Context Information Management using **NGSI-LD API**

Presented by: **ETSI ISG CIM**  For: **Public review**

15 January 2019

© ETSI 2019

PERMISSION TO RE-USE THIS MATERIAL IS GRANTED (provided that whole pages are copied unmodified, including ETSI (C) Copyright)
Contents

This introduction has independent sections with increasing technical details:

- ETSI ISG CIM Mission: link up all data sources
- How info-exchange can help cities
- How can all THAT Information be handled ?
- Example: The happy policeman
- Information Model and Query Language
- Architectures
- Problem: A babel of Ontologies
- Not alone !
- NGSI-LD CURRENT STATUS
MISSION
ETSI ISG CIM: Mission

... to make it easier for END-USERS and CITY DATABASES and IoT internet-of-things and 3rd-party APPS to exchange INFO
Context Information Management: exchange data **AND** definitions (ontology)

- **User Apps**
  - Simple coding!
  - High data rate
  - Ontology re-use

- **A.I.**
  - Context Information Management

- **Open Data**
  - Solve the issues
  - Highly diverse databases
  - Restricted access

- **IoT**
  - Proprietary Data

© ETSI 2019
Exchanging data and ontology allows Users and A.I. to see **meaning**.
Context Information Management: exchange data AND definitions (ontology)

... and: PROVENANCE, licensing, privacy! USAGE, Billing, FOAF info, errors!
HOW INFO-EXCHANGE CAN HELP CITIES
# Common Smart City Components

<table>
<thead>
<tr>
<th>Smart homes</th>
<th>Smart parking lot</th>
<th>Healthcare</th>
<th>Weather &amp; water systems</th>
<th>Transportation &amp; Vehicular traffic</th>
<th>Environmental pollution</th>
<th>Surveillance Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demand Response</td>
<td>Number of cars</td>
<td>Tracking</td>
<td>Weather condition</td>
<td>Camera monitoring</td>
<td>Green house gas monitoring</td>
<td>CCTV</td>
</tr>
<tr>
<td>Fire detection</td>
<td>Departure &amp; arrivals</td>
<td>Identification</td>
<td>Water quality</td>
<td>Environment Monitoring</td>
<td>Energy efficiency monitoring</td>
<td>Violent detection</td>
</tr>
<tr>
<td>Temperature monitoring</td>
<td>Environment Monitoring</td>
<td>Data gathering</td>
<td>Water leakage</td>
<td>Travel scheduling</td>
<td>Renewable energy usage</td>
<td>Public place monitoring</td>
</tr>
<tr>
<td>Security systems</td>
<td>Mobile ticketing</td>
<td>Sensing</td>
<td>Water level</td>
<td>Traffic jam reduction</td>
<td>Air quality monitoring</td>
<td>People &amp; object tracking</td>
</tr>
<tr>
<td>Social network supporting</td>
<td>Traffic congestion control</td>
<td>Water contamination</td>
<td>Water level</td>
<td>Assisted driving</td>
<td>Noise pollution monitoring</td>
<td>Traffic police</td>
</tr>
</tbody>
</table>

© ETSI 2019
“Review of Smart Cities based on IoT”. [https://pdfs.semanticscholar.org/3f4b/5b92464281610010c0c4264d62893567e03c.pdf](https://pdfs.semanticscholar.org/3f4b/5b92464281610010c0c4264d62893567e03c.pdf) Energies 2017, 10, 421
Example: Multi-Modal Transport

Lots of people use Google to find public transport from A→ B?

Why is it so successful?

✔ OK, it has good maps ... 😊
✔ Google has convinced many/most municipalities to publish transport routes/schedules in a simple common format GTFS


Image © Google https://goo.gl/maps/Bz7thKruraR2
Can ETSI Enable such open systems? e.g. Electric Vehicle charger Maps?

See commercial examples
https://chargemap.com or www.zap-map.com/live/
Can ETSI Enable such open systems? e.g. Environmental Data overviews?

