Enabling the IP Multimedia Subsystem (IMS) With Java™ Technology

Alka Gupta

Staff Engineer
Sun Microsystems

www.sun.com

Nasir Khan

Staff Engineer
SL JSR 289
BEA Systems
www.bea.com

TS-8152

Goal of This Talk

Learn how to develop Java™ technology services for the new telecoms network—IMS

Agenda

Overview of IMS

The Role of Java Technology in IMS

SIP Servlet Overview

IP Multimedia Subsystem: History

- IMS stands for IP Multimedia Subsystem
- The IMS is a user-centric service-oriented Architecture for communication systems
- 3GPP is a collaboration agreement established among various telecommunications standards bodies
- 3GPP is the standards body defining the “Third Generation Mobile Network” specifications that build on GSM
- The IMS began as a subsystem of mobile networks but has since been adopted by other standards bodies for use in other IP networks

IP Multimedia Subsystem: Relation to Internet Standards

- The IMS is an architecture containing standard protocols and schema’s
  - Developed through the Internet Engineering Task Force
- The IMS architecture was defined before any protocols were selected
  - The IMS is not “based” on SIP, although SIP is now essential
- The 3GPP specifications split network functions into logical groupings
  - Defines “black boxes” with specific behaviors and protocol-based interfaces
  - There is a relatively explicit definition of functional groupings as “implementable units” which may be sourced from different providers with little (no) functional overlap with other network elements

Why IMS?

- Service providers
- Network equipment providers
- Operators
- End users

Why IMS? (Cont.)

- Service providers
    - Combines services from fixed, mobile and broadband networks
    - Combine multiple media sessions (video, voice, music, pictures, text, data)
    - Common, Integrated platform for fast cost-effective introduction of new services
    - Leverage common applications, and subscriber data across multiple access networks

Why IMS? (Cont.)

- Network equipment providers
  - “Game changer” beyond the core
  - Requires end-to-end enhanced quality of Service (QoS), service blending, security and resource management
  - Opens new service opportunities in integration, services and multi-vendor
  - Increased network traffic accelerates growth

Why IMS?

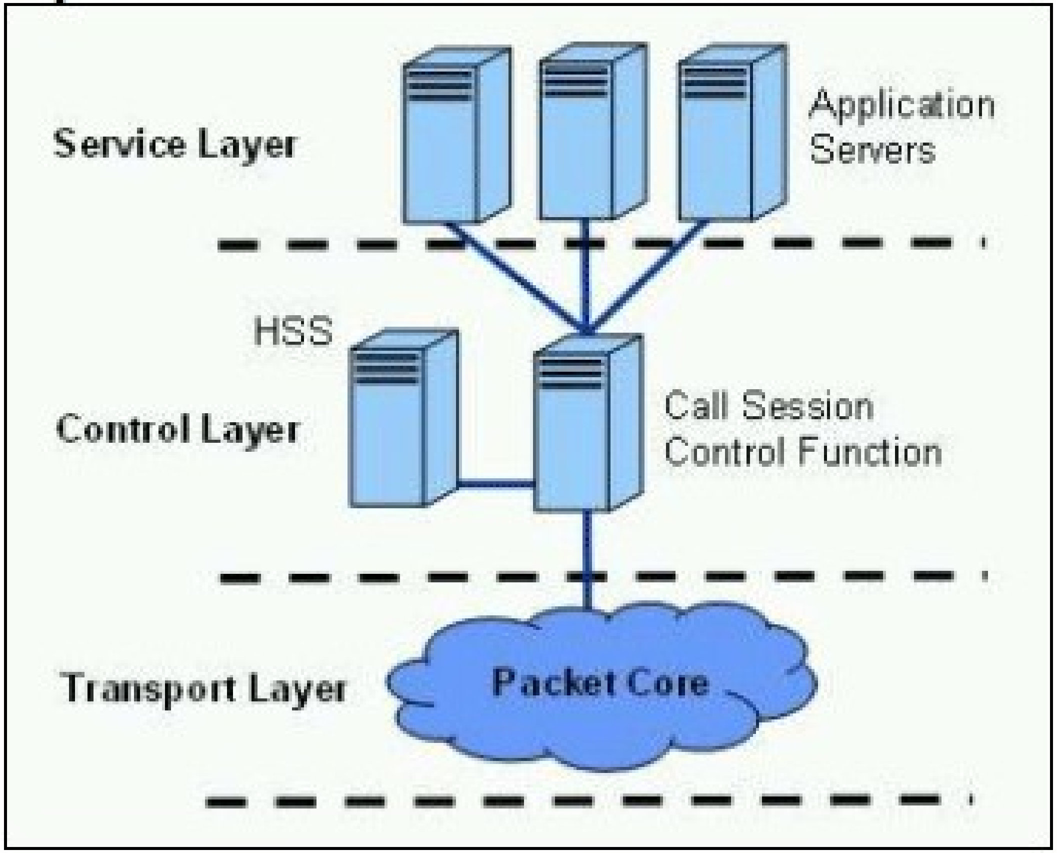
- End users
  - Ease of Use, productivity and fun
  - Real personalization
  - Seamless combination of real-time voice, text, video, images, games, Web, e-mail, and more
  - Device independent person-to-person multimedia
- Operators
  - Customer retention by enriching their experience
  - Fast creation of attractive services, deliver converged, network agnostic, real-time, multimedia applications
  - Flexible charging model
  - Replacement of silo solutions

