Managing Context Information at large scale (Advanced Topics)

OPEN APIs FOR OPEN MINDS

FIWARE Lab — Spark your imagination

FIWARE Ops — Easing your operations

www.fiware.org
@Fiware

Contact twitter
@fermingalan

Contact email
fermin.galanmarquez@telefonica.com
kengunnar.zangelin@telefonica.com

(Reference Orion Context Broker version: 1.3.0)

Advanced Features

Orion Context Broker

- Pagination
- Metadata
- Compound attribute/metadata values
- Type browsing
- Geo-location
- Query filters
- Registrations & context providers
- Multitenancy
- Service paths
- CORS
- Notifying services in private networks

Pagination

- Pagination helps clients organize query and discovery requests with a large number of responses.

- Three URI parameters:
  - limit
    - Number of elements per page (default: 20, max: 1000)
  - offset
    - Number of elements to skip (default: 0)
  - count (option)
    - Returns total elements (default: not return)

Pagination

- Example, querying the first 100 entries:
  - GET <orion_host>:1026/v2/entities?limit=100&options=count

- The first 100 elements are returned, along with the following header in the response:
  - Fiware-Total-Count: 322

- Now we now there are 322 entities, we can keep querying the broker for them:
  - GET <orion_host>:1026/v2/entities?offset=100&limit=100
  - GET <orion_host>:1026/v2/entities?offset=200&limit=100
  - GET <orion_host>:1026/v2/entities?offset=300&limit=100

Pagination

- By default, results are ordered by entity creation date
- This behavior can be overridden using **orderBy** URI parameter
  - A comma-separated list of attributes. Results are ordered by the first attribute. On ties, the results are ordered by the second attribute and so on. A "!" before the attribute name means that the order is reversed.
- Example: get the first 10 entities ordered by temp in ascending order, then humidity in descending order

  GET <orion_host>:1026/v2/entities?limit=20&offset=0&orderBy=temp,!humidity

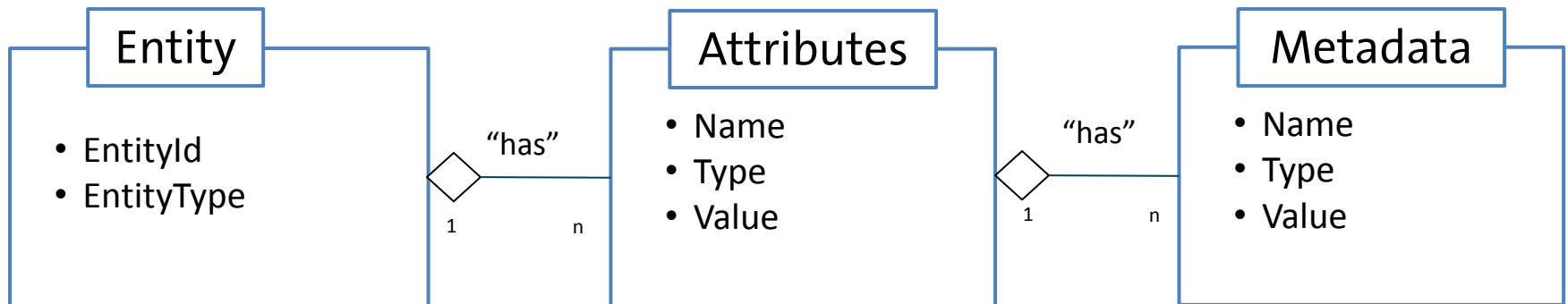- **dateCreated** and **dateModified** can be used to ordering by entity creation and modification date, respectively

FIWARE   FIWARE Lab   FIWARE Ops

Metadata

- Users may attach metadata to attributes
- Reserved metadata: ID, Location
- Examples:

```
…
"temperature": {
    "type": "Float",
    "value": 26.5,
    "metadata": {
    {
     "accuracy": {
      "type": "Float",
      "value": 0.9
     }
    }
}
…
```

```
…
"temperature": {
    "type": "Float",
    "value": 26.5,
    "metadata": {
    {
     "average": {
      "type": "Float",
      "value": 22.4
     }
    }
}
…
```

Complete NGSI Model

| Entity		Attributes		Metadata
• EntityId • EntityType	"has" 1 n	• Name • Type • Value	"has" 1 n	• Name • Type • Value

Compound Attribute/Metadata Values

- Attributes and metadata can have a structured value. Vectors and key-value maps are supported.

- It maps directly to JSON's objects and arrays.

Compound Attribute/Metadata Values

- Example: we have a car whose four wheels' pressure we want to represent as a compound attribute for a car entity. We would create the car entity like this:

```
{
    "type": "Car",
    "id": "Car1",
    "tirePressure": {
        "type": "kPa",
        "value": {
            "frontRight": "120",
            "frontLeft": "110",
            "backRight": "115",
            "backLeft": "130"
        }
    }
}
```
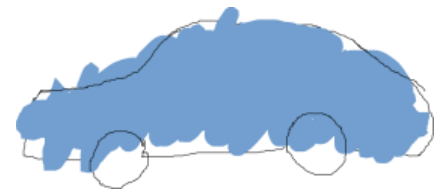
Type Browsing

- GET /v2/types
  - Retrieve a list of all entity types currently in Orion, including their corresponding attributes and entities count
- GET /v2/types/{typeID}
  - Retrieve attributes and entities count associated to an entity type

PRO TIP

GET /v2/contextTypes?options=values
Retrieves just a list of all entity types without any extra info

