HydroServer Security and Access Control Design

Jeffery S. Horsburgh, Ketan Patil, David G. Tarboton, Kimberly A. T. Schreuders

10-19-2010

# 1. Introduction

Academic researchers who are collecting data within experimental watersheds, observatories, and research sites have need for the ability to control access to both private and public data resources in their data collection, management, and publication process. One mechanism for publishing data from experimental sites is using a CUAHSI Hydrologic Information System (HIS) HydroServer. HydroServer consists of a set of software components that together enable the publication of hydrologic observations data on the Internet. Components of HydroServer include: 1) and Observations Data Model (ODM), which is a relational database schema for storing hydrologic observations data; 2) ODM Tools and Utilities – a set of software programs for loading data into and managing data within ODM databases; 3) the WaterOneFlow Web Services, which publish the contents of an ODM database on the Internet in Water Markup Language (WaterML) format; 4) a capabilities database and web service that publish the capabilities of a HydroServer on the Internet; 5) the HydroServer website, which lists and provides access to the data services hosted on a HydroServer; and 6) the Time Series Analyst and HydroServer Map applications, which provide simple, Internet-based visualization tools for data hosted on a HydroServer.

To address the need for controlling access to data hosted and published on a HydroServer, the design for new functionality that enables HydroServer Administrators and data owners to control access to both public and private data resources hosted on a HydroServer is described in this document.

# 2. Definitions

The following are specific definitions for some of the terminology in this document:

***Access Control Rules*** – Rules that are specified by a data owner/provider and that limit who can access data resources and what kinds of operations (e.g., read, update, delete) can be performed.

***Authentication*** – The process by which a data consumer provides credentials that verify his or her identity.

***Authorization*** – The process by which a data owner/provider enables a data consumer to access/download a data resource.

***Client Application*** – A software program that accesses or downloads data or metadata from a HydroServer.

***Data Consumer*** – A person or software program that wants to download data hosted on a HydroServer.

***Data Owner*** – The person or organization that collected the data and wishes to control its distribution.

***Data Resource*** – An identifiable unit of data to which access control rules can be applied and that can be downloaded.

***HydroServer Administrator*** – The person or persons that have full administrative access to a HydroServer and that have authority to grant access control privileges.

***HydroServer User List*** – A list of data consumers that have been registered on a HydroServer.

***HydorServer Resource List*** – A list of data resources that are hosted on a HydrServer.

***Metadata*** – Descriptive information about data, including where the data was collected, which variables were measured, what methods were used, who collected the data, etc.

***Registration*** – The process by which a data consumer’s user profile is added to the HydroServer User List so that the data consumer can be given access to data resources.

***Security and Access Control*** – The overall process by which data owners/providers control which data consumers can access their data resources.

***User Profile*** – A list of attributes that identify a data consumer.

# 2. Architecture of Security and Access Control System

The HydroServer security and access control system is made up of the components illustrated in Figure XX. The following are brief descriptions of each of the components shown in the figure:

1. **Client application**: Client applications are those that request data or metadata resources hosted on a HydroServer in behalf of a data consumer. HydroDesktop is an example client application.
2. **HIS Central Security Certificate Authority**: The HydroServer security and data access control system relies on security certificates as the mechanism for identifying users within the system. In order to be valid, security certificates must be issued by a trusted certificate authority (CA). HydroServers will rely on a CA hosted at HIS Central. Data consumers will log into the HIS Central CA using a single sign-on account and will be issued a certificate that they can use across all HydroServers.
3. **WaterOneFlow Web Services**: The WaterOneFlow (WOF) Web services contain methods for accessing data and metadata stored within an Observations Data Model (ODM) database on a HydroServer. The WOF services must ensure that data consumers are both authenticated and authorized before returning protected data and metadata resources.
4. **ODM Database:** ODM is a relational database schema within which hydrologic observations are stored on a HydroServer.
5. **Security Web Service:** The Security Web Service is a Web service application programmer’s interface (API) for accessing public security functions on a HydroServer. For example, these functions include requesting creation of a user within a HydroServer’s Use List and requesting that a particular user be authorized to access a protected data resource.
6. **Security Core:**  The Security Core component is a set of internal functions that are accessed by the WOF web services and the Security Web Service and that provide common functionality for authenticating Data consumers, authorizing data access requests, and resolving which resources in the HydroServer Resource List have been requested based on parameters passed to the web services.
7. **Security Database:** The Security Database stores the HydroServer User List, Resource List, and Access Control Rules.
8. **Security Admin Tool:** The Security Admin Tool is a software application that enables HydroServer administrators to maintain the contents of the Security Database (e.g., the User List, Resource List, and Access Control Rules. It accesses the Security Database through the Security Web Service and Security Core component.

