Standardizing Ajax Development

OpenAjax Hub 1.1 & SMash (Secure Mashups)

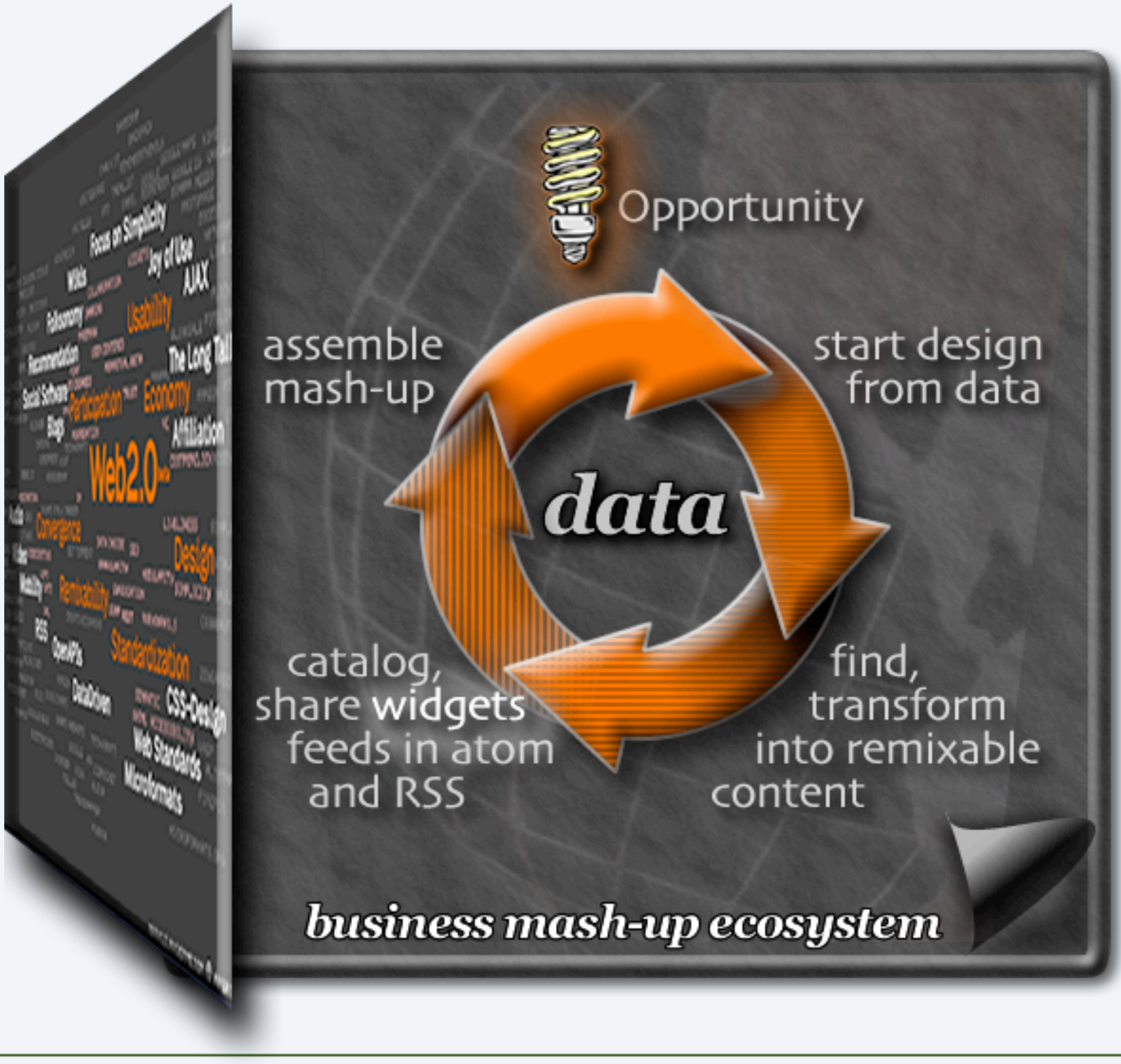
Jon Ferraiolo and Sumeer Bhola
IBM

March 19, 2008

Agenda

- Mash Up Recap
- Introducing OpenAjax Alliance
- OpenAjax Hub1.0
- OpenAjax Hub 1.1 (and SMash)

Reshaping of Enterprise: *emerging "self service" business pattern*



Web 2.0 Content Ecosystem

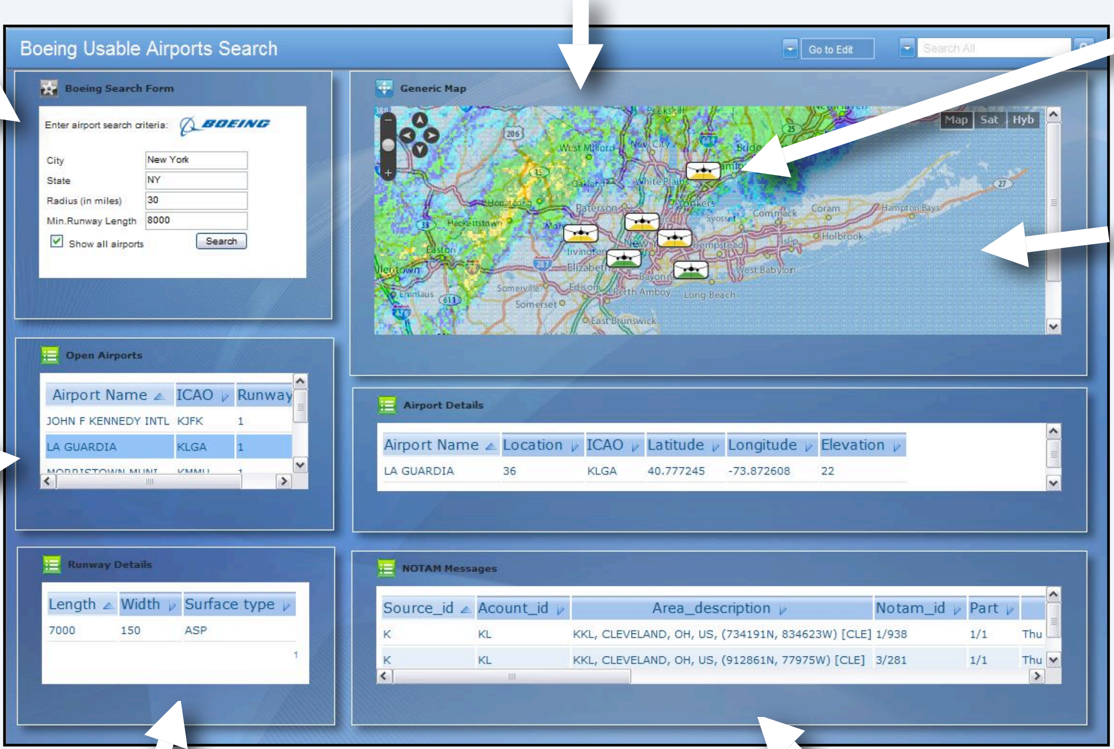
- enterprise *mash-ups* - enabling "web apps" creation by LOBs & subject matter experts
- ease of access to the data that can be combined in different ways to meet ad hoc business opportunities
- designing for *re-mixability*
  - combine data for diverse information services
  - transforming into portable, re-mixable assets & services
  - discover-ability of content both internet & intranet
- exploiting *emergent* business opportunities

NEO Airport Mashup

Operator Queries

Yahoo Map

Airport Location/Status Data (Colored Icons)



Boeing Usable Airports Search

Boeing Search Form

Enter airport search criteria:

City	New York
State	NY
Radius (in miles)	30
Min.Runway Length	8000

☑ Show all airports

Search

Open Airports

Airport Name	ICAO	Runway
JOHN F KENNEDY INTL	KJFK	1
LA GUARDIA	KLGA	1
MORRISTOWN MUNI	KMMU	1

Airport Details

Airport Name	Location	ICAO	Latitude	Longitude	Elevation
LA GUARDIA	36	KLGA	40.777245	-73.872608	22

Runway Details

Length	Width	Surface type
7000	150	ASP

1

NOTAM Messages

Source_id	Acount_id	Area_description	Notam_id	Part	
K	KL	KKL, CLEVELAND, OH, US, (734191N, 834623W) [CLE]	1/938	1/1	Thu
K	KL	KKL, CLEVELAND, OH, US, (912861N, 77975W) [CLE]	3/281	1/1	Thu

Weather Data (Overlay)