Geo-location

- Entities can have an attribute that specifies its location
- Several attribute types can be used
  - geo:point (for points)
  - geo:line (for lines)
  - geo:box (for boxes)
  - geo:polygon (for polygons)
  - geo:json (for arbitrary geometries, in GeoJson standard)

- Example: create an entity called Madrid

  …and create a couple more towns:
  - Leganés
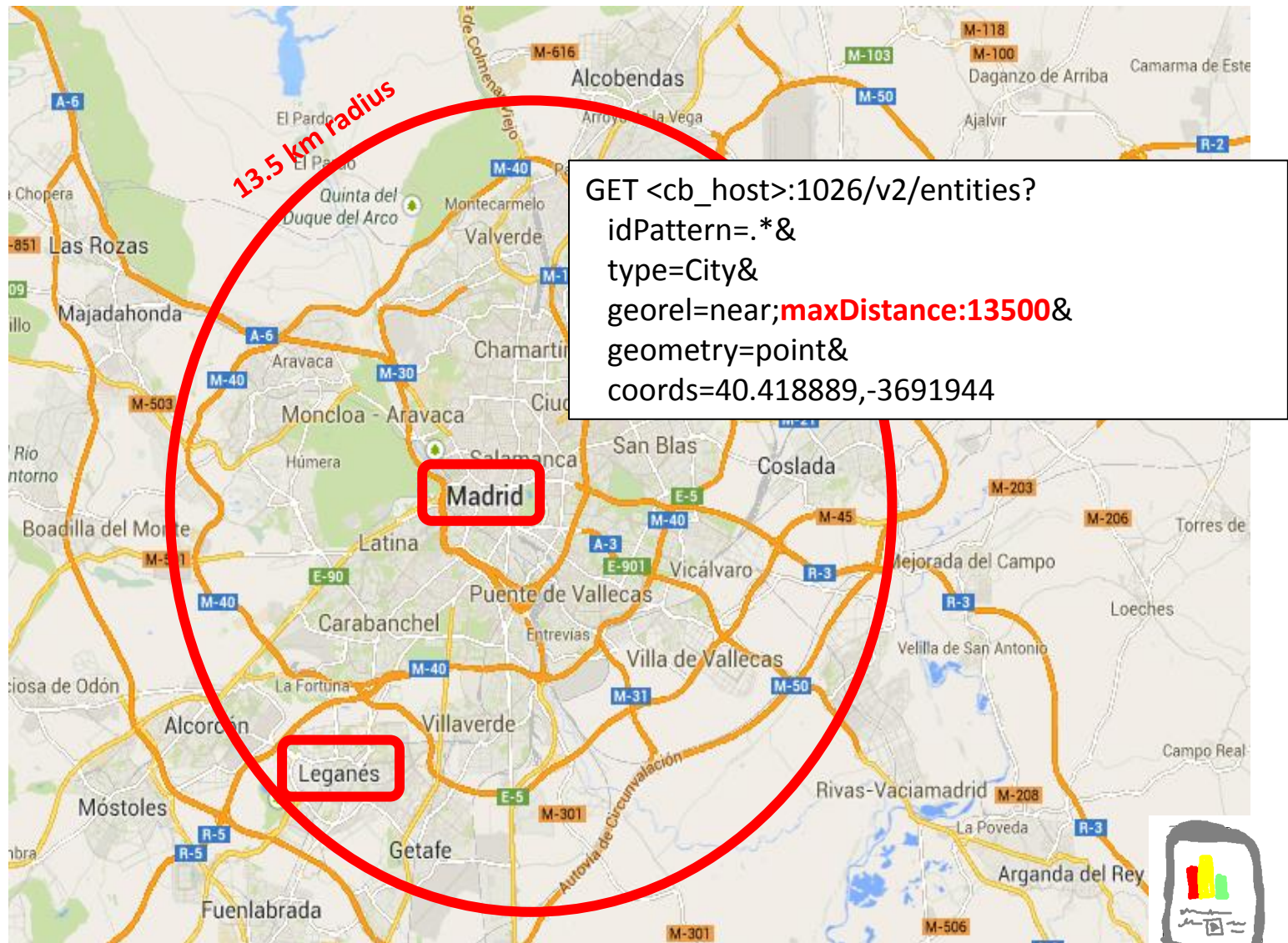 - Alcobendas

```
POST <cb_host>:1026/v2/entities
{
  "type": "City",
  "id": "Madrid",
  "position": {
    "type": "geo:point",
    "value": "40.418889, -3.691944"
  }
}
```
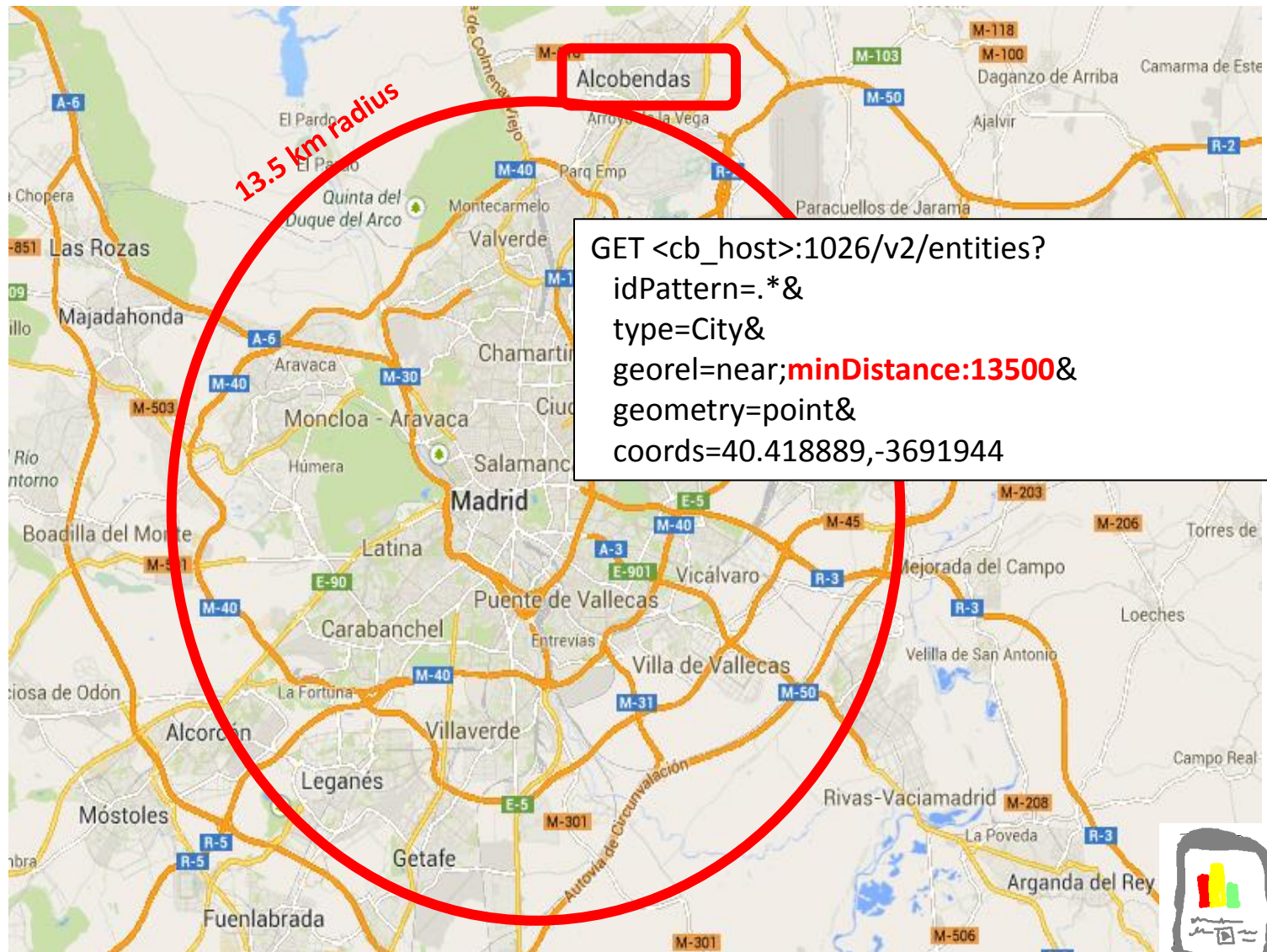
Geo-location – Circle
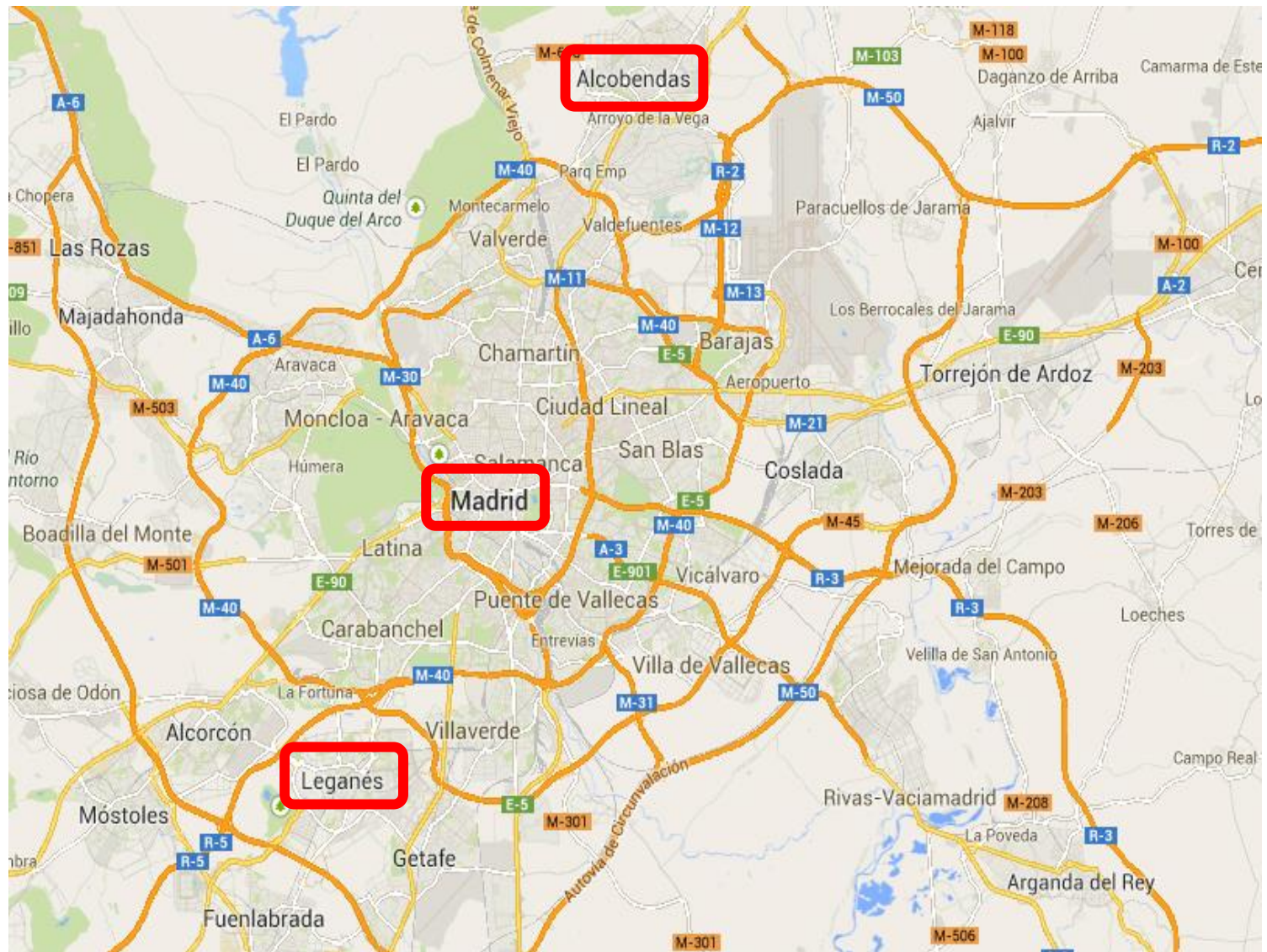
Geo-location – Max distance

