



ACCELERATE YOUR IOT ADOPTION

F4W DEVELOPERS' WORKSHOP

www.eglobalmark.com



PLAN



What is the question again ?

As a data producer : where do I push my data ?

As an app developer: where do I consume available information?

As a use case provider: what is the added value of the F4W approach compared to what I have in house ?

As a F4W project: where is all of this deployed ? By use cases or centrally ?

INTRODUCTION

FIWARE vs NGSI-LD

FIWARE: a foundation emerged from an EU program

- Animating a community of developers & users (800+ SMEs, ...)
- Maintaining a catalog of Open Source generic enablers to create end-to-end solutions
- Developing data models in cooperation with TMForum
- Contributing to the development of the ETSI NGSI-LD specification it builds upon

ETSI NGSI-LD: a specification built within ETSI standardisation body

- A RESTful API to handle context information
- A cross-domain data model to exchange context information across domains
- Retro-compatibility with former NGSIv2 (still maintained in FIWARE, not considered in F4W)

RESOURCES

FIWARE ecosystem: open-source enablers, community

www.fiware.org

Standards

There are 3 results

☐ Download All ☐ Export List

Sort By

Relevant

☐ ETSI GS CIM 009 V1.2.2 (2020-02)

Published



Context Information Management (CIM); NGSI-LD API

An update is in preparation. [DETAILS](#) [ALERT](#)

☐ ETSI GS CIM 009 V1.2.1 (2019-10)

Published



Context Information Management (CIM); NGSI-LD API

An update is in preparation. [DETAILS](#) [ALERT](#)

☐ ETSI GS CIM 009 V1.1.1 (2019-01)

Published



Context Information Management (CIM); NGSI-LD API

An update is in preparation. [DETAILS](#) [ALERT](#)

The screenshot displays the FIWARE Catalogue interface. At the top, there's a navigation bar with 'Home', 'Enablers', 'Bundles', and 'Tools'. The main content is divided into three sections: 'FIWARE GENERIC ENABLERS', 'DOMAIN SPECIFIC ENABLERS', and 'BUNDLES'. The 'Generic Enablers' section lists six categories: Data/Context, Internet of Things, Advanced UI, Security, I2ND, and APPS, each with a brief description. The 'Domain Specific Enablers' section lists five categories: Manufacturing, Media, eHealth, Energy, and Agrifood, each with a brief description. The 'Bundles' section lists two categories: Business Framework Consumption and Data Context Streams, each with a brief description.

ETSI ISG CIM

(NGSI-LD standardisation)

Mapping NGSI-LD API
to FIWARE



API & MODEL



REQUIREMENTS



Handle data heterogeneity

- Data sources: IoT, open data, proprietary, document, ...
- Data format: within and across domains

