Session 4: IoT

FORTH
INSTITUTE OF COMPUTER SCIENCE

TELECOM SudParis

IP PARIS

UNIVERSITY OF CRETE

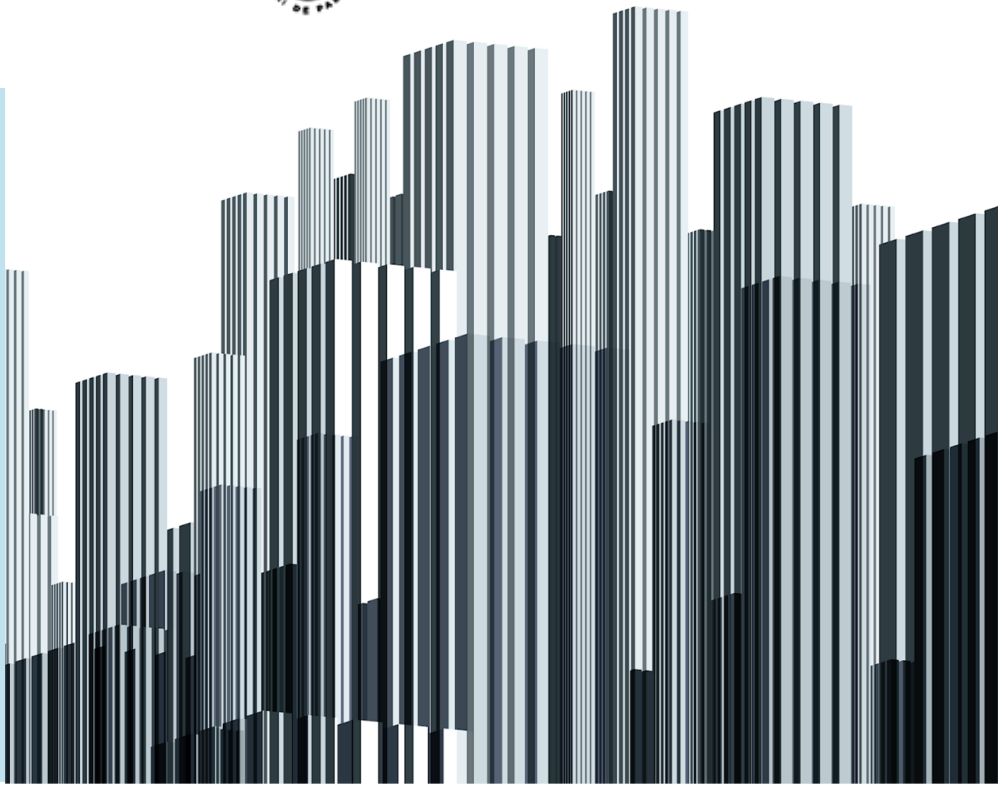
EPAnEK 2014–2020
OPERATIONAL PROGRAMME
COMPETITIVENESS • ENTREPRENEURSHIP • INNOVATION

ComDeX: A Context-aware Federated Platform for IoT-enhanced Communities

Nikolaos Papadakis, Georgios Bouloukakis, Kostas Magoutis

17th ACM International Conference On Distributed And Event-Based Systems

29th June 2023 Neuchatel, Switzerland



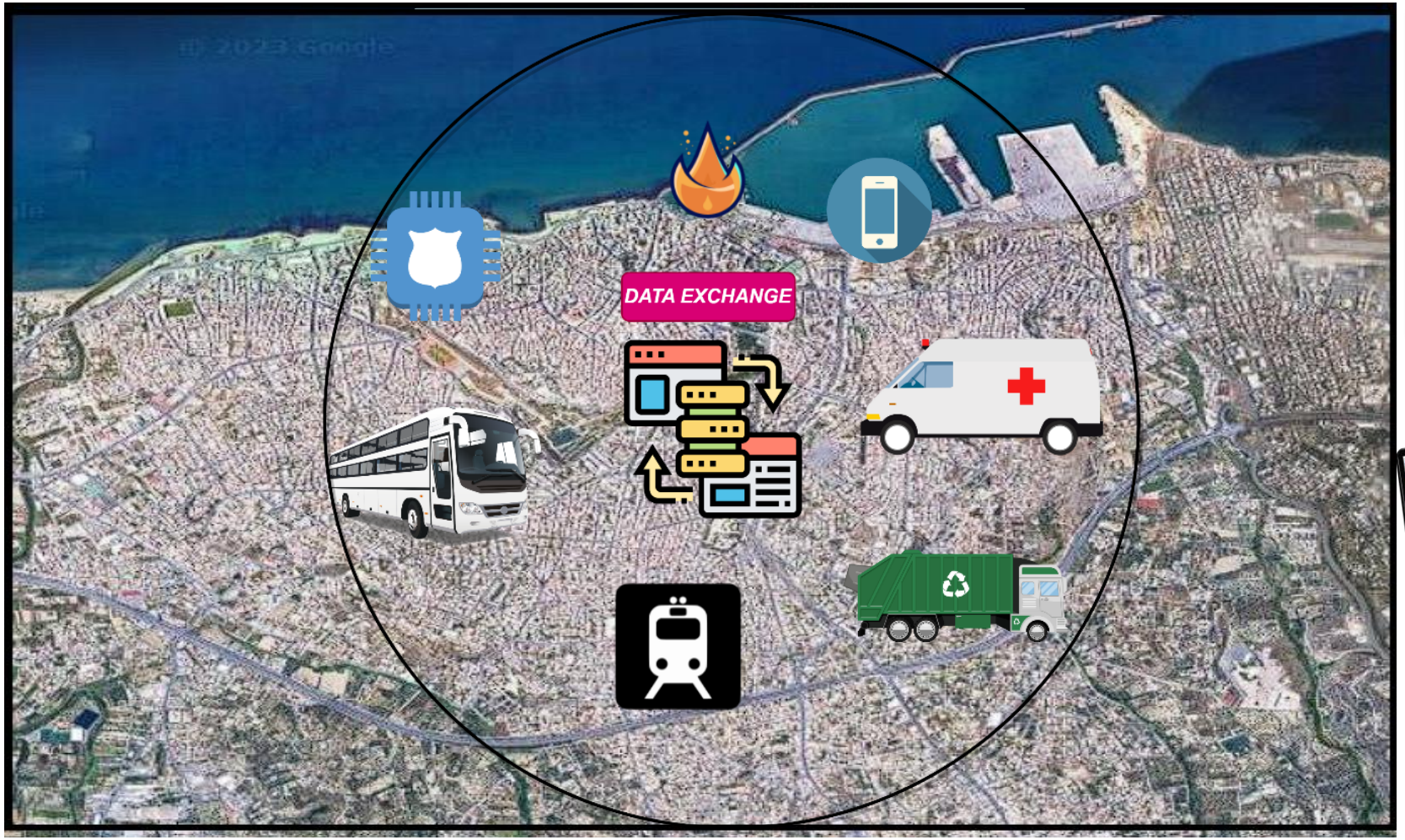
The imminent rise of smart cities and IoT communities

Efficient | Sustainable | Safe

Smart Cities embrace a variety of Internet and mobile technologies in order to improve the quality of life for their citizens, transform the efficiency of public services, and generate new sources of growth for companies

How easy is this?

Motivation Scenario: Complexity of Confluence - Challenging Aspects of Smart Communities

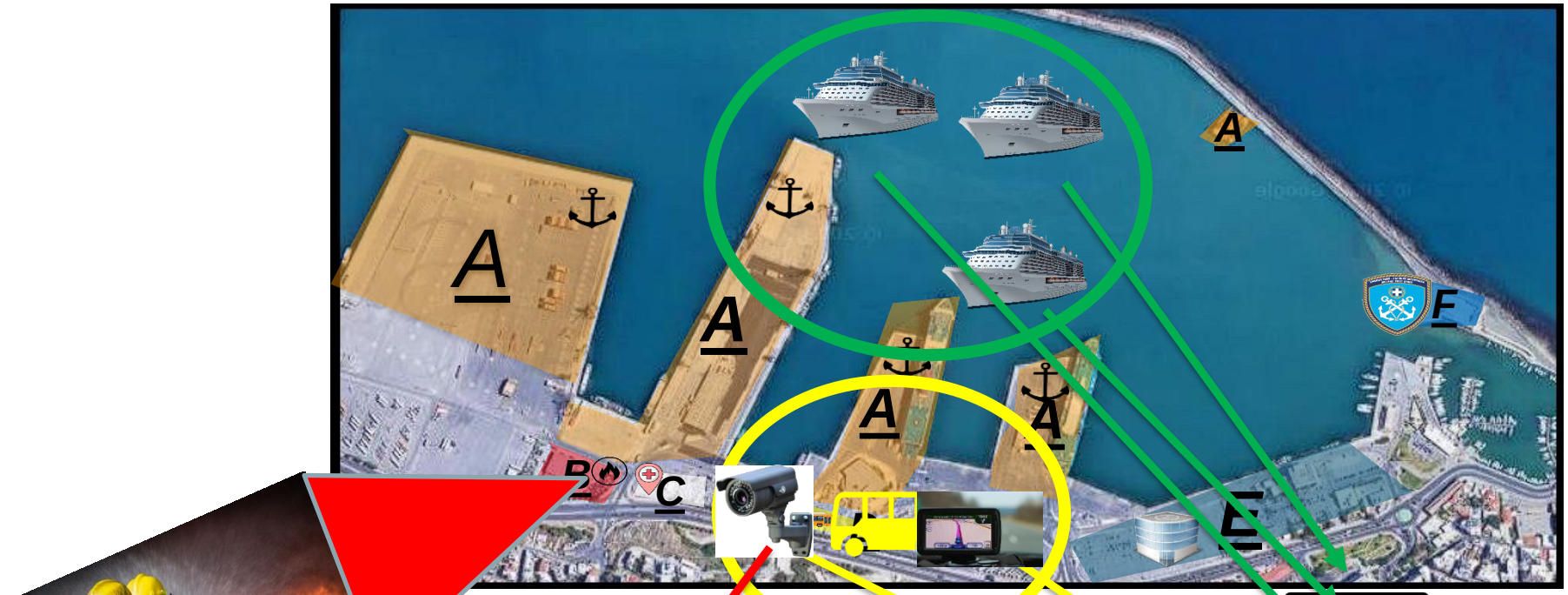


Port authority (A)
Firefighters (B)
Medic Station (C)
City Bus Transportation (D)
Private Businesses (E)
Coast Guard (F)

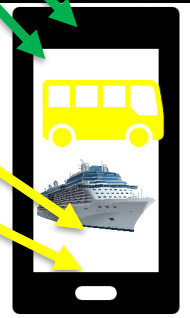



Each have their own applications, devices, data, servers etc.

