



|  |
| --- |
| X-Road Architecture  **Technical Specification** |
| Version: 1.4  20.12.2016  16 pages  Doc. ID: ARC-G |

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
| 02.06.2015 | 0.1 | Initial version | Margus Freudenthal |
| 03.07.2015 | 0.2 | Updated component diagram, added technology matrix, updated interface descriptions, added references | Margus Freudenthal |
| 08.09.2015 | 0.3 | Made editorial changes | Vello Hanson |
| 08.09.2015 | 0.4 | Added signed document download interface and UIs to component diagram, added components to deployment diagram | Margus Freudenthal |
| 17.09.2015 | 0.5 | Incorporated feedback from Vitali | Margus Freudenthal |
| 18.09.2015 | 1.0 | Editorial changes made | Imbi Nõgisto |
| 13.10.2015 | 1.1 | Incorporated feedback from Vitali | Ilja Kromonov |
| 28.10.2015 | 1.2 | Typos fixed | Siim Annuk |
| 16.12.2015 | 1.3 | Add environmental monitoring | Ilkka Seppälä |
| 20.12.2016 | 1.4 | Added operational monitoring | Kristo Heero |

Table of Contents

[1 Introduction 6](#_Toc470024509)

[1.1 Overview 6](#_Toc470024510)

[1.2 Design Goals 6](#_Toc470024511)

[1.3 References 7](#_Toc470024512)

[2 System Components 8](#_Toc470024513)

[2.1 Central Server 8](#_Toc470024514)

[2.2 Security Server 8](#_Toc470024515)

[2.3 Information System 9](#_Toc470024516)

[2.4 Time-Stamping Authority 9](#_Toc470024517)

[2.5 Certification Authority 9](#_Toc470024518)

[2.6 Configuration Proxy 10](#_Toc470024519)

[2.7 Operational Monitoring Daemon 10](#_Toc470024520)

[3 Protocols and Interfaces 11](#_Toc470024521)

[3.1 X-Road Message Protocol 11](#_Toc470024522)

[3.2 Protocol for Downloading Configuration 11](#_Toc470024523)

[3.3 Message Transport Protocol 11](#_Toc470024524)

[3.4 Service Metadata Protocol 12](#_Toc470024525)

[3.5 Download Signed Document 12](#_Toc470024526)

[3.6 Management Services Protocol 12](#_Toc470024527)

[3.7 OCSP Protocol 13](#_Toc470024528)

[3.8 Time-Stamping Protocol 13](#_Toc470024529)

[3.9 Security Server User Interface 13](#_Toc470024530)

[3.10 Central Server User Interface 14](#_Toc470024531)

[3.11 Store Operational Monitoring Data 14](#_Toc470024532)

[3.12 Operational Monitoring Query 14](#_Toc470024533)

[3.13 Operational Monitoring Protocol 14](#_Toc470024534)

[3.14 Operational Monitoring JMX 14](#_Toc470024535)

[3.15 Environmental Monitoring Protocol 14](#_Toc470024536)

[3.16 Environmental Monitoring JMX 14](#_Toc470024537)

[4 Technology Matrix 15](#_Toc470024538)

[5 Deployment View 16](#_Toc470024539)

License

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

# Introduction

## Overview

X-Road is a system for enabling secure communication between organizations. This document describes technical architecture of the X-Road core. The goal is to give general overview of the X-Road system and the components that it contains. Detailed description of components and protocols can be found in separate documents. For information on the processes implemented by the X-Road components, refer to the use case documentation.

## Design Goals

The following list contains main design goals and design decisions of the X-Road system.

