CATIE

Prototypage IoT avec 6TRON

Partie 1 : Contexte et Outils

20 janvier 2025 – Thomas Habrant

RÉGION Nouvelle-Aquitaine

Objectifs

- Comprendre les spécificités de l'IoT
- Découvrir les outils utilisés en entreprise pour développer un système embarqué
- Développer un objet connecté

Partie 1 : Contexte et Outils

Contexte

- L'Internet des Objets
- Les technos clés de l'IoT
- Les contraintes de l'embarqué

6TRON

- Du prototype à l'objet l'embarqué
- Une plateforme matérielle
- Un support logiciel
- Ressources

Travaux Pratiques

- Installation des outils
- Test de l'installation
- Tutoriel Git
- Blinky !

De M2M…

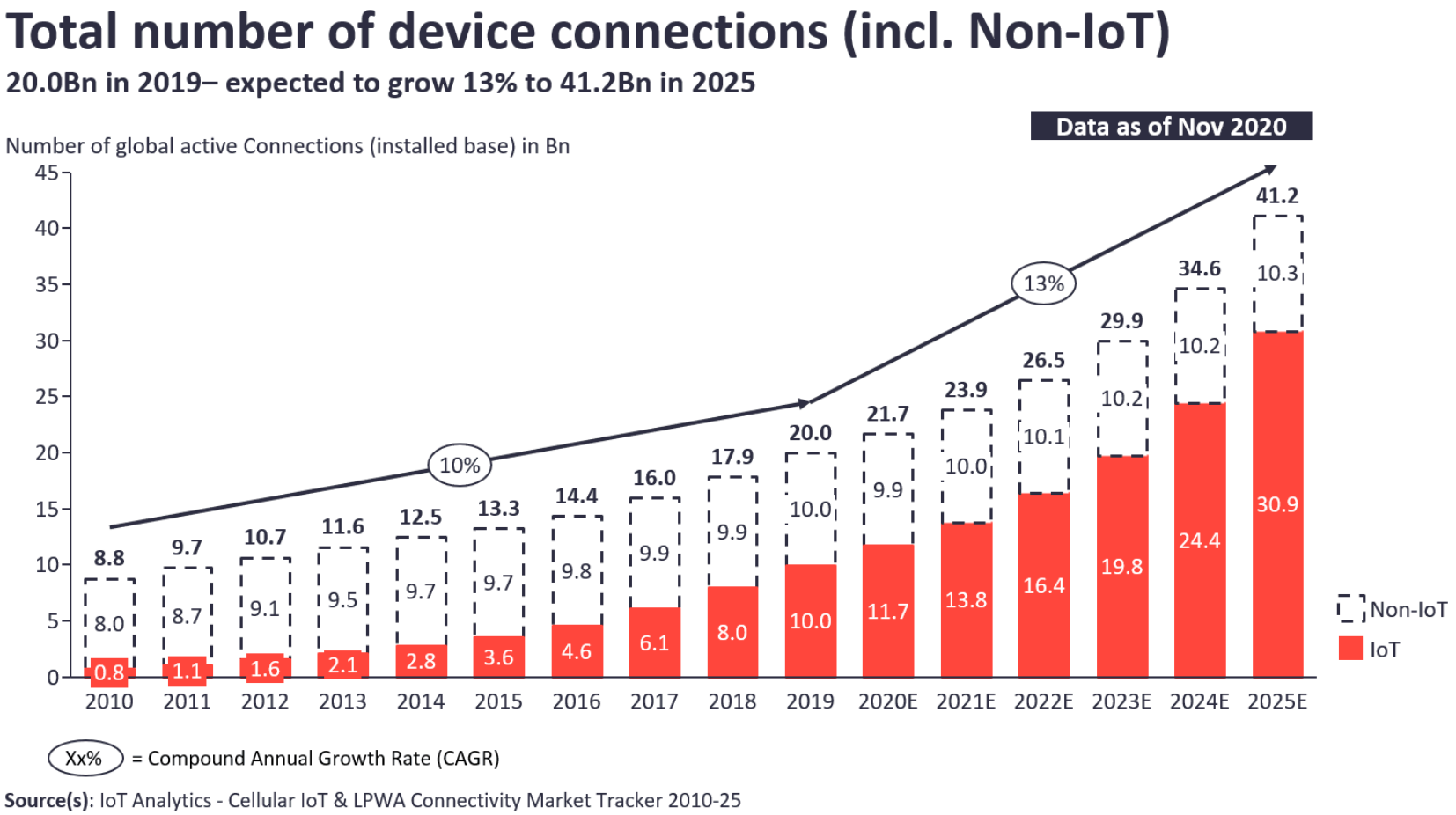
- M2M : Machine to Machine
  Technologies donnant à des objets "intelligents" des moyens d'interagir sans intervention humaine
- Exemples :
  - Automobile : système de freinage anti-bloquant (ABS), régulateur de vitesse
  - Industrie, Santé : contrôle à distance d'équipements
  - Services : télésurveillance, affichage des temps d'attente à un arrêt de bus