Motivation: Widespread IoT Applications



Fire Handling Monitor



Application
Normal User

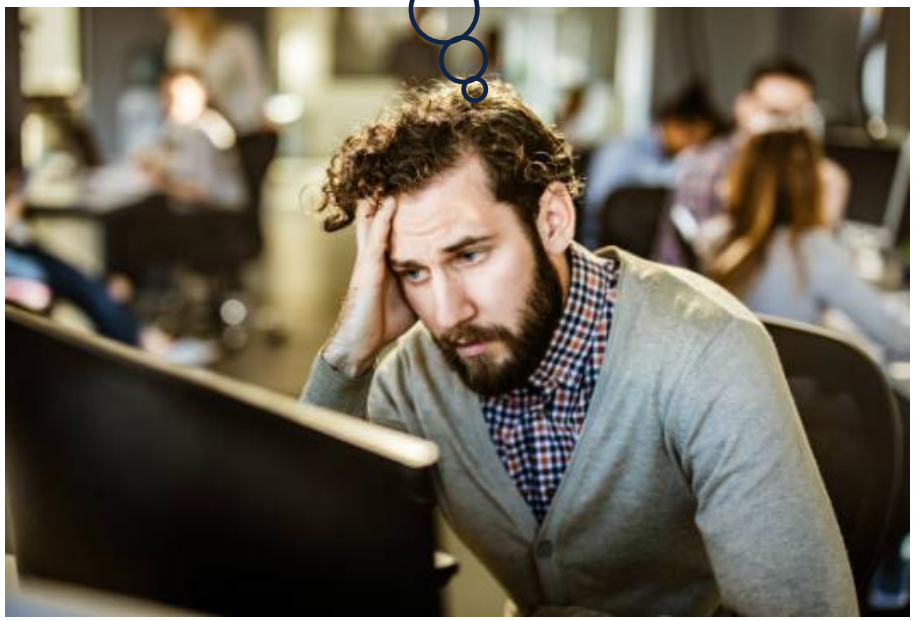
Smart Transportation Application

Motivation: Widespread IoT Applications, a challenging endeavor

We want to be able to easily access information while withholding our own

We are using a proprietary data model, we only want to share bus timetables!

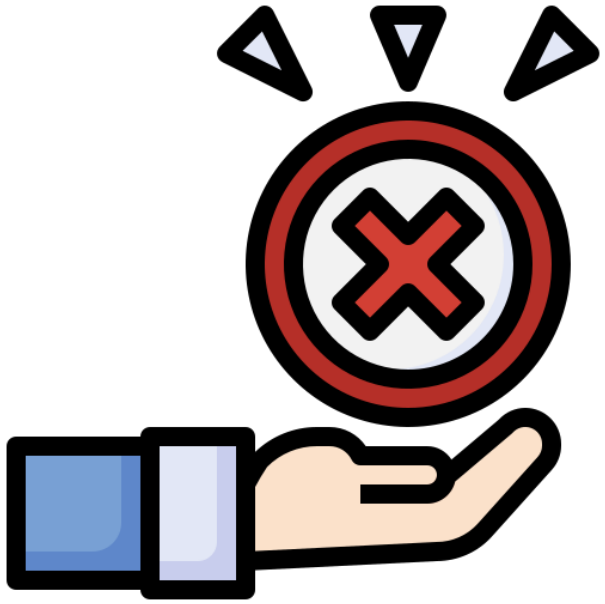
Information Integration?
How to do handle the distributed data exchange?
Selective Sharing?
Data sovereignty?



Requirements for enabling widespread IoT applications

- IoT systems must be designed to enable:
- **Cross-community collaboration:** Enable seamless connection of diverse stakeholders and enable efficient data exchange
- **Openness and Interoperability**: IoT systems should provide interoperability between different devices, services, and applications. They should have open standards and protocols to facilitate seamless interaction across different platforms.
- **Data Sovereignty:** Data providers should have control over their data and its exposure
- **Scalability:** IoT systems should be scalable to manage the rapid increase in the number of interconnected devices and the data they produce.

Existing Solutions



- Lack service programming models or rely solely on their own private data models and interfaces.

Tricomi et al., SMARTCOMP, 2019.
Cirillo et al., IEEE Internet of Things Magazine, 2019.

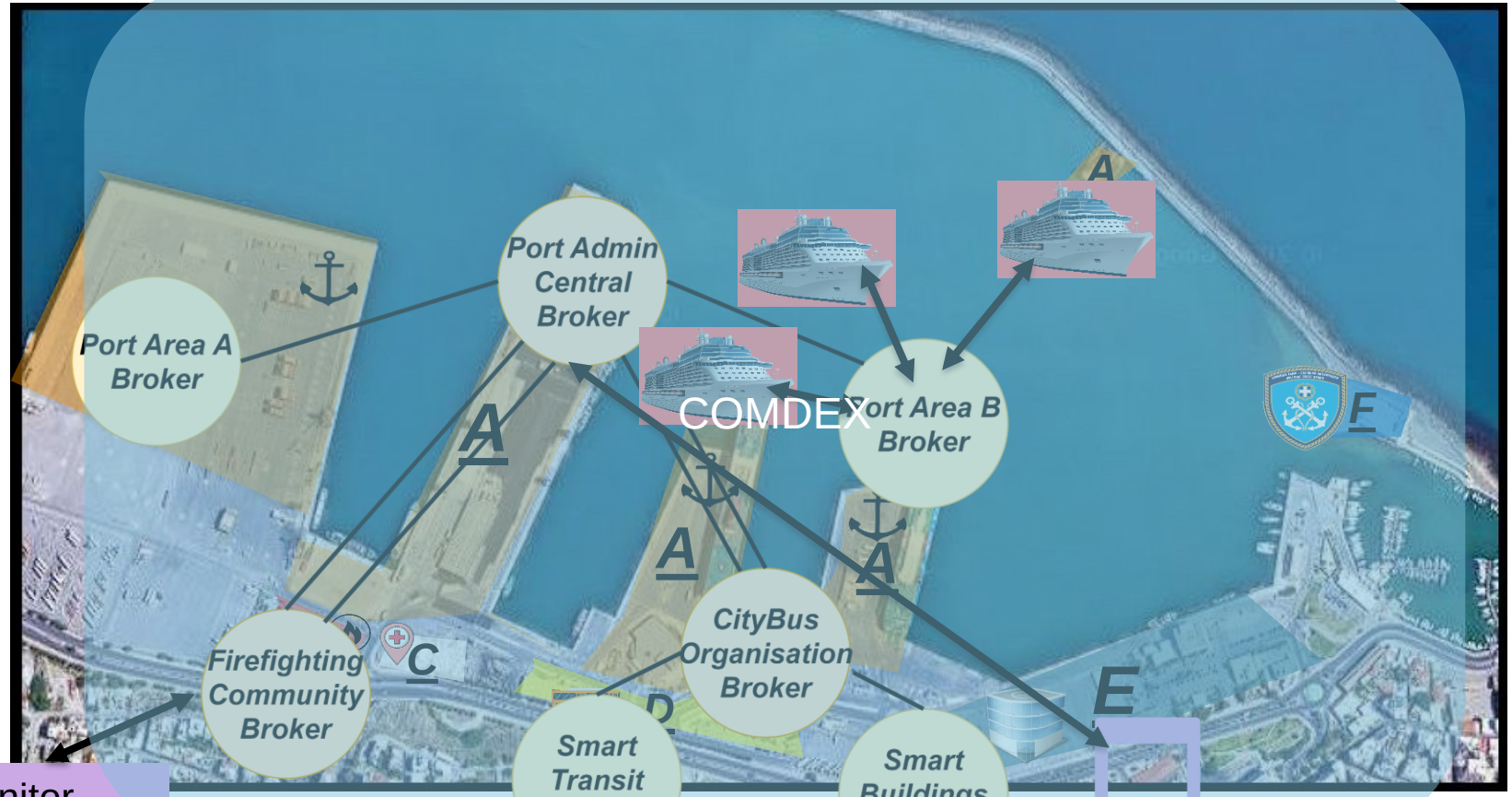
- Offer data dissemination techniques without appropriate considerations for selective sharing and maintaining data sovereignty

Salehi et al., Middleware, 2020.
Morelli et al., FedCSIS, 2020.

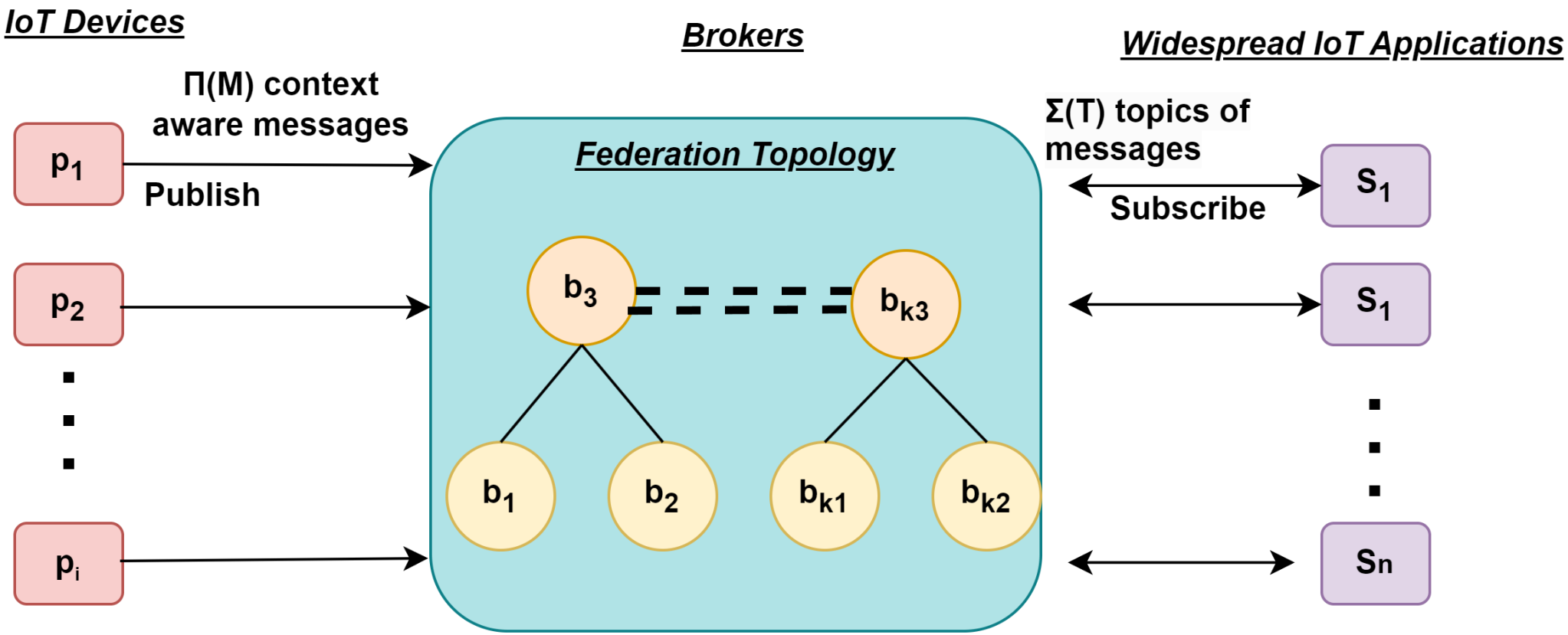
- Focus solely on a single smart space. They don't offer a solution to challenges faced in a federation of smart communities

Adal et al., BuildSys '21, 2021.
Yus et al., ACM Transactions on Internet Technology – TOIT, 2022.

