





Talking The Language of Enterprise Mashups

Deepak Alur
deepak@jackbe.com
VP Engineering
&
Raj Krishnamurthy
raj@jackbe.com
Chief Architect

About JackBe

-  Founded 2002
-  Chevy Chase, MD
-  Empowering Business Users through Enterprise Mashups
-  Applications deployed to over 4 million end-users across 50+ customers



About the Speakers

Deepak Alur
VP Engineering

Co-author of *Core J2EE
Patterns*

Previously Principal
Engineer @ Sun
Microsystems

Raj Krishnamurthy
Chief Architect

Previously Principal
Architect @ eBay.com &
Sun Microsystems

Mashup

‘A Mashup is a website or application that combines content from more than one source into an integrated experience.’

Source: Wikipedia

But, this can apply to anything!!!

Enterprise Mashups

- 🏊 User Driven and User Focused
- 🏊 Visual and Non-Visual
- 🏊 Client-side and Server-side (mostly)
- 🏊 Enterprise Services plug and play
- 🏊 Dynamic, ad-hoc, situational
- 🏊 Secure and Governed
- 🏊 Sharable, customizable
- 🏊 Near zero cost create and consume

The Need for a Mashup Server

Services are maturing

- Services are Immutable
- Services are 'Instant Legacy'
- Services are *not* user friendly

We want to use Services

- User-oriented mashup activities (or actions)
- User-generated mini-apps
- User-driven Integration / Tooling

The New 'Front-tier'

