Dynamic visualisation in the IPython Notebook

www.cawcr.gov.au

Brianna Laugher
Hobart, Tasmania
July 2013, PyCon AU

Australian Government
Bureau of Meteorology

The Centre for Australian Weather and Climate Research
A partnership between CSIRO and the Bureau of Meteorology

CSIRO

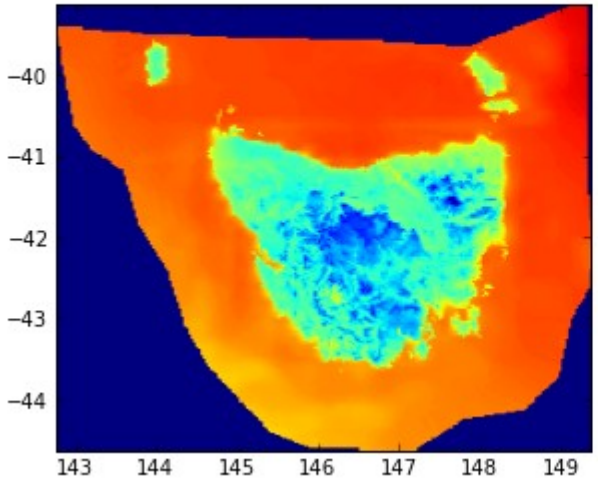
Slides & code

github.com/pfctdayelise/dapbook

What we have...

```
from pydap.client import open_url
dataset = open_url("http://127.0.0.1:8001/IDT71003_TAS_MinT_SFC.nc")

mint = dataset['MinT_SFC']
lats = dataset['latitude']
lons = dataset['longitude']

# Just pull out data for the first day
day0 = mint[0]
data = np.squeeze(day0)
```

```
extent = [min(lons), max(lons), min(lats), max(lats)]
_ = plt.imshow(data, origin='lower', vmin=0, vmax=20, extent=extent)
```

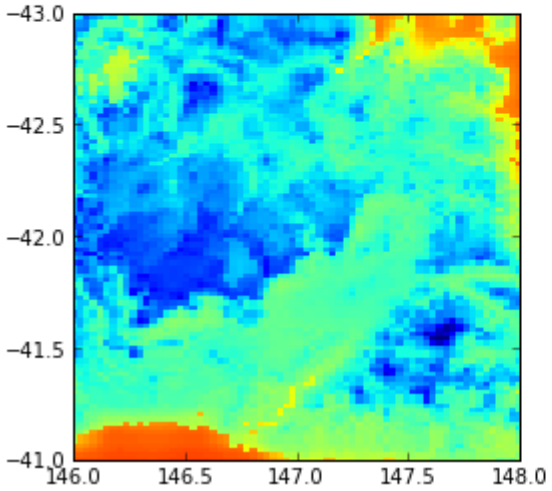


Australian Government
Bureau of Meteorology

The Centre for Australian

What we have...

```
# Plot around Hobart.
# Setting interpolation to nearest means NO interpolation
# ie. we see the data as blocky because it actually is.
_ = plt.imshow(data, interpolation='nearest',
               origin='lower', vmin=0, vmax=20, extent=extent)
_ = plt.axis([146, 148, -41, -43])
```



Australian Government
Bureau of Meteorology

The Centre for Australian

The problem

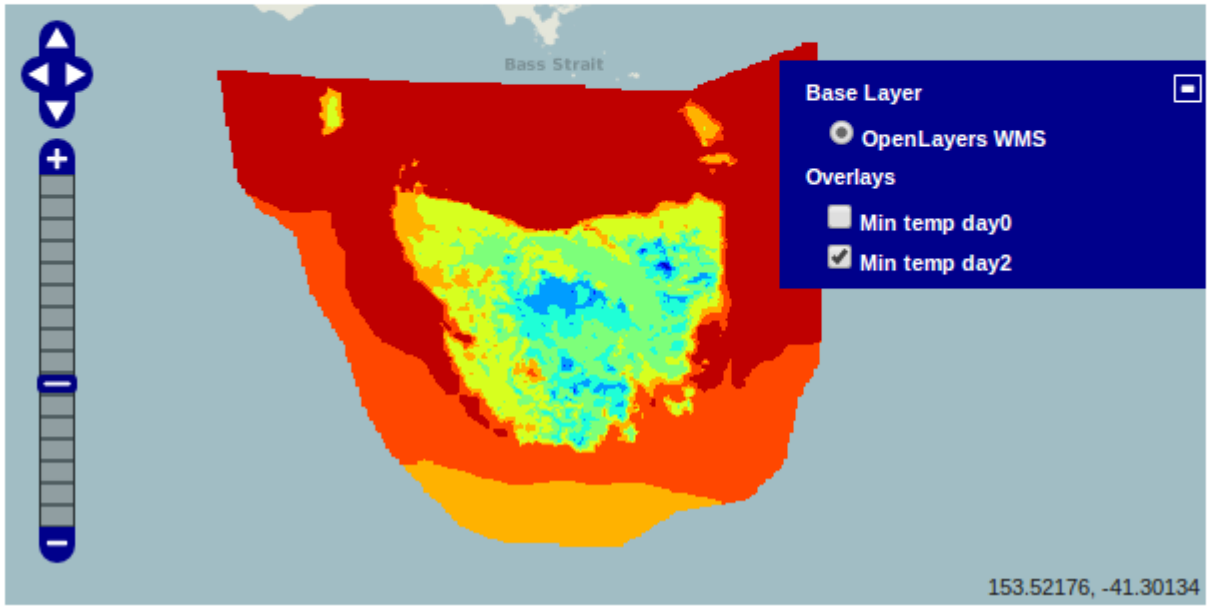
Can't zoom
Can't pan
Can't layer

IPython Notebook makes it easy to explore datasets and plot them with matplotlib...