* X-Road is **decentralized** – the data exchange happens directly between organizations. There are no intermediaries. If the two organizations have established secure connection, the continuous data exchange depends only on availability of the organizations and the network between them.
* **Ownership of data** – X-Road does not change ownership of data. The data owner (service provider) controls who can access particular services.
* **Availability** is a central concern – the protocols are designed so that there is no single bottleneck in the system. Additionally, no component should become a single point of failure.
* All the messages processed by the X-Road are usable as **digital evidence**. The technical solution must comply with requirements for digital seals according to eIDAS [EIDAS]. This implies support for secure signature creation devices (SSCDs).
* All the communication is implemented as **service calls** using the SOAP [SOAP] protocol. The services are described using the WSDL [WSDL] language.
* **Cross-border services** – it is possible for an organization to invoke services provided by an organization belonging to a different instance of X-Road.
* **Encapsulating the security protocol** – the security measures and the security protocol are encapsulated in standard components. The organizations are not required to implement security-related functionality for data exchange.
* **Standardization** – X-Road aims to standardize the communication protocol between organizations. This enables the organizations to connect to any number of service providers without implementing additional protocols. X-Road core does not perform protocol and data conversion. If necessary, these conversions can be performed by the organization's information system.
* **No predetermined roles** – once an organization has joined the X-Road infrastructure, it can act as both service client and service provider without having to perform any additional registration.
* **Two-level authentication** – X-Road core handles authentication and access control on the organization level. End-user authentication is performed by information system of the service client.

## References

ARC-CP X-Road: Configuration Proxy Architecture. Cybernetica AS.

ARC-CS X-Road: Central Server Architecture. Cybernetica AS.

ARC-SS X-Road: Security Server Architecture. Cybernetica AS.

ARC-MA X-Road: Monitoring Architecture

ARC-OPMOND X-Road: Operational Monitoring Daemon Architecture. Cybernetica AS.

BATCH-TS Freudenthal, Margus. Using Batch Hashing for Signing and Time-Stamping. Cybernetica Research Reports, T-4-20, 2013.

EIDAS EU Regulation No 910/2014 – Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC

OCSP X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP. Internet Engineering Task Force, RFC 6960, 2013.

PKCS10 Certification Request Syntax Standard. RSA Laboratories, PKCS #10.

PR-GCONF X-Road: Protocol for Downloading Configuration. Cybernetica AS.

PR-MSERV X-Road: Management Services Protocol. Cybernetica AS

PR-MESS X-Road: Profile of Messages. Cybernetica AS.

PR-MESSTRANSP X-Road: Message Transport Protocol. Cybernetica AS.

PR-META X-Road: Service Metadata Protocol. Cybernetica AS.

PR-OPMON X-Road: Operational Monitoring Protocol. Cybernetica AS.

PR-OPMONJMX X-Road: Operational Monitoring JMX Protocol. Cybernetica AS.

SOAP Simple Object Access Protocol (SOAP) 1.1, 2000.

TSP Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP). Intenet Engineering Task Force, RFC 3161, 2001.

UG-SIGDOC X-Road: Signed Document Download and Verification Manual. Cybernetica AS.

WSDL Web Services Description Language (WSDL) 1.1, 2001.

# System Components

Figure 1 shows the main components and interfaces of the X-Road system. The components that are not part of the X-Road core are shown on grey background. The components and the interfaces are described in detail in the following sections.

