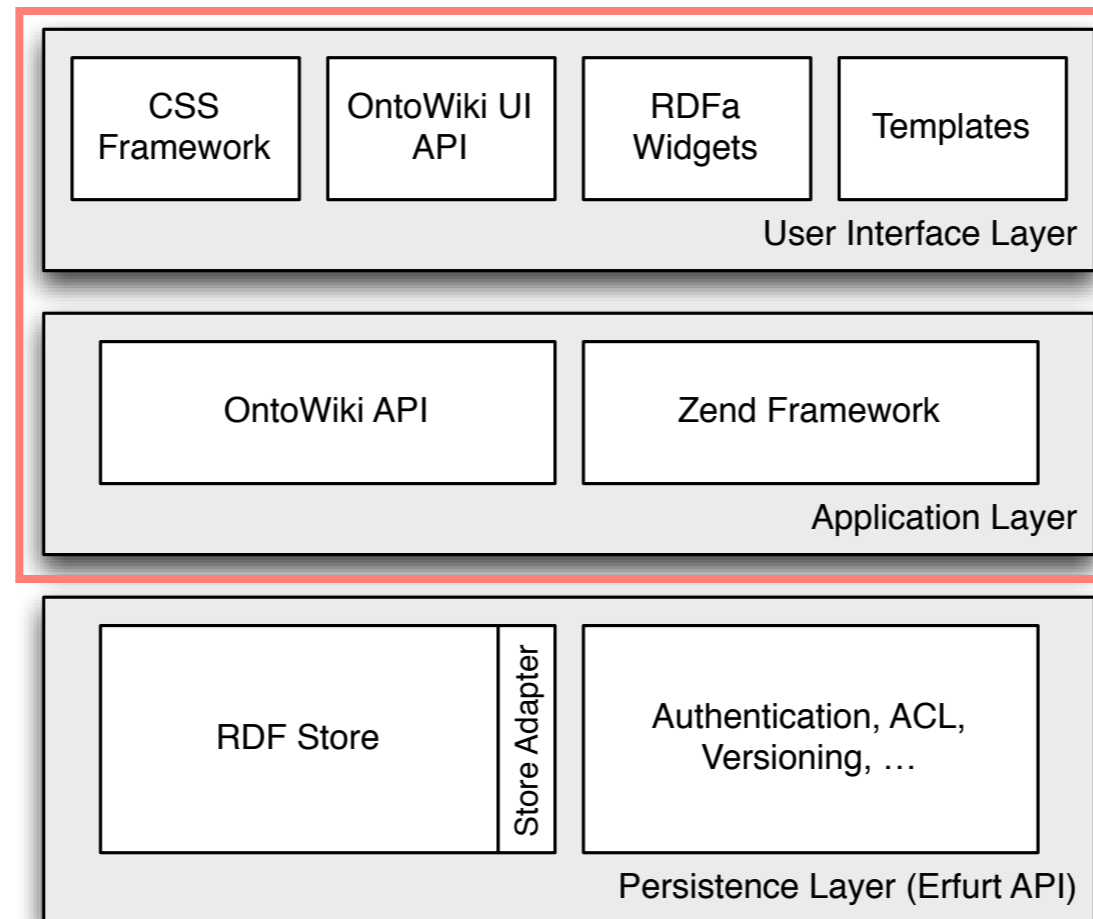


OntoWiki

Extension and Customization

Norman Heino

Architecture



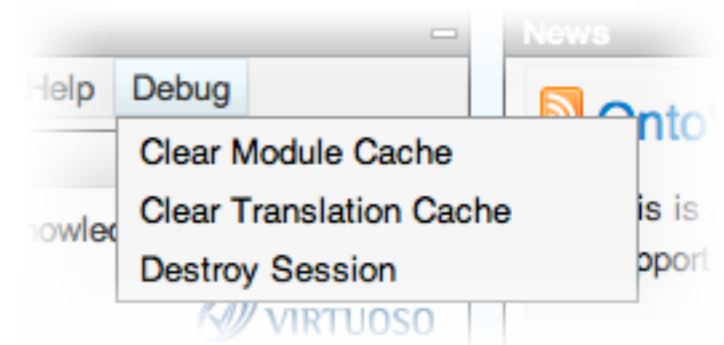
Zend Framework Intro

- PHP Framework for web applications
- Extensive documentation
- Outstanding code quality
- BSD-licensed
- Flexibly usable

