

JavaOne 2014

29th September 2014



Building Small and Efficient Internet of Things Applications with Concierge

Jan S. Rellermeyer, IBM Research

Tim Verbelen, iMinds - Ghent University



© 2014 IBM Corporation



Important Disclaimers

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.

WHILST EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

ALL PERFORMANCE DATA INCLUDED IN THIS PRESENTATION HAVE BEEN GATHERED IN A CONTROLLED ENVIRONMENT. YOUR OWN TEST RESULTS MAY VARY BASED ON HARDWARE, SOFTWARE OR INFRASTRUCTURE DIFFERENCES.

ALL DATA INCLUDED IN THIS PRESENTATION ARE MEANT TO BE USED ONLY AS A GUIDE.

IN ADDITION, THE INFORMATION CONTAINED IN THIS PRESENTATION IS BASED ON IBM’S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM, WITHOUT NOTICE.

IBM AND ITS AFFILIATED COMPANIES SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION.

NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, OR SHALL HAVE THE EFFECT OF:

- CREATING ANY WARRANT OR REPRESENTATION FROM IBM, ITS AFFILIATED COMPANIES OR ITS OR THEIR SUPPLIERS AND/OR LICENSORS

Introduction to the speakers: Jan S. Rellermeyer

- 8 years experience in doing systems research

- Recent work focus:
 - Mobile Enterprise Systems
 - Embedded Systems
 - Programming Languages and Runtimes

- My contact information:
 - rellermeyer@us.ibm.com
 - Linked in rellermeyer
 - Twitter @rellermeyer
 - Slideshare.net rellermeyer



Introduction to the speakers: Tim Verbelen

- 5 years experience doing research on mobile cloud computing

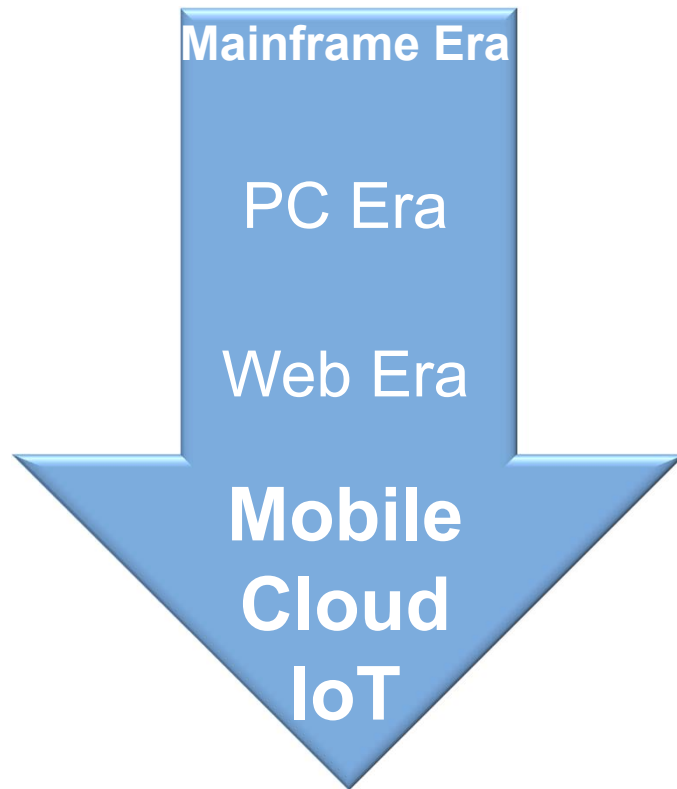
- Recent work focus:
 - Middleware for mobile cloud computing
 - Concierge
 - Internet of Things

- My contact information:
 - tim.verbelen@intec.ugent.be
 - Linked in tim-verbelen



Internet of Things

We've been here before ...



or not?

