ADUCID Architecture

Version 3.1.0.RC3

Release date June 17, 2016

Antala Staška 2027/79, 140 00 Prague 4 | T +420 271 100 100 | F +420 271 100 101
Vídeňská 125, 619 00 Brno | T +420 547 100 100 | F +420 547 100 101
Sady 5. května 59, 301 00 Pilsen | T +420 271 100 100 | F +420 271 100 101
Teslova 30, 821 02 Bratislava | T +421 (2) 4821 3111 | F +421 (2) 4821 3199

ISO 9001 ISO 14001 ISO/IEC 27001 ISO/IEC 20000
CZ > Corp. ID 25313029 | Registration Court RC Brno, File no. B2113
SK > Corp. ID 35787546 | Commercial Register DC Bratislava, Section Sa file 2431/B

Table of Contents

Contents

1. Purpose of this document

This document describes ADUCID® at the system level necessary for integrating with target application.

To understand this document, the reader is required to have knowledge of web technologies, programming and integration of web applications.

2. Basic components

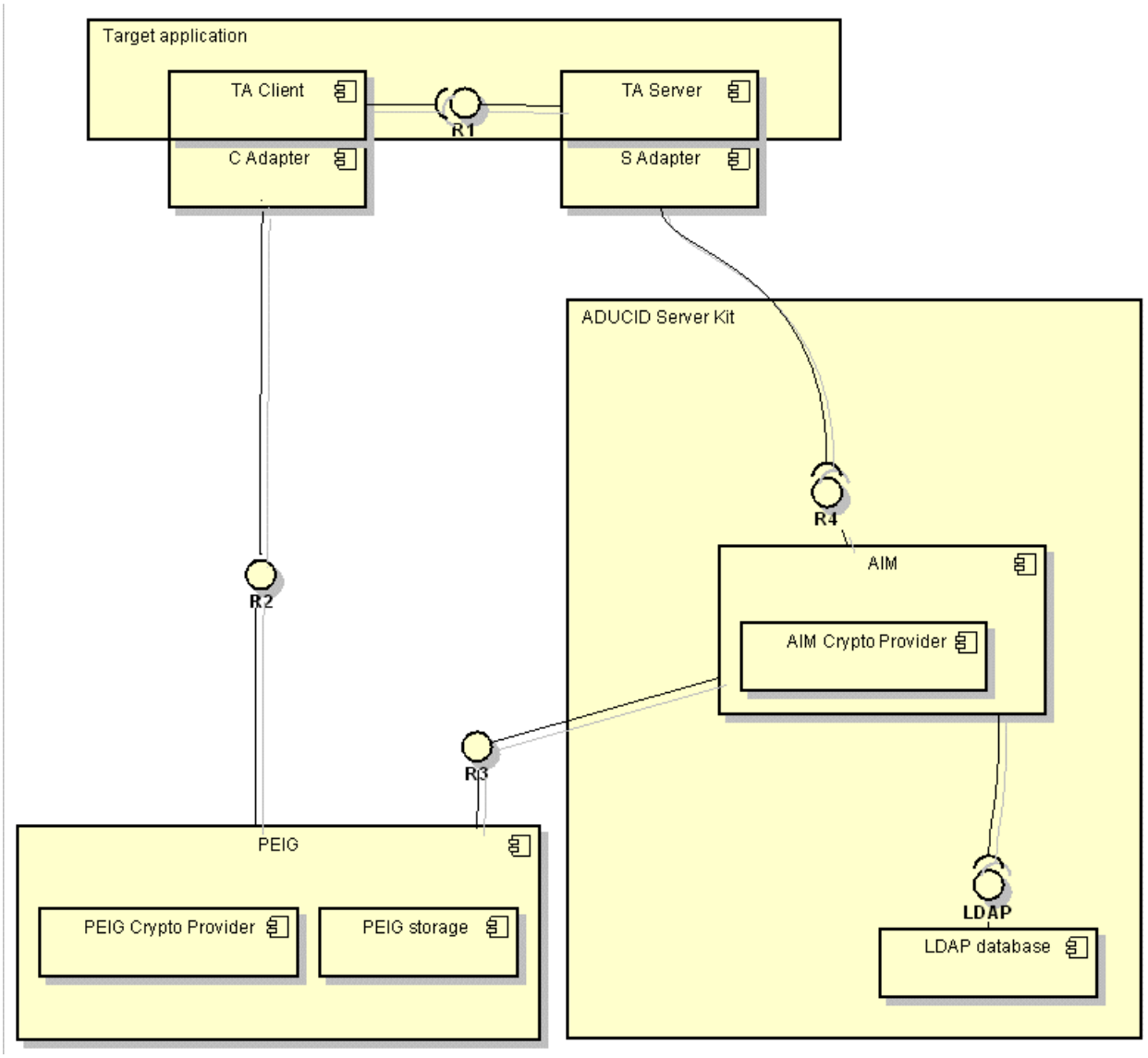

Figure 2-1 ADUCID® system diagram for integration

2.1. Target application

Target application is any application that uses ADUCID® services. Examples of such applications include web applications with standard thin client (standard browser), mobile applications with server counterpart, classic client-server applications, or even universal system tools which are not normally viewed as applications, e.g. a VPN system for remote access, Wi-Fi connection, terminal access (e.g. X terminal), etc.

From the system point of view, the application consists of a client part and a server part (TA Client and TA Server), which communicate with each other in their own manner via an R1 interface.

2.2. Server part of ADUCID®

The entire server part of ADUCID® along with the operating system and all third-party systems required to operate the server part of ADUCID® are supplied as complete virtual appliances.

The server part of ADUCID® consists of following parts:

2.2.1. AIM

ADUCID® Identity Machine – delivers ADUCID® server functionality, performs all ADUCID® operations and provides access to user data stored along with electronic identities in the LDAP database.

AIM is controlled by the target application using the R4 interface. Using this interface, it also provides services for working with user data.

