Università di Pisa

Data-aware application placement and routing in the Cloud-IoT Continuum

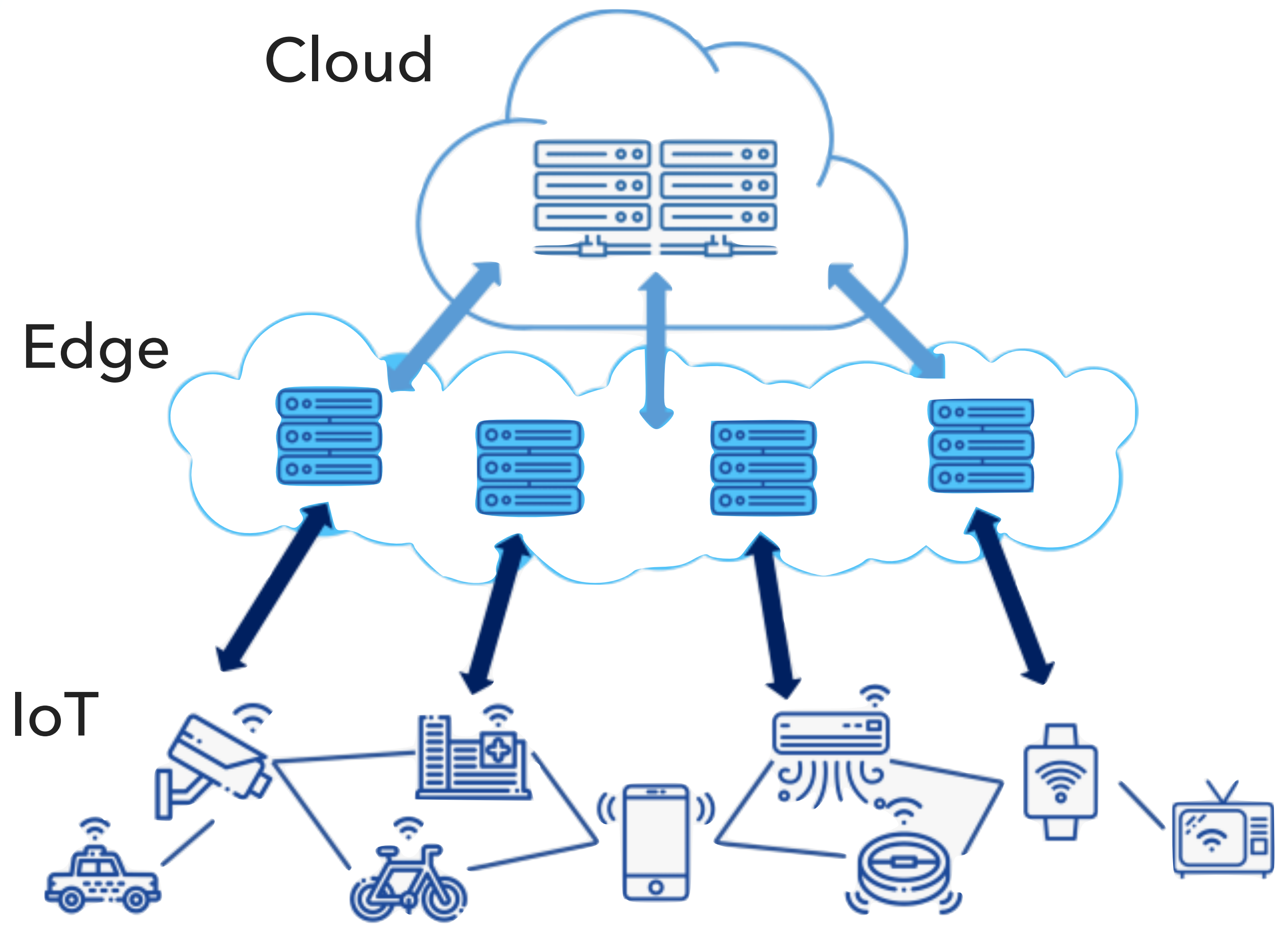
Jacopo Massa

jacopo.massa@phd.unipi.it

