






Article

Water Data Explorer: An Open-Source Web Application and Python Library for Water Resources Data Discovery

Giovanni Romero Bustamante ¹, Everett James Nelson ^{1,*}, Daniel P. Ames ¹, Gustavious P. Williams ¹, Norman L. Jones ¹, Enrico Boldrini ², Igor Chernov ³ and Jorge Luis Sanchez Lozano ¹

¹ Department of Civil and Environmental Engineering, Brigham Young University, Provo, UT 84602, USA; gio.busrom@gmail.com (G.R.B.); dan.ames@byu.edu (D.P.A.); gus.p.williams@byu.edu (G.P.W.); njones@byu.edu (N.L.J.); jorgessanchez7@gmail.com (J.L.S.L.)

² National Research Council of Italy (CNR), Institute of Atmospheric Pollution Research (IIA), 10, 00015 Sesto Fiorentino, FI, Italy; enrico.boldrini@cnr.it

³ World Meteorological Organization (WMO), CH-1211 Geneva, Switzerland; ichernov@wmo.int

* Correspondence: jimn@byu.edu; Tel.: +801-422-7632

Abstract: We present the design and development of an open-source web application called Water Data Explorer (WDE), designed to retrieve water resources observation and model data from data catalogs that follow the WaterOneFlow and WaterML Service-Oriented Architecture standards. WDE is a fully customizable web application built using the Tethys Platform development environment. As it is open source, it can be deployed on the web servers of international government agencies, non-governmental organizations, research teams, and others. Water Data Explorer provides uniform access to international data catalogs, such as the Consortium of Universities for the Advancement of Hydrologic Science (CUAHSI) Hydrologic Information System (HIS) and the World Meteorological Organization (WMO) Hydrological Observing System (WHOS), as well as to local data catalogs that support the WaterOneFlow and WaterML standards. WDE supports data discovery, visualization, downloading, and basic data interpolation. It can be customized for different regions by modifying the user interface (i.e., localization), as well as by including pre-defined data catalogs and data sources. Access to WDE functionality is provided by a new open-source Python package called “Pywaterml” which provides programmable access to WDE methods to discover, visualize, download, and interpolate data. We present two case studies that access the CUAHSI HIS and WHOS catalogs and demonstrate regional customization, data discovery from WaterOneFlow web services, data visualization of time series observations, and data downloading.



Citation: Bustamante, G.R.; Nelson, E.J.; Ames, D.P.; Williams, G.P.; Jones, N.L.; Boldrini, E.; Chernov, I.; Sanchez Lozano, J.L. Water Data Explorer: An Open-Source Web Application and Python Library for Water Resources Data Discovery. *Water* **2021**, *13*, 1850. <https://doi.org/10.3390/w13131850>

Academic Editor: Sajjad Ahmad

Received: 5 May 2021

Accepted: 28 June 2021

Published: 2 July 2021

Keywords: observation networks; WHOS; CUAHSI; Tethys; HydroShare; HydroServer

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Monitoring water quantity and quality using data from observation networks is fundamental to water resources study, management, and decision-making. Observations can include different variables related to water quantity, such as precipitation, streamflow, water depth, and those related to water quality such as temperature, turbidity, and the concentration of phosphorous, nitrogen, and other chemical components. There are challenges in accessing and using this information data because of consistency, archiving, and accessibility of the water data stored in various systems [1].

Data archiving and accessibility are challenging because it is often difficult to locate, obtain, and compare data between different regions, especially when different parties and agencies collect, store, and manage data in different formats. Data accessibility can be a challenge due to political, economic, and cultural barriers among the different data providers, or from the requirements and standards from different organizations. Dissemination of water data is complicated because of the lack of integration and interoperability

across various data archives within geosciences. All of these issues make it difficult to find and access water data from various archives [2,3].

The use of service-oriented architecture (SOA) design patterns facilitates data archiving and dissemination across multiple users and institutions. Data archiving generally involves storing data in relational databases that can be updated and accessed by different users. Similarly, dissemination of hydrological data from different countries at a transboundary level, or even from different agencies within a country, requires the standardization of various aspects of data storage and dissemination technologies. Discovery and access to data is crucial to provide the hydrological information required for the sustainable development of nationally and internationally shared water resources [4]. The technology to provide hydrological data sharing between organizations requires a cyberinfrastructure that provides data interoperability between data systems from the different organizations. An appropriate cyberinfrastructure lowers the technical barriers to data sharing and dissemination and provides organizations with the tools necessary to share their data once the other barriers (e.g., political, economic, etc.) have been resolved.

A number of different SOA cyberinfrastructures have been developed to share and store spatial discrete observation data, including, for example, the Consortium of Universities for the Advancement of Hydrologic Science (CUAHSI) Hydrologic Information System (HIS) [5], the World Meteorological Organization (WMO) Hydrological Observing System (WHOS) [6], the Critical Zone Observatory Integrated Data Management System (CZO-Data) [7], the Integrated Earth Data Applications (IEDA) and EarthChem system [8–10], and the Integrated Ocean Observing System (IOOS) [11]. The architectures of these systems differ in terms of databases, software, and hardware; however, they all are designed to enhance access to water resources observation and model data, and to facilitate the dissemination and sharing of these data.

The CUAHSI HIS project [12–14] was an early leader in the area of water resources SOA cyberinfrastructure. CUAHSI established: a system with a relational database schema called the Observations Data Model (ODM) [15,16]; data servers called HydroServers [17–19]; client tools for accessing data from these servers, including HydroDesktop [20], WaterML R [21], and HydroClient [22]; and a central catalog to find data called HIS Central, which stores searchable metadata and supports data discovery services [23]. The CUAHSI HIS uses a community-controlled shared vocabulary for hydrologic terms [24] and defined formal protocols for communication between system components, including WaterOneFlow web services and the WaterML data transfer encoding protocol [25,26]. The CUAHSI HIS includes, at present, 97 registered data servers sharing time series data for 1,222,585 sites around the globe and a total of 10,353,663,916 data values [27]. CUAHSI HIS protocols have been implemented by other data providers and software systems including the WHOS data broker [28]. The CUAHSI HIS project [5,12,13] was an early leader in the area of water resources SOA cyberinfrastructure. CUAHSI established a system with a relational database schema called the Observations Data Model (ODM) [14,15]; data servers called HydroServers [16–18]; client tools for accessing data from these servers, including HydroDesktop [19], WaterML R [20], and HydroClient [21]; and a central catalog to find data called HIS Central, which stores searchable metadata and supports data discovery services [22]. The CUAHSI HIS uses a community-controlled shared vocabulary for hydrologic terms [23] and defined formal protocols for communication between system components, including WaterOneFlow web services and the WaterML data transfer encoding protocol [24,25]. The CUAHSI HIS includes, at present, 97 registered data servers sharing time series data for 1,222,585 sites around the globe and a total of 10,353,663,916 data values [26]. CUAHSI HIS protocols have been implemented by other data providers and software systems including the WHOS data broker [27].

WHOS is a services-oriented framework linking hydrological data and users through an information system that provides data registration, data discovery, and data access [29,30]. WHOS supports the publication of customized data subsets using the concept of “views” to provide data capabilities to various organizations and users [31].

For example, a “basin view” can contain all the data sets that are collected and shared by neighboring countries in a specific basin. Client components built by others can access the WHOS views for data discovery, download and visualization using WaterOneFlow web services. WHOS is a services-oriented framework linking hydrological data and users through an information system that provides data registration, data discovery, and data access [28,29]. WHOS supports the publication of customized data subsets using the concept of “views” to provide data capabilities to various organizations and users [30]. For example, a “basin view” can contain all the data sets in a specific basin that are collected and shared by neighboring countries. Client components built by others can access the WHOS views for data discovery, download and visualization using WaterOneFlow web services.

To our knowledge, there is not an open-source client interface tool that can be customized for specific regions and datasets that accesses data from systems using WaterOneFlow web services such as those provided by CUAHSI HIS and WHOS systems. We designed Water Data Explorer (WDE) to be an easy-to-use interface to these distributed open data systems. WDE can be configured for specific regions or datasets, providing users and managers with a focused interface to their region.

One of the challenges encountered when developing an open-source client interface is the heterogeneity between the data responses of the WaterOneFlow web service methods because responses can contain different attributes. To manage this issue, WDE addresses data response heterogeneity by using the standard WaterML 1.0 response attributes, which provide a standard subset of attributes. Our main design goal for the WDE, which is described in this paper, is to serve as a client component for SOA systems using WaterOneFlow web services that can be customized for a region of interest while allowing data discovery, download, and basic analysis. We use both CUAHSI HIS and WHOS systems as examples to demonstrate the WDE development and capabilities. We selected the CUAHSI and WHOS systems because both systems use WaterOneFlow web services, and they are the main systems used globally to archive and distribute hydrology data.

The remainder of this paper is organized as follows: Section 2 presents the design and architecture of WDE; Section 3 presents a case study using WDE to access the CUAHSI HIS and WHOS systems; in Section 4 we discuss the WDE application and present some conclusions regarding the work; and we include supplementary information about how to access the software source code and data used in the case studies.

2. Software Design and Architecture

2.1. Pywaterml Python Package

We developed Pywaterml, a Python package, as part of this work. Pywaterml is a library that connects to the different WaterOneFlow web services and retrieves time series data from systems that use the WaterML 1.0 standard. Pywaterml provides the core functions of WDE and allows others to access these functions through Application Programming Interfaces (API) for use in other applications. Pywaterml is free and open-source and is available for download and installation in any Python environment using the PyPI (Python Package Index) or Conda package management systems.

