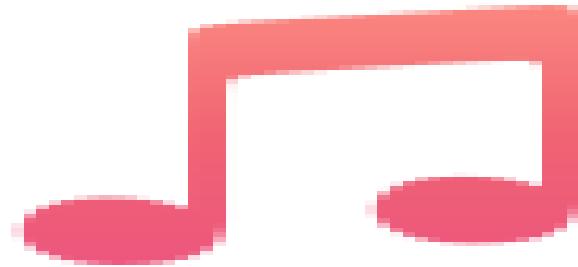


Architecture et Sécurité des Objets Connectés

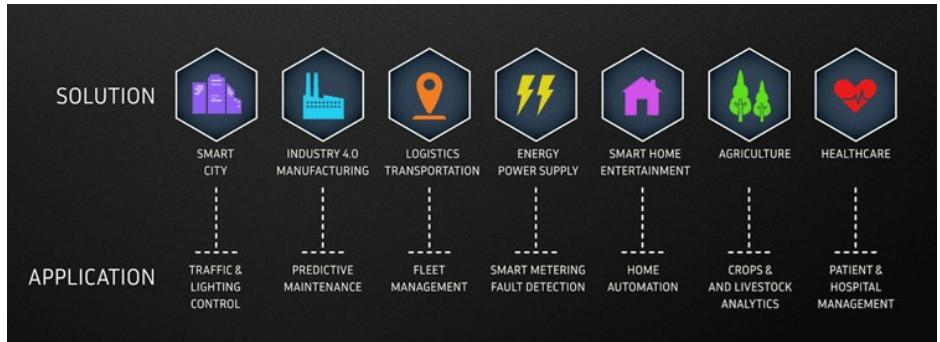
Exploration de la Chaîne IoT, de l'Objet au Cloud

Mansour GUEYE



Qu'est-ce que l'Internet des Objets ?

« **infrastructure mondiale** pour la société de l'information, qui permet de disposer de services évolués en interconnectant des objets (physiques ou virtuels) grâce aux **technologies de l'information** et de la communication interopérables existantes ou en évolution »

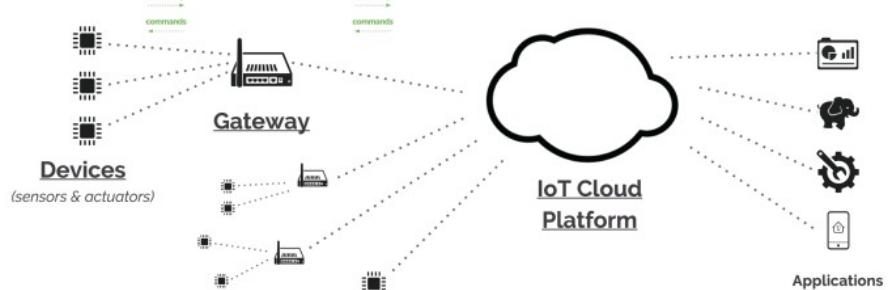


Architecture Typique des Solutions IoT

Les solutions IoT mettent généralement en œuvre une architecture composée de plusieurs couches clés.

- ✓ **Objets contraints** : Ils font transiter leurs données.
- ✓ **Passerelles** : Elles acheminent les données des objets à travers le réseau.
- ✓ **Plateformes IoT** : Elles hébergent les serveurs qui reçoivent les données.
- ✓ **Applications** : Elles se connectent aux plateformes IoT.

Cette structure en 3 couches (Objets, Passerelles, Plateformes) est fondamentale



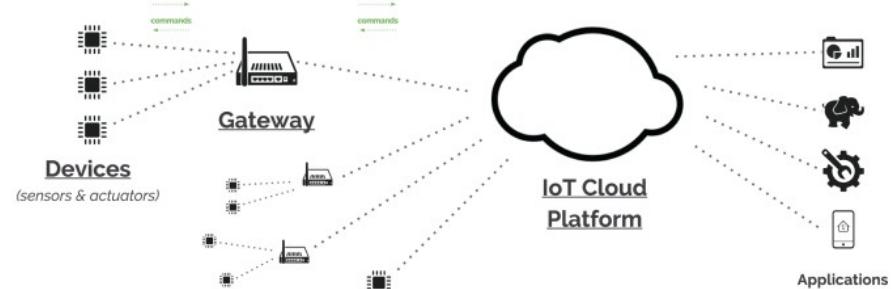
Standardisation de l'Architecture IoT

En raison du développement rapide de l'Internet des Objets, il était important de définir une architecture type pour uniformiser les solutions.

L'Internet Architecture Board (IAB) a édité la RFC 7542 dans ce cadre.

La communication entre objets

- La communication entre les objets vers les plateformes IoT
- La communication des objets vers une passerelle
- le partage de données en "back-end"



Fonctionnalités des Plateformes IoT

Les **plateformes IoT** possèdent plusieurs fonctionnalités-clés pour gérer le cycle de vie des données et des objets.

- La **connectivité et le routage** des messages.
- L'enregistrement **des objets et leur gestion**.
- La **gestion et le stockage** des données.
- La **gestion des événements**, l'analyse et la représentation des données.
- L'utilisation d'une **API pour l'intégration des applications** .



Objets Contraints et Passerelles

Nous explorerons ces couches sous différents angles

- **Réseau** : Couches réseaux, standards de communication sans fil pour l'IoT.
- **Sécurité** : Mécanismes de défense contre les attaques réseau ou logiciel .
- **Matériel** : Le domaine de l'embarqué (micro-contrôleur, capteurs/actionneurs, bus de données, puces radios)
- **Logiciel** : Solutions pour programmer un objet contraint (ex: RIOT).



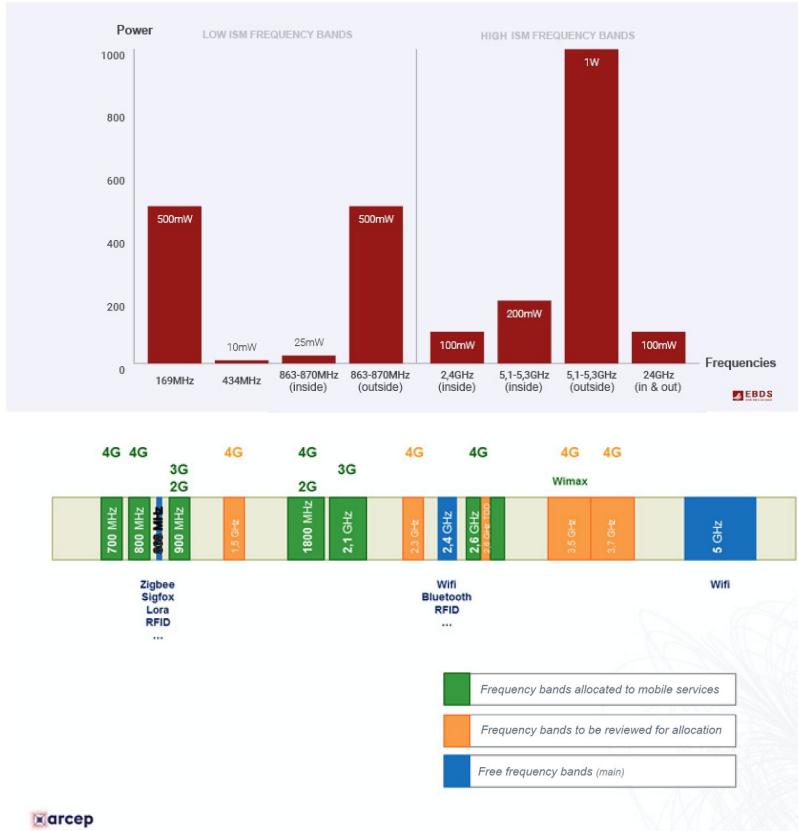
Gestion des Données : Défis et Solutions

Le nombre croissant **d'objets connectés** va générer une quantité gigantesque de données (estimations pour 2025 : 38 milliards d'objets, 175 milliards de téraoctets de données).

- ✓ Il est crucial de stocker ces **données** de manière **fiable et pérenne**, notamment dans des bases de données .
- ✓ L'expérience en **systèmes de gestion de base de données** est nécessaire pour faire face à cette problématique.
- ✓ L'amélioration continue des systèmes de base de données existants sera requise pour gérer ces **volumes**.

