IoT et sécurité

❶⇒Qu'est-ce que «l'IoT» ;

❷⇒Les menaces sur l'IoT ;

❸⇒Des vulnérabilités spécifiques ;

❹⇒Les mécanismes de sécurité ;

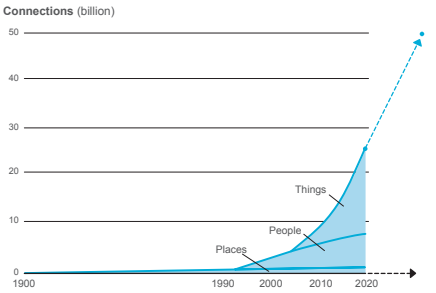
❺⇒Analyse de risques : approche réseau ;

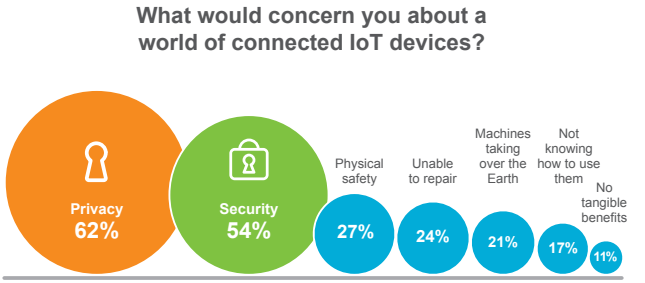
❻⇒Approche matérielle : le «root of trust» ;

❼⇒Chiffrement et IoT.

1 «*The Internet of Things*»

Jusqu'à 26 milliards d'objets connectés en 2020





Des éco-systèmes très différents

- un **large spectre** de «*devices*» :
  - capteurs fortements contraints construit avec de l'électronique imprimable ;
  - véhicules autonomes comme des camions, voitures et avions ;
- des **scénarios d'usage** étendus :
  - surveillance de la température ;
  - surveillance de processus industriels critiques ;
- des **attentes de sécurité** différentes :
  - public : sécurité des données personnelles et «*privacy*» ;
  - entreprise : sécurité des secrets industriels et des infrastructures critiques ;
- **mais des bénéfices considérables** :
  - **analyse de données**, «*big data*» ;
  - **automatisation** ;
  - **optimisation** des ressources et des traitements ;

«*The Internet of Things*»

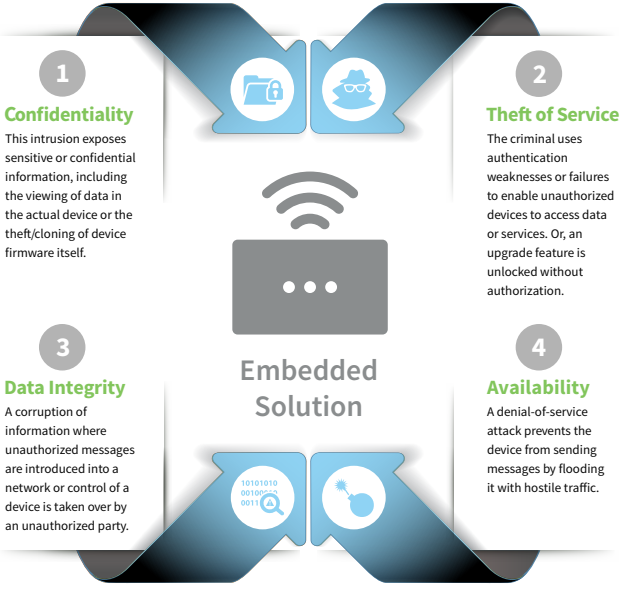
Une définition et une infrastructure

Un réseau d'objets connectés à Internet capables de collecter et d'échanger des données.

