HSLU Hochschule Luzern

Hallo ROS!

ROSCON DE 2024

HSLU
Technik & Architektur
11. Dezember 2024

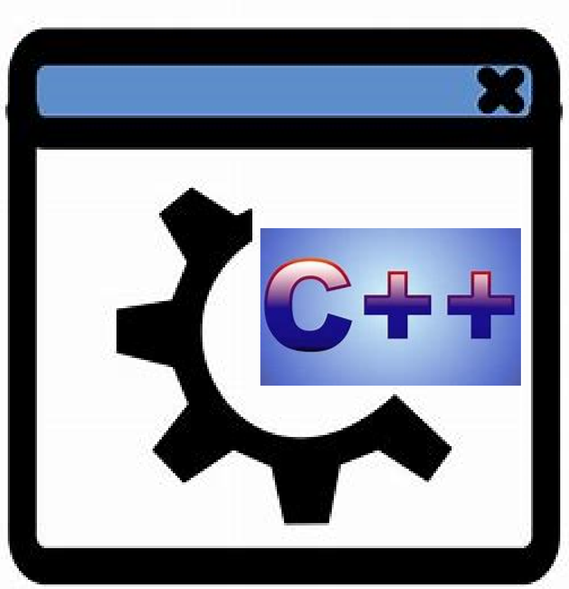
FH Zentralschweiz

Programm

- Problemstellung: Kommunikation zwischen ROS und ROS-externen Systemen / Applikationen
- Anwendungsfall 1: Web User Interface
  - Web UI: Konfiguration, http Server and ROS bridge websocket
  - Web UI für ROS2, Unterschiede zum Web UI für ROS1
- Anwendungsfall 2: Python: Konfiguration Websocket Client (ROS1 und ROS2)
- Anwendungsfall 3: Verwendung von eprosimas fastDDS zur Kommunikation zwischen ROS und einer C++ Applikation (nur für ROS2)
- Fazit ROS Schnittstellen

Hallo ROS!

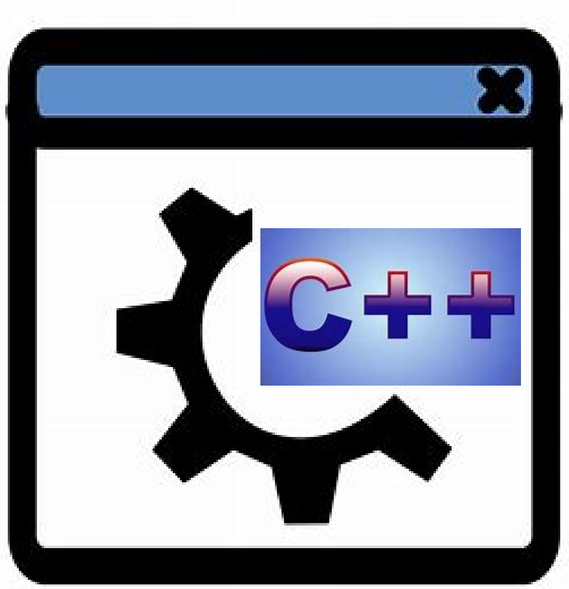
Quelle



Quelle Quelle2

Hallo ROS!

