Thomas Lenz, Klaus Stranacher, Thomas Zefferer: "Towards a Modular Architecture for Adaptable Signature-Verification Tools". In Proceedings of the 9th International Conference on Web Information Systems and Technologies, pp. 325-334, INSTICC, 2013.

TOWARDS A MODULAR ARCHITECTURE FOR ADAPTABLE SIGNATURE-VERIFICATION TOOLS

Thomas Lenz, Klaus Stranacher and Thomas Zefferer
Secure Information Technology Center - Austria, Inffeldgasse 16a, Graz, Austria
{*thomas.lenz, klaus.stranacher, thomas.zefferer*}*@a-sit.at*

Keywords: Electronic Signatures, Verification, Testing, Web Services

Abstract: The verification of electronic signatures represents a key component of security-sensitive applications. Signature-verification tools need to meet several requirements regarding security, reliability, usability, and accessibility. A conducted survey revealed that existing signature-verification tools often meet only a subset of these requirements. In most cases, available tools support a limited set of document and signature formats only, or do not feature appropriate interfaces that allow both end users and third-party applications to access the tool's functionality in a convenient way. This complicates the development of electronic signature based third-party applications and reduces the usability for end users. To solve this problem, we propose a new architecture for Web based signature-verification tools. The proposed architecture follows a plug-in based approach that eases the integration of new signature formats and interfaces. The practical applicability of the proposed architecture is demonstrated by means of a concrete implementation covering different use cases. This implementation demonstrates that the proposed architecture facilitates the realization of signature-verification tools that are able to meet all requirements of end users and third-party applications. This way, the proposed architecture and the implemented solution contribute to the security, usability, and efficiency of present and future electronic signature based applications.

1 INTRODUCTION

Electronic signatures are an integral component of various solutions related to security sensitive fields of application. Such solutions typically have strict requirements regarding data integrity, authenticity, and non-repudiation of origin. Electronic signatures are perfectly suitable to meet these requirements. The importance of electronic signatures has also been recognized by legislative bodies. For instance, the European Union has harmonized the use of electronic signatures in EU Member States through the Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures (The European Parliament and the Council of the European Union, 2000), henceforth referred to as EU Signature Directive. The EU Signature Directive distinguishes between advanced and qualified electronic signatures. Compared to advanced electronic signatures, qualified electronic signatures have to fulfill several additional security requirements[1] and are defined to be legally equivalent to handwritten signatures.

The legal equivalence to handwritten signatures makes qualified electronic signatures especially useful for the realization of e-government solutions provided by the public sector. Numerous countries are already issuing e-ID and e-signature tokens to their citizens. Citizens can use these tokens to remotely authenticate at e-government portals, to carry out electronic procedures, and to electronically sign documents. Smart card based e-ID and e-signature tokens have for instance been issued to citizens in Austria[2], Belgium[3], Estonia[4], Germany[5], Portugal[6], or Spain[7]. A few countries such as Austria[8] and Estonia[9] additionally allow citizens to use their mobile phones as

[1]These requirements basically cover the use of secure signature-creation devices (e.g. smart cards or similar secure elements) and reliance on qualified electronic signatures.

[2]https://www.buergerkarte.at/

[3]http://eid.belgium.be/en/

[4]http://www.id.ee/

[5]http://www.personalausweisportal.de

[6]http://www.cartaodecidadao.pt/

[7]http://www.dnielectronico.es/

[8]https://www.handy-signatur.at/Default.aspx

[9]http://e-estonia.com/components/mobile-id

e-ID and e-signature tokens. In numerous countries, also service providers from the private sector rely on e-ID and e-signature tokens issued by the public sector. For instance, several e-banking portals support a secure user authentication based on available e-ID and e-signature tokens and allow users to authorize financial transactions remotely using electronic signatures. Additionally, electronic signatures are also frequently used in the private domain, e.g. to sign electronic contracts or similar bilateral agreements. Required signature tokens and signing certificates can usually be acquired from national certification authorities. Due to their various fields of application in both the public and the private domain, electronic signatures are increasingly gaining importance all over the world.

A key advantage of electronic signatures compared to handwritten signatures is their verifiability. Given a handwritten signature, its authenticity and originality is difficult to determine in practice. Additionally, subsequent modifications that have been applied to e.g. a signed contract might be difficult to detect in case of handwritten signatures. This is not the case for electronic signatures. The validity, authenticity, and originality of electronic signatures can be unambiguously determined by means of cryptographic operations that are based on strong mathematical foundations. Each electronic signature is unambiguously linked to the identity of a certain person by means of an electronic certificate that has been issued by a trusted third party, i.e. a certification authority. Additionally, each subsequent modification of a signed document immediately invalidates the electronic signature. Thus, by verifying its electronic signature, integrity, authenticity, and non-repudiation of origin of signed data can be reliably assessed.