Embedded Devices are Changing



Image: www.honeywell.com

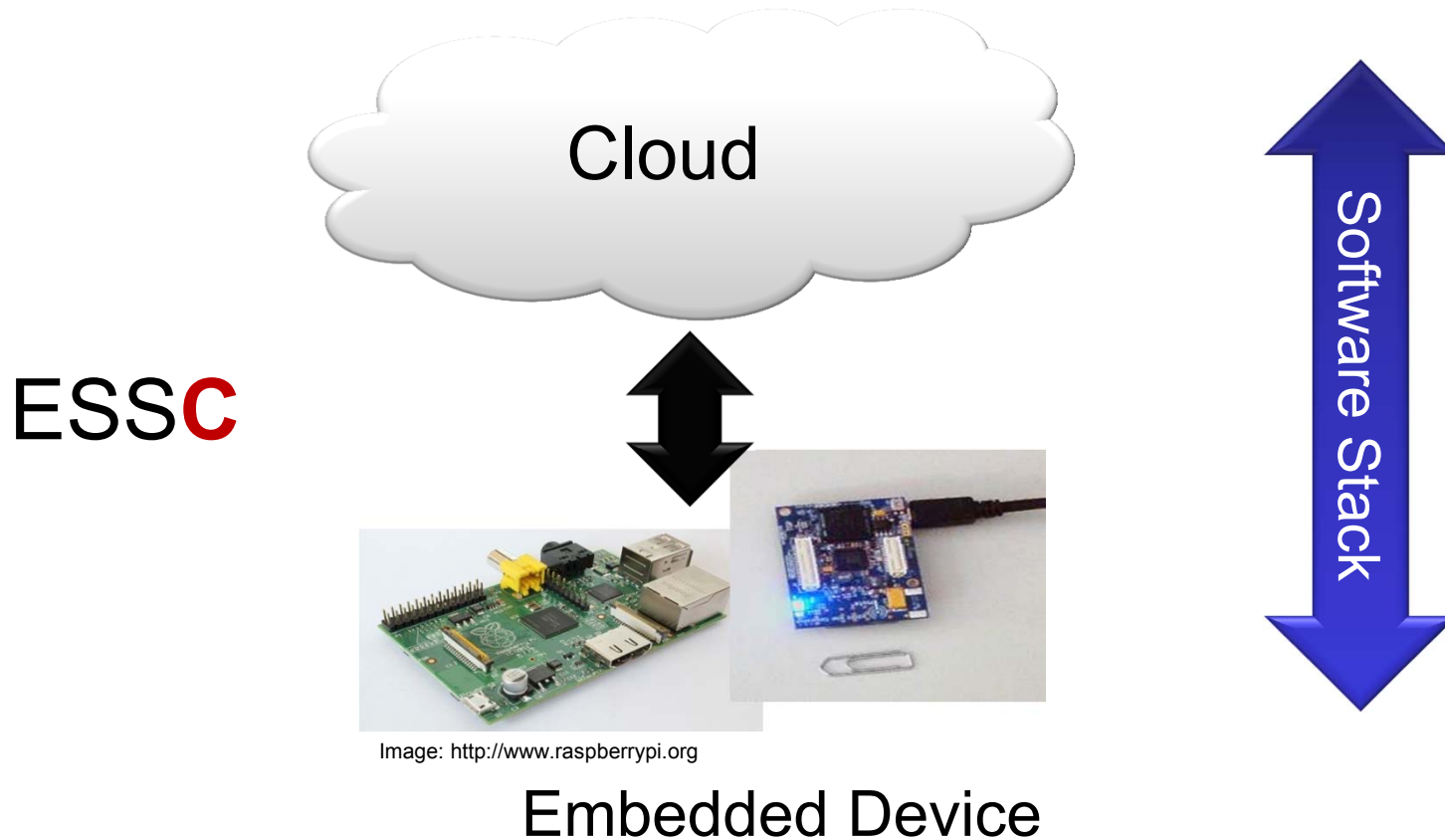


Image: www.apple.com



Image: www.nest.com

Continuous Platform Experience



Embedded and IoT: Why OSGi?

Java can run almost everywhere

From embedded device to the cloud

Dynamic linking, adaptable



Updating and maintaining long-running Java software

really better than “reboot”?

Complexity of large application

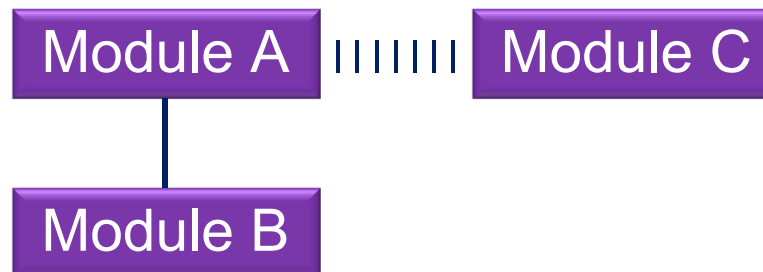
the mystical classpath



OSGi Bundles and Services

- Bundles = modules
- Bundles = JAR files with additional metadata
- Package dependencies = tight coupling
- Services = loose coupling

- Lifecycle!



Example: The BUG Platform

Embedded Linux Device

Runs Java (PhoneME) and (good old) Concierge OSGi

Pluggable hardware
modules

Registers OSGi services

Applications are OSGi
bundles



Image: <http://www.buglabs.com>

Concierge

Question: What is the (minimum) overhead of OSGi?

My answer: Concierge

OSGi R3 core implementation developed during Jan's PhD at ETH Zurich optimized for mobile and embedded devices, has a footprint of only 86kB
 People found the source code readable and understandable: 7 Java classes + 7 inner classes.

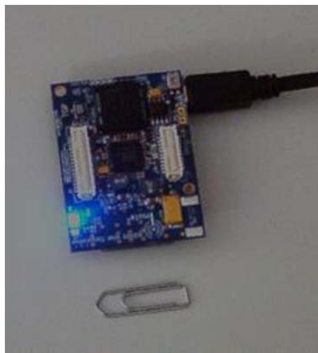


Image: <http://www.linksys.com>



[J.S. Rellermeyer and G. Alonso: Concierge – A Service Platform for Resource-Constrained Devices. In: EuroSys 2007].

