Integration Gets All Mashed Up: Bridging Web 1.0 and Web 2.0 Applications

Andreas Krohn

Product Manager
Kapow Technologies
www.kapowtech.com

Session TS-8544

Integration Gets All Mashed Up

Bridging Web 1.0 and Web 2.0 applications

Why are Mashups something to consider, what are the technologies involved, and how does it compare to what you do today?

Agenda

Mashup: The New Kid on the Block

Enabling Technologies

The Components of a Mashup

The Mashup Programming Model

Life of a Developer in a Mashup World

Demo

Agenda

Mashup: The New Kid on the Block

Enabling Technologies

The Components of a Mashup

The Mashup Programming Model

Life of a Developer in a Mashup World

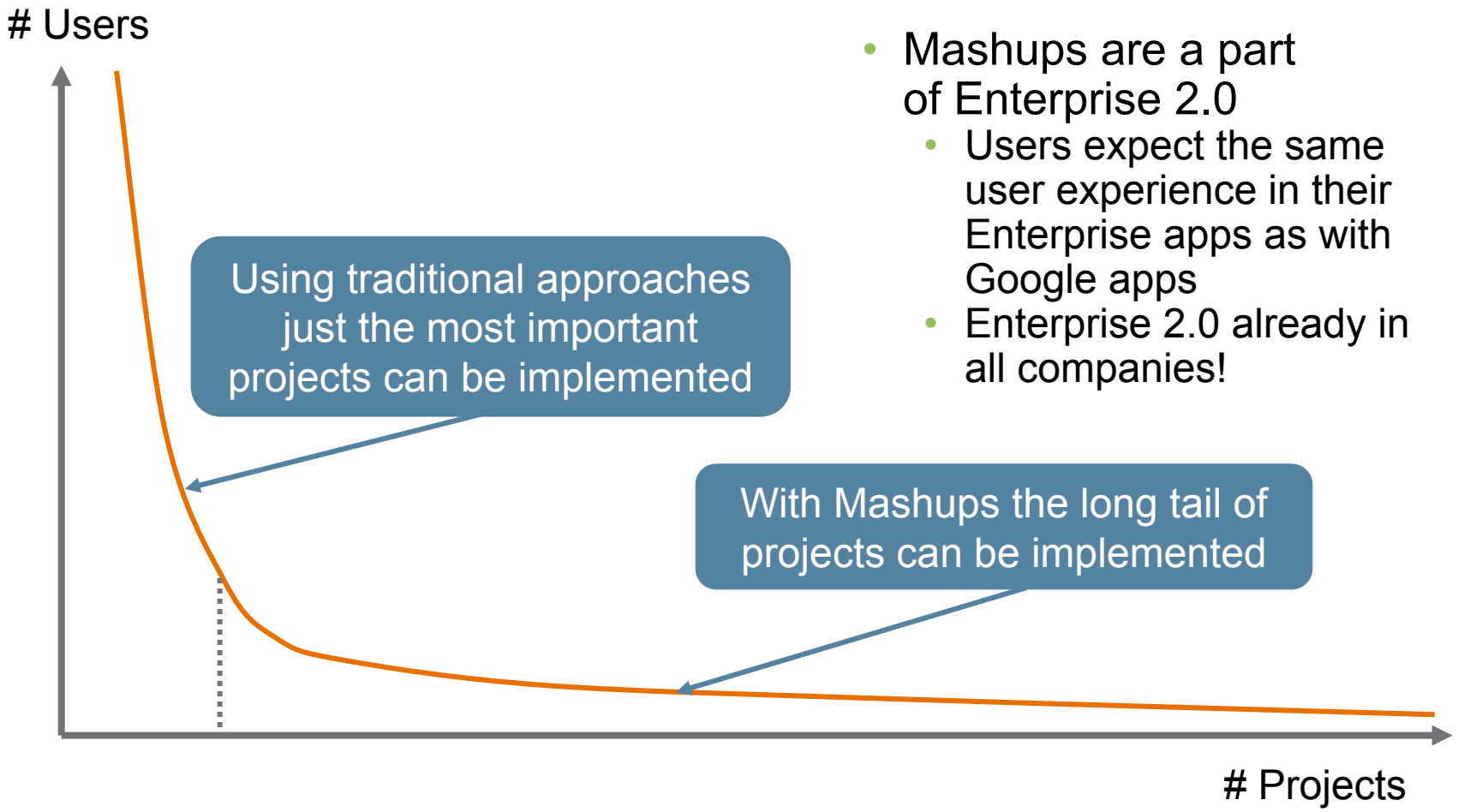
Demo

Definition

- “A mashup is a website or application that combines content from more than one source into an integrated experience”
- The Web as a Platform
  - Services, not packaged software
  - Remixable data sources
- Examples
  - A website + Google Maps = most mashups today
  - Intranet apps + intranet sites + …
    = Enterprise mashups tomorrow

Sources: Wikipedia.org, “What Is Web 2.0” by Tim O’Reilly
(http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html?page=1)

The Longtail of Projects



- Mashups are a part of Enterprise 2.0
  - Users expect the same user experience in their Enterprise apps as with Google apps
  - Enterprise 2.0 already in all companies!