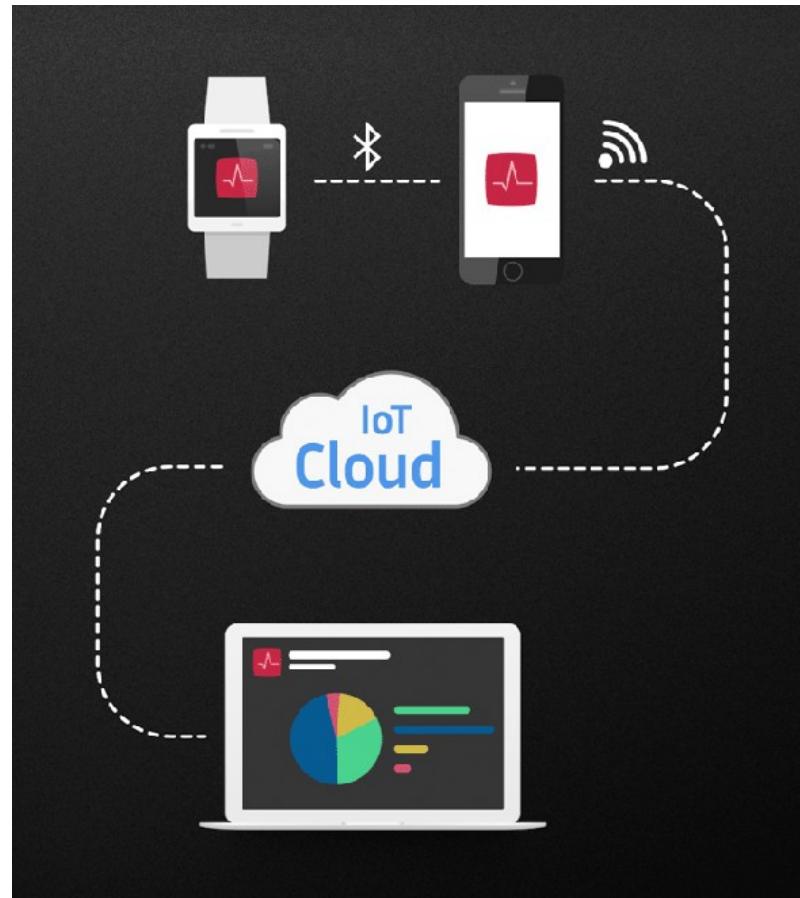


Fréquences sous Licence : Avantages et Inconvénients



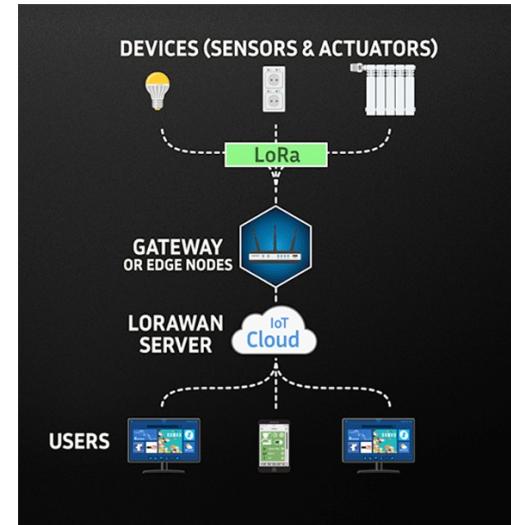
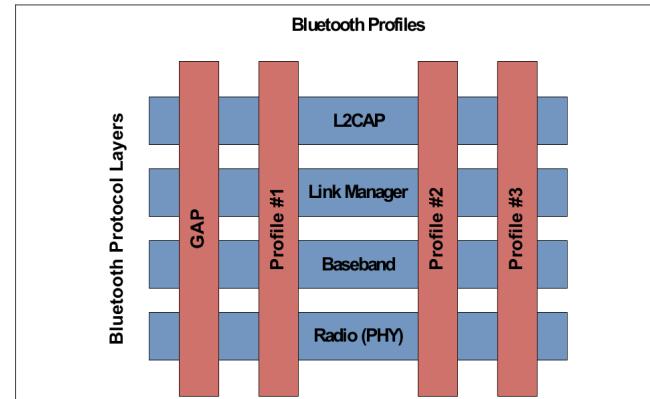
- **Avantages :** Utiliser des fréquences sous licence, via un opérateur, garantit une qualité de service et une bonne connexion.
- **Inconvénients :** Cela nécessite de souscrire un forfait et implique une dépendance à l'opérateur et à son infrastructure

Exemples d'Applications IoT



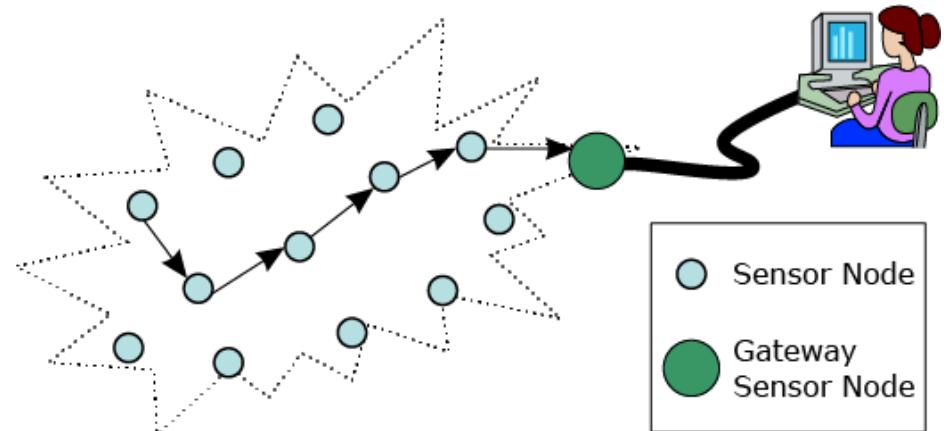
BLE et GATT

- La spécification **BLE** décrit des profils/services **GATT** pour l'interopérabilité entre objets (ex: montre et smartphone).
- GATT introduit la notion de **client et de serveur**.
- **Actions possibles** : Consultation (lecture), modification, notification .
- Pour recevoir/accéder aux données, le client (smartphone) établit une connexion avec le serveur



Exemple : Bâtiment Intelligent (IEEE 802.15.4)

- Le protocole **IEEE 802.15.4** est adapté à une utilisation pour un bâtiment intelligent.
- Il fonctionne sur le même principe que les **réseaux IP** avec un modèle en couches.
- Certaines couches ont été adaptées pour être utilisables avec des **capteurs très contraints** (très peu de mémoire, quelques dizaines de kilooctets)



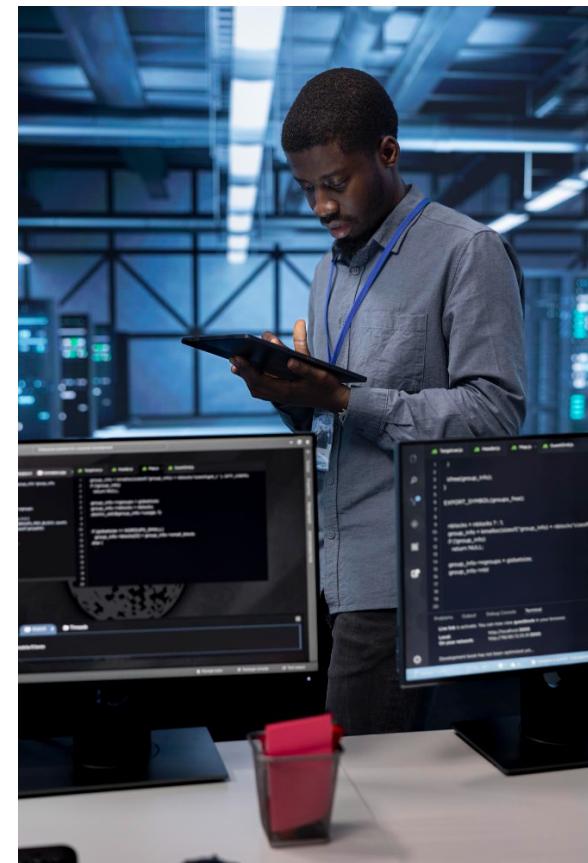
Autonomie dans les Réseaux 802.15.4

- Dans un réseau **802.15.4**, les capteurs restent endormis la plupart du temps .
- Cela implique une **latence importante**.
- Même si cela engendre des contraintes de **routage** fortes, l'autonomie globale des **capteurs** peut aller jusqu'à plusieurs années



