Horizon 2020 Programme
DG CNECT
Next-generation Internet

NGI TRUST

Project acronym: **DECTS**

Project title: **Deaf Emergency Chat and Training System**

Third Party: **OwnYourData**

Deliverable 2.2

Design Specification

Deliverables leader:	**OwnYourData**
Authors:	Christoph Fabianek, Mario Murrent, Wolfgang Kampichler, Richard Prinz, Eduard Gringinger
Due date:	2020-10-15
Actual submission date:	2020-09-18
Dissemination level:	Public

Abstract: This report is part of a third-party project DECTS that has received funding from the NGI_Trust, the European Union's Horizon 2020 research and innovation programme under grant agreement No. 825618.

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

Document Revision History

Date	Version	Author/Editor/Contributor	Summary of main changes / Status
2020-03-24	0.1	Christoph Fabianek	Initial document
2020-05-24	0.2	Mario Murrent	Updated initial document
2020-06-09	0.3	Christoph Fabianek	
2020-06-17	0.4	Christoph Fabianek, Mario Murrent	draft version ready for review
2020-07-02	0.5	Christoph Fabianek, Eduard Gringinger	draft version ready for coaching session
2020-09-16	0.6	Christoph Fabianek	document final end-to-end tests
2020-09-18	1.0	Christoph Fabianek, Eduard Gringinger	Final version

Disclaimer

Copyright

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

Table of content

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

Executive Summary

The deliverable provides the design description of the DECTS project which implements and evaluates an end-to-end workflow for deaf and hard of hearing persons for consent management, secure and privacy preserving personal data provisioning in case of an emergency, and rolling out the existing Austrian solution at the European level.

The document includes a general introduction to the project in chapter 1, the methodology including a stakeholder analysis in chapter 2, and the actual requirements in chapter 3. The document is concluded with an outlook and a glossary.

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

1 Introduction

1.1 Background / Deliverable Description

Article 9 of the UN convention of the rights of persons with disabilities requires countries to take measures for the full and equal participation of persons with disabilities (including access to communication and information services) and the European Disability Strategy 2010-2020 also calls for the principle of accessibility at all levels. Despite this, there are still about 1 million deaf and hard of hearing persons in Europe who currently rely on outdated technology (e.g. fax) and help from others to make an emergency call.

The good news is that existing standards and technologies can provide an adequate and barrier-free solution. DEC112 (Deaf Emergency Call System) already implemented an emergency infrastructure (compliant to NENA NG9-1-1 and ETSI TS 103479) including a mobile app to enable deaf and hard of hearing persons to access emergency services in Austria. This solution has been in operation since February 2019. A lot has been learned about the actual needs of emergency callers as well as call takers in control rooms.

This document describes the requirements In the DECTS project (Deaf Emergency Chat and Training System) and the main goals are:

- Research and development of consent management technology to exchange training chat protocols between deaf persons and call takers in control rooms.
- Allow secure and privacy-preserving data provisioning of pre-recorded personal information in the course of an emergency chat.
- Extend DEC112 to operate at European level by implementing national and international accessibility to emergency services.

1.2 Relation to other DECTS deliverables

This design document is one out of 3 documents providing the detailed description about this project:

- D2.1 Requirements Document: lists functional and non-functional requirements for consensus management, data provisioning, and emergency call routing
- D2.2 Design Specification: describes and depicts the system design together with API endpoints and data formats of the various components; it also includes translated texts for the multi-language mobile app, chat bot, and viewer application
- D2.3 Data Management Plan: outlines used data, processing steps, as well as storage and archiving means

2 Architecture, work done and current status

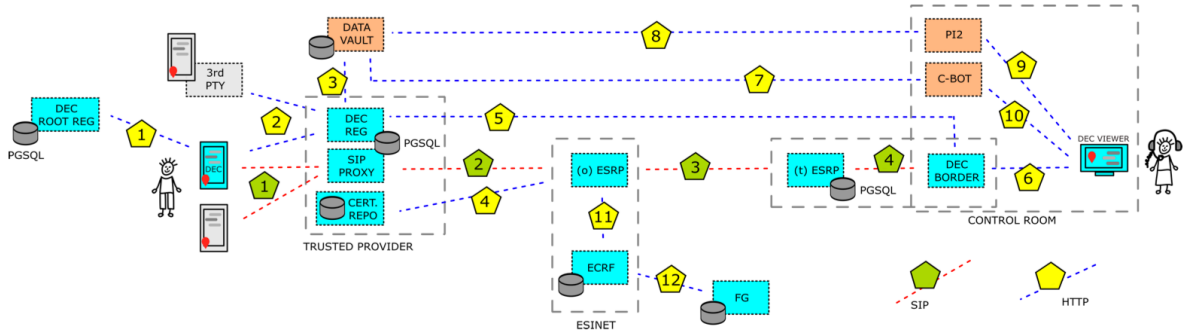
This section presents the performed work in the DECTS project regarding architecture of the overall solution, current status, and outlook for further development.

2.1 Methodology

- Starting with an overall architecture based on the existing DEC112 system

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

- add building blocks for new functionalities (Data Vault & PI2)



- extend functions (DEC112 App, Chatbot, Viewer)
- monthly End-to-End Tests to verify functionality and validate design
- reach out to existing users (organisations who used DEC112 either in a productive system or through tests) and validate functionality in phase 3

2.2 Maintenance and next steps

Currently, the DEC112 system is maintained by Wolfgang Kampichler, Richard Prinz and Mario Murrent (company: MeeCode by Mario Murrent) with the help of volunteers and funded through various research grants and company donations.

The OwnYourData Data Vault is maintained and operated by the public charity to foster personal use of data (legal name: Verein zur Förderung der selbstständigen Nutzung von Daten)- a non-profit association registered in Austria.

Both groups are working now for some years in their respective domain and have joined forces in this project. Because of the fruitful collaboration and the established user base they will continue to operate the described system and there are plans to find a dedicated non-profit association to also provide a legal entity for upcoming projects.

3 Functional Components

3.1 DEC112 App

The DEC112 App provides an easy, reliable and secure way for hearing impaired people to text for help in an emergency through a simple and intuitive interface.

The DEC112 App is standard compliant: where applicable available standards are used (emergency chat transmission) and through active collaboration in various groups and organizations additional standards will be adopted.

The relevant data flows / workflows for the DEC112 app are described in detail in chapter 4 and include:

- User Registration
- Profile Data Management
- Emergency Chat

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

On the first startup of the DEC112 App a registration wizard guides the user through the registration process (Figure 3.1 left).

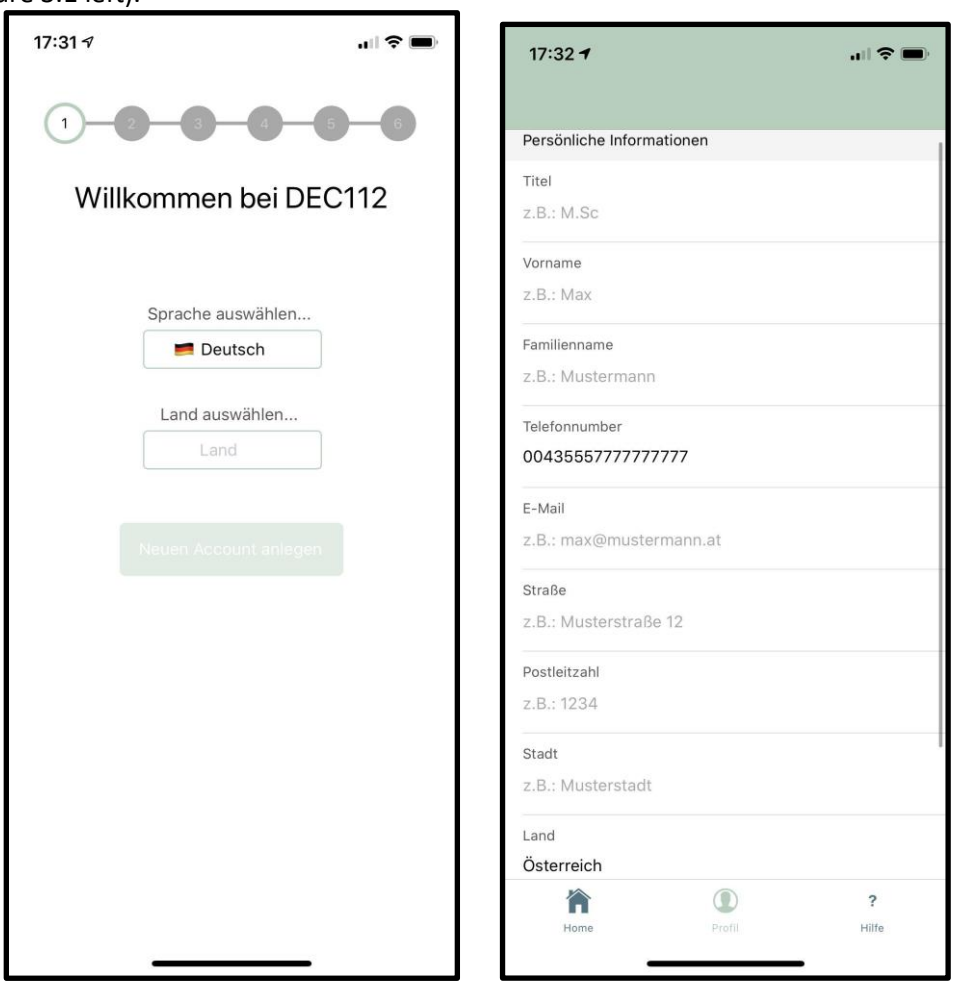


Figure 3.1: Registration Wizard and Profile Screen

After successful completion the user can enter personal data on the profile screen (Figure 3.1 right).

The main screen provides controls for all available emergency services (Figure 3.1 left). And the DEC112 App also has a separate test mode which allows the user to test emergency chats for the respective services (Figure 3.2 right).