Agenda

Mashup: The New Kid on the Block

Enabling Technologies

The Components of a Mashup

The Mashup Programming Model

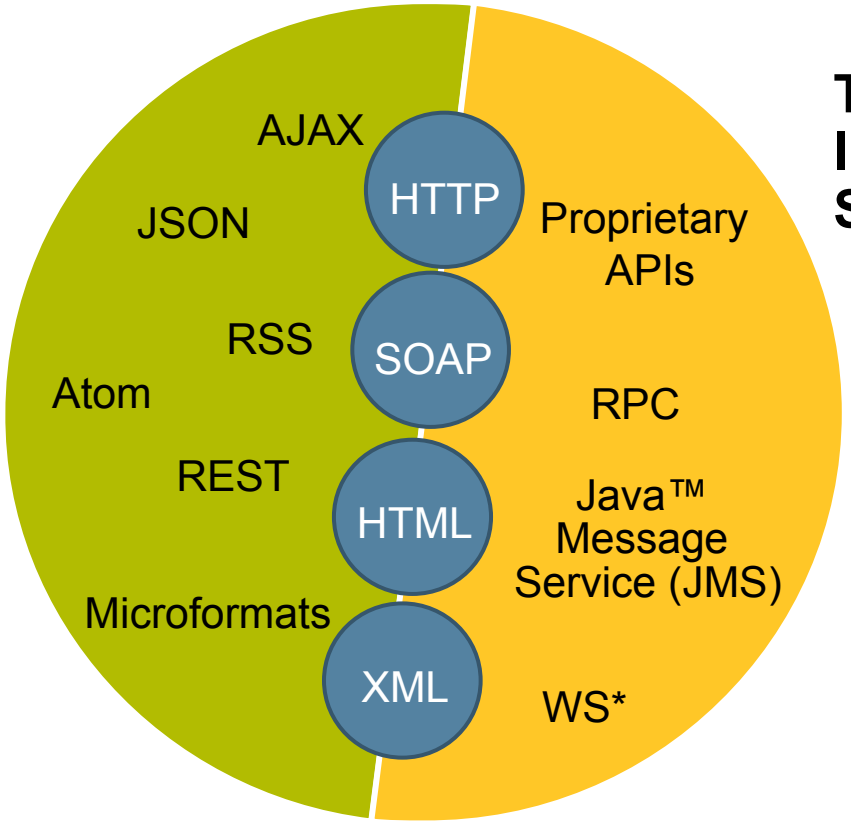
Life of a Developer in a Mashup World

Demo

Mashups and Traditional Integration

Mashups Web 2.0

- Lightweight
- De-facto standards
- Versatile and multipurpose
- Uses the Web as a platform



Traditional Integration SOA Web 1.0

- Tight integration
  - Risky
  - Code changes
  - Brittle
- Cumbersome
- Integration is a big project

Syndication Feeds

- RSS
  - One-way
  - XML format
  - Many flavours
- Atom Syndication Protocol
  - One-way
  - XML format
  - IETF standard
  - Can carry anything (binaries, text, html, etc.)
- ROME (RSS and Atom Utilities)
  - Generate RSS/Atom feeds
  - Parse RSS/Atom feeds

Atom Publishing Protocol (APP)

- Protocol to publish and edit Web resources over REST
  - Everything is a resource and can be represented in APP
- User authentication
- Full CRUD support
  - POST = Create entries
  - GET = Retrieve information (introspection/collections/entries)
  - PUT = Updated entries
  - DELETE = Delete entries
- Examples
  - Apache Abdera
  - Google data API

REST Web Services

- Representational State Transfer
- Resources are available on the web over HTTP
- Resources are addressable (via URLs)
- Powerful combination with:
  - JavaScript™ programming language Callback functions and JSON
  - Atom publishing protocol
- Examples
  - Java™ technology REST Annotation (JRA)
  - Restlet
  - Java API for XML Web Services (JAX-WS) can consume and produces REST services
  - Java Specification Request (JSR) 311—Java API for RESTful Web Services

REST Web Services—Example

- URL:
  - http://service.openkapow.com/andreas/javaonepresentation
- Parameters:
  - searchText=mobile
  - resultformat=json
  - callbackFunction=myCallback
- REST call:
  - http://service.openkapow.com/andreas/javaonepresentation?searchText=mobile&resultformat=json&callbackFunction=myCallback
- Result:

```
[{ "presentation":
   {"title":"Introduction to Mobile Java Technology …",
    "url":"http://developers.sun.com/learning/javaoneonline/2006/mobility/...",
    "year":"2006"}
}]
```