...but for gridded datasets, it really would be nice to have them on a map!

...what we want

```
import dapbook
b = dapbook.PydapWMS(server='http://127.0.0.1:8001/IDT71003_TAS_MinT_SFC.nc.wms',
                     layers=[('MinT_SFC', 'Min temp day0', 0),
                             ('MinT_SFC', 'Min temp day2', 2),
                     ], centerlat=-42, centerlon=147, initialzoom=6,
                     mapwidth=600, mapheight=300)
b.html
```



Australian Government
Bureau of Meteorology

The Centre for Australian

What is this data?

NetCDF file – common format for gridded data (could easily have been HDF5)

Commonly used by earth science communities

Sample TAS MinT grid from the Australian Digital Forecast Database produced by BoM, see http://www.bom.gov.au/catalogue/data-feeds.shtml for details and to download sample grids

ADFD grids feed into MetEye - http://www.bom.gov.au/australia/meteye/

MetEye

LATEST WEATHER

Current Temp, Rain, Wind ...

FORECASTS

Rainfall Forecasts

Temperature Forecasts

3-hourly

Air temperature

Feels like

Daily

Daily maximum

Overnight minimum

Updated 5 Jul 2013, 7:20 AM AEST

Clear Info

Overlay

Latest rain radar

Light Mod Heavy Offline

Updated 5 Jul 2013, 4:00 PM AEST

Storms, Snow, Fog, Frost ...

Humidity Forecasts

Wind Forecasts

Waves Forecasts

Thursday Friday Saturday Sunday Monday Tuesday Wednesday

loading...

5 Jul 2013

Darwin

Perth

Adelaide

Brisbane

Sydney

Canberra

Melbourne

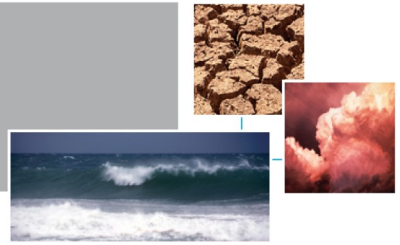
Hobart

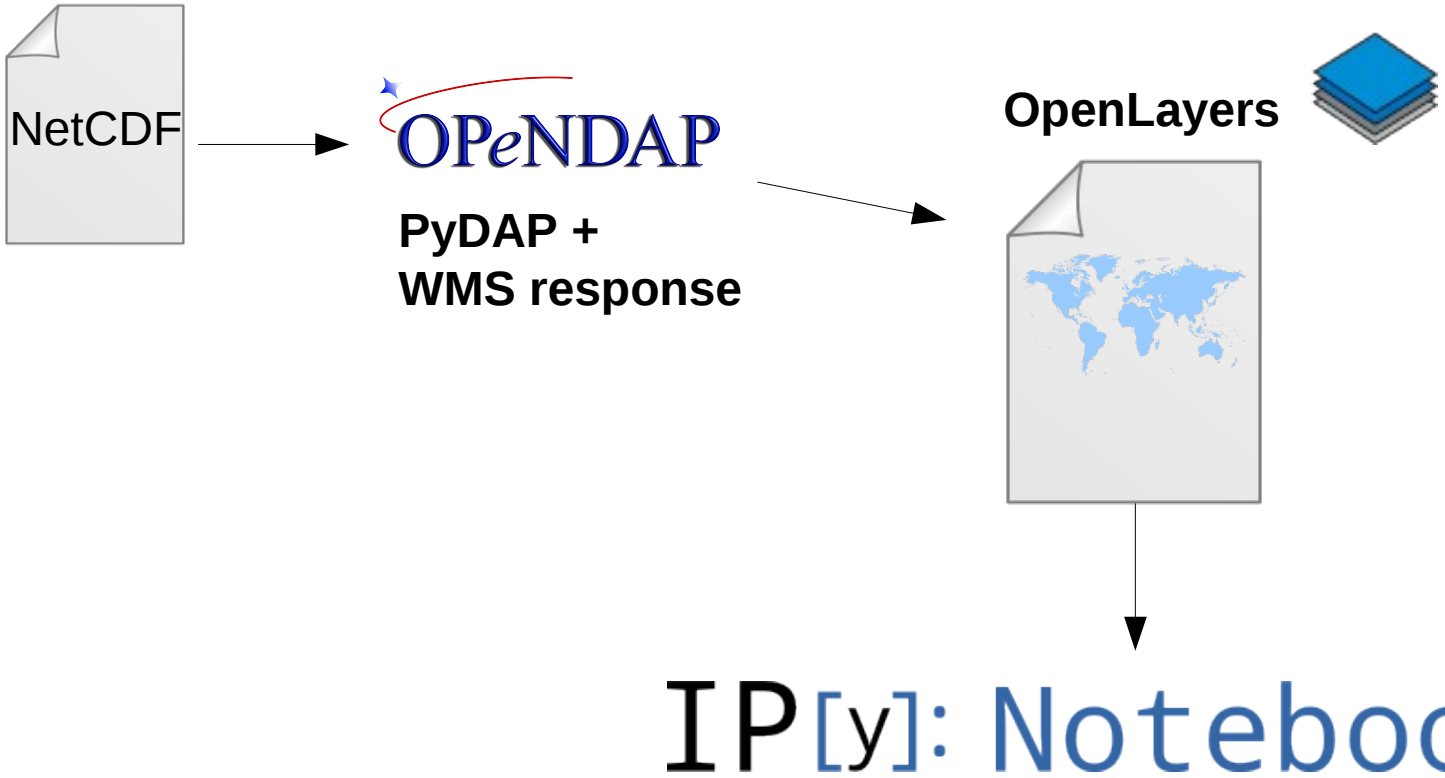
LEGEND

Temperature (°C)

45 +
42 – 45
39 – 42
36 – 39
33 – 36
30 – 33
27 – 30
24 – 27
21 – 24
18 – 21
15 – 18
12 – 15
9 – 12
6 – 9
3 – 6
0 – 3
-3 – 0
-6 – -3
< -6
no data

How to get there?





