

ROS-2 communication optimisation using Zenoh ROS2 Bridge

ROSConDE 2023 - November, 24th 2023 - Karlsruhe

Phani Gangula, MSc.

Senior Solutions Architect

Phani@zettascale.tech

Challenges

“The most dangerous phrase in the language is, ‘We’ve always done it this way.’” - Rear Admiral Grace Hopper

Why DDS Cannot Scale?

There are some fundamental aspects of the DDS protocol that don't make it scalable nor suitable for open systems

These relates to its:

- Discovery Protocol — $O(T \cdot R \cdot W \cdot P^2)$ (Topics, Readers, Writers, Participants)
- Reliability protocol (resource usage linear in the number of matching readers)
- Liveliness Protocol
- Semantics exposed to application w.r.t. what you can know about your system, e.g. liveliness, number of matching readers/writers, finding topics, etc.



RPC over DDS

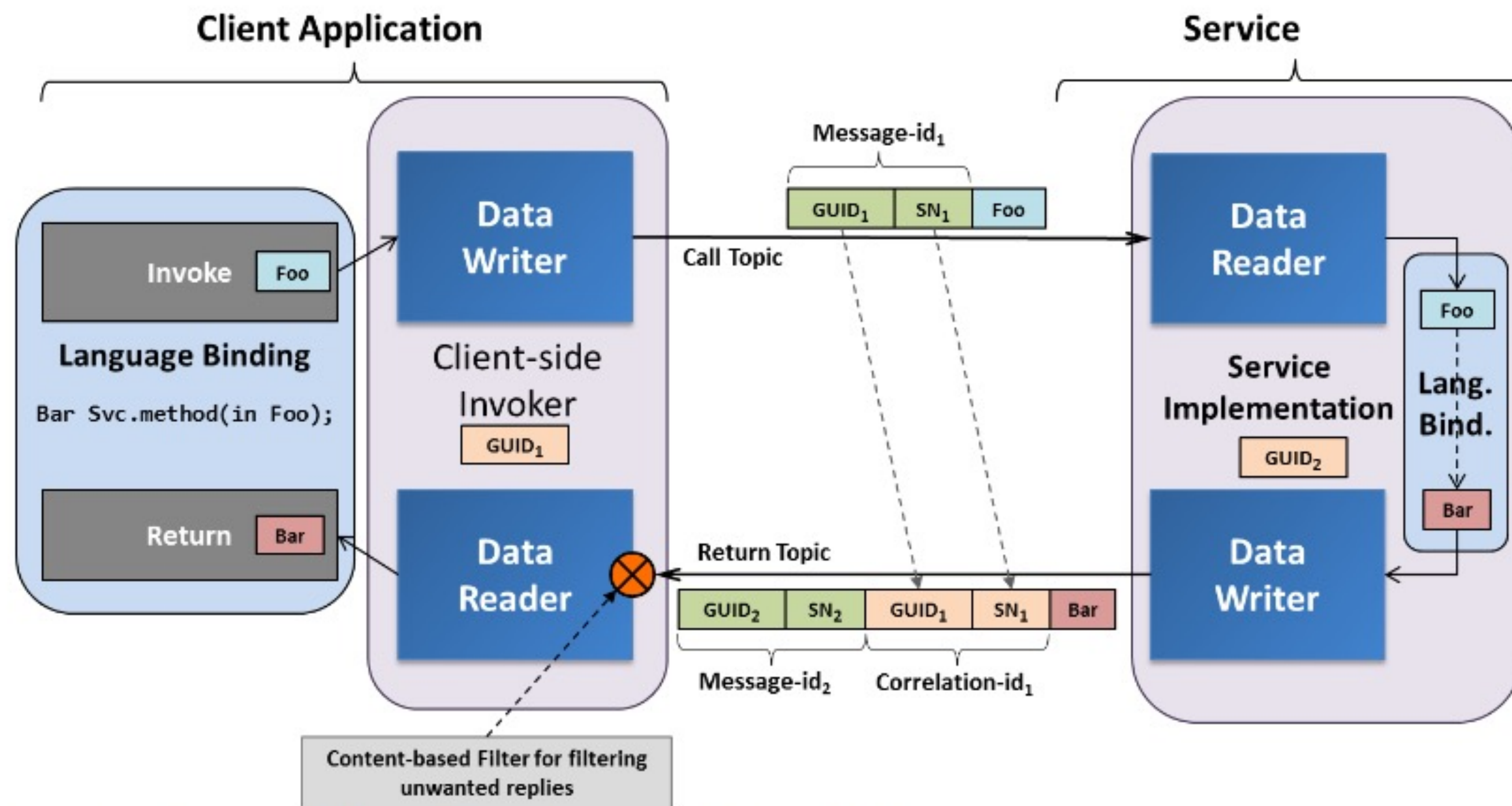


Figure 7.1 - Conceptual View of Remote Procedure Call over DDS

Intrinsics / Open Robotics in 2023

Zenoh selected as the first non
DDS protocol to be natively
supported

Intrinsic implementing Zenoh
RMW as the major
contribution for next ROS2
release

Very insightful report, worth
reading

2023-09 ROS 2 RMW alternate

| | |
|--|----|
| Abstract | 1 |
| User Challenges with DDS | 2 |
| DDS has a fully-connected graph of participants..... | 2 |
| DDS uses UDP multicast for discovery..... | 3 |
| DDS can have difficulty transferring large data..... | 3 |
| DDS can struggle on some WiFi networks..... | 4 |
| DDS tends to have complex tuning parameters..... | 4 |
| Vendor specific non-standard DDS extensions..... | 4 |
| Next Steps..... | 5 |
| Requirements gathering | 5 |
| User Survey..... | 5 |
| Demographics..... | 6 |
| Technical Data..... | 6 |
| Alternative middleware options..... | 8 |
| Requirements..... | 9 |
| Comparative analysis of currently available middlewares | 11 |
| Complete list of investigated middlewares..... | 11 |
| Performance..... | 13 |
| Middlewares X Requirements..... | 13 |
| Conclusion | 14 |
| Appendix A | 15 |

Conclusion

Requirements from ROS 2 users were gathered, and middleware options that are available today were investigated. The research has concluded that Zenoh best meets the requirements, and will be chosen as an alternative middleware. Zenoh was also the most-recommended alternative by users. It can be viewed as a modern version of the TCPROS implementation, and meets most of the ROS 2 requirements. There are still a number of design decisions to be made regarding this implementation; those details will be discussed on <https://discourse.ros.org> as development begins.

Zenoh

Distribute, compute, and store data at the most convenient location at any time



Zenoh

Pub/Sub/Query protocol that **Unifies data in motion, data at rest and computations** from embedded microcontrollers up the data centre

Provides **location-transparent** abstractions for **high performance** pub/sub and **distributed queries** across heterogeneous systems

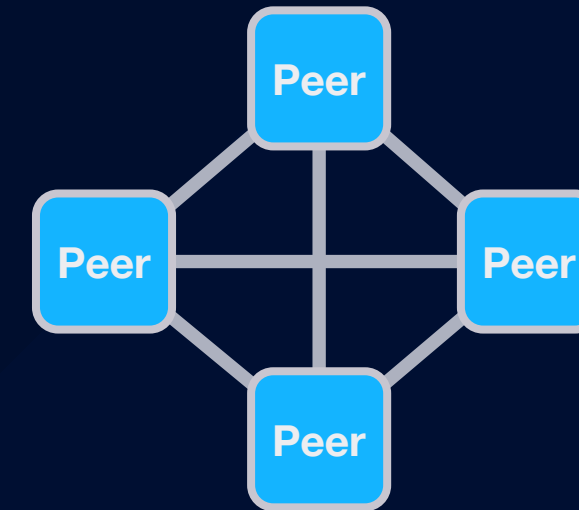
Provides **universal abstractions** for cloud-to-device **data-flow programming**

