THE MAKING OF EDGE INTELLIGENCE

ADLINK
Leading EDGE COMPUTING

Advanced Technology Office
28 rue Jean Rostand
91400, Orsay
France

Angelo Corsaro, PhD

Chief Technology Officer
ADLINK Tech. Inc.
angelo.corsaro@adlinktech.com

ATO

ADLINK

ADLINK at a Glance

Established August **1995**	**Chairman** Jim Liu **President** Daniel Yang	Headquarters in **Taipei, Taiwan**
Publicly Traded Since **2002**	Taiwan Stock Exchange Listing **TAIEX: 6166**	Revenue (USD) **$297M** (Y2016) **$350M** (Y2017)
>1900 Employees	USA · Germany Canada · UK France · Singapore India · China Korea · Japan Israel	Capital (USD) **$73M** Market Cap (USD) **$468M**

Vision

To be the catalyst for industry empowered by artificial intelligence

Mission

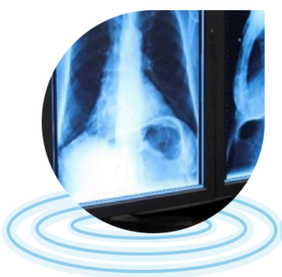
To effect positive change in industry by connecting people, places and assets with artificial intelligence through the delivery of leading EDGE, robust and reliable hardware and software solutions that directly address mission-critical business and technology challenges

Business Units Focus


IoT Software


Machine Vision


Gaming


Medical Imaging


Telecom & Network

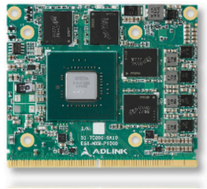

Military

IST : IoT Solutions & Technologies

- Open Instr. : Data Acquisition, Module Instr.
- Smart Machine : Motion, Vision and Industry IO
- Connected Factories, IoT Software & Solutions

EPM: Embedded Platforms & Modules

- Embedded Box and Panel Computer for faster TTM
- Intelligent Modules : CPU, GPGPU, VPU and FPGA
- Application-Ready Platform for Healthcare & Gaming



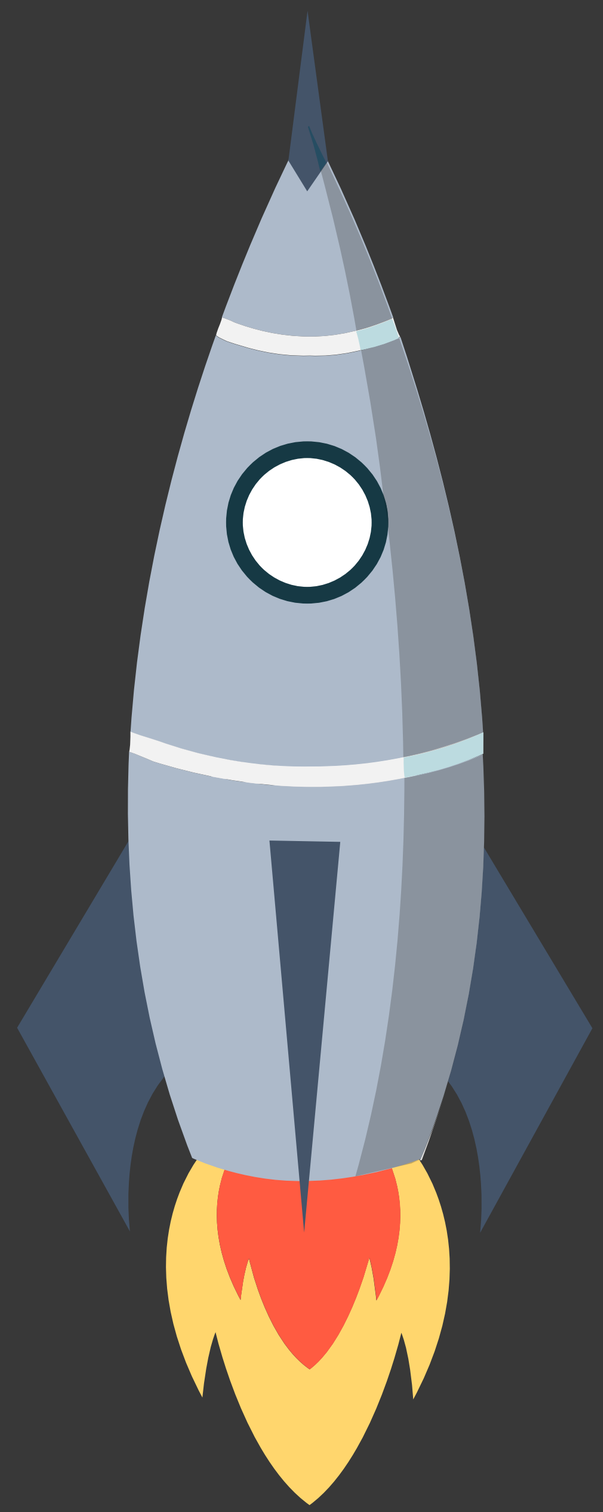
NCP : Networking, Communications, Public

- Extreme rugged systems for harsh environments
- High performance computing for telecom/Mobile Edge
- Modular architecture for flexibility, dynamic expansion

ADLINK's Advanced Technology Office

ATO

INNOVATING TOGETHER



Mission

Scout

Identify and de-risk technologies that can improve ADLINK competitiveness on the market.

Innovate

Promote technology innovation and accelerate technology adoption across ADLINK BUs.

Evangelise

Promote ADLINK technology vision, excellence and thought leadership.

Win

Accelerate innovation adoption to out-innovate competitors and win market shares.

Areas of Focus

Distributed Computing
Algorithms, architectures and technologies for scalable, fault-tolerant and secure large distributed systems.

Heterogeneous Computing Infrastructure
Improve the abstraction level for programming (distributed) systems leveraging FPGA/GPU/CPUs.

Analytics
Real-Time Analytics spanning from data processing to computer vision.

Fog Computing
Unified abstraction for compute, storage and networking compatible with the non-functional requirements of IIoT systems (i.e. real-time).

Low Power Network and Devices
Protocol research for low-power network and constrained devices.

Recent Awards

Data Economy is proud to run for the first time the Power 200, a biannual global list showcasing the 200 personalities around the world who are leading the data centre, cloud and data sectors through charting new innovations or technological breakthroughs, sheer investment or business acumen, or exceptional entrepreneurial skill sets.

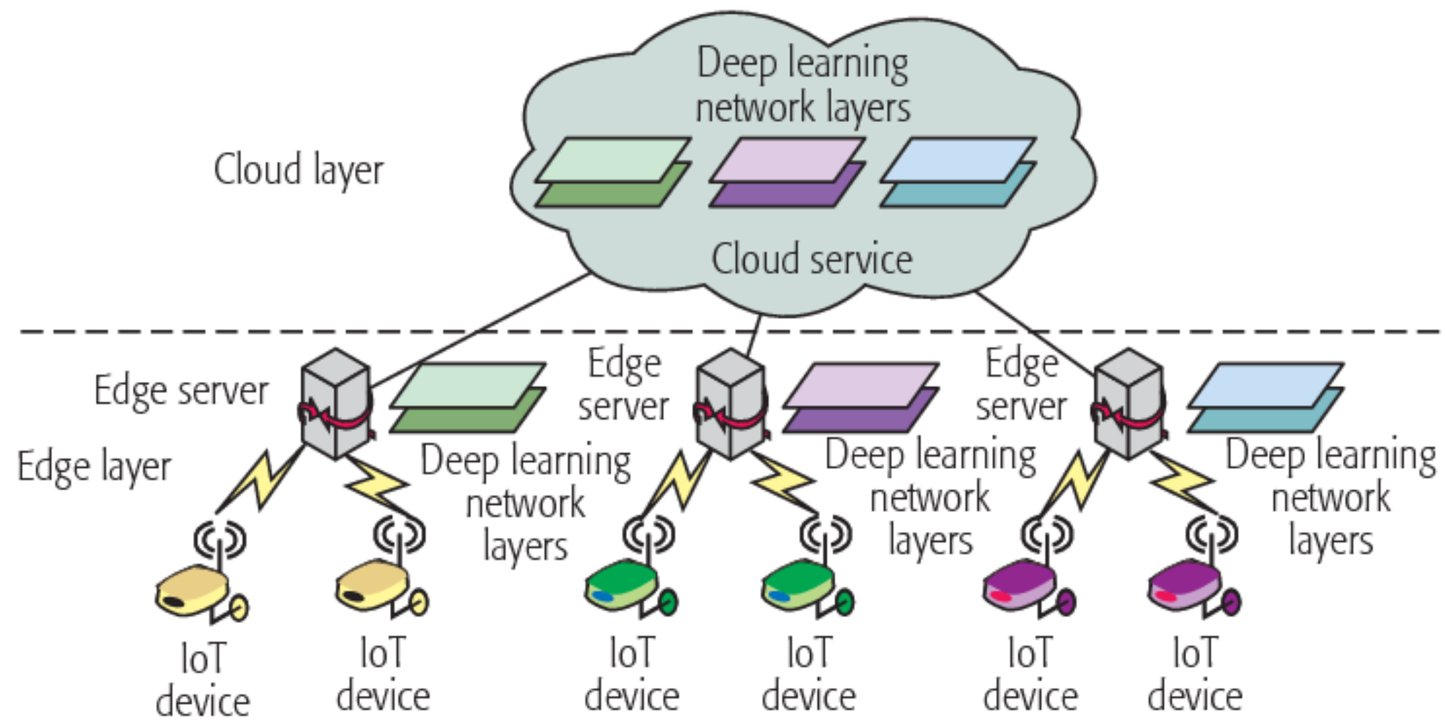
Data Economy is proud to run for the first time “#Edge50”, showcasing the 50 personalities who are leading edge computing through charting new innovations or technological breakthroughs, sheer investment or business acumen, or exceptional entrepreneurial skillsets.

Edge Intelligence?

Edge Intelligence

Edge Intelligence is often associated with the execution of compute intensive AI algorithms – such as DNN – at the edge

This is just one of many cases, not all of which requires the same computing capabilities as DNN.



Edge vs Intelligent Edge

The main difference between Edge and Intelligent Edge is about the kind of applications deployed at the edge.

This give raise to **two orthogonal problems**:

- **Edge Infrastructure**
- **Edge-friendly AI Algorithms** (many exist already)

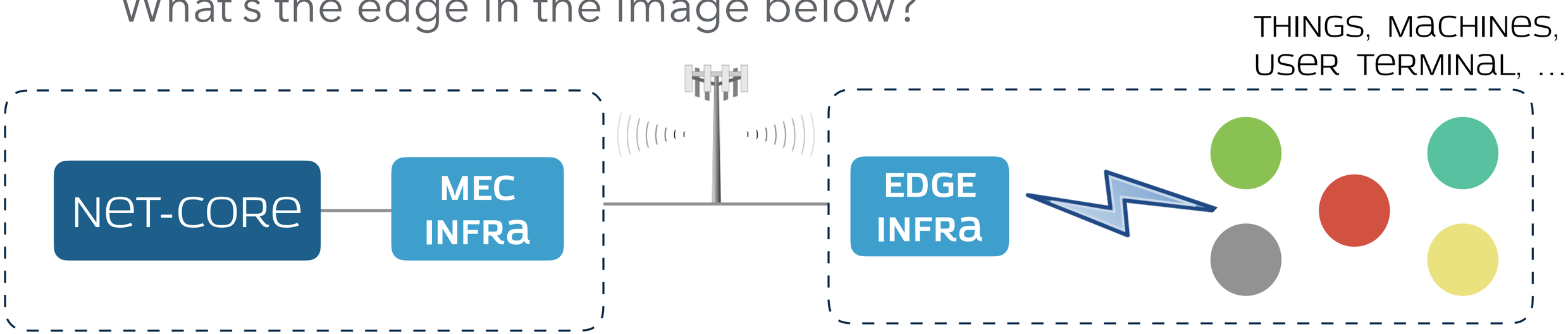
The first problem need to be solved in order to have any kind of Edge, Intelligent or not.

Edging the Edge

The Fuzzy Edge

The edge is an extremely fuzzy concept as it depends entirely from infrastructure ownership structure and application domain.

What's the edge in the image below?



Hamletic Moment

To edge, or not to edge: that is the question:
Whether 'tis nobler in the mind to suffer
The slings and arrows of outrageous boundaries,
Or to take arms against a sea of edges,
And by opposing end them?

Reflection Time...

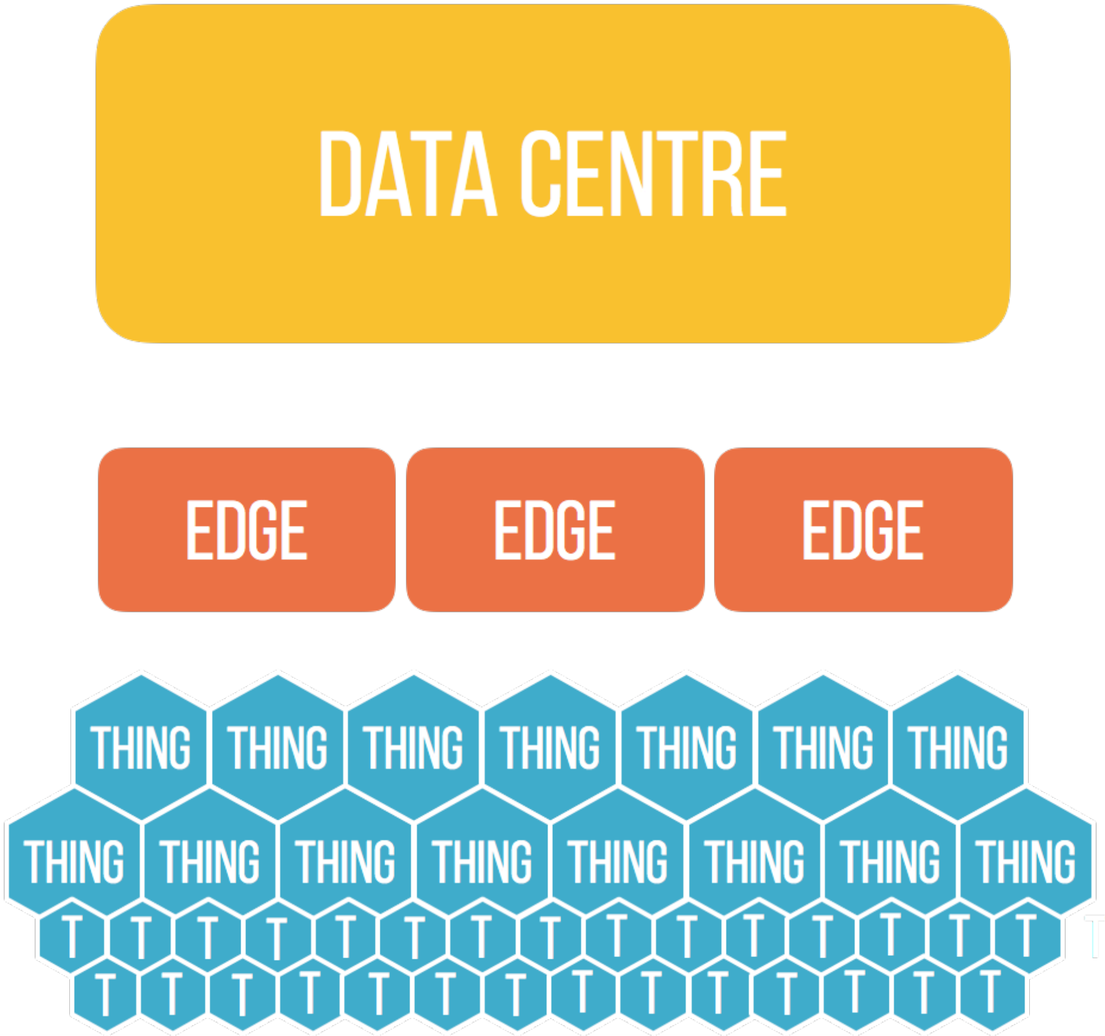
Hardware Tiers in (I)IoT / MEC

A generic IoT/IIoT/MEC system has **three** different **hardware tiers**

Off-premises data-centre which may be private or public

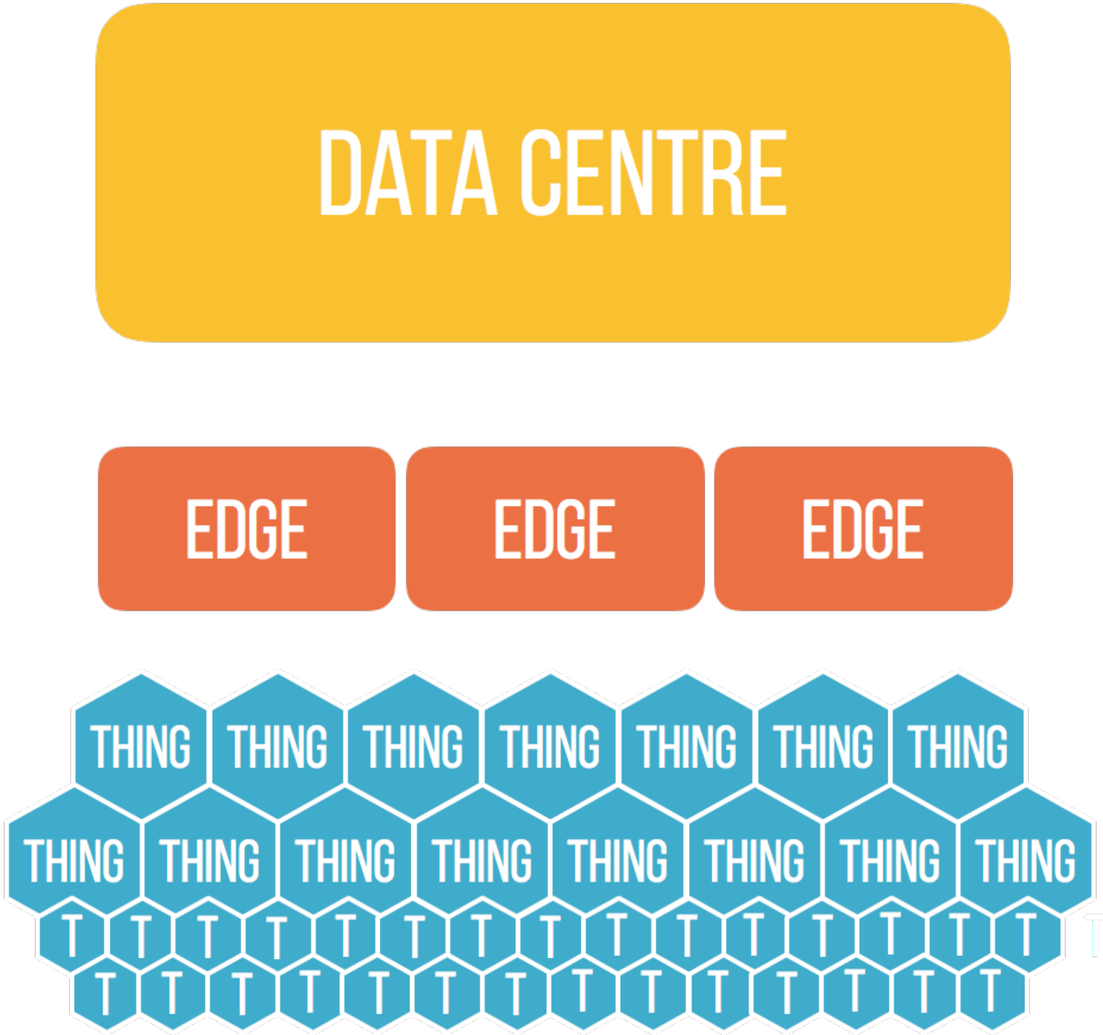
On-premises edge infrastructure

Things with computational, communication and storage capacity



Hardware Tiers in (I)IoT / MEC

The key architectural variations that are discussed today all depends on the **bias**, **or lack of thereof**, on a **specific tier**



