Cloud-Edge Continuum

Antonio Brogi

Department of Computer Science
University of Pisa

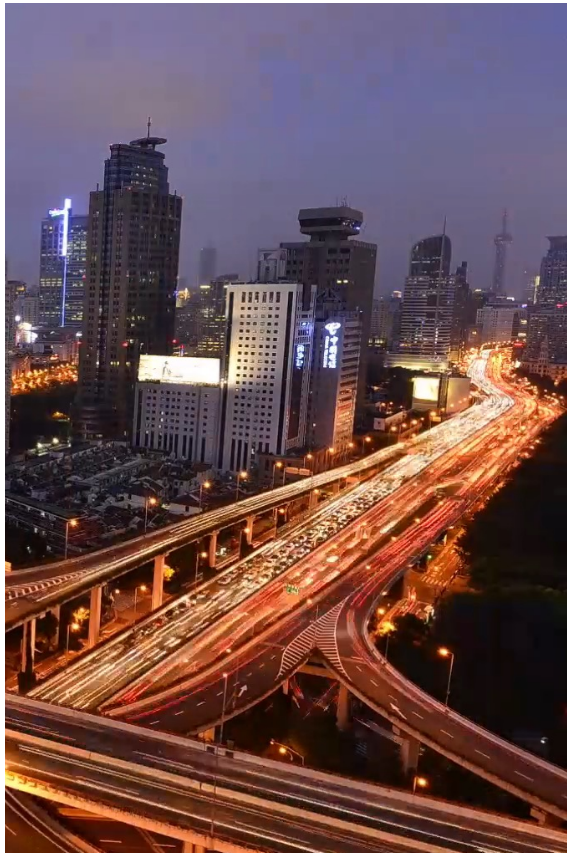
Cloud-Edge Continuum

Motivations

Pervasive IoT applications

Embedded AI

Energy production plants



Smart Cities

IoT applications

sense | process | actuate

where?

Traditional deployment models

Cloud

IoT + Cloud

Data processed in the Cloud

\+ Unlimited processing and storage capabilties

\- Very many data transferred

\- Mandatory Internet connectivity

\- High latencies

IoT

Edge

IoT + Edge

Data processed at the Edge

\+ Low latencies

\- Limited processing and storage capabilities

IoT

So much data, really?

www.leverege.com/calculator

1	Moisture Sensor	SIZE:	8	bytes	1min
1	pH Sensor	SIZE:	8	bytes	1min
1	Thermometer	SIZE:	8	bytes	1h
1	Water Quality Monitor	SIZE:	56	bytes	1h
1	Videocamera	SIZE:	56250	bytes	1s

Resources Used

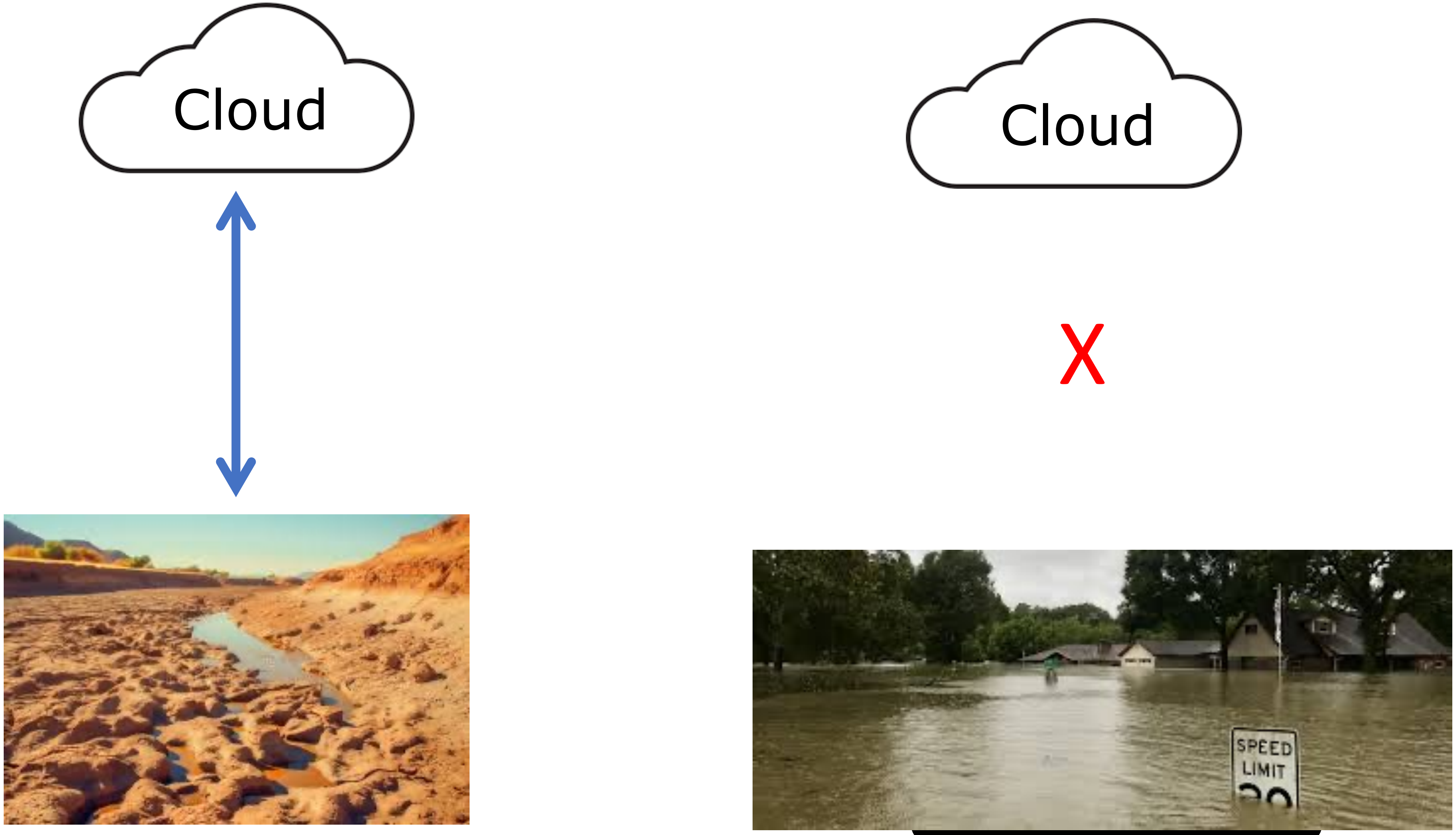
Resource	Value	Unit
Data Transfer	145800.74	MB / month
Cloud Storage	437402.21	MB / month
Updates	~2.68 M	per month