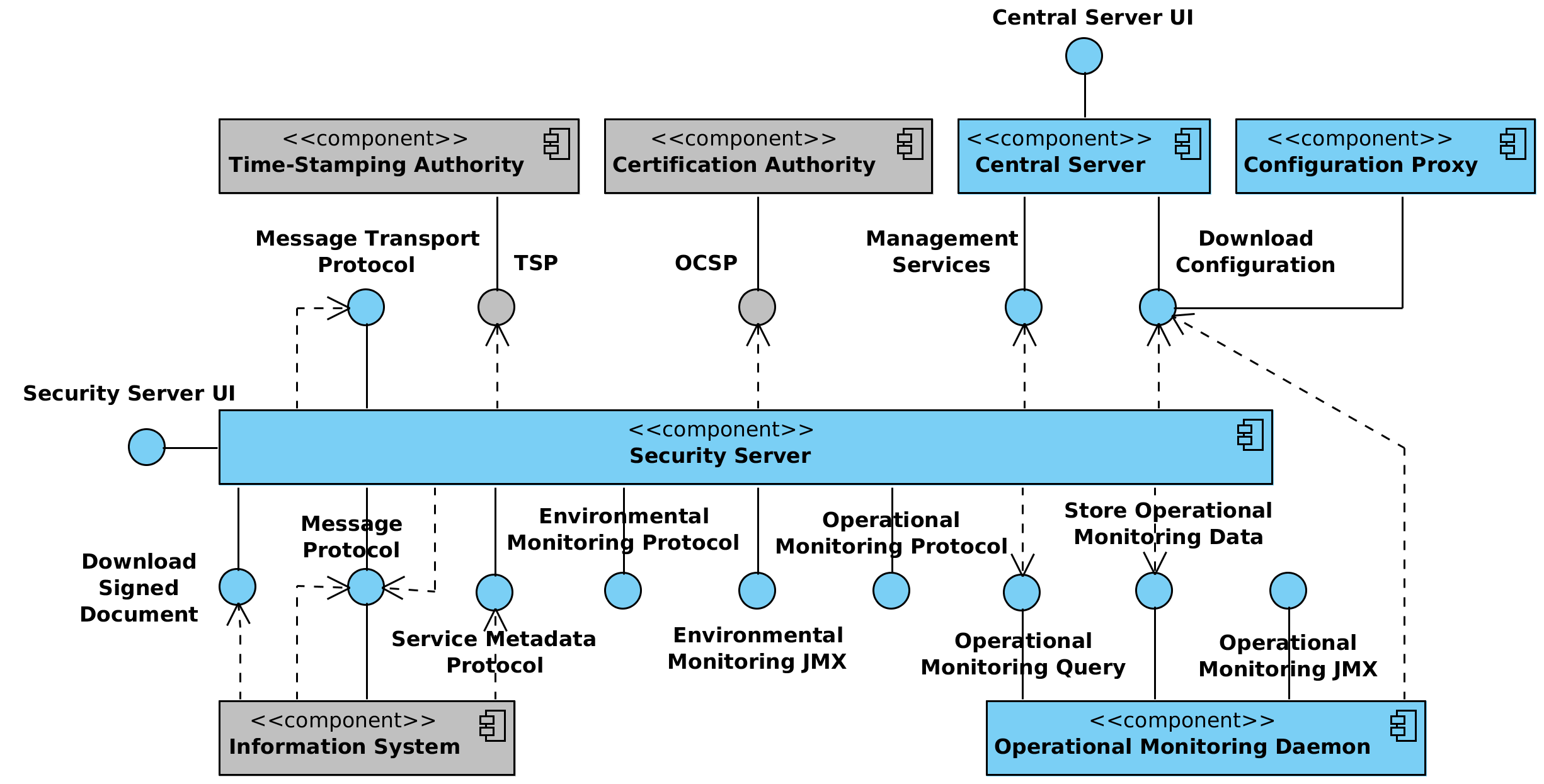


Figure 1. Logical structure of X-Road

## Central Server

Central server (see [ARC-CS] for details) manages the database of X-Road members and security servers. In addition, the central server contains the security policy of the X-Road instance. The security policy consists of the following items:

* list of trusted certification authorities,
* list of trusted time-stamping authorities,
* tunable parameters such as maximum allowed lifetime of an OCSP response.

Both the member database and the security policy are made available to the security servers via HTTP protocol (see Section 3.2 ). This distributed set of data forms the global configuration.

In addition to configuration distribution, the central server provides interface for performing management tasks such as adding and removing security server clients. These tasks are invoked from the user interface of the security servers. The management services are implemented as standard X-Road services and offered via central security server. See Section 3.6 for details.

## Security Server

The security server (see [ARC-SS] for details) mediates service calls and service responses between information systems. The security server encapsulates the security aspects of the X-Road infrastructure: managing keys for signing and authentication, sending messages over secure channel, creating the proof value for messages with digital signatures, time-stamping (see Section 3.8 ) and logging. For the service client and the service provider information system, the security server offers a SOAP-based protocol (see Section 3.1 ). This protocol is the same for both the client and the service provider, making the security server transparent to the applications.

A single security server can host several organizations (multi-tenancy). The organization managing the security server is the server owner, the hosted organizations are security server clients.