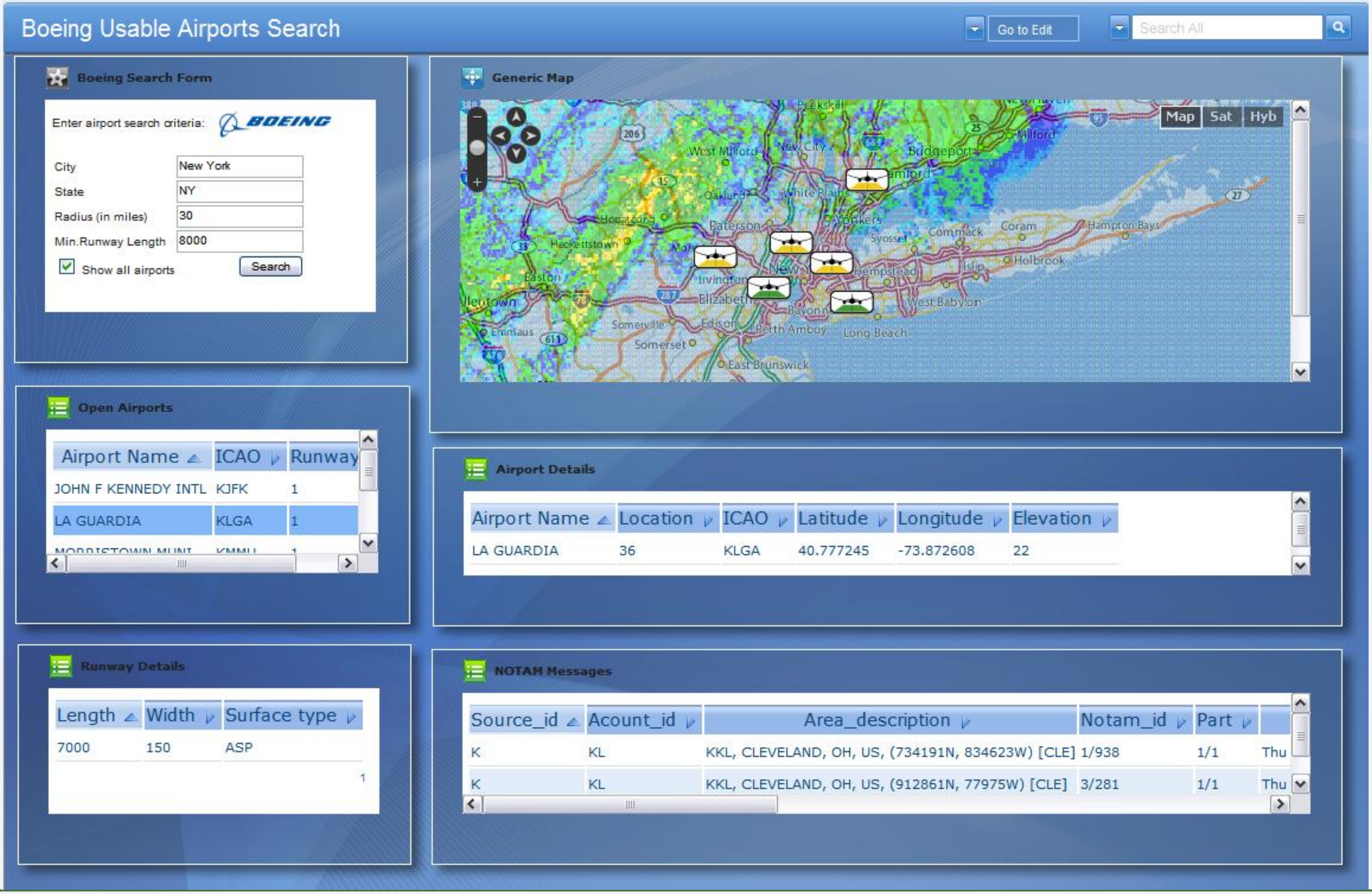
Airport Detail

Airport Data

Runway Data

Alert/Warning (NOTAM) Data

NEO Airport Mashup



Boeing Usable Airports Search

Boeing Search Form

Enter airport search criteria: BOEING

City	New York
State	NY
Radius (in miles)	30
Min.Runway Length	8000

☑ Show all airports Search

Open Airports

Airport Name	ICAO	Runway
JOHN F KENNEDY INTL	KJFK	1
LA GUARDIA	KLGA	1
MORRISTOWN MUNI	KMMU	1

Airport Details

Airport Name	Location	ICAO	Latitude	Longitude	Elevation
LA GUARDIA	36	KLGA	40.777245	-73.872608	22

Runway Details

Length	Width	Surface type
7000	150	ASP

1

NOTAM Messages

Source_id	Acount_id	Area_description	Notam_id	Part	
K	KL	KKL, CLEVELAND, OH, US, (734191N, 834623W) [CLE]	1/938	1/1	Thu
K	KL	KKL, CLEVELAND, OH, US, (912861N, 77975W) [CLE]	3/281	1/1	Thu

Quick history of Ajax

- **Late 1990's to 2001**
  - DHTML (dynamic HTML)
  - IE5 adds XMLHttpRequest
  - Microsoft suspends development of IE
- **2000-2005**
  - Other browsers implement each others' features and quirks, including XMLHttpRequest
- **2003-2005**
  - Pioneering Web developers make use of Ajax techniques
  - Feb 2005: Jesse James Garrett dubs the term "AJAX"

Emergence of Ajax toolkits

- **In beginning, Google (and others) showed the way**
  - Google Suggest, GMail, Google Maps
- **Initial industry skepticism**
  - OK, fine for Google, but too difficult for everyone else
- **But almost immediately, Ajax toolkits emerged**
  - Easy-to-use JavaScript libraries that hide browser dependencies
  - Sometimes with:
    - *Server framework integration (e.g., J2EE/JSF, .NET/ASP)*
    - *IDE integration (~10 Eclipse-based Ajax IDEs, MS Atlas/VS, Dreamweaver)*
    - *Declarative markup language (e.g., Laszlo/LZX, Nexaweb/XAP)*
- **Today: ~200 Ajax toolkits**
  - Often open source
  - Each with their own unique approach and advantages

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**

Agenda

- Introducing OpenAjax Alliance
- OpenAjax Hub1.0
- OpenAjax Hub 1.1 and SMash

OpenAjax Hub 1.0

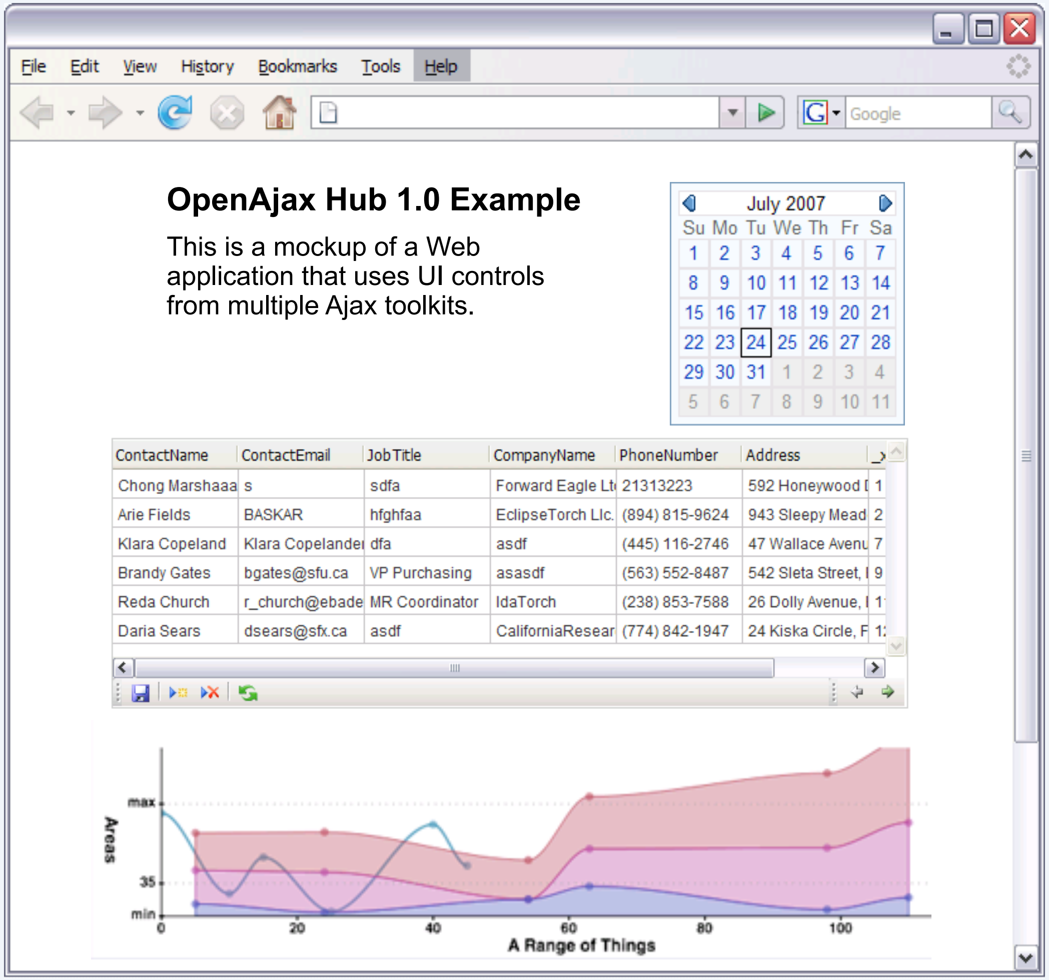
- **What is it?**
  - Small bit of standard JavaScript (< 3K after compaction)
  - Enables multiple Ajax runtimes to work together
- **Version 1.0 features**
  - Ajax library registration
    - *OpenAjax.hub.registerLibrary()*
  - Simple publish/subscribe engine (the pub sub hub)
    - *OpenAjax.hub.publish(topicName,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



OpenAjax Hub 1.0 Example

This is a mockup of a Web application that uses UI controls from multiple Ajax toolkits.

July 2007

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

ContactName	ContactEmail	JobTitle	CompanyName	PhoneNumber	Address	_x
Chong Marshaaa	s	sdfa	Forward Eagle Lt	21313223	592 Honeywood	1
Arie Fields	BASKAR	hfghfaa	EclipseTorch Llc.	(894) 815-9624	943 Sleepy Mead	2
Klara Copeland	Klara Copelande	dfa	asdf	(445) 116-2746	47 Wallace Avenu	7
Brandy Gates	bgates@sfu.ca	VP Purchasing	asasdf	(563) 552-8487	542 Sleta Street,	9
Reda Church	r_church@ebade	MR Coordinator	IdaTorch	(238) 853-7588	26 Dolly Avenue,	1
Daria Sears	dsears@sfx.ca	asdf	CaliforniaResear	(774) 842-1947	24 Kiska Circle, F	1

Example – under the hood

```
<html>
  <head>
    ...
    <script type="text/javascript" src="OpenAjax.js"/>
    <script type="text/javascript" src="UTILS.js"/>
    <script type="text/javascript" src="CALENDAR.js"/>
    <script type="text/javascript" src="CHARTS.js"/>
    <script type="text/javascript" src="DATAGRID.js"/>
    <script type="text/javascript">
     ...
     function MyCalendarCallback(...) {
       OpenAjax.hub.publish("myapp.newdate", newdate);
     }
     ...
     function NewDateCallback(eventname, publisherData, subscriberData) {
       ...update the given visualization widget...
     }
     OpenAjax.hub.subscribe("myapp.newdate", NewDateCallback);
     ...
    </script>
  </head>
  ...
```

Agenda

- **Introducing OpenAjax Alliance**
- **OpenAjax Hub1.0**
- **OpenAjax Hub 1.1 and SMash**
  - Hub 1.1: New features
  - Mashups
    - *Security Issues*
    - *SMash technology overview*
  - Hub 1.1: Details

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 (i.e., integrate work from Security Task Force)
  - Pub/sub between clients and servers (i.e., integrate work from Communications Hub Task Force)

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)

