



|  |
| --- |
| X-Road: Security Server Configuration  **Data Model** |
| **Version: 1.2**  **11.12.2015**  **23 pages**  **DM-SS** |

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
| 11.06.2015 | 0.1 | Initial version | Mait Märdin, Margus Freudenthal |
| 21.08.2015 | 0.4 | Added sections related to saving database history | Mait Märdin |
| 31.08.2015 | 0.5 | Removed backend URL that was removed from code | Margus Freudenthal |
| 20.09.2015 | 1.0 | Editorial changes made | Imbi Nõgisto |
| 19.10.2015 | 1.1 | Indexes added | Martin Lind |
| 11.12.2015 | 1.2 | Small fixes | Siim Annuk |

Table of Contents

[License 5](#_Toc438061475)

[1 General 6](#_Toc438061476)

[1.1 Preamble 6](#_Toc438061477)

[1.2 Database Version 6](#_Toc438061478)

[1.3 Creating, Backing Up and Restoring the Database 6](#_Toc438061479)

[1.4 Saving Database History 7](#_Toc438061480)

[1.5 Entity-Relationship Diagram 7](#_Toc438061481)

[1.6 List of Stored Procedures 9](#_Toc438061482)

[1.7 List of Triggers 9](#_Toc438061483)

[2 Description of Entities 10](#_Toc438061484)

[2.1 ACCESSRIGHT 10](#_Toc438061485)

[2.1.1 Indexes 10](#_Toc438061486)

[2.1.2 Attributes 10](#_Toc438061487)

[2.2 CERTIFICATE 10](#_Toc438061488)

[2.2.1 Attributes 10](#_Toc438061489)

[2.3 CLIENT 11](#_Toc438061490)

[2.3.1 Indexes 11](#_Toc438061491)

[2.3.2 Attributes 12](#_Toc438061492)

[2.4 DATABASECHANGELOG 12](#_Toc438061493)

[2.4.1 Attributes 13](#_Toc438061494)

[2.5 DATABASECHANGELOGLOCK 14](#_Toc438061495)

[2.5.1 Attributes 14](#_Toc438061496)

[2.6 GROUPMEMBER 14](#_Toc438061497)

[2.6.1 Indexes 15](#_Toc438061498)

[2.6.2 Attributes 15](#_Toc438061499)

[2.7 HISTORY 15](#_Toc438061500)

[2.7.1 Attributes 15](#_Toc438061501)

[2.8 IDENTIFIER 16](#_Toc438061502)

[2.8.1 Attributes 16](#_Toc438061503)

[2.9 LOCALGROUP 16](#_Toc438061504)

[2.9.1 Indexes 17](#_Toc438061505)

[2.9.2 Attributes 17](#_Toc438061506)

[2.10 SERVERCONF 17](#_Toc438061507)

[2.10.1 Indexes 17](#_Toc438061508)

[2.10.2 Attributes 17](#_Toc438061509)

[2.11 SERVICE 17](#_Toc438061510)

[2.11.1 Indexes 18](#_Toc438061511)

[2.11.2 Attributes 18](#_Toc438061512)

[2.12 SERVICE\_SECURITYCATEGORIES 18](#_Toc438061513)

[2.12.1 Indexes 18](#_Toc438061514)

[2.12.2 Attributes 19](#_Toc438061515)

[2.13 TSP 20](#_Toc438061516)

[2.13.1 Indexes 20](#_Toc438061517)

[2.13.2 Attributes 20](#_Toc438061518)

[2.14 UIUSER 20](#_Toc438061519)

[2.14.1 Attributes 20](#_Toc438061520)

[2.15 WSDL 20](#_Toc438061521)

[2.15.1 Indexes 21](#_Toc438061522)

[2.15.2 Attributes 22](#_Toc438061523)

# License

This work is licensed under the Creative Commons Attribution-ShareAlike 3.0 Unported Li-cense. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/.

# General

## Preamble

This document describes database model of X-Road security server.

## Database Version

This database assumes PostgreSQL version 9.3. Ubuntu 14.04 default settings are used.

## Creating, Backing Up and Restoring the Database

This database is integrated into X-Road security server application. The database management functions are embedded into the application user interface.

The database, the database user and the data model is created by the application's installer. The database updates are packaged as application updates and are applied when the application is upgraded. From the technical point of view, the database structure is created and updated using liquibase[[1]](#footnote-1) tool. The migration scripts can be found both in application source and in file system of the installed application.

