

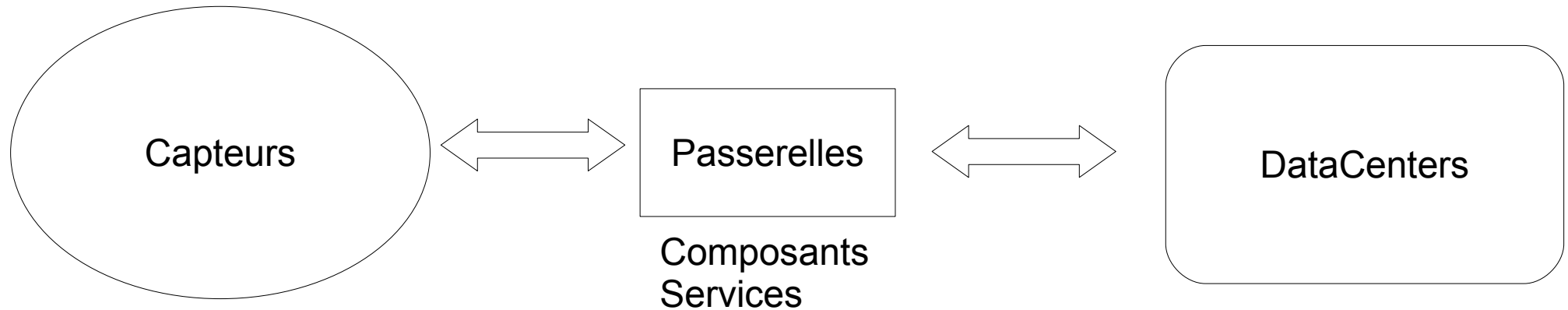


AMAZONES

Ambient **M**iddleware **A**rchitectures:
Services **O**riented, **N**etworked,
Efficient and **S**ecured

