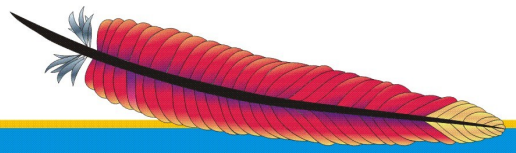
ApacheCon

Apache Roller and blogs as a web development platform

Dave Johnson
Sun Microsystems, Inc.

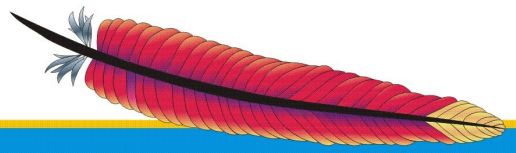


Session Goals: to learn...

How blogs, wikis, feeds and planets enable the self-service web

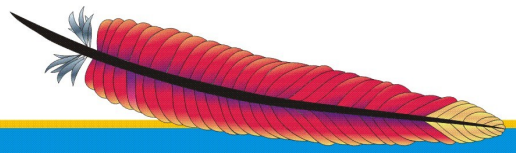
Techniques for automating and integrating blogs, wikis, feeds and planets

How Roller and be extended and customized via templates, plugins and scripting



Agenda

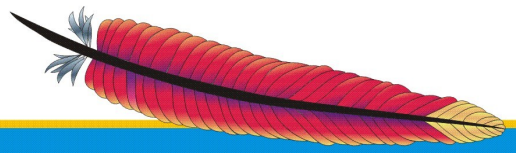
- **Overview**
- Blog, wiki and feed Web services
- Blog customization via templates
- Blog server customization via plugins
- Example: Scripting languages in Roller
- Planet server aggregation
- Use cases: Software business blogs



The self-service web

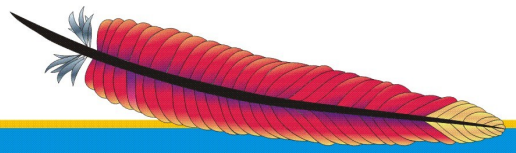
- chmod 777 web
- Let everybody write the web

- Get IT and web-masters out of the way
- Give the tools to the users
  - Blogs, wikis, planets and feeds



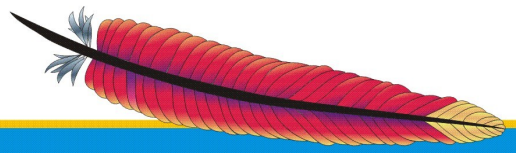
The self-service web

- Blogs, wiki and feed technologies
- Make it easy for your users to:
  - Read, write and monitor the web
  - To build interesting sites & seed communities
  - Create useful information feeds
- With scripts, templates
- Or no code at all



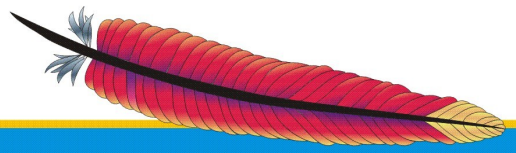
With no code at all?

- Easy web publishing and feed generation
  - Easy interface for writers
  - RSS/Atom feeds for entries and comments
  - Filtered feeds by category and tags
  - Search feeds
  - Aggregated feeds
  - Podcast / media feeds

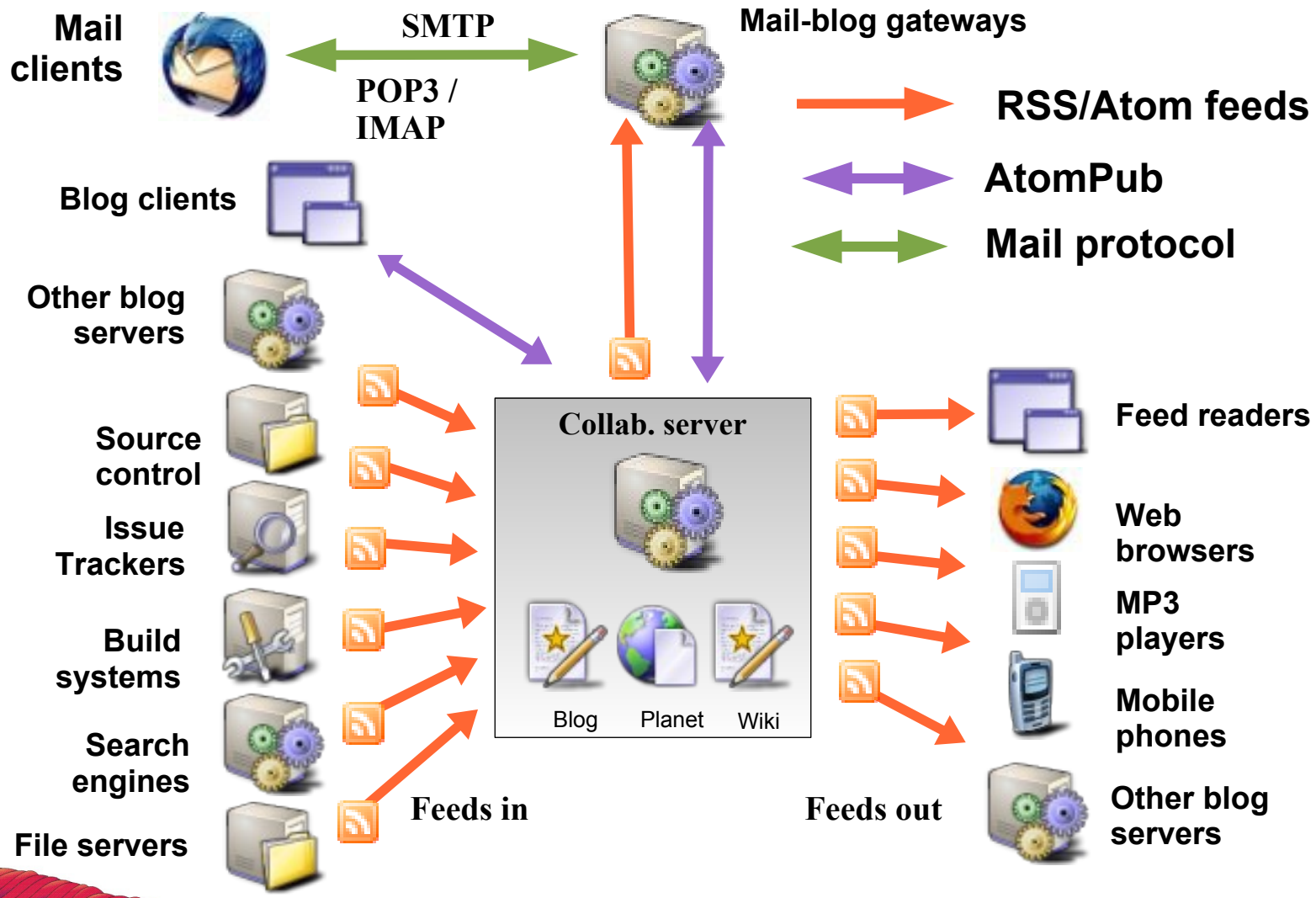


Or with just a little code...