OntoWiki API

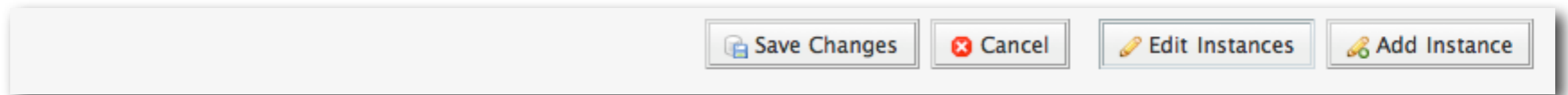
- Modify OntoWiki's UI data structures
- Menus, toolbar, navigation, model-base

Menus



- `OntoWiki_Menu::setEntry(...);`
- Entries can be Links, Separators or menu instances
- Window menus & context menus
- Serialize to JSON

Toolbar

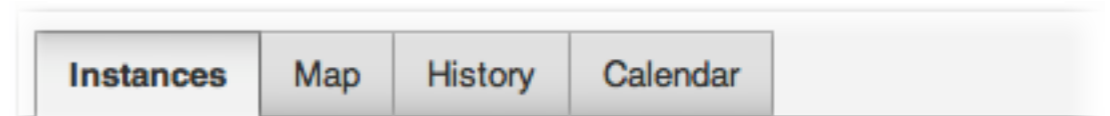


- OntoWiki_Toolbar
- Default Buttons:

Constant Name	CSS class	Function
CANCEL	Cancel	Cancel an operation
SAVE	Save	Save current changes
EDIT	Edit edit-enable	Enter editing mode
ADD	Add	Add a new entity
EDITADD		Add a new entity by editing another
DELETE	Delete	Delete the current selection
SUBMIT	Submit submit	Save changes
RESET	Reset reset	Reset changes

```
OntoWiki_Toolbar::appendButton(
    OntoWiki_Toolbar::SUBMIT,
    array(/* config */)
);
```

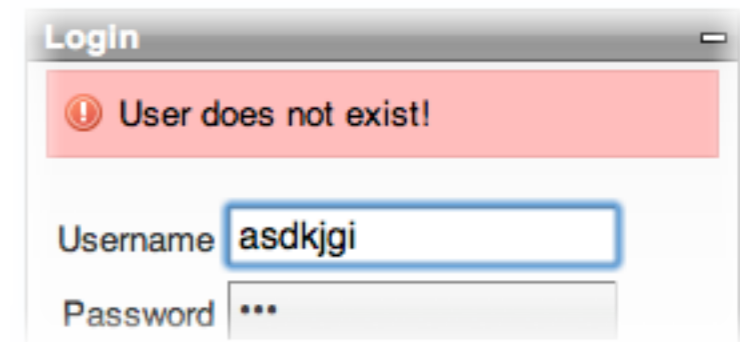
Navigation



- List of links
- Components can register with Navigation
- Can be disabled:

```
OntoWiki_Navigation::disableNavigation();
```

Messages



- Any window can have a message
- Application keeps message stack displayed automatically in main view
- Message types: success, warning, info, error

```
OntoWiki_Application::appendMessage(  
    'DBpedia successfully loaded.',  
    OntoWiki_Message::SUCCESS  
);
```

