Clearing Platform Specification

Non-functional Requirements

Author

conology GmbH

Contact:

Willm Tüting

Tel.: +49 (0) 175 520 96 02

willm.tueting@conology.net

Status of this document: 25.09.2023

Inhalt

1 Security

Scope

Interfaces (APIs) for data exchange of the systems involved in clearing, particularly the communication

- between two clearing platforms
- between a partner and a clearing platform

NOT in the scope of this description, and thus entirely the responsibility of each individual platform, are

- securing a platform when accessed via management interfaces (e.g. SSH),
- security when users access via a GUI. The use of TLS (see below) and secure authentication of users is required

1.1 Objectives

Description of the specifications and mechanisms to ensure the following properties when exchanging data between the systems involved in the clearing process:

A) Confidentiality

Data relevant for data protection (especially personal data), but also data relevant for business, SHALL be protected against inspection by uninvolved third parties.

Since the messages exchanged between the parties are sent over the public Internet, measures must be taken on the transport route to protect against unauthorized access.

B) Integrity / immutability

It MUST be ensured that no changes can be made to the transmitted messages during transport.

C) Authenticity

Incoming messages MUST be reliably and verifiably assigned to a sender on the receiver side.

D) Authorization

It must be ensured that only authorized communication partners can send and receive messages. For this purpose, the sender and receiver of messages must be able to verify each other's identity, and the receiver of a message must be able to decide on the authorization of a sender and, if necessary, prevent the acceptance of messages from unauthorized senders.

For incoming messages, the receiver MUST be able to verify whether the sender is authorized for the respective operation.

1.2 Measures

1.2.1 Encryption

The goals of confidentiality and integrity are achieved by encrypting all communication using TLS in HTTP communication.

When using TLS the **minimum standards of the BSI for the use of Transport Layer Security (TLS)** have to be considered.

Among other things, this requires that

- all transmitted messages MUST be encrypted using TLS
- the minimum TLS version required by all participants is 1.2 (1.0, 1.1 are outdated and insecure, see https://datatracker.ietf.org/doc/rfc8996/)
- permitted (but not mandatory) is the use of newer versions (1.3 or following)
- transmission of security relevant (e.g. access tokens) or personal data as part of a URL has to be excluded, because this data is vulnerable on different levels (logs, browser, ...)

Certificate

The X.509 certificates used for TLS connection establishment (TLS handshake) MUST fulfil the following properties

- signed by an authorized certificate authority, no self-signed certificates may be used. An explicit whitelist of certificate authorities is NOT maintained.
- Type: X.509 version 3
- Algorithm: "SHA256 with RSA"
- RSA key length (public key): at least 2048 bits
- Validity period: maximum 12 months

1.2.2 Authentication

Mutual authentication of provider and user is the prerequisite for checking the authenticity of messages and based on this, the authorization of requests.

1.2.2.1 Interface Provider

The required use of TLS implicitly authenticates the provider of an interface against the consumer (user). In order for a connection to be established, an X.509 certificate is presented by the provider, which contains the domain name of the provider's interface. As a result of this process, the consumer (user of the interface) can be sure that the provider is the owner of the domain that was used for the request to establish the connection.

1.2.2.2 Interface Consumer

In order for the provider of an interface to decide whether to accept messages from a user, it must be able to verify the user's identity with certainty.

The following procedures are supported by all platforms for this purpose:

- **A - Basic**: HTTP Basic Authentication. Simple implementation and sufficient security
- **B - Advanced**: OAuth2 Client Credentials Grant Flow + Bearer Tokens. Increased security by reducing the attack surface when credentials are transferred.

A partner MUST determine during onboarding which of these methods is supported by their system and should be used to communicate with the platform.

