ROLLER: AN OPEN SOURCE BLOGGING PLATFORM

**Dave Johnson**

Staff Engineer S/W
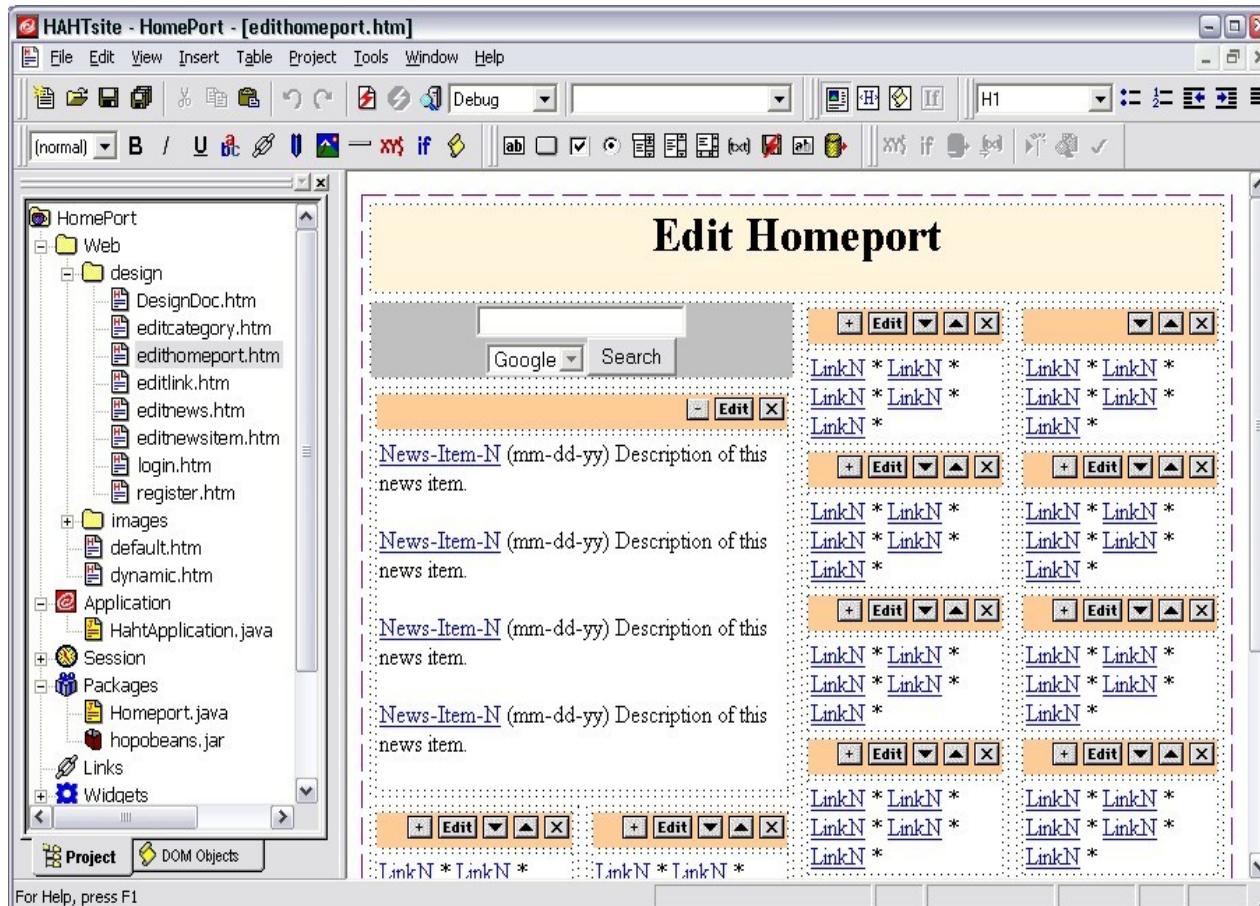
Sun Microsystems, Inc.

Sun microsystems

Agenda

- **Roller history**
- Roller features
- Roller community
- Roller internals: backend
- Roller internals: frontend
- Customizing Roller
- Roller futures

Roller started as an EJB example...

- Homeport – a home page / portal (2001)

... became an O'Reilly article

- Ditched EJBs and HAHTsite IDE (2002)
- Used all open source tools instead and thus...

O'REILLY®
ONJava.com™
The Independent Source for Enterprise Java

Published on ONJava.com (http://www.onjava.com/)
http://www.onjava.com/pub/a/onjava/2002/04/17/wblogosj2ee.html
See this if you're having trouble printing code examples

Building an Open Source J2EE Weblogger

by David Johnson
04/17/2002

As a Java developer, you should be aware of the tremendous wealth of open source development software that is available for your use -- even if you have no desire to release any of your own software as open source. In this article, I will introduce you to some of the most useful open source Java development tools by showing you how I used these tools to develop a complete database-driven Web application called Roller.

... and escaped into the wild

I am allowing others to use my installation of Roller for their weblogging. Hopefully this will provide a means for enhancing the Roller user base as well as provide a nice environment for communication and expression.

Anthony Eden
August 8, 2002

... and found a new home at Apache

- Apache Roller (incubating)
  > Incubation period: June 2005 - ???

Agenda

- Roller history
- **Roller features**
- Roller community
- Roller internals: backend
- Roller internals: frontend
- Customizing Roller
- Roller futures

Roller features: standard blog stuff

- Individual and group blogs
- Hierarchical categories
- Comments, trackbacks and referrers
- File -upload and Podcasting support
- User editable page templates
- RSS and Atom feeds
- Blog client support (Blogger / MetaWeblog API)
- Built-in search engine

Sun microsystems

Edit Entry

Create & Edit   Preferences

New Entry | Entries | Comments | Categories | Bookmarks | File Uploads | Referrers

Editing weblog entry in weblog roller

Title   Status, CC: World

Status   **Published (Last updated 6/3/06 12:38 AM)**

Category   /Open Source

Pub Time   23 : 13 : 14   6/1/06   America/New_York

Permalink   http://rollerweblogger.org/page/roller?entry=status_cc_world8

Summary (optional)

[Style] [Font] [Size]

B *I* U

It's been a while since my last status, CC:World post, so here's an update on the things I'm working on these days.

☐ View Source

Content

[Style] [Font] [Size]

B *I* U

It's been a while since my last status, CC:World post, so here's an update on the things I'm working on these days.

RSS and Atom in Action. Because I added those two new chapters on Windows RSS and ROME, Manning had to renumber about 3/4 of the book. That took a bit longer than expected, but now the work is done and I've got the whole book in one big PDF file. I'll do one final review this weekend and, if we can quickly wrap up the loose ends and the index, we'll be off to the printers before the end of June.

☐ View Source

Post to Weblog   Save as Draft   Delete Entry   Preview

Comments
None

Pending Entries
None

Recent Drafts
John Lennon invented the iPod ...
Mugshot.org: Red Hat social network...

Recent Entries
Today's links [June 03, 2006] ...
User-written product reviews on sun...
Apache Roller 2.3 (incubating) rele...
Status, CC: World
ROME and Abdera
Get rollin' with Sun Java Syst...
Today's links [May 25, 2006] ...
And speaking of SAS
There is life after SAS
Today's links [May 24, 2006] ...
Netbeans 5.5 (beta) and other new s...
+1 to Atom Reference Implementation...
JavaOne factoids
Progress towards open source Java ...
JavaOne: putting the web back in we...
JavaOne arrival
Today's links [May 12, 2006] ...
Roller Admin Protocol
JavaOne 2006 Bloggers meet-up: time...
WS-* Kool-Aid and JavaOne ...