Customization Options

- Views/Templates
- Themes
- Localization

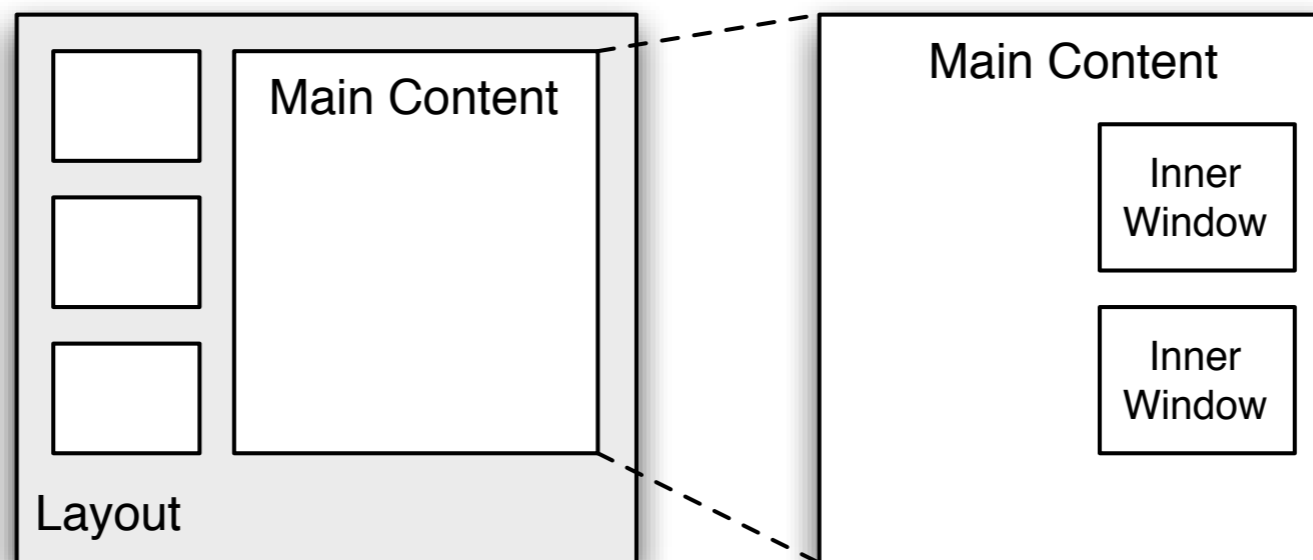
Localization

- Based on Zend_Translate
- CSV files
- Locale resolution
 - OntoWiki: folder-based
 - Extensions: filename-based

Views

- GUI part of controller
- Allow to modify what OntoWiki displays and how it does so
- Example: Maps view, ...

Templates



Templates (contd.)

- Folder: application/views/templates
- Components: <controller>/<action>.phtml
- Modules: subfolder within module folder
- Themes can have own templates folder that replaces defaults

Zend Template Tools

- Partials
- Placeholder
- View helper

Partials

- Render sub templates from within templates

```
$this->partial('template.phtml', array(/* params */));
```

```
$this->partialLoop('template.phtml', $data);
```

Placeholder

- Inter-template container

- Called by name

- Output: `$this->placeholder('foo');`

- Set content: `$this->placeholder('foo')->bar = $this->data;`

- Capture content:

```
$this->placeholder('foo')->captureStart();  
    // echo ...  
$this->placeholder('foo')->captureEnd();
```

View Helper

- Specialized placeholder
 - HeadTitle, HeadMeta, HeadScript, HeadStyle

- Append to title:

```
$this->headTitle()->append($subTitle);
```

- Prepend Script:

```
$this->headScript()->prependFile('jquery.js');
```

Themes

- CSS, JavaScript, images, templates
- Allow to modify way OntoWiki displays things
- Behaviour & look applied to CSS classes

CSS Framework

- Uses generic classes
 - Windows
 - Drop-down & context menus
 - Tabbed content
 - Message Boxes
 - Tables, lists

OntoWiki UI API

- Provides OntoWiki UI behaviour
- jQuery plug-ins
- Based on CSS classes

RDFa Widgets

- Editing widgets based on extracted statements
- Work on page model
- Can probably work on more than one statement

Use RDFa in your templates!

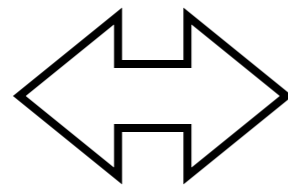
Break!



MVC Recap

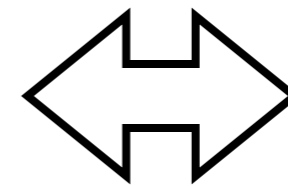
Model

- Business logic
- SPARQL is here!



Controller

- Component
- Controller's methods are *Actions*



View

- OntoWiki_View class
- Templates run in View's context

OntoWiki Models

- OntoWiki provides Model base class that:
 - Builds query fragments
 - fetches naming property from result set

