Building Secure Mashups With OpenAjax

Jon Ferraiolo
IBM and OpenAjax Alliance

TS-5030

OpenAjax alliance

Java

Sun microsystems

You will learn:

- Mashups - the promise and challenges
- OpenAjax Alliance mashup initiatives

You will see:

- OpenAjax Mashups in action

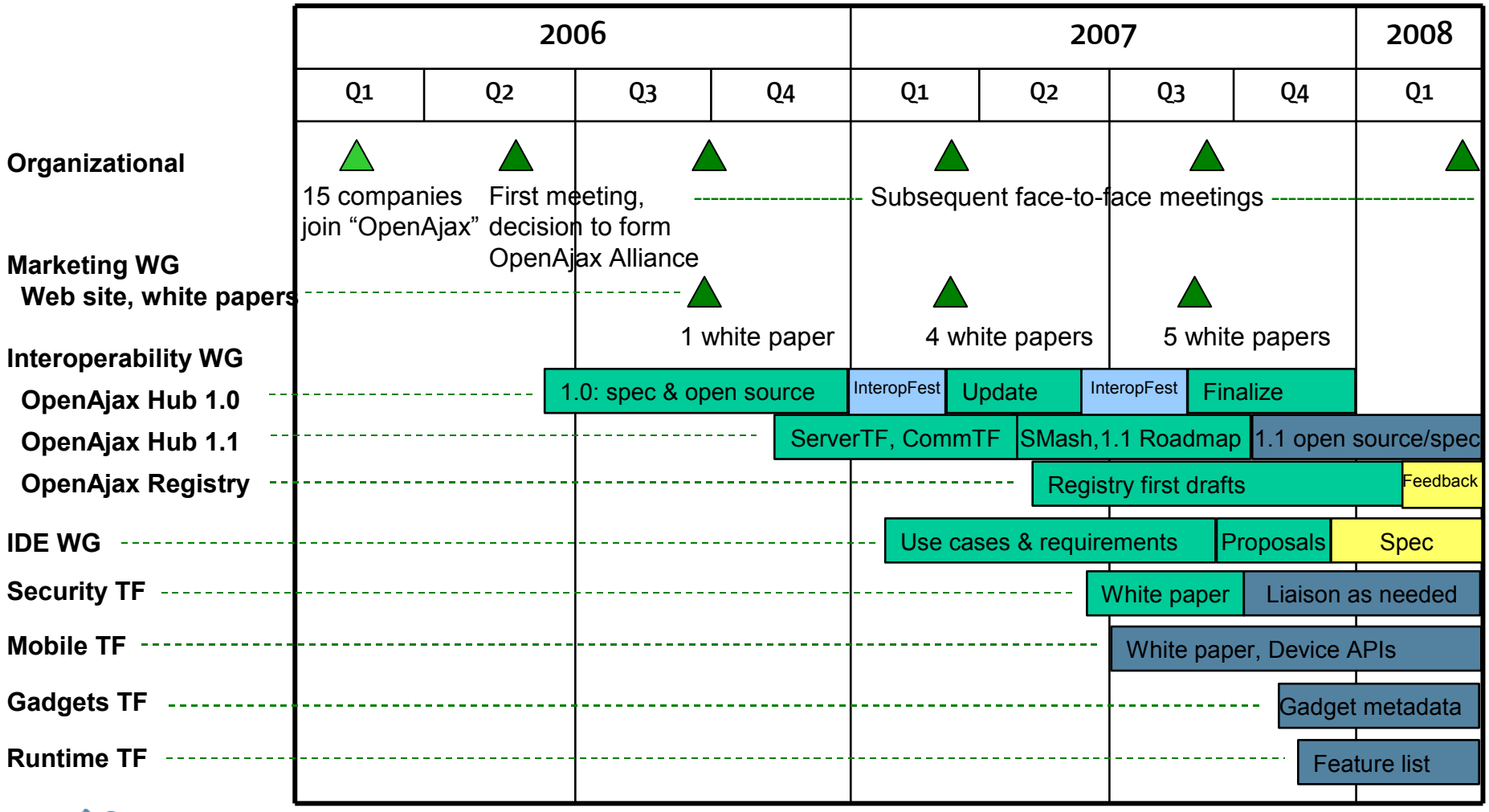
Agenda

- Introducing OpenAjax Alliance
- Secure Mashup Initiatives Overview
- OpenAjax Hub 1.1
- OpenAjax Metadata for Widgets
- Demo
- Summary

Why did the industry form OpenAjax Alliance?

- Interoperability problems across Ajax toolkits
  - Sometimes toolkits step on each other
  - Almost never do toolkits integrate with each other
  - Interoperability/integration is necessary for mashups to work
- Education
  - For IT managers and Web developers, Ajax can be complex and confusing – tyranny of choice
- Help drive the future of the Ajax ecosystem