pages.di.unipi.it/massa

RESEARCH

CONTEXT

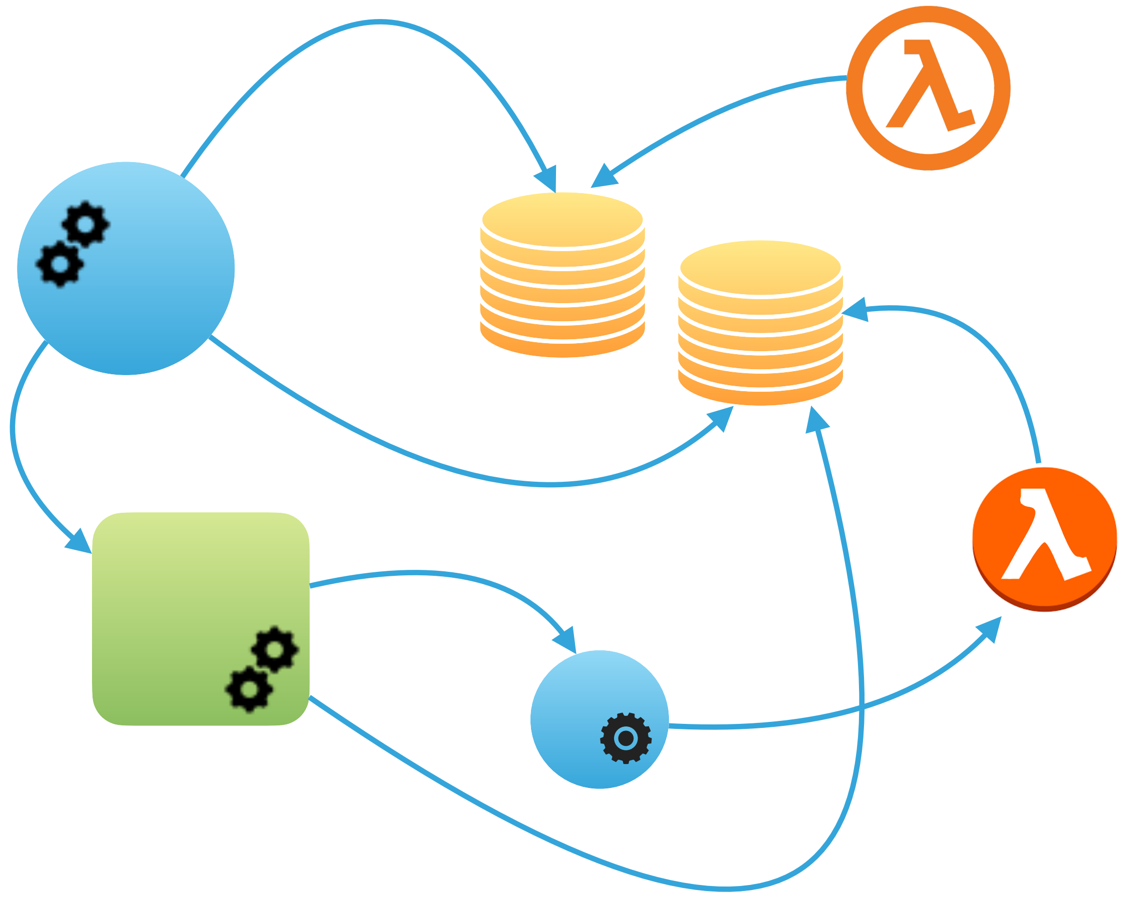


(~ hundreds)
↑ processing ↑ storage

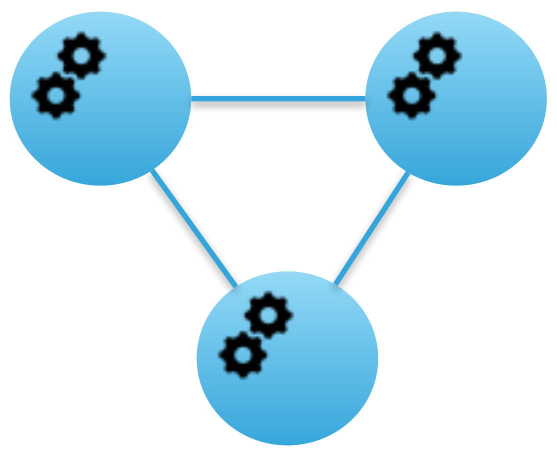
(~ thousands)