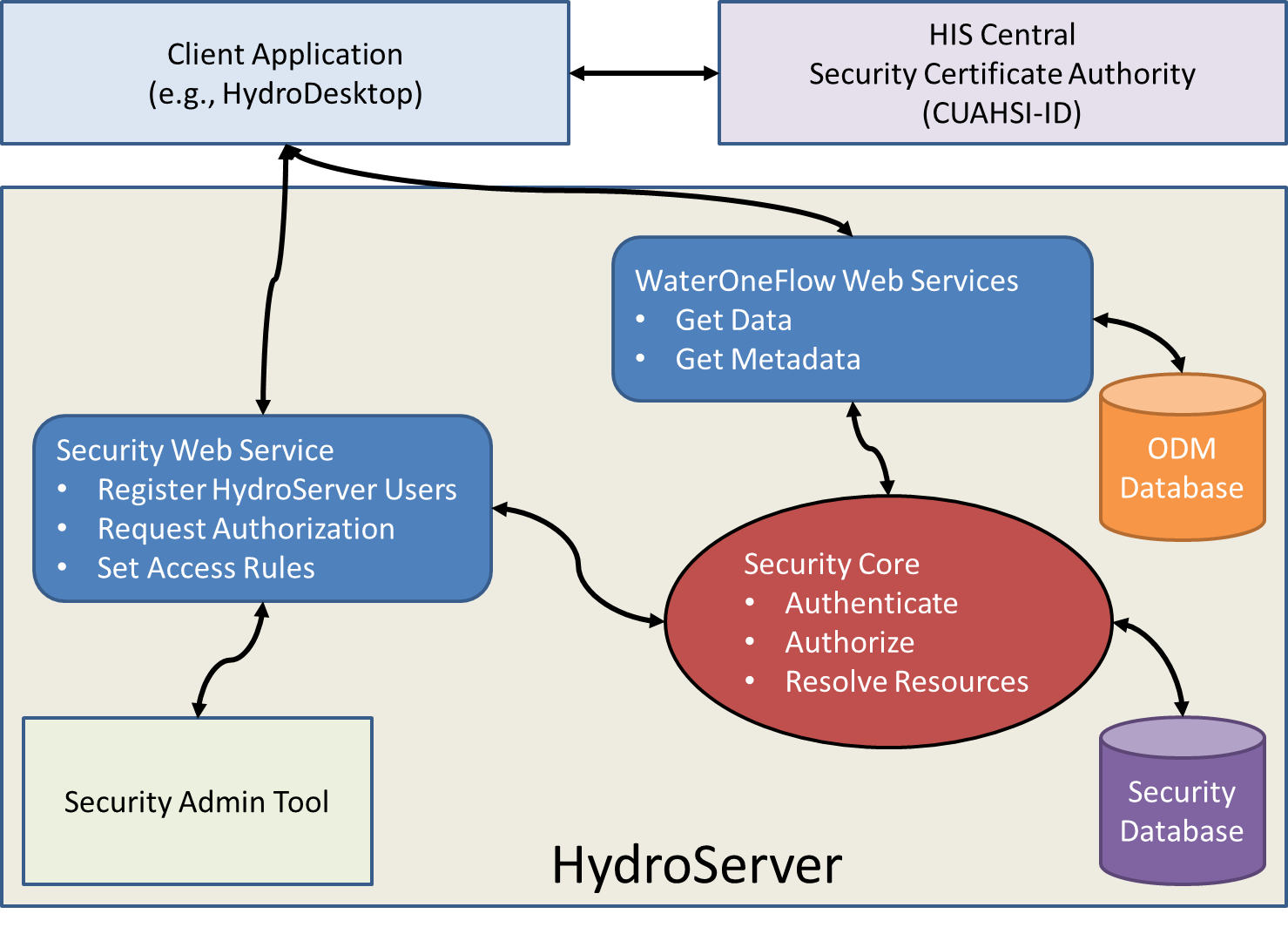


Figure XX. Architecture of the HydroServer Security and Access Control System.