OpenAjax Alliance – Today



OpenAjax Hub 1.0

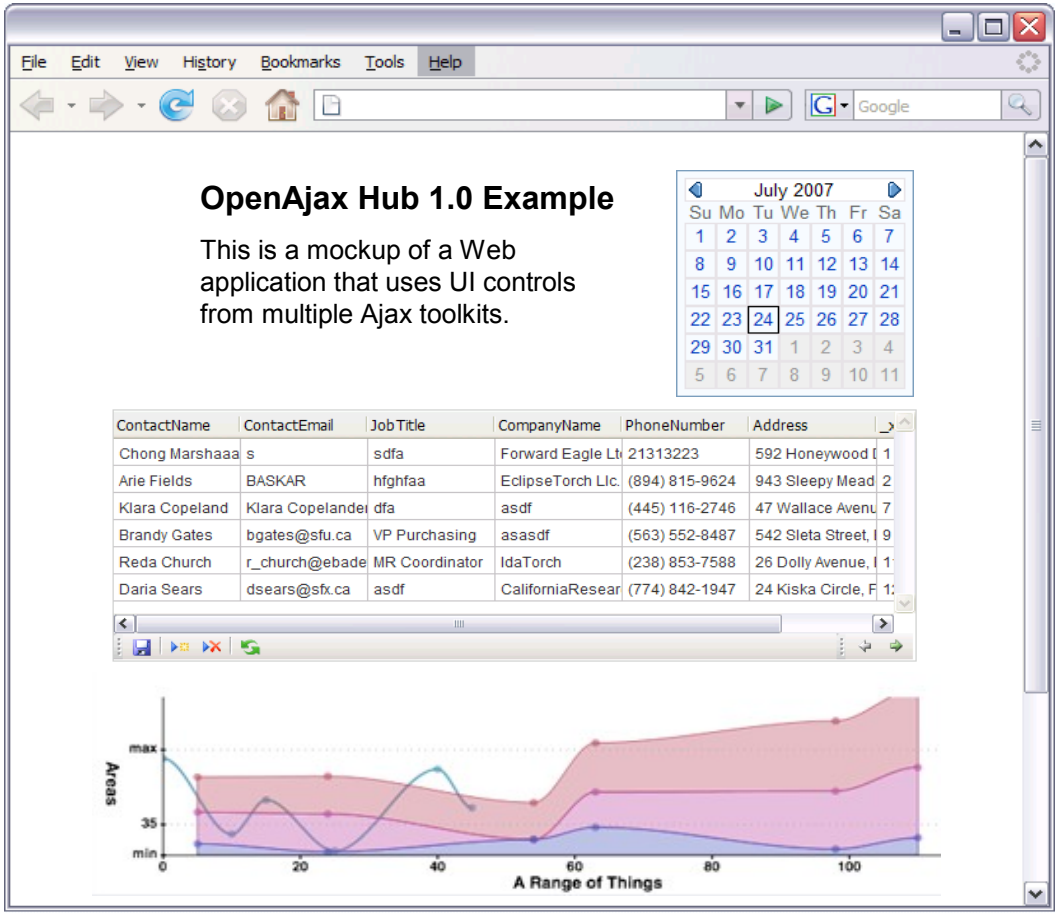
- What is it?
  - Small bit of standard JavaScript™ technology (< 3K after compaction)
  - Enables multiple Ajax runtimes to work together
- Version 1.0 features
  - Ajax library registration
    - OpenAjax.hub.registerLibrary()
  - Simple publish/subscribe engine (the pub sub hub)
    - OpenAjax.hub.publish(topicName, payload)
    - OpenAjax.hub.subscribe(topicName, callbackFunction)

OpenAjax Hub 1.0 – an example

Assume multiple Ajax toolkits:

- **UTILS.js – Various utils, inc. XHR**
- **CALENDAR.js – Calendar control**
- **DATAGRID.js – Powerful tables**
- **CHARTS.js – Charting utilities**

The visual controls need to react to new server data and to each other and update their views appropriately



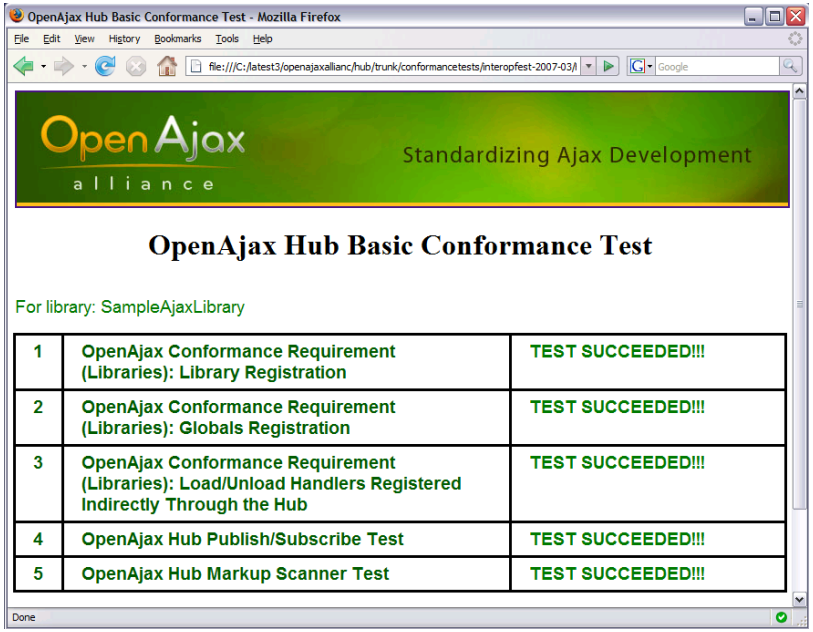
Example – under the hood

```
<html>
  <head>
    <script src="OpenAjax.js"/>
    <script src="UTILS.js"/>
    <script src="CALENDAR.js"/>
    <script src="CHARTS.js"/>
    <script src="DATAGRID.js"/>
    <script>
function MyCalendarCallback(...) {
  OpenAjax.hub.publish("myapp.newdate", newdate);
}
function NewDateCB(eventname, pubData, subData) {…}
OpenAjax.hub.subscribe("myapp.newdate", NewDateCB);
    </script>
  </head>
```