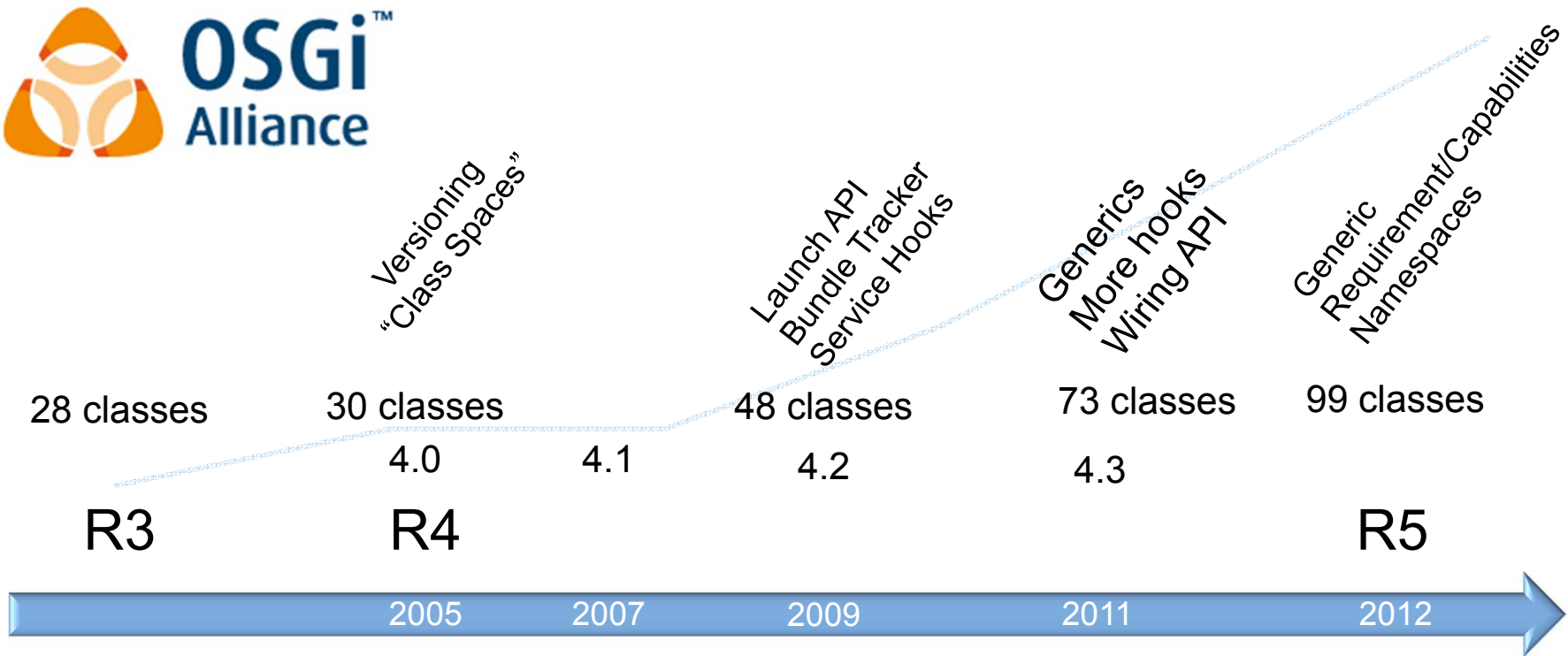
Eclipse Concierge

Goals:

- Full OSGi core R5 compatibility
- Keep a small footprint to work well on embedded devices
- Remain “readable”
- Remain backwards-compatible
 - Java 5
 - Java 1.4?
- Being a sandbox for innovation around OSGi



Challenges



Eclipse Concierge

Goals:

- Full OSGi core R5 compatibility
 - **One test away from it**
- Keep a small footprint to work well on embedded devices
 - **Currently 250kB**
- Remain “readable”
 - **Currently 9 classes**
- Remain backwards-compatible
 - **Java 5**



How does it perform?

Experiments

Beaglebone Rev A5

- AM335x 720MHz ARM Cortex-A8
- 256 MiB DDR2 RAM
- 4 GiB microSD card
- Angstrom Linux
- [Java SE Embedded 1.7.0_21-b11](#) and
- [Java SE Embedded build 1.8.0_06-b23 compact1](#)



Raspberry Pi B

- ARM1176JZF-S 700 MHz ARM 11
- 512 MiB SDRAM (shared with GPU)
- 8 GiB SDHD card
- Raspbian Linux
- [Java SE build build 1.8.0-b132](#)



