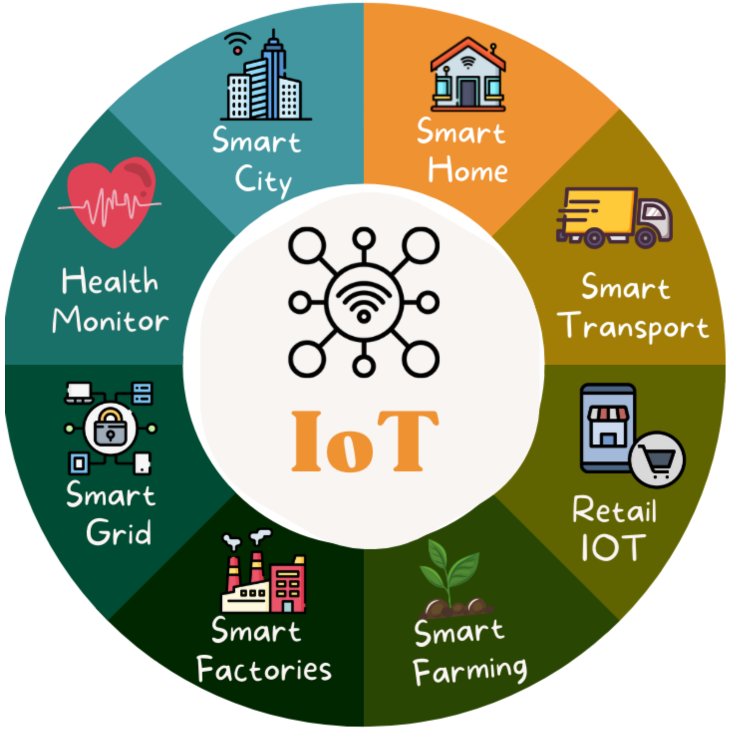
Benchmarking Context-Aware Services for Smart-Transportation IoT Data Exchange

Efstratios Ntallaris, Georgios Bouloukakis, Kostas Magoutis

Introduction

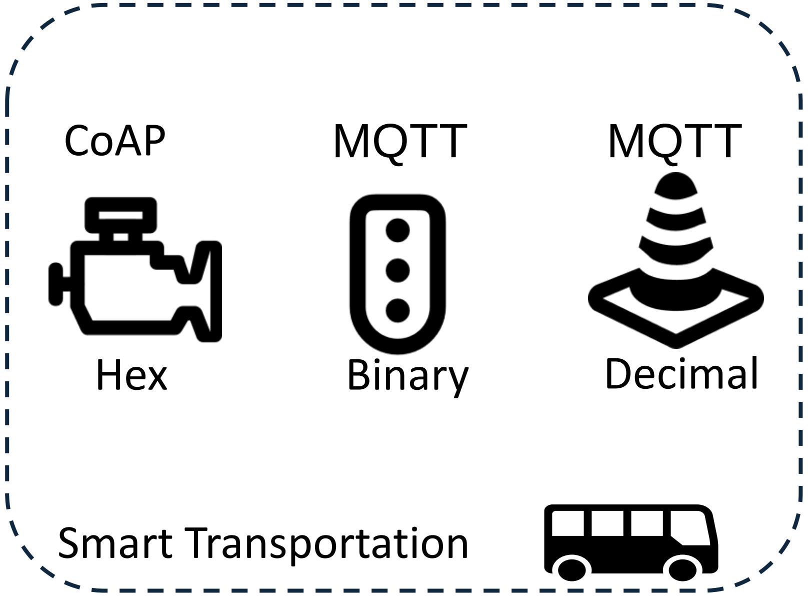
- High demand for IoT Applications
- Smart Transportation is a critical domain



IoT Data Management – Protocols and Sensors

- Diverse devices and sensors
- Different protocols and payloads

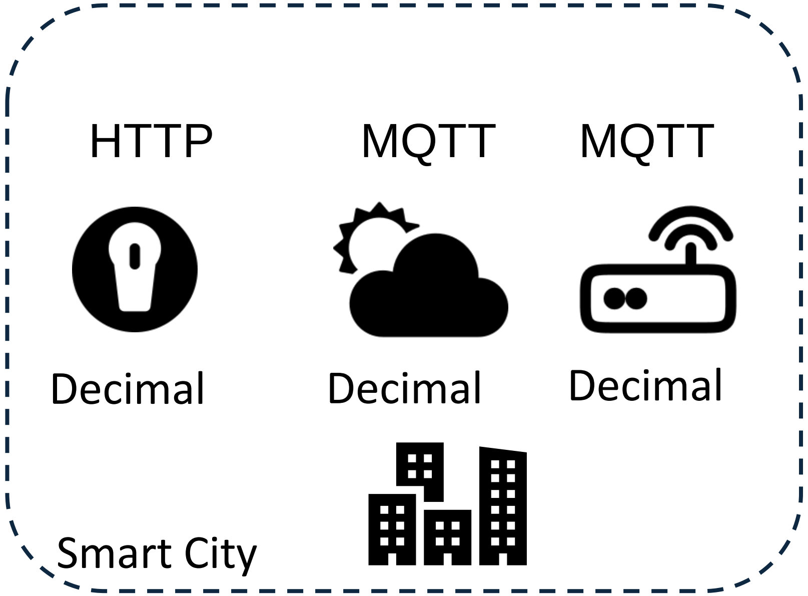
Need of a unifying data format and services



