



**HAL**  
open science

## **Enabling IOT-enhanced transportation systems using the NGSI protocol**

Georgios Bouloukakis, Chrysostomos Zeginis, Nikolaos Papadakis, Panagiotis Zervakis, Dimitris Plexousakis, Kostas Magoutis

### ► **To cite this version:**

Georgios Bouloukakis, Chrysostomos Zeginis, Nikolaos Papadakis, Panagiotis Zervakis, Dimitris Plexousakis, et al.. Enabling IOT-enhanced transportation systems using the NGSI protocol. The 12th International Conference on the Internet of Things(IoT 2022), Faculté de génie du design industriel (IDE), Nov 2022, Delft, Netherlands. pp.33-40, <10.1145/3567445.3567460>. <hal-03854196>

**HAL Id: hal-03854196**

**<https://inria.hal.science/hal-03854196v1>**

Submitted on 15 Nov 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

# Enabling IoT-enhanced Transportation Systems using the NGSI Protocol

Georgios Bouloukakis<sup>1</sup>, Chrysostomos Zeginis<sup>2</sup>, Nikolaos Papadakis<sup>2,3</sup>, Panagiotis Zervakis<sup>3</sup>,  
Dimitris Plexousakis<sup>2,3</sup>, Kostas Magoutis<sup>2,3</sup>

georgios.bouloukakis@telecom-sudparis.eu, {zegchris, papadakni, dp, magoutis}@ics.forth.gr, csd3267@csd.uoc.gr

<sup>1</sup>Télécom SudParis, Institut Polytechnique de Paris, France

<sup>2</sup>Institute of Computer Science (ICS), Foundation for Research and Technology - Hellas (FORTH)

<sup>3</sup>Computer Science Department, University of Crete, Greece

## ABSTRACT

This paper presents a model-based approach to facilitate the development of IoT applications in Transportation Systems. Existing public transportation services are provided by relying on standard data models such as GTFS. However, such models are limited in representing IoT-based infrastructures and the locations that IoT devices cover (e.g., bus seating areas). We introduce a context-aware publish/subscribe IoT platform that supports synchronous data requests, asynchronous notifications and analytics applications. Data requests are created using the system's context, which in our case is based on a transport bus system. Both static and dynamic context properties are modeled by extending the NGSI smart data models. We then introduce a GTFS-to-NGSI mapping tool to enable the enhancement of existing GTFS-based transportation systems with IoT capabilities. We develop a prototype of our platform and we demonstrate the applicability of our approach using open data from the Roma Mobilità bus transportation system.

## CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

## KEYWORDS

Transportation Systems, IoT, Data modeling

### ACM Reference Format:

Georgios Bouloukakis<sup>1</sup>, Chrysostomos Zeginis<sup>2</sup>, Nikolaos Papadakis<sup>2,3</sup>, Panagiotis Zervakis<sup>3</sup>, Dimitris Plexousakis<sup>2,3</sup>, Kostas Magoutis<sup>2,3</sup>. 2022. Enabling IoT-enhanced Transportation Systems using the NGSI Protocol. In *Proceedings of the 12th International Conference on the Internet of Things (IoT '22)*, November 7–10, 2022, Delft, Netherlands. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3567445.3567460>

## 1 INTRODUCTION

With the advent of Smart Technologies in recent years, Internet of Things (IoT) devices constitute an integral part for various activities in our every daily life (e.g., Health Monitoring Devices, Home Automation, Smart Agriculture, Smart Transportation, etc.). Among

them, Smart Transportation [24] has attracted significant interest by the research community and the vehicle manufacturing, adopting new IoT solutions that improve urban mobility. In particular, Smart Transportation refers to the integrated application of modern technologies and management strategies in transportation systems. These technologies aim to provide innovative solutions relating to different modes of transport and traffic management and enable users to be better informed and make safer and 'smarter' use of transport networks. For example, *smart buses* [16] can be equipped with advanced computing and wireless communication enabling the monitoring and collection of data such as driving behavior and passenger flows, giving bus operators insights into its fleet operation and allowing them to make service improvements or timetable rearrangements when necessary.

However, Smart Transportation produces huge amount of data that should appropriately be structured, stored, and analyzed in order to extract useful information for improving transportation conditions. There are available a number of data structures and specifications for transit information, most notably the *TransXChange* XML format<sup>1</sup>, mainly used in the UK, and the *GTFS* (General Transit Feed Specification)<sup>2</sup> developed by Google, used to publish transit data in a format that can be consumed by a wide variety of software applications. GTFS is now used by thousands of public transport providers. GTFS-Realtime<sup>3</sup> is an extension of GTFS, allowing for transport agencies to provide changes to schedules and vehicle positions as they occur.

Existing public transportation services rely mainly on GTFS and GTFS-RT standard data models, which have limitations in representing IoT-based infrastructures and the context information that IoT devices provide. NGSI-LD [1] is an information model and API used generally for the IoT ecosystem and provides tools for publishing, querying and subscribing to context information. In this paper, we propose extensions to the NGSI-LD model and tools for mapping GTFS-RT to NGSI-LD, in order to provide richer context in transportation systems. Our experience so far with our prototype implementation in a bus-system and the mapping of GTFS-RT to NGSI-LD provides experimental evidence of the value represented by the new ways to relate to context within bus and bus-station spaces and in relation to real-world transportation systems. In a nutshell, the main contributions of this paper are the following:

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

*IoT '22*, November 7–10, 2022, Delft, Netherlands

© 2022 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-9665-3/22/11.