London Air Pollution: Real-time Air Quality Index (AQI)

Source: http://aqicn.org/city/london (no endorsement implied)
Examples for open systems?

e.g. Show the air-pollution geomap near you?

e.g. Show the combined traffic/crime/rental "heat map"
    to help locate a new appartment
    to help city-planners

e.g. Combine public-transport usage data with "special deals"
    on tickets data, to help determine optimum usage ?

e.g. Compare hospital admissions data, with weather and pollution data,
    to help plan emergency services (i.e. reduce spare capacity,
    but make sure surges can be covered)

...
HOW CAN ALL THAT INFORMATION BE HANDLED?
Firstly: use Open Data 5 Star Model (W3C, Berners-Lee)

Open Data

The World Wide Web Consortium (W3C) has developed a five star model to describe different characteristics of open data, and its usefulness for people wishing to reuse it. It is being used globally as a model for assessing data readiness for re-use.

The 3-star level is considered the minimum standard for release of government’s public data of single agencies for re-use.

The 4/5-star level is considered required for multi-agency / multi-city scenarios.

* Data is visible, licensed for reuse (CC), but requires considerable effort to reuse.

** Data is visible, licensed, and easy to reuse, but not necessarily by all.

*** Data is visible and easy to reuse by all (not restricted to using specific software).

**** Data is visible, easy to use and described (with meta data) in a standard fashion.

***** Data is visible, easy to use, described in a standard fashion and meaning is clarified by being linked to a common definition (i.e. Ontology).

Adapted from: [www.slideshare.net/FI-WARE/fiware-global-summit-ngsild-ngsi-with-linked-data](http://www.slideshare.net/FI-WARE/fiware-global-summit-ngsild-ngsi-with-linked-data)
Third: „Keep it simple“ in the info exchange

Goals (maybe not all can be 100% achieved at same time?)

✔ Exchange info between any two systems
✔ Keep context information and relationships with data
✔ Minimal complexity (but as much as really needed)
✔ Attractive to programmers
✔ Adapt to security / privacy (GDPR, ENISA, ...)

Practical Assumptions (do all agree?)

✔ Federated architecture
✔ Linked Data compatible
✔ Query & Notify in same API
Context Information Management: Joining Verticals
Information-centric with developer-friendly JSON-LD

NGSI-LD Advantages
• information-centric
• JSON-LD syntax
• joining verticals

Context Information Management Layer

User Apps

IoT

Open Data

Machine Reasoning Systems

A.I.

APPs

APPs

APPs

EXAMPLE:
Citizen Complaints Photo-App Application

Wi-Fi 5G LPWAN

Machine Reasoning Systems

APPs

A.I.

APPs

APPs

NGSI-LD Advantages
• information-centric
• JSON-LD syntax
• joining verticals

Context Information Management Layer

User Apps

IoT

Open Data
Context Information Management Layer: Information-centric with developer-friendly JSON-LD

**NGSI-LD Advantages**
- information-centric
- joining verticals
- JSON-LD syntax

**EXAMPLE:** Citizen Complaints Photo-App Application

- User Apps
- Information Systems
- IoT (Wi-Fi, 5G, LPWAN)
- Machine Reasoning Systems
- A.I.
- Data Publication Platforms
- Proprietary Data
Keeping it simple: NGSI-LD API Features ( + Limits )

Information Model is Graph-based & information-centric
✔ Core concepts include Entities and Relationships
✔ Entities can have Properties and Relationships
✔ Relationships/Properties can also have Properties, Relationships

Referencing (some) defined/hierarchical vocabularies/ontologies
✔ All terms are unambiguously defined
✔ Allows users to reference their familiar information definitions

Model and Query language (is constrained so more predictable)
✔ Federation of (independent) information sources, anywhere
✔ Queries: based on entity type or ID, can filter results, can constrain scope (time, geography), constrained not to traverse
ETSI ISG CIM Deliverables


