

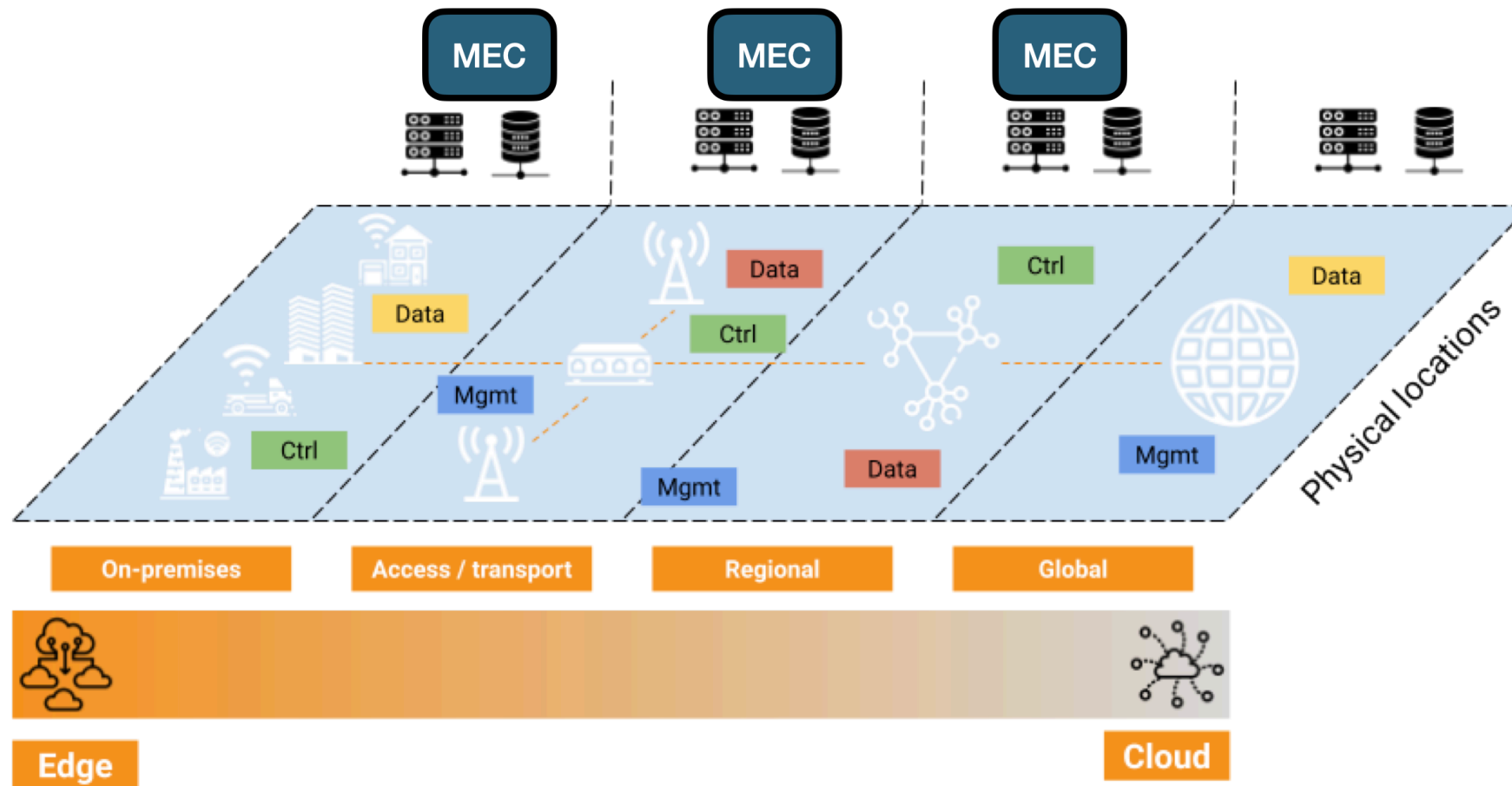
# Eclipse Zenoh and ETSI MEC

Presented by: **Luca Cominardi (ADLINK)** For: **ETSI MEC#198**

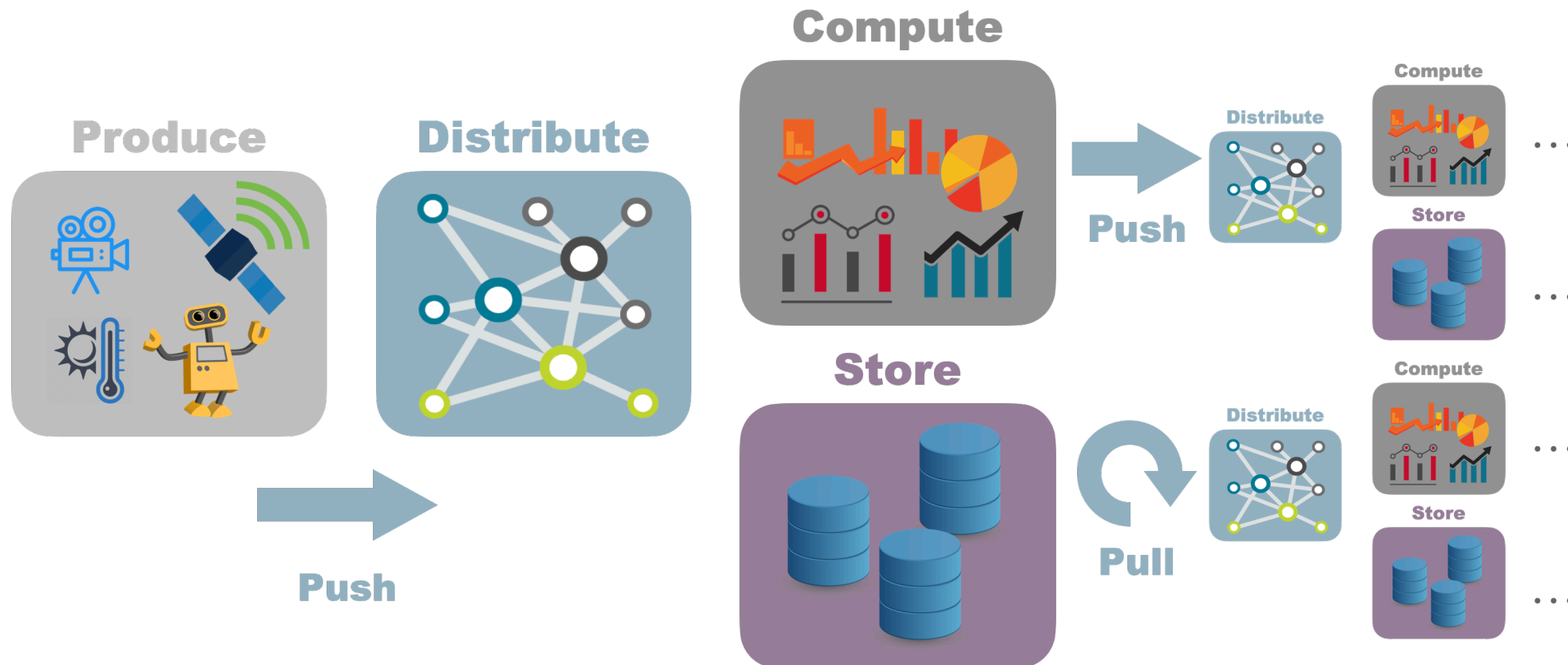
16.03.2021

# Context

# Edge to Cloud continuum



# The Data Journey



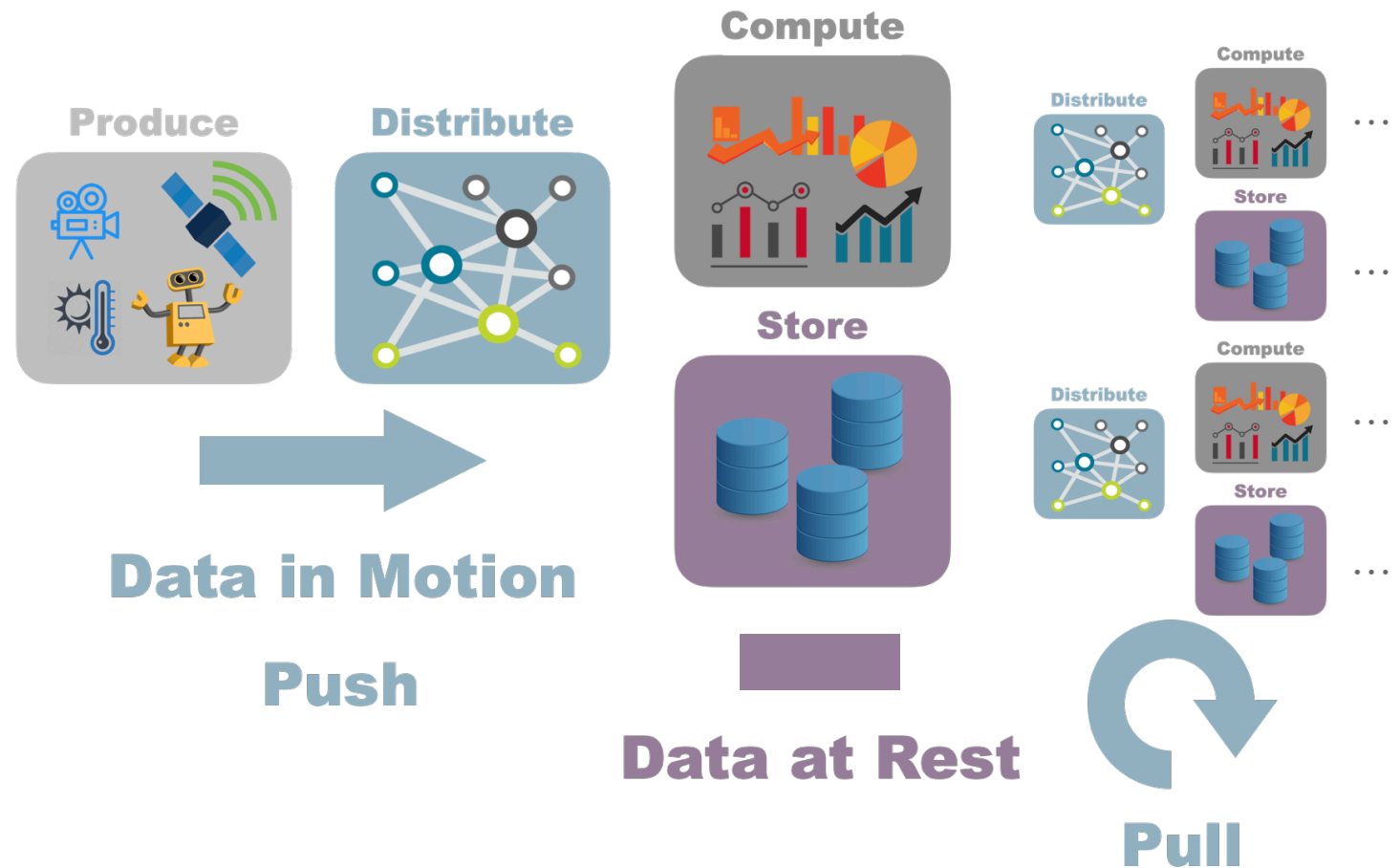


# Moving and Resting

Technologies for dealing with **data in motion** and **data at rest** have belonged historically to different families: **OT** vs **IT**.

**Publish/Subscribe** is today the **leading paradigm** for dealing with **data in motion** (push).

**Databases** (SQL and NoSQL) are the **leading paradigm** to deal with **data at rest** (pull).



# Technological Fragmentation

The increasing **availability** of and **storage, compute** capabilities on **devices** is creating new **opportunities** for **computing** and **storing** data much **closer** its **production**.

**Existing technologies** for data in motion and data at rest **fall short** in **supporting** this **scenario**. More importantly fail to provide a **unified data management**.





Zero Overhead  
Pub/Sub  
Store/Query  
Compute

# Eclipse zenoh

**Unifies data in motion, data in-use, data at rest and computations.**

It carefully **blends** traditional **pub/sub** with **distributed queries**, while retaining a level of **time and space efficiency** that is well beyond any of the mainstream stacks.

It provides built-in support for **geo-distributed storages** and **distributed computations**.

