Primärsystemintegration (Gesundheitsämter)

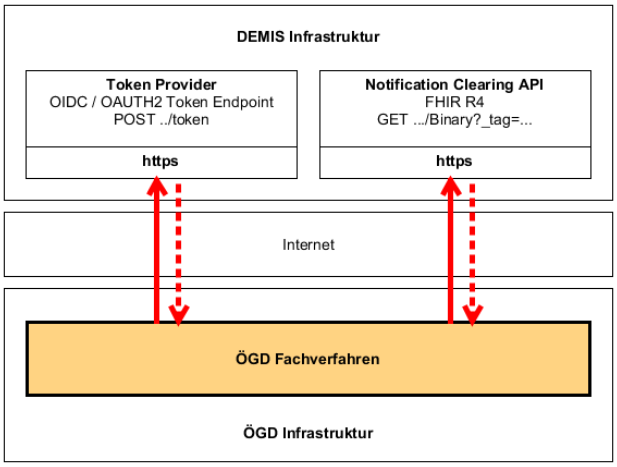
Unterseiten:

- Implementierungshinweise

Inhalt

Komponenten (Überblick)

Die folgende Abbildung bietet einen Überblick über die für den Abruf und die Verarbeitung von Meldungen relevanten Komponenten:

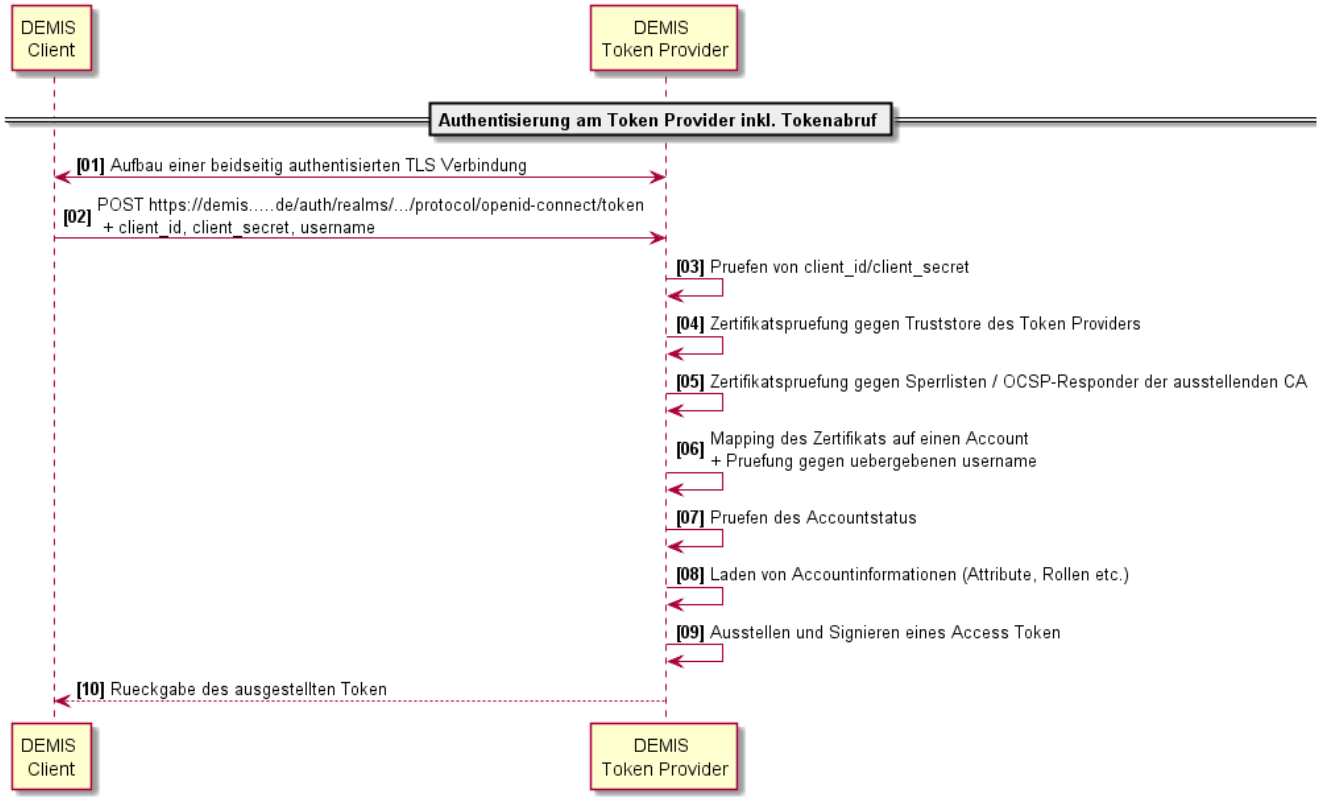


Token Provider – Sämtliche Zugriffe auf die zentralen Fachdienste der DEMIS-Infrastruktur (z.B. *DEMIS Notification Clearing API*) sind durch ein Access Control System geschützt. Um die für den jeweiligen Anwendungsfall benötigten Operationen aufrufen zu können, muss der Client ein standardisiertes Sicherheitstoken im Header des Aufrufs mitliefern. Diese Sicherheitstoken werden nach erfolgreicher Authentifizierung des jeweiligen Nutzers durch den *D EMIS Token Provider* ausgestellt. Die Authentifizierung der Nutzer erfolgt dabei zertifikatsbasiert. Der *DEMIS Token Provider* übernimmt ebenfalls die Verwaltung der Nutzer-Accounts.