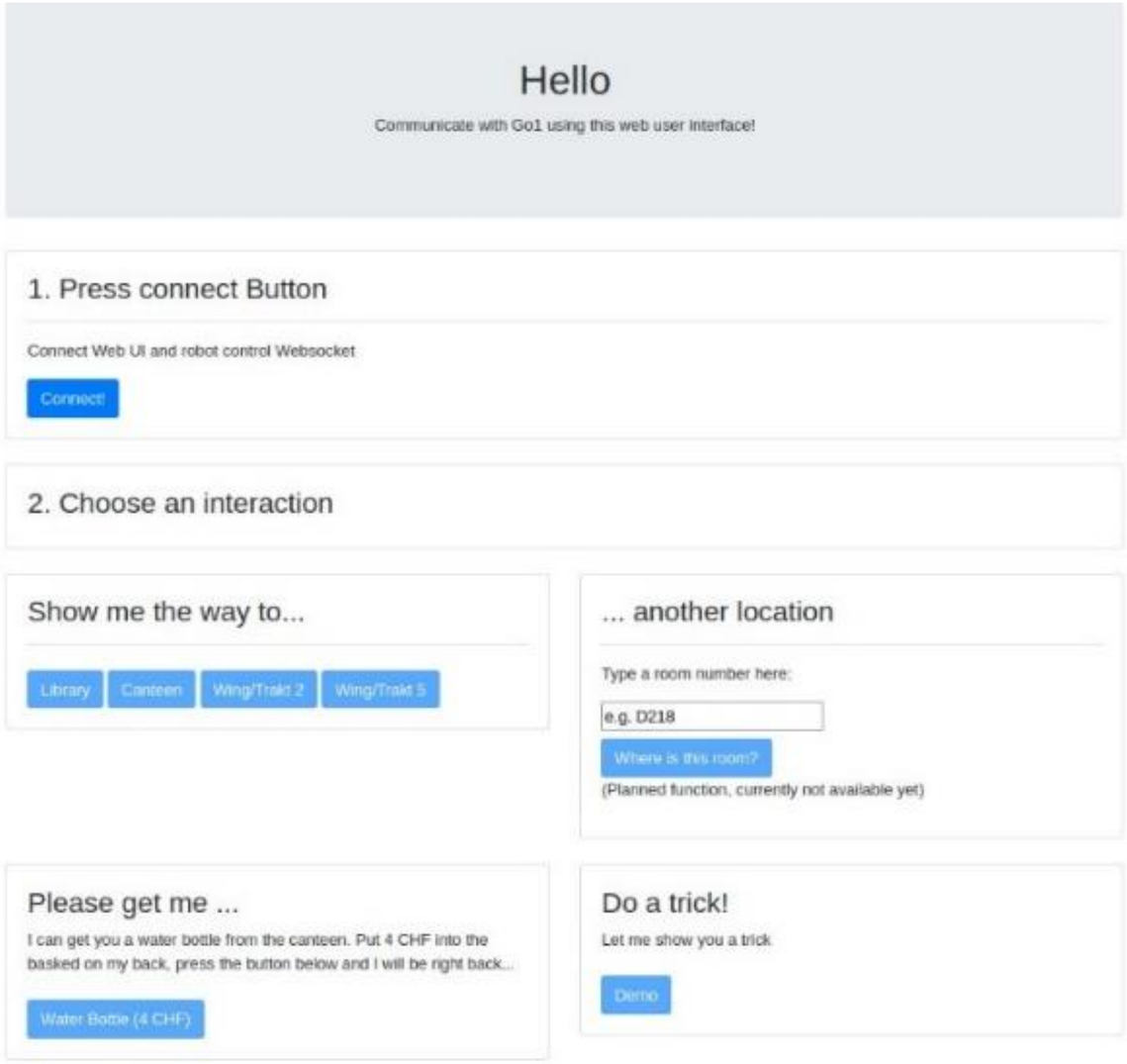
Quelle

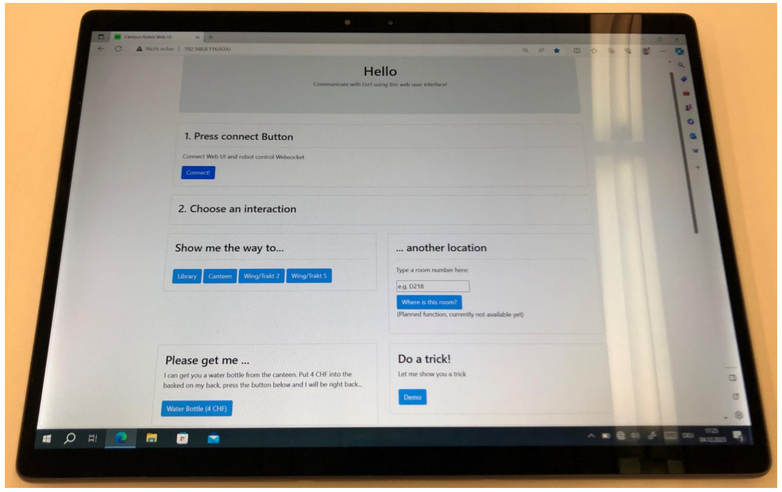


Quelle Quelle2

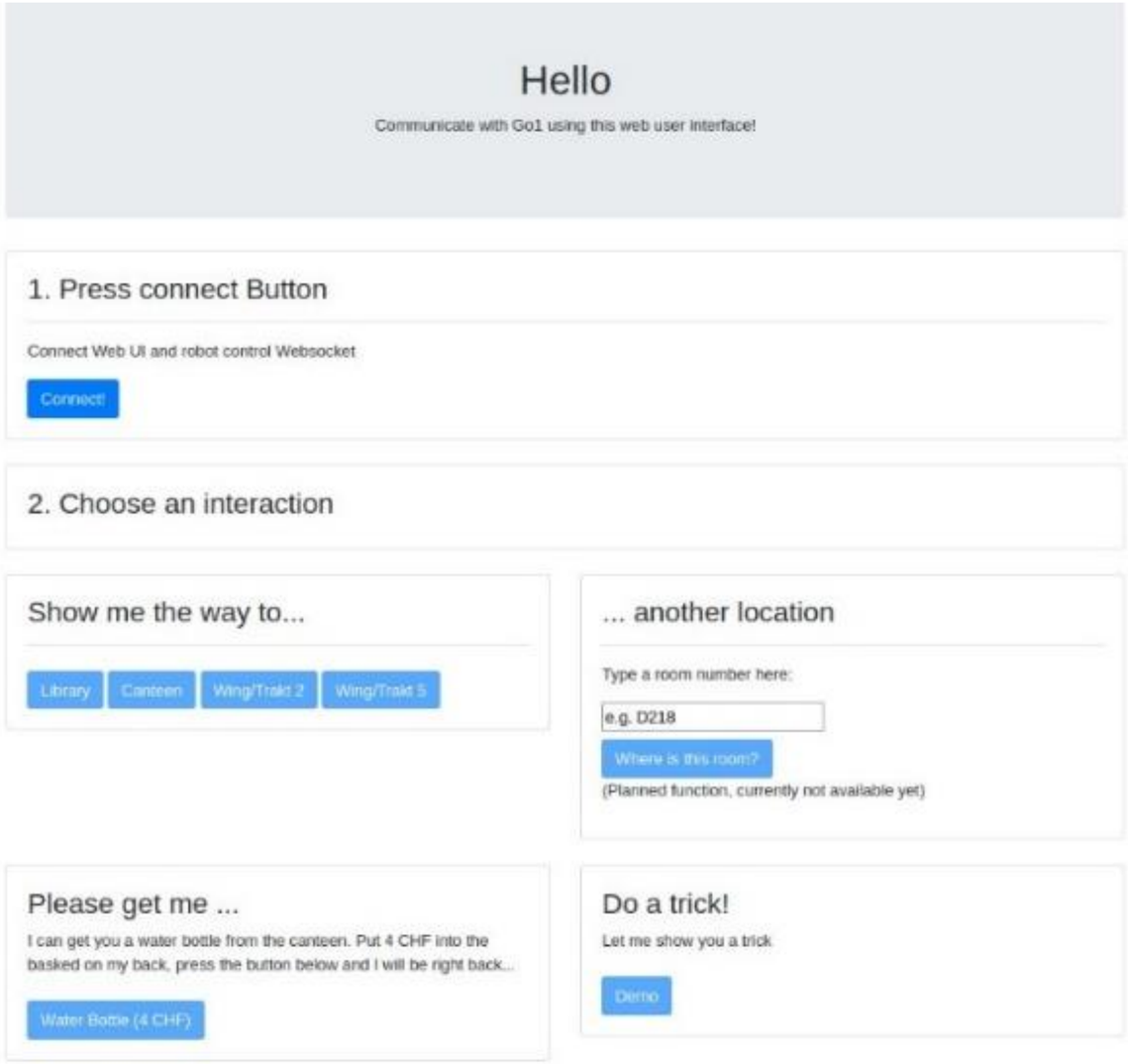
https://gitlab.switch.ch/hslu-horw-campus-robot/home