**Open-source project** under the umbrella of the Eclipse Edge Native Working Group.

**EDGE**  
NATIVE



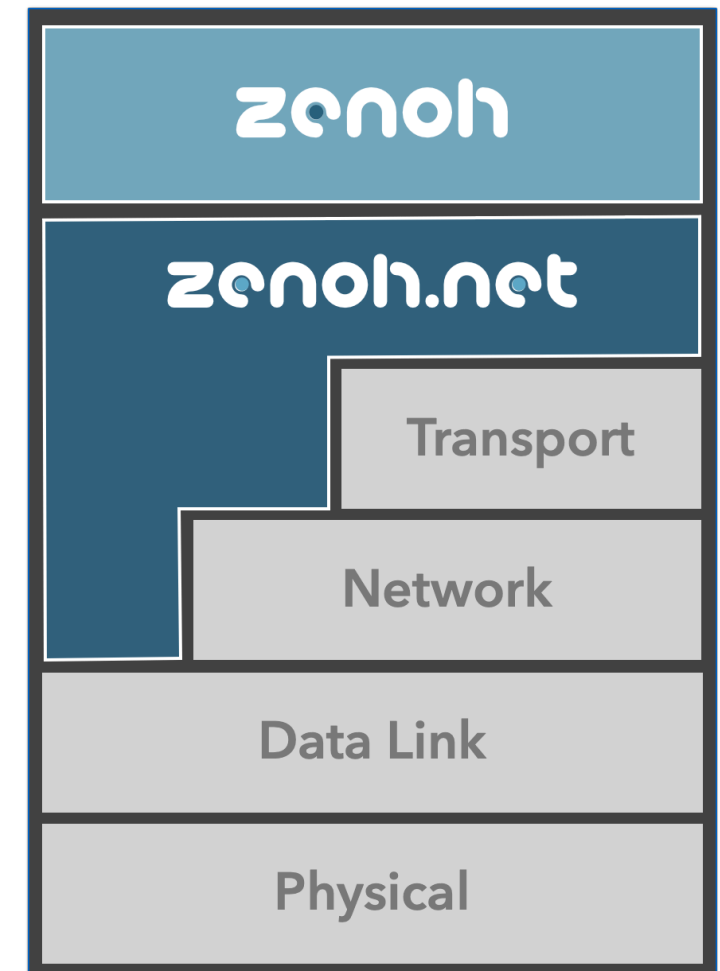
# Protocol Stack

## zenoh.net

A **zero overhead networking protocol** that provides **composable primitives** for **pub/sub** and **generalised distributed queries**.

## zenoh

A **framework** that **leverages zenoh.net primitives** to provide an **opinionated implementation** of **pub/sub**, **geo-distributed storage** and **computations**.





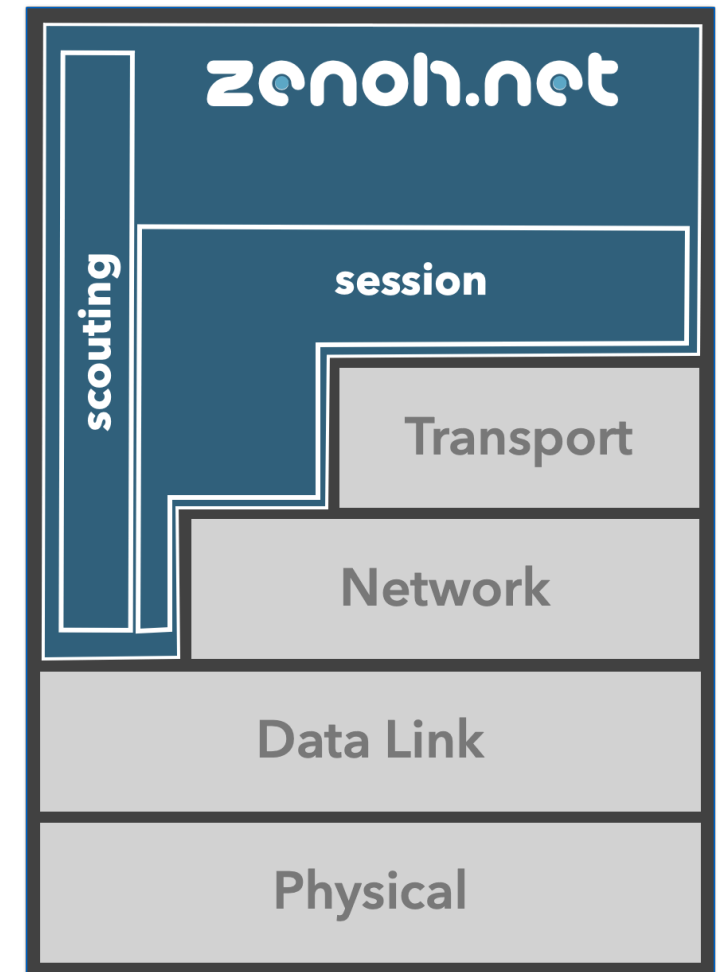
**zenoh.net**



Implements a **networking layer** capable of running above a Data Link, Network or Transport Layer.

It provides primitives for **efficient pub/sub** and **distributed queries**. It supports **fragmentation** and **ordered reliable delivery**.

It provides a pluggable **scouting** abstraction for discovery.





