

# IBM WebSphere Application Server Communications Enabled Applications

## CEA architecture



This presentation provides an overview of the architecture of the IBM WebSphere® Application Server Communications Enabled Applications feature and how the product components work together to provide communication services.

---

## Table of contents

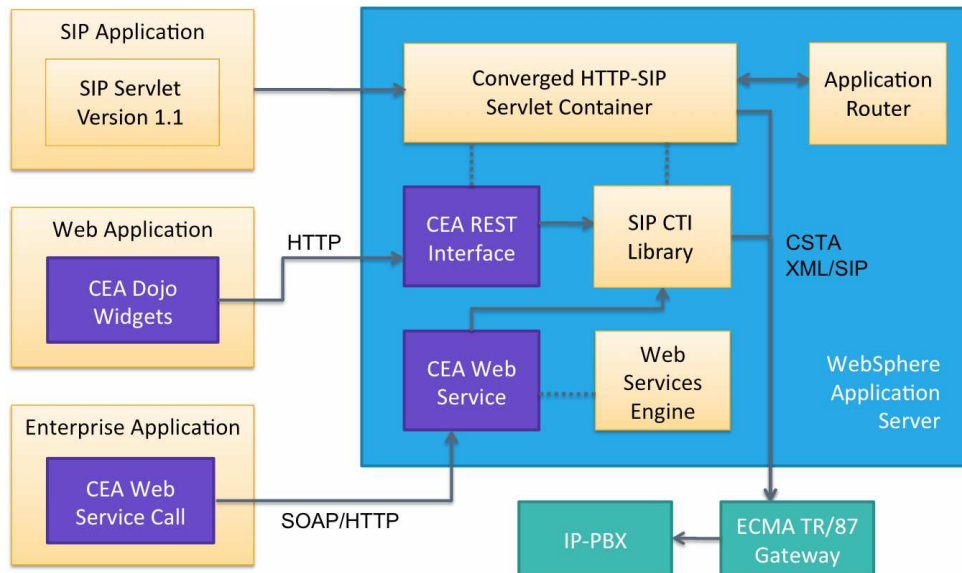
- Architecture overview
- CEA call flow
- CEA collaboration flow

The first section of this presentation gives an overview of the Communications Enabled Applications (CEA) architecture. The next two sections walk through two usage scenarios – one for making a telephone call using an embedded ClickToCall widget in a web page, and one for sharing session data across a web link – and show how the CEA components drive telephony and collaboration services.

## ***Architecture overview***

This section gives an overview of the architecture of the WebSphere Application Server Communications Enabled Applications feature.

