Smart Control of the Climate Resilience in European Coastal Cities

The sensor services: interfacing the world

Panagiotis Chatzikamaris (Tero)

Work Package 5: Pre/post-EBA Interventions Evidence Collection and Knowledge Marketplace

SCORE WEBINAR #4 | 25 SEPTEMBER 2023

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101003534

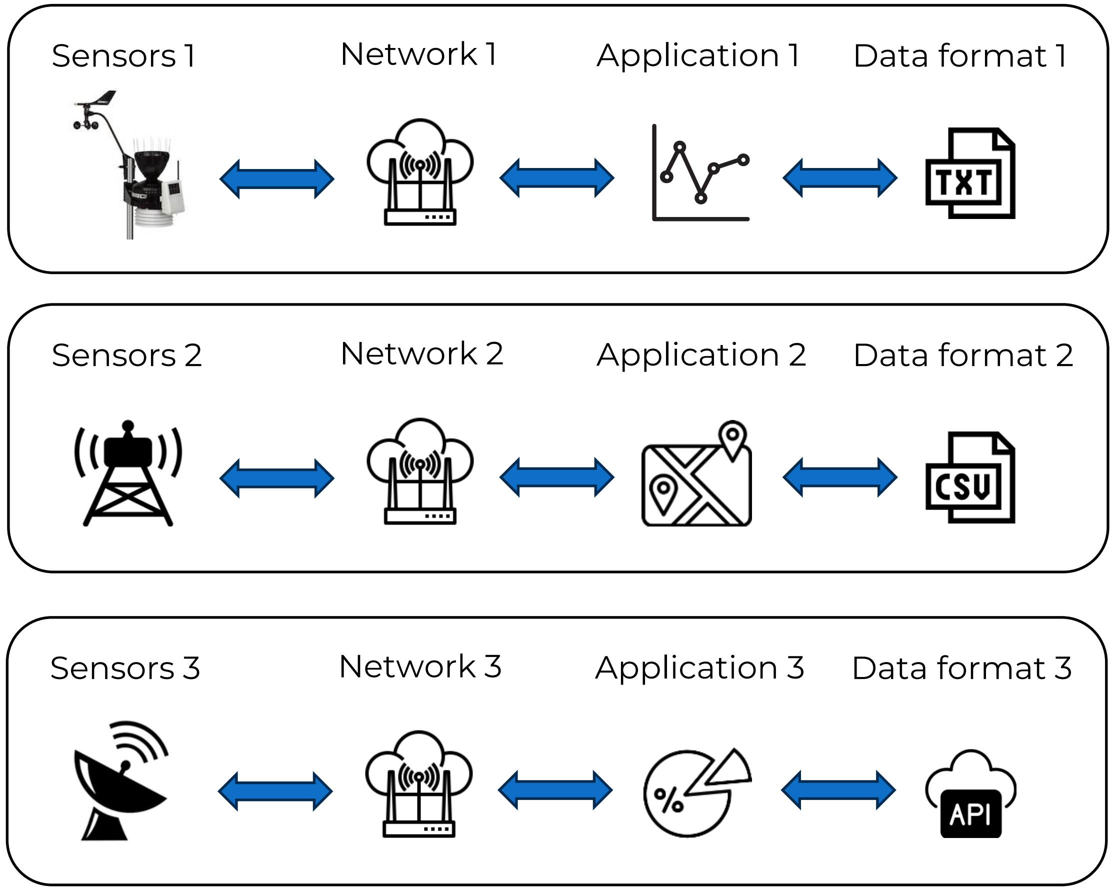
The requirements

- Deal with the great heterogeneity of data formats and communication methods and protocols.
- Comply with standards and specifications relevant to spatial data, such as the ones set by the Open Geospatial Consortium (OGC)
- Standardise the spatial data collected from various observation networks and sensors to be deployed during the SCORE project
- Ensure data interoperability, following open standards and formats, as a key factor for successful knowledge sharing and exchange of information between multiple systems and applications
- Build a citizen science platform to support the activities of complementing the institutional sensor data with data from low-cost sensors
- **Location matters!**

The challenges (1/2)

Many different data sources providing sensor data in different formats

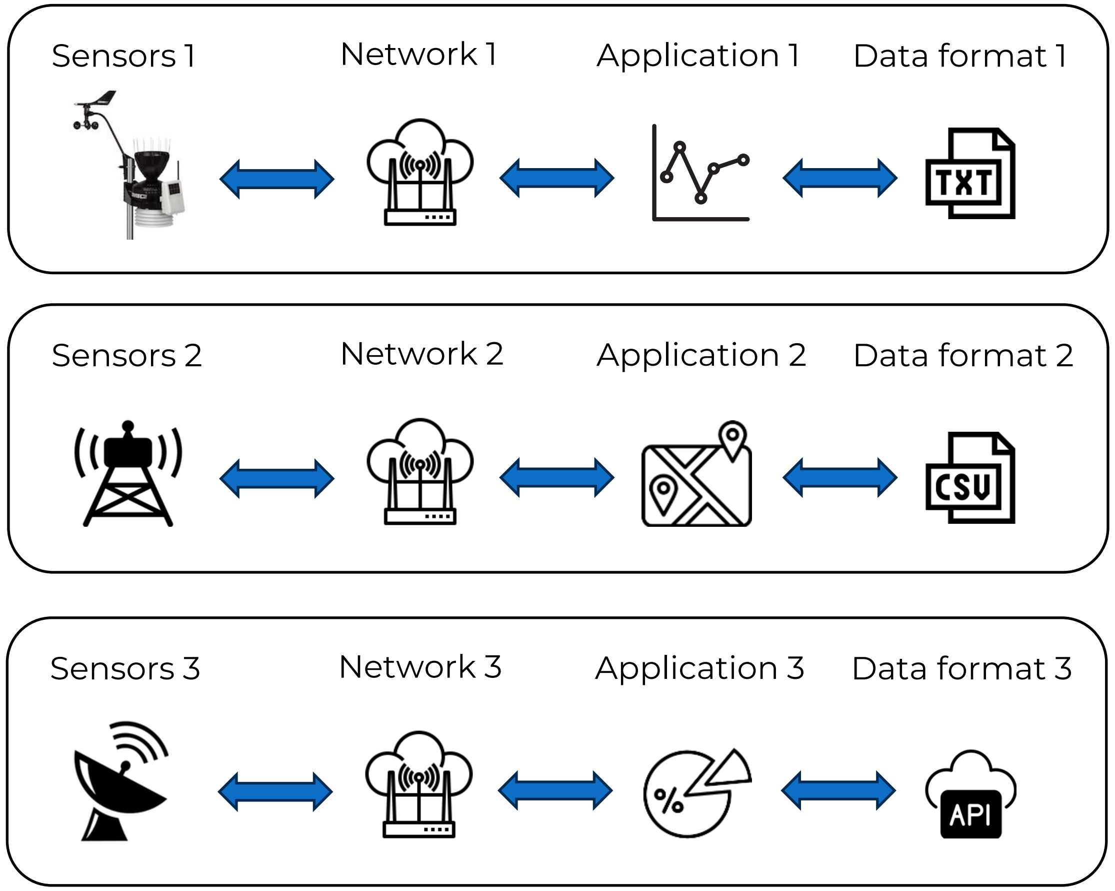
CCLL sensor networks/professional sensors



The challenges (1/2)