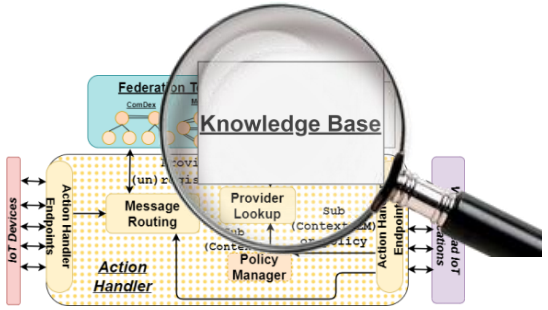
Introducing ComDeX: High Level View



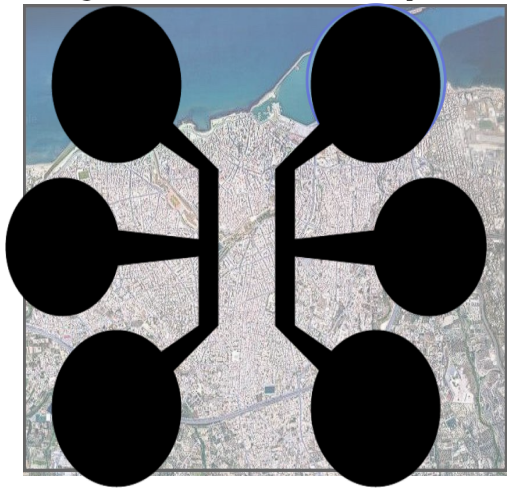
Introducing ComDeX: High Level View

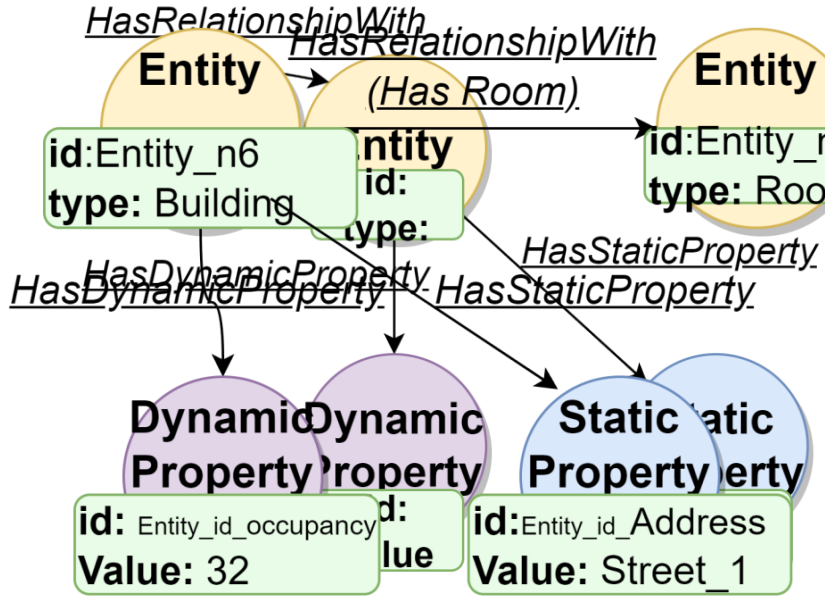


ComDeX Architecture: Modeling Static/Dynamic Data

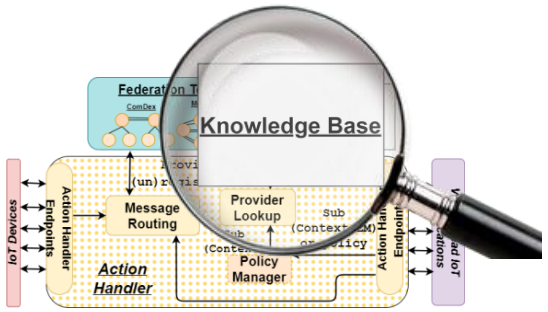


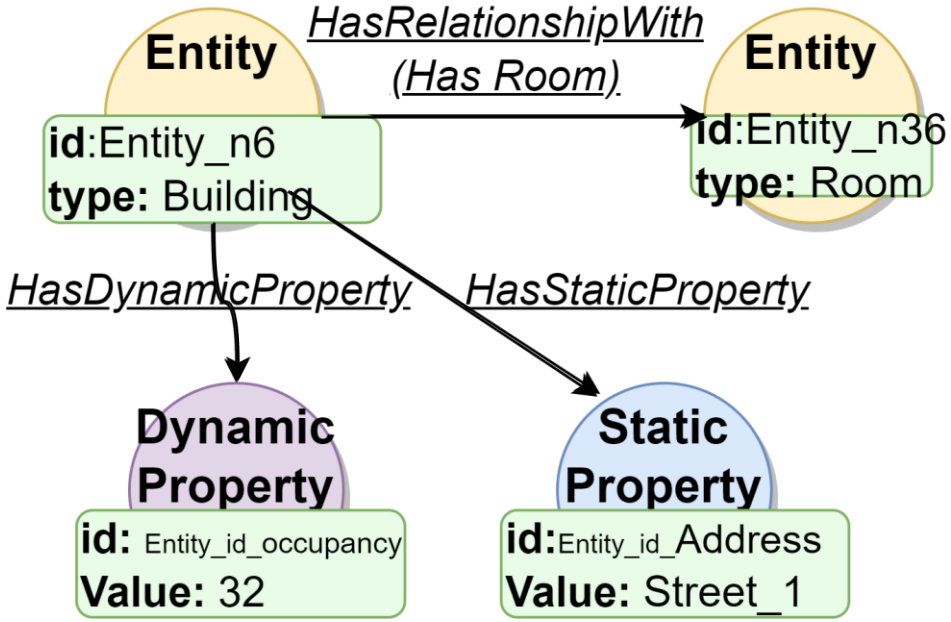
Information of real-world objects and concepts





Context-aware pub/sub messaging





Algorithm 1 Algorithm to split of data in property graph into messages on specific topics

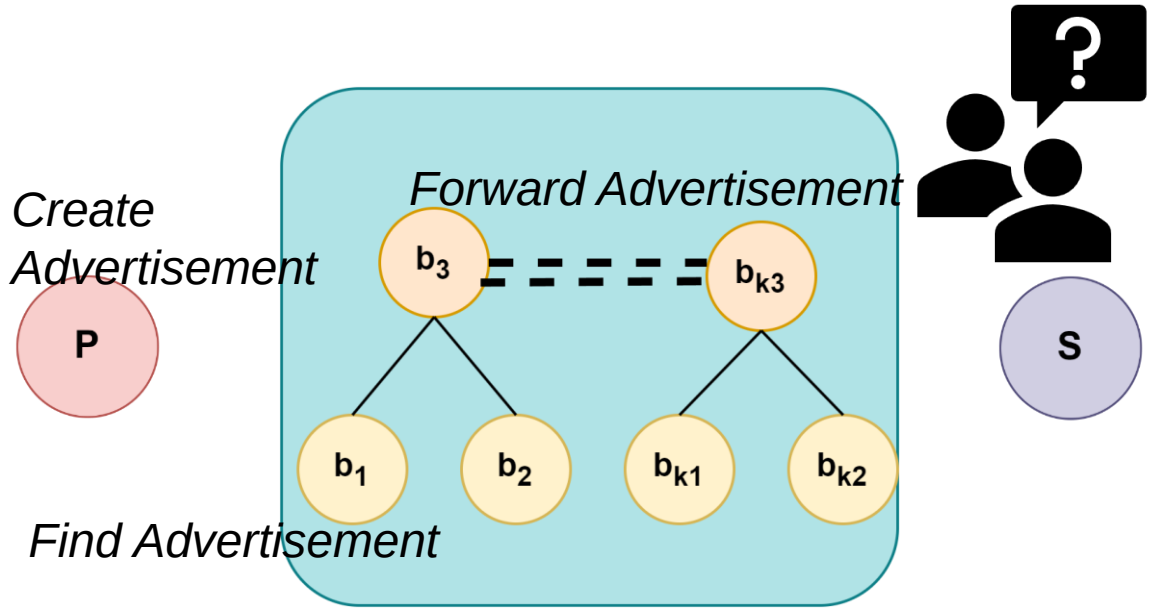
```
1: //Input: Property Graph //Output: ComDEX Messages
2: procedure SPLITTING P.GRAPH:
3:     for each node x where node_type equals "Entity" do
4:         ε_j ← x
5:         ε_j.attr.ea_x ← edges.of.x
6:         for each ε_j.attr.ea_x do
7:             t_j ← ε_j.type + ε_j.id + ea_x.type + ea_x.id
8:             m_j.topic ← t_j
9:             m_j.payload ← ea_x.value
10:            print(m_j)
11:        end for
12:    end for
13: end procedure
```

Messages
Topic: *Building/Entity_n6/HasDynamicProperty/Occupancy* ***Payload:*** *value:32*
Topic: Building/Entity_n6/HasStaticProperty/address **Payload:** value:Street_1
Topic: Building/Entity_n6/HasRelationshipWIth/HasRoom **Payload:** value:Street_1

Context Aware Distributed Data Discovery

Messages
Topic: *Building/Entity_n6/HasDynamicProperty/Occupancy* ***Payload:*** *value:32*
Topic: Building/Entity_n6/HasStaticProperty/address **Payload:** value:Street_1
Topic: Building/Entity_n6/HasRelationshipWIth/HasRoom **Payload:** value:Street_1