OpenAjax InteropFests

- Objectives:
  - Verify that OpenAjax Hub is reliable, performant, and suitable
  - Allows members to check if they are OpenAjax Conformant

Jan-March 2007



12 toolkits participated
http://www.openajax.org/member/wiki/InteropFest_2007_March

July-Sept 2007

14 organizations, 20 toolkits participated
http://www.openajax.org/member/wiki/InteropFest_1.0

InteropFest Participants

Participating organizations	Participating toolkits
24SevenOffice	AjaxEngine
Apache XAP	Apache XAP
Dojo Foundation	Dojo Toolkit
IBM	Ext
ILOG	ILOG JViews
DWR/Getahead	IT Mill Toolkit
IT Mill	jMaki
Lightstreamer	JQuery
Microsoft	Lightstreamer
Nexaweb	Microsoft Ajax Library
OpenLink SW	Nexaweb Ajax Client
Open Spot	OAT: OpenLink AJAX Toolkit
Software AG	OpenSpot CalcDesk
Sun Microsystems	Prototype
TIBCO	script.aculo.us
	Software AG's webMethods/CAF
	TIBCO General Interface
	24SevenOffice Vili
	YUI

OpenAjax Hub 1.0 status

- Status
  - *Approved*
- Specification
  - http://www.openajax.org/member/wiki/OpenAjax_Hub_1.0_Specification
- Reference implementation at SourceForge
  - http://openajaxallianc.sourceforge.net

Agenda

- Introducing OpenAjax Alliance
- Secure Mashups Overview
- OpenAjax Hub 1.1
- OpenAjax Metadata for Widgets
- Demo
- Summary

Mashups – the self-service business pattern

Business value of mashups

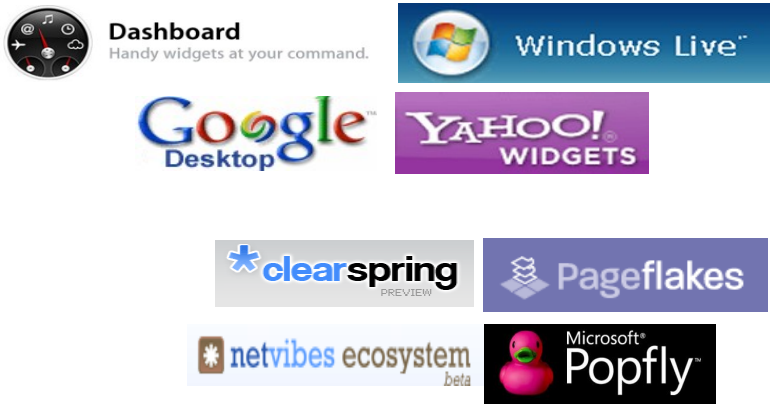
- **Faster, cheaper delivery of applications**
  - Save time and money through reuse and lightweight integration techniques
  - Lower skill set needed to assembly new applications
- **Support innovation and new business opportunities**
  - Users empowered to innovate and explore
- **Gain valuable insights**
  - Due to remixing enterprise and web information
- **Better align IT and business**
  - Do-it-yourself IT will be expected by Facebook generation
- **Extend reach and value of SOA**
  - Service reuse illuminates the business value of SOA

Mashup software

- **Mashup tools**
  - Widget and feed discovery
  - Application assembly
  - Instant deployment
- **Widgets**
  - Pre-packaged, remixable mini-applications
  - Usually tied to a back-end web service
    - Sometimes leveraging previous investment in SOA
  - Public or company-private
  - Key enabler of the long tail

