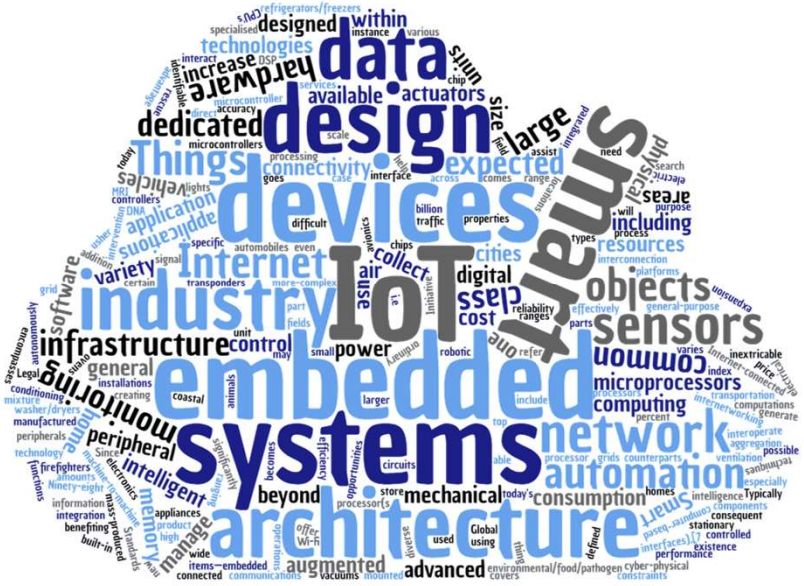
Associate Professorship of Embedded Systems and Internet of Things
Department of Electrical and Computer Engineering
Technical University of Munich