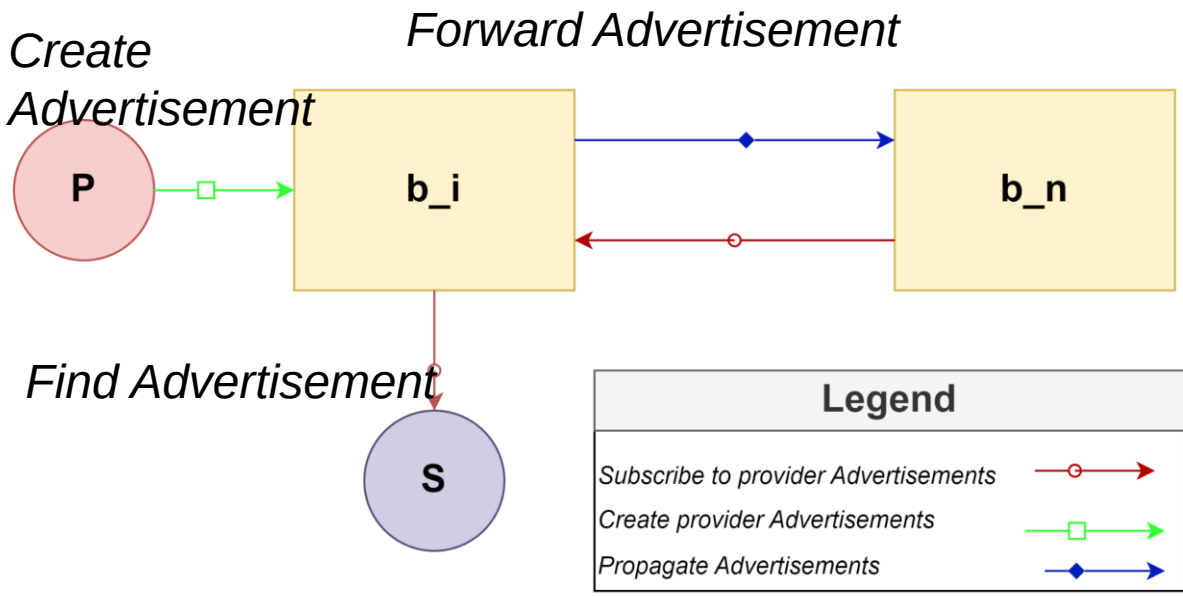
Advertisement Message
Per Entity Type: *provider/ connection_info/entity_type* ***Per Entity ID:****provider/connection_info/entity_type/entity_id*
Example
Per Entity Type: *provider/broker1/port/Building* ***Per Entity ID:****provider/broker1/port/Building/Enitity_n6*



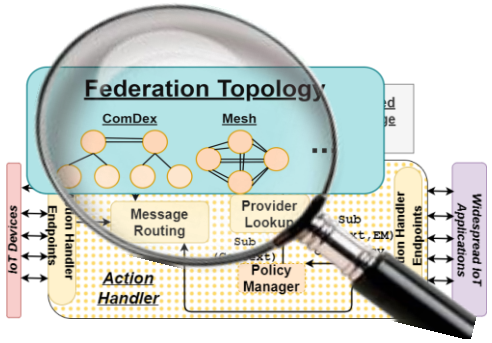
Context Aware Distributed Data Discovery

Messages
Topic: *Building/Entity_n6/HasDynamicProperty/Occupancy* ***Payload:*** *value:32*
Topic: Building/Entity_n6/HasStaticProperty/address **Payload:** value:Street_1
Topic: Building/Entity_n6/HasRelationshipWIth/HasRoom **Payload:** value:Street_1

Advertisement Message
Per Entity Type: *provider/ connection_info/entity_type* ***Per Entity ID:****provider/connection_info/entity_type/entity_id*
Example
Per Entity Type: *provider/broker1/port/Building* ***Per Entity ID:****provider/broker1/port/Building/Enitity_n6*



Distributed Broker Setup: Federation Topology

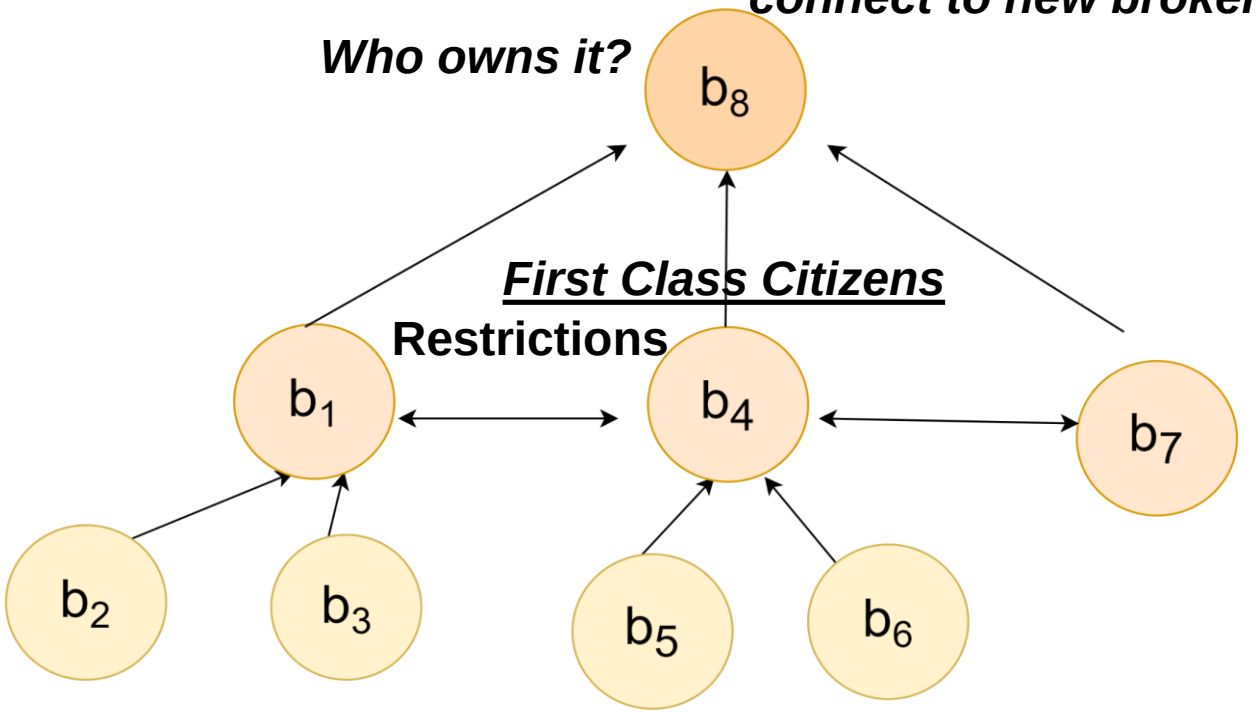


Existing Community Applications have to connect to new broker to discover data!

Who owns it?

