

# Debian dans votre (future) voiture ?

Arnaud Ferraris  
arnaud.ferraris@collabora.com

Capitole du Libre 2023

# Qui suis-je ?

- Libriste des années 2000
- Professionnel de Linux embarqué depuis 10 ans
- Ingénieur logiciel @ Collabora
  - Intégration et maintenance de distributions pour l'embarqué
- Développeur Debian
  - Fondateur et mainteneur de Mobian





**Au commencement était...**  
**le MHz**

# Il n'y a pas si longtemps...

- Linux embarqué il y a 10 ans
  - Faible puissance
    - Cœur unique, généralement < 1 GHz
    - Parfois des microcontrôleurs (sans MMU !)
  - Stockage et RAM limités (quelques dizaines de Mo)
- Systèmes (très) contraints
  - Nécessité d'optimiser au maximum !



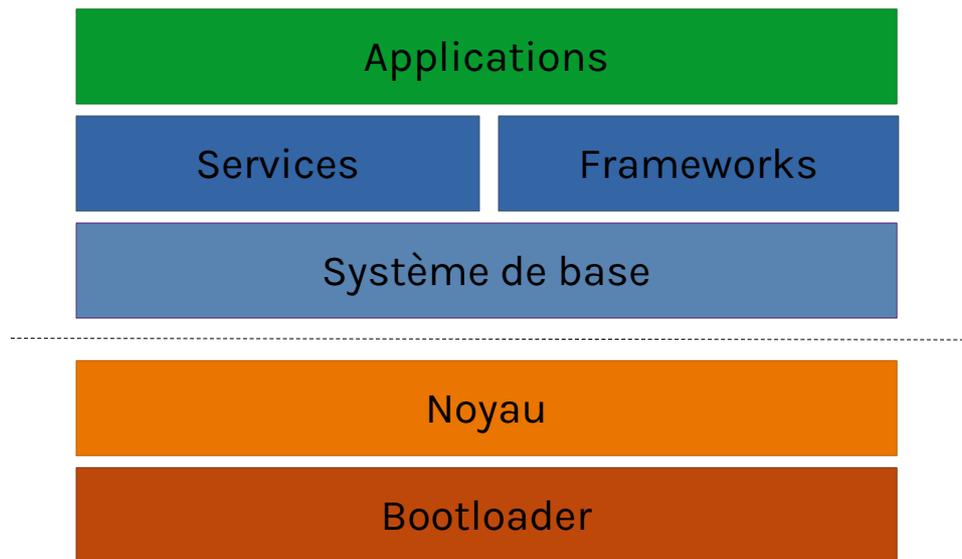
# Les systèmes embarqués modernes

- Généralisation des SoC ARMv8
  - Architecture 64 bits
  - Multi-cœurs (2-8, parfois plus !) > 1 GHz
- RAM et stockage à profusion
  - Plusieurs Go de mémoire
  - Stockage exprimé en dizaines/centaines de Go
- Quel niveau d'optimisation est nécessaire ?



# Anatomie d'un système embarqué

- Logiciels bas-niveau
  - Bootloader (u-boot, Barebox...)
  - Noyau Linux, souvent modifié
- Système d'exploitation
  - Souvent basé sur Yocto
- Application(s) métier





COLLABORA

# À propos de Yocto

# Yocto : un projet de référence..

- Projet soutenu par la Linux Foundation
- Standard de fait pour les systèmes Linux embarqués
  - Communauté importante
  - Nombreux exemples
  - Relativement bien documenté
- Nombreux BSPs disponibles comme “layers” Yocto
- Permet (à peu près) tout



# ...mais loin d'être parfait !

- Compilation demande beaucoup de ressources
- Manque de fiabilité du cache
- Nombreux “layers” → complique le suivi des modifications
- Mise à jour problématique : <https://col.la/eb23yocto>
- Très (trop) complexe pour un non-initié
- Permet (à peu près) tout



# Debian pour l'embarqué ?

# Les distributions “traditionnelles”

- Généralement disponibles pour x86 (64, voire 32 bits)
- De plus en plus de ports ARM 64 bits
- Fonctionnement par paquets binaires
  - Flatpak peut être une alternative intéressante (Fedora Silverblue)
- Ensemble cohérent et stable
  - Fréquentes mises à jour de sécurité