Any Topology

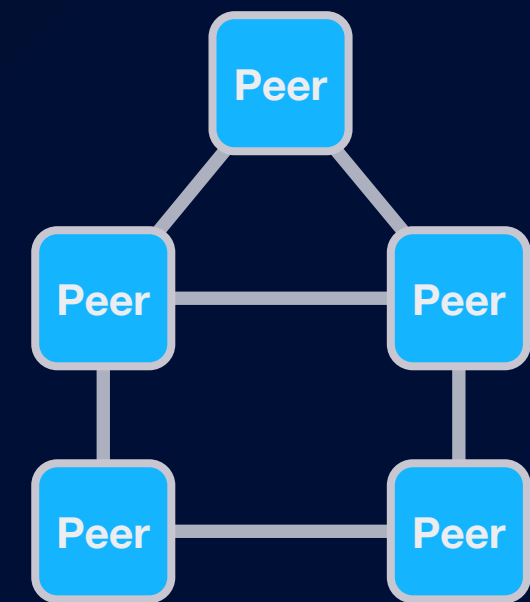
Peer-to-peer

Clique and mesh topologies

Clique



Mesh



Any Topology

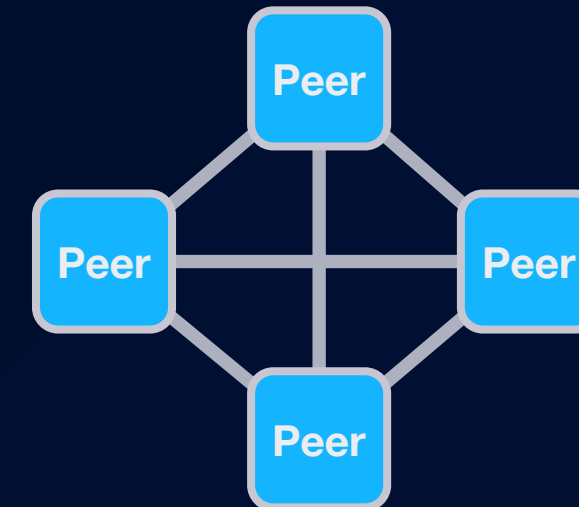
Peer-to-peer

Clique and mesh topologies

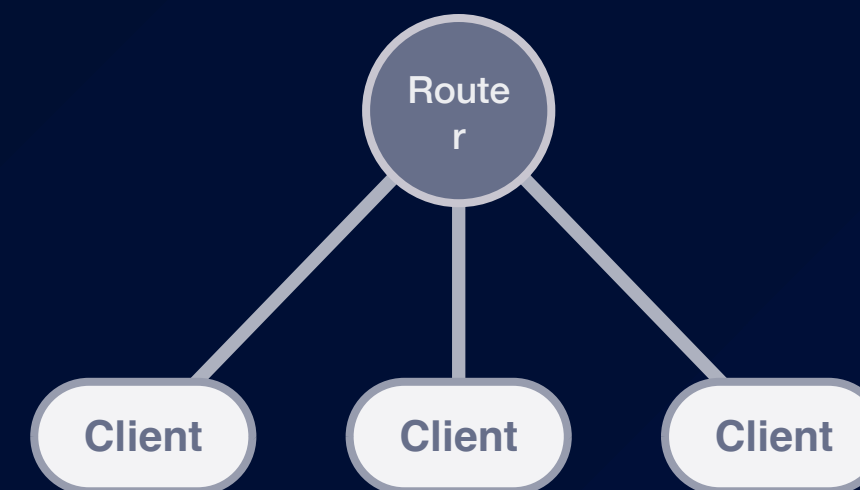
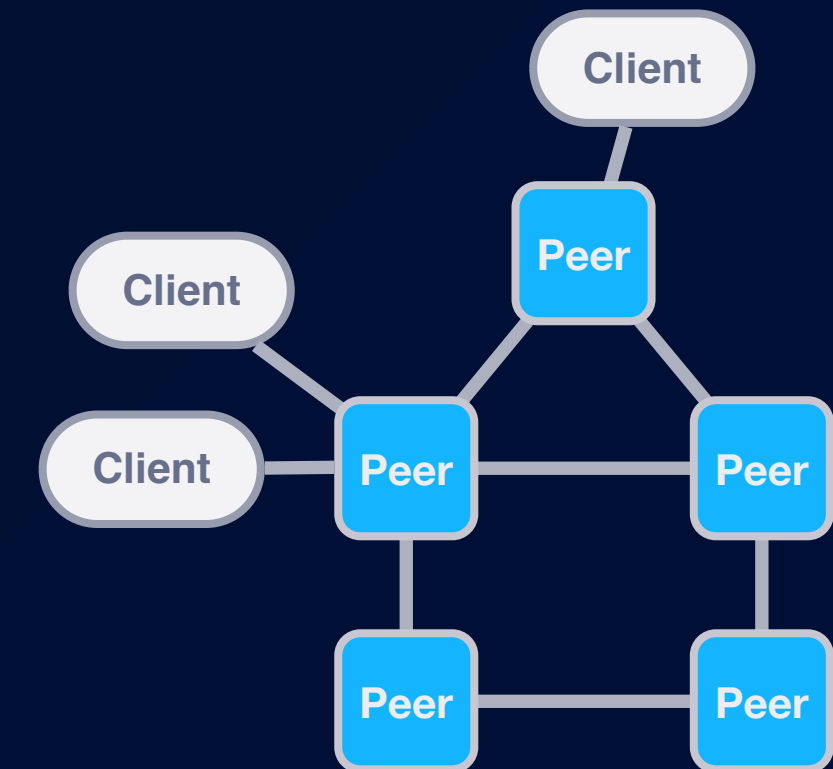
Brokered

Clients communicate through
a router or a peer

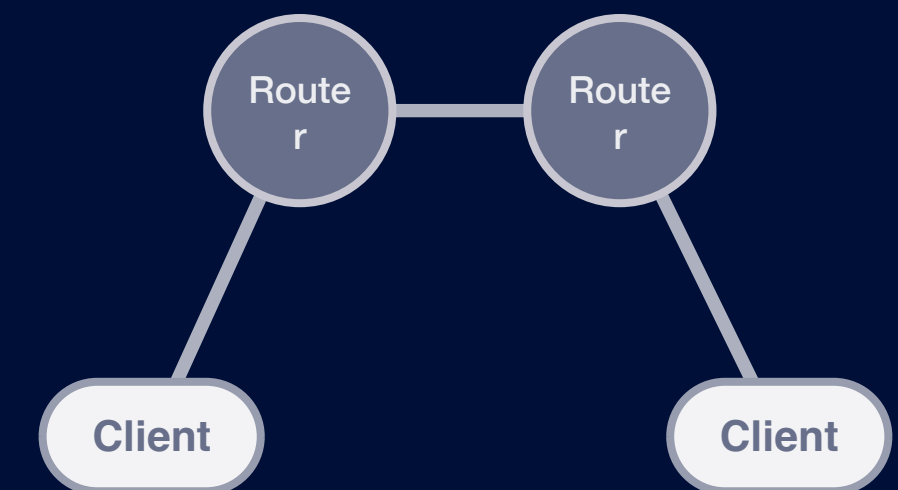
Clique



Mesh



Brokered



Routed

Any Topology

Peer-to-peer

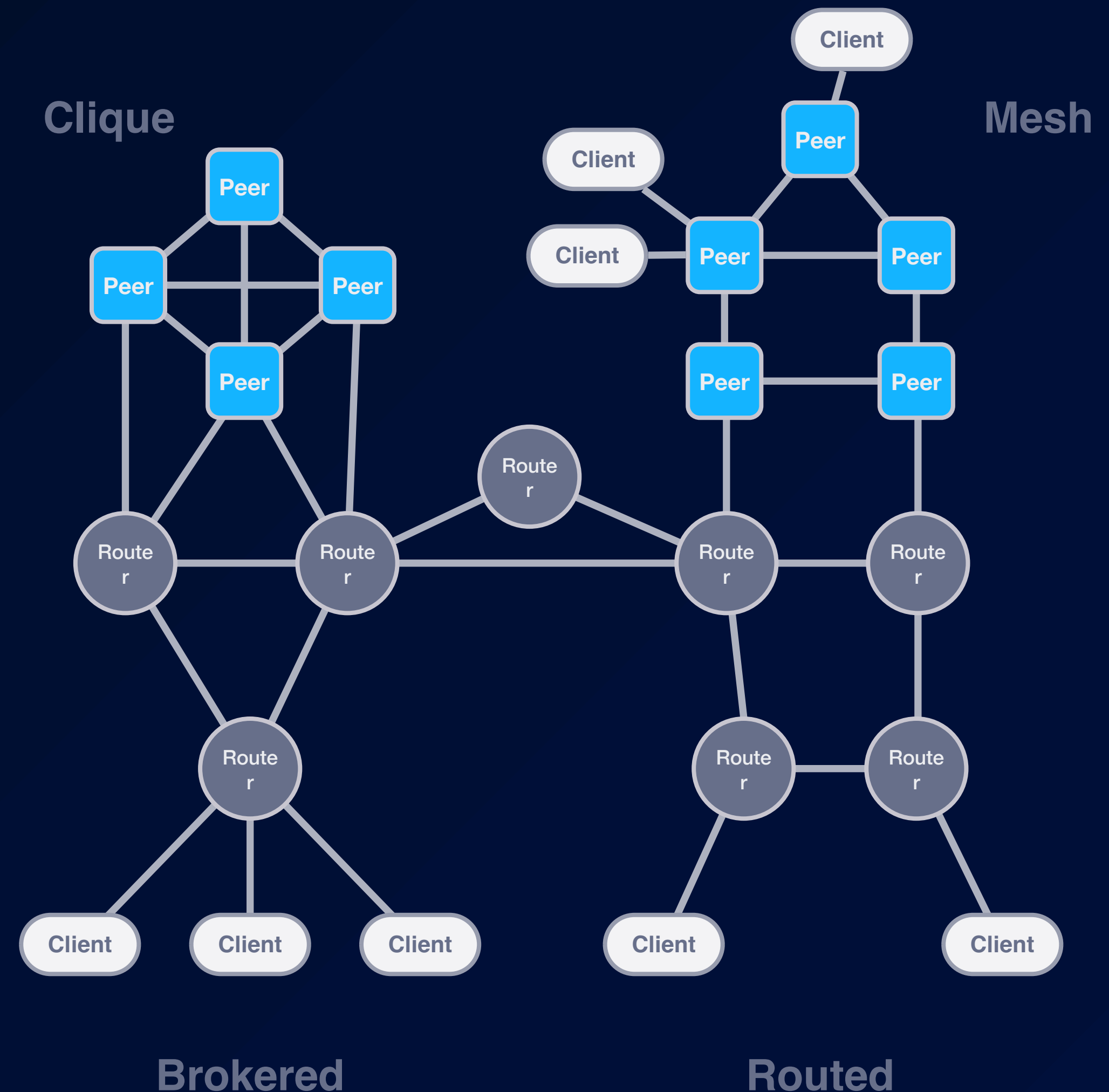
Clique and mesh topologies

Brokered

Clients communicate through a router or a peer

Routed

Routers forward data to and from peers and clients



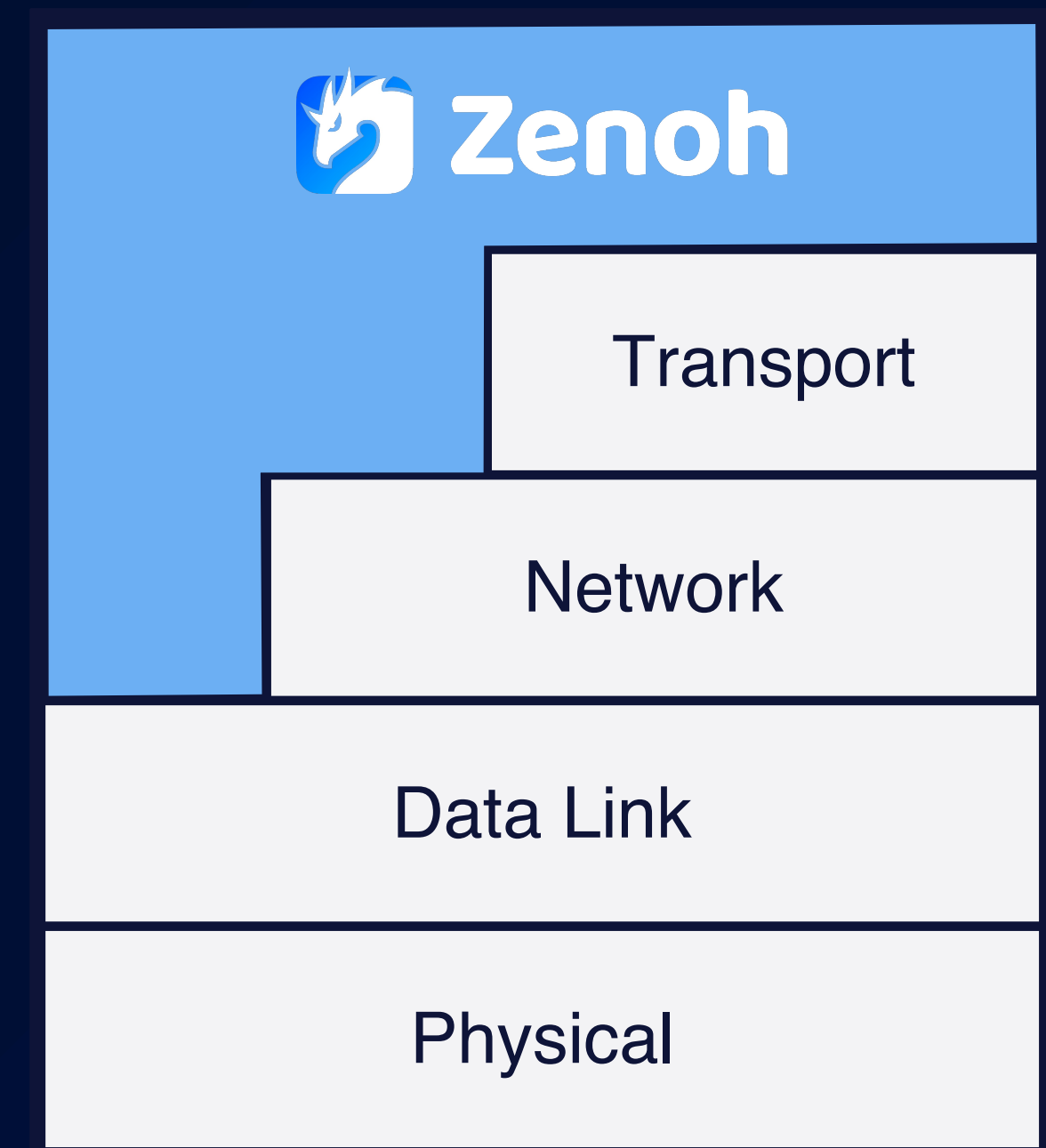
Runs Everywhere

Written in Rust for security, safety and performance

Native libraries and **API bindings** for many **programming languages**, e.g., Rust, C/C++, Python, JS, REST, C# and Kotlin

Supports **network technologies** from **transport layer down-to** the **data link**. Currently runs on, TCP/IP, UDP/IP, QUIC, Serial, Bluetooth, OpenThreadX, Unix Sockets.

