SiaFOO

The Name

- Needed a name
- Saw Discovery channel about Siafu ant.
  - Largest colony of any social insect
  - Eat everything (up to 100k animals in a single raid)
  - Ants build things and work together (like Siafoo)

What is it?

- Code / Knowledge sharing site
- A sort of a wiki, but not really
- Store
  - Useful scripts
  - Utility functions
  - Boilerplate code (like GUI startups)

Motivation

- Reduce wasted time looking for:
  - Boilerplate code
  - Random bugs
- Home for:
  - Useful scripts and small code (too small for SVN)
  - Prototypes
  - Misc programming information (like tips, and tricks)

What can it do?

- Post code **Snippets**
- Write **Articles**
- Organize stuff into **Libraries**
- Collaborate using **Groups**

Siafoo vs Blogs

- Anyone can edit code and articles
  - Information doesn't age as much
  - Bugs are fixed without the author's intervention
  - No need to look at comments
- Easy to locate and organize snippets/articles/etc
- License, original source and author

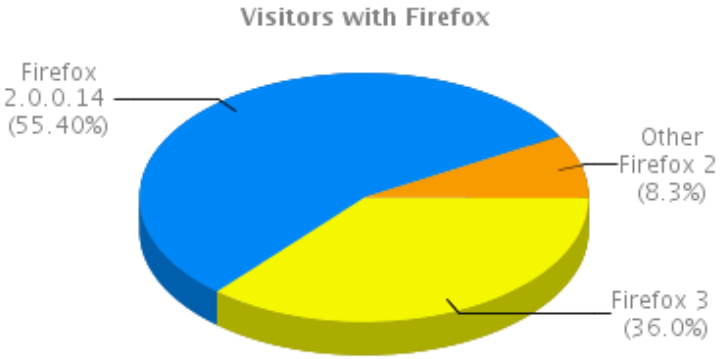
Features

- Code highlighting
- reStructured Text
- Access Controls
- All content is versioned
- Update stream (like Trac timeline)

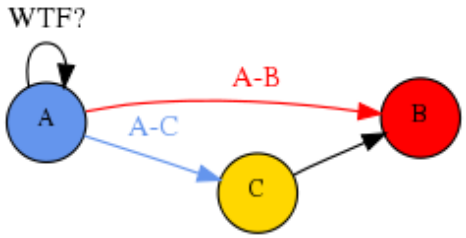
Features (cont.)

- Algorithms
- Sharing
  - Sending
  - Embedding
    - Dynamic (with JavaScript)
    - Static (Copy/Paste highlighted HTML
- Subscriptions
- XMLRPC

reStructured Text



$$\nabla \cdot E = \frac{1}{\epsilon_0}\rho$$
$$\nabla \cdot B = 0$$
$$\nabla \times E = -\frac{\partial B}{\partial t}$$
$$\nabla \times B = \mu_0 J + \mu_0 \epsilon_0 \frac{\partial E}{\partial t}$$




```
1  import java.io.*;
2  import javax.servlet.*;
3
4  public class Hello extends GenericServlet {
5      public void service(ServletRequest request, ServletResponse response)
6              throws ServletException, IOException {
7          response.setContentType("text/html");
8          final PrintWriter pw = response.getWriter();
9          pw.println("Hello, World!");
10         pw.close();
11     }
12 }
```

reST (cont)

- A text formatting “language”
- Looser / Cleaner syntax than Wikis
- Expandable
- Directives (.. foo::) for “complex” operations

Siafoo specific reST

- Graph::Easy (PyPerl)
  - Graphviz, Graph::Easy syntax
  - Database caching
- LaTeX
- Misc
  - Code blocks, Snippets, etc.
  - Charts
  - Amazon
  - YouTube