- des **objets** qui nous entourent au quotidien embarquent :
  - des capteurs plus ou moins sophistiqués ;
  - des SoC, «*System-On-Chip*» ;
- les **données capturées** d'un objet sont :
  - envoyées à une passerelle qui les envoie ensuite sur Internet ;
  - stockées dans le «*cloud*» pour y être analysées ;
- après **analyse**, les données traitées sont transmises à une application IoT qui :
  - exploite ces données suivant différents besoins ;
  - est bâtie sur une plateforme offrant un langage commun capable de faire communiquer capteurs et actionneurs embarqués dans les objets
- il existe **différentes plateformes** :
  - certaines basées sur le «*cloud*» pour intégrer les données de nombreux objets ;
  - d'autres supportant le développement d'applications IoT ;

2 Les menaces sur l'IoT

- **ressources disponibles limitées** : faible capacité en batterie, mémoire et vitesse de traitement
  $\Longrightarrow$ ne peut pas supporter les mesures de sécurité habituelles ;
- **manque d'intérêt pour les données** : les données de l'IoT ne sont pas forcément vues comme importante, ce n'est pas la motivation première des attaques : c'est le défi qu'ils représentent ;
- **disponibilité des outils** : tous les outils pour modifier/analyser/étudier les IoTs sont disponibles pour tous ;
- **pas besoin d'un accès physique** : utilisation de communication sans fil ;
- **interface différente et limitée** : les rapports d'erreurs et de sécurité peuvent être facilement ignorés ;
- des **ports d'accès physiques** : utilisés pour la programmation ou le déboguage.



Des attaques médiatisées : «Mirai Botnet» constitué de caméras IP et de router personnels ;

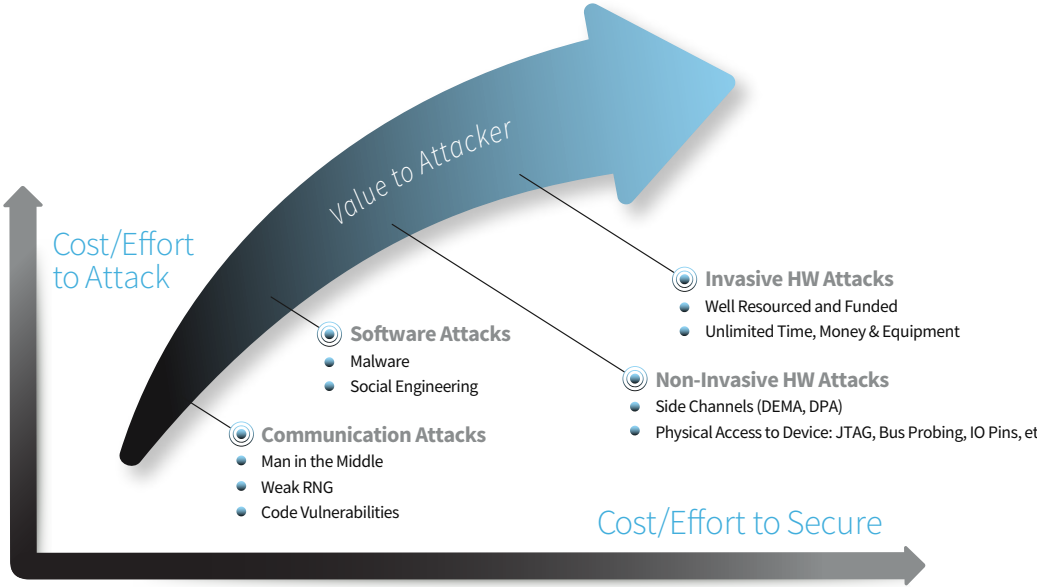
«DDoS» : détournement d'IoT pour

- ▷ créer un botnet pour attaquer une cible à l'intérieur ou à l'extérieur du réseau IoT ;
- ▷ épuiser la batterie de l'IoT victime.

Les menaces sur l'IoT

La sécurité est un équilibre entre le coût et le bénéfice

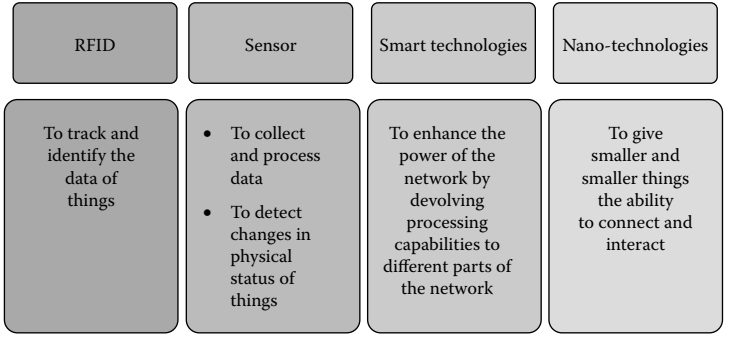
- ☐ un attaquant disposant de **suffisamment** de **temps**, **d'argent** et **d'expertise** peut hacker n'importe quel système ;
- ☐ le but de la sécurité est de rendre le **coût** de l'attaque **trop important** par rapport au gain espéré ;



