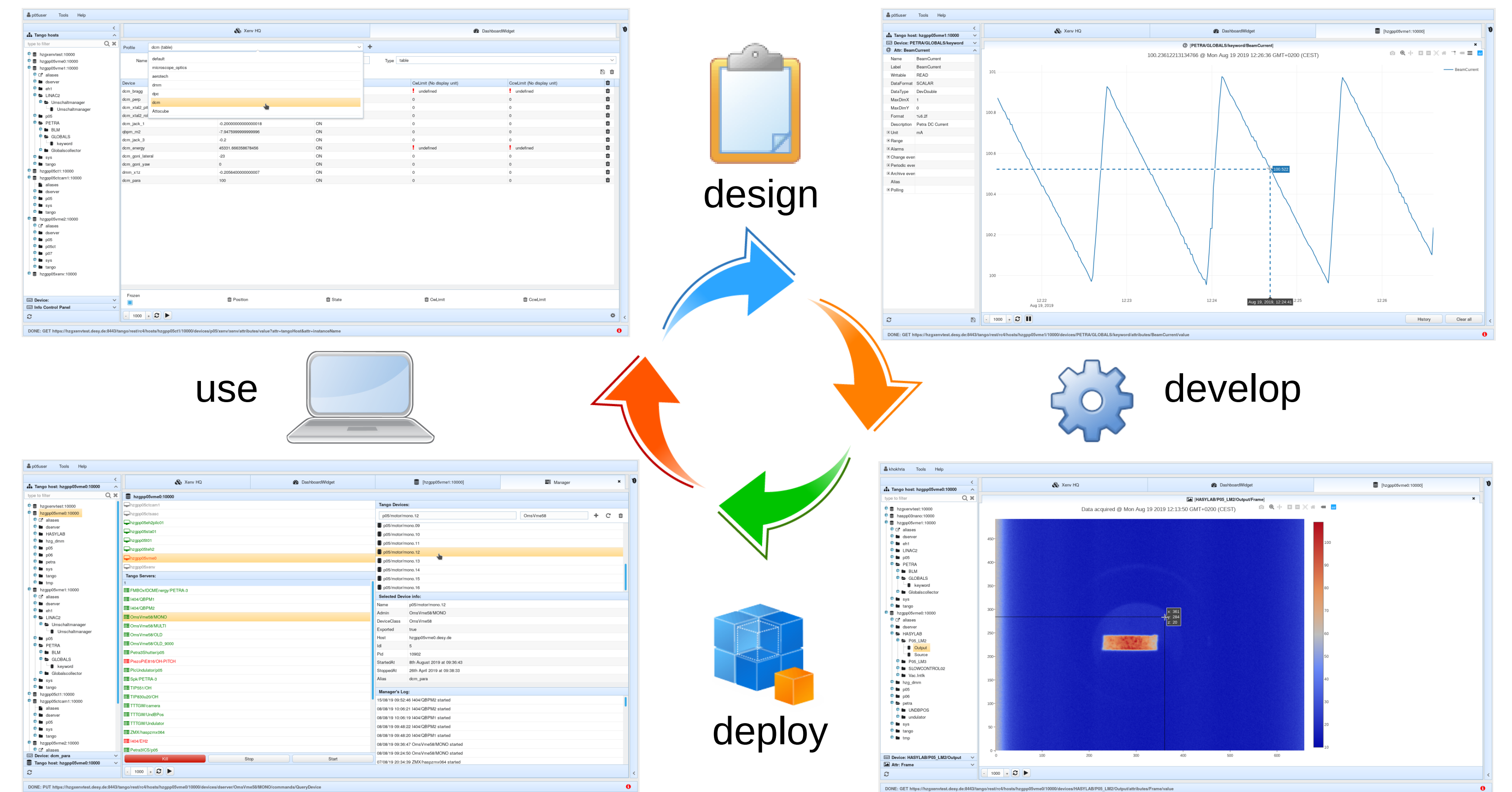
"Waltz offers a couple of very convenient features if you have to work with a distributed setup of tango device servers at the beamline. It is my preferred tool for early commissioning of new components."