Notification Clearing API – Die *DEMIS Notification Clearing API* dient als Abrufpunkt für verschlüsselte IfSG-Meldungen. Die Schnittstelle dieses Dienstes basiert auf der REST API von HL7 FHIR (https://www.hl7.org/fhir/). Für den Meldungsabruf wir jedoch lediglich ein kleiner Teil dieser API benötigt.

Abläufe (Überblick)

Tokenausstellung



[01] Der *DEMIS Client (z.B. ÖGD-Fachverfahren)* initiiert den Aufbau einer beidseitig authentisierten TLS-Verbindung. Die Übertragung der Informationen zwischen dem *DEMIS Client* und den Diensten der DEMIS Infrastruktur erfolgt somit grundsätzlich transportverschlüsselt und integritätsgeschützt. Ein Aufruf von Operationen am *DEMIS Token Provider* ist somit nur für Nutzer mit einem validen Zertifikat möglich.

[02] Der *DEMIS Client* sendet einen POST Request an den Token Endpoint des *DEMIS Token Provider*. Der Request beinhaltet für die Ausstellung des Token relevante Informationen.

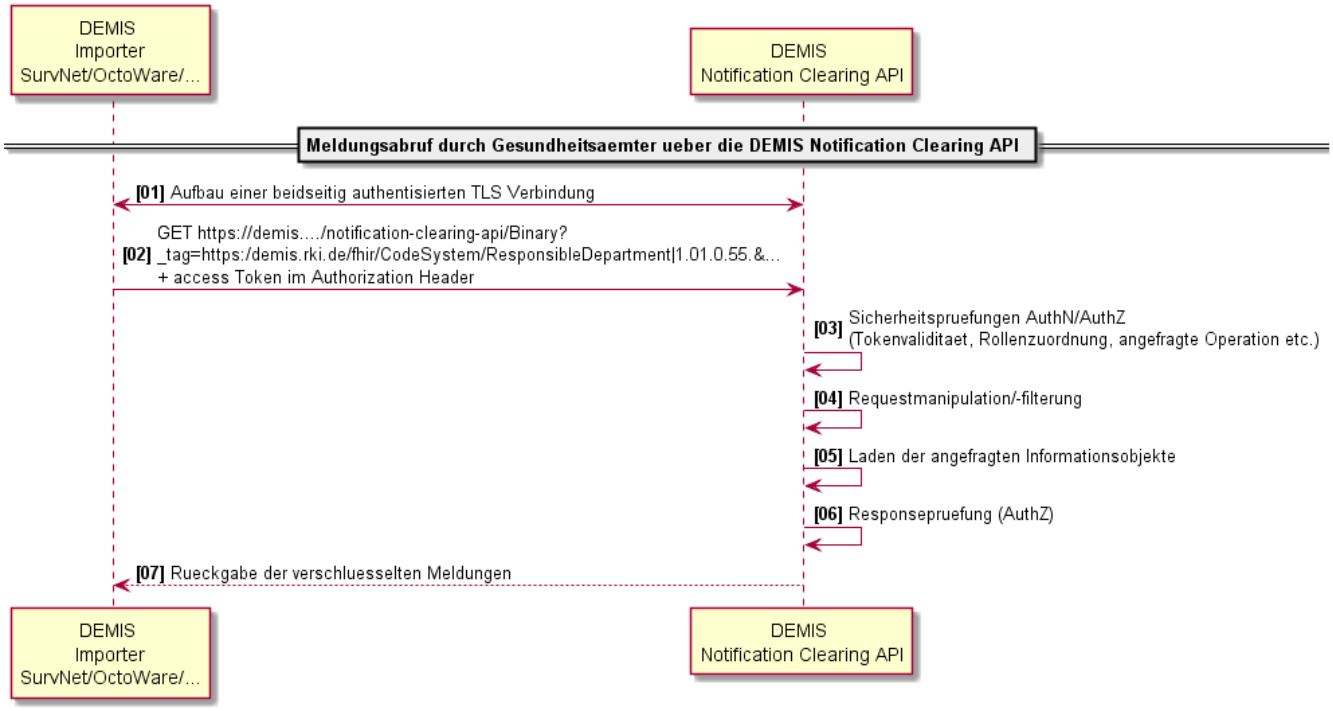
[03] - [06] Der *DEMIS Token Provider* validiert die übergebenen Parameter (inkl. Zertifikate) gegen die interne Konfiguration. Inkorrekte/ungültige Anfragen werden durch eine Fehlermeldung quittiert.