stephane.frenot@inria.fr
<http://amazones.gforge.inria.fr>

Architectures ciblées

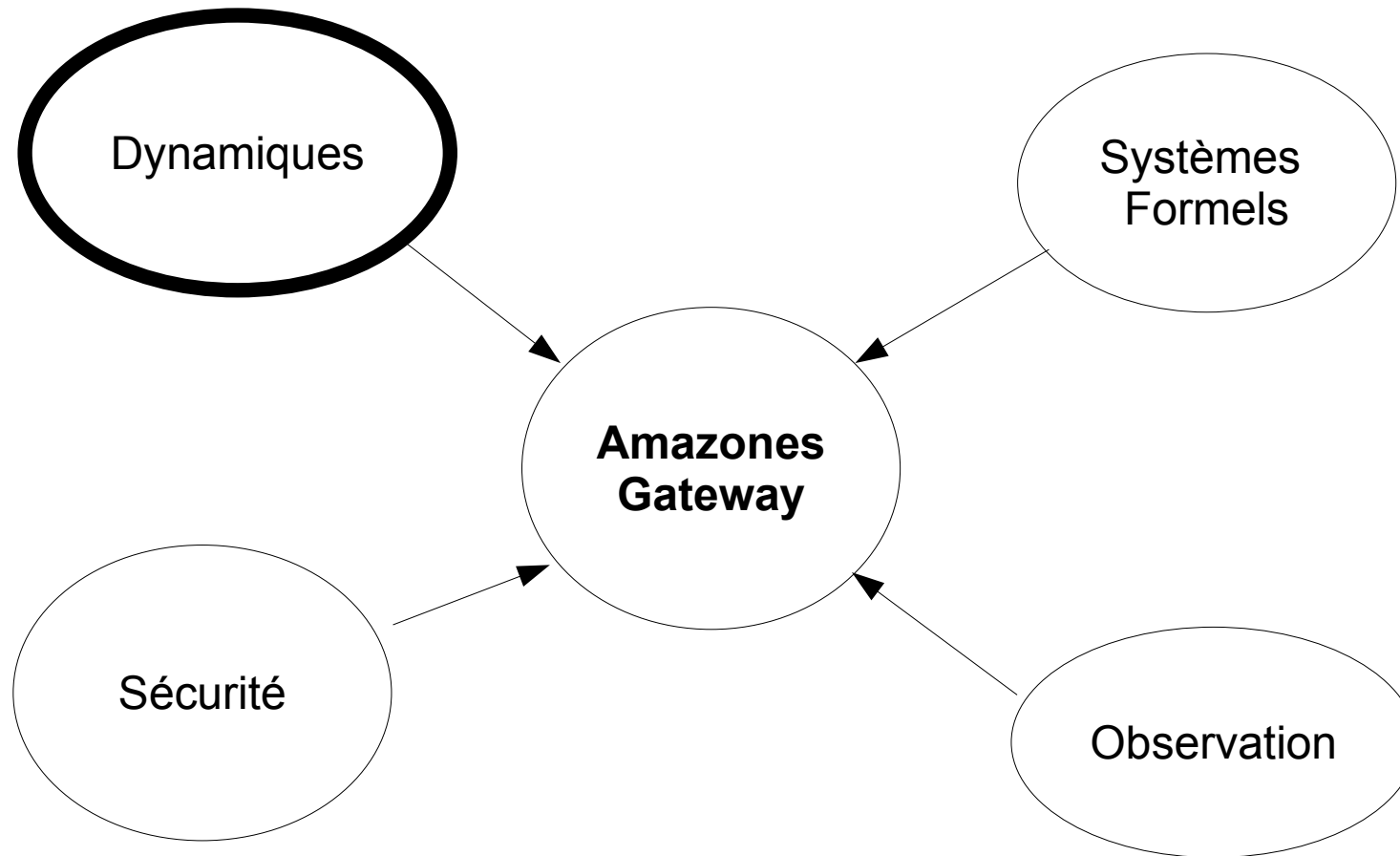


Fixes :

- Bâtiment intelligent
- Transport

Mobiles :

- Informatique ambiante
- Informatique spontanée





Par exemple OSGi



- Système dynamique ?
- Cible des systèmes de petite taille ?



Par exemple OSGi



- Système dynamique ?
 - Chargement de bundles

- Cible des systèmes de petite taille ?
 - Compatible J2ME



Par exemple OSGi



- **Systeme dynamique ?**
 - **Chargement de bundles**
 - ==> Le bundle contient des ressources qui ne sont jamais utilisées
- **Cible des systemes de petite taille ?**
 - **Compatible J2ME**
 - ==> Il existe quelques differences entre J2SE et J2ME



Par exemple OSGi



- **Systeme dynamique ?**
 - **Chargement de bundles**
 - ==> Le bundle contient des ressources qui ne sont jamais utilisées
 - + Ne pas télécharger les classes inutilisées
- **Cible des systemes de petite taille ?**
 - **Compatible J2ME**
 - ==> Il existe quelques differences entre J2SE et J2ME
 - + Utiliser un J2SE standard



Fonctionnement



> start <http://remotelocation/test.jar>

- 1) Récupération tcp de test.jar et sauvegarde sur le cache local
- 2) Extraction de l'archive sur le cache local
- 3) Fabrication d'un BundleLoader, chargeur de ressources dédié
- 4) Chargement du manifest
- 5) Contrôle de la cohérence du bundle avec le run-time
- 6) Instanciation reflective de l'activator
- 7) Invocation de la méthode start
- 8) Chargement des classes nécessaires



