



Architectures des systèmes IoT

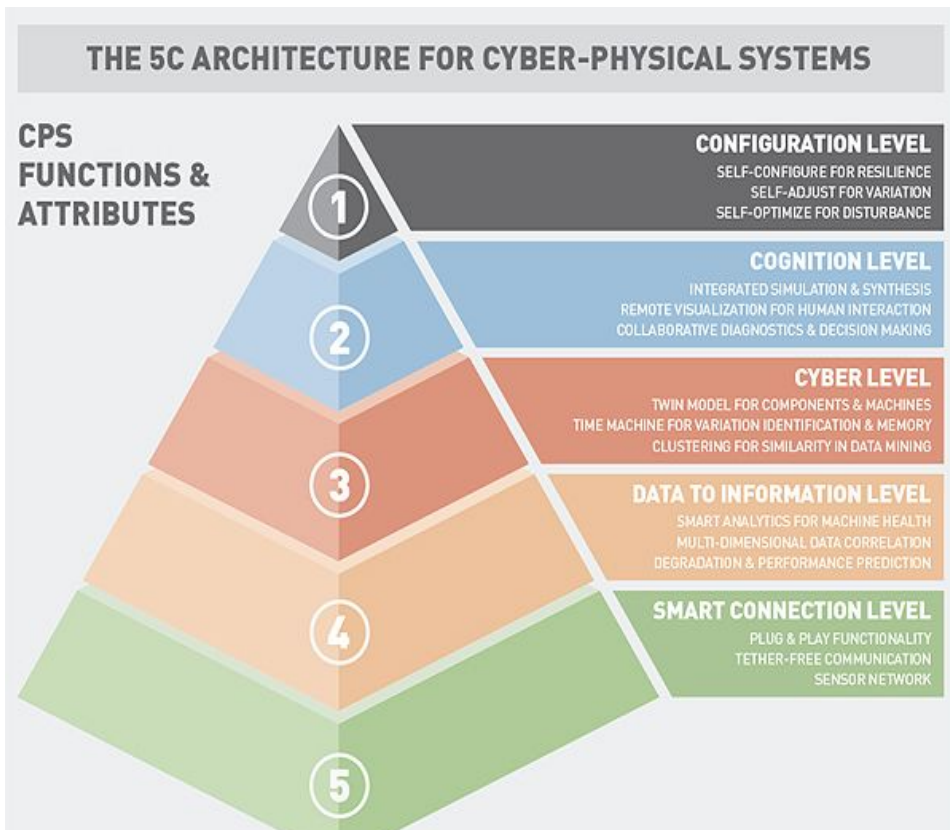
Frédéric Camps - LAAS/CNRS
fcamps@laas.fr

Agenda

- Architecture globale du cyber-système,
- Modèle abstrait d'architecture de l'IoT,
- OS et Hardware,
- Notion d'architecture système,
- Architectures classiques : self, home, city, work, commune,
- Les architecture verticales et les évolutions,
- Les architectures Cloud, Edge, Fog,
- Les services cloud comme solution industrielle : LaaS, PaaS, SaaS, Caas, Daas, Baas,
- Les composants d'une plateforme open source,
- IoT et design pattern.

Source: Lee, J., Bagheri, B., Kao, H.-A., 2015. A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. Manufacturing Letters 3, 18–23.

Architecture globale / Les Systèmes Cyber-Physiques (CPS)



• **C5. Au niveau Connexion**, le CPS opère sur un réseau Plug&Play et utilise des données envoyées par un réseau de capteurs ; IoT, IIoT.

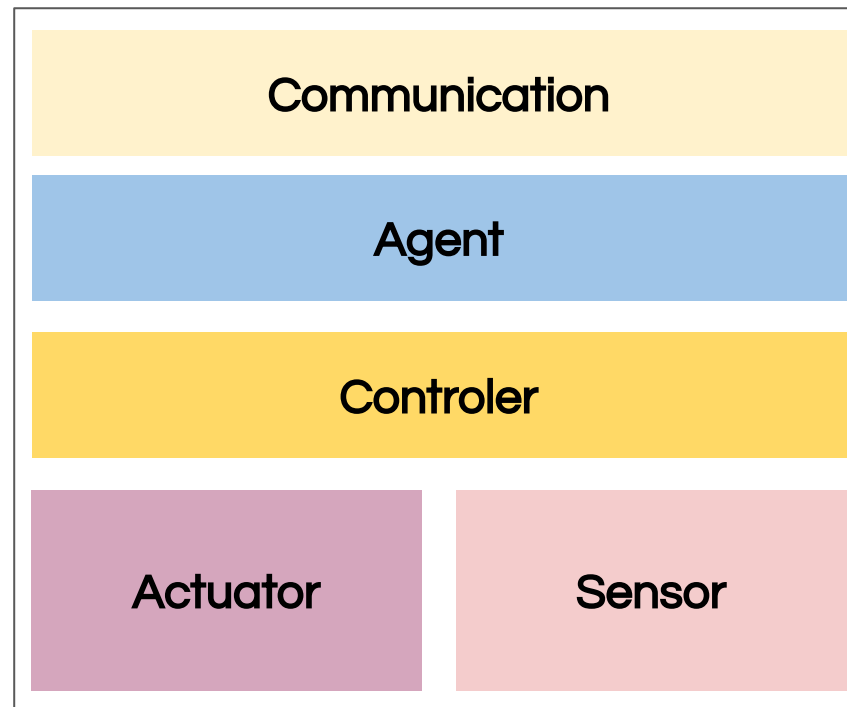
• **C4. Au niveau Conversion**, le CPS sait traiter l'information et la retranscrire en informations de plus haut niveau ;

• **C3. Au niveau Cyber**, le CPS a une connaissance des autres CPS de l'environnement et peut interagir avec eux pour enrichir son propre traitement d'information ;

• **C2. Au niveau Cognition**, le CPS est capable d'établir un diagnostic basé sur des simulations de son propre comportement et une analyse différentielle des données de capteurs ;

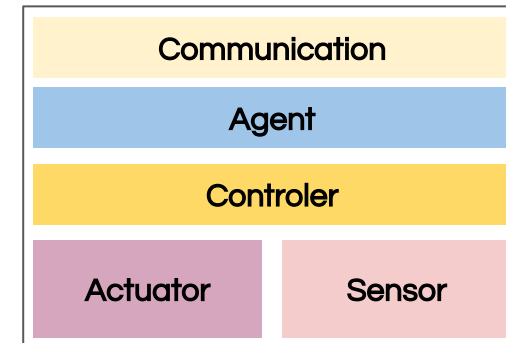
• **C1. Au niveau Configuration**, le CPS peut s'adapter seul en cas de défaillance, se reconfigurer ou ajuster de manière autonome ses paramètres afin de retourner à un comportement nominal.

IoT : Abstract Architecture Model



An IoT "Thing" anatomy

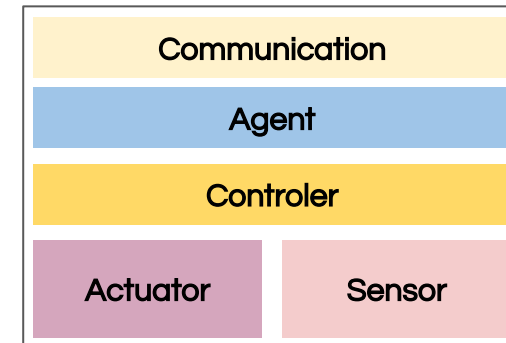
IoT : Abstract Architecture Model



Couche de communication

- Médias courte distance : Wifi, Bluetooth, NFC...
- Médias longue distance : Ethernet filaire, 3G, 4G, LoRa, SigFox, NB-IoT...
- Protocoles : TCP/UDP/IP, HTTP, CoAP (Constrained Application Protocol) ...

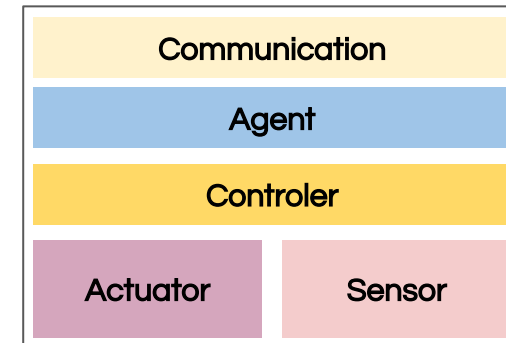
IoT : Abstract Architecture Model



Agent est un programme embarqué qui :

- Fonctionne sur l'appareil IoT,
- Signale l'état d'un actif ou d'un environnement,
- Agit comme un pont entre le contrôleur et le cloud,
- Décider quelles données envoyer et quand les envoyer...
- Les algorithmes intelligents sont situés dans l'agent.

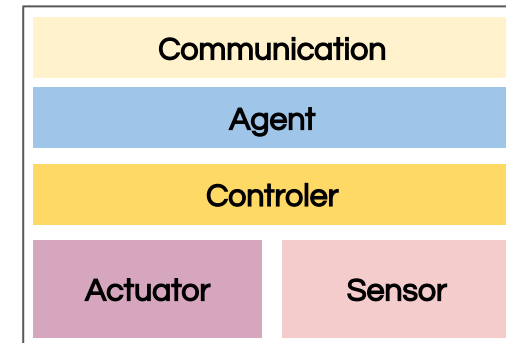
IoT : Abstract Architecture Model



Capteur

- Les capteurs lisent et signalent l'état réel des produits connectés, des machines et des environnements locaux.
- Ils sont les yeux et les oreilles du système, surveillant les éléments environnementaux tels que :
 - Capteurs de température,
 - Capteurs de lumière,
 - Capteurs d'humidité,
 - Capteurs de vibrations,
 - Capteurs de gaz,
 - Récepteurs GPS,
 - Capteur de présence,,
 - Capteur inertiel,
 - **New!** capteur composite sur Android...

IoT : Abstract Architecture Model



Actionneurs

- Les actionneurs affectent l'état électromécanique ou logique d'un produit ou d'un environnement.
- Ce sont les mains et les pieds du système :
 - Lumières,
 - Vannes,
 - Moteurs,
 - Commandes (actions "soft", distribution de fichiers, mises à jour du firmware) ...

IoT : Abstract Architecture Model

Caractéristiques globales (fonctionnelles et non fonctionnelles)

- Adapté à l'utilisation
- Mise à jour
- Contrôle distant
- Résilience
- Sécurité
- Intelligent
- Maintenable
- Fiable
- Ergonomie
- Compatibilité
- Réutilisabilité
- Performance
- Consommation énergie
- Coût (achat, maintenance, exploitation...)
- Utilisation en réseau
- Niveau d'intégration
- Facilité de déploiement
- Conformité aux normes
- Embarquabilité du système
- Interopérable
- Ecologique
- Autonome
- Multi-protocoles ...

Un produit particulièrement complexe !

<https://devopedia.org/iot-operating-systems>

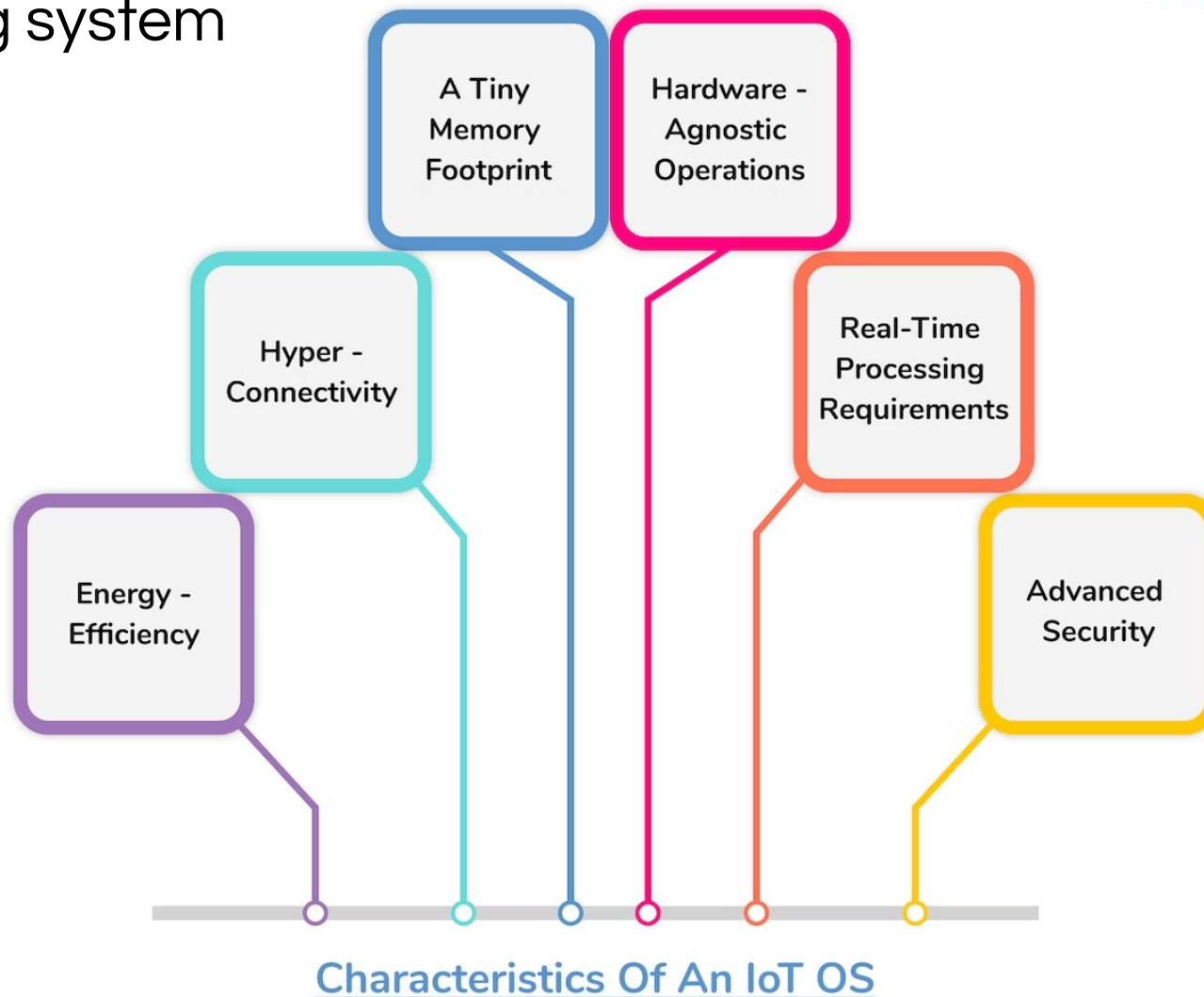
Operating system

- **Avec OS :**
 - Une grande souplesse pour le développement d'un IoT,
 - Il peut aussi demander plus de ressource système (pas toujours),
 - Il peut consommer plus d'énergie,
 - L'OS possède une complexité à maîtriser.

- **Sans OS :**
 - Un accès direct aux ressources,
 - Parfois plus complexe (driver, interface ...),
 - Une réduction de la consommation plus facile à maîtriser.

<https://www.intuz.com/top-iot-operating-systems-for-iot-devices>

Operating system

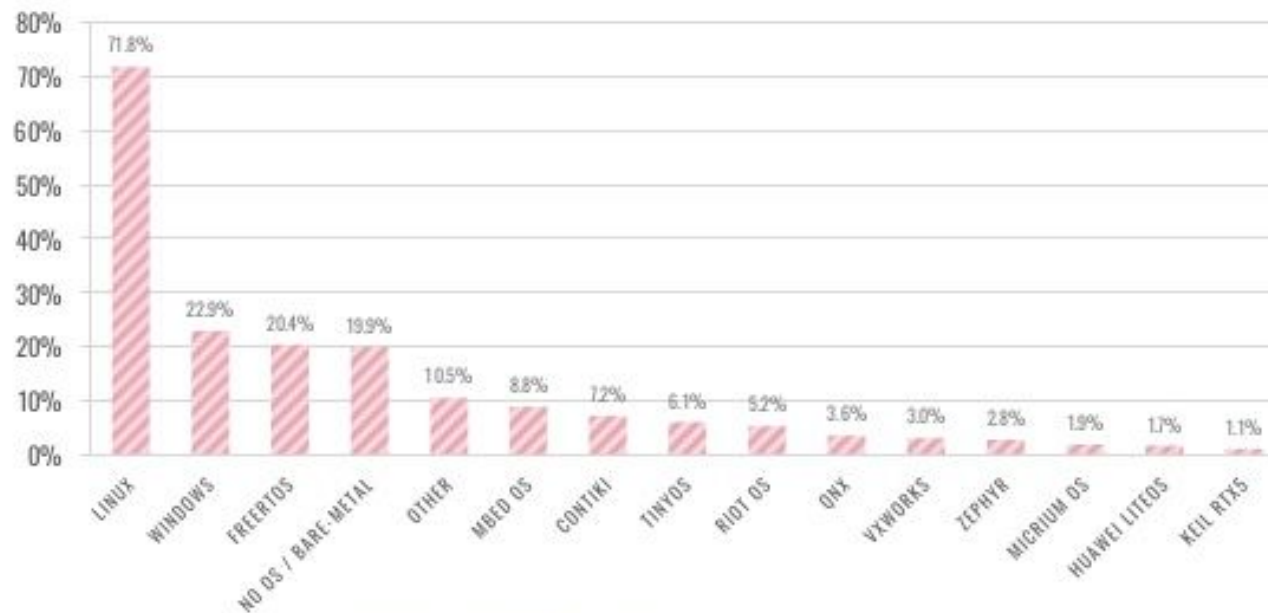


<https://devopedia.org/iot-operating-systems>

Operating system

IoT OPERATING SYSTEMS

Which operating system(s) do you use for your IoT devices?



Copyright (c) 2018, Eclipse Foundation, Inc. / Made available under a [Creative Commons Attribution 4.0 International License \(CC BY 4.0\)](https://creativecommons.org/licenses/by/4.0/).