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

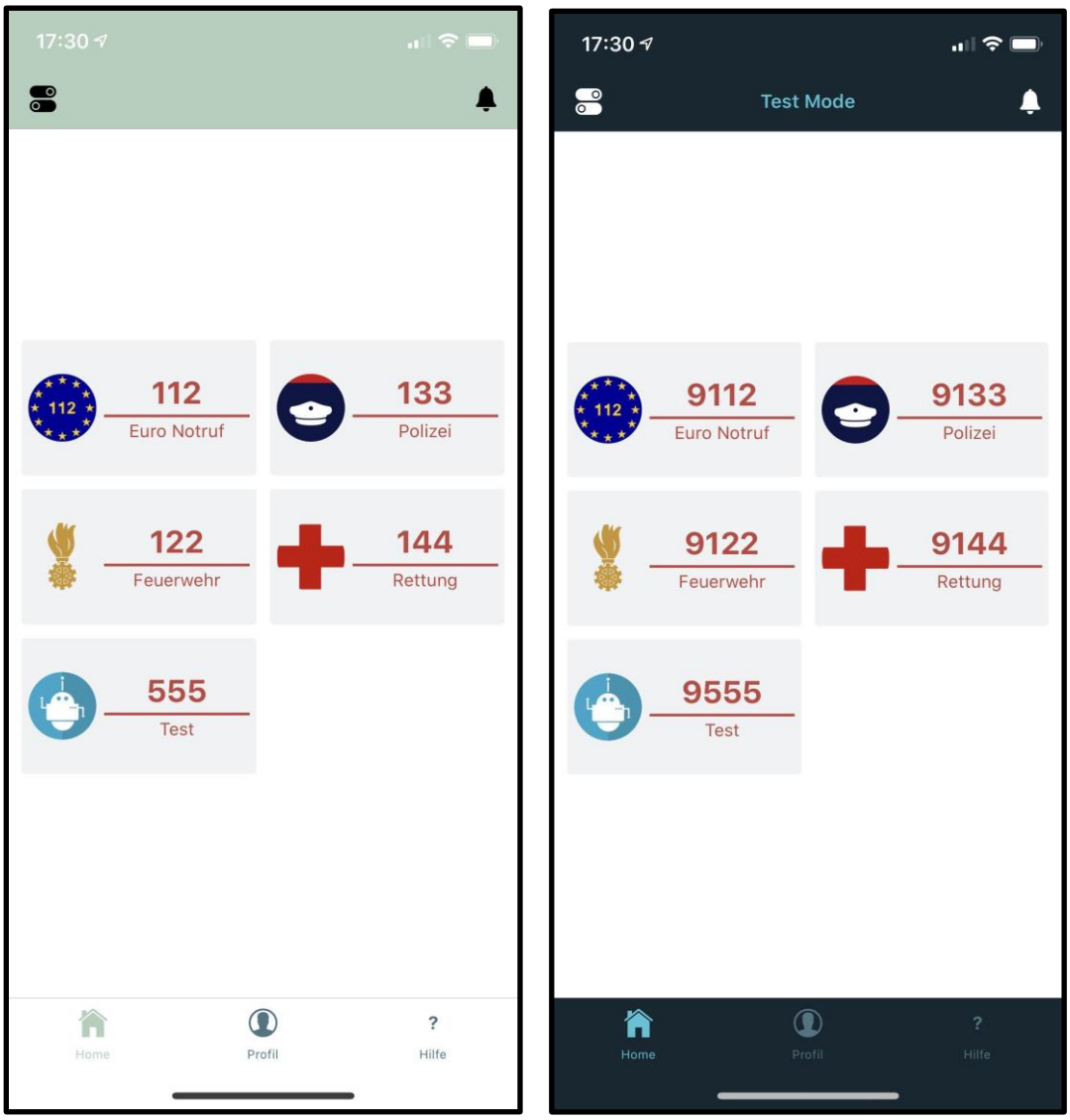


Figure 3.2 Main Screen (left) and Test Mode (right)

3.2 Registration & Root API

The DEC112 Registration API is responsible for managing DEC112 registrations and to provide local configuration information for any clients - in this project the DEC112 App. These configuration options include available emergency services (Fire, Police, Health) and training options.

The Root API provides a list of DEC112 registries and country specific resources. A user selects during client setup the country of residence and chooses from the list of available options provided by the Root API.

3.3 Data Vault

The OwnYourData Data Vault is a Personal Data Store based on the following principles:

- *Open & Free:* all code is available on Github under https://github.com/ownyourdata/oyd-pia2 and the service is available for free under https://data-vault.eu

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

- *Secure:* personal data is stored encrypted (asymmetric encryption allows plugins to write data using a public key) and data processing services are executed in a restricted environment ("sandbox") to ensure information is not leaked externally
- *No lock-in:* choose where you want to store and process your data / move data and services to your trusted location
- *Extensible:* plugins allow to add any 3rd party services (data sources, data processing services, and visualization methods) with fine-granular permission settings
- *Audited:* to guarantee immutability and timeliness of all operations (login, plugin changes, data operations) a linked audit log is maintained and hash values are stored in the Ethereum blockchain
- *Standard compliant:* where applicable available standards are used (authentication, data formats) and through active collaboration in various groups and organizations additional standards will be adopted

In the DECTS project the Data Vault holds personal information automatically provided during an emergency chat and consent information about training chats. The relevant data flows / workflows for managing personal and consent information are described in detail in chapter 4 and include:

- *User Registration* - see section 4.2
  triggered when a user creates an account in the DEC112 App and chooses to store personal information in the Data Vault and reference it through a DID
- *User Deletion* - see section 4.3
  triggered when a user removes the account in the DEC112 App
- *Personal Information Provision during an Emergency Chat* - see section 4.5
  triggered by an emergency chat that includes a DID referencing personal information in the Data Vault
- Storing Consent on a Chat Protocol - see section 4.6
  triggered when a user gives consent to use the chat protocol of a training chat
- Updating Consent for a Chat Protocol - see section 4.7
  triggered when a user updates/revokes consent for a given chat protocol

The functionality for managing personal and consent information is provided through 2 plugins: DEC112 and Consent.

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

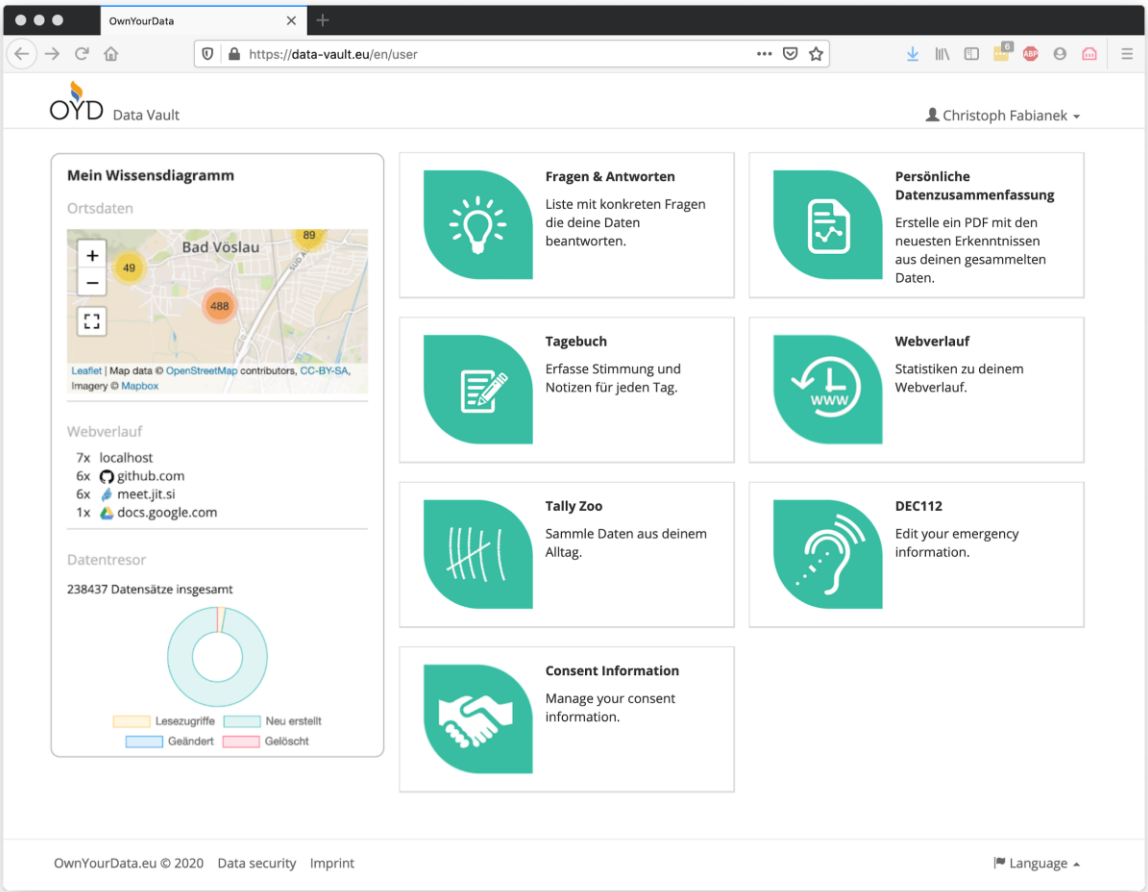


Figure 3.1: Data Vault with DEC112 and Consent Plugin installed

The DEC112 plugin displays the personal information provided during an emergency chat and allows editing.

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

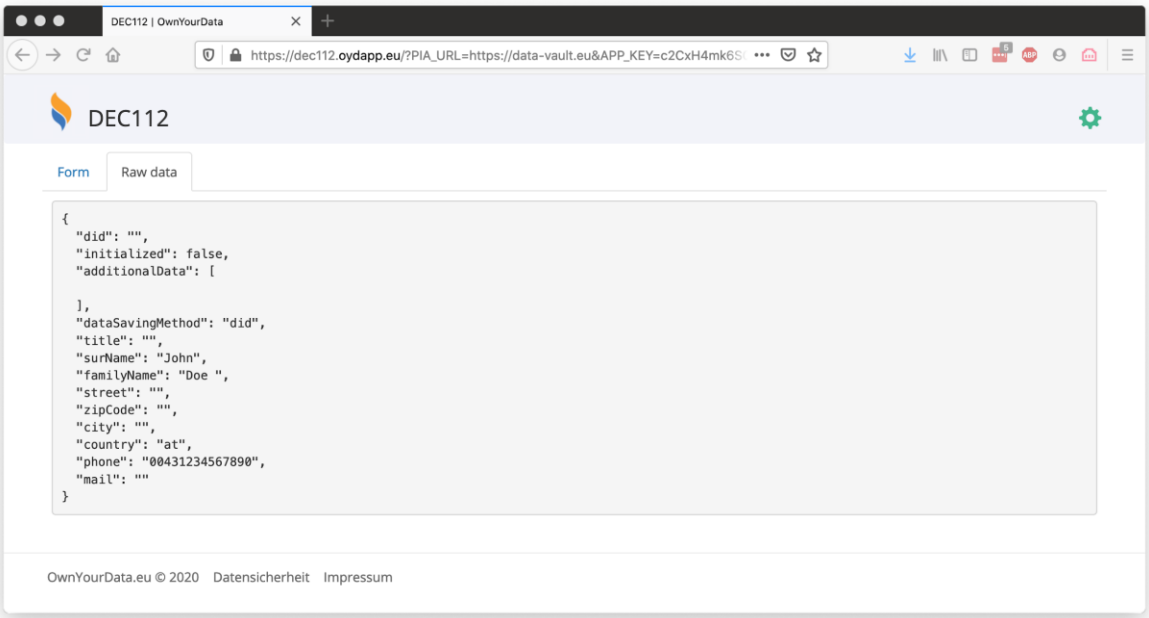


Figure 3.2: DEC112 Plugin

The Consent plugin displays existing consent records and allows individual editing.