… À IoT

- IoT : Internet des Objets
  Réseau d'objets et dispositifs interconnectés par Internet
- Les objets ne communiquent pas entre eux directement,
  mais via une plateforme
- Exemples :
  - Smart Home : Amazon Alexa, Google Home, thermostats connectés, etc.
  - Wearables : montres, ceintures fitness, etc.
  - B2B : places de parking, conteneurs ordures, etc.

Contexte – L'Internet des Objets

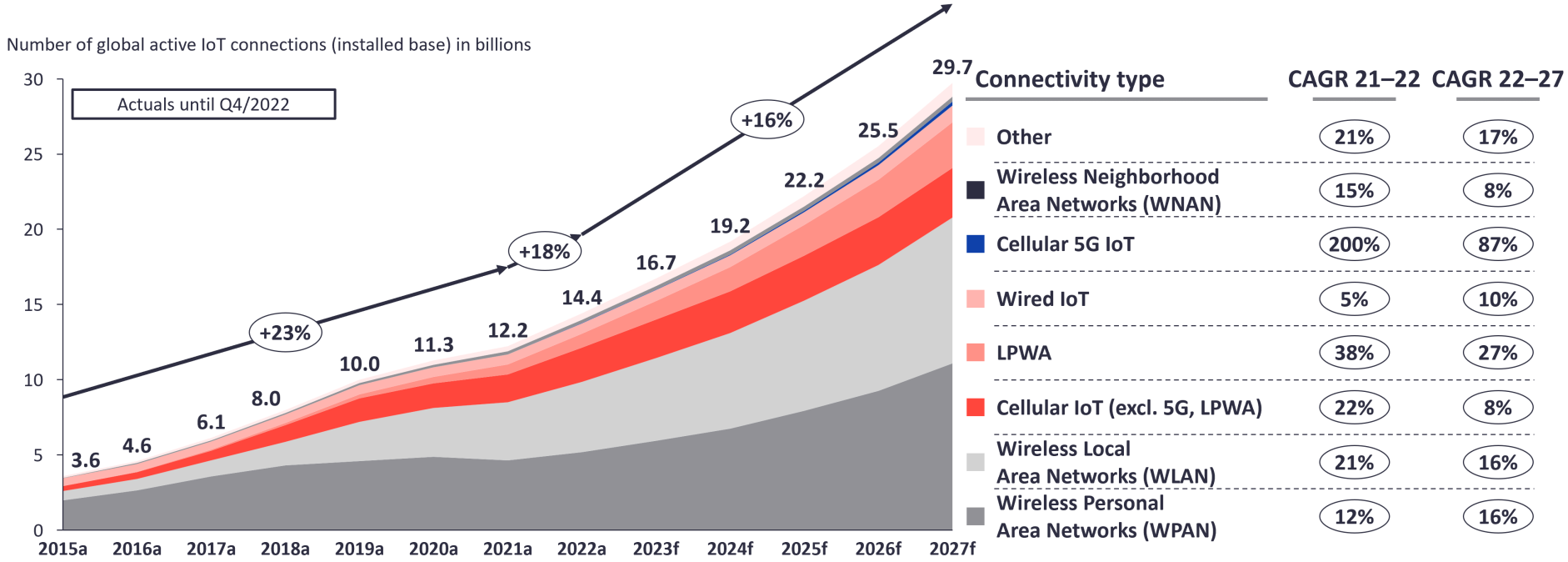
- Croissance exponentielle
- IoT > non-IoT smartphones, ordinateurs, etc.



