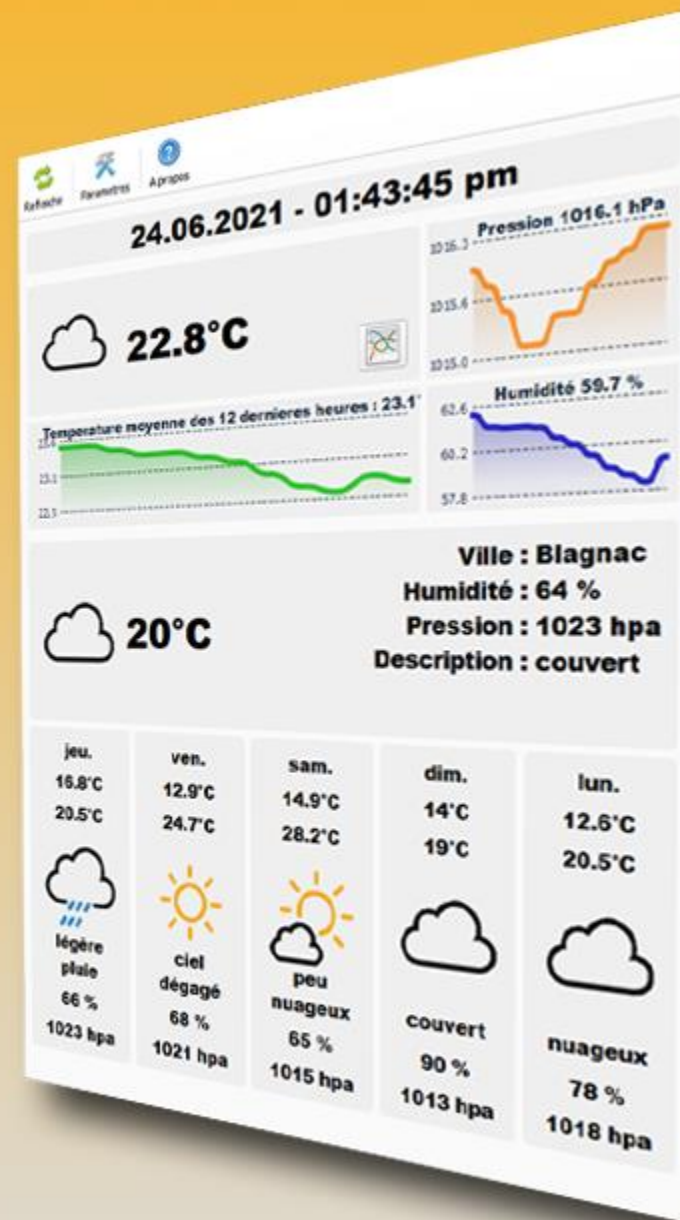


09/06/21
au
24/06/2021



PROJET FINAL

RESKILLING C/C++ Developpeur Embarqué
pour AKKA TECHNOLOGIES via AJC Formation

Lucas SANER
Mickael ANTHEAUME
Stephane CUILLERDIER

SOMMAIRE

LA SPECIFICATION

L'ARCHITECTURE DU PROJET

CHOIX TECHNIQUES

REALISATION

DEMONSTRATION

CONCLUSION

The logo features a white, multi-lobed circular shape centered on a solid yellow background. The text "LA" is positioned above the word "SPECIFICATION", both in a bold, dark brown, sans-serif typeface. The word "SPECIFICATION" is larger and spans across the width of the white shape.

LA SPECIFICATION

STATION MÉTÉO :

- L'objectif de ce projet est de concevoir une application PC Station Météo.
- On souhaite afficher sur cette station Météo des informations météorologiques de 2 points géographiques différents:
 - en mer, ce qu'on appellera la « Balise Mer »
 - Température en °C/°F de -40°C à 50°C
 - Résolution : 0.1°C relevé toutes les 10 minutes
 - Taux d'humidité
 - Pression
 - d'une ville choisie, ce qu'on appelle la « Balise Ville »
 - Température en °C/°F de -40°C à 50°C
 - Résolution : 0.1°C relevé toutes les 10 minutes
 - Gestion de l'affichage de pictogrammes associés
 - Affichage de la Ville
 - Graphique prévisionnel pour les 5 jours suivants
 - Ainsi que l'affichage de l'heure et de la date

Il faudra aussi créer une partie d'administration permettant de configurer certains paramètres :

- Format de l'heure 12H ou 24H
- Choix de la Ville
- Unité de Température Fahrenheit ou Celsius
- Possibilité de choisir les styles d'affichages :
 - Famille de Police
 - Couleur : (Chaque style sera décliné en Mode Jour/Nuit)
 - Choix de la langue •: Anglais / Français

Une partie facultative, s'il reste du temps :

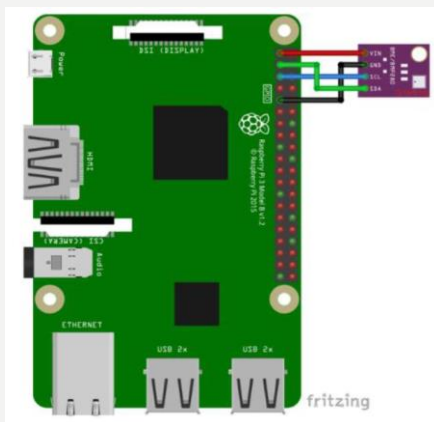
- vous enregistrerez toutes les heures les informations de la balise au sein d'une base de données. Le but sera d'afficher la température moyenne des 12 dernières heures et de l'afficher au sein de votre station.

MATÉRIEL MIS A DISPOSITION :

La balise Mer (en plein cœur de la mer) ce compose :

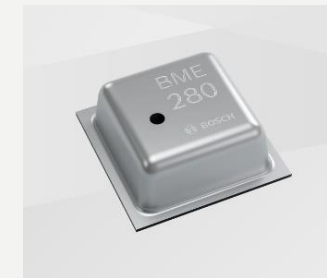
- Raspberry Pi 3 Model B+
- Marque: U:Create
- Processeur: ARM
- Vitesse du processeur: 1.40 GHz
- Nombre de cœurs: 4
- Taille de la mémoire vive: 1GB
- Type de technologie sans fil: 802.11bgn, 802.11ac
- Nombre de ports USB 2.0: 4
- Accès en TCP ou HTTP

Branchement du capteur sur le Raspberry Pi en I2C



Et d'un capteur d'humidité BME280:

- Capteur de température :
 - Température : -40...85°C
 - Précision : 0,01°C
- Capteur d'humidité
 - Humidité : 0...100%
 - Temps de réponse : 1 s
 - Précision : ±3%
- Capteur de pression
 - Pression : 300...1100 hPa
 - Bruit de mesure : 0.2 Pa
- Interface : I2C



Un programme mis à disposition permet de tester l'état de fonctionnement du capteur.

```
pi@pi:~$ python bme280.py
Chip ID      : 96
Version      : 0
Temperature  : 23.59 c
Pressure     : 984.583980249 hPa
Humidity     : 34.0987183674 %
```



L'ÉQUIPE

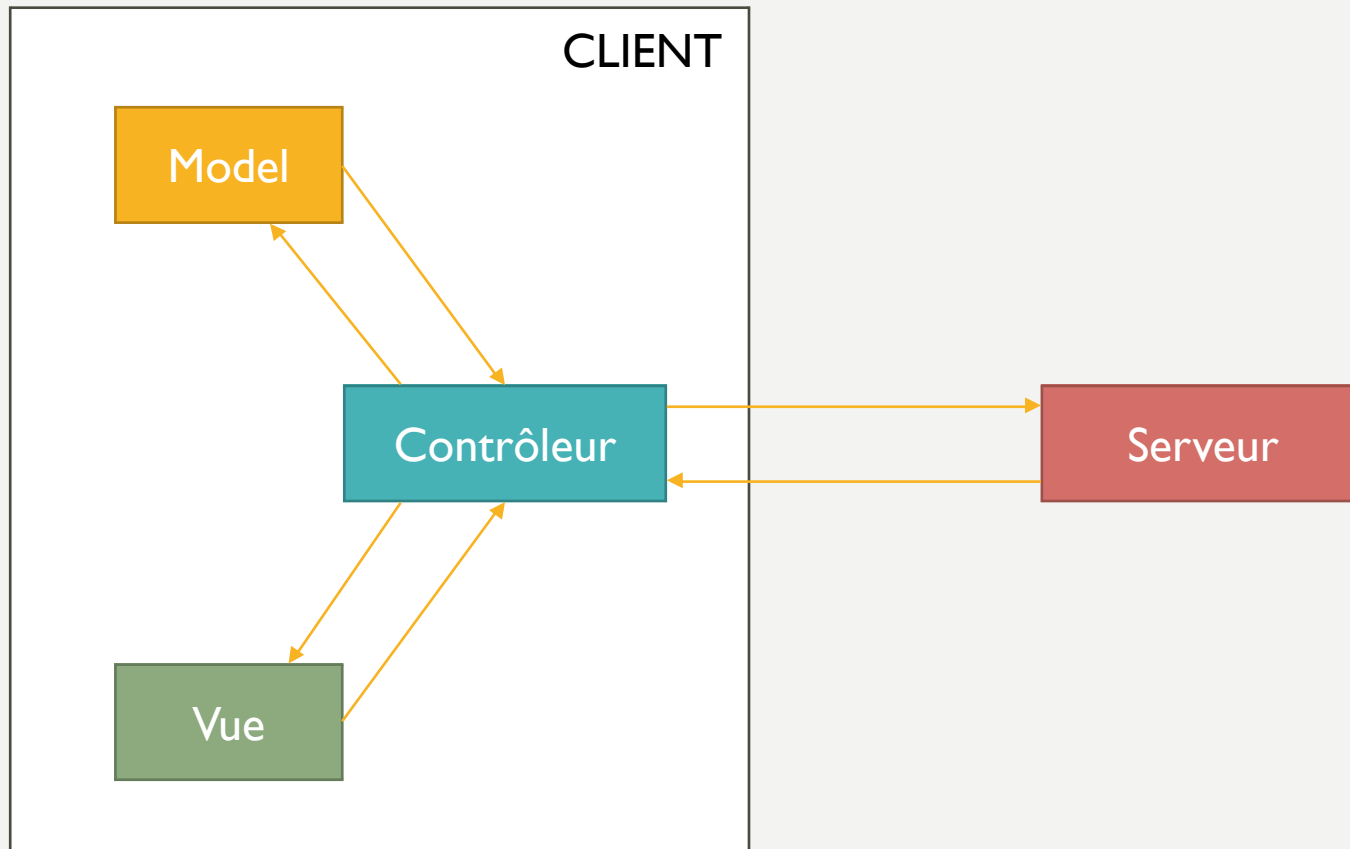
PARTICIPANTS AU PROJET :

