Java. Cloud. Leadership.

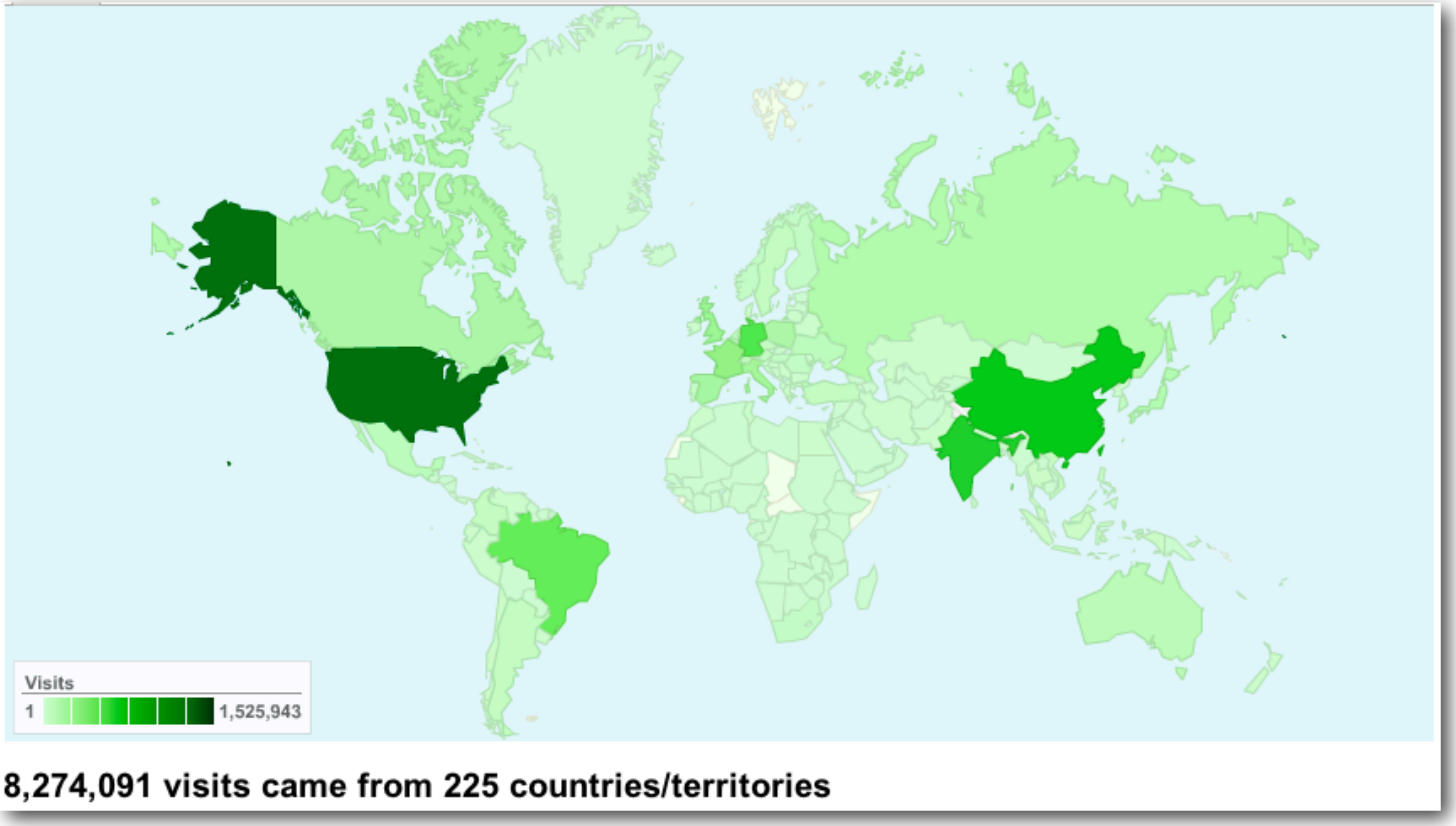
2012 - Year of the Developers

Bruno Georges
Director, Software Engineering
JBoss by Red Hat

Overview

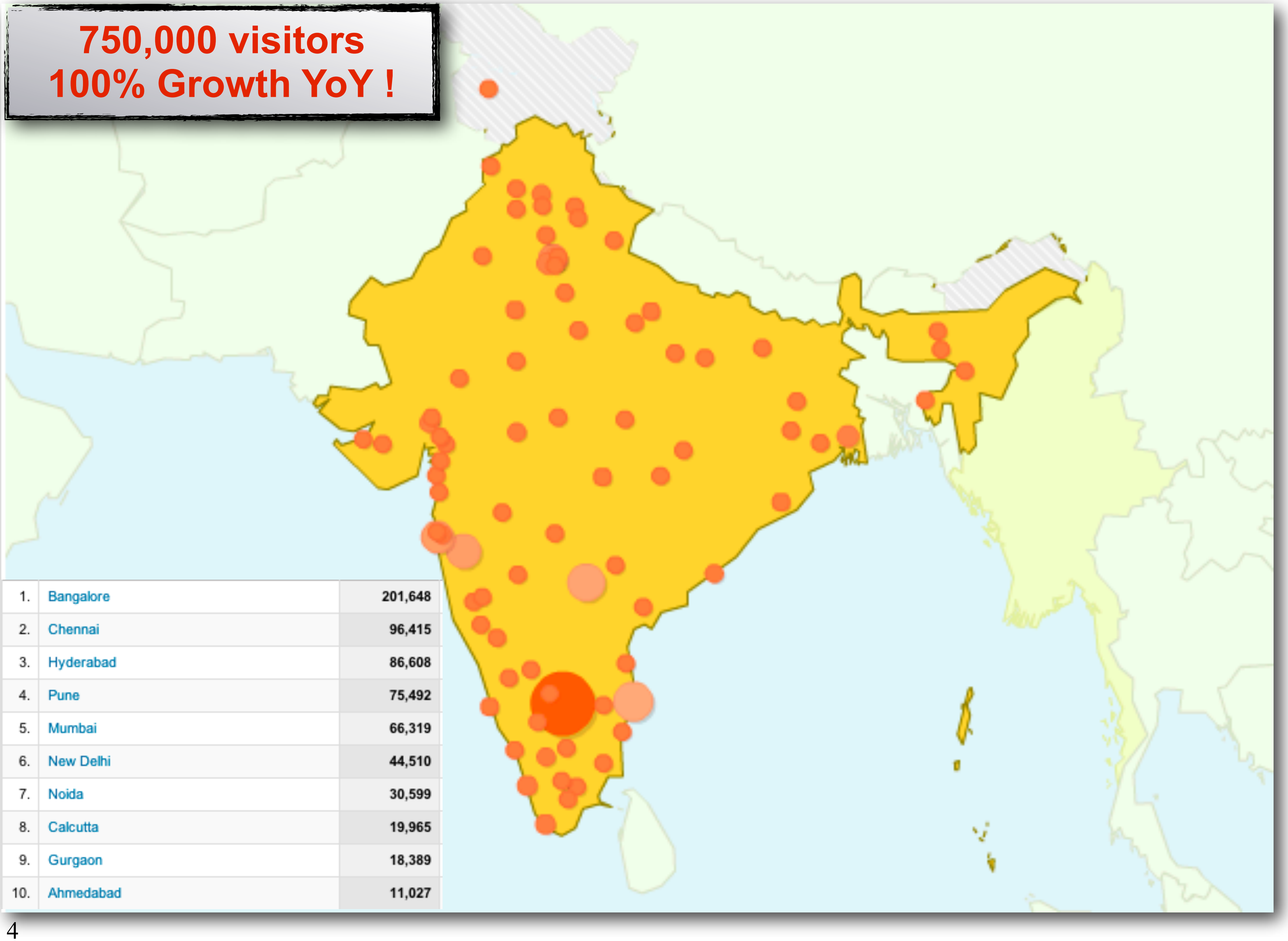
- JBoss Projects & Developer Community
- Where are we today and why?
- Mobile and Cloud
    - Ubiquitous computing in the large
    - The real cloud!
- Hardware and software forces in action
- What does this mean for today’s middleware offerings?
- The future of Java and standards

JBoss.org visitors in 2011 - Worldwide



8,274,091 visits came from 225 countries/territories

JBoss.org visitors in 2011 - India



750,000 visitors
100% Growth YoY !

1.	Bangalore	201,648
2.	Chennai	96,415
3.	Hyderabad	86,608
4.	Pune	75,492
5.	Mumbai	66,319
6.	New Delhi	44,510
7.	Noida	30,599
8.	Calcutta	19,965
9.	Gurgaon	18,389
10.	Ahmedabad	11,027