# Le cas “Debian”

- Projet communautaire
- “The Universal Operating System”
  - Distribution de référence, nombreux dérivés (Ubuntu, Mint, Raspbian...)
- Nombreuses architectures matérielles supportées
  - 9 architectures officielles
    - x86 (32/64 bits), ARM v5/7/8, MIPS 64, PowerPC 64, RISC-V 64, S390x
  - Ports communautaires utilisant l’infrastructure du projet (ex : m68k)



# Debian : personnalisation

- Moins complexe qu'on ne le croit
  - Format de paquet bien documenté
  - Nombreux outils pour faciliter le packaging
- Possibilité de ne modifier qu'un nombre limité de paquets
  - Le reste peut être téléchargé directement des miroirs officiels
- Création d'images personnalisées
  - Plusieurs outils disponibles, notamment debos



# Exemple : image basique avec debos

```
architecture: amd64
```

```
actions:
```

```
- action: debootstrap
  suite: bookworm
  components:
    - main
  mirror: http://deb.debian.org/debian
  variant: minbase
```

```
- action: apt
  packages:
    - adduser
    - gnome-core
    - grub-pc
    - linux-image-amd64
    - sudo
```

```
- action: run
  chroot: true
  command: adduser --gecos user --disabled-password user; adduser user sudo; echo "user:user" | chpasswd
```

```
- action: image-partition
  imagename: debos-image.img
  imagesize: 4GB
  partitiontype: msdos
  mountpoints:
    - mountpoint: /
      partition: root
  partitions:
    - name: root
      fs: ext4
      start: 1MB
      end: 100%
      flags: [ boot ]
```

```
- action: filesystem-deploy
```

```
- action: run
  chroot: true
  command: grub-install /dev/vda && update-grub2
```



# Debian : mises à jour

- Distribution stable
  - Pas de changements radicaux
  - Mises à jour de sécurité
  - Corrections de bugs majeurs
- Debian testing en tant que “rolling release”
  - Tester les mises à jour de façon incrémentale
  - Évite un gros “rebase” à chaque nouvelle version stable





# Étude de cas : Mobian

# Mobian en bref



- Distribution dérivée de Debian
- Cible les téléphones et tablettes
- Plusieurs architectures supportées
  - x86-64
  - ARMv7 et v8
  - RISC-V 64



# Mobian, concrètement

- Très petit nombre de paquets modifiés ou ajoutés
  - ~35 paquets sources
- Place centrale de GitLab CI
  - Pipelines pour construction et publication des paquets
  - Runners auto-hébergés pour ARM64 et RISC-V 64
  - Noyaux mis à jour automatiquement sur la dernière stable
- Migration et mises à jour manuels



# Mobian : organisation et maintenance

- Projet communautaire
- Très petite équipe
  - 5 contributeurs réguliers (~0,5 équivalent temps plein)
  - Majorité du travail directement dans Debian
- Approche adaptée aux projets à faibles ressources
  - Maintenance facilement automatisable
  - Permet de se concentrer sur les fonctionnalités métier





# Étude de cas : Apertis

# Apertis en bref

- Distribution dérivée de Debian
- Cible les systèmes embarqués industriels
  - Initialement à destination de l'automobile
  - Élargi depuis, plus générique
- Plusieurs architectures supportées
  - x86-64
  - ARMv7 et v8