Using the HydroServer security and access control system, each Get request (e.g., GetSites, GetValues) from a client application to the WaterOneFlow web services must be accompanied by a valid security certificate. Certificates will be issued to data consumers by the HIS Central CA and will identify the data consumer that issued the Get request. The WaterOneFlow Web Services will be required to check each Get request to ensure that the person submitting the request is authenticated (has a valid certificate and exists in the HydroServer’s User List) and authorized (based on access control rules stored in the Security Database) to access the resource that they have requested. Client applications can request that data consumers be added to the HydroServer User List through the Security Web Service. Client applications can also request authorization to access resources on behalf of a user through the Security Web Service. An administrative tool enables HydroServer administrators to manage the contents of the Security Database (i.e., the User List, the Resource List, and the Access Control Rules) through the Security Web Service.

As an example, the overall process for accessing protected data resources on a HydroServer involves the following steps:

1. Using a client application or by interacting directly with the HIS Central CA, a data consumer creates a user account with the HIS Central CA and logs in.
2. The HIS Central CA issues a security certificate to the data consumer that is stored on the data consumer’s hard drive. The security certificate will identify the data consumer across all HydroServers.
3. Using a client application or calling a HydroServer’s Security Web Service directly, the data consumer passes their security certificate to the HydroServer and requests that their user profile (contained within their certificate) be registered on the HydroServer.
4. The HydroServer Security Web Service adds the data consumer’s user profile from their certificate to the HydroServer User List.
5. Using a client application or calling the HydroServer Security Web Service directly, the data consumer passes their security certificate to the HydroServer and requests authorization to access one or more protected data resources.
6. The HydroServer Security Service queues the authorization request to be acted upon by a HydroServer Administrator.
7. A HydroServer Administrator uses the Security Admin Tool to approve the authorization request, giving the data consumer permission to access the protected data resources that they have requested. The data consumer is notified that access has been granted.
8. Using a client application or calling the WaterOneFlow Web Service directly, the data consumer passes their security certificate and requests to download a protected data resource.
9. The WaterOneFlow Web Service validates the data consumer’s security certificate, ensures that the data consumer has been registered in the HydroServer User List and that the data consumer has been authorized to access the data, and then returns the protected data resource to the data consumer.

# 3. Design Diagrams

In the following sections, specific processes are described that have been identified as requirements in the design for HydroServer security. Each of the processes is described by a sequence diagram formulated to identify the key functional requirements that the systems designed for each process will have to support. These diagrams also show the interaction of the system components and define the interfaces and messages transmitted between components.

## 3.1. Client Requests User Registration

**Goal:** A data consumer uses a client application like HydroDesktop to request that their user profile be added to a HydroServer’s User List.

**Summary:** Data consumers cannot be authorized to access protected resources on a HydroServer until their profile exists in the HydroServer User List. Client applications like HydroDesktop can use this sequence to request that the data consumer’s profile be added to the User List on a HydroServer. These requests will be approved automatically, provided that the data consumer provides a valid security certificate, which contains their profile, with their request. By default, data consumers in the HydroServer User List have no access permissions until they are granted by a HydroServer Administrator.

**Actors:** Data consumer, HydroServer Client (HydroDesktop), HydroServer Security Service, HydroServer Security Core

**Pre-Conditions:** The data consumer has obtained a valid security certificate from the HIS Central CA.

**Triggers:** A data consumer wants to be authorized to access protected resources on a HydroServer.

**Post Conditions:** The data consumer’s profile is added to the HydroServer’s User List. An optional notification is sent to the HydroServer Administrator with an alert that a new user profile has been added to the HydroServer User List.

**Notes:** None.

Figure XX. Client Requests User Registration Sequence Diagram.

## 3.2. Administrator Authorizes User

**Goal:** A HydroServer Administrator creates access control rules that enable a data consumer whose profile exists in the HydroServer User List to access protected data resources on a HydroServer.

**Summary:** At any time, a HydroServer Administrator can create access control rules that authorize data consumers from a HydroServer’s User List to access protected data resources on a HydroServer. The HydroServer Administrator logs into the Admin tool and uses a graphical user interface to create appropriate access control rules.

