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

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.
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
Motivation: Widespread IoT Applications, a challenging endeavor

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?

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

- 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

**IoT Devices**

- $p_1$
- $p_2$
- ...
- $p_i$

$\Pi(M)$ context aware messages

**Publish**

**Brokers**

- $b_1$
- $b_2$
- $b_{k1}$
- $b_{k2}$
- $b_3$

**Federation Topology**

**Widespread IoT Applications**

$\Sigma(T)$ topics of messages

- $S_1$
- ...
- $S_n$

**Subscribe**
ComDeX Architecture: Modeling Static/Dynamic Data
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:     $e_j \leftarrow x$
5:     $e_j.attr.ea_x \leftarrow$ edges.of.$x$
6:     for each $e_j.attr.ea_x$ do
7:       $t_j \leftarrow e_j.type + e_j.attr.ea_x.type + ea_x.id$
8:       $m_j.topic \leftarrow t_j$
9:       $m_j.payload \leftarrow ea_x.value$
10:      print($m_j$)
11:   end for
12: end for
13: end procedure

Messages

<table>
<thead>
<tr>
<th>Topic</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>Building/Entity_n6/HasDynamicProperty/Occupancy</td>
<td>value:32</td>
</tr>
<tr>
<td>Building/Entity_n6/HasStaticProperty/address</td>
<td>value:Street_1</td>
</tr>
<tr>
<td>Building/Entity_n6/HasRelationshipWith/HasRoom</td>
<td>value:Street_1</td>
</tr>
</tbody>
</table>
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/Entity_n6

Create Advertisement

Find Advertisement

Forward Advertisement
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/Entity_n6`

**Legend**

- Subscribe to provider Advertisements
- Create provider Advertisements
- Propagate Advertisements

---

Create Advertisement

Forward Advertisement

Find Advertisement
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

Community A: Smart Port Authority Community

Community B: InterCityBus Transit Community

Community C: Firefighting Community

APP
Restrictions

Who owns it?

Existing Community Applications have to connect to new broker to discover data!
ComDeX Architecture: Action Handler

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

---

**LVL N**

- $\pi(m_n)$
- $\sigma(T_l)$
- $\sigma(s_l)$
- $\pi(m_j)$

**LVL 2**

- $A_{b2} = a_1, \ldots, a_j$
- $E_{b2} = \emptyset$
- $\sigma(t_2)$
- Forward($A_{b1}, b_{b1}$)

**LVL 1**

- $A_{b1} = a_1, \ldots, a_j$
- $E_{b1} = \varepsilon_1, \ldots, \varepsilon_j$
- $\sigma(t_1)$
- $\pi(m_j)$
- $\sigma(s_1)$
- $\pi(m_n)$

---

ComDeX: A Context-aware Federated Platform for IoT-enhanced Communities
From theory to practice: Prototype Implementation
From theory to practice: Prototype Implementation
Experimental Evaluation: Testbed

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Instance Type</th>
<th>Instance Family</th>
<th>Instance Size</th>
<th>VCPUs</th>
<th>Memory (GIB)</th>
<th>Network Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brokers</td>
<td>c5.large</td>
<td>c5</td>
<td>large</td>
<td>2</td>
<td>4</td>
<td>Up to 10 Gigabit</td>
</tr>
<tr>
<td>Subscribers/Publishers</td>
<td>t3.nano</td>
<td>t3</td>
<td>nano</td>
<td>2</td>
<td>0.5</td>
<td>Up to 5 Gigabit</td>
</tr>
<tr>
<td>Publisher (§8.2)</td>
<td>t3.large</td>
<td>t3</td>
<td>large</td>
<td>2</td>
<td>8</td>
<td>Up to 5 Gigabit</td>
</tr>
</tbody>
</table>
Experimental Evaluation: Metrics

Advertisement Installation Time = \text{arrival\_time\_at\_top\_broker} – \text{creation\_time\_at\_edge\_broker}

Advertisement Installation Time = t_3 - t_1

Subscription Notification Latency: the time, from the creation of a Publication \(\pi_j\) at a broker \(b_k\) until its reception by an interested subscriber \(s_j\)
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)
Approx. 15k entities:
Buildings, rooms, devices e.t.c

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
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

Try out ComDeX!

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