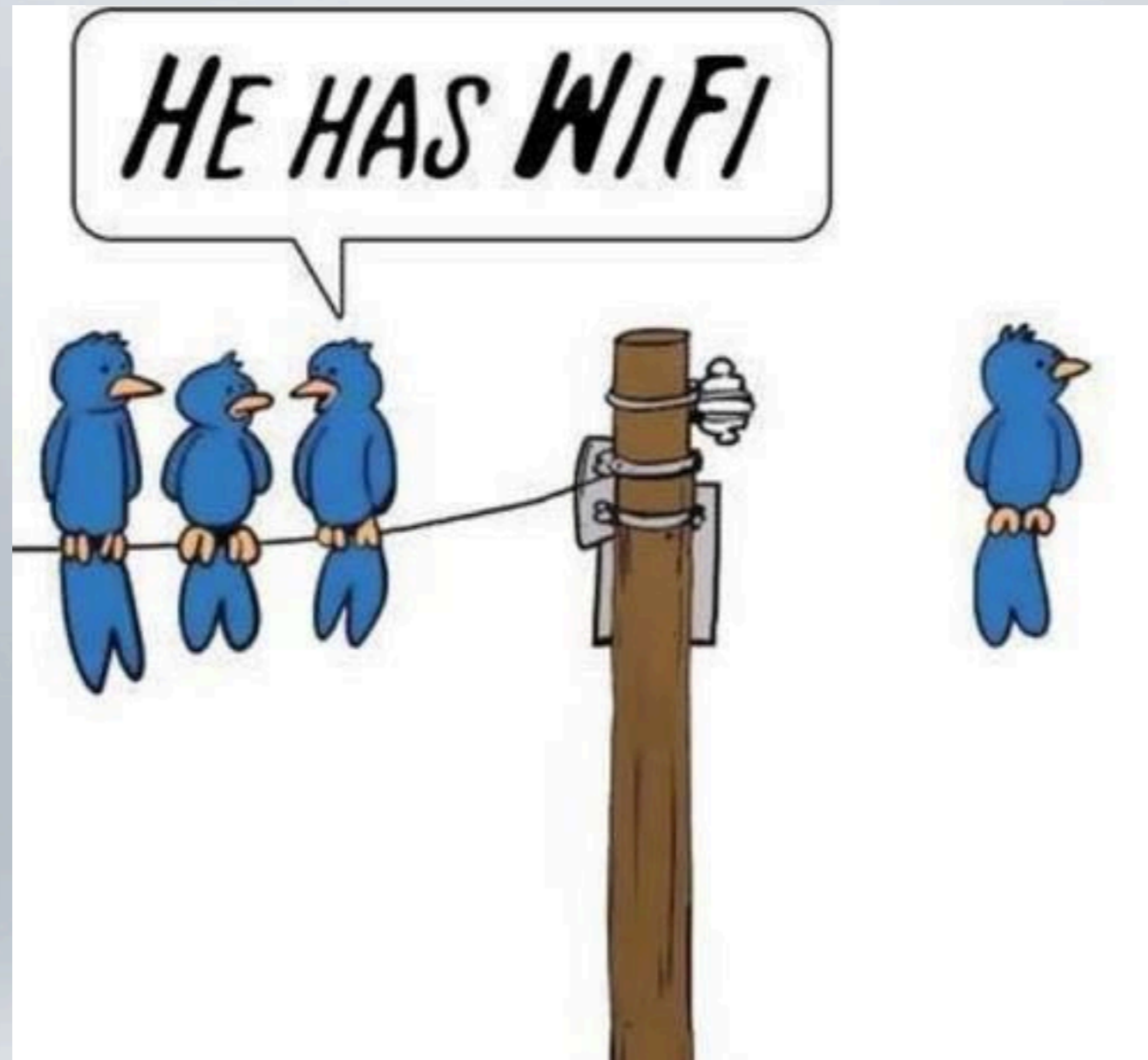


INTERNET OF THINGS

Gilles BLANC — gilles.blanc@linacs-consulting.com

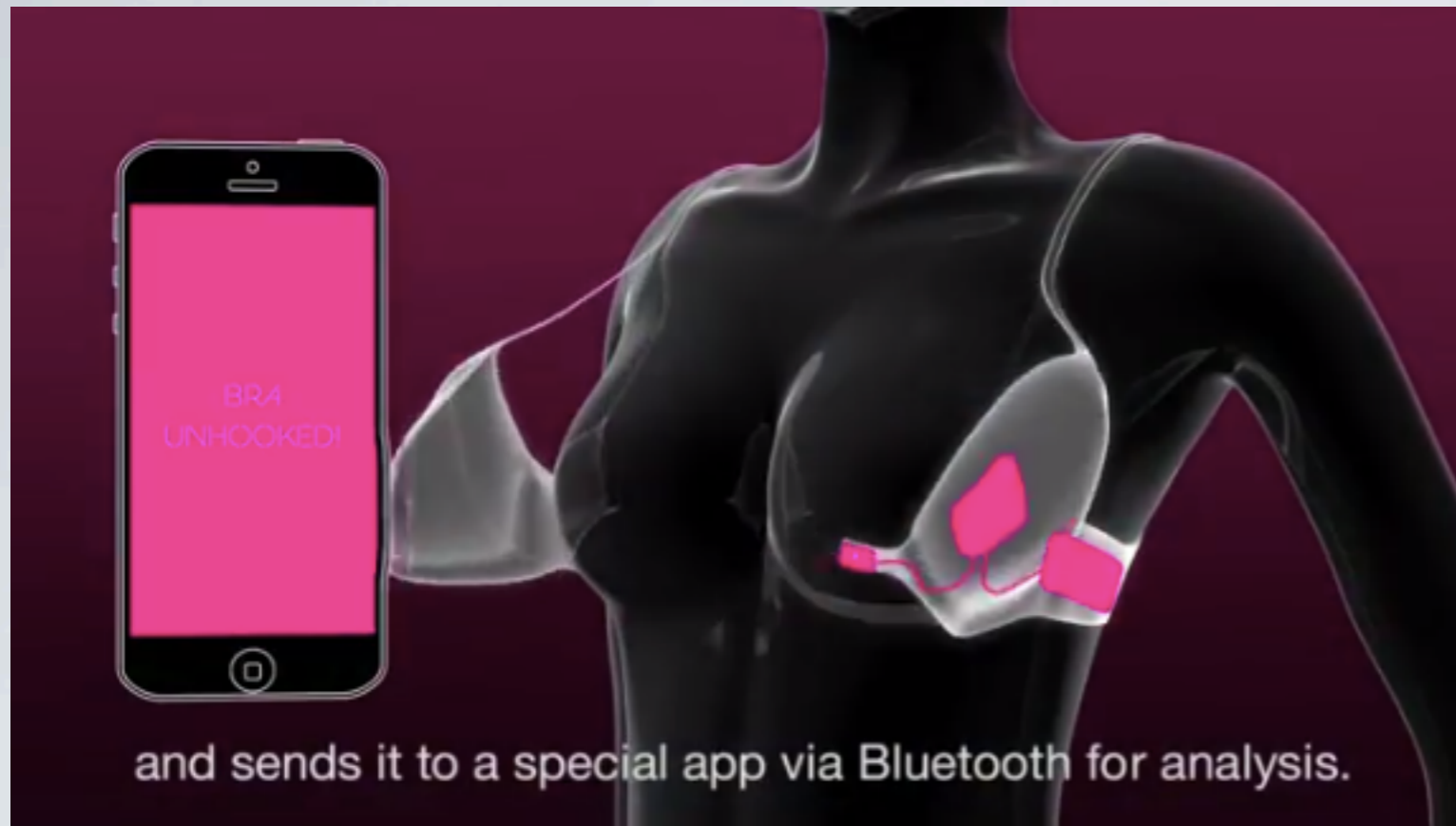




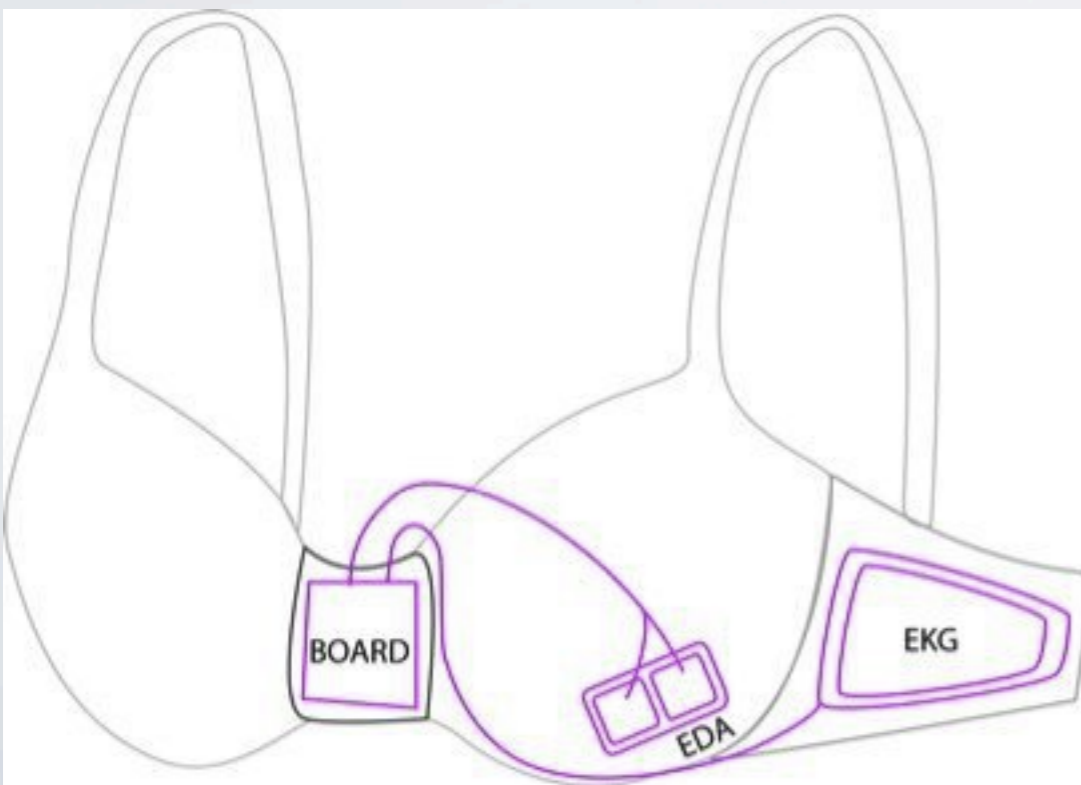
PRÉSENTATION GÉNÉRALE

MAIS QU'EST-CE DONC ??

- Tout appareil contenant du logiciel embarqué et capable de communication
- Un concept marketing
 - « Machine to Machine » aka M2M ne marchait pas très bien
 - ne dépassait pas le cadre des professionnels du secteur depuis 15 ans !
 - n'était pas grand public (marché du « Consumer electronic » aka CE)
 - c'est la crise, il faut trouver de nouvelles idées (mais comme ça n'existe pas, on renomme comme pour le cloud et le big data)
- Un peu moins présomptueux, la version française : « objets connectés »
 - objet : n'importe quoi (iot vs "internet of everything"), du bracelet à la station spatiale internationale (ou presque) en passant par le T-shirt et le soutien-gorge
 - tant que c'est connecté à autre chose via un protocole (radio de préférence) quelconque, plutôt en émission
 - périmètre assez flou (mais pas les drones, étrangement : notion de capteur pas applicable, pas assez passif ?)
- Tout et (beaucoup de) n'importe quoi



Victoria's secret, pour sportives, \$75 (retiré de la vente ?)



Microsoft, détection de boulimie, autonomie de 3 à 4 heures

TOPOLOGIE DE L'IOT

- **Wearable**

- habits connectés, smart watch, bracelets, lunettes...
- instrumentation de l'individu (de préférence sportif)

- **Healthcare/Quantified self**

- idem, mais plus axé santé (poids, chocs articulaires...)

- **Silver economy**

- instrumentation du vieux
- problème moderne

- **Industrie du futur**

- instrumenter les usines
- optimiser les processus de production

- **Retail** : achats (processus, optimisations, caisse...)

- **Home automation**

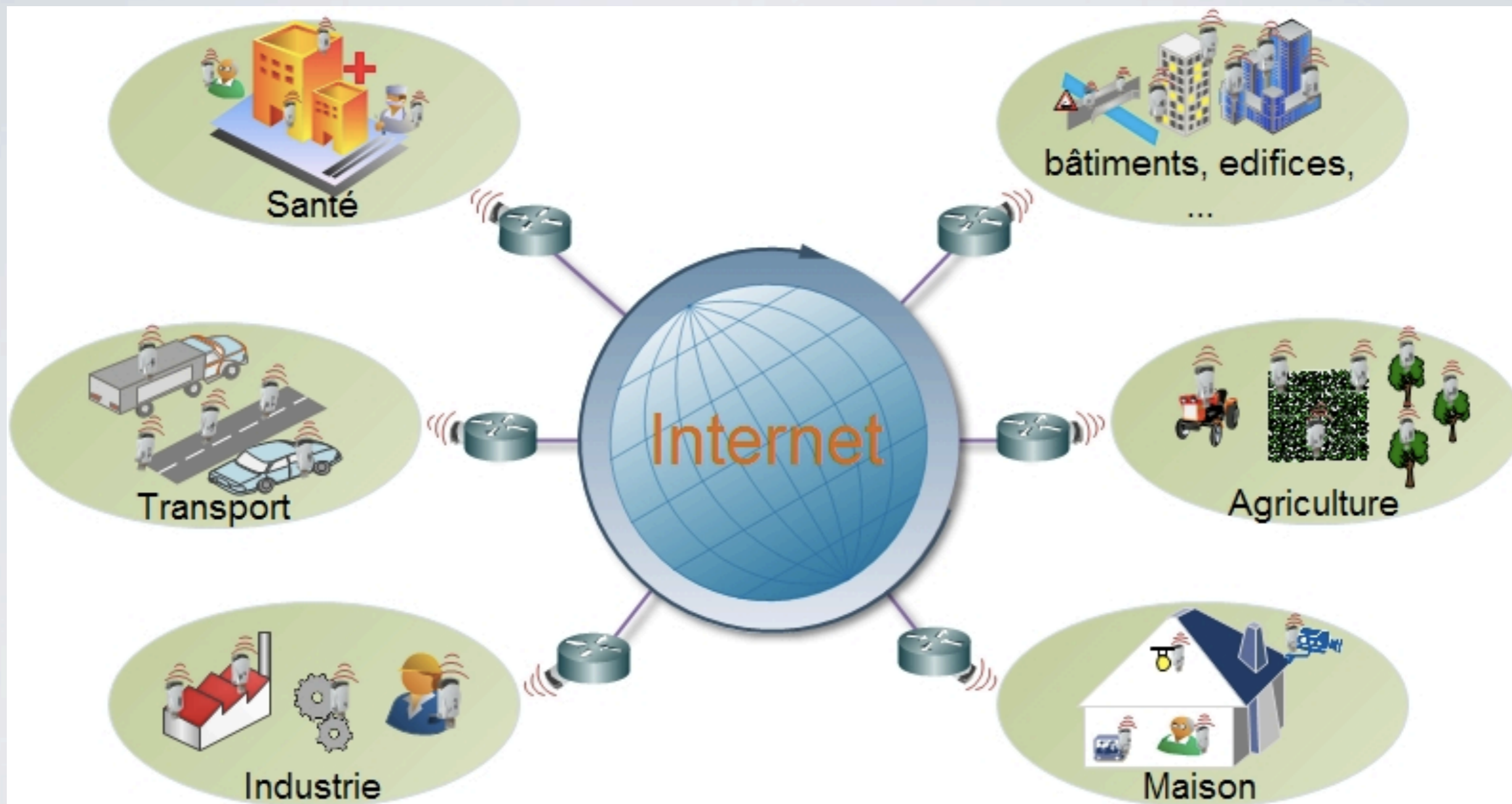
- domotique (since 1980)
- instrumentation de la maison, du placard aux volets en passant par le frigo et la prise électrique

- **Smart cities**

- instrumentation de la ville et des bâtiments
- **Énergie / SmartGrids** : éclairage, effacement (climatisation), compteurs, optimisation des usages
- **Voiture connectée**, parking

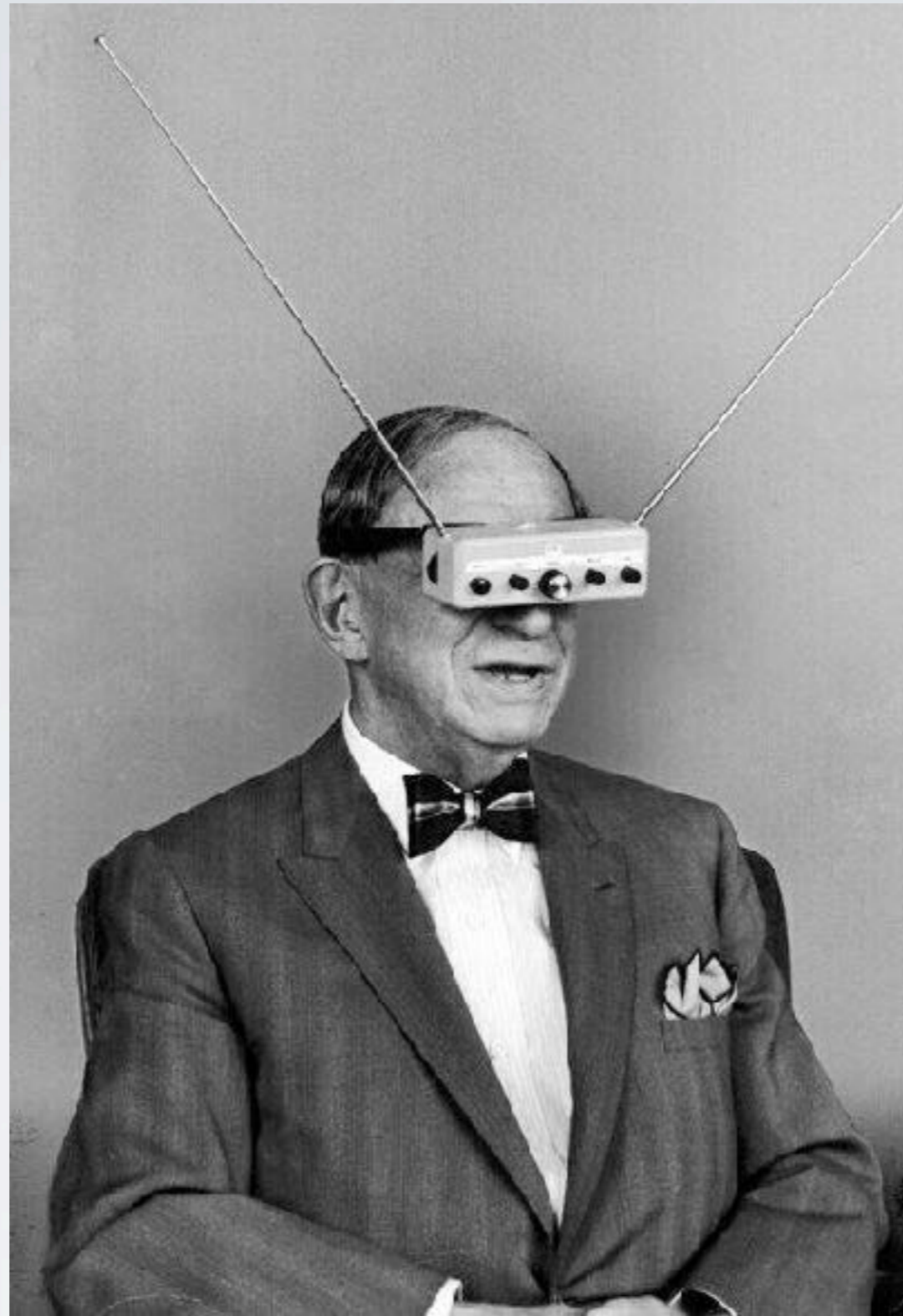
- **Bornes de recharge électrique**

- le moins grand public en apparence
- un marché potentiellement énorme et plus certain que tous les autres



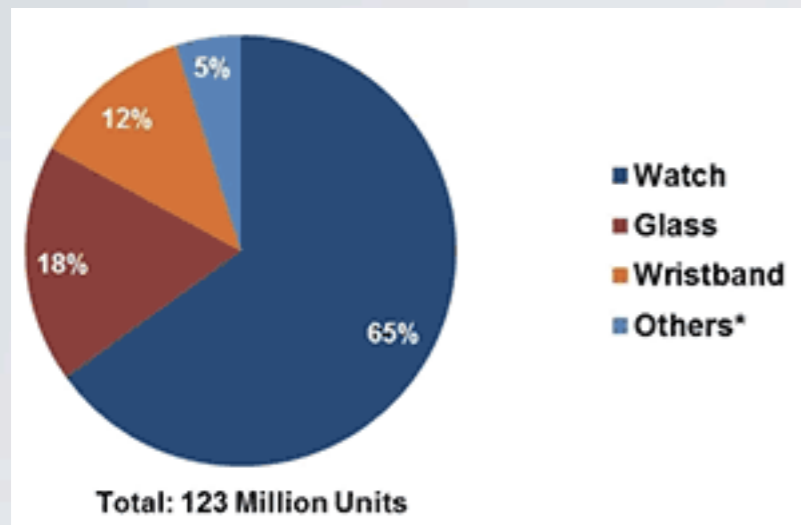


Recharge de voiture électrique, 1905



Inventor Hugo Gernsback demonstrating his television goggles,
1963

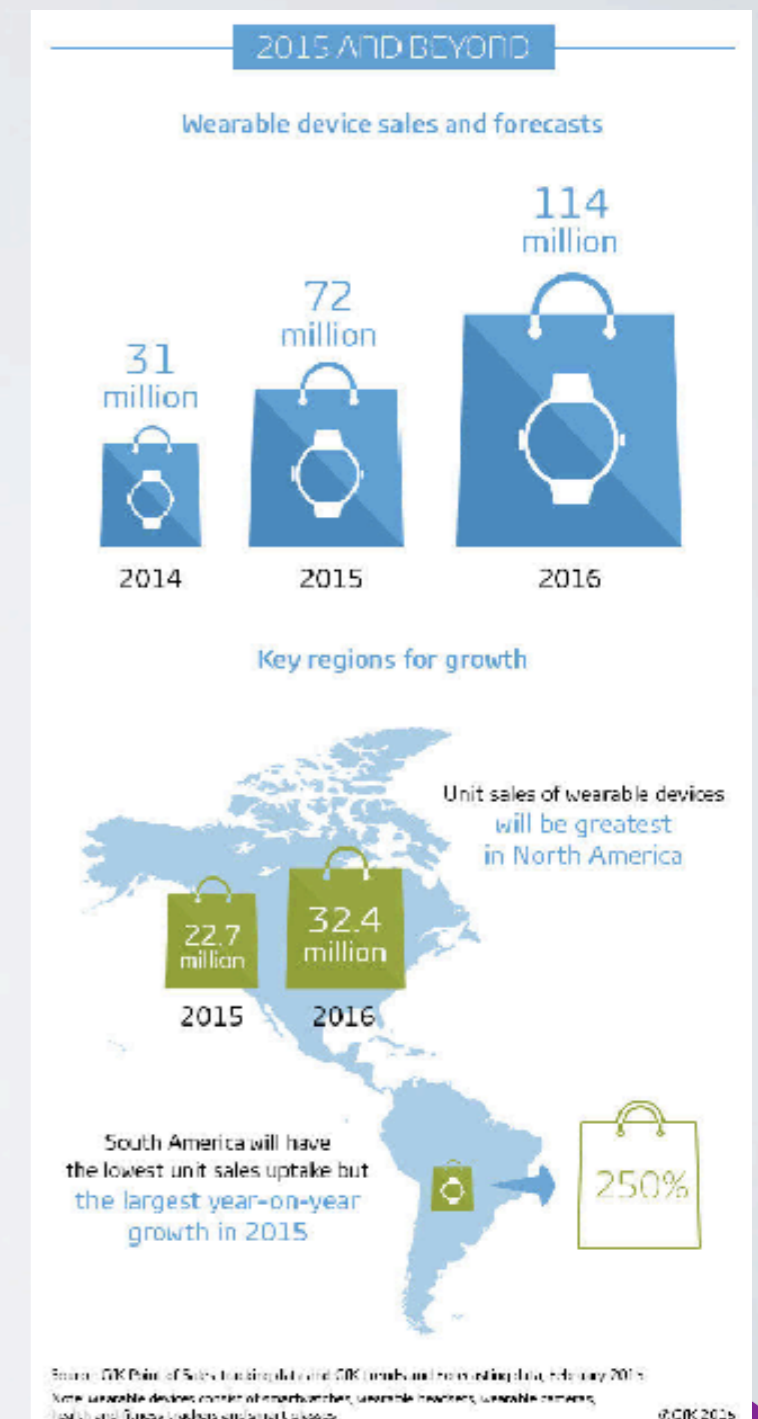
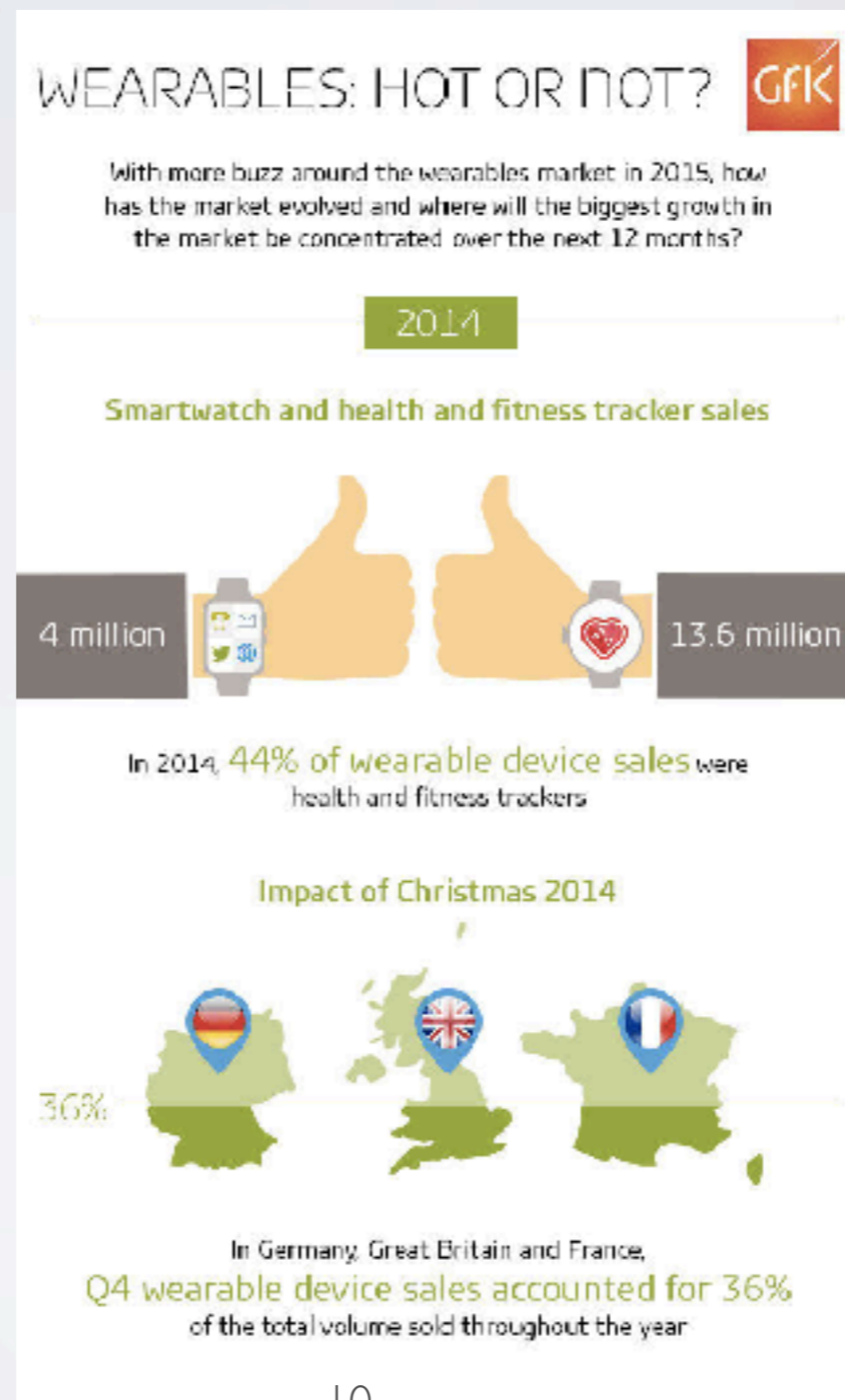
NOSTRADAMUSSISME



*Other: clothing, apparel, etc.
Source: IDATE, in Wearables and its verticals 2018, Dec 2014

« Les opérateurs télécoms affûtent leurs armes dans la bataille qui s'annonce autour des objets connectés. Selon les prévisions, on attend entre 28 et 80 milliards de ce type d'objets (du thermostat à la montre...) d'ici à 2020, avec un chiffre d'affaires estimé à plus de 7.000 milliards de dollars. »

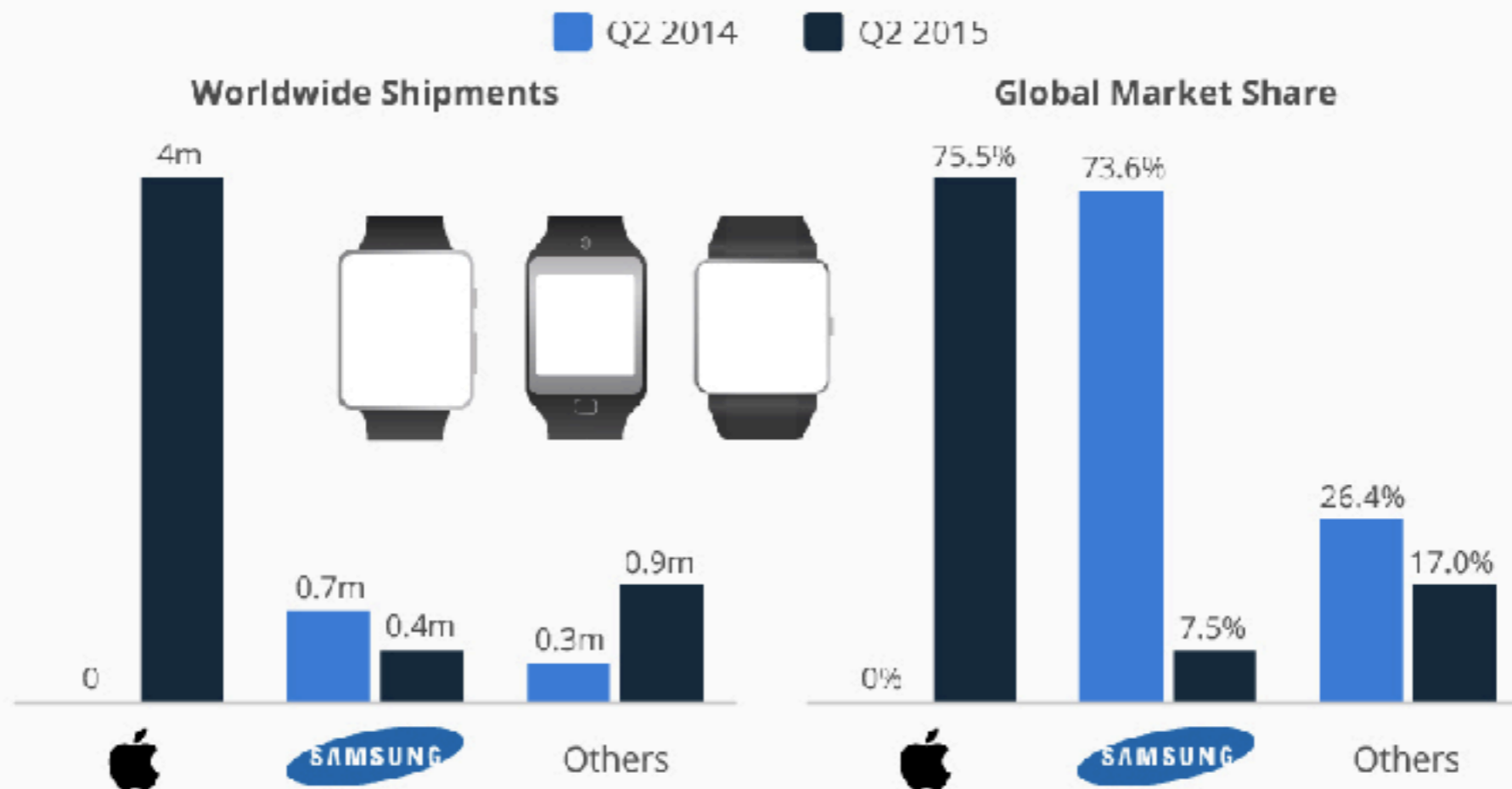
<http://www.lesechos.fr/tech-medias/hightech/021343207067-les-operateurs-face-au-defi-des-objets-connectes-1157729.php?zO3FMZI7J4kpVyTf.99>



SMARTWATCH MARKET (?)

Apple Is Already Dominating the Smartwatch Market

Estimated unit shipments and global market share of leading smartwatch vendors



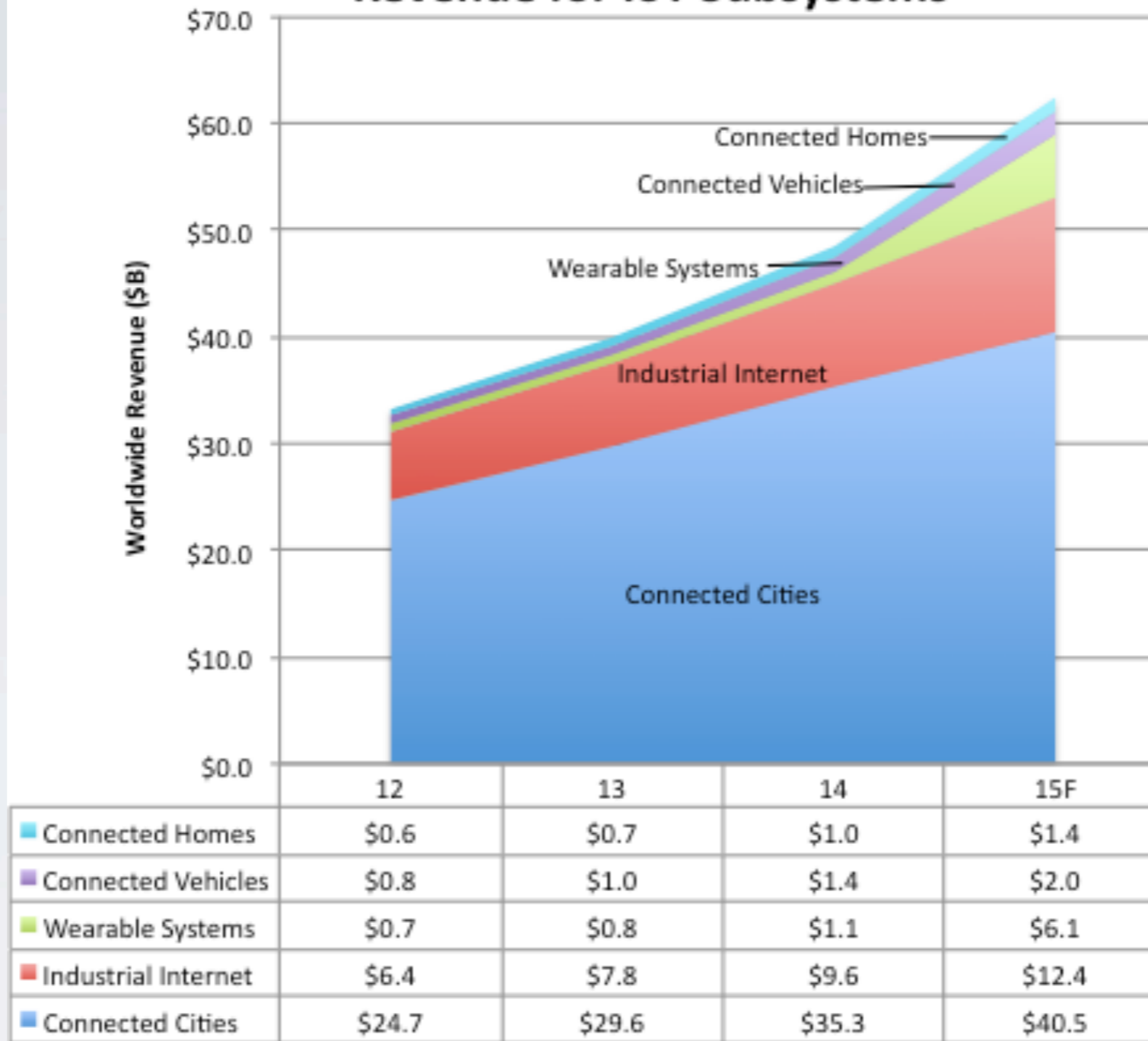
@StatistaCharts Source: Strategy Analytics

statista

GRANDS ACTEURS

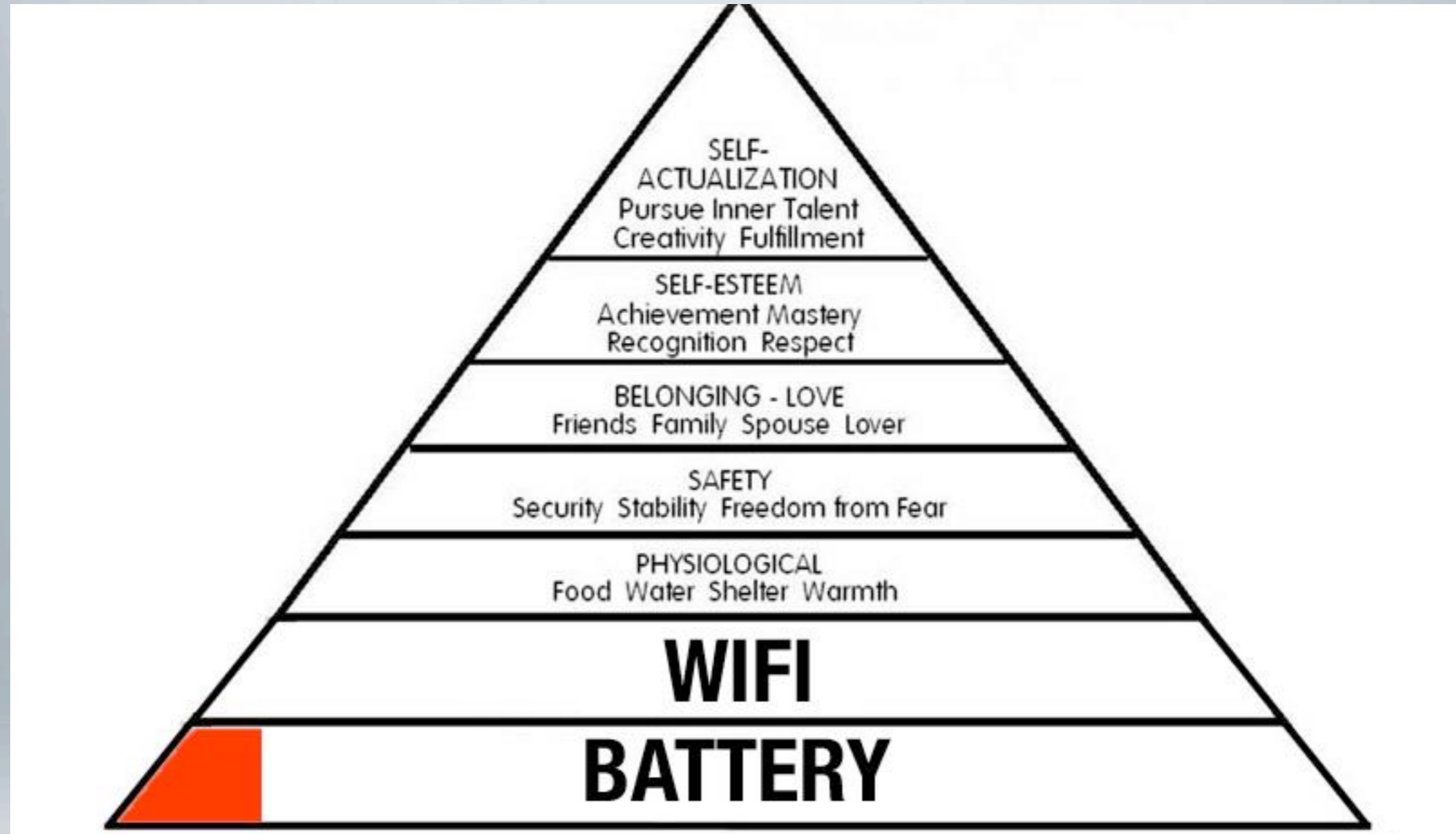
- Smartwatch : Apple, Samsung
- Sport : Fitbit, Nike
- Health : Withings
- Android Wear : Sony, LG, Motorola, Asus
- Smart home : Bticino (Legrand), Somfi, Castorama, Ijenko, Fibaro, Bosch, MyFox, DiO, Nest
- Smart cities : Alstom, Legrand, Schneider, Hager, Bouygues

Revenue for IoT Subsystems



Source: IC Market Drivers 2015 Update

<http://www.icinsights.com/news/bulletins/Wearable-Systems-Give-Major-Boost-To-Total-IoT-Sales-In-2015/>



ARCHITECTURER UN PROJET IOT

RÉFLÉCHIR À LA PROBLÉMATIQUE

- Qu'est-ce que je veux faire ?
- Qu'est-ce que je veux vendre ? Quel est le business model ? (hw, service, hw + service)
- Combien vais-je en vendre ?
- À quel prix vais-je le vendre ?
- Dans quel environnement va se trouver mon objet ?
- Quelle durée de vie, quelle évolutivité ?
- En déduire le matériel, le logiciel et l'architecture générale adéquats, le tout étant très intrinsèquement lié, en fonction des coûts et de ce qui est techniquement possible !

CHOIX DU HARD

- deux grandes familles : MCU ou CPU ?

- MCU

- microcontrôleur (sans MMU)
- très basse consommation d'énergie, peut tourner sur pile bouton durant des années ou energy harvesting
- un seul espace d'adressage = peu d'adressage mémoire (quelques ko, voire 2Mo pour les systèmes graphiques), peu de processus (avec risque de plantage global du système)
- ARM M0, M3, M4 (gamme STM32 par exemple), Artik1 de Samsung, Microchip's XLP 16-bit (PIC24F16KA102), TI MSP430, Freescale MCI 322x et gamme Kinetis, Atmel SAM3X8E (M3) ou ATmega32u4 (AVR 8 bits)...