Handle legacy: do not replace but connect

Allow flexibility

- Deploy new services/use cases/apps when needed

Handle scalability and data protection

- Distributed/federated approaches

Ease adoption

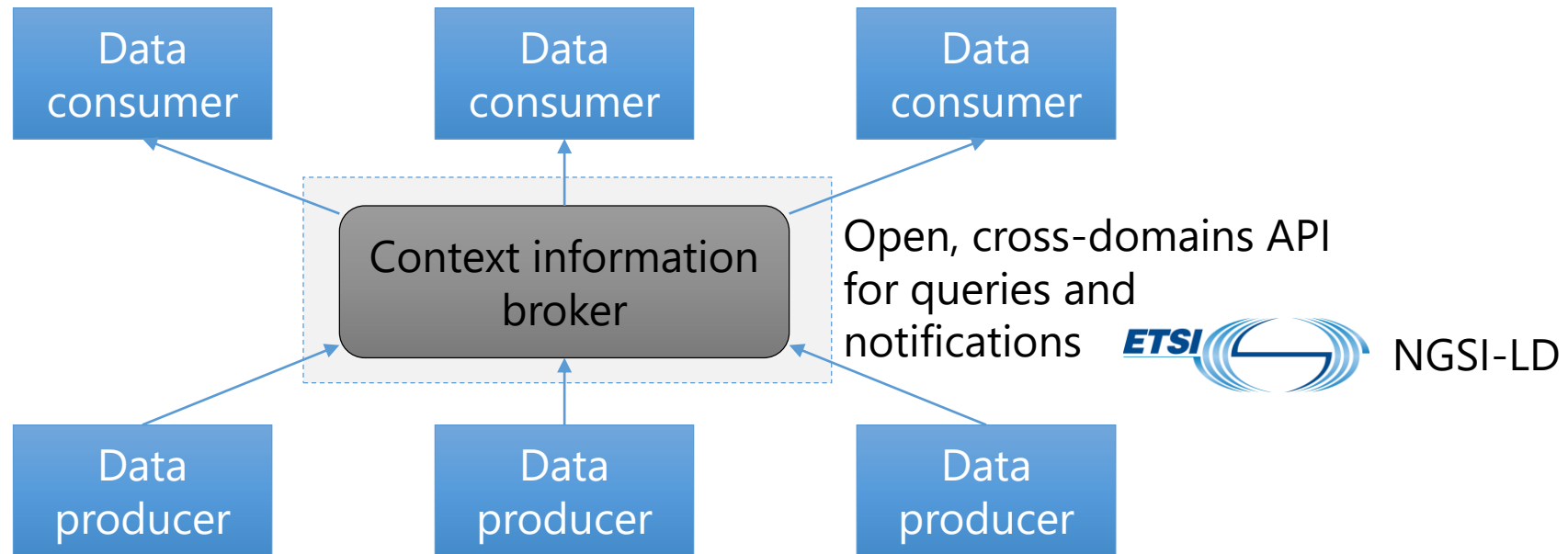
- Developers' friendliness

Allow batch & stream processing, ML, IA

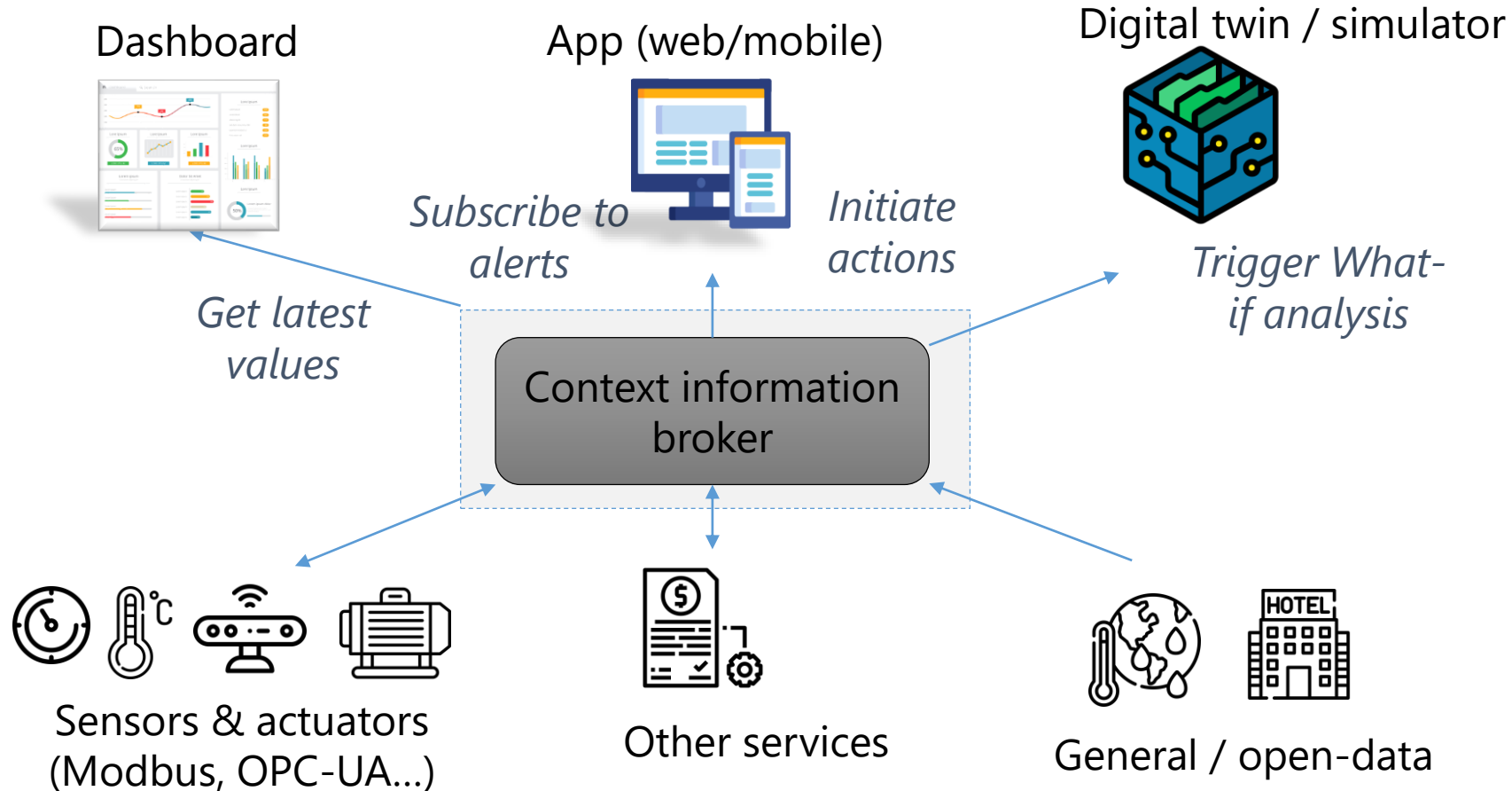
ALL IS ABOUT CONTEXT INFORMATION EXCHANGE



Data contextualised in time, space and relations to other data

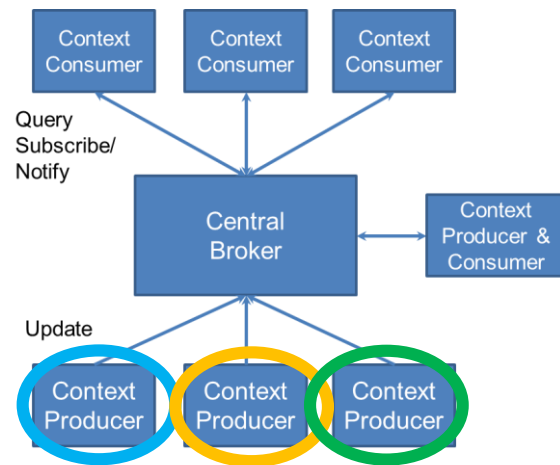
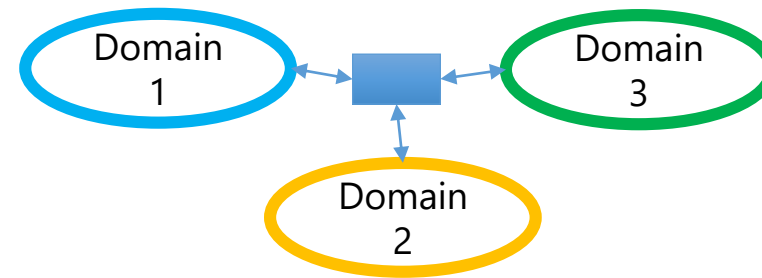


USAGE EXAMPLE

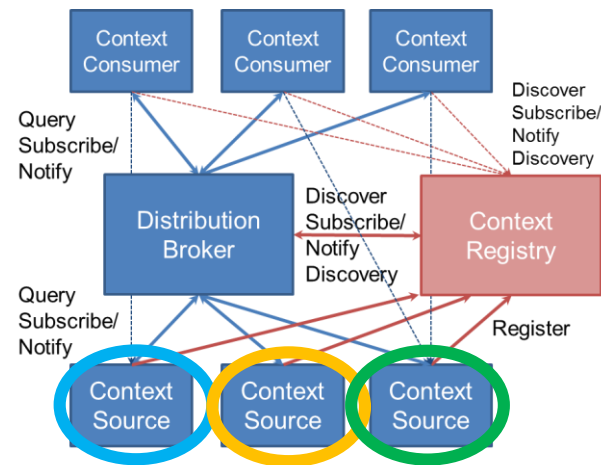


"Icons made by [Freepik](#), [phatplus](#), [wichai.wi](#) from www.flaticon.com"

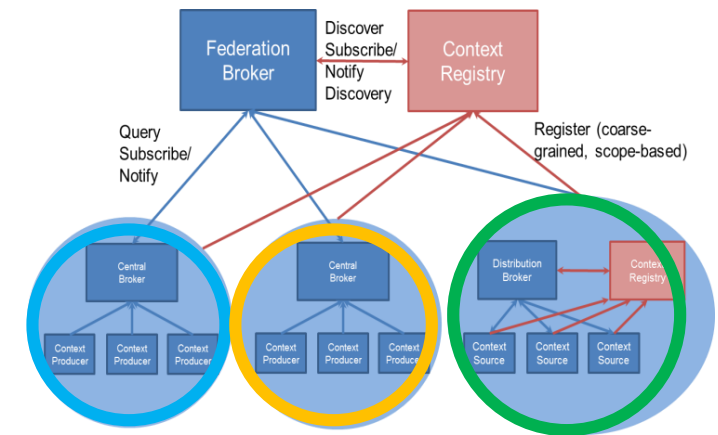
NGSI-LD DEPLOYMENT ARCHITECTURES



Centralised



Distributed



Federated

Pictures courtesy of ETSI

ETSI ISG CIM 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 of 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 geographic scope, constrained not to traverse graph (only one level at a time)