# zenoh.net Key Abstractions

---

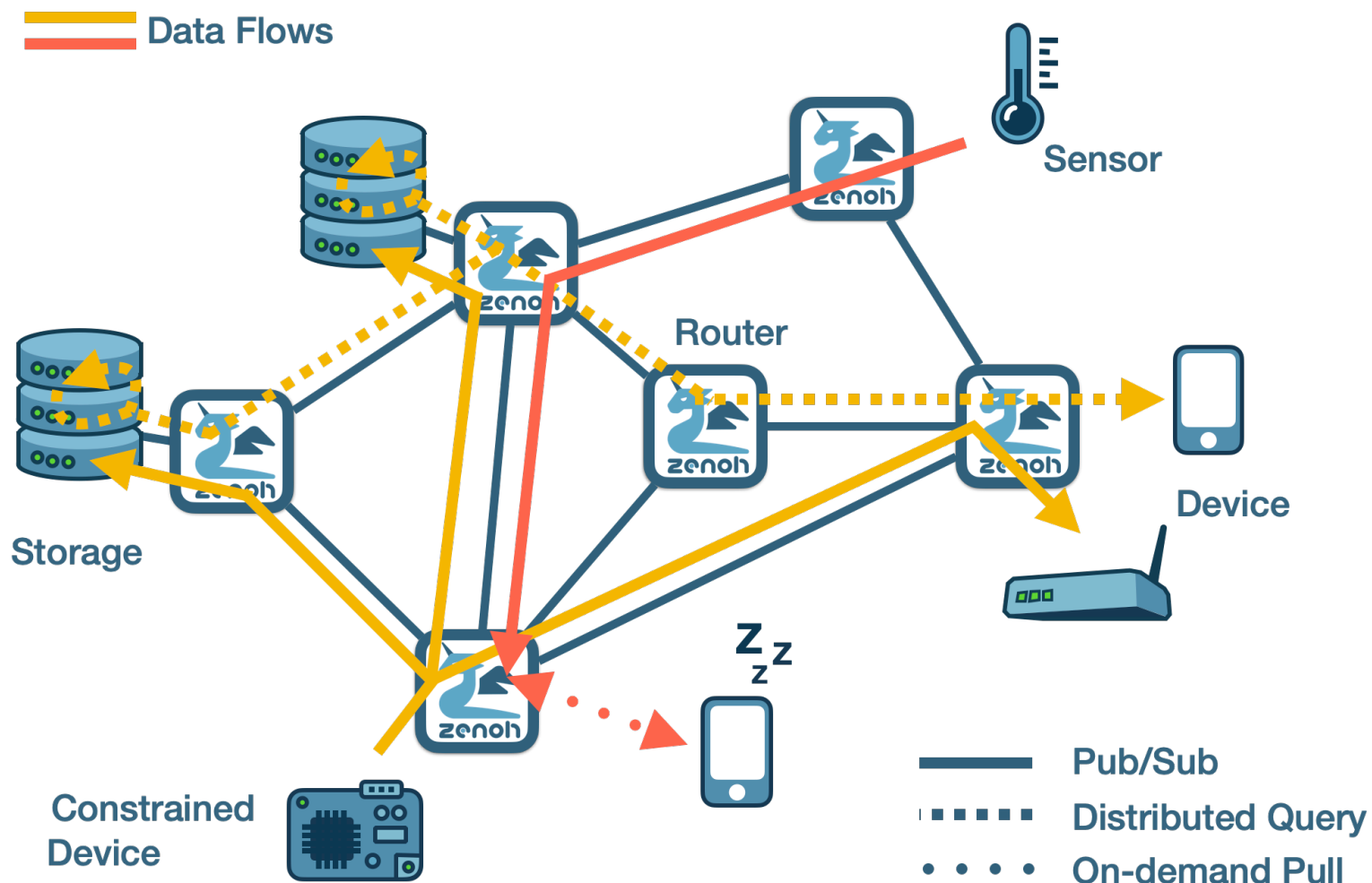
**Resource.** A **named data**, in other term a (key, value)  
(/home/kitchen/sensor/temp, 21.5), (/home/kitchen/sensor/hum, 0.67)

**Publisher.** A **spring** of values for a key expression  
/home/kitchen/sensor/temp, /home/kitchen/sensor/hum

**Subscriber.** A **sink** of values for a key expression  
/home/kitchen/sensor/temp, /home/kitchen/sensor/\*

**Queryable.** A **well** of values for a key expression  
/home/\*\*

# zenoh.net Putting all Together



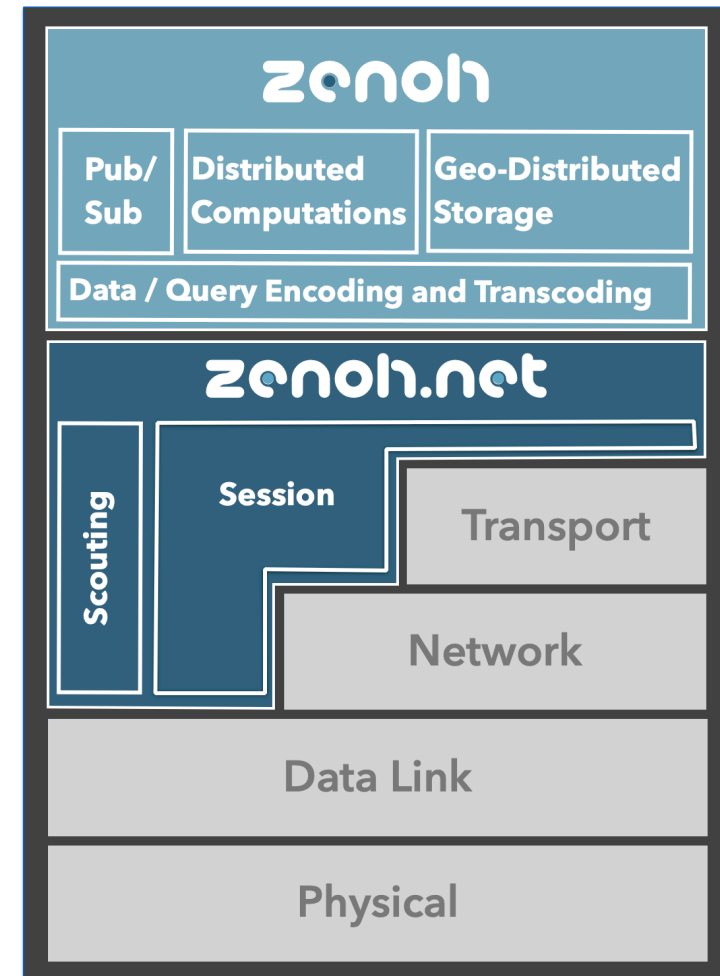


zenoh

Provides a **high level API** for **pub/sub** and **geo-distributed storages and computations**.

It handles **data** representation **transcoding** and provides an implementation of **geo-distributed storage** and **distributed computed values**.

Defines a series of **supported data encoding**, such as **JSON**, **Properties**, **Relational**, **Raw**, etc., along with **transcoding**.