Introducing the Mashup Layer

Web 1.0

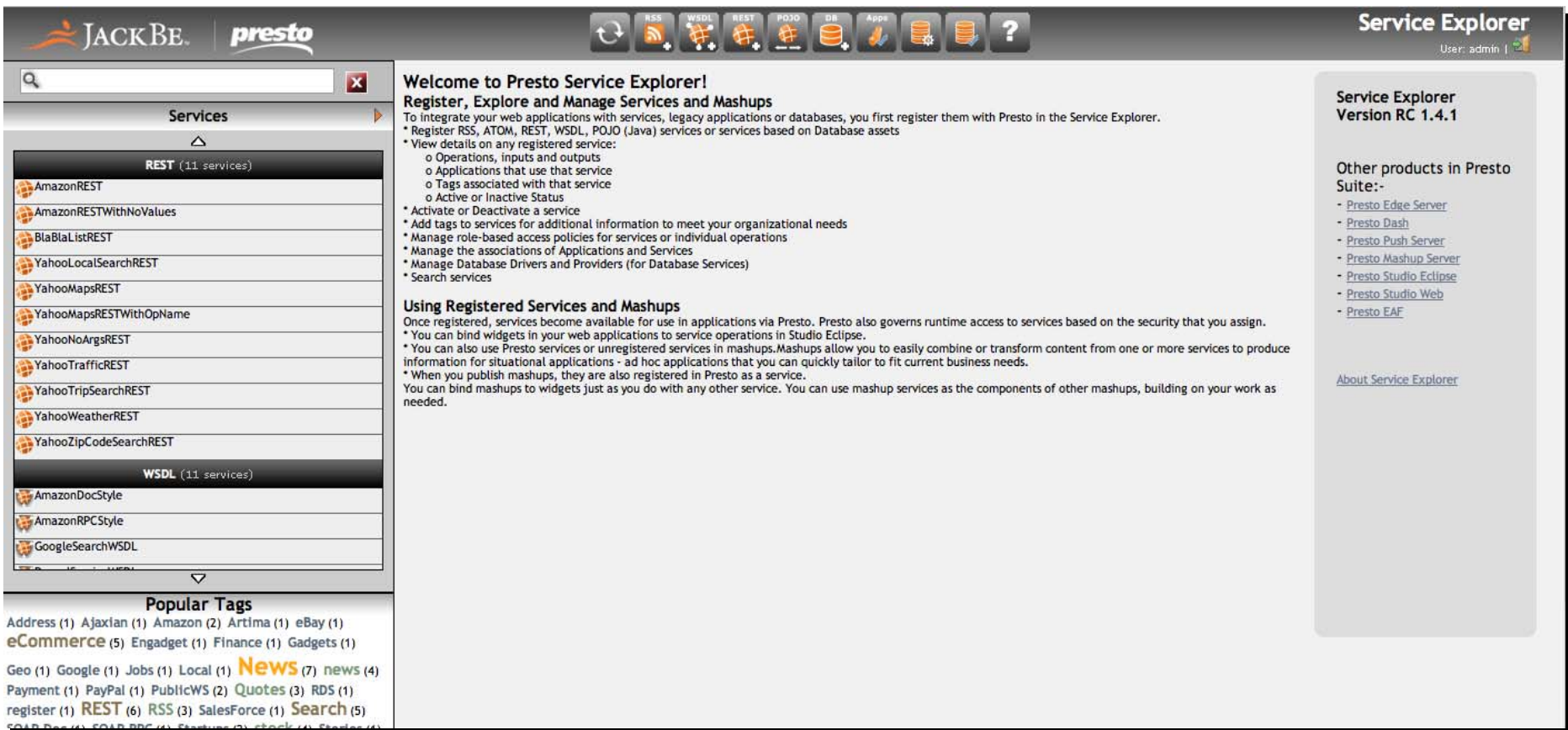


Web 2.0



Demo Time

Presto Service Explorer



The screenshot shows the Presto Service Explorer web application. The interface includes a search bar, a navigation menu, and a main content area with a welcome message and service details.

Services

- REST (11 services)
 - AmazonREST
 - AmazonRESTWithNoValues
 - BlaBlaListREST
 - YahooLocalSearchREST
 - YahooMapsREST
 - YahooMapsRESTWithOpName
 - YahooNoArgsREST
 - YahooTrafficREST
 - YahooTripSearchREST
 - YahooWeatherREST
 - YahooZipCodeSearchREST
- WSDL (11 services)
 - AmazonDocStyle
 - AmazonRPCStyle
 - GoogleSearchWSDL

Popular Tags

Address (1) Ajaxian (1) Amazon (2) Artima (1) eBay (1) eCommerce (5) Engadget (1) Finance (1) Gadgets (1) Geo (1) Google (1) Jobs (1) Local (1) News (7) news (4) Payment (1) PayPal (1) PublicWS (2) Quotes (3) RDS (1) register (1) REST (6) RSS (3) Salesforce (1) Search (5) SOAP (1) SOAP RPC (1) Status (1) stocks (1) stories (1)

Welcome to Presto Service Explorer!
Register, Explore and Manage Services and Mashups

To integrate your web applications with services, legacy applications or databases, you first register them with Presto in the Service Explorer.


- Register RSS, ATOM, REST, WSDL, POJO (Java) services or services based on Database assets
- View details on any registered service:
 - Operations, inputs and outputs
 - Applications that use that service
 - Tags associated with that service
 - Active or Inactive Status
- Activate or Deactivate a service
- Add tags to services for additional information to meet your organizational needs
- Manage role-based access policies for services or individual operations
- Manage the associations of Applications and Services
- Manage Database Drivers and Providers (for Database Services)
- Search services

Using Registered Services and Mashups

Once registered, services become available for use in applications via Presto. Presto also governs runtime access to services based on the security that you assign.

- You can bind widgets in your web applications to service operations in Studio Eclipse.
- You can also use Presto services or unregistered services in mashups. Mashups allow you to easily combine or transform content from one or more services to produce information for situational applications - ad hoc applications that you can quickly tailor to fit current business needs.
- When you publish mashups, they are also registered in Presto as a service.

You can bind mashups to widgets just as you do with any other service. You can use mashup services as the components of other mashups, building on your work as needed.