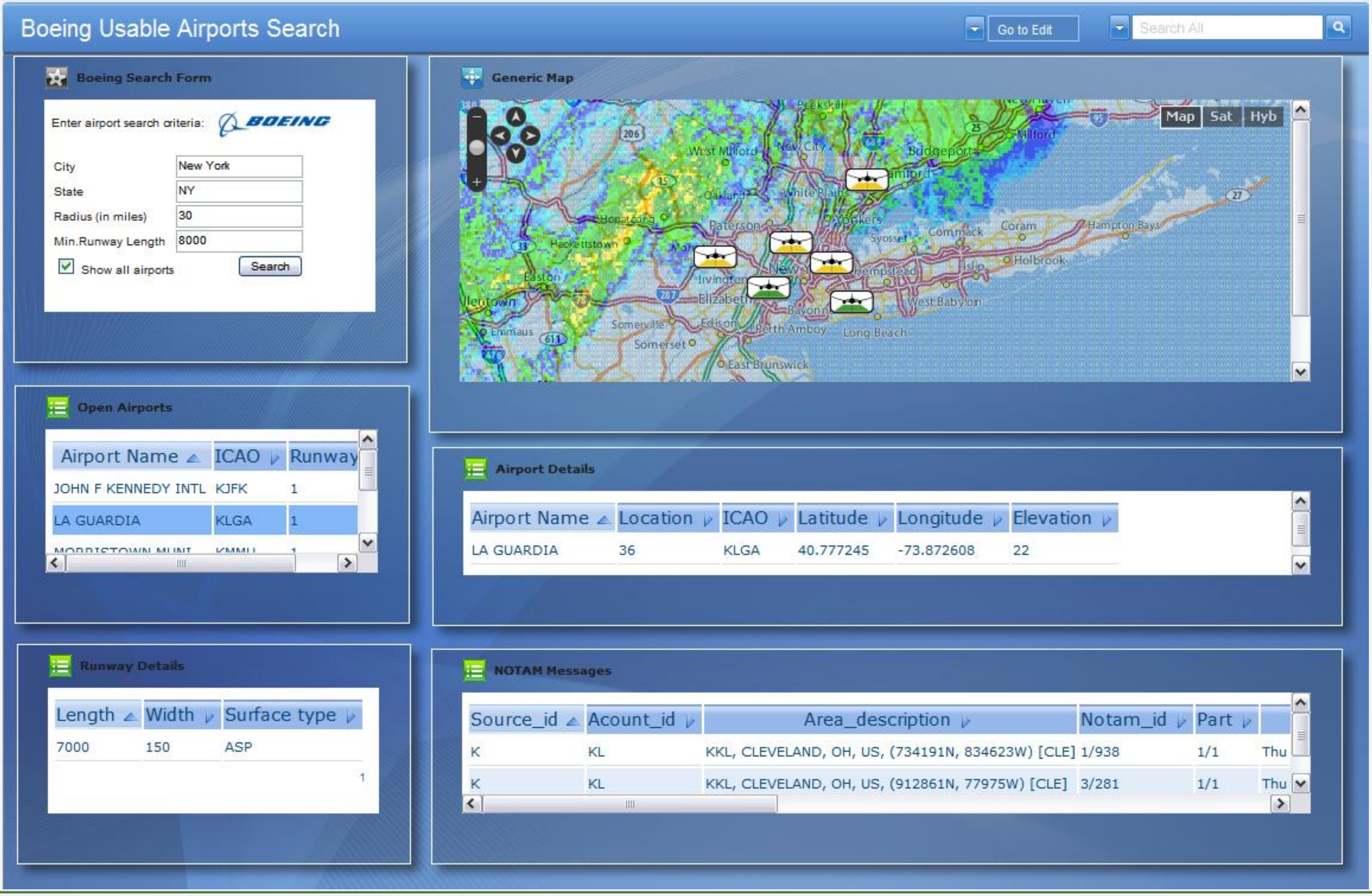
Mashups: security issues

- Browser same-origin policy prevents interaction across origins
- Typical Solution: bypass same-origin policy by
  - Proxying content (server-side mashups)
  - Include scripts from another server (client-side mashups)
- Non-existent security: mixing active content from multiple trust domains

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, Flash etc.
  - Will still work when browsers add native support for secure cross-frame messaging

Security vulnerabilities



Enter airport search criteria:

City	New York
State	NY
Radius (in miles)	30
Min.Runway Length	8000

☑ Show all airports

Airport Name	ICAO	Runway
JOHN F KENNEDY INTL	KJFK	1
LA GUARDIA	KLGA	1
MORRISTOWN MUNI	KMMU	1

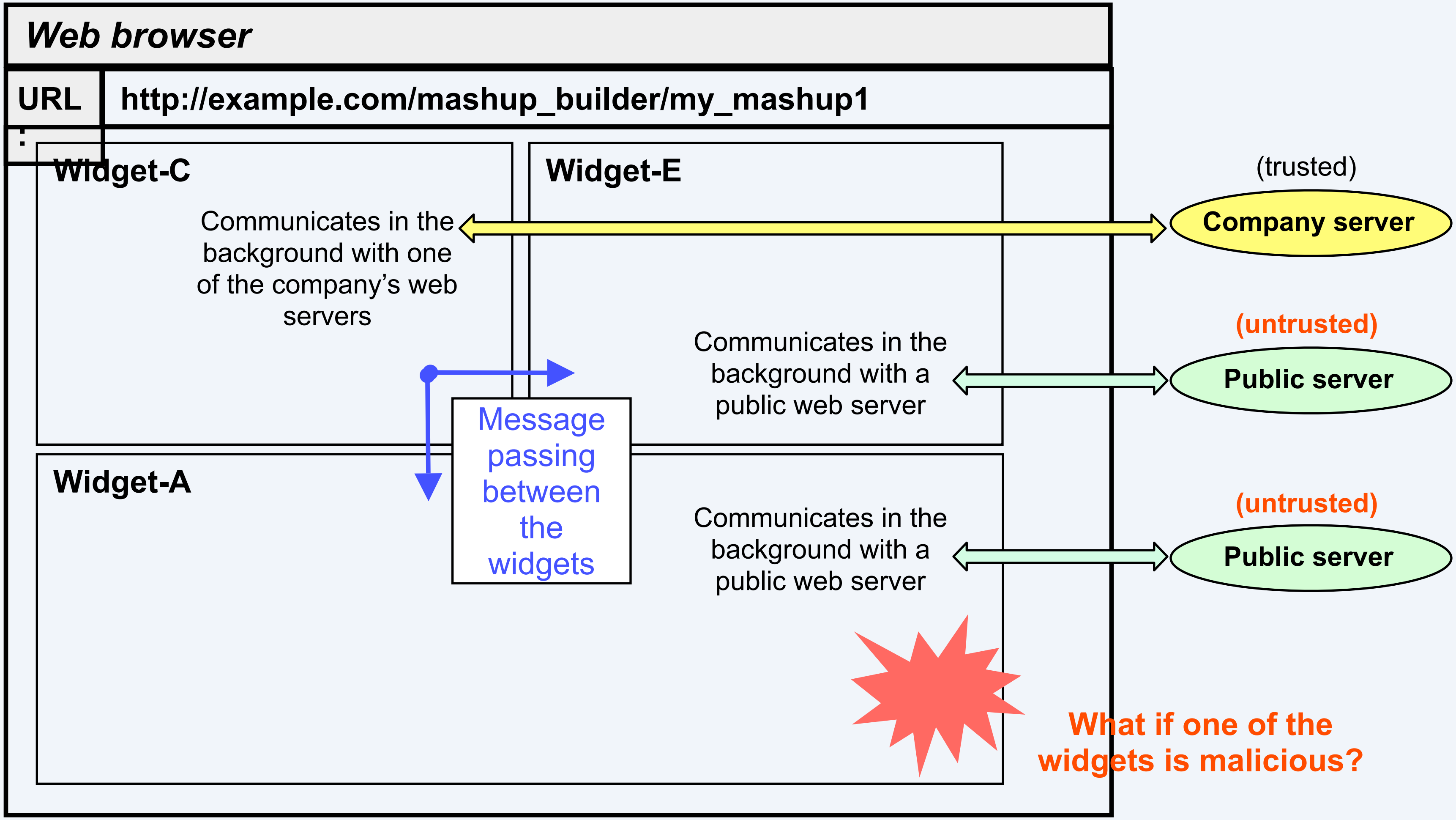
Airport Name	Location	ICAO	Latitude	Longitude	Elevation
LA GUARDIA	36	KLGA	40.777245	-73.872608	22

Length	Width	Surface type
7000	150	ASP

1

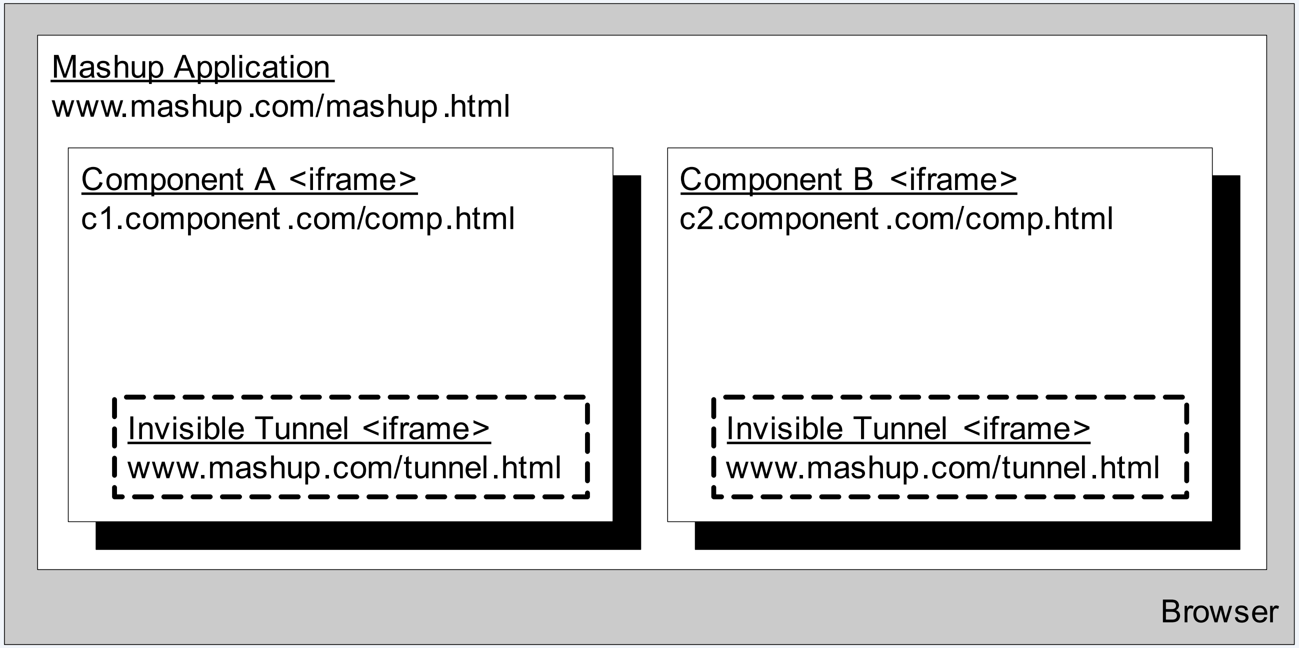
Source_id	Acount_id	Area_description	Notam_id	Part	
K	KL	KKL, CLEVELAND, OH, US, (734191N, 834623W) [CLE]	1/938	1/1	Thu
K	KL	KKL, CLEVELAND, OH, US, (912861N, 77975W) [CLE]	3/281	1/1	Thu