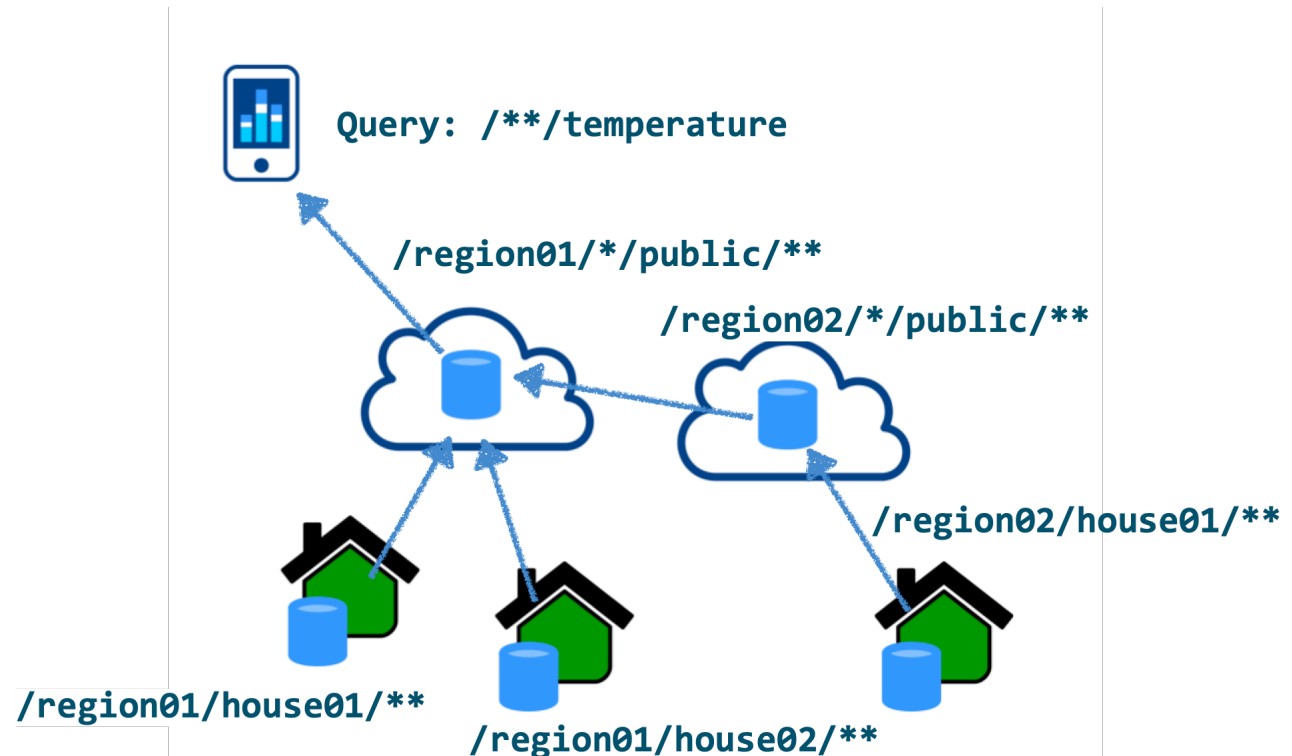




# zenoh Geo Distributed Storage

The **ownership** of data is **specified** through **key expressions**.

**Queries** are able to **resolve** data in a **location transparent** manner.



# Summary

# zenoh.net and zenoh Protocol Highlights

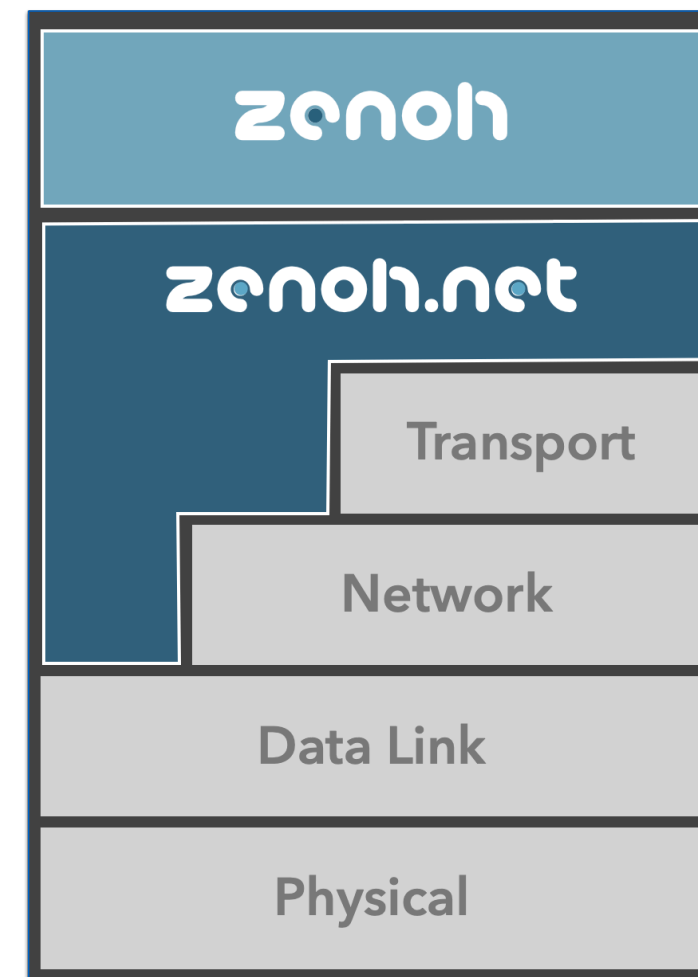
Supports **push** and **pull pub/sub** along with **distributed queries** and **geo-distributed storages**.

**Location-transparent access** to data.

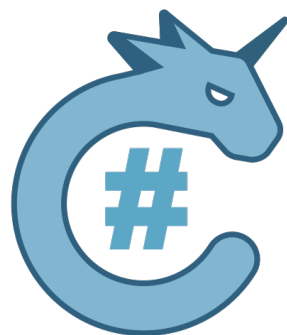
Supports for **peer-to-peer** and **routed communication**.

Minimal **wire overhead** for user data is **4 bytes**.

**Most wire/power/memory efficient protocol** in the market to provide connectivity to extremely **constrained targets**.