IMS Value Proposition

- Quality of Service (QoS)
  - Predictable experience, not just best effort
- Charging
  - Services and service combinations, not just volume
  - Flexible business models (QoS, duration, volume)
- Integration of different services
  - Foster ecosystem of service developers
  - Rapid deployment of new services
  - Dynamic combination of voice, data and video
  - Real fixed-mobile convergence

IMS Architecture

Layered IMS Architecture



Source: *Unstrung Insider*

IMS Architecture

- Transport layer
  - Network agnostic (GSM, CDMA, 3G, SS7, others)
  - Standard interfaces between Transport and Control layer
  - Potentially independent of vendors in packet core and other layers
- Control layer
  - Home Subscriber Server (HSS)
  - Call Session Control Function (CSCF)
  - Media gateways (BGCF, MGCF, MRF…)

IMS Architecture

- Service layer
  - Standard open protocols such as ISC (IMS Session Control)
  - SIP application servers
  - OSA and SCP

Agenda

Overview of IMS

The Role of Java Technology in IMS

SIP Servlet Overview

Trend Toward IT Infrastructure

- Telecoms want to build next generation network using proven IT infrastructure
  - Leverage portals, HTTP Container, Web Services, etc.
  - Off-the-shelf software
  - Next generation applications blur the boundary
  - Container managed software
    - State management, persistence, transactions
- The world is converging on IP as transport
- Java technology is having the effect on telecoms networks that it had on enterprise seven years ago
  - Standardized interfaces using the JCP
    - JSR 116 and JSR 289

SIP Protocol Basics

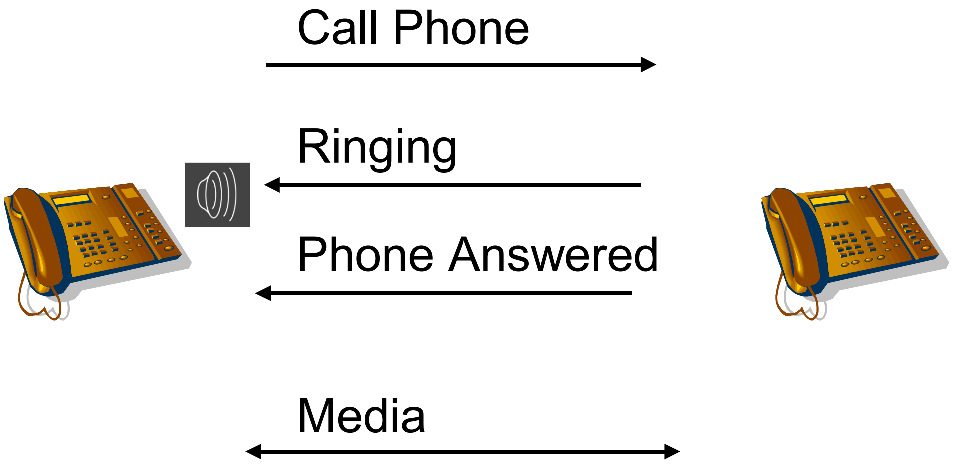
- SIP is based on HTTP
  - Text-based protocol
  - Headers followed by a message body
  - A lot of the same status codes
    - 200 OK, 401 Unauthorized, 500 Internal Server Error
  - Some new ones introduced
    - 180 ringing, 300 multiple choices
  - Similar Digest authentication mechanism
  - Uses UDP, can use TCP or SCTP for larger packets

Differences from HTTP

- HTTP Synchronous and Client Server
  - Client Requests, immediate server response
- SIP asynchronous and peer-to-peer
  - UAS accepts requests, responds later
  - Messages may be retransmitted due to timer expiration
- HTTP may front other databases or servers, but client is only aware of HTTP server
- SIP servers often involve complex network interactions
  - Client may have direct communication with other network elements, for media, or for SIP messages

