



**INRAE**

**l'institut Agro**  
agriculture • alimentation • environnement



- Développement et intégration d'une API SensorThings au sein d'un écosystème de services numériques pour la science ouverte

Webinaire SensorThings

Réseau SIST & Pôle INSIDE - 29 septembre 2022

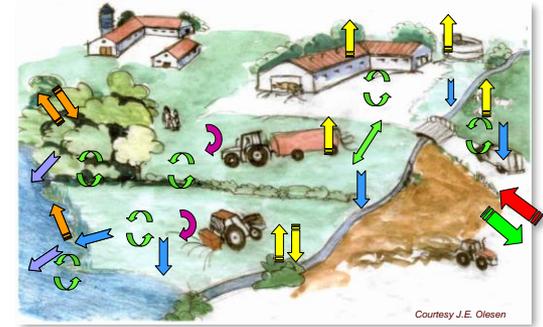
Mario Adam & Hervé Squidant – UMR SAS

# ➤ Plan

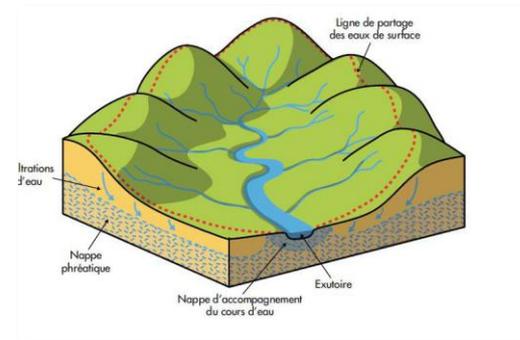
- Contexte
- Développement et déploiement du service SensorThings
- Alimentation du service SensorThings en données
- Les clients SensorThings
- Déploiement effectif
- Retour des utilisateurs
- SDI & STA

# ➤ Contexte : Thèmes ou objets de recherche à l'UMR SAS

- Systèmes Agricoles - Agronomie

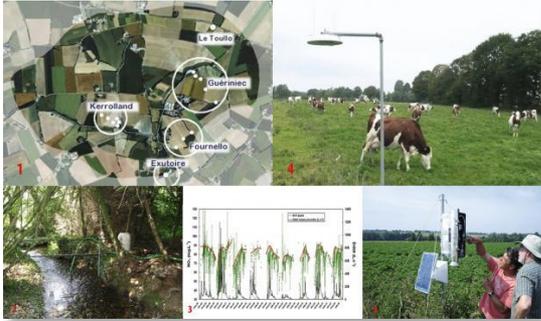


- Science du Sol – Pédologie



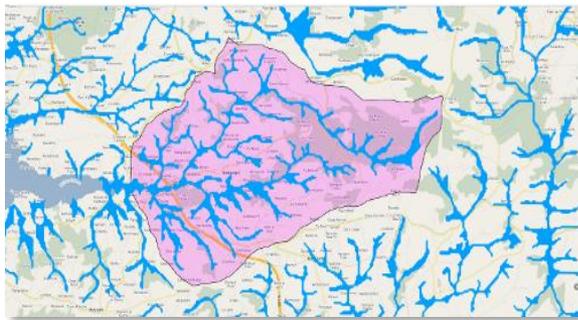
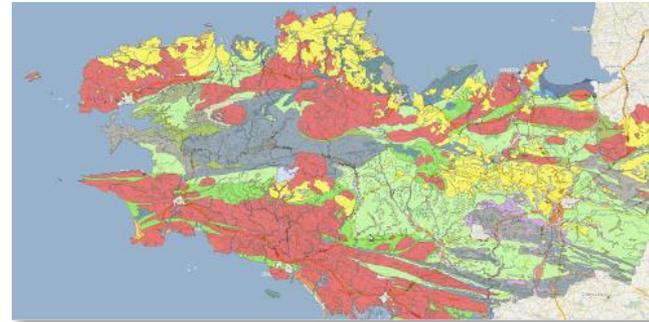
- Hydrologie quantitative & qualitative

# ➤ Contexte : La production de données à l'UMR SAS



**Observatoires labélisés (ORE – SOERE PRO)**

**Référentiels pédologiques**

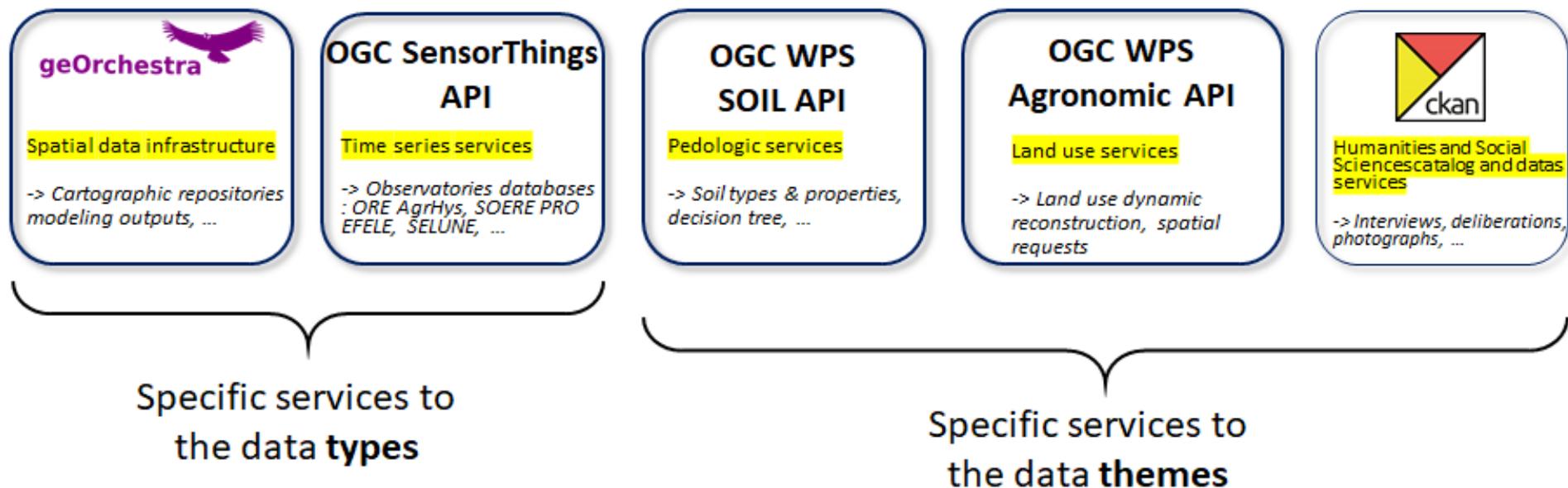


**Sorties de la modélisation hydrologique**

# Contexte : Une approche « full services » pour la diffusion données

- Volonté de contribuer à **l'écosystème open data** conformément aux principe d'INSPIRE (2007) et du FAIR (2016).
- Développement et déploiement de services de données **interopérables**, **modulaires** et reposant sur des **logiciels libres**.