Many different data sources providing sensor data in different formats

CCLL sensor networks/professional sensors



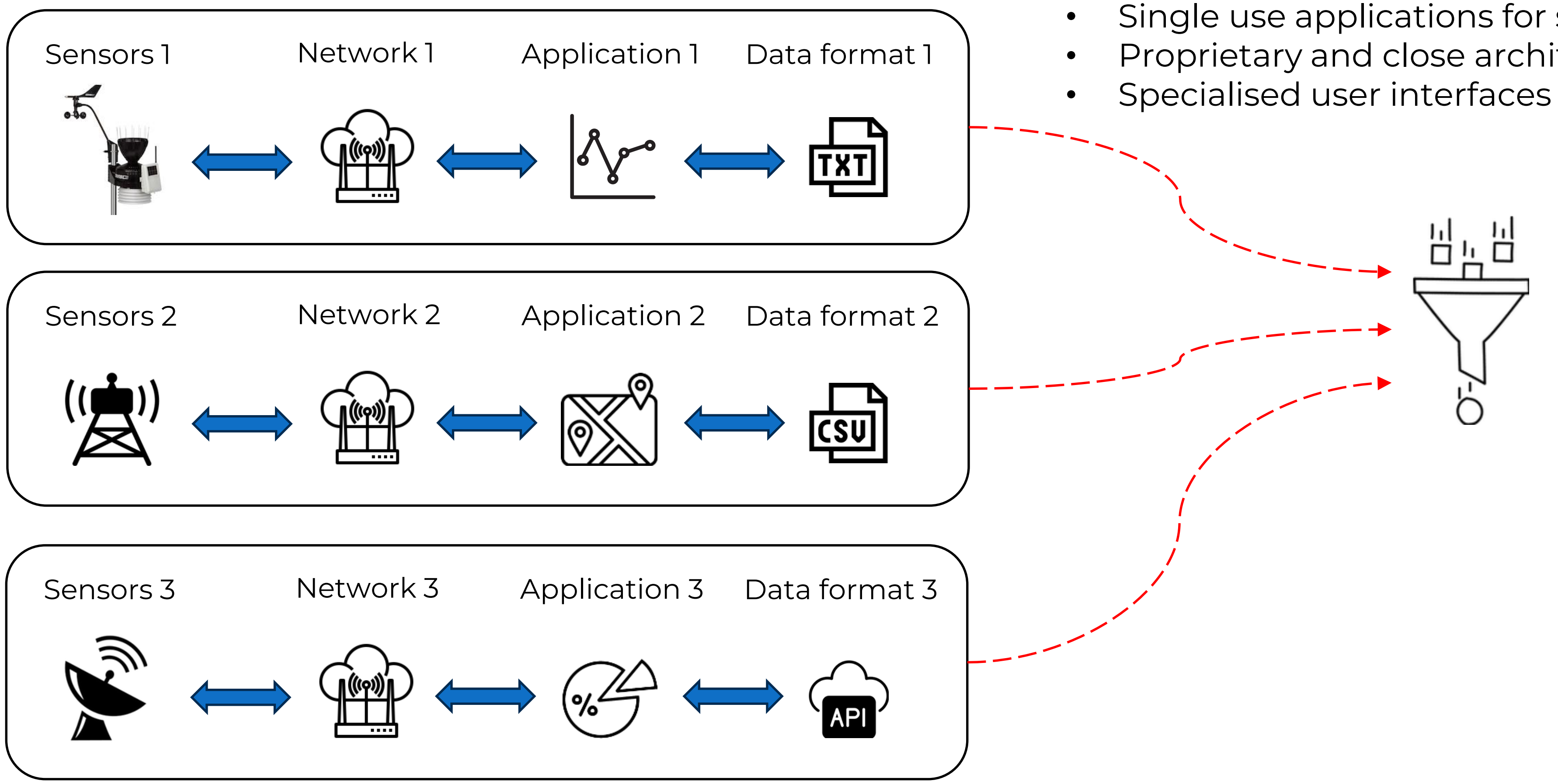
Disadvantages

- Single use applications for specific purpose
- Proprietary and close architectures
- Specialised user interfaces

The challenges (1/2)

Many different data sources providing sensor data in different formats

CCLL sensor networks/professional sensors



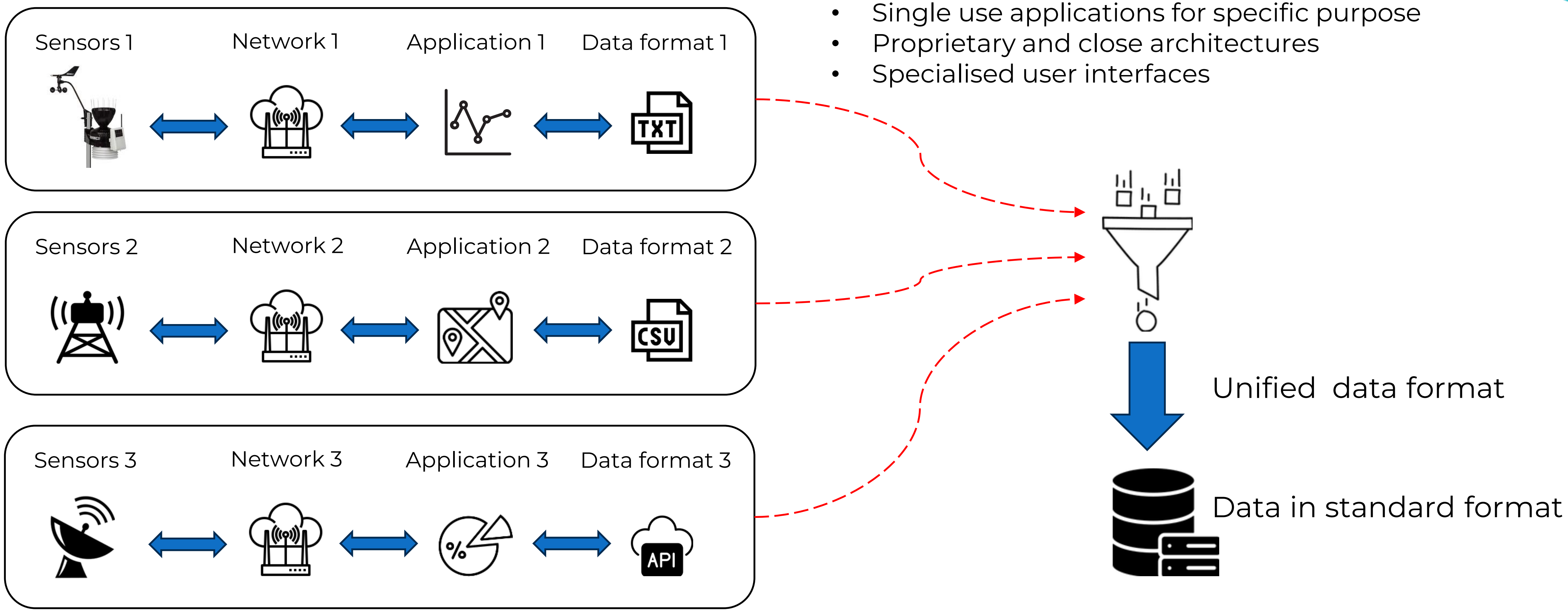
Disadvantages

- Single use applications for specific purpose
- Proprietary and close architectures
- Specialised user interfaces

The challenges (1/2)