OSGi Frameworks

Eclipse Equinox

- version 3.9.1.v20140110-161
- 1.4 MiB

Apache Felix

- version 4.4.1
- 527kiB

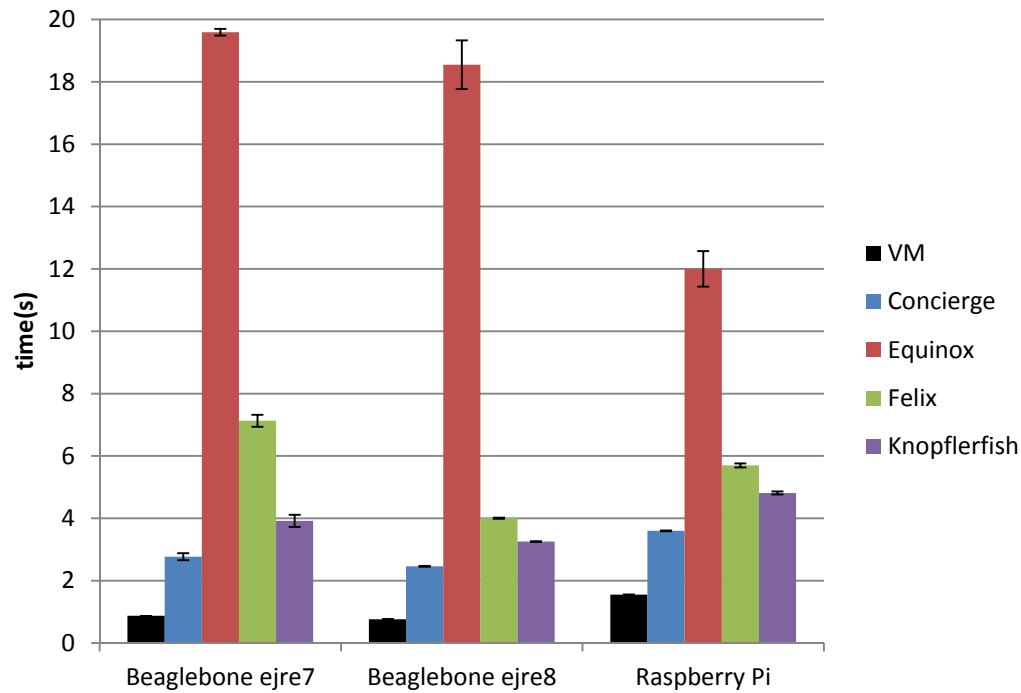
Knopflerfish

- version 7.1.2
- 607kiB / 318kiB compact

Concierge

- aa345b9b92783cecdea7147d222057ea564bdf44
- 245kiB

Startup Time



VM = startup time of the JVM

Concierge is consistently the fastest framework in terms of startup time

Service Registry

Stress test

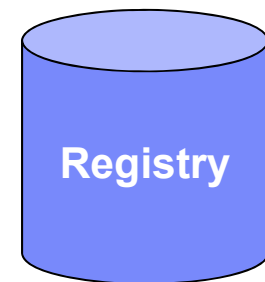
Register 10000 services

- Each one has a random value for the same key
 - Range: byte => collisions

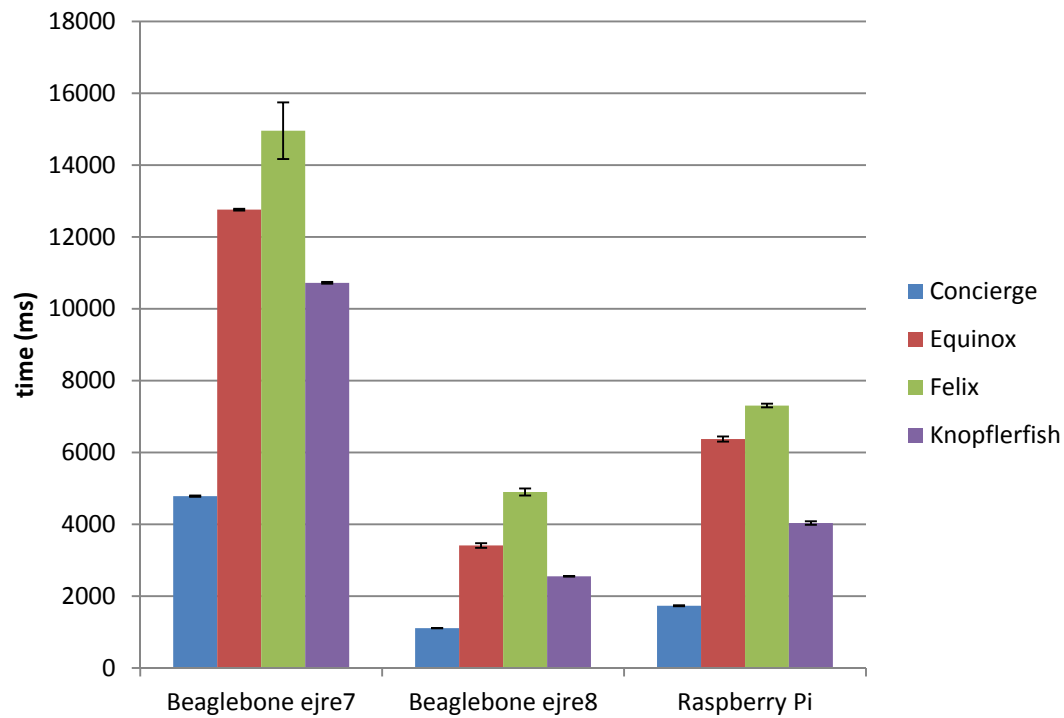
Making 1000 lookups (getServiceReferences) for a random value

- Range: byte

Unregistering the 10000 services



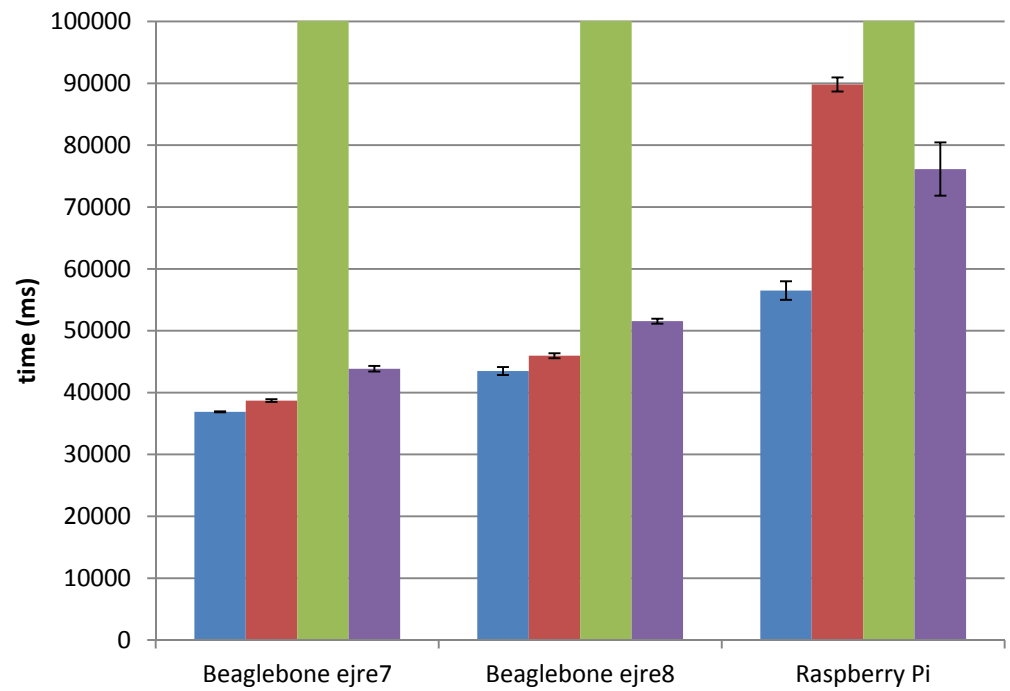
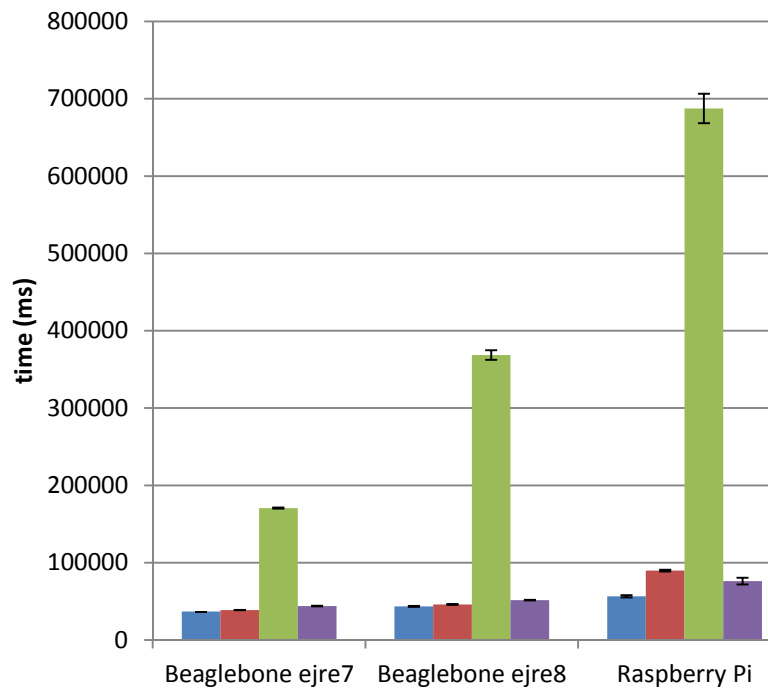
Service Registry – Service Registration



