RE4DY
MANUFACTURING DATA NETWORKS

RE4DY TOOLKIT

Name of the Tool	eIDAS
Tool Owner	Industry Commons Foundation
Version	1.0
Date	Nov 2025
Version	V1.0

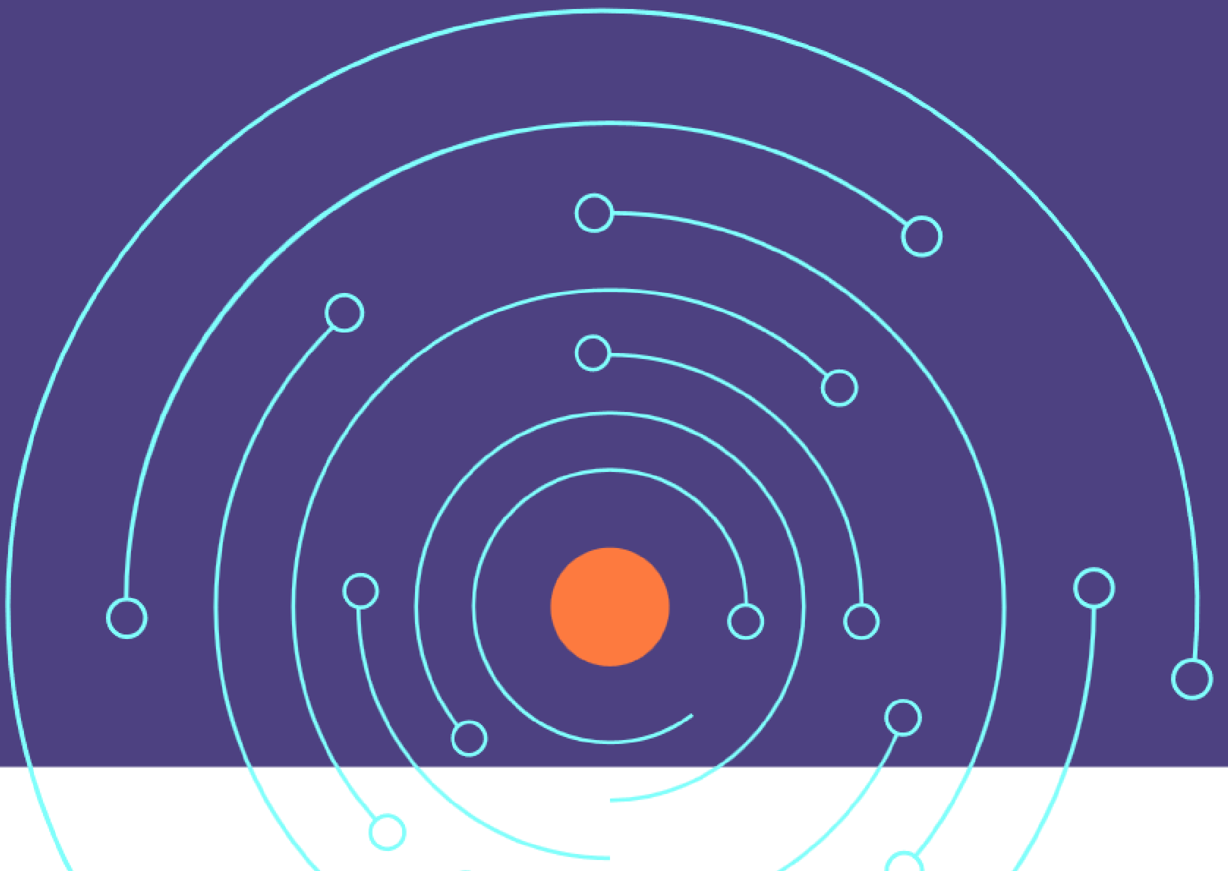

Table of contents

1. Component Description

The eIDAS solution adopts an electronic identification scheme that may be used as proof of identity for individuals across Europe. It enables citizens of a European Member State to verify their identity when using online services in other Member states, thus facilitating electronic transactions. The eIDAS network is composed of a number of eIDAS nodes, one per Member State implemented at the national level, and each node may either request or provide cross-border authentication. In the RE4DY context, the functionality provided by eIDAS will be utilized for the identity management of the RE4DY user base. In particular, the eIDAS node will provide authentication for individuals joining the RE4DY platform during registration, allowing secure access to our services.