Powered by Struts

Blog client support
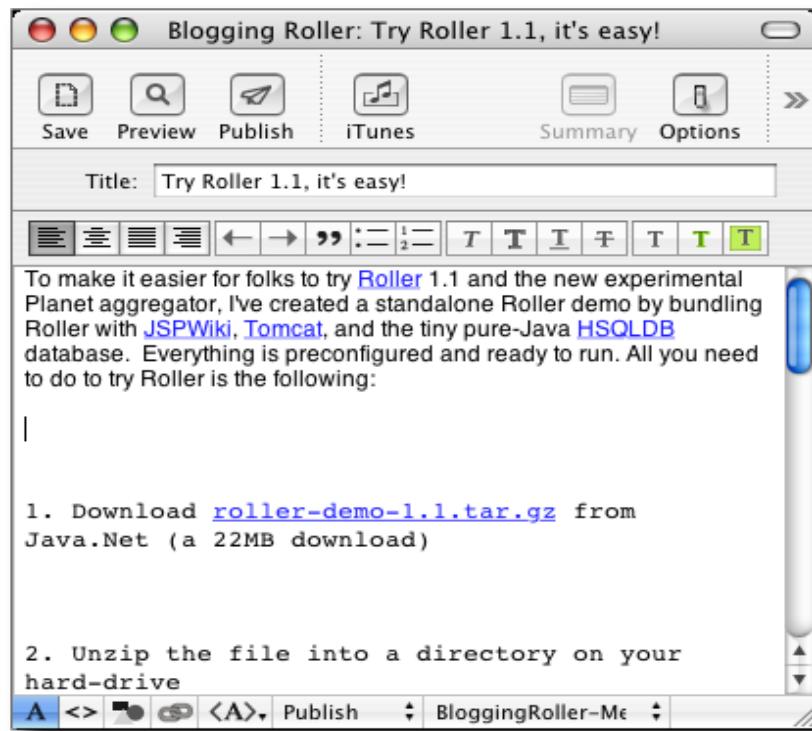
- XML-RPC based Blogger and MetaWeblog API
- Lots of blog clients work with Roller, for example:

ecto

http://ecto.kung-foo.tv

For Mac OSX and Windows

Multiple blogs per user

Multiple users per blog

Roller features: conversation support

- Get feedback via **comments** and **trackbacks**
  - > Enable / disable them at entry, blog and site-wide levels
  - > Email notification of new comments and trackbacks
  - > Comment moderation for both

- Talk back by sending **pings** and trackbacks
  - > Send trackbacks to blog posts that you're writing about
  - > Ping any set of Weblogs.com compatible services

- Promote your favorite web sites on your **blogroll**
  - > Bookmark management with OPML import and export

- See where today's hits are coming from
  - > Display of days **referrers** and referrer hit counts

Planet-style community aggregator

- To include local and off-site blogs on front page

Roller features: developer features

- Web services support
  - > Blogger and MetaWeblog API (XML-RPC based)
  - > IETF Atom Publishing Protocol (draft 8)
  - > Roller Administration Protocol (REST based)

- Themes
  - > Add your own blog themes
  - > Enable / disable user editing of themes

- Plug-ins allow you to:
  - > Hook into to entry rendering process (e.g. allow wiki syntax)
  - > Provide alternative comment authenticators (e.g. CAPCHA)
  - > Provide alternative persistence providers (e.g. JDO, JPA)

Limitations

- Relatively small community compared to Wordpress
  - > Small number of plug-ins and themes

- Java dependency limits ISP options
  - > A good and Java-smart ISP can be hard to find

- Clustering requires some custom development
  - > At BSC, we use **memcached** and our own search engine

- SSO requires some custom development
  - > NSCU, Yale and others have done it

Why choose Roller?

- Proven full-featured blogging solution for big sites
  - > Sun, IBM and others depend on it for corporate blogging

- Standard Java web application architecture

- Open source and Apache licensed

- Active and growing community at Apache

Agenda

- Roller history
- Roller features
- **Roller community**
- Roller internals: backend
- Roller internals: frontend
- Customizing Roller
- Roller futures

What is the Roller community?

- The developers who write the code?
  - > Seven *committers* design, develop, test and release the code

- The system admins who build and run the sites?
  - > Big sites: JRoller, blogs.sun.com, IBM devWorks, Yale, NCSU
  - > Tens and perhaps hundreds of sites using Roller

- The bloggers who use the software?
  - > Thousands and perhaps tens of thousands of active bloggers

- A: all of the above

Roller community processes

- Challenges
  - > Part of community needs to move fast (monthly releases)
  - > Moving too fast can scare away new contributors

- Need a flexible monthly release cycle

- Need transparent planning
  - > To encourage contributors, enable collaboration
  - > Plus, it's the open source way...

Roller release cycle

- Plan allows monthly releases, but doesn't mandate

  - > Second to last Thursday of month is RC day
    - > Committers: think we need a month release?
    - > Then create an RC on RC day and call for week of testing

  - > Last Thursday of month is release day
    - > Optionally: create release branch in SVN
    - > Create another RC and if confident: call for release vote
    - > Once released, create a SVN tag

  - > Rhythm can help us sync with new contributors

Roller planning process: features

- **Want to add a feature?**
- Write a proposal on the wiki
- Coordinate with folks writing other similar proposals
- Introduce proposal on dev list, call for feedback
- Tweak proposal, get consensus
- Do the work
- Submit patch or commit code to appropriate branch

Roller planning process: releases

- **Want to make a release?**
- Write a proposal on the wiki
  - > Reference all JIRA issues to be addressed
  - > Reference all feature proposals to be included
- Tweak proposal, get consensus
- Do the work
- Follow the release cycle: on RC day, create an RC

Great, so why is Roller still incubating?
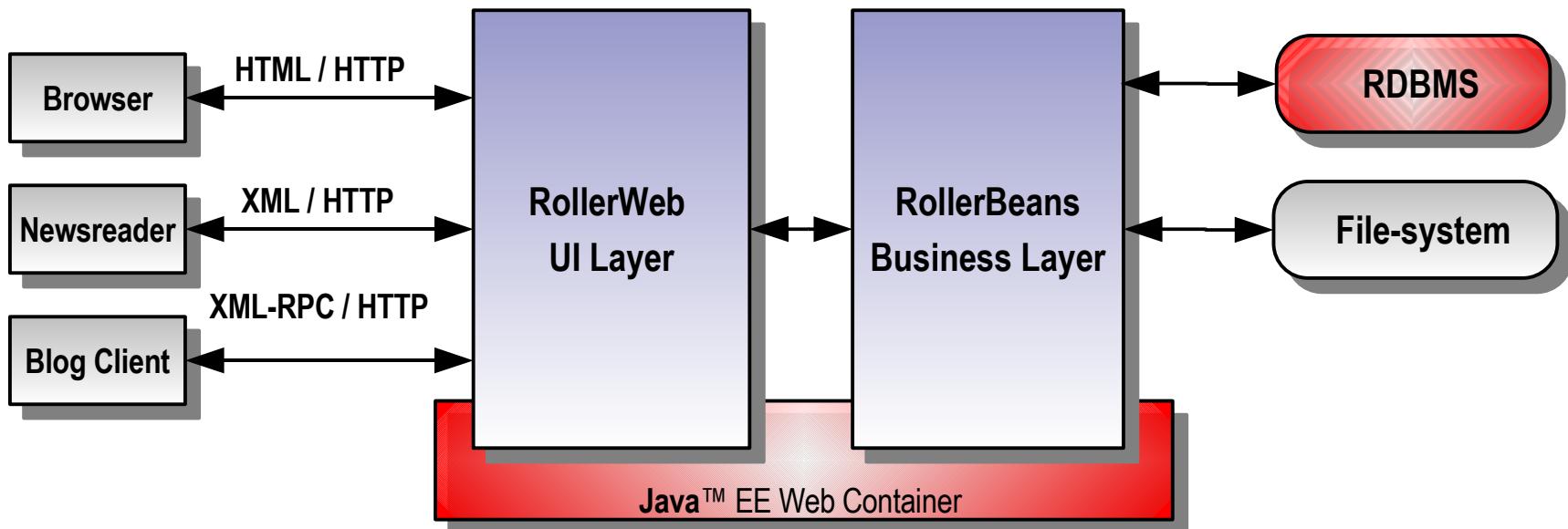
- ✅ Contributor agreements and grants
- ✅ Apache license headers
- ✅ Use Apache package names
- ✅ Migrate infrastructure
  - ✅ Source code and mailing lists
  - ⏰ Blog, wiki and bug tracker (post graduation?)
- ✅ Diverse and growing community
- ✅ Team follows the "Apache way"
- ⚠️ Do **???** about LGPL dependencies