✔ DMI/CIM-001-AB (MI) Annotated Bibliography (ETSI members only)
✔ DGR/CIM-002-UC (GR CIM 002) Use Cases ➔ PUBLIC VERSION HERE
✔ DGS/CIM-006-MOD0 (GS CIM 006) Information Model(s)
✔ DGR/CIM-007-SEC (GR CIM 007) Security and Privacy
✔ DGR/CIM-008-NGSI-LD-Primer (GR CIM 008) API Intro for Developers
✔ DGS/CIM-009-NGSI-LD-API (GS CIM 009) ➔ PUBLIC VERSION HERE
Notation for instance diagrams

NGSI-LD Categories/classes/types are round-edged rectangles
- Rdf/rdfs predicates (such as rdf:type rdfs:subclassof) are represented normally, as labels on the corresponding arcs between the instances or classes

NGSI-LD Entity Instances are solid (square-angled) rectangles with text in bold
- text attached to the rectangle, not pasted as separate text block

NGSI-LD Relationships are diamonds, optionally with four "buttons"
- with arc coming in and out
- relationship should be read as a label attached to the underlying arc (directed edge) of the main graph
- Use verbs to describe relationships (as in entity-relationship diagrams)

NGSI-LD Properties are ovals
- with corresponding arc coming in and out
- Properties can be read as a label attached to the underlying arc (directed edge) of the graph, à la RDF
- Arc may be omitted if the oval is represented as adjacent to the corresponding entity or relationship, à la property graph

Values are unfilled hexagons, rather long

All elements must have line boundaries for better printing and PDFs. Any filled colours can be used to aid readability: except Values are white
Example: City Hall has smart lampposts

The townhall database has locations and functions of its smart lampposts

Town Hall and Police Department share info → both more efficient
Example: Police report an accident

Every Police Department generates tons of data (defined forms)

The police reports there was an accident at a certain time and what was damaged.
Example: Police report an accident

The police reports there was an accident at a certain time and what was damaged.

"Mercedes" 2017-07-29T12:00:00Z [8.672, 49.398]

Vehicle: A4567

Town Hall and Police Department share info → both more efficient
Example: Combined data Exchange

Vehicle

- rdf:type
- urn:ngsi-ld: Vehicle: A4567
  - inAccident
  - observedAt: 2017-07-29T12:00:00Z
  - location: [49.398, 8.672]
  - hasAttached
  - rdf:type
  - providedBy: urn:ngsi-ld: Org: Officer123
  - brandName: "Mercedes"

LegalEntity

- rdf:type
- urn:ngsi-ld: Org: Officer123
  - providedBy

StreetFurniture

- rdf:type
- urn:ngsi-ld: SmartLamppostB: Downtown1
  - location: [49.398, 8.672]
  - hasAttached

Sensor

- rdf:type
- urn:ngsi-ld: Sensor: Cam1
  - trafficFluidity: 0.9
  - accuracy: 5%

Entity Type

Entity Instance

Relationship

Property

Value
Example: Data exchange between databases - two non-IoT databases and one IoT

The townhall records smart lampposts in its databases

The town hall records e.g. that a webcam is attached to a specific lamppost and delivers videodata about the street

The police department records traffic accidents

A policeman records lamppost at position XY was hit by a car

Townhall can get a notification that a specific lamppost and the data from that lamppost may be affected

Police can query previous data from camera (nearby)
Example: Combined data Exchange: flows

urn:ngsi-ld: Vehicle: A4567

urn:ngsi-ld: SmartLamppostB: Downtown1

urn:ngsi-ld: Sensor: Cam1

Register Lamppost ID1, Location

Register Videocam, Location on Lamppost ID1

Stream videos

Lamppost ERR

Reports to Police

Requests history stream

Views stream

Ask ERRs

ERR history

Notes LamppostID

CRASH

© ETSI 2019
Example: Entity "Vehicle" and its Context in NGSI-LD

```json
{
  "id": "urn:ngsi-ld:Vehicle:A4567",
  "type": "Vehicle",
  "brandName": {
    "type": "Property",
    "value": "Mercedes"
  },
  "inAccident": {
    "type": "Relationship",
    "object": "urn:ngsi-ld:SmartLamppostB:Downtown1",
    "observedAt": "2017-07-29T12:00:00Z",
    "providedBy": {
      "type": "Relationship",
      "object": "urn:ngsi-ld:Org:Officer123"
    }
  }
}
```

