AN ARCHITECTURAL OVERVIEW OF HYDROSHARE, A NEXT-GENERATION HYDROLOGIC INFORMATION SYSTEM

JEFF HEARD (1,2), DAVID TARBOTON (3), RAY IDASZAK (1,2), JEFF HORSBURGH (3), DAN AMES (4), ALEX BEDIG (5), ANTHONY M. CASTRONOVA (3), ALVA COUCH (5), PABITRA DASH (3), CUYLER FRISBY (4), TIAN GAN (3), JON GOODALL (6), STEPHEN JACKSON (7), SHAUN LIVINGSTON (4), DAVID MAIDMENT (7), NICK MARTIN (4), BRIAN MILES (2), STEPHANIE MILLS (1), JEFF SADLER (4), DAVID VALENTINE (8), LAN ZHAO (9)

(1): Renaissance Computing Institute (RENCI), Chapel Hill, NC 27517, USA
(2): University of North Carolina at Chapel Hill, Chapel Hill, NC 27599, USA
(3): Civil and Environmental Engineering, Utah State University, Logan, UT 84322, USA
(4): Civil and Environmental Engineering, Brigham Young University, Provo, Utah 846, USA
(5): Computer Science, Tufts University, Medford, MA 02155, USA
(6): Civil and Environmental Engineering, University of Virginia, Charlottesville, VA 22904, USA
(7): Civil, Architectural and Environmental Engineering, University of Texas at Austin, Austin, Texas 78712, USA
(8): San Diego Supercomputer Center, University of California at San Diego, 10100 Hopkins Drive, La Jolla, CA 92093, USA
(9): School of Civil Engineering, Purdue University, West Lafayette, IN 47907, USA

HydroShare is an online, open-source, collaborative system being developed for sharing hydrologic data and models as part of the NSF's Software Infrastructure for Sustained Innovation (SI2) program. The goal of HydroShare is to enable scientists to easily discover and access hydrologic data and models, retrieve them to their desktop, or perform analyses in a distributed computing environment that may include grid, cloud, or high performance computing. Scientists may also publish outcomes (data, results or models) into HydroShare, using the system as a collaboration platform for sharing data, models, and analyses. HydroShare involves a large distributed software development effort requiring collaboration between domain scientists, software engineers, and software developers across eight U.S. universities, RENCI, and CUAHSI. HydroShare expands the data sharing capabilities of the Hydrologic Information System of the Consortium of Universities for the Advancement of Hydrologic Sciences, Inc. (CUAHSI): It broadens the classes of data accommodated, enables sharing of models and model components, and leverages social media functionality to enhance collaboration around hydrologic data and models. The HydroShare architecture is a stack of storage and computation, web services, and user applications. A content management system, Django+Mezzanine, provides user interface, search, social media functions, and services. A geospatial visualization and analysis component enables searching, visualizing, and analyzing geographic datasets. A web browser is the main interface to HydroShare, however a web services applications programming interface (API) supports access through HydroDesktop and other hydrologic modeling systems, and the architecture separates the interface layer and services layer exposing all functionality through these web services. This presentation will describe key components of HydroShare and discuss how HydroShare is designedto enable better hydrologic science concomitant with sustainable open-source software practices.

INTRODUCTION

HydroShare is an online collaborative system being developed for open sharing of hydrologic data and models [17]. It is being developed by an interdisciplinary team of domain scientists, university software developers, and professional software engineers that involves eight U.S. universities, RENCI, and CUAHSI. The goal of HydroShare is to enable scientists to easily discover, access, and share hydrologic data and models and perform analyses in a distributed computing environment that may include grid, cloud, or high performance computing model instances as necessary. Scientists may publish and share data, results, or models in HydroShare. HydroShare expands the data sharing capability of the CUAHSI Hydrologic Information System [1,5,6,16], which is focused on time series data, by supporting a broader class of data, including geospatial data used in hydrology [15]. HydroShare will take advantage of emerging social media functionality to enhance information about and collaboration around hydrologic data and models. The HydroShare web interface is the primary method of access ensuring ubiquitous access and client platform independence. A primary construct of HydroShare is a resource adhering to a Resource Data model accommodating data and models in supported or generic formats. Resources with supported formats within HydroShare are advantaged with additional analysis, visualization, and discovery tools.