Many different data sources providing sensor data in different formats

CCLL sensor networks/professional sensors

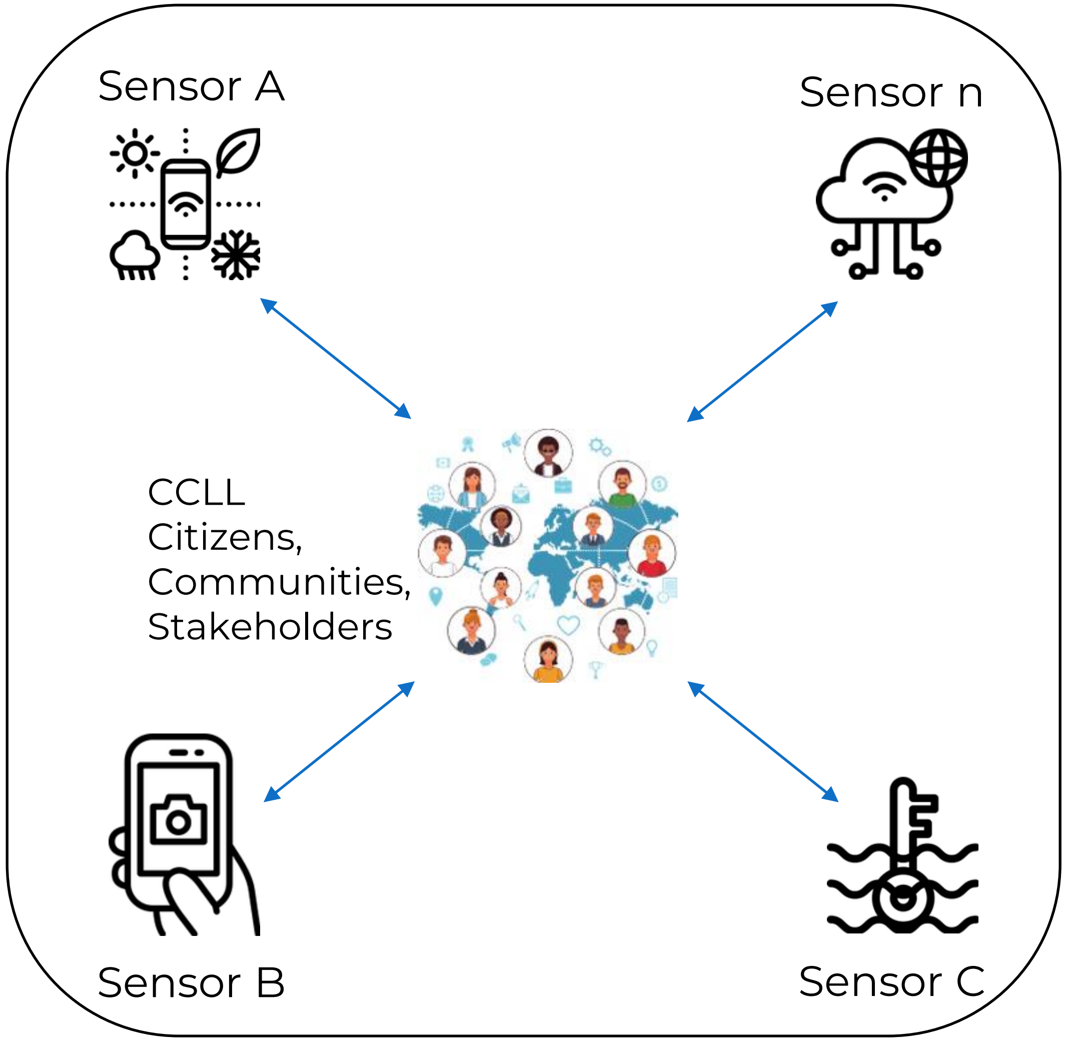


Disadvantages

- Single use applications for specific purpose
- Proprietary and close architectures
- Specialised user interfaces

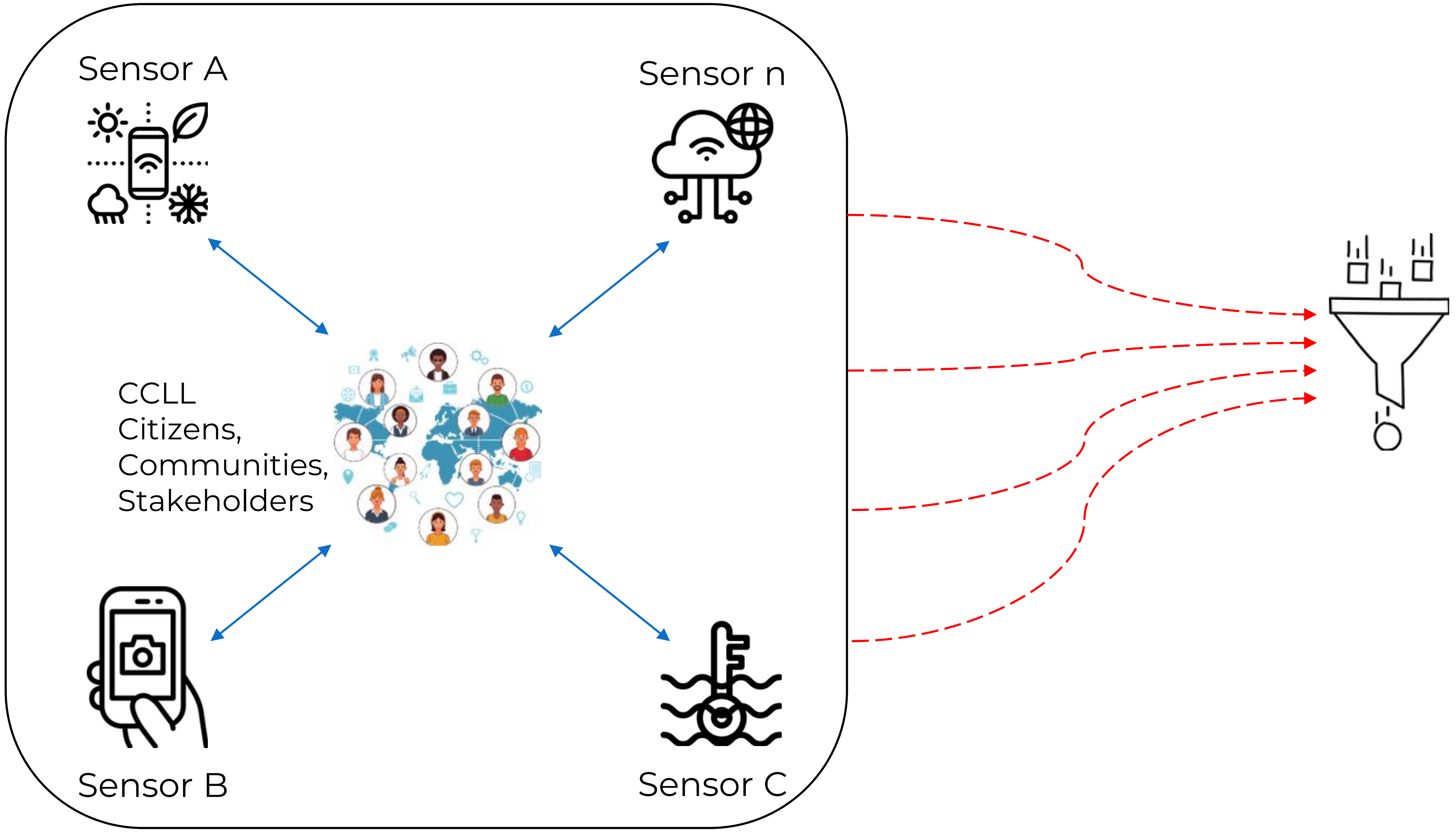
The challenges (2/2)