In practice, the verification of electronic signatures is actually no trivial task as it requires the application of complex cryptographic methods. Hence, there is a need for appropriate tools that implement this functionality. Tools and Web based services that allow for a convenient verification of electronic signatures have already been deployed in several countries. Austria is a representative example, since this country has started to rely on electronic signature based solutions early (Leitold H., 2002). Austrian citizens are provided with a Web based verification tool that can be used to upload and verify electronically signed documents (Zefferer et al., 2011). This tool supports various document and signature formats and is therefore able to act as single point of contact for the verification of arbitrary electronic signatures.

Unfortunately, the Austrian signature-verification tool is optimized for manual use by citizens, i.e. the only opportunity to interact with this tool is by manually uploading signed documents. This significantly aggravates an automated use of this tool, e.g. the integration of the tool's functionality into third-party applications, which for instance need to implement an on-the-fly verification of electronically signed documents. To overcome this issue, we present an improved signature-verification tool that enhances the existing solution by extended interfaces and communication capabilities. This way, approved components and key functionality of the existing solution are maintained, while access to the tool's functionality is facilitated.

The remainder of this paper is structured as follows. In Section 2, general requirements of signature-verification solutions are defined. In Section 3, we show that existing signature-verification solutions are often not able to meet all of these requirements. In Section 4, we present an enhanced architecture of the approved Austrian signature-verification tool that improves its functionality and accessibility. We demonstrate the practical applicability of our approach by presenting a concrete implementation that is based on the proposed architecture in Section 5. Finally, conclusions are drawn in Section 6.

2 REQUIREMENTS

The secure and reliable verification of electronic signatures represents a key component of electronic signature based applications and services. Signature-verification tools must meet various requirements in order to satisfy the needs of users and service providers. Requirements that need to be met by signature-verification tools have been identified and discussed in detail in (Zefferer et al., 2011). However, these requirements mainly focus on the end-user perspective and do not cover aspects that are specific to providers of electronic signature based applications and services. Considering the needs of both users and service providers, the following key requirements for signature-verification tools can be derived.

- **Security:** Electronic signatures are typically used in security-sensitive applications. If signature-verification tools process security-sensitive data, these data need to be protected appropriately in order to assure their confidentiality. Furthermore, signature-verification tools need to be resistant against attacks that threaten to illegally influence results of signature-verification processes.
- **Reliability:** End users and service providers that make use of signature-verification tools must be able to rely on the results of signature-verification

processes. Thus, signature-verification tools must be able to correctly distinguish between valid and invalid electronic signatures at any time. Given the legal equivalence of qualified electronic signatures to handwritten signatures, the correctness of presented verification results is of particular importance.

- **Usability:** The requirement for usability covers several aspects such as simplicity of verification processes from the user point of view, hiding of complexity, platform independence, or avoidance of local software installations. An important aspect for both end users and service providers that make use of signature-verification tools is the requirement for a single point of contact. Signature-verification tools need to provide users and service providers a single interface, through which arbitrary document and signature formats can be verified.
- **Accessibility:** The functionality provided by signature-verification tools must be easily accessible for both end users and providers of third-party applications. To meet this requirement for end users, the provided user interface needs to comply with established usability and accessibility standards such as the Web Content Accessibility Guidelines (WCAG) (World Wide Web Consortium, 2008a). In order to meet the accessibility requirement for providers of third-party applications, signature-verification tools need to implement appropriate interfaces that can be accessed by external applications to carry out signature-verification processes.

The above defined requirements are rather generic and not bound to a specific legal framework. Depending on the legal and organizational environment, in which a signature-verification tool is deployed, several additional requirements might apply. For instance, in the special case of Austria, signature-verification tools have to support proprietary signature formats that are frequently used in Austrian e-government solutions. Different legal requirements of different countries have already led to the development of numerous signature-verification tools. In the next section, available signature-verification tools are surveyed and their capabilities to meet the above defined requirements are assessed.

3 EXISTING SOLUTIONS

The growing importance of electronic signature based solutions and the plurality of legal requirements have led to the development of different signature-verification solutions. Depending on the context of the deployment, these solutions usually meet a subset of the requirements defined in Section 2.