13.5 km radius

```
GET <cb_host>:1026/v2/entities?
  idPattern=.*&
  type=City&
  georel=near;maxDistance:13500&
  geometry=point&
  coords=40.418889,-3691944
```

Geo-location – Min distance

13.5 km radius

```
GET <cb_host>:1026/v2/entities?
    idPattern=.*&
    type=City&
    georel=near;minDistance:13500&
    geometry=point&
    coords=40.418889,-3691944
```

More geo-relationships

- Apart from **near**, the following georel can be used
  - georel=coveredBy
  - georel=intersects
  - georel=equals
  - georel=disjoint
- See NGSIv2 Specification for a detailed description

Query filters

- For the **GET /v2/entities** operation

- By **entity type**   `GET <cb_host>:1026/v2/entities?`**type=Room**

- By **entity id list**   `GET <cb_host>:1026/v2/entities?`**id=Room1,Room2**

- By **entity id pattern** (regex)   `GET <cb_host>:1026/v2/entities?`**idPattern=^Room[2-5]**

- By **entity type pattern** (regex)   `GET <cb_host>:1026/v2/entities?`**typePattern=T[ABC]**

- By **geographical location**
  - Described in detail in previous slides

Query filters

- By **attribute value** (q)

  attribute name

  ```
  GET <cb_host>:1026/v2/entities?q=temperature>25
  ```

  attribute sub-key (for compound attribute values only)

  ```
  GET <cb_host>:1026/v2/entities?q=tirePressure.frontRight >130
  ```

- By **metadata value** (mq)

  attribute name

  metadata name

  ```
  GET <cb_host>:1026/v2/entities?q=temperature.avg>25
  ```

  metadata sub-key (for compound metadata values only)

  ```
  GET <cb_host>:1026/v2/entities?q=tirePressure.accuracy.frontRight >90
  ```

- See full details about **q** and **mq** query language in NGSIv2 specification
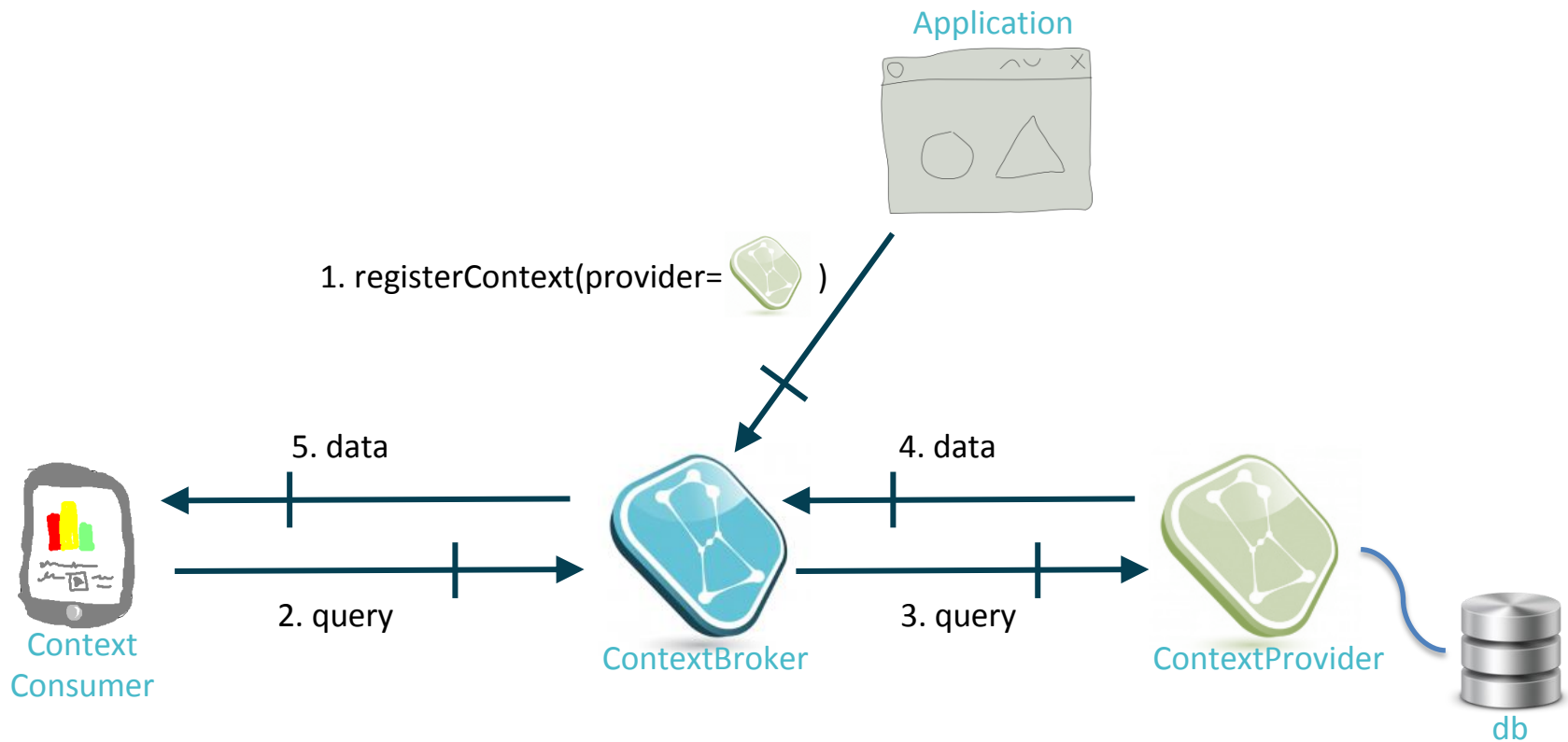
Query filters