Les Défis des Expérimentations IoT Traditionnelles

- **Coût financier** de l'équipement.
- **Coût en temps** pour les expérimentations.
- **Processus fastidieux** : programmer cartes une à une, recharger batteries, déployer selon topologie physique, mettre en place solution pour récupérer les logs



Architecture Matérielle des Objets de l'IoT

Exploration de la Chaîne IoT, de l'Objet au Cloud

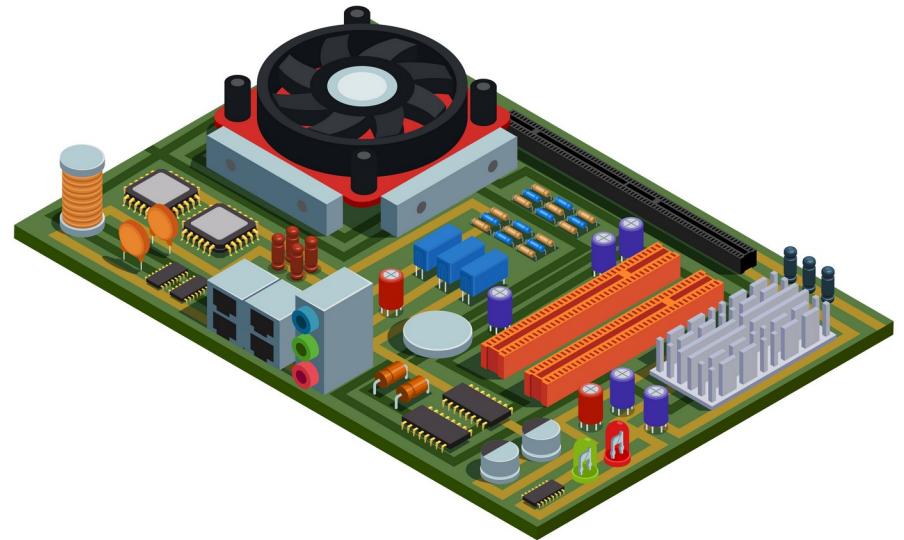
Qu'est-ce qu'un Système Embarqué ?

Un objet connecté est une application grand public d'un système embarqué communicant.

- Présents dans l'industrie depuis la fin des années soixante.
- Rendus possibles par la miniaturisation des transistors et des circuits intégrés.

Définition : Système électronique et informatique, conçu pour une tâche spécifique, autonome, fonctionnant si besoin en temps réel.

Optimisés et doivent répondre à des contraintes.



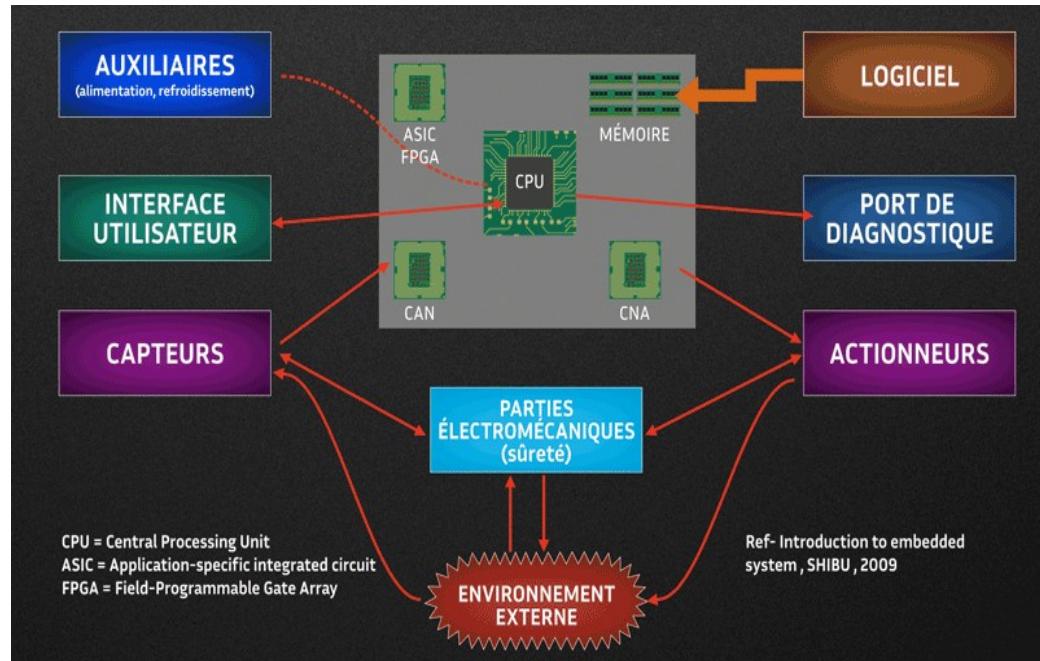
Contraintes des Systèmes Embarqués et Objets Connectés

Contraintes Générales : Coût de fabrication, Puissance de traitement/mémoire, Consommation énergétique, Encombrement optimal ("form factor").

Contraintes Spécifiques IoT : Bande passante pour l'envoi de messages, Portée radio ou technologie filaire.

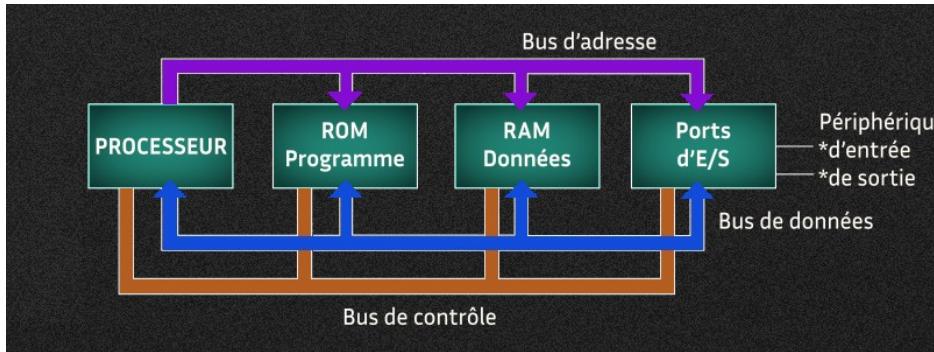


Composants Matériels d'un Système Embarqué



- **Micro-contrôleur (MCU)** : Cœur, coordonne E/S .
- **Mémoire (ROM/RAM)** : Stockage firmware/variables
- **Capteurs/Actionneurs** : Interaction avec l'environnement
- **Alimentation électrique** : Fournit et régule le courant .
- **Bus de communication locaux** : Relient MCU aux périphériques (CAN, I2C, SPI, UART, etc.).
- **Puces communicantes externes** : Pour communications externes (radio, filaire).
- **JTAG/JLINK** : Pour reprogrammer le firmware

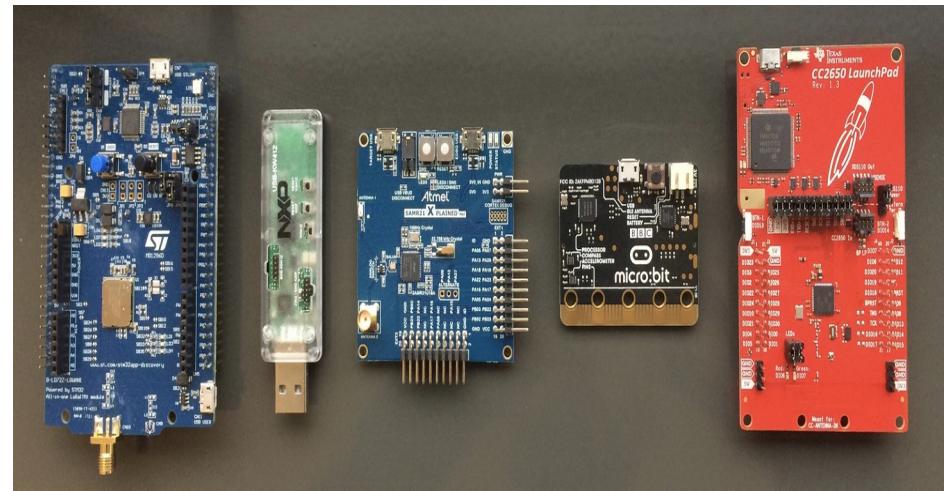
Le Micro-contrôleur (MCU) en Détail



- **Le micro-processeur** est la pièce maîtresse du micro-contrôleur.
- Il se compose d'un **processeur** (CPU) (**unité de contrôle, horloge, registres, unités de traitement**)
- Le micro-processeur interagit avec des mémoires (**ROM pour programme, RAM pour variables**) et des entrées/sorties (I/O) vers les périphériques.
- Ces éléments sont reliés par des bus (**bus d'adresses, bus de données, bus de contrôle**)

Du Micro-processeur au Micro-contrôleur

- **Un processeur seul ne suffit pas ; il a besoin de composants supplémentaires (mémoire, etc.).**
- Pour simplifier **la conception, les fabricants** proposent des briques complètes : le micro-contrôleur (MCU).
- L'ensemble des composants vitaux (**CPU, RAM, ROM, périphériques**) sont intégrés sur une seule puce miniaturisée.



