Dependability for Java Mobile Code
-
A pragmatic research view

Pierre Parrend
CITI Laboratory
INRIA – ARES Team
Lyon, France

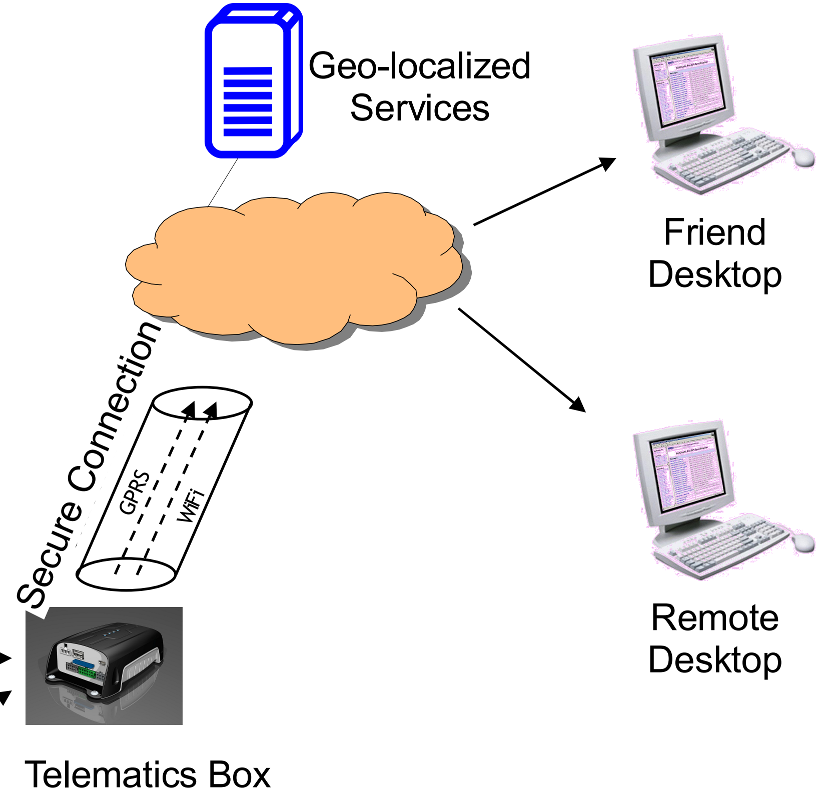
OWASP Swiss Chapter
Meeting July 2007
Zürich

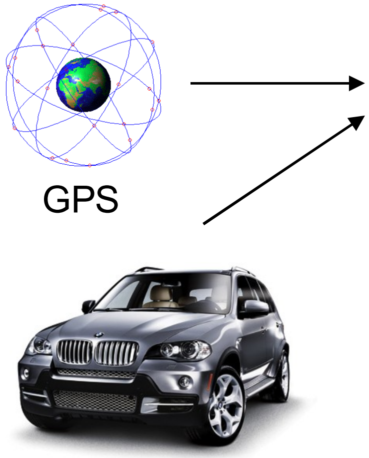
The Vision

- A Net of Applications
  - Interconnected world
    - Web Servers, Handheld Devices, Home PC, Home Boxes
    - Each device can consume and use services
  - Shifting programming model
    - Client Server webs apps are no longer satisfactory for mobile devices
    - Ressource limited devices need extensible execution environment: Mobile Java Apps (MIDP, OSGi, ...)
  - Consequence on Security
    - Specific approach to security concerns

The Vision

- Example – On-board desktop





Summary

- **The OWASP and the Java World**
  - The OWASP Java Project
  - From Client-Server to extensible Applications
- Dependability for Java Mobile Code
- A Contribution for Hardened OSGi Platforms

Java and the OWASP

- The OWASP Java Project
  - Started 30 June 2006
  - Mailing List: 111 members
  - Articles: 26
  - Growing ...
- Related Development Projects
  - LAPSE
    - Lightweight Analysis for Program Security in Eclipse
    - Benjamin Livshits

Java and the OWASP

- The OWASP Java Project
  - Targeted at Web Application Servers
  - Focus on 4 questions
    - J2EE Security for Architects
    - J2EE Security for Developpers
    - J2EE Security for Deployers
    - J2EE Security for Analysts and Testers
  - Work in progress

From Client-Server to Extensible Applications

- Motivation
  - Restricted applications for mobile devices
- Classical Web Client-Server Approach
  - Deskop Browser - rich user experience requires sufficient client side-resources (memory, screen size)
  - Java Applets, Web start (and many others) for Web-based applications
- Connection and Apps for Mobile Devices
  - Wap access for mobile devices
  - Default apps for mobile devices

From Client-Server to Extensible Applications

- Solution: Extensible Component Platforms for embedded devices
  - Existing technologies
    - Java MIDP, OSGi
  - Target systems
    - Mobile phones, automotive entertainment, home gateways, e-health systems
  - Features
    - Discovery of Apps Repositories
    - Installation of new Apps during runtime
    - Multi-Application systems
    - Uninstallation of Apps

From Client-Server to Extensible Applications

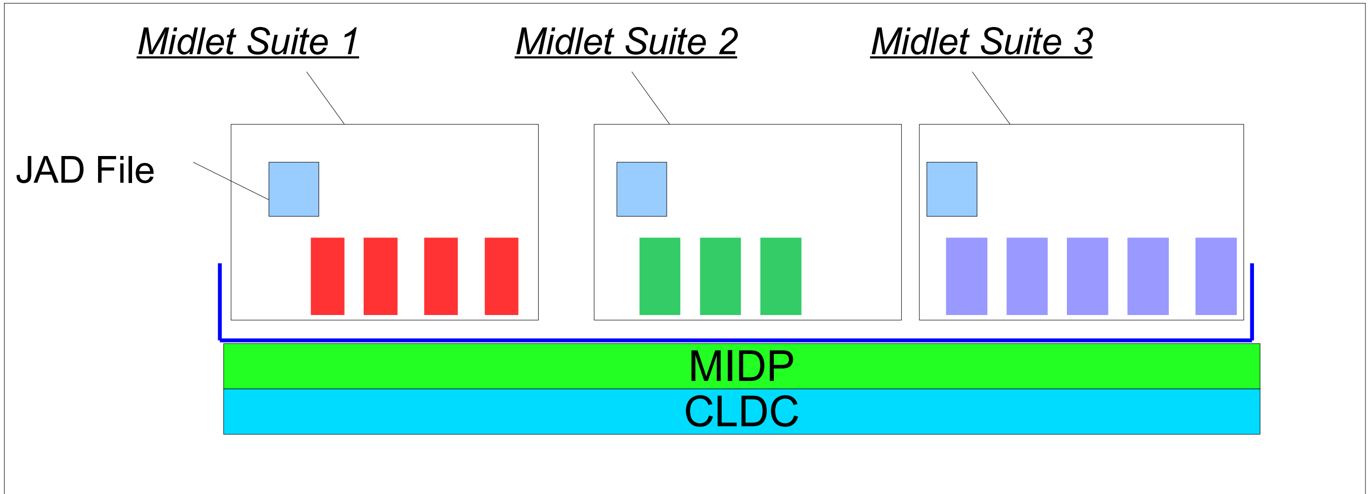
- Extensible Component Platforms prove to be powerfull for server management too
  - Benefits
    - No reboot required
    - Centralized (and possibly remote) component management
    - Transparent update of System and Applications
  - Eclipse IDE
    - Based on OSGi Equinox
  - IBM Websphere 6.1
  - JBoss
    - OSGi Felix

From Client-Server to Extensible Applications

- Java Extensible Component Platforms
  - MIDP vs. OSGi
  - MIDP
    - CLDC (Connected Limited Device Configuration) Profile
    - Very lighweigth environments
    - e.g.: Mobile Phones
  - OSGi
    - J2ME CDC (Connected Device Configuration) Foundation Profile
    - Leightweight or standard environments
    - e.g: PDAs

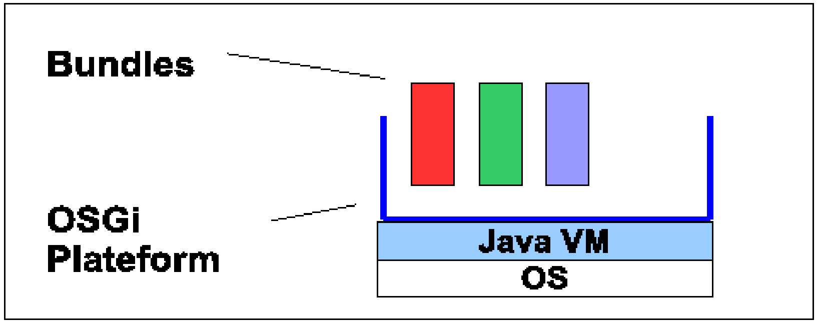
From Client-Server to Extensible Applications