Figure 1 diagrams how Pywaterml executes WaterOneFlow web services methods, receives responses, and formats the data in JavaScript Object Notation (JSON), WaterML 1.0, or comma-separated values (CSV) file formats. First, Pywaterml connects with the selected WaterOneFlow web service using a Simple Object Access Protocol (SOAP)-based web service client. Second, Pywaterml requests and retrieves different data types using the standard WaterOneFlow web service methods. WaterOneFlow has six different methods for different types of data, such as GetSites for obtaining a list of sites with data, and GetSitesInfo for obtaining the metadata, or description, of a site. Pywaterml has five methods to analyze the retrieved data: two customized methods that extend the standard WaterOneFlow methods; and three additional analysis methods. These are described later in the paper. Third, Pywaterml formats the data obtained from the different methods using any of three data format standards (JSON, WaterML 1.0, CSV). Once the data have been

retrieved and formatted, they can be used for data discovery, download, or visualization in the client component. WDE has capabilities for these tasks, or someone using the API can use custom tools for these tasks.

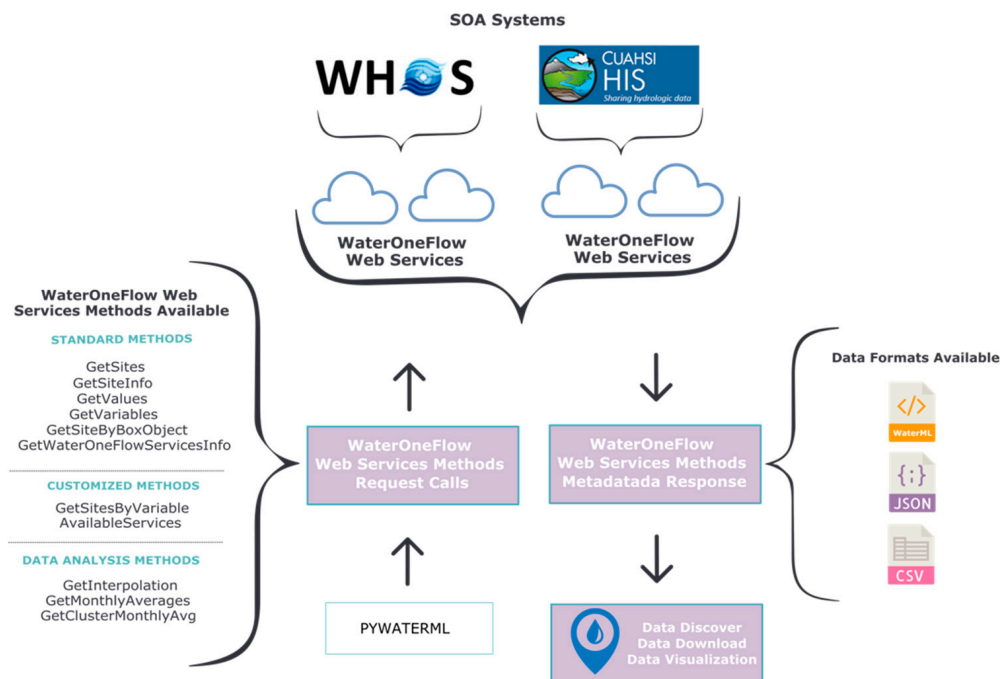


Figure 1. Pywaterml Package Functionality.

2.2. Tethys Application Framework

We developed WDE using the Tethys Platform framework [32–34]. The Tethys Platform consists of three major components for developing and deploying web applications for spatial time series data: Tethys Software Suite, Tethys Software Development Kit (SDK), and Tethys Portal. Tethys Software Suite includes file dataset management, user account management, spatial database storage, geoprocessing, mapping and visualization, and distributed computing functions. Tethys SDK provides APIs to access the tools in the software suite. The Tethys SDK provides the tools to customize WDE creating custom settings and persistent storage for a selected area or region. Tethys Portal allows users to install a generic version of WDE and customize it through the web user interface. The SDK allows more customization, but is more complex to use, while Tethys Portal allows WDE to be easily modified to focus on a specific region or data sources.

WDE can be modified to customize the name of the application displayed in the user interface, to add a Web Mapping Service (WMS) vector layer representing a regional boundary in which the observation sites will lie, and to assign a database to download and store data retrieved from the SOA systems. This allows users to create a version of WDE for their area with a specific name and boundary or mapping information. This means that the WDE instance appears as a unique data interface. For example, administrator users can customize the WDE user interface for a national organization by providing the organization name, a WMS layer that serves as a polygon boundary that defines the area of interest containing the observation sites, and a database to store the metadata from the different WaterOneFlow web services

2.3. WDE Organization

WDE uses catalogs to organize and manage the different WaterOneFlow web services. WDE can create catalogs from existing WaterOneFlow web services. WaterOneFlow web services can contain other WaterOneFlow web services that are catalogs of other WaterOne-

Flow web services, describing a network of WaterOneFlow servers. These WaterOneFlow catalogs provide metadata for other services, such as the HIS Central Registry, or a WHOS view. Not all WaterOneFlow web services provide web service catalog functionality. WaterOneFlow web services can provide either a catalog of different data servers or the data server itself. We designed the WDE structure to manage three different levels: catalogs, servers, and sites. The bottom level is the site, which is a representation of an observation site which contains both metadata describing the site and observation data. At the next level, a server represents a collection of sites, and at the top level a catalog represents a collection of servers.

WDE uses different WaterOneFlow web services methods to retrieve metadata for the different levels. Figure 2 depicts the different WaterOneFlow web services methods used to retrieve data at each WDE level and shows the databases WDE uses to store the downloaded metadata from the responses of these methods.

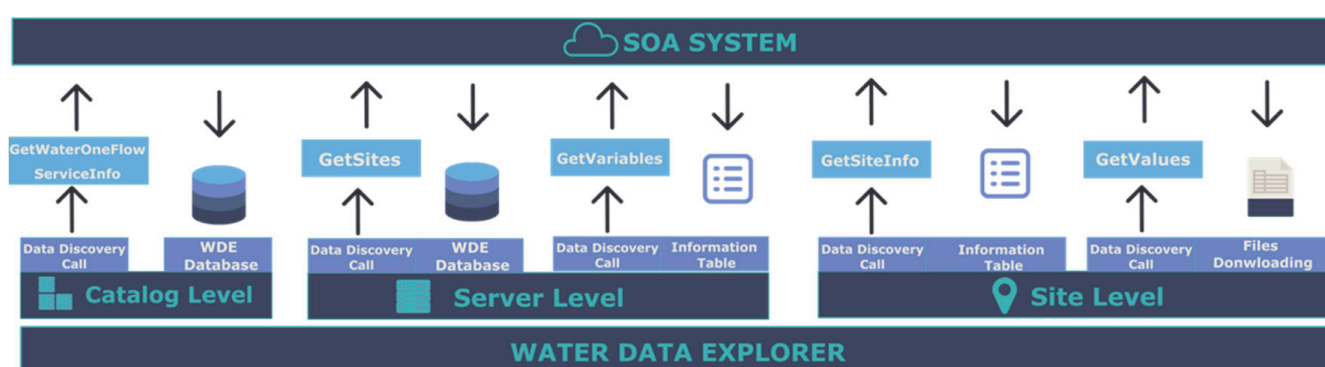


Figure 2. WDE Structure Level.

WDE uses Pywaterml to access the web services methods at each level with Pywaterml methods with the same name as the WaterOneFlow method.

- The catalog level retrieves information using the `GetWaterOneFlowServiceInfo` WaterOneFlow method, then stores the metadata it receives in the WDE local database catalog table.
- The server level uses the methods `GetSites` and `GetVariables` to retrieve data and to store the metadata it receives in the server table in the WDE local database.
- The site level retrieves metadata using two methods: `GetSiteInfo` and `GetValues`, but it does not store the retrieved metadata in the local database. Instead, it downloads the content it receives to local storage.

WDE stores the catalog and WaterOneFlow server level responses to avoid multiple network requests that would be required to load the metadata at WDE startup. For example, geospatial visualization of the different sites requires calling the `GetSites` method for each WaterOneFlow service contained in each catalog of the application. However, having the response saved in the local WDE database reduces the loading time and removes the need to request remote data every time WDE starts. By contrast, WDE does not store metadata at the site level because it would require downloading metadata for each site. Generally, queries are made for a relatively small number of sites and variables and do not require storing the entire database locally. Consequently, every time there is a request for metadata related to a specific site or time series observation values, WDE makes a new request using Pywaterml to download the data.

2.4. Data Discovery

2.4.1. Data Discovery Overview

WDE uses two types of data discovery: 1) across all the WaterOneFlow web services that have been registered to any WaterOneFlow catalog such as HIS Central, or 2) within a

single WaterOneFlow web service that has not been registered to a WaterOneFlow catalog. The first data discovery type is managed at the WDE catalog level. It provides a complete discovery of the catalog metadata and can access any of the WaterOneFlow web services methods associated with the catalog. The second data discovery type is managed at the WDE server level and makes discovery calls to WaterOneFlow web services that do not act as a catalog. For the CUAHSI HIS system, both types of discovery are available and allow WDE to access datasets that are documented at the HIS Central catalog, or it can access databases that are stored in individual or regional HydroServers. The WHOS system also supports both types of discovery and allows WDE to access customized datasets ("views") from the WHOS broker.

