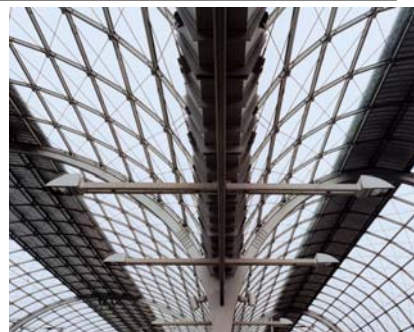


Using JBoss to Manage WiMAX Networks

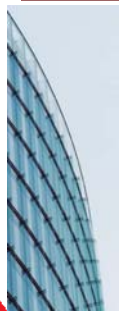
Ermanno Cavalli, Siemens Spa
JBoss World 2006,
Las Vegas, June 14th, 2006

Agenda

- Project Framework: The NetViewer Suite
- System Architecture
- Why JBoss?
- The Future

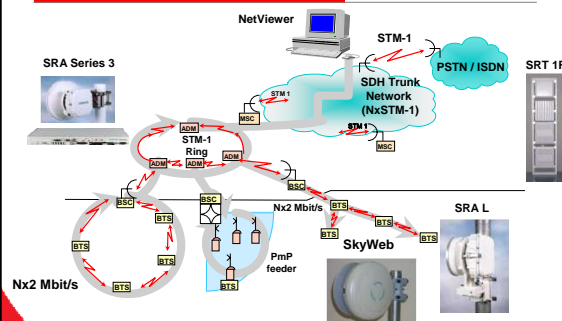


The NetViewer NME Suite

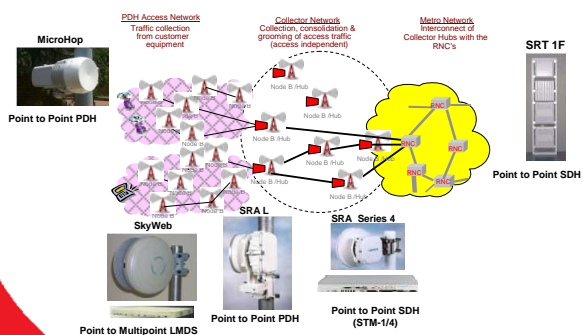


- The NetViewer NME Suite is the application solution for **Network Management** developed by the Microwave Networks Business Unit of Siemens Spa.
- Siemens sells it either as a stand-alone solution, dedicated to customers who need networks composed only of Microwave Radio, or as component of a larger suite of Network Management products developed in other Siemens Business Units, in order to offer a complete management solution for both **fixed** and **mobile** telecommunication networks, **wired** or **wireless**.
- As such, the NetViewer NME Suite offers a set of **open, standard interfaces** in order to ease the task of integrating it with Network Management applications from other manufacturers, based on transport technologies as **SNMP, CORBA, OSSJ/XML/SOAP/WebServices**.

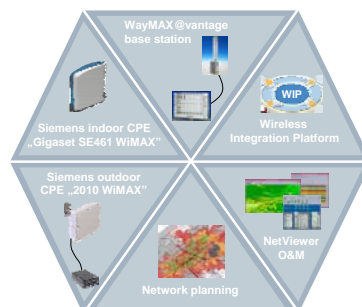
The role of Microwave Networks products in GSM networks



The role of Microwave Networks products in UMTS networks



The role of Microwave Networks in WiMAX Networks



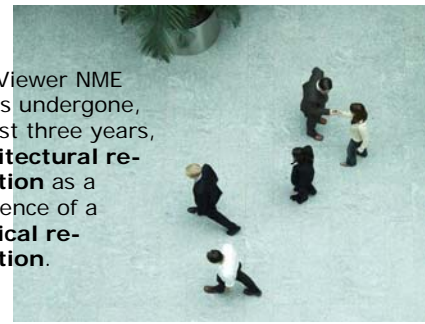
Agenda

- Project Framework: The NetViewer Suite
- System Architecture
- Why JBoss?
- The Future



Evolution of System Architecture

The NetViewer NME Suite has undergone, in the last three years, an **architectural re-orientation** as a consequence of a **strategic re-orientation**.



The need for a Strategic re-orientation

- The **strategic re-orientation** has been mandated by customers requiring less and less **stand-alone NMS**, each dedicated to the management of equipment of a single manufacturer, and mostly to a single technological domain.
- It was common practice that a single equipment manufacturer offered a different NMS for managing different technologies, e.g. transport fiber optics gear (Add-Drops), access fiber optics gear (DSLAM), radio access networks gear (GERAN, UTRAN), core networks gear (switches).