Citizen science



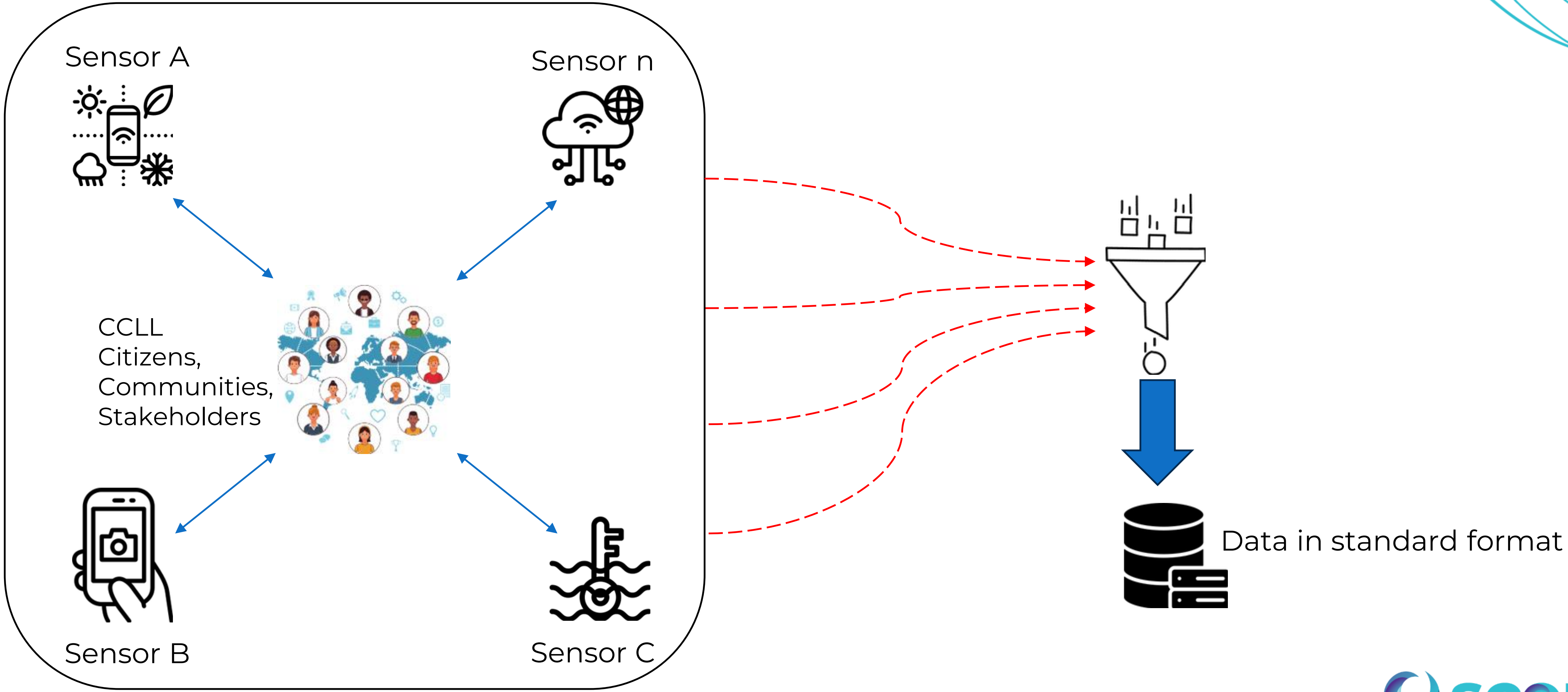
The challenges (2/2)

Citizen science



The challenges (2/2)

Citizen science



The solution – OGC SensorThings API (1/2)

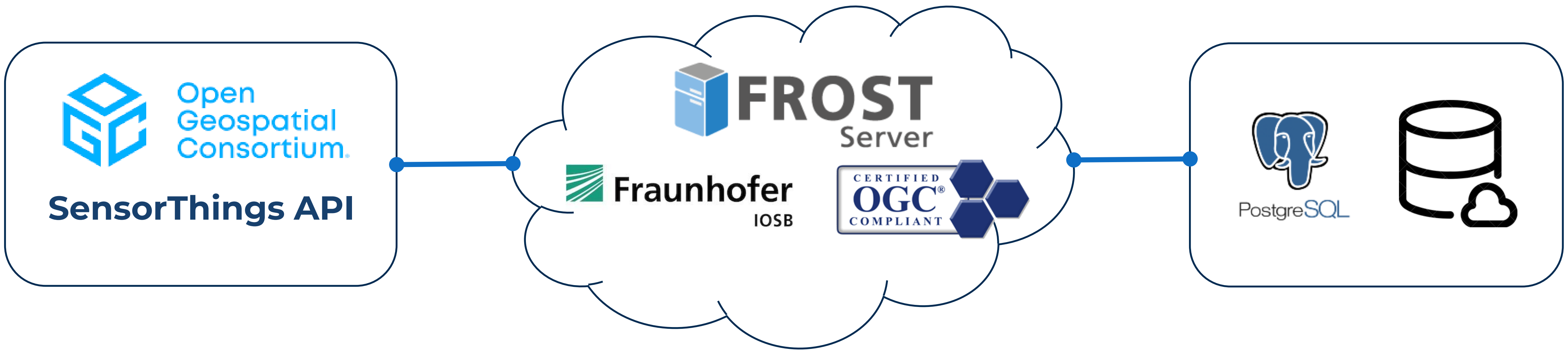
SensorThings API features:

- An Open Geospatial Consortium (OGC) standard that provides an open, geospatial-enabled and unified way to interconnect devices, data and applications over the Web.
- Data model designed based on the OGC/ISO 19156:2011 Observation and Measurement (O&M) model
- 2 Parts: a) Sensing (data management and sharing), b) Tasking (IoT devices control)
- Lightweight RESTful HTTP (GET, POST, DELETE, PATCH) service,
- JSON encoded data
- Event-driven (MQTT)
- IoT spatial data interoperability providing the ability between two or more (IoT) systems or components to exchange sensor data and metadata and to use the information that has been exchanged.
- Location aware sensing, designed for both simple and complex geospatial applications

OGC SensorThings API

Deployment

- Implementation of the OGC SensorThings API is based on the Fraunhofer Open Source SensorThings API Server (FROST Server). The FROST Server connects to the database server which is based on the open-source PostgreSQL database management system with the PostGIS extension enabled.



OGC SensorThings API

Data model

- ✦ The SensorThings API data model consists of eight entities that are linked to each other.

1 **Thing**: a physical or virtual object where a sensor is located. (e.g. Weather station device, a room, etc.)

2 **Location**: describes the physical location of a Thing (e.g. Local University)

3 **Sensor**: describes the sensor that provides values (e.g. thermometer)

4 **ObservedProperty**: specifies the phenomenon that is being measured by Sensor (e.g. temperature)

5 **Observetion**: a single measurement value (e.g values of the thermometer)

6 **Datastream**: is a collection of Observations (e.g a time-series of thermometer observations)