Database backup functionality is built into the application. The backup operation can be invoked from the web-based user interface or from the command line. The backup contains dump of all the database structure and contents. When restoring the application, first the software is installed and then the configuration database is restored together with all the other necessary files. This produces a working security server.

Note: backing up of security server does not include message log that is managed using different tools.

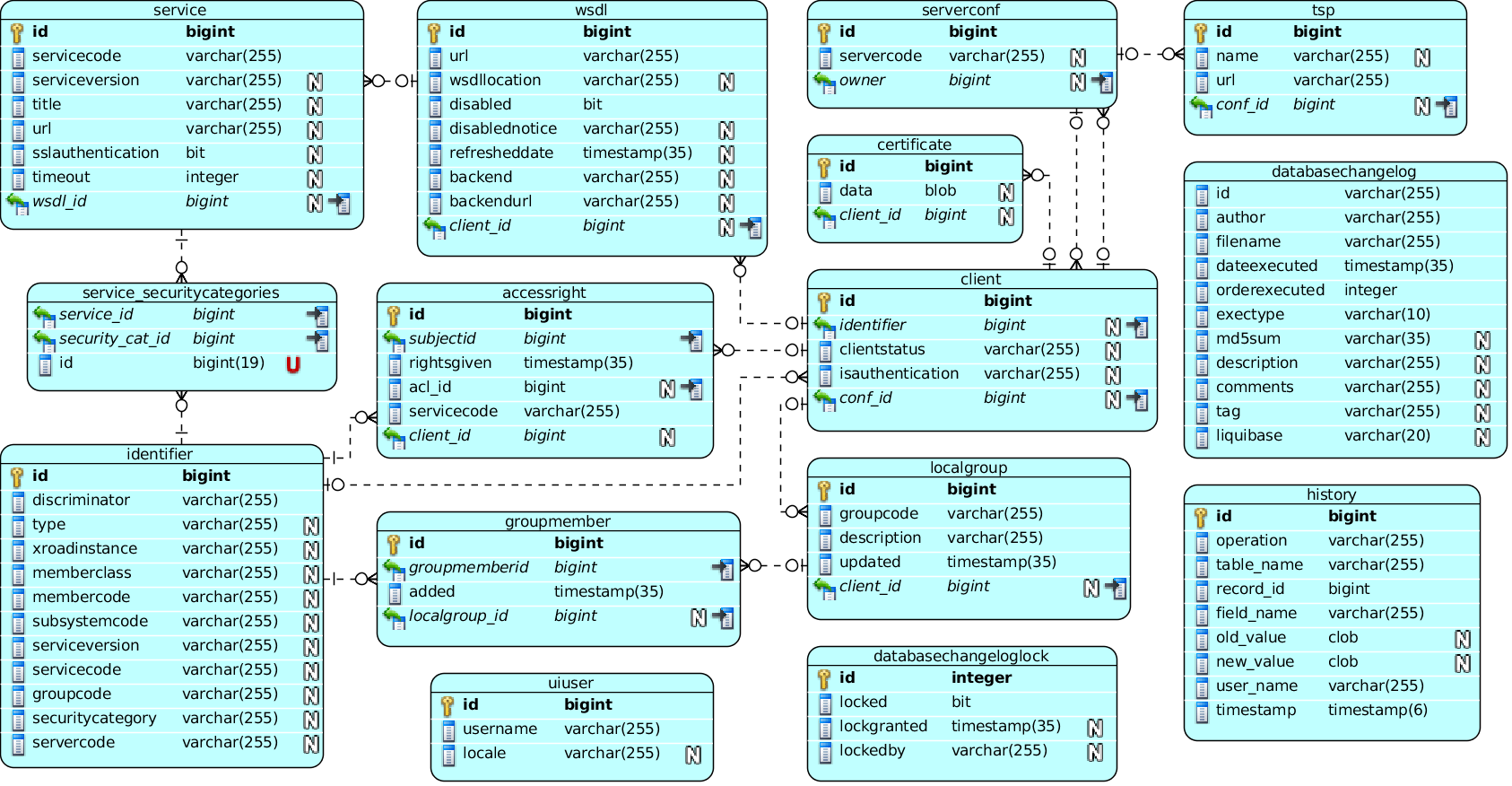
## Saving Database History

This section describes a general mechanism for storing history of the database tables. All the history-aware tables have associated trigger *update\_history* that records all the modifications to data. All the tables of security server database are history-aware, except for

* **history**,
* **databasechangelog** and
* **databasechangeloglock**.