<https://doi.org/10.1145/3567445.3567460>

<sup>1</sup><https://www.gov.uk/government/collections/transxchange>

<sup>2</sup><https://gtfs.org/>

<sup>3</sup><https://developers.google.com/transit/gtfs-realtime>

- (1) A generic approach that can be following from application developers to represent in a standard way the interior of transportation systems and enhance it with IoT capabilities.
- (2) An extension of the public transportation NGSI-LD model is proposed by adding entities that represent static and dynamic properties of the bus interior space as well as the space of bus stations.
- (3) A tool for mapping GTFS-RT to NGSI-LD data models for public transportation systems is proposed. This tool this enables to further attach IoT devices to bus transportation systems.
- (4) A prototype platform that collects real data in GTFS-RT from the Rome bus system and stores them using the extended NGSI-LD entities.

The rest of this paper is organized as follows: In Section 2 we discuss related work in IoT devices and data models for smart transportation systems and NGSI-LD based platforms. Section 3 describes our proposed IoT platform architecture for transportation systems. In Section 4 we extend the FIWARE NGSI-LD Urban Mobility models with new static and dynamic properties of bus interiors and bus-station spaces, relating them with existing IoT devices that can sense/actuate dynamic bus properties. In Section 5 we describe our novel mapping of GTFS-RT to NGSI-LD data models for public transportation systems. In Section 6 we describe our prototype implementation of the IoT platform architecture for bus transportation systems, as well as our experience with mapping GTFS-RT to NGSI-LD entities with a real-world dataset from the Roma Mobilita bus transportation system. Finally, in Section 7 we conclude this paper.

## 2 RELATED WORK

IoT devices and technologies can be leveraged in public transportation systems to provide greater safety, more efficient travel, improved vehicle and aircraft maintenance, and more strategic traffic management. This section describes existing IoT devices, technologies and platform used in public transportation systems [6].

**IoT devices for transportation systems.** Data from mobile or fixed IoT devices are leveraged to estimate traffic congestion, propose optimal route options to minimize traveling times and reduce car emissions/energy consumption. In addition, IoT devices have been widely used to estimate occupancy of commuters in buses, tramways, metros and trains. There are three main methodologies to estimate occupancy: (i) monitoring passengers (using on-board technologies such as WiFi, infrared sensors, etc.); (ii) monitoring of the overall load on the vehicle (using suspensions or air springs technologies); and (iii) using ticket-based solutions such as ERF (Electronic Registering Fareboxes) [4, 17].

Recent approaches tend to estimate occupancy by relying on some of the following IoT devices [17] (summarized in Table 1). **Treadle sensors/tread mats** are generally located on the ground in vehicle doorways. They measure passenger footfall using a pressure switch located in a floor mat. The system can misinterpret the direction of passengers and passenger numbers but despite this, manufacturers declare that this type of APC can have an accuracy of 95% under most operating conditions. The **infra-red sensors** use emitters that are generally set parallel to one another so that

Technology	Special Vehicle Requirements	Granularity	Accuracy	Cost	Privacy
<i>Treadle Mat Sensors</i>	Raised entrances followed by steps	Fine-Grain Occupancy	High	Average	Protected
<i>WiFi</i>	None	Coarse-Grain Occupancy	Average	Low	Concerns
<i>Infrared Sensors</i>	None	Fine-Grain Occupancy	High	High	Protected
<i>Cameras</i>	None	Fine-Grain Occupancy	High	High	Unprotected
<i>Electronic Registering Fareboxes</i>	Enough Free Space at Entry	Fine-Grain Occupancy	Average	Average	Concerns
<i>Suspension load sensor</i>	Compatible shock absorbers	Coarse-Grain Occupancy	Low	Average	Protected
<i>Time of Flight Sensors</i>	None	Fine-Grain Occupancy	High	Average	Protected

**Table 1: Properties of Automated Passenger Counting Technologies**

the LED interruption occurs in the direction of crossing and the entry direction can thus be discriminated from the exit one (because of their “bar” configuration; this type of infra-red sensor is also defined as *barrier sensors*). Finally, **Time-of-Flight sensors** [3] detect people as they enter and exit a room. Although called “People Counter”, they simply detect a shape under the door jam and determine the direction of travel. Then, the device uses this information to determine whether the shape enters or exits the room.

There are various considerations when choosing an automated passenger counting technology for smart public transportation. There is no single correct solution for this. For example, cameras combined with state of the art image recognition technologies [13] show performance that satisfies the specifications in terms of speed and counting outcomes. However, they violate anonymity and are intrusive. Treadle mats have special requirements as raised entrances are necessary, followed by steps, as one step is insufficient to distinguish between activation for entry to, or departure from, a vehicle. With WiFi, even if it is cheap, accuracy falls as numerous people may carry multiple WiFi-enabled devices or they may carry no devices. Using Electronic Registering Fareboxes, due to the amount of trespassers using public transportation, it is impossible to correctly count the number of passengers. Employing, suspension load sensors, it is challenging to account for the dynamic load’s fluctuation on the shock absorbers, and can estimate overall load rather than individual passenger count.

**Data models for transportation systems.** To represent data that model transportation activities such as setting schedules, managing fleets, issuing tickets, providing real-time data, a numerous of diverse data models have been developed [18]. Bus transportation systems rely on different data models to represent both static and real-time aspects of fixed-route transit data (see Table 2). The **General Transit Feed Specification (GTFS)**, defines a common format for public transportation schedules and associated geographic information. GTFS “feeds” let public transit agencies publish their transit data (stops, routes, trips, fares, etc.) and developers create applications that consume that data in an interoperable way. **GTFS-realtime (GTFS-RT)**, an extension to the general specification GTFS, is a feed specification that allows public transportation agencies to provide real-time updates (e.g., geographic data) about their fleet to application developers. Transit agencies provide three different types of live feed-trip updates (delays, cancellation, changed routes), service alerts (stop relocation, events affecting a station, route, or entire network), or vehicle positions. Additional fields can be added to GTFS datasets to accommodate a variety of

application-specific needs, even if these fields are not in the official specification. Thus data models exist that are essentially extensions to GTFS, which focus on a specific additional functionality. Some of these are GTFS-ride [8] and GTFS-flex [20] (demand-responsive transportation services).

