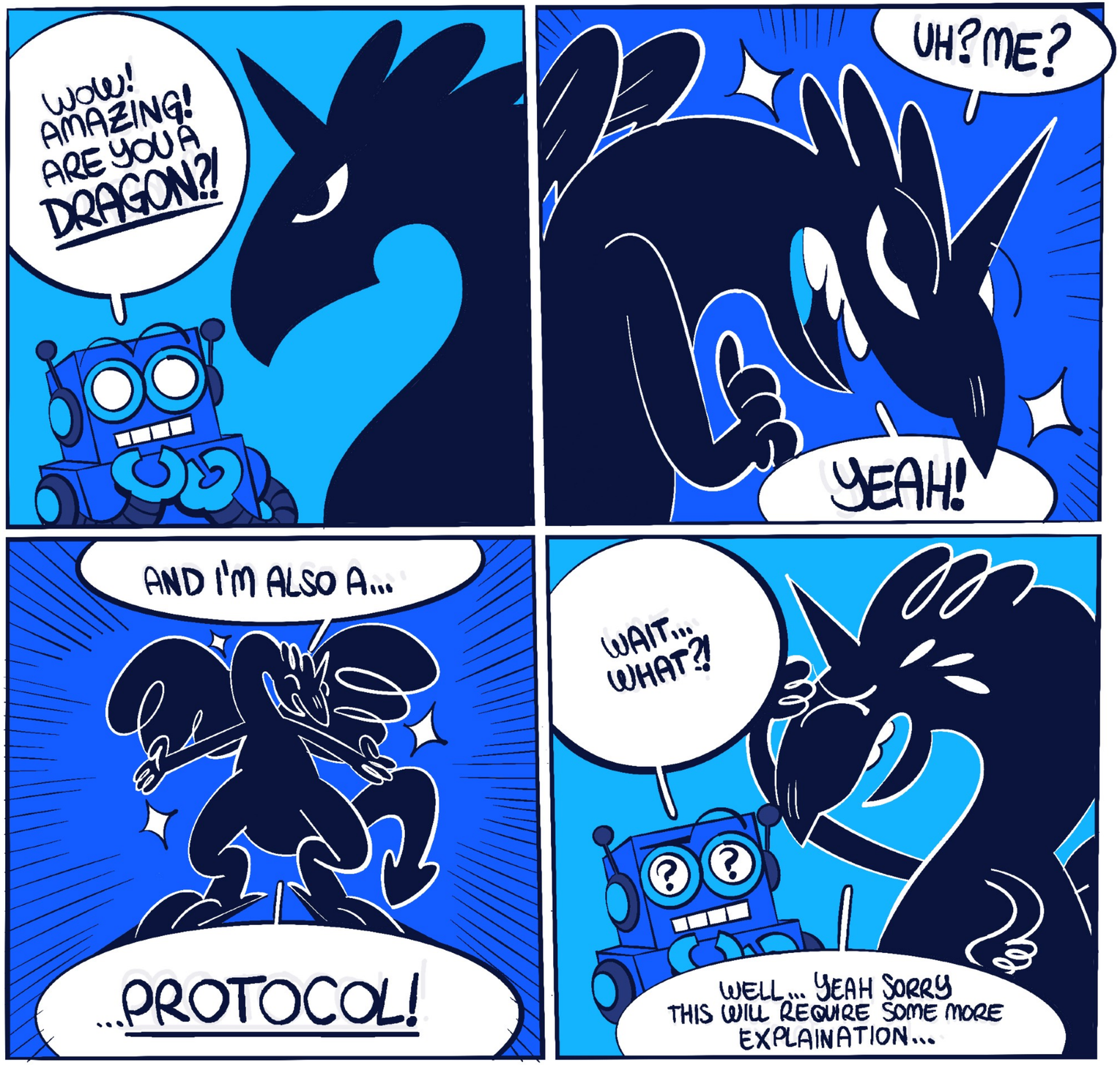
RMW Zenoh Workshop

ROSCon UK 2025 - Edimburgh

Julien Enoch

Senior Solutions Architect
julien.e@zettascale.tech

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

Universal Abstractions

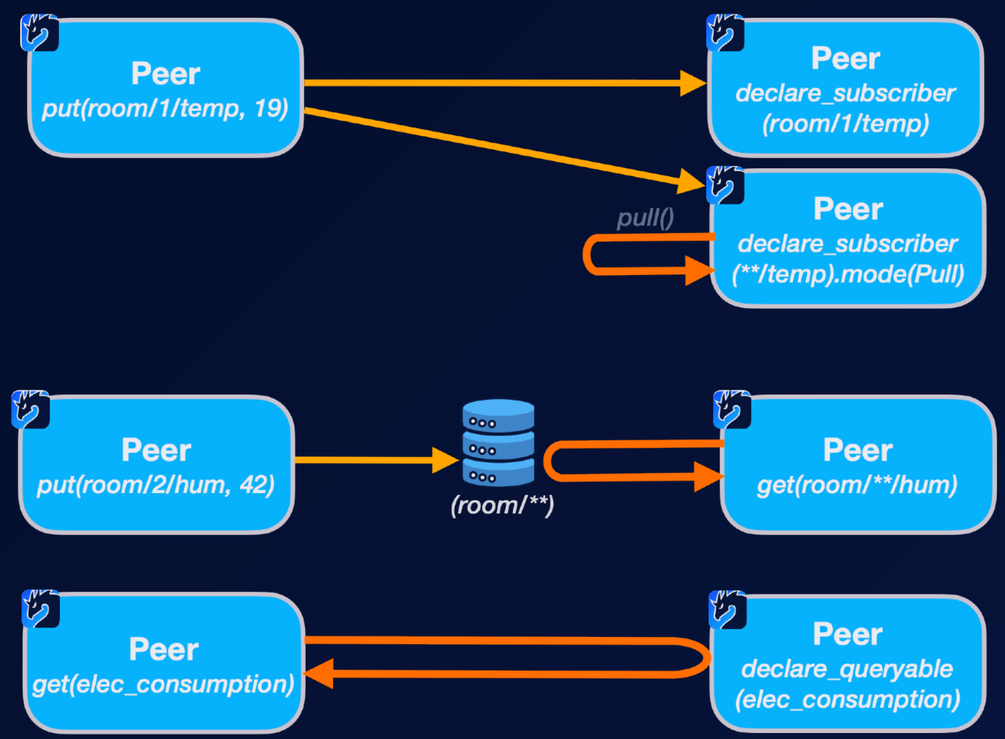
Zenoh's abstraction are **universal** since they allow to express the key patterns in distributed computing, namely:

Publish/Subscribe. Trivially supported by Zenoh's **Publisher** and **Subscriber**

Remote Computation. A **Queryable** represents a generalised computation, since it can transparently deal with replication and partitioning

Storage. Represented by the combination of a **Queryable** and a **Subscriber**

Additionally all these primitives enjoy location transparency by the virtue of being data-centric.



