ALAP - AgiLe Authentication Provider

Description of the Agile Authentication Provider (ALAP)

Version 0.1, 23.11.2015

Andreas Fitzek - andreas.fitzek@egiz.gv.at

Summary: This document describes the architecture and the interfaces of the AgiLe Authentication Provider ALAP.
# Table of Contents

1 Motivation .......................................................................................................................... 3

2 Architecture .......................................................................................................................... 3
   2.1 Overview ......................................................................................................................... 4
   2.2 Frontend Authentication Service .................................................................................... 4
   2.3 Backend Services ........................................................................................................... 5
      2.3.1 Session Service ........................................................................................................ 5
      2.3.2 Logging Service ....................................................................................................... 6
      2.3.3 Repository Service ................................................................................................ 6
      2.3.4 Policy Service ......................................................................................................... 6
      2.3.5 Cryptographic Service ............................................................................................ 6
   2.4 Frontend Application Services ....................................................................................... 6

3 Authentication Plugins .......................................................................................................... 7
   3.1 Username Password Plugin ............................................................................................ 7
      3.1.1 Registration .............................................................................................................. 7
      3.1.2 Authentication ........................................................................................................ 7
   3.2 SMS TAN Plugin ............................................................................................................ 8
      3.2.1 Registration .............................................................................................................. 8
      3.2.2 Authentication ........................................................................................................ 8
   3.3 Timebased One Time Password Plugin ......................................................................... 9
      3.3.1 Registration .............................................................................................................. 9
      3.3.2 Authentication ....................................................................................................... 10
   3.4 TAN Application Plugin ............................................................................................... 11
      3.4.1 Registration ............................................................................................................. 11
      3.4.2 Authentication ...................................................................................................... 12
1 Motivation

The Commission Implementing Regulation (EU) 2015/1502 states in 2.2.1 „Electronic identification means characteristics and design“ that for Assurance level „Substantial“ and „High“ one element needed is: „The electronic identification means utilizes at least two authentication factors from different categories.“

Authentication factors need to be easy to use for users to be accepted, as well as provide best possible security. Usually these authentication factors are integrated into the systems, which authenticate the users.

An implementation of an authentication factor might become vulnerable in a short amount of time, because a new security flaw was found. Therefore a deep integration in authentication systems might cause a long response time to insecure „factor implementations. Authentication factors therefore could be used externally, and integrated in a pluggable way.

A common approach is to users the modern smartphones, to run authenticator software, and act as an authentication factor. Because of the variety of modern smartphones, there is also a big variety in available technology to the users. We want to ensure the best possible security constraints, for as many devices as possible. Multiple different authentication factors should be provided, and the user should be guided, to the best possible authentication factor, that works on the user’s devices. This approach allows the use of new security technologies, as soon as these are available. Even if not all users, are able to use these.

To integrate the authentication factors into the system ALAP uses multiple Identity Providers to increase the security of an authentication. We implemented lightweight Identity Providers, which each perform one authentication step. These different Identity Providers are therefore our authentication factors. An authentication factor could therefore easily be excluded or added to the system, because it is reduced to a configuration option.
2 Architecture

2.1 Overview

ALAP is a distributed, highly scalable system, which provides authentication of end-users according to a definable policy and based on multiple authentication plugins. The authentication plugins are external web applications, which provide different authentication factors. The system is extendable, by registering new plugins.

An authentication plugin is a web application, which implements an OpenID Provider (OP).

ALAP is a multi-tier application. It consists of multiple frontend and backend services, which are connected through an AMQP based message bus. Image 1: Architectural Overview shows the architectural overview of the ALAP system.

![Image 1: Architectural Overview](image)

2.2 Frontend Authentication Service

The frontend authentication service is the main entry point to the ALAP. It provides a user interface for users to login and to manage their account. It also acts as Relying Party (RP), when talking with the different authentication plugins. It performs the necessary steps to authenticate an end-user according to the defined security policy.
The frontend authentication service also offers a REST-based API to the authentication plugins, to access plugin and user specific configurations, as well as plugins specific identity lookups.

For the user to configure his authentication plugins, the frontend authentication service provides a list of links, to the configuration pages of authentication plugins. The authentication plugins can use the REST-based API to perform the configuration with the user. The actual configuration data used by the authentication plugins can be stored in the ALAP system.

![Image 2: Frontend authentication service account management](image)

### 2.3 Backend Services

The backend services provide common functionality to the frontend application services and use the frontend authentication service to perform the needed authentication steps.

#### 2.3.1 Session Service

The backend session service, allows the creation and destruction of system sessions and transactions. Sessions and transactions can have arbitrary attributes. The session service can be used by all services connected to the AMQP broker.
2.3.2 Logging Service
The backend logging service provides an event log, to log information, which helps investigations, in the case of an incident. The service acts as a black hole sink, therefore only functionality for writing and reading to the logging service is available, but no functionality to delete log entries.