- Lucas SANER
- Mickael ANTHEMAUME
- Stéphane CUIILLERDIER

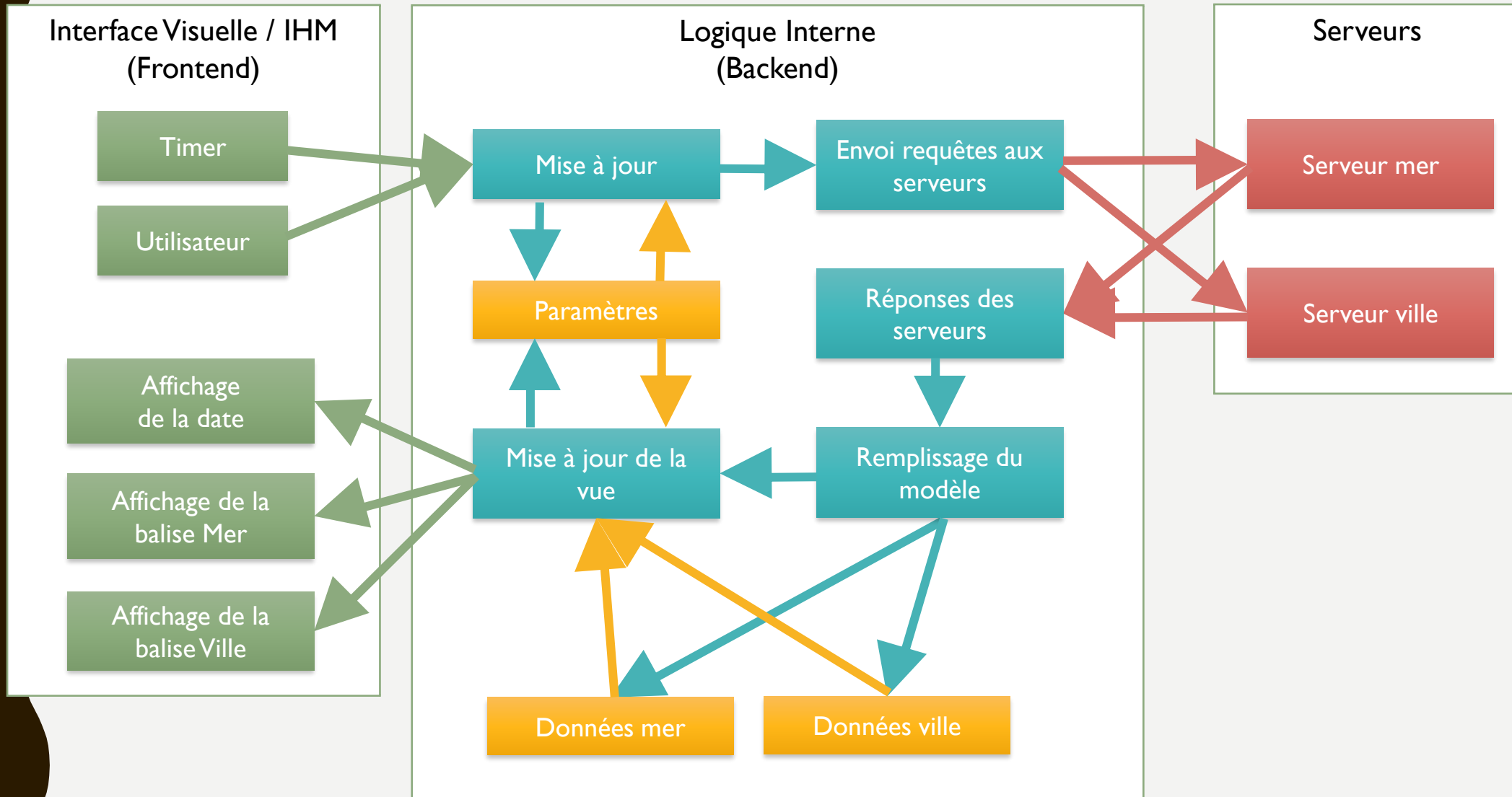


L'ARCHITECTURE DU PROJET

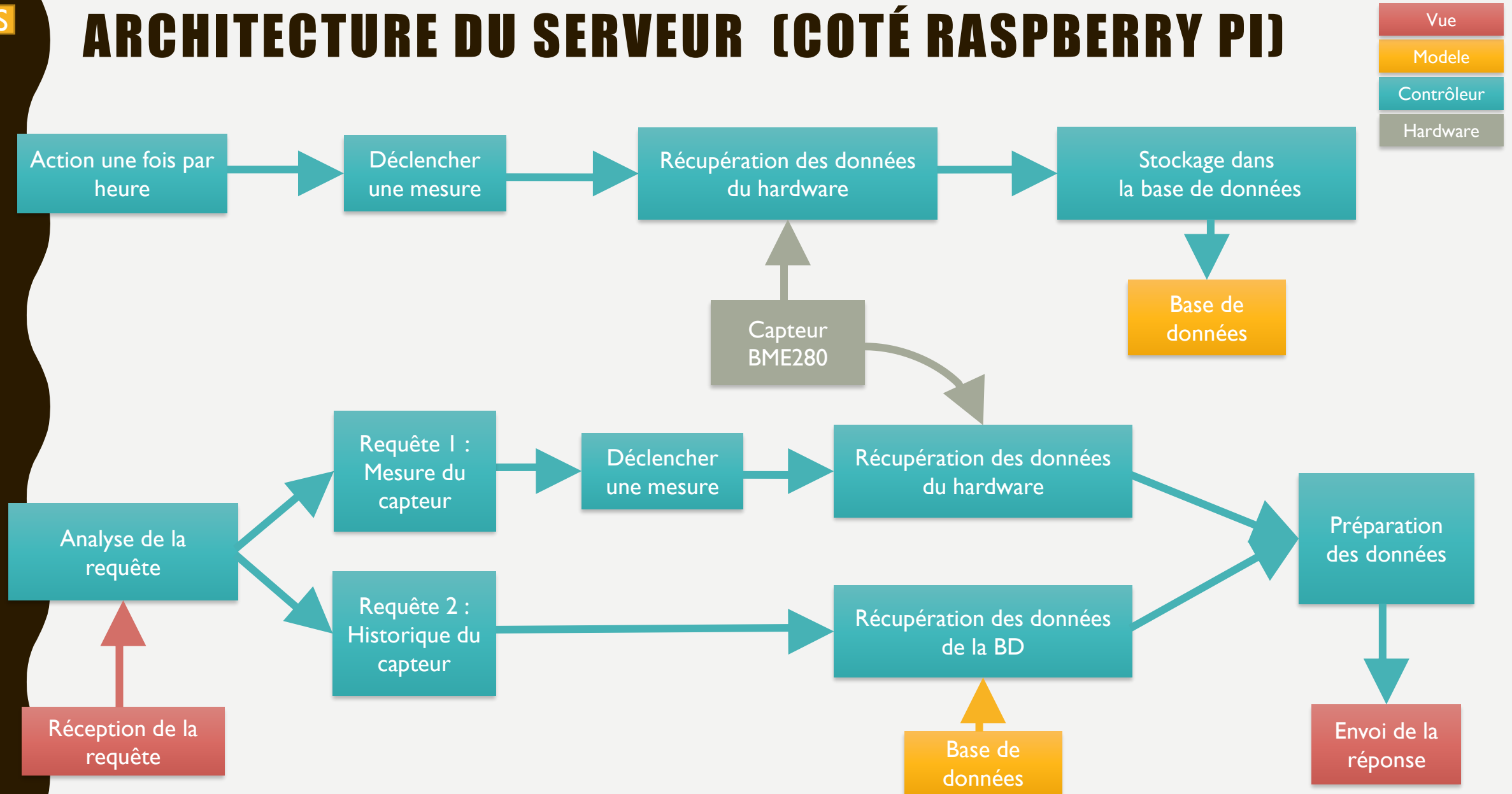
MVC : MODEL-VUE-CONTRÔLEUR DANS UN CONTEXTE CLIENT-SERVEUR



ARCHITECTURE DU CLIENT (COTÉ UTILISATEUR)



ARCHITECTURE DU SERVEUR (COTÉ RASPBERRY PI)



CHOIX TECHNIQUES

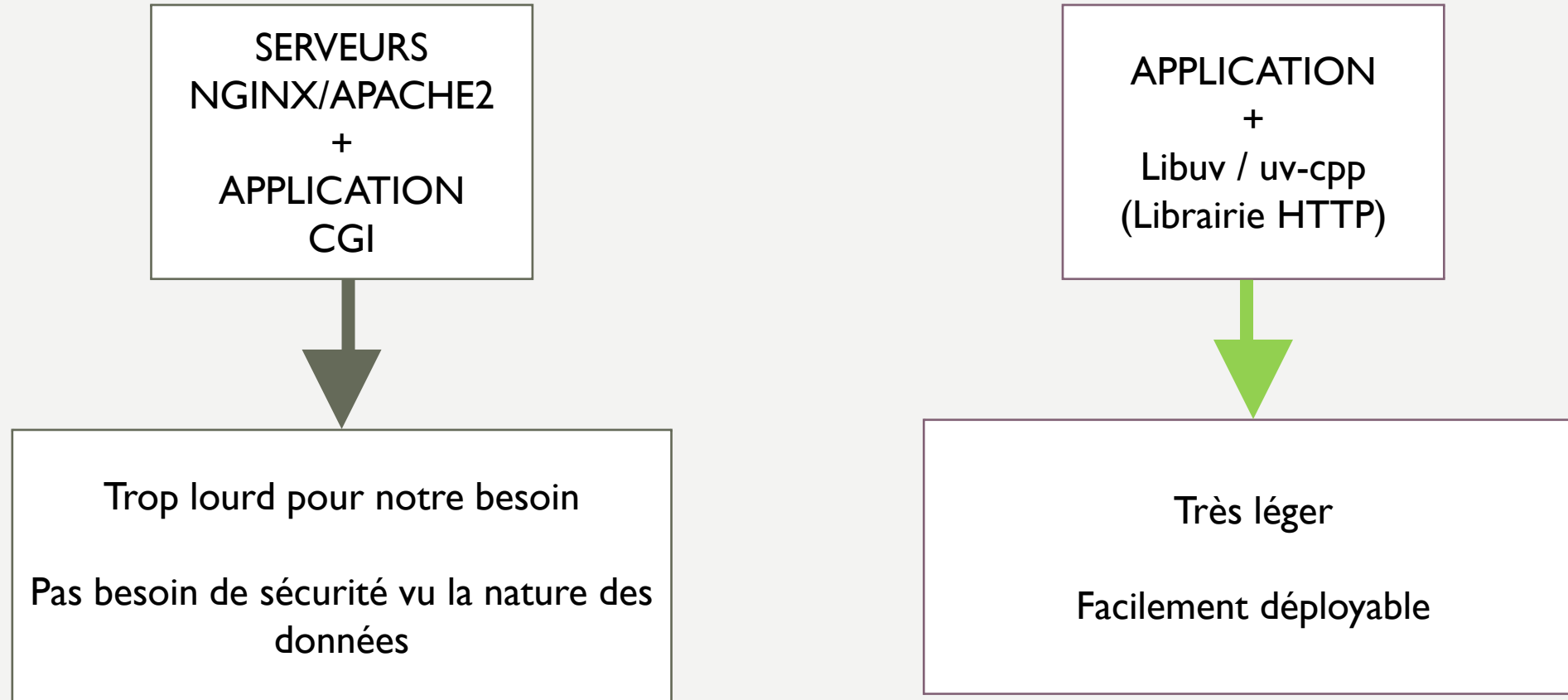
COMPARAISON DES APIS WEB TROUVÉES

API'S:	Infoclimat	MeteoMatics	AccuWeather	OpenWeatherMap	MeteoConcept
Nombre d'appels serveur autorisé	5000 appels/jr	1000 appels/14jr	50 appels/jr	1000 appels/jr	500 appels/jr
Gratuité du service	Gratuit	Payant (essai 14jr)	Gratuit	Gratuit	Gratuit
Type de recherche par ville	Coordonnées géographiques (lat-long)	Coordonnées géographiques (lat-long)	Ville Code postal Coordonnées géographiques (lat-long)	Ville Code postal Coordonnées géographiques (lat-long)	Ville
Données : Température pression humidité icône	Pas d'icone	Oui	Oui	Oui	Oui
Prévisionnel 5 jours	Oui	Oui	Oui	Oui	Oui

COMPARAISON DES APIS WEB TROUVÉES

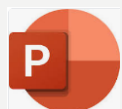
API'S:	Infoclimat	MeteoMatics	AccuWeather	OpenWeatherMap	MeteoConcept
Nombre d'appels serveur autorisé	5000 appels/jr	1000 appels/14jr	50 appels/jr	1000 appels/jr	500 appels/jr
Gratuité du service	Gratuit	Payant (essai 14jr)	Gratuit	Gratuit	Gratuit
Type de recherche par ville	Coordonnées géographiques (lat-long)	Coordonnées géographiques (lat-long)	Ville Code postal Coordonnées géographiques (lat-long)	Ville Code postal Coordonnées géographiques (lat-long)	Ville
Données : Température pression humidité icône	Pas d'icone	Oui	Oui	Oui	Oui
Prévisionnel 5 jours	Oui	Oui	Oui	Oui	Oui

CHOIX DU SERVEUR WEB HTTP



OUTILS / LIBRAIRIES / MATÉRIELS

Outils de productivité :



Powerpoint



Github



Discord



Mobaxterm

Matériel :



