JAIN SLEE TRIALS AND EXPERIENCES

Vodafone ES

London, 29th October 2003

vodafone

Index of the Presentation

- Drivers for considering JAIN SLEE
- Scope of Trials
- Service Creation with JAIN SLEE
- Performance Results
- Integration of Resource Adaptors
- Horizontal Scalability
- Fail- Over & Recovery
- Service Integration Aspects (OSS, BSS/ J2EE)
- OSA/ Parlay & JAIN SLEE

Drivers For Considering JAIN SLEE

- ✓ **Evolution of Intelligent Network**:
  - Java flexibility and availability to create new applications
  - Suitable environment for running telco- type services:
    - High throughput
    - Low latency
    - State model oriented

- ✓ **Standard**:
  - SW Portability
  - Smooth integration of JAIN based Adaptors

- ✓ **Flexible Architecture**:
  - Distributed
  - Several layer configurations
  - Horizontal & Vertical Scalability

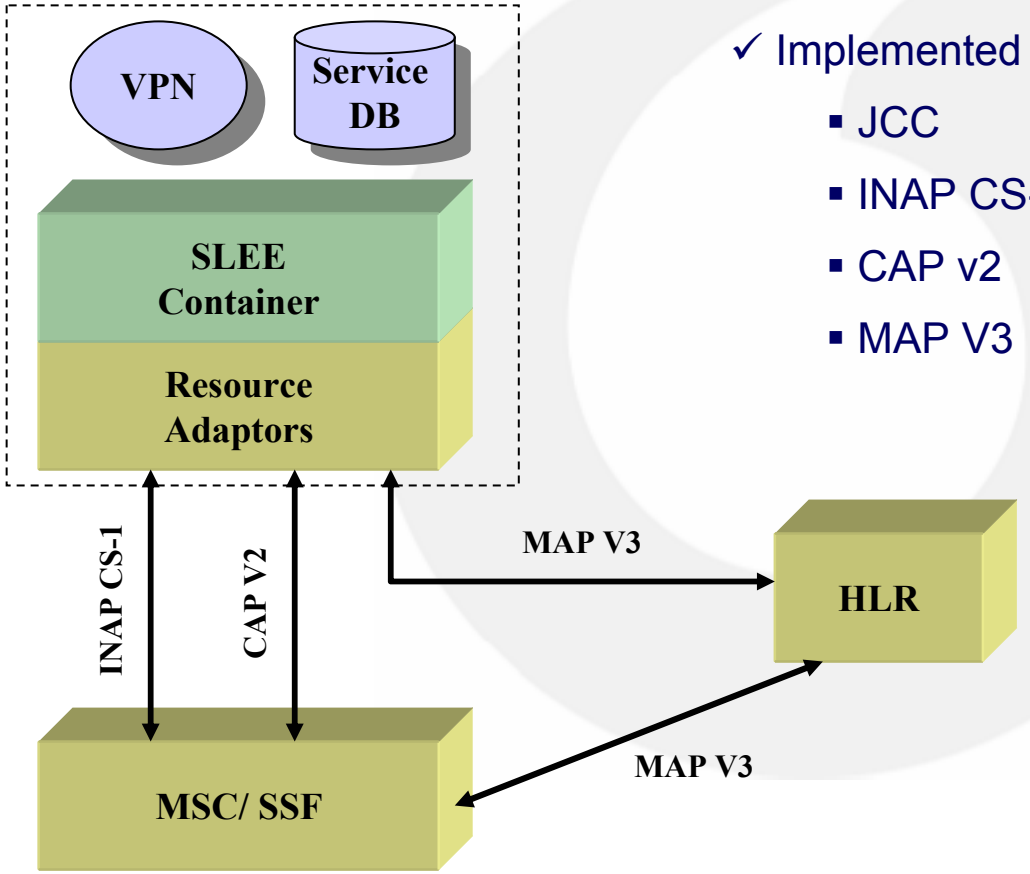
Scope of the Trials

- ✓ **Services to test:**
  - Voice service: Virtual Private Network
  - 3rd party Gateway: Messaging Manager
- ✓ **Selected Vendors**:
  - **Open Cloud** (co- author of JAIN SLEE spec)
  - **jNETx** (OSA/ Parlay & JAIN SLEE)

  (both using **SUN** HW and **Ulticom** SS7 Stack)
- ✓ **Areas Covered by the Testing**:
  - *Functionality*
  - *Performance* (Java vs. legacy platforms)
  - *System Tests* (fail-over, redundancy, stability, load balancing, scalability)