# zenoh.net and zenoh APIs



zenoh-c#



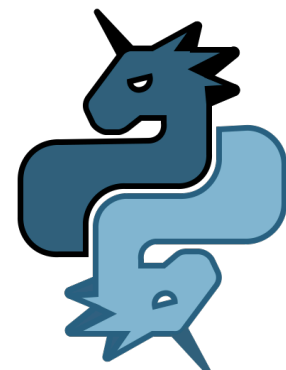
zenoh-rust



zenoh-c



zenoh-pico



zenoh-py



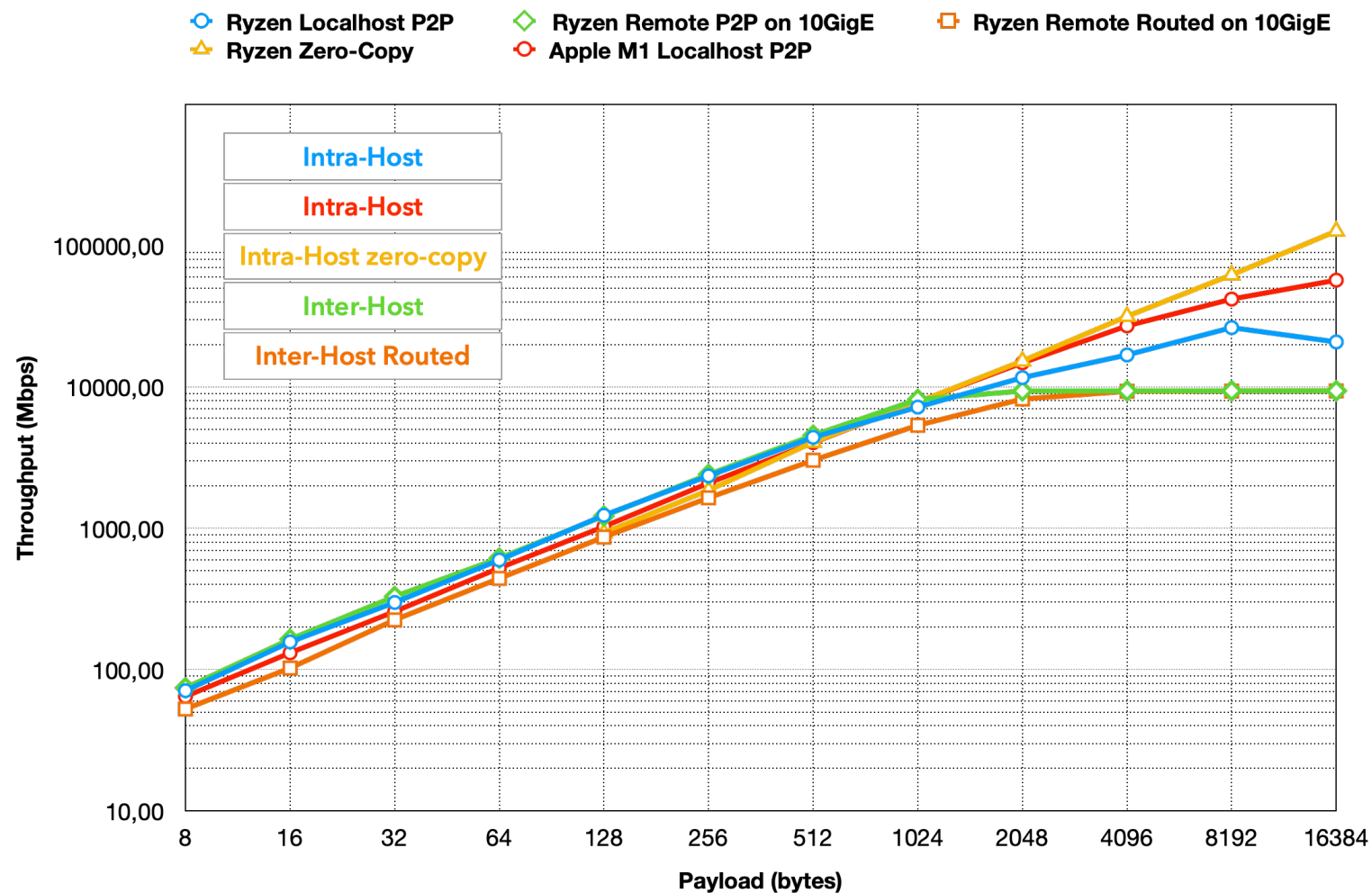
zenoh-java




zenoh-go

Coming Up

# zenoh.net Throughput (Mbps)





A photograph of two young men sitting on a large number of empty, light blue stadium seats. The man on the left has dreadlocks, wears glasses, a denim vest over a plaid shirt, and has red headphones around his neck. The man on the right wears a grey beanie, sunglasses, and a brown jacket. They are both smiling and looking at a white smartphone held by the man on the right. The background is a vast expanse of empty seats stretching into the distance under bright, natural light.

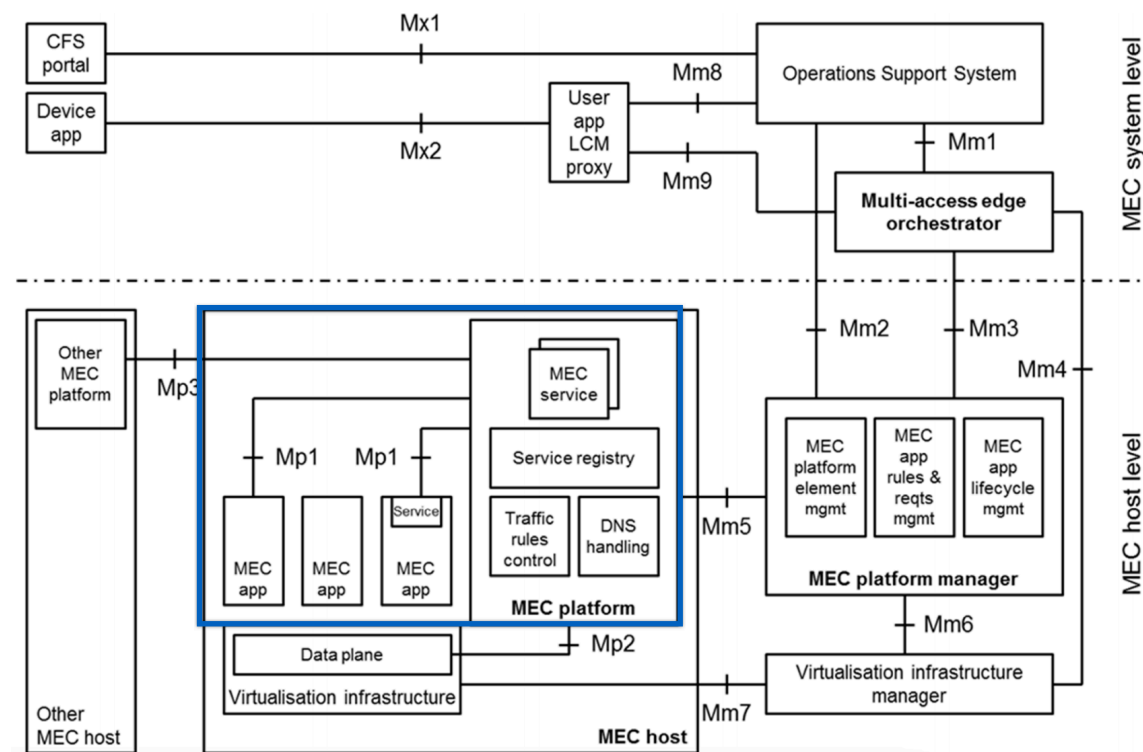
# Relationship between Eclipse Zenoh and ETSI MEC



# zenoh and Scouting (MEC011)

The **Mp1** reference point between the MEC platform and the MEC applications provides service registration, **service discovery**, and communication support for services.

Zenoh can exploit MEC service discovery to ease the **initial bootstrapping** in mobile environments: **router** and **peer scouting**.