2.4.2. Catalog-Level Data Discovery

Data discovery at the WDE catalog level can be performed two different ways: (1) general discovery, and (2) country-based discovery. General discovery at the catalog level accesses web services that are registered in a WaterOneFlow web services catalog. Country-based discovery restricts the discovery to the web services within a selected region. The country-based discovery uses latitude/longitude polygons that define the selected countries or region. During discovery, each site is filtered by to determine if the site lies within the polygons. WDE performs country-based discovery on the local WDE database, which is different to general discovery, which uses Pywaterml to access the WaterOneFlow web services methods on the SOA systems.

2.4.3. Server-Level Data Discovery

The WDE Server level has two different methods: (1) general discovery, and (2) variable discovery for the WaterOneFlow web services represented as servers. The general discovery method uses Pywaterml to discover new WaterOneFlow web services in SOA systems that can either be part of a catalog or separate, making it possible to expand the number of servers in the catalog. Variable data discovery uses Pywaterml to operate on different WaterOneFlow servers. The general discovery procedure at the server level is similar to the catalog-level discovery. It requests information using WaterOneFlow web services methods and stores the retrieved metadata in the local WDE database. Variable data discovery does not save the retrieved metadata in the WDE database, but instead presents the data as an information table in WDE.

2.4.4. Site-Level Data Discovery

WDE site-level discovery has two different methods: (1) general, and (2) time series discovery for each site on the WaterOneFlow servers. General discovery retrieves the metadata for a site, such as: site name, supervising organization, and observed variables using the GetSitesInfo method. The general discovery procedure at the site level is different to the one performed at the catalog and server levels because it does not store the metadata in the local WDE database, but provides the data as a file download containing the site metadata.

2.4.5. Metadata Harvesting for Caching Purposes

WDE uses a PostgreSQL database that is part of Tethys to store the metadata received from general data discovery at the WDE catalog and server levels. Data from the variable data discovery at the server level and the general and time series data discovery at the site level are not stored locally but are presented in WDE or are available for download.

General discovery at the WDE Catalog level stores the following metadata: (i) name, (ii) description, and (iii) URL of the WaterOneFlow web services. The metadata stored from the general discovery consists of: (i) site names, (ii) site codes, (iii) site geospatial locations, and (iv) site network. WDE stores a local copy of these metadata in the Tethys PostgreSQL database. These data are stored in one database using two tables: one for different WDE catalogs, and the other for servers, as shown in Figure 3. Storing data in the local database allows WDE to quickly access and present data that have already been discovered so

that discovery does not occur each time WDE is run, which can be time consuming and place undue loads on the servers. For example, for the CUAHSI HIS system, the Server table contains all the metadata for each site on the server which is retrieved using the GetSites method. For the WHOS system, the Server table contains the metadata for each site associated with any from the different WHOS custom views. Similarly, the Catalog table contains metadata for each catalog retrieved using the GetWaterOneFlowServicesInfo method to a HIS Central catalog (CUAHSI HIS) or a WHOS customized view.

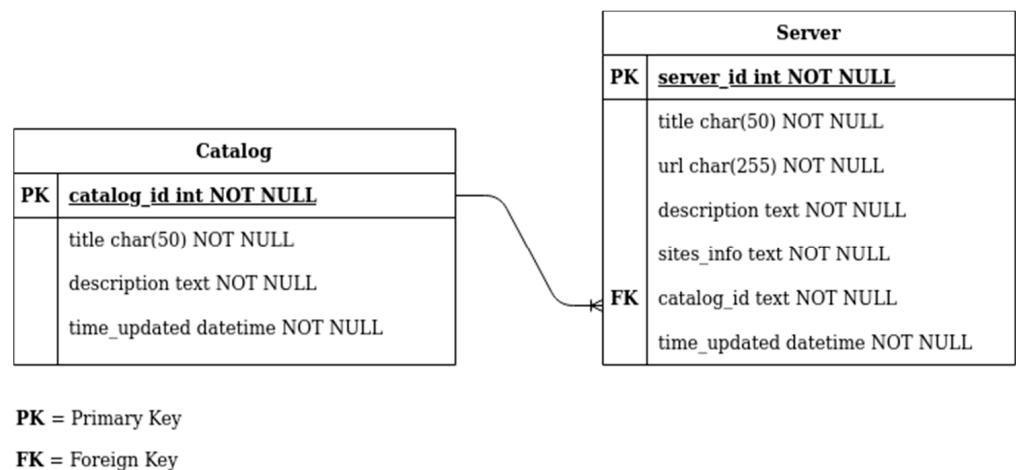


Figure 3. WDE PostgreSQL Schema for the catalog and server tables.

2.5. Data Download

Time series data discovery finds and retrieves observations of the different variables for a selected site using the GetValues method. Similar to the general discovery at the site level, the time series observation values are not stored in the local WDE database but can be downloaded to a file in one of three formats: CSV, NetCDF, or XML (WaterML 1.0. and WaterML 2.0.). CSV and WaterML 1.0. formats are provided by the Pywaterml package because they are common formats for data exportation. WaterML 2.0. and NetCDF formats are supported because they are standardized data file formats used for water data. These formats are used internally by WDE. WDE does use data in JSON format internally, but this format it is fully supported by Pywaterml for data exportation.

2.6. Data Visualization

WDE includes tools for visualizing geospatial site information and time series data. The WDE User Interface (UI), shown in Figure 4, includes a map on the right-hand side which displays the discovered sites. The site information and time series data visualization for a selected site are displayed in the lower portion of the map. This includes both the metadata to describe the site and the data and time series plots for exploration.

The WDE map view provides visualizations for the following information: (i) site name, (ii) territory of origin, (iii) supervising organization, and (iv) geospatial location (latitude and longitude). WDE presents a table with the following fields: (i) observed variables, (ii) units, and (iii) temporal extent. The site information is displayed as soon as the metadata are retrieved. In the data view, WDE displays a time series plot after the user chooses the variable of interest and selects a site. After the site and variable are selected, WDE requests the data and, after receiving it using the GetValues method, produces the time series data. WDE uses the Plotly JavaScript library [34] to implement the time series data visualization with options for both time-series and box-and-whisker plots.

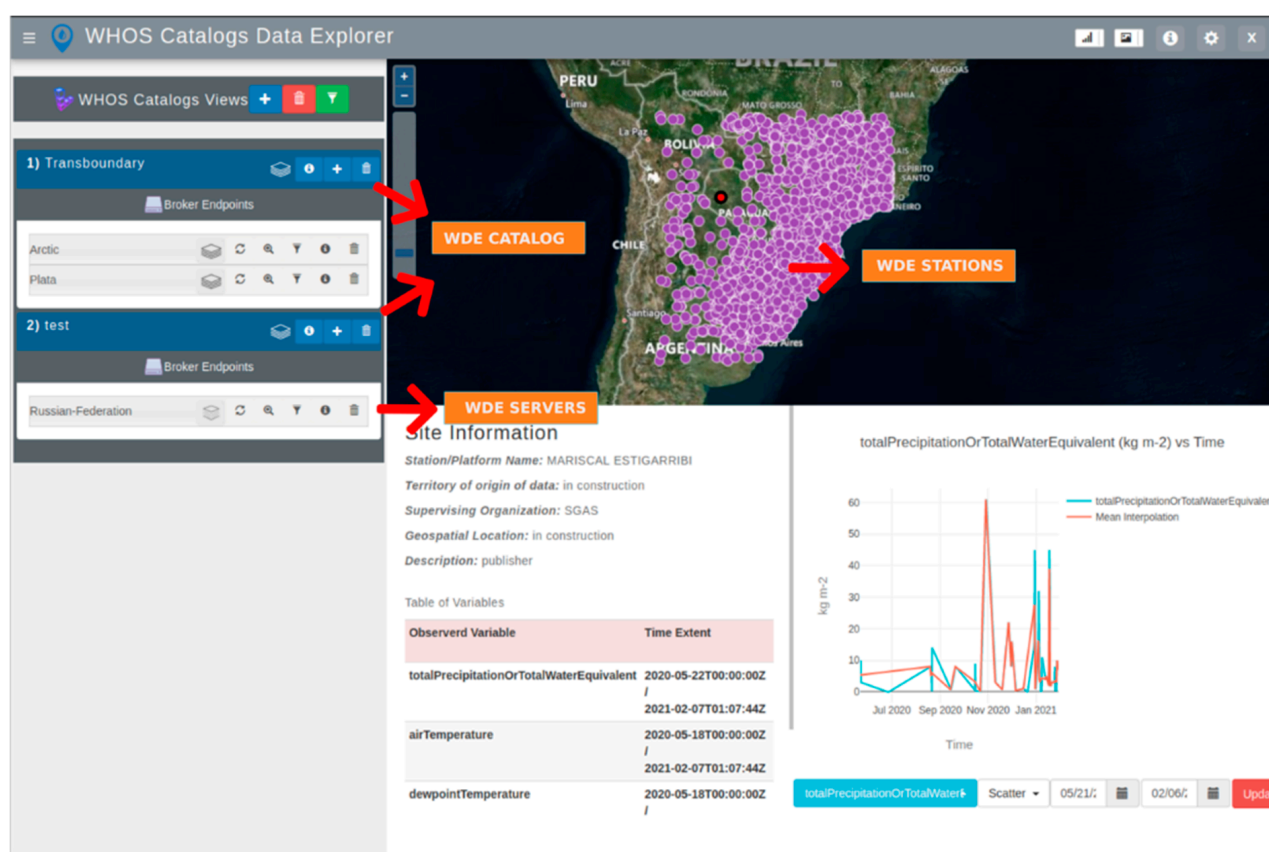


Figure 4. WDE User Interface.

