

Towards Interoperability: An Architecture for Pan-European eID-based Authentication Services

Arne Tauber¹, Bernd Zwattendorfer¹, Thomas Zefferer¹, Yasmin Mazhari²,
Eleftherios Chamakiotis²

¹ E-Government Innovation Center ¹
{Arne.Tauber, Bernd.Zwattendorfer, Thomas.Zefferer}@egiz.gv.at
² Gov2u
{yasmin, leexam}@gov2u.org

Abstract. In the last years several EU Member States have rolled out smart-card based electronic ID (eID) solutions to their citizens. Not all of these solutions are directly compatible to each other. However, with respect to the i2010 e-Government initiative and the upcoming EU Services Directive, cross-border identification and authentication is now on the agenda of all EU Member States. In this paper we present a smart-card based eID identification and authentication solution, which supports smart-cards from different Member States. The proposed solution can be easily integrated into existing authentication and identity management solutions and does not necessarily require any additional client software to be installed by citizens.

Keywords: Authentication, Identification, Interoperability, Smart Card, eID, CAS.

1 Introduction

Enabled by the success story of the Internet, numerous aspects of everyday life have been shifted to or have been adapted for the World Wide Web. This has resulted in a continuously increasing number of services being offered online nowadays. National government agencies of various European countries have reacted to this trend and provide a considerable number of web based services for citizens as well. The electronic provision and improvement of administrative procedures is commonly known under the term "e-Government" and includes services for income taxes, personal documents, change of address, and many more.

One basic challenge of e-Government applications is the secure and reliable identification and authentication of citizens. As administrative procedures usually involve security- or privacy-sensitive data, it is crucial that this data is processed by, and disclosed to, authorized parties only. Therefore, government agencies have to be

¹ EGIZ is a joint initiative of the Austrian Federal Chancellery and the Graz University of Technology.

aware that a citizen involved in an online administrative procedure is really the person they claim to be. When carrying out such procedures conventionally face to face in an office, citizens can easily be authenticated by verification of a presented ID document, e.g. a passport or identity card. In e-Government applications, where citizens usually interact with an online application remotely over the Internet, the reliable and secure authentication of citizens is no trivial task.

The widely used username/password based approaches provide weak authentication only. A number of vulnerabilities are known for these authentication mechanisms [1]. Hence, several EU Member States rely on smart-card technology to identify and authenticate citizens in e-Government processes. Most of these Member States have already rolled out smart-card based electronic IDs (eIDs) to their citizens and several others are in the preparation stage of a country-wide roll-out. An overview of current smart-card based eID solutions of different EU Member States has been provided by Siddhartha Arora [2].

In the last years, many smart-card aware eID applications and solutions have been put in place in several EU Member States on a national level. Unfortunately, most of these solutions are only applicable in a purely domestic scenario, for which the respective solution has been designed. Interoperability between different Member State specific solutions is usually not supported. This is a major drawback since an increasing number of administrative procedures has to be carried out on a pan-European level, especially in the context of the European Services Directive [4]. Hence, the cross-border interoperability and the mutual recognition of national eIDs are of great importance for the next steps towards a European Administrative Space (EAS). The importance of this topic is emphasized by the efforts and focus put by the European Commission into the accomplishment of the European STORK project [21], which aims to establish a pan-European eID interoperability platform.

In this paper we address this issue by presenting a smart-card based authentication and identification solution that supports eIDs of different EU Member States. Any online application that relies on the proposed solution is able to authenticate users from different Member States. Our approach relies on qualified electronic signatures created with the citizen's national eID card ensuring cross-border authentication backed by the mutual recognition of qualified certificates.

In the last years, Austrian e-Government initiatives have created a set of tools for online identification and authentication of citizens using the Austrian citizen card, the national eID. In the remainder of this paper we present a solution based on an adapted version of these tools being able to authenticate citizens of other EU Member States. So far, several different models of identity systems have been established. We discuss these models in Section 2 and show which model fits our solution. In Section 3 we briefly introduce a popular example for each of these models to outline benefits and shortcomings. We discuss the system architecture of our eID identification and authentication approach in detail in Section 4. In order to demonstrate the applicability of our solution in practice, we implemented an authentication plug-in for a popular and widely-used authentication management system. We briefly present this case study in Section 5. Thereafter, we give some remarks on our solution in Section 6. Finally, conclusions are drawn.