1.2.3 Authentication methods in detail

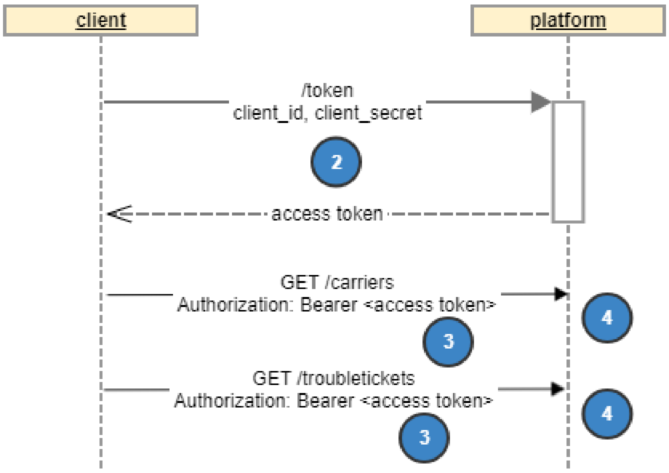
1.2.3.1 HTTP Basic

The definition of the procedure is standardized as RFC7617, see https://datatracker.ietf.org/doc/html/rfc7617

Each user of an interface receives a set of credentials from the provider, consisting of "username" and "password". These credentials must be included in every request sent by the user to the provider's interface. The provider can verify the authenticity of the messages by verifying the access data and assigning the request to a user (partner, platform) in a verifiable, secure and unambiguous manner.

1.2.3.2 OAuth 2.0 Client Credentials Grant Flow

The procedure follows the following scheme



1. each user of an interface receives a set of access credentials (client credentials) from the provider, consisting of "client id" and "client secret".
2. these credentials are used by the user to obtain an access token
3. in every request sent by the user to the provider's interface, the access token must be sent as well
4. the provider can assign the access token to a user (partner, platform) in a verifiably secure and unambiguous manner, and thus verify the authenticity of the messages.

1.2.3.2.1 Token creation

The OAuth 2.0 **Client Credentials Grant** (https://datatracker.ietf.org/doc/html/rfc6749#section-4.4) flow is used to obtain the access tokens (step 2).

For this purpose, the provider provides a dedicated token endpoint, which delivers the access token in exchange for the client credentials.

Token Request

```
POST /token HTTP/1.1
Host: clearing.de

grant_type=client_credentials
&client_id=xxxxxxxxxx
&client_secret=xxxxxxxxxx
```

Token Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store

{
  "access_token":"MTQ0NjJkZmQ5OTM2NDE1ZTZjNGZmZjI3",
  "token_type":"Bearer",
  "expires_in":3600,
  "refresh_token":"IwOGYzYTlmM2YxOTQ5MGE3YmNmMDFkNTVk",
  "scope":"clearing"
}
```

The access tokens generated in this way are short-lived (one hour in the above example) and thus offer a considerably smaller attack surface than (long-lived) access data.

NO specifications are made regarding the structure of the access tokens. The provider is responsible for ensuring that it is not possible for an attacker to improperly generate tokens on behalf of third parties.

Common implementation variants for this are

- **Signed tokens**: Usually implemented as **Json Web Tokens** (JWT) https://datatracker.ietf.org/doc/html/rfc7519. In this case, a data packet containing information about the user's identity is signed by the provider with a non-public key. By checking the signature of these tokens, the provider can verify the authenticity of the tokens and prevent attackers from generating new tokens. Signed tokens allow the provider to forgo sessions or similar server-side state keeping. See also https://ldapwiki.com/wiki/JSON%20Web%20Tokens
- **Opaque tokens**: The token consists of a reference (identifier) to a server-side context (usually a session) in which the user's identity is stored. An attacker cannot manipulate this context and thus cannot generate new tokens on behalf of third parties. See also https://ldapwiki.com/wiki/Opaque%20token.

1.2.3.2.2 Access Tokens usage

Access tokens are used in the form of bearer tokens, see **The OAuth 2.0 Authorization Framework: Bearer Token Usage**, https://datatracker.ietf.org/doc/html/rfc6750