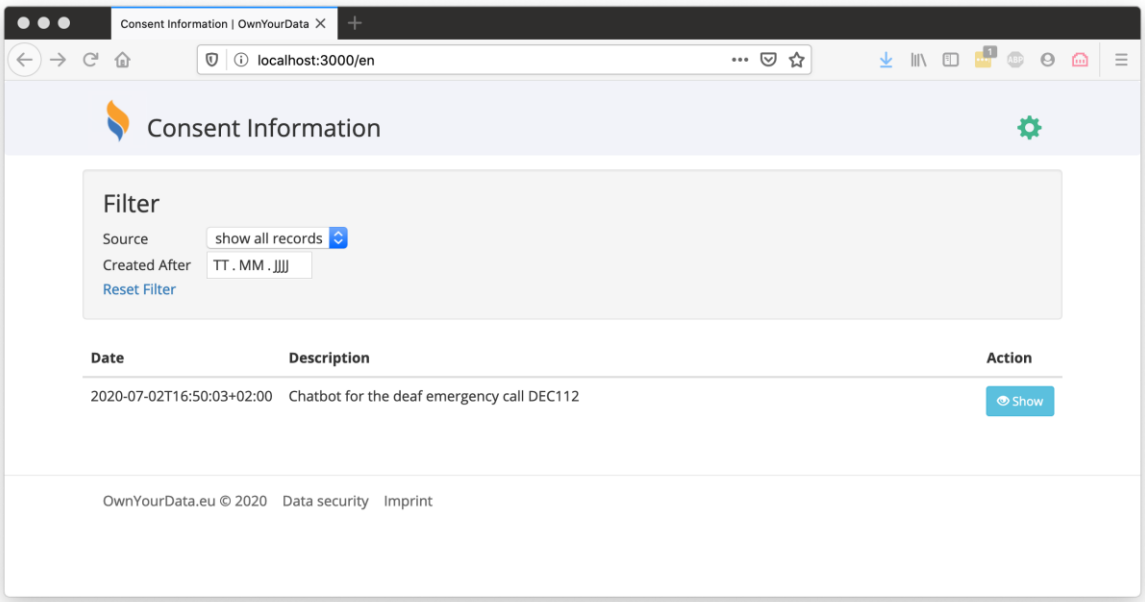


Figure 3.3: Consent Plugin

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

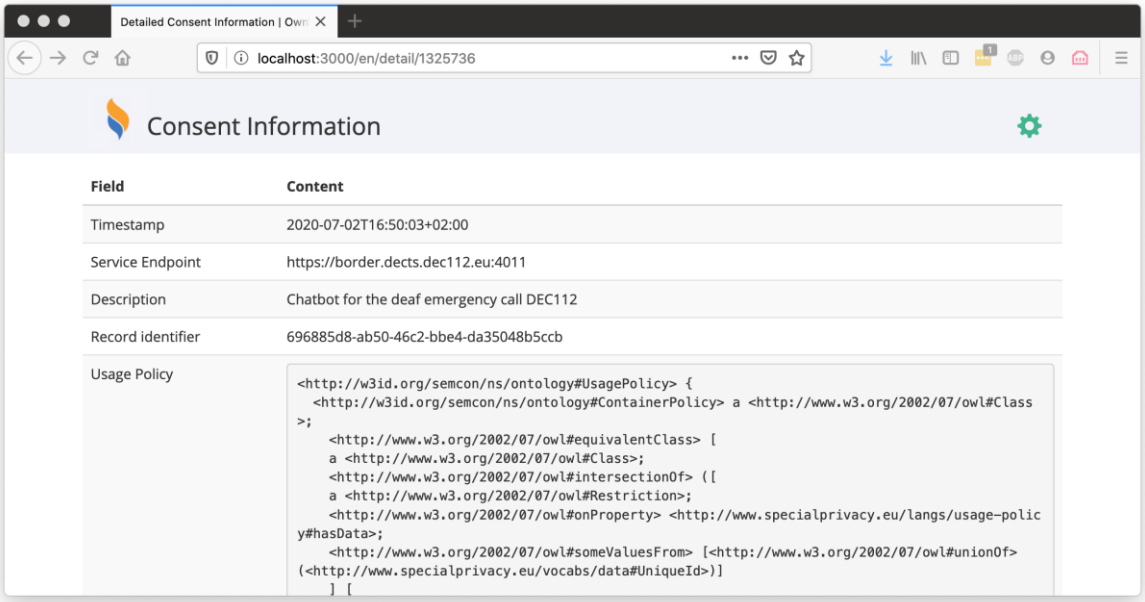


Figure 3.4: Detailed Consent Information

3.4 ESRP, ECRF & PRF

ECRF, ESRP and PRF are core services of a next generation emergency IP network. The emergency call routing function (ECRF) supports the location to service translation (LoST) protocol by which location information and a service uniform resource name (URN) serve as input to a mapping function that returns a uniform resource identifier (URI) addressing the most appropriate PSAP for the caller's location.

The emergency services routing proxy (ESRP) is the base routing function for emergency calls, and the primary input to an ESRP is a SIP message, which means that only the call setup via SIP signaling is routed through intermediate functional elements. Media (audio, video or text) is transmitted end to end. The output is a SIP message with a route header (possibly) rewritten and, in some cases, additional manipulation of the message content. To do this, the ESRP maintains an interface to the ECRF for location-based routing information. Emergency calls are routed to the appropriate PSAP based on the location of the caller

The policy routing function (PRF), typically a component of an ESRP, determines the next hop in the SIP signaling path using a policy. Policies consist of rules that include conditions (like for instance a certain SIP header field and value, date or time) and corresponding actions. Actions may be additional header fields to be added or new routing targets (different to the mapping provided by the ECRF). Figure 3.5. gives an example of rules structured in a yaml file to be used in order to route specific numbers or users to service endpoints configured for DECTS.

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

```yaml
# prf rule 0
- rule: DECTS default
  id: R0
  priority: 1
  default: sip:border@border.dects.dec112.eu;transport=tcp
  transport: tcp
  - actions:
    add: >
      Call-Info: <urn:dec112:endpoint:chat:service.dec112.at>;purpose=dec112-ServiceId
    route: sip:border@border.dects.dec112.eu
# prf rule 1
- rule: DECTS Emergency Chat
  id: R1
  priority: 1
  default: sip:border@border.dects.dec112.eu;transport=tcp
  transport: tcp
  - conditions:
    header: >
      To: sip:112@root.dects.dec112.eu,
      To: sip:122@root.dects.dec112.eu,
      To: sip:133@root.dects.dec112.eu,
      To: sip:144@root.dects.dec112.eu
  - actions:
    add: >
      Call-Info: <urn:dec112:endpoint:chat:service.dec112.at>;purpose=dec112-ServiceId
    route: sip:border@border.dects.dec112.eu
# prf rule 2
- rule: DECTS Training Chat
  id: R2
  priority: 1
  default: sip:border@border.dects.dec112.eu;transport=tcp
  transport: tcp
  - conditions:
    header: >
      To: sip:9122@root.dects.dec112.eu,
      To: sip:9133@root.dects.dec112.eu,
                                                                    1,1           Top
```

Figure 3.5: PRF rules yaml file

3.5 Border

At an Emergency Control Center (or Public Safety Answering Point) a call taker answers emergency calls and follows a predefined workflow. Text messages are quite different to a plain voice call but must be integrated in the same workflow. The Border implements a gateway to connect to the other DEC112 services (described above) and triggers - adaptable to local needs - functions in the control center upon receiving an emergency message.

3.6 Chat Bot

The Chat Bot is based on a Semantic Container[1] to provide a number of data management functionalities: logging, data access & tracing, Usage Policies, and immutable data records. It is extended by the functionality to register for chat messages at the Border (see section 3.5) and provides based on a configurable rule set predefined answers for incoming messages. At the end of each chat a participant is asked if the chat messages can be shared for training purposes. In case consent is given and the participant uses a DID to specify a trusted service endpoint the consent information is sent there. In this project the OwnYourData Data Vault (see section 3.3) is used as a trusted service endpoint.

[1] https://OwnYourData.eu/semcon

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

3.7 PI2 (Personal Identifiable Information)

The PI2 service provides for the Viewer a convenient endpoint to resolve DIDs, query emergency information provided by the specified service endpoint and decrypts this emergency information using the Shamir Secret Sharing Scheme in case the Personal Data Store supports this type of encryption.

3.8 Viewer

The Viewer displays incoming emergency chats provided by the Border. It is a reference implementation for Emergency Control Centers that do not yet integrate emergency chat messages in their call processing equipment. The viewer is a web-based UI that displays location, reference data and chat features.

