



# Mise à jour d'un système Linux embarqué « Over The Air »

---

Pierre-Jean Texier

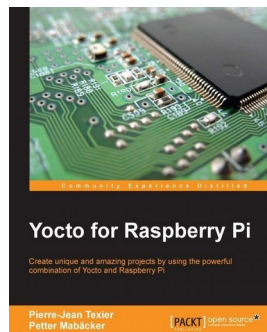
Epita / LAFON - Journées Technologiques 2021  
Mardi 20 Avril 2021



- Ingénieur Linux Embarqué - **LAFON** (groupe **Madic**)



- 30 ans
- **FOSS** enthusiast
- Contributions : U-Boot, Kernel Linux, Yocto/OE, Buildroot ...
- Co-auteur "*Yocto for Raspberry Pi*" and contributeur/auteur *GNU/Linux magazine France* et *Open silicium* (RIP)





- LAFON
- Mise à jour des systèmes embarqués
- Quelques projets Open-Source
- SWUpdate
- Cas d'usage
- Conclusion





## ▪ En quelques mots ...

- Créé en 1959
- Fait partie du groupe **Madic** depuis 2006
- Industriel français leader dans les Énergies automobiles (*stockage, distribution et gestion*) et les *Paielements sans surveillance*
- Environ **400** salariés

## ▪ Métiers





## ▪ Implantations :

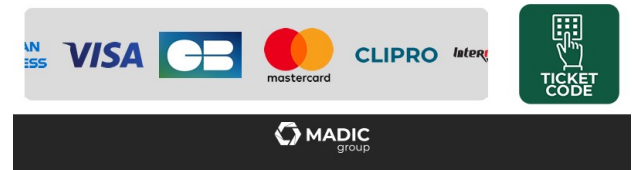
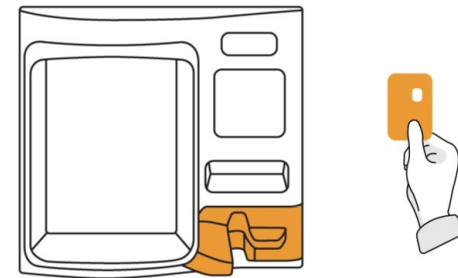
### Sites LAFON et filiales industrielles

- |   |  |
|---|--|
|  LAFON (siège)<br>Bassens FRANCE                     |  TLM & OD (payment)<br>Exeter ROYAUME-UNI |
|  LAFON (usine)<br>Périgny FRANCE                     |  LAFON ESPANA (siège)<br>Madrid ESPAGNE   |
|  LAFON (usine)<br>Faye-l'Abbesse FRANCE              |  LAFON ESPANA (usine)<br>Burgos ESPAGNE   |
|  P2M (trucks)<br>Ludres FRANCE                       |  R-LAFON (usine)<br>Leon ESPAGNE          |
|  MADIC ITALIA (payment)<br>Cardano al campo ITALIE |  LAFON PETROLYNA<br>Ain Berda ALGÉRIE   |





- **Services**
  - Orienté monétique
  - Orienté système
- **Contraint par les normes ... :**
  - PCI DSS
  - GIE Carte Bancaire
  - P2PE
  - ...
- **Contraint par la sécurité ... :**
  - Trustzone
  - Secure Boot, ...
  - ...
- **Utilisation de l'Open Source**
  - Qt (GUI)
  - Yocto/OpenEmbedded (*build system*)
  - cURL, libevent, libxml2, ... (*applicatif*)
- **Un produit : APL3.5 (nouvelle génération)**
  - [Spécifications](#)







Commerçants autrement



## SELECTIONNEZ VOTRE CARBURANT

**SP98**  
3,000 EUR/L



**SP95**  
2,000 EUR/L

**CAZOLE**  
1,000 EUR/L





## Lafon : quelques clients



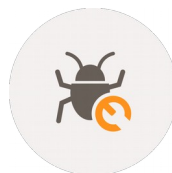


# *Mise à jour des systèmes embarqués*

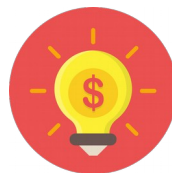
# Pourquoi ?



- Bugs logiciels



- Ajouts de fonctionnalités



- Correctifs liés à la sécurité (CVE) - <https://cve.mitre.org/>

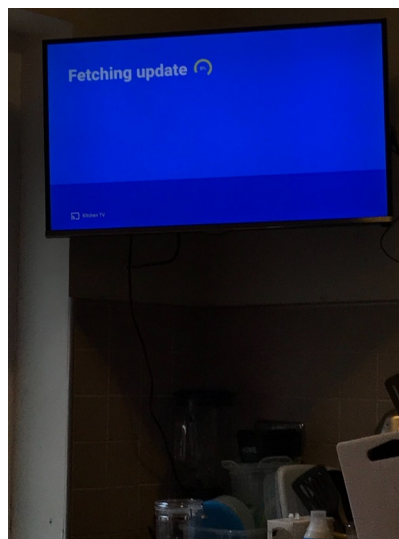
The screenshot shows the CVE database interface. At the top, there's a navigation bar with links like 'CVE List', 'CNAs', 'WGAs', 'Boards', 'About', and 'News & Blogs'. Below this is a search bar and a 'TOTAL CVE Records: 132258' indicator. The main content area displays details for 'CVE-2021-27138'. It includes a description: 'The boot loader in Das U-Boot before 2021.04.rc2 mishandles use of unit addresses in a FIT.' and a list of references with GitHub links. A 'Printer-Friendly View' link is also present.