arquillian

HornetQ

80 JBoss Projects ! 100,000 Community Members!

RESTEasy

Infinispan

RichFaces

as 7

HIBERNATE

Ceylon

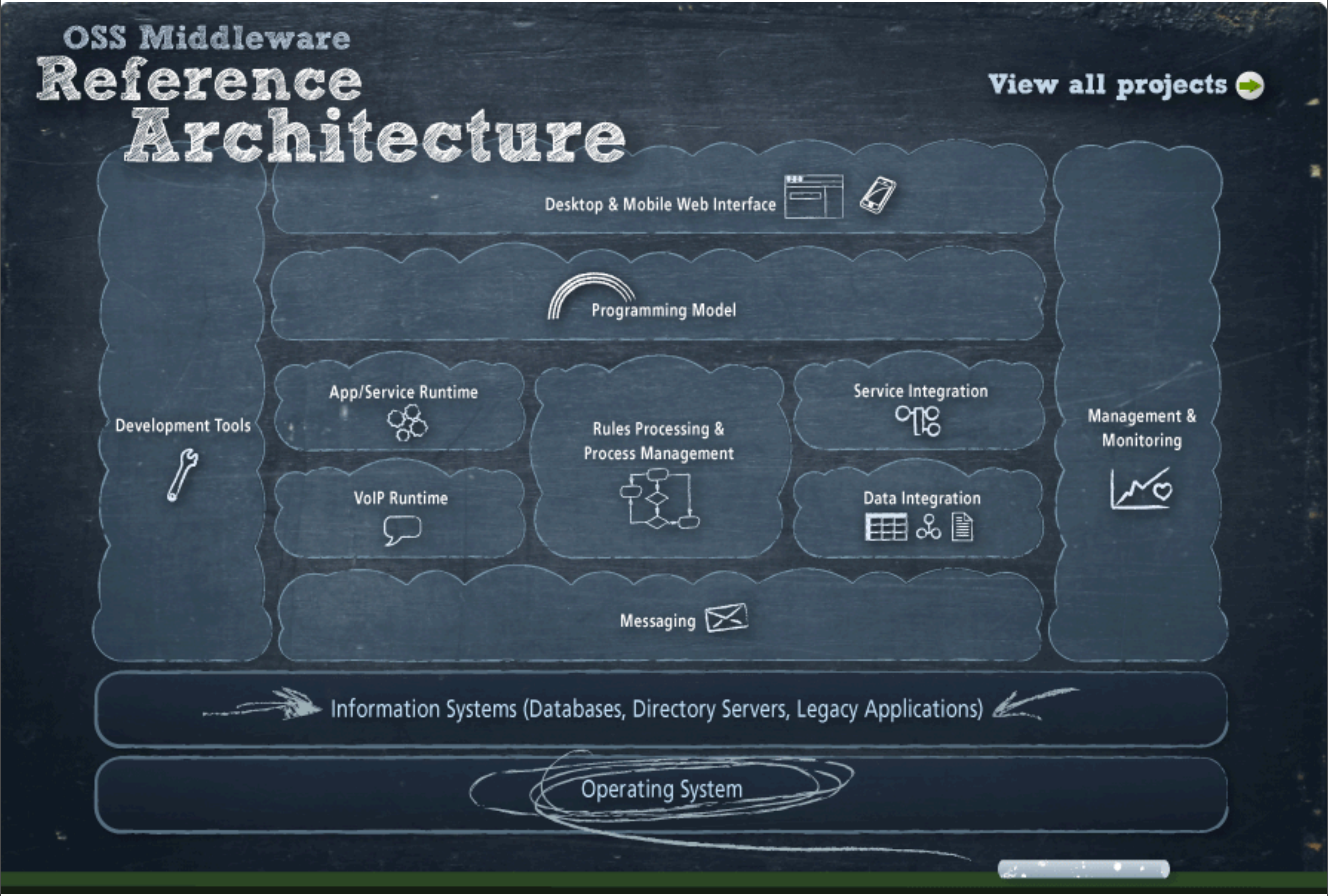
Weld

SWITCHYARD

- **62,600** active wiki, forum and blog users
- **9,680** active issue tracker users

IRON JACAMAR

JBoss Drools



OSS Middleware Reference Architecture

View all projects

Desktop & Mobile Web Interface

Programming Model

Development Tools

App/Service Runtime

VoIP Runtime

Rules Processing & Process Management

Service Integration

Data Integration

Management & Monitoring

Messaging

Information Systems (Databases, Directory Servers, Legacy Applications)

Operating System

Today …

- •512Meg memory is standard on smart phones, 64Gig storage
- •256Gig USB sticks are becoming the norm
- •100Gig ethernet at work and 30mbps to the home
- •64 bit quad core processors in laptops, 1GHz ARM in iPhone
- •WiFi throughout many cities

The times have changed

- There are already more mobile devices than computers
- There are 4x more processors on the planet than people
  - Most have TCP stacks
    - dsPIC33FJ12GP 16-bit microcontroller has as much horsepower as a VAX (40MIPs), can handle 16+ sensors, and is 1/8 the size of a penny
  - 30 million iPads already
    - 1 in 2 Americans predicated to have smart phones by the end of 2011 compared to 1 in 10 in 2008

Past, present and future

- The laptop concept originally devised for children
  - Now laptops replace desktop
- Java and Linux helped to popularise middleware
- Java still leading, but not as cool any more
  - Ceylon, Ruby, Scala, Erlang, JavaScript, …
- Coolness is iPhones, Androids, POH5, …
- Those are the new frontiers of application development

“Little’s law” (thanks to Parkinson)