COMPONENTS

HydroShare is a complex, multitier application that will serve as the next generation Hydrologic Information System. As such, it innovatively combines content management with social networking, high performance computing, and enterprise level data management to support the sharing of science data, metadata models, and processes around water and hydrology. We have centered our efforts around a number of best-of-breed Open Source components to realize the HydroShare system. These components include:

- Django – A web application platform [3]
- Django's ORM – Object Relational Mapping between Python classes
- Mezzanine – Content Management System meta-framework [10]
- Tastypie – Quickly generate RESTful APIs from ORM models
- Geoanalytics – RENCI's Geospatial processing, content management, and visualization system [4]
- iRODS – RENCI's enterprise storage management middleware [7]
- PostgreSQL with PostGIS – Enterprise class relational database with OGC standard geospatial extensions [11]
- Celery – A task queuing system capable of supporting large distributed computing jobs

Figure 1 represents the HydroShare architecture as envisioned. Gray boxes represent frameworks that are integral to HydroShare. HydroShare Resources extend the notion of a CMS page to include science data and metadata as well as deferred processes and models. Each different type of content stored is its own HydroShare Resource type. Examples of different Resource types include time series, geographic features, geographic rasters, multidimensional space-time datasets, and sample-based observations. It will also be possible to have resources that are references to data outside the system supported by an external webservice (e. g. CUAHSHI HIS WaterOneFlow web services or OGC geospatial web services), and composite

resources comprised of a collection of other resources. Basic create, read, update, and delete functionality as well as metadata editing functionality for all HydroShare resources is exposed through the HydroShare API. Any HydroShare resource type may also have additional API functions that offer additional resource-specific functionality such as subsetting, processing, and manipulation.

Science data and science metadata are housed in iRODS. This metadata includes Dublin Core data as well as extensions to support the cataloging of HydroShare resources and type-specific metadata that must be part of the resource. System metadata is housed in PostGIS. This may include a partial replication of the science metadata as well as system-only information.

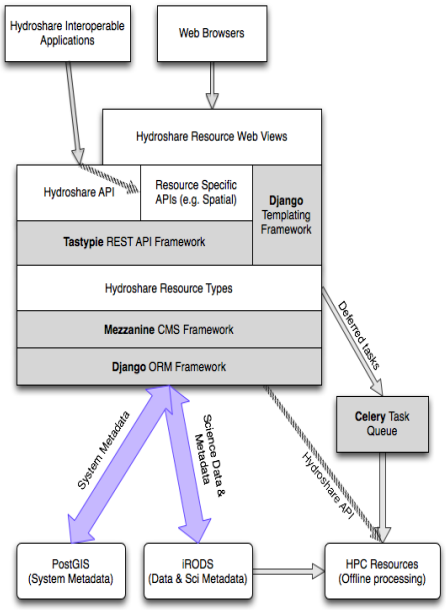

Figure 1. HydroShare architecture

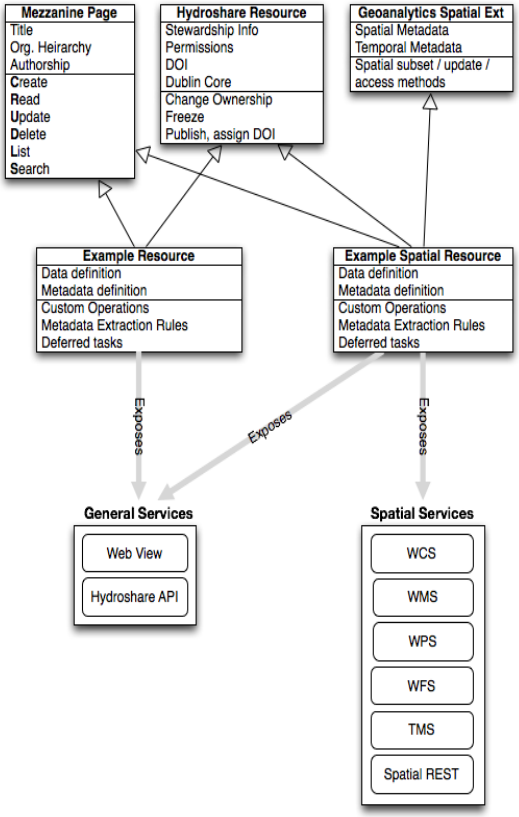

Figure 2. Relationship of resources to the ORM. This shows two example resources and the services they expose. Inheriting from HydroShare Resource automatically exposes the HydroShare API.

