UNIVERSITÉ
CÔTE D'AZUR

Introduction aux Systèmes et Logiciels Embarqués

Présentation: Stéphane Lavirotte
Auteurs: ... et al*

(*) Cours réalisé grâce aux documents de :
Bootlin, Stéphane Lavirotte

Mail: Stephane.Lavirotte@univ-cotedazur.fr
Web: http://stephane.lavirotte.com/
Université Côte d'Azur

UNIVERSITÉ
CÔTE D'AZUR

Des Systèmes Embarqués

- ✓ Système embarqué:
  - Système électronique et informatique
  - Autonome
  - Souvent temps réel
  - Spécialisé dans une tâche
  - Ressources limitées (taille, capacité, consommation)
- ✓ Critère de comparaison des Systèmes Embarqués
  - Classe de processeur (microprocesseur / microcontrôleur)
  - Gestion de mémoire (avec ou sans MMU)
  - Système d'exploitation (avec ou sans OS)
  - Type de système d'exploitation (normal ou temps réel)
- ✓ Depuis le début du cours
  - Microprocesseur avec MMU avec OS « Normal »

UNIVERSITÉ CÔTE D'AZUR

Exemple d'Objet Connecté

- ✓ Nabaztag – 2005 (1er OC grand public)
- ✓ Nabaztag:tag – 2006 (ARM 7 sans MMU)
- ✓ Karotz – 2011
  - ARM 9 avec MMU, Linux, 256Mio Flash 64Mio RAM
  - se connecte à Internet par WiFi 802.11b/g
  - communique avec son utilisateur en émettant des messages vocaux (enceinte), lumineux (led) ou en remuant les oreilles (moteurs)
  - diffuse des informations du type météo, bourse, qualité de l'air, circulation routière du périphérique de Paris, arrivée d'e-mail, etc.
  - Lit des tag RFID passif et déclenche des services à la lecture de lire associés à ces tags
  - Port USB pour lire des fichiers musicaux
  - Webcam et microphone
- ✓ Et si on regardait à l'intérieur…

UNIVERSITÉ
CÔTE D'AZUR

Un Lapin sans sa Peau !

UNIVERSITÉ CÔTE D'AZUR

Autres Exemples: un NAS et un Routeur WiFi

NAS: FreeNAS

- ✓ Unité de stockage réseau
- ✓ Réseau filaire10/100/1000
- ✓ Multiple fonctionnalités: DHCP, Firewall, applications, ...
- ✓ Interface Web de Configuration