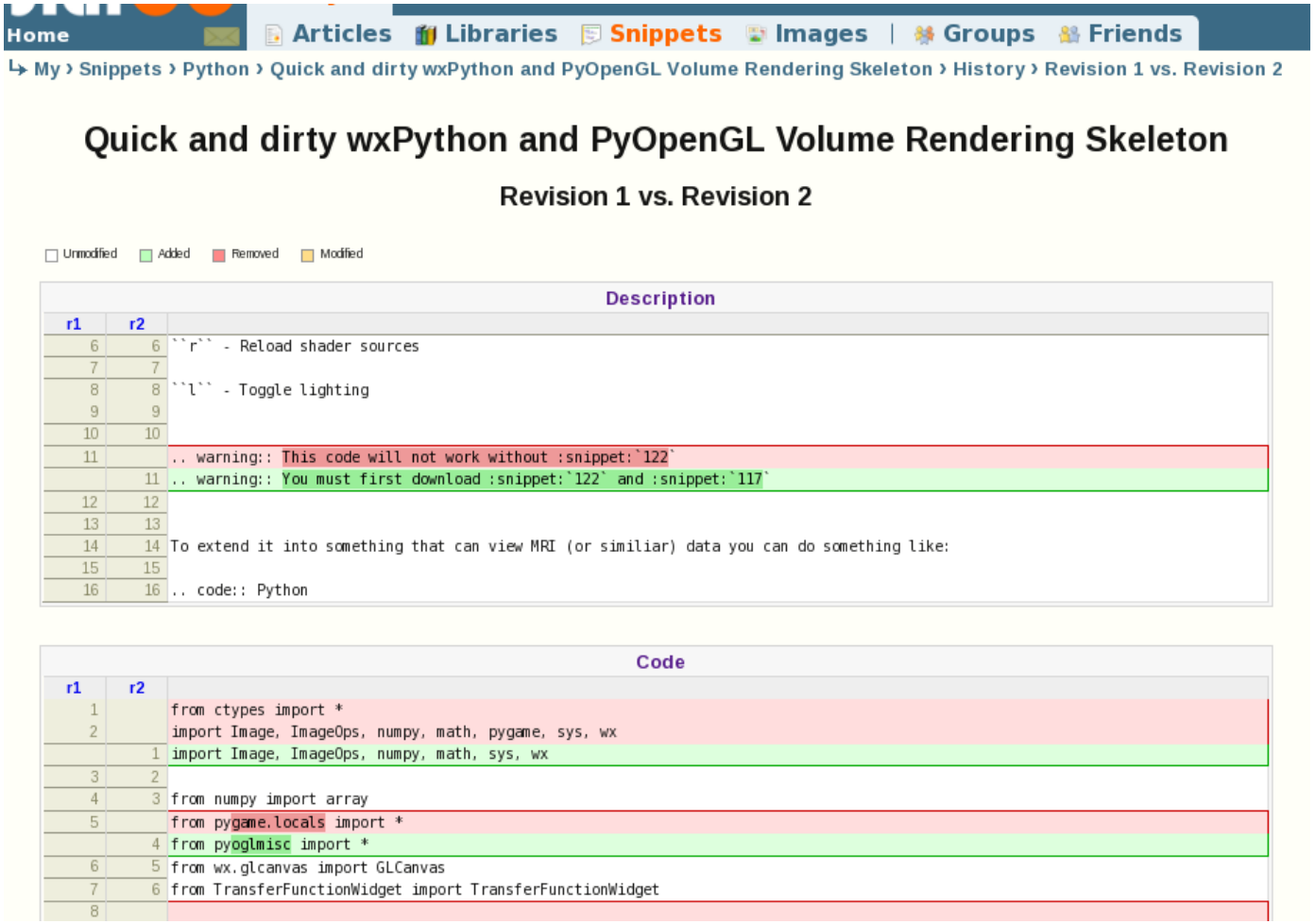
Code highlighting

- Pygments
- ~110 languages
- Easily extendible
  - New languages (like GLSL and COBOL)
  - Built-ins for new languages (NumPy, Blender-Python, Nodebox, etc.)