Fonctionnement Sans chargement

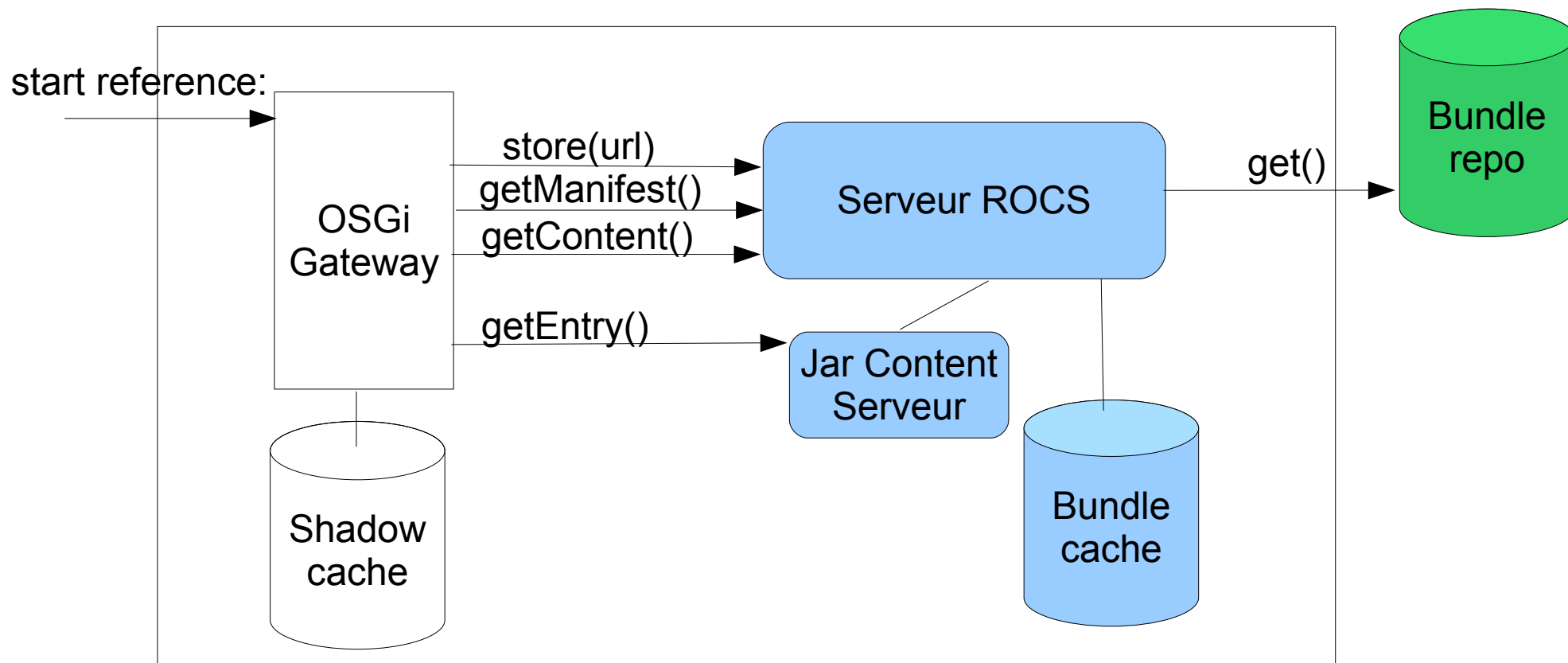


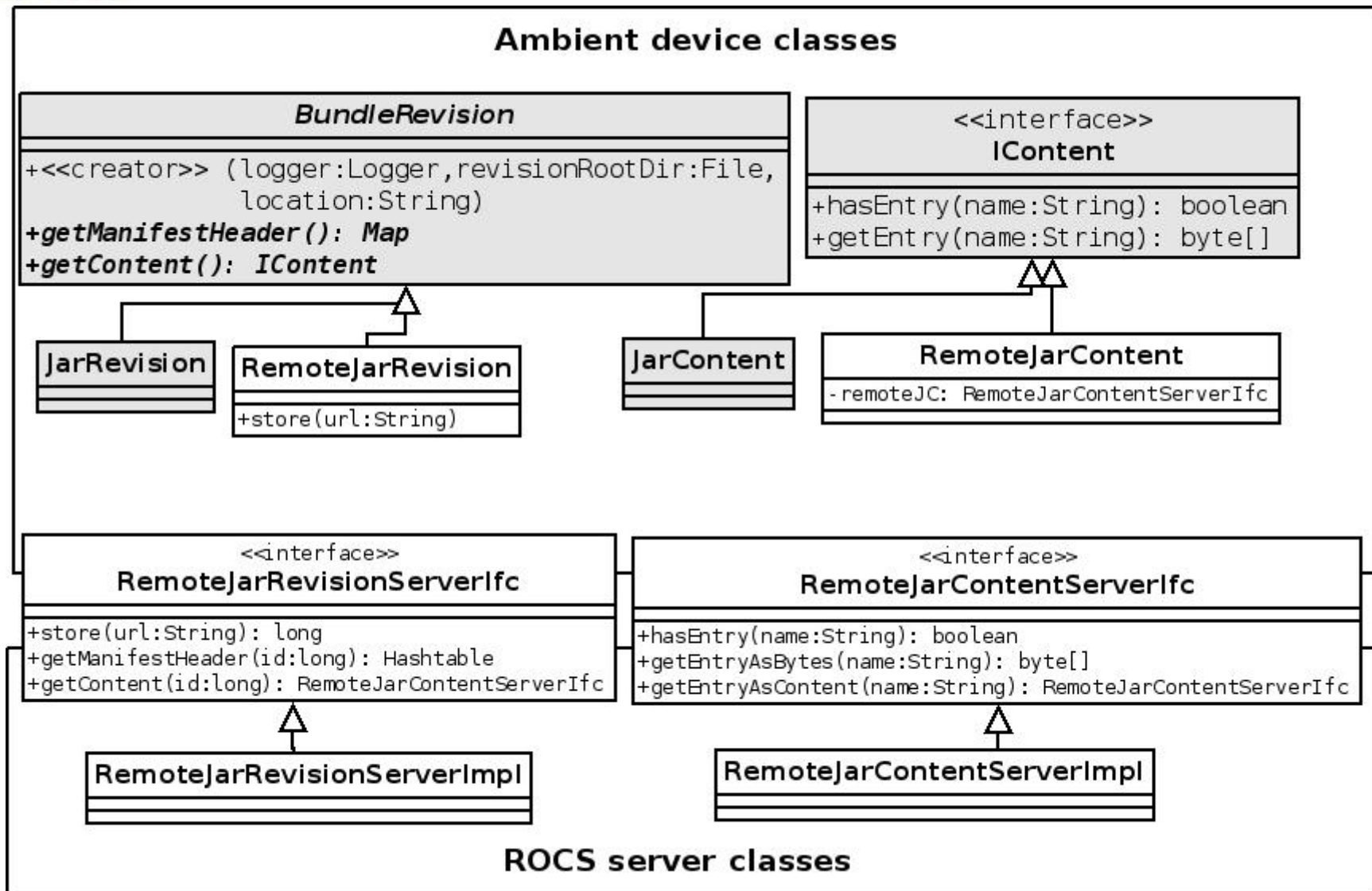
> start **reference**:<http://remotelocation/test.jar>

- 1) Récupération tcp de test.jar et sauvegarde sur le cache local
- 2) Extraction de l'archive sur le cache local
- 3) Fabrication d'un **BundleLoader**, chargeur de ressources dédié
- 4) Chargement du manifest
- 5) Contrôle de la cohérence du bundle avec le run-time
- 6) Instanciation reflective de l'activator
- 7) Invocation de la méthode start
- 8) **Chargement des classes nécessaires**