- With a little know-how
  - Custom blog pages and feeds
  - Feed aggregations
- With a little help from the IT department
  - Custom publishing clients
  - Weblog entry plugins
  - Weblog rendering plugins

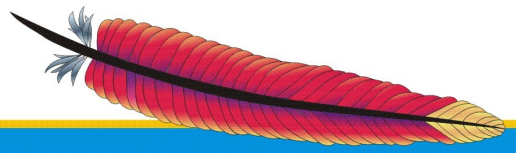


Feed-based integration



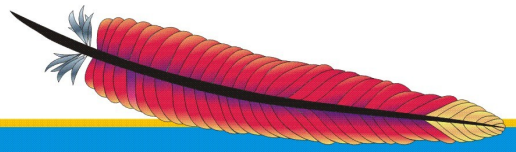
Agenda

- Overview
- **Blog, wiki and feed Web services**
- Blog customization via templates
- Blog server customization via plugins
- Example: Scripting languages in Roller
- Planet server aggregation
- Use case: Software business blogs



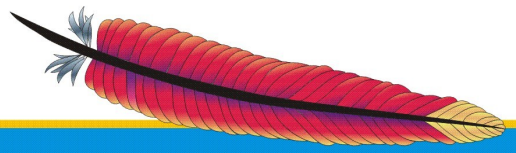
Blog, wiki & feed web services

- RSS/Atom feeds
- MetaWeblog API
- Atom Publishing Protocol
- Roller Admin Protocol
- Wiki web services



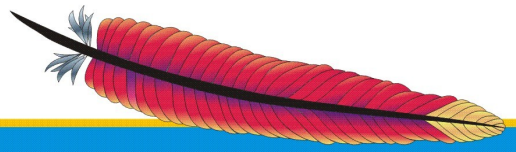
Meta Weblog API

- Based on XML-RPC
- Supports
  - Multiple Blogs per User
  - Posting, editing and deleting Blog Posts
  - Get X recent Blog Posts, but no paging
  - Uploading binary files



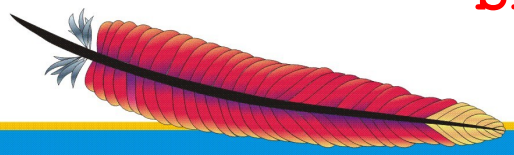
MetaWeblog API: methods

getUserBlogs	Get blogs as array of structures
newPost	Create new blog post by passing in structure
getPost	Get blog post by id
getRecentPosts	Get most recent N blog posts
editPost	Update existing blog post
deletePost	Delete blog post specified by id
newMediaObject	Upload file to blog (e.g. picture of my cat)
getCategories	Get categories allowed in blog



Blog with Apache XML-RPC

```
Hashtable post = new Hashtable();
post.put("title", title);
post.put("description", description);

XmlRpcClientConfigImpl xmlrpcConfig =
    new XmlRpcClientConfigImpl();
xmlrpcConfig.setServerURL(new URL(endpoint));
XmlRpcClient client = new XmlRpcClient();
client.setConfig(xmlrpcConfig);

String result = (String)client.execute(
    "metaWeblog.newPost", new Object[] {
    blogid, username, password, post, true});
```

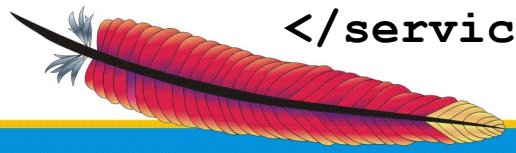


Atom Publishing Protocol

- REST based publishing protocol
- Based on Atom feed format
- Create, edit, update & retrieve collections
  - Blog entries, podcasts, uploads, etc.
  - GData: photos, calendar events, etc.
  - OpenSocial: people, activities, persistence
- Toolkits
  - Apache Abdera (incubating)
  - ROME Propono

AtomPub: Service Doc

```
<service xmlns="http://www.w3.org/2007/app">

   <workspace title="My Weblog" >
      <collection title="Weblog Entries"
         href="http://hostname/app/myweblog/entries">
         <accept>entry</accept>
      </collection>

      <collection title="Uploaded Files"
         href="http://hostname/app/myweblog/resources">
         <accept>image/*</accept>
      </collection>
   </workspace>

</service>
```



AtomPub collection <feed>

```
<feed xmlns="http://www.w3.org/2005/Atom">

  <link rel="next"
    href="http://hostname/app/myweblog/entries/60"/>

  <link rel="previous"
    href="http://hostname/app/myweblog/entries/30"/>
  ...
  <entry> ... </entry>
  <entry> ... </entry>
  <entry> ... </entry>
  <entry> ... </entry>
  ...
</feed>
```

URIs for next and previous portions of collection

<entry> in a collection

```
<entry>
  <title>Unit test results</title>
  <link rel="alternate"
    href="http://localhost/roller/page/bill?entry=post1" />
  <link rel="edit"
    href="http://localhost/roller/app/bill/entry/757" />
  <category term="/UnitTests" />
  <id>http://host/roller/status/entry/report1</id>
  <updated>2005-12-27T22:08:03Z</updated>
  <published>2004-10-13T01:07:59Z</published>
  <content type="html">Content goes here...</content>
  <app:control>
    <app:draft>no</app:draft>
  </app:control>
</entry>
```

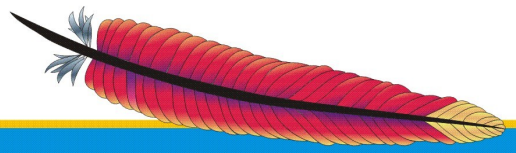
Entry's permalink

Edit URI for entry

Draft status

Posting a blog with Abdera

```
Abdera abdera = new Abdera();
Factory factory = abdera.getFactory();

Client client = new CommonsClient(abdera);
client.addCredentials(endpoint, null, null,
   new UsernamePasswordCredentials(username, password));

Document service_doc = client.get(start).getDocument();
Service service = service_doc.getRoot();
Collection collection = service.getWorkspaces().get(0)
   .getCollections().get(0);
String uri = collection.getHref().toString();

Entry entry = factory.newEntry();
entry.setTitle("Hey people");
entry.setContentAsHtml("I'm <b>blogging</b> this!");
Response response = client.post(uri, entry);
```



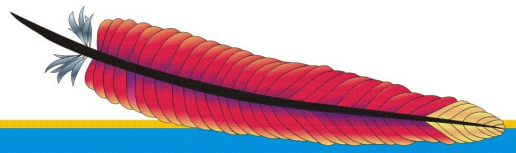
Blog with ROME Propono

```
ClientAtomService service =
   AtomClientFactory.getAtomService(
    endpoint, uname, pword);

ClientWorkspace workspace =
   (ClientWorkspace)service.findWorkspace("My Weblog");

ClientCollection collection =
   (ClientCollection)ws.findCollection(null, "entry");

ClientEntry entry = collection.createEntry();
entry.setTitle("Hey people");
entry.setContent(
    Content.HTML, "I'm <b>blogging</b> this!");
collection.addEntry(entry);
```

Roller Admin Protocol (RAP)

- REST based protocol for Roller
  - Add, edit and delete Users
  - Add, edit and delete Weblogs
  - Add, edit and delete Members

- Some similarities to AtomPub protocol
  - We should make it AtomPub compliant!