- J. Hammel, Beamline scientist, P05, DESY, Hamburg, Germany

Introduction

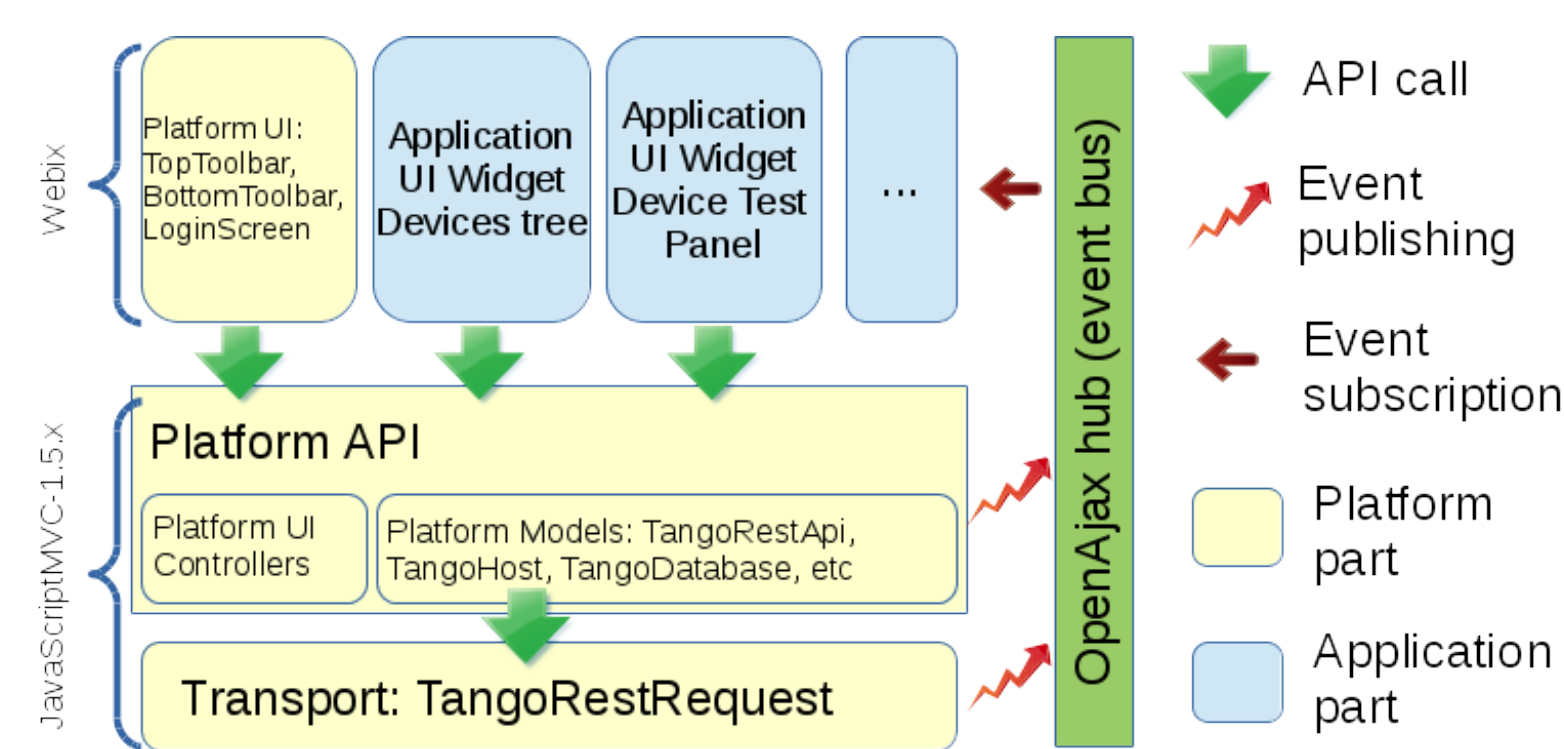
Waltz is a multi-purpose Tango Controls web application that provides the interface between the Tango Control system and the scientific users who define and calibrate their experiments. It can also be used for live monitoring of a big scientific installations like ESRF or DESY.

Waltz was designed to be used in two ways – as a platform and as an end-user application. In this poster we are describing Waltz as a platform for Tango Controls web applications and list some of the features from the end-user application to show possibilities of the platform.