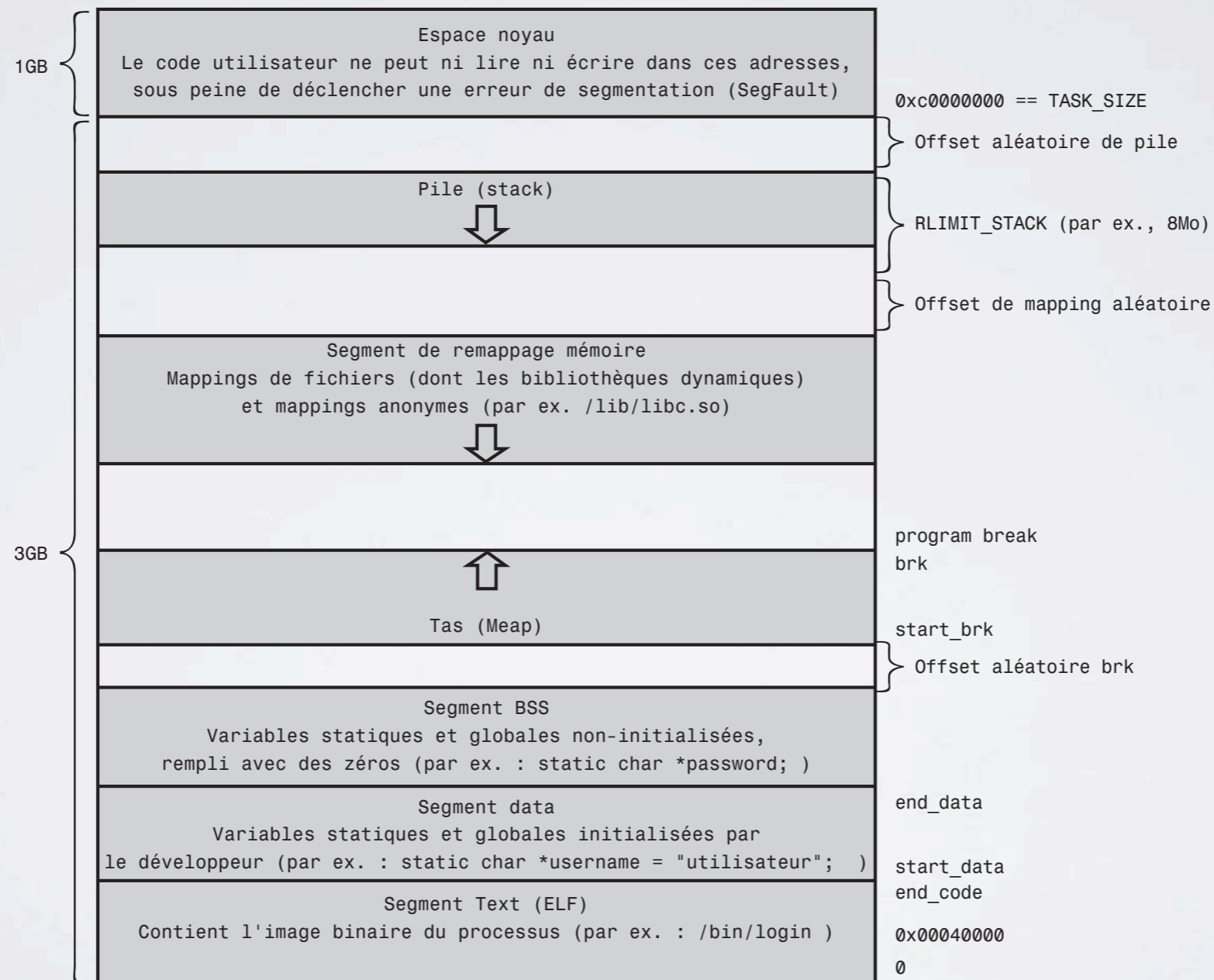
- CPU

- microprocesseur (avec MMU)
- consommation d'énergie nécessitant une batterie rechargeable
- souvent accompagné de DSP et autres bloc IPs (cf SoC)
- ARM Cortex A5, A7, A9 ; Intel Quark (x86, 32 bits)
- Hybride CPU + MCU
 - meilleur des deux mondes
 - calcul + temps-réel
 - exemple : iMx6-SoloX

MÉMOIRE VIRTUELLE & MMU

- notion de mémoire réelle et de mémoire virtuelle, à travers un composant matériel du CPU, la Memory Management Unit
- un microcontrôleur (PIC, Coldfire, ARM M0...) n'a pas de MMU :
 - adressage direct en mémoire réelle
 - pas de protection (un programme peut en faire planter un autre, voire tout le système)
- un CPU de type x86, ARM Cortex, etc. a une MMU :
 - adressage en mémoire virtuelle, chaque processus est séparé
 - problématique de « translation » des adresses (tables/caches matériels et logiciels)
 - espaces protégés : un programme qui plante n'a pas d'impact pour les autres (bac à sable)

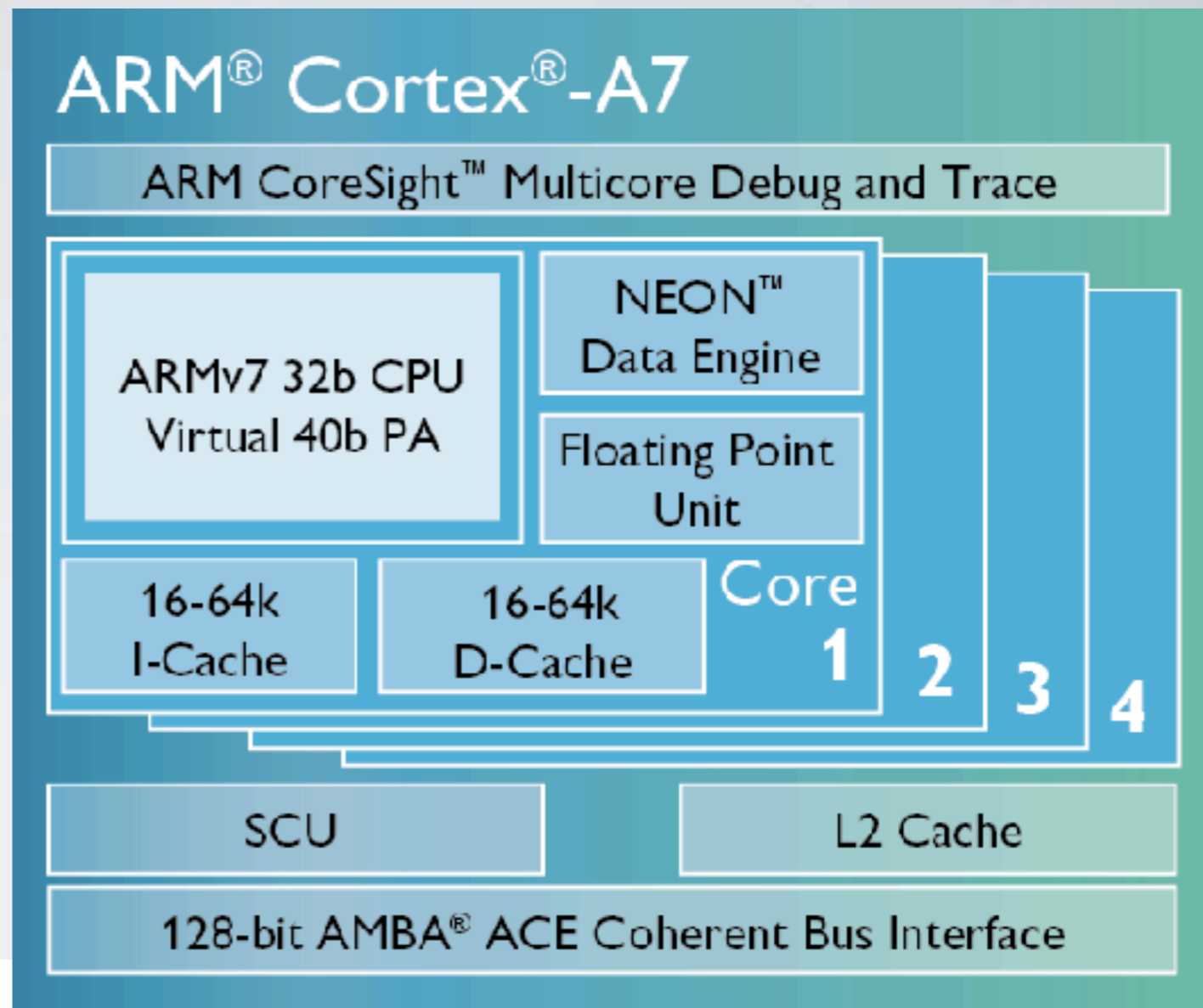
EXEMPLE : PROCESSUS LINUX, MÉMOIRE VIRTUELLE



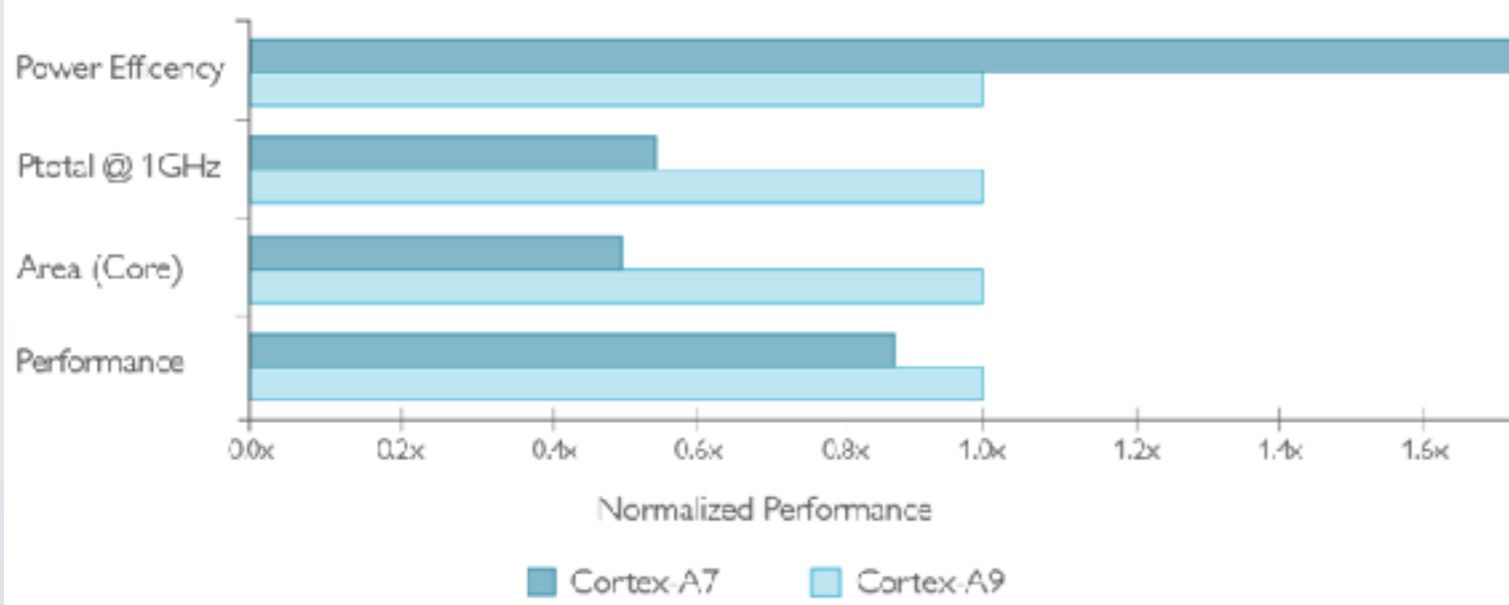
famille STM 32

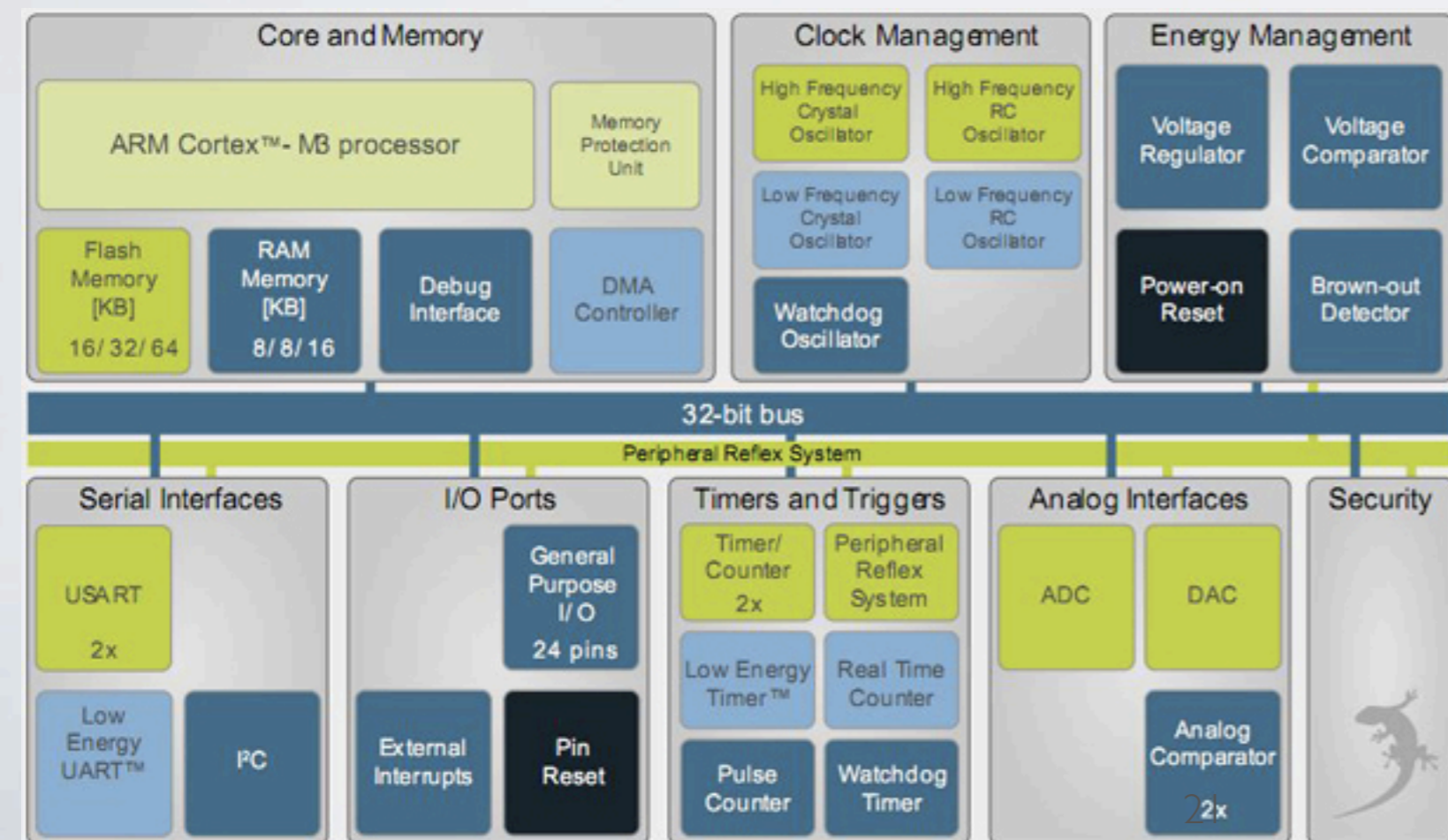
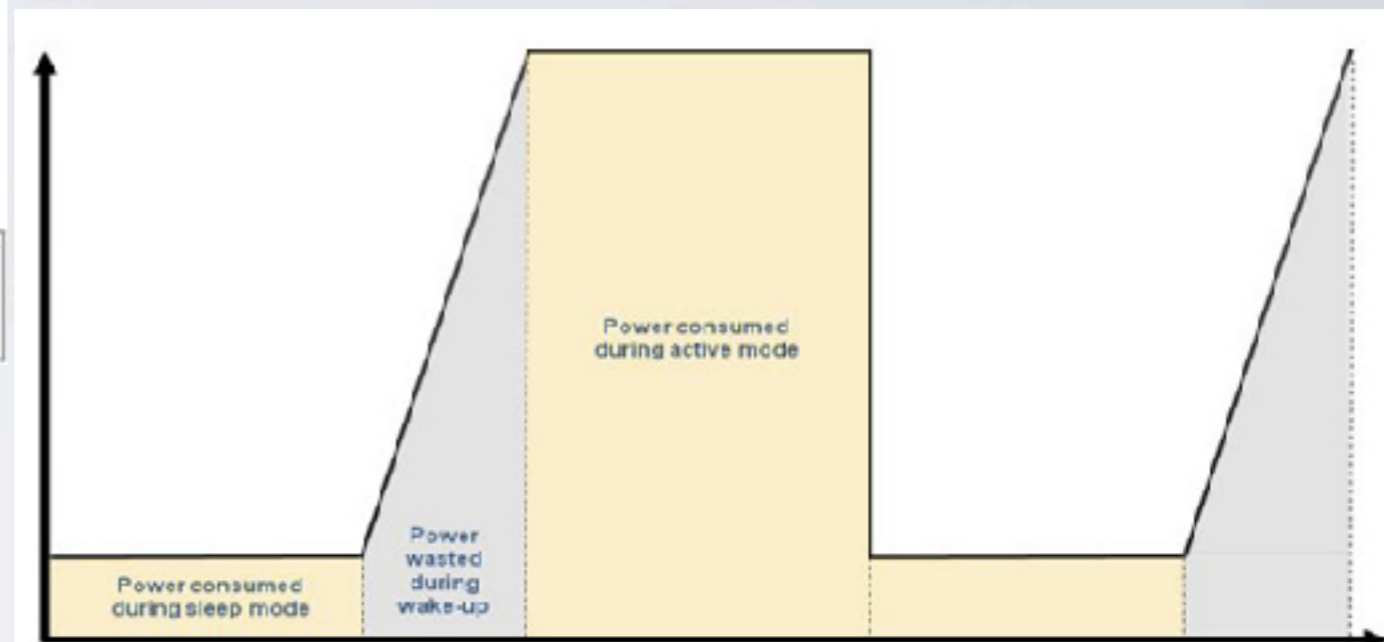
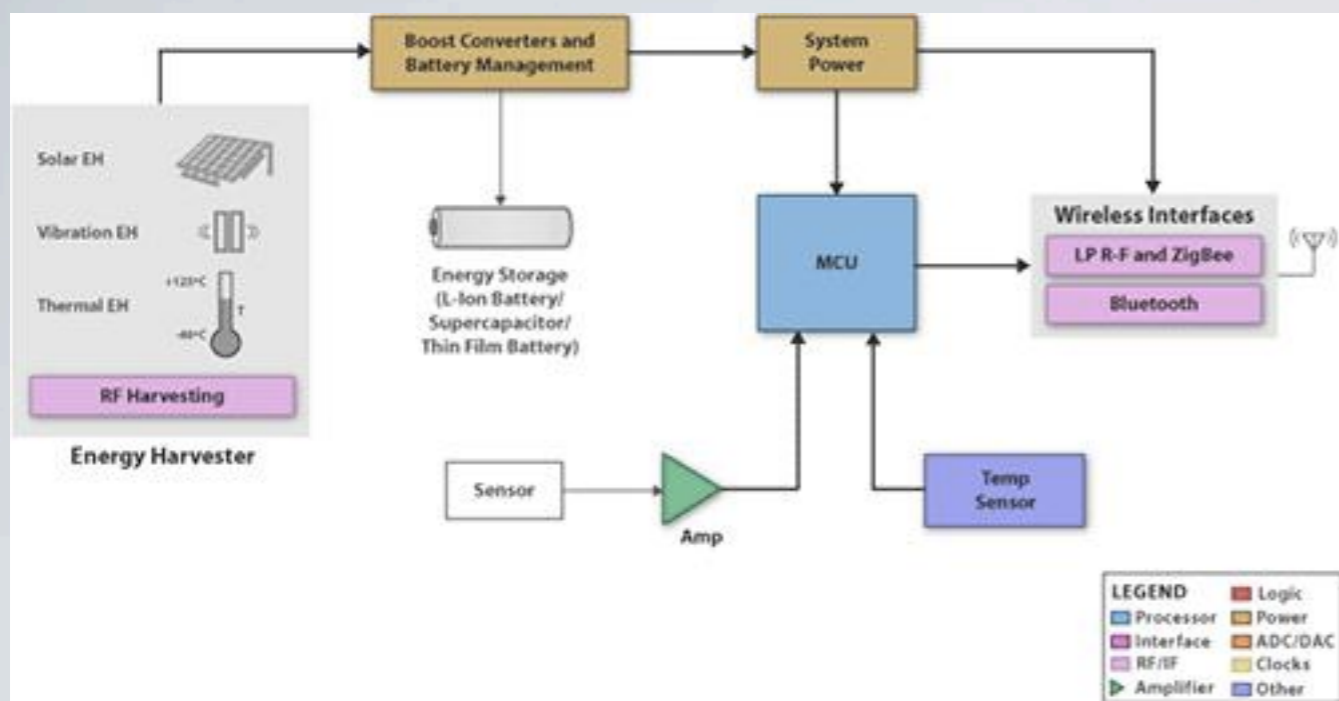


ARM Cortex A7



Cortex-A7 Power Efficiency Relative to Cortex-A9





<http://www.digikey.com/en/articles/techzone/2011/jun/ultra-low-power-mcus-enable-energy-harvesting-designs>

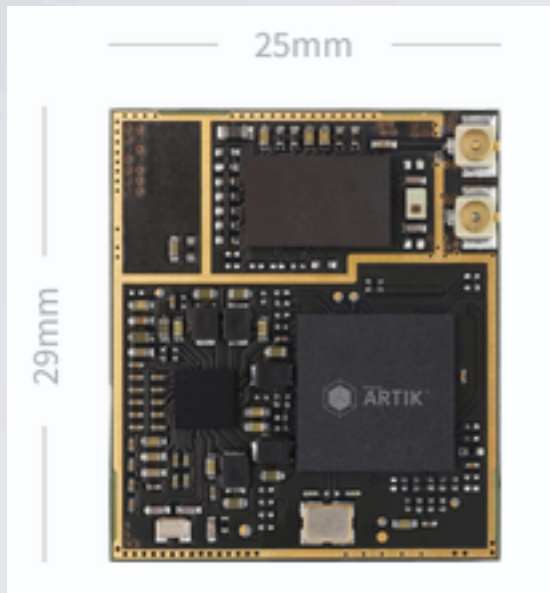
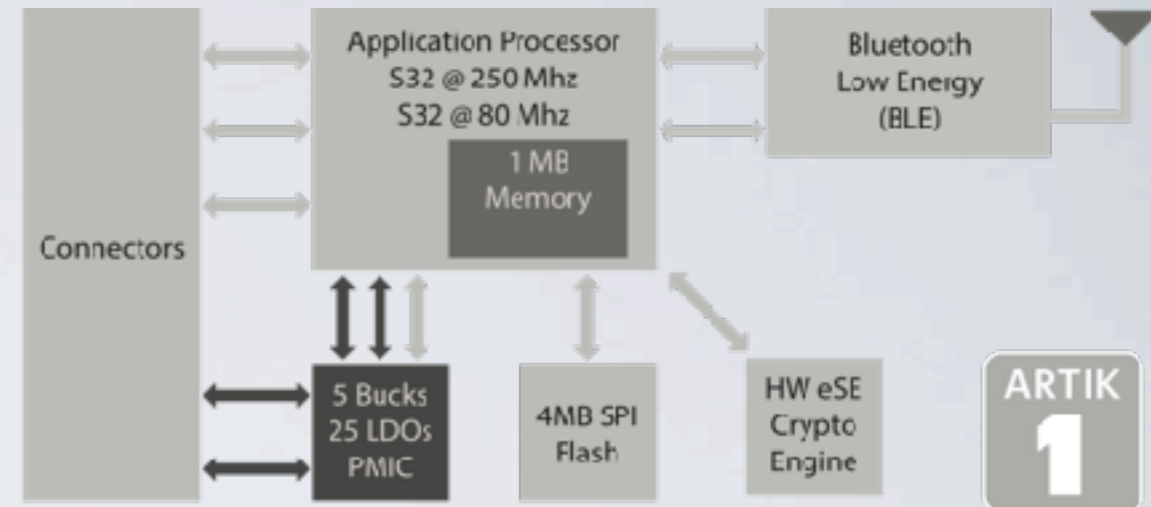


ARTIK 1

Smallest module in its class

- Small size (12mm x 12mm)
- Low power consumption
- Built in BLE

[LEARN MORE](#)

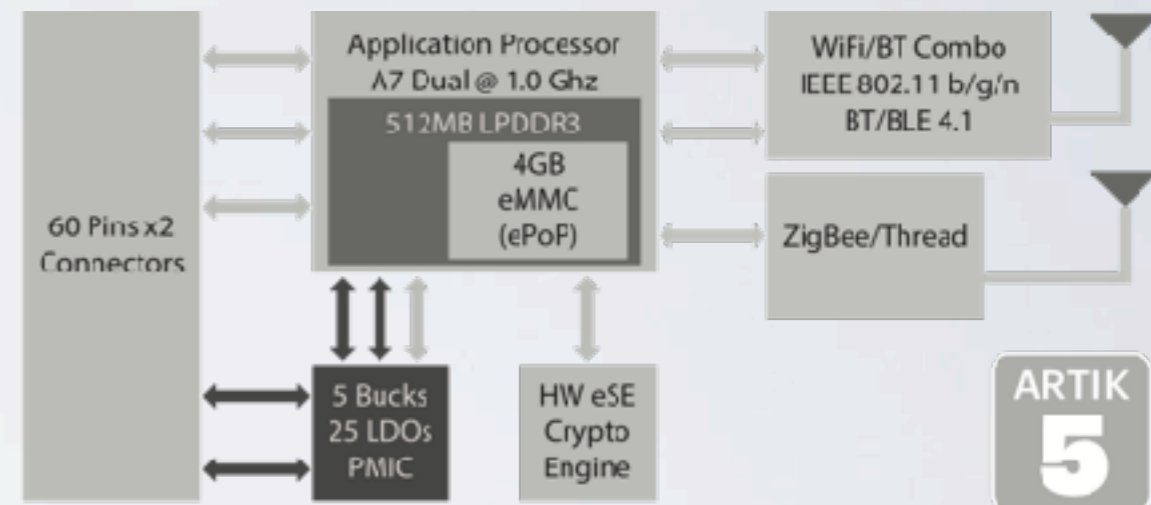


ARTIK 5

Balances size and power

- Dual-core ARM chip
- Hardware video decoding (720p)
- Built-in WiFi, BLE, ZigBee, Thread

[LEARN MORE](#)

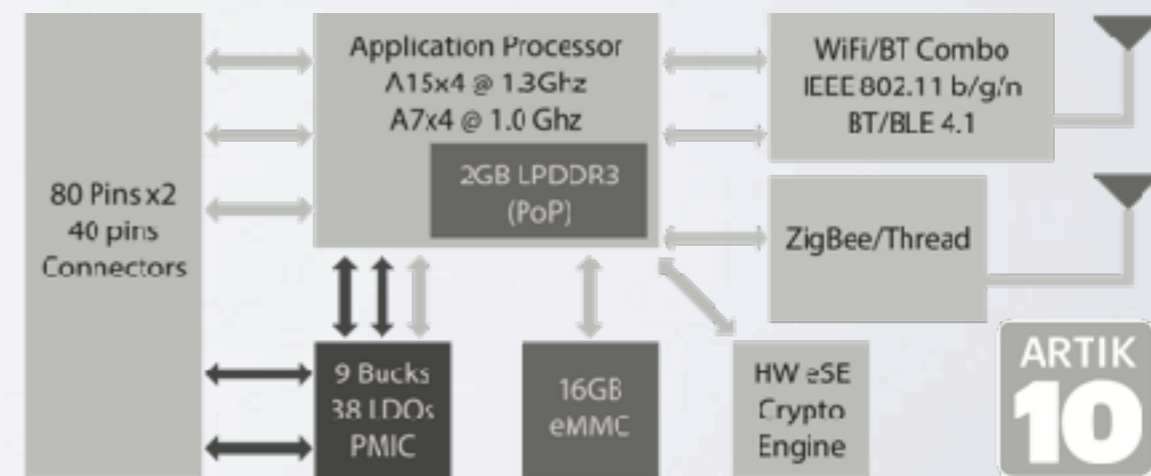


ARTIK 10

Industry-leading performance

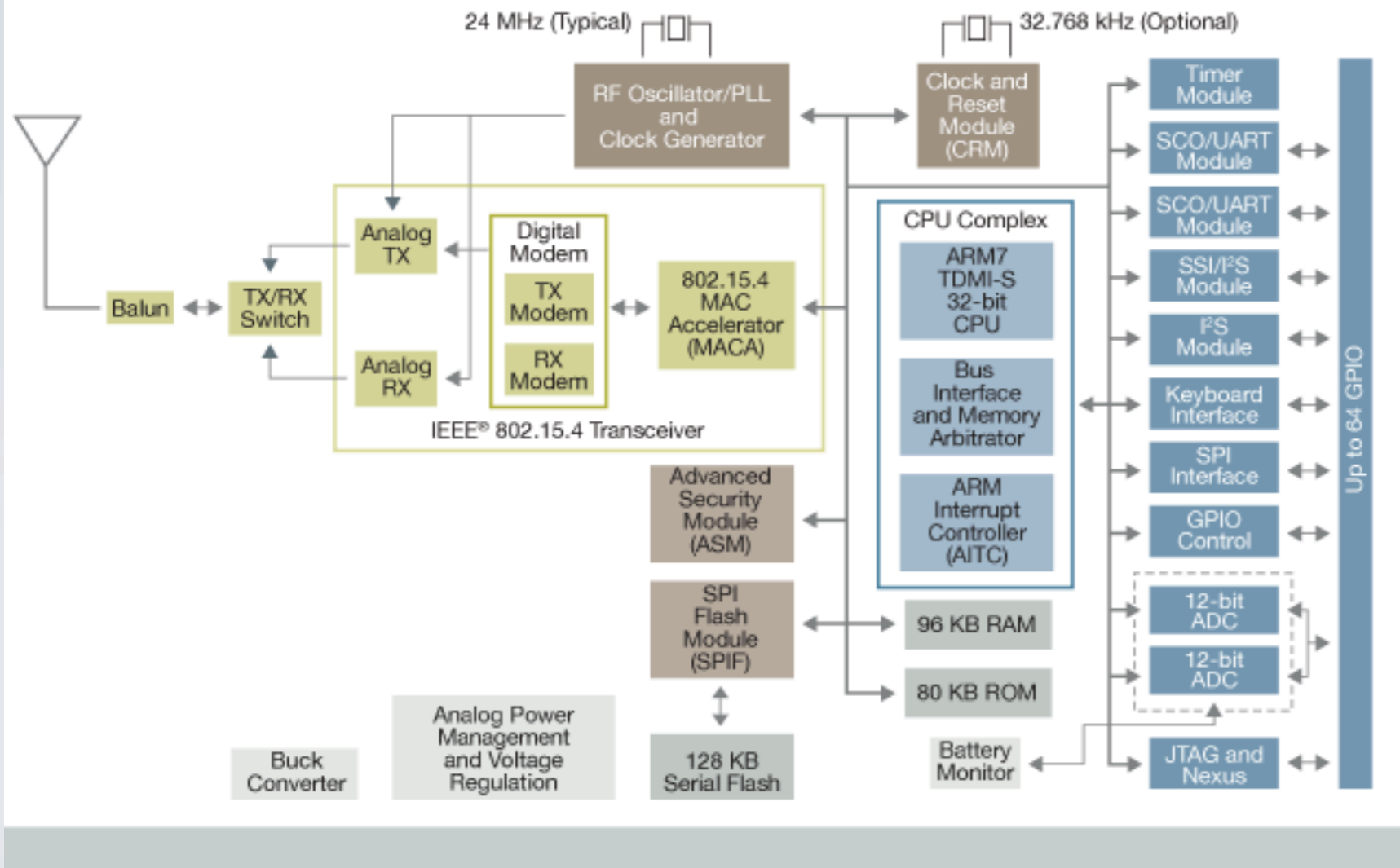
- Octa-core ARM chip
- Hardware video decoding (1080p)
- Built-in WiFi, BLE, ZigBee, Thread

[LEARN MORE](#)





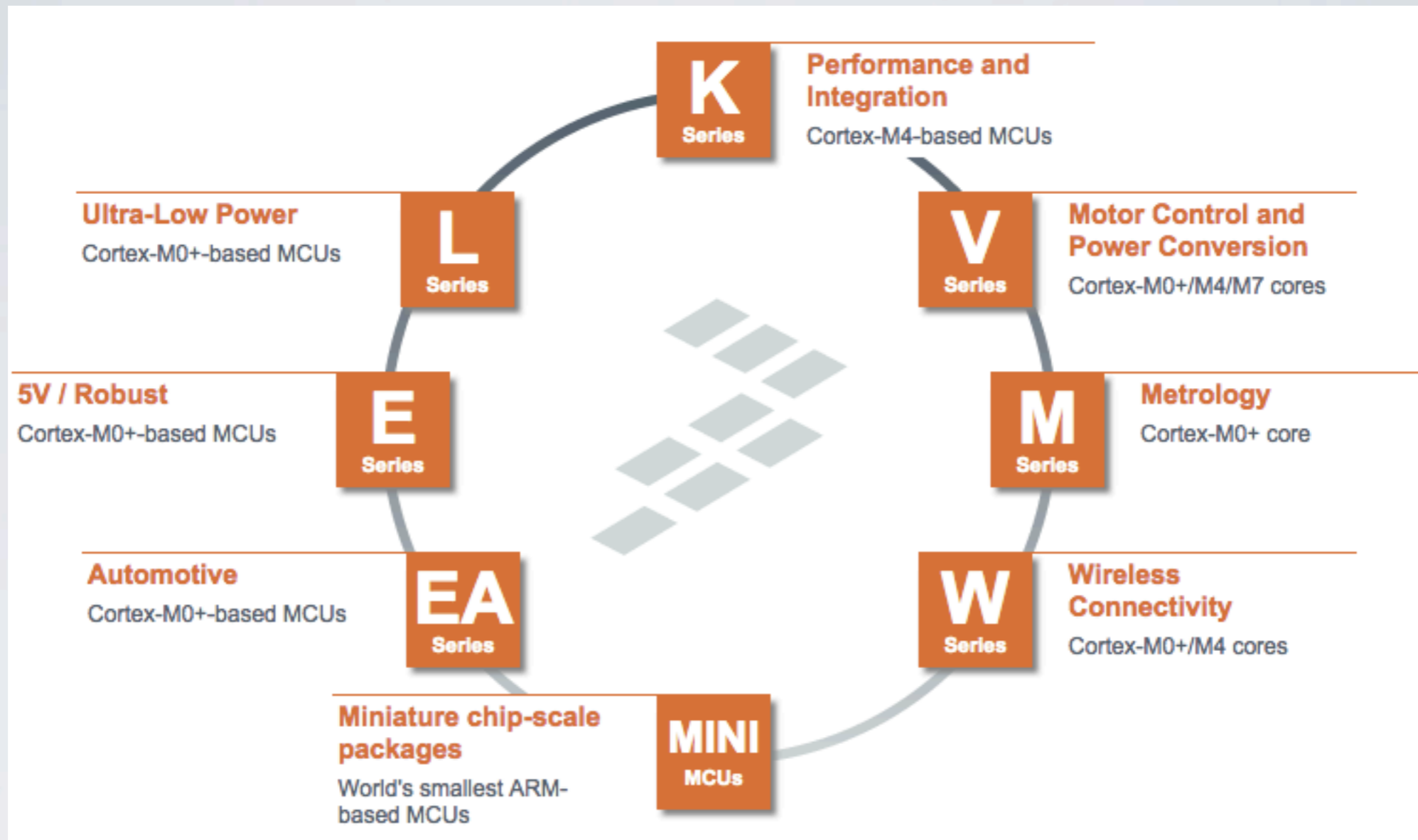
MC1322x Block Diagram



Features

1. Low power
 1. 21 mA typical current consumption in RX mode with MCU active
 2. 29 mA typical current consumption in TX mode with MCU active
2. 128 KB serial flash
3. 96 KB RAM (device operates from RAM)
4. 80K ROM containing bootcode, all device drivers and compliant IEEE 802.15.4 MAC
5. MAC accelerator (sequencer and DMA interface)
6. AES 128-bit hardware encryption/decryption with random number generator
7. JTAG debug port
8. Nexus extended feature debug port
9. No external RF components required
 1. RF matching components and balun inside package
 2. Only external connections are crystal and 50 ohm matched antenna

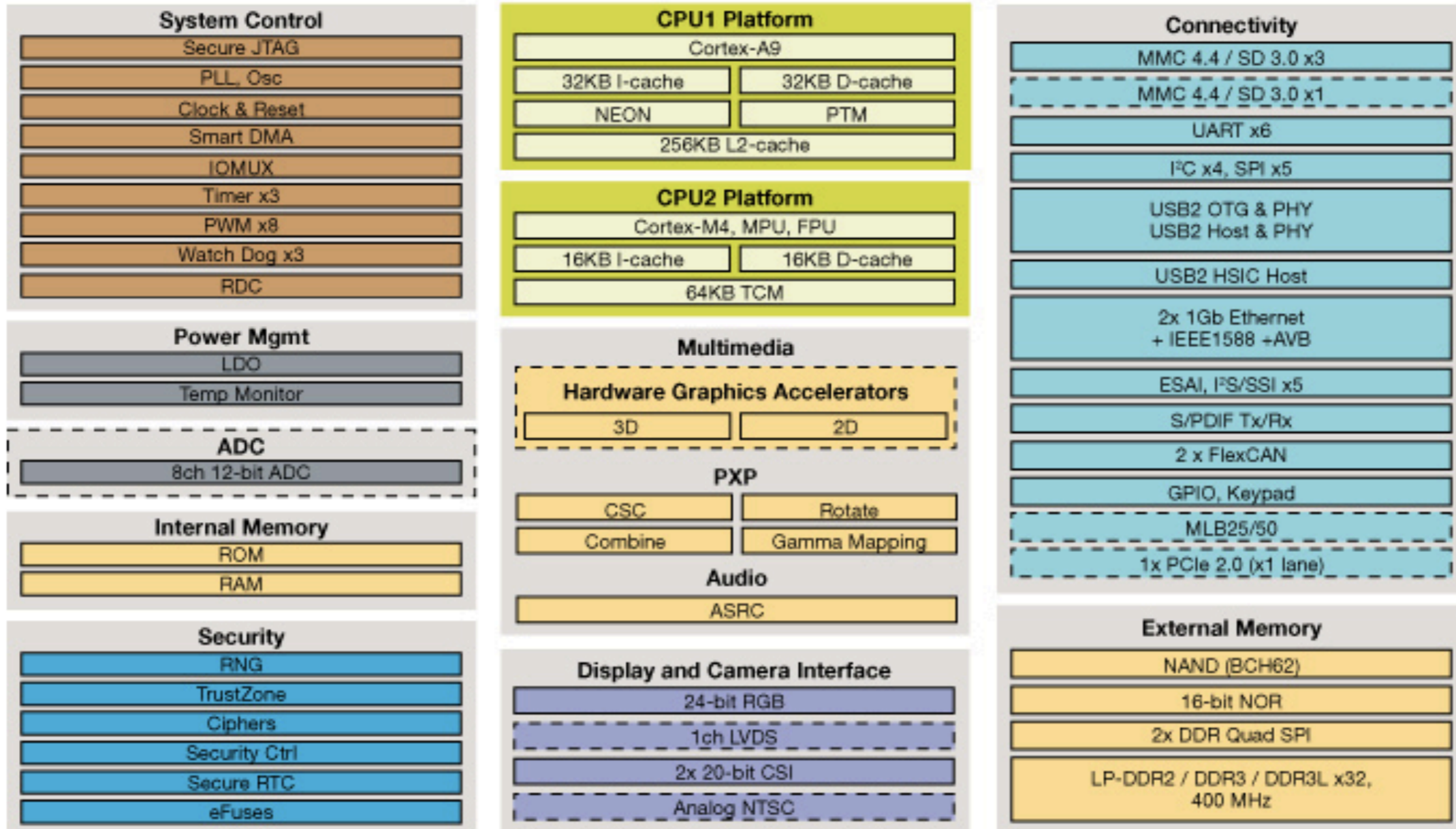
<http://www.freescale.com/products/wireless-connectivity/2.4-ghz-wireless-solutions/2.4-ghz-802.15.4-rf-and-32-bit-arm7-mcu-with-128kb-flash-96kb-ram:MC13224V>



<http://www.freescale.com/products/arm-processors/kinetis-cortex-m:KINETIS>



i.MX 6SoloX Applications Processor Block Diagram



Standard Feature Optional Feature

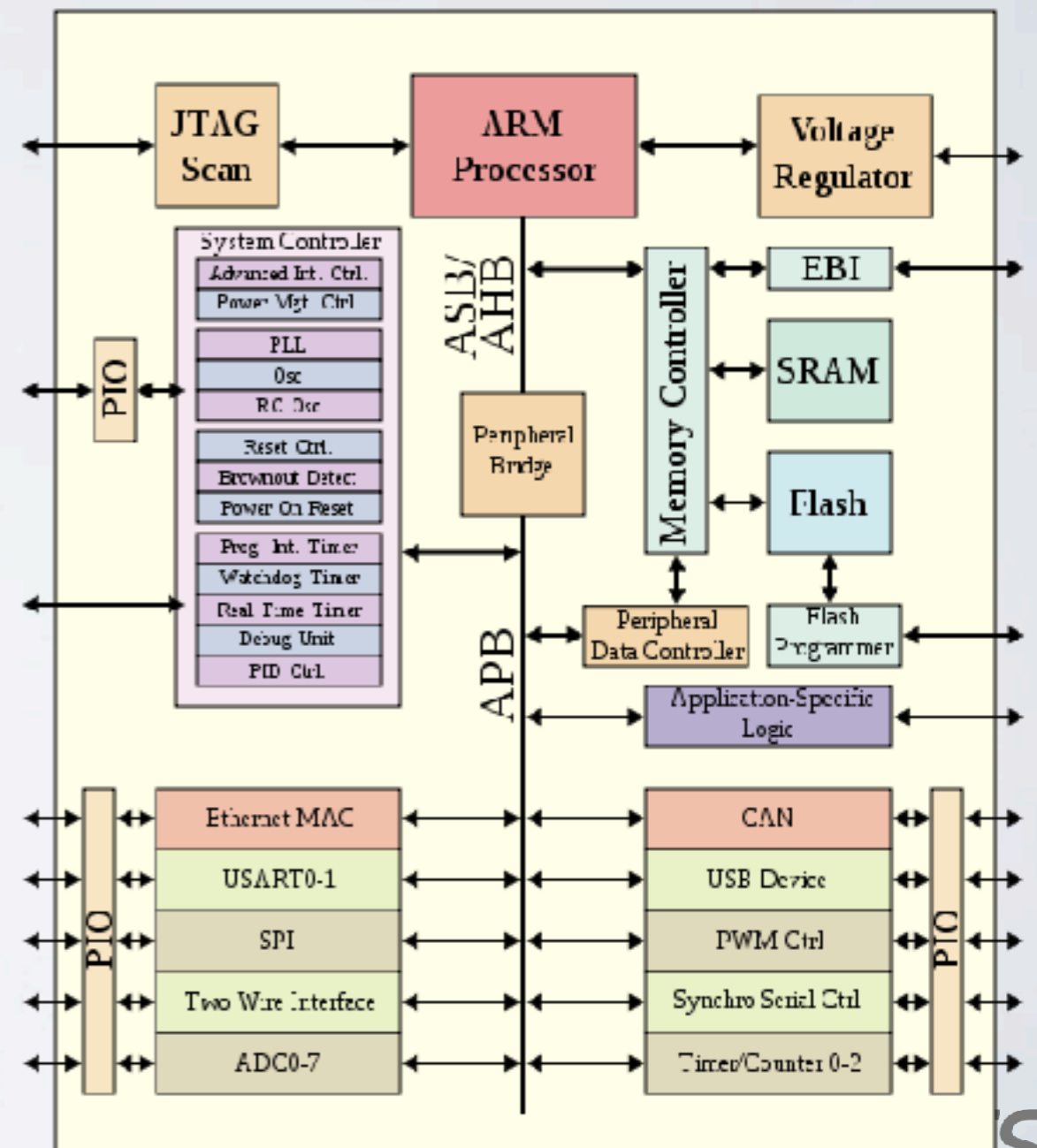
<http://www.freescale.com/products/arm-processors/i.mx-applications-processors-based-on-arm-cores/>

[i.mx-6-processors:IMX6X_SERIES](#)



SYSTEM ON CHIP (SOC)

- MCU, CPU ou DSP — les SoCs multiprocesseurs (MPSoC) ont plus d'un coeur de processeur
 - bloc de mémoire parmi : ROM, RAM, EEPROM et flash
 - source de temps (clocks) dont oscillateurs et Boucle à phase asservie (phase-locked loops)
 - périphériques dont counter-timers, real-time timers et power-on reset generators
 - interfaces externes dont des standards industriels comme USB, FireWire, Ethernet, USART, SPI
 - interfaces analogiques dont ADCs et DACs
 - régulateur de tension et gestion d'alimentation (power management)
- Pour les ARM Cortex-A* : AllWinner, Rockchip, Exynos de Samsung, MediaTek, OMAP de Texas Instrument, Tegra de nVidia, Snapdragon de Qualcomm...