## Architecture



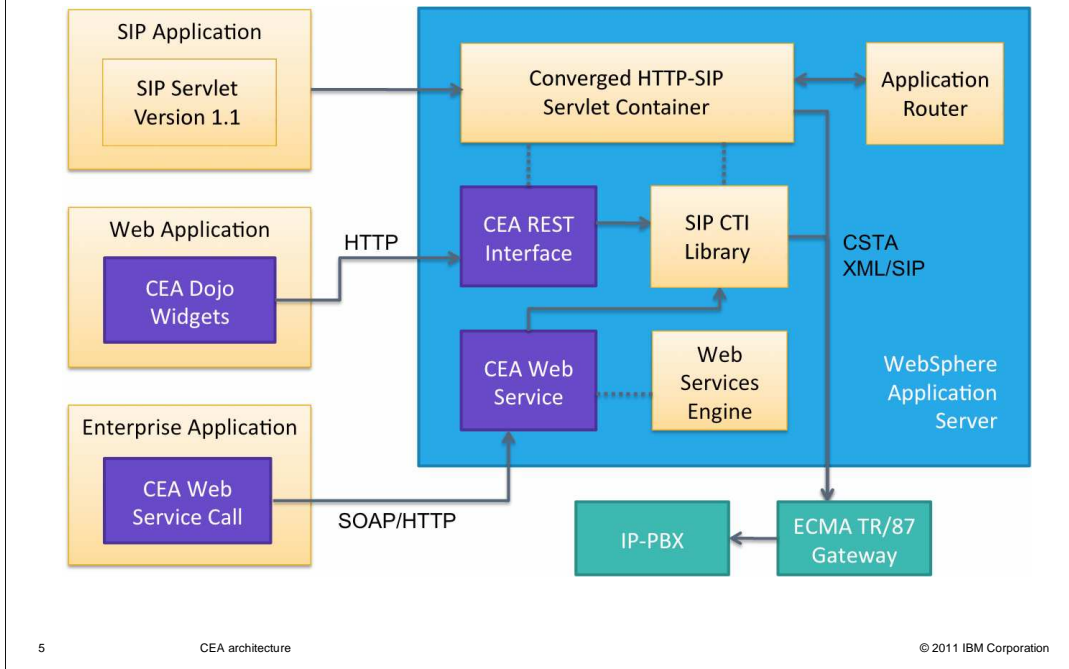
4

CEA architecture

© 2011 IBM Corporation

This diagram provides an overview of the IBM WebSphere Application Server architecture for communications applications. WebSphere Application Server supports SIP-based communications applications that directly implement communications functions. The Communications Enabled Applications (CEA) feature extends this support to include business applications that require specific communications features. In the diagram shown here, both the enterprise application and the Web application are considered types of business applications. Business applications communicate outside of the application server through the SIP computer telephony integration (CTI) layer, accessing an IP-PBX through a standard ECMA TR/87 gateway. In order to use your IP-PBX with the CEA feature, it must support the ECMA TR/87 standard protocol or use an ECMA TR/87 compliant gateway.

## Architecture



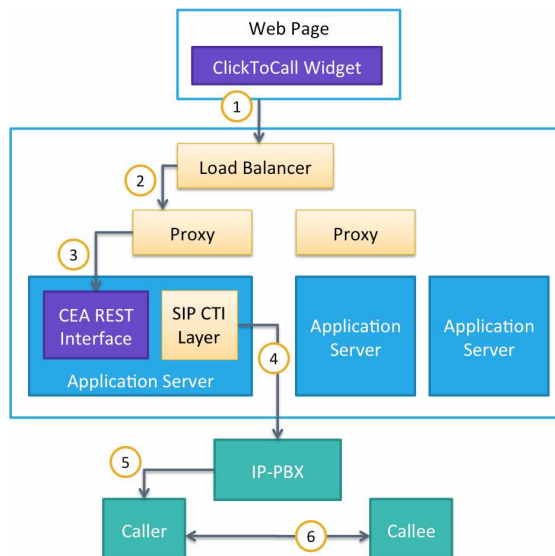
A SIP application is any application that contains a SIP servlet, including converged applications that contain both SIP and other Java™ EE components. The SIP applications that you run on WebSphere Application Server can be built using the functionality in the SIP servlet 1.1 specification, or JSR 289. The converged HTTP and SIP container in the application server provides JSR 289 support, which includes communication with an application router. The application router is a new component under JSR 289 that provides a flexible mechanism for building end-to-end SIP services out of multiple components. It exists outside of the SIP container and communicates with the container to let it know which applications need to be invoked next in a given application chain.

Business applications can access the CEA communications services through a REST interface or a Web services interface. The Web application shown in the diagram is an application that has a Web front end – for example, a WAR file or that includes Web pages. Any page in the Web application can contain new CEA widgets, built using the Dojo toolkit. These widgets provide an easy interface to telephony services – like making a call, or monitoring for a call to come in. The widgets communication over HTTP with the CEA REST interface in the application server. The enterprise application shown here can be any application that performs operations for your business – it may or may not have a Web front end, and is typically packaged as an EAR file or a JAR file. This application might contain a call to the new CEA Web service interface, communicating with the application server using standard communication protocols, like SOAP over HTTP.

## ***CEA call flow***

This section walks through a usage scenario for making a call with a ClickToCall widget, showing how the CEA components enable making the telephone call.

## Scenario: Making a call (1 of 2)



1. A user clicks the ClickToCall widget, which sends an HTTP REST request
2. The HTTP request gets filtered through the edge components, if present
3. The application server receives and interprets the REST request