ETSI GS MEC 003 V2.2.1 (2020-12) Figure 6-1: Multi-access edge system reference architecture

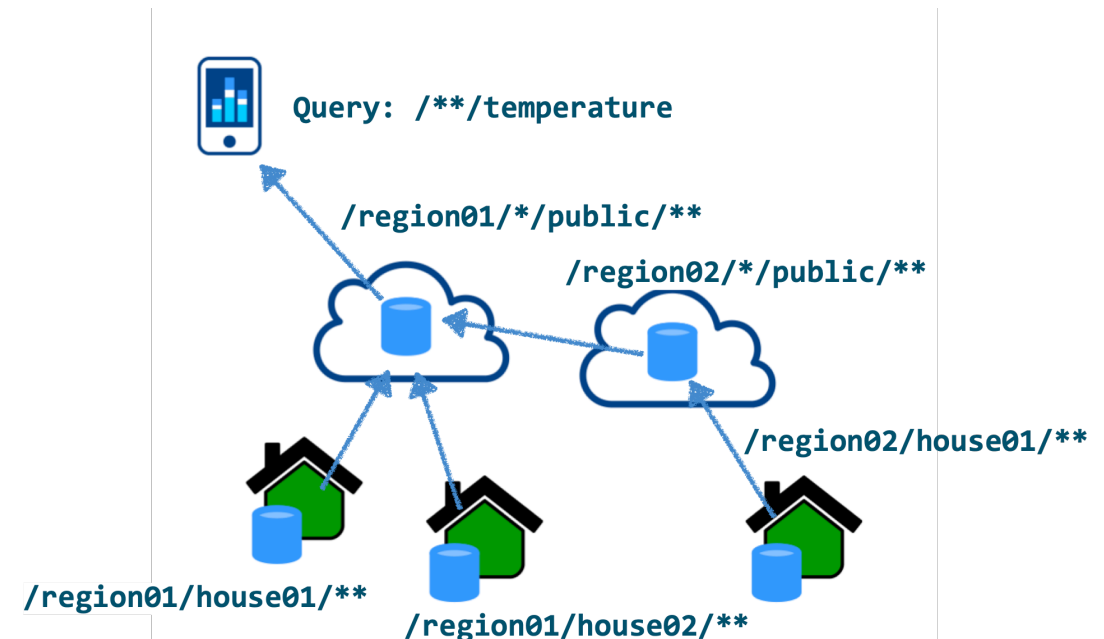
# zenoh as Transport Protocol (MEC011)

**Zenoh** can be used as additional transport for **MEC applications** and **MEC platform**.

Supports **pull** mode and **distributed queries**.

**Location-transparent** access to data.

Unified transport protocol for **constrained devices** and **cloud-like applications**.

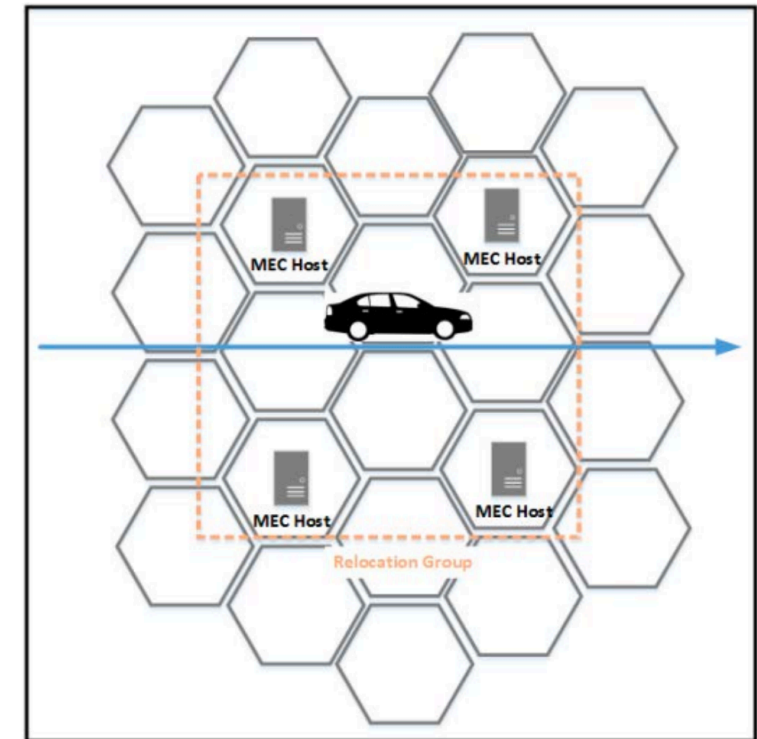


# zenoh User context transfer (MEC021)

**Geo-distributed storages** and **location-transparent access** are natively supported by Zenoh.

**Application self-controlled** user context **transfer** is provided **out-of-the box** when the application uses Zenoh.

**MEC assisted** user context **transfer** can be easily provided if the MEC platform uses Zenoh underneath for **geo-distributed storages**.



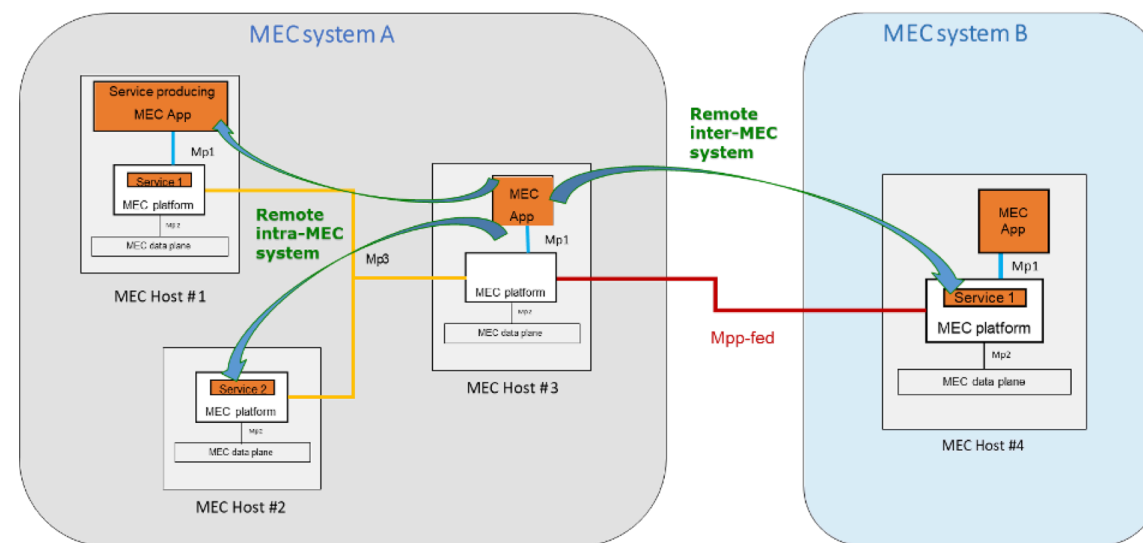
ETSI GR MEC 018 V1.1.1 (2017-10)  
Figure 4.4-1: Preconfigured Relocation Group

Zenoh supports **routing of data**: out-of-the-box **service consumption across MEC systems**.

