



# Embedding workflows in your website

Aleksandra Pawlik  
myGrid Team  
University of Manchester

VLIZ, 2014-10-06 / 2014-10-08  
<http://www.taverna.org.uk/>



This work is licensed under a  
[Creative Commons Attribution 3.0 Unported License](http://creativecommons.org/licenses/by/3.0/)





# Seen it already

- You have already seen an example of a website that runs Taverna workflows
  - the [BioVeL portal](#)
- How was it done?

## Taverna Player

- Looks after the running of the workflows
- Surfaces them for a website



# Lightweight embedding

- Embed like a YouTube video
- Embedded workflow is passed the URI of data

```
<iframe src="http://portal.org/runs/new?  
        embedded=true&  
        workflow_id=1&  
        input_uri=http://scratchpad.org/taxa/1234/data"  
>  
</iframe>
```

- This level of integration is lightweight
  - Science showcases
  - One off analyses

<https://github.com/myGrid/taverna-player/wiki/Embedding>



# Tightness of integration

## Lightweight embedding

- Run a specified workflow
  - Chosen by the host website's administrator
- Results are not stored in the host
- Workflow run retains host app look and feel

## Tight integration

- Run any workflow
  - That the host website is authorized to see
- Results are available for further analysis
- Workflow appears as part of the host website

## Common

- Workflows are run within Taverna Player in the host app
- Interactions are presented to the user
- Results can be downloaded



# Taverna Player

- A Ruby on Rails plugin library
  - Hooks into host application's
    - Workflow model
    - Authentication and authorization system
  - Provides a REST interface
- Talks to Taverna Server's REST interface
  - Uploads the workflow, sets inputs
  - Presents workflow interactions to the user
  - Retrieves results, logs and provenance data