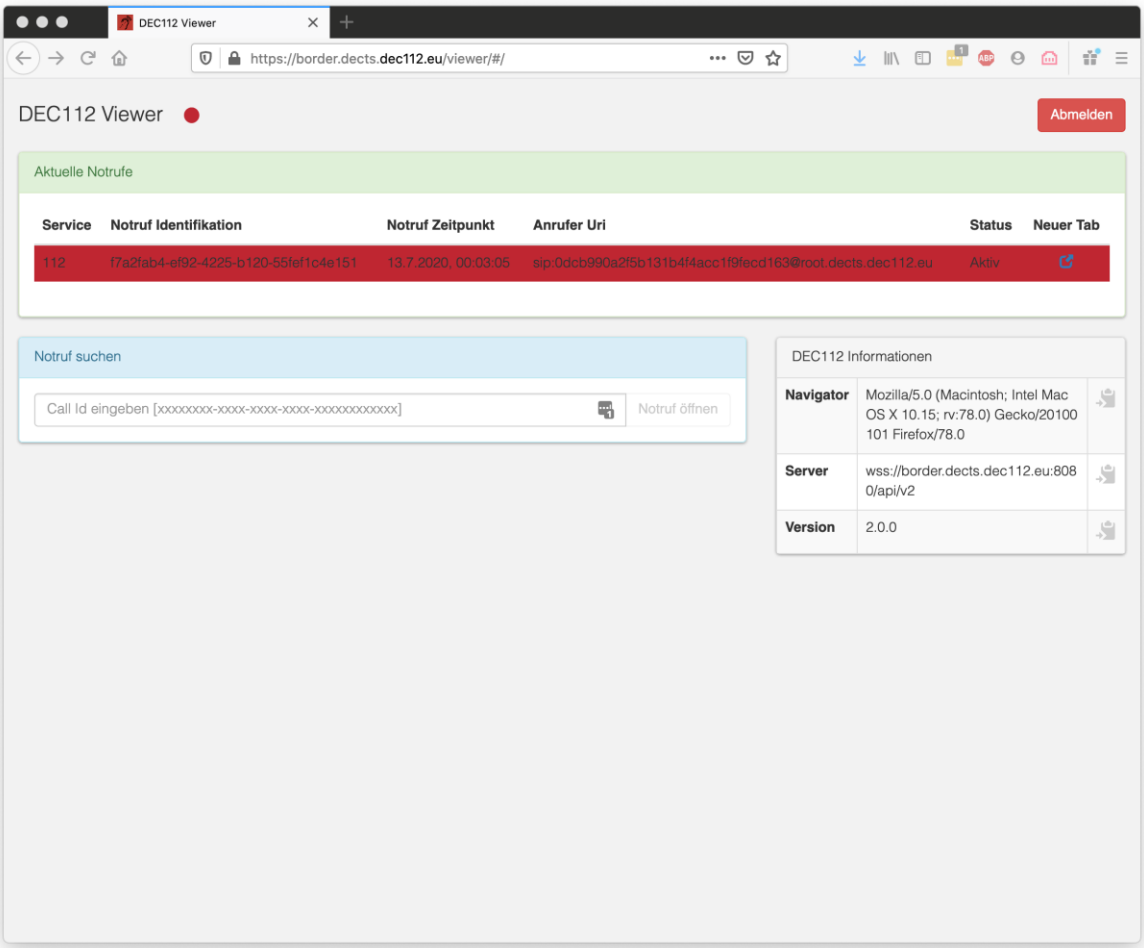


Figure 3.6: Overview of Emergency Chats

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

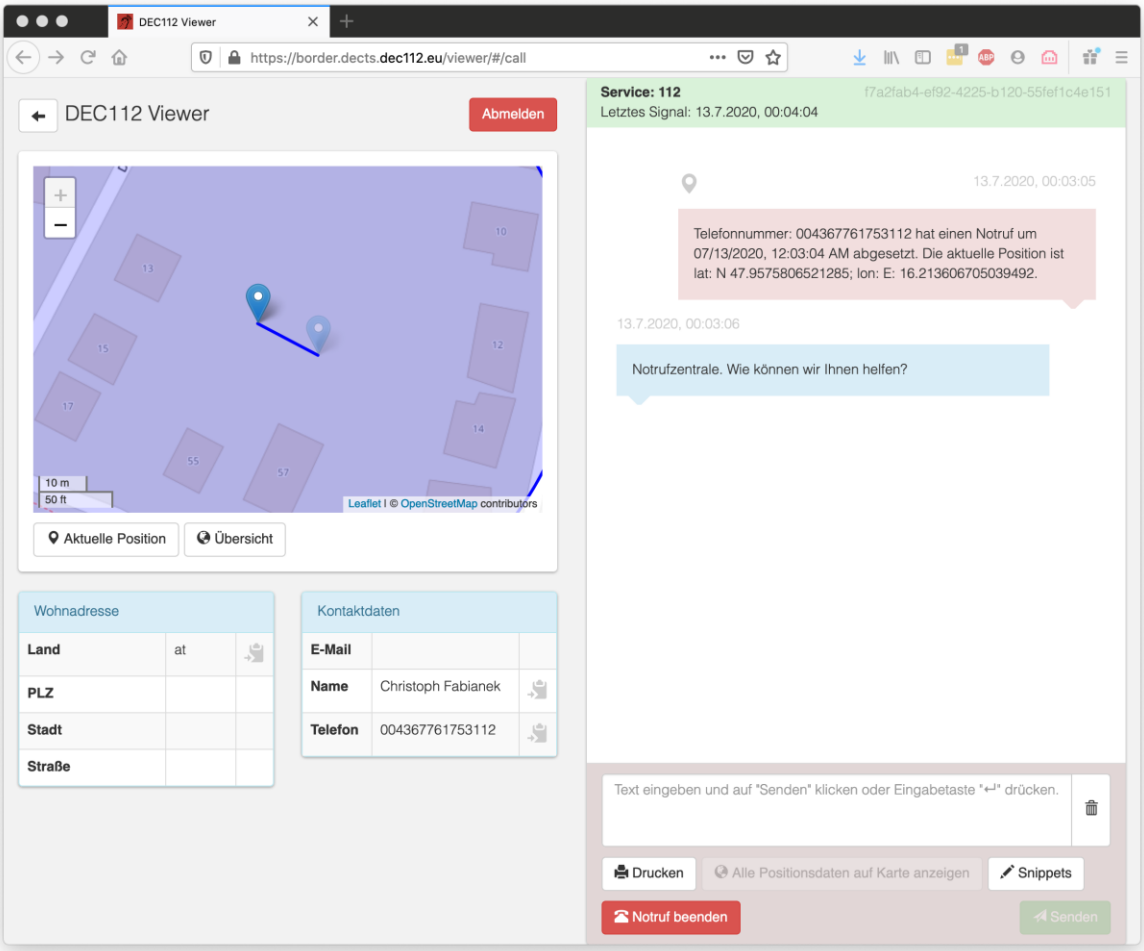


Figure 3.7: Chat View for an Emergency Chat

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

4 Workflows & Interfaces

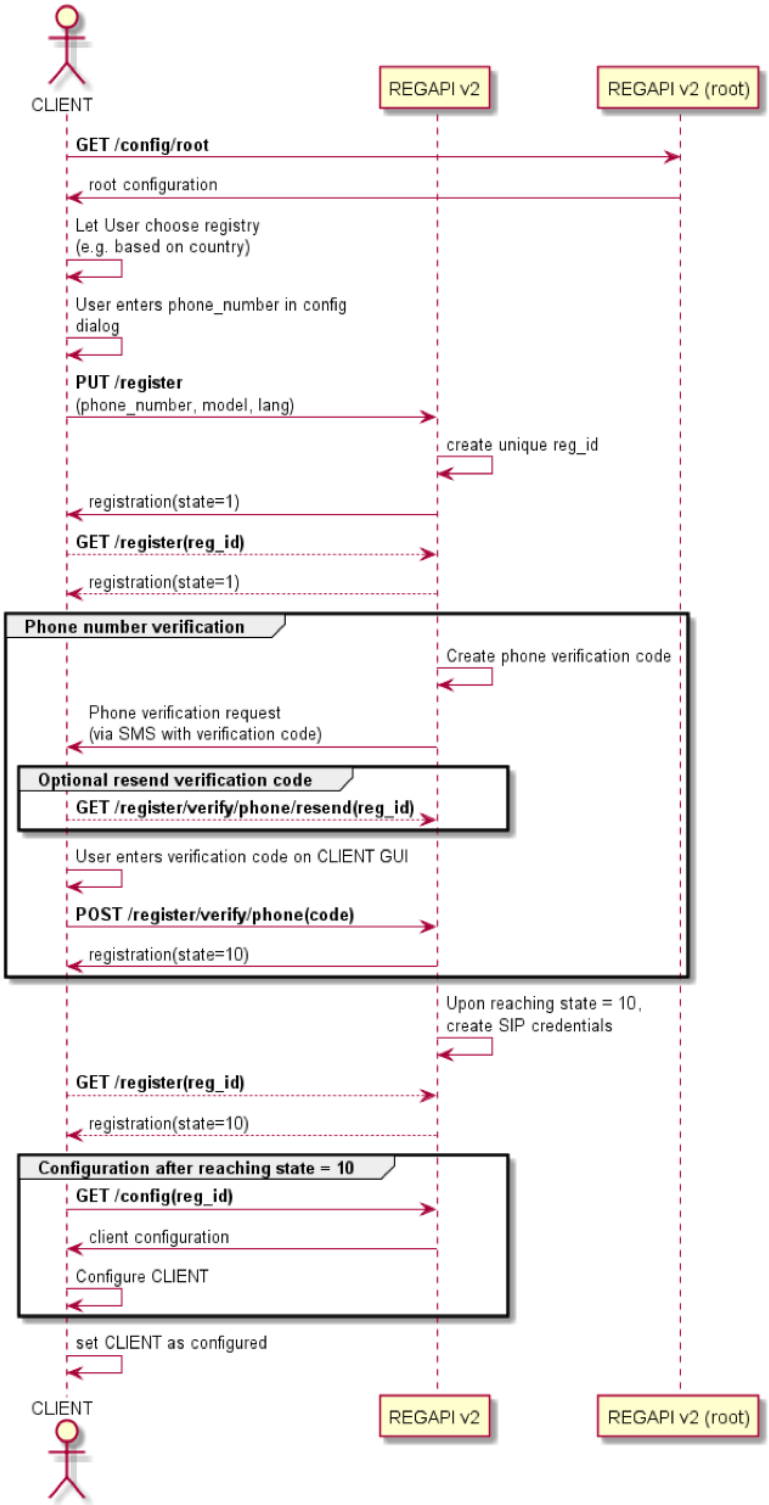


Figure 5.1: Registration Procedure

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

4.1 DEC112 App Startup

The root configuration provides to the DEC112 App a list of DEC112 registries and other resources (e.g. language resources).

API Endpoint for Root Configuration

A user can select the local registry during initial registration..

REST Request to retrieve root configuration:

```
http method: GET
http://host:port/api/v2/config/root?api_key={api_key}
```

Response example:

```
{
 "root": {
          "country_registries": [
          "name": "AT",
          "registration_api: [
          "type": "dec112_v1",
          "url": "https://service.dec112.at"
          }
          ]
 },
 "runtime_ms": "0.49305"
}
```

4.2 User Registration

DEC112 uses the concept of an arbitrary “registration” to identify a DEC112 user. The Registration-API does not store any personal data. A phone number verification can optionally be enabled server side. In this case a phone number is needed and is only stored until successfully verification. A registration is uniquely identified by a reg_id generated server side during registration.

API Endpoint for User Registration

Creates a new registration.

The optional model attribute contains information about the device a user is using. Its purpose is to help identify and solve possible DEC112 App problems on different platforms. If not available or not provided, “unknown” will be used.

To provide localized content in the DEC112 App and responses from REGAPI an optional lang attribute containing an ISO639-1 two letter language code can be specified. If omitted the first available language configured server-side will be used which, in a default installation is “en” for English.

An optional phone number is required to receive the SMS verification code.

NGI_Trust project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 825618.