3. Results

3.1. Case Studies

This section presents WDE instances customized for specific regions. As part of this demonstration, we show data from both the WHOS and CUAHSI HIS systems. To demonstrate these capabilities, we created two different WDE regional customizations, one to demonstrate access to the HIS Central catalog and one to demonstrate access to WHOS customized dataset views. In both case studies, the goal was to show how WDE could be customized for a specific region, then discover, download, and visualize data from different WaterOneFlow web services at both the WDE catalog and server levels. As part of this demonstration, we show the ability to generate geospatial visualization and data plots. We demonstrate time series data retrieval and storage in a local file using XML (WaterML 1.0, WaterML 2.0) and CSV file formats. We show visualizations from a number of different sites. All the data shown in these demonstrations were retrieved using Pywaterml methods from within WDE to access WaterOneFlow web services.

We call the first WDE case study “WHOS Views”. It demonstrates access to WHOS customized dataset views for the La Plata Basin in South America and for an Arctic region. This case study presents: (1) a catalog of transboundary regions of the La Plata Basin in South America and the Arctic region, and (2) a catalog of all the countries currently providing data for these regions using WHOS (Figure 5).

We called the second WDE case study “HIS Central”. We demonstrate access to HIS Central catalog WaterOneFlow web services for the same regions. The only difference between the two case studies is the different WaterOneFlow web services used. Both applications have the same data discovery, download, and visualization capabilities.

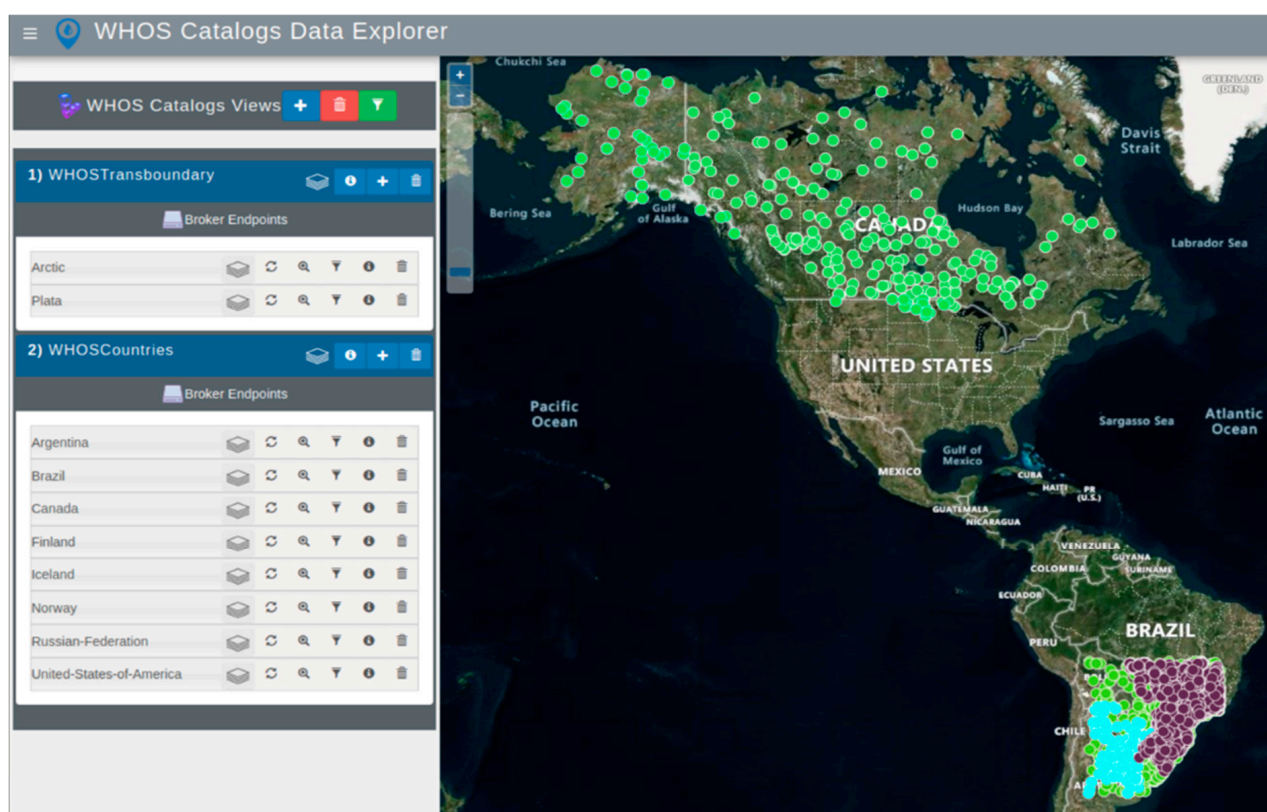


Figure 5. Customized Versions for the WHOS System.

3.2. Regional Customization

WDE can be customized to display different titles in the upper-left corner and to use the Web Mapping Service (WMS) layer to create a boundary that represents an area of interest. For each of the two case studies, WHOS Views and HIS Central, we created customized views, but we did not include a WMS layer for the boundary containing the observation sites. We did not add a WMS boundary layer because the geographic extent of the area of interest covers multiple countries on different continents, and the different colors representing the set of observation sites also makes data retrieved from the different servers distinguishable. If a boundary is added, it is generally either a watershed, region, or country outline.

3.3. Data Discovery

The WDE homepage presents a base map without any sites. This view allows the user to turn off or on the display of the different sites found using the WDE Catalog and server-level discovery methods. For the HIS Central case study, WDE performed general data discovery in the HIS Central catalog for any sites in the selected regions. In the WHOS Catalogs case study, it performed general data discovery in both the transboundary regions and the data provider countries catalog. For both case studies, the general discovery created a WDE catalog in the WDE database. After general data discovery, WDE displayed the retrieved data in the WDE catalog list.

In the WHOS Views case study, WDE discovered the transboundary and countries WaterOneFlow web services and their available web services, as shown in Figure 5. In the HIS Central case study, WDE did not discover all the WaterOneFlow web services from the HIS Central catalog because some contain large amounts of data, such as with the NWIS daily values. NWIS Unit values are available through WaterOneFlow web services from the U.S. Geological Survey (USGS) that slow down performance because of

their size. Therefore, we designed the HIS Central customized WDE to only discover six WaterOneFlow web services without a specific selection criterion, as depicted in Figure 6.

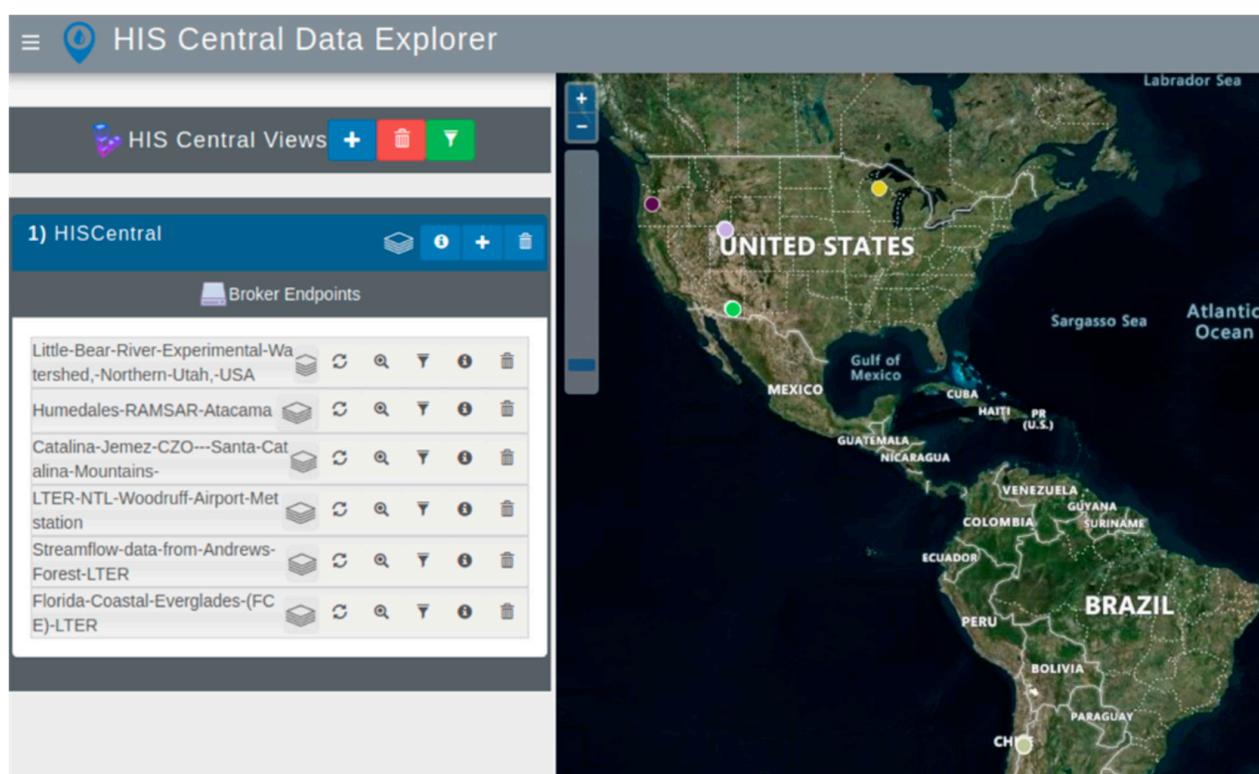


Figure 6. WDE Customized Versions for the CUAHSI Central.