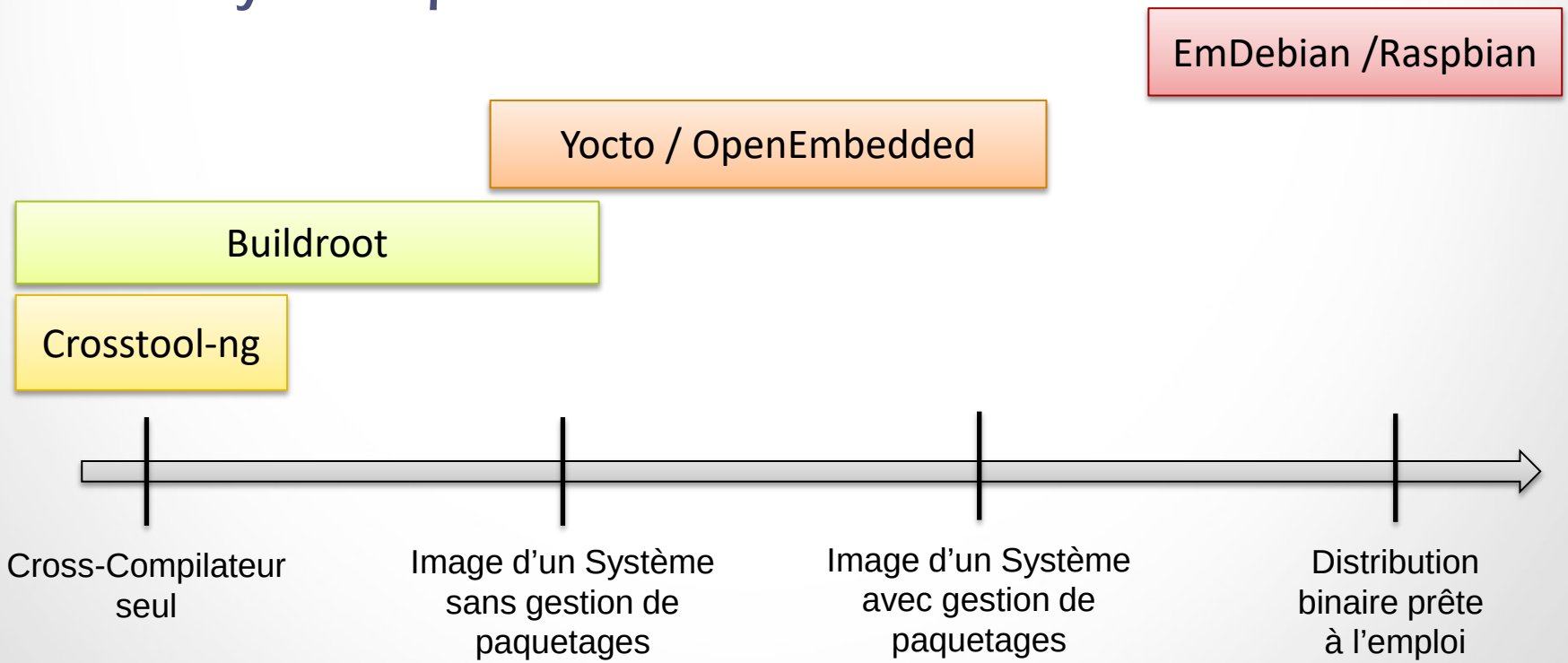
Switch/Routeur WiFi

- ✓ Réseau filaire 10/100/1000
- ✓ IPv4, IPv6
- ✓ Serveur DHCP
- ✓ Firewall
- ✓ Cache DNS