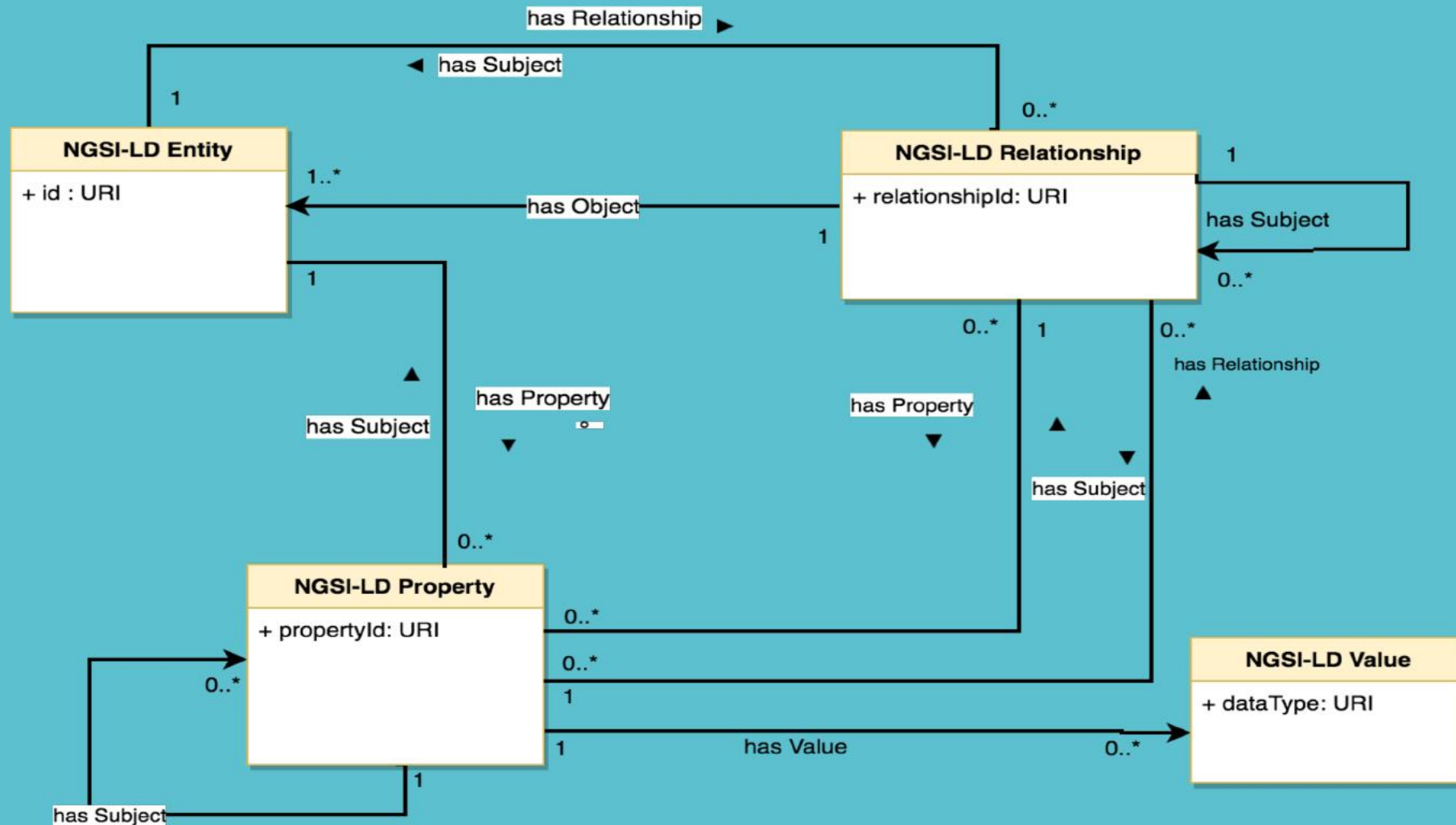
Information Model - Highlights

- **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**
 - An NGSI value can be a single value (Number, String, boolean), or complex (Array, Structured Value)
- Relationships have an **object**
 - A URI which points to another entity (target of the relationship). Target can be a collection.

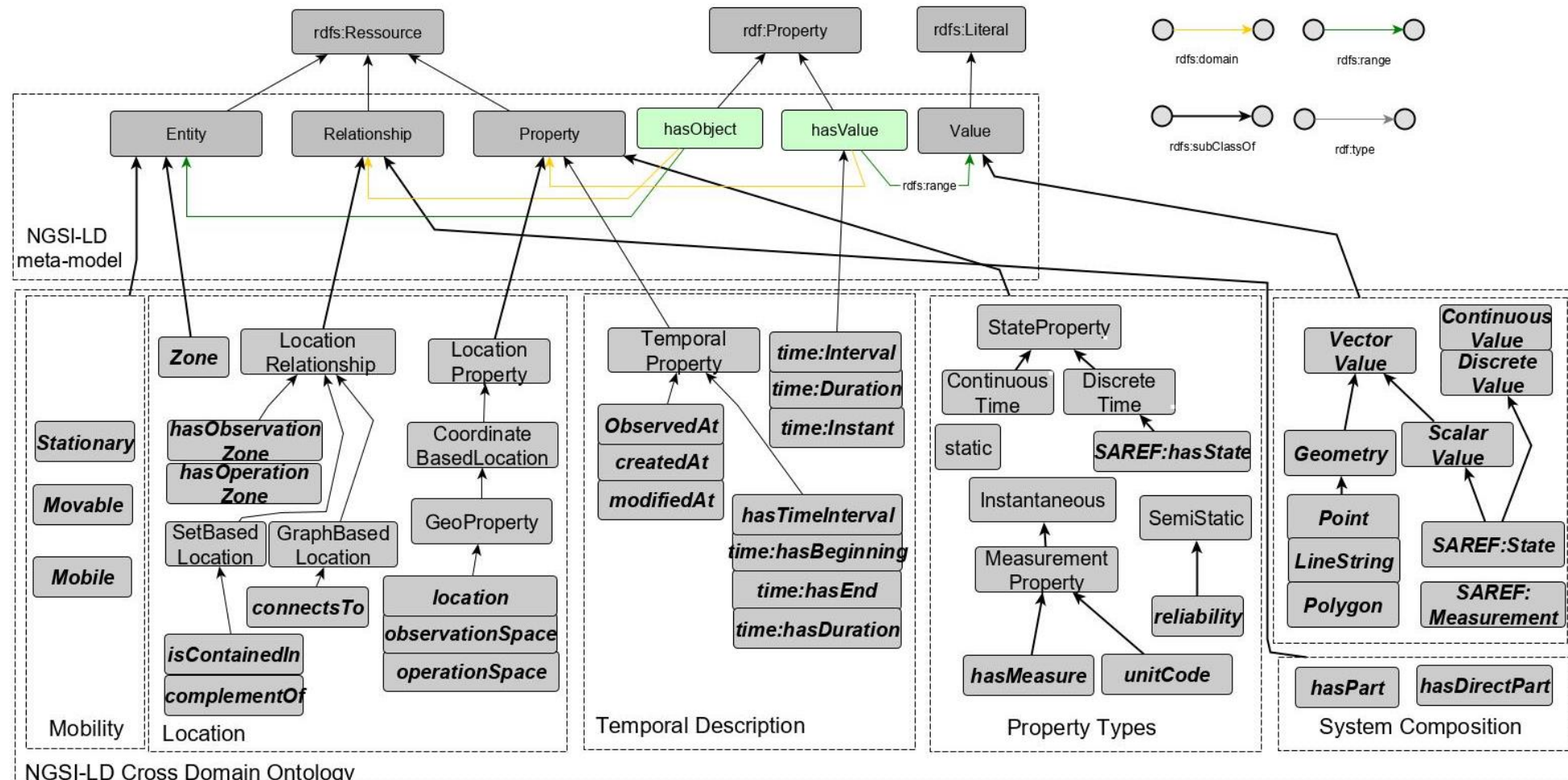
Information Model – Highlights (II)