2 Models of Identity Systems

Identification and authentication of digital identities are not new problems. Hence, various approaches for identity systems and identity management systems already exist. However, not all identity systems follow the same methodological approach. For instance, some solutions focus more on central storage of identification data; in contrast others rely on federated data repositories. To get a better understanding of how our solution can be classified, we briefly present theoretical models of identity systems in this section. Palfrey and Gasser [5] distinguish between three major types of identity models which are outlined below in a nutshell. The distinctive criterion, in fact, is who has control over the identification data.

2.1 User-Centric Model

Figure 1 illustrates the basic setup of the user-centric model. A user attempts to access a protected resource or application of a service provider. Usually, the service provider initiates an authentication process by forwarding or redirecting the user to an identity provider. An identity provider is responsible for managing and issuing identity information to service providers. Hence, the identity provider handles the identification and authentication process with the user and transfers the identification data back to the service provider. Identification data depicts all user related information like unique identifiers or other user attributes such as name or date of birth.

In this model, the user always carries the responsibility for releasing identification data to the identity provider and service provider respectively. Personal information is only transferred to a requesting service provider if the user explicitly gives his consent to do so. At any time, the user remains the owner of the data and not the identity provider.

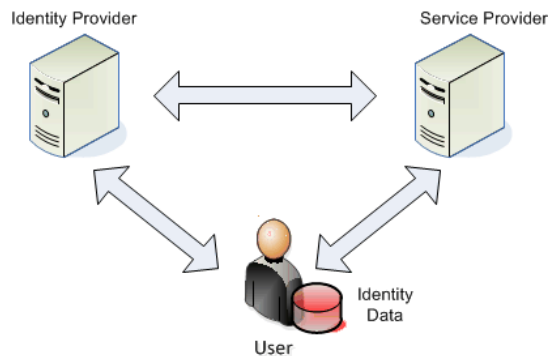


Fig. 1. User-Centric Model

2.2 Centrally-Controlled Model

This type of model is the dominant approach currently used in the Internet. Usually - if users want to access a certain service – they are requested to provide appropriate identity information for registration. This identity information is then stored centrally in data repositories of the identity provider. When requesting a protected application of a service provider, the user has to first authenticate with the identity provider, who then forwards the appropriate authentication information to the service provider.

Regarding privacy, the user is not in control anymore, e.g. of which data is kept in the identity provider's database or which information is transferred to the service. Figure 2 shows such a centrally-controlled model.

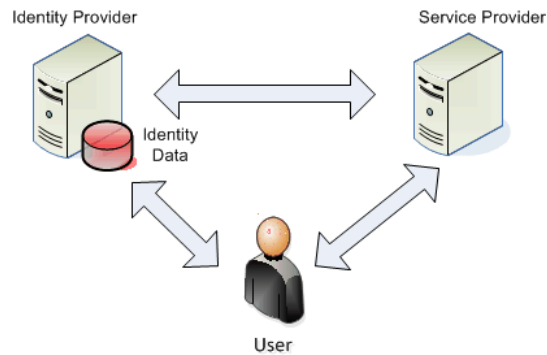


Fig. 2. Centrally-Controlled Model

2.3 Federated Model

In this model, user data is distributed over various identity providers. In contrast to the centrally-controlled model, identity information is not stored in a central location. However, the identification data can be easily shared between identity providers via linkage of the data repositories. Thus, the identity data of a user is stored in a federated way. Usually, identity federation is achieved by special trust relationships between identity providers and the agreement on a common identifier for a specific user. The federated model is illustrated in Figure 3.