(~billions)
↓ processing ↓ storage

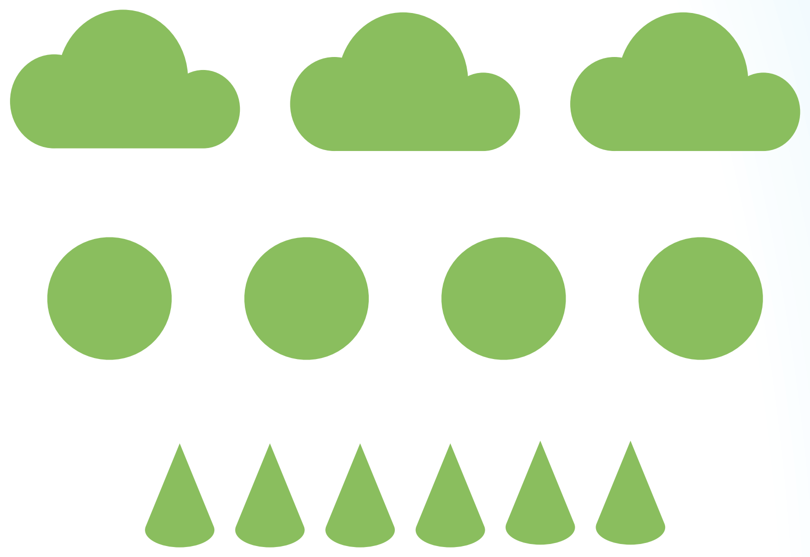
Microservice-based architecture

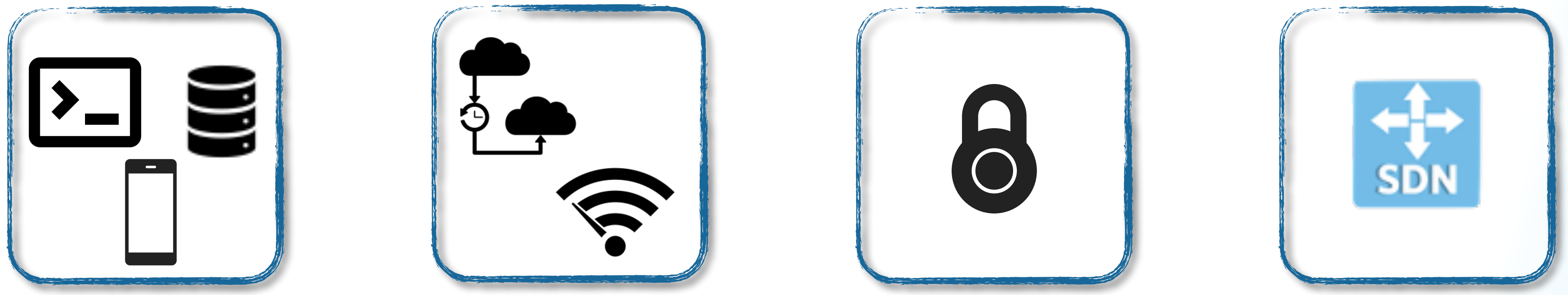


PROBLEM



Where to ***place*** application services and how to ***route traffic*** between them in a context- and QoS-aware manner?