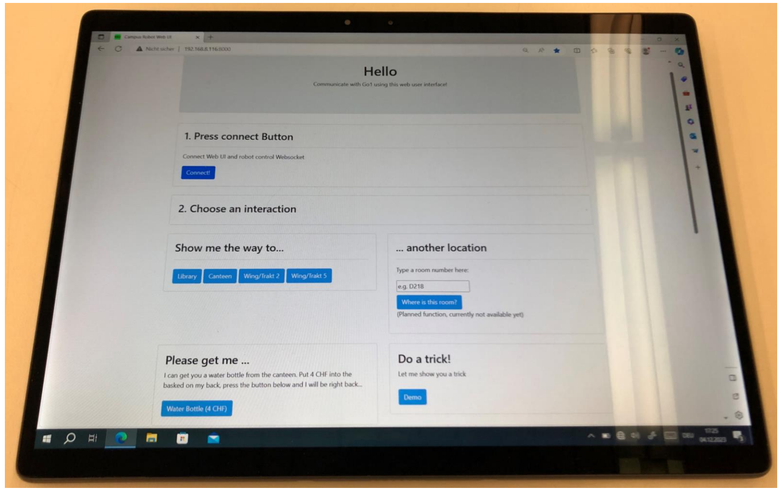
ROS Schnittstelle zu Web User Interface



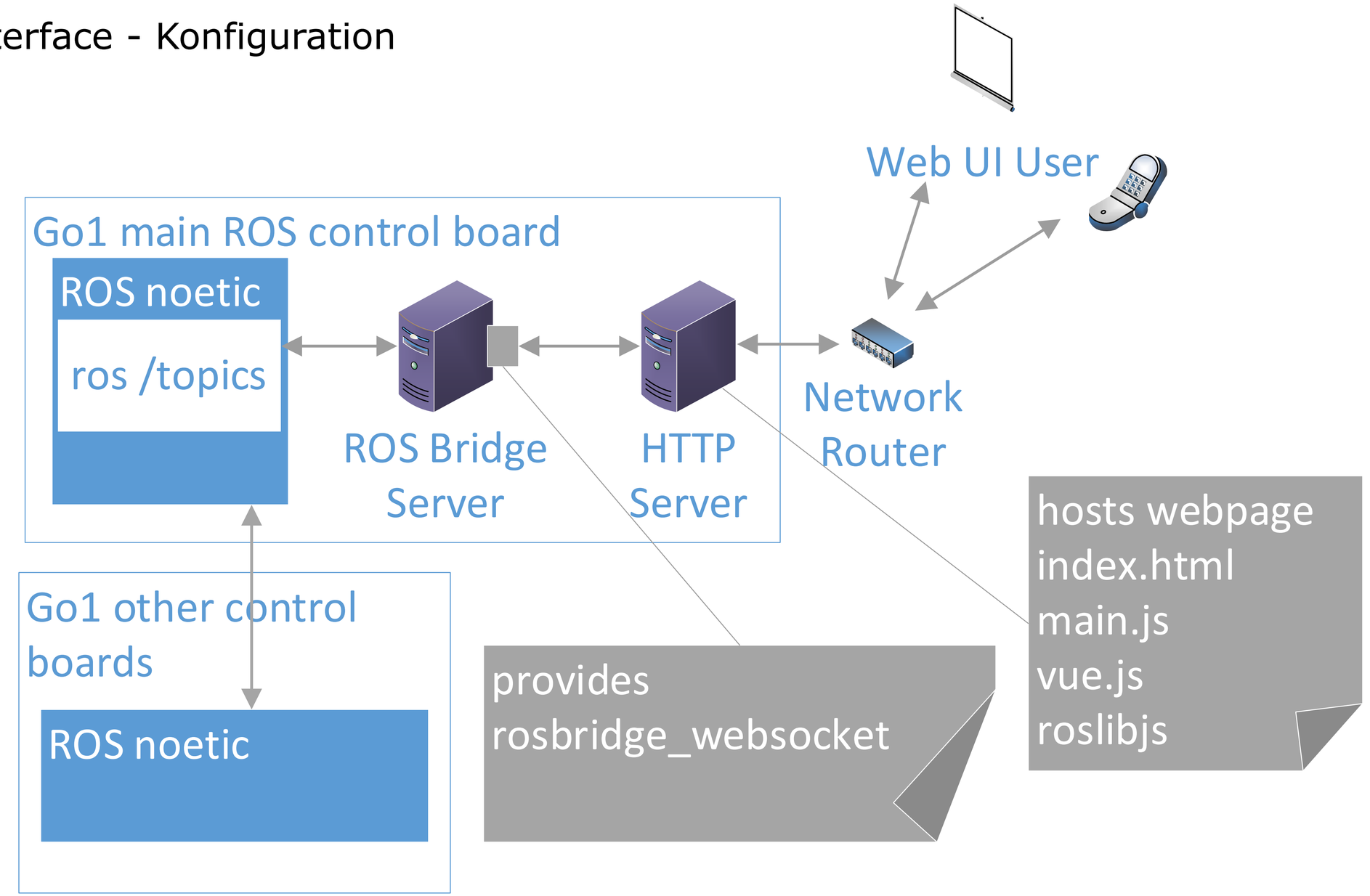


ROS Schnittstelle zu Web User Interface

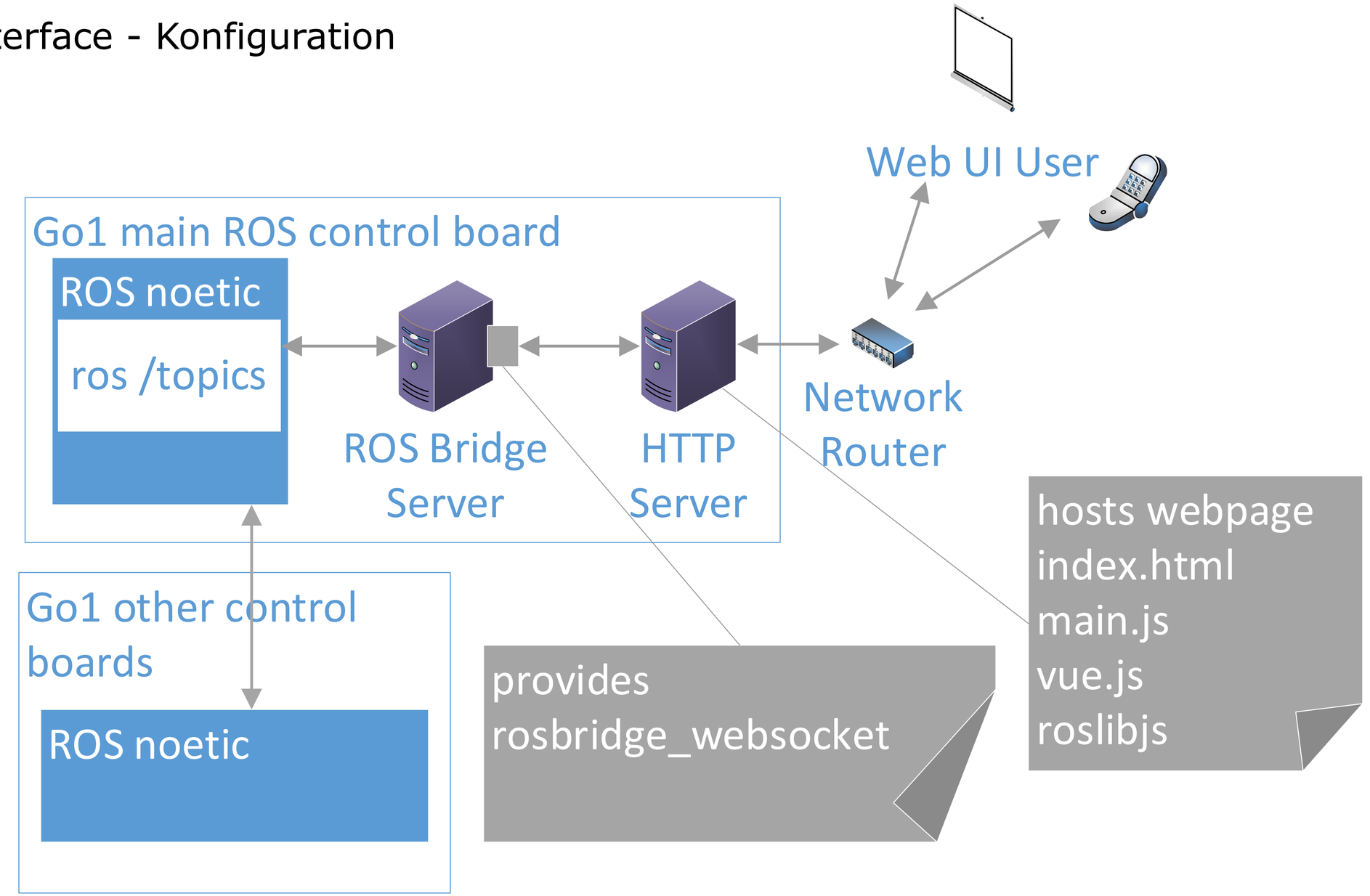




Web User Interface - Konfiguration



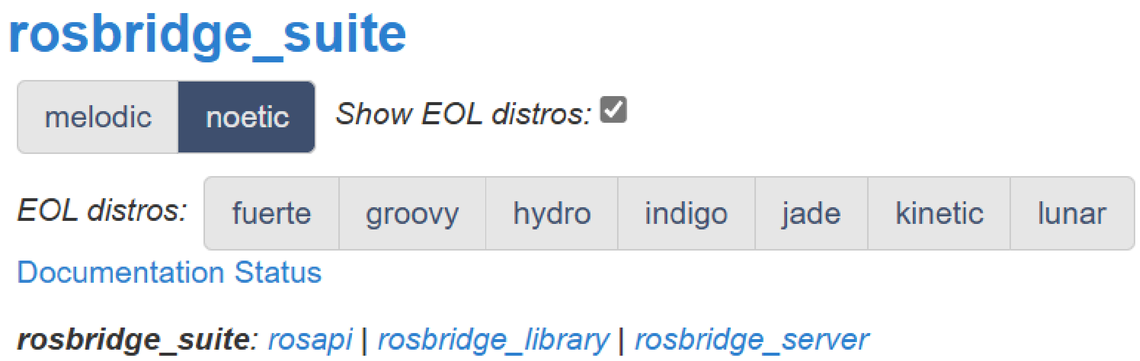
Web User Interface - Konfiguration



Web User Interface – ROS1 vs. ROS2

ROS1

- unser Zielsystem
- rosbridge_suite für ROS1



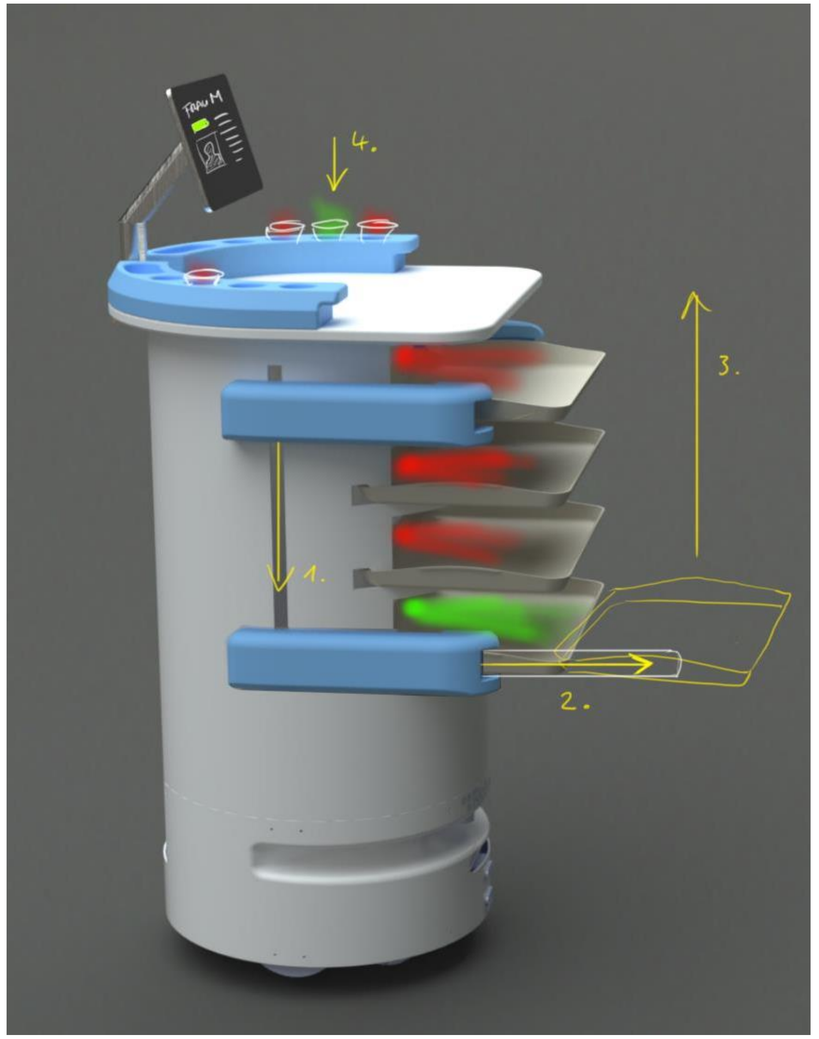
ROS2

- für ROS2 auch getestet
- rosbridge_suite für ROS2

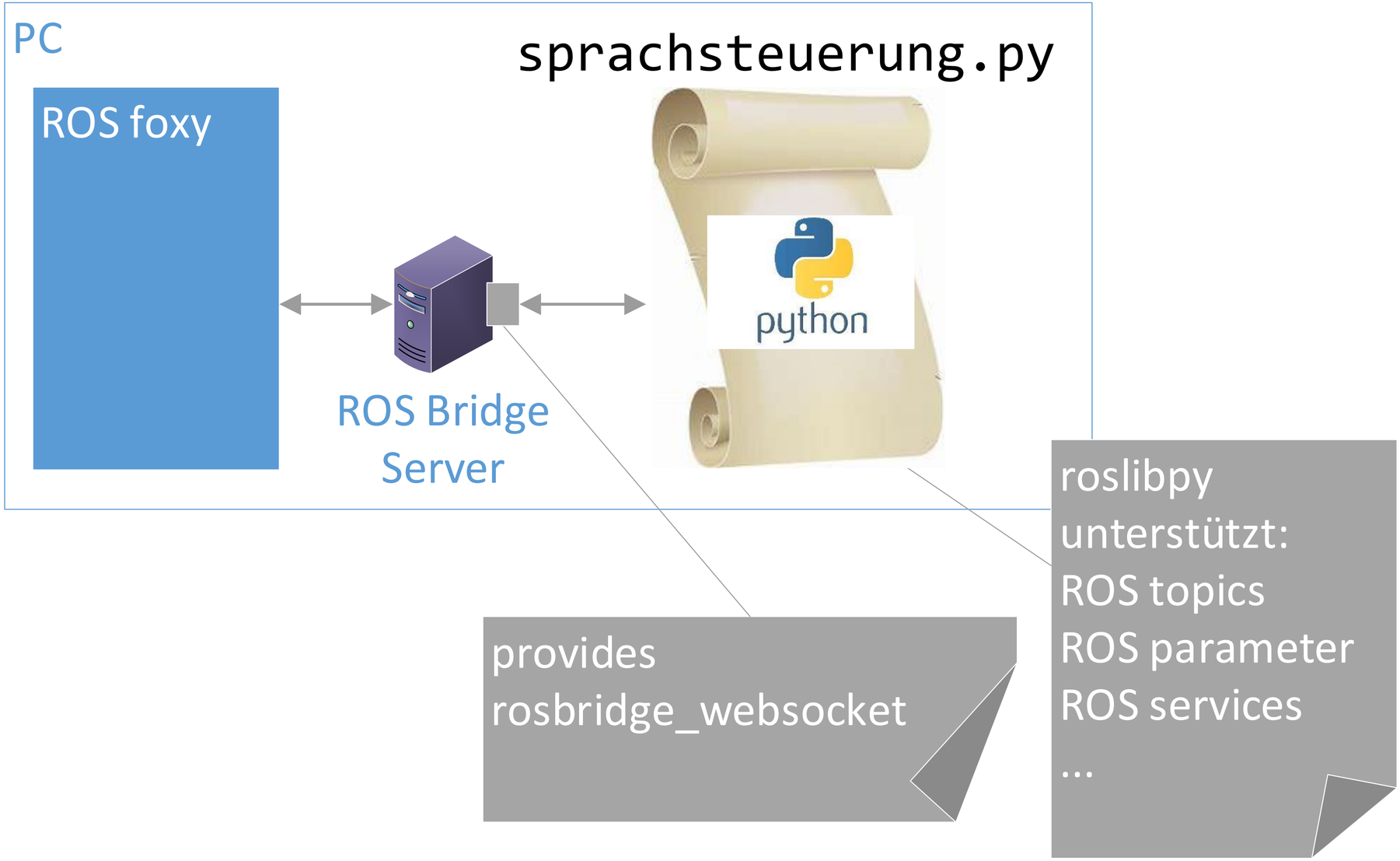
- Implementierung Webseite und http Server unabhängig von der ROS Version

ROS Schnittstelle mit Python Script

`sprachsteuerung.py`



Python Script - Konfiguration



Python Script – ROS1 vs. ROS2

ROS1

- rosbridge_suite für ROS1 verfügbar
- rosbridge_suite mit ROS1 noetic mit dem GUI Anwendungsfall validiert

rosbridge_suite

melodic | noetic | *Show EOL distros:* ☑

EOL distros: fuerte | groovy | hydro | indigo | jade | kinetic | lunar

Documentation Status

rosbridge_suite: *rosapi* | *rosbridge_library* | *rosbridge_server*

ROS2

- unser Zielsystem
- rosbridge_suite für ROS2

rosbridge_suite

ros | foxy v1.3.1 | ros | galactic v1.3.1 | ros | humble v2.0.0 | ros | rolling v1.3.2