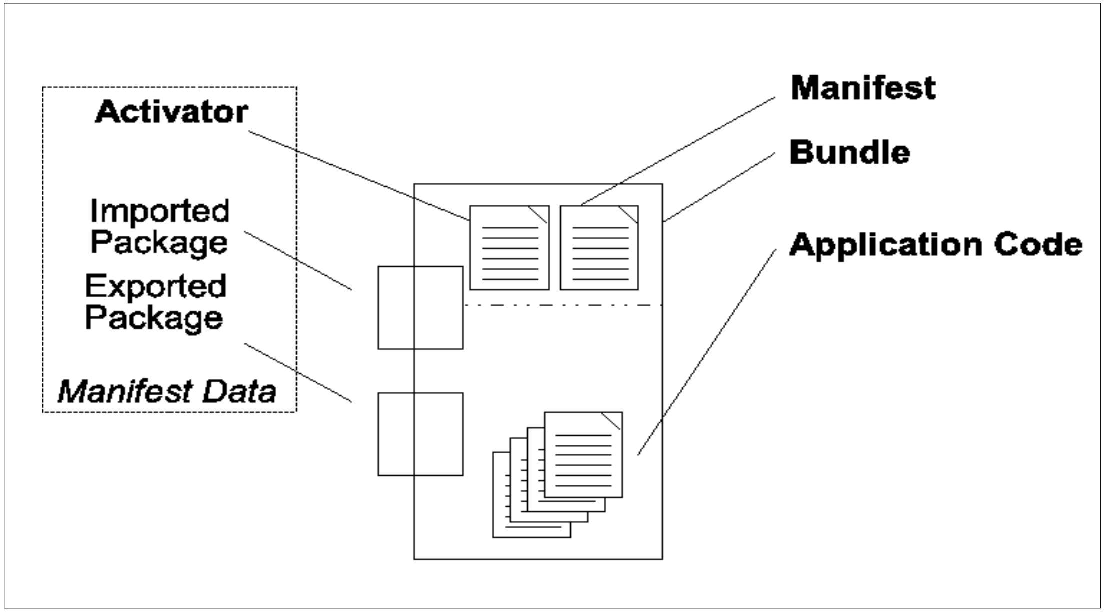
- MIDP
  - Mobile Information Device Profile
  - Defined by Sun
  - Applications
    - Middlet Suites
    - Defined in an external JAD File
      - Java Application Descriptor



From Client-Server to Extensible Applications

- OSGi
  - Was 'Open Service Gateway Initiative'
    - Is now an adjective
  - Forstered by the OSGi Alliance
  - The Platform - The Bundles

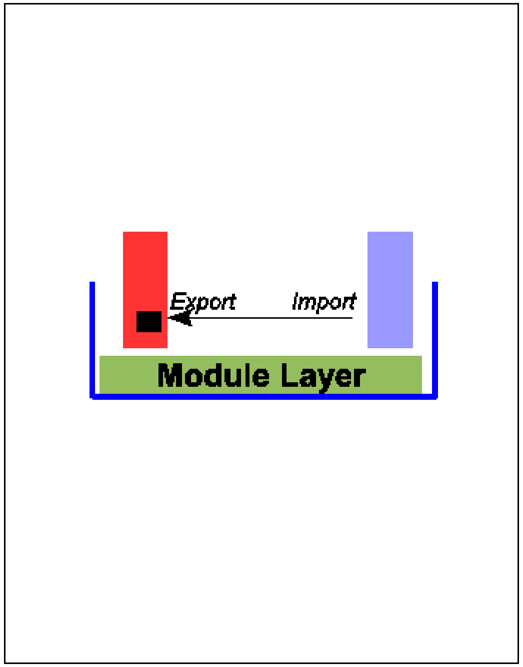


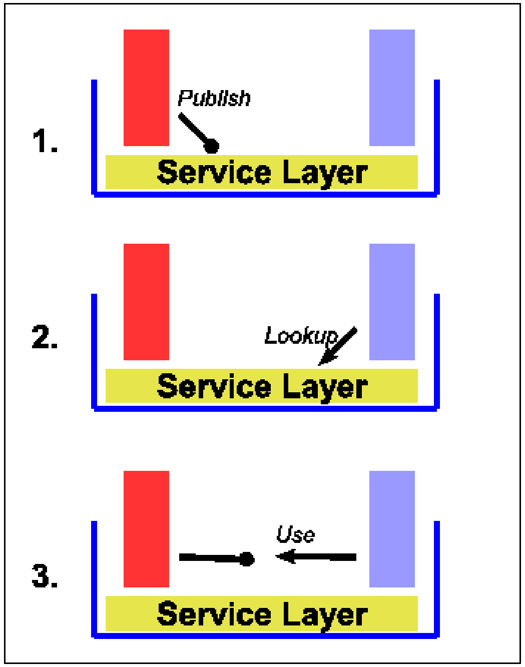


24/07/2007

From Client-Server to Extensible Applications

- OSGi
  - Communication between bundles
    - Package or Services
    - Internal Description, enables Dependency Resolution
      - And thus dynamic discovery





Summary

- The OWASP and the Java World
- **Dependability for Java Mobile Code**
  - From Security to Dependability
  - Security for Java Mobile Code: State of the Art
- A Contribution for Hardened OSGi Platforms

From Security to Dependability

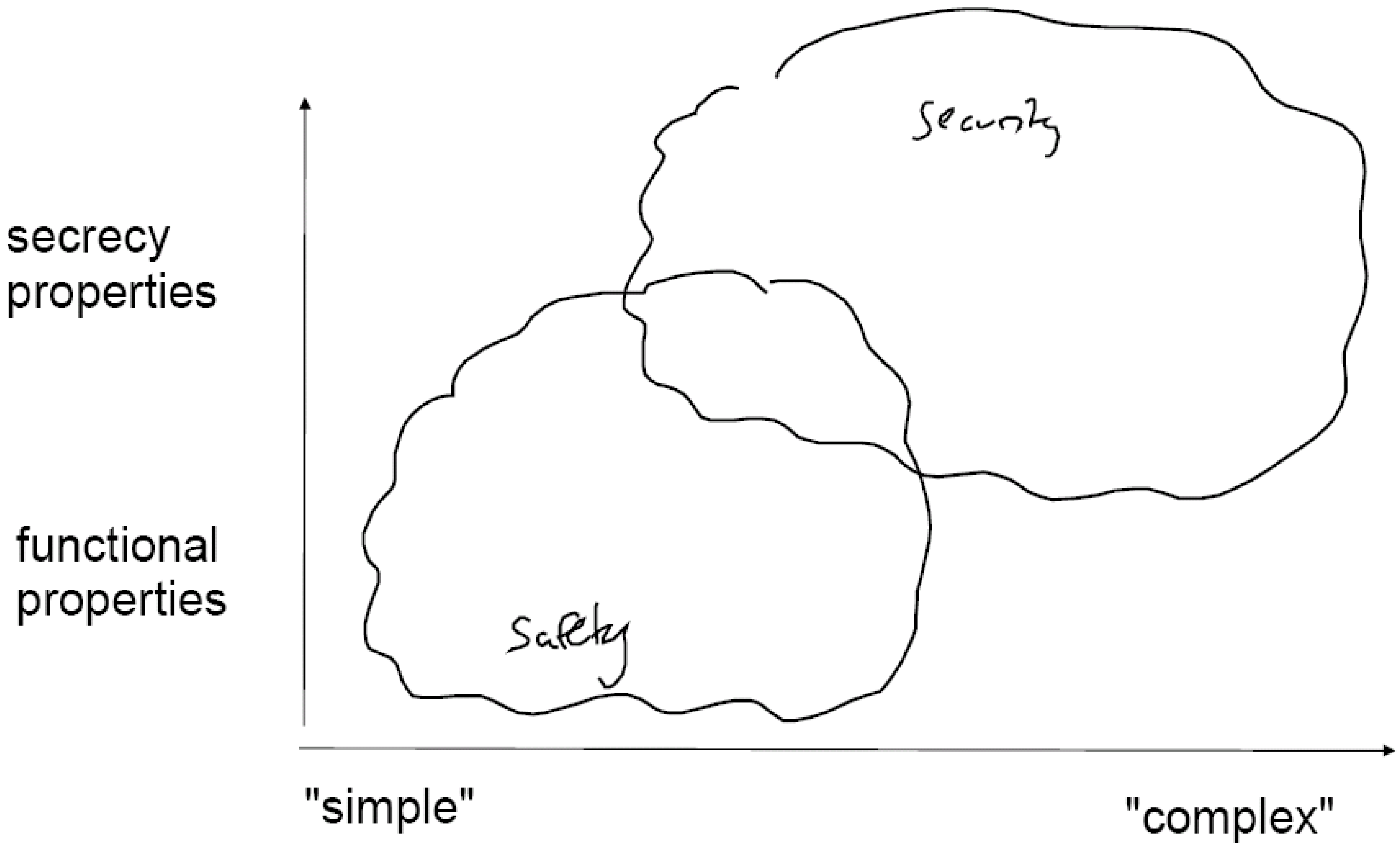
- Java Extensible Component Platforms: an Evolving Threat Model
  - Web Servers
    - Hackers can come from the Internet
    - Attack Surface is kept as small as possible
  - Extensible Component Platforms
    - Hackers can come from the Internet
    - Hackers can hide malware in Components
    - Attack Surface is as big as the Specification ...
      - Or at least is made of all actions the Component is allowed to do

From Security to Dependability

- A new approach to security is required
  - A firewall is not enough
    - AAA model outdated
  - Control on code is more necessary than ever
    - It is so easy to block a system when executing code on it
  - Current JVMs are designed for secure execution of single applications
    - Multi-Application save ressource
    - But are likely to bring big troubles
  - Dependability
    - Security + Robustness