Available on **embedded** and **extremely constrained devices** and **networks** — 5 bytes minimal overhead

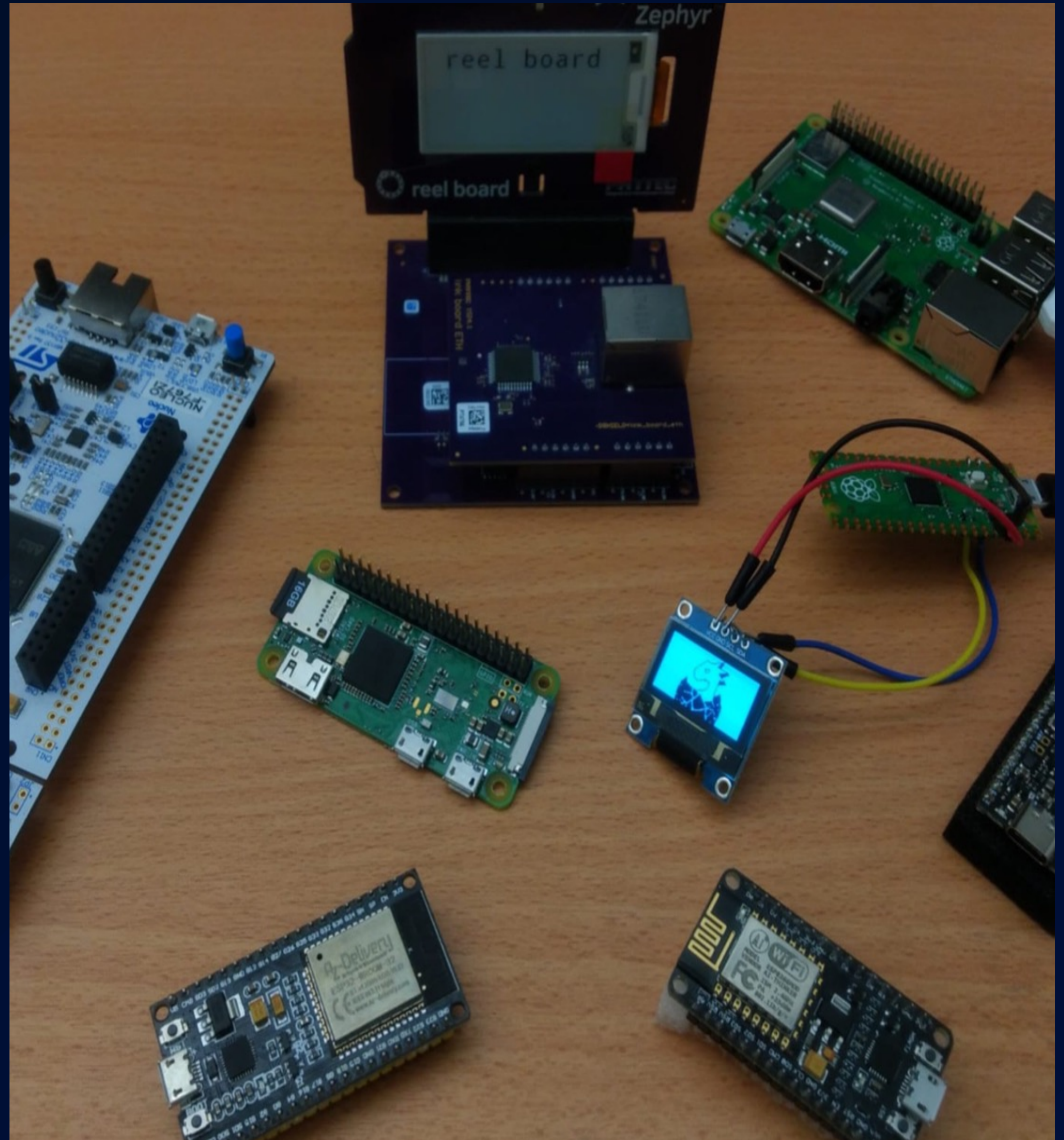


Zenoh on MCUs

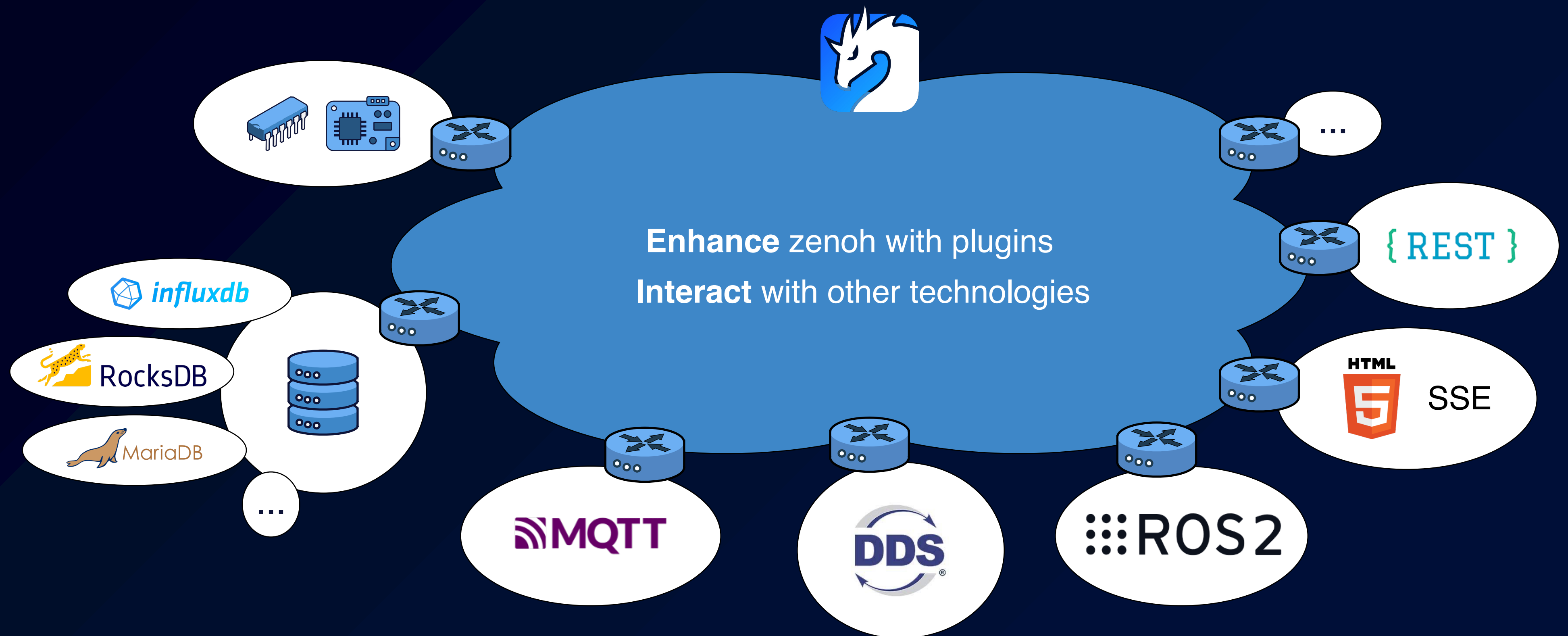
We provide a MISRA-C implementation of the Zenoh protocol targeting extremely embedded targets

Working on a no-std Rust version for bare-metal micro-controllers

The reference for usability and performance [see this blog](#)