- ✓ Interface Web de Configuration

UNIVERSITÉ CÔTE D'AZUR

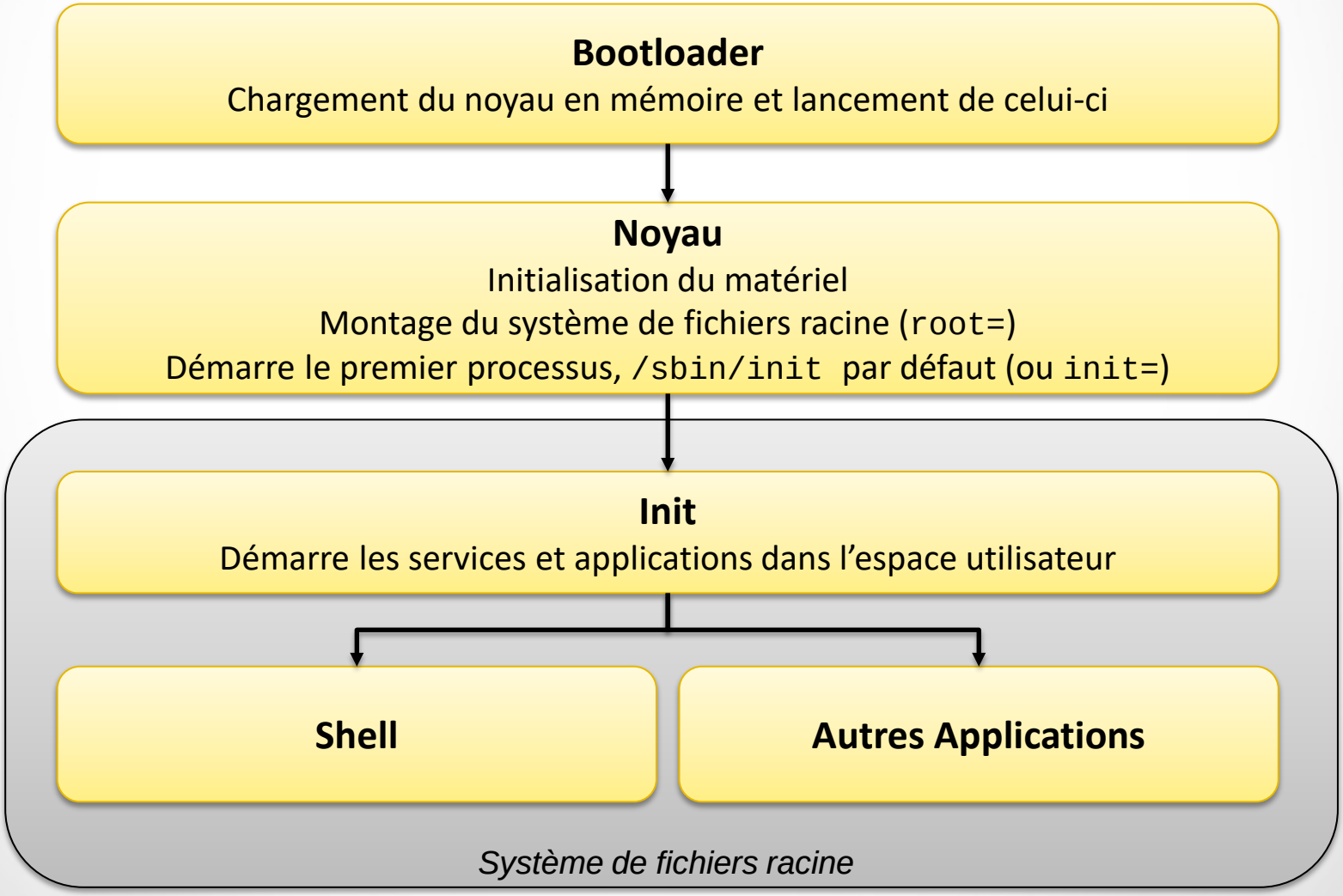
Positionnement des Types de Systèmes de Constructions

