

30/08/2021

# Environnemental Sensing

Présentation

Environmental Sensing



Confidential C

## TABLE DES MATIERES

|       |  |    |
|-------|--|----|
| 1     | Introduction .....   | 3  |
| 1.1   | Contexte .....   | 3  |
| 1.2   | Objectifs .....  | 4  |
| 1.3   | Données environnementales .....                              | 4  |
| 1.4   | Cadre normatif .....   | 4  |
| 1.5   | Objectifs .....  | 5  |
| 1.6   | Contenu .....  | 5  |
| 2     | Cas d'usage et bénéfices .....                               | 6  |
| 2.1   | Mesure d'exposition – interopérabilité horizontale .....     | 6  |
| 2.2   | Cycle de vie des données – interopérabilité verticale .....  | 7  |
| 2.3   | Assimilation de données – interopérabilité horizontale ..... | 7  |
| 2.4   | Bénéfices attendus .....                                     | 8  |
| 3     | Présentation .....   | 11 |
| 3.1   | Principes généraux .....                                     | 11 |
| 3.2   | Structure de données .....                                   | 12 |
| 3.3   | Opérations .....   | 13 |
| 3.4   | Connecteurs .....  | 14 |
| 3.5   | Utilisation .....  | 14 |
| 3.6   | Exemples .....   | 15 |
| 3.6.1 | Mesure simple .....  | 15 |
| 3.6.2 | Mesure sur un trajet .....                                   | 16 |
| 3.6.3 | Mesure multiple .....  | 17 |
| 4     | Structuration .....  | 18 |
| 4.1   | Modèle de données .....                                      | 18 |
| 4.1.1 | ESObject .....   | 18 |
| 4.1.2 | ESObs .....  | 19 |
| 4.1.3 | ESValue .....  | 20 |
| 4.2   | Traitements .....  | 20 |
| 5     | Mise en œuvre .....  | 22 |
| 5.1   | Principes .....  | 22 |

|       |  |    |
|-------|--|----|
| 5.2   | Exemples .....                         | 22 |
| 5.2.1 | Capteur.....                           | 22 |
| 5.2.1 | Stockage en Base de données.....       | 22 |
| 5.2.2 | Extraction d'une Base de données ..... | 23 |
| 5.2.3 | Visualisation d'une Observation .....  | 23 |
| 6     | FAQ.....                               | 26 |
| 7     | annexe.....                            | 28 |
| I.    | Annexe 1 : Modèle de données .....     | 29 |
| I.    | Annexe 2 : Mapping Bluetooth .....     | 30 |

## 1 INTRODUCTION

---

### 1.1 CONTEXTE

On constate une grande diversité de solutions traitant de données environnementales et paradoxalement, l'interopérabilité entre ces solutions reste très faible, ce qui oblige à mettre en œuvre des interfaces spécifiques et conduit à dénaturer les données d'origine par des retraitements successifs.

Cette absence d'interopérabilité a pour conséquence une absence de solution transversale aussi bien au niveau des cas d'usage (intégration des contextes personnels, familiaux, professionnels, régionaux) qu'au niveau du cycle de vie des données environnementales : (1) production des données (acquisition, observation, modélisation), (2) de leur partage (réseaux, stockage), (3) de l'exploitation des informations (restitution, analyse).

La mise en place d'une interopérabilité forte, est conditionnée par :

- L'existence d'une structure de donnée unique et partagée (standard)
- La disponibilité d'outils ouverts d'exploitation de cette structure (connecteurs),

Trois conditions supplémentaires sont nécessaires pour que cette structuration soit adoptée et appliquée :

- Une mise en œuvre simple et adaptée à chaque besoin,
- Une construction de « convergence » plutôt que de « remise en cause » de l'existant,
- Une compatibilité avec la qualification « d'intérêt général » des données environnementales

### 1.2 ATTENTES

Les données environnementales (y compris mobilité et énergie) sont identifiées comme données « d'intérêt général » et sont donc soumises à une exigence d'interopérabilité croissante.

Le développement de villes et plus généralement de systèmes intelligents, plus adaptatifs et efficaces nécessite une gestion technique renforcée s'appuyant sur des données ouvertes de plus en plus nombreuses.

L'accès à une information environnementale localisée, fiable et compréhensible est une demande citoyenne relayée au travers de collectifs toujours plus importants.

### 1.3 OBJECTIFS

Ce document propose une solution d'interopérabilité des données environnementales applicable aussi bien au niveau de données unitaires (ex. émises par un capteur) que de données agrégées à différentes échelles de temps ou d'espace.

La structure de données proposée s'appuie sur les standards existants qu'elle complète en assurant une convergence.

Cette structure est implémentée au travers de connecteurs utilisables sur toute la chaîne de traitement (du capteur produisant une donnée au système d'analyse et de restitution d'un ensemble de données) incluant les interfaces avec les équipements réseau, les systèmes de calcul et les bases de données.

Les modèles de données et les connecteurs sont partagés et accessibles en open-source (licence Creative Commons BY-SA).

### 1.4 DONNEES ENVIRONNEMENTALES

Le Conseil National du Numérique dans son avis de juillet 2020 intitulé « **Faire des données environnementales des données d'intérêt général** » définit les données environnementales de la façon suivante :

*« Les données environnementales peuvent être définies de manière extensive comme toute donnée, par nature ou par destination, relative à l'environnement, à son état et/ou à ses flux d'interaction. Cette définition présente l'avantage de pouvoir qualifier plusieurs types de données – à l'instar des données agricoles, des données naturalistes ou des données relatives à la mobilité – comme des données environnementales. »*

Cette définition donne un cadre étendu à la notion de données environnementales, néanmoins, dans la structuration définie nous nous limiterons à une approche structurale : les données concernées sont qualifiées d'environnementales dans le sens où elles intègrent trois dimensions principales pour qualifier un centre d'intérêt :

- Une dimension spatiale,
- Une dimension temporelle,
- Une dimension physique évaluable

### 1.5 CADRE NORMATIF

Cette structuration respecte les concepts et principes définis dans le standard ISO 19156 – « Observation and Measurement » dont le champ d'application est rappelé ci-dessous :

*“This International Standard defines a conceptual schema for observations, and for features involved in sampling when making observations. These provide models for the exchange of information describing observation acts and their results, both within and between different scientific and technical communities.*

*Observations commonly involve sampling of an ultimate feature-of-interest. This International Standard defines a common set of sampling feature types classified primarily by topological dimension, as well as samples for ex-situ observations. The schema includes relationships between sampling features (subsampling, derived samples).*

*This International Standard concerns only externally visible interfaces and places no restriction on the underlying implementations other than what is needed to satisfy the interface specifications in the actual”.*

## 1.6 OBJECTIFS

Les objectifs poursuivis par la solution proposée sont de :

- Faciliter l’usage et le partage des données environnementales
- Banaliser aussi bien les équipements d’acquisition de données (capteurs) que les applications de traitement,
- Mettre en œuvre une architecture logicielle remplaçant toutes les opérations de codage/décodage (interfaces) par l’utilisation de connecteurs standards,
- Respecter et s’appuyer sur les principaux standards existants
- Partager et développer collectivement un ensemble de connecteurs open-source répondant à toutes les situations (plateforme)

## 1.7 CONTENU

Ce document décrit les bénéfices attendus de cette solution sur la base d’exemples de cas d’usage (chapitre 2).