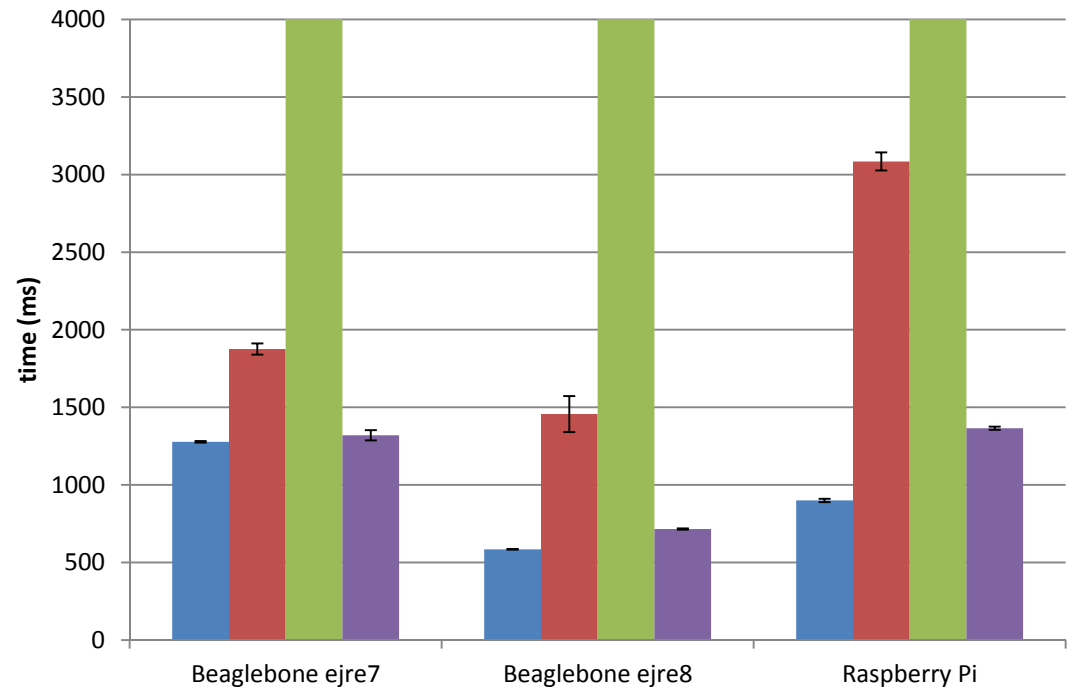
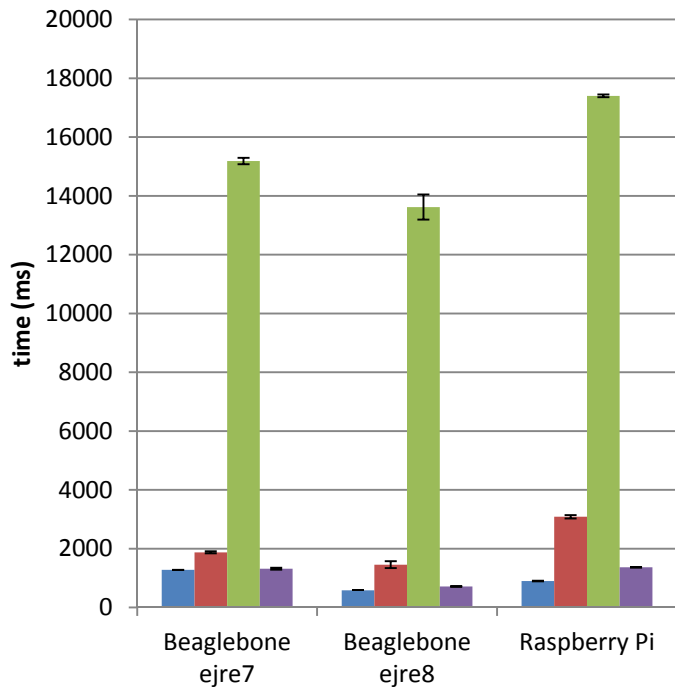
Concierge is consistently the fastest framework

