

Sign Live! cloud suite gears

wp ais qseal

Oktober 2025

intarsys GmbH

Sign Live! cloud suite gears

wp ais qseal

Version 8.14

cloud suite gears

intarsys GmbH
Sign Live! cloud suite gears wp ais qseal
Version 8.14

All rights reserved
© 2021 intarsys GmbH
www.intarsys.de

Preface

- Author and company

This book has been provided by different authors from the development staff of intarsys GmbH.

- Trademarks

Wherever possible and where the authors were aware of a trademark claim, such designations are marked as trademarks in this book.

jPod is a trademark of intarsys consulting.

Sun, Java and JavaScript are trademarks of Oracle

Microsoft and Windows are trademarks of Microsoft Corporation.

- Who should read this book

This book provides both an overview of the product design and architecture and a reference for using the components and services.

So, this is the document for architects, developers and operators.

- Reviews and comments

We make constant efforts to improve our documentation and meet your requirements. Your comments are welcome and are a valuable resource for us.

Email support@intarsys.de

Website www.intarsys.de

Contents

Preface	5
▪ Author and company	5
▪ Trademarks	5
▪ Who should read this book	5
▪ Reviews and comments	5
Contents	6
1. Overview	9
2. Operational requirements	10
2.1 General requirements	10
2.2 Organizational requirements	10
2.2.1 IT requirements	10
2.2.2 Personal requirements	12
3. Solution design	13
3.1 Overview	13
3.2 Technical solution components	13
3.2.1 Sign Live! cloud suite gears	13
3.2.2 Authorization Service	14
3.2.3 Browser	14
3.2.4 Email Client	14
3.2.5 AIS	14
4. Operational approaches	15
4.1 Standard case	15
4.2 Special case: System administrator is legal representative of the key owner	15
5. System setup	16
5.1 Requirement fulfillment	16
5.2 Installation	17
5.2.1 Sign Live! cloud suite gears	17
5.2.2 Authorization Service	17
5.3 Configuration & Initialization	17

5.3.1	Master password	18
5.3.2	Activating the DB-backed keystore device	18
5.3.3	AIS device	18
5.3.4	Control panel 2FA	19
5.3.5	Creating access credentials	19
5.3.6	Disabling the gears control panel	20
5.4	Operation	21
5.4.1	General system operation	21
5.4.2	Access key control using the gears control panel	21
5.4.3	Access key control using direct server access	22
6.	External References	23

1. Overview

The Swisscom AIS remote signing service offers the creation of Qualified and Regulated Electronic Seals in accordance with EU and Swiss legislation (eIDAS / ZertES).

In order to leverage the benefits of these high-standard seals, an integrator has to fulfill special requirements, ensuring that the overall seal issuance and creation process complies with the rules and requirements stated in the CEN 419 241 standard.

Sign Live! cloud suite gears has been approved by Swisscom for operating Qualified and Regulated Seals in a customer environment. Hence the application provides methods, technical components and instructions for ensuring secure operation adhering to the underlying requirements and attestations.

This book describes the requirements for operation of AIS remote Qualified or Regulated Seals using gears and provides instructions, patterns and configuration for an adequate, secure application setup.

2. Operational requirements

2.1 General requirements

Access and use of the private key allowing the creation of a Qualified or Regulated Electronic Signature require sole control by the key owner, i.e. the applying organization. Using AIS, the private key itself is remotely managed by the trust service provider, i.e. Swisscom.

Within the technical communication protocol between the key owner's client application (i.e. gears) and the Swisscom-side signature service (AIS), Swisscom applies client-authenticated mTLS in order to verify the authenticity of a request and authorize access to the private key. The client TLS credentials used to authenticate comprise a private key and a certificate. In the following, we'll refer to these elements by the terms "access key" and "access certificate".

The access key is the only means of stating the permission to use the remotely managed private key. Thus, the access key has to be protected as if it were the private key used to seal user data itself. The protection requirements comprise:

- The access key must be protected.
- Use of the access key must be solely controlled by the private key owner.
- Two-factor-authentication (2FA) must be applied to ensure the authenticity of the person controlling the access key.
- Secure registration of the access key with the remote signing service.

2.2 Organizational requirements

2.2.1 IT requirements

2.2.1.1 IT components

The product shall only be used in an environment as described in the following. Supported IT components can be found in [1].

2.2.1.2 Network connectivity

Client and server environment must be adequately protected against threats from internal and external networks.