From Security to Dependability

- Dependability



secrecy properties

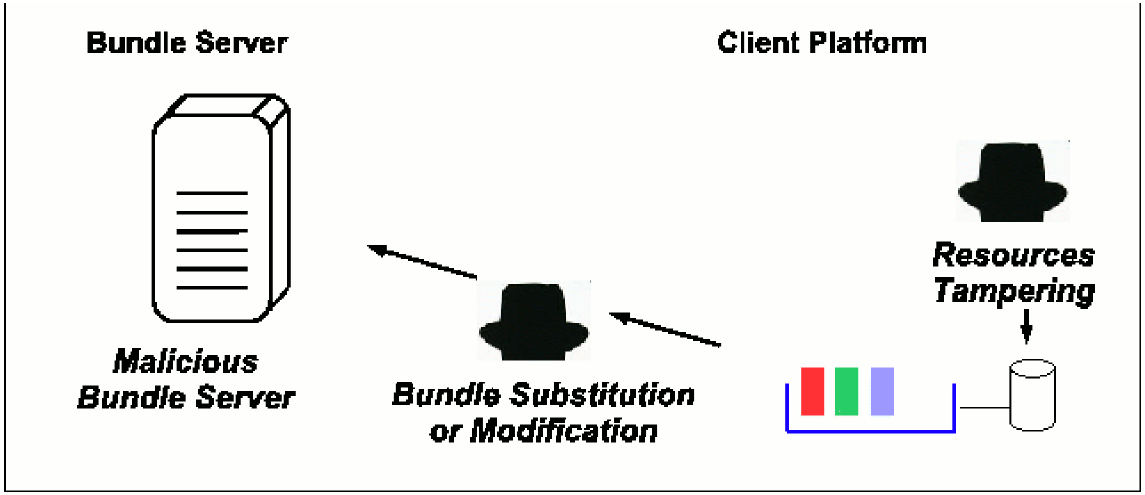
functional properties

"simple"

"complex"

From Security to Dependability

- Threat Model for Extensible Component Platform
  - Deployment



From Security to Dependability

- Threat Model for Extensible Component Platform
  - Execution – At the Example of the OSGi Platform
    - Each Element of the Execution Platform Can be the source of vulnerabilities
    - JVM
      - Execution Platform
      - API
    - OSGi Platform
      - Life-Cycle Layer – bundle management
      - Module Layer – package management
      - Service Layer

Security for Java Mobile Code: State of the Art

- Principle of Security for Java Code
  - Strong Data Typing
    - No buffer overflow
  - Automatic Memory Management
    - No memory leak
  - Bytecode verification
    - Before execution
  - Secure Class Loading
    - Permission mechanism

Security for Java Mobile Code: State of the Art

- MIDP Security

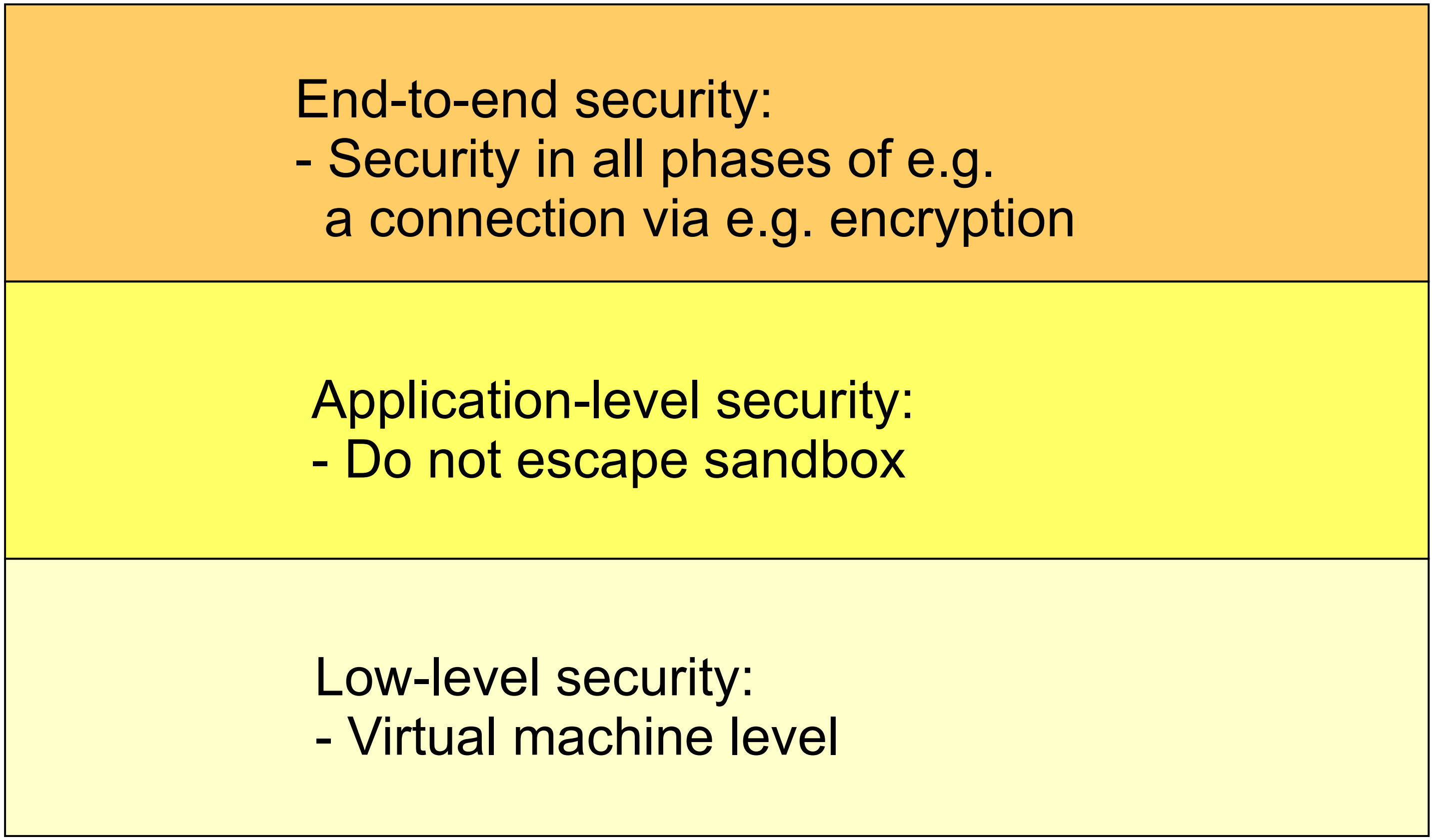
  Tommi Mikkonen, Uni. Tampere (Fi.)

  - Three security levels
    - Low-level ~ virtual machine level security
    - Application-level ~ applications do not escape ’sandbox’
    - End-to-end ~ Security in all phases of e.g. a connection via e.g. encryption
  - Digital signature to enable trusted applications (only after CLDC 1.1)
    - Manufacturer, operator, trusted 3rd party, untrusted
    - Needed for phone calls, push networking features, etc
    - User authorization may also be used if the trust level is not enough for certain feature
  - Midlet Signature: in the JAD File

Security for Java Mobile Code: State of the Art

- MIDP Security

Tommi Mikkonen, Uni. Tampere (Fi.)



End-to-end security:
- Security in all phases of e.g. a connection via e.g. encryption

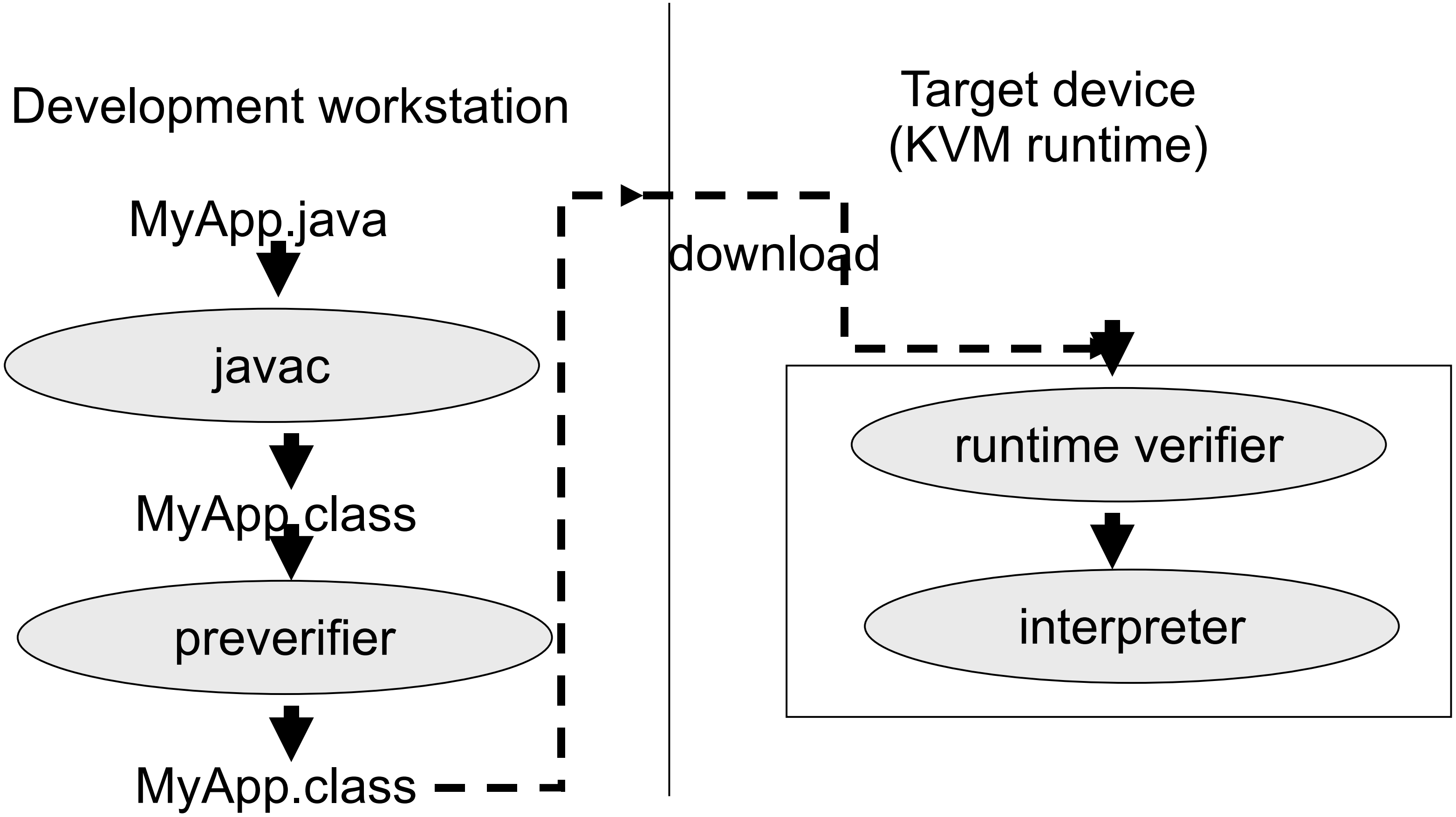
Application-level security:
- Do not escape sandbox

