zetta scale

rmw_zenoh

An alternative open-source middleware for ROS 2

Phani Gangula

Senior Solutions Architect

phani@zettascale.tech

Background

Zenoh: The RMW Alternate

Zenoh has been selected as the first non-DDS protocol to be natively supported in ROS2

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

The full report is available here

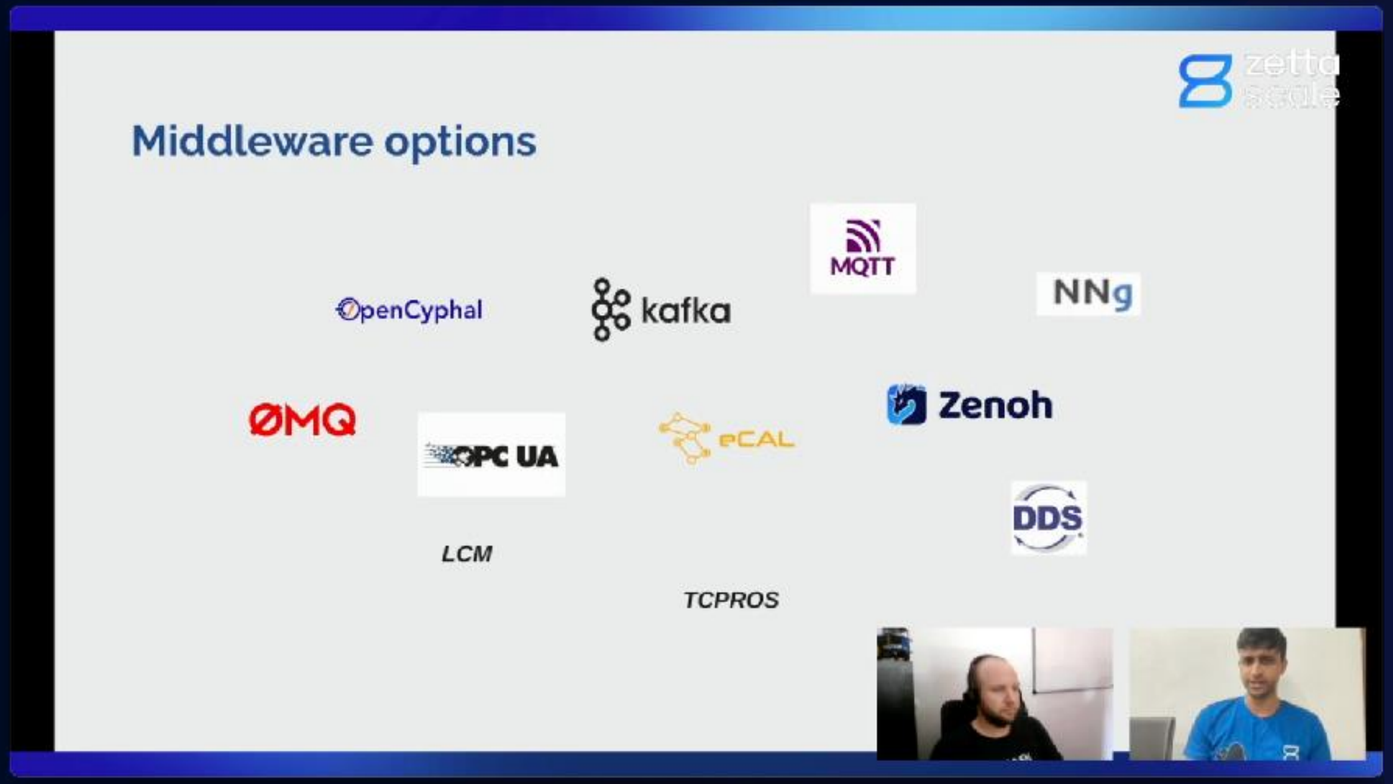
2023-09 ROS 2 RMW alternate

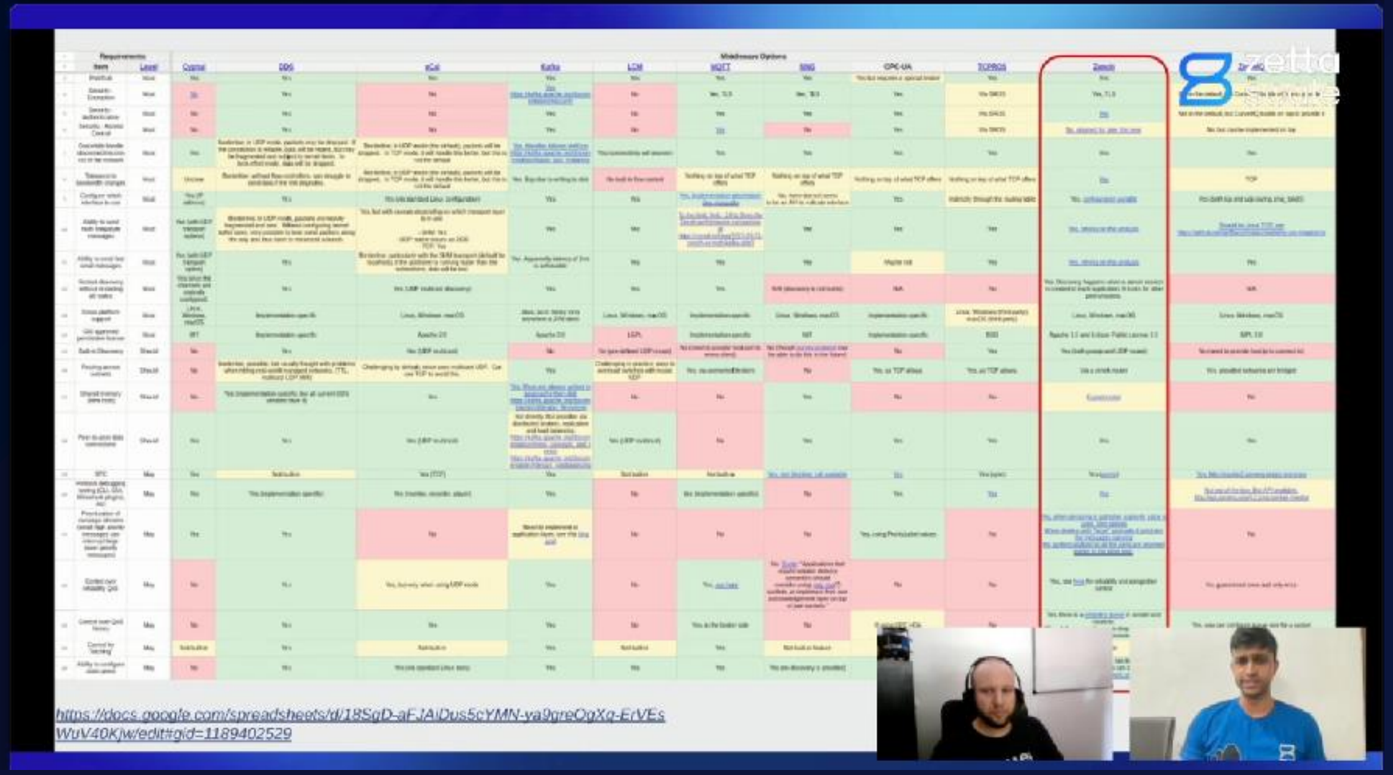
Conclusion

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.