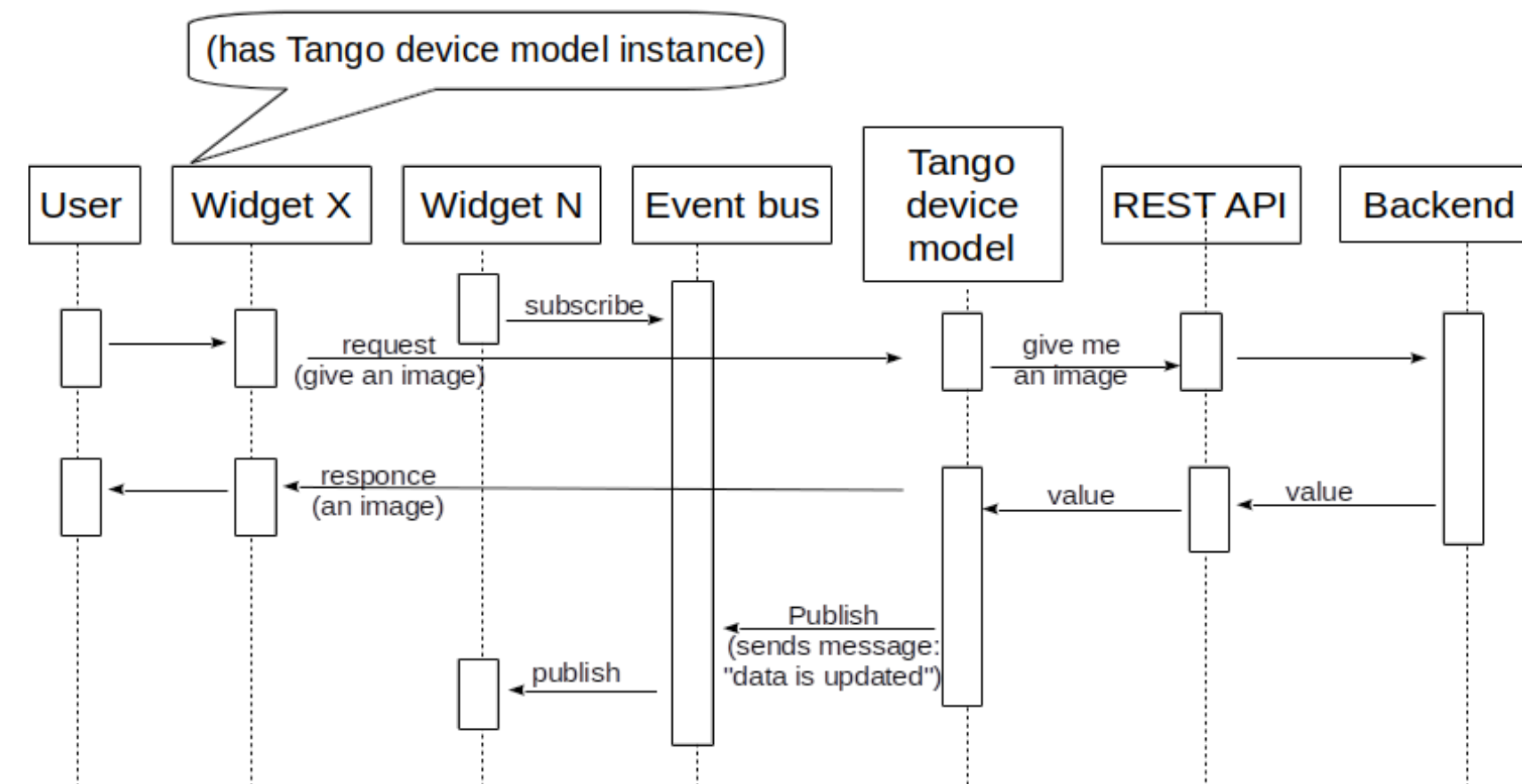


Waltz platform insights

1. Layered architecture



3. User and widgets interaction sequence diagram



Waltz considered as a platform that can be used to implement integrated and coherent web-based GUIs for Tango Controls. Flexibility of Waltz is achieved by the layered architecture (Fig. 1); internal event bus (Fig. 2, 3, 4) and widgets (e.g., UI components) which can be used as building blocks for rich functionality (Fig. 2, 4). Waltz offers a number of convenient APIs: Tango device model API; reusable functional components (e.g., mixins); UI builder API. All above allows us to apply Agile development approach (Fig. 5).