**SmartDataModels** have been introduced by FIWARE Foundation<sup>4</sup> to enable data portability in applications for Smart Cities, Smart Environment, Smart Energy, to name a few. There are two data models (compliant to NGSI version 2.0 and NGSI-LD protocols) that deal with transportation data representation: (i) *Transportation*<sup>5</sup> describes the main entities involved with smart applications that deal with transportation issues (traffic monitoring, vehicle diagnostics, traffic violations). This set of entities is primarily associated with the automotive and smart city vertical segments and related IoT applications; and (ii) *Urban Mobility*<sup>6</sup> leverages GTFS fields and maps them with NGSI attributes. Nonetheless, for some entity types, extra attributes are suggested in order to better support the data model using the NGSI information model. This data model was first adopted and developed in the SynchroniCity project [5]. **NGSI-LD based platforms.** SynchroniCity deployed a major large scale pilot across multiple cities using minimal interoperability mechanisms [5]. NGSI-LD is used to provide a multi-modal transportation routing service in the city of Santander. Fed4IoT [7] uses unique IoT virtualization technologies to federate IoT and Cloud Infrastructures in order to create scalable and interoperable Smart Cities applications. They use several different standards and data models for their platform, including oneM2M, NGSIv2 and NGSI-LD. The Smart City Information System project [10] serves as a conduit for disseminating information and policies relating to smart cities, such as plans, projects, R&D, and governance, that are pushed by the public. To cover the needs of the project, they developed their own NGSI-LD compliant context broker called DataHub. SWAMP [11] uses a hands-on approach to develop an IoT-based smart water management platform for precision irrigation in agriculture using FIWARE technologies and the NGSI-LD protocol. There are many other recent projects that use NGSI-LD technologies such as agriculture (DEMETER [15], IoF2020 [19]), aquaculture (iFishIENCi [14]) and smart buildings (DomusSapiens<sup>7</sup>, Phoenix<sup>8</sup>).

This paper leverages the aforementioned state-of-the-art IoT technologies to introduce a generic architecture for transportation systems. We extend existing NGSI-LD models to precisely model static properties of bus systems, as well as dynamic properties that are attached to IoT devices. This allows data from existing GTFS-RT bus systems to be collected and converted to IoT-enhanced NGSI-LD data representations.

### 3 IOT PLATFORM FOR TRANSPORTATION SYSTEMS

While current IoT platforms and related technologies support the development of advanced transport IoT applications, it is challenging to realize advanced transportation systems due to the following: (i) existing platforms are developed for specific agencies and they

<sup>4</sup><https://smartdatamodels.org/>

<sup>5</sup><https://github.com/smart-data-models/dataModel.Transportation>

<sup>6</sup><https://github.com/smart-data-models/dataModel.UrbanMobility>

<sup>7</sup><https://www.jogroup.eu/en/projects/domus-sapiens>

<sup>8</sup><https://cordis.europa.eu/project/id/893079>

	GTFS	GTFS-RT	Existing NGSI-LD	Our proposed NSGI-LD
<i>File Format</i>	.txt	.proto	.ngsi-ld	.ngsi-ld
<i>Static Data</i>	Yes	No	Yes	Yes
<i>Dynamic Data</i>	No	Yes	Yes	Yes
<i>IoT-Device Physical Properties</i>	No	No	Yes	Yes
<i>Dynamic Data Observation/Device Decoupling</i>	-	-	No	Yes
<i>Vehicle Data</i>	No	Some	Yes	Yes
<i>Vehicle Spatial Context Data</i>	-	No	No	Yes
<i>Station Data</i>	Yes	No	Yes	Yes
<i>Station Spatial Context Data</i>	No	-	No	Yes

Table 2: Existing data models of transportation systems

require the agency's application to leverage its services. How can we develop applications that can be portable to multiple transportation systems. For example, leveraging the same bus application in the commuter's home town (e.g., Rome) or in a city (e.g., Heraklion) that visited for leisure. (ii) there are ongoing efforts to enhance transportation systems using IoT devices and technologies. However, development solutions are constrained to specific devices. (iii) while the GTFS standard is used to describe static and real-time information of such systems, data notifications are not supported, as well as the compatibility with the IoT is limited. In addition, the context representation of transportation systems is not supported as well as in-depth representation, including static/dynamic information for bus areas, doors, stations, etc.

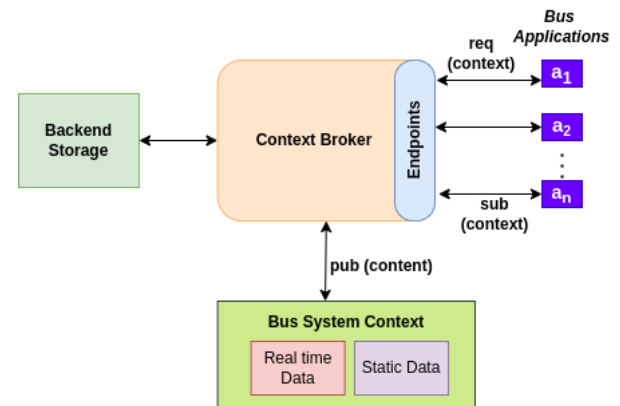


Figure 1: The Bus System Prototype Implementation.

To enable the development of advanced services/applications for transportation systems, we propose an IoT-based platform architecture. As depicted in Fig. 1, this consists of the following basic components:

*Transportation System Context:* it includes both static and real-time data that are related to transportation systems. For example, for bus systems, static data include information related to bus stations, stops, routes, bus fleet, to name a few. Real-time data are related to dynamic bus locations, occupancy, arrivals, etc. Note that real-time data is usually provided from IoT devices and it can be enriched with fine-grained context information. For example, the real-time

occupancy at the back seating area of a specific bus or the number of people entered through the back door during a specific day.