- All terms are defined inside the `@context` links
- The unique URI of each term is sufficient to guarantee whether two applications are using the same data definitions.
NGSI-LD INFORMATION MODEL
NGSI-LD Information Model:
Has Entities, Relationships, Properties, Values
NGSI-LD Information Model

NGSI Entity → Physical or virtual object.
✓ It has (one) Entity Type.
✓ Uniquely identified by an Entity Id (URI)

Entity has zero or more attributes identified by a name
✓ Property --› Static or dynamic characteristic of an entity
  ✓ GeoProperty (geospatial context)
  ✓ TemporalProperty (time context)
✓ Relationship → Association with a Linked entity (unidirectional)

Properties have a value
✓ Can be a single value (Number, String, boolean), or complex (Array, Structured Value)

Relationships have an object
✓ A URI pointing to another entity (target of the relationship). Target can be a collection.
NGSI-LD Information Model

Cross-Domain, core properties for giving context to your information are defined in a mandatory way, to be used by API operations (e.g. geo queries)

✔ location → Geospatial location, encoded as GeoJSON.
✔ observedAt → Observation timestamp, encoded as ISO8601. (timestamp)
✔ createdAt → Creation timestamp (of entity, attribute). dateCreated in NGSIv2
✔ modifiedAt → Update timestamp (of entity, attribute). dateModified in NGSIv2
✔ unitCode → Units of measurement, encoded as mandated by UN/CEFACT.

Recommended practice

✔ Use URIs to identify your entities.
✔ A URN schema is available in NGSI-LD. It shows what entity type an id refers to
  ✔ urn:ngsi-ld:<Entity_Type_Name>:<Entity_Identification_String>
Meta-Model

An NGSI-LD Entity is a subclass of rdfs:Resource.
An NGSI-LD Relationship is a subclass of rdfs:Resource.
An NGSI-LD Property is a subclass of rdfs:Resource.
An NGSI-LD Value can be a subclass of rdfs:Literal (but can be complex)
An NGSI-LD Property shall have a value stated through hasValue of rdf:Property.
An NGSI-LD Relationship shall have an object stated through hasObject which is of type rdf:Property.
Meta-Model

An NGSI-LD Entity is a subclass of rdfs:Resource.

An NGSI-LD Relationship is a subclass of rdfs:Resource.

An NGSI-LD Property is a subclass of rdfs:Resource.

An NGSI-LD Value can be a subclass of rdfs:Literal (but can be complex).

An NGSI-LD Property shall have a value stated through hasValue of rdf:Property.

An NGSI-LD Relationship shall have an object stated through hasObject which is of type rdf:Property.

JOINS TWO WORLDS:

This gains the advantages of **RDF** and the **Linked Data standards** plus the expressive **Property Graphs** (which are de facto industry standard from their use in graph databases), by providing our meta-model with reification (statements about statements).

- Export/import from/to [Property graph]/[RDF graph]
- Can be easily expressed in JSON-LD (blank nodes are implicit) where JSON-LD is a W3C specification
- Model is suitable for SPARQL queries. (But not used for the API definition currently)
Cross-Domain Ontology provided in NGSI-LD

Need geospatial and temporal elements to be uniquely defined for interoperability
Cross-Domain Ontology provided in NGSI-LD

Need geospatial and temporal elements to be uniquely defined for interoperability.

What further high-level cross-domain elements are needed?
See WI-006 and examples in SAREF...
GETTING OUT THE INFO: QUERIES
Why don't we ...