[07] - **[08]** Der *DEMIS Token Provider* prüft nun den Status des identifizierten Accounts. Für den Fall, dass der Account deaktiviert ist oder andere Anmeldevoraussetzungen nicht erfüllt werden, wird die Anfrage mit einer entsprechenden Fehlermeldung quittiert. Sind die Voraussetzungen für eine Anmeldung erfüllt, lädt der Token Provider alle relevanten Nutzerinformationen aus der Datenbank. Dazu gehören u.a. Identitätsattribute, wie z.B. sprechende Organisationsbezeichnungen aber auch die Zuordnung zu bestimmten Anwendungsrollen.

[09] Der *DEMIS Token Provider* bettet nun einen vorkonfigurierten Satz von Identitätsattributen in eine JSON-Datenstruktur ein, versieht diese mit einer Gültigkeitsdauer und signiert sie.

[10] Die ausgestellten Token werden nun eingebettet in eine JSON-Datenstruktur an den aufrufenden *DEMIS Client* zurückgegeben. Das enthaltene Access Token kann für den Zeitraum seiner Gültigkeit für die Autorisierung von Zugriffen auf Dienste der zentralen DEMIS Infrastruktur genutzt werden.

Meldungsabruf



[01] – Der *DEMIS Client* (Importer/Fachverfahren) baut eine beidseitig authentisierte TLS Verbindung auf. Die Übertragung der Informationen zwischen *DE MIS Client* und den Diensten der DEMIS Infrastruktur erfolgt somit grundsätzlich transportverschlüsselt und integritätsgeschützt. Ein Aufruf von Operationen an der *DEMIS Notification Clearing API* ist nur für Nutzer mit einem validen Zertifikat möglich.

[02] – Über einen GET Request fragt der *DEMIS Client* (Importer / Fachverfahren) verschlüsselte Meldungen über die *DEMIS Notification Clearing API* ab. Im http-Authorization-Header des Requests befindet sich ein vom *DEMIS Token Provider* ausgestelltes Access Token.

Beim Request handelt es sich um eine Suchanfrage wie sie HL7 FHIR für die REST API definiert (https://www.hl7.org/fhir/search.html). Gesucht wird üblicherweise nach allen verschlüsselten Meldungen, die dem jeweiligen Gesundheitsamt zugeordnet sind und nach einem bestimmten Zeitpunkt in das System eingestellt wurden. Bezogen auf die Abbildung der Datenstrukturen als FHIR Binary Ressourcen ergibt sich folgende exemplarische Suchanfrage:

```
GET https://demis.rki.de/notification-clearing-api/fhir/Binary?_tag=https://demis.rki.de/fhir/CodeSystem
/ResponsibleDepartment|1.01.0.55.&_lastUpdated=ge2020-05-16T14%3a07%3a19.025%2b01%3a00
```

[03] – Es erfolgt eine feingranulare Zugangs- und Zugriffssteuerung innerhalb der *DEMIS Notification Clearing API*. Hierfür werden die folgenden Verarbeitungsschritte innerhalb der Komponente durchgeführt:

- **Validierung des übermittelten Access Tokens** – Ein dedizierter Interceptor überprüft die Validität des im Authorization Header des Request übermittelten Access Tokens. Dies beinhaltet:
    - die Validierung der Struktur des Authorization Headers,
    - die Prüfung der Signatur des übermittelten Tokens,
    - die Prüfung des korrekten Ausstellers des übermittelten Tokens,
    - die Prüfung der zeitlichen Gültigkeit des übermittelten Tokens sowie
    - die Prüfung der korrekten „Audience“ des übermittelten Tokens.
- **Überprüfen der korrekten Nutzerrolle** – Im Token Provider werden den einzelnen Nutzern fachliche Rollen für die jeweiligen Dienste zugeordnet. Diese Rollen werden als Attribute im ausgestellten Access Token kodiert. Nur Nutzer, denen eine passende Rolle zugeordnet wurde (hier: "lab-notification-receiver") dürfen die Ihnen zugeordneten FHIR-Binary Ressourcen abfrufen. Ein dedizierter Interceptor übernimmt diese Prüfung innerhalb der *DEMIS Notification Clearing API*.

Aufrufe, mit fehlenden oder ungültigen Access Token oder fehlenden Rollenzuordnungen werden vom System abgelehnt und mit einer Fehlermeldung quittiert.

Hinweis: *Für eine der kommenden Ausbaustufen des DEMIS Systems ist es angedacht, Client Zertifikat und Access Token noch enger aneinander zu binden (z.B. durch die Einbettung des Zertifikatsfingerprints in ein Attribut des Tokens). Somit könnte aus dem derzeitigen „Bearer-Token-Szenario“ ein „Holder-of-Key-Szenario“ entstehen.*

[04] – Grundsätzlich wäre es möglich, Suchanfragen gegen die API zu stellen, die keine Einschränkungen bezüglich der Zuordnung zu einem bestimmten Gesundheitsamt vornehmen. Um entsprechende Szenarien zu unterbinden, wird - zusätzlich zur Autorisierung - grundsätzlich eine Manipulation /Einengung der Suchanfrage durchgeführt, die garantiert, dass nur eigene Meldungen gesucht werden. Dazu wird aus dem Access Token der Identifier des jeweiligen Gesundheitsamtes extrahiert und einem speziellen Such-„tag“ hinzugefügt.