Service Explorer
User: admin | 

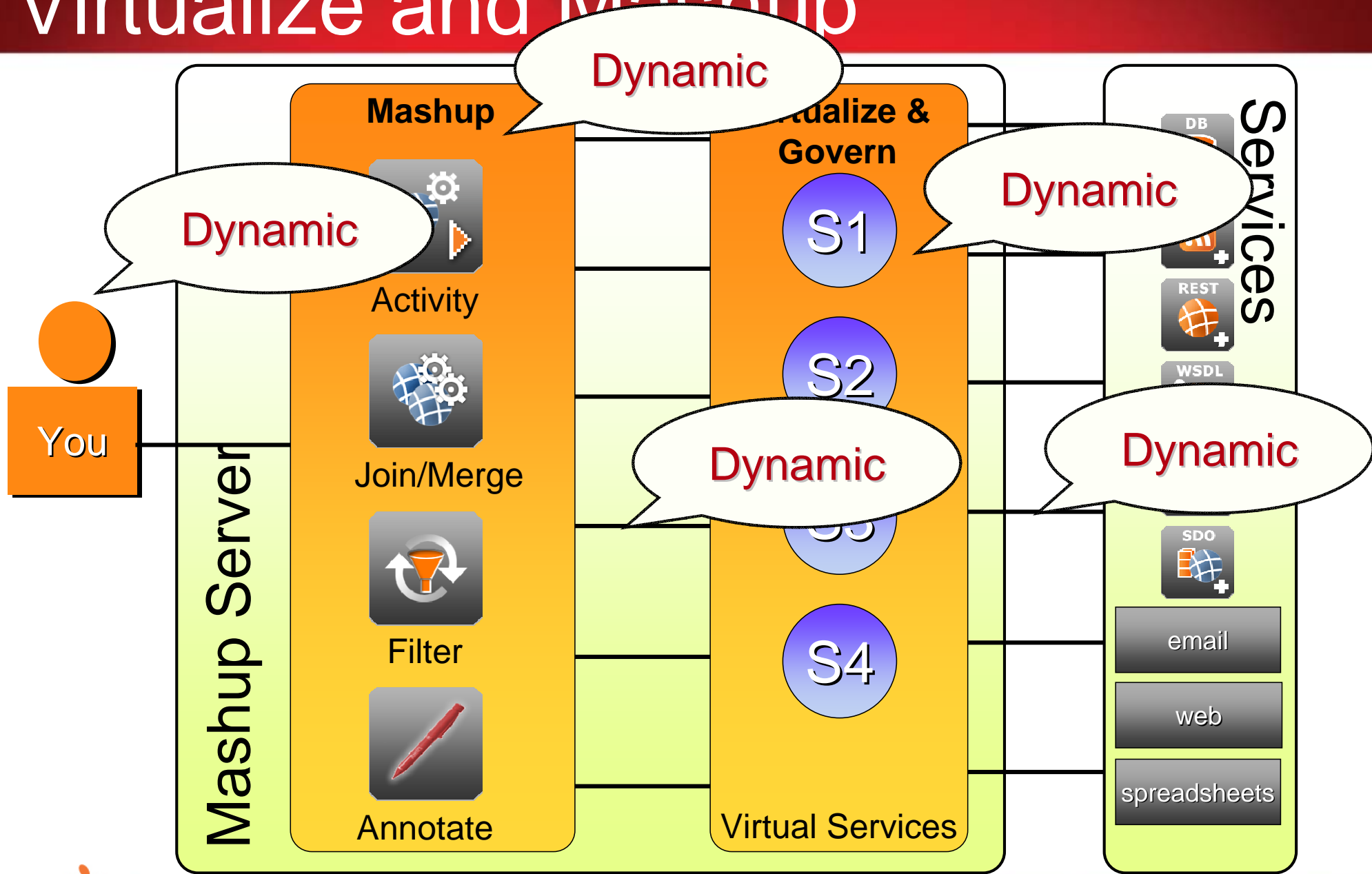
Service Explorer
Version RC 1.4.1

Other products in Presto Suite:-

- [Presto Edge Server](#)
- [Presto Dash](#)
- [Presto Push Server](#)
- [Presto Mashup Server](#)
- [Presto Studio Eclipse](#)
- [Presto Studio Web](#)
- [Presto EAF](#)

[About Service Explorer](#)

Virtualize and Mashup



Mashup Patterns

- Invoke Services
- Filter
- Sort
- Join
- Merge data
- Data Construction & Transformation
- Annotation
- For-Each
- Web Clipping
- Scripting

- Mashup Server
- Mashup Language
- Mashup Composer
- Mashlets

Service Invocation

output = **Invoke** *service* , *operation* , *inputs*

xml

wsdl
rest
dao [governed services]
rss/atom
pojo

Simple Service Invocation

```
< mashup name="YahooHotJobsFilter">
  < invoke service="YahooHotJobsRSS" operation="getFeed"
    outputvariable="result"/>
</ mashup >
```

secured, monitored, cached

normalized output/feed

mashup
service

Service Invocation Complex type

```
<constructor outputvariable="lookup">
  <ns:ItemLookup>
    <ns:SubscriptionId>0525E2PQ81DD7ZTWTK82</ns:SubscriptionId>
    <ns:Validate>>false</ns:Validate>
    <ns:Request>
      <ns:ItemId>{$itemId}</ns:ItemId>
      <ns:IdType>ASIN</ns:IdType>
      <ns:ResponseGroup>Large</ns:ResponseGroup>
    </ns:Request>
  </ns:ItemLookup>
</constructor>
```

Literal

Dynamic value

```
<invoke service="AmazonDocStyle" operation="ItemLookup"
  inputvariables="lookup" outputvariable="reviews" />
```