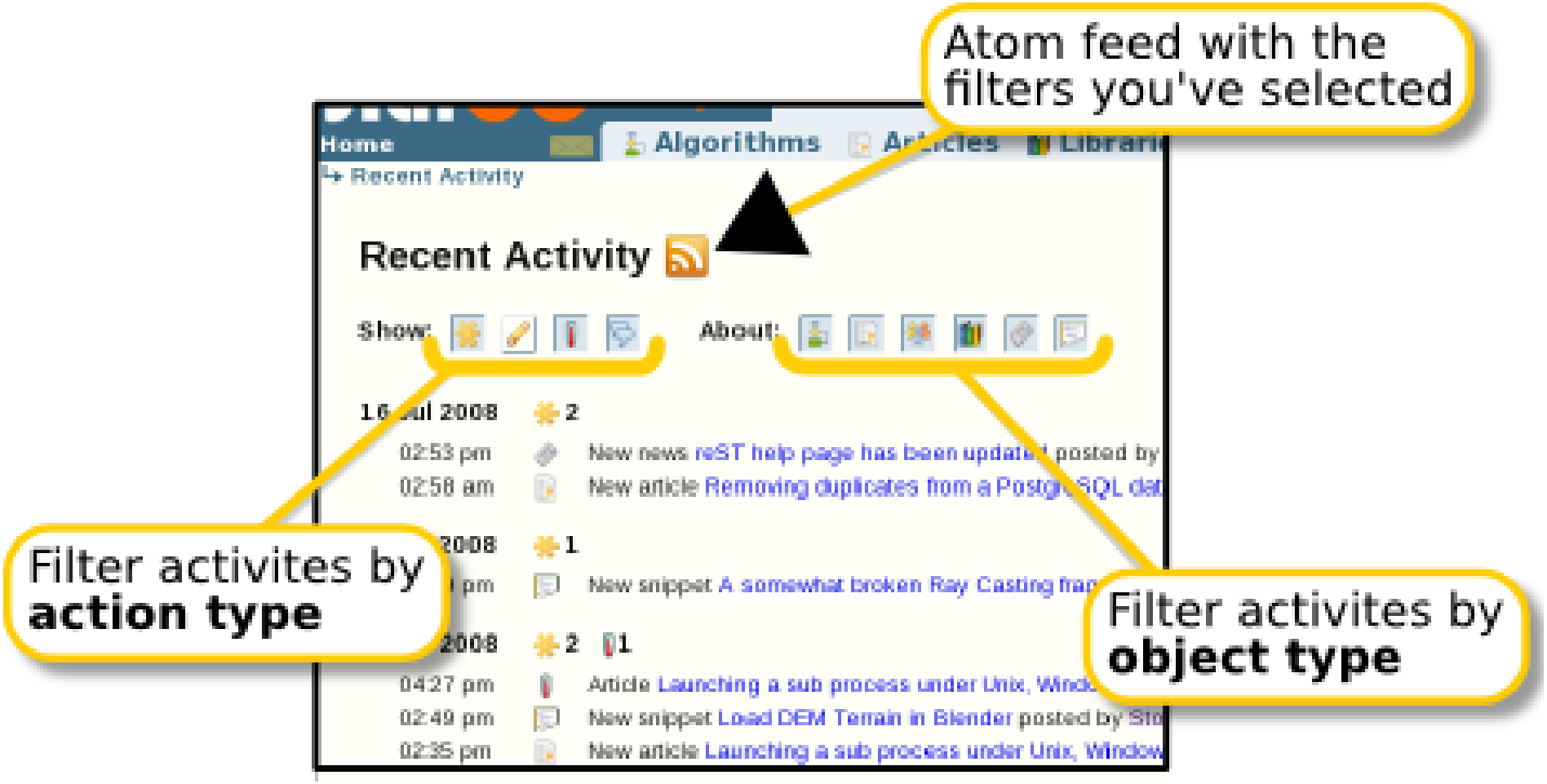
Access Controls

- View/Edit/Etc permissions
  - User / Group ownership
  - “Friends” permission
- Moving to ACLs soon