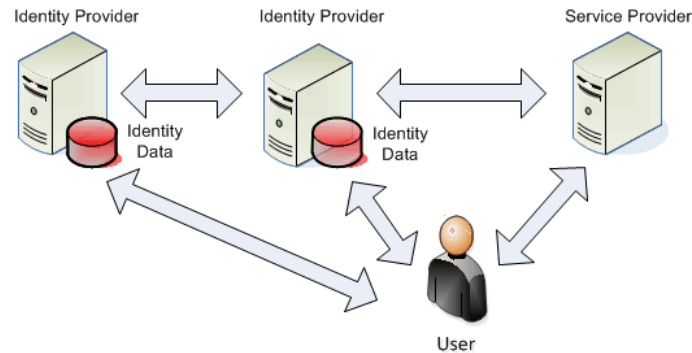


Fig. 3. Federated Model

3 Related Work

Taking the three different identity models into account, many approaches addressing identification and authentication already exist. In order to give an overview of existing solutions and to identify the relative benefits and shortcomings of different approaches, this section gives examples of existing and implemented models that the authors consider relevant.

3.1 Information Cards - Windows CardSpace

Information Cards representing personal digital identities define a technology for decentralized identity management within a so-called identity meta-system [6]. The concept of Information Cards can be realized by so-called Identity Selectors that manage various digital identities. Windows CardSpace [7] defines a well-known implementation of such an Identity Selector. CardSpace has been developed by Microsoft and is included in Microsoft Windows Vista and Microsoft Windows 7. However, other Identity Selector implementations such as the Higgins project [8] also exist.

By adopting Windows CardSpace, users should be provided with a more secure identification and authentication mechanism by using security tokens rather than e.g. simple username/password schemes. CardSpace's main aim is to facilitate user identification and authentication processes on web sites or web services and to enhance the achievable level of security.

Information Card's analogies in non-electronic environments are different IDs and plastic cards in a user's wallet. A so-called Identity Selector running on the user's client represents the user's wallet that contains various virtual identification and authentication cards (Information Cards). Information Cards only contain meta-information about how personal information can be retrieved from an identity provider that actually stores this information. For authentication at Web sites e.g. supporting Windows CardSpace (being denoted as "relying parties"), the user is

requested to select an appropriate card satisfying the requested claims by the web site out of his virtual wallet. The InfoCard is transmitted to the corresponding identity provider, which issues a security token including the requested information. This security token is transmitted to the requesting web site and the included information is verified by the relying party. In case of successful verification, the user is authenticated without any additional user interactions such as typing in usernames or passwords.

Since only the user has the possibility to select the desired InfoCard, and thus releases only required identity information, this model can be classified as a user-centric one.

3.2 Google Accounts Authentication

Google released its accounts authentication service [9] in 2006, with the help of which service and application providers no longer need to host or maintain a separate authentication service of their own. The provided services can be protected by a user's Google account. Access to a web application is granted if the user presents valid Google authentication data (username/password). Hence, the authentication process can be seen as outsourced to Google with no need for the application provider to handle any login information. Instead, the service provider receives an authentication token indicating a successful login.

The complete user's identification and authentication data is stored in central repositories managed by Google. Thus Google accounts authentication can be seen as a centrally-controlled identity model.

3.3 Liberty Alliance Project

The main objectives of the Liberty Alliance project [10] are:

- Development of open-standard-based specifications for federated identity management and identity-based web services, independent from the network architecture
- Providing an open and secure single sign-on solution using de-centralized authentication and authorization
- Secure management of user data and personal information for enterprises, considering privacy and policy issues

The Liberty Alliance Project has already released many frameworks covering such diverse topics as identity federation, identity web services, identity governance or identity assurance. The focus of the Liberty Alliance project lies on identity management and identity federation and thus can be seen as an example of a federated model.

4 System Architecture

In this section we discuss our proposed eID based authentication architecture and introduce its basic building blocks and design concepts. The entire system follows a smart-card based user-centric approach as discussed in Section 2.1. Hence, all relevant authentication and identification data are stored on smart-cards that are under the sole control of the respective user. This section shows how smart-cards (eIDs from several EU Member States) act in concert with corresponding central server-side authentication components and afford service providers and users a comfortable way to carry out identification and authentication processes.

Our approach is based on several open-source authentication and identification components being frequently used within Austrian e-Government solutions. Relying on established and approved components guarantees a high level of security of our authentication system. Basically, these components form the identity provider (IdP) that is used by service providers to carry out operations that involve a user's smart-cards for authentication. Thus, the IdP can be regarded as a middleware between service providers (hosting online applications) and the user's local system.