Agenda

- Roller history
- Roller features
- Roller community
- **Roller internals: backend**
- Roller internals: frontend
- Customizing Roller
- Roller futures
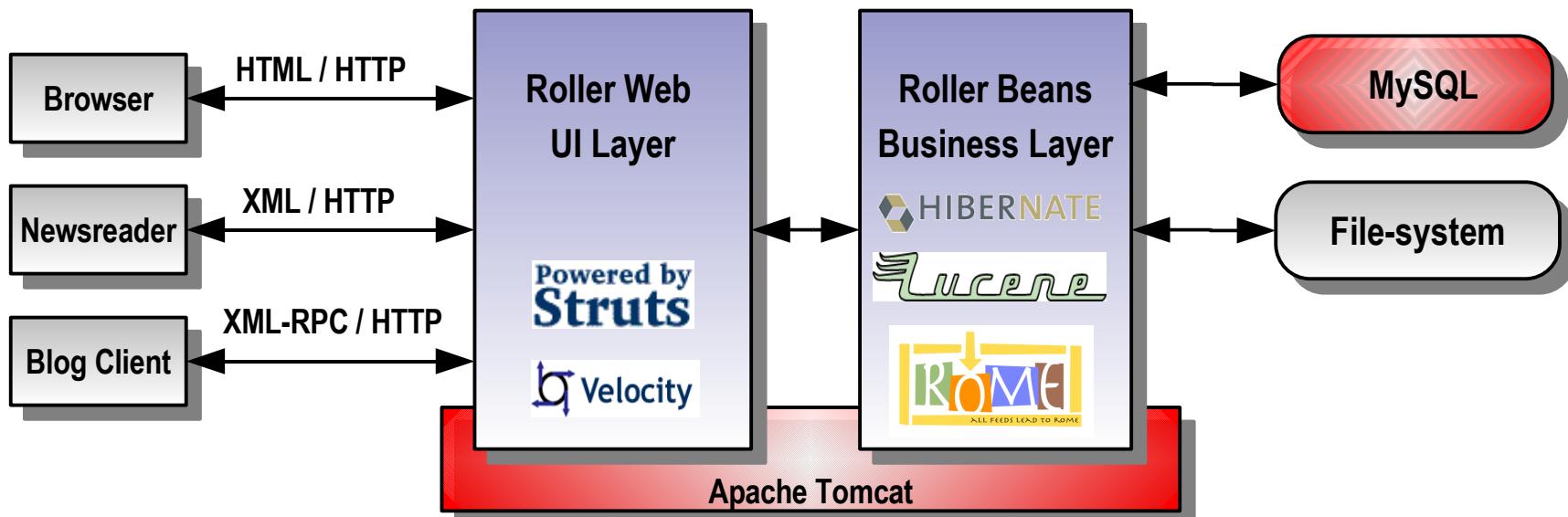
Enterprisey level architecture

- ## Web UI via Java™ Servlets and JSP
  - > Front controller, Web MVC and open session in view patterns

- ## Persistence via Java™ Database Connectivity
  - > Factory, facade and data mapper patterns

Java geek level architecture

- Roller Web: Web and UI layer
  - > Editor UI via **Struts and JSP,** blog and feed rendering via **Velocity**
  - > Feed parsing via **ROME**, Blogger API via **Apache XML-RPC**

- Roller Beans: Business and persistence layer
  - > **Hibernate** for DBMS, **Lucene** for search and file-system for uploads
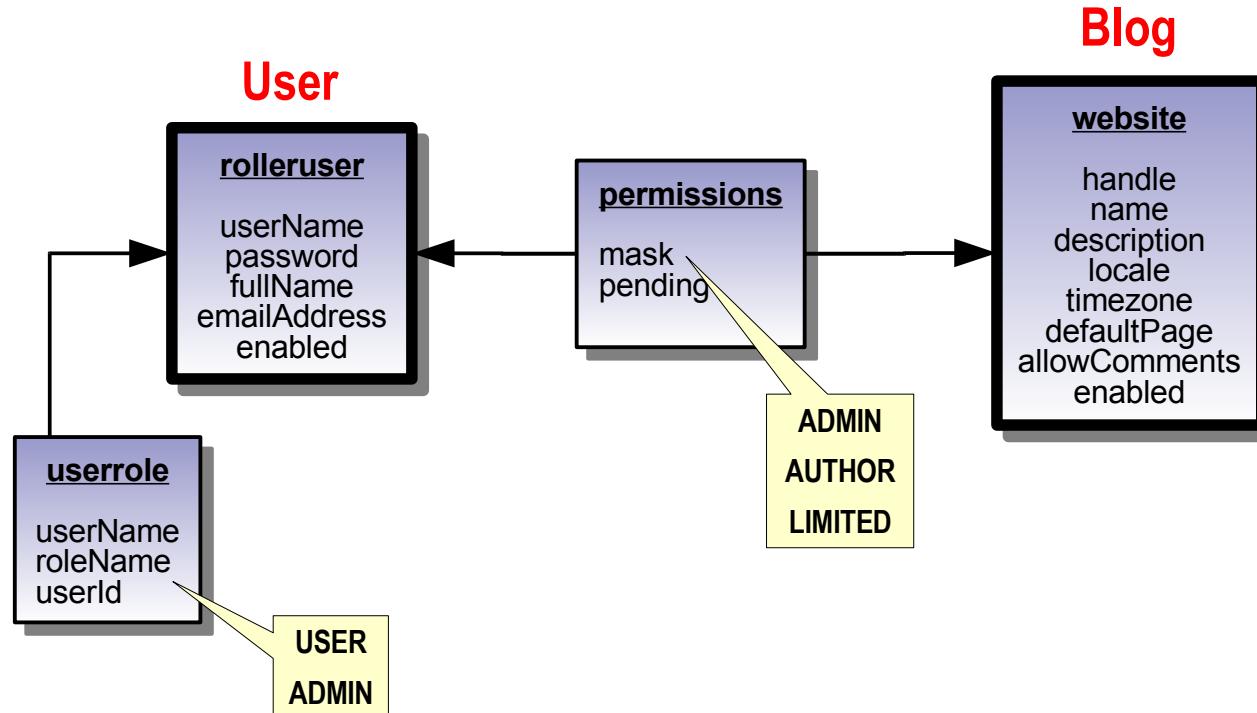
First, let's talk business