Using the R3 interface, it communicates with the client part of ADUCID®.

Another part of AIM is the provider of cryptographic services (AIM Crypto Provider) that can be implemented through different manners - e.g. as a software library or hardware device (HSM, etc.).

2.2.2. LDAP

Data for electronic identities and user data along with other operational data is stored in the standard LDAP database.

2.2.3. SQL database

HyperSQL database is used to store ADUCID events and licensing logs.

2.2.4. Admin applications

ADUCID comes with a set of support applications. All admin applications require a particular role, proofing and personal factor (first admin gets these automatically).

2.2.4.1. UserAdmin

UserAdmin is a user and PEIG management tool. It also shows statistics and licensing.

2.2.4.2. SecAdmin

SecAdmin is meant to configure security parameters of ADUCID AIM (encryption algorithms, key length, expiration periods etc.)

2.2.4.3. RoleAdmin

Allows administrators to assign ADUCID roles. Without a role, an ADUCID user has only very limited rights. There are special groups to configure / administrate AIM features:

- useradmin – can manage users and their PEIGs

- operator – can manage user PEIGs only
- roleadmin – can assign user roles
- secadmin – can alter AIM security
- regadmin – can proof users

2.2.5. Proofing applications

See Identity proofing

2.2.6. Profile application

A simple application that demonstrates user log on and PEIG (replica) self-management.

2.3. Client part of ADUCID®

PEIG® is the fundamental client element of ADUCID® that fully manages electronic identities of its user.

PEIG documentation is located at http://www.aducid.com/support

Currently PEIG is supported on Windows 7, Windows 8, Windows 10, Mac OS X, Android (4.0+) and iOS 7+ devices

Windows and OSX machines also support PEIG located on a USB token.

2.3.1. Windows PEIG PC, OSX PEIG

Windows/ OSX PEIG runs on the same computer as user's browser. Browser processes AIM-proxy script which calls PEIG Redirect Adapter module on local host port.

2.3.1.1. PEIG USB

Windows / OSX PEIG has option to store identity files on a USB. This feature is accessible for every PEIG PC / OSX installation inserting USB disk and preparing it for ADUCID.