Il présente ensuite les principes mis en œuvre et nécessaires à l’atteinte des objectifs ci-dessus (chapitre 3 et 4) et en particulier :

- la structure de données permettant de prendre en compte les différentes données environnementales définies,
- les opérations de traitement sur ces structures (création, formatage, assemblage, extraction),
- les formats d’échanges (fichiers, textuels, binaires),
- les connecteurs pour les interfaces (réseaux, stockage, visualisation, traitement)

Il présente enfin quelques exemples et principes de mise en œuvre.

## 2 CAS D'USAGE ET BENEFICES

Ce chapitre illustre à partir d'exemples les bénéfices d'une interopérabilité.

### 2.1 MESURE D'EXPOSITION – INTEROPERABILITE HORIZONTALE

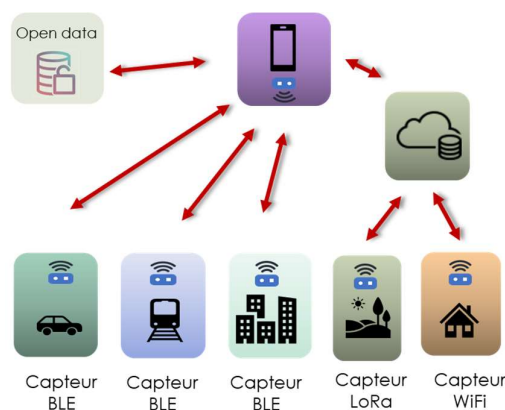
L'exposition d'une personne à un paramètre environnemental nécessite une mesure de cette exposition pour toutes ses activités quotidiennes (au domicile, dans les transports, sur son lieu de travail, dans les lieux publics, en extérieur).

Exemple de cas d'usage :

*Un scénario de mesure de cette exposition pourrait par exemple être le suivant :*

- *J'installe des capteurs à mon domicile (intérieur, extérieur),*
- *Mon véhicule est équipé par le constructeur d'un capteur,*
- *Les lieux publics mettent à disposition les informations (issues de capteurs ou de modélisation),*
- *Pour les lieux extérieurs non couverts par des capteurs, je prends en compte les données open-data disponibles*

*Aujourd'hui, aucune solution ne couvre l'ensemble de ce scénario et l'absence d'interopérabilité entre capteurs et applications nous interdit un « mixage » de solutions.*



La mise en œuvre de connecteurs standards permet de répondre à ce type de scénario :

- Les capteurs Bluetooth disposés dans les véhicules ou les lieux publics fournissent les données suivant le service Bluetooth ESS (Environmental Sensing Service) mis à niveau en septembre 2021.

- Les capteurs individuels utilisant un réseau LPWAN (extérieur) ou TCP/IP (intérieur) envoient leurs données au format standard défini,
- Elles sont stockées sur une base de données respectant le format standard,
- Une application indépendante à la fois des capteurs et des dispositifs de stockage accède à l'ensemble des données émises via les connecteurs définis.
- Les traitements applicatifs effectués sont également indépendants de la nature des données traitées.

## 2.2 CYCLE DE VIE DES DONNEES – INTEROPERABILITE VERTICALE

L'exploitation des données environnementales nécessite de pouvoir gérer l'ensemble du cycle de vie (acquisition / transmission / stockage / traitement / analyse / restitution).

Exemple de cas d'usage :

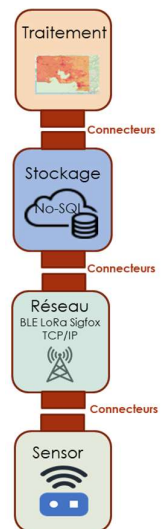
*Je mets en place un réseau citoyen de mesures environnementales, pour cela, je communique aux personnes de mon réseau :*

- Quelques références de capteurs à acquérir en fonction des réseaux retenus,
- Les coordonnées de la base de données partagée que j'ai créée (configuration Cloud standard),
- Quelques références d'applications de restitution des informations.

*Sur ce périmètre, les solutions proposées actuellement sont « fermées », ou bien ne couvrent qu'une partie du cycle de vie.*

Les connecteurs proposés permettent de faire communiquer les principaux composants :

- Les capteurs intègrent le connecteur permettant de générer la structure de données,
- Les connecteurs réseaux assurent le codage / décodage suivant le format d'échange propre au réseau,
- Les connecteurs base de données stockent les données selon le format de la base de données et les restituent suivant les API adaptés,
- Les connecteurs de traitement génèrent les structures de données adaptées aux traitements à réaliser et produisent les informations utilisateurs souhaitées



## 2.3 ASSIMILATION DE DONNEES – INTEROPERABILITE HORIZONTALE

Les données environnementales sont disponibles sous différents formats, à différentes échelles spatiales ou temporelles. L'intégration de ces données est souvent complexe et nécessite une expertise et des outils spécifiques.



Exemple de cas d'usage :

*Je souhaite connaître le niveau de concentration d'un polluant donné au voisinage de mon domicile. Je dispose :*

- *D'une valeur moyenne annuelle sur ma région par maille de 100m x 100m (open-data)*
- *De relevés de mesures horaires de plusieurs stations fixes situées entre 1 km et 10 km (open data)*
- *De mesures ponctuelles relevées lors de mes déplacements alentours mais avec une précision très faible (capteur mobile)*

*Ces données sont de natures différentes. Il n'existe pas d'outils simples permettant d'assembler ces données et d'en extraire des informations de synthèse.*

La structure de données proposée permet de représenter au sein d'un même objet ces trois types d'informations. Les techniques d'analyse disponibles peuvent alors être appliquées automatiquement à cette structure. Dans le cas présent, l'utilisation d'un procédé « d'assimilation de données » permet de fournir la réponse optimale à la question posée.

## 2.4 BENEFICES ATTENDUS

La mise en œuvre de la structure de données et des connecteurs associés permet d'obtenir une interopérabilité à différents niveaux :

- Interopérabilité matérielle entre les équipements d'acquisition, les équipements réseaux et les équipements de stockage,
- Interopérabilité entre données de nature différente,
- Interopérabilité des outils d'analyse et de traitement de ces données

Une première segmentation permet d'identifier les bénéfices suivants :

- Chaîne d'acquisition (capteurs, micro-contrôleurs, réseau) :
  - Compte tenu de l'absence de standardisation des données, chaque contributeur de cette chaîne traite les données dans un format spécifique. Ceci oblige de mettre en œuvre des fonctions de codage/décodage des informations recues.
  - Certains fournisseurs étendent leur périmètre à des plateformes d'accès à leurs données,
  - **L'intégration de connecteurs standard permet de banaliser les équipements et leur intégration dans un système complet.**
- Plateformes IoT :
  - Les plateformes généralistes proposent de nombreux services couvrant l'ensemble des fonctions liées aux données mais n'offrent pas de fonctions directement utilisables pour les données environnementales (absence de standards).