- Business layer is made up of:

  > Plain old **Java objects**
  >> i.e. POJOs

  > Interfaces that create, retrieve, update and delete
  >> i.e. **Manager interfaces** that do CRUD of POJOs
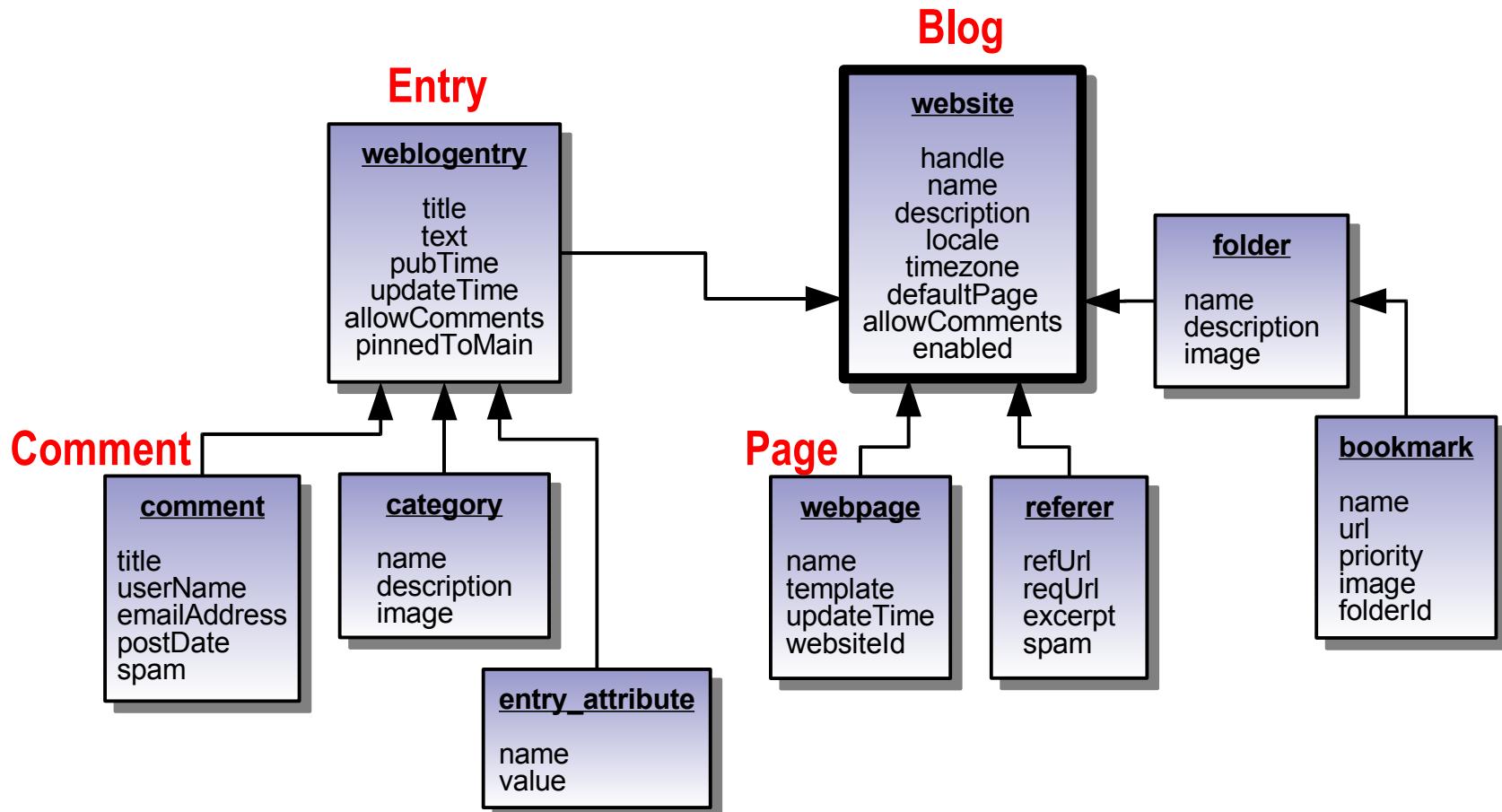
Roller POJOs: users and blogs

