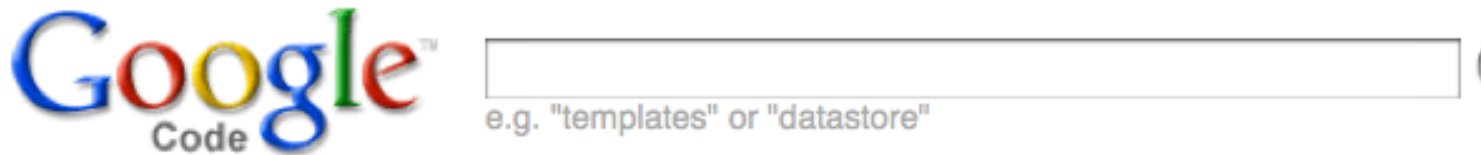


1. Put your Webapp in the Cloud
2. Virtual Observatory
3. Web Services

django in the cloud

- deploy your webapp to googleland
– where no evil can be done!



Google App Engine

[Home](#)

[Docs](#)

[FAQ](#)



Run your web apps on Google's infrastructure.

Easy to build, easy to maintain, easy to scale.

django in the cloud

- Google App Engine
 - Run on Google machines
 - Your own domain or.appspot.com
 - Java or Python
 - Persistent storage
 - Free but many quotas
 - Scalable – advantage of the cloud
 - Does one thing well: running web apps
- Amazon
 - LAMP based
 - not so easy with django ...

App Engine Does One Thing Well

- App Engine handles HTTP(S) requests, nothing else
 - Think RPC: request in, processing, response out
 - Works well for the web and AJAX; also for other services
- App configuration is dead simple
 - No performance tuning needed
- Everything is built to scale
 - “infinite” number of apps, requests/sec, storage capacity

Security

- Prevent the bad guys from breaking (into) your app
- Constrain direct OS functionality
 - no processes, threads, dynamic library loading
 - no sockets (use urlfetch API)
 - can't write files (use datastore)
 - disallow unsafe Python extensions (e.g. ctypes)
- Limit resource usage
 - Limit 1000 files per app, each at most 1MB
 - Hard time limit of 10 seconds per request
 - Most requests must use less than 300 msec CPU time
 - Hard limit of 1MB on request/response size, API call size, etc.
 - Quota system for number of requests, API calls, emails sent, etc
 - Everything an app does is limited by quotas, for example:
 - request count, bandwidth used, CPU usage, datastore call count, disk space used, emails sent, even errors!
 - If you run out of quota that particular operation is blocked (raising an exception) for a while (~10 min) until replenished

Why Not LAMP?

- Linux, Apache, MySQL/PostgreSQL, Python/Perl/PHP/Ruby
- LAMP is the industry standard
- But management is a hassle:
 - Configuration, tuning
 - Backup and recovery, disk space management
 - Hardware failures, system crashes
 - Software updates, security patches
 - Log rotation, cron jobs, and much more
 - Redesign needed once your database exceeds one box
- “We carry pagers so you don’t have to”

google apps and django

- easy to port django to google apps
- add special main.py to django app
 - launches it
- add app.yaml
- django database api is like google datastore

django

```
class Greeting(models.Model):  
    author = models.ForeignKey(Event)  
    content = models.TextField()  
    date = models.DateTimeField()
```

appengine

```
class Greeting(db.Model):  
    author = db.UserProperty()  
    content = db.StringProperty(multiline=True)  
    date = db.DateTimeProperty()
```

google apps and django

django

```
urlpatterns = patterns('',
    (r'^polls/$', 'pollapp.polls.views.index'),
    (r'^polls/(?P<poll_id>\d+)/$', 'pollapp.poll.views.detail'),
    (r'^polls/(?P<poll_id>\d+)/results/$', 'pollapp.poll.views.results'),
    (r'^polls/(?P<poll_id>\d+)/vote/$', 'pollapp.poll.views.vote'),
)
```

appengine

```
def main():
    application = webapp.WSGIApplication([('/', MainPage),
                                          ('/tryit', ImagePage),
                                          ('/imagedata', ImageData),
                                          ('/sample', SamplePage),
                                          ('/viewsource', ViewSource),
                                          ],
                                          debug=True)
    wsgiapp.handlers CGIHandler().run(application)
```


put your webapp in the google cloud

step 1: get the SDK

Download the Google App Engine SDK

By downloading, you agree to be bound by the [Terms](#) that govern use of the App Engine SDK.

Please note: The App Engine SDK is under **active development**, please keep this in mind as you explore its capabilities. See the [SDK Release Notes](#) for the information on the most recent changes to the App Engine SDK. If you discover any issues, please feel free to notify us via our [Issue Tracker](#).

Google App Engine SDK for Python

Platform	Version	Package	Size	SHA1 Checksum
Windows	1.2.1 - 04/22/09	GoogleAppEngine_1.2.1.msi	2.7 MB	6e16a9845962edcaa5c4ceb03b0e30785ebdb2e1
Mac OS X	1.2.1 - 04/22/09	GoogleAppEngineLauncher-1.2.1.dmg	3.8 MB	1b88bb2ca36850da934602acb7a0c2d652c4762a
Linux/Other Platforms	1.2.1 - 04/22/09	google_appengine_1.2.1.zip	2.8 MB	06d8300b202ff574ff7c6f0e38709af540c76f4b

For more information on the SDK for Python:

- [Overview](#)
- [Getting Started Guide](#)

Google App Engine SDK for Java Early Look

Version	Package	Size	SHA1 Checksum
1.2.0 - 04/07/09	appengine-java-sdk-1.2.0.zip	18.6 MB	a2c486bb088320b5045bfe3ae552e24d9438f0d4

For more information on the Google App Engine SDK for Java:

- [Overview](#)
- [Getting Started Guide](#)

step 2: port to google datastore

(looks like django model!)

```
from google.appengine.ext import db
```

```
class Greeting(db.Model):  
    author = db.UserProperty()  
    content = db.StringProperty(multiline=True)  
    date = db.DateTimeProperty(auto_now_add=True)
```

```
g = Greeting(author=users.get_current_user(), content="hello!", date=now())  
greetings = Greeting.all()  
g = Greeting.all(). filter(author=users.get_current_user()). order('-date')
```