[05] – Sobald die Requestprüfung abgeschlossen wurde, wird die Datenbank des Dienstes von der *DEMIS Notification Clearing API* durchsucht und entsprechende FHIR Binary Ressourcen geladen.

[06] – Jede der Suchanfrage entsprechende Binary Ressource wird nun analysiert, um sicherzustellen, dass tatsächlich eine Zuordnung zum entsprechenden Gesundheitsamt garantiert werden kann. Auch hier erfolgt die Überprüfung gegen den aus dem Access Token extrahierten Identifier des Gesundheitsamtes. Das entsprechende Merkmal in der Binary Ressource ist das „tag“, welches bereits als Suchparameter genutzt wurde. Alle Elemente, welche dem Kriterium nicht entsprechen, werden vom Ergebnis entfernt (doppelte Sicherheit).

[07] – Die Ergebnisse der Suche (verschlüsselte angereicherte Meldevorgänge [namentlich]) werden nun an den *DEMIS Client* (Importer/Fachverfahren) i m Gesundheitsamt zurückgegeben.

Tokenausstellung

Der folgende Abschnitt liefert Hintergrundinformationen bezüglich der korrekten Ansteuerung des *DEMIS Token Providers*. Diese Hinweise sollen Hersteller bei der korrekten Implementierung entsprechender Funktionalität sowie der Analyse von Problemen unterstützen.

Mutual TLS

Voraussetzung für die Anmeldung am DEMIS Token Provider und die anschließende Ausstellung eines Sicherheitstokens ist der Aufbau einer beidseitig authentisierten TLS-Verbindung mit dem entsprechenden Endpunkt. Informationen aus dem für den Verbindungsaufbau verwendeten Zertifikat sowie das Ergebnis der Zertifikatsprüfung bilden die Grundlage der Nutzeridentifizierung und -authentifizierung. Ein entsprechendes Zertifikat für die DEMIS Produktivumgebung kann über die DEMIS-Geschäftsstelle des Robert Koch-Instituts (demis-support@rki.de) durch die entsprechenden Stellen des ÖGD beantragt werden. Zertifikate für den Zugang/Zugriff auf die DEMIS Testumgebung müssen durch Hersteller gesondert angefordert werden.

Zulässige Ciphersuites

Folgende Ciphersuites können für den Aufbau der TLS-Verbindung genutzt werden:

```
ECDHE-ECDSA-AES128-GCM-SHA256
ECDHE-RSA-AES128-GCM-SHA256
ECDHE-ECDSA-AES256-GCM-SHA384
ECDHE-RSA-AES256-GCM-SHA384
ECDHE-ECDSA-CHACHA20-POLY1305
ECDHE-RSA-CHACHA20-POLY1305
DHE-RSA-AES128-GCM-SHA256
DHE-RSA-AES256-GCM-SHA384
```

Implementierungshinweise

Für fast alle Programmiersprachen existieren Bibliotheken, die den Aufbau einer beidseitig authentisierten TLS-Verbindung unterstützen. Im Zusammenhang mit deren Nutzung sind jedoch grundlegende Sicherheitsaspekte zu betrachten. Von besonderer Bedeutung sind in diesem Zusammenhang die folgenden Aspekte:

- **Sichere Schlüsselspeicherung** - Das mit dem Clientzertifikat assoziierte private Schlüsselmaterial muss bestmöglich vor unberechtigten Zugriffen geschützt werden. In diesem Zusammenhang sollte mit entsprechenden Mitteln des Betriebssystems der Zugriff auf die das Schlüsselmaterial enthaltene Datei eng eingegrenzt werden. Zusätzlich sollte das Schlüsselmaterial niemals unverschlüsselt im Dateisystem abgelegt werden. Entsprechende Standards für eine verschlüsselte Speicherung in speziellen Container existieren und werden von den einschlägigen Bibliotheken umfassend unterstützt. Das für die Ableitung des Containerschlüssels gewählte Passwort sollte entsprechend sicher gewählt werden. Entsprechende Empfehlung hierzu finden sich in einschlägigen Dokumenten des BSI.
- **Truststore Management** - Es sollte darauf verzichtet werden, den Default-Truststore zu verwenden, der üblicherweise mit TLS-Implementierungen ausgeliefert wird. Die zum Einsatz kommenden Serverzertifikate für die DEMIS-Test- und Produktivumgebung stammen aus einer speziellen D-Trust CA. Ausschließlich diese CA sollte im entsprechenden Truststore als vertrauenswürdig hinterlegt werden.
- **Prüfen des Serverzertifikats** - Das Serverzertifikat sollte entsprechend der üblichen Herangehensweise (Hostname-Verification, Certificate Chain Validation etc.) validiert werden.