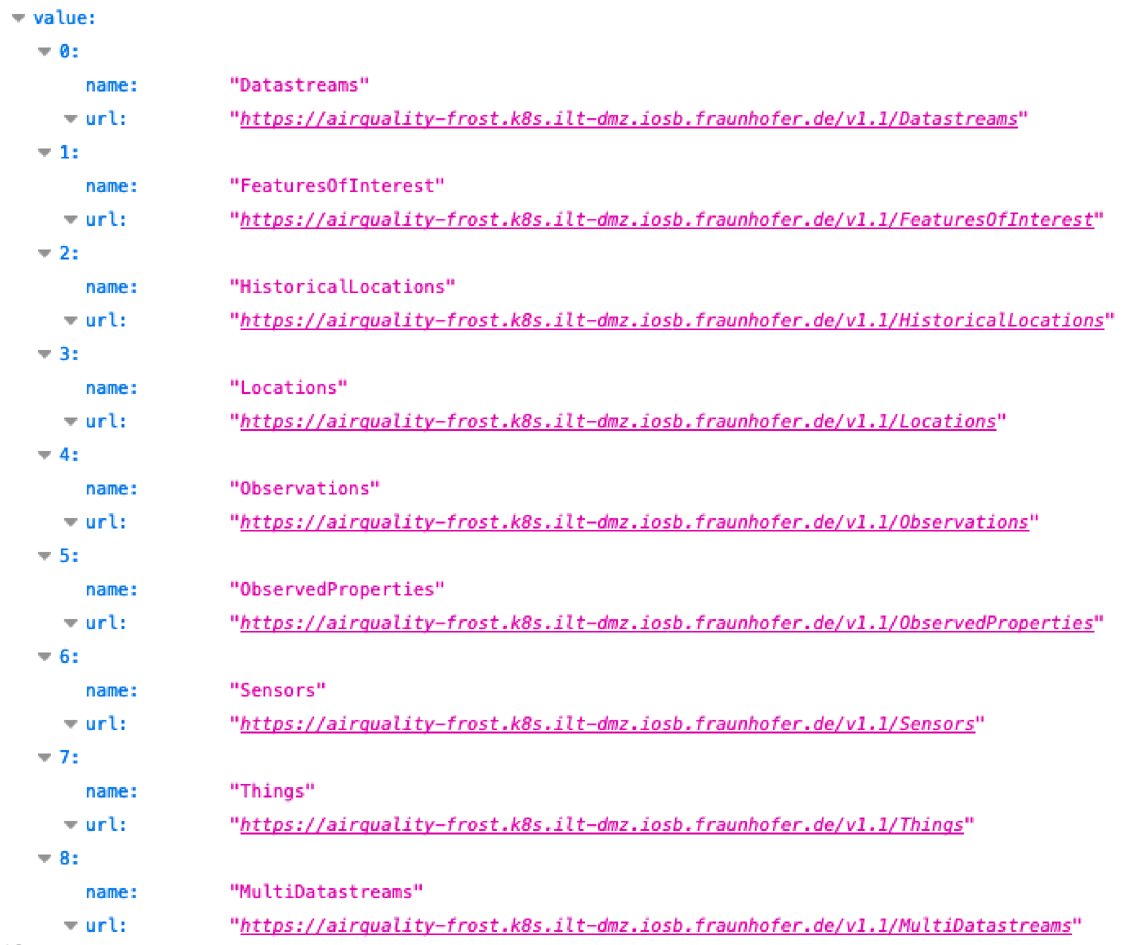
7 **HistoricalLocation**: provides the times of the current and previous locations of the Thing

8 **FeatureOfInterest**: the geographic area or location where the Observation was made (e.g. location of Local University)

OGC SensorThings API

Getting data

Case study: https://airquality-frost.k8s.ilt-dmz.iosb.fraunhofer.de/v1.1



```
▾ value:
  ▾ 0:
      name:     "Datastreams"
    ▾ url:      "https://airquality-frost.k8s.ilt-dmz.iosb.fraunhofer.de/v1.1/Datastreams"
  ▾ 1:
      name:     "FeaturesOfInterest"
    ▾ url:      "https://airquality-frost.k8s.ilt-dmz.iosb.fraunhofer.de/v1.1/FeaturesOfInterest"
  ▾ 2:
      name:     "HistoricalLocations"
    ▾ url:      "https://airquality-frost.k8s.ilt-dmz.iosb.fraunhofer.de/v1.1/HistoricalLocations"
  ▾ 3:
      name:     "Locations"
    ▾ url:      "https://airquality-frost.k8s.ilt-dmz.iosb.fraunhofer.de/v1.1/Locations"
  ▾ 4:
      name:     "Observations"
    ▾ url:      "https://airquality-frost.k8s.ilt-dmz.iosb.fraunhofer.de/v1.1/Observations"
  ▾ 5:
      name:     "ObservedProperties"
    ▾ url:      "https://airquality-frost.k8s.ilt-dmz.iosb.fraunhofer.de/v1.1/ObservedProperties"
  ▾ 6:
      name:     "Sensors"
    ▾ url:      "https://airquality-frost.k8s.ilt-dmz.iosb.fraunhofer.de/v1.1/Sensors"
  ▾ 7:
      name:     "Things"
    ▾ url:      "https://airquality-frost.k8s.ilt-dmz.iosb.fraunhofer.de/v1.1/Things"
  ▾ 8:
      name:     "MultiDatastreams"
    ▾ url:      "https://airquality-frost.k8s.ilt-dmz.iosb.fraunhofer.de/v1.1/MultiDatastreams"
```

HTTP requests

https://base.url/v1.1 → data model entities

https://base.url/v1.1/Things → get collection of data related to entity

https://base.url/v1.1/Things → get collection of data related to entity

https://base.url/v1.1/Things(1) → get data for particular entity

https://base.url/v1.1/Things(1)/Datastreams → get data from “linked” entity

OGC SensorThings API

Getting data – tailoring responses

SensorThings API provides the following request parameters for customizing the response data.

$top: specify the maximum number of objects to be returned. The usual default setting for $top is 100.

$skip: used for paging, skip over the first n records and provide records from the n + 1 on.

$count: return the total number of objects in the response. The usual default setting for $count is false.

$orderBy: used to specify that the returned objects should be ordered by a specific attribute, either ascending or descending.

$select: specify exactly which attributes are to be provided in the response.

$filter: specify filters that control which entities are returned.

$expand: create a response returning multiple object types nested within each other.

HTTP requests

https://base.url/v1.1/Datastreams?$top=5

https://base.url/v1.1/Datastreams?$skip=10

https://base.url/v1.1/Datastreams?$count=true

https://base.url/v1.1/Observations?$orderBy=attribute asc

https://base.url/v1.1/Observations?$select=attribute1,attribute2...

https://base.url/v1.1/Observations?$filter=attribute1 gt 5

https://base.url/v1.1/Datastreams?$expand=linkedentity

The solution – Data Collectors (2/2)

Data collectors for CCLL sensor networks/professional sensors

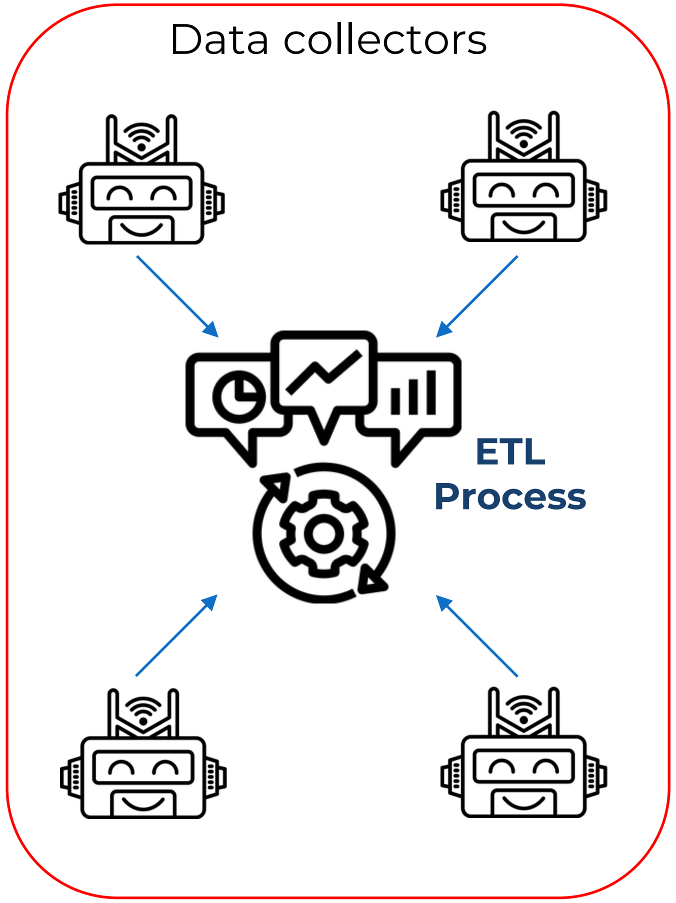
- ✦ Individual executable scripts that trigger processes to communicate with different systems and applications and collect sensor data.
- ✦ Can be scheduled to run continuously at different time intervals to collect real-time data with different sampling rates.
- ✦ Can read data with different structures and data formats such as json, txt, csv, xml, excel, etc.
- ✦ Can connect to third party APIs both public and private that require authentication/authorization.
- ✦ Integrate the process that is responsible for the transformation and publication of heterogenous data to the standard format that is supported by the SensorThings API.