BUT also have GQL:

```
greetings = db.GqlQuery("SELECT * FROM Greeting ORDER BY date DESC LIMIT 10")
```

step 3: running locally

bluegreenhat/
main.py
app.yaml

```
# main.py
import os, sys
os.environ.setdefault('SETTINGS_MODULE', 'bluegreenhat.settings')
sys.path.append(os.path.join(os.path.dirname(__file__), 'handlers'))

# Google App Engine imports
from google.appengine.ext.webapp import urlmap

# Force Django to release its settings
from django.conf import settings
settings._target = None

import django.core.handlers.wsgi
import django.core.signals
import django.db
import django.dispatch

# Log errors
def _django_exception_dispatcher_connect(
    _django_exception_dispatcher_disconnect):
    _django_exception_dispatcher_disconnect()
    _django_exception_dispatcher_disconnect()
    _django_exception_dispatcher_disconnect()

def main():
    # Create a Django application for WSGI
    application = django.core.handlers.wsgi.WSGIHandler()
    # Run the WSGI CGI handler with that application.
    urlmap.run_wsgi_application(application)

if __name__ == '__main__':
    main()
```

```
$ dev_appserver.py myapp/
```

point browser to localhost:8080

```
application: bluegreenhat
version: 1
runtime: python
api_version: 1
handlers:
- url: /*
  script: main.py
```

```
# main.py

import os, sys
os.environ["DJANGO_SETTINGS_MODULE"] = "mashname.settings"
sys.path.append("/home/brox/tmp/mashname")

# Google App Engine imports.
from google.appengine.ext.webapp import util

# Force Django to reload its settings.
from django.conf import settings
settings._target = None

import django.core.handlers.wsgi
import django.core.signals
import django.db
import django.dispatch.dispatcher

# Log errors.
#django.dispatch.dispatcher.connect(
#    log_exception, django.core.signals.got_request_exception)

# Unregister the rollback event handler.
django.dispatch.dispatcher.disconnect(
    django.db._rollback_on_exception,
    django.core.signals.got_request_exception)

def main():
    # Create a Django application for WSGI.
    application = django.core.handlers.wsgi.WSGIHandler()

    # Run the WSGI CGI handler with that application.
    util.run_wsgi_app(application)

if __name__ == "__main__":
    main()
```

main.py

step 4: register, upload, run

Registering the Application

You create and manage App Engine web applications from the App Engine Administration Console, at the following URL:

<http://appengine.google.com/>

Sign in to App Engine using your Google account. If you do not have a Google account, you can [create a Google account](#) with an email address and password.

To create a new application, click the "Create an Application" button. Follow the instructions to register an application ID, a name unique to this application. If you elect to use the free appspot.com domain name, the full URL for the application will be <http://application-id.appspot.com/>. You can also purchase a top-level domain name for your app, or use one that you have already registered.

Edit the `app.yaml` file, then change the value of the `application:` setting from `helloworld` to your registered application ID.

Uploading the Application

To upload your finished application to Google App Engine, run the following command:

```
appcfg.py update helloworld/
```

Enter your Google username and password at the prompts.

You can now see your application running on App Engine. If you set up a free appspot.com domain name, the URL for your website begins with your application ID:

<http://application-id.appspot.com>

Interested in trying our new Java language support? [Learn more](#)

Dismiss

Create an Application

Application Identifier:

.appspot.com Yes, "bluegreenhat" is available!

You can map this application to your own domain later. [Learn more](#)

Application Title:

Displayed when users access your application.

Authentication Options (Advanced): [Learn more](#)

Google App Engine provides an API for authenticating your users. If you choose not to use this, anyone in the world will be able to access your application. However, if you choose to use this, you'll need to specify now who can sign in to your application:

Open to all Google Accounts users (default)

If your application uses authentication, anyone with a valid Google Account may sign in. (This includes all Gmail Accounts, but does "not" include accounts on any Google Apps domains.)

[Edit](#)

Terms of Service:

I. Your Agreement with Google

1.1. Your use of the Google App Engine service (the "Service") is governed by this agreement (the "Terms"). "Google" means Google Inc., located at 1600 Amphitheatre Parkway, Mountain View, CA 94043, United States, and its subsidiaries or affiliates involved in providing the Service.

1.2. In order to use the Service, you must first agree to the Terms. You can agree to the Terms by actually using the Service. You understand and agree that Google will treat your use of the Service

I accept these terms.

Application Registered Successfully

The application will use **bluegreenhat** as an identifier. This identifier belongs in your application and this identifier cannot be changed.

If you use Google authentication for your application, **Blue Green Hat** will be displayed on your application.

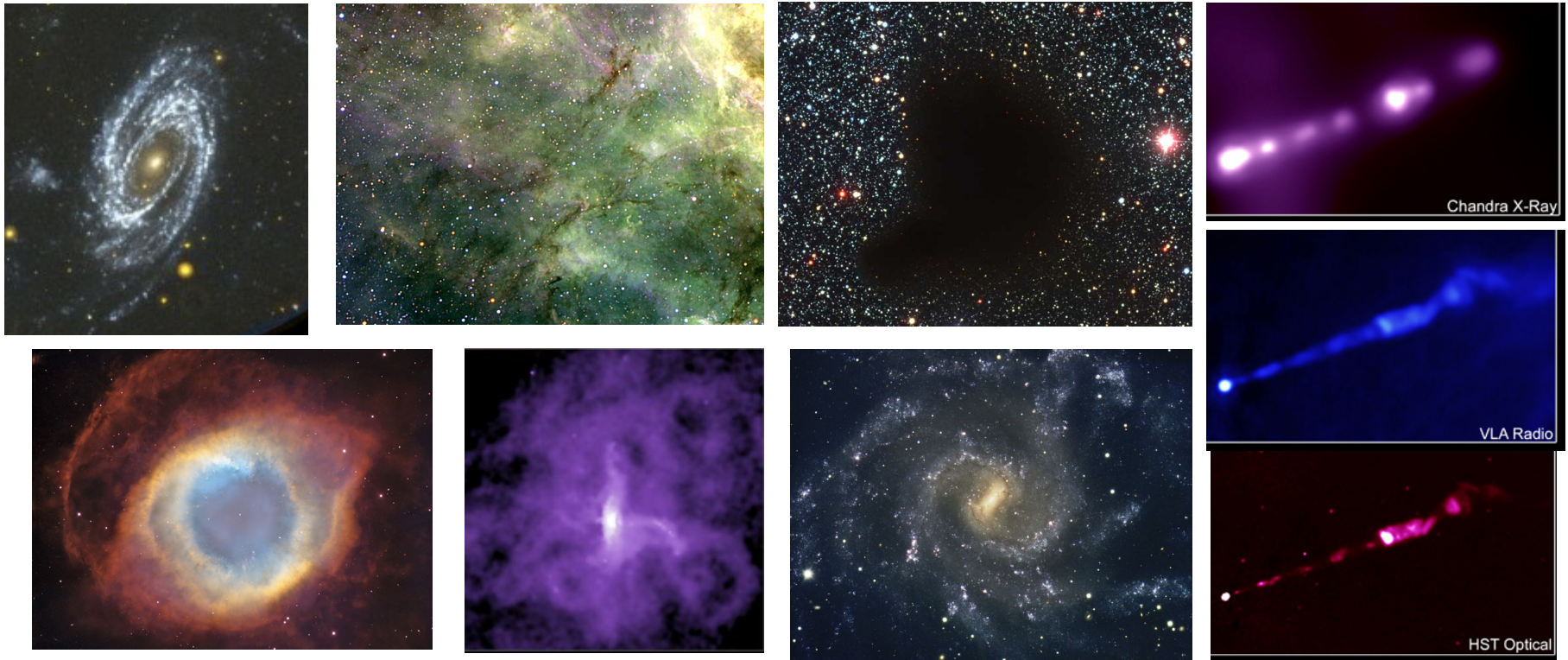
Choose an option below:

- View the [dashboard](#) for Blue Green Hat.
- Use [appcfg.py](#) to upload and deploy your application code.
- Add [developers](#) to collaborate on this application.

Virtual Observatory

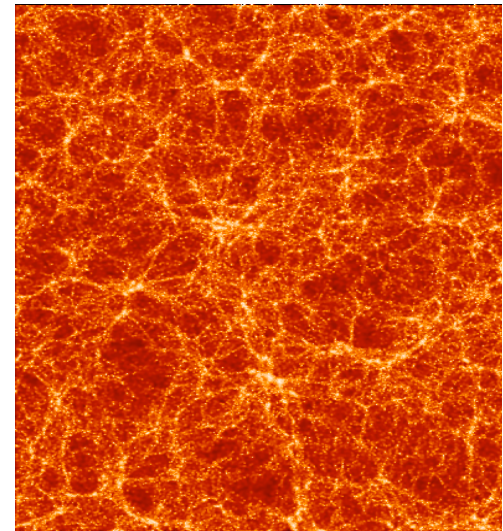
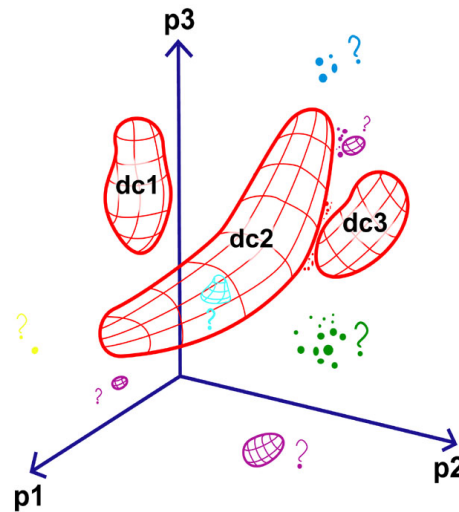
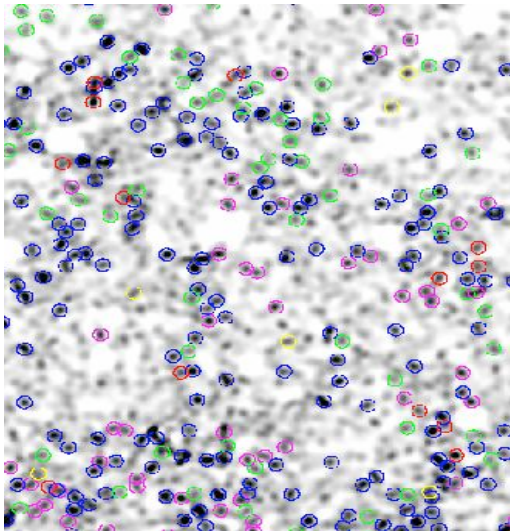
Toward a “new astronomy”

- Past:
Observations of small, carefully selected samples



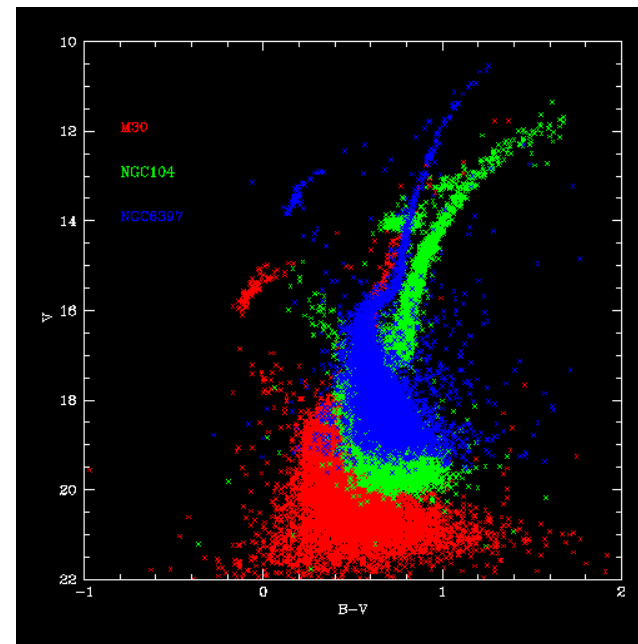
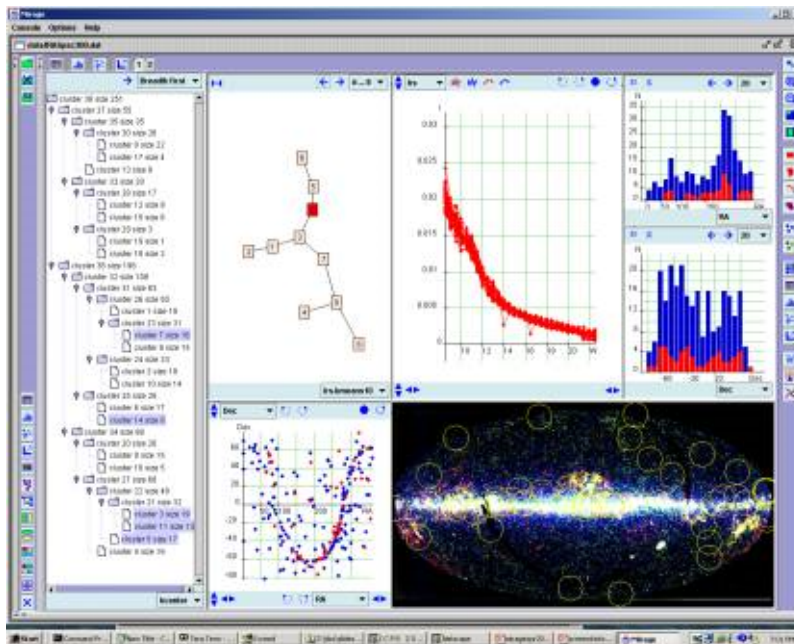
Toward a “new astronomy”