- 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 *NGSlv2*
 - **modifiedAt** → Update timestamp (of entity, attribute). *dateModified* in *NGSlv2*
 - **unitCode** → Units of measurement, encoded as mandated by **UN/CEFACT**.
- Recommended practice
 - Use URIs to identify your entities.
 - A URN schema is provided off-the-shelf. It enables to know in advance what entity type an id refers to
 - urn:ngsi-Id:<Entity_Type_Name>:<Entity_Identification_String>

Information Model (as UML) – NGSI LD



CROSS DOMAIN + INFORMATION MODEL



JSON-LD (RDF friendly) representation (a.k.a. **NGSI LD**)

```
{
  "id": "urn:ngsi-ld:Vehicle:A4567",
  "type": "Vehicle",
  "brandName": {
    "type": "Property",
    "value": "Mercedes"
  },
  "isParked": {
    "type": "Relationship",
    "object": "urn:ngsi-ld:OffStreetParking:Downtown1",
    "observedAt": "2017-07-29T12:00:04Z",
    "providedBy": {
      "type": "Relationship",
      "object": "urn:ngsi-ld:Person:Bob"
    }
  },
  "@context": [
    "http://uri.etsi.org/ngsi-ld/coreContext.jsonld",
    "http://example.org/cim/commonTerms.jsonld",
    "http://example.org/cim/vehicle.jsonld",
    "http://example.org/cim/parking.jsonld"
  ]
}
```



NGSI LD

```
{
  "id": "urn:ngsi-ld:OffStreetParking:Downtown1",
  "type": "OffStreetParking",
  "availableSpotNumber": {
    "type": "Property",
    "value": 121,
    "observedAt": "2017-07-29T12:05:02Z",
    "reliability": {
      "type": "Property",
      "value": 0.7
    }
  },
  "providedBy": {
    "type": "Relationship",
    "object": "urn:ngsi-ld:Camera:C1"
  },
  "location": {
    "type": "GeoProperty",
    "value": {
      "type": "Point",
      "coordinates": [-8.5, 41.2]
    }
  },
  "@context": [
    "http://uri.etsi.org/ngsi-ld/coreContext.jsonld",
    "http://example.org/cim/parking.jsonld"
  ]
}
```

Simplified representation (keyValues)

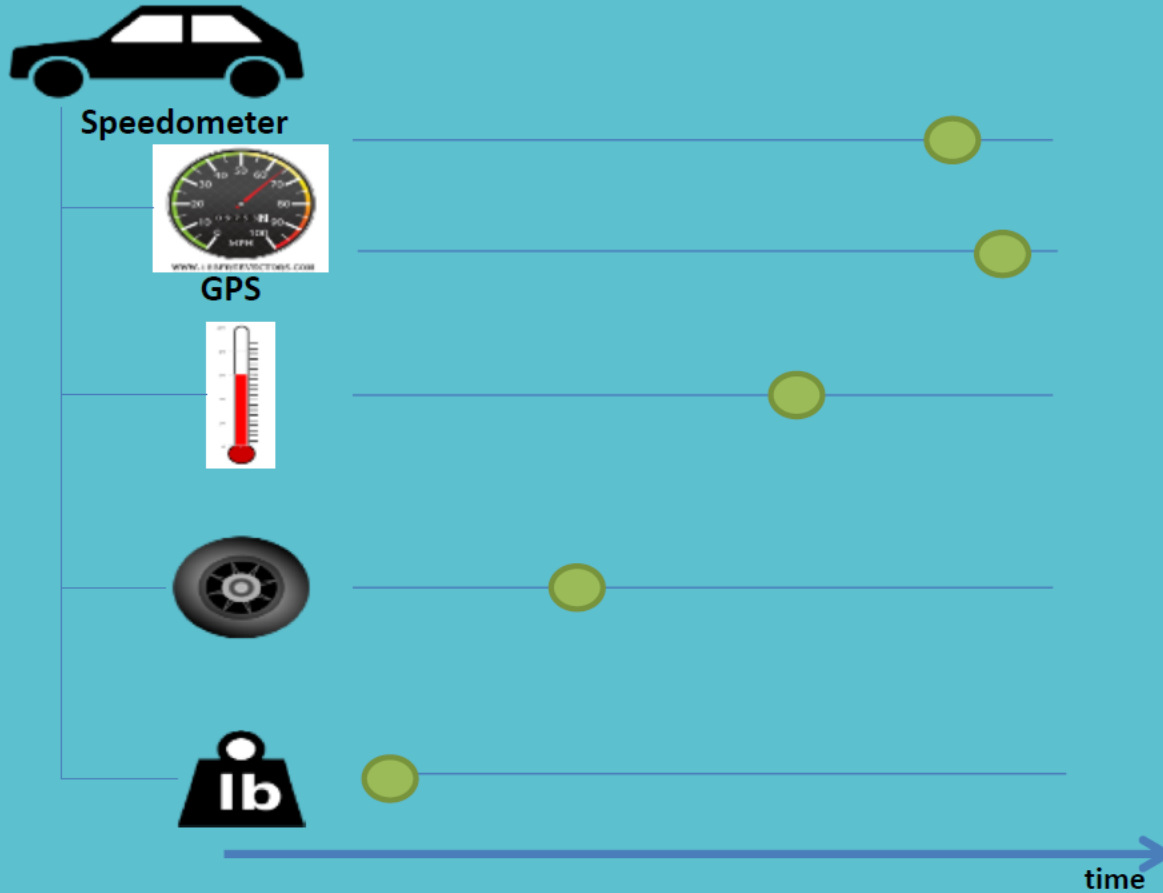
```
{
  "id": "urn:ngsi-ld:OffStreetParking:Downtown1",
  "type": "OffStreetParking",
  "name": "Downtown One",
  "availableSpotNumber": 121,
  "totalSpotNumber": 200,
  "location": {
    "type": "Point",
    "coordinates": [-8.5, 41.2]
  },
  "@context": [
    "http://uri.etsi.org/ngsi-ld/coreContext.jsonld",
    "http://example.org/cim/parking.jsonld"
  ]
}
```