OntoWiki URLs

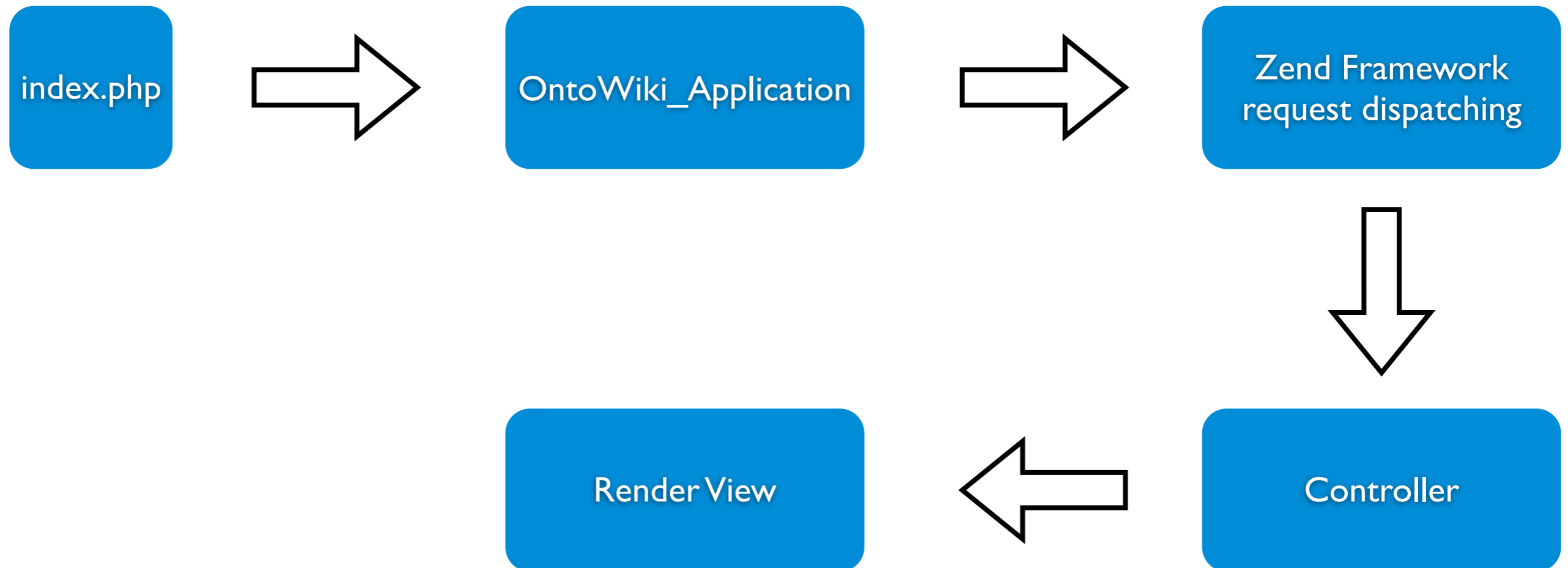
```
http://localhost/OntoWiki/list/r/foaf:Person/p/2
```

- Controller/Action name or Route name
- Parameters
 - *r*: `http://xmlns.com/foaf/0.1/Person`
 - *p*: `2`