Beispiele

Folgende Beispiele sollen für verschiedene Tools, die häufig im Rahmen der Entwicklung genutzt werden, zeigen, wie die zertifikatsbasierte Authentisierung konfiguriert werden kann:

curl

```
curl -verbose --key key.pem --cert cert.pem --request POST 'https://test.demis.rki.de/live-test/...' ...
```

Aus einem p12 Zertifikat kann man die cert.pem und die key.pem Zertifikate extrahieren, damit sie mit dem angegeben curl Befehl funktionieren:

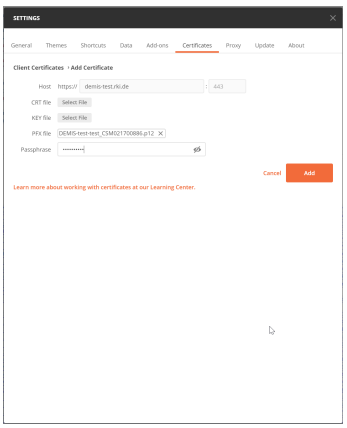
```
openssl pkcs12 -in path.p12 -out cert.pem -clcerts -nokeys
openssl pkcs12 -in path.p12 -out key.pem -nocerts -nodes
```

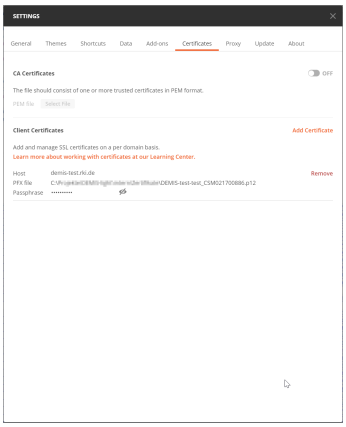
Nach Eingabe des p12 Passworts, werden die Dateien extrahiert. Diese kann man mit dem verlinkten curl Befehl verwenden.

Hinweis: Im angeführten Beispiel wird das Schlüsselmaterial - anders als in den Implementierungshinweisen empfohlen - unverschlüsselt gespeichert. Dies sollte nur mit dem Schlüsselmaterial der Testumgebung in dieser Form erfolgen.

postman

Einstellungen unter “File/Settings/Certificates”





Kommunikation mit dem Token-Endpoint

Endpunktadressen

Der Abruf von Token kann über die folgenden Endpunktadressen angestoßen werden:

PRODUKTION - https://demis.rki.de/auth/realms/OEGD/protocol/openid-connect/token

TEST - https://test.demis.rki.de/live-test/auth/realms/OEGD/protocol/openid-connect/token

Request-Struktur

Es muss ein http-POST-Request mit spezifischen Parametern an die jeweilige Endpunktadresse gesendet werden. Diese Parameter sind `x-www-form-urlencoded` zu übertragen:

```
client_id=demis-importer
client_secret=secret_client_secret
username=[placeholder]
grant_type=password
```

Username:

Bitte entnehmen Sie Ihre individuelle DEMIS-Kennung aus der Informationsmail des Robert Koch-Instituts zur DEMIS-Registrierung und Zertifikatsbereitstellung. Der *username* in der Konfiguration entspricht dem CN des Zertifikats ohne das vorgestellte "GA-". Er entspricht üblicherweise dem vom RKI vergebenen SiteCode.

Beispiele

curl

```
curl --verbose --key key.pem --cert cert.pem --request POST 'https://test.demis.rki.de/live-test/auth/realms/OEGD/protocol/openid-connect/token' --header 'Content-Type: application/x-www-form-urlencoded' --data-urlencode 'client_id=demis-importer' --data-urlencode 'client_secret=secret_client_secret' --data-urlencode 'username=test-test' --data-urlencode 'grant_type=password'
```

Ergebnis im Erfolgsfall

200 OK


```
{
    "access_token": "eyJhbGciOiJ...H9A",
    "expires_in": 600,
    "refresh_expires_in": 1800,
    "refresh_token": "eyJhbG...2Os",
    "token_type": "bearer",
    "not-before-policy": 0,
    "session_state": "82b56439-f5ce-41d8-a29b-544f2688ee58",
    "scope": "profile"
}
```


Mögliche Fehlermeldungen

http response code	response body	Mögliche Ursache

400 Bad Request	<html> <head><title>400 The SSL certificate error</title></head> <body>…</body> </html>	Das für die Authentisierung genutzte Zertifikat ist nicht korrekt oder ungültig. Bitte prüfen Sie, dass das für die jeweilige Umgebung (TEST, PROD) benutzte Zertifikat korrekt konfiguriert ist. Überprüfen Sie die zeitliche Gültigkeit des verwendeten Zertifikats. Überprüfen Sie den Status des Zertifikats.
400 Bad Request	{ "error": "unauthorized_client", "error_description": "INVALID_CREDENTIALS: Invalid client credentials" }	Die angegebene client_id ist nicht korrekt bzw. es wurde keine client_id angegeben.
401 Unauthorized	{ "error": "unauthorized_client", "error_description": "Invalid client secret" }	Das angegebene client_secret ist nicht korrekt.
401 Unauthorized	{ "error": "invalid_grant", "error_description": "Invalid user credentials" }	Der in den Parametern angegebene username ist inkorrekt .
403 Forbidden	<html> <head><title>403 Forbidden</title></head> <body>…</body> </html>	Bitte stellen Sie sicher, dass ein gültiges Clientzertifikat für den beidseitig authentisierten Verbindungsaufbau verwendet wird.
500 Internal Server Error	{ "error": "unknown_error" }	Der in den Parametern angegebene username korrespondiert ggf. nicht mit den Angaben im CN des SubjectDNs im für die Authentisierung genutzten Zertifikat. ggf. auch andere Fehler
503 Service Unavailable		Das System wird gewartet und ist nicht erreichbar.