*Context broker*: it is a publish/subscribe message broker that is responsible for collecting both static and real-time *context-enriched* data from publishers (e.g., buses, stations, etc). The context broker supports *asynchronous subscriptions* by matching and forwarding data to interested subscribers (i.e., applications, see below); and synchronous requests using the back-end storage component (see below). Subscriptions or requests of data are received using the exposed endpoints of the context broker.

*Backend Storage*: it receives data from the context broker to be stored or requests of data to serve future requests. While the backend storage is mainly used to store static information, data analytics can be supported using real-time data that are stored in a time-series database.

*Applications*: can be developed either by requesting static data or by subscribing to receive real-time notifications. In both cases the transportation system context can be used as a *parameter* to requests or as a *filter* subscriptions. For instance, *what is the current occupancy in the back seating area of the bus Roma Center* or *notify me when the bus Roma Center is near my current location and the closest bus station*. Finally, *analytics applications* can be developed using the real-time data that is stored in the Backend Storage Server for historical purposes.

In this paper, we leverage the FIWARE smart data models to enable the IoT-enhanced, context-aware and generic representation of transportation systems. We demonstrate this *generic modeling approach* using the bus transportation system. To explore unprecedented capabilities of smart bus systems, we extend existing FIWARE models by including the spatial properties of bus areas which are further decoupled from IoT devices. IoT devices are used to dynamically update characteristics of bus areas without constraining them to rely on specific IoT devices. This work aims to enable developers building portable applications for any transport bus system by relying on a model-driven approach [2].

## 4 NGSI-LD ENTITIES FOR BUS TRANSPORTATION SYSTEMS

In this section, we rely on the FIWARE NGSI to provide specific models for bus transportation systems. Note that, smart transportation system designers may follow this approach to model the interior spaces of any public transportation vehicle. In particular, while existing NGSI models provide models for vehicles, these are too generic to be used for modeling buses. Thus, bus transportation system developers have to arbitrarily define properties related to buses. We extend the NGSI *Transportation* models by adding properties related to static and dynamic properties of the interior bus space, as well as to the space of bus stations. Our properties enable the relation of existing IoT devices that can sense/actuate dynamic properties of bus transportation systems. Our data models can be found at: <https://github.com/SAMSGBLab/iotspaces-DataModels>

### 4.1 The NGSI-LD Bus model

We use the NGSI *Transportation* data model (see §2) to extend the following entities: (i) *Vehicle Model*: to model a particular vehicle

model, including all properties which are common to multiple vehicle instances; (ii) *Vehicle*: to model a specific vehicle. Fig. 2 presents the extended *Bus Models* and their relationships with other existing as well as with new entities.

The *Bus Vehicle Model* entity includes all properties (length, width, etc.) which are common to multiple buses belonging to the same model. The *Bus* entity includes properties for a specific bus (e.g., the Citaro G Centre Bus in Roma 3). To accurately model specific areas of buses, we must take into consideration the vehicle types used in the targeted region. In Europe, buses constructed for urban transportation must follow strict standards [21]. In this paper, we select the *Mercedes Citaro G (3 door) version* since it is commonly used in EU bus transportation systems<sup>9</sup>.

As depicted in Fig. 3, we first define the bus extent (i.e., its spatial properties) which corresponds to the *busExtent* property. To make this property flexible and generic, each extent can be defined using diverse coordinates such as 2D, 25D and 3D (see Listing 1 lines [4-9]). Then, the Bus entity has relationships to entities used to further define specific areas (*hasAvailableAreas* relationship) of the bus (e.g., driver area, seating areas, doors, etc., see Listing 1 lines [10-12]), as well as IoT devices placed in specific locations of the bus (e.g., WiFi, GPS, etc., see Listing 1 lines [14-16]). Finally, a Bus static (capacity) and dynamic (occupancy) attributes. Dynamic attributes can be associated with observations captured from IoT devices (Listing 1 lines [18-21]).

```

1 ...
2 "type": "Bus",
3 "name": "Centre Bus Roma 3",
4 "busExtent": {
5   "type": "Polygon",
6   "measurementUnit": "m",
7   "dimensions": "2D",
8   "coordinates":
9     [[ [0,0], [18.125,0], [18.125,2.55], [0,2.55], [0,0] ] ],
10  "hasAvailableAreas": [
11    "urn:ngsild:Area:id:1",
12    "urn:ngsild:Area:id:door1",
13  "RefBusVehicle Model": [urn:ngsild:BusVehicleModel:YZX02],
14  "Devices": [
15    "urn:ngsild:device_N01", "urn:ngsild:device_N02",
16    "urn:ngsild:device_N03"],
17  "capacity": "117",
18  "Observations": [
19    "urn:ngsild:observation_N01",
20    "urn:ngsild:observation_N02",
21    "urn:ngsild:observation_N03"],
22 ...

```

Listing 1: The NGSI-LD Bus entity.

*Bus Area* entities are used to define multiple areas of the bus (driver area, door area, etc.). As depicted in Fig. 3, such entities can be defined using the *relativePosition* property (polygon extent for the driver area, see Listing 2 lines [4-9]); or line extent for the door 3 area, see Listing 2 lines [13-20]). Finally, similar to Bus entities, bus areas have dynamic attributes that can be associated with observations (e.g., occupancy, people count, etc.) captured from IoT devices (Listing 2 lines [10-11]).

```

1 ...
2 "type": "Bus Area",
3 "name": "Driver Area",
4 "relativePosition": {
5   "type": "Polygon",
6   "measurementUnit": "m",

```

<sup>9</sup>Note that system designers can follow the approach presented in this paper to introduce similar models for other bus vehicle types and in general for any public transportation vehicle.

