Central Instance of Ontoserver

Medical Data Integration Center (MeDIC)
University of Cologne, Faculty of Medicine and University Hospital Cologne

Muhammad Adnan 28.06.2023

AGENDA

- Intro to Ontoserver
  - FHIR Terminology Server
  - Interoperability! A solution
- Deployment at Cologne
- Network Architecture
- Authentication
- Access to Central Instance
  - DFN PKI / GÉANT
  - IP Allowed list
- Toubleshooting

Intro to Ontoserver

- Turnkey, high performance HL7 FHIR Terminology Service
- Access to all FHIR CodeSystems, SNOMED CT, and LOINC
- Support all those operations on
- FHIR Resources
- LOINC
- SNOMED CT {with support of ECL (Expression Constraint Language)}

- Ontoserver API examples:
- https://documenter.getpostman.com/view/634774/TVsuBmc9
- Implementation examples: https://aehrc.github.io/fhir-ts-exemplars/

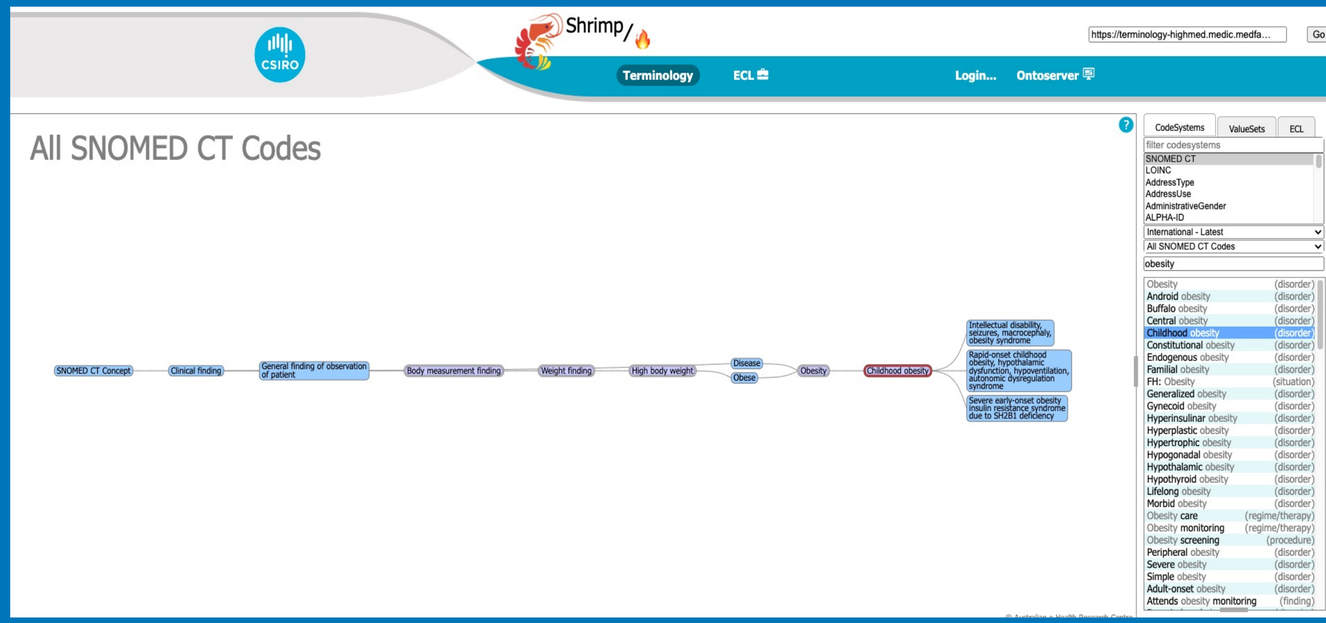
Interoperability! A solution

- Terminology Use: Problem

- Each clinical providers use their own interpretation of a patient condition
- Multiple type of docuementation
- Several terms used to describe Patient condition
- Cause confusion at the end!

- Bigger Picture: how to make this communication interoperable?