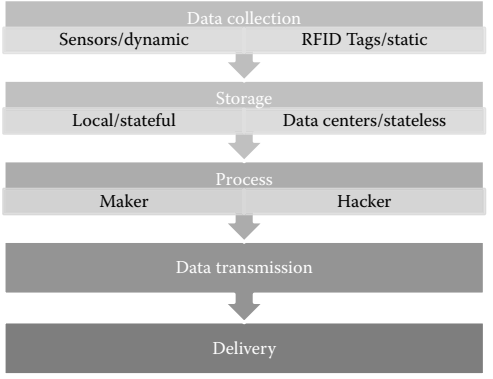
- ▷ **attaques matérielles invasives** : «*reverse engineering*», utilisation de micro-sonde sur le processeur ;
- ▷ **attaques logicielles passives** : exploiter une vulnérabilité dans le code du firmware de l'objet ;
- ▷ **attaques sur les communications** : exploiter des vulnérabilités dans les protocoles réseaux, dans la cryptographie utilisée ou dans les clés de chiffrement utilisées ;

3 IoT : un catalogue de vulnérabilités

Technologies



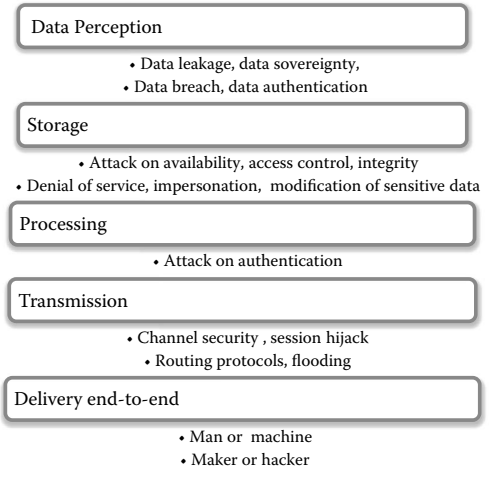
Les différentes phases présentes dans un éco-système IoT



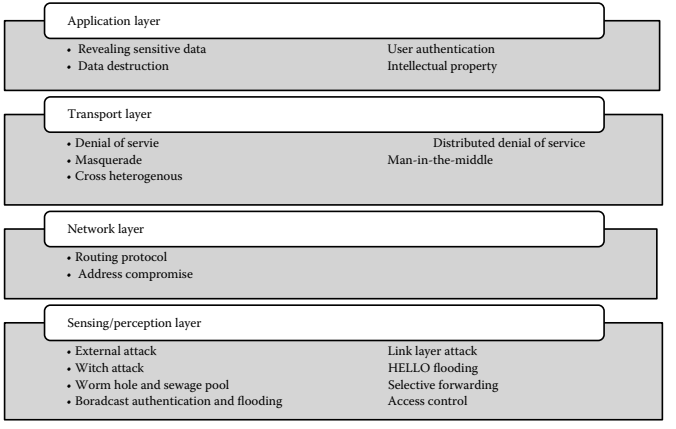
- «*Intelligent Processing*» : les données stockées dans les DCs, «Data Centers», sont analysées.
- «*Data Transmission*» : du capteur, RFID, SoC vers le DC, du DC vers les unités de traitement et des processeurs vers les contrôleurs, devices ou utilisateurs.
- «*Data Delivery*» : remise des résultats à temps et sans altérations.