The security server manages two types of keys. The authentication keys are assigned to a security server and used for establishing cryptographically secure communication channels with the other security servers (see Section 3.3 ). The signing keys are assigned to the security server's clients and used for signing the exchanged messages. The keys can be stored either on hard disk (software token) or on an SSCD.

The security server downloads and caches up-to-date global configuration and certificate validity information (see Section 3.7 ). Caching allows the security server to operate even when the information sources are unavailable.

The security server contains an optional monitoring component that keeps track of environmental properties such as running processes, available disk space, installed packages etc. The monitoring component publishes this data via environmental monitoring service (see Section 3.11) and monitoring JMX (see Section 3.12) interfaces.

## Information System

The information system (IS) uses and/or provides services via the X-Road.

For the service client IS, the security server acts as an entry point to all the X-Road services (see Section 3.1 ). The client IS is responsible for implementing an user authentication and access control mechanism that complies with the requirements of the particular X-Road instance. The identity of the end user is made available to the service provider by including it in the SOAP message. The client can discover the X-Road members and available services by using the X-Road metadata protocol (see Section 3.4 ).

The service provider information system implements a SOAP service and makes it available over the X-Road. For this purpose, the service must conform to the X-Road message protocol (see Section 3.1 ). The service must be accompanied by the service description implemented in the WSDL language.

## Time-Stamping Authority

The time-stamping authority issues time stamps that certify the existence of data items at a certain point of time. The time-stamping authority must implement the time-stamping protocol described in Section 3.8 .

X-Road uses batch time-stamping (see [BATCH-TS]). This reduces the load of the time-stamping service. The load does not depend on the number of messages exchanged over the X-Road, instead it depends on the number of security servers in the system.

## Certification Authority

The certification authority (CA) issues certificates to security servers (authentication certificates) and to X-Road member organizations (signing certificates). All the certificates are stored in the security servers. The CA must be able to process certificate signing requests conforming to [PKCS10].

The CA must distribute certificate validity information via the OCSP protocol (see Section 3.7 ). The security servers cache the OCSP responses to reduce the load in the OCSP service and to increase availability. The load on the OCSP service depends on the number of certificates issued.

## Configuration Proxy

The configuration proxy (see [ARC-CP] for details) implements both the client part and the server part of the configuration distribution protocol (see Section 3.2 ). The configuration proxy downloads the configuration, stores it, and makes it available for download. Thus, the configuration proxy can be used to increase system availability by creating an additional configuration source and reduce load on the central server.

## Operational Monitoring Daemon

The main functionality of the operational monitoring daemon (see [ARC-OPMOND] for details) is to collect and store operational data of the X-Road security server and make it available for external monitoring systems via corresponding interfaces.

# Protocols and Interfaces

## X-Road Message Protocol

X-Road Message Protocol is used by service client and service provider information systems for communicating with the X-Road security server.

The protocol is a synchronous RPC style protocol that is initiated by the client IS or by the service provider's security server.

The X-Road Message Protocol is based on SOAP over HTTP(S) and adds additional header fields for identifying the service client and the invoked service. See [PR-MESS] for technical details.

This protocol (together with the Message Transport Protocol) forms the core of the X-Road data exchange. If the involved components are not available, then the data exchange is not possible. X-Road architecture makes possible to improve the availability of the involved components by using redundancy.

## Protocol for Downloading Configuration

Configuration clients download the generated global configuration files from the central server.

The configuration download protocol is a synchronous protocol that is offered by the central server. It is used by configuration clients such as security servers and configuration proxies.

The protocol is based on HTTP and MIME multipart messaging (see [PR-GCONF] for details). The configuration is signed by the central server to protect it against modification. Usually the configuration consists of several parts. The protocol allows configuration clients to check whether the configuration has changed and only download the modified parts.

X-Road security servers (and operational monitoring daemons) maintain a local copy of the global configuration, which they periodically update from their respective configuration source. This cached global configuration has a validity period, which, in general, is longer than the period at which configuration clients are configured to update their local copy. Security servers continue to be fully operational while the cached global configuration remains valid. However, an out-of-date copy of the global configuration severely restricts the management capabilities of security server administrators and forbids security servers from processing incoming requests. As such, a short downtime of the interface is permissible within the limits of the configured configuration validity period.