Hinweis: Weitere Fehlermeldungen sind möglich. Die enthaltenen Beschreibungstexte sind dann jedoch zumeist aussagekräftig genug, um das Problem zu analysieren und zu lösen.

Struktur des Access Tokens

Für die weiteren Verarbeitungsschritte ist der in die Response-Datenstruktur eingebettete `access_token` von besonderem Interesse. Hierbei handelt es sich um einen JSON Web Token (JWT) gemäß RFC7519, der folgende Inhalte transportiert:

HEADER	**{ "alg": "RS256", "typ": "JWT", "kid": "tmi4F0p2voWPmdTVESY-CdylK9Gkt2Ycny9HC_OlMF8" }**

BODY	{
	"exp": 1597840018,
	"iat": 1597839418,
	"jti": "e21f4269-b4bd-437d-b1ae-392d616cdb78",
	"iss": "https://test.demis.rki.de/live-test/auth/realms/OEGD",
	"aud": "notification-clearing-api",
	"sub": "99ca7a5f-89d0-4451-8319-662fcfa3dab2",
	"typ": "Bearer",
	"azp": "demis-importer",
	"session_state": "82b56439-f5ce-41d8-a29b-544f2688ee58",
	"acr": "1",
	"resource_access": {
	"notification-clearing-api": {
	"roles": [
	"lab-notification-receiver"
	]
	}
	},
	"scope": "profile",
	"organization": "Test Test",
	"preferred_username": "test-test"
	}

SIGNATURE	**[Signature Bytes]**

Das Access Token ist als in seiner enkodierten Form als Bestandteil des Requests eingebettet im `Authorization` header (Bearer Token) an die *DEMIS Notification Clearing API* zu senden.

Abruf von Meldevorgängen/Meldungen

Der folgende Abschnitt liefert Hintergrundinformationen bezüglich der korrekten Ansteuerung der *DEMIS Notification Clearing API*. Diese Hinweise sollen Hersteller bei der korrekten Implementierung entsprechender Funktionalität sowie der Analyse von Problemen unterstützen.

Wie bereits im Überblick (s.o.) beschrieben, ruft der Client (z.B. ÖGD-Fachverfahren) für den jeweiligen Empfänger individuell verschlüsselte Meldevorgänge/Meldungen ab. Diese müssen vor ihrer Verarbeitung also zunächst entschlüsselt werden. Sowohl die verschlüsselten als auch die unverschlüsselten Meldevorgänge nutzen als zugrundeliegenden Standard für das Informationsmodell **HL7 FHIR (R4).** Auch die für den Informationsabruf implementierte API richtet sich strikt nach dem entsprechenden Standard.

Struktur und Inhalt verschlüsselter Meldevorgänge / Meldungen

Verschlüsselte Meldevorgänge werden in DEMIS als HL7 FHIR Binary Ressourcen abgebildet. Diese sind von Ihrer Struktur her recht einfach aufgebaut. Das folgende Beispiel demonstriert den Aufbau exemplarisch:


```
{
  "resourceType": "Binary",
  "id": "1552",
  "meta": {
    "versionId": "1",
    "lastUpdated": "2020-05-05T23:32:35.784+02:00",
    "source": "#acbIrL6UPNNzwBo6",
    "tag": [
      {
        "system": "https://demis.rki.de/fhir/CodeSystem/ResponsibleDepartment",
        "code": "1.11.0.11.09.",
        "display": "Assigned to 1.11.0.11.09."
      },
      {
        "system": "https://demis.rki.de/fhir/CodeSystem/RelatedNotification",
        "code": "b452e47f-d19b-4a21-8c9a-20471d19a0f6",
        "display": "Relates to message with identifier: b452e47f-d19b-4a21-8c9a-20471d19a0f6"
      },
      {
        "system": "https://demis.rki.de/fhir/CodeSystem/ResponsibleDepartmentPrimaryAddress",
        "code": "1.11.0.11.09.",
        "display": "Patient primary address department: 1.11.0.11.09."
      }
    ]
  },
  "contentType": "application/cms",
  "data": "TUlBR0NTcUd ... [more bytes] ... FBQUFBQQ=="
}
```


resourceType - Typ der FHIR Ressource - Dieses Feld wird entsprechend der gestellten Anfrage beim Abruf von verschlüsselten Meldevorgängen immer "Binary" enthalten
id - ID der Binary Ressource im System. Unter dieses ID kann die Ressource für die Dauer ihrer Speicherung direkt abgerufen werden. Derzeit erfolgt die automatische Löschung von Meldevorgängen nach 90 Tagen
meta.versionId - Version der Binary Ressource - In der aktuellen Ausbaustufe des Systems wird es lediglich eine Version je Meldevorgang geben
meta.lastUpdated - Zeitpunkt der letzten Änderung der Ressource. Dieser Zeitstempel entspricht hier der Anlage des Meldevorgangs/Meldung im System und wird für die weiter unten beschriebenen Suchanfragen genutzt.
meta.source - Quelle der Ressource - Internes Metadatum ohne Nutzen für die abrufenden Stellen.
meta.tag - Verschiedene Metainformationen über den verschlüsselten Meldevorgang, z.B. dessen Identifier, das zuständige Gesundheitsamt und das für den Hauptwohnsitz der betroffenen Person zuständige Gesundheitsamt.
contentType - Type des im "data" Feld enthaltenen Inhalts - Für verschlüsselte Meldungsvorgäng wird die immer "application/cms" sein
data - Payload - enthält den verschlüsselten Meldevorgang in base64-kodierter Form