- User has global role
- User has permissions in blogs

Roller POJOs: blogs and entries

- A blog has entries, pages and comments

Blog

Entry

website

handle
name
description
locale
timezone
defaultPage
allowComments
enabled

weblogentry

title
text
pubTime
updateTime
allowComments
pinnedToMain

folder

name
description
image

Comment

comment

title
userName
emailAddress
postDate
spam

category

name
description
image

entry_attribute

name
value

Page

webpage

name
template
updateTime
websiteId

referer

refUrl
reqUrl
excerpt
spam

bookmark

name
url
priority
image
folderId

Roller POJO markup

Hibernate mapping generation

```java
/** @hibernate.class table="website"
  * @struts.form include-all="true"
  */
public class WebsiteData extends org.roller.pojos.PersistentObject
    implements java.io.Serializable {

  /**
   * @hibernate.property column="description"
   *    non-null="false" unique="false"
   * @roller.wrapPojoMethod type="simple"
   */
  public String getDescription() {
     return this.description;
  }
  public void setDescription(String description) {
     this.description = description;
  }
  ...
```
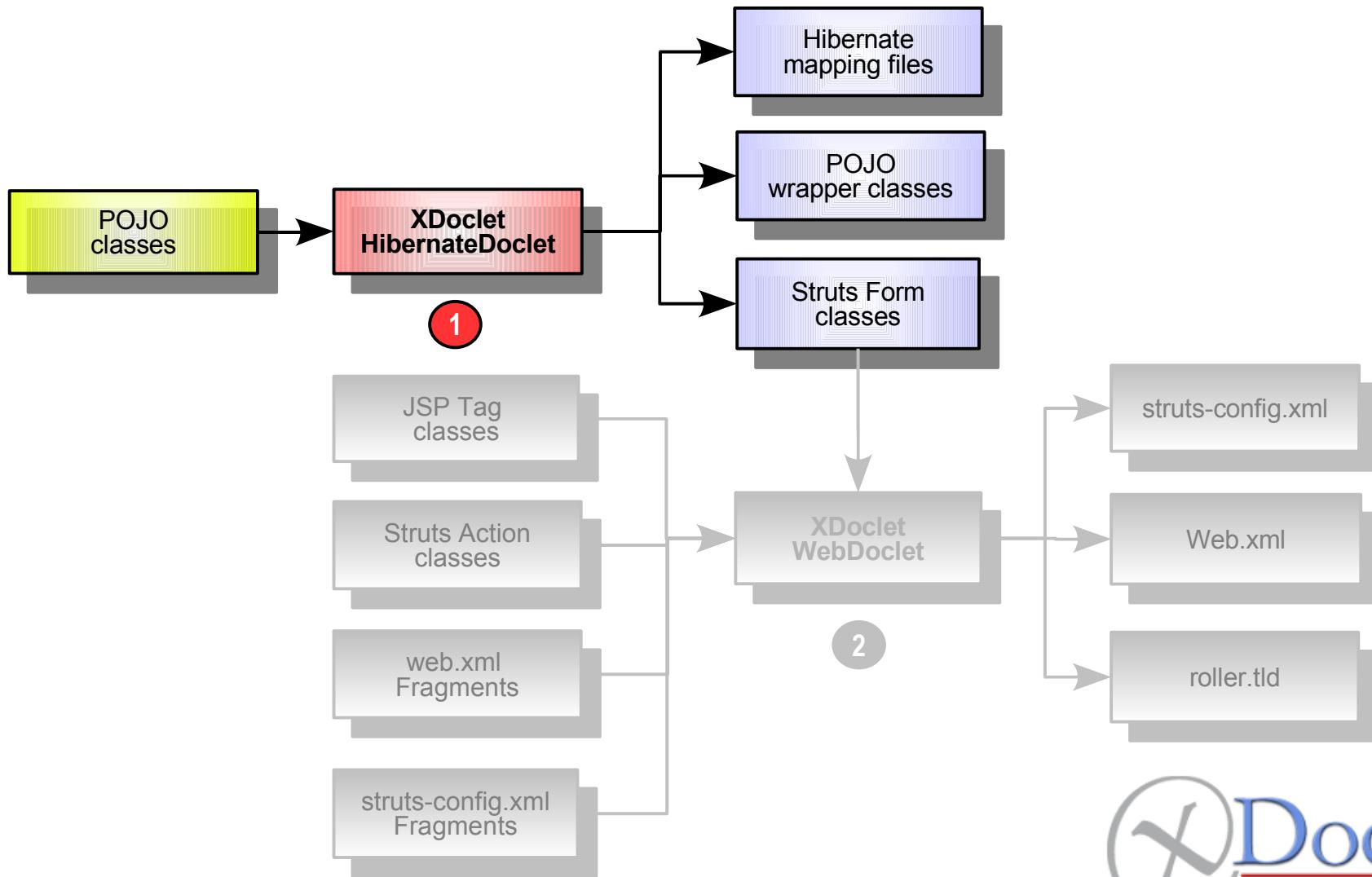
Struts form bean generation
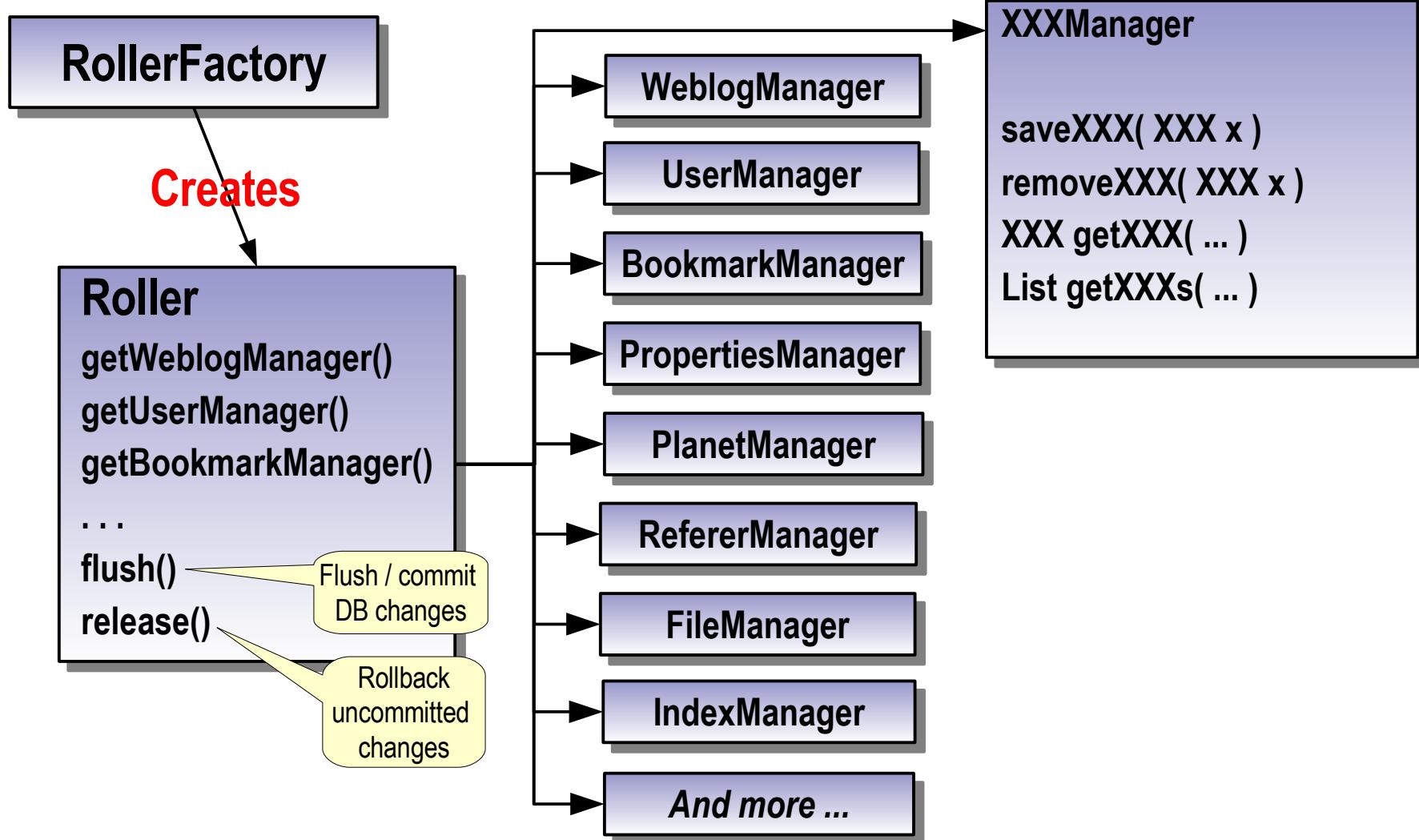
Hibernate mapping generation

POJO wrapper generation

XDoclet
Attribute Oriented Programming

XDoclet code gen, phase 1

```
POJO classes  →  XDoclet HibernateDoclet  (1)
```

XDoclet HibernateDoclet →
- Hibernate mapping files
- POJO wrapper classes
- Struts Form classes