- By using Standard Code and Code Systems
- Independent of language
- Consistent across systems



Deployment of Ontoserver

Docker Compose

http://ontoserver.csiro.au/docs/6/ provides a basic example

Runs up two containers on a host
A few things to change

- atom.syndication.feedLocation

- Currenlty defaults to

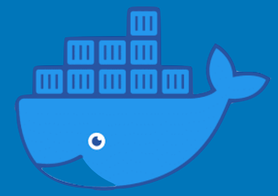
https://api.healthterminologies.gov.au/syndication/v1/syndication.xml
and
https://r4.ontoserver.csiro.au/synd/syndication.xml fallback

- Start with (or central HiGHmed)
https://r4.ontoserver.csiro.au/synd/syndication.xml

- authentication.oauth.endpoint.client_id.0
- authentication.oauth.endpoint.client_secret.0

- Provide more memory if you can

```yaml
version: '3'
volumes:
  onto:
    driver: local
  pgdata:
    driver: local
services:
  db:
    image: postgres:12
    volumes:
      - pgdata:/var/lib/postgresql/data
    healthcheck:
      test: ["CMD-SHELL", "pg_isready -U postgres"]
      interval: 10s
      timeout: 5s
      retries: 5
    environment:
      - POSTGRES_HOST_AUTH_METHOD=trust
  ontoserver:
    image: quay.io/aehrc/ontoserver:ctsa-6
    container_name: ontoserver
    read_only: true
    security_opt:
      - no-new-privileges
    depends_on:
      - db
    ports:
      - "8443:8443"
      - "8080:8080"
    environment:
      - spring.datasource.url=jdbc:postgresql://db/postgres
      # These two lines are specific to deployment in Australia only
      - authentication.oauth.endpoint.client_id.0=NCTS_CLIENT_ID
      - authentication.oauth.endpoint.client_secret.0=NCTS_CLIENT_SECRET
      - JAVA_OPTS=-Xmx2G        # Minimum
    #       - JAVA_OPTS=-Xmx8G      # Preferred
    volumes:
      - onto:/var/onto
      - /tmp
      - /var/log
```



External Database

- this example uses a liked database container with default connection settings

- when setting up external database with other settings, please update the following settings:

- spring.datasource.url=jdbc:postgresql://<host>/<database>
- spring.datasource.username=<username>
- spring.datasource.password=<password>

- Ontoserver requires a PostgreSQL version 10 or newer

Important Configuration

•Initial setup of up-stream (source):

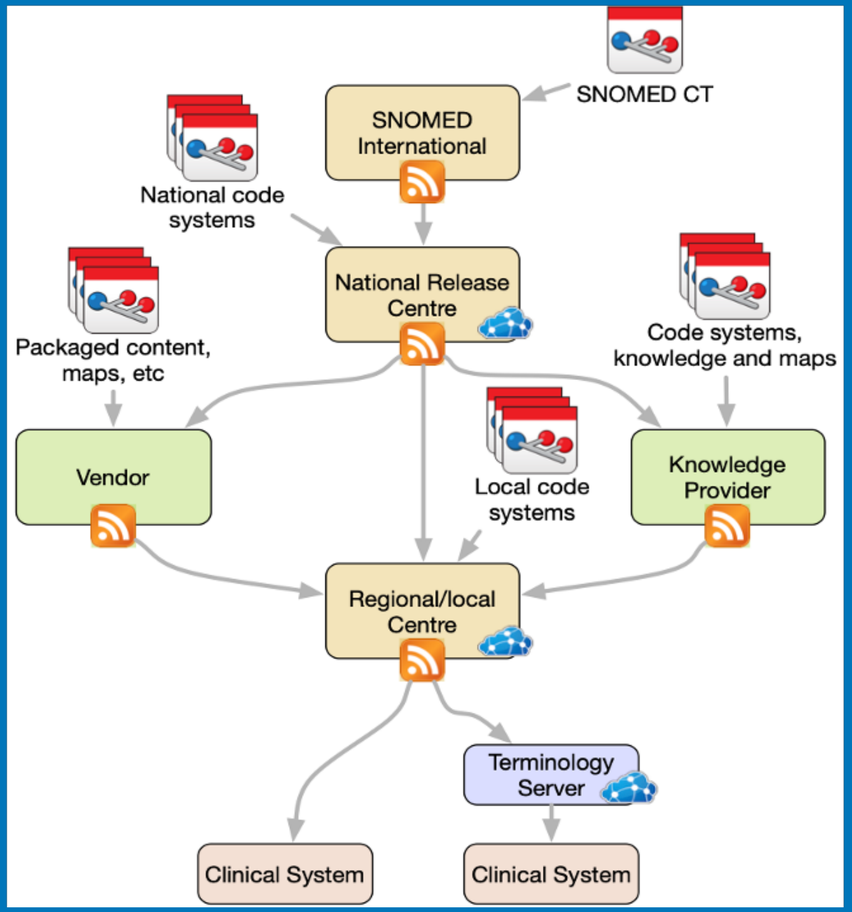
- Feed location
  atom.syndication.feedLocation=file://syndication.xml

