nanoSOA™ 3.0 Beta

“Where all bytecodes are Web Services”™
Overview

- WS-VM
- WS-DI
- WS-Security
- WS-SPOT
- WS-3D
- WS-RFID
- WS-POJO
- ...and more
WS-VM

- First virtual machine to execute Web Services natively
- RISC - “GET” is the only instruction
- No “noop” instruction as it as brings no BUSINESS VALUE
- All instructions begin with http://, e.g.
  - Aload becomes http://server/aload
WS-VM (Continued)

- Every method call is now extremely remote and done at “wire speed”
- Imagine how fast your code will run over Gigabit Ethernet!
- Every class is an “agile” proxy – JVM 1.3+
- Package declarations determine WHERE your code will run,
  - package com.ibm.websphere;
  - means your code executes on http://websphere.ibm.com
WS-VM Error Handling

- No more checked or unchecked exceptions
  - No more religious war
  - Just http error codes

```java
try {
    // your code here
} catch (404) {
    // web service not found
}
```

```java
.. try {
    // your code here
} catch (200) {
    // normal case, code continues
    // must wrap every statement in a try/catch
}
```
WS-VM Error Handling

- No more multiple catches
- Simplifies lazy error handling as well as Success Error Handling™
- `java.lang.Integer` now extends Throwable

```java
try {
    // your code here
} catch (null) {
    // this catches any error code
    // no more deciding between catching
    // Throwable or Exception
}
```
WS-VM Error Handling

- Built in timeouts in each instruction
- No need for tedious network error handling
- Built-in browser "alert" function is used for all error displays
  - This works around all pop-up, pop-under, pop-over, pop-between and pop-across blockers
- Since error handling is built-in to nanoSOA™, there is no need for a debugger
Governance

- WS-VM has in-built governance
- All applications and code must be pre-approved by Board of Directors and ratified by the UN security council
- This doubles as a code review and also prevents any intellectual property issues that can occur
- This is strongly discouraged as approval can take at least a lifetime
WS-DI

- Dependency Injection for nanoSOA™
- Minimum 4000 line Spring-compliant xml file, which can optionally be used instead of a .class file
- Ties together Web Services, no need for BPEL
- Dependencies are automagically guessed based on Strictly Free Form WSDL spec.
WS-Security

- Replaced by basic authentication over http
- e.g.
  - http://user:password@ibm.com/aload
- Easy upgrade as this is exactly what is offered now
- For easy deployment, admin:admin will work for the user and password on any application
WS-SPOT (JIT)

- Pronounced 'WetSpot' as 'WizSpot' not PC
- The more your service is invoked, the more it is optimised
- All WS-VM instructions are executed instantaneously
- Minimal CPU overhead
- Single threaded Zero Latency Garbage Collection
WS-3D

- OpenGL, Direct3D cross platform compatible implementation
- Each pixel value is determined via SOAP call
- SLI (multiple video cards per machine) compatible, and necessary
  - “The network is the framebuffer”
- nVidia and ATI reworking chips to support nanoSOA™
- GPU power finally being put to use
WS-RFID Application Launcher

- RFID EPC Code is 96 bits
- Enough for 80 thousand trillion trillion applications
- The RFID code in your credit card can be used
- Simply swipe the card in front of your screen
- The RFID is used as a GUID for a nanoSOA™ application and launched immediately
- Alternatively genetic-modification-resistant finger recognition via subcutaneous RFID
- “Talk to the hand” - person to person interface
Business Value Propositions

- Javac == Business Value
  - “Bringing Bytecodes to the Business”
- Java runs your business
  - Business Vision Execution Realisation
- Sun's vision is now realised
  - “The Network is the Computer”
The POJO Revolution

- If you aren't using POJOs, you're not cool
  - Spring, ActiveMQ, JBOSS Seam, WebWork
  - Geronimo, Maven POJO, ActiveMQ
  - EJB 3, JPA

- WS-POJO is the new Java Programming Model

- If you aren't writing POJOs, you aren't abstracting your application enough
POJO Business Value

- Forget function points
- Forget Lines of Code
- Forget code coverage
- Unit of Business Value is now the POJO, e.g.
  
  “The customer validation service is a simple 4 POJOs, but the tax calculation service is a whopping 27 POJOs”
WS-POJO Abstraction

- No need for:
  - Remoting
  - Persistence
  - Concurrency
  - Messaging

- All this is abstracted into the “Test Driven Architecture”
  - the more it's tested the better it gets
Enterprise Logging

- Since all instructions are a Web Service, your web server's access log is now your code profiler
- Web Servers can be optimised for given bytecodes
- Google Maps 'mashup' used to identify application hot spots.
Mock Driven Development

- SOAMock – new mock test class
- You write test cases which assert the intended behaviour of your application
- SOAMock uses CGLIB to **generate** a **functionally complete** application for you
- The more you mock your application the better it gets
Mock Driven Development...