146 GB/month

437 GB/month
5.2 TB/year

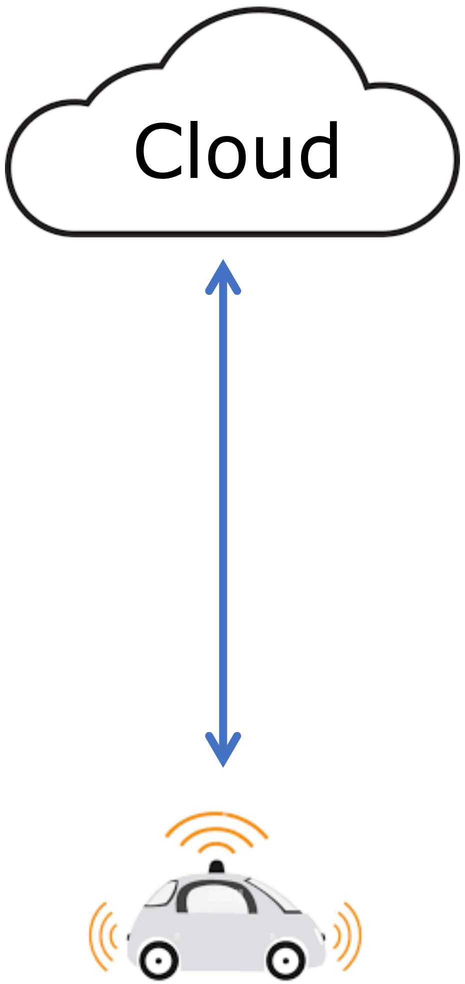
Mandatory Internet connectivity

e.g. water flooding management must work in critical situations



High latencies

e.g. autonomous vehicles need to stop promptly



[Autonomous vehicles: ethics issues, too]

Scenario:

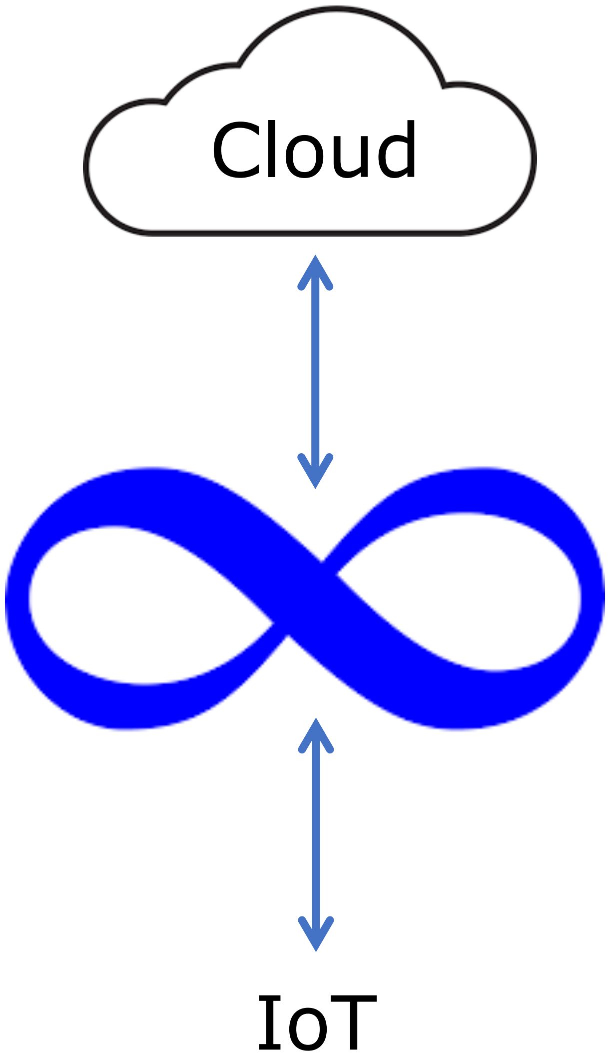
- After turning right, vehicle detects children in the middle of the street
- The children are too close for the vehicle to be able to stop in time
- What should the vehicle (be programmed to) do?