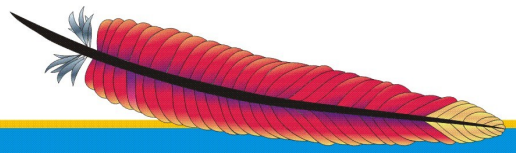


RAP: Service Doc

```
<service xmlns="http://purl.org/roller/rap#">
  <workspace title="Collections for administration">
    <collection title="Weblog administration entries"
      href="http://suntoy:8080/roller/rap/weblogs">
      <member-type>weblog</member-type>
    </collection>
    <collection title="User administration entries"
      href="http://suntoy:8080/roller/rap/users">
      <member-type>user</member-type>
    </collection>
    <collection title="Member administration entries"
      href="http://suntoy:8080/roller/rap/members">
      <member-type>member</member-type>
    </collection>
  </workspace>
</service>
```

RAP: User Set

```
<users xmlns="http://purl.org/roller/rap#"
   href="http://example.org/roller/rap/users">

  <user href="http://example.org/roller/rap/users/mort">
    <name>mort</name>
    <full-name>Mortimer Snerd</full-name>
    <password>edgar</password>
    <locale>en</locale>
    <timezone>US/Pacific</timezone>
    <email-address>mort.snerd@example.org</email-address>
    <date-created>1141863448000</date-created>
  </user>

</users>
```

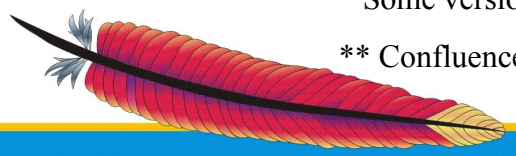


WikiRPCInterface

- Based on XML-RPC
- Supported by:
  - Apache JSPWiki (incubating), MoinMoin, Twiki, PHPWiki
- Limitations:
  - No paging, read-only*
  - Not supported by Confluence**
- Future
  - WikiRPCInterface2 vs. AtomPub for wikis

* Some versions allow write access.

** Confluence has its own XML-RPC and SOAP based APIs.

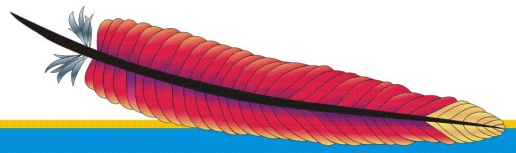


WikiRPCInterface: methods

getRecentChanges	Get recent changes as array
getPage	Get the raw Wiki text of page, latest version
getPageVersion	Get the raw Wiki text of page, by version
getPageHTML	Return page in rendered HTML, latest version
getPageHTMLVersion	Return page in rendered HTML, by version
getPageInfo	Get page info as struct, latest version
getPageInfoVersion	Get page info as struct, by version
getAllPages	Returns a list of all pages

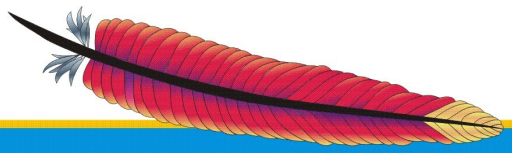
Agenda

- Overview
- Blog, wiki and feed Web services
- **Blog customization via templates**
- Blog server customization via plugins
- Example: Scripting languages in Roller
- Planet server aggregation
- Use case: Software business blogs



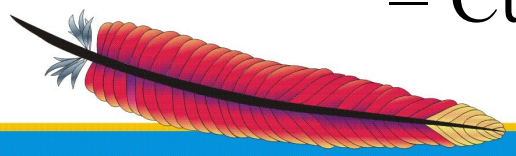
Roller Weblog Templates

- Each weblog's layout and styles defined by a small set of templates.
- Template language: Apache Velocity
  - Other languages possible via plugins
- Data available to templates via *Models*



Why customize templates?

- Change design and layout of your blog
- Add new pages and feeds to your blog
  - HTML, XHTML pages
  - JSON pages for Ajax effects
  - Custom RSS/Atom feeds
- Using data from
  - Roller's data model
  - RSS/Atom feed data from built-in Planet
  - Custom model objects



Roller page model objects

PageModel
locale
tags
weblog
weblogCategory
weblogEntriesPager
weblogEntry
isPermalink
...

UtilitiesModel
decode()
encode()
escapeHTML()
escapeJavaScript()
escapeHTML()
formatDate()
formatISO8601Date()
...

URLModel
home()
feed()
login()
logout()
page()
resource()
search()
...