Filter

Result = Invoke 'YahooHotJobsRSS.getFeed'



Result = Filter Result /rss/channel/item[matches(description, 'ajax')]

Service Filter

```
< mashup name="YahooHotJobsFilter">
```

```
< input name="param" type="string" default="Ajax"/>
```

```
< output name="filtered_result" type="document"/>
```

contract

```
< invoke service="YahooHotJobsRSS" operation="getFeed"  
  outputvariable="result"/>
```

governed service

```
< filter input="result" output="filtered_result"
```

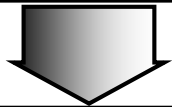
```
  filterexpr = "/rss/channel/item[matches(description, $param)]"/>
```

xpath filter

```
</ mashup >
```

Sort

Result = **Invoke** 'YahooHotJobsRSS.getFeed'



Result = **Filter** Result /rss/channel/item[matches(description, 'Ajax')]



Result = **Sort** Result /rss/channel/item/title **descending**

Sort

```
< mashup name="Jobs">  
  < input name="param" type="string" default="Ajax"/>  
  < output name="sorted_result" type="document"/>  
  
  < invoke service="YahooHotJobsRSS" operation="getFeed"  
    outputvariable="result"/>  
    ↓  
  < filter inputvariable="result"  
    filterexpr = "/rss/channel/item[matches(description, $param)]"  
    outputvariable="result"/>  
    ↓  
  < sort inputvariable="result"  
    sortexpr="/rss/channel/item"  
    sortkeys="title"  
    sortdir="descending"  
    outputvariable="sorted_result"/>  
  </ mashup >
```

xml, json, pojo, rss/atom

Union

Feed1 = Invoke ['http://planetjdk.org/feedfeed.atom-1.0'](http://planetjdk.org/feedfeed.atom-1.0)

Feed2 = Invoke ['http://ajaxian.com/index.xml'](http://ajaxian.com/index.xml)

Feed3 = Invoke ['http://feeds.feedburner.com/ProgrammableWeb'](http://feeds.feedburner.com/ProgrammableWeb)

Union Feed1, Feed2, Feed3

homogeneous structures

Union

```
<externalinvoke endpoint="http://planetjdk.org/feed.atom-1.0"  
    feedtype="atom" outputvariable="feed1"/>
```

```
<externalinvoke endpoint="http://ajaxian.com/index.xml"  
    feedtype="atom" outputvariable="feed2"/>
```

```
<externalinvoke endpoint="http://feeds.feedburner.com/ProgrammableWeb"  
    feedtype="atom" outputvariable="feed3"/>
```

```
<union inputvariables="feed1, feed2, feed3"  
    outputvariable="resultFeed"/>
```

Join

```
<join outputvariable="result"
```

```
  joincondition="$portfolioA/Performance/Month =  
    $portfolioB/Performance/Month" />
```

xml
 json
 pojo
 rss/atom



Join

```
<join outputvariable="consolidatedData">
```

```
  joincondition="$portfolioA/Performance/Month =  
                $portfolioB/Performance/Month">
```

```
    <select name="Performances">
```

```
      <Performance><Month>{$portfolioA/Month}</Month>
```

```
        <Rate>{$portfolioA/Rate}</Rate>
```

```
        <Rate>{$portfolioB/Rate}</Rate>
```

```
      </Performance>
```

```
    </select>
```

custom select



```
</join/>
```

Data Annotation

```
<SalesLead>  
<Title>Director of Vendor Relations</Title>  
<Rating>Hot</Rating>  
<Street>321 Westcott Building</Street>  
<City>Tallahassee</City>  
....  
</SalesLead>
```

service response



```
<SalesLead class="vcard">  
<Title>Director of Vendor Relations</Title>  
<Rating>Hot</Rating>  
<Street>321 Westcott Building</Street>  
<City>Tallahassee</City>  
<geo:lat>30.4424</geo:lat>  
<geo:long>84.303</geo:long>  
....  
</SalesLead>
```

microformats

annotated
service response

geo-coordinates



Mashlets,
maps, timeline,..

Annotate

.....

sales data (\$locationstr)

<externalinvoke

endpoint="http://local.yahooapis.com/MapsService/V1/geocode"
appid="YahooDemo"
output="xml"
location="\$locationstr"
outputvariable="georesult"/>

yahoo geo-coding
service

<annotate variable="saleLead" **expr**="." >

element geo:lat { \$georesult//y:Latitude/string() },
element geo:long { \$georesult//y:Longitude/string() },
attribute "class" { "vcard" }

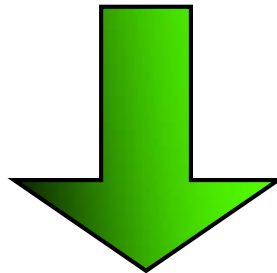
annotate
salesforce data

</annotate>

.....