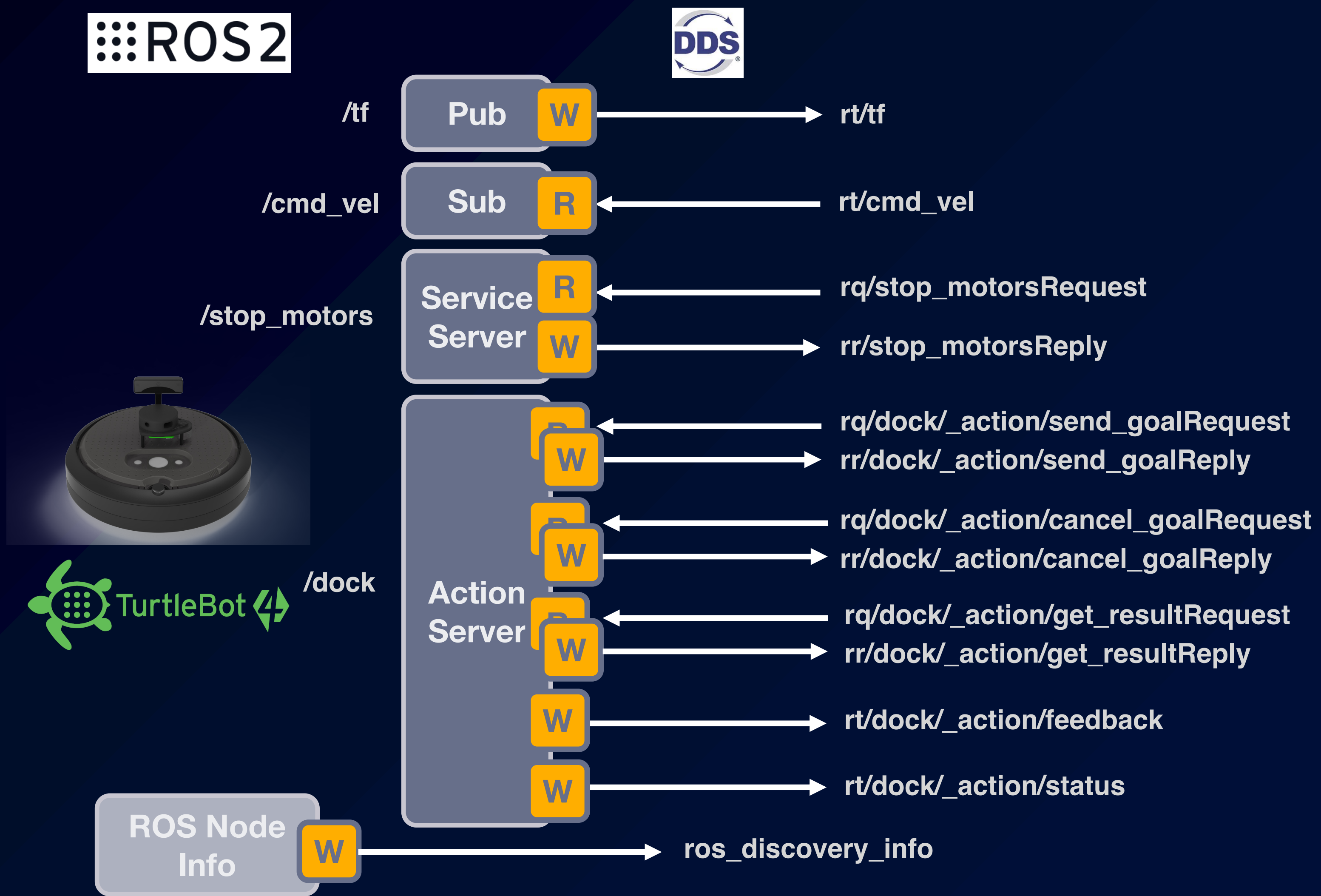


Extensible

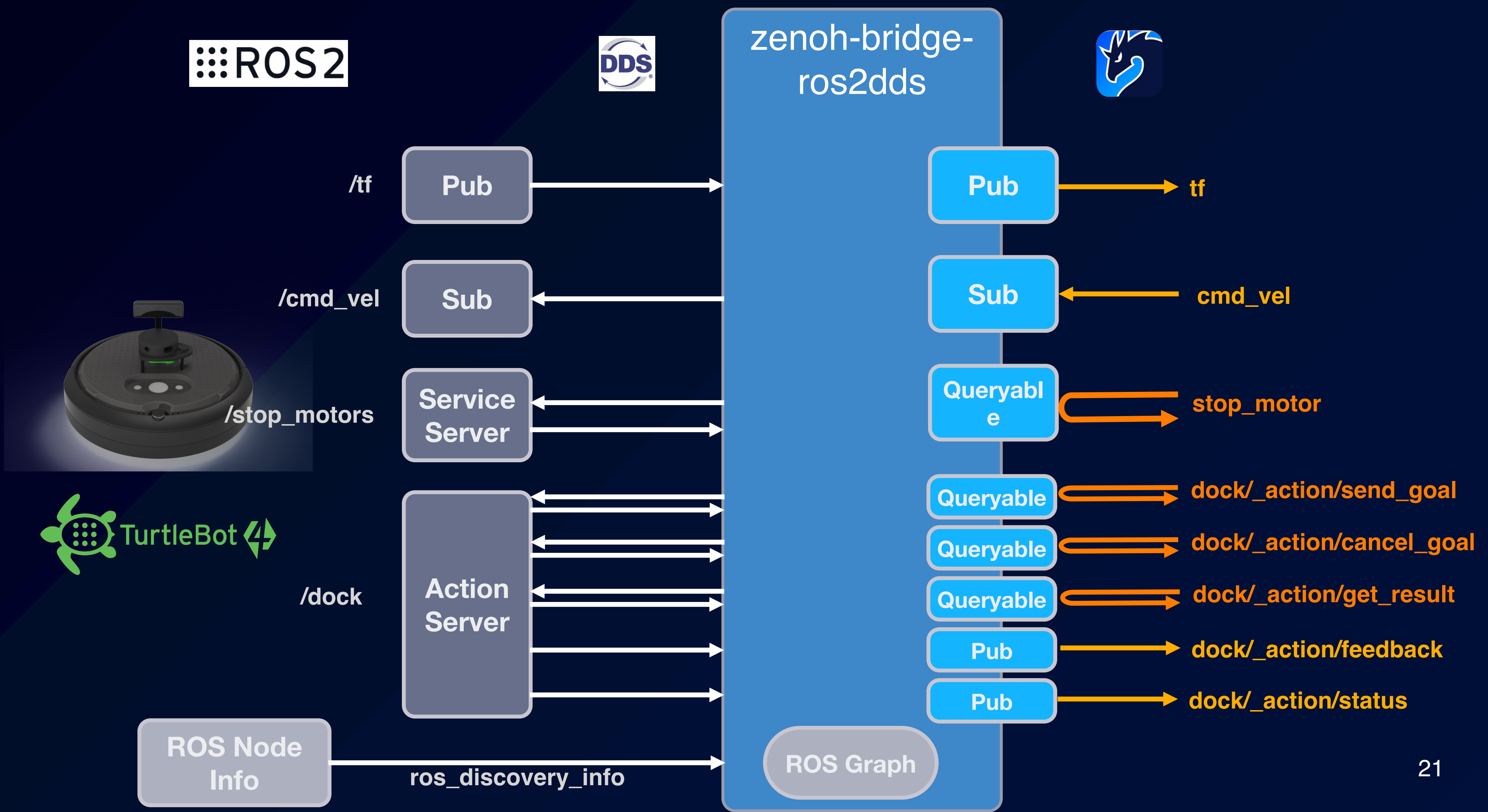


Zenoh plugin for ROS 2 over DDS

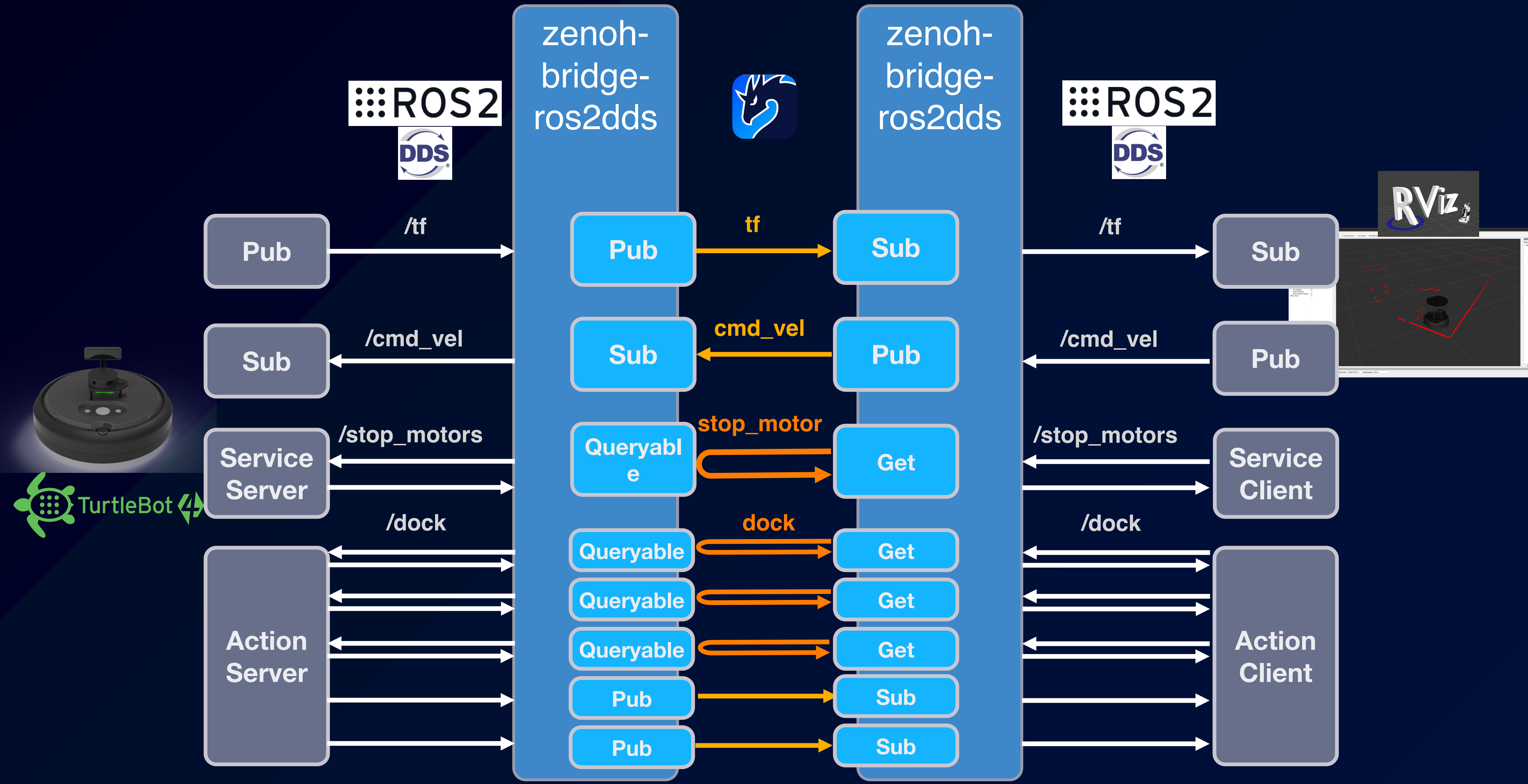
ROS 2 over aDDS



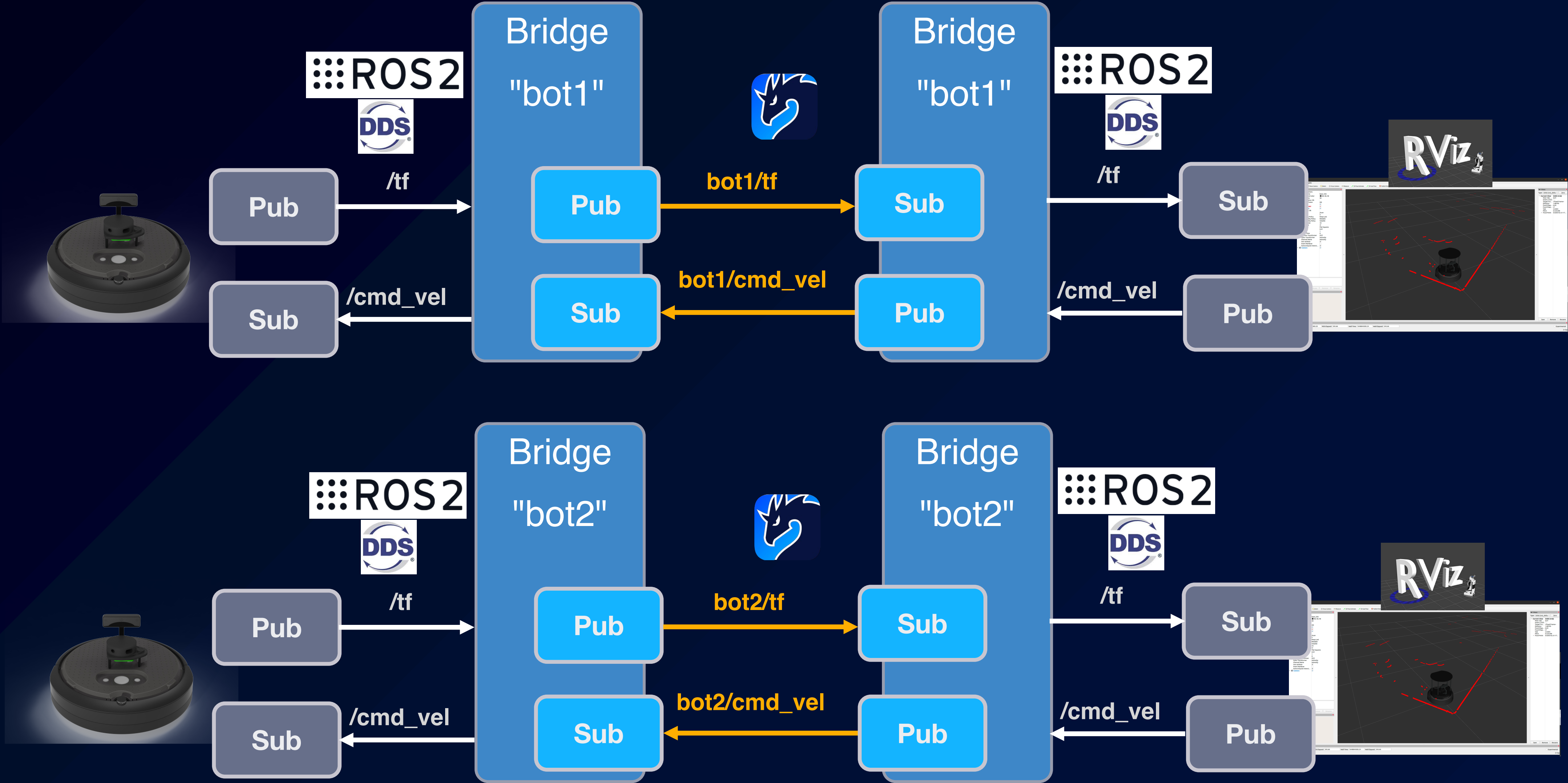
Mapping to Zenoh



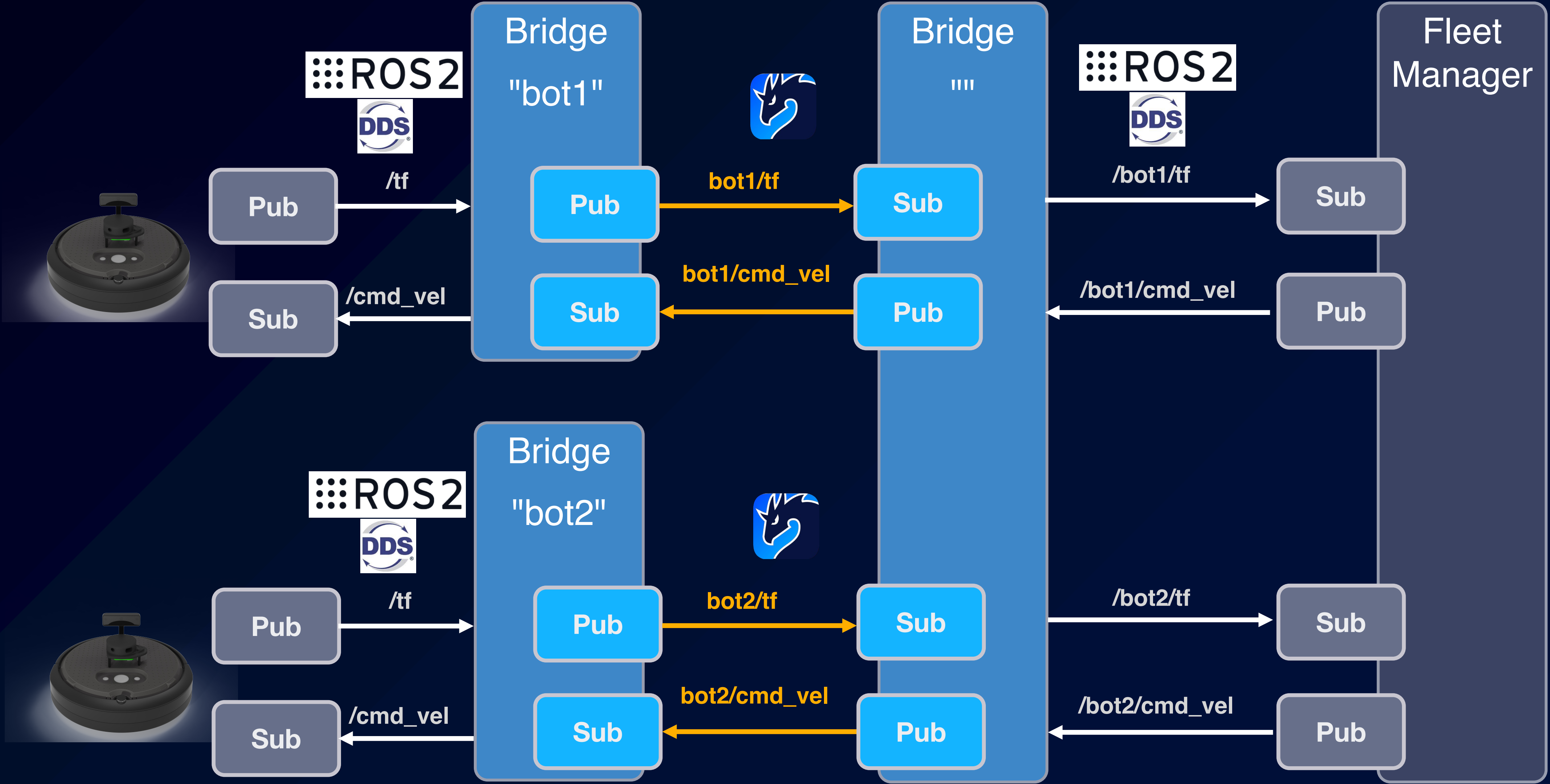