We tested country-based discovery in both case studies. WDE provides a country-based discovery menu after selecting the green button in the toolbar to the right of the view names, shown in both Figures 5 and 6. This button allows the user to select a country if the customized WDE contains services from multiple countries. The menu lists the different countries in which WDE discovered WaterOneFlow web services. For the WHOS Views case study, the countries with available data were Canada, Iceland, Brazil, Russia, Argentina, Bolivia, Paraguay, Finland, Uruguay, Norway, and the United States of America. In the HIS Central case study, the only two countries with available data were Chile and the United States of America.

For discussion purposes, in the WHOS Views case study, we selected Brazil for the country-based discovery because it is part of the “Plata” server. Figure 7 shows the results of the country-based discovery for the WaterOneFlow web services that have data within Brazil. It reports that “Plata” is the only WDE server that contains sites inside Brazil.

To demonstrate general discovery at the server level, for the WHOS Catalogs case study we used the Humedales Ramsar Atacama WaterOneFlow web service registered at CUAHSI HIS Central. Figure 8 shows the results of the server general discovery. For the WHOS Catalog case study, WDE conducted variable discovery for the WaterOneFlow web service at the “Plata” server in the Transboundary catalog, with the results shown in Figure 9. The metadata from the different variables belonging to the La Plata WaterOneFlow web service are displayed in a table containing variable name, measurements units, and WHOS variable code.

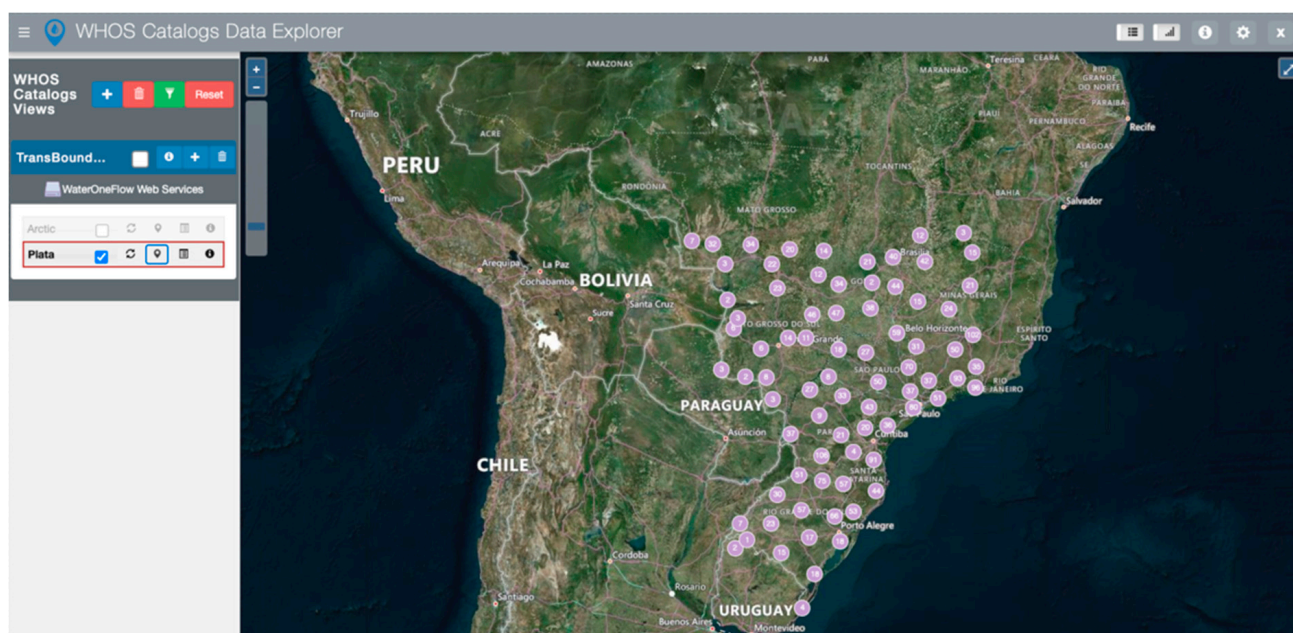


Figure 7. Country search for WaterOneFlow Web Services in Brazil showing the available server on the left and a plot of the observation sites on the map.

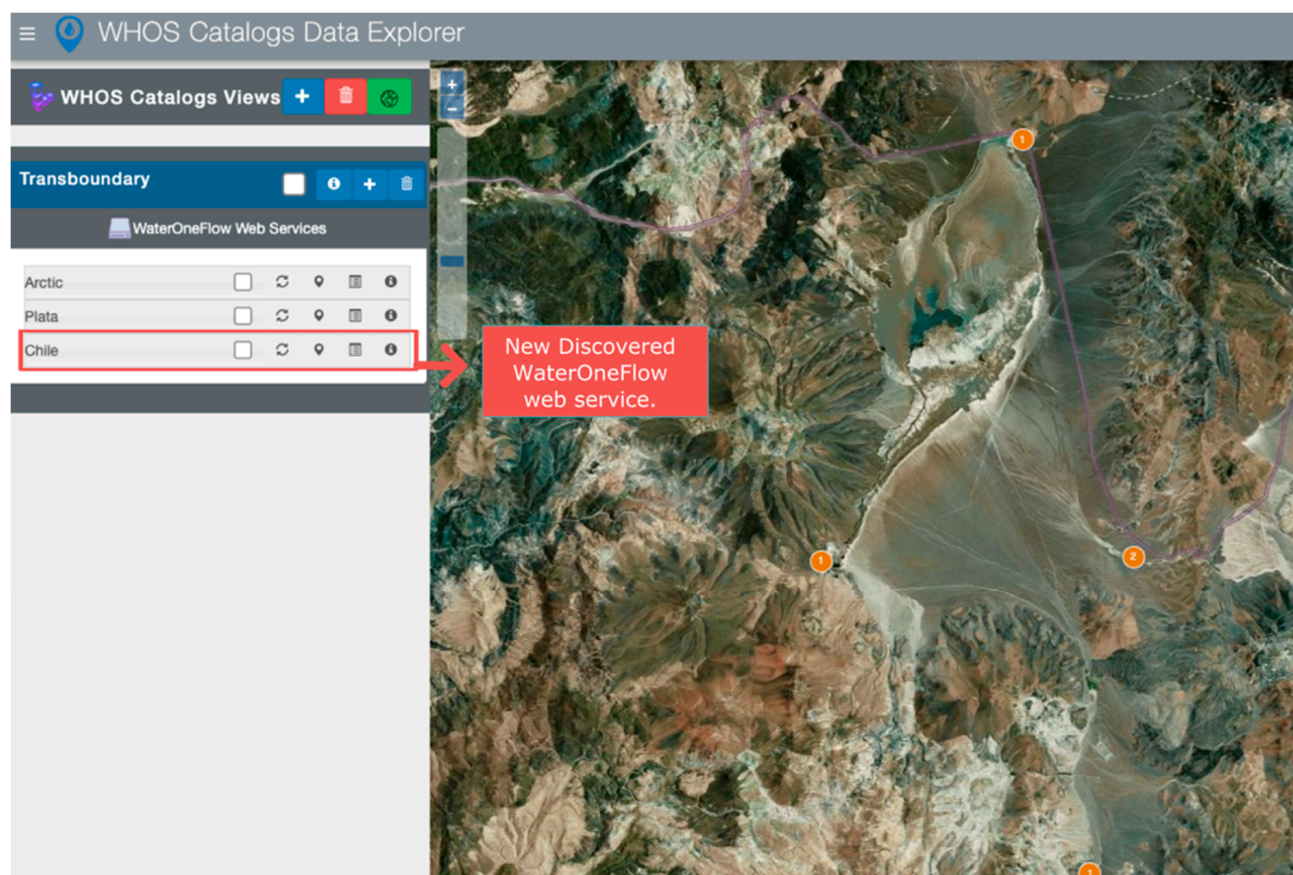


Figure 8. Server Level General Discovery for WHOS Catalogs.

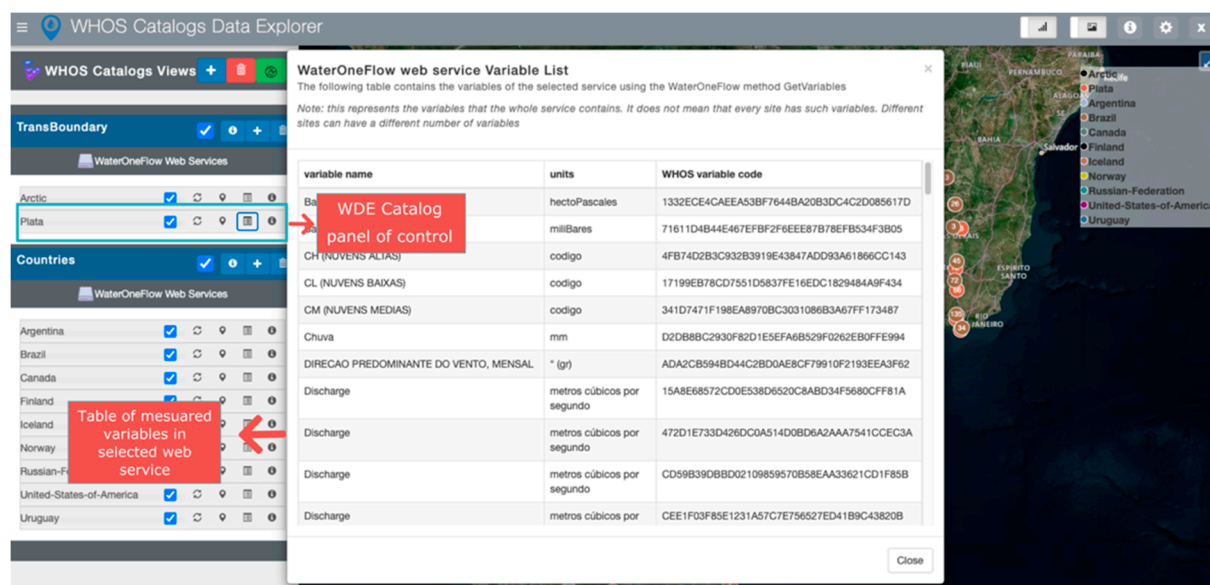


Figure 9. WDE Variable Discovery at Server Level.