When a row is created, updated or deleted in one of the history-aware tables, the trigger *update\_history* is activated and invokes the stored procedure *add\_history\_rows.* For each changed column, *add\_history\_rows* inserts a row into the history table. The details of the stored procedures are described in section 1.6.

## Entity-Relationship Diagram



## List of Stored Procedures

1. **add\_history\_rows**: Detects the changes made as a result of the operation it is invoked on, and calls the **insert\_history\_row** stored procedure to insert a row to the history table, for each changed field. For insertions and deletions, a history record is inserted for each field of the original table.
2. **insert\_history\_row**: Inserts a single row with values corresponding to a changed field in one of the database tables. Invoked by the **add\_history\_rows** stored procedure.

## List of Triggers

1. **update\_history**: Invokes the **add\_history\_rows** stored procedure upon insertions, updates and deletions of records. Created for each history-aware table.

# Description of Entities

## ACCESSRIGHT

Access right of a security server client or a group of clients to use a particular service. An access right record is created when an access right for a service is granted. The record is deleted when the service is removed from the system configuration or the access right is forfeited. The record is never modified.

### Indexes

|  |  |
| --- | --- |
| **Name** | **Columns** |
| AUTHORIZEDSUBJECT\_ACL\_ID\_fkey | acl\_id |
| AUTHORIZEDSUBJECT\_SUBJECTID\_fkey | subjectid |

### Attributes

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Modifiers** | **Description** |
| id [PK] | bigint | NOT NULL | Primary key. |
| servicecode | character varying(255) | NOT NULL | The service code part of the service identifier. |
| subjectid [FK] | bigint | NOT NULL | Identifier of a subject that is authorized to access the service. Can be either a member, a subsystem, global group or local group. References **id** attribute of **IDENTIFIER** entity. |
| rightsgiven | timestamp without time zone | NOT NULL | The time when the access right was granted. |
| service\_provider\_id [FK] | bigint |  | The security server client who provides the service. References **id** attribute of **CLIENT** entity. |

## CERTIFICATE

Trusted authentication certificate associated with an information system belonging to a particular security server client. A certificate record is created when a certificate is uploaded for a security server client. The record is deleted when the certificate is deleted from the system configuration. The record is never modified.

### Attributes

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Modifiers** | **Description** |
| id [PK] | bigint | NOT NULL | Primary key. |
| data | oid |  | X.509 public key certificate in binary DER form. |
| client\_id [FK] | bigint |  | The security server client whose information system server uses this authentication certificate. References **id** attribute of **CLIENT** entity. |

## CLIENT

Member or subsystem that is using this security server. The security server owner is also registered as a client.

For owner, the record is created when the administrator initializes the security server. For security server users, the record is created when the administrator adds new client in the user interface.

The client record is deleted when the administrator removes the client in the user interface. The client record corresponding to the owner cannot be deleted.

The client record is modified when administrator changes parameters in the user interface or when automatic status update occurs (see below).

The field clientstatus shows the progress of registering in central server the connection between this security server client and this security server. Only in “registered” state can the security server exchange messages on behalf of this client.

* “saved” -- initial state. Client enters it immediately after creation. From this state the administrator can send registration request to the central server.
* “registration in progress” -- the administrator has successfully sent registration request to the central server. In this state the security server is waiting for approval of the client registration request. When the security server receives a global configuration that contains connection between the security server and the client, it enters the “registered” state.
* “registered” -- the registration request sent to the central server is approved and the connection between the client and the security server is registered in the global configuration. In this state the security server can exchange messages on behalf of the client.
* “deletion in progress” -- the security server has successfully sent client deletion request to the central server. From this state, the only possible action is to delete the client from security server configuration.
* “global error” -- the client was in state “registered”, but the connection between the client and the security server has been deleted from the global configuration. From this state the administrator can either wait for updated global configuration (in case the deletion was caused by an error), contact the systems administrator of the central server or delete the client.