25

<https://devopedia.org/iot-operating-systems>

Operating system

Open source: TinyOS, RIOT, Contiki, Mantis OS , Nano RK, LiteOS, FreeRTOS, Apache Mynewt, Zephyr OS , Ubuntu Core 16 (Snappy), ARM mbed, Yocto, Raspbian ...

TinyOS et Contiki sont parmi les plus anciens. RIOT est plus récent et possède une communauté active de développeurs. FreeRTOS est parmi les plus populaires.

OS commerciaux: Android Things, Windows 10 IoT , WindRiver VxWorks, Micrium μ C/ OS , Micro Digital SMX RTOS , MicroEJ OS , Express Logic ThreadX, TI RTOS , Freescale MQX, Mentor Graphics Nucleus RTOS , Green Hills Integrity , Particule.

Windows 10 IoT se décline en trois versions : IoT Enterprise, IoT Mobile Enterprise et Core.TI RTOS et Freescale MQX ciblent les chipsets respectifs.

<https://devopedia.org/iot-operating-systems>

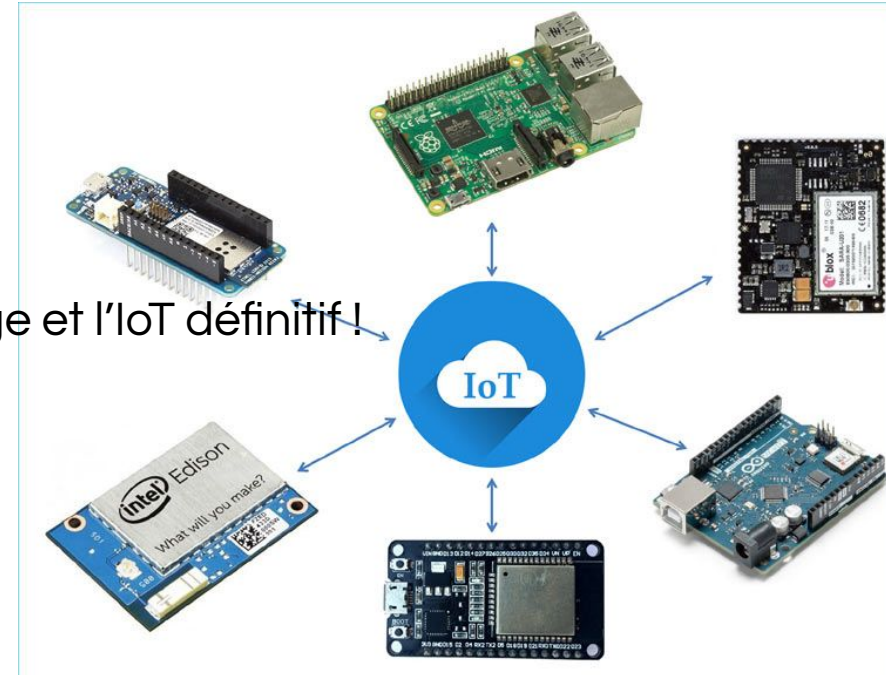
Operating system

	Contiki	RIOT	FreeRTOS	TinyOS	uClinux	Mbed
Architecture	monolithic	microkernel RTOS	microkernel RTOS	monolithic	monolithic	monolithic
Scheduler	cooperative	preemptive, tickless	preemptive, optional tickless	cooperative	preemptive	preemptive
Programming model	event-driven, protothreads	multi-threading	multi-threading	event-driven	multi-threading	Event driven, single-thread ²
Targeted device class ¹	Class 0, 1	Class 1,2	Class 1,2	Class 0	>Class 2	Class 1,2
Supported MCU families or vendors	AVR®, MSP430™, ARM® Cortex-M®, PIC32, 6502	AVR®, MSP430™, ARM® Cortex-M®, x86	AVR®, MSP430™, ARM®, x86, 8052, Renesas	AVR®, MSP430™, px27ax	ARM7™, ARM® Cortex-M®	ARM® Cortex-M®
Programming languages	C	C, C++	C	nesC	C	C, C++
License	BSD	LGPLv2	modified GPL	BSD	GPLv2	Apache License 2.0

<https://devopedia.org/iot-operating-systems>

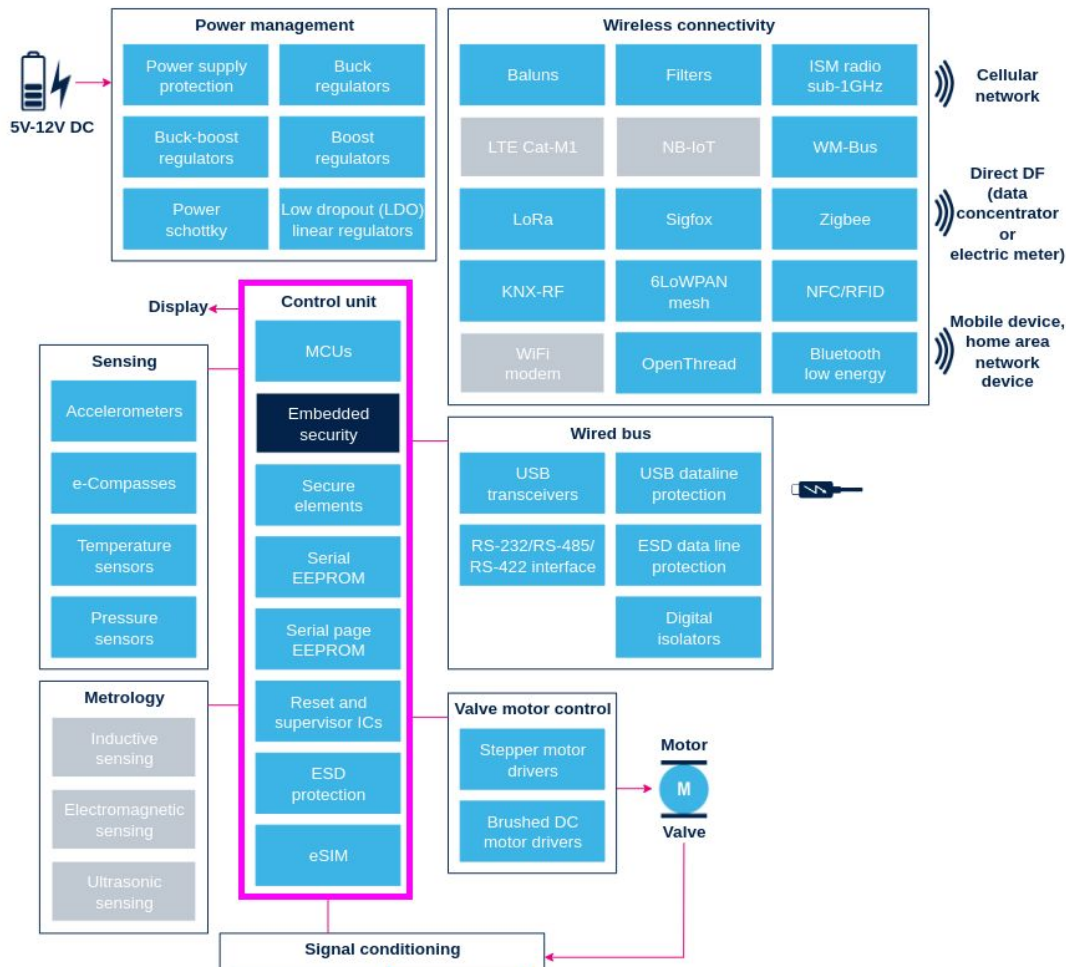
IoT : Hardware

- Ne pas confondre une plateforme de prototypage et l'IoT définitif !
- Prendre en compte:
 - Température,
 - Humidité,
 - Pression,
 - Résistance UV,
 - CEM (compatibilité électromagnétique),
 - Environnement radiatif, accélération,
 - Niveau de tension (adaptation des E/S), courant (thermique),
 - Bus (SPI, I2C, UART, USART, RS232, CAN, DAC, USB, Firewire, OneWire, GPIO),
 - Communication: Ethernet, Bluetooth , BLE, WIFI, LoRA, NFC, 3/4/5G, 4G LTE-M, ZigBee ...
 - 8/16/32 bits,
 - Chiffrement,
 - Simple ou multi-coeur, architecture Harvard ou von Neumann,
 - FPGA, GPU,
 - Mémoire vive, mémoire SRAM (permanente)
 - Langage utilisé (C/C++, Python, Java, LUA, script ...),
 - Environnement de développement, JTAG, CMSIS,
 - Driver disponible,
 - Coût et approvisionnement.
- La plupart des kit de dev. offre des possibilités très réduite ! Attention !



<https://www.st.com/en/applications/metering/gas-metering.html>

IoT : Hardware / MCU



▼ MCUs	20 total entries	Open full selector
STM32L0 Series	Sort by: A-Z ▼	Datasheet
STM32L4 Series		
STM32L4+ Series	★ STM32L063R8 - Ultra-low-power Arm Cortex-M0+ MCU with 64 Kbytes of Flash memory, 32 MHz, USB, LCD, AES	
STM32L5 Series	ACTIVE	
► Secure elements	★ STM32L083RZ - Ultra-low-power Arm Cortex-M0+ MCU with 192-Kbytes of Flash memory, 32 MHz CPU, USB, LCD, AES	
► eSIM	ACTIVE	
► Serial EEPROM	★ STM32L083VZ - Ultra-low-power Arm Cortex-M0+ MCU with 192-Kbytes of Flash memory, 32 MHz CPU, USB, LCD, AES	
► Reset and supervisor ICs	ACTIVE	
► ESD protection	STM32L053C6 - Ultra-low-power Arm Cortex-M0+ MCU with 32 Kbytes of Flash memory, 32 MHz CPU, USB, LCD	
► Serial page EEPROM	ACTIVE	
	STM32L053C8 - Ultra-low-power Arm Cortex-M0+ MCU with 64 Kbytes of Flash memory, 32 MHz CPU, USB, LCD	
	ACTIVE	

Agenda

- Architecture globale du cyber-système,
- Modèle abstrait d'architecture de l'IoT,
- **Notion d'architecture système,**
- Architectures classiques : self, home, city, work, commune,
- Les architecture verticales et les évolutions,
- Les architectures Cloud, Edge, Fog,
- Les services cloud comme solution industrielle : LaaS, PaaS, SaaS, Caas, Daas, Baas,
- Les composants d'une plateforme open source,
- IoT et design pattern.

Qu'est ce qu'une architecture ?

- Une architecture définit les différents éléments d'un système,
- Les acteurs (humains et systèmes),
- Les interfaces qui interconnectent les éléments,
- Les comportements dynamiques.

<https://www.itu.int/en/publications/Pages/default.aspx>

L'architectures d'une plateforme IOT (simplifiée) / Normalisation

UIT-T Y.2060 - Union internationale des télécommunications (documents gratuits)

- Y.4000/Y.2060 ([Overview](#) of the IoT)
- Y.4050/Y.2069 ([Terms and definitions](#) for the IoT)
- Y.4100/Y.2066 (Common [requirements](#) of the IoT)
- Y.4103/F.748.0 (Common [requirements](#) for IoT applications)
- Y.4552/Y.2078 (Application support models of the IoT)
- Y.4111/Y.2076 (Semantics-based requirements and framework of the IoT)
- [Y.4113 \(Requirements of the network for the IoT\)](#)
- Y.4453 (Adaptive software framework for IoT devices)
- Y.4101/Y.2067 (Common requirements and capabilities of a gateway for IoT applications)
- Y.4112/Y.2077 (Requirements of the plug and play (PnP))
- Y.4401/Y.2068 (Functional framework and capabilities of the IoT)
- Y.4806 (Security capabilities supporting safety of the IoT)

<https://www.itu.int/en/publications/Pages/default.aspx>

L'architectures d'une plateforme IOT (simplifiée) / Normalisation

ISO/IEC - (documents payants)

- ISO/IEC 20924:2018 (IoT definition and vocabulary)
- ISO/IEC 21823-1:2019 (Interoperability for IoT systems – Part 1: Framework)
- [ISO/IEC 22417:2017 \(IoT use cases\)](#)
- ISO/IEC 29161:2016 (Unique identification for the IoT)
- ISO/IEC 29181-9:2017 (Future Network – Problem statement and requirements, Part 9: Networking of everything)
- [ISO/IEC 30141:2018 \(IoT reference architecture\)](#)

<https://www.itu.int/en/publications/Pages/default.aspx>

L'architectures d'une plateforme IOT (simplifiée) / Normalisation

IEEE standards - (documents payants)

- P2413 (Architectural framework (AF) for the IoT)
- P1451-99 (Harmonization of IoT devices and systems)
- P1931.1 (AF for real-time onsite operations facilitation for the IoT)
- P2668 (Maturity index of IoT: evaluation, grading and ranking)

<https://www.etsi.org/technologies/consumer-iot-security>
https://www.etsi.org/deliver/etsi_en/303600_303699/303645/02.01.01_60/en_303645v020101p.pdf
https://ec.europa.eu/commission/presscorner/detail/fr/ip_21_5634
https://fr.wikipedia.org/wiki/Directive_RED

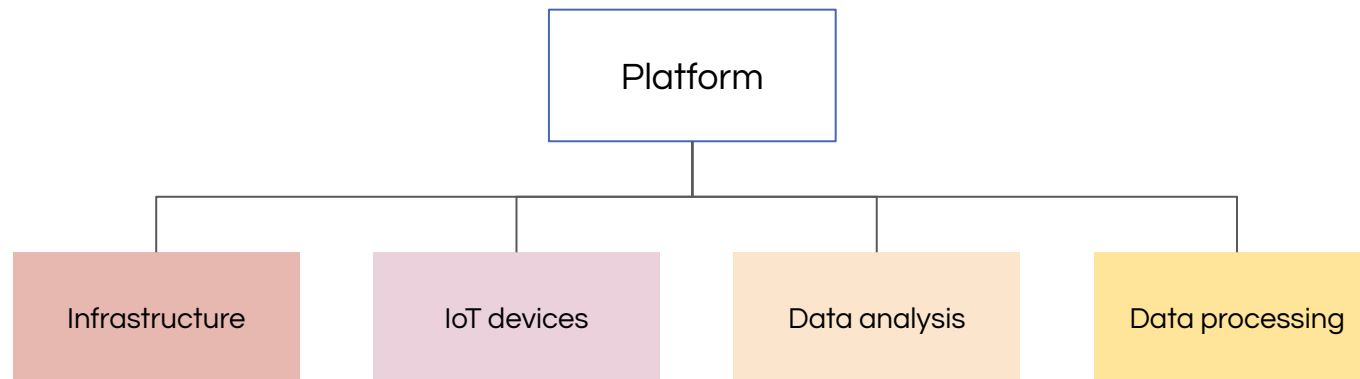
L'architectures d'une plateforme IOT (simplifiée) / Normalisation

Normes de sécurité IoT - (documents payants/gratuits)

- [ETSI EN 303 645](#) (Cyber Security for Consumer Internet of Things: Baseline Requirements. European Standard)
- La directive RED (directive européenne 2014/53/UE1)
- ISO/IEC 29181-5:2014 (Future Network -- Problem statement and requirements)
- X.1362 (Simple encryption procedure for IoT environments)
- Y.4102/Y.2074 (Requirements for IoT devices and operation of IoT applications during disasters)
- Y.4455 (Reference architecture for IoT network service capability exposure)
- Y.4118 (IoT requirements and technical capabilities for support of accounting and charging)
- Q.3952 (The architecture and facilities of a model network for IoT testing)
- Y.4702 (Common requirements and capabilities of device management in the IoT)
- Q.3913 (Set of parameters for monitoring IoT devices)

Architecture de haut niveau (simplifiée)

- Une architecture basée sur des composants permet de remplacer un composant par une autre version voire une autre technologie.
- L'architecture du système est évolutive et modulaire.



