

# TangoJS – a web-based interface for TANGO Control System

Michał Liszcz<sup>1</sup>, Włodzimierz Funika<sup>1,2</sup>, Łukasz Żytniak<sup>3</sup>

<sup>1</sup> AGH, Faculty of Computer Science, Electronics and Telecommunication, Dept. of Computer Science, al. Mickiewicza 30, 30-059, Kraków, Poland

<sup>2</sup> AGH, ACC Cyfronet AGH, ul. Nawojki 11, 30-950, Kraków, Poland

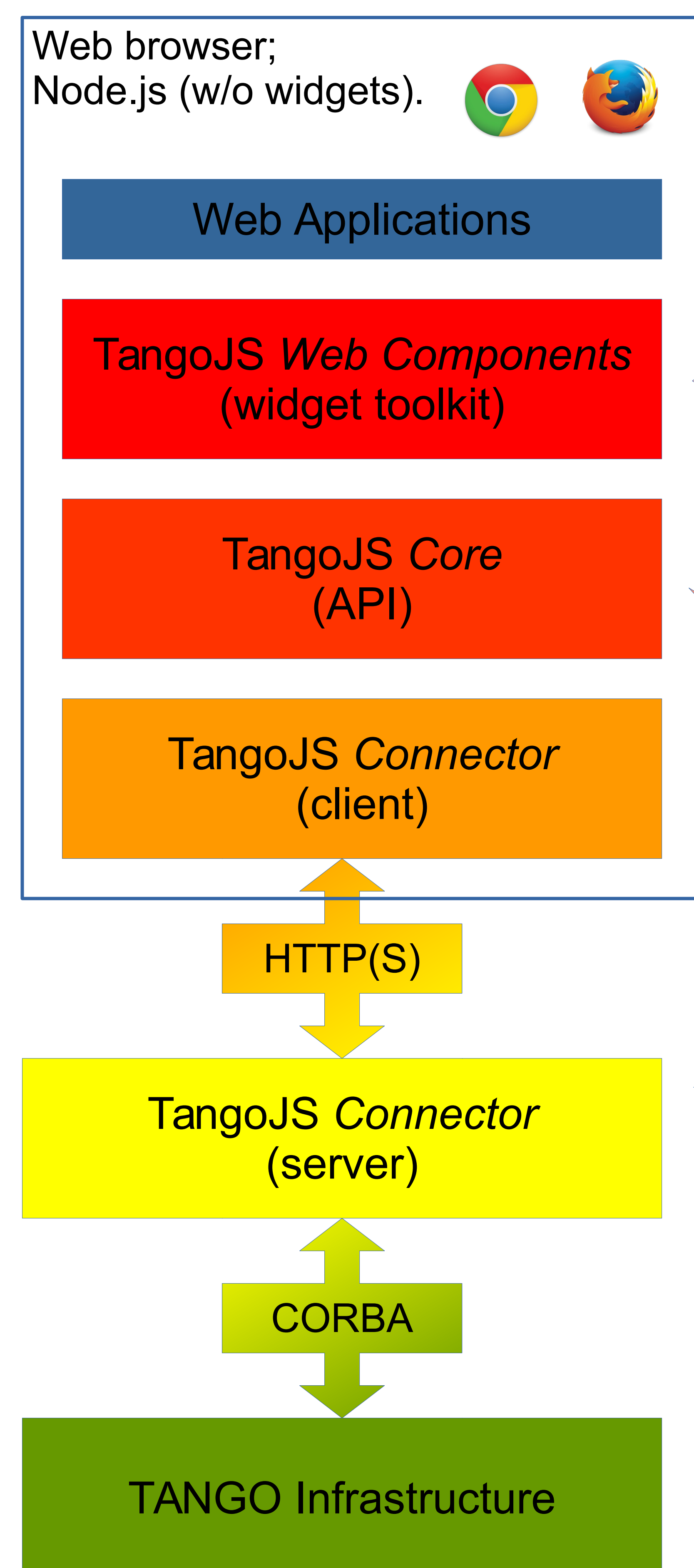
<sup>3</sup> National Synchrotron Radiation Centre Solaris, ul. Czerwone Maki 98, 30-392, Kraków, Poland