IoT : un catalogue de vulnérabilités

Attaques sur les différentes phases



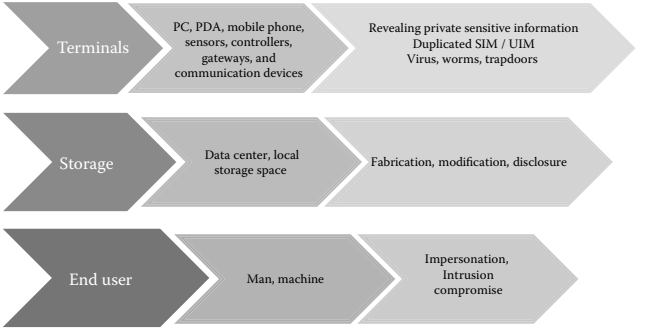
- «*External Attack*» : utilisation du Cloud, confiance dans le tiers fournissant le service ;
- «*Wormhole Attack*» : transmission de données d'un IoT à un autre endroit géographique ;
- «*Selective Forwarding*» : bloquer certains messages mais pas d'autres ;
- «*Witch Attack*» : remplacer un IoT en panne par un malicieux ;
- «*Data sovereignty*» : l'IoT est réparti sur toute la planète, mais les lois des différents pays s'appliquent ;
- «*Attack on Availability*» :
  - Flooding by attackers : attaques sur les DCs
  - Flooding by legitimates (flash crowd) : surcharge du à des accès simultanés d'utilisateurs légitimes ;
  - Flooding by spoofing : attaque par vol d'identité ;
  - Flooding by aggressive legitimates : des utilisateurs légitimes qui font des requêtes très nombreuses, très rapidement ;



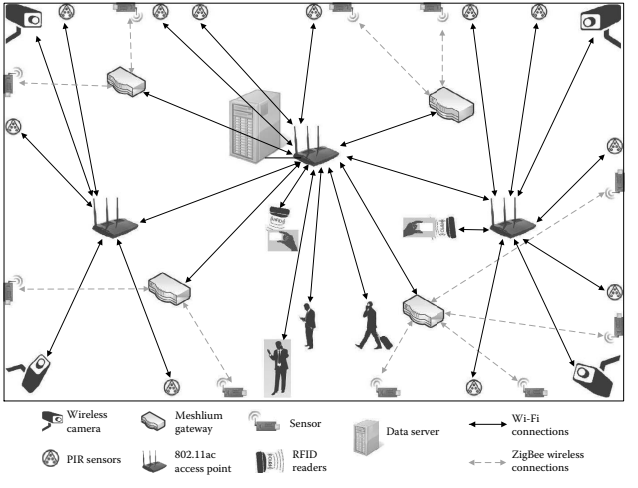
IoT : un catalogue de vulnérabilités

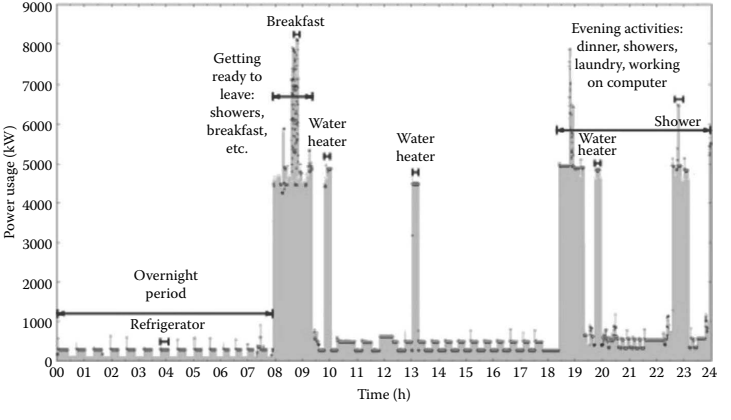
- «*DDoS*», «*Distributed Denial of Service*» : attaque sur les DCs qui empêchent les utilisateurs légitimes d'accèder à ces DCs :
  - «*Network level*» : épuiser les ressources du CSP, «*Cloud Service Provider*», avec des connexions ouvertes à moitié (SYN et SYN/ACK mais pas ACK) ;
  - «*Service level*» : des requêtes malicieuses vers les services ;
- «*IP spoofing attack*» des attaques associées à des DDoS sur les DCs dont l'origine semble être les clients légitimes
  - *Hidding Attack* : envoyer de nombreux paquets avec des adresses d'origine aléatoires ;
  - «*Reflection Attack*» : envoyer des paquets à n'importe quelle destination avec l'adresse d'origine de la victime ;
  - «*Impersonation Attack*» : l'adresse d'origine des paquets correspondent à celles d'utilisateurs légitimes et l'attaquant tente une attaqte «*M-I-T-M*» ;
- «*Eavesdropping*» : interception du trafic pour obtenir des accès non autorisés ;
- «Replay Attack» : rejouer des paquets pour obtenir des accès non autorisés ;
- «*Backdoor*» : utiliser des accès de déboguages ;
- «*Sybil Attack*» : créer/voler différentes identités d'IoT pour influencer le fonctionnement du système ;
- «*Byzantine failure*» : attaquer un ou plusieurs serveurs pour dégrader les performances du Cloud ;