Runs Everywhere

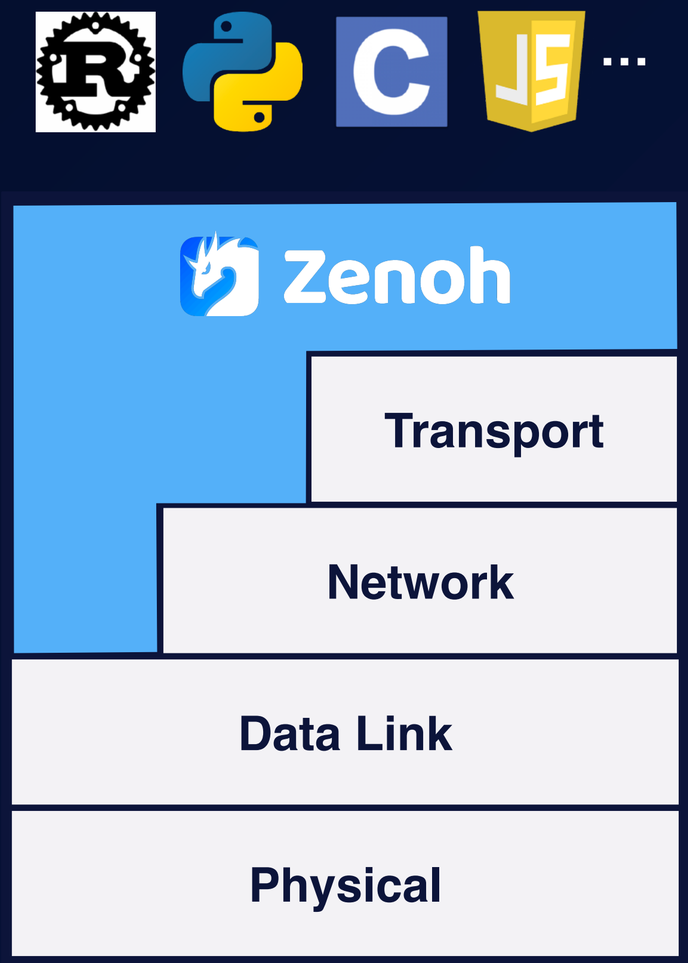
Written in Rust for security, safety and performance

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

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, Shared Memory

Available on **embedded** and **extremely constrained devices** and **networks – 5-6** bytes minimal overhead