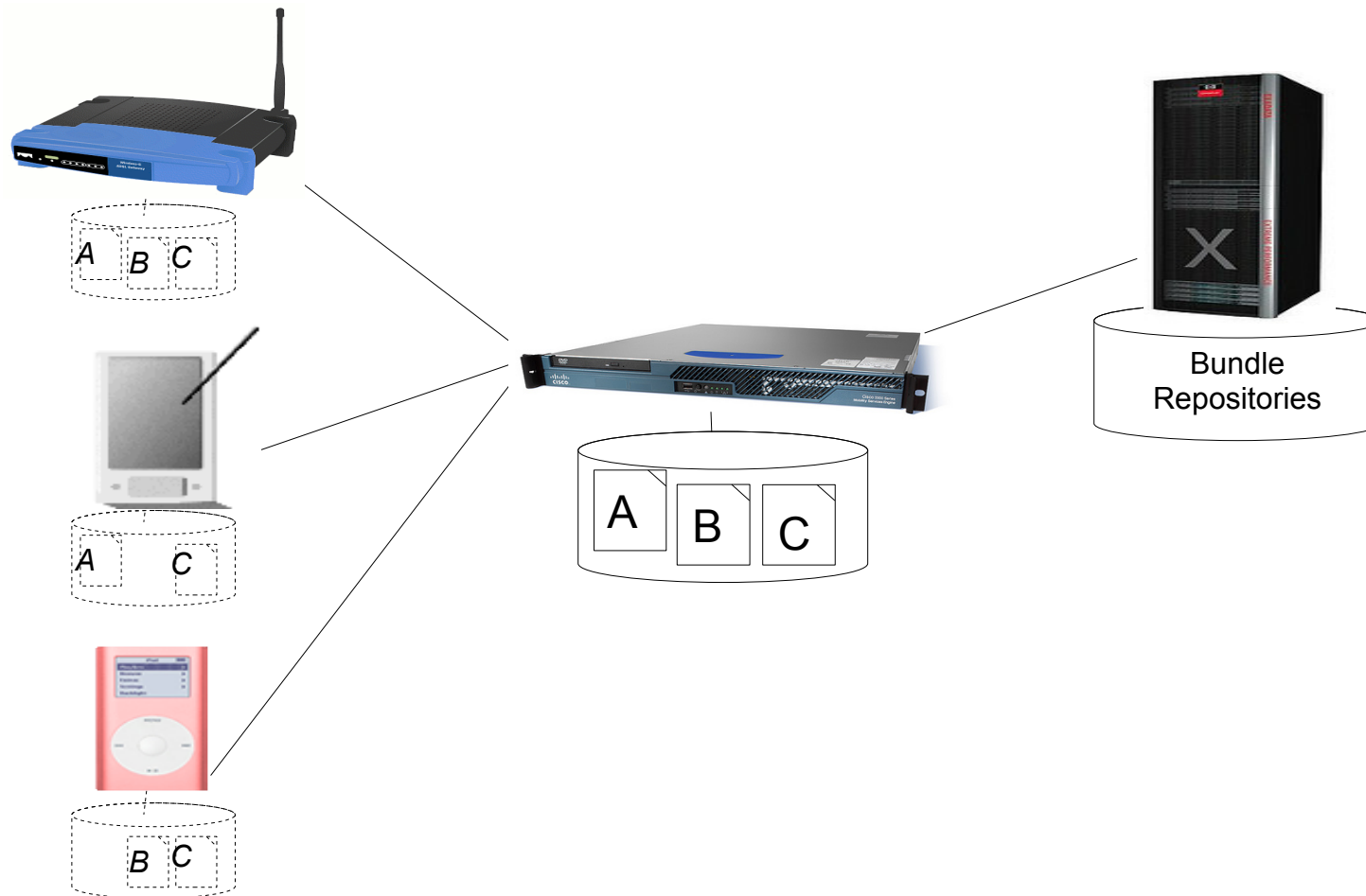
Fonctionnement

Sur un serveur de classes





Shéma général





So What ?



- On ne gagne de la place que pour les bundles standards !!!