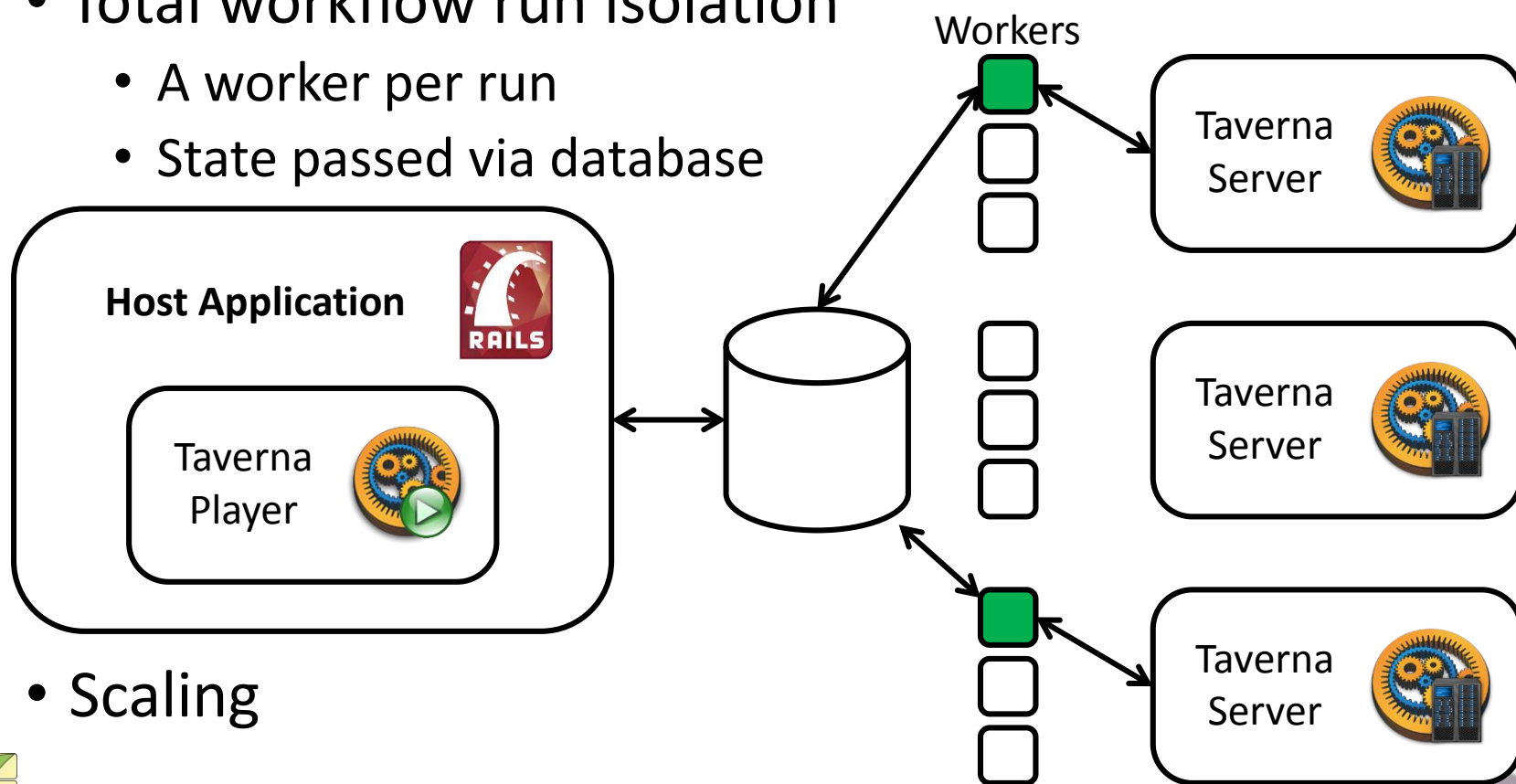
# Taverna Player

- Surfaces a workflow run in three ways:
  - As a Web interface in the browser
    - In the host application
  - As an embeddable widget
    - In any Web page (c.f. YouTube videos)
  - As a REST-based Web Service
- All look-and-feel and styling is derived from the host application
  - Rails's hierarchical layouts and views



# Taverna Player

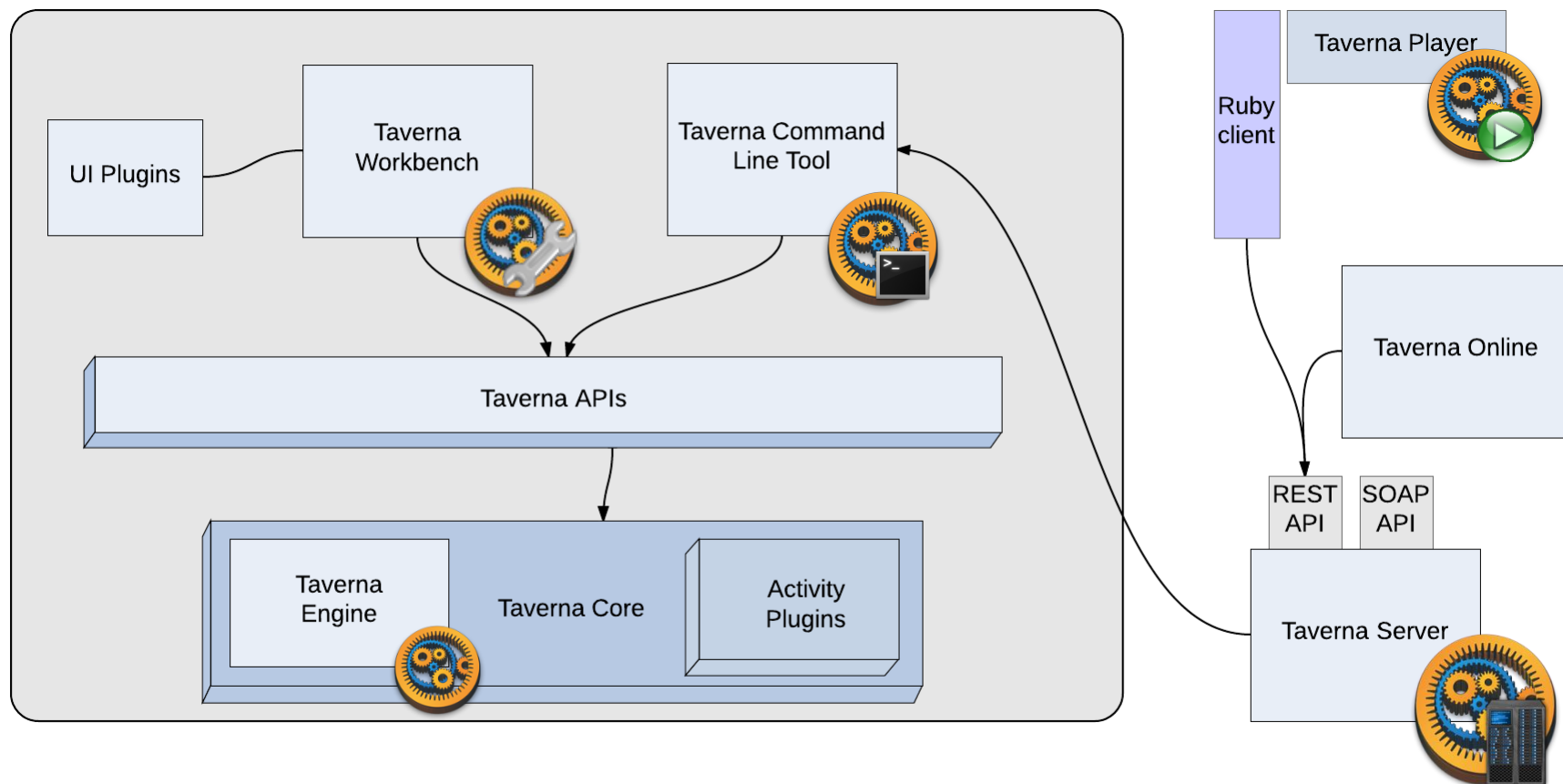
- Total workflow run isolation
  - A worker per run
  - State passed via database