Abruf von verschlüsselten Meldevorgängen / Meldungen

Mutual TLS

Für die Nutzung der *DEMIS Notification Clearing API* ist der Aufbau einer beidseitig authentisierten TLS-Verbindung erforderlich. Es gelten die hier getroffe nen Festlegungen und Hinweise in entsprechender Form.

Endpunktadressen

Der Abruf von verschlüsselten Meldevorgängen / Meldungen erfolgt über die folgenden Endpunktadressen:

- **PRODUKTION** - https://demis.rki.de/notification-clearing-api/fhir/...
- **TEST** - https://test.demis.rki.de/live-test/notification-clearing-api/fhir/...

Vorbemerkungen

Der Abruf von Meldung über die *DEMIS Notification Clearing API* ist für die Gesundheitsämter bzw. die zum Einsatz kommende Software an bestimmte Voraussetzungen geknüpft:

- Grundsätzlich dürfen nur Systeme, die über ein gültiges Access Token verfügen, verschlüsselte Meldungen über die *DEMIS Notification Clearing API* abrufen. Dabei gilt es zu beachten, dass die zum Einsatz kommenden Token nur eine bestimmte zeitliche Gültigkeit besitzen (derzeit konfiguriert auf 10 Minuten). Ist ein Token abgelaufen, muss es erneuert werden. Aufrufe an der *DEMIS Notification Clearing API* mit ungültigem Token werden verworfen.
- Gesundheitsämter dürfen nur verschlüsselte Meldungen abrufen, die für ihr eigenes Gesundheitsamt bestimmt sind. Anhand eines in der entsprechenden Binary Ressource hinterlegten Tags (s.o.) wird durch das Access Control System der *DEMIS Notification Clearing API* unter Zuhilfenahme der im Access Token hinterlegten Informationen entschieden, ob eine Meldung abgerufen werden darf oder nicht.

Suchanfragen

Wie eingangs erwähnt, setzt die *DEMIS Notification Clearing API* die durch HL7 FHIR definierte RESTful API um. Entwickler, die mit dem entsprechenden Standard noch nicht vertraut sind, seien insbesondere die folgenden Seiten empfohlen:

- https://www.hl7.org/fhir/http.html - Beschreibung der FHIR RESTful API - relevant für die Umsetzung des Anwendungsfalls ist hier jedoch primär die "search"-interaktion auf Type-Level
- https://www.hl7.org/fhir/search.html - Beschreibung von Besonderheiten beim Suchen von Ressourcen
- https://www.hl7.org/fhir/binary.html - Beschreibung des für die Abbildung des verschlüsselten Meldevorgangs genutzten Binary Ressource
- https://www.hl7.org/fhir/http.html#paging - Paging von Suchergebnissen - Relevant für den Fall, dass die Ergebnismenge einer Suche einen eingestellten Schwellwert überschreitet

  Hinweis: Die API wird nur in dem für die Umsetzung des Anwendungsfalls erforderlichen Maße implementiert. Somit ist nur ein Bruchteil der im Standard enthalten Funktionalität zu berücksichtigen.

Der DEMIS-Importer, welcher sowohl als produktives Werkzeug für Abruf und Entschlüsselung von Meldevorgängen als auch als Implementierungsreferenz genutzt wird, verwendet die API in folgender Art und Weise:

- Gesucht wird direkt auf dem Ressourcen-Typ Binary

  ```
  GET https://demis.rki.de/notification-clearing-api/fhir/Binary?...
  ```

- Um Probleme beim Meldungsabruf zu vermeiden, empfiehlt sich ein Vorgehen, dass darauf angelegt ist, ausschließlich die für das „eigene“ Gesundheitsamt bestimmten Meldungen anzufragen. Hilfreich hierbei ist die Verwendung eines bestimmten Suchkriteriums („_tag“):

  ```
  GET https://demis.rki.de/notification-clearing-api/fhir/Binary?_tag=https://demis.rki.de/fhir/CodeSystem
  /ResponsibleDepartment|1.11.0.08.01.&...
  ```

- Die entsprechende Anfrage würde jedoch sämtliche für dieses Gesundheitsamt hinterlegten Meldungen im System suchen. Aus diesem Grund empfiehlt sich eine weitere Einschränkung des Such-/Ergebnisraumes, z.B. über den „_lastUpdated“ Parameter:

  ```
  GET https://demis.rki.de/notification-clearing-api/fhir/Binary?_tag=https://demis.rki.de/fhir/CodeSystem
  /ResponsibleDepartment|1.11.0.08.01.&_lastUpdated=ge2021-01-12T14%3a07%3a19.025%2b01%3a00&...
  ```