Security vulnerabilities


Web browser
URL
http://example.com/mashup_builder/my_mashup1
:
Widget-C
Communicates in the background with one of the company's web servers
Widget-E
Communicates in the background with a public web server
Widget-A
Communicates in the background with a public web server
Message passing between the widgets
(trusted)
Company server
(untrusted)
Public server
(untrusted)
Public server

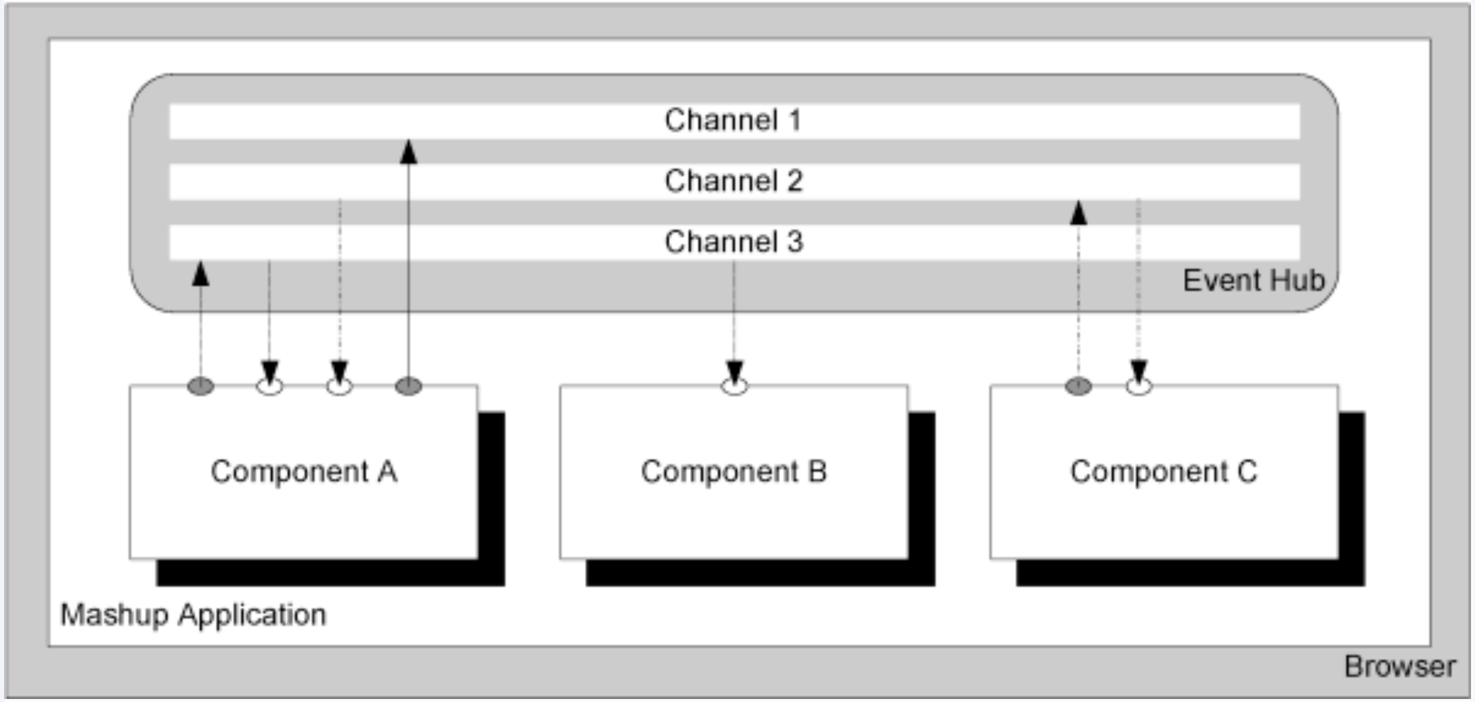
What if one of the widgets is malicious?

SMash: Implementation Approach


Mashup Application
www.mashup.com/mashup.html
Component A <iframe>
c1.component.com/comp.html
Invisible Tunnel <iframe>
www.mashup.com/tunnel.html
Component B <iframe>
c2.component.com/comp.html
Invisible Tunnel <iframe>
www.mashup.com/tunnel.html
Browser

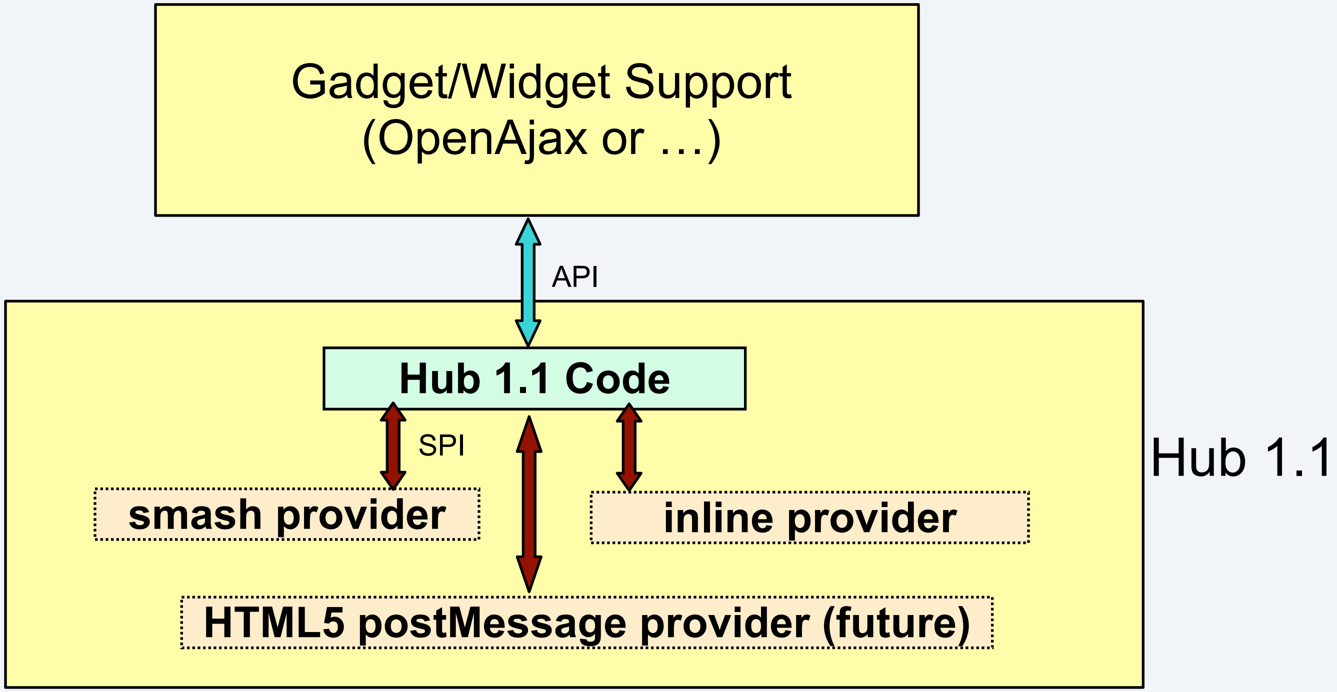
- **Enforcement of component boundaries: Using `frame` isolation and fragment ids for parent-child frame communication**
  - Event Hub implemented by Mashup application
- **Technical challenges addressed by SMash**
  - Enabling communication between frames
  - Integrity of communication and one-way authentication (component to mashup)
  - Frame-Phishing attacks

SMash: Abstractions



- **Isolated browser-side components**
  - A component has named ports: sends/receives messages on its own ports
- **Event hub**
  - Implements (named) channel abstraction to which ports are wired
  - No namespace clashes: port naming is local to a component
  - Security policy specified in component-port wiring