URLs & URI Parameters

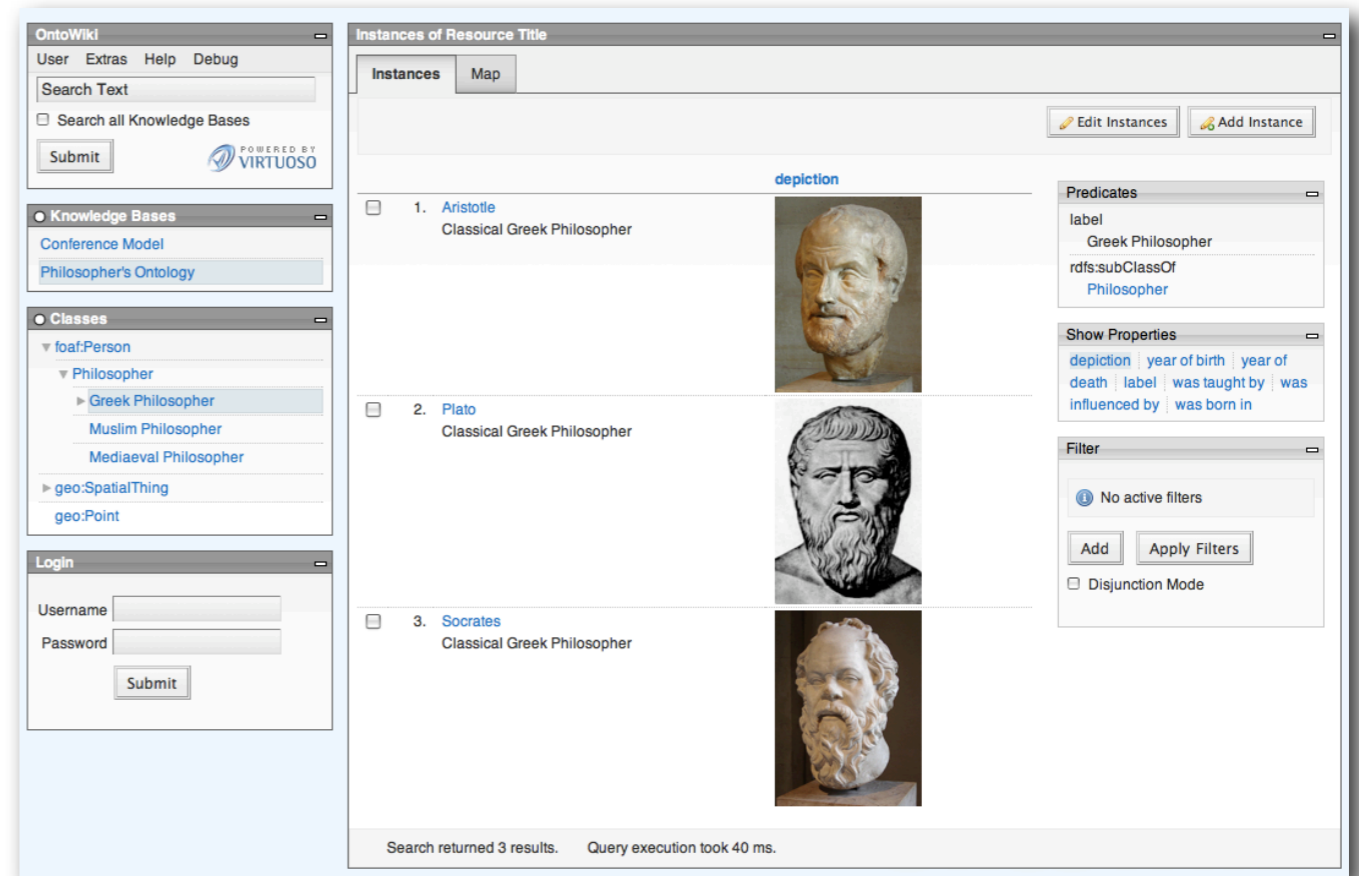
- URI parameter: r
- URI local parts or cURIs are automatically expanded
- URL construction by means of `OntoWiki_Url`
 - Can automatically compact URIs