2.3.2. iOS PEIG and Android PEIG

PEIG can run on mobile phones or tablets with Android 4.0+ or iOS 8+. PEIG can be called in used in two ways:

- QR code scanning using phone / tablet camera
- calling aducid:// URI schema (on Android Intent is preferred)

Users can authenticate into desktop application scanning a QR code using mobile PEIG (or QR code displayed by another PEIG).

Native applications are also supported. They can call PEIG directly using schema / intent or use PEIG API (Papi) – See Integration with mobile applications

2.3.3. Personal factor (PF)

To proof his/ her identity ADUCID provides a feature called personal factor. PF something only user know or has.

Secret

The most common PF is some kind of secret. User can define a picture sequence or number sequence.

NFC

On Android phone a user can use a NFC tag as his / her secret.

Apple Touch ID

On iOS devices with fingerprint sensor (iPhone 5S or better) user can activate fingerprint factor as addition to his / her secret.

2.3.4. Replicas

With ADUCID user is not limited to just one device. On the contrary he / she can have “unlimited” number of devices called replicas. Replicas are PEIG with different cyber id referring to the same person.

To create a replica, the user needs another PEIG. Then he / she starts replica process (from application). Using a device with a camera it’s very simple – user just scans QR code display on one device using another.

2.4. PEIG proxy QR code

PEIG can act as super secure feature for QR code authentication. In this case PEIG authentication is turned off (but PEIG is running). QR code is not server by AIM-proxy but rather created by PEIG-proxy module. This is one of most secure setups in ADUCID topology (see Binding documentation for details) and definitely secure than displaying QR code using browser.

2.5. Interfaces

ADUCID uses 4 basic interfaces:

2.5.1. R1

R1 is an application interface handled by customer application itself. In mobile application integration R1 is encapsulated in Papi (PEIG API).

2.5.2. R2

R2 is an interface between client application and PEIG. This communication can be handled via:

- Uri scheme on mobile phones scheme **aducid://**
- Redirect adapter (from browser to Windows / OSX PEIG) is local port 44240
- Scanning a QR code
- Integrated using Papi.

2.5.3. R3

R3 is an internal interface between PEIG and AIM which uses http transport and SOAP protocol.

2.5.4. R4

R4 is interface between server application and AIM. Like R3 it uses http (or https) transport and SOAP protocol.

R4 is “a low level” layer. It is encapsulated in ADUCID WEB SDK or ADUCID JAVA SDK for simplified integration.

2.6. Communication between components

A fundamental unit of activity in ADUCID® is an operation. Such operation can be an authentication, creation (initialization) of an electronic identity, a change in electronic identity, etc. Operations can have their own parameters.

The server part of the target application first requests the required operation via the R4 control interface and specifies its AIM parameters.

The result is a unique, one-time authId operation identifier that can be initiated either by the target application or AIM.

The client part of the target application transmits the PEIG® startup event through the R2 interface. The startup event includes authId and the R3 interface URL.

Communication between the client part and the server part of the target application via the R1 interface is handled by the target application.

The startup event from the R2 interface is sent to PEIG®. PEIG® then starts processing the operation by transmitting the authId to R3 URL. AIM manages the entire operation process that consists of transmitting and processing several messages between PEIG® and AIM.

In terms of communication, PEIG® is a client and an AIM a server. In terms of control, the operation is managed by the AIM (based on target application request submission).

When the operation is concluded, a random, one-time secret authKey is generated on PEIG® (if successful), which is then transmitted to the client part of the target application along with authId via R2.

The server part uses authId and authKey for further communication with AIM via the R4 interface in order to obtain electronic identity attributes and to work with user data (personal objects). In order for these requests to be carried out successfully, correct values for authId and authKey (one-time secret that was transmitted at the end of a successful operation at the client part of the target application) must be transmitted.