REST Request

```
http method: PUT
http://host:port/api/v2/register?api_key={api_key}
{
  "model": "Fairphone 3",
  "lang": "de",
  "phone_number": "004312345678"
}
```

Response example:

```
{
  "reg_id": "5b80637f-df3b-fe13-de27-6791a5178028",
  "lang": "de",
  "state": 1,
  "phone_verified_ts": null,
  "registered_ts": "2018-04-27T10:13:02.453Z",
  "runtime_ms": "0.49305"
}
```

API Endpoint for Deleting a User

Deletes an existing registration. Any CLIENT must call this method upon forced deregistration initiated through the user by means of CLIENT GUI or when the CLIENT software is uninstalled.

REST Request

```
http method: DELETE

http://host:port/api/v2/register/{reg_id}?api_key={api_key}
```

Response example:

```
{
  "runtime_ms": "0.49305"
}
```

API Endpoint for Checking Registration Status

Get the state of an existing registration. This method allows a CLIENT to determine the current state of a registration. The CLIENT is only allowed to enable full DEC112 functionality at registration state 10.

REST Request

```
http method: GET

http://host:port/api/v2/register/{reg_id}?api_key={api_key}
```

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

Response example:

```
{
  "reg_id": "5b80637f-df3b-fe13-de27-6791a5178028",
  "lang": "de",
  "state": 1,
  "phone_verified_ts": null,
  "registered_ts": "2018-04-27T10:13:02.453Z",
  "runtime_ms": "0.49305"
}
```

API Endpoint for Verification Code

Verifies a registration phone number. This method is only useful if the REGAPI server is configured with enabled phone number verification.

REST Request

```
http method: POST

http://host:port/api/v2/register/verify/phone/{reg_id}?api_key={api_key}
{
  "code": "4711-0815"
}
```

Response example:

```
{
  "reg_id": "5b80637f-df3b-fe13-de27-6791a5178028",
  "lang": "de",
  "state": 10,
  "phone_verified_ts": null,
  "registered_ts": "2018-04-27T10:13:02.453Z",
  "runtime_ms": "0.49305"
}
```

API Endpoint for Verification Code Resending

Allows resending a possibly lost phone verification SMS code up to two times.

REST Request

```
http method: POST

http://host:port/api/v2/register/verify/phone/{reg_id}/resend?api_key={api_key}
```

Response example:

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

```
{
  "runtime_ms": "0.49305"
}
```

API Endpoint for Handling Additional User Information

Handle additional, arbitrary data for a registration. Any type of data can be stored in a configured Personal Data Store that makes this data available through a Decentralized Identifier (DID).

This method can only be used when registration state equals 10 (fully verified) otherwise an error will be returned.

Note: When the server is configured to delete a phone number after successful verification the phone_number attribute in the request is required. When the server is configured to store a registrations phone number after successful registration, the phone_number attribute in the request is ignored and the registrations stored phone number is used instead.

Note: This is a convenience function. Additional functions to manage private data stored using a DID must be performed on the CLIENT or manually by the user on the DID specified service endpoint.

REST Request

```
http method: POST

http://host:port/api/v2/register/data/{reg_id}?api_key={api_key}
```

Response example:

```
{
  "runtime_ms": "0.49305"
}
```

API Endpoint for Requesting Configuration

Returns the CLIENT configuration for a given registry ID. Configuration contains SIP credentials needed to connect and authenticate on SIP server. Credentials are automatically generated on the server during initial registration when (optional) phone number verification successfully completes and contains no personal data.

Also included in the response is a list of available services a user can call. A CLIENT must configure its GUI elements (call buttons / icons) according to the provided service list.

REST Request

```
http method: GET

http://host:port/api/v2/config/{reg_id}?api_key={api_key}
```

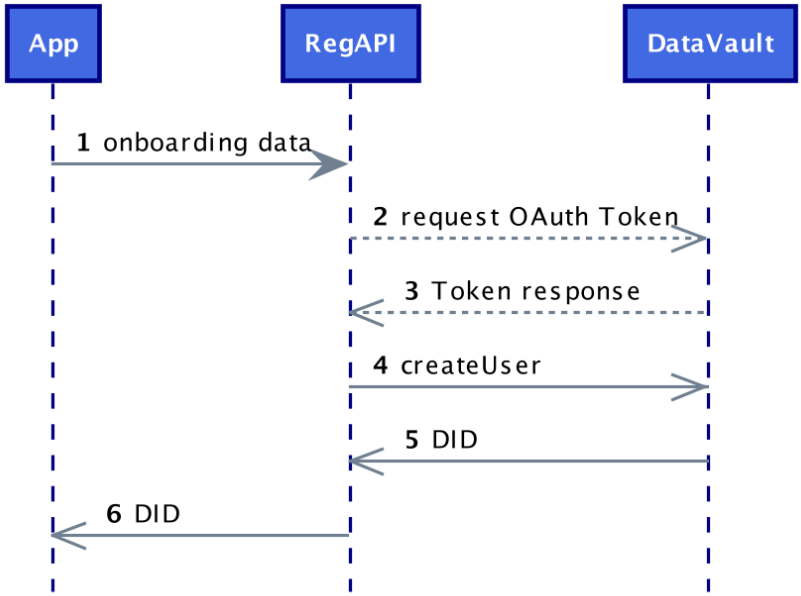
Response example:

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

```
{
 "reg_id": "5b80637f-df3b-fe13-de27-6791a5178028",
 "server": "ws://service.dec112.at",
 "public_id": "sip:6da87ddcfa10ac65ea00437f@service,dec112.at",
 "private_id: "6da87ddcfa10ac65ea00437f",
 "password": "4460af63bd663ffc109",
 "realm": "service.dec112.at",
 "services": [
          {
          "urn": "urn:service.police",
          "enabled": true,
          }
 ],
 "runtime_ms": "0.49305"
}
```


4.2.1 Storing Additional Information in the Data Vault

When the user chooses to store the additional information provided during the onboarding process on a service endpoint provided in a DID the DEC112 App sends the additional information to the RegAPI which forwards it to the Data Vault and receives a DID that is then stored in the DEC112 App.



To store the additional information in the Data Vault the following steps are performed:

1. DEC112 App sends request as described in section 5.2 User Registration

   RegAPI confirms that the phone number already exists in the system and then forwards the information to the Data Vault

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

2. RegAPI requests an token to write to the Data Vault:
   HTTP POST https://data-vault.eu/oauth/token
   header: application/json
   body:
   {
   client_id: <App-Key provided by OwnYourData>,
   client_secret: <App-Secret provided by OwnYourData>,
   grant_type: "client_credentials"
   }
3. Data Vault responds upon confirmation of the id/secret with a token (valid for 2 hours)
   response:
   {
   access_token: <access token>
   }
4. RegAPI sends additional information to the Data Vault:
   HTTP POST https://data-vault.eu/api/dec112/register
   header: "Authorization: Bearer <token>"
   body:
   {
   phone_number: wizardData.profile.phone,
   payload: any information provided by the user
   }
5. Data Vault validates that the phone number does not exist and creates a new user and stores the provided data for this user; then it generates a DID with a DID Document pointing to the Data Vault
   response:
   {
   did: <DID>
   }
6. RegAPI responds to the request from step 1
   response:
   {
   did: <DID>
   }

Error scenarios:

- status 500: if no valid DID was received in time (the DEC112 App will inform the user that managing the data via DID was not successful and falls back to storing the data on the phone)

4.3 Account Deletion

When a user requests to delete the account in the DEC112 App a request is sent to the RegAPI with the following steps:

1. DEC112 App sends request as described in section 5.2 User Registration

   RegAPI confirms that the phone number already exists in the system, deletes the account and if a DID was provided forwards this to the Data Vault

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

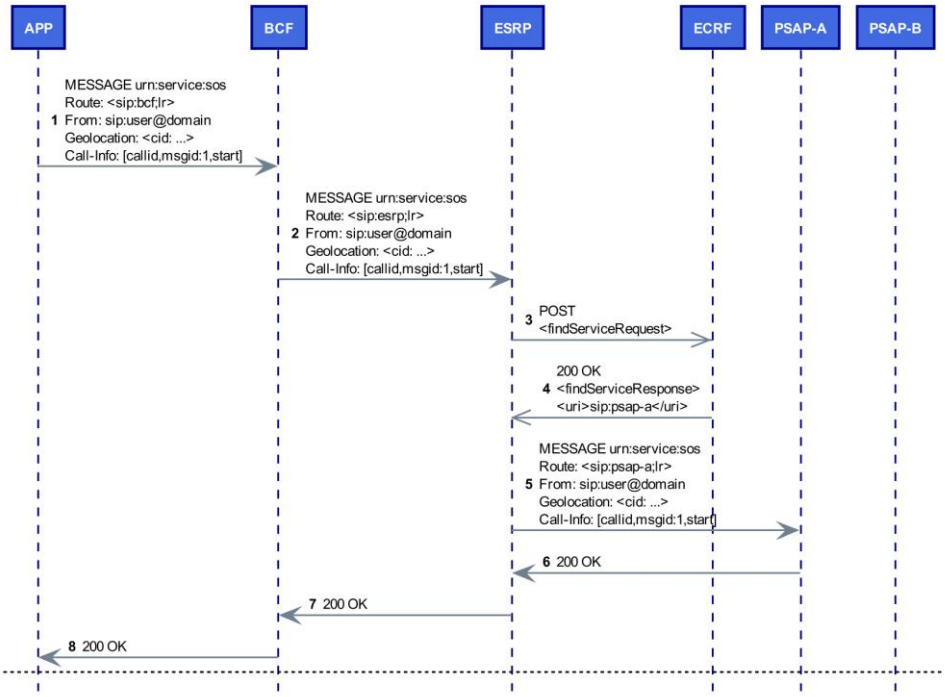
2. RegAPI requests an token to write to the Data Vault:
   HTTP POST https://data-vault.eu/oauth/token
   header: application/json
   body:
   {
   client_id: <App-Key provided by OwnYourData>,
   client_secret: <App-Secret provided by OwnYourData>,
   grant_type: "client_credentials"
   }
3. Data Vault responds upon confirmation of the id/secret with a token (valid for 2 hours)
   response:
   {
   access_token: <access token>
   }
4. RegAPI forwards information about the delete request to the Data Vault:
   HTTP DELETE https://data-vault.eu/api/dec112/revoke
   header: Authorization: Bearer <token>
   body:
   {
   phone_number: wizardData.profile.phone
   }
5. Data Vault confirms phone number and deletes available information
   response: status 200
6. RegAPI confirms account delete by responding to the request from step 1
   response: status 200

4.4 Emergency Chat Routing

When a user initiates an emergency chat, the initial request sent to the trusted provider SIP proxy is to register the device/user identity (refer to User Registration) with the proxy. This ensures that messages sent from a PSAP to the DEC112 App are properly routed. Immediately, after successful registration the following messages are sent (refer to the message sequence chart below):