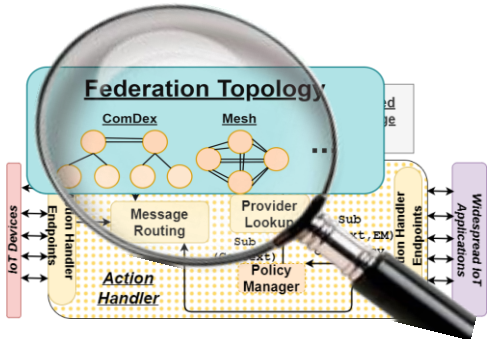
First Class Citizens

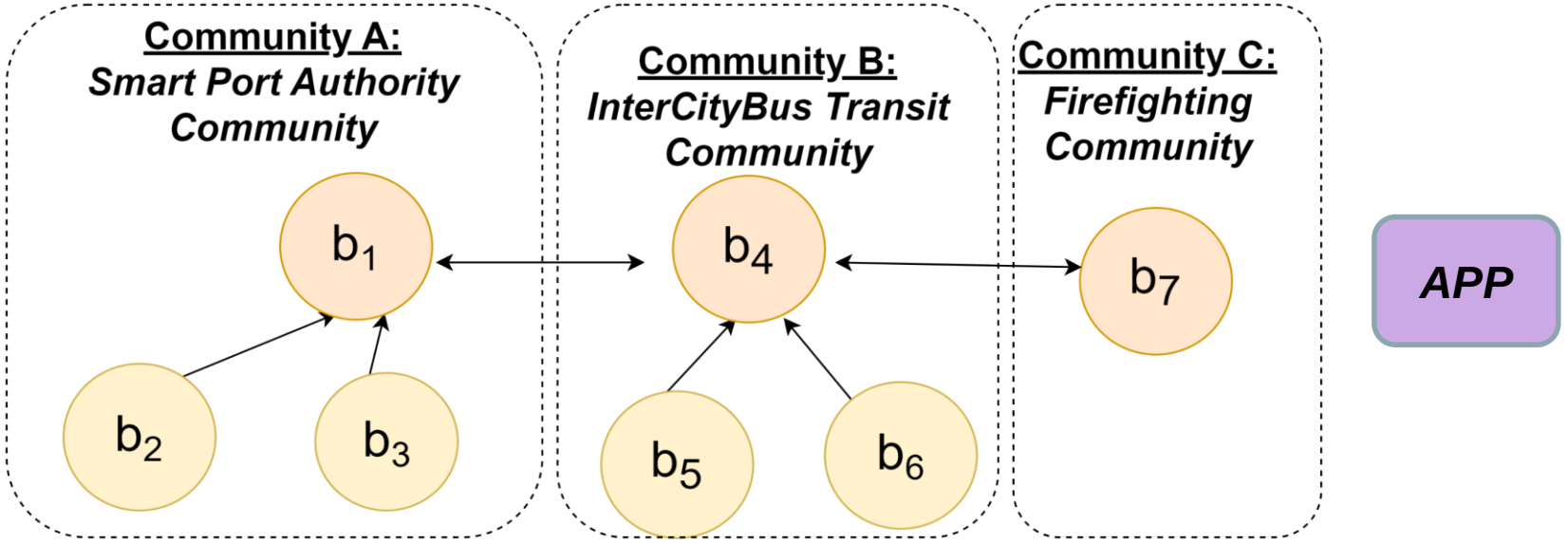
Restrictions



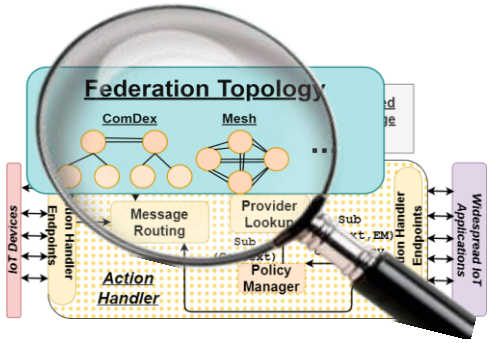
APP

Distributed Broker Setup: Federation Topology

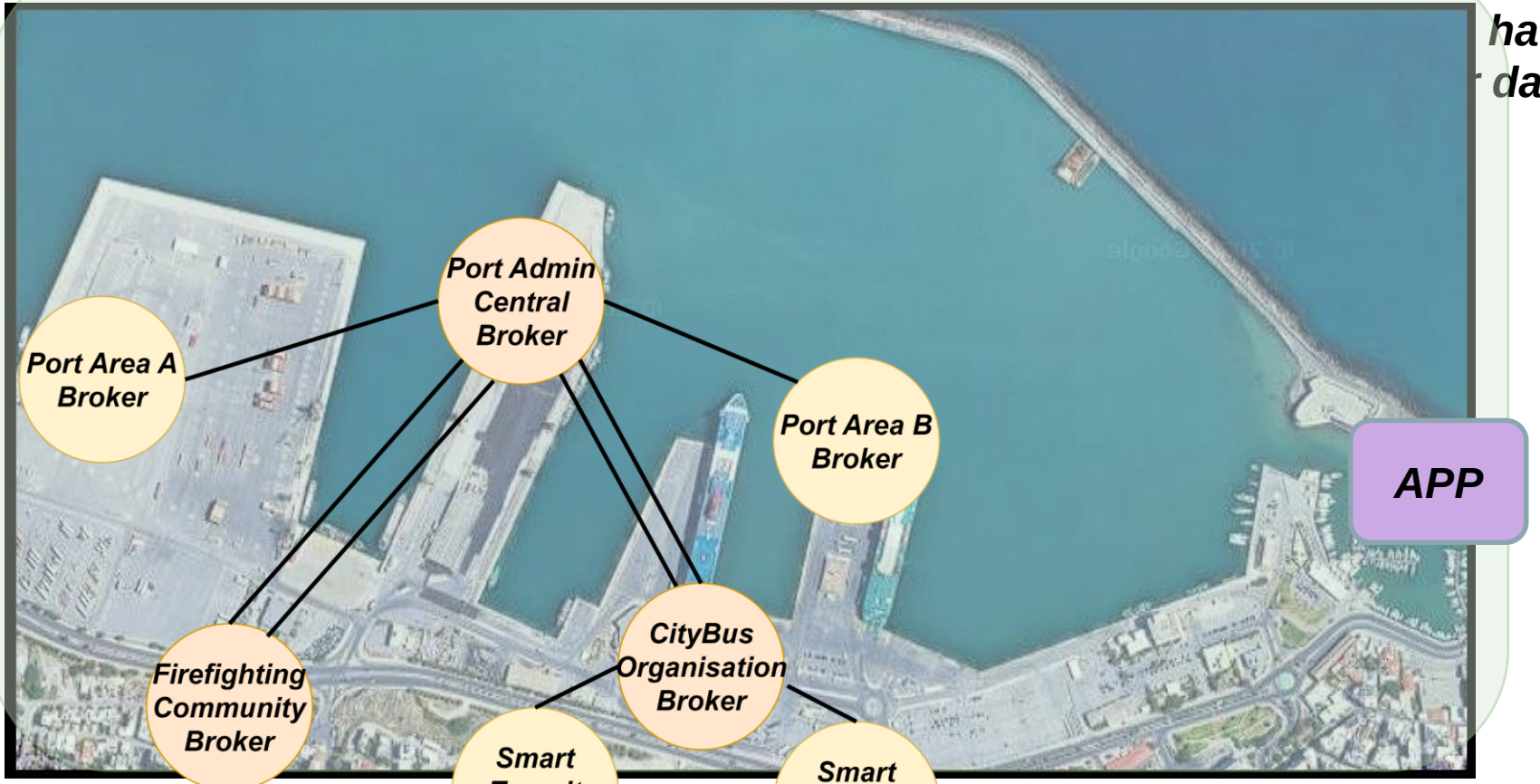




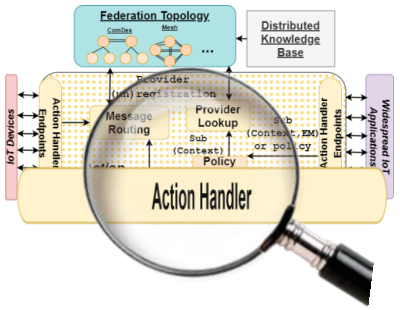
Distributed Broker Setup: Federation Topology



have to data!

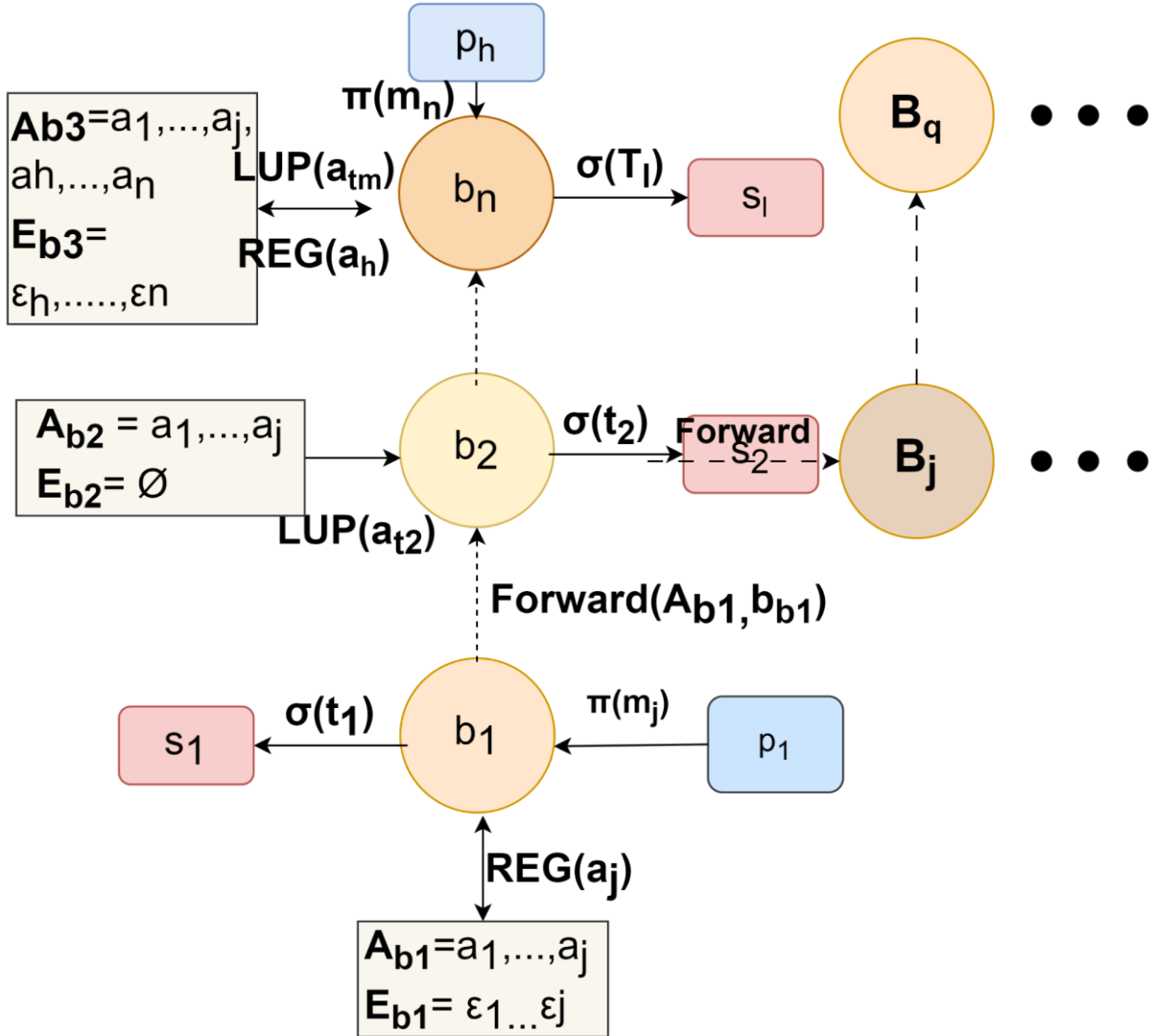


ComDeX Architecture: Action Handler

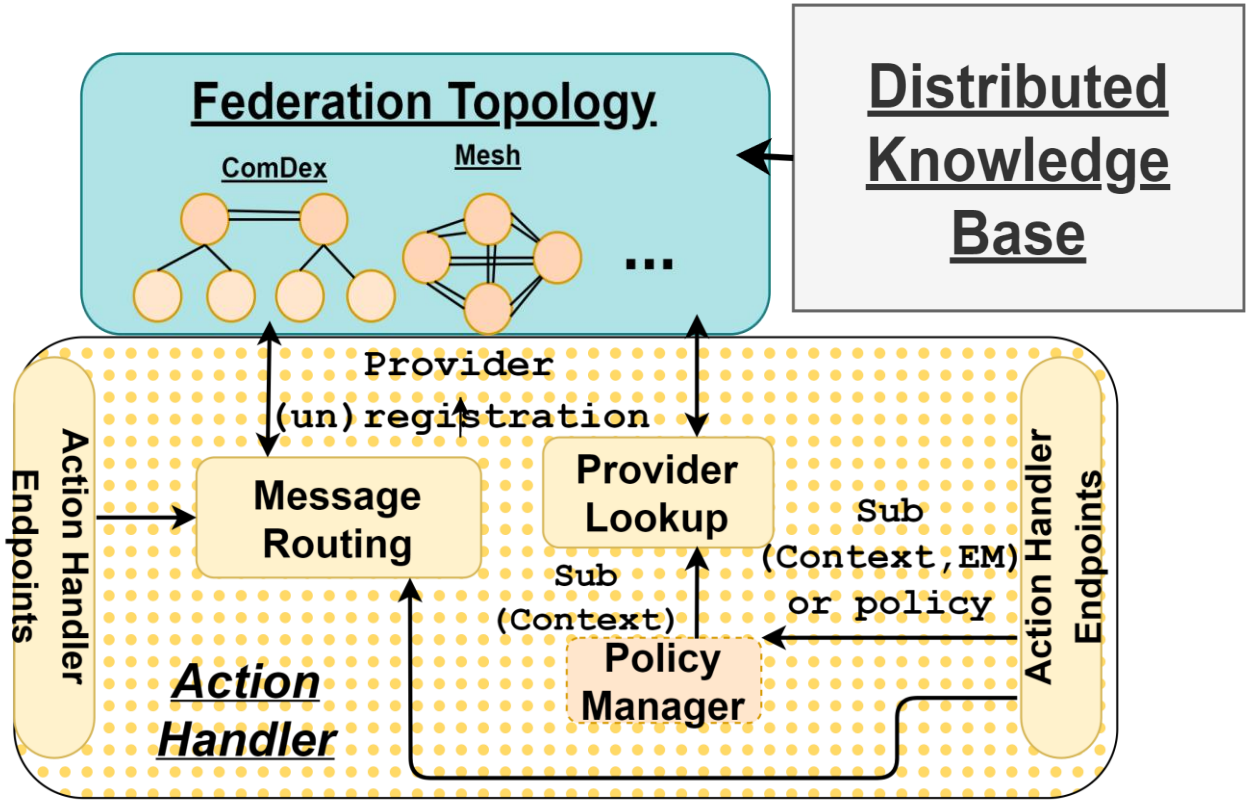


Actions:

- ✓ Publish Data **(π)**
- ✓ Request/Subscribe to Data **(σ)**
- ✓ Provider Registration **(REG)**
- ✓ Provider Lookup **(LUP)**