So What ?



- On ne gagne de la place que pour les bundles standards !!!
 - ==> Pourquoi ne pas faire un bundle du Classpath ?
 - Exemple : Jamv+OSGi || GnuClasspath
 - Quel est le gain ?
 - Les 15Mo du classpath standard sont déportés



So What ?



- On ne gagne de la place que pour les bundles standards !!!
 - ==> Pourquoi ne pas faire un bundle du Classpath ?
 - Exemple : Jamv+OSGi || GnuClasspath
 - Quel est le gain ?
 - Les 15Mo du classpath standard sont déportés
 - Avoir un bootstrap minimal de Java/OSGi



Comment faire alors



- Bundle contenant le classpath
 - Il exporte l'intégralité des packages java, javax...
 - Normalement un bundle n'a pas le droit de le faire
- Chaque bundle doit exprimer les imports spécifiques
- On charge les classes en priorité par les bundles
 - - Normalement la délégation parente doit être primordiale
- Il faut un minimum de classes dans le boot initial : réseau, chargeurs de classes, jeux de caractères...
 - - Restriction du chargeur de classe sur le principe de package d'origine



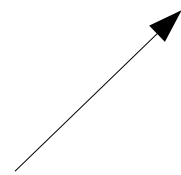
Quelques résultats



- NSLU2 (8MB flash, 32MB sdram), Jamvm, GnuClasspath

Bundle	Size (kB)
... shell.jar	50
... shell.tui.jar	11
... bundlerepository.jar	129
... log.jar	22
... mosgi.jmx.remotelogger.jar	12
... mosgi.jmx.agent.jar	99
... mosgi.jmx.registry.jar	13
... mosgi.jmx.rmiconnector.jar	83
... mosgi.console.ifc.jar	9
... mosgi.managedelements.osgiprobres.jar	11
... mosgi.managedelements.attributeaccessor.jar	5
... mosgi.managedelements.attributetester.jar	7
... mosgi.managedelements.attributeviewer.jar	15
Total	473