Contexte – L'Internet des Objets

- BLE, Wi-Fi ++
- 5G IoT, LPWAN ↗

Global IoT market forecast (in billions of connected IoT devices)

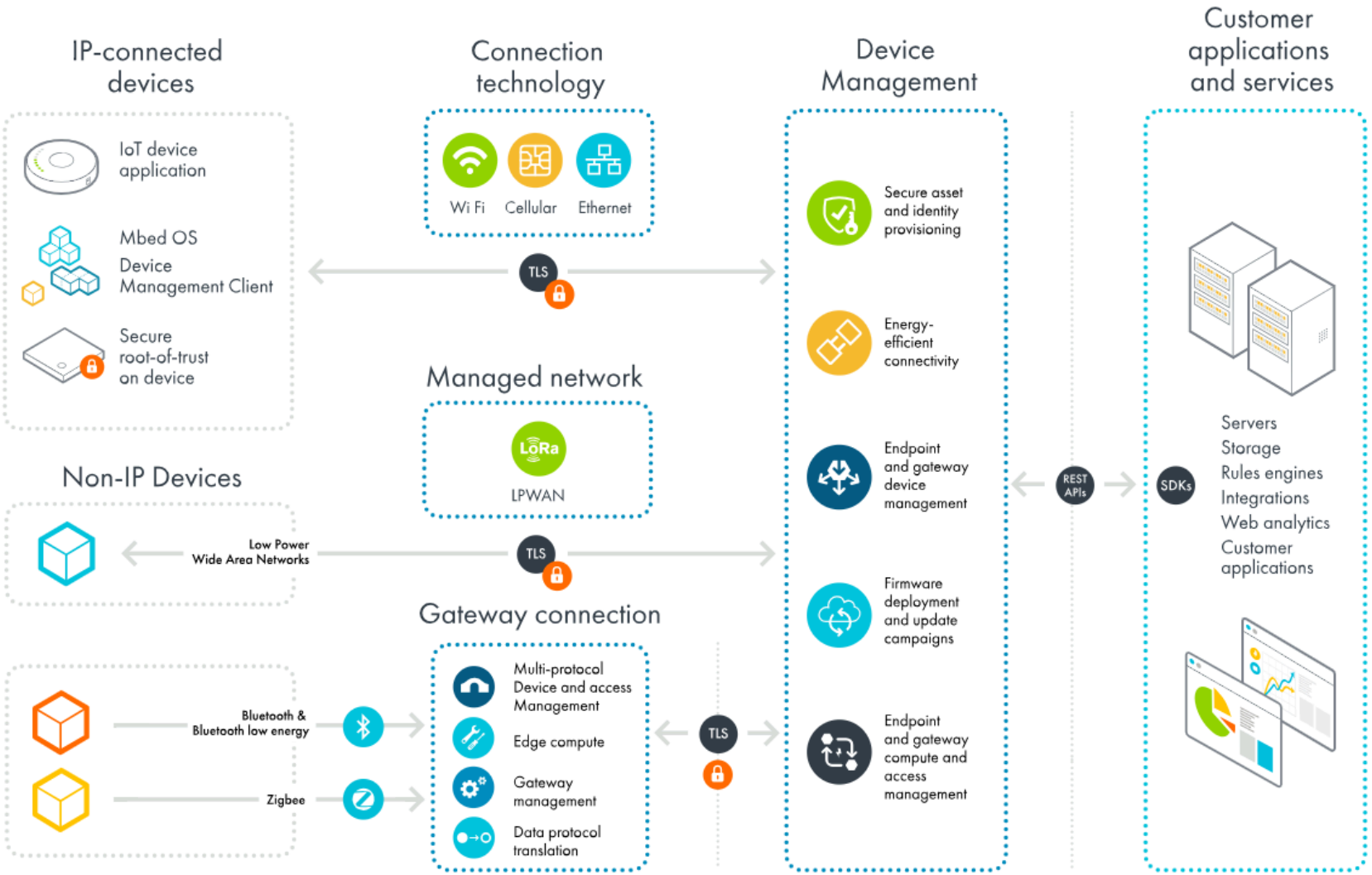


Connectivity type	CAGR 21–22	CAGR 22–27
Other	21%	17%
Wireless Neighborhood Area Networks (WNAN)	15%	8%
Cellular 5G IoT	200%	87%
Wired IoT	5%	10%
LPWA	38%	27%
Cellular IoT (excl. 5G, LPWA)	22%	8%
Wireless Local Area Networks (WLAN)	21%	16%
Wireless Personal Area Networks (WPAN)	12%	16%

Du capteur au cloud

- Identification : RFID, GPS, etc.
- Capteurs : centrale inertielle, température, caméra, etc.
- Connexion réseau : Bluetooth, Wi-Fi, ZigBee, LoRa, 5G, etc.
- Sécurité (authentification, intégrité, confidentialité) : chiffrement, élément sécurisé, etc.
- Plateforme IoT : intégration des données, systèmes décisionnels et IA, base de données, etc.