- “Work expands to use the power available”
  - Basic word processors on PCs
    - Publisher-quality implementations now on laptops
  - Games pushing the envelope from Pong through Space Invaders to CoD
    - MVCC
    - Distributed systems
    - Grids
  - Mobile devices contain more and more personal data
    - Wallets via NFC
  - Disconnected operation is the normal situation

Application complexity

- Types of applications increasing in complexity
  - Online purchases
  - Distributed peer-to-peer interactions
- Enterprise requirements becoming a necessity
  - Security and identity
  - High performance, low latency, reliable messaging
  - Database updates with transactions
  - Workflows as inter-app interactions increase

Cloud meets mobile

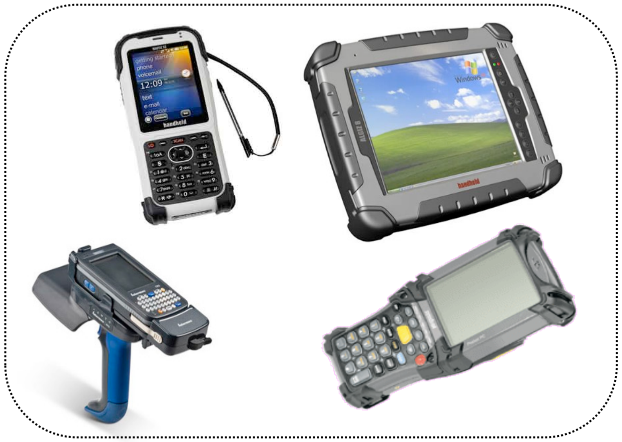
- Public Clouds important
- Private Clouds probably more important
  - Security and data consistency implications
- But Ubiquitous Computing has become a reality
- Mobility and embedded devices are the real Cloud
  - Thin clients aren’t the future
    - Shannon’s Limit

Ubiquitous computing

Smartphones and Tablets

- Smartphones and tablet shipments overtaking PCs
- Multi-device support the reality

Field Service / Warehouse Devices

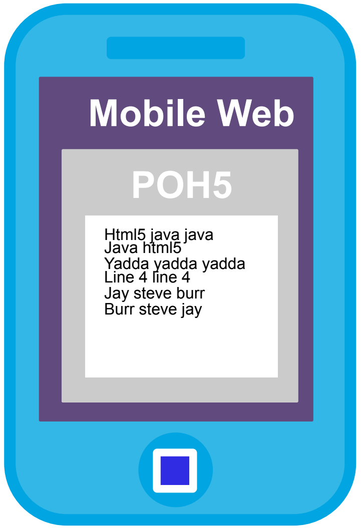


- Primarily Windows based
- Application tied to device and OS

Embedded / Unattended

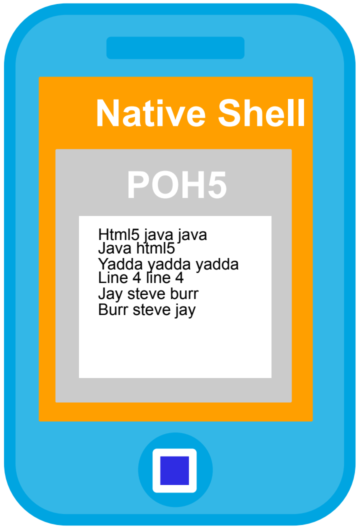
- New use cases for embedded processors / computing

Several types of mobile application



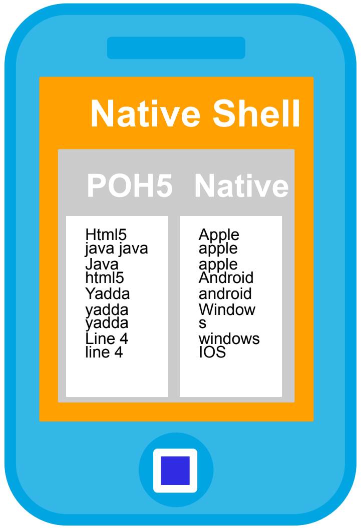
Web Apps

Written in HTML5, JavaScript and CSS3. Quick to develop with traditional web skills but less use of device capabilities.



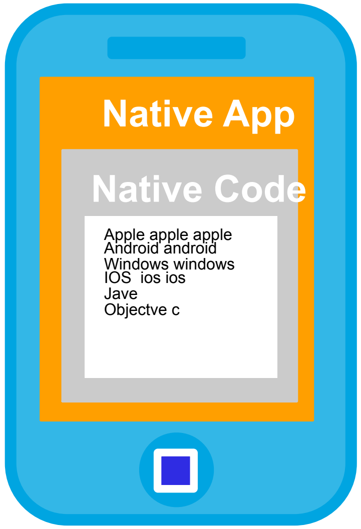
Hybrid Apps

HTML5 wrapped in native shell. Provisioned and accessed as a native app. Leverages web skills and HTML5 with appearance of native app.