- ✓ Systèmes à produire
  - Une fois pour toute
  - Une déclinaison de systèmes proches
  - Un système qui soit modulaire et évolutif



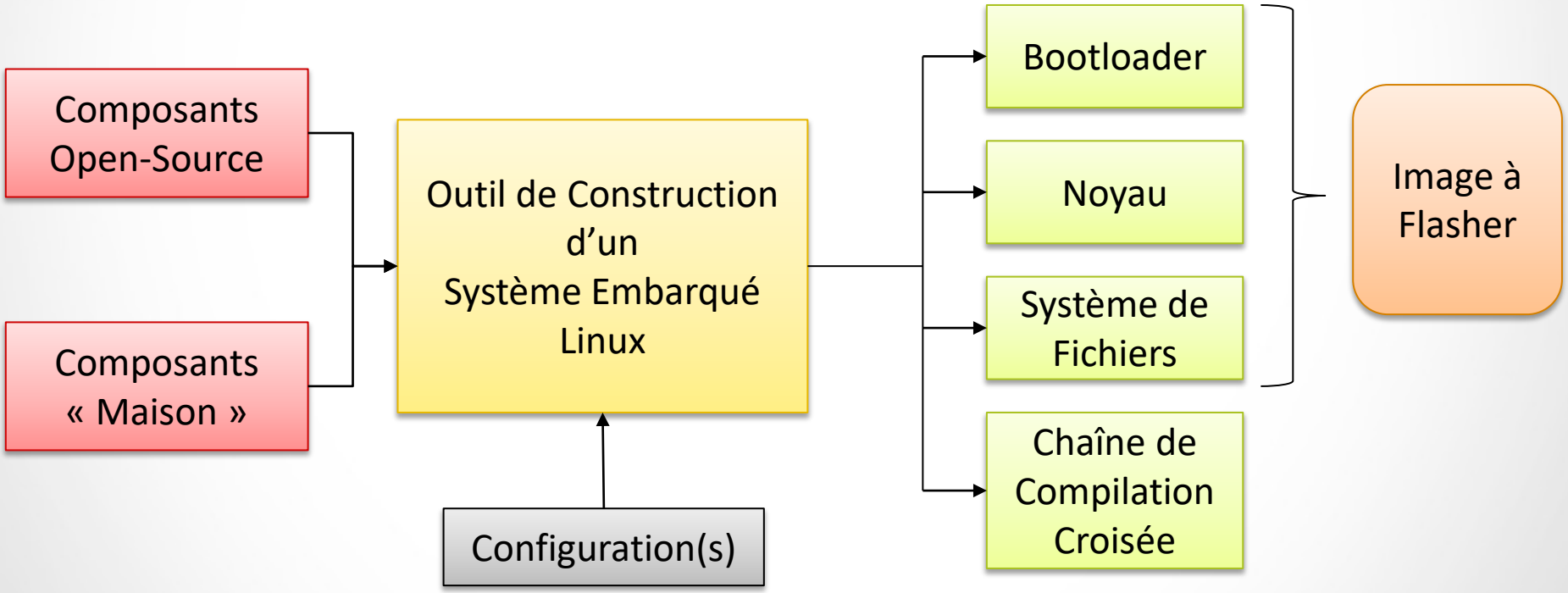
UNIVERSITÉ CÔTE D'AZUR

Résumé des Episodes Précédents



UNIVERSITÉ CÔTE D'AZUR

Principe de Production d'un Système Embarqué avec OS



UNIVERSITÉ CÔTE D'AZUR

Comparaison de la Création de Systèmes Embarqués

	Avantages	Inconvénients
Création manuelle complète (« From scratch »)	Flexibilité maximale Très bonne expérience pour l'apprentissage Compréhension complète du système	Gestion des dépendances Compatibilité des versions Difficile à reproduire et automatiser
Distributions binaires (Debian, Ubuntu, Fedora)	Facile à créer et installer	Difficile à configurer Difficile à optimiser Difficile à reconstruire Gros systèmes Pas disponible pour tout
Construire son propre système (Buildroot, Yocto, …)	Très grande flexibilité Construction à partir des sources Reproductible Utilise la cross-compilation	Pas aussi simple qu'une distribution binaire Temps de compilation du système complet

UNIVERSITÉ
CÔTE D'AZUR

Distributions Prêtes à l'Emploi

La grande distribution:
des « trucs » prêts à l'emploi
mais pas/peu modifiables

UNIVERSITÉ CÔTE D'AZUR

Distributions Binaires pour l'Embarqué

- ✓ Distribution d'applications par paquetages
  - Système identique aux distributions GNU/Linux desktop
  - Peut être utilisé par tout matériel compatible au niveau binaire
  - Intéressant au démarrage d'un projet
    - Pas besoin de créer un système de fichiers « from scratch »
    - Beaucoup d'applications et outils prêt à l'emploi
  - Facilité de mise à jour, ajout ou suppression d'applications
    - Système de paquetages binaires
  - Pas de nécessité de tout recompiler
  - Gère les dépendances
- ✓ Inconvénients:
  - Système produit pas optimum en taille et performance
  - Pas de configurabilité maximum (fonctionnalité des applications)

+ = Raspberry Pi OS

- ✓ Une distribution Debian spécifique pour Raspberry
  - Une image à copier sur la carte SD (win32diskimager, balena)
  - Des paquetages pour l'ajout ou retrait de fonctionnalités
- ✓ Avantages
  - Très nombreux paquetages disponibles
  - Fournit les outils et pilotes spécifiques à la plateforme
  - Facilité d'emploi
    - On se retrouve comme sur un desktop
- ✓ Inconvénients
  - Par vraiment embarqué (distribution minimale ~1Go)
  - Prévu pour une utilisation Desktop et pas Embarqué

UNIVERSITÉ CÔTE D'AZUR

Conclusion Partielle

- ✓ Démarrage très rapide
  - Facilité d'installation d'un système déjà très complet
  - Beaucoup de fonctionnalités déjà disponibles
  - Sympathique pour les amateurs ou un prototypage rapide
- ✓ Mais…
  - Difficile à configurer finement
    - suppression de fonctionnalités non souhaitées, …
  - Difficile reproductibilité et automatisation des modifications apportées
  - Approche pas professionnelle pour un produit fini
- ✓ Donc système de construction d'une image nécessaire

UNIVERSITÉ
CÔTE D'AZUR

Construction d'une image / distribution personnalisée

... et personnalisable

UNIVERSITÉ CÔTE D'AZUR

Outils pour la Construction de Systèmes Embarqués

- ✓ Buildroot
  - Crée une image d'un système complet sans paquetage, simple à configurer et à modifier
- ✓ Yocto/OpenEmbedded
  - Crée une distribution complète avec paquetages binaires, puissant, mais plus complexe à maîtriser
- ✓ PTXdist
  - Reproducible Embedded Linux Systems
- ✓ LTIB
  - Linux Target Image Builder
- ✓ OpenBricks
  - Création facile et configuration facile de systèmes embarqués
- ✓ OpenWRT
  - Création de systèmes embarqués pour les routers et gateways
- ✓ Et bien d'autres encore...