CoAP MQTT MQTT

Hex Binary Decimal

Smart Transportation

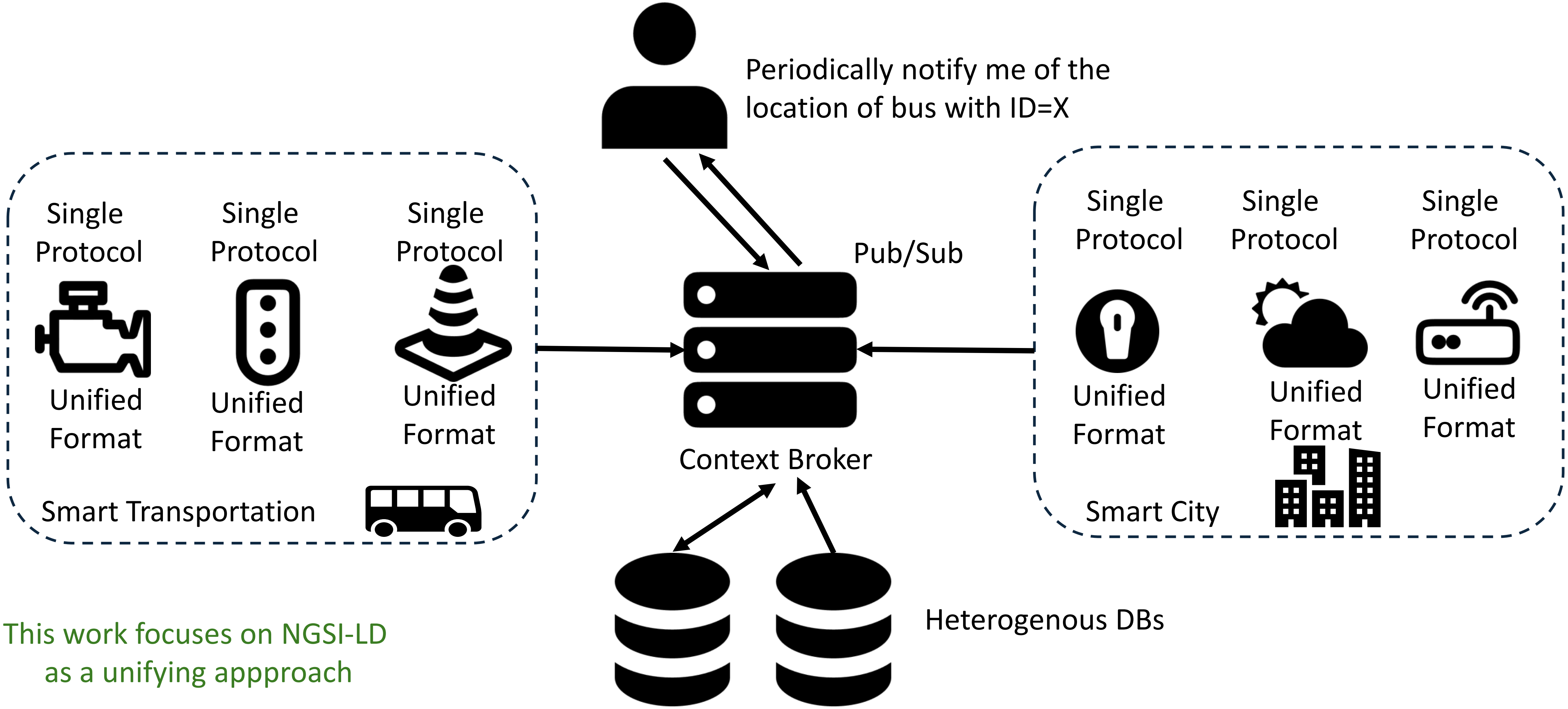


HTTP MQTT MQTT

Decimal Decimal Decimal

Smart City

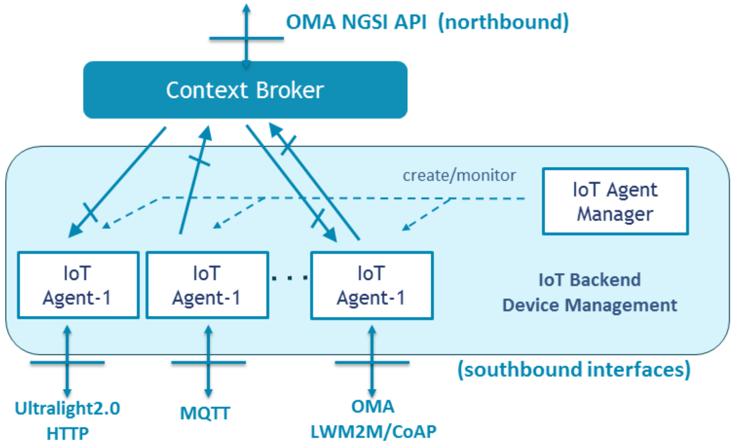
IoT Data Management – A unifying approach

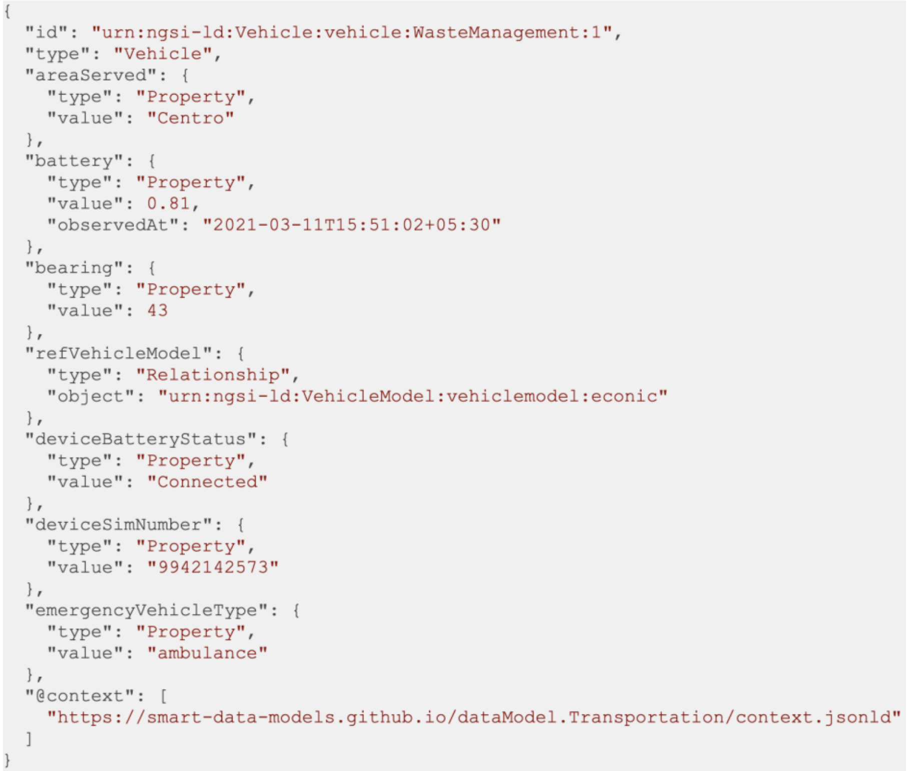


This work focuses on NGSI-LD as a unifying appproach

Next Generation Service Interface with Linked Data (NGSI-LD)

- NGSI-LD Protocol
  - Standard for information representation and exchange in the IoT data space
  - Faciliates semantic interoperability
  - Enables seamless data sharing across systems
  - Standard for context aware applications
  - Publicly available smart data models