### Indexes

|  |  |
| --- | --- |
| **Name** | **Columns** |
| CLIENT\_CONF\_ID\_fkey | conf\_id |
| CLIENT\_IDENTIFIER\_fkey | identifier |

### Attributes

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Modifiers** | **Description** |
| id [PK] | bigint | NOT NULL | Primary key. |
| identifier [FK] | bigint |  | Identifies the security server client. References **id** attribute of **IDENTIFIER** entity. |
| clientstatus | character varying(255) |  | Current status of the client. Possible values are “saved”, “registration in progress”, “registered”, “deletion in progress”, “global error” |
| isauthentication | character varying(255) |  | Type of HTTPS authentication that is used with the client's information systems. Possible values are the following.   * “NOSSL” -- the client can connect with HTTP or HTTPS protocol. For HTTPS connection, no authentication is used. * “SSLNOAUTH” -- the client can only connect with HTTPS protocol. No certificate-based authentication is used. * “SSLAUTH” -- the client can only connect with HTTPS protocol. The client must authenticate the connection with certificate. |

## DATABASECHANGELOG

Liquibase migration of the database. A record is created when the administrator updates the software package containing this database and the database structure needs to be modified. The record is never modified or deleted. This table has a technical nature and is not managed by X-Road application software.

### Attributes

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Modifiers** | **Description** |
| id | character varying(255) | NOT NULL | The identifier of the migration. |
| author | character varying(255) | NOT NULL | The author of the migration. |
| filename | character varying(255) | NOT NULL | The filename containing the migration script. |
| dateexecuted | timestamp with time zone | NOT NULL | The time when the migration was executed. Used with orderexecuted to determine rollback order. |
| orderexecuted | integer | NOT NULL | The order number in which the migration was executed. Used in addition to dateexecuted to ensure order is correct even when the databases datetime supports poor resolution. |
| exectype | character varying(10) | NOT NULL | The type of the execution that was performed. Possible values are EXECUTED, FAILED, SKIPPED, RERAN, and MARK\_RAN. |
| md5sum | character varying(35) |  | The MD5 hash of the migration script when it was executed. Used on each run to ensure there have been no unexpected changes to the migration script. |
| description | character varying(255) |  | Short auto-generated human readable description of the migration. |
| comments | character varying(255) |  | The comments of the migration. |
| tag | character varying(255) |  | The tag of the migration. |
| liquibase | character varying(20) |  | The version of the Liquibase that performed the migration. |

## DATABASECHANGELOGLOCK

Lock used by Liquibase to allow only one migration of the database to run at a time. This table has a technical nature and is not managed by X-Road application software.

### Attributes

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Modifiers** | **Description** |
| id [PK] | integer | NOT NULL | Primary key. Id of the lock. Currently there is only one lock. |
| locked | boolean | NOT NULL | Set to "1" if the Liquibase is running against this database. Otherwise set to "0". |
| lockgranted | timestamp with time zone |  | Date and time when the lock was granted. |
| lockedby | character varying(255) |  | Human-readable description of who the lock was granted to. |

## GROUPMEMBER

Member of a local group. A group membership record is created when the administrator adds a new subsystem to a local group. The record is deleted when the administrator removes the subsystem from the local group. The record is never modified.

### Indexes

|  |  |
| --- | --- |
| **Name** | **Columns** |
| GROUPMEMBER\_GROUPMEMBERID\_fkey | groupmemberid |
| GROUPMEMBER\_LOCALGROUP\_ID\_fkey | localgroup\_id |

### Attributes

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Modifiers** | **Description** |
| id [PK] | bigint | NOT NULL | Primary key. |
| groupmemberid [FK] | bigint | NOT NULL | Identifier of the member or the subsystem who is a member of the local group. References **id** attribute of **IDENTIFIER** entity. |
| added | timestamp with time zone | NOT NULL | The time when the group member was added. |
| localgroup\_id [FK] | bigint |  | The local group. References **id** attribute of **LOCALGROUP** entity. |

## HISTORY

Operations (insertions, updates and deletions of records) on the tables of this database, for the purpose of auditing. Each record corresponds to the change of a single field. The record is created in the manner described in section 1.4. The record is never modified or deleted.