**Actors:** HydroServer Administrator, Admin Tool, Security Service, Security Core

**Pre-Conditions:** The data consumers and protected data resources to which the access control rules apply exist in the HydroServer User List and Resource List, respectively.

**Triggers:** A HydroServer Administrator wants to create access control rules.

**Post Conditions:** One or more new access control rules is created in the HydroServer Security Database.

**Notes:** None.

Figure XX. Administrator Authorizes User Sequence Diagram.

## 3.3. Client Requests Authorization for a User

**Goal:** A data consumer uses a client application like HydroDesktop to request authorization to access a protected data resource on a HydroServer.

**Summary:** Data consumers must be authorized by a HydroServer administrator to access protected resources on a HydroServer. Client applications like HydroDesktop can use this sequence to request that the data consumer be authorized to access a protected resource. The authorization request is then placed in a queue that must be acted upon by a HydroServer Administrator to be approved or denied.

**Actors:** Data consumer, Client Application (HydroDesktop), Security Service, Security Core

**Pre-Conditions:** The data consumer has obtained a valid security certificate from the HIS Central CA. The data consumer’s profile already exists in the HydroServer User List. The protected data resource that the data consumer wishes authorization to access exists in the HydroServer Resource List.

**Triggers:** A data consumer wishes to be authorized to access a protected data resource on a HydroServer.

**Post Conditions:** An authorization request is queued within the HydroServer Security Database to be acted upon by a HydroServer Administrator. An optional notification is sent to the HydroServer Administrator with an alert that new authorization requests are pending.

**Notes:** None.

Figure XX. Client Requests Authorization for a User Sequence.

## 3.4. Administrator Manages Authorization Requests

**Goal:** A HydroServer Administrator approves or denies data consumers’ requests for authorization to access protected data resources on a HydroServer.

**Summary:** Requests from data consumers to be authorized to access protected resources on a HydroServer are queued in the Security Database and must be acted upon by a HydroServer Administrator. The HydroServer Administrator logs into the Admin tool and uses a graphical user interface to review the requests that have been made and approve or deny each one.

**Actors:** HydroServer Administrator, Admin Tool, Security Service, Security Core

**Pre-Conditions:** Authorization requests from data consumers have been queued in the Security Database. An optional notification has been sent to the HydroServer Administrator with an alert that new authorization requests have been queued in the HydroServer Security Database.

**Triggers:** A HydroServer Administrator wishes to respond to pending authorization requests.

**Post Conditions:** Pending authorization requests are removed from the queue as they are approved or denied. New access control rules are created.

**Notes:** None.

Figure XX. Administrator Manages Authorization Requests Sequence Diagram.

## 3.5. Client Gets Metadata for a User

**Goal:** A client application, like HydroDesktop, retrieves metadata from a HydroServer for a data consumer.

**Summary:** Client applications, like HydroDesktop or the HIS Central metadata harvesters, need the ability to retrieve metadata about data resources hosted on a HydroServer. Some of the metadata may be protected. Because of this, client applications must be authenticated and authorized to retrieve metadata records. In the case that a client application is not authenticated, requests for metadata will be treated as if they are from an anonymous user and only unprotected metadata will be returned.

**Actors:** Data consumer, Client Application, WaterOneFlow Web Service, Security Core, ODM Database

**Pre-Conditions:** The data consumer has obtained a valid security certificate from the HIS Central CA. The data consumer exists in the HydroServer User List. The data consumer has been authorized to access the metadata records that have been requested.

**Triggers:** A data consumer wants to access metadata about data resources hosted on a HydroServer. The HIS Central Metadata Harvester wants to access metadata about data resources hosted on a HydroServer for the purpose of cataloging the contents of the HydroServer at HIS Central.

**Post Conditions:** Metadata records are returned to the data consumer.

**Notes:** None.

Figure XX. Client Gets Metadata for a User Sequence Diagram.

## 3.6. Client Gets Data for a User

**Goal:** A client application, like HydroDesktop, retrieves data from a HydroServer for a data consumer.

**Summary:** Client applications, like HydroDesktop, need the ability to retrieve data resources hosted on a HydroServer. Some of the data may be protected. Because of this, client applications must be authenticated and authorized to retrieve data. In the case that a client application is not authenticated, requests for data will be treated as if they are from an anonymous user and only unprotected data will be returned.