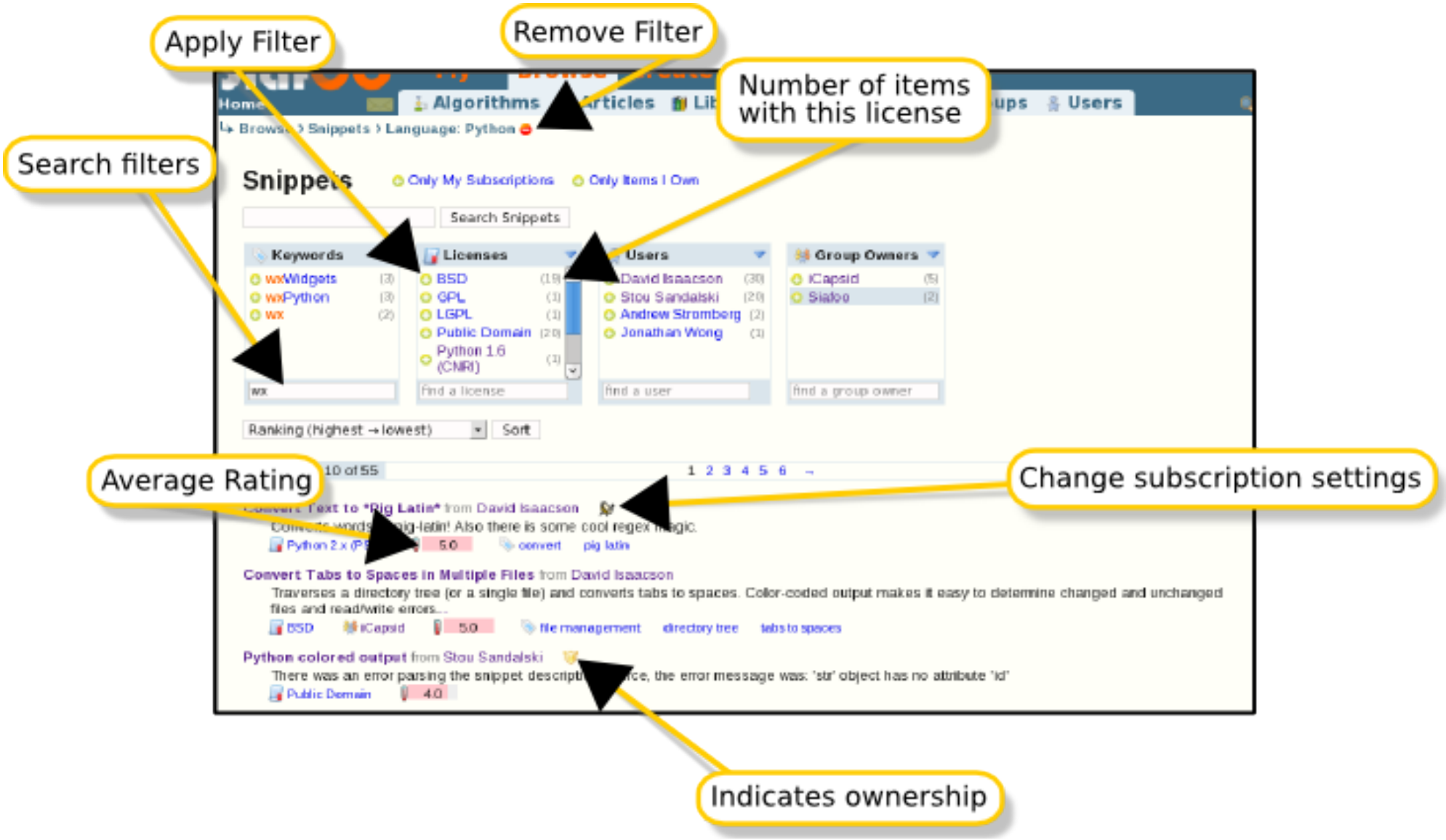
Versioning


Home
Articles
Libraries
Snippets
Images
Groups
Friends
My › Snippets › Python › Quick and dirty wxPython and PyOpenGL Volume Rendering Skeleton › History › Revision 1 vs. Revision 2
Quick and dirty wxPython and PyOpenGL Volume Rendering Skeleton
Revision 1 vs. Revision 2
Unmodified
Added
Removed
Modified
Description
r1 r2
6 6 ``r`` - Reload shader sources
7 7
8 8 ``l`` - Toggle lighting
9 9
10 10
11 .. warning:: This code will not work without :snippet:`122`
11 .. warning:: You must first download :snippet:`122` and :snippet:`117`
12 12
13 13
14 14 To extend it into something that can view MRI (or similiar) data you can do something like:
15 15
16 16 .. code:: Python
Code
r1 r2
1 from ctypes import *
2 import Image, ImageOps, numpy, math, pygame, sys, wx
1 import Image, ImageOps, numpy, math, sys, wx
3 2
4 3 from numpy import array
5 from pygame.locals import *
4 from pyoglmisc import *
6 5 from wx.glcanvas import GLCanvas
7 6 from TransferFunctionWidget import TransferFunctionWidget
8