Evaluation of ROS as an Addition to the W3C WoT

by Marcus Wolf Schmidt

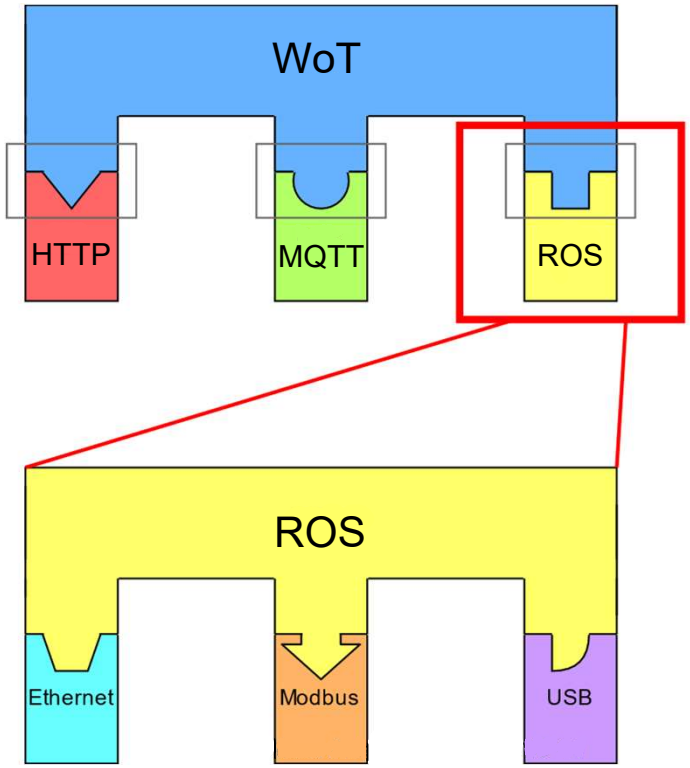
July 3rd, 2020

W3C WoT Binding Templates Task Force Weekly Call

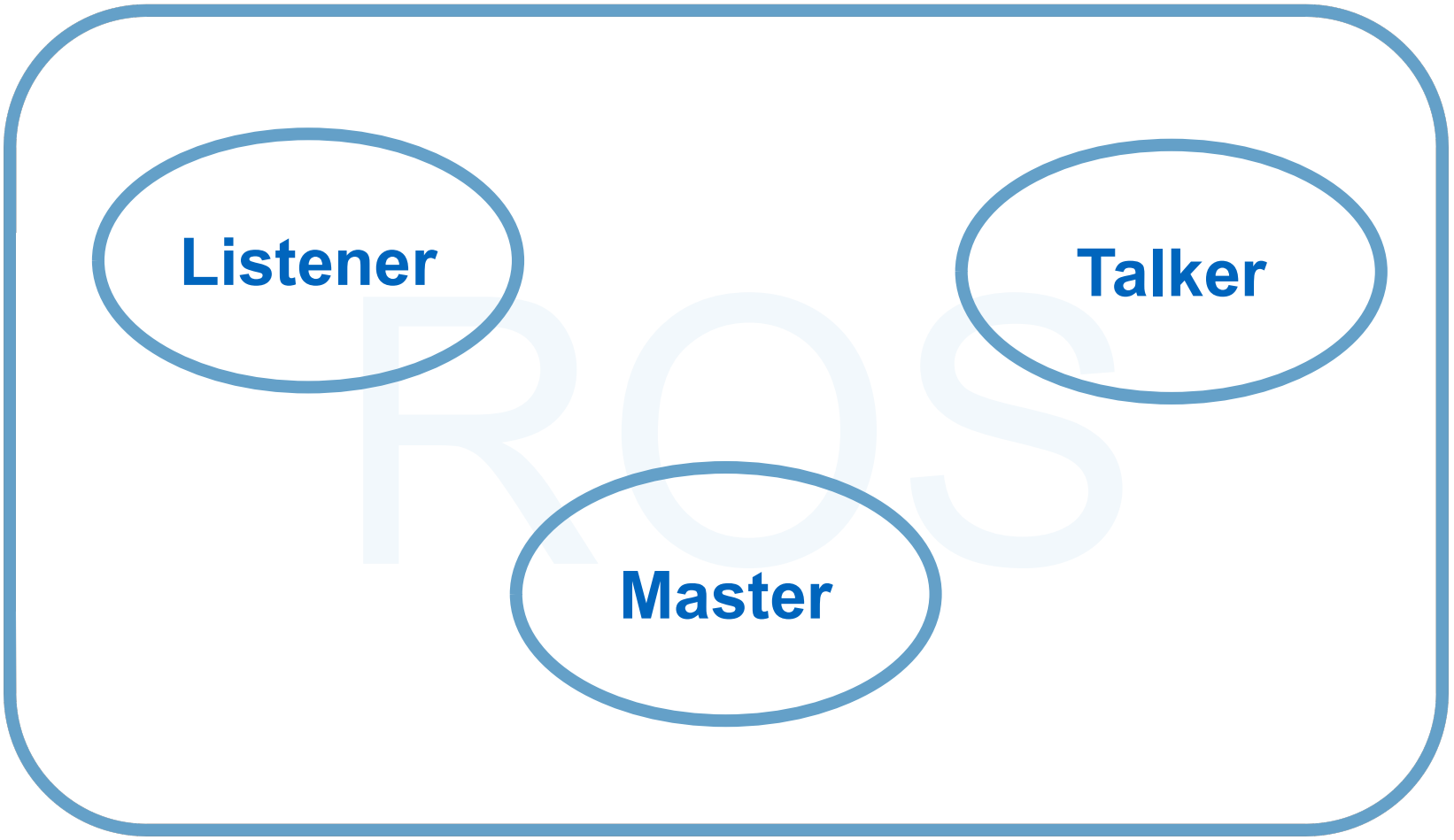


Robot Operating System (ROS)

- Open Source *middleware* for robotics applications
- Allows varies protocols to be used (e.g. Ethernet, Modbus, USB)
- Focus on reusability of code (like WoT scripting API)
- Interest from the W3C WoT Working Group: GitHub Issue