Avantages d'utiliser un MCU pour les Systèmes Embarqués

- **Gain de place** : Unique puce avec broches E/S.
- **Gain de consommation d'énergie** : Chemins plus courts entre composants.
- **Gain de temps** : Certification matérielle simplifiée.

Réduction des coûts de fabrication.

Les fabricants proposent une grande variété de références pour couvrir tous les besoins.



Périphériques et Fonctions embarqués dans le MCU

En plus du CPU et de la mémoire, le MCU embarque des fonctions périphériques utiles pour la programmation.

- **Timers** : Liés aux cycles d'horloge, utilisent des prescalers pour de plus longues périodes.
- **Interruptions** : Gérées par le MCU, liées à des événements internes/externes (bouton, message radio) Permettent de mettre en pause le code principal pour exécuter une fonction spécifique .
- **Watchdog** : Garantit un état cohérent, évite le blocage Compte à rebours à recharger ; sinon, interruption/redémarrage.
- **Convertisseur Analogique/Numérique (CAN/ADC)** : Transforme signal analogique en valeur numérique (précision 8-12 bits)....
- **Entrées/Sorties digitales (GPIO)** : Broches externes configurables en entrée/sortie

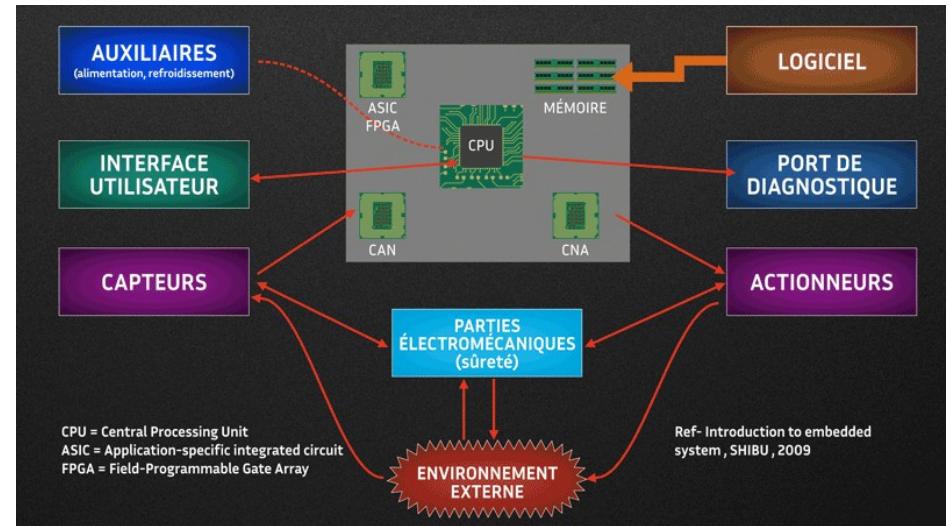


Les Bus de Données Locaux

Leur rôle est de relier les **E/S** du MCU aux périphériques du système embarqué.

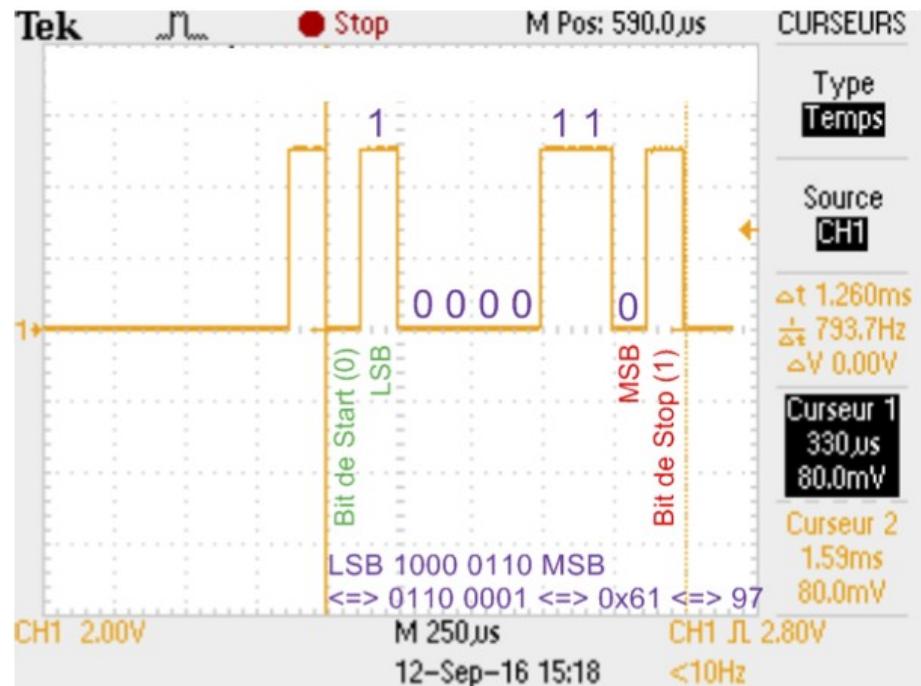
Le type de périphérique impose souvent un bus particulier.

UART, I2C, SPI sont des exemples clés.



Bus de Données : UART

- **UART** (Universal Asynchronous Receiver/Transmitter) est un bus série .
- **Sa vitesse est définie en bps** (bits par seconde)
- **Vitesse paramétrable** (ex: 4800 à 921600 bps)
- Les **deux UARTs** en communication doivent avoir la même configuration
- Les données sont transmises sous forme de trames (**Start bit, données, Stop bit**) .



Bus de Données : I2C

Bus série utilisant trois fils (SDA, SCL, GND).

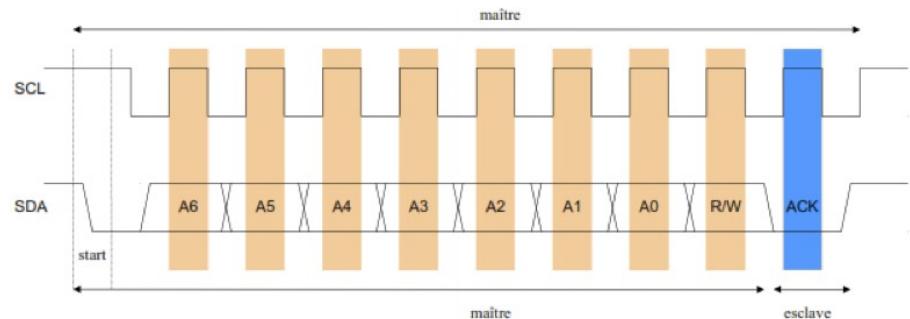
- **SDA** : Ligne de données bidirectionnelle .
- **SCL** : Ligne d'horloge (cadence les messages).
- **GND** : Masse (référence).

Modèle Maître-Esclave, le Maître initie les messages .Peut être multi-maîtres.Chaque abonné a une adresse sur 7 bits (max 128 adresses)

Chaque octet transféré est acquitté.

Déroulement : Condition Start, envoi adresse destinataire+mode (lecture/écriture), Acquittement par Esclave, Transfert de données (lecture/écriture) + Acquittement, Condition Stop

Cas d'usages : Interroger un capteur, relier deux MCUs, lire/écrire mémoire Flash .



Bus de Données : SPI

Serial Peripheral Interface, conçu par Motorola. Bus série pour transmission synchrone de données. Modèle Maître et un ou plusieurs Esclaves (multipoints) . Communication en full duplex .

Utilise quatre lignes unidirectionnelles ..

- **MOSI** : Maître -> Esclave (données) .
- **MISO** : Esclave -> Maître (données) .
- **SCK** : Horloge (générée par le Maître, synchronise la transmission).
- **SS** : Sélection Esclave (une ligne par Esclave).