STOCKAGE

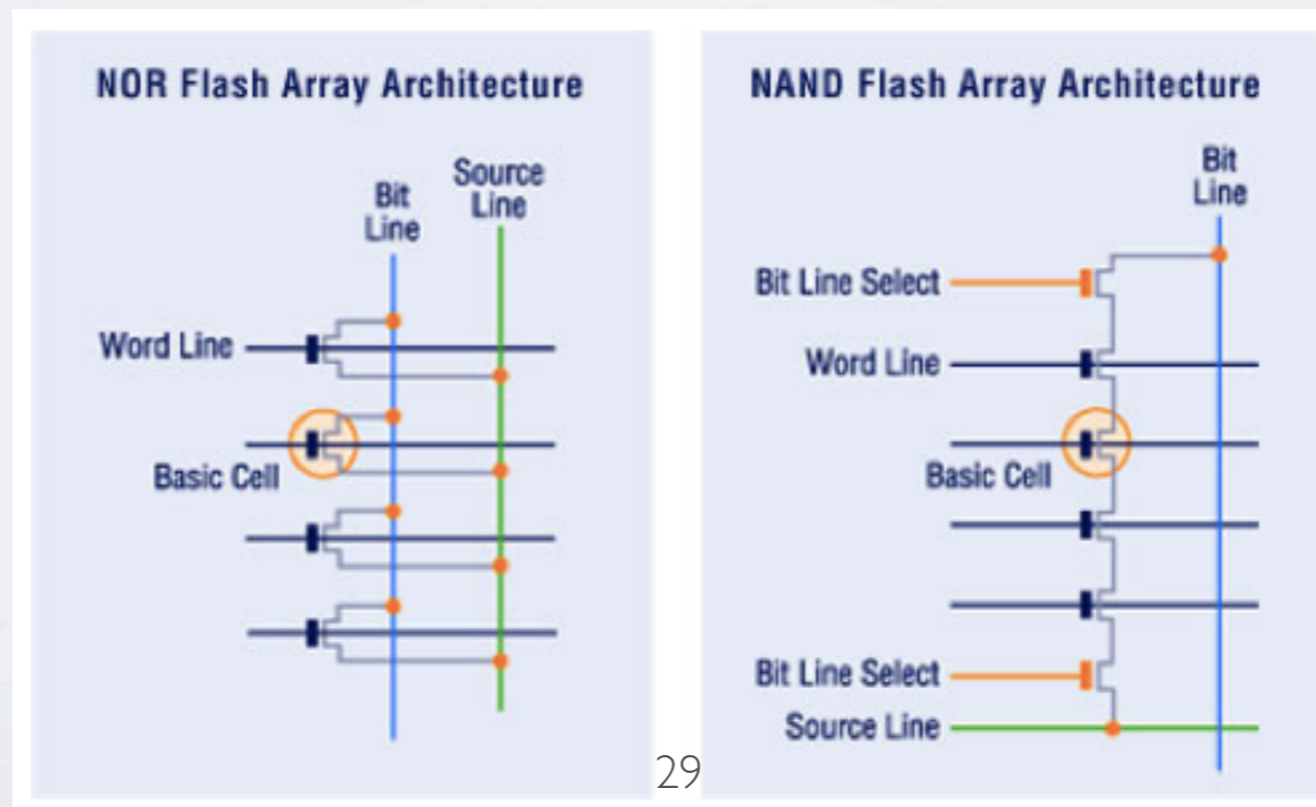
- Différents types de stockage de type flash
 - ROM : espace interne au SoC, non effaçable (contient le microboot)
 - EEPROM : puce externe effaçable électriquement, pour de petites données de configuration (par exemple : adresse MAC)
 - Flash (NAND/NOR) : espace de stockage du système, pour le développeur
 - taille très variable : 512 ko si MCU, mais 256 Mo si MMU
 - eMMC : même techno qu'une carte SD, mais sur puce soudée (ou dans le millefeuille), derrière un contrôleur (précision importante !), utilisée pour un espace > 256 Mo (4 à 16 Go !)
 - SDCard/MMC : connecté directement sur les PINs du SOC mais derrière un contrôleur et très sensible aux changements de tension (NE JAMAIS METTRE EN PRODUCTION !!!)
 - SSD : sur bus SATA ou PCIe, pour des espaces > 16 Go (128, 256 voire 512), existe en version industrielle (a totalement remplacé les anciennes CompactFlash)

LA FLASH 1/2

- Mémoire de stockage rémanent de plus en plus rencontrée, depuis plusieurs années dans les systèmes embarqués (beaucoup plus d'espace mémoire que les EEPROM/ROM classiques)
- Transistors comme support (basé sur l'effet tunnel — Fowler-Nordheim)
- Par défaut, les bits sont tous à 1
- Programmer (« écrire ») = changer certains bits de 1 à 0 en appliquant des tensions
<http://www.tomshardware.fr/articles/ssd-flash-disques,2-393-2.html>
- Effacer = changer les bits à 1
- Deux types : NAND et NOR
- Programmation et effacement au bit ou à l'octet
 - pour la NAND :
 - taille de l'unité de programmation = la page (généralement 4ko)
 - taille de l'unité d'effacement = le bloc (généralement 256 ou 512ko)
 - pour la NOR : accès aléatoire (c'est-à-dire à n'importe quelle donnée, comme dans la RAM)

LA FLASH 2/2

- Durée de vie limitée sur les cycles de lecture/écriture : ~ 100K à ~ 1M effacements pour la NAND, ~ 10K à ~ 100K pour la NOR
- Problèmes physiques de corruption de données
 - parfois même en programmant une page de NAND à côté de celle qui se corrompt !
Ou encore si cycle de lecture interrompu brutalement, pire si écriture.
 - bit flipping (rare sur NAND, extrêmement rare sur NOR)
 - bad blocks : problématique propre à la NAND, il y a toujours de mauvais blocs dès la fabrication (leur densité dépend du prix...)

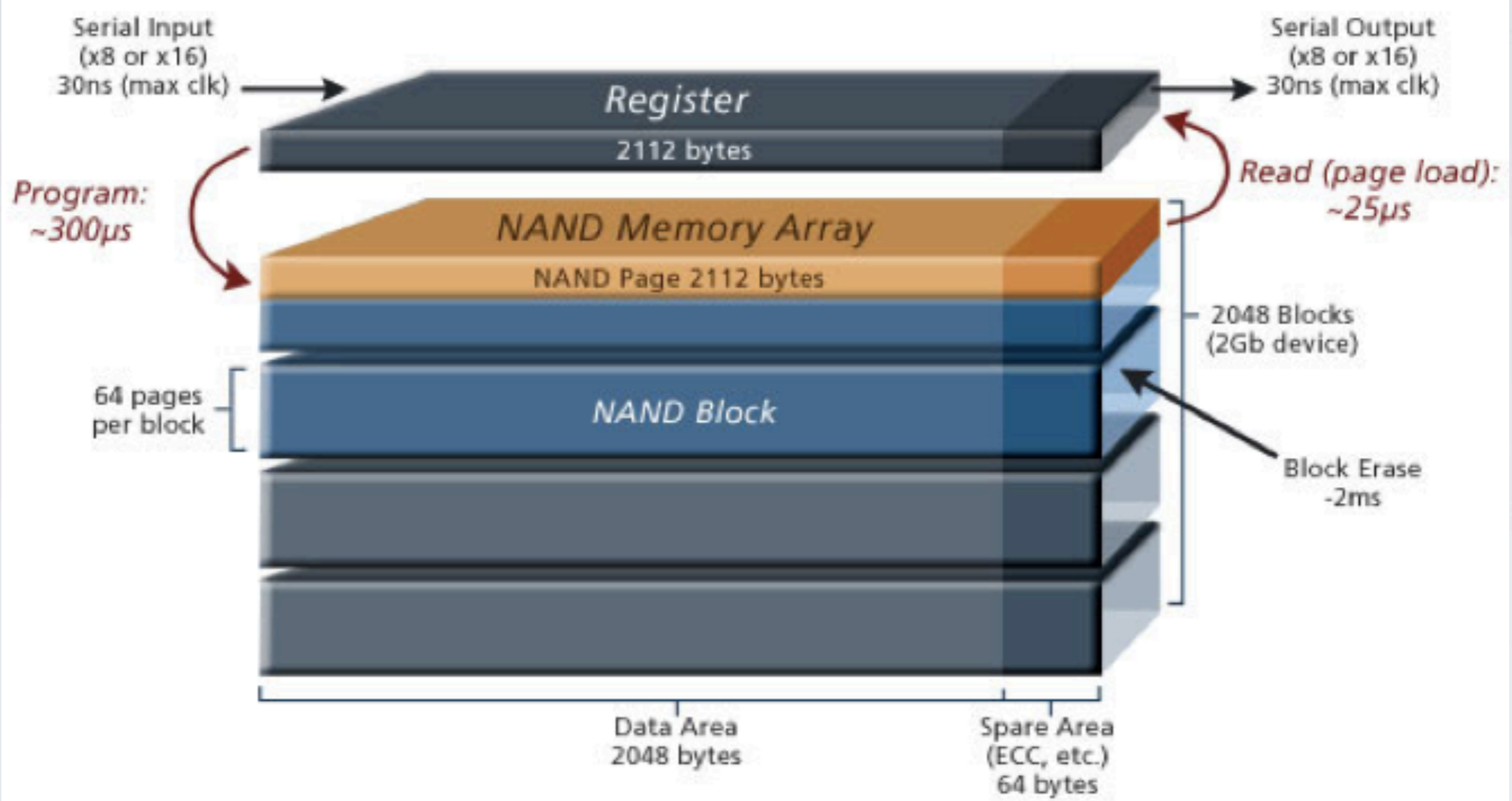


NAND VS NOR

- La Flash NOR est accessible directement bit à bit (aléatoirement), mais plus lentement
 - très lent : lecture en $12\mu\text{s}$, effacement à 750ms
 - utilisation comme extension parfaite de la RAM : possibilité de XIP (eXecute In Place)
 - haute consommation électrique
 - utilisation pour de petites données pérennes (zones de boot, de paramètres système rarement modifiés...)
 - beaucoup plus cher !
- La Flash NAND n'est accessible que par blocs, copiés dans la RAM à chaque accès
 - plus rapidement : $25\mu\text{s}$ de lecture d'une page (puis $0,03\mu\text{s}$ pour les pages du même bloc), $300\mu\text{s}$ d'écriture, 2ms pour effacer un bloc
 - ne peut au mieux exécuter directement que la première page, les autres devant être copiées en RAM (pas de XIP)
 - basse consommation électrique
 - un cycle de programmation = lecture du bloc en RAM, modification en RAM, effacement Flash, reprogrammation Flash, libération RAM

LA NAND

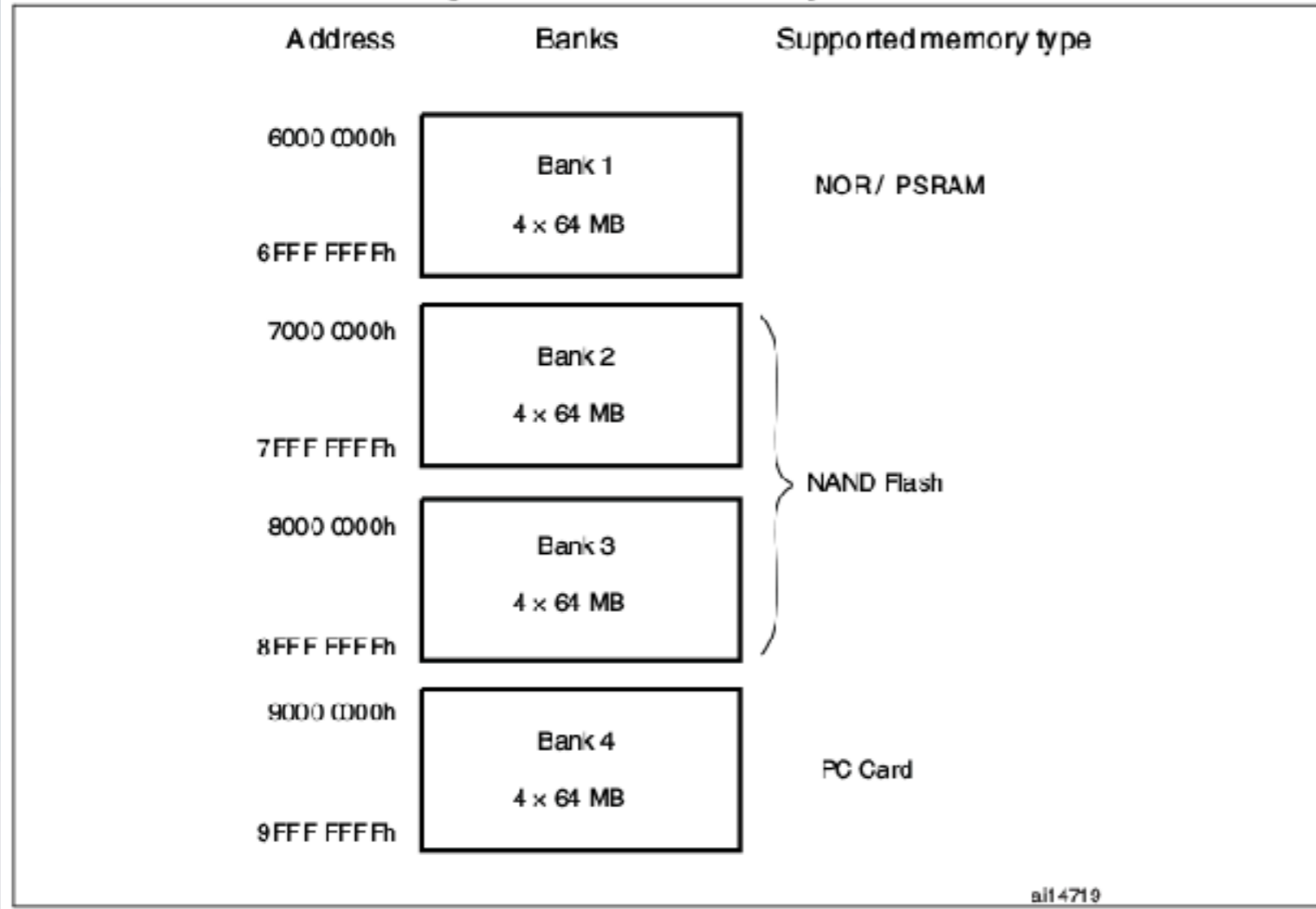
- NAND standard = Single-Level Cells (arrivée prochaine des MLC — 4 niveaux sur 2 bits par cellule —, beaucoup de problématiques à résoudre: plus d'espace mais moins fiables)
- Accès séquentiel par bus série
- Chaque page comporte aussi (en plus) d'autres données de redondance (spare area) dont l'Error Correction Code (ECC)
- Problématique de durée de vie : *wear leveling*, c'est-à-dire écrire de manière uniforme sur les blocs et non toujours sur les mêmes
- Peu chère, plus d'espace mémoire
- Largement utilisée (remplace même les systèmes avec NOR ou NOR+NAND plus onéreux, mais plus fiable — bootloader sur la NOR par exemple)



CONTRÔLEUR, SPI, BRANCHEMENT DIRECT

- La gestion des problématiques liées à la Flash peut se faire par contrôleur
 - bufferise les blocs
 - implémente les différents algorithmes (dont la gestion des mauvais blocs et le wear leveling) : *Flash Translation Layer* matériel et opaque (vu de l'OS, émulation de périphérique bloc tel un disque dur)
 - cas des clés USB, disques SSD, CompactFlash, carte SD, eMMC, etc.
 - « veuillez retirer en toute sécurité » : on ne sait pas où en sont les données, sauf à demander de toutes les synchroniser (commande Linux *sync*, et évidemment aussi en cas de démontage)
 - SENSIBLE AUX COUPURES DE COURANT, très dangereux en production, corruptions potentielles de données
- Sur un système embarqué, la gestion électronique de la Flash NAND ou NOR est directement adressée par le CPU
 - vue mémoire contiguë à la RAM (par *bank*)
 - gestion directe des entrées/sorties mais aussi des gestions d'erreurs/bad blocks : nécessite des algorithmes/pilotes spécifiques
 - beaucoup plus fiable : permet au système embarqué de gérer parfaitement les coupures de courant ! (Aucune corruption du système de fichiers : au pire, on perd les données du fichier en cours d'écriture/encore bufferisé en RAM)
- Une EEPROM est branchée sur un bus SPI du CPU (effacement complet + réécriture complète)

Figure 432. FSMC memory banks



http://www.chibios.org/dokuwiki/doku.php?id=chibios:community:plans:external_ram

Organisation de la mémoire sur S3C2440.

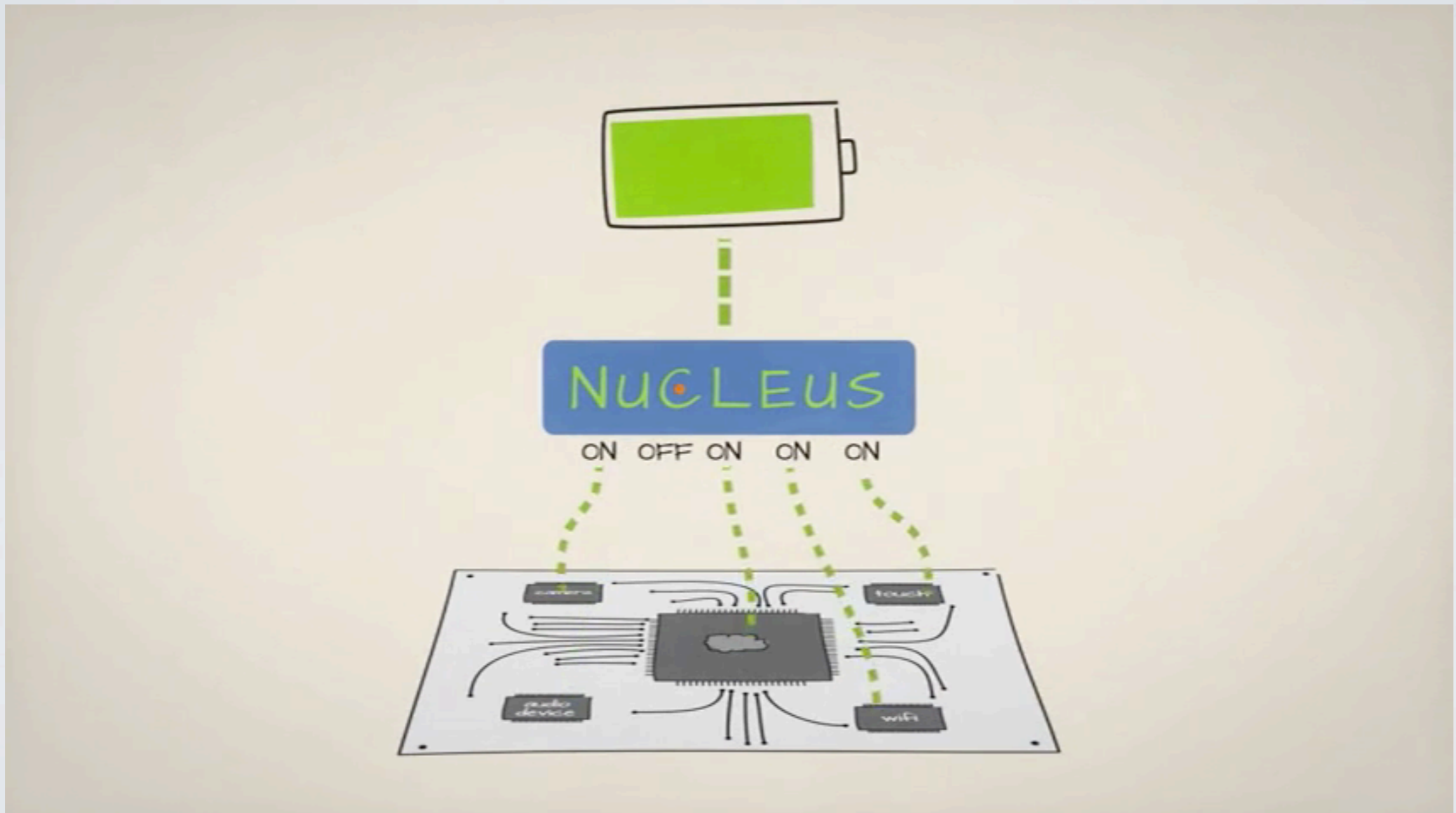
CHOIX DU SOFT

- En fonction du hard choisi
 - MCU
 - cas courant : micro noyau (RTOS, mais peu importe en réalité)
 - si beaucoup de RAM et Flash (> 2Mo chacun) : μ CLinux ?
 - CPU
 - cas courant : OS complet
 - si peu de RAM/Flash (< 4Mo chacun) : μ CLinux ou autre micronoyau (MMU = id)
- Micro noyau RTOS
 - (très) faible consommation électrique, très faible empreinte mémoire, support connectivité : possibilité de batterie avec forte autonomie
 - mais DIY, usage déterminé (problématique d'évolutivité du produit)
- OS complet
 - consommation électrique plus élevée = secteur ou recharge de batterie type Li-ion plus fréquente !
 - parfois astuces de type mise en veille profonde & réveil sur interruption (type livre électronique) = beaucoup plus forte autonomie

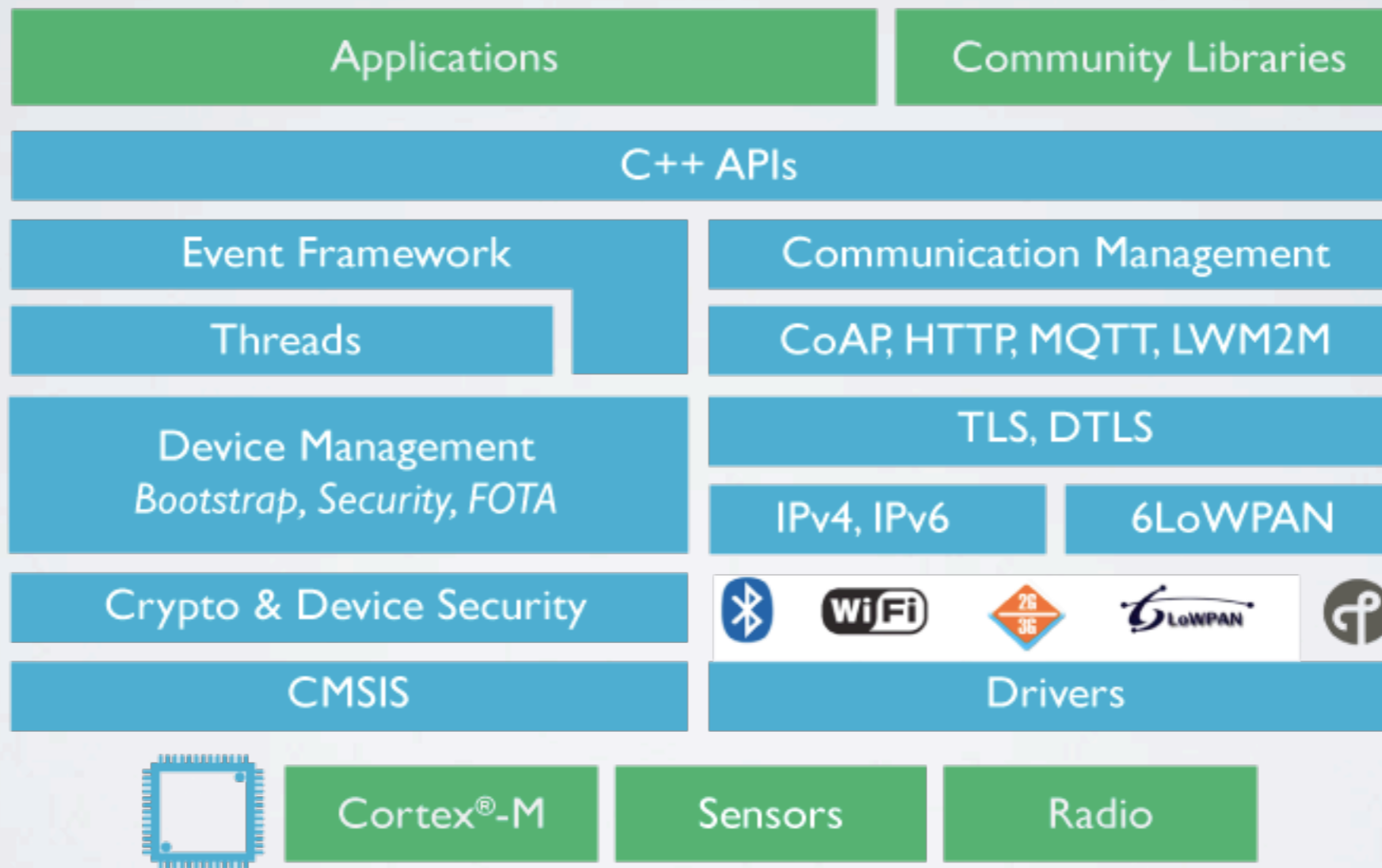
MICRO NOYAU TEMPS-RÉEL

- mbed (<https://www.mbed.org/>)
 - nouveau venu (sortie stable prévue pour novembre 2015), mais installé par défaut sur beaucoup de devices/devices « mbed ready »
 - maintenu par ARM (mais manifestement fermé)
- Nucleus (<http://www.mentor.com/embedded-software/nucleus/>)
 - le plus ancien, sorte de standard de fait (utilisé dans tous les téléphones portables ! Pile GSM)
 - propriétaire MAIS sources disponibles (Mentor Graphics)
 - outils de développement avancés disponibles
- Lepton (open source, <https://github.com/lepton-distribution>)
 - récent mais 250k lignes de code, français
 - 130KB flash mini, 16KB RAM
 - OpenSource, greffons
- Micrium (<http://micrium.com/iot/overview/>)
 - l'un des noyaux propriétaires RTOS les plus déployés
- FreeRTOS (<http://www.freertos.org/>)
 - libre (édité par une société)
 - très utilisé, notamment dans tous les produits Withings
 - facile à prendre en main, mais parfois des manques (exemple : pile USB difficilement disponible)
- Contiki (<http://www.contiki-os.org/>)
 - libre (édité par une société)
 - moins utilisé que FreeRTOS, mais tout de même bien implémenté dans le paysage
 - mais gère 6lowPan
- Zephyr (www.zephyrproject.org)
 - Linux Foundation (février 2016), suite Rocket (Wind River)
 - Apache 2.0

NUCLEUS

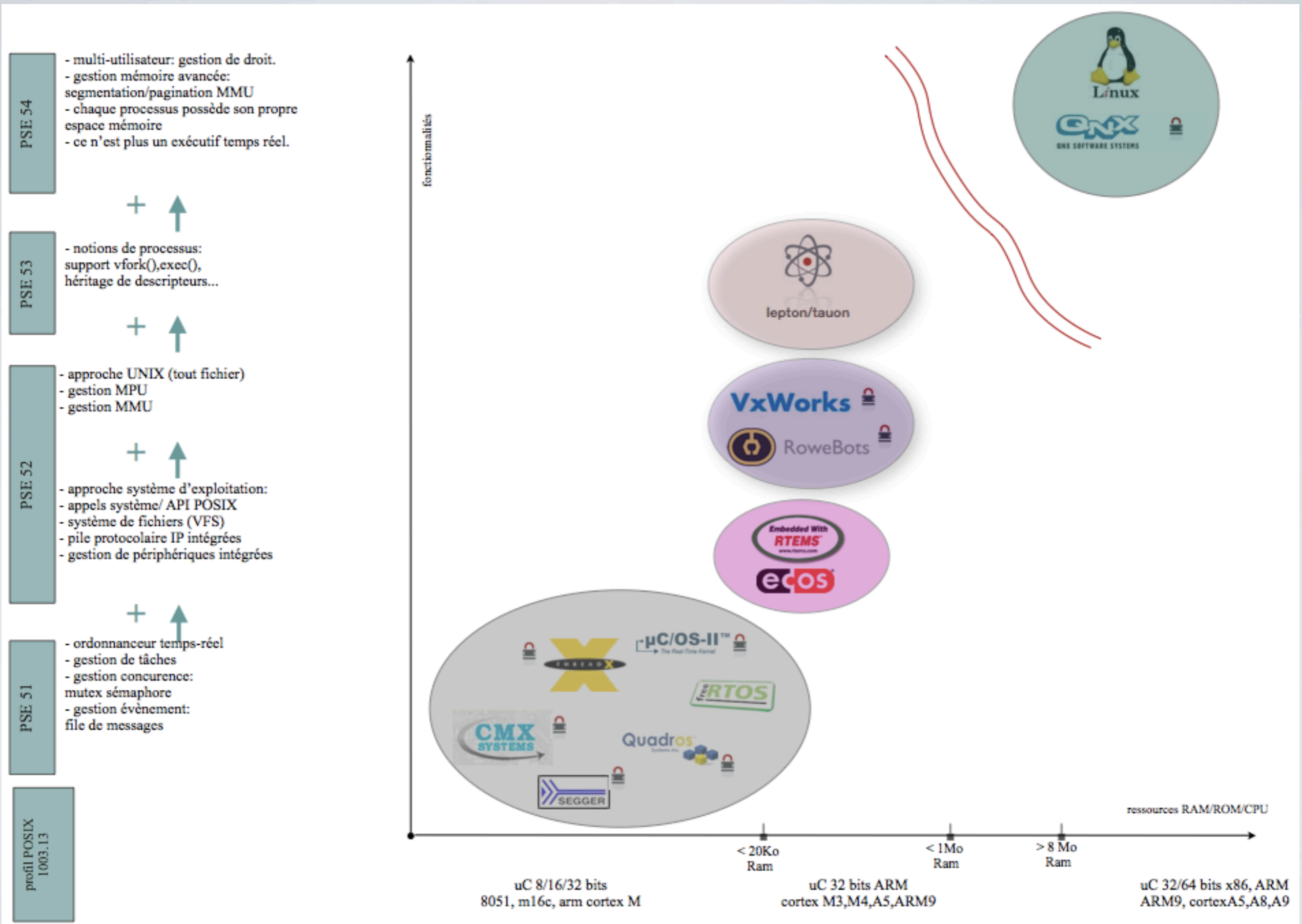


MBED



OS COMPLET

- Linux
 - libre, extrêmement répandu, large base existante, devenu le choix par défaut dans les projets industriels, UNIX-like
 - gros impact récent de Yocto comme solution de génération automatisée
 - excellents outils de communication : l'origine même de Linux !
 - possibilité d'applications, large spectre de pilotes matériels
 - homogénéité de la solution : sur le device, sur le relai/répétiteur/gateway, sur le serveur central, sur le cluster de calcul big data
 - cf cours vendu séparément
- QNX
 - UNIX-like, très belle architecture (micro-noyau avec pilotes dans l'espace utilisateur !), mais fermé
 - plus vraiment
- Windows
 - fermé, consommateur de ressource, mal fichu, abandonné de tous
 - en fait, non.



(crédit : o l Oée, Philippe le Boulanger)



CHOIX DE LA CONNECTIVITÉ

ENVOI/RÉCEPTION DE DONNÉES

- L'objet connecté envoie des données, par définition
 - capteur(s) : accéléromètre, gyroscope, etc.
 - envoi en live (pas de décharge manuelle avec de l'USB ou en sortant une carte SD par exemple)
 - poids des données variable, mais souvent faible
 - potentiellement, reçoit des données (par exemple : détection de clés, de lunettes, géolocalisation de vélo...)
- L'envoi de données nécessite un transport physique, un protocole de transport logiciel, une optimisation pour tenir compte des problèmes de connectivités et d'alimentation (sur batterie le plus souvent)
- L'loT hérite de l'état de l'art du M2M

ÉTAT RÉEL DU M2M

- Diversité... ou bazar ?
 - matériel
 - influence alimentation de l'appareil connecté : autonome (sur batterie) ou par secteur
 - diversité de la communication radio : WiFi, Bluetooth, Bluetooth 4.0 (low energy), NFC/RFID, GPRS/3G/4G (LTE), 802.15.4, SigFox, LoRa
 - ou courant porteur (PLC : G3-PLC, HomePlug Green PHY, ...)
 - généralement le but est d'éviter les connexions filaires dédiées et encombrantes (Ethernet, CAN, RS485...)
 - protocoles
 - communication radio (IoT) : ZigBee, 6LoWPan, EnOcean, ZWave, Meter-Bus, ...
 - applicatifs : quelles API ? *Qui parle comment à qui ?*

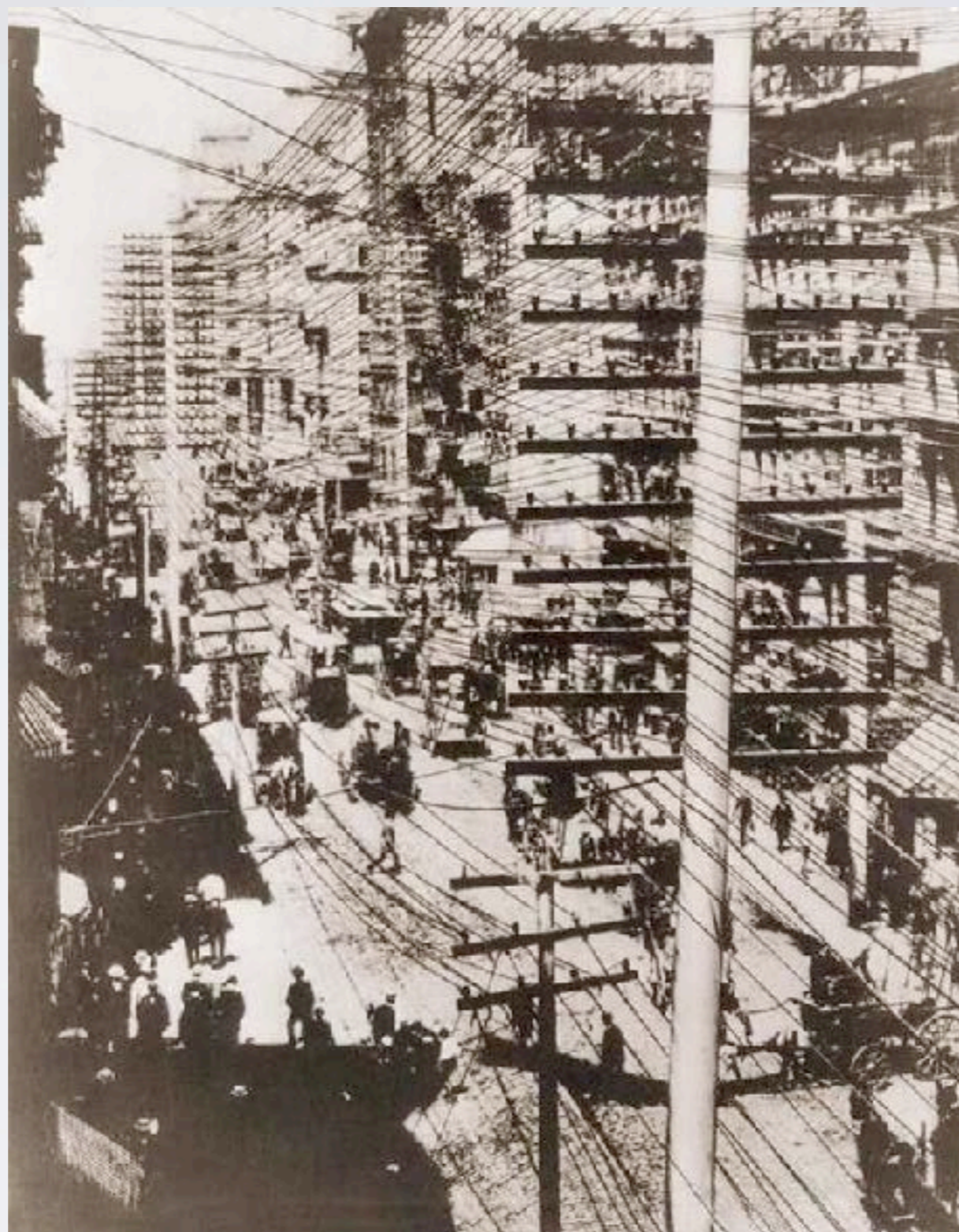
ÉTAT DE LA NATION M2M



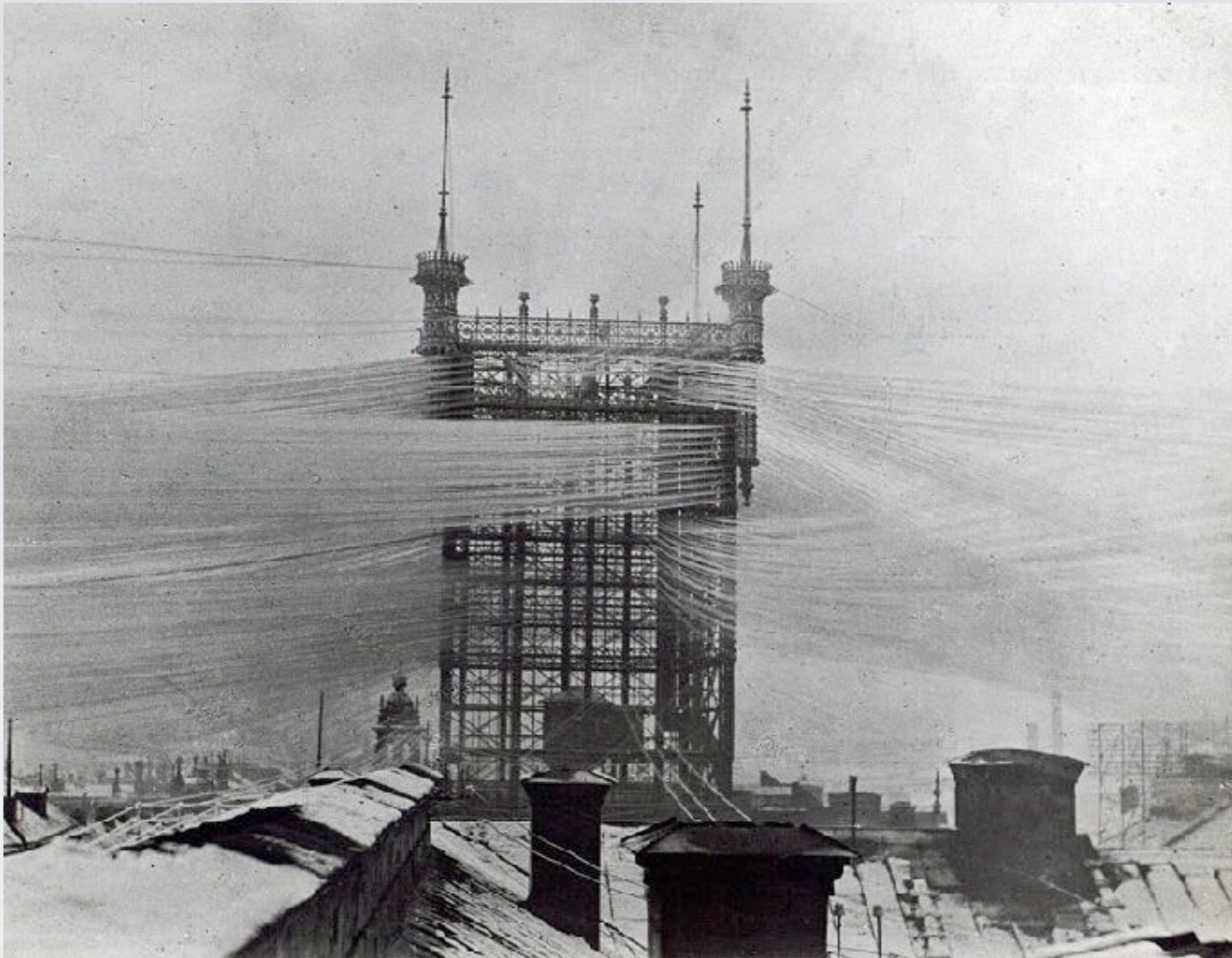
Besoin d'unification : qui sera Qin Shihuangdi ?