ROS-to-ROS via Zenoh Bridge



Namespace on the bridge



Multi-robots via namespaces

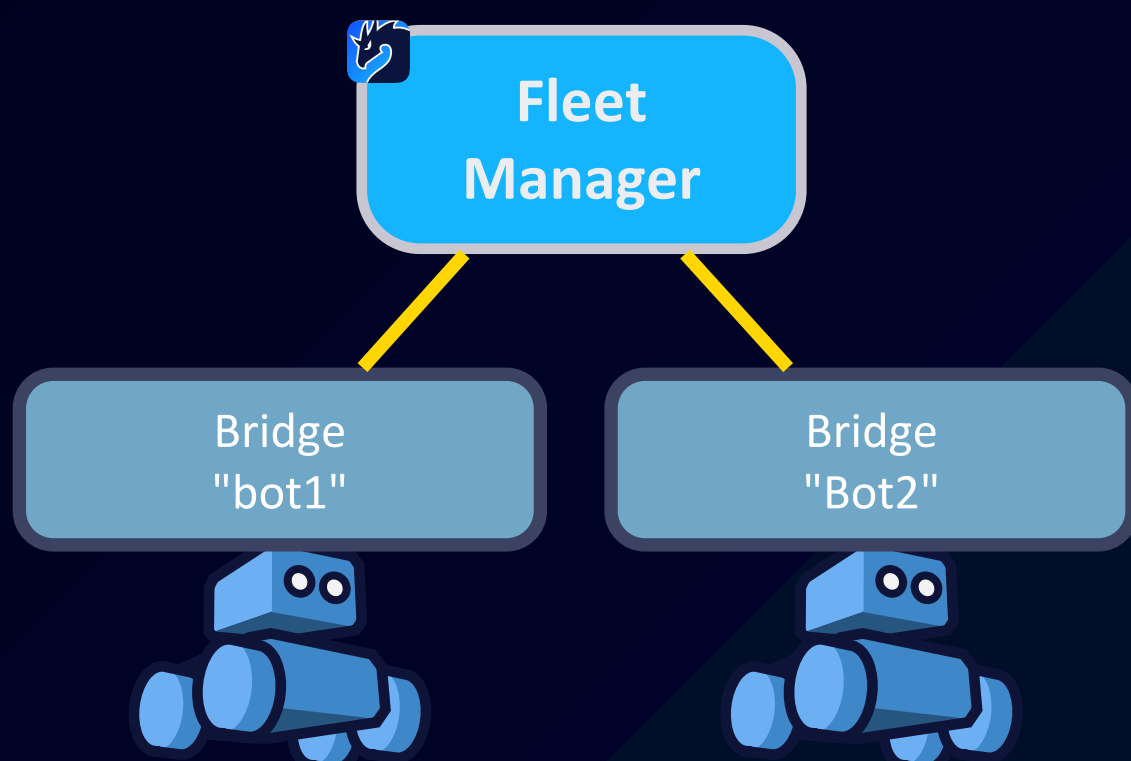


New Zenoh Bridge for ROS 2

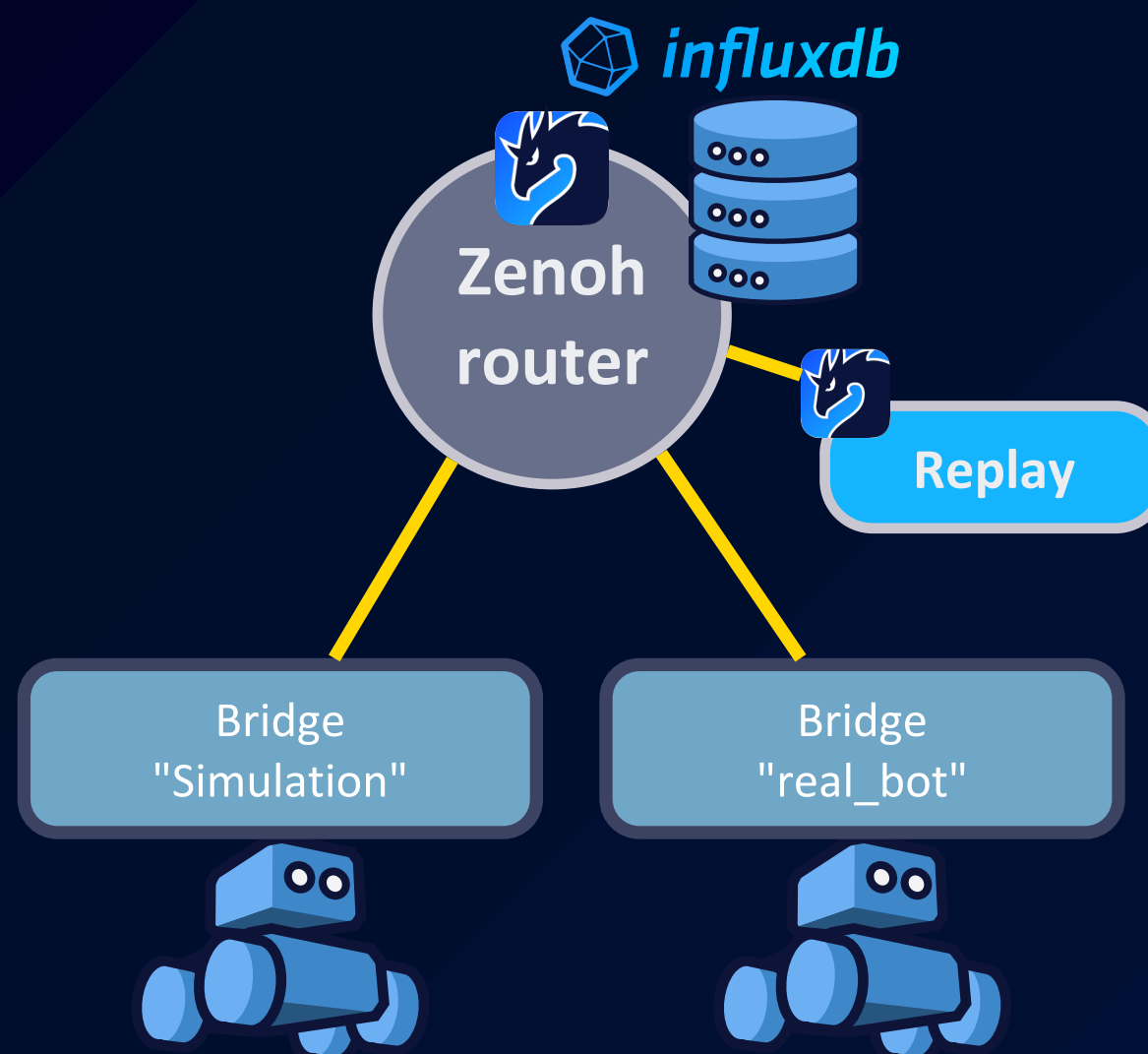
<https://github.com/eclipse-zenoh/zenoh-plugin-ros2dds>