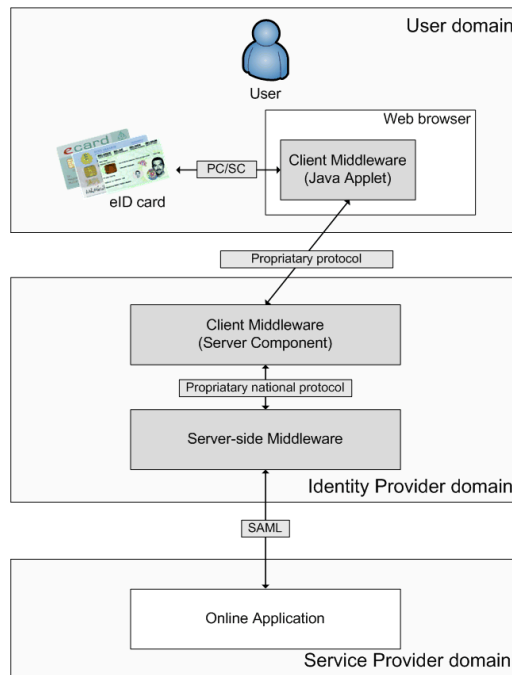


Fig. 4. eID based Authentication Architecture

Figure 4 shows the basic architecture and main components of our authentication system. The entire setup can be split up into three separate domains. The Service Provider domain contains the protected online application that can be accessed via common web browsers and requires user authentication. On the contrary, the User

domain comprises the user's local system including the smart-card and a web browser that serves as interface between the user and the protected online application. The user does not directly authenticate with the service provider. The IdP domain acts as trusted intermediary authentication middleware and carries out the whole authentication process with the user.

The IdP consists of three components. Besides the two major server-side components residing in the IdP domain, we can find a small client component in the User domain as well. All components communicate with each other over well-defined protocols. In order to guarantee confidentiality and integrity of the transmitted data, all interfaces provide support for TLS/SSL, which contributes to a higher degree of security of our solution. In the next subsections we discuss all single components of our IdP that build up the authentication path from the user's eID to the service provider.

4.1 Server-side Middleware

The server-side middleware is the only component of our IdP that directly interacts with the service provider. If an unauthenticated user requests a protected resource of an online application that requires user authentication, the service provider calls the server-side middleware component in order to trigger an appropriate user authentication process. The server-side middleware component is based on the open source module MOA-ID (Modules for Online Applications - Identification) [12], which is a project developed for user authentication and identification based on the Austrian national eID, the Austrian Citizen Card. When MOA-ID is triggered for starting an authentication process with an Austrian citizen, actually two process steps are carried out. In a first step, MOA-ID requests the user's identity information (identification process). This identity information is stored in a special XML data structure on the Austrian citizen card. After having successfully read the identity information from the card, the user is asked to create a qualified electronic signature indicating her willingness to authenticate at the online application (authentication process). The digital signature is verified by MOA-ID and on success, an authentication token (SAML assertion [11]) containing a unique identifier as well as additional relevant identity information (e.g. given name, family name and date of birth) is passed to the service provider. All communication between the server-side middleware MOA-ID and the user's eID is carried out using a client middleware, which we discuss in the next subsection. A well-defined communication interface called Security Layer hereby conveys XML commands from MOA-ID to the user's browser and thus to the client middleware in order to access the eID. The security architecture of the Austrian citizen card and the Security Layer interface is discussed in detail in [13].

In order to also support foreign eID cards, we adapted the identification and authentication components of MOA-ID. Foreign eID cards usually store identity information as part of their qualified signature certificate and not in a special data structure like in the Austrian case. Hence, in the identification process step, MOA-ID must distinguish between Austrian and foreign eID cards. In case of a foreign eID card, the identity information is read out of the digital certificate instead of a special

XML structure. The authentication process step remains the same since foreign users are prompted to create a qualified electronic signature as well. Our modifications in MOA-ID mainly affect signature verification and the creation of a unique identifier for foreign citizens. For successful signature verification, all root and intermediate certificates of certification authorities of the various Member States must be installed in the certificate- and trust-store of MOA-ID in order to be able to build a trusted certificate chain. Enhancements such as ETSI Trust Status Lists (TSL) [14] would facilitate this integration. Due to data privacy protection reasons, unique national identifiers (e.g. tax number) of foreign citizens will not be directly used for user identification and passed to the application. A secure one-way-derivation of these identifiers is transmitted to the requesting online application instead.