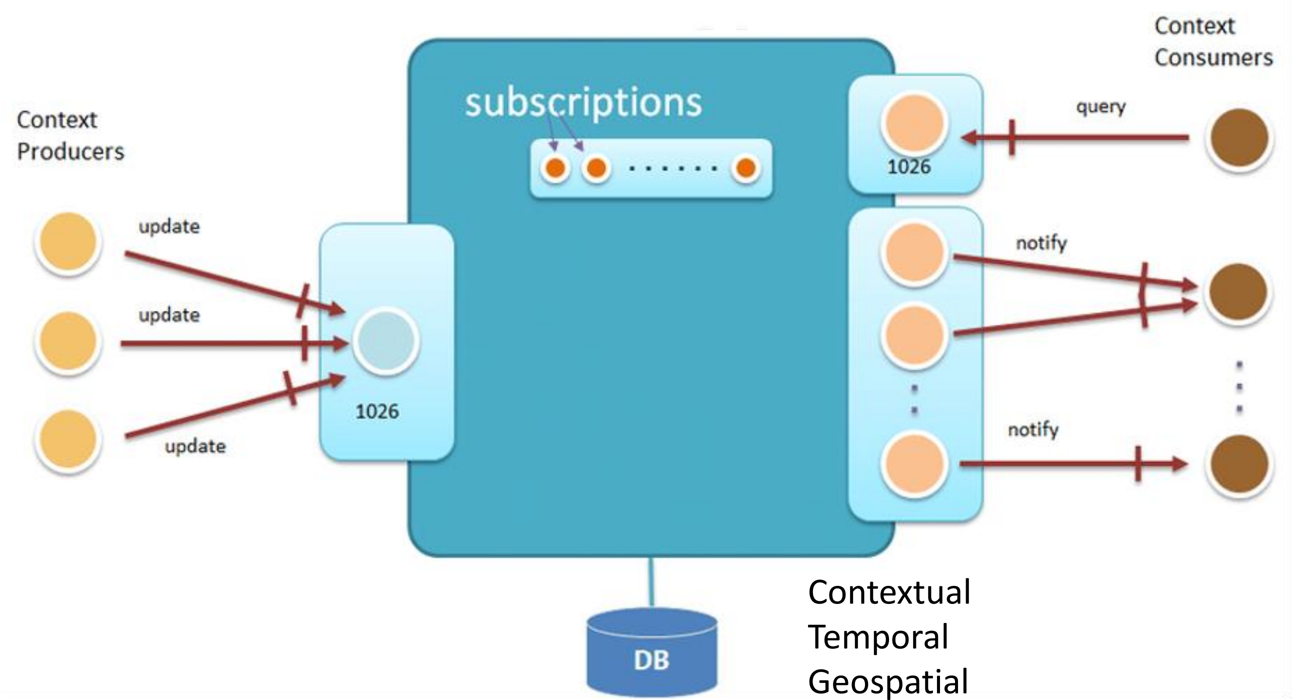


```
{
  "id": "urn:ngsi-ld:Vehicle:vehicle:WasteManagement:1",
  "type": "Vehicle",
  "areaServed": {
    "type": "Property",
    "value": "Centro"
  },
  "battery": {
    "type": "Property",
    "value": 0.81,
    "observedAt": "2021-03-11T15:51:02+05:30"
  },
  "bearing": {
    "type": "Property",
    "value": 43
  },
  "refVehicleModel": {
    "type": "Relationship",
    "object": "urn:ngsi-ld:VehicleModel:vehiclemodel:econic"
  },
  "deviceBatteryStatus": {
    "type": "Property",
    "value": "Connected"
  },
  "deviceSimNumber": {
    "type": "Property",
    "value": "9942142573"
  },
  "emergencyVehicleType": {
    "type": "Property",
    "value": "ambulance"
  },
  "@context": [
    "https://smart-data-models.github.io/dataModel.Transportation/context.jsonld"
  ]
}
```

https://www.fiware.org/smart-data-models/

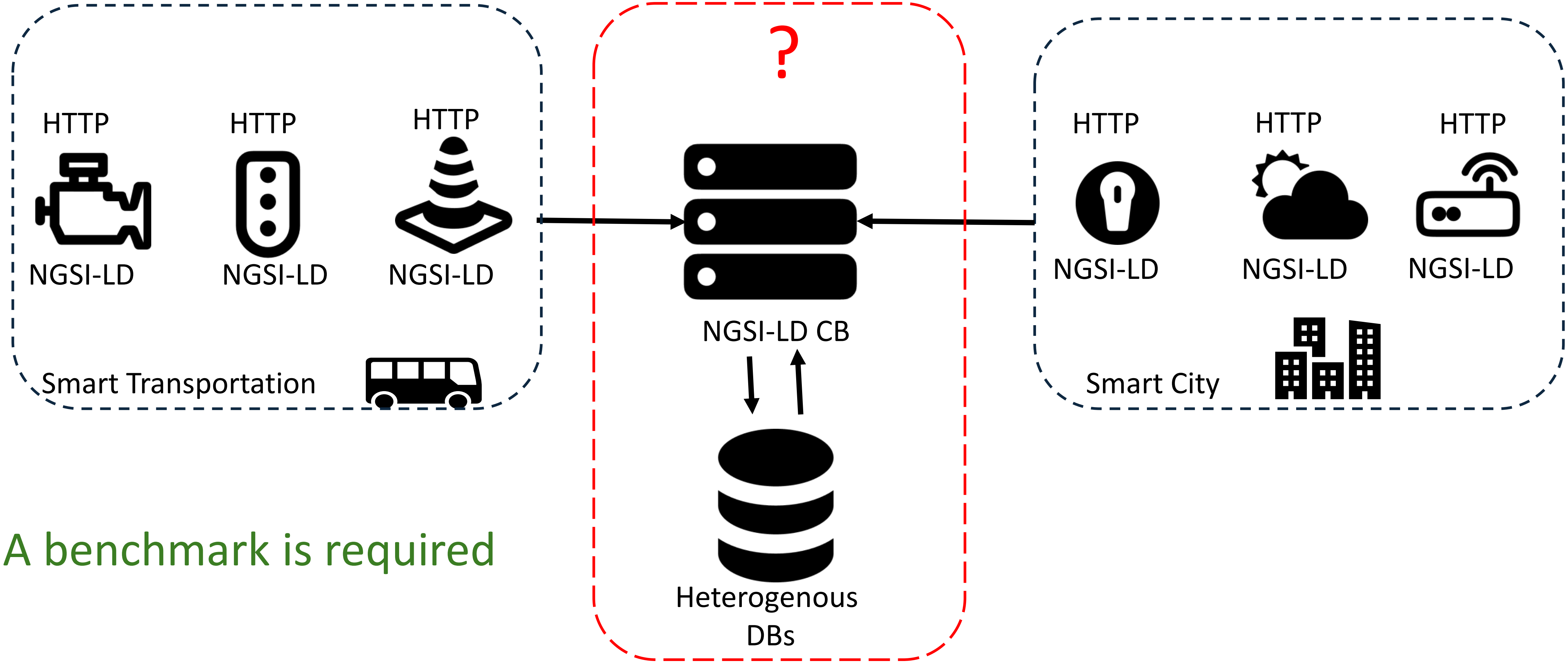
NGSI-LD Context Brokers (CBs)