Java 8 helps performance

Service Registry – Service Lookup



Service Registry – Service Unregistration



Resolver

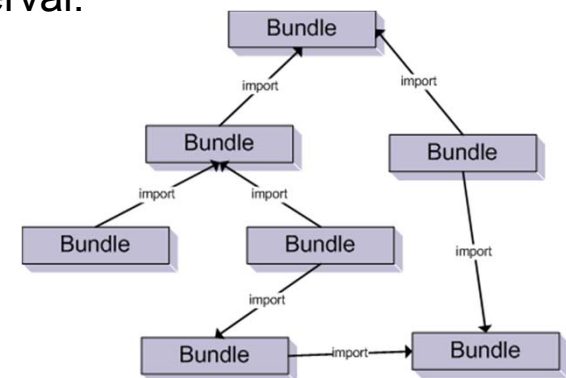
Generate 1000 “random bundles”

- Can either import or export up to 5 packages
- Choice of 50 packages in total
 - For exports: generate a random package version from [1.0.0-21.0.0)
 - For import: generate a random import range from the interval.

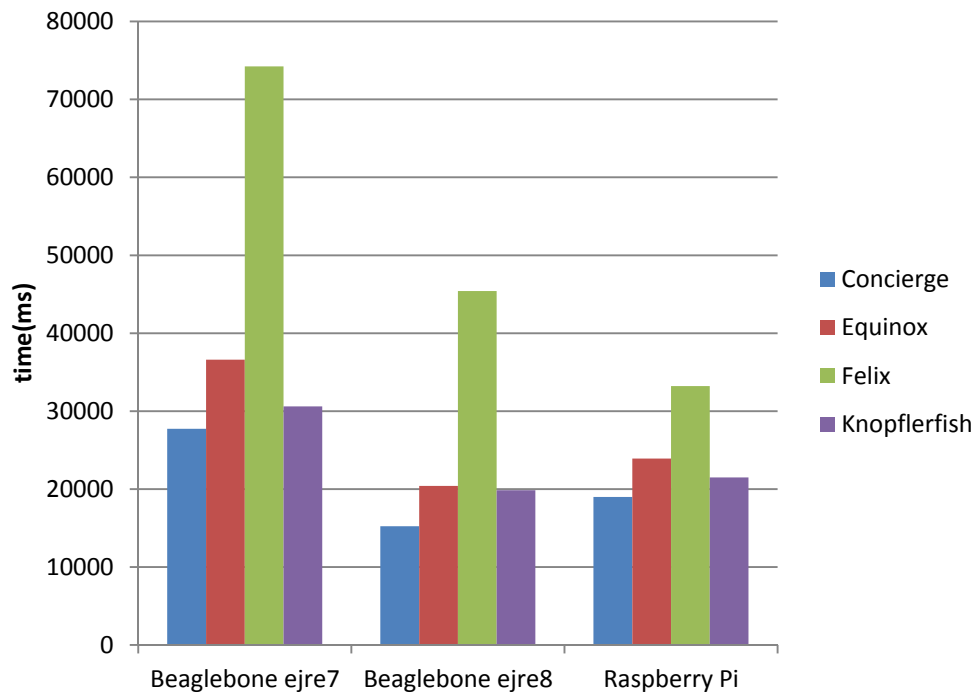
Install all 1000 bundles

Resolve the bundles

Benchmark turned out to be rather volatile...