Low-level security:
- Virtual machine level

Security for Java Mobile Code: State of the Art

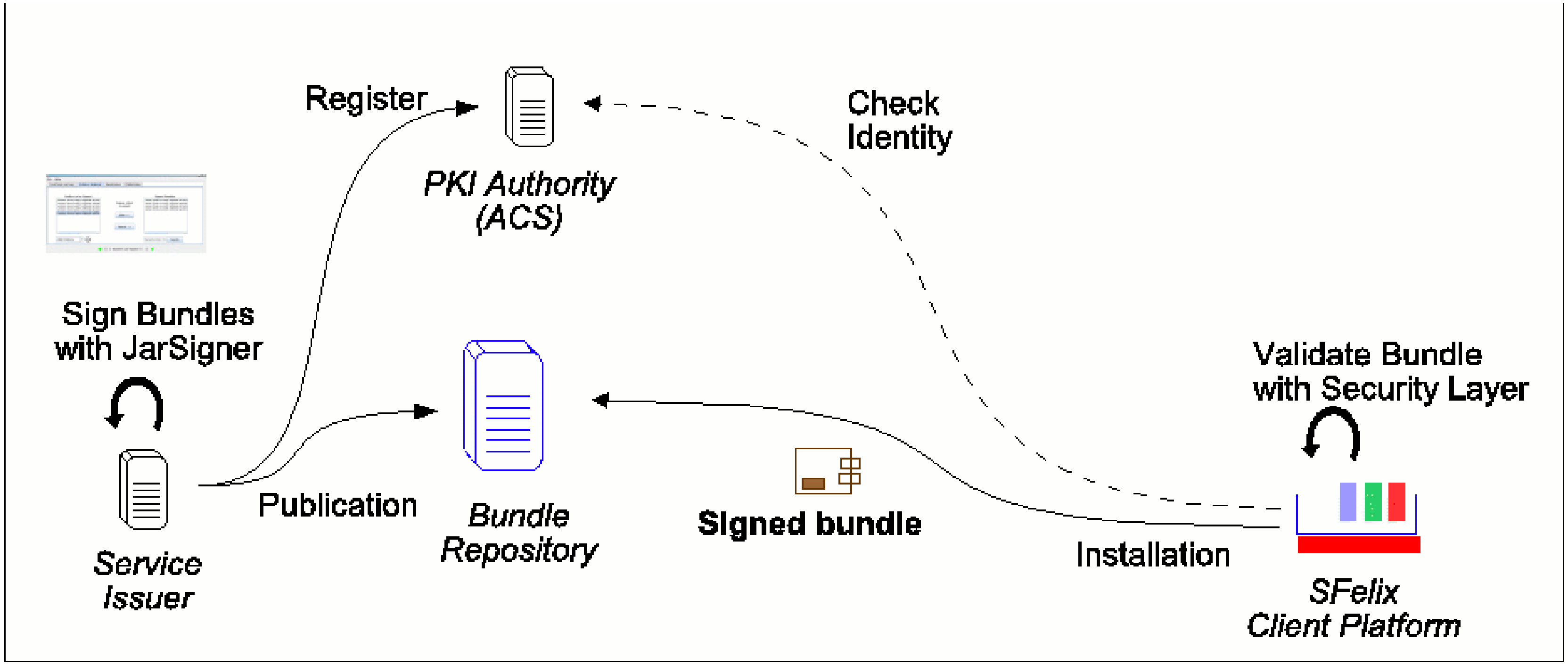
- MIDP Security

<u>*Tommi Mikkonen, Uni. Tampere (Fi.)*</u>



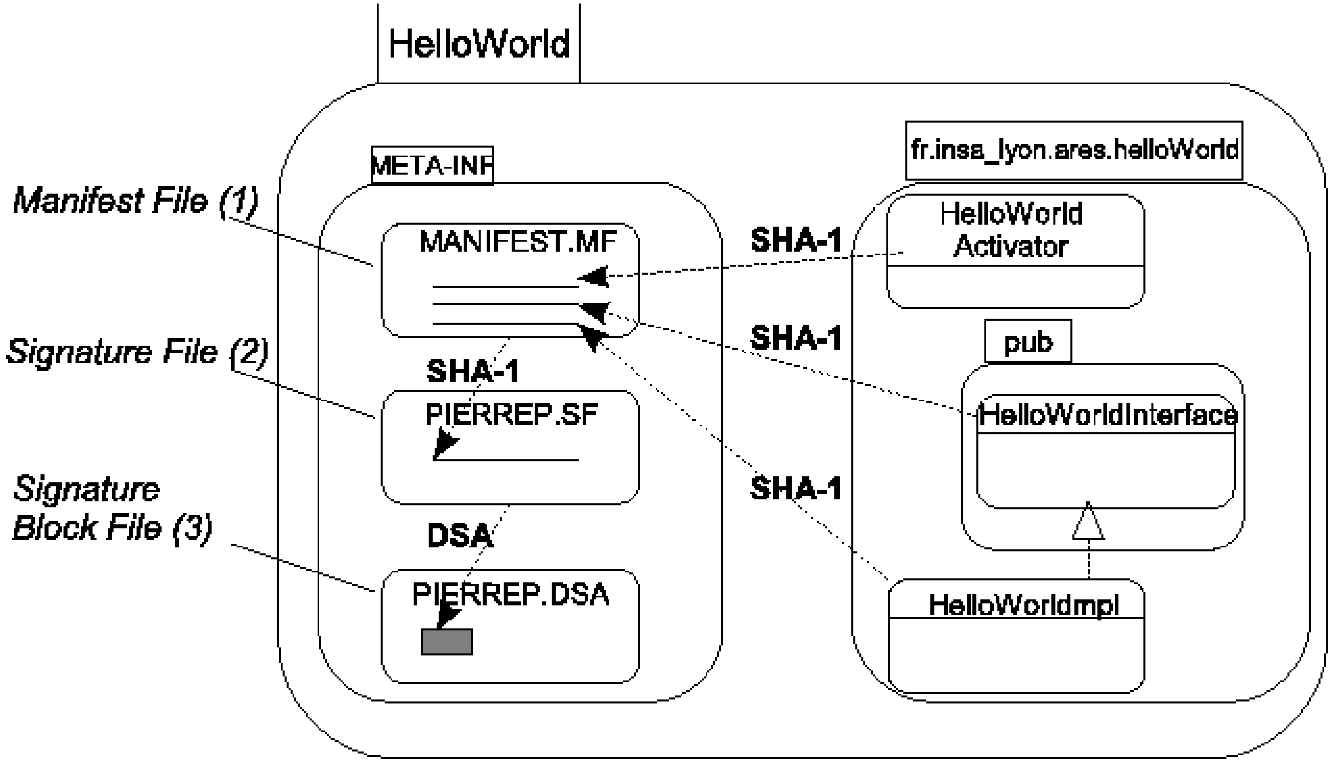
Security for Java Mobile Code: State of the Art