JavaScript Programming Language: Back With a Vengeance

- AJAX = Asynchronous JavaScript technology and XML
  - In the core of all Web 2.0 applications
  - Richer clients by building parts of the application on the client
  - Based on XMLHTTPRequests over HTTP
  - AJAX modifies the DOM (no need for page refresh)
  - JSON = JavaScript Object Notation
- AJAX libraries
  - Don't reinvent the wheel, use already existing AJAX libraries
  - POJO
  - DWR
- Java technology and AJAX
  - JavaServer™ Faces components to wrap AJAX
  - Google Web Toolkit to generate JavaScript programming language from Java code
  - Java Pet Store 2.0 Reference Application

HTML: The World's Most Common API

- Almost all apps today have an HTML front end
- Data in HTML is in application-specific format
- Web scraping: not a bad word
  - Repackage functionality from HTML
  - Access data from HTML
  - Publish this data/functionality as REST, RSS, ATOM…
- Examples
  - Java Mozilla HTML parser
  - Openkapow

Agenda

Mashup: The New Kid on the Block

Enabling Technologies

The Components of a Mashup

The Mashup Programming Model

Life of a Developer in a Mashup World

Demo

Mashup Builders

- Produce the user interface of a Mashup
- Today this is done by programming
  - JavaServer Pages™ (JSP™) technology
  - JavaServer Faces technology
  - etc.
- Tomorrow this is done by assembling
  - Mashups by non-developers
  - Connecting widgets to create composite applications
- Examples
  - Yahoo Pipes
  - IBM QEDWiki
  - Google Gadgets (consumed in IBM and BEA Portals)
  - Project jMaki
  - Teqlo

Widgets

- a.k.a. gadgets, flakes, teqlets, modules, etc.
- The building blocks of Mashups
  - Encapsulates logic and presentation
  - Solves one specific problem
- GUI usually in HTML/CSS/JavaScript programming language
- Logic from underlying services
  - REST
  - Atom
  - APIs

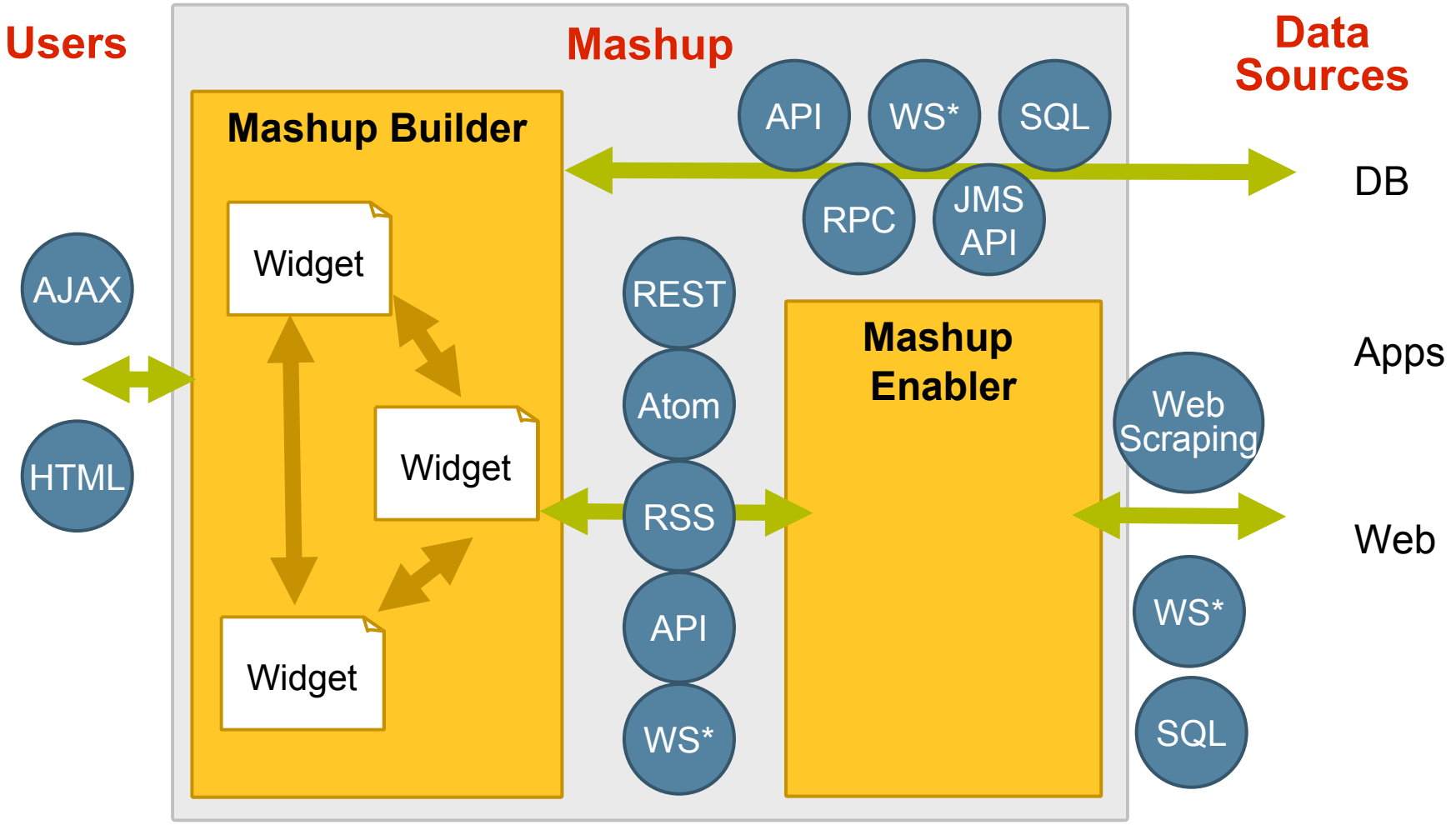
What if There Are No Available Services?