Raspberry PI



Capteur BME280

Outils de développement :



C/C++



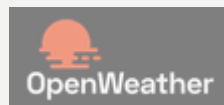
Framework QT (StyleSheet, Translator, Settings)



Libuv : Serveur HTTP



Sqlite3



OpenWeatherMap (Api Web)



Json



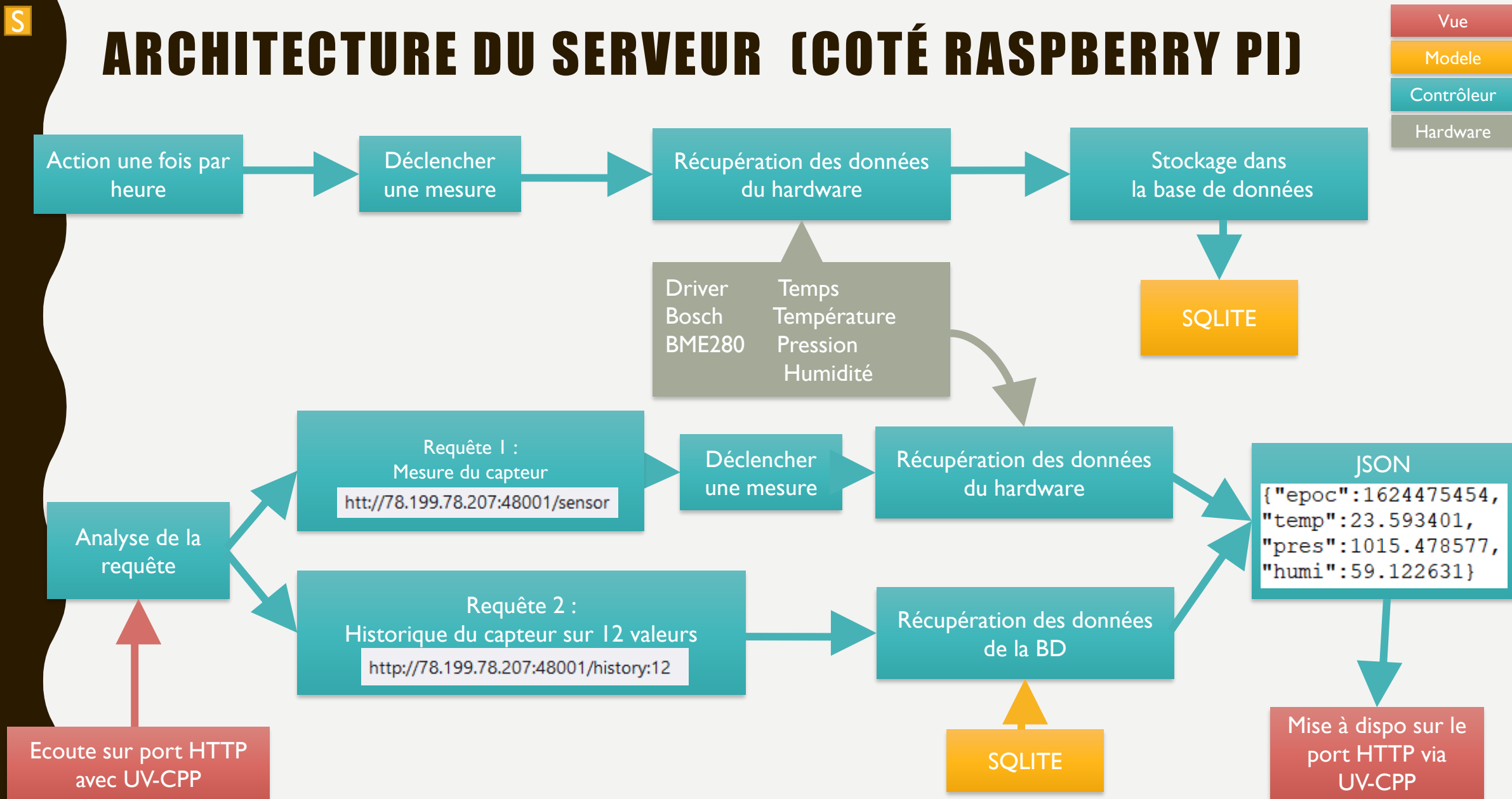
Fichier INI



RÉALISATION

CONCEPTION DU SERVEUR

ARCHITECTURE DU SERVEUR (COTÉ RASPBERRY PI)



DÉCLENCHEMENT D'UNE MESURE DU CAPTEUR EN I2C

```
bool SensorBME280::GetSensorBME280Datas(
    SensorBME280DataStruct *vSensorBME280DataStruct)
{
    {
        struct identifier id;
        if ((id.fd = open(m_I2cBus.c_str(), O_RDWR)) < 0)
        {
            fprintf(stderr, "Failed to open the i2c bus %s\n", m_I2cBus.c_str());
            exit(1); // the service will restart
        }
    }

    {
        id.dev_addr = BME280_I2C_ADDR_PRIM;
        if (ioctl(id.fd, I2C_SLAVE, id.dev_addr) < 0)
        {
            fprintf(stderr, "Failed to acquire bus access and/or talk to slave.\n");
            exit(1); // the service will restart
        }
    }

    {
        struct bme280_dev dev = {BME280_I2C_INTF,
            UserI2cRead, UserI2cWrite,
            UserDelayUs, &id};
        int8_t rslt = bme280_init(&dev);
        if (rslt != BME280_OK)
        {
            fprintf(stderr, "Failed to initialize the device (code %d).\n", rslt);
            exit(1); // the service will restart
        }
    }

    {
        usleep(9000);
        rslt = GetSensorBME280DataNormalMode(&dev, vSensorBME280DataStruct);
        if (rslt != BME280_OK)
        {
            fprintf(stderr, "Failed to stream sensor data (code %d).\n", rslt);
            return false;
        }
    }

    close(id.fd);

    return true;
}
```

Ouverture du BUS I2C du capteur



Acquisition des droits d'accès pour le contrôle du BUS



Initialisation du capteur



Acquisition des mesures : (Température, pression, humidité)



Fermeture du BUS I2C du capteur

EXTRACTION DEPUIS LA BASE DE DONNÉES

Requête SQL pour extraire les 12 dernières mesures depuis la base de données SQLite :

```
select * from tbl_bme280_sensor_history order by epoc_time desc limit 12;
```

```
pi@pi:/usr/share/bme280Server $ sqlite3 database.db3
SQLite version 3.27.2 2019-02-25 16:06:06
Enter ".help" for usage hints.
sqlite> .tables
tbl_bme280_sensor_history
sqlite> select * from tbl_bme280_sensor_history order by epoc_time desc limit 12;
1624474569|23.0199|1015.53|56.6401
1624470969|23.8253|1015.11|61.2338
1624467369|23.5236|1015.29|59.2408
1624463769|23.389|1015.35|59.6951
1624460169|23.4263|1015.3|60.2304
1624456569|23.4887|1015.4|60.138
1624452969|23.5784|1015.48|59.959
1624449369|23.5859|1015.66|59.8327
1624445769|23.6956|1015.31|59.5925
1624442169|23.7804|1015.06|59.3967
1624438569|23.5984|1014.96|60.577
1624434969|24.2766|1014.98|59.9618
sqlite>
```

Envoi des données par le port HTTP

78.199.78.207:48001/history:12 170% ☆ Rechercher

JSON Données brutes En-têtes

Enregistrer Copier Formater et indenter

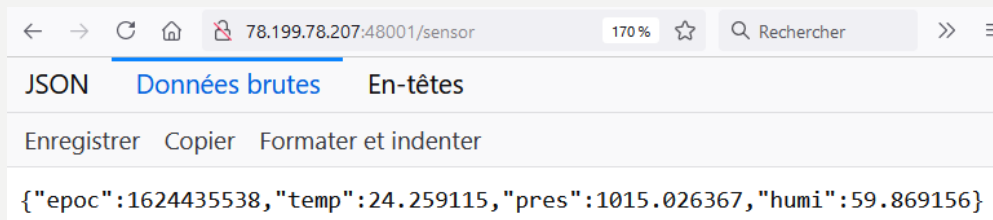
```
{
  "count": 12,
  "history": [
    {
      "epoc": 1624474569,
      "temp": 23.0199,
      "pres": 1015.53,
      "humi": 56.6401
    },
    {
      "epoc": 1624470969,
      "temp": 23.8253,
      "pres": 1015.11,
      "humi": 61.2338
    },
    {
      "epoc": 1624467369,
      "temp": 23.5236,
      "pres": 1015.29,
      "humi": 59.2408
    },
    {
      "epoc": 1624463769,
      "temp": 23.389,
      "pres": 1015.35,
      "humi": 59.6951
    },
    {
      "epoc": 1624460169,
      "temp": 23.4263,
      "pres": 1015.3,
      "humi": 60.2304
    },
    {
      "epoc": 1624456569,
      "temp": 23.4887,
      "pres": 1015.4,
      "humi": 60.138
    },
    {
      "epoc": 1624452969,
      "temp": 23.5784,
      "pres": 1015.48,
      "humi": 59.959
    },
    {
      "epoc": 1624449369,
      "temp": 23.5859,
      "pres": 1015.66,
      "humi": 59.8327
    },
    {
      "epoc": 1624445769,
      "temp": 23.6956,
      "pres": 1015.31,
      "humi": 59.5925
    },
    {
      "epoc": 1624442169,
      "temp": 23.7804,
      "pres": 1015.06,
      "humi": 59.3967
    },
    {
      "epoc": 1624438569,
      "temp": 23.5984,
      "pres": 1014.96,
      "humi": 60.577
    },
    {
      "epoc": 1624434969,
      "temp": 24.2766,
      "pres": 1014.98,
      "humi": 59.9618
    }
  ]
}
```

x12

```
std::string SensorBME280::ConvertSensorBME280DatasToJSON(const uint64_t& vEpoc, const float& vTemp, const float& vPres, const float& vHumi)
{
    int n = snprintf(buffer, BUFFER_LENGTH, "{\"epoc\":%llu,\"temp\":%f,\"pres\":%f,\"humi\":%f}\", vEpoc, vTemp, vPres, vHumi);
    if (n <= 0)
    {
        printf("Err : n <= 0 in ConvertSensorBME280DatasToJSON");
        return "";
    }
    return std::string(buffer, n);
}
```