- Les plateformes spécialisées qui s'appuient sur ces plateformes généralistes, des outils open-source ou encore sur des outils de plus bas niveaux, proposent une structuration propriétaire et développent de nombreuses interfaces.
- Les plateformes internationales dédiées aux enjeux climatiques utilisent des outils spécialisés liés aux standards géomatiques (OGC). Ces outils sont complexes et ne peuvent être déclinés pour des besoins spécifiques.
- **L'intégration de connecteurs standard intégrés aux plateformes du marché permet de réduire le nombre d'interfaces et de mettre en place des traitements sur des structures de données de plus haut niveau.**
- Editeurs de solutions de conception ou d'aménagement :
  - Les éditeurs (ex. Dassault Systèmes, Autodesk) s'appuient sur des structures de données internes,
  - Des standards existent pour faciliter l'interopérabilité entre éditeurs (ex. IFC : Industry Foundation Classes),
  - **L'intégration dans les IFC du standard défini permet de déployer les connecteurs standards dans les solutions de conception et d'aménagement.**
- Intégration applicative :
  - De nombreux acteurs sont présents sur ce marché en ciblant des cas d'usage identifiés. Ces acteurs utilisent les services apportés par les fournisseurs de capteurs ou de plateformes pour mettre en place des solutions spécifiques.
  - **L'intégration de connecteurs standard permet d'élargir le périmètre des solutions proposées tout en en réduisant le coût de mise en oeuvre.**
- Exploitation de données (acteurs publics, privés ou citoyens) :
  - Les gestionnaires / exploitants de données environnementales construisent, exploitent et mettent à disposition des ensembles de données toujours plus étendus.
  - Ils intègrent des passerelles vers les solutions de leur structure (ex. SIG pour les métropoles, systèmes de monitoring et de pilotage) et vers les acteurs locaux et régionaux (ex. open data)
  - **L'usage de solutions standardisées et open-source est un élément de politique technique qui permet de faciliter l'intégration et l'agrégation de nouvelles données.**
- Activités d'étude et d'aménagement :
  - Les acteurs de ces secteurs sont confrontés aux difficultés d'accès à des données fiables et précises et également à l'absence d'interopérabilité des données échangées.
  - **La mise en place de standard de données de plus haut niveau permet d'obtenir des données mieux structurées et de faciliter les échanges.**
- Citoyen et utilisateur final :
  - Les utilisateurs finaux sont confrontés à l'éparpillement des solutions proposées et sont demandeurs d'interopérabilité et d'accès à des données plus complètes et de plus haut niveau que celles accessibles aujourd'hui.

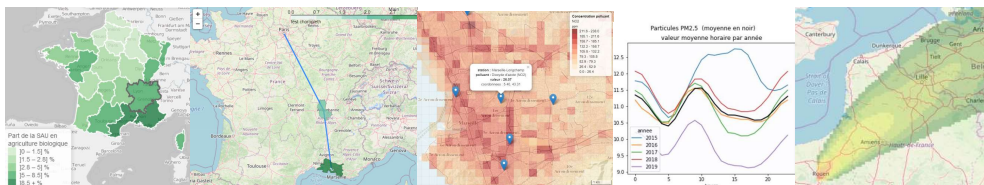
- **La solution proposée répond à cette attente.**

### 3 PRESENTATION

#### 3.1 PRINCIPES GENERAUX

La structure de données mise en place est issue du concept d'« *Observation* » tel qu'il est défini dans la norme ISO19156. Elle permet de représenter par exemple :

- Des données unitaires issues de capteurs,
- Des résultats de modélisation,
- Des répartitions géographiques,
- Des historiques temporels ou sur un trajet,
- ...



Une Observation est représentée sous la forme d'un objet informatique unique. Celui-ci peut alors être traité ou bien exporté sous différents formats neutres sans avoir à adapter les traitements à la nature des données qui composent cet objet.

#### Standard

La structuration retenue s'appuie notamment sur les standards et outils suivant :

- ISO 19156:2011 - Observations and Measurement (TC211 - géomatique)
- Référentiel OGC - SWE, (en particulier SensorThings API Part 1: Sensing, SensorML, Coverage, WaterML Part 1: TimeSeries) et ODM2 : Observation Data Model
- Format de données textuelles JSON, GeoJSON
- TIFF 6.0, NetCDF pour les formats de fichiers
- Format de données binaires : Bluetooth (Environmental Sensing Service)
- Structures Xarray, Pandas et GeoPandas de traitement des données multi-dimensionnelles

Le travail de mise en cohérence mené avec Bluetooth SIG rend maintenant celui-ci compatible avec cette structuration (intégration des données air à l'« Environmental Sensing Service » adoptée le 14 septembre 2021 : <https://www.bluetooth.com/specifications/specs/gatt-specification-supplement-5/>).

#### Cas d'usage

La structure retenue est permissive et permet de traiter tous les niveaux de complexité et de complétude liés aux dimensions définies. Elle reste également ouverte sur la nature des mesures effectuées.

### Interopérabilité

La solution retenue est applicable au niveau de :

- Equipements à faible complexité logicielle (ex. sensor) : solution accessible en C++ avec un nombre restreint de bibliothèques externes,
- Différents types de réseaux (LoRa, Bluetooth, SigFox, TCP/IP) : les données sont utilisables sous format binaire (LoRa, Bluetooth, SigFox ) et sous format textuel JSON (TCP/IP)
- Des systèmes de traitements de données : La structure proposée est compatible avec les structures de haut niveau (ex. Xarray, Pandas, GeoPandas)
- Base de données : L'intégration est native avec des bases de données NoSQL (ex. Mongo DB)
- Fichiers : Les principaux formats de fichiers sont pris en compte (ex. Tiff / geoTiff, csv, NetCDF, Shape file).

### Modularité

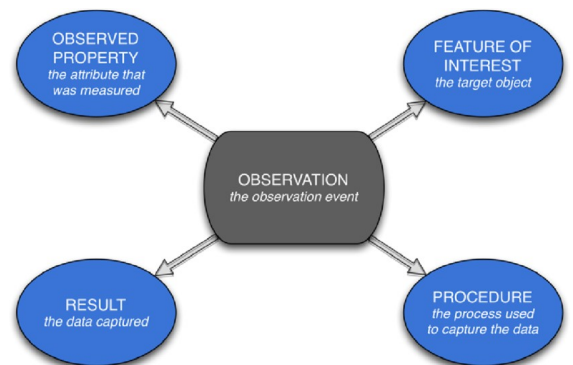
Plusieurs mécanismes sont mis en place pour intégrer de nouveaux types de données et de nouvelles fonctionnalités :

- Ajout de « classes filles » à des classes déjà existantes,
- Ajout de données dans les dictionnaires,
- Utilisation de classes d'objets dédiées à des usages spécifiques,
- Ajout de connecteurs open-source ou privés.

## 3.2 STRUCTURE DE DONNEES

Une Observation est caractérisée par :

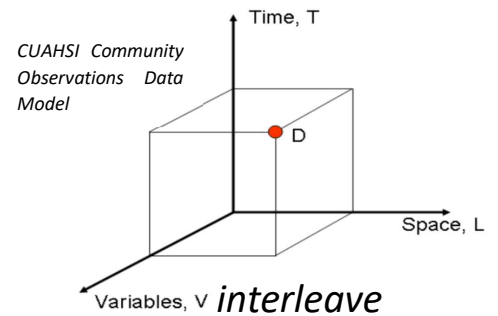
- "observed property" : la propriété observée,
- "feature of interest" : l'objet (le plus souvent un lieu) de l'observation,
- "procedure" : le mode d'acquisition de l'information (capteur, modèle...)
- "result" : résultat de l'observation ou de la mesure



Le résultat est un ensemble de valeurs référencées selon 3 dimensions :

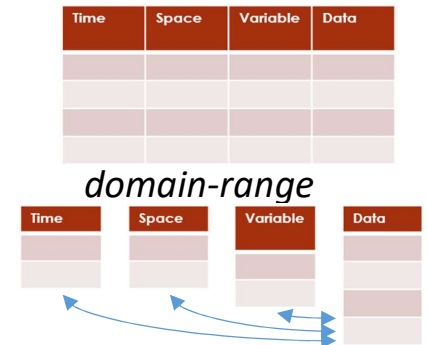
- Une dimension temporelle,
- Une dimension spatiale,
- Une dimension physique (propriété observée)

Il est représenté par une matrice à 3 dimensions, chaque dimension étant indexée par des valeurs temporelles, spatiales et physiques.