→ Data as a Service (DaaS)



# ➤ Contexte : ... pour alimenter des services de modélisation

- Reposant sur les mêmes principes (**interopérables, modulaires, logiciels libres**).
  - Pour diffuser des traitements, des modèles, des outils d'aide à la décision.
- ➔ Software as a Service (SaaS)

**WPS OGC  
API MNTSurf**

Brique élémentaire de modélisation hydrologique

-> MNT, modèle de drainage, variables dérivées, indice de Beven...

95	97	96
96	100	98
101	102	104

**WPS OGC  
API SIMFEN**

Modèle hydrologie quantitative

-> Calcul débit, réseau hydrologique, bassins non jaugés

**WPS OGC  
API Sol-AID**

Aide à la prescription de la fertilisation azotée

-> Calcul MH

$Mh = Vp \cdot Jn$

Vp = vitesse potentielle de minéralisation. Elle dépend:  
 ◊ Sol  
 ◊ Histoire culturale

Jn = jours normalisés. Ils dépendent:  
 ◊ Climat  
 ◊ Sol  
 ◊ Culture de l'année

...

**Autres Services de calcul**

Autres traitements, modèles, AOD

Ex : modèle TNT<sup>2</sup>, ...



**LEGO de services**

# > Pourquoi développer notre service « maison » ?

L'intégration des capteurs Lora (poster directement le payload dans l'API), la possibilité de gérer plusieurs bases sur le même service, la possibilité d'ajouter nos propres besoins comme la représentation graphique dans l'API nous ont poussé à développer notre propre solution sous NodeJS en TypeScript.



Le STAE-NO (SensorThings outils compatible soit assuré. Api Enhance Node) est né. L'API est 100% compatible avec la norme afin que l'utilisation des

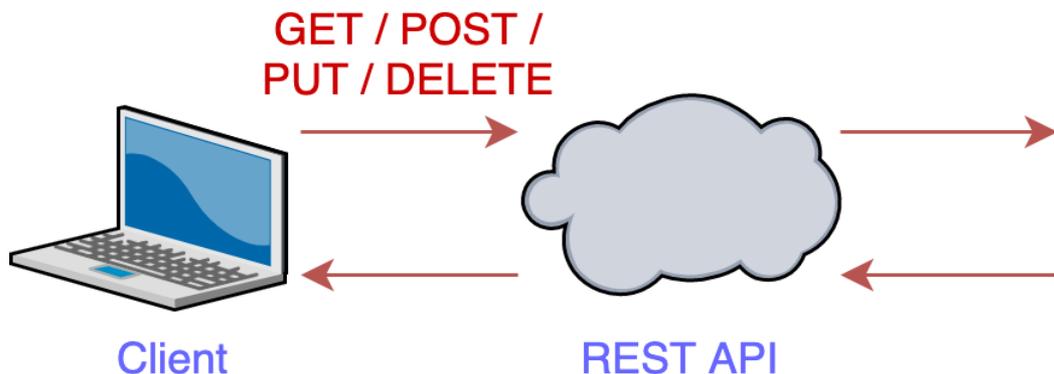
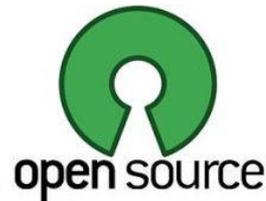
## > Comment

Un serveur avec NodeJS, un fichier de configuration et c'est tout.  
Une version docker sera disponible courant octobre.