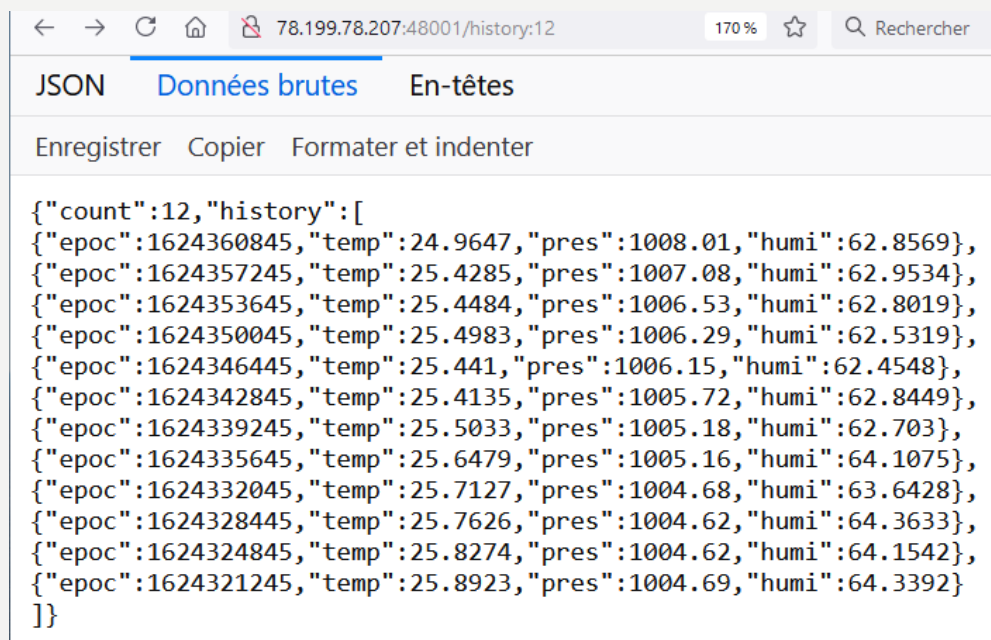
URL'S HTTP DISPONIBLES

`http://ip:port/sensor`



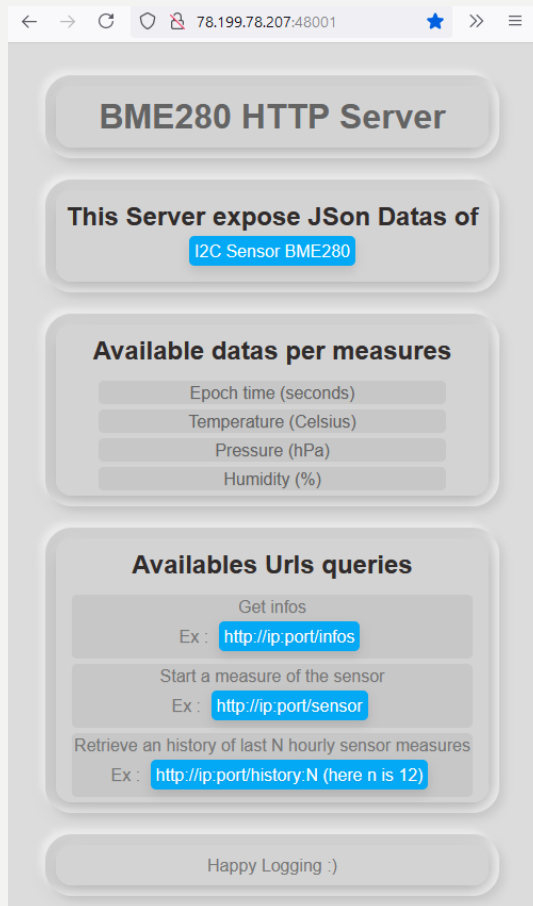
`http://ip:port/history:N`
(n est un nombre de 1..1e6)

Ici les 12 dernières mesures

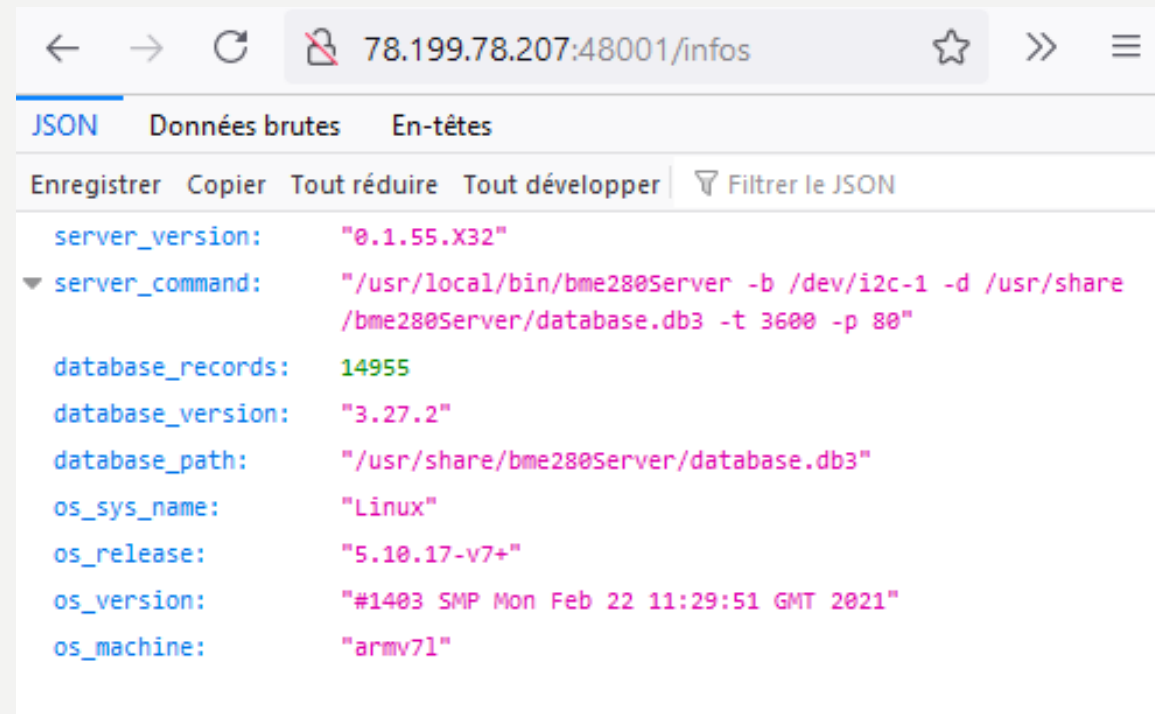


URL'S HTTP DISPONIBLES

<http://ip:port/>



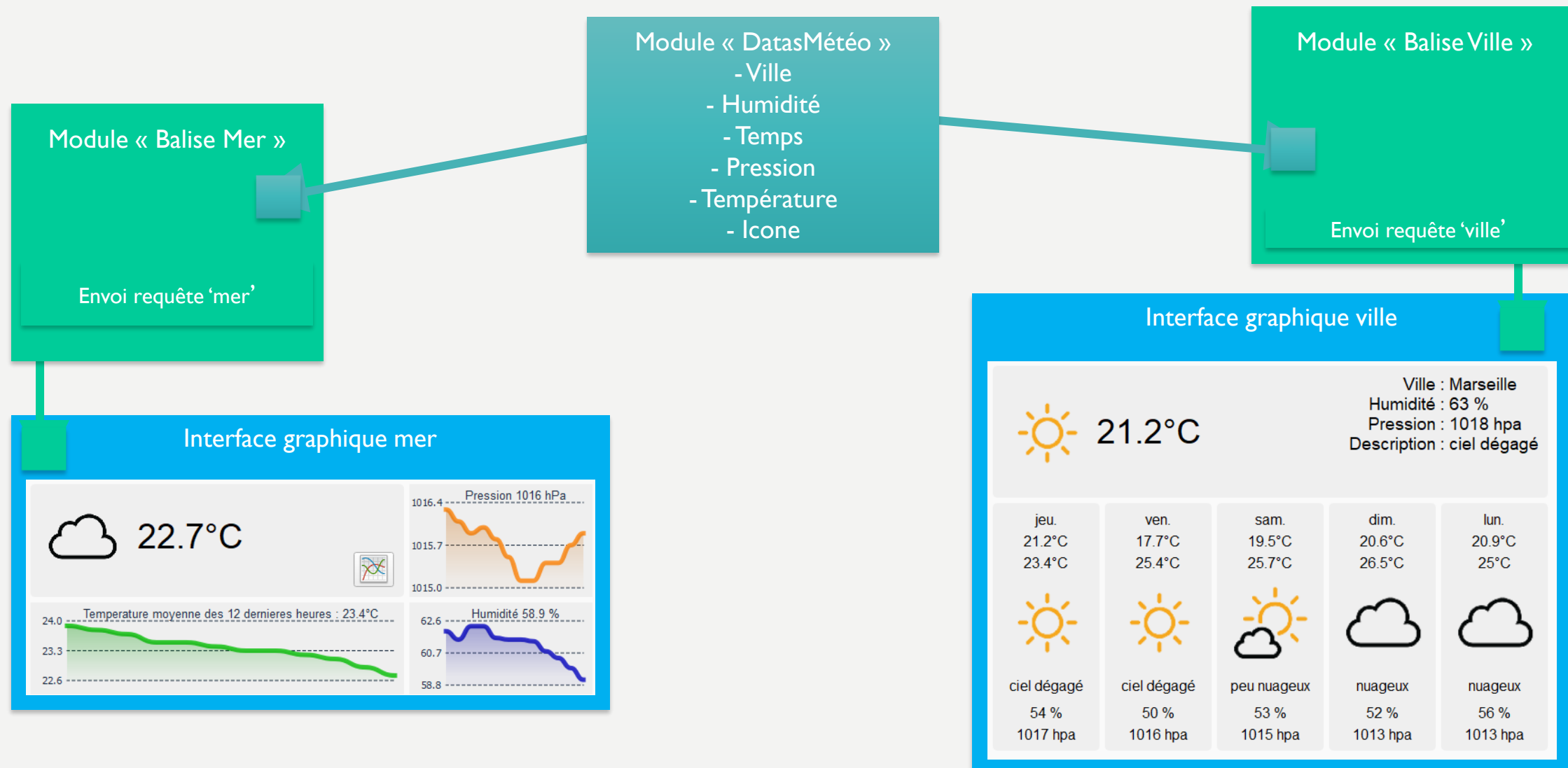
<http://ip:port/infos>



A thick, wavy, yellow line with a white outline, running vertically along the left side of the slide.