## Introduction

Control of the expensive and sensitive hardware components in large installations like scientific facilities may be a challenging task. In order to conduct an experiment, multiple elements like motors, ion pumps, valves and power-supplies have to be orchestrated. To address this problem, the TANGO Control System [1] has been developed at ESRF synchrotron-radiation facility. This poster presents TangoJS – a modular, standard-based library for building TANGO clients for web browsers.

## TangoJS Stack

TangoJS uses layered architecture, built on top of existing TANGO infrastructure.



## Motivation

Recently, web-based approach has become crucial in building accessible and adaptive GUI applications. Unfortunately, TANGO-based applications cannot run in web browsers. Here TangoJS comes into play, allowing for rapid development of TANGO clients, integrating neatly with modern frameworks and using standard frontend development tools.



## Widget Toolkit

- Collection of UI widgets;
- Built using the standard Web Components API [3];
- All components are derived from `HTMLElement`;
- May be connected together to build a synoptic panel;
- No framework required;
- Almost no dependencies required (just plot library);
- Inspired by the Taurus [2] library (for Python).

**label**  
Visualizes name, value and status of selected attribute.

**line-edit**  
Writable variant of label. May be a text-box, a spinner or a toggle.

**command-button**  
When pressed, invokes a command on a device.

**state-led**  
Visualizes state of a device, e.g. on/off, alert, error.

**trend**

**form**

**device-tree**

sine_trend	0.99116...	u	
attribute name (obtained from the database)	current value (device is polled at constant rate)	unit	quality
scalar	0	u	

writable field  
(depends on attribute type)

state: ON state: OFF state: FAULT state: ALARM

A set of buttons – each invokes specified command and calls user-defined callback when result is obtained.

tangojs/test/dev1	OFF	
full device name (obtained from the database)	textual state representation (status)	state visualized with multi-colored

Plots values of multiple attributes over time.

Groups attributes and detects best widgets.

Displays devices hierarchy using tree-view.

- Set of interfaces, enums, structs and constants;
- Partly generated from the TANGO IDL;
- Interfaces similar to the TANGO Java API;
- Passes all calls to the underlying connector;
- Runs in all modern browsers and Node.js;
- Plain Javascript – no dependencies required.

- Proxy that connects TangoJS and TANGO;
- Separated in server-side and client-side parts;
- Multiple configuration options possible: HTTP, WebSocket, in-memory mock, etc.;
- Server-side: possible integration with existing TANGO-REST servers, e.g. mTango [4];
- Client-side: consuming server's API.

- Hardware;
- Database;
- Device Servers.

Fig. 1. Storage ring magnet in National Synchrotron Radiation Centre Solaris.



## TangoJS Assets

Creating TangoJS applications is dead simple and requires only basic knowledge of web-development and Node.js.

- All parts of the TangoJS stack are available in npm Registry. Just drop desired dependencies in your package.json file;
- Each widget is a self-contained HTML file, which may be included in your page (using standard HTML Imports feature);
- Components may be created using declarative syntax (HTML) or programmatically (with DOM manipulation APIs).

```
<tangojs-line-edit
  model="tangojs/test/dev1/scalar"
  poll-period="1000"
  show-name
  show-unit
  show-quality>
</tangojs-line-edit>
```

Sample project and more details are available at <https://tangojs.github.io>.

## References

1. TANGO Controls website: <http://www.tango-controls.org/>,
2. Taurus project website: <http://www.taurus-scada.org>,
3. Web Components technical standard: <https://www.w3.org/standards/techs/components>,
4. mTango project website: <https://bitbucket.org/hzgwpn/mtango/wiki/Home>

## Acknowledgments

This research is partly supported by AGH grant no. 11.11.230.124.

The project is developed in collaboration with National Synchrotron Centre Solaris.