API part – conceived for:

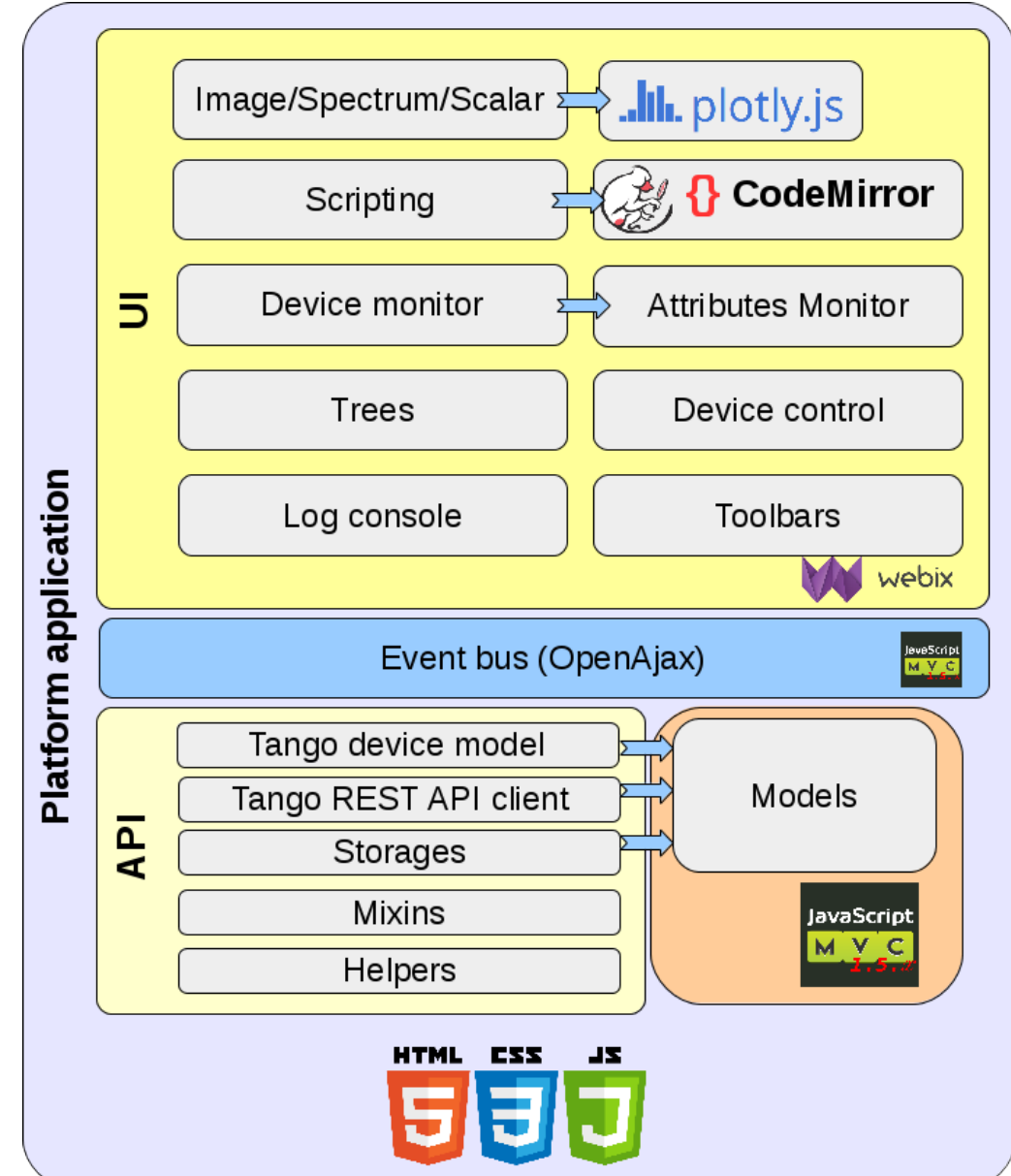
- building UI;
- errors handling;
- interacting with Tango device model;
- reusing functional components.

UI part – implemented following the concept of smart components. Each UI component is completely standalone and can be used as a building block for more complex widgets or even dedicated applications.