For instance, Unizeto Technologies SA[10] provides a publicly available Web based signature verification tool called WebNotarius[11]. Having its roots in Poland, WebNotarius supports the verification of public-key certificates issued by certified Polish certification authorities. Additionally, WebNotarius supports the verification of different document and signature formats including PKCS#7 (RSA Laboratories, 1993), CMS (Housley, 2009), S/MIME (Ramsdell and Turner, 2010), and XMLDSig (World Wide Web Consortium, 2008b).

A similar Web based signature-verification tool is provided by the German company signagate[12]. Compared to WebNotarius, signagate is restricted to the PDF file format (Adobe Corporation, 2008) and does not support different document and signature formats. Another Web based verification tool for electronically signed PDF documents is provided by the company Secured Signing[13]. The company ascertia[14] offers Web based tools for the verification of electronically signed PDF and XML files. A solution for the verification of XML signatures called MOA-SP[15] is also provided by the Austrian government to facilitate the integration of XML signatures into Austrian e-government applications. MOA-SP is available as open source and features API and Web service based interfaces for the verification of electronically signed XML documents, namely XMLDSig (World Wide Web Consortium, 2008b) and XAdES (ETSI TS 101 903, 2010).

There exist also several signature-verification activities at a European level. The tool SD-DSS[16], commissioned by the EU Commission, supports the verification of signature formats defined by the Commission Decision on establishing minimum requirements for the cross-border processing of documents (European Commission, 2011). Additionally, the EU large scale pilots PEPPOL[17] and SPOCS[18] addressed issues concerning the validation of electronic sig-

[10] http://www.unizeto.pl/
[11] http://www.webnotarius.eu
[12] http://www.signagate.de/
[13] http://www.securedsigning.com/
[14] http://www.ascertia.com/
[15] https://joinup.ec.europa.eu/software/moa-idspss/description
[16] https://joinup.ec.europa.eu/software/sd-dss/home
[17] http://www.project.peppol.eu/
[18] http://www.eu-spocs.eu/

natures. PEPPOL developed a signature validation service with focus on public procurement processes. Within SPOCS, signature verification is part of the validation of electronic documents concerning the issues raised by the EU Services Directive (The European Parliament and the Council of the European Union, 2006).

All above mentioned solutions show very well the key problem of current signature-verification solutions. Most solutions are limited to the verification of certain document and signature formats such as XML or PDF. Hence, these solutions do not satisfy the usability requirement for a single point of contact for the verification of all document and signature formats. Even the tool WebNotarius, which supports several different formats, only covers a subset of all possible document and signature formats. This is actually not surprising, as proprietary signature formats exist in several countries due to national legal requirements. For instance, in Austria a proprietary PDF signature format (Leitold et al., 2009) has been introduced for the public sector in order to meet specific legal requirements (Leitold et al., 2010). Support for all national and international, standardized and proprietary signature formats rapidly increases the complexity of signature-verification tools.

This situation is even aggravated by the fact that especially proprietary signature formats are subject to frequent revisions and updates. As electronic signatures need to retain their validity even if the underlying signature format is updated, signature-verification tools have to maintain and support different versions of signature formats. This again increases the complexity of such tools and renders their development and maintenance difficult. As a first solution to this problem, Stranacher and Kawecki have proposed a mechanism to incorporate external verification services (Stranacher and Kawecki, 2012). However, this proposal lacks on an appropriate and efficient document and signature format detection.

In order to cope with the growing diversity of different document and signature formats, a Web based signature-verification tool has been developed in Austria. This tool features a modular design and implements an efficient format detection engine that eases the integration of new document and signature formats. The signature-verification tool that has been discussed in detail in (Zefferer et al., 2011) has basically proven its practical applicability during several years of productive operation. Still, this tool suffers from several limitations. For instance, the tool is indented for manual use only. Users can upload documents to be verified through a Web based user interface. As this is the only supported interface, the tool's functionality cannot be easily accessed by external applications to carry out signature verifications. Furthermore, the given limitation to a Web based interface complicates the provision of the tool's functionality through new communication channels and emerging technologies such as mobile apps. Thus, this tool is obviously not able to meet accessibility requirements for service providers and third-party applications.

In summary it can be stated that there is currently no perfect solution available. From the Austrian perspective, powerful tools such as WebNotarius that have been developed in other countries are no alternative, as these solutions do not support proprietary document and signature formats that are specific to Austria. Hence, these solutions do not meet the predefined requirement for provision of a single point of contact for the verification of all document and signature formats. Unfortunately, the existing Austrian solution that supports these proprietary document and signature formats does not feature appropriate interfaces to meet the predefined requirement for accessibility. Hence, third-party applications are not able to access the functionality provided by the available tool in order to implement fully automated signature-verification processes of electronically signed documents.