Hybrid Apps +

Hybrid app tweaked with native code to access device capabilities.



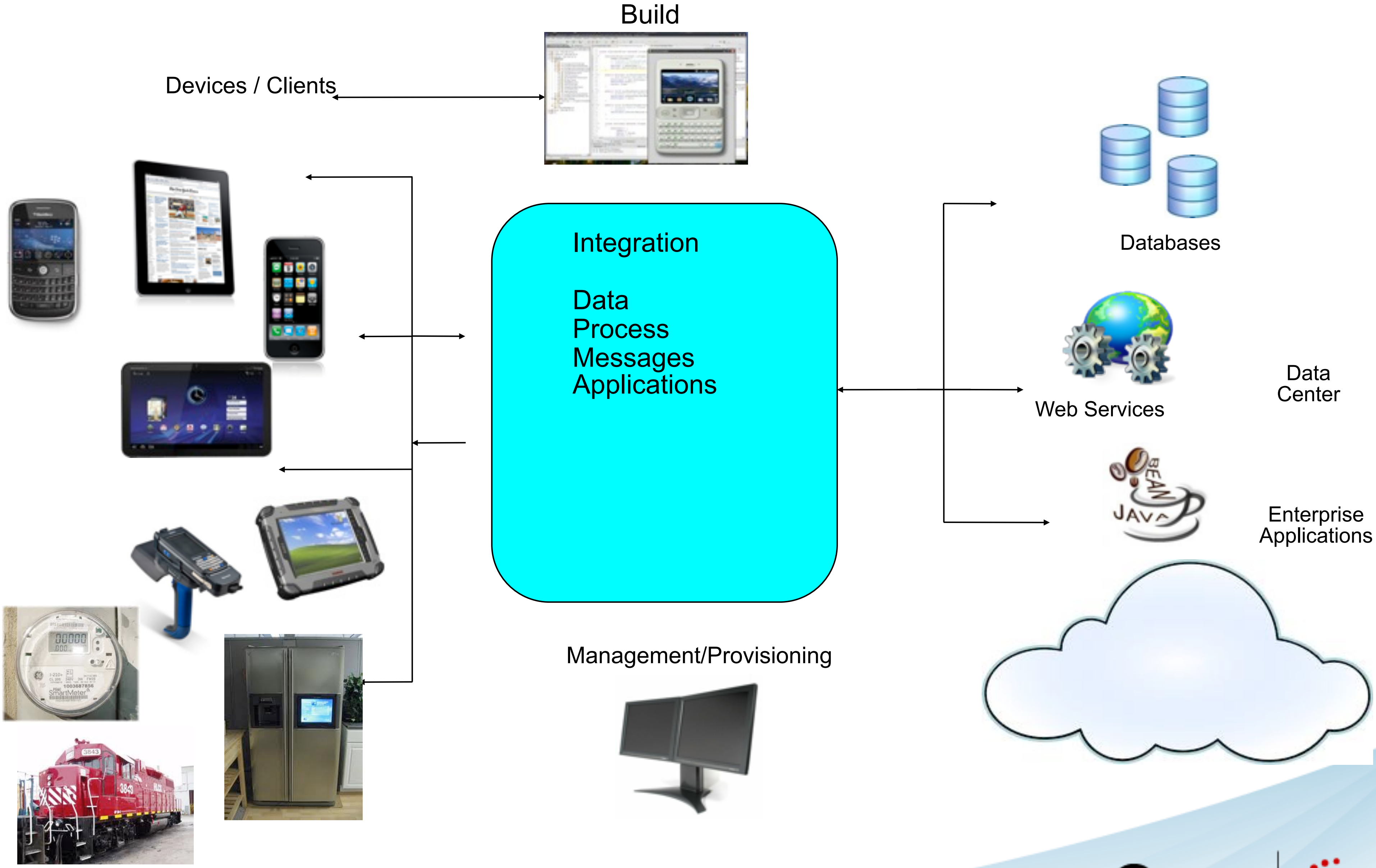
Native App

Platform-specific written with OS specific SDK to deliver fullest capabilities.

“New age” development

- New architectures
- New implementations
- New frameworks
- New operating systems
- New new new ..?