Australian Government
Bureau of Meteorology

The Centre for Australian

IPython notebook

"a web-based interactive computational environment where you can combine code execution, text, mathematics, plots and rich media into a single document"

- Inspired by Mathematica, SAGE notebooks
- Built-in support for numpy, matplotlib
- Half-interpreter, half-script
- Great way to "show your work"
- Useful for tuning fiddly APIs (ahem matplotlib)
- Perfect for tutorials!

Run a local server, and/or

Publish your notebook as .ipynb and use http://nbviewer.ipython.org/

IPython notebook

```
pip install ipython
pip install tornado pyzmq  # needed for notebook, not ipython shell
ipython notebook --pylab inline
```

IPython notebook

Rich display – essentially more types of `__repr__`

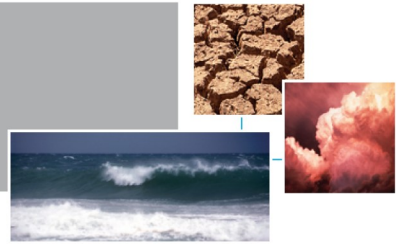
- **HTML**
- **JSON**
- **PNG, JPEG, SVG**
- **LaTeX**

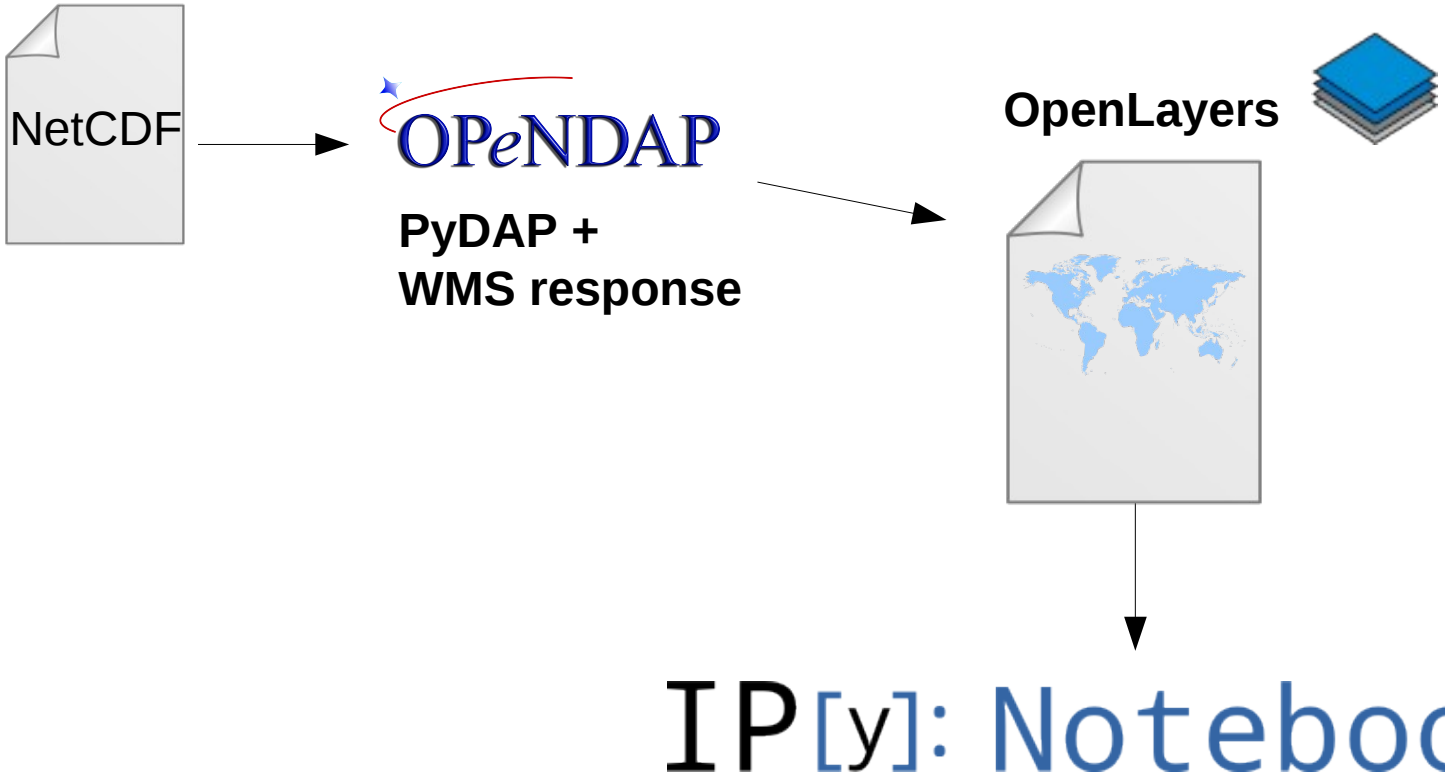
Expect this API to change (improve) in future versions, esp. for JavaScript

Make a class "Notebook compatible" by adding `html`, `png` etc properties:

```
from IPython.display import Image
class Foo(object):
    ….

    @property
    def png(self):
        Return Image(...)
```

How to get there?