- python Skript unabhängig von der ROS Version!

rosbridge_suite „Knackpunkte“

- unabhängige Entwicklung von ROS nodes und WebUI bzw. Python Skript
  - trotzdem muss das Interface definiert und synchronisiert werden

- hier helfen die ROS Type-Definitionen und ROS Message-Definitionen

turtlesim/Spawn Service

File: turtlesim/Spawn.srv

Raw Message Definition

```
float32 x
float32 y
float32 theta
string name # Optional.  A unique name will be created and returned if this is empty
---
string name
```

```
# create a Service of name and type
turtle_spawn_service = roslibpy.Service(client, 'spawn', 'turtlesim/srv/Spawn')
# provide that data to the type. See http://docs.ros.org/en/noetic/api/turtlesim/html/s
request = roslibpy.ServiceRequest({'x': 4, 'y': 4, 'theta': 0.2, 'name': 'pyturtle'})
```

rosbridge_suite „Knackpunkte“

- unabhängige Entwicklung von ROS nodes und WebUI bzw. Python Skript
  - trotzdem muss das Interface definiert und synchronisiert werden

- hier helfen die ROS Type-Definitionen und ROS Message-Definitionen

turtlesim/Spawn Service

File: turtlesim/Spawn.srv

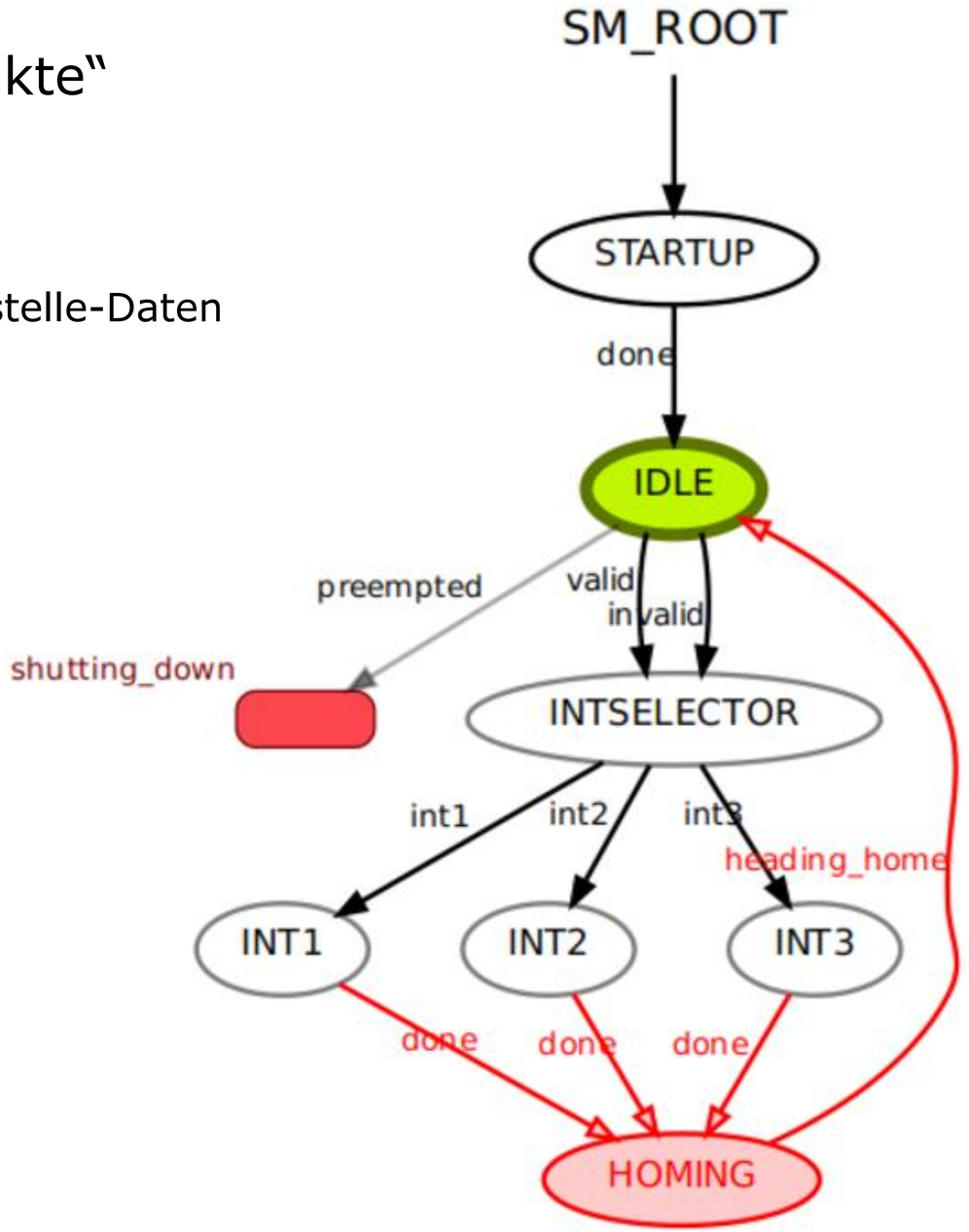
Raw Message Definition

```
float32 x
float32 y
float32 theta
string name # Optional.  A unique name will be created and returned if this is empty
---
string name
```

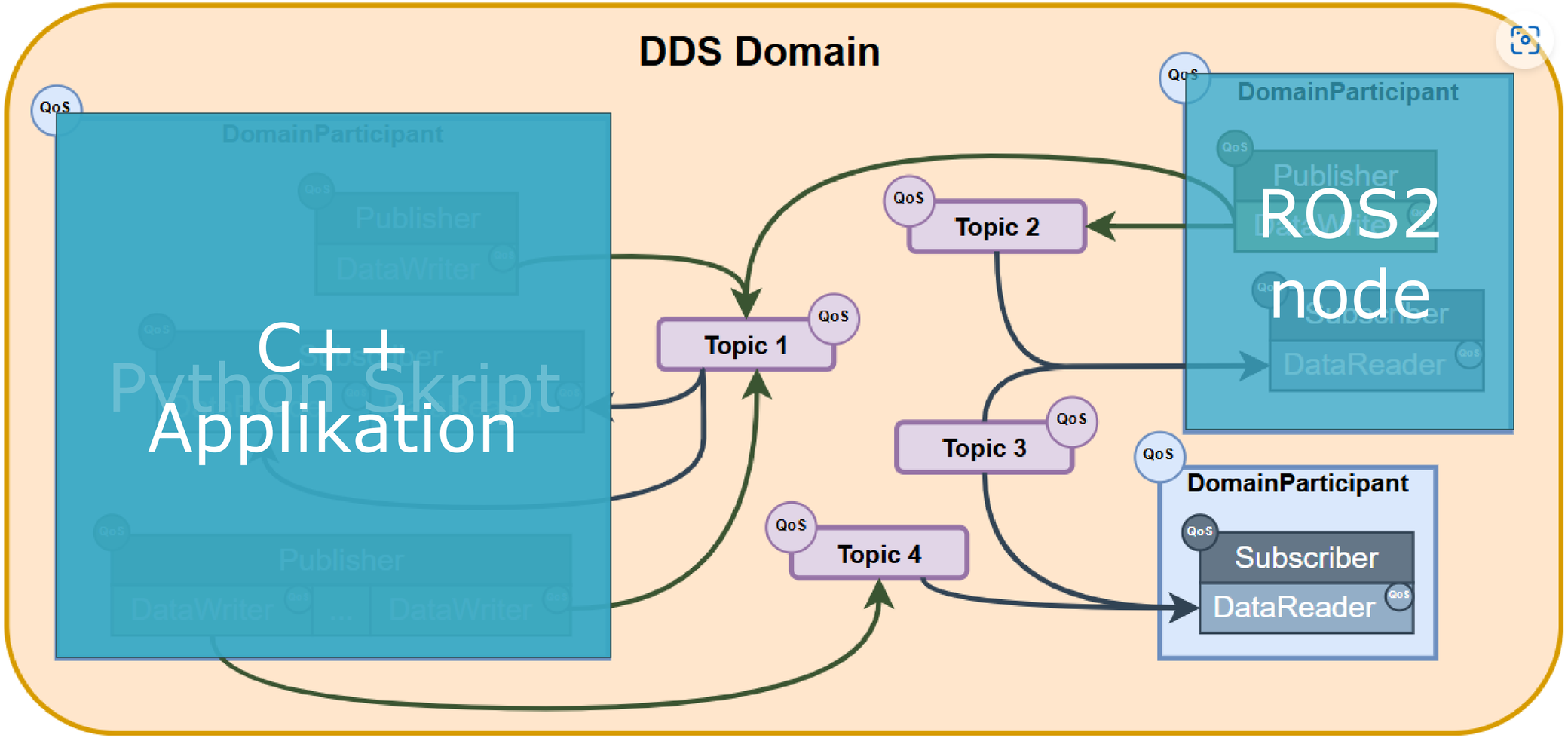
```
# create a Service of name and type
turtle_spawn_service = roslibpy.Service(client, 'spawn', 'turtlesim/srv/Spawn')
# provide that data to the type. See http://docs.ros.org/en/noetic/api/turtlesim/html/s
request = roslibpy.ServiceRequest({'x': 4, 'y': 4, 'theta': 0.2, 'name': 'pyturtle'})
```