The product shall only be connected to a network in conjunction with a firewall, so that attacks from internal and external networks can be recognized and mitigated.

The operational environment has to be protected from malware, so that malicious software being installed on the respective systems is recognized and malicious behavior is mitigated.

2.2.1.3 Operational constraints

The product has to be operated in a protected environment. During operation, the following constraints must be respected for correct use:

Access control

The operator has full control over the storage media attached to the system.

Staff

Administrators of the system have to be trustworthy. They are aware of the contents of the documentation provided with the product and especially follow security requirements and guidelines. No other persons have access to the system.

API use

The APIs exposed by the system for seal creation must be protected against unauthorized use. The product provides a range of mechanisms and capabilities for this purpose, which can be taken from [1] and [2].

Connectivity

The product must be enabled to create an IP connection to the AIS signature service operated by Swisscom.

Protection against manipulation of the system

Organizational and technical measures must be taken in order to ensure that only authorized persons can configure and operate the system.

Control of the documents to be processed

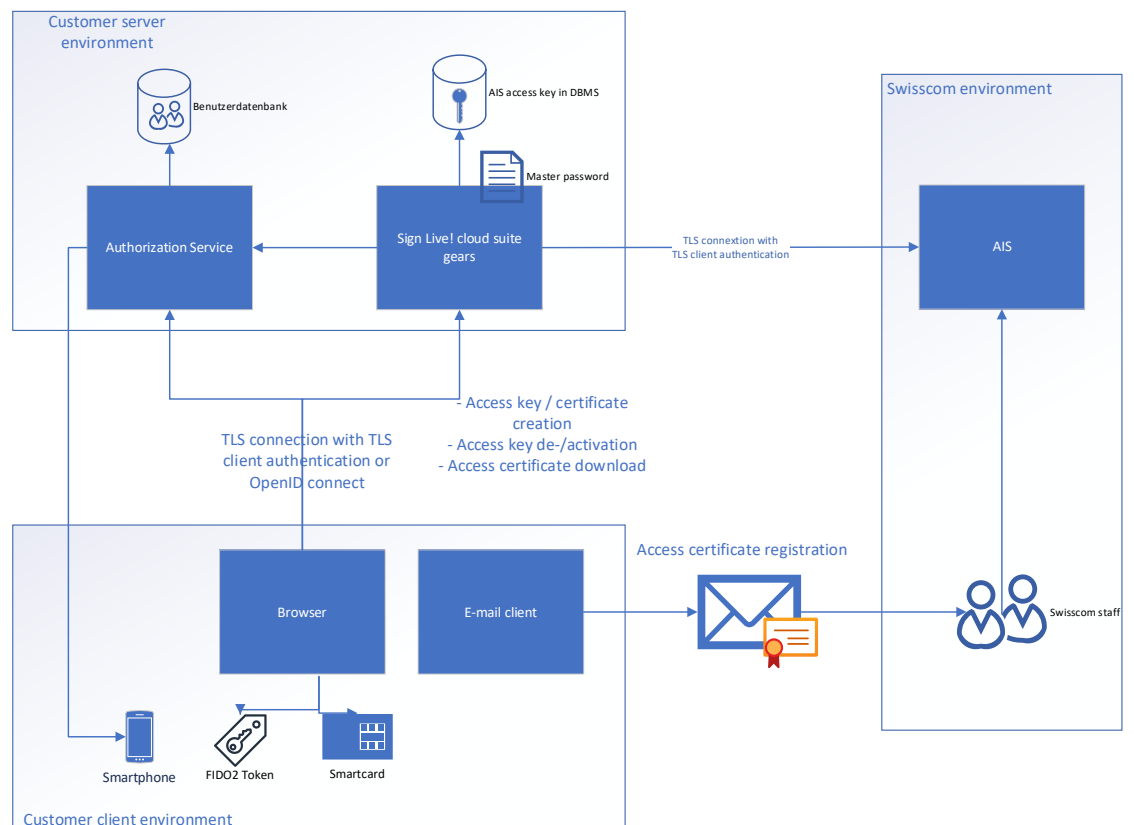
Organizational and technical measures must be taken in order to ensure that only such documents access the system through its APIs, which really may and shall be sealed. This can be enforced e.g. by permissions controlled by a delivering business application.

2.2.2 Personal requirements

The operating staff has to be aware of the contents of the documentation provided with the product and especially this document. Beyond, they have all technical and organizational knowledge required to setup and operate the operational environment according to the system requirements.

3. Solution design

3.1 Overview



3.2 Technical solution components

The following sections outline the relevant components.