**Actors:** Data consumer, Client Application, WaterOneFlow Web Service, Security Core, ODM Database

**Pre-Conditions:** The data consumer has obtained a valid security certificate from the HIS Central CA. The data consumer exists in the HydroServer User List. The data consumer has been authorized to access the data that have been requested.

**Triggers:** A data consumer wants to access data hosted on a HydroServer.

**Post Conditions:** Data are returned to the data consumer.

**Notes:** None.

Figure XX. Client Gets Data for a User Sequence Diagram.

## 3.7. Security Authenticate User

**Goal:** A HydroServer service determines the identity of a data consumer that has made a data access request.

**Summary:** Services on a HydroServer (e.g., WaterOneFlow) must have the ability to determine the identity of the data consumer that issued a request to access data or metadata. In the case that a request is not authenticated, it will be treated as if it is from an anonymous user and only unprotected data and/or metadata will be returned.

**Actors:** WaterOneFlow web service (or other HydroServer Service), Security Core

**Pre-Conditions:** A data access request has been made – e.g., a Get request to a WaterOneFlow Web Service.

**Triggers:** A data access request has been made – e.g., a Get request to a WaterOneFlow Web service.

**Post Conditions:** A UserInformationList is sent back to the calling service that contains the user’s ID and profile information.

**Notes:** None.

Figure XX. Security Authenticate User Sequence Diagram.

## 3.8. Security Resolve Resources

**Goal:** A HydroServer service determines which specific data resources have been requested from the query parameters passed to the data access service (e.g., a WaterOneFlow Get method).

**Summary:** Access control rules for time series of hydrologic time series hosted on a HydroServer are set at the Data Series level. Because WaterOneFlow Web Services will allow for multiple Data Series to be returned with one GetValues call, the WaterOneFlow Web Services require a method for resolving which specific data series match the query parameters that a data consumer has passed to the WaterOneFlow GetValues method. Once a list of series has been resolved, the WaterOneFlow service can then test each series to see if the data consumer is authorized and only return the ones that the data consumer is authorized to access. A similar situation exists with the WaterOneFlow metadata methods, where a single web service method call can return many metadata records.

**Actors:** WaterOneFlow web service (or other HydroServer Service), Security Core

**Pre-Conditions:** A data or metadata access request has been made.

**Triggers:** A data or metadata access request has been made.

**Post Conditions:** A list of HydroServer data resources matching the request’s query parameters is returned to the calling service.

**Notes:** None.

Figure XX. Security Resolve Resources Sequence Diagram.

## 3.9. Security Authorize Request

**Goal:** A HydroServer service (e.g., WaterOneFlow) determines whether a data consumer is authorized to access a specific data resource.

**Summary:** When a data access request is received by a HydroServer service (e.g., a WaterOneFlow GetValues request), the service needs the capability to determine whether the data consumer is authorized to access the data. The same is true of metadata.

**Actors:** WaterOneFlow web service (or other HydroServer Service), Security Core

**Pre-Conditions:** A data or metadata access request has been made.

**Triggers:** A data or metadata access request has been made.

**Post Conditions:** True (Authorized) or False (Not Authorized).

**Notes:** None.

Figure XX. Security Authorize Request Sequence Diagram.

# 4. Application Programming Interfaces

This section details the Application Programming Interfaces (APIs) that are either part of or use the HydroServer security and data access control functionality. Interfaces that may be used internally within each of the following components are not described.

## 4.1. Security Web Service

The HydroServer Security Web Service provides a mechanism for data consumers to register their user profile with a HydroServer and to request authorization to access specific data resources. It also provides other HydroServer services with functionality for maintaining the Security Database.