- Better integration of the **ROS graph**
(all ROS topics/services/actions can be seen across bridges)
- Better support of **ROS toolings**
(ros2, rviz2...)
- Configuration of a **ROS namespace** on the bridge
(instead of on each ROS Nodes)
- Services and Action as **Zenoh Queryables**
- Even **more compact discovery information** between the bridges

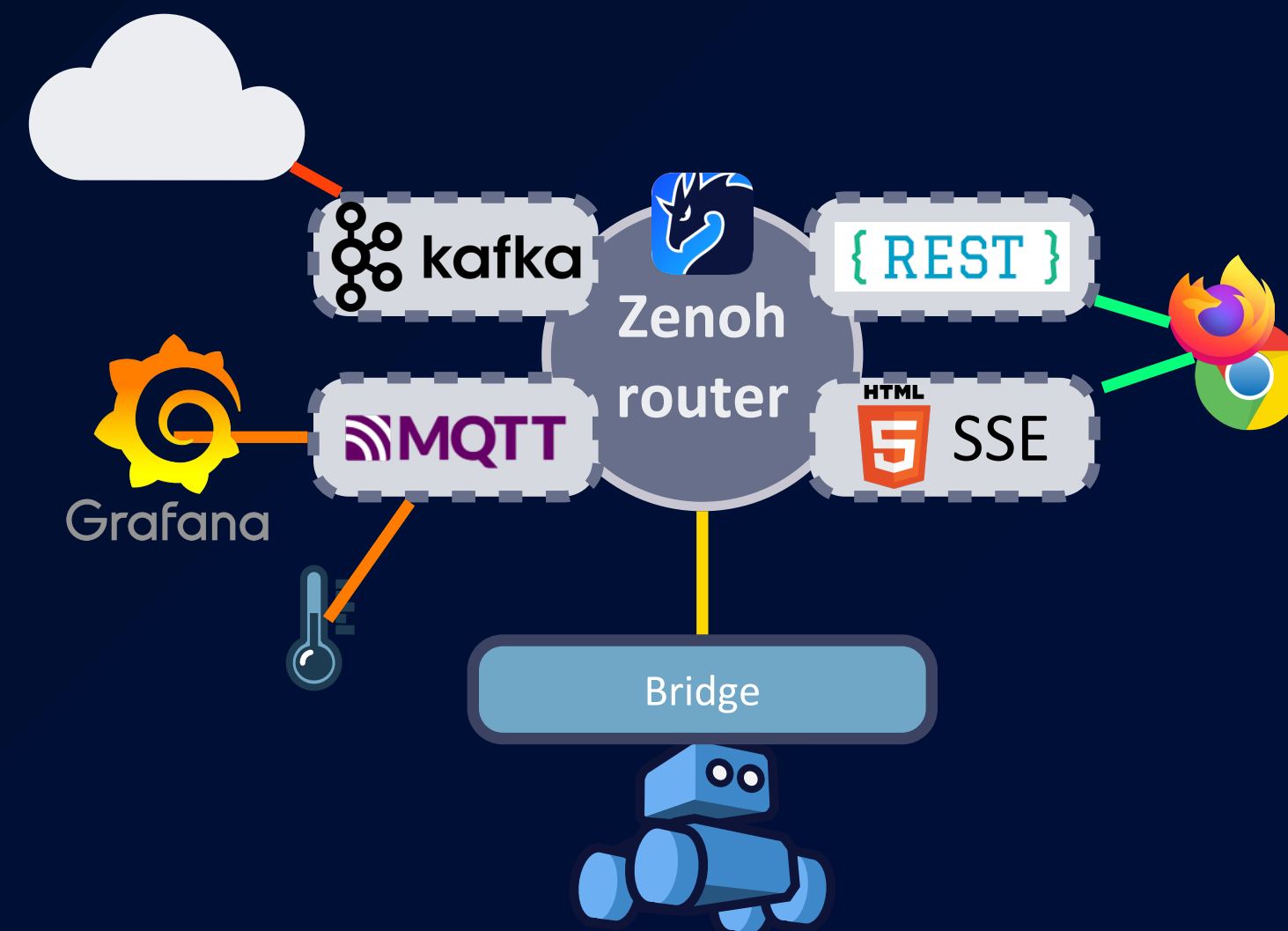
Use cases



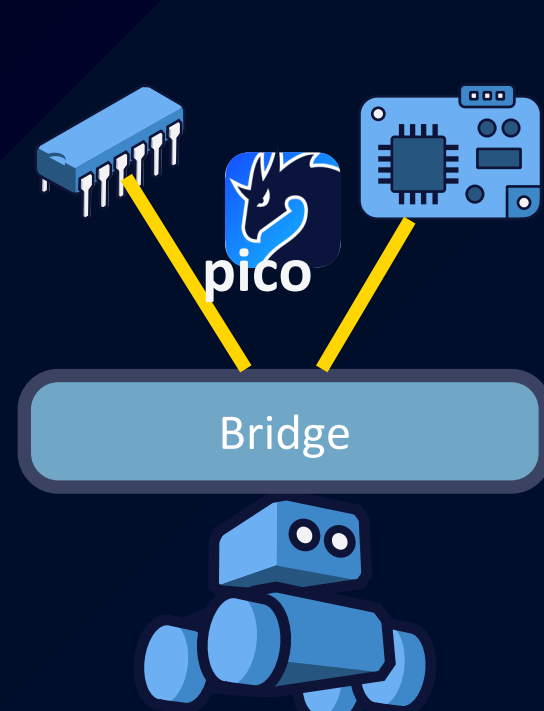
Fleet Management



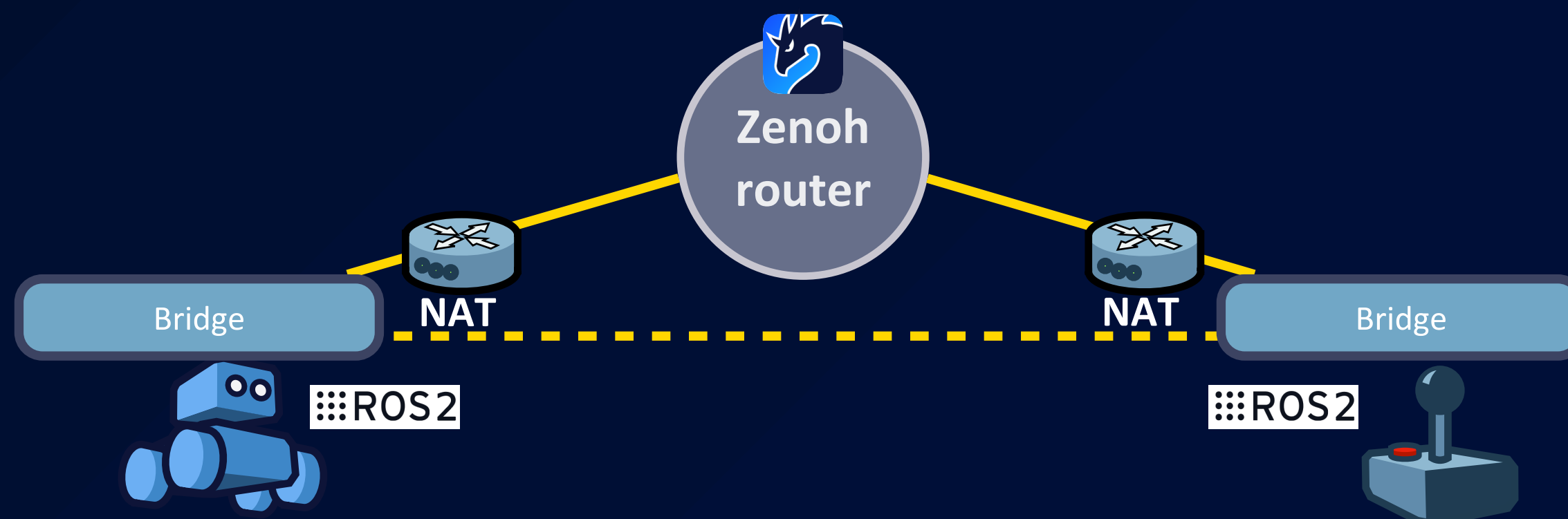
Record / Replay



Integrations



Pub/Sub with MCUs

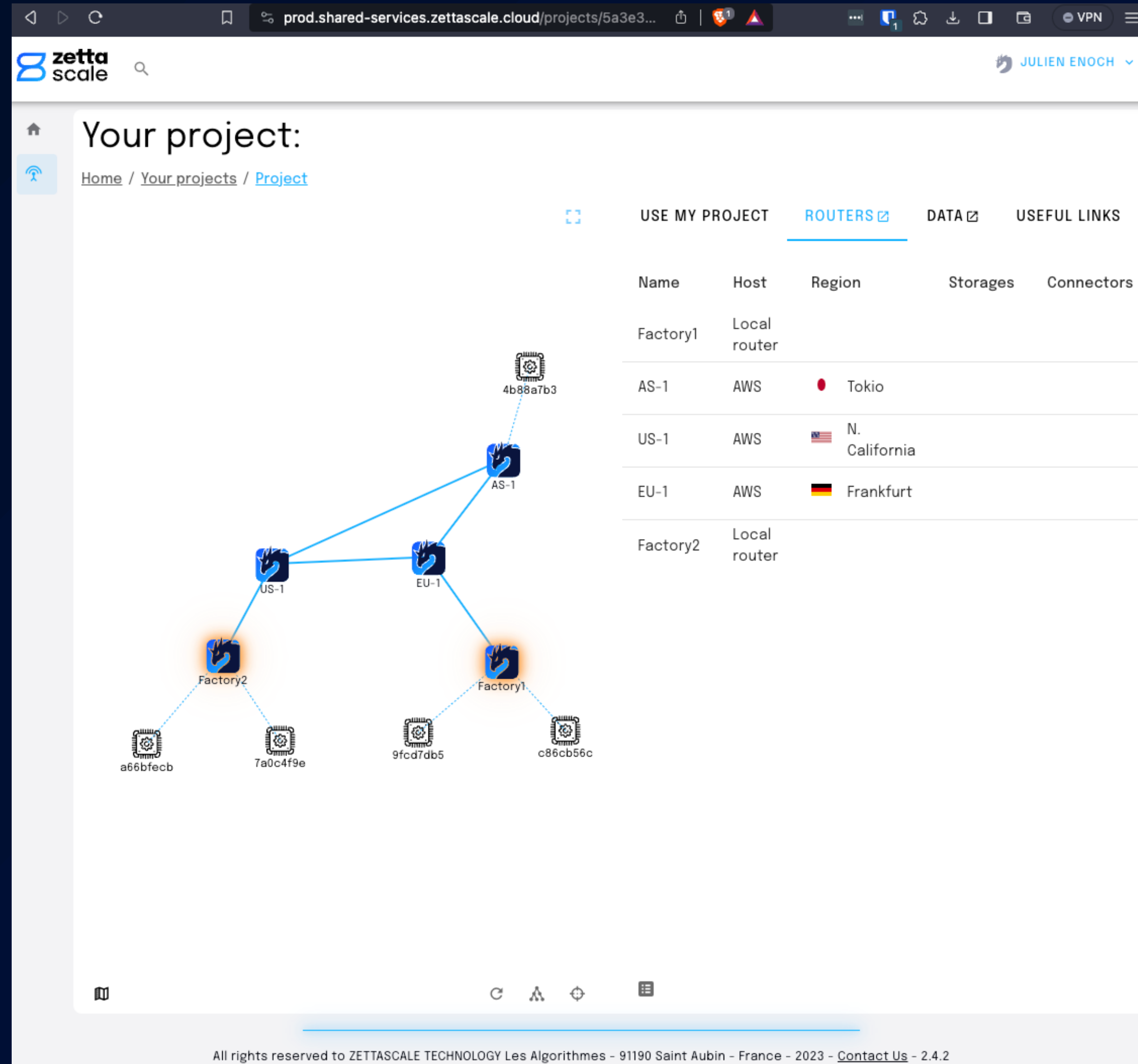