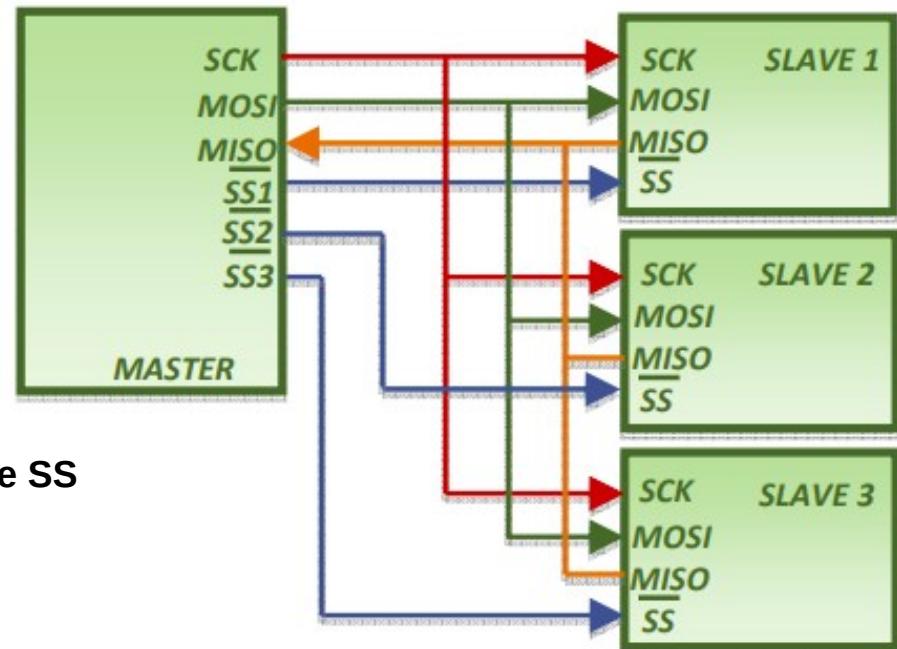
Le **Maître active** l'horloge et sélectionne l'**Esclave via sa ligne SS** dédiée .

Pas de mécanisme d'acquittement standardisé.

Plus flexible que **I2C** sur le nombre de bits par message .

Débit plus important (jusqu'à 20 Mbits/s).

Le nombre **d'esclaves** est limité par le nombre de broches SS disponibles sur le Maître.



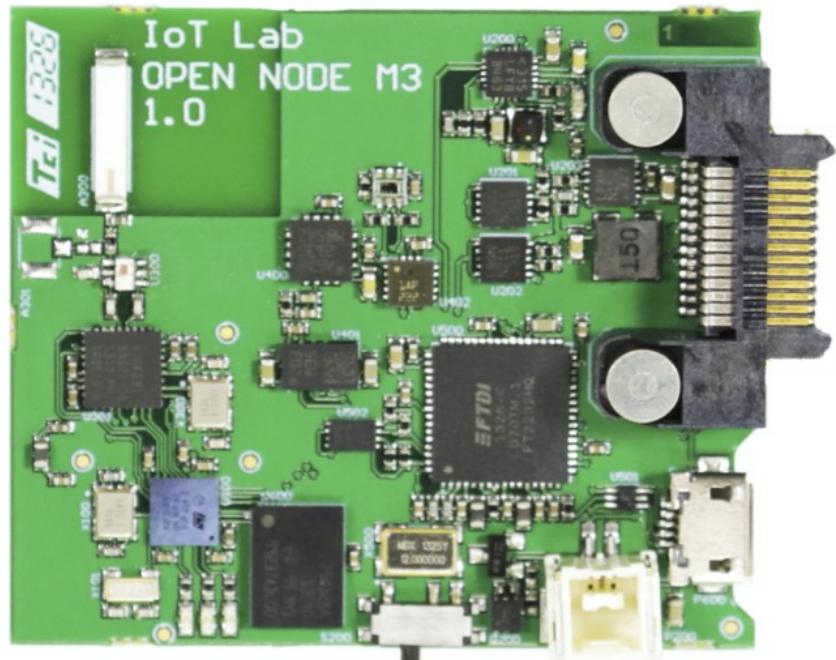
Solutions Logicielles pour Programmer

Exploration de la Chaîne IoT, de l'Objet au Cloud

Spécificités de la Programmation pour Objets Connectés

La programmation doit tenir compte des contraintes fortes de ces objets.

- Contraintes de **mémoire**
- Contraintes de **performance**
- Contraintes de **sécurité**
- Contraintes de **consommation d'énergie**



Modèles de Programmation et Langages

Gestion Mémoire : Allocation par couche ou passage en référence .

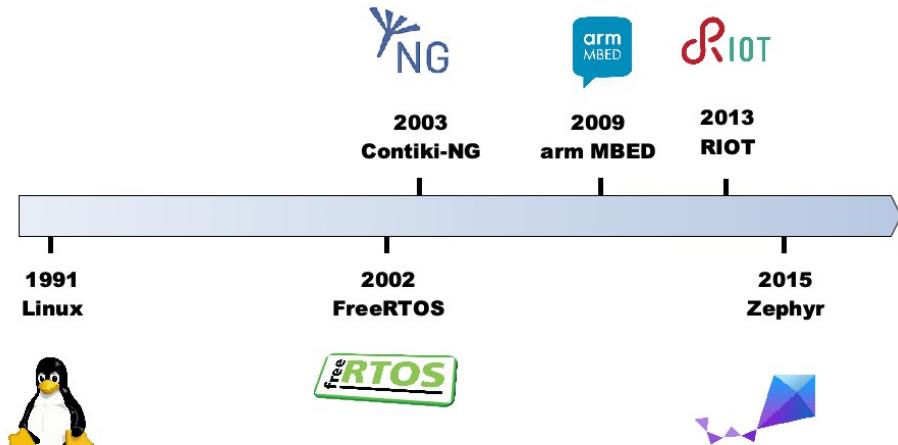
Modèle de Programmation OS :

- **Multithread** : Chaque tâche a son propre contexte .
- **Orienté événement** : Chaque tâche est déclenchée par un événement (ex: interruption).

Langages :

- **Propriétaires** : Adaptés, sûrs, performants.
- **Standards (C, C++)** : Portabilité, outils de débogage standards.

Il existe de nombreux langages et OS



Optimisation de la Consommation : la Puce Radio

La puce radio est généralement le composant le plus énergivore sur un objet connecté (sauf si équipé d'écran)

L'émission radio (**TX**) consomme sensiblement plus que la réception (**RX**) .

Optimisations :

- **Limiter** les messages d'émissions au niveau de l'application.
- **Optimiser** la réception en mettant périodiquement la puce radio en veille.

Ces optimisations sont souvent gérées au niveau de la couche MAC (liaison).



Vers des Systèmes Énergétiquement Autonomes

- Pour atteindre une autonomie énergétique totale, la source d'alimentation doit produire plus d'énergie que l'objet n'en consomme.
- **Solutions courantes** : Modules de recharge + batterie (LiPo) + sources d'énergie renouvelable (panneau solaire, éolienne).
- La taille de la source renouvelable doit être suffisante pour la production souhaité



Protocoles de Communication IoT

Exploration de la Chaîne IoT, de l'Objet au Cloud

Les Réseaux Basse Consommation (WPAN / LRWPAN)

- Le Module 4 introduit **les réseaux sans fil à l'échelle personnelle** (WPAN) et faible débit (LRWPAN).

Sujets abordés :

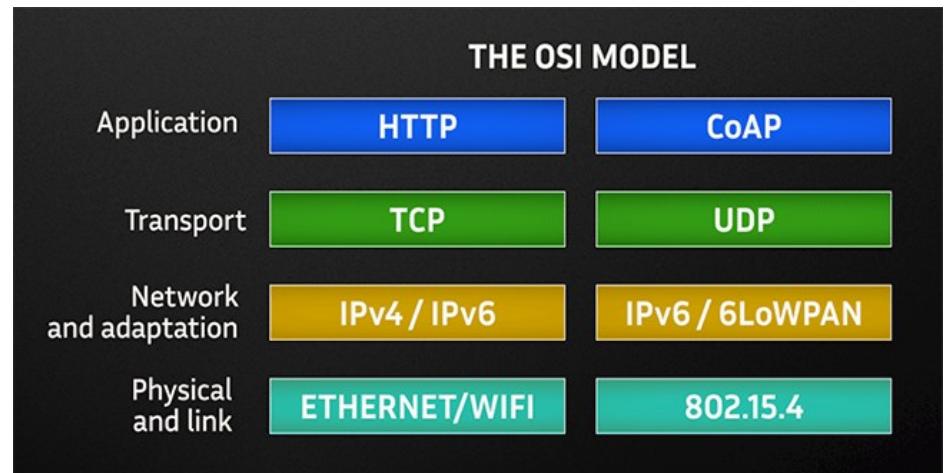
- **Topologies** (schémas d'interconnexion).
- **Contraintes** (énergie, perte de connexion) et caractéristiques attendues (portée, débit, latence, fiabilité) .
- **Le modèle OSI** (référence pour l'interconnexion).