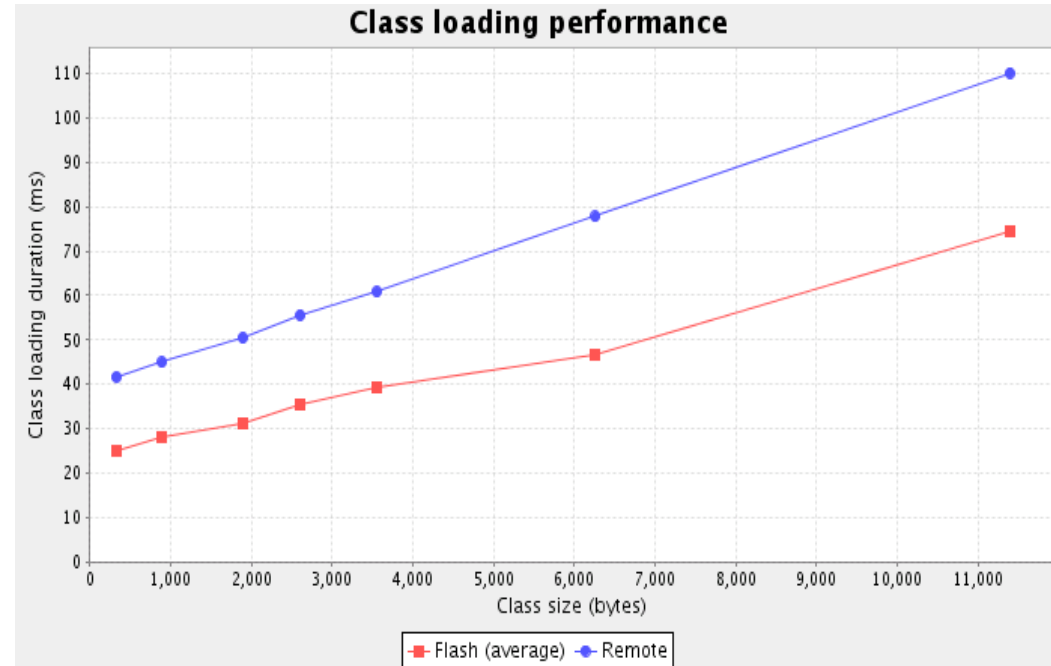
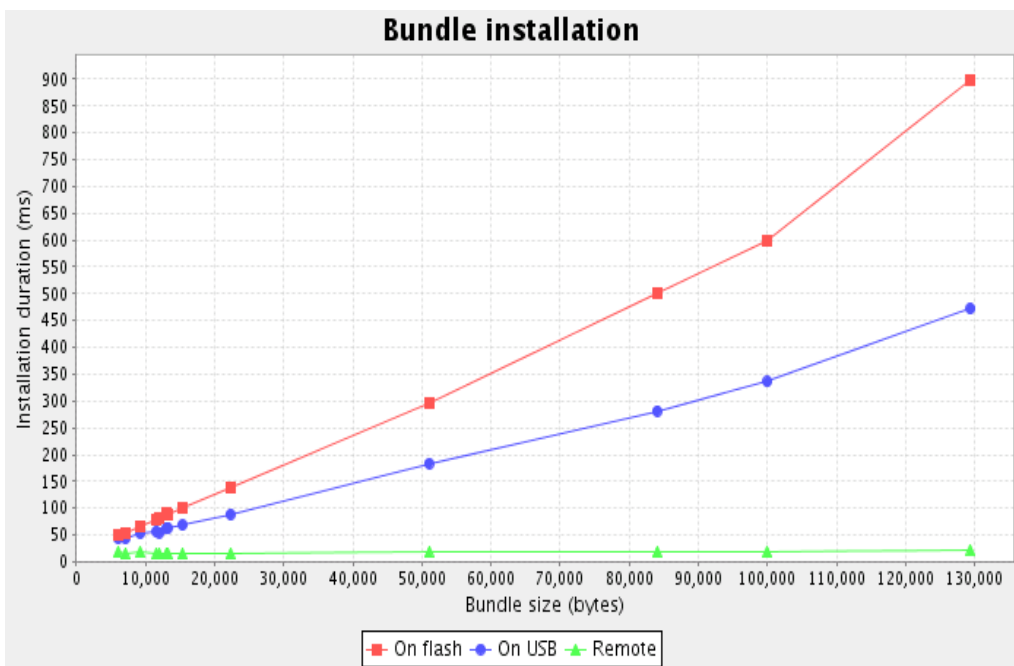
Layer	Size (kB)
... OS	5 000
... JamVM runtime	500
... Felix classes	350
... Standard Classpath classes	9 000
... 13 Bundles	500
Total	15 350



Layer	Size (kB)
... OS	5 000
... JamVM runtime	500
... Felix classes	350
... Standard Classpath classes	2 000
... 13 Bundles	5
Total	7 855



- NSLU2 (8MB flash, 32MB sdram), Jamvm, GnuClasspath



ROCS	off (s)	on (s)	Gain (%)
Startup	1.9	1.9	0
Install	12	6.2	48
Resolve	15	15.3	-2
Total time	29	23.4	19