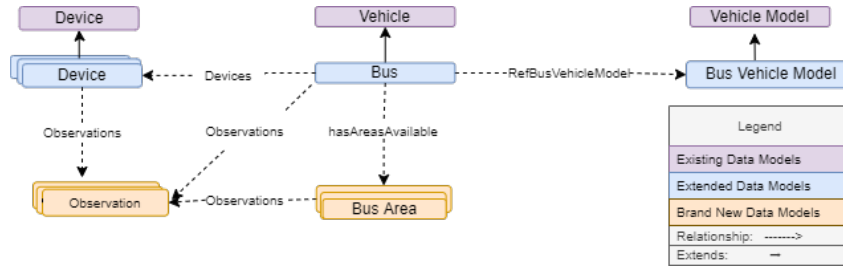


Figure 2: The proposed NGSI-based transport bus system data model

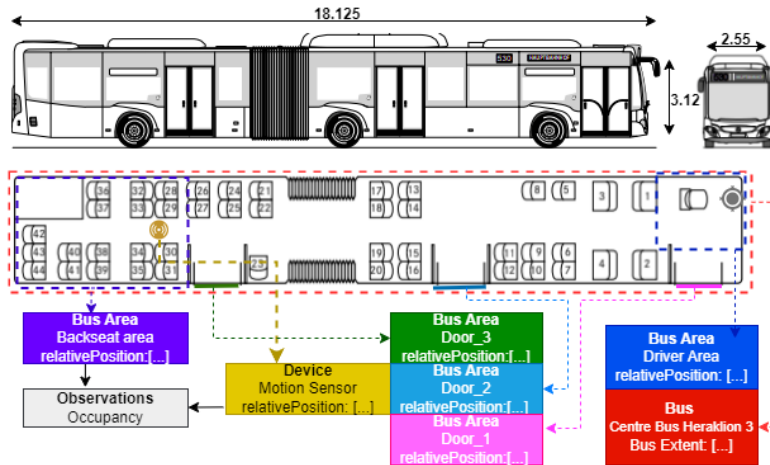


Figure 3: Bus areas to model their entities and attributes.

```

7  "dimensions": "2D",
8  "coordinates":
9  [[ [17.5,0],[18.125,0],[18.125,1],[17.5,1],[17.5,0]] ],
10 "Observations": [
11   "urn:ngsild:observation_N01",
12 ...
13 "type": "Bus Area",
14 "name": "Door_3",
15 "relativePosition": {
16   "type": "line",
17   "measurementUnit": "m",
18   "dimensions": "2D",
19   "coordinates":
20   [[ [5.5,2.55],[7,2.55]] ]
21 ...

```

Listing 2: The NGSI-LD Bus Area entity (driver example).

Then, we use the NGSI *Smart Sensing* data model to extend the *Device* NGSI-based entity to enable the decoupling of specific sensor observations (captured attributes) from devices. Based on Listing 3, along with other properties, the *captures* property is associated to one or more observations. Thus, developers will be able to design portable applications using high-level observations instead of attributes of specific devices (inspired by the SemIoTic approach [22, 23]).

```

1 ...
2 "type": "Device",
3 "category": "sensor",
4 "captures": ["urn:ngsild:Observation_N03"],
5 "depth": 3,
6 "deviceState": "ok",
7 "direction": "Outlet",

```

```

8 "ipAddress": [
9   "99.999.999.99" ],
10 "value": "15",
11 ...

```

Listing 3: The NGSI-LD device entity.

Finally, Listing 4 shows the *Observation* entity that is used to define multiple observations to be associated with existing or future IoT devices.

```

1 ...
2 "id": "urn:ngsild:Observation_N01",
3 "type": "Observation",
4 "name": "Occupancy",
5 "observationDateTime": "2022-02-21T07:36:17+0000",
6 "value": 15,
7 ...

```

Listing 4: The NGSI-LD observation entity.

### 4.2 The NGSI-LD Bus Station model

By following a similar approach to the one presented in the previous subsections, we now extend (Fig. 4) the *Transport Station* entity of the NGSI *Transportation* data model (see §2). In a bus transportation system, this entity could represent a bus station or a bus stop and thus, we introduce the *Bus Station* entity. The extent of a bus station can be splitted to multiple areas such as bus stop, ticket booth, bus parking, etc., using the *Bus Station Area* entity. Then, static and dynamic properties can be assigned to a *Bus Station* or *Bus Station*

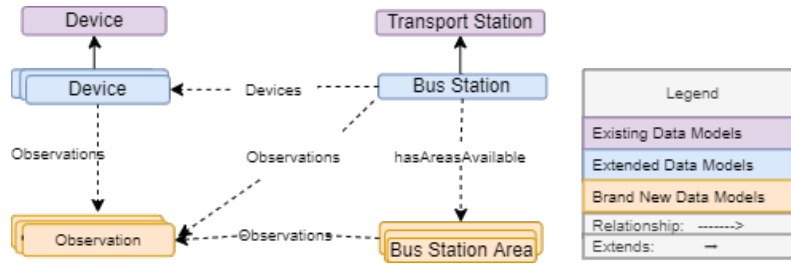


Figure 4: The proposed NGSI-LD bus station model.

Area entities. Dynamic properties are updated using IoT devices and their Observations.

## 5 GTFS-RT TO NGSI-LD MAPPING

As already mentioned GTFS-realtime (RT) is an extension of the GTFS and serves as a feed specification that allows public transportation agencies to provide real time data about the status of their fleet to developers. Although the expressiveness of GTFS-RT is adequate for travel agencies to monitor their fleet and find disruptions in their service, it does not support the modelling of IoT devices that can provide additional features for improving services provided to passengers of public transportation (e.g. people counting in buses [12], meteorological data in specific areas [9]). To the best of our knowledge, there is no existing work on completely mapping GTFS-RT to NGSI-LD entities. This section proposes such a mapping and provides specific examples from data retrieved from Rome’s urban transportation services (Roma Mobilita<sup>10</sup>).

