Talking The Language of Enterprise Mashups

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

JACKBE™
the user is the killer app

About JackBe

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

JACKBE™

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’

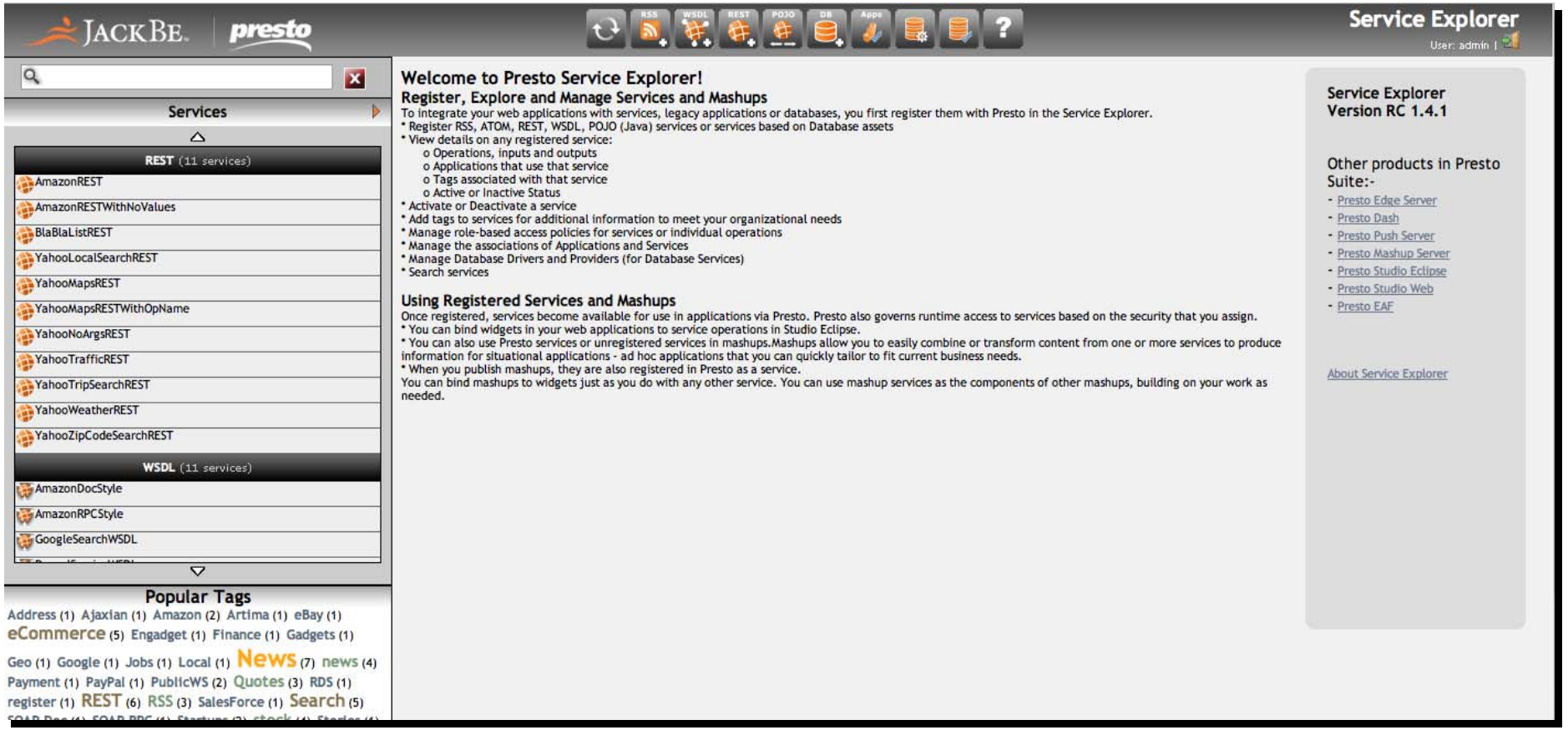
Introducing the Mashup Layer

Web 1.0

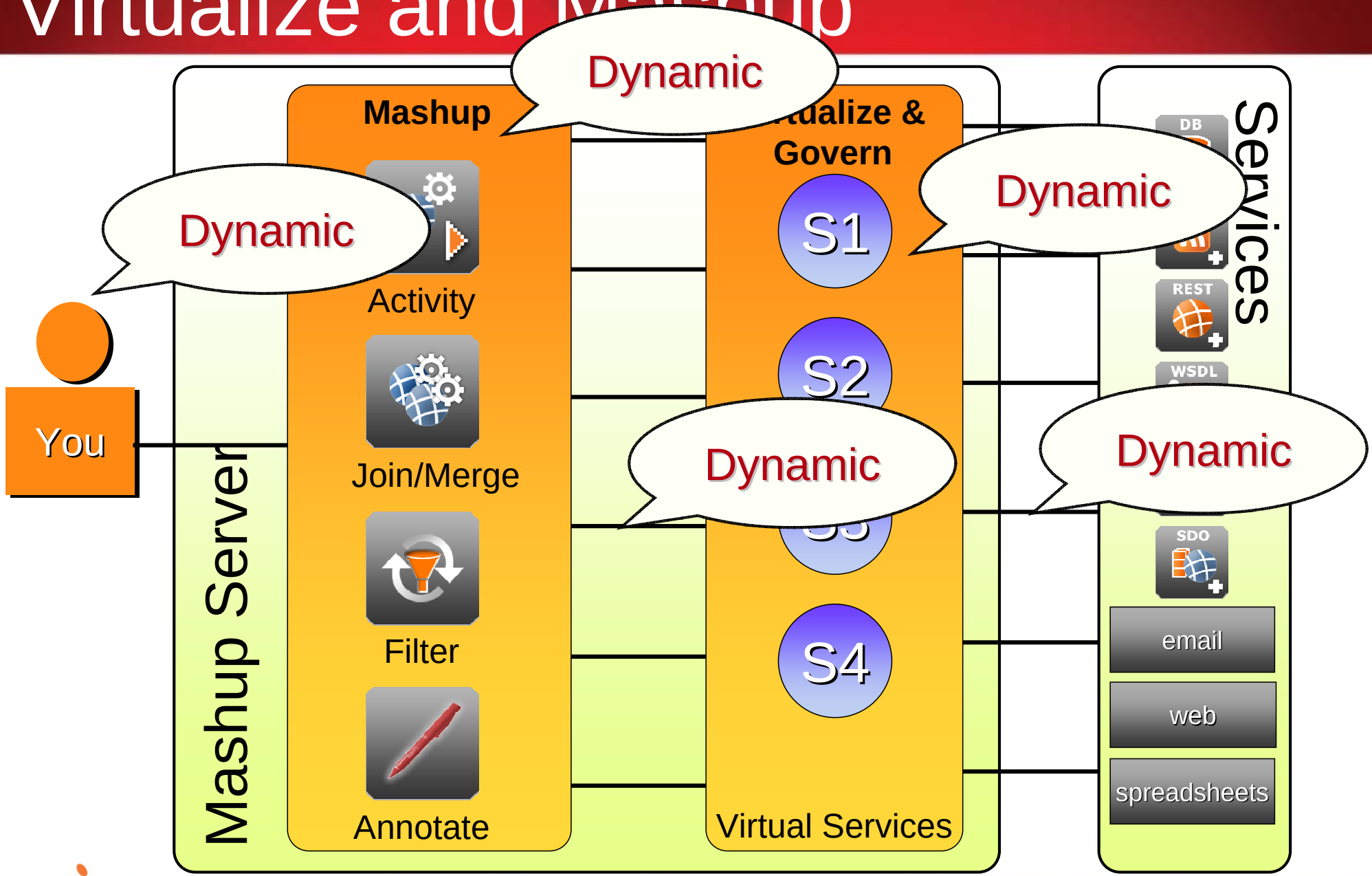
Web 2.0

Demo Time

Presto Service Explorer



Virtualize and Mashup



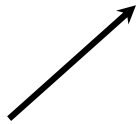
Mashup Patterns

- 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
rss/atom
pojo

[governed services]

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>
```

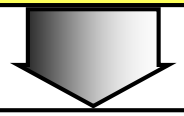
Literal

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"/>

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

  <filter input ="result" output ="filtered_result"
        filterexpr = "/rss/channel/item[matches(description, $param)]"/>

</mashup>
```

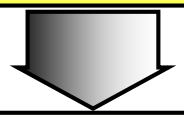
contract

governed service

xpath filter

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’

Feed2 = **Invoke** ‘http://ajaxian.com/index.xml’

Feed3 = **Invoke** ‘http://feeds.feedburner.com/ProgrammableWeb’

Union Feed1, Feed2, Feed3

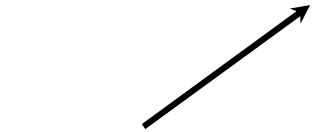
homogeneous structures

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>

<join/>
```

custom select

Data Annotation

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

service response

microformats

```
<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>
```

annotated service response

geo-cordinates

Mashlets, maps, timeline,..

Annotate

.....

sales data ($locationstr)

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

<annotate variable="saleLead" expr="." >
   element geo:lat { $georesult//y:Latitude/string() },
   element geo:long { $georesult//y:Longitude/string() },
   attribute “class” { "vcard" }
</annotate>
```

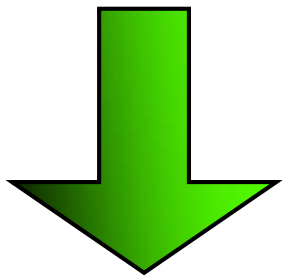
yahoo geo-coding service

annotate salesforce data

.....