## Message Transport Protocol

The X-Road Message Transport Protocol is used by security server to exchange service requests and service responses.

The protocol is a synchronous RPC style protocol that is initiated by the security server of the service client.

The protocol is based on HTTPS and uses mutual certificate-based TLS authentication. The SOAP messages received from the client and the service provider IS are wrapped in MIME multipart message together with additional security-related data, such as signatures and OCSP responses. See [PR-MESSTRANSP] for details.

This protocol (together with X-Road message protocol) forms the core of the X-Road data exchange. If the involved components are not available, then the data exchange is impossible. X-Road architecture makes possible to improve the availability of the involved components by using redundancy.

## Service Metadata Protocol

The X-Road Service Metadata Protocol can be used by the service client information systems to gather information about the X-Road instance. In particular, the protocol can be used to find X-Road members, services offered by these members and the WSDL service descriptions.

The protocol is a synchronous RPC style protocol that is initiated by the service client IS.

Some of the information services are implemented as HTTP(S) GET requests to simplify client IS implementation. The other information services are called as standard X-Road services (see Section 3.1 ). See [PR-META] for further details.

The Service Metadata Protocol is used for client IS configuration and therefore the availability, throughput and latency of its implementing components are not critical to the functioning of the X-Road.

## Download Signed Document

The service for downloading signed documents can be used by the information systems to download signed containers from the security server's message log. In addition, the service provides a convenience method for downloading global configuration that can be used to verify the signed containers.

The protocol is a synchronous RPC-style protocol that is initiated by the IS. The service is implemented as HTTP(S) GET requests. See [UG-SIGDOC] for further details.

The Download Signed Document protocol is used by IS for downloading data stored in the security server and therefore the availability, throughput and latency of its implementing components are not critical to the functioning of the X-Road.

## Management Services Protocol

The management services are called by security servers to perform management tasks such as registering a security server client or deleting an authentication certificate.

The management service protocol is a synchronous RPC-style protocol that is offered by the central server. The service is called by security servers.

The management services are implemented as standard X-Road services (see Section 3.1 for details) that are offered by the organization managing the X-Road instance. The exception is the authentication certificate registration service that, for technical reasons, is implemented directly by the central server. Full details of the management services are described in [PR-MSERV].

In general, the management services are not critical to operation of X-Road and therefore their availability is not paramount. If the management services are unavailable, the security servers cannot manage their clients and authentication certificates. Some actions (such as removing clients and certificates) can be performed manually by central server administrator, without using the management services. The management service operations are not time-critical (the security server user explicitly chooses to send the management request and the user interface does not imply that this operation is instantaneous).

## OCSP Protocol

The OCSP protocol (see [OCSP]) is used by the security servers to query the validity information about the signing and authentication certificates.

OCSP protocol is synchronous protocol that is offered by the OCSP responder belonging to a certification authority.

In X-Road, each security server is responsible for downloading and caching the validity information about its certificates. The OCSP responses are sent to the other security servers as part of the message transport protocol (see Section 3.3 ). This ensures that the security servers do not need to discover the OCSP service used by the other party. Additionally, this arrangement supports the situation where access to the OCSP service is either restricted to certificate owners or is subject to charges.

The security servers never include nonce field in the OCSP request. This allows the OCSP service to employ various optimization strategies, such as pre-creating the OCSP responses.

Because OCSP responses are used in the process of certificate validation, failure of the OCSP service effectively disables X-Road message exchange. When the cached OCSP responses cannot be refreshed, the security servers are unable to communicate. Thus, the lifetime of the OCSP responses determines the maximum amount of time that the OCSP service can be unavailable. The lifetime is defined by the owner of the central server and can vary between different instances of X-Road.

## Time-Stamping Protocol

The Time-stamping protocol (see [TSP]) is used by security servers to ensure long-term proof value of the exchanged messages. The security servers log all the messages and their signatures. These logs are periodically time-stamped to create long-term proof.