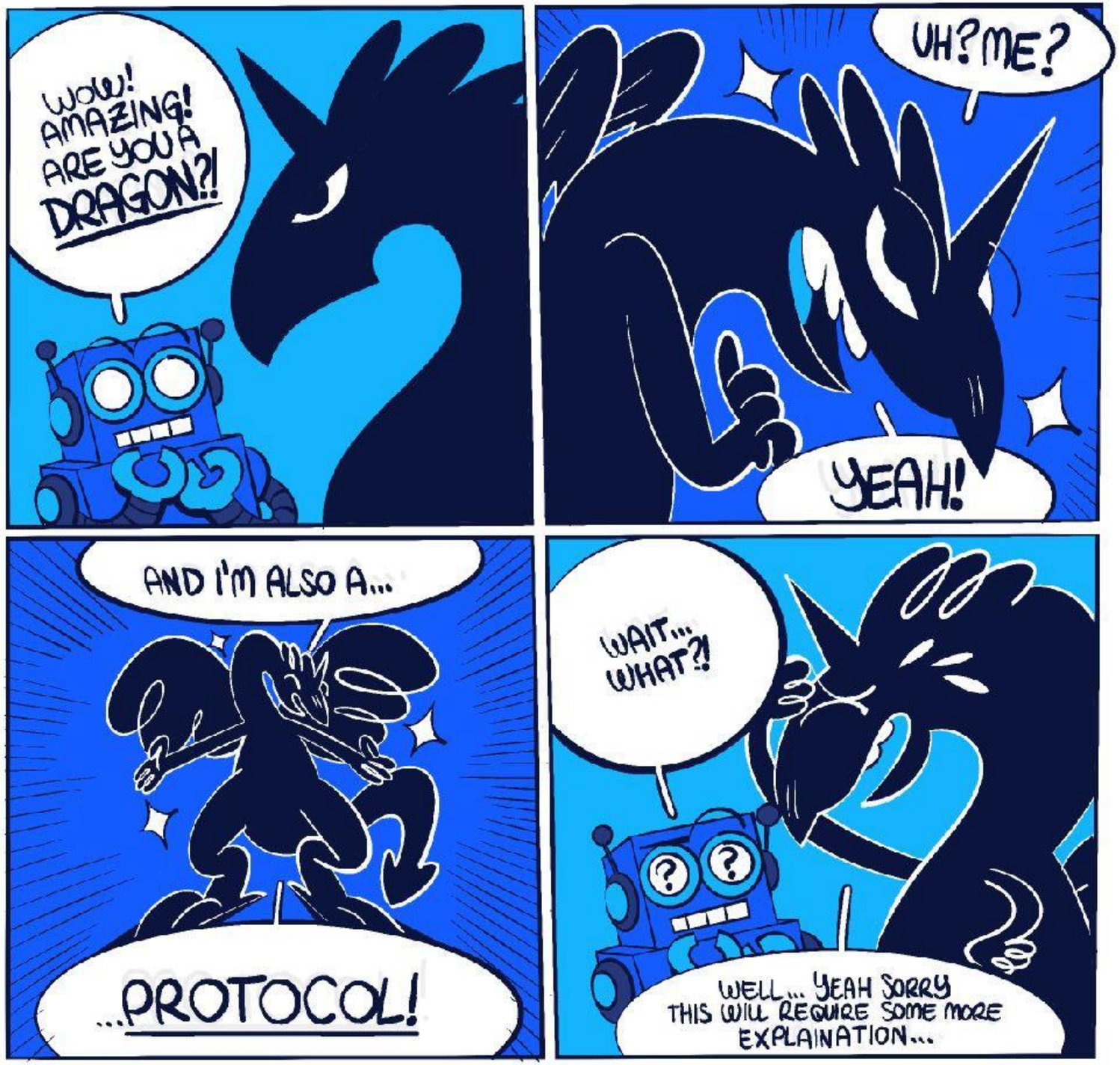
Best Fit

Zenoh stands-out as the protocol that satisfies essentially all requirements identified by the survey





Zenoh


WOW! AMAZING! ARE YOU A DRAGON?!
UH? ME?
YEAH!
AND I'M ALSO A...
...PROTOCOL!
WAIT... WHAT?!
WELL... YEAH SORRY THIS WILL REQUIRE SOME MORE EXPLAINATION...

zetta scale

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

Built-in support for zero-copy and shared memory

Dragons teach us that if we want to climb high we have to do it against the wind.

Runs Everywhere

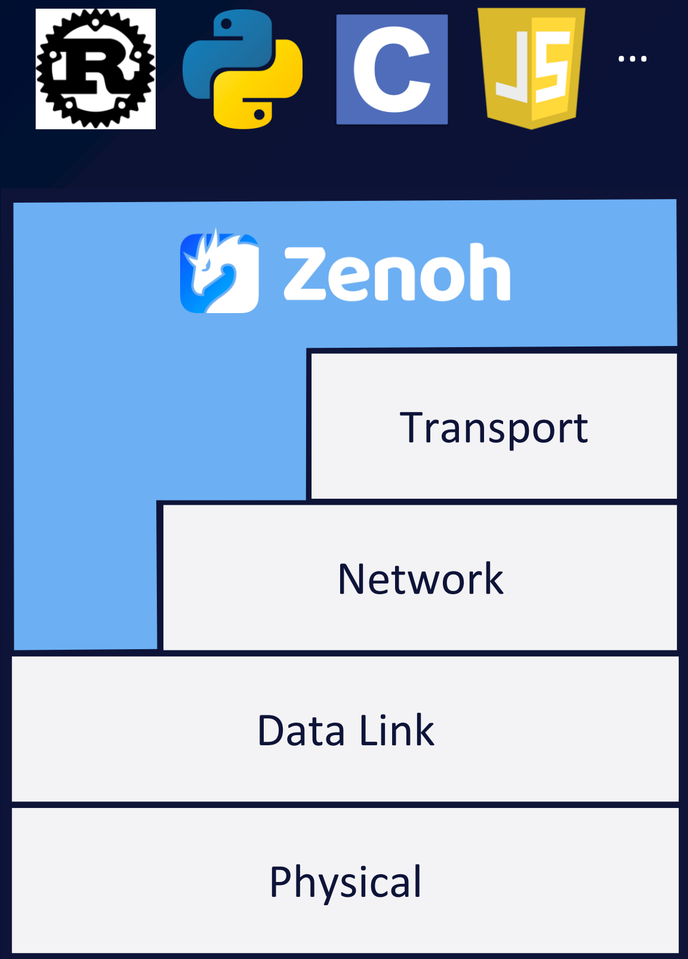
Written in Rust for security, safety and performance

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

Built-in support Shared Memory and Zero Copy

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

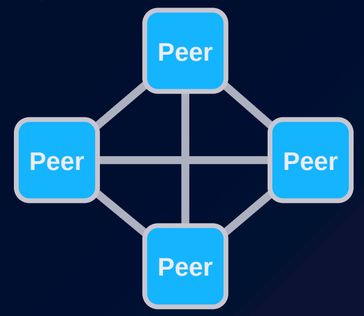


Any Topology

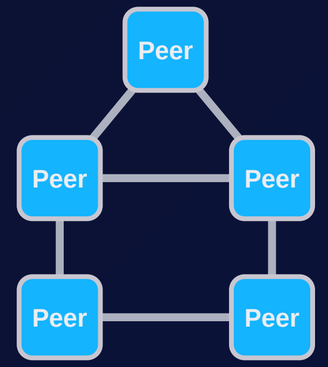
Peer-to-peer

Clique and mesh topologies

Clique



Mesh



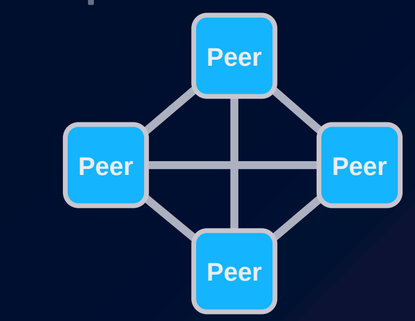
Any Topology

Peer-to-peer

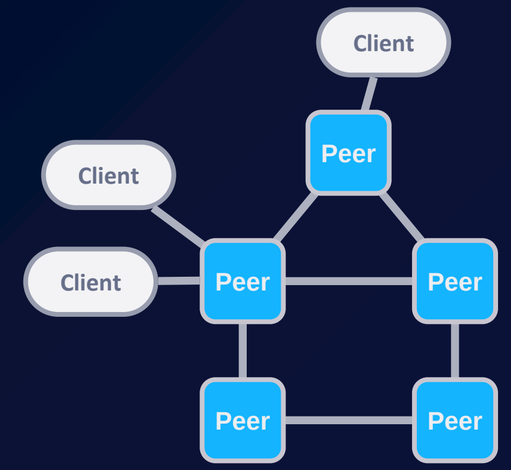
Clique and mesh topologies

Brokered

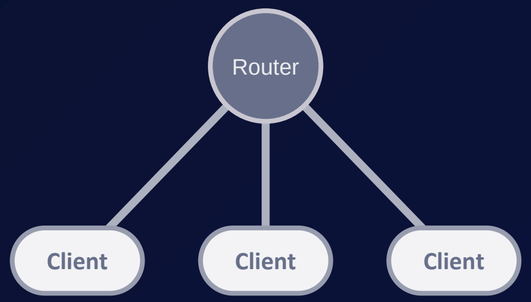
Clients communicate through a router or a peer