3.2.1 Sign Live! cloud suite gears

Sign Live! cloud suite gears is the web application running in a customer's server environment, consisting of a backend application and UI components (single page applications).

General configuration is performed by means of system environment variables, Java system properties, properties files, XML configuration files and specific files (e.g. a master password file).

Data storage is performed in the file system and within a database (DBMS). By default, a file-based database (H2) is used, which is automatically created in the local file system. Alternatively, an external DBMS (e.g. MariaDB) can be used. See [1] for details on database connectivity.

3.2.2 Authorization Service

The authorization service is an optional system component allowing access control to protected operational and administrative actions within the gears UIs, especially the gears control panel.

It is not part of the product and thus presents an external dependency.

3.2.3 Browser

The user interacts with the user interface provided by gears and the authorization service through the local browser. See [1] for details on browser support.

3.2.4 Email Client

The user's email client is used to provide registration information (e.g. the access certificate) to Swisscom's registration authority. A secure message channel is not assumed.

3.2.5 AIS

AIS is the remote signing service provided by Swisscom. Gears authenticates with AIS using the operator's access key and certificate on an mTLS-secured HTTPS connection.

4. Operational approaches

Section 3 provided an overview of the overall system and relevant components. However, the final setup in the customer event doesn't need to match all of its elements and may vary due to differing personal assignments of responsibility and IT capabilities.

In general, we distinguish between two cases with corresponding preconditions:

- Standard case
- Special case: System administrator is legal representative of the key owner

4.1 Standard case

The standard case assumes a separation of responsibilities. It is assumed that the system administrator is not entitled to control the activation of the access key. This case allows for administrative and operative system access by authorized persons (e.g. log archival, system maintenance) without education and entitlement specific to seal creation.

4.2 Special case: System administrator is legal representative of the key owner

We have a special case, if the following preconditions are met:

- The system administrator (synonymous for all persons with system access to the server environment) is also legal representative of the applying organization or officially entitled by such a person. Thus, the system administrator has all permissions to control the use and activation of the access key.
- Access to the server environment is only possible after successful two-factor authentication (2FA). This includes the operating system and all storage media. (Example: smartcard-based Windows login)

5. System setup

5.1 Requirement fulfillment

In order to comply with the requirements stated in section 2, the following measures must be respected:

All configuration data is provided by a system administrator, using the configuration means described in [1].

A master password is set, which is individual to the gears installation and only known to the system administrator.

Any security-sensitive information which is placed by gears or the system administrator in the server environment's file system is encrypted. The encryption key is derived from the master key. Proper encryption of any textual information is performed through the password tool provided in the gears control panel using a cryptdec adhering to the stated rules (see [1]).

Security-sensitive information in this sense is:

- Database password
- Access credentials for an authorization service
- persistently stored passwords

The standard case (see section 4.1) requires the use of two-factor-authentication (2FA) for authentication to the gears control panel. 2FA can be achieved by

- TLS Client Authentication using an authentication certificate stored in a hardware device (e.g. a smartcard or USB token).
- integration of Keycloak, allowing for
 - WebAuthn using a FIDO2-capable token or end user device
 - smartphone-based authentication using time-based OTP (TOTP) and Google Authenticator
- integration of an individual authorization service which supports the OpenID connect (OIDC) protocol

2FA can be skipped for the control panel only in the special case (see section 4.2) and if the control panel is not used for access key activation. The gears control panel has thus to be made unreachable by server-side enforcement.

It is strongly recommended to use the control panel and thus provide the necessary security means. Otherwise, you'll have to abstain from all control panel functions (at least for now), which include the password protection tool, license introspection and more to come. (There may be some more fine-grained access control in the future, taking the burden of deactivating the control panel as a whole.) Beyond, the control panel allows you to deactivate AIS Qualified and Regulated seals during runtime, while still retaining the overall gears operation. Without the control panel, all gears services are taken offline – at least for a moment.

5.2 Installation

All components shown in section 3 are considered commodities and thus assumed to be available, with the exception of

- Sign Live! cloud suite gears
- Authorization service

5.2.1 Sign Live! cloud suite gears

Please install Sign Live! cloud suite gears following the instructions and constraints described in the section on “Installation” of [1].

5.2.2 Authorization Service

If no OIDC-compliant authorization service is available, we assume the installation of a dedicated Keycloak instance. Please follow the Keycloak Server Installation and Configuration Guide [3] for proper installation and setup.