4.2 Client Middleware

The client middleware is the second core component of our IdP and complements the authentication system. The client middleware is in charge of preparing the creation of XML signatures, accessing the user's smart-card, performing required smart-card based operations, and providing the server-side middleware with relevant smart-card related data and created XML signatures. Our client middleware implementation is based on the Modular Open Citizen Card Architecture (MOCCA) [15]. MOCCA is a Java based open source project that has originally been developed for the Austrian e-Government. For our IdP, the functionality of MOCCA has been extended in order to achieve compatibility with other European smart-card based eID solutions as well.

The client middleware of our IdP solution basically consists of two components. A server component residing in the domain of the IdP is responsible for the communication with the server-side middleware MOA-ID, while smart-card access is implemented by a Java Applet running on the user's local system. In the following subsections, these two major components and their interactions are described in more detail.

4.2.1 Server Component

During an authentication process, the server component directly communicates with the server-side middleware over the Security Layer protocol interface. Whenever the server-side middleware MOA-ID needs to access the user's smart-card (e.g. to read certificate data or to create an electronic signature), it sends an appropriate XML request to the server component of MOCCA, which decodes the obtained request. The server component subsequently initiates the requested smart-card based operation, which is forwarded to the client component of MOCCA.

In general, the server component of MOCCA, the server-side middleware MOA-ID, and even the service provider's online application can be deployed on the same server. Since all interfaces between the main components of our authentication system can be secured by TLS/SSL, all components can also be deployed on different servers and connected over public networks without a loss of usability or security.

4.2.2 Client Component (Java Applet)

To access smart-cards that reside in the User domain, Java Applet technology is used by our authentication system. The Java Applet, which runs in the user's web browser environment, basically forms the second component of the client middleware. Once started in the user's web browser, the Java Applet receives requests from the server component via a well-defined interface. These requests specify what data to read, or operations to perform, on the user's smart-card. Results of the performed smart-card based operations (e.g. read certificates, computed signature values, etc.) are returned to the server component over the same interface. Basically the whole client middleware may run as an applet in the user's browser. However, in order to minimize the processing load on the user's side, e.g. XML digital signature processing operations like canonicalization etc., these operations are carried out by the server component of MOCCA. The applet itself is only in charge of processing the small amount of data required for card-based operations. All smart-card communication relies on the PC/SC protocol [16] and the exchange of appropriate application protocol data units (APDU). In order to interact with the PC/SC stack, the Java SmartCard I/O API - and thus a Java 6 runtime - is required on the user's browser.

While the basic functionality of the Java Applet has been provided by MOCCA, several adaptations of this open source solution were necessary. To support cards other than the Austrian eID card, the existing card recognition mechanism has been extended. Furthermore, existing methods for the accomplishment of smart-card based operations have also been enhanced so as to achieve compliance with different European eID cards. In the next section we show how these eID IdP components can be applied in practice together with popular and widely-used authentication solutions.

5 Integration in Central Authentication Service

In this section we show how our eID identity provider solution can be integrated into the Central Authentication Service (CAS) identity management system. By combining the centrally-controlled CAS authentication model with our user-centric approach, we established a hybrid solution resulting in a kind of federated authentication model.

CAS is a popular identity management solution, originally designed and developed by Shawn Bayern of Yale University, meanwhile maintained as open-source project by the Java Architectures Special Interest Group (JA-SIG) [17]. Its hallmark is a single-sign-on (SSO) protocol that allows users to access multiple services by requiring them to provide their authentication credentials only once. However, in contrast to Liberty Alliance or Shibboleth [18], CAS does not support trust federation with multiple identity providers.