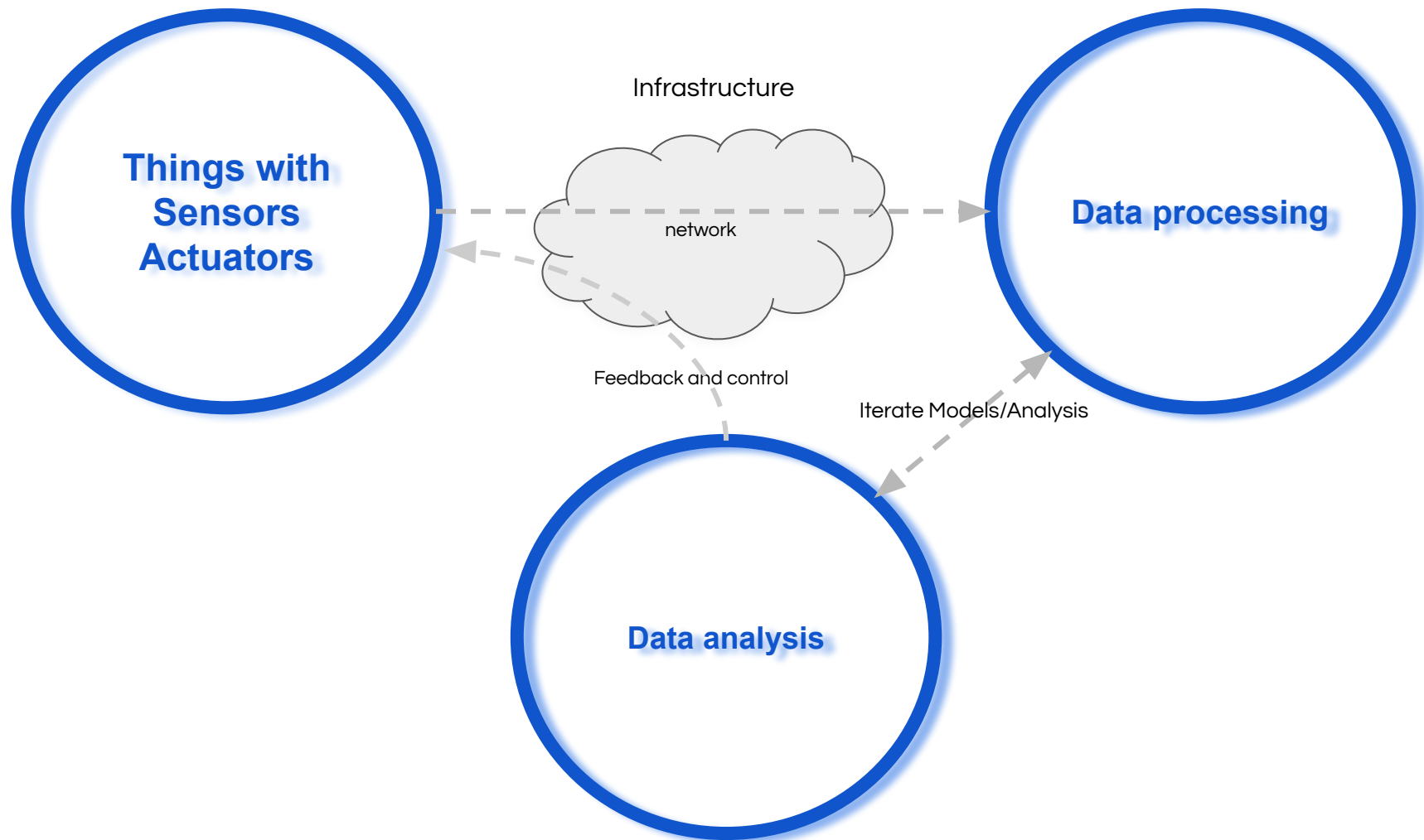
<http://www.etsi.org/>

Les 4 composants majeurs d'une plateforme IOT (simplifiée)

- **Le parc d'objets IoT:** géographiquement dispersé, technologie hétérogène,
- **Infrastructure:** radio, filaire, bus de terrain, protocole ...
- **Data processing:** normalisation des données, queuing, database, search engine, règles d'analyse, comportement event driven,
- **Data analysis:** modèle, analyse, feedback vers IoT, IA

<http://www.etsi.org/>

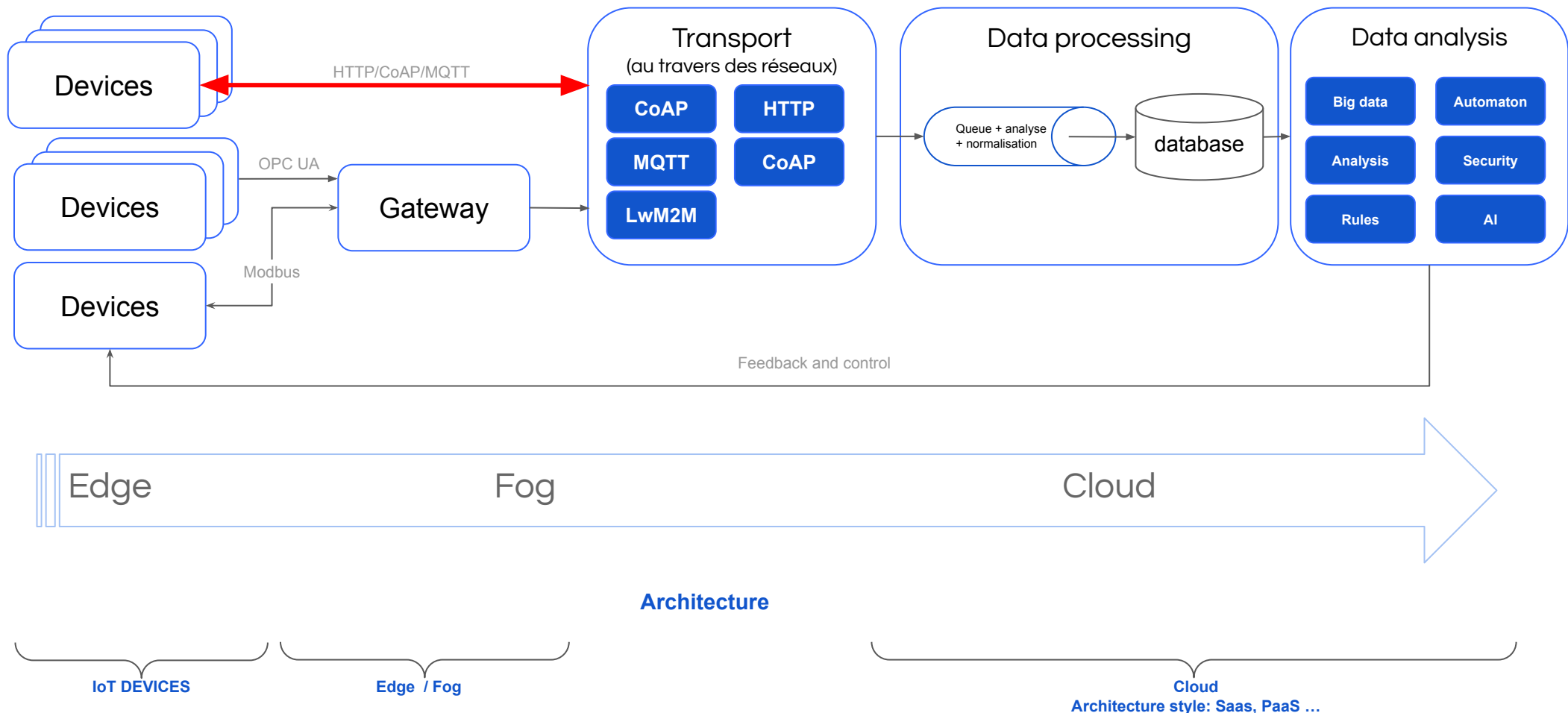
Les 4 composants majeurs d'une plateforme IOT (simplifiée)



IEEE INTERNET OF THINGS JOURNAL, VOL. 7, NO. 10, OCTOBER 2020 - P10091 /

Landscape of Architecture and Design Patterns for IoT Systems - <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9120234>

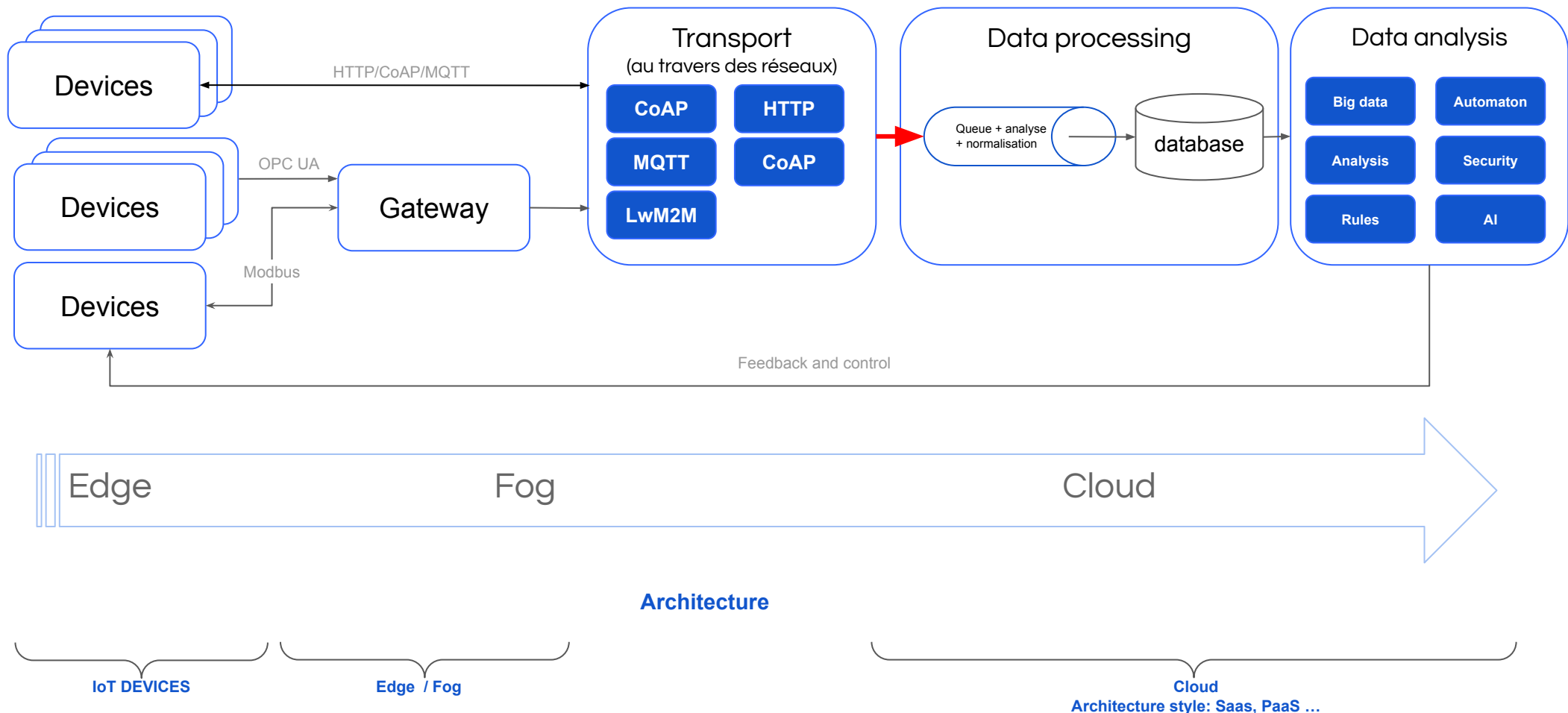
Les 4 composants majeurs d'une plateforme IOT (simplifiée)



IEEE INTERNET OF THINGS JOURNAL, VOL. 7, NO. 10, OCTOBER 2020 - P10091 /

Landscape of Architecture and Design Patterns for IoT Systems - <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9120234>

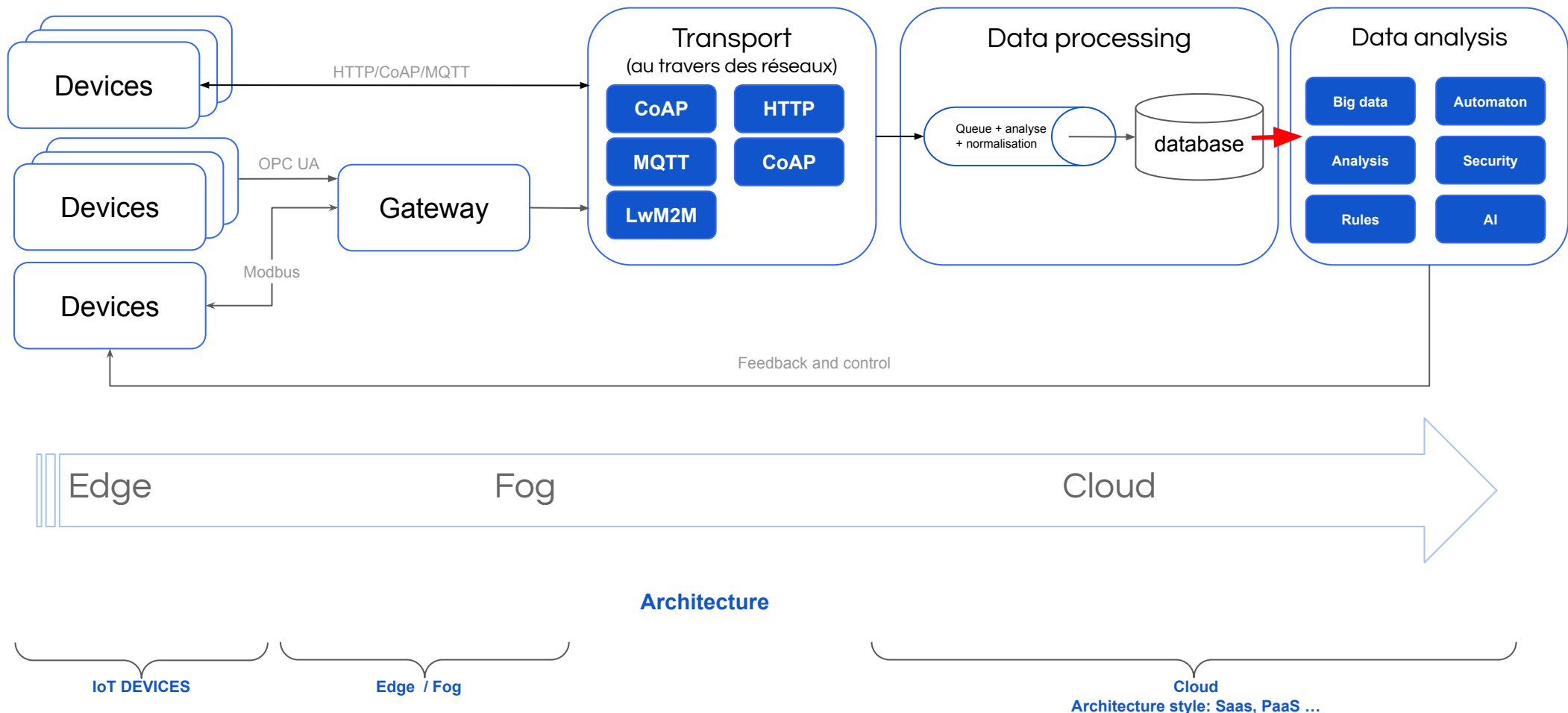
Les 4 composants majeurs d'une plateforme IOT (simplifiée)



IEEE INTERNET OF THINGS JOURNAL, VOL. 7, NO. 10, OCTOBER 2020 - P10091 /

Landscape of Architecture and Design Patterns for IoT Systems - <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9120234>

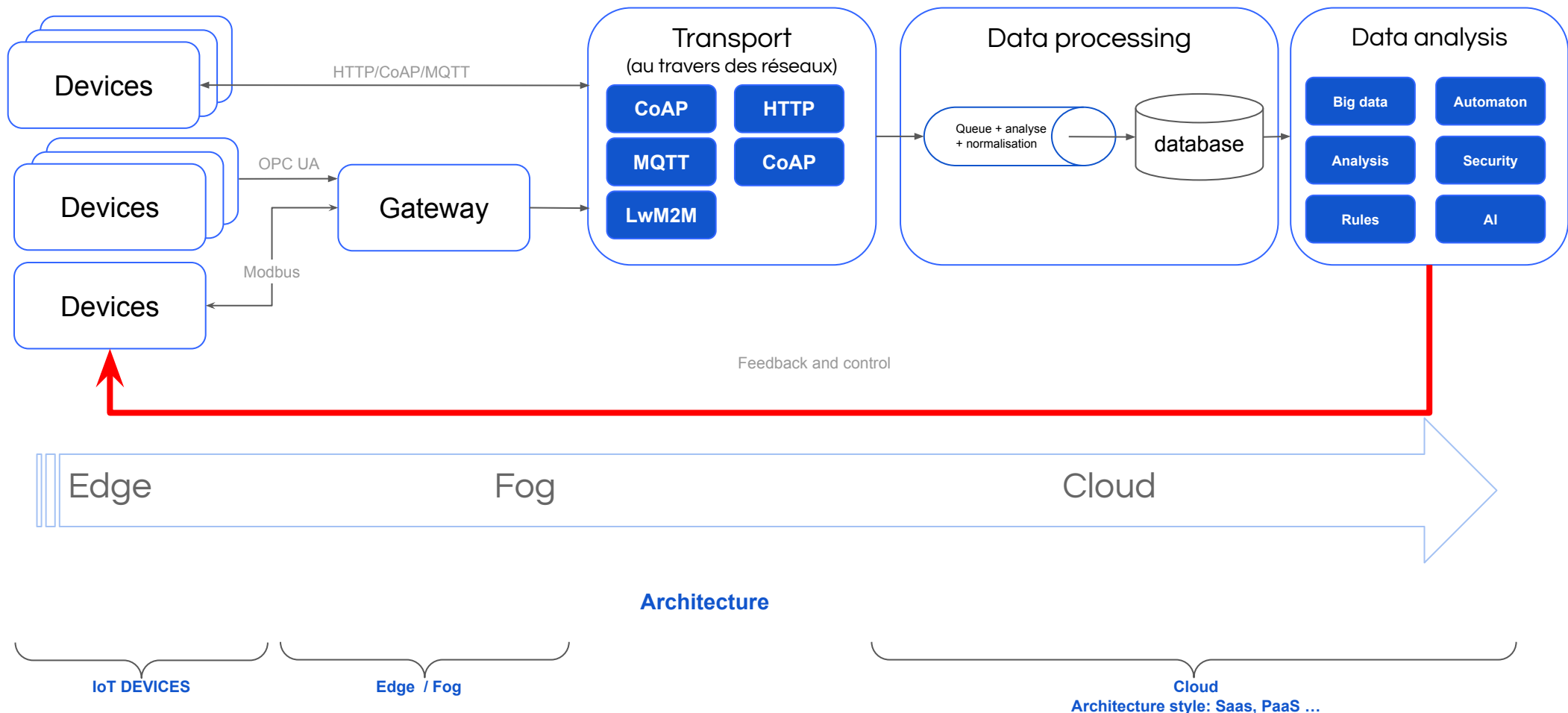
Les 4 composants majeurs d'une plateforme IOT (simplifiée)