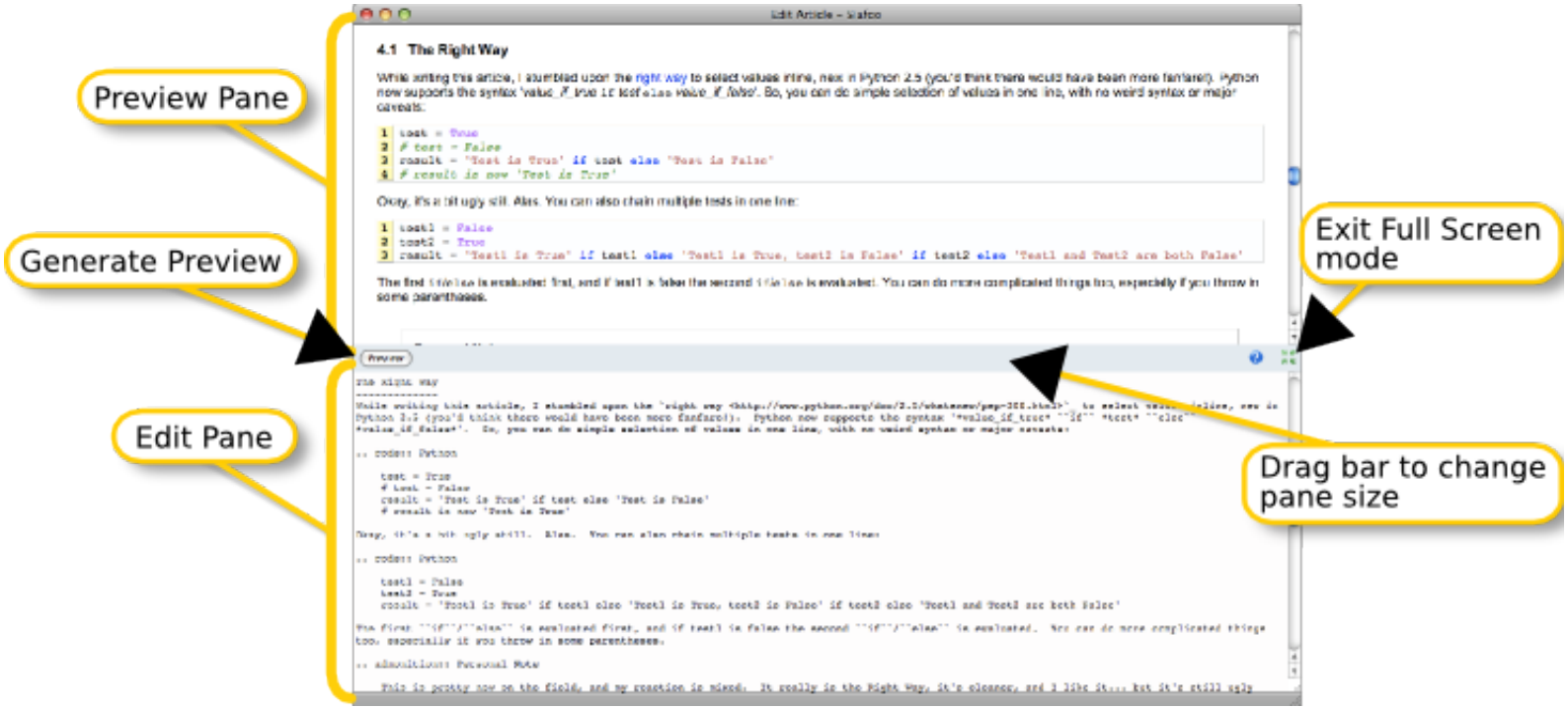
Update Stream



Filters


Apply Filter
Remove Filter
Number of items with this license
Search filters
Average Rating
Change subscription settings
Indicates ownership
Algorithms
Users
Snippets
Only My Subscriptions
Only Items I Own
Search Snippets
Keywords
Licenses
Users
Group Owners
wxWidgets
wxPython
wx
BSD
GPL
LGPL
Public Domain
Python 1.6 (CNRI)
David Isaacson
Stou Sandalski
Andrew Stromberg
Jonathan Wong
iCapsid
Sialoo
Ranking (highest → lowest)
Sort
Convert Tabs to Spaces in Multiple Files from David Isaacson
Traverses a directory tree (or a single file) and converts tabs to spaces. Color-coded output makes it easy to determine changed and unchanged files and read/write errors...
Python colored output from Stou Sandalski