Software, Hardware, IoT

Latency, Bandwidth

Security *(only a few)*

SDN Routing

[1] Salaht et al., *"An overview of service placement problem in fog and edge computing"*, CSUR, 2020
[2] Brogi et al. , *"How to place your apps in the fog: State of the art and open challenges"*, SPE, 2020

RELATED WORK [1,2]

OBJECTIVE

WE AIMED TO:

- *Devise* a declarative modelling of Cloud-IoT infrastructures and multi-service applications, to determine eligible *application placements and data traffic routings* across Cloud-IoT resources in a context-, QoS-, and **data-aware** manner.
- *Exploit* **continuous reasoning** to speed-up decision making at runtime.
- *Implement* and assess the proposed solution in a **Prolog open-source tool**.

DATA TYPE

Size

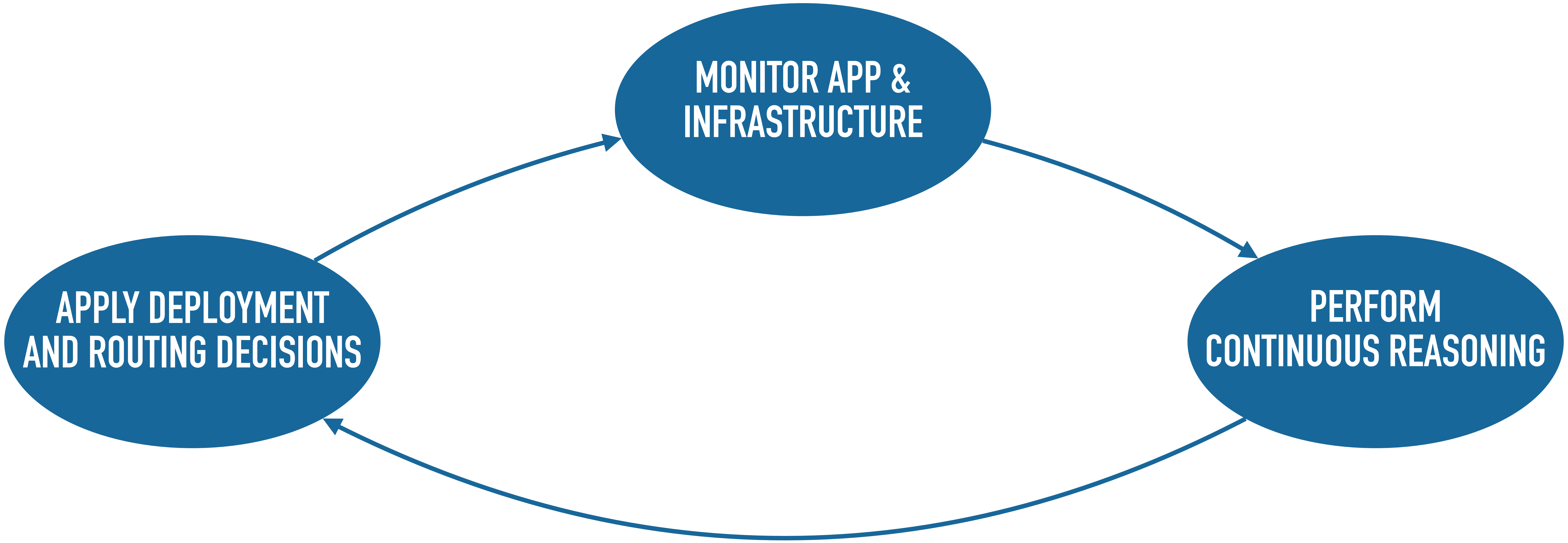