MenuModel

PreviewModel

MessageModel

SearchResultsFeedModel

CalendarModel

Pager
homeLink
homeName
nextLink
nextName
prevLink
prevName
items

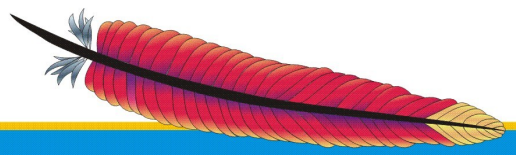
ConfigModel

SearchResultsModel

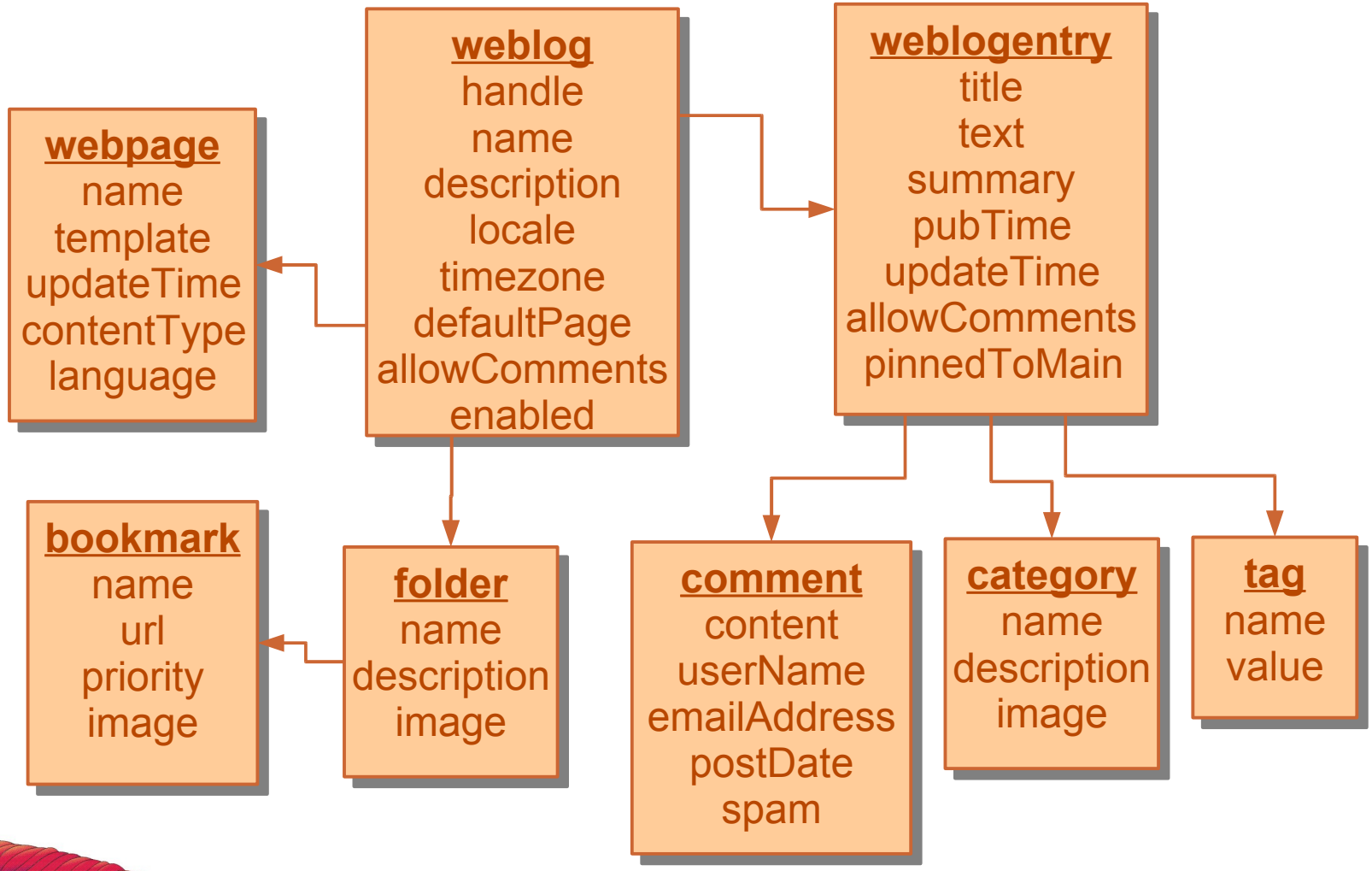
PlanetModel

FeedModel

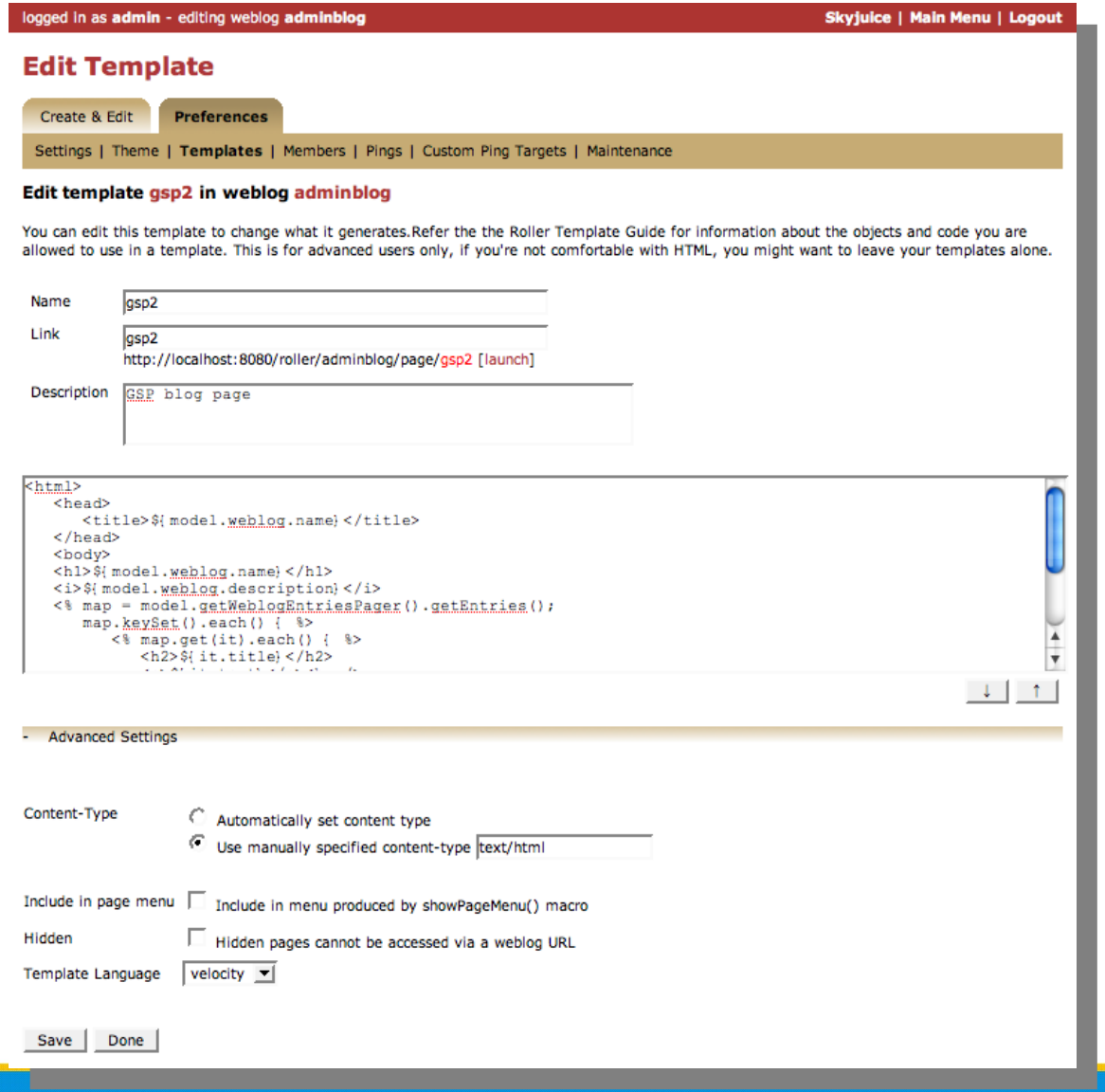
SiteMo



Roller data model

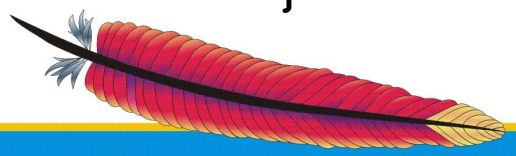


Roller Template Editor



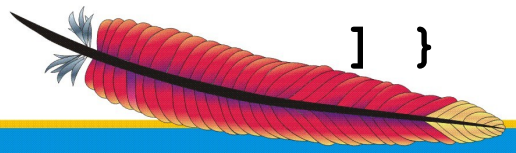
Output blog entries as JSON

```
#set($pager = $model.getWeblogEntriesPager())
{
#if($pager.nextLink) nextLink:"$pager.nextLink", #end
#if($pager.prevLink) prevLink:"$pager.prevLink", #end

data: [
#foreach($day in $pager.getEntries().keySet())
   #set($entries = $pager.getEntries().get($day))
   #foreach($entry in $entries)
      #if($comma),#else#set($comma = true)#end
      {id: "$entry.website.handle:$entry.anchor",
       pubTime: "$entry.pubTime", title:"$entry.title"}
   #end
#end ]
}
```



Blog entries as JSON w/paging

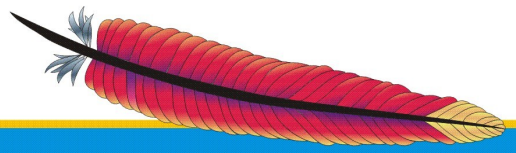
```
{ nextLink:"/roller/page/dojoentries?page=1",
data: [
  { id:      "roller:latest_links14",
    pubTime: "2007-03-21 22:42:13.0",
    title:   "Latest links: March 21, 2007" }
  { id:      "roller:jruby_on_roller",
    pubTime: "2007-03-20 21:46:23.0",
    title:   "JRuby on Roller" }
  { id:      "roller:congrats1",
    pubTime: "2007-03-20 13:02:07.0",
    title:   "Congrats" }
  { id:      "roller:roller_on_sjs_web_server",
    pubTime: "2007-03-20 13:02:02.0",
    title:   "Roller on SJS Web Server 7.0" }
  ...
] }
```



Blog to JSON to DOJO table

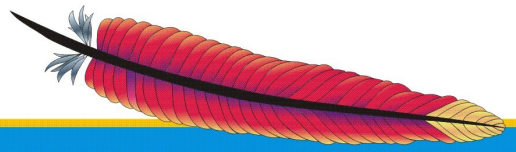
Agenda

- Overview
- Blog, wiki and feed Web services
- Blog customization via templates
- **Blog server customization via plugins**
- Example: Scripting languages in Roller
- Planet server aggregation
- Use case: Software business blogs



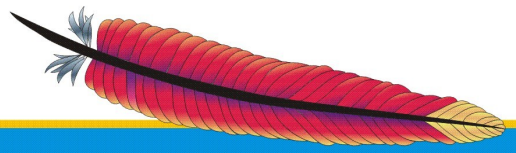
Nine type of Roller plugins

- **Page Model Plugin - add your own models for use in templates**
- Weblog Editor UI Plugin
- **Weblog Entry Plugin - Transform entry content at display time**
- Comment Authentication Plugin
- **Comment Validation Plugin - Add comment validation rules**
- **Renderer Plugin - Plugin your own template language**
- Request Mapper Plugin
- Cache System Plugin
- Repeatable Task



But there is no plugin API?