Attaques sur les «*composants*» :



IoT : attaque sur la vie privée



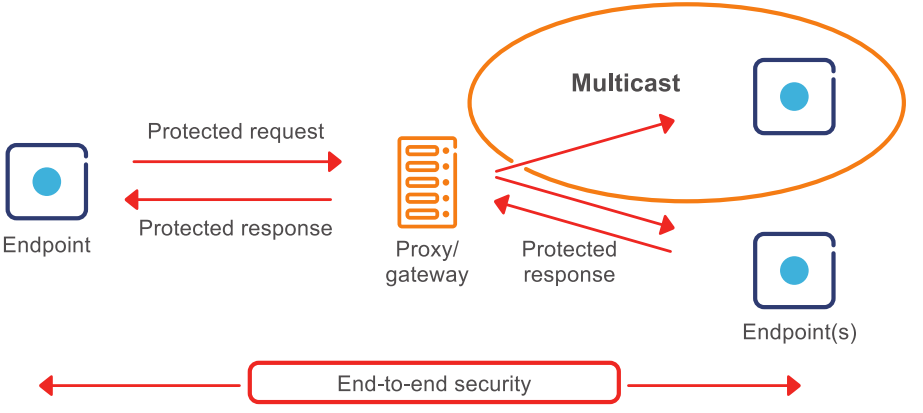


Méthode	**coût**		**Techno**
Packet Encryption	bas	le mécanisme le plus employé	FIPS-197/AES
Replay Protection	bas	empêcher le renvoi de paquets enregistrés	date/AES CTR
		utilisation du chiffrement par chaînage en mode «counter».	
Message Authentication Code	bas	empêcher la modification des messages	HMAC
		Intégrité et signature des données qui peuvent rester en clair	
Port Protection	bas	empêcher l'accès physique à un port de debogage série ou JTAG	
		Utiliser des mots de passe d'accès définis en usine	
Secure Bootloader	moyen	garantir que seul des firmwares autorisés s'exécutent	
		HMAC du firmware, vérification des mises à jour, utilisation d'un TEE	
Pre-Shared Keys	bas	le mécanisme le plus accessible	
		Clés installées par un canal sécurisé dans l'objet	
SSH	haut	utilisable uniquement avec des systèmes basés Unix	
Public Key Exchange	haut	besoin d'un OS sur l'IoT	ECC
		Gestion de clés et de certificats	
TLS	haute	besoin d'un OS sur l'IoT	Max Fragment Length, RFC6066
		Gestion de clés, de certificats et de buffers TCP suffisants	
WPA2	haute	besoin d'un OS sur l'IoT	WiFi
		Pile TCP/IP évoluée pour disposer de l'authentification serveur	

5 «*The Internet of Things*» : analyse des risques, approche réseau

- Pour être utile : communication **temps réel** et **bi-directionnelle** vers Internet ;
  ⟹ type de communication notoirement difficile à sécuriser

- besoin d'un **nouveau modèle** de sécurité
  ⟹ protocoles et «best practices» pour les serveurs, les ordinateurs personnels et les smartphones sont bien connus mais inapplicables directement
  ⟹ besoin de solution «*plug and play*» : pas de mise en service, de logiciel et de firmware à mettre à jour à effectuer par l'utilisateur
  ⟹ décaler le problème de la sécurisation du constructeur matériel vers la couche réseau : plus de flexibilité et de robustesse ;
  ⟹ assurer une sécurité bout en bout, «*end-to-end*».