- Multi-wavelength data for millions of objects
- Data Mining, Outliers, Correlations, etc
- Theory vs Experiment



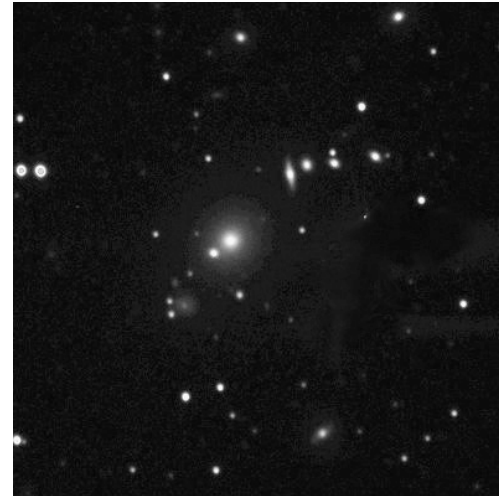
Toward a “new astronomy”

- Tools
- Visualization
- Data Mining

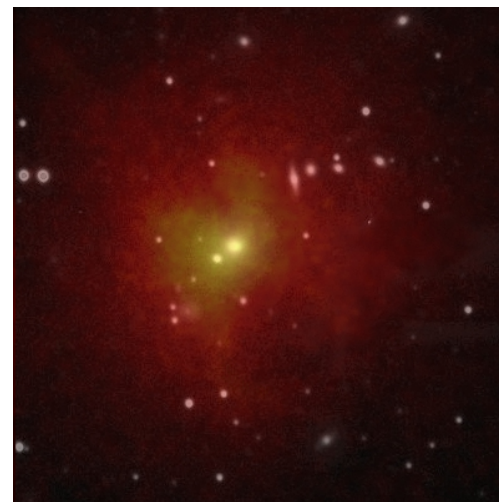


Virtual Observatory is:

- Uniform interfaces
 - standards
- Data discovery
 - & publication
- Data federation
 - science is in the join
- Usability
 - “I just want to”
- Remote analysis
 - big data, services, storage
- Big computing
 - terabytes & teraflops



Optical



Optical + X-ray

Hydra A

OpenSkyQuery

Open SkyQuery - Microsoft Internet Explorer

File Edit View Favorites Tools Help

NVO Open SkyQuery

Home Query Import Tutorial Help

National Virtual Observatory

Nodes

- Rosat
- XMM-Newton
- GALEX
- DL5
- RC3
- SDSS
- SDSSDR2
- TwoDf
- Twoqz
- USNOB
- GOODS
- HDFN
- HDFS
- UDF
- ISO
- TWOMASS
- IRAS
- PSCz
- ADIL
- FIRST
- IVSS
- NDWFS
- IVORegistry
- sxds_skycode

Build **Edit** **Submit**

```
SELECT o.objid, o.ra,
       o.dec, o.type, t.objid,
       t.ra, t.dec, t.j_m,
       o.i
FROM
  SDSS:PhotoPrimary o, TWOMASS:PhotoPrimary t
WHERE XMATCH(o, t) < 3.5 AND
      Region('CIRCLE J2000 182.5 -0.89 8') AND
       o.type = 3 AND
       o.i < 21 AND
       t.j_m < 18 AND
       (o.i - t.j_m) > 2
```

SDSS:PhotoPrimary

- Add Selection
- Add Condition
- Remove 'PhotoPrimary'

objid, skyVersion, run, rerun, camcol, field, obj, mode, nChild

Sample Queries

- XMatch/Region
- XMatch/Region 2
- Three Node Match
- Brown Dwarf Search
- MyData XMatch (upload)
- Xmatch t.* (upload)
- ABELL Xmatch (upload)
- Single Node Query
- Single Node Join