Struts Form classes ↓

JSP Tag classes
Struts Action classes
web.xml Fragments
struts-config.xml Fragments
→ XDoclet WebDoclet (2) →
- struts-config.xml
- Web.xml
- roller.tld
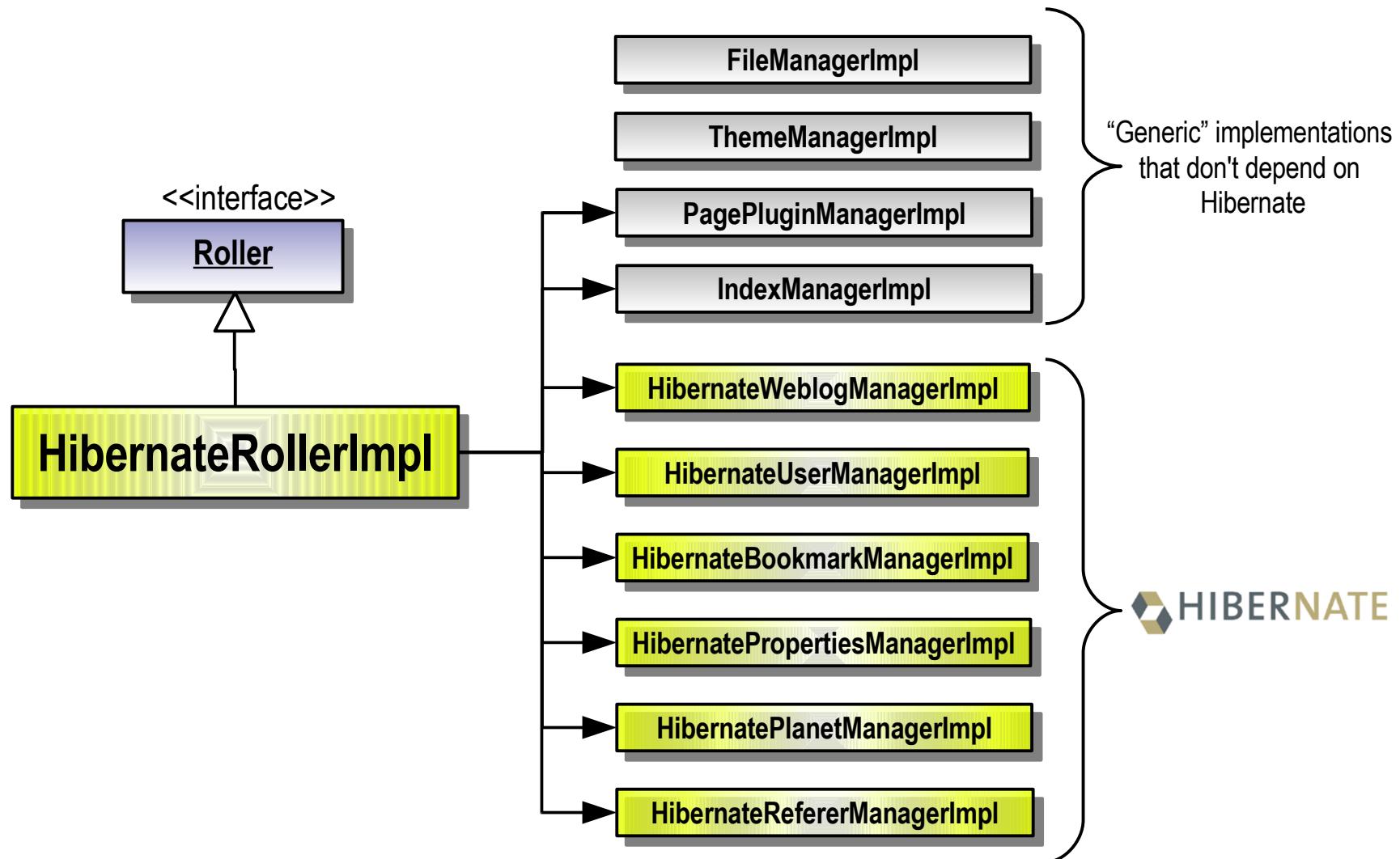
XDoclet
Attribute Oriented Programming
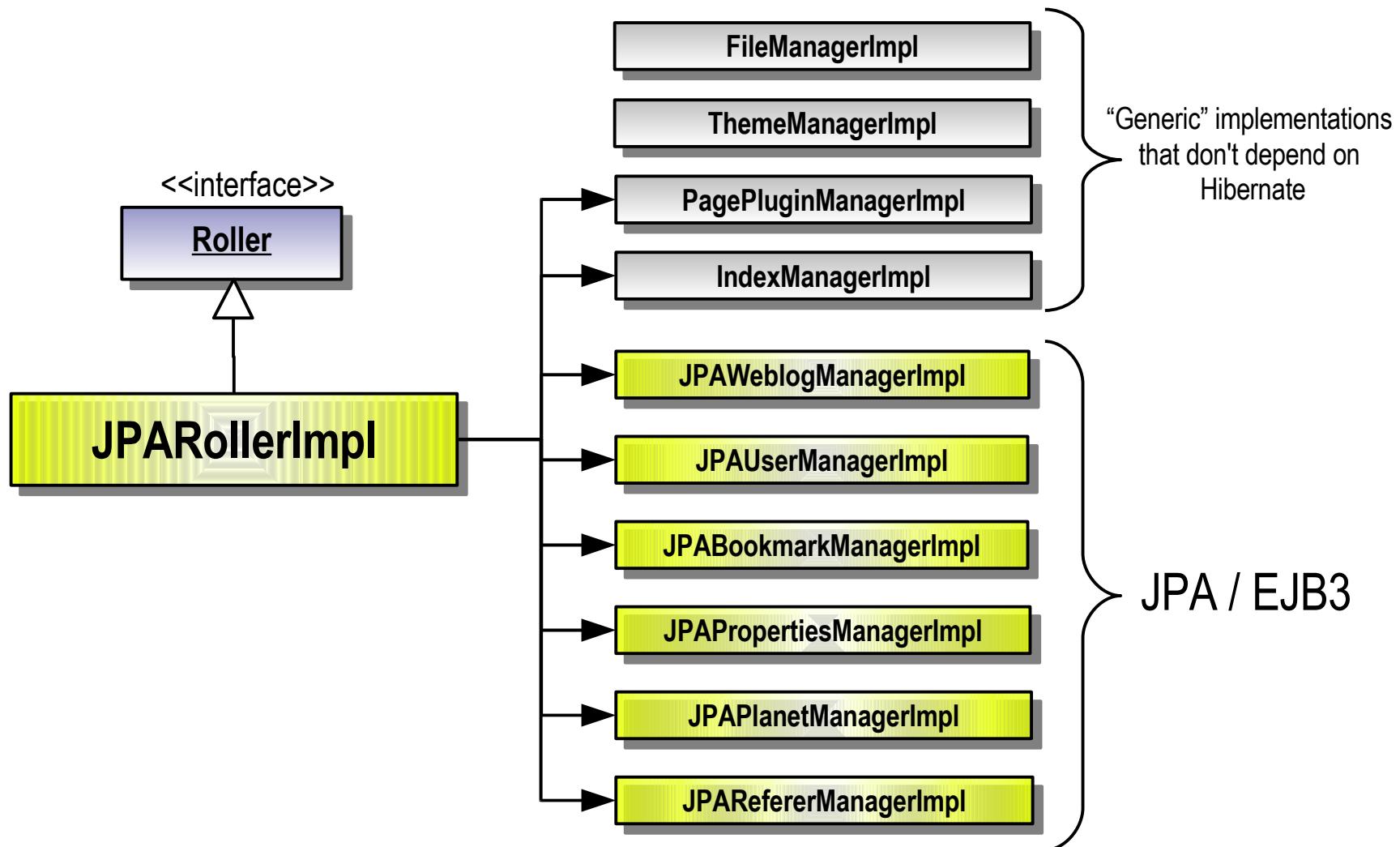
The Roller manager interfaces

RollerFactory

Creates

Roller

getWeblogManager()

getUserManager()

getBookmarkManager()

. . .

flush() — Flush / commit DB changes

release() — Rollback uncommitted changes

WeblogManager

UserManager

BookmarkManager

PropertiesManager

PlanetManager

RefererManager

FileManager

IndexManager

And more ...

XXXManager

saveXXX(XXX x)

removeXXX(XXX x)

XXX getXXX(...)

List getXXXs(...)