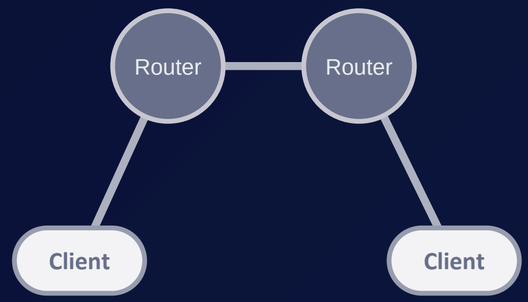
Clique



Mesh



Brokered



Routed

Any Topology

Peer-to-peer

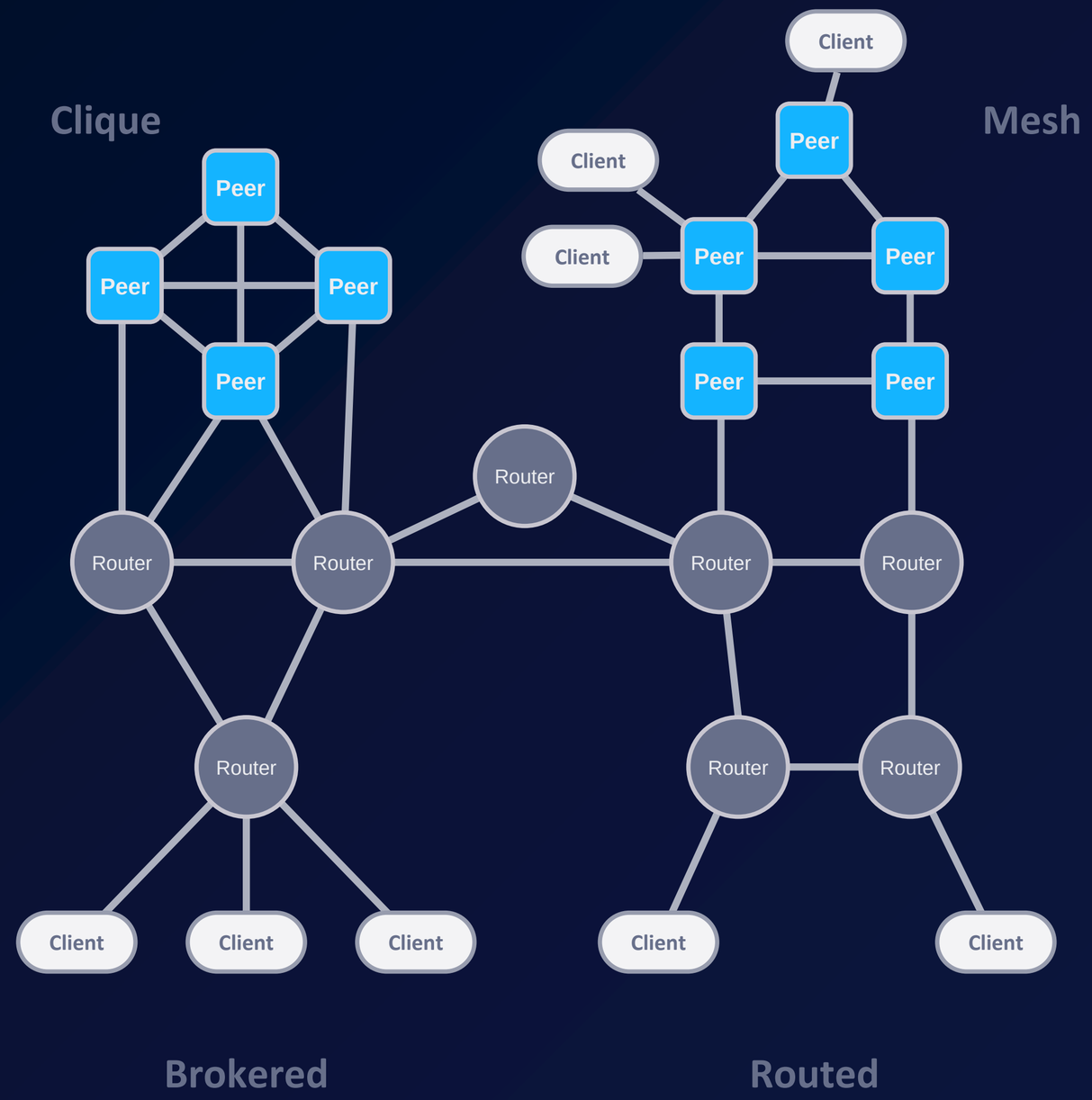
Clique and mesh topologies

Brokered

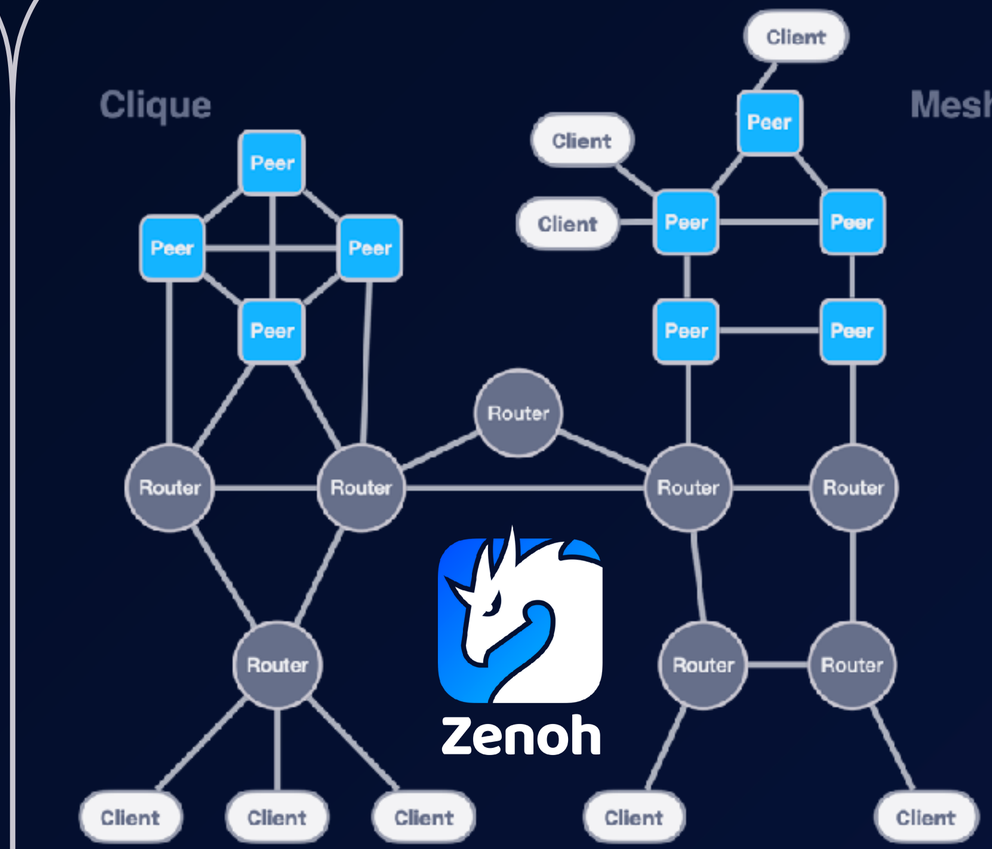
Clients communicate through a router or a peer