Select

```
<item>
  <title>Star City</title>
  <link>http://liftoff.msfc.nasa.gov/news/</link>
  <description>How do Americans get ready to work with Russians
aboard the International Space Station? They take a crash course in
culture, language and protocol at Russia's Star City.</description>
  <pubDate>Tue, 03 Jun 2003 09:39:21 GMT</pubDate>
  <guid>http://liftoff.msfc.nasa.gov/2003/06/03.html#item573</guid>
</item>
```

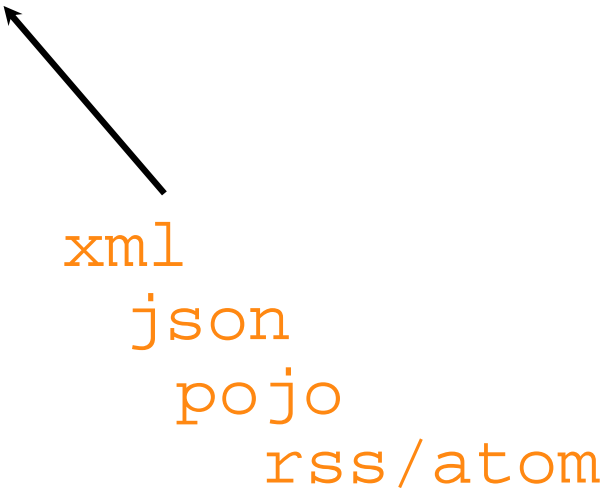


```
<item>
  <title>Star City</title>
  <link>http://liftoff.msfc.nasa.gov/news/</link>
</item>
```

Select

```
<select inputvariable="Data" outputvariable="selectedData"  
        selectexpr="/rss/channel/item">  
  <columns>  
    <column>title</column>  
    <column>description</column>  
  </columns>  
</select>
```

xml
json
pojo
rss/atom



Javascript

```
<externalinvoke endpoint="https://www.google.com/accounts/ClientLogin"  
  method="post"  
  accountType="GOOGLE"  
  Email="xxxxxxxx@gmail.com"  
  Passwd="yyyyyyyyyy"  
  service="blogger"  
  source="jackbe-jems"  
  header="$reqheaders"  
  outputvariable="result"/>
```

```
<script type="text/javascript">  
  <![CDATA[  
    var r = new String(result)  
    var ar = r.split("=");  
    auth = ar[ar.length-1];  
    auth = auth.slice(0, -1)  
  ]>  
</script>
```

```
SID=xxx.....zzzzzzz  
LSID=yyyy....wwwww  
Auth=<authentication token>
```

E4X - Rhino engine
script java objects
variable propagation

```
<assign variable="auth" ..../>
```