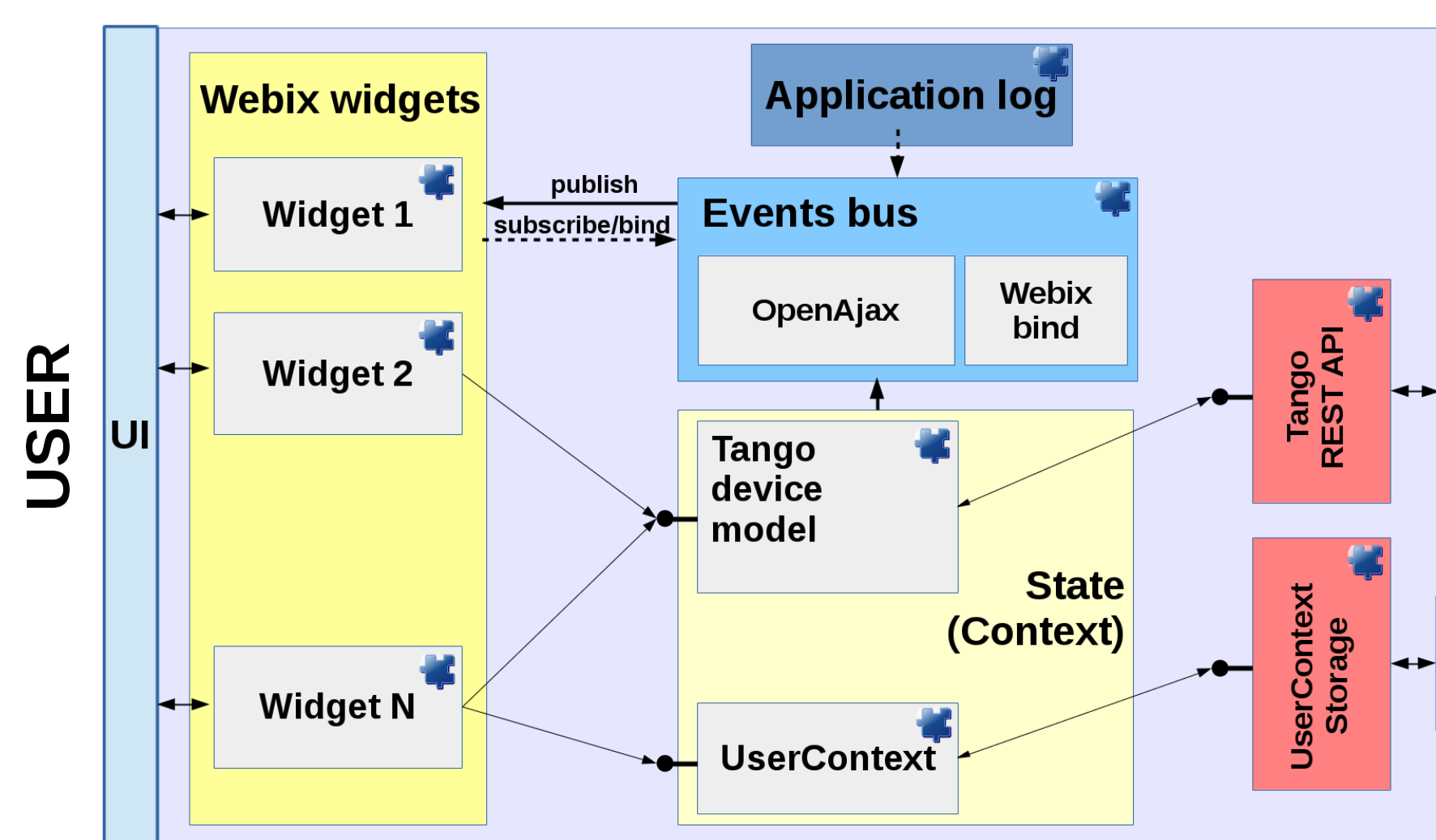
Find out more:



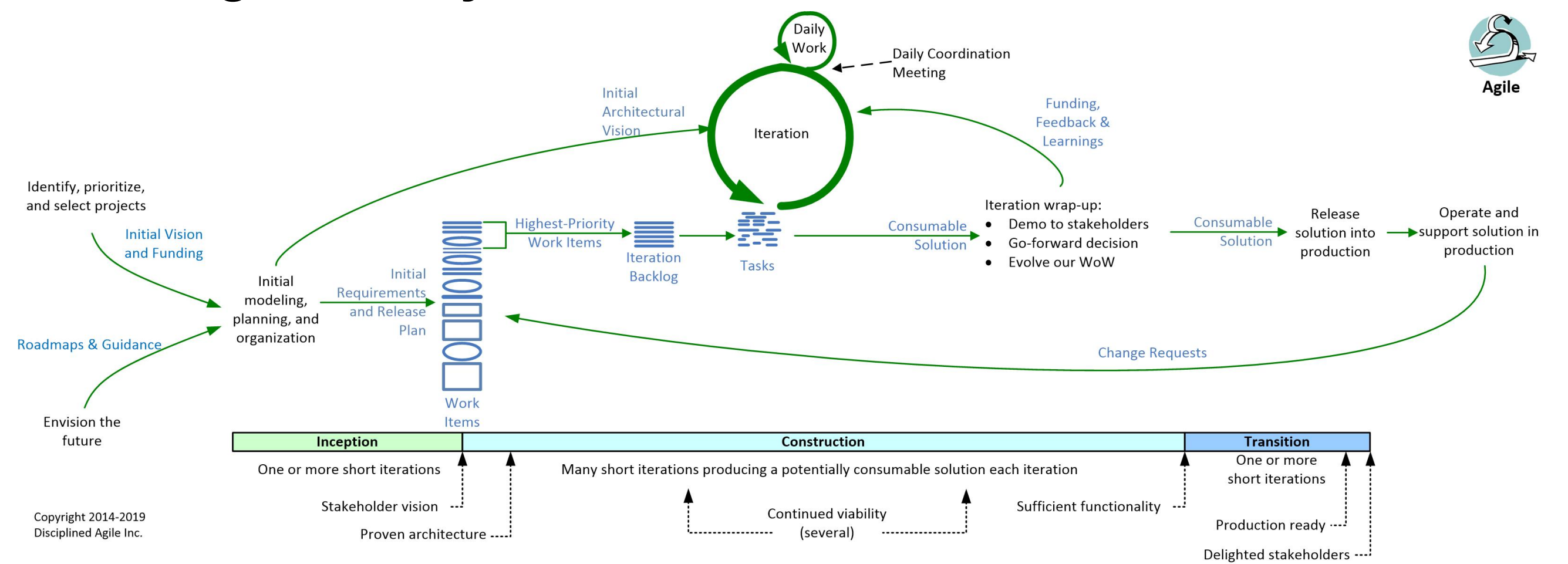
2. Modules and dependencies



4. Components and Connectors diagram



5. Agile life-cycle



Development platform

What has been done on the platform

- Tango hosts tree;
- Tango device control panel;
- Information/properties panels;
- Tango device configuration/monitoring;
- Tango hosts manager;
- Terminal;
- User settings with;
- All kinds of filtering.

Waltz platform APIs

UI Builder API – application's UI layout, defines which widgets and content will be available in which panels/views;

PlatformContext – single entry point to application's state, has references to application data;

UserContext – storage of user specific data from widgets;

Mixins – enrich JS object's functionality by using common functions (e.g. Stateful allows widget to persist its state, OpenAjaxListener allows widget to subscribe to event bus);

Tango Device Model API – a high level client API familiar to Tango developers, an abstraction of Tango device model on top of Tango REST API;

Widget API – easy way to create new widgets or reuse existing;

Subscriptions – allow Waltz to receive notifications about native Tango events with minimal overhead.

Easy deployment

Waltz is distributed as a standard Java Web Application Archive. So it can be deployed at any Java EE compliant application server. That allows to have a **standard enterprise infrastructure** which scales very well. For instance, it is possible to deploy:

- Waltz behind a **load balancer**;
- **multiple Waltz's instances** simultaneously;
- **beamline specific use-cases** from Waltz branches.