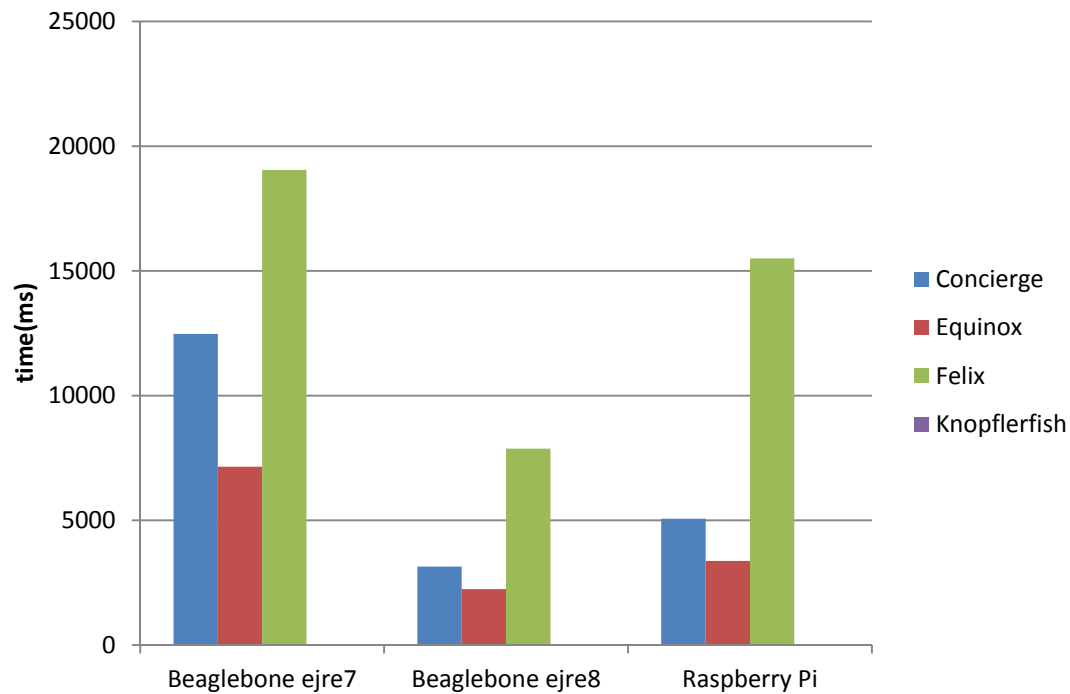
Resolver - Install



Performance is good

Operation tends to be I/O-bound

Resolver - Resolve



Equinox is optimized for resolving a large number of bundles

We need to optimize more

Could not get Knopflerfish to resolve

Wearable Devices

We all carry “embedded” devices



Image: www.samsung.com



Image: www.catwig.com



image: www.samsung.com

OSGi for mobile cloud computing

Suppose the following application:

viewer



logic

sensors

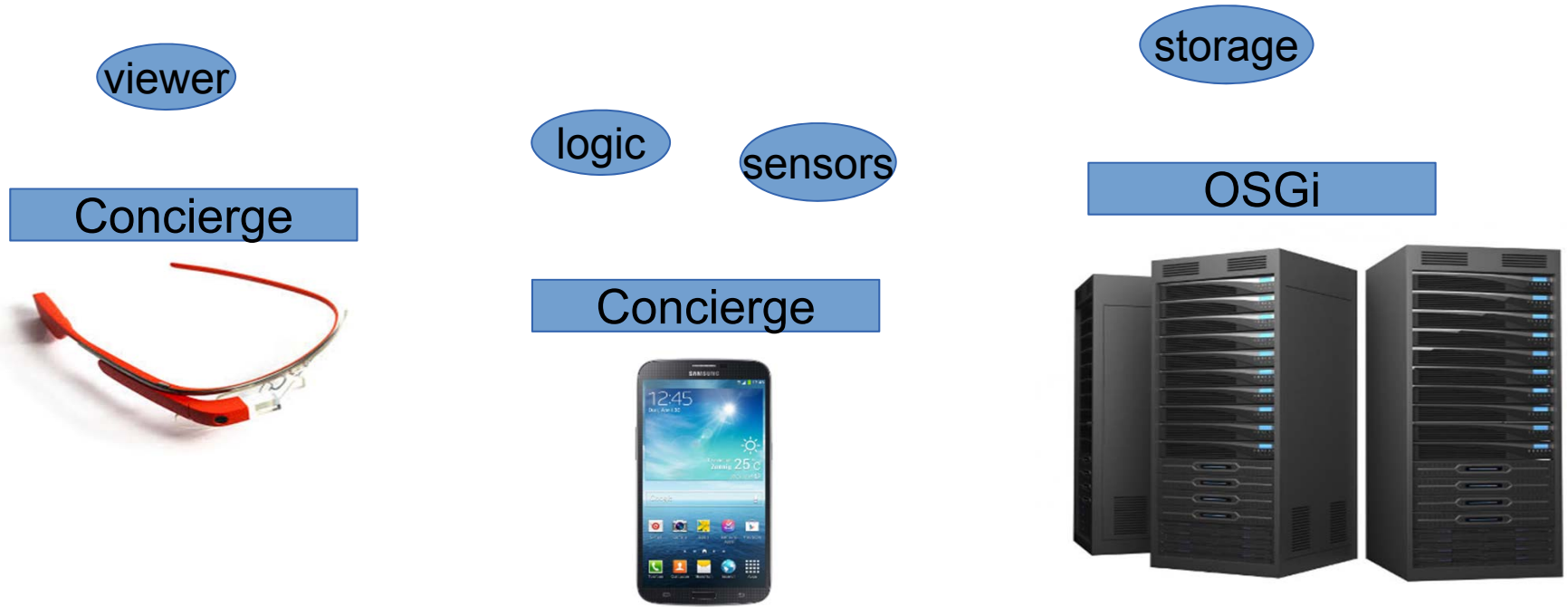


storage



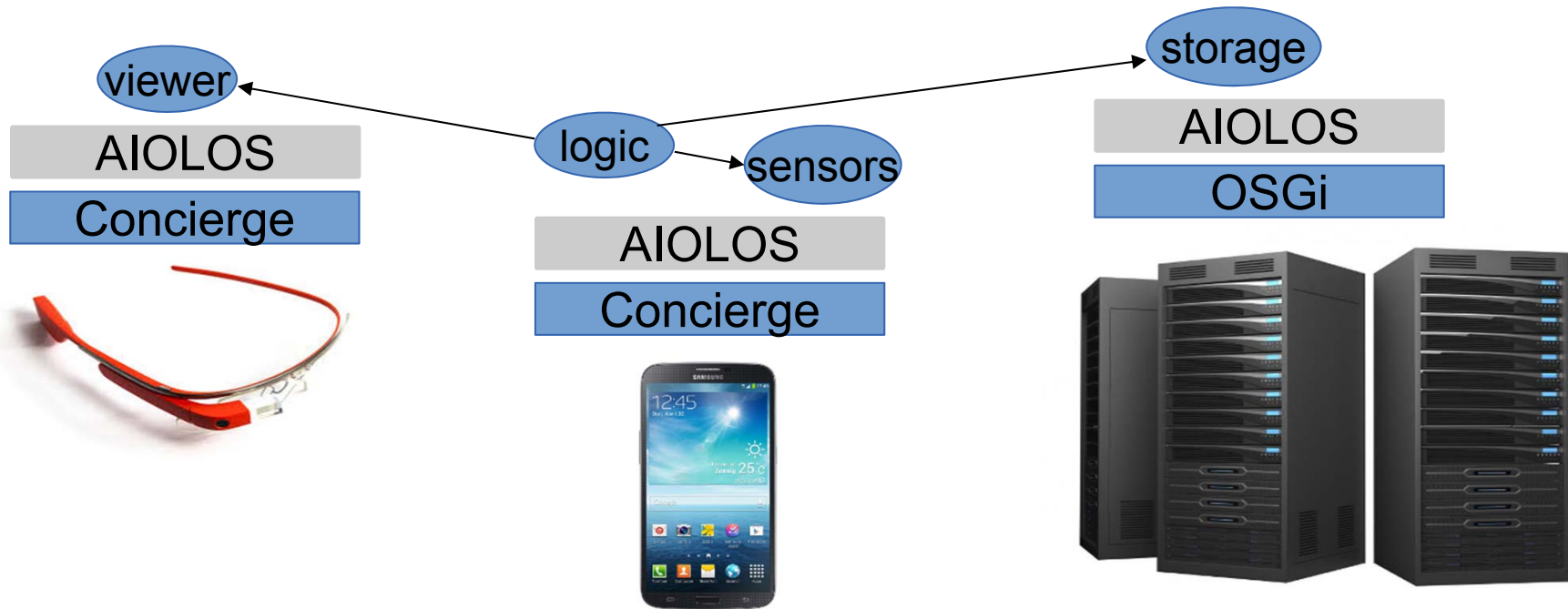
OSGi for mobile cloud computing (2)