Time-stamping protocol is a synchronous protocol that is provided by the time-stamp authority. However, the security servers use the time-stamping protocol in an asynchronous manner. Security servers log all the messages that are exchanged with other security servers. These messages are time-stamped asynchronously using batch time-stamping (see [BATCH-TS]). This is done to decouple availability of the message exchange from availability of the time-stamping authority, to decrease the latency of message exchange, and to reduce load on the time-stamping authority.

Because time-stamping is used in an asynchronous manner, temporary unavailability of the time-stamping service does not directly affect the X-Road message exchange. However, if the security servers fail to time-stamp the accumulated messages for certain time period then it may become difficult to prove the exact time of the message exchanges. To minimize this risk the security servers will stop forwarding messages if the time-stamping has been failing for some time. The maximum allowed time period between logging of a message and acquiring a time stamp for that message is defined by the owner of the central server and can very between different instances of X-Road.

## Security Server User Interface

The security server user interface is used by the security server administrator to configure and manage the security server.

## Central Server User Interface

The central server user interface is used by the central server administrator to configure and manage the central server.

## Store Operational Monitoring Data

This protocol is used by the X-Road security server to store its cached operational monitoring data into the database of the operational monitoring daemon. The protocol is a synchronous RPC-style protocol based on JSON over HTTP(S).

The interface is described in more detail in [ARC-OPMOND].

## Operational Monitoring Query

The operational monitoring query interface is used by the X-Road security server to retrieve operational monitoring data from the operational monitoring daemon. The asynchronous RPC-style X-Road operational monitoring protocol [PR-OPMONPR-OPMONPR-OPMONPR-OPMON] (based on [PR-MESS]) is used.

The interface is described in more detail in [ARC-OPMOND].

## Operational Monitoring Protocol

This interface is used by external monitoring systems to gather operational information of the security server. The protocol is synchronous RPC style protocol that is initiated by the external monitoring system. The protocol is described in more detail in [PR-OPMON].

## Operational Monitoring JMX

This interface is used by a local monitoring system (e.g. Zabbix) to gather local operational health data of the security server via JMXMP. The interface is described in more detail in [ARC-OPMOND] and [PR-OPMONJMX].

## Environmental Monitoring Protocol

The environmental monitoring interface responds to queries for monitoring environmental data from security server's serverproxy interface. The environmental monitoring data is collected by environmental monitoring service.

## Environmental Monitoring JMX

The environmental monitoring JMX service publishes environmental monitoring data via JMX interface. The environmental monitoring data is collected by environmental monitoring service.

# Technology Matrix

Table 1 presents the list of technologies used in the X-Road and mapping between the technologies and X-Road components.

Table 1. Technology matrix of the X-Road

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Technology** | **Security server** | **Central server** | **Configuration proxy** | **Operational Monitoring Daemon** |
| Java 8 | X | X | X | X |
| C | X | X |  |  |
| Logback | X | X | X | X |
| Akka 2.X | X | X |  | X |
| Jetty 9 | X | X |  |  |
| JRuby 1.7 | X | X |  |  |
| Ubuntu 14.04 | X | X | X | X |
| PostgreSQL 9.3 | X | X |  | X |
| PostgreSQL 9.4 |  | X[[1]](#footnote-1) |  |  |
| nginx | X | X | X |  |
| PAM | X | X |  |  |
| Liquibase | X |  |  | X |
| upstart | X | X | X | X |
| PKCS #11[[2]](#footnote-2) | X | X | X |  |
| Dropwizard Metrics | X |  |  | X |

# Deployment View

Figure 2 shows deployment view of a basic X-Road instance. In practice, all the components can use redundancy to improve availability and throughput. The deployment options for various components are described in the detailed architecture documents.

The diagram also shows what components are installed and hosted by any given organization. The governing authority installs and maintains central server and central security server. The configuration proxy is an optional component that is typically used for distributing configuration to federated X-Road instances. The service client and service provider organizations host their information system and security server that connects the information system to the X-Road.

