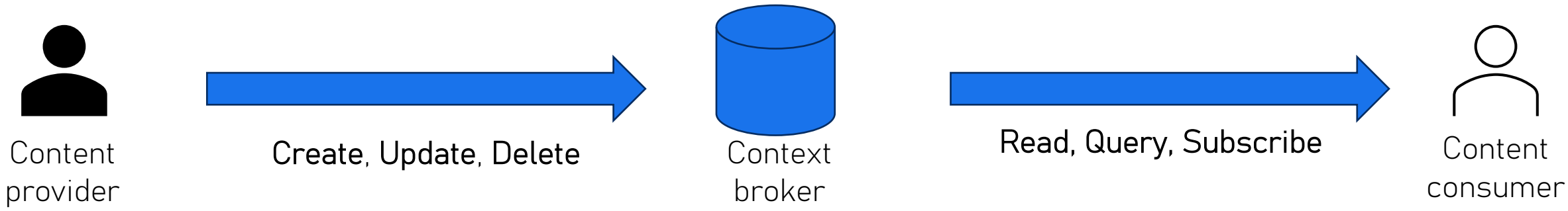
Secure and Efficient Data Spaces over Named Data Networking

Yannis Thomas, Nikos Fotiou, Iakovos Pittaras,
George Xylomenos

Data Spaces

- A collection of technologies that enables interoperable and secure data sharing
- Entities:
  - Context broker
  - Content provider
  - Content consumer



The ETSI NGSI-LD API – Item representation

- Data items are represented as JSON-LD objects, and they have:
  - An identifier
  - A type
  - Attributes and corresponding values
- The attributes that may exist for a type are defined in a context


```
{
    "id": "urn:ngsi-ld:Car:001",
    "type": "Car",
    "brand": {
        "type": "Property",
        "value": "BWM"
    },
    "dateVehicleFirstRegistered": {
        "type": "Property",
        "value": "2012"
    },
    "emissionsCO2": {
        "type": "Property",
        "value": "22"
    },
    "@context": [
        "https://example.com/data-models.context-ngsild.jsonld"
    ]
}
```

The ETSI NGSI-LD API – Data access

- Retrieve all/some attributes based on item id
- Retrieve all/some attributes of all items of a specific type
- Temporal queries
- Subscription to changes in attributes

Can the existing approach be improved?

- Context broker is centralized ➡️ single point of failure, scalability issues
- Context broker can manipulate data ➡️ broker should be trusted
- Application layer API, end-to-end encrypted ➡️ Subscriptions for the same event are handled using multiple unicast connections, no caching support

Our contributions:

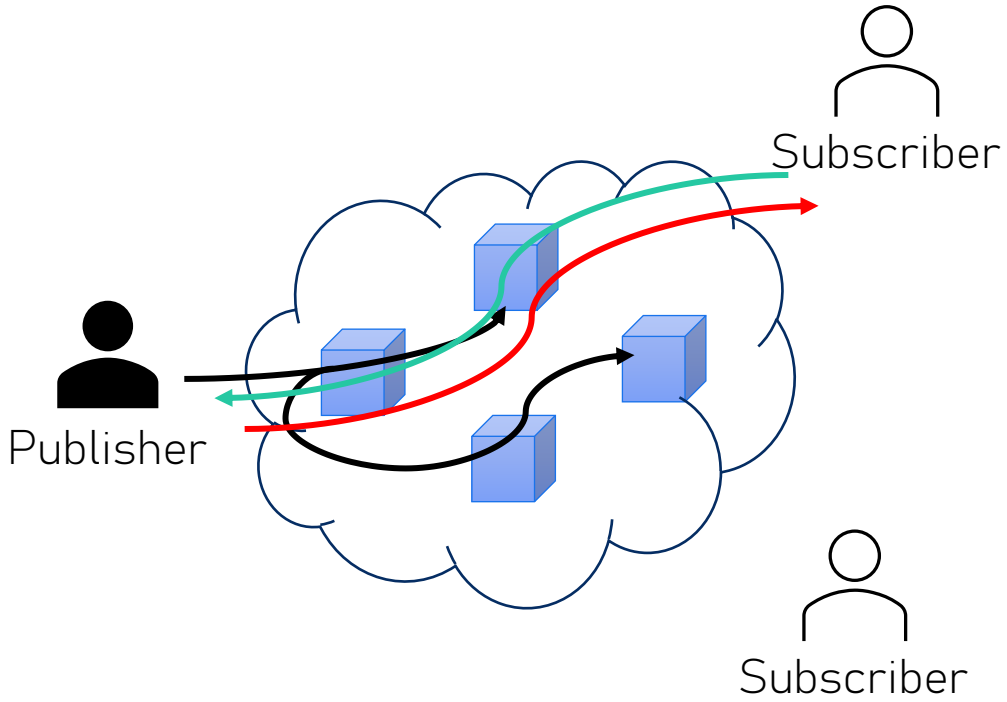
- Use an Information-Centric Networking architecture as an in-network, distributed “broker”
- Use data-centric security solution