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

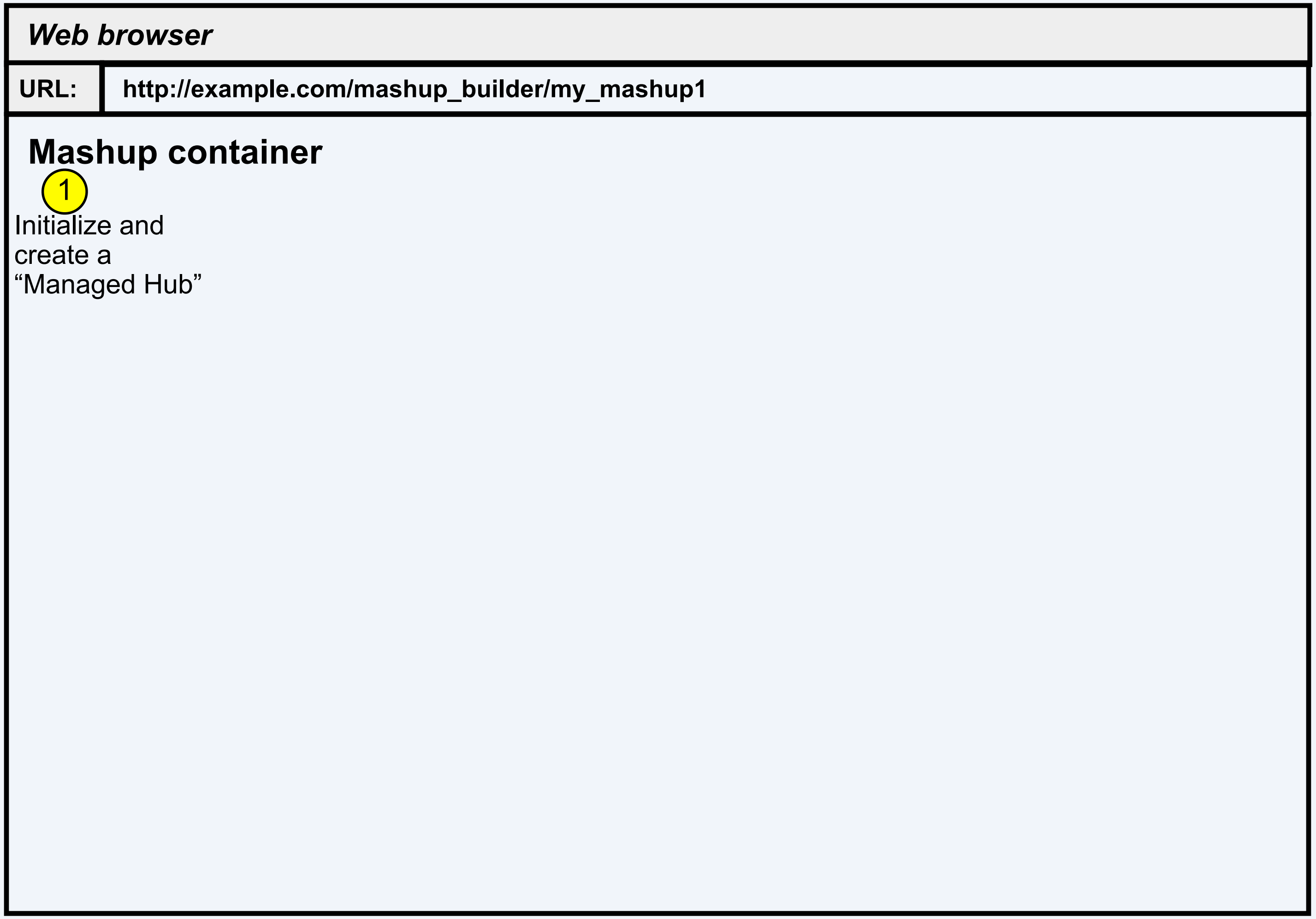
OpenAjax Hub 1.1: the steps

Web browser

URL: http://example.com/mashup_builder/my_mashup1

Mashup container

OpenAjax Hub 1.1: the steps



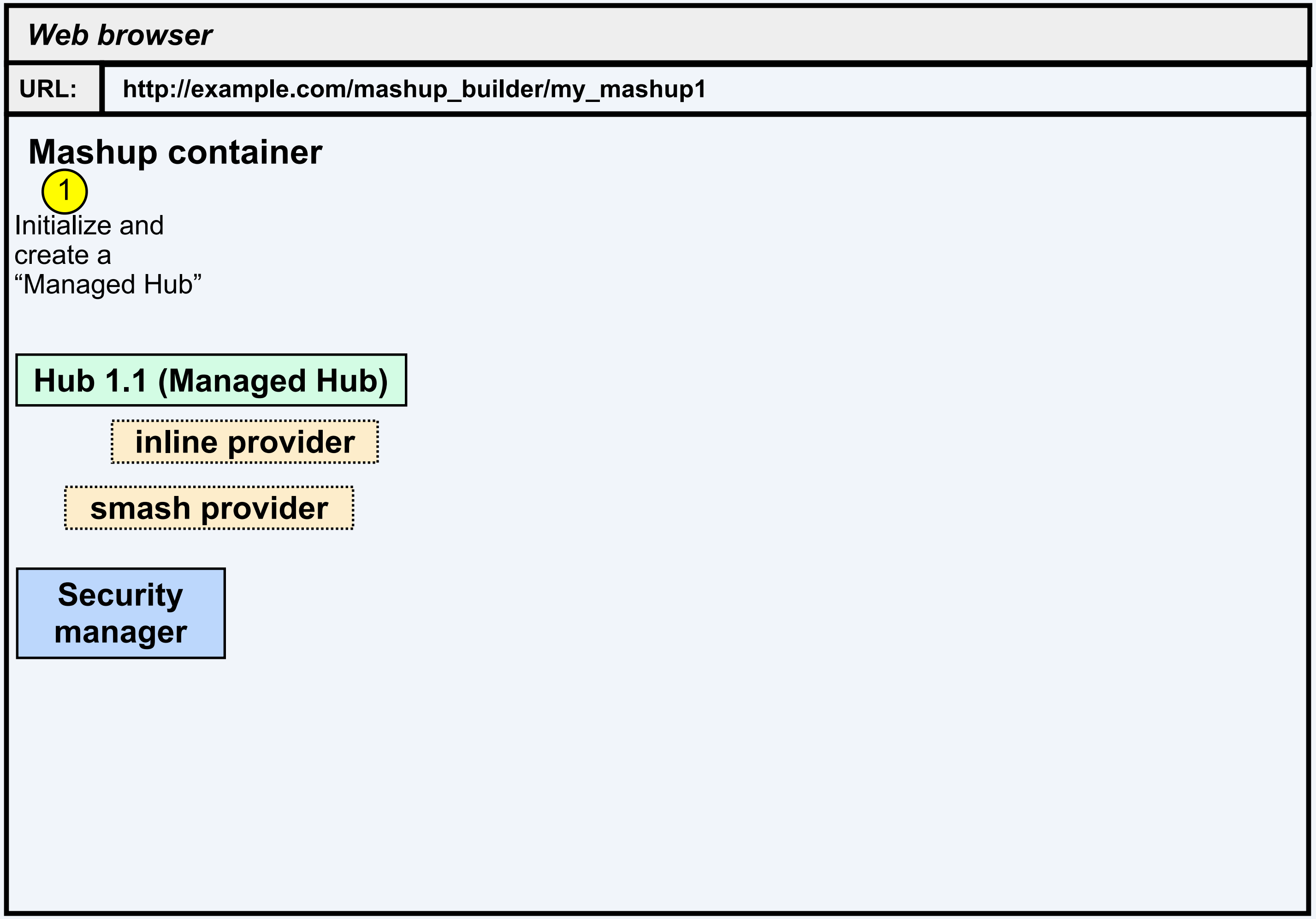
Web browser

URL: http://example.com/mashup_builder/my_mashup1

Mashup container

1 Initialize and create a “Managed Hub”

OpenAjax Hub 1.1: the steps



Web browser

URL: http://example.com/mashup_builder/my_mashup1

Mashup container

1 Initialize and create a “Managed Hub”

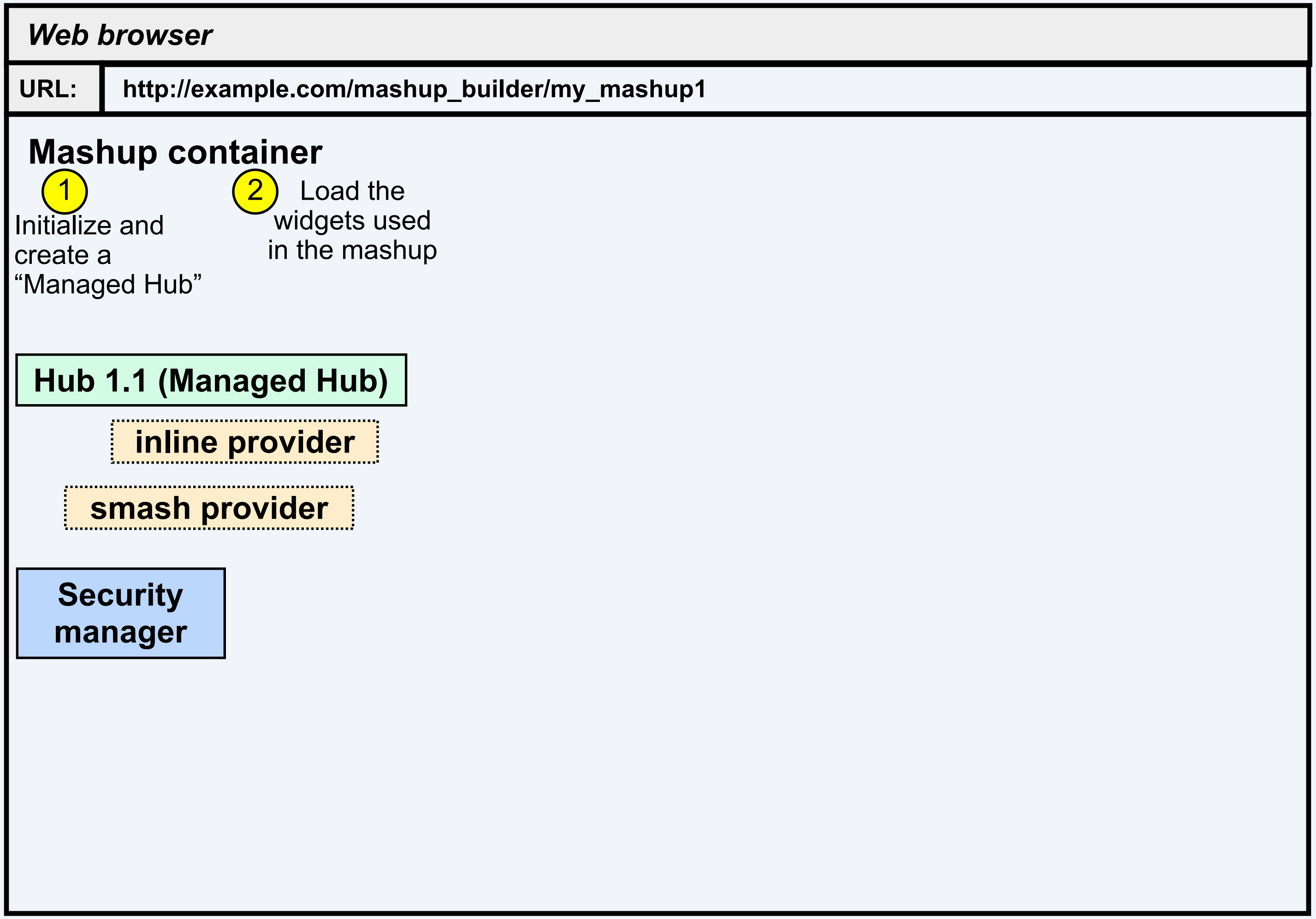
Hub 1.1 (Managed Hub)