Australian Government
Bureau of Meteorology

The Centre for Australian

JavaScript mapping libraries

Leaflet and OpenLayers:

- ≈**Specify layers as “tile layers” or “WMS tile layers” (different APIs)**
- ≈**Can set projection as required**
- ≈**Can reproject points and vectors (but not map tiles) on-the-fly**

WMS

Web Map Service

Supports 2+ request types – GetMap, GetCapabilities

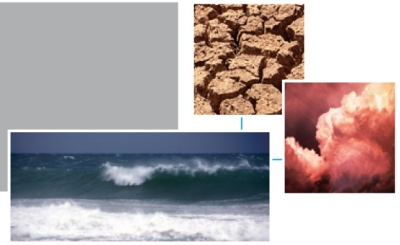
http://test.pydap.org/coads.nc.wms?SST[0]&LAYERS=SST&TRANSPARENT=true&FORMAT=image%2Fpng&SERVICE=WMS&VERSION=1.1.1&REQUEST=GetMap&STYLES=&SRS=EPSG%3A900913&BBOX=-180,-90,1252164.27125,1252254.27125&WIDTH=256&HEIGHT=256

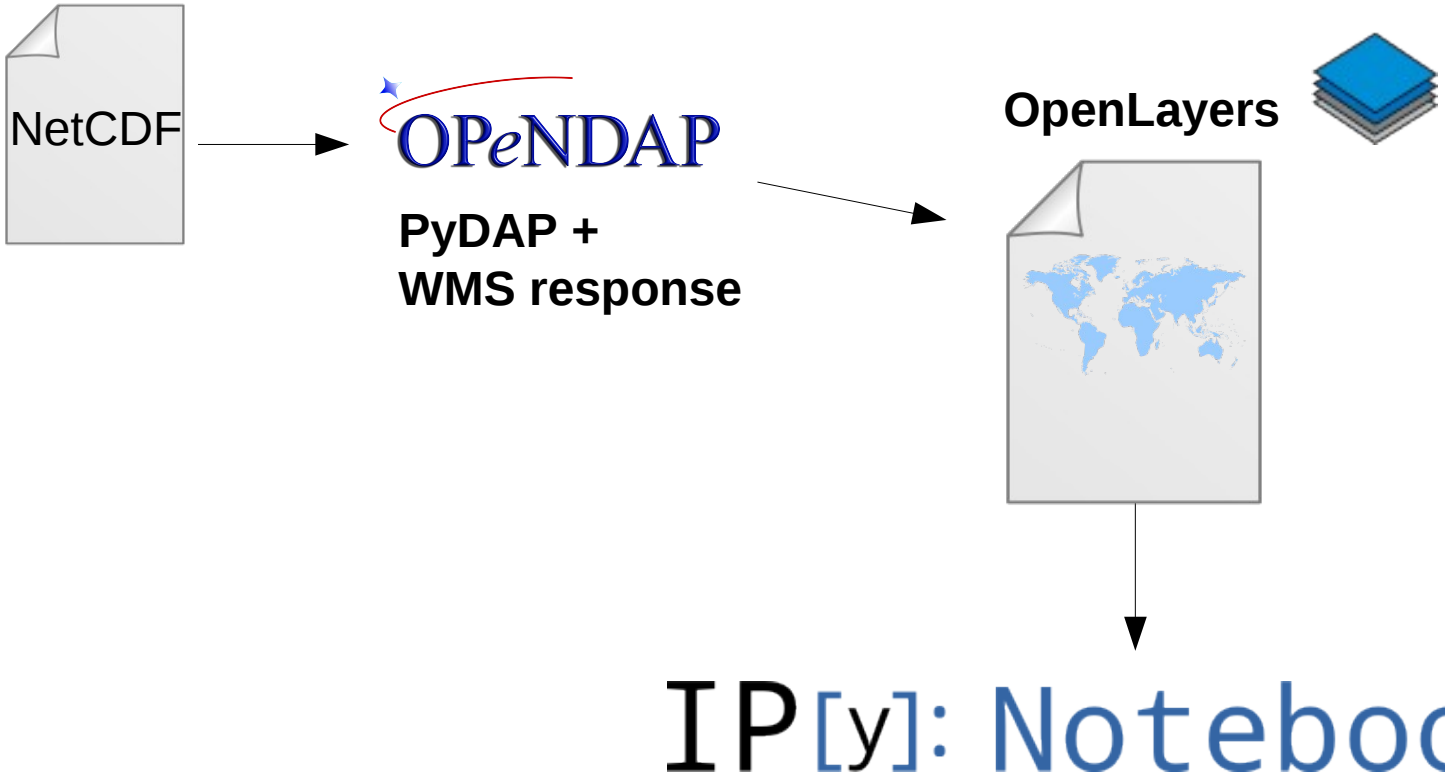
REQUEST=GetMap
SERVICE=WMS
VERSION=1.1.1

LAYERS=MinT_SFC
SRS=EPSG:900913
BBOX=-180,-10018844.17,10018574.17,-90

FORMAT=image/png
TRANSPARENT=true
HEIGHT=256
WIDTH=256

How to get there?