WEB APPLICATION FRAMEWORK

HydroShare is built as a web-service and web application. Django is a web application framework written in Python that follows the Model-View-Controller (MVC) model for

splitting responsibility in software components. HydroShare follows this model as well, with each *HydroShare Resource* type defined as one or more Models and Views with the API serving as the Controller component.

The Django framework consists broadly of the following:

- An endpoint-mapper to translate between HTTP URLs and specialized application code in Views
- An object-relational mapper (ORM)
- An extensible authentication, user profile, and permissions component
- An HTML template engine that provides quick generation of user-readable material based on data objects in the system
- An automatically generated administrator's interface

OBJECT-RELATIONAL MAPPING

Django's ORM provides a mapping between a relational database and the Python and HTML template code. The Django ORM allows a developer to create specialized Model subclasses whose members relate to columns in a database table. The Django ORM is capable of expressing standard relationships between models, such as foreign keys, one-to-one, and many-to-many relationships and can express type, indexing preferences, and database constraints that are enforced both in code and at the database level.

The Django ORM is used in HydroShare to code all *HydroShare Resource* content types. The ORM maps system-level metadata of these content types to the database. System metadata is defined as any metadata needed to display, clarify, or authorize access to resource content.

The ORM represents a standard Object-Oriented model. Figure 2 illustrates the inheritance relationships that provide HydroShare functionality to Resources. By extending the notion of a CMS page with HydroShare Resource data and Resource type specific data, one can expose a new HydroShare Resource type in the system. This is automatically available as a webview and through the HydroShare API. Other services may be exposed by writing specific services or including other packages such as Geoanalytics.

CONTENT MANAGEMENT

HydroShare entails the display, organization, discoverability, and searchability of content in the form of *HydroShare Resources.* A HydroShare Resource defines a family of content in terms of its necessary and optional data, metadata, relationships, and processes. An instance of a *HydroShare Resource* is a package of data and metadata managed as a whole. The HydroShare Resource Data Model is based on the Open Archives Initiative's Object Reuse and Exchange (OAI-ORE) standard [9], which is a standard for the description and exchange of aggregations of web resources. The OAI-ORE Abstract Data Model is described at http://www.openarchives.org/ore/1.0/datamodel.html. HydroShare will use the BagIt File Packaging Format for its resources [2]. Each *HydroShare Resource Instance* is assigned a unique, unchanging URL endpoint from which it can be accessed on the Web.

HydroShare also introduces a simple and transparent permission system defined by its stakeholders to promote the open sharing of hydrologic data while controlling access where necessary. This permission system provides functionality to manage the sharing of their HydroShare Resources, including stewardship, view and edit permissions, and the ability to

"freeze" a resource from further editing or deletion when it is formally published, and to assign a DOI (http://www.doi.org). HydroShare additionally defines versioning and relationships of *HydroShare Resource Instances* to preserve provenance and history of the data and models under its stewardship.

The Mezzanine Content Management System extends Django to provide the concept of Pages, object hierarchy, and ownership of content. It also provides facilities for visualizing that content as web pages and searching through metadata. Mezzanine does the heavy lifting of content visualization and separating interface from implementation in HydroShare's content model. HydroShare Resources are specializations of Mezzanine's Pages so as to leverage all of the in-built capabilities of Mezzanine.

Mezzanine also provides for inline editing of content and this capability is heavily used throughout HydroShare to provide a user-friendly way to let a user modify his or her HydroShare Resources.

GEOSPATIAL PROCESSING AND VISUALIZATION

Many HydroShare Resource types are envisioned to have a geospatial component. HydroShare makes use of RENCI's Geoanalytics framework, an add-in for Django and Mezzanine, to handle this component. Geoanalytics provides geospatial data, processing, and visualization services for HydroShare Resources, automatically enabling geospatial resources with:

- OGC Web Feature Service (WFS) – GIS integration, spatial, and indexed attribute-based querying and subsetting of spatial data [12]
- OGC Web Map Service (WMS) – Provides styled layers with styles written in Carto CSS [13]
- Tiled Map Service (TMS) – Provides cached styled layers similar to WMS
- JSON and JSONP rest APIs for individual spatial data records - spatial, and indexed attribute-based querying and subsetting of spatial data for Web Mapping and online processing
- OGC Web Processing Service (WPS) – for wrapping models for use with data in HydroShare Resources [14]
- Layered web map visualizations using OpenLayers and commercial or open tile APIs like Google Maps and OpenStreetMaps

DATA STORAGE ROLES OF IRODS AND POSTGIS

Data storage is a complex issue in a system with varying needs for ephemeral, indexed, and persistent storage solutions. Unlike a traditional Content Management System, HydroShare has needs both for accessing compute services and for storing very large datasets on systems where persistence and data quality are assured. We have chosen therefore a suite of systems whose roles we outline in this section.

iRODS

iRODS is an enterprise storage management middleware that provides "grid-storage" [7]. iRODS has fine-grained customizability to support metadata and rules that perform action based on metadata and access. It provides complete control over storage deployment and behavior for large and complex storage systems, and thus it is a perfect system for providing

storage for large, persistent files like those contained in HydroShare Resources. The science data and metadata components of a HydroShare Resource are stored within the iRODS system.