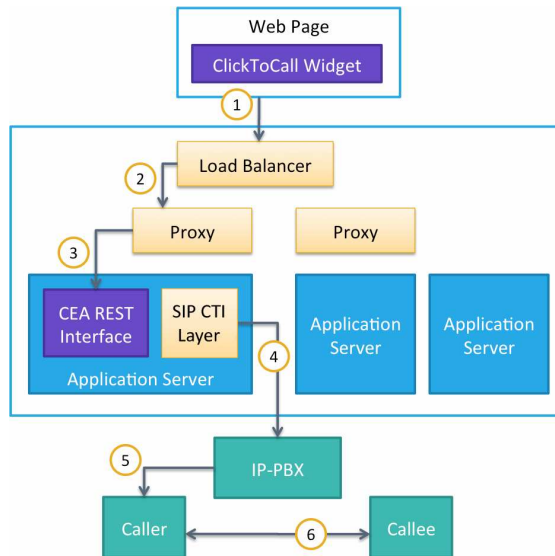
7

CEA architecture

© 2011 IBM Corporation

In step 1, a user – perhaps a customer visiting some company’s Web site – enters a telephone number into the ClickToCall widget on the company’s Web page, requesting to be connected over a telephone link to someone at the company, perhaps a customer service representative. This causes an HTTP REST request to be generated and sent to the application server. In step 2, the HTTP request gets sent through the edge components that might be present in front of the application server, to balance traffic across the infrastructure – for example, the request might get filtered through a load balancer and a SIP proxy before reaching the application server. The HTTP REST request reaches the application server in step 3, and the server processes and interprets the request to determine the appropriate next action.

## Scenario: Making a call (2 of 2)



4. The application server contacts the IP-PBX through the SIP CTI layer, making a call request
5. The IP-PBX notifies the caller to call the callee
6. A call is established between the caller and callee

In the case of this example, the REST request is trying to create a telephone call between a customer visiting a company's Web site and a customer service representative. In step 4, the application server takes the information about the call that needs to be connected and sends it out to the IP-PBX, through the standard SIP CTI layer in the server. When the IP-PBX receives the request to create the call in step 5, it notifies the caller – in this case, the caller is the customer who might be accessing a company's Web site, to call the callee, who might be a customer service representative for the company. Then, in step 6, a call is established between the callee and the caller, and the customer and the customer service representative can begin speaking with each other.

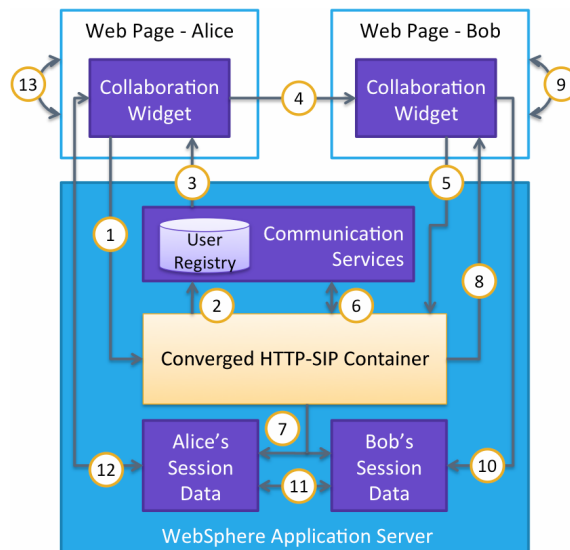


## ***CEA collaboration flow***

This section walks through a usage scenario for sharing session data across a Web link, showing how the CEA components make multi-modal collaboration possible.

## Scenario: Web collaboration (1 of 3)

1. Alice enables collaboration using the widget
2. Container calls services application, putting Alice in the registry and establishing Alice's session data
3. Services application provides Alice with a URI that peers can use to collaborate
4. Alice sends Bob the peer collaboration URI



10

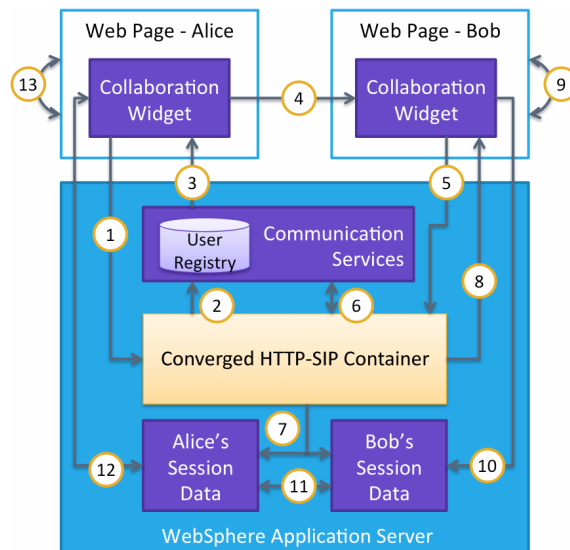
CEA architecture

© 2011 IBM Corporation

In this scenario, Alice is browsing some Web pages, and she wants to ask Bob for help in finding and evaluating products that she's considering for purchase. Since this Web page includes the collaboration widget, Alice begins in step 1 by clicking on the widget to indicate that she wants to begin a shared session with Bob. This request gets sent to the application server and intercepted by the container, which, in step 2, calls into the communication services layer to add Alice to the user registry and establish Alice's session data. In step 3, the server responds to Alice, providing a URI that she can use to collaborate with Bob. Alice needs to provide this URI to Bob so that they are able to collaborate and view the same session information at the same time. So, in step 4, Alice contacts Bob – over e-mail or instant messaging or some other mechanism – to send him the URI.