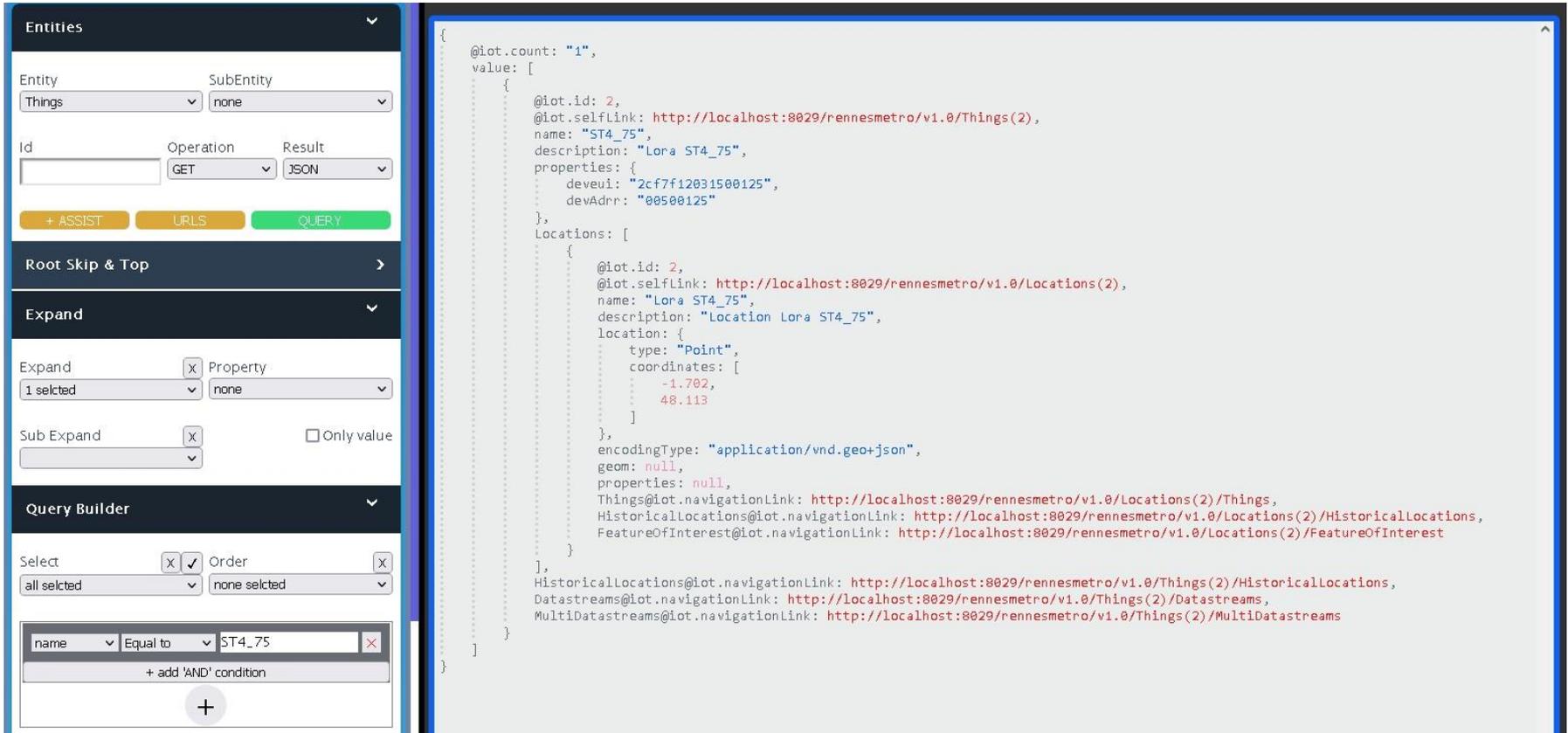


# ➤ La stack

# STAE-ND



# ➤ Outil intégrer a dans l'API (route /Query)



The image shows a web interface on the left and a JSON response on the right. The interface is divided into several sections: 'Entities', 'Root Skip & Top', 'Expand', and 'Query Builder'. The 'Query Builder' section shows a query condition: 'name Equal to ST4\_75'. The JSON response on the right is a complex object with nested arrays and objects, including fields like '@iot.count', '@iot.id', '@iot.selfLink', 'name', 'description', 'properties', and 'Locations'.

```
{
  @iot.count: "1",
  value: [
    {
      @iot.id: 2,
      @iot.selfLink: http://localhost:8029/rennesmetro/v1.0/Things(2),
      name: "ST4_75",
      description: "Lora ST4_75",
      properties: {
        deveuil: "2cf7f12031500125",
        devAddr: "00500125"
      },
      Locations: [
        {
          @iot.id: 2,
          @iot.selfLink: http://localhost:8029/rennesmetro/v1.0/Locations(2),
          name: "Lora ST4_75",
          description: "Location Lora ST4_75",
          location: {
            type: "Point",
            coordinates: [
              -1.702,
              48.113
            ]
          },
          encodingType: "application/vnd.geo+json",
          geom: null,
          properties: null,
          Things@iot.navigationLink: http://localhost:8029/rennesmetro/v1.0/Locations(2)/Things,
          HistoricalLocations@iot.navigationLink: http://localhost:8029/rennesmetro/v1.0/Locations(2)/HistoricalLocations,
          FeatureOfInterest@iot.navigationLink: http://localhost:8029/rennesmetro/v1.0/Locations(2)/FeatureOfInterest
        }
      ],
      HistoricalLocations@iot.navigationLink: http://localhost:8029/rennesmetro/v1.0/Things(2)/HistoricalLocations,
      Datastreams@iot.navigationLink: http://localhost:8029/rennesmetro/v1.0/Things(2)/Datastreams,
      MultiDatastreams@iot.navigationLink: http://localhost:8029/rennesmetro/v1.0/Things(2)/MultiDatastreams
    }
  ]
}
```

# ➤ Format csv (Importation csv dans un ou plusieurs flux)

The screenshot displays a web application interface. On the left is a sidebar with the following elements:

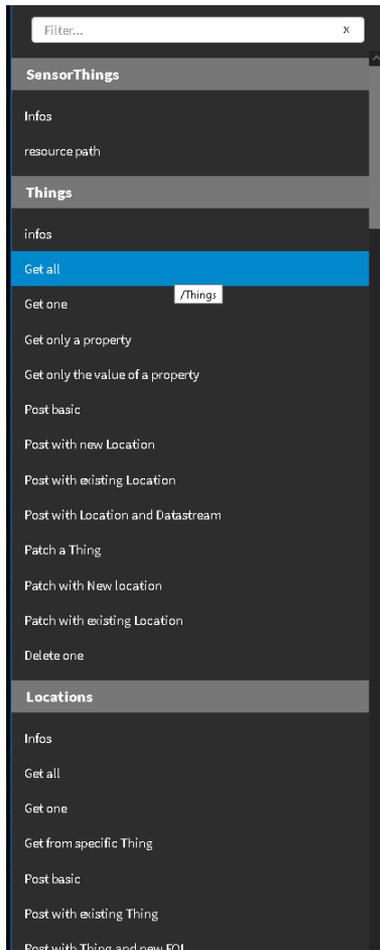
- Entities (dropdown)
- Entity: Things (dropdown), SubEntity: none (dropdown)
- Id: [input field]
- Operation: GET (dropdown), Result: CSV (dropdown)
- + ASSIST (button), URLS (button), QUERY (button)
- Root Skip & Top (dropdown)
- Expand (dropdown)
- Query Builder (dropdown)
- Import File (dropdown)
  - File csv, json... (button)
- Logs (dropdown)
- Assist (dropdown)
- \$resultFormat=CSV (text input)

On the right is a table with the following columns: @IOT.ID, @IOT.SELFLINK, NAME, DESCRIPTION, PROPERTIES, and LOCATIONS@IOT. The table contains 11 rows of data:

@IOT.ID	@IOT.SELFLINK	NAME	DESCRIPTION	PROPERTIES	LOCATIONS@IOT.
1	http://localhost:8029/rennesmetro/v1.0/Things(1)	THR_AGR0_01C02D2D	Station Campbell CS650	{deveui:8cf9574000002d2d,devAddr:00002d2d}	http://localhost:8029/rennesm
2	http://localhost:8029/rennesmetro/v1.0/Things(2)	ST4_75	Lora ST4_75	{deveui:2cf7f12031500125,devAddr:00500125}	http://localhost:8029/rennesm
3	http://localhost:8029/rennesmetro/v1.0/Things(3)	ST4_50	Lora ST4_50	{deveui:2cf7f12031500113,devAddr:00500113}	http://localhost:8029/rennesm
4	http://localhost:8029/rennesmetro/v1.0/Things(4)	ST4_25	Lora ST4_25	{deveui:2cf7f1203150012a,devAddr:0050012a}	http://localhost:8029/rennesm
5	http://localhost:8029/rennesmetro/v1.0/Things(5)	ST3_75	Lora ST3_75	{deveui:2cf7f1202520017e,devAddr:0020017e}	http://localhost:8029/rennesm
6	http://localhost:8029/rennesmetro/v1.0/Things(6)	ST3_50	Lora ST3_50	{deveui:2cf7f12025200178,devAddr:00200178}	http://localhost:8029/rennesm
7	http://localhost:8029/rennesmetro/v1.0/Things(7)	ST3_25	Lora ST3_25	{deveui:2cf7f1202520013b,devAddr:0020013b}	http://localhost:8029/rennesm
8	http://localhost:8029/rennesmetro/v1.0/Things(8)	ST1_75	Lora ST1_75	{deveui:2cf7f1202520010a,devAddr:0020010a}	http://localhost:8029/rennesm
9	http://localhost:8029/rennesmetro/v1.0/Things(9)	ST1_50	Lora ST1_50	{deveui:2cf7f120252000df,devAddr:002000df}	http://localhost:8029/rennesm
10	http://localhost:8029/rennesmetro/v1.0/Things(10)	ST1_25	Lora ST1_25	{deveui:2cf7f120252000c3,devAddr:002000c3}	http://localhost:8029/rennesm
11	http://localhost:8029/rennesmetro/v1.0/Things(11)	A_DEFINIR	Lora A_DEFINIR	{deveui:2cf7f120315000f8,devAddr:005000f8}	http://localhost:8029/rennesm

```
{
  "header": true,
  "nan": true,
  "duplicates": true,
  "columns": {
    "1": {
      "datastream": "1",
      "featureOfInterest": "1"
    }
  }
}
```

# ➤ Une documentation complète (avec de nombreux exemples).



A dark-themed sidebar navigation menu for the SensorThings API. It features a search bar at the top with the text 'Filter...' and a close button 'x'. Below the search bar, there are two main sections: 'SensorThings' and 'Locations'. Each section has an 'Infos' link and a 'Get all' link. The 'Get all' link for 'Things' is currently selected and highlighted in blue. Other links include 'Get one', 'Get only a property', and 'Get only the value of a property'.

## Things | Get all

1.0.0

Retrieve all Things.

GET

`http://sensorthings.geosas.fr/v1.0/Things`

Example usage:

`/v1.0/Things`

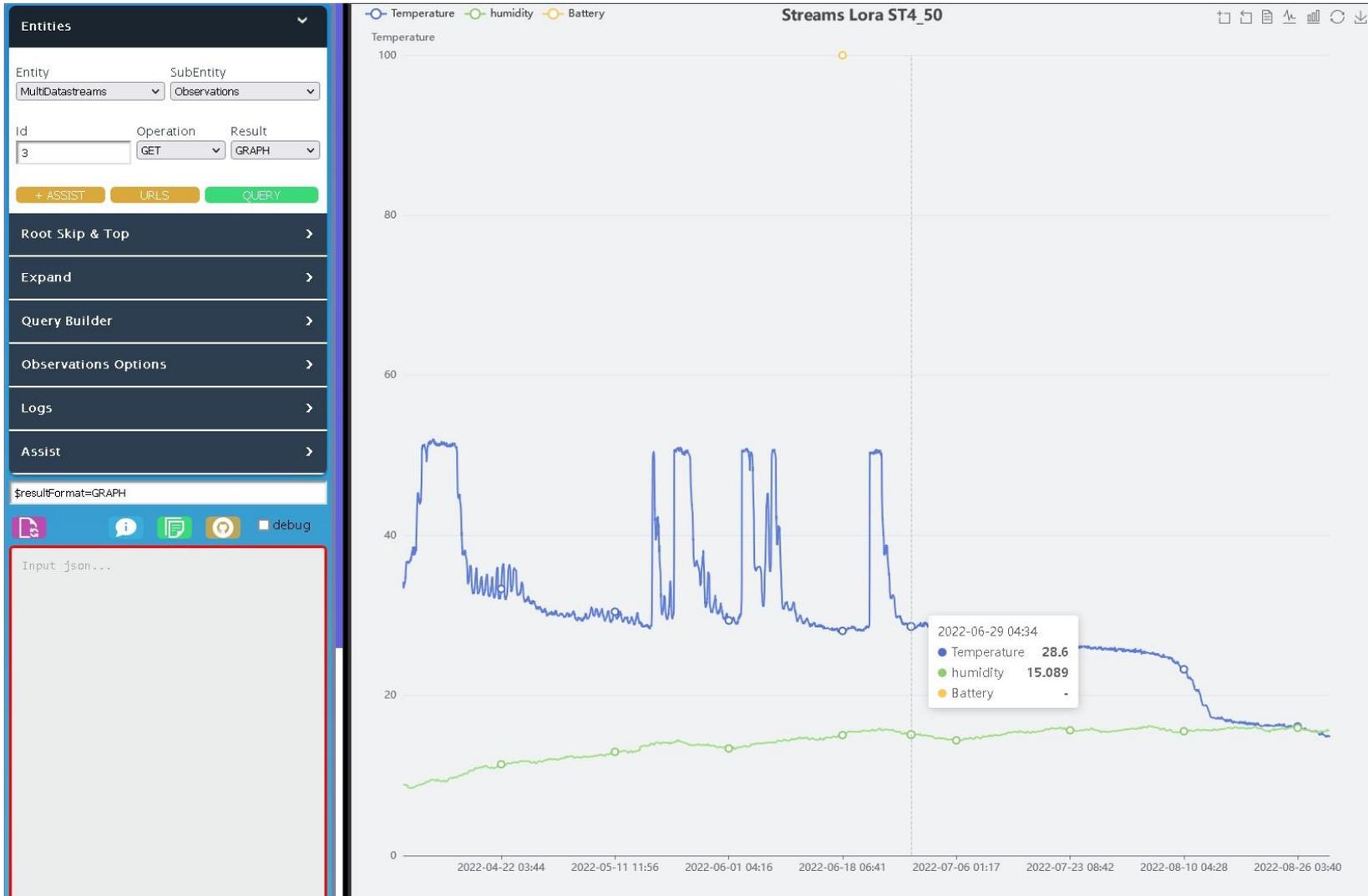
### Success 200

Champ	Type	Description
id	<b>requis</b> number	@iot.id
selfLink	<b>requis</b> relation	@iot.selfLink
Locations	<b>requis</b> relation	Locations@iot.navigationLink
HistoricalLocation	<b>requis</b> relation	HistoricalLocation@iot.navigationLink
Datastreams	<b>requis</b> relation	Datastreams@iot.navigationLink
MultiDatastreams	<b>requis</b> relation	MultiDatastreams@iot.navigationLink

Success-Response:

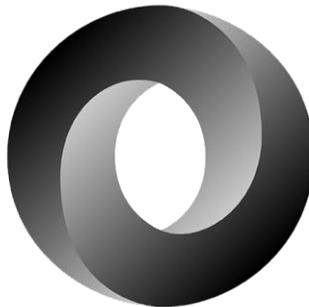
```
{
  "@iot.count": "20",
  "value": [
    {
      "@iot.id": "1",
      "@iot.selfLink": "http://sensorthings.geosas.fr/v1.0/Things(1)",
      "description": "A SensorThingWeb thing Number one",
      "name": "SensorWebThing 1",
      "properties": {
```

# ➤ Format graph (Apache e-chart aussi dans l'api)



# ➤ Alimentation Serveur ST API en données :

```
{
  "gps": [],
  "data": {
    "humidity": 19.9,
    "temperature": 14.55
  },
  "lora": {
    "freq": 867.7,
    "lsnr": 10.2,
    "rssi": -94,
    "data_rate": "SF7BW125"
  },
  "deveui": "2cf7f120252000c3",
  "gateway": {
    "best": "7076FF00560602AB",
    "number": 25
  },
  "lorawan": {
    "adr": true,
    "fcnt": 39769,
    "port": 2,
    "type": "confirmed_data_up",
    "devaddr": "002000c3"
  },
  "sensor_id": "2cf7f120252000c3",
  "timestamp": "2022-09-21T10:14:37+02:00",
  "payload_ciphared": null,
  "payload_deciphared": "010610d6380000010710bc4d0000db82"
}
```



JSON



INRAE

l'institut Agro  
agriculture • alimentation • environnement



Webinaire SensorThings

29 septembre 2022

# Client SensorThings

## Solution ad hoc : Viewer « découverte »

- Développement en cours
- Javascript, Leaflet, Highcharts
- Mobilisable dans différents contextes : standalone, intégré dans une viewer carto, ...

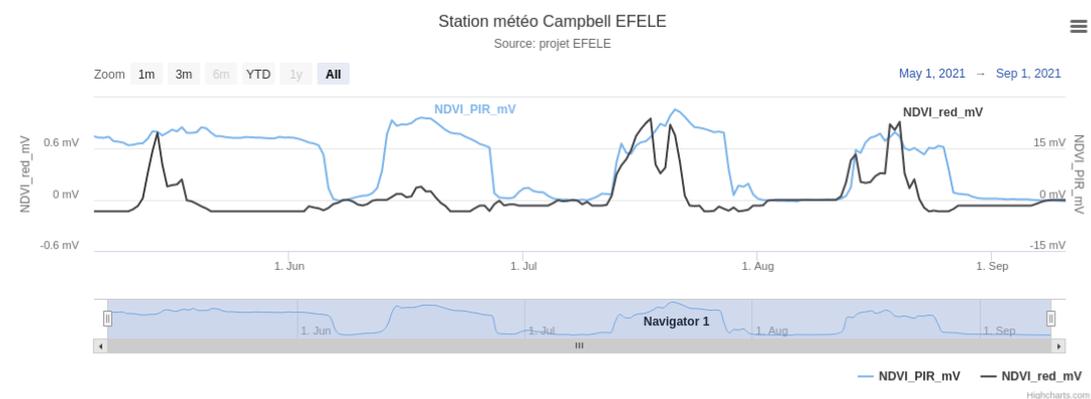
**EFELE**  
Meteo EFELE

### Thing

Station météo Campbell EFELE ▾

### Datastreams

- Hum\_relative (%)
- NDVI\_PIR\_mV (mV)
- NDVI\_red\_mV (mV)
- Patm\_kPa (kPa)
- Pluie\_mm (mm)
- R\_net\_corr\_W\_m2 (W/m<sup>2</sup>)
- R\_net\_W\_m2 (W/m<sup>2</sup>)
- Rg\_kJ\_m2 (kJ/m<sup>2</sup>)
- Rg\_W\_m2 (W/m<sup>2</sup>)
- Vent\_direction\_sd\_deg (\*)
- Vent\_direction\_moy\_deg (\*)
- Vitesse\_vent\_moy\_m\_s (m/s)
- Temp\_air\_celsius (°C)



On peut écrire des trucs



### Les propriétés observées

- Relative humidity, Définition
- NDVI, Définition
- Atmospheric pressure, Définition
- Precipitation amount, Définition
- Radiation, Définition
- Global radiation, Définition
- Wind direction standard deviation, Définition
- Wind direction, Définition
- Wind speed, Définition
- Temperature, Définition

### Les Sensors

- HMP60 Temperature & Relative Humidity Probe, Définition
- Capteur IR et PIR, Définition
- CS100 Setra Barometric Sensor, Définition
- ARG100 Raingauge, Définition
- NR-LITE2 Net Radiometer, Définition

# Client SensorThings

## Solution ad'hoc : Mviewer (POC + dév en cours)

The screenshot displays the Mviewer web application interface. At the top, there is a green header with the OGC logo and the text "Démonstrateur SensorThings". A search bar labeled "Rechercher" is located in the top right corner. On the left side, a navigation menu lists "Services", "Couches géographiques Réseaux", and "Couches géographiques Surfaces".