Transmission Rate

Sources *(sensors)*

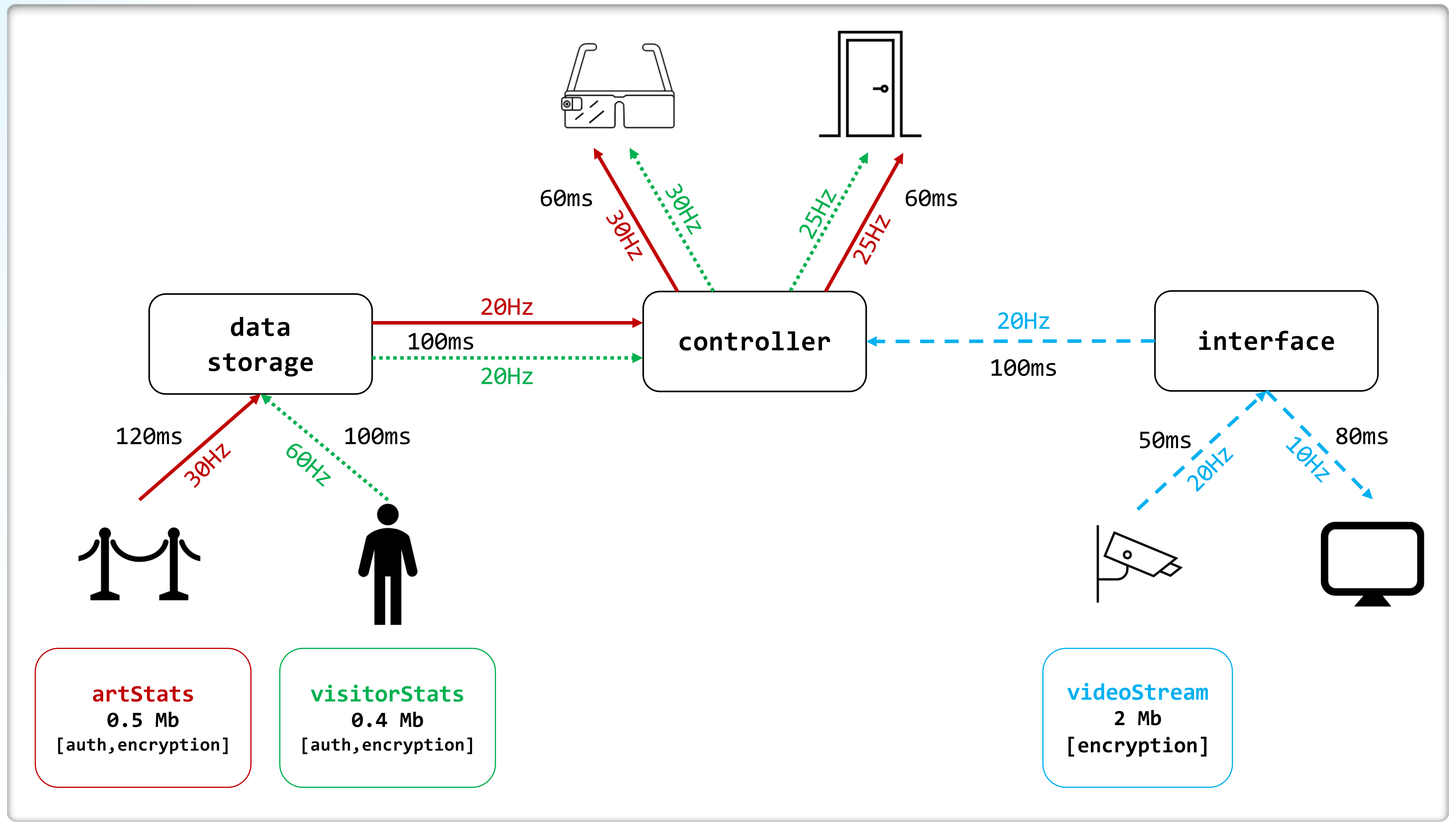
Targets *(actuators)*

CONTINUOS REASONING

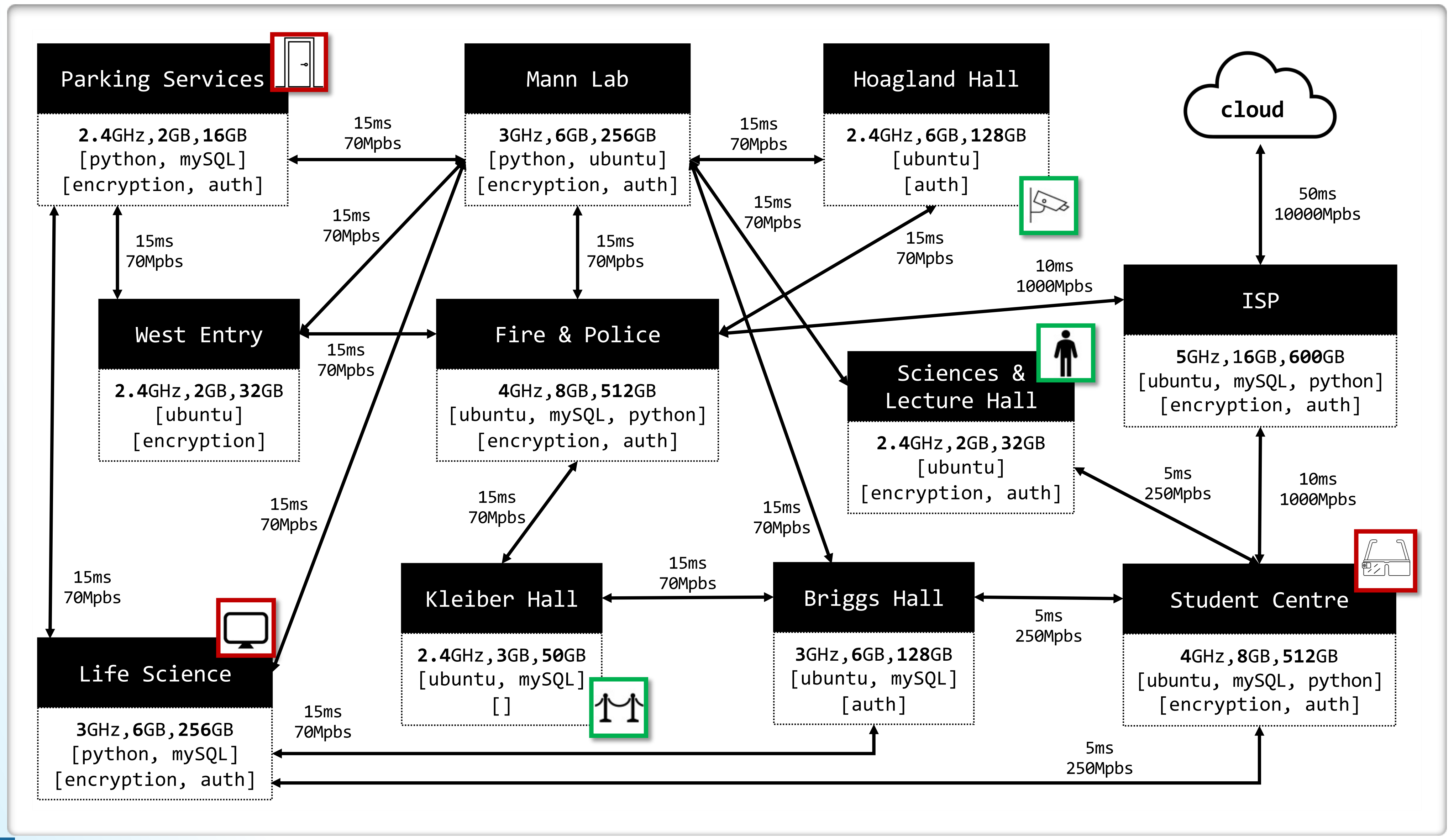
- **Adapt** placement and routing at **runtime.**
- Triggered at each *infrastructure/application change*.
- Partial re-deployment, focusing only on *suffering services.*
- **Speed-up** the whole placement and routing search process.