Make sure that any login to the authorization service, which leads to a granted access to the gears control panel, enforces the use of two-factor-authentication!

5.3 Configuration & Initialization

Gears configuration requires the following steps:

1. Set a master password for the gears installation.
2. Start the system.
3. Activate the DB-backed keystore device.
4. Configure the AIS device.
5. Configure 2FA for the gears control panel (conditional).
6. Restart the system.
7. Create the AIS access credentials.
8. Disable the control panel, store the access key password and restart the system (conditional).

The system will be up and running after step 2, even though the configuration is not yet finished. Thus, whenever you need to store a password in your configuration, you can already apply the control panel's password tool (see section "Password Tool" of [1]) for proper password protection prior to inserting it into the configuration files.

The following sections give details on the configuration and initialization steps. Procedures for start and restart are considered well-known.

5.3.1 Master password

Setting an installation-specific master password is obligatory. Please refer to the section on configuring "Basic security" in [1] for details on changing the master password.

Note: It is important that you set the password **before** generating any keys or encrypted passwords, as the master password will be part of any protection mechanism.

5.3.2 Activating the DB-backed keystore device

The TLS client credentials will be stored in a database. Section "DB-backed keystore device" of [4] introduces two Spring profiles to be activated in your installation:

```
spring.profiles.active=builtinkeystore,dbkeystore
```

Refer to the configuration section on "Using profiles" in [1] for more information on setting Spring profiles.

Unless the default, file-based H2 database is used, refer to the section on "Data source" in [1] for proper setup of DB connectivity.

5.3.3 AIS device

You need to set up the AIS device for use of a SSLContext which applies TLS client credentials stored in the previously activated DB-backed keystore device.

Add an XML configuration file according to section "Custom bean definition" of [1] and place the following bean definition in there:

```

<bean id="aisSslContextProvider"
class="de.intarsys.tools.ssl.ConfigurableSslContextProvider">
  <property name="protocol" value="TLS" />
  <property name="trustAll" value="{ais.tls.trustAll:false}" />
  <property name="sslSessionTimeout" value="{ais.tls.sessionTimeout:60}" />
  <property name="keyManagerProvider" >
    <bean class="de.intarsys.security.app.ssl.DeviceBasedKeyManagerProvider">
      <property name="signerDevice" ref="databaseKeystoreDevice" />
      <property name="signerArgs">
        <map>
          <entry key="signerIdentifier"
            value="{ais.tls.keyIdentifier:usage=signature}" />
          <entry key="signerPassword"
            value="{ais.tls.keyPassword:}" />
        </map>
      </property>
    </bean>
  </property>
</bean>

```

This will override the standard AIS device configuration's *aisSslContextProvider*, so that all AIS signature requests are routed through this protocol component.

By default, this bean definition selects a key with usage signature from the database. The password is assumed to be given through the control panel. The session timeout of 60 seconds ensures that the connection state is reused for a maximum of 60 seconds and then renegotiated. Thus, changes to the access key's activation state (e.g. password cleared) will take effect after 60 seconds at the latest.

See section "TLS client authentication with signer device" of [4] for further information.

5.3.4 Control panel 2FA

If the control panel is used for later operation, you must guard the access to it and setup two-factor-authentication.

Refer to the section on "OAuth 2.0 login to gears control panel" of [4] for information on connecting an external authorization service, e.g. Keycloak, for this purpose.

Alternatively, you can use client TLS authentication based on smartcards or similar hardware tokens for achieving the 2FA requirement. Refer to the section on "API security" of [1] for a general introduction – but be prepared to setup a more complex reverse proxy scenario with TLS termination and multiple virtual hosts in this case.

5.3.5 Creating access credentials

The access credentials, i.e. the TLS client key and certificate, are generated through the gears control panel. The process is described in the section on "Principal creation" of [4]. Please consider the following rules when entering the certificate data:

Subject

At least the following fields shall be filled:

- Name
- Organization
- Country

Usage

The following usages shall be checked:

- Signature and authentication

Certificate Properties

A validity of 3 years should be chosen.

Protection

Use characters such that the password can be considered secure according to your organization's password policies.

Upon startup, the creation of the certificate may take about 10 seconds, so please be patient and don't leave the page. After the certificate has been created, you're redirected to the new principal's page. Download the resulting certificate (see device management section on "Principal" in [1]) and jot down the SHA-256 fingerprint shown in the page's certificate section. This certificate must be sent to Swisscom for registration with your AIS customer account.

Remember or securely store the password used to protect the private key.