Named-Data Networking

An Information-Centric Networking Architecture

Overview

- Publishers advertise content identifiers
- Subscribers send an “interest” in a content identifier
- The publisher forwards the content



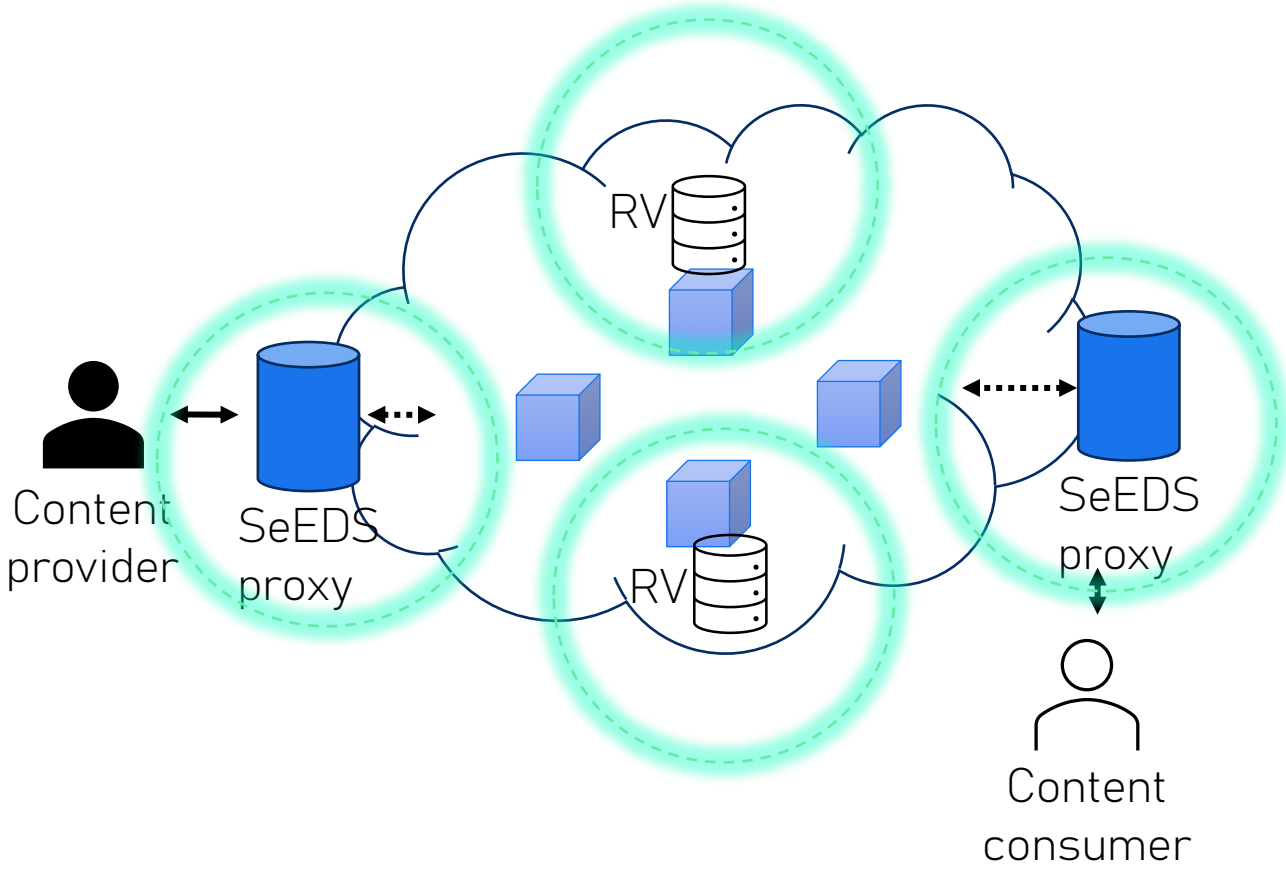
Advantages

- Forwarded content can be cached
- Interests for the same content item are aggregated ➡️ content is delivered using multicast
- The same content identifier can be advertised from multiple locations ➡️ multihoming is inherently supported