- Roller is really an app, not an API
- Plugins should use these APIs:
  - Roller Manager Interfaces
  - Roller POJOs
  - Roller Configuration
  - Roller Utilities
  - Roller Request objects
- We need to add a stable/formal Plugin API!



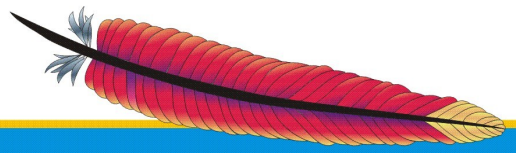
Roller Weblog Entry Plugins

- Transform entries at display time
- Implement this interface

```
public interface WeblogEntryPlugin {
    public String getName();
    public String getDescription();
    public void init(Weblog weblog);
    public String render(WeblogEntry e, String s);
}
```

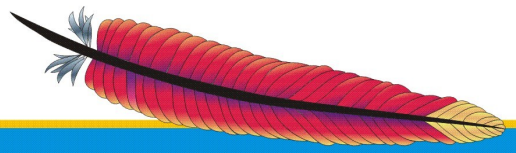
- Add class to config property:

```
plugins.page=
```



Weblog Entry Plugin examples

- JSPWiki plugin
- Emoticon plugin
- Technorati Tag plugin
- Wikipedia plugin
- Google plugin
- Textile plugin



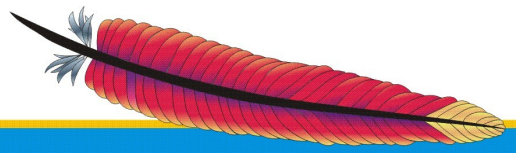
Roller Page Model Plugins

- Plugin new models for use in templates
- Implement model interface

```
public interface Model {
    public String getModelName();
    public void init(Map args); // called each request
}
```

- Add to appropriate configuration properties

```
rendering.pageModels=       // for weblog templates add
rendering.searchModels=     // your plugin to these three
rendering.previewModels=    // lists of model classes
rendering.feedModels=       // for built-in feeds
```



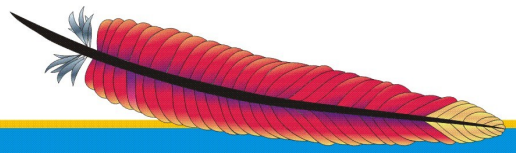
Comment Validation Plugins

- Plugin your own comment validation
- Implement this interface:

```
public interface CommentValidator {
   public String getName();
   public int validate(Comment c, RollerMessages m);
}
```

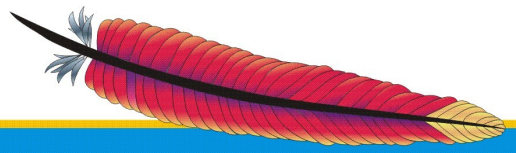
- Return 0 if spam and 100 if not
- Add your class to the list of validators

```
comment.validator.classnames=
```



Existing Comment Validators

- ExcessSizeCommentValidator
- ExcessLinksCommentValidator
- AkismetCommentValidator
- BlacklistCommentValidator



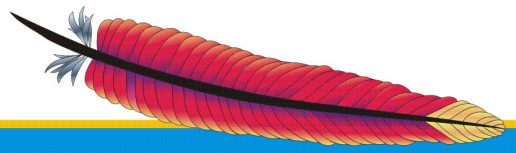
Roller Renderer Plugins

- Allows you to plugin your own view
- Implement this factory interface:

```
public interface RendererFactory {
    public Renderer getRenderer(Template template);
}
```

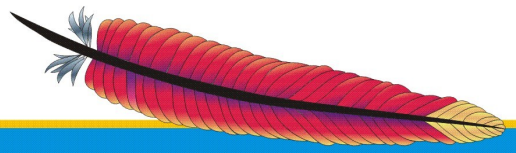
- Return null if you don't want to render
- And this render interface:

```
public interface Renderer {
    public void render(Map model, Writer writer)
        throws RenderingException;
}
```



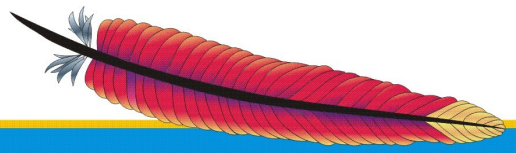
Agenda

- Overview
- Blog, wiki and feed Web services
- Blog customization via templates
- Blog server customization via plugins
- **Example: Scripting languages in Roller**
- Planet server aggregation
- Use case: Software business blogs



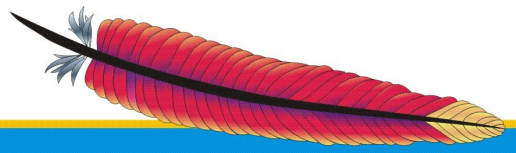
Scripting in languages in Roller

- Author pages using scripting languages
  - Via renderer plugins (available separately)
  - JRuby, Javascript, Jython, etc.
  - Groovy Server Pages (GSP)
- Script Roller via the "Roller API"
  - Import/export blog data
  - Provision blogs and users



Example JRuby page template

```
$out.println "<html><head>"
$out.println "<title>#{$model.weblog.name}</title>"
$out.println "</head><body>"
$out.println "<h1>#{$model.weblog.name}</h1>"
$model.weblogEntriesPager.entries.keySet().each {
    |day|
    $model.weblogEntriesPager.entries.get(day).each {
        |entry|
        $out.println "<h3>#{entry.title}</h3>"
        $out.println "<p>#{entry.text}</p>"
    }
}
$out.println "</body></html>"
```



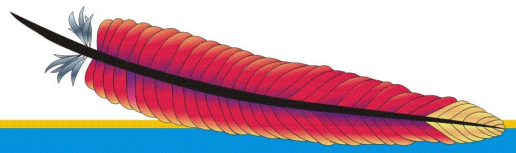
Example GSP page template

```
<html>
   <head>
      <title>${model.weblog.name}</title>
   </head>
   <body>
   <h1>${model.weblog.name}</h1>
   <i>${model.weblog.description}</i>
   <% map = model.getWeblogEntriesPager().getEntries();
      map.keySet().each() { %>
          <% map.get(it).each() { %>
            <h2>${it.title}</h2>
            <p>${it.text}</p><br />
          <% }
      }%>
   </body>
</html>
```

Groovy ex.: provisioning a blog

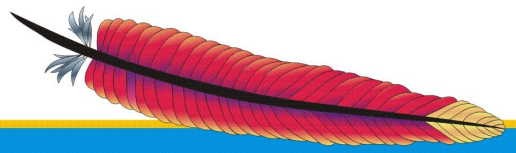
```
umgr = roller.getUserManager()
user = umgr.getUserByUserName(uname);

weblog = new org.apache.roller.weblogger.pojos.Weblog(
   handle, user, blogname, description,
   user.getEmailAddress(), user.getEmailAddress(),
   theme, user.getLocale(), user.getTimeZone());

umgr.addWebsite(weblog);
roller.flush();

blog = umgr.getWebsiteByHandle(handle);
println "Created blog ${blog.name} with id ${blog.id}";
```