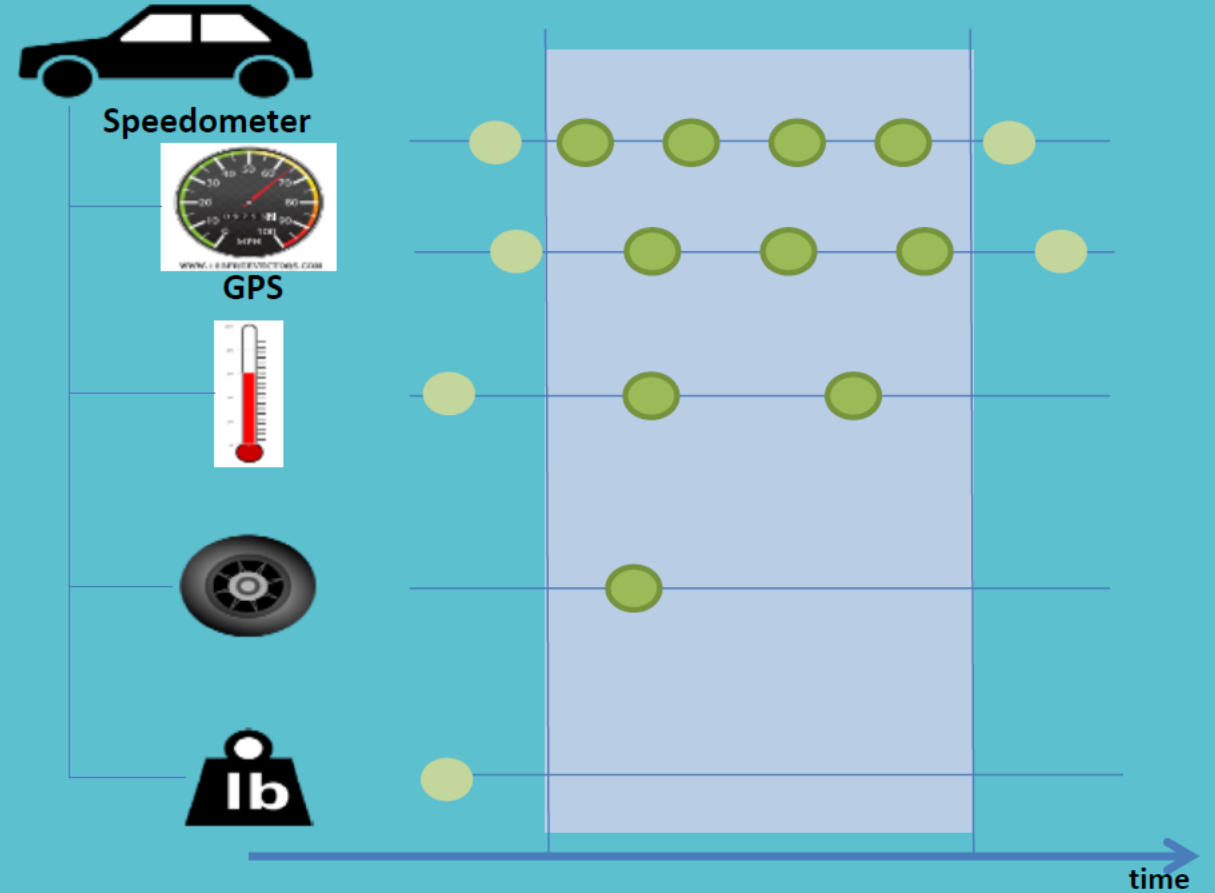
Equivalent in NGSI-LD and NGSIv2



Temporal Queries (e.g. for trend analysis)

Regular Query
"most recent"



Temporal Query



-  included in result set
-  not included in result set

NGSI-LD API = Queries & Notifications

- Periodic
- On change
- On event
 - > Query based
 - > Geoproperties (near, within, ...)

The grammar that defines the query language in ABNF format [12] is described below (it has been validated using <https://tools.ietf.org/tools/bap/abnf.cgi>) and shall be supported by implementations:

```
Query = (QueryTerm / QueryTermAssoc) *(logicalOp (QueryTerm / QueryTermAssoc))
QueryTermAssoc = %x28 QueryTerm *(logicalOp QueryTerm) %x29 ; (QueryTerm)
QueryTerm = Attribute
QueryTerm = Attribute Operator ComparableValue
QueryTerm = / Attribute equal CompEqualityValue
QueryTerm = / Attribute unequal CompEqualityValue
QueryTerm = / Attribute patternOp RegExp
QueryTerm = / Attribute notPatternOp RegExp
Attribute = attrName / compoundAttrName / attrPathName
Operator = equal / unequal / greaterEq / greater / lessEq / less
ComparableValue = Number / quotedStr / dateTime / date / time
OtherValue = false / true
Value = ComparableValue / OtherValue
Range = ComparableValue dots ComparableValue
ValueList = Value 1*(%x2C Value) ; Value 1*(, Value)
CompEqualityValue = OtherValue / ValueList / Range / URI
equal = %x3D %x3D ; ==
unequal = %x21 %x3D ; !=
greater = %x3E ; >
greaterEq = %x3E %x3D ; >=
less = %x3C ; <
lessEq = %x3C %x3D ; <=
patternOp = %x7E %x3D ; ~
notPatternOp = %x21 %x7E %x3D ; !=
dots = %x2E %x2E ; ..
attrNameChar = / DIGIT / ALPHA
attrNameChar = / %x5F ; _
attrName = 1*attrNameChar ; attrName *(. attrName)
compoundAttrName = attrName *(%x5B (attrName) %x5D) ; . attrName *([ attrName ])
quotedStr = String ; 'char'
andOp = %x3B ; &
orOp = %x7C ; |
logicalOp = andOp / orOp
```

See <https://fiware-tutorials.readthedocs.io/en/latest/subscriptions/index.html>



NGSI-LD API OPERATIONS

General Operations

- Entity create
- Entity update
- Entity partial update
- Entity delete
- Entity retrieval
- Queries
- Subscriptions

Registry Operations

- CSRegistryEntry create
- CSRegistryEntry update
- CSRegistryEntry partial update
- CSRegistryEntry delete
- CSRegistryEntry retrieval
- CSRegistryEntry query
- CSRegistryEntry subscription

Batch Operations

- Batch Entity Creation
- Batch Entity Create/Update (Upsert)
- Batch Entity Update
- Batch Entity Delete

Temporal Operations

- Create/Update Temporal Entity Representation
- Add Attributes to Temporal Entity Rep.
- Delete Attribute from Temporal Entity Rep.
- Modify Attribute Instance in Temporal Entity Rep.
- Delete Attribute Instance from Temporal Entity Rep.
- Delete Temporal Entity Representation
- Retrieve Temporal Entity Evolution
- Query Temporal Entity Evolution