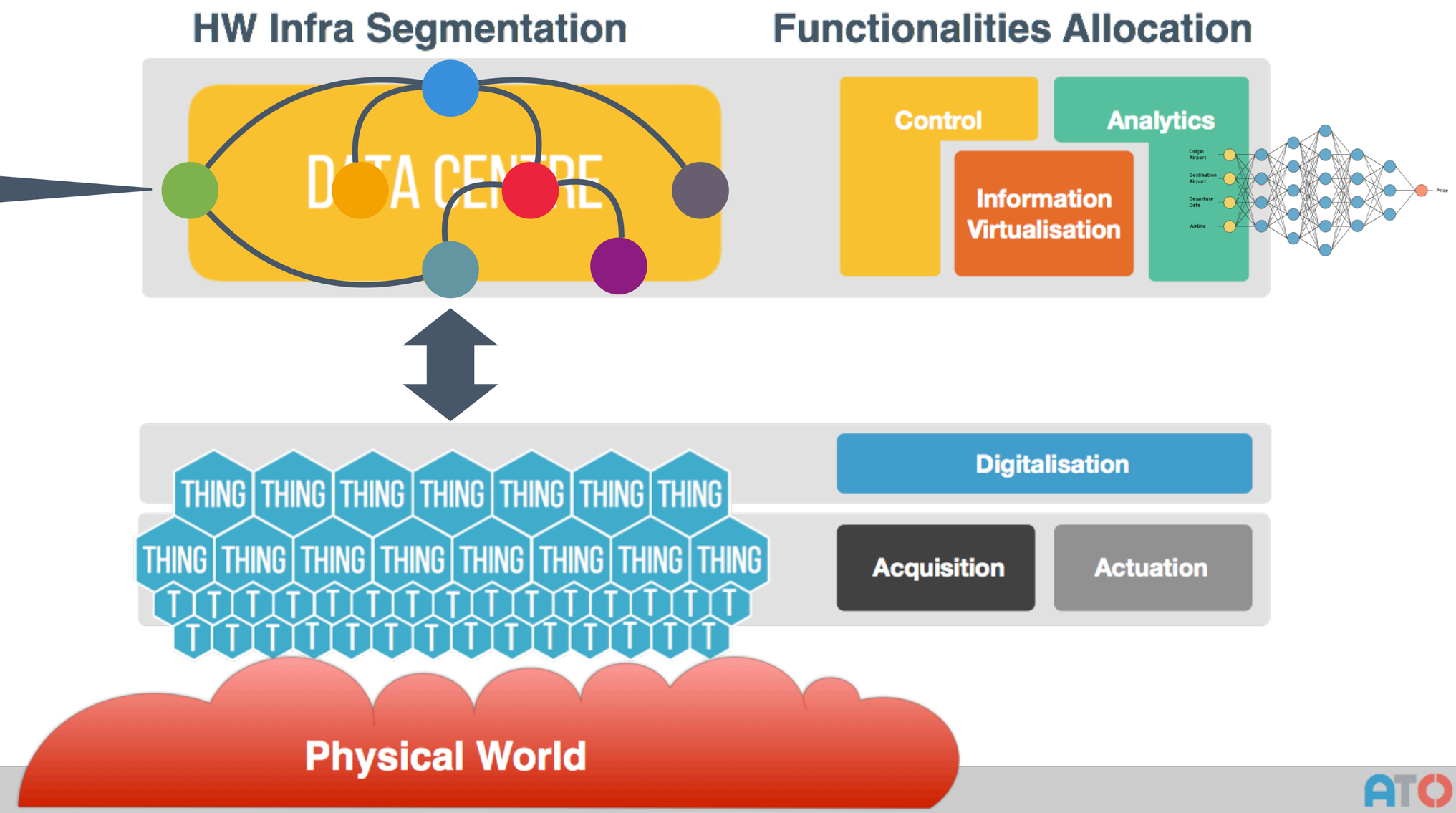
Cloud-Centric Perspective

The cloud infrastructure is **mature** and **operationally convenient**…

Yet **cloud centric architectures don't fit well** for a **large class** of **IoT/IIoT applications**

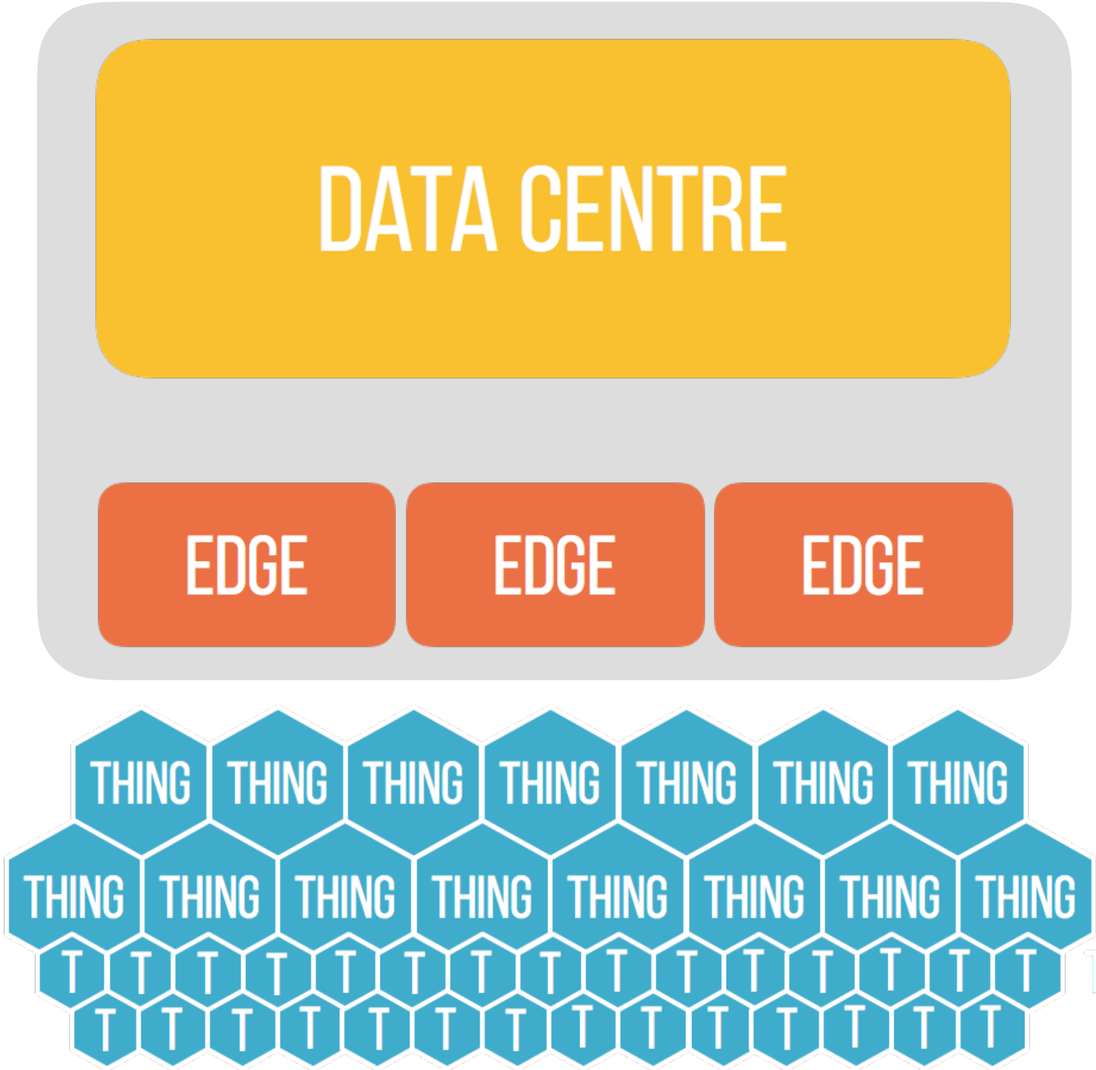
Cloud Centric Perspective

The application is deployed, managed and monitored using the Cloud IaaS infrastructure



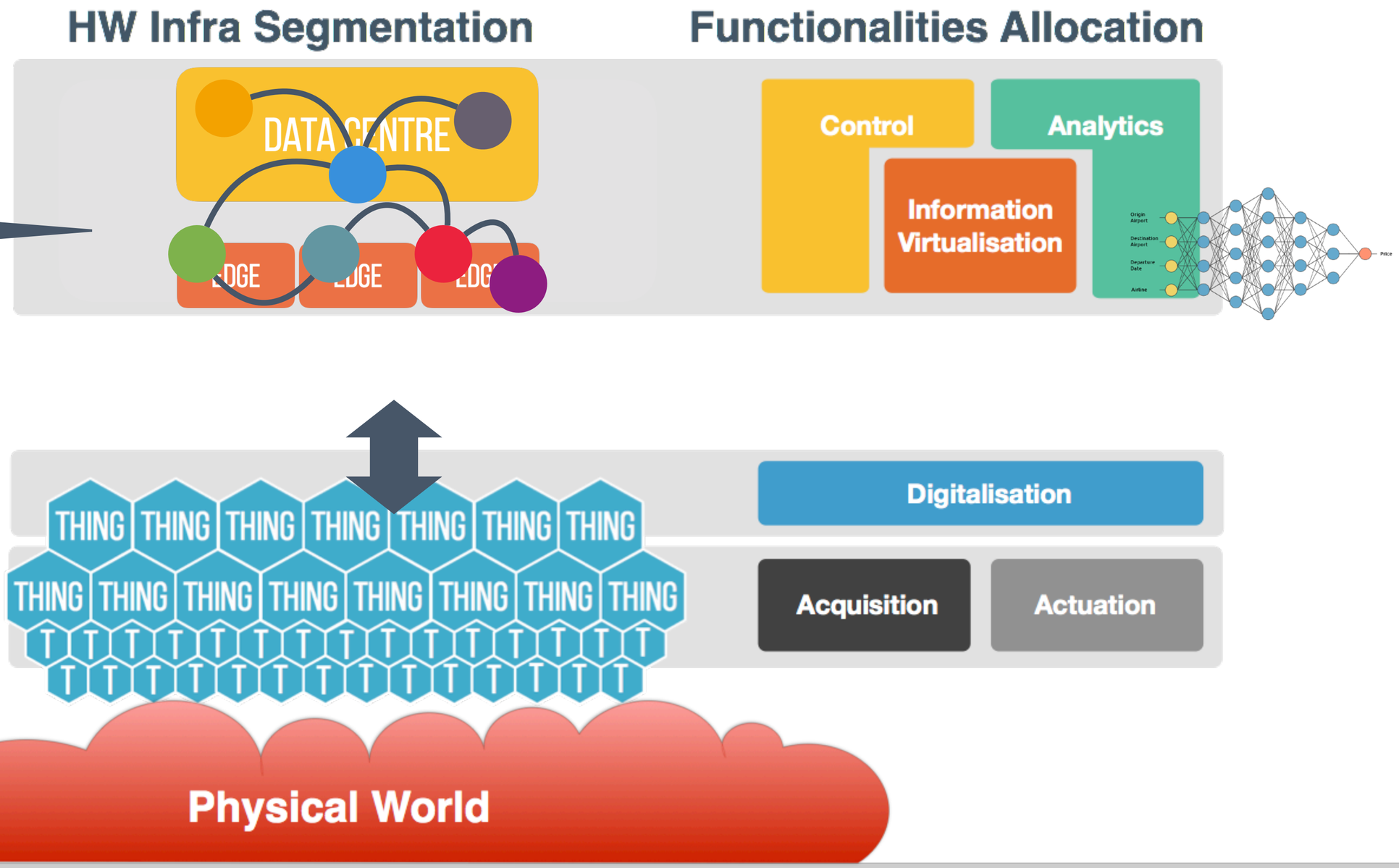
Edge-Centric Perspective

The main idea of Edge-Centric architecture is that of providing edge-clouds to reduce some of the short-comings of traditional Cloud Centric architectures



Edge-Centric Perspective

The application is deployed, managed and monitored using the Cloud IaaS infrastructure which cooperates with edge Cloud infra.



Cloud, Edge, and Things

Cloud Computing gives **operationally convenient abstractions** and tools to **manage** and **provision data-centre resources**

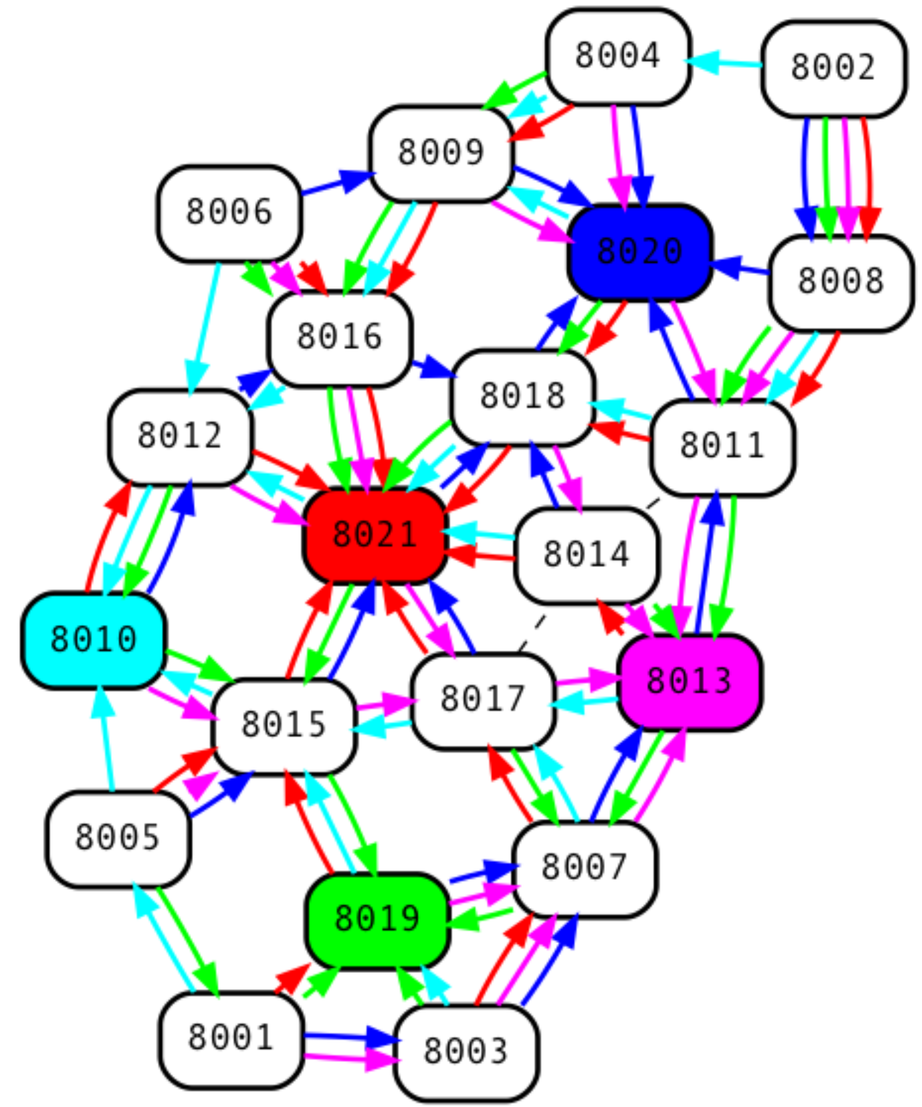
Edge Computing alleviates some of the challenges posed by cloud computing at the cost of introducing some fragmentation in the infrastructure.

But how about the **Things**? In a large class of system we need to manage and provision them too…

What's Really Needed?

1

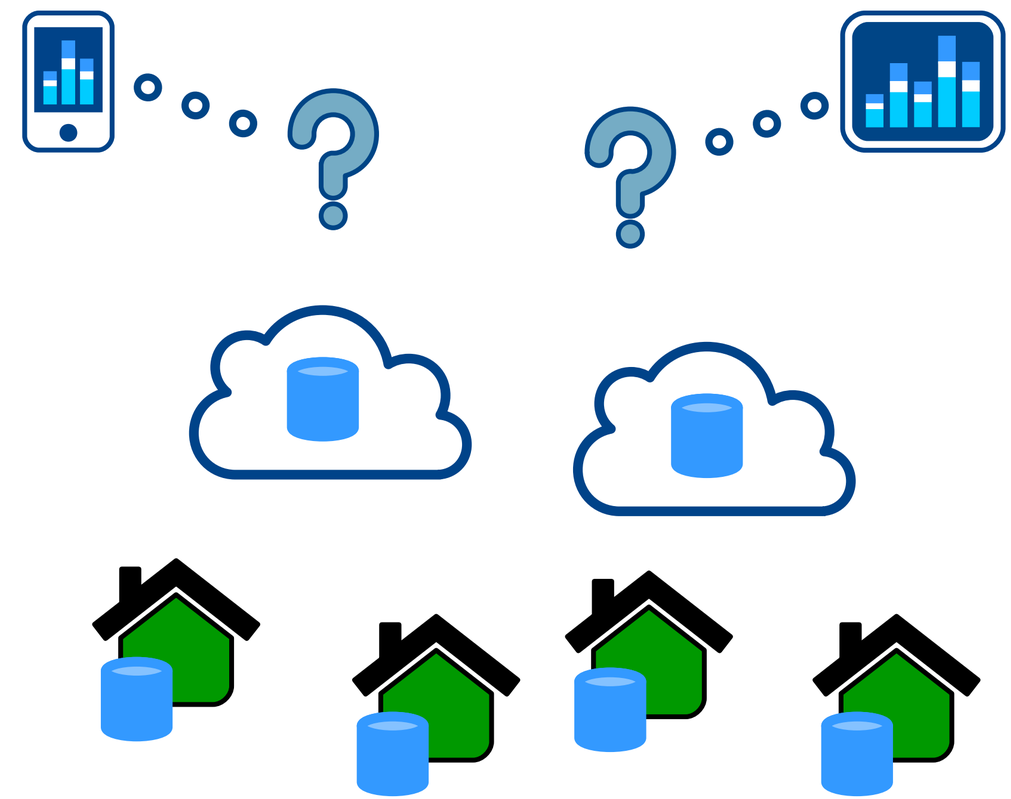
A scalable, location transparent, data-centric layer that allows us to effectively get the data where needed while minimising resource usage



What's Really Needed?

2

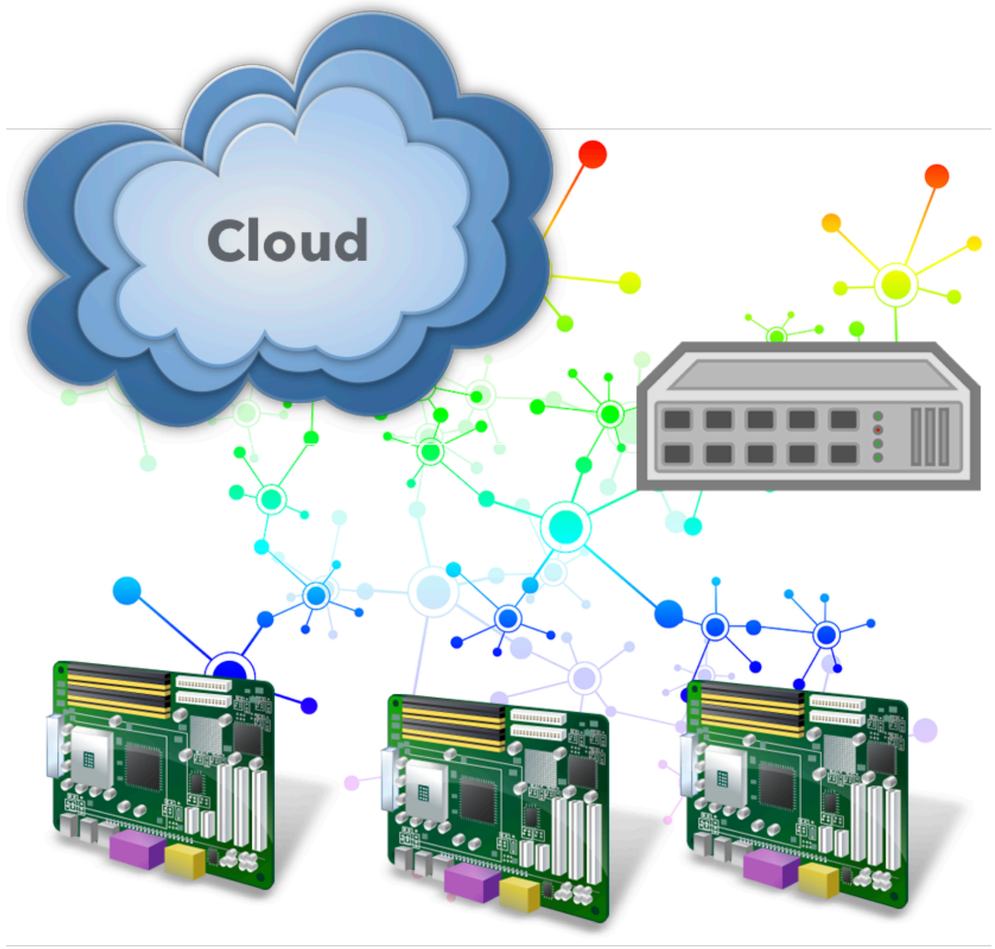
A geo-distributed, location transparent, storage infrastructure that allows us to store data where it makes sense to support local computing while maintaining location transparent access to it.