Australian Government
Bureau of Meteorology

The Centre for Australian

PyDAP

2 parts: OPeNDAP server, OPeNDAP client

Client

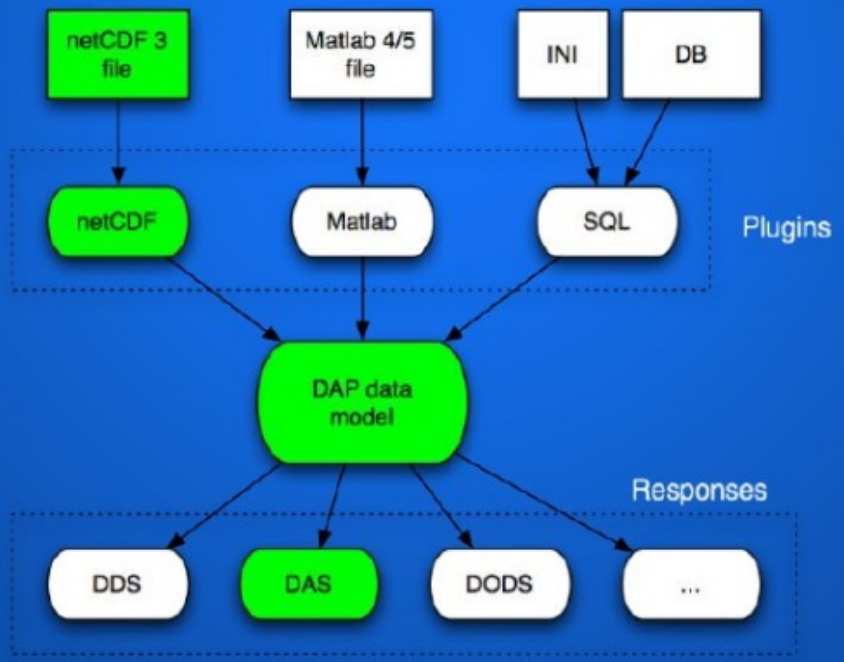
- “Lazy loading” of data from any OPeNDAP server into numpy (incl. slices, subsets)
- Don't have to care about file format of original data

Server

- WSGI app
- Handlers (input): NetCDF, HDF5, SQL, CSV, remote!, ...
- Responses (output): DAS/DDS/DODS, HTML, ASCII, WMS, KML, XLS, ...

PyDAP

Plugins and responses



http://localhost:8080/file.nc.das

PyDAP

```
pip install pydap pydap.handlers.netcdf pydap.responses.wms
paster create -t pydap myserver
cp mydata.nc myserver/data/

# starts a server like http://test.pydap.org/
# run server with 4 workers, better for serving up map tiles
gunicorn_paster -w 4 -b 127.0.0.1:8001 myserver/server.ini
```

PyDAP

Index of /

Parent directory

Filename	Download	Last modified
IDT71003_TAS_MinT_SFC.nc	1 MB	06/25/2013 06:52:40 PM
coads.nc	5 MB	06/25/2013 07:01:39 PM

pydap/3.1

Australian Government
Bureau of Meteorology

The Centre for Australian

PyDAP

127.0.0.1:8001/IDT71003_TAS_MinT_SFC.nc.html

Google

Download data from IDT71003_TAS_MinT_SFC%2Enc

Parent directory

Global attributes

NC_GLOBAL

creationTimeString
Wed Mar 28 21:59:44 2012
creationTime
1332971984
Conventions
COARDS

HTML form | Map | Ferret | GrADS | IDL | Pydap | Other clients

Downloading data

In the form below you can specify desired variables and their dimensions, and have the data downloaded in different formats depending on the server configuration.

- latitude
- time
- longitude
- MinT_SFC

Download data as ASCII
Reset

Australian Government
Bureau of Meteorology

The Centre for Australian

PyDAP

HTML form | Map | Ferret | GrADS | IDL | Pydap | Other clients

Downloading data with Pydap

To access this dataset using the Pydap Python module:

```
$ python
>>> from pydap.client import open_url
>>> dataset = open_url("http://127.0.0.1:8001/IDT71003_TAS_MinT_SFC.nc")
>>> import pprint
>>> pprint.pprint( dataset.keys() )
['latitude', 'time', 'longitude', 'MinT_SFC']
```

Projections 101

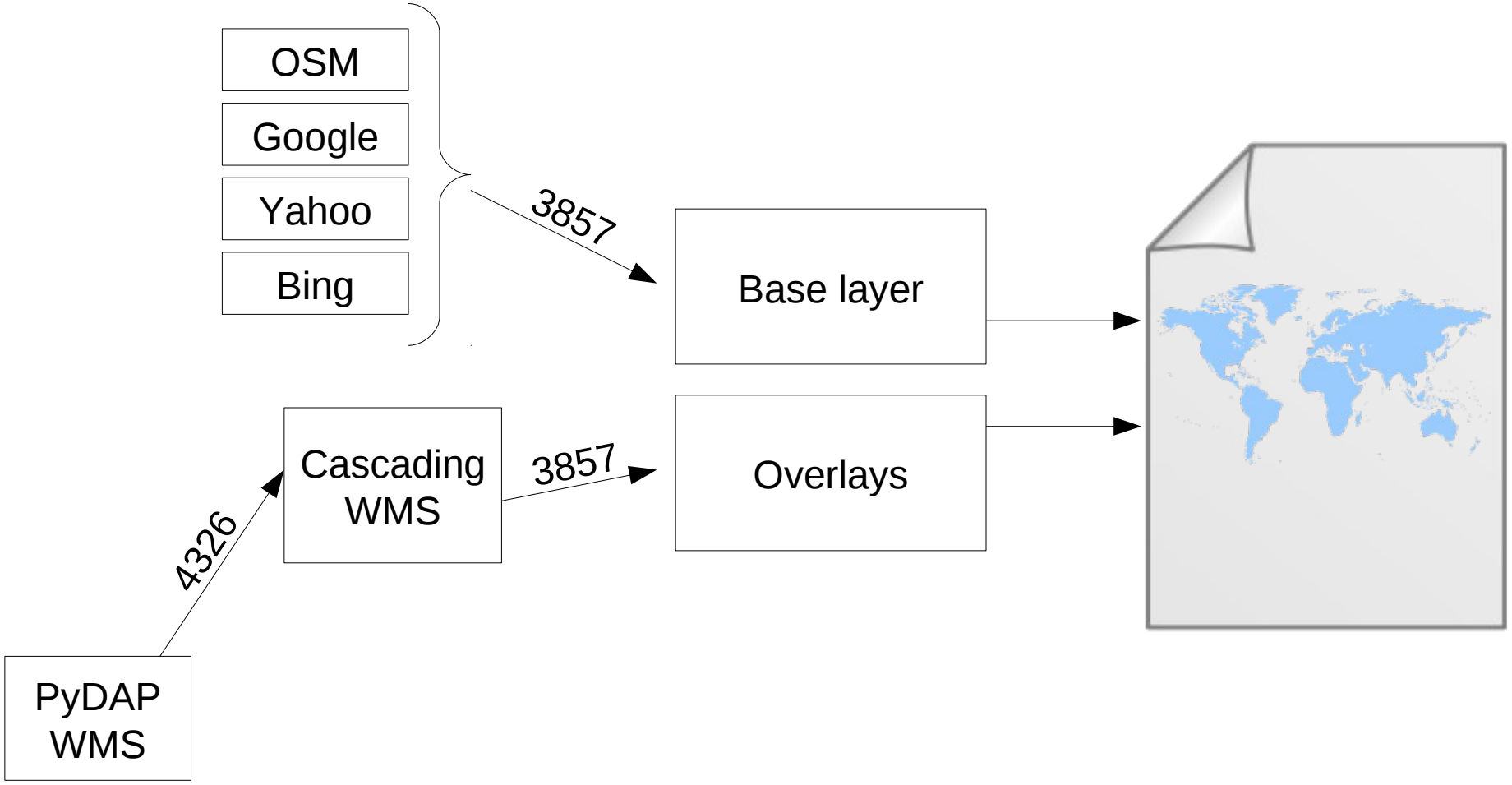
Mapping libraries can reproject points and vectors, but not map tiles...