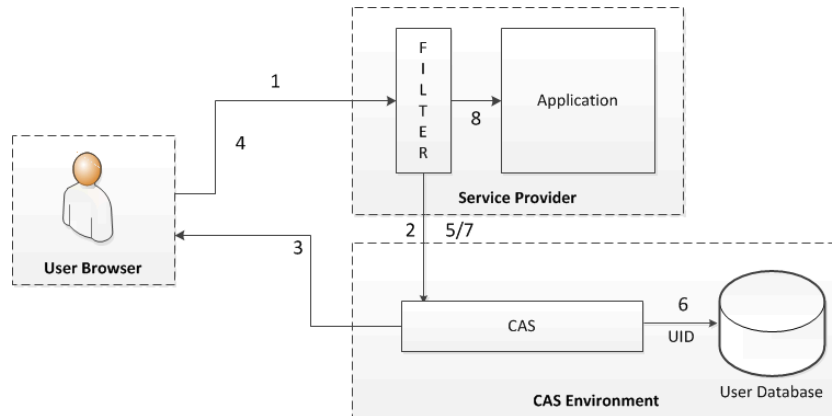


Fig. 5. General CAS authentication process flow

As illustrated in Figure 5, CAS authentication requires a central SSO or ticketing server, which has to be called before gaining access to the protected area of a service provider (SP) (1). Users have to provide their authentication information to the CAS server (e.g. by entering username/password) (2), which issues a SP-specific ticket after successful authentication (3). Users have to provide this ticket to the application of the SP (4), which must then validate the ticket against the CAS server (5/6). After successful authentication, the user gets access to the requested application (7/8) and is automatically authenticated when requesting applications of other service providers protected by this CAS instance. Applications with legacy authentication mechanisms can easily be “CAS-ified” by installing a so-called authentication interceptor. For instance, for the Java Servlet API there is a servlet filter available that handles all authentication parts and integrates seamlessly into each web application. Similar implementations are available for web servers (Apache etc.), special web applications (content management systems etc.) or server-side technologies (PHP, JSP, etc.).

The modular CAS architecture allows third parties to integrate custom authentication handlers that have to deal with two basic items: credentials and principals. Credentials are some kind of evidence in order to authenticate users. Some examples are the well-known username/password credentials, an X.509 certificate, a SAML token, etc. A principal represents the authenticated user. An authentication handler must thus evaluate if the user can be authenticated using the given credentials and if an associated principal can be resolved. Our proposed solution provides a custom authentication handler, which is invoked when CAS is accessed using a SAML ticket passed by our eID identity provider as credential. In general, principals may be of any kind. In our case study we used an LDAP user entry as principal data.