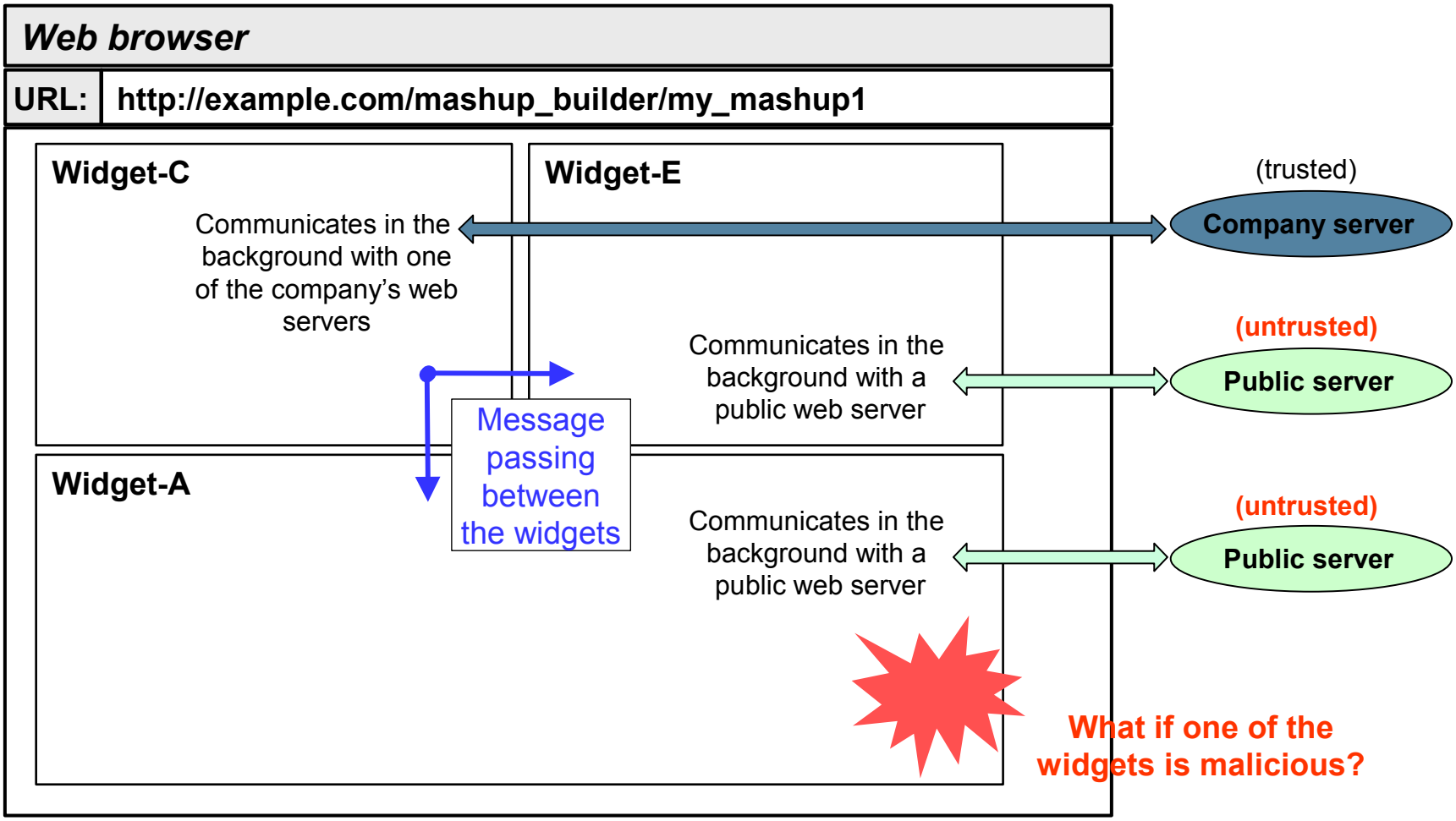
Widget innovation – no shortage

Industry challenges

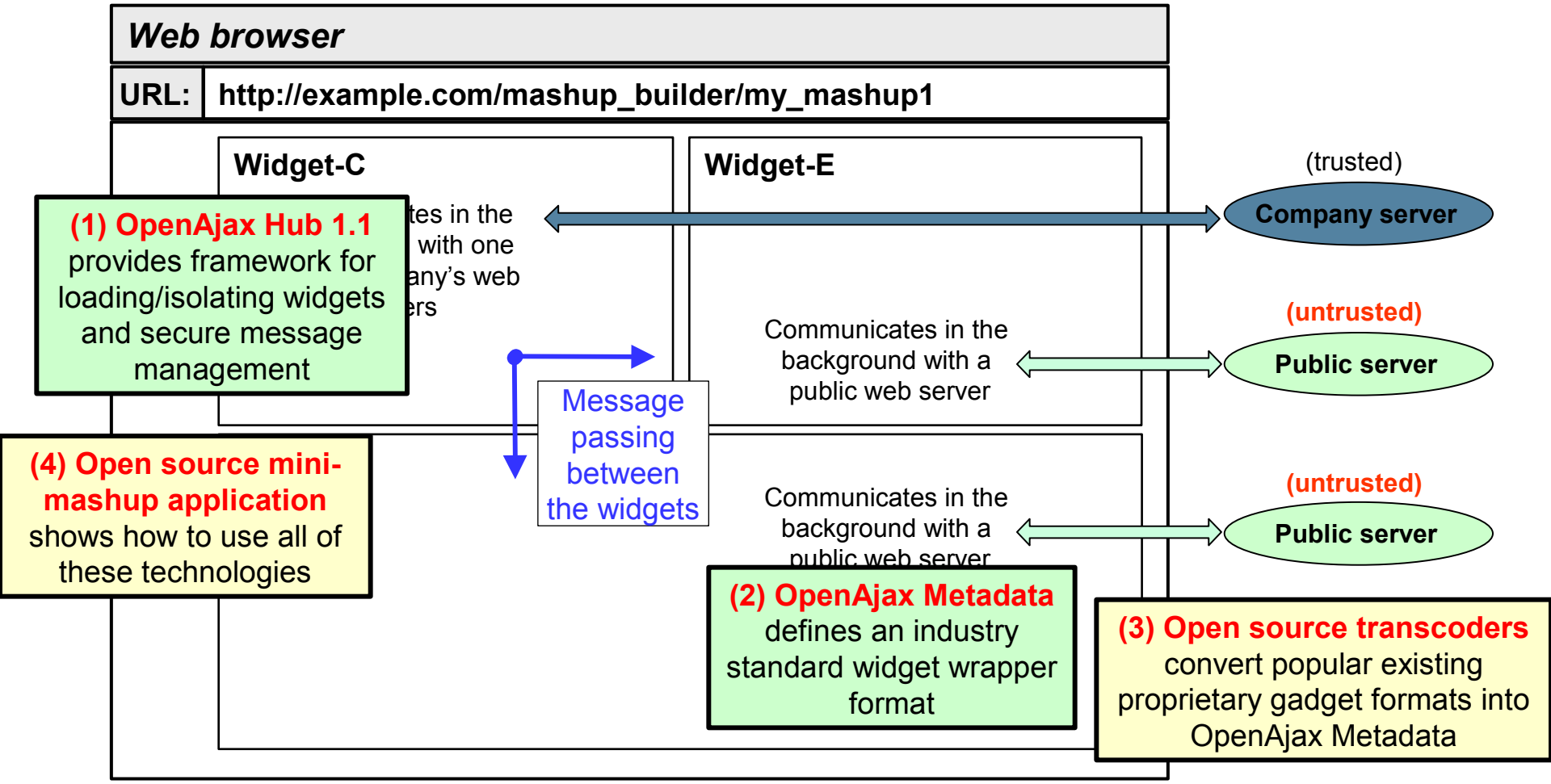


- Interoperability
  - Dozens of proprietary technologies
  - *Good news: many use the "Web Runtime" (i.e., Ajax)!*
  - Bad news: even when using the Web Runtime, widgets are not interoperable
- Security
  - The power of mashups – comes largely from discovering and integrating great widgets from 3rd parties
  - But 3rd party widgets might be malicious

Security vulnerabilities



OpenAjax – Addressing the challenges



Agenda

- Introducing OpenAjax Alliance
- Secure Mashup Initiatives Overview
- OpenAjax Hub 1.1
- OpenAjax Metadata for Widgets
- Demo
- Summary

OpenAjax Hub 1.1 – New features

- OpenAjax Hub 1.0 addresses pub/sub within a single browser frame
- OpenAjax Hub 1.1 adds the following:
  - Pub/sub across frames
  - Framework for secure mashups