IEEE INTERNET OF THINGS JOURNAL, VOL. 7, NO. 10, OCTOBER 2020 - P10091 /

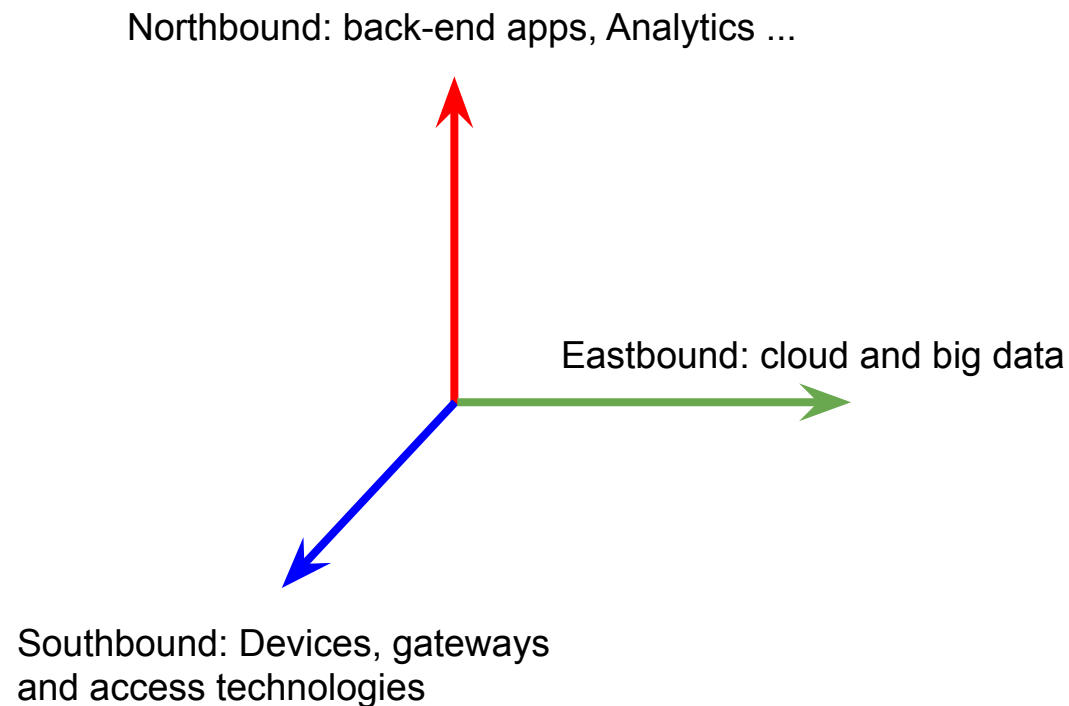
Landscape of Architecture and Design Patterns for IoT Systems - <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9120234>

Les 4 composants majeurs d'une plateforme IOT (simplifiée)



https://en.wikipedia.org/wiki/Northbound_interface

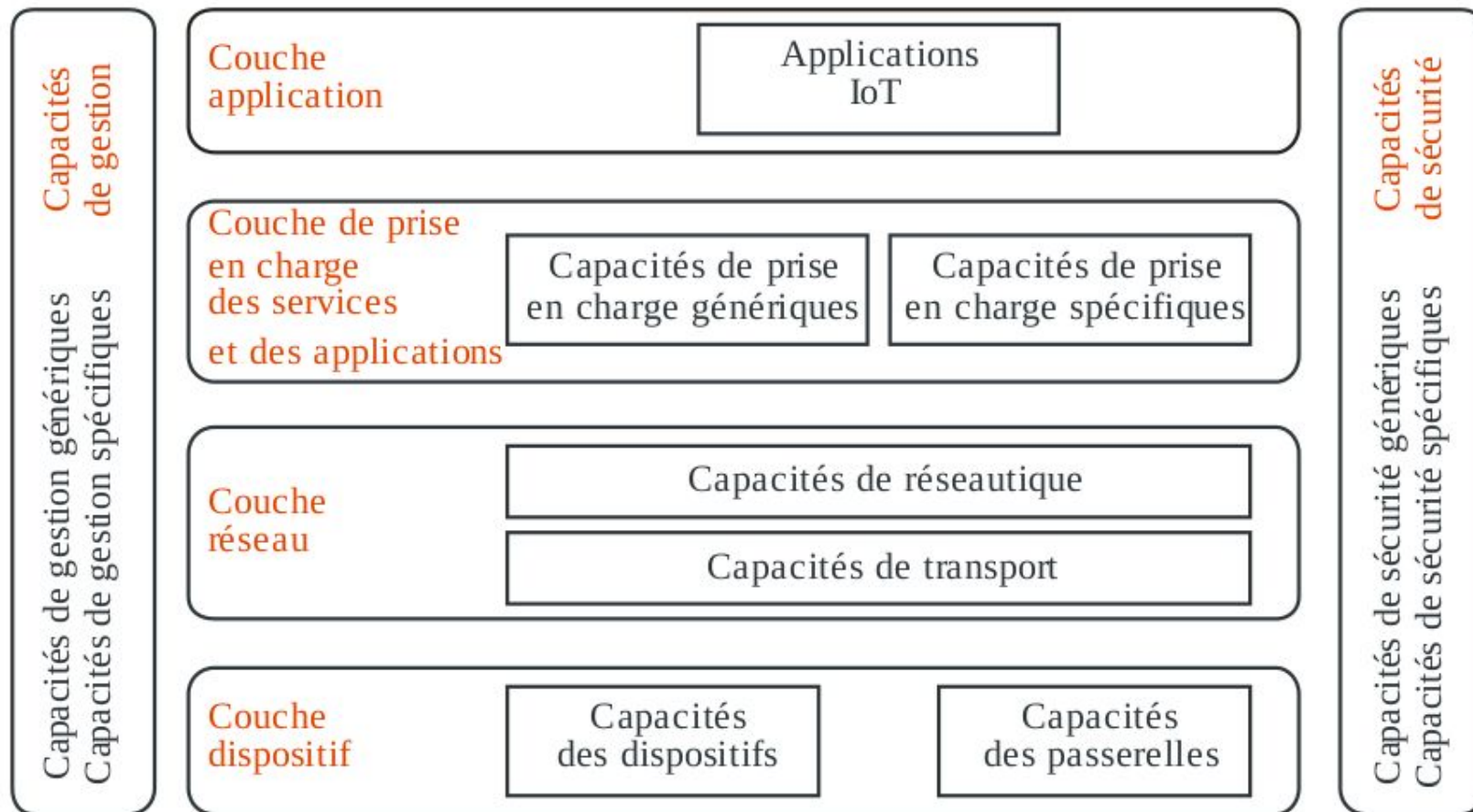
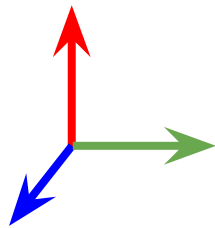
L'architectures d'une plateforme IOT (simplifiée)



<https://www.itu.int/rec/T-REC-Y.2060-201206-I/fr>

L'architectures d'une plateforme IOT (simplifiée) / Normalisation

UIT-T Y.2060 - Union internationale des télécommunications



Y.2060(12)_F04

https://en.wikipedia.org/wiki/Northbound_interface

L'architectures d'une plateforme IOT (simplifiée)

Les schémas d'architecture:

- Les interfaces northbound est la partie supérieure du composant,
- Les interfaces southbound, dans la partie inférieure,
- Ces termes s'appliquent dans presque tous les types de réseau ou de système informatique.

Agenda

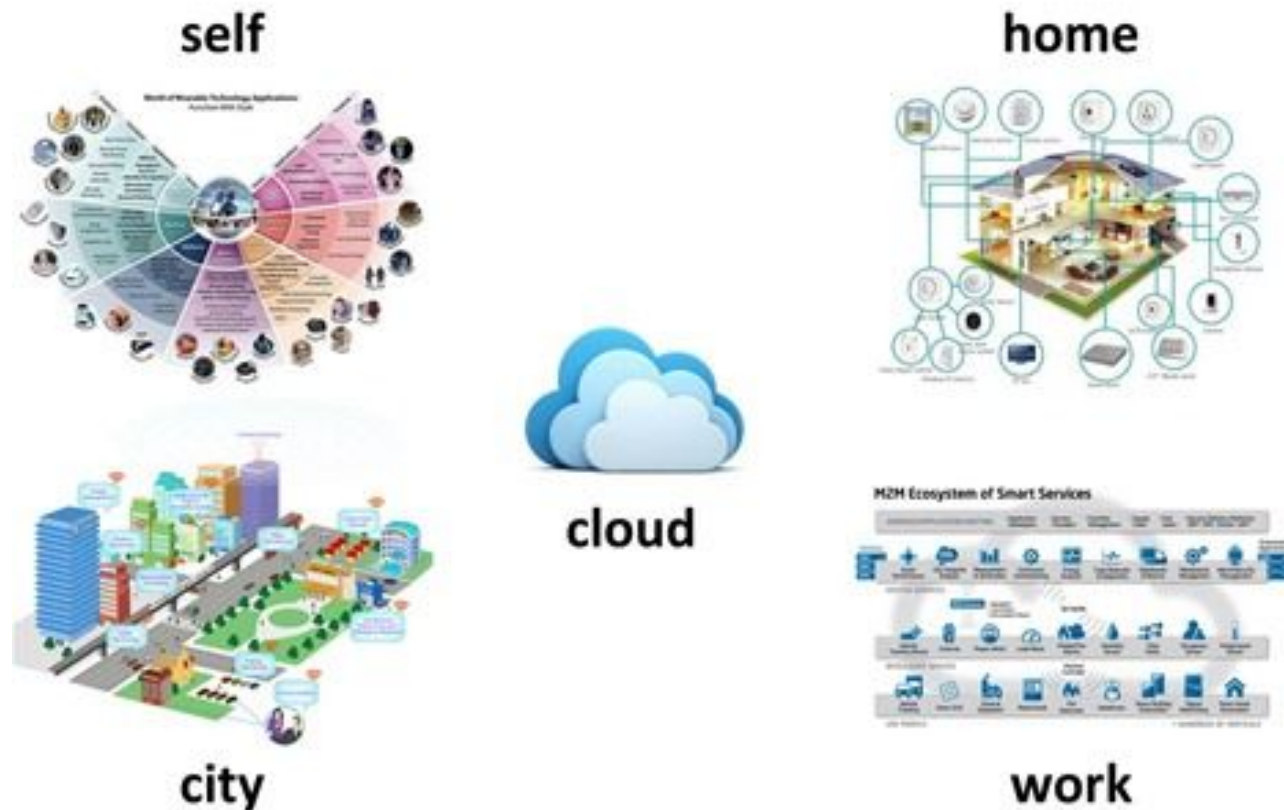
- Architecture globale du cyber-système,
- Modèle abstrait d'architecture de l'IoT,
- Notion d'architecture système,
- **Architectures classiques : self, home, city, work, commune,**
- Les architecture verticales et les évolutions,
- Les architectures Cloud, Edge, Fog,
- Les services cloud comme solution industrielle : LaaS, PaaS, SaaS, Caas, Daas, Baas,
- Les composants d'une plateforme open source,
- IoT et design pattern.

Self : utilisation personnelle que l'on porte sur soi

Home : utilisation dans la domotique

City : ville intelligente

Work : système industriel



<https://aws.amazon.com/marketplace/solutions/IoT>

Famille de Use Case

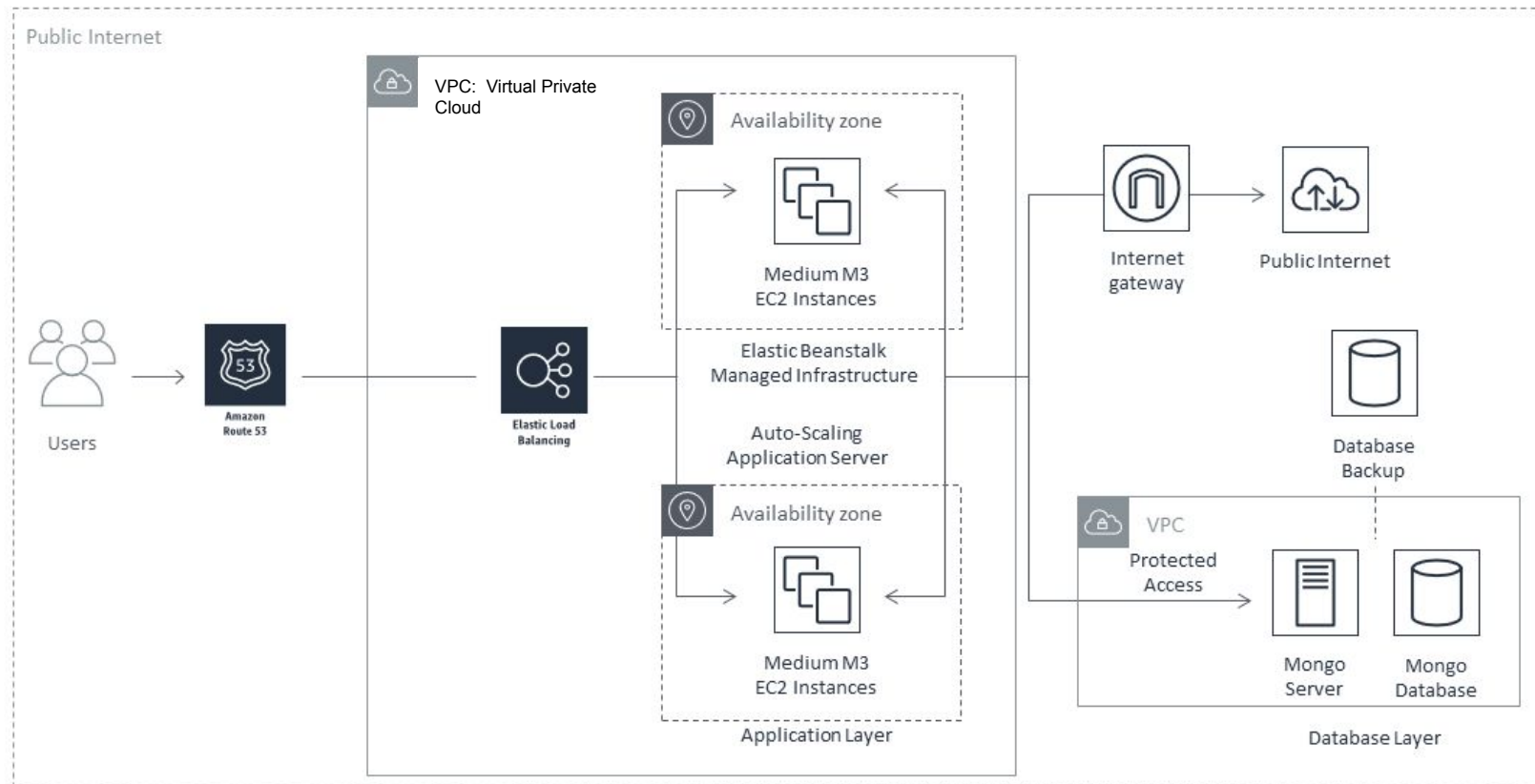
- Les Use case sont définis en sous famille,
- Les architectures sont différentes car les spécifications sont différentes:
 - Délais de bout en bout,
 - Prise de décision temps réel,
 - Gestion des événements,
 - Stockage,
 - Sécurité,
 - Sûreté de fonctionnement,
 - Coût ...

<https://aws.amazon.com/marketplace/solutions/IoT>

<https://aws.amazon.com/marketplace/solutions/IoT/data-visualization>

Famille de Use Case (AWS): Exemple de sous-famille

- Smart home and city : Visualisation de données à partir de plusieurs sources de données

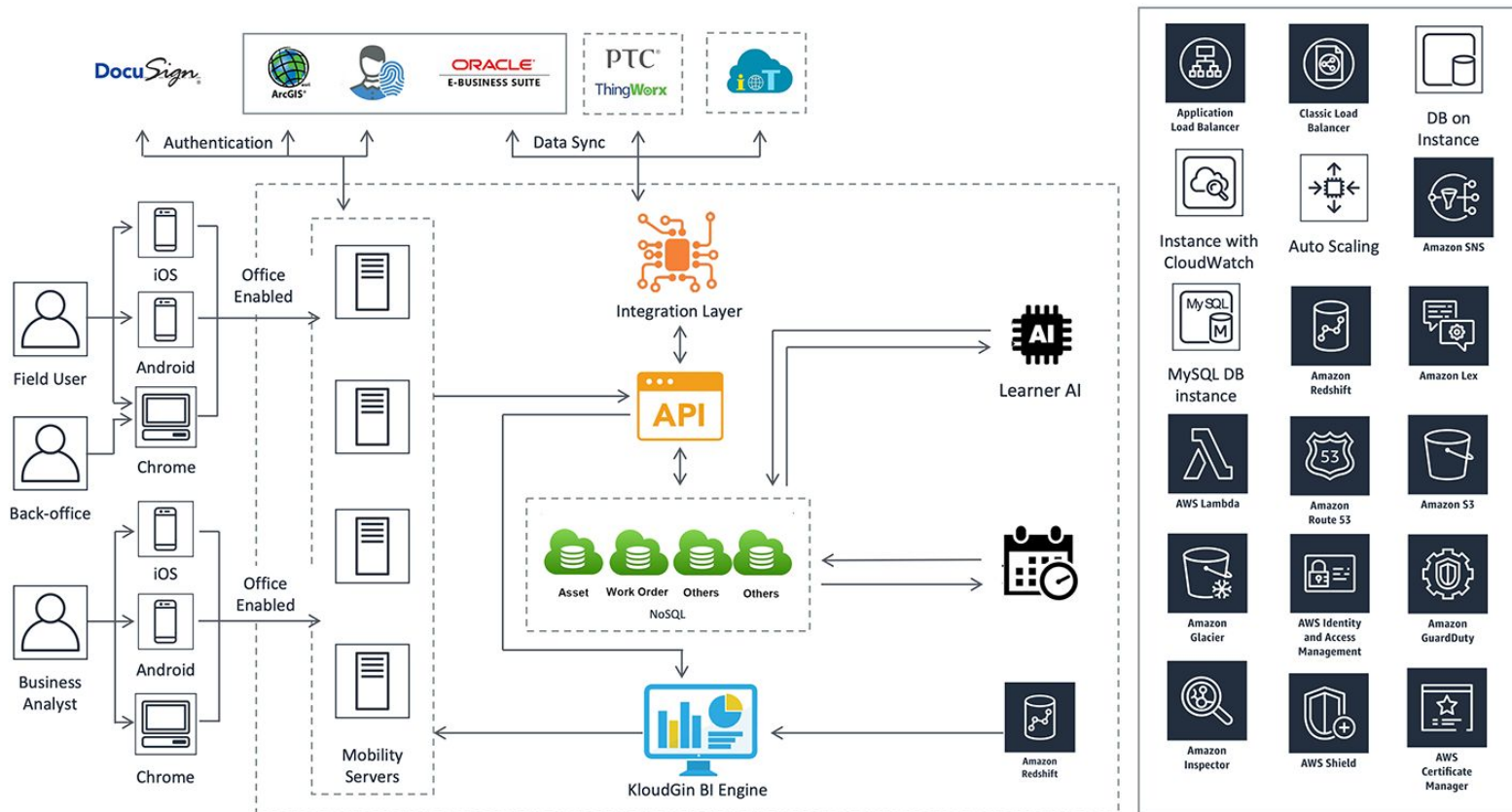


<https://aws.amazon.com/marketplace/solutions/IoT>

<https://aws.amazon.com/marketplace/solutions/IoT/monitoring-response-automation>