inline provider

smash provider

Security manager

OpenAjax Hub 1.1: the steps



Web browser

URL: http://example.com/mashup_builder/my_mashup1

Mashup container

1 Initialize and create a “Managed Hub”

2 Load the widgets used in the mashup

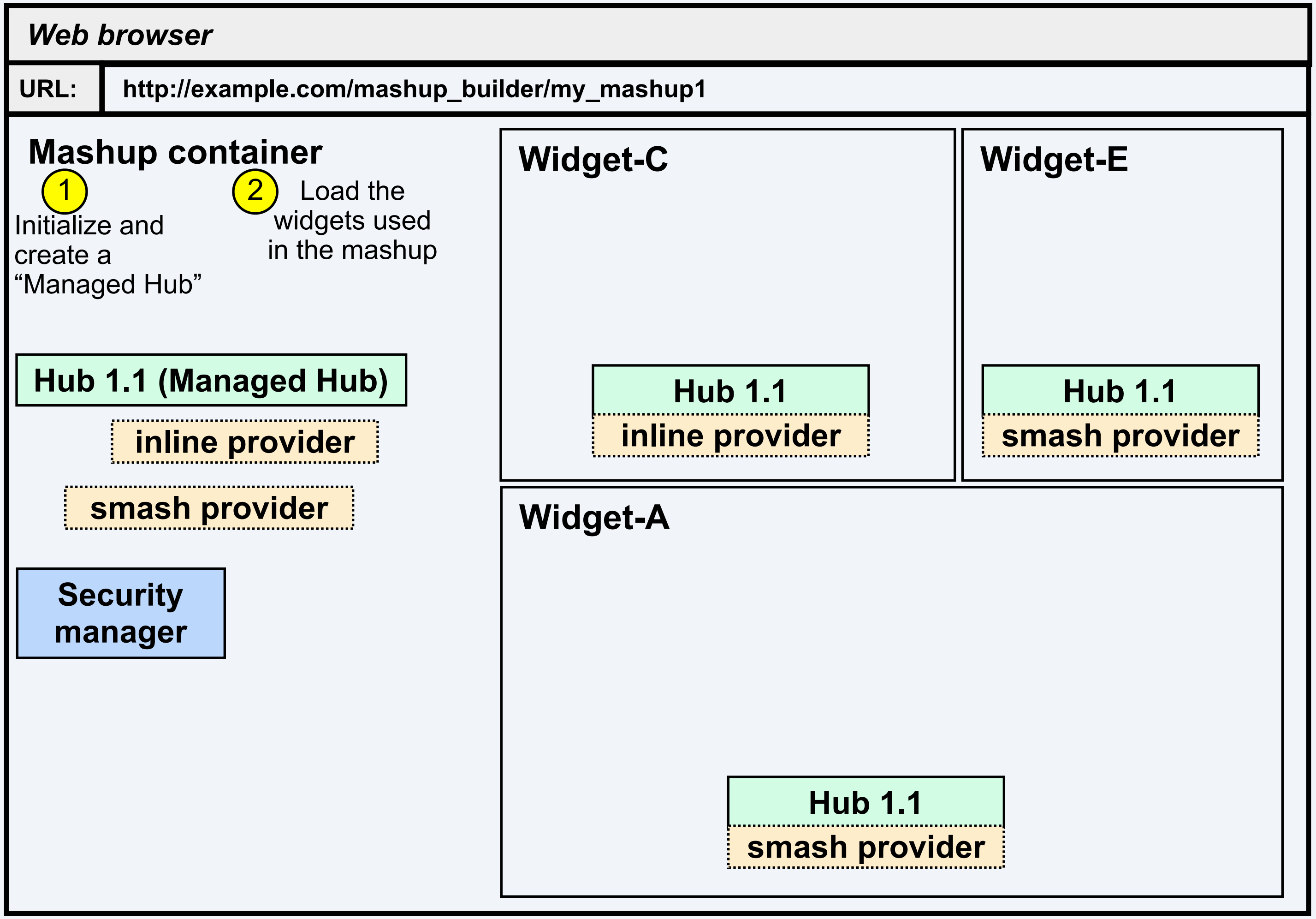
Hub 1.1 (Managed Hub)

inline provider

smash provider

Security manager

OpenAjax Hub 1.1: the steps



Web browser

URL: http://example.com/mashup_builder/my_mashup1

Mashup container

1 Initialize and create a “Managed Hub”

2 Load the widgets used in the mashup

Hub 1.1 (Managed Hub)