Le Protocole IEEE 802.15.4

Standard de communication clé pour les LRWPAN .

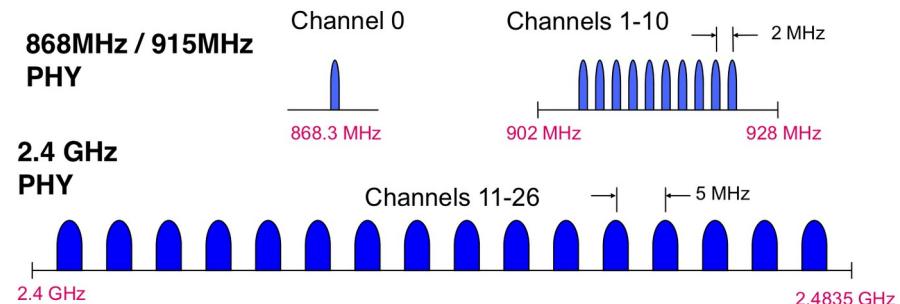
- **Concepts** : modulation, gestion des collisions (CSMA/CA).
- **CSMA/CA** : Écouter avant de parler, mais ne garantit pas l'accès au médium dans un temps donné .
- **Synchronisation pour économiser l'énergie** : Utilisation de balises (Beacons) envoyées par un coordinateur....
- Le coordinateur organise l'accès et le transfert via une Superframe.
- **Superframe** : Période Active (Contention Access Period - CAP, Contention Free Period - CFP) et Période Inactive. Le premier Slot de la période Active est dédié au Beacon .



Accès Garanti et Saut de Fréquence (802.15.4e)

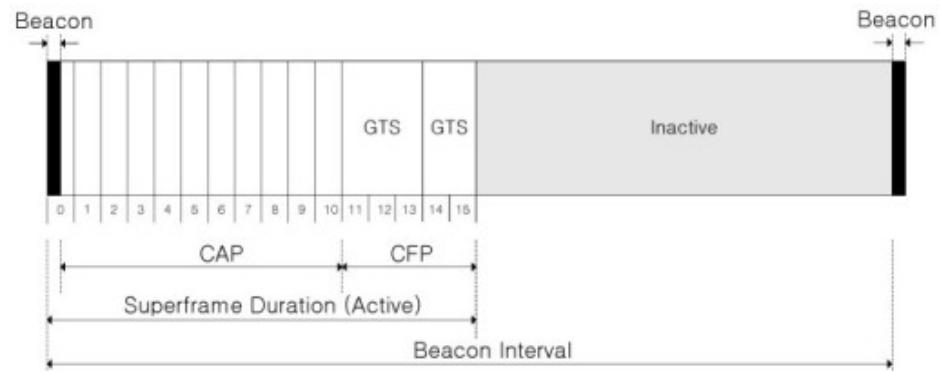
L'amendement IEEE 802.15.4e (2012) a ajouté des mécanismes pour mieux supporter les marchés industriels.

- **Créneaux de temps garantis (GTS)** : Un objet peut demander et obtenir un créneau pendant lequel il est seul à communiquer.
- C'est une forme de TDMA (Time-Division Multiple Access).
- **Saut de fréquence (Channel Hopping)** : Utiliser différentes fréquences pour plus de robustesse face aux obstacles/interférences.
- Ces techniques nécessitent d'allouer un bloc temps/fréquence à chaque lien.
- Des stratégies comme MSF (minimal scheduling function) permettent cela.
- La fréquence utilisée peut changer à chaque slotframe en fonction du nombre de créneaux écoulés (ASN) et d'un décalage de canal (channelOffset).



Rejoindre un Réseau 802.15.4e

- Les communications initiales pour rejoindre le réseau se font dans des cellules partagées sur une fréquence choisie aléatoirement par le coordinateur.
- **Le coordinateur transmet des paquets de contrôle et des balises améliorées** (enhanced beacons, EB) contenant les informations sur la slotframe.
- **Les objets souhaitant rejoindre le réseau écoutent sur les fréquences disponibles** jusqu'à recevoir une balise, qui leur fournit les informations pour la synchronisation et l'insertion.



6LoWPAN et IPv6 pour l'IoT

Le LoWPAN et l'utilisation d'IPv6 .

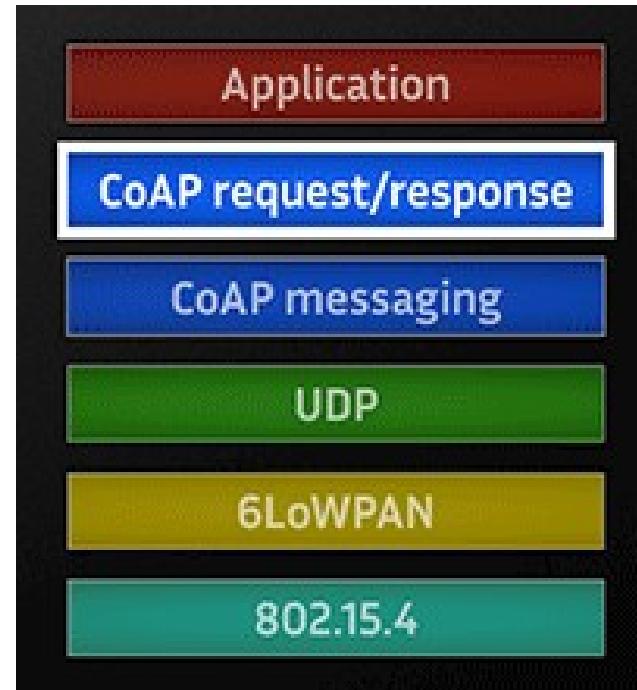
- **6LoWPAN** (IPv6 over Low-Power Wireless Personal Area Networks) est un mécanisme qui permet d'adapter le protocole **IPv6** aux caractéristiques des réseaux 802.15.4, notamment leur faible taille de paquet.
- Il supporte l'autoconfiguration sans état d'adresses **IPv6**.
- Les objets peuvent configurer leur adresse **IPv6** en fonction des messages reçus des routeurs.

CoAP (Constrained Application Protocol)

CoAP est un protocole d'application Internet conçu pour les environnements contraints de l'IoT.

- Il s'appuie sur un modèle client-serveur, similaire à **HTTP**.
- Les clients envoient des requêtes sur des ressources identifiées par des URIs.
- Contrairement à **HTTP** qui utilise **TCP**, **CoAP** fonctionne de manière asynchrone en s'appuyant sur des datagrammes **UDP**.

La fonctionnalité d'Observation (Observe option) permet aux clients de s'abonner à des ressources et de recevoir des notifications lors de changements, optimisant la bande passante et économisant l'énergie .