*Nota : Cette représentation matricielle « domain range » est préférée à une représentation tabulaire « interleaved » qui associe à chaque valeur du résultat les valeurs temporelles, spatiales et physiques.*

Des propriétés communes (indicateurs) sont associées à chaque Observation. Elles permettent d'effectuer des traitements sur les Observations sans avoir à en connaître la composition (ex. boîtes englobantes, type d'observation, volumétrie...).



### 3.3 OPERATIONS

Une Observation peut être décrite partiellement et faire l'objet de compléments ultérieurs. On distingue plusieurs types d'opérations :

Les opérations directement liées aux Observations

- Fonctions de construction (création, modification, fusion, restriction).
- Fonctions de structuration (Indexation, normalisation)
- fonctions d'analyse (complétude, typologie, indicateurs).

Les opérations liées à des formats de données

- Format textuel : format JSON
- Format binaire : format Bluetooth, LoRa
- Format fichier : TIFF, GeoTIFF, NetCDF, Shape file
- Format objet : Xarray, GeoPandas

Les opérations liées au traitement de ces données

- Recherche et extraction,
- Ajustement à un point de vue (granularité spatiale et temporelle, vue suivant des dimensions spécifiques),
- Traitements multiples et globaux (statistique, machine learning, assimilation)



### 3.4 CONNECTEURS

Les connecteurs sont les interfaces entre un objet Observation et une application ou un format d'échange.

La liste des connecteurs est extensible en fonction des besoins et concerne :

- Les capteurs,
- Les réseaux,
- Les formats neutres
- Les bases de données
- Les traitements de restitution (ex. visualisation)
- Les applications spécifiques (en entrée / sortie)
- Les modules d'analyse et de retraitement

### 3.5 UTILISATION

Les résultats d'une Observation peuvent être de différentes natures (ex. valeur numérique, texte, données structurées).

Les cas d'usage sont multiples et découlent des trois dimensions spatiales, temporelles et physiques.

On distingue notamment les usages suivants :

| resultat | location | datation | usage   | type             |
|----------|----------|----------|---|------------------|
| aucune   | aucune   | aucune   | configuration : données de paramétrage  | config           |
|          |          | unique   | top d'indication de fonctionnement  | top              |
|          |          | multiple | multi-top d'information   | multiTop         |
|          | unique   | aucune   | geo localisation de l'émetteur  | point            |
|          |          | unique   | localisation ponctuelle de l'émetteur (tracking)  | track            |
|          |          | multiple | localisation fixe   | fixedTrack       |
|          | multiple | aucune   | zoning : délimitation d'une zone  | zoning           |
|          |          | unique   | localisation d'un device réparti  | multiTrack       |
|          |          | multiple | zone spatiotemporelle (déformable dans le temps)<br>localisations multiples sur un trajet | timeLoc          |
| unique   | aucune   | aucune   | information sans dimension spatiale ou temporelle   | measure          |
|          |          | unique   | relevé de valeur  | record           |
|          |          | multiple | relevé de valeur moyen  | meanRecord       |
|          | unique   | aucune   | caractéristique locale  | feature          |
|          |          | unique   | mesure ou information ponctuelle  | obsUnique        |
|          |          | multiple | mesure moyenne d'un capteur fixe  | obsMeanFixed     |
|          | multiple | aucune   | caractéristique d'une zone géographique   | areaFeature      |
|          |          | unique   | mesure moyenne sur une zone / grille (device réparti)                                     | obsMeanArea      |
|          |          | multiple | mesure moyenne sur un trajet  | meanTimeLoc      |
|          |          | multiple | mesure moyenne sur une zone dans le temps   |                  |
| multiple | aucune   | aucune   | serie de mesures sans contexte  | multiMeasure     |
|          |          | unique   | mesure échantillonnée non localisée   | multiRecord      |
|          |          | multiple | historique de mesures   | measureHistory   |
|          | unique   | aucune   | mesures locales variables   | featureVariation |
|          |          | unique   | mesure échantillonnée ponctuelle et localisée   | obsSampled       |
|          |          | multiple | séquence de mesure d'un capteur fixe  | obsSequence      |
|          | multiple | aucune   | mesure sur une zone (device réparti)  | measureLoc       |
|          |          |          | mesure sur une grille (device réparti)  |                  |
|          |          |          | mesure sur un profil vertical   |                  |
|          |          | unique   | observation sur une zone (device réparti)   | obsLoc           |
|          |          |          | observation sur une grille (device réparti)   |                  |
|          |          |          | observation sur un profil vertical  |                  |
|          |          | multiple | observation d'un trajet   | obsTimeLoc       |
|          |          |          | observation sur une zone dans le temps  |                  |
|          |          |          | observation sur un profil vertical dans le temps  |                  |
|          |          |          | observation d'une grille dans le temps  |                  |

### 3.6 EXEMPLES

Pour plus de lisibilité, les exemples sont donnés dans le format JSON.

#### 3.6.1 Mesure simple

**Format réduit :**



```
{
  "type": "observation",
  "dateTime": "2021-05-04T12:05:00",
  "coordinates": [14, 42.2],
  "propertyList": { "PropertyType": "PM10", "unit": "ppm" },
  "value": 120
}
```

### Format complet (informations générées) :

```
{
  "type": "observation",
  "attributes": {
    "typeobservation": "obsUnique",
    "datation": {
      "typedatation": "date",
      "dateTime": "2021-05-04T12:05:00"},
    "location": {
      "typelocation": "point",
      "coordinates": [14, 42.2]},
    "property": {
      "typeproperty": "list",
      "propertyList": {"PropertyType": "PM10", "unit": "ppm"}},
    "result": {
      "typeresult": "real_int_str",
      "value": [120, [0,0,0]]},
    "information": {
      "nvalloc": 1,
      "nvaldat": 1,
      "nvalprop": 1,
      "nvalres": 1,
      "boundingBoxMin": [14, 42.2],
      "boundingBoxMax": [14, 42.2],
      "timeBoxMin": "2021-05-04T12:05:00",
      "timeBoxMax": "2021-05-04T12:05:00",
      "complet": true,
      "score": 111,
      "tauxMeasure": 1,
      "tauxEchantillon": 1,
      "dim": 0,
      "nEch": 1}
  }
}
```

### 3.6.2 Mesure sur un trajet

Exemple d'une mesure en 3 point sur un trajet

#### Format réduit :

```
{
  "type": "observation",
  "dateTime": [
    "2021-05-04T12:05:00",
    "2021-07-04T12:05:00",
    "2021-02-04T12:05:00"
  ],
  "coordinates": [
    [14, 42.2],
    [24, 22.9],
    [20, 25]
  ],
  "propertyList": { "PropertyType": "PM10"},
  "value": [45, 5, 40]
}
```