Contexte – Les technos clés de l'IoT



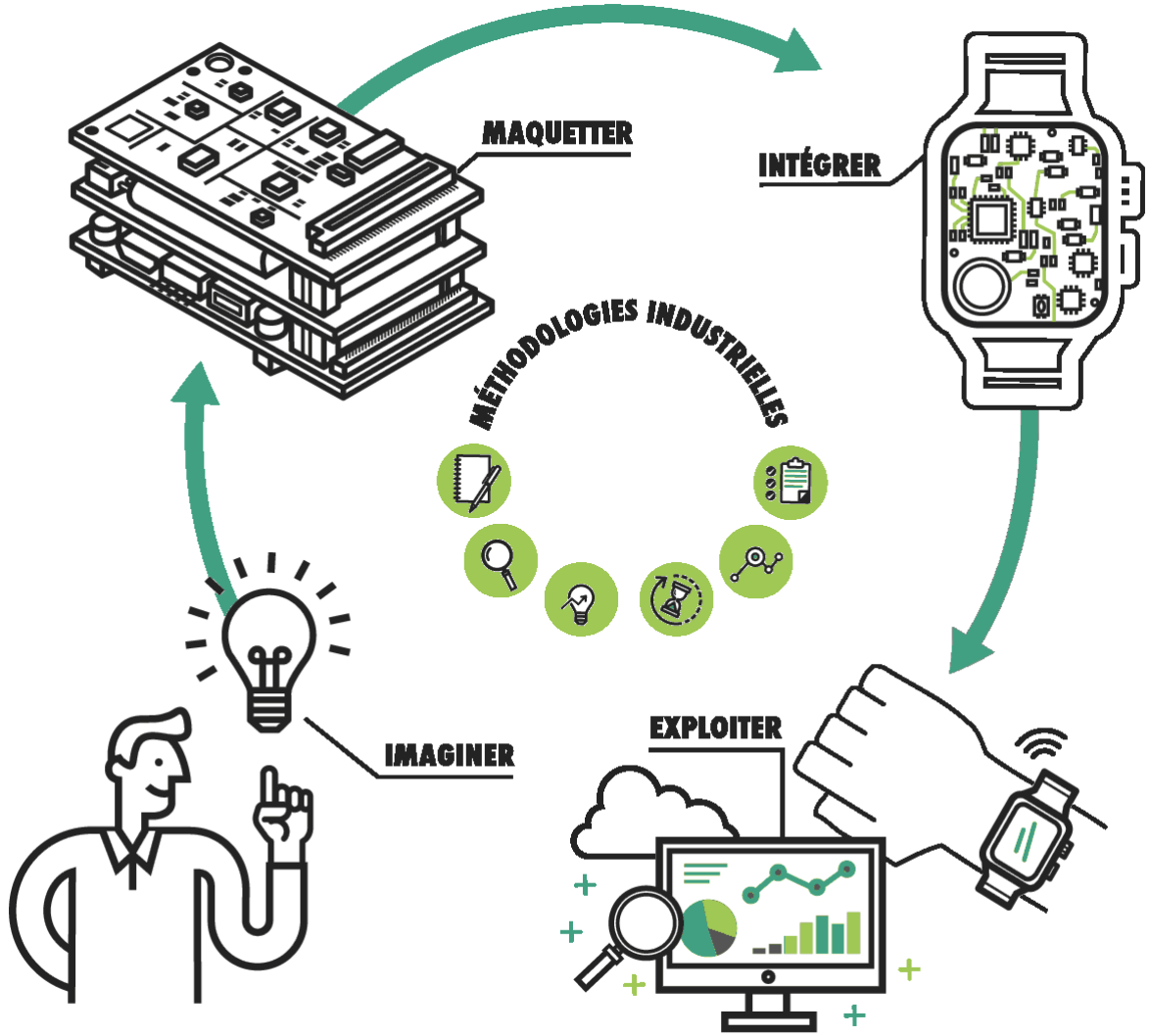
Contexte – Les contraintes de l'embarqué

> Un système embarqué est un système électronique et informatique autonome, souvent temps réel, spécialisé dans une tâche précise.

Soumis à certaines contraintes techniques :