- Serve as intermediaries between IoT devices and applications
- Collect, manage and serve contextual data
- HTTP Protocol
- **Orion-LD, Stellio, Scorpio** FIWARE Open APIs for Open Minds



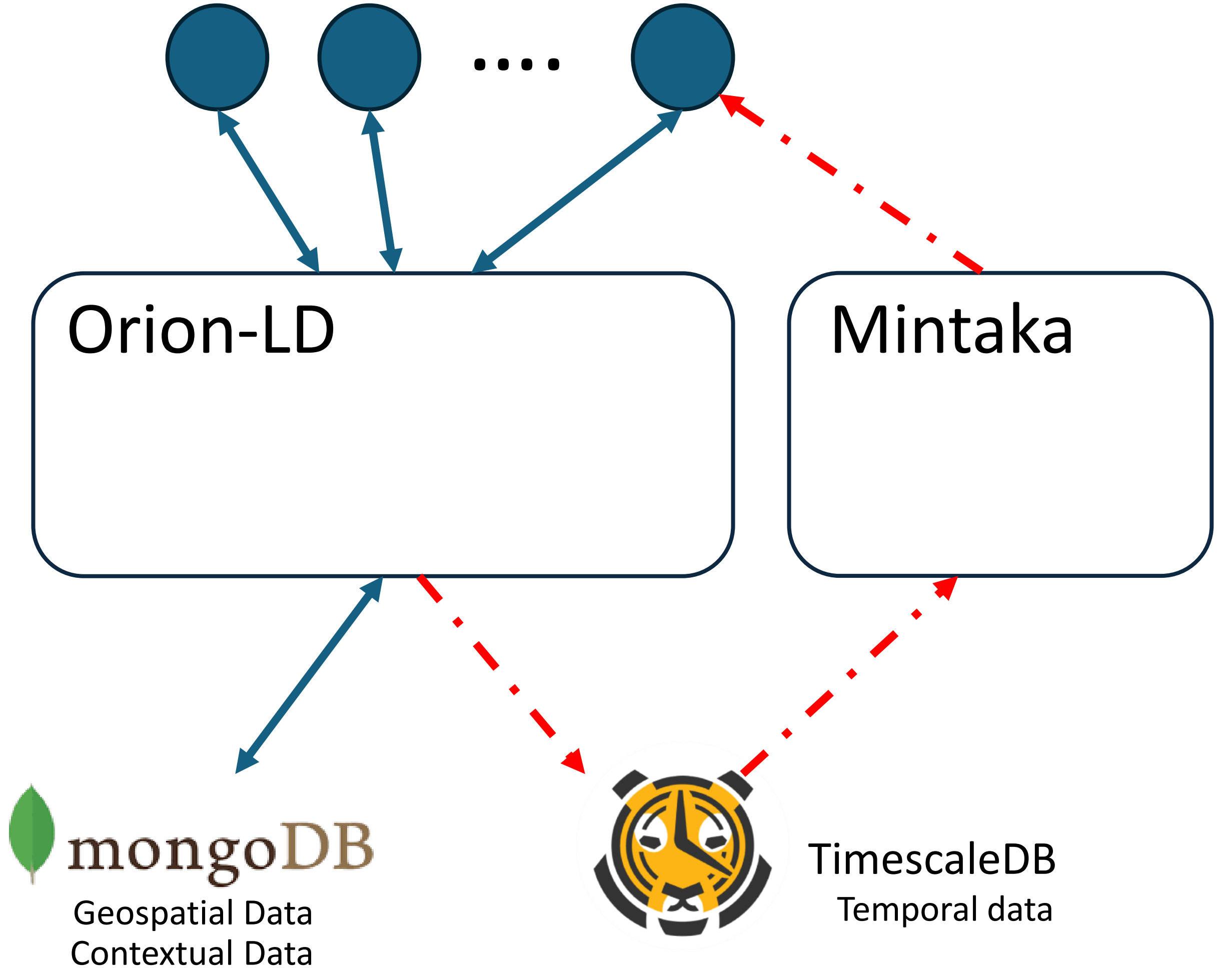
Contextual
Temporal
Geospatial

Which Context-Broker to select?