INTEROPÉRABILITÉ & BUS LOGICIELS

- Efforts de standardisation
 - Commission européenne :
<http://ec.europa.eu/digital-agenda/en/news/conclusions-internet-things-public-consultation>
 - Internet of Things Architecture :
<http://www.iot-a.eu/public>
 - OneM2M : <http://www.onem2m.org/>
 - Eclipse foundation : <http://iot.eclipse.org/protocols.html>
 - et encore, : MQTT, OCPP, OSGi, etc.
 - ou plus simplement, (vieux) standards plus industriels/bas niveau spécialisés : ARINC 429 (avionique), CAN (automobile), Modbus (automates programmables), Profibus (bus de terrain propriétaire), KNX (bâtiments)



telephone wires in Manhattan, 1887



tour téléphonique de Stockholm, 1890

RAPPEL : 7 COUCHES OSI

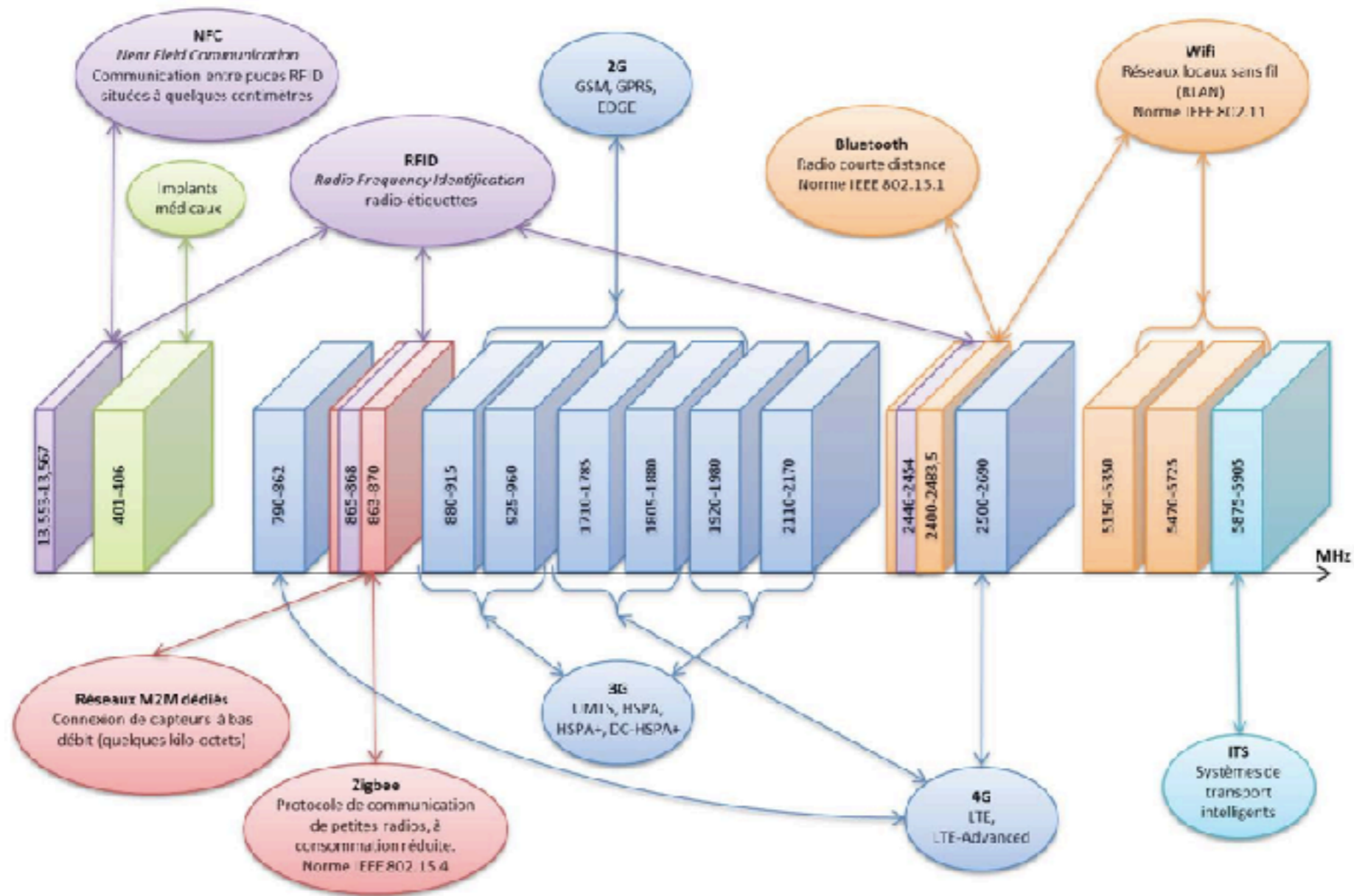
Application	interface vers les services réseaux	HTTP, FTP, SMTP, SSH, telnet
Présentation	mise en forme des données	HTML, CSS, GIF
Session	gestion des connexions	RPC, PAP, SSL, SQL
Transport	vérification des données	TCP, UDP, NETBEUI
Réseau	routage des données	ARP, IPv4, IPv6, IPsec, ICMP
Liaison	transmission des trames	PPP, IEEE 802.2, L2TP, MAC, Ethernet, I ² C, CAN, Wi-Fi
Physique	transmission de bits sur le support physique	DSL, USB, ISDN, 10BASE-TX, Bluetooth

CHOIX DE LA CONNECTIVITÉ MATÉRIELLE : RADIO

- énormément de choix : WiFi (norme IEEE 802.11x), Bluetooth BLE (norme 802.15.1), implémentations de 802.15.4 (Zigbee, 6LowPan), RFID/NFC, Z-Wave, EnOcean, MyriaNed, Insteon, GPRS/3G/4G LTE, SigFox, LoRa, DASH7, M-Bus, OCARI, RuBee (IEEE 1902.1), Wavenis, WirelessHART, ISA100, etc.
- dépend de plusieurs facteurs
 - portée désirée
 - taille des données à transmettre, vitesse d'établissement de la connexion
 - consommation électrique, besoin de mémoire : à mettre en relation avec la vitesse de connexion et le fait que l'objet dort 99% du temps (métrique de référence : une batterie pour petits objets contient 300mAh à 3000mAh)
 - robustesse & sécurité : sensibilité ou brouillage, chiffrement, appairage vs man-in-the-middle
 - facilité d'utilisation et de mise en place
 - fréquence utilisée => propagation à travers l'eau, le béton...
 - usage de destination spécialisé du protocole (domotique, compteur électrique, bâtiment, contrôle industriel, outdoor ou indoor, etc.)
 - écosystème existant
- <http://www.oezratty.net/wordpress/2015/reseaux-m2m-1/>
<http://www.oezratty.net/wordpress/2015/reseaux-m2m-2/>

	mémoire	autonomie sur piles	nombre de noeuds	vitesse de transfert	portée	sécurité	facilité/ écosystème
WiFi	1 Mo +	jours (50/100mAh)	256+	11(a)-54(b)-108(g)-320(n)-1000(ac) Mb/s	100 m	faible (défaut) à forte (WPA2)	non/oui
Bluetooth	250 ko +	mois (30 mA)	255	1 Mb/s	10 m (qqm mètres à 100m)	pairage + chiffrement	moyen/moyen
BLE		1-2 ans batterie 1000mAh (15mA)		~270 Kb/s	10-30m	+ AES 128bits	oui/high-end consumers
NFC/RFID		energy harvesting 15-50 mA	2	106,212 ou 424 kbit/s	RFID : 10 à 200m NFC : 10cm (pas normalisé)	pairage en proximité	oui/moyen
Zigbee	4-32 ko	années (15 mA)	65 000+	20-250 kb/s	100 m	médiocre	oui/moyen
6lowPan		années		250 kbit/s		forte (AES + IPsec + SSL)	oui/non
Z-Wave (Sigma)		années	232	20-250 kb/s 9 à 40 kbps	300 m mesh, sinon ~50 m	très faible	non/plutôt non
EnOcean		energy harvesting		125 kbit/s paquet 14 o	300 m outdoor 30 m indoor		oui/Siemens
GPRS, 3G, 4G LTE		heures	une centaine par BTS	1,9Mb/s (3G) à 1Gb/s(4G)	400m (urbain) à 10km (rural)	très, très forte	carte SIM + abo/excellent
SigFox		mois					en cours (couverture France + US)
LoRa		mois 60-100 mA		<1 kb/s	15 km théorique		en cours

LES FRÉQUENCES DE L'INTERNET DES OBJETS EN FRANCE*



Source: ANFR

* Schéma simplifié des principales bandes de fréquences utilisées pour l'Internet des objets

• <https://sanscontact.wordpress.com/2014/12/08/mettre-le-spectre-en-lumiere/>

• Les fréquences de l'internet des objets (c) ANFR



sensinode

Is the Internet Protocol enough?

Web
100s - 1000s of bytes



Internet

Inefficient content encoding

Huge overhead, difficult parsing

Requires full Internet devices



Web of Things
10s of bytes

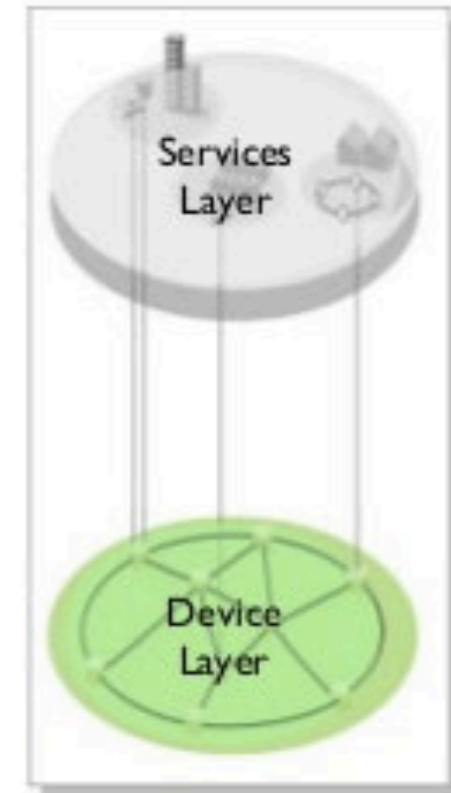
Efficient Objects

Efficient Web

Optimized IP access



Internet of Things



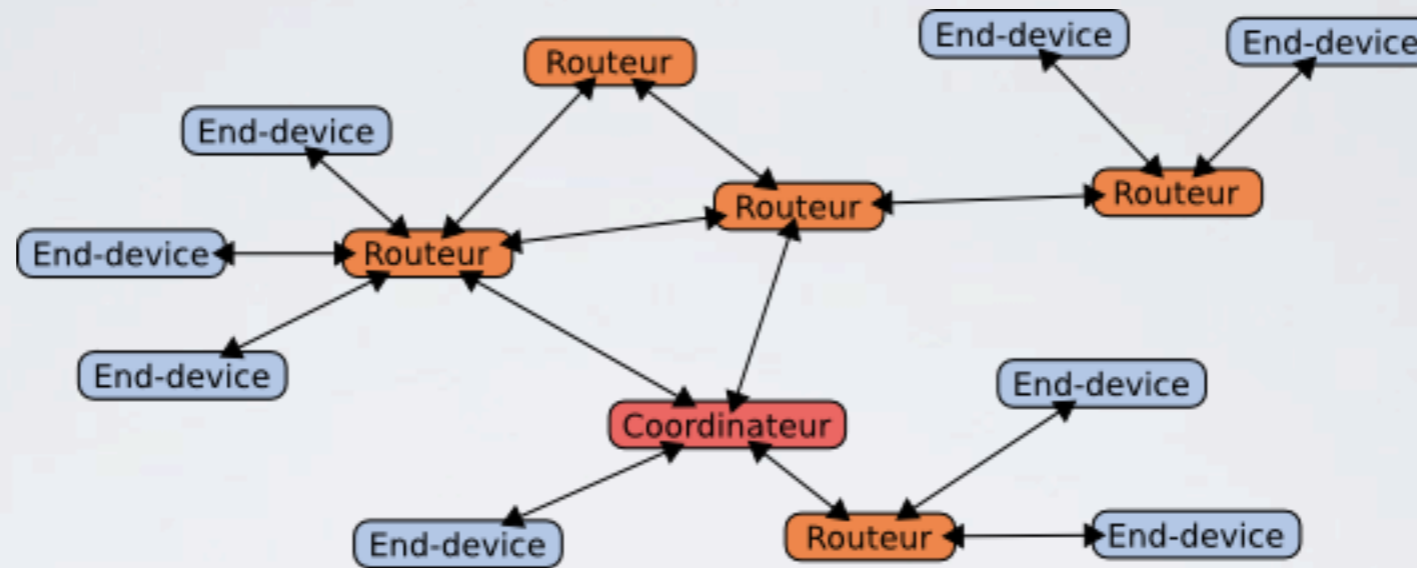
©Sensinode 2013

[Sensinode: - Zach Shelby: Is the Internet Protocol enough? \(Full Presentation\)](#)

802.15.4

- protocole de communication défini par l'IEEE destiné aux réseaux sans fil
- famille des LR WPAN (Low Rate Wireless Personal Area Network)
 - faible consommation
 - faible portée
 - faible débit des dispositifs utilisateurs
- organisation du réseau :
 - coordinateur : un seul par réseau, objet du réseau, au courant de tous les objets, chef d'orchestre indispensable
 - dispositif de fin (end-devices, objets à simple émission)
 - routeurs optionnels (objets), en plus du coordinateur
- <http://www.ieee802.org/15/pub/TG4.html>

DÉTAILS 802.15.4



- dispositif ayant toutes les fonctions possibles (FFD : Full Function Device) ou limitées (RFD : Reduced Function Device)
- topologie en étoile, maillée et en arborescence
- FFD peut communiquer avec tous, RFD ne peut communiquer qu'avec FFD (donc au moins un FFD, en coordinateur)
- 2 modes de fonctionnement de la couche MAC, selon le type de topologie utilisé et le besoin en débit garanti :
 - non-beacon : méthode CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance, utilisée aussi par WiFi)
 - beacon : envoi à période régulière d'une balise pour synchroniser les dispositifs, garantissant un débit au capteur ayant un Guaranteed Time Slot (GTS)

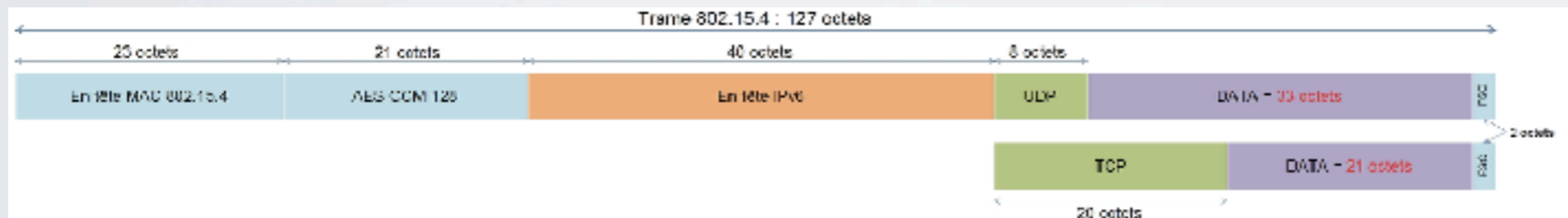
ZIGBEE



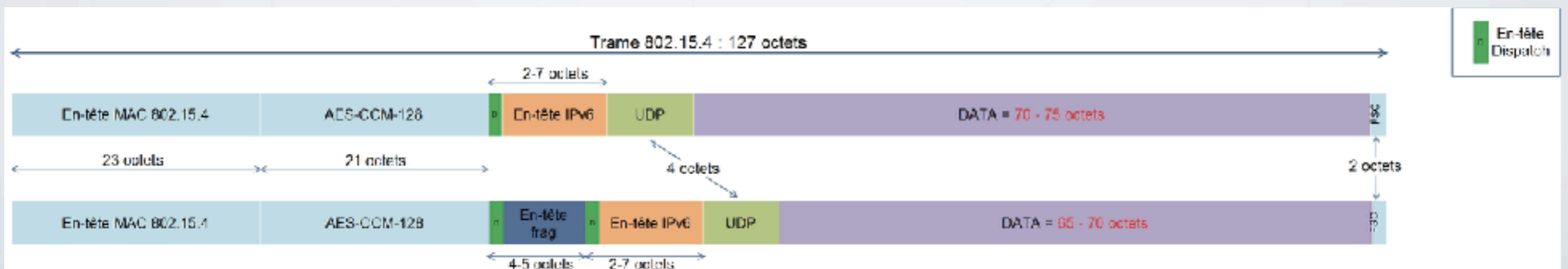
- protocole de haut niveau
- communication de petites radios, à consommation réduite
- implémentation de IEEE 802.15.4
- pour réseaux à dimension personnelle : Wireless Personal Area Networks : WPAN
- depuis 2004, référence assez commune
- simple à mettre en oeuvre, peu cher (~\$3)
- configuration du réseau maillé (mesh) automatique, en fonction de l'ajout ou de la suppression de nœuds
- problèmes de compatibilité
 - différents profils appelés "standards" par l'Alliance ZigBee (Home Automation, Remote Control, Smart Energy, Health Care, Retail Services ou 3D Sync, etc.), qui sont des piles protocolaires à l'implémentation incomplète
 - couches Zigbee implémentées dans le hard plus ou moins complètes et ne réussissant pas à communiquer
 - ambiguïtés entre les protocoles 802.15.4 (cf Xbee, qui parfois implémente du Zigbee, souvent non, parfois incomplet...)
- à la mise sous tension, recherche par l'objet d'un coordinateur avec lequel il va s'appairer : problème de sécurité potentiel

6LOWPAN

- IPv6 Low power Wireless Personal Area Networks ou IPv6 LoW Power wireless Area Networks
- IPv6 sur norme 802.15.4, par encapsulation et compression d'entêtes IPv6 (RFC 4919 & 4944)
- très complet : <https://fr.wikipedia.org/wiki/6LoWPAN>
- résoud beaucoup des problèmes de Zigbee (sécurité, découverte de voisins...), au prix d'une plus grande complexité
- utilisé aussi en PLC, sur RFID et Bluetooth, mais ne paraît pas encore tout à fait sec...



Problème d'intégration d'un paquet IPv6 sur une trame 802.15.4



Trame 802.15.4 sur IPv6 avec compression de l'entête IPv6 et entête de fragmentation.

NFC/RFID

- radio-identification (RFID, fin années 1960) : méthode pour mémoriser et récupérer des données à distance en utilisant des marqueurs (radio-étiquettes/RFID tag/RFID transponder), ie puce + antenne
 - pour objets, par exemple avec code barre (retail)
 - passeport, carte de transport, carte de paiement
 - animaux en sous-cutané
 - 30 milliards d'étiquettes RFID produites dans le monde, 1 milliard de transistors par être humain
- Near Field Communication = Communication en champ proche
 - extension de la norme ISO/CEI 14443 (RFID)
 - mode émulation de carte : l'appareil se comporte comme une carte (possibilité d'être alimenté par la boucle radio !)
 - mode lecteur ou actif : lecture de carte ou tag
 - mode pair-à-pair : échange d'information entre deux appareils NFC
- Usage du NFC pour appairer deux appareils sur un protocole de transmission plus rapide : Beam (Android ICS) avec Bluetooth, S-Beam (Samsung) avec Wi-Di (WiFi-Direct)
- 500+ millions NFC-equipped smartphones

802.15.1 BLUETOOTH/BLE

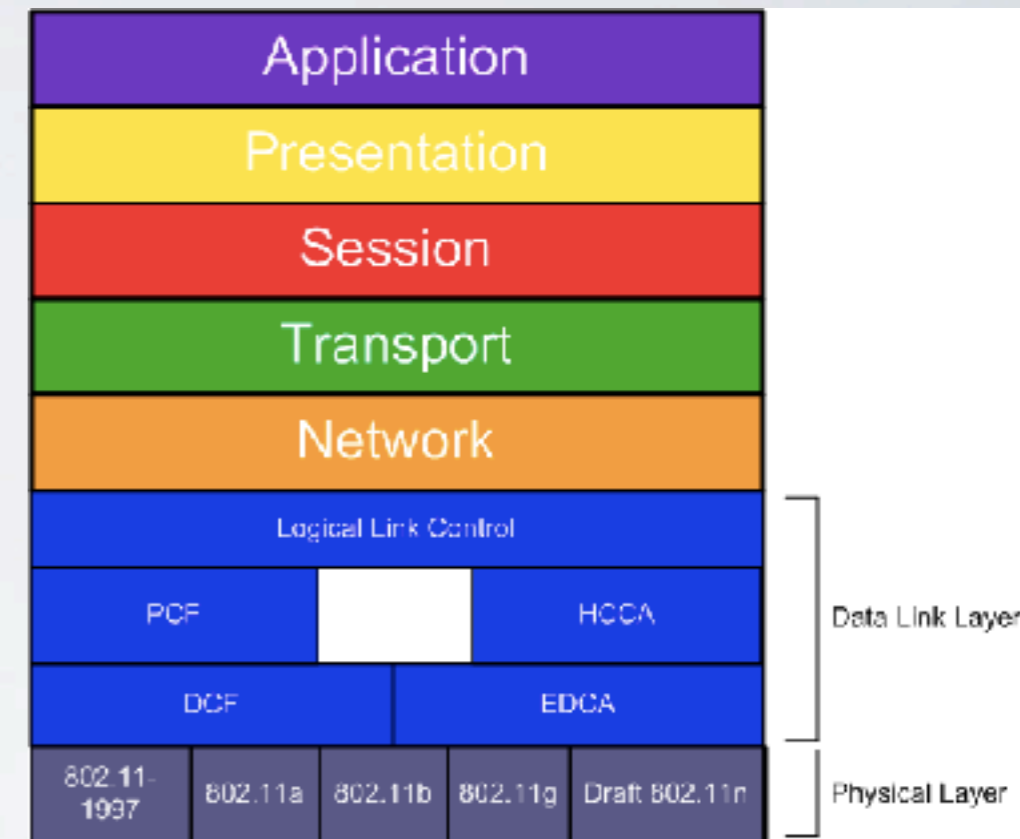
- Échange bidirectionnel de données à très courte distance et utilisant des ondes radio UHF (Ultra Hautes Fréquences), depuis 1994, pour unifier les communications téléphone portable-ordinateur puis autres appareils
- Topologie en étoile entre maîtres et esclaves (un esclave peut avoir plusieurs maîtres, mais les esclaves ne communiquent pas entre eux)
- Découverte et profilage des appareils (Service Discovery Protocol, complexe)
- Canaux voix et data :
 - RFCOMM (Radio Frequency Communication) : émule du RS232, peut faire passer de l'IP à 360 kbit/s maximum (ex : téléphone portable)
 - OBEX (OBject EXchange) : transfert d'objets binaires basé sur un protocole de l'IrDA (communication infra-rouge, à la manière de HTTP)
- Sécurité : authentification, pairage (association), création/modification de clés, chiffrement
<https://developer.bluetooth.org/TechnologyOverview/Pages/LE-Security.aspx>

BLUETOOTH/BLE DANS L'IOT

- Bluetooth v2 n'est pas souvent une bonne solution à cause de l'appairage, et de standards incompatibles (chez Apple notamment).
- Bluetooth 4 (BLE = Bluetooth Low Energy = Bluetooth Smart) est bien meilleur (mais incompatible !)
 - deux fois moins d'énergie pour la transmission (15mA), temps minimum de transmission 30 fois inférieur (3 ms vs 100 ms)
 - compatible avec les derniers smartphones
 - 10 à 30 mètres (100m en théorie) à 270KBps
 - prochaine évolution permettant des réseaux mesh (bonne alternative à Zigbee)

802.11 WLAN/Wi-Fi

- le plus répandu, depuis 1997
- plusieurs normes : legacy (1997), a (99), b (99), g (2003), n (2009), ac (2014)
- plusieurs amendements, dont une version 3.7 GHz pour 5km de portée
- portée usuelle entre 20 et 50 m, 100 m en extérieur
- taux de transfert : 200 à 450 Mbit/s pour 802.11n, 433 à 1300 Mbit/s pour 802.11ac (n et ac sont MIMO : [https://fr.wikipedia.org/wiki/MIMO_\(t%C3%A9l%C3%A9communications\)](https://fr.wikipedia.org/wiki/MIMO_(t%C3%A9l%C3%A9communications)))
- mode infrastructure : un ou plusieurs Access Point avec un SSID (Service Set Identifier) pour plusieurs machines partageant le même SSID
- mode Ad hoc : machines entre elles, sans AP, à condition d'avoir le même SSID (variante : Wi-Fi Direct, où les machines négocient qui est l'AP)
- mode Bridge : zones de couverture au même SSID reliées par Ethernet filaire
- mode Répéteur : extension de couverture par relais (divise la bande passante)
- https://fr.wikipedia.org/wiki/IEEE_802.11, <https://fr.wikipedia.org/wiki/Wi-Fi>



WiFi & IOT

- Espoir messianique
 - le WiFi est implanté partout, sur les box FAI, les téléphones, les routeurs, déployé dans les villes, les bâtiments (hot spots/ bornes WiFi)
 - tout le monde sait s'en servir (quoique... Reste de la pure magie pour 90% de la population)
 - hop, c'est plié !
- Oui, mais...
 - pas encore low energy, nécessite aussi pas mal de ressources en mémoire et CPU
 - couverture au domicile pas si extraordinaire : problème d'obstacles très atténuants pour le signal (répétiteurs via CPL ?)
 - compliqué à appairer : trouver le bon SSID et rentrer une clé WPA2
 - solution hardware comme le CC3000 de TI lorsqu'on n'a pas d'écran
 - impossible pour l'itinérant, possible pour le domotique ?
 - apparition du WPS : la solution ?

CC3100

- Chip capable de passer en mode AP sur appui de bouton : on s'y connecte, charge SSID et clé WPA2 via une interface (par exemple depuis un téléphone portable), puis reboot en mode client

- <http://www.ti.com/product/cc3100/description>

- Less Than 7KB of Code Memory and 700 B of RAM Memory Required for TCP Client Application

- Advanced Low-Power Modes

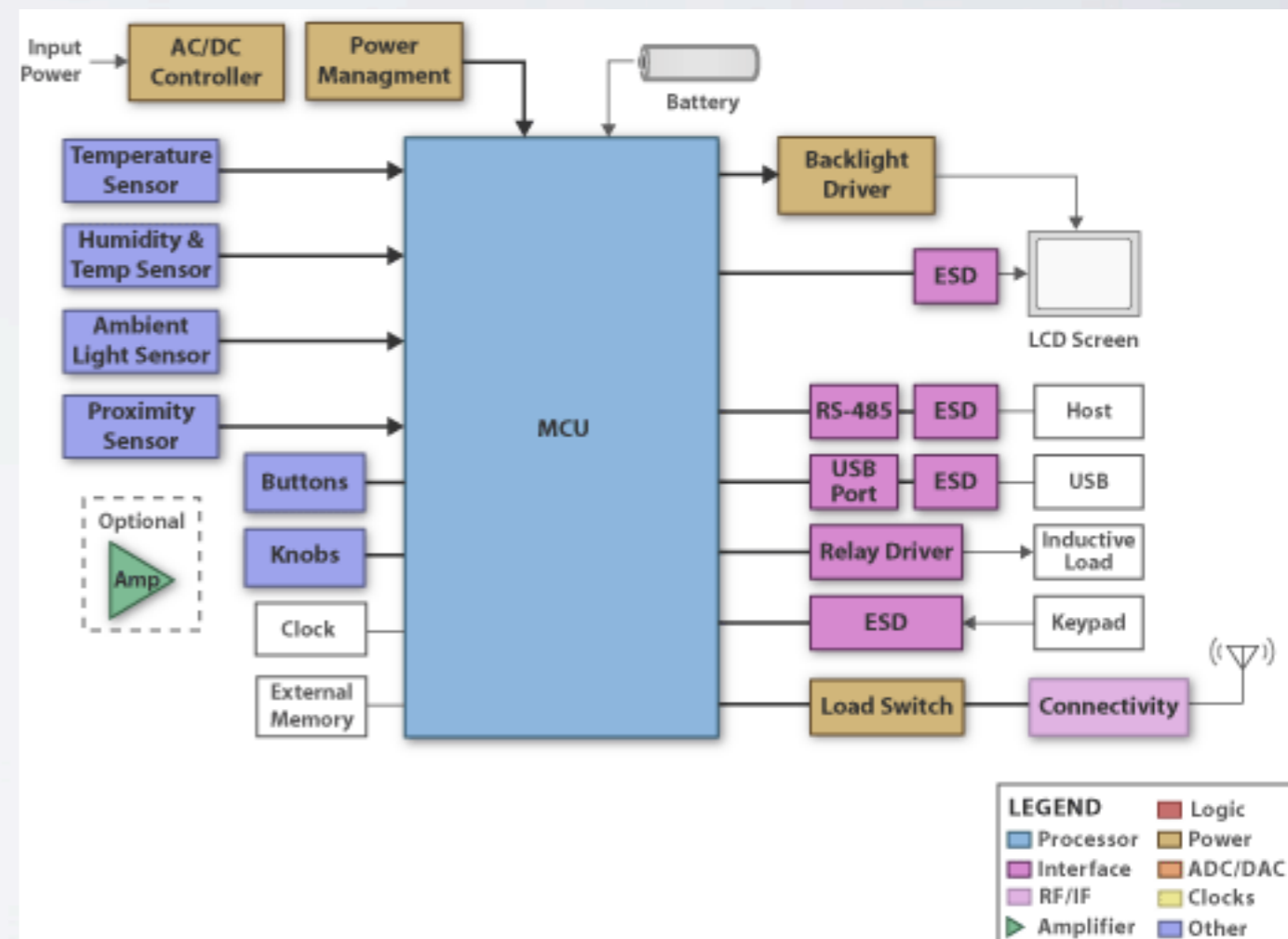
- Hibernate with RTC: 4 μ A

- Low-Power Deep Sleep (LPDS): 115 μ A

- RX Traffic (MCU Active): 53 mA @ 54 OFDM

- TX Traffic (MCU Active): 223 mA @ 54 OFDM, Maximum Power

- \$6.70 | 1ku



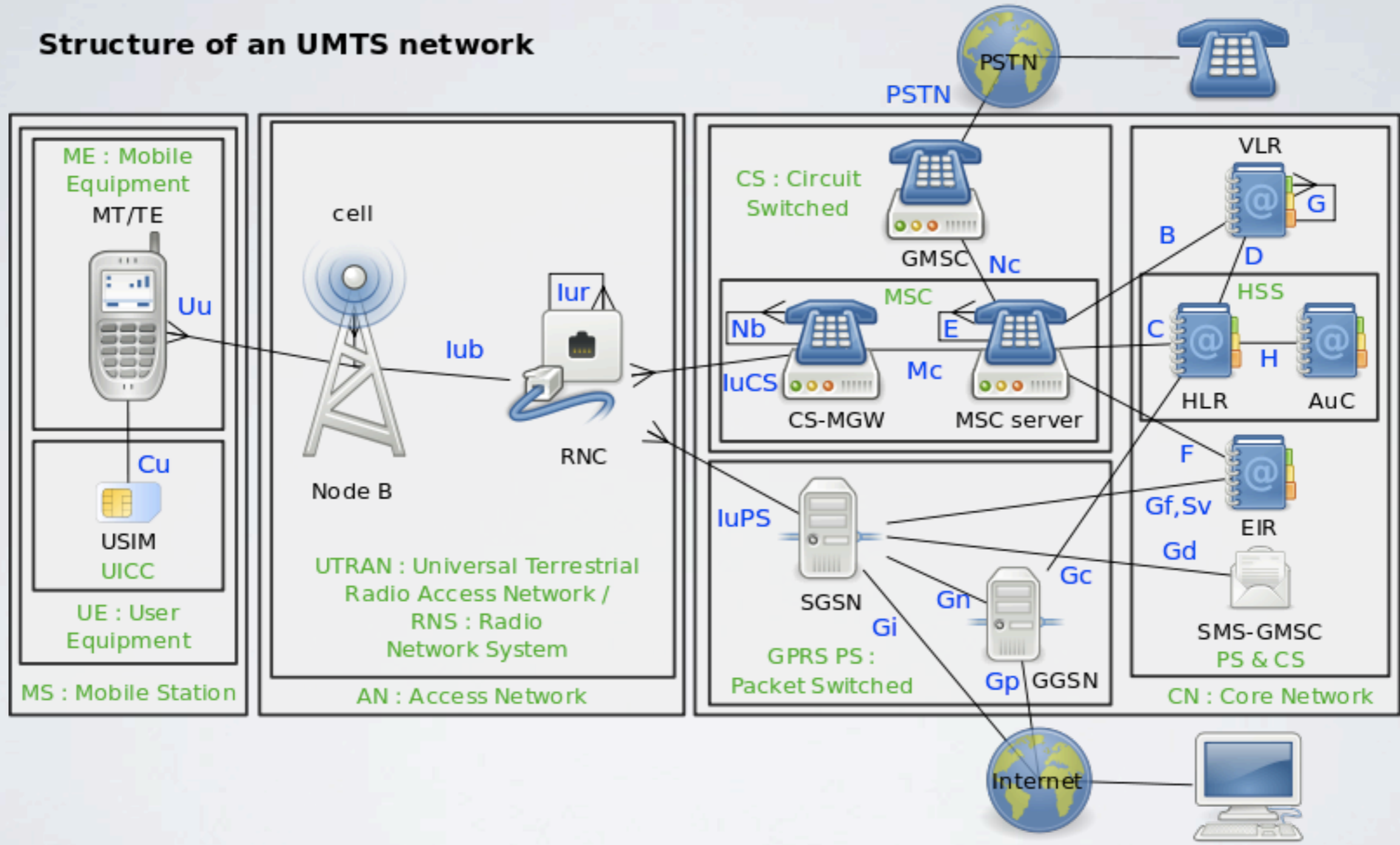
WPS À LA RESCOUSSE

- 4 nouvelles (2007) méthodes d'appairage sans avoir besoin de clavier ou d'écran sur l'appareil :
 - PIN (Personal Identification Number) : l'utilisateur lit un numéro sur une étiquette ou un écran du nouvel appareil et le reporte sur le point d'accès (AP) ou registrar (méthode avec grave faille de sécurité : <https://code.google.com/p/reaver-wps/>)
 - PBC : l'utilisateur presse un bouton (physique ou virtuel), à la fois sur le point d'accès et sur le nouvel appareil (plus moins accessible sur la box)
 - NFC : l'utilisateur approche le nouvel appareil du point d'accès pour établir une connexion NFC entre eux (en espérant que le NFC ne porte pas trop et que quelqu'un en profite)
 - USB : l'utilisateur se sert d'une clé USB pour transférer les données entre le nouvel appareil et le point d'accès (utilisé par D-link pour la plupart des clés USB WiFi ; nécessite une adaptation mécanique)
- Sécurité plus précaire, mais bien plus grande facilité
- On peut parier sur la solution NFC, à terme, mais pour le moment, les box de FAI n'implémentent au mieux que la méthode PBC (Livebox 2 par exemple)

TÉLÉPHONIE : 2G, 3G, 4G LTE

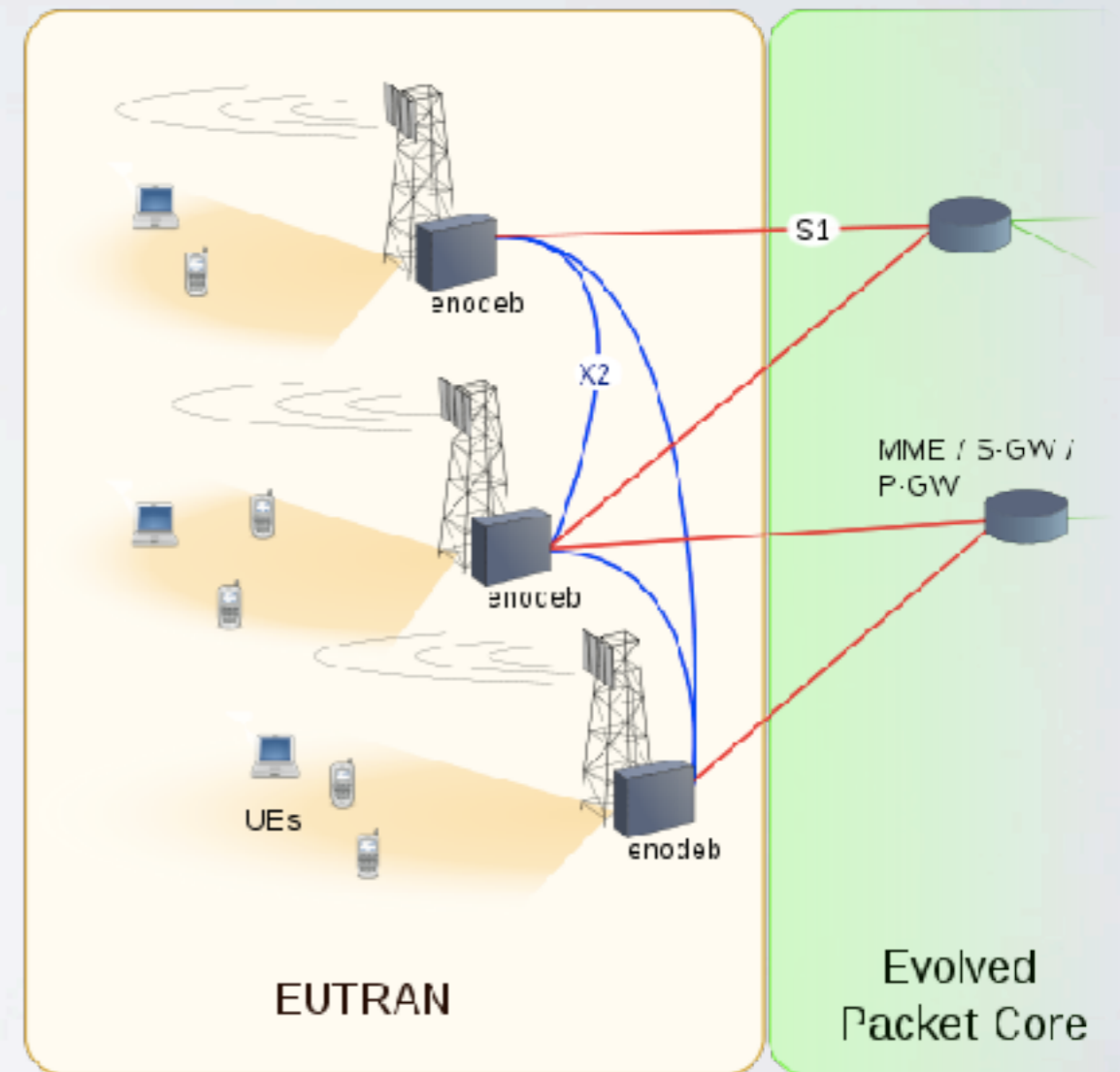
- Différentes normes successives, qui s'accumulent en mode QoS :
 - 2G (2e génération) = GSM (Global System for Mobile Communications, 1987-1991) : pas de data (9,05 kbps de voix)
 - 2,5G = GPRS (General Packet Radio Service, 2000-2005) : théorique max 171,2 kbit/s, débit max pratique 50 kbit/s, débit usuel 17,9 kbit/s (2 ko/s)
 - 2,75G = EDGE (Enhanced Data Rates for GSM Evolution, 2005) : 384 kbps théorique, 64 kbps pratique
 - 3G ou UMTS (Universal Mobile Telecommunications System, 2000-2007) : 1,9 Mbps dans des conditions idéales théoriques, 144 kbps zone rurale, 384 kbps zone urbaine
 - 3G+ = HSPA/HSPA+ (High Speed Packet Access+, 2008-2012)/H+ dual carrier : 42 Mbit/s down/11,5 Mbit/s up, 4 fois moins en pratique (10 Mbps)
 - 4G (3.9G) = LTE (Long Term Evolution, 2010-2013), : 150 Mbps théorique, 40 Mbps en pratique
 - 4G+ = LTE-Advance (normé en 2011, en cours de déploiement) : 1 Gbps à l'arrêt
 - 5G : en cours de normage, 10Gps
- MAIS : nécessite une carte SIM, un abonnement payant, soumis aux problèmes de QoS/couverture par les BTS, encombrement...
- NE JAMAIS UTILISER LES SMS POUR DES DONNÉES : toujours utiliser un canal de données, seul dont la latence est sûre

Structure of an UMTS network



LTE

- [https://fr.wikipedia.org/wiki/LTE_\(r%C3%A9seaux_mobiles\)](https://fr.wikipedia.org/wiki/LTE_(r%C3%A9seaux_mobiles))
- débit montant théorique allant jusqu'à 86,4 Mbit/s crête (75 Mbit/s utiles)
- temps de latence RTT (Round Trip Time) proche de 10 ms (contre 70 à 200 ms en HSPA et UMTS)
- Évolution poussée par Qualcomm pour l'IoT : LTE MTC (Machine Type Communication)



4G LTE MTC



- <https://www.youtube.com/watch?v=U-rtuwHleyU>
- <https://www.qualcomm.com/invention/technologies/lte/mtc>
- projet !