Table ??. Web service methods defined in the HydroServer Security Service (HS\_Security).

|  |  |
| --- | --- |
| **Web Service Method** | **Parameters** |
| RegisterUser() | (Certificate) -> String |
| GetAuthorizationInfo() | (ResourceParametersList, Status, Certificate) - > AuthorizationInfoList |
| GetResourceInfo() | (ResourceParametersList, Certificate) -> ResourceInfoList |
| GetResourceInfoByGUID() | (ResourceGUID), Certificate) -> ResourceInfo XML Object |
| GetUserInfo() | (UserParametersList, Certificate) - > UserInfoList |
| RequestAuthorization() | (OperationType, ResourceType, ResourceParametersList, Certificate) -> String |
| SetAccess() | (UserID, ResourceGUID, AccessLevel, Certificate) - > Boolean |

**HS\_Security.RegisterUser**(*Certificate*) -> String

This method is used to register a data consumer within the User List of a HydroServer.

**Parameters**: Certificate – a valid security certificate issued by the HIS Central CA

**Returns**: A message indicating that status of the user registration

**Return Type**: String

**Raises**: Exceptions.InvalidCertificate

Exceptions.InvalidRequest

**HS\_Security.GetAuthorizationInfo**(*ResourceParametersList, Status, Certificate*) - > AuthorizationInfoList

This method is used to get a list of the access control rules stored within the Security Database.

**Parameters**: ResourceParametersList – a list of query parameters that can be used to resolve the specific resources for which access control rules are to be returned

Status – the status of the access control rules to be returned (e.g., pending or approved)

Certificate – a valid security certificate issued by the HIS Central CA

**Returns**: An XML object containing a list of access control rules for the resources specified in the ResourceParametersList, having status defined by Status, and that the user identified in the certificate is authorized to see

**Return Type**: AuthorizationInfoList

**Raises**: Exceptions.InvalidCertificate

Exceptions.NotAuthorized

Exceptions.InvalidResource

Exceptions.InvalidRequest

**HS\_Security.GetResourceInfo**(*ResourceParametersList, Certificate*) - > ResourceInfoList

This method is used to retrieve information about data resources stored in the Security Database Resource List.

**Parameters**: ResourceParametersList – a list of query parameters that can be used to resolve the specific resources for which information is to be returned

Certificate – a valid security certificate issued by the HIS Central CA

**Returns**: An XML object containing a list of data resource information from the Security Database Resource List

**ReturnType**: ResourceInfoList

**Raises**: Exceptions.InvalidCertificate

Exceptions.NotAuthorized

Exceptions.InvalidResouce

Exceptions.InvalidRequest

**HS\_Security.GetResourceInfoByGUID**(*ResourceGUID, Certificate*) - > ResourceInfo XML Object

This method is used to retrieve information about a single data resource stored in the Security Database Resource List.

**Parameters**: ResourceGUID – a GUID that can be used to resolve the specific resource for which information is to be returned

Certificate – a valid security certificate issued by the HIS Central CA

**Returns**: An XML object containing resource information from the Security Database Resource List

**ReturnType**: ResourceInfo XML Object

**Raises**: Exceptions.InvalidCertificate

Exceptions.NotAuthorized

Exceptions.InvalidResouce

Exceptions.InvalidRequest

**HS\_Security.GetUserInfo**(*UserParametersList, Certificate*) - > UserInfoList

This method is used to retrieve information about users stored in the Security Database User List.

**Parameters**: UserParametersList – a list of parameters that identify users for which information is to be returned

Certificate – a valid security certificate issued by the HIS Central CA

**Returns**: An XML object containing a list of user profile information from the Security Database User List.

**ReturnType**: UserInfoList

**Raises**: Exceptions.InvalidCertificate

Exceptions.NotAuthorized

Exceptions.InvalidUser

Exceptions.InvalidRequest

**HS\_Security.RequestAuthorization**(*OperationType, ResourceType, ResourceParametersList, Certificate*) -> String

This method is used by data consumers to request authorization to access protected resources on a HydroServer.

**Parameters:** OperationType – the type of operation to be performed, e.g., create, read, update, delete

ResourceType – the type of resource that is to be operated upon, e.g., time series data, etc.

ResourceParametersList – a list of query parameters that can be used to resolve the specific resources that the user wants to operate upon

Certificate – a valid security certificate issued by the HIS Central CA

**Returns:** A message indicating that the request has been queued.

**Return Type:** String

**Raises:** Exceptions.InvalidCertificate

Exceptions.InvalidResource

Exceptions.InvalidRequest

**HS\_Security.SetAccess**(*UserID, ResourceGUID, AccessLevel, Certificate*) - > Boolean

This method is used to create access control rules within the Security Database.