Any Topology

Peer-to-peer

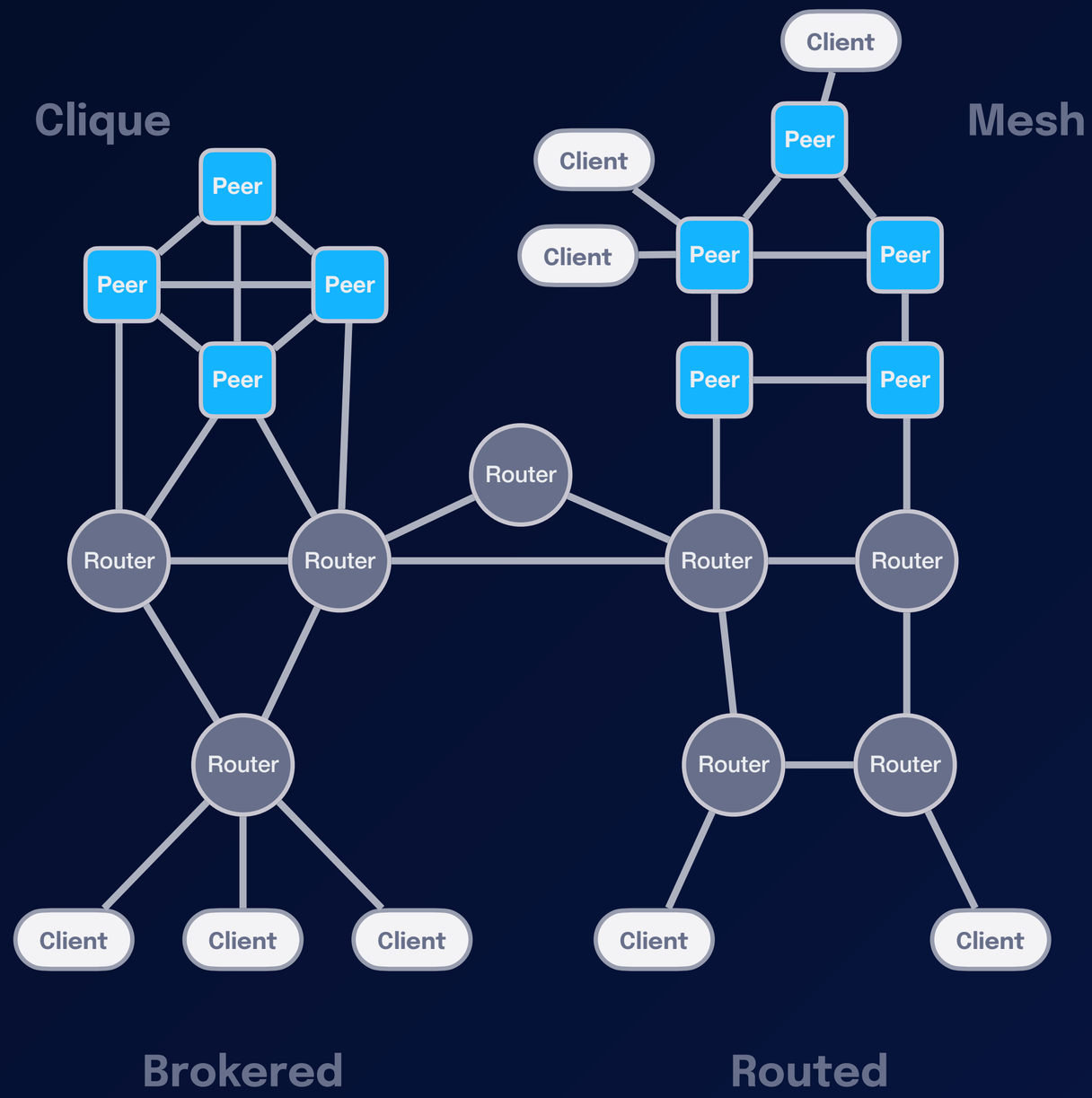
Clique and mesh topologies

Brokered

Clients communicate
through a router or a peer

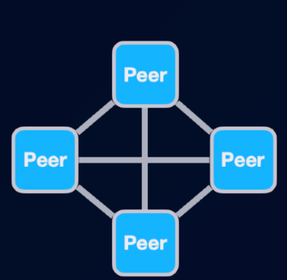
Routed

Routers forward data to and
from peers and clients



zetta scale

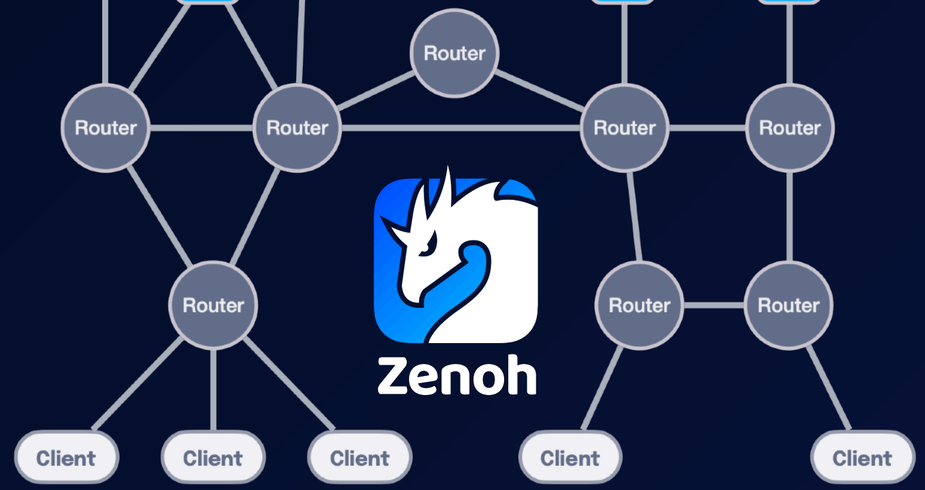
Topology in Perspective



DDS

Clique

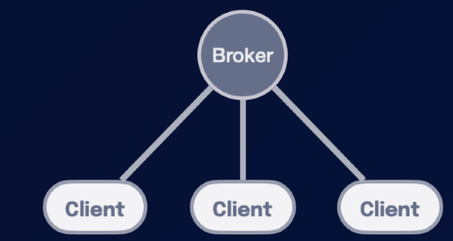
Mesh



Zenoh

Brokered

Routed



MQTT

Plug-Ins

Storage Plugins

influxdb

RocksDB

amazon S3

MINIO

Main Memory

File System

Runtime Plugins

Zenoh Flow

Router

Protocol Plugins