-Enable security
  -Ontoserver.security.enabled
  -Ontoserver.security.readOnly.fhir=true
  -Ontoserver.security.readOnly.api=true
  -Ontoserver.security.readOnly.synd=true

-Config memory
  -JAVA_OPTS=-Xmx16G

-Syndication base URLs
  -Ontoserver.sysd.base={{source_url}}/synd
  -Ontoserver.fhir.base={{source_url}}/fhir

-Detailed documentation and minimal configuration:
-https://ontoserver.csiro.au/docs/6/



Deployment at Uniklinik Köln

Local Deployment (Test & Production)

- to enable redundency
- enable testing locally
- enables us to provide access locally
- Deploys as a pair of Docker containers
  - Ontoserver and Database (& http caching proxy)
  - docker-compose file to configure
  - Deployed at Uniklinik Koeln data center

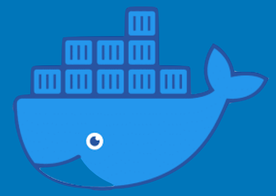
- Central Deployment (Test & Production)

- provide more control over disaster recovery

- Deploys as a pair of Docker containers
  - Ontoserver and Database (& http caching proxy)
  - Read-Only access

  - *docker-compose* file to configure

  - Deployed at Uniklinik Koeln data center



Authentication mechanism

Mutual TLS – DFN PKI /
GÉANT PKI

DFN User certificate

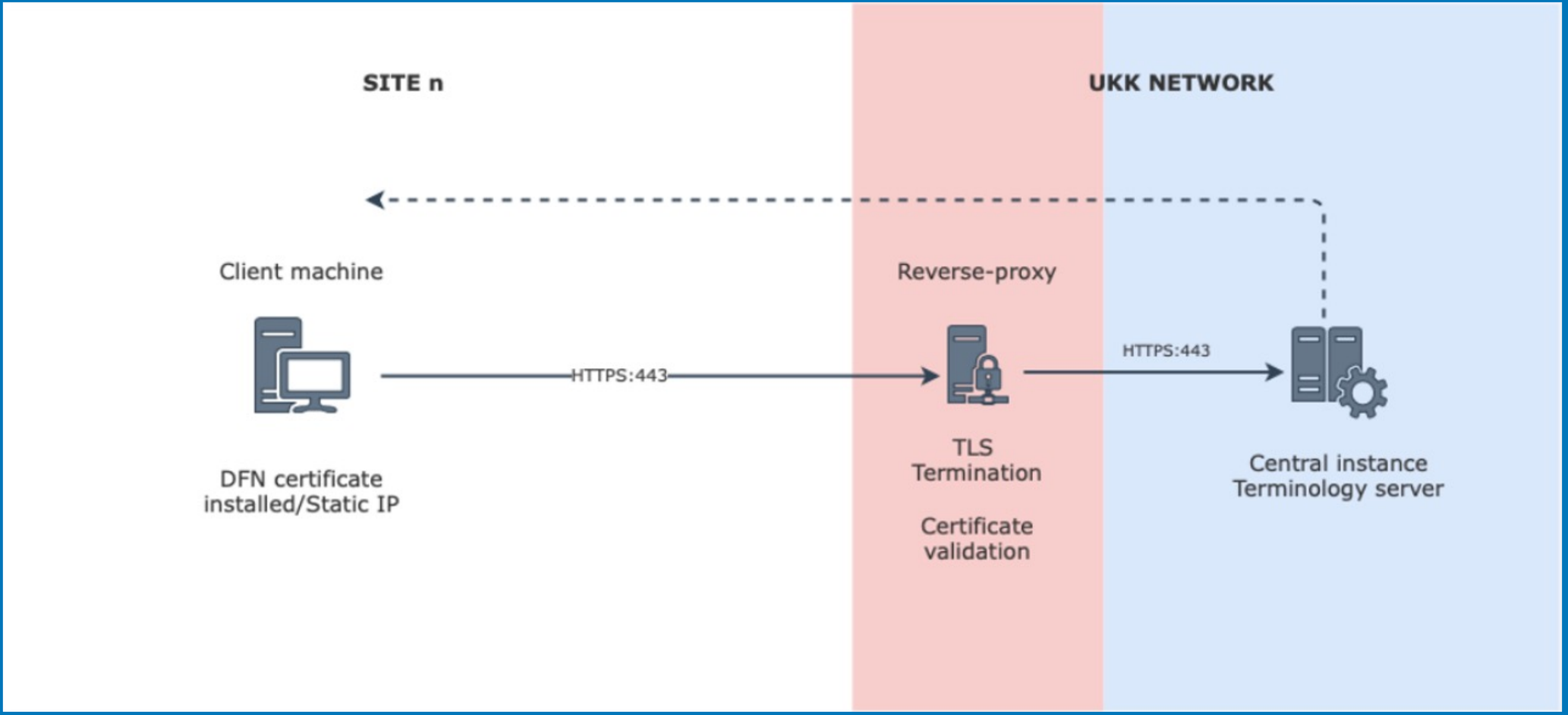
DFN 802.1X Client certificate
- clientAuth, serverAuth