Conclusions

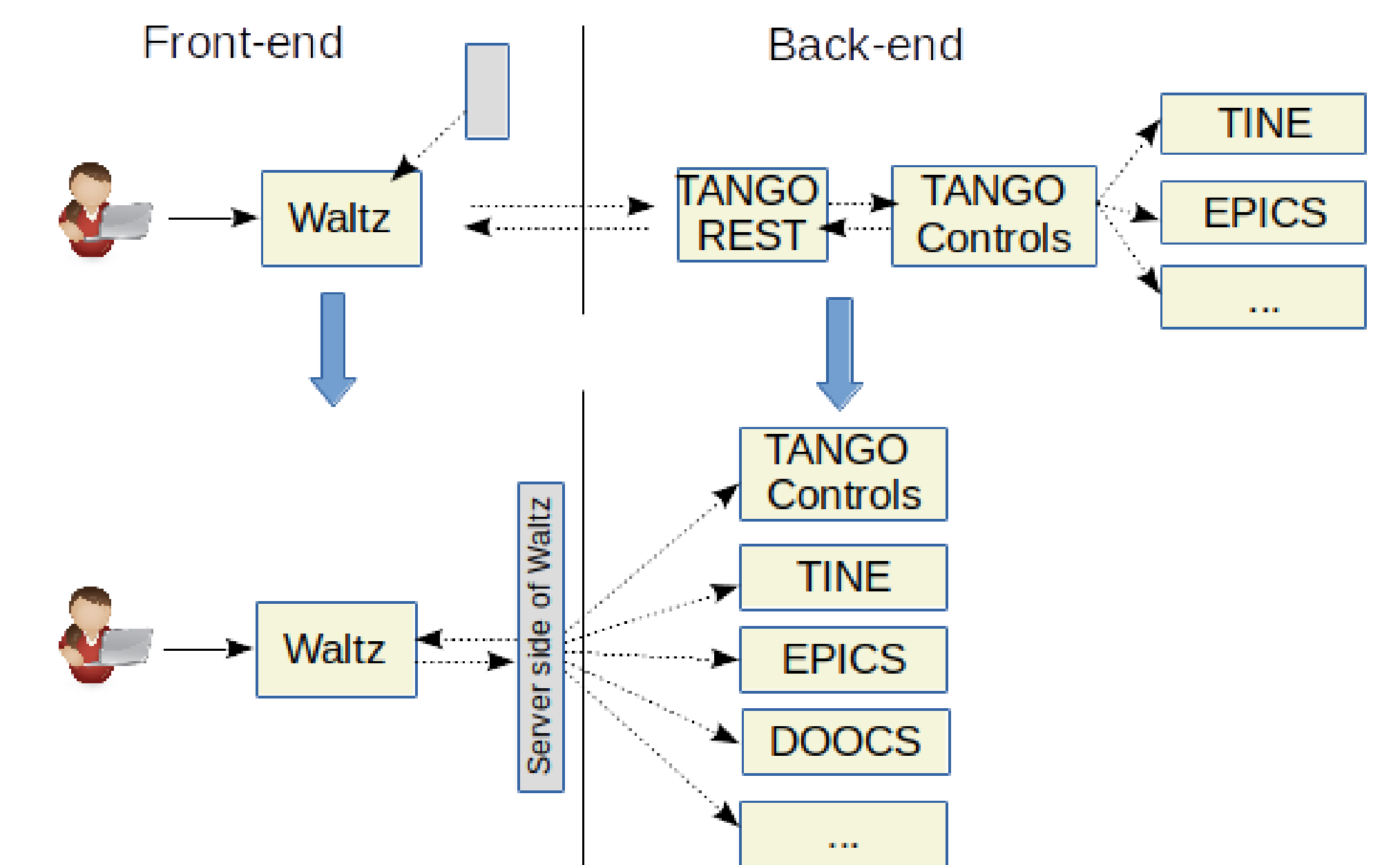
Waltz is a platform for Tango Controls web/mobile applications development. It implements features of enterprise web applications that greatly improves user experience with SCADA systems.

Among these features are: integrated UI; storing per-user basis data; deploying multiple instances of the application with specific for each use case widgets; multiple views based on user role, etc.

Waltz has proven itself to be very useful for beamlines commissioning at DESY for several month.

It is planned to evolve Waltz and make it more powerful in terms of end-user application (visual scripting editors, SVGs) and in terms of developer's platform. We foresee further developments to make integration with other control systems and to have a direct communication with hardware.

Future plans



- Enrich server side:
 - integration with **other control systems** (EPICS, TINE, etc);
 - **direct access** to hardware;
 - creation of **communication protocol** based on reactive streams;
- Split back-end into **micro-services**;
- Have **unified communication protocol**;
- Introduce **different user roles**;
- Extend **scripting** capabilities;
- Create **mobile friendly interface**.