We need software modularity!



OSGi for mobile cloud computing (3)

We need transparent remote communication!



AIOLOS framework : bundle distribution at runtime



<http://aiolos.intec.ugent.be>

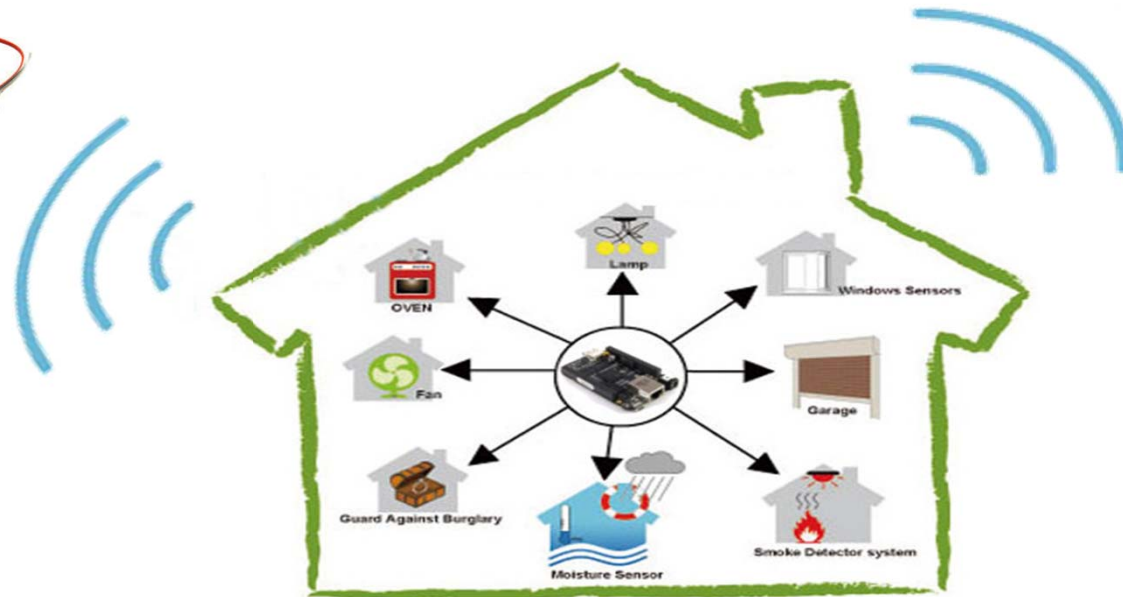
AIOLOS provides:

- Service discovery and binding
- Fast remote service calls
- Runtime (re)deployment
- Service monitoring

[Tim Verbelen et al.: AIOLOS: middleware for improving mobile application performance through cyber foraging. In: Journal of Systems and Software 85 (11)].

Demo

What's next? IoT



Eclipse IoT – towards an IoT gateway on top of Concierge



- provides open source implementations for the most common services needed by M2M applications (e.g. Bluetooth, serial, ...)



- a flexible framework for smart home and ambient assisted living (AAL) solutions (i.e. data handling, rule engines, ...)



- MQTT client/server implementations



- A CoAP protocol implementation

Eclipse Concierge

Concierge is an OSGi framework optimized for embedded devices and the Internet of Things

We are in the process of transitioning several Eclipse IoT projects to Concierge

It's fast, it's pleasant, you should try it, too.



Project Homepage

<http://projects.eclipse.org/projects/rt.concierge>

Clone it from here



<http://git.eclipse.org/c/concierge/org.eclipse.concierge.git>

developerWorks®

www.ibm.com/developer



Discover

new technical resources.

Develop

your coding skills.

Connect

with developers.



Big data & analytics



Cloud computing



DevOps



Mobile development



Security

ibm.biz/javaone2014

Visit **Booth 5511** to learn about Cloud, DevOps and Mobile solutions

Copyright and Trademarks

© IBM Corporation 2014. All Rights Reserved.

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., and registered in many jurisdictions worldwide.

Other product and service names might be trademarks of IBM or other companies.

A current list of IBM trademarks is available on the Web – see the IBM “Copyright and trademark information” page at URL: www.ibm.com/legal/copytrade.shtml