## Scenario: Web collaboration (2 of 3)

5. Bob loads the peer collaboration URI into his Web browser
6. Container calls services application, putting Bob in the registry and finding Alice in the registry
7. Link is established between Alice and Bob
8. Container sends Bob data exchange URIs (send/fetch); modal windows activate in each widget



11

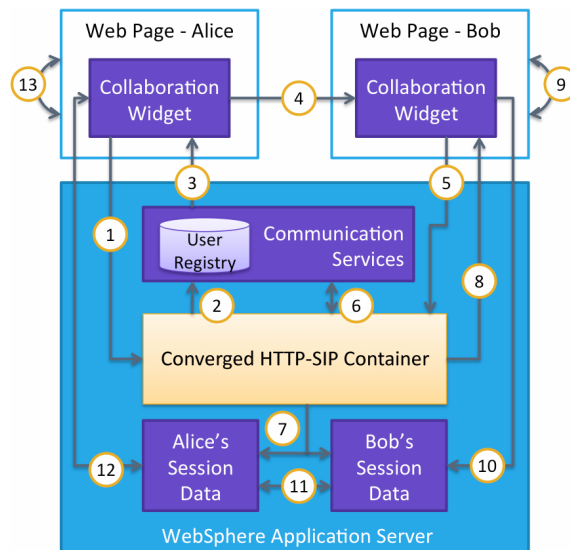
CEA architecture

© 2011 IBM Corporation

Once Bob receives the collaboration URI, he can load it into his browser in step 5. This notifies the container that Bob is trying to join the shared collaboration session with Alice. In step 6, the container interacts with the communication services layer, adding Bob to the user registry, establishing Bob's session data, and finding Alice's session data. Then, in step 7, a link is established between Alice's and Bob's session data. Once this link is established, Bob will receive a response back from the server in step 8. This response includes the send and fetch URIs that are used to exchange data with Alice. At that point, modal windows are activated in both of Alice's and Bob's collaboration widgets

### Scenario: Web collaboration (3 of 3)

9. Bob modifies the page by highlighting text or filling in a form
10. Bob's widget sends events to the application server using the send URI
11. Container sends data to Alice's linked session
12. Alice's widget polls for events on the fetch URI
13. Events from Bob's widget are displayed in Alice's widget



12

CEA architecture

© 2011 IBM Corporation

At this point, the collaboration session is established between Alice and Bob. In step 9, Bob can begin performing actions in the Web page – like highlighting text or filling in a form – to help Alice find what she's looking for. In step 10, Bob's widget sends these events back to the application server on the send URI for collaboration with Alice. When the container receives these updates, it sends the data across to Alice's linked session, like in step 11. Throughout this process, Alice's widget is polling for events coming in on the fetch URI, as shown in step 12. If, while polling for events, Alice's collaboration widget discovers some new event, Alice's widget gets updated with the new events – like seeing the text Bob has highlighted. The polling and updating in steps 12 and 13 continue throughout the collaboration session.

## ***Summary***

This section contains a summary of this presentation.

## Summary

- The WebSphere Application Server Communications Enabled Applications (CEA) feature supports two general types of applications:
  - SIP servlet applications, including SIP servlet 1.1
  - Business applications, through a REST interface that is exposed through CEA widgets or through a Web services interface

WebSphere Application Server supports SIP servlet applications that implement communications function. The Communications Enabled Applications (CEA) feature extends this support to include business applications that require specific communications features. SIP servlet applications can take advantage of the features in the JSR 289 SIP servlet 1.1 specification. Applications that include a Web front end can use the CEA REST interface for communications services and directly embed telephony and collaboration function into Web pages using simple CEA widgets. An application without a Web front end can take advantage of CEA through a Web services interface.



## Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send email feedback:

[mailto:iea@us.ibm.com?subject=Feedback\\_about\\_WASv8\\_CEA\\_Architecture.ppt](mailto:iea@us.ibm.com?subject=Feedback_about_WASv8_CEA_Architecture.ppt)

This module is also available in PDF format at: [../WASv8\\_CEA\\_Architecture.pdf](..WASv8_CEA_Architecture.pdf)

You can help improve the quality of IBM Education Assistant content by providing feedback.



## Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, and WebSphere are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

Java, and all Java-based trademarks and logos are trademarks of Oracle and/or its affiliates.

Other company, product, or service names may be trademarks or service marks of others.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2011. All rights reserved.