### 3.6.3 Mesure multiple

Exemple d'une mesure en 3 point à 3 instants différents

**Format réduit :**

```
{
  "type": "observation",
  "dateTime": [
    "2021-05-04T12:05:00",
    "2021-07-04T12:05:00",
    "2021-02-04T12:05:00"
  ],
  "coordinates": [
    [14, 42.2],
    [24, 22.9],
    [20, 25]
  ],
  "propertyList": {"PropertyType": "PM10"},
  "value": [5, 25, 50, 4, 20, 40, 2, 10, 20]
}
```

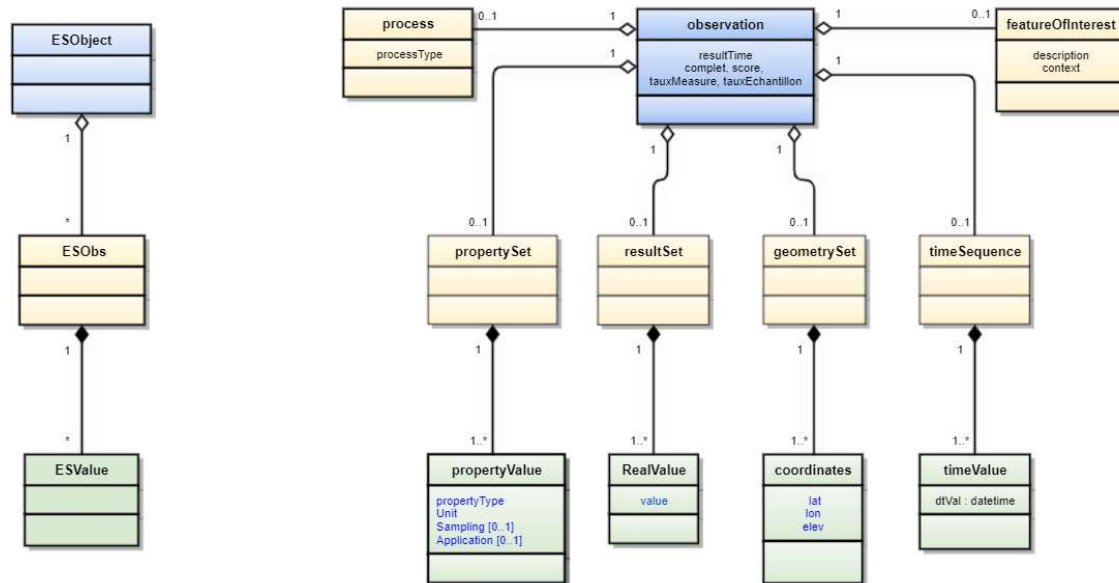
**Format Xarray :**

```
<xarray.Dataset>
Dimensions: (loc: 3, prop: 1, time: 3)
Coordinates:
  lon      (loc) float64 42.2 22.9 25.0
  lat      (loc) float64 14.0 24.0 20.0
  * loc     (loc) <U12 '[14.0, 42.2]' '[24, 22.9]' '[20, 25]
  * time    (time) datetime64[ns] 2021-05-04T12:05:00 ... 2021-02-04T12:05:00
  timestr   (time) <U19 '2021-05-04T12:05:00' ... '2021-02-04T12:05:00'
  * prop     (prop) object {'PropertyType': 'PM10'}
Data variables:
  value     (time, loc, prop) float64 5.0 25.0 50.0 4.0 20.0 40.0 2.0 10.0 20.0
```

## 4 STRUCTURATION

### 4.1 MODELE DE DONNEES

La structure des observations est la suivante :



Une Observation (EObject) est une agrégation d'objets représentant chacun une caractéristique de cet objet (ESObjs).

Ces objets sont eux-mêmes composés d'éléments (ESValue) correspondant aux valeurs prises par les caractéristiques ESObjs. L'exemple à droite du schéma illustre ces principes.

Le modèle général des classes d'objets est fourni en annexe.

#### 4.1.1 EObject

Cette classe ne comporte qu'un seul objet : Observation.

Celui-ci est constitué des attributs correspondants aux caractéristiques externes de l'Observation ainsi que des liens vers les objets ESObjs qu'il agrège :

- Name : nom de l'observation effectuée
- Process : description du mode d'acquisition des données
- Interest : objet physique de l'observation (Feature of Interest)
- RelatedObsId : Observation de référence (informations non dupliquées)
- Property : propriétés qui font l'objet d'une évaluation
- Result : résultats des propriétés évaluées
- Location : localisation spatiale des évaluations effectuées

- Datation : localisation temporelle des évaluations effectuées
- Parameter : Informations spécifiques

Il peut représenter aussi bien des objets simples de dimension 0 (ex une mesure unique ponctuelle) que des objets de dimensions 5 (3 dimensions au niveau spatial + 1 dimension temporelle + 1 dimension liée aux propriétés observées).

Une entité « Location » peut être composée de points ou de polygones, ce qui permet par exemple de représenter l'évolution d'un panache de fumée qui se transforme et se déplace ou encore d'identifier des zones géographiques (ex découpage cartographique administratif).

De même, une entité « Datation » peut être composée d'instantanés ou d'intervalles, ce qui permet de distinguer des mesures instantanées de mesures sur une durée.

#### 4.1.2 ESObs

Cette classe intègre les objets cités ci-dessus.

Les objets ESObs sont composés des éléments suivants :

- Liens vers les valeurs descriptives observées (ESValue)
- Informations de synthèse (ex. boîte englobante)
- Informations spécifiques

Dans une première version, les objets ESObs pris en compte sont les suivants :

- Location :
  - SetLocation : ensemble d'objets représentés par des coordonnées géométriques (points ou polygones),
  - GridLocation : objet décrivant un découpage spatial matriciel d'une zone
- Datation :
  - SetDatation : ensemble d'objets d'horodatage (instantanés ou intervalles)
  - GridDatation : objet décrivant un découpage temporel linéaire d'un intervalle
- Result :
  - ResultSet : ensemble d'objets de type result
- Property :
  - PropertySet : ensemble d'objets de type property

Ces types d'objets permettent de représenter la majorité des cas d'usage :

- Mesures ponctuelles issues de capteurs fixes ou mobiles,
- Mesures sur des zones non ponctuelles (ex. découpage géographique)
- Relevés matriciels (ex. relevé satellite)
- Mesures sur des intervalles de temps,
- Evolutions complexes (ex. panache de fumée)

Les objets Name, Process, Parameter, Interest, RelatedObsId sont composés d'attributs de type texte.

### 4.1.3 ESValue

Cette classe intègre les objets terminaux des Observations associés aux objets ESObs.

Les objets ESValue sont constitués des éléments suivants :

- Property :
  - PropertyValue : Ensemble de propriétés faisant l'objet d'une évaluation
- Result :
  - IntValue / RealValue / StringValue : valeurs prises en compte dans les évaluations. Ces valeurs disposent également d'un index (coordonnées) permettant de les relier aux autres objets ESObs (location, datation, property)
- Location :
  - LocationValue : Cet objet représente les lieux associés à l'évaluation (coordonnées géodésiques ou projetées) sous forme de points ou bien de polygones représentant des entités surfaciques.
- Datation :
  - DatationValue : Objet représentant un positionnement temporel soit ponctuel soit sous forme d'intervalle

## 4.2 TRAITEMENTS

Ils sont mis à disposition sur le site GitHub hébergeant les informations et développements réalisés (deux langages sont pris en compte actuellement : C++, Python).