Cloud-Edge Continuum

Motivations

The idea

Cloud-Edge Continuum

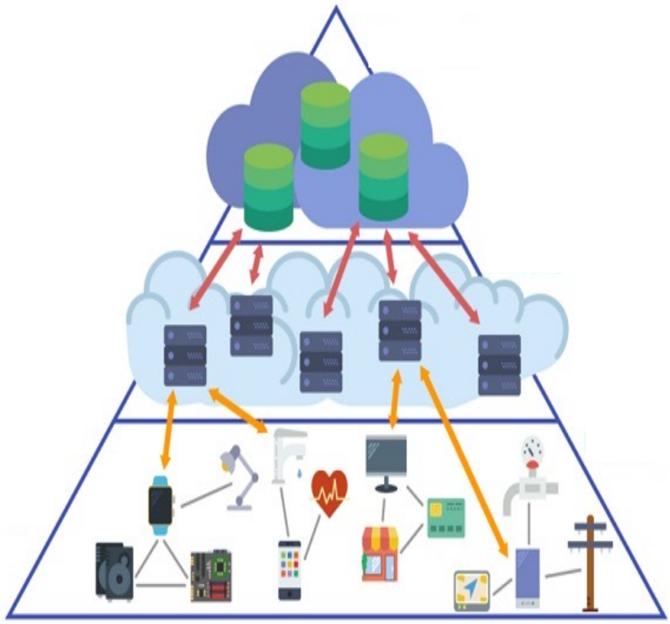


Extending the Cloud towards the IoT

with a **distributed, heterogenous infrastructure**

to get the "best of both worlds"

- computing power ✓
- connectivity ✓
- latency ✓

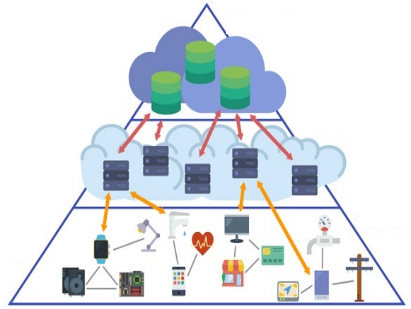


Next-gen applications

Containerised,

microservice-based applications

deployed on

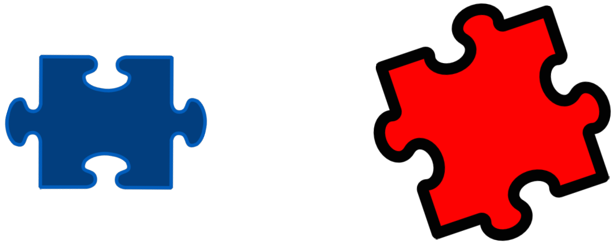


a **continuous Cloud-Edge infrastructure**

Cloud-Edge Continuum
Application placement

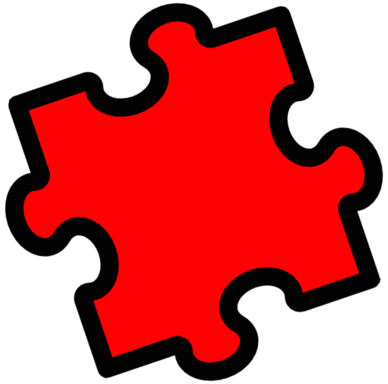
How to suitably **place**
a composite application
in the Cloud-Edge Continuum?

It is challenging (and NP-hard)