Since GTFS-RT files are raw text files without an easily readable format we propose a mapping to structured NGSI-LD files in JSON format. For this mapping we leverage the urban mobility NGSI-LD model<sup>11</sup>. This data model defines in a harmonised representation the formats and semantics that will be used by applications both to consume and to publish data for urban mobility services.

In Listing 5, a simple header of a Trip Update feed is presented with basic header information such as: the GTFS-RT version; the incrementality of the dataset; and the moment where this dataset was generated on the server. The corresponding NGSI-LD file in JSON format is depicted in Listing 6, where a structured JSON file defines key-value pairs along with context information. Due to space limitations, we have not included GTFS-RT entities for trip updates and service alerts or vehicle positions (we provide extended examples with complete mappings in our repository<sup>12</sup>). Furthermore, Figure 5 demonstrates an NGSI-LD representation of a GTFS-RT Feed Message. The *hasFeedEntity* relationship categorizes the type of entities (*Trip Updates*, *Bus Position* or *Service Alerts*), which, in turn, are described in NGSI-LD format using specific relationships (*Trip Descriptor* or *Bus Descriptor*). For the latter, external non GTFS-RT specific entities are leveraged to provide more details for the three basic entities. We adapt the proposed NGSI-based bus transportation system datamodel (described in Section 4) by replacing

all Vehicle-related entities/relationships with the respective Bus model ones.

```

1 {                                     //FeedMessage
2   1: "1.0"                             //gtfs_realtime_version
3   2: 0                                  //incrementality
4   3: 1644784163                        //timestamp
5 }

```

Listing 5: Sample GTFS-RT feed message

```

1 {
2   "@context": "https://locationofourcontext.jsonld",
3   "id": "urn:ngsi-ld:FeedMessage:8314",
4   "type": "FeedMessage",
5   "FeedHeader": {
6     "type": "Property",
7     "value": {
8       "gtfs_realtime_version": "1.0",
9       "incrementality": 0,
10      "timestamp": 1644784163,
11    }
12 },
13 "hasFeedEntity": {
14   "type": "Relationship",
15   "object": ["urn:ngsi-ld:TripUpdates:1"]
16 },
17 }

```

Listing 6: Mapping of GTFS-RT (listing 5) to NGSI-LD

## 6 PROTOTYPE IMPLEMENTATION

We now present the prototype implementation of the platform presented in §2. As depicted in Fig. 6, this consists of the following components: (i) the context broker that is implemented using Orion-LD and Mintaka; (ii) the backend storage using MongoDB and TimescaleDB; (iii) Applications that rely on the NSGI-LD extended data model to create requests/subscriptions; and (iv) an MQTT Mosquitto broker that collects sensors data from vehicles and converts them to NGSI-LD based data using NGSI IoT agents. For our experiments we used a Time-of-Flight sensor<sup>13</sup> to simulate the counting of passengers passing the entrance of a bus. In our case we placed the sensor at the top of a working office entrance (Figure 7). This sensor is based on an emitter that sends photons which are reflected by the target (i.e. a person) and detected by the receiver. The time difference between the emission and the reception determine that distance of the target in very high accuracy.

**Orion-LD** is a data management Context Broker that supports both the NGSI-LD and NGSI-v2 APIs. Orion-LD follows the ETSI definition for NGSI-LD and has been proven to be a reliable and fast NGSI-LD broker that adheres to the NGSI-LD API specification

<sup>10</sup><https://romamobilita.it/>

<sup>11</sup><https://github.com/smart-data-models/dataModel.UrbanMobilita>

<sup>12</sup><https://github.com/SAMSGLab/iotspaces-DataModels>

<sup>13</sup><https://www.st.com/en/imaging-and-photonics-solutions/time-of-flight-sensors.html>

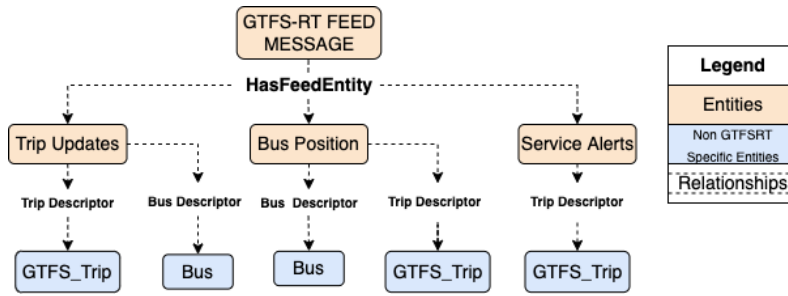


Figure 5: NGSi-LD representation of GTFS-RT format

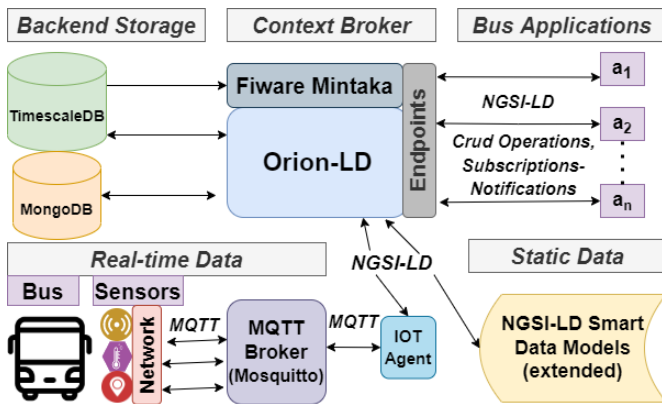


Figure 6: Transportation System Prototype Implementation.