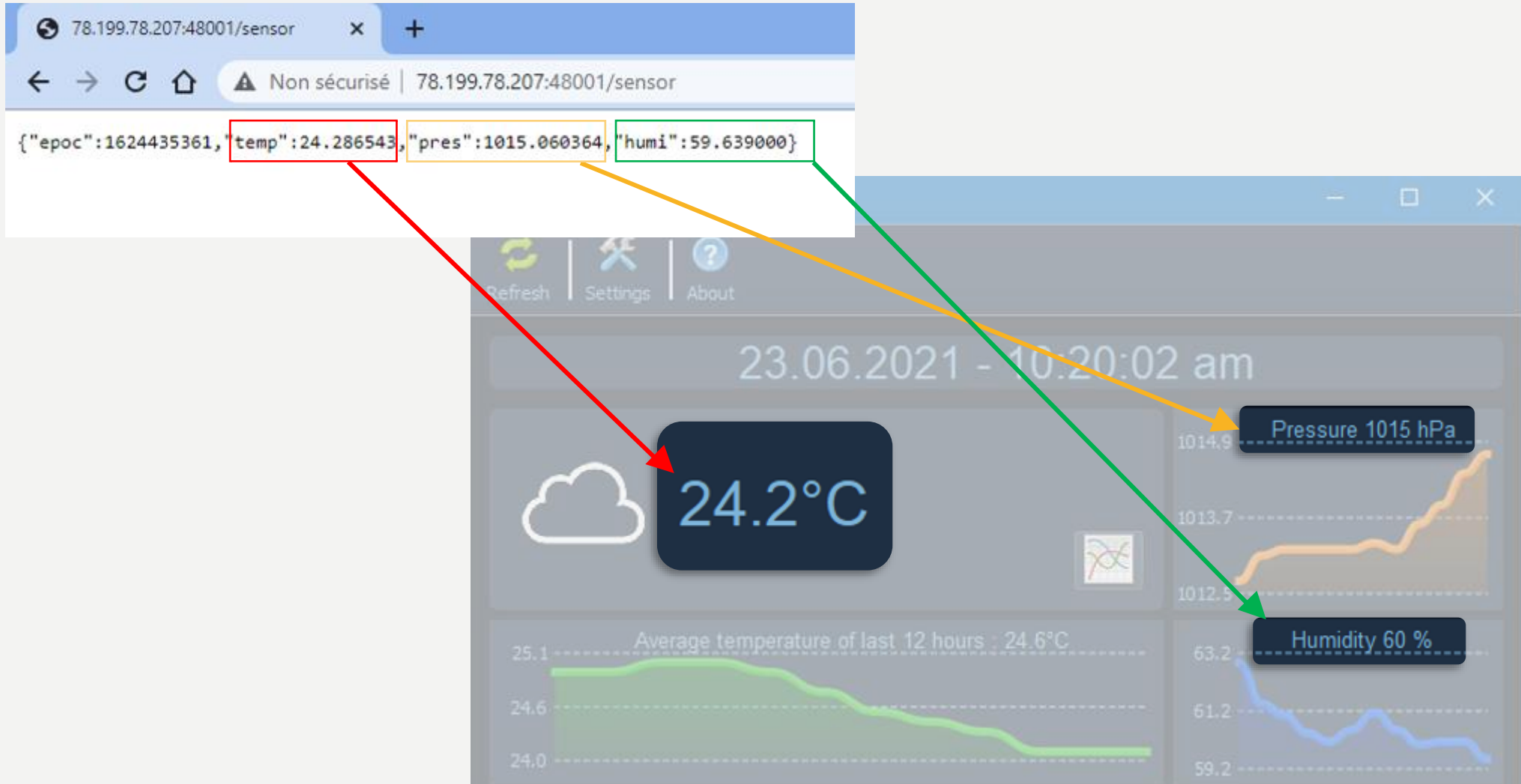
CONCEPTION DU CLIENT

ARCHITECTURE DE L'APPLICATION CLIENT



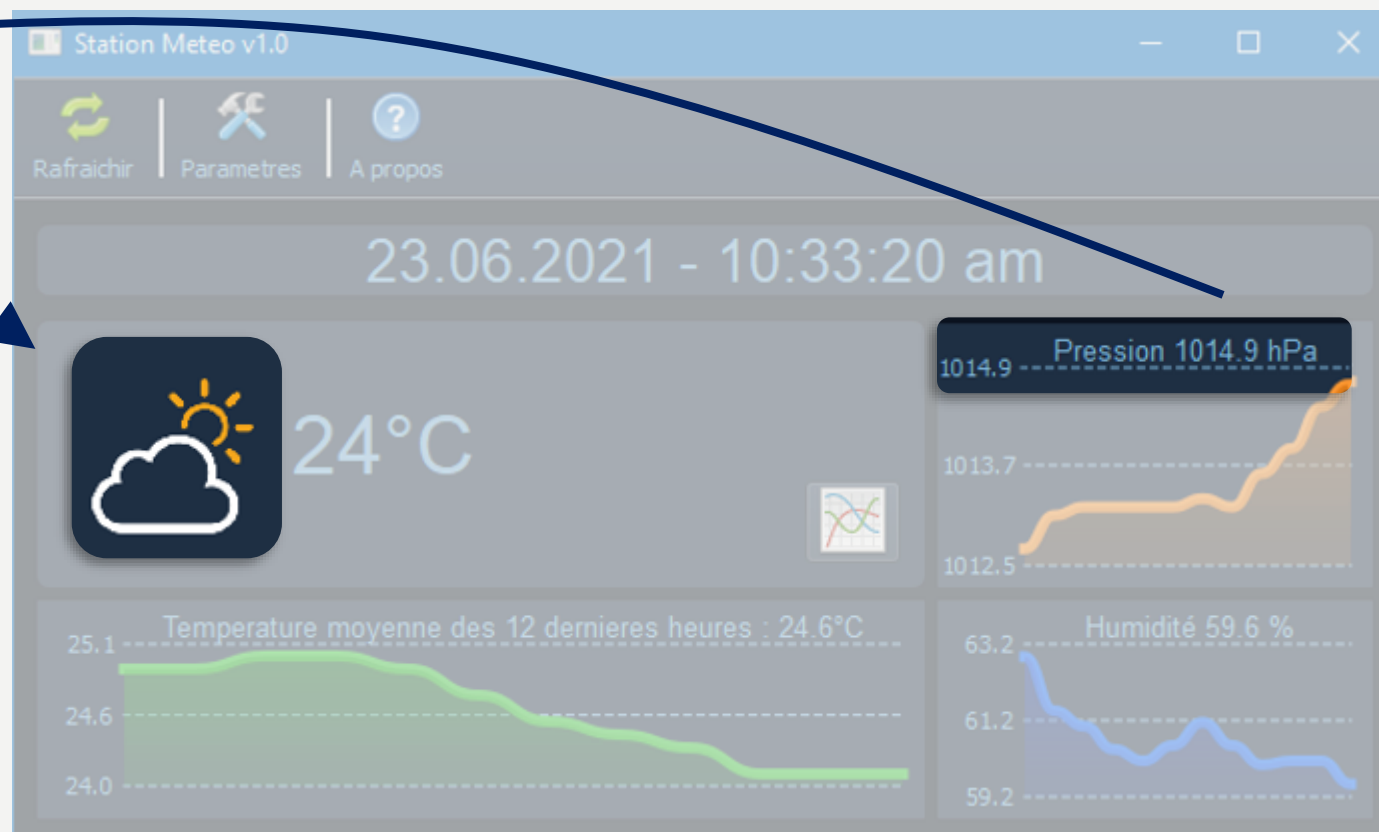
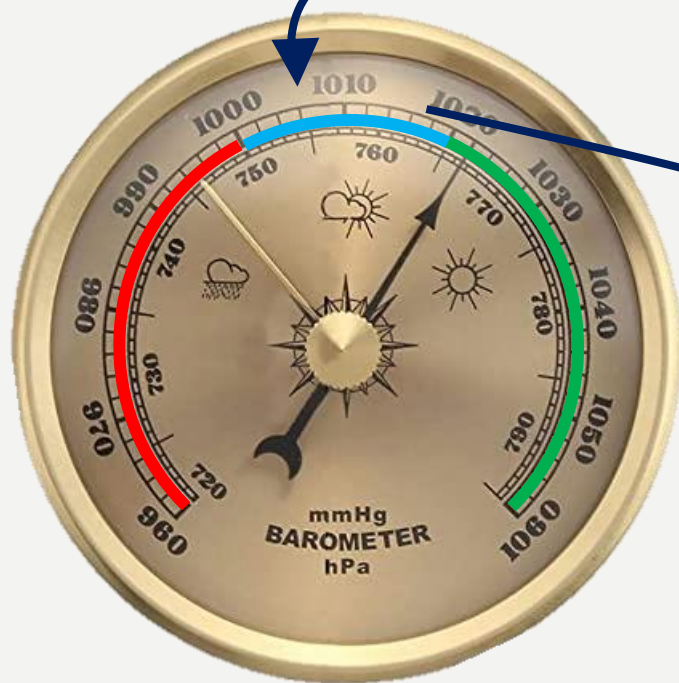
BALISE MER : RÉCUPÉRATION DES MESURES

Récupération des paramètres température, pression, humidité



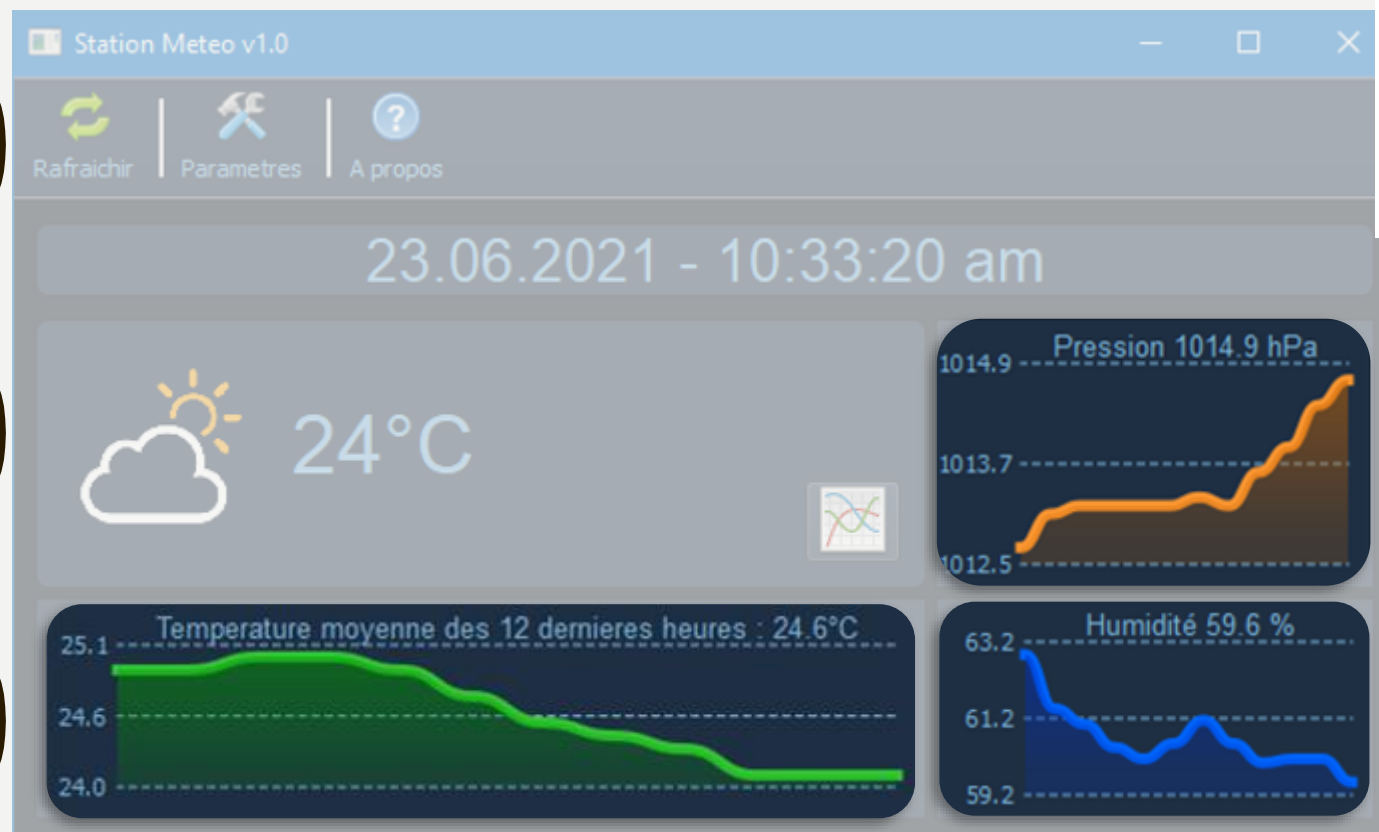
BALISE MER : ILLUSTRATION FONCTION DE LA PRESSION

Exploitation des données de pression



BALISE MER : GRAPHIQUE DES MESURES

Représentation graphique des données



← → ↺ 🏠 78.199.78.207:48001/history:12 170% ☆ 🔍 Rechercher

JSON **Données brutes** En-têtes

Enregistrer Copier Formater et indenter

```
{
  "count": 12,
  "history": [
    {
      "epoc": 1624474569,
      "temp": 23.0199,
      "pres": 1015.53,
      "humi": 56.6401
    },
    {
      "epoc": 1624470969,
      "temp": 23.8253,
      "pres": 1015.11,
      "humi": 61.2338
    },
    {
      "epoc": 1624467369,
      "temp": 23.5236,
      "pres": 1015.29,
      "humi": 59.2408
    },
    {
      "epoc": 1624463769,
      "temp": 23.389,
      "pres": 1015.35,
      "humi": 59.6951
    },
    {
      "epoc": 1624460169,
      "temp": 23.4263,
      "pres": 1015.3,
      "humi": 60.2304
    },
    {
      "epoc": 1624456569,
      "temp": 23.4887,
      "pres": 1015.4,
      "humi": 60.138
    },
    {
      "epoc": 1624452969,
      "temp": 23.5784,
      "pres": 1015.48,
      "humi": 59.959
    },
    {
      "epoc": 1624449369,
      "temp": 23.5859,
      "pres": 1015.66,
      "humi": 59.8327
    },
    {
      "epoc": 1624445769,
      "temp": 23.6956,
      "pres": 1015.31,
      "humi": 59.5925
    },
    {
      "epoc": 1624442169,
      "temp": 23.7804,
      "pres": 1015.06,
      "humi": 59.3967
    },
    {
      "epoc": 1624438569,
      "temp": 23.5984,
      "pres": 1014.96,
      "humi": 60.577
    },
    {
      "epoc": 1624434969,
      "temp": 24.2766,
      "pres": 1014.98,
      "humi": 59.9618
    }
  ]
}
```

```
QPointF lp(x, y0), cp = lp;
path.moveTo(lp);
for (int i = 1; i < m_Serie.size(); i++)
{
    const auto& v = m_Serie.at((int)i);
    const double r = (v - minY) / (maxY - minY);
    const double y = siLerp(y_top, y_bottom, 1.0 - r);
    const double x_mid = x + x_step * 0.5;
    x += x_step;
    cp = QPointF(x, y);
    path.cubicTo(QPointF(x_mid, lp.y()), QPointF(x_mid, y), cp);
    lp = cp;
}
painter.save();
painter.setPen(QPen(QBrush(m_SerieColor), m_LineThickness));
painter.drawPath(path);
painter.restore();
```

GÉNÉRAL : INTERFACE D'ADMINISTRATION

The image displays the 'Station Meteo v1.0' application window and its 'Paramètres' (Settings) dialog box.