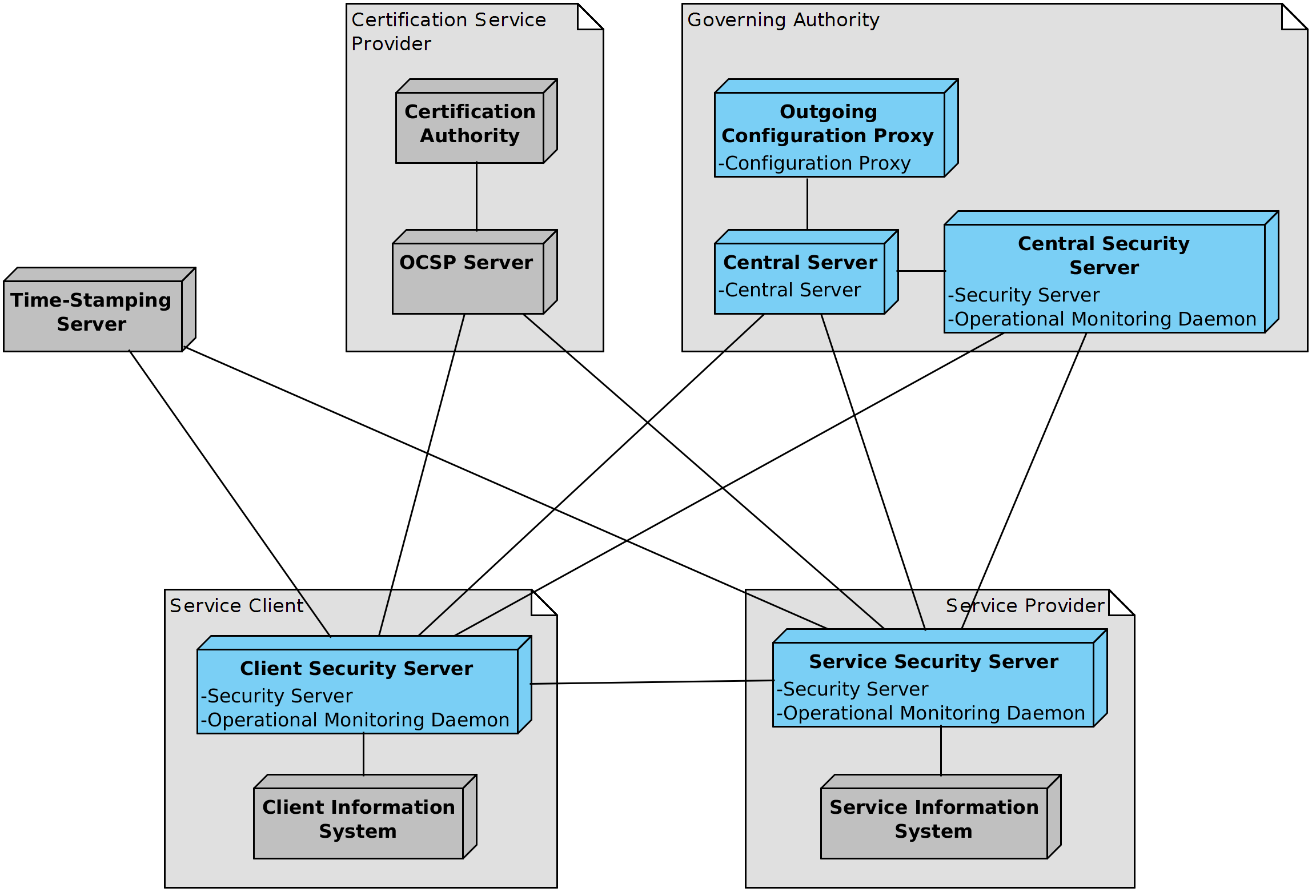


Figure 2. Deployment view of X-Road

1. PostgreSQL 9.4 is used in High-Availability installation of X-Road central server. [↑](#footnote-ref-1)
2. The use of hardware cryptographic devices requires that a PKCS #11 driver is installed and configured in the system. [↑](#footnote-ref-2)