Select

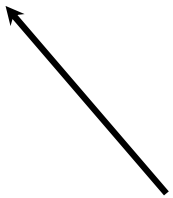
```
<item>
      <title>Star City</title>
      <link>http://liftoff.msfc.nasa.gov/news/</link>
      <description>How do Americans get ready to work with Russians
aboard the International Space Station? They take a crash course in
culture, language and protocol at Russia's Star City.</description>
      <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="xxxxxxxxx@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>

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

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

```
E4X - Rhino engine
script java objects
variable propagation
```

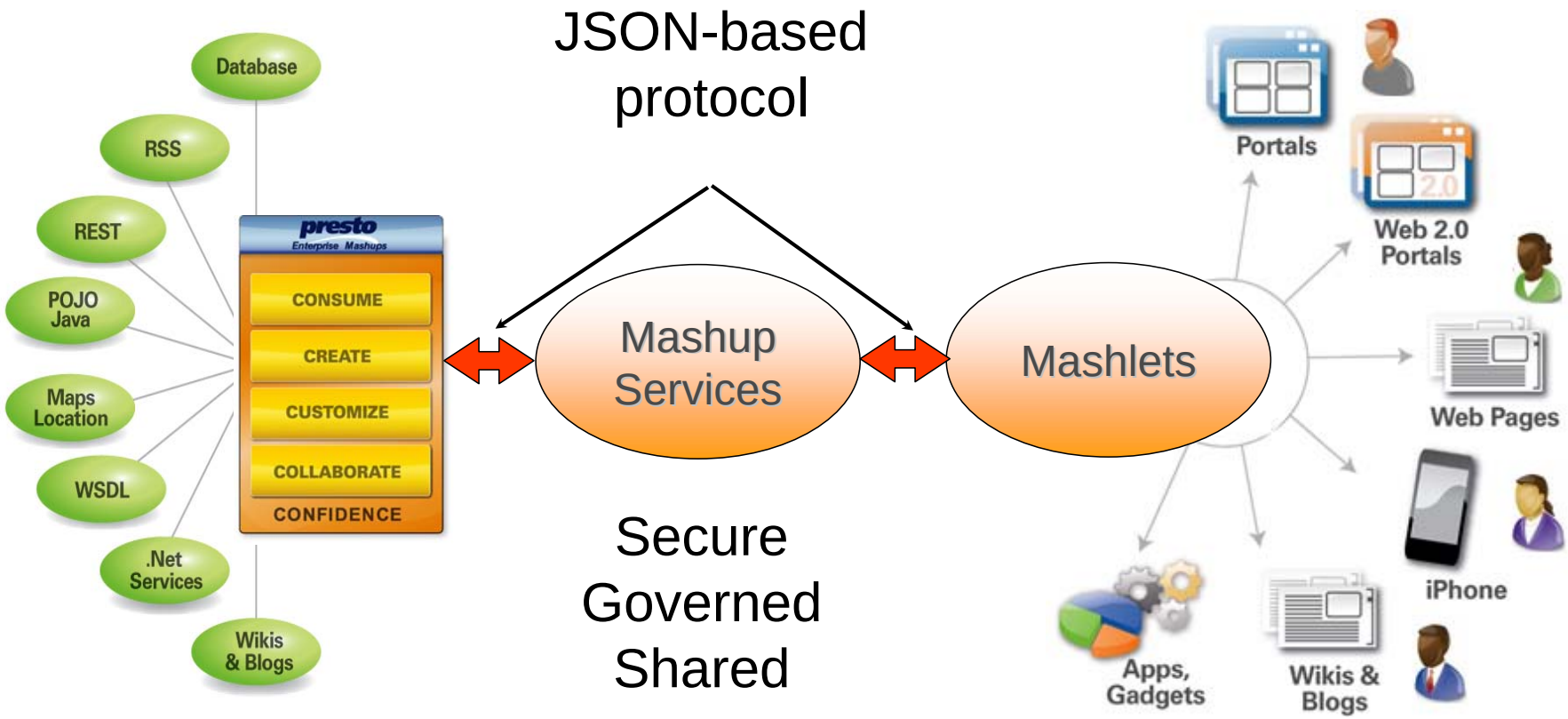
Web clipping

```
<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 Paramaterization
- 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.

JACKBE TV

The Enterprise Web 2.0 Blog

JACKBE News

Talking The Language of Enterprise Mashups

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

JACKBE™
the user is the killer app