What's Really Needed?

3

An infrastructure that allows us to federate compute, storage, I/O and communication resources regardless of their location (Fog Infrastructure)

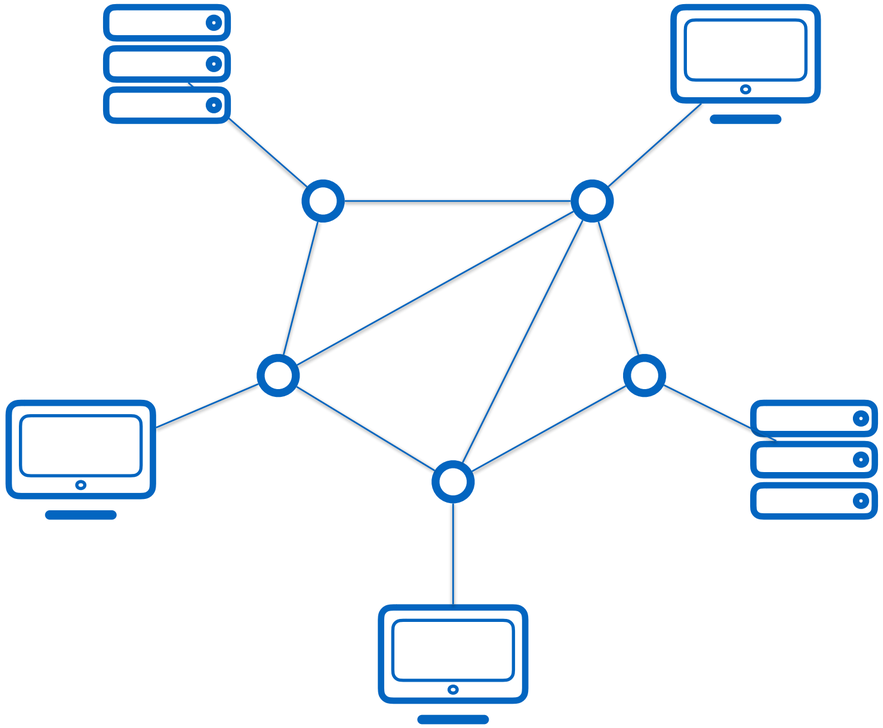


Filling the Gap

Connectivity

Internet protocol

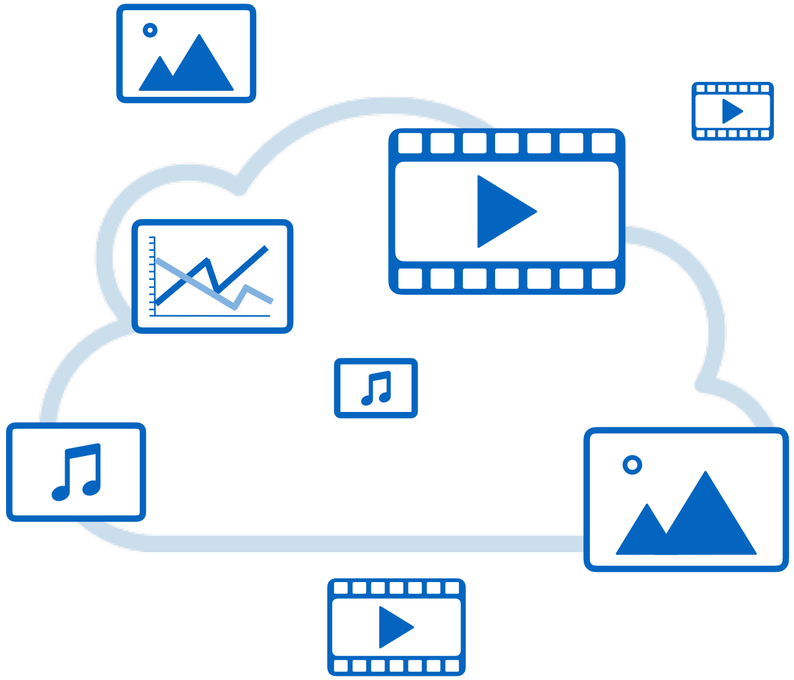
For historical reasons, internet has been built on a **host-centric** communication model. (**machine-to-machine**)



The Internet today

Most of the application on the internet today are **data / content centric**.

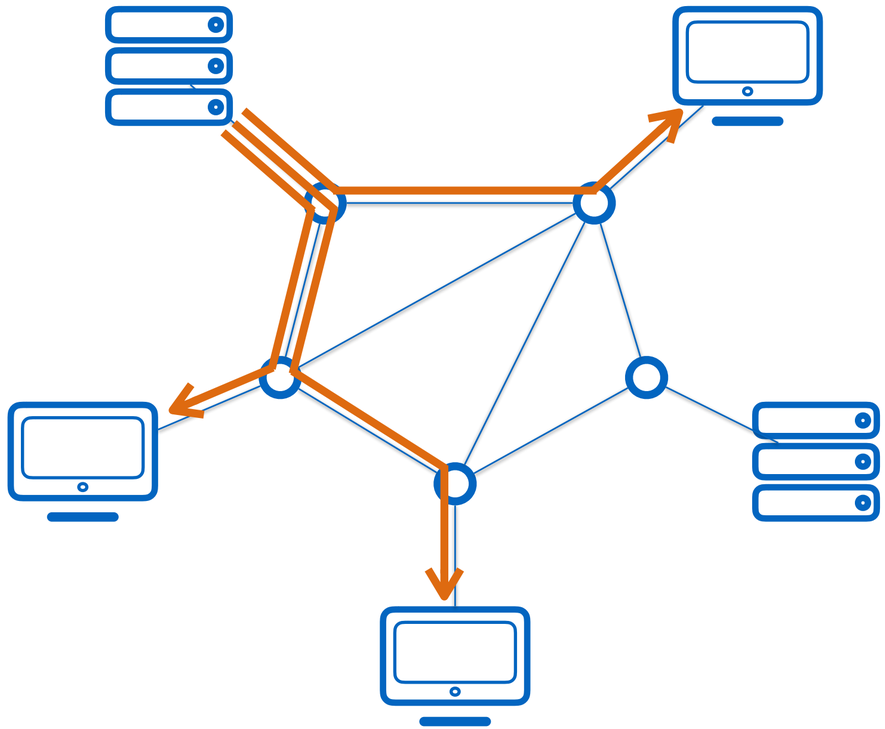
What matters to the user is the data not as much who has it…



Internet protocol

The internet protocol is inherently **one-to-one.** Broadcast and multicast communications are not viable in wide networks.

Thus the **diffusion** of the same data to multiple consumers is very **inefficient**.



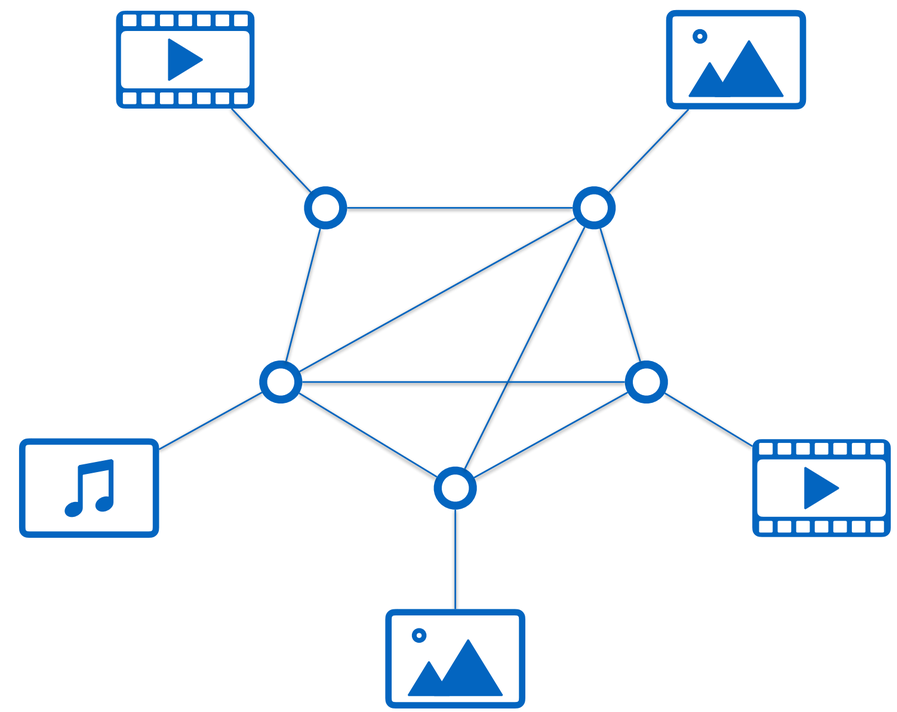
NDN / CCN

Named Data Networking
Content Centric Networking

NDN is based on a **data centric** networking paradigm.

Data samples are identified with **hierarchical names**.

NDN is Inherently **Pull** and best suited for **static data**.



`/com.adlink/fr/employees/olivier.hecart`

Data Distribution Service

DDS is a great **data centric** technology which embraces powerful concepts like **strong decoupling** between publishers and subscribers.

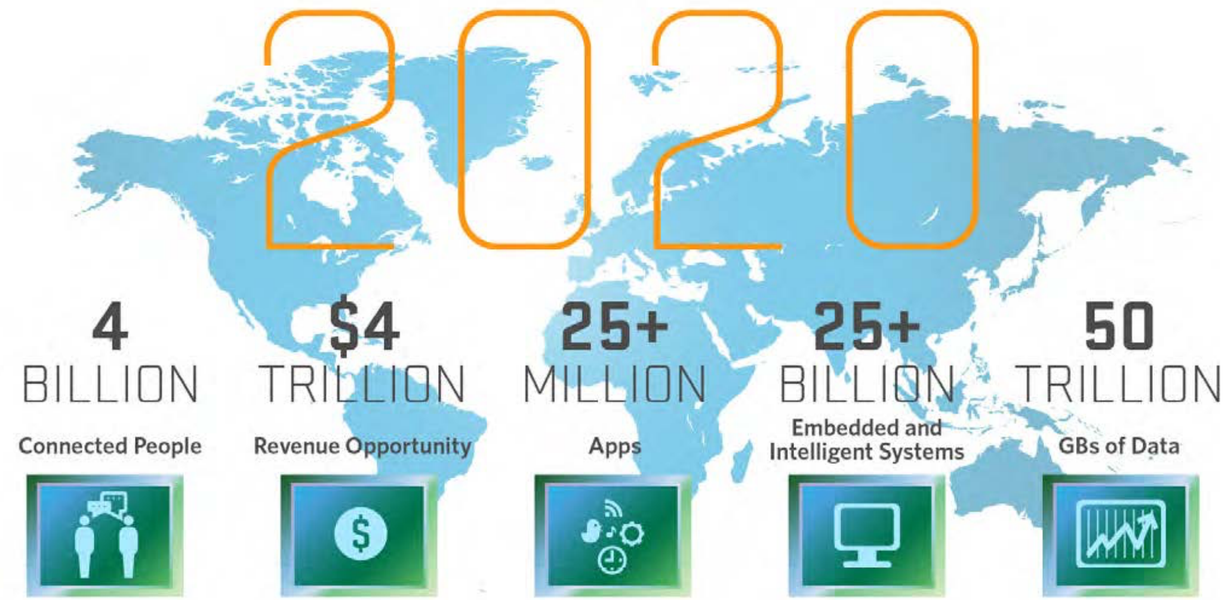
But DDS has been designed for small to medium systems and suffers from major **scalability** issues on larger systems.

It is **push** based which makes uneasy to **retrieve specific data** or to properly **filter** data streams.

The Internet of Tomorrow

With the raise of IoT, the different devices connected to the internet 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.



Internet scale data-centric protocol that unifies **data-sharing** between **any kind of device** including those **constrained** with respect to the **node resources**, such as **computational resources** and **power**, as well as the **network**.

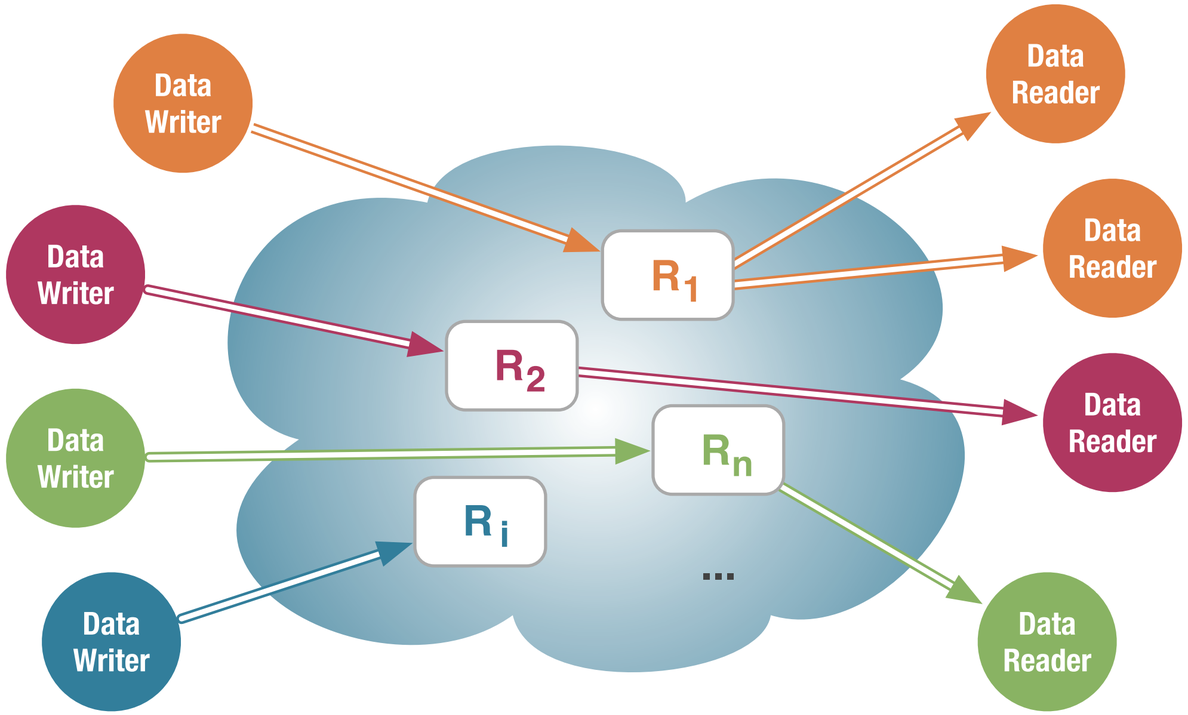
zenoh

the zero network over-head protocol

Conceptual Model

zenoh provides a **data-centric** abstraction in which **applications** can **read** and **write data** **autonomously** and **asynchronously**.

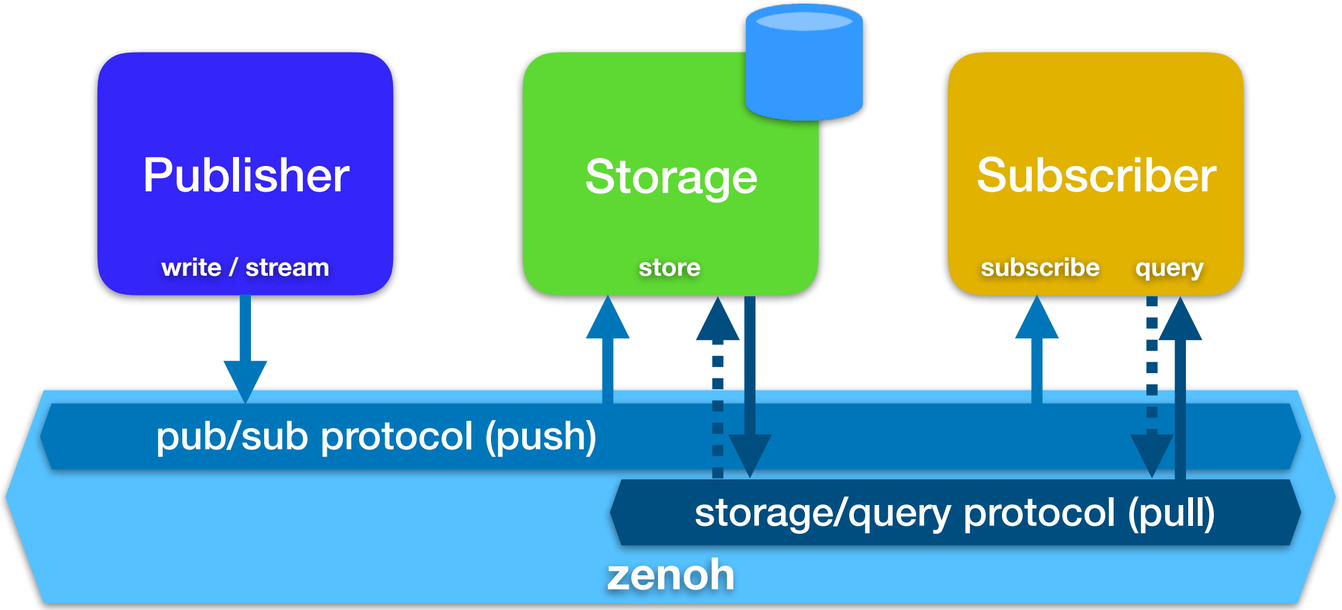
The **data read** and **written** by zenoh applications is associated with one or more **resources** identified by a **URI**.




```
-- These are Resources
/myhouse/floor/1/musicroom/LightStatus
/myhouse/floor/2/musicroom/LightStatus
/myhouse/floor/2/bedroom/erik/LightStatus

-- These are Selections
/myhouse/floor/2/bedroom/*/LightStatus
/myhouse/**/LightStatus
/myhouse/**
```

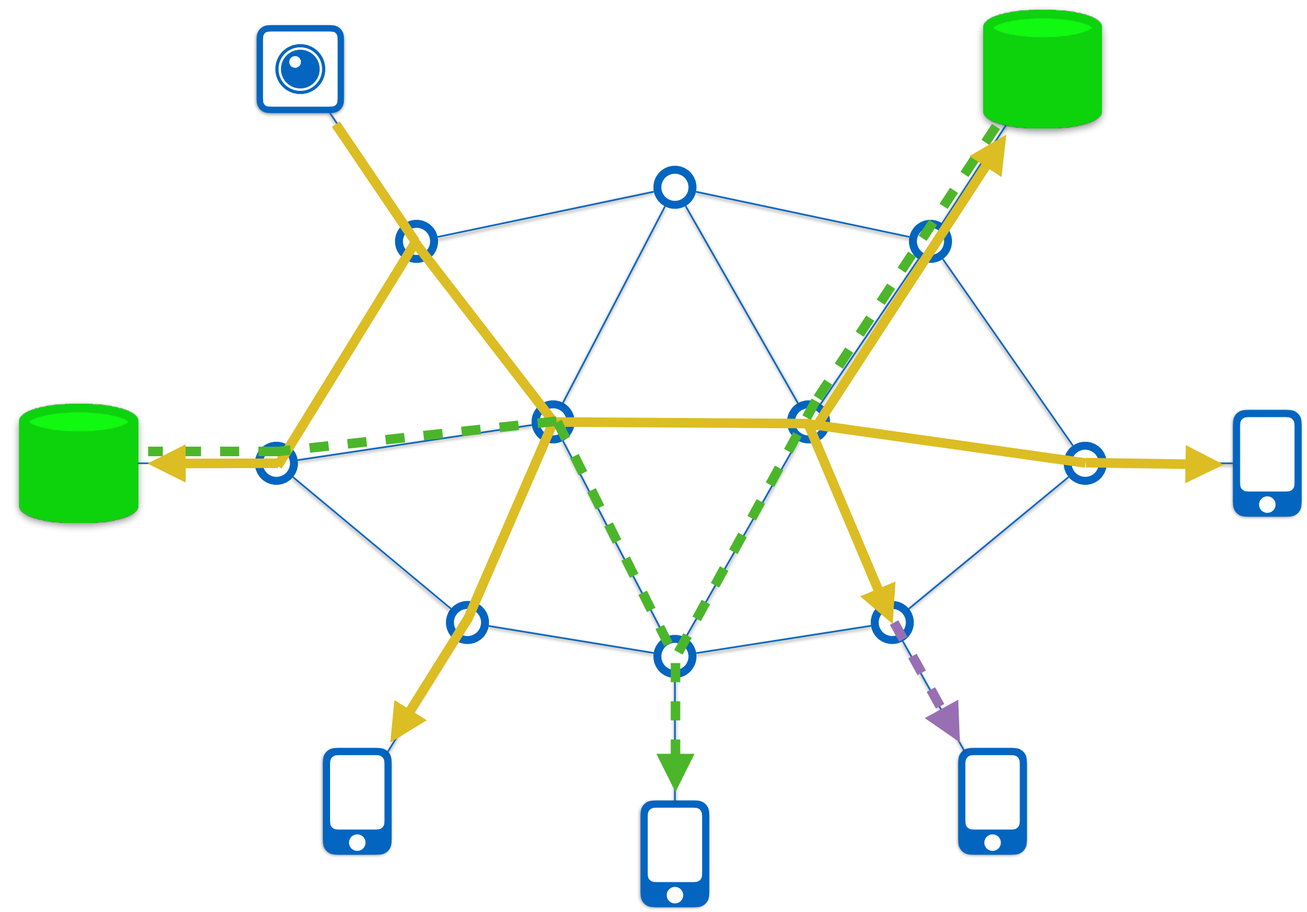
Conceptual Model

Data can be **pushed** to **subscribers** and **storages** and be **queried** from **storages**.