- Data not available in public APIs/Services
  - Hidden in corporate databases
  - Legacy systems
  - Behind firewalls
- This is a key question in all Mashups that are more than toys!

The Solution: Mashup Enablers

- Access the functionality without an API
  - Needs to handle multi-page navigation
  - Sessions
  - Stability and robustness
- Serve functionality to Mashup Builders
    - REST
    - RSS
    - Atom
- Makes all internal and external resources available
- Examples
  - Openkapow
  - Feed43

Inside a Mashup



Agenda

Mashup: The New Kid on the Block

Enabling Technologies

The Components of a Mashup

The Mashup Programming Model

Life of a Developer in a Mashup World

Demo

The Mashup Programming Model

- Lightweight programming model
  - Support lightweight programming models that allow for loosely coupled systems
  - Think syndication, not coordination
  - Design for “hackability” and remixability
- What does this mean?
  - Uncontrolled reuse
  - Collaborative composition
  - Internet-wide interoperability
  - User-centric approach

Source: “What Is Web 2.0” by Tim O’Reilly (http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html?page=1), SOA 2.0: Why a Revision is Really Necessary (http://www.managebility.org/blog/stuff/why-soa-2-dot-0)

Coding vs. Assembling

- Focus on functionality, not technology
  - Assembling is data-centric
  - Coding is logic-centric
- Ad hoc application assembly of situational apps
  - Quicker and easier way to get to the data
  - Enables access to the longtail of problems
- Less code to write
  - Writing widgets, mashup builders, and enablers

Mashups Compared to SOA

- SOA
  - Server-centric
  - Well defined
  - Connects systems
- Mashups
  - Client centric
  - Web-based
  - Ad hoc
  - Connects users
- Mashups extend SOA

Agenda

Mashup: The New Kid on the Block

Enabling Technologies

The Components of a Mashup

The Mashup Programming Model

Life of a Developer in a Mashup World

Demo

What Does This Mean for a Developer?

- Be aware that there is more to mashups than Google maps
- You are not in total control
  - Data will be remixed
  - Mashup Enablers are your friends not your enemies
- Provide:
  - APIs
  - REST Services
  - RSS/Atom feeds

Mashups: A Pragmatic View

- New way to think about what is possible
- Opens new doors
  - More data than ever available
  - More functionality than ever available
- No Silver Bullet
  - Ignore the hype, use what works for you
- Less work, more results

Agenda

Mashup: The New Kid on the Block

Enabling Technologies

The Components of a Mashup

The Mashup Programming Model

Life of a Developer in a Mashup World

Demo

DEMO

Building and deploying a REST service

Summary

- Mashups are all about:
  - Building situational applications ad hoc
  - Reuse and remix both data and functionality
- Using Mashups means that:
  - You can do more in less time
  - You don’t have to reinvent the wheel
  - You are not in total control of how your applications and data will be used

For More Information

- Kapow Technologies (http://kapowtech.com)
- Openkapow (http://openkapow.com)
- Sun Developer Network: Next Generation Web Technologies and Projects (http://developers.sun.com/web/glossary/index.jsp)
- ProgrammableWeb (http://programmableweb.com)
- Enterprise Web 2.0 by Dion Hinchcliffe (http://blogs.zdnet.com/Hinchcliffe)

Integration Gets All Mashed Up: Bridging Web 1.0 and Web 2.0 Applications

Andreas Krohn

Product Manager
Kapow Technologies
www.kapowtech.com

Session TS-8544