- OSGi Security
  - Secure Deployment



Security for Java Mobile Code: State of the Art

- OSGi Security
  - Digital Bundle Signature

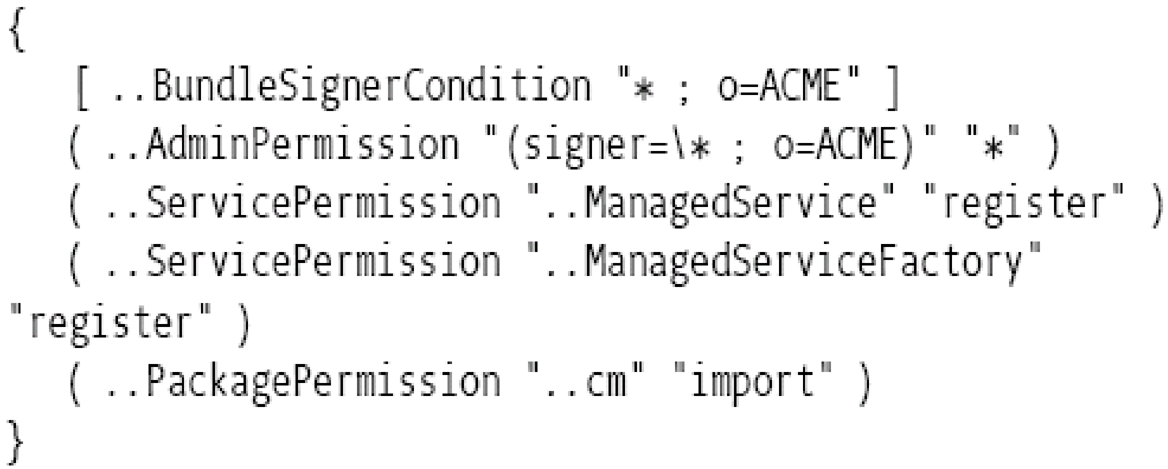


Security for Java Mobile Code: State of the Art

- OSGi Security
  - Java Permissions
  - OSGi Permissions
    - AdminPermission
      - Lifecycle, metadata, listener, execute
    - PackagePermission
      - Export, import
    - ServicePermission
      - Register, get

Security for Java Mobile Code: State of the Art

- OSGi Security
  - Permission Management
    - At runtime
  - Conditional Permissions
    - Perform additional check



```
{
   [ ..BundleSignerCondition "* ; o=ACME" ]
   ( ..AdminPermission "(signer=\* ; o=ACME)" "*" )
   ( ..ServicePermission "..ManagedService" "register" )
   ( ..ServicePermission "..ManagedServiceFactory"
"register" )
   ( ..PackagePermission "..cm" "import" )
}
```

Security for Java Mobile Code: State of the Art

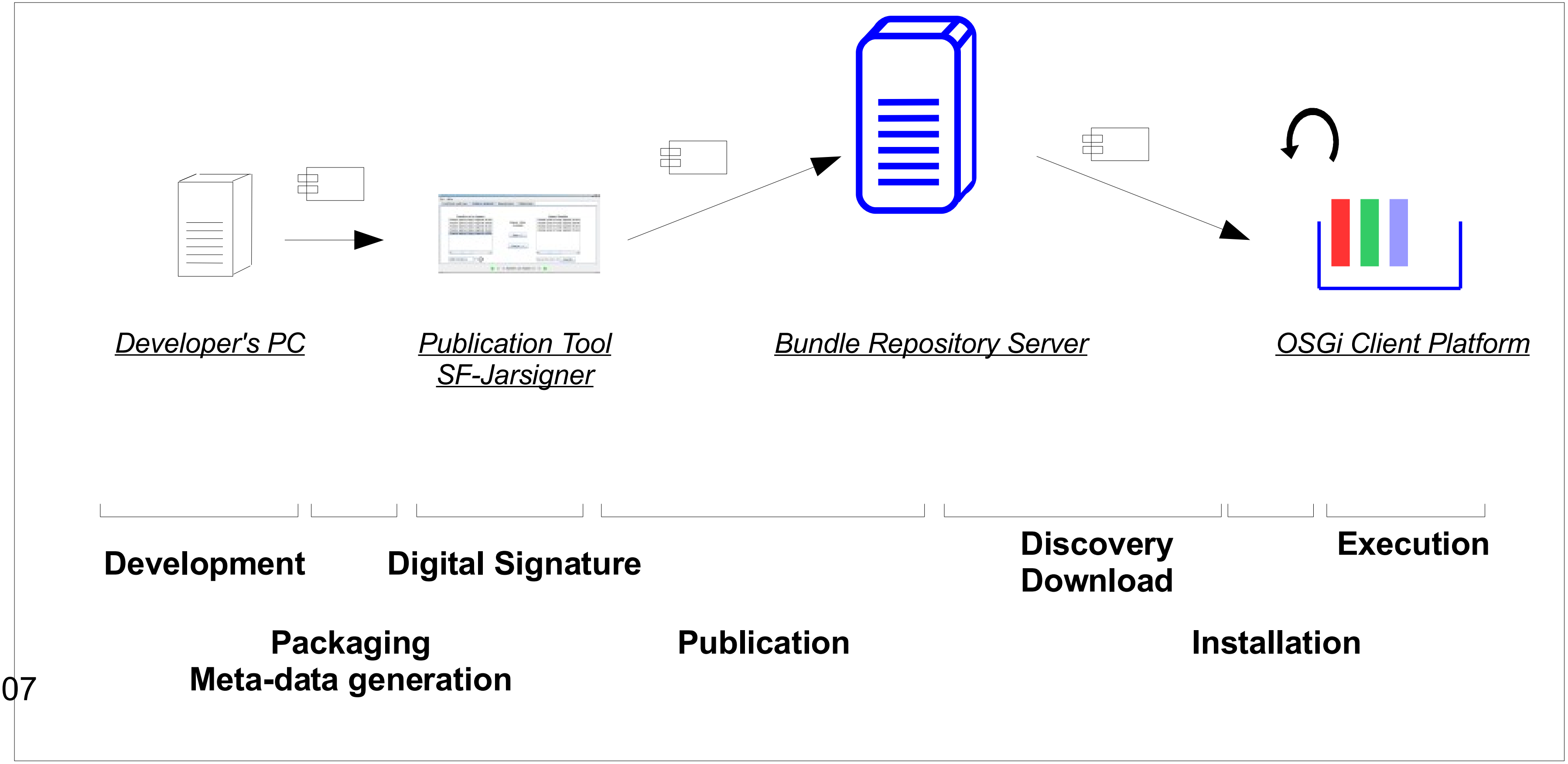
- Current Security Level
  - Secure Deployment
  - Restrictions on execution are possible
- Requirements
  - No Guarantee on the executed code
    - Simply trust the Issuer
  - Research efforts
    - Proof Carrying Code
      - Can only proove subsets of programming languages
      - 'I can tell you that your virus will never crash', Peter Lee

Summary

- The OWASP and the Java World
- Dependability for Java Mobile Code
- **A Contribution for Hardened OSGi Platforms**
  - Engineering Dependable Applications
  - Toward a Hardened OSGi Platform