**Parameters**: UserID – a valid UserID from the Security Database User List

ResourceGUID – a valid ResourceGUID from the Security Database Resource List

AccessLevel – the specific permission that is being granted

Certificate – a valid security certificate issued by the HIS Central CA

**Returns**: True or False

**Return Type**: Boolean

**Raises**: Exceptions.InvalidCertificate

Exceptions.NotAuthorized

Exceptions.InvalidUser

Exceptions.InvalidResource

Excpetions.InvalidRequest

## 4.2. Security Core Component

The Security Core Component consists of a common set of functions used by both the Security Web Service and the WaterOneFlow Web Services to perform authentication and authorization of users as well as other essential security functions. All of the Security Core functions are internal to a HydroServer and are not exposed via a public web service interface. The Security Core component will be referenced by the Security Service and the WaterOneFlow Web Services.

Table ??. Functions defined in the HydroServer Security Core Component (HS\_SecurityCore).

|  |  |
| --- | --- |
| **Function** | **Parameters** |
| EmailNotification() | (Message, MailServerSettings) - > Boolean |
| GetAuthorizationInfo() | (ResourceParametersList, Status, UserID) - > AuthorizationInfoList |
| IsAuthorized() | (OperationType, ResourceType, ResourceGUID, UserID) - > Boolean |
| QueueAuthorizationRequest() | (OperationType, UserID, ResourceGUID) - > Boolean |
| RegisterUser() | (UserProfile) - >Boolean |
| ResolveResource() | (ResourceParametersList) - > ResourceInfoList |
| ResolveUser() | (UserProfileList) - > UserInfoList |
| SetAccess() | (UserID, ResourceGUID, AccessRule) - > Boolean |
| ValidateCertificate() | (Certificate) - > Boolean |

**HS\_SecurityCore.EmailNotification**(*Message, MailServerSettings*) - > Boolean

This function is used by HydroServer services to send a notification to a HydroServer Administrator.

**Parameters**: Message – as string that contains the text of the notification

MailServerSettings – a list of parameters that specify how to send the notification (e.g., SMTP server address, etc.)

**Returns**: True or False

**ReturnType**: Boolean

**Raises**: Exceptions.MailServerTimeout

Exceptions.InvalidRequest

**HS\_SecurityCore.GetAuthorizationInfo**(*ResourceParametersList, Status, UserID*) - > AuthorizationInfoList

This function is used by HydroServer services to get a list of access control rules from the Security Database.

**Parameters**: ResourceParametersList - a list of query parameters that can be used to resolve the specific resources for which access control rules are to be returned

Status – the status of the access control rules to be returned (e.g., pending or approved)

UserID – the ID of the requestor from the HydroServer User List

**Returns**: A list of access control rules for the resources specified in the ResourceParametersList, having status defined by Status, and that the requestor identified by the UserID is authorized to access

**ReturnType**: AuthorizationInfoList

**Raises**: Exceptions.InvalidUser

Exceptions.NotAuthorized

Exceptions.InvalidResource

Exceptions.InvalidRequest

**HS\_SecurityCore.IsAuthorized**(*OperationType, ResourceType, ResourceGUID, UserID*) - > Boolean

This function is used by HydroServer services to determine whether a data consumer is authorized to access a particular data resource.

**Parameters**: OperationType – the type of operation to be performed (e.g., read, create, delete)

ResourceType – the type of resource to be operated upon

ResourceGUID – the GUID of the specific resource for which access is requested from the Security Database Resource List

UserID – the ID of the requestor from the Security Database User List.

**Returns**: True or False

**ReturnType**: Boolean

**Raises**: Exceptions.InvalidResource

Exceptions.InvalidUser

Exceptions.InvalidRequest

**HS\_SecurityCore.QueueAuthorizationRequest**(*OperationType, UserID, ResourceGUID*) - > Boolean

This function is used by the Security Web Service to queue an authorization request within the Security Database.

**Parameters**: OperationType – the type of operation to be performed (e.g., read, create, delete)

UserID – the ID of the requestor from the Security Database User List

ResourceGUID – the GUID of the resource for which access is requested from the Security Database Resource List

**Returns**: True or False

**ReturnType**: Boolean

**Raises**: Exceptions.InvalidResource

Exceptions.InvalidUser

Exceptions.InvalidRequest

**HS\_SecurityCore.RegisterUser**(*UserProfile*) - > Boolean

This function is uses by the Security Web Service to add a user profile to the Security Database User List.