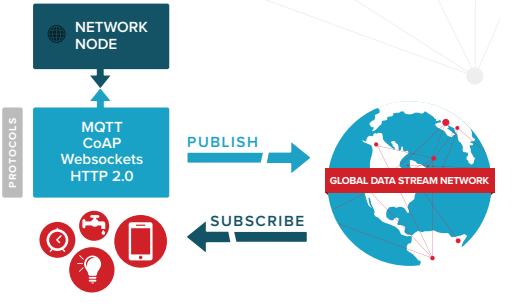


«*The Internet of Things*» : préconisations

- les objets **ne doivent pas disposer** de port ouvert en **entrée** :
  - pour qu'un serveur puisse envoyer des données, «*push*», à un objet, celui-ci doit être en attente : disposer d'un port de connexion sur lequel le serveur peut se connecter.

    $\Longrightarrow$ risque massif de sécurité :
    - ▷ installation de malware ;
    - ▷ modification et/ou vol de données ;
    - ▷ attaque par «DoS» ;
    - ▷ exécution de code ;

  $\Longrightarrow$ l'objet connecté doit effectuer **uniquement des connexions sortantes** :
  - ⋆ utiliser le modèle publier/s'abonner, «*publish/subscribe*», pour disposer de lien de communication bi-directionnel ;
  - ⋆ supporter le «*scaling*» du modèle : jusqu'à 50 milliards d'objets connectés
    - ▷ serveurs hautes performances ;
    - ▷ nombreux points de présence répartis sur la planète ;



Les protocoles adaptés : MQTT, CoAP, Websockets et HTTP 2.0.

«*The Internet of Things*» : préconisations

- **Chiffrement de «bout en bout»**, «*end to end*» :
  - utilisation de **TLS**, «*Transport Layer Security*» :
    - chiffrement des communications de l'objet au serveur ;
    - authentification du serveur et aussi de l'objet ;
  - utilisation **d'AES** pour le chiffrement des données produites par l'objet connectés ;

- utilisation du **modèle d'enveloppe** :
  - les données de l'objet sont chiffrées par AES pour le destinataire final avec une clè secrète partagée ;
  - ces données chiffrées sont intégrées dans une enveloppe contenant des **données en clair** pouvant être utilisées par les intermédiaires :
    - filtrage/routage ;
    - analyse

L'intégralité de l'enveloppe est échangée de manière chiffrée au travers de TLS.

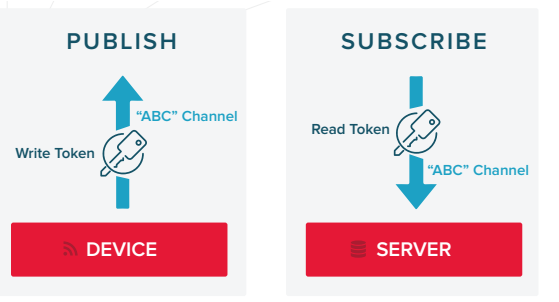
«*The Internet of Things*» : préconisations

- **Contrôle d'accès sécurisé par jeton** :
  - contrôle d'accès sur «qui» peut transmettre «quoi» en émission comme en réception
    - millions d'objets connectés essayant d'écouter les bon canaux et les bons topics

    $\Longrightarrow$ inefficace et non sécurisé de laisser ces objets filtrer les topics auxquels ils n'ont pas souscrit ;

    $\Longrightarrow$ c'est au «réseau» de le faire ;

  - intégration d'un système de jeton dans le modèle publier/souscrire :
    - distribuer un jeton à un élément pour donner un droit d'accès à un canal de données
    - contrôle d'accès fin :
      - quel jeton est crée,
      - quel «device» le reçoit
      - pour quelles données ce jeton autorise l'accès



C'est le réseau d'interconnexion qui contrôle le trafic :

- *autorisation d'accès ;*
- *choisir quels appareils peuvent parler ou écouter sur le réseau*

suivant les jetons que le réseau distribue.

Groupe de travail IETF : ACE, «*Authentication and Authorization for Constrained Environments*»

16 novembre 2017 : `https://tools.ietf.org/html/draft-ietf-ace-oauth-authz-09`
basé sur OAuth v2.0 et CoAP.