ROS Nodes



ROS Master Node

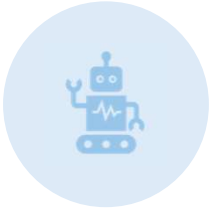
Only ONE per ROS environment

Helps Nodes find each other

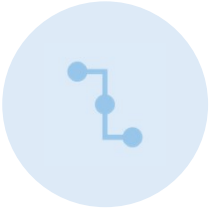
Communicates over XMLRPC

Provides a parameter server

Simple ROS Nodes



Multiple Nodes per environment and device/robot

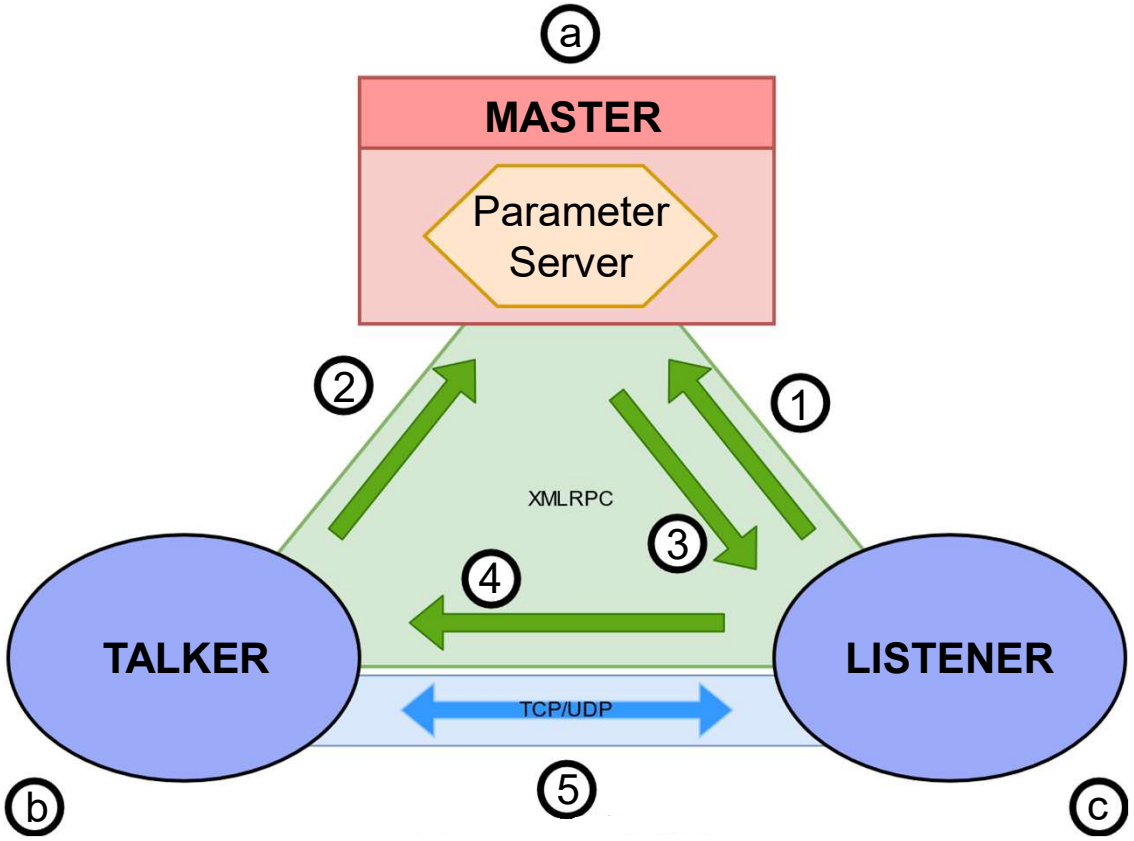


Nodes exchange data directly

Communicates over XMLRPC and other protocols

Nodes register themselves on the Master

ROS Architecture



ROS and the W3C WoT

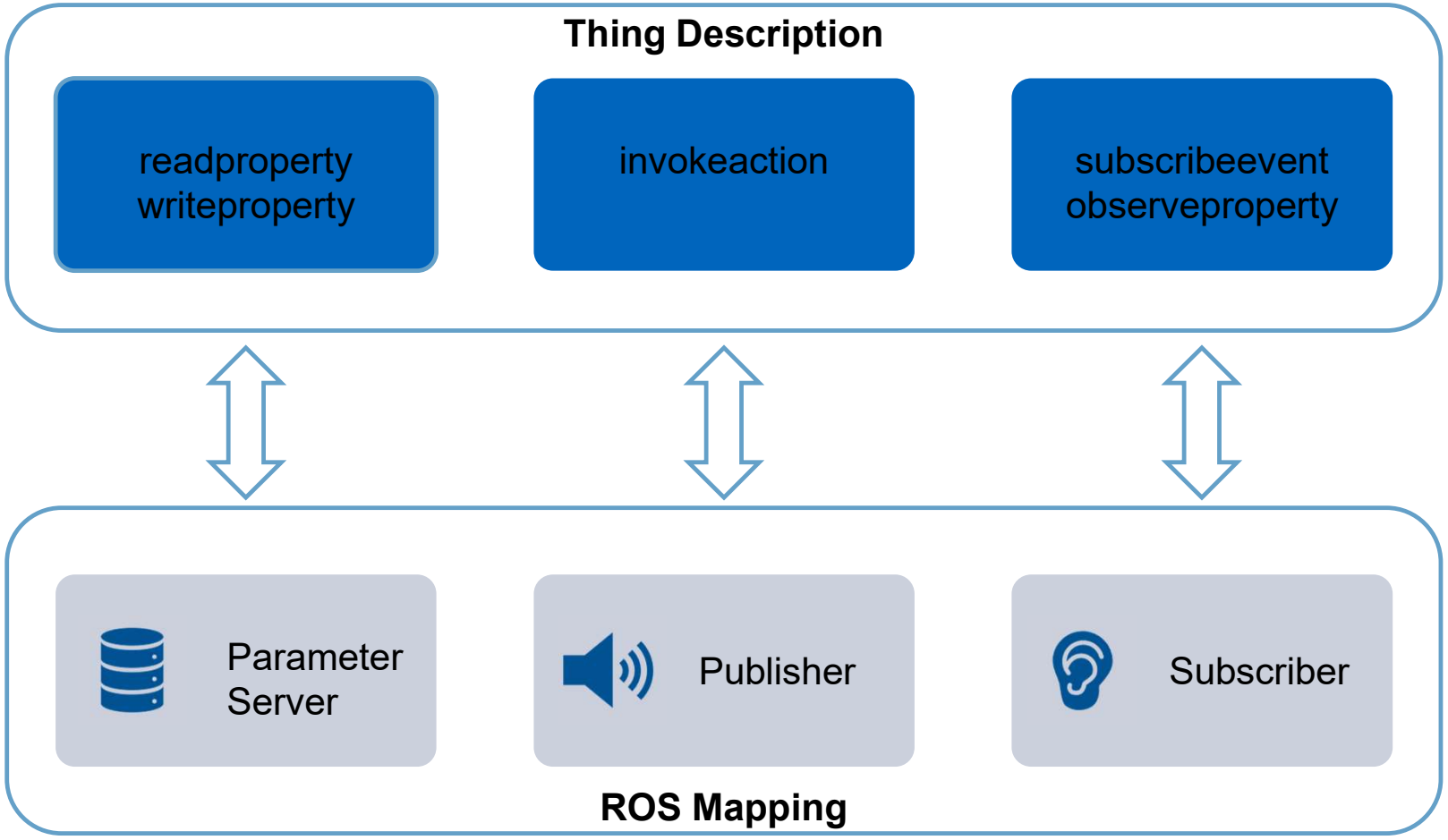
Publisher

Subscriber

Parameter Server

Services

W3C WoT Thing Description



What about Services

Publisher

Subscriber

Parameter Server

Services

Thing Description Vocabulary

```
{
"@context": ["https://www.w3.org/2019/wot/td/v1",
            {"ros":"http://www.example.org/ros-binding"}],
"title": "Uarm",
"properties": {
      "location":{
            "title":"Return location",
            "type":"integer",
            "forms": [{
                  "href":"ros.xmlrpc://192.168.0.100:11311/uarm/properties/location",
                  "contentType":"application/json",
                  "op":"readproperty",
                  "ros:methodName":"getParam"
                  }]
       }
 },
 "actions":{
      "beep":{
            "title":"Beep",
            "forms":[{
                  "href":"ros.tcp://192.168.0.100:11311/uarm/actions/beep",
                  "contentType":"ROS/String",
                  "op":"invokeaction",