On distingue :

- Les fonctions de construction :
  - Création : à partir de composants élémentaires ou bien par structures textuelles au format Json
  - Modification : ajout de données (mécanismes similaires à la création) ou suppression de données
  - Extension : ajout d'une partie des données d'une autre Observation pour simplifier la construction d'une Observation
  - Addition : L'addition de deux Observations consiste à cumuler les résultats des deux Observations sous un même format
  - Réduction : Fonction inverse de réduction du nombre de résultats par filtrage
- Les fonctions de structuration :
  - Indexation / désindexation : représentation sous forme matricielle ou « à plat »

- Complétion : ajout de données résultats vides (null ou nan) pour disposer d'un jeu de résultat complet
  - Normalisation : suppression des données inutiles et ordonnancement
- les fonctions d'analyse :
  - Mesure de complétude (taux de mesure, taux d'échantillonnage)
  - Typologie d'observation (cas d'usage, dimension)
  - Indicateurs (nombre de valeurs)
  - Boîtes englobantes (temporelle ou spatiale)
- Les interfaces :
  - Format textuel : format JSON, GEOJSON, csv, IFC
  - Format binaire : format Bluetooth, LoRa
  - Format fichier : TIFF, GeoTIFF, Shapefile
  - Format objet : Xarray
- Les connecteurs :
  - LoRa, SigFox
  - Bluetooth,
  - RestFul
  - MongoDB
  - Pandas, Numpy, Xarray
  - LeafLet
  - ...

La liste des connecteurs est extensible en fonction des besoins.

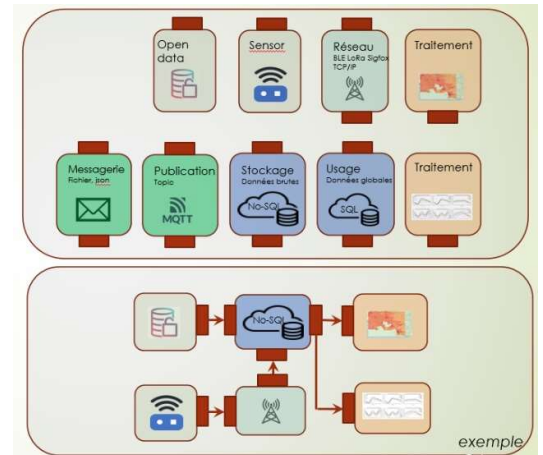
## 5 MISE EN ŒUVRE

### 5.1 PRINCIPES

L'utilisation des connecteurs permet de s'affranchir de la façon dont les données sont codées, ce codage étant masqué au sein de l'objet « Observation ».

Le partage et l'accès aux données s'effectue alors par un assemblage de connecteurs.

Le chapitre ci-dessous illustre cette mise en œuvre sur quelques exemples.



### 5.2 EXEMPLES

#### 5.2.1 Capteur

L'exemple ci-dessous illustre un capteur mobile qui enregistre une information (ex. température) de façon périodique et la transmet dès qu'il est connecté à un réseau.

Le capteur (exemple : microcontrôleur programmable en C++) utilise les fonctions suivantes :

```
Observation obs = Observation(); // création d'un objet observation vide
int nprop1 = obs.init(PropertyValue("Temp", "°C")); // ajout d'une propriété correspondante aux mesures à effectuer

for (int i = 0; i < 6; i++) { // simule une boucle de mesure
    int nres = obs.addValueSensor(RealValue(25+i), TimeValue(2021, 5, 4+i, 12, 5, 0), Coordinates(14+i, 42), nprop1);
    // ajoute à l'objet obs une mesure constituée d'une valeur mesurée, de l'instant de mesure et des coordonnées
    // et l'associe à la propriété prop1
}
```

Le capteur envoie ensuite l'observation constituée des différentes mesures (ex format json pour un envoi via API Rest) :

```
string json_a_envoyer = obs.json();
```

Composition de la donnée Json générée :

```
{"type":"observation","propertyList":{"property": "Temp", "unit": "deg C"},"coordinates":[[14, 42], [15, 42], [16, 42], [17, 42], [18, 42], [19, 42]],"dateTime":["2021-05-04T12:05:00", "2021-05-05T12:05:00", "2021-05-06T12:05:00", "2021-05-07T12:05:00", "2021-05-08T12:05:00", "2021-05-09T12:05:00"],"value":[25, 26, 27, 28, 29, 30]}
```

#### 5.2.1 Stockage en Base de données

La base de données doit pouvoir stocker tout type d'observation correspondant aux différents cas d'usage. C'est le cas des bases « NoSQL » de type « document » qui stockent directement des formats Json.

A titre d'exemple, l'envoi d'une Observation dans une base MongoDB s'effectue très simplement par une requête HTTP POST (ex. Python ci-dessous) :

```
url = "https://webhooks.mongodb-realm.com/api/client/v2.0/app/observ
r = rq.post(url, data=obs.json())
```

Pour faciliter les recherches, la donnée Json peut être enrichie d'informations complémentaires (exemple identique au cas ci-dessus) :

```
{"type": "observation", "dateTime": ["2021-05-04T12:05:00", "2021-05-05T12:05:00", "2021-05-06T12:05:00",
"2021-05-07T12:05:00", "2021-05-08T12:05:00", "2021-05-09T12:05:00"], "coordinates": [[14, 42], [15, 42], [16, 42],
[17, 42], [18, 42], [19, 42]], "propertyList": {"property": "Temp", "unit": "deg C"}, "value": [25, 26, 27, 28, 29,
30], "information": {"typeobs": "obsPath", "typeloc": "multipoint", "typedat": "multidate", "typeprop": "list",
"typeres": "multireal_int_str", "nvalloc": 6, "nvaldat": 6, "nvalprop": 1, "nvalres": 6, "boundingBoxMin": [14,
42], "boundingBoxMax": [19, 42], "timeBoxMin": "2021-05-04T12:05:00", "timeBoxMax": "2021-05-09T12:05:00",
"complet": true, "score": 222, "tauxMeasure": 1, "tauxEchantillon": 1, "dim": 1, "nEch": 6}}
```

### 5.2.2 Extraction d'une Base de données

La recherche d'Observation dans une base NoSQL permet d'exploiter plusieurs mécanismes :

- Recherche textuelle (base indexée au format texte)
- Recherche par boîte englobante spatiale ou temporelle (informations boundingbox et timebox)
- Recherche par attributs sur la base des attributs stockés

Le résultat de la recherche est une liste d'Observation qui peuvent être fusionnées en une seule pour faciliter les traitements ultérieurs.