**Parameters**: UserProfile – a list of attributes describing the user to be added to the User List.

**Returns**: True or False

**ReturnType**: Boolean

**Raises**: Exceptions.InvalidProfile

Exceptions.InvalidRequest

**HS\_SecurityCore.ResolveResource**(*ResourceParametersList*) - > ResourceInfoList

This function is used by the HydroServer services to resolve which specific resources listed in the Security Database Resource List (as identified by their ResourceGUIDs) match the resource Parameters in the ResourceParametersList (e.g., which specific data series match the SiteCode and VariableCode passed to a WaterOneFlow GetValues request).

**Parameters**: ResourceParametersList – a list of query parameters that are evaluated against the Security Database Resource List to resolve the specific resources that match the criteria and their ResourceGUIDs

**Returns**: A list of resources from the Security Database Resource List that matches the specified parameters in the ResourceParametersList

**ReturnType**: ResourceInfoList

**Raises**: Exceptions.InvalidRequest

**HS\_Security.ResolveUser**(*UserProfileList*) - > UserInfoList

This function is used by the Security Service to retrieve information about a user from the Security Database User List, including the UserID.

**Parameters**: UserProfileList – A list of user profiles

**Returns**: A list of users from the Security Database User List, including the UserIDs

**ReturnType**: UserInfoList

**Raises**: Exceptions.InvalidProfile

Exceptions.InvalidRequest

**HS\_SecurityCore.SetAccess**(*UserID, ResourceGUID, AccessRule*) - > Boolean

This function is used by the Security Service to create new access control rules in the Security Database.

**Parameters**: UserID – the UserID for a user in the Security Database User List

ResourceGUID – the ResourceGUID for a data resource in the Security Database Resource List

AccessRule – the access control rule to apply (e.g., grant read access, deny read access)

**Returns**: True or False

**ReturnType**: Boolean

**Raises**: Exceptions.InvalidUser

Exceptions.InvalidResource

Exceptions.InvalidRequest

**HS\_SecurityCore.ValidateCertificate**(*Certificate*) - > Boolean

This function is used by HydroServer services to determine whether a security certificate that has been passed with a request is valid and was issued by the HIS Central CA.

**Parameters**: Certificate – a security certificate issued by the HIS Central CA

**Returns**: True or False

**ReturnType**: Boolean

**Raises**: Exceptions.InvalidRequest

## 4.3. WaterOneFlow Web Service

The WaterOneFlow Web Service enables data consumers to retrieve metadata and data stored in an ODM database on a HydroServer.

Table ??. Web service methods defined in the HydroServer WaterOneFlow Service (HS\_WaterOneFlow).

|  |  |
| --- | --- |
| **Web Service Method** | **Parameters -> Return** |
| GetSiteInfo() | (SiteCode, Certificate) -> String |
| GetSiteInfoMultipleObject() | (SiteCode(), Certificate) -> GetSiteInfoMultipleObjectResponse |
| GetSiteInfoObject() | (SiteCode, Certificate) -> GetSiteInfoObjectResponse |
| GetSites() | (SiteCode(), Certificate) ->String |
| GetSitesByBoxObject() | (WestLong, SouthLat, EastLong, NorthLat, IncludeSeries, Certificate) -> GetSitesByBoxObjectResponse |
| GetSitesObject() | (SiteCode(), Certificate) -> GetSitesObjectResponse |
| GetValues() | (SiteCode, VariableCode, StartDate, EndDate, Certificate) -> String |
| GetValuesForASiteObject() | (SiteCode, StartDate, EndDate, Certificate) -> GetValuesForASiteObjectResponse |
| GetValuesObject() | (SiteCode, VariableCode, StartDate, EndDate, Certificate) -> TimeSeriesResponse |
| GetVariableInfo() | (VariableCode, Certificate) -> String |
| GetVariableInfoObject() | (VariableCode, Certificate) -> VariablesResponse |
| GetVariables() | (Certificate) -> String |
| GetVariablesObject() | (Certificate) -> GetVariablesObjectResponse |