«*The Internet of Things*» : préconisations

- **Surveillance du status d'un objet** :
  - surveiller constamment le status «online/offline», la présence, d'un appareil :
    $\Longrightarrow$ **Alerter l'utilisateur** ou le système de surveillance si un appareil **arrête d'émettre ou de recevoir** des données
    Que ce soit dans l'IoT des particuliers ou bien industriel : capteur sur une plateforme pétrolière, système de surveillance de domicile, appareil électroménager etc.
    $\Longrightarrow$ un appareil offline peut signifier :
    - une tentative locale de manipulation, «*tampering*» ;
    - une situation de perte de connexion Internet ou une panne de courant ;
  - disposer d'un **canal indépendant et sécurisé** pour échanger des données de présence pour chaque appareil qui peut être personnalisé :
    - statut online/offline ;
    - température ;
    - accélération ;
    - géolocalisation ;

    Par exemple :
    - *la serrure d'une porte connectée peut alerter son propriétaire du changement de statut de la serrure si son téléphone n'est pas à 10m de là ;*
    - *si un ensemble de capteurs d'une usine de génération d'énergie solaire passent offline, le réseau peut dépêcher un technicien pour identifier le problème ;*

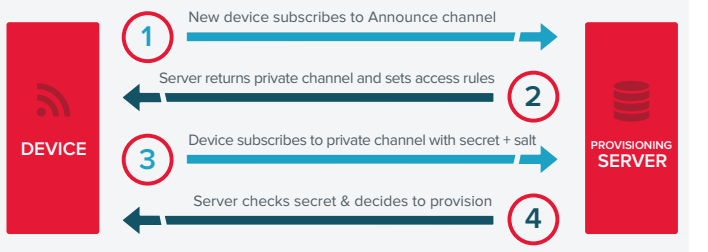
«*The Internet of Things*» : préconisations

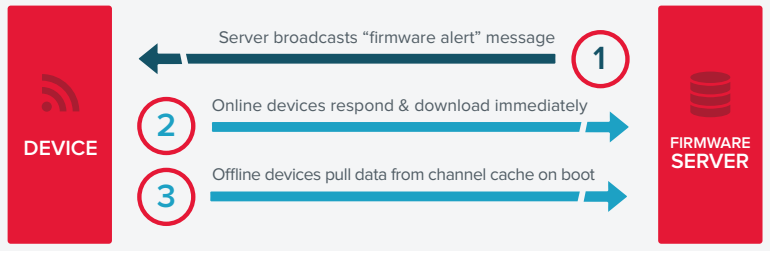
- «User friendly» configuration et mise à jour :
  - différentes étapes dans la vie d'un objet connecté :
    - l'objet est opérationnel et se connecte à Internet ;
    - l'objet doit être configuré et son logiciel doit être maintenu à jour ;

    Exemple : un utilisateur vient d'acheter un système de 6 caméras connectées avec détection de mouvement pour la sécurité de sa maison : il espère que tout va fonctionner...

    mais il doit :
    - *configurer son firewall qui bloque leur connexion ;*
    - *mettre à jour leur firmware : mise à jour des fonctionnalités et correction des failles de sécurité ;*

    et souvent, si cela marche il ne fera plus de mise à jour...
- Utilisation du modèle publier/souscrire avec les ports HTTP en sortie (port 80 et 443) :
  - 1. souscription à un canal d'annonce et s'annonce su le réseau ;
    2. le serveur renvoie un canal privé partagé :
       - ▷ définir les règles d'accès ;
       - ▷ intégrer l'objet ;



  - 

    La mise à jour du firmware peut être faire automatiquement :
    1. le serveur informe les objets au travers d'un canal en diffusion ;
    2. chaque objet effectue sa mise à jour ;
    3. en cas d'un objet offline, il fera la mise à jour au moment où il redeviendra online ;

6 Protection matérielle : «*roots of trust*»

- ☐ **protection automatique** de leur fonctionnement et des données contenus ;