```
GET /troubletickets HTTP/1.1
Host: clearing.de
Authorization: Bearer MTQ0NjJkZmQ5OTM2NDE1ZTZjNGZmZjI3
```

1.2.4 Safe channel

Since there is no central element in the network of platforms, the access data must be exchanged in pairs for each communication partner in a secure way. This process is part of the onboarding of new platforms.

2 Data privacy

2.1 GDPR conformancy

The following statements relate to the processing of personal data by a platform. Since platforms generally only process this data on behalf of a partner, an agreement on commissioned data processing (AVV) must be concluded for each partner-platform relationship. The basis for this AVV(s) is the information below.

Clearing cases contain personal data, namely data of

- Customers of partners (predominant part)
- employees of partners (if necessary, this part can be limited to organizational and access data)

Since a platform is itself a data-holding system for this data (implementation of status model, support of partners without own data storage, ...), a number of requirements have to be fulfilled according to GDPR:

- occasion-related processing and storage: the collected personal data may only be stored in the context of the occasion (processing of the clearing case) and for this duration (if necessary incl. an appropriate grace period for follow-up work / inquiries). Accordingly, a deletion concept for this data is required. This is created and implemented by each platform.
- Rights of data subjects: Claims for information, deletion, correction of personal data, etc. MUST be implemented for each platform involved.

Retention period: tickets and associated messages should be retained for a maximum of 6 months after a ticket is closed. In compliance with legal requirements, storage is as short as possible and as long as necessary for the contractual purpose. GDPR-relevant data MUST be pseudonymized or completely deleted when the clearing case is closed. Billing-relevant data MAY kept separately until the objection period after billing and does NOT contain any personal data.

2.2 Tenant separation

A platform acts as a service provider for a set of partners (=tenants), and accordingly holds data about clearing cases in which these partners are involved.

A platform MUST ensure that a partner can only view data of those clearing cases in which the partner is involved.

For this purpose, a platform MUST be able to at any time:

- uniquely assign incoming messages to a client
- be able to assign GUI actions to a client

3 Availability

In order to achieve high availability of the overall system, minimum standards are agreed. These are based on the cornerstones proven at WBCI and include the following aspects:

- The platforms SHOULD be operated 7 days / week x 24 hours / day and be accessible from the Internet.
- Following WBCI, an accessibility for fault reports*) is expected Monday to Friday from 09:00 to 17:00, but not on federal and regional holidays, leading to a response within <= 8h for severe faults (emergency/Prio1).
- A coordinated standard maintenance window (e.g. Wednesday 12:00 -13:00) is provided, but this is only used when actually needed. In general, information about necessary downtimes or maintenance work with relevant risks should be provided at least 3 workdays in advance in order to avoid unnecessary inquiries in case of non-availability (idea: "info board" to be maintained at a central location by all service providers themselves and visible to all participants, ideally subscribable).
- In trouble-free operation, the platforms SHOULD provide a suitable technical response within 10 seconds for at least 90% of all messages / requests via the defined interfaces (APIs according to TMF).
- Each platform ensures that "its" information remains available, even if other platforms or connected partner systems are temporarily unavailable.

*) Footnote for clarity: disruptions here are problems with the technical usability of the clearing platform(s). It is not about the disruptions that make clearing necessary.

4 Quantity structure

Throughput

An initial survey has determined a demand / volume of approx. 15,000 tickets per week. This results - with an approximate equal distribution over 5 working days per week and 8 hours per day - in approx. 6 tickets per minute.

Data volume

Assuming the retention times defined in the GDPR compliance section and <=5 messages/ticket, this means that approx. 400,000 tickets and approx. 1,000,000 messages must be stored simultaneously.

The data are preliminary data in the total market. In case of a distribution on several platforms, the requirements decrease proportionally. Due to long-running tickets (e.g.: continued supply), the requirement for the storage period and thus also the required storage volume can increase in part.