// For more examples: http://tinyurl.com/2l2no4

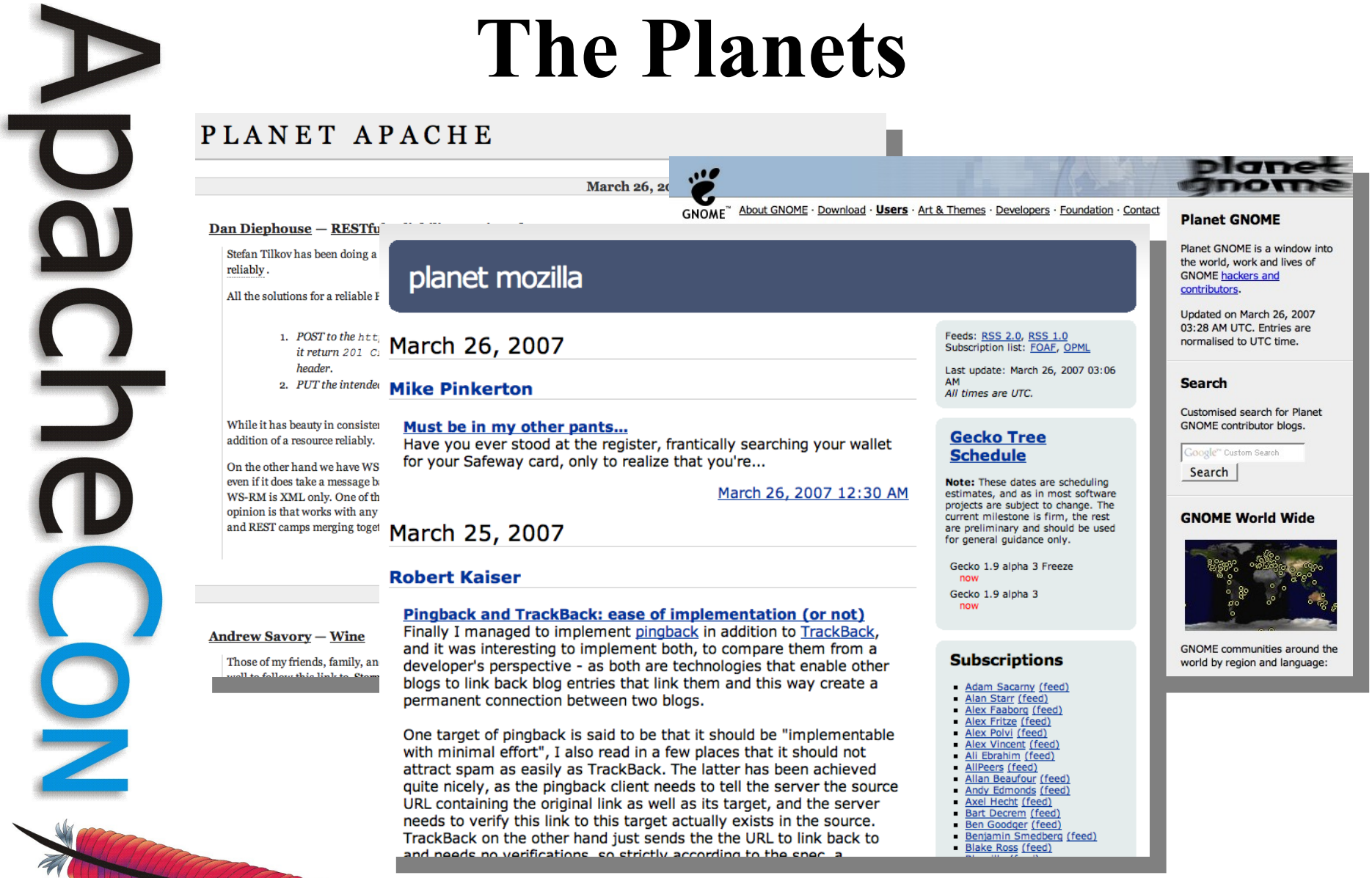


Planet server aggregation

- Overview
- Blog, wiki and feed Web services
- Blog customization via templates
- Blog server customization via plugins
- Example: Scripting languages in Roller
- **Planet server aggregation**
- Use case: Software business blogs

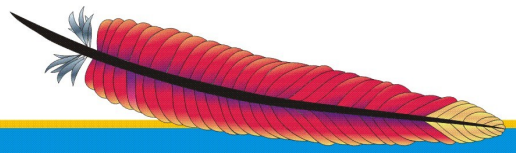


The Planets


PLANET APACHE
March 26, 2
Dan Diephouse — RESTfu
Stefan Tilkov has been doing a
reliably .
All the solutions for a reliable F
1. POST to the htt
it return 201 C
header.
2. PUT the intende
While it has beauty in consiste
addition of a resource reliably.
On the other hand we have WS
even if it does take a message b
WS-RM is XML only. One of th
opinion is that works with any
and REST camps merging toge
Andrew Savory — Wine
Those of my friends, family, an
GNOME
About GNOME · Download · Users · Art & Themes · Developers · Foundation · Contact
planet mozilla
March 26, 2007
Mike Pinkerton
Must be in my other pants...
Have you ever stood at the register, frantically searching your wallet for your Safeway card, only to realize that you're...
March 26, 2007 12:30 AM
March 25, 2007
Robert Kaiser
Pingback and TrackBack: ease of implementation (or not)
Finally I managed to implement pingback in addition to TrackBack, and it was interesting to implement both, to compare them from a developer's perspective - as both are technologies that enable other blogs to link back blog entries that link them and this way create a permanent connection between two blogs.
One target of pingback is said to be that it should be "implementable with minimal effort", I also read in a few places that it should not attract spam as easily as TrackBack. The latter has been achieved quite nicely, as the pingback client needs to tell the server the source URL containing the original link as well as its target, and the server needs to verify this link to this target actually exists in the source. TrackBack on the other hand just sends the the URL to link back to
Feeds: RSS 2.0, RSS 1.0
Subscription list: FOAF, OPML
Last update: March 26, 2007 03:06 AM
All times are UTC.
Gecko Tree Schedule
Note: These dates are scheduling estimates, and as in most software projects are subject to change. The current milestone is firm, the rest are preliminary and should be used for general guidance only.
Gecko 1.9 alpha 3 Freeze
now
Gecko 1.9 alpha 3
now
Subscriptions
Adam Sacarny (feed)
Alan Starr (feed)
Alex Faaborg (feed)
Alex Fritze (feed)
Alex Polvi (feed)
Alex Vincent (feed)
Ali Ebrahim (feed)
AllPeers (feed)
Allan Beaufour (feed)
Andy Edmonds (feed)
Axel Hecht (feed)
Bart Decrem (feed)
Ben Goodger (feed)
Benjamin Smedberg (feed)
Blake Ross (feed)
Planet GNOME
Planet GNOME is a window into the world, work and lives of GNOME hackers and contributors.
Updated on March 26, 2007 03:28 AM UTC. Entries are normalised to UTC time.
Search
Customised search for Planet GNOME contributor blogs.
Search
GNOME World Wide
GNOME communities around the world by region and language:

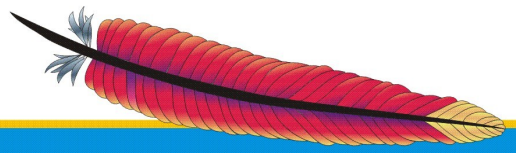
What's a Planet?