  - Pub/sub between clients and servers (i.e., Comet)

Mashups: security issues

- Browser same-origin policy prevents interaction across origins
- Typical Solution: bypass same-origin policy by
  - Dynamic SCRIPT tag to another server (client-side)
  - Proxying content (server-side mashups)
  - “IFrame proxy” (window.location fragment identifier)

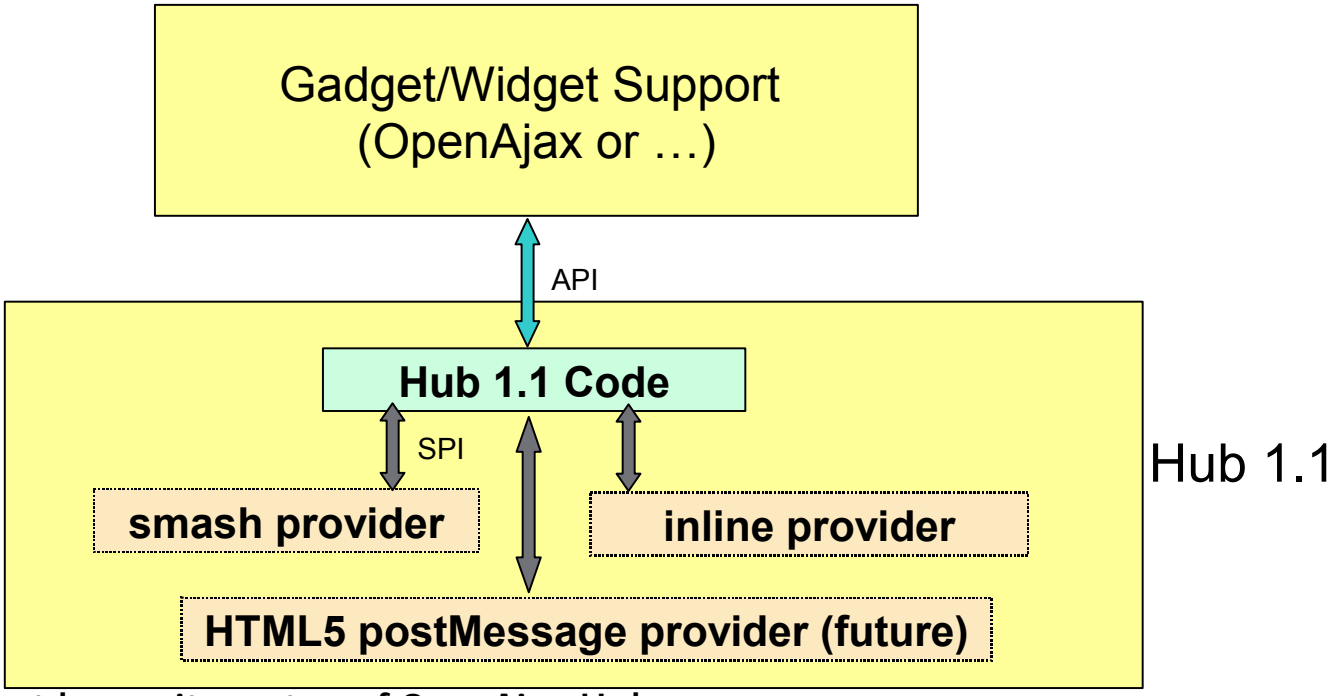
SMash

- SMash stands for “Secure Mashups”
  - Secure handling of 3rd party mashup components
  - Runs in today’s browsers (without plugins)
- Designed and implemented at IBM™ Research (beginning of 2007)
  - Open-sourced (openajaxallianc.sourceforge.net) in August 2007
  - Research Paper describing SMash in WWW 2008 Conference
- High-level APIs, independent of implementation technology
  - Fragment communication, HTML5 postMessage, Java™ platform, Flash etc.
  - Will still work when browsers add native support for secure cross-frame messaging