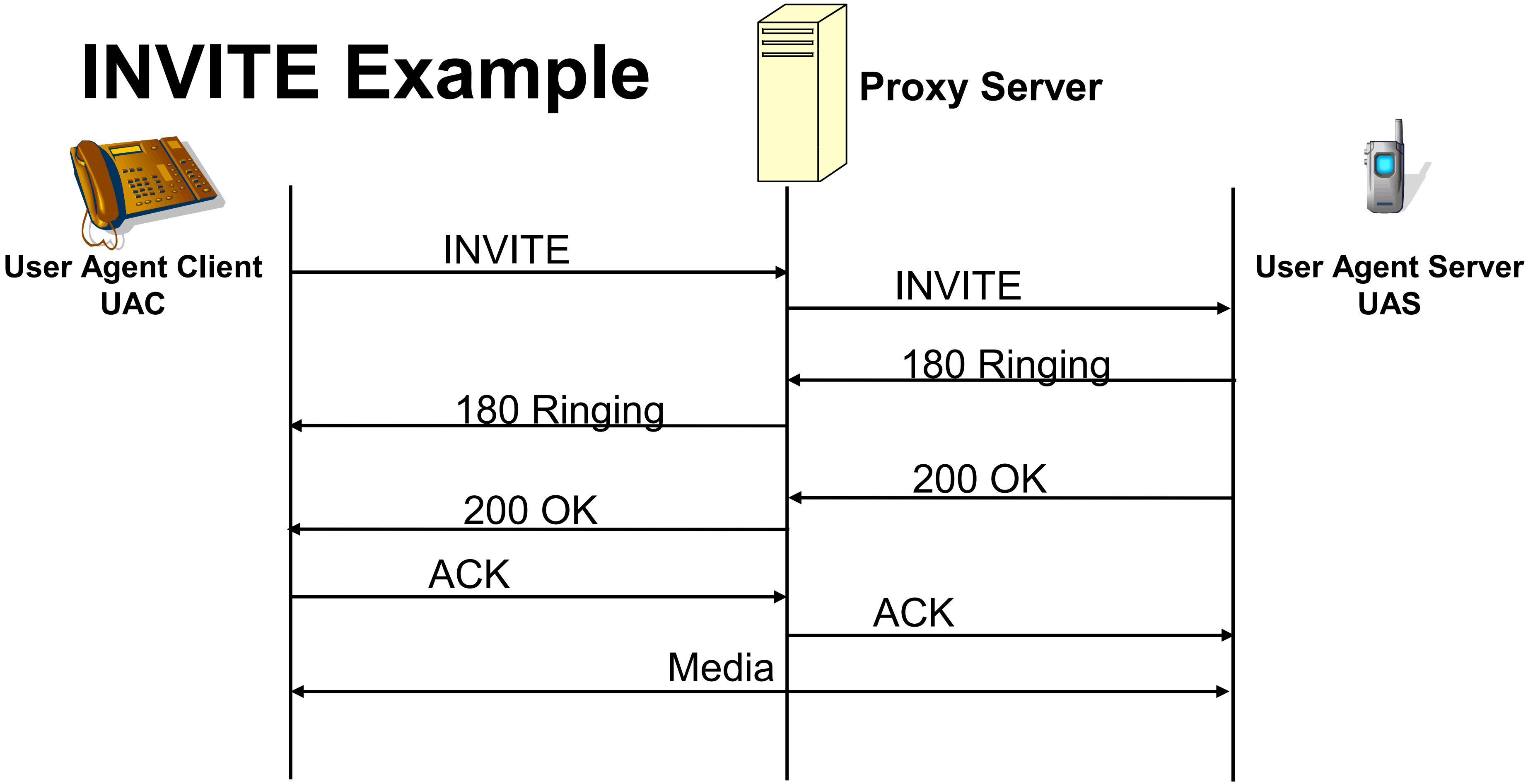
SIP Basics
(Session Initiation Protocol)

- Signaling Layer or Control Plane = Meta data about the call



- Media Layer = Sounds bits themselves

INVITE Example



- Most SIP-enabled devices act as both client and server, hence they’re UAs
- Some act only as clients or only as servers, hence UAC and UAS