WIMAX

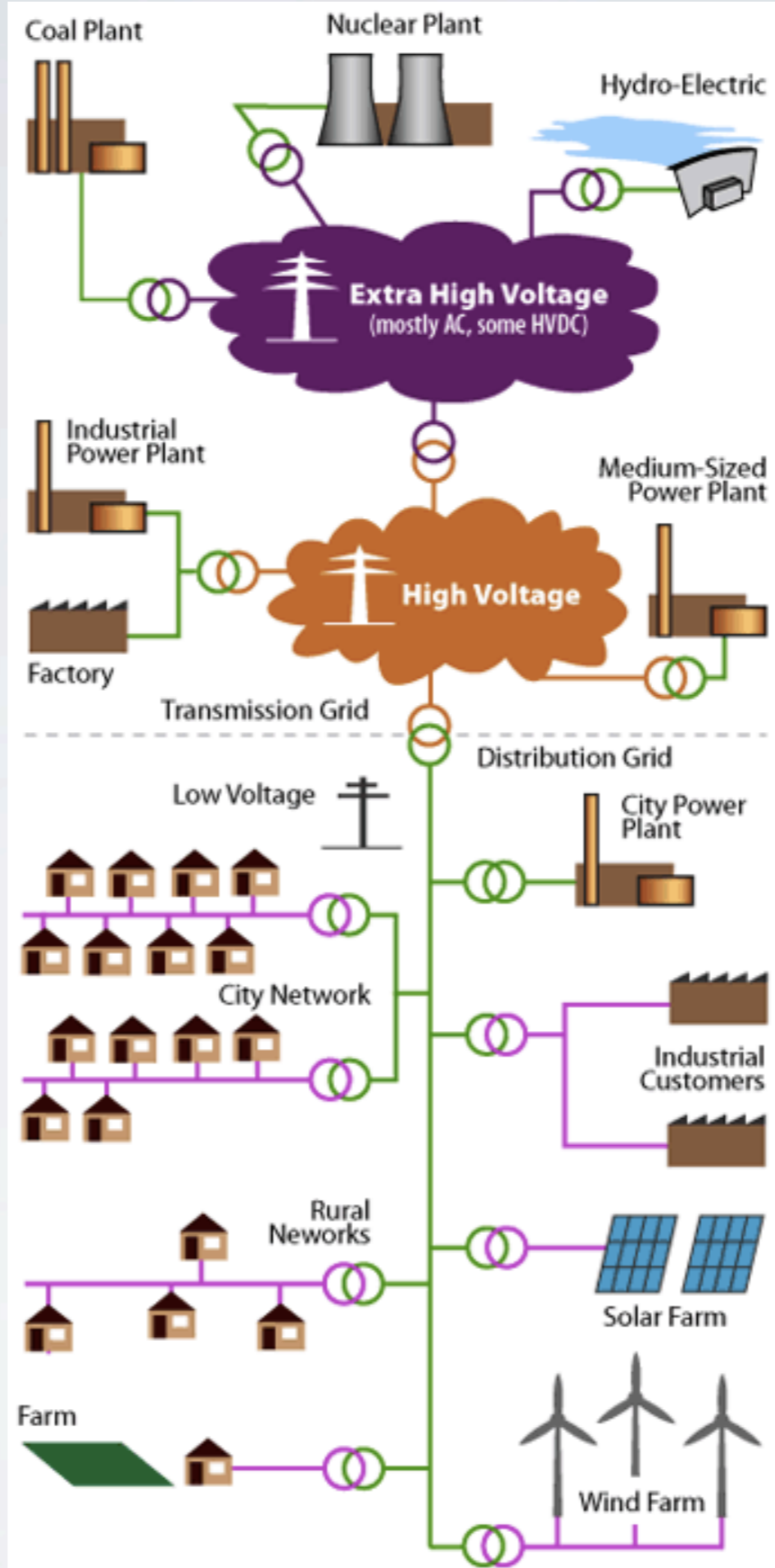
- IEEE 802.16
- 70 à 240 Mb/s
- 50 km sous visibilité directe, 1 à 7 km sinon
- volonté d'interopérabilité, gamme de fréquences de 2 à 66 GHz : convergence du Wi-Fi et de l'UMTS (3G)
- WiMax mobile 802.16e
 - connexion à haut débit en situation de mobilité (2006), 30 à 46 Mbps
 - dépassé par LTE pour la norme 4G, performances bien en deça
- Installations pour le dernier kilomètre des FAI (par ex en secteur rural où l'ADSL coûte trop cher) : échec cuisant de Bolloré en France (débit et connectivité très insatisfaisants)
- protocole zombie faute d'avoir pu trouver son réel positionnement et d'avoir trop tardé ?

M-BUS, OCARI, ETHERCAT, KNX

- Meter-Bus : standard européen (EN 13757-2 physical and link layer, EN 13757-3 application layer) de lecture distante des données de compteurs électriques ou gaz
 - aussi disponible pour d'autres types de compteurs
 - communication sur 2 fils, plus variante radio Wireless M-Bus (EN 13757-4)
 - couches physique OSI (data link par IEC 60870, couche réseau optionnelle, application par EN 1434-3, les autres inutilisées)
- OCARI (Optimisation des Communications Ad hoc pour les Réseaux Industriels)
 - capteurs sans fils industriels
 - consortium par EDF, DCNS, Telit, LATTIS, LIMOS, INRIA, LRI
 - Télédosimétrie en centrale nucléaire, supervision temps-réel de la radioprotection, détection d'incendie., surveillance de machines pour la maintenance prédictive, instrumentation mobile pour les essais en chantier, contrôle commande en boucle ouverte
 - <https://fr.wikipedia.org/wiki/OCARI>
- EtherCAT : pour les réseaux industriels, vieille norme toujours assez largement utilisée (mais bien peu documentée...)
- KNX : utilisé dans les bâtiments pour de la domotique « industrielle »

CHOIX DE LA CONNECTIVITÉ MATÉRIELLE : COURANT PORTEUR EN LIGNE

- PLC (Power Line Communication) = CPL (Courant Porteur en Ligne)
- X10
 - le plus ancien (1970) & répandu (usage domotique), existe aussi en radio
 - très insuffisant, perte de données, pas de sécurité...
 - <https://en.wikipedia.org/wiki/X10> (industry standard)
- Insteon : pour la domotique (<https://en.wikipedia.org/wiki/Insteon>)
- HomePlug Green PHY : pics de débit à 10 Mbit/s, pour smart grid, compteurs intelligents, thermostats, domotique, recharge de véhicules électriques
<https://en.wikipedia.org/wiki/HomePlug>
- G3-PLC (2012) : mêmes domaines (utilisé pour Linky), standard ouvert, supporte IPv6/6LoWPAN
<http://www.g3-plc.com/>
- ou plus simplement : POE (Power over Ethernet), dispense du 12V/15W (ou plus si POE+) sur l'Ethernet
https://en.wikipedia.org/wiki/Power_over_Ethernet



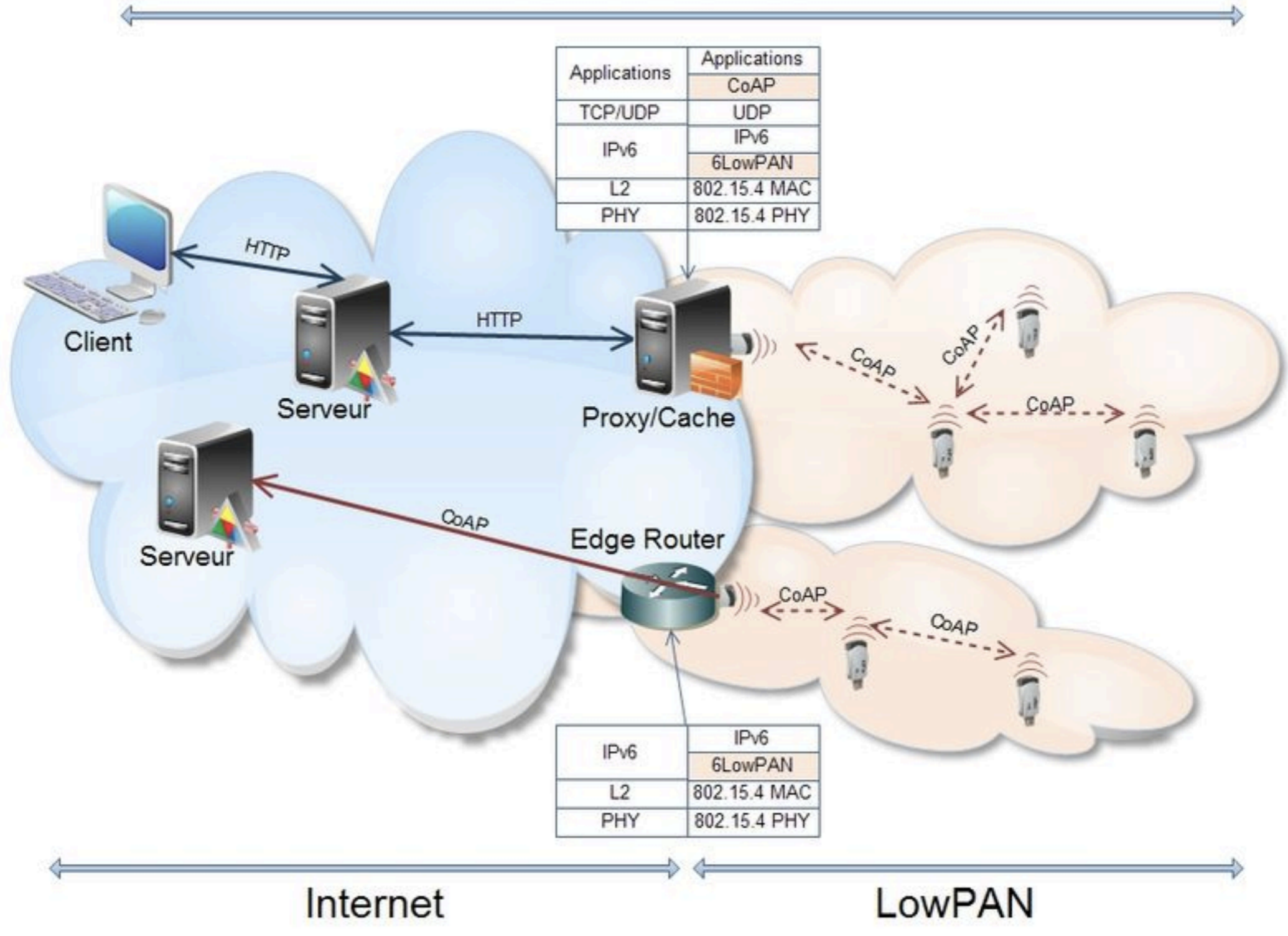
CHOIX DE LA CONNECTIVITÉ LOGICIELLE (PROTOCOLE) 1/2

- Synthèse complète des très nombreux protocoles logiciels : <http://postscapes.com/internet-of-things-protocols>
- MQTT (Message Queuing Telemetry Transport) :
 - protocole de transport extrêmement léger
 - publish/subscribe, sur TCP/IP
 - open source & standardisé, éprouvé
 - <http://mqtt.org/>
- CoAP (Constrained Application Protocol) :
 - normalisé par CoRE (Constrained RESTful Environments) de l'IETF (Internet Engineering Task Force)
 - pour capteurs contraints en ressources et énergie, connectés sur des réseaux peu fiables avec grand nombre d'appareils
 - sur UDP mais aussi interfaçable avec HTTP, compatible 6LoWPAN
 - https://en.wikipedia.org/wiki/Constrained_Application_Protocol

CHOIX DE LA CONNECTIVITÉ LOGICIELLE (PROTOCOLE) 2/2

- XMPP (Extensible Messaging and Presence Protocol) :
 - base de Jabber, très réutilisable pour la communication entre de très nombreux clients, aussi utilisé pour les machines à laver, sèche-linges, réfrigérateurs, etc.
 - mais considère que la connexion est du TCP persistant et peu efficace en binaire, donc peu pratique sur les réseaux peu fiables (LLNs : Low-power and Lossy Networks)
 - cependant, nouvelles normes [XEP-0322](#), [XEP-323](#) et [XEP-324](#) pour l'IoT
- RESTful HTTP over TCP :
 - particulièrement attractif pour connecter les appareils des utilisateurs étant donné la disponibilité universelle de piles HTTP sur de nombreuses plateformes
 - marche bien sur de petits LLNs requérant des latences de message de plusieurs secondes (home energy management, etc.)
 - https://en.wikipedia.org/wiki/Representational_state_transfer
- LWM2M (Lightweight M2M) : standard système de l'Open Mobile Alliance, incluant DTLS, CoAP, Block, Observe, SenML et Resource Directory (interface appareil-serveur + structure objet)

Architecture REST

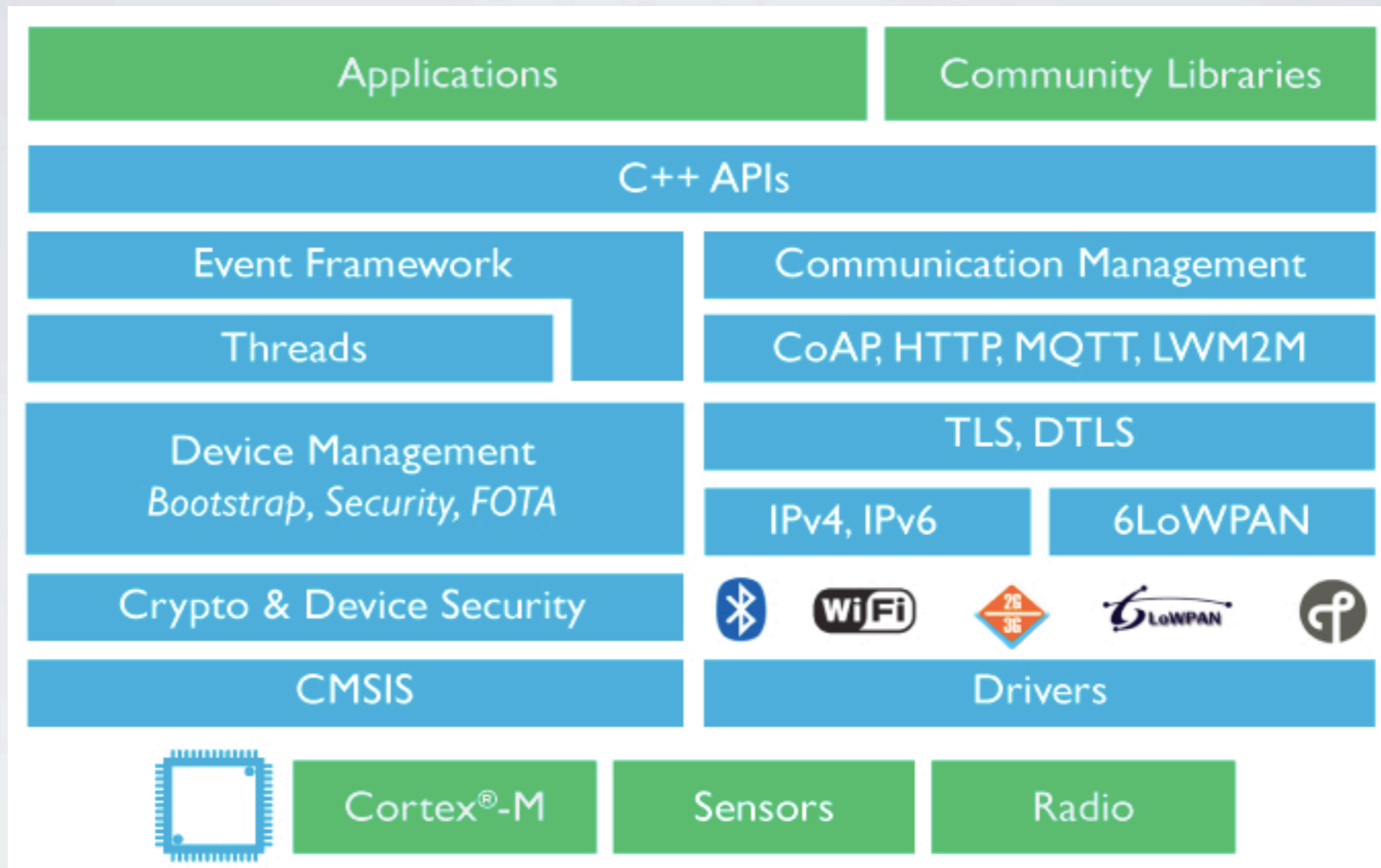


- CoAP dans une architecture REST

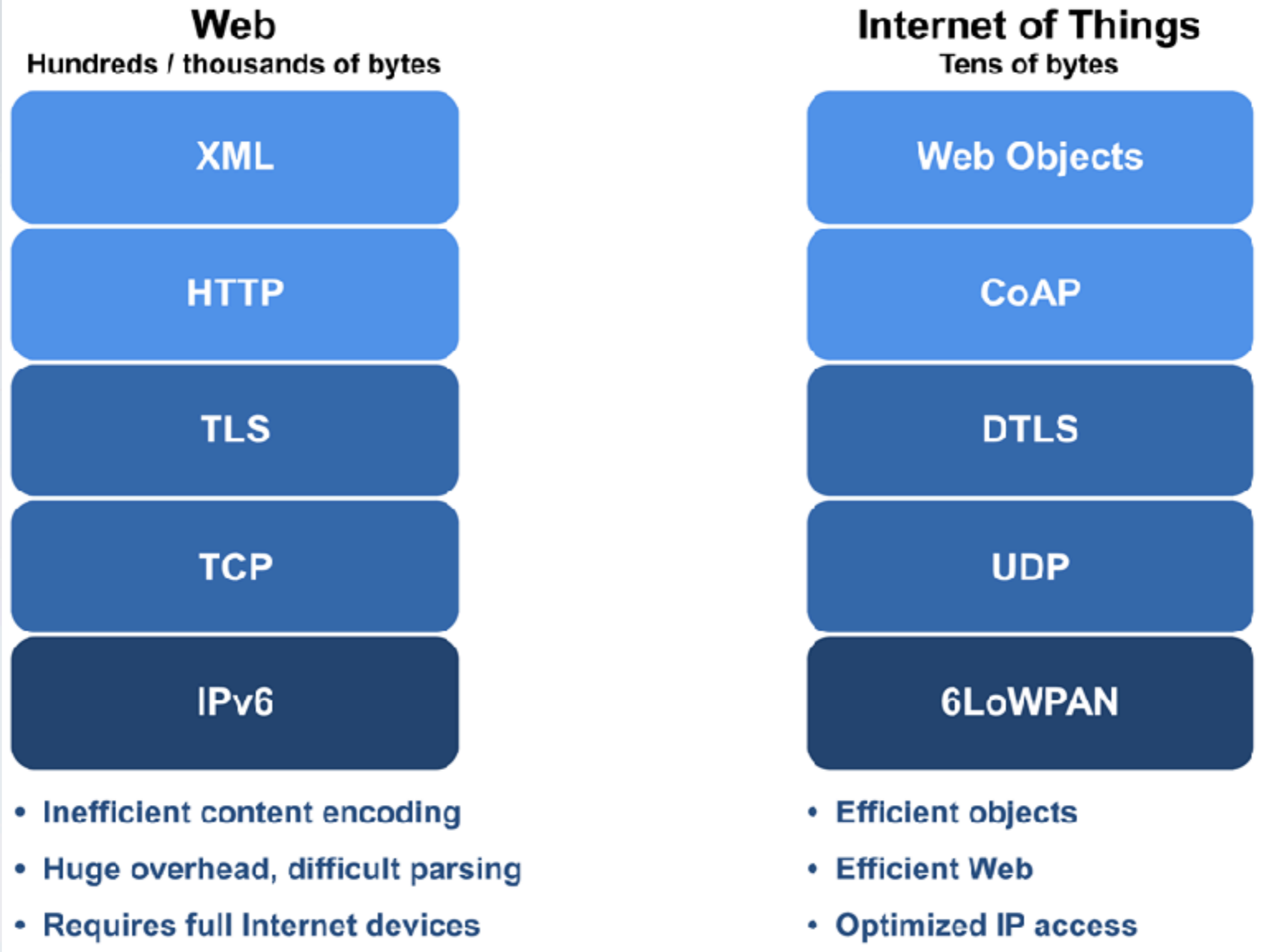
COMPARAISONS

Protocol	CoAP	XMPP	RESTful HTTP	MQTT
Transport	UDP	TCP	TCP	TCP
Messaging	Request/Response	Publish/Subscribe Request/Response	Request/Response	Publish/Subscribe Request/Response
2G, 3G, 4G Suitability (1000s nodes)	Excellent	Excellent	Excellent	Excellent
LLN Suitability (1000s nodes)	Excellent	Fair	Fair	Fair
Compute Resources	10Ks RAM/Flash	10Ks RAM/Flash	10Ks RAM/Flash	10Ks RAM/Flash
Success Storied	Utility Field Area Networks	Remote management of consumer white goods	Smart Energy Profile 2 (premise energy management/home services)	Extending enterprise messaging into IoT applications

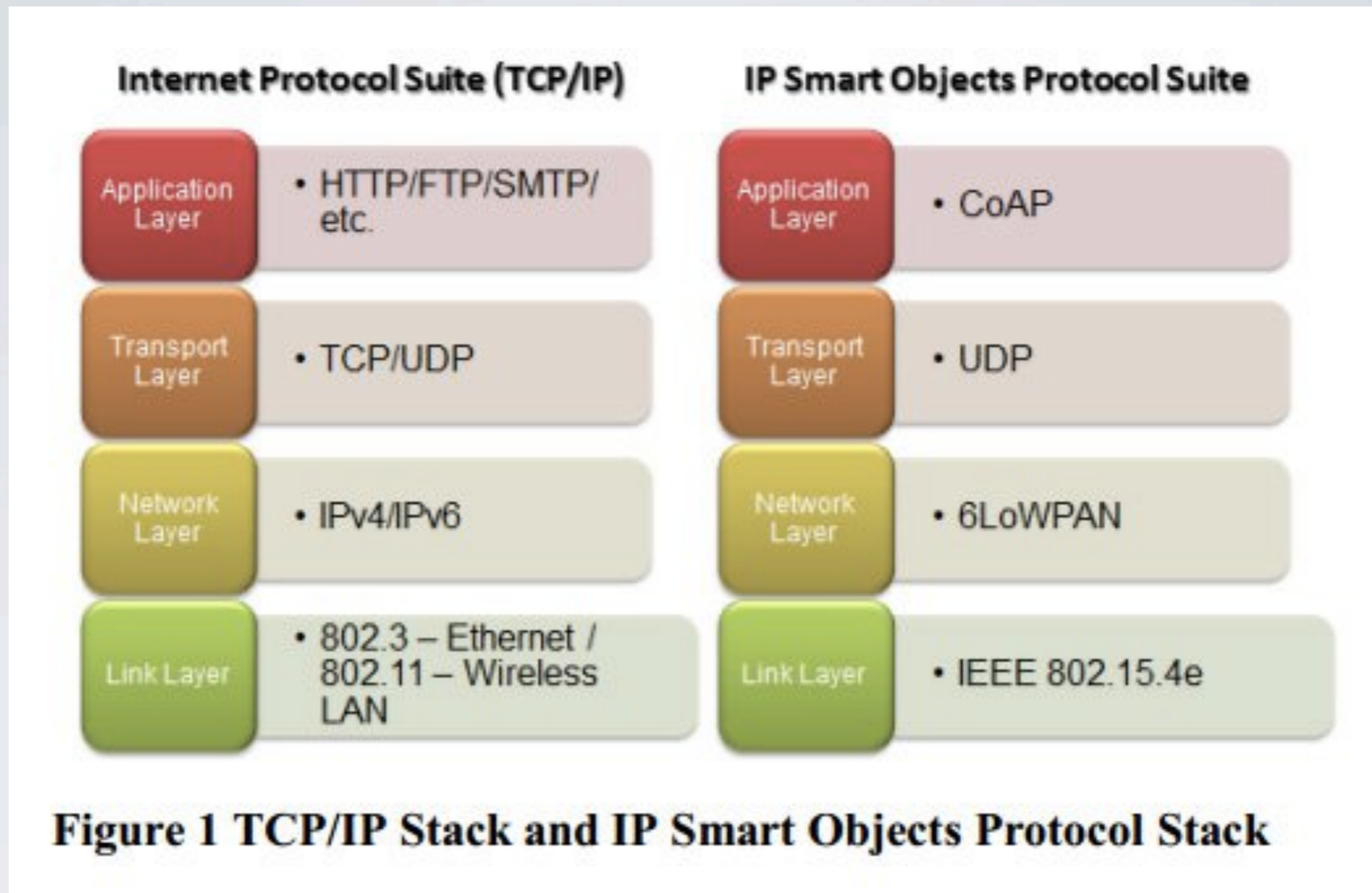
- <http://blogs.cisco.com/ioe/beyond-mqtt-a-cisco-view-on-iot-protocols>



Architecture de Mbed (rediffusion)



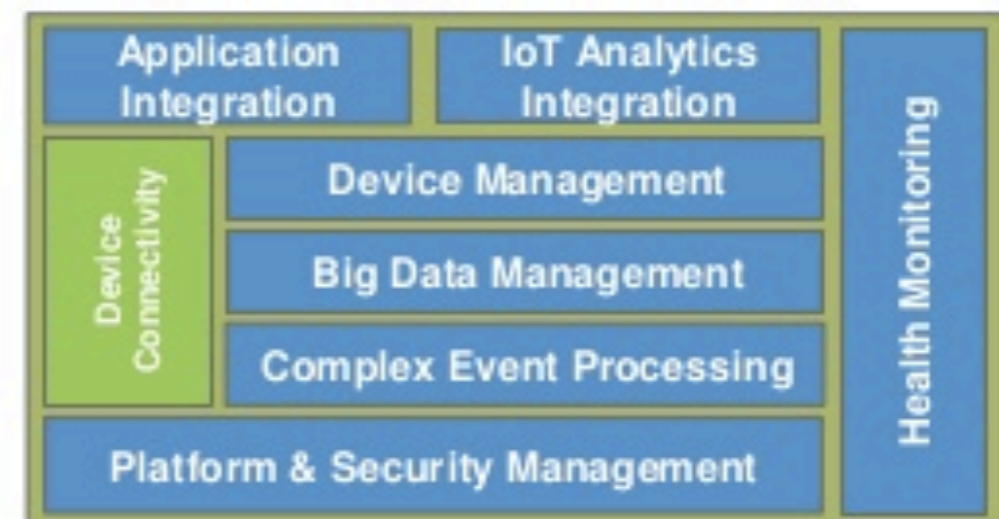
• <http://micrium.com/iot/internet-protocols/>



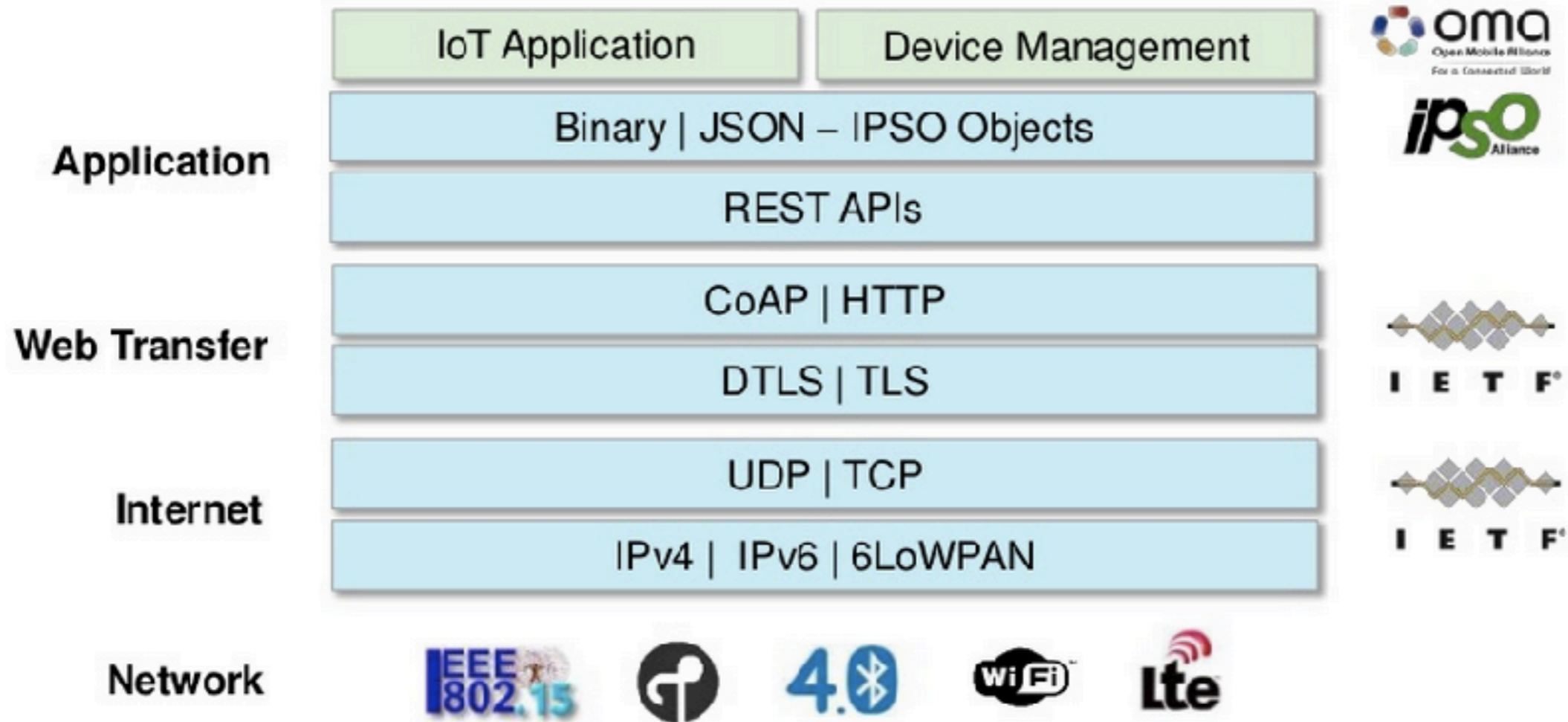
- Graphic via Ronak Sutaria and Raghunath Govindachari from Mindtree Labs in "Making sense of interoperability: Protocols and Standardization initiatives in IOT"

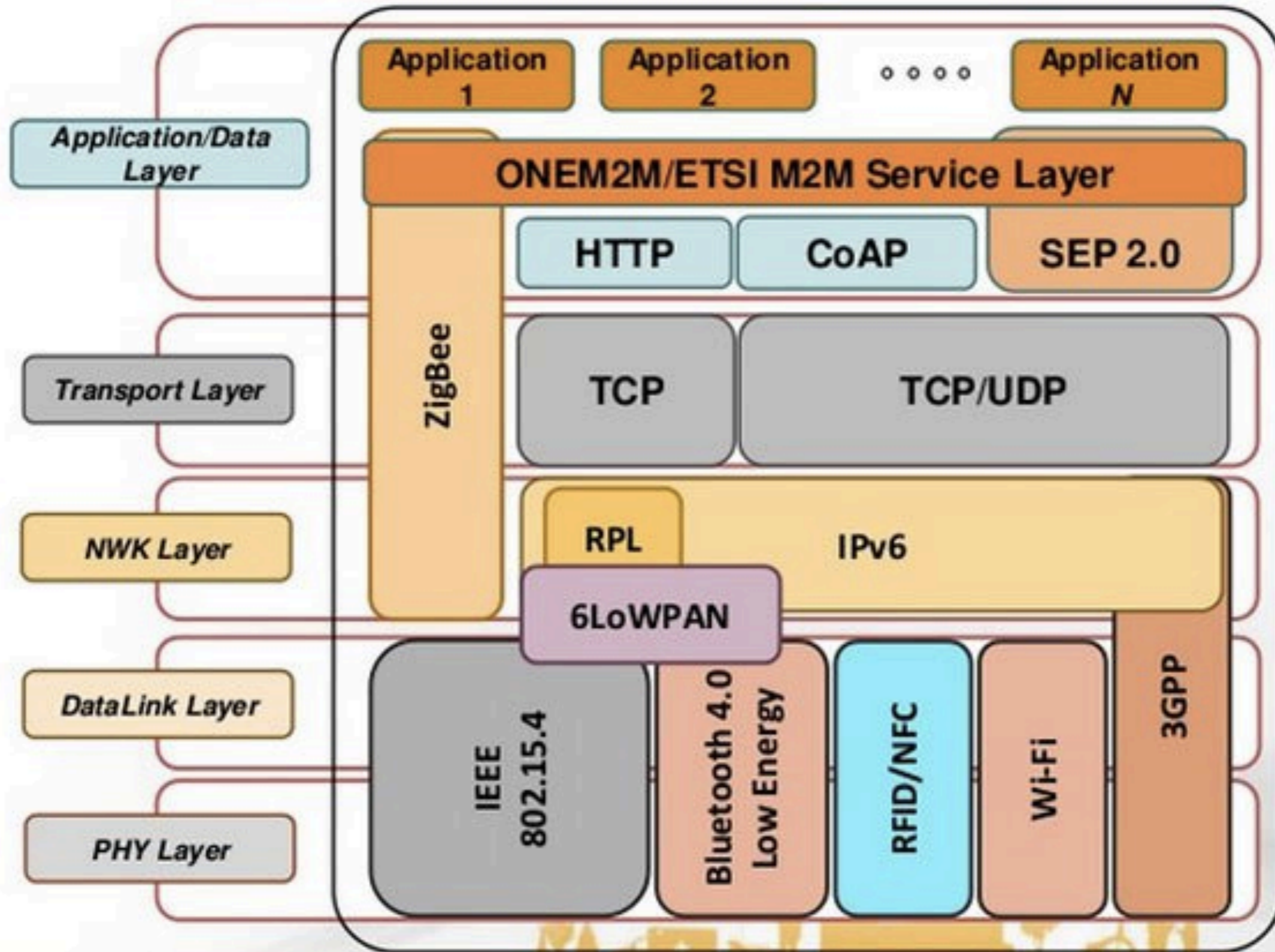
Why MQTT

- M2M Messaging Protocol
- Low Bandwidth / Low Power
- 2-way Communication
- Publish and Subscribe
- Hierarchical Topic Namespaces
- Data Payload Agnostic
- Device Initiated Connection
- Firewall-friendly
- SSL and Authenticated
- Large ecosystem



Remember the I in IoT!

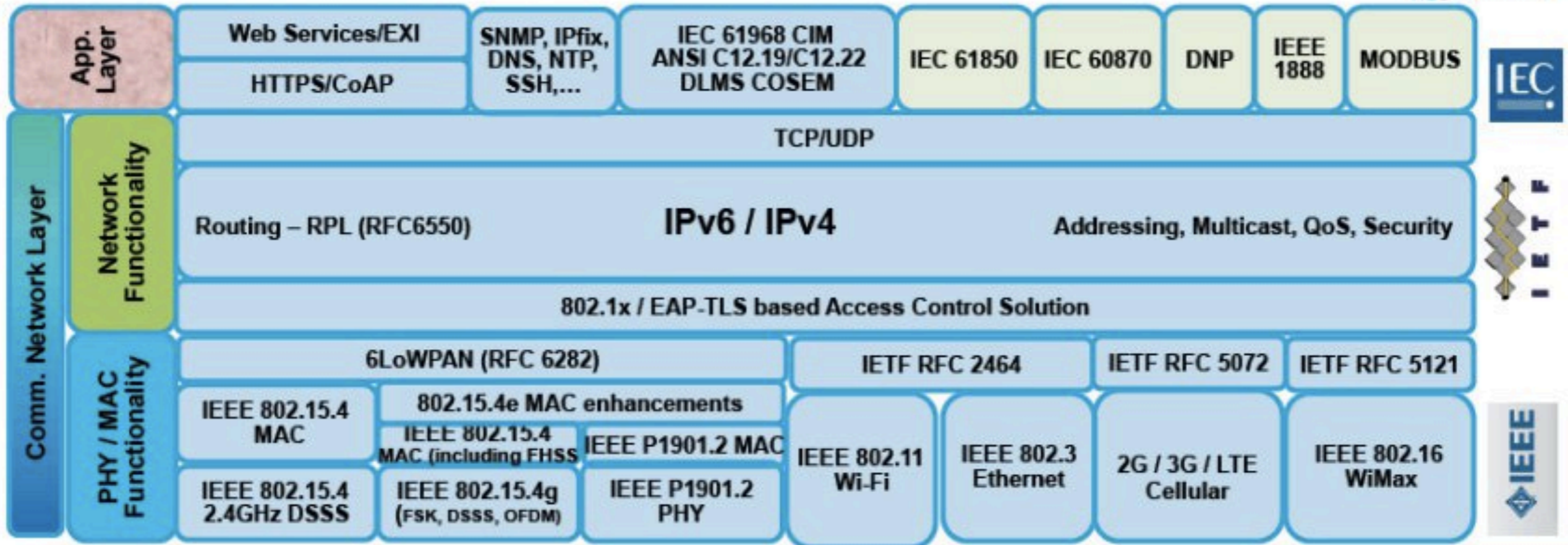






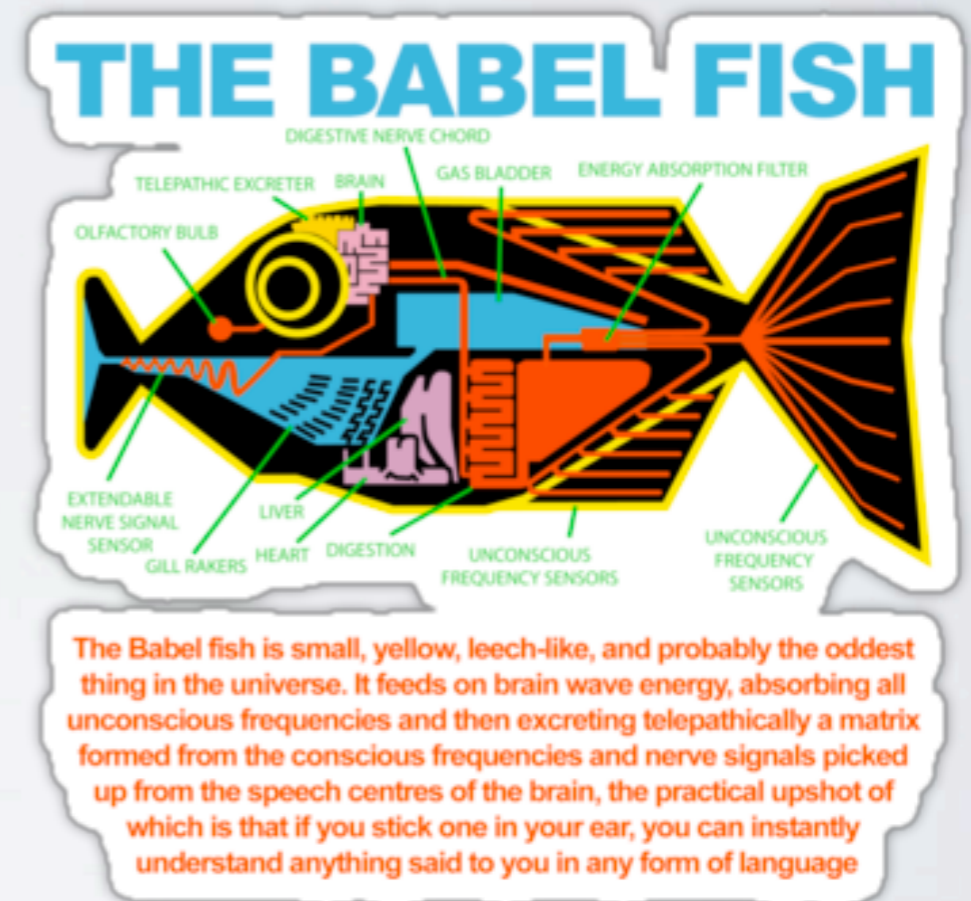
Or in industry speak ...

Open Standards Reference Model

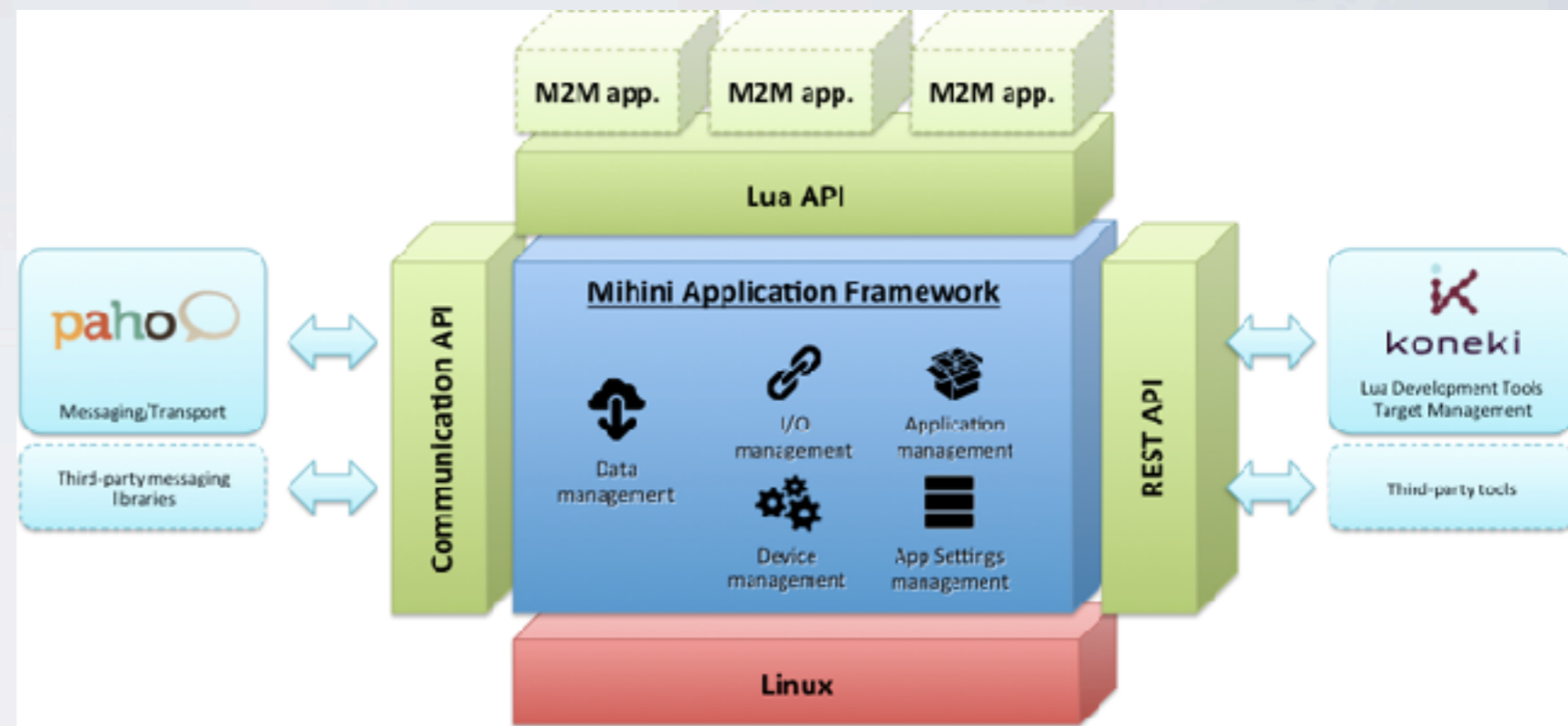


LA SOLUTION ?