Engineering Dependable Applications

- Requirement
  - Life-Cycle long support of security
- The Bundle Life-Cycle



Development

Packaging
Meta-data generation

Digital Signature

Publication

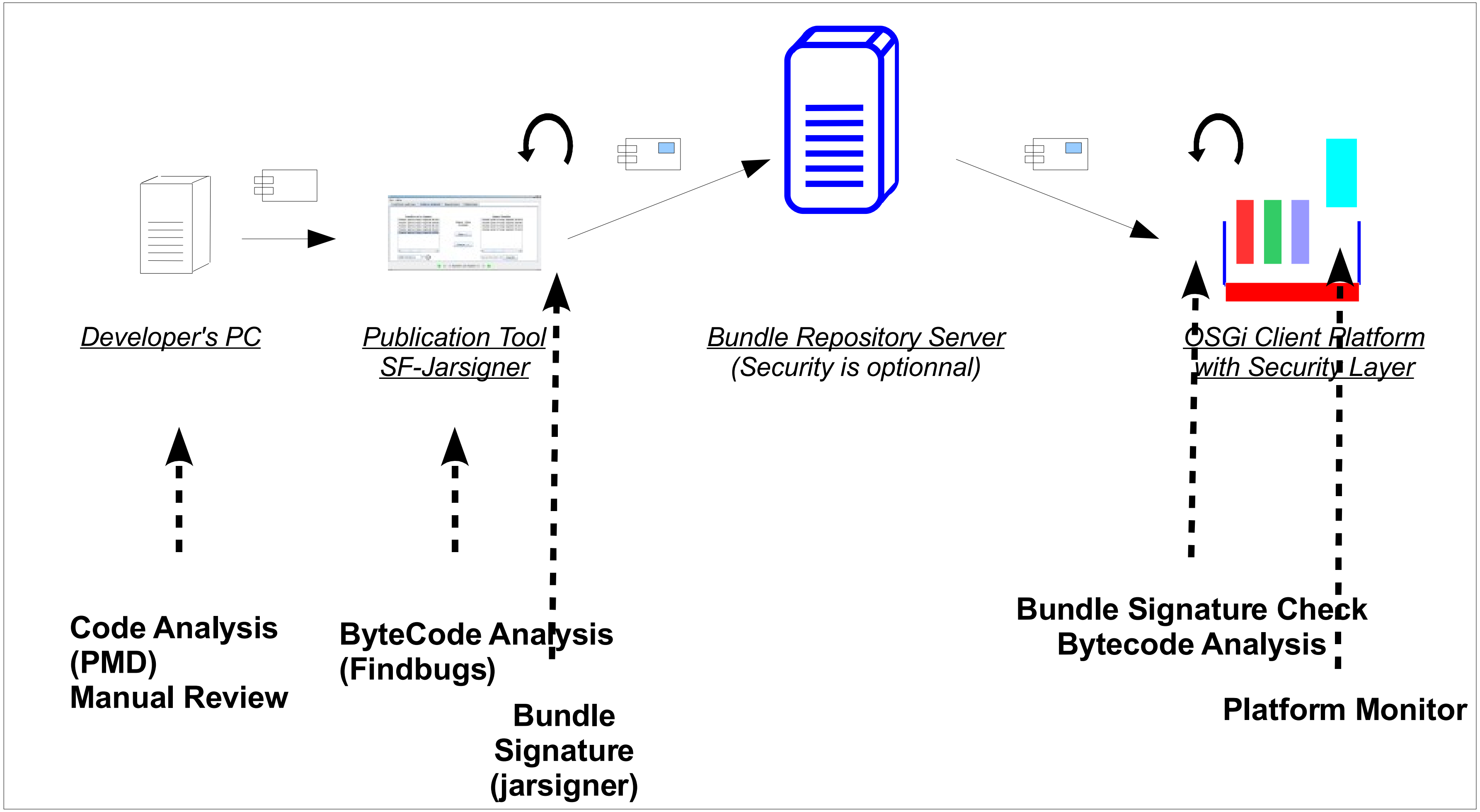
Discovery
Download

Installation

Execution

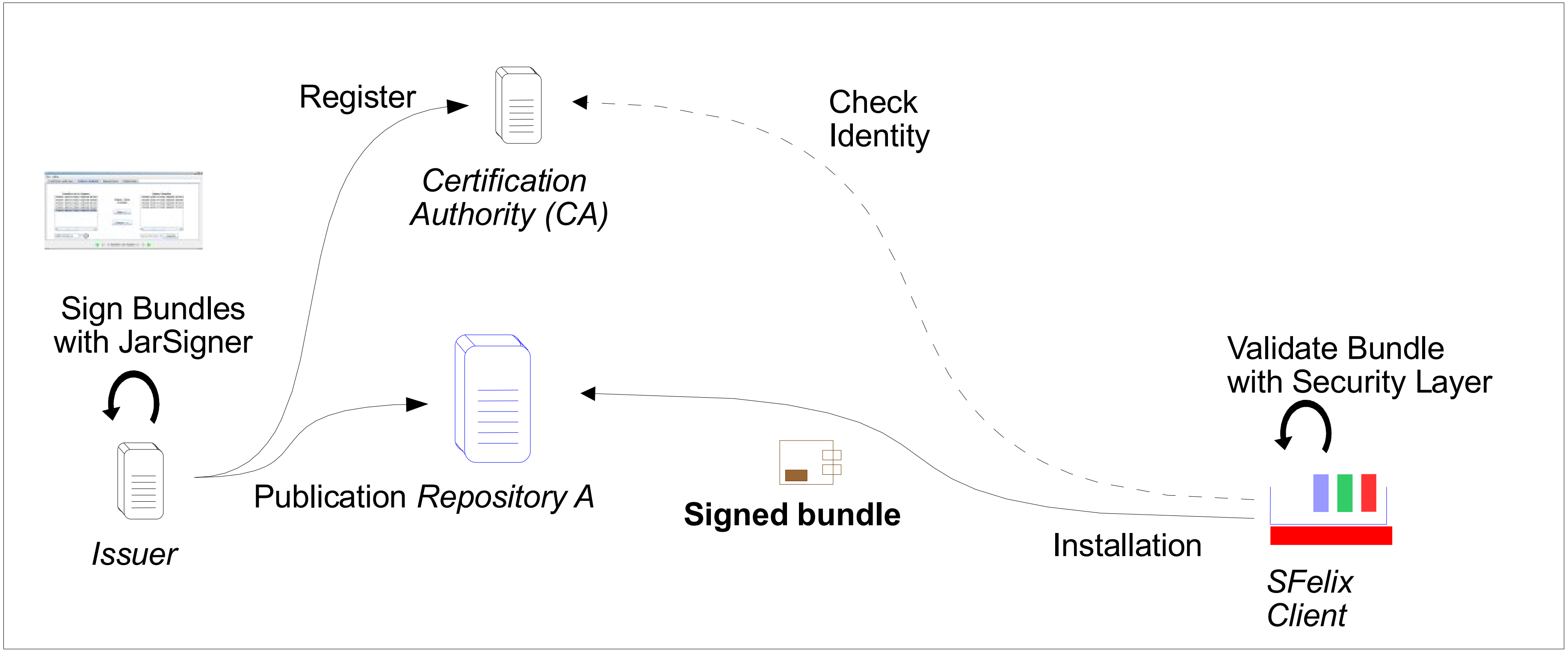
Engineering Dependable Applications

- Secure Coding throughout Bundle Life-Cycle



Engineering Dependable Applications

- Tools for Secure Deployment of OSGi Bundle
  - SF-Jarsigner, http://sf-jarsigner.gforge.inria.fr
  - SFelix, http://sfelix.gforge.inria.fr

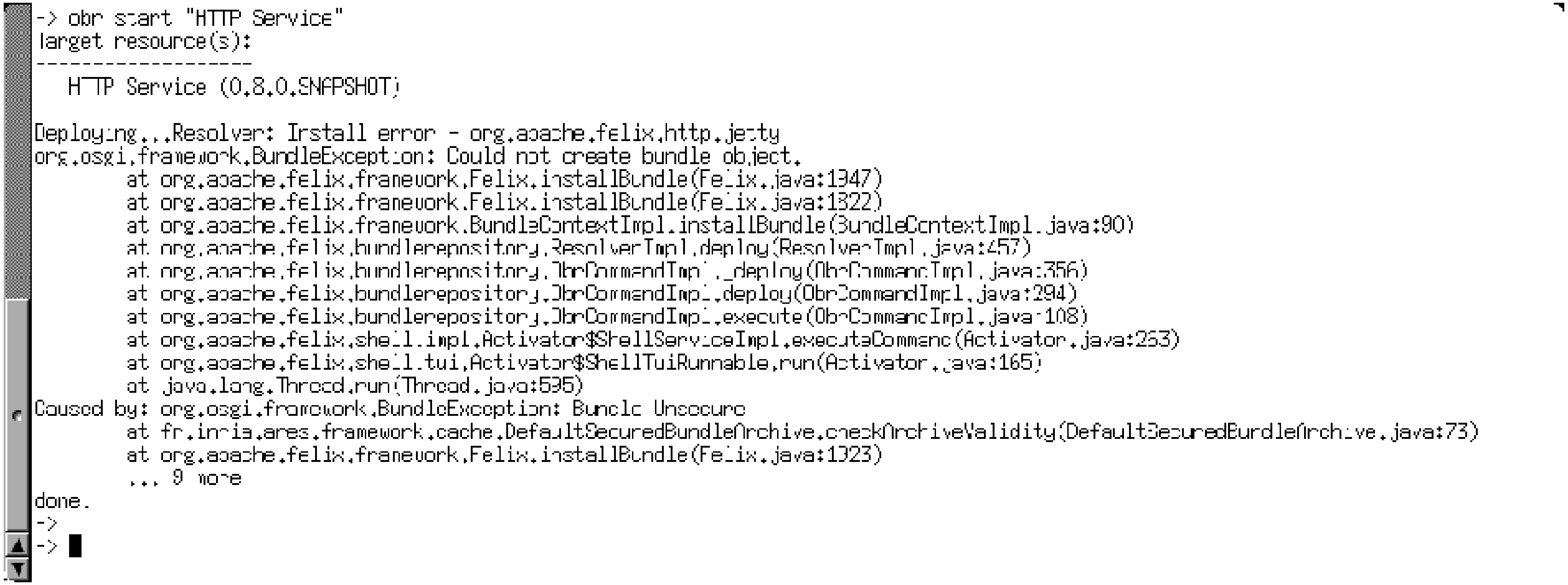

Register
Certification Authority (CA)
Check Identity
Sign Bundles with JarSigner
Issuer
Publication Repository A
Signed bundle
Installation
Validate Bundle with Security Layer
SFelix Client

Engineering Dependable Applications

- Sfelix
  - http://sfelix.gforge.inria.fr/
  - Sfelix v0.1
    - OSGi Release 4 Implementation of the Bundle Signature Validation Process
    - Beware of JVM-only solutions !
  - Sfelix v0.2
    - Robust against ill-coded Bundles
    - In a near future – still need to be published