Our architecture for *Secure and Efficient Data Space* – SeEDS

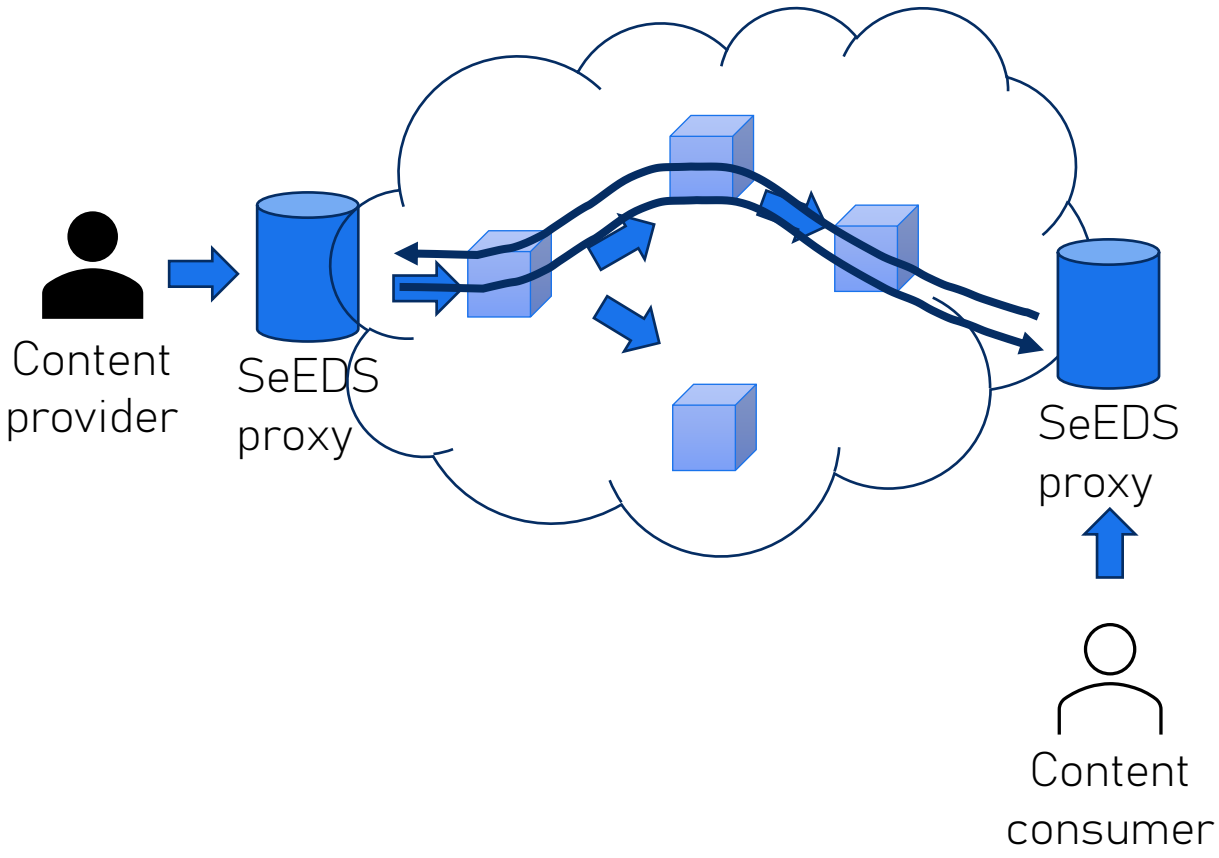
Overview

- A proxy is responsible for translating ETSI NGSI-LD API messages to the corresponding NDN messages
- In network nodes act as “registries” for “types”
  - They include “pointers” to the actual storage location of a data item



Get item by Id

1. Item Creation
   A. Content provider ➞ HTTP POST item
   B. SeEDS proxy ➞ Announce item Id
2. Get item
   A. Content consumer ➞ HTTP GET item
   B. SeEDS proxy ➞ Interest item Id
   C. SeEDS proxy ➞ Data