To overcome this problem, we propose an enhancement of the existing Austrian signature-verification tool. The proposed enhancements improve the accessibility of this tool especially for third-party applications and facilitate access to the tool's functionality. The architectural design of the proposed solution is presented in the next section.

4 ARCHITECTURAL DESIGN

The proposed solution is based on the Austrian Web based signature-verification tool that has been discussed in the previous section. The Web based approach followed by this tool has proven to be advantageous in terms of security and usability during several years of productive operation (Zefferer et al., 2011). Figure 1 illustrates the general architecture of this tool. Key component of the entire solution is the *Process Flow Engine*, which coordinates the different steps of a signature-verification process. A signature-verification process basically consists of two steps. *First*, the document and signature format of the document to be verified is determined. This task is accomplished by the *Format Detection Engine*. For each supported format, an appropriate *Format Detection Plug-in* has been implemented. Internally, the dif-

ferent Format Detection Plug-ins are organized hierarchically in a tree structure. This way, the runtime for format-detection processes grows only logarithmically with an increasing number of format-detection plug-ins. *Second*, the electronic signature is verified by the *Verification Engine*. Depending on the determined document and signature format, an appropriate *Verification Plug-in* is selected to accomplish this task. The selected Verification Plug-in extracts all signatures from the provided document and cryptographically verifies their validity based on the provided document data.

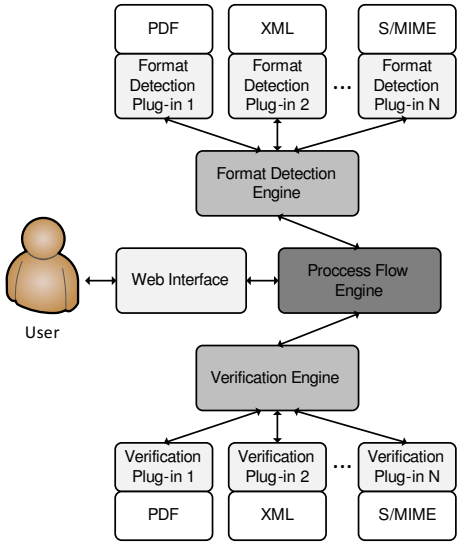

Figure 1: Architecture of the Austrian Web based signature-verification tool.

Files to be verified as well as verification results are exchanged with end users by means of a *Web Interface*. Users are provided with a Web form in order to upload signed documents to be verified. Results are directly presented in the users' Web browsers. After completion of the signature-verification process, users can additionally download an electronically signed version of the verification report in PDF format.

The concept shown in Figure 1 has turned out to be able to especially meet the expectations and requirements of end users. However, several problems arise if the functionality of this tool has to be accessed by third-party applications. As this tool has originally been developed for end users, its functionality can basically be accessed through the provided Web based interface only[19]. This limitation renders an integration of the tool's functionality into third-party applications for automated signature verifications difficult.

In order to solve this issue and to facilitate an integration of the tool's functionality into third-party applications, we have enhanced the tool's general architecture. The resulting extended architecture of our solution is illustrated in Figure 2.

The lack of appropriate interfaces to access the tool's functionality through different communication channels and technologies has turned out to be the main drawback of the existing solution. Our improved solution solves this issue by replacing the original Web interface by a more powerful I/O Engine. This I/O Engine supports different I/O channels and communication technologies. The I/O Engine adopts the approved plug-in based approach that is already successfully followed by both the Format Detection Engine and the Verification Engine. This way, the entire solution remains modular and easily adaptable to future requirements.

The adapted architecture and the introduced I/O Engine facilitate the realization of various additional use cases. Figure 2 illustrates some of them. Of course, the approved Web based interface is also compatible to the new architecture. A special I/O Plug-in can implement the required interface. However, the enhanced architecture is not limited to a Web based interface any longer. For instance, the proposed architecture also allows the tool's functionality to be accessed through a SOAP based Web-service interface provided by the tool's I/O Engine. The modular design also allows for efficient implementations of test frameworks that automatically run verification tests on well-defined test documents stored in local databases. This could be especially useful for the maintenance and further development of the tool. Finally, the proposed architecture also allows for the integration of new and emerging technologies such as smartphone apps that access the tool for an on-the-fly verification of electronic signatures.