From theory to practice: Prototype Implementation



From theory to practice: Prototype Implementation

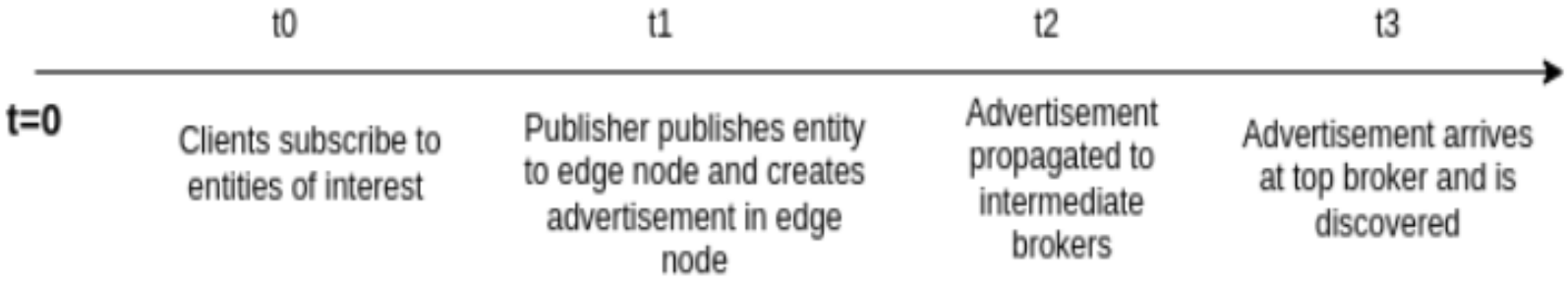
Experimental Evaluation: Testbed

Functionallity	Instance Type	Instance Family	Instance Size	VCPUs	Memory (GIB)	Network Performace
Brokers	c5.large	c5	large	2	4	Up to 10 Gigabit
Subscribers/Publishers	t3.nano	t3	nano	2	0.5	Up to 5 Gigabit
Publisher (§8.2)	t3.large	t3	large	2	8	Up to 5 Gigabit

amazon web services

EC2

Experimental Evaluation: Metrics



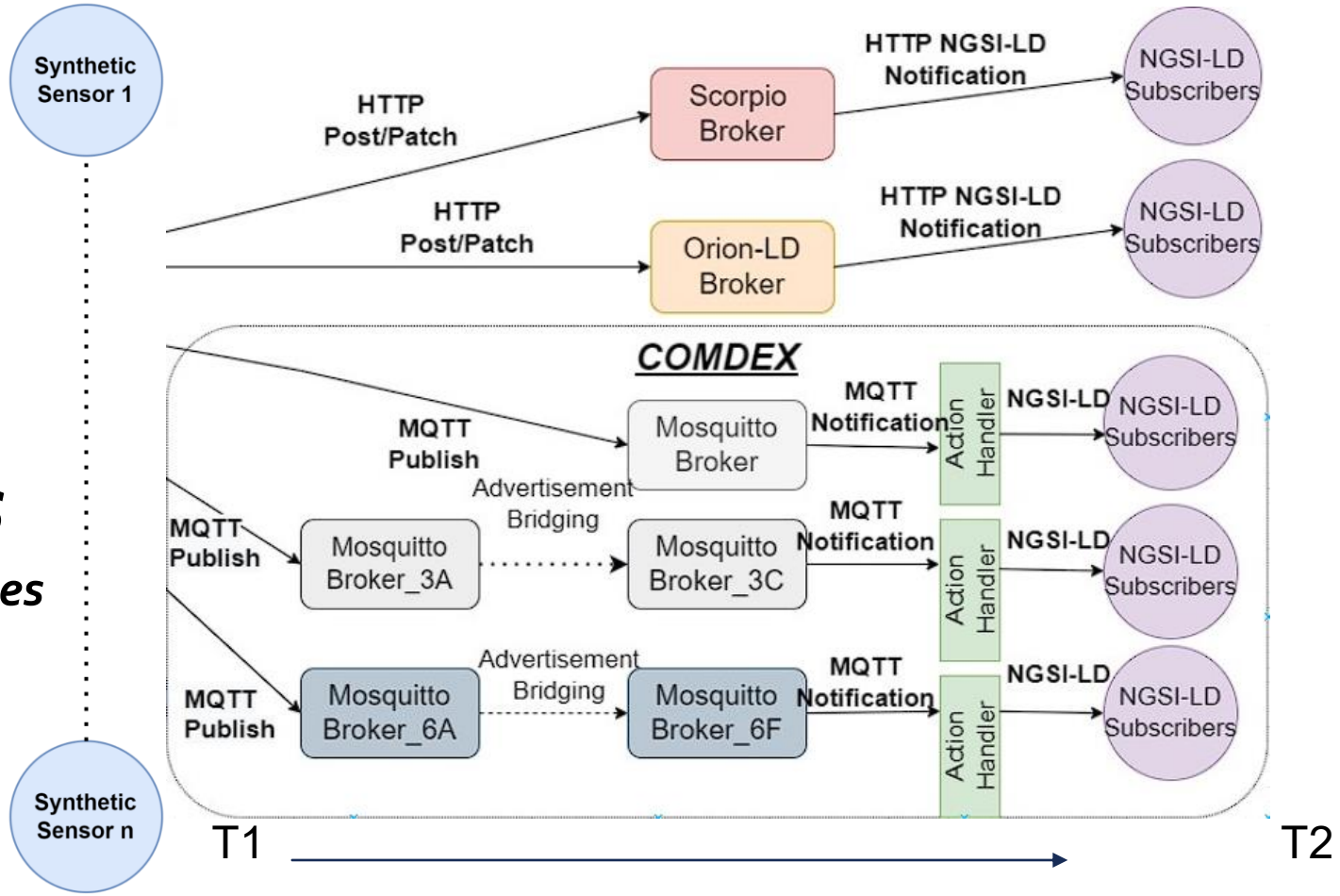
Advertisement Installation Time= arrival_time_at_top_broker – creation_time_at_edge_broker
Advertisement Installation Time = t3 - t1

Subscription Notification Latency: the time, from the creation of a Publication πj at a broker bk until its reception by an interested subscriber sj

Experiment 1: Experimental Setup

2000 entities

Random values



Subscription Notification Latency

Experiment 1: Experimental Results Normal Case

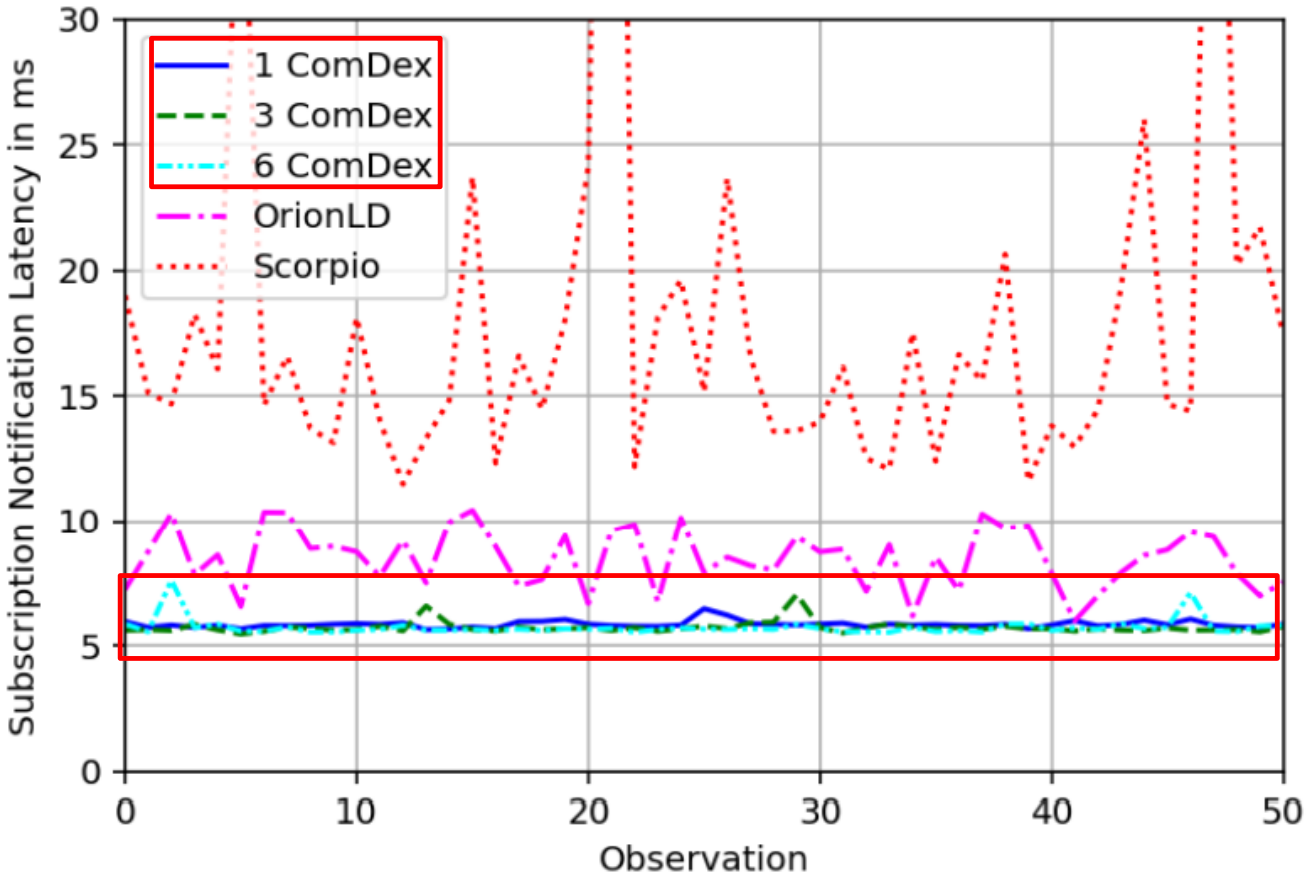
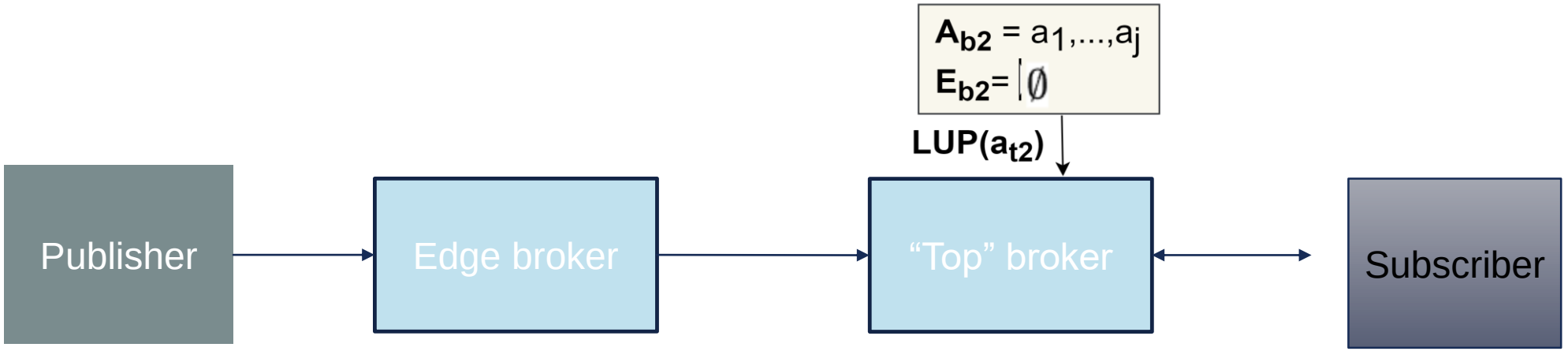