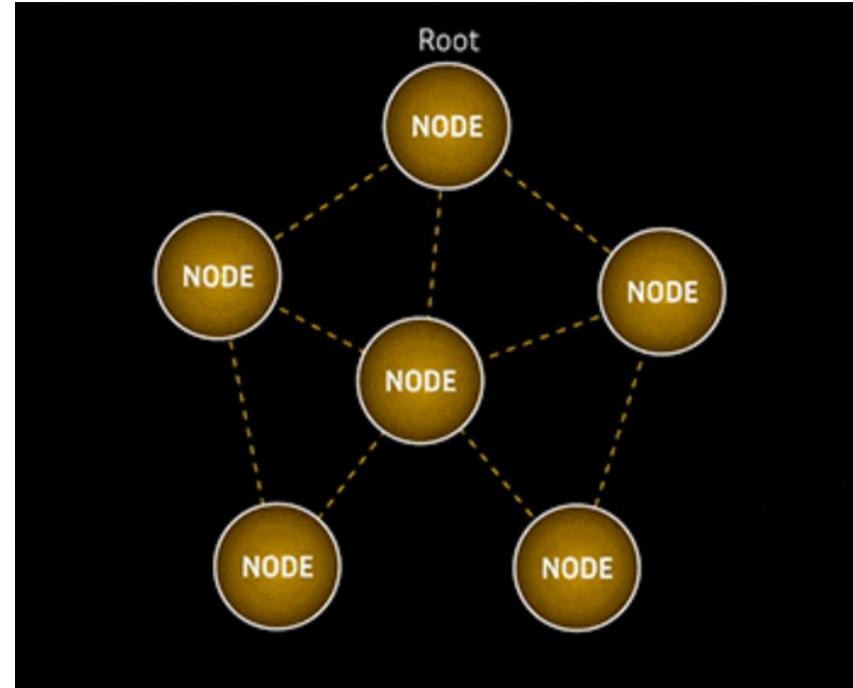


Routage pour les Réseaux Basse Consommation (RPL)

RPL (Routing Protocol for Low-Power and Lossy Networks) est le mécanisme standard de routage pour les réseaux d'objets connectés utilisant IPv6 et fonctionnant en faible puissance

Il construit une topologie orientée acyclique (**DODAG - Destination Oriented Directed Acyclic Graph**).

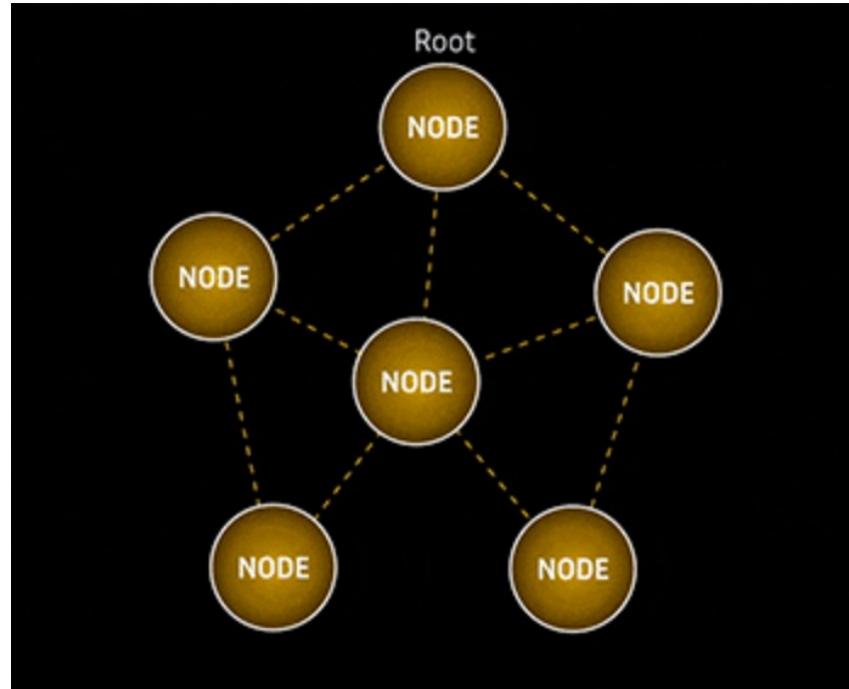
Un déploiement peut contenir plusieurs **instances**, chacune avec une fonction d'objectif définissant comment les routes sont sélectionnées et optimisées. Un nœud peut participer à plusieurs instances .



Messages de Contrôle RPL

RPL utilise des messages spécifiques pour construire et maintenir la topologie :

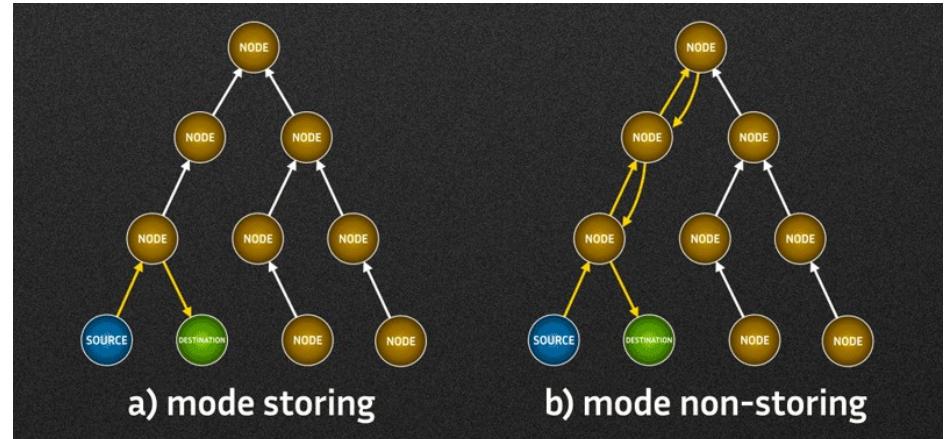
- **DIO (DODAG Information Object)** : Annonces de routage pour construire le DODAG. Leur propagation est gérée par l'algorithme Trickle pour économiser l'énergie .
- **DIS (DODAG Information Sollicitation)** : Demandes d'information par les nœuds pour rejoindre le réseau ou obtenir des infos récentes.
- **DAO (Destination Advertisement Object)** : Permettent le trafic descendant en signalant au parent préféré qu'ils peuvent être contactés⁷⁵ . Peuvent être désactivés si le trafic descendant n'est pas nécessaire.



Gestion de la Mémoire avec RPL : Modes Storing vs Non-Storing

Pour gérer le trafic descendant (P2M, P2P), RPL stocke des informations de routage basées sur les DAOs

- **Mode storing** : Les infos sont stockées de manière distribuée dans le graphe. Chaque nœud a une table de routage pour son sous-arbre. Optimise les flux P2P (messages ne remontent que jusqu'à l'ancêtre commun). Nécessite une capacité de stockage suffisante sur les objets.
- **Mode non-storing** : Les infos remontent jusqu'à la racine du DODAG. Les tables de routage sont plus petites.



Réseaux LPWAN (Low Power Wide Area Network de Communication IoT)

Exploration de la Chaîne IoT, de l'Objet au Cloud

LPWAN : Concepts Clés

Les **LPWAN** sont des réseaux sans fil basse consommation conçus pour de longues portées .

Utilisent des bandes de fréquences non licenciées.

Concept de **Duty Cycle** : Limite le temps d'émission des dispositifs pour éviter de saturer la bande .



LPWAN : Exemple Sigfox

- **Sigfox** est un pionnier dans cette technologie.
- Définit une modulation à 100 bits/s et des formats de messages simples adaptés aux processeurs embarqués

Exemple : Bornes à incendie Bayard utilisant le module Copernic (détection utilisation, fuites, chocs)

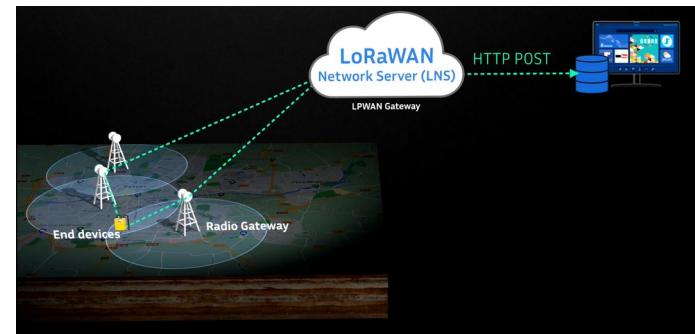
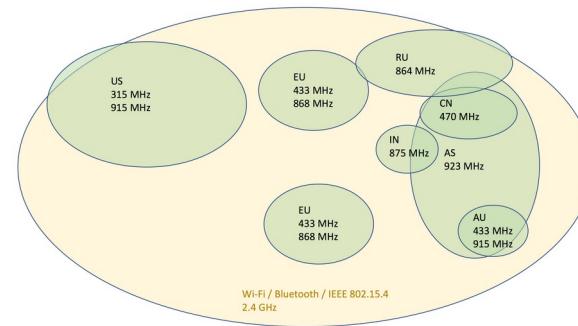


Le Protocole LoRaWAN

LoRaWAN est le protocole réseau (couches supérieures) qui s'appuie sur la modulation **LoRa**.

Il fédère les acteurs et assure l'interopérabilité (contrairement à l'utilisation de la modulation **LoRa** seule avec des formats propriétaires).