A benchmark is required

CBs underlying Architectures: Orion-LD



Applications/Users

Orion-LD

Mintaka

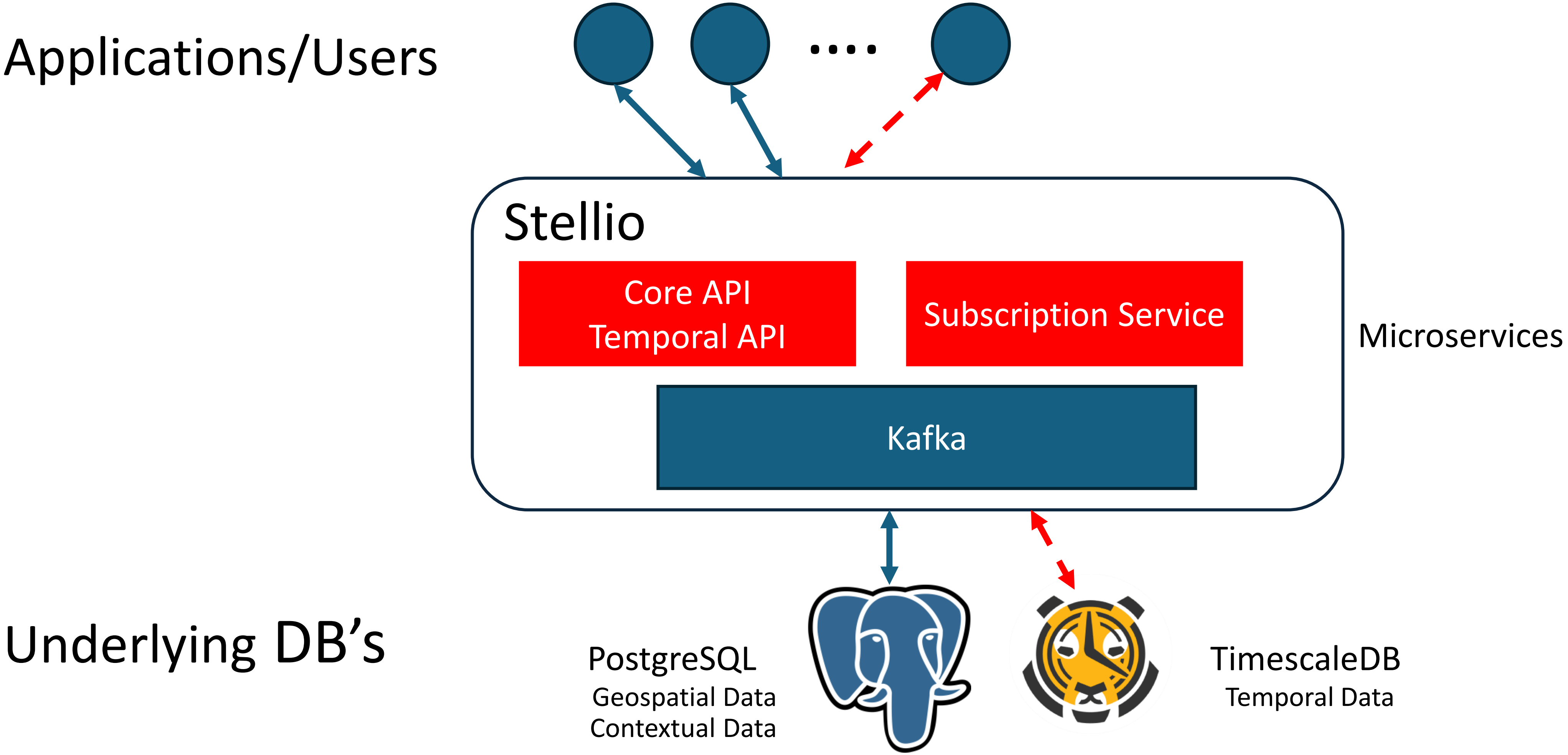
Monolithic

Underlying DB’s

Geospatial Data
Contextual Data

TimescaleDB
Temporal data

CB’s underlying Architectures: Stellio



Applications/Users

....