UNIVERSITÉ CÔTE D'AZUR

Buildroot

- ✓ Buildroot
  - http://buildroot.org/
  - Supporte de nombreuses architectures
  - Télécharge automatiquement les sources et applique les patches
  - Peut compiler la plupart des applications dont vous avez besoin
    - BusyBox, bzip2, Cairo, dbus, Dillo, DirectFB, Dropbear, lighthttpd, Python, Qtopia4, sqlite, thttpd, tinyX, xorg...
  - Produit une image du système de fichiers
  - Très facile à mettre en œuvre (identique au noyau Linux)
    - `make menuconfig`
    - `make`

UNIVERSITÉ CÔTE D'AZUR

Builroot Utilisation

- ✓ Configurations
  - De Buildroot lui-même:
    - `make menuconfig`
  - Du Noyau:
    - `make linux-menuconfig`
  - De la librairie C (uClibC)
    - `make uclibc-menuconfig`
  - De Busybox:
    - `make busybox-menuconfig`

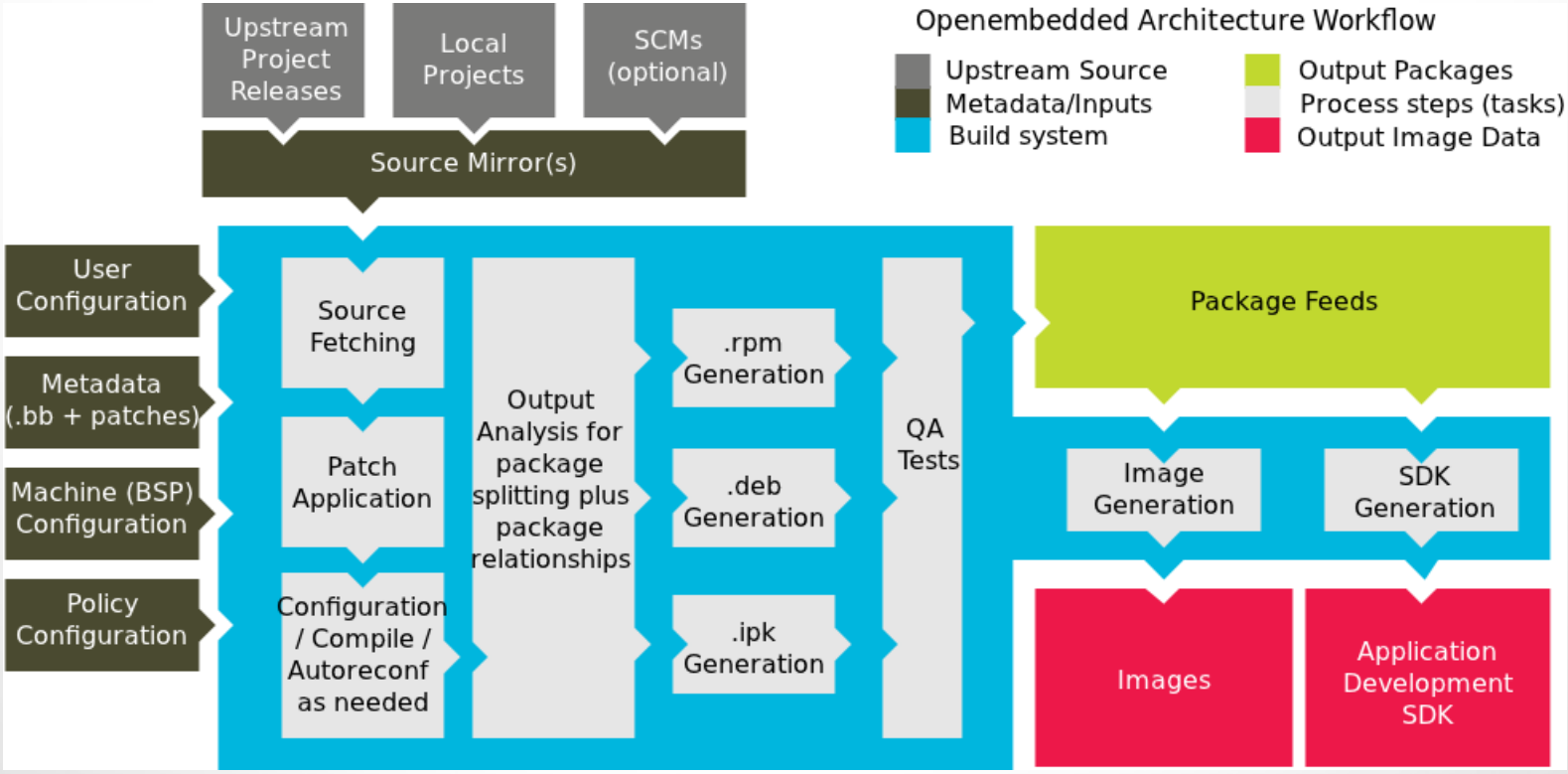
OpenEmbedded

- ✓ Build framework for Embedded Linux
  - http://www.openembedded.org/
  - Supporte de nombreuses architectures
  - Facile à adapter
  - Fonctionne sur de nombreuses distributions Linux
  - Compile de nombreuses applications (plus de 1000 paquetages)
    - Incluant GTK+, Xwindows, Mono, Java, ...
  - Permet l'utilisation de glibc ou de uClibc
- ✓ Adopté par Yocto project comme système de construction