CVE-2021-27138

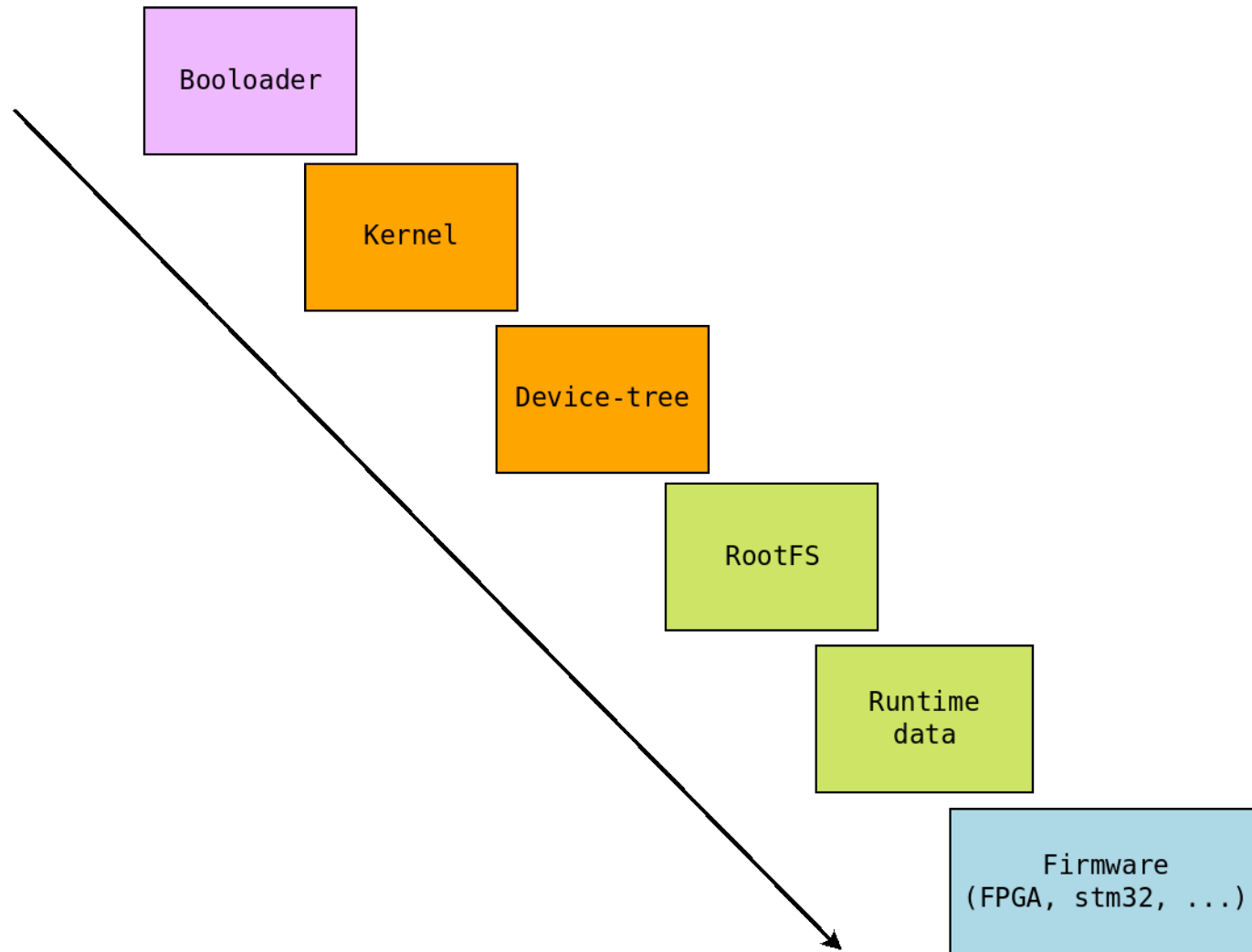
# Spécial dans l'embarqué ?



- Accessibilité : *pas d'accès physique ...*
- Disponibilité : *pas toujours facile de prendre le contrôle ...*
- Alimentation : *peu fiable dans certains cas ...*
- Connectivité : *faible bande-passante*
- Durée de vie sur site : *> 10 ans*
- ...



# Quels composants ?



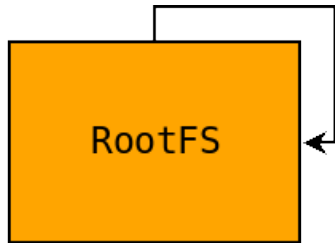
# Sur quelle base ?



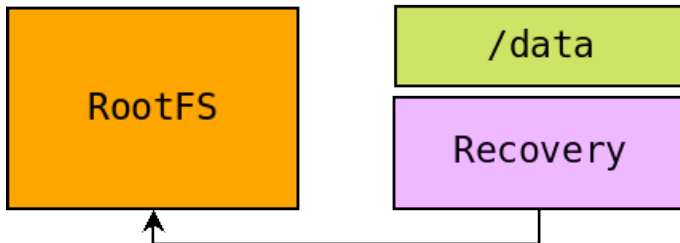
- **Fichier**
  - A éviter, difficile de garantir l'atomicité (-)
- **Gestionnaire de paquets**
  - *rpm, deb, opkg*
  - Facile (+) mais difficile à maintenir -> gestion des dépendances (-)
  - Non Atomique et non applicable pour l'embarqué (-)
- **Container (docker, ...)**
  - Concept intéressant (+)
  - Implique de gérer les applications dans un container (-)
- **Image complète**
  - Cas le plus courant dans l'embarqué
  - Facile à mettre en œuvre (+)
- **Delta (xdelta3, zchunk, casync, librsync, ...)**
  - Faible bande-passante (+)
  - Complexe (-)
  - Risques sur la corruption du système de fichiers principal (-)