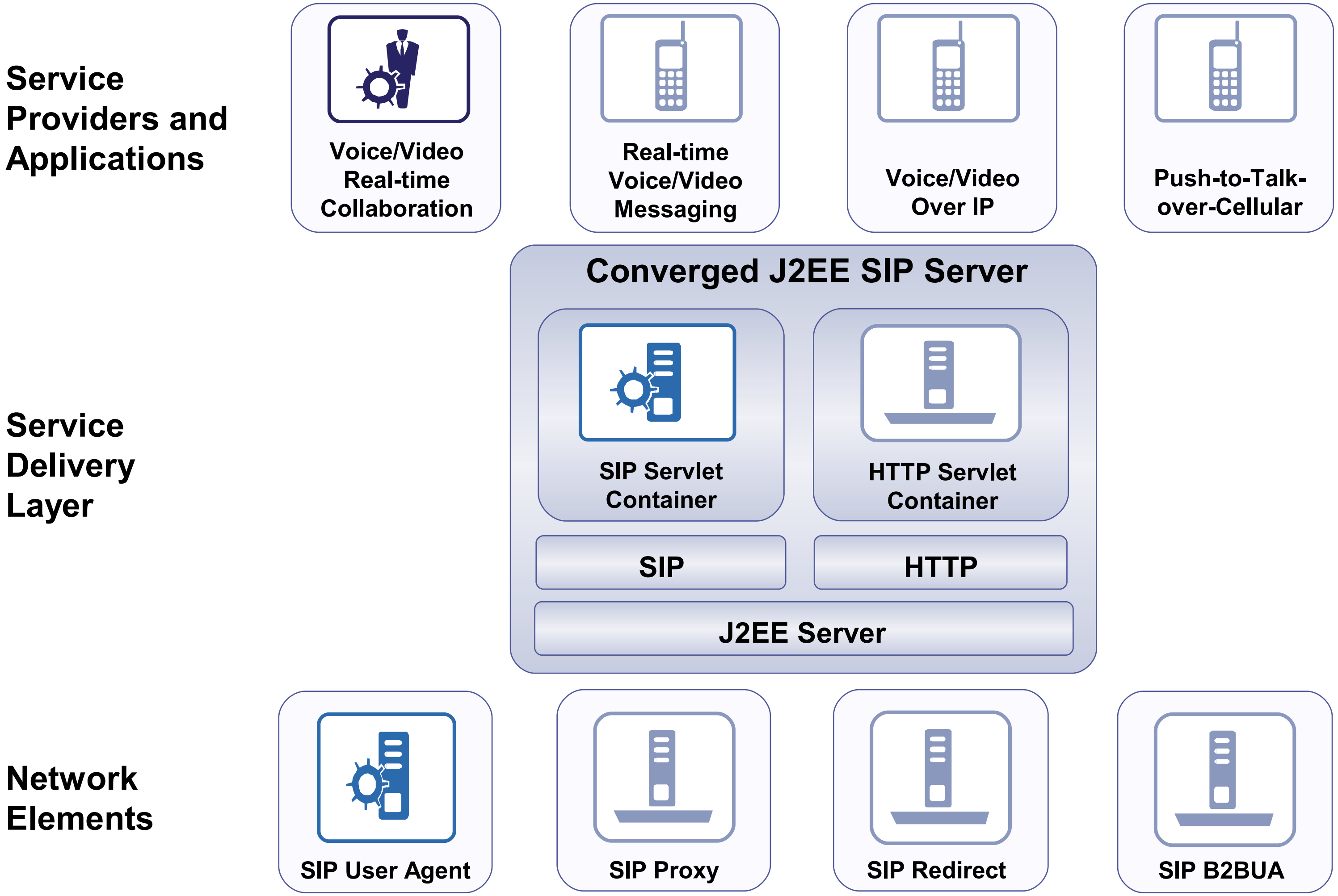
Basic Server Types

- SIP proxy
  - Receives SIP requests from a UA or another proxy and forwards or proxies the request to another location
- Redirect server
  - Receives a request from a UA or proxy and returns a redirection response (3xx), indicating where the request should be retrieved
- Registrar server
  - Receives a SIP registration request and updates the user’s location information into a database of user locations
- B2BUA
  - Is a UAS and UAC combine. Most real world applications are modeled like this
- Inter-working gateway
  - Translates SIP call into another network, like SS7