inline provider

smash provider

Security manager

Widget-C

Hub 1.1

inline provider

Widget-E

Hub 1.1

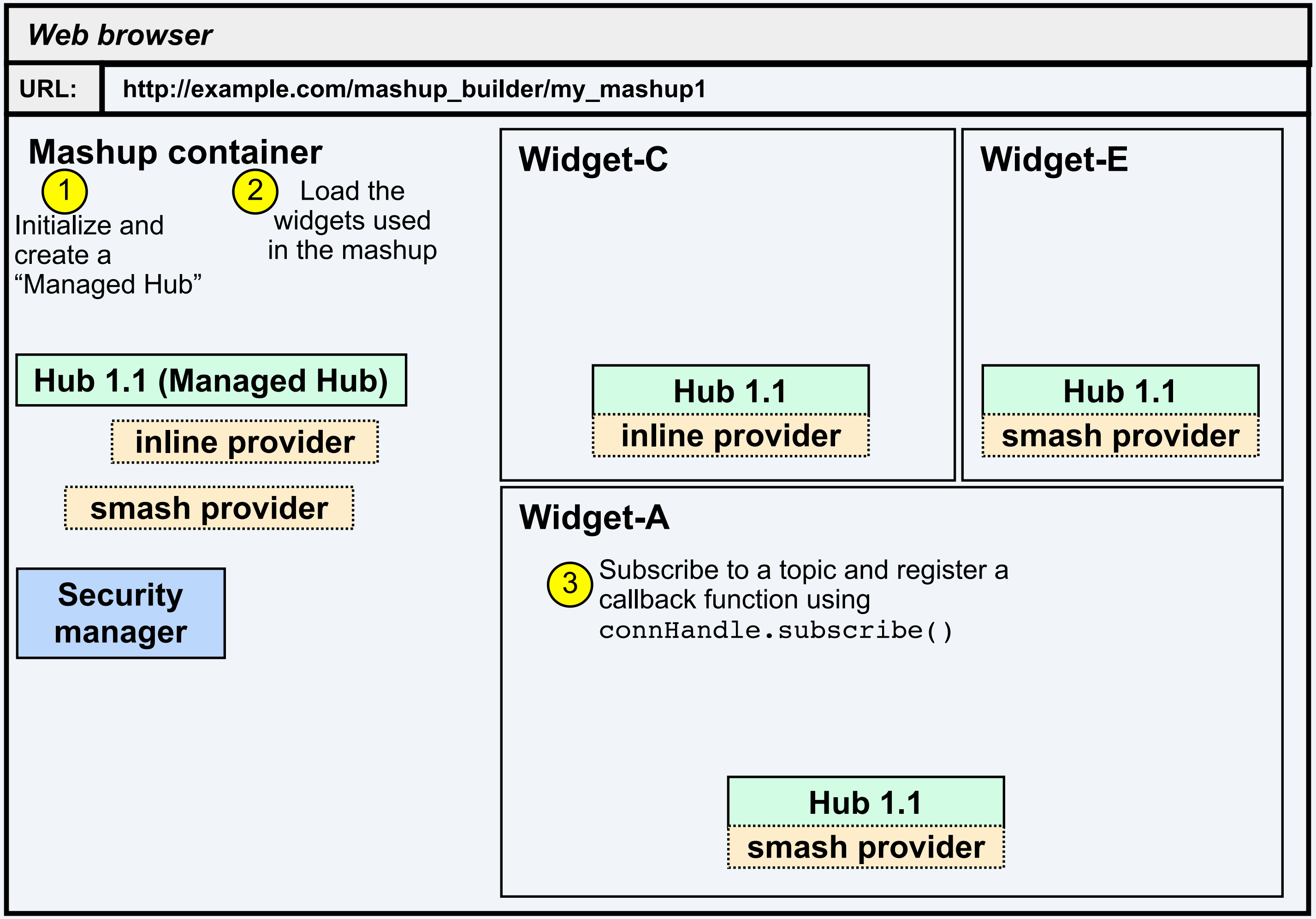
smash provider

Widget-A

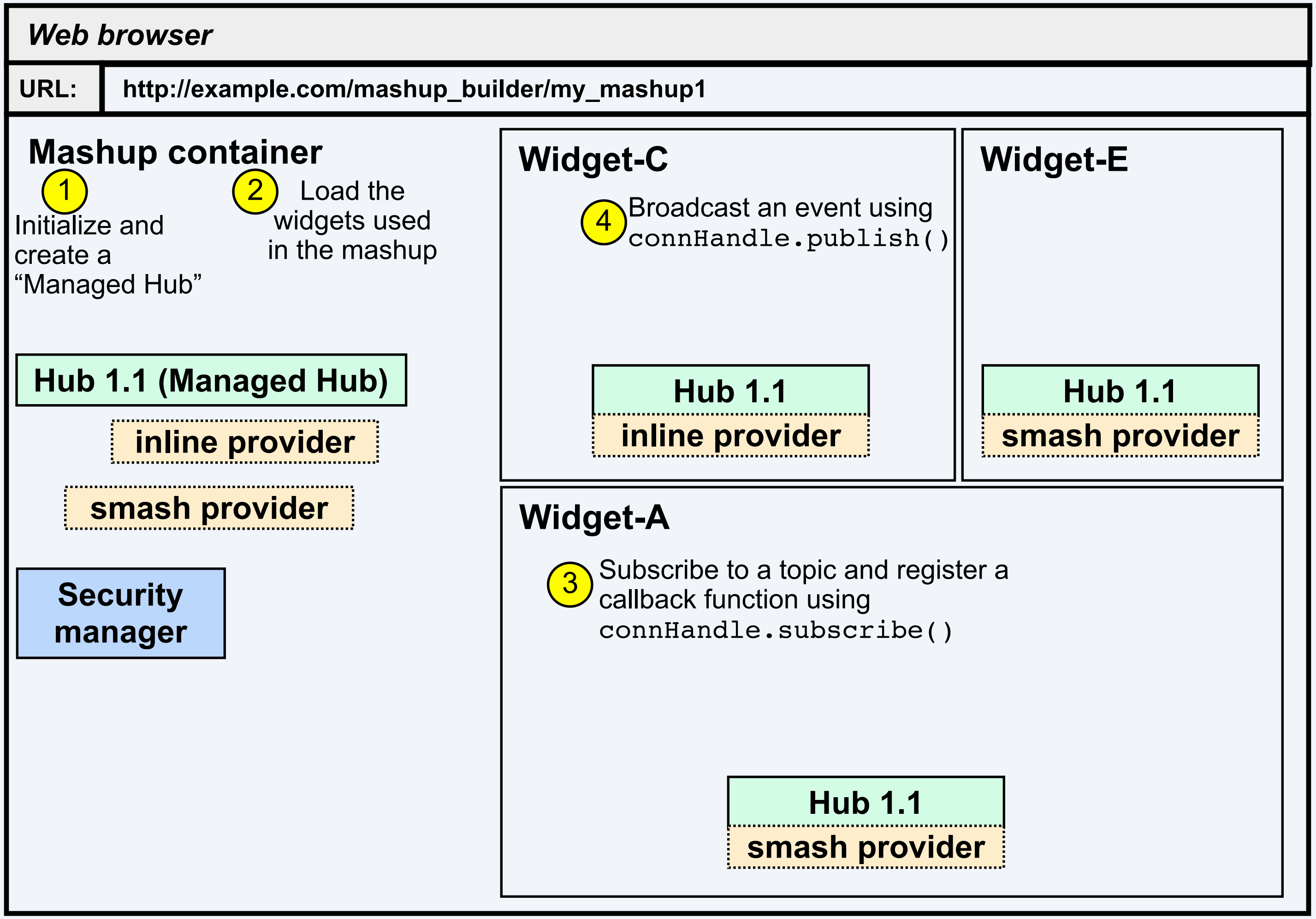
Hub 1.1

smash provider

OpenAjax Hub 1.1: the steps


Web browser
URL:
http://example.com/mashup_builder/my_mashup1
Mashup container
1
Initialize and create a "Managed Hub"
2
Load the widgets used in the mashup
Hub 1.1 (Managed Hub)
inline provider
smash provider
Security manager
Widget-C
Hub 1.1
inline provider
Widget-E
Hub 1.1
smash provider
Widget-A
3
Subscribe to a topic and register a callback function using connHandle.subscribe()
Hub 1.1
smash provider

OpenAjax Hub 1.1: the steps



Web browser

URL: http://example.com/mashup_builder/my_mashup1

Mashup container

1 Initialize and create a "Managed Hub"

2 Load the widgets used in the mashup

Hub 1.1 (Managed Hub)

inline provider

smash provider

Security manager

Widget-C

4 Broadcast an event using `connHandle.publish()`

Hub 1.1

inline provider

Widget-E

Hub 1.1

smash provider

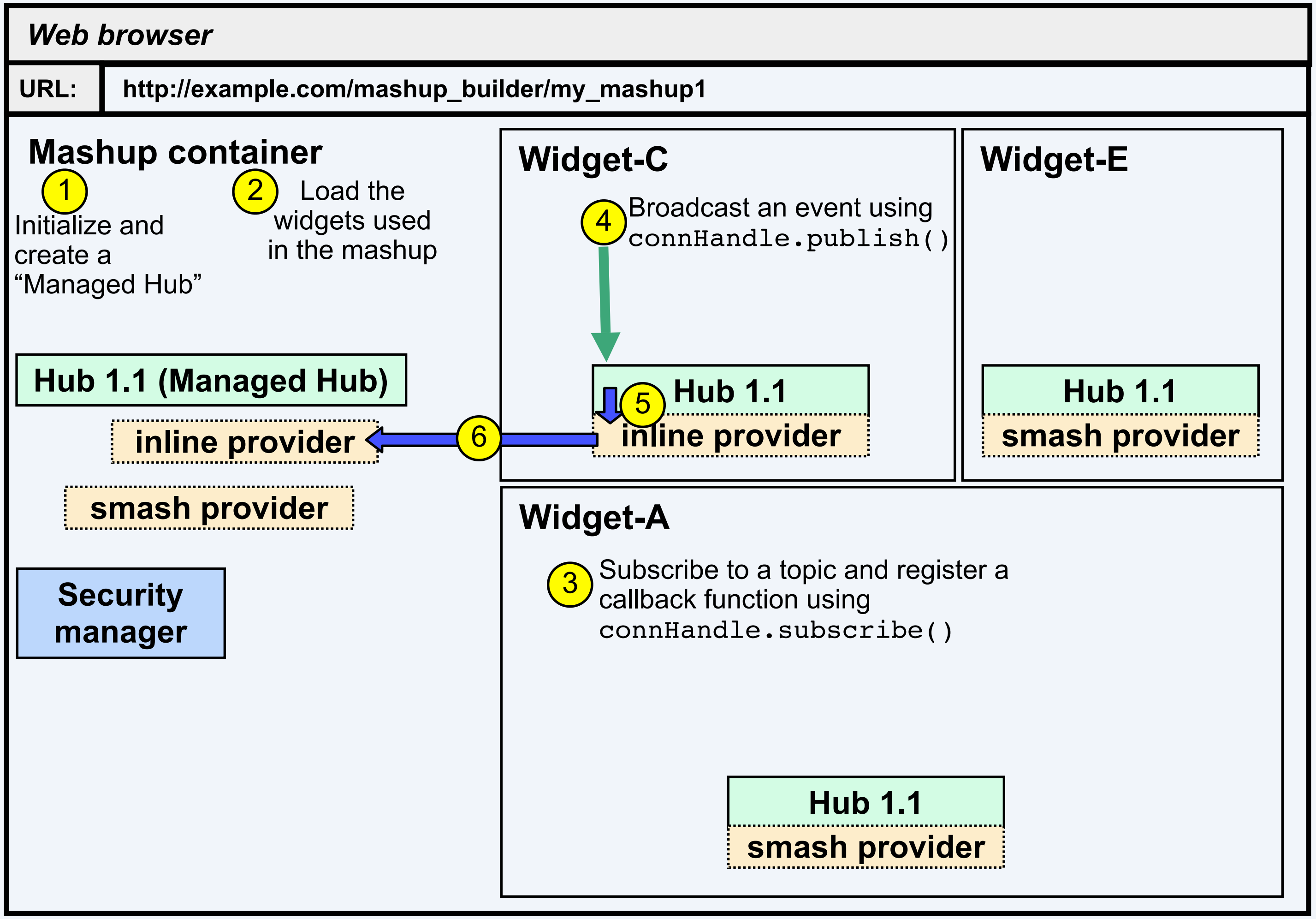
Widget-A

3 Subscribe to a topic and register a callback function using `connHandle.subscribe()`

Hub 1.1

smash provider

OpenAjax Hub 1.1: the steps



Web browser

URL: http://example.com/mashup_builder/my_mashup1

Mashup container

1 Initialize and create a “Managed Hub”

2 Load the widgets used in the mashup

Hub 1.1 (Managed Hub)