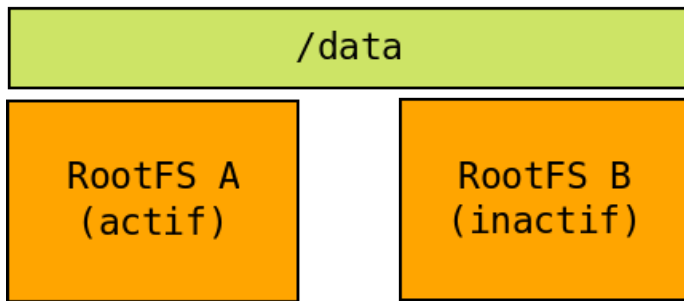
# Quels mécanismes ?



*Run Time*



*Asymétrique*



*Symétrique*

- **En fonctionnement :**

- Non Atomique (quelques exceptions)
- Downtime très court
- Qui ? : Package managers, AGL

- **« Maintenance » :** initrd/initramfs

- Robuste
- Retour en arrière impossible si erreur
- Downtime -> long
- Qui ? : Android (avant Nougat)

- **A/B ou Seamless update :**

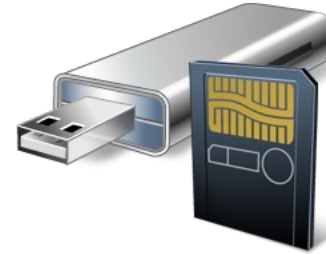
- Robuste, mais coûteux en espace de stockage
- Retour en arrière possible si erreur
- Downtime -> court
- Toujours opérationnel
- Qui ? : **Android** (depuis Nougat)

# Et comment ?



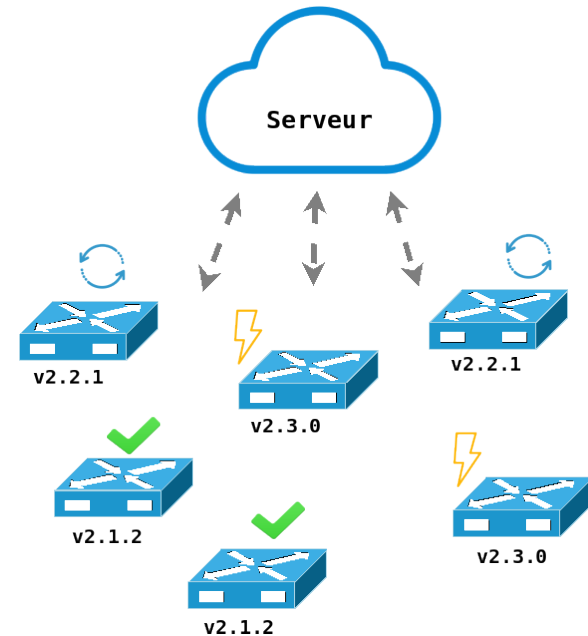
## ▪ Mises à jour sur site

- Pas de connectivité
- Accès physique :
  - Gestion par clé USB/carte SD
  - Interactif
  - Déplacement d'un technicien €€€



## ▪ Mises à jour distantes : OTA

- Pas d'accès physique :
  - HTTPS, FTP, SFTP, ...
  - Pas ou peu d'interaction (forcé, planifié)
- Serveur pour la gestion des périphériques :
  - Mise à jour programmée
  - Campagne de mise à jour
  - Inventaire des périphériques
  - Statut des périphériques
  - Version logicielle courante
  - Gestion des artefacts





- Doit être capable de mettre à jour l'ensemble des composants
  - **Bootloader = dangereux**
- Doit s'interfacer avec le Bootloader (e.g Environnement *U-Boot*)
- Doit être Robuste (coupures de courant et pertes de connexions réseau)
  - **L'opération doit être atomique = Pas d'installation partielle**
- Ne doit pas rendre le périphérique inutilisable (*Fail-safe*)
  - **Notion de Rollback**
- Doit disposer d'un espace pour les données persistentes (*Stateless FS*)
  - **Partition dédiée**
- Doit être sécurisé
  - **Empêcher une action non autorisée**
- Doit permettre une gestion Locale et Distante (OTA)
- Doit permettre d'effectuer des tests (*Sanity check*) avant validation



