

TACT factory mobile agency

Systemes Embarqués



Fournisseur **souple** et **réactif**
d'applications mobiles innovantes avec
une **méthodologie** projet **rigoureuse**.

Présentation

Mickael Gaillard (Architecte logiciel)

mickael.gaillard@tactfactory.com

Yoan Pintas (Lead Developer)

yoan.pintas@tactfactory.com

Erwan LeHuitouze (Lead Developer)

erwan.lehuitouze@tactfactory.com



Présentez-vous

Votre :

Nom Prénom

Age

Spécialité (langage, administratif)

Orientation professionnelle

Objectif du cours

Planning session 1

1-Définition

-

3-Vue dynamique

-

2-Vue statique

-

Présentation

Embedded system : Système Embarqué ou enfoui
système électronique et informatique autonome, souvent temps réel,
spécialisé dans une tâche bien précise.



Historique :

> Apollo Guidance Computer (AGC) en 1961, développé par Charles Stark Draper du Massachusetts Institute of Technology (MIT).
« considéré comme l'élément le moins fiable du projet Apollo »

Domaines d'application

Pourquoi : doublement d'object connecté tous les 2/3ans

- Télécoms : Smartphone, Internet, routeur (box ADSL)
- Multimedia : Télévision, Console de jeux,
- Electroménager : Lave vaisselle, Lave linge, Four micrond.
- Aéronautique : Avions, Radar, Satellite
- Automobile : GPS, ABS, ESP
- Transport ferroviaire
- Construction électrique (Systèmes de surveillance, Gestion électrique, contrôle d'accès)
- Cartes à puce
- Robotique : UAV, AUV, Humanoid, Autonomous cars



Principe de base !

- Encombrement,
- Consommation Énergétique (autonomie),
- Réactivité,
- Criticité (robustesse, sécurité, fiabilité) :
 - Qualité du code
 - Nombreux tests
 - Classification de la couverture de tests

Category	HCL Practices	Key Challenges
Healthcare & Transportaion Infrastructure	<ul style="list-style-type: none">• Medical Avionics• Automotive	<ul style="list-style-type: none">• Reliability• Real time
Internet & Telecom Infrastructure	<ul style="list-style-type: none">• Data comm.• Telecom	<ul style="list-style-type: none">• Performance• Throughput
Consumer	<ul style="list-style-type: none">• Entertainment• PDAs	<ul style="list-style-type: none">• Foot print• Algorithmic
Industrial	<ul style="list-style-type: none">• Energy• Automation• Marine	<ul style="list-style-type: none">• Safety• Efficiency

Architecture Système généralités

Stack :

- Espace utilisateur (Applications)
- Api OS (Posix ou Windows)
- Noyaux (kernel)
- Boot-loader
- Bios (ou EFI)
- Hardware

Variation : Dédier vs Préemptive (Temps partagé) vs Temps réels

Architecture OS

Dédier

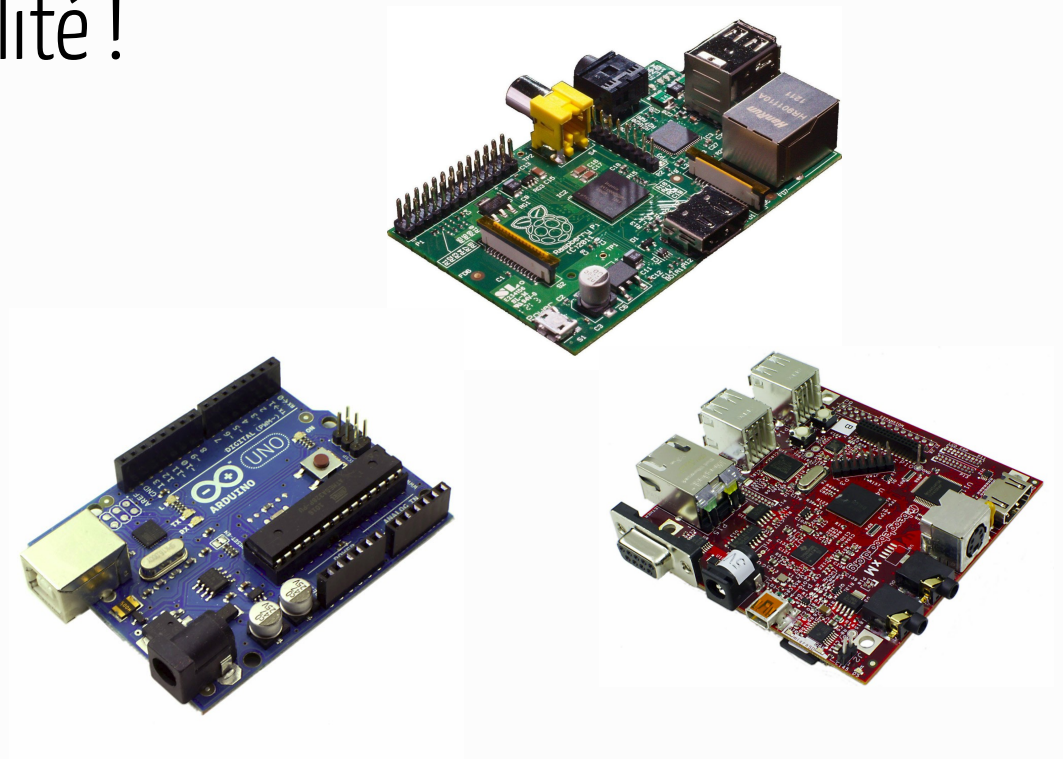
Préemptive (Temps partagé)