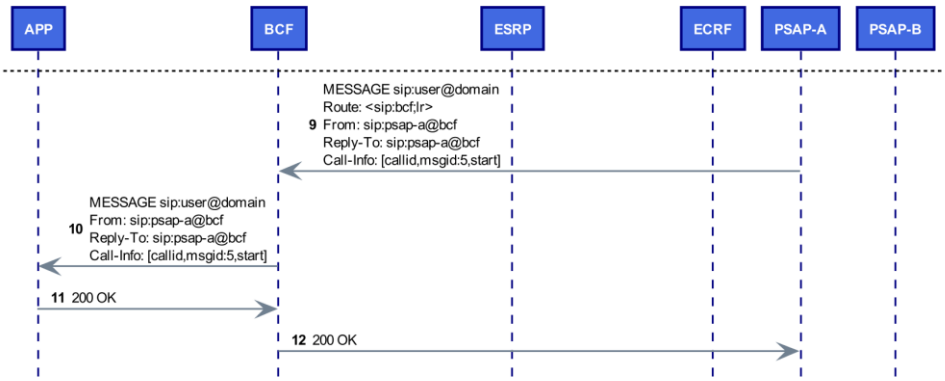
1. Initial message (marked as start message) forwarded via the trusted provider SIP proxy or BCF (Border Control Function) including location information and service urn and DID.
2. The BCF (i.e. trusted provider SIP proxy) forwards the request to its configured ESRP (located within the regional ESInet, refer to Architecture, work done and current status).
3. The ESRP requests routing information at the ECRF by issuing a *findServiceRequest*.
4. The ECRF respond includes the authoritative mapping for the given service urn and location (*findServiceResponse*).
5. After consulting the PRF (Policy Routing Function; not shown in the diagram), the message is forwarded to the PSAP that serves the region including the location of the calling device (note that PSAP is synonym for any functional element involved in receiving the message (refer to Architecture, work done and current status).
6. (7. and 8.) Final response (200 OK) routed back to the origin.

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.



In response to the first message received from the DEC112 App, the PSAP sends a first message back to the application including information how the PSAP can be reached for further message exchange as illustrated in the following message sequence chart:

9. The trusted provider SIP proxy or BCF receives the response message sent by the PSAP
10. The message is being forwarded to the DEC112 App.
11. (12.) Final response (200 OK) routed back to the origin.

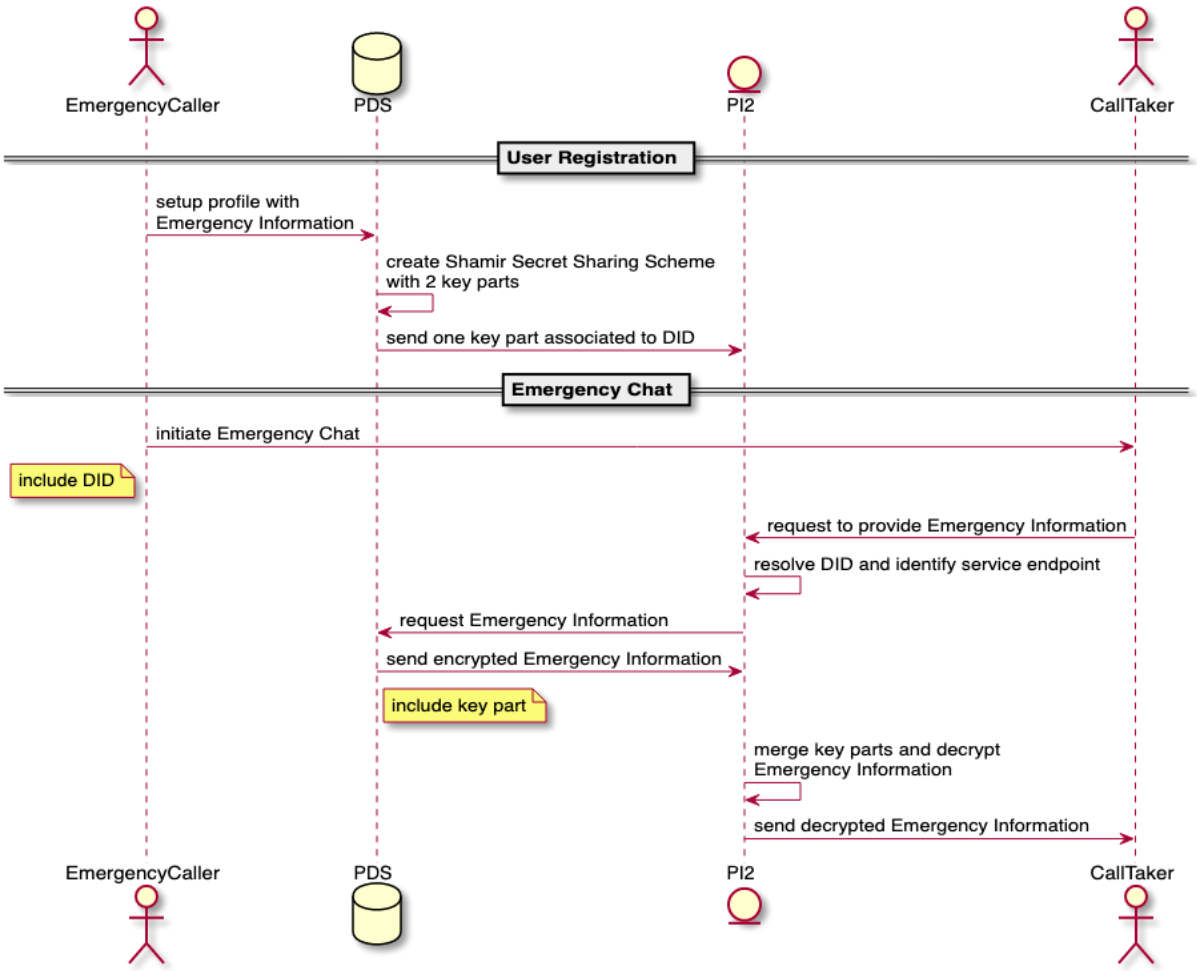


Right after the initial message exchange DEC112 App and PSAP maintain information that identifies a conversation and further messages are exchanged without the involvement of ESRP and ECRF. A specific message identifier is used to close such a conversation. In parallel, a PSAP may request personal information using the DID received with the first message as described in the following chapter.

4.5 Personal Information Provision during an Emergency Chat

When a user stores his or her profile data in a Personal Data Store (referenced in a DID) the following sequence diagram describes the process of delivering emergency information to the Call Taker.

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.



It is important to note that data is stored fully encrypted in the PDS and can only be decrypted by the Emergency Caller logging in with SMS authentication (providing a one-time-password for login) or using the key pair generated on account setup based on Shamir's Secret Sharing Scheme. One part of the key is stored in the PDS and the other part is stored in registered PI2s hosted by emergency service organizations.

4.6 Storing Consent on a Chat Protocol

- User confirms use of chat protocol for training purposes at the end of a training chat
- reference to data together with usage policy is sent to Data Vault for storage

4.7 Revoking Consent for a Chat Protocol

- User revokes consent for given training chat
- Data Vault sends request to Chatbot to update/revoke consent
- Chatbot (Semantic Container) revokes consent information and recursively forwards revocation to subsequent recipients of training chat data

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

5 System Testing

5.1 Test Cases

To demonstrate the functionality of the system the following end-to-end test cases were used for verification.

1) User Registration

   Prerequisites:

   - DEC112 App installed on smartphone
   - Root Registry and Registration API set up
   - Data Vault accepts connections from Registration API

   Test:

   - start DEC112 App and follow onboarding wizard
   - verify phone number via SMS
   - provide further information in onboarding wizard
   - choose to store emergency information in Personal Data Store
   - log in to PDS via phone number
   - edit emergency information

   Verification:

   - user has status 10 in RegAPI
   - DID and DID document is created
   - emergency information is stored encrypted in the Data Vault
   - PI2 received key part for decrypting emergency information

2) Training Chat

   Prerequisites:

   - DEC112 App installed and registered
   - Root Registry and Registration API set up
   - ESRP, ECRF and Border configured to route training chat
     - 555 - Chatbot
     - 122 - Fire
     - 133 - Police
     - 144 - Health
   - Chatbot set up
   - Data Vault accepts connections from Chatbot

   Test:

   - initiate Chatbot
   - navigate through Chatbot messages
   - confirm use of chat history
   - visit usage terms page on DEC112 website
   - show usage policy on Data Vault
   - change policy
   - try to retrieve chat history from chatbot

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

Verification:

- correct language on DEC112 App and chatbot
- chatbot responses are correct
- usage policy is stored in Data Vault
- chatbot (Semantic Container) adheres to chosen usage policy

3) Emergency Chat

Prerequisites:

- DEC112 App installed and registered
- Root Registry and Registration API set up
- ESRP, ECRF and Border configured to route training chat
  - 555 - Chatbot
  - 122 - Fire
  - 133 - Police
  - 144 - Health
- Viewer and PI2 set up

Test:

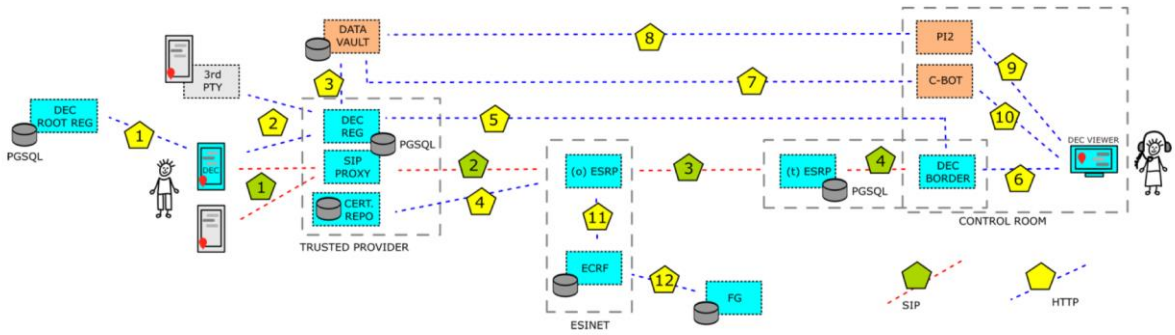
- Initiate emergency chat
- answer chat on viewer
- additional information is displayed

Verification:

- correct language on DEC112 App and Viewer
- correct additional information is displayed on viewer

5.2 End-to-End Tests

In the course of the project a number of end-to-end tests were performed to track development and ensure successful integration of the numerous components of DECTS.



root.dects.dec112.eu IP: 194.182.173.16 intern: 10.182.173.16 (dects-srv-a)	esrp.dects.dec112.eu IP: 194.182. 174.130 intern: 10.182. 174.130 (dects-srv-a)	border.dects.dec112.eu IP: 194.182.174.186 intern: 10.182.174.186 (dects-srv-c)

Figure 6.1 DECTS system

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

5.2.1 May 2020

The tests were performed on May 5th, 2020.