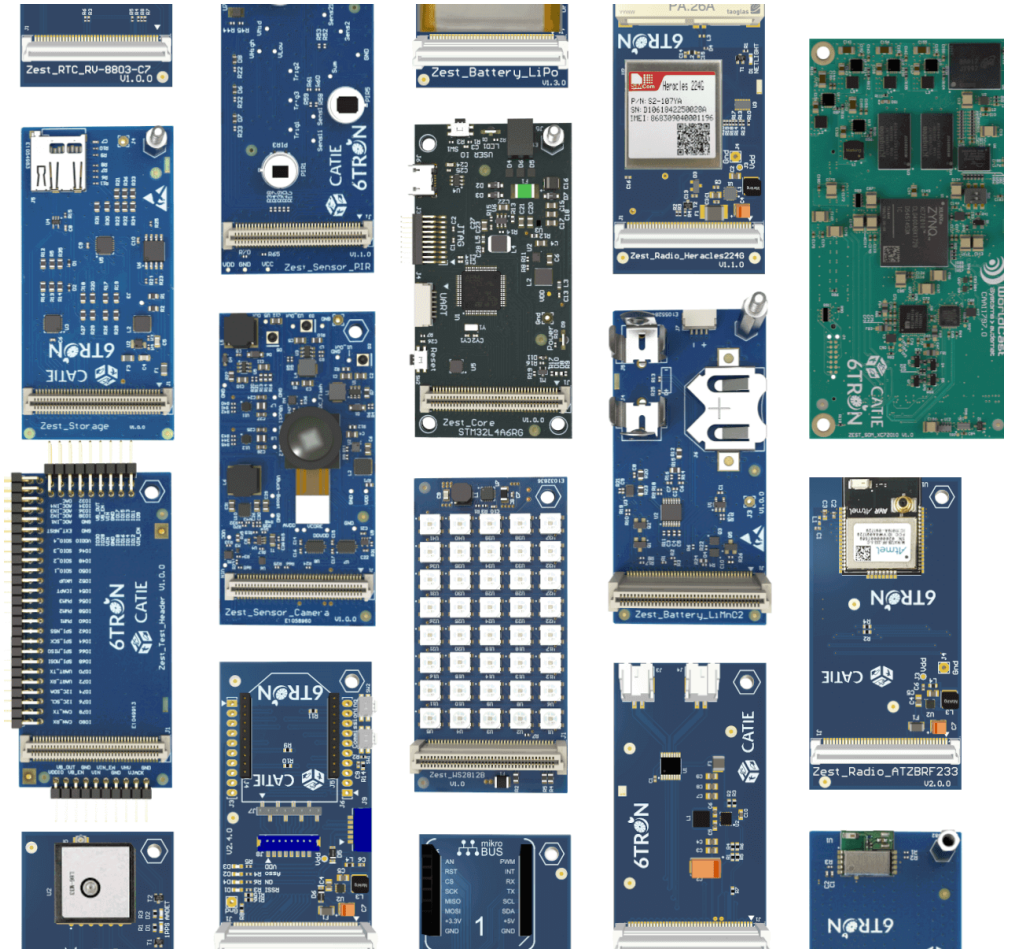
- Ressources limitées (mémoire et puissance)
- Consommation d’énergie
- Miniaturisation
- Temporel : les temps d'exécution et l'échéance temporelle d’une tâche sont déterminés
- La sécurité : mise à jour du firmware