Technology

Python programming language

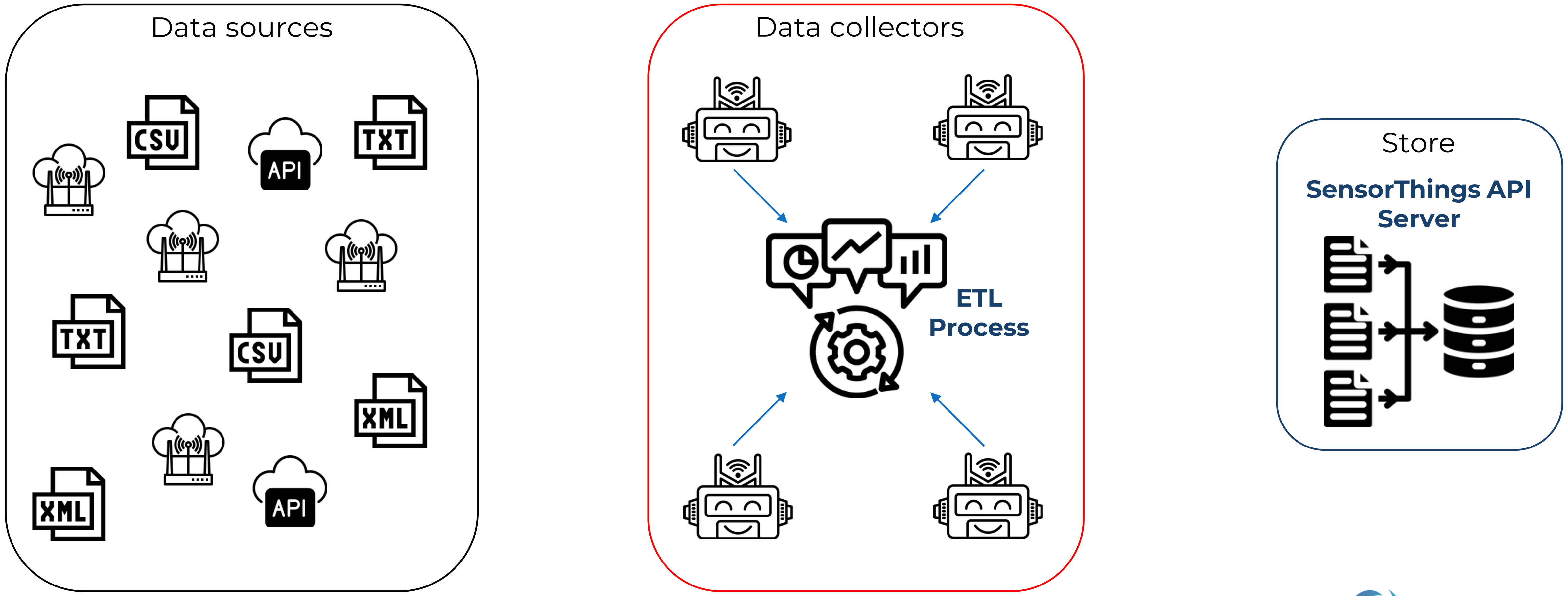
Data Collectors

How do they work?



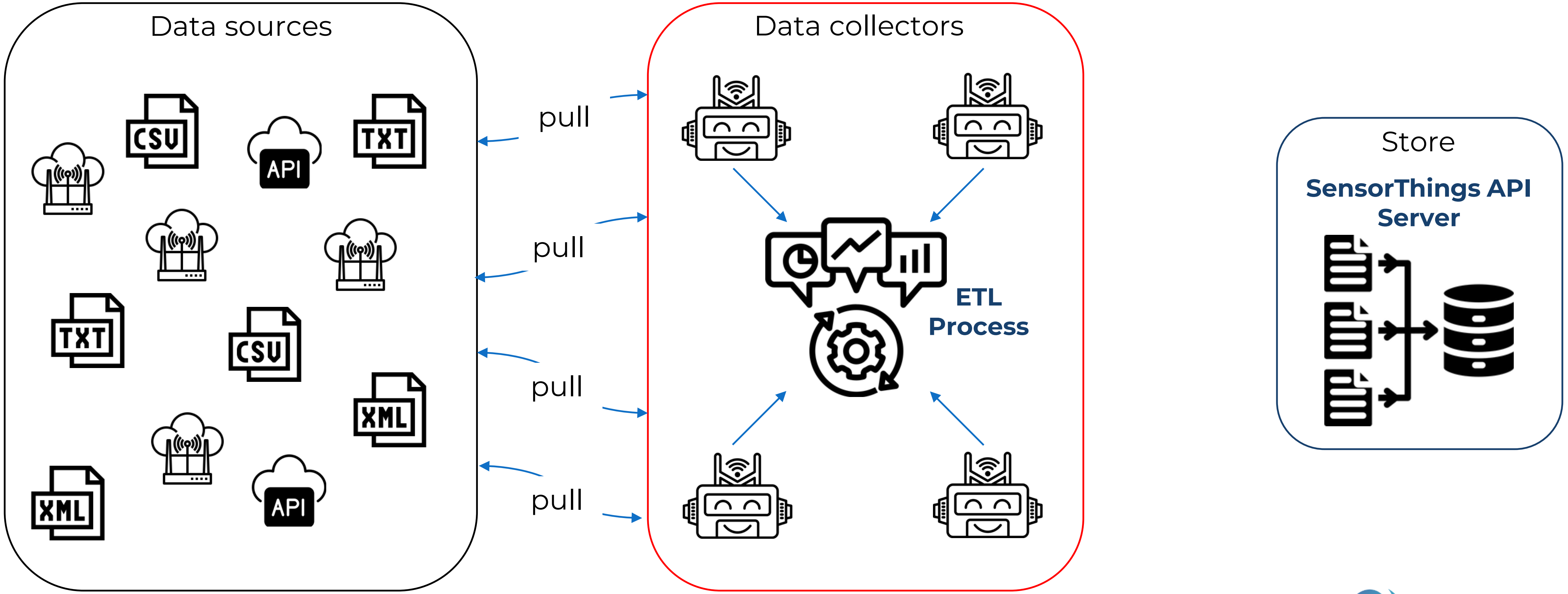
Data Collectors

How do they work?