Cross-WAN bridging (routed or direct)

Demo

Zetta PaaS

- **Provision** Zenoh Routers in Cloud or on Premises
- **Generate** configurations and TLS certificates
- **Monitor** the full Zenoh infrastructure and connected peers/clients

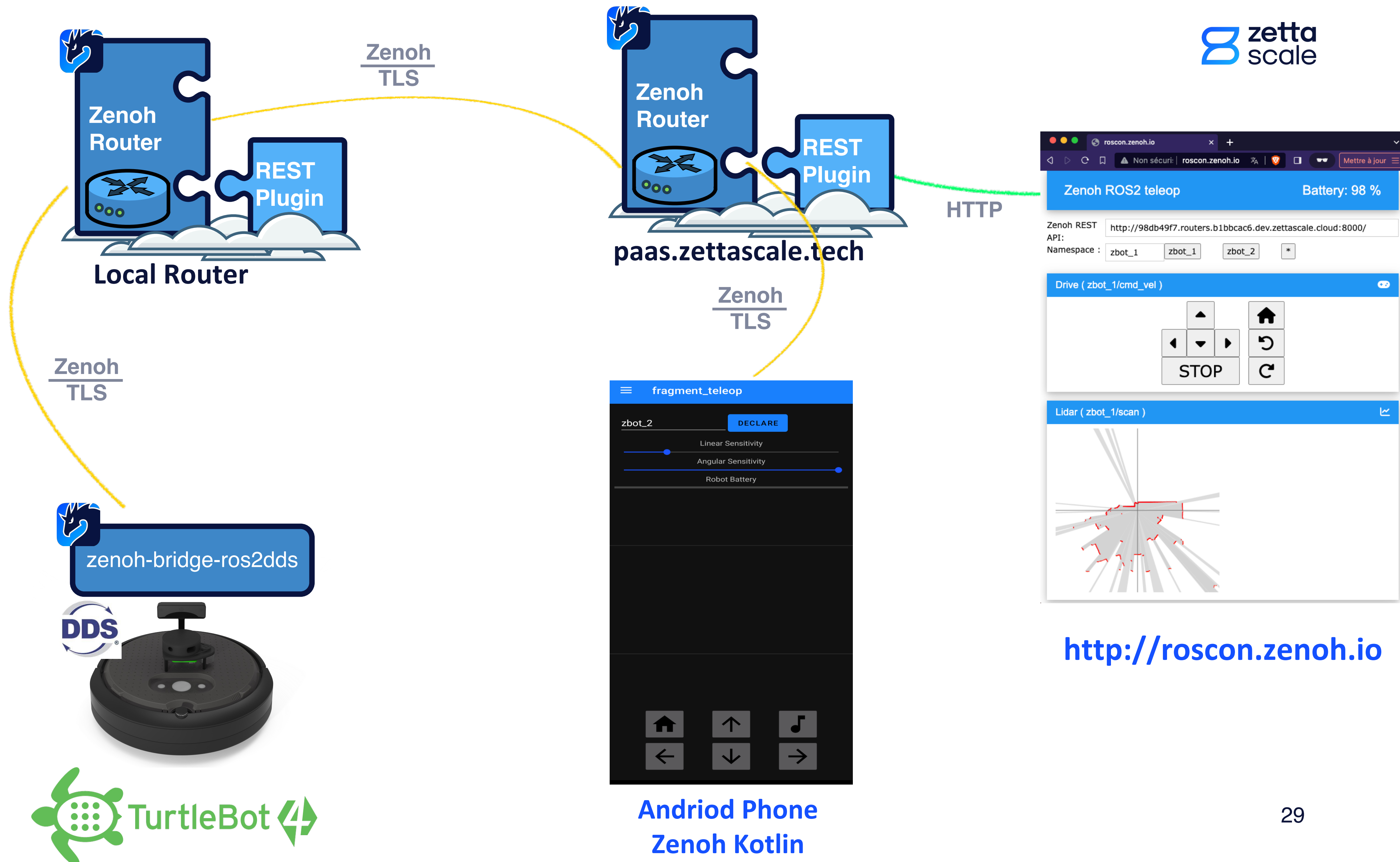


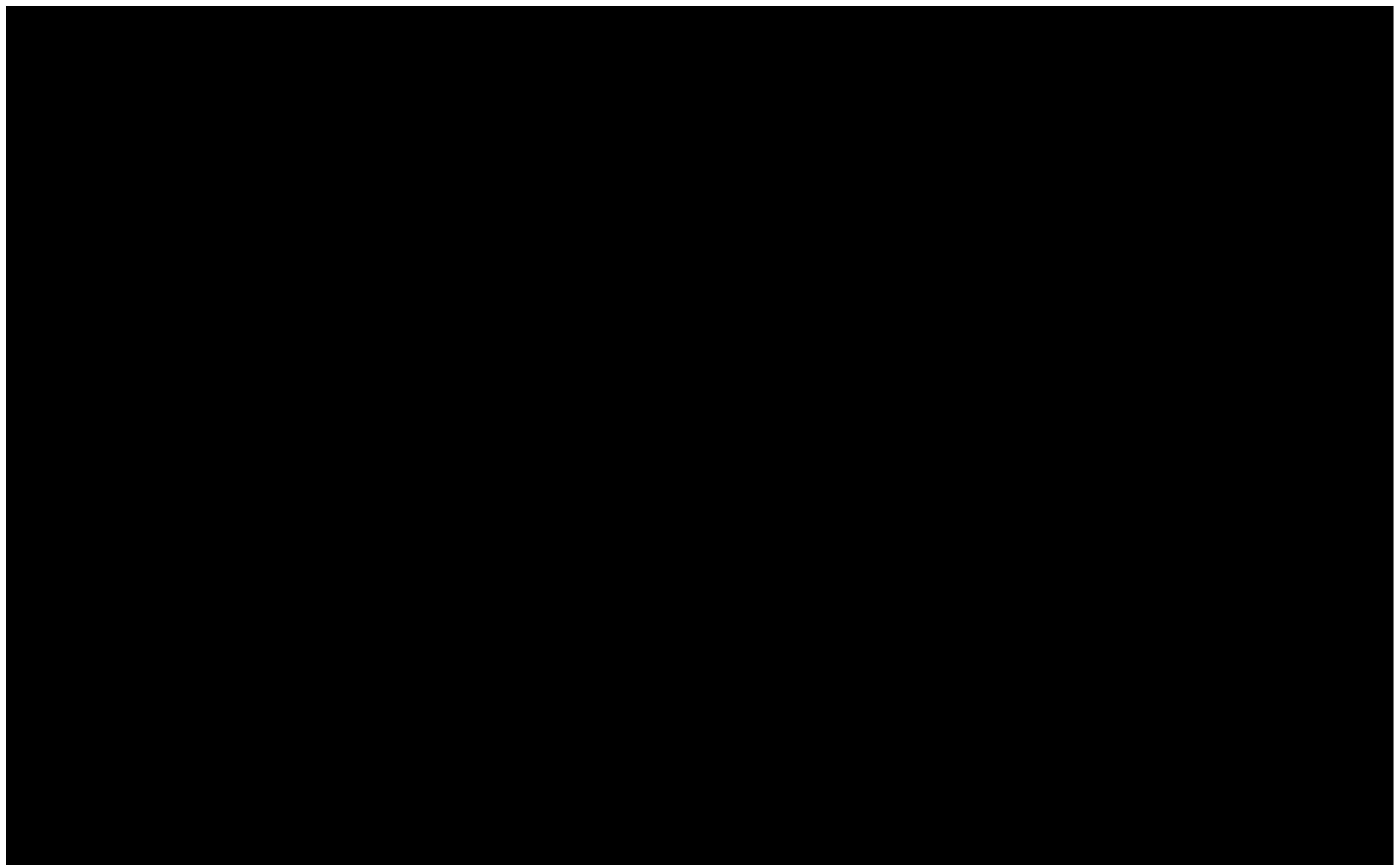
The screenshot displays the ZettaScale web interface for a project. The main section shows a network diagram with nodes labeled AS-1, US-1, EU-1, Factory1, and Factory2, connected by lines. The diagram includes IP addresses for each node: 4b88a7b3, a66bfecb, 7a0c4f9e, 9fcd7db5, and c86cb56c.