Base layers and overlays need to be requested in the same projection!

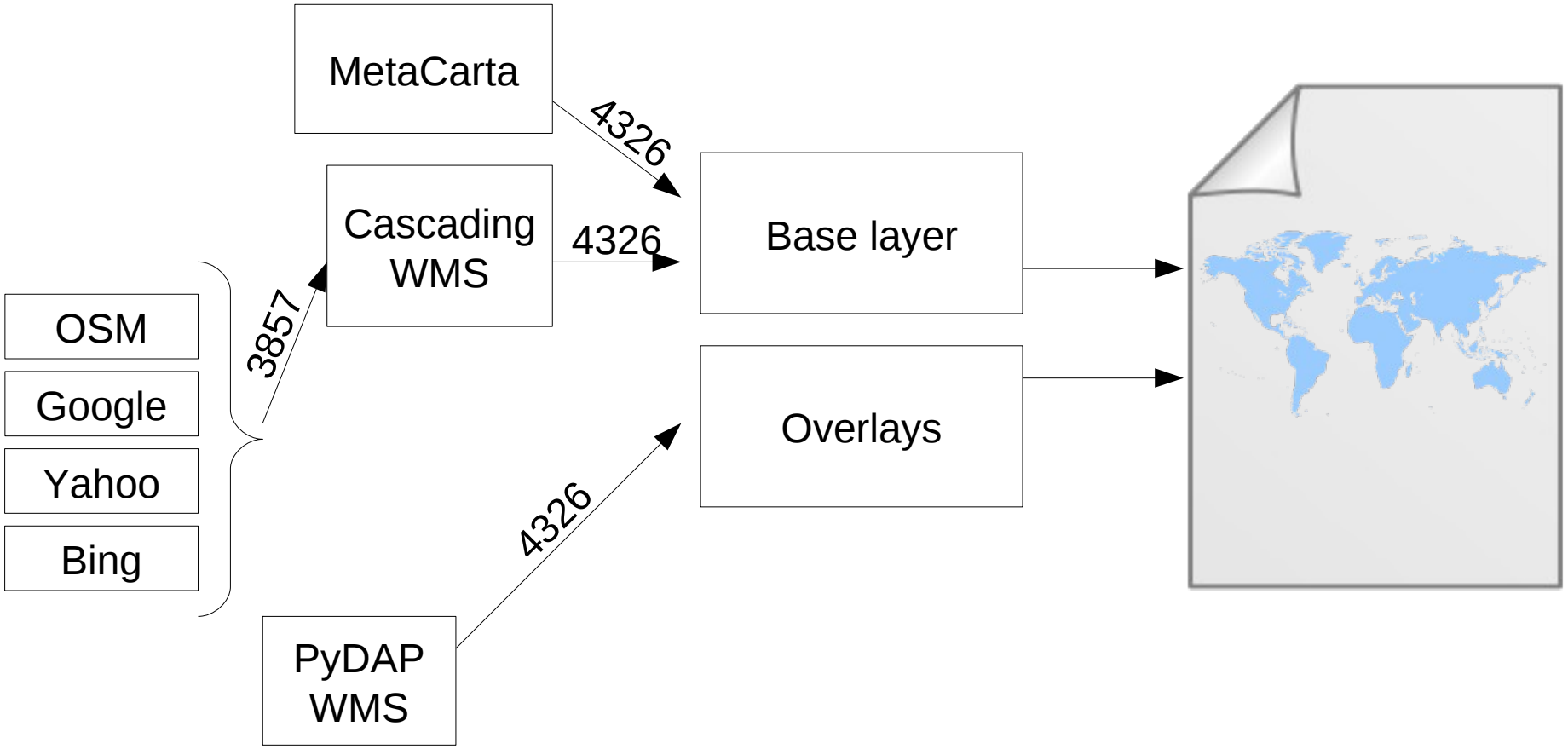
Projections 101

CRS	**EPSG:4326**	**EPSG:3857 (aka EPSG:900913)**
Example of	Geographic coordinate system (represents a globe/part of a globe)	Projected coordinate system, “Spherical Mercator” (represents a map)
Used by	“GIS enthusiasts”, Metacarta	“almost all free and commercial tile providers” (Google, OSM, Yahoo, Bing)
Coords as	Lat-lons in decimal degrees	“Metres”
PyDAP WMS can serve map tiles in	√	
OpenLayers supports by default	√	√
Leaflet supports by default	√ But... issue #1207 “EPSG 4326 Support Broken for TileLayers” and... CloudMade doesn't provide WMS.	√