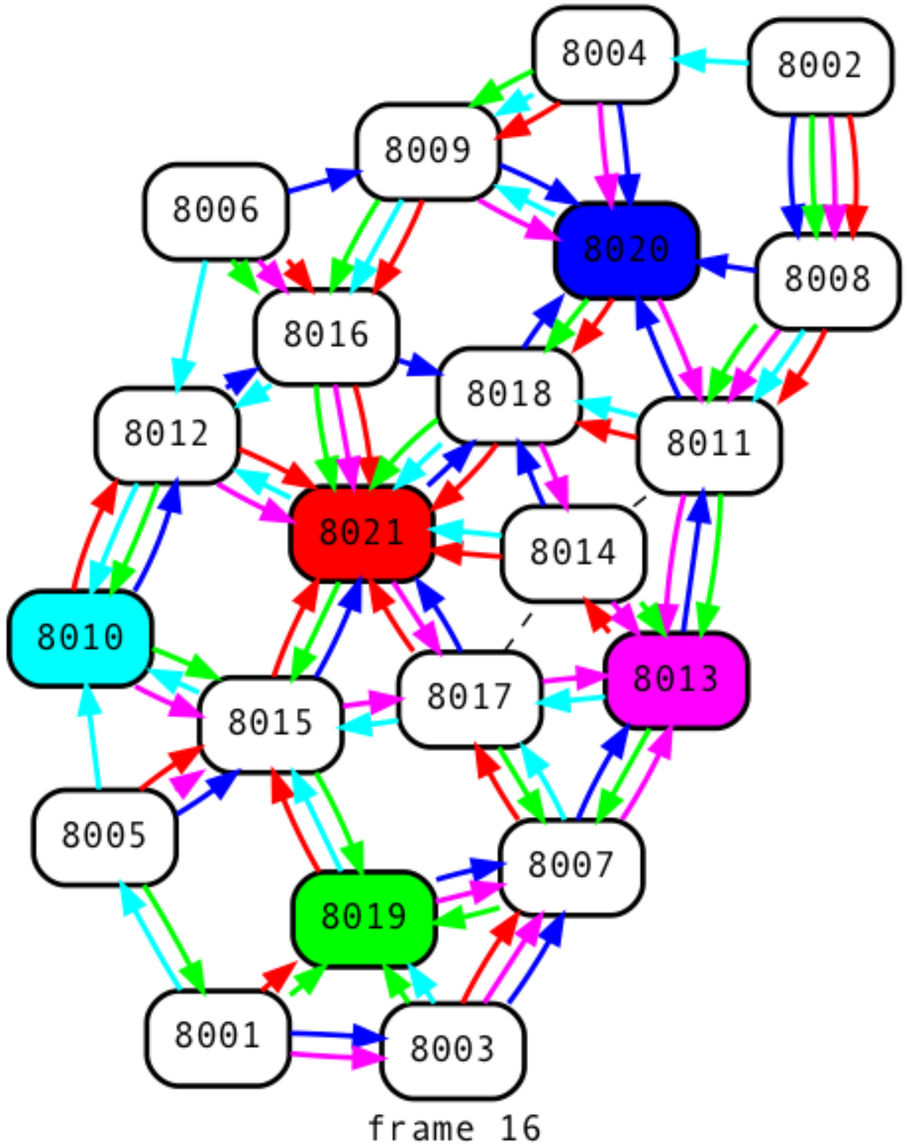
Conceptual Model

Data can be **pushed-to**, **pulled** or **queried-from** applications periodically or asynchronously.



Routing

Adaptative,
fault tolerant,
brokering and **routing**.



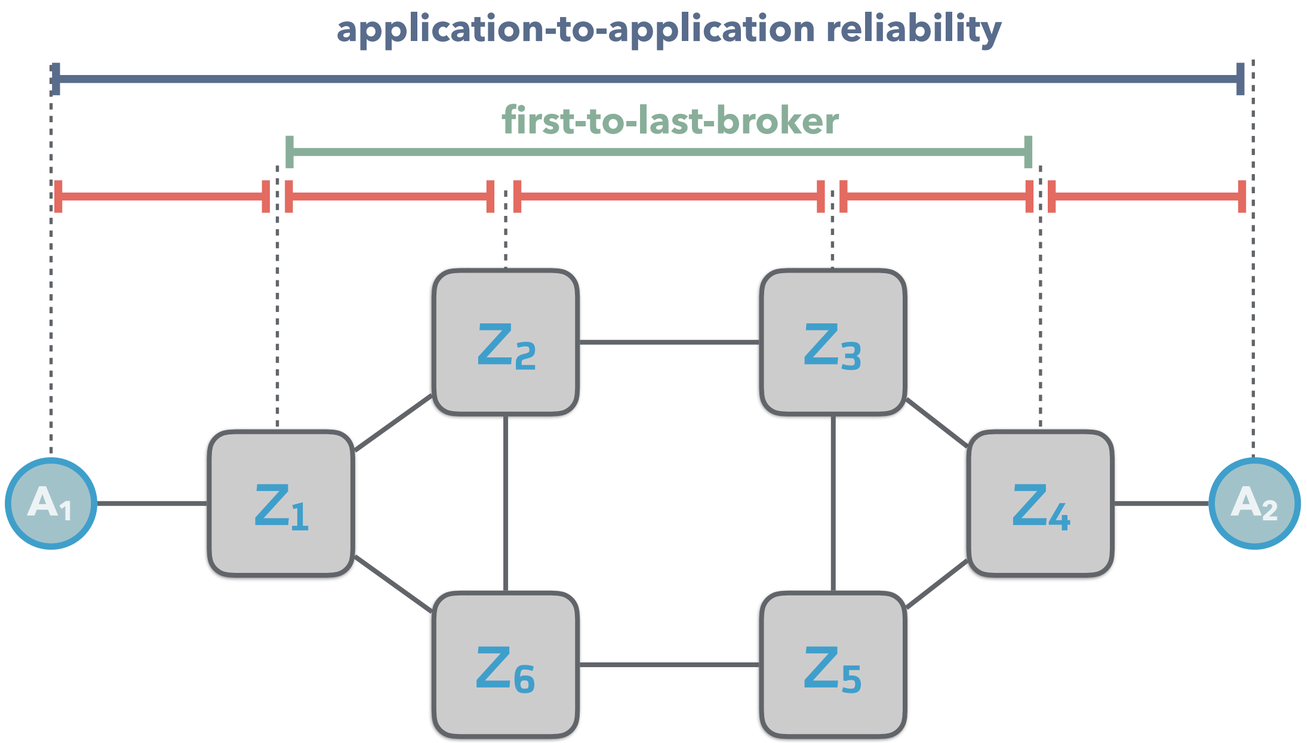
Reliability & Ordering

Zenoh supports 3 levels of reliability :

- **Hop to hop reliability**.
  Ensures reliability and ordering when NO failures.

- **App-to-app reliability**.

- **First-to-last-broker reliability**.
  More scalable than app-to-app reliability.



Zenoh clients

- Low footprint C client
- OCaml client
- Python client
- Bash client

A piece of code

Publisher :

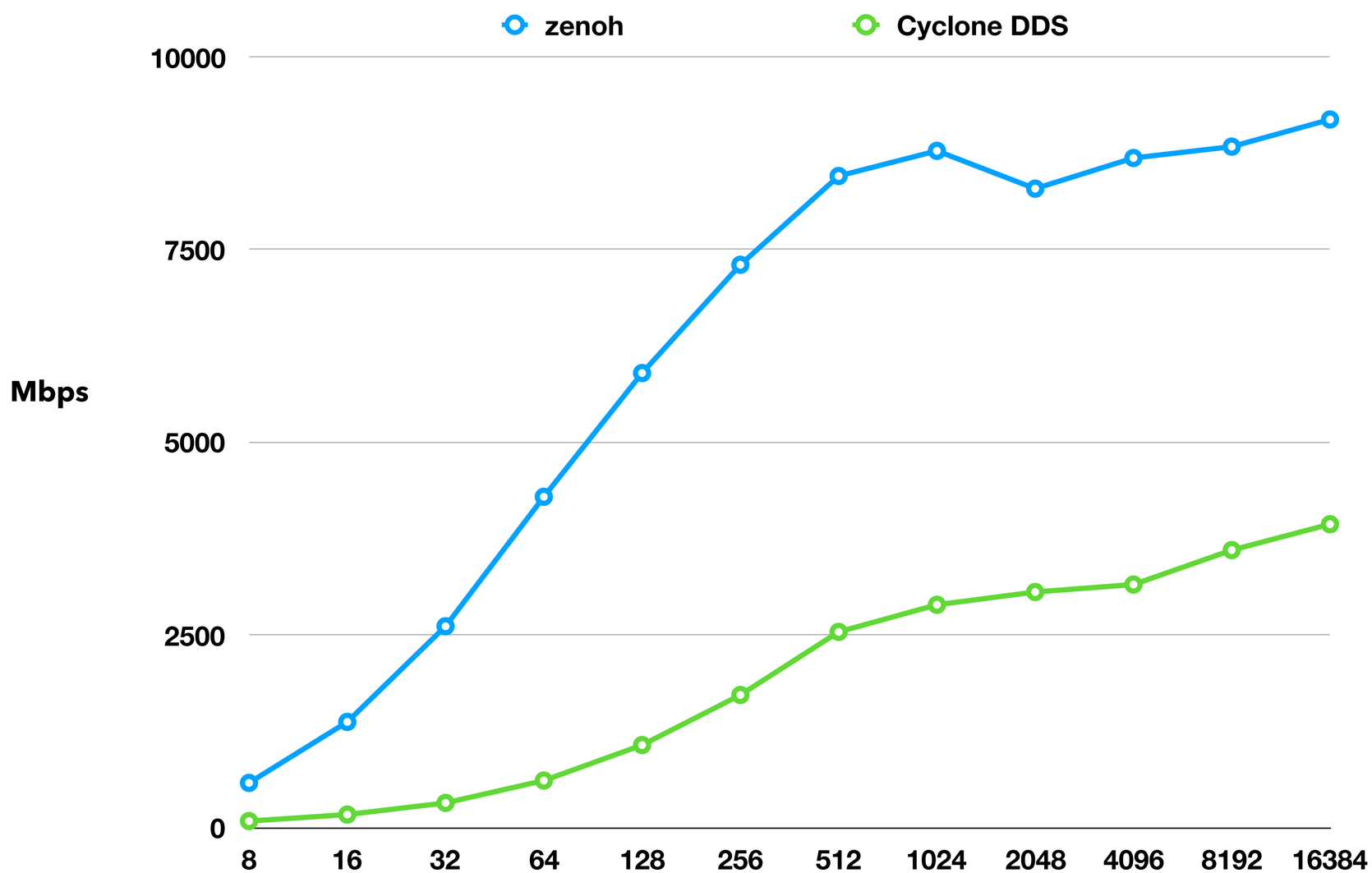
```
z = zenoh.connect('127.0.0.1:7447')
pub = z.declare_publisher('/demo/hello')
pub.write('Hello world'.encode())
```

Subscriber :

```
z = zenoh.connect('127.0.0.1:7447')
z.declare_subscriber('/demo/**', lambda rid, data:
    print('received {}'.format(data.tobytes().decode())))
```

Performance

Throughput


zenoh
Cyclone DDS
Mbps
10000
7500
5000
2500
0
8
16
32
64
128
256
512
1024
2048
4096
8192
16384

Key Highlights

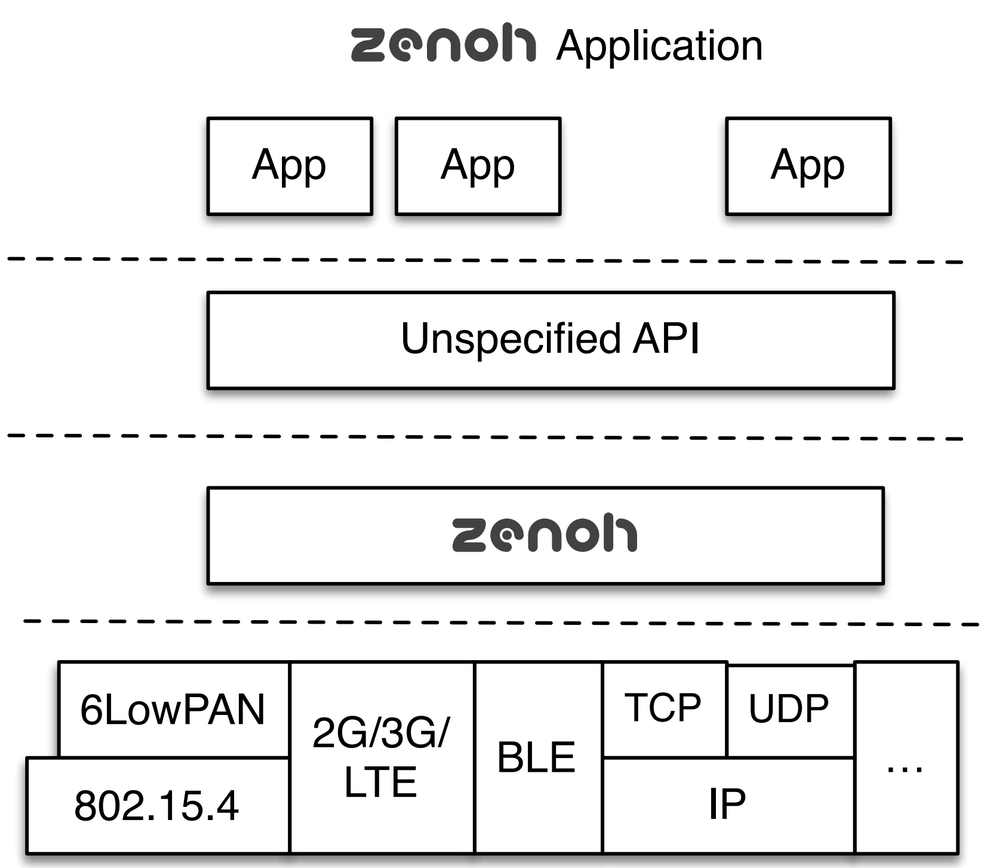
Key Highlights

Extremely Resource Constrained Environments

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

Support for:

- **Peer-to-peer** and **brokered communication**
- **Batched data and deltas**
- **Ordered reliability and fragmentation**
- **Queries**



Key Highlights

Protocol implementation for a **8-bit micro-controllers** takes **300 Bytes of RAM** and has **wire-overhead** of **4** bytes for **data samples**

```
 7 6 5 4 3 2 1 0
+-+-+-+-+-+-+-+-+
|R|S|P| SDATA   |
+---------------+
~       SN      ~
+---------------+
~   RID | SID   ~
+---------------+
~      PRID     ~
+---------------+
~   [Payload]   ~
+---------------+
```

Storage

Home/Building Management

Imagine that we have a collection of houses in a residence or equivalently buildings on a business park that we would like to monitor and manage.

In other terms, we would like to read, write and observe data specific to the house/building.

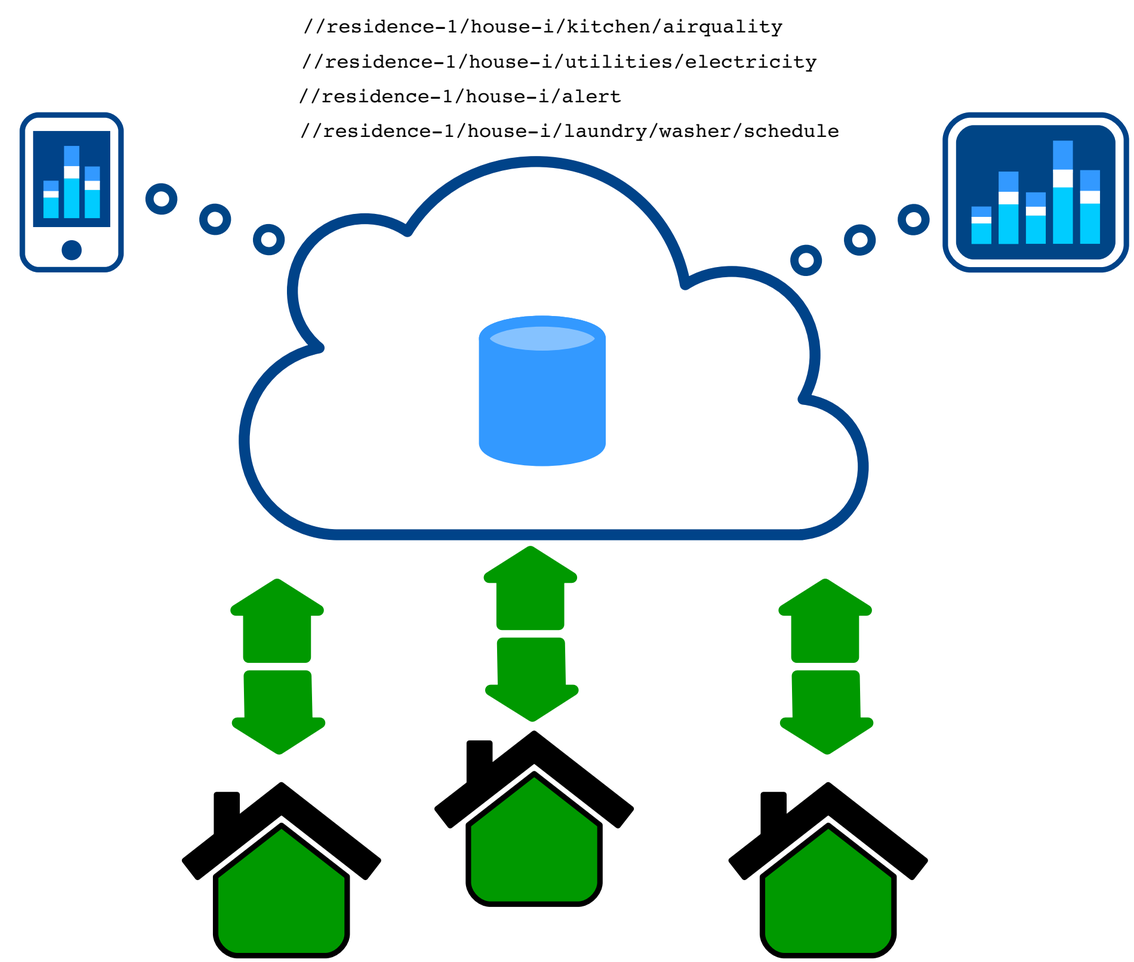
More importantly we would want to do this from anywhere.

Cloud Centric Home/Building Management

One solution is to push the data to the **Cloud**

Applications can use the cloud as the place to go and get or set any information concerning our house/buildings.