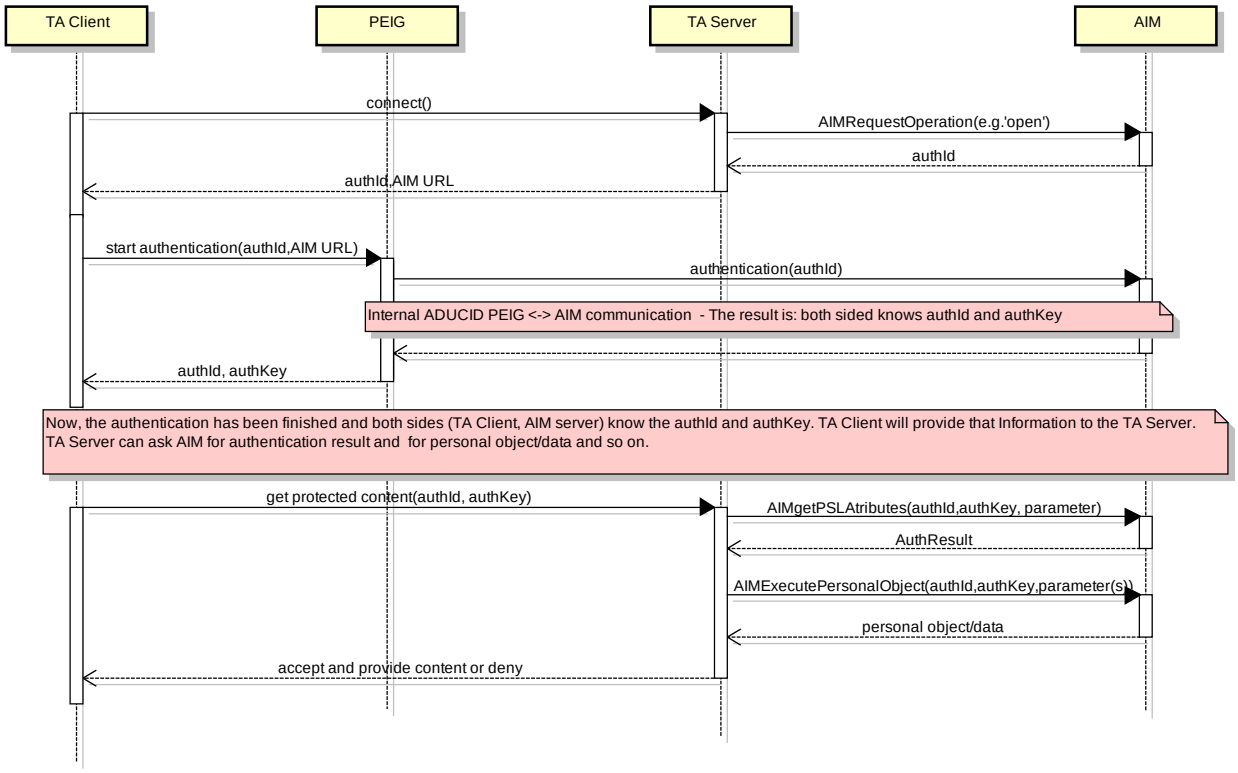

Figure 2-2 ADUCID® communications diagram

PEIG finishes its activity by using a return URI. The return URI is transferred from AIM to PEIG during PEIG activity. The return URI is typically https://, to activate a registered web browser.

3. Integration with a target application

3.1. Server side integration

ADUCID supports many ways how to integration with target application:

3.1.1. Web service ADUCID - “WSA”

WSA is meant for all platforms **except** Java (Java is also possible but we provide a full module so WSDL import is redundant).

Web service is a standalone “application” which must be deployed on same server as AIM. Its default address is like this:

```
https://[hostname]/wsa
```

There you will see WSA Interface homepage where you can download WSDL file to generate client.

```
https://[hostname]/wsa?wsdl
```

Import WSDL into your project, e.g. Visual Studio.