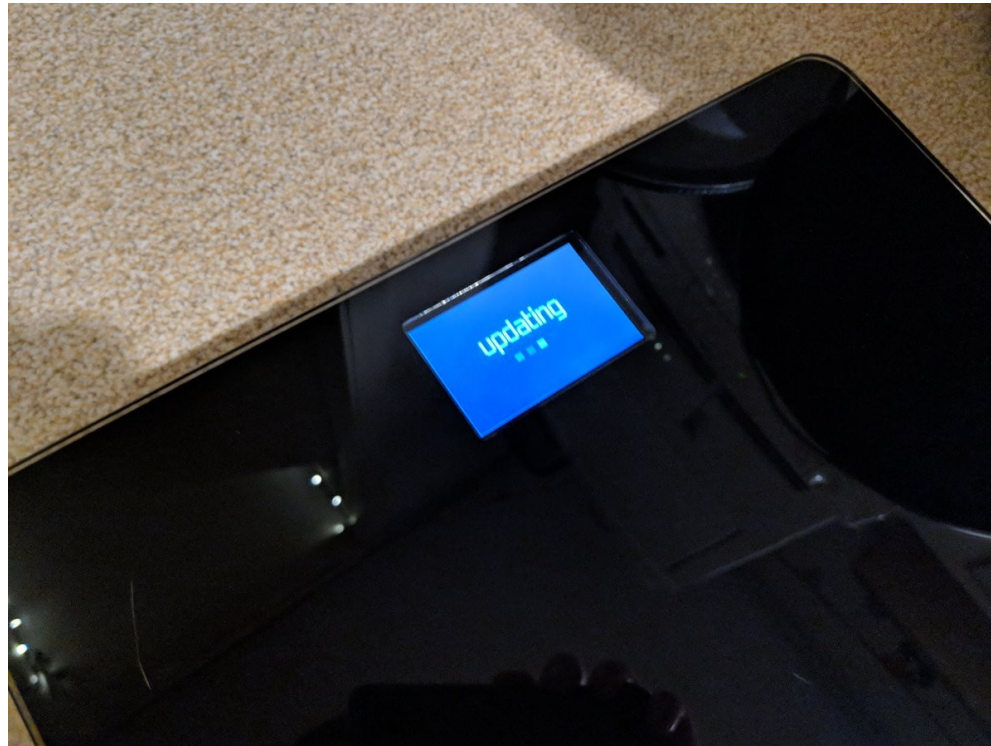
# *Downtime*



Installation de la mise à jour du système en cours...




## Exemple 2/4






Close

### Software update

 Updating...

Update


Make sure all your lights are **connected to the mains** to get them up to date. It can take up to 1 hour per light or accessory to download and lights may briefly turn off while updating.




Chair

LTW011 | 1.15.2\_r19181

Updating...






TV

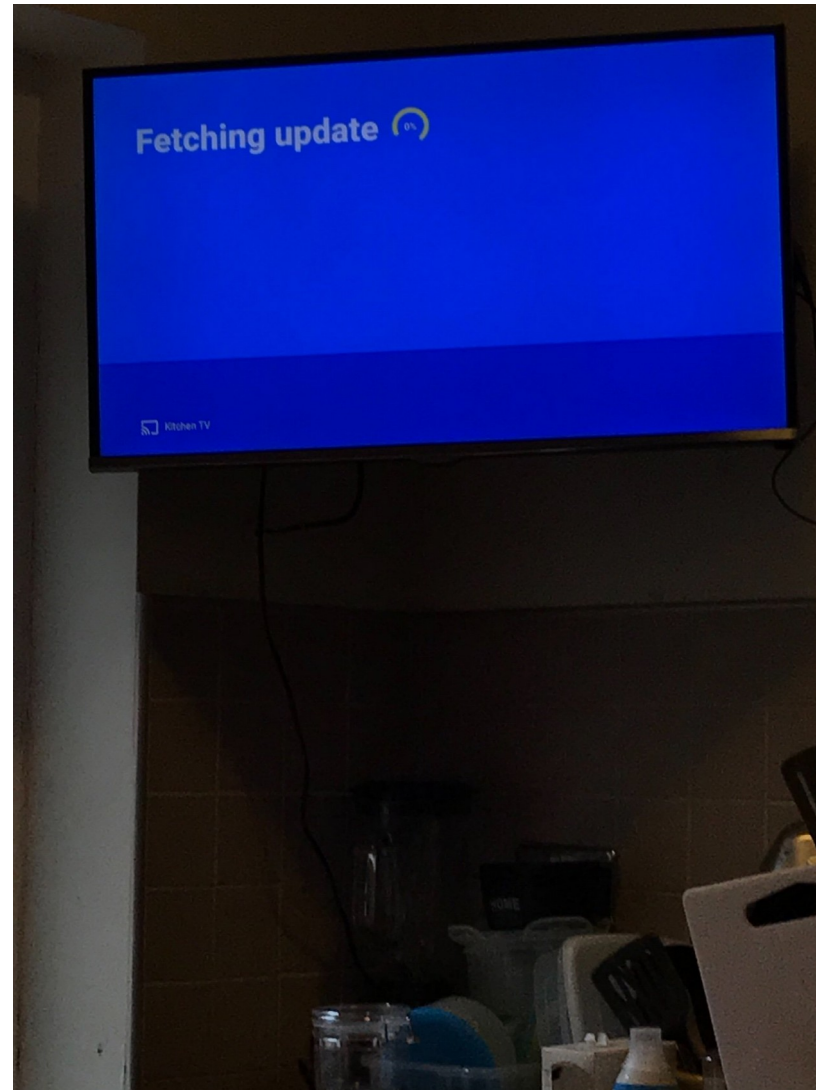
LTW011 | 1.15.2\_r19181

Updating...