Query language on Attribute values

- Value of a Property
- Object of a relationship

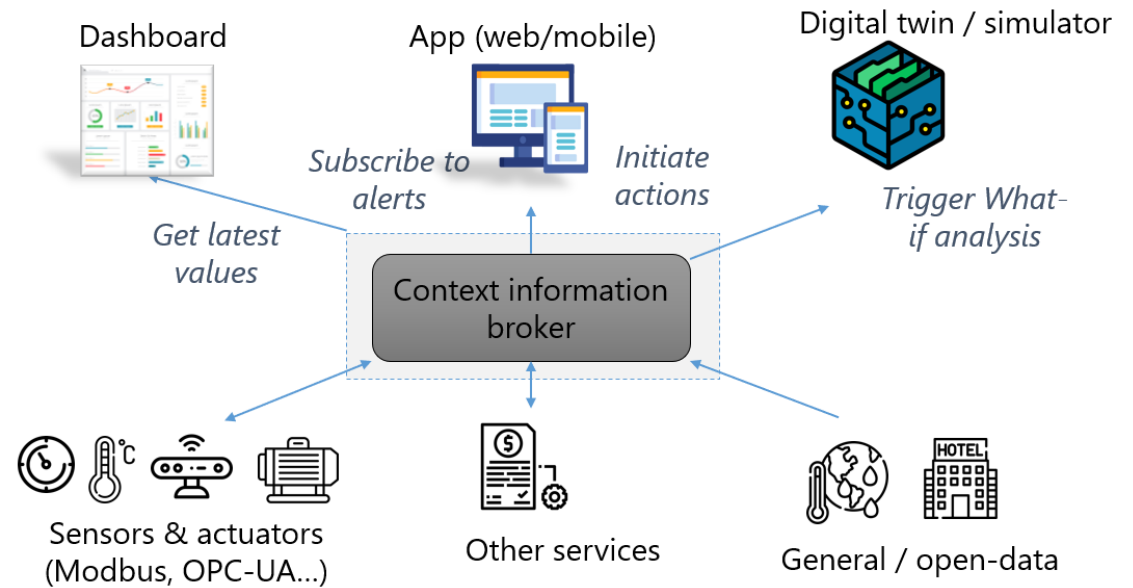
Temporal queries datatype

- time (absolute timestamp)
- timerel (before, after, between)
- endTime
- timeProperty (name< of property handling temporal information)

Geoqueries

- Represented through GeoJSON geometries. Basic GeoProperties defined:
 - Location
 - observationSpace
 - OperationSpace
- Advanced queries
 - Near
 - Equal
 - Disjoint
 - Intersect
 - Within
 - Contains
 - Overlaps<

HOW TO START



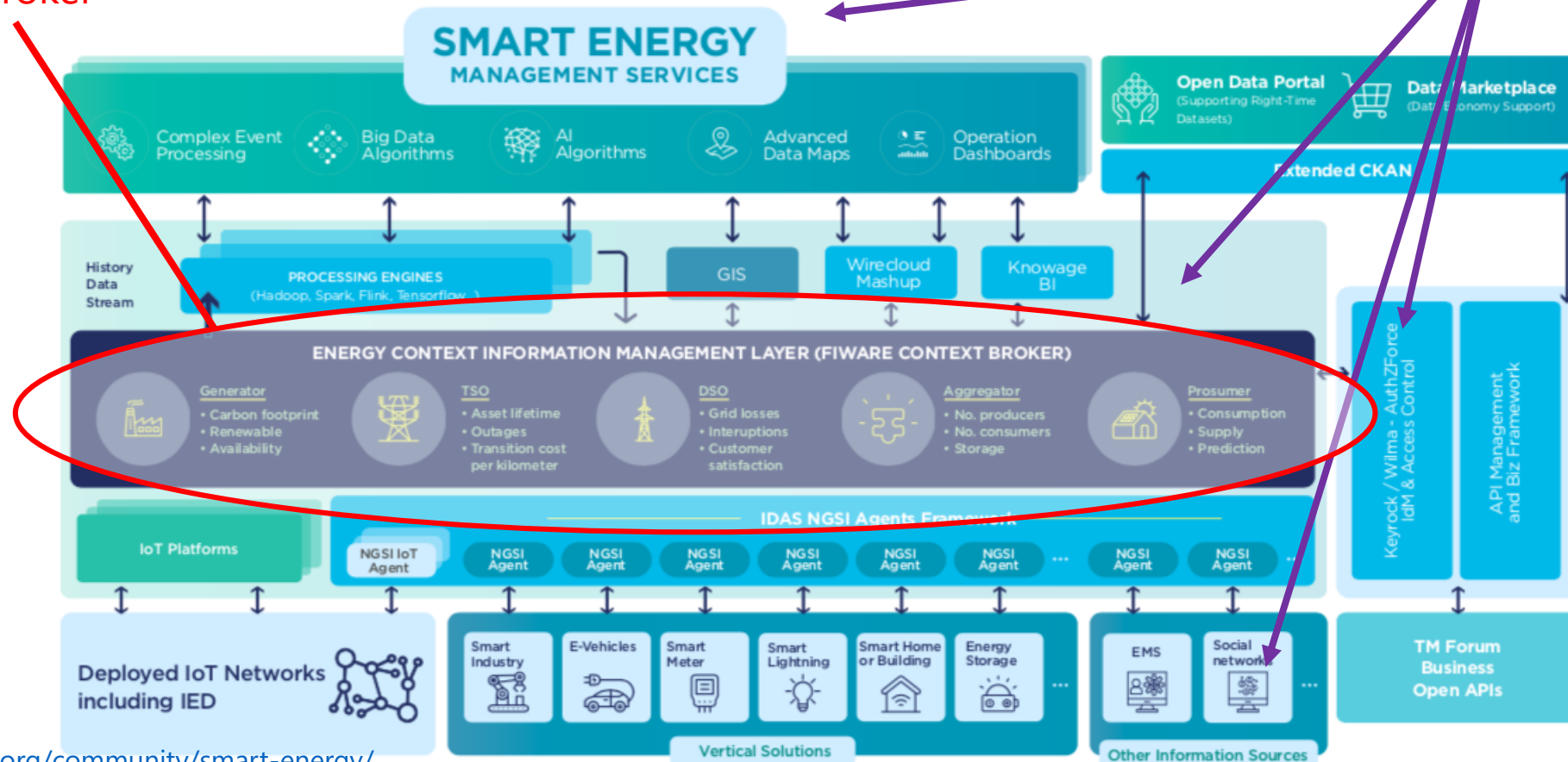
OVERALL FIWARE “ARCHITECTURE”