                  "ros:registerClass":"Publish"
                  }]
       }
 },
 "events":{
      "error":{
            "title":"Error Event",
            "forms":[{
                  "href":"ros.tcp://192.168.0.100:11311/uarm/events/error",
                  "contentType":"ROS/String",
                  "op":"subscribeevent",
                  "ros:registerClass":"Subscribe"
                  }]
       }
}
}
```

Vocabulary Table

Vocabulary	Description	Type
ros:methodName	Defines the interaction method	“getParam“, “setParam“
ros:registerClass	Defines the object class	“Publish“, “Subscribe“, (“Service“)

Vocabulary Mapping

op value	ROS
readproperty	getParam
writeproperty	setParam
observeproperty	Subscribe
invokeaction	Publish
subscribeevent	Subscribe

Discussion

1. How to describe the ROS URI Scheme: HTTP or new URI Scheme
2. What `contentType` to choose: Relevant GitHub issue in Thing Description

Discussion Results (added after presenting)

Instead of `ros.xmlrpc://rosmaster` in `href` of `forms`, we will use `http://rosmaster` and `subprotocol:"ros.xmlrpc"`

Reasoning behind:

- Since the protocol over the wire is HTTP, this is more adequate
- A subprotocol makes it easier to describe the protocol that happens over the payload, otherwise we would need specify the concrete payload mechanism that XMLRPC uses in the DataSchema of each interaction.

Discussion Results

Instead of `application/json` in the `contentType`, we will use `application/xml` since the actual data on the wire is XML but the DataScheme allows us to properly describe the needed structure.

Discussion Results: Example TD

```
{"@context": ["https://www.w3.org/2019/wot/td/v1",
                {"ros":"http://www.example.org/ros-binding"}],
 "title": "Uarm",
 "properties": {
        "location":{
              "title":"Return location",
              "type":"integer",
              "forms": [{
                    "href":"http://192.168.0.100:11311/uarm/properties/location",
                    "contentType":"application/xml",
                    "subprotocol":"ros.xmlrpc",
                    "op":"readproperty",
                    "ros:methodName":"getParam"
                    }]
         }
 },
 "actions":{
        "beep":{
              "title":"Beep",
              "forms":[{
                    "href":"http://192.168.0.100:11311/uarm/actions/beep",
                    "contentType":"ROS/String",
                    "subprotocol":"ros.tcp",
                    "op":"invokeaction",
                    "ros:registerClass":"Publish"
                 }]
        }
 },
 "events":{
        "error":{
              "title":"Error Event",
              "forms":[{
                    "href":"http://192.168.0.100:11311/uarm/events/error",
                    "contentType":"ROS/String",
                    "subprotocol":"ros.tcp",
                    "op":"subscribeevent",
                    "ros:registerClass":"Subscribe"
                 }]
 }}
```

Conclusion

Thank you for your attention