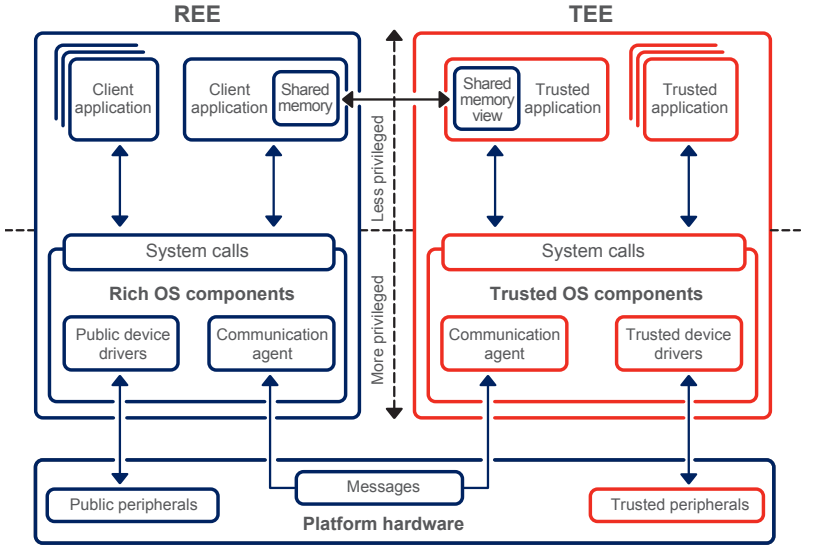
⟹ *les données* sensibles *conservées dans un stockage non-sécurisé doivent être chiffrées et leur intégrité protégée pour fournir une fonction de* ***stockage sécurisé*** *;*

- ☐ **vérification basée sur la cryptographie** du logiciel embarqué et des mises à jour ;
- ☐ possibilité de **mise à jour à distance**, «OTA», «*Over-The-Air*», du firmware même en cas de malware ;
- ☐ **espace mémoire suffisant** pour permettre d'aller vers une version antérieure de firmware en cas de faille critique mais de manière sécurisée : empêcher une attaque par «*rollback*».

⟹ *ces propriétés de sécurité doivent être* ***isolées*** *des applications présentes sur l'objet.*

- ☐ **chemin sécurisé** : les données à protéger ne doivent pas circuler par des canaux non sécurisés

⟹ *adaptation du DMA pour gérer des canaux sécurisés, plusieurs bus de données, MMU avec tables distinctes, etc.*



Isolation basée hardware

- ▷ **Trusteed Execution Environment** :
  - ◇ **réalise** toutes les opérations de sécurité évoquées plus haut ;
  - ◇ **protège** les applications contre :
    - ⋆ les **autres applications** ;
    - ⋆ un **système d'exploitation compromis** ;

  ⟹ Pour un «bootstrapping» automatique et sécurisé : **installation des «*credentials*»** lors de leur fabrication **en usine**.

- ▷ **Rich Execution Environment** : l'environnement normal d'exécution.

7 Chiffrement et IoT

- Cryptographie **asymétrique** :
  - utilisable sur la plupart des processeurs embarqués ;
  - difficulté sur des processeurs «*ultra low-cost*» ;

  $\Longrightarrow$ remplacement par de la cryptographie «post-quantum» : taille beaucoup plus grande des clés et signatures ;

- Cryptographie **symétrique** :
  - le coût énergétique est négligeable par rapport à celui des communications sans fil ;

- Cryptographie «***Lightweight***» :
  - réservée aux environnements fortement contraints : étiquette RFID, capteurs, carte sans-contact, *etc.*
  - combinaison de «*Block Ciphers*», «*Stream Ciphers*» et «*Hash Functions*»/MAC demandant de faible ressources en lors de l'exécution (taille RAM et puissance CPU) et en espace mémoire (taille faible des clés et des données) ;

Symétrique	**ECC**	**DH/DSA/RSA**	**recommandé**
80	163	1024	
112	233	2048	RFC 7525
128	283	3072	ENISA 2013
192	409	7680	
256	571	15360	

  - Chiffrement **asymétrique** :
    - implémentation efficace de courbes elliptiques (Curve25519), mais prends du temps si non accéléré matériellement ;
    - obligatoire pour l'utilisation de TLS.