Option 1: Render overlays in EPSG:3857



Option 2: Pick base layer in EPSG:4326



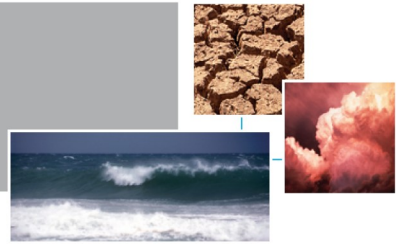
Cascading WMS

- MapServer
  - C
  - MIT license
- GeoServer + GeoWebCache
  - Java
  - GPL, LGPL licenses
  - http://maps.opengeo.org/geowebcache/demo
    - Includes OpenStreetMap, NASA “Blue Marble”

Putting it all together

Australian Government
Bureau of Meteorology

The Centre for Australian



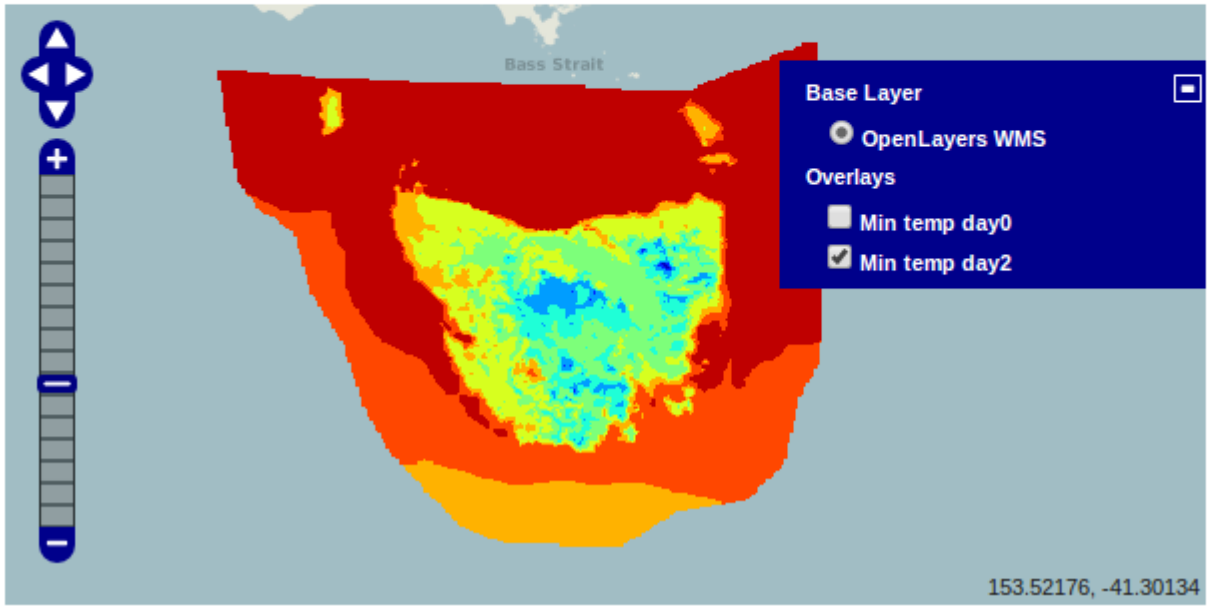
```python
import os
from genshi.template import TemplateLoader
from IPython.display import HTML

loader = TemplateLoader(os.getcwd())


class PydapWMS(object):
    """A simple object for displaying data from a Pydap WMS."""
    def __init__(self, server='http://test.pydap.org/coads.nc.wms',
                 layers=None, centerlat=0, centerlon=0, initialzoom=2,
                 mapwidth=400, mapheight=300, template='template.html'):
        self.wmsserverurl = server
        self.layers = layers if layers else [('SST', 'Sea Surface Temperature', 0)]
        self.centerlat = centerlat
        self.centerlon = centerlon
        self.initialzoom = initialzoom
        self.mapwidth = mapwidth
        self.mapheight = mapheight
        self.template = template

    def __repr__(self):
        return 'PydapWMS({})'.format(str(self.__dict__))

    @property
    def html(self):
        tmpl = loader.load(self.template)
        html = tmpl.generate(**self.__dict__).render('html', doctype='html')
        src = 'data:text/html;base64,{}'.format(html.encode('base64'))
        iframe = '<iframe src="{}" width="{}" height="{}"></iframe>'
        return HTML(iframe.format(src, self.mapwidth + 50, self.mapheight + 50))
```

template.html

```
19          var openlayer = new OpenLayers.Layer.WMS("OpenLayers WMS",
20                  "http://maps.opengeo.org/geowebcache/service/wms",
21                  {layers: 'openstreetmap', 'format': 'image/png'});
22          map.addLayer(openlayer);
23          <py:for each="(layerid, layername, layerindex) in layers">
24              map.addLayer(new OpenLayers.Layer.WMS(
25                  "${layername}",
26                  "${wmsserverurl}?${layerid}[${layerindex}]",
27                  {
28                      layers: "${layerid}",
29                      transparent: "true",
30                      format: "image/png",
31                  },
32                  { isBaseLayer: false}));
33          </py:for>
34          map.setCenter(new OpenLayers.LonLat(${centerlon}, ${centerlat}), ${initialzoom});
35          }
36      </script>
37  </head>
38  <body onload='init();'><div id="map" style="width: ${mapwidth}px; height: ${mapheight}px"></div>
39  </body>
40  </html>
```