Sigmas **Region** **Clear**

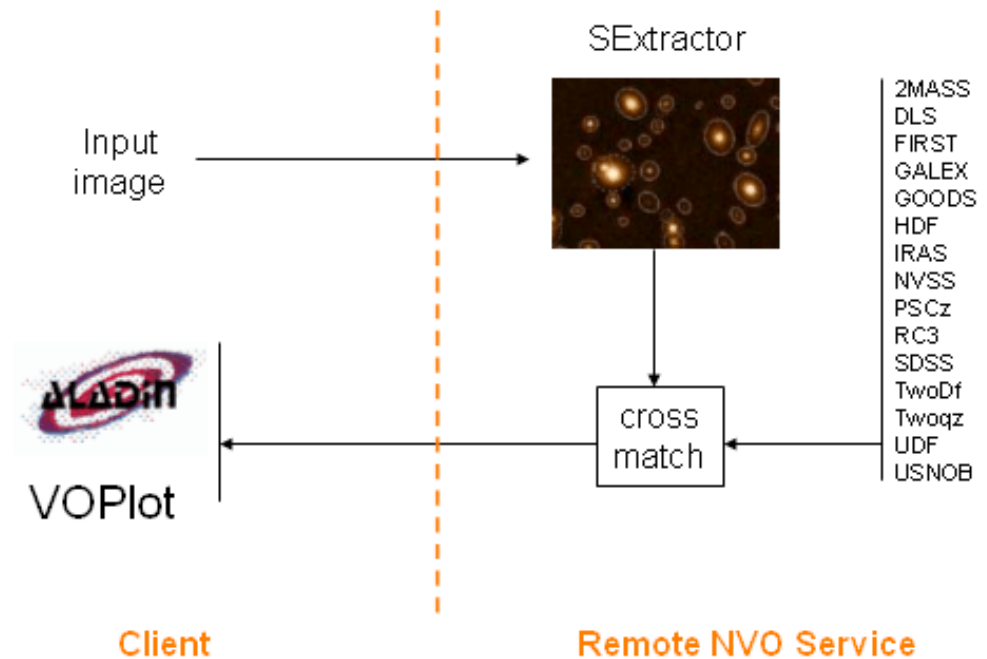
Version: v1_0_9
US-VO.org

Web Enabled Source Identification

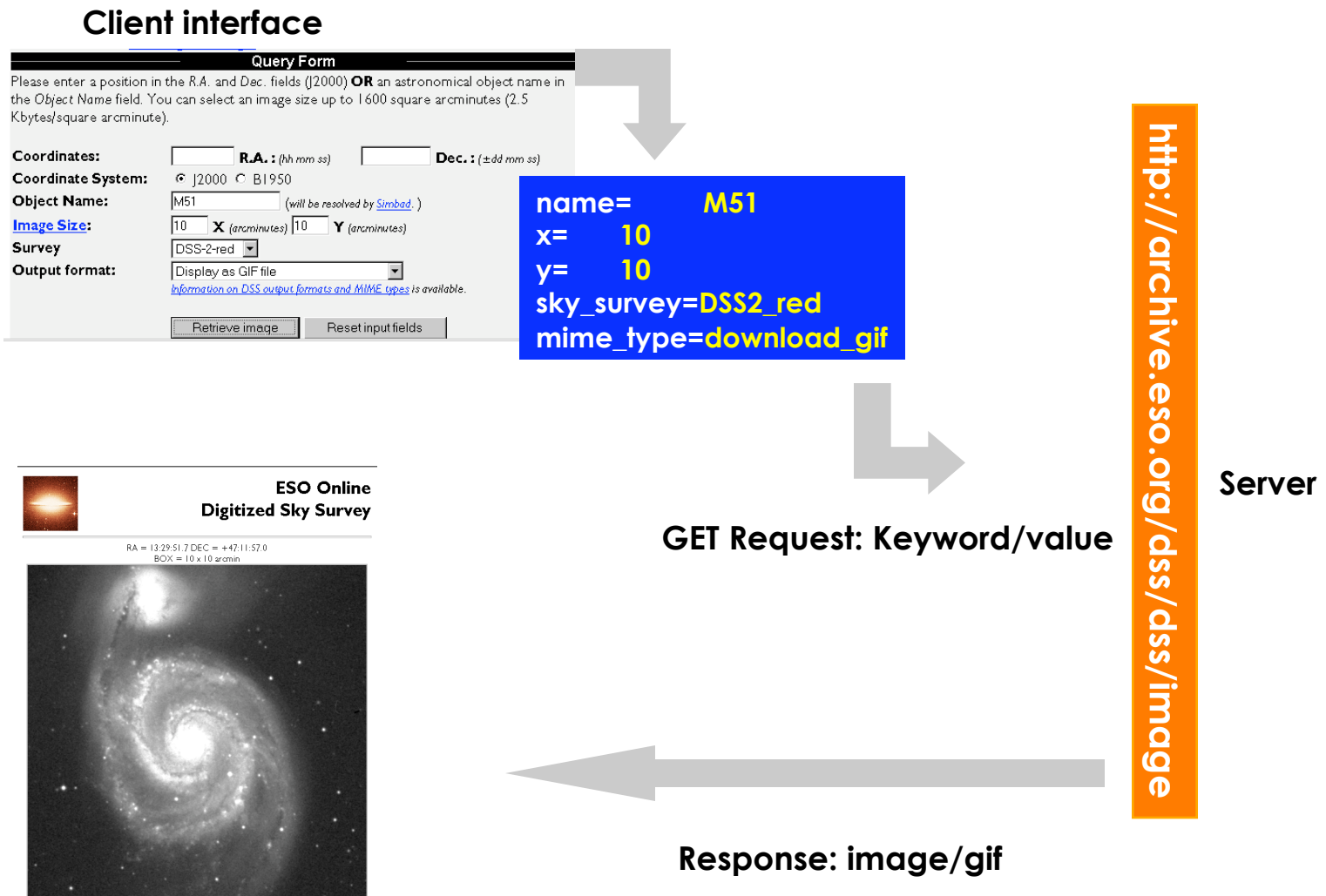
with Cross-Matching (WESIX)

Upload images to SExtractor and cross-correlate the objects found with selected survey catalogs.

This NVO service does source extraction and cross-matching for any astrometric FITS image. The user uploads a FITS image, and the remote service runs the SExtractor software for source extraction. The resulting catalog can be cross-matched with any of several major surveys, and the results returned as a VOTable. The web page also allows use of Aladin or VOPlot to visualize results.