The AIS device configured in section 5.3.3 will automatically use the access credentials created here, as long as there is **exactly one** principal.

Otherwise, you'll have to uniquely identify the principal to use in the device configuration. In this case, identify the certificate's serial number in the principal's certificate section and set the property "ais.tls.keyIdentifier" accordingly. Example:

```
ais.tls.keyIdentifier=serialnumber=123456
```

5.3.6 Disabling the gears control panel

In case the control panel is not used for key de-/activation and given that the requirements in section 4.2 are met, you can deactivate access to the control panel. The gears control panel will still be in operation, but by tweaking access control appropriately **within the gears configuration** it will be de facto inaccessible.

As there's no possibility to provide the key password at runtime without the control panel, you have to store the password in the server-side configuration. Encrypt it using the control panel's password tool and securely store the encrypted result string.

Add an XML configuration file according to section “Custom bean definition” of [1] and place the following definitions in there:

```
<security:authentication-manager id="securityRealmControlAuthenticationManager">
  <security:authentication-provider>
    <security:user-service>
      <security:user name="operator" password="{noop}" authorities="ROLE_OPERATOR" />
    </security:user-service>
  </security:authentication-provider>
</security:authentication-manager>

<bean id="securityRealmControlAuthenticationFilter"
  class="org.springframework.security.web.authentication.www.BasicAuthenticationFilter">
  <constructor-arg ref="securityRealmControlAuthenticationManager" />
</bean>
```

This will attach HTTP Basic Authentication to the control realm and an empty user base¹. Hence there'll be no possibility to access the system, as there are no users being authorized.

5.4 Operation

5.4.1 General system operation

General system operation is specific to the type of operating system (Windows, Linux) and the type of operation (Windows service, Linux service, Docker container, ...). Please consider the section on “Installation” of [1] for environmental constraints. Others than that, we assume that you're familiar with the operational environment.

5.4.2 Access key control using the gears control panel

This is the preferred means of controlling access key activation and deactivation.

5.4.2.1 Activating the access key

Activating the key is achieved by caching the corresponding password.

1. Open the control panel and log in.
2. Navigate to the “Device Providers > Keystore... > database” and select the principal created for TLS client authentication.
3. Request PIN cache and enter the password. You may select persistent storage **only if the preconditions in section 4.2 are met**.

5.4.2.2 Deactivating the access key

Deactivating the key is achieved by removing the corresponding password from the cache.

¹ An user with empty password is not available. At least one user is needed.

1. Open the control panel and log in.
2. Navigate to the “Device Providers > Keystore... > database” and select the principal created for TLS client authentication.
3. Clear PIN cache.

5.4.3 Access key control using direct server access

This is an alternative approach to controlling access key activation and deactivation. It assumes that **the preconditions in section 4.2 are met** and that the access key’s password is stored on the server-side as mentioned in section 5.3.6.

5.4.3.1 Activating the access key

Activation is achieved by statically setting the access key’s encrypted password in the gears properties. Assign it to the gears property “ais.tls.keyPassword”, adhering to the conventions on secret properties (see section “Secret” of [1]). Example:

```
ais.tls.keyPassword=${secret.aes-0#2lZ3Wbj6Mfb4Ue7KRwGudfaXvSu8L+sXEvwKipu7s+8=}
```

Upon gears startup, the access key will be active.

5.4.3.2 Deactivating the access key

Activation occurs automatically upon gears shutdown.

If gears overall operation shall be resumed without key activation, remove the property “ais.tls.keyPassword” and restart gears.

6. External References

- [1] intarsys GmbH, Sign Live! cloud suite gears manual.
- [2] intarsys GmbH, Sign Live! cloud suite gears wp security.
- [3] keycloak.org, "Server Installation and Configuration Guide," [Online]. Available: https://www.keycloak.org/docs/latest/server_installation/index.html.
- [4] intarsys GmbH, Sign Live! cloud suite gears incubator.
- [5] Swisscom AG, "All-in Signing Service Reference Guide," [Online]. Available: http://documents.swisscom.com/product/1000255-Digital_Signing_Service/Documents/Reference_Guide/Reference_Guide-All-in-Signing-Service-de.pdf.
- [6] intarsys GmbH, Sign Live! cloud suite gears cookbook.
- [7] intarsys GmbH, Sign Live! cloud suite gears tutorial.
- [8] intarsys GmbH, Sign Live! Security Applications Developers Guide.
- [9] intarsys GmbH, Sign Live! cloud suite gears cookbook.