OpenAjax Hub 1.1: Concepts

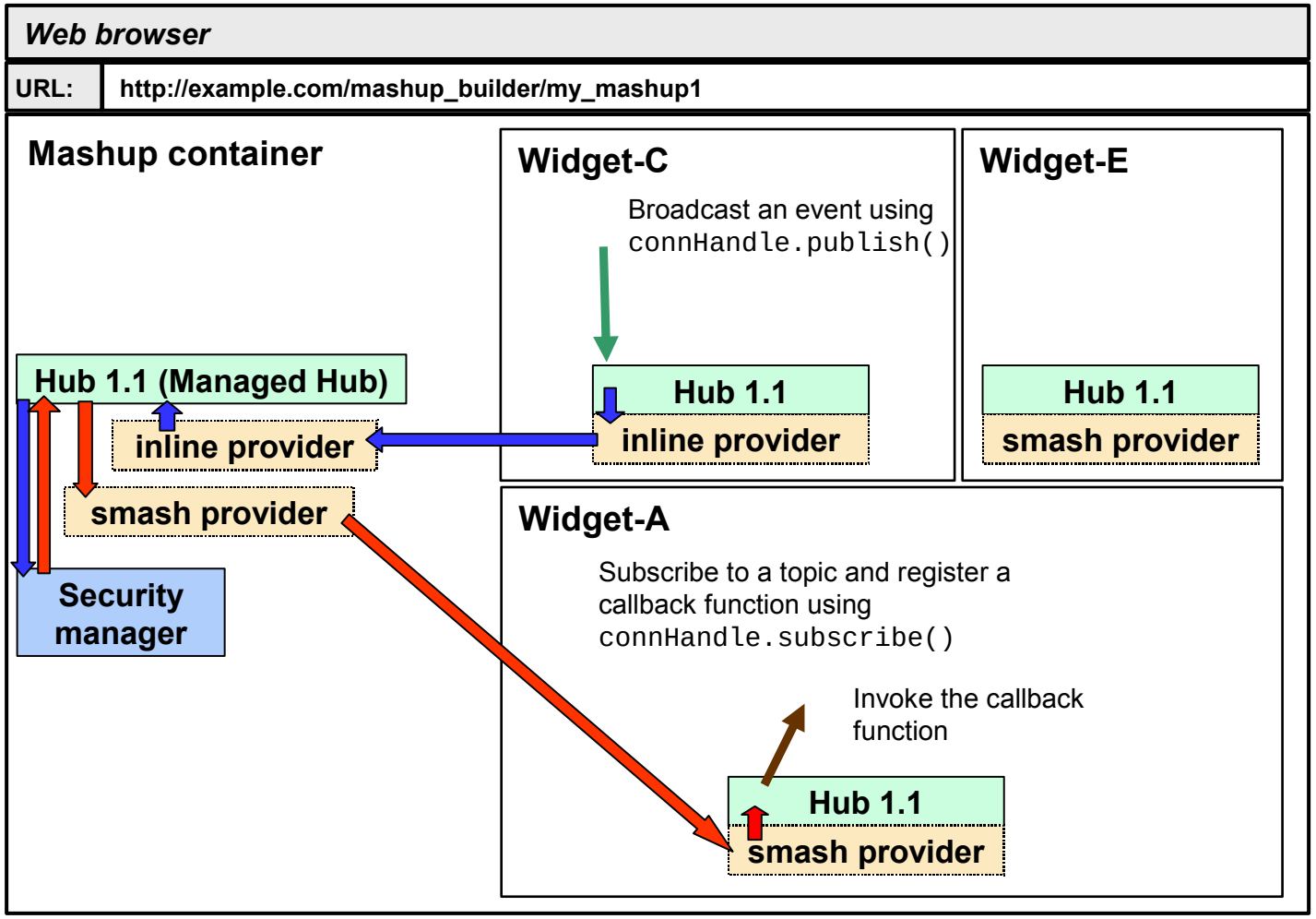
- Managed hub-instances
  - A frame/window can have multiple managed hub-instances
  - Hub-instance has one manager, multiple clients
- Fine-grained policy hooks for manager
  - For security policy, mediation between incompatible clients etc.
  - No policy encoded in hub
- Providers: Multiple communication providers for client to hub-instance communication
  - Provider and Hub SPI
  - Current providers: inline, smash (using code from SMash)