Full Screen Editor



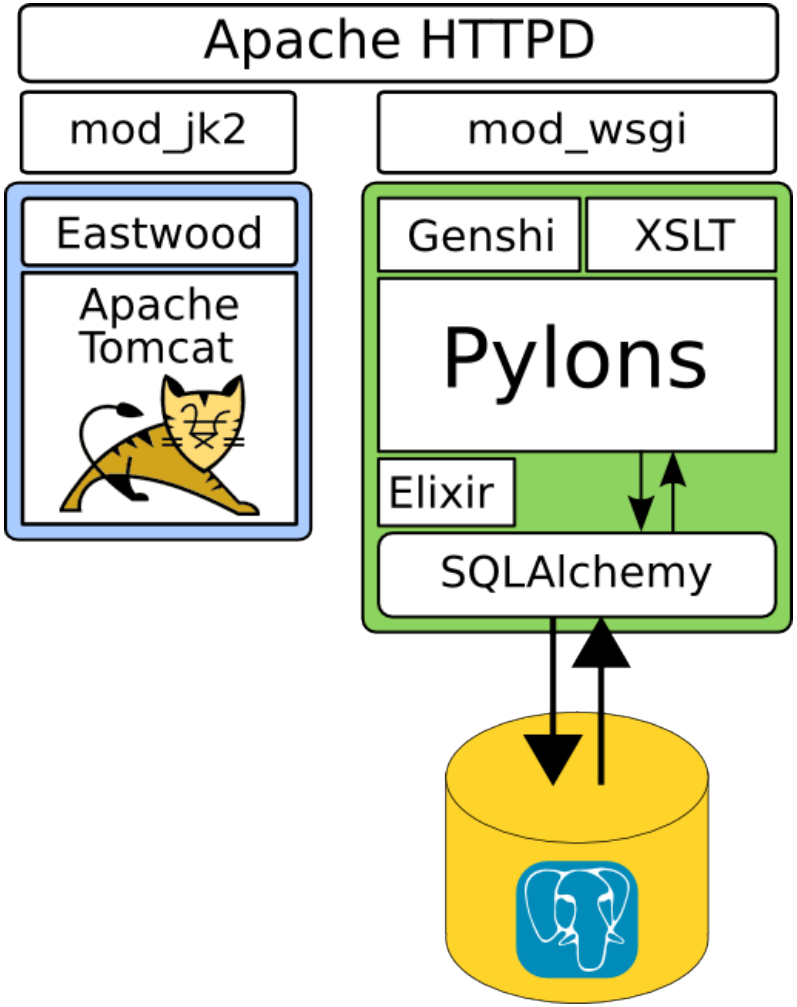
Social Networking?

- Siafoo is not a social networking site, but it has some of the same elements
  - Friends and Groups (mostly for access control)
  - Messages
- Useful without the social networking part
- The point is to “Do Stuff”

How it works...

Application Stack

- Fairly straight forward web app
- Apache httpd
- PostgreSQL
- Python (Perl and Java)
- Runs on SL (but it doesn't matter)



Scientific Linux

- We know SL from UCD work
- Cool:
  - Just RHEL / CentOS
  - Stable (1 year release cycle)
  - Good hardware vendor support
  - Xen Virtualization
- Not So Cool:
  - Older Python, PostgreSQL, Apache, etc.
  - Xen GUI is archaic

Why Python?

- A general purpose programming language
- Lots of libraries and features
- Not too slow
- We already knew Python...

Why PostgreSQL?

- “Real” database server (sorry)
- Data integrity comes first
- High performance / scalability
- XML Support
- Full Text Search (TSearch2)
- Single storage engine (InnoDB/MyISAM/... ?!)

SQLAlchemy

- Object Relational Mapper (ORM)
  - Object-based representation of database
  - Relation setup (one-one, many-one, one-many)
- SQL Expressions (SE)
  - Execute arbitrary SQL statements with automatic data coercion
  - Example: func.foo('bar', 1337, [1,2,3,4])
- (Tip: Freeze required version in setup.py)

Elixir

- Thin wrapper around SQLAlchemy's ORM
- Extensions: Versioning, Encryption, List plugin
- Cleaner code (than pure SA)