Stellio

Core API
Temporal API

Subscription Service

Microservices

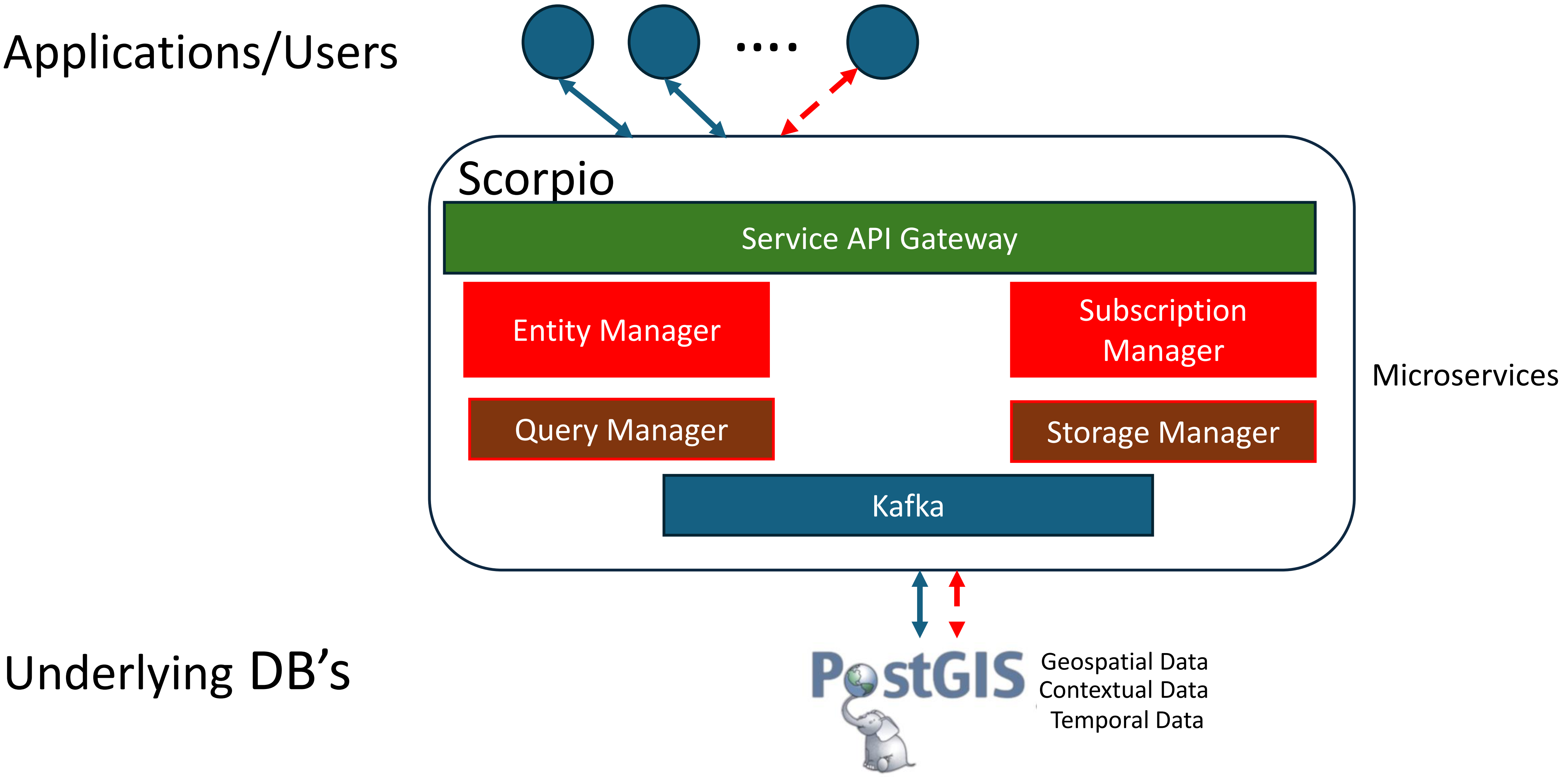
Kafka

Underlying DB’s

PostgreSQL
Geospatial Data
Contextual Data

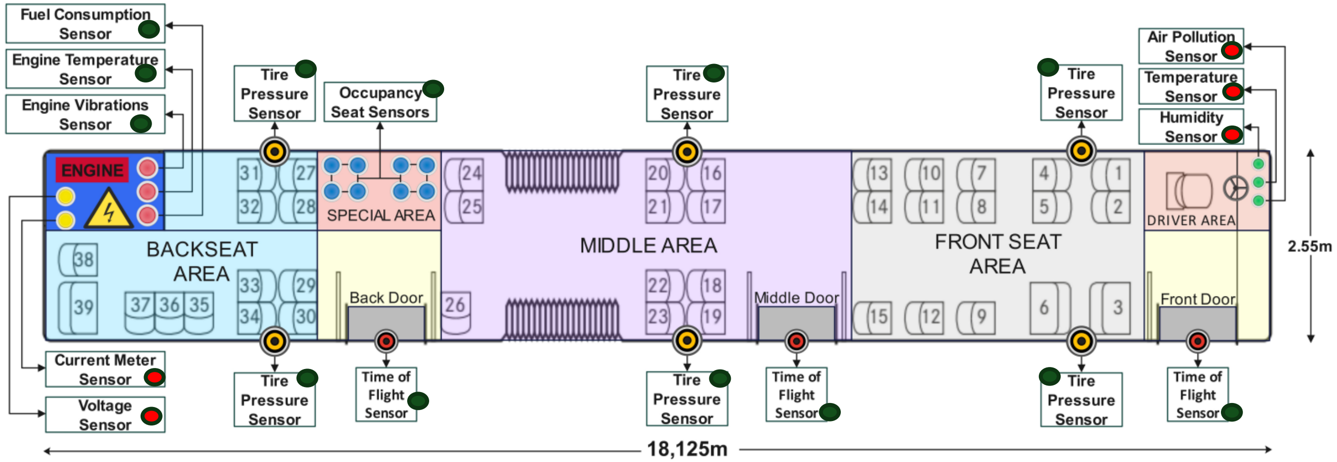
TimescaleDB
Temporal Data

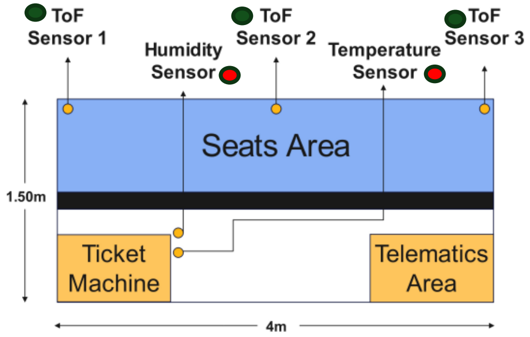
CB’s underlying Architectures: Scorpio



The SCB Benchmark: Overview

- **Objectives**
  - Evaluate performance of CB’s in Smart transportation environments
  - Measure data ingestion and querying capabilities
- **Uses NGSI-LD for data modelling**
- **Rich Models (Buses, Bus Stations, IoT devices)**
- **Data Generation**
  - **Real-Time Data (Bus Positions)**: Collected live from the Bus Service in Ioannina, Greece
  - **Existing Datasets**: Includes traffic data (Environmental etc) from Aarhus, Denmark 🔴
  - **Synthetic Data**: Generated to emulate various sensor reports 🟢





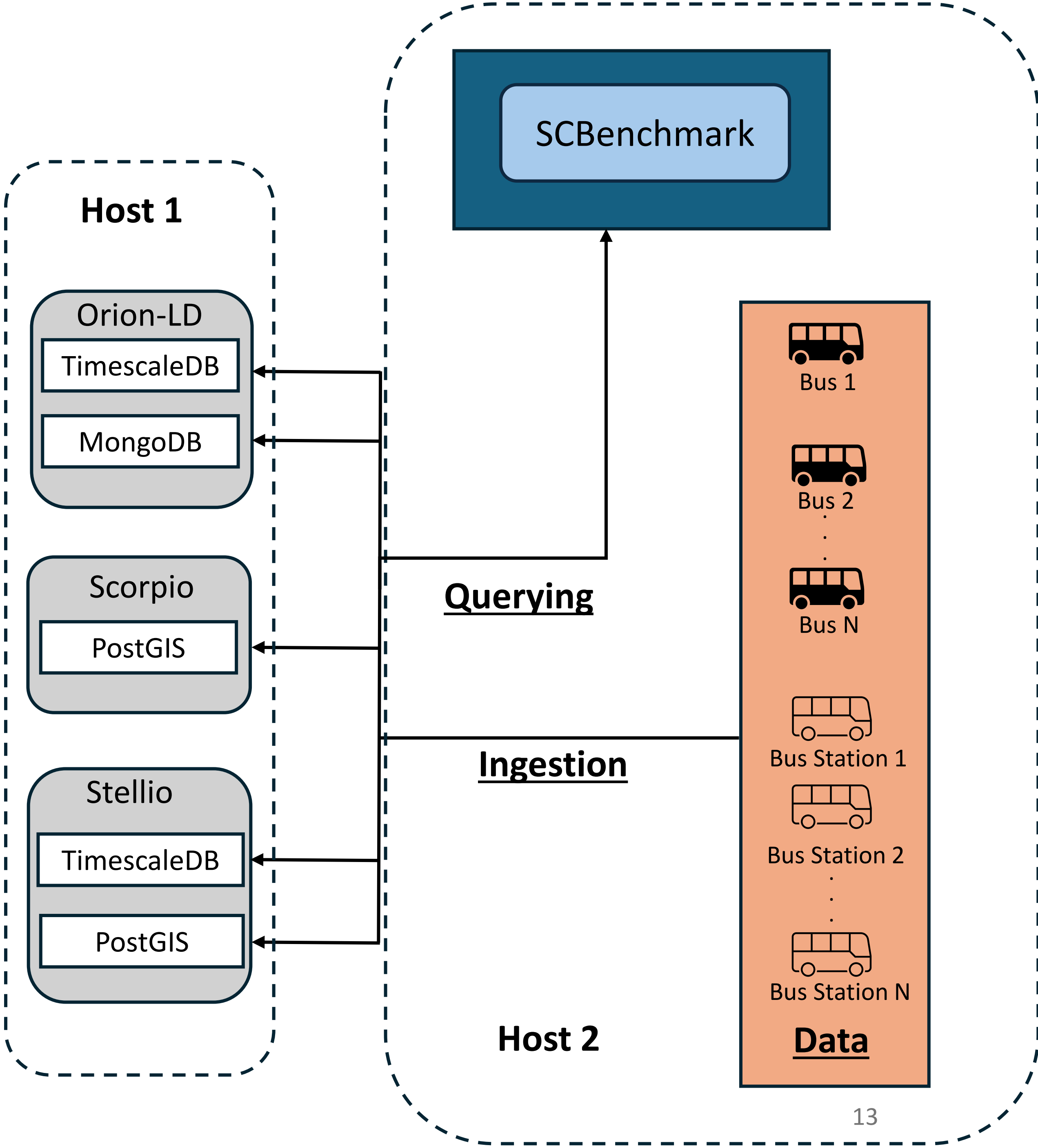
Georgios Bouloukakis et al “Enabling IoT-enhanced Transportation Systems using the NGSI Protocol”, IoT '22