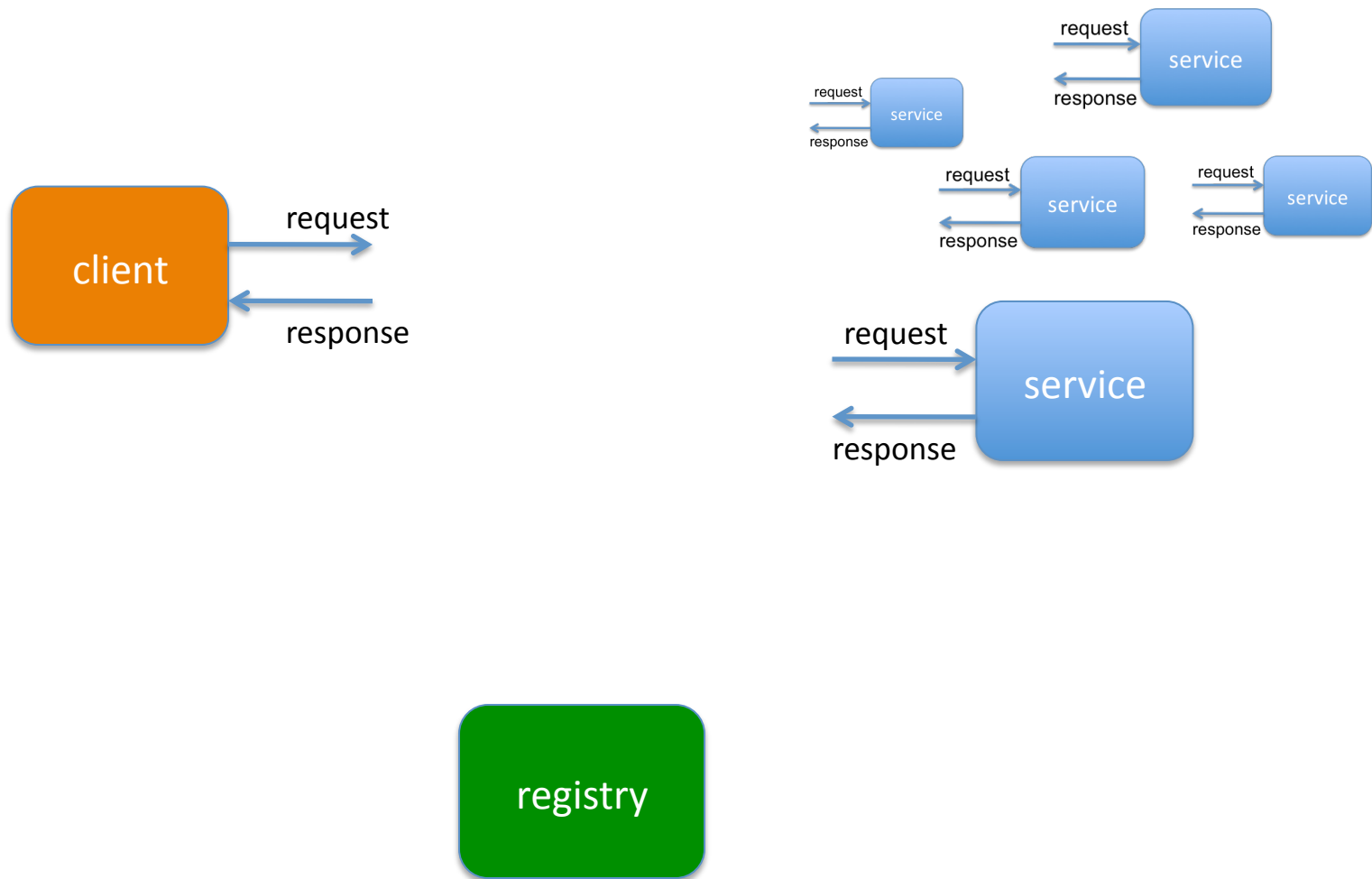
Example Service



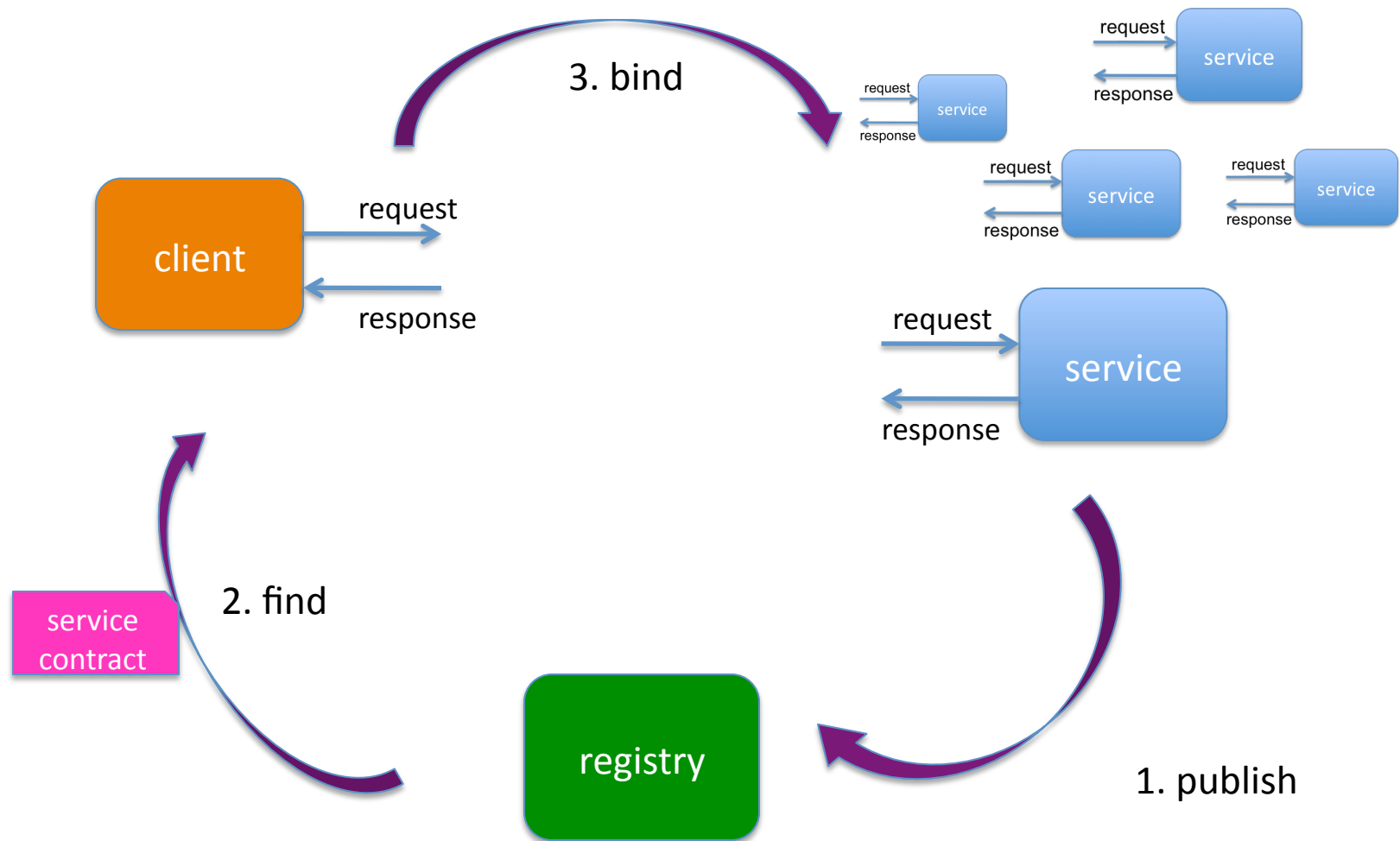
Web Services

- Principle: **Click or Code**
- Protocol: simple **REST/GET** or sophisticated **SOAP**
- Standards
 - Basic service profile
 - formal description (WSDL), input parameters, output formats, capabilities
 - VOResource (who and what is it)
 - Security (single sign-on, authentication and authorization)
 - Asynchronous (batch) services
 - Distributed data storage (VOSpace)
- Coordinated with IVOA, Open Grid Forum

Service Oriented Architecture



Service Oriented Architecture



VO Data Services

- Cone Search
 - First standard NVO service:
 - radius+position \Rightarrow list of objects
 - encoded as VOTable
- Simple Image Access Protocol
 - “cone search for images”
 - images are referenced by URL
- Simple Spectrum Access Protocol
 - spectra have subtleties \rightarrow protocol more complicated