1) User Registration

Prerequisites: all components were available and configured

- ***Root Registration*** available for Austria (AT) http://root.dects.dec112.eu/api/v2/config/root?api_key=DEC2017%23dev
- available call buttons on DEC112 App:
  - 112 - Chatbot
  - 122 - Fire
  - 133 - Police
  - 140 - Bergrettung
  - 144 - Health
  - 555 - Test (Echo Service)
- ***DEC112 2.0 App*** available as version 0.0.9 through Testflight, Google Beta, and Expo client; available languages: English, German, French - known limitations:
  - dedicated test mode for agency specific emergency chats not available yet => emergency button 112 is configured to be answered by chatbot
  - placeholders in configuration from Root Registration are not yet parsed => demo mode is used for emergency & test chats
- ***Registration API*** available and configured for DECTS environment
- ***Data Vault:*** user registration is only available as hard-coded API endpoint; open items:
  - create user during user registration
  - encrypt data with phone number
  - create DID for user
  - implement login with phone number & SMS code
  - implement plugin for editing personal data
  - create shared secret and send key-part to PI2
  - show any access to personal data in Audit Log

Tests: the following steps were performed to test User Registration on an iPhone (6S, iOS 13.4.1) and Android (Nexus 5, Android 6).

- start DEC112 App and follow onboarding wizard - tested with German, English, and French UI; observations:
  - step #1 - on switch to FR, drop-down is empty
  - step #2 - not used?
    - note: reserved for future use to backup and restore and account
  - step #3 - Drop-down in FR version empty
  - step #4 - in FR mix of English and French text
  - step #5 - no SMS code is sent (not reproducible)
  - step #6 - no text in FR version (also message box empty)
  - step #7 - no text in FR version
  - create profile - no text in FR version
  - create profile DID selection - no text in FR version
  - switching to debug mode and entering password does not work in FR
- verify phone number via SMS - not working

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

- choose to store emergency information in Personal Data Store (DID Method)- OK
- provide further information in onboarding wizard - after clicking save: no information is sent to Data Vault
- log in to PDS via phone number - no available yet
- edit emergency information - not available yet

2) Training Chat

Prerequisites: all components were available and configured

- ***Root Registration*** available for Austria (AT)
- ***DEC112 2.0 App*** available as version 0.0.9
- ***Registration API*** available and configured for DECTS environment
- **ESRP & ECRF incl. PRF** available and configured
  - 112 - Chatbot
  - 122 - Fire
  - 133 - Police
  - 140 - Bergrettung
  - 144 - Health
  - 555 - Test (Echo Service)
- **Border** available and configured for DECTS environment
- **Chatbot** available and configured; available languages German, English, French
- **Data Vault** no interface is available yet to store consensus information for test chats

Tests: the following steps were performed to test the Chatbot

- initiate Chatbot observation:
  - DEC112 App: when initiating a chat and no endpoint is available there is no button to end the call setup
  - DEC112 App: Android version does not deliver calls (not reproducible)
  - DEC112 App: routing of calls to sip:112@service.dec112.at (should be sip:112@root.dects.dec112.eu)
- navigate through Chatbot messages - OK for German language; English and French setup not yet available
- confirm use of chat history - OK
- visit usage terms page on DEC112 website - webpage https://www.dec112.at/de/chatbot not available
  - note: URLs will be http://www.dec112.eu/chatbot?language=de or http://www.dec112.at/chatbot?language=en
- show usage policy on Data Vault - not available yet
- change policy - not available yet
- try to retrieve chat history from chatbot - not available yet

3) Emergency Chat

Prerequisites: all components were available and configured

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

- ***Root Registration*** available for Austria (AT)
- ***pjchat*** command line tool to initiate chats in the DEC112 environment
- ***Registration API*** available and configured for DECTS environment
- **ESRP & ECRF incl. PRF** available and configured
    - 112 - Chatbot
    - 122 - Fire
    - 133 - Police
    - 140 - Bergrettung
    - 144 - Health
    - 555 - Test (Echo Service)
- **Border** available and configured for DECTS environment
- **Viewer** available and configured with version v2.0.0-beta.2; available languages German, English, French
- **PI2** available with a basic interfaces to demonstrate resolving a DID and returning back information
- **Data Vault** no interface is available yet to provide personal information based on PI2 requests

Tests: the following steps were performed to test Emergency Chats

- Initiate emergency chat - use pjchat to initiate an emergency chat to 122: successful
- answer chat on viewer - incoming chat is indicated in the viewer - accepting call works - ending chat works
- displaying additional information via DID resolving - Viewer restarts (logs off) upon receiving a call that includes a DID in a Call-Info header Call-Info: <did:example:123456789abcdefghi>;purpose=EmergencyCallData.DID
- switch between different languages - EN works fine - DE works fine - FR only shows EN language labels

<u>Learning from 1st End-to-End Tests</u>

- mandatory conference call with all participants before the test to discuss setup and configuration
- smoke test with pjchat and Echo bot to ensure functionality
- monitor logs from Live-System to identify chats that are routed wrong

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

5.2.2 June 2020

The tests were performed on June 9th, 2020.

1) User Registration

Prerequisites: all components were available and configured

- ***Root Registration*** available for Austria (AT) http://root.dects.dec112.eu/api/v2/config/root?api_key=DEC2017%23dev
- available call buttons on DEC112 App:
  - Live Mode
    - 112 - Chatbot
    - 122 - Fire
    - 133 - Police
    - 144 - Health
    - 555 - Test (Echo Service)
  - Test Mode
    - 112 - Chatbot -> routed to English chatbot
    - 122 - Fire -> routed to German chatbot
    - 133 - Police -> routed to German chatbot
    - 144 - Health -> routed to German chatbot
    - 555 - Test (Echo Service)
- ***DEC112 2.0 App*** available as version 0.3.6-ota on iOS and Android devices provided through Expo client; available languages: English, German, French - known limitations:
  - placeholders in configuration from Root Registration are not yet parsed => demo mode is used for emergency & test chats
- ***Registration API*** available and configured for DECTS environment
- ***Data Vault:*** user registration is available with the following limitations:
  - create fake DID for user
  - implement login with phone number & SMS code
  - implement plugin for editing personal data
  - create shared secret and send key-part to PI2
  - show any access to personal data in Audit Log

Tests: the following steps were performed to test User Registration on an iPhone (6S, iOS 13.4.1) and Android (Nexus 5, Android 6).

- start DEC112 App and follow onboarding wizard - tested with German, English, and French UI; observations:
  - message to enter profile data ("Profil aktualisieren") is only shown for a second and then invisible; only when restarting the app the message dialog keeps showing (@Mario)
  - improve text for choosing DID storage method (@Christoph)
  - no error message when wrong SMS code is entered (@Mario)
  - deleting verification code is really hard (@Mario)
  - french translation (@Mario)
    - page 1: "Select a language…", "Select a country…" not translated
    - page 5: not translated
    - page 6: not translated
    - buttons not (Police, Ambulance) and Heading "Test Mode" not translated

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

        - Profile storage-selection-method-page not translated
        - Family Name: "FNom..."?
        - French: message box when starting chat: German text & French buttons
        - German: message box when starting chat: English text & German buttons
        - German: Heading "Test Mode" not translated
- choose storage methode - tested with local storage and DID; observations:
    - fake DID (@Christoph)
- log in to PDS via phone number - observations:
    - not yet possible via SMS (@Christoph) - using hash value of phone number makes it possible to login
- edit emergency information - observations:
    - information is only displayed but not yet possible to be edited (@Christoph)
- delete registration - observations:
    - clicking "Anmeldung löschen" on help screen nothing happened; clicked again a white screen was displayed and app need to be restarted; upon restart the onboarding wizard was displayed (@Mario)
    - clicking "Anmeldung löschen" on help screen started the onboarding wizard but already filled out with data from last onboarding; restarting fixed the app fixed the problem (@Mario)

2) Training Chat

Prerequisites: all components were available and configured

- ***Root Registration*** available for Austria (AT)
- ***DEC112 2.0 App*** available as version 0.0.9
- ***Registration API*** available and configured for DECTS environment
- **ESRP & ECRF incl. PRF** available and configured
- **Border** available and configured for DECTS environment
- **Chatbot** available and configured in German and English
- **Data Vault** no interface is available yet to store consensus information for test chats

Tests: the following steps were performed to test the Chatbot

- initiate chat to Chatbot - OK
- navigate through Chatbot messages - OK for German and English language - observations: - strange rendering with footer displayed twice: (@Mario)

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

- confirm use of chat history - OK
- visit usage terms page on DEC112 website - OK
- show usage policy on Data Vault - not available yet
- change policy - not available yet
- try to retrieve chat history from chatbot - not available yet

3) Emergency Chat

Prerequisites: all components were available and configured

- ***Root Registration*** available for Austria (AT)
- ***pjchat*** command line tool to initiate chats in the DEC112 environment
- ***Registration API*** available and configured for DECTS environment
- **ESRP & ECRF incl. PRF** available and configured
- **Border** available and configured for DECTS environment
- **Viewer** available and configured with version v2.0.0-rc.2; available languages German, English, French
- **PI2** available with interfaces (requiring authentication) to demonstrate resolving a DID and returning back information; secret sharing for encrypted emergency information is not available yet
- **Data Vault** provides emergency information but not yet encrypted; however PI2 needs to authenticate against the Data Vault to retrieve data

Tests: the following steps were performed to test Emergency Chats

- Initiate emergency chat from DEC112 App and pjchat - OK
- answer chat on viewer - incoming chat is indicated in the viewer - accepting call works - ending chat works - observations:
  - the name of the user shall not be displayed in the first text message but only the phone number (@Mario)
  - the name of the user shall not be displayed in the last text message ("Christoph13 Fabianek (Phone: 004367761753112) hat den Notruf um 06/11/2020, 10:27:45 PM beendet.") but only the phone number (@Mario)

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

  - the first message response ("Emergency Control Center here! How can we help you?") is always in English. Where can this be configured? (@Richard)
  - currently, the date is displayed twice in the first text message: "...Notruf um 06/09/2020, Jun 9, 2020..." this should be time HH:MM:SS and then date MM/DD/YYYY (@Mario)
  - there should be an indication (maybe a checkmark) when a message was successfully delivered from the Viewer to the DEC112 App (@Gabriel, @Richard)
  - kill DEC112 App and send message from Viewer -> Viewer terminates (@Gabriel, @Richard); also: logging in again, there is a message that the App was put in the background (when it is actually not available anymore)
  - when the Viewer ends a chat this is displayed at the DEC112 App but it is still possible to enter text messages in the App; it's also necessary to end the call (together with confirmation) to end the call at the DEC112 App after being informed that the call ended (@Mario)
  - Android app crashes when sending a message while the Viewer is closed at the same time (@Mario)
  - Android app crashes when ending a call in the DEC112 App (@Mario)
- displaying additional information via DID resolving - OK
- switch between different languages
  - EN works fine
  - DE works fine
  - FR works fine