“Mobility” and the Enterprise Customer



Build

Devices / Clients

Integration

Data
Process
Messages
Applications

Databases

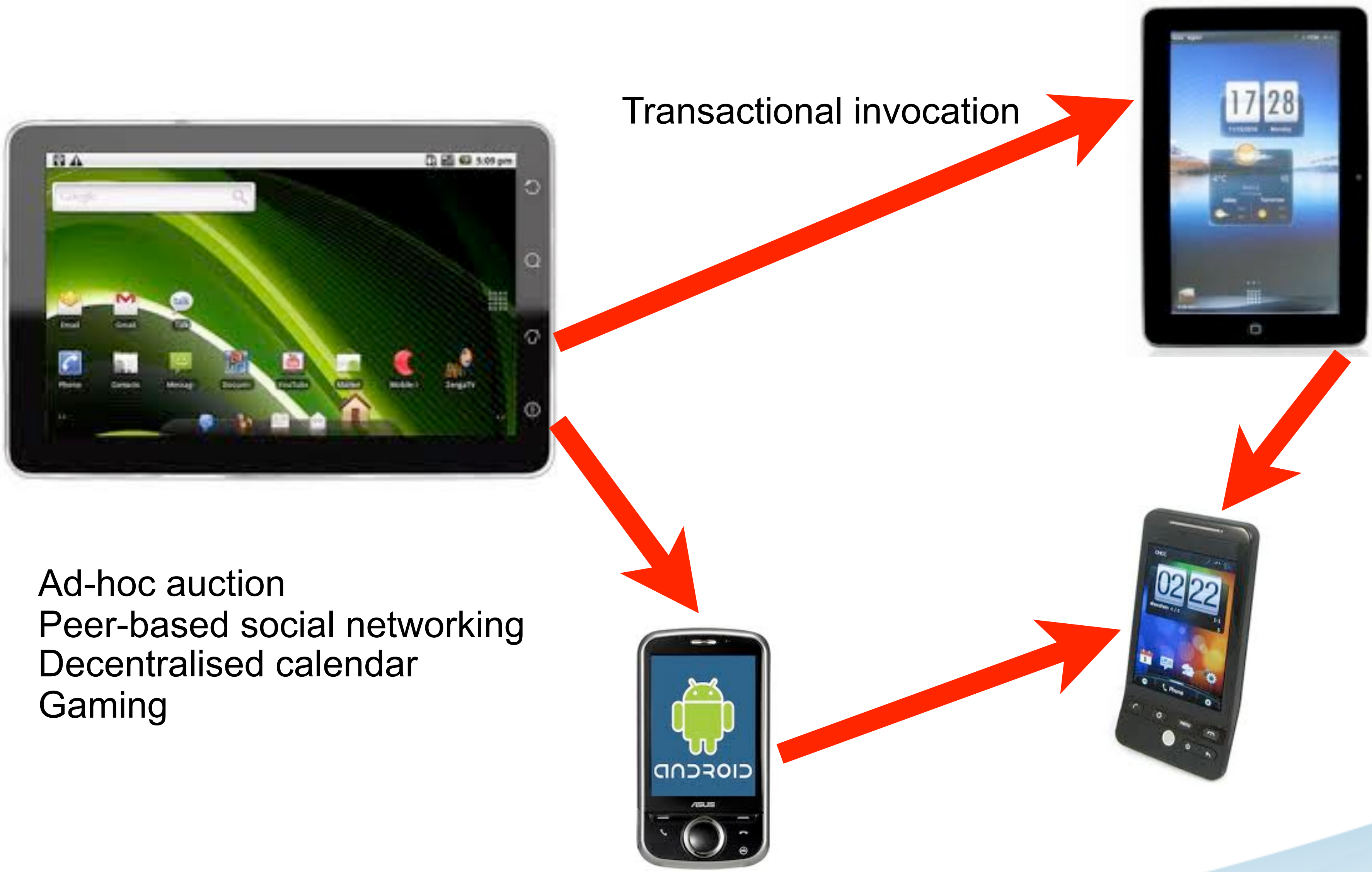
Web Services

Data
Center

Enterprise
Applications

Management/Provisioning

Mobility and “enterprise” apps



Transactional invocation

Ad-hoc auction
Peer-based social networking
Decentralised calendar
Gaming

For whom the bell tolls? Middleware?

- Hold on ... haven’t we heard this death knell before?

Remember Web Services/REST/<blank> as death of middleware?

- Not if you are objective!

Middleware over the years

- RPC first used in the 1970’s to link Unix systems
- Bespoke implementations through the 80’s
  - Argus, Emerald, Arjuna, Camelot/Avalon, ISIS
- Standards began to evolve
  - ANSA
  - DCE
- More cross-vendor industrial standards
  - CORBA
  - J(2)EE
  - Web Services

40 years of middleware shows ...

- Many things common
  - Security
  - Messaging
  - Transactions
  - Replication/Cacheing
  - Data store (e.g., database)
  - Distribution
  - Multi-tenancy (multi-threaded/multi-application)
- The industry has spent 40+ years designing enterprise infrastructures

So what does this mean?

- Middleware is needed whatever the deployment environment
  - Mainframes, servers, laptops etc.
- Don’t tie the definition of middleware to an implementation
- Mobile and Cloud should not be new silos for developers!
- Enterprise requirements transcend deployment realities

So are cloud/mobile the death of middleware?