3.4. Visualizing Data

We demonstrated data visualization for both case studies. Data visualization uses two different displays: one in the Site Information panel for time series visualization, and the other in the map for geospatial visualization. To generate a time series plot, we select sites and a variable of interest. Figure 10 depicts time series visualization for the air temperature variable in the WHOS Catalogs case study for Plata WaterOneFlow web service. Figure 11 shows a time series visualization for the reservoir storage variable in the HIS Central case study for the CALVIN_HHS WaterOneFlow web service.

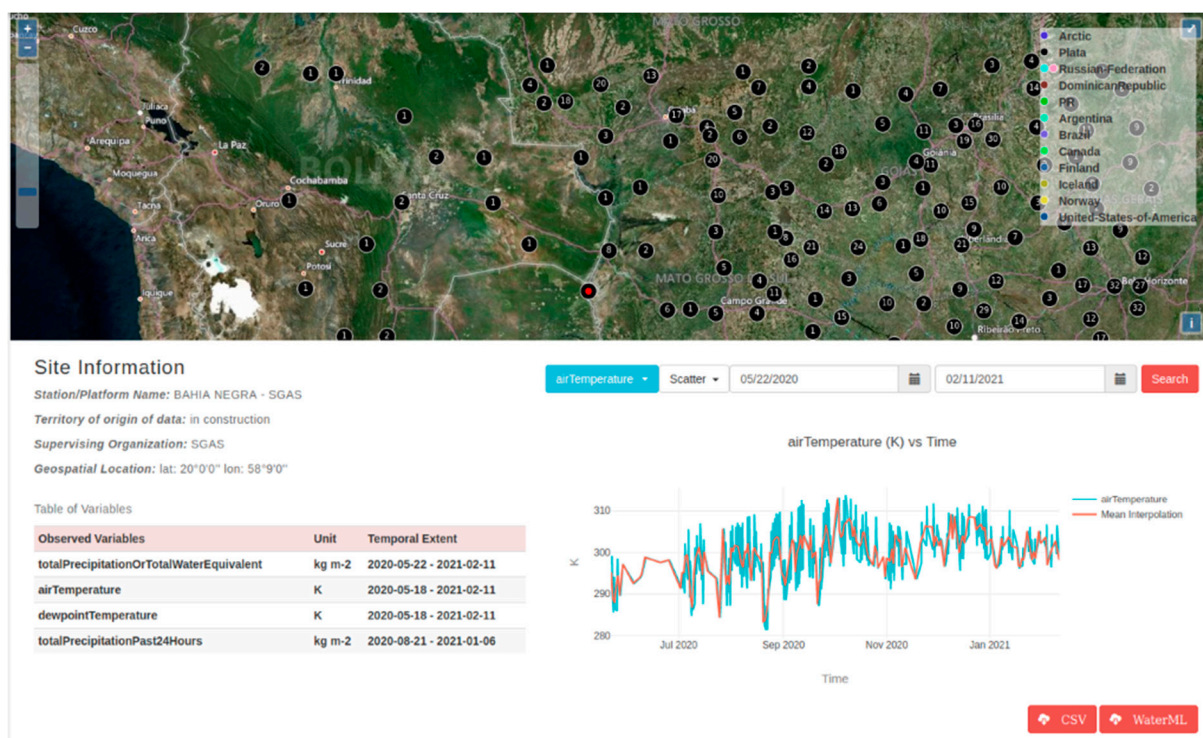


Figure 10. Visualization for the Air Temperature.

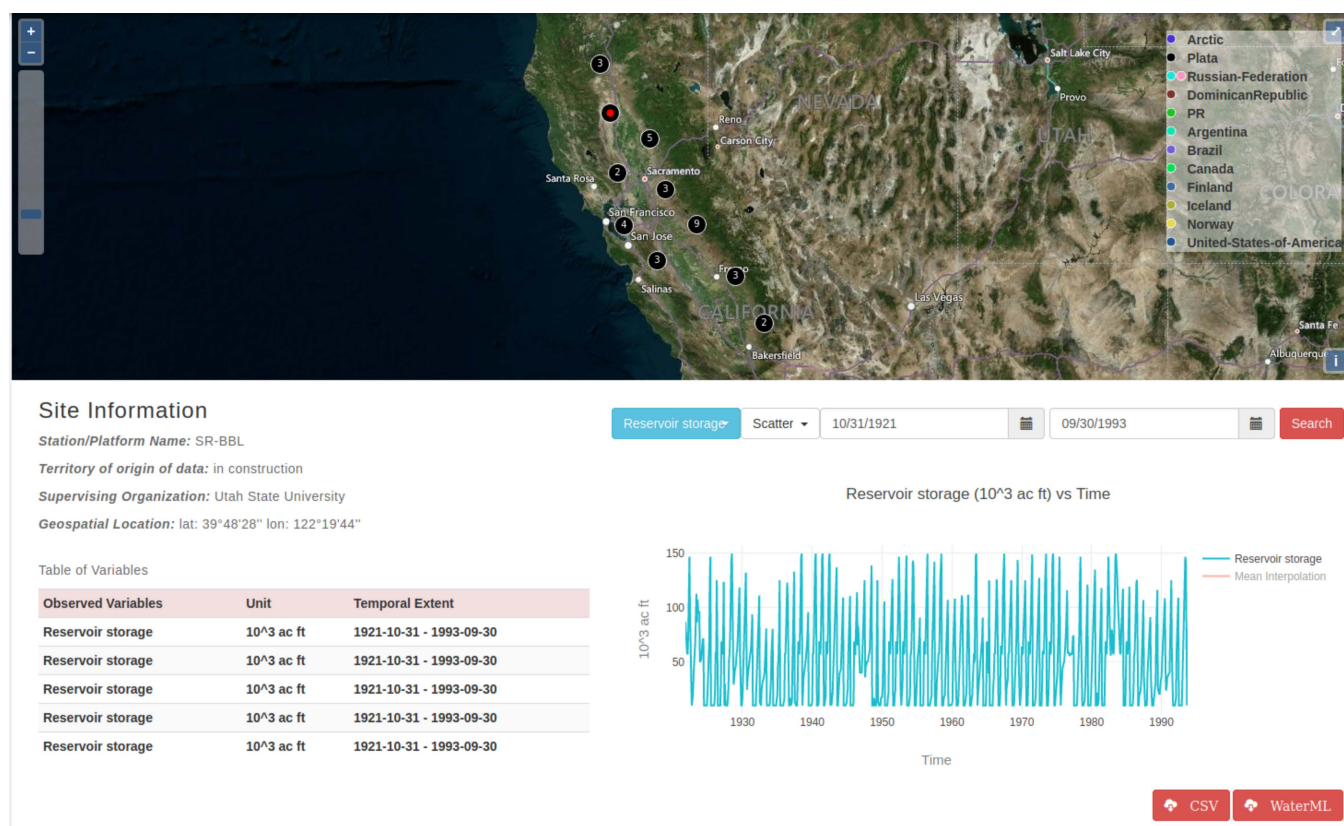


Figure 11. Visualization for the Reservoir Storage Variable.

3.5. Data Downloading

We demonstrate WDE data downloading in both case studies for two different instances, the first demonstrating data downloading to the local data base, and the second demonstrating the visualization of retrieved data that are not copied into the local database but are used internally in WDE. For these data we also demonstrate saving them to a local file.

In the first instance, we demonstrated the first data downloaded using the discovered data from the HIS Central case study, and from the country and transboundary dataset of customized views for the WHOS Catalogs case study. We downloaded metadata for the WaterOneFlow services to the WDE PostgreSQL database.

In the second instance, we demonstrated time series data visualization. Figure 12 shows the download of a CSV file associated with the Plata server from the WHOS Catalog case study. A plot of the data can be seen in the WDE screenshot. For this demonstration, we downloaded the total precipitation data for the Mariscal Estigarribia site. We saved the data in a CSV format; the WaterML format is also available but was not used here.