UNIVERSITÉ CÔTE D'AZUR

yocto PROJECT

- ✓ Fournit des modèles, outils et méthodes pour la construction de systèmes embarqués
- ✓ Soutenu par la Linux Fondation



UNIVERSITÉ CÔTE D'AZUR

Des Outils Commerciaux

- ✓ Attention: commercial ne signifie pas propriétaire
- ✓ Contournement de la licence GPL
  - Distribution des sources à leur utilisateurs
  - Et la plupart du temps à la communauté
  - La GPL n'oblige pas à partager les sources avec un 1/3 quelconque
- ✓ La plupart du temps les outils graphiques sont quant à eux distribuées sous licence propriétaires
  - Mal signalé sur les sites Internet
  - Besoin de s'enregistrer pour obtenir un kit et les infos sont dedans

UNIVERSITÉ CÔTE D'AZUR

Avantages des Outils Commerciaux

- ✓ Avantages techniques
  - Bon niveau de test
  - Bon support des versions de noyau
  - Inclusion de patchs très rapide
- ✓ Chaîne de développement complète
  - Noyau, utilitaires, pour une large liste de plates-formes supportées
- ✓ Intégration d'utilitaire pour
  - génération automatique noyau, initrd, filesystem
- ✓ Outils graphiques
- ✓ Outils de développement pour diverses plates-formes
  - Linux, Windows, Solaris, ...
- ✓ Services Support
  - Intéressant si pas de support dans votre société
  - Support à long terme même pour des versions considérées comme obsolètes.

UNIVERSITÉ CÔTE D'AZUR

montavista™

- ✓ http://www.mvista.com/
- ✓ Le Leader du marché actuellement
  - Organise des conférences (MontaVista Vision)
  - Emploie les principaux développeurs actifs du noyau en particulier sur plate-forme arm
  - Les développements noyau sont éventuellement partagés avec la communauté
  - La plupart des drivers sont intégrés dans la branches principale du noyau
  - Les outils de développement graphiques sont propriétaires

WIND RIVER

- ✓ http://windriver.com/
- ✓ Wind River Linux
  - Beaucoup d'expérience sur l'embarqué et le temps réel avec VxWorks
  - Intégration, test et support sur Linux aussi rigoureux que sur VxWorks
  - Support de versions récentes des sources du noyau
    - Inclus aussi les patches temps réels
    - Ainsi que les patches temps réel dur (RTLinux)

UNIVERSITÉ CÔTE D'AZUR

Autres entités

- ✓ De nombreuses entités fournissent des produits intéressants:
  - Denx Software Engineering
    - http://denx.de/
  - TimeSys
    - http://timesys.com/
  - Sysgo
    - http://sysgo.com/
  - Lynx Software Technology
    - http://lynuxworks.com/
  - Koan Software
    - http://koansoftware.com/

UNIVERSITÉ CÔTE D'AZUR

Conclusion pour l'Entreprise

- ✓ Outils Commerciaux
  - Intéressant si pas de support en interne dans la société
  - Nécessite un budget
  - Permet de se focaliser sur le travail réel à réaliser
  - Possibilité d'avoir des contrats de sous-traitance avec le fournisseur d'environnement
- ✓ Outils de la Communauté (en général libres)
  - Le meilleur choix en cas de budget restreint
  - Le meilleur choix pour se faire une expérience personnelle sur les systèmes embarqués et apprendre les bonnes pratiques
- ✓ Dans notre cadre académique
  - Nous utiliserons bien sûr ce qui est le plus formateur (2ème solution)

UNIVERSITÉ CÔTE D'AZUR

Où en est-on à mi-parcours ?

- ✓ Assez de connaissances pour basculer maintenant sur du matériel
- ✓ Fin de la première partie du cours sur les systèmes embarqués avec OS « normal »
- ✓ Suite du cours:
  - Microcontrôleur sans OS
  - Microcontrôleur avec OS Temps Réel
  - Introduction des OS Temps Réel
  - Architectures mixtes (microcontrôleur et microprocesseur)