The Hibernate implementation

```
<<interface>>
   Roller
```

HibernateRollerImpl

FileManagerImpl

ThemeManagerImpl

PagePluginManagerImpl

IndexManagerImpl

"Generic" implementations that don't depend on Hibernate

HibernateWeblogManagerImpl

HibernateUserManagerImpl

HibernateBookmarkManagerImpl

HibernatePropertiesManagerImpl

HibernatePlanetManagerImpl

HibernateRefererManagerImpl

HIBERNATE

Someday... the JPA implementation

FileManagerImpl

ThemeManagerImpl

PagePluginManagerImpl

IndexManagerImpl

"Generic" implementations that don't depend on Hibernate

<<interface>>
Roller

JPARollerImpl

JPAWeblogManagerImpl

JPAUserManagerImpl

JPABookmarkManagerImpl

JPAPropertiesManagerImpl

JPAPlanetManagerImpl

JPARefererManagerImpl

JPA / EJB3

So, for example...

```
Roller roller = RollerFactory.getRoller();
WeblogManager mgr = roller.getWeblogManager();

WeblogEntryData entry = new WeblogEntryData();

entry.setTitle("My first blog post");
entry.setText("&lt;p&gt;Hi everybody!&lt;p&gt;");
entry.setAnchor("hi_everybody");
entry.setPubTime(new Date());
entry.setUpdateTime(new Date());
entry.setWebsite(weblog);
entry.setCreator(user);
entry.setCategory(weblog.getDefaultCategory());

mgr.saveWeblogEntry(entry);
```

Agenda

- Roller history
- Roller features
- Roller community
- Roller internals: backend
- **Roller internals: frontend**
- Customizing Roller
- Roller futures

Two parts of the Roller UI
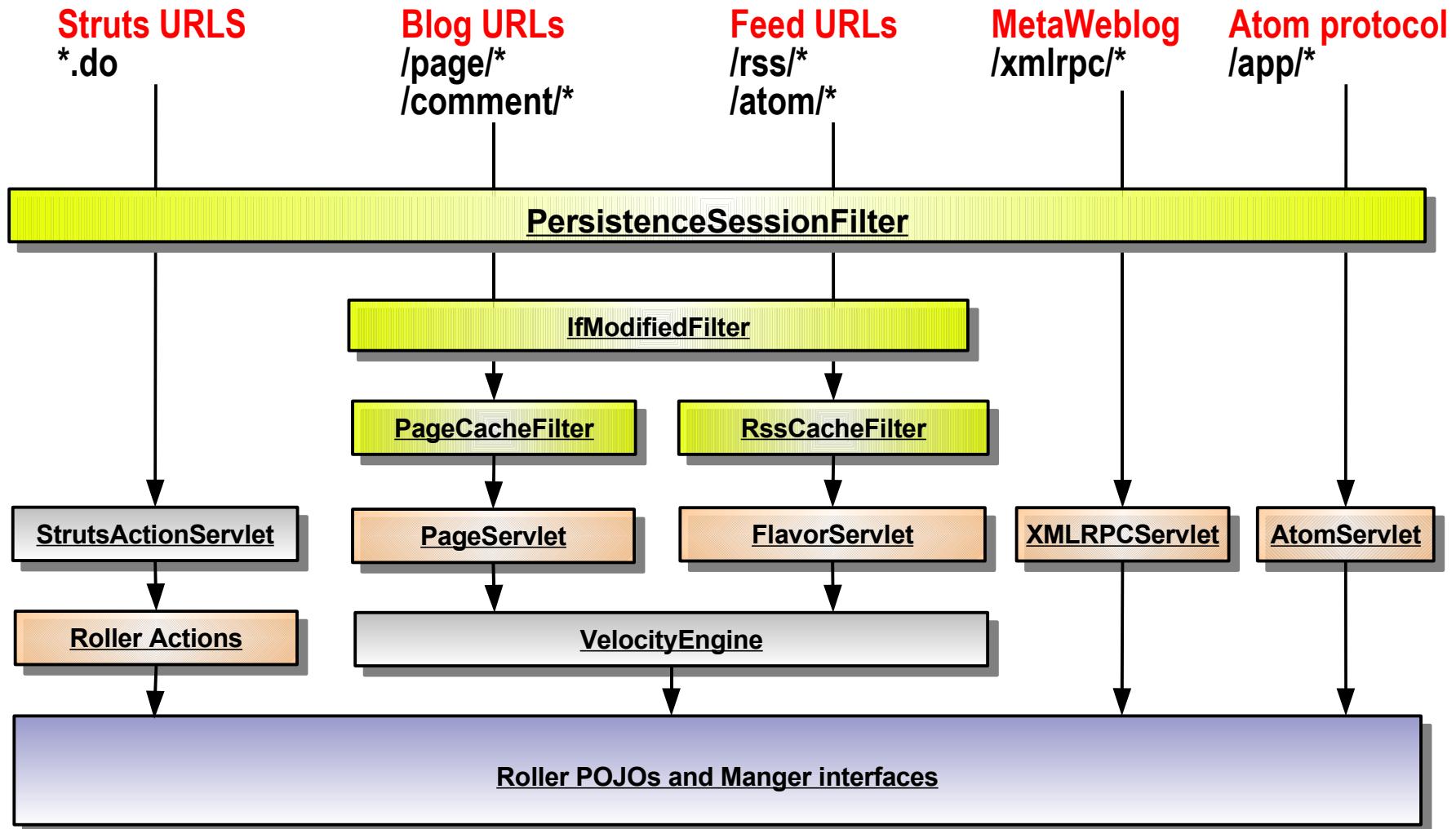
- **#1 - The blog and page rendering engine**
  - > Each blog's layout & style defined by Velocity templates
  - > Templates call Roller page models and Velocity macros

- **#2 - The authoring and admin UI**
  - > Classic Struts/Tiles and JSP user interface

- Both call Roller POJOs and manager interfaces

Open session in view

- The *Open session in view* pattern
- All requests go through `PersistenceSessionFilter`

```java
Roller = RollerFactory.getRoller();
try {
    // Pass request into Roller request stack
    chain.doFilter(request, response);
}
catch (Throwable t) {
    // handle error
}
finally {
    // rollback any uncommitted changes
    roller.release();
}
```
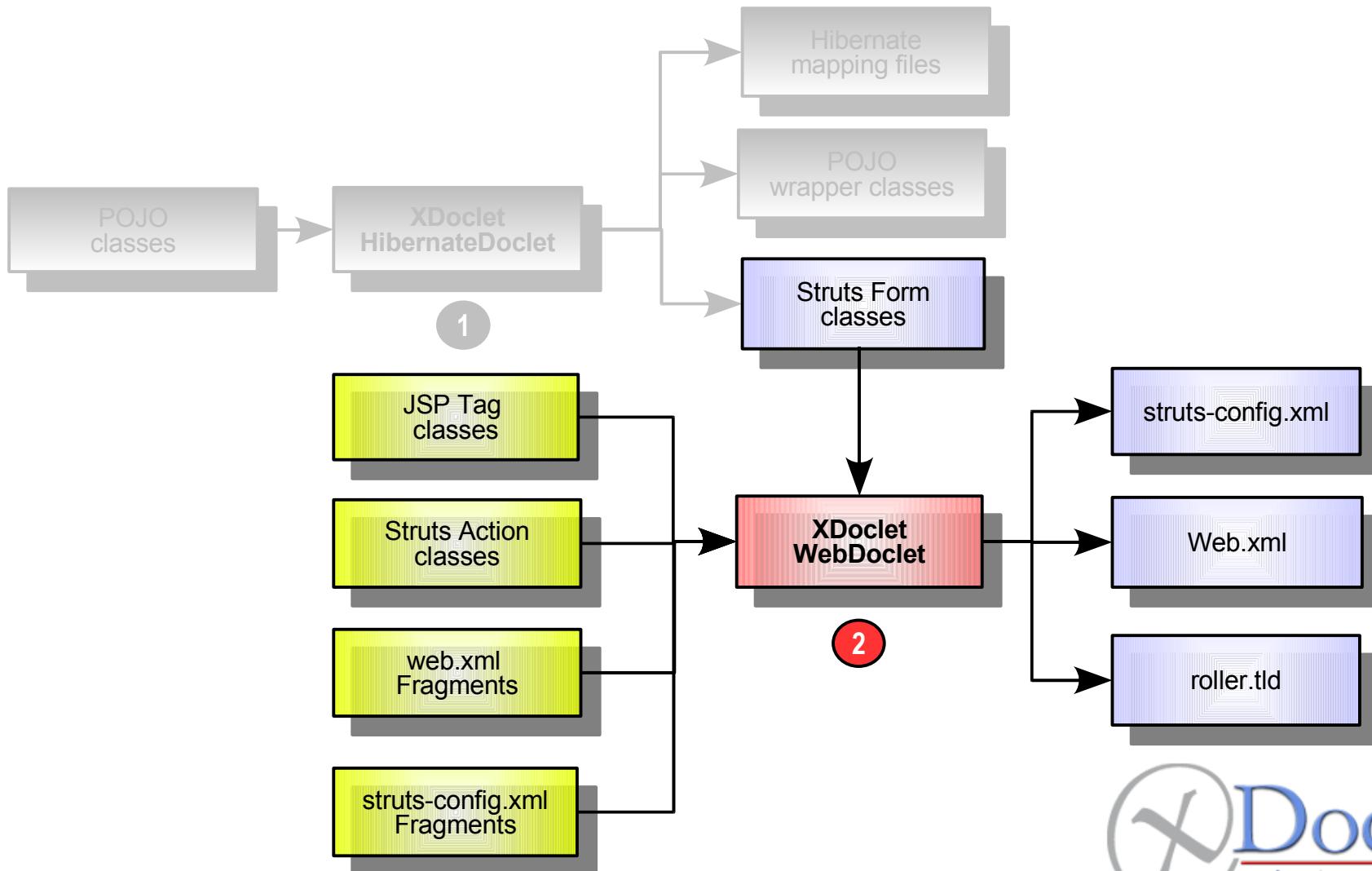
Web request stack