- Filters can be also used in subscriptions
    - id
    - type
    - id pattern
    - type pattern
    - attribute values
    - metadata value
    - geographical location

POST <cb_host>:1026/v2/subscriptions
…
{
  "subject": {
   "entities": [
    {
    "id": "Car5",
    "type": "Car"
    },
    {
    "idPattern": "^Room[2-5]",
     "type": "Room"
    },
    {
     "id": "D37",
    "typePattern": "Type[ABC]"
    },
   ],
   "condition": {
    "attrs": ["temperature"],
    "expression": {
    "q": "temperature>40",
    "mq": "humidity.avg==80..90",
     "georel": "near;maxDistance:100000",
     "geometry": "point",
     "coords": "40.418889,-3.691944"
    }
    }
   },
   …
}

FIWARE
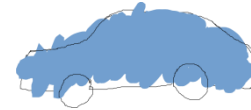
Registration & Context Providers

- Uncached queries and updates

Application

1. registerContext(provider=)

5. data

4. data

Context
Consumer

2. query

ContextBroker

3. query

ContextProvider

db

Registration & Context Providers

```
POST <cb_host>:1026/v1/registry/registerContext
...
{
  "contextRegistrations": [
    {
      "entities": [
        {
          "type": "Car",
          "isPattern": "false",
          "id": "Car1"
        },
        "attributes": [
          {
            "name": "speed",
            "type": "float",
            "isDomain": "false"
          }
        ],
        "providingApplication": "http://contextprovider.com/Cars"
      }
    ],
    "duration": "P1M"
}
```
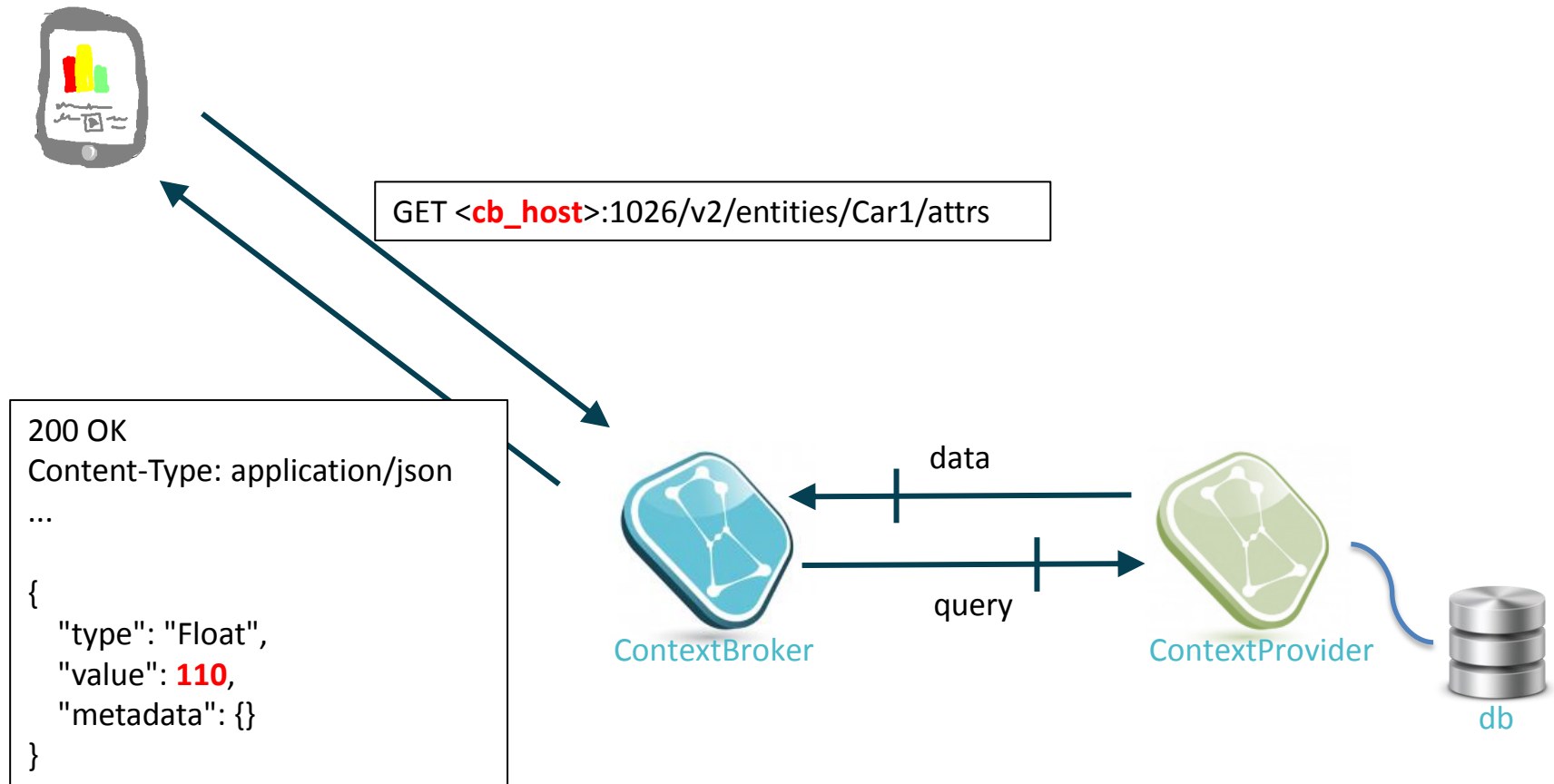
```
200 OK
...
{
  "duration" : "P1M",
  "registrationId" : "52a744b011f5816465943d58"
}
```