Similar to MEC021, user context transfer is provided **out-of-the-box** also when **interacting with the cloud**.

Once **MEC-Cloud** interaction is established at system level, applications have **transparent access to data** regardless their location.

NDN-based approach adopted by Zenoh makes unnecessary to discover APP instances, focus on **data** rather than **endpoints**.



Draft ETSI GR MEC 035 V2.0.16 (2021-03)

Figure 6.5.2-1: MEC federation scenario enabling edge service consumption across MEC systems.

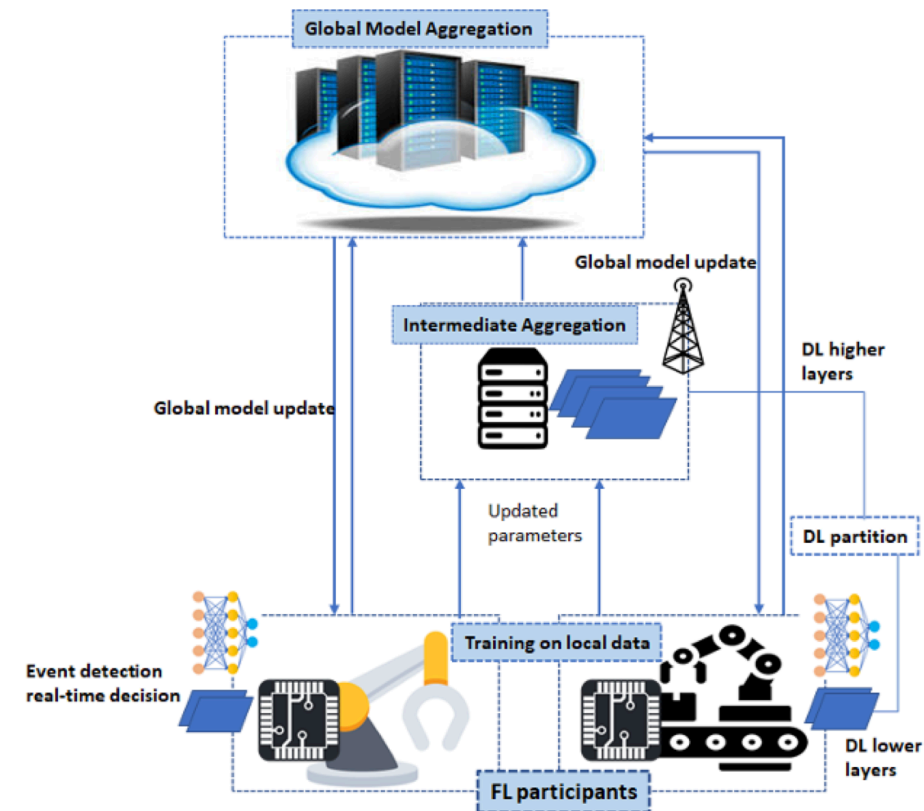
# zenoh Constrained Devices (MEC036)

Zenoh provides **minimal** wire, memory, and power **overhead**.

Zenoh supports device **duty-cycles** like **sleep mode**.

Zenoh supports **fire-and-forget** pub-mode, which is very well suited for **real-time** constrained devices (e.g., robots, sensors).

No need of multiple protocols stitched together to support an **end-to-end data semantic**.



Draft ETSI GR MEC 036 V3.0.4 (2021-01)  
Figure: Smart Factory – Augmented FL-DL approach



# Concluding remarks



# ETSI MEC and **zenoh**

---

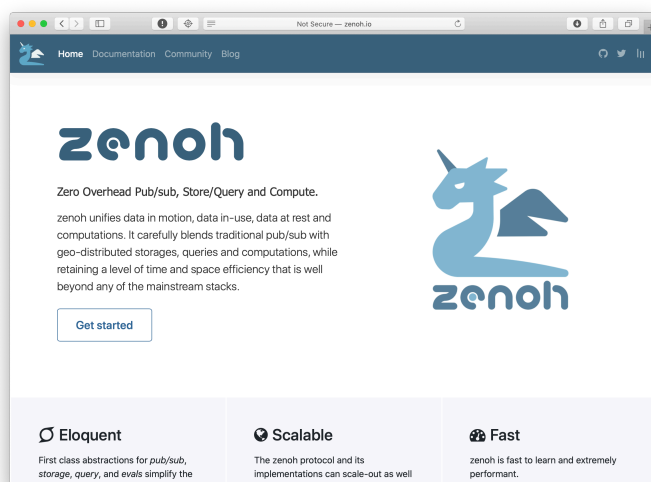
ETSI MEC defines a set of specifications to enable an IT-like environment at the Edge: **APIs** and **procedures**.

Zenoh provides a unified protocol to efficiently cope with **data distribution**, **data storage** and **data management** across an edge-to-cloud continuum.

ETSI MEC provides the necessary support to Zenoh when operating in a **mobile environment** like **service discovery**.

Zenoh can **simplify** the **development** and **operation** of (constrained) **applications** interacting with the **MEC platform** and the **Cloud**.

# References



<http://zenoh.io/>



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



<https://gitter.im/atolab/zenoh>

# Backup

# zenoh.net Naming Data

Following the tradition of Named Data Networking protocols, **data is named by a sequence of byte arrays** — called **key** — such as:

```
/home/kitchen/sensors/temp  
/home/kitchen/sensors/C202
```

**Data interest and intents are expressed by means of keys regular expressions**, such as:

```
/home/*/sensors/temp  
/home/**/C202
```

# zenoh.net Selecting Data

Uses **selector** to **defines data sets**. A selector is composed by a **key expression**, and optionally a **predicate**, a **projection** and a set of **properties**.

```
/myhome/*/sensor/temp?value>25  
/mycar/dynamics?speed>25#acceleration
```

**zenoh.net** uses the **key-expression** to **route the query**, but **does not interpret the predicate nor the properties**. It also provide different **policies** to control **query consolidation** and **completeness** and potentially **quorums**.



# zenoh.net and zenoh primitives mapping