rosbridge_suite „Knackpunkte“

- Logik – Verarbeitung der Schnittstelle-Daten

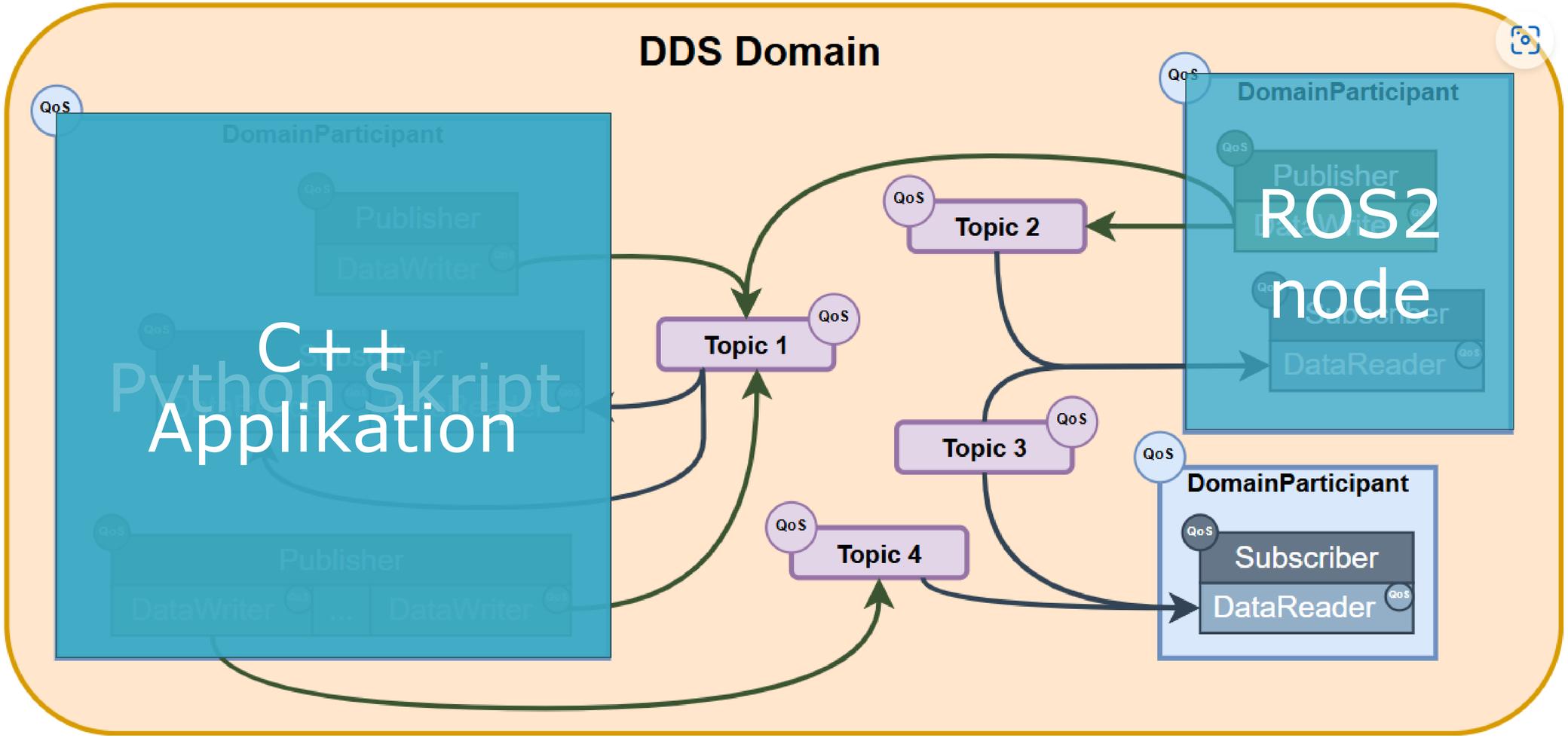


ROS Schnittstelle mit c++ Applikation (eProsima fastDDS)