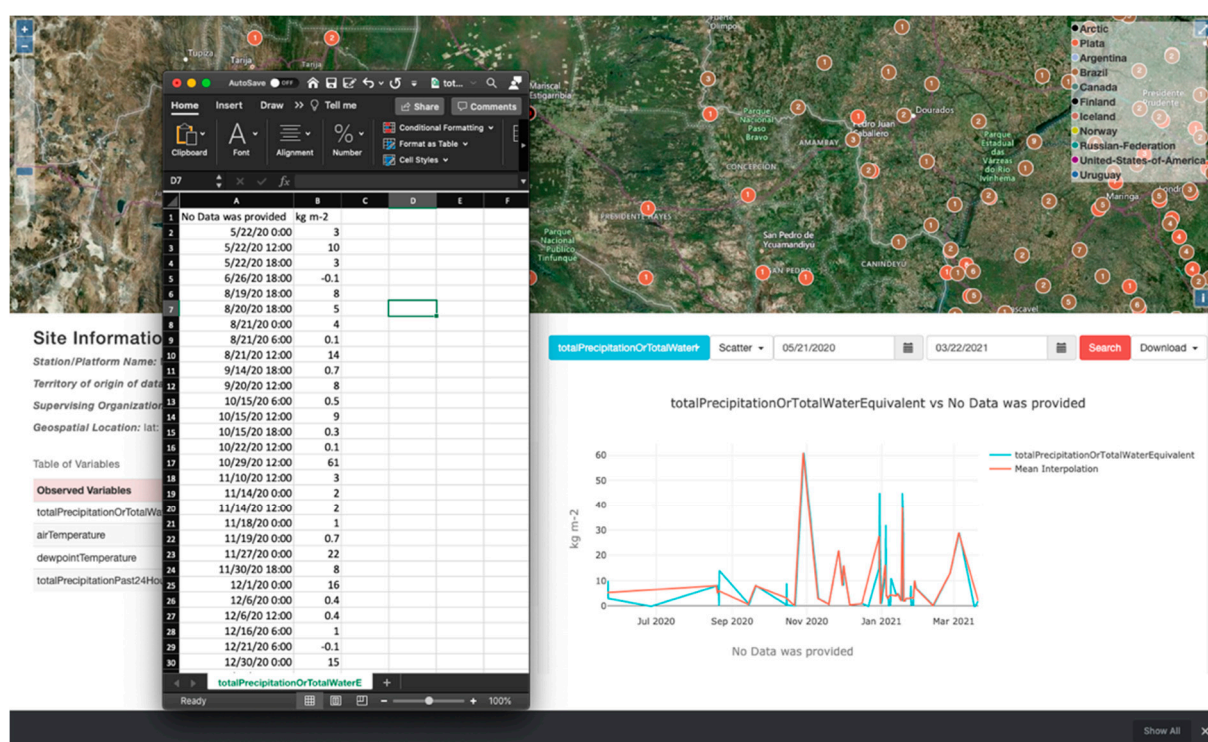


Figure 12. Data Downloading in CSV Format in the WDE Site Panel.

4. Discussion and Conclusions

In the WHOS Catalogs case study, WDE used the transboundary and country region catalogs and discovered two and nine WaterOneFlow services, respectively. In the HIS Central case study, WDE discovered six different WaterOneFlow web services. In discovering the different WaterOneFlow web services, we encountered that the data retrieved using the WaterOneFlow methods provided data with different WaterML structured responses in both systems. As a result, the Pywaterml was designed to use a common WaterML response structure as the baseline for the WDE data discovery in both case studies. Depending on the SOA system WaterOneFlow web services responses, Pywaterml might need further customization to suit different WaterML response structures.

Another important challenge of working with the WaterOneFlow web services responses was the time of the data retrieval, which limited the different data discoveries at each WDE level. For example, data discovery for given variables at the server and catalog levels is not possible without making an API request to each site in the WaterOneFlow web service. For this reason, data discovery for given variables was not included as a data discovery method due to the amount of time taken for execution.

As a web application developed in the Tethys Platform framework, WDE provides users the ability to customize the application for different regions and for different WaterOneFlow web services. This can provide important “branding” and lead to greater acceptance of the WDE and associated regional water data services.

In the process of developing WDE, we found that there was not a single Python library that connects to WaterOneFlow web services and executes the multiple methods they expose through their API. We developed the Pywaterml library to act as the core of WDE, and also as a stand-alone Python package that can connect to the different WaterOneFlow web services and execute their different methods to discover and download data. Since Pywaterml is a standalone library, it can be included in other applications with general or specific needs for accessing observational time series data from WaterOneFlow servers. The Pywaterml library provides modularity of the data discovery and data downloading functionalities as a client component.

Together, WDE and Pywaterml provide a complete open-source SOA client interface tool that can be customized for specific regions and datasets accessible from multiple WaterOneFlow web services. WDE provides end users with the ability to create a regional tool to provide data discovery, download, and analysis. Pywaterml provides extensibility to other applications to replicate and enhance WDE functionality.

Supplementary Materials: WDE web application is available online at <https://tethys-staging.byu.edu/apps/>.

Author Contributions: Conceptualization, methodology, software, analysis, investigation and writing- original draft preparation, G.R.B.; writing- review and editing G.R.B., E.J.N., D.P.A., N.L.J., G.P.W.; supervision, software E.B., J.L.S.L. and I.C.; supervision, project administration, and funding acquisition E.J.N., D.P.A., G.P.W., N.L.J. All authors have read and agreed to the published version of the manuscript.

Funding: We acknowledge the support of Brigham Young University's (BYU) Civil and Environmental Engineering Department and the World Meteorological Organization for their support assisting and supervising the development of this research. Support was also provided by the United States National Aeronautics and Space Administration (NASA), grant numbers 80NSCC18K0440 and 80NSSC20K0157 and by the National Science Foundation under collaborative grants ACI 1148453 and 1148090 for the development of HydroShare (<https://www.hydroshare.org>, accessed on 15 April 2021).

Data Availability Statement: The Pywaterml package is available for distribution via the Python Package Index (PyPI) and Conda-Forge and as source code via a GitHub repository at <https://github.com/BYU-Hydroinformatics/pywaterml> [35], accessed on 15 April 2021. Each distribution provides code documentation and examples as well as test cases. The source code of the demonstration web app for WDE is available through GitHub at <https://github.com/BYU-Hydroinformatics/Water-Data-Explorer> [36], accessed on 15 April 2021. Installation instructions for the specialized Django framework, Tethys, are referenced from the GitHub repository for WDE. Finally, the data used for the CUAHSI HIS test cases can be found in the following service endpoints while the data used for the WHOS test cases is still experimental and confidential:

- http://hydroportal.cuahsi.org/littlebearriverwof/cuahsi_1_1.asmx?WSDL [37], accessed on 13 January 2021
- http://hydroportal.cuahsi.org/Ramsar_atacama/cuahsi_1_1.asmx?WSDL [38], accessed on 13 January 2021
- http://hydroportal.cuahsi.org/czo_catalina/cuahsi_1_1.asmx?WSDL [39], accessed on 13 January 2021
- https://hydroportal.cuahsi.org/LTERNTLWoodruff/cuahsi_1_1.asmx?WSDL [40], accessed on 15 April 2021
- http://hydroportal.cuahsi.org/Andrewsforestlter/cuahsi_1_1.asmx?WSDL [41], accessed on 15 January 2021
- http://hydroportal.cuahsi.org/FCElTER/cuahsi_1_1.asmx?WSDL [42], accessed on 13 January 2021

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Challenges to Hydrological Observations. Available online: <https://public.wmo.int/en/bulletin/challenges-hydrological-observations> (accessed on 21 March 2020).
2. Lavers, D.A.; Harrigan, S.; Andersson, E.; Richardson, D.S.; Prudhomme, C.; Pappenberger, F. A vision for improving global flood forecasting. *Environ. Res. Lett.* **2019**, *14*, 121002. [CrossRef]
3. The World Bank Group; GFDDR. *Assessment of the State of Hydrological Services in Developing Countries*; World Bank Group: Washington, DC, USA, 2018.
4. UN-WATER Water: Transboundary Waters: Sharing Benefits, Sharing. Available online: https://scholar-google-com.erl.lib.byu.edu/scholar_lookup?hl=en&publication_year=2008&author=UN-Water&title=Transboundary+waters%3A+sharing+benefits%2C+sharing+responsibilities+%5Bonline%5D (accessed on 17 February 2021).