Famille de Use Case (AWS): Exemple de sous-famille

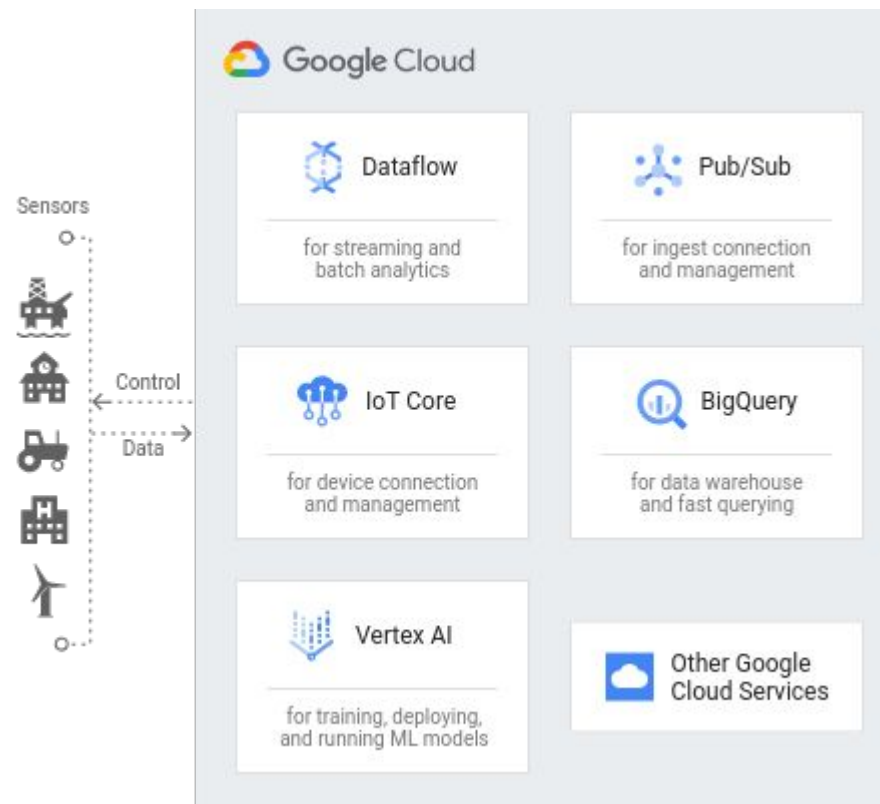
- Smart home and city : Automatisation de la surveillance et de la réponse



<https://cloud.google.com/solutions/iot>

Famille de Use Case (Google): Exemple de sous-famille

- Processus industriel : Automatisation d'un process

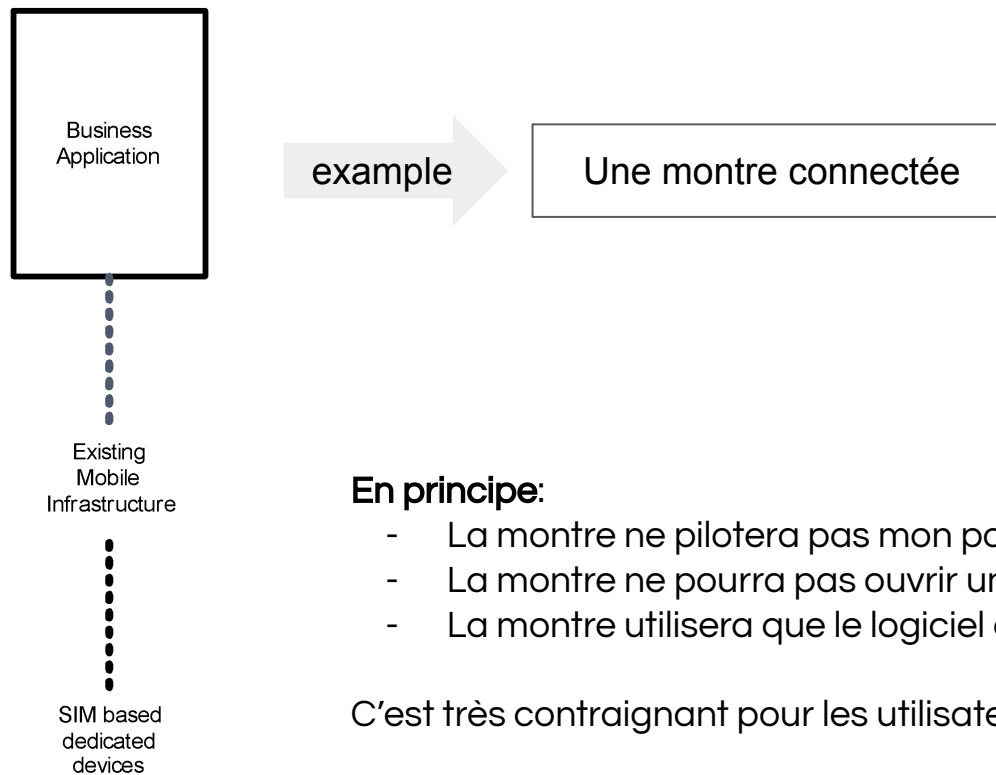


Agenda

- Architecture globale du cyber-système,
- Modèle abstrait d'architecture de l'IoT,
- Notion d'architecture système,
- Architectures classiques : self, home, city, work, commune,
- **Les architecture verticales et les évolutions,**
- Les architectures Cloud, Edge, Fog,
- Les services cloud comme solution industrielle : LaaS, PaaS, SaaS, Caas, Daas, Baas,
- Les composants d'une plateforme open source,
- IoT et design pattern.

<https://www.simfony.com/iot-platforms-vertically-versus-horizontally-layered-architecture/>
<https://sensative.com/yggio/horizontal-model/>

Architecture of vertical applications...



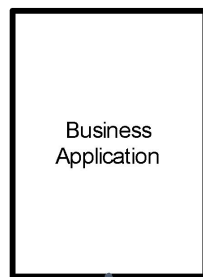
En principe:

- La montre ne pilotera pas mon portail,
- La montre ne pourra pas ouvrir une porte,
- La montre utilisera que le logiciel du fabricant ...

C'est très contraignant pour les utilisateurs !

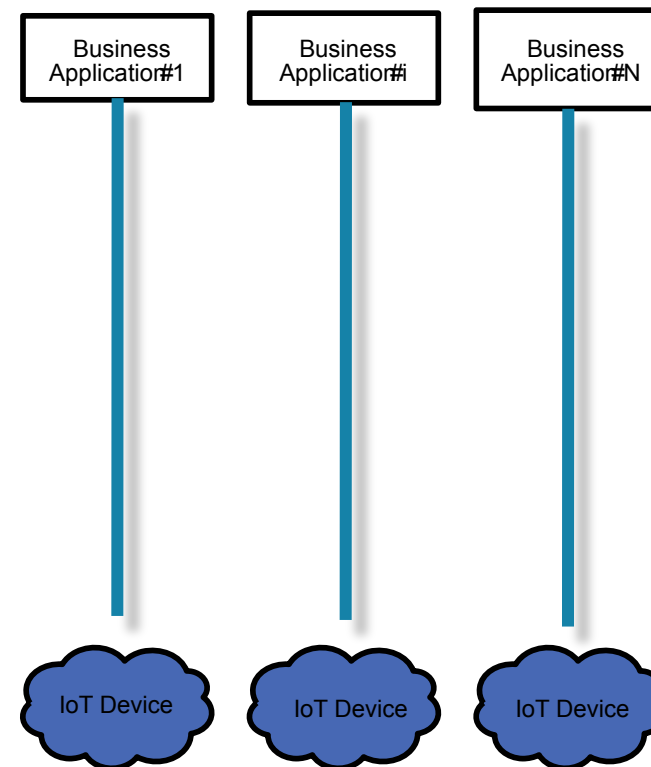
Architecture of vertical applications...

Vertical applications...

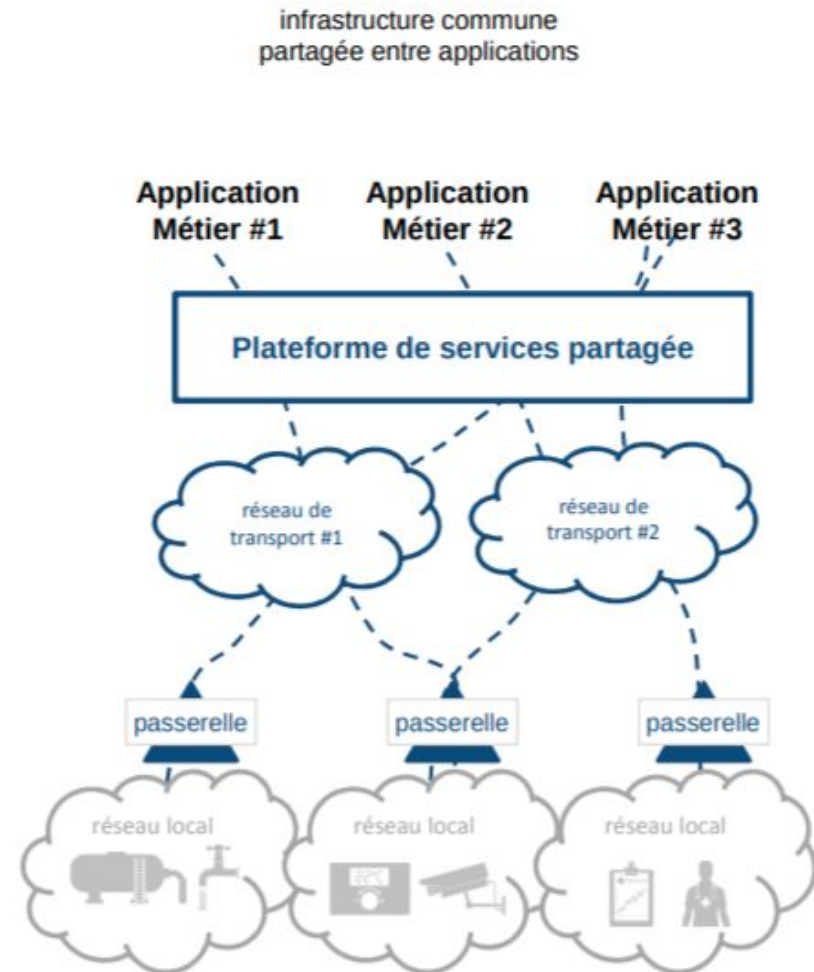
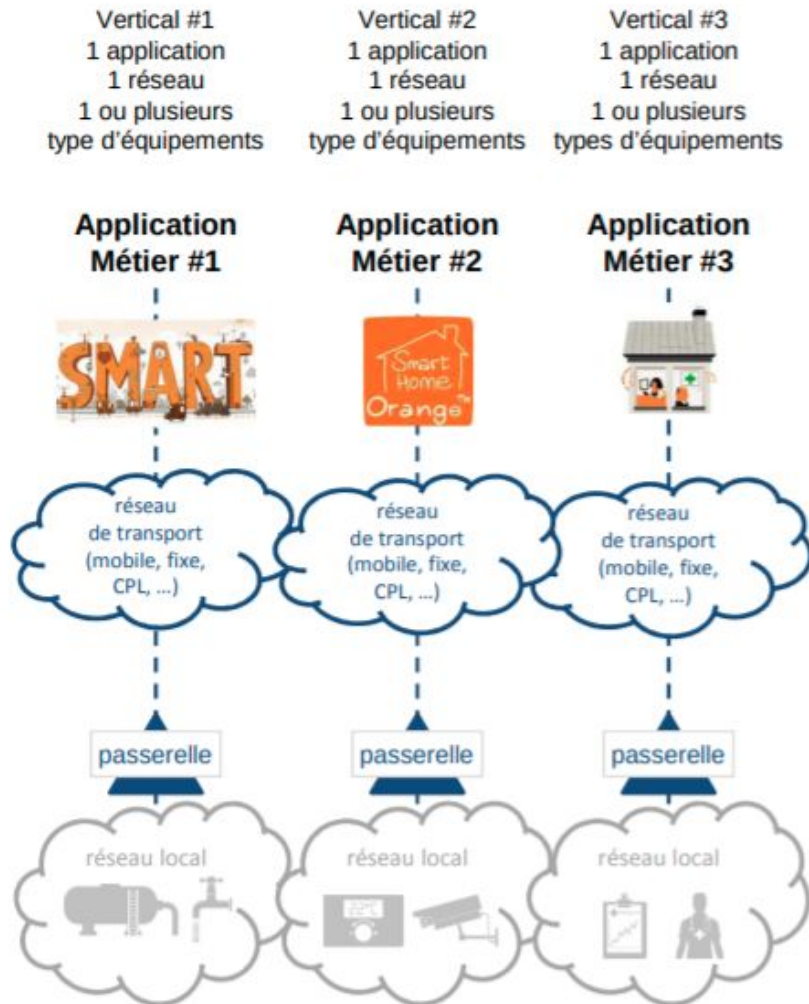


example

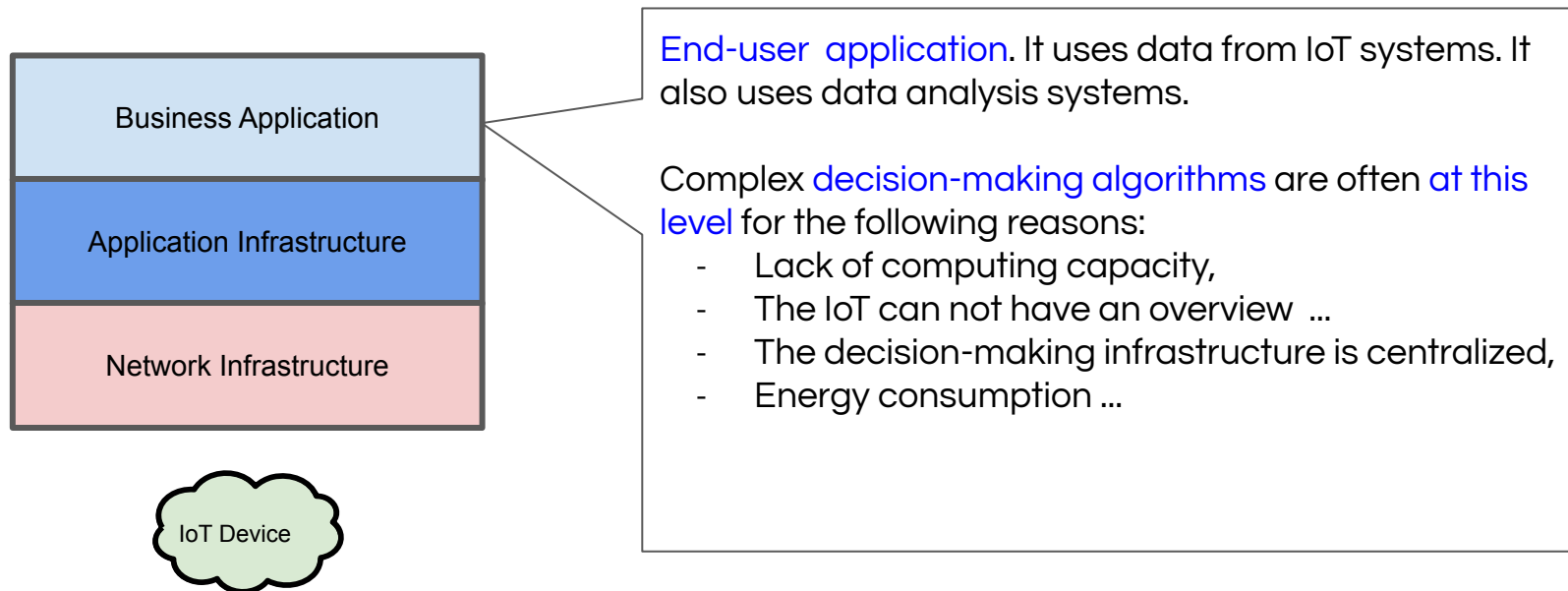
Les applications ne partagent pas de points communs dans l'infrastructure, environnements et éléments de réseau



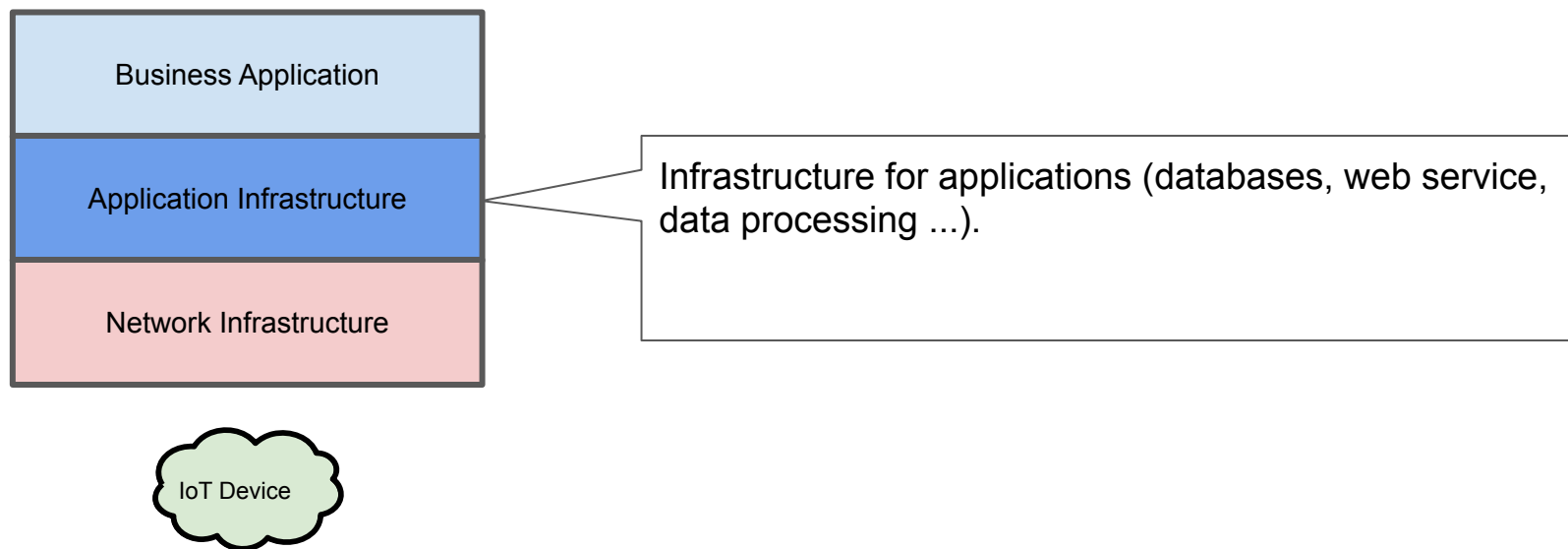
Architecture of vertical applications...