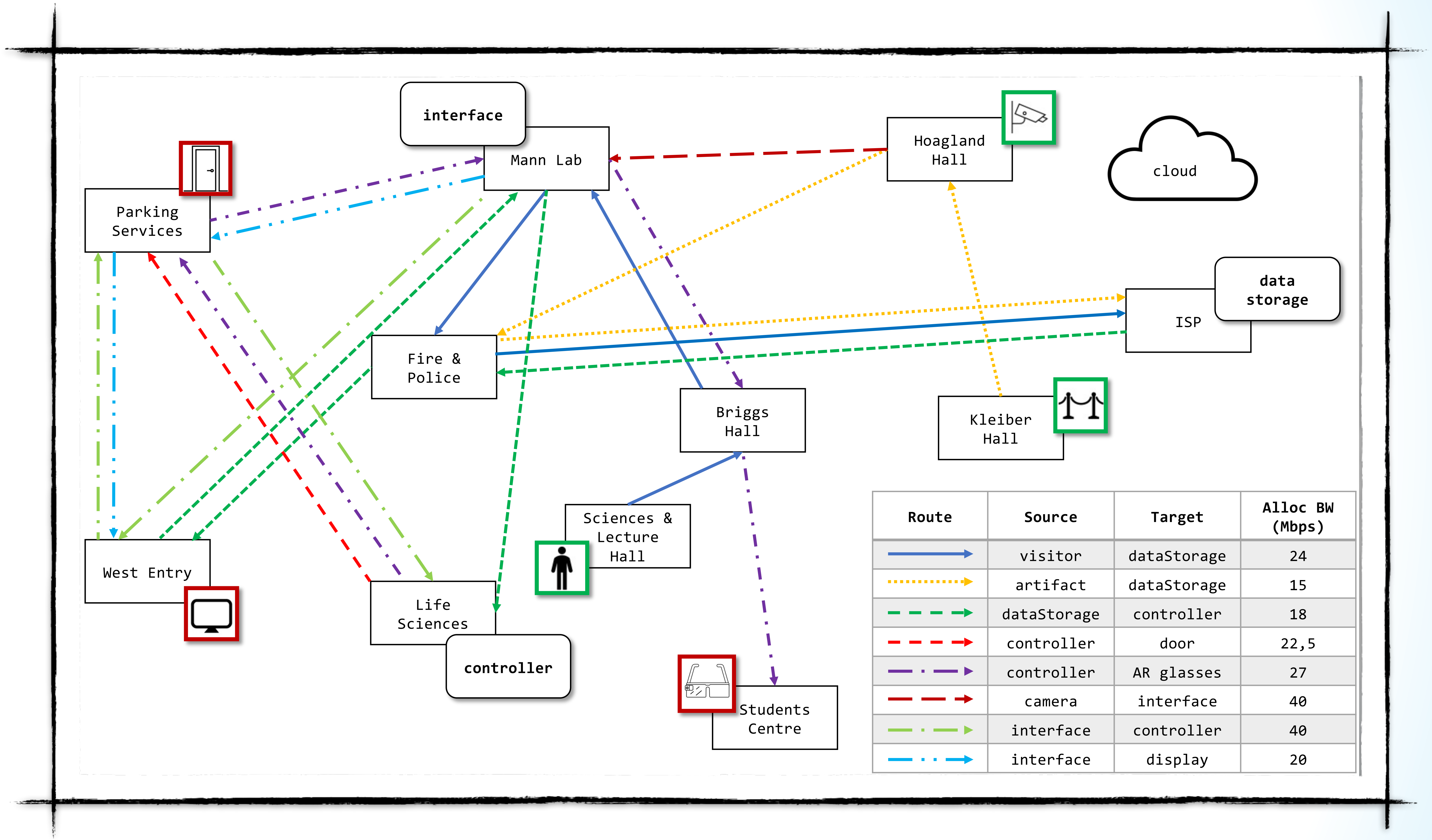
DA-PLACER: Data-Aware Placer



Example application



Example infrastructure



Route	Source	Target	Alloc BW (Mbps)
	visitor	dataStorage	24
	artifact	dataStorage	15
	dataStorage	controller	18
	controller	door	22,5
	controller	AR glasses	27
	camera	interface	40
	interface	controller	40
	interface	display	20