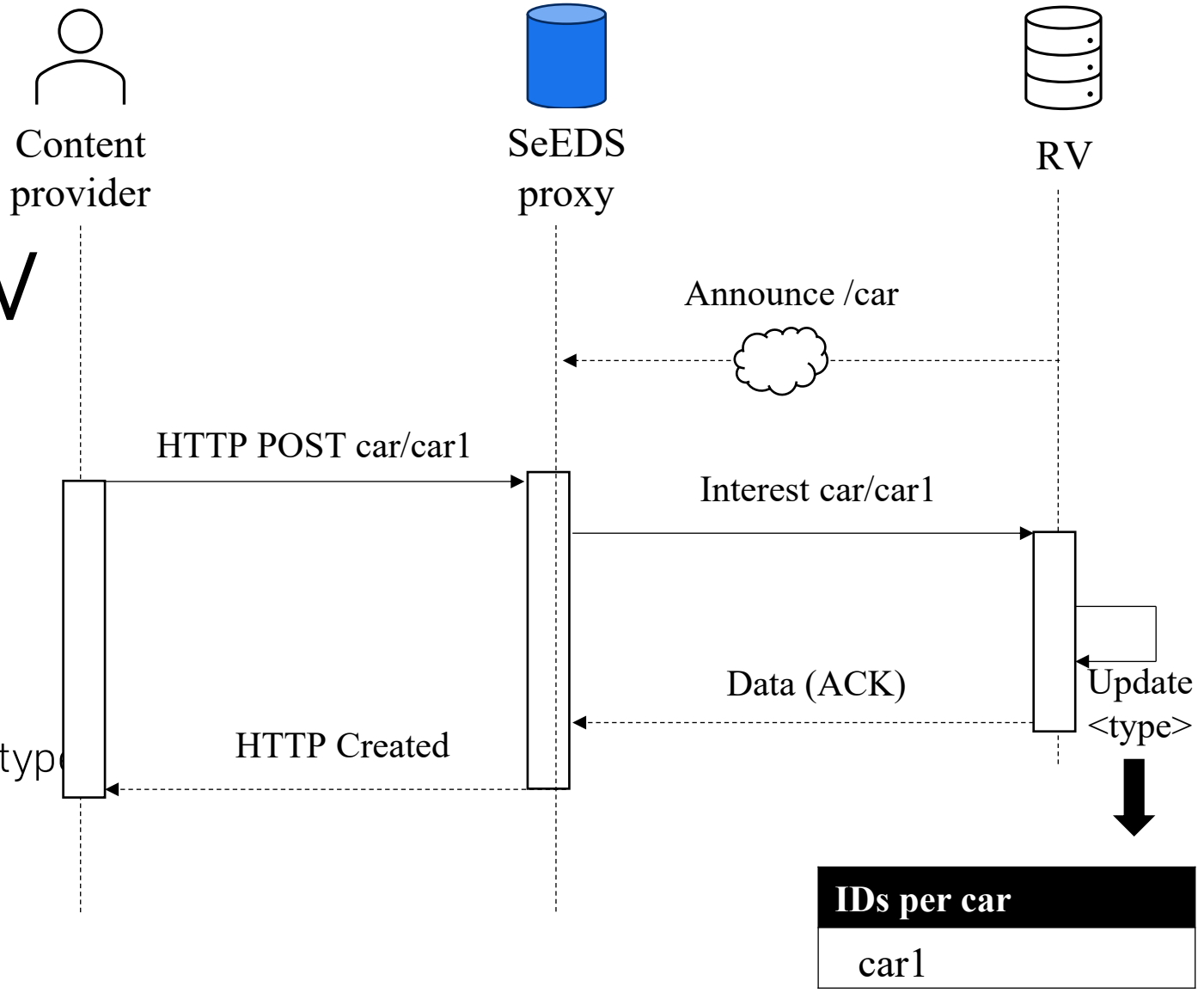


Discussion

- Straightforward use of NDN
- Many performance gains due to NDN's properties:
  - Support for caching
  - Support for multihoming ➞ Resilience to failures
  - Request aggregation ➞ Burst requests are served using multicast, even more gains for subscriptions (future work)
- But many-to-one communication required for requests by type, are not supported
- Our approach: a Rendezvous (RV) point