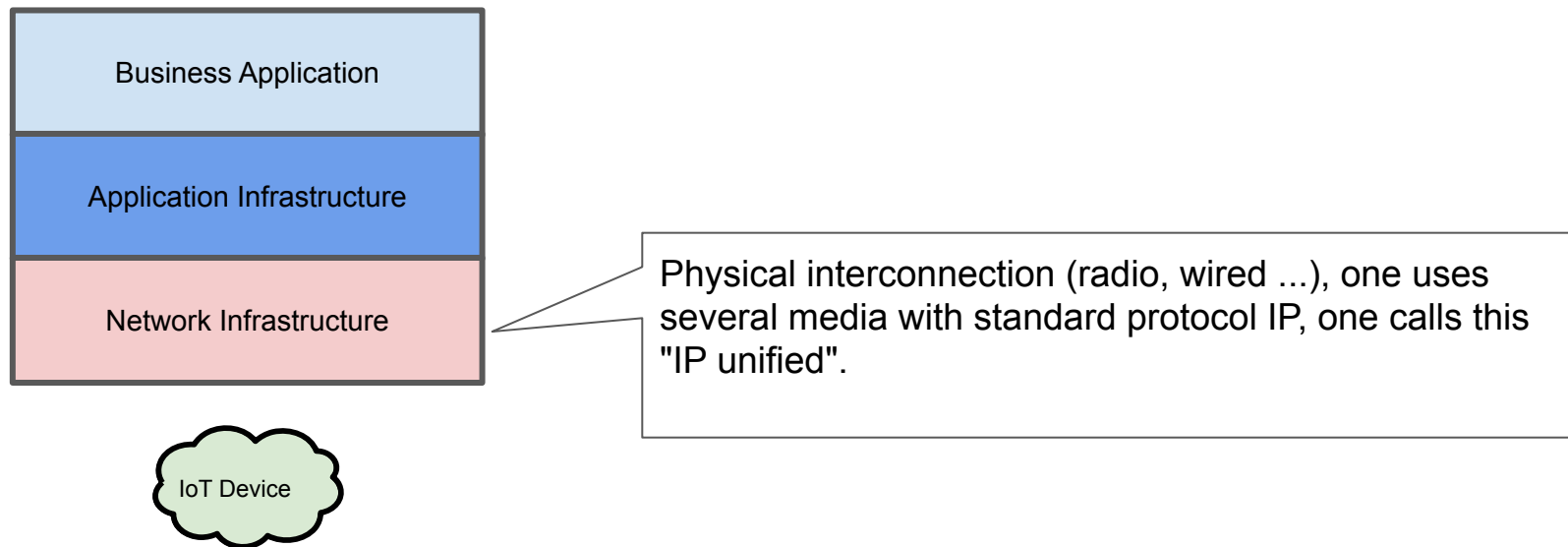
Architecture of vertical applications...



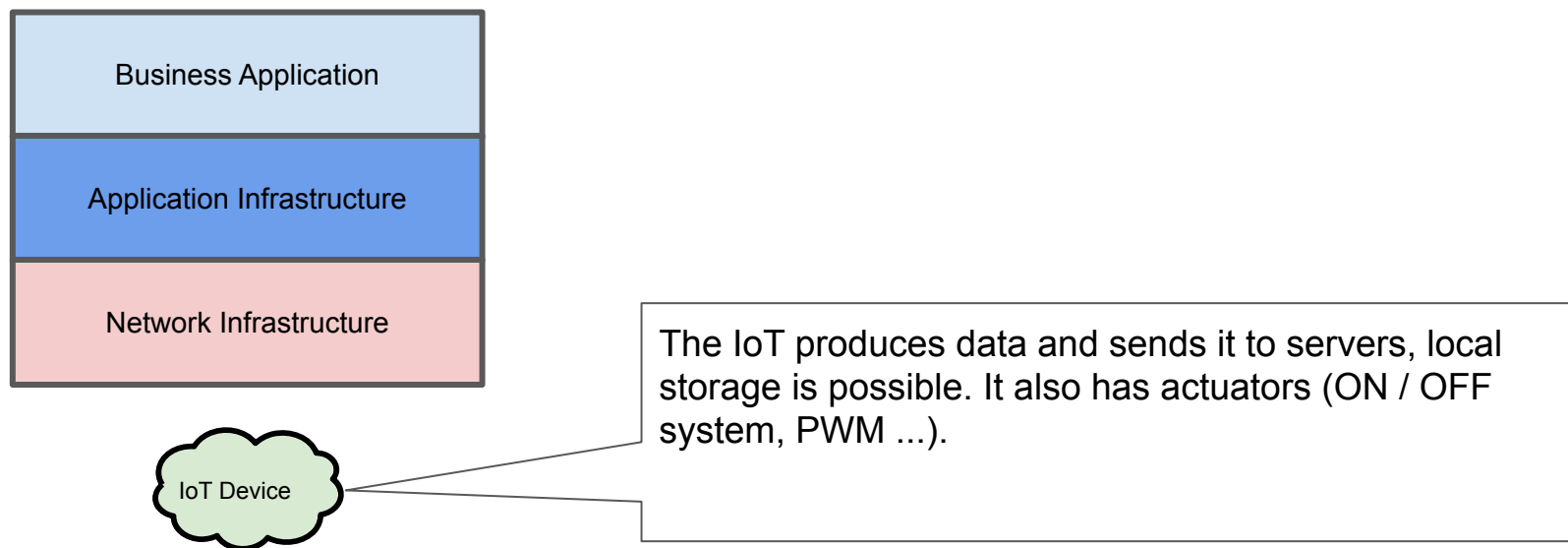
Architecture of vertical applications...



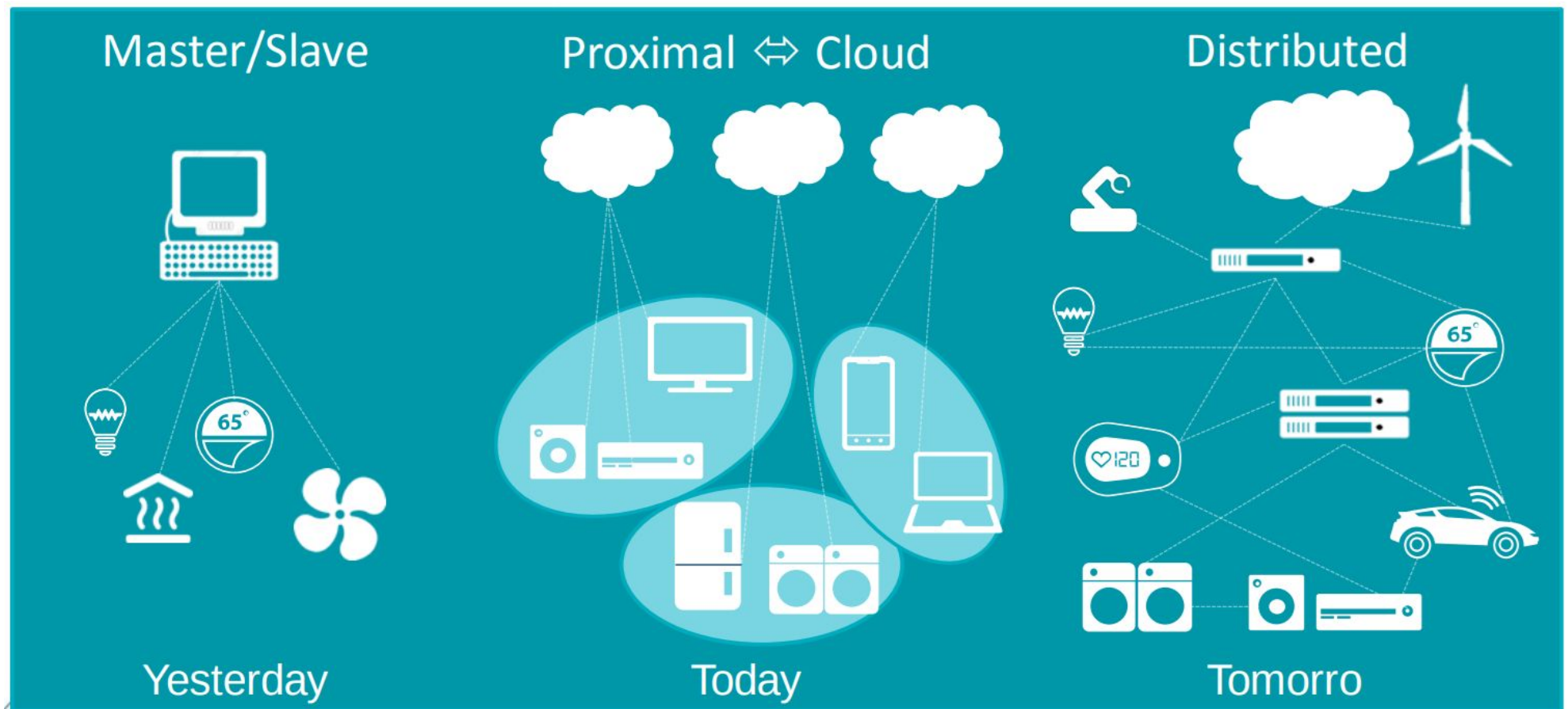
Architecture of vertical applications...



Architecture of vertical applications...



Tendance et évolution de l'IoT



Tendance et évolution de l'IoT

- Evolution vers des systèmes interopérables,
 - Tous les systèmes se connaissent et peuvent communiquer,
 - Un système peut utiliser les capacités / information d'un autre,
- Utilisation massive des infrastructures,
- Pilotage intelligent avec un **changement de paradigme M2M** (Machine 2 Machine).

Tendance et évolution de l'loT

- Embarquer de l'IA à moindre coût dans tous les systèmes ...
- Développer le B2B (Business 2 Business),
- Utilisation du NB-IOT → IoT ubiquitaire,
- Opérateur de L'loT → IoT Satellite as a Service:
 - Prise de photo,
 - Géolocalisation,
 - Communication ...

Agenda

- Architecture globale du cyber-système,
- Modèle abstrait d'architecture de l'IoT,
- Notion d'architecture système,
- Architectures classiques : self, home, city, work, commune,
- Les architecture verticales et les évolutions,
- **Les architectures Cloud, Edge, Fog,**
- Les services cloud comme solution industrielle : LaaS, PaaS, SaaS, Caas, Daas, Baas,
- Les composants d'une plateforme open source,
- IoT et design pattern.

<https://www.vmware.com/>

Architecture cloud (l'informatique dans les nuages)

Pourquoi opter pour une architecture Cloud ?

- Coût → mesure du ROI (Return on Investment)
- Fiabilité
- Sécurité
- Performance
- Support / aide technique
- Passage à l'échelle
- Accélérer la fourniture de ressource
- Garantir la conformité avec les réglementations
- Concentration sur le métier de l'entreprise ...

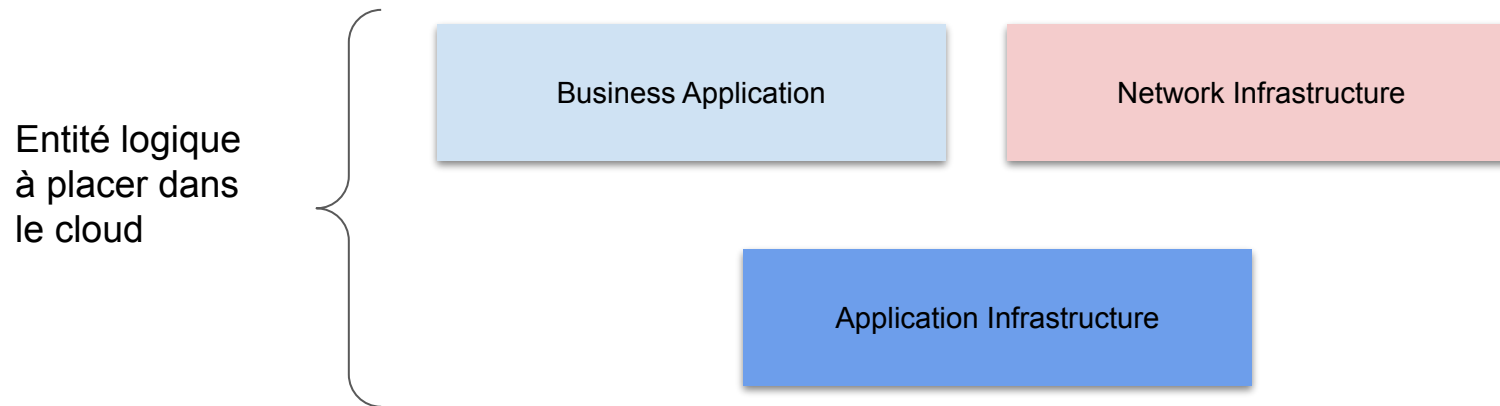
<https://www.itu.int/en/publications/Pages/default.aspx>

L'architectures d'une plateforme / Normalisation ITU

- Y.4552/Y.2078 (Application support models of the IoT)
- Y.4111/Y.2076 (Semantics-based requirements and framework of the IoT)
- Y.4113 (Requirements of the network for the IoT)
- Y.4453 (Adaptive software framework for IoT devices)
- Y.4101/Y.2067 (Common requirements and capabilities of a gateway for IoT applications)
- Y.4112/Y.2077 (Requirements of the plug and play (PnP))
- Y.4401/Y.2068 (Functional framework and capabilities of the IoT)
- Y.4806 (Security capabilities supporting safety of the IoT)

<https://www.vmware.com/>

Entité dans le cloud



Le cloud va fournir des **services** et **infrastructure** pour répondre à chaque couche logique :

- Logiciel: OS, middleware, gestion
- Matériel: serveur, disques
- Service: support, debug

<https://www.vmware.com/>

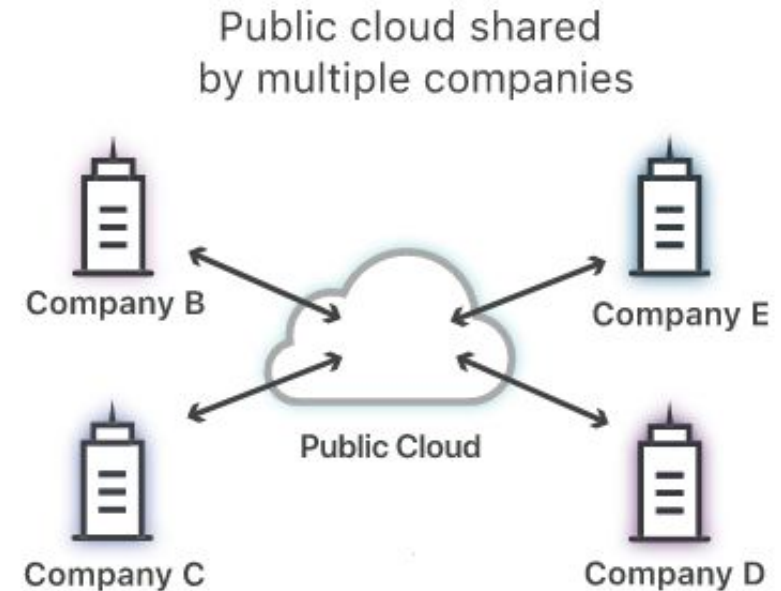
Composants

- L'architecture Cloud est la manière dont les **composants** technologiques **se combinent** pour créer un Cloud (nuage virtuel),
- Les **ressources** sont regroupées via la technologie de virtualisation (ou pas) et **partagées** sur un réseau,
- Les composants d'une architecture Cloud sont les suivants:
 - Une plate-forme frontale (le client ou le terminal utilisé pour accéder au Cloud)
 - Une plate-forme back-end (serveurs et stockage)
 - Un réseau
 - Des logiciels de paramétrage et de gestion

<https://www.vmware.com/>

Types d'architectures Cloud

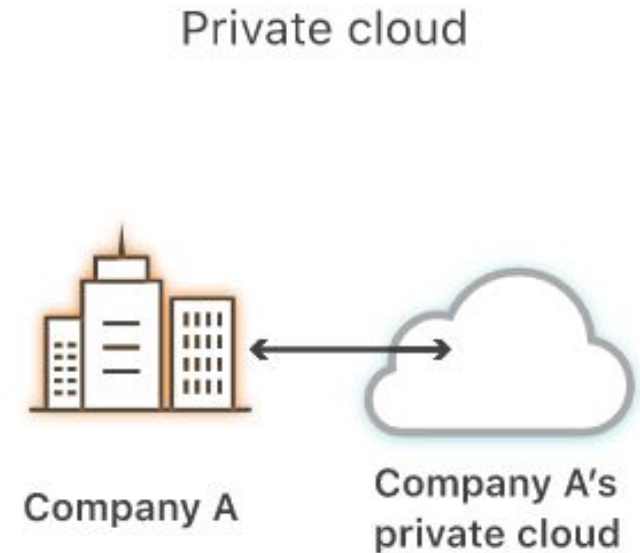
- Architectures Cloud public :
 - Les ressources informatiques sont **détenues et exploitées par un fournisseur** de services,
 - Ces **ressources sont partagées** et redistribuées entre plusieurs locataires,
 - Les **coûts** d'exploitation **réduits**, une **scalabilité simplifiée** et une **maintenance réduite**.