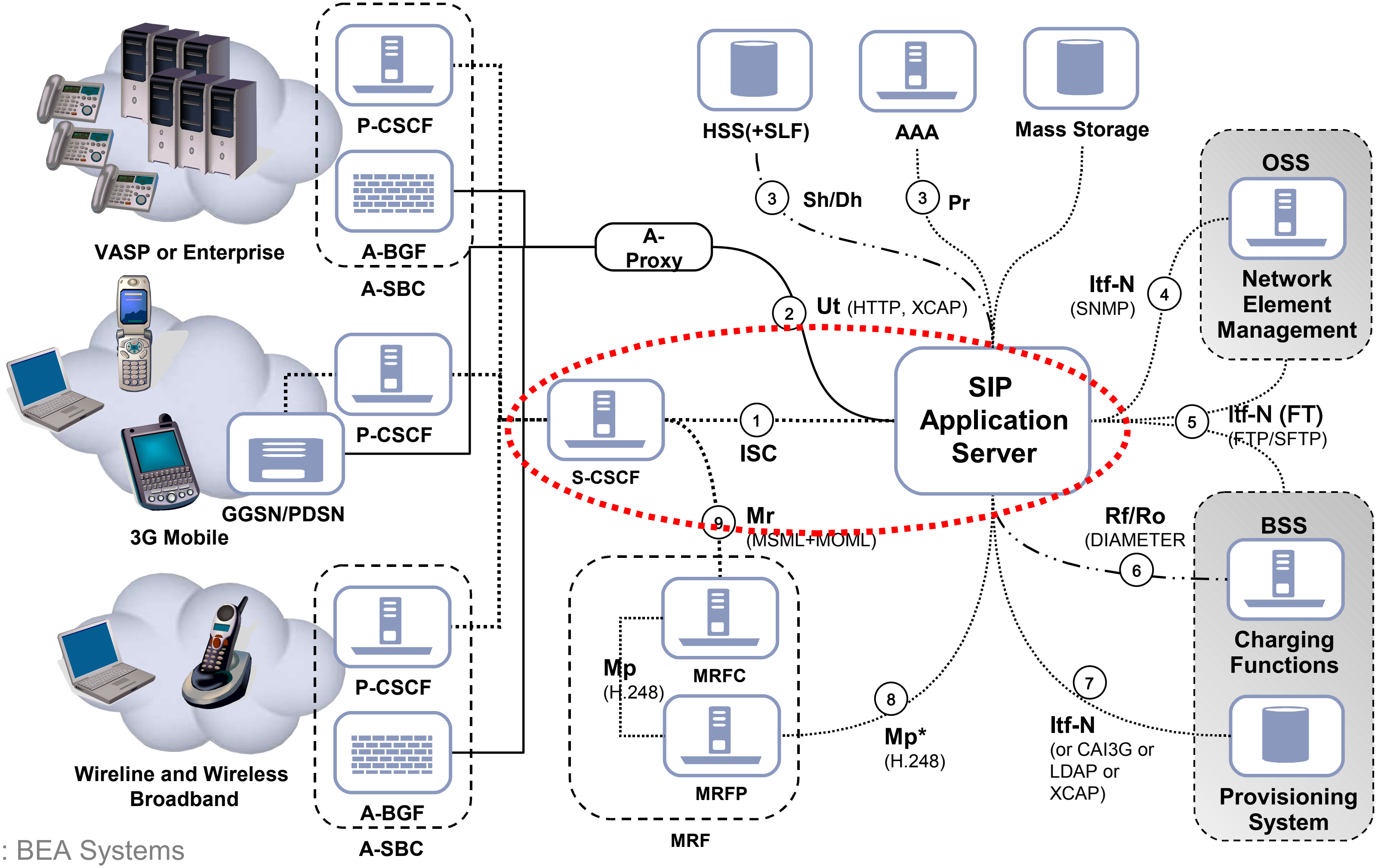
SIP Applications

- Basic Routing functions: (simple number translation, Find me, etc.)—Redirect, proxy, B2BUA
- Complex routing: (header manipulation, transcoding, etc.)—B2BUA
- Media capable: (IVR, voice mail, etc.)—B2BUA
- Presence and Location: (Presence server, client, watchers)—Mostly B2BUA
- Converged: (click-to-call, conferencing etc.): proxy or B2BUA with HTTP component

Converged J2EE™ SIP Container



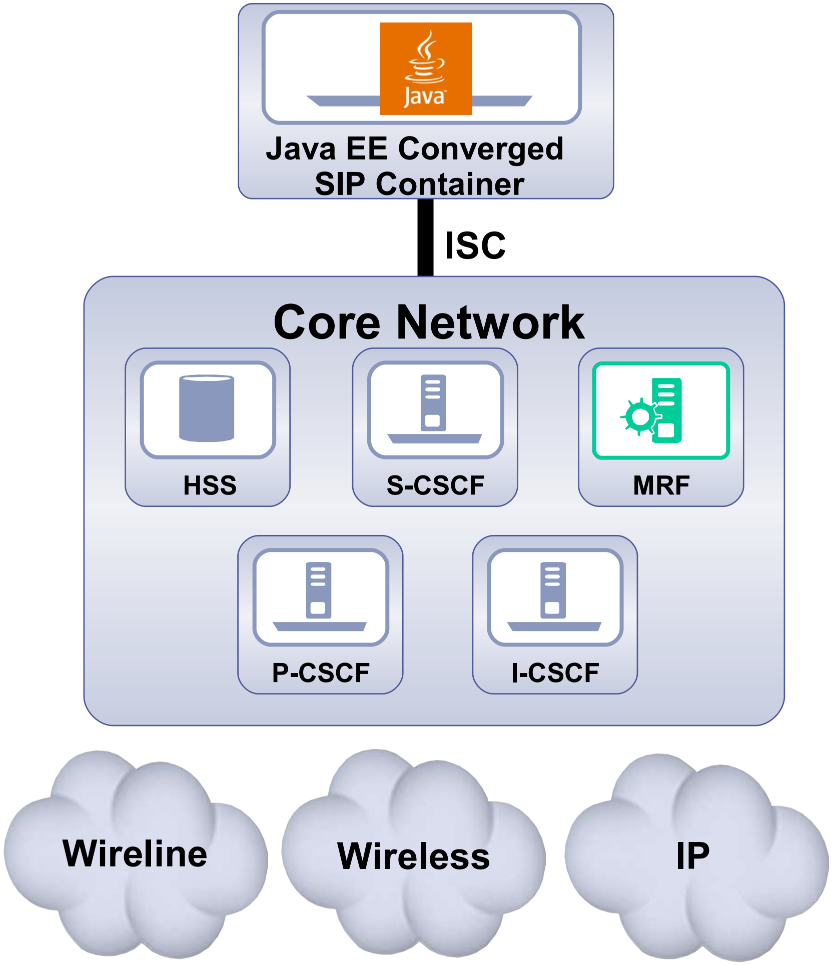
IP Multimedia Subsystem (IMS)