### Attributes

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Modifiers** | **Description** |
| id [PK] | bigint | NOT NULL | Primary key. |
| operation | character varying(255) | NOT NULL | Name of the database operation (possible values are INSERT, UPDATE, DELETE). |
| table\_name | character varying(255) | NOT NULL | Name of the table the operation was made on. |
| record\_id | bigint | NOT NULL | ID of the record that was inserted, updated or deleted, in the original table. |
| field\_name | character varying(255) | NOT NULL | Name of the column that was inserted, updated or deleted. |
| old\_value | text |  | Previous value of the column if applicable (NULL for INSERT operations). |
| new\_value | text |  | New value of the column if applicable (NULL for DELETE operations). |
| user\_name | character varying(255) | NOT NULL | Name of either the logged in user of the UI or the database user behind the connection, that initiated the operation. |
| timestamp | timestamp without time zone | NOT NULL | Date and time of the operation. |

## IDENTIFIER

Identifier that can be used to identify various objects on X-Road. An identifier record is only created together with records of other entities and only one record of each identifier is ever created. For example, if a security server client record is created and its identifier is not found among identifier records, new one is created. The record is never modified or deleted.

### Attributes

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Modifiers** | **Description** |
| id [PK] | bigint | NOT NULL | Primary key. |
| discriminator | character varying(255) | NOT NULL | Technical attribute, specifying the Java class to which the identifier is mapped. Possible values are C (ClientId), S (ServiceId), CS (CentralServiceId), GG (GlobalGroupId), LG (LocalGroupId), SC (SecurityCategoryId), SS (SecurityServerId). The corresponding Java classes are located in the ee.ria.xroad.common.identifier package. |
| type | character varying(255) |  | Specifies the type of the object that the identifier identifies. Possible values, defined in enum *ee.ria.xroad.common.identifier.XroadObjectType*, are MEMBER, SUBSYSTEM, SERVICE, CENTRALSERVICE, GLOBALGROUP, LOCALGROUP, SERVER, SECURITYCATEGORY. |
| xroadinstance | character varying(255) |  | X-Road instance identifier. Present in identifiers of all types, except LOCALGROUP. |
| memberclass | character varying(255) |  | Member class. Present in identifiers of MEMBER, SUBSYSTEM, SERVER and SERVICE type. |
| membercode | character varying(255) |  | Member code. Present in identifiers of MEMBER, SUBSYSTEM, SERVER and SERVICE type. |
| subsystemcode | character varying(255) |  | Subsystem code. Present in identifiers of SUBSYSTEM and SERVICE type. |
| serviceversion | character varying(255) |  | Service version. Present in identifiers of SERVICE type. |
| servicecode | character varying(255) |  | Service code. Present in identifiers of SERVICE type. |
| groupcode | character varying(255) |  | Group code. Present in identifiers of GLOBALGROUP and LOCALGROUP type. |
| securitycategory | character varying(255) |  | Security category. Present in identifiers of SECURITYCATEGORY type. |
| servercode | character varying(255) |  | Security server code. Present in identifiers of SERVER type. |

## LOCALGROUP

Group of members and/or subsystems. The group is local to a security server client and is used in access rights management. Local groups are connected to a security server client and can only be used for services belonging to that client. A local group record is created when the administrator adds a new local group to a security server client. The record is modified when the administrator changes the description of the group. The record is deleted when the administrator deletes the group or the security server client for whom the group is defined.

### Indexes

|  |  |
| --- | --- |
| **Name** | **Columns** |
| LOCALGROUP\_CLIENT\_ID\_fkey | client\_id |

### Attributes

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Modifiers** | **Description** |
| id [PK] | bigint | NOT NULL | Primary key. |
| groupcode | character varying(255) | NOT NULL | The code of the group. |
| description | character varying(255) | NOT NULL | The description of the group. |
| updated | timestamp with time zone | NOT NULL | The time when the description of the group was last updated. |
| client\_id [FK] | bigint |  | The security server client for whom the local group is defined. References **id** attribute of **CLIENT** entity. |

## SERVERCONF

The top-level configuration of the security server, specifying the owner and the code of this security server. This table contains only one record that is created when the security server is initialized. The record is never modified or deleted.

### Indexes

|  |  |
| --- | --- |
| **Name** | **Columns** |
| SERVERCONF\_OWNER\_fkey | owner |

### Attributes

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Modifiers** | **Description** |
| id [PK] | bigint | NOT NULL | Primary key. |
| servercode | character varying(255) |  | The code of this security server. |
| owner [FK] | bigint |  | The security server client who is the owner of this security server. References **id** attribute of **CLIENT** entity. |

## SERVICE

Service provided by a security server client. A service record is created when the administrator adds or refreshes a WSDL of a security server client, and a new service description is found in the WSDL. The record is modified if the administrator edits the service parameters in the user interface. The record is deleted when the administrator deletes the WSDL containing the service description or when the administrator deletes the security server client owning the WSDL.