On the right side, there is a table titled "USE MY PROJECT" with tabs for "ROUTERS", "DATA", and "USEFUL LINKS". The "ROUTERS" tab is active, showing a list of routers with columns for Name, Host, Region, Storages, and Connectors.

| Name | Host | Region | Storages | Connectors |
|----------|--------------|---------------|----------|------------|
| Factory1 | Local router | | | |
| AS-1 | AWS | Tokio | | |
| US-1 | AWS | N. California | | |
| EU-1 | AWS | Frankfurt | | |
| Factory2 | Local router | | | |

At the bottom of the interface, there is a footer with the text: "All rights reserved to ZETTASCALE TECHNOLOGY Les Algorithmes - 91190 Saint Aubin - France - 2023 - [Contact Us](#) - 2.4.2"

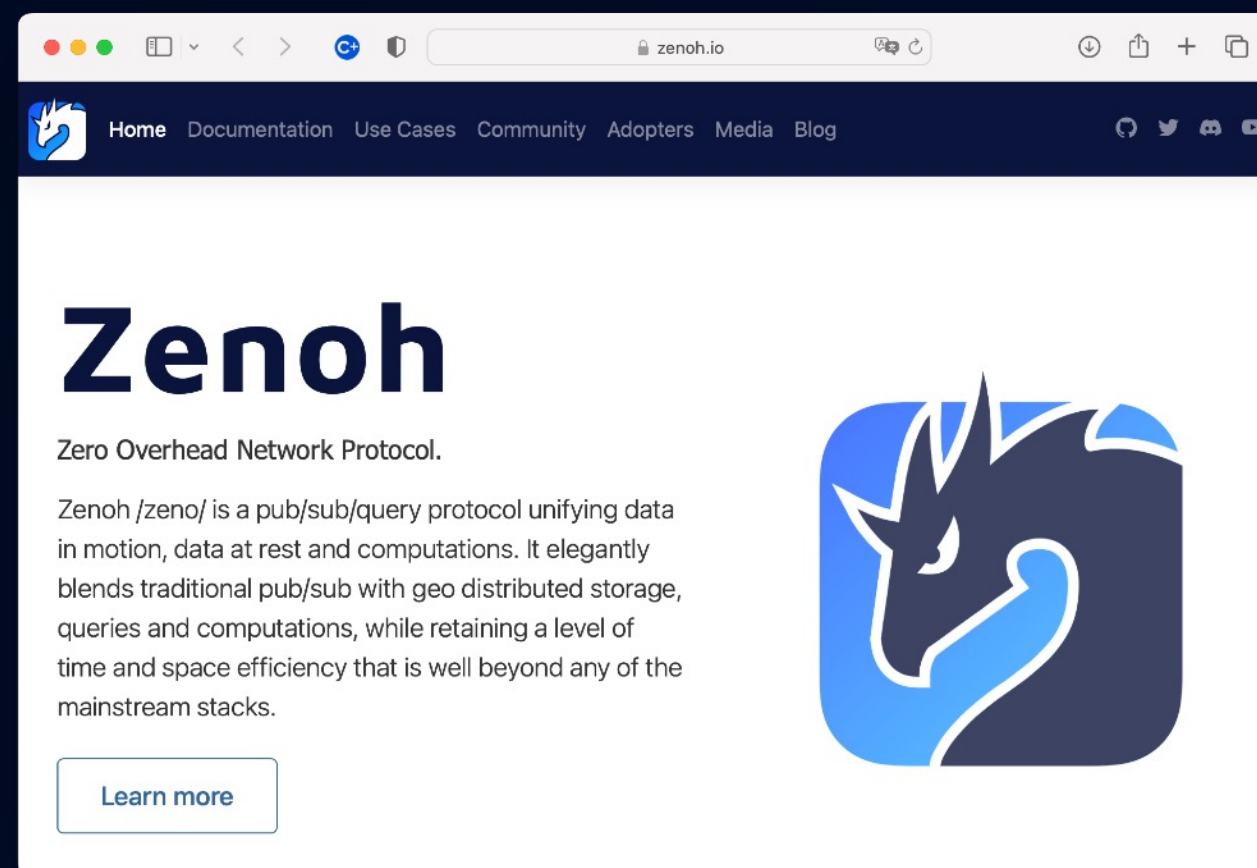




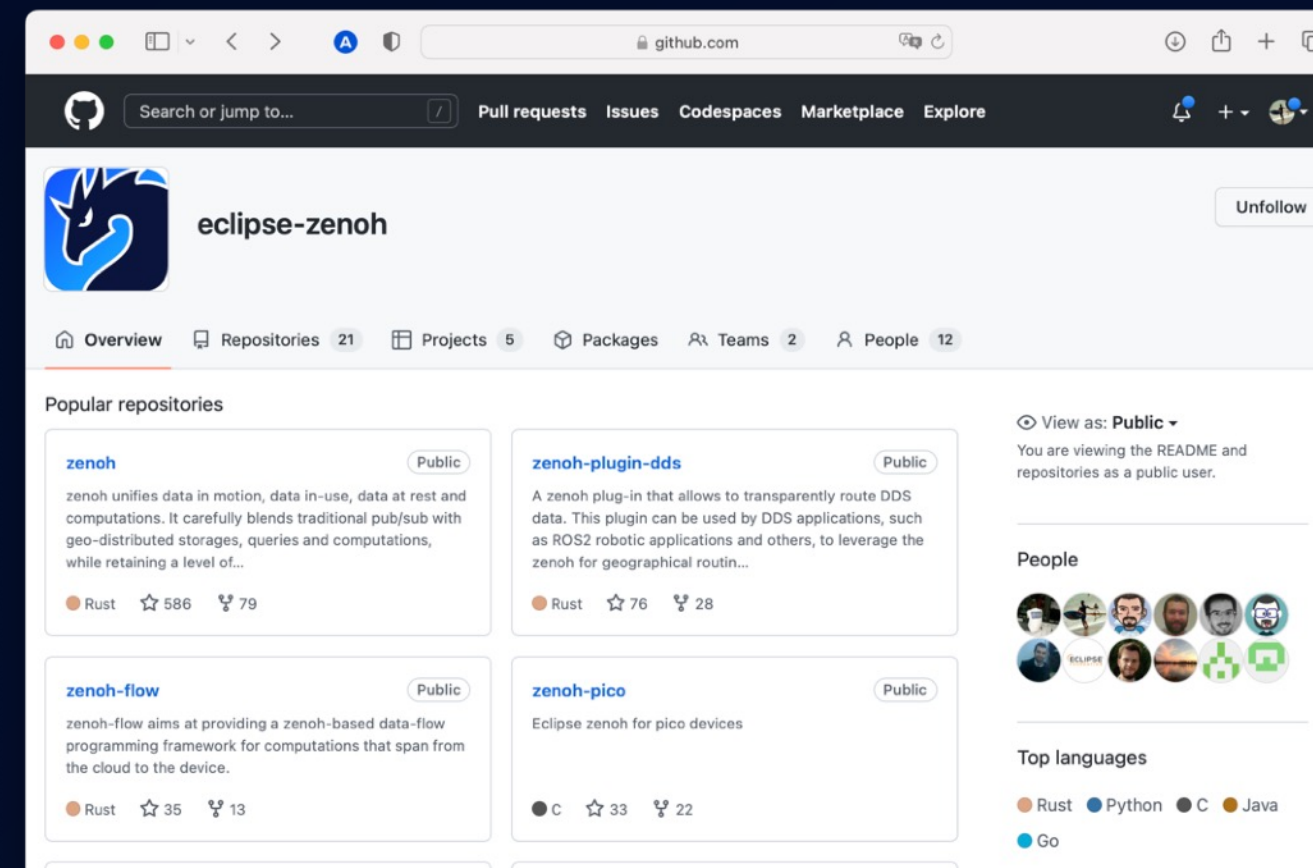
Don't forget to visit
Zenoh's website...

...Zenoh's GitHub ...

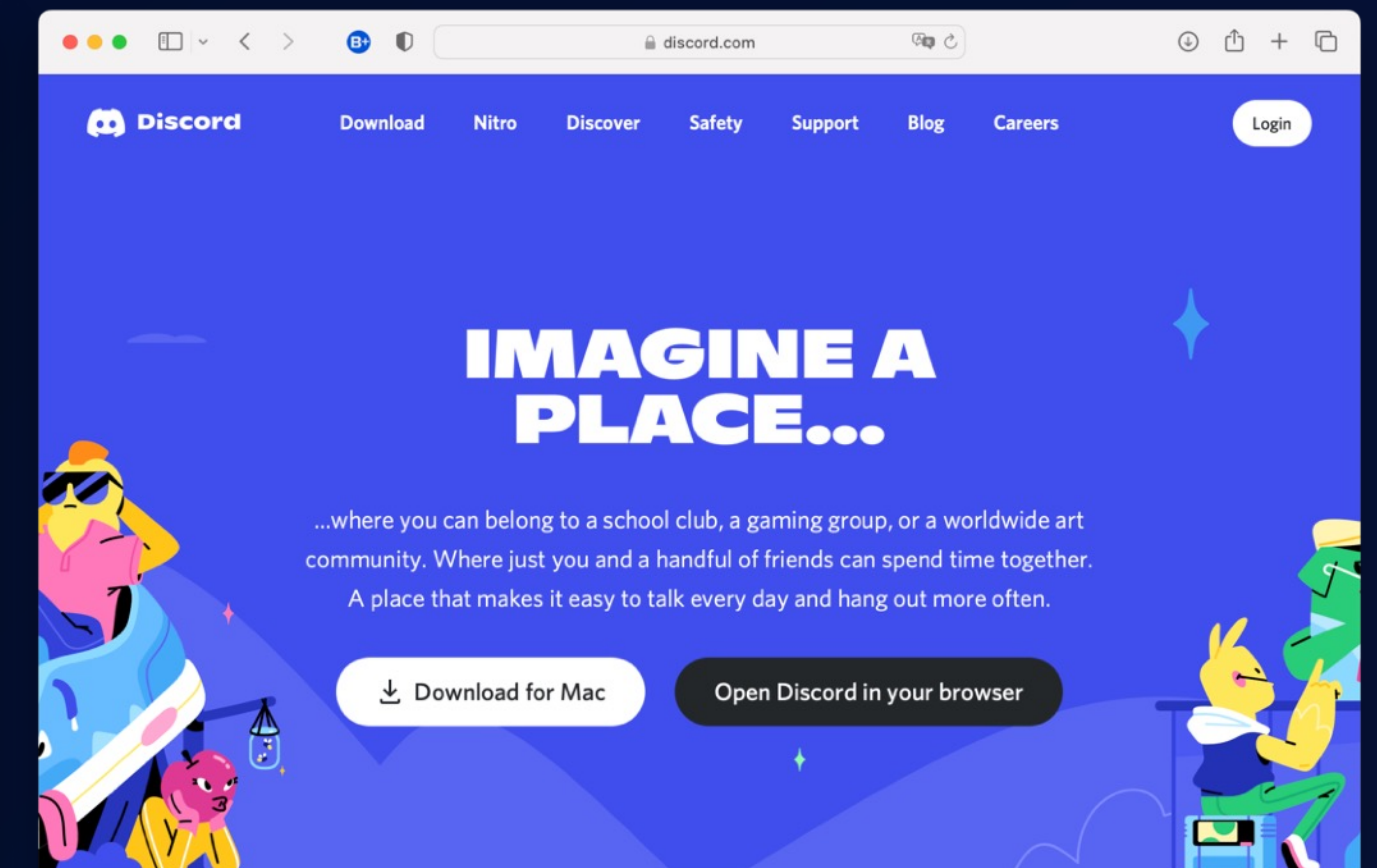
...and to join Zenoh's
Discord server!



<https://zenoh.io/>



[https://github.com/
eclipse-zenoh/zenoh](https://github.com/eclipse-zenoh/zenoh)



[https://discord.gg/
2GJ958VuHs](https://discord.gg/2GJ958VuHs)

Thank You

Patience, persistence and
perspiration make an unbeatable
combination for success.