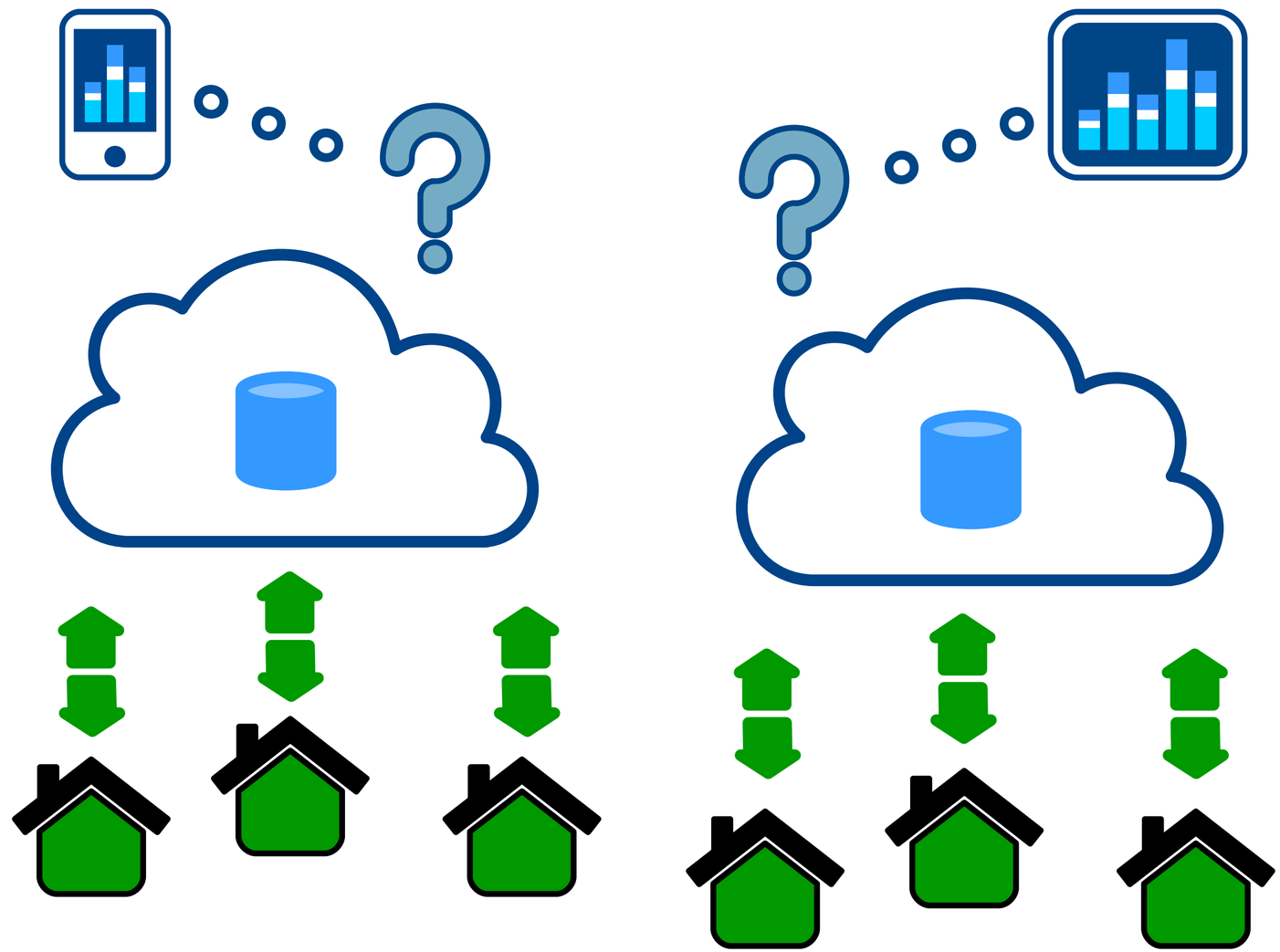
That is a very commont approach… But, is this a solution to the problem, or is it just *delaying the problem*?



Scaling Out

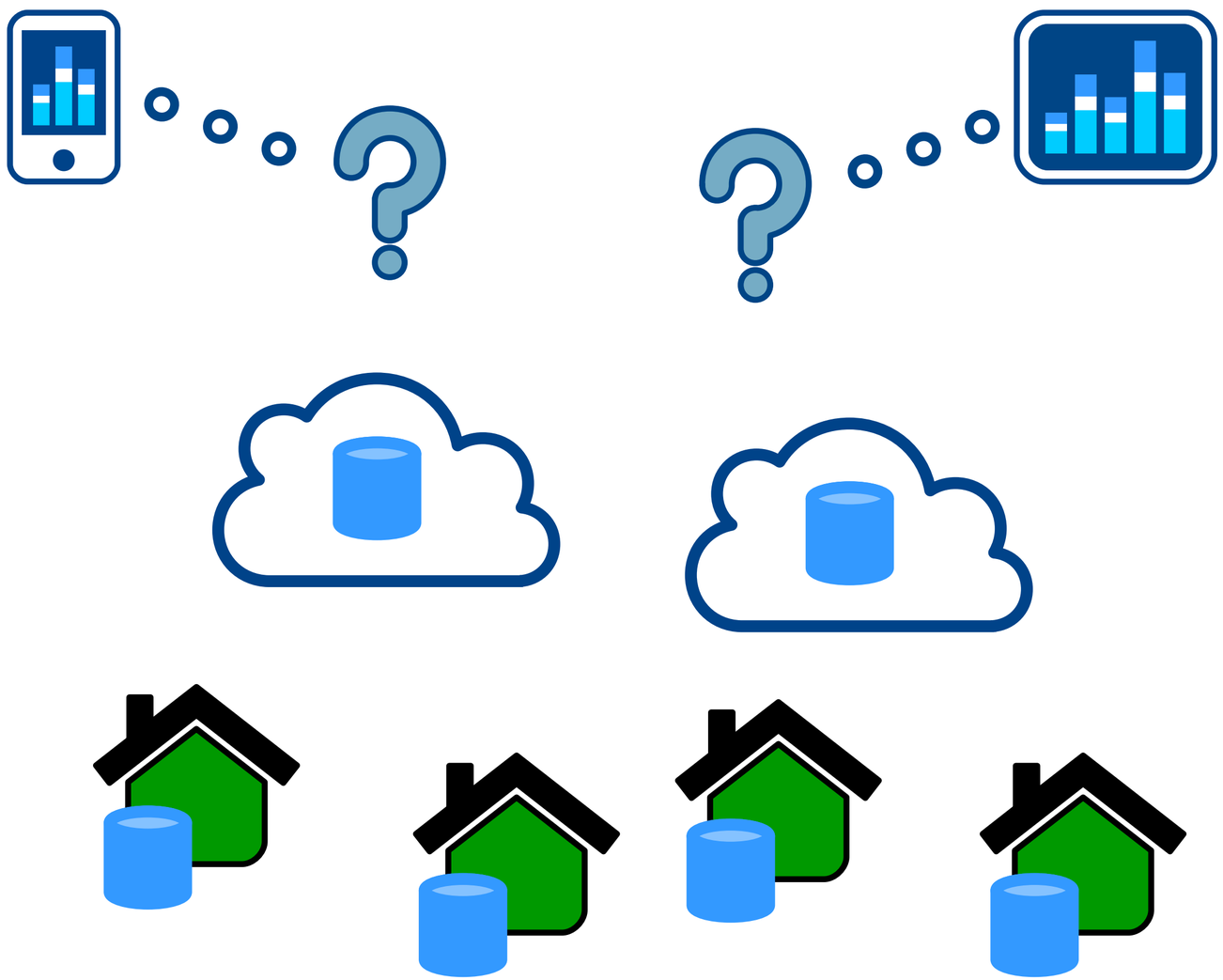
At some point **scaling-up** won't be a solution and we will have to **scale-out** and leverage multiple cloud regions.

With multiple cloud regions the unifed view of the system is lost, and we are back to the starting point.



Home/Building Management – Edge Computing

As we tackle the problem, it would also make sense to address edge computing and ensure that our solution will allow for **location transparent** and **uniform** access to data even if that is *living* on the edge.



Introducing Yaks

YAKS provides a **distributed service** that **implements** an **eventually consistent**, **scalable**, **location transparent, high performance**, and **distributed key/value store** with pluggable back-ends and front-end.

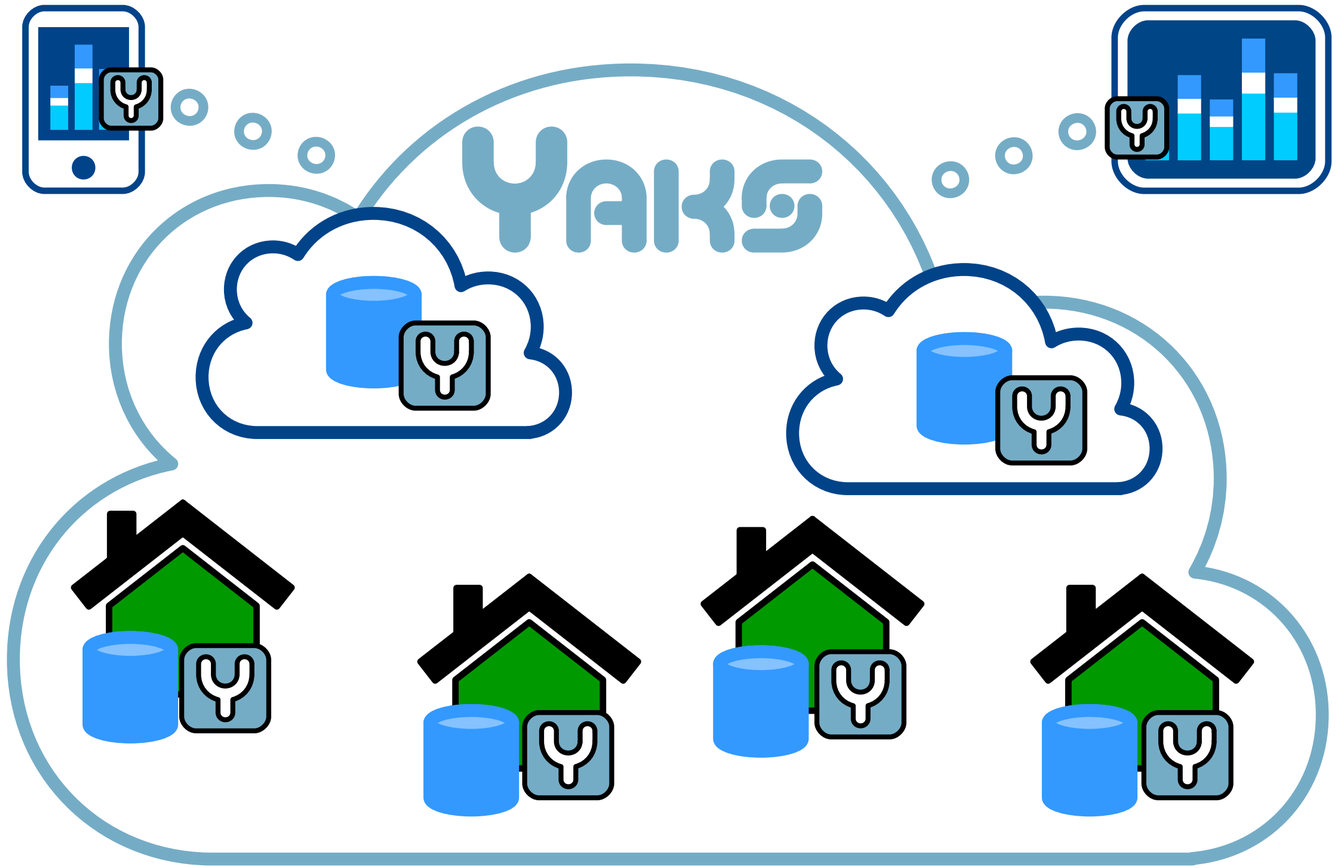
YAKS is equipped with **dynamic discovery** and supports extremely well **dynamic environments**

YAKS **data is globally accessible without requiring local replication** as in traditional key/value stores.

Home/Building Management with YAKS

Regardless of wether data is the device, the edge infrastructure or the cloud, **YAKS** provides location transparent access through a distributed key-value store abstraction.

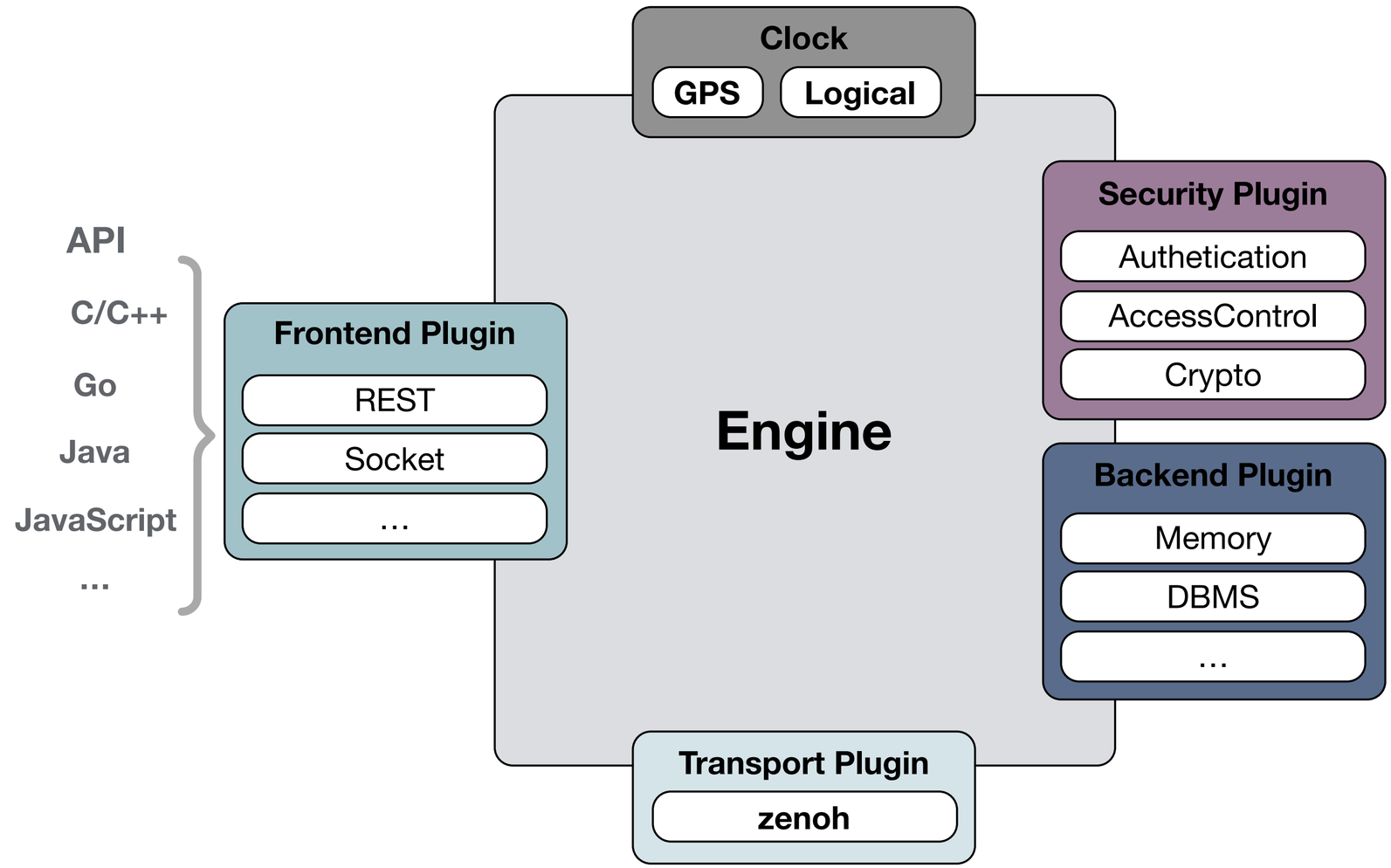
All the details concerning how to get the data from were it is to were it needs to are handled by **YAKS**.



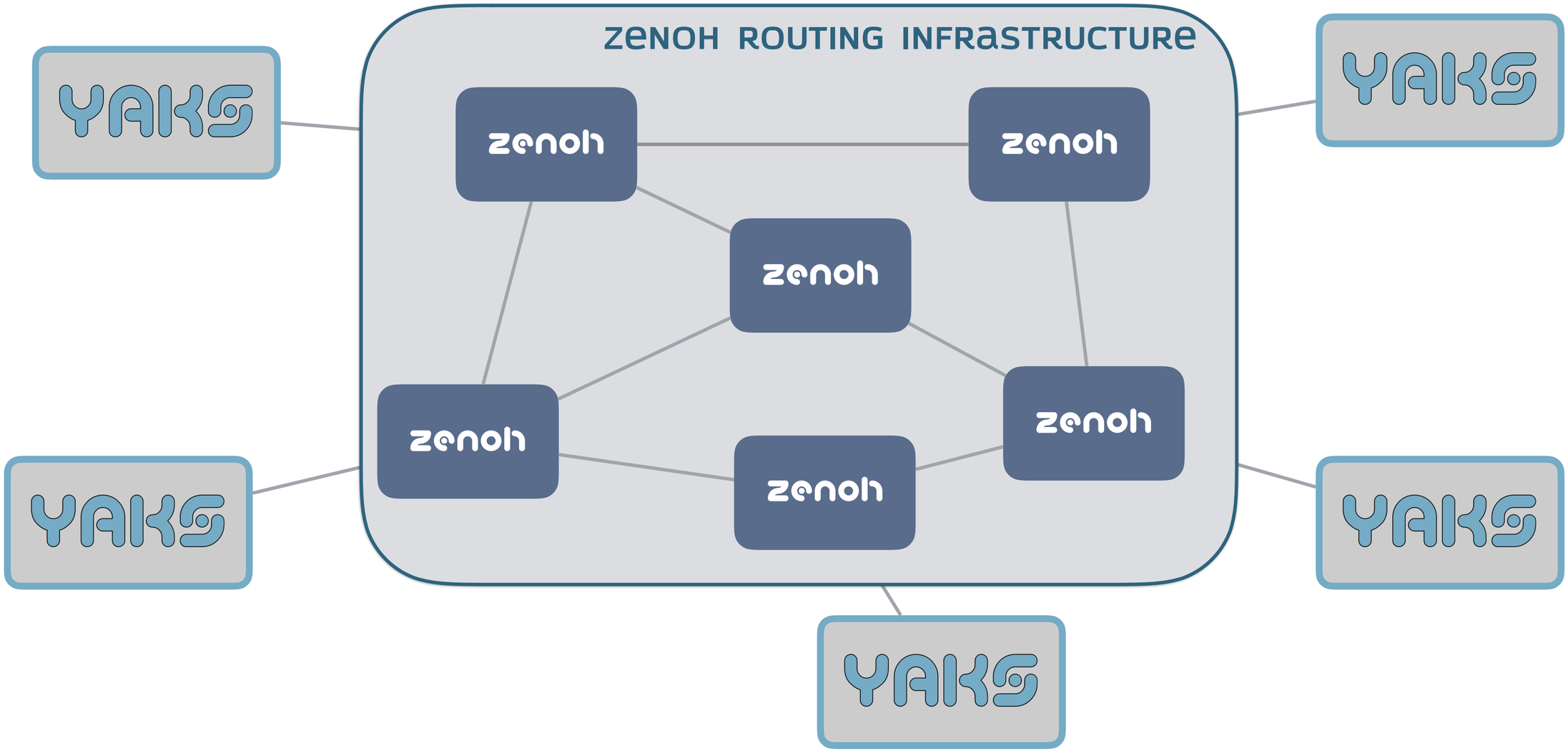
Architecture

YAKS Architecture

YAKS has a modular architecture relying on different kinds of plugins



Deployment Topology



Abstractions

YAKS

YAKS is a **distributed service** to define, manage and operate on key/value spaces

The key abstractions at the core of yaks are **Path**, **Value**, **Selector**, **Storage**, **Workspace**, and **Admin Space**

YAKS Values

A YAKS **Value** is defined by the following tuple:

$$v = \langle e, c, t \rangle$$

Where **e** is the encoding and **c** represents the content and **t** is a logical timestamp used for ordering.

YAKS Path

A **Path** in YAKS is a string having the following format:

$$/s_1/s_2/\ldots/s_n$$

Where **s_i does not contain** wildcard characters such as '*' and '**'

Example:
/com/adlink/factory/shanghai/line/1/machine/
/net/icorsaro/home/livingroom/lightbulb/10

YAKS Selector

A **Selector** in YAKS is the conjunction of an expression identifying a set of keys and optionally a predicate on values

$$/se_1/se_2/\ldots/se_n[?\,[predicate]\,[(properties)]]\,[\#projection]$$

Where:

- **se_i** may contain wildcard characters such as '*****' and '******'
- the **predicate** has the form: f_1 **op** v_1& f_2 **op** v_2&… &f_n **op** v_n
  (where op can be <, <=, =, >=, > and !=)
- the **properties** is a semicolon separated list of ***key=value***
- the **projection** is a semicolon separated list of fields to project.

Example:

/net/icorsaro/home/*/lightbulb?luminosity>50#id

/net/icorsaro/home/*/consumption/statistics?(start=yesterday;end=now)#average;std

Selector / Path matching

'*' to match 1 segment (full or partial):

Examples:

/home/bob/*/light	matches	/home/bob/kitchen/light
/home/bob/room*/light	matches	/home/bob/room1/light
/home/bob/*/light	doesn't match	/home/bob/floor1/room2/light

'**' to match several segments (full):

Examples:

/home/bob/**/light	matches	/home/bob/kitchen/light
		/home/bob/floor1/room2/light
/home/bob/**	also matches	/home/bob/floor2/room1/temp

YAKS Storage

A **Storage** in YAKS is defined by means of a selector **s** and backend **B.** Where the back-end **B** may be one of supported backends, such as main-memory, DBMS, etc.