Station Meteo v1.0 Interface:







- Buttons: Rafraîchir, Paramètres, A propos.
- Date/Time: 23.06.2021 - 10:41:00 am
- Current Weather: Cloud icon, 23.7°C
- Pressure: 1014.9 hPa (graph)
- 12h Average Temperature: 24.6°C (graph)
- Humidity: 59.7% (graph)
- Location: Ville : Créteil
- Humidity: 90 %
- Pressure: 1020 hpa
- Description : couvert
- Forecast Table:

mer.	jeu.	ven.	sam.	dim.
13.7°C	13.2°C	11.4°C	14.1°C	16.6°C
16.3°C	19.5°C	20.9°C	24.7°C	23.2°C
Cloud icon	Cloud icon	Cloud icon	Cloud icon	Cloud icon
couvert	couvert	couvert	couvert	couvert
79 %	68 %	69 %	66 %	82 %
1020 hpa	1021 hpa	1020 hpa	1016 hpa	1013 hpa

Paramètres Dialog Box:

- Format de l'heure: ☐ 24H ☒ 12H
- Unité de Temperature: ☒ Celsius ☐ Fahrenheit ☐ Kelvin
- Police: MS Shell Dlg 2
- Styles: ☒ Jour ☐ Nuit
- Langue: FR
- Delai de mise à jour (Min): 10
- Balise Mer: IP 78.199.78.207, Port 48001
- Balise Ville: Clef API 467fdde738c8df563afa33a978b53563, Ville Créteil
- Buttons: Confirm (green check), Cancel (red X), Save (floppy disk)

BALISE VILLE : FORECAST ET MESURES INSTANTANÉES

<div> 15°C</div> <div>Ville : Paris Humidité : 78 % Pression : 1021 hpa Description : couvert</div>				
jeu.	ven.	sam.	dim.	lun.
13.1°C 18.4°C	11.7°C 21.6°C	13.8°C 25.4°C	16.1°C 19.4°C	14.7°C 22.5°C
				
couvert	nuageux	couvert	légère pluie	légère pluie
70 %	74 %	74 %	88 %	69 %
1021 hpa	1020 hpa	1015 hpa	1013 hpa	1016 hpa

← Mesures instantanées

← Prévisionnel

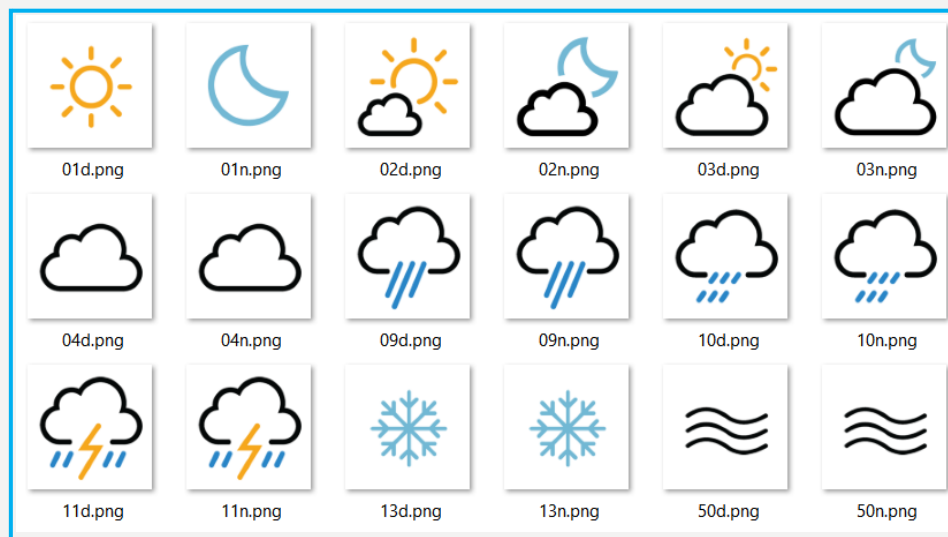
Nous utilisons un url par fonction

BALISE VILLE : DESCRIPTIF URL

- Url pour les mesures actuelles :
- <https://api.openweathermap.org/data/2.5/weather?q=paris,fr&appid=58e08b52cadfc9c96fc8354666cec712&lang=fr>
- Url pour le prévisionnel :
- <https://api.openweathermap.org/data/2.5/forecast?q=paris,fr&appid=58e08b52cadfc9c96fc8354666cec712&lang=fr>

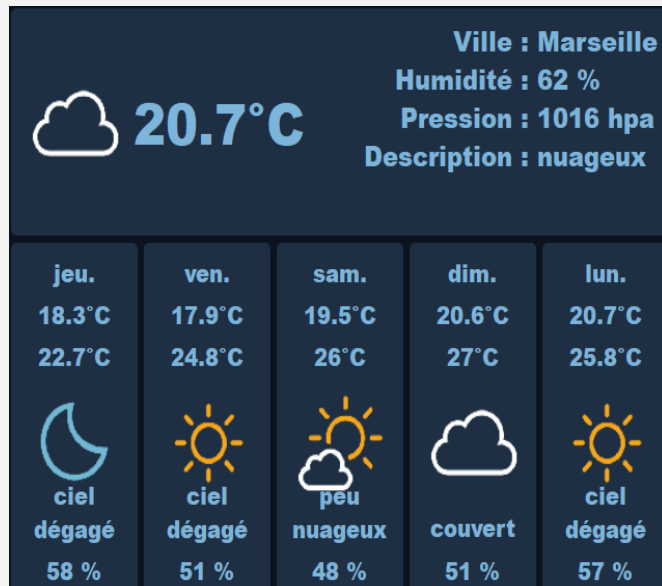
BALISE VILLE : DESCRIPTIF FICHIER JSON DE L'API

JSON	Données brutes	En-têtes
Enregistrer	Copier	Tout réduire
Tout dével		
coord:	{...}	
weather:		
0:		
id:	804	
main:	"clouds"	
description:	"couvert"	
icon:	"04d"	
base:	"stations"	
main:		
temp:	288.98	
feels_like:	288.79	
temp_min:	287.74	
temp_max:	290.66	
pressure:	1020	
humidity:	83	
visibility:	10000	
wind:		
speed:	0.89	
deg:	42	
gust:	4.02	
clouds:	{...}	
dt:	1624454739	
sys:	{...}	
timezone:	7200	
id:	2988507	
name:	"Paris"	
cod:	200	



BALISE VILLE : CHANGEMENT DE VILLE

Grâce au paramètre de changement de la ville disponible dans notre URL, nous avons ajouté une option de changement de ville dans notre menu paramètre, qui viendra changer la ville recherchée dans l'Url afin d'en afficher les mesures.



Paramètres

Général

Format de l'heure ☐ 24H ☒ 12H

Unité de Temperature ☒ Celsius ☐ Fahrenheit ☐ Kelvin

Police

Styles ☐ Jour ☒ Nuit

Langue

Delai de mise à jour (Min)

Balise Mer

IP

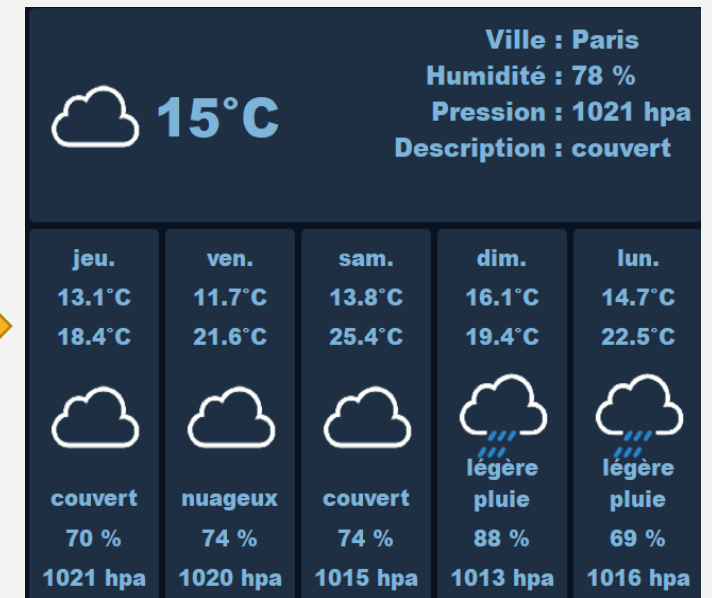
Port

Balise Ville

Ville

Ville

☒ ☐ ☐



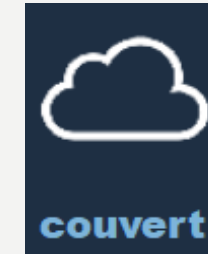
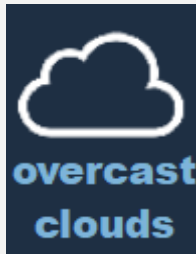
<https://api.openweathermap.org/data/2.5/weather?q=paris,fr&appid=58e08b52cadfc9c96fc8354666cec712&lang=fr>

BALISE VILLE : CHANGEMENT DE LANGUE

Notre API nous permet de modifier la langue (par défaut anglaise) des descriptions météo reçues.






Nous avons donc utilisé cette fonction dans notre programme.

Cette fonction sera utilisée lorsque nous modifierons la langue dans le menu paramètres.



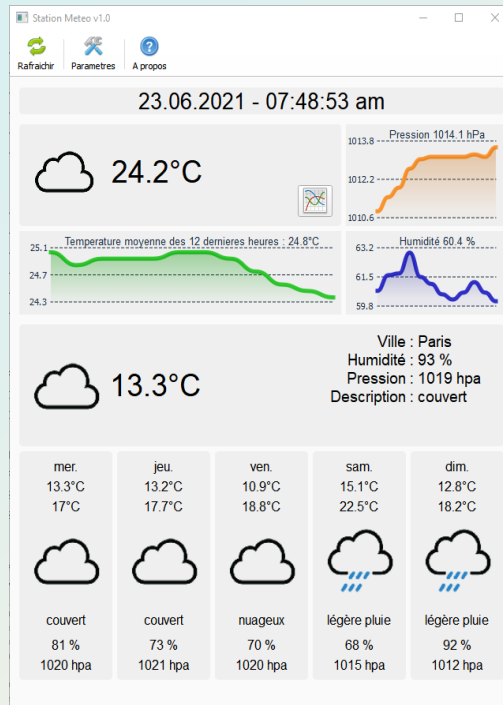
<https://api.openweathermap.org/data/2.5/weather?q=paris,fr&appid=58e08b52cadfc9c96fc8354666cec712&lang=fr>

BALISE VILLE : DESCRIPTIF PRÉVISIONNEL

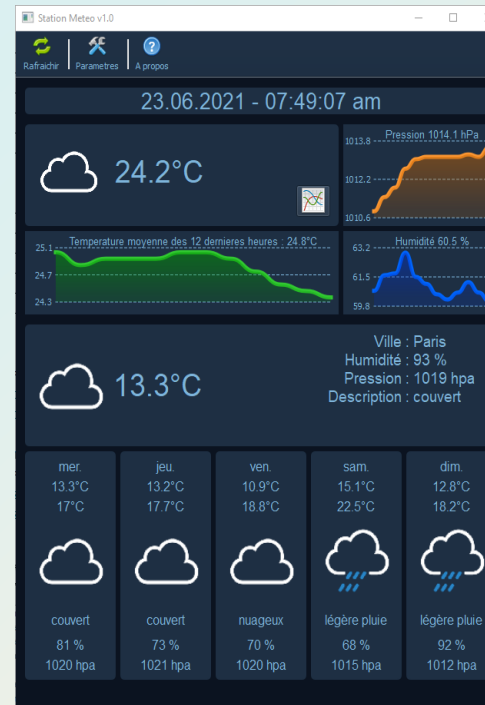
Jour :	jeu.	ven.	sam.	dim.	lun.
Température Minimale :	11.9°C	10.6°C	11.3°C	12.9°C	13.7°C
Température Maximale :	18.9°C	21.1°C	22.1°C	22.1°C	23°C
Icone :					
Description :	ciel dégagé	ciel dégagé	peu nuageux	légère pluie	nuageux
Humidité :	55 %	63 %	65 %	62 %	55 %
Pression :	1016 hpa	1016 hpa	1017 hpa	1016 hpa	1015 hpa