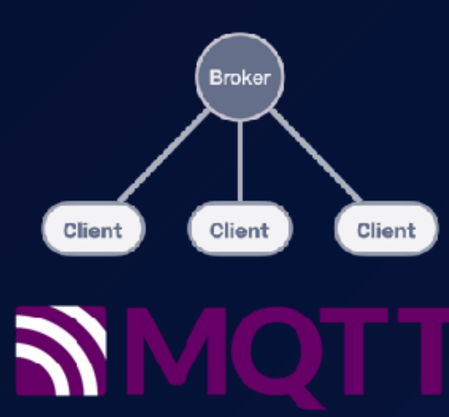
Routed

Routers forward data to and from peers and clients



Topology in Perspective



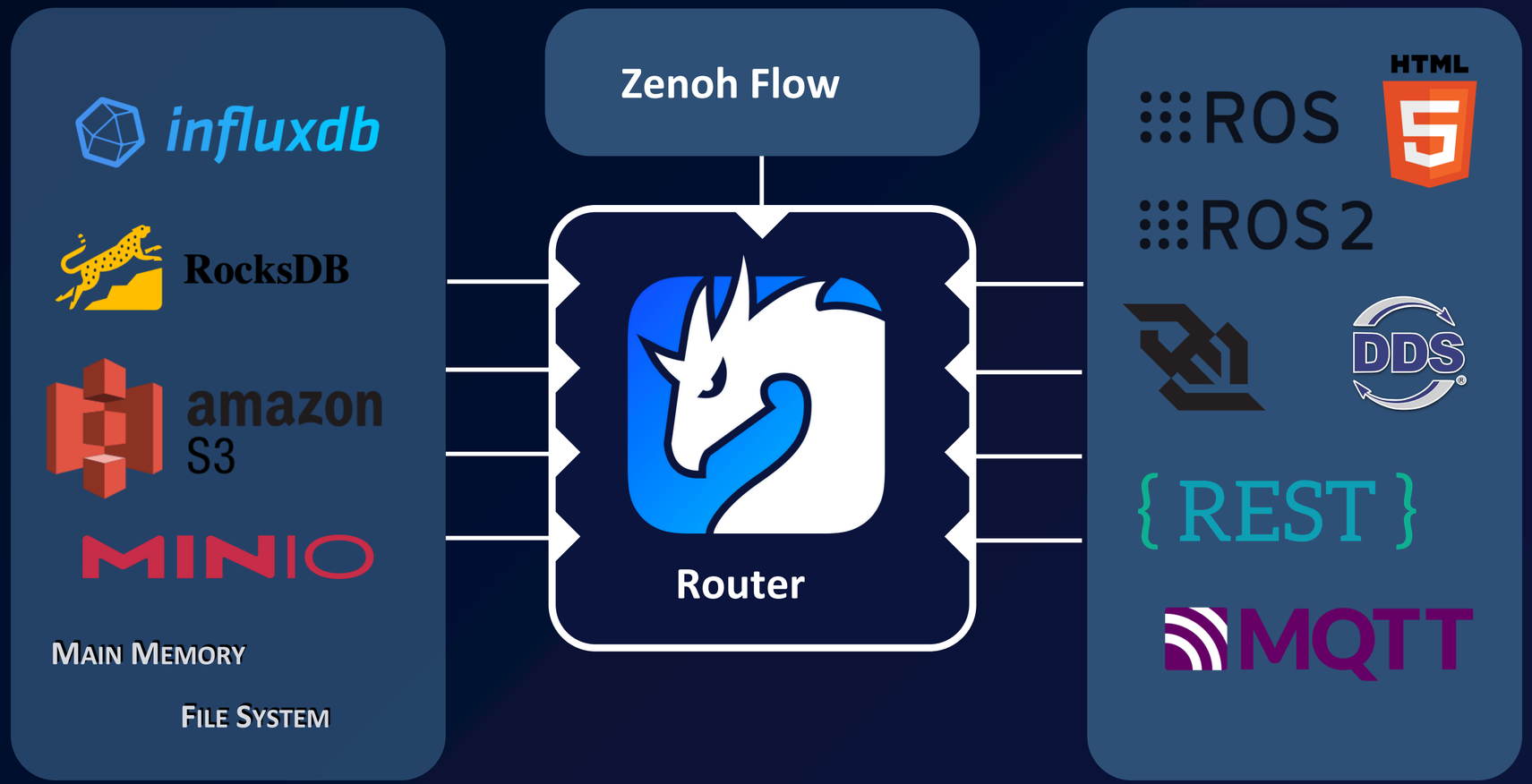


Plug-Ins

Storage Plugins

Runtime Plugins

Protocol Plugins



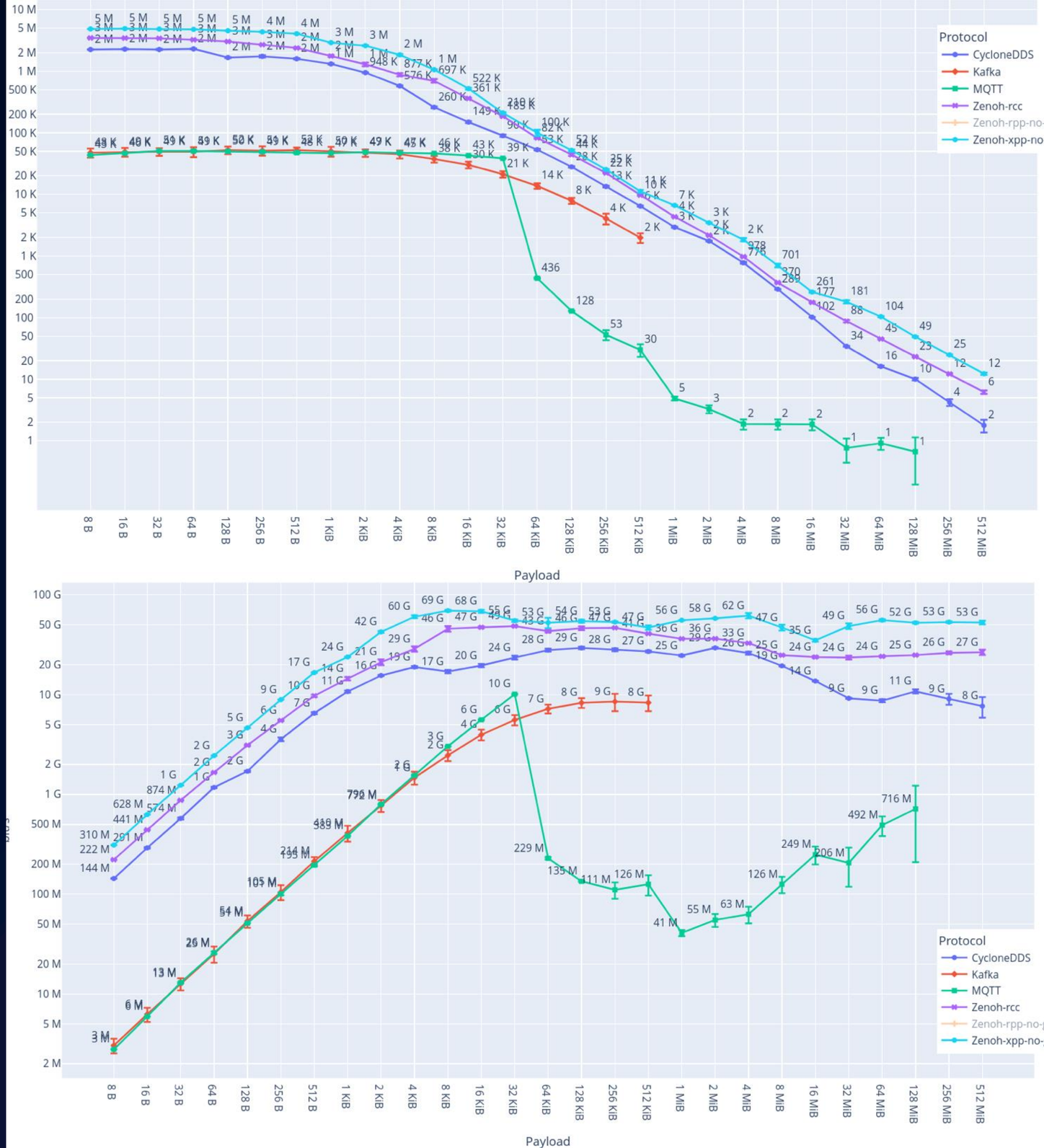
Zenoh vs DDS, MQTT & Kafka

Zenoh can deliver at peak performance of ~70Gbps at 8Kb payload:

- 3.3x higher than DDS
- 23x higher than Kafka
- 35x than MQTT (higher for larger payload)