Then you create instance of AducidApiServiceClient and make calls to ADUCID and get authentication result with a few rows of code:

Simplified authentication request with personal factor verification written in C#:

I. Starting page

This page starts authentication request. As a parameter of verifyLF method we provide URL with final page where authentication is evaluated:

```
protected void Page_Load(object sender, EventArgs e) {

   var aducid = new AducidApiServiceClient();

   string host = "http://" + HttpContext.Current.Request.Url.Host + ":" + HttpContext.Current.Request.Url.Port;

   var redirect = aducid.verifyLF(host + "/result");

   Response.Redirect(redirect);

    }
```

II. Result page

On the result page we evaluate the authentication request using aducid.getResult

There are two basic results – authentication is OK or an error is thrown:

```
protected void Page_Load(object sender, EventArgs e) {

   string queryString = HttpContext.Current.Request.Url.Query;

   var query = HttpUtility.ParseQueryString(queryString);

   var sb = new StringBuilder();

   var aducid = new AducidApiServiceClient();
```

```
   try {

      var result = aducid.getResult(query.Get("authId"), query.Get("authkey"));

      sb.Append("<table class=\"table table-bordered\">");

      sb.Append("<tr><td>authId:</td><td>" + result.authId + "</td></tr>");

      sb.Append("<tr><td>authKey:</td><td>" + result.authKey + "</td></tr>");

      sb.Append("<tr><td>operationName:</td><td>" + result.operationName +
"</td></tr>");

      sb.Append("<tr><td>userDatabaseIndex:</td><td>" + result.userDatabaseIndex +
"</td></tr>");

      sb.Append("<tr><td>peigReturnName:</td><td>" + result.peigReturnName +
"</td></tr>");

      sb.Append("</table>");

   }

   catch (Exception ee)

      { sb.Append("Error:" + ee.Message);}

   l1.Text = sb.ToString();

}
```

3.1.2. Java SDK Web Platform

Tomcat only

The simplest way is to integrate a Java application with ADUCID. It is based on Java SDK Client API. It implements all necessary calls directly to your application returning authentication (operation) result into your session variables. So you get a web controller with set of almost all operations ready to use.

3.1.3. Java SDK Client API

Java only

For Java use this Client API instead of Web service WSA

See *ADUCID SDK JAVA*

3.1.4. R4 Client

R4 clients is the lowest level of ADUCID framework. All other server APIs are based on it. We don't recommend it for common integration as it requires immense knowledge of ADUCID architecture. All tasks can be done using previous methods.

3.1.5. Advanced integration

For advanced concepts of AUDICD integration *see "ADUCID Integration Manual Advanced"*

3.1.6. Adapters

3.1.6.1. Tomcat Adapter

Tomcat adapter defines new Tomcat server authentication method named “ADUCID”. It extends standard Tomcat class AuthenticatorBase. Application can use the adapter in context definition as a valve to manage application authentication/authorization. See ADUCID Tomcat Adapter documentation for details.

3.1.6.2. Spring Security Adapter

Spring Security adapter extends standard Spring techniques to authenticate/authorize user. Implementation is based on AuthenticationProvider interface implementation. Adapter can be used in applications based on Spring technology to manage application access rights. See ADUCID Spring Security Adapter documentation for details.

3.2. Integration with web browsers

Integration of web applications supports three different types of communication between the client (web browser) and the server: redirect. URI and QR code.

3.2.1. Redirect adapter

Redirect adapter handles the communication between the client and the server adapter via standard HTTP redirect functionality. No adjustments or plug-ins for standard browsers are required. Client redirect adapter is part of Windows / OSX PEIG and expects an HTTP redirect to the ADUCID® port (**44240**) that initiates authentication.

PEIG with redirect adapter cannot be used on a multiuser system (e.g. Terminal Server).

3.2.2. URI scheme

The current operating system (including mobile phone operating systems and workstation operating systems) brings a new way of secure communication between applications in a protected user space. It is the registered, specific URI schema.

If the supporting application is registered in the operation system with the URI schema during installation, the operating system ensures activation of the registered application and transfer of the URI parameters to the application.

This option is used by PEIG for mobile phones.

3.2.3. QR code

If there is not redirect nor URI scheme managed by PEIG, ADUCID QR code is displayed. This code can be scanned using a mobile PEIG to process the request.