- Um die weiter unten beschriebene Problematik der Begrenzung der Gesamtergebnismenge optimal adressieren zu können, kann es u.U. erforderlich sein, eine zweite Anfrage mit angepasstem _lastUpdated-Zeitstempel "nachzuschieben". Um diese korrekt formulieren zu können, ist das Wissen um den lastUpdated-Zeitpunkt der neuesten in der Gesamtergebnismenge enthaltenen Binary-Ressource wichtig. Um diese nicht lokal auf dem Client ermitteln zu müssen, empfiehlt sich die Sortierung der zurückgegebenen Ressourcen nach dem "_lastUpdated" Attribut:

  ```
  GET https://demis.rki.de/notification-clearing-api/fhir/Binary?_tag=https://demis.rki.de/fhir/CodeSystem
  /ResponsibleDepartment|1.11.0.08.01.&_lastUpdated=ge2021-01-12T14%3a07%3a19.025%2b01%3a00&_sort=_lastUpdated
  ```

- Übersteigt die Anzahl der Ergebnisse der Suchanfrage einen im System konfigurierten Schwellwert, wird nur noch ein Subset der Ressourcen in der Responsenachricht zurückgegeben. Weitere Ressourcen sind über separate Requests abzurufen. Die Urls dieser Requests können der Responsenachricht entnommen werden (vgl. Paging in FHIR).

  Wichtig: Neben dem Schwellwert für das Paging existiert ein weiterer interner Schwellwert für das System, welcher die zulässige Gesamtergebnismenge einer Suchanfrage begrenzt. Dies ist notwendig, um das System und die Datenbank vor Überlast zu schützen. Es ist somit in bestimmten Konstellationen möglich, dass trotz des korrekten Abrufens aller Pages, nicht alle Ergebnisse zurück geliefert werden, die die Kriterien der Suchanfrage erfüllen. Weitere Details zu diesem Verhalten und eine Lösungsmöglichkeit werden in der Dokumentation bzw. dem Quelltext des DEMIS-Importer aufgezeigt.

Entschlüsseln von Meldevorgängen / Meldungen

Die im Rahmen des Meldungsabrufs heruntergeladenen Binary-Ressourcen enthalten die Meldevorgänge/Meldungen in einer für das jeweilige Gesundheitsamt verschlüsselten Form. Für die Darstellung der verschlüsselten Daten wird die Cryptographic Message Syntax (CMS) entsprechend RFC 5652 verwendet. Hierbei handelt es sich um eine ASN.1 Datenstruktur, die sämtliche für die Entschlüsselung benötigten Informationen in strukturierter Form vorhält.

Meldungen werden derzeit durch ein hybrides Verfahren geschützt. Die Meldung selbst wird mit einem symmetrischen Schlüssel verschlüsselt. Dieser symmetrische Schlüssel wiederum mit dem öffentlichen Schlüssel des jeweiligen Gesundheitsamtes. Derzeit kommen folgende Algorithmen für die Verschlüsselung zum Einsatz:

- symmetrisch: AES256_CBC
- asymmetrisch: RSAES_OAEP

Für die Entschlüsselung der CMS Datenstruktur kann auf eine Vielzahl von Bibliotheken zurückgegriffen werden. Eine „manuelle“ Verarbeitung durch das Fachverfahren ist nicht erforderlich. Für die Entschlüsselung muss das den Gesundheitsämtern ausgestellte Zertifikat bzw. das zugehörige private Schlüsselmaterial verwendet werden.

Weiterverarbeitung von Meldevorgängen / Meldungen

Entschlüsselte Meldevorgänge liegen derzeit ausschließlich als FHIR-XML-Repräsentationen vor, wenngleich der FHIR-Standard grundsätzlich auch andere Repräsentationsarten (z.B. JSON) unterstützt. Für die Weiterverarbeitung der Meldung im jeweiligen Fachverfahren empfiehlt sich dennoch der Einsatz einer existierenden FHIR-Bibliothek, da diese den Umsetzungsaufwand erheblich reduzieren kann.

Hinweise zur grundlegenden Meldungsstruktur und den Inhalten können der Seite FHIR Profile entnommen werden.

Weitere Hinweise

Systemverhalten im Wartungsfall

- Sofern sich die DEMIS-Backendinfrastruktur im Wartungsmodus befindet, wird ein „503 Service Unavailable“ zurückgegeben. Der Meldungsabruf sollte für eine konfigurierbare Dauer eingestellt und danach wieder aufgenommen werden.

Mitsenden des User-Agent

- Ein abrufender Client soll immer einen User-Agent der Form

  ```
  User-Agent: <product> / <product-version> <comment>
  ```

  mitschicken, siehe https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/User-Agent

Implementierungsbeispiele

- Der DEMIS-Importer kann als Implementierungsbeispiel für den Meldungsabruf und die -entschlüsselung genutzt werden.
- Der Quelltext kann derzeit direkt über das RKI (demis-support@rki.de) angefragt werden. Zukünftig ist eine Veröffentlichung auf github vorgesehen.