OpenAjax Hub 1.1: Architecture

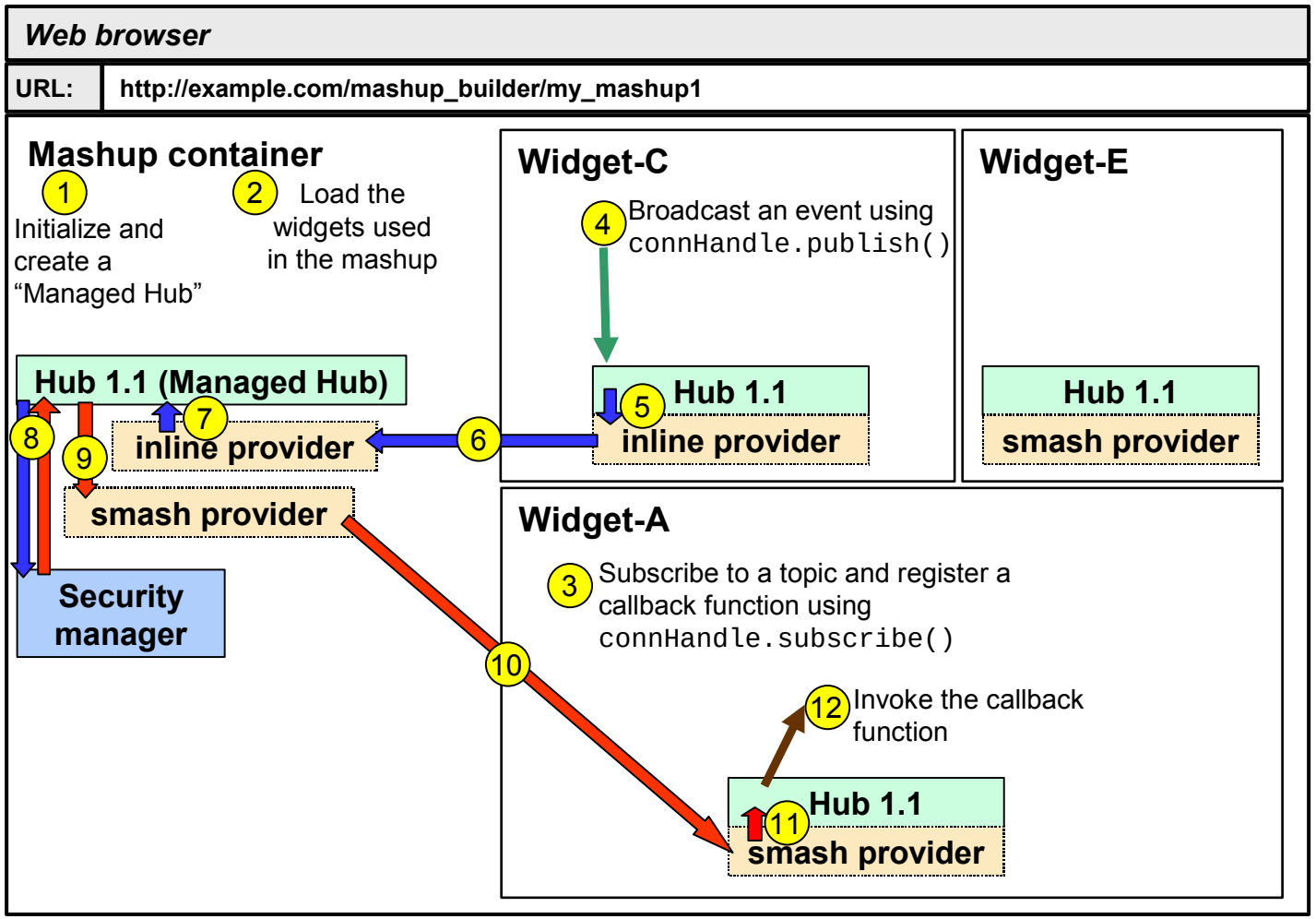


- Gadget/Widget layer sits on top of OpenAjax Hub 1.1
- Hub supports composite gadgets with
  - any level of nesting
  - any combination of gadget types (inline, iframe, …) e.g. inline gadget-foo composed of iframe gadget-bar and inline gadget-baz

OpenAjax Hub 1.1: Simple example



OpenAjax Hub 1.1: the steps



Hub 1.1 status

- Specification
  - First draft spec – far along
  - http://www.openajax.org/member/wiki/OpenAjax_Hub_1.1_Specification
- Reference implementation at SourceForge
  - First implementation (far along)
  - http://openajaxallianc.sourceforge.net
- Timeline for Hub 1.1
  - **Now:** Detailed review within Interoperability Working Group
  - **Spring 2008:** Stable, complete spec
  - **July-September 2008:** InteropFest (with OpenAjax Metadata)
  - **Fall 2008:** Finalize and approve

Agenda

- Introducing OpenAjax Alliance
- Secure Mashup Initiatives Overview
- OpenAjax Hub 1.1
- OpenAjax Metadata for Widgets
- Demo
- Summary

Widget innovation – no shortage

OpenAjax Metadata – industry problems

- IDE interoperability problem
  - Countless Ajax libraries
  - Each library has its own approach to documenting
    - JavaScript APIs
    - UI controls
  - *As a result, difficult to deliver visual authoring tools that integrate with the full set of Ajax libraries in the industry*
- Mashup interoperability problem
  - Dozens of widget formats (Google, Yahoo, Apple, Microsoft…)
  - Current industry situation:
    - Widgets developers provide multiple versions of their widgets
    - To do a mashup, you usually need a programmer
  - *As a result, difficult to deliver visual mashup tools that integrate with the full set of widgets in the industry*

OpenAjax Metadata – what it provides

- (IDE WG) Ajax library metadata "intermediary" standard
  - Standard XML for describing
    - JavaScript APIs
    - Widgets
  - Committee includes
    - Adobe, Aptana, Dojo, Eclipse, IBM, Microsoft, Sun, TIBCO, and Zend
- (Gadgets TF) Mashup gadgets "intermediary" standard
  - Standard XML for describing a mashup component
  - But mashup gadgets have special needs
    - List of topics that they produce and consume (i.e., pub/sub)
    - Security issues related to secure mashups