Zenoh's latency 10us, 6us with ultra-low latency support

- 25us for MQTT
- 75us for Kafka
- 8us DDS

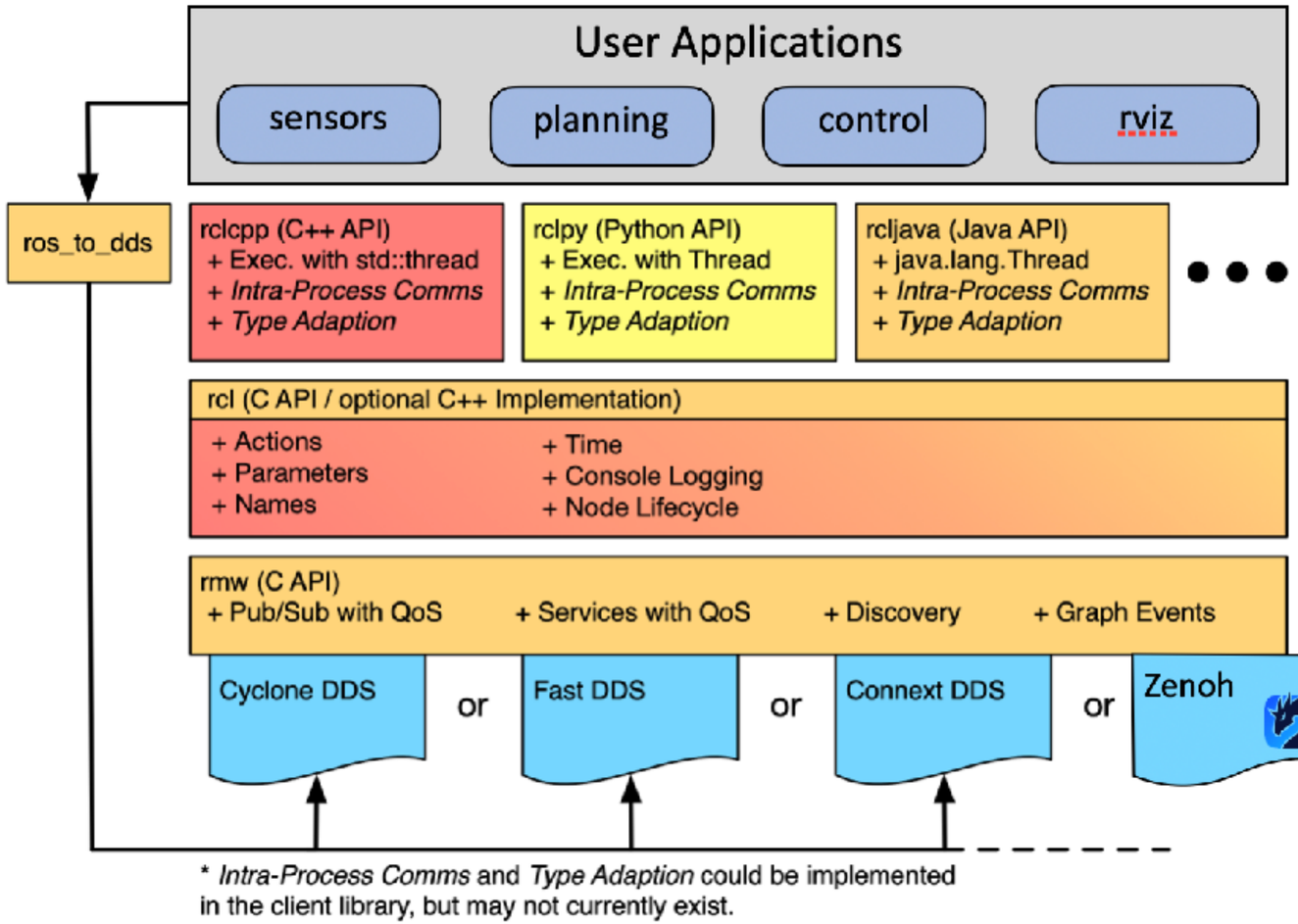


zetta scale

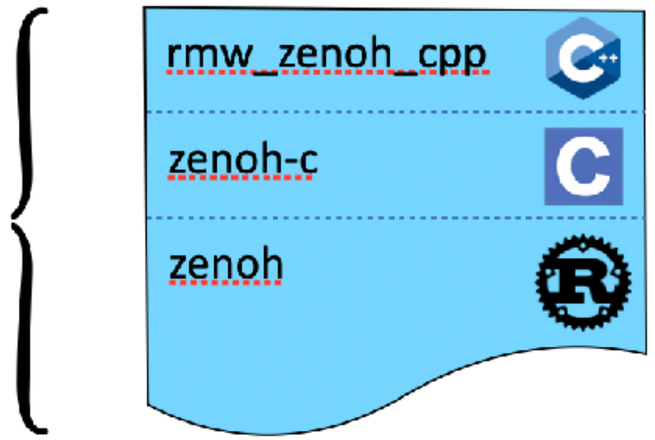
rmw_zenoh

An RMW implementation based on Zenoh

ROS 2 — Modular Architecture

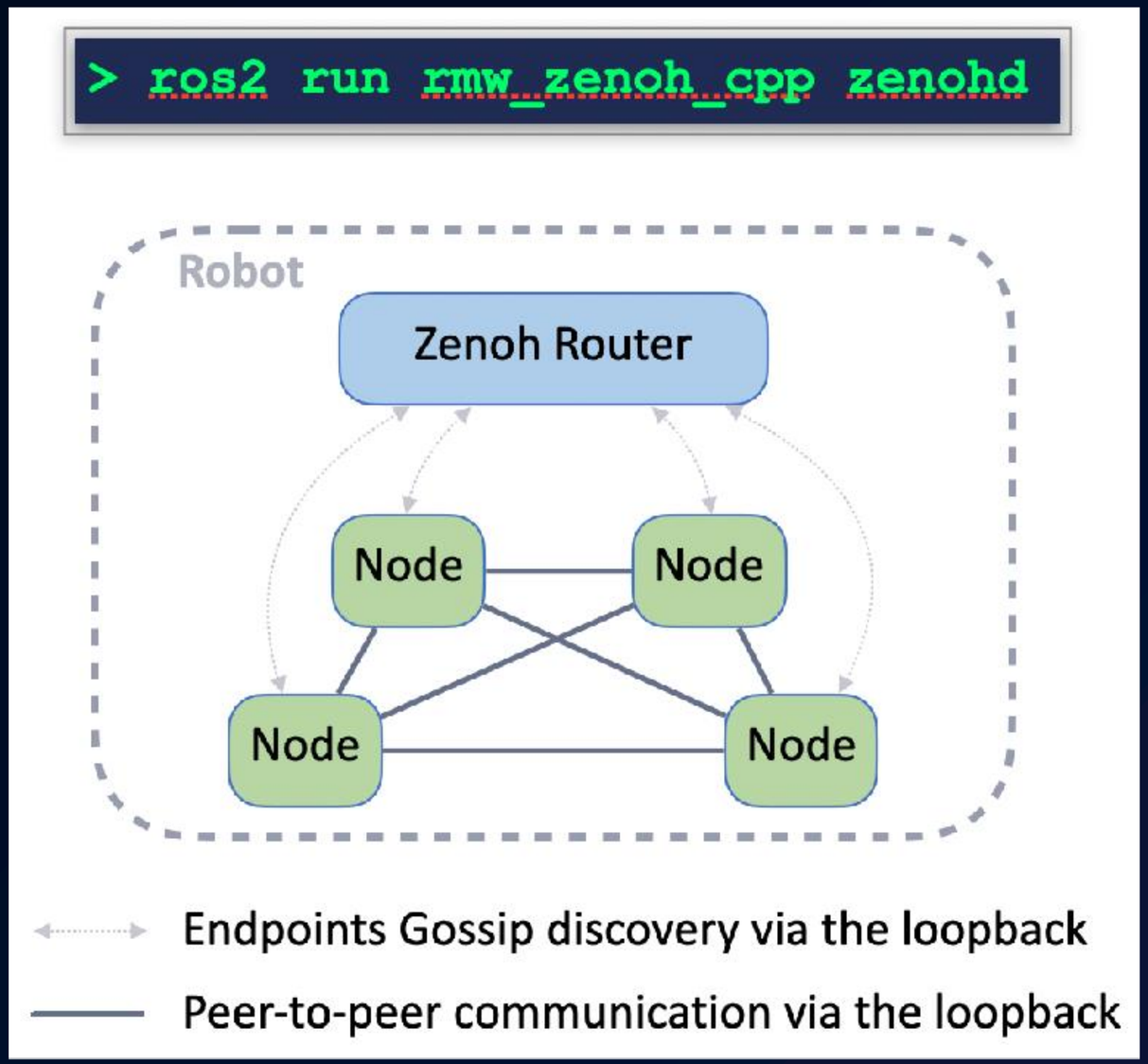


https://github.com/ros2/rmw_zenoh



https://docs.ros.org/en/rolling/Concepts/Advanced/About-Internal-Interfaces.html#internal-api-architecture-overview

