

	<h1 style="text-align: center;">Introduction à l'IIoT Industrial Internet of Things</h1>
Sommaire :	
I - L'IIoT (Industrial Internet of Things).....	1
II - L'IIoT (Industrial Internet of Things).....	1
III - Protocoles Applicatifs de l'IIoT.....	1
IV - Architecture d'une solution IIoT.....	2
V - Les plateformes IIoT.....	3

I - L'IIoT (Industrial Internet of Things)

L'**Internet of Everything (IoE)** est la connexion en réseau de **personnes**, de **processus**, de **données** et d'**objets**.

L'objectif de l'**Internet of Things (IoT)** est de connecter ce qui ne l'est pas encore, rendant ainsi les objets accessibles sur Internet.

II - L'IIoT (Industrial Internet of Things)

Dans le secteur industriel, l'**IIoT (Industrial Internet of Things)** fonctionne sur la base de capteurs connectés placés sur les différents équipements de la chaîne de production. Ces capteurs génèrent une multitude de données qui permettent de moduler et d'optimiser les processus industriels en temps réel.

Les solutions **IIoT** sont principalement utilisées dans les secteurs de la **fabrication** et de l'**énergie** :

- Dans le secteur de la **fabrication**, l'**IIoT** permet par exemple d'avoir de la visibilité sur le fonctionnement de l'usine. Les données fournies par les capteurs des machines sont analysées en temps réel et transférées aux systèmes de contrôle, ce qui améliore l'efficacité opérationnelle et métier.
- Les entreprises du secteur de l'**énergie** utilisent l'**IIoT** pour mieux surveiller les ressources sur le terrain. Les appareils **IIoT** sont ainsi capables de recueillir des données en temps réel sur les performances des réseaux électriques et des réseaux d'eau même lorsque les ressources sont distribuées sur de vastes zones géographiques.

III - Protocoles Applicatifs de l'IIoT

Un **protocole applicatif** est un ensemble de règles définissant le mode de communication entre deux applications informatiques. Il se base sur les protocoles de transport **TCP** et **UDP**.

L'infrastructure du web classique n'est pas adaptée à la majorité des applications **IIoT**. En effet, certains objets connectés, dits **contraints**, sont limités par de petits microcontrôleurs avec de petites quantités de mémoire (flash et RAM). Par conséquent, pour palier à ces contraintes il faut des protocoles moins verbeux avec un nombre limité de messages et de petites tailles.

- On peut identifier trois familles de protocoles applicatifs :
- Protocoles de messagerie (publication et abonnement) : **MQTT**, **XMPP** et **AMQ** ;
 - Protocoles de transfert web (protocole **HTTP**) : **CoAP**, **API ReST** ;
 - Protocole réseau (connexion **TCP**) : **Websocket**.

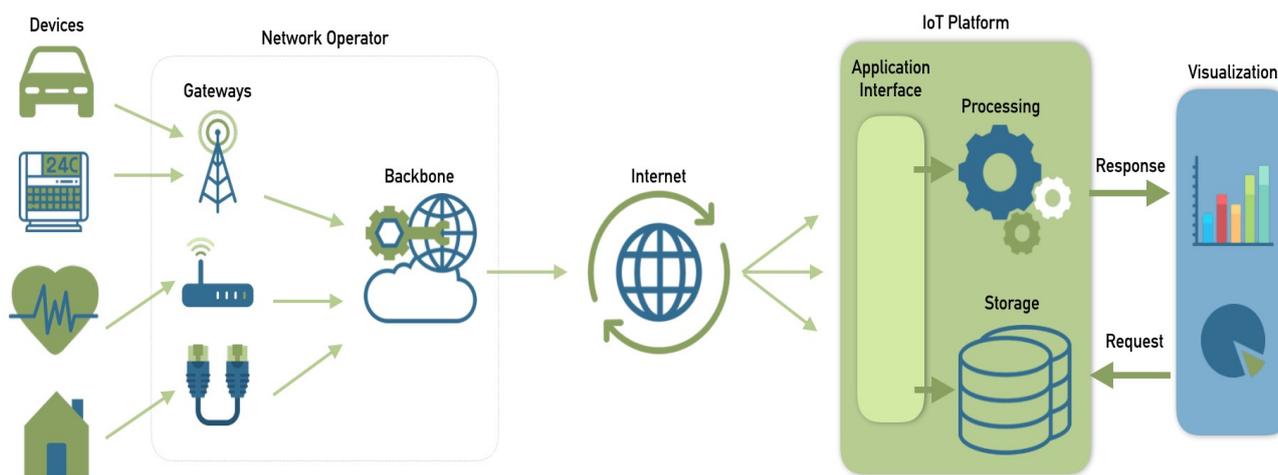
Les protocoles applicatifs les plus adaptés aux objets connectés contraints (limités en ressources d'énergie et de calcul) sont **MQTT** et **CoAP**.