3.3. Integration with mobile applications

The simplest way how to integrate ADUCID to a mobile application is Papi - PEIG API described in *ADUCID Mobile Application Integration manual.*

4. Identity proofing

ADUCID provides authentication. It guarantees that PEIG is identified every single time by AIM and also PEIG always recognized particular AIM.

Target application which uses ADUCID authentication needs some key to bind authentication to its own user database. It might use AUDCID UID (User database index), e-mail or any other attribute.

As result a particular PEIG (or more PEIGs in replica) are bound to target application. So that application “knows” which user requests an operation.

This is only technical perspective of authentication. The most important issue for service provider is to know if that PEIG is owned by “right” person – or “proofed” person. This is accomplished by process called Identity proofing.

4.1. Proofing scenarios

Thera are several ways how to proof a user. He/she can get to an office, show ID and get proofed. Or he/ she can be visited by person how can verify his / her identity. Or user can fill in a form, send it with a copy of her / his ID.

To proof someone’s identity there has to be some administrator with right to verify and approve users. This administrator has to have role called “registrator” and has to be proofed and verified using personal factor.

ADUCID demonstrates and supports these basic scenarios:

4.1.1. Activation code

User goes to an office and meets an administrator. Administrator fills in user details, verifies his / her ID. As result he gives / sends him an activation code.

Using this code user can finish the proofing process by providing it to proofing application.

4.1.2. Registration form

In this scenario user fills in a form and sends it to registration point (scan of ID might be required). Then he/ she goes to the office, administrator verifies this form and approves the user.

4.1.3. QR proofing - admin fills form, user scans

As in first scenario a uses comes in an office and meets an administrator. But no activation code is created / sent. Instead user scans a QR code displayed on administrator’s PEIG.

4.1.4. QR proofing - user fills form, admin scans

Scenario where user can be at home and administration visits him / her. User has a form prepared, administrator checks it and then scans a QR code displayed on user’s PEIG.

4.1.5. Identity link proofing

If one AIM contains proofed identities, it can act as identity provider for other AIMs (this scenario must be enabled and supported by both sides).

4.2. Proofing level

ADUCID recognizes two proofing levels – with or without personal factor. AIM either support personal factor proofing or not (this decision should be done when AIM is installed).

All proofing methods are bound to this setting

4.3. ADUCID proofing support

Proofing is supported by ADUCID server methods and adapters. Application developer can use these methods / adapters to approve (proof) user and evaluate proofing status e.g. you can allow login only for proofed users who successfully provide their personal factor.

SDK methods also support proofing form and proofing code.

All identity proofing scenarios are demonstrated in proofing applications. These applications can be installed with ADUCID Server Kit as an option.

ADUCID UserAdmin application shows current proofing status of a particular user.

See Javadoc and Tomcat adapter documentation for details.

5. Advanced concepts

5.1. ADUCID standard operations

The fundamental element of ADUCID®'s activity is an operation. The target application requests AIM to perform an operation, AIM along with PEIG® then perform the operation and make the result available to the application. The application can then use the result of the operation (e.g. use a positive authentication result to grant access to information to a specific user in the scope of that user's assigned rights, or use a negative result to deny access).

Standard applications only use the "open" operation, which performs user authentication.

Applications that manage identities (Identity Management) use other operations that support the execution of the entire lifecycle of the identity and other activities. For illustration, a list of supported operations of ADUCID® is provided with a brief description of each operation:

- Initialization of an identity (II – Identity Initialization – "init") - PEIG® and AIM together form a new unique electronic identity.
- Use of an identity (IU – Identity Use – "open") - PEIG® and AIM together validate the eID and provide a link to user information (authentication).
- Change in an identity (IC – Identity Change – "change") - PEIG® and AIM together change the existing internal values of the identity while preserving the entire context of personification (including all associated personal data).
- Termination of an identity (IE – Identity End – "delete") - PEIG® and AIM together invalidate the electronic identity and prevent anyone from performing any operation using this identity.
- Reparative change of an identity (RC – Reparative Identity Change – "rechange") - Change of an identity performed if the validity of previous identity has expired.
- Reparative initialization (RI – Reparative Identity Init – "reinit") – identical with II, performed if corresponding identity exists on PEIG® (this operation's purpose is to restore AIM).
- Confirmation of a link between identifiers (IL – Identity Link – "link") – PEIG® and two AIMs form a unique, one-time shared identity, and its connection to user information for both AIMs.
- Extended Use or XUSE is a n advanced operation to create replicas, display dialogs and work with personal factor.