Figure 8.2: Subscription notification latencies (§8.1.1)

Experiment 2: Worst Case Scenario

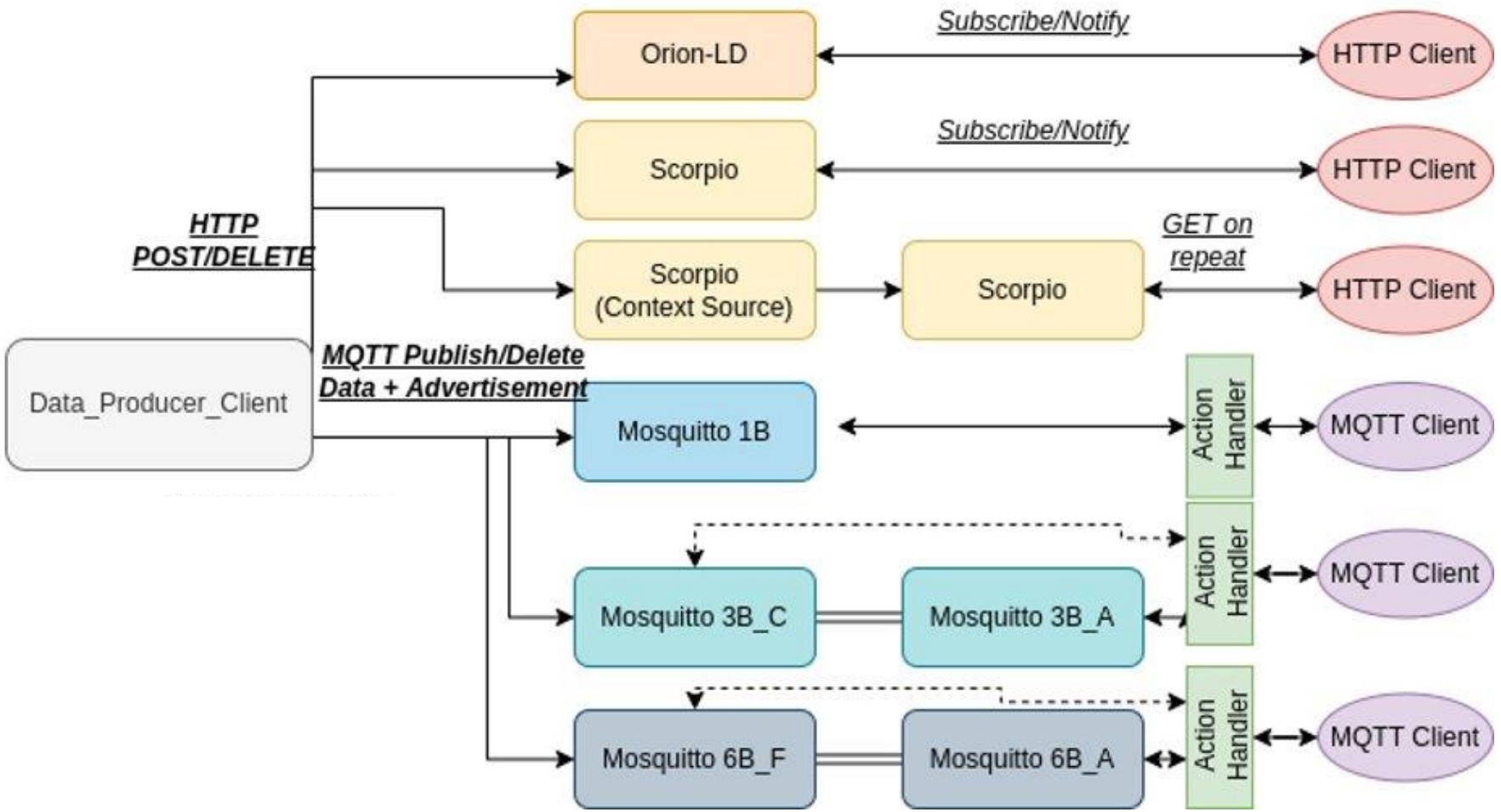
Client needs to "re-discover" the data source and connect to it for every entity requested.



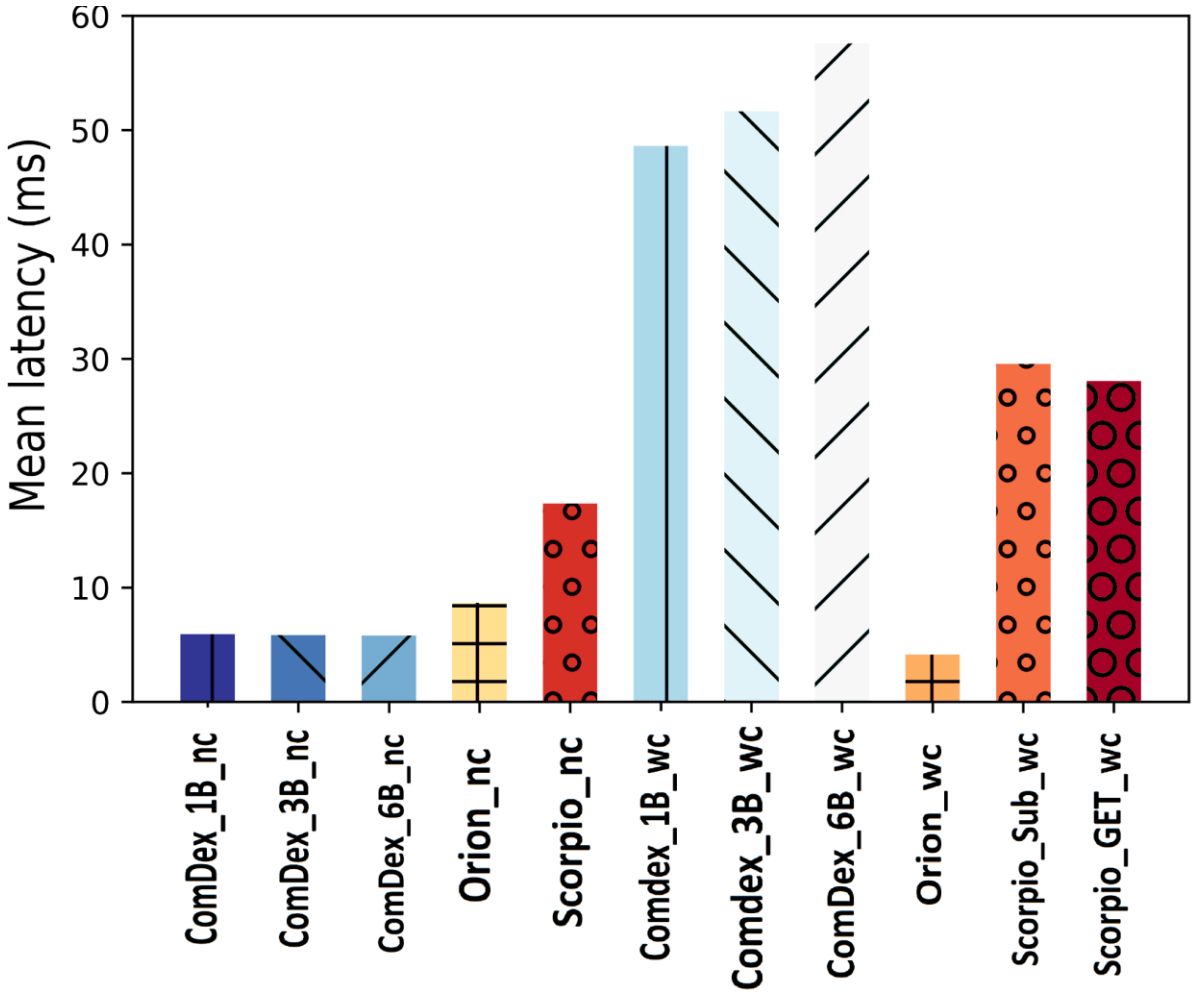
The time that would take for the requested data to arrive would be:

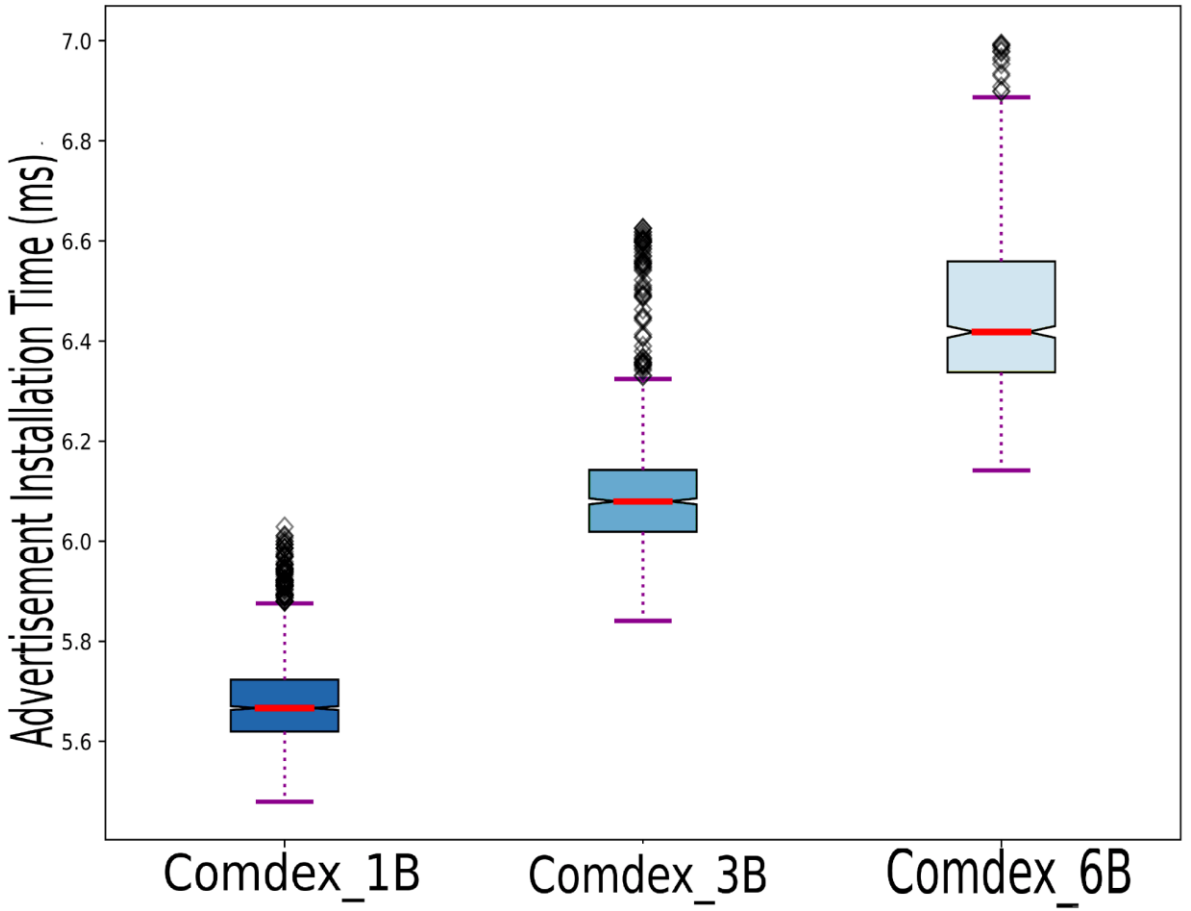
Advertisement_installation_time + discovery_by_the_action_handler + creation_of_new_broker_connections/subscriptions + notification_time_from_broker_to_client