- Scaling



# Taverna all together

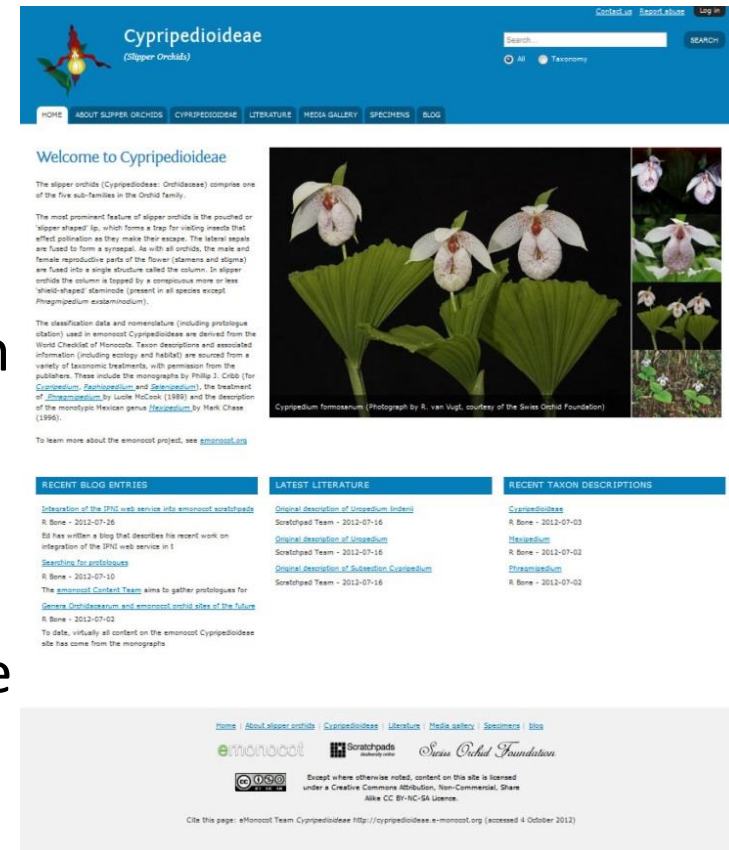






# Workflows in Scratchpads

- Virtual Research Environments
- Hosted websites for biodiversity data
- Virtual research & publication platform
- Curated data and analysis
- Completely open access & open source
- Modular & flexible



Cypripedioideae  
 (Slipper Orchids)

Welcome to Cypripedioideae  
 The slipper orchids (Cypripedioideae: Orchidaceae) comprise one of the five sub-families in the Orchid family.



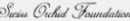
The most prominent feature of slipper orchids is the pouch or slipper-shaped lip, which forms a trap for visiting insects that effect pollination as they make their escape. The lateral sepals are fused to form a synsepal. As with all orchids, the male and female reproductive parts of the flower (stamens and stigma) are fused into a single structure called the column. In slipper orchids the column is topped by a conspicuous more or less wing-shaped idiomorph (present in all species except *Phragmipedium exstaminatum*).