Request Lifecycle



Extension Types

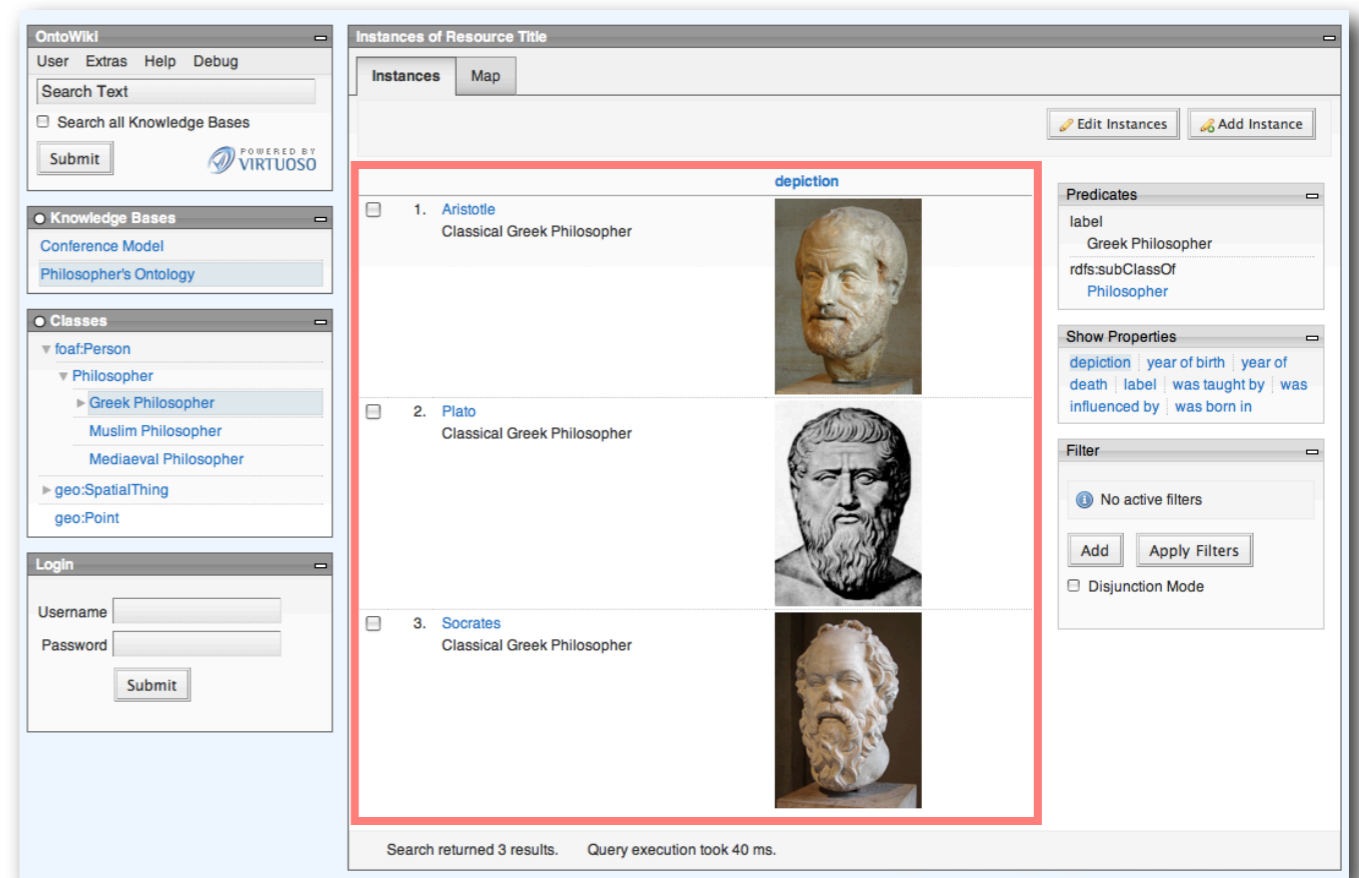
- Components
- Modules
- Plug-ins






The screenshot displays the OntoWiki interface. On the left, there is a navigation menu with sections for 'OntoWiki' (User, Extras, Help, Debug), 'Knowledge Bases' (Conference Model, Philosopher's Ontology), 'Classes' (foaf:Person, Philosopher, Greek Philosopher, Muslim Philosopher, Mediaeval Philosopher, geo:SpatialThing, geo:Point), and 'Login' (Username, Password, Submit). The main content area is titled 'Instances of Resource Title' and shows a list of three instances: 1. Aristotle (Classical Greek Philosopher), 2. Plato (Classical Greek Philosopher), and 3. Socrates (Classical Greek Philosopher). Each instance is accompanied by a small image of a bust. The interface also includes a search bar, a 'Submit' button, and a 'POWERED BY VIRTUOSO' logo. On the right side, there are panels for 'Predicates' (label: Greek