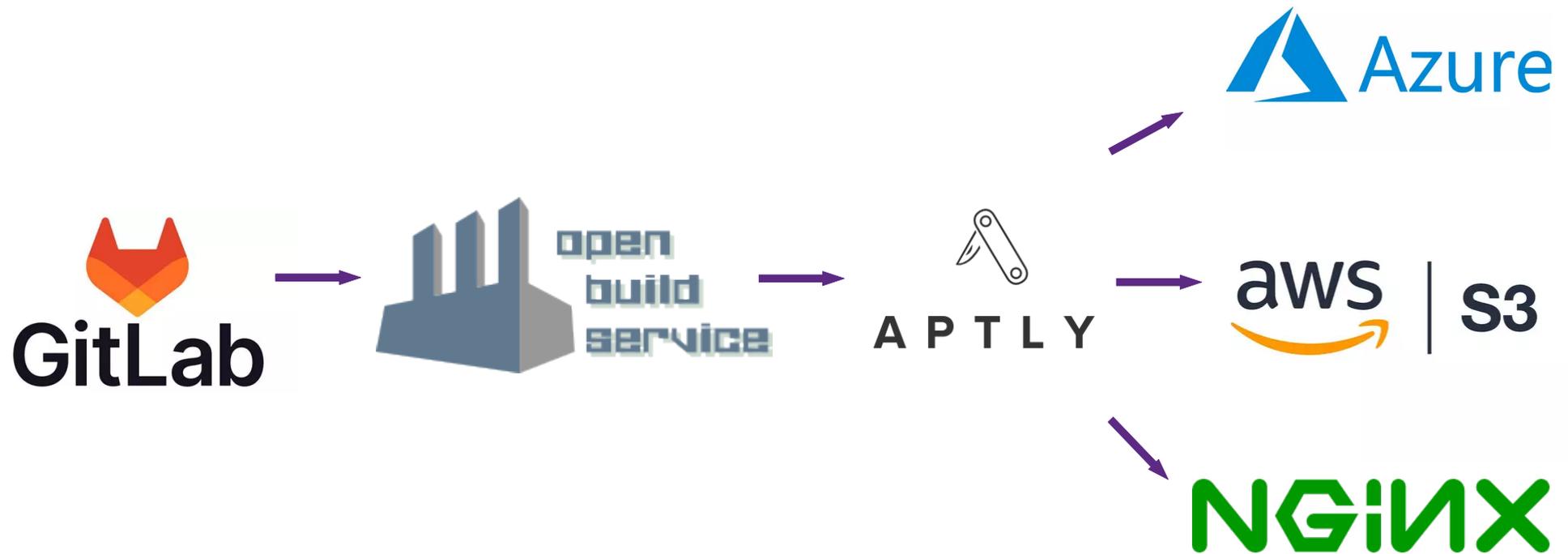


# Apertis, concrètement

- Fork complet de Debian
  - Basé sur Debian stable
  - Une version majeure par an, mise à jour mineure chaque trimestre
- Archive organisée par rôles
  - *target, development, sdk...*
  - Pas de GPLv3 dans *target*
- Infrastructure dédiée



# Apertis, concrètement



# Apertis : cibles supportées

- Quelques cibles de référence
  - Au moins une par architecture supportée + QEMU
    - i.MX6 (SabreLite)
    - Renesas R-Car M3 / i.MX8M / Raspberry Pi 4
    - UP Squared 6000 (Intel Atom)
  - Bootloader et noyaux maintenus pour ces cibles
- Objectif : fournir des exemples facilement adaptables



# Apertis : organisation et maintenance

- Projet professionnel
- Équipe plus vaste
  - Environ 10-15 ETP (évolutions incluses)
- Documentation abondante
  - Choix de conception et orientations futures
  - Détails des aspects majeurs du système
  - Guides pas à pas



# Apertis : fonctionnalités avancées

- Schéma de partitionnement A/B
- Mises à jours atomiques et déploiement distant
  - OSTree + Hawkbit
- Framework applicatif
  - Recommandations pour le développement/déploiement d'applications
  - Paquets Debian ou Flatpak
  - Utilise les technologies standard de la communauté (Wayland, PipeWire, D-Bus...)





COLLABORA

# Conclusion

# Debian dans votre (future) voiture ?

- Plusieurs projets en développement et en production
  - Automobile
  - Spatial
  - Objets connectés “puissants”
- Plusieurs approches possibles
  - Choix selon les ressources disponibles et les objectifs du projet
- Alternative à considérer



## Liens utiles

- Debian : <https://debian.org/>
  - Developer's Reference : <https://debian.org/doc/manuals/developers-reference/>
  - Guide d'empaquetage : <https://debian.org/doc/manuals/debmake-doc/>
- Mobian : <https://mobian.org>
  - Wiki : <https://wiki.debian.org/Mobian>
- Apertis : <https://apertis.org>
  - Apertis @ Bosch Measurement Tools : <https://col.la/apertisbmt>



Des questions ?





**Merci !**



COLLABORA

**Open First**



On recrute !  
[col.la/careers](https://col.la/careers)



COLLABORA

Open First