- A “community aggregator”
- Portal-like web application which displays weblog posts from a group of closely related but separately hosted weblogs
- Provides aggregated feeds so that readers may subscribe to the group as a whole
- Most popular?
  - Planet (Python)
  - See also: Sam Ruby's Venus fork of Planet



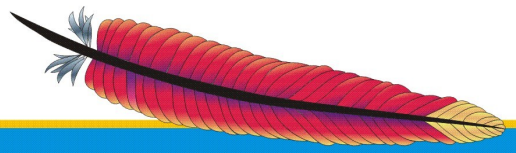
Roller Weblogger & Planet

- Roller Weblogger has built-in aggregation
  - A planet in a blog Server
  - Used on blogs.sun.com
- Roller Planet: standalone planet server
  - Multi-Planet server with web UI
  - Based on Roller architecture
  - Powers planets.sun.com



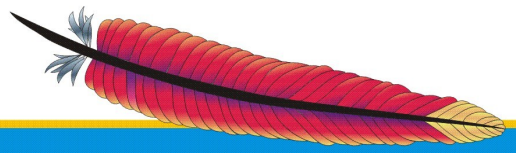
Agenda

- Overview
- Blog, wiki and feed Web services
- Blog customization via templates
- Blog server customization via plugins
- Example: Scripting languages in Roller
- Planet server aggregation
- **Use case: Software business blogs**

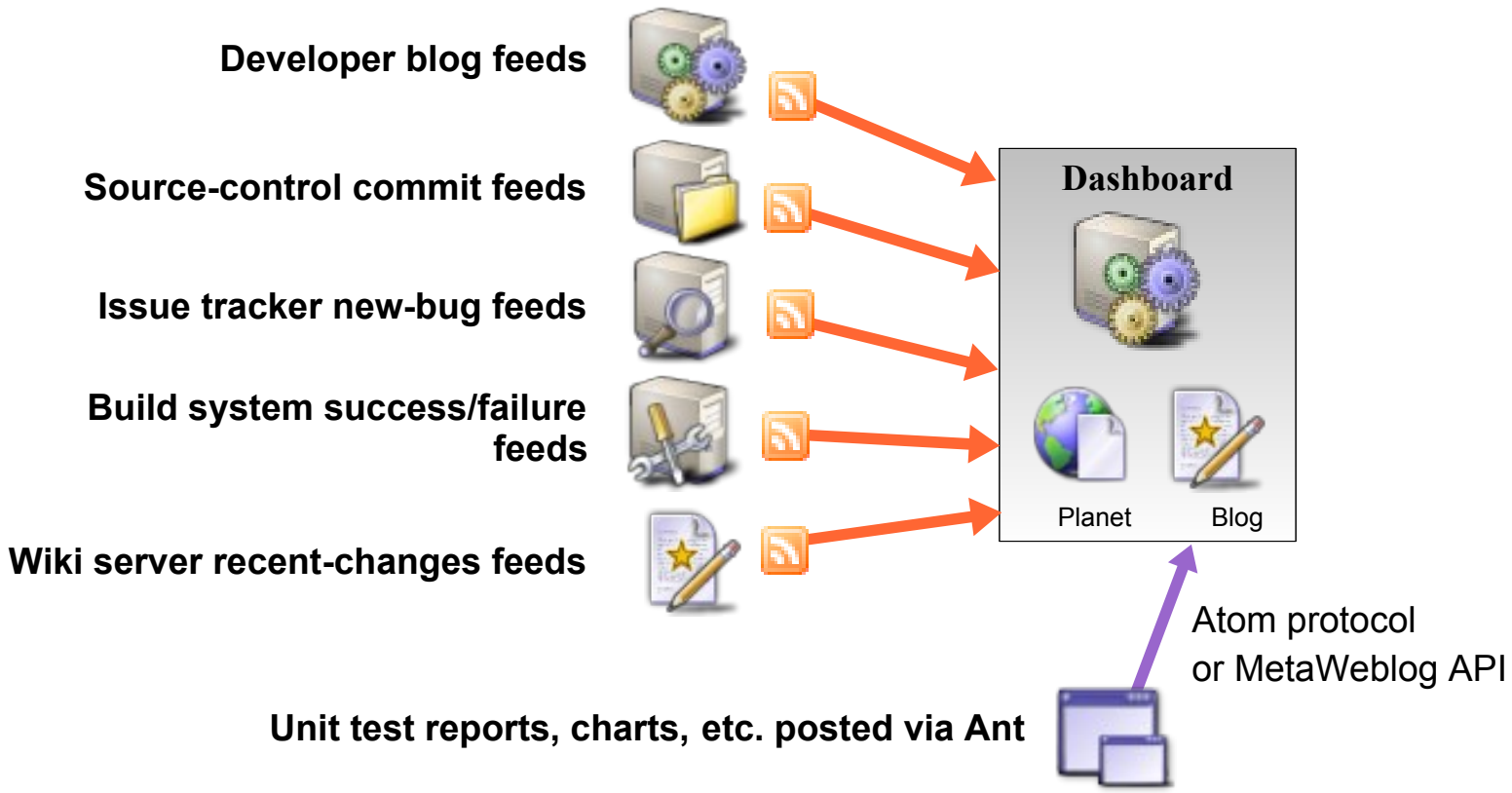


Blogs in the development dept.

- Development dashboard
- Inputs
  - Feeds from developer systems
  - Test results published by Ant
- Outputs
  - Planet site for developer blogs
  - Variety of aggregated & filtered feeds



Development Dashboard

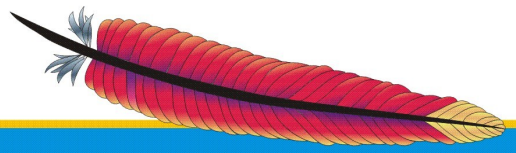


Roller project's dev-dashboard

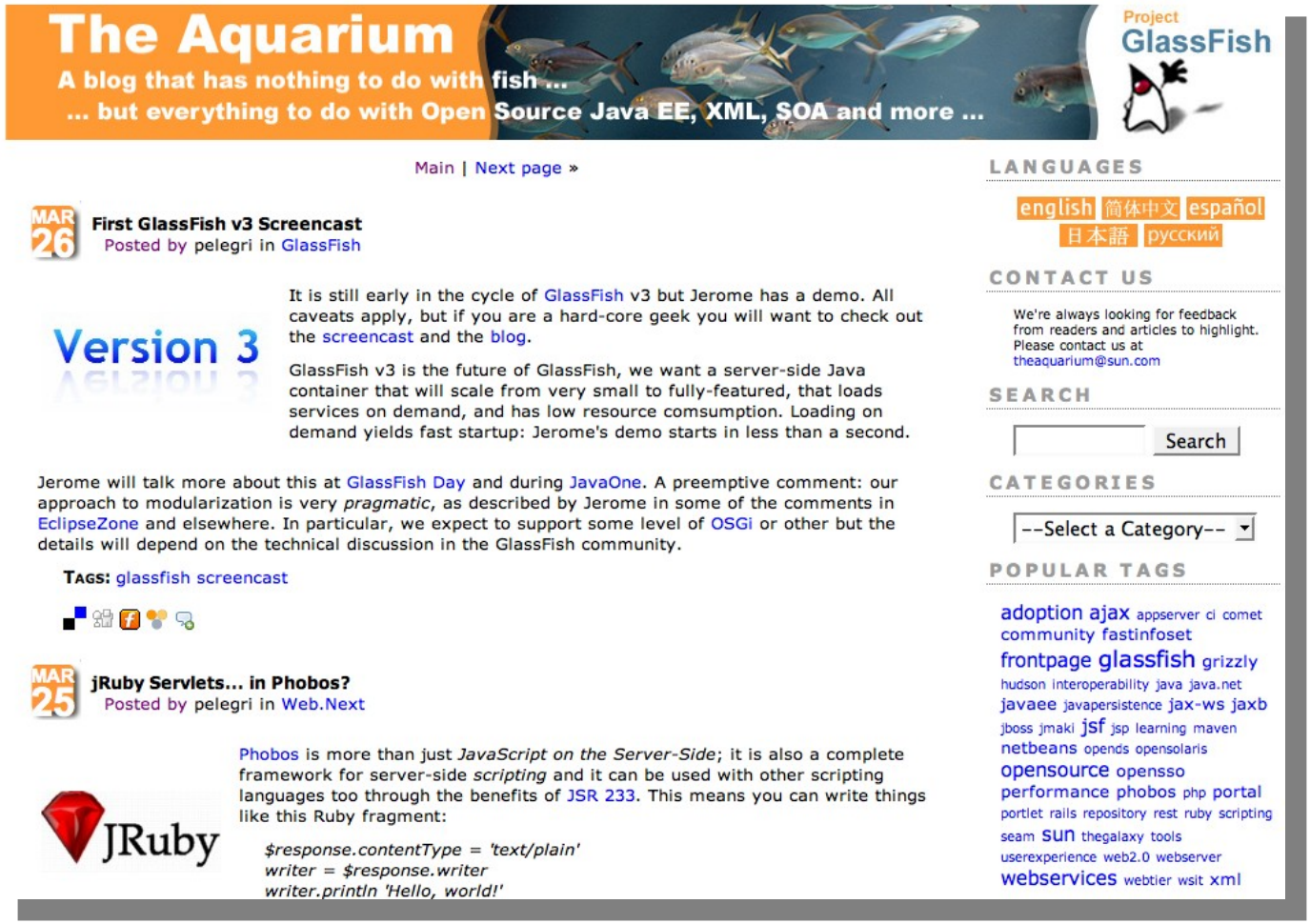
Blogs in the marketing dept.