GÉNÉRAL : CHANGEMENT DE THÈME JOUR / NUIT

Thème Jour



Thème Nuit



Configuration par fichier de style QSS

```
QMainWindow { background-color:#0B1320; }
QDialog { background-color:#0B1320; }
QToolBar { background-color: #1E2F43; }
QPushButton { color: #7DB4DA;}
QPushButton:hover { background-color: #1F4083; }
QPushButton {
    background-color: #1E2F43;
    color: #7DB4DA;
}
QPushButton:hover
{
    background-color: #1F4083;
    color: #E0E0E0;
}
QLabel{ color: #7DB4DA;}
QRadioButton
{
    color: #7DB4DA;
    padding: 0 5 0 0;
}
QRadioButton::indicator:checked
{
    height: 10px;
    width: 10px;
    border-style:solid;
    border-radius:5px;
    border-width: 2px;
    border-color: #7DB4DA;
    background-color:#7DB4DA;
}
QRadioButton::indicator:!checked
{
    height: 10px;
    width: 10px;
    border-style:solid;
```

GÉNÉRAL : CHARGEMENT / SAUVEGARDE DES PARAMÈTRES

Chargement depuis un fichier INI

```
void GlobalSettings::SettingsStruct::Load(const QString& vFilePathName)
{
    GlobalSettings::SettingsStruct _default;

    QSettings config(vFilePathName, QSettings::IniFormat);

    m_FormatHourEnum = (FormatHourEnum)config.value("FormatHourEnum", (int)_default.m_FormatHourEnum).toInt();
    m_TemperatureUnit = (TemperatureUnitEnum)config.value("TemperatureUnit", (int)_default.m_TemperatureUnit).toInt();
    m_Ville = config.value("Ville", _default.m_Ville).toString();
    m_FontFamily = config.value("FontFamily", _default.m_FontFamily).toString();
    m_ApiKey = config.value("ApiKey", _default.m_ApiKey).toString();
    m_IP = config.value("IP", _default.m_IP).toString();
    m_Port = config.value("Port", _default.m_Port).toString();
    m_Language = config.value("Language", _default.m_Language).toString();
    m_Style = (StyleEnum)config.value("Style", (int)_default.m_Style).toInt();
    m_RefreshDelayInMinutes = config.value("RefreshDelayInMinutes", _default.m_RefreshDelayInMinutes).toUInt();
}
```

Sauvegarde dans un fichier INI

```
void GlobalSettings::SettingsStruct::Save(const QString& vFilePathName)
{
    QSettings config(vFilePathName, QSettings::IniFormat);

    config.setValue("FormatHourEnum", (int)m_FormatHourEnum);
    config.setValue("TemperatureUnit", (int)m_TemperatureUnit);
    config.setValue("Ville", m_Ville);
    config.setValue("FontFamily", m_FontFamily);
    config.setValue("ApiKey", m_ApiKey);
    config.setValue("IP", m_IP);
    config.setValue("Port", m_Port);
    config.setValue("Language", m_Language);
    config.setValue("Style", (int)m_Style);
    config.setValue("RefreshDelayInMinutes", m_RefreshDelayInMinutes);

    config.sync();
}
```

Le contenu du fichier INI

```
[General]
FormatHourEnum=0
TemperatureUnit=0
Ville=marseille
FontFamily=MS Shell Dlg 2
ApiKey=467fdde738c8df563afa33a978b53563
IP=78.199.78.207
Port=48001
Language=FR
Style=0
RefreshDelayInMinutes=10
```

GÉNÉRAL : GESTION MULTILINGUE

Traduction de l'application

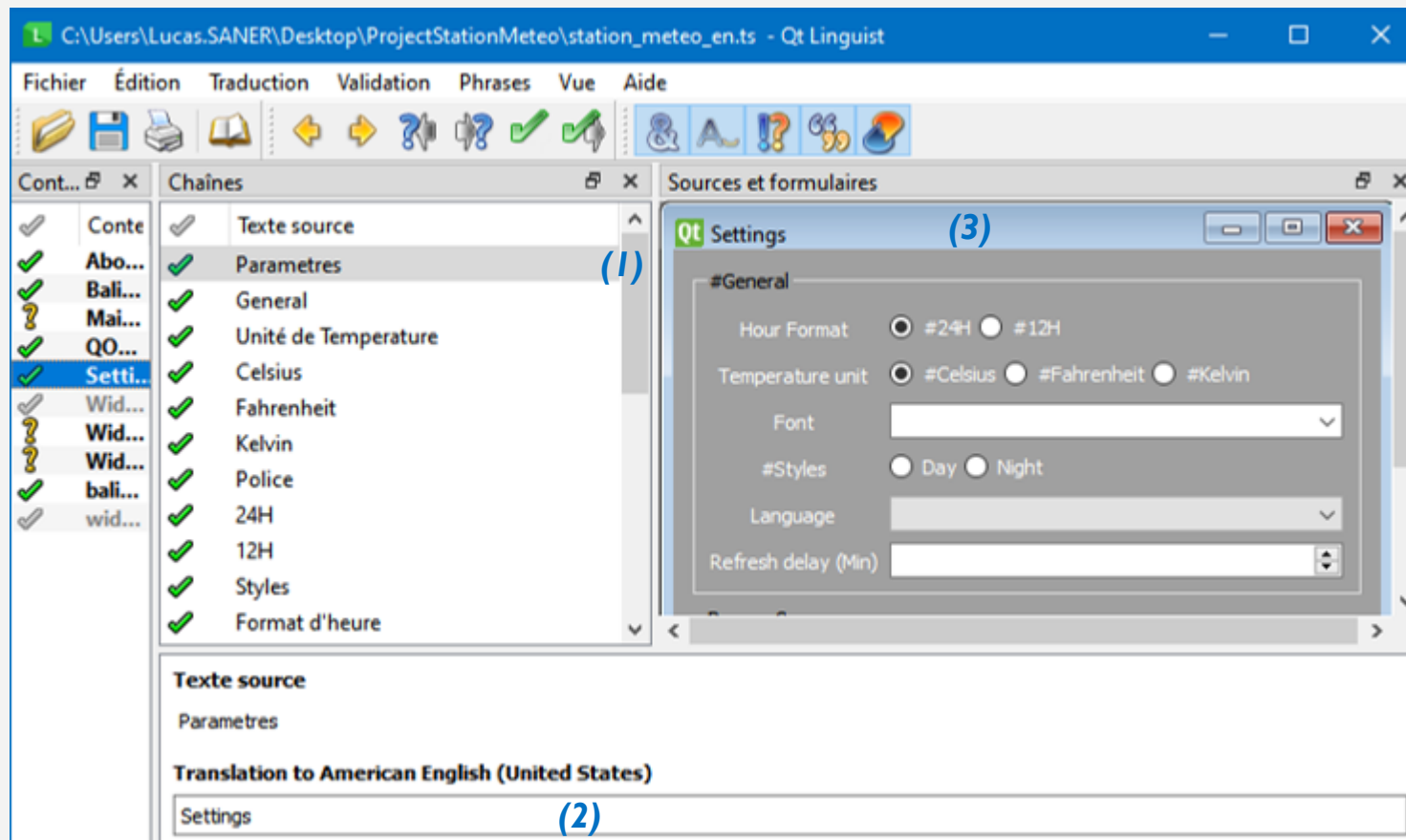
```
ui->customGraphPressure->SetSerieName("Pression ");
```



```
ui->customGraphPressure->SetSerieName(QObject::tr("Pression "));
```

GÉNÉRAL : GESTION MULTILINGUE

Traduction de l'application



GÉNÉRAL : GESTION MULTILINGUE

Traduction de l'application

Paramètres

Général

Format de l'heure ☐ 24H ☒ 12H

Unité de Temperature ☒ Celsius ☐ Fahrenheit ☐ Kelvin

Police

Styles ☐ Jour ☒ Nuit

Langue

Delai de mise à jour (Min)

Balise Mer

IP

Port

Balise Ville

Clef API

Ville

✓ ✗ 💾



Settings

General

Hour Format ☐ 24H ☒ 12H

Temperature unit ☒ Celsius ☐ Fahrenheit ☐ Kelvin

Font

Styles ☐ Day ☒ Night

Language

Refresh delay (Min)

Beacon Sea

IP

Port

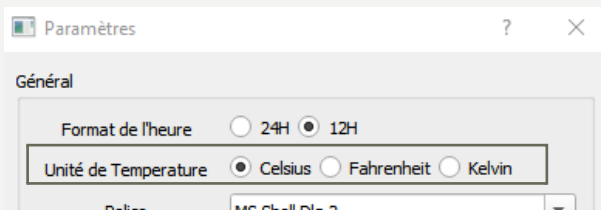
Beacon City

Api Key

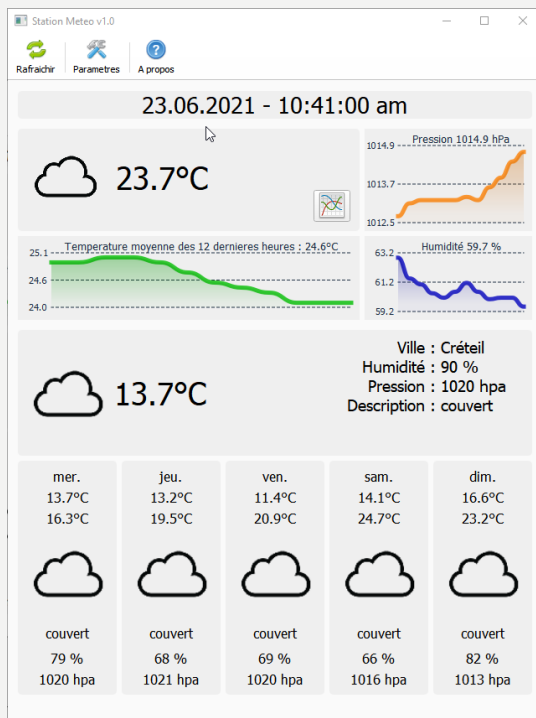
City

✓ ✗ 💾

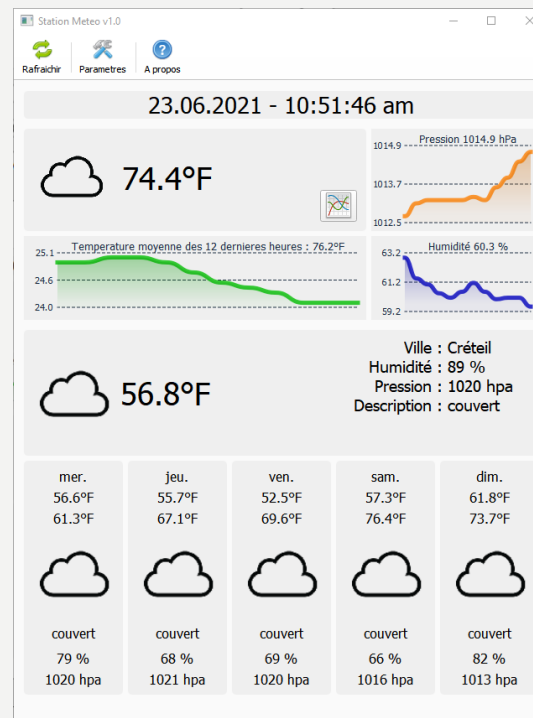
GÉNÉRAL : CHANGEMENT DE L'UNITÉ DES MESURES