The classification data and nomenclature (including protologue citation) used in eMonocot Cypripedioideae are derived from the World Checklist of Honorata. Taxon descriptions and associated information (including ecology and habitat) are sourced from a variety of taxonomic treatments, with permission from the publishers. These include the monographs by Philip J. Cribb (for *Cypripedium*, *Phragmipedium*, and *Selenicium*), the treatment of *Phragmipedium* by Lucile McCook (1989) and the description of the monotypic Mexican genus *Selenicium* by Mark Chase (1996).

To learn more about the eMonocot project, see [eMonocot.org](http://eMonocot.org)

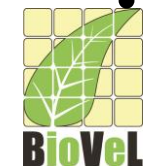
RECENT BLOG ENTRIES	LATEST LITERATURE	RECENT TAXON DESCRIPTIONS
<a href="#">Integration of the IPNI web service into eMonocot scratchpads</a> R. Bone - 2012-07-26 Ed has written a blog that describes his recent work on integration of the IPNI web service in 1	<a href="#">Original description of <i>Selenicium</i> <i>indense</i></a> Scratchpad Team - 2012-07-16 <a href="#">Original description of <i>Selenicium</i></a> Scratchpad Team - 2012-07-16 <a href="#">Original description of <i>Selenicium</i> <i>curvicaule</i></a> Scratchpad Team - 2012-07-16	<a href="#">Cypripedioideae</a> R. Bone - 2012-07-03 <a href="#">Mexicanum</a> R. Bone - 2012-07-02 <a href="#">Phragmipedium</a> R. Bone - 2012-07-02

[Home](#) | [About slipper orchids](#) | [Cypripedioideae](#) | [Literature](#) | [Media gallery](#) | [Specimens](#) | [Blog](#)

Except where otherwise noted, content on this site is licensed under a Creative Commons Attribution-NonCommercial-Share Alike CC BY-NC-SA license.

Cite this page: eMonocot Team. Cypripedioideae <http://cypripedioideae.e-monocot.org> (accessed 4 October 2012)





# Workflows in Scratchpads - cont

Antk ID guide | [Contact us](#) | [Report abuse](#) | [Cite me](#) | [Language](#) | [Log in](#)

HOME ABOUT TAXA LUCID KEY CHECK

Antkey ID guide | introduced ants

Search... SEARCH

All Taxonomy

HOME ABOUT TAXA LUCID KEY CHECKLISTS MEDIA VIDEOS GLOSSARY LITERATURE SPECIMENS FORUMS BLOG

Home » [Pheidole megacephala](#) » [Myrmicinae](#) » [Pheidole](#) » Pheidole megacephala

Run status: Waiting for user input

Bio STIF Biodiversity Spatial Temporal Interactive interface Version 0.2.0 Help

Continue the Workflow Abort the Workflow

Selection layer Map selector tools Aggregation type Hide coordinates Longitude: -34.97632, Latitude: 7.26301

1194 results with location information

0 results with time information

2012-01-01 2012-04-01 2012-07-01 2012-10-01 2013-01-01 2013-04-01 2013-07-01 2013-10-01

species\_point

1-100 of 1194 Results Page 1 of 12

	decimalLatitude	nameComplete	occurrenceID
<input type="checkbox"/>	-24.95694	Pheidole megacephala	CASC-BLF05316-CASENT0019150
<input type="checkbox"/>	-13.08333	Pheidole megacephala	CASC-BLF09426-CASENT0041336
<input type="checkbox"/>	-16.01	Pheidole megacephala	CASC-BLF06812-CASENT0025506



Cancel



# IPython Notebook

- IPython Notebook
  - originally developed by Fernando Perez of University of Berkeley
  - browser-based environment for interactive computing
  - <http://ipython.org/>
- write, edit and re-run Python scripts
- interactive data visualization
- report presentation
- save, record, share notebook runs



# Taverna Player Client

- Uses Taverna Player and its Server to run workflows within an IPython Notebook
- Data passed from the Notebook to the executing Taverna Workflows
- Workflow run's requests for data answered within the Notebook using Taverna's interaction service
- Results retrieved from the run and fed back into the Notebook
- Available from the PyPi registry – <http://pypi.python.org/pypi/tavernaPlayerClient>
- Tested with BioVeL workflows for data refinement and ecological niche modelling.