Application requirements

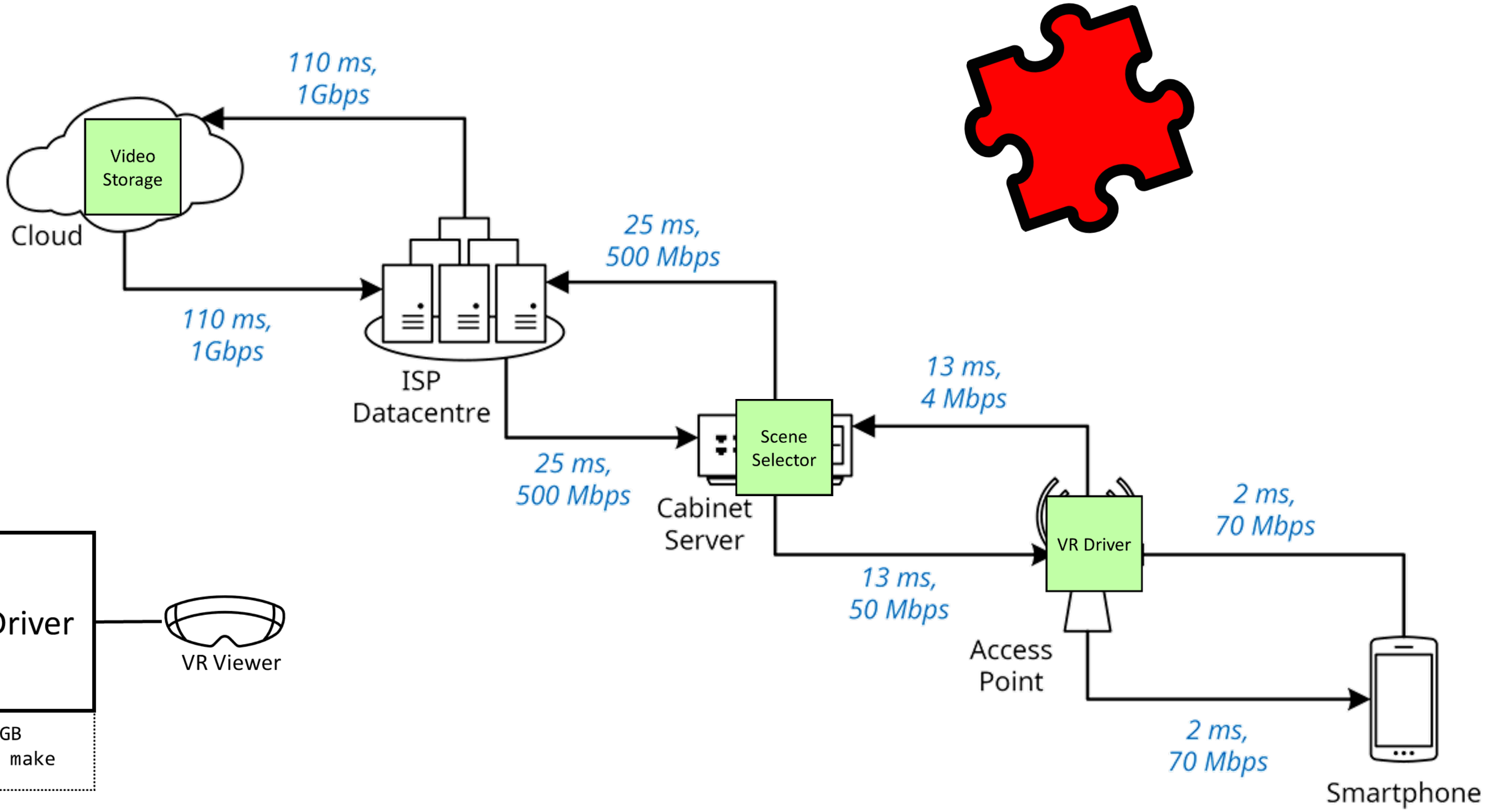
- Hardware
- Software
- QoS (e.g. response time, reliability)
- Data awareness
- Security and trust
- …