Source: BEA Systems

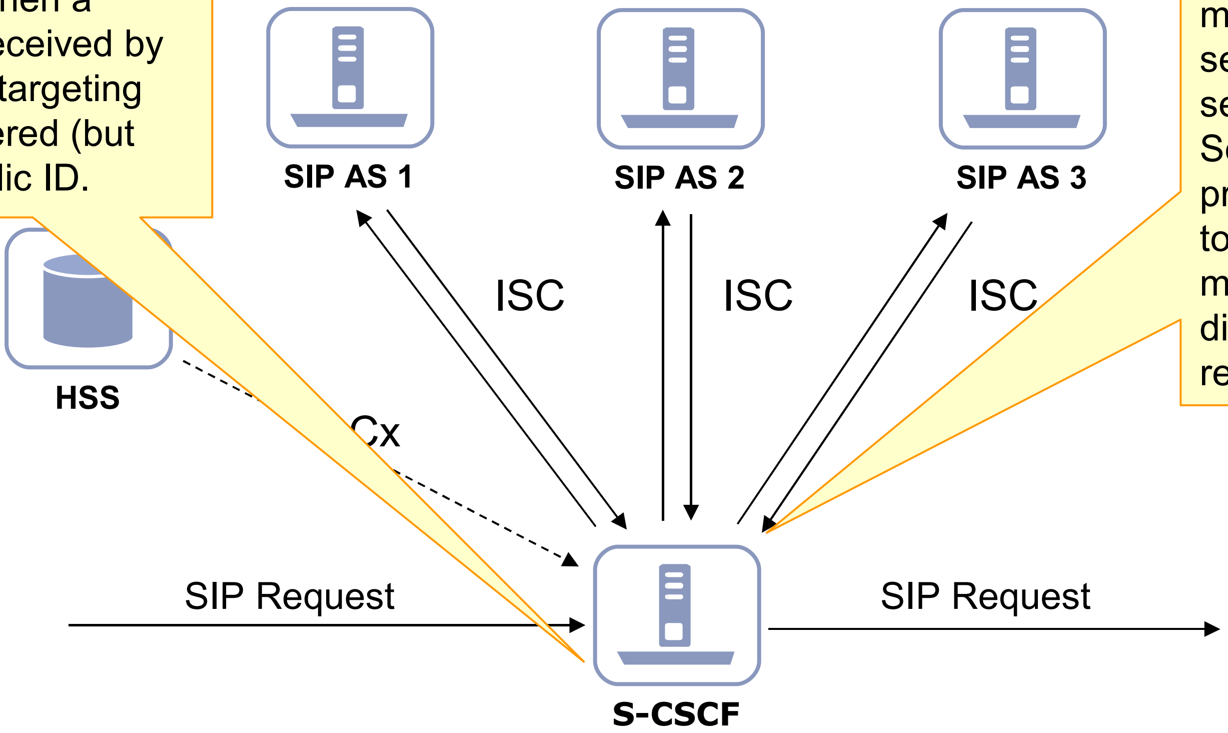
Role of Container in IMS

- Integrates with IMS network via ISC interface
- Standardized on SIP
- Standardized application server for in-house and third-party development
- Combine services from fixed, mobile, broadband networks



IMS Service Control

An S-CSCF instance is assigned to each user when that user REGISTERs with the system or when a request is received by the I-CSCF targeting an unregistered (but known) Public ID.



Filter Criteria govern the forwarding of SIP messages to one or several application servers. Application Servers may either proxy requests back to the S-CSCF or may teminate the dialog and send a response.