Query Types

- **General Queries (retrieve current state):**
  - Bus location retrieval (Q1)
  - Temperature threshold analysis (Q2)
  - Fuel Efficiency Ranking (Q3)
- **Temporal Queries (retrieve past state):**
  - Average air pollution assessment (Q4)
  - Daily bus occupancy analysis (Q5)
- **Geospatial Queries (location based):**
  - Proximity-based bus search (Q6)
  - Bus stop proximity search (Q7)

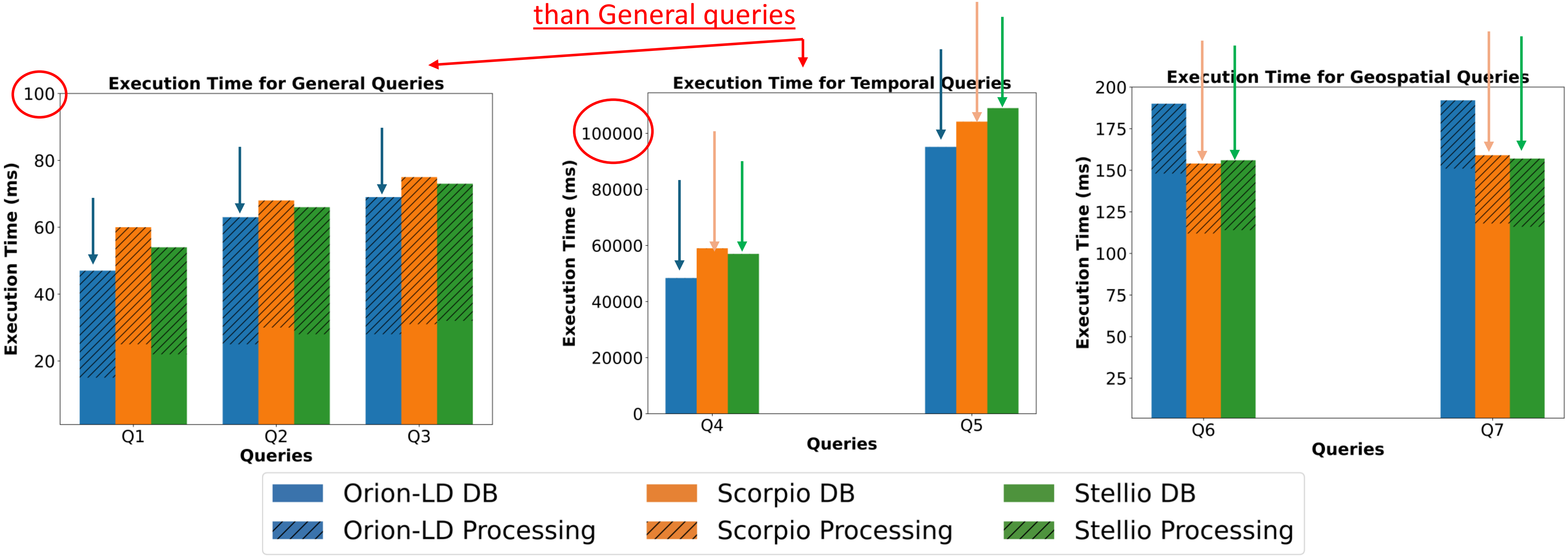
Benchmark Setup

- Two Servers
- Hardware Specifications:
  - ➢ Intel Xeon Bronze 3206R 1.9 GHz, 32GB DDR4.
  - ➢ Dell Micron 480GB SSD, Toshiba 2TB HDD.
- Cache size set to 2 GB in all broker DB's
- **Data Generation for bus fleet and stations:** > 204 GBs