Context management availability functionality not yet specified in NGSIv2. Thus, a NGSIv1 operation is used to create the registration.

Registration & Context Providers

GET <**cb_host**>:1026/v2/entities/Car1/attrs

```
200 OK
Content-Type: application/json
...

{
   "type": "Float",
   "value": 110,
   "metadata": {}
}
```

data

query

ContextBroker

ContextProvider

db

Multitenancy

- Simple multitenant model based on logical database separation.

- It eases tenant-based authorization provided by other components.

- Just use an additional HTTP header called "Fiware-Service", whose value is the tenant name. Example:

  Fiware-Service: Tenant1

Context Broker

Tenant1

Tenant2

...
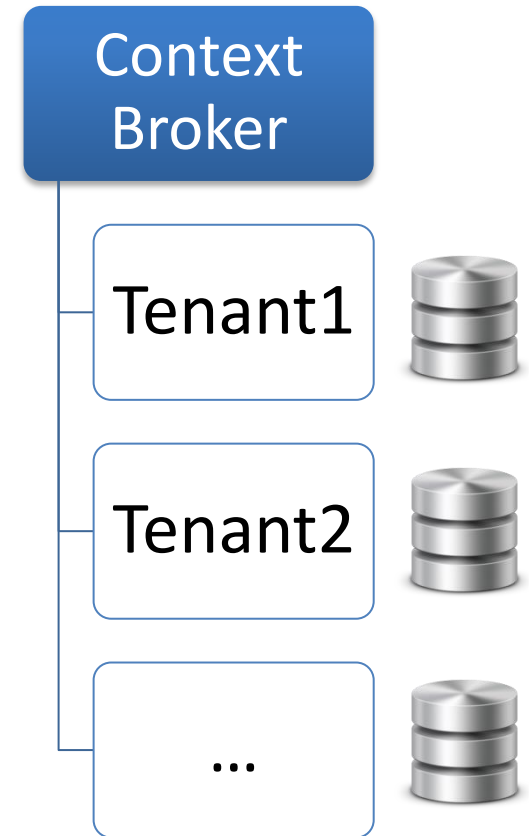
Service Paths

- A service path is a hierarchical scope assigned to an entity at creation time (with POST /v2/entities).
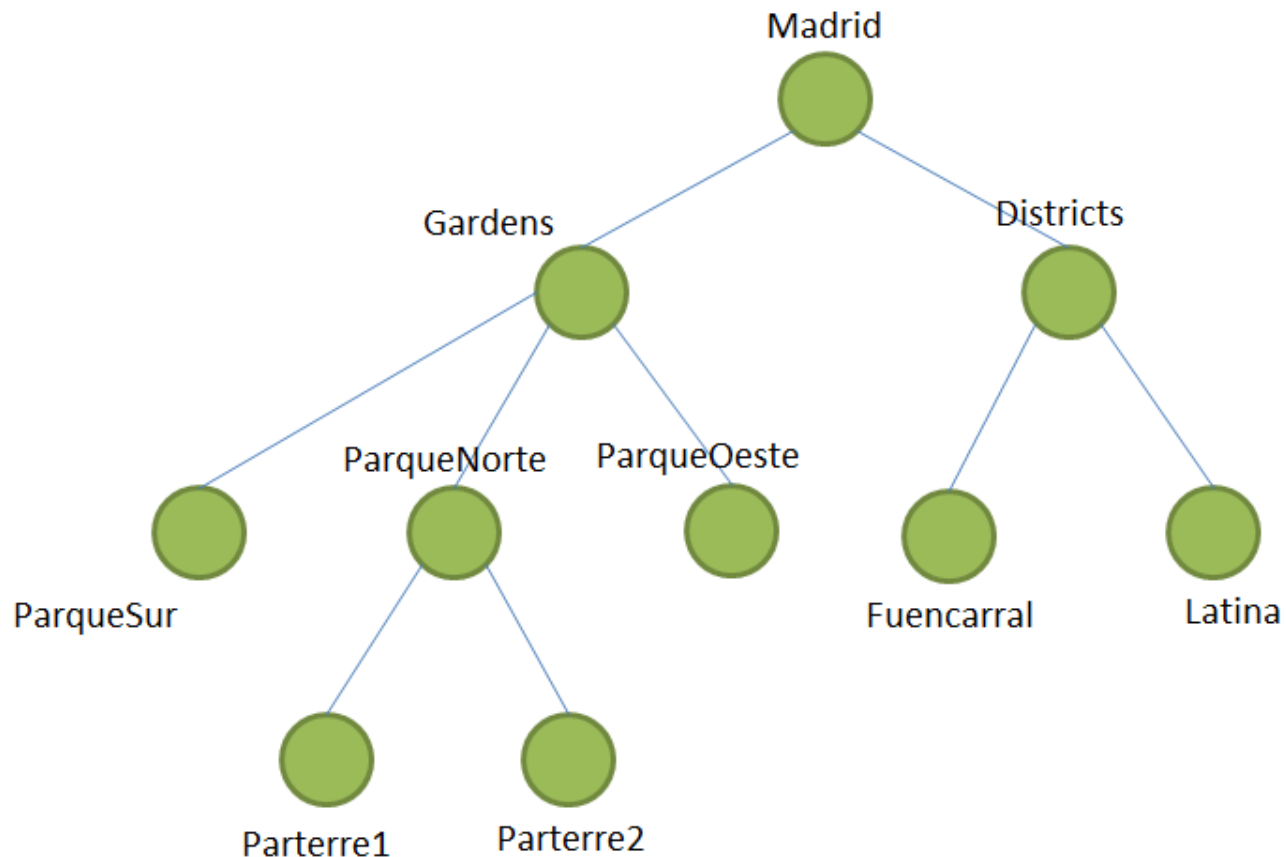
Service Paths