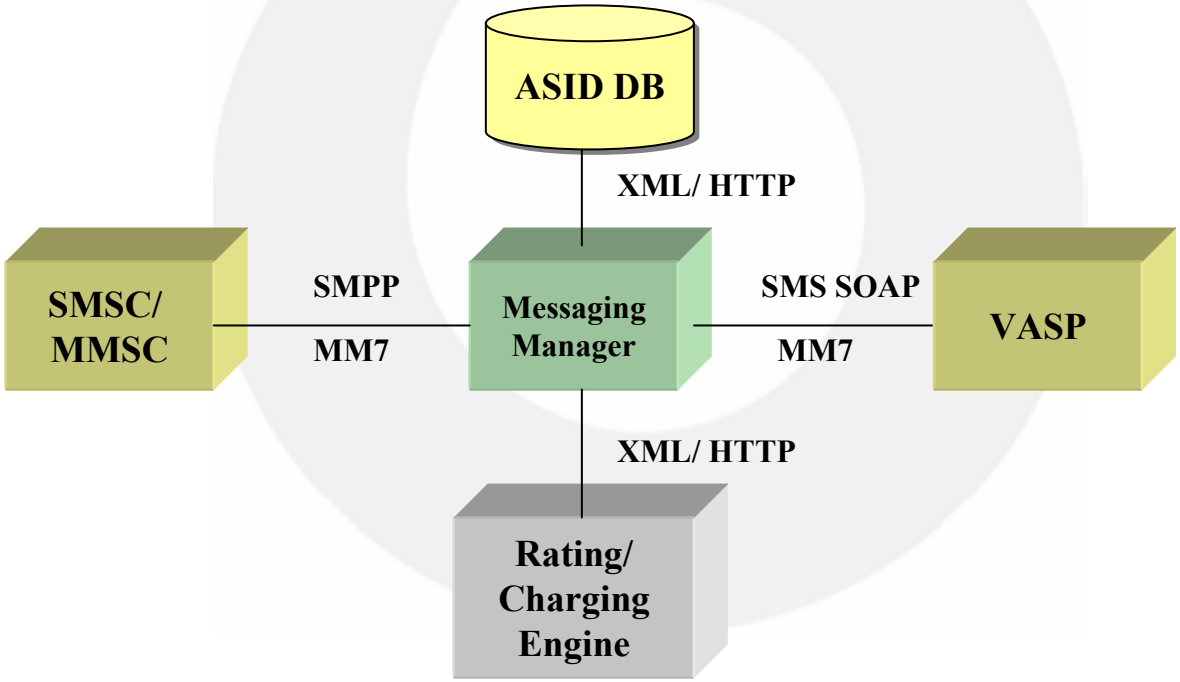
VPN Service

- ✓ CDR's in SCP (including call duration)
- ✓ BNR Location Retrieval (ATI to HLR)
- ✓ Implemented RA's:
  - JCC
  - INAP CS-1
  - CAP v2
  - MAP V3



Messaging Manager

- ✓ Messaging Gateway between a mobile user and a 3rd party
- ✓ Both SMS and MMS supported
- ✓ Notification of Delivery



Service Creation with JAIN SLEE

✓ No need to explicitly manage concurrency, events and call states

✓ The limit to available functionality was in the API implementation, rather than in the SLEE itself

✓ Trade- off between high- level SCE and Java code optimisation

✓ Rich set of API's:

- Graphical SDK
- High- level API's:
  - OSA/ Parlay (CORBA, SOAP)
  - JAIN Application Interfaces (JCC, etc.)
  - Java Rulebook API's
- Low level: JAIN Protocol API's

Performance Results

✓ Traffic Generators were built for protocols INAP CS-1, CAP v2, MM7, SMPP

✓Similar performance levels to those obtained with SS7 legacy platforms

✓ Dependency in cluster configuration and its internal communication (not part of the SLEE spec)

✓ Graceful degradation in case of overflow

✓ Optimisation aspects:

- JVM Issues (tuning of the Garbage Collector)
- Automotic Java code generation not the best choice for performance
- Handling of large objects (e.g. a MMS with a medium/ large size picture)

Integration of Resource Adaptors

✓ RA's not part of the JAIN SLEE specification as yet (supposedly scheduled for JAIN SLEE 1.x or 2.0)

✓ Generic implementation of the JCC API, so that a number of protocols can be plugged in underneath (INAP, CAP)

✓ Required tuning of the communication between SLEE container and SS7 signalling API's provided by the stack SW (**Signalware**)

✓ Aspects to consider when designing a Resource Adaptor:

- Fault Tolerance
- Asynchronous API
- Abstract the service logic from network details