5. Tarboton, D.G.; Horsburgh, J.S.; Maidment, D.R.; Whiteaker, T.; Zaslavsky, I.; Piasecki, M.; Goodall, J.; Valentine, D.; Whitenack, T. Development of a community hydrologic information system. In Proceedings of the 18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation, Modelling and Simulation Society of Australia and New Zealand and International Association for Mathematics and Computers in Simulation, Cairns, Australia, 13–17 July 2009; pp. 988–994.
6. Boldrini, E.; Mazzetti, P.; Nativi, S.; Santoro, M.; Papeschi, F.; Roncella, R.; Olivieri, M.; Bordini, F.; Pecora, S. WMO Hydro-logical Observing System (WHOS): A Collaborative Implementation Approach. *Geophys. Res. Abstracts* **2019**, *21*, 13620.
7. Zaslavsky, I.; Whitenack, T.; Williams, M.; Tarboton, D.; Schreuders, K.; Aufdenkampe, A.K. The initial design of data sharing infrastructure for the critical zone observatory. In Proceedings of the Stroud Water Research Center, University of California, Santa Barbara, CA, USA, 28 September 2011; Jones, M.B., Gries, C., Eds.; University of California: Santa Barbara, CA, USA, 2011; pp. 145–150.
8. Lehnert, K.; Walker, J.; Carlson, R.; Hofmann, A.; Sarbas, B. Building the EarthChem system for advanced data management in igneous geochemistry. In Proceedings of the AGU Fall Meeting Abstracts, San Francisco, CA, USA, 13–17 December 2004.
9. Lehnert, K.A.; Carbotte, S.M.; Ryan, W.B.F.; Ferrini, V.; Block, K.; Arko, R.A.; Chan, C. IEDA: Integrated earth data applications to support access, attribution, analysis, and preservation of observational data from the ocean, earth, and polar sciences. In Proceedings of the Geophysical Research Abstracts, Vienna, Austria, 3–8 April 2011; Volume 13.
10. Lehnert, K.A.; Walker, D.; Block, K.A.; Ash, J.M.; Chan, C. EarthChem: Next developments to meet new demands (invited). In Proceedings of the AGU Fall Meeting Abstracts, San Francisco, CA, USA, 14–18 December 2009; Volume 12, p. V12C-01.
11. IOOS Data Management Planning and Coordination. Available online: <https://ioos.noaa.gov/data/contribute-data/data-management-planning-coordination/> (accessed on 9 March 2021).
12. Maidment, D.R. Bringing Water Data Together. *J. Water Resour. Plan. Manag.* **2008**, *134*, 95–96. [\[CrossRef\]](#)
13. Maidment, D.R. Hydrologic Information System Status Report. 2005, p. 224. Available online: <https://hydrology.usu.edu/dtarb/HISStatusSept15.pdf> (accessed on 15 April 2021).
14. Horsburgh, J.S.; Aufdenkampe, A.K.; Mayorga, E.; Lehnert, K.A.; Hsu, L.; Song, L.; Jones, A.; Damiano, S.G.; Tarboton, D.G.; Valentine, D.; et al. Observations Data Model 2: A community information model for spatially discrete Earth observations. *Environ. Model. Softw.* **2016**, *79*, 55–74. [\[CrossRef\]](#)
15. Horsburgh, J.S.; Tarboton, D.; Maidment, D.R.; Zaslavsky, I. A relational model for environmental and water resources data. *Water Resour. Res.* **2008**, *44*, 44. [\[CrossRef\]](#)
16. Conner, L.G.; Ames, D.P.; Gill, R.A. HydroServer Lite as an open source solution for archiving and sharing environmental data for independent university labs. *Ecol. Inform.* **2013**, *18*, 171–177. [\[CrossRef\]](#)
17. Horsburgh, J.S.; Tarboton, D.G.; Schreuders, K.A.T.; Maidment, D.R.; Zaslavsky, I.; Valentine, D. *HydroServer: A Platform for Publishing Space-Time Hydrologic Datasets*; American Water Resources Association: Orlando, FL, USA, 2010.
18. Tarboton, D.G.; Horsburgh, J.; Schreuders, K.; Maidment, D.; Zaslavsky, I.; Valentine, D. The HydroServer platform for sharing hydrologic data. In Proceedings of the AGU Fall Meeting Abstracts, San Francisco, CA, USA, 13–17 December 2010; Volume 2010, p. H53H-03.
19. Ames, D.P.; Horsburgh, J.; Cao, Y.; Kadlec, J.; Whiteaker, T.; Valentine, D. HydroDesktop: Web services-based software for hydrologic data discovery, download, visualization, and analysis. *Environ. Model. Softw.* **2012**, *37*, 146–156. [\[CrossRef\]](#)
20. Kadlec, J.; StClair, B.; Ames, D.P.; Gill, R.A. WaterML R package for managing ecological experiment data on a CUAHSI HydroServer. *Ecol. Inform.* **2015**, *28*, 19–28. [\[CrossRef\]](#)
21. Hooper, R.P.; Seul, M.; Pollak, J.; Couch, A. Realizing the potential of the CUAHSI water data center to advance earth science. In Proceedings of the AGU Fall Meeting Abstracts, San Francisco, CA, USA, 1 December 2015; Volume 42, p. H42A-03.
22. Whitenack, T. *CUAHSI HIS Central 1.2*; San Diego Supercomputer Center: San Diego, CA, USA, 2010; p. 42.
23. Horsburgh, J.S.; Tarboton, D.G.; Hooper, R.P.; Zaslavsky, I. Managing a community shared vocabulary for hydrologic observations. *Environ. Model. Softw.* **2014**, *52*, 62–73. [\[CrossRef\]](#)
24. Taylor, P.; Cox, S.; Walker, G.; Valentine, D.; Sheahan, P. WaterML2.0: Development of an open standard for hydrological time-series data exchange. *J. Hydroinformatics* **2013**, *16*, 425–446. [\[CrossRef\]](#)
25. Valentine, D.W.; Zaslavsky, I.; Whitenack, T.; Maidment, D. Design and implementation of CUAHSI WaterML and WaterOneFlow web services. In Proceedings of the AGU Fall Meeting Abstracts, IN53C-08, San Francisco, CA, USA, 10–14 December 2007; Volume 53.
26. CUAHSI CUAHSI HydroClient. Available online: <https://data.cuahsi.org> (accessed on 16 March 2021).
27. Boldrini, E.; Mazzetti, P.; Nativi, S.; Santoro, M.; Papeschi, F.; Roncella, R.; Olivieri, M.; Bordini, F.; Pecora, S. WMO Hydro-logical Observing System (WHOS) broker: Implementation progress and outcomes. In Proceedings of the EGU General Assembly Conference Abstracts, Vienna, Austria, 4–8 May 2020; p. 14755.
28. Organization (WMO). *WIGOS Guide to the WMO Integrated Global Observing System*; WMO: Geneva, Switzerland, 2019; ISBN 978-92-63-11165-4.
29. Pecora, S.; Lins, H.F. E-monitoring the nature of water. *Hydrol. Sci. J.* **2020**, *65*, 683–698. [\[CrossRef\]](#)
30. Boldrini, E.; Mazzetti, P.; Nativi, S.; Santoro, M.; Papeschi, F.; Roncella, R.; Olivieri, M.; Bordini, F.; Pecora, S. WMO Hydro-logical Observing System (WHOS) broker: Implementation progress and outcomes. In Proceedings of the Copernicus Meetings, Vienna, Austria, 3–8 May 2020.

31. Nelson, E.J.; Pulla, S.T.; Matin, M.A.; Shakya, K.; Jones, N.; Ames, D.P.; Ellenburg, W.L.; Markert, K.N.; David, C.H.; Zaitchik, B.; et al. Enabling Stakeholder Decision-Making with Earth Observation and Modeling Data Using Tethys Platform. *Front. Environ. Sci.* **2019**, *7*. [[CrossRef](#)]
32. Swain, N.R.; Christensen, S.D.; Snow, A.; Dolder, H.; Espinoza-Dávalos, G.; Goharian, E.; Jones, N.L.; Nelson, E.J.; Ames, D.P.; Burian, S.J. A new open source platform for lowering the barrier for environmental web app development. *Environ. Model. Softw.* **2016**, *85*, 11–26. [[CrossRef](#)]
33. Swain, N. Tethys Platform: A Development and Hosting Platform for Water Resources Web Apps. Ph.D. Thesis, Brigham Young University, Provo, UT, USA, 2015.
34. Plotly JavaScript Graphing Library. Available online: <https://plotly.com/javascript/> (accessed on 16 April 2021).
35. Romero Bustamante, E.G.; Ames, D.P.; Nelson, E.J.; Williams, G.; Jones, N.L. Pywaterml. Zenodo. 2021. Available online: <https://doi.org/10.5281/zenodo.4678818> (accessed on 15 April 2021).
36. Romero Bustamante, E.G.; Ames, D.P.; Nelson, E.J.; Williams, G.; Jones, N.L.; Boldrini, E.; Chernov, I. Water Data Explorer. Zenodo. 2021. Available online: <https://doi.org/10.5281/zenodo.4678966> (accessed on 15 April 2021).
37. Horsburgh, J.S.; Stevens, D.K.; Tarboton, D.G.; Mesner, O.; Spackman Jones, A.; Gurrero, S. Little Bear River Experimental Watershed, Northern Utah, USA. 2016. Available online: https://hiscentral.cuahsi.org/pub_network.aspx?n=52 (accessed on 13 January 2021).
38. Ministerio del Medio Ambiente (MMA), R. de A., Chile Humedales RAMSAR Atacama 2021. Available online: https://hiscentral.cuahsi.org/pub_network.aspx?n=5607 (accessed on 13 January 2021).
39. University of Arizona Catalina-Jemez CZO—Santa Catalina Mountains. 2015. Available online: https://hiscentral.cuahsi.org/pub_network.aspx?n=177 (accessed on 13 January 2021).
40. Lead PI, N.; Magnuson, J.; Carpenter, S.; Stanley, E. North Temperate Lakes LTER Meteorological Data—Woodruff Airport 1989—Current 2019. Available online: <https://lter.limnology.wisc.edu/index.php/dataset/north-temperate-lakes-lter-meteorological-data-woodruff-airport-1989-current> (accessed on 15 April 2021). [[CrossRef](#)]
41. Johnson, S.; Rothacher, J. Stream discharge in gaged watersheds at the Andrews Experimental Forest, 1949 to present. Long-Term Ecological Research. Forest Science Data Bank, Corvallis, OR. [Database]. 2018. Available online: <http://andlter.forestry.oregonstate.edu/data/abstract.aspx?dbcode=HF004> (accessed on 15 January 2021). [[CrossRef](#)]
42. FCE LTER Florida Coastal Everglades (FCE) LTER 2020. Available online: https://hiscentral.cuahsi.org/pub_network.aspx?n=5664 (accessed on 13 January 2021).