ROS

HTML

ROS2

DDS

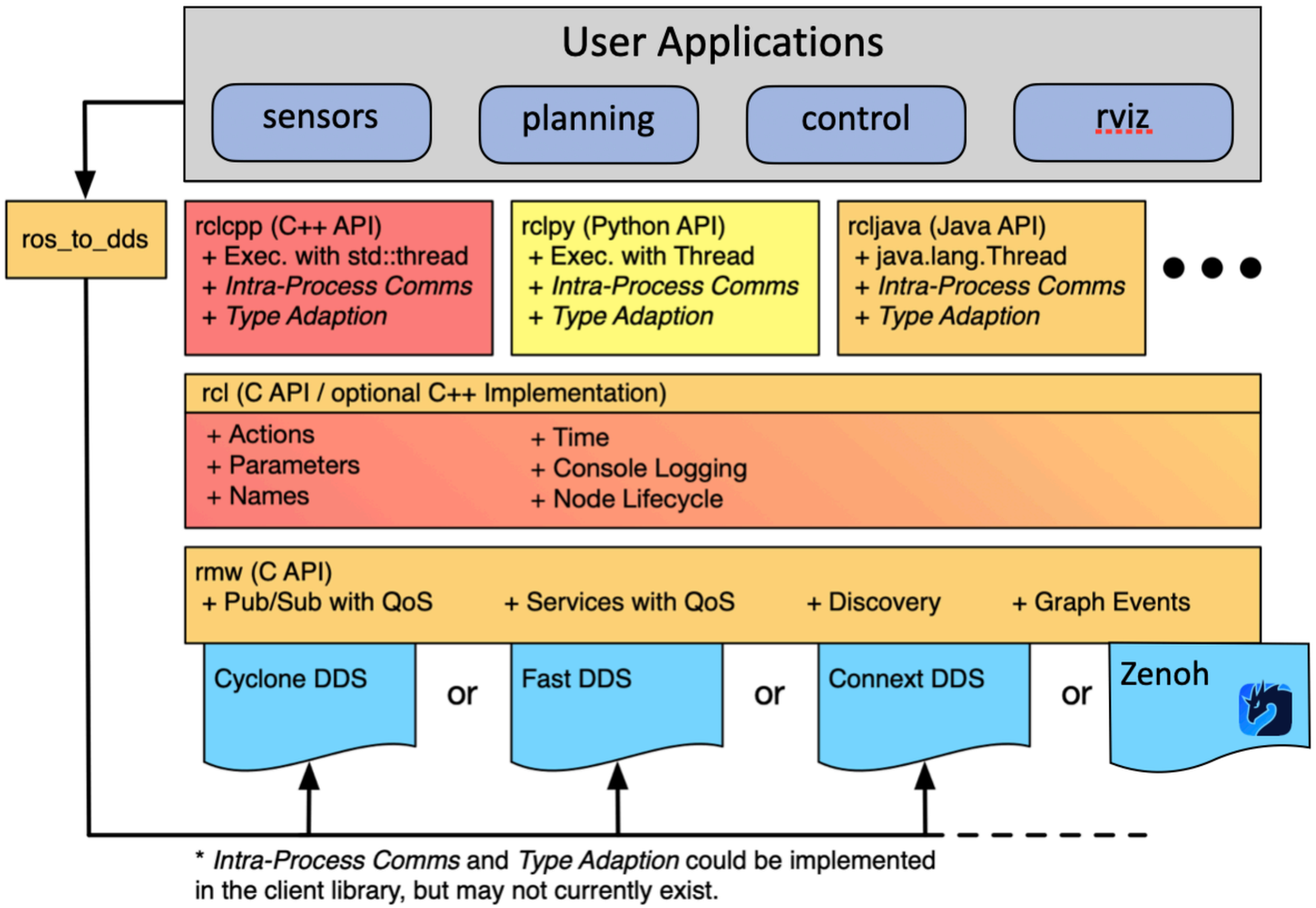
{ REST }

MQTT

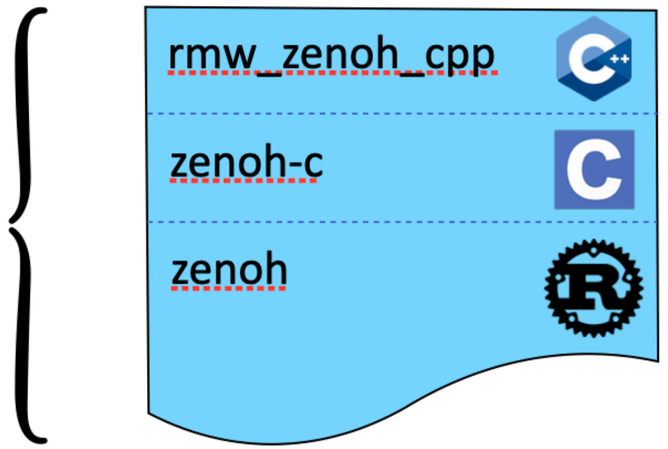
zetta scale

RMW Zenoh

ROS 2 has a modular architecture



https://github.com/ros2/rmw_zenoh



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

ROS to Zenoh Mapping - primitives

Data encoding	Still DDS **CDR** encoding via code generation: .msg/.srv/.action => .idl => serializer/deserializer code
Publisher / Subscriber	Zenoh Publisher / Subscriber
Service Server / Client	Zenoh Queryable / Querier
Actions	Still mapped by RCL on 3 Services and 2 Topics (thus 3 Queryables and 2 Publishers)
Parameters	Still mapped by RCL on Services (thus Queryables)
Entities discovery and ROS Graph	Zenoh Liveliness tokens (lightweight and reactive)

ROS to Zenoh Mapping - Key Expressions

Topics / Services:

`<domain_id>/<fully_qualified_name>/<type_name>/<type_hash>`

`0/robot1/chatter/std_msgs::msg::dds_::String_/RIHS01_df668c74…`

`0/add_two_ints/example_interfaces::srv::dds_::AddTwoInts_/RIHS01_e118de6b…`

`0/fibonacci/_action/send_goal/action_tutorials_interfaces::action::dds_::Fibonacci_SendGoal_/RIHS01_a0603060…`