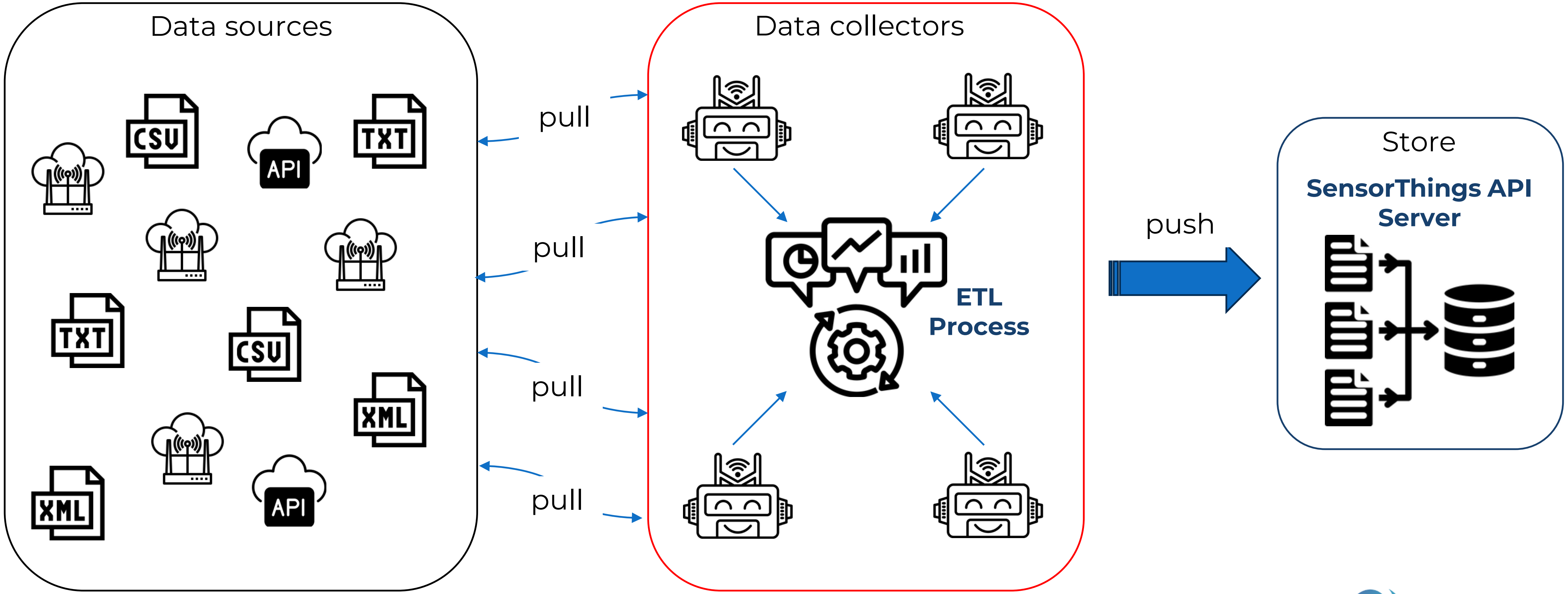
Data Collectors

How do they work?



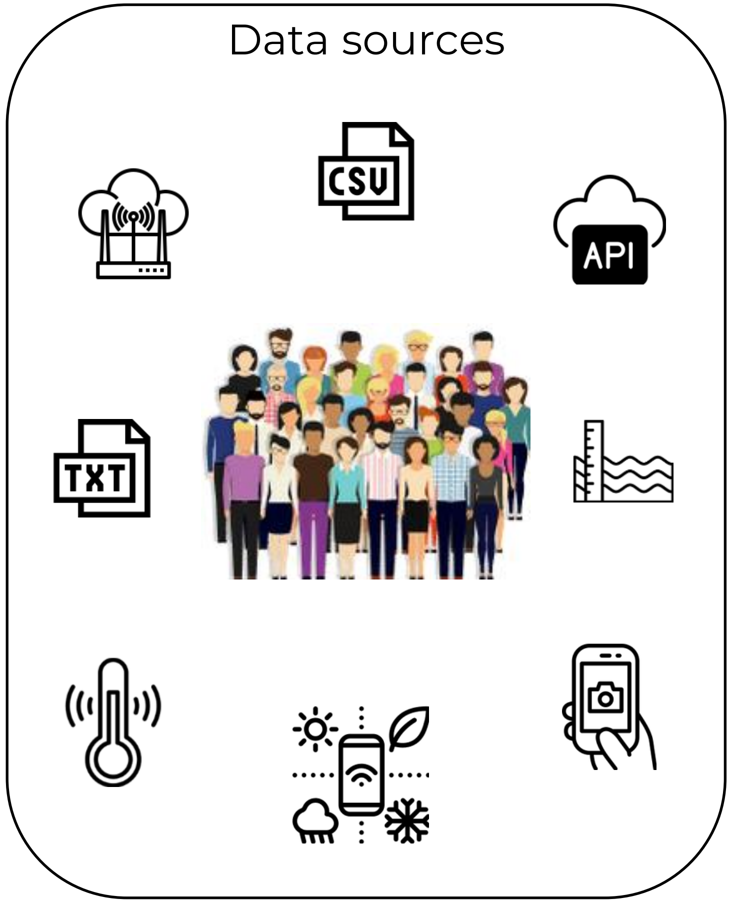
Data Collectors

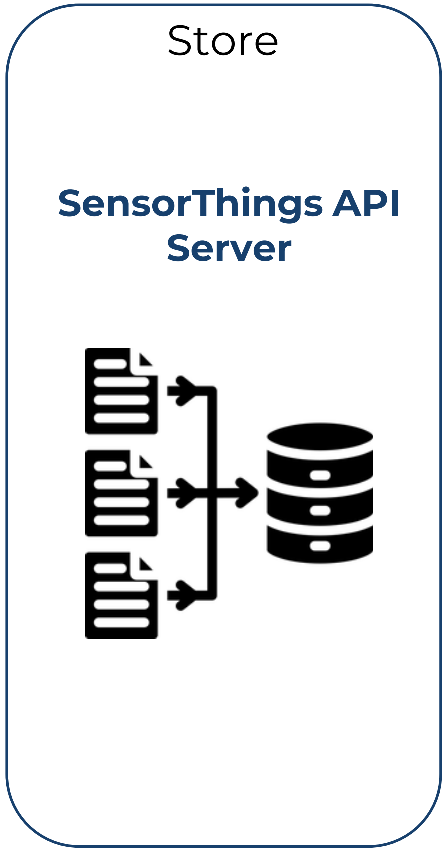
How do they work?



Citizen science

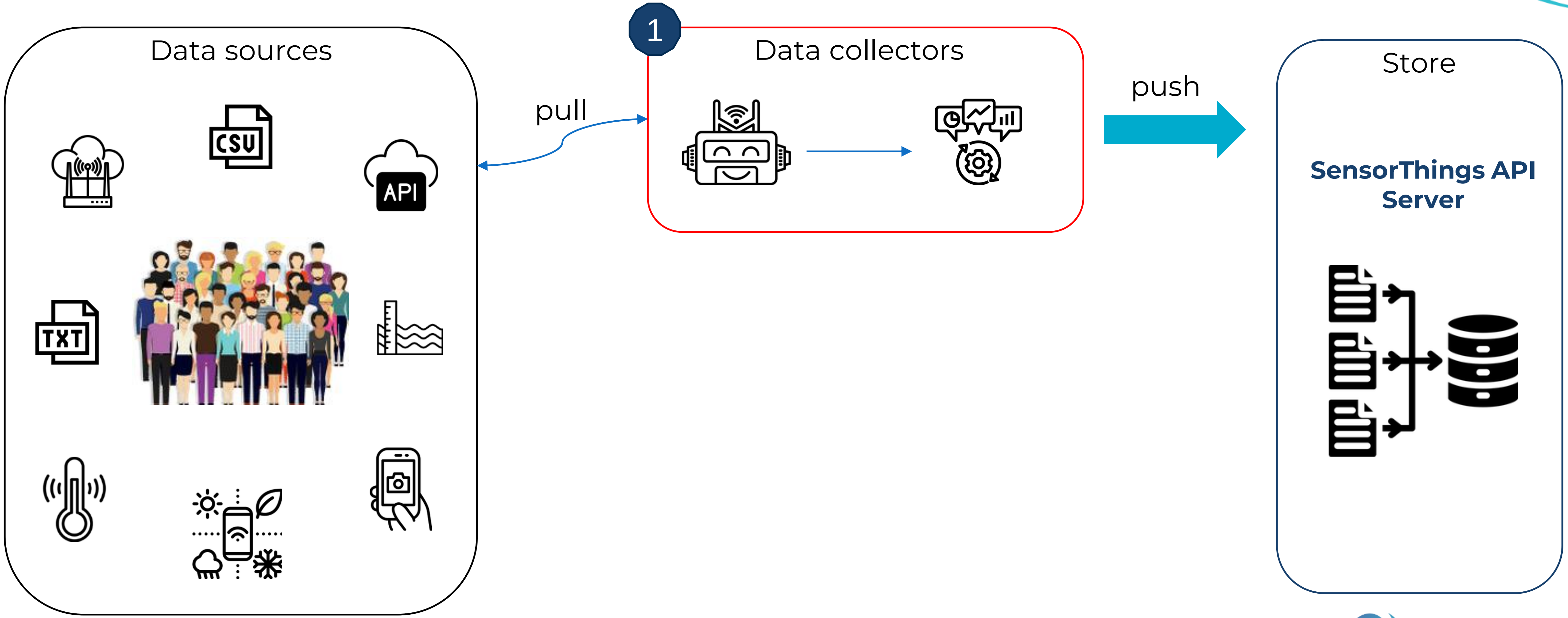
Data collection from citizen sensors at CCLLs