- Many commonalities with “traditional” middleware
  - Enterprise requirements for all but trivial apps
- Obviously Java is not the only application language
  - But why not a common runtime?
  - Lots of popular JVM-based languages
- Lack of cloud/mobile standards

Present and future directions

- As an industry we must build on our existing implementations
- We must provide a natural upgrade path for existing users
  - We cannot afford to repeat the DCE/CORBA, DCOM/.NET or CORBA/J(2)EE days
- Evolution rather than revolution
- Why?

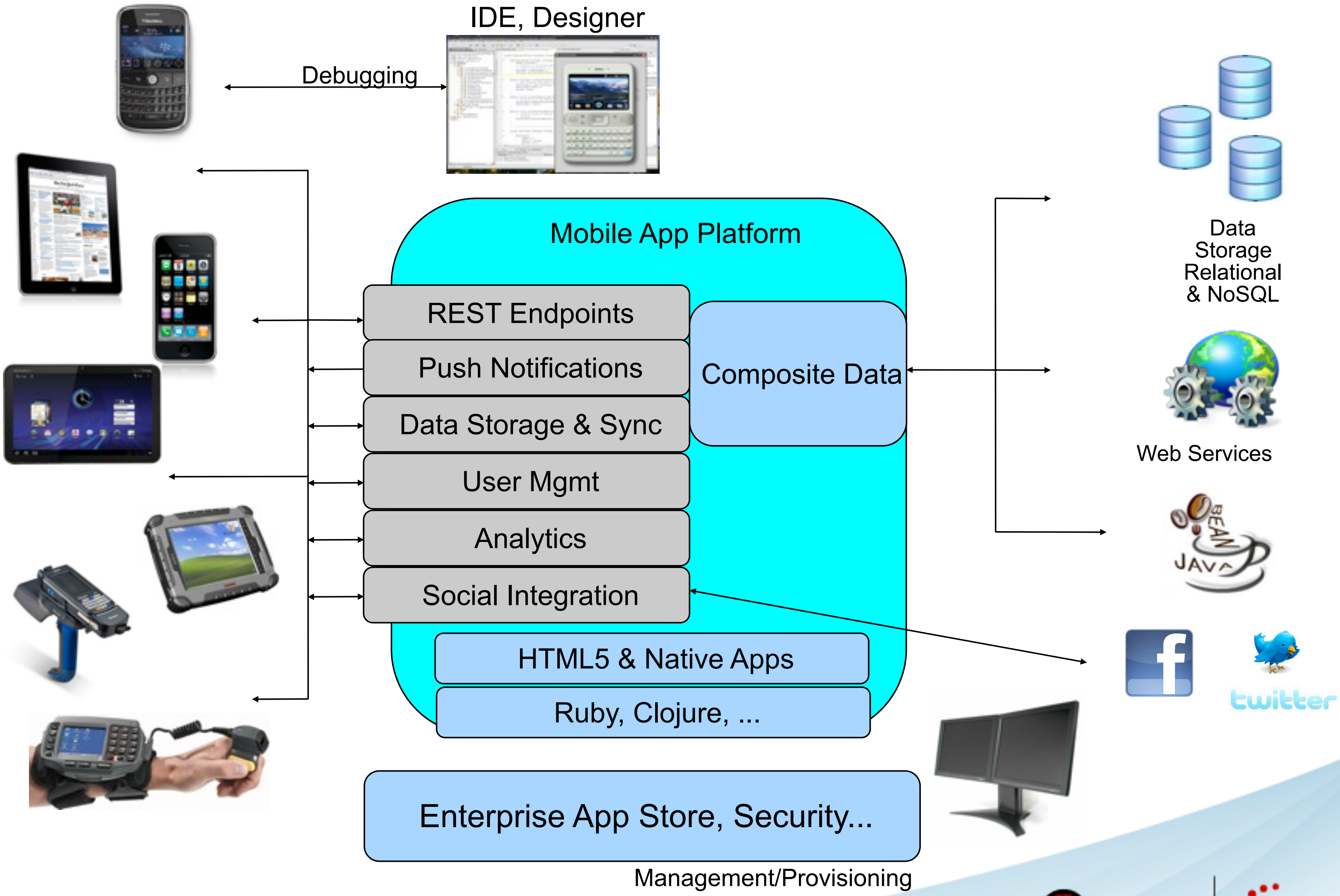
Some ubiquitous computing realities

- Trust is important!
- Trust is measured in:
    - Who is providing the service?
        - And are they doing it in a way that matches my requirements?
    - Are they living up to my required QoS
        - Fault tolerance, performance, etc.
- Several well publicised Cloud outages and intrusions
    - Mobile viruses, identity theft ...
- Still very early in the adoption cycle

Java Enterprise Edition 6

- Turns out that EE6 has many of the required capabilities
  - Standards based too!
- EE6 represents a great evolution for 40 years of work!
- New capabilities (e.g., JAX-RS, CDI, BeanValidation)
- Input from wider open source communities and users
- Profiles

Mobile Computing Architecture



IDE, Designer

Debugging

Mobile App Platform

REST Endpoints

Push Notifications

Data Storage & Sync

User Mgmt

Analytics

Social Integration

Composite Data

HTML5 & Native Apps

Ruby, Clojure, ...

