SIP Servlets - JAIN SLEE Interoperability

Table of Contents

JAIN SLEE is a more complex specification than SIP Servlets, and it has been know as heavyweight and with a steep learning curve. However JAIN SLEE has standardized a high performing event driven application server, an execution environment with a good concurrency model and powerful protocol agnostic capabilities thus covering a variety of Telco protocols.

SIP Servlets on the other hand is much simpler and easier to get started with. Its focus is on extending the HTTP Servlets and Java EE hosting environments with SIP capabilities. SIP Servlets is more of a SIP programming framework, while JSLEE is a complete, self sufficient application platform. The fact that SIP Servlets is focused on SIP and Java EE makes it a natural fit to build JEE converged applications.

Table 1. SIP Servlets / JAIN SLEE Comparison Table

SIP Servlets	**JAIN SLEE**
Application Architecture	
Based on HTTP Servlets. Unit of logic is the SIP Servlets	Component based, Object Orientated architecture. Unit of logic is the Service Building Block
Composition through Application Router	Composition through parent-child relationship
Application State	
Servlets are stateless	SBBs may be stateful
Shared state stored in a session and visible to all Servlets with access to the session	SBB state is transacted and a property of the SBB itself. Shared state may be stored in a separate ActivityContext via a type safe interface
Concurrency Control	
Application managed: use of Java monitors	System Managed: isolation of concurrent transactions
Facilities (Utilities for Applications)	
Timer, Listeners	Timer, Trace, Alarm, Statistics, Profiles.
Protocol Support	
SIP, HTTP and Media (JSR 309)Protocol agnostic.	Consistent event model, regardless of protocol/resource
Availability Mechanisms	
Container managed state (session object) that can be replicated	Container managed state (SBB CMP, Facility, ActivityContext) that can be replicated
No transaction context for SIP message processing	Transaction context for event delivery
Non transacted state operations	Container managed state operations are transacted
Facilities are non transacted	Facilities, timers, are transacted

SIP Servlets	JAIN SLEE
No defined failure model	Well defined and understood failure model via transactions
Management	
No standard management mechanisms defined	JMX Interface for managing applications, life cycle, upgrades, …

JSLEE and SIP Servlets target different audiences with different needs, but they can be complementary in a number of real world cases.

SIP Servlets focuses on SIP and its integration with Java EE. It is also more of a SIP framework within Java EE. JSLEE is an event driven application server with protocol agnostic architecture, spanning any legacy or potential future protocols. SIP Servlets applications are generally simpler to implement and accelerate time to market for Web and SIP deployment scenarios. JSLEE has a steeper learning curve and covers a wider set of target deployment environments.

As JBoss is the only vendor to implement both specifications through Restcomm , this makes it a natural fit to build converged and interoperable JSLEE/SIP Servlets applications that are able to comply with standards in a portable manner. We built an application that could leverage standards all the way without resorting to vendor proprietary extensions by making SIP Servlets and JSLEE work together. Our "JSLEE and SIP-Servlets Interoperability with Mobicents Communication Platform" paper describes our approach and the possible different approaches we have identified to achieve the goal of interoperability between SIP Servlets and JSLEE.

You can also use our JSLEE/SIP Servlets interoperability example, showcasing our approach.