Back to the notebook

```
import dapbook
b = dapbook.PydapWMS(server='http://127.0.0.1:8001/IDT71003_TAS_MinT_SFC.nc.wms',
                     layers=[('MinT_SFC', 'Min temp day0', 0),
                             ('MinT_SFC', 'Min temp day2', 2),
                     ], centerlat=-42, centerlon=147, initialzoom=6,
                     mapwidth=600, mapheight=300)
b.html
```



Australian Government
Bureau of Meteorology

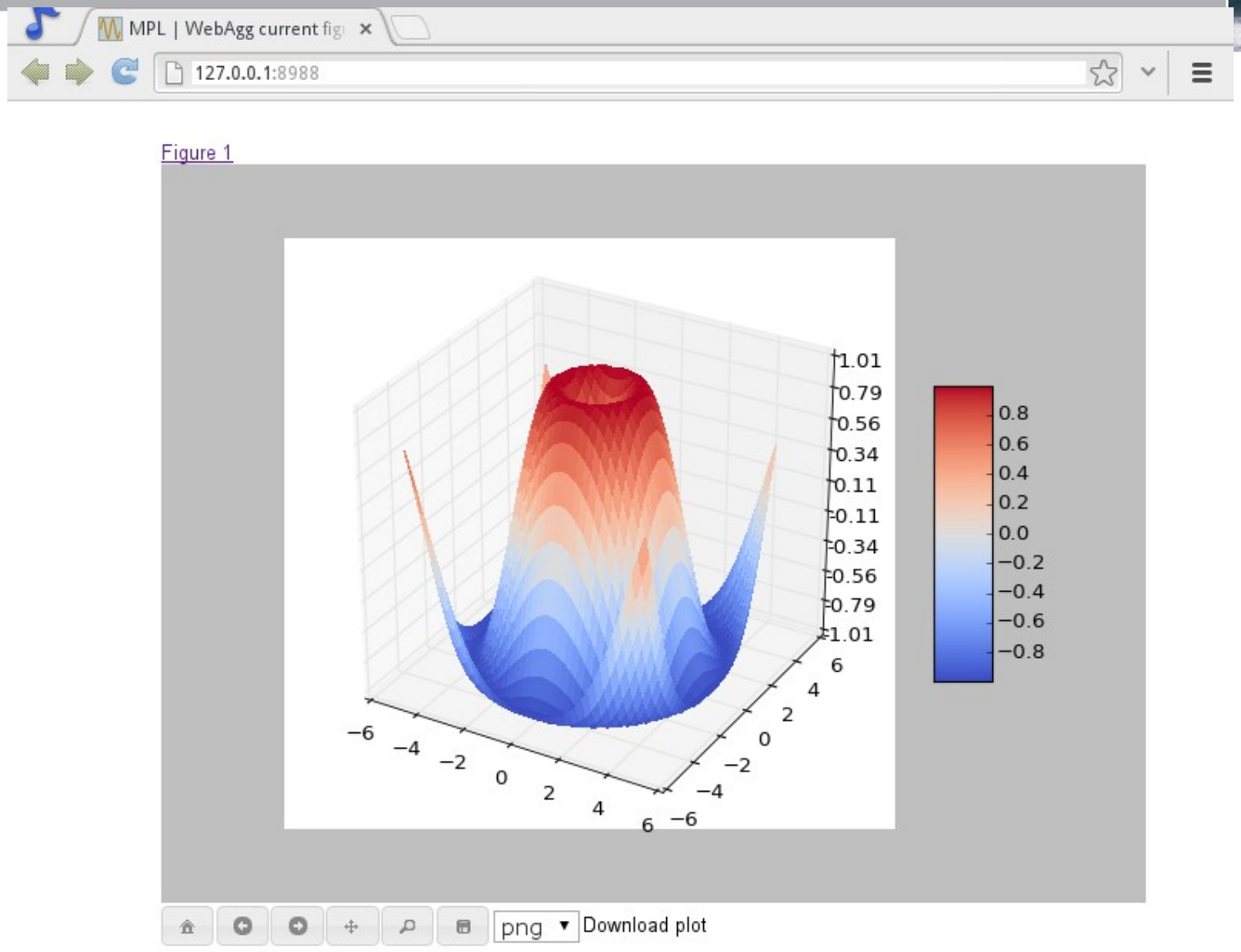
The Centre for Australian

Other options – GeoDjango?

First impression -
seems to be a lot more about vector layers than gridded data

No particular mention of netcdf, gridded data, WMS

The future – HTML5 backend for matplotlib?



Credits

With thanks to:

Nathan Faggian, for the original leavis, and willingness to share his ideas

Leafvis – a WMS-ish app, DIY map tile rendering.

https://github.com/nfaggian/leafvis

And also:

Roberto De Almeida

James Sofra

Danielle Madeley

Roald de Wit

Credits

Map tiles photo: “Carcassonne” by Tom & Katrien, licensed CC-BY-SA.
http://www.flickr.com/photos/inferis/283379928/

PyDAP slide: by PyCon 2007 talk by Roberto De Almeida.
http://www.scribd.com/doc/2864/PyCon-2007

Australian Government
Bureau of Meteorology

The Centre for Australian Weather and Climate Research
A partnership between CSIRO and the Bureau of Meteorology

CSIRO

Brianna Laugher

Email: b.laugher@bom.gov.au
Web: www.cawcr.gov.au
Slides & code:
https://github.com/pfctdayelise/dapbook

Thank you

www.cawcr.gov.au