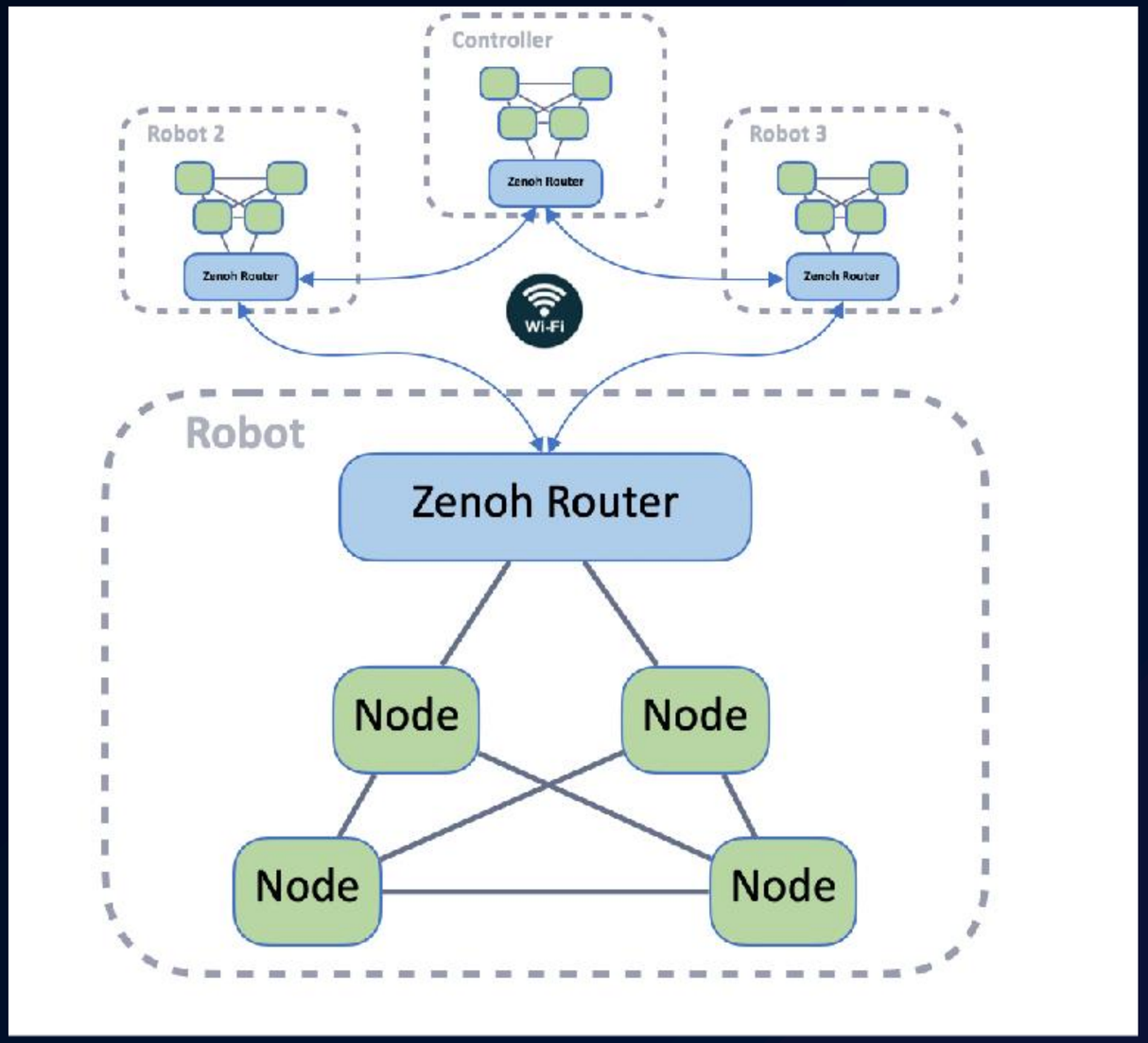
Zenoh Router



Local Communication

- The router listen on tcp/0.0.0.0:7447
- Each Node connect to the router on tcp/127.0.0.1:7447
- The router acts as a broker for endpoints discovery
- Nodes establish peer-to-peer connections via 127.0.0.1
- Select the transport for local communication independently than that for R2X
- Leverage SHM Optimisation (coming up)

Zenoh Router

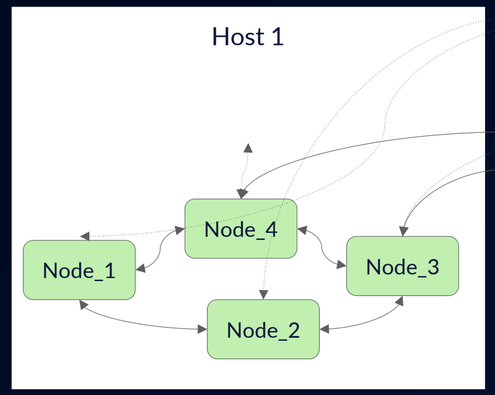


External Communications (R2X):

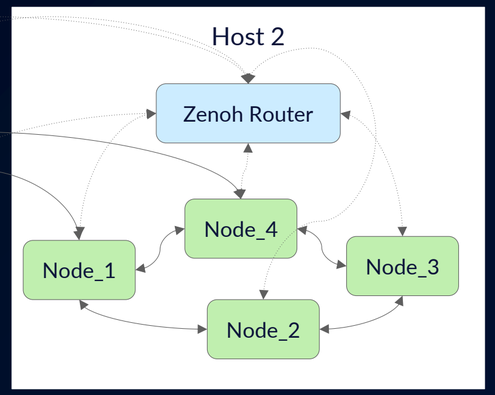
- May use different transport for R2X, i.e. TCP, TLS, QUIC, etc.
- Transparent Batching and compression
- Access Control and Downsampling
- Smaller surface of attack