Résumé des caractéristiques des principaux protocoles applicatifs utilisés :

Protocole	MQTT	XMPP	AMQP	CoAP	WebSocket	API REST
Type de protocole	Messagerie	Messagerie	Messagerie	Transfert web	Réseau	Transfert web (HTTP)
Modèle de communication	Publish/Subscribe	Publish/Subscribe Request/Response	Producer/Consumer	Request/Response	bidirectionnel	Request/Response
Transport	TCP/IP	TCP/IP	TCP/IP	UDP/IPv6 /6LowPAN	TCP/IP	TCP/IP
Sécurité	TLS/SSL	TLS/SSL	TLS/SSL	DTLS	TLS/SSL	TLS/SSL
Format	Binaire, Text (json, xml, csv)	XML	Binaire, Text	Binaire, Text	Binaire, Text	Binaire, Text
Contraintes sur les objets connectés	Fortes	Faibles/Moyennes	Moyennes	Fortes	Faibles	Faibles
Principaux framework	Emqtt, HiveMQ, Mosquitto, Eclipse Paho	Jabber, XMPPFramework	RabbitMQ, StormMQ	Eclipse Californium, nCoAP	jetty, websocket, Apache Tomcat	Django REST, Apache Tomcat, Node.js, Ruby on Rails

IV - Architecture d'une solution IoT

L'architecture complète d'une solution **IoT** se compose des éléments suivants :



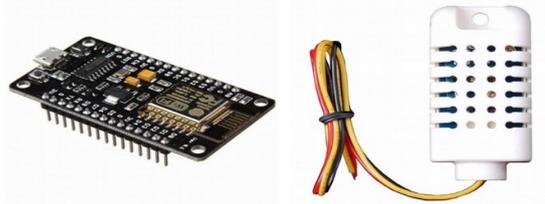
Profiter des avantages de connecter tout ce qu'il est possible de connecter est très certainement un objectif intéressant. Toutefois, il faut utiliser des **pratiques de sécurité optimales** afin de garantir que ces **connexions resteront privées et sécurisées**.

Aujourd'hui afin de réaliser un périphérique **IoT** (un objet connecté), il est possible d'utiliser des périphériques bon marché programmables par l'utilisateur, comme le **Raspberry Pi**, l'**Arduino**, l'**ESP8266**, l'**ESP32** ou d'autres plates-formes de prototypage.

Raspberry Pi
+
Capteur DS18B20



ESP8266 ou ESP32
+
Capteur DHT22



V - Les plateformes IoT

Pour piloter ses objets connectés, collecter, traiter et stocker ses données, nous aurons besoin d'une **plateforme IoT** souvent issue du « **cloud computing** ».

Le **cloud computing** (en français, « **informatique dans les nuages** ») fait référence à l'utilisation de la mémoire et des capacités de calcul des ordinateurs et des serveurs répartis dans le monde entier et liés par un réseau.