Why don't we just use SQL?
Why don't we just use SPARQL?
Why don't we allow native Graph queries?
Because ...
✔ every database has its preferred approach, we cannot do all
✔ distributed database query technologies advance rapidly
✔ NGSI-LD API allows to import selected data into your favourite database and "do the queries as you prefer"
✔ we aim for a robust, simple-as-feasible subset of queries
Queries by Entity and Type

Query by id

✔ GET /entities/urn:ngsi-Id:OffStreetParking:ABCDE
✔ GET /entities?id=urn:ngsi-Id:OffStreetParking:AB23E&type=OffStreetParking

Query by list of IDs

✔ GET /entities?id=urn:ngsi-Id:OffStreetParking:AB23E,
  urn:ngsi-Id:OffStreetParking:FF11AA&type=OffStreetParking

Query by type

✔ GET /entities?type=OffStreetParking

Query by list of types

✔ GET /entities?type=OffStreetParking,OnStreetParking

Query by idPattern (if URIs are structured)

✔ GET /entities?type=OffStreetParking&idPattern=.*FF$
Queries with restrictions

Query entities that match restrictions (logical “and” using ‘;’)

✔ GET /entities?q=<Expression>; <Expression> ....

Restrictions on Values and on data types (Text, Number, DateTime...)

✔ Equal → brandName==Mercedes
✔ Equal with multiple alternatives → brandName==Mercedes,Audi
✔ Unequal → brandName!=Mercedes
✔ Greater than → temperature>20;temperature>=20
✔ Less than → temperature<10;temperature<=10
✔ Match pattern → brandName~=cedes$
✔ Match range (closed interval) → temperature==10..20
Queries by structured properties

Queries over Structured Values
✓ GET /entities?type=Building&q=address[street]==Franklinstrasse

Specific attributes can be requested
✓ GET /entities?type=Room&q=temperature>20&attrs=temperature,capacity
Queries by location

List entities located at a certain threshold distance to a reference geometry.
✓ /entities?georel=near;maxDistance==2000&geometry=point&coordinates=[-2.35, 40.78]

List entities that exist entirely within a reference geometry.
✓ /entities?georel=coveredBy&geometry=polygon&coordinates=[[[-80.190,25.774],[-66.118,18.466],[-64.757, 32.321],[-80.190, 25.774]]]
✓ georel → (near, coveredBy, intersects, equals, disjoint)
✓ geometry → (point, bbox, polygon, line)
✓ coordinates → as per GeoJSON
ETSI ISG CIM: Assumptions

API is agnostic to deployed architecture (centralized, distributed, federated)

✔️ Migration between architectures, without changing Applications
✔️ Portability of Applications, across architectures/deployments
✔️ Actual choice of architecture depends on (changeable) trade-offs

Applications only need know one URL where the API is exposed

URIs are used to represent all Entities and Relationships and Properties

✔️ systems and developers may use structured URI or also URLs
✔️ choice depends on trade-offs, security concerns, etc

Entities and Relationships and Properties are first class citizens of the API

All entities must reference some ontology (to define their type)
Various Architectures possible

Centralised

Distributed

Federated

Simplicity

NGSI-LD API
Various Architectures possible

Centralised

Distributed

Federated

More flexibility

NGSI-LD API
Various Architectures possible

NOTE: functionality of Context Broker or Context Registry is described in a separate presentation. Their roles are to allow discovery and authorised access to repositories of Context Information, as well as scaleable execution of Queries.
PROBLEM:
A BABEL OF ONTOLOGIES
Context Information Management: exchange data AND definitions (ontology)

More: PROVENANCE, licensing, privacy? USAGE, Billing, FOAF info, errors?
Context Information Management: WARNING! Re-Use definitions, not invent own!

Babel = "everyone using different words, in different ways, for same things"

Chaos = "using SAME terms to have DIFFERENT meanings"
Issue: Lack of Domain Consensus

Need consensus in each domain
e.g. Here we see 4 in Buildings
e.g. Here we see dozens in IoT ??