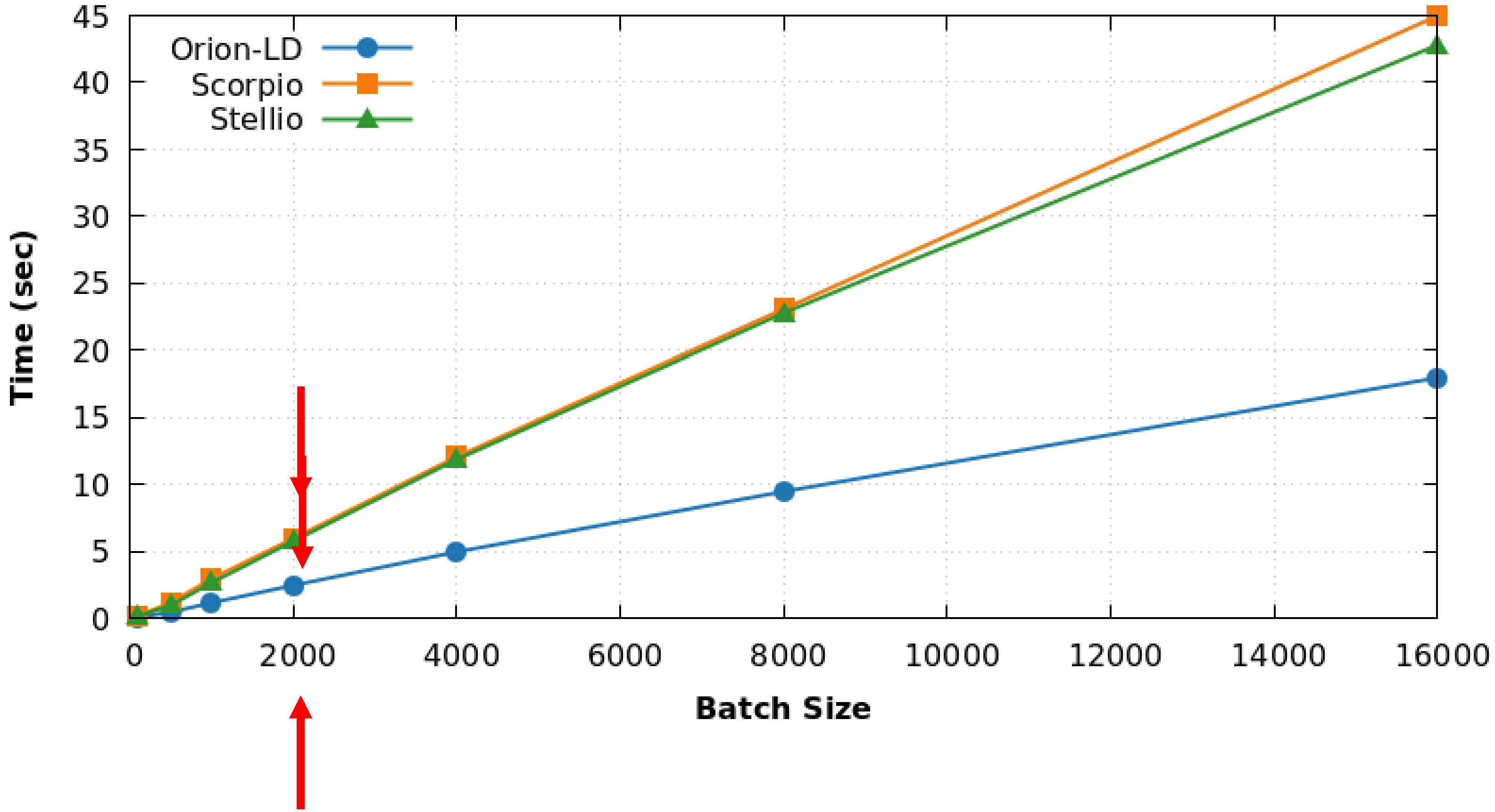
Results: Query Performance

Temporal Queries take 10x to 100x longer than General queries



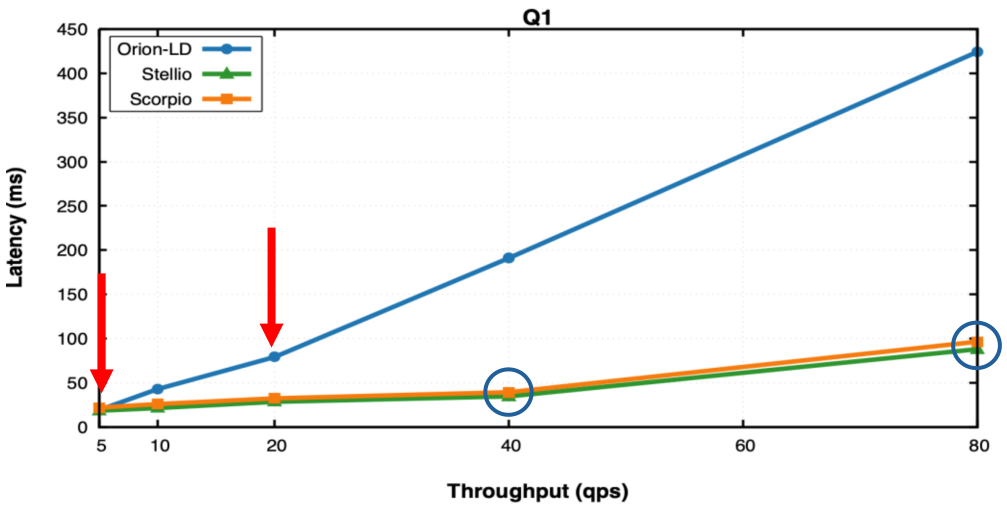
- General Queries (Q1, Q3)
  - All brokers perform comparably
  - Orion-LD slightly better
- Temporal Queries (Q4, Q5)
  - Orion-LD slightly better
  - Orion-LD and Stellio both use TimescaleDB
  - PostGIS is comparable
- Geospatial Queries (Q6,Q7)
  - Stellio and Scorpio performed better
  - Better support for spatial data (PostGIS) (PostgreSQL)

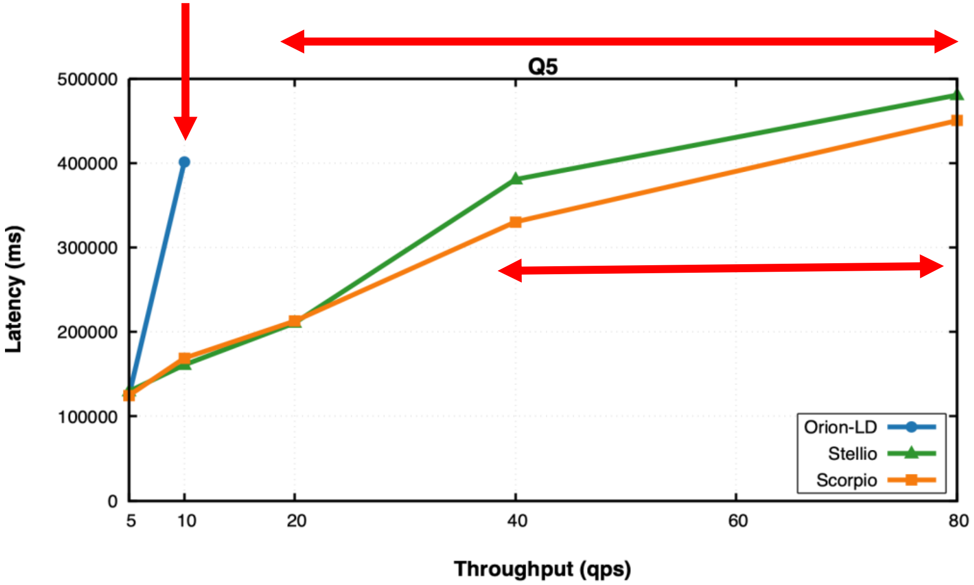
Results: Ingestion



- Orion-LD better ingestion performance
- Scorpio and Stellio similar performance

Results: Scalability





- For Q1:
  - ➢ Orion-LD performed well at low throughput
  - ➢ Orion-LD, sharp increase after 20 qps
  - ➢ Stellio and Scorpio handle higher query rates effectively.

- For Q5:
  - ➢ Orion-LD performed well at low throughput
  - ➢ Orion-LD, sharp increase at 10 qps
  - ➢ Timeout after 10 qps
  - ➢ Stellio, Scorpio scale much better

Conclusion and Future Work

- SCBenchmark: a benchmark for evaluating IoT context-aware services using real testbed in the city of Ioannina, Greece.

Context Broker	Ingestion	General Queries	Geospatial Queries	Temporal Queries	Scalability
Orion-LD	✓	✓	✓	✗	✗
Scorpio	✗	✓	✓	✗	✓
Stellio	✗	✓	✓	✗	✓

- Future work:
  - Explore strategies for optimizing temporal query processing
  - Investigate application-specific caching mechanisms

Q&A

Thank You!

Any Questions?

https://github.com/satrai-lab/scbenchmark