A **Storage** with selector **s** will store <path,value> for which **s** matches the path.

KV Space Operations

YAKS primitives to operate on the key/value space are:

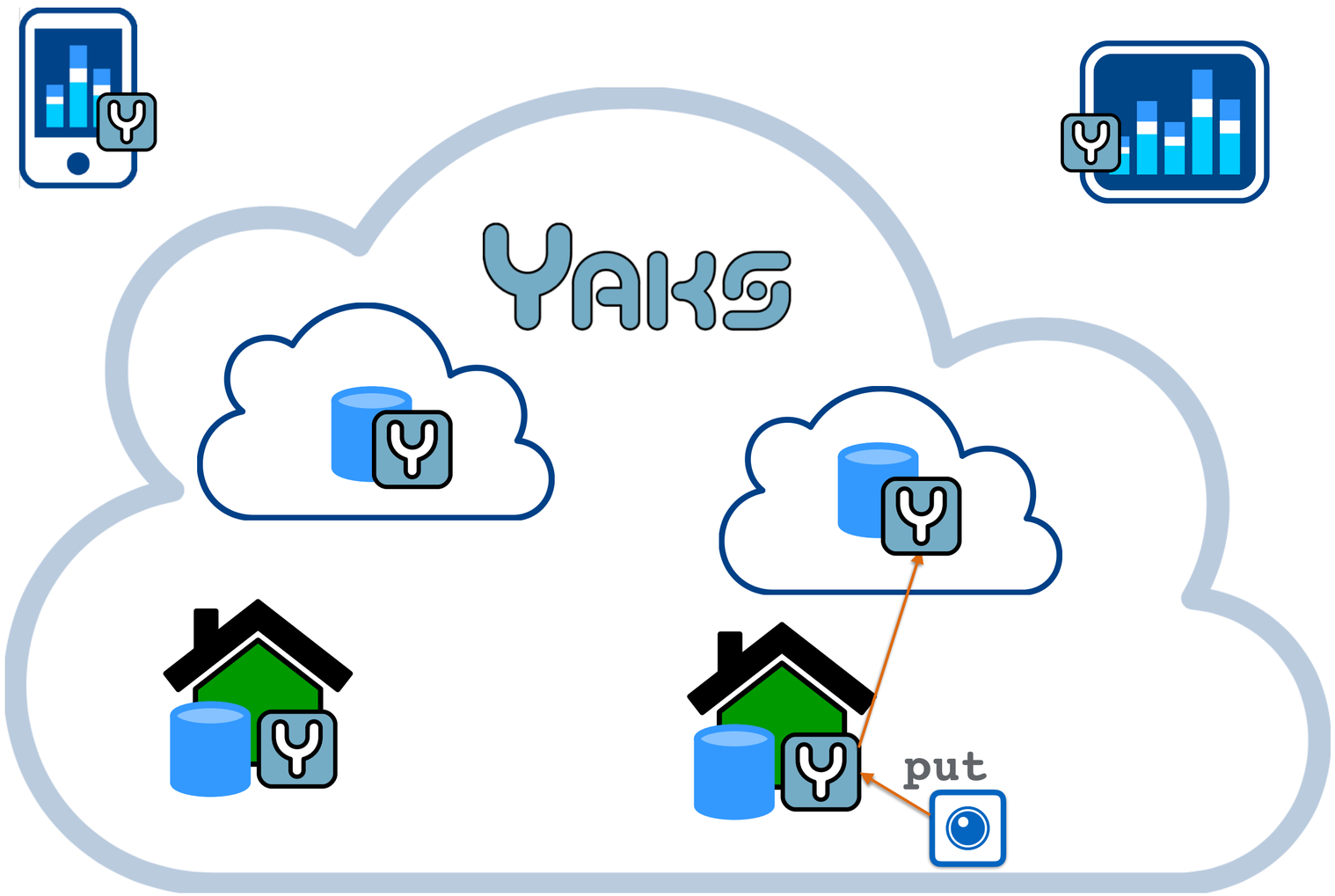
- **put**, **update**, **remove**, **get**
- **subscribe/unsubscribe**
- **register_eval/unregister_eval**, **eval**

Put/Get

Data are published via **put/update**.

Matching **storages** receive and store the data.

Later on, applications can **query** the data

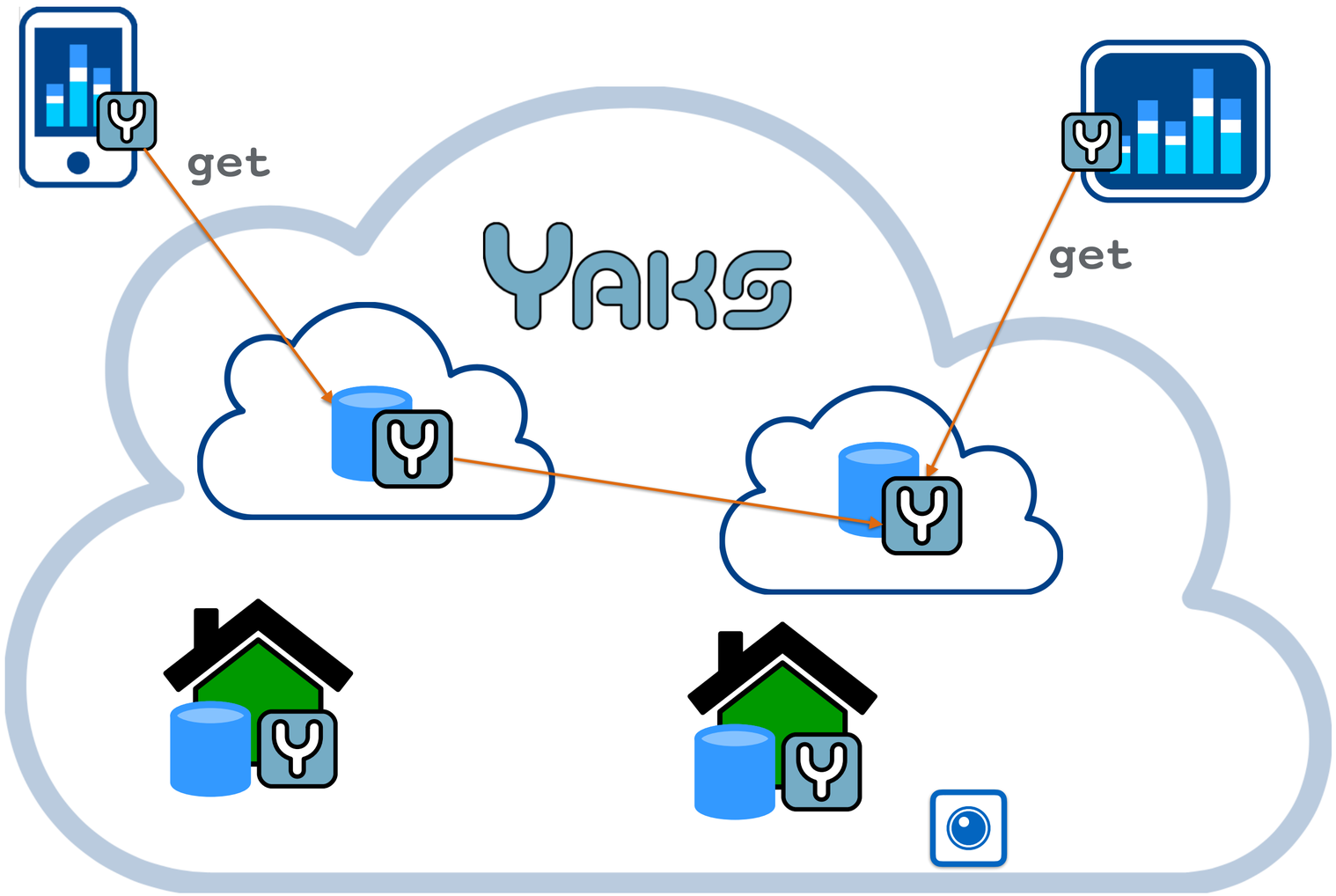


Put/Get

Data are published via **put/update**.

Matching **storages** receive and store the data.

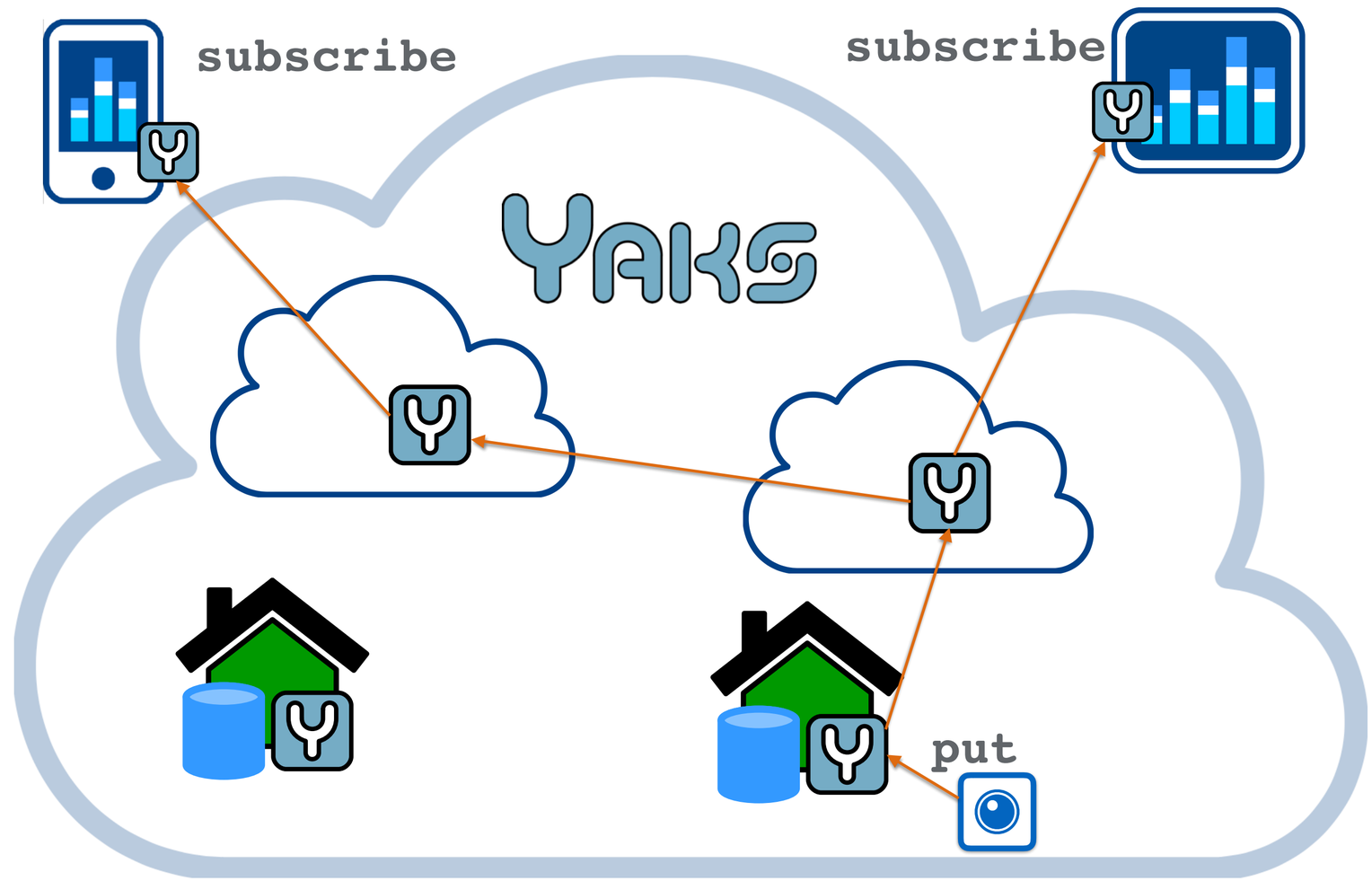
Later on, applications can **query** the data



Subscriptions

Data are published via **put/update**.

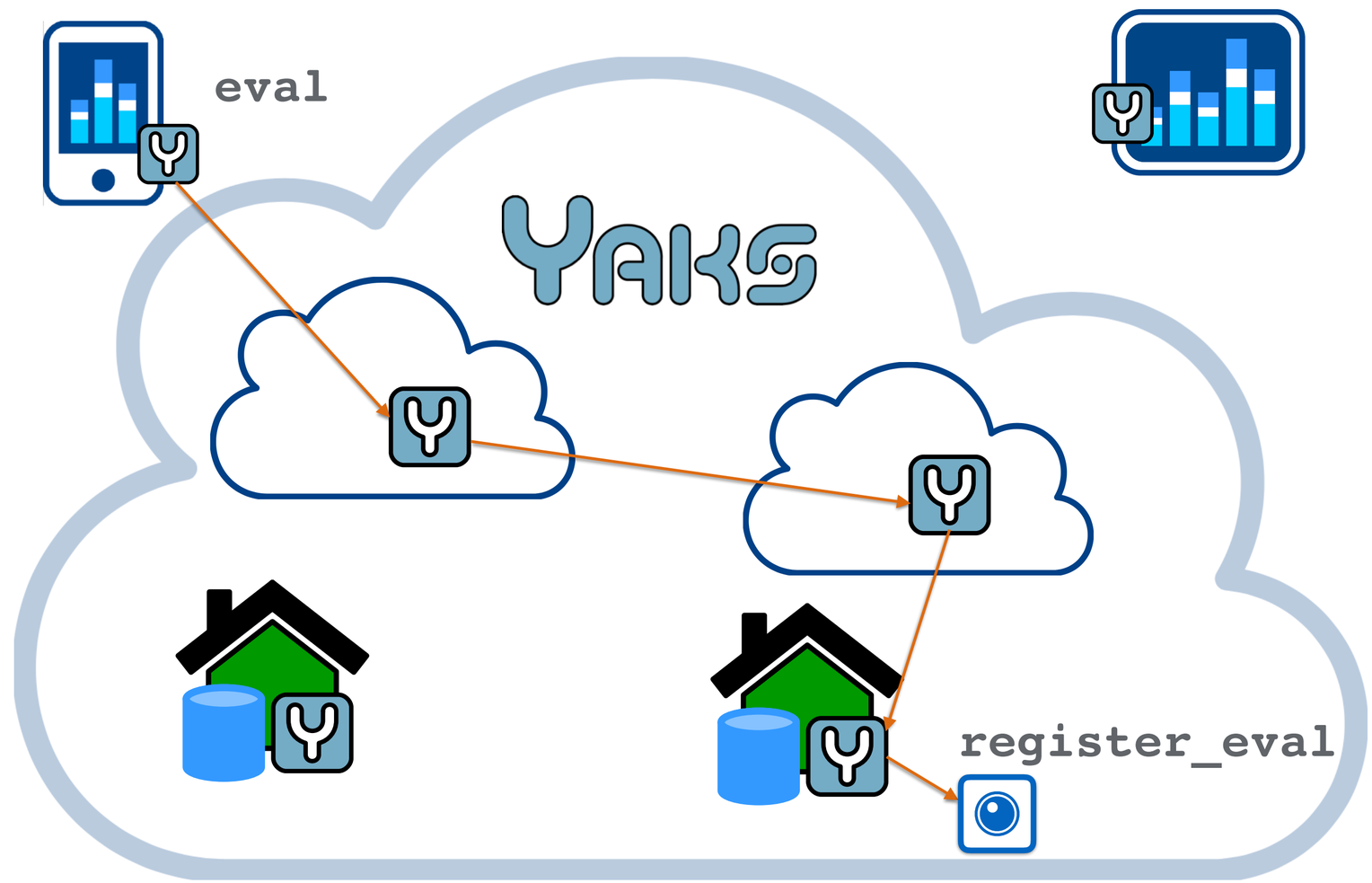
Yaks routes the publications to all matching **subscribers**.



Evals

Any application can **bind** a computation with a path.

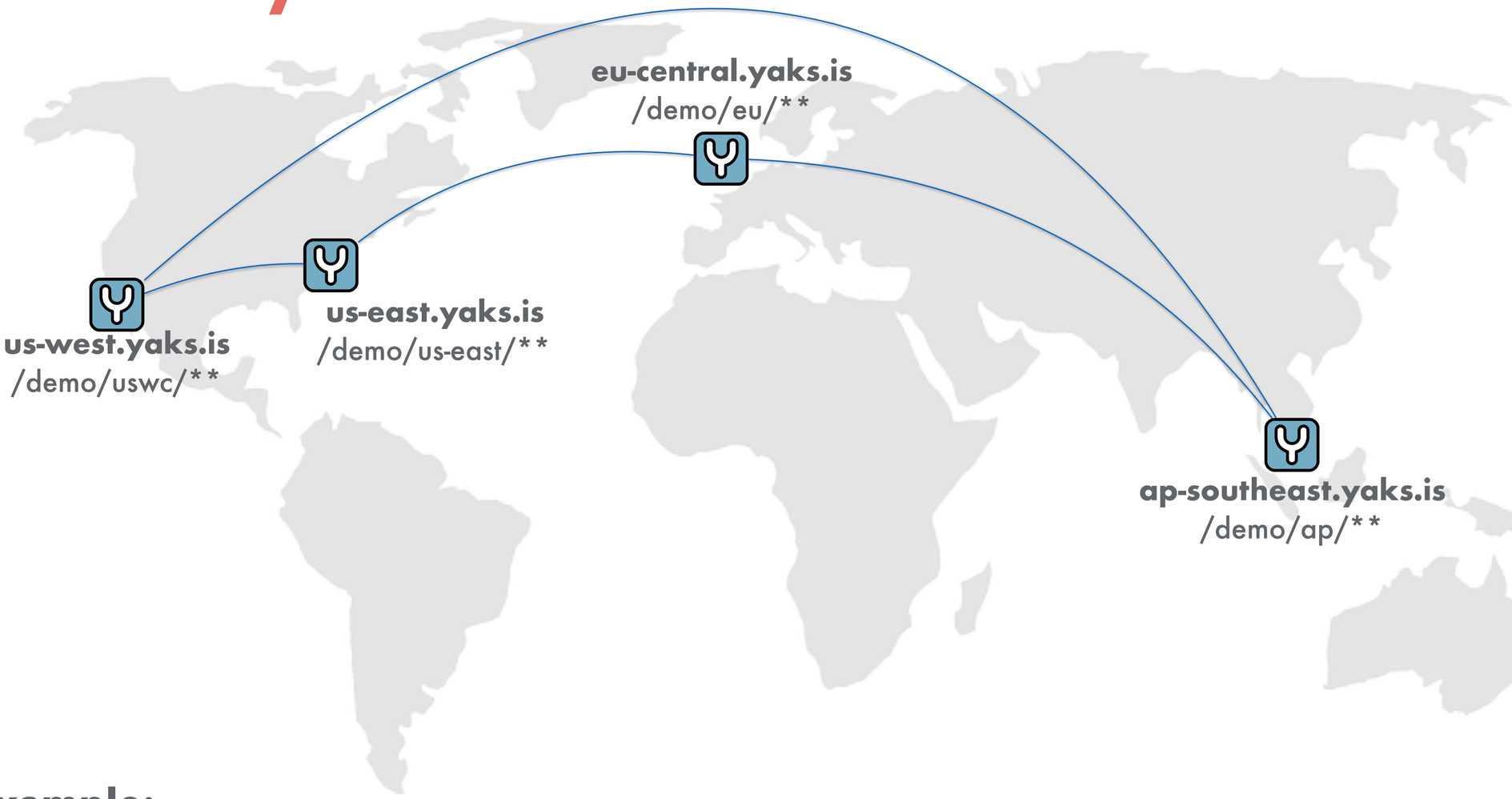
Applications can trigger the execution of these functions by **evaluating** the **path**.



Key Primitives

```
class Workspace(object):

    def put(self, path, value)

    def update(self, path, value)

    def remove(self, path)

    def get(self, selector)

    def subscribe(self, selector, listener)

    def unsubscribe(self, subscription_id)

    def register_eval(self, path, callback)

    def unregister_eval(self, path)

    def eval(self, selector)
```

Play with it....