DA-Placer Output

DA-PLACER (II)

```
application(museuMonitor, [interface, controller, dataStorage]).

% service(ServiceId, [SWReqs], [HWReqs], [DataIds]).
service(interface, [ubuntu], (2.4, 4, 128), [videoStream]).
service(controller, [python, mySQL], (3, 6, 256), [artStats, visitorStats, videoStream]).
service(dataStorage, [mySQL, ubuntu], (5, 4, 512), [artStats, visitorStats]).

% dataType(DataId, Size, [SecReqs]).
dataType(artStats, 0.5, [encryption]).
dataType(visitorStats, 0.4, [auth, encryption]).
dataType(videoStream, 2, [auth, encryption]).

%e2e(A, B, MaxLatency, [(DataId, DataRate)]).
e2e(rArt, dataStorage, 120, [(artStats, 30)]).
e2e(rVst, dataStorage, 100, [(visitorStats, 60)]).
e2e(dataStorage, controller, 100, [(artStats, 20), (visitorStats, 20)]).
e2e(controller, rGls, 60, [(artStats, 30), (visitorStats, 30)]).
e2e(controller, rDor, 60, [(artStats, 25), (visitorStats, 25)]).
e2e(rCam, interface, 50, [(videoStream, 20)]).
e2e(interface, rVid, 80, [(videoStream, 10)]).
```

Example application

```
sensor(cam20, camera, [videoStream]).
actuator(video3, display).
actuator(door27, smartdoor).
actuator(glass4, smartphone).

node(parkingServices, [python, mySQL], (2.4, 2, 16), [encryption, auth], [door27]).
node(westEntry, [ubuntu], (2.4, 2, 32), [encryption], [video3]).
node(kleiberHall, [ubuntu, mySQL], (2.4, 3, 50), [], [art42]).
node(hoaglandAnnex, [ubuntu], (2.4, 6, 128), [auth], [cam20]).
node(briggsHall, [ubuntu, mySQL], (3, 6, 128), [auth], []).
node(mannLab, [ubuntu, python], (3, 6, 256), [encryption, auth], []).
node(lifeSciences, [python, mySQL], (3, 6, 256), [encryption, auth], []).
node(sciencesLectureHall, [ubuntu, mySQL], (3, 6, 256), [encryption, auth], [vst38]).
node(firePolice, [ubuntu, mySQL, python], (4, 8, 512), [encryption, auth], []).
node(studentCenter, [ubuntu, mySQL, python], (4, 8, 512), [encryption, auth], [glass4]).
node(isp, [ubuntu, mySQL, python], (5, 16, 600), [encryption, auth], []).
node(cloud, [ubuntu, mySQL, python], (6, 32, 10000), [encryption, auth], []).

link(isp, firePolice, 10, 1000).
link(firePolice, isp, 10, 1000).
```

Example infrastructure

```
:- daplacer(museuMonitor, Placement, Routes).

% on(Service, Node)

Placement = [on(dataStorage, isp),
             on(controller, lifeSciences),
             on(interface, mannLab)],

% ((source, target), AllocatedBandwidth, Route)

Routes = [((dataStorage, controller), 18, [isp, firePolice, westEntry, mannLab, lifeSciences]),
          ((interface, controller), 40, [mannLab, westEntry, parkingServices, lifeSciences])
          ...
```

DA-Placer Output

NEXT STEPS

PROGRESS W.R.T THE STATE-OF-THE-ART

- Prolog prototype (https://github.com/di-unipi-socc/daplacer) that can be used to:
  - model data, services and IoT devices in a **data-aware manner,**
  - **jointly place** both data and services.
- security requirements
- runtime adaptation (***continuous reasoning*** *approach)*

LIMITATIONS AND FUTURE WORK

- extending the model to account for **serverless/FaaS**
- **multi-objective optimisation** (evaluate the *goodness* of a solution, "*greenness*" included)
- **further management decision** (*scalability, undeploy, Osmotic*)
- **identify interesting application contexts** (*AI applications, …*)
- **validate placement and routing** solutions on real testbeds
- **increase prototype usability** (e.g. *user-friendly tools*)

THANK YOU FOR YOUR ATTENTION!