Other configurable topologies

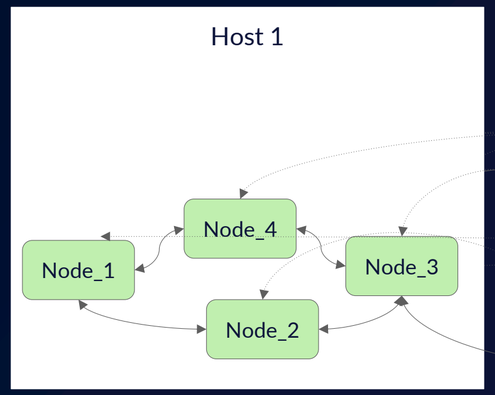
https://github.com/ros2/rmw_zenoh



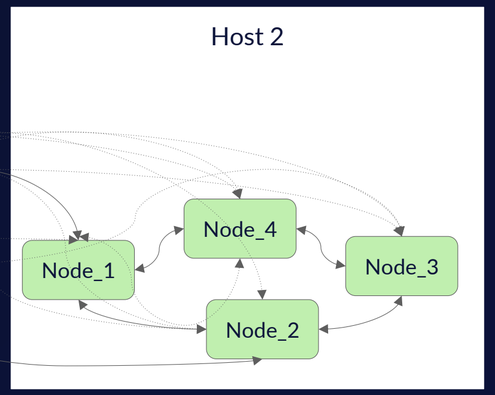
TCP
Discovery



TCP
Transport



UDP
multicast
Discovery

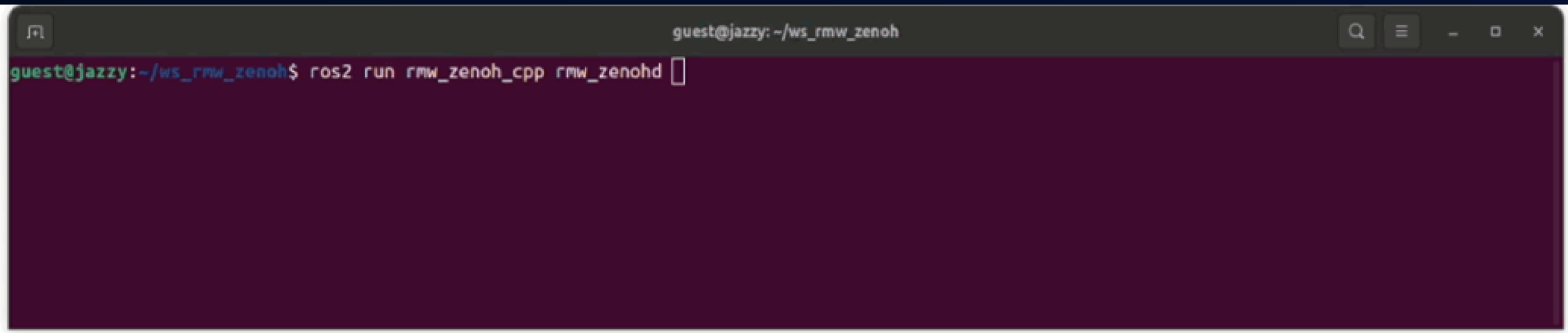


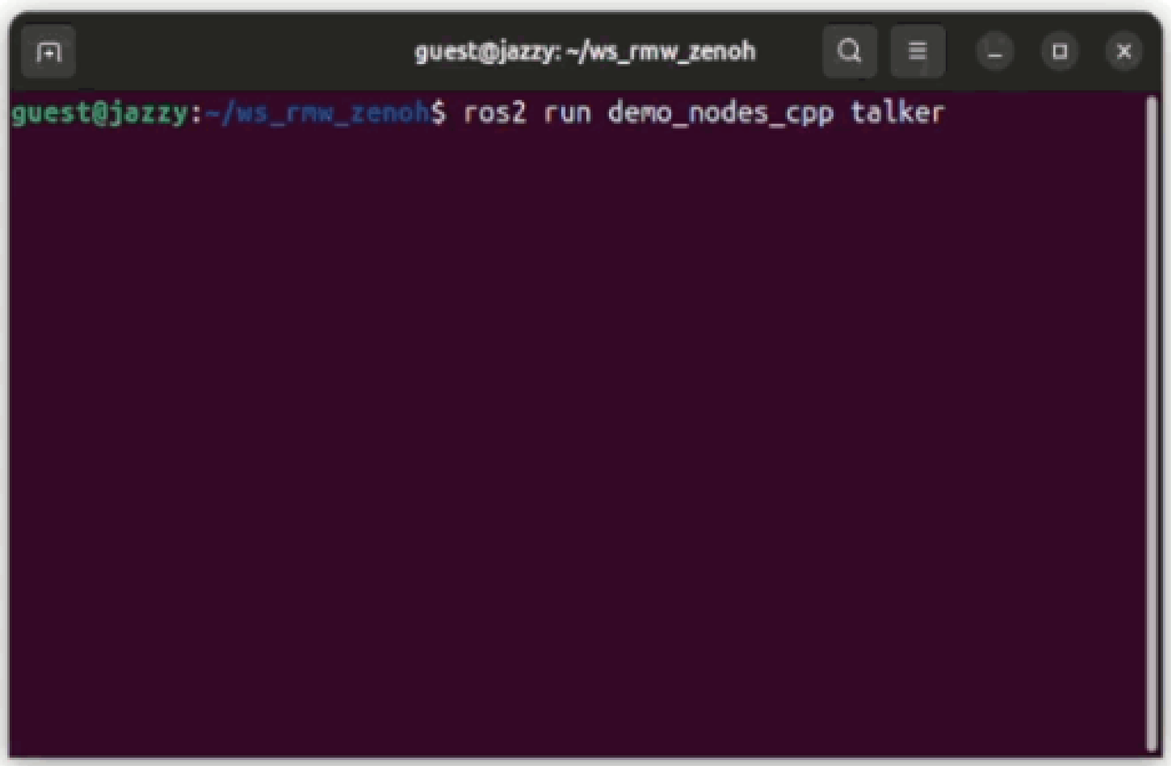
TCP
Transport

zetta scale

Router for discovery,
but peer-to-peer communications

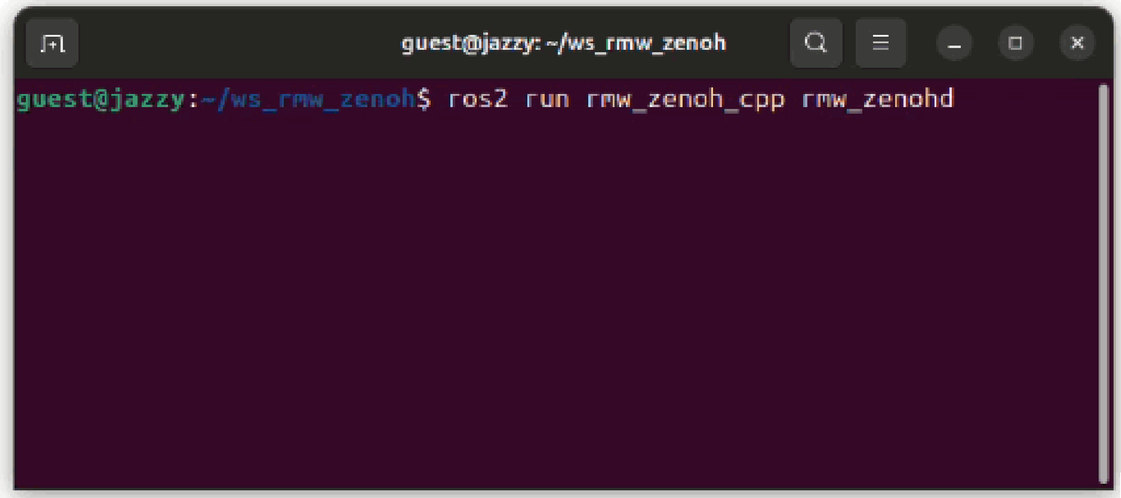
If router crashes, peer-to-peer communications remain