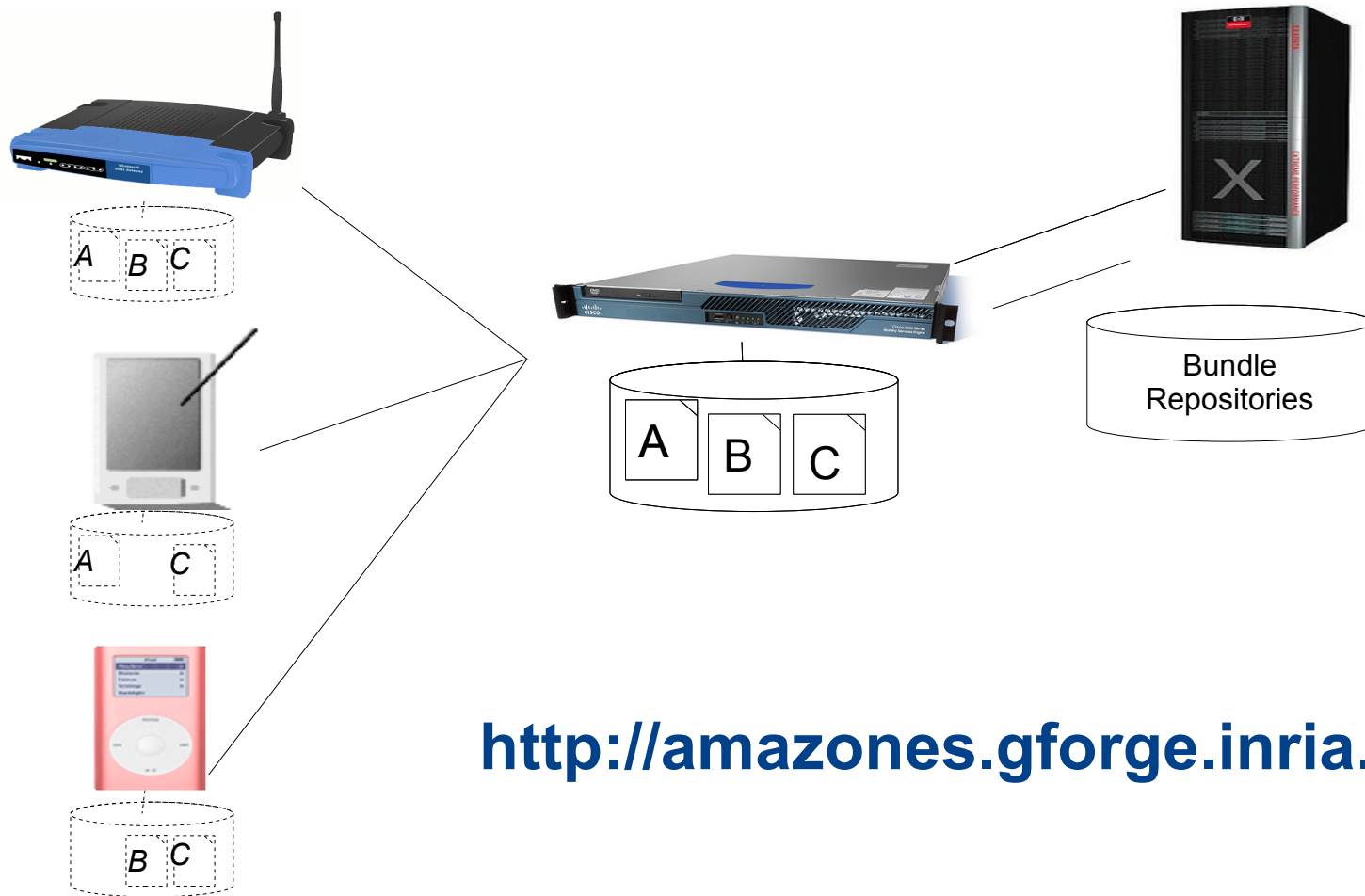
Quelques commentaires



- Violation de certaines règles de chargements
 - Oui, mais environnement de chargement dédié
 - Certaines semblent très « legacy »
- Portabilité pour des composants « natifs »
 - Au contraire, les bibliothèques natives sont bien mieux gérées
- Environnement de boot complexe
 - Oui, mais possibilité de remettre en cause des classes de base au run-time

Conclusion

- ROCS...CLOUD-OS



<http://amazones.gforge.inria.fr/rocs>