2. Input

Since the eIDAS node does not use a database, but plain text configuration files to record users, we have developed a Python API which is responsible for making the necessary changes in these files so as to create new users. When adding a new eIDAS user to the system, it is necessary to record a set of attributes such as first and last names, national identification number and gender. These attributes constitute the input of the endpoint implemented by our Python API. It is also possible to create new, custom attributes that may be stored along with the given user information.

Furthermore, the first phase of the user registration process requires first authenticating with eIDAS. We have developed an authentication back-end using Spring Boot which is responsible for communicating with the eIDAS node during this process. The back-end service provides an endpoint through which authentication is possible. It accepts a username and password which are then forwarded to the eIDAS node to fulfill the request.

3. Output

In the eIDAS node, upon successful user creation, the API returns the message “User inserted Successfully”. If the username or national identification number of the user provided have already been recorded under a user that already exists, the process fails with “User Already Exists”.

Our Spring Boot authentication back-end is responsible for interfacing with the eIDAS node to achieve authentication. Upon successful authentication with eIDAS, it returns a JSON object, eidas_response, embedded with all the related attributes stored in the eIDAS node as well as an assertion from the eIDAS node.

4. Information Flow

Adding an eIDAS user

An eIDAS administrator is able to add a new eIDAS user through the eIDAS Management Web UI. The administrator inserts user credentials and information into a form and the Management Web UI forwards them to the eIDAS API. Since the eIDAS node does not use a database but file-based storage for users, the relevant files are modified so that the new user is registered in the eIDAS node. If the user was inserted successfully, the administrator user is notified in the front-end.