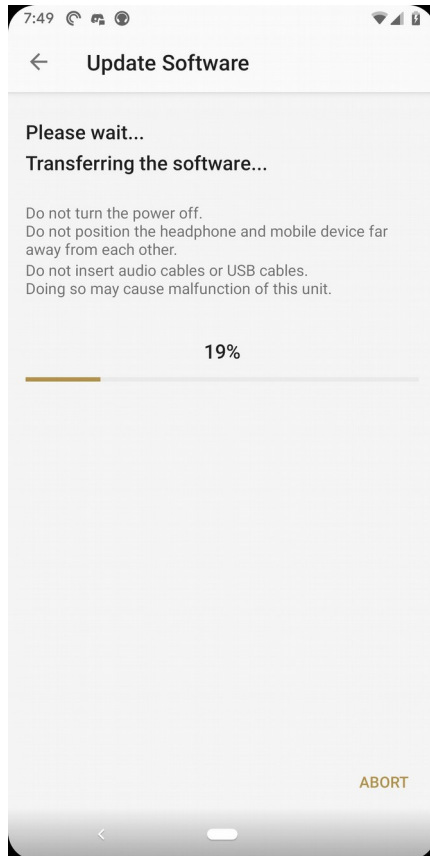




## Exemple 4/4

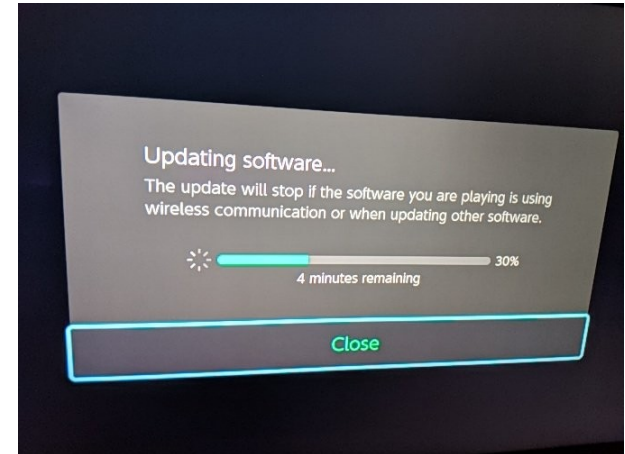


# Exemple(s) ...



## Updating your watch (4%)

Update in progress. Please keep your watch close to your smartphone and do not quit the app.

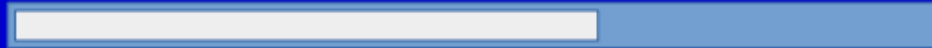




*Robuste ?*



Updating device. Do not turn off!



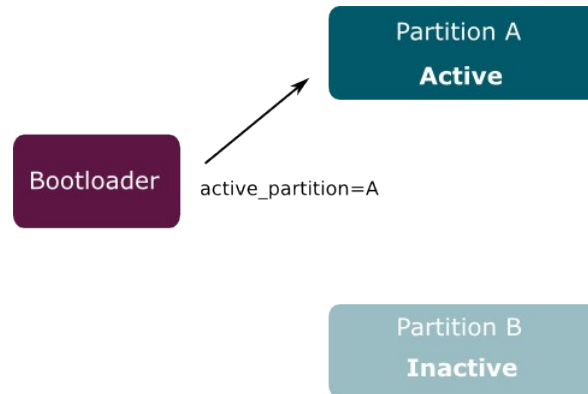
# Coupure de courant ?!



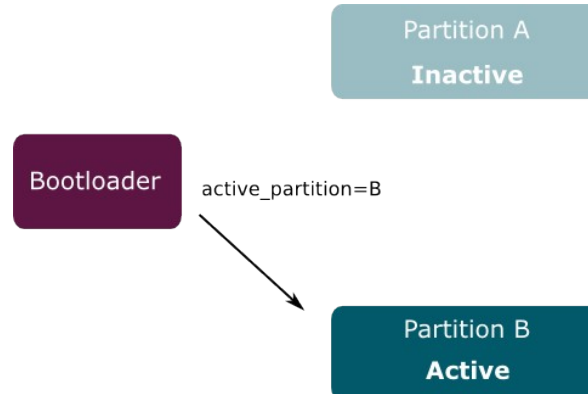
# Atomicité : schéma A/B



- Démarrage sur A



- Mise à jour de B
- **Coupure de courant !**
- Démarrage sur A



- Mise à jour de B



*Rollback ...*





*« Être capable de revenir sur une version stable (et fonctionnelle)  
lors de la détection d'un problème »*



**Ryan Negri** ✓

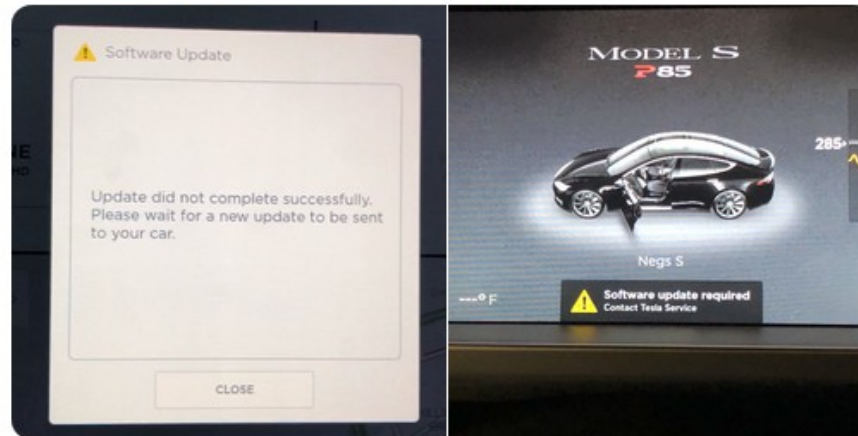
@RyanNegri

Follow



Software update issue with the Model S today. Car is immobile at the moment.

Tesla tech support and roadside assistance have been very helpful - we're hoping to avoid the tow by forcing a new update to the car.