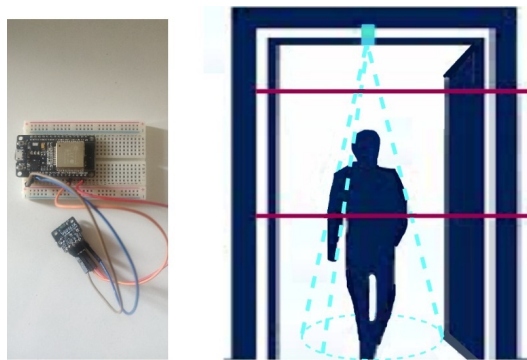


Figure 7: The Time of Flight Sensor

version 1.3.1. Alongside Orion-LD, we leverage **Mintaka**, an implementation of the NGSi-LD temporal retrieval API. It relies on the Orion-LD Context Broker to provide the underlying database and should be viewed as a component of Orion-LD.

To store context data information such as data entities, subscriptions and registrations, Orion-LD relies on **MongoDB**, which is an open source NoSQL database management program. In addition Orion-LD leverages **TimescaleDB** to provide historical time series data (NGSi-LD temporal retrieval). TimescaleDB is an open-source database that makes SQL for time-series data scalable. It's built on

top of PostgreSQL and bundled as a PostgreSQL extension, with automatic time and space partitioning (partitioning key) and complete SQL support.

To receive data from sensors, Eclipse Mosquitto is leveraged that is a lightweight system used on a wide range of vehicle sensors, from low-power single-board computers to full-fledged servers. It acts as a central communication point, passing MQTT topics between the IoT Agent and IoT devices as necessary. An IoT Agent is a component that allows a group of devices to use their own native protocols (e.g., MQTT) to deliver data to the context broker.

We now examine a potential use case using our architecture. Imagine that there is a potential passenger that is wheelchair-bound and wants to know if the bus nearest to their current position can accommodate them. The information that the passenger wants is whether the bus is wheelchair accessible and if the wheelchair spots are currently unoccupied (often occupied by strollers and standing passengers in tightly packed buses). Our architecture, combined with the NGSi-LD extended data models provide an end-to-end path for estimating bus occupancy. Sensor(s) transmit the calculated occupancy data for the specified handicapped-spot bus areas via native protocols to the MQTT-based broker. The broker forwards the data to an IoT agent, which translates it to NGSi-LD and then passes it to the Orion-LD broker (to the appropriate entity, e.g., "Observation"). The data is then stored in the mongoDB database as the latest handicapped-area(s) occupancy information. A GET request to the fitting attribute that holds the bus wheelchair accessibility (e.g. yes/no) and the occupancies ("Observations") that correspond to the handicapped "bus area(s)", would be enough to provide what was requested by a passenger application.

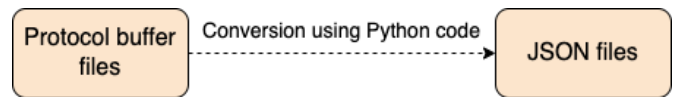


Figure 8: The conversion for the data collected from Rome.

For the the prototype implementation we collected data from the Roma Mobilita bus services for one week. This dataset takes into account all the routes in hours with various traffic and congestion conditions. We focus mainly on the GTFS-RT entities which are in protocol buffers format. For that reason, as depicted in Fig. 8, the raw text files were processed and converted to JSON, using a python script, in order to be more comprehensible. These files

were analyzed and were used for the mappings from GTFS-RT to NGSi-LD entities, as shown in section 5.

## 7 CONCLUSION & FUTURE WORK

In conclusion, we have shown that our extensions to the existing NGSi-LD models and tools for mapping GTFS-RT to NGSi-LD offer novel ways to provide richer context in transportation systems. Our experience so far with our prototype implementation of the bus-system prototype and the mapping of GTFS-RT to NGSi-LD provides evidence of the value represented by the new ways to relate to context within bus/bus-station spaces and in relation to real-world transportation systems. There is still the need to validate the correctness of the mapping of GTFS-RT to our NGSi-LD data model via an automated way, such as performing the reverse conversion and checking for differences from the initial files. Our ongoing evaluation extends validation of our prototypes in real transportation systems.

## ACKNOWLEDGMENTS

We thankfully acknowledge support by the Greek Research Technology Development and Innovation Action “RESEARCH-CREATE-INNOVATE”, Operational Programme on Competitiveness, Entrepreneurship and Innovation (EIIAνEK) 2014-2020, Grant T2EAK-02848 (SmartCityBus). We also acknowledge discussions with the consortium of the SmartCityBus project in relation to the broader smart transportation system that is being developed within the project, which helped shape the requirements for our work.