PostGIS

Postgresql (with PostGIS extensions) provides relational storage for system-level data and metadata. System-level data and metadata deal with content management, indexing of data for search, users, authentication, authorization, and other issues needed to make the HydroShare system perform fluidly.

DISTRIBUTED COMPUTING

Certain long-running tasks in HydroShare, such as indexing, data summarization, and other resource-specific tasks may cause the main system to experience heavy loads. These heavy loads are offloaded to ad-hoc external processing via the Celery Task Queue. The Celery Task Queue supports gang-style computing where processors join and leave processing queues at will and resources can be rapidly marshaled to handle a sudden influx of computing tasks.

This approach is especially appropriate for systems such as HydroShare that will receive infrequent but large updates to certain data sources that may require significant processing for their visualization and metadata indexing components. These tasks can be offloaded so that web server performance does not suffer. In addition, the HydroShare Geoanalytics component makes heavy use of Celery in the WFS, WMS, and WCS components.

USER INTERFACES

Website

The beta website interface to HydroShare can be accessed at http://beta.HydroShare.org[1]. The web interface enables the user to search, browse, create, read, update, delete, and list HydroShare Resource Instances within the system.

Application Programmer Interface (API)

The HydroShare API is a developer interface to the HydroShare system paralleling the website interface. The intent of the HydroShare API is to provide a generic API capable of performing common tasks on all HydroShare Resource instances regardless of the HydroShare Resource content type. This API will allow applications to create, read, update, browse, share, delete, search, and list HydroShare Resource Instances within other applications such as specialized GIS systems, HydroDesktop, and other applications.

HydroShare Resource APIs specialized to a particular HydroShare Resource type may also exist and will follow similar patterns to the main HydroShare Resource API. These APIs focus on subsetting and analysis of data contained within a single HydroShare Resource Instance or to compare data inside one or more HydroShare Resource Instances.

---

[1] HydroShare is in development at the time of this writing. The production version of HydroShare will be available at http://www.HydroShare.org/.

SOFTWARE DEVELOPMENT PROCESS

HydroShare is a collaborative effort between eight universities. Because this collaboration is large-scale it bears noting that we follow an Agile methodology to develop HydroShare, heavily focused on autonomous development and integration of Open Source components. The software frameworks and developer tools we have chosen, such as Django, Mezzanine, and iRODS with development environments managed via GitHub and Docker, and a process managed through Pivotal Tracker, support this wide-area collaborative software development process.

CONCLUSION

The architecture presented has been designed to provide a community collaboration site that enables users to easily discover and access data and models, retrieve them to a desktop computer, or perform analyses in a distributed computing environment that includes grid, cloud, or high performance computing model instances as necessary. While element of this functionality are available through other general purpose collaboration, file sharing systems, social interaction, and web publication systems (e.g. Google Drive, Drop Box, Figshare, YouTube, Facebook) HydroShare development is focused on integrating hydrology domain value added functionality that gathers best practice tools into a collaboration system for use targeted specifically at the hydrology community. We envision that HydroShare will enable more rapid advances in hydrologic understanding through the collaborative data sharing, analysis, and modeling that it enables. Understanding will be advanced through the ability to integrate information from multiple sources. The provenance and metadata that HydroShare maintains will enhance reproducibility and transparency of the research conducted using HydroShare resources. While focused primarily on the hydrology community, it is a goal that the capabilities developed will have value and be extensible to other subject areas.

ACKNOWLEDGEMENTS

This work was supported by the National Science Foundation under collaborative grants OCI-1148453 and OCI-1148090. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Ames, D. P., J. S. Horsburgh, Y. Cao, J. Kadlec, T. Whiteaker and D. Valentine, "HydroDesktop: Web services-based software for hydrologic data discovery, download, visualization, and analysis," *Environmental Modelling & Software*, Vol. 37, (2012), pp. 146-156, http://dx.doi.org/10.1016/j.envsoft.2012.03.013.