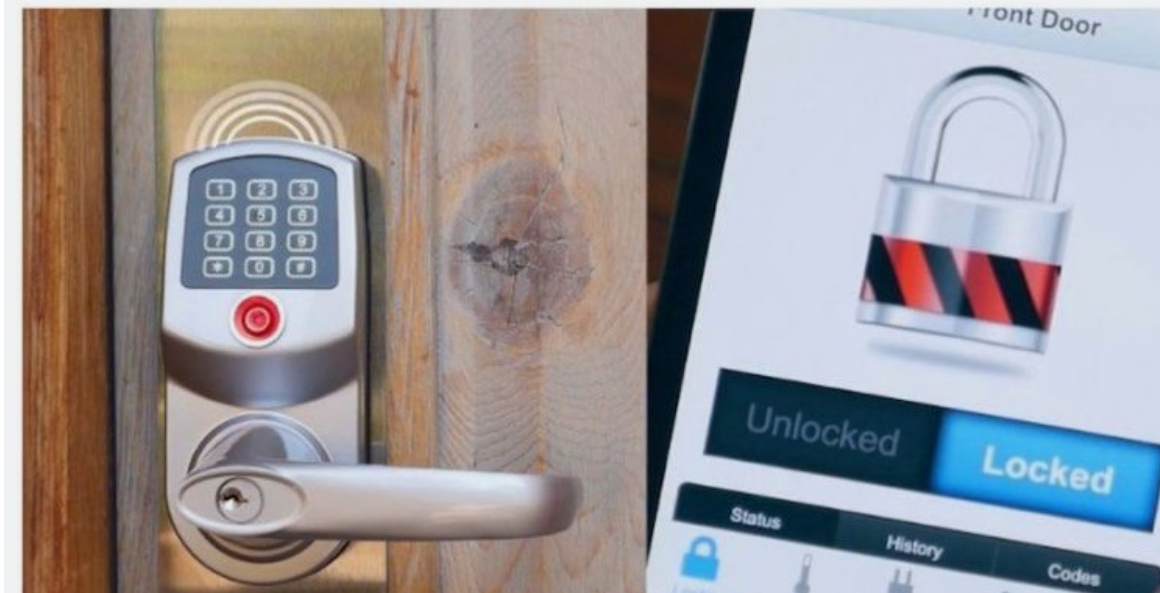
11:47 AM - 5 Sep 2018



## Update gone wrong leaves 500 smart locks inoperable

Fatal error leaves customers scrambling for fixes that can take a week or longer.

DAN GOODIN - 8/15/2017, 12:07 AM





## Plusieurs cas de figures :

- Echec d'une mise à jour
  - Détecté par l'updater
- Mise à jour réussie mais problème Noyau (Kernel Panic)
  - BUG logiciel !
- Mise à jour réussie mais problème sur l'applicatif métier
  - BUG logiciel !



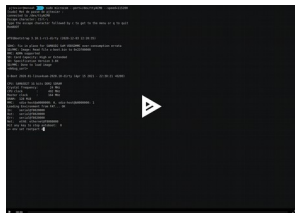
- Démarrage sur A
- Mise à jour de B
  - Mise à jour du flag **active\_partition** (flag bootloader)
- **Reboot**
- Démarrage sur B
- Détection de l'anomalie (kernel panic, sanity check, watchdog, ...)
  - Reboot
- **Rollback**
  - Mise à jour du flag **active\_partition** (flag bootloader)
- Démarrage sur A

# Rollback : fondamentaux !

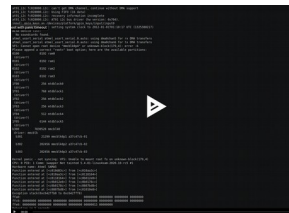


## Gestion des « Kernel panic » :

- Utilisation de **CONFIG\_PANIC\_TIMEOUT**
  - Pour définir le **timeout** avant redémarrage sur un Kernel Panic



Sans "panic timeout"



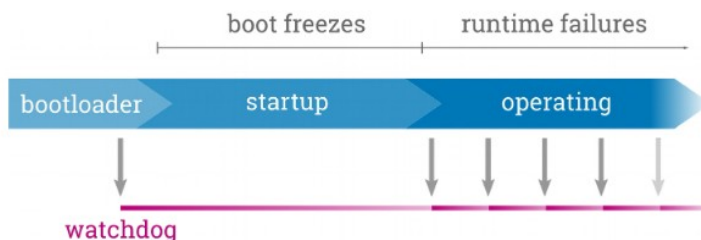
Avec "panic timeout"

## Sanity check :

- Après la mise à jour, il faut vérifier si le système est opérationnel avant de valider celle-ci :
  - Vérification de l'applicatif métier (UI, services, serveur, ...),
  - Vérification de la configuration système (base de donnée, ...),
  - ...