Enterprise App Store, Security...

Management/Provisioning

Data
Storage
Relational
& NoSQL

Web Services

“Java EE is too bloated ?”

- Differentiate the standard from implementations!
  - Let’s not live in the past
- It is possible to be lightweight and enterprise ready

The Open Source Java application server *reignited*

Designed for flexibility.
Amped with electrifying speed.
Launch your Java EE applications in a flash!

Lightning Fast... *start-up / deployment / configuration*

Standards

- JCP reinvigorated
- Java ME
  - Merging of Java ME EC with SE/EE
  - ME-next?
- Java PaaS
  - Various non-standard solutions
  - OSGi, OMG, JCP
- Open source helps drive experience
  - Don’t standardise too early!

But there are still open areas

- It’s not all doom and gloom
- But it’s not all perfection either
- Several key issues remain
  - Reliability
  - Development models
  - Data

Trustworthy systems

- Can we build a system that …
    - Only services authorised users
        - Service cannot be denied
        - Information cannot be stolen
    - Is always available
        - Out less than 1 second per 100 years
        - 1950‘s **90%** availability
        - 2010 **99.99%** availability for well managed sites (50 minutes/year)
- Software reliability is not keeping pace with hardware
    - Getting more expensive and less reliable
    - Solution so far:
        - Write fewer lines in High level languages
        - Still not improving fast enough

Problems with development models

- We have yet to make parallel programming easy
  - Only automatic parallelism has “won”
  - Managing computer clusters is a major cost
- New computer architectures are highly parallel
  - Many-core chips are a reality
    - Core’s per chip will only increase
  - But we don’t know how to utilise them efficiently
- So, the scale-up problem is not solved

The data problem

- There is a movement away from traditional databases
- The bandwidth problem still remains
  - Economic necessity mandates putting the data near the application
  - The cost of wide-area networking has fallen more slowly than all other IT hardware costs
  - But how can data be in multiple places at the same time?
- Flavours if “SQL”
  - OldSQL
  - Not Only SQL (aka No SQL)
  - NewSQL

Conclusions

- Cloud and mobile will evolve
- Enterprise middleware applications aren’t going away
  - The industry cannot afford to track multiple platforms
  - Middleware components should be available to all
- The next decade will be defined by ubiquitous computing
- There are still areas that need to be addressed
  - NoSQL
  - Multi-tenancy
  - Transactions

2012 - Year of the Developers

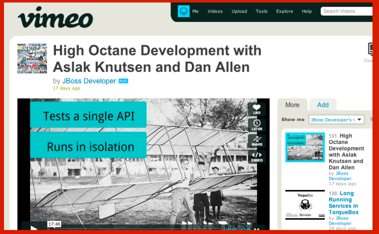
THANK YOU.

Want to keep In touch?

JBoss Developer

on Vimeo

vimeo.com/jbossdeveloper



High Octane Development with Aslak Knutsen and Dan Allen

by JBoss Developer PLUS
17 days ago

131. High Octane Development with Aslak Knutsen and Dan Allen
by JBoss Developer
17 days ago

130. Long Running Services in TorqueBox
by JBoss Developer
29 days ago

JBoss

on Facebook

facebook.com/JBossApac
facebook.com/JBoss

JBoss by Red Hat
Software

Wall — JBoss by Red Hat · E

Pierre Fricke
Got a SOA? You're Almost to the Cloud.

Integrate cloud apps and services into your enterprise.

JBoss

on LinkedIn

linkedin.com/company/jboss

JBossNews
@JBossNews Everywhere
News about JBoss middleware and other things related to JBoss
http://www.jboss.com

About @JBossNews
1,441 Tweets · 2,128 Following · 6,612 Followers

Recent images · view all

You and @JBossNews
You follow accounts that follow @JBos

Similar to @JBossNews · view
mreinhold Mark Reinhold · Fo
Java, OpenJDK, etc.
RedHatSummit RedHatSumm
The 2011 Red Hat Summit and
jodastephen Stephen Colebo
Java, Fantom, Joda-Time, JSR-

Following · view all

JBossNews JBossNews
Quick poll: What problems can Complex Event Processing solve in your organization?
14 Jan

JBossNews JBossNews
#WebLogic to #JBoss Enterprise Application Platform Migration Guide: red.ht/nyUzp6
14 Jan

JBossNews JBossNews
Whitepaper: Expanding Active Decision-Making: The Power of Integrating Business Rules and Events: red.ht/utrZrH #BRMS #jboss

@JBossNews

on Twitter

twitter.com/jbossnews
twitter.com/jbossdeveloper