### 5.2.3 Visualisation d'une Observation

Une observation peut être restituée de plusieurs façons :

- représentation 1D : courbe, familles de courbes,
- représentation 2D : surfaces et familles de surfaces,
- représentation 3D : volumes et familles de volumes,
- représentation sur des supports cartographiques,
- représentation sous forme de vidéos temporelles

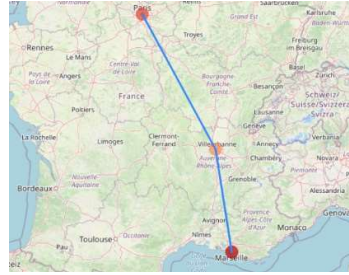
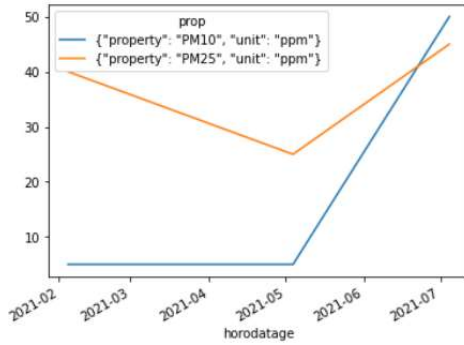
**Exemple d'une trajectoire :**

```
{"type": "observation", "dateTime": ["2021-05-04T00:00:00", "2021-07-04T12:05:00", "2021-02-04T12:05:00"], "coordinates":
[[48.87, 2.35], [45.76, 4.83], [43.3, 5.38]], "propertyList": [{"property": "PM10", "unit": "ppm"}, {"property":
"PM25", "unit": "ppm"}], "value": [5, 25, 50, 45, 5, 40]}
```

La visualisation s'effectue par une simple commande :

```
ob = Observation(text_json)
ob.plot()
```



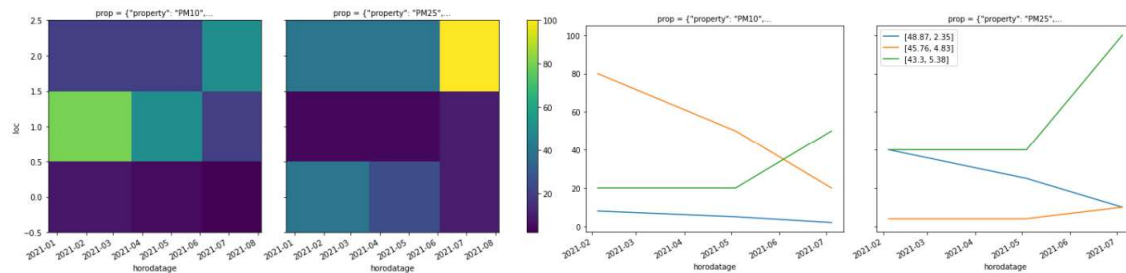


## Exemple d'une grille :

```
{ "type": "observation", "dateTime": [ "2021-05-04T00:00:00", "2021-07-04T12:05:00", "2021-02-04T12:05:00" ], "coordinates": [[ 48.87, 2.35 ], [ 45.76, 4.83 ], [ 43.3, 5.38 ] ], "propertyList": [ { "property": "PM10", "unit": "ppm" }, { "property": "PM25", "unit": "ppm" }, { "value": [ 5, 25, 50, 4, 20, 40, 2, 10, 20, 10, 50, 100, 8, 40, 80, 4, 20, 40 ] } ] }
```

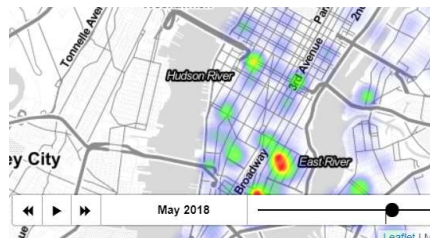
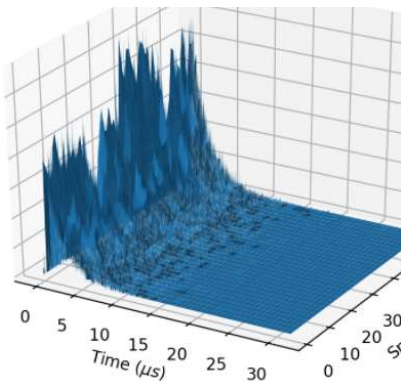
La visualisation s'effectue par la même commande :

```
ob = Observation(text_json)
ob.plot()
```



## Exemple de représentation spatiale :

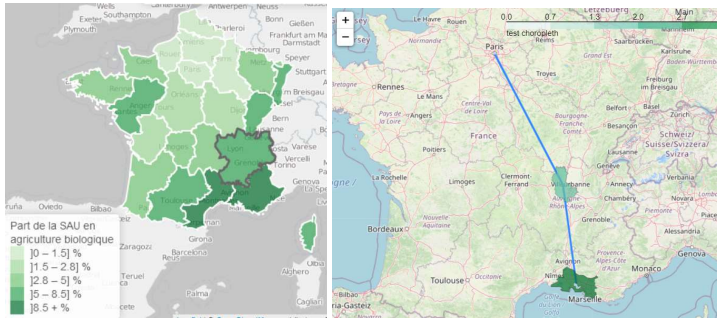
Des représentations spatiales intégrant également la dimension temporelle sont également intégrées (ex. via Pyplot ou bien via Leaflet).



## Exemple d'une choroplèthe :

Il s'agit de représentations intégrant des espaces géographiques définis (ex. ci-dessous avec des espaces correspondant à des découpages administratifs).

La seule différence avec les exemples précédents est de remplacer les points (définis par ses coordonnées) par des polygones (listes de points) délimitant les espaces.



## 6 FAQ

### Quelles sont les données concernées ?

Toutes les données numériques ou alphanumériques associées à un thème et/ou une localisation et/ou une datation. C'est un périmètre très large qui dépasse le cadre des données environnementales.

Cela concerne par ailleurs aussi bien des données unitaires (ex. issues d'un capteur) que des données groupées spatialement (ex. issue d'un relevé satellite) ou temporellement (historique de mesures).

### Pourquoi un nouveau standard alors qu'il en existe déjà ?

Il n'existe pas de standards opérationnels adressant aussi bien des données simples (ex. issues de capteurs) que des données complexes intégrant les trois dimensions (spatiale, temporelle, physique) et couvrant l'ensemble du cycle de vie des données.

Les standards existants sont génériques (ex. ISO « Observation & Measurement ») ou bien sont spécialisés soit sur des données particulières (ex. données géographiques), soit sur des étapes spécifiques du cycle de vie (ex. communication Bluetooth), soit sur des structures de données génériques (ex. Xarray / Pandas) pour les données multi-dimensionnelles), soit sur des outils spécifiques (ex bibliothèque graphique Leaflet).

Il s'agit donc plus de mettre en cohérence les différents standards et de les rendre applicables à tout type de situation. Par exemple, le standard Bluetooth « Environmental Sensing » n'intégrait pas les données liées à la qualité de l'air, celles-ci sont maintenant intégrées.

### Pourquoi ce standard serait-il utilisé ?

C'est lié à une demande d'interopérabilité croissante :

- Il est difficilement compréhensible de devoir accéder à des données environnementales avec des moyens et des applications différentes si l'on se trouve par exemple au domicile, sur son lieu de travail, dans un lieu public ou en pleine nature. La seule réponse à cette attente est de s'appuyer sur un standard de données commun.
- Le standard est intégré dans des « connecteurs » publics et open-source utilisables pour chaque type d'environnement logiciel (ex. C++, Python, javascript), chaque type de réseau LAN, LPWAN, PAN et chaque type de cas d'usage
- Les connecteurs simplifient les activités de développement, d'intégration ou de commercialisation puisqu'ils prennent en charge les opérations fastidieuses

de codage/décodage (ex entre capteur, réseau, base de données, visualiseur, carte, logiciel de modélisation, fichiers...)

### **Pourquoi s'intéresser à des capteurs dont le cout est de quelques euros ?**

Ce sont les matériels qui ont la plus forte contrainte puisqu'ils disposent de fonctions logicielles réduites. Si cela fonctionne sur ce type de matériel, c'est généralisable à tout type de matériel.

Par ailleurs, la majeure partie du cout de mise en œuvre d'un capteur est liée à son intégration dans un système complet qui peut être fortement réduit dans un environnement standardisé.