### Indexes

|  |  |
| --- | --- |
| **Name** | **Columns** |
| SERVICE\_WSDL\_ID\_fkey | wsdl\_id |

### Attributes

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Modifiers** | **Description** |
| id [PK] | bigint | NOT NULL | Primary key. |
| servicecode | character varying(255) | NOT NULL | The code of the service. |
| serviceversion | character varying(255) |  | The version of the service. |
| title | character varying(255) |  | The title of the service. |
| url | character varying(255) |  | The URL of the service. |
| sslauthentication | boolean |  | A flag indicating whether the certificate of the service provider should be verified for SSL/TLS connections. NULL value is interpreted as true. Trusted service provider certificates are stored as **CERTIFICATE** entities. |
| timeout | integer | T | The maximum time in seconds that the service provider can take to respond to a query. |
| wsdl\_id [FK] | bigint |  | The WSDL of which this service is part of. References **id** attribute of **WSDL** entity. |

## SERVICE\_SECURITYCATEGORIES

Security category of a service. Currently security categories is a disabled feature and services cannot be assigned security categories.

### Indexes

|  |  |
| --- | --- |
| **Name** | **Columns** |
| SERVICE\_SECURITYCATEGORIES\_SECURITY\_CAT\_ID\_fkey | security\_cat\_id |
| SERVICE\_SECURITYCATEGORIES\_SERVICE\_ID\_fkey | service\_id |
| service\_securitycategories\_id\_key | id |

### Attributes

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Modifiers** | **Description** |
| id [PK] | bigint | NOT NULL | Primary key. |
| service\_id [FK] | bigint | NOT NULL | The service for which the security category is assigned. References **id** attribute of **SERVICE** entity. |
| security\_cat\_id [FK] | bigint | NOT NULL | The identifier of the security category. References **id** attribute of **IDENTIFIER** entity. |

## TSP

Timestamping service provider (TSP) that is used by the security server to time-stamp messages stored in the message log. Only connection parameters to the TSP are included. The data needed for verifying time stamps is read from the global configuration. A TSP record is created when the administrator adds a new TSP in the user interface. The record is deleted when the administrator deletes the TSP in the user interface. The record is never modified.

### Indexes

|  |  |
| --- | --- |
| **Name** | **Columns** |
| TSP\_CONF\_ID\_fkey | conf\_id |

### Attributes

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Modifiers** | **Description** |
| id [PK] | bigint | NOT NULL | Primary key. |
| name | character varying(255) |  | The name of the TSP. Used for displaying in the user interface. |
| url | character varying(255) | NOT NULL | The URL of the TSP. The security server will send time-stamping request using HTTP POST method. |

## UIUSER

Preferences of the user interface user. A record is created when the user changes the user interface language for the first time. The record is modified on later changes to the language. The record is never deleted.

### Attributes

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Modifiers** | **Description** |
| id [PK] | bigint | NOT NULL | Primary key. |
| username | character varying(255) | NOT NULL | Name of the user who has customized their user interface language. |
| locale | character varying(255) |  | The preferred language code. Valid values are 'en' for English, and 'et' for Estonian. |

## WSDL

Pointer to a WSDL containing the descriptions of services provided by a security server client. A WSDL record is created when the administrator adds a new WSDL to a security server client in the user interface. The record is modified when the administrator refreshes, enables or disables the WSDL. The record is deleted when the administrator deletes the WSDL or the security server client owning the WSDL.

### Indexes

|  |  |
| --- | --- |
| **Name** | **Columns** |
| WSDL\_CLIENT\_ID\_fkey | client\_id |

### Attributes

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Modifiers** | **Description** |
| id [PK] | bigint | NOT NULL | Primary key. |
| url | character varying(255) | NOT NULL | The URL of the WSDL. The URL points to the information system of the security server client. |
| wsdllocation | character varying(255) |  | The location of the WSDL file in security server file system. Currently not used. |
| disabled | boolean | NOT NULL | A flag indicating whether the WSDL and all its services are disabled. |
| disablednotice | character varying(255) |  | The error message returned in response to a call to a service belonging to a disabled WSDL. |
| refresheddate | timestamp with time zone |  | The time when the WSDL was last refreshed. |
| client\_id [FK] | bigint |  | The security server client providing the services described in this WSDL. References **id** attribute of **CLIENT** entity. |

1. See http://www.liquibase.org/ [↑](#footnote-ref-1)