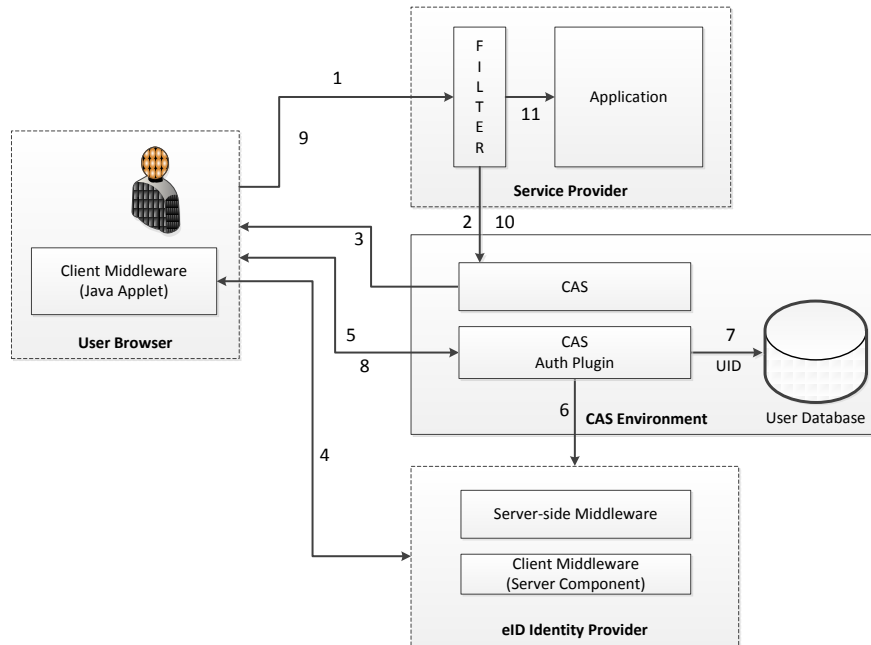


Fig. 6. CAS security architecture for eID based authentication

Figure 6 illustrates our CAS security architecture for eID based authentication with its four entities: user, service provider, CAS authentication server and eID identity provider. The authentication process works as follows: a user trying to access the web resource of a service provider is intercepted by an authentication filter (1). The filter determines that the user is not yet authenticated and thus has to be redirected to the central CAS login page (2). The user has the choice to login either with username and password or with an eID. When selecting the eID option (3), the MOCCA client middleware starts as an applet, automatically determines the eID card type and requests the user to enter the secret signature PIN code (4). Figure 7 illustrates this step in the case of a Belgium eID card. The user can also optionally display the signature data to be signed in a separate window. After successful signature verification and authentication with the eID identity provider, the MOCCA client middleware redirects the user together with the issued SAML ticket back to the CAS login page (5). Our authentication provider is implemented and hooked into CAS in such a way that it is automatically invoked when the login URL contains a SAML ticket (so-called SAML artifact) as HTTP parameter. The plug-in connects to the eID identity provider and fetches the SAML assertion belonging to the SAML ticket (6). In case of an invalid ticket, an appropriate error message is returned to the user. Otherwise, the assertion contains the unique ID (UID) of the authenticated user, which is used to query the LDAP database to determine the associated user's principal data (7). If no principal can be resolved, the user is redirected to a registration page with pre-filled form data extracted from the SAML assertion. Otherwise, the user is redirected to the service provider (8, 9) with a valid CAS authentication ticket as

HTTP parameter. The authentication filter of the service provider validates this ticket against the CAS server (10) and in case of success, access to the web resources is granted (11).

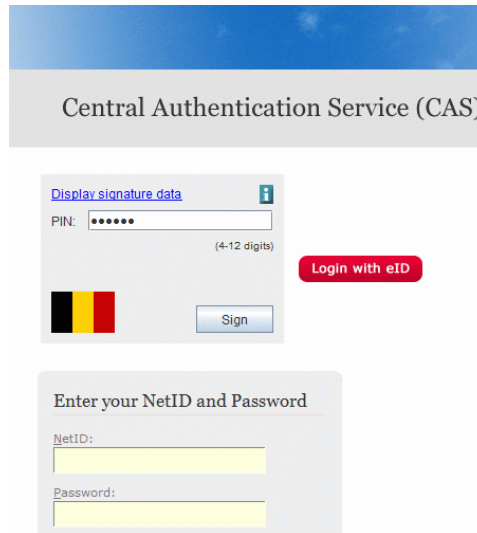


Fig. 7. CAS login sample with Belgium eID

Some remarks on eID based registration: in contrast to the default username/password authentication mechanism, an eID based registration does not necessarily require any user interaction and may be carried out seamlessly. The unique ID enables each user to be recognized and uniquely identified again at each login. A similar registration-less eID approach has been introduced in [20].

The CAS authentication plug-in is able to pass personal data to a particular registration page so that web forms can be pre-filled for the user. The range of personal details depends on their availability from the eID token. In case of an Austrian citizen, UID, given name, family name and date of birth are available. Other examples are Belgian and Estonian eIDs, which hold the same values as the Austrian eID, except for date of birth.

6 Remarks

Due to an open architecture and open interfaces, our solution can be easily integrated in existing environments. This way, access control to already existing online applications can be significantly increased in terms of security and usability.

Due to our modular design, basically any eID card that stores appropriate unique identifiers and which is capable of creating qualified electronic signatures can be easily integrated in our solution with minimal effort. So far, we have implemented card support for the Belgian and the Estonian eIDs, among others. Given the required

card specifications, including the respective APDU sequences, any eID card (e.g. Slovenian or Spanish eID) can be integrated into our solution.

Our solution follows a user-centric approach and is only suited to Member States with smart-card based eID solutions. The European large scale pilot STORK [21] (Secure idenTity acrOss bordeRs linKed) aims at enabling cross-border recognition of eIDs by establishing a platform for European eID interoperability. Due to its decentralized interoperability model, it will support any kind of governmental eID. In the STORK case, all citizens (except those from Germany and Austria) are redirected to their national identity provider when authenticating with a service provider that resides in a foreign Member State. Albeit supporting more kinds of governmental eIDs, this approach will depend on decentralized infrastructure components of other Member States. In our solution, by contrast, all authentication components can be located within the domain of the service provider. Under certain circumstances, this may ensure a continuous operation and faster process flows compared to decentralized approaches.