Example:

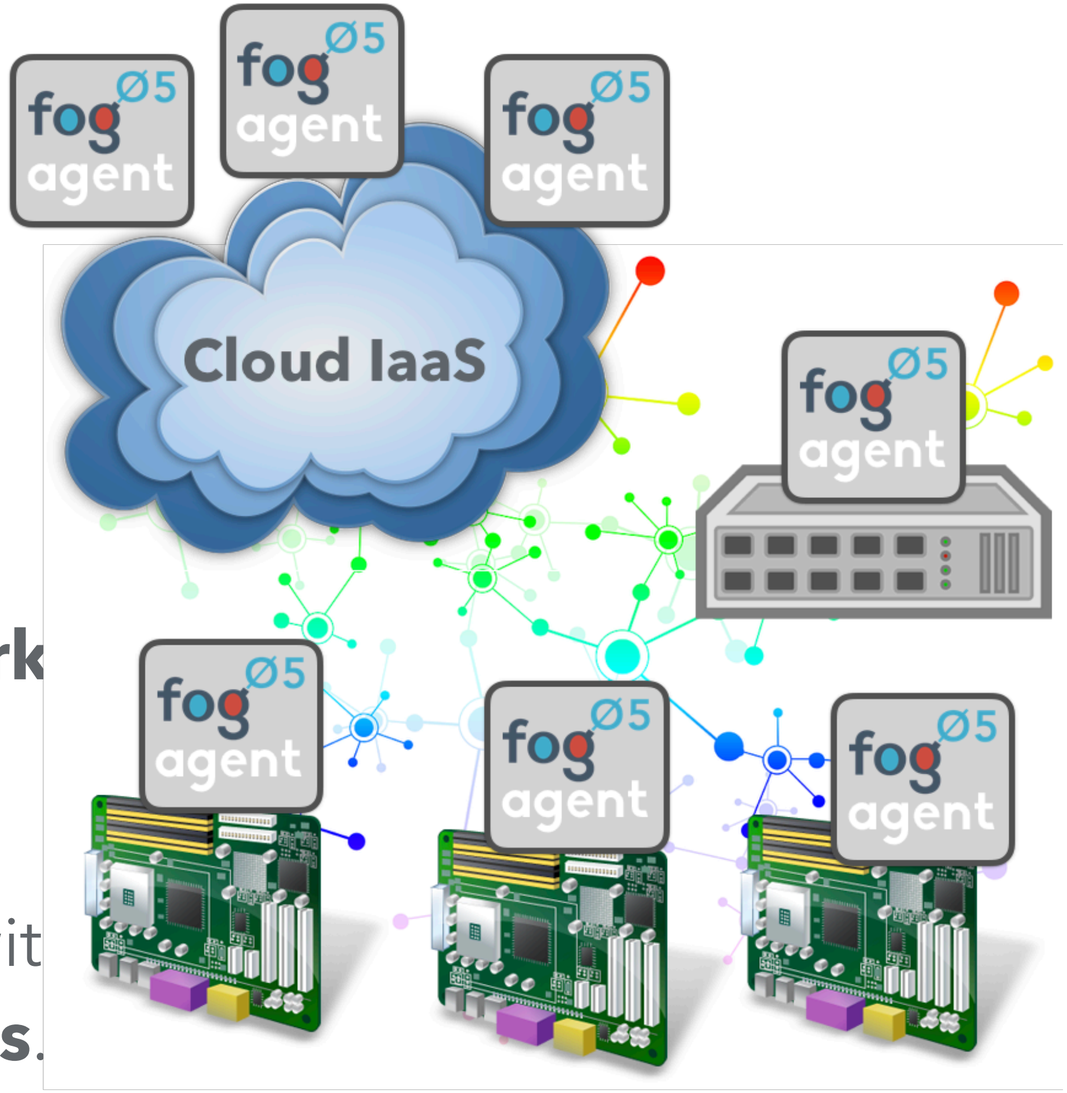
- Put data: `curl -X PUT -d 'Hello World!' http://us-west.yaks.is:8000/demo/eu/test`
- Get data: `curl http://ap-southeast.yaks.is:8000/demo/*/test`

Infrastructure Management

fogØ5 Vision

fogØ5 aims at providing a **decentralised infrastructure** for **provisioning** and **managing** (1) **compute**, (2) **storage**, (3) **communication** and (3) **I/O resources** available **anywhere across** the **network**

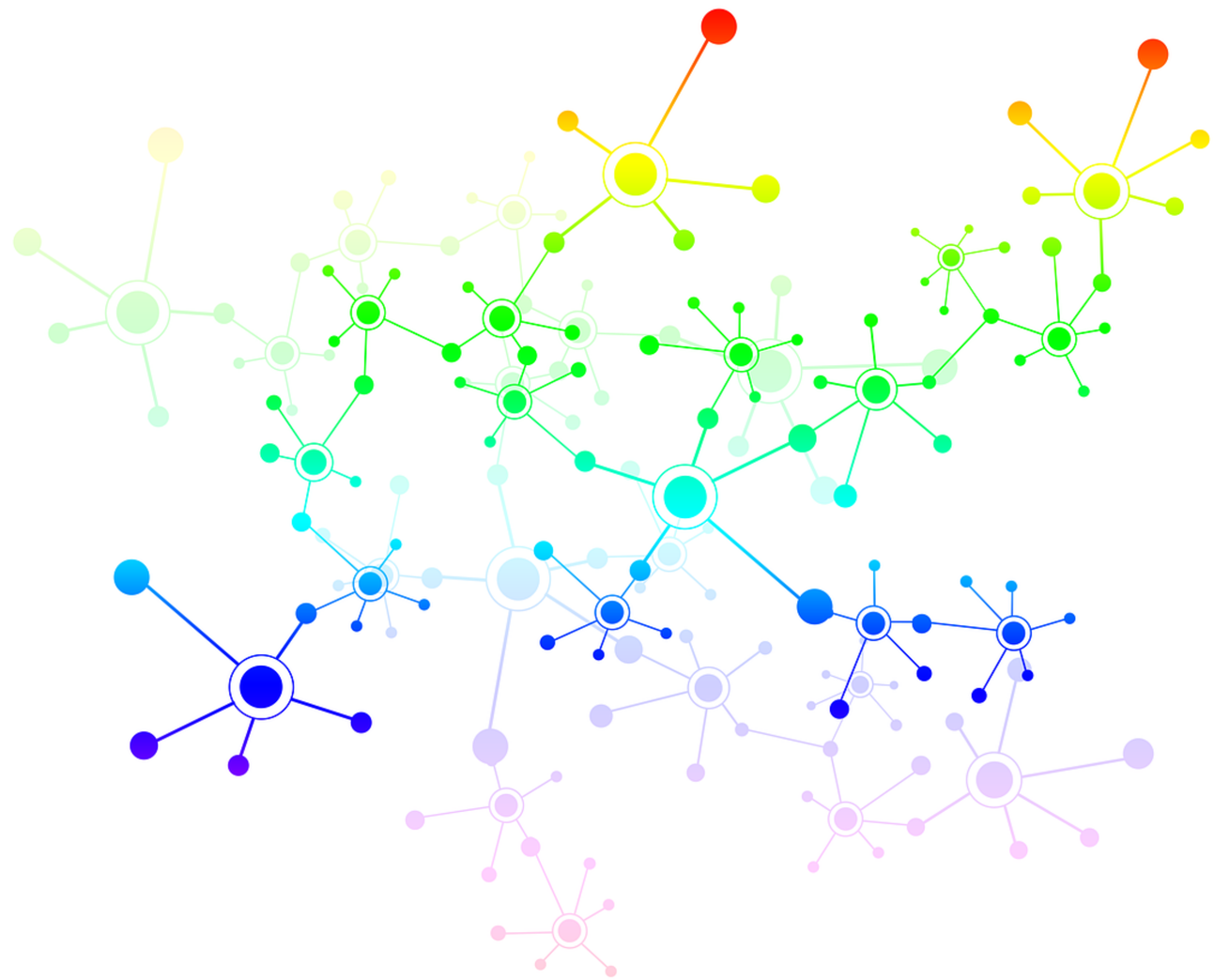
fogØ5 addresses **highly heterogeneous systems** even those wit extremely **resource-constrained nodes**.



fogØ5 — Decentralised Design

fogØ5 can manage and provision any network connected device on which it agent is running

Its **decentralised architecture** allows to manage the system from anywhere and does not need any specific set of nodes running as “servers”



Architecture

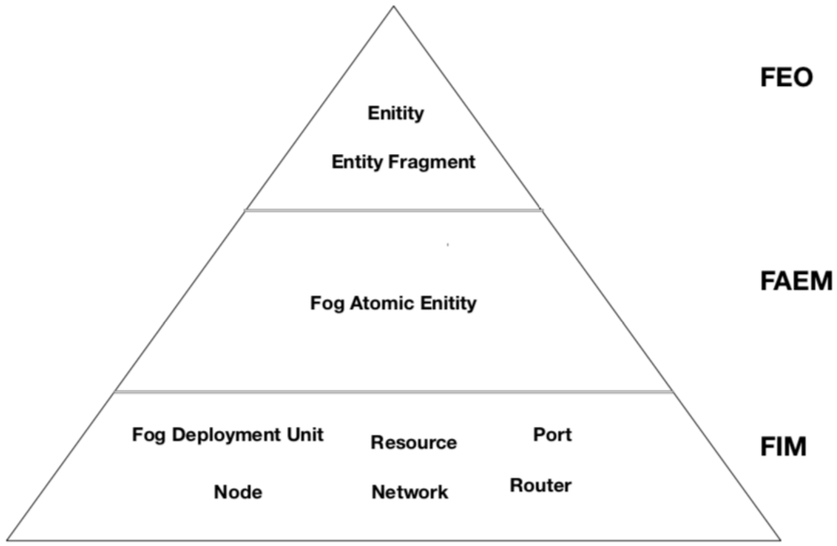
fog Ø5 Modules

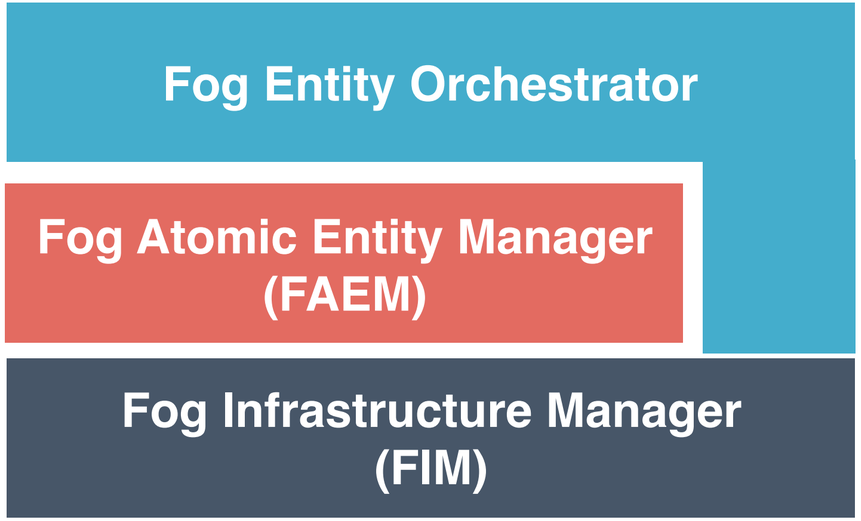
Fog Infrastructure Manager

Virtualises the **hardware** infrastructure, such as **computational**, **communication**, **storage** and **I/O resources**, and abstract the key **primitives** provided by system software, such as the **OS**

Provides **primitives** for **managing** these virtualised **infrastructure**

Provides **infrastructure** level **monitoring** information.





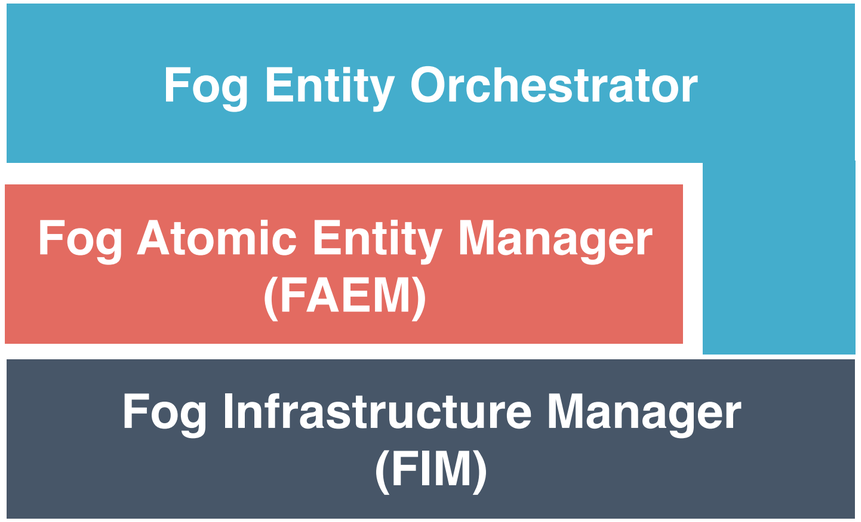
fog05 Modules

Fog Atomic Entity Manager

Manages the Fog Atomic Entity (**FAE**) **life-cycle** and maps then into **FDU** to be deployed by the FIM.

Triggers the FAE specific **monitoring plug-ins** in response to relevant events such as migration, failure, etc.

This information may be used by the Fog Entity Orchestrator (FEO) to trigger re-allocation, restart, etc.



The main **abstraction** provided by the FAEM is the **Fog Atomic Entity**

fog Ø5 Modules

Fog Entity Orchestrator

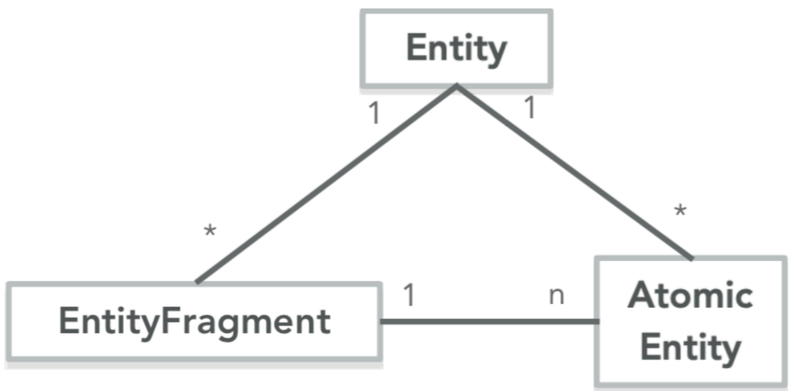
Validates the entity specification

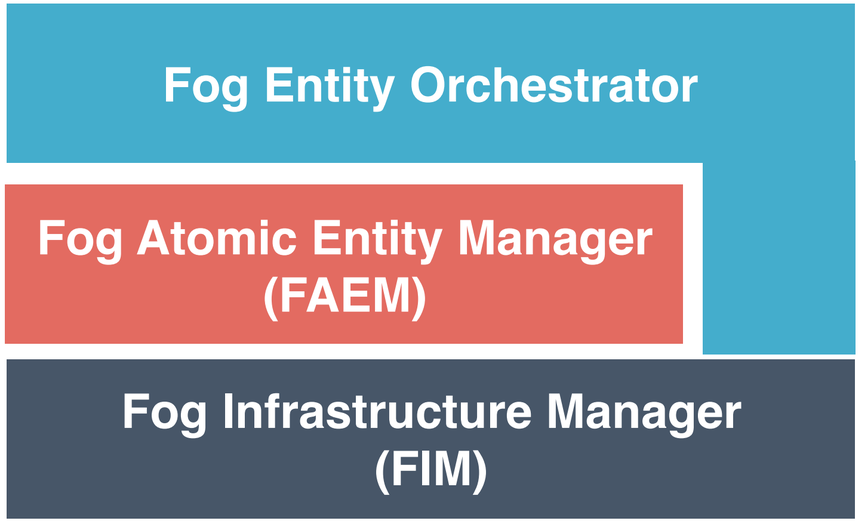
Decides based on available resources and entity **constraints** if it can be accepted.

Device an **allocation** of the entity that **optimises resource utilisation** while satisfying the entity's **functional** and **non-functional requirements**

Executes the **allocation** by proper coordination with the FAEM and FIM

Continuously **monitors** and **reconfigures entity allocation** to ensure that the constraints are satisfied.

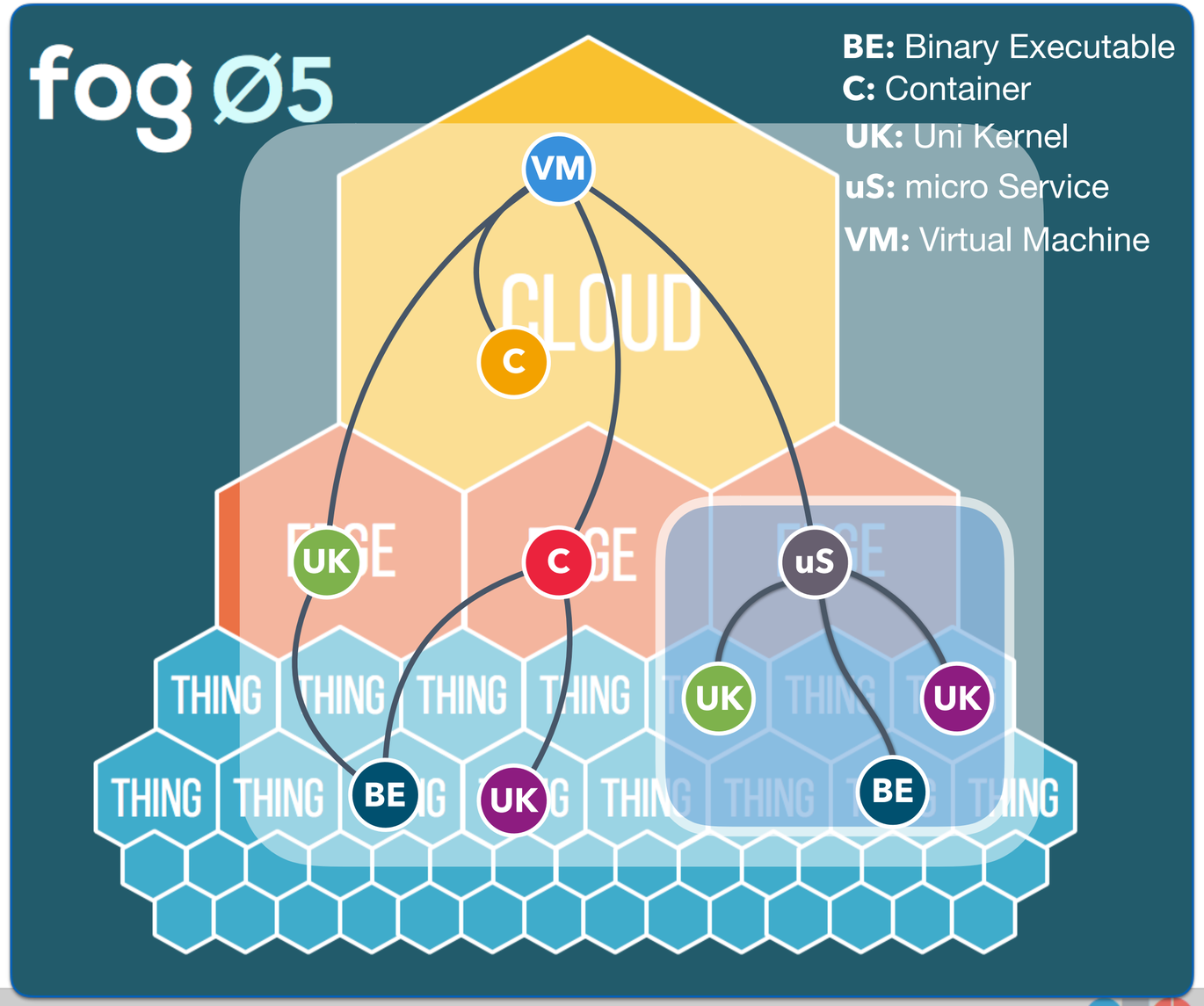




fog Ø5 Entity

An **entity fragment** is directed acyclic graph of atomic entities.

An entity is a directed acyclic graph of atomic entities and entity fragments.



Information Model

fogØ5's information model defines the describes associated with nodes, entities and networks.

Additionally it provides an abstract way to describe applications and relations between them.

It is implemented as a set of YANG Models.


```
{
    "uuid": "1e57bb0e-cb41-4b23-bca3-1fa761a83c02",
    "name": "test_1",
    "computation_requirements": {
        "cpu_arch": "x86_64",
        "cpu_min_freq": 0.0,
        "cpu_min_count": 1,
        "ram_size_mb": 128.0,
        "storage_size_gb": 10.0
    },
    "image": {
        "uri": "file:///home/ubuntu/nginx.tar.gz",
        "checksum": "ef3aa35e64cafef4d87190104adc023b8fa2f20b26c2e6697afde4a39673670a",
        "format": "tar.gz"
    },
    "hypervisor": "LXD",
    "migration_kind": "LIVE",
    "interfaces": [
        {
            "name": "eth0",
            "is_mgmt": false,
            "if_type": "INTERNAL",
            "mac_address": "be:ef:be:ef:00:01",
            "virtual_interface": {
                "intf_type": "VIRTIO",
                "vpci": "0:0:0",
                "bandwidth": 10
            },
            "cp_id": "2192bb19-a3f7-4b59-9473-eeec9573fb3b"
        }
    ],
    "io_ports": [],
    "connection_points": [
        {
            "uuid": "2192bb19-a3f7-4b59-9473-eeec9573fb3b",
            "pair_id": "6cc2aa30-1dcf-4c93-a57e-433fd0bd498e"
        }
    ],
    "depends_on": []
}
```

Relation with ETSI NFV and MEC IM

fogØ5 information model is a super-set of the ETSI (European Telecommunications Standards Institute) MEC and ETSI NFV

Specifically, fogØ5 supports the declaration of I/O constraints.