The main area features a map of Brittany with several blue circular markers indicating sensor locations. A modal window titled "SensorThings" is open, showing the service address "gs-wq.brgm-rec.fr/FROST-Server/v1.0/" and a button "AFFICHER SUR LA CARTE".

Below the map, the "Informations" section provides details for a specific sensor. It includes a "Zoom" control with options for 1m, 3m, 6m, YTD, 1y, and All. The data visualization shows a line graph of pH levels over time, with the x-axis representing years from 2004 to 2015 and the y-axis representing pH units. The graph shows a relatively stable pH level around 7.0. Below the graph, there are several checkboxes for different data series, with "pH in Eau on Eau brute at à Saint-Georges-de-Didon" selected.

At the bottom of the page, there is a large, faint watermark logo.

# Clients SensorThings

## Solution pré-existante : Grafana

- Proposé sous la forme d'un plugin SensorThings : **LinkSmart**
- 1<sup>er</sup> Tests intéressants pour le monitoring des services et des instances SensorThings
- A priori, plus limité pour l'exploration des time series

## Grafana OGC SensorThings Plugin

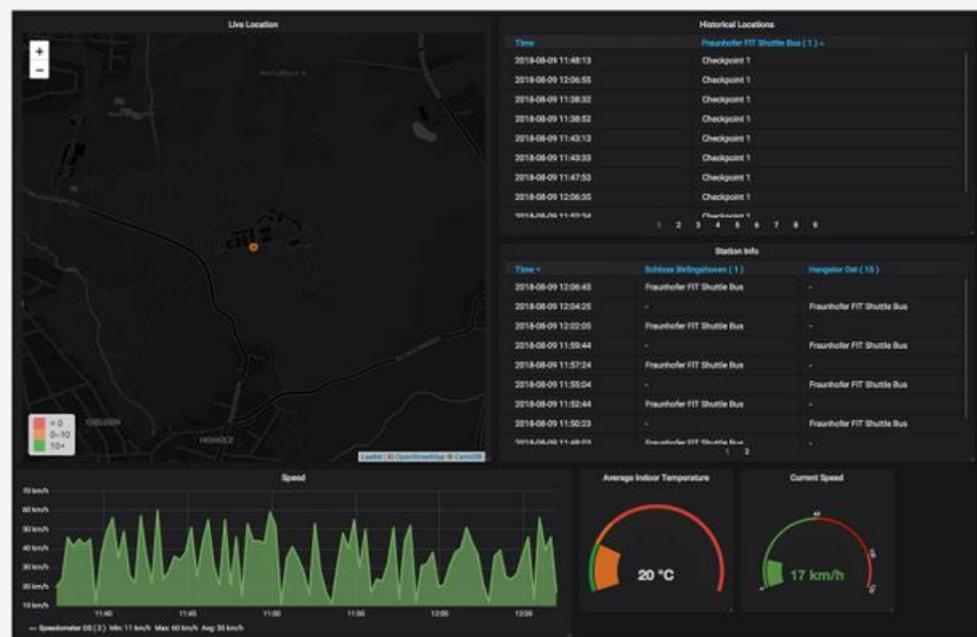
This plugin enables the visualization of sensor and location data from an [OGC SensorThings](#) server on [Grafana](#).

It provides:

- Time-series visualization on [Graph](#)
- Time-series and location history in [Table](#)
- Sensor data in [Singlestat](#)
- Location of things on [Worldmap Panel](#)
- [JSONPath](#) support for extracting values from [OM\\_Observation](#) observations

## Demo

A live dashboard showing SensorThings data in Grafana panels: [SensorThings Dashboard](#)



# ➤ Déploiements effectifs

## En production

- Lora Rennes Métropole : <https://api.geosas.fr/rennesmetro/v1.0/>
  - En production depuis septembre 2021
  - 8 capteurs réparties sur l'agglo de Rennes
  - Données météo en milieu urbain (°C, humidité) + monitoring (% batterie)
  - Fréquence : 5 minutes
  - Choix de regrouper les observations via un Multidatastream
  
- Agri4cast France : <https://api.geosas.fr/agri4cast/v1.0/>
  - En production depuis avril 2022
  - Maille météo 25x25 km
  - Données météo (Radiation, ETP, pluie, vent, °C)
  - Fréquence : données quotidiennes depuis 1979
  - L'objet de l'observation est représenté par un polygone



## En cours : 2 observatoires labélisés

- ORE AgrHyS
- SOERE PRO EFELE

→ Processus de mapping observatoire / Service ST itératif intégrant métrologues, scientifiques et « numériques ».



# Retour utilisateurs

## 2 exemples

### 1. Enseignante/Chercheure en hydrologie : Service Lora Rennes Métropole

- Ses usages :
  - Monitoring : STA → « Moulinettes » Python -> CSV → visualisation dans Matlab
  - En entrée de modélisation hydrologique. statique pour le moment, dynamique à terme.
- Son avis :
  - un « progrès énorme »
  - Simplicité et autonomie dans toute la chaine pour passer du capteur au modèle
  - La STA laisse la liberté d'utiliser ses propres outils



### 2. Développeur de services web : Service Agri4cast France

- Ses usages :
  - Moissonnage dynamique pour la modélisation : Modèle TNT (Transfert de Nitrate/Marées vertes) pour scientifiques
  - Moissonnage dynamique dans un projet diagnostique agronomique/OAD -> Gestionnaires, Conseillers agricoles
- Son avis :
  - Constat : complexité pour accéder librement et rapidement aux données météo
  - Gain : Simplicité, rapidité, autonomie pour brancher le STA à d'autres outils/services développés en interne



# SDI & STA : metadata ISO 19115/19139

## Grille de 25x25 km d'accès aux données météorologiques Agri4cast sur la France

Données météorologiques Agri4cast basées sur des observations et spatialisées sous la forme d'une grille composée de mailles carrées de 625 km<sup>2</sup> (25 km x 25 km) sur la France.