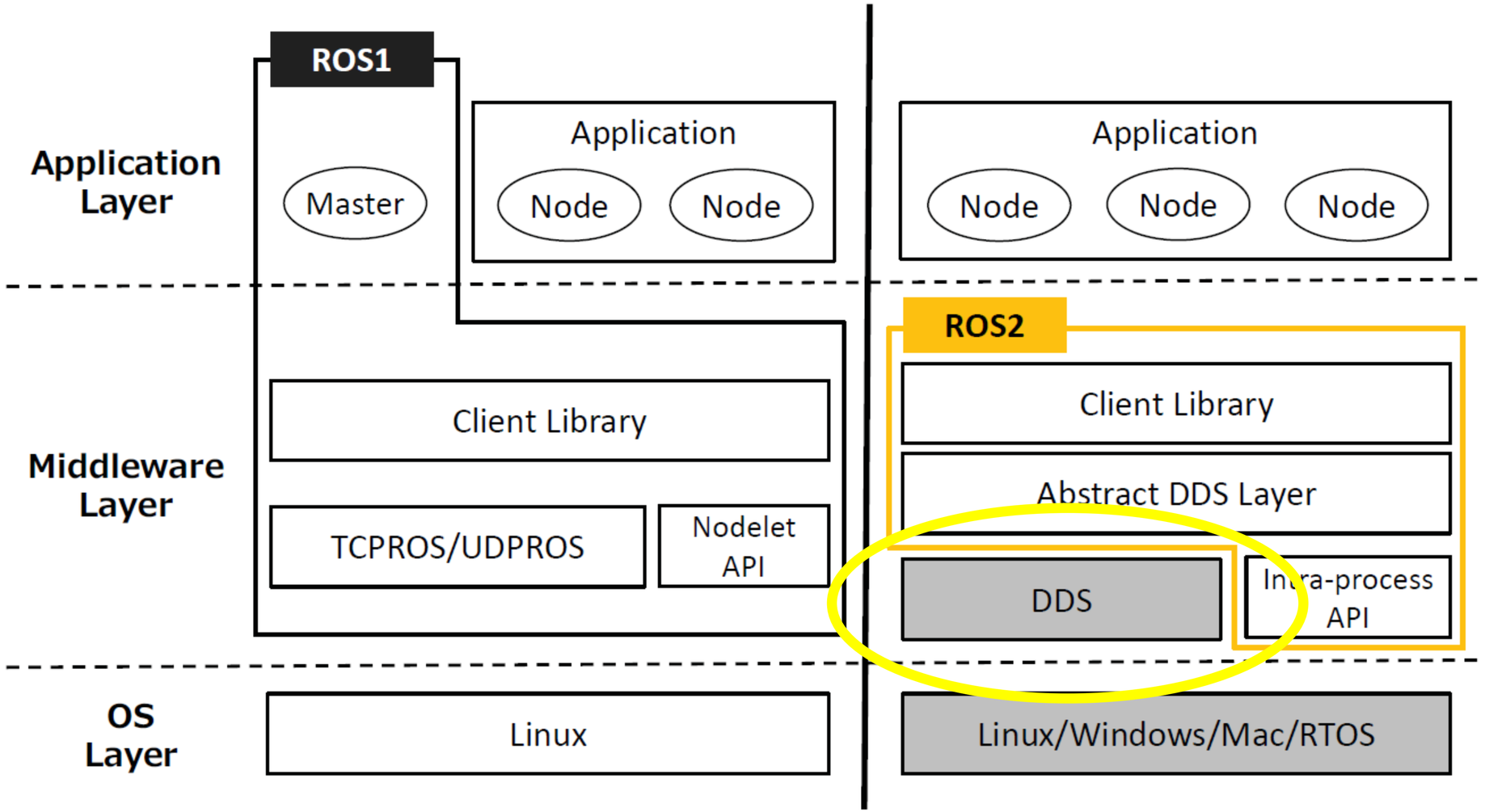
Quelle

ROS Schnittstelle mit c++ Applikation (eProsima fastDDS)



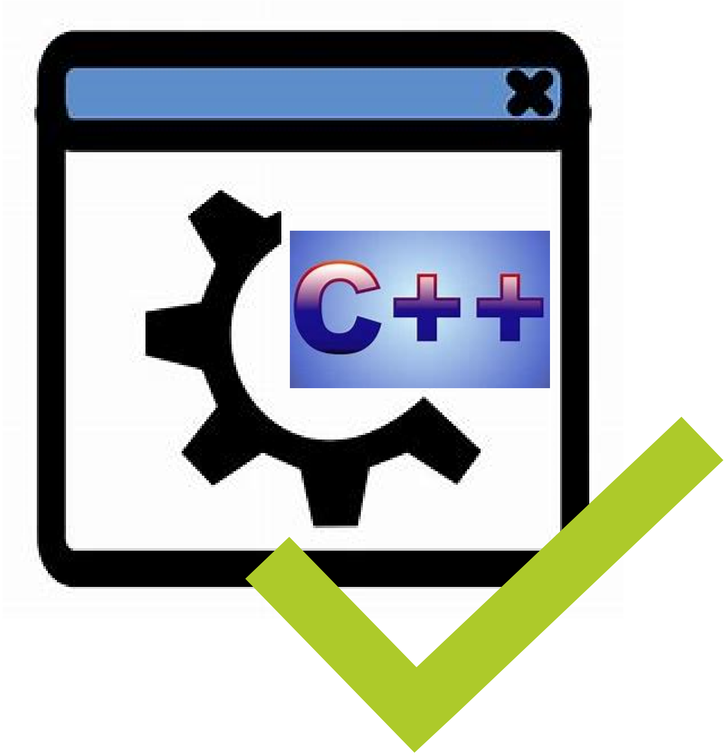
Quelle

DDS – ROS1 vs. ROS2

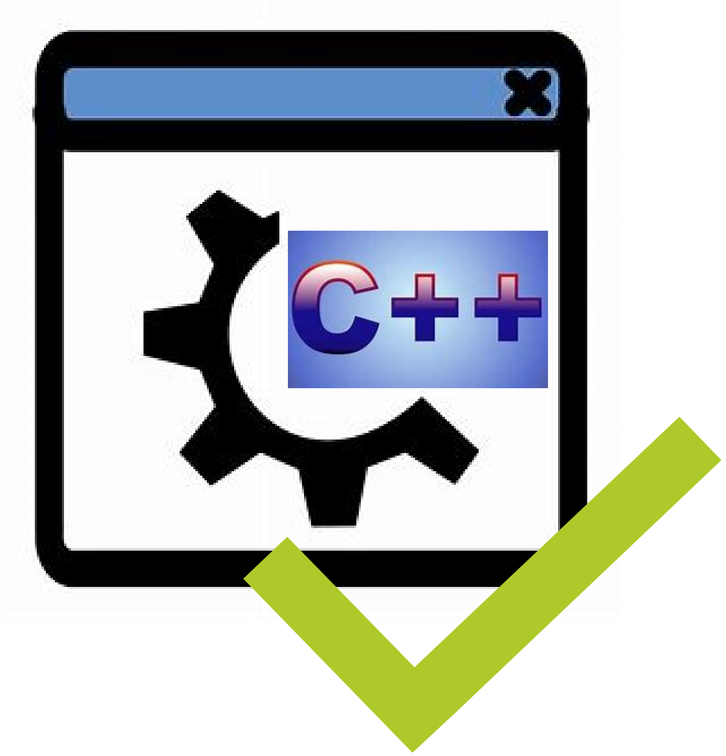


Quelle: Exploring the performance of ROS2, Takuya Azumi, 2016

eProsima fastDDS – unser Status und Ausblick



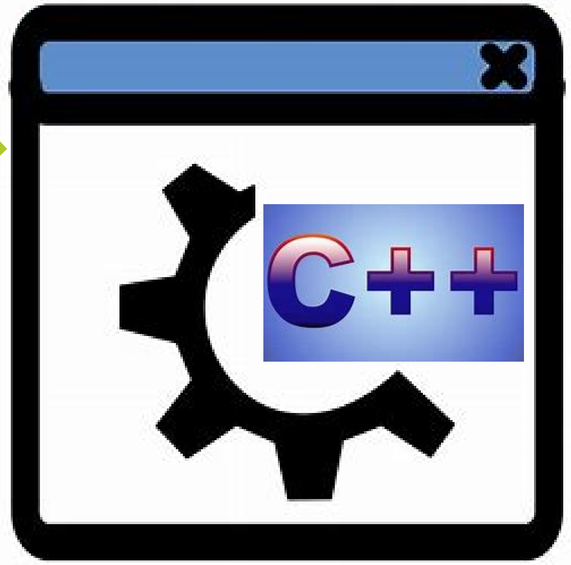
eProsima fastDDS – unser Status und Ausblick