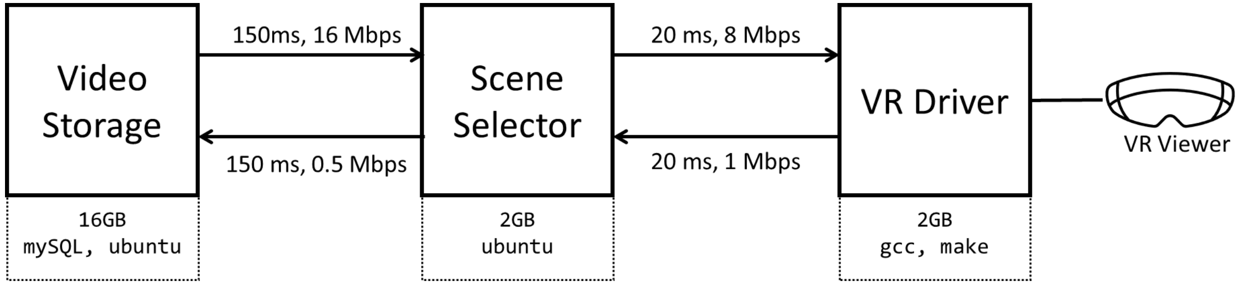


Infrastructure is

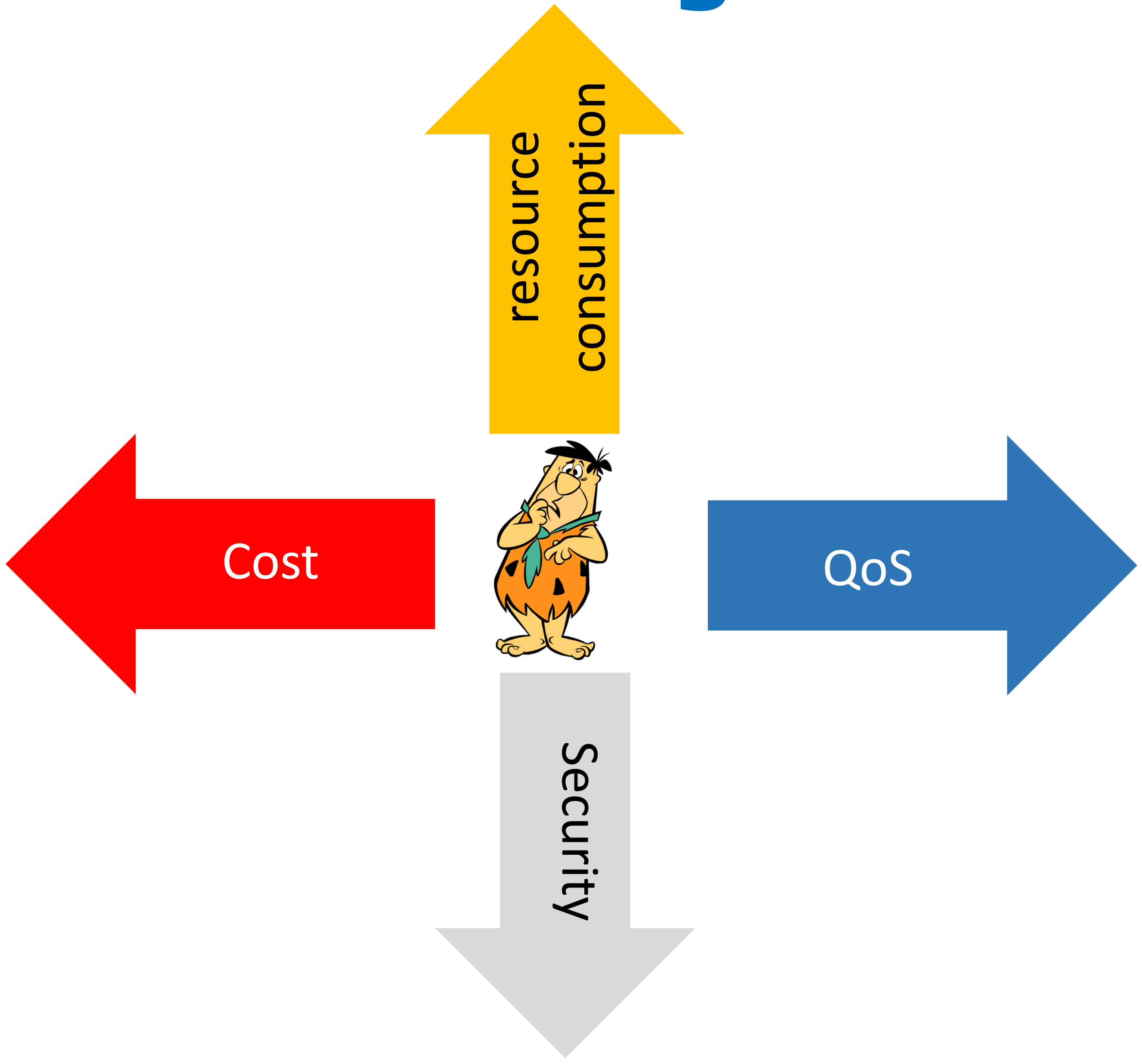
- Heterogeneous
- Large*
- Dynamic**

Example: Simple VR Application



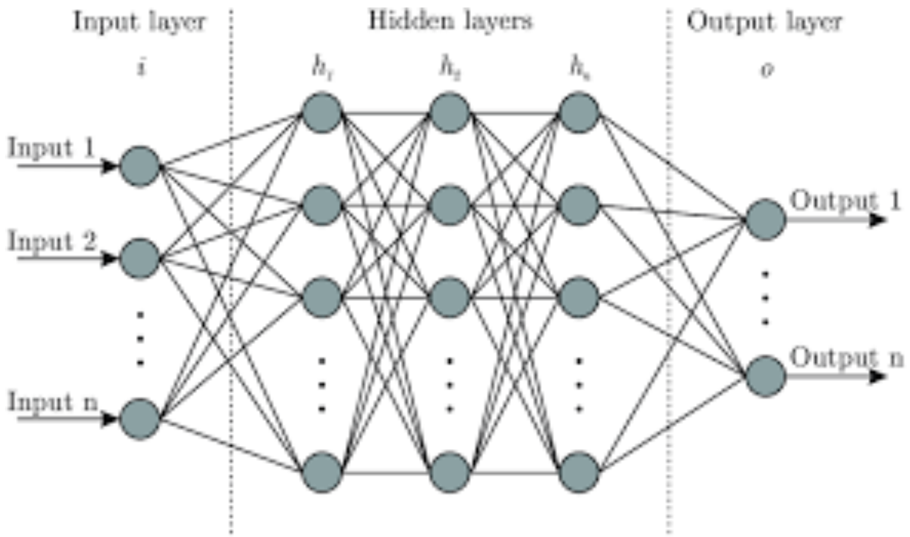


Need to master orthogonal dimensions



Application placement: approaches (1/3)

ML



Infrastructure is very dynamic

Lack of explainability

Application placement: approaches (2/3)