VO Data Services

- **Astronomical Data Query Language**
 - For database queries
 - Core SQL functions plus astronomy-specific extensions
 - Sky region, Xmatch
- **SkyNode**
 - Exposes relational databases
 - Accepts ADQL query
 - “Full” SkyNodes support positional cross-match function
 - OpenSkyQuery portal
 - show database structure
 - query tools

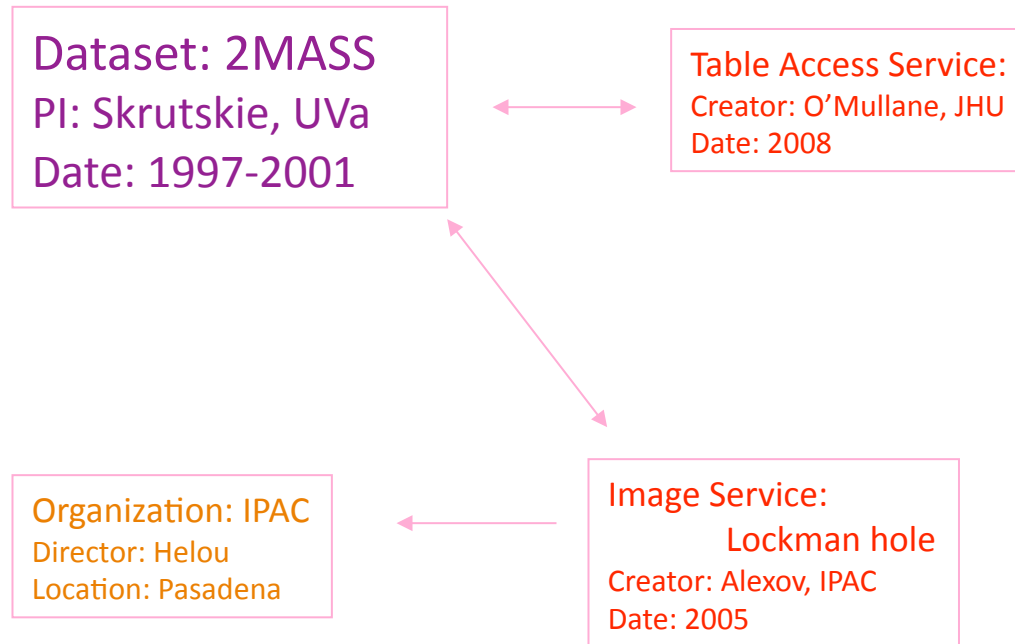
Registry

- **publish -- find -- bind**
- Registry Metadata
 - Descriptions of
 - data collections
 - data delivery services
 - organizations, etc.
 - Based on Dublin Core with astronomy-specific extensions
 - Represented as XML schema; extensible
 - Contents stored in Resource Registries
 - exchange metadata records through the Open Archives Initiative Protocol (OAI-PMH)



Registry Content:

Datasets, Standard Services, Organizations



Identifier format

`ivo://ipac.caltech/Lockman-Hole`

Registry record

Please print in type
Presidential Election Campaign (See page 21.)

Schedule R (Form 1040) **Dataset Description** OMB No. 1545-2002 Attachment Sequence No.

Department of the Treasury Internal Revenue Service (99) Attach to Form 1040. See Instructions for Schedule R (Form 1040).

Names shown on Form 1040 Your social security num

Form 5754 **Organization** Recipients of gambling winnings should see the instructions on the back of this form. Payers of gambling winnings should see the separate instructions for Forms W-2G

Department of the Treasury

SCHEDULE J (Form 1040) **Service semantics** Form Income Tax

Department of the Treasury Attach to Form 1040

Form 1040 Department of the Treasury—Internal Revenue Service **2002** (99)

U.S. Individual Income Tax Return

For the year Jan. 1-Dec. 31, 2002, or other tax year beginning 20 ending

LABOR (See instructions on page 1.) Use this label. Otherwise, please print or type your name and address. Presidential Election Campaign (See page 21.)

CURATION

Note. Checking "Yes" will not change your tax or reduce your refund. Do you, or your spouse if filing a joint return, want \$3 to go to this fund?

Machine number	
Address	
Amount received	Fed

1	Social security
2	Amount on line 1
3	Dividends, interest, capital gains, etc.
4	Other income

Distributed Registry



Ongoing harvesting July 08
(CfA, ESO, NOAO soon)

Web Service Glossary

- Web service (think a (code) library on the web)
 - A piece of software available over a network with a formal description of how it is called and what it returns that a computer can understand
- REST (think HTTP GET and POST)
 - An approach to web services that uses the existing infrastructure of the web
- SOAP (think envelopes)
 - An approach to web services that uses an XML-based messaging framework
- WSDL (think a contract)
 - An XML description of a web service (normally SOAP) and how to interact with it

What we won't cover

- Security:
 - I want to protect my data/resources
- Attachments
 - I want to upload/download a file
- State
 - I want the service to remember things
- Asynchrony
 - It's going to take some time
- Messaging
 - I want more than just request/response

Web services in the VO

- SkyNodes, Open SkyQuery and WESIX
- Footprint Services
- Spectrum Services
- Registry Interface
- VOSpace
- CasJobs
- CDS
- VOServices
- Cone search

Numquam ponenda est pluralitas sine necessitate

- REST (*Representation State Transfer*) is an architectural style not an implementation:
 - Distinguishes resources and operations
 - Each resource is identified by a URI
 - The only operations are the HTTP methods

REST by analogy

HTTP Method	Think	Description	/proc analogy
PUT	CREATE	Create a resource with the user specified id	Start a new process
GET	RETRIEVE	Retrieve a resource representation	Get the status of a process
POST	UPDATE	Update a resource/Append to a resource/Create a resource with a server assigned id	Amend a process
DELETE	DELETE	Delete a resource	Kill a process

By accident, not by design

- Parameters in HTTP GET URLs:
 - <http://processes.com/123/status>
 - <http://processes.com/services?action=getStatus&pid=123>
- HTTP GET is *safe* – it does nothing else than retrieval
- HTTP GET, PUT and DELETE are *idempotent* – the effects on the system of one or N identical requests are the same
- Consider HTTP GET with:
`http://processes.com/services/action=cancelJob&pid=123`
- Services which maintain idempotency whilst allowing parameterized URLs are **accidentally RESTful**
- 85% of web service traffic, 6x faster *allegedly*