Engineering Dependable Applications

- Sfelix

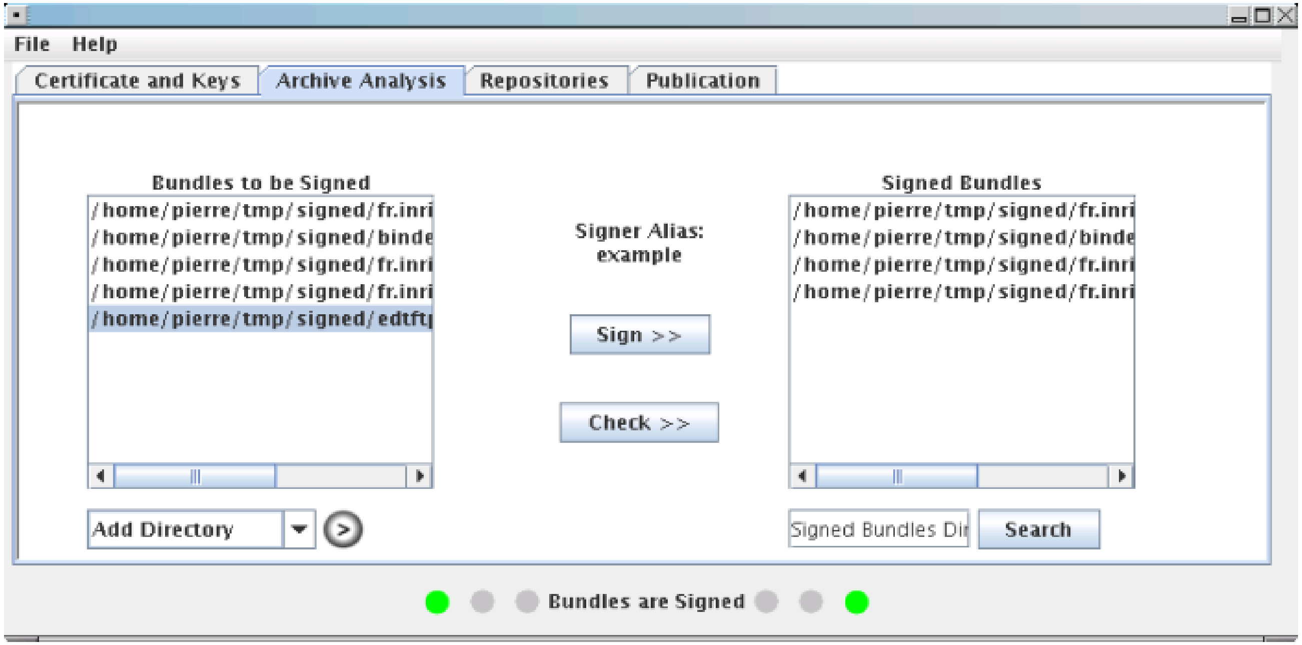


```
-> obr start "HTTP Service"
Target resource(s):
--------------------
   HTTP Service (0.8.0.SNAPSHOT)

Deploying...Resolver: Install error - org.apache.felix.http.jetty
org.osgi.framework.BundleException: Could not create bundle object.
        at org.apache.felix.framework.Felix.installBundle(Felix.java:1347)
        at org.apache.felix.framework.Felix.installBundle(Felix.java:1322)
        at org.apache.felix.framework.BundleContextImpl.installBundle(BundleContextImpl.java:90)
        at org.apache.felix.bundlerepository.ResolverImpl.deploy(ResolverImpl.java:457)
        at org.apache.felix.bundlerepository.ObrCommandImpl._deploy(ObrCommandImpl.java:356)
        at org.apache.felix.bundlerepository.ObrCommandImpl.deploy(ObrCommandImpl.java:294)
        at org.apache.felix.bundlerepository.ObrCommandImpl.execute(ObrCommandImpl.java:108)
        at org.apache.felix.shell.impl.Activator$ShellServiceImpl.executeCommand(Activator.java:263)
        at org.apache.felix.shell.tui.Activator$ShellTuiRunnable.run(Activator.java:165)
        at java.lang.Thread.run(Thread.java:595)
Caused by: org.osgi.framework.BundleException: Bundle Unsecure
        at fr.inria.ares.framework.cache.DefaultSecuredBundleArchive.checkArchiveValidity(DefaultSecuredBundleArchive.java:73)
        at org.apache.felix.framework.Felix.installBundle(Felix.java:1323)
        ... 9 more
done.
->
->
```

Engineering Dependable Applications

- The SF-JarSigner Tool
  - http://sf-jarsigner.gforge.inria.fr/
  - The Archive Analysis PanelF



Bundles to be Signed

/home/pierre/tmp/signed/fr.inri
/home/pierre/tmp/signed/binde
/home/pierre/tmp/signed/fr.inri
/home/pierre/tmp/signed/fr.inri
/home/pierre/tmp/signed/edtft

Signer Alias:
example

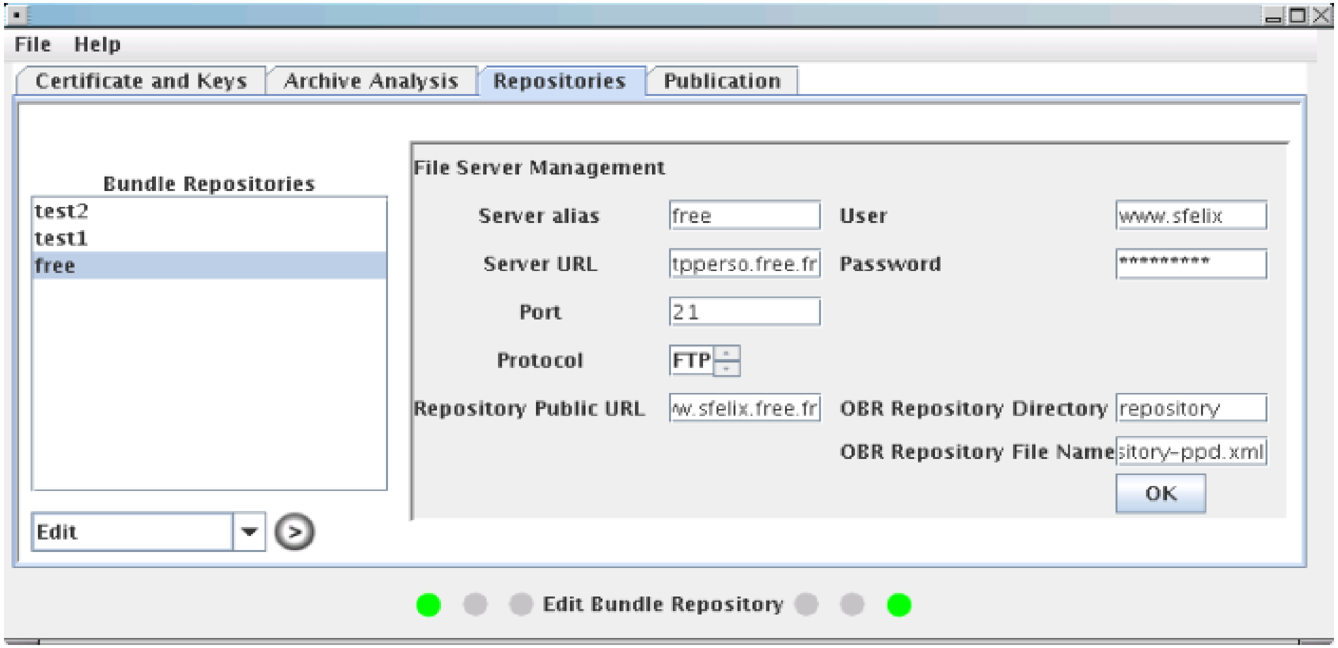
Signed Bundles

/home/pierre/tmp/signed/fr.inri
/home/pierre/tmp/signed/binde
/home/pierre/tmp/signed/fr.inri
/home/pierre/tmp/signed/fr.inri

● ● ● Bundles are Signed ● ● ●

Engineering Dependable Applications

- The SF-JarSigner Tool
  - The Bundle Repository Management Panel



Bundle Repositories

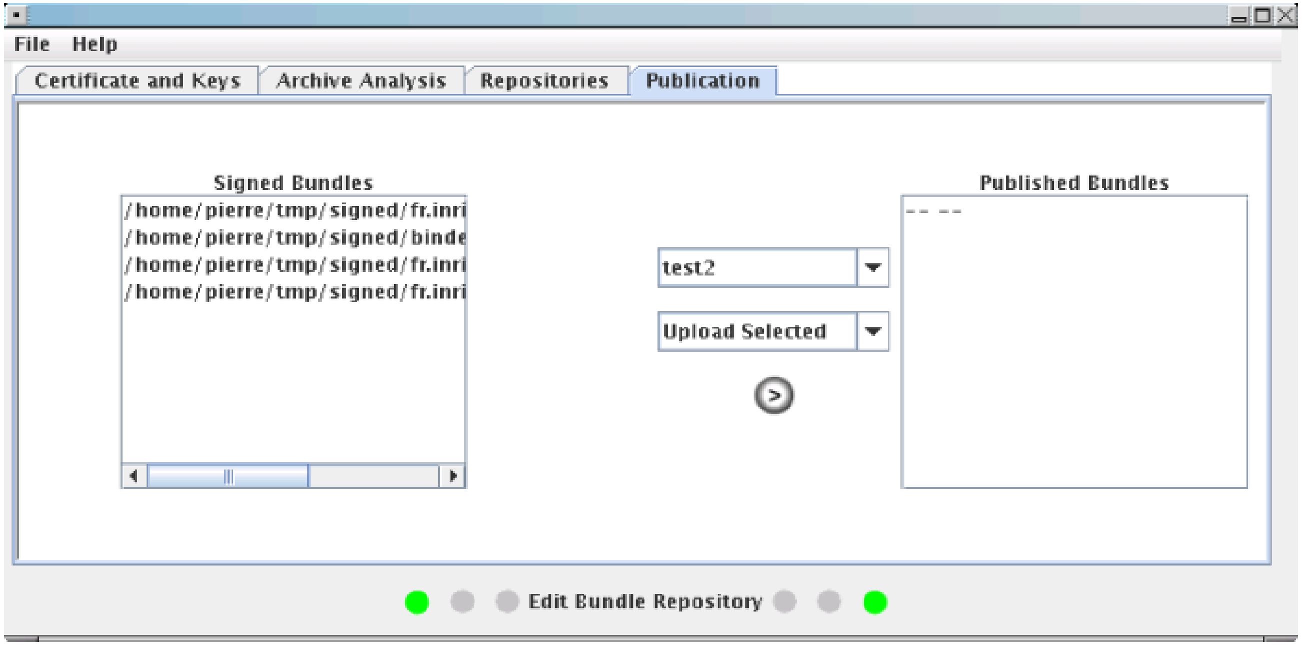
test2
test1
free

File Server Management

Server alias	free	User	www.sfelix
Server URL	tpperso.free.fr	Password	**********
Port	21		
Protocol	FTP		
Repository Public URL	w.sfelix.free.fr	OBR Repository Directory	repository
		OBR Repository File Name	sitory-ppd.xml