[2] Boyko, A., J. Kunze, J. Littman, L. Madden and B. Vargas, "The BagIt file packaging format (v0.97) network working group internet draft", (2012), http://tools.ietf.org/html/draft-kunze-bagit-10, accessed 2/6/2014.

[3] Django, "Django - A high-level Python Web framework that encourages rapid development and clean, pragmatic design", (2014), https://www.djangoproject.com/, accessed 3/20/2014.

[4] Heard, J. R. "Technical report TR-11-03, Geoanalytics", RENCI, (2011), http://www.renci.org/technical-reports/tr-11-03/.

[5] Horsburgh, J. S., D. G. Tarboton, D. R. Maidment and I. Zaslavsky, "A relational model for environmental and water resources data," *Water Resour. Res*., Vol. 44, (2008), W05406, http://dx.doi.org/10.1029/2007WR006392.

[6] Horsburgh, J. S., D. G. Tarboton, K. A. T. Schreuders, D. R. Maidment, I. Zaslavsky and D. Valentine, "Hydroserver: A platform for publishing space-time hydrologic datasets," *2010 AWRA Spring Specialty Conference Geographic Information Systems (GIS) and Water Resources VI,* Orlando Florida, *American Water Resources Association*, Middleburg, Virginia, (2010), TPS-10-1, http://www.awra.org/tools/members/Proceedings/1003conference/doc/abs/JefferyHorsburgh_7cb420e3_6602.pdf.

[7] IRODS, "IRODS: Data grids, digital libraries, persistent archives, and real-time data systems", (2010), https://www.irods.org/, accessed 6/6/10.

[8] Kunze, J. and Baker, T., "The dublin core metadata element set", *Dublin Core Metadata Initiative,* (2007), http://tools.ietf.org/html/rfc5013.

[9] Lagoze, C., H. Van de Sompel, P. Johnston, M. Nelson, R. Sanderson and S. Warner, *Open archives initiative object reuse and exchange: ORE user guide – primer,* (2008), http://www.openarchives.org/ore/1.0/primer, accessed 10/23/2012.

[10] McDonald, S. *et al*., "Mezzanine", (2014), http://mezzanine.jupo.org/.

[11] "OpenGIS Simple Features Specification for SQL", Open GIS Consortium, Inc., (2009), http://www.opengeospatial.org/standards/is.

[12] "OpenGIS Web Feature Service 2.0 Interface Standard", Open GIS Consortium, Inc., v.2.0 (2010), http://www.opengeospatial.org/standards/wfs.

[13] "OpenGIS Web Map Service (WMS) Implementation Specification", Open GIS Consortium, Inc., v.1.3.0 (2006), http://www.opengeospatial.org/standards/wms.

[14] "OpenGIS Web Processing Service", Open GIS Consortium, Inc., v.1.0, (2007), http://www.opengeospatial.org/standards/wps.

[15] Salas, F. R., E. Boldrini, D. R. Maidment, S. Nativi and B. Domenico, "Crossing the digital divide: an interoperable solution for sharing time series and coverages in Earth sciences," Nat. Hazards Earth Syst. Sci., Vol. 12, No. 10, (2012), pp. 3013-3029, http://dx.doi.org/10.5194/nhess-12-3013-2012.

[16] Tarboton, D. G., J. S. Horsburgh, D. R. Maidment, T. Whiteaker, I. Zaslavsky, M. Piasecki, J. Goodall, D. Valentine and T. Whitenack, "Development of a community hydrologic information system," *18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation,* ed. R. S. Anderssen, R. D. Braddock and L. T. H. Newham, *Modelling and Simulation Society of Australia and New Zealand* and *International Association for Mathematics and Computers in Simulation*, (2009), pp, 988-994, http://www.mssanz.org.au/modsim09/C4/tarboton_C4.pdf.

[17] Tarboton, D. G., R. Idaszak, J. S. Horsburgh, J. Heard, D. Ames, J. L. Goodall, L. Band, V. Merwade, A. Couch, J. Arrigo, R. Hooper, D. Valentine and D. Maidment, "HydroShare: Advancing collaboration through hydrologic data and model sharing," *7th International Congress on Environmental Modelling and Software*, ed. D. Ames and N. Quinn, San Diego, California, USA, *International Environmental Modelling and Software Society (iEMSs)*, (submitted 2014).