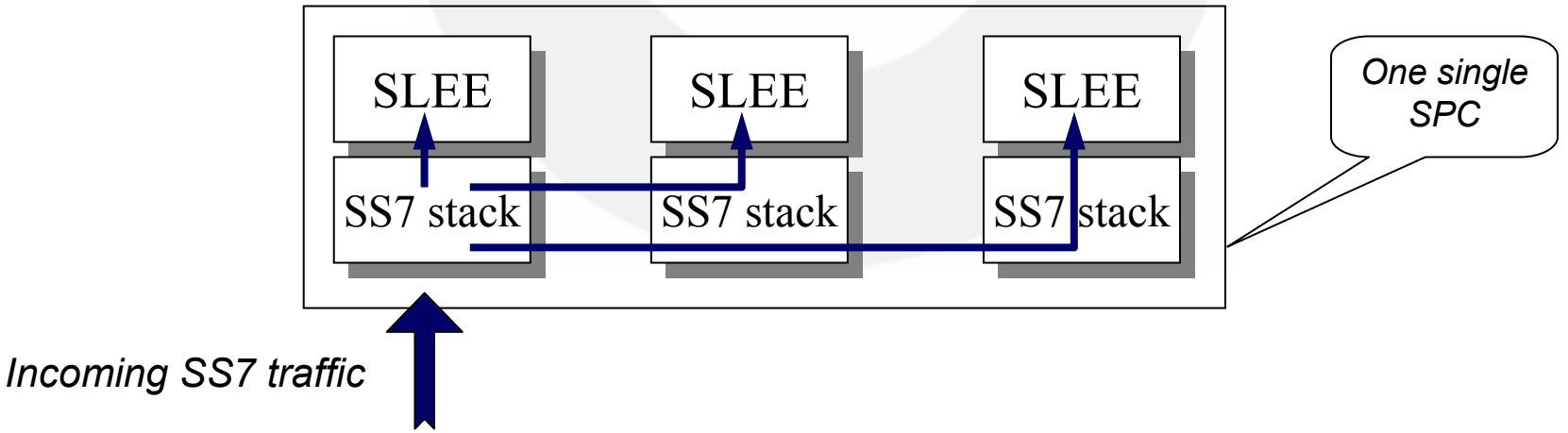
Horizontal Scalability

✓ Objective: Linear growth in performance as new servers are added to the cluster

✓ Impact: Degree of replication information between servers in the cluster to deal with potential SW or HW failures.

✓ Requires *load sharing* mechanisms, as those provided by Ulticom at the SS7 level.

✓ Scalability at SS7 integration level: Deploy a number of processes between SS7 stack and SLEE to make sure this does not become a bottleneck.



Fail- Over and Recovery

✓ *Main objective*: No single point of failure in the system. This requires:

- Redundancy of servers and communication paths.
- Replication of Information

✓ *Important aspects*:

- Elapsed time until the cluster detects a node failure and switches calls to standby node.
- Different fail- over behaviour at SLEE (HA/ FT) and SS7 levels (FT).

✓CDR's for ongoing calls at time of failure are generated by the standby SCP, closing properly dropped calls.

✓ Recovery does not take place until active SCP has registered again on the stack.

Service Integration Aspects

- ✓ **Service *Database*:** Use of in- memory database:
  - Use of the SLEE *Profile facility*
  - Online updates
  - Session data in memory vs. persistance
  - Need to keep synchronisation between IMDB and RDBMS
- ✓ ***Billing***:
  - A specific Resource Adaptor can be created for this purpose
  - or alternatevely, CDR data can be written to persistent database.

- ✓ **Management Interface**:
  - SMF (configuration changes in SLEE, RA's and SS7 stack)
  - JMX interface (as specified by JAIN SLEE 1.0 PFD2)

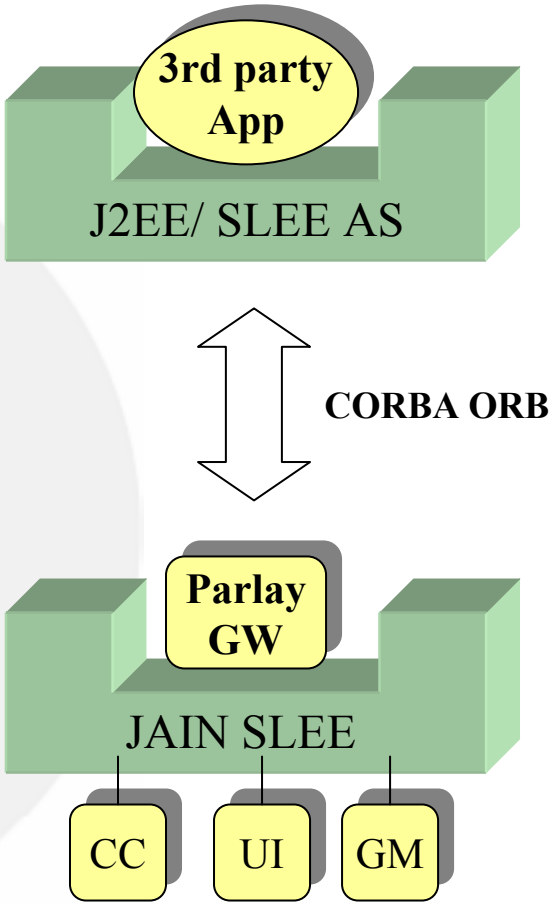
OSA/ Parlay & JAIN SLEE

✓ *Not compiting, but rather complementing each other*:

- SLEE: runtime environment
- OSA/ Parlay: services architecture & set of high- level interfaces for 3rd parties service creation

✓ *Some limited functionality found in OSA*:

- lack of support of certain required protocol parameters
- some SCS not very well adapted to real time communications (e.g. GM)



End of the Presentation

Ramón García Cadarso

Centre of Excellence (VF ES)

ramon.garcia1@vodafone.com

+34 607 13 52 40