MILP

- $X_{ij}^{k} = BINARY$
- $\sum_{j=1}^{n} X_{0j}^{k} = \sum_{i=1}^{n} X_{i0}^{k} = 1;$ *k=1,2…m : Every route should start from depot and end on depot only,*
- $\sum_{i=1}^{n} X_{ih}^{k} = \sum_{j=1}^{n} X_{hj}^{k} <= 1$, *h=1,2,…n; k=1,2…m :every node should be selected at most once and every node served should have a in as well as out arc,*
- $\sum_{k=1}^{m} \sum_{j=1}^{n} X_{hj}^{k} = \sum_{k=1}^{m} \sum_{i=1}^{n} X_{ih}^{k} = 1;$ *h=1,2…n : Every node should be selected at least once,*
- $\sum_{j=1}^{n} \sum_{i=1}^{n} D_i X_{ij}^{k} <= Q;$ *k=1,2… m :Carrier can't carry more than q quantity,*
- $\sum_{k=1}^{m} \sum_{j=1}^{n} \sum_{i=1}^{n} D_i X_{ij}^{k} = \sum_{i=1}^{n} D_i;$ *Total supply to nodes should equal to total demand,*
- $X_{hj}^{k} + X_{ih}^{k} <= 1;$ *for i,j=1,2…n; h=1,2…n for every k=1,2…m: Every node visited should have an arc to other then its preceding node.*

Finds optimal solutions

Hard to read, hard to code non-numerical info

Slow to run

Application placement: approaches (3/3)

Declarative programming

1) Declare what an eligibile placement is

service S can be placed on node N ***if***
hardware reqs of S are met by N ***and***
software reqs of S are met by N

services S1 … Sm can be placed on nodes N1 … Nm ***if***
service S1 can be placed on node N1 ***and***
… ***and***
service Sm can be placed on node Nm ***and***
QoS reqs of S1 … Sm are met

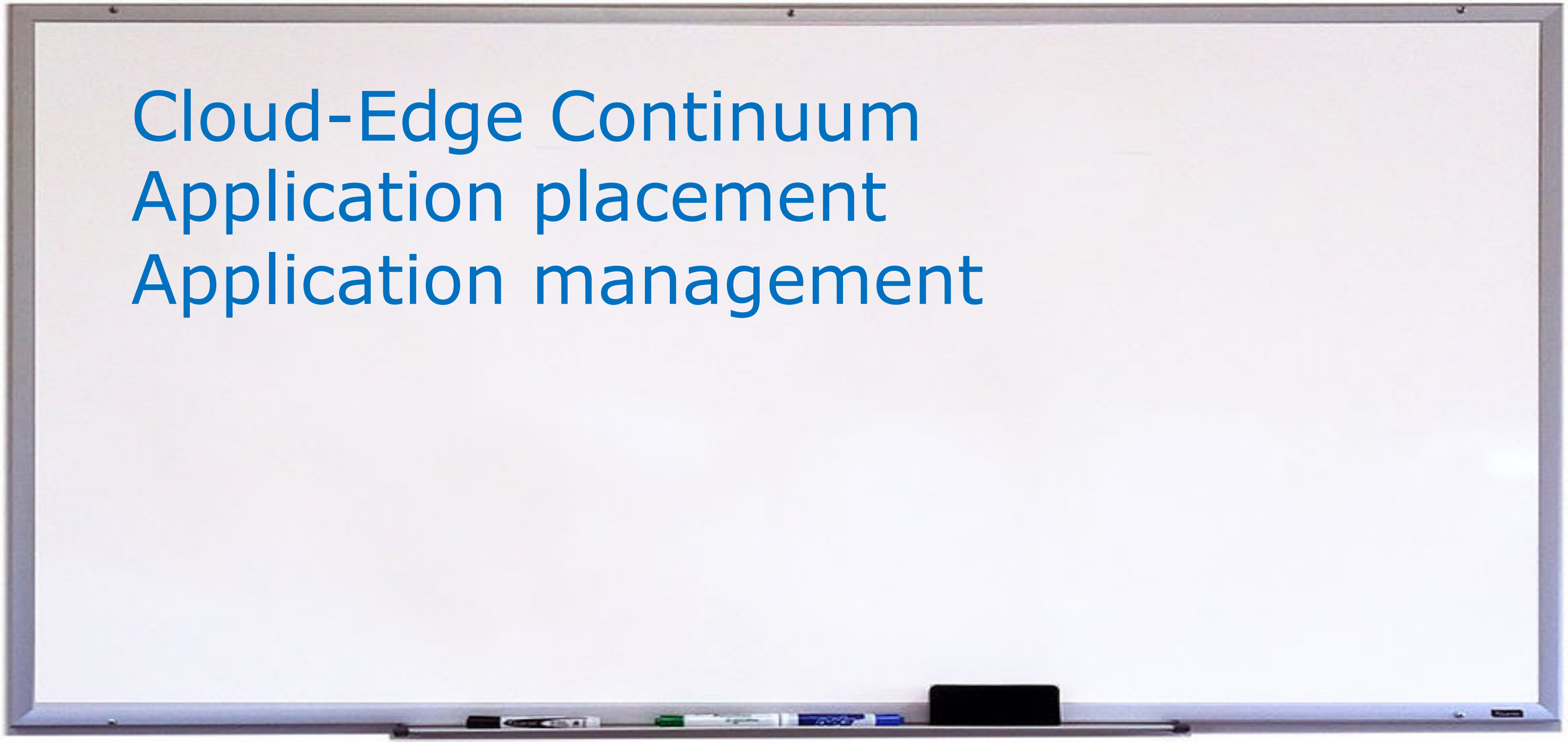
2) Let the inference engine look for it!