## Gestion des défaillances :

- Watchdog :
  - Pour détecter les « freezes »
  - Solution matérielle
    - « dev/watchdog »



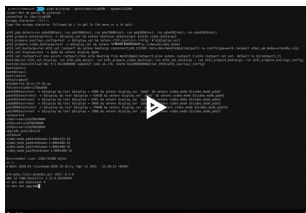


## Gestion des cycles de redémarrage répétitif : exemple avec U-Boot

- Implémentation *Bootcount Limit*

```
diff --git a/configs/sama5d27_som1_ek_mmc_defconfig
b/configs/sama5d27_som1_ek_mmc_defconfig
index 5176dbbb08..1302ebce9a 100644
--- a/configs/sama5d27_som1_ek_mmc_defconfig
+++ b/configs/sama5d27_som1_ek_mmc_defconfig
@@ -50,6 +50,9 @@ CONFIG_SYS_RELOC_GD_ENV_ADDR=y
 CONFIG_DM=y
 CONFIG_SPL_DM=y
 CONFIG_SPL_DM_SEQ_ALIAS=y
+CONFIG_BOOTCOUNT_LIMIT=y
+CONFIG_BOOTCOUNT_ENV=y
+CONFIG_BOOTCOUNT_BOOTLIMIT=1
 CONFIG_CLK=y
 CONFIG_SPL_CLK=y
 CONFIG_CLK_AT91=y
--
```

- bootcount : variable incrémentée à chaque (re)démarrage
- bootlimit : pour définir le nombre maximal de redémarrage
- altbootcmd : séquence alternative si bootcount > bootlimit
- upgrade\_available : pour activer/désactiver la gestion de bootcount







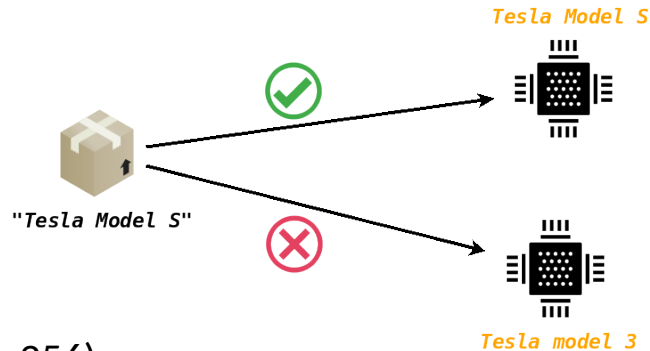
# *Sécurité*

# Rollback : fondamentaux !



## De façon générale :

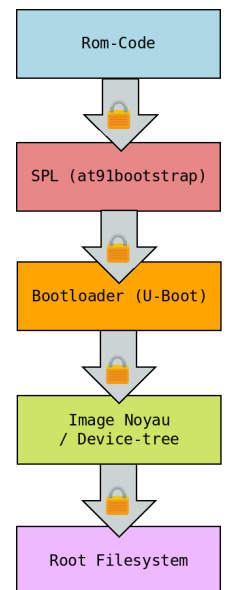
- Gestion des compatibilités (Même matériel != Même produit)



- Contrôle d'intégrité (sha256)

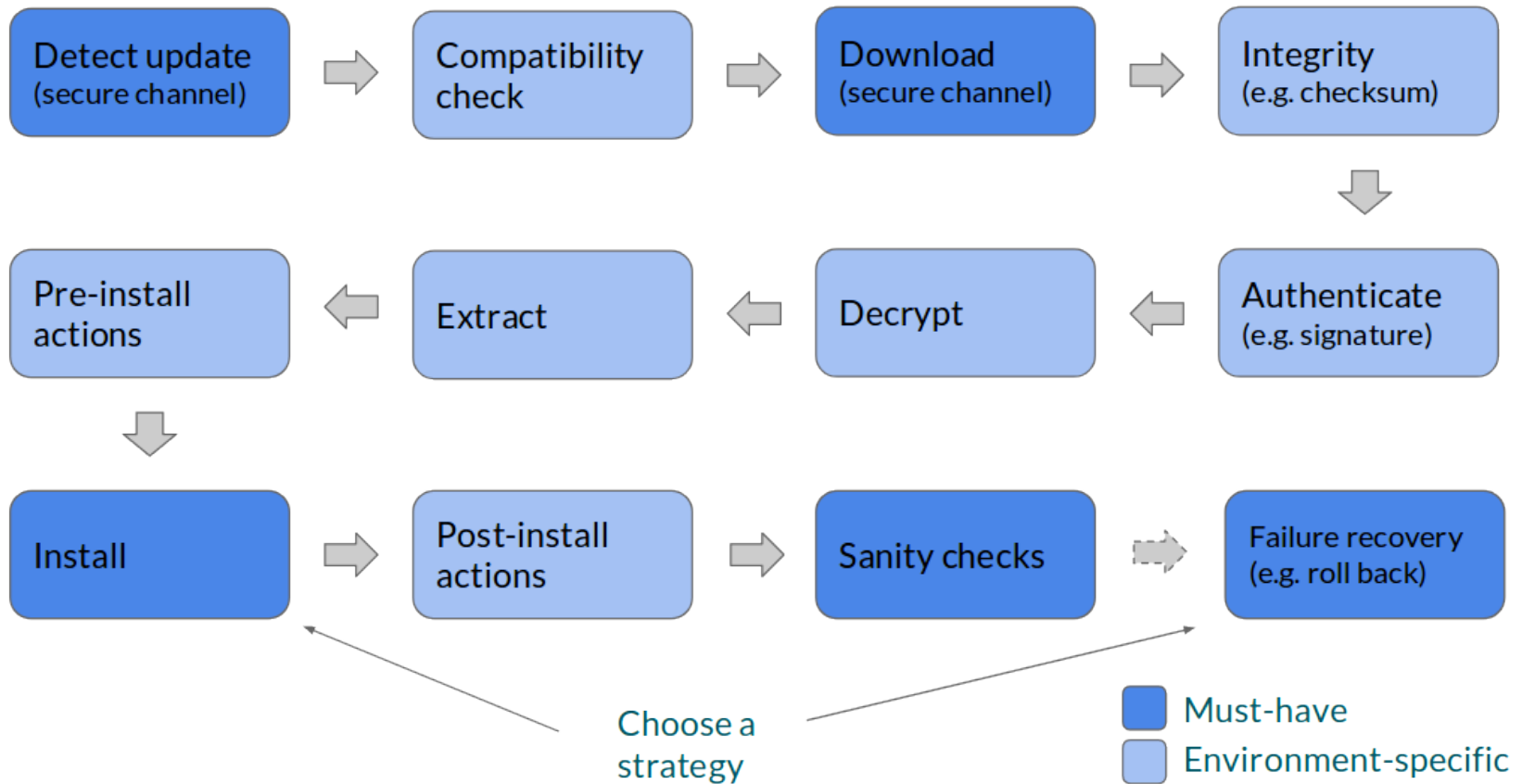
## Sécurité :