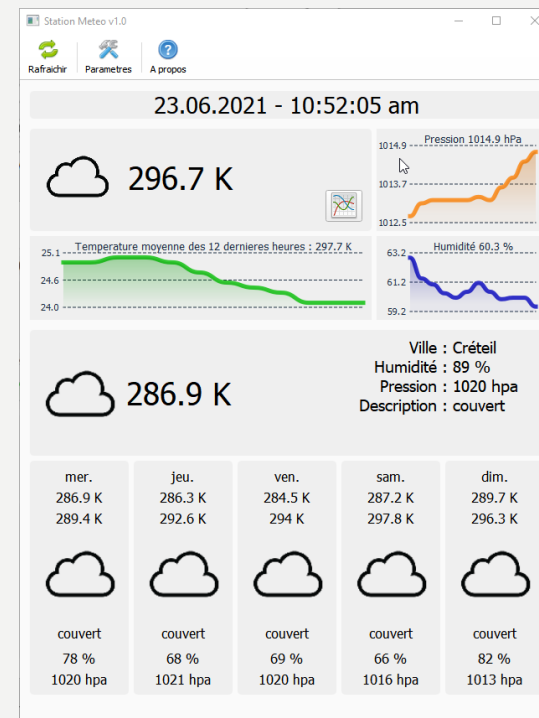
Affichage en Celsius
(default)



Affichage en Fahrenheit

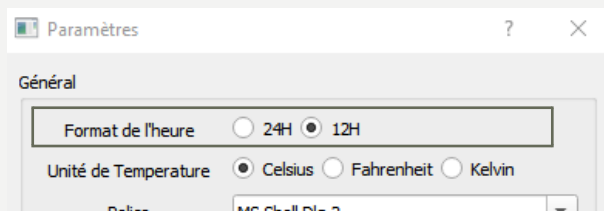


Affichage en Kelvin

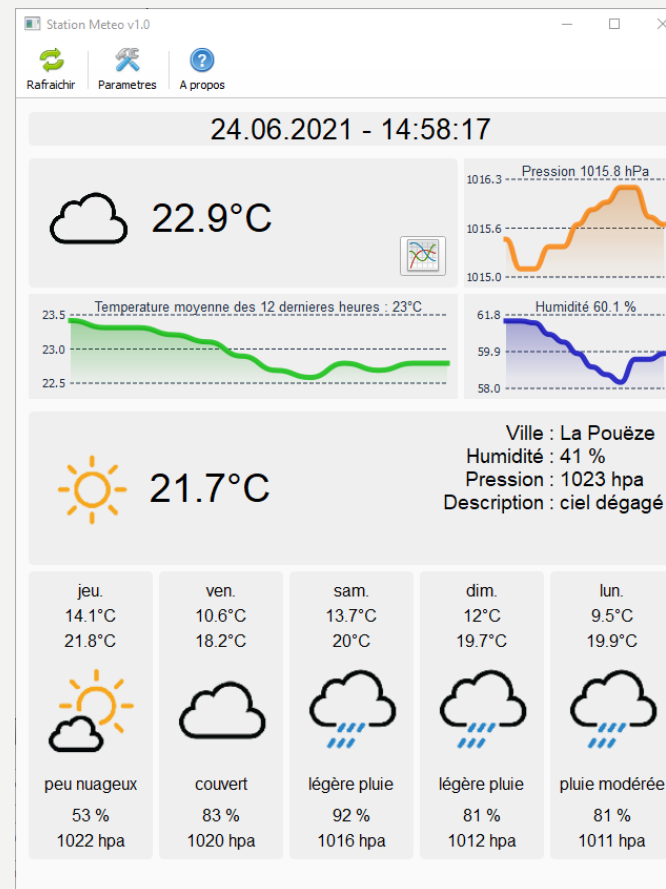
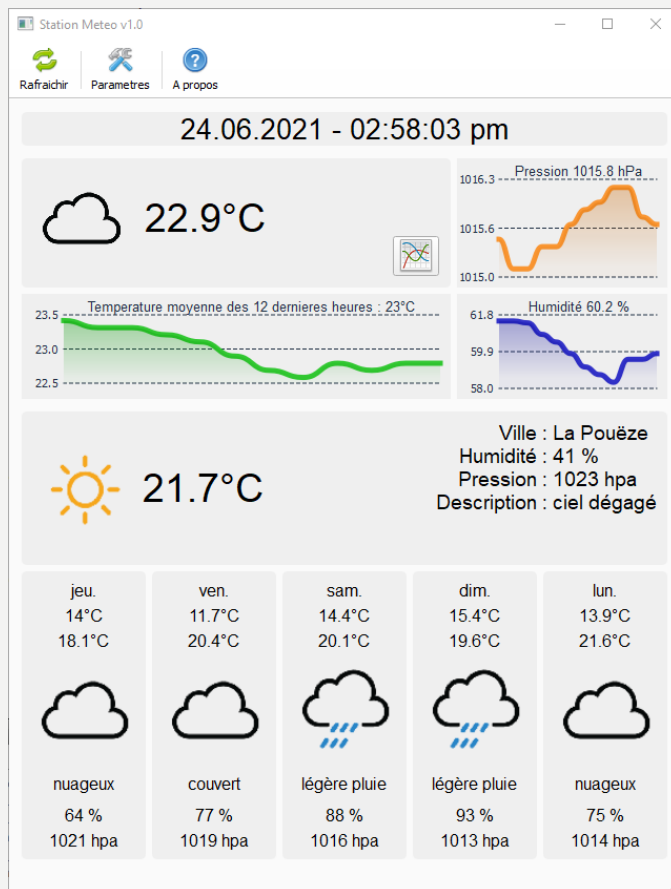


GÉNÉRAL : CHANGEMENT DE FORMAT D'HEURE

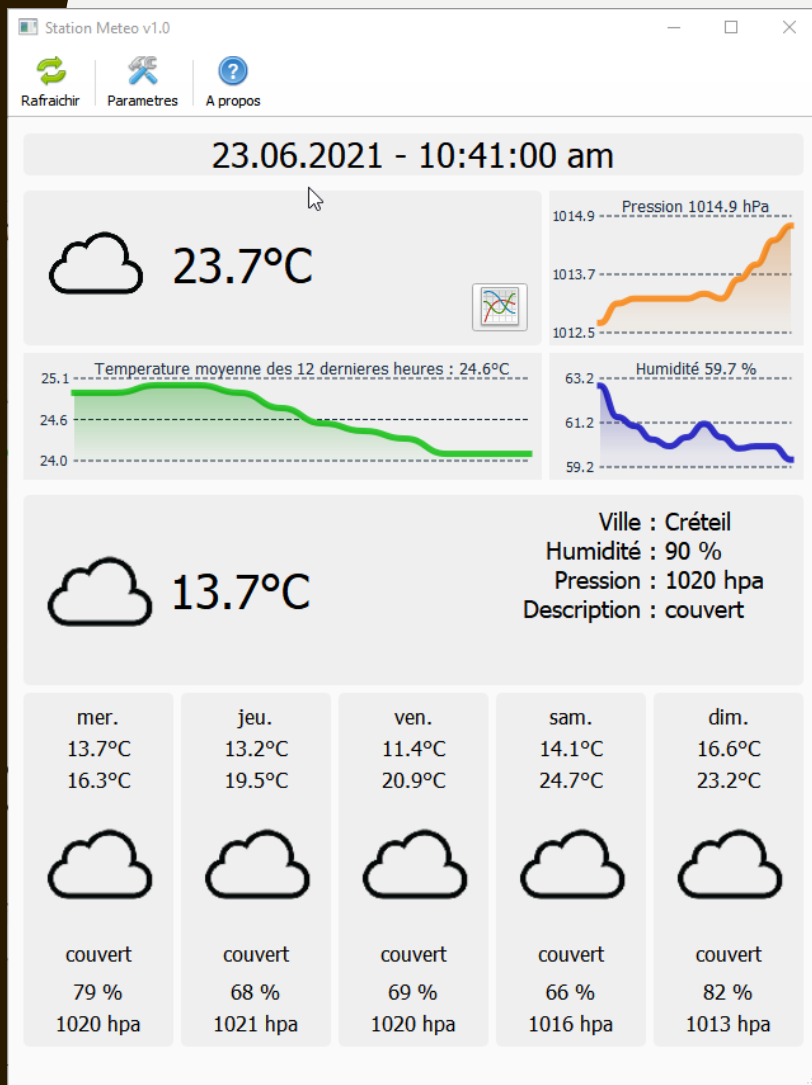
Format 12H



Format 24H



GÉNÉRAL : CHANGEMENT DE POLICE



Sélection de la police « Mistral »

Paramètres

Général

Format de l'heure ☐ 24h ☒ 12h

Unité de Temperature ☒ Celsius ☐ Fahrenheit ☐ Kelvin

Police **Mistral**

Styles ☒ Jour ☐ Nuit

Langue FR

Delai de mise à jour (Min) 10

Balise Mer

IP 78.199.78.207

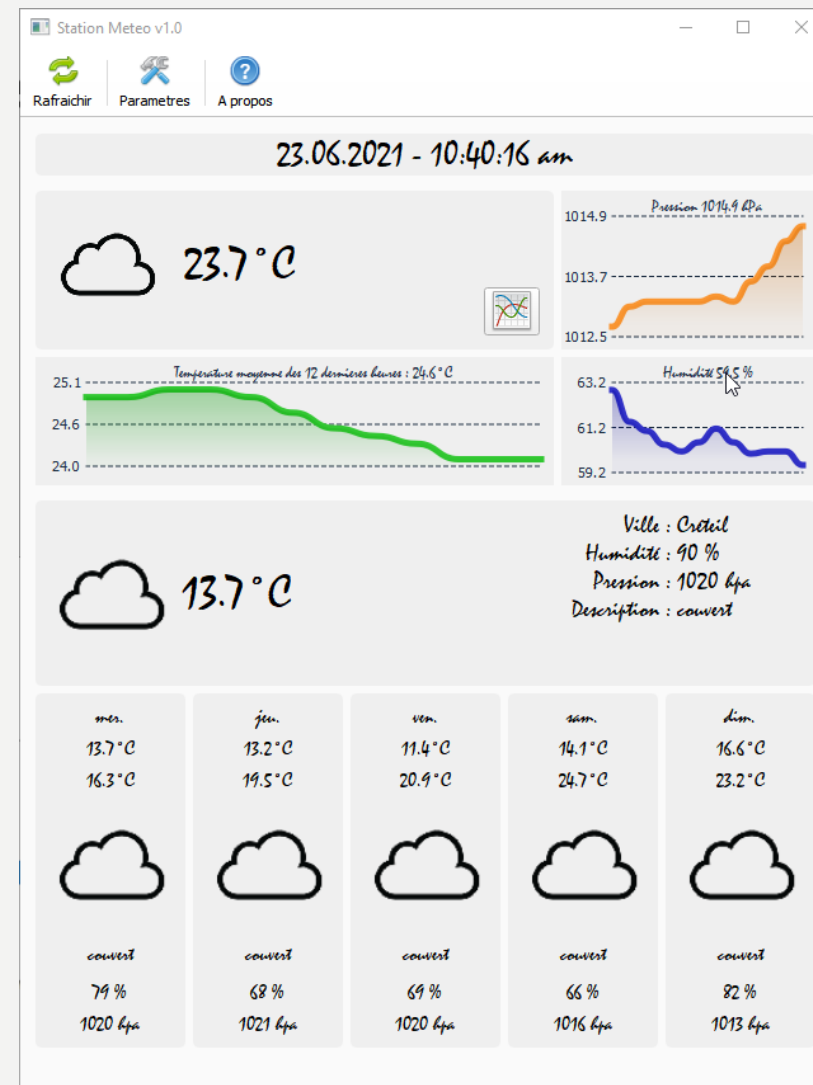
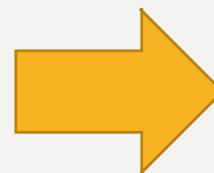
Port 48001

Balise Ville

Clef API 467fdde738cdf563afa33a978b53563

Ville Créteil

✓ ✗ 💾





DÉMONSTRATION

CONCLUSION

CONCLUSION

- Respect de la spécification :
 - Toutes les fonctionnalités demandées et la partie facultative ont été implémentées
- Evolutions possible :
 - Convertir les requêtes serveur en asynchrone
 - Ajouter d'autres langues
 - Pouvoir proposer à la lecture plus de paramètres sur l'api web
 - Proposer un graphique prévisionnel des 5 jours
 - Pouvoir se connecter à plus d'un IP dans le cas de la balise mer
 - Personnalisation du thème par l'utilisateur
- Apports personnel :



REMERCIEMENTS