? placement([s1,..sm],P).

Easy to read

Easy to code non-numerical info

Explainable



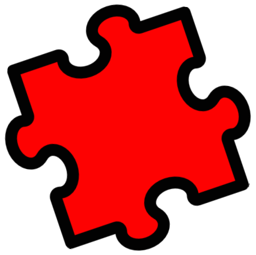
Cloud-Edge Continuum
Application placement
Application management

Things change …

Wile E. Coyotes’s syndrome

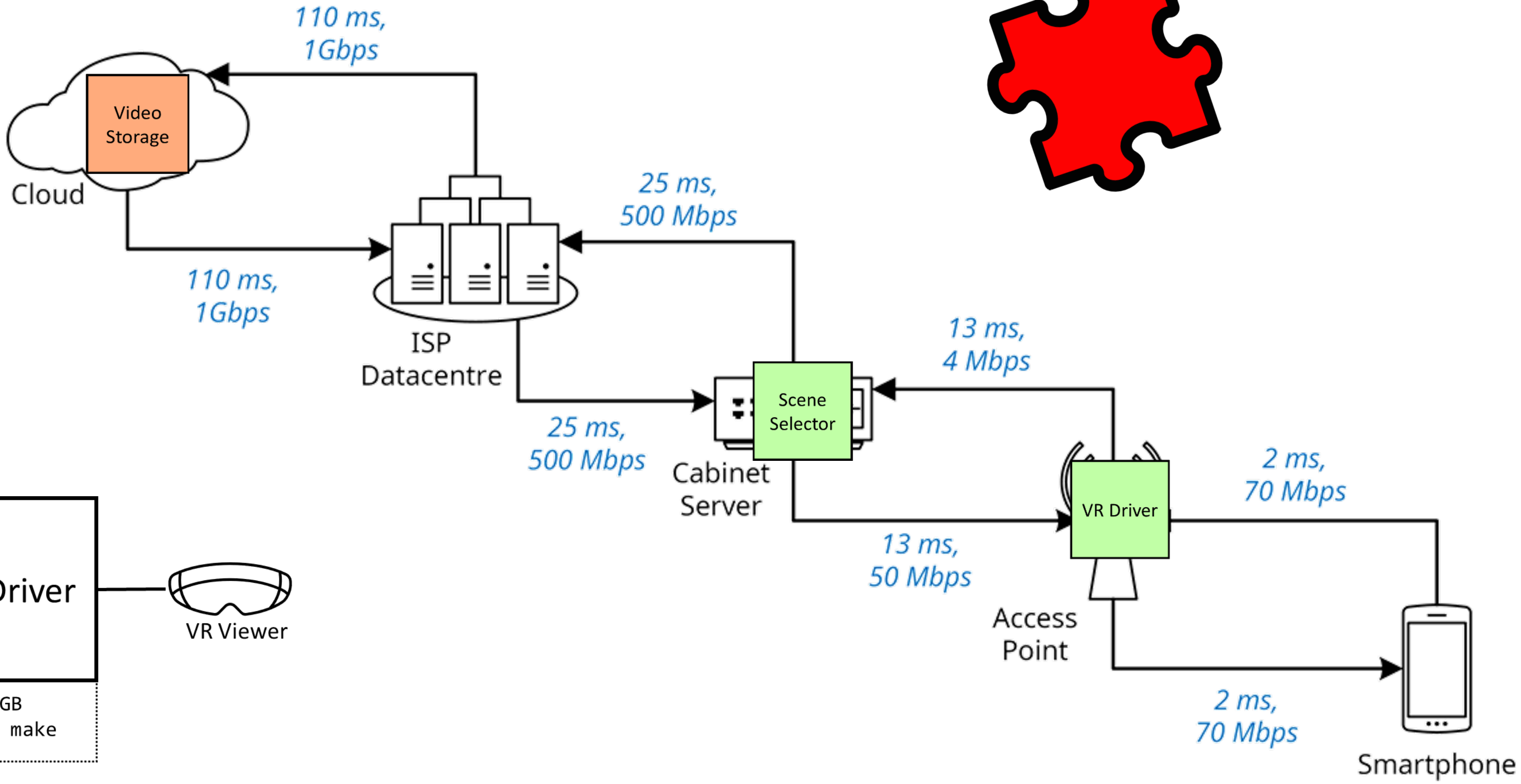
Things constantly change …

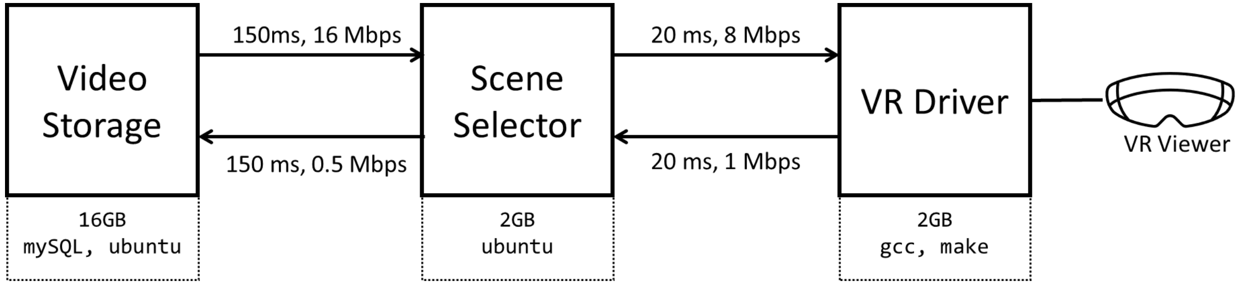
- Infrastructure constantly changes
  - Node’s workload changes
  - Latency and bandwidth vary
  - Nodes can join and (suddenly) leave
  - Nodes and connections can (temporarily) fail
  - …



- Applications change too
  - Codebase’s changes
  - Reqs can change
  - …

Example: Simple VR Application





How to

constantly and suitably **manage**
application deployments

after first deployment?

Monitoring