- Communication sécurisée (OTA)
- Cryptographie (signature/vérification/chiffrement)
- Secure Boot
  - garantir l'intégrité de l'ensemble des éléments de la chaîne de démarrage pouvant être mis à jour.





*Pour résumer ...*





## *Quelques projets « Open Source »*



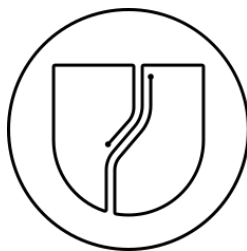
mender



updatehub



RAUC



SWUpdate

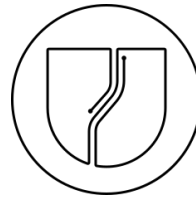


Balena OS



# *SWUpdate*





« *software update agent for embedded system* »

- <https://github.com/sbabic/swupdate>
- Maintenu par **Stefano Babic** de la société **DENX**
- Bien documenté
- Principalement écrit in **C**, avec un « binding » **LUA**
- Client léger (environ **400 Ko**)
- Format simple : **CPIO** archive (**.swu**)
- HTTP(S), FTP, SSH -> (libcurl) pour l'aspect réseau
- SD/eMMC, UBI, Raw NAND, NOR/SPI NOR





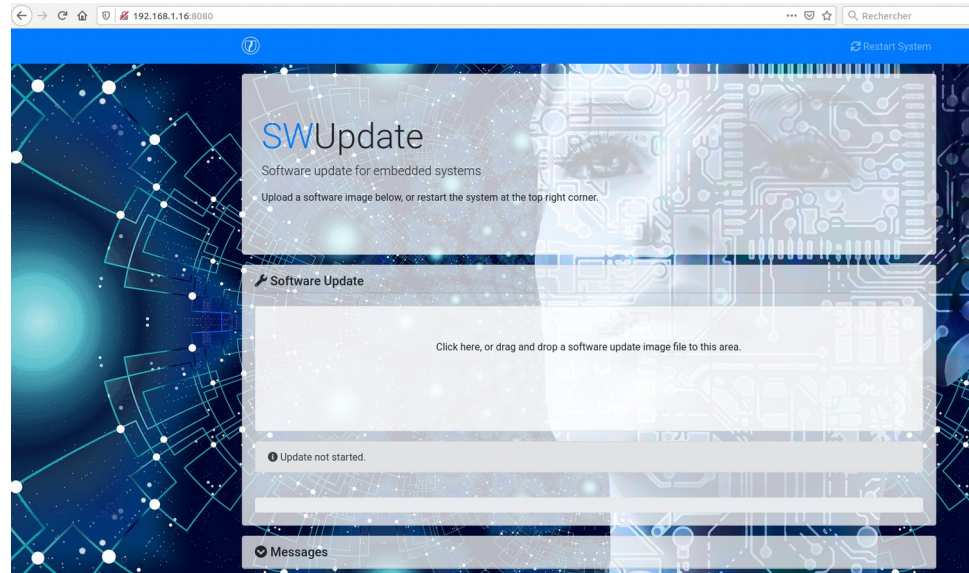
- Rollback non intégré (quelques implémentations disponibles)
- **IPC** : Unix Domain Socket pour interagir avec le « core » (progression, inventaire, ...)
- Mise à jour locale (USB et serveur web) ou distante (**Hawkbitt**)
- Interface graphique en Lua pour le mode recovery (**RescueGUI**)
- Gestion scripts pre/post install
- Gestion microcontrôleur en UART (**ucfw**), gestion maître/esclave (**SWU forwarder**) et beaucoup d'autres « *handlers* »
- Build system : Buildroot et Yocto/OE
- De nombreux exemples (*raspberrypi*, *sama5d27*, *wandboard*, *beaglobone*)
- <https://swupdate.org/>



*Démo !*



- Mise à jour via le **webservice** interne



- Mise à jour via **Hawkbitt** avec inventaire « custom »

```
...
identify : (
  { name = "manufacturer";    value = "EPITA"; },
  { name = "version";        value = "1.0.0"; },
  { name = "hardware";       value = "Microchip"; },
  { name = "model";          value = "SAMA5d27-SOM1-EK1"; }
);
...
```



# *Conclusion*



- Un mécanisme de mise à jour est obligatoire (**le penser au début du projet**)
- Toujours déployer un logiciel bien testé (CI/CD) :
  - Gitlab,
  - TBOT,
  - Labgrid,
  - Jenkins,
  - ...
- Faire l'état de l'art en début de projet
- Oublier les solutions « Maison »
  - **Moins d'efforts, moins de coûts ...**
  - Utiliser un « *framework* » open source permet de bénéficier de la communauté (code review, features, ...)





 **LAFON** *recrute !*





# Q&A



*@texierp*



*@pjtexier*