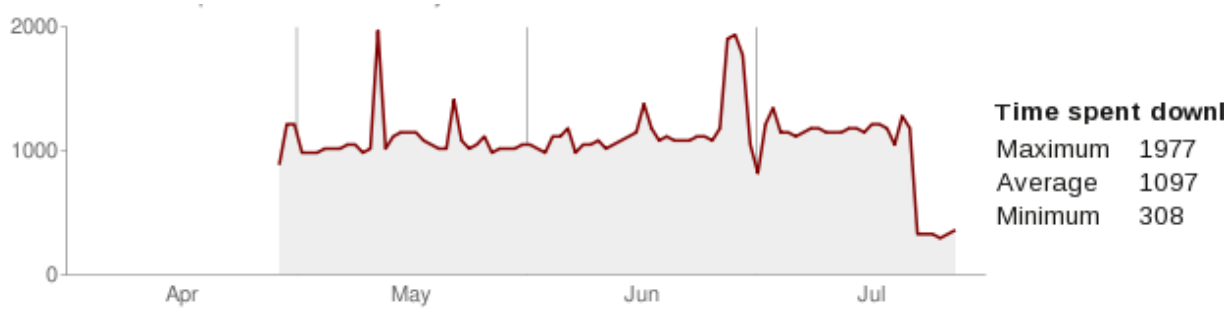
```
class Person(Entity):
    name = Field(String(128))
    addresses = OneToMany('Address')

class Address(Entity):
    email = Field(Unicode(128))
    owner = ManyToOne('Person')
```

Pylons

- MVC Framework
- Choice of template engines
- Out of the box SQLAlchemy or SQLObject
- Ruby On Rails style routing and AJAX
- Slightly harder to setup (than cherypy or php) but easier to manage in the long run
- Unit testing integration (nose)

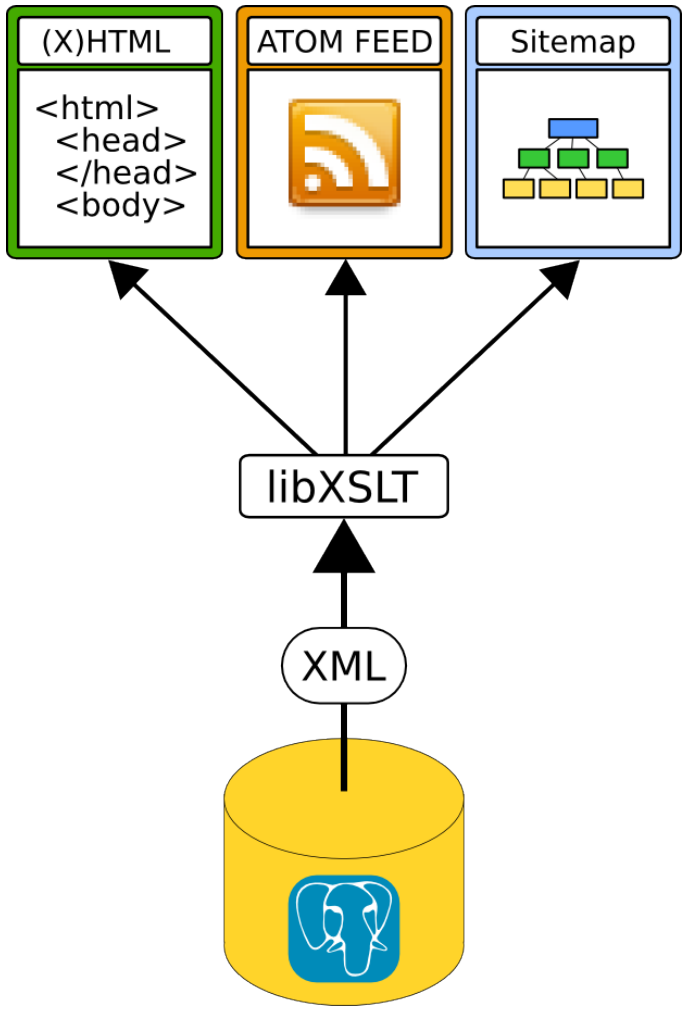
Genshi

- Parse, generate, and process HTML, XML
- Clean XML-like templates (feels like JSP)
- Stream filters and match templates
- Not as fast as text-based engines
- Static includes (since v0.5)