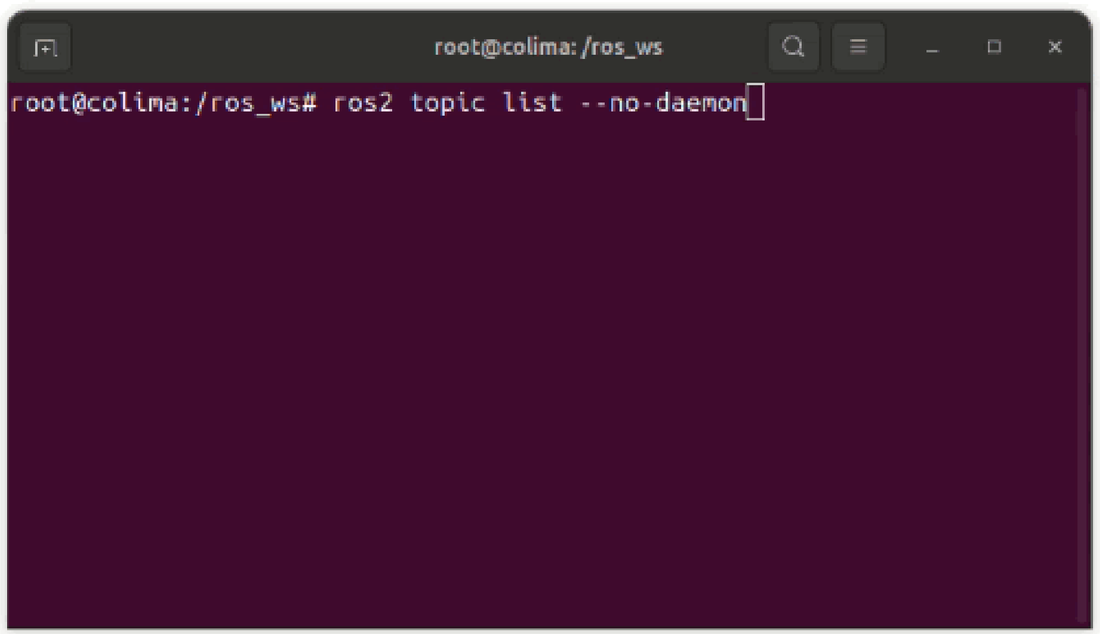


Discovery is robust

10 Nodes, 1000 Topics discovered in less than 1s over WiFi

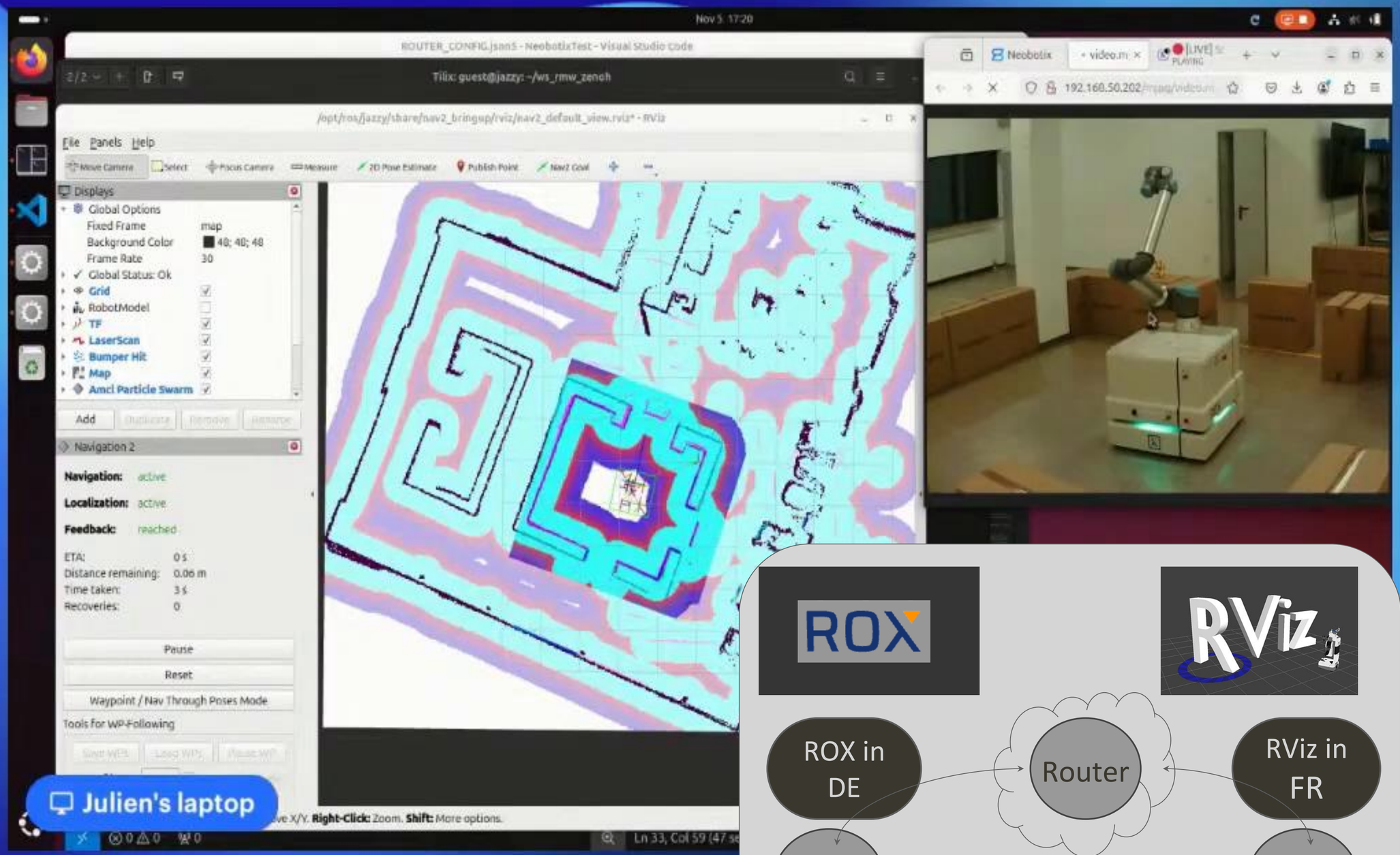


Wi-Fi



Transport reliably over many network hops

zetta scale

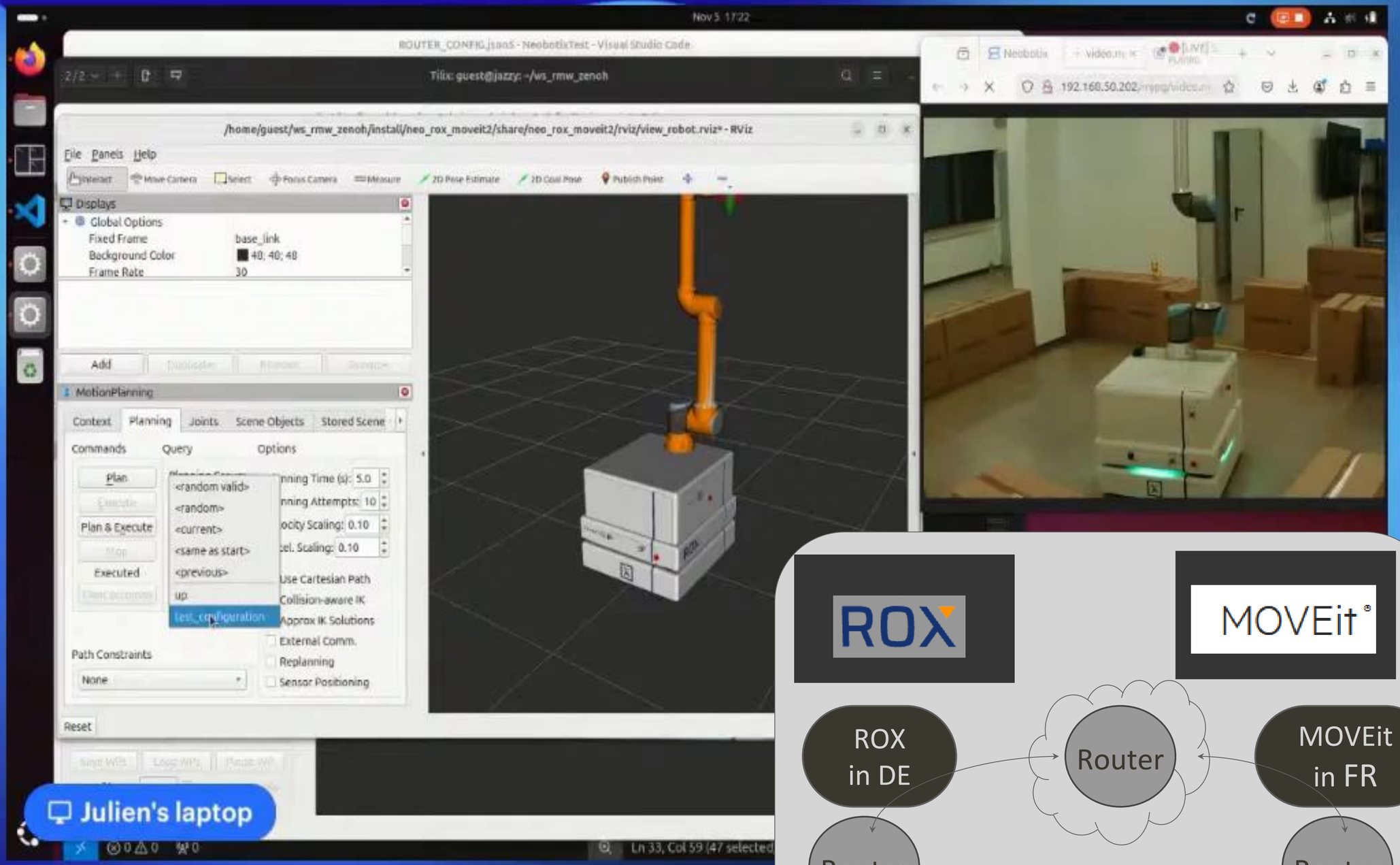

Nov 5 17:20
ROUTER_CONFIG.json5 - NeobotixTest - Visual Studio Code
Tilix: guest@jazzy: ~/ws_rmw_zenoh
/opt/ros/jazzy/share/nav2_bringup/rviz/nav2_default_view.rviz* - RViz
File Panels Help
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
Publish Point
Nav2 Goal
Displays
Global Options
Fixed Frame map
Background Color 48; 48; 48
Frame Rate 30
Global Status: Ok
Grid
RobotModel
TF
LaserScan
Bumper Hit
Map
Amcl Particle Swarm
Add
Navigation 2
Navigation: active
Localization: active
Feedback: reached
ETA: 0 s
Distance remaining: 0.06 m
Time taken: 3 s
Recoveries: 0
Pause
Reset
Waypoint / Nav Through Poses Mode
Tools for WP-Following
Neobotix
video.m
192.160.50.202
Julien's laptop

ROX
RViz
ROX in DE
Router
RViz in FR
Router
Router


NEOBOTIX
Pradheep


Julien Enoch

zetta scale


Nov 5 17:22
ROUTER_CONFIG.json5 - NeobotixTest - Visual Studio Code
Tilix: guest@jazzy: ~/ws_rmw_zenoh
/home/guest/ws_rmw_zenoh/install/neo_rox_moveit2/share/neo_rox_moveit2/rviz/view_robot.rviz* - RViz
File Panels Help
Interact Move Camera Select Focus Camera Measure 2D Pose Estimate 2D Goal Pose Publish Point
Displays
Global Options
Fixed Frame base_link
Background Color 40; 40; 40
Frame Rate 30
Add
MotionPlanning
Context Planning Joints Scene Objects Stored Scene
Commands Query Options
Plan
Plan & Execute
Executed
<random valid>
<random>
<current>
<same as start>
<previous>
up
test_configuration
Use Cartesian Path
Collision-aware IK
Approx IK Solutions
External Comm.
Replanning
Sensor Positioning
Path Constraints
None
Reset
192.168.50.202
Julien's laptop
Ln 33, Col 59 (47 selected)


NEOBOTIX
Pradheep


Julien Enoch

ROX
MOVEit
ROX in DE
Router
MOVEit in FR
Router
Router

Downsample when needed

```
2024-10-16T18:56:51.257913Z DEBUG acc-0 ThreadId(04) zenoh_link_
pted TCP connection on [::ffff:127.0.0.1]:7447: [::ffff:127.0.0.
2024-10-16T18:56:52.770449Z DEBUG acc-0 ThreadId(04) zenoh_link_
pted TCP connection on [::ffff:192.200.40.12]:7447: [::ffff:192.
2024-10-16T18:56:53.282829Z DEBUG acc-0 ThreadId(04) zenoh_link_
pted TCP connection on [::ffff:127.0.0.1]:7447: [::ffff:127.0.0.
2024-10-16T18:56:59.265744Z DEBUG acc-0 ThreadId(04) zenoh_link_
pted TCP connection on [::ffff:127.0.0.1]:7447: [::ffff:127.0.0.
2024-10-16T18:57:03.267460Z DEBUG acc-0 ThreadId(04) zenoh_link_
pted TCP connection on [::ffff:127.0.0.1]:7447: [::ffff:127.0.0.
2024-10-16T18:57:07.269792Z DEBUG acc-0 ThreadId(04) zenoh_link_
pted TCP connection on [::ffff:127.0.0.1]:7447: [::ffff:127.0.0.
2024-10-16T18:57:11.275589Z DEBUG acc-0 ThreadId(04) zenoh_link_
pted TCP connection on [::ffff:127.0.0.1]:7447: [::ffff:127.0.0.
2024-10-16T18:57:15.279280Z DEBUG acc-0 ThreadId(04) zenoh_link_
pted TCP connection on [::ffff:127.0.0.1]:7447: [::ffff:127.0.0.
2024-10-16T18:57:19.282812Z DEBUG acc-0 ThreadId(04) zenoh_link_
pted TCP connection on [::ffff:127.0.0.1]:7447: [::ffff:127.0.0.
```

```
                                                    min: 0.032s max: 0.036s std dev: 0.00151s window: 20
average rate: 30.136
                                                    min: 0.027s max: 0.038s std dev: 0.00298s window: 20
average rate: 30.020
                                                    min: 0.020s max: 0.044s std dev: 0.00445s window: 20
average rate: 30.140
                                                    min: 0.029s max: 0.038s std dev: 0.00227s window: 20
average rate: 29.885
                                                    min: 0.017s max: 0.059s std dev: 0.00824s window: 20
average rate: 29.945
                                                    min: 0.030s max: 0.037s std dev: 0.00108s window: 20
```

Known limitations

- `rclcpp::shutdown()` must explicitly be called before program termination.
- Router must be manually started (for now).
- Liveliness and deadline QoS events not supported.

zetta scale

Thank You

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