- Dans les bandes non licenciées, le moment d'émission des données est crucial.
- **Le duty-cycle** limite les indications du réseau.
- **LoRaWAN** se base sur les propriétés du hasard pour les émissions : la méthode Aloha.
- **Aloha** (transmission aléatoire) a une limite théorique de transmission de 18% de la capacité maximale Au-delà, le réseau devient instable .

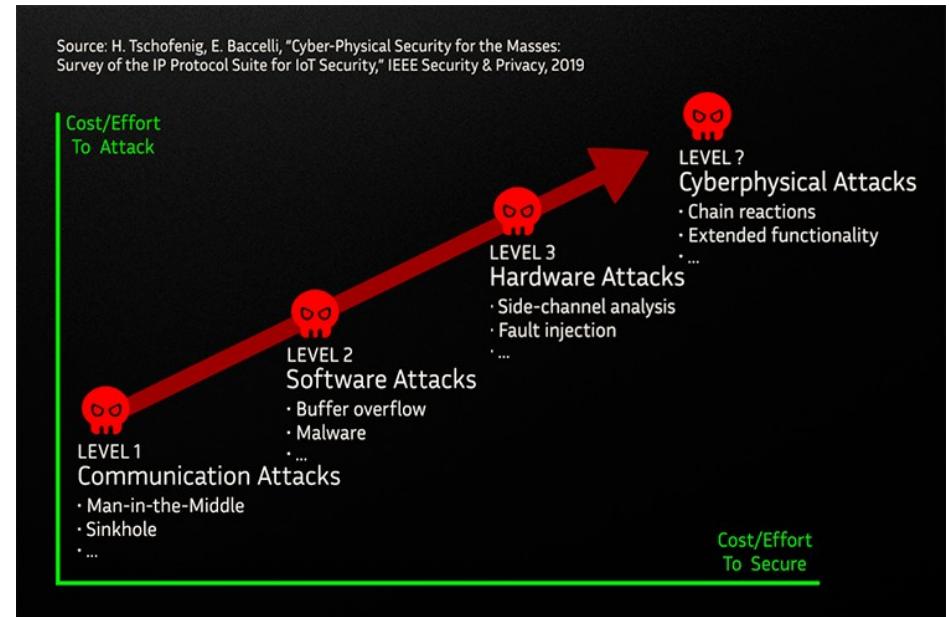


Sécurité de l'IoT

Exploration de la Chaîne IoT, de l'Objet au Cloud

Introduction à la Sécurité de l'IoT : Défis et Attaques

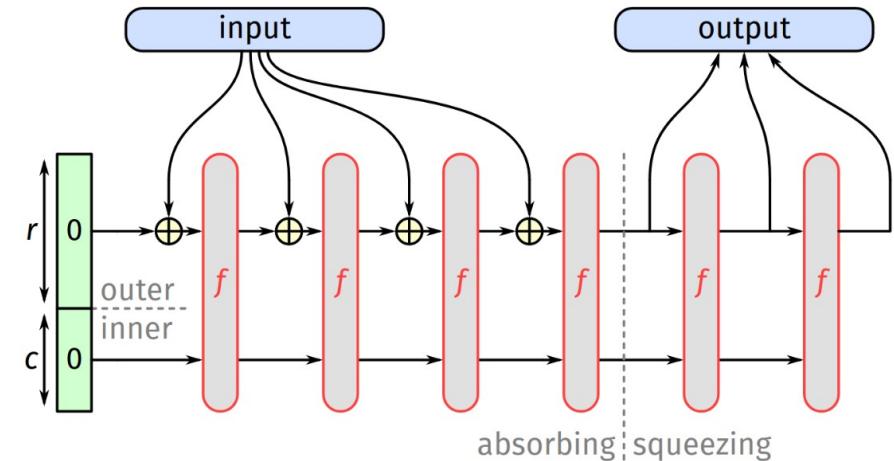
- Les **objets connectés** présentent des défis de sécurité spécifiques par rapport à Internet .
- Identifier les différents **types d'attaques possibles** contre les objets connectés .
- Comprendre **les défis et les contraintes** pour sécuriser une application IoT .



Cryptographie pour les Objets Connectés

Le Module 6 couvre les bases de la cryptographie appliquée à l'IoT

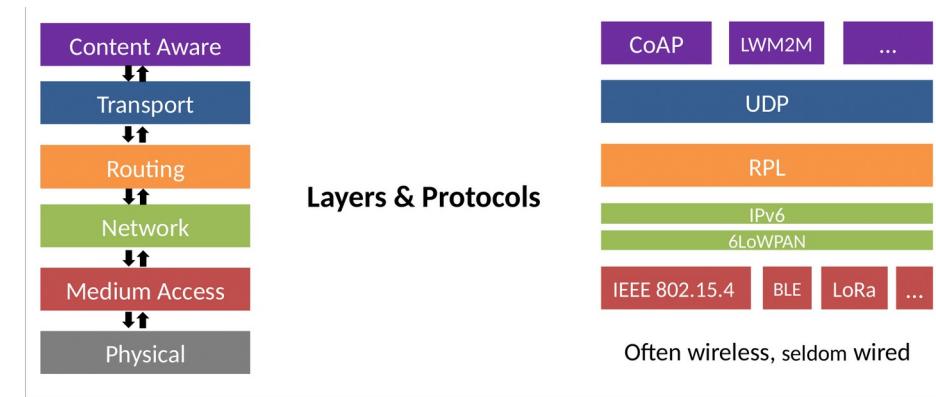
- **Exemple:** Fonction de hachage Keccak .
- Keccak utilise un processus basé sur une éponge ("sponge function") avec **deux phases** : absorption et essorage .
- Keccak est une fonction de sortie extensible (XOF), capable de générer une sortie de longueur arbitraire⁸⁷ .



Sécurité des Réseaux pour les Objets Connectés

Comment sécuriser les communications dans les réseaux IoT .

- La pile réseau est généralement modulaire, avec **des protocoles à différentes couches** (physique, accès au support, réseau, routage, transport, sensible au contenu).
- Une figure illustre les catégories de protocoles et une pile réseau IoT basse consommation (**IEEE 802.15.4, 6LoWPAN, IPv6, RPL, UDP, CoAP, LwM2M, etc.**).



Sécurité des Communications : OSCORE

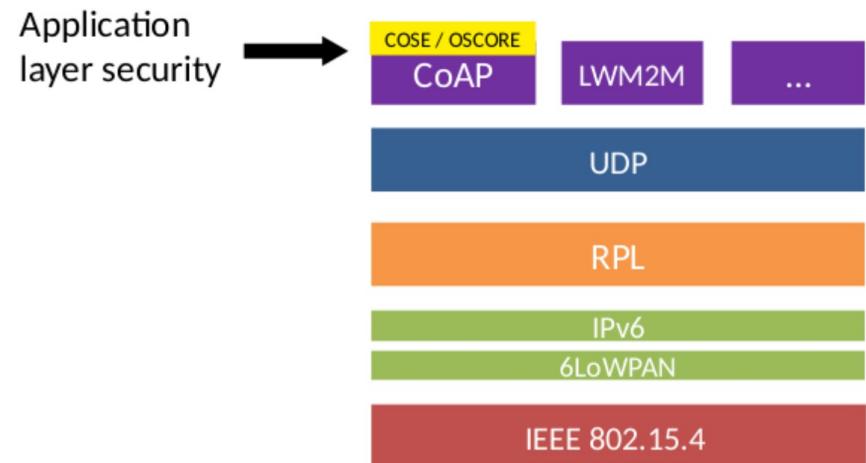
OSCORE est un protocole de sécurité spécifiquement conçu pour sécuriser les communications basées sur CoAP.

Un paquet **OSCORE** est un paquet CoAP avec une Option OSCORE et une suite de chiffrement.

L'Option **OSCORE** contient des informations comme un numéro de séquence et un ID permettant au récepteur de déterminer le contexte de sécurité (la clé pré-partagée) pour déchiffrer.

Le chiffrement **COSE** (CBOR Object Signing and Encryption) est utilisé pour chiffrer le texte clair OSCORE (charge utile CoAP, options, code méthode).

Les options **CoAP** chiffrées (options internes) restent opaques aux proxys intermédiaires, car elles sont destinées au point d'extrémité OSCORE.



Mise à Jour Sécurisée du Logiciel (SUIT)

(Les extraits fournis ne contiennent pas de détails spécifiques sur ce sujet, mais le sommaire mentionne "Mise à jour sécurisée d'un logiciel d'objets connectés (SUIT)".