6TRON – Du prototype à l'objet embarqué

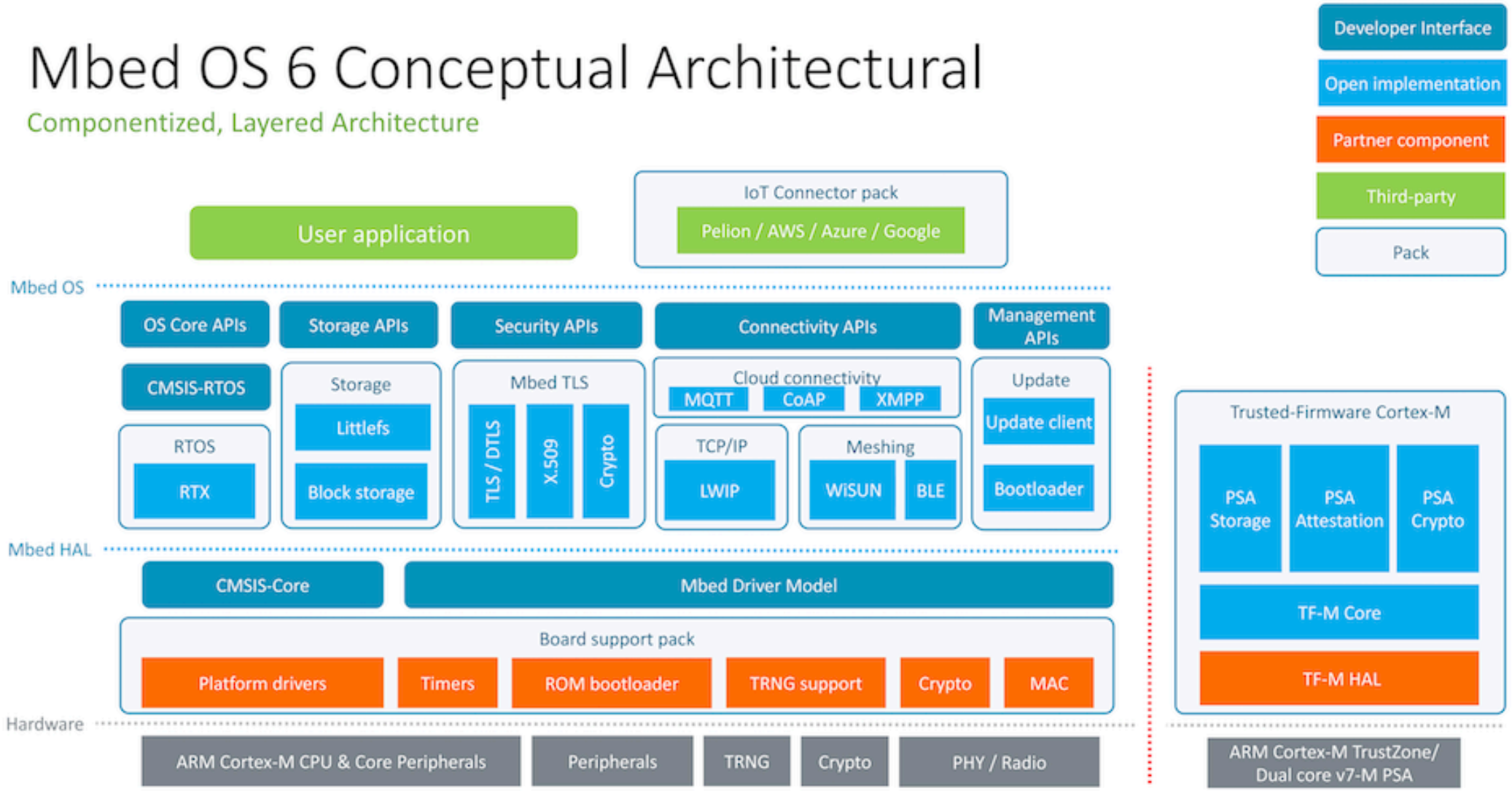


6TRON – Une plateforme matérielle

- 8 ans d'existence
- 60+ cartes disponibles
- du microcontrôleur au FPGA
- 50+ projets accompagnés



6TRON – Un support logiciel



6TRON - Ressources

- Documentation en ligne :
  https://6tron.io
- Dépôts GitHub :
  https://github.com/topics/6tron

Travaux Pratiques - Outils utilisés

Outils nécessaires au développement :

- Git : logiciel de gestion de versions
- VS Code : environnement de développement intégré
- GNU Toolchain : chaîne de compilation GNU
- Mbed CLI : utilitaire de configuration et de gestion de la compilation
- 6TRON Flash : utilitaire de programmation
- J-Link tools : drivers et outils pour sonde de débogage

Travaux Pratiques - Création de l'environnement

Télécharger ou cloner le dépôt Git suivant :

https://github.com/catie-aq/6tron_training-mbed-os-basic-template

Puis, dans un terminal :

```
cd 6tron_training-mbed-os-basic-template
```

```
export MBED_GCC_ARM_PATH="$HOME/.local/opt/gcc-arm-none-eabi/gcc-arm-none-eabi-10.3-2021.07/bin/"
export PATH=$MBED_GCC_ARM_PATH:$PATH
```

```
cd 6tron_training-mbed-os-basic-template
code .
```

Compilation et programmation

Dans VS Code, ouvrir un premier terminal pour compiler l'application et programmer la flash :

```
mbed deploy
mbed compile
sixtron_flash
```

La programmation est elle volatile ou non volatile ?

Visualiser l'exécution du programme

Ouverture de la liaison série dans un second terminal :

```
minicom -D /dev/ttyUSB0 -b 9600
```

Visualiser les messages :

```
Alive!
Alive!
Alive!
```

Travaux Pratiques - Git

Git est un logiciel de gestion de version décentralisé, libre et gratuit.

> Un **logiciel de gestion de versions** agit sur une arborescence de fichiers afin de conserver toutes les versions des fichiers, ainsi que les différences entre les fichiers.