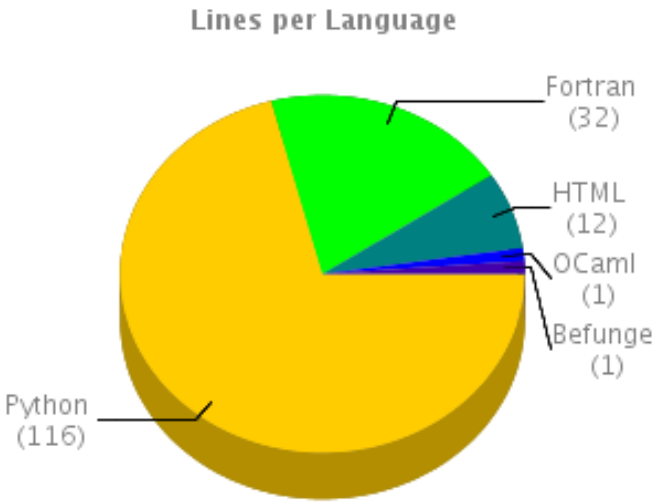
XSLT

- Good
  - Very very fast
  - Less code
  - Easy updates
- Bad
  - Hard to maintain for complex stuff
  - Stay away from LXML
  - libXML2/libXSLT are hard to use



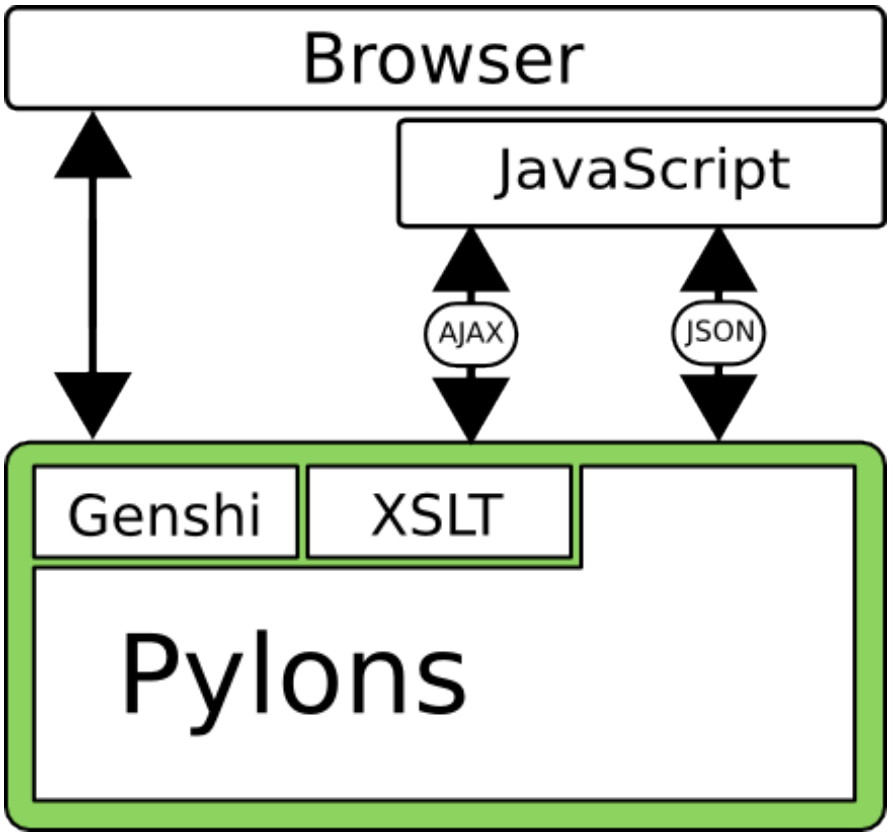
Eastwood

- Google Charts API
  - Encodes chart in request string
- jFreeChart
- Bad
  - Dataset size limits
  - Values are not exact
  - No caching (yet)



Front-End

- Mootools
- Asynchronous Requests / Responses
  - AJAX
  - JSON



Mootools

- Lightweight and modular
- Very well documented
- Clean API
- Compatible with Safari 2+, FF2+, Opera 9+, IE6+
- Fixes IE bugs

Future Plans

- Pastebin (private/public)
- Blogs (hosted and/or aggregated)
- Eclipse Plugin (other plugins)
- Todos / Tasks / Micro-bugs
- Code lint (other analysis and stats)

Feedback

- What wastes your time (other than meetings)?
- What can Siafoo do to save you time?
- How can it be more useful to groups?

More Information

- Siafoo: http://www.siafoo.net/help/about
- Pylons: http://www.pylonshq.com
- Elixir: http://elixir.ematia.de
- Eastwood: http://www.jfree.org/eastwood/