- Product news & links blog
- Developer & evangelist blogs
- Employee blogs via aggregation
- User blogs via keyword blog-search
- Tag based feeds for the online-store

- All self-service web style

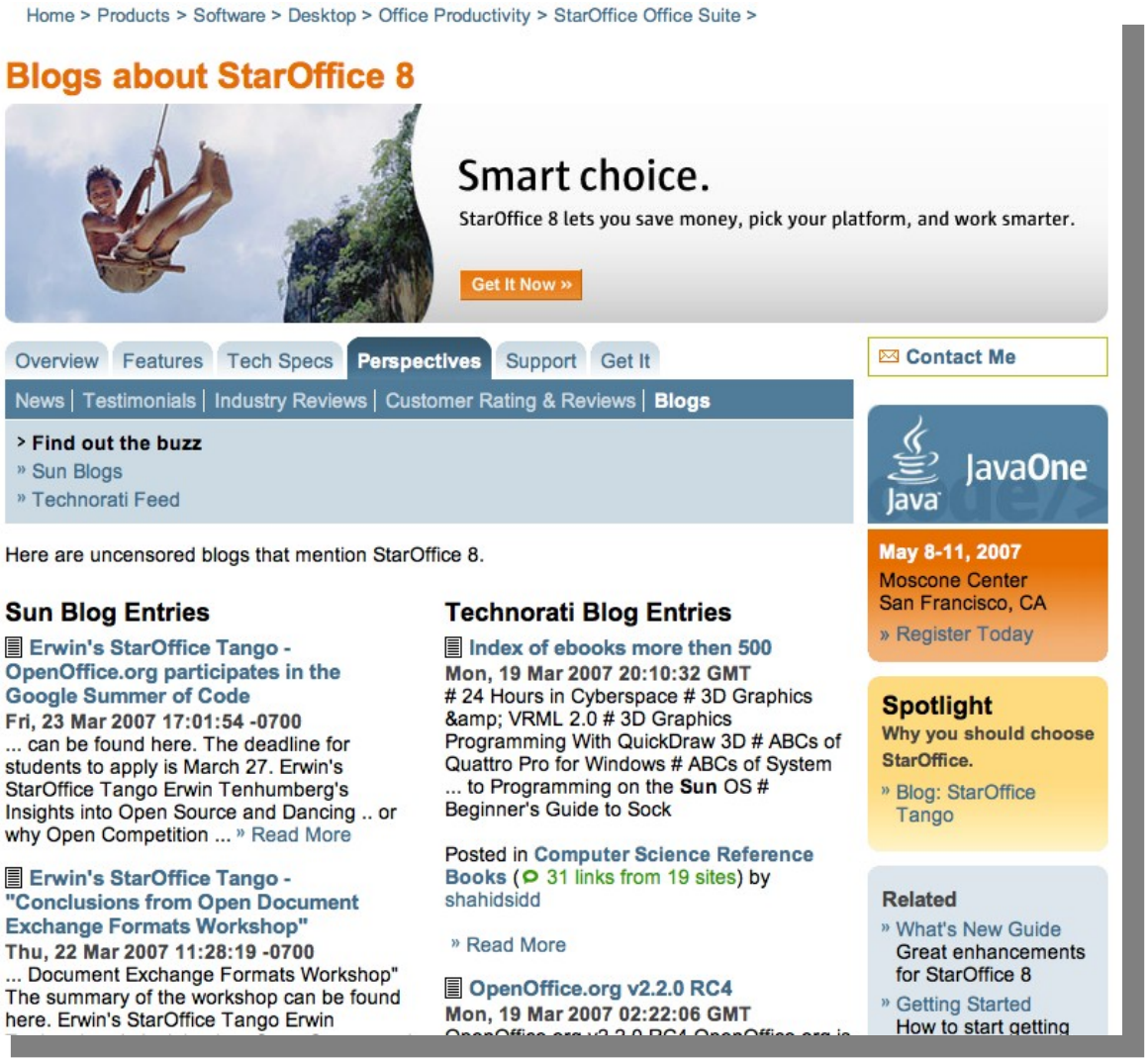


Product news & links blogs



Developer & evangelist blogs

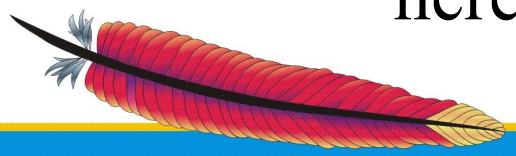
Blogs in the on-line store



Summary

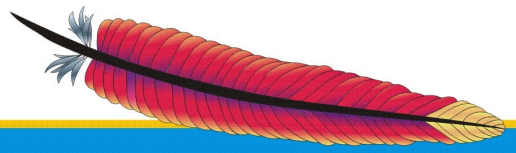
- Put blogs, planets, wikis and feeds in place
- Support and encourage their use
- Trust and empower your users/employees
- Enable the self-service web

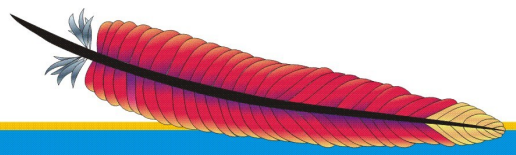
- Lend a hand with development of
  - Themes, customizations & plugins for your users
  - Development of Roller, Abdera and JSPWiki here at the Apache Software Foundation



For more information

- Apache Roller
  - http://roller.apache.org
  - http://roller.dev.java.net (additional plugins & themes)
- Apache JSPWiki
  - http://incubator.apache.org/jspwiki
- Apache Abdera (Incubating)
  - http://incubator.apache.org/abdera/
- Project ROME and ROME Propono
  - http://rome.dev.java.net



Q & A time

Questions?