Agenda

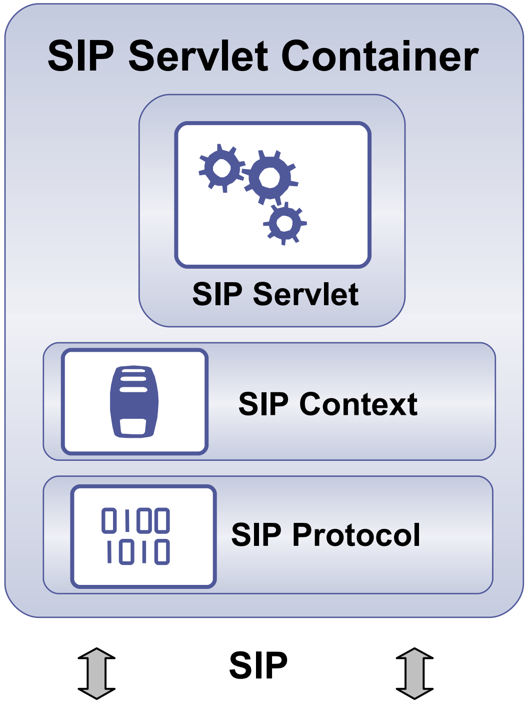
Overview of IMS

The Role of Java Technology in IMS

SIP Servlet Overview

Container Introduction

- JSR 116 SIP Servlet API
- SIP protocol stack—Implements key RFCs
- SIP session and transaction management
- SIP servlet lifecycle management
  - Shared container startup/shutdown with HTTP servlet container
- Concurrent execution of SIP and HTTP applications
  - Links HTTP/SIP servlet contexts
- Management via Admin Console
  - JMX/MBean-based
- JAAS-based security via WLS
  - SIP/HTTP apps can share same user ID and password



Extension to HTTP Servlet Model

- HTTP Servlets need only send responses
  - Only defined for origin servers
- SIP Servlets are broader
  - May operate in proxies or B2BUAs
  - Functions required
    - Send responses
    - Proxy requests
    - Initiate requests
    - Receive responses

- Only asynchronous events in HTTP Servlets are requests
- In SIP, asynchronous events are requests and responses
  - Need doResponse() in addition to doInvite()
- Model is the same though
  - Event arrives to system (request or response)
  - Servlet is determined
  - Service method invoked

SIP Servlet Specification

Events: Receive request or response, timeout

Single entry point: Defined by **`javax.servlet.Servlet:`**

```
void service(ServletRequest req, ServletResponse resp) throws …;
```

which dispatches on whether event is request or response

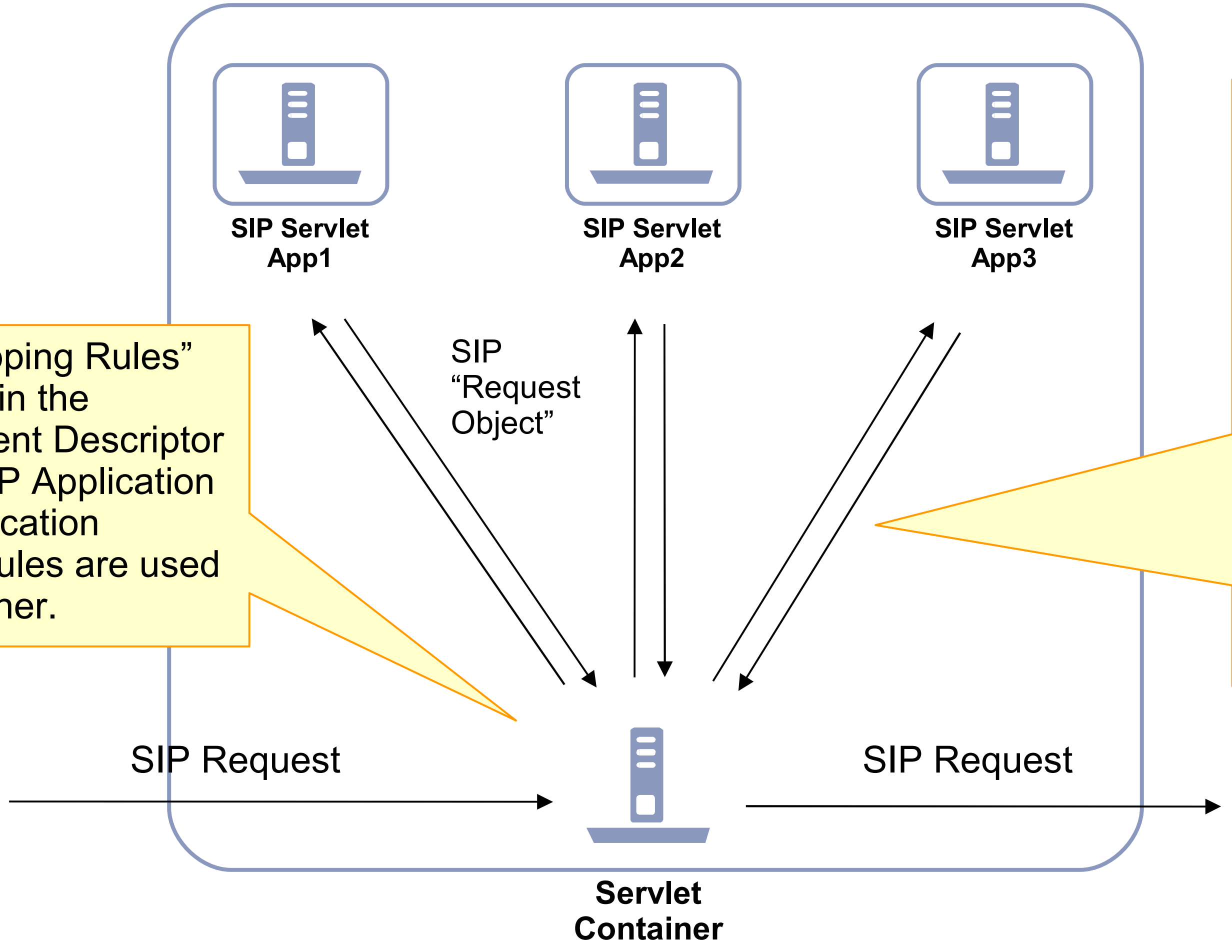
- void doRequest(SipServletRequest) throws ServletException
- Dispatches on SIP method:
  - doInvite(SipServletRequest req), doAck, doOptions, doBye, doCancel, doRegister
  - Extension methods