Philosopher, rdfs:subClassOf: Philosopher), 'Show Properties' (depiction, year of birth, year of death, label, was taught by, was influenced by, was born in), and 'Filter' (No active filters, Add, Apply Filters, Disjunction Mode). At the bottom, it states 'Search returned 3 results. Query execution took 40 ms.'

Components

- MVC Controllers
- Often provide view
- Can serve other request



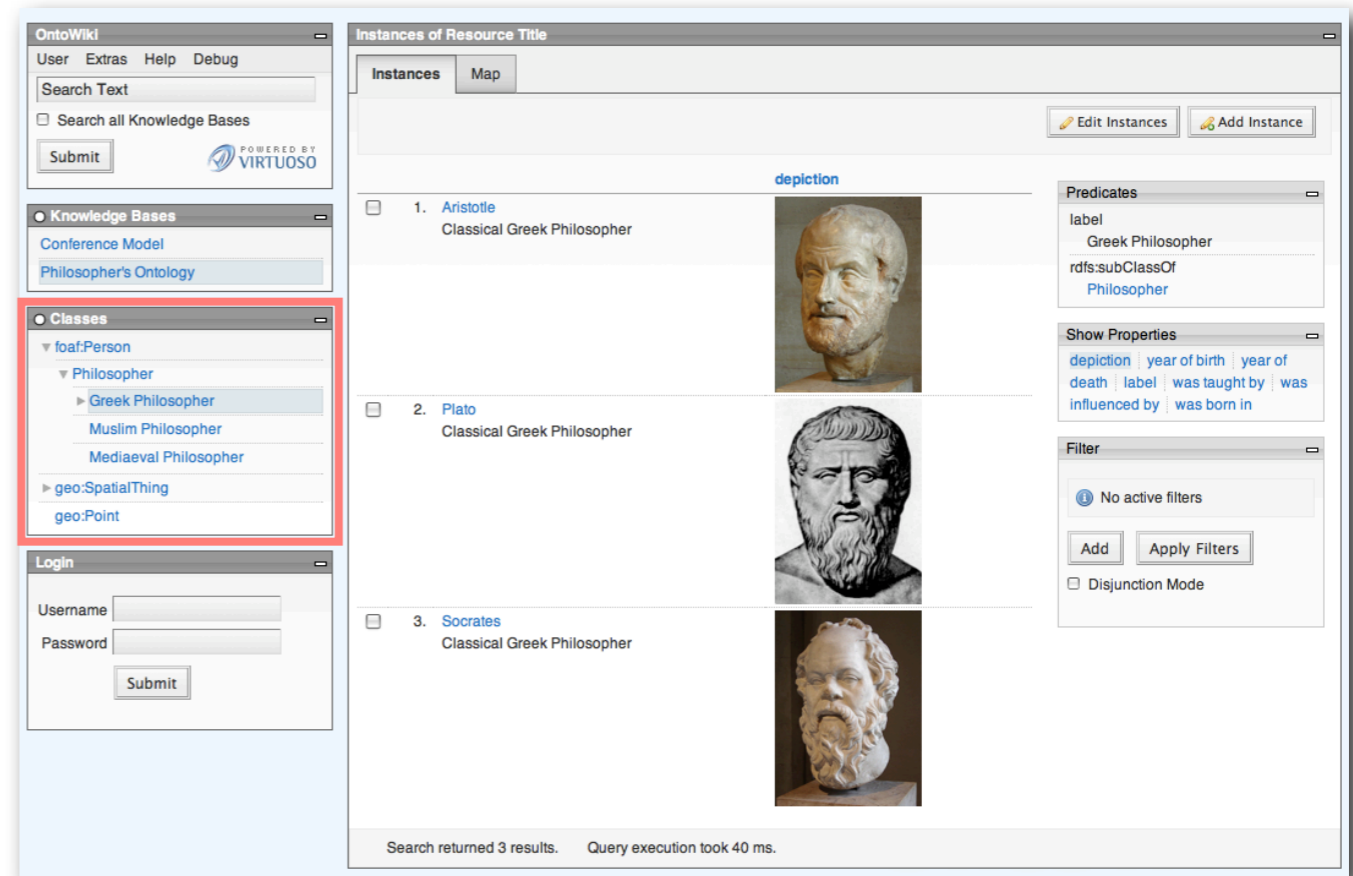
The screenshot displays the OntoWiki web interface. On the left, there is a navigation menu with sections for 'OntoWiki' (User, Extras, Help, Debug), 'Knowledge Bases' (Conference Model, Philosopher's Ontology), 'Classes' (foaf:Person, Philosopher, Greek Philosopher, Muslim Philosopher, Mediaeval Philosopher, geo:SpatialThing, geo:Point), and 'Login' (Username, Password, Submit). The main content area is titled 'Instances of Resource Title' and shows a list of three results for the property 'depiction':