- The result: large operators had 50+ NMS in their Network Operations Center.
- Characteristics of such old-fashioned systems:
 - ✓ the main interface to access data was **GUI**
 - ✓ Architecture mainly was **client-server**
 - ✓ Development language was **C++**, because Java was not considered fast enough

The target of Strategic re-orientation

- The solution was clearly an evolution towards the introduction of the basic concepts of **Service Oriented Architecture** in the Network Management world:
 - ✓ the main interface to access data is not GUI anymore but open, standardized data interfaces, which enable the NMS to offer services on demand on the Enterprise Service Bus. Important enabling technologies have emerged such as **CORBA, XML, WebServices**.
 - ✓ Architecture has evolved from client-server to **multi-tier**
 - ✓ Development language is now **Java**, with C++ being more and more confined to areas where real-time is requested



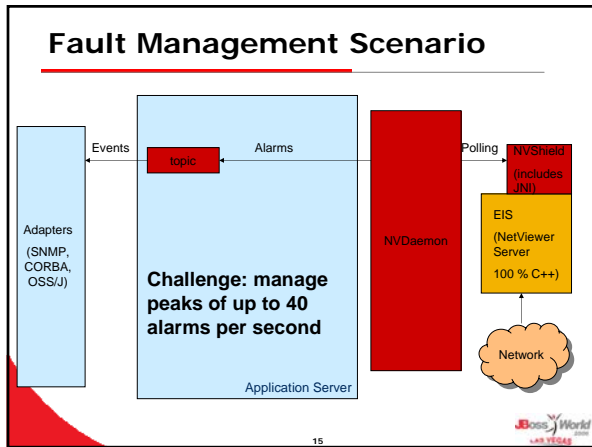
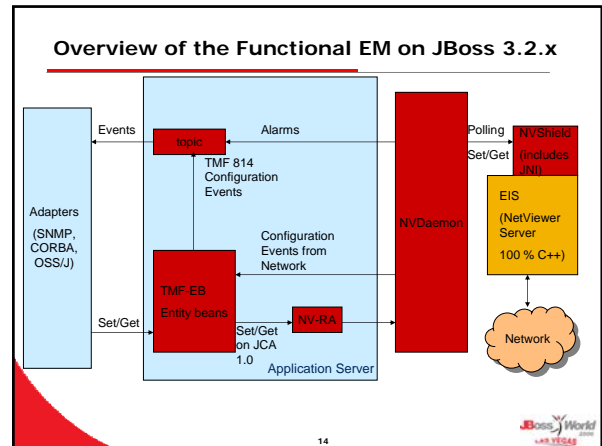
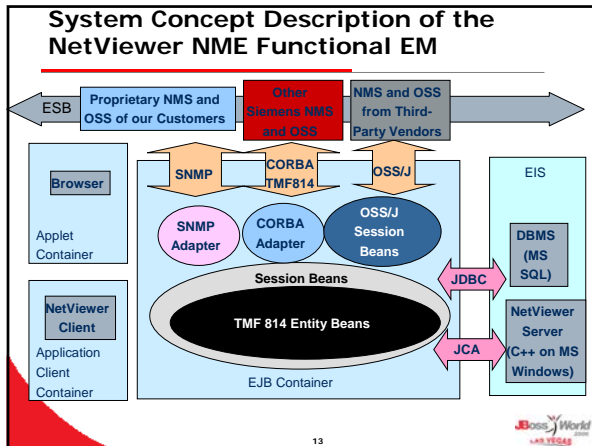
The Architectural Re-Orientation

- The starting point was a C++ client-server application.
- The option of getting rid of that as a whole, and restarting writing everything from scratch was not a possible option, because it was necessary a continuous presence on the market.
- Therefore, an evolutionary path was adopted:
 - ✓ The **client** was first moved to **J2SE**, with minimal modifications to the server.
 - ✓ Next step was working on the **server**. The choice was to preserve the lower layers of the server, which connect to the managed network and therefore need real-time behaviour. The innovation was the introduction, on top of the lower-layer server, of a **standardized data model**, in order to enable the system to support standard services for the **ESB**. The technological choice for such innovation was **J2EE**.