Struts URLS
*.do

Blog URLs
/page/*
/comment/*

Feed URLs
/rss/*
/atom/*

MetaWeblog
/xmlrpc/*

Atom protocol
/app/*

PersistenceSessionFilter

IfModifiedFilter

PageCacheFilter

RssCacheFilter

StrutsActionServlet

PageServlet

FlavorServlet

XMLRPCServlet

AtomServlet

Roller Actions

VelocityEngine

Roller POJOs and Manger interfaces

Struts actions markup

```java
/**
 * @struts.action path="/BookmarkSave" name="bookmarkFormEx"
 *    validate="true" input="/BookmarkEdit.do"
 *
 * @struts.action-forward name="Bookmarks"
 *    path="/Bookmarks.do?method=selectFolder"
 */
public class BookmarkSave extends Action {

    public ActionForward execute(
        ActionMapping       mapping,
        ActionForm          actionForm,
        HttpServletRequest  req,
        HttpServletResponse res) throws Exception {

        // save a bookmark

    }
}
```

XDoclet code gen, phase 2

POJO classes → XDoclet HibernateDoclet **(1)**

XDoclet HibernateDoclet → Hibernate mapping files

XDoclet HibernateDoclet → POJO wrapper classes

XDoclet HibernateDoclet → Struts Form classes

JSP Tag classes → XDoclet WebDoclet

Struts Action classes → XDoclet WebDoclet

web.xml Fragments → XDoclet WebDoclet

struts-config.xml Fragments → XDoclet WebDoclet

Struts Form classes → XDoclet WebDoclet **(2)**

XDoclet WebDoclet → struts-config.xml

XDoclet WebDoclet → Web.xml

XDoclet WebDoclet → roller.tld

XDoclet
Attribute Oriented Programming

Blog and feed display

- Each blog defined by user-editable templates
- Feeds defined by templates too, but not editable

- Why Velocity templates?
  - > Easy to limit objects available to users
  - > Easy to store templates in a database
  - > Velocity is easy to learn

Agenda

- Roller history
- Roller features
- Roller community
- Roller internals: backend
- Roller internals: frontend
- **Customizing Roller**
- Roller futures

As an end user you can

- Add categories, bookmarks to your blog
- Customize your blog templates
- Override Roller macros
- Create your own blog themes

Page templates & macros

- Each blog can define any number of pages

- Pages can use any of > 50 different macros

displayWeblogCalendar	showEntryPermalink	showSearchForm
evaluateString	showEntryText	showStatusMessage
formatDate	showImage	showThemeImage
genOpml	showLinkbacks	showThemeImagePath
includePage	showMenu	showThemePath
refererDisplayUrl	showNavBar	showThemePath
showBasicNavBar	showNewsfeed	showThemeScript
showBigWeblogCalendar	showOneBookmark	showThemeScriptPath
showBookmarks	showOpmlLink	showThemeStyle
showCommentForm	showPageDescription	showThemeStyleImport
showComments	showPageName	showThemeStylePath
showCommentsDiv	showPageUpdateTime	showTrackbackURL
showCommentsLink	showRSSAutodiscoveryLink	showVersion
showCommentsPageLink	showRSSBadge	showWeblogCalendar
showContentLanguage	showRSSLinks	showWeblogCategoryChooser
showContentType	showRecentEntries	showWeblogEntries
showDayPermalink	showReferers	showWebsiteDescription
showEditorNavBar	showResourceImage	showWebsiteTitle
showEntryDate	showResourcePath	spaces
showEntryDescription		

Velocity

Example blog page template

```html
<html>
<head>
   <title>#showWebsiteTitle()</title>
   #showRSSAutodiscoveryLink()
</head>
<body>
<table border="0" align="center" width="95%"><tbody><tr>
   <td class="entries" width="80%" valign="top">
      <h1>#showWebsiteTitle()</h1>
      <p class="descrip">#showWebsiteDescription()</p>
      #showWeblogEntries("_day" 15)
   </td>
   <td class="sidebar" width="20%" valign="top">
      <h2>Calendar</h2>
      #showWeblogCalendar()<br>
      <h2>RSS Feeds</h2>
      #showRSSBadge()
      <h2>Links</h2>
      #showBookmarks("Blogroll" true false)
   </td>
</tr></tbody>
</table>
</body>
</html>
```

Without modifying Roller source

- Use Roller's web services interfaces
  - > Automate blogging via MetaWeblog or Atom protocol
  - > Automate admin via Roller Admin Protocol

- Add new Velocity page models
- Add new Weblog Entry plugins
- Plug-in your own comment authenticator
- Plug-in your own caching implementation

Agenda

- Roller history
- Roller features
- Roller community
- Roller internals: backend
- Roller internals: frontend
- Customizing Roller
- **Roller futures**

Futures: tagging

- UI for adding tags to entries
- Macros for displaying tag cloud on weblogs
- Macros for displaying site wide tag cloud
- Macros for searching via tags
- Tag-based dynamic feeds

Futures: protected weblogs

- Weblogs and weblog entries can be
  - > Shared with world
  - > Shared with all users on Roller server
  - > Shared only with members of weblog

- Comments can be allowed for
  - > Anonymous users
  - > Only users identified by OpenID
  - > Only registered users of Roller site
  - > Only members of same weblog

Futures: modular Roller

- Mix and match components
  - > Core Roller user management (required)
  - > Blog server module
  - > Planet server module
- Suite of other modules
  - > Wiki
  - > Forums
  - > Photo gallery

Conclusion

- Join the fun..

- Give Roller a try on your on server or JRoller.com
- Report bugs and RFEs to the Roller issue tracker
- Join the mailing lists and help out with development

- For more information:
  - > http://rollerweblogger.org

Thanks for listening!

Dave Johnson

davidm.johnson@sun.com