YANG models have also been defined for fogØ5 abstractions.


```
{
    "uuid": "1e57bb0e-cb41-4b23-bca3-1fa761a83c02",
    "name": "test_1",
    "computation_requirements": {
        "cpu_arch": "x86_64",
        "cpu_min_freq": 0.0,
        "cpu_min_count": 1,
        "ram_size_mb": 128.0,
        "storage_size_gb": 10.0
    },
    "image": {
        "uri": "file:///home/ubuntu/nginx.tar.gz",
        "checksum": "ef3aa35e64cafef4d87190104adc023b8fa2f20b26c2e6697af
        "format": "tar.gz"
    },
    "hypervisor": "LXD",
    "migration_kind": "LIVE",
    "interfaces": [
        {
            "name": "eth0",
            "is_mgmt": false,
            "if_type": "INTERNAL",
            "mac_address": "be:ef:be:ef:00:01",
            "virtual_interface": {
                "intf_type": "VIRTIO",
                "vpci": "0:0:0",
                "bandwidth": 10
            },
            "cp_id": "2192bb19-a3f7-4b59-9473-eeec9573fb3b"
        }
    ],
    "io_ports": [],
    "connection_points": [
        {
            "uuid": "2192bb19-a3f7-4b59-9473-eeec9573fb3b",
            "pair_id": "6cc2aa30-1dcf-4c93-a57e-433fd0bd498e"
        }
    ],
    "depends_on": []
}
```



```
vnfd:vnfd-catalog:
    vnfd:
    -   id: alpine_vnfd
        name: alpine_vnf
        short-name: alpine_vnf
        description: Simple VNF example with a Alpine
        vendor: OSM
        version: '1.0'
        logo: alpine.jpg
        connection-point:
            -   name: eth0
                type: VPORT
        vdu:
        -   id: alpine_vnfd-LXD
            name: alpine_vnfd-LXD
            description: alpine_vnfd-LXD
            count: 1
            vm-flavor:
                vcpu-count: 1
                memory-mb: 256
                storage-gb: 1
            image: file:///home/ubuntu/bench.tar.gz
            interface:
            -   name: eth0
                type: EXTERNAL
                virtual-interface:
                    type: VIRTIO
                    bandwidth: '0'
                    vpci: 0000:00:0a.0
                external-connection-point-ref: eth0
        mgmt-interface:
            cp: eth0
```

Architecture

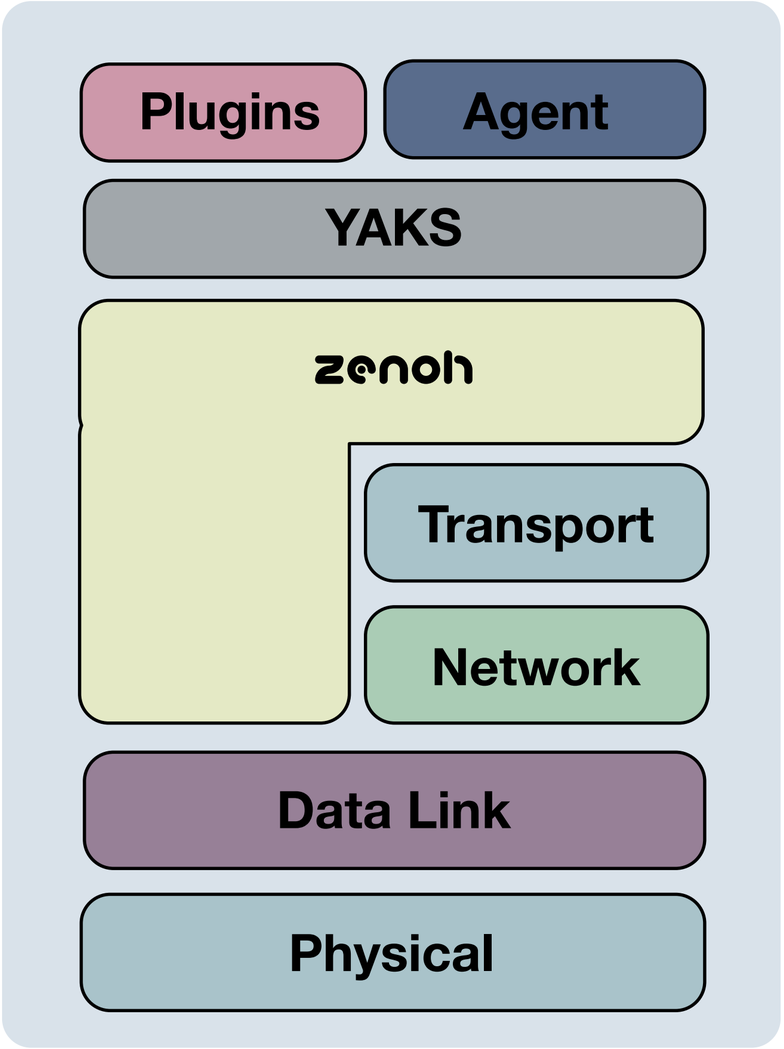
fogØ5 is composed by:

NDN. At its lowest level, it leverages a Named Data Network (**NDN**) infrastructure based on zenoh. DDS can also be used as a transport – not necessarily an NDN

YAKS. A distributed key-value store that leverages the NDN for scalability

Agent. The core logic of **fogØ5**, it takes care of managing, monitoring and orchestrating entities through plugins

Plugins. Plugins provide supports for atomic entities, OS, networks, etc.



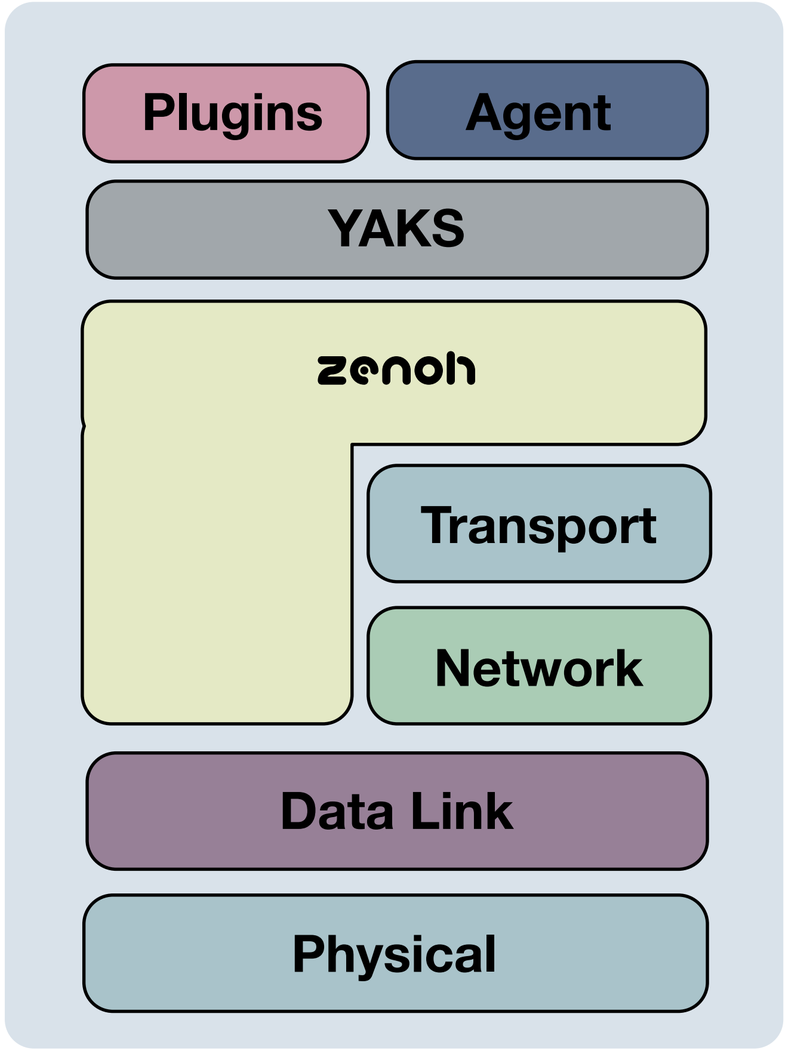
Plugins

fogØ5 leverages plugins interact and manage:

- Atomic Entities (Runtimes)
- Networks
- OSes
- Monitoring
- Resource Orchestration
- Resource Management

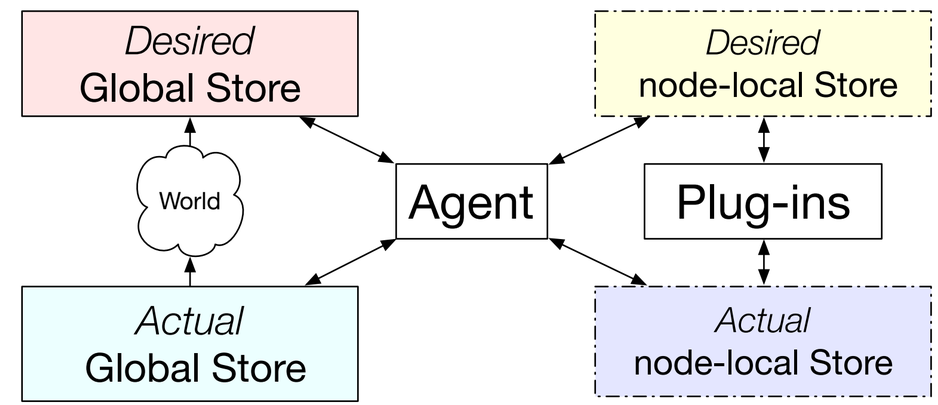
For each type of plugin an interface has been defined.

For instance, plugins that manage atomic entities have to implement the FSM for the kind of atomic entity they will be managing.

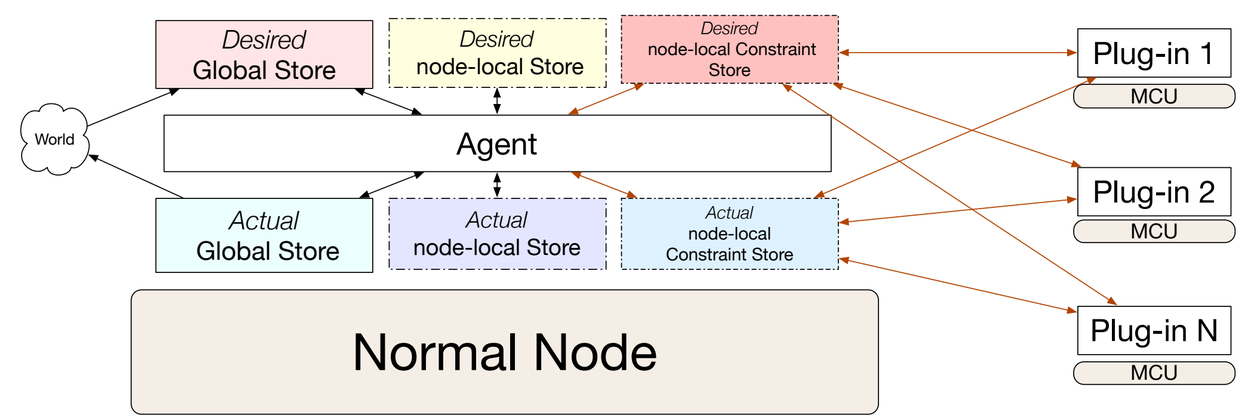


Local/Global and Actual/Desired

fogØ5 uses YAKS to maintain the actual and desired state for global and node-specific information



This separation ensures that there is never write concurrency on the actual state and that the evolution is entirely under the control of the agent



Interact with fogØ5

To interact with **fogØ5** we provide a set of API for Python3.

These API uses interact with **fogØ5** using the distributed data store.

The demo that we show uses this API.

API Docs: https://atolab.github.io/fog05-doc/fog05.html#module-fog05.api

```
input('press enter to onboard descriptor')
a.fdu.onboard(fdu_d, wait=True)
input('Press enter to define')
a.fdu.define(e_uuid, n1, wait=True)
input('Press enter to configure')
a.fdu.configure(e_uuid, n1, wait=True)
input('Press enter to run')
a.fdu.run(e_uuid, n1, wait=True)
```

fog05 – The Eclipse Project

ECLIPSE FOUNDATION

Members Working Groups Projects More

Home / Projects / Eclipse IoT / Eclipse fog05

Eclipse fog05

Since June 2018

Overview | Downloads | Who's Involved | Developer Resources | Governance | Contact Us

Early IoT applications, especially those addressing the consumer market, have been embracing cloud-centric architectures in which data is pushed up to the cloud. It is within the cloud the everything takes place before eventually pushing some data or action back to the edge. This architectural approach leverages the availability and operational maturity of the cloud but it is not generally applicable in IoT / Cyber Physical Systems (CPS).

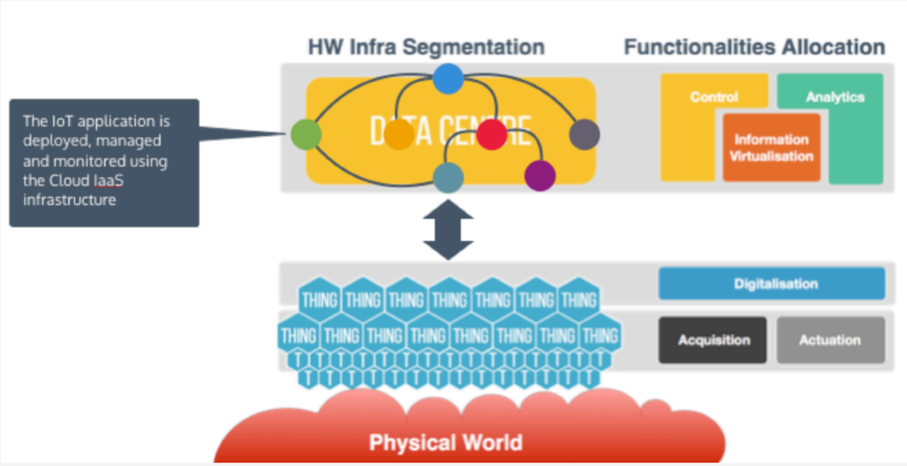


Figure1– Cloud centric architecture.

Search or jump to... / Pull requests Issues Marketplace Explore

eclipse / fog05

Unwatch 7 Star 4 Fork 3

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Insights

No description, website, or topics provided.

159 commits | 2 branches | 1 release | 1 contributor | View license

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

gabrik Merge pull request #25 from gabrik/master ··· Latest commit ac6bbf0 5 days ago

bin	added signal catching for sigterm and ctrl-c and ctrl-break on windows	24 days ago
etc	removing subscriptions when connection to yaks is closed	2 months ago
fog05	using syslog also on windows/mac	5 days ago
plugins	logging process pid on native plugin	7 days ago
.gitignore	eclipse_initial	4 months ago
CONTRIBUTING.md	updated readme added contributing file	4 months ago
Introduction.md	Added Introduction.md	4 months ago
LICENSE.md	double licensing update	3 months ago
MANIFEST.in	agent autoload plugins, manifest, connector to OSM	4 months ago
Makefile	added RO and RM plugin interfaces	6 days ago
README.md	Update README.md	7 days ago
TODO.md	updated TODO.md	4 months ago
configure	updated makefile and configure	7 days ago
setup.py	updated network creation script with shebang at start of file	2 months ago
setup_api.py	added setup_api.py for single api installation	2 months ago

fogØ5 OpenFog and 5GPPP

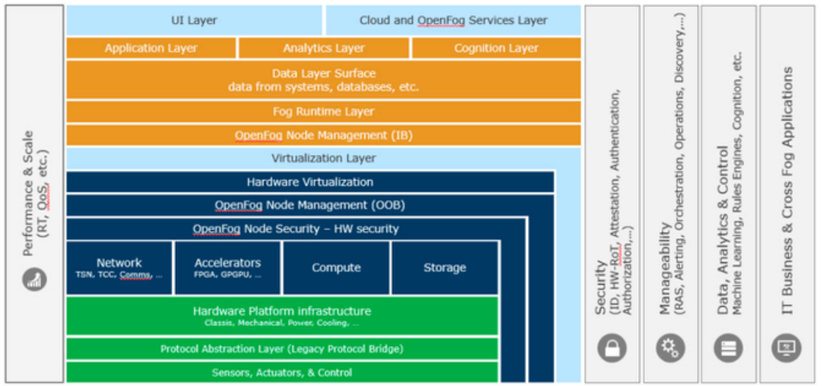
fogØ5 is one of the infrastructure identified as **compliant** with the **5G principles** and requirements by the **EU 5GPPP working group**

fogØ5 architecture is compatible with the **OpenFog Reference Architecture.** Additionally **fogØ5** is used as the reference fog platform in several test-beds

5G-PPP Software Network Working Group

From Webscale to Telco, the Cloud Native Journey

The OpenFog Reference Architecture: Open, Secure & Interoperable



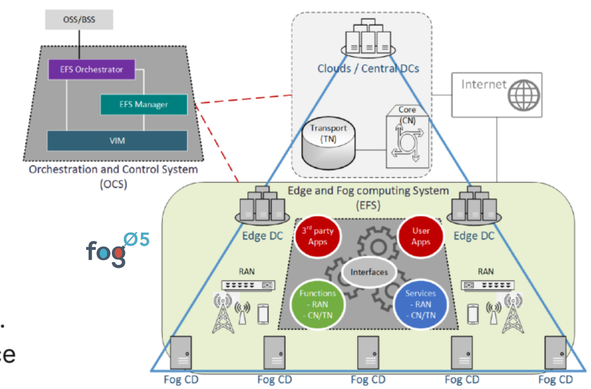
Source: The OpenFog Consortium 2017

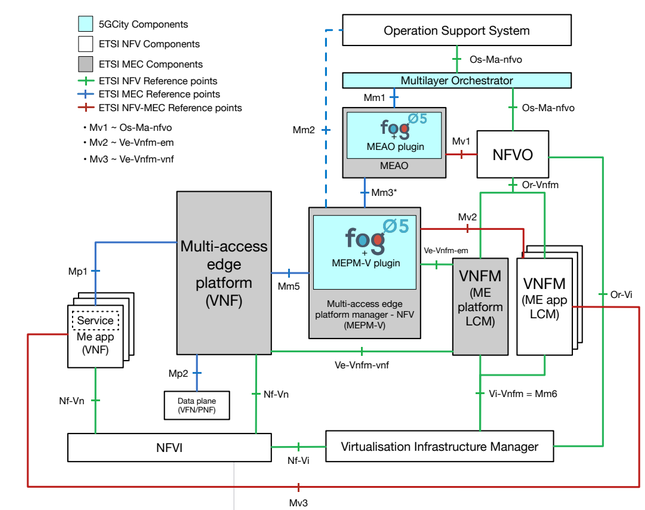
fog05 Users and Press

Wolfram Willuhn • 1:13 AM
Hi Angelo

grazie mille for your presentation today about FOG05.
Great presentation and great work on this open source
I appreciate it.
All the success with this and all your other projects!!!

Ciao, Wolfram
data science for IoT and process intelligence





Sayan Chakraborty • 12:54 AM
Really enjoyed the FogOS talk and demo especially non-IP network. I think this will make it ideal for Industrial IoT applications !

Wrapping Up

Key Takeaways

1

Edge Intelligence is just a specific aspect of Edge Computing that poses some **new challenges** with respect to **AI algorithms but shares** the bulk the **infrastructural challenges** with **"traditional" edge applications**

Key Takeaways

2

The Edge is fuzzy in essence and limiting by nature. We should **focus** on **infrastructure** that allows to **unify** the **computational**, **communication**, **communication** and **I/O resources end-to-end**

Key Takeaways

3

The **world outside** of the **data-center** is **constrained**, **heterogeneous**, and **tricky**. Yet, **that's the place** where the **difference** can be **made**.

ADLINK

Leading EDGE COMPUTING

aTO

INNOVATING TOGETHER

Angelo Corsaro, PhD

Chief Technology Officer
ADLINK Tech. Inc.
angelo.corsaro@adlinktech.com