La table attributaire de cette couche permet d'établir le lien entre l'objet géographique (la maille) et les chroniques d'observation météorologiques via un service conforme au standard OGC SensorThings (<https://api.geosass.fr/agri4cast/v1.0/>)

Les observations sont :

- température maximale de l'air (°C),
- température minimale de l'air (°C),
- température moyenne de l'air (°C),
- vitesse moyenne quotidienne du vent à 10 m (m/s),
- pression atmosphérique (hPa),
- précipitations cumulées (mm/jour),
- évapotranspiration potentielle d'un couvert végétal (mm/jour),
- rayonnement global total (kJ/m<sup>2</sup>/jour).

Les données météorologiques sont quotidiennes de 1979 à la dernière année civile révolue. Plus d'informations sur le portail Agri4cast : <https://agri4cast.jrc.ec.europa.eu/DataPortal/index.aspx>

Mise à jour continue

### Téléchargements et liens

 Couche géographique de la grille composée de mailles de 25km x 25km	Visualiser
Cette donnée est publiée dans le service de visualisation (WMS) disponible à l'adresse <a href="http://psncalc.agrocampus-ouest.fr/tntserver/wms/">http://psncalc.agrocampus-ouest.fr/tntserver/wms/</a> , couche <code>tnt_data:agri4cast_sensorthings</code> .	
 Interface de visualisation et téléchargement des données météo <a href="https://geosass.fr/sentinelhub/wms_sensorthing.html">https://geosass.fr/sentinelhub/wms_sensorthing.html</a>	Ouvrir le lien
 Couche géographique de la grille composée de mailles de 25km x 25km <a href="http://psncalc.agrocampus-ouest.fr/tntserver/wfs?service=WFS&amp;version=2.0.0&amp;request=GetFeature&amp;typeName=tnt_data:agri4cast_sensorthings&amp;outputFormat=SHAPE-ZIP">http://psncalc.agrocampus-ouest.fr/tntserver/wfs?service=WFS&amp;version=2.0.0&amp;request=GetFeature&amp;typeName=tnt_data:agri4cast_sensorthings&amp;outputFormat=SHAPE-ZIP</a>	Ouvrir le lien
Service OGC SensorThings Agri4cast <a href="https://api.geosass.fr/agri4cast/v1.0/">https://api.geosass.fr/agri4cast/v1.0/</a>	Ouvrir le lien

### Ressources associées

 Service de téléchargement des données météorologiques Agri4cast sur la France (API OGC SensorThings) (Service associé) L'API Agri4Cast diffuse des données météorologiques interpolées sur une grille de 25x25 km sur la France à partir d'observations	Service associé
--	-----------------

### Aperçu



### Aucune évaluation ★

- Voir tous les commentaires
- Ajouter votre commentaire

### Étendue spatiale



### Étendue temporelle

#### Date de publication

2022-06-22

#### Période



#### Fourni par

Gé@SAS.fr

#### Mis à jour :



# SDI & STA : Service metadata ISO 19119

## Service de téléchargement des données météorologiques Agri4cast sur la France (API OGC SensorThings)

L'API Agri4Cast diffuse des données météorologiques interpolées sur une grille de 25x25 km sur la France à partir d'observations météorologiques réalisées sur l'union européenne et les pays voisins. Les données sont quotidiennes de 1979 à la dernière année civile révolue. Plus d'informations sur le portail Agri4cast : <https://agri4cast.jrc.ec.europa.eu/DataPortal/Index.aspx>

### Téléchargements et liens

<a href="https://api.geosas.fr/agri4cast/v1.0/">https://api.geosas.fr/agri4cast/v1.0/</a>	<input type="button" value="Ouvrir le lien"/>
 Documentation d'API OGC SensorThings Agri4cast API Doc <a href="https://geosas.fr/web/?page_id=5984">https://geosas.fr/web/?page_id=5984</a>	<input type="button" value="Ouvrir le lien"/>

### Ressources associées

 <a href="#">Grille de 25x25 km d'accès aux données météorologiques Agri4cast sur la France</a> (Données associées) Données météorologiques Agri4cast basées sur des observations et spatialisées sous la forme d'une grille composée de mailles carrées de 625 km² (25 km x 25 km) sur la France. La table attributaire de cette couche permet... <a href="#">plus...</a>	<input type="button" value="Données associées"/>
--	--

### À propos de cette ressource

<b>INSPIRE Service taxonomy</b>	<ul style="list-style-type: none"> <li>infoSensorDescriptionService <input type="button" value="Q"/></li> </ul>
<b>Registre de thème INSPIRE</b>	<ul style="list-style-type: none"> <li>Systèmes de maillage géographique <input type="button" value="Q"/></li> <li>Caractéristiques géographiques météorologiques <input type="button" value="Q"/></li> </ul>
<b>GEMET</b>	<ul style="list-style-type: none"> <li>météorologie <input type="button" value="Q"/></li> </ul>
<b>Autres mots clés</b>	<ul style="list-style-type: none"> <li>Sensorthings <input type="button" value="Q"/></li> <li>API <input type="button" value="Q"/></li> </ul>
<b>Identificateur de ressource unique</b>	<ul style="list-style-type: none"> <li>fr-agri4cast-sensorthings</li> </ul>

<b>Contraintes légales</b>	<p>Pas de restriction d'accès public</p> <p>Usage libre sous réserve des mentions obligatoires sur tout document de diffusion : "Source : Joint Research Centre of the European Commission"</p> <p>Licence ouverte : <a href="https://www.etalab.gouv.fr/licence-ouverte-open-licence">https://www.etalab.gouv.fr/licence-ouverte-open-licence</a></p>
----------------------------	--