- void doResponse(SipServletResponse) throws ServletException
- Dispatches on status code:
  - doProvisionalResponse(resp)
  - doErrorResponse(resp)
  - doSuccessResponse(resp)

“Standard” Platform: SIP Servlet API

- The SIP Servlet API is a Java Community Process^{SM} services standard (JSR 116/JSR289)
- The SIP Servlet API specifies a programming model based on the “Cascaded Services Model (SERL)”
    - Allows SIP/HTTP applications to be transparently co-located on a single host or deployed on different hosts within the network without the need to modify the application code
- Interaction between applications is managed by the “Servlet Container”
- The relative “signaling” performance between such “co-located” applications and the same applications deployed on separate AS instances is 10 to 100 times more efficient
    - The conceptual “architecture” of the IMS network is not affected and no standards are compromised

Application Composition



The "Mapping Rules" declared in the Deployment Descriptor for the SIP Application and Application Routing rules are used by container.

Mapping Rules govern the "routing" of SIP messages between applications within the container. SIP Applications may either "proxy" requests back to container or may terminate the dialog and send a response.

Servlet Mapping Rules Example

```
<servlet-mapping>
<servlet-name>servlet1</servlet-name>
<pattern>
      <and>
            <or>
                  <equal>
                  <var>request.method</var>
                  <value>INVITE</value>
                  </equal>
                   <equal>
                  <var>request.method</var>
                  <value>SUBSCRIBE</value>
                  </equal>
            </or>
            <or>
                  <equal>
                  <var>request.method</var>
                  <value>MESSAGE</value>
                  </equal>
                  <not>
                        <equal>
                        <var>request.from.display-name</var>
                        <value>joe</value>
                        </equal>
                  </not>
            </or>

      </and>
</pattern>
</servlet-mapping>
```

```
(Method=”INVITE” OR
Method=”SUBSCRIBE”) AND
(Method = “MESSAGE” OR (NOT
Header = “from” Match =
“joe”))
```

Proxy Code Example

```
// Sample code example that proxies all request.
// This example uses generic doRequest method

protected void doRequest(SipServletRequest request)
               throws ServletException, IOException {
    Proxy proxy = request.getProxy();
    proxy.setSupervised(false);
    URI requestURI = (URI)request.getRequestURI();
    proxy.proxyTo(requestURI);

}
```

Enhancements in SIP Servlet API 1.1

- Application composition standardized
- Formal distinction between callee and caller services
  - Therefore SIP servlets can now easily determine on whose behalf they are being invoked
- The ability to map certain communication features to SIP Servlets and invoke those features in an independent manner
- Enhanced SIP Servlet control of Application Invocation
  - SIP servlets should be able to convey their intentions about how they wish subsequent service invocation to take place

Enhancements in SIP Servlet API 1.1

- Enhance specification to support additional RFC, addressing IMS specific needs and behaviours (Path, Service-Route, etc.)
- Provide explicit support for B2BUA type object
- The ability to move seamlessly between HTTP and SIP servlets within a convergence application (and other J2EE components)
- Enhancements for the ordering and dynamic application of Servlet Mapping Rules
- Support for multi-homed hosts
- Aligned with Servlet Spec 2.5
- Lightweight container model. Annotations

Summary

- IMS is the telecom network of the future
- Open Java technologies will have a role
- Developers will be able to leverage this network
- Applications drivers are video, voice, data, and messaging

For More Information

List

- http://jcp.org/en/jsr/detail?id=116
- http://jcp.org/en/jsr/detail?id=289
- http://www.sipservlet.org/
- http://dev2dev.bea.com/wlcp/

Q&A

Enabling the IP Multimedia Subsystem (IMS) With Java™ Technology

Alka Gupta

Staff Engineer
Sun Microsystems

www.sun.com

Nasir Khan

Staff Engineer
SL JSR 289
BEA Systems
www.bea.com

TS-8152