EC has begun regulating to reduce these barriers to trade / efficiency!
e.g. INSPIRE Directive (deadline 2019)

„To ensure that the spatial data infrastructures of the Member States are compatible and usable in a Community and transboundary context, the INSPIRE Directive requires that common implementing Rules (IR) are adopted …“

Metadata, Data Specifications, Network Services, Data and Service Sharing, Spatial Data Services, Monitoring and Reporting
Thousands of Ontologies

Libraries of thousands of vocabularies/ontologies, overlapping

✓ ISO/IEC 11179 Metadata Registries
✓ FAIRsharing.org
✓ Project Open Data
✓ Open Metadata Registry
✓ BARTOC Basel Register of Ontologies
✓ Biomedical Ontologies
✓ ...

Matching up ontologies is MUCH harder than (re)using same ones!

ETSI SmartM2M has SAREF working to help consolidate/bridge ontologies ...
SAREF: Smart Applications Reference Ontology (ongoing work within ETSI SmartM2M)

Methods:
✔ Analysis of nn semantic assets in the domain
✔ Selection of a short list (nn assets) to use for the reference ontology
✔ Translation of each of the nn semantic assets into OWL ontology
https://w3id.org/saref
✔ Mapping between semantic assets using SAREF

small extract as example
Non-Domain ontologies needed too: Different systems use different terms for ... 

system usage information
work process flow information
notification, update timing and retraction information
authentication and authorization information of various types
encryption and security key information
...

Source: [http://aura.abdn.ac.uk/bitstream/handle/2164/6051/baillie_acm_jdiq_15.pdf?sequence=1](http://aura.abdn.ac.uk/bitstream/handle/2164/6051/baillie_acm_jdiq_15.pdf?sequence=1)
Non-Domain ontologies needed too:

**KPI** are often cited, but not in same way

An extensible system to model KPIs, including key properties like accuracy [527]

Non-Domain ontologies needed too: **Provenance** is often needed, but not fully defined

A provenance-aware quality ontology [567]

Source: [http://aura.abdn.ac.uk/bitstream/handle/2164/6051/baillie_acm_jdiq_15.pdf?sequence=1](http://aura.abdn.ac.uk/bitstream/handle/2164/6051/baillie_acm_jdiq_15.pdf?sequence=1)
Conclusion: There is a severe Lack of Meta-data: Provenance, Quality, Licensing

Do you label the source and measurement technique?
Do you use timestamps and geospatial attributes with accuracies?
Do you include a license condition? Or is it "only in the company"
Do you tag personal information for faster GDPR responses?

DO YOUR PARTNERS include this data?

Source: http://aura.abdn.ac.uk/bitstream/handle/2164/6051/baillie_acm_jdiq_15.pdf?sequence=1
NOT ALONE!
Collaborations are needed by ETSI ISG CIM

User Apps

Provenance

IoT

oneM2M

W3C WoT

GSMA

(5G)

A.I.

Context Information Management

ISG CIM API [NGSI-LD]

SAREF

Context Information Ontologies

Context Information Ontologies

Context Information Ontologies

+ EU H2020 Projects

Open Data

Usage

Proprietary Data

BDVA

ITU-T FG DPM

INSPIRE

FIWARE

© ETSI 2019
NGSI-LD CURRENT STATUS
Conclusions re Information Management

So many Smart City services ....

So many issues in enabling exchange of meaningful, usable information

Do not wait for perfection ...
Get started, collaborate, standardize, improve
Thank You!

Contact for ETSI ISG CIM:
ISGSupport@etsi.org

Chairman:
Lindsay Frost (NEC)

Vice-chairman:
Christophe Collinet (EuroCities)

Open pages for consensus material:
https://docbox.etsi.org/ISG/CIM/Open
+ visit at: https://portal.etsi.org/CIM