<https://www.vmware.com/>

Types d'architectures Cloud

- Architecture Cloud privé (VPS Virtual Private Cloud)
 - Cloud **détenu et géré en privé**, généralement dans le propre Data Center de la société,
 - Le Cloud privé peut être **géographiquement dispersé**,
 - Généralement plus **coûteuse** que les solutions de Cloud public,
 - Davantage **personnalisable** et peut offrir des **options strictes de sécurité** et de conformité des données.

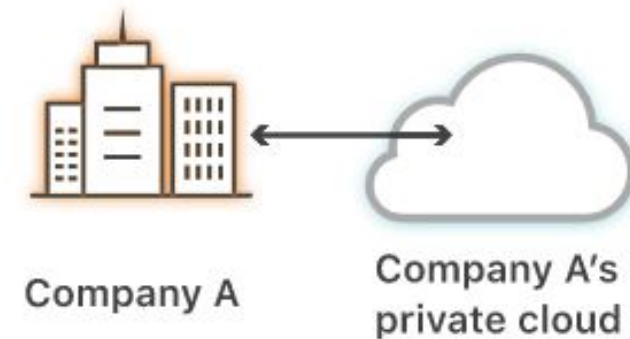


<https://www.vmware.com/>

Types d'architectures Cloud

- Architecture "Hosted Private Cloud"
 - Cloud **détenu par le fournisseur** mais hébergé sur site ou dans le cloud,
 - Les ressources ne sont pas partagées,
 - Le fournisseur doit maintenir les systèmes.

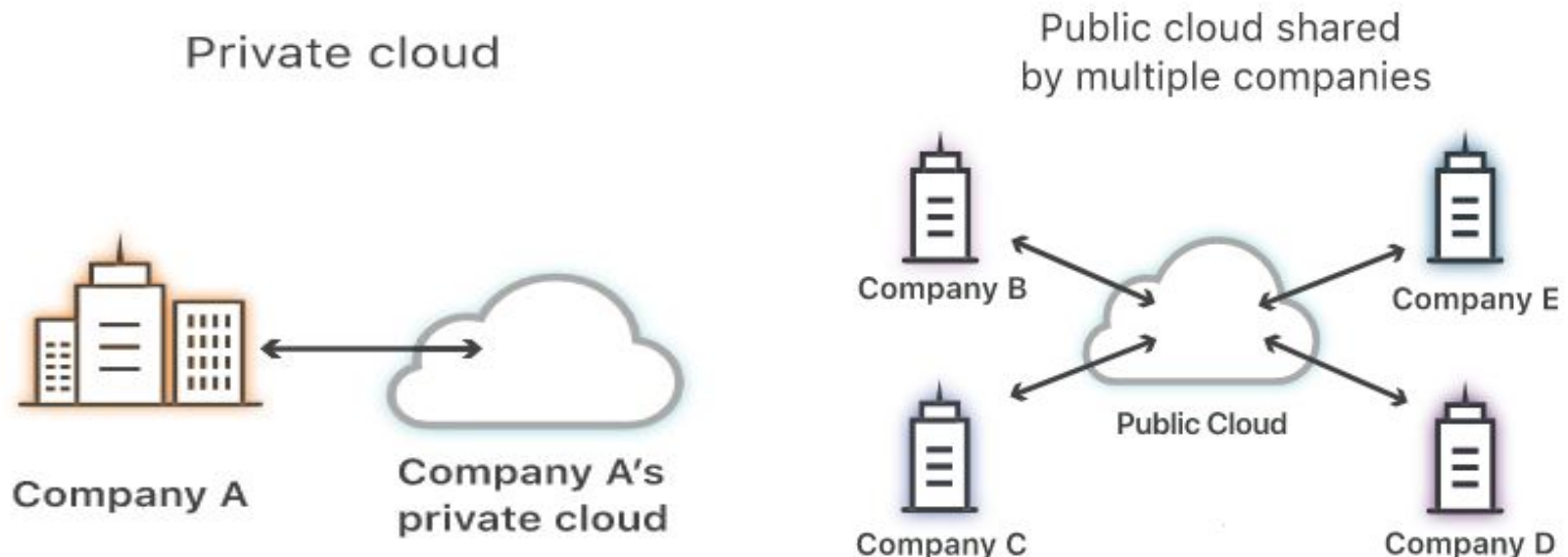
Hosted Private Cloud



<https://www.vmware.com/>

Types d'architectures Cloud

- Architecture Cloud hybride :
 - Cloud public + Cloud privé,
 - Nécessite une base solide pour garantir une infrastructure et des opérations cohérentes.



<https://connect.factorysystemes.fr/>

Architecture CLOUD FOG EDGE

L'architecture cloud est un point de concentration distant qui ne permet pas tout:

- Vitesse d'accès,
- Temps de traitement,
- Remontée massive des informations,
- Stockage parfois inutile ...
- 2022: 40 zettaoctets de données dans le monde cette année. Et d'ici 2025, il est prévu que la sphère de données mondiale atteindra 175 ZB, dont 90 ZB au total proviendront d'appareils périphériques.

La charge est exponentielle, surtout avec l'avènement de l'IIoT. Ce qui induit à une réorganisation vers une infrastructure plus appropriée.

<https://connect.factorysystemes.fr/>

Architecture CLOUD FOG EDGE

- Une architecture **centralisée** (Cloud ou serveurs locaux),
- Exploiter au maximum les capacités de calcul des machines virtuelles ou physiques distantes,
- La **latence** peut impacter fortement les moyens de production,
- Cette couche se concentre essentiellement sur le Big Data, le Data analysis et le machine learning, application Web ...

<https://connect.factorysystemes.fr/>

<https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.500-325.pdf>

Architecture CLOUD FOG EDGE

- FOG : l'informatique dans le brouillard,
- Une extension du Cloud Computing,
- Physiquement situés sur le réseau d'entreprise et relais entre les équipements couche basse IoT et la couche haute IT (Information Technology),
- Un traitement local des applications de production en temps réel,
- Le Fog computing est principalement assimilé à des passerelles de communications industrielles de terrain vers des protocoles IoT.

<https://connect.factorysystemes.fr/>

<https://shop.factorysystemes.fr/product-tag/edge-computing/>

Architecture CLOUD FOG EDGE

- Le Edge computing est directement lié aux **moyens de production** et autres capteurs,
- Le Edge computing réalise localement le traitement applicatif et les calculs,
- Prise de décision en temps réel où aucun temps de latence n'est acceptable,
- L'analyse complète des données peut être réalisée dans le Cloud computing.

<https://connect.factorysystemes.fr/>

<https://shop.factorysystemes.fr/product-tag/edge-computing/>

<https://www.omg.org/cloud/deliverables/CSCC-Cloud-Customer-Architecture-for-IoT.pdf>

Architecture CLOUD FOG EDGE

“on-premises data center” (serveurs de données sur site)

- Modèle hybride local / cloud, ou tout local,
- “All On-Premise”: tout en local,
- Stockage partiel / total des données sur site,
- Traitement puis transfert partiel sur le cloud,
- Utilisation pour des processus rapides de pilotage des unités de production.

<https://connect.factorysystemes.fr/>

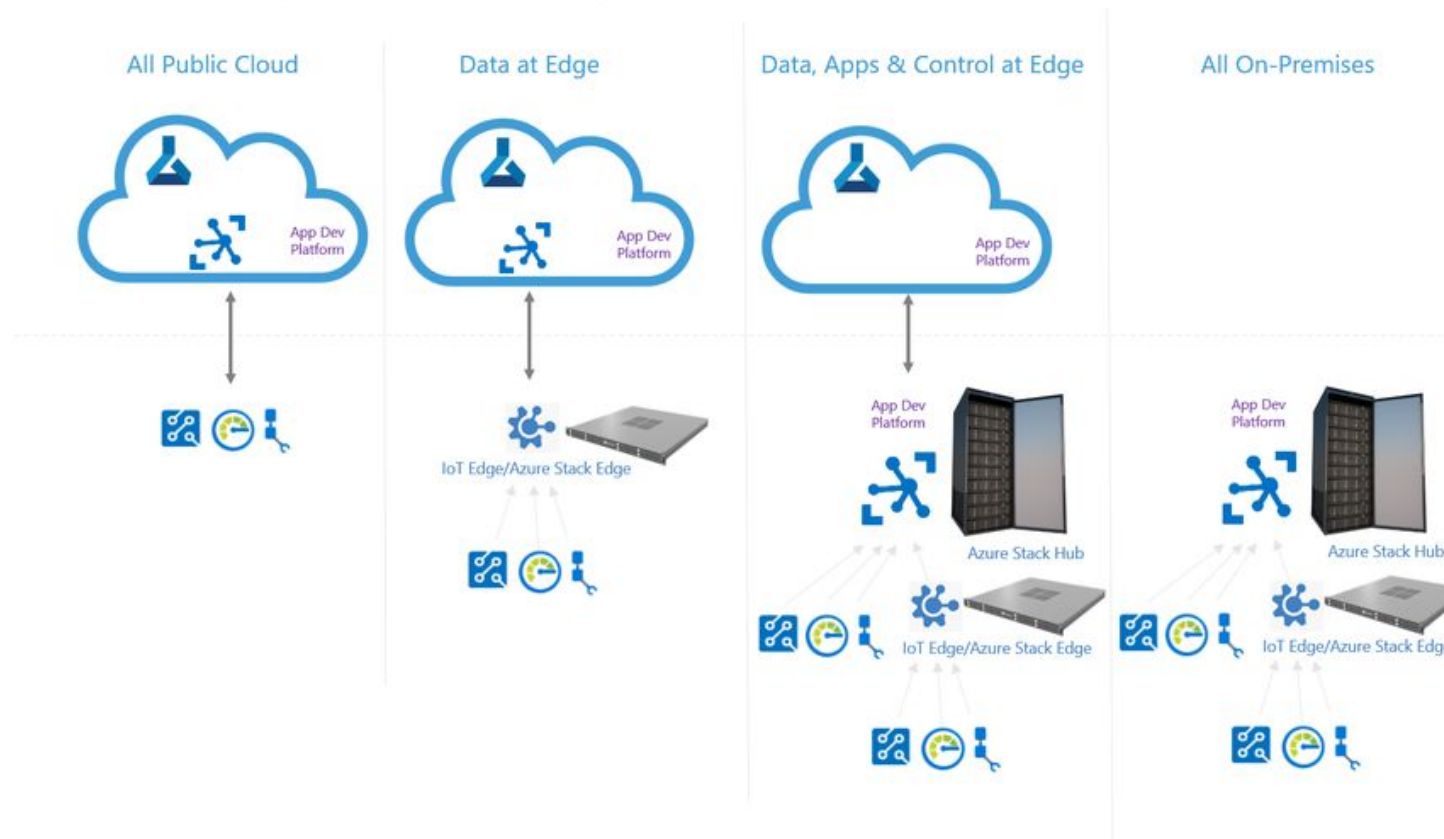
<https://shop.factorysystemes.fr/product-tag/edge-computing/>

<https://www.omg.org/cloud/deliverables/CSCC-Cloud-Customer-Architecture-for-IoT.pdf>

Architecture CLOUD FOG EDGE

on-premises data center (serveurs de données sur site) / Exemple

Processing on the Edge - Continuum



<https://connect.factorysystemes.fr/>

Architecture CLOUD FOG EDGE / Traitement des données

CLOUD

Traitement Big Data
Logique Business
Stockage données

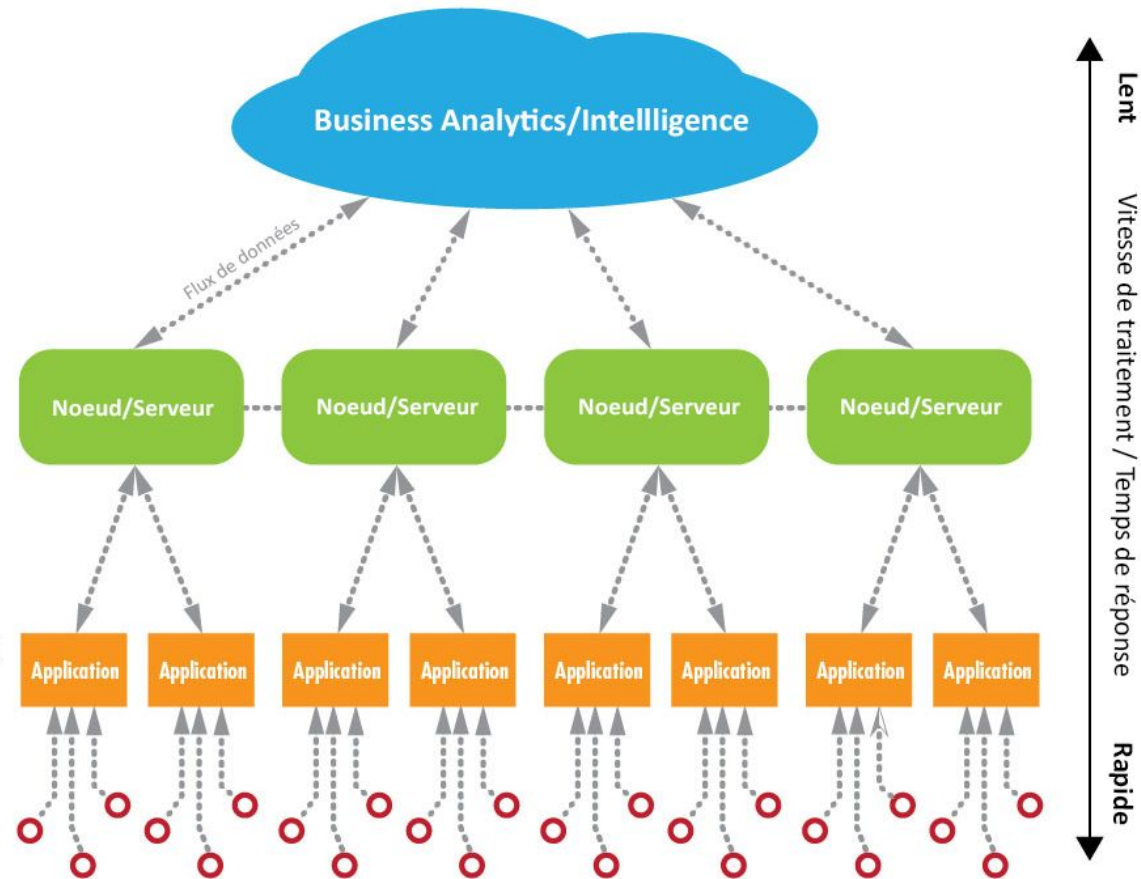
FOG

Réseau local
Analyse données temps réel
Virtualisation/Standardisation

EDGE

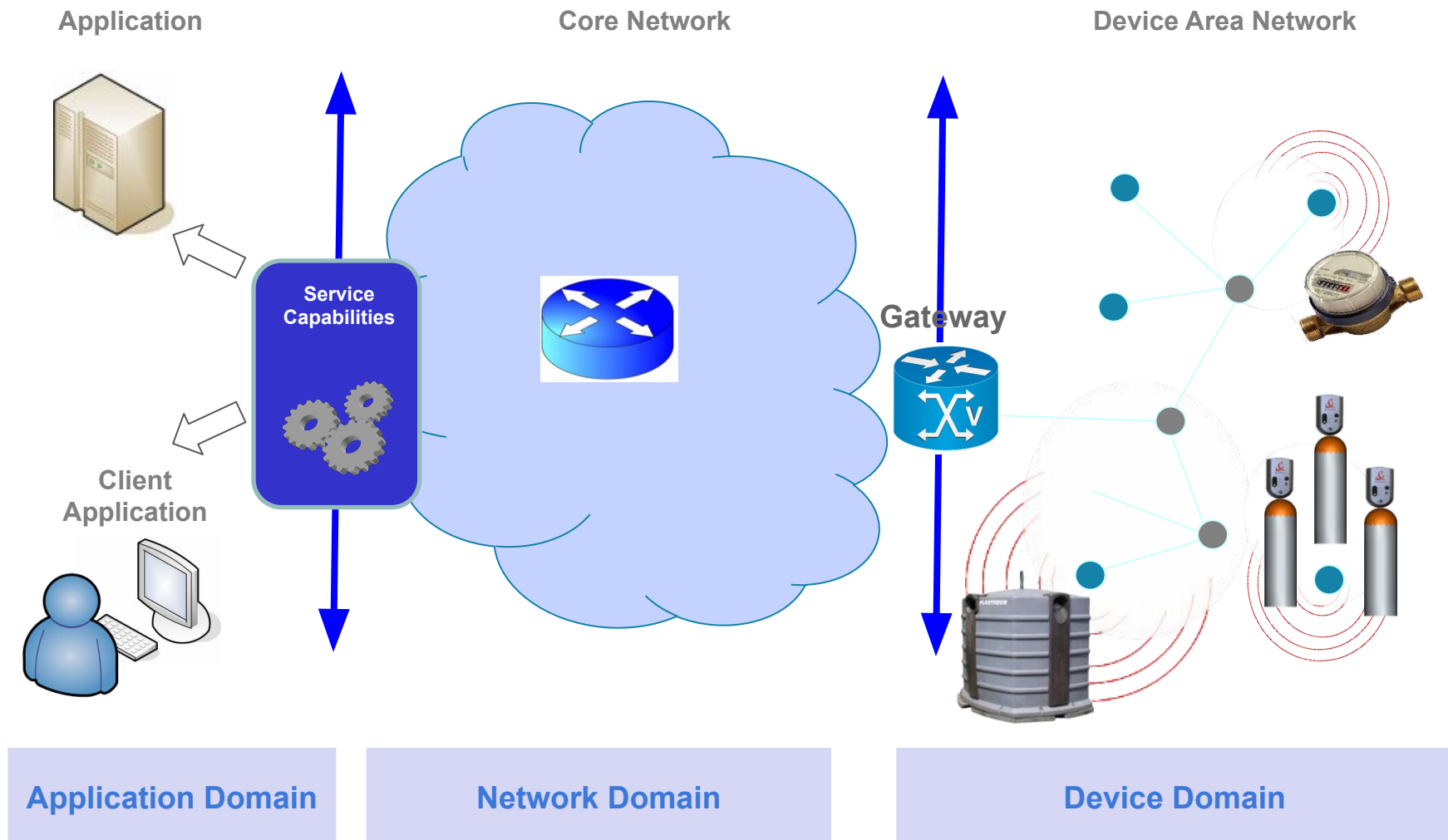
Large volume de données temps réel
Visualisation données à la source /Local
Systèmes embarqués
PCs industriels
Passerelles/Gateway
Stockage micro données

Capteurs & Contrôleurs (création des données)



<http://www.etsi.org/>

Structure commune du réseau



<http://www.etsi.org/>

Structure commune: Passerelle (Gateway)

- Une passerelle gère le trafic entre les réseaux qui utilisent des protocoles différents.
- Une passerelle est responsable de la traduction du protocole et d'autres tâches d'interopérabilité.
- Un périphérique de passerelle IoT est utilisé car certains périphériques ne contiennent pas la pile réseau requise pour la connectivité Internet, un périphérique de passerelle agit comme un proxy, recevant les données des périphériques et les conditionnant pour la transmission sur TCP/IP.