Temps réels

Plate-forme de développement

Objectif : Tester sur un système proche de la réalité !
(sans pour autant investir dans le dédié...)

- Picduino (PIC)
- Arduino (MicroController)
- RaspberryPi (ARM)
- BeagleBone (ARM)



Émulation cible

Objectif : simuler sans le hardware final !!

Solution : Virtualisation !

- Vserver (jail & chroot)
- Emulation global (les poupées ruses)
- Paravirtualisation (xen)
- Virtualisation (xen, kvm, Microsoft hyperV, Vmware esx)

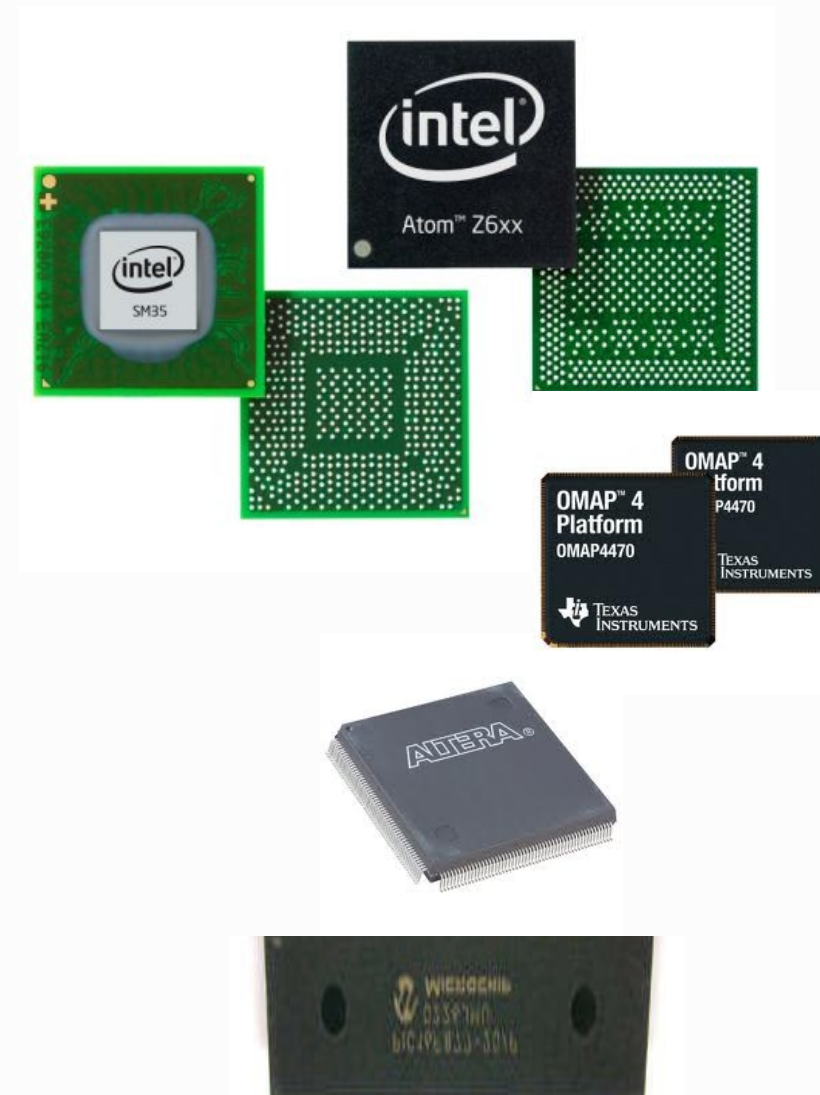
Architecture CPU

CPU :

- intel x86 (Amd, Via...) et AMD64
- ARM
- Mips
- Sparc

DSP & FPGA

Micro-contrôleur (Pic...)



Chaîne de compilation <générique>

Source == Compilateur ==> binaire exécutable

|

Linkeur

|

Binaire exécutable global
(*.exe & *.dll)

Chaîne de compilation <x86>

Source == Compilateur (x86) ==> binaire exécutable (x86)

|

Linkeur

|

Binaire exécutable global (x86)
(*.exe & *.dll)

Traditionnel chaîne de compilation !!
Nombreux outils d'assistance... & Rapide !