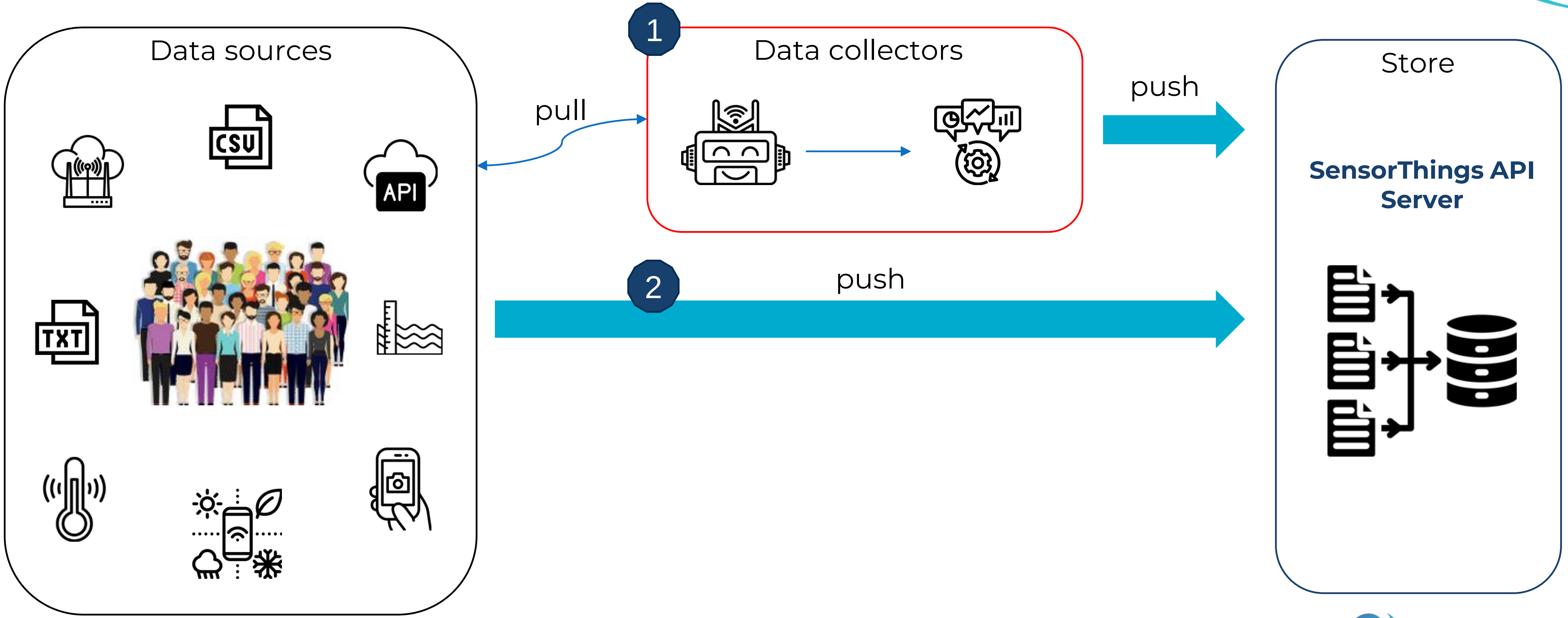
Citizen science

Data collection from citizen sensors at CCLLs



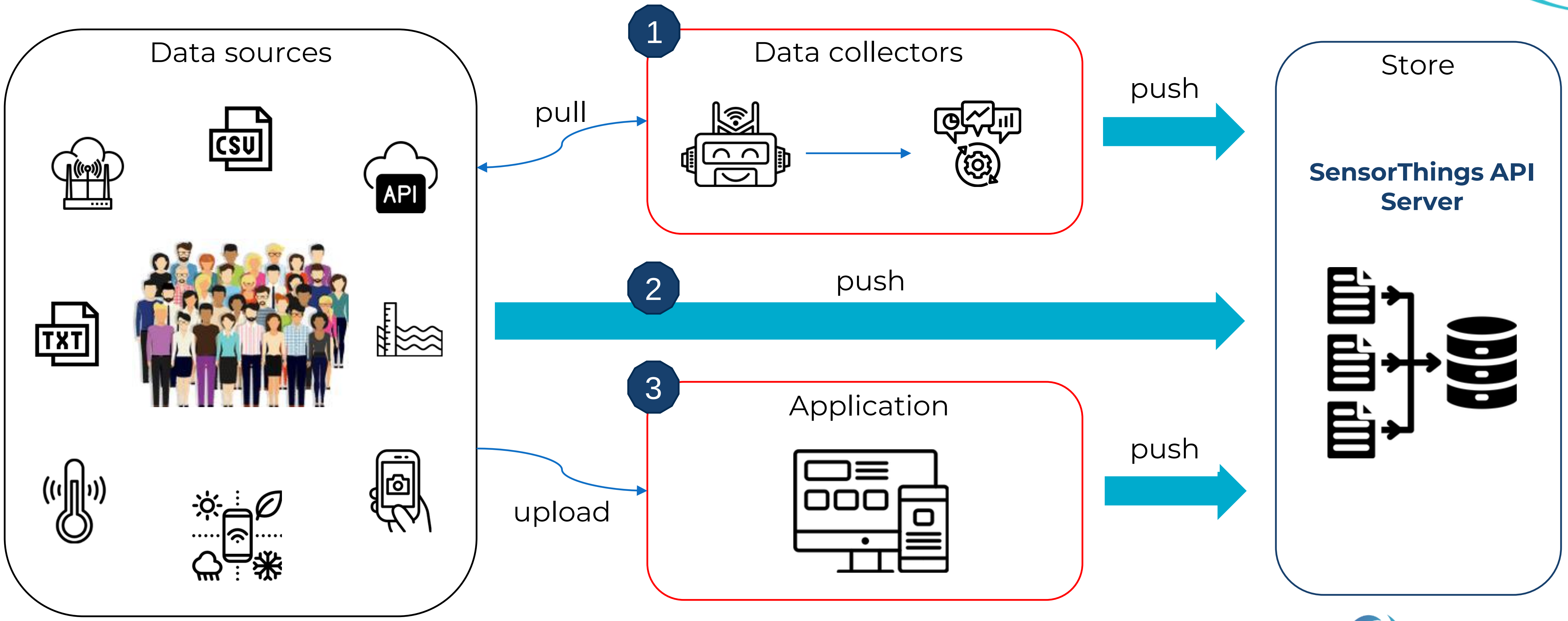
Citizen science

Data collection from citizen sensors at CCLLs



Citizen science

Data collection from citizen sensors at CCLLs



score

Smart Control of the Climate Resilience in European Coastal Cities

Thank you!

www.score-eu-project.eu

contact@score-eu-project.eu

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101003534