IP Allowed-list

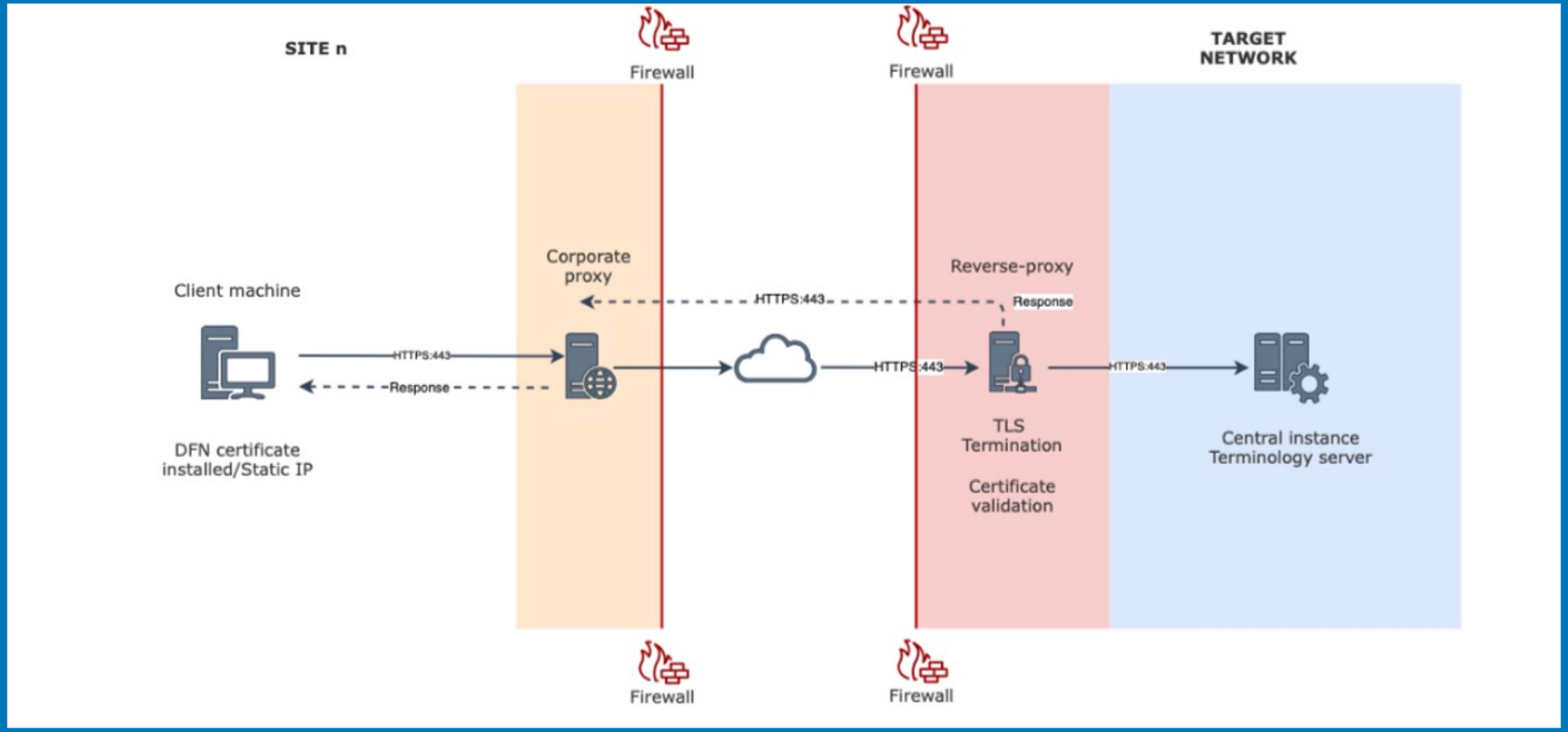
Dedicated / Static IP
Address

Admission form signed by PI
or Site representative

Deployment at Uniklinik Köln



Deployment at Uniklinik Köln



Accessing the CTS

Access as User

- Obtain appropriate DFN / Géant certificate from your institution
- Convert the certificate to a PKCS (.pfx or p12) format

```
openssl pkcs12 -export -out certificate.p12 -in certificate.pem -inkey privateKey.pem
```