7 Conclusion

In this paper we discussed an identification and authentication solution, which supports eIDs from several EU Member States. As called for by the European i2010 initiative, and with respect to the EU Services Directive, cross-border authentication and eID interoperability is now on the agenda of all Member States.

The authors have based their work on several open source approaches, mainly the Austrian e-Government modules MOA-ID [12] and MOCCA [15], and the Yale University's CAS [17], respectively. These modules, which have a specific focus on a national smartcard environment (MOA-ID, MOCCA) and on username password authentication (CAS), have been combined.

The foreign eID card integration concerned to a great extent the modification and adaption of the Java applet code residing in the user's domain, i.e. card recognition and cryptographic functions, as well as the signature verification adoption in the server-side middleware MOA-ID. In order to show applicability in practice, we implemented an authentication plug-in for the popular and widely-used CAS identity management solution. The result was enhancing a CAS (i.e. username and password) based environment with different smart-card based national eIDs.

Nevertheless, due to the open architecture and open interfaces, our solution can also be easily integrated in other environments.

References

1. Kessler, G. C.: Passwords – Strengths and Weaknesses. In: Internet and Networking Security, J.P. Cavanagh (ed.), Auerbach, 1997
2. Siddhartha Arora, National e-ID card schemes: A European overview, Information Security Technical Report, Volume 13, Issue 2, May 2008, Pages 46-53, ISSN 1363-4127, DOI: 10.1016/j.istr.2008.08.002.

3. The European Parliament and the Council of the European Union: Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures, 1999
4. The European Parliament and the Council of the European Union: Directive 2006/123/EC of the European Parliament and of the Council of 12 December 2006 on services in the internal market, 2006
5. Palfrey J., Gasser U.: Digital Identity Interoperability and eInnovation, Case Study, November 2007, Berkman Publication Series
6. Microsoft Corporation, "Microsoft's Vision for an Identity Metasystem", May 2005, <http://www.identityblog.com/stories/2005/07/05/IdentityMetasystem.htm>
7. Microsoft Corporation, Windows CardSpace, <http://www.microsoft.com/windows/products/winfamily/cardspace/default.msp>
8. Higgins, Open Source Identity Framework, <http://www.eclipse.org/higgins/>
9. Authentication for Web Applications, <http://code.google.com/apis/accounts/docs/AuthForWebApps.html>
10. The Liberty Alliance Project, <http://www.projectliberty.org/>
11. OASIS TC, Security Assertion Markup Language (SAML), http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security
12. Modules for Online Applications – Identification (MOA-ID) <http://egovlabs.gv.at/projects/moa-idsps/>
13. Leitold, H., Hollosi, A., Posch, R., Security Architecture of the Austrian Citizen Card Concept, Proceedings of 18th Annual Computer Security Applications Conference, 2002.
14. ETSI TS 102231 – Electronic Signatures and Infrastructures (ESI); Provision of harmonized Trust-service status information, v3.1.1, 10/2009.
15. Center, M., Orthacker C., Bauer, W., Minimal-Footprint Middleware for the Creation of Qualified Signatures, Proceedings of WEBIST 2010 – International Conference on Web Information Systems and Technologies.
16. Interoperability Specification for ICCs and Personal Computer Systems. Available from <http://www.pcscworkgroup.com/specifications/overview.php>
17. Central Authentication Service (CAS), Java Architectures Special Interest Group (JA-SIG), <http://www.jasig.org/cas>
18. Shibboleth, a project of the Internet2 Middleware Initiative. <http://shibboleth.internet2.edu/>
19. Ivkovic, M, Leitold, H, Rössler, T 2009, 'Interoperable elektronische Identität in Europa', in 7. *Information Security Konferenz* (pp. 175 – 190)
20. Orthacker, C.; Zwattendorfer, B.: Seamless eID Integration into Web Portals. - in: Electronic Government: Proceedings of ongoing research and projects of EGOV 09. (2009), S. 297 - 304
21. Secure Identity Across Borders Linked (STORK), <https://www.eid-stork.eu/>