Experiment 2: Experimental Setup



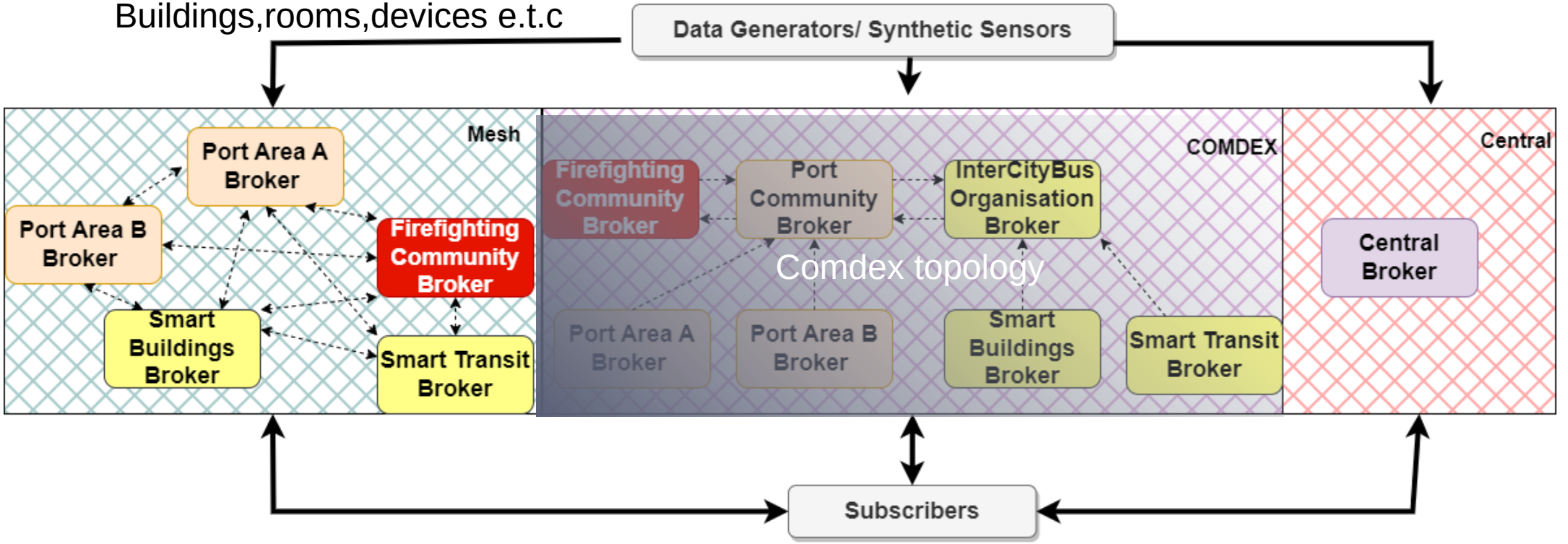
Experiment 2: Experimental Results (B)





Experiment 3: Experimental Setup

Approx. 15k entities:
Buildings,rooms,devices e.t.c



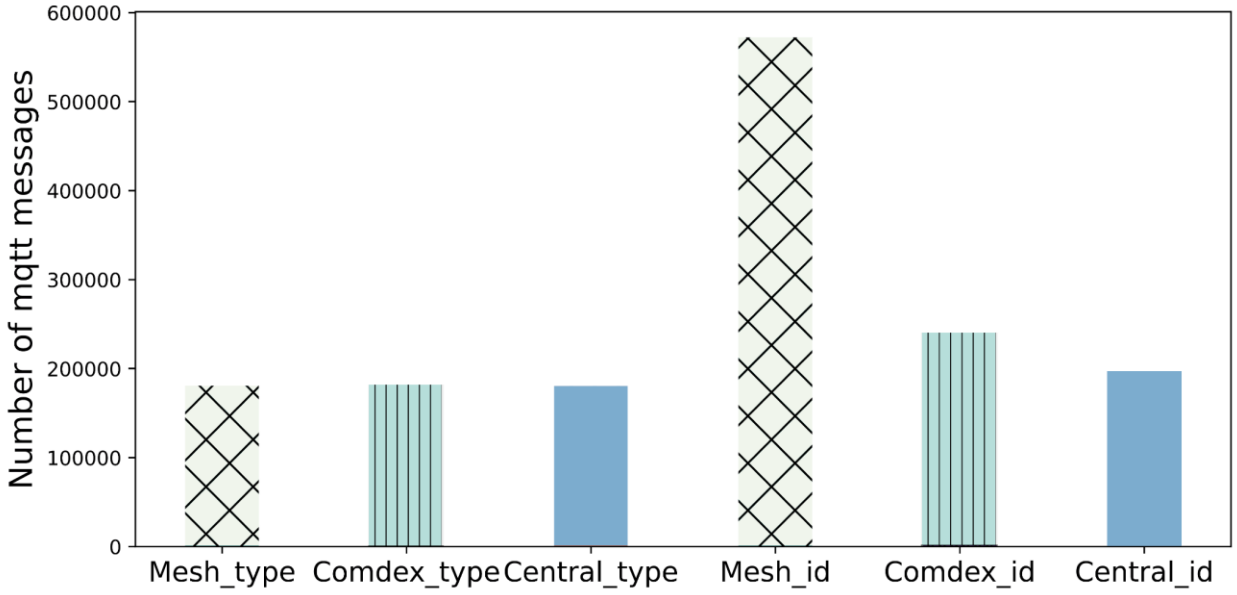
Ads for every type

Ads for every id

Devices and observations using real traces

R. Kumar, M. Swarnkar, G. Singal and N. Kumar, "IoT Network Traffic Classification Using Machine Learning Algorithms: An Experimental Analysis," in IEEE Internet of Things Journal, vol. 9, no. 2, pp. 989-1008, 15 Jan.15, 2022, doi: 10.1109/JIOT.2021.3121517.

Experiment 3: Experimental Results (A)



Number of messages required for the creation of 15000 entities

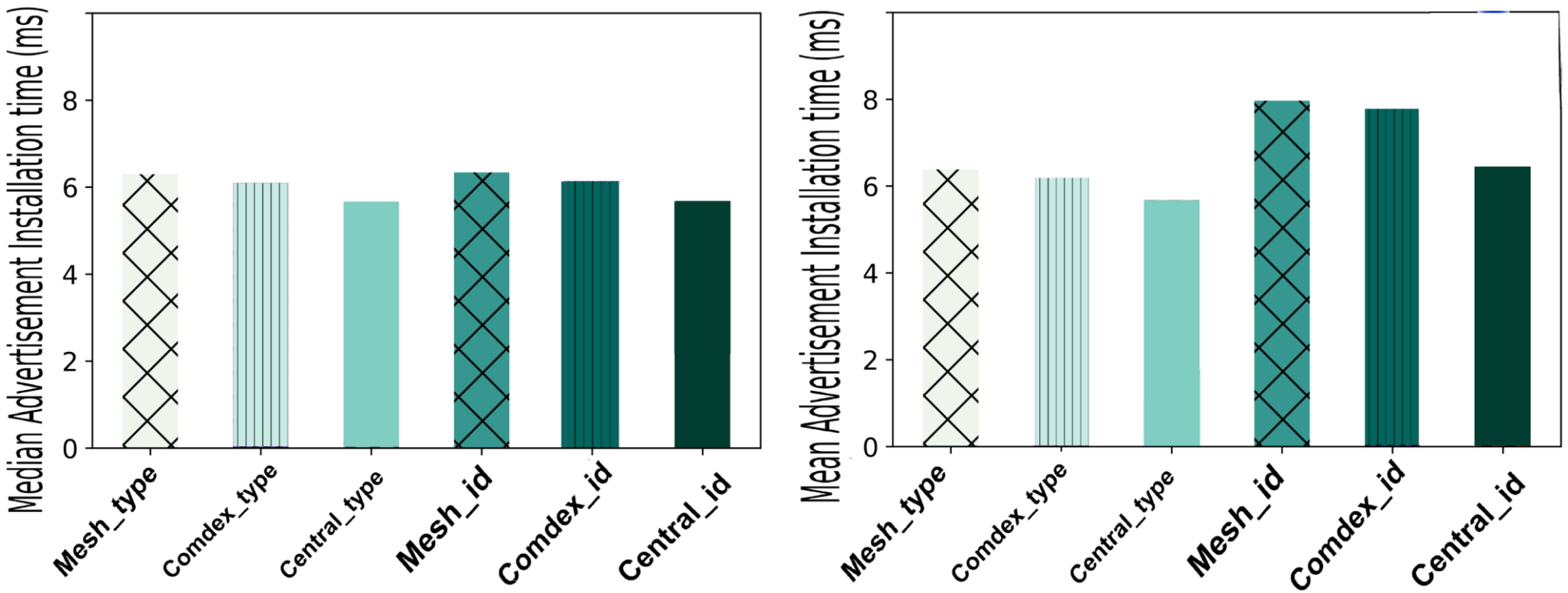
Advertise each different entity type:

```
'provider/' + broker_address + '/' + broker_port + '/'+broker_area+'/' + entity_context + '/'
    + entity_type
```

Advertise each different entity id:

```
'provider/' + broker_address + '/' + broker_port + '/'+broker_area+'/' + entity_context + '/'
    + entity_type + '/' + entity_id
```

Experiment 3: Experimental Results (B)



Advertisement installation times for different topologies and advertisement granularities

Towards the future

- ***What's Next?***
  - Handle QoS guarantees and policies dynamically
  - Handle prototype security.
  - Facilitate the integration of 3rd party platforms

Questions?

Thank you for your time.

Contact:
nikolaos.papadakis@telecom-sudparis.eu
papadakni@ics.forth.gr

Contact

https://samsgblab.github.io/ComDeX/

https://github.com/SAMSGBLab/ComDeX

Try out ComDeX!

SCAN ME