Smart Energy inspiration

Core component: the
NGSI-LD broker

Enablers

- Fiware catalog
- Specific (i.e. epanet)



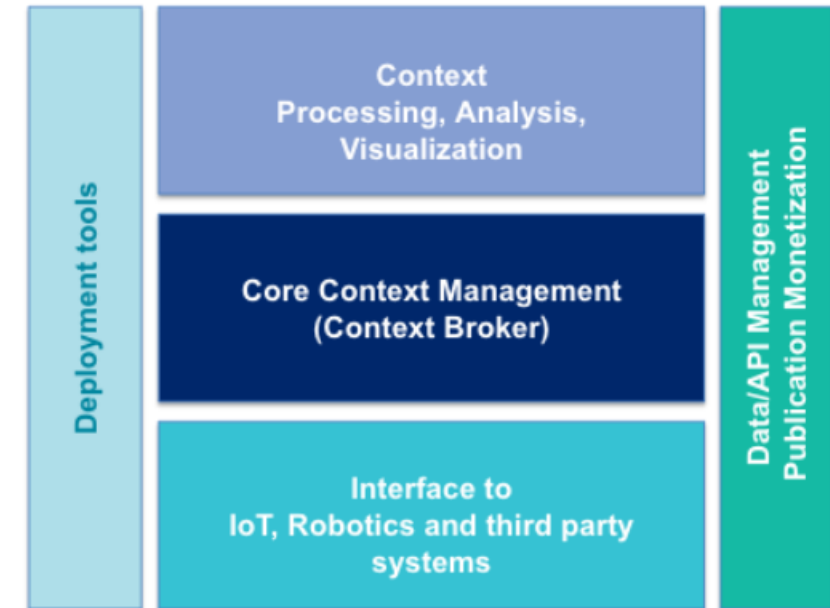
<https://www.fiware.org/community/smart-energy/>

The FIWARE Catalogue is a curated framework of **open source platform components** which can be assembled together and with other third-party platform components to accelerate the development of **Smart Solutions**.

The main and only mandatory component of any “**Powered by FIWARE**” platform or solution is the **FIWARE Orion Context Broker Generic Enabler**, which brings a cornerstone function in any smart solution: the need to manage context information, enabling to perform updates and bring access to context.

Building around the **FIWARE Context Broker**, a rich suite of complementary FIWARE components are available, dealing with:

- **Interfacing with the Internet of Things (IoT), Robots and third-party systems**, for capturing updates on context information and translating required actuations.
- **Context Data/API management, publication, and monetization**, bringing support to usage control and the opportunity to publish and monetize part of managed context data.
- **Processing, analysis, and visualization of context information** implementing the expected smart behavior of applications and/or assisting end users in making smart decisions.



Migration NGSIv2 -> NGSI-LD ongoing

<https://www.fiware.org/developers/catalogue/>

1ST STEP: CHOOSE A NGSI-LD CONTEXT BROKER



Options

1. **Implement your own**
2. **Deploy one available off the shelves**
 - > ORION-LD (
 - > SCORPIO
 - > STELLIO (EGM)
 - > ...
3. **Consume a end-point made available by a partner/project**

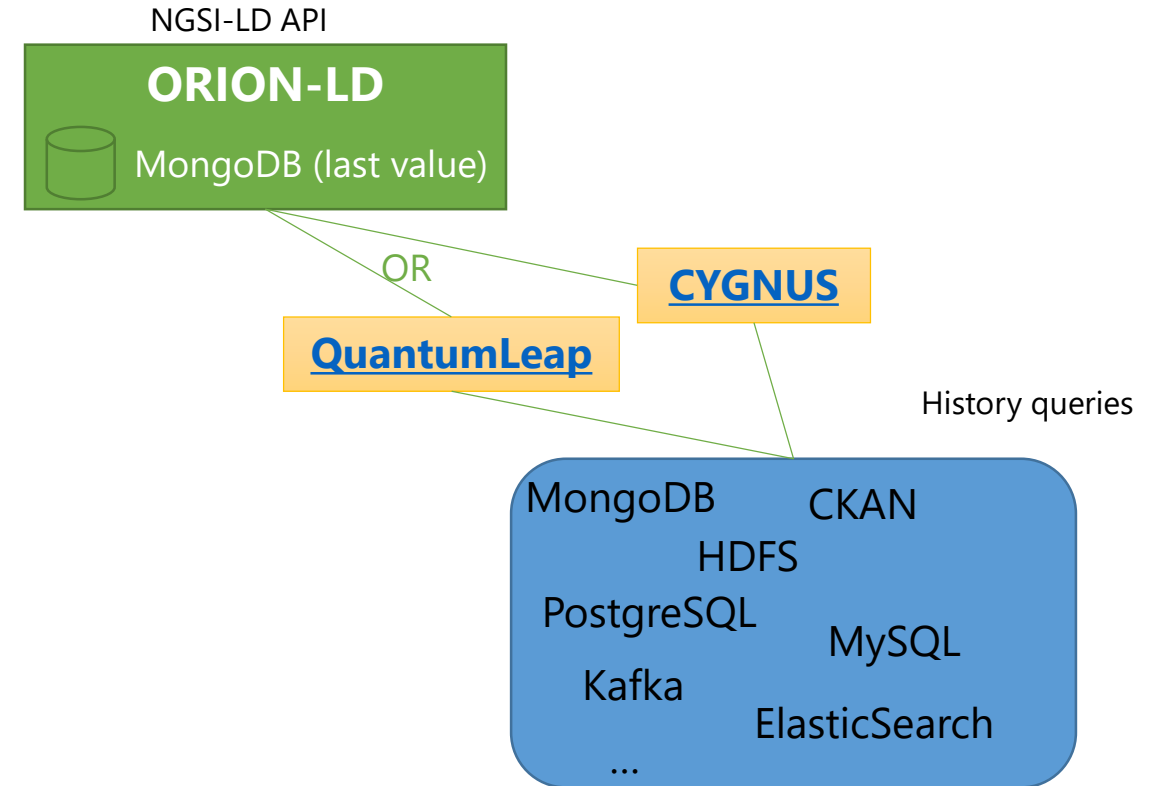
Note: NGSI-LD specification:

- **Allows for time-series requests**
- **Does not impose anything on storing the history**

NGSI-LD BROKERS DEPLOYMENT STRATEGIES

ORION-LD

- By default: keep only last value
- With additional enablers
 - > Store short term history
 - > Store long term history