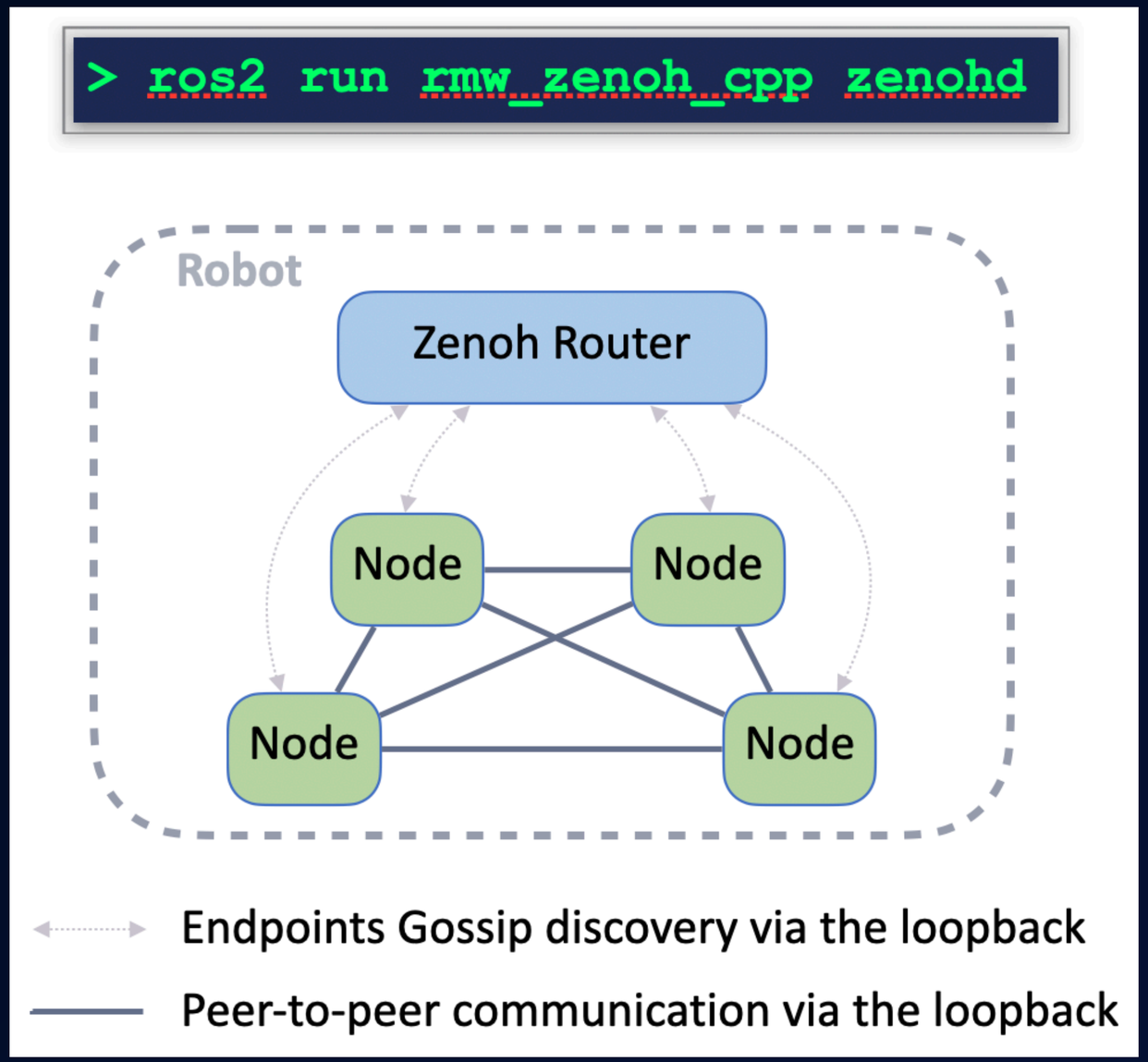
TRANSIENT_LOCAL topics:

the Publisher declares an additional Queryable on

`<domain_id>/<fully_qualified_name>/<type_name>/<type_hash>/@adv/pub/<zid>/<eid>/_`