5.2. Binding

The issue of authentication results from linking the target application together with the protection of the data channel between the client and server part of the target application. This is called "binding".

Different user scenarios exist for how to link a target application to ADUCID authentication. They have different user and security features. It is possible to take snapshot of a QR code by using a mobile phone, when the QR code is displayed on a workstation screen to log in, or it is possible to use PEIG from hard disk of the same workstation where the web browser is running, or it is possible to use a web browser on a mobile phone or tablet.

The AIM security manager can select what binding scenarios will be supported by AIM and what scenarios will be disabled. This is possible through the AIM "binding mode" attribute configuration.

See aducid-binding.docx for further details.

6. Documentation overview

6.1. Documents for PEIG users

User documentation is located at http://www.aducid.com/support

This URL opens from clients when they click on “HELP”.

6.2. Documents for service/application/identity providers

Overview of documents:

Document name	Contents
ADUCID ServerKit Installation Guide	Preparing the environment for ADUCID Server Kit installation: Deployment scenarios, network configuration Import to VMware infrastructure System configuration First start
ADUCID ServerKit Administration Guide	ADUCID Server Kit configuration and maintenance guide: Configuring AIM, UIM and related components High availability configuration Log configuration and maintenance Configuring monitoring systems Configuring attributes in LDAP for external applications Backup and restore

6.3. Documents for application integrators and developers

These documents require knowledge of web technologies, programming and integration of web applications.

Document name	Contents
ADUCID SDK JAVA	Description of ADUCID’s JAVA integration library: • SDK architecture • Example of integration using ADUCID SDK • Dependencies on third party libraries • Sample application - Hello World
ADUCID Mobile Application Integration	How to integrate a mobile application (iOS, Android) with ADUCID technology
ADUCID Integration Manual -Tomcat	Guide for implementing the ADUCID technology into the Tomcat web container
ADUCID Integration Manual – Spring Security	Guide for implementing the ADUCID technology with the Spring Security Framework

ADUCID Advanced Integration Manual	Description of how to integrate web applications with ADUCID technology: • Available methods of integration with ADUCID • Description of the AIM component's interface • Description of ADUCID authentication
ADUCID binding	Description of designed user scenarios how to link a target application to ADUCID authentication

7. Abbreviations

Below is a summary of used abbreviations and their meaning:

ADUCID®

ADUCID® (Automatic Distributed and User Centric Electronic Identity) is a new authentication system that functions on the principle of providing services and infrastructures of electronic identities. It is an identification and authentication framework based on new ideas, rules, procedures and implementations for work and support of a unified method of authentication.

The main purpose of ADUCID® is to provide identification and authentication services in the cybernetic world of ICT systems using the ADUCID® secure authentication layer.

ADUCID® provides:

- Electronic identity services
- Secure authentication services
- Essential infrastructure for listed services

PEIG®

PEIG® (Personal Electronic Identity Gadget) is a device that provides full management capabilities for its user's electronic identities. Using the user identity, it also provides automatic authentication between the client application (used by the user) and the server part of the target application (that the user is accessing).

AIM

ADUCID® Identity Machine - Implements ADUCID® server functionality itself. It performs all ADUCID® operations and provides access to user data stored along with electronic identities in the LDAP database.

Through a standard network interface (web services), it provides target applications with services related to administration of the CyberID/eID. AIM contains an administrator and a user graphic interface (called UIM). AIM can also provide authorization services (including administration of authorization attributes to the relevant CyberID/eID).

AIM-proxy

Specialized module for web applications used to communicate with the client web browser upon authentication of HTML applications. This component enables ADUCID® to login into UIM without modifying the browser (redirect login).

PF

Personal factor