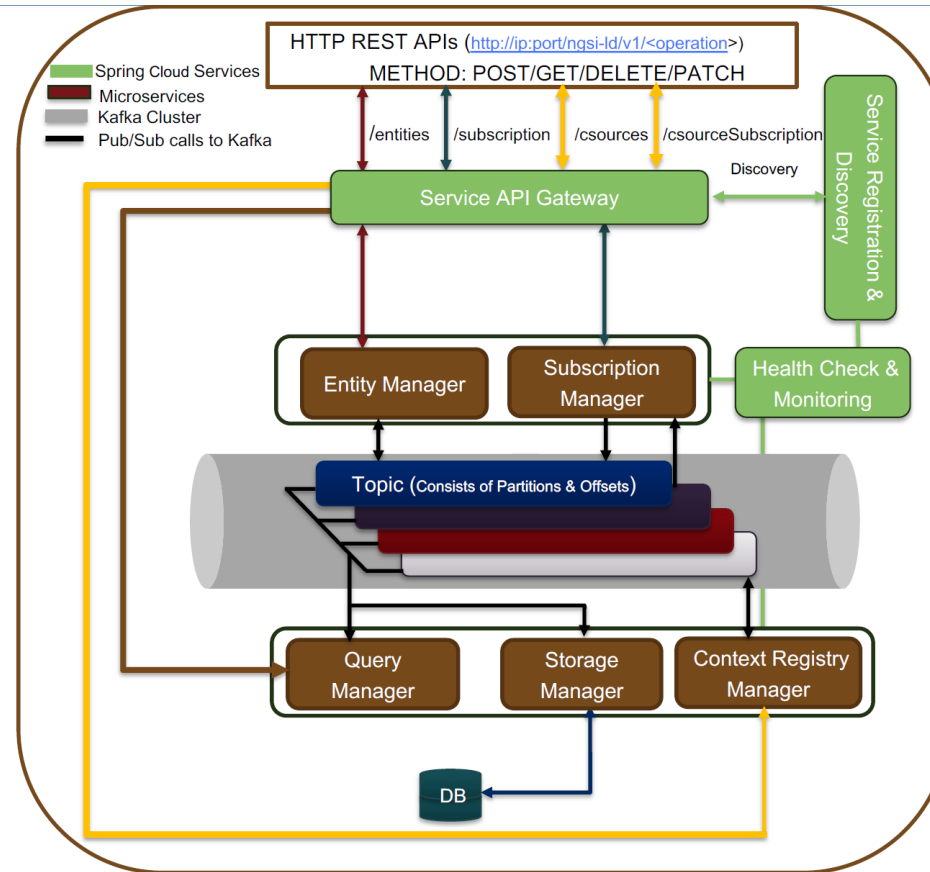
<https://fiware-academy.readthedocs.io/en/latest/core/orion-ld/index.html>

<https://github.com/FIWARE/context.Orion-LD>

NGSI-LD BROKERS DEPLOYMENT STRATEGIES

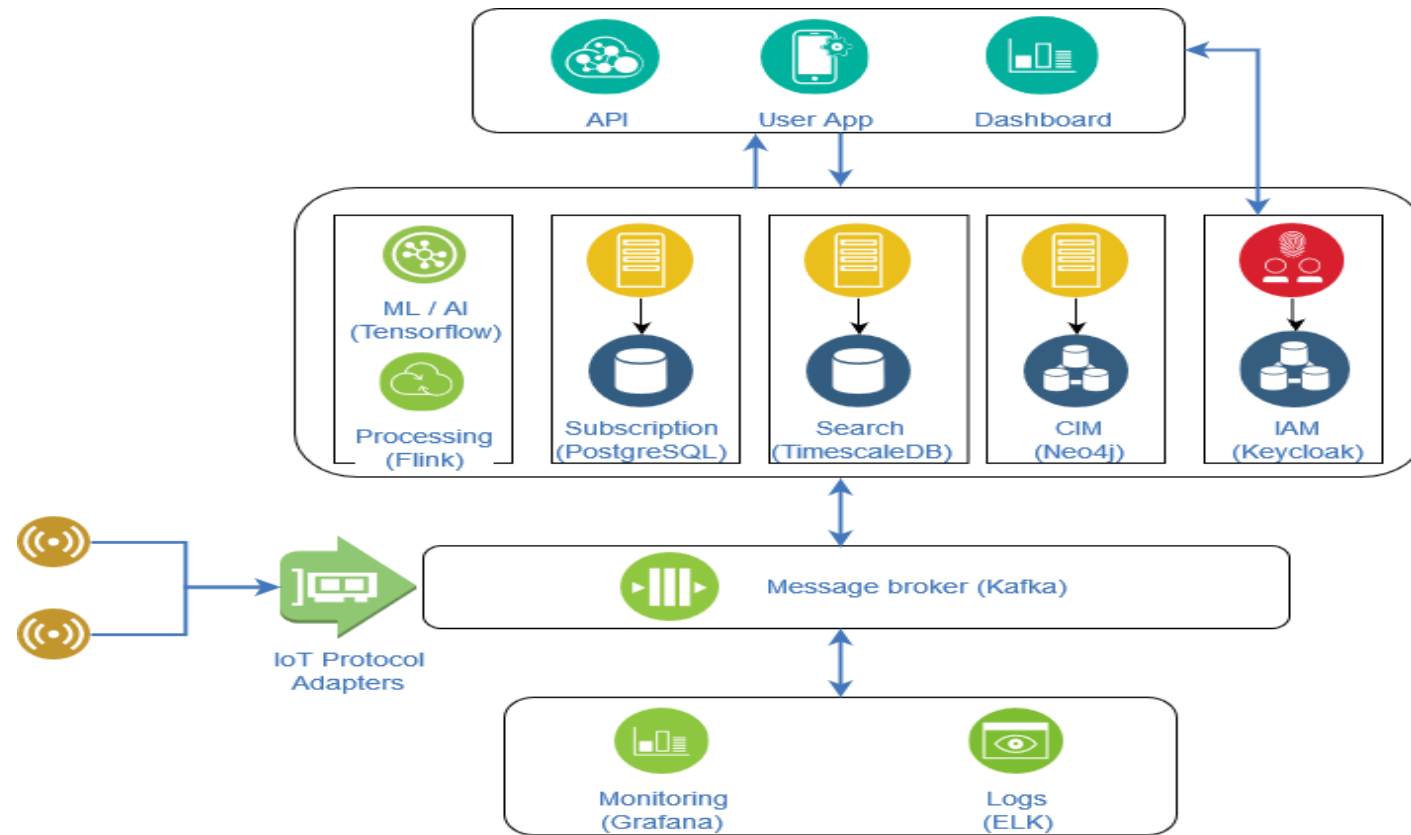


SCORPIO



NGSI-LD BROKERS DEPLOYMENT STRATEGIES

STELLIO



Endpoint available

Plan for public release before June (OS)

INTERACTING WITH NGSI-LD API



Demo + Q&A

CONCLUSION

Did we answer some questions

As a data producer : where do I push my data ?

As an app developer: where do I consume available information?

As a use case provider: what is the added value of the F4W approach compared to what I have in house ?

As a F4W project: where is all of this deployed ? By use cases or centrally ?



www.eglobalmark.com

THANK YOU



Franck Le Gall

COO

Tel: +33.6.20.03.54.20

E.mail: Franck.le-gall@eglobalmark.com