<u>Learning from 2nd End-to-End Tests</u>

- process (team call, pre-tests, documentation) worked very well
- introduce JIRA for bug tracking
- roadmap for inputs (new features) from Christoph
  - July tests: access log, consensus info from chatbot, SMS login in Data Vault, encrypted emergency information
  - August tests: DEC112 plugin for editing emergency information, Consensus plugin for editing consensus data, updating consensus information in Chatbot
  - September tests: full-fledged DID creation

5.2.3 July 2020

The tests were performed on June 14th and 16th, 2020 and any observations where documented in JIRA[2]

<u>1) User Registration</u>

Prerequisites: all components were available and configured

- ***Root Registration*** available for Austria (AT) http://root.dects.dec112.eu/api/v2/config/root?api_key=DEC2017%23dev
- available call buttons on DEC112 App:
  - Live Mode

[2] https://meecode.atlassian.net/secure/RapidBoard.jspa?rapidView=14&projectKey=DECTS&view=planning&selectedIssue=DECTS-48&issueLimit=100

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

        - 112 - Chatbot
        - 122 - Fire
        - 133 - Police
        - 144 - Health
        - 555 - Test (Echo Service)
    - Test Mode
        - 112 - Chatbot -> routed to English chatbot
        - 122 - Fire -> routed to German chatbot
        - 133 - Police -> routed to German chatbot
        - 144 - Health -> routed to German chatbot
        - 555 - Test (Echo Service)
- ***DEC112 2.0 App*** available as version 0.3.7-ota on iOS and Android devices provided through Testflight; available languages: English, German, French
- ***Registration API*** available and configured for DECTS environment
- ***Data Vault:*** user registration is available with the following limitations:
    - create fake DID for user
    - editing personal data is not available yet (only display)
    - create shared secret and send key-part to PI2 not available yet

Tests: the following steps were performed to test User Registration on an iPhone (6S, iOS 13.4.1) and Android (Nexus 5, Android 6).

- start DEC112 App and follow onboarding wizard - tested with German, English, and French UI
- log in to PDS via phone number
- edit emergency information, observation:
    - information is only displayed but not yet possible to be edited
- delete registration

<u>2) Training Chat</u>

Prerequisites: all components were available and configured

- ***Root Registration*** available for Austria (AT)
- ***DEC112 2.0 App*** available as version 0.0.9, 0.3.7-ota
- ***Registration API*** available and configured for DECTS environment:
    - 9112 - mapped to English Chatbot
    - 9133 - mapped to French Chatbot with agency type Police
    - 9122 - mapped to German Chatbot with agency type Fire
    - 9144 - mapped to German Chatbot with agency type Health
- **ESRP & ECRF incl. PRF** available and configured
- **Border** available and configured for DECTS environment
- **Chatbot** available and configured in German, English, and French
- **Data Vault** no interface is available yet to store consensus information for test chats

Tests: the following steps were performed to test the Chatbot

- initiate chat to Chatbot
- navigate through Chatbot messages
- confirm use of chat history

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

- visit usage terms page on DEC112 website
- show usage policy on Data Vault
- revoke consent - not available yet
- try to retrieve chat history from chatbot - not available yet

3) Emergency Chat

Prerequisites: all components were available and configured

- ***Root Registration*** available for Austria (AT)
- ***pjchat*** command line tool to initiate chats in the DEC112 environment
- ***Registration API*** available and configured for DECTS environment
- **ESRP & ECRF incl. PRF** available and configured
- **Border** available and configured for DECTS environment
- **Viewer** available and configured with version v2.0.0
  available languages German, English, French
- **PI2** available with interfaces (requiring authentication) to demonstrate resolving a DID and returning back information; secret sharing for encrypted emergency information is not available yet
- **Data Vault** provides emergency information but not yet encrypted; however, PI2 needs to authenticate against the Data Vault to retrieve data

Tests: the following steps were performed to test Emergency Chats

- Initiate emergency chat from DEC112 App and pjchat
- answer chat on viewer - incoming chat is indicated in the viewer - accepting call works - ending chat works
  observations:
  - the name of the user shall not be displayed in the first text message but only the phone number (@Mario)
  - the name of the user shall not be displayed in the last text message ("Christoph13 Fabianek (Phone: 004367761753112) hat den Notruf um 06/11/2020, 10:27:45 PM beendet.") but only the phone number (@Mario)
  - the first message response ("Emergency Control Center here! How can we help you?") is always in English. Where can this be configured? (@Richard)
  - currently, the date is displayed twice in the first text message: "...Notruf um 06/09/2020, Jun 9, 2020…" this should be time HH:MM:SS and then date MM/DD/YYYY (@Mario)
  - there should be an indication (maybe a checkmark) when a message was successfully delivered from the Viewer to the DEC112 App (@Gabriel, @Richard)
  - kill DEC112 App and send message from Viewer -> Viewer terminates (@Gabriel, @Richard); also: logging in again, there is a message that the App was put in the background (when it is actually not available anymore)
  - when the Viewer ends a chat this is displayed at the DEC112 App but it is still possible to enter text messages in the App; it's also necessary to end the call (together with confirmation) to end the call at the DEC112 App after being informed that the call ended (@Mario)
  - Android app crashes when sending a message while the Viewer is closed at the same time (@Mario)

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

    - Android app crashes when ending a call in the DEC112 App (@Mario)
- displaying additional information via DID resolving - OK
- switch between different languages - EN works fine - DE works fine - FR works fine

5.2.4 September 2020

The tests were performed on August 16th and 17th, 2020 and any observations where documented in JIRA[3].

1) User Registration

Prerequisites: all components were available and configured

- ***Root Registration*** available for Austria (AT) http://root.dects.dec112.eu/api/v2/config/root?api_key=DEC2017%23dev
- available call buttons on DEC112 App:
    - Live Mode
        - 112 - Chatbot
        - 122 - Fire
        - 133 - Police
        - 144 - Health
        - 555 - Test (Echo Service)
    - Test Mode
        - 9112 -> routed to English chatbot
        - 122 - Fire -> routed to French chatbot and indicate type Fire
        - 133 - Police -> routed to German chatbot and indicate type Police
        - 144 - Health -> routed to Spanish chatbot and indicate type Health
        - 555 - Test (Echo Service)
- **DEC112 2.0 App** available as version 0.2.1, 0.5.3-ota on iOS and Android devices provided through Testflight; available languages: English, German, French, Spanish, Romanian
- **Registration API** available and configured for DECTS environment
- **Data Vault** available and configured

Tests: the following steps were performed to test User Registration on an iPhone (6S, iOS 13.7) and Android (Nexus 5, Android 6).

- start DEC112 App and follow onboarding wizard - tested with German, English, and French UI
- log in to PDS via phone number
- edit emergency information, observation:
    - information is only displayed but not yet possible to be edited
- delete registration

Observations:

3 https://meecode.atlassian.net/secure/RapidBoard.jspa?rapidView=14&projectKey=DECTS&view=planning&selectedIssue=DECTS-48&issueLimit=100

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

- the SMS text with the verification code during the registration process is in German when "German" is selected as language and otherwise "English" for English, Spain, French, and Romanian

A screen recording for the User Registration walk-through is available here:
https://www.loom.com/share/687576d4f9ab4789b0323bd97fc6814a

2) Emergency Chat

Prerequisites: all components were available and configured

- ***Root Registration*** available for Austria (AT)
- ***pjchat*** command line tool to initiate chats in the DEC112 environment
- ***Registration API*** available and configured for DECTS environment
- **ESRP & ECRF incl. PRF** available and configured
- **Border** available and configured for DECTS environment
- **Viewer** available and configured with version v3.0.0
  available languages German, English, French
- **PI2** available with interfaces (requiring authentication) to demonstrate resolving a DID and returning back information including secret sharing for encrypted emergency information
- **Data Vault** provides encrypted emergency information

Tests: the following steps were performed to test Emergency Chats

- Initiate emergency chat from DEC112 App and pjchat
- answer chat on viewer - incoming chat is indicated in the viewer - accepting call works - ending chat works
- displaying additional information via DID resolving - OK
- switch between different languages - EN, DE, ES, FR, and RO work fine

The screen recording above (https://www.loom.com/share/687576d4f9ab4789b0323bd97fc6814a) includes the demonstration for making an emergency chat.

3) Training Chat

Prerequisites: all components were available and configured

- ***Root Registration*** available for Austria (AT)
- ***DEC112 2.0 App*** available as version 0.2.1, 0.5.3-ota
- ***Registration API*** available and configured for DECTS environment:
  - 9112 - mapped to English chatbot
  - 9122 - mapped to French chatbot with agency type Fire
  - 9133 - mapped to German chatbot with agency type Police
  - 9144 - mapped to Spanish chatbot with agency type Health
- **ESRP & ECRF incl. PRF** available and configured

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

- **Border** available and configured for DECTS environment
- **Chatbot** available and configured in German, English, Spanish, French, and Romanian
- **Data Vault** interface and plugin available to store and manage consensus information for training chats

Tests: the following steps were performed to test the Chatbot and Consent Management

- initiate chat to Chatbot
- navigate through Chatbot messages
- confirm use of chat history
- visit usage terms page on DEC112 website
- show usage policy on Data Vault
- display data tracing
- copy data from Chatbot to another Semantic Container
- display updated data tracing
- revoke consent and verify recursive propagation of revocation

A screen recording for using the chatbot including a walk-through of the Consent Management plugin in the OwnYourData Data Vault is available here:
https://www.loom.com/share/aa3750be693147b8b3ea20bd8659bada

6 Conclusions

This document outlined the design based on the requirements identified in Deliverable 2.2. The design was gradually implemented and verified through monthly End-to-end Tests. Feedback from internal reviews and user groups was taken into account and especially privacy considerations were discussed lively. The implementation is available on Github in the following repositories:

- https://github.com/ownyourdata
- https://github.com/sem-con
- https://github.com/DEC112

The project will continue the development of the DEC112 solution and aims for a European-wide deployment.

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.

Appendix

Glossary

Below is a list of acronyms and abbreviations used throughout the document.

Abbr.	Definition
DEC112	Deaf Emergency Call
DID	Decentralized Identifier
ESRP	Emergency Service Routing Proxy
JSON	JavaScript Object Notation
PDS	Personal Data Store
PI2	Personally Identifiable Information - service component to resolve a DID and provide associated emergency information
PSAP	Public Safety Answering Point

NGI_Trust project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825618.