When to use REST

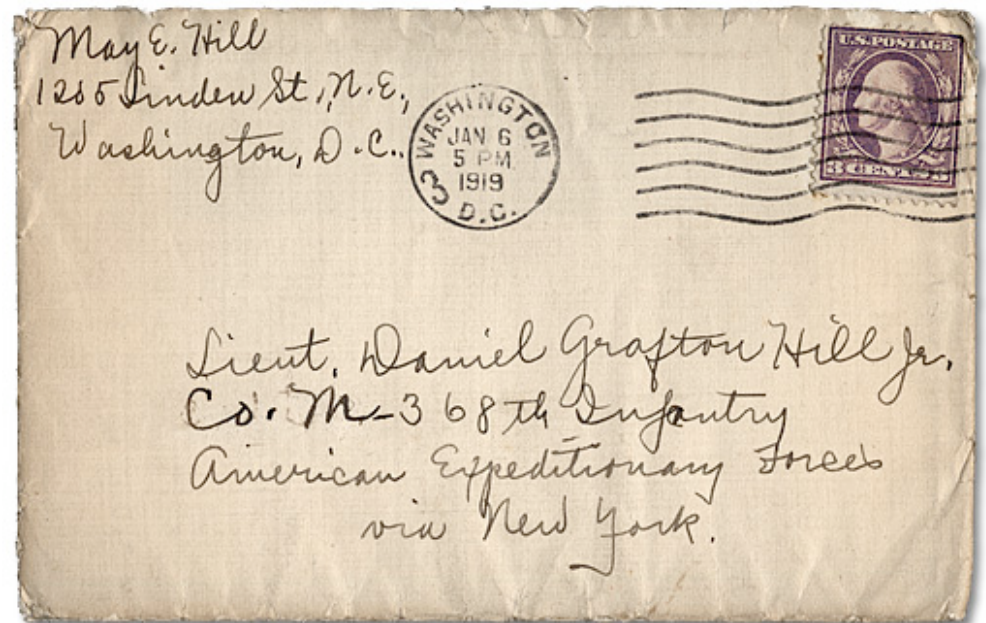
- RESTful services are good for WWW-type functionality
- Limitations:
 - No formal description:
 - WADL not widely used yet
 - Not an issue for pure REST (no parameters in URL)
 - No messaging infrastructure:
 - Reliable messaging
 - Message routing
 - No message level security:
 - Digital signatures
 - No resource lifecycle management
 - No transaction support
 - No asynchronous event notification

What is SOAP?

- Simple Object/Service-Oriented Access Protocol
 - <http://wanderingbarque.com/nonintersecting/2006/11/15/the-s-stands-for-simple/>
- W3C specification
 - <http://www.w3.org/TR/soap>
- An XML-based messaging framework for exchanging information between peers in a decentralized, distributed environment
- Defines the message structure but not the message content (needs other technologies)
- Fundamentally stateless (no memory of what has happened previously)
- One-way message exchange paradigm

Anatomy of a SOAP message

- SOAP envelope
 - SOAP header (optional)
 - SOAP body
 - Message payload (actual data)
 - XML namespace and schema



SOAP example

Request:

```
<soap:Envelope xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance xmlns:xsd=http://www.w3.org/2001/
  XMLSchema xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ComovingLineOfSight xmlns="http://skyservice.pha.jhu.edu">
      <z>float</z>
      <hubble>float</hubble>
      <omega>float</omega>
      <lambda>float</lambda>
    </ComovingLineOfSight>
  </soap:Body>
</soap:Envelope>
```

Response:

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/
  XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ComovingLineOfSightResponse xmlns=http://skyservice.pha.jhu.edu>
      <ComovingLineOfSightResult>float</ComovingLineOfSightResult>
    </ComovingLineOfSightResponse>
  </soap:Body>
</soap:Envelope>
```

What is WSDL?

- Web Services Description Language
- W3C specification
 - <http://www.w3.org/TR/wsdl>
- An XML grammar for describing the public interface of a web service in terms of:
 - its exposed operations
 - the message formats
 - protocol bindings (e.g. SOAP, HTTP)
- Most commonly used with SOAP
 - defines format of message content in the SOAP body

Anatomy of a WSDL

```
<definitions>  
  <types>  
  <message>  
  <portType>  
    <operation>  
      <input>  
      <output>  
      <fault>  
    </operation>  
  </portType>  
  <binding>  
  <service>  
</definitions>
```

A WSDL document describes:

- a set of **services** which exchange **messages**
- the data in the **messages** is defined in a set of **types**
- each message transfer in a service defines an **operation**
- **operations** have **bindings** to specific implementations using protocols like HTTP

BURGERS

Served with French Fries and a Dill Pickle

The Real McCoy 14.95

10oz hereford beef burger topped with cheddar cheese, bacon, onion rings, slice tomato & bbq sauce

10 oz Hereford Beef Burger 11.95

lettuce & tomato, add bacon, mozzarella, swiss, american, cheddar or blue cheese, \$1.00 each

Grilled Turkey Burger 11.95

lettuce, tomato & russian dressing

Grilled Vegetable Burger 10.95

lettuce, tomato & ranch dressing

SANDWICHES

Served with French Fries and a Dill Pickle

BBQ Chicken 11.95

vine ripe tomato & cilantro mayo
on grilled tuscan bread

French Dip Au Jus 12.95

thin sliced roast beef & horseradish aioli
on a french baguette

Classic Turkey Club 11.95

choice of bread (whole wheat, rye or white)

BLT 11.95

bacon, lettuce, tomato, avocado & mayo
on toasted home style white

Fresh Corned Beef on Rye 11.95

Maryland Crab Cake 13.95

lettuce, tomato & tartar sauce on a toasted bun

Blackened Mahi Mahi 13.95

lettuce, tomato, cajun remoulade & mango
salsa on open faced grilled tuscan bread

Pig 'n' Whistle Sandwich 13.95

pulled roasted pork, mozzarella cheese
& bbq sauce on a roll

PH: 212.832.2021

What is WSDL

It is the shared document that allows a request-response conversation

Using SOAP and WSDL

- Start with WSDL (*contract-first*) vs. start with code (*contract-last*)
- Frameworks provide APIs:
 - to handle SOAP:
 - *(De)serialization*: converting code object to/from XML representation
 - and code bindings derived from WSDL:
 - *Stub*: client-side proxy to service
 - *Skeleton*: server-side to handle request and response
- Alternate approach:
 - SOAP (and WSDL) are just XML
 - Use XSLT, curl, JavaScript, etc.