Conceptual model
Host-centric vs Data-centric

For historical reasons, internet has been built on a **host-centric** communication model. *(machine-to-machine)*

But what matters to the user is the **data** not as much who has it...

The **diffusion** of the same data to multiple consumers is very **inefficient**.
Zenoh provides a data-centric abstraction in which applications can produce and consume data autonomously and asynchronously.
URI based data organisation

• Data is organised as a **Key/Value space**.

• Keys follow an **URI scheme**.
  
  /myhouse/floor01/musicroom/LightStatus
  /myhouse/floor02/musicroom/LightStatus
  /myhouse/floor02/bedroom/erik/LightStatus

• **Data sets** can be identified using **wildcards**.
  
  /myhouse/floor02/bedroom/*/LightStatus
  /myhouse/floor*/bedroom/*/LightStatus
  /myhouse/**
  /myhouse/**/LightStatus
IT/OT Convergence
Push/Pull

The convergence between IT and OT is creating an increasing need to properly integrate the traditional data-at-rest query-based IT world with the Data in Motion, **pub/sub oriented** OT world.
Conceptual Model

Data can be
• pushed to subscribers and storages
• computed on demand
• queried from storages and evals
Data at rest
Cloud-Based Solution

One common approach is to use the cloud as the place to store and retrieve information.

But what about:
- Latency?
- Privacy?
- Connectivity?
Decentralisation

What if we want to keep some of the data locally?

That would make sense from energy, processing and privacy perspectives.

But if we keep data locally, how can we still provide global access to it?
Decentralisation

Query: /**/temperature

/region01/**/public/**
/region01/house01/**
/region01/house02/**
/region02/**/public/**
/region02/house01/**
Data in motion
Heterogeneous environment

The different devices connected to the system use very **heterogeneous networking** technologies (TCP/IP, BLE, 3G, 6LowPan, …).

Some endpoints are **extremely constrained** w.r.t computational, communication resources as well as energy.
Protocol details

Use of **Variable Length Encoding**.

Minimal **overhead** of **3 bytes** in data messages.

Protocol implementation for a **8-bit micro-controllers** takes **300 Bytes of RAM**.

**Independent** of underlying **transport**.
zenoh routers

Bring **connectivity** between devices
• in different **subnetworks**
• using different **transports**

Allow **efficient diffusion** of the same data to different devices.
Adaptative & Fault-tolerant Routing
zenoh unifies data in motion, data in-use, data at rest and computations.

It carefully blends traditional pub/sub with geo-distributed storages, queries and computations, while retaining a level of time and space efficiency that is well beyond any of the mainstream stacks.
Data can be **pushed-to**, **pulled** periodically or asynchronously or **queried-from** storages and evals.
Router plugins

**http plugin**

Provides access to the zenoh data space through a **REST API**

Offers a web based **administration tool**

**storages plugin**

- Memory
- SQL databases
  - SQLite
  - MariaDB
  - PostgreSQL
- Time series databases
  - **influxdb**
Zenoh clients

zenoh-c
zenoh-ocaml
zenoh-python
zenoh-java
zenoh-go
zenoh-cat
Publish:

```
ws = Zenoh.login().workspace()
ws.put('/demo/hello', Value('Hello world'))
```

Subscribe:

```
ws = Zenoh.login().workspace()
ws.subscribe('/demo/**', lambda data: print('received {}'.format(data)))
```

Query:

```
ws = Zenoh.login().workspace()
result = ws.get('/demo/hello?(name=World)')
```
Innovating Together

Olivier Hécart

ADLINK Tech. Inc.
olivier.hecart@adlinktech.com