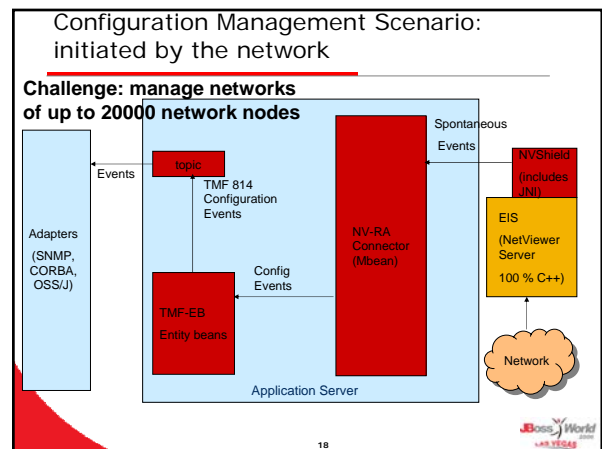
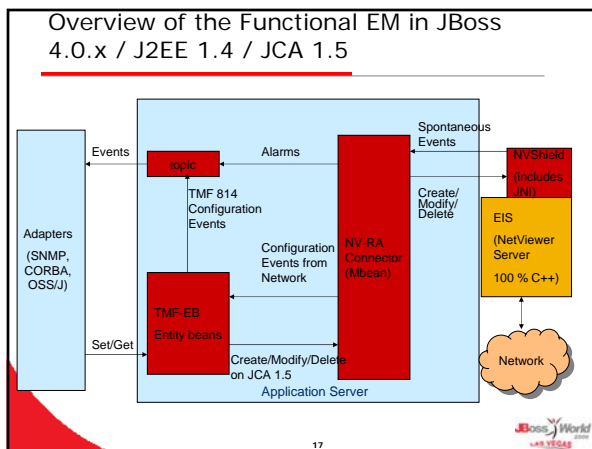
The Architectural Benefits of J2EE

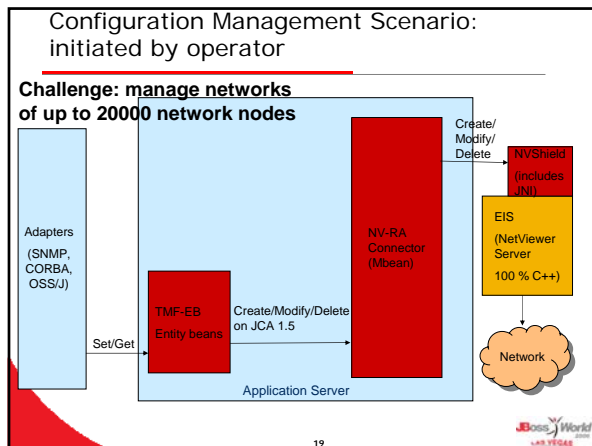
- The benefits of choosing a J2EE Application Server are introduced in the following, including:
 - ✓ Facilitates support for **integration of legacy components** of the NetViewer Suite, via standard interfaces as **JCA** and **JDBC**
 - ✓ Facilitates implementation of chosen **data model** (TMF 814) via **EJB**;
 - ✓ Facilitates implementation of the **northbound interfaces** for the **ESB: SNMP, CORBA, OSSJ/XML/SOAP/WebServices**





- ### Drawback of JCA 1.0 for Network Management Applications
- JCA 1.0 does not fit to applications where most of the spontaneous events are initiated from the EIS rather than the client side.
 - JCA 1.0 does not support the management of inbound events: therefore we had to introduce a separate process, NVDaemon, that polls for events, and which lives across the boundary of the application server.
 - This problem was overcome by JCA 1.5, which was integrated into J2EE 1.4 specifications.





Agenda

- Project Framework: The NetViewer Suite
- System Architecture
- **Why JBoss?**
- The Future

20

Why JBoss and not other commercial application servers

- JBoss has been chosen after having evaluated two other commercial application servers.
- JBoss has highlighted an almost complete absence of faults and very good performance.
- Economical advantages of using JBoss are evident, but the main driver was **quality**

21

Comparison with other Application Servers

- The first application server evaluated was Sun ONE AS 6.0, but we discarded it because of a number of problems in its integration with MS Access and because its IDE (Sun ONE Studio) had a number of issues, at the time.
- The second one was Borland BES 5.1, but we discarded it because it appeared to be too much dependent on a CORBA framework, therefore we encountered problems in interoperating with other J2EE applications based on JBoss and BEA WebLogic.

22

Very good quality and performance

- We have been using JBoss for almost three years and have faced almost no bugs: the only major one was a problem with **cascade deletes** in JBoss 3.2.6.
- The performance on the Microsoft Windows platform are very good: the only doubt is in JMS persistence on disk when queues are too large for memory.

23

Economical Advantages

- The economical advantages are evident.
- TCO for development licences has gone to zero: this is a very big advantage for large companies with large number of developers.

24

Agenda

- Project Framework: The NetViewer Suite
- System Architecture
- Why JBoss?
- **The Future**



25



The Future

- **Current Performance Issues:**
 - ✓ Start-up time relatively slow
 - ✓ Probably due to usage of EJB2: entire objects are made persistent
 - ✓ Usage of EJB3 to make persistent only what is strictly necessary
- **Architectural Improvement:**
 - ✓ Make use of Java Dynamic Proxy concept for associating remote instances (ServiceLocator Pattern)

(thanks to Sanyam Rana and Archana VS from Siemens SCS Bangalore for contributions)

26



Thank You Very Much for your attention



27