- problématique M2M
➔ trop de protocoles
- une solution
➔ faire des passerelles (gateway)
- projet Mihini de la fondation Eclipse : facilitateur de traduction universelle



MIHINI 1/2



- un projet libre
 - initié chez Sierra Wireless
 - sous la houlette de la fondation Eclipse
- un framework Linux complet à déployer : abstraire toutes les difficultés d'implémentation du M2M
- dans un écosystème M2M libre
 - Paho (<http://www.eclipse.org/paho/>) : M2M messaging (MQTT)
 - Koneki (<http://www.eclipse.org/koneki/>) : développement, simulation, tests

POINTS FORTS DE MIHINI 2/2

- un runtime pour l'embarqué
 - tous les outils pour monter ses applications communicantes sans se soucier des problématiques de transport, encapsulation, disponibilité du réseau, protocoles divers (commandes AT, MQTT, etc.) → abstraction
 - gestion transparente des files de messages optimisées, avec priorités, attentes, agrégation
 - gestion de commandes distantes, d'upgrades, etc.
 - impémentation en Lua + C & API
- un projet bien ficelé
 - porté par l'Eclipse foundation
<http://www.eclipse.org/mihini/>
 - repository git public (<http://git.eclipse.org/c/mihini>)
 - documenté
 - communauté, présence sur twitter (@Mihini)

ARCHITECTURE IoT

- objet connecté... mais à quoi ?
 - à Internet ? Mais comment ??
 - la plupart des protocoles réseaux matériels ne sont que de courtes portées
- Au choix :
 - passer par un relai
 - par un téléphone portable
 - par une box FAI
 - par une gateway personnalisée
 - module GPRS/3G/4G
 - SigFox/LoRa

SCHÉMA



VIA TÉLÉPHONE PORTABLE

- Idée : utiliser le téléphone portable de l'utilisateur pour transmettre les données
 - sert aussi à la visualisation, à la récupération des données depuis le cloud, etc.
 - à présent, tout le monde a un smartphone
 - en outdoor, passe par le réseau GSM
 - en indoor, l'utilisateur configure souvent le WiFi de sa box ADSL
- Appairage du device et du téléphone
 - par NFC : nécessite de poser le téléphone (authentification induite par la proximité), données transmises assez faibles (quelques dizaines de ko)
 - par bluetooth : nécessite une distance courte suffisante, mais aussi de confirmer que l'appareil appairé est bien le bon (pour éviter le man-in-the-middle ou les erreurs), données transmises à fort débit (10aine ko/s BLE, x10 en v2)
 - mix des deux : « beam » sur Android, utilise le NFC pour l'authentification de la communication Bluetooth
- N'oubliez pas de recharger souvent votre téléphone... (Beaucoup sollicité en communication = batterie en rade rapidement)

VIA BOX DU FAI

- Idée : dans la maison, se servir du point de connexion à Internet qu'est la box du fournisseur d'accès (câble/xDSL/fibre)
 - ne sont pas encore prêtes à relayer les connexions de protocoles domotiques
 - limité au WiFi
 - SSID personnalisé par l'utilisateur (souvent laissé au défaut usine...)
 - mot de passe personnalisé par l'utilisateur (plus ou moins faible)
 - canal tout le temps laissé par défaut (crée des zones de saturation...)
 - passe parfois mal les murs (utiliser des relais CPL ?)
- Implique de configurer l'objet pour se connecter au WiFi
 - rentrer le mot de passe et le SSID sur un appareil qui ne dispose pas d'écran
 - des solutions existent (compliquées)
 - espoir du WPS ?
- Pourvu qu'il n'y ait pas de coupure Internet... (Ou la mauvaise idée de brancher des systèmes de sécurité sur un SPOF avéré)

VIA UNE PASSERELLE PERSONNALISÉE

- La course à la gateway
 - insuffisance des box des FAI (aka ISP)
 - box-routeur-WiFi plus efficace : <https://on.google.com/hub/> (2 ports gigabit Ethernet dont un Wan, 1 connecteur USB 3.0, 13 antennes Wi-Fi, Bluetooth et zigbee + HP = \$199)
 - box pour smart home : Somfy, Castorama (Blyssbox par Myxyty), Fibaro (solution complète Z-Wave)
 - besoins métiers particuliers, avec des protocoles hétérogènes : concentrateur des smart cities, domotique pour bâtiments (protocole KNX), bornes de recharge électrique, etc.
 - concentrateur de données dans les smart grids
- Nécessite de faire un travail de réalisation électronique & réseau, à des prix bas/maîtrisés — mission impossible
- Gateway de traduction universelle : <http://1248.io/products.php> (mort ?)

A CONTEXT-AWARE PLATFORM FOR A MEANINGFUL DIGITAL LIFE IN THE HOME

Thanks to a powerful set of resources, the IJENKO platform operates with any smart object, bridging non-IP sensors & actuators and IP devices into a single experience for your users. Our context-aware platform learns, analyses, predicts, coaches and acts on behalf of the end-user depending on the priorities defined by them, and their overall life environment: the energy pricing signals, the demand-side management programs they have subscribed to, the weather, their presence or absence. End-users are able to choose almost any branded device on the market and create the right interactions between their Home gateway and any IP or non-IP devices.

The Service Providers' users will be the powerful creators of their own reality and be able to push their home to its fullest potential.

IJENKO PLATFORM DESIGNED TO INTERACT WITH:



<http://www.ijenko.com/internet-of-everything>

INVASIONS BARBARES ?

- Faute de protocole standard et commun, les (nouveaux) acteurs de l'Internet vont faire... leur propre standard !?
- Samsung, ARM et Google créent le Thread Group
<http://www.lemondeinformatique.fr/actualites/lire-internet-des-objets-samsung-arm-et-google-creent-le-thread-group-58094.html>
- Google : OS Brillo et protocole de communication avec téléphone Weave
http://www.frandroid.com/marques/google/286644_google-presente-brillo-weave-lunification-objets-connectes
- Qeo/AllJoyn : alliance AllSeen, Technicolor, Haier, LG, Panasonic, Qualcomm, Sharp, Silicon Image et TP-Link (25 membres)
http://www.lembarque.com/internet-des-objets-technicolor-apporte-son-logiciel-distribue-qeo-a-lalliance-allseen_001633

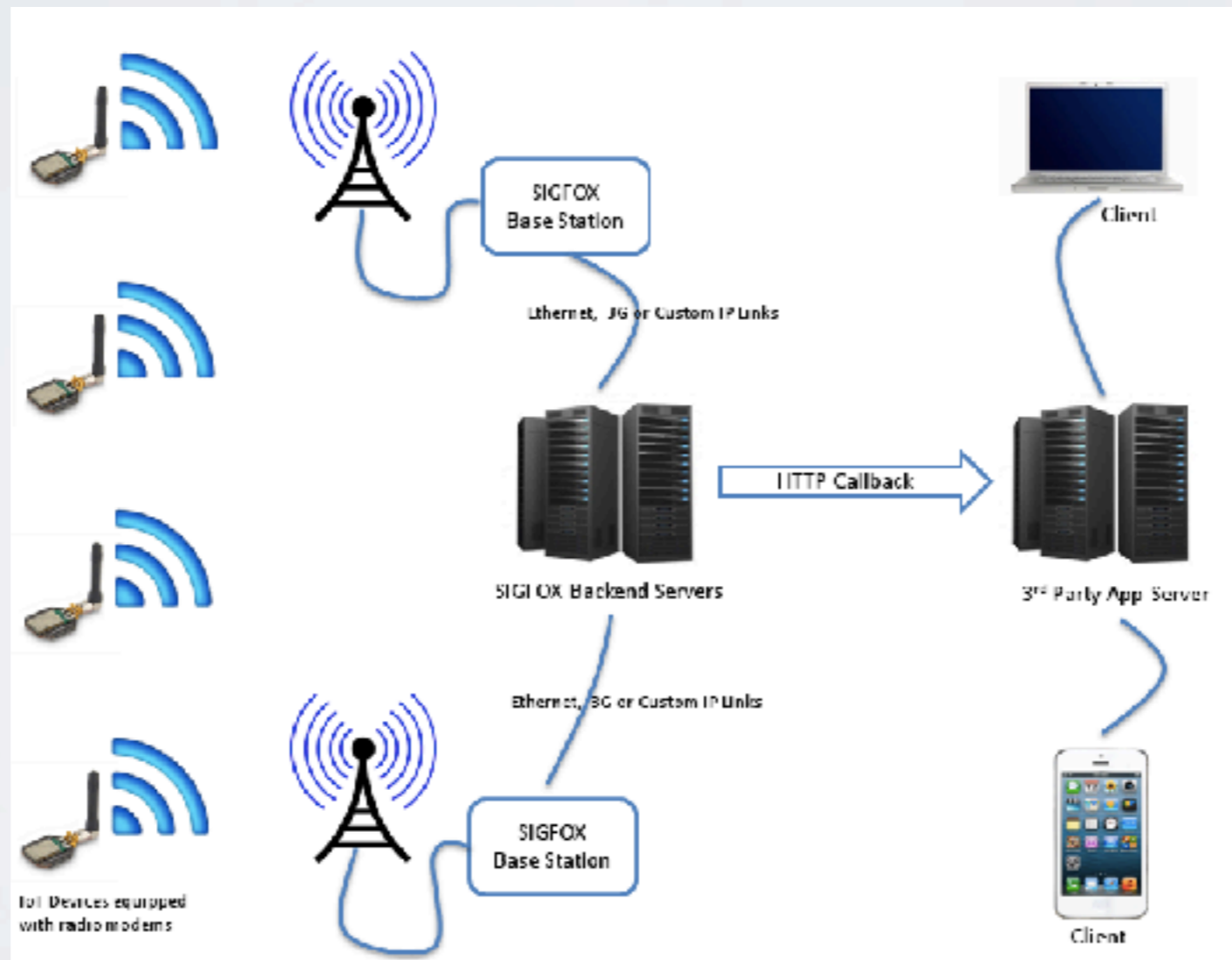
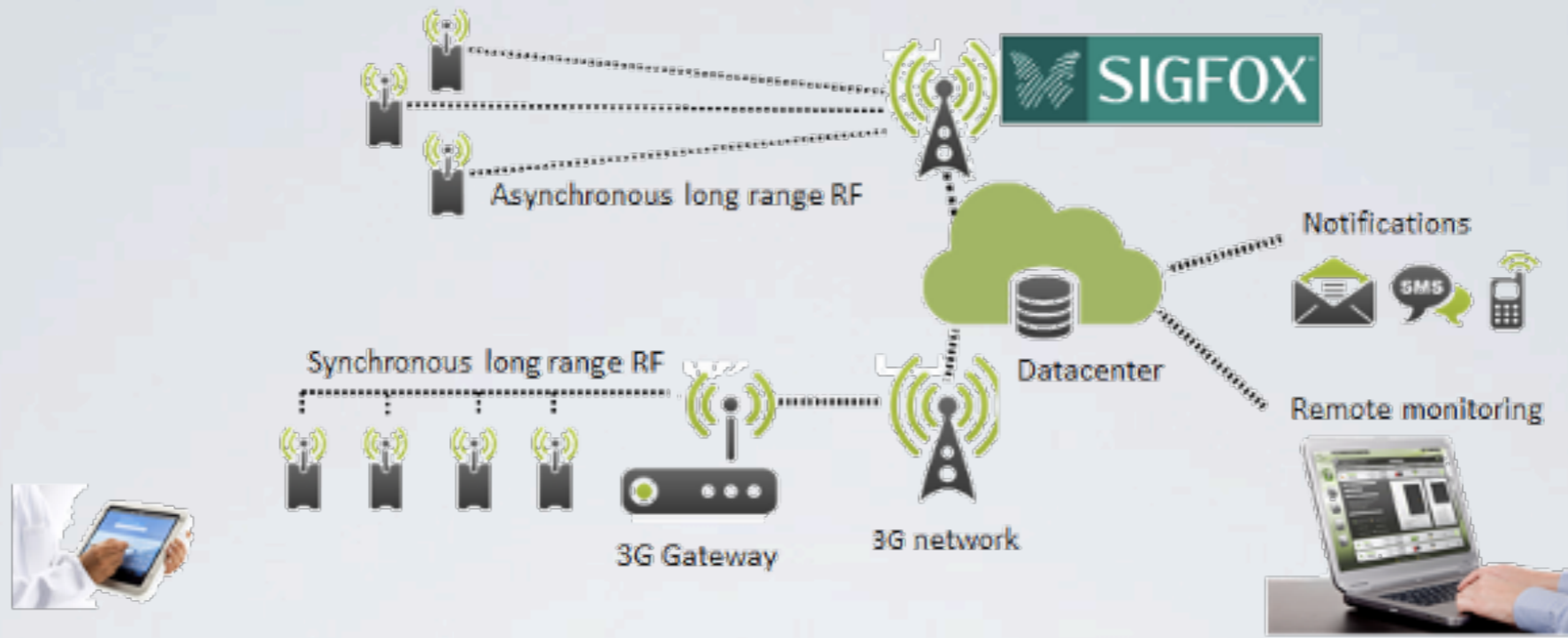


PAR UN MODULE GSM

- Utilisation directe du réseau téléphonique
 - déjà existant, excellente couverture du territoire
 - roaming pour l'international
- Dépendant de la couverture réseau
 - en outdoor, couverture du territoire, saturation des antennes-relais, problème des tunnels, des transports sous-terrains
 - en indoor, problématique de la captation du signal
- Nécessite une puce SIM (peut être incluse « en dur » dans l'électronique du module — verrouillée sur un opérateur particulier)
- Nécessite un abonnement téléphonique ! (Spécifique M2M, mais tout de même à gérer & payer)
- Solutions matérielles de modules par Telit, Sierra Wireless/Wavecom, ...

PAR SIGFOX/LORA

- Pas de solution réellement convenable pour l'envoi de petites données de l'IoT en longue distance pour faible consommation électrique
- Solutions inadaptées à l'envoi de peu de données sur de grosses couvertures de territoire sans prise de tête (réseau téléphonique = carte SIM, conso électrique...)
- Low-Power Wide-Area Network (LPWAN) : <https://en.wikipedia.org/wiki/LPWAN> (11 solutions listées)
- Apparition de deux concurrents majeurs :
 - SigFox : plutôt spécialisé en couverture outdoor, installe des antennes radio très grande portée (ex : retrouver son vélo volé, faire communiquer son pioupiou <http://pioupiou.fr/>), a débauché Anne Lauvergeon, Toulousains à SF, la compagnie déploie son propre réseau (pas forcément sur les mêmes fréquences dans tous les pays, problème de compatibilité potentiel)
 - LoRa (Long Range Wide-area network) : plutôt spécialisé en couverture indoor, poussé par Bouygues (pour ses bâtiments) et Orange alors que techno de Semtech



COMPARAISONS

- <http://www.disk91.com/2015/news/technologies/iot-m2m-quick-panorama-on-main-technologies/>

	modem price	antenna price	1 year fee	documentation	conso TX	conso RX	rx life duration, continuously
Sigfox	2-3€* (avant: 18 €)	5 €	7 à 14 €	bonne	45mA	15 mA	200 h
LoRa	2x €	5 €	inconnu	pas vraiment mais bientôt	100 mA	25 mA	120 h
2G/3G	à partir de 15 € (GPRS)	1 €	30 €	bonne	2000 mA	100 mA	30 h

* http://www.lembarque.com/le-cout-des-modules-radio-de-connexion-au-reseau-sigfox-chute-a-deux-dollars_005490

Auto Oven Cooks Hot Dogs



WANT a hot dog on the road? Just plug this heater into your car's electrical system. It cooks two wieners in three to five minutes. Priced at \$3.95, it also comes in a 115-volt version for the home. It is called the Hot Dog Sizzler® and is made by the Thomas Manufacturing Co., of Chicago.

PROTOTYPAGE (RAPIDE)

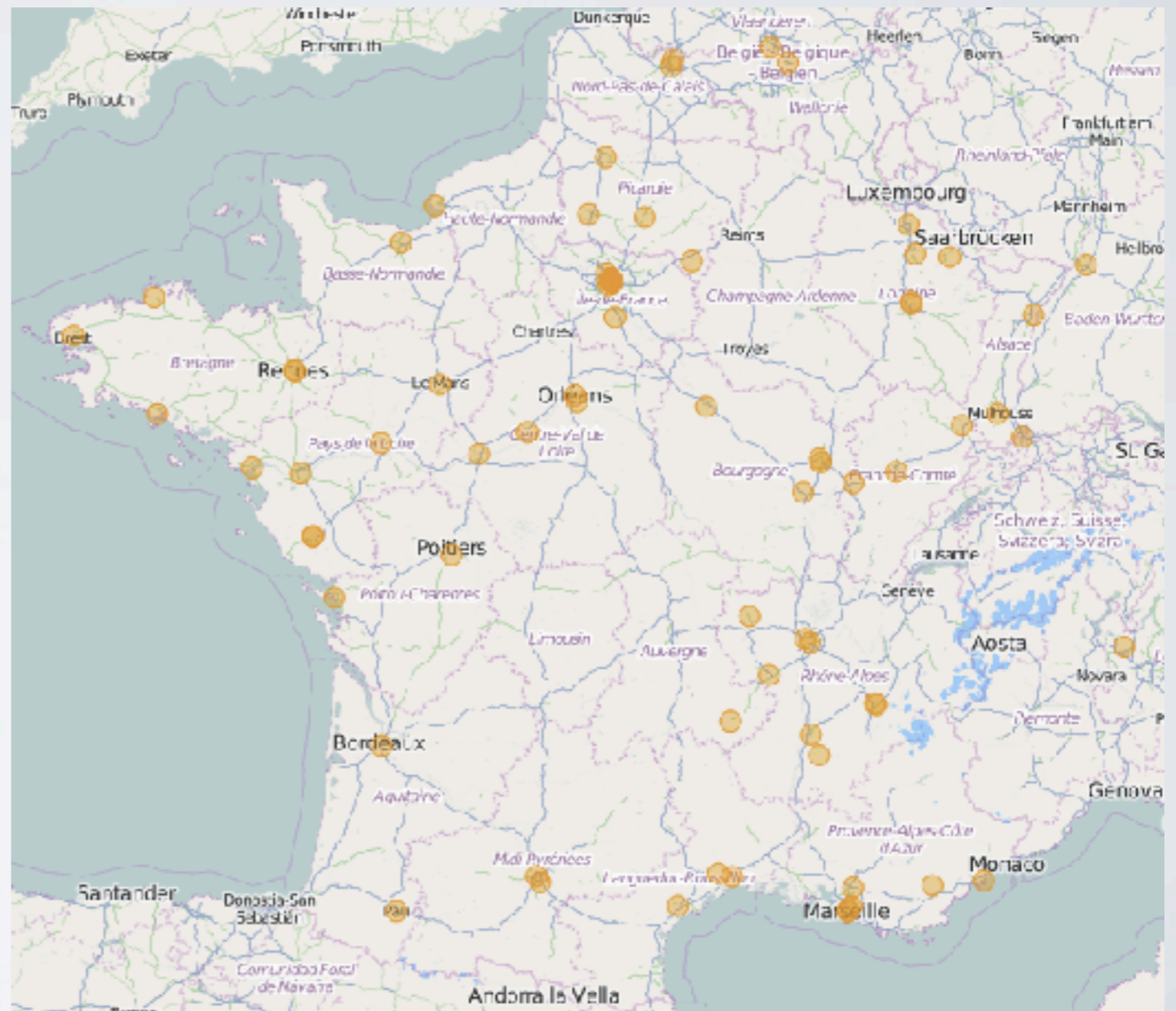
Arduino, RPi, Beaglebone, Udoo, Edison...

LA MÉCANIQUE

- Apparition d'outils et de lieux de prototypage rapide
 - les Fablabs : lieux dédiés avec machines et experts (exemple : Usine10)
 - les imprimantes 3D
 - très pratique pour fabriquer des boîtes à façon, sans faire de moule
 - nécessite un apprentissage (manière d'imprimer pour que ça tienne)
 - encore assez limité, mais évolution rapide
 - se souvenir que ça reste du prototypage, rien à voir avec une étude mécanique de matériaux !
- Savoir se servir de 3DS et autres logiciels de modélisation 3D
- Apprentissage par la théorie et... par le bouche à oreille/l'expérience

CARTE DES FABLABS

- <http://www.gotronic.fr/ins-carte-des-fablabs-50.htm>
- http://fablabo.net/wiki/Cartographie_des_fablabs_fran%C3%A7ais
- ne se recoupent pas vraiment
- il en manque...

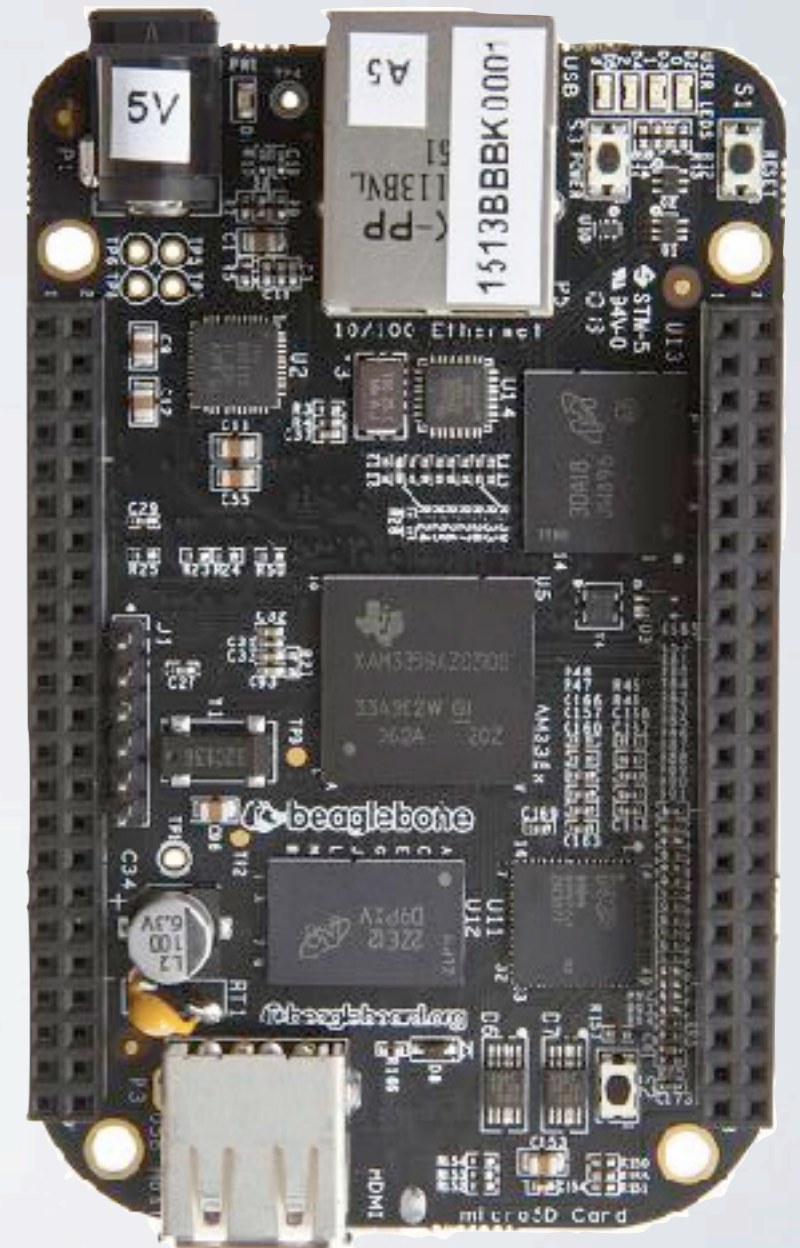


L'ÉLECTRONIQUE

- Pléthore de cartes de prototypage rapide
 - 2008 : apparition de petites cartes chinoises puis émergence de la Beagleboard
 - depuis : invasion de petites cartes avec CPU ou MCU
 - Beaglebone, Raspberry Pi, Arduino, Linino, Udo0
 - kits Sparkfun : <https://www.sparkfun.com/>
 - instrumentation via carte SD/MMC et/ou mini/micro-USB
 - entre 25 et 120€ généralement
 - plus ou moins faciles à interfacer, parfois des écosystèmes de cartes filles d'extension
- Attention : à ne pas mettre en production, pas fait pour !! Il existe parfois des cartes à l'architecture équivalente pour la production.

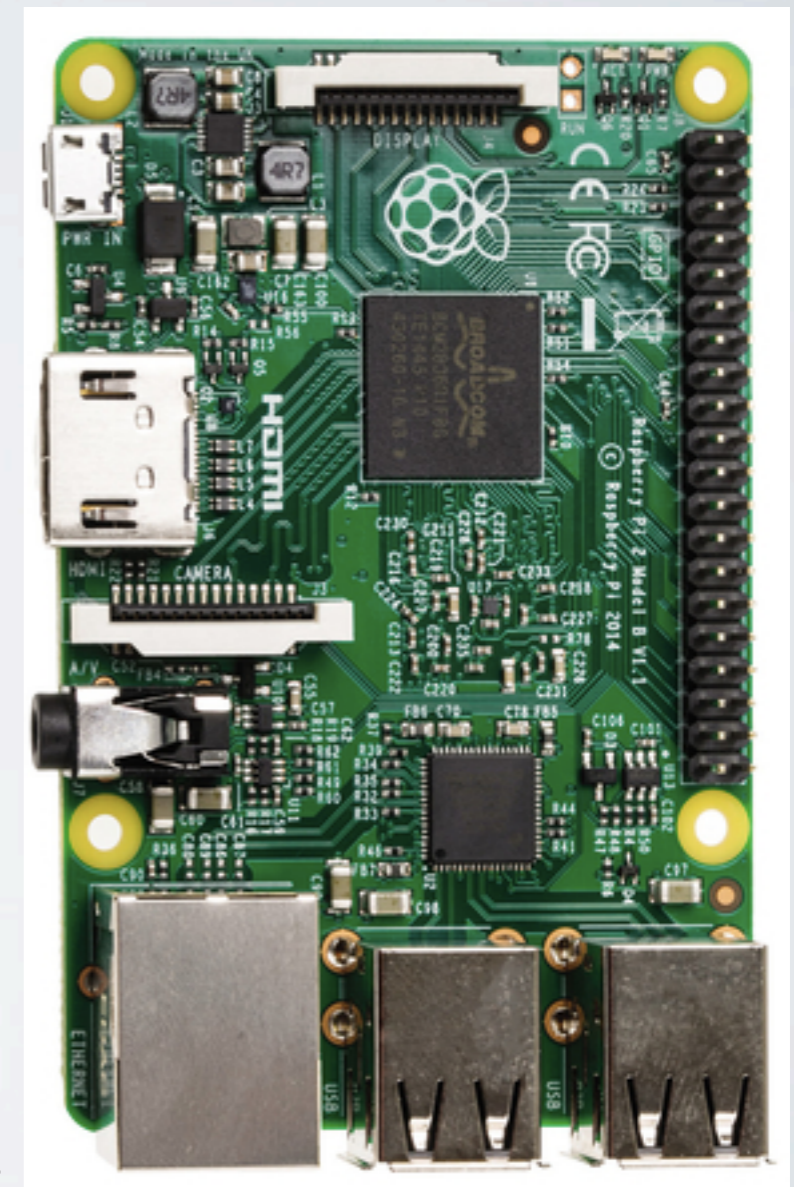
BEAGLEBONE BLACK

- TI, \$89 MSRP, credit-card-sized Linux computer, openHardware
- succède à la Beaglebone, qui succède à la Beagleboard-xM, et avant Beagleboard
- Processor: AM335x 1GHz ARM® Cortex-A8
- 512MB DDR3 RAM
- 4GB 8-bit eMMC on-board flash storage
- 3D graphics accelerator
- NEON floating-point accelerator
- 2x PRU 32-bit microcontrollers USB client for power & communications
- USB host
- Ethernet
- HDMI
- 2x 46 pin headers



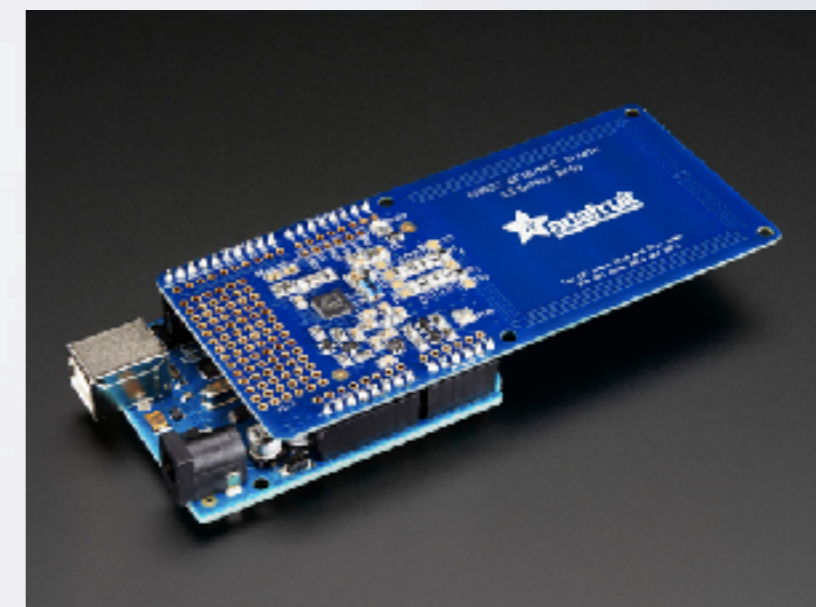
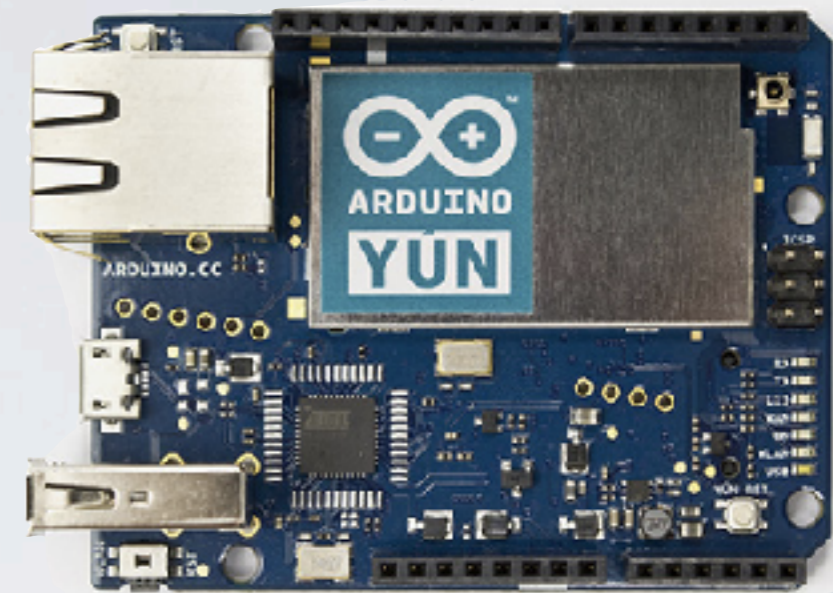
RASPBERRY PI 2 MODEL B

- Element 14 (Farnell), \$39, évolution de la RPi 2, qui évolue de la RPi model B, et RPi première du nom (avec soudures dégoules et aucune protection de tension)
- 900MHz quad-core ARM Cortex-A7 CPU
- 1GB RAM
- Like the (Pi 1) Model B+, it also has:
 - Full HDMI port
 - Ethernet port
 - Combined 3.5mm audio jack and composite video
 - Camera interface (CSI)
 - Display interface (DSI)
 - Micro SD card slot
 - VideoCore IV 3D graphics core
- 4 USB ports
- 40 GPIO pins
- reboote quand photographiée au flash : <http://korben.info/le-raspberry-pi-2-photosensible.html>



ARDUINO

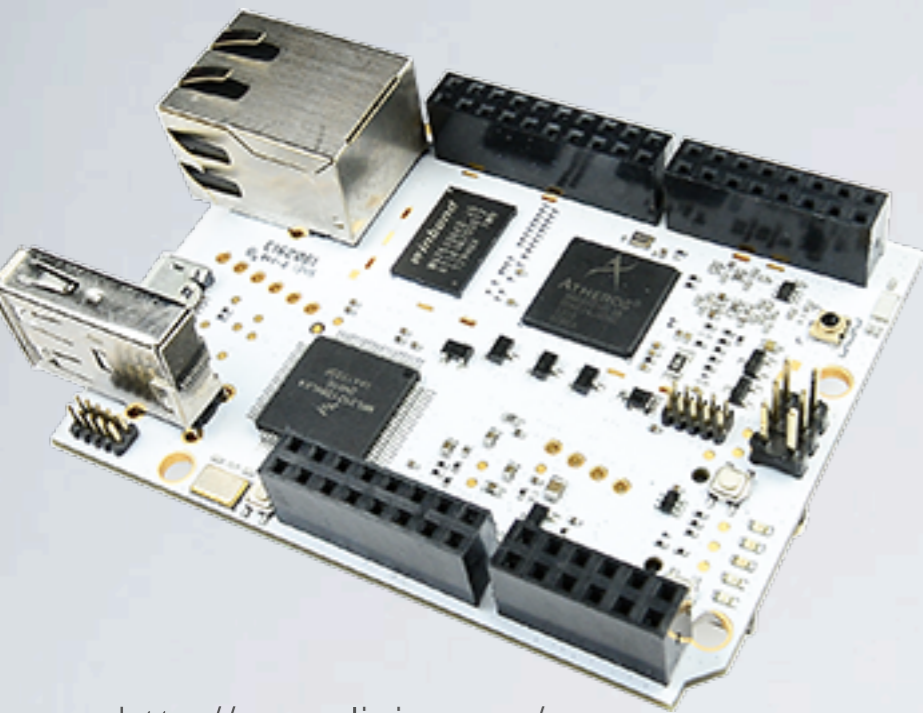
- Petites cartes de microcontrôleurs (programmation en C, communication série) et de « shields » compatibles (cartes d'extension avec des fonctionnalités précises : GSM, NFC, Ethernet...), montés sur des broches normalisées (openHardware)
- Yún : Microcontroller ATmega32U4, Operating Voltage 5V ; Input Voltage 5 ; Digital I/O Pins 20 ; PWM Channels 7 ; Analog Input Pins 12 ; DC Current per I/O Pin 40 mA ; DC Current for 3.3V Pin 50 mA ; Flash Memory 32 KB (of which 4 KB used by bootloader) ; SRAM 2.5 KB ; EEPROM 1 KB ; Clock Speed 16 MHz
- Arduino GSM Shield
- Adafruit PN532 NFC/RFID Controller Shield for Arduino + Extras
- XBee & Arduino : <http://xbeearduino.fr/>



ENTRY LEVEL	<div style="display: flex; justify-content: space-between; margin-bottom: 5px;"> ARDUINO/GENUINO UNO ARDUINO PRO ARDUINO PRO MINI ARDUINO/GENUINO MICRO </div> <div style="display: flex; justify-content: space-between; margin-bottom: 5px;"> ARDUINO NANO ARDUINO/GENUINO STARTER KIT ARDUINO BASIC KIT </div> <div style="display: flex; justify-content: space-between;"> ARDUINO MOTOR SHIELD </div>
ENHANCED FEATURES	<div style="display: flex; justify-content: space-between; margin-bottom: 5px;"> ARDUINO/GENUINO MEGA ARDUINO ZERO ARDUINO DUE ARDUINO PROTO SHIELD </div>
INTERNET OF THINGS	<div style="display: flex; justify-content: space-between; margin-bottom: 5px;"> ARDUINO YÚN ARDUINO ETHERNET SHIELD ARDUINO GSM SHIELD ARDUINO WIFI SHIELD 101 </div>
WEARABLE	<div style="display: flex; justify-content: space-between; margin-bottom: 5px;"> ARDUINO GEMMA LILYPAD ARDUINO USB LILYPAD ARDUINO MAIN BOARD </div> <div style="display: flex; justify-content: space-between;"> LILYPAD ARDUINO SIMPLE LILYPAD ARDUINO SIMPLE SNAP </div>
3D PRINTING	<div style="display: flex; justify-content: space-between;"> MATERIA 101 </div>

BOARDS
 MODULES
 SHIELDS
 KITS
 ACCESSORIES
 COMING NEXT

LININO



- <http://www.linino.org/>
- Linux distribution based on OpenWRT (Linino)
- Freedog KL25Z: MIPS board based on the Atheros AR9331 and the Kinetis L Series KL25Z MCUs built on ARM® Cortex™-M0+ processor
 - WiFi support, USB-A port, micro-SD card slot, 20 digital input/output pins (of which 7 can be used as PWM outputs and 12 as analog inputs), 8 MHz crystal oscillator, micro USB connection, SWD header, and 3 reset buttons.
 - external row of the pin-header is Arduino compatible.
- Linino ONE: MIPS board based on the Atheros AR9331 and the ATmega32u4
 - WiFi support, 20 digital input/output pins (of which 7 can be used as PWM outputs and 12 as analog inputs), 16 MHz crystal oscillator, micro USB connection, ICSP header, 2 reset buttons, 1 user button
 - Add-ons : dogRJ45, dogUSB



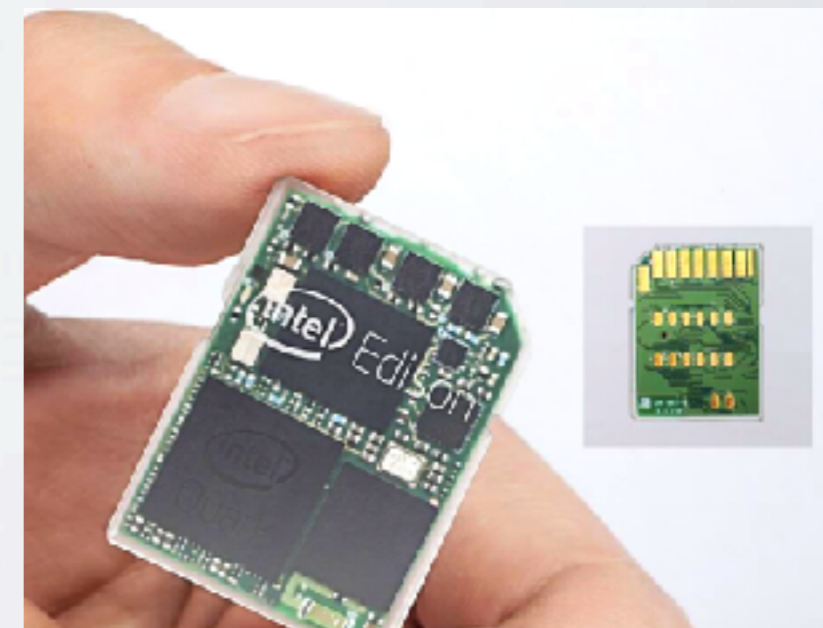
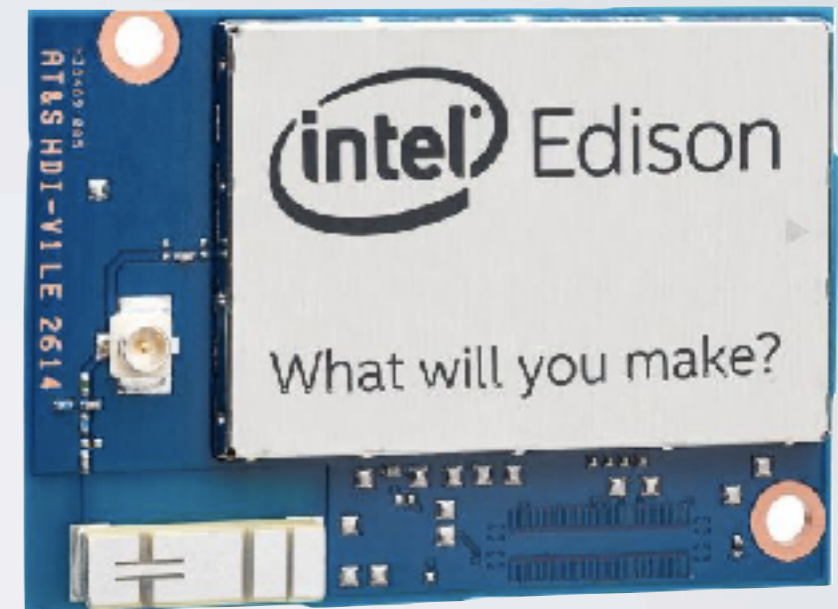
MINNOWBOARD MAX

- <http://www.minnowboard.org/meet-minnowboard-max/>
\$145.95, openHardware
- 64-bit Intel® Atom™ E38xx Series SoC
- Integrated Intel® HD Graphics with Open Source hardware-accelerated drivers for Linux OS, micro HDMI connector
- DDR3 RAM (1Gb or 2Gb)
- 8 MB SPI Flash (System Firmware Memory, UEFI)
- 1 – Micro SD SDIO
- 1 – SATA2 3Gb/sec
- 1 – USB 3.0 (host)
- 1 – USB 2.0 (host)
- 1 – Serial debug via FTDI cable (sold separately)
- 10/100/1000 Ethernet
- 8 – Buffered GPIO pins (2 pins support PWM)
- 99 x 74mm (2.9 x 3.9in)
- I2C & SPI bus
- 0 – 70 deg C



EDISON

- Processeur double cœur hautes performances et microcontrôleur simple cœur pour la collecte des données complexes dans un boîtier basse consommation
- SoC 22 nm Intel Atom avec processeur double cœur à 500 Mhz et un microcontrôleur programmable (MCU) à 100 Mhz
- 1 Go de mémoire, 4 Go de stockage
- Connectivité Wi-Fi, Bluetooth LE 4.0
- 40 interfaces GPIO multiplexées avec options de cartes d'extension pour couvrir l'ensemble du cycle de conception d'un projet et améliorer la flexibilité
- <http://www.intel.fr/content/www/fr/fr/do-it-yourself/edison.html>



WANDBOARD

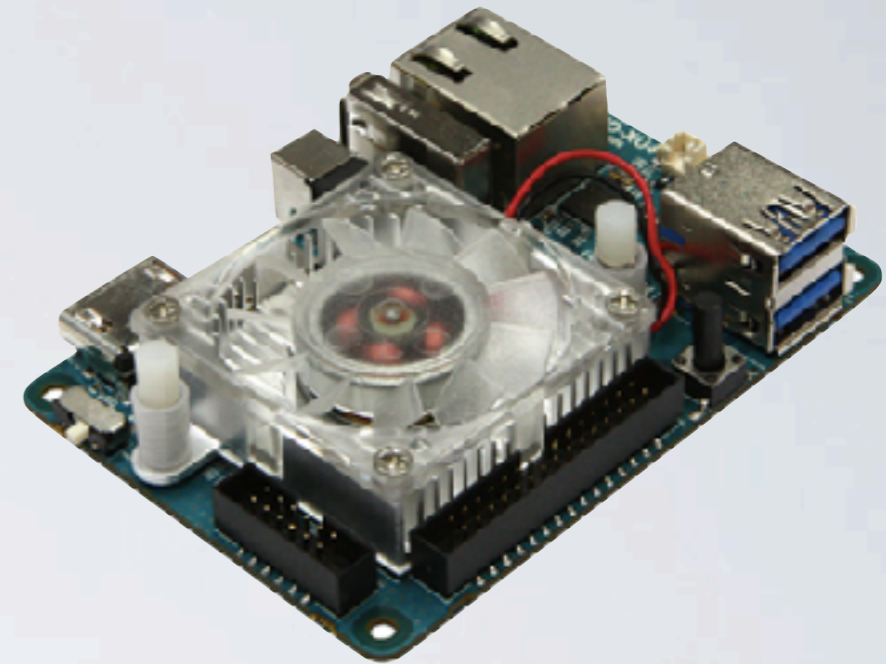
- fabriquée par Technexion
- communauté libre, openHardware
- très impliqué dans le projet Yocto