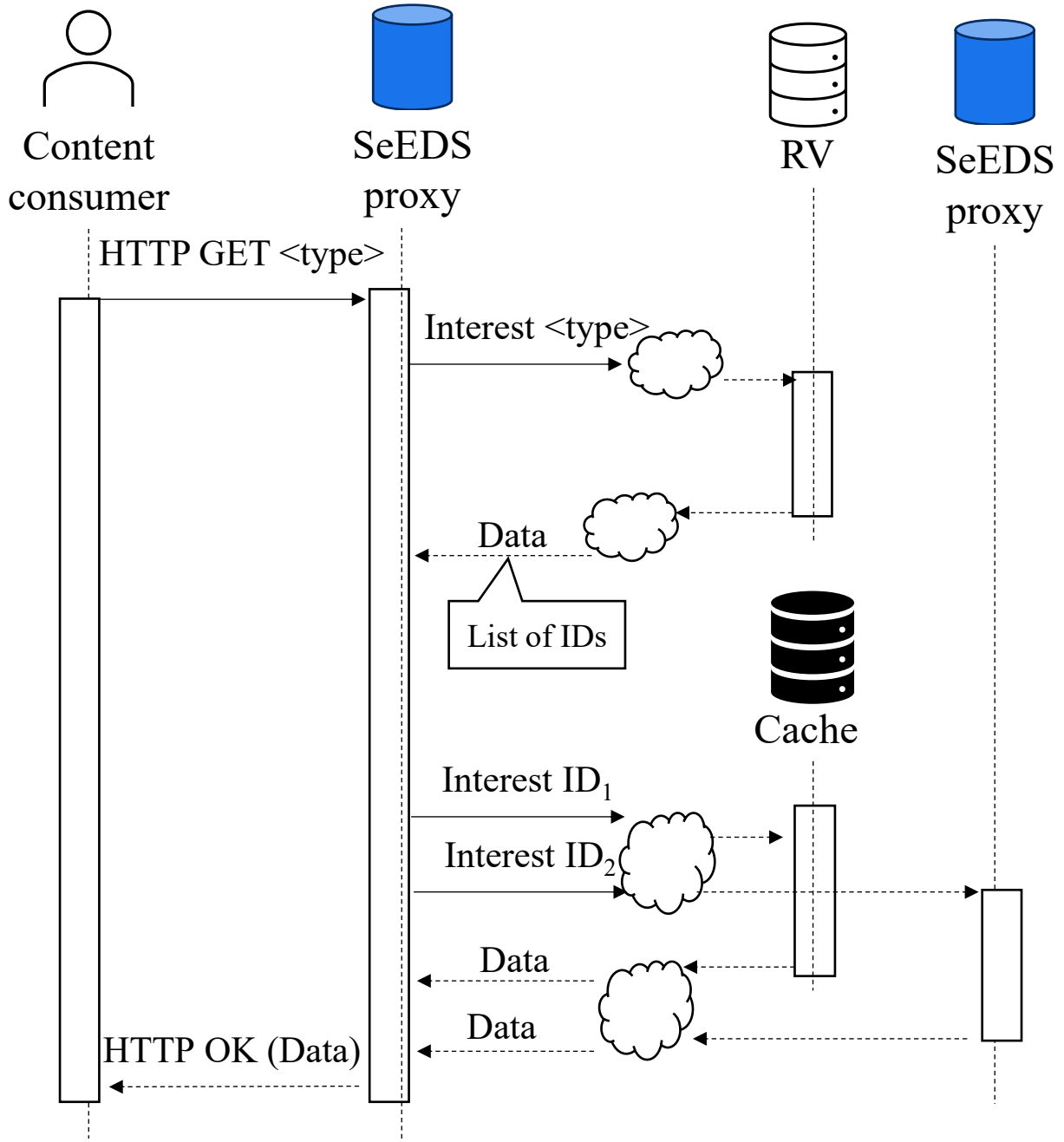
Item creation with RV

1. RV maintains items per type
2. RV announces a "type"
3. Content provider ➞ HTTP POST item
4. SeEDS proxy ➞ Interest for the item typ
5. RV updates its internal db
6. RV ➞ Data
7. SeEDS proxy ➞ HTTP Created



Get items by type

1. Content consumer ➞ HTTP GET type
2. SeEDS proxy ➞ Interest for type
3. RV ➞ Lists of IDs for the type
4. SeEDS proxy ➞ Interest for each type
5. Items arrive
6. SeEDS proxy ➞ HTTP OK with data



Discussion

- Items can be retrieved from multiple sources ➞ this is not supported by the existing approach

Security mechanisms

- Data integrity protection that supports selective disclosure[1]