### Aperçu



thumbnail

### Aucune évaluation ★

- 
- 

### Étendue spatiale



### Étendue temporelle

**Date de publication**  
2022-06-01

**Fourni par**

GéoSAS.fr

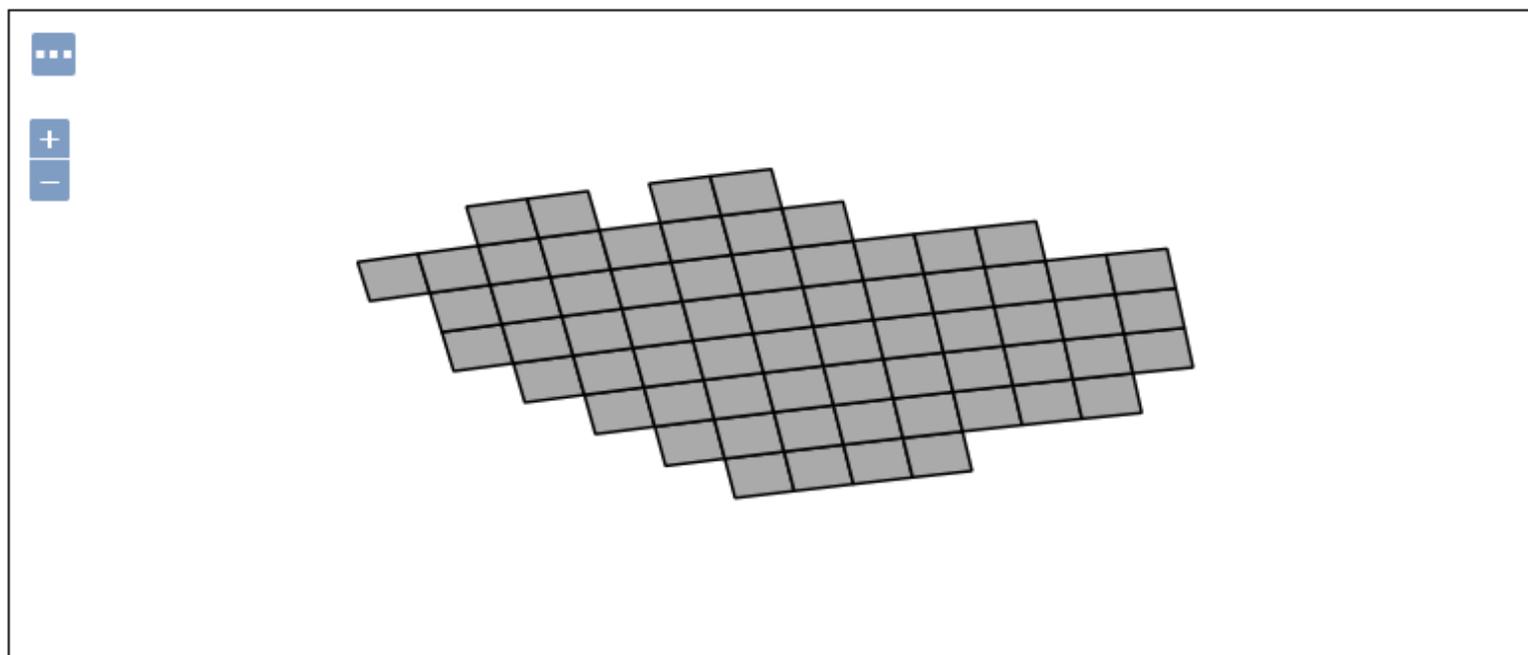
**Mis à jour :**

il y a un mois



## Principe

- Publication via Geoserver d'une couche géographique intégrant des liens vers STA : Datastreams/Observations



Scale = 1 : 4M

-2.36206, 48.25195

### agri4cast\_sensorthings

fid	name	descriptio	radiation	ETP	precipitat	pression
agri4cast_sensorthings.fid-7158c80d_183324e8a69_1b12	91073	Maille Agri4Cast, Grid_Code : 91073	<a href="https://api.geosas.fr/agri4cast/v1.0/Datastreams(405)/Observations">https://api.geosas.fr/agri4cast/v1.0/Datastreams(405)/Observations</a>	<a href="https://api.geosas.fr/agri4cast/v1.0/Datastreams(406)/Observations">https://api.geosas.fr/agri4cast/v1.0/Datastreams(406)/Observations</a>	<a href="https://api.geosas.fr/agri4cast/v1.0/Datastreams(407)/Observations">https://api.geosas.fr/agri4cast/v1.0/Datastreams(407)/Observations</a>	<a href="https://api.geosas.fr/agri4cast/v1.0/Datastreams(408)/Observations">https://api.geosas.fr/agri4cast/v1.0/Datastreams(408)/Observations</a>

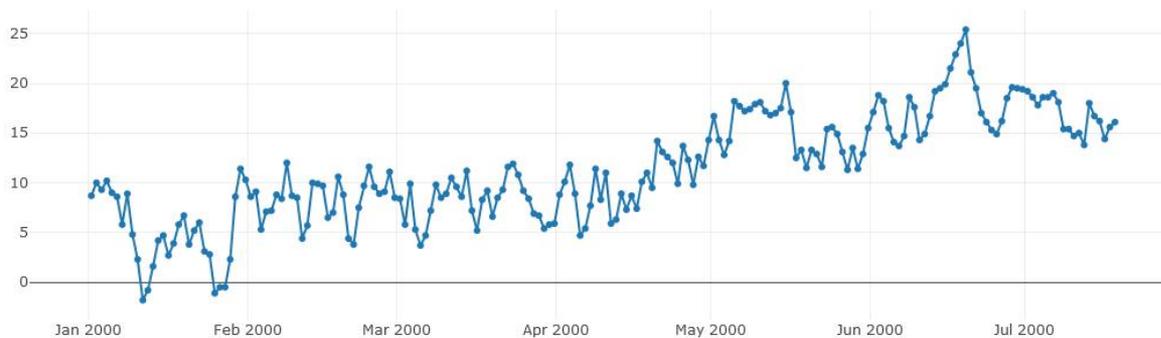
## Exemple de réutilisation

- Via un viewer dédié WMS & STA

wms+sensorT



Temperature moyenne journaliere de l'air en : Degree Celsius °C



# Merci !

## Ressources

- Forge **STAE-ND** : <https://github.com/Mario-35/api-sensorthing>
- STA Lora - Rennes Métropole : <https://api.geosas.fr/lora/v1.0/>
- STA Météo Agri4Cast : <https://api.geosas.fr/agri4cast/v1.0/>
- Démonstrateur ST cartographique : <https://geosas.fr/sensorthings/>
- Portail GéoSAS : <https://geosas.fr>