Engineering Dependable Applications

- The SF-JarSigner Tool
  - The Bundle Publication Panel



Signed Bundles

/home/pierre/tmp/signed/fr.inri
/home/pierre/tmp/signed/binde
/home/pierre/tmp/signed/fr.inri
/home/pierre/tmp/signed/fr.inri

test2

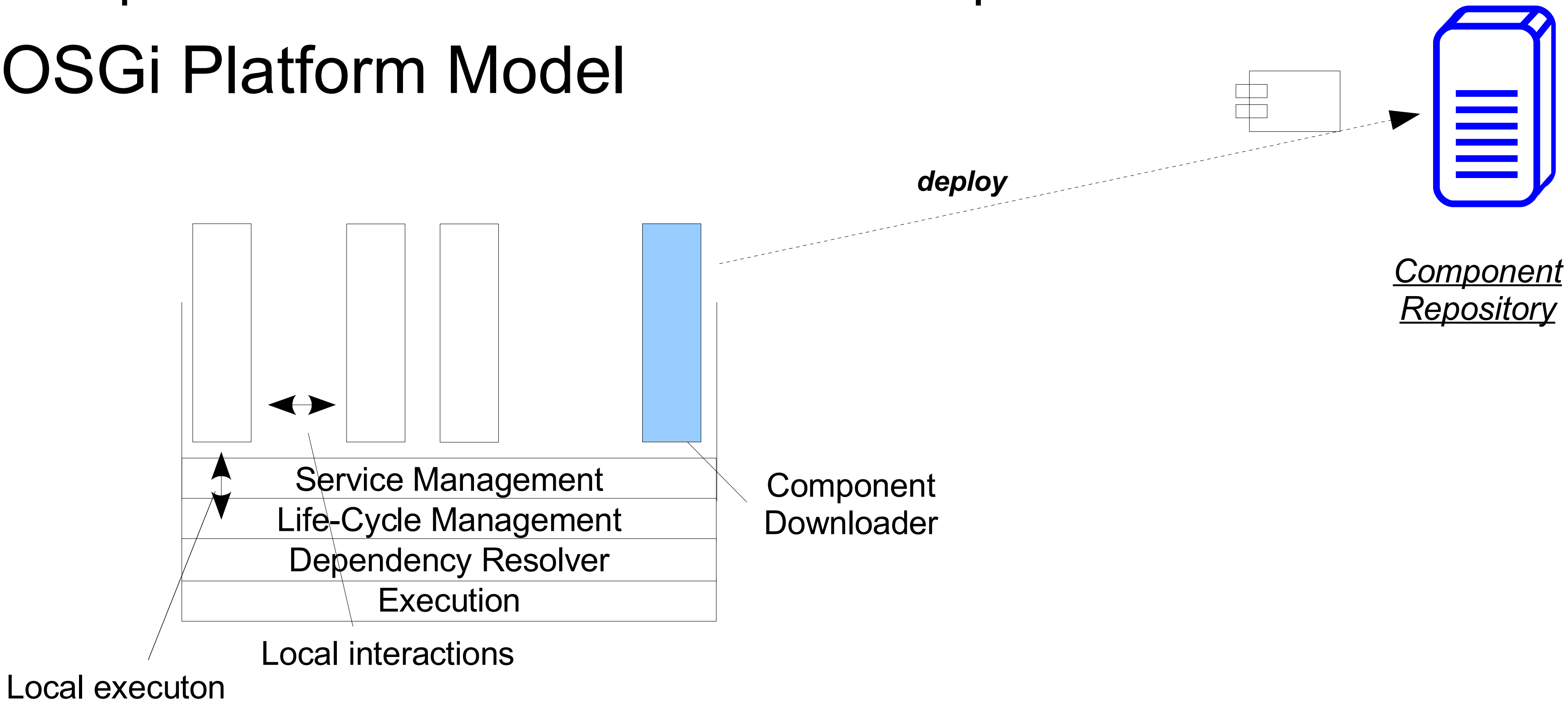
Upload Selected

Published Bundles

-- --

Toward a Hardened OSGi Platform

- Requirements
  - Specification for an hardened OSGi platform
- OSGi Platform Model

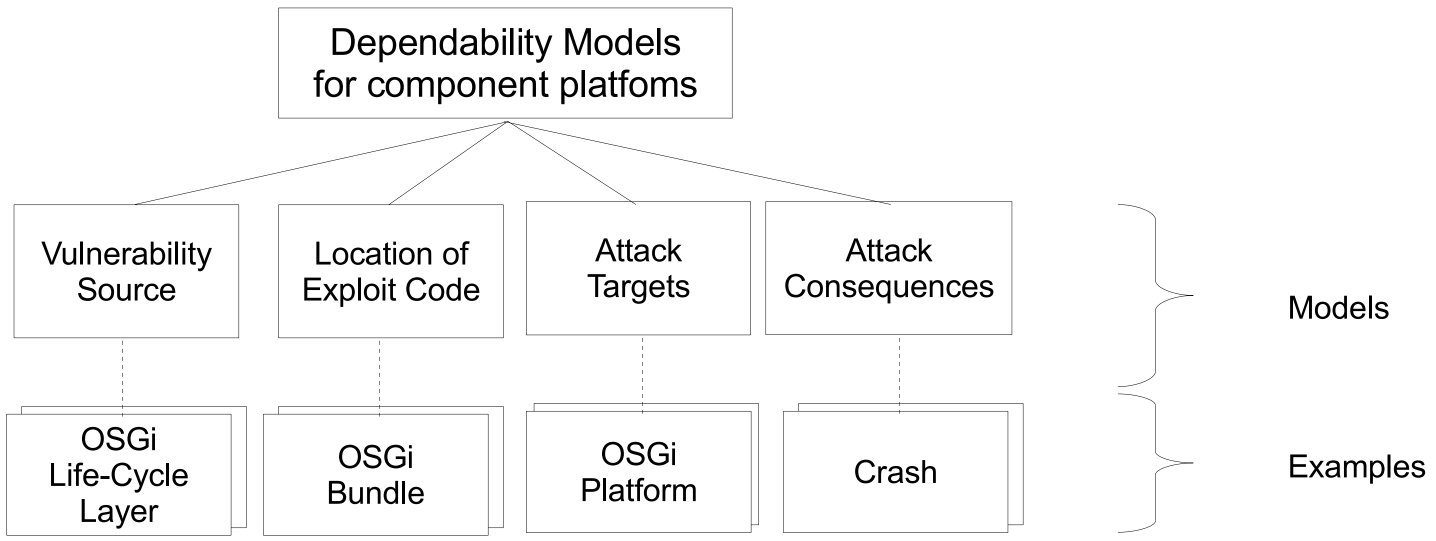


Toward a Hardened OSGi Platform

- **The Semi-formal Vulnerability Pattern for the OSGi Extensible Component Platform**
  - Reference
    - Vulnerability Pattern (VP) Id
    - Taxonomy-based characterization
  - Description
    - More Text
  - Protection
    - Actual Protection
    - Potential ones
  - Implementation
    - Robust and Vulnerable platforms
    - Implementation case coverage

Toward a Hardened OSGi Platform

- **Specific Taxonomies for the OSGi extensible Component Platform**



Toward a Hardened OSGi Platform

- **Building a robust OSGi Platform**
  - Identified Protection Mechanisms
    - Platform hardening
    - Java Permissions
    - Code Analysis
  - Hardened OSGi Platform
    - INRIA Sfelix Project Prototype, V0.2
    - http://sfelix.gforge.inria.fr/
    - 8 vulnerabilities out of 29 patched
    - 13 more are protected with Java Permissions
    - 75 % of vulnerabilities prevented
    - Felix: 48%
    - Equinox: 58%

Toward a Hardened OSGi Platform

- **Recommandations for the OSGi Specifications**
  - Do not rely on the embedded Java Archive verifier
    - OSGi R4, Paragraph 2.3
  - Bundle Resolution Process should be robust
    - Ignore duplicate imports (currently: abort; see R4 par. 3.5.4; Equinox ignores)
    - Handle large manifests without radical performance breakdown
  - Bundle Start Process
    - Start the Bundle Activator in a separate process (R4 par. 4.3.5)
  - OSGi Service Registration
    - Explicit limitation of the number of registered services (R4 par. 5.2.3)
    - Absolute Maximum could be 50 ?

Toward a Hardened OSGi Platform

- **Recommandations for the OSGi Specifications**
  - Bundle Installation process
    - Maximum storage size of bundle archive (for embedded devices) (R4 par. 4.3.3)
    - Should be performed before download when relevant
  - Bundle Uninstallation process
    - Remove Bundle data on the local file system (R4 par. 4.3.8)

Conclusions

- Java Mobile Apps are taking off
  - OWASP is active in the applicative domain too
  - Shift from Security to Dependability focus
- Need of a Life-Cycle long control
  - Security keeps being a management-level question
- OSGi is one solution
  - With so far only reduced implemented security features

Questions ?