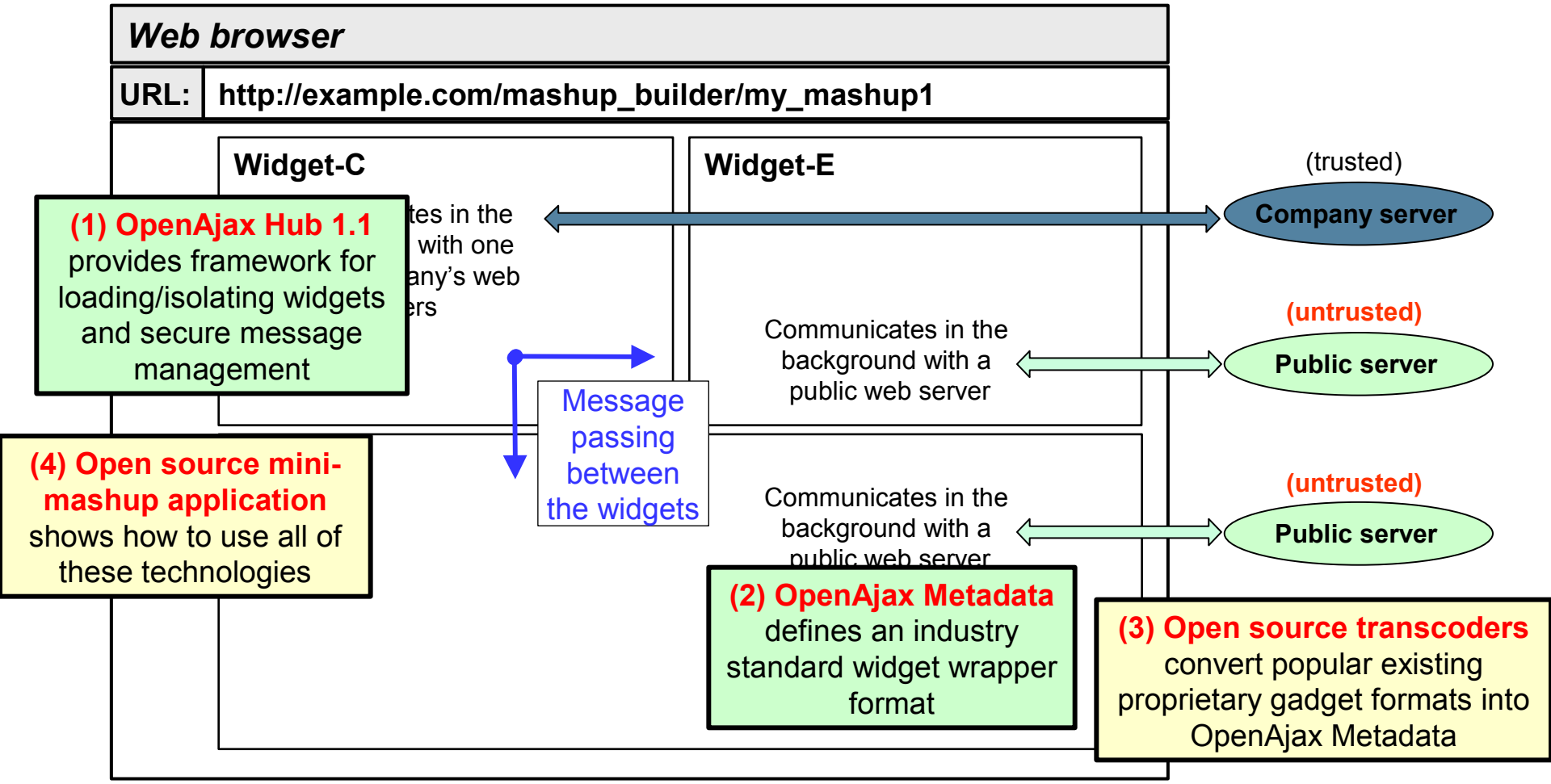
OpenAjax Metadata sample code

```
<widget xmlns="http://ns.openajax.org/widgets">

  <properties>
    <property name="Addr" type="Loc" listen="true" />
  </properties>

  <content>
    <!-- HTML+JavaScript go here -->
  </content>

</widget>
```

OpenAjax Metadata status

- Specification
  - First draft spec - far along
  - http://www.openajax.org/member/wiki/OpenAjax_Metadata_Specification
- Open source
  - Gadget transcoders
  - Mini mashup application
- Timeline for OpenAjax Metadata
  - **Now:** Finishing spec within IDE Working Group
  - **Spring 2008:** Stable, complete spec
  - **July-September 2008:** InteropFest (with OpenAjax Hub 1.1)
  - **Fall 2008:** Finalize and approve

OpenAjax – Addressing the challenges



OpenAjax Mashups in action

Summary

- Mashups offer both promise, but have challenges
  - Security
  - Interoperability
- OpenAjax Alliance is addressing the challenges
  - OpenAjax Hub 1.1
  - OpenAjax Metadata

For More Information

- **Web site:** http://www.openajax.org
- **Wiki:** http://www.openajax.org/member/wiki
- **Blog:** http://www.openajax.org/blog
- **Mail list:** public@openajax.org
- **Email:** Jon Ferraiolo <jferrai@us.ibm.com>

THANK YOU

Jon Ferraiolo, IBM & OpenAjax Alliance

TS-5030