  - JSON objects are represented as a list of “disclosures”
  - The list is signed using either BBS+ (selective disclosure with full unlikability) or ECDSA + salt (selective disclosure)
  - Any entity can hide a message from the list and still the recipient can verify the integrity of the revealed messages
- Proxy to RV authentication using W3C Decentralized Identifiers
  - Each content owner owns a Decentralized Identifier used a content name prefix
  - An RV can verify if a proxy is authorized to advertise a prefix

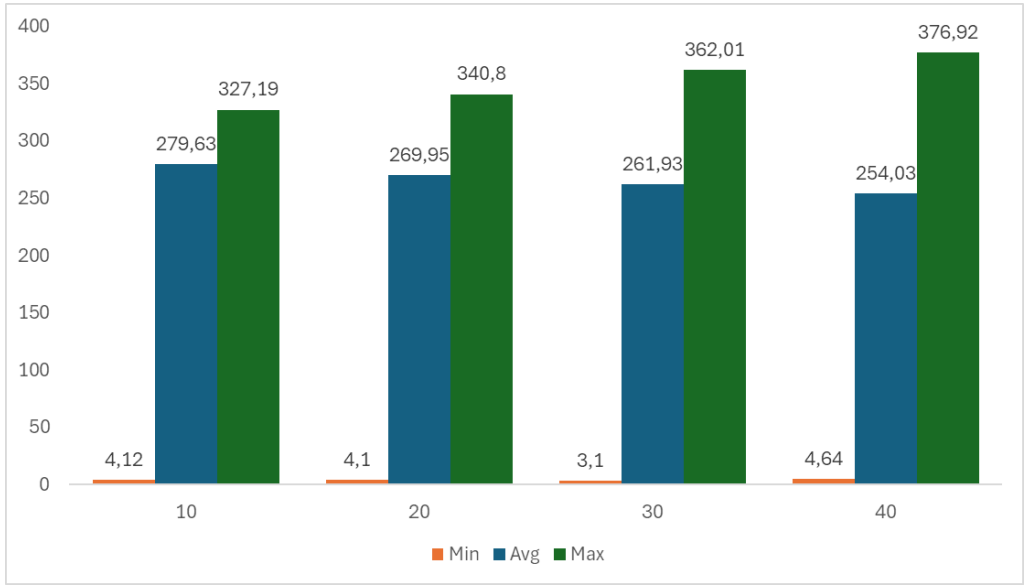
[1] N Fotiou, G Xylomenos, Y Thomas, “Data integrity protection for data spaces, “Proceedings of the 17th European Workshop on Systems Security, 2024