## 7 ANNEXE

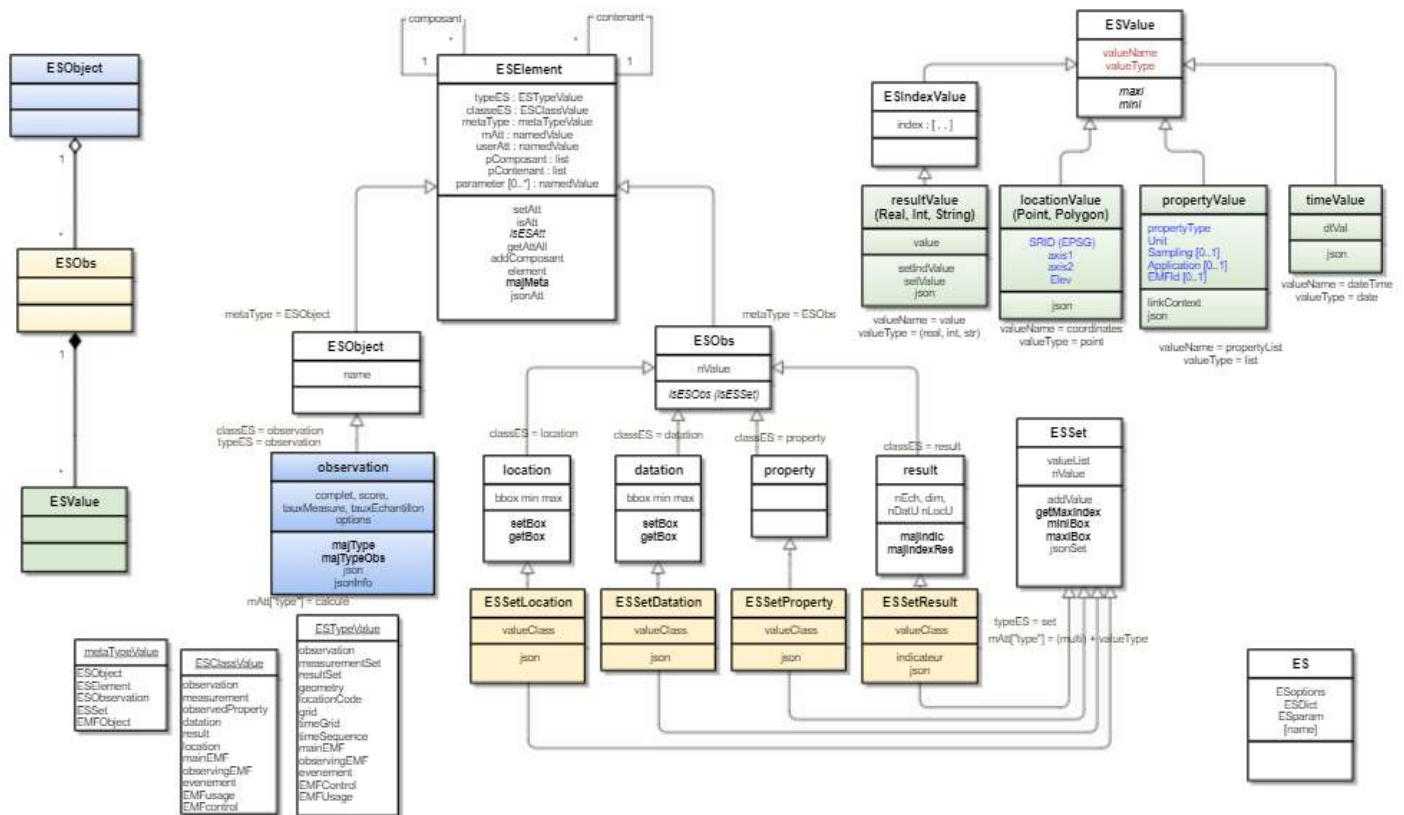
---

## I. Annexe 1 : Modèle de données

Le diagramme ci-dessous représente :

- En blanc : les classes abstraites
- En couleur : Les classes instanciables

### Héritage de classes

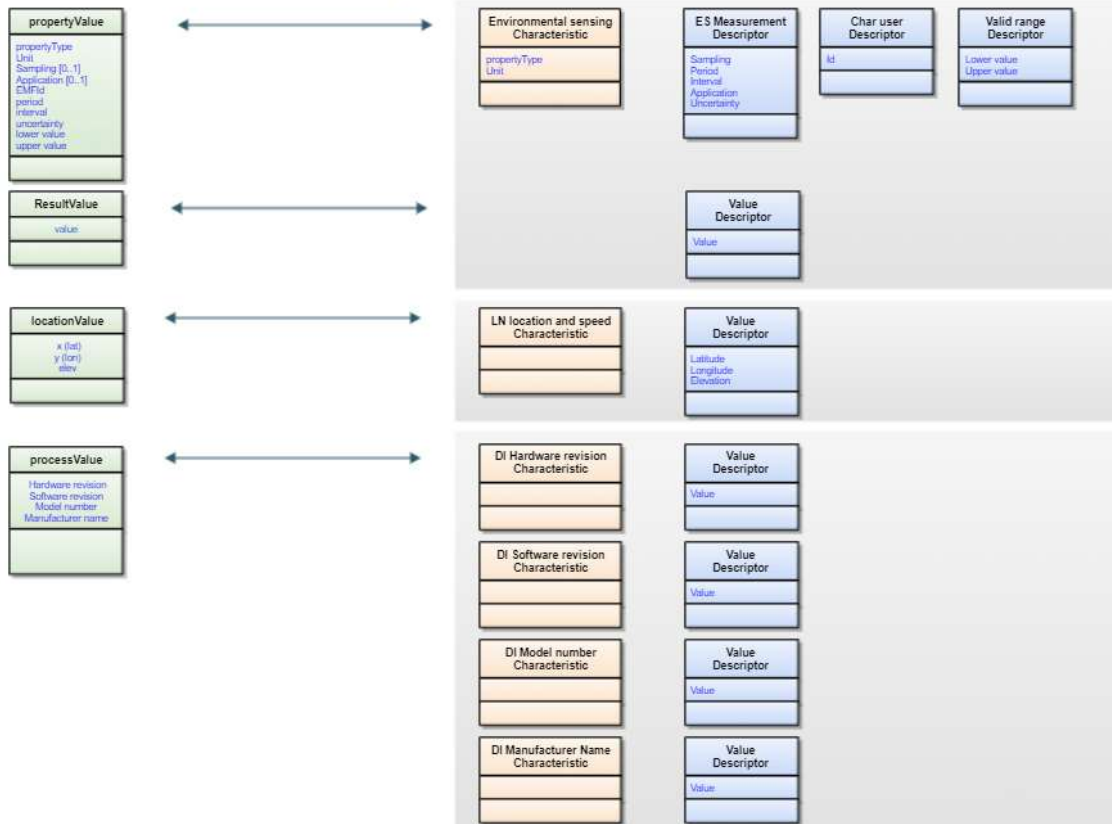


## I. Annexe 2 : Mapping Bluetooth

Le diagramme ci-dessous représente la correspondance du modèle avec les données Bluetooth décrites dans les Services Bluetooth suivants :

- Environmental Sensing Service
- Location and Navigation Service
- Device Information Service

### Mapping Bluetooth



- Chaîne d'acquisition (capteurs, micro-contrôleurs, réseau) :
- Plateformes IoT :
- Intégration applicative :
- Exploitation de données (acteurs publics, privés ou citoyens) :
- Activités d'étude et d'aménagement :
- Citoyen et utilisateur final :