2.3.3 Repository Service
The repository service is the data access layer of the application. It stores the available application domains, the domain policies, authentication plugins, plugin configurations, and user information, as well as user and plugin specific configuration data.

2.3.4 Policy Service
The policy service enforces the configured domain policies. Based on the requested action and the requested domain the policy service checks if the requirements for the execution of the action in the domain are met. The policy is a JavaScript program, which is executed. A provided function checks if a specific authentication plugin was already successfully authenticated in the current transaction or. This check verifies the ID Token delivered by the authentication plugin as OpenID Provider to the frontend authentication service, against the plugin configurations from the repository service. If the policy conditions are met, the request is forwarded to the specific target system, or a success response is generated. If the conditions are not met, the policy should produce authentication paths. These authentication paths describe the possible authentication path through the available authentication plugins the user can perform to authenticate this request. These authentication paths are then used to instruct the frontend authentication service to perform the actual authentication.

2.3.5 Cryptographic Service
The cryptographic service can only be accessed through the policy system. It allows to read a user certificate and to create a digital signature based on the user’s private key. The cryptographic service should use a Hardware Security Module for these operations.

2.4 Frontend Application Services
Multiple frontend application services can be implemented that require authenticated tasks. These services provide the actual business service. An example of such a frontend application service is an implementation of the security layer. But also an Identity Provider based on well-known protocols like SAML2 or OpenID Connect could be implemented. The frontend application services use the backend policy engine to authenticate sessions and transaction for certain operations, like signature creation or authorization to access resources.
3 Authentication Plugins

3.1 Username Password Plugin

3.1.1 Registration
Via the frontend authentication service the user can access the configuration page of the Username Password Plugin (UP). On this page the user can once choose his username and always change his password.

![Image 3: Configuration of Username and Password](image3.png)

3.1.2 Authentication
The Username Password Plugin (UP) is able to identify the end-user. The user enters his username and his password. The plugin uses the frontend authentication service to lookup the identity of the given username, and to fetch the configuration for the identified user. The configuration contains the hashed password of the user. This is checked against the provided password. If those match, then the user is proofed, that he knows the password associated with the given username. Image 4 shows the authentication form used to provide the username and password.
3.2 SMS TAN Plugin

3.2.1 Registration
Via the frontend authentication service the user can access the configuration page of the SMS TAN Plugin. On this page the user can setup his mobile phone number to receive SMS codes.

3.2.2 Authentication
One authentication plugin needs proof that the user possesses the registered SIM card. To do this, the SMS TAN Plugin sends a TAN code to the registered mobile phone number of the requested user. This TAN code has to be entered in the authentication
form by the user. If the TAN code entered through the web form matches the TAN sent to the mobile phone, then the user proved possession of the SIM card, because he was able to receive the SMS. Image 6 shows the authentication form for the SMS TAN plugin.

3.3 Time-based One Time Password Plugin

A Time-based One Time Password (TOTP) is an HMAC based alphanumeric code as defined in RFC6238. Basically an authenticator application and a server share the same secret information and based on the current timestamp and this secret information a PIN code is generated. Everybody with knowledge of this secret information is able to calculate the correct code.

3.3.1 Registration

Via the frontend authentication service the user can access the configuration page of the Time-based One Time Password Plugin. On this page the user can pair an authenticator application like the EGIZ Authentication or the Google Authenticator with the plugin. Therefore the user scans the QR code with the authenticator application and enters the generated code, to validate the pairing process.
3.3.2 Authentication

This plugin shares a secret with an authenticator application of the user and requires the correct code to be entered.
3.4 TAN Application Plugin

The TAN Application plugin shares a secret AES key and initialization vector with the server. The initialization vector is changed for each authorization process and requires the knowledge of the previous initialization vector and the AES key. This authentication can only be performed when the EGIZ Authenticator is used.

3.4.1 Registration

Via the frontend authentication service the user can access the configuration page of the TAN Application Plugin. On this page the user can pair an authenticator application like the EGIZ Authentication with the plugin. Therefore the user scans the QR code with the authenticator application and accepts the pairing request in the authenticator application. The authenticator application connects to the authentication plugin and performs the initial pairing protocol. Once this is done the
user is automatically redirected to the success page of the configuration.

3.4.2 Authentication

To perform an authentication the user has to scan the presented QR code with the paired authenticator application. In the authentication application the user has to accept the authentication request. Then the authenticator application performs the authorization protocol, based on the secret AES key and the current initialization vector with the authentication plugin. This plugin does not require the user to enter any information into the web form. The user only has to scan the QR code.
Signaturdaten

QR-Code scannen

In der Authenticator App den QR-Code scannen.

Verwenden Sie die Kamera Ihres mobilen Gerätes, um diesen QR-code zu scannen.

Vergleichswert: UIIT2Osqay0O7

Abbrechen

Image 10: Authorization form for the TAN Application
Document history

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Author(s)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>23.11.2015</td>
<td>Andreas Fitzek</td>
<td>Initial Version</td>
</tr>
</tbody>
</table>