inline provider

smash provider

Security manager

6

Widget-C

4 Broadcast an event using `connHandle.publish()`

Hub 1.1

5

inline provider

Widget-E

Hub 1.1

smash provider

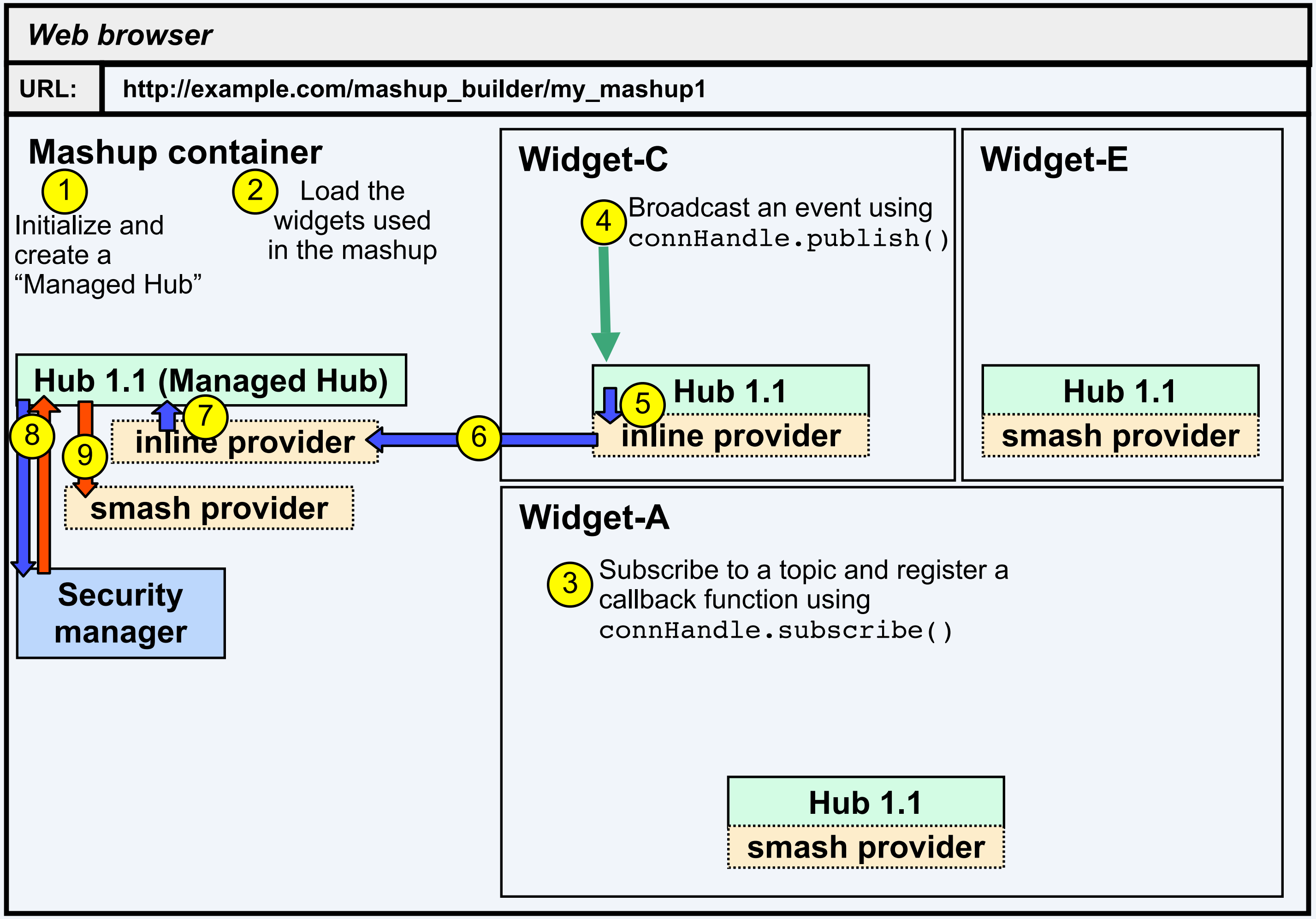
Widget-A

3 Subscribe to a topic and register a callback function using `connHandle.subscribe()`

Hub 1.1

smash provider

OpenAjax Hub 1.1: the steps



Web browser

URL: http://example.com/mashup_builder/my_mashup1

Mashup container

1 Initialize and create a “Managed Hub”

2 Load the widgets used in the mashup

Widget-C

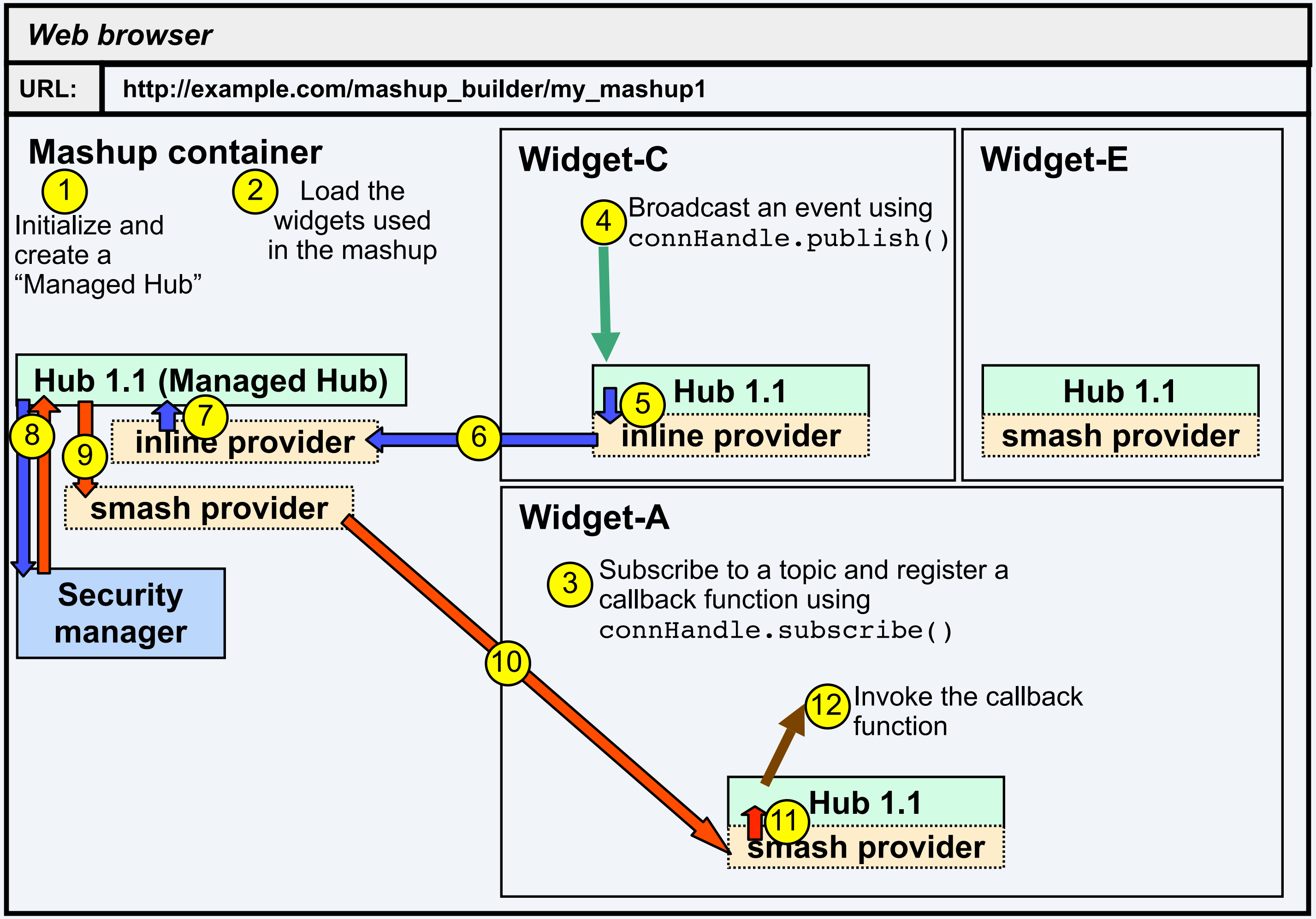
4 Broadcast an event using `connHandle.publish()`

Widget-E

Widget-A

3 Subscribe to a topic and register a callback function using `connHandle.subscribe()`

OpenAjax Hub 1.1: the steps



Web browser

URL: http://example.com/mashup_builder/my_mashup1

Mashup container

1 Initialize and create a "Managed Hub"

2 Load the widgets used in the mashup

Widget-C

4 Broadcast an event using `connHandle.publish()`

Widget-E

Widget-A

3 Subscribe to a topic and register a callback function using `connHandle.subscribe()`

12 Invoke the callback function

Sample Code of Mashup Container

```
/* Create a new hub-instance, and get 'connection handle' */

managedHub = OpenAjax.hub.createManagedHub(pubPolicyCallback,
   subPolicyCallback);

/* Bind Widget-A to the managedHub hub-instance */

managedHub.bind("Widget-A");

/* Setup widget-A to use smash provider */

smash.prepareForLoad({clientName:"Widget-A", uri:"http://
   widgeta.foo.com"});

/* Load widget in its own iframe */

...

/* publish */

managedHub.publish("topic2", {label1:["v1, "v2"]});

/* subscribe */

subscriptionHandle = managedHub.subscribe("topic3",
   successCallback, eventCallback);
```

Sample Code (continued)

- Callbacks to Mashup Container

```
function pubPolicyCallback(topic, data, pubClientName, subClientName) {
  /* Make decision based on topic, and publisher, subscriber identity */
  return true;
}
function subPolicyCallback(topic, subClientName) {
  /* Make decision based on topic and subscriber identity */
  return true;
}
function successCallback(success, subscriptionHandle) {
  if (success) {
    ...
  }
}
function eventCallback(subscriptionHandle, topic, data) {
```

Sample Code of Widget-A

```
hubConnection
  =OpenAjax.hub.connect({clientName:“Widget-A",
  providerName:"http://providers.openajax.org/
  smash", callback:connectCallback})

function connectCallback(success,
  hubConnection) {

  if (!success) { …}

}

…

hubConnection.publish(…)

subscriptionHandle = hubConnection.subscribe(…)
```

Current providers for Hub 1.1

SMash provider

- **Supports client in an iframe communicating with hub-instance in parent frame**
  - Uses some code and ideas from the SMash project
- **Security features:**
  - Mutual authentication based on domain (client to parent, parent to client)
  - Integrity and secrecy of communication between client and parent frame
    - *Integrity based on secret security token generated in browser*
      - Can plugin any cryptographically secure PRNG
      - Defaults to crypto.random (Firefox), Math.random (other browsers – not cryptographically secure)
  - Protection against frame-phishing attacks

Current providers for Hub 1.1

Inline provider

- For manager and client sharing the same frame
- Mutually trusting so no security issues

postMessage (HTML5) provider (forthcoming)

- Uses postMessage inter-frame communication API supported by future browsers
- Currently supported in Opera 8
- Will be in Firefox 3, IE 8 and (next version of) Safari

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
- Process
  - Lead with open source
  - Get mashups features working first, then add communications features
- 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

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>**

Backup Slides

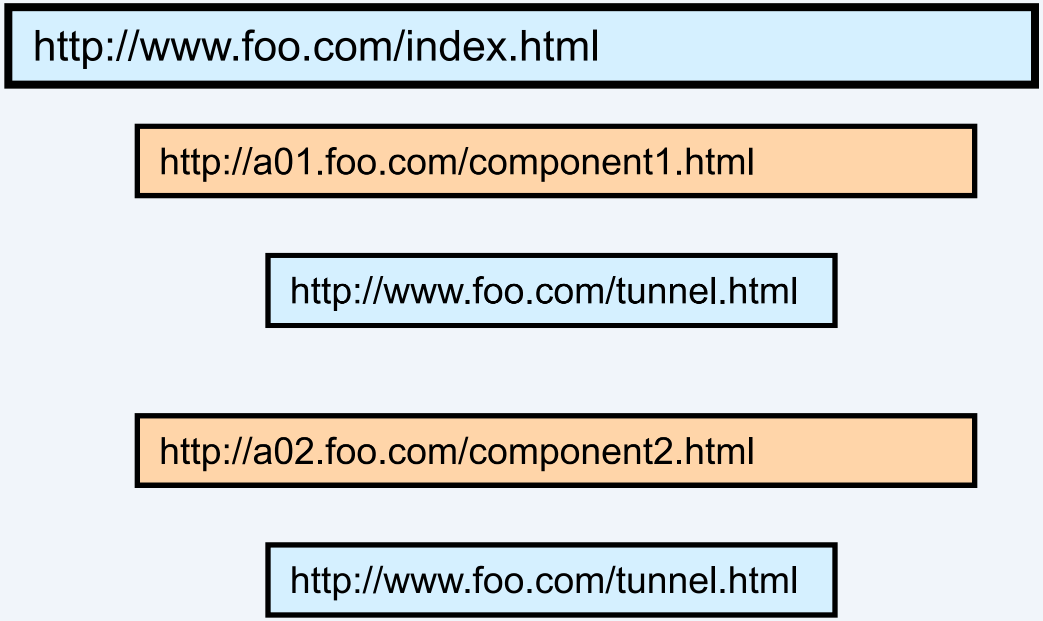
How SMash works

Web site layout:

./index.html
./component1.html
./component2.html
./tunnel.html

/etc/hosts changes

```
127.0.0.1  mashup.foo.com
127.0.0.1  a01.foo.com
127.0.0.1  a02.foo.com
127.0.0.1  a03.foo.com
127.0.0.1  a04.foo.com
etc.
```



http://www.foo.com/index.html

http://a01.foo.com/component1.html

http://www.foo.com/tunnel.html

http://a02.foo.com/component2.html

http://www.foo.com/tunnel.html

Mashup application (index.html):

- Manages instantiation of all mashup components
- Manages all cross-frame communications

Mashups

Enterprise Content
User Generated Content
MashupHub
All Items
Feeds
Widgets
Categories:
Tags:
Drag & Drop Feeds & widgets