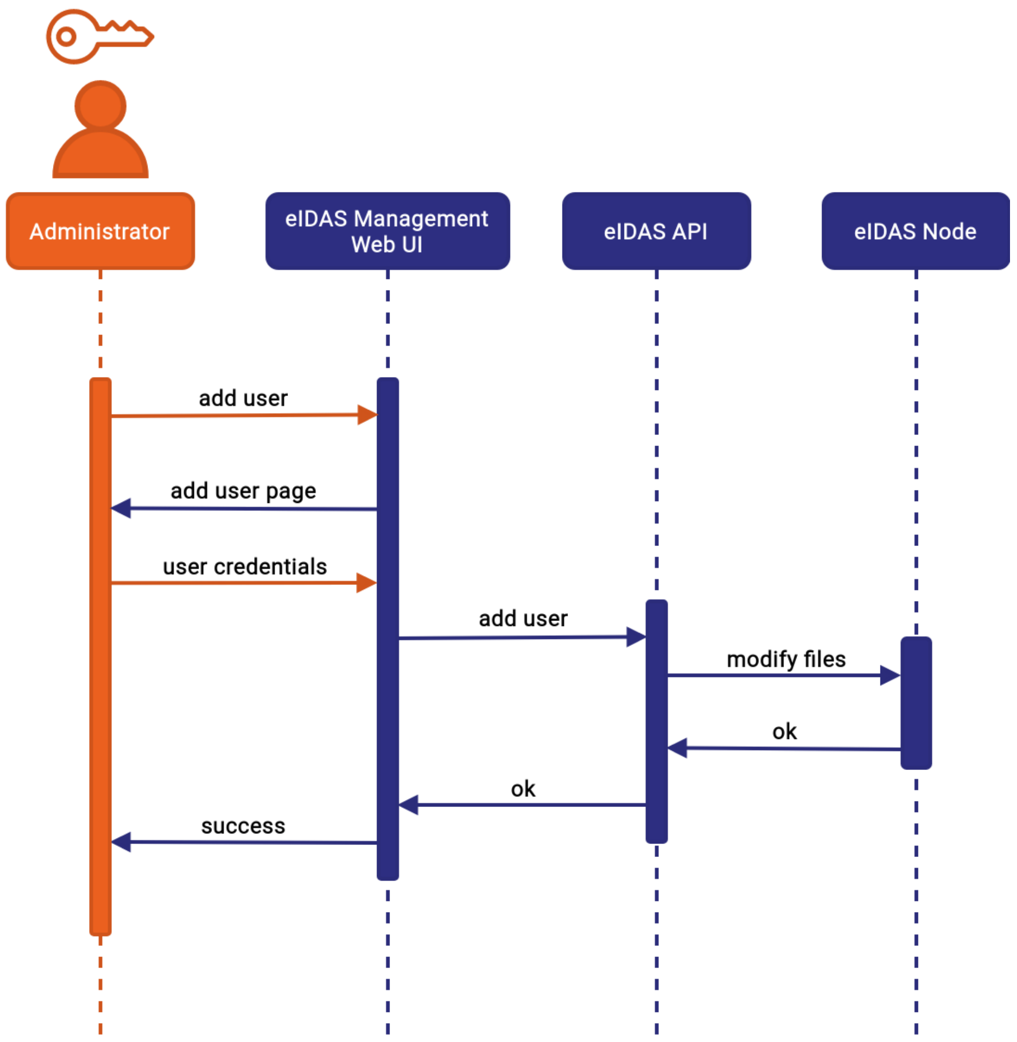

Figure 1 UML Diagram of the administrator adding a new eIDAS user to the system and the communication of the eIDAS management front-end, eIDAS API and eIDAS node.

RE4DY Registration

Registering begins through the RE4DY front-end application where the user is initially presented with an eIDAS login form to insert their eIDAS credentials. The credentials are sent from the front-end to the authentication back-end service and from there they are sent to the eIDAS node for authentication. If the credentials match with an existing eIDAS user, the eIDAS node returns an access token to the authentication back-end which is then forwarded to the front-end. This is where the involvement of the eIDAS component ends and the user is allowed to create their RE4DY account.

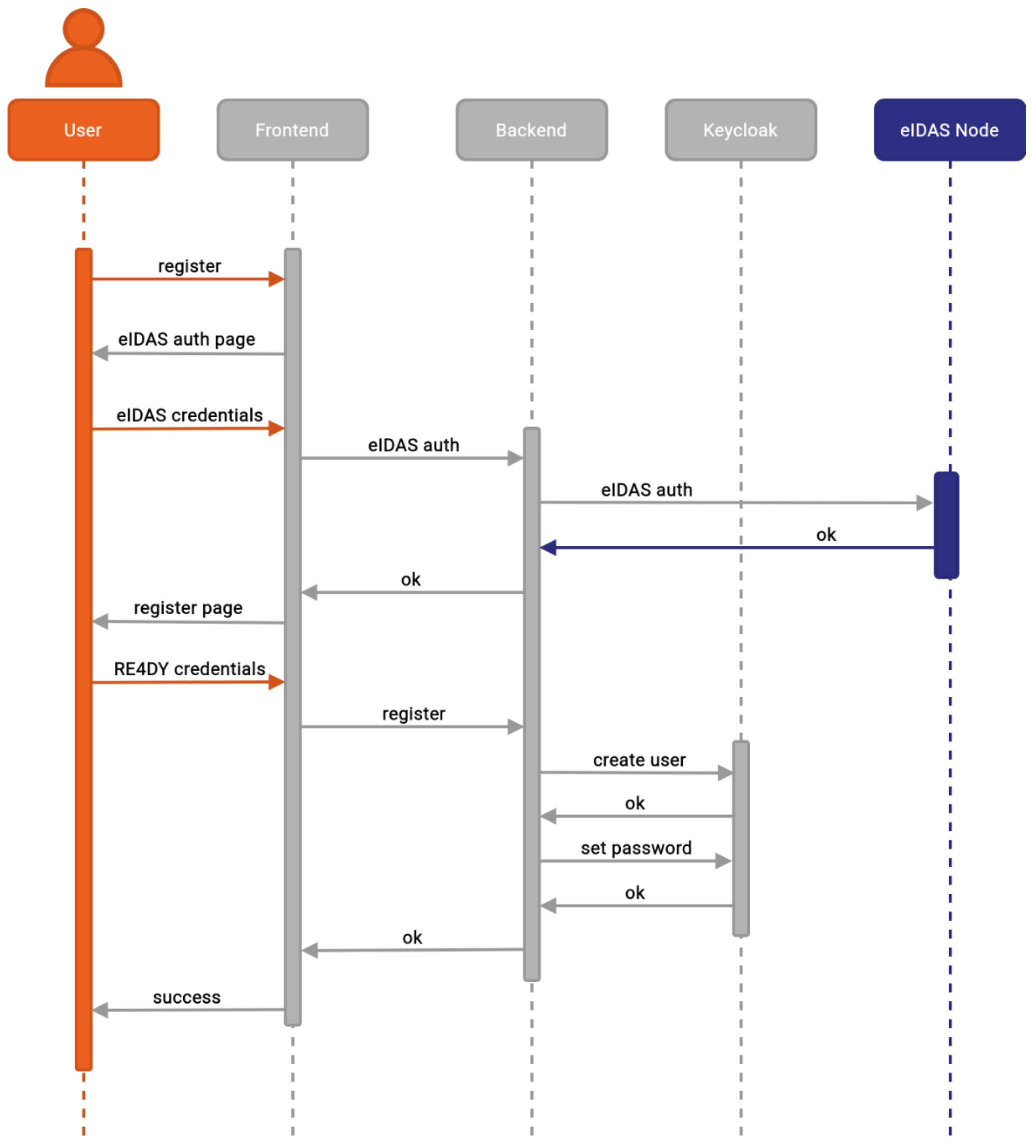

Figure 2 UML Diagram of the registration process and the communication of the front-end, authentication back-end, Keycloak and eIDAS services.