<http://www.etsi.org/>

Structure commune: Domaine application

- Application utilisateur final,
- Application de catalogue prête à l'emploi, juste quelques-unes pour la personnalisation,
- L'application utilise toute la capacité du système : du capteur au big data :
 - La prise de décision,
 - Analyse,
 - Interaction avec le milieu naturel...

<http://www.etsi.org/>

Structure commune

Domaine réseau

- Indépendant des technologies de transport,
- Toute la technologie de convergence IP,
- Utilisation du système de passerelle ou accès direct aux plateformes M2M

Domaine capteurs

- Système de capteurs et d'actionneurs,
- Système de communication,
- Effectue une tâche demandée.

Agenda

- Architecture globale du cyber-système,
- Modèle abstrait d'architecture de l'IoT,
- Notion d'architecture système,
- Architectures classiques : self, home, city, work, commune,
- Les architecture verticales et les évolutions,
- Les architectures Cloud, Edge, Fog,
- **Les services cloud comme solution industrielle : LaaS, PaaS, SaaS, Caas, Daas, Baas,**
- Les composants d'une plateforme open source,
- IoT et design pattern.

<https://www.redhat.com/>

Business Application

Network Infrastructure

Application
Infrastructure

Service cloud

Trois principaux modèles de service pour répondre aux entités logiques:

- **SaaS** : Software as a Service
- **PaaS** : Plate-forme sous forme de service
- **IaaS** : Infrastructure sous forme de service

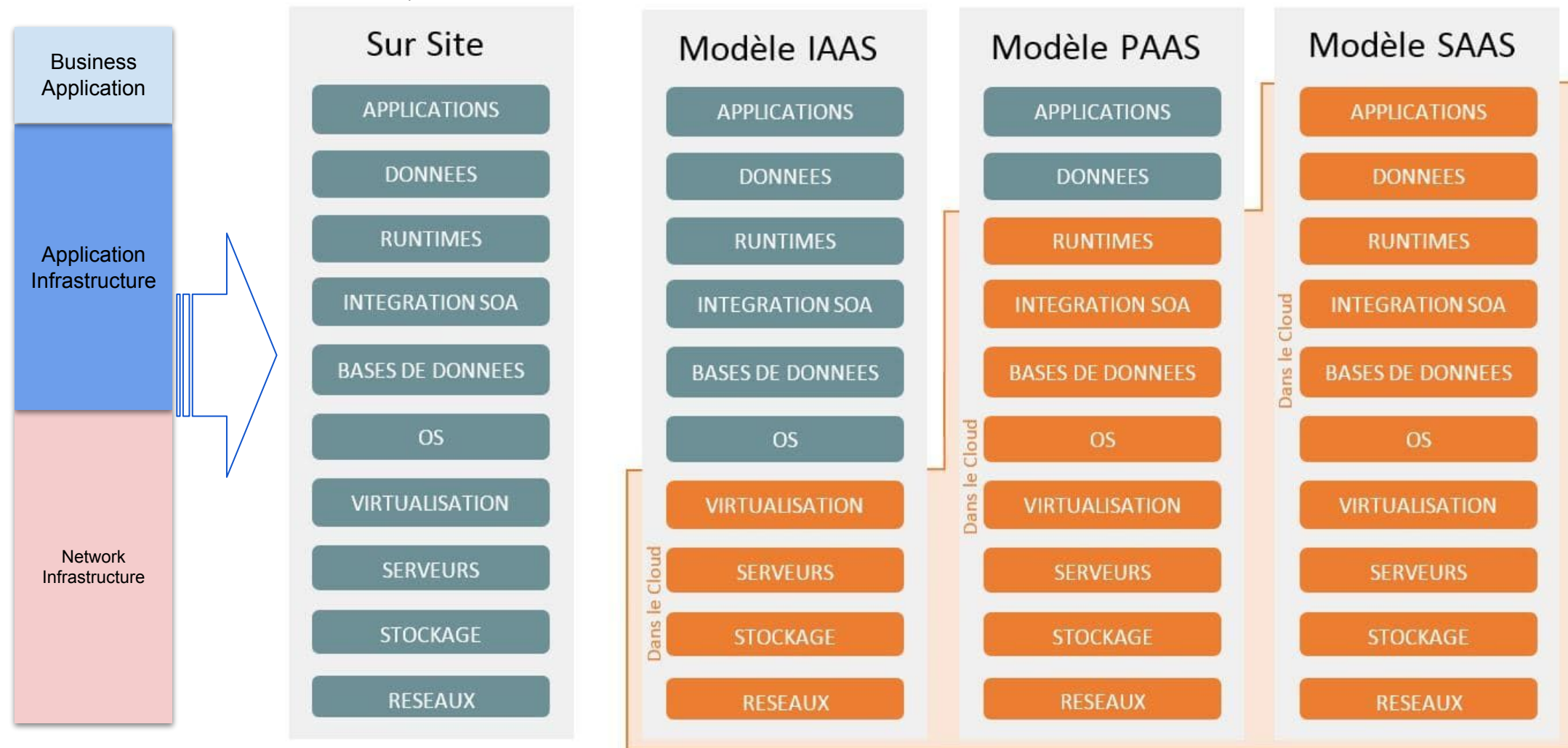
Autres services:

- **Caas** : Containers-as-a-Service, applications conteneurisées
- **DaaS** : Data As A Service, un dépôt de données
- **BaaS** : Backend As A Service: outils pour le développement logiciel

<https://www.vmware.com/>

Type de service cloud

on-premise



<https://www.vmware.com/>

Infrastructure as a service (IaaS) :

- Le fournisseur de services fournit **un système de virtualisation sans aucun autre service**,
- Les utilisateurs gèrent leurs, **OS, logiciels et applications**, et ne paient que pour la capacité dont ils ont besoin à un moment donné.

<https://www.vmware.com/>

Plate-forme as a service (PaaS) :

- Plate-forme informatique et une pile de solutions sous forme de service permettant à une entreprise de **créer une application ou un service**,
- Le fournisseur de services fournit **les réseaux, les serveurs et le stockage, les OS** requis pour **héberger** une application,
- L'utilisateur final supervise les paramètres de **déploiement et de configuration du logiciel**.

<https://www.vmware.com/>

Software as a Service (SaaS) : la plateforme de service gère presque tout: les applications, les logiciels, les OS (Operating System).



<https://www.vmware.com/>

Infrastructure Caas (Containers-as-a-Service) :

- CaaS est considéré comme une sorte de sous-ensemble de l'Infrastructure-as-a-Service (IaaS),
- Un service cloud qui permet de gérer et de déployer des applications à l'aide d'une abstraction basée sur des conteneurs,
- CaaS est particulièrement utile aux développeurs pour créer des applications conteneurisées plus sécurisées et également évolutives,
- **Portabilité** : les applications développées dans des conteneurs sont autonomes
- **Évolutivité** : les conteneurs ont la capacité de s'adapter horizontalement, ce qui signifie qu'un utilisateur peut multiplier des conteneurs identiques
- **Efficacité** : les conteneurs nécessitent moins de ressources que les machines virtuelles (VM) car ils n'ont pas besoin d'un système d'exploitation distinct.
- **Orchestration de conteneurs**: automatiser le développement, la gestion et la mise à l'échelle des applications en regroupant les conteneurs exécutés

<https://www.vmware.com/>

Infrastructure Desktop as a Service (DaaS):

- Offre de cloud computing où un fournisseur de services fournit des bureaux virtuels aux utilisateurs finaux sur Internet,,
- Environnement de travail complet dans le cloud,
- Les utilisateurs peuvent se connecter à leur bureau virtuel depuis n'importe où.

<https://www.vmware.com/>

Infrastructure BaaS (Backend As A Service) :

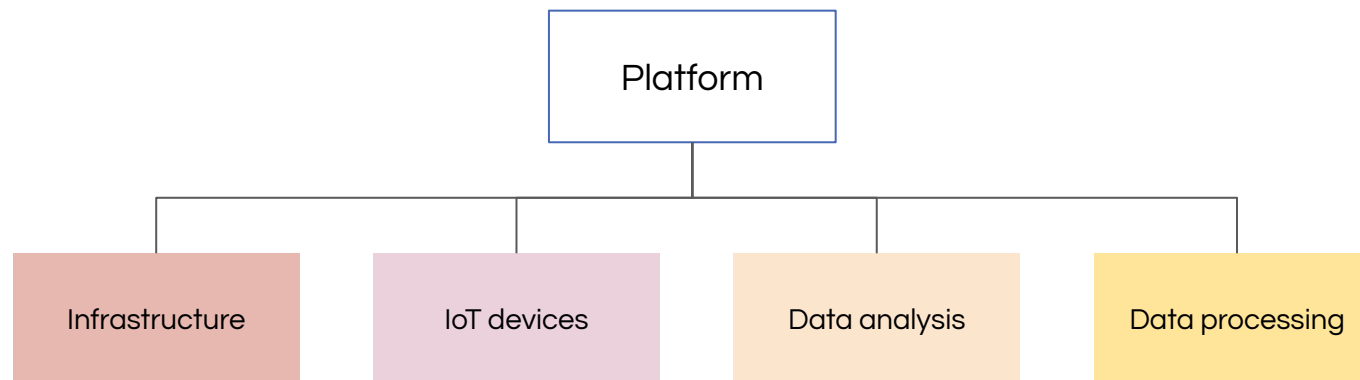
- Service cloud dans lequel les développeurs externalisent tous les aspects de l'arrière-plan d'une application web ou mobile,
- Les fournisseurs de BaaS fournissent des logiciels pré-écrits pour ces activités
 - authentification des utilisateurs,
 - la gestion des bases de données,
 - la mise à jour à distance ...
- Mobile-Backend-as-a-Service (MBaaS) est un BaaS destiné spécifiquement à la création d'applications mobiles.

Agenda

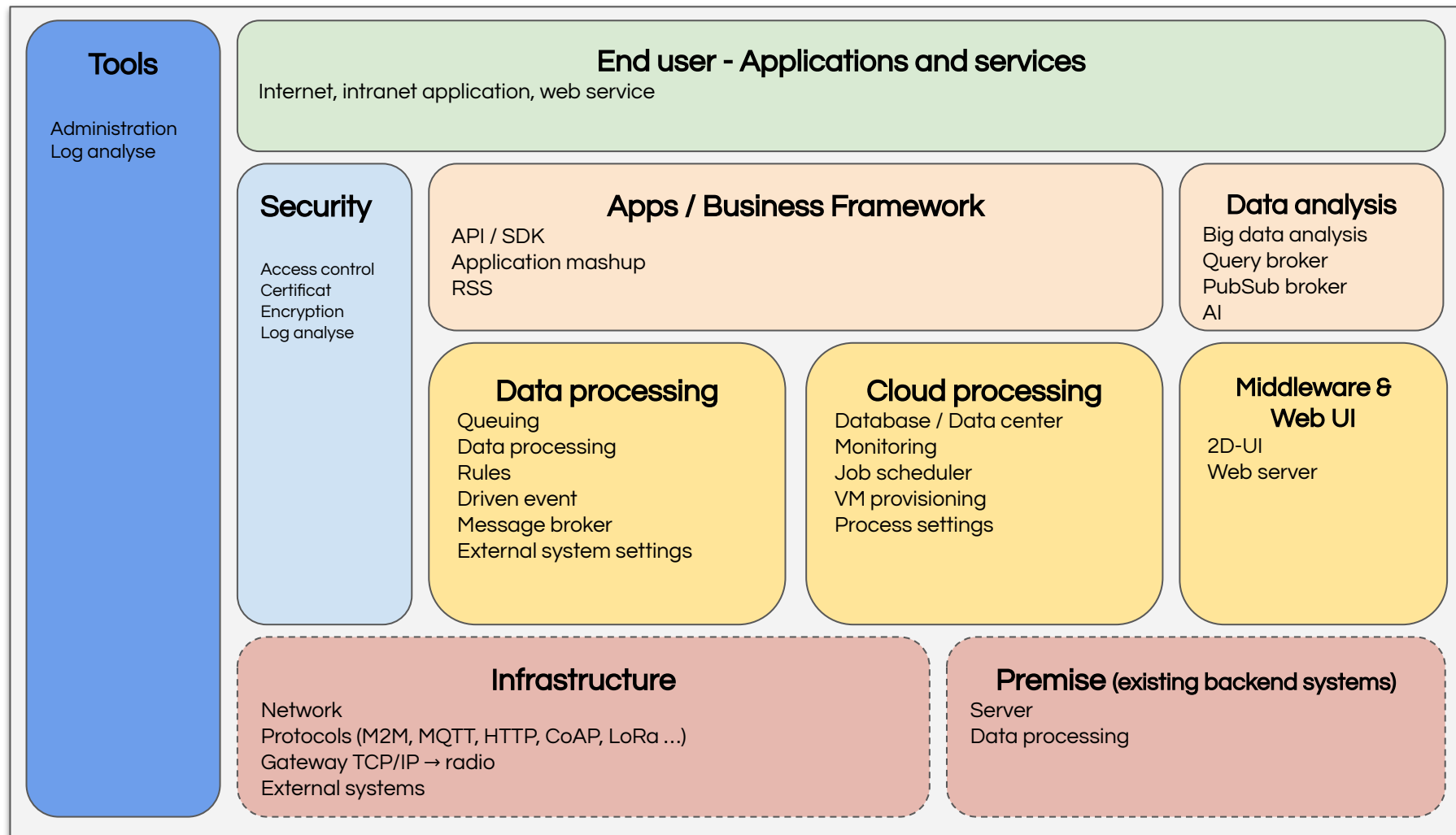
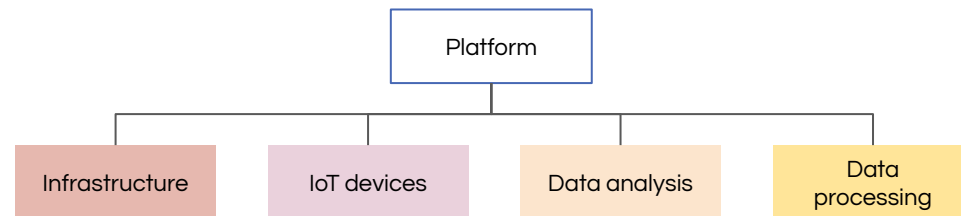
- Architecture globale du cyber-système,
- Modèle abstrait d'architecture de l'IoT,
- Notion d'architecture système,
- Architectures classiques : self, home, city, work, commune,
- Les architecture verticales et les évolutions,
- Les architectures Cloud, Edge, Fog,
- Les services cloud comme solution industrielle : LaaS, PaaS, SaaS, Caas, Daas, Baas,
- Les solutions open source : notion de l'open source, licence, mise en oeuvre, communauté,
- **Les composants d'une plateforme open source,**
- IoT et design pattern.

Architecture de haut niveau (simplifiée)

- Une architecture basée sur des composants permet de remplacer un composant par une autre version voire une autre technologie.
- L'architecture du système est évolutive et modulaire.



Architecture de haut niveau



Agenda

- Architecture globale du cyber-système,
- Modèle abstrait d'architecture de l'IoT,
- Notion d'architecture système,
- Architectures classiques : self, home, city, work, commune,
- Les architecture verticales et les évolutions,
- Les architectures Cloud, Edge, Fog,
- Les services cloud comme solution industrielle : LaaS, PaaS, SaaS, Caas, Daas, Baas,
- Les solutions open source : notion de l'open source, licence, mise en oeuvre, communauté,
- Les composants d'une plateforme open source,
- **IoT et design pattern.**

<https://computationpatterns.com/1262/iot-design-patterns/>
<https://community.arm.com/>

Les modèles de conception (Design pattern)

- Les modèles de conception sont des solutions réutilisables à des problèmes courants,
- L'IoT présente des problèmes de conception dans de nombreux domaines et à de nombreux niveaux du système,
- Il existe de nombreux cas d'utilisation divers, avec des contraintes de ressources différentes et avec de nombreux standards,
- Le but est de fournir une compréhension de l'utilisation des différents composants du système dans un contexte système.

<https://computationpatterns.com/1262/iot-design-patterns/>
<https://community.arm.com/>

Les modèles de conception (Design pattern)

- Il existe des Pattern pour pratiquement tout ! :
 - Architecture des logiciels,
 - Architecture des systèmes,
 - Architecture de communication ...