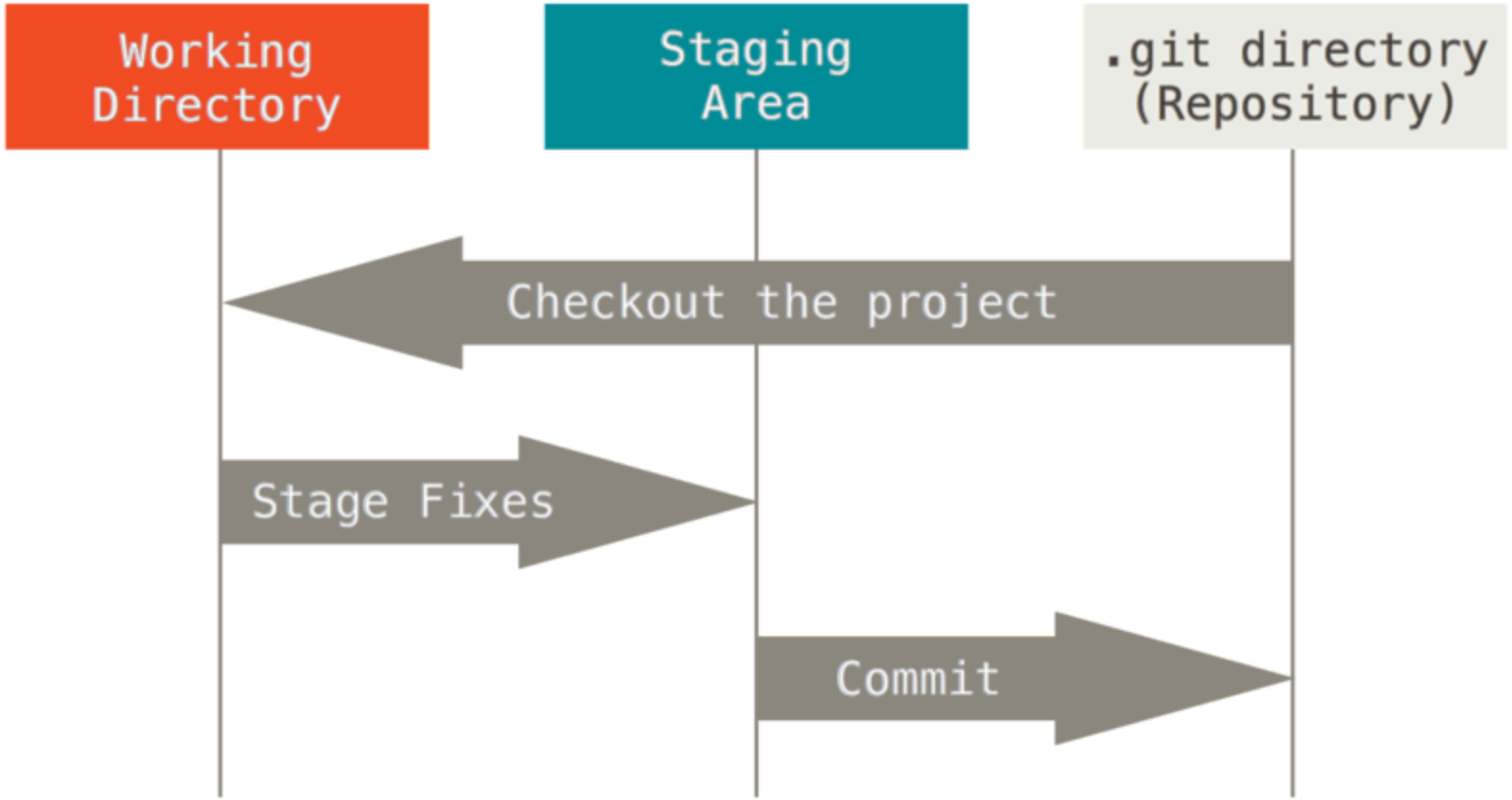
> Un **logiciel libre** est un logiciel dont l'utilisation, l'étude, la modification et la duplication par autrui en vue de sa diffusion sont permises, techniquement et juridiquement.

Git est le logiciel de gestion de version le plus utilisé, depuis plus de 10 ans.
Git est à la base de GitHub, le plus important hébergeur de code informatique.