More info at https://github.com/ros2/rmw_zenoh/blob/rolling/docs/design.md

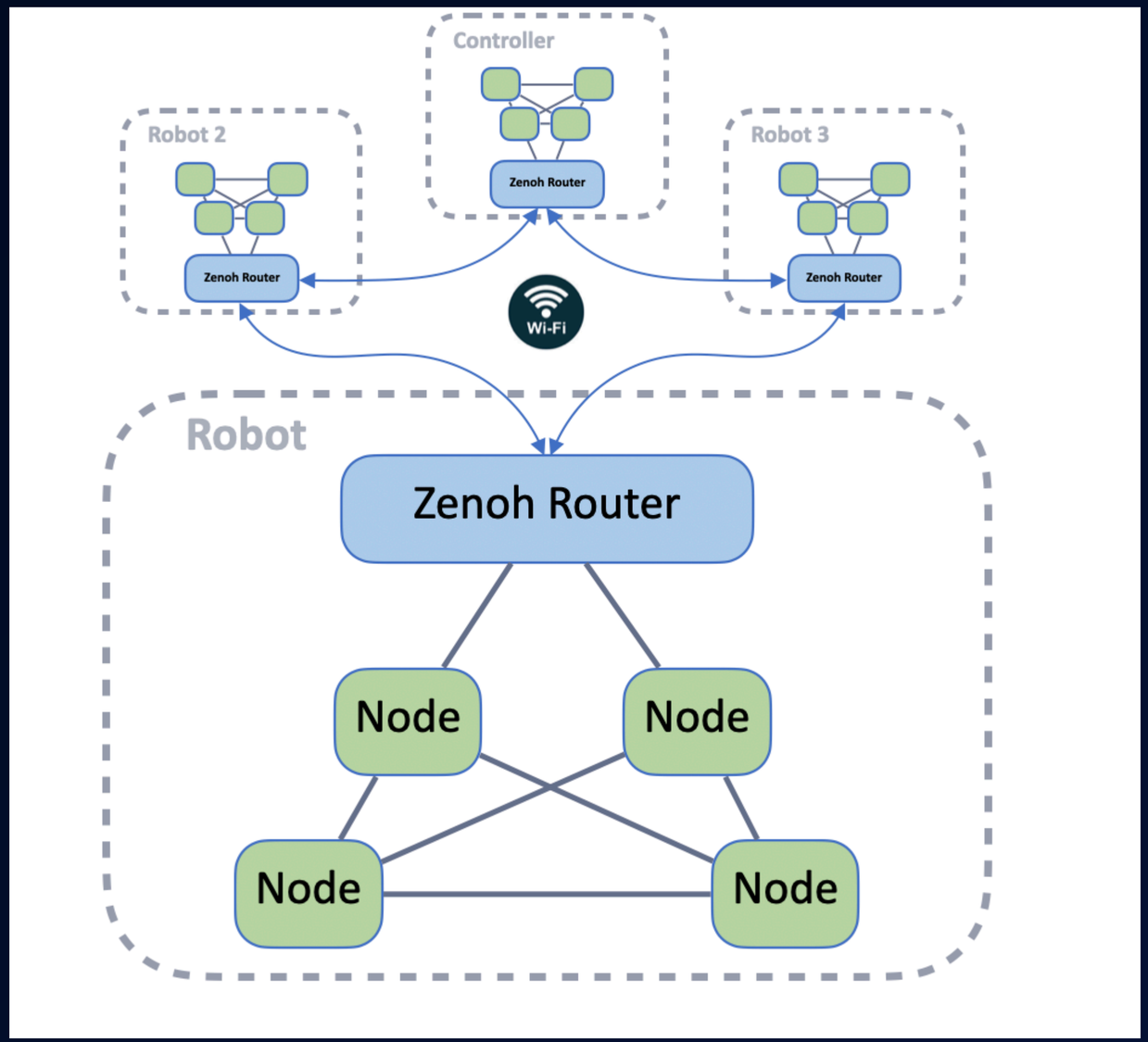
Zenoh Router



For Discovery:

- 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

Zenoh Router



For external communications:

- Configure the router to connect to other routers, via TCP, TLS, QUIC…
- Benefits:
  - Less connections, less overheads
  - Batching for better throughput
  - Smaller surface of attack
  - Single point to configure Access Control and Downsampling

Installation

- Available for **Humble**, **Jazzy, Kilted** and **Rolling**
- Either from ROS repositories
- Either build from sources, as any ROS 2 package:

```
> cd ros_workspace
> git clone https://github.com/ros2/rmw_zenoh src/rmw_zenoh
> rosdep install --from-paths src --ignore-src --rosdistro $ROS_DISTRO -y
> colcon build --cmake-args -DCMAKE_BUILD_TYPE=Release
> export RMW_IMPLEMENTATION=rmw_zenoh_cpp
```

Already available in your image:

zettascaletech/roscon2025_workshop

Configuration

- 1 configuration file (json5, json or yaml) per Node and Router
- Default config files in sources under : `rmw_zenoh/rmw_zenoh_cpp/config/`
- **`$ZENOH_SESSION_CONFIG_URI`** for Nodes config
  - Most of the time nothing to change here
- **`$ZENOH_ROUTER_CONFIG_URI`** for Router config
  - External connectivities
  - TLS keys and certificates
  - Access control
  - Downsampling
  - …

zetta scale

Thank You

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