Chaîne de compilation <arm>

Source == Compilateur (arm) ==> binaire exécutable (arm)

|

Linkeur

|

Binaire exécutable global (arm)
(*.exe & *.dll)

C'est long tous les outils ne sont compatibles.
????

Chaîne de compilation croisé

Source == Compilateur (x86) ==> binaire exécutable (arm)

|

Linkeur

|

Binaire exécutable global (arm)
(*.exe & *.dll)

Avantage du x86 pour ARM !

TP 0 – Compile Raspberry



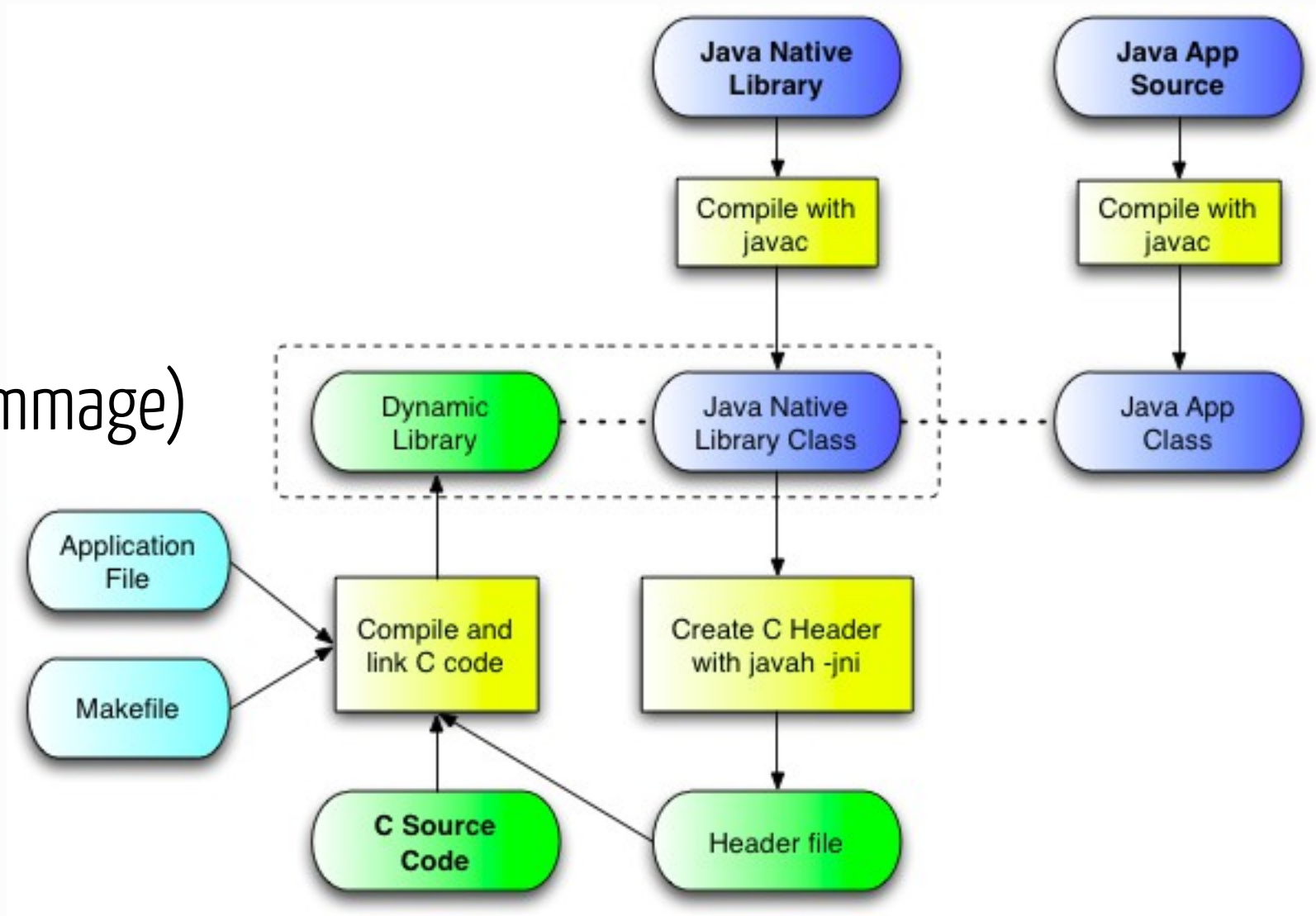
Débugger sur x86 puis compiler pour ARM

A vous de jouer !

Cas de code manager

Etude de Cas : Android

- Source C/C++
- Android/make
- WrapperJava (basé sur le nommage)



TP 1 – Compile Android



Compiler pour ARM, ARMHF, x86, Mips un code natif pour une app android

A vous de jouer !

Gestion de la sécurité

Etude de Cas : Android

- Source C/C++
- Android/make
- WrapperJava (basé sur le nommage)

Questions ?

Creative Commons 2013 TACTfactory.

Change log

- Mickael Gaillard 2013 : Initial document