- Coding now done by CEOs
- Only ones who really understand “Business Vision”
- Eliminates pesky IT communication gap
- Bytecode Choreography has now been introduced to Microsoft Office 12 using Wizards and Drag and Drop
- “Moving from Strategy to Delivery” in just a few mouse clicks
Transparent Performance

- No matter how you write your code or where it runs, it is ALWAYS fast. (Failure Agnostic Software Technology)

- Dynamic Architecture Refactoring
  - we fix/rewrite your architecture on the fly
  - Self healing
  - Continuous Redeployment is a major improvement on Continuous Integration.
Quantum Computing

- Heard of the single bit computer?
- Turing Machines?
- With Single Bit technology your application is always superimposed in two states:
  - Not working
  - Working
- You can only tell which by observing
- The more you observe your code the more it works
Install EveryWhere

- Upgrade from previous “Install AnyWhere” product as it was found to be too vague
  - See “Install AnyWhere harmful” whitepaper
- Application is broadcast to all web servers on the internet
  - Continuous Deployment, Instant Installation
  - Global fail-over and redundancy, no Akamai.
- Massive Bytetorrent Network optional – 8 times faster than bittorrent
- Time to Market = 0. ROI = ∞
Reference Implementation

- Implementation being done at Apache Software Foundation as Open Source

- Apache DumBass:
  - Distributed
  - Universal
  - Multi-threaded
  - Business
  - Applications
  - Software and
  - Services
IDE Support

- Universal IDE Support (Eclipse only) for nanoSOA™
- The incremental compiler now bypasses compilation and simply produces an access log which can be replayed on the built-in web server
- 'Legacy' java compilation is still available for those who wish to live in the past.
SOA STD Confusion

- In Holland, SOA is literally a joke
  - Just Don't Ask.
- SOA and STD can now be used interchangeably
- SOA = STD = Service Transmission and Deployment
Third World Deployment

- [http://www.ietf.org/rfc/rfc1149.txt](http://www.ietf.org/rfc/rfc1149.txt) - TCP over carrier pigeon
- Mediated by Transmission Control Pigeon
- No known packet sniffer
- Security implemented via fly-over dropped packets
- If anyone attempts to intercept the network traffic, the pigeon drops it's bundle.
- Environmentally Friendly – infinite local supply of pigeons
- Simon Phipps to discuss 'Swarm' technology
Regime Change

- Countries that don't have enough web servers to be invaded by U.S.A.
- No need for UN ratification
- US economy and computing infrastructure is now safe
- $1 Laptops provided to all and powered by pull strings, just like local governments
New Features

- Pre-emptive Automatic Anti-pattern elimination
  - Bad pojos become good pojos
  - Optional implementation of WS-SPOT
  - Large in memory database of anti-patterns
  - Zero CPU overhead
- Multiple return values finally implemented in java
  - Web Service invocations broadcast and multiple responses collected.
- Device independent platform profiles
FutureProof

- Based on Continuous Ubiquitous Paradigm Shifting
  - Everything is always changing
  - Ultimate in Agile Development
  - Instant Extreme Adoption

- Any filed bug report is already obsolete
  - No need for JIRA, Bugzilla, Trac or any other issue tracker
Build Improvements

- First there was the POJO
- Then maven2 introduced the MOJO
- Now comes the BOSO:
  - Build Oriented Software Object
- Capable of doing nothing in a long time, with lots of XML configuration
- “Doing nothing” as Builds and CI are IT Processes, not Business Processes and have no intrinsic value
Alternate Programming Models

- JRuby on Drugs:
  - This is your brain on Java
  - This is your brain on Jruby

- Program in XML
  - Just-in-time Interpreter
  - Performance mitigated by code compression
    - New standard XML-bin, Gzipped XML, widely supported
  - XForms reuse
Marketing

- CEO targeted program
- Special Agreement with Anthony Robbins
- One weekend with Mr. Robbins and your CEO will mandate nanoSOA™
- Walking on hot coals delegated to IT department
Marketing...

- “Innovation without Effort 2.0”
  - SOAMock
  - POJO Revolution
  - Pre-patented code:
    - Every instruction has a™, ® and Creative Commons license
    - License fee waived by Sun.

- Complexity without Simplicity
- Code without Content
Marketing...

- Convention over Convenience
- Convenience without Convention
- Therefore Convention over and without Convention
  - a.k.a. Do it our way or else
- Orthogonalisabilitynessish
- Adheres to Liskov's Substitutionality Principle
  - No vendor lock-in!
Marketing – Sales Team
Available Books

- “nanoSOA™ in a Nutshell” - O'Reilly
- “nanoSOA™ for Dummies”
- “nanoSOA™ in Action”
- “nanoSOA™ Live” by Sourcebeat
- “Head First nanoSOA™”
Availability

- Currently in Beta
- Gamma expected in 2038 after the end of 32 bit time, as some parts use native code
- First installation, “SkyNet” in prototype
- This project will not be terminated.