5. Internal Architecture

For the facilitation of various operations of the eIDAS node, an eIDAS API has been implemented. This is because the eIDAS node does not use a database which means internal files need to be directly modified for its configuration as well as the registration of new eIDAS users. The eIDAS API may be used by the eIDAS Management Web UI for the creation of users or by the authentication back-end during the first stage of the RE4DY registration, which involves authenticating with eIDAS credentials.

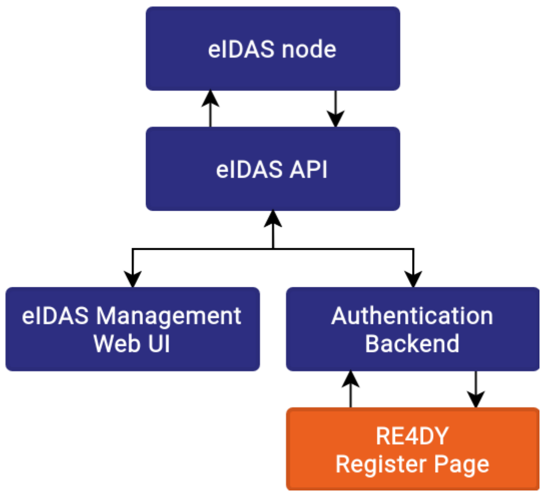

Figure 1 Internal architecture of the eIDAS component and related services

6. API

```
POST {{eidas-api}}/addUser
```

- **Input**:

```
{
    "username": "johndoe",
    "password": "12345678",
    "name": "John",
    "surname": "Doe",
    "identifier": "ZX1234567",
    "birthday": "1986-05-06",
    "placeOfBirth": "Greece,+Athens",
    "extraName": "someCustomAttribute",
    "extra": "someCustomAttributeValue",
    "gender": "Male",
    "phone": "1234567891"
}
```

- **Output (HTML):**
```
<!-- on success -->
<div>User inserted Successfully</div>
<div class="message">User inserted Successfully</div>
...

<!-- on duplicate username or identifier -->
<div class="error">User Already Exists</div>
```
- **Description**:
This endpoint is implemented by the API developed to facilitate the user creation process in the eIDAS node. The username and identifier fields are unique per user, meaning that if another user exists with the same username or identifier, the user creation will fail.

• `POST {{auth-backend}}/auth/authenticate`

- **Input**:
```
{
    "username": "newuser",
    "password": "1234",
}
```
- **Output**:
```
{
    "eidas_response": "eyJ0e...",
}
```
- **Description**:
This endpoint is implemented by our Spring Boot authentication back-end. It interfaces with the eIDAS node to determine whether the provided credentials are associated with an existing eIDAS user. On success, the returned eidas_response may be decoded to retrieve the assertion given by the eIDAS node as well as the attributes which correspond to the user.

The API documentation is also available in the Annex as a Swagger editor .yaml file.

7. Implementation Technology

The eIDAS artifact is officially distributed as a zipped archive which includes both binaries and source code required to execute the node. It is written in Java and utilizes the Tomcat web server for this purpose. We have created a Docker image which installs the required dependencies and initializes the eIDAS node, the main goal being to simulate the functionality of the eIDAS Node of a European Member State.

Furthermore, we have developed a small Python API to facilitate the management and configuration of the node and the creation of users and roles, since these procedures require the modification and replacement of settings across several server configuration files. We have also developed a minimal web page using pure HTML, CSS and JavaScript to additionally provide a graphical interface through which these procedures may be performed.

For the authentication, authorization and user management throughout the RE4DY project, we have implemented a Spring Boot back-end, written in Java. The back-end communicates both with the eIDAS node and with Keycloak, our Identity Manager of choice. Finally, to demonstrate the eIDAS-related functionality, a proof-of-concept user registration and login web UI has been developed using the React framework.

8. Comments

None.