## REFERENCES

- [1] Abid Ahmed, Medvedev Alexey, Hassani Alireza, Le Gall Franck, Tropea Giuseppe, Martinez Juan Antonio, Frost Lindsay, and Bauer Martin. 2021. Guidelines for Modelling with NGSi-LD. *ETSI White Paper 4* (2021).
- [2] L. Balmelli, D. Brown, M. Cantor, and M. Mott. 2006. Model-Driven Systems Development. *IBM Syst. J.* 45, 3 (jul 2006), 569–585. <https://doi.org/10.1147/sj.453.0569>
- [3] Attila Bonyár, Attila Géczy, G. Harsanyi, and Hanák Péter. 2018. Passenger Detection and Counting Inside Vehicles For eCall- a Review on Current Possibilities. 221–225. <https://doi.org/10.1109/SIITME.2018.8599285>
- [4] D. K. Boyle. 1998. Passenger counting technologies and procedures. Washington, D.C.: Transportation Research Board, National Research Council.. In *TCRP synthesis 29*. 12–13.
- [5] Flavio Cirillo, David Gómez, Luis Diez, Ignacio EliceGUI Maestro, Thomas Barrie Juel Gilbert, and Reza Akhavan. 2020. Smart city IoT services creation through large-scale collaboration. *IEEE Internet of Things Journal* 7, 6 (2020), 5267–5275.
- [6] Sabeur Elkosantini and Saber Darmoul. 2013. Intelligent Public Transportation Systems: A review of architectures and enabling technologies. In *International Conference on Advanced Logistics and Transport*. 233–238. <https://doi.org/10.1109/ICAdLT.2013.6568465>
- [7] Pedro Gonzalez-Gil, Antonio F Skarmeta, and Juan Antonio Martinez. 2020. The security framework of Fed4IoT. In *Proceedings of the Workshop on Cloud Continuum Services for Smart IoT Systems*. 1–6.
- [8] Porter J. David, Carleton Phillip, Hoover Sylvan, and Fields Ben. 2018. *STATEWIDE DATA STANDARDS TO SUPPORT CURRENT AND FUTURE STRATEGIC PUBLIC TRANSIT INVESTMENT*. Tech. rep. School of Mechanical, Industrial and Manufacturing Engineering Oregon State University. [https://www.oregon.gov/ODOT/Programs/ResearchDocuments/SPR\\_803\\_Final%20Strategic%20Public%20Transit%20Investment.pdf](https://www.oregon.gov/ODOT/Programs/ResearchDocuments/SPR_803_Final%20Strategic%20Public%20Transit%20Investment.pdf)
- [9] Jаланey Jabamony and Ganesh Ramaswamy Shanmugavel. 2019. IoT based bus arrival time prediction using Artificial Neural Network (ANN) for smart public transport system (SPTS). *International Journal of Intelligent Engineering & Systems* 13 (2019).
- [10] Seungmyeong Jeong, Seongyun Kim, and Jaeho Kim. 2020. City data hub: Implementation of standard-based smart city data platform for interoperability. *Sensors* 20, 23 (2020), 7000.
- [11] Carlos Kamienski, Juha-Pekka Soininen, Markus Taumberger, Ramide Dantas, Attilio Toscano, Tullio Salmon Cinotti, Rodrigo Filev Maia, and André Torre Neto. 2019. Smart water management platform: IoT-based precision irrigation for agriculture. *Sensors* 19, 2 (2019), 276.
- [12] Janine Kniess, Julio Cezar Rutke, and William Alberto Cruz Castañeda. 2021. An IoT Transport Architecture for Passenger Counting: A Real Implementation. In *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. 613–617.
- [13] Claire Labit-Bonis, Jérôme Thomas, and Frédéric Lerasle. 2021. Visual and automatic bus passenger counting based on a deep tracking-by-detection system. (Oct. 2021). <https://hal.archives-ouvertes.fr/hal-03363502> working paper or preprint.
- [14] Maxime Lafont, Samuel Dupont, Philippe Cousin, Ambre Vallauri, and Charlotte Dupont. 2019. Back to the future: IoT to improve aquaculture: Real-time monitoring and algorithmic prediction of water parameters for aquaculture needs. In *2019 Global IoT Summit (GloTS)*. IEEE, 1–6.
- [15] Juan A López-Morales, Juan A Martínez, and Antonio F Skarmeta. 2021. Improving energy efficiency of irrigation wells by using an iot-based platform. *Electronics* 10, 3 (2021), 250.
- [16] Rajesh Kannan Megalingam, Nistul Raj, Amal Lehar Soman, Lakshmi Prakash, Nivedha Sathesh, and Divya Vijay. 2014. Smart, public buses information system. In *2014 International Conference on Communication and Signal Processing*. 1343–1347. <https://doi.org/10.1109/ICCSP.2014.6950068>
- [17] Ivano Pinna and Bruno Dalla Chiara. 2010. Automatic passenger counting and vehicle load monitoring. *Ingegneria Ferroviaria* 65 (02 2010), 101–138.
- [18] Mario Scrocca, Marco Comerio, Alessio Carenini, and Irene Celino. 2020. *Turning Transport Data to Comply with EU Standards While Enabling a Multimodal Transport Knowledge Graph*. 411–429. [https://doi.org/10.1007/978-3-030-62466-8\\_26](https://doi.org/10.1007/978-3-030-62466-8_26)
- [19] Harald Sundmaeker, Conny Graumans, Juanjo Hierro, Jason Roesbeke, D Urdu, and Peter van der Vlugt. 2020. *IoF2020-Position Paper on a Common European Data Space: Expert Workshop on a Common European Agricultural Data Space*. Technical Report. IoF2020 Internet of Food & Farm.
- [20] Craig Thomas and Shippy Weston. 2020. *GTFS Flex – What Is It and How Is It Used?* Technical Report. National Center for Applied Transit Technology. [https://n-catt.org/wp-content/uploads/2020/12/GTFS-Flex\\_WhitePaper\\_Final.pdf](https://n-catt.org/wp-content/uploads/2020/12/GTFS-Flex_WhitePaper_Final.pdf)
- [21] UN/ECE. 2015. Regulation No 107 of the Economic Commission for Europe of the United Nations (UNECE) – Uniform provisions concerning the approval of category M2 or M3 vehicles with regard to their general construction. *Official Journal of the European Union* (2015), 1–115. <http://data.europa.eu/eli/reg/2015/107/oj>
- [22] Roberto Yus, Georgios Bouloukakis, Sharad Mehrotra, and Nalini Venkatasubramanian. 2019. Abstracting Interactions with IoT Devices Towards a Semantic Vision of Smart Spaces. In *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation* (New York, NY, USA) (*BuildSys ’19*). 91–100. <https://doi.org/10.1145/3360322.3360859>
- [23] Roberto Yus, Georgios Bouloukakis, Sharad Mehrotra, and Nalini Venkatasubramanian. 2022. The SemIoTic Ecosystem: A Semantic Bridge between IoT Devices and Smart Spaces. *ACM Transactions on Internet Technology – TOIT* (2022).
- [24] Fotios Zantalis, Grigorios Koulouras, Sotiris Karabetos, and Dionisis Kandris. 2019. A Review of Machine Learning and IoT in Smart Transportation. *Future Internet* 11, 4 (2019). <https://www.mdpi.com/1999-5903/11/4/94>