WANDBOARD
SOLO 79 USD
DUAL 99 USD
QUAD 129 USD

	Wandboard Solo	Wandboard Dual	Wandboard Quad
Processor	Freescale i.MX6 Solo	Freescale i.MX6 Duallite	Freescale i.MX6 Quad
Cores	Cortex-A9 Single core	Cortex-A9 Dual core	Cortex-A9 Quad core
Graphic engine	Vivante GC + Vivante GC 320 880	Vivante GC + Vivante GC 320 880	Vivante GC + Vivante GC 355 2000 + Vivante GC 320
Memory	512 MB DDR3	1GB DDR3	2GB DDR3

ODROID



- ODROID-XU4, US\$74.00

- Octa Core Linux Computer
- Samsung Exynos5422 Cortex-A15 2.0Ghz quad core and Cortex-A7 quad core

- Mali-T628 MP6 (OpenGL ES 3.0/2.0/1.1 and OpenCL 1.1 Full profile)

- eMMC5.0 HS400 Flash Storage

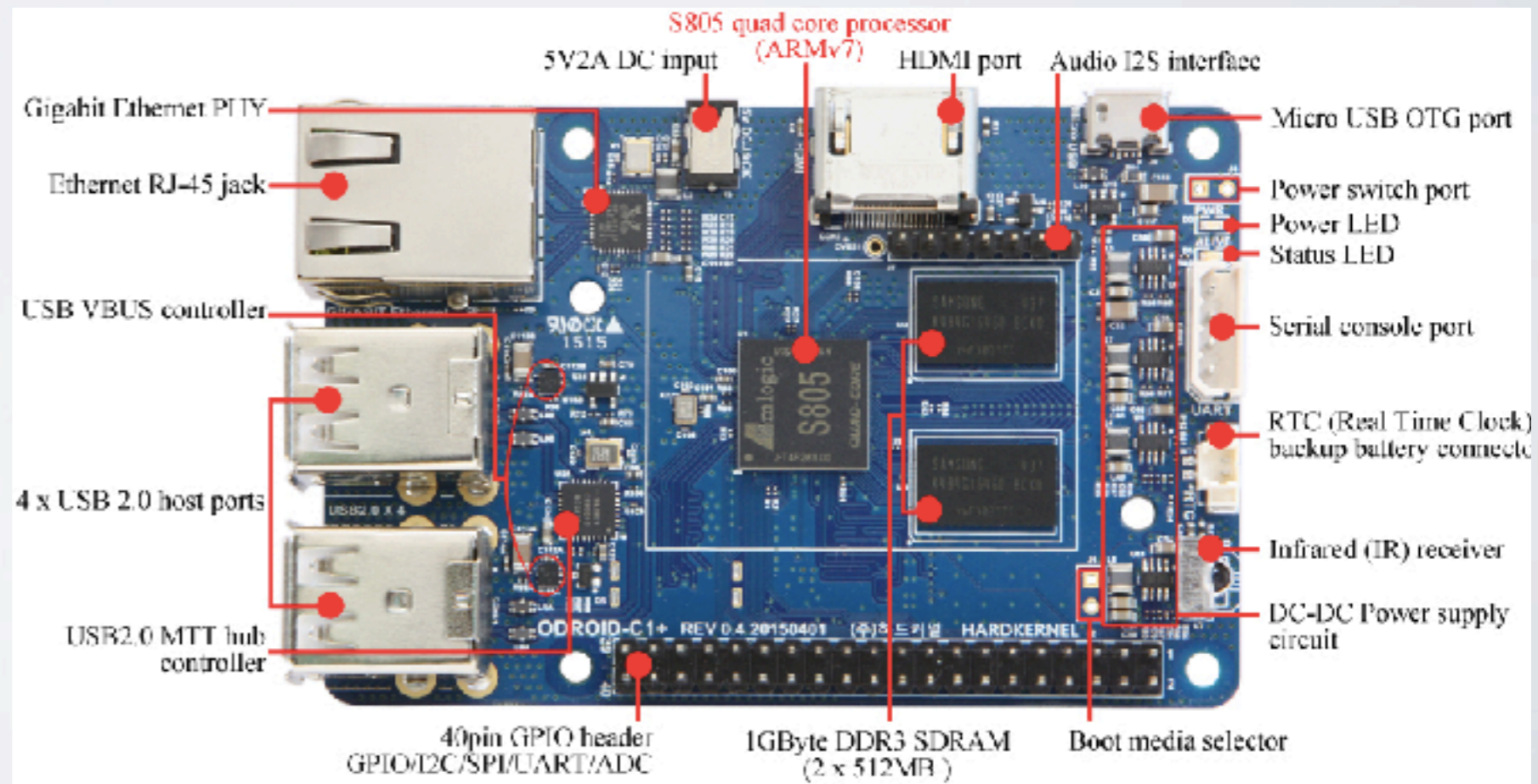
- 2 x USB 3.0 Host, 1 x USB 2.0 Host,

- Gigabit Ethernet port

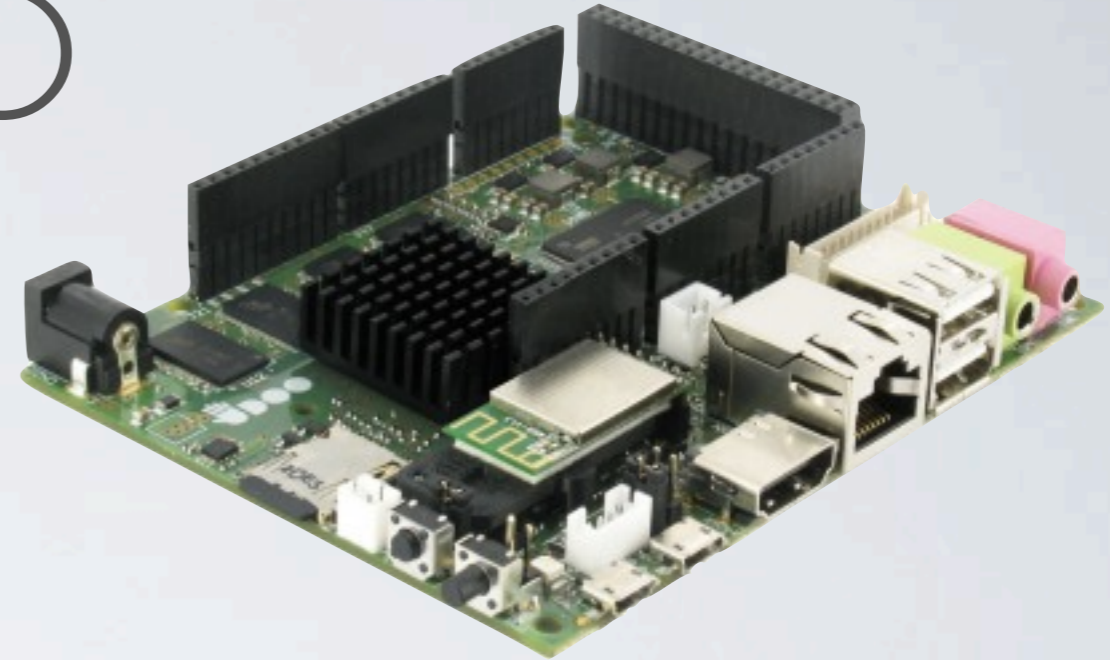
- 82 x 58 x 22 mm

- ODROID-C1+, US\$37.00

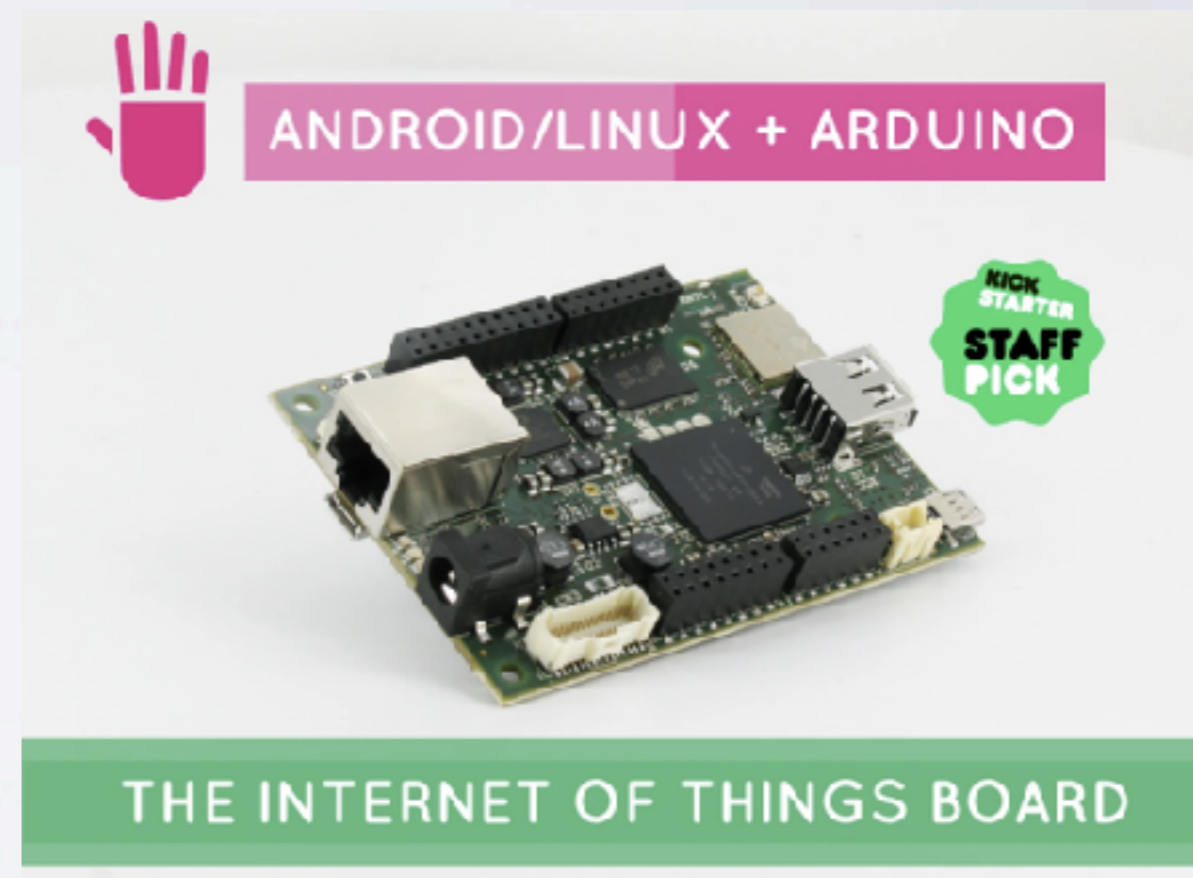
- 85 x 65 x 18 mm



UDOO



- Cartes Seco à base d'iMx6, openHardware
- Dual (\$115), Quad (\$135), Neo (SoloX)
 - RAM DDR3 1GB
 - 76 fully available GPIO: 62 digital + 14 digital/analog
 - Interfaçage Arduino
 - Dual/Quad : Wi-Fi, Ethernet, 2 OTG, 2 USB host
 - Neo : + BT 4.0 + accéléromètre/magnétomètre
 - pas de flash (uniquement MicroSD)
- Écrans (LVDS) & caméras (CSI) adaptés



L'INFORMATIQUE

- Se servir de langages faciles à monter
 - Python : parfait équilibre entre les fonctionnalités bas niveau et le haut-niveau de programmation
 - Lua : revival de ce langage scripté très performant (autrefois beaucoup utilisé dans les jeux)
 - NodeJS : très performant pour du script
- Se servir de son PC pour faire une première maquette sous Linux
- Réaliser son (premier ?) POC sur du Linux embarqué avec une distribution « large » (Ubuntu, Raspbian, Poky...), facile à instrumenter et aux paquets facilement disponibles et installables
- Ne pas se mettre toutes les embuches dès le début : découper son projet et valider les blocs de faisabilités au fur et à mesure !



PROBLÈMES D'INDUSTRIALISATION

industrialisation matérielle & logicielle

EFFET SMARTPHONE



1990s

- « si je l'ai dans ma poche, c'est que je peux le faire moi-même »
 - (certes, avec \$40m de R&D)



2014

- « je peux certainement l'acheter en prêt-à-intégrer »
 - (certes, envoie un mail à Qualcomm pour voir... #LOL)

LE RÊVE DES INDUSTRIELS COMME DES STARTUPERS

- solution sur étagère prête à l'emploi
- tout intégré
- bouton poussoir
- à pas cher



CAS (MALHEUREUSEMENT) COURANTS

- Utilisation de tablettes du commerce
 - encastrées
 - sur des chantiers, en milieu industriel
- Utilisation d'un mini-PC de bureautique
- Utilisation d'une carte électronique lambda non industrielle (Raspberry Pi, Beagleboard, bidule chinois à base d'Allwinner revendu sur Alibaba...)
- Utilisation de Windows (non CE/Mobile)
- Utilisation d'une distribution Linux pour bureau (Ubuntu, Fedora, Gentoo...)

EXEMPLE



???

Approvisionnement ?

Résistance en milieu industriel
? (vibrations, température,
poussière)

Réactivité ?

Gestion mémoire ?

Uptime ?

Salon du livre 2014, machine
d'impression « intelligente »

AUTRES EXEMPLES

- système embarqué sur des véhicules de chantier : PC avec SSD 20Go pour stocker une Ubuntu (⇒ migration vers du Yocto à 120Mo et CPU Freescale iMx6 automotive)
- cartes PC avec un CPU Intel standard et non pas garanti au moins 5 ans, dans des bornes à plusieurs milliers d'exemplaires
- objet connecté en milieu aquatique : utilisation d'un smartphone Sony sous Android (parce qu'amphibie...)
- utiliser une Cubox achetée sur eBay plutôt qu'une carte industrielle pour un système domotique en extérieur parce que « c'est moins cher » (de 40€...)
- tout un tas de Raspberry Pi, partout : des lasers, des bornes outdoor...

Souhait



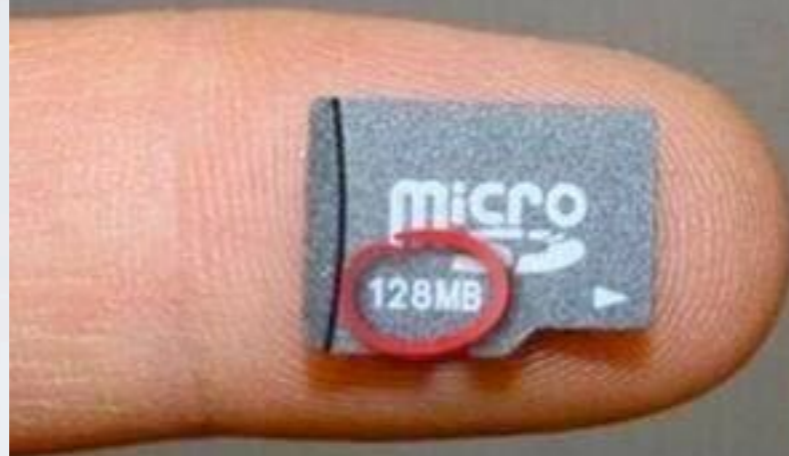
Réalité



POURQUOI ?

- baisse drastique du coût du matériel + arrivée d'Internet et de la vente en ligne
- arrivée salvatrice des cartes-à-pas-cher
 - initié par TI avec la BeagleBoard (mi-2008)
 - avant : cartes de dev très chères et difficiles à obtenir (légère tendance en 2006-2007 à devenir moins cher, vers 450€, sur du Chinois avec support assez lamentable)
 - maintenant : BeagleBone, RPi, Arduino, et tout un tas de petits acteurs dans cette lignée
 - changement de paradigme, *software rules* ! (Et de plus en plus difficile de générer de la valeur sur le seul hardware)
- MAIS effets secondaires
 - perte de vue de la part des ingénieurs, mal formés, des vrais coûts et de la différence entre prototypage et industrialisation !
 - cancer de la RPi : utilisées partout et surtout où il ne faut pas (sous couvert d'aider les amateurs : l'enfer est pavé de bonnes intentions !)

2005



2014



DÉCANDENCE LOGICIELLE

- effets secondaires de la baisse de coût du matériel sur le logiciel
 - pourquoi optimiser quand on a de l'espace à revendre ?
 - pourquoi optimiser quand « ça boote pas si lentement, au pire on prendra un CPU plus rapide » ?
- montée en puissance de Linux dans une industrie qui peine encore à comprendre le code en C sur du PIC
- renouveau des systèmes lents et lourds, des temps de boot infinis, des bugs à répétition sur l'interface...
 - téléphones portables & tablettes
 - set top box
 - télévisions



Un système { Android, Windows, iOS }

EFFET IKEA®

- Mis en évidence par Michael I. Norton, Daniel Mochon et Dan Ariely en 2011
- Surévaluation du travail fait manuellement avec beaucoup de peine, par rapport à un produit fini objectivement supérieur
- "When labor leads to love"
- Plus généralement : lutter contre les dissonances cognitives !

SAVOIR QUAND SOUS-TRAITER

- il est important de connaître ses propres limites d'expertise technique et de savoir piloter d'autres experts de disciplines complémentaires !
- *« Le Maître dit : savoir qu'on sait quand on sait, et savoir qu'on ne sait pas quand on ne sait pas, c'est là la vraie connaissance. »*

(Confucius, *Entretiens avec ses disciplines*, II-17, VI^{ème} siècle avant JC)

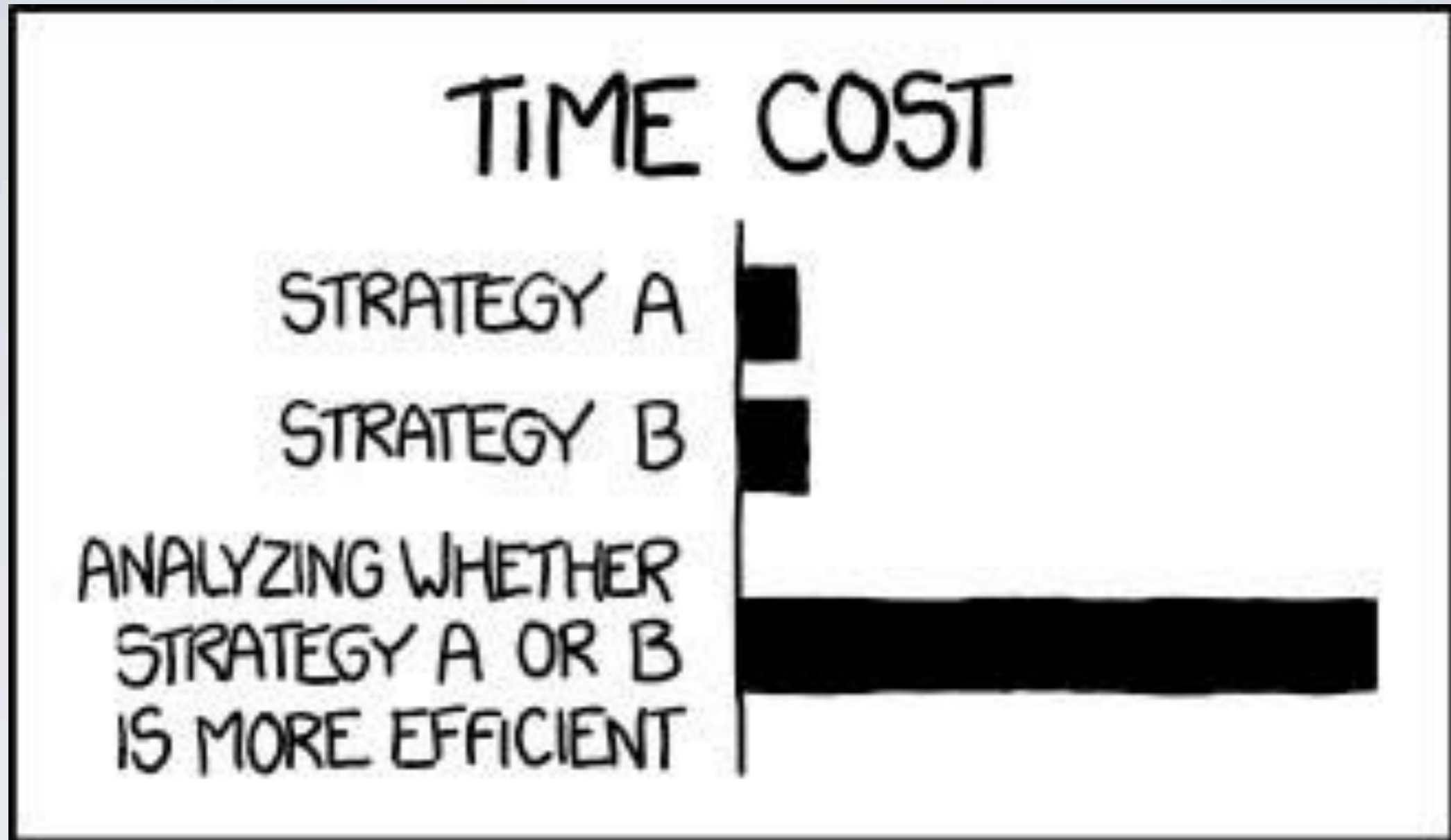
- Si vous pensez que faire appel à professionnel coûte cher, attendez de voir ce que vous coûtera un amateur !

(Anonyme, XXI^{ème} siècle après JC)

THERE IS ALWAYS SOMEONE...



... WHO WILL DO IT CHEAPER!



BON À SAVOIR

prérequis pour comprendre de quoi on parle

VOCABULAIRE BARBARE

- **NRE** (Non-recurring engineering) : coût de dev unique au projet, à amortir sur le nombre de pièces (compte la R&D, le proto, les tests...)
- **coûts de production** : coûts pour maintenir la production du produit
- **TCO** : Total Cost Ownership (coût total de la solution, incluant ABSOLUMENT TOUT)
- **TTM** : Time To Market
- **opportunity cost**/coût d'opportunité : perte des biens auxquels on renonce lorsqu'on procède à un choix (ou coût de la meilleure autre option non-réalisée)
- **willingness to pay**/propension à payer (prix d'acceptabilité) : prix qu'est prêt à payer un consommateur pour un bien
- **BOM** (Bill of Material) : la liste des éléments électroniques nécessaires (composants) et leurs prix unitaires associés
- **0A, 0B** : premier et deuxième prototype matériel (carte électronique, avec plus ou moins de straps dessus pour corriger les erreurs)
- **BSP** (Board support package) : la distribution logicielle très basique de support du matériel
- **POC** (Proof of Concept) : assemblage plus ou moins branlant montrant la faisabilité du projet
- **MVP** (Minimum Viable Product) : produit minimum pour être utilisé dans le cadre de la mise en production (concept plutôt logiciel)

IDÉES DE GRANDEUR

- multiplication par 3 entre le coût de sortie usine et le coût consommateur (*rule of thumb*) : logistique + coût interne fabricant marge fabricant + coût interne revendeur + marge revendeur + taxes
- effet de levier important entre la BOM et le prix final de production (1€ sur un CPU peut engendrer un delta de 10€ parce qu'il faut connecter plus de PINs, avec des composants différents en sortie, parce que la consommation électrique du circuit augmente, etc.)
- le coût total de la BOM dépend fortement du nombre de pièces
- une fois l'appro déterminé, difficile de renégocier les prix par rapport au nombre de pièces (il faut refaire la BOM/le circuit : ne scale pas !)

ÉTAPES D'UN PROJET

- ▶ trouver un matériel de POC dimensionné/adapté
- ▶ développer une carte personnalisée
 - ▶ BOM
 - ▶ conception du circuit
 - ▶ routage
 - ▶ OA & test matériel
 - ▶ OB
 - ▶ carte finale
- ▶ intégration mécanique

- ▶ développer sur PC un premier jet du logiciel métier (la valeur ajoutée)
- ▶ développer/porter sur le matériel de POC
- ▶ développer le logiciel métier (jusqu'au MVP ?) sur la PF matérielle de POC
- ▶ développer un BSP pour matériel final
- ▶ tester la carte sur le BSP développé
- ▶ adapter, intégrer
- ▶ livrer, itérer

tests de bout en bout sur petite série puis production

BOM

- Bill Of Material
- Liste des composants nécessaires + prix associé
- Le prix dépend en réalité de la quantité
 - certains prix sur catalogue, d'autres... un peu aléatoires
 - une fois quotés, les prix des composants sont difficiles à faire évoluer, à moins de refaire le projet
- Le CPU/MCU et l'écran sont généralement les plus chers (mais < \$10 généralement)
 - un petit écart de prix peu néanmoins avoir un effet levier important : plus un CPU/MCU est complexe et plus il tire de composants
 - alourdit la BOM et complexifie le routage
 - attention au PIN MUX entre composants presque équivalents... mais pas tout à fait
- Attention aux composants « piège » : chers, complexes à router, nécessitant une/des pull down (augmente la consommation électrique du circuit), voire trop haut (complexité mécanique de placement)

CHAUSSES-TRAPPES

- routage : entre CPU et RAM, nombre de couches et rayonnement, erreurs sur les broches, straps
- alimentation : instable, insuffisante, pull-ups sur composants en sortie...
- intégration mécanique : taille/forme du PCB, hauteur des composants, dissipation thermique...
- matériel de POC trop éloigné du matériel final
 - perte de temps sur le BSP (portage)
 - pré-validation potentiellement erronée (vitesse de CPU, des I/O, tailles mémoires flash & RAM)
- drivers
 - fermés : incompatibilités entre eux, impossibilité de les faire évoluer
 - insuffisants/incomplets : grosse perte de temps pour un support suffisant
- usage de plus en plus courant d'eMMC (dû à la très mauvaise optimisation du logiciel utilisé...)

LES CLASSIQUES OUBLIÉS

- la trésorerie d'avance
 - payer le matériel d'avance à l'usine
 - délais de paiements
 - délais de production (attente de tous les composants — dont les fournisseurs peuvent temporiser pour optimiser leurs prix)
 - fluctuation du prix à l'unité (monnaie, probabilité de pannes)
 - gérer le stock
- la logistique
 - chargement initial de l'application
 - le transport
 - la douane
- la redistribution
- les marquages et/ou certifications
 - par exemple : CE pour électronique grand public, « grand-e » pour l'automobile, « petit-e » pour le ferroviaire, DO-254/DO-178 pour l'avionique
 - auto-déclaratif, sauf pour du très spécifique type transport/médical (certifications), MAIS la concurrence a tout intérêt à dénoncer une non-conformité
 - le plus important : la CEM (Compatibilité électromagnétique)
 - pour les certifs : les taux de pannes/MTBF
 - assemblage d'OEM CE n'est pas CE : refaire systématiquement les essais de certification
 - les fabricants OEM font (forcément) des tests dans les conditions qui les arrangent : toujours revérifier a minima (test d'investigation), ne pas faire confiance...
 - coût : ~50K€, 1500€/jour × 15j si pas de problème sur produit fini — n'arrive jamais si radio, si compliqué...

LES CLASSIQUES OUBLIÉS (2)

- le workflow des usages et la connectivité (« objet connecté »... à quoi, comment ?)
- la maintenance
 - les mises à jour (correctifs, sécurité, évolution)
 - le SAV
 - l'évolution du produit
- le watchdog
- l'obsolescence matérielle & les problèmes d'appro
- les farces chinoises (changement de composant électronique sans rien dire, incompréhension des process, délais en tout genre...)
- le nouvel an chinois

MVP

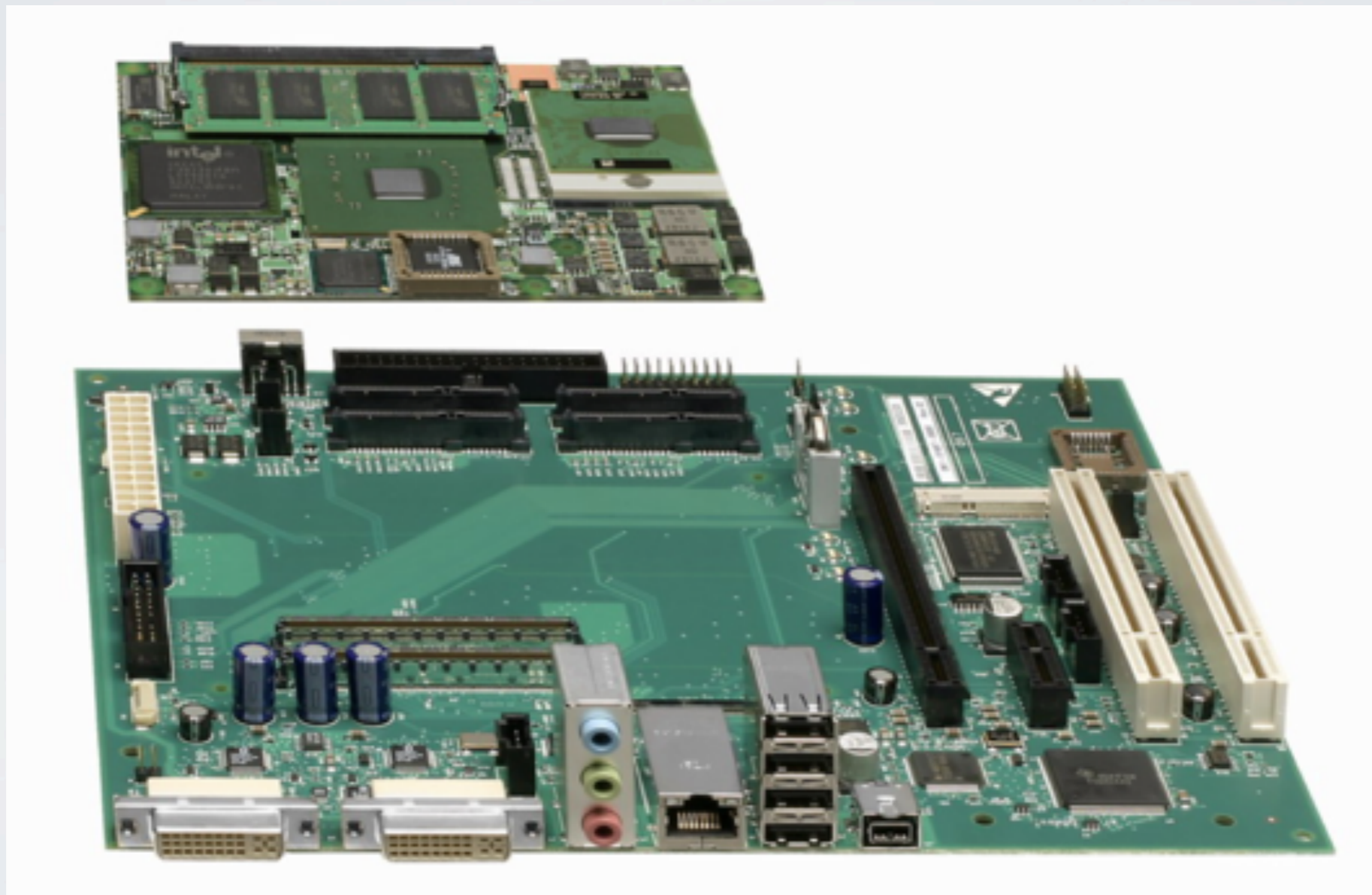
- Minimum Viable Product : Produit minimum viable, moment où il y a juste assez pour l'utilisateur, mais pas encore bien sec (après l'étude de faisabilité)
- Popularisé par Eric Ries dans son ouvrage de référence « Lean startup », pour les applications web
- Peut s'appliquer (assez mal) à la mécanique et l'électronique (utilisation d'imprimante 3D & de matériel POC, voire un OA), mais surtout à l'informatique (avec mise à jour distante ou rapatriement pour maj...)
- Ballon d'essai pour tester les fonctionnalités auprès des utilisateurs et des investisseurs, avant de lancer la grosse machine d'industrialisation (post-levée de fonds)
 - encore temps de se tromper pour corriger le tir
 - l'industrialisation coûte cher et est très peu flexible
 - que se passe-t-il si des fonctionnalités sont finalement attendues et absolument nécessaires pour l'utilisateur, et impactent le circuit électronique ?...
- Règle : on doit toujours avoir un peu honte de son MVP (sinon, c'est qu'on est allé trop loin)



SOLUTIONS MATÉRIELLES

où l'on va parler silicium

LE CONCEPT DU MODULE



AVANTAGES TECHNIQUES DU MODULE

- I/O sur broches (« pinoche »)
- BSP / OS validés par le fabricant
- Ecosystème mature
- Meilleure fiabilité du design par réduction/absence de câbles
- Différentes gammes d'un même module en fonction de la T°/résistance milieu industriel
- Architecture modulaire & évolutive
- Maîtrise du cycle de vie produit à travers la maîtrise de la carte porteuse et de la connectivité applicative
- Contraintes restantes
 - carte porteuse à développer
 - intégrer les contraintes thermiques dès l'origine du design...

AVANTAGES BUSINESS DU MODULE

- Développement d'un système *full custom* sans avoir à supporter les coûts et le temps de développement de la fonction CPU (généralement, from scratch : ~1 million \$, ~1 an)
- Réduction du TTM
 - disponibilité immédiate, même des derniers CPU !
 - validation de la fonction sur le module
- Possibilité de concentrer ses investissements sur sa couche métier
- L'amortissement nécessite un quantitatif adapté au business model
 - rentable jusqu'à ~10.000 pièces
 - facilité l'évolution du produit
 - flexible (coûts en fonction de la taille de la commande)
- NRE = conception de la carte porteuse personnalisée (souvent à partir d'une porteuse existante)
- matériel de POC = carte de dev accueillant le même module, tirant beaucoup d'I/O donc assez proche du matériel final

FONCTION CPU SUR MODULE

- Carte de petite taille contenant CPU, RAM et Flash
- Autres fonctionnalités du CPU (UART, I2C, SPI, GPIO, PCIe, LCD...) sur les broches
- 2 grandes familles
 - COM
 - facteur de forme et pinoche standard (plusieurs)
 - interchangeable, pas de dépendance à un fournisseur
 - mais contraint par le plus petit commun dénominateur
 - module propriétaire
 - plus de possibilités de pins I/O, de facteurs de forme réduits
 - mais potentielle dépendance au fournisseur

LE COMPUTER ON MODULE



- Pour i3, i5, i7
- Facteur de forme
 - "Ultra" 84x55 mm – 46 cm²
 - "Compact" 95x95 mm – 91 cm²
 - "Standard" 95x125 mm – 118 cm²
 - "Extended" 110x155 mm – 170 cm²
- PCI express (16x), SATA
- Mais complexe, alimentation 12V



LE COMPUTER ON MODULE



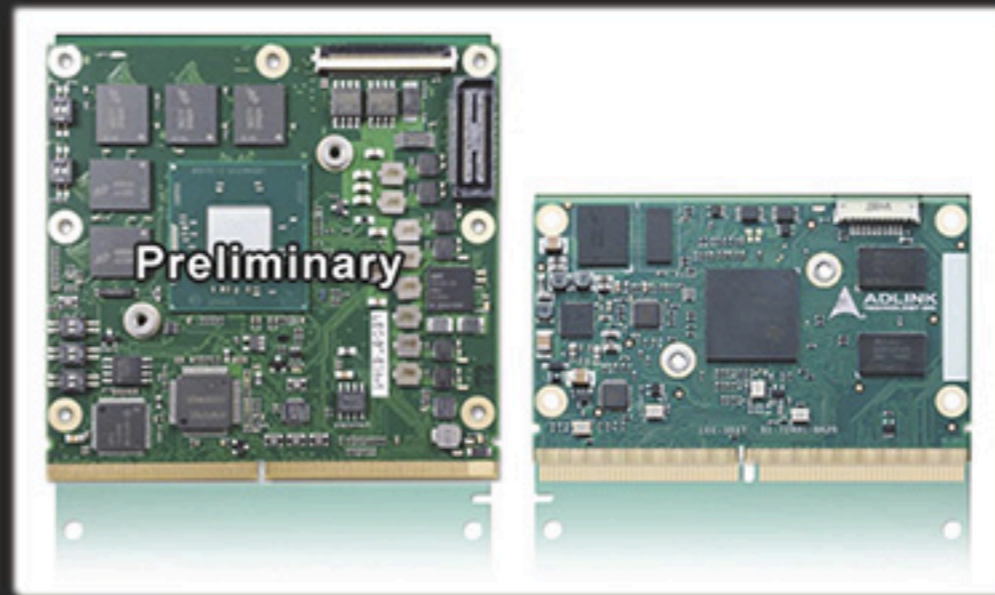
- Atom (x86), FreeScale i.MX6 (ARM)
- Facteur de forme 70x70 mm (70x40 mm pour le μ QSeven)
- Alimentation 5V : porteuse plus simple (USB...) et autonomie
- Solution de refroidissement intégrée dans le design module
- Mais pas de support des CPU Intel puissants



SMARC

- SMARC = Smart Mobility ARChitecture
- 82 mm × 50 mm pour du basse énergie compact
- 82 mm × 80 mm pour hautes performances et refroidissement

SMARC in a nutshell



314-pin MXM 3.0 connector

Functions on connector

24-bit RGB
24-bit LVDS
HDMI
Displayport
PCIe
SATA
USB
USB OTG
Gigabit LAN
SDIO/eMMC
Camera IN (Par / Ser)
UART
CAN bus
SPI
i2C / I2S
GPIO

SODIMM

- Facteur de forme 67,4x25,1 mm
- Jusqu'à 200 broches
- Similaire au format de DDR RAM des PC portables



Freescale™ i.MX28 (ARM9)



i.MX6 (A9, Single, Dual ou Quad)



Freescale™ Vybrid
(Cortex™-A5 + Cortex™-M4)

AUTRES MODULES CPU (SOM) PROPRIÉTAIRES (EXEMPLES)



- facteur de forme/pinochage Tamonten™ (nouvelle génération compatible)
- SO-Dimm double hauteur (67.6 mm x 55 mm)
- NVidia Tegra pour calculs intensifs
 - ARM® Cortex™-A9 MPCore™
 - FPU : ARM® NEON™
 - quad core (+ un low power companion core)
 - 4x up to 1.4GHz
 - powerful 3D embedded GPU on-die



- Kurz Industrie-Elektronik GmbH EPC35 CPU Modul
- facteur de forme μ QSeven MAIS pinochage propre (pour sortir les signaux temps-réel)
- ARM TI Sitara AM335x SoC
 - 275 MHz ... 800 MHz (Cortex-A8)
 - Coprocesseur NEON
 - PRU (programmable realtime unit)
 - DDR3-SDRAM, max. 1 Go



- Technexion Pico-IMx7-eMMC
 - NXP i.MX7 Solo / Dual
 - ARM Cortex-A7 single core @800Mhz + Cortex-M4
 - ARM Cortex-A7 dual core @ 1Ghz + Cortex-M4
 - Up to 2GB LPDDR3, onboard eMMC (default 4GB)
 - 9x GPIO, 4x PWM, 2x I²C, 1x I²S, 1x SPI, 2x UART, USB-OTG, SDIO (4-bit), 24-bit TTL RGB, MIPI CSI Camera, MIPI DSI Display, PCIe, RGMII (gigabit LAN), Flex CAN version 2.0B Compliant
- 36 x 40 mm, 8 grams
- Certification Compliant with CE, FCC, RoHS, REACH directives

SBC

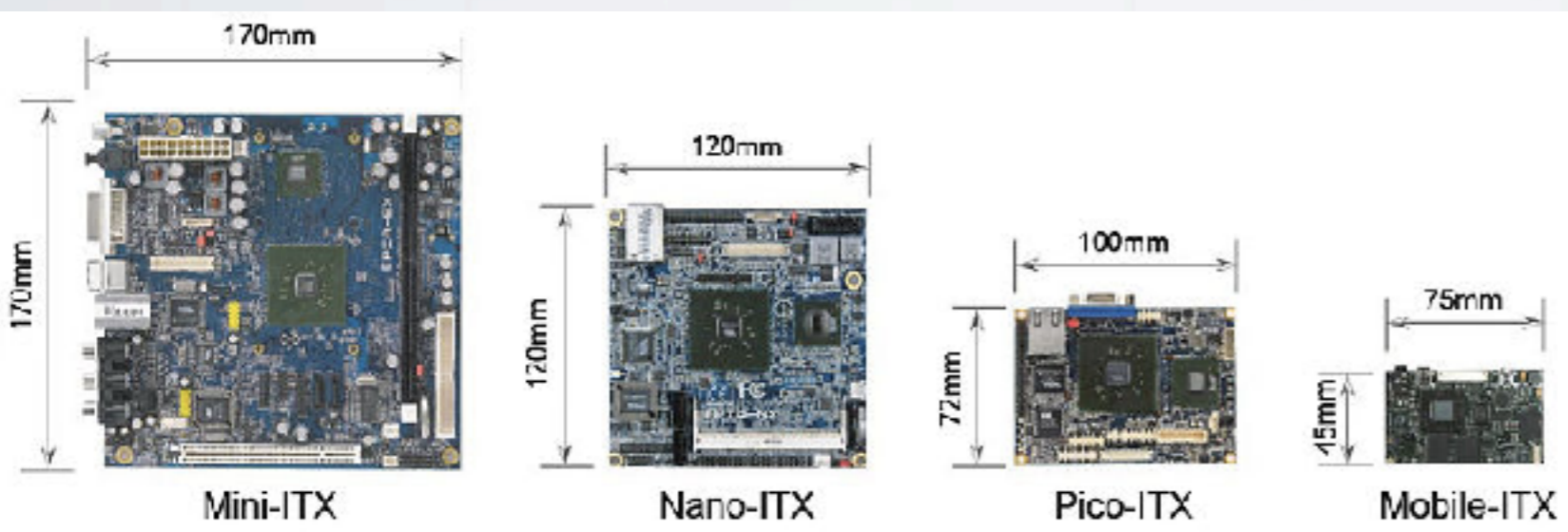
- Single board computer
- standards Mini-ITX, Nano-ITX, Pico-ITX, Mobile-ITX
- Atom, iMx6...
- Aussi des formats propriétaires



iWave i.MX6 Pico ITX SBC

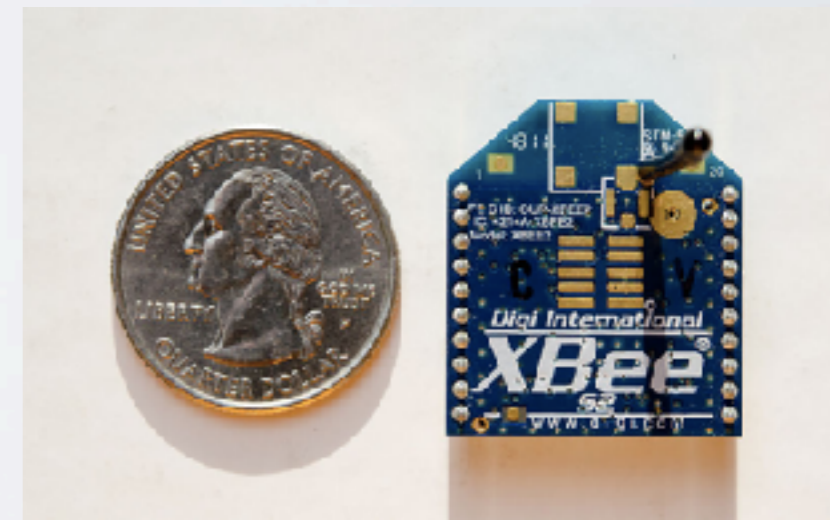
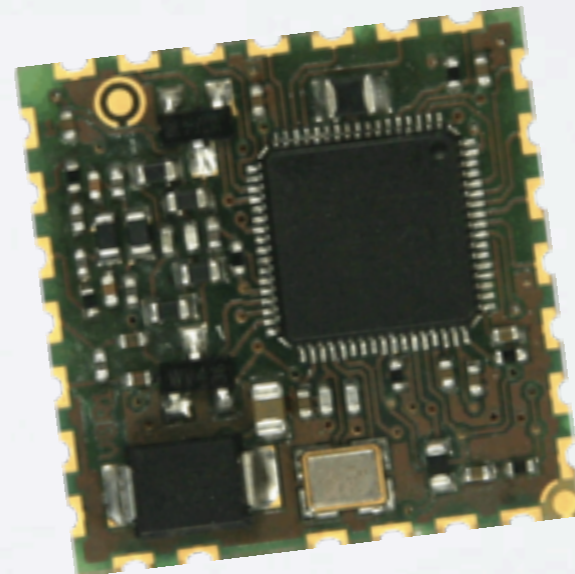


SECO μ SBC-i.MX6
(80 x 67 mm)



MODULES DE CONNECTIVITÉ

- À connecter en UART, SPI...
- PLC : HomePlug GreenPhy & G3
- 4G LTE
- Zigbee
- ...