Fazit

roslibpy +
rosbridge_
suite

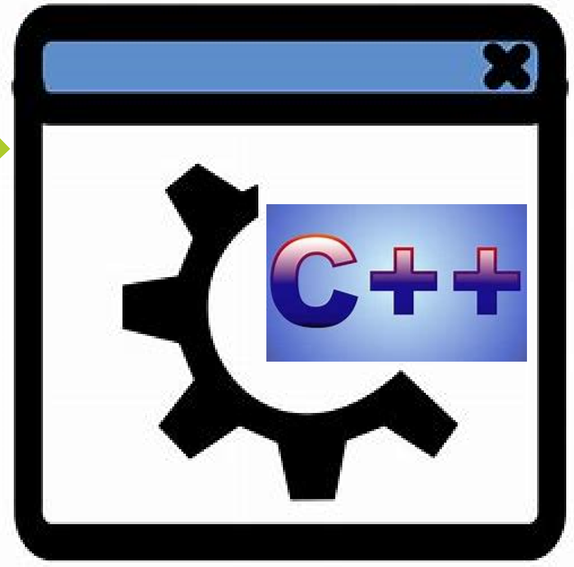
eProsima
fastDDS



Fazit

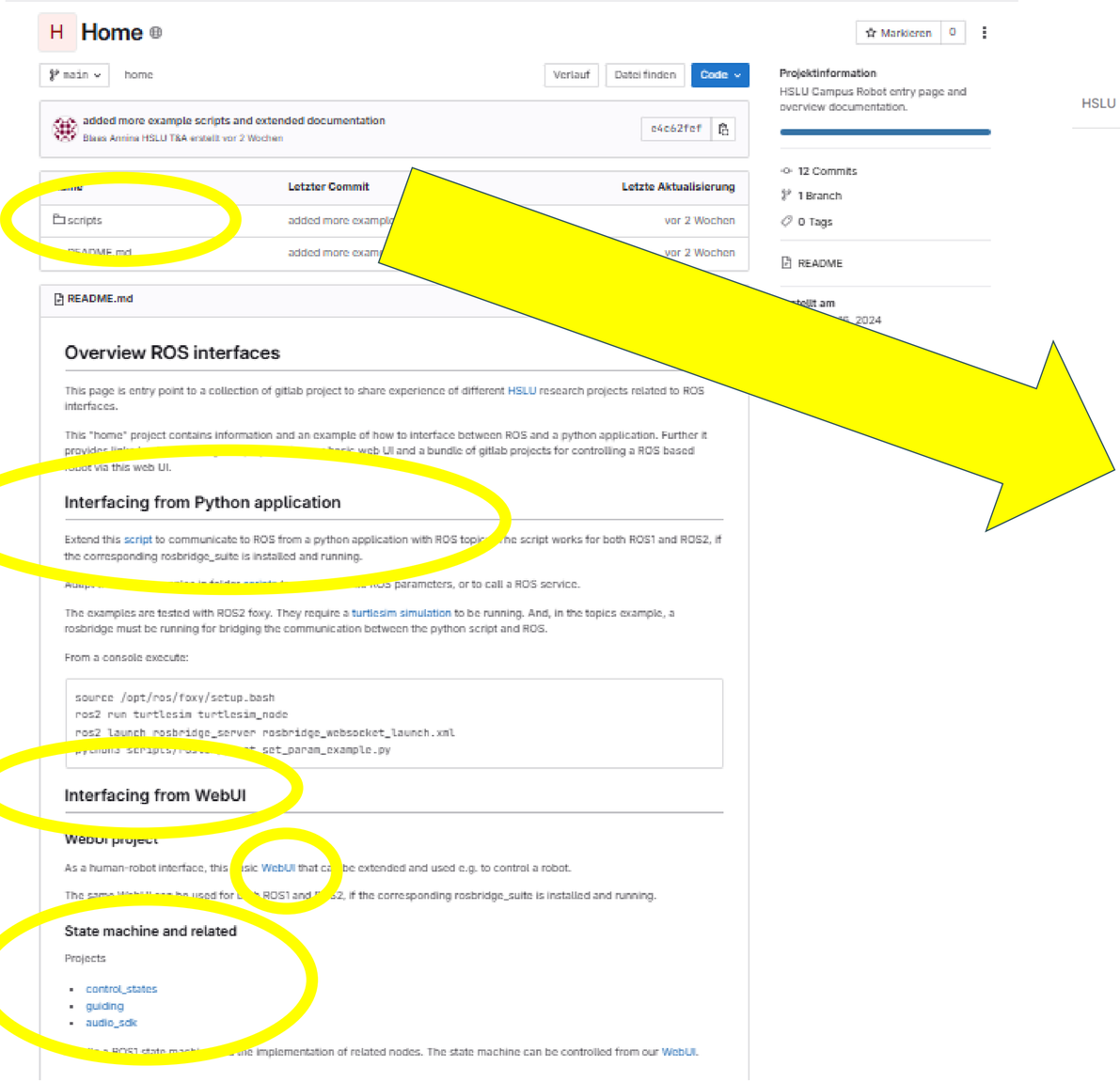
roslibpy +
rosbridge_
suite

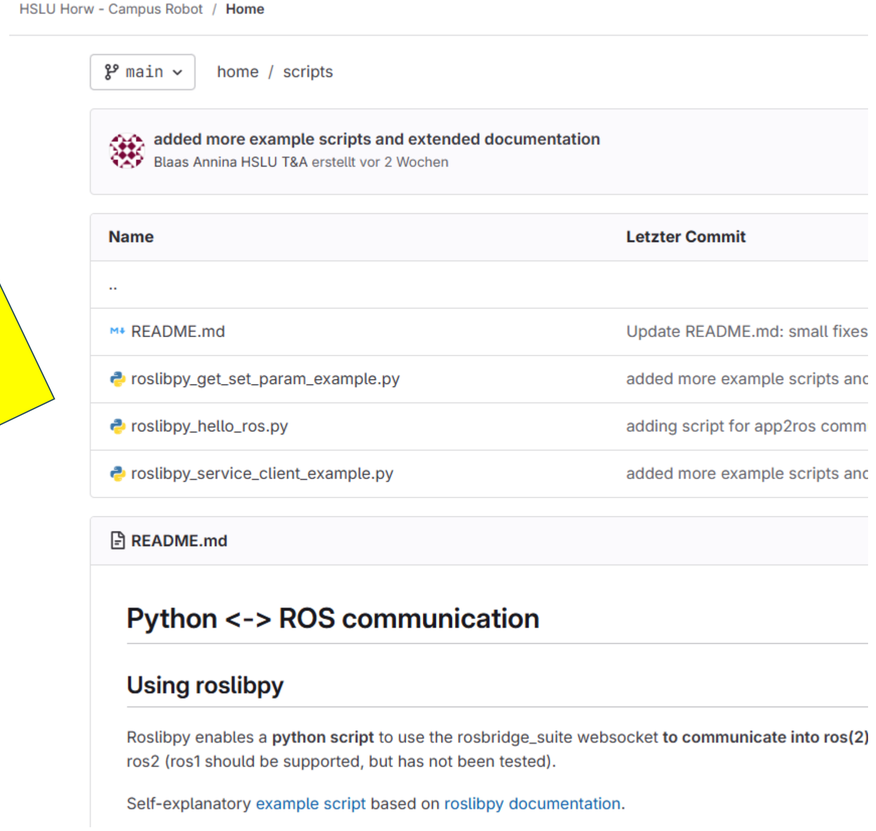
eProsima
fastDDS



vue.js +
roslibjs +
rosbridge_suite

Veröffentlichung auf https://gitlab.switch.ch/hslu-horw-campus-robot/home





Referenzen und weiterführende Links

rosbridge_suite für ROS1

rosbridge_suite für ROS2

roslibjs

roslibpy

eProsima fast DDS Dokumentation

eProsima C++ Tutorial

eProsima Python Tutorial

eProsima fast DDS usage with ROS2

HSLU Hochschule Luzern

Quelle

Hochschule Luzern
Technik & Architektur
Institut für Maschinen- und Energietechnik IME
CC Mechanische Systeme
Annina Blaas
Senior Wissenschaftliche Mitarbeiterin
annina.blaas@hslu.ch

FH Zentralschweiz