Types de services Cloud	Rôle	Exemples
IaaS (Infrastructure as a Service)	Ils donnent accès aux ordinateurs, physiques et virtuels, ainsi qu'aux fonctions et à l'espace de stockage du réseau. IaaS c'est comme la location d'une cuisine. Vous pouvez utiliser tous les différents appareils (mixeurs, blenders, éviers) et vous avez la possibilité de louer une cuisine dotée de meilleurs appareils si vous en avez besoin.	Amazon Elastic Compute Cloud (Amazon EC2), Rackspace, Google Compute Engine
PaaS (Platform as a Service)	Ils comprennent des environnements de programmation, des plateformes de test d'application et des lanceurs d'application. PaaS c'est comme une sortie au restaurant. Vous ne gérez pas les appareils de la cuisine, mais vous pouvez demander au serveur ou au chef de faire les choses selon vos souhaits.	AWS Elastic Beanstalk, Microsoft Azure, Google App Engine
SaaS (Software as a Service)	Ils sont les applications et logiciels réels fournis via Internet. SaaS c'est comme un buffet à volonté. Vous avez accès à n'importe quelle nourriture qui y est servie. Vous ne contrôlez pas ce qui est fait ni comment, mais vous pouvez en utiliser autant que vous le voulez.	Dropbox, Slack, Spotify, YouTube, Microsoft Office, Gmail

Les offres PaaS (Platform as a Service)

Ici le fournisseur est un « **cloud provider** » qui vous propose une bibliothèque de briques logicielles pouvant être déployées à la demande sur des machines de leur data-centers. Dans ce modèle, nul besoin de se soucier de la gestion des serveurs, opérée par le « **cloud provider** ».

Trois plateformes de type **PaaS (Platform as a Service)** offrant des services **IoT** aboutis se partagent le marché : **Amazon Web Services (AWS)**, **Microsoft Azure** et **Google Cloud Platform**.

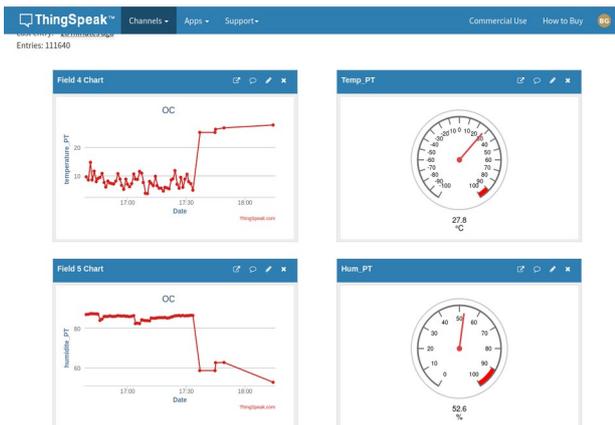
Les offres Open Source

Le logiciel des solutions open-source est distribué librement, et son code source est ouvert à tous. Une telle solution est maintenue par une communauté de développeurs bénévoles et/ou par une entreprise qui utilise cette même solution, ou qui commercialise du support et/ou de l'intégration autour de l'outil. Bien souvent, ces entreprises proposent une version opérée de la solution, une sorte de version **SaaS**. D'autres mettent à disposition une version open-source, tout en vendant une version entreprise comportant des fonctions supplémentaires.

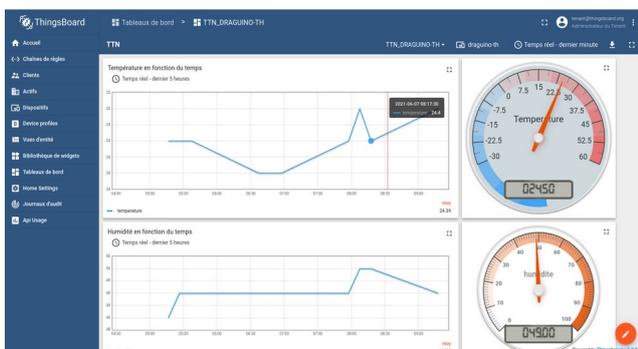
Voici une liste de **plateformes IoT** open-source :

- Freeboard.io (que la vue) ;
- Dweet.io (sans la vue) ;
- Octoblu (sans la vue) ;
- ThingSpeak ;
- ThingsBoard.io ;
- Kuzzle.io ;
- Ubidots ;
- Kibana ;
- WebThings ;
- Home Assistant (HA).

ThingSpeak :



ThingsBoard :



Home Assistant :