FAIRE (FAIRE) SON PCB

- Printed Circuit Board (« sa carte », en bon français)
- deux cas :
 - petit embarqué : le COM est trop gros/trop consommateur de courant/pas adapté
 - > 10K pièces : problème de riche (et de chinoiserie à venir)
- deux cas à éviter :
 - prendre une solution made in Alibaba sur étagère : aucune maîtrise, durée de vie faible, nécessité de faire des stocks sur une solution peu éprouvée (quelle résistance dans le temps ?)
 - exploser son NRE en faisant un PCB trop complexe quand un COM suffirait largement : biais très français !!

MADE IN FRANCE

- peu de manufacturiers (désindustrialisation de la France...)
 - exemples : Eolane, BSE, Asteel, Altrics
 - d'autres tous petits acteurs pour les petites séries/les protos et maintenant www.petiteserieelectronique.com (service Altrics)
 - difficiles à trouver (très peu de comm'...)
- avantages :
 - circuits courts d'approvisionnement
 - maîtrise
 - évite les marquages CE (en B2B du moins)
- inconvénients :
 - prix unitaire bien plus cher !
 - certaines technos inaccessibles (par exemple concevoir une tablette de moins d'un cm d'épaisseur...)
 - pas tous ne savent souder du CMS (Composants montés en surface), nécessite de lourds investissements
 - entre-deux : fabriquer certaines pièces à l'étranger (Chine ou autres) et faire l'assemblage (et potentiellement d'autres éléments) en France ; choisir une usine qui a aussi une implantation aux US/Afrique du Nord/Europe de l'Est

ATTENTION...

- approvisionnement de composants
 - accords avec partenaires habituels
 - minimiser les composants étranges
 - parfois les apporter soi-même (accords directs avec les fournisseurs)
- délais de livraison des composants
- pas de réglage
- automatisation assemblage & test (pas d'opération manuelle en France ! Trop cher !!)
- actuellement, usines un peu à vide, prennent de la petite série (pour faire tourner les chaînes)

MADE IN CHINA

- essentiellement la région de Shenzhen
- énormément de manufacturiers & ODM
 - des (très, très) gros : Foxconn (taïwanais, 1,3M employés), Pegatron (taïwanais, 104K employés) ZTE (85K employés), T&W (10K employés)...
 - une myriade de sous-traitants sur place, plus ou moins spécialisés dans des champs (télécom, STB, tablettes, téléphones...)
 - dur de s'y retrouver et de trouver le bon partenaire !
- avantage :
 - beaucoup moins en production à l'unité, pour des volumes impossibles à atteindre en France (aucune chaîne de production suffisante)
 - des technos de miniaturisation inaccessibles en France
 - beaucoup plus d'experts en électronique !
- inconvénients :
 - communication complexe : décalage horaire, anglais chinois et... culture très différente !
 - travail électronique d'une qualité à la hauteur du coût
 - logiciel dégueulasse
 - avoir une personne (de préférence chinoise) sur place pour suivre la production
 - problèmes de trésors (tout payable d'avance en dollars + taxes d'importation à chaque arrivage)
 - délais de livraison (avion rapide mais cher, cargo lents mais prix dérisoire — 2 à 3 semaines)
 - risques de piratage !
- entre-deux : choisir un OEM et garantir la durée de vie d'un projet compatible sur étagère, ou le faire adapter

MADE IN AILLEURS

- USA
 - parlent anglais
 - concepteurs qui ont inondé le marché des petites cartes
 - CircuitCo : BeagleBoard, Minnowboard
- Italie / Allemagne
 - pas de marquage CE obligatoire (reste dans l'espace européen)
 - proche
 - moins de barrière de langage
- offshore
 - moins cher, intéressant pour les produits avec assemblage ou pour monter en volume

NOUVEAUX MODÈLES (POUR SOFTEUX/COMMERCIAUX)

- prototypage électronique en ligne : www.protoelectronique.com
- sous-traitants faisant du produit de A à Z en marque blanche sur commande
 - OmptimalDesign: design industriel, méca, élec, logiciel, prototypage rapide
 - logiciel chez Room5 (<http://room5.com/>), hardware chez Electric Imp (<https://electricimp.com/>)
 - du design à la logistique en passant par le hard, le soft et la méca, en France : Canon Bretagne
- accélérateurs pour produit
 - sur dossier (concurrence rude), investissement et conception de matériel électronique pour startups + crowdfunding + formation
 - Hax Accelerator : injection 25K\$ contre 6% du capital ou 100K contre 9%, 111 jours, Shenzhen
 - Chinaccelerator : 30K\$, 7% du capital, 3 mois à Shanghai
 - Y combinator (US)



SOLUTIONS LOGICIELLES

du côté de chez nous

COMPLEXITÉ LOGICIELLE

- de plus en plus de code, de plus en plus de couches
- complexité des fonctionnalités \Rightarrow complexité du logiciel
- interdépendances
- problématiques de pérennité et de maîtrise du code !
- problématiques d'architecture firmware à penser correctement dès le début



30% des industriels ont déclaré en 2011 vouloir migrer vers Android.

Pour le moment, rien ne s'est passé.
Linux embarqué résiste. Est-ce vraiment étonnant ?

#EPICFAILS

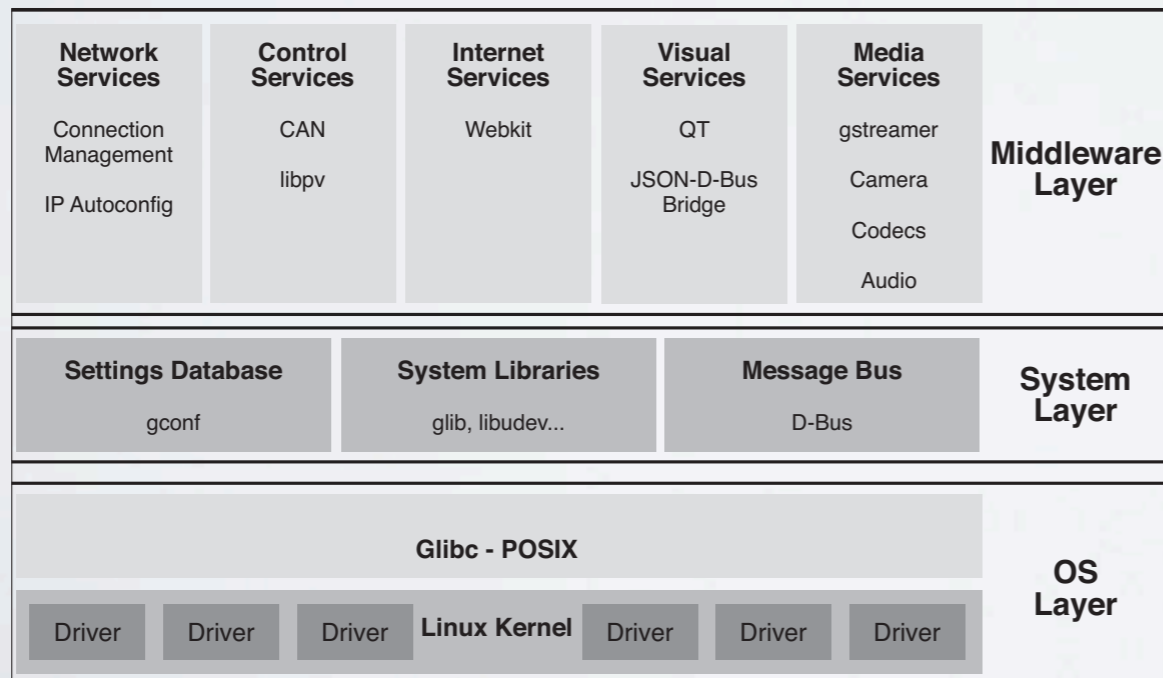
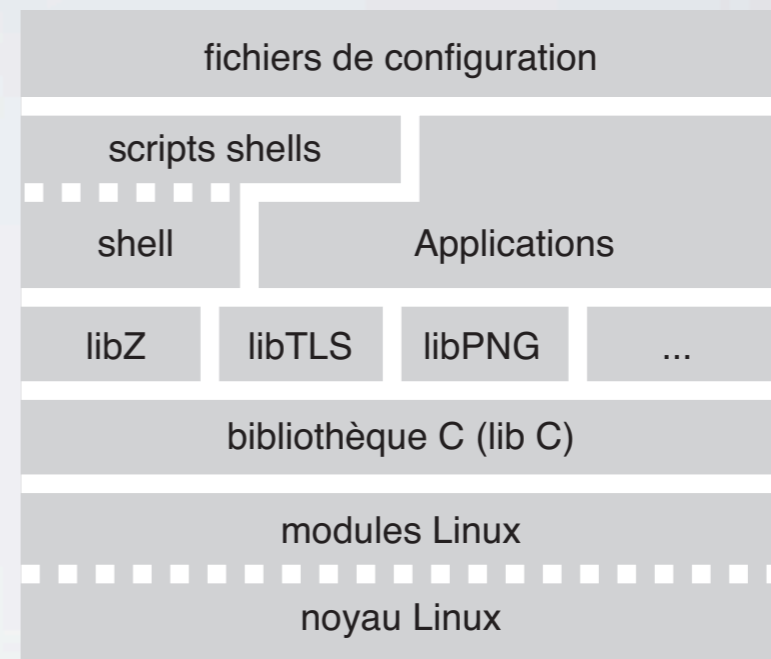
- Toyota's killer firmware: Bad design and its consequences (2013)
<http://www.edn.com/design/automotive/4423428/Toyota-s-killer-firmware--Bad-design-and-its-consequences>
(stack overflow, buffer overflow, unsafe casting, and race conditions between tasks)
- redémarrer son Boeing 787 avant 248 jours pour ne pas risquer une coupure d'alimentation électrique (en vol)
<http://www.slate.fr/story/101101/bug-boeing-787-courant>
- crash A400M : « logiciel de contrôle des moteurs mal installé à l'usine d'assemblage »

L'APPROCHE MODULAIRE

- Penser l'architecture de son système/code par modules
- Chaque module correspond à une fonctionnalité importante définie (périmètre fonctionnel)
- Les modules d'applicatifs haut-niveau (interfaces utilisateur/middleware) communiquent entre eux par un bus logiciel
- Modules *précisément interdépendants* (le moins possible) : la chute de l'un n'entraîne pas l'ensemble ! (Notion de résilience)

ARCHITECTURES PAR BRIQUES

Système Linux embarqué « standard »



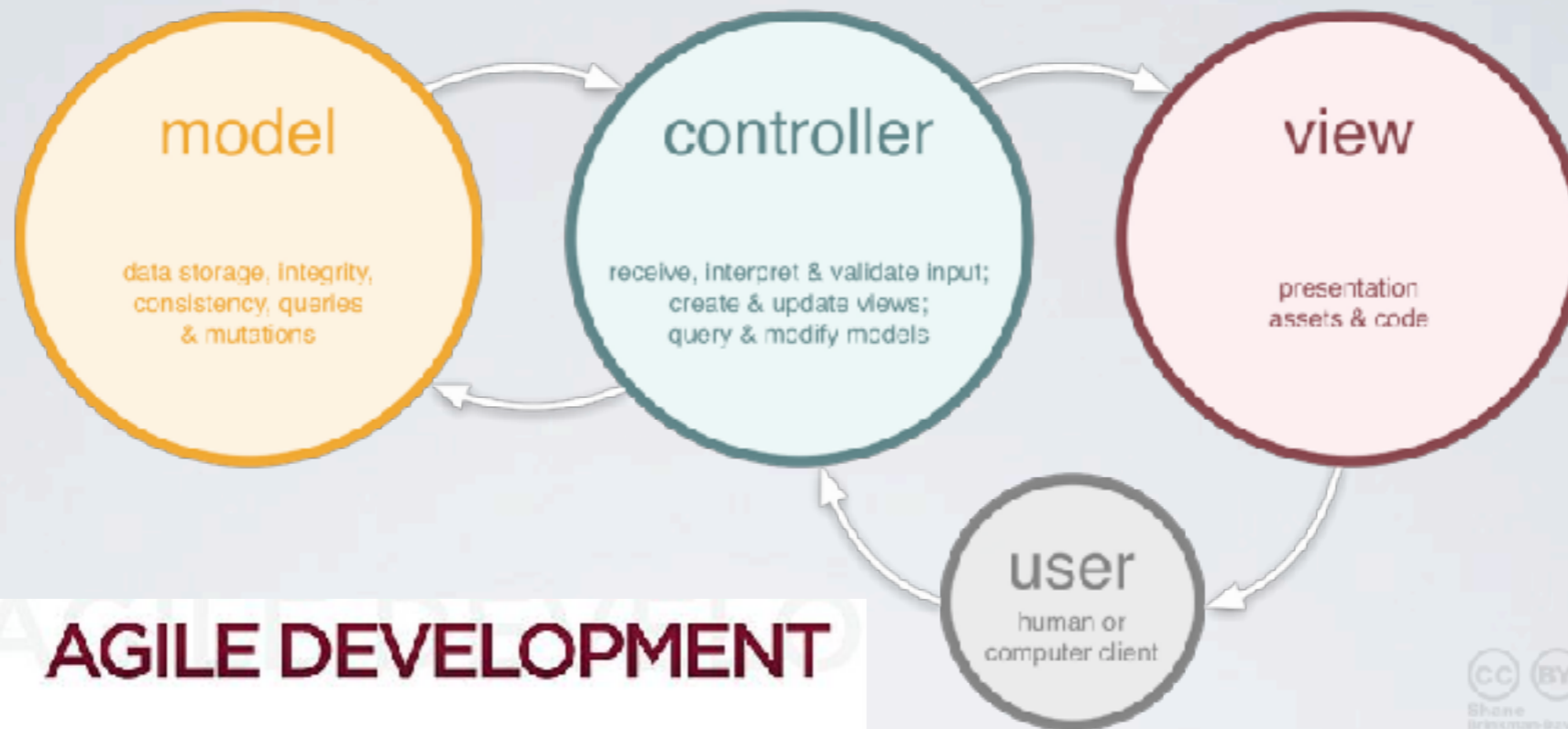
BEST PRACTICES

- dédier les threads à l'exploitation du multicoeur
- ne pas créer de thread à la place d'un processus par (fausse) facilité
- ne pas mélanger des fonctionnalités critiques et des fonctionnalités mineures (par exemple la consultation par affichage)
- créer des APIs (interfaces de communication) précises, évolutives, rétro-compatibles
- watchdogs, auto-diagnostic de pannes et auto-réparation

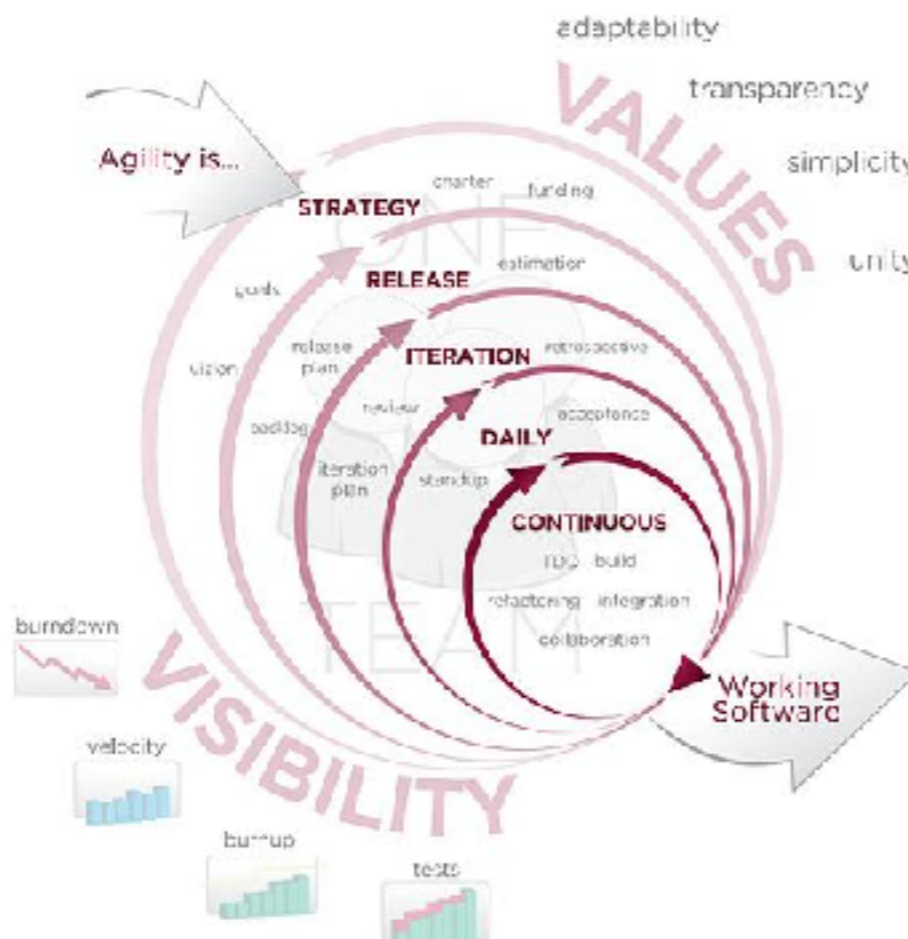


MENER SON PROJET PAR MODULARITÉ

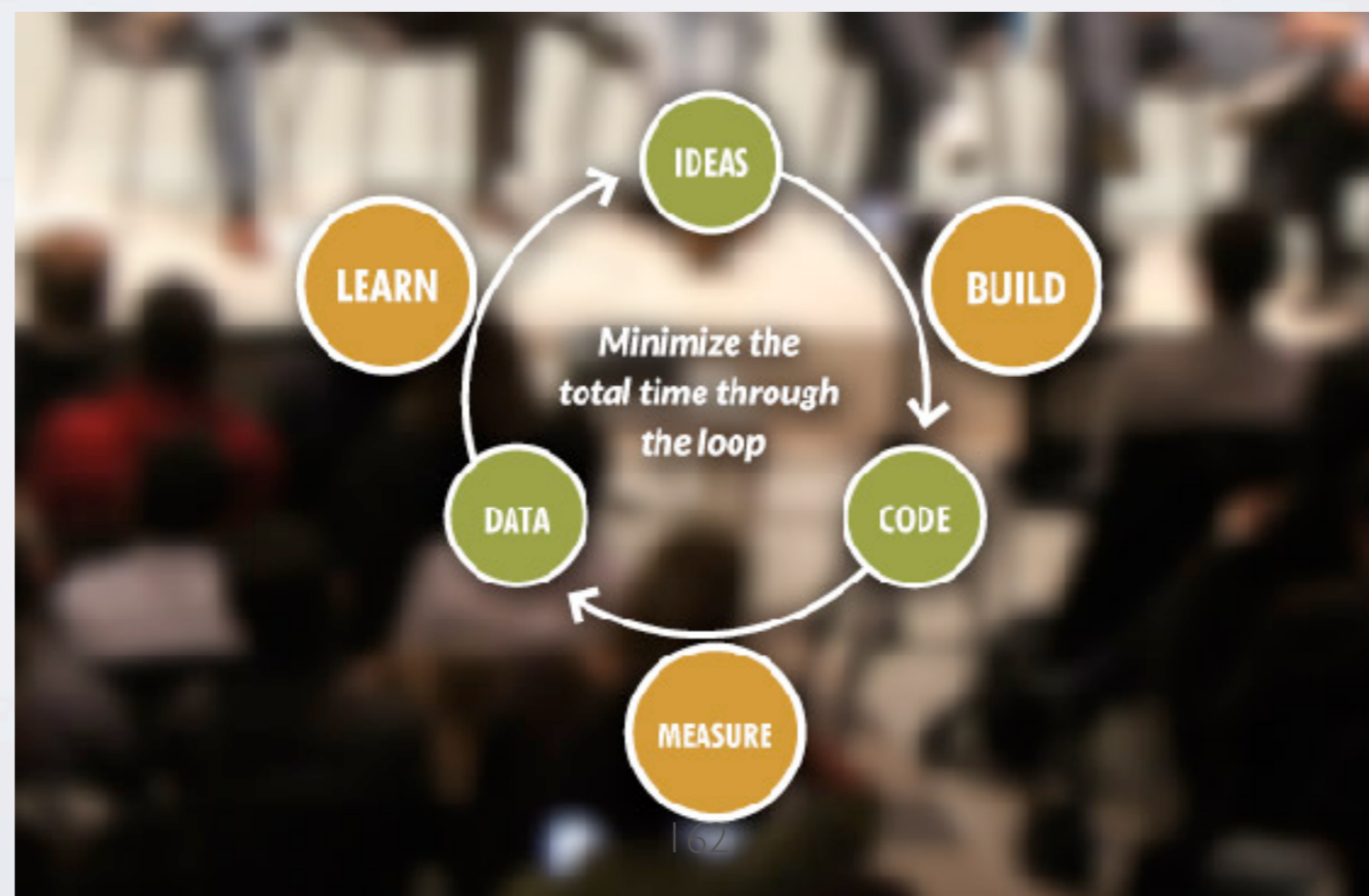
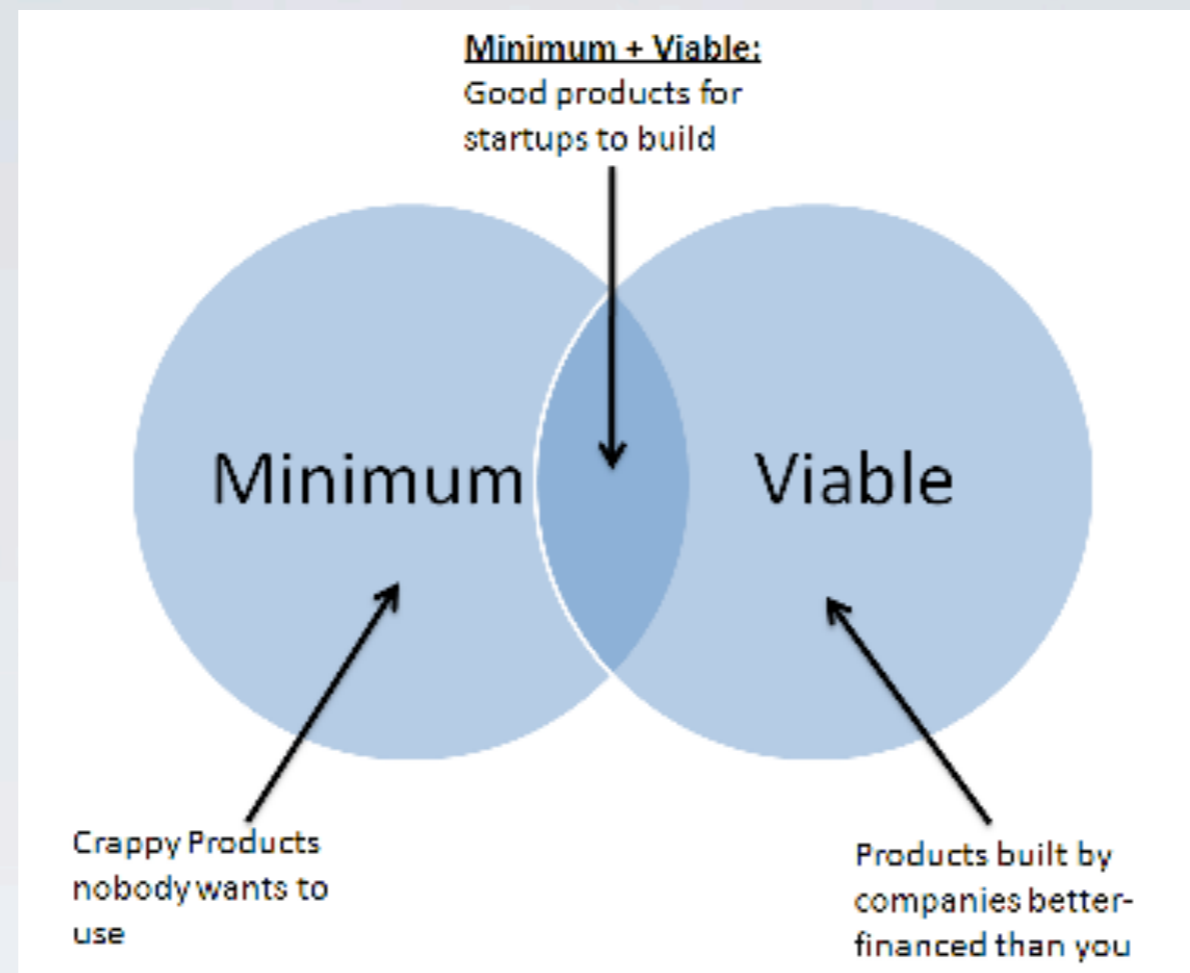
- La modularité permet de faire évoluer en parallèle les fonctionnalités de son produit
- À rapprocher aux nouveaux mouvements de conception logicielle :
 - approche MVC : séparer l'accès aux données, la couche graphique et l'algorithmie
 - management Agile : développement par cycles
 - lean startup : produit minimum viable et itération
 - software craftsmanship : concentration sur l'aspect artisanal de conception du (beau) code

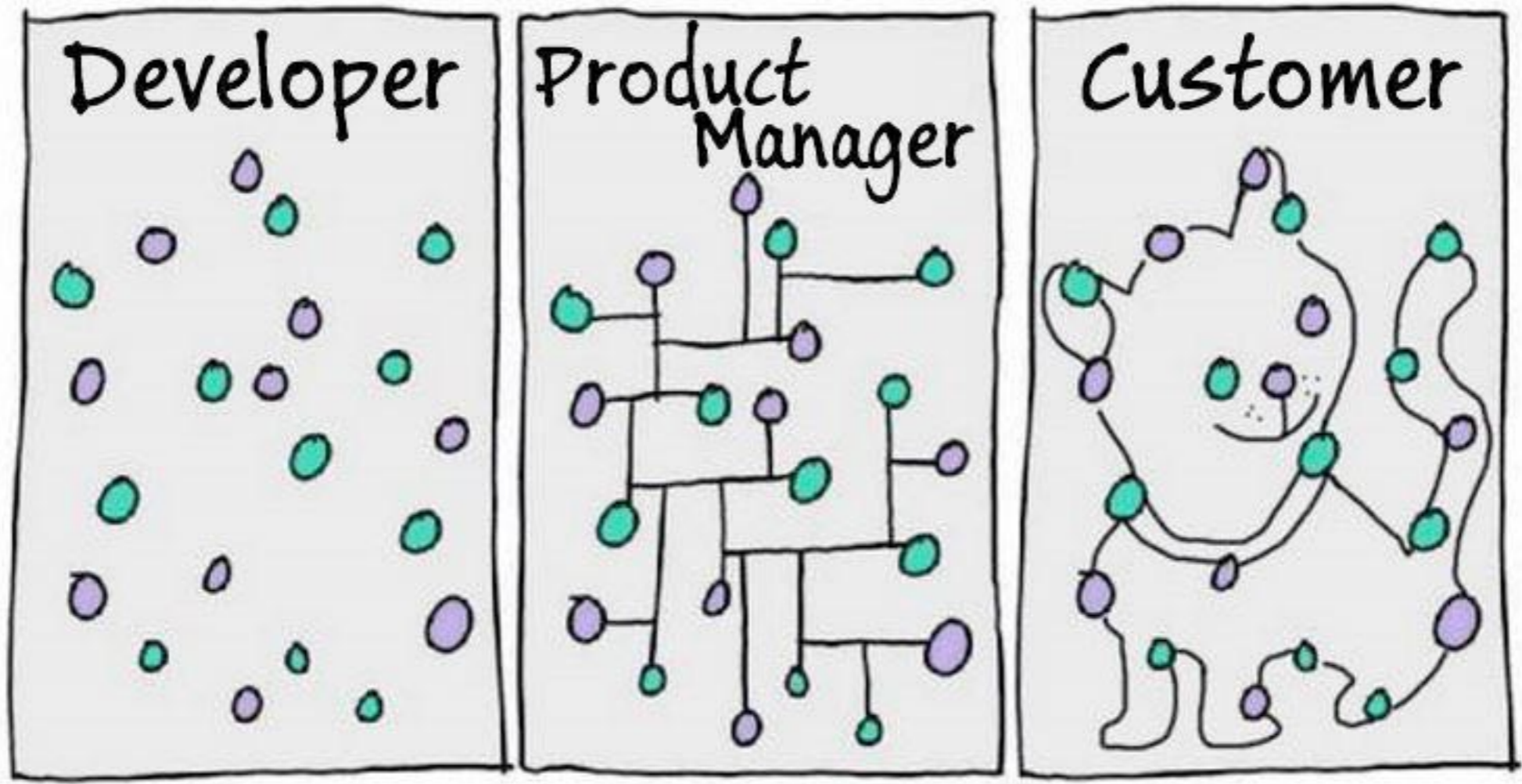


AGILE DEVELOPMENT



ACCELERATE DELIVERY





LA MAÎTRISE

- qu'est-ce que maîtriser son logiciel ?
 - disposer des sources
 - disposer des connaissances pour étudier les sources
 - maîtriser les processus de compilation, de codage et d'intégration
- la maîtrise du socle logiciel est la fondation solidifiée nécessaire à l'innovation

LE LOGICIEL LIBRE INDUSTRIEL

- Architecture modulaire intrinsèque : origine dans le développement par équipes disjointes éloignées, sur différents projets (notion de « distributions Linux »)
- Choisir des briques éprouvées et pérennes
- Garder le contrôle grâce à l'accès au code source
- Industrialiser sa chaîne de production logicielle (usine logicielle : Yocto)
- Système d'exploitation Linux, couche graphique Qt...

ARCHITECTURE FIRMWARE

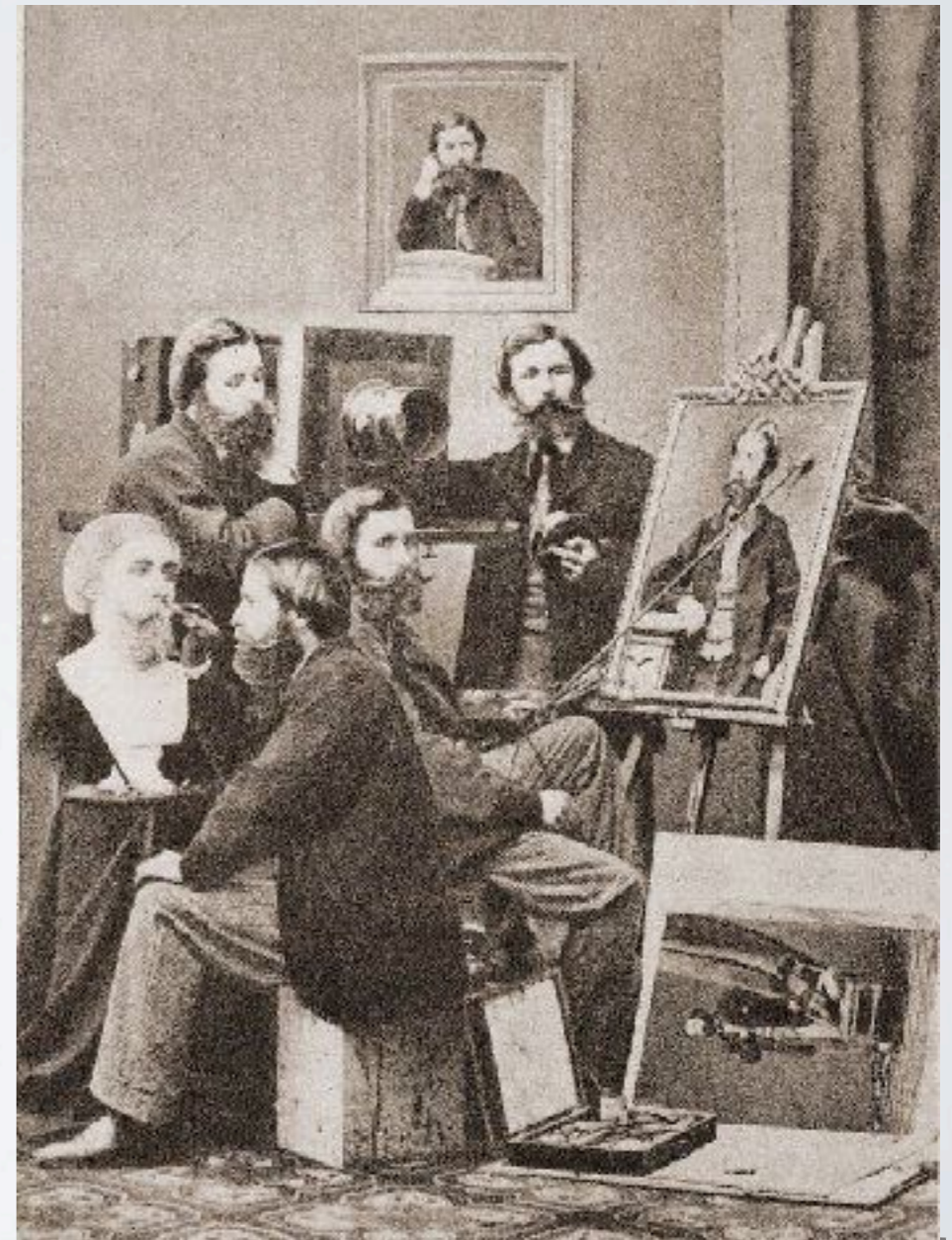
- penser son architecture le plus tôt possible
 - impact sur la consommation de CPU (et donc l'alimentation)
 - impact sur les types et les tailles de mémoire (Flash & RAM)
 - dépendance forte au système de mise à jour choisi, donc à la stabilité du système dans le temps et au SAV
- dans les faits : du ni-fait-ni-à-faire, problématique très souvent négligée au profit de la « pure » technique

POURQUOI INDUSTRIALISER ?

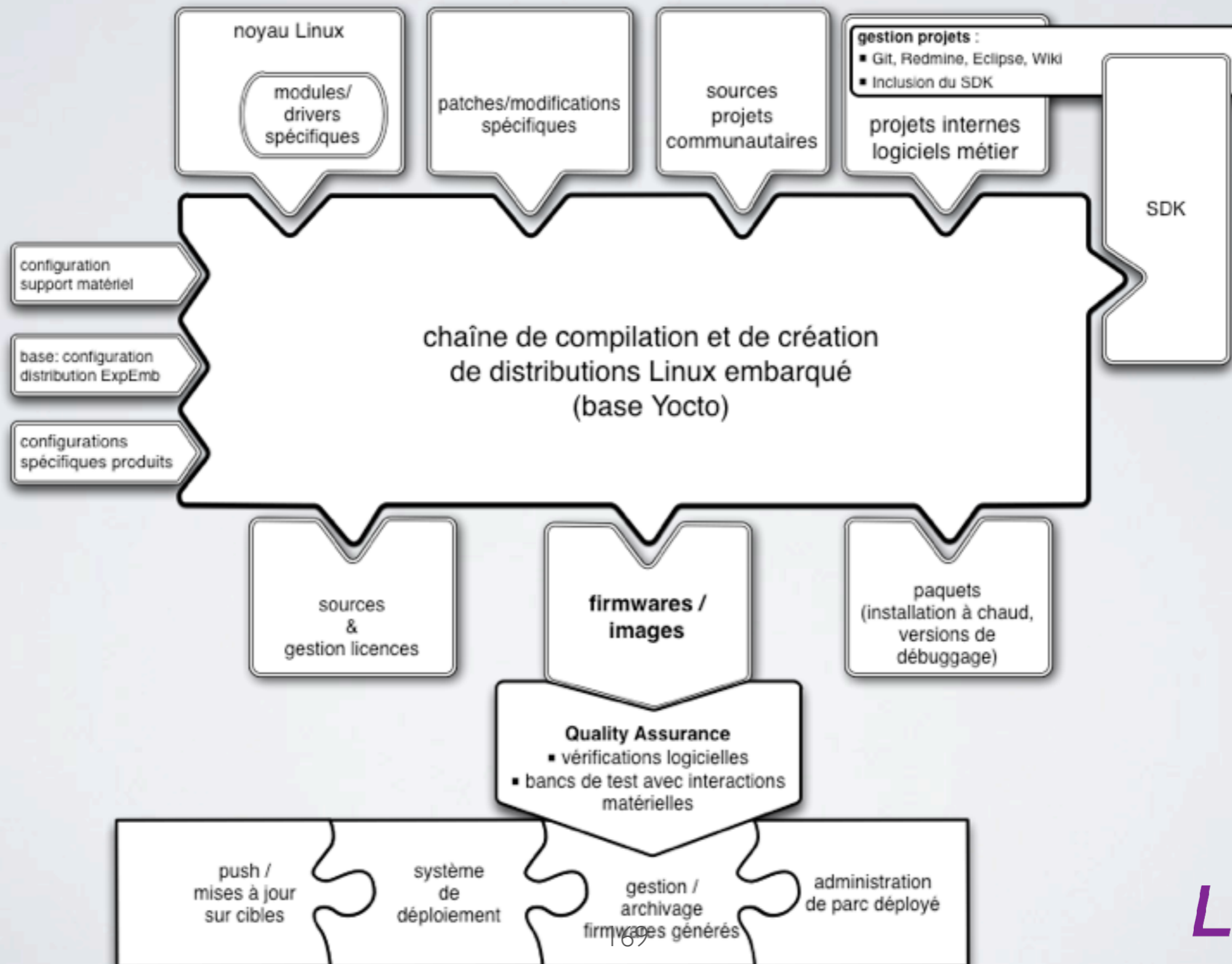
- passer de la phase d'expérimentation/POC à la phase industrielle
- buts :
 - rendre **reproductible**
 - rendre **solide, fiable**
 - assurer la **pérennité** dans le temps
 - **maîtriser** sa chaîne de **processus** de création et d'intégration
- facile à comprendre pour le hardware, plus difficile pour le software

PROBLÈME DE LA REPRODUCTIBILITÉ

- un système non-reproductible n'est pas industrialisé
- nécessité de **maîtriser** les sources ! Pas de boîte noire !
- compilation à partir de zéro, automatisée
- intégration de son propre code source spécifique/propriétaire



LEAN SOFTWARE FACTORY



SÉCURITÉ

- Problème TRÈS embarrassant pour l'IoT : données (très) personnelles (parfois médicales !), contrôle-commande (ouverture de la maison, chauffage, éclairage public), etc.
- Gros refoulement de ces problématiques
 - chiffrement inexistant ou faible, appairage aléatoire, man-in-the-middle, DDOS...
 - on en parle pour ne pas en parler
 - problèmes d'architecture globale, de puissance de calcul, de protocoles choisis, de la complexité du logiciel embarqué
- Utiliser des protocoles solides (6LoWPAN plutôt que Zigbee...), des clés de chiffrement (SSL par exemple)
- Implique probablement une gestion de clés de chiffrement, donc une gestion de type PKI : autorité de certification, récusation, séquestre, durée de vie, remplacement, etc. (très pénible !)
- Ne pas oublier que la sécurité de l'ensemble d'un système est au niveau de son point le plus faible (si l'on utilise des clés symétriques, il suffit d'attaquer le matériel pour dumper la clé... Une seule solution : la puce DRM, avec les implications que cela a pour le matériel à produire et l'initialisation en usine...)

MISES À JOUR

- Importance de la mise à jour pour :
 - les évolutions
 - la maintenance
 - la correction de bugs,
 - la sécurité
 - la configuration générale
- Nécessite une architecture particulière et résiliente (cf cours Linux embarqué)
- Nécessite d'avoir assez de RAM, de Flash, et de bande-passante ! (Assez rare d'avoir les trois à la fois)
- Mettre à jour/maintenir par une intervention physique coûte TRÈS cher, à éviter absolument !



CONCLUSION

- un projet IoT EST un projet industriel/avec une large part d'industrialisation
- un projet industriel n'est pas un long fleuve tranquille
- un projet industriel ne s'improvise pas
- la télé vous ment
- la magie n'existe pas
- une mort douloureuse attend la grande majorité des projets
- il y a des moyens de s'en sortir, mais il faut y penser tôt et a minima être épaulé en permanence par quelqu'un du métier, expérimenté

QUESTIONS



Gilles BLANC — LINACS
<http://linacs-consulting.com>
gilles.blanc@linacs-consulting.com
+33 (0)6 70 33 00 03