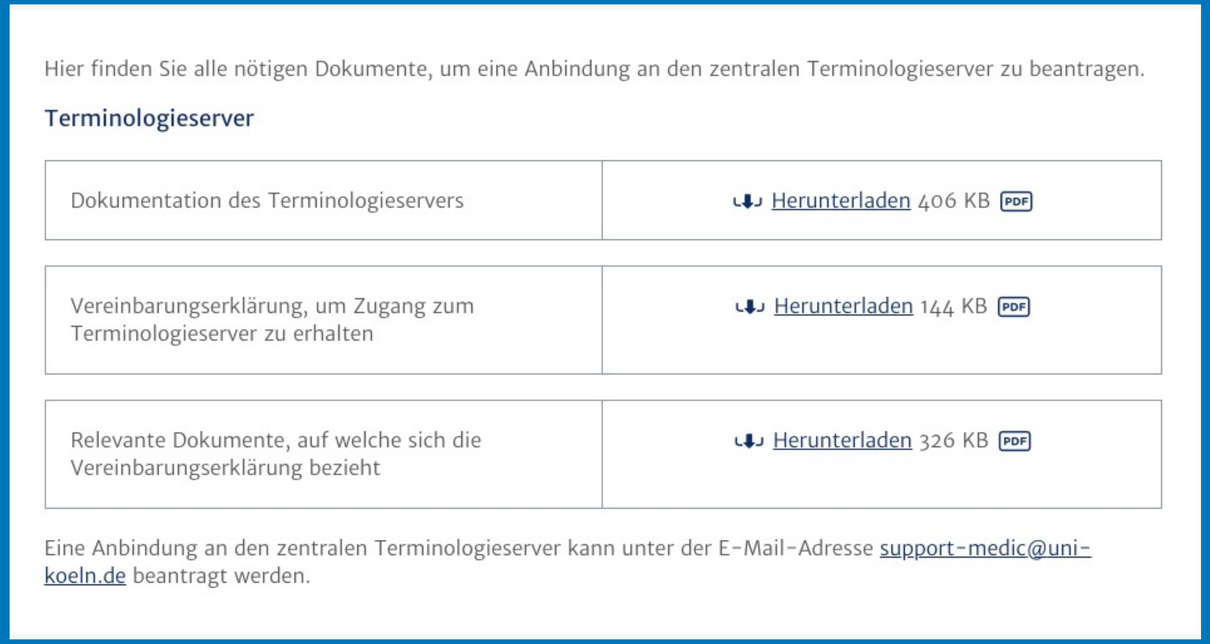
- Upload certificate to browser or client
- https://terminology-highmed.medic.medfak.uni-koeln.de/fhir/metadata

Access Requests for Central instance

- We have a website, link below
- https://www.uk-koeln.de/forschung/medical-data-integration-center-medic/

- Link for forms and information about accessing MeDIC services.
- https://www.uk-koeln.de/forschung/medical-data-integration-center-medic/dokumente-formulare/

- Useful email address:
- support-medic@uni-koeln.de



Hier finden Sie alle nötigen Dokumente, um eine Anbindung an den zentralen Terminologieserver zu beantragen.

Terminologieserver

Dokumentation des Terminologieservers	Herunterladen 406 KB PDF
Vereinbarungserklärung, um Zugang zum Terminologieserver zu erhalten	Herunterladen 144 KB PDF
Relevante Dokumente, auf welche sich die Vereinbarungserklärung bezieht	Herunterladen 326 KB PDF

Eine Anbindung an den zentralen Terminologieserver kann unter der E-Mail-Adresse support-medic@uni-koeln.de beantragt werden.

Troubleshooting

- Use the right certificate
- For browsers use .pfx or .p12
  - Try *icognito* mode
- Preferred configuration for client machine – Use a proxy like Nginx to load complete certificates chain (.pem, .crt format)
- Contact support team: support-medic@uni-koeln.de