Agenda

1. Opening
2. Zenoh: The Genesis
4. Zenoh and Zenoh-Flow Hands-on
5. Zenoh High Performance Networking with Rust
6. Q&A

Welcome
Unifies data in motion, data at rest and computations from embedded microcontrollers up to data centres

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

Facilitates geo-distributed storage and integration with third party technology in a plug-and-play fashion
Application Domains in Real Life
Indie Autonomous Challenge
Robotic Systems with ROS/ROS2
Online Gaming
The Core Team

Angelo Corsaro  Luca Cominardi  Julien Loudet  Olivier Hecart  Julien Enoch  William Liang

Gabriele Baldoni  Carlos Guimaraes  Sreeja Nair  Pierre Avital  Darius Maitia  Zenoh Community
...with an open community in Discord
https://discord.gg/vSDSpqnbkm

Eclipse Zenoh is fully open-source
https://github.com/eclipse-zenoh
Zenoh

Zero Overhead Pub/sub, Store/Query and Compute.

zenoh /zeno/ unifies data in motion, data at rest and computations. It elegantly blends traditional pub/sub with geo distributed storage, queries and computations, while retaining a level of time and space efficiency that is well beyond any of the mainstream stacks.

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

https://zenoh.io/

...and the blog

Zenoh-Pico: Above and Beyond

09 June 2022 -- Paris.

In a previous blog post, we introduced Zenoh-Pico, an implementation of Zenoh for microcontrollers and embedded devices, along with a preliminary performance results and its integration on off-the-shelf robots (by bridging both legacy ROS2+DDS and Zenoh systems or by making it a full-fledged Zenoh system).

In this post, we will dive deeper on Zenoh-Pico, show how Zenoh-Pico is capable of:

- exchanging close to 2.5M msg/s for small payloads, and over 25 Gbps for larger messages,
- achieving end-to-end latency (i.e., one way delay) as small as 45 μsec and 15 μsec for unicast and multicast transports, respectively,
- minimizing the overhead in the wire down to 5 bytes per data transmission,
- fitting all its capabilities in less than 50KB footprint, which can be quickly reduced to ~15KB in tailored compilation setups, and
- provides simple to use and yet powerful APIs.