- In order to use a service path we put in a new HTTP header called "Fiware-ServicePath". For example:

  Fiware-ServicePath: /Madrid/Gardens/ParqueNorte/Parterre1

- Properties:

  - A query on a service path will look only into the specified node
  - Use "ParentNode/#" to include all child nodes
  - Queries without Fiware-ServicePath resolve to "/#"
  - Entities will fall in the "/" node by default

ParqueSur   ParqueNorte   ParqueOeste

Parterre1   Parterre2
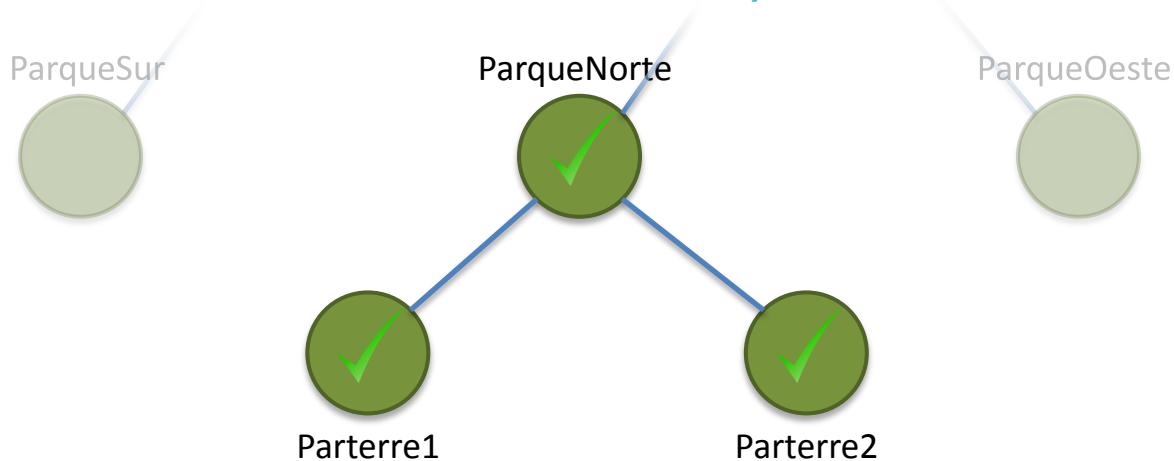
Service Paths

- Properties (continued):
  - You can OR a query using a comma (,) operator in the header
    - For example, to query all street lights that are either in ParqueSur or in ParqueOeste you would use:
      - *ServicePath: Madrid/Gardens/ParqueSur, Madrid/Gardens/ParqueOeste*
    - You can OR up to 10 different scopes.
  - Maximum scope levels: 10
    - Scope1/Scope2/…/Scope10
  - You can have the same element IDs in different scopes (be careful with this!)
  - You can't change scope once the element is created
  - One entity can belong to only one scope
  - It works not only with queries, but also with subscriptions/notifications
  - It works not only in NGSI10, but also with registrations/discoveries (NGSI9)
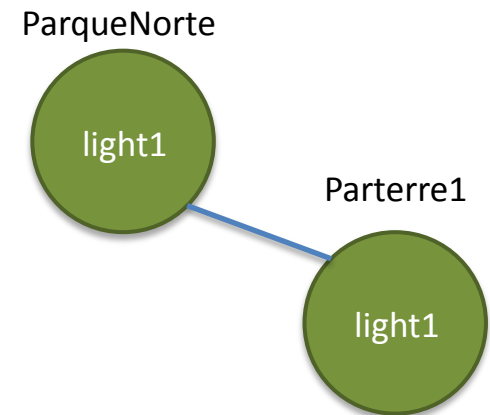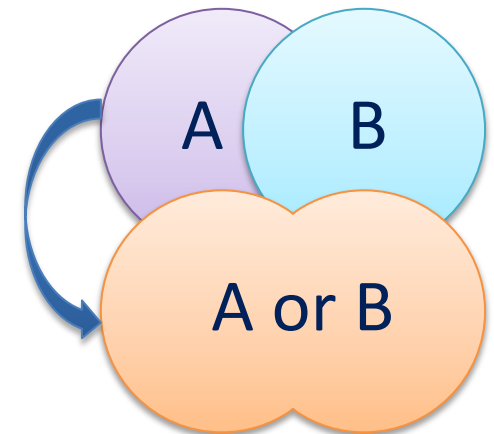
Cross-Origin Resource Sharing (CORS)