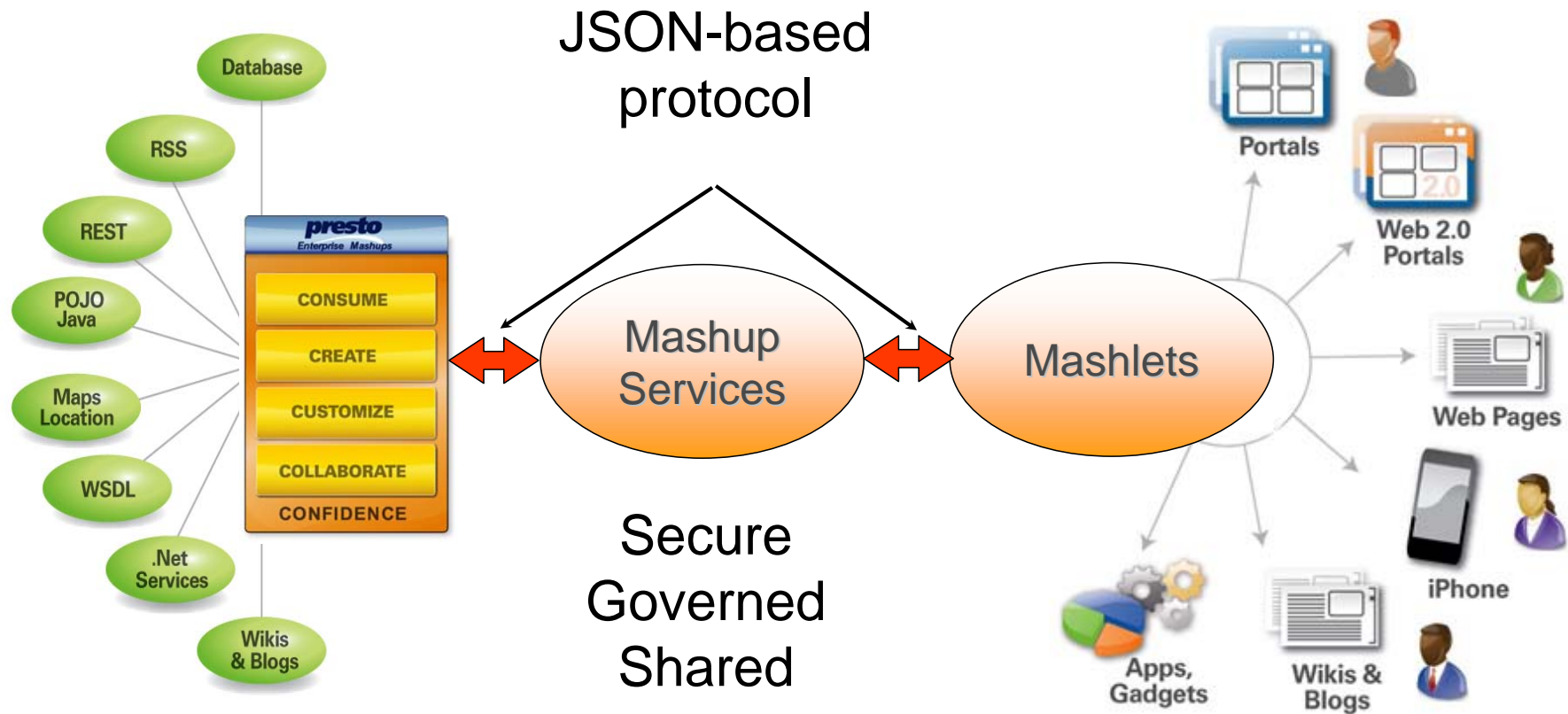
Web clipping

```
<externalinvoke outputvariable = "searchresult"  
    endpoint="http://www.google.com/search?q=ruby"/>  
  
<foreach variable="query" items="$searchresult//xhtml:a[@class='l']">  
    <appendresult outputvariable="result">  
        <itemlink>{$query/@href}</itemlink>  
    </appendresult>  
</foreach>  
  
<queries>  
    <itemlink href="http://www.ruby-lang.org"/>  
    <itemlink href="http://www.ruby-lang.org/en/20020101.html"/>  
    <itemlink href="http://en.wikipedia.org/wiki/Ruby_programming_language"/>  
    <itemlink href="http://en.wikipedia.org/wiki/Ruby"/>  
    <itemlink href="http://www.rubyonrails.org"/>  
    <itemlink href="http://www.rubycentral.org"/>  
    <itemlink href="http://www.w3.org/TR/ruby"/>  
    <itemlink href="http://www.youtube.com/watch?v=JMDcOViViNY"/>  
    <itemlink href="http://www.rubycentral.com/book"/>  
    <itemlink href="http://weblog.infoworld.com/tcdaily/archives/2007/07/tamino_xml_serv.html"/>  
</queries>
```

Other features

- If-Else
- For Each
- Parallel
- OnError/OnTimeOut
- Assign
- Template
- Dynamic Parameterization
- XQuery Scripting

From Services to Mashlets



To learn more...

Events, Videos, Whitepapers, Blogs,
Articles, Contests and Trial Software at
www.jackbe.com

October 24, 2007
3:00 pm - 4:00 pm Eastern DT [Talking the Enterprise Mashup Language](#)

November 2007

November 9, 2007
12:00 pm - 1:00 pm Eastern ST [Enterprise Mashup Bootcamp: What, Why and How?](#)

November 28, 2007
3:00 pm - 4:00 pm Eastern ST [Mashing the Corporate Portal](#)

December 2007

December 12, 2007
12:00 pm - 1:00 pm Eastern ST [Mashups in Financial Services](#)



Take a Survey,
Enter to win
an iPod.



The Enterprise Web 2.0 Blog

JACKBE News

SDFE

Talking The Language of Enterprise Mashups

FRE

Thank you for attending!
For more information,
go to www.jackbe.com.