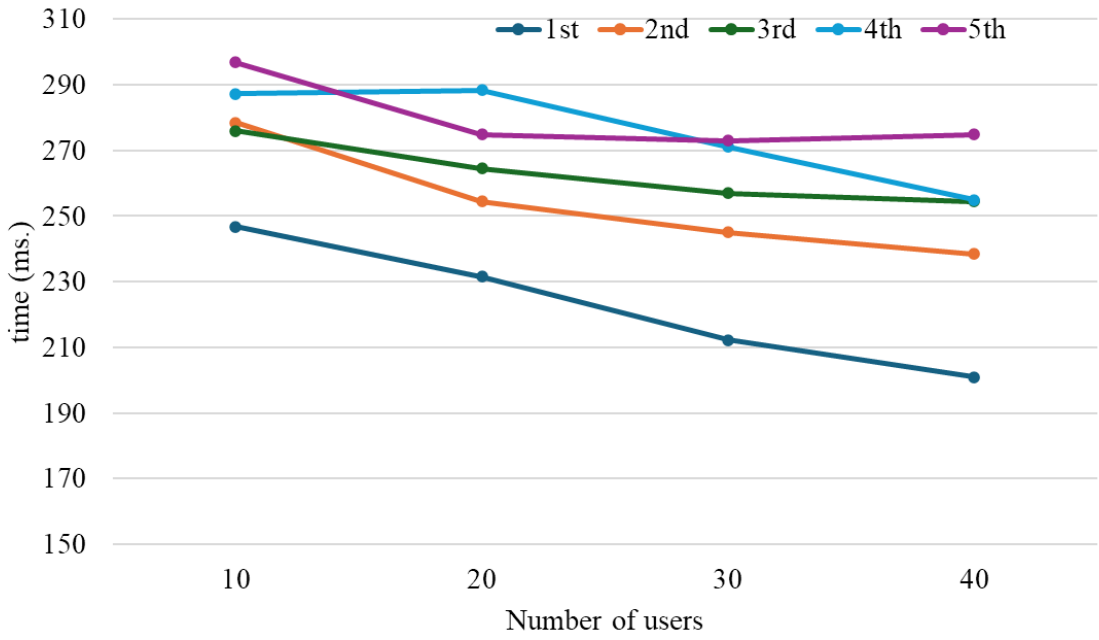
Performance results

Implementation & Setup

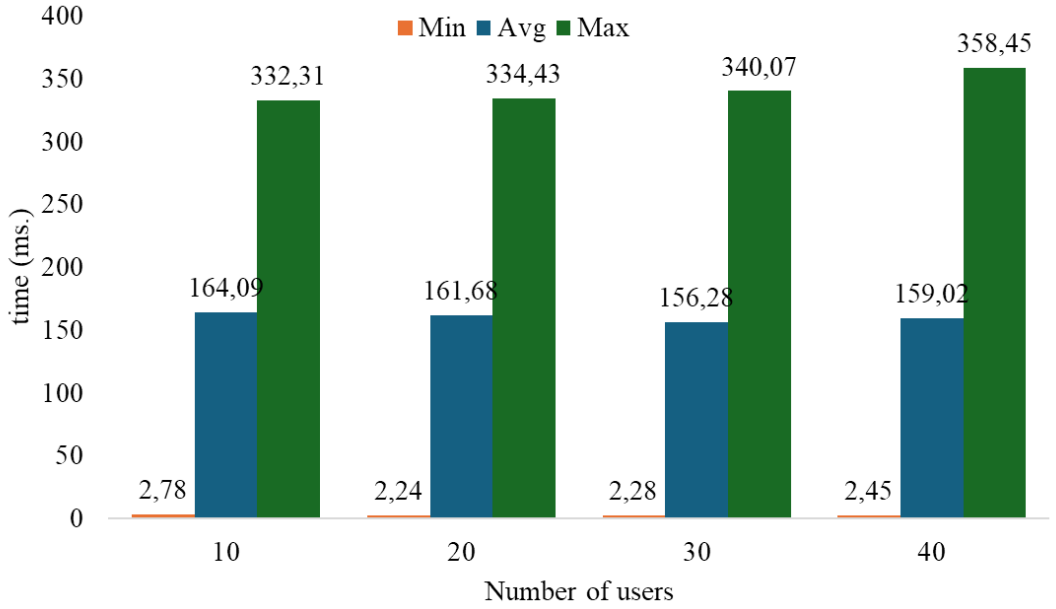
- Baseline implementation running in NDN testbed and mini-NDN
  - SeEDS Proxy: HTTP to/from NDN
  - Regular NDN network behind Proxy
  - SQL Database for storage, to be migrated to FIWARE ORION broker
- Single SeEDS Proxy connected to NDN network
  - NDN network delay: 300 ms
  - O=100 objects, Zipf popularity
  - O*0.1 to O*0.4 consumers make requests for 2 minutes

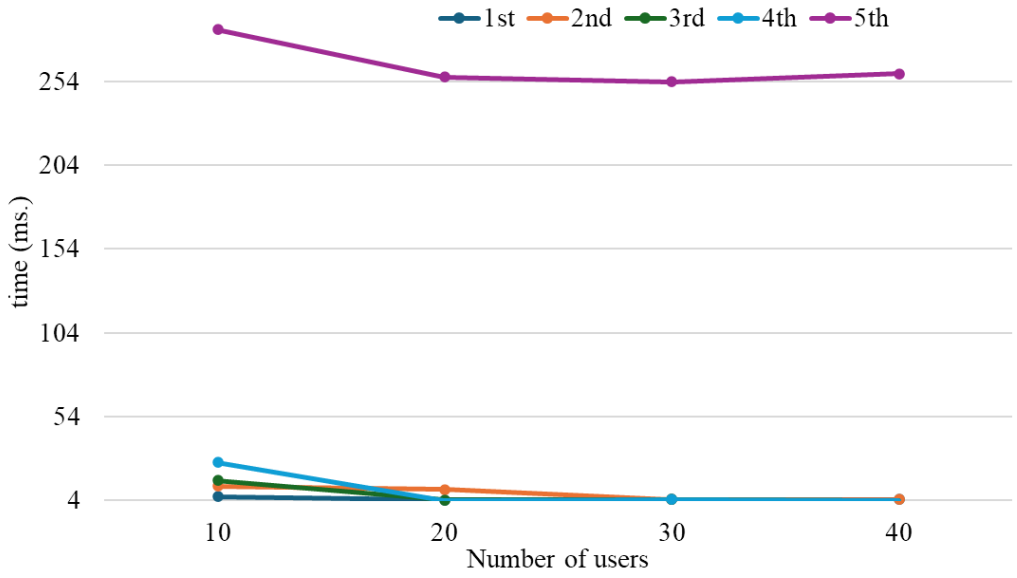
Impact of request aggregation





Impact of caching





Key take-aways and next steps

- NDN can be used for implementing distributed in-network context brokers
- Mapping from ETSI NGSI-LD API to NDN API is straightforward ➞ This enables the development of proxies allowing the integration of legacy endpoints
- NDN's support for advanced communication paradigms offer great advantages
- But needs support for one-to-many communication ➞ this work
- But needs new security mechanisms ➞ our previous work (see our paper for references)
- But needs support for subscriptions➞ ongoing work
- But needs support for temporal queries➞ ongoing work

Thank you

xgeorge@aueb.gr