Instances	Map
<input type="checkbox"/> 1. Aristotle Classical Greek Philosopher	
<input type="checkbox"/> 2. Plato Classical Greek Philosopher	
<input type="checkbox"/> 3. Socrates Classical Greek Philosopher	

On the right side of the interface, there are panels for 'Predicates' (label: Greek Philosopher, rdfs:subClassOf: Philosopher), 'Show Properties' (depiction, year of birth, year of death, label, was taught by, was influenced by, was born in), and 'Filter' (No active filters, Add, Apply Filters, Disjunction Mode). At the bottom, it states 'Search returned 3 results. Query execution took 40 ms.'

Modules

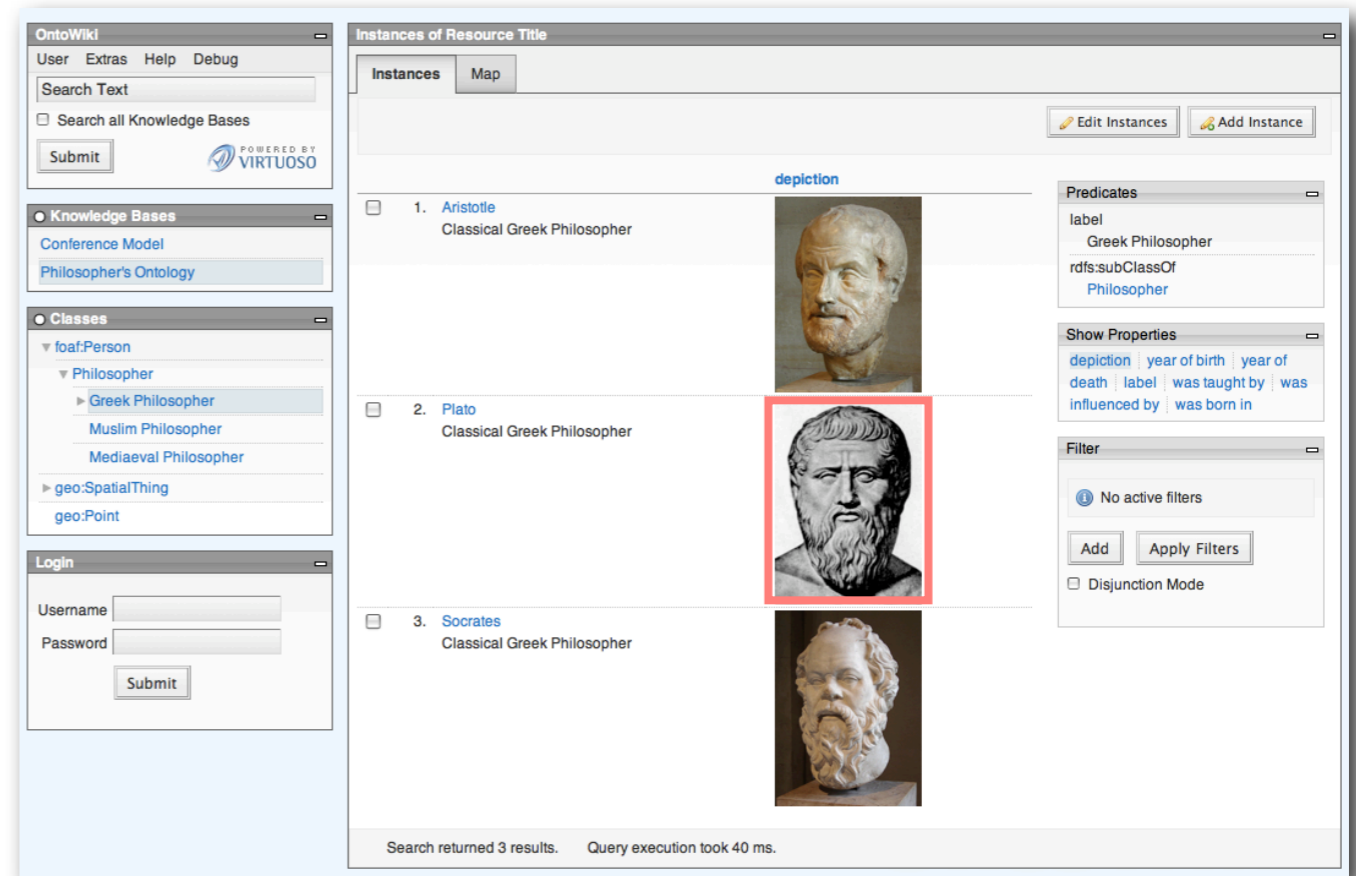
- Small windows
- Provide additional GUI Elements



The screenshot displays the OntoWiki interface. On the left, there are several small, stacked windows: 'OntoWiki' with a search bar and 'Submit' button; 'Knowledge Bases' with 'Conference Model' and 'Philosopher's Ontology'; 'Classes' with a tree view showing 'foaf:Person', 'Philosopher', 'Greek Philosopher', 'Muslim Philosopher', 'Mediaeval Philosopher', 'geo:SpatialThing', and 'geo:Point'; and 'Login' with 'Username' and 'Password' fields and a 'Submit' button. The main window, titled 'Instances of Resource Title', shows search results for 'Aristotle' (Classical Greek Philosopher), 'Plato' (Classical Greek Philosopher), and 'Socrates' (Classical Greek Philosopher). Each result includes a small image of a bust. The 'Aristotle' entry is selected, and its 'depiction' property is highlighted. On the right side of the main window, there are panels for 'Predicates' (showing 'label: Greek Philosopher' and 'rdfs:subClassOf: Philosopher'), 'Show Properties' (listing 'depiction', 'year of birth', 'year of death', 'label', 'was taught by', 'influenced by', 'was born in'), and 'Filter' (with 'No active filters' and 'Apply Filters' button). At the bottom, it states 'Search returned 3 results. Query execution took 40 ms.'

Plug-ins

- Arbitrary code
- Register for certain events



The screenshot displays the OntoWiki web interface. On the left, there is a navigation menu with sections for 'OntoWiki' (User, Extras, Help, Debug), 'Knowledge Bases' (Conference Model, Philosopher's Ontology), 'Classes' (foaf:Person, Philosopher, Greek Philosopher, Muslim Philosopher, Mediaeval Philosopher, geo:SpatialThing, geo:Point), and 'Login' (Username, Password, Submit). The main content area is titled 'Instances of Resource Title' and shows a table of instances for the property 'depiction'. The table lists three results: 1. Aristotle (Classical Greek Philosopher) with a bust image, 2. Plato (Classical Greek Philosopher) with a bust image highlighted by a red border, and 3. Socrates (Classical Greek Philosopher) with a bust image. On the right side, there are panels for 'Predicates' (label: Greek Philosopher, rdfs:subClassOf: Philosopher), 'Show Properties' (depiction, year of birth, year of death, label, was taught by, was influenced by, was born in), and 'Filter' (No active filters, Add, Apply Filters, Disjunction Mode). At the bottom, it states 'Search returned 3 results. Query execution took 40 ms.'