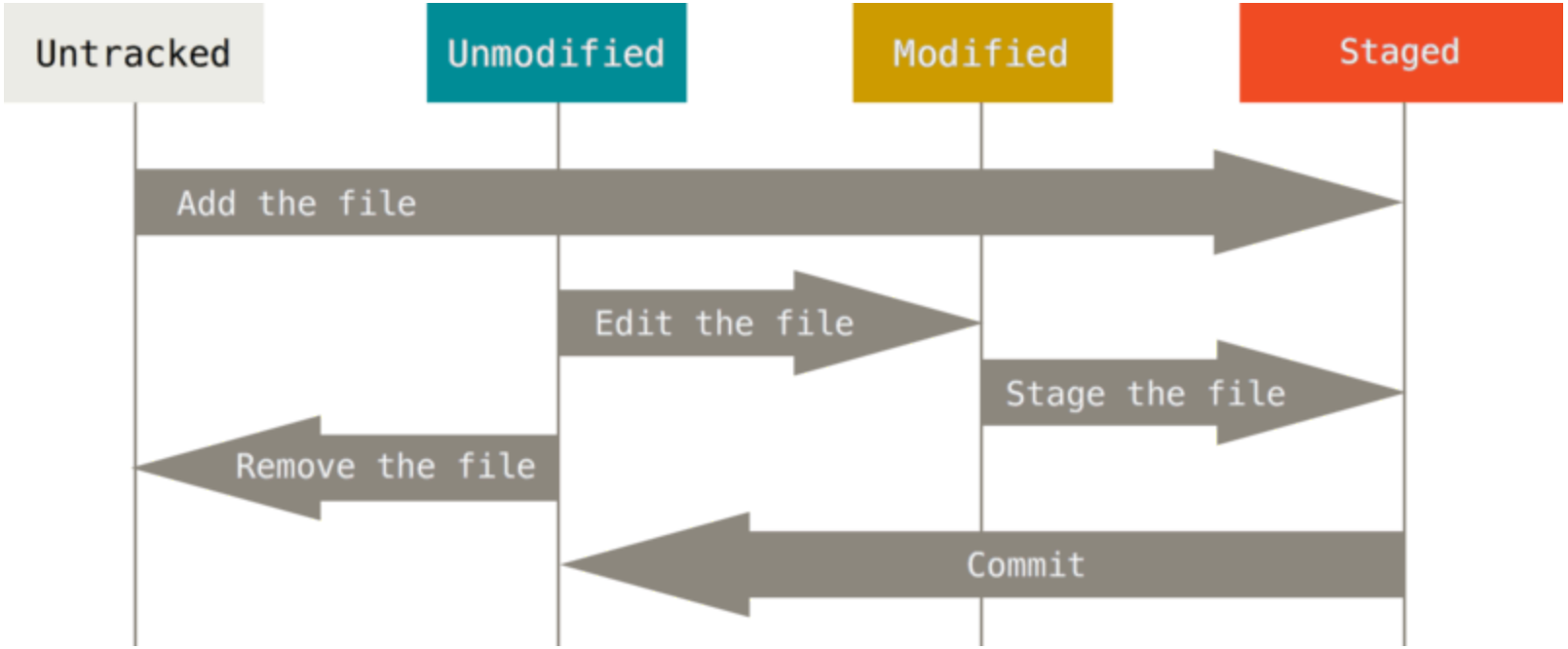
Travaux Pratiques - Git, notions clés

- `Repository` / Dépôt : système de stockage des fichiers versionnés, sous forme d'arbre de versions
- `Commits` : enregistrement d'une nouvelle version du dépôt
- `Branches` : Branche de l'arbre des versions du dépôt
- `Working Files` / Copie de travail : zone d'édition des fichiers du dépôt
- `Remote` : dépôt distant
- `Clone` : copie locale d'un dépôt distant

Travaux Pratiques - Git, dépôt local



Travaux Pratiques - Git, cycle de vie



Travaux Pratiques - Git, commandes usuelles

- `git init` : crée un nouveau dépôt
- `git clone` : clone un dépôt distant
- `git status` : affiche l'état de la copie locale
- `git diff` : affiche les modifications de la copie locale
- `git add FICHIERS` : indexe les FICHIERS
- `git commit` : crée un commit des fichiers indexés
- `git log` : affiche la liste des commits effectués sur une branche
- `git pull` : récupère les dernières modifications d'un remote et les fusionne
- `git push` : publie les nouvelles révisions sur le remote

Travaux Pratiques - Blinky

- (Optionnel) Créer un dépôt Git sur votre compte GitHub
- (Optionnel) Changer le dépôt distant de votre projet :

```
git remote set-url origin URL-DEPOT-GITHUB
```

- **Modifier l'application for faire clignoter la LED en utilisant la classe `DigitalOut` de Mbed OS**

> Documentation Mbed OS :
> https://os.mbed.com/docs/mbed-os/latest/apis/index.html

Travaux Pratiques - Blinky (fin)

Enregistrer la nouvelle version dans GitHub :

```
git status              # On constate que *main.cpp* est modifié
git diff                # on affiche et vérifie les différences
git add main.cpp        # On ajoute les modifications à l'index
git commit -m MESSAGE # On enregistre la nouvelle version
git push origin main  # On met à jour le dépôt distant
```

Merci !

Des questions ?

t.habrant@catie.fr