- Useful for programming clients that run entirely in browser without backend
- Support in GET requests
- Controlled by the **-corsOrigin** CLI parameter at boot time

```
GET <cb_host>:1026/v2/entities/Car1
```
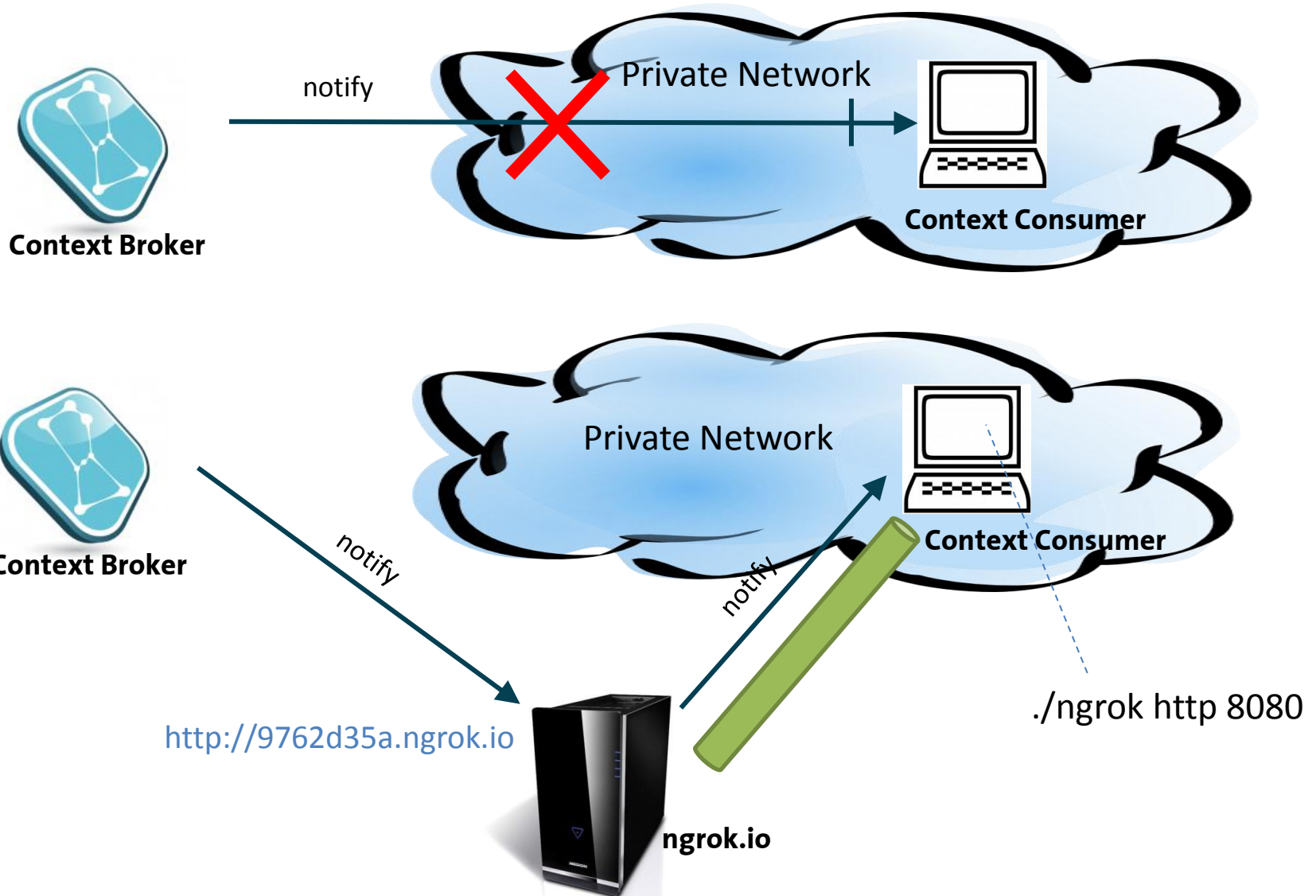
```
200 OK
Access-Control-Allow-Origin: *

...
{
   "speed": [
   ....
}
```

Notifiying services in private networks

notify

Private Network

Context Consumer

Context Broker

notify

Private Network

Context Consumer

Context Broker

notify

./ngrok http 8080

http://9762d35a.ngrok.io

ngrok.io

FIWARE

FIWARE Lab FIWARE Ops

Would you like to know more?

- ## The easy way
  - This presentation: google for "fermingalan slideshare" and search the one named "Managing Context Information at large scale"
  - Orion User Manual: google for "Orion FIWARE manual" and use the first hit
  - Orion Catalogue page: google for "Orion FIWARE catalogue" and use the first hit

- ## References
  - NGSIv2 Specification
    - http://telefonicaid.github.io/fiware-orion/api/v2/stable/
  - This presentation
    - http://www.slideshare.net/fermingalan/fiware-managing-context-information-at-large-scale
  - Orion Catalogue:
    - http://catalogue.fiware.org/enablers/publishsubscribe-context-broker-orion-context-broker
  - Orion support trhough StackOverflow
    - Ask your questions using the "fiware-orion" tag
    - Look for existing questions at http://stackoverflow.com/questions/tagged/fiware-orion

Thanks!

FIWARE

OPEN APIs FOR OPEN MINDS

FIWARE Lab
Spark your imagination

FIWARE Ops
Easing your operations

www.fiware.org
@Fiware