# Taverna in IPython example

## Client Creation

```

In [3]: from tavernaplayerclient import *
In [4]: c = Client('http://dev.at.biovel.eu', 'player', 'player')
In [5]: workflows = c.workflows
In [6]: for w in c.workflows: print w.identifier, w.title
  
```

```

1 Various output types
2 Ecological niche modelling
3 Select Model For Me with co
4 Define with PartitionFinder
5 Partitioning environmental
6 Partitioning environmental
10 Partitioning environmental
12 Partitioning environmental
13 Bioclim workflow with inte
15 A workflow
19 BioVel ESW DIFF - ESM Stat
  
```

```
In [20]: esw = c.get_workflow(20)
```

```
In [21]: esw_run_template = esw.run_template
```

```
In [22]: print esw.description
```

```
In [ ]: for k in esw_run_template.inputs.keys(): print k
```

```
In [23]: namelist = "now,2050"
```

```
In [24]: layerlist = projection_urls[0] + "," + projection_urls[1]
```

```
In [25]: esw.run("Example esw run",
                {"namelist": namelist,
                 "layerlist": layerlist,
                 "png_size": "800"})
```

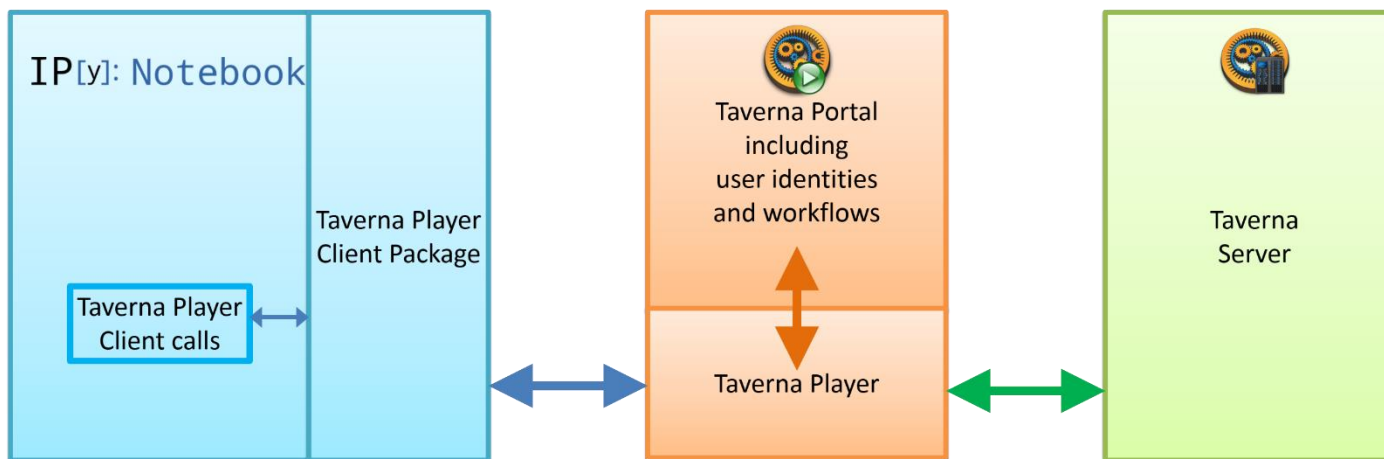
## Interaction with Run



## Organization of Data



# Taverna and IPython architecture



Flows			
Authorize Client	Notebook -> Player -> Portal Portal -> Player -> Notebook	Start run	Notebook -> Player -> Server
List workflows	Notebook -> Player -> Portal Portal -> Player -> Notebook	Interact with run	Server -> Player -> Notebook Notebook -> Player -> Server
Get workflow and run template	Notebook -> Player -> Portal Portal -> Player -> Notebook	Return results	Server -> Player -> Notebook



# Summary

- Taverna Player is very flexible
- Examples of integration into
  - Ruby on Rails – BioVeL Portal
  - Drupal – Scratchpads
  - Python – IPython Notebook
- Different levels of integration
  - Simple iframe to
  - Communication via REST API
- More information at <https://github.com/myGrid/taverna-player/wiki>