Need of effective (lightweight, fault-tolerant) monitoring of

- applications
- infrastructure

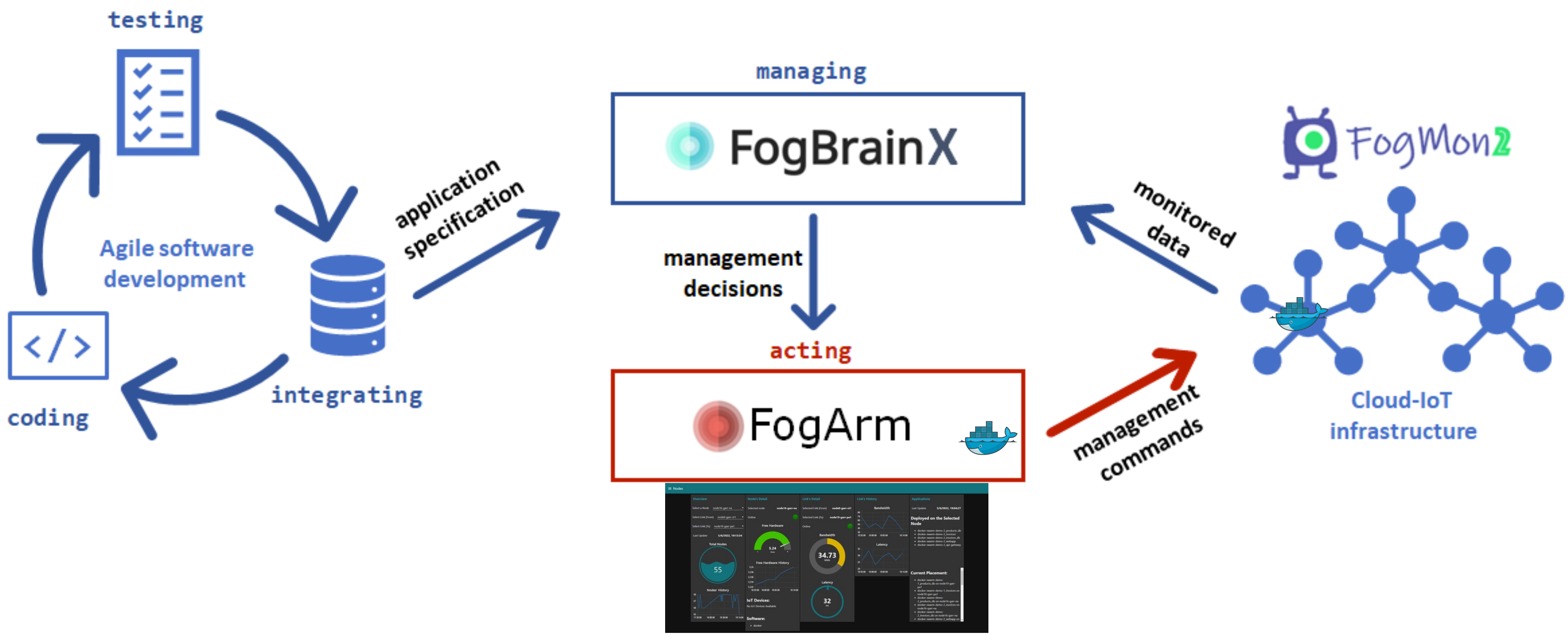
Continuous reasoning

Differential analysis

- focusing on last changes
- re-using previously computed results

to re-place/migrate, restart, scale application services

Monitor | Reason | Enact



G. Bisicchia, S. Forti, E. Pimentel, A. Brogi. Continuous QoS-compliant Orchestration in the Cloud-Edge Continuum. Software: Practice and Experience. 2024. (In press.)

Cloud-Edge Continuum
Application placement
Application management
Concluding remarks

https://www.youtube.com/watch?v=lCQ0AAYO0mQ

Some directions for future work

Security & privacy

Resilience

Continuum $\longleftrightarrow$ AI

Sustainability

Business models