We have evaluated the practical applicability of the proposed architectural design by realizing three of the above-mentioned use cases in practice. Details on the realization of these use cases are provided in the next section.

5 USE CASES

Core component of the proposed architecture is the I/O Engine that replaces the original Web based user interface and allows for the implementation of different I/O Plug-ins. These plug-ins implement different communication technologies and make the

[19] Actually, the tool provides also a command line based user interface. However, this interface is not appropriate for an integration of the tool's functionality into remote third-party applications either.

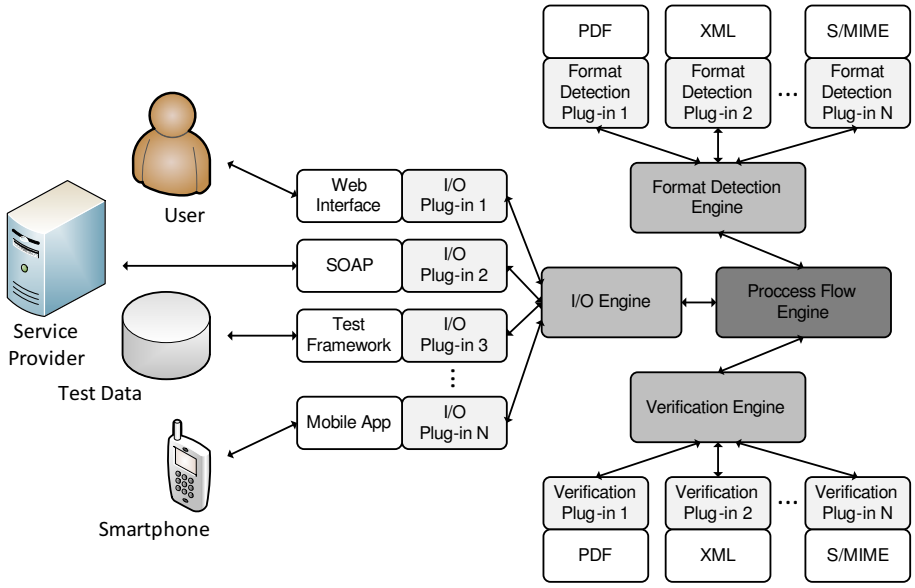

Figure 2: Enhanced architecture of the Austrian Web based signature-verification tool.

signature-verification tool's functionality accessible through different communication channels. This way, the proposed tool can be used in different application scenarios. We have demonstrated the applicability of our solution by implementing solutions for three concrete use cases, which are discussed below in more detail.

5.1 Use Case 1: Web Interface

Use Case 1 covers the scenario, in which an end user wants to access the tool's functionality in order to verify an electronically signed document. This is basically the scenario, which the predecessor of the proposed tool has been developed for. For this scenario, a Web based interface has turned out to be the most appropriate solution. Therefore, our implementation of this use case relies on the approved Web based interface of the original tool. Of course, it was necessary to redesign the existing API interface in order to connect the Web based user interface to the new I/O Engine of our solution. While several internal components have been redesigned, the user interface itself has not been changed. This way, existing productive instances of the tool can easily be upgraded and do not require end users to deal with new interfaces. Figure 3 illustrates the implemented Web interface that can be used by end users to upload and verify signed documents.

5.2 Use Case 2: SOAP

This scenario covers the case, in which a remote third-party application makes use of the signature-verification tool's functionality. In order to allow third-party applications to access our tool through a well-defined interface, we have implemented another I/O Plug-in. This plug-in provides a standardized interface, which can be used by remote applications to carry out automated signature-verification process.

This is actually no completely new approach. There are already different standards that define appropriate interfaces for remote signature-verification services. Popular examples are the OASIS Digital Signature Service (OASIS, 2007) and MOA-SP, which is mainly used in Austrian e-government solutions. However, these standardized interfaces specify the verification of a limited set of electronic signatures only. Consequently, these services are not suitable for scenarios, in which different document and signature formats need to be supported. Our solution specifies a new signature-verification interface, which fulfills these additional requirements.

The developed I/O Plug-in implements a Web service, which uses the SOAP protocol (Gudgin et al., 2007) to exchange information. Beneath the SOAP protocol, the Hypertext Transfer Protocol (HTTP) (Fielding et al., 1999) is used as carrier for the SOAP message. This is reasonable, because HTTP is popular, frequently used, and widely supported. SOAP messages being exchanged over the implemented Web service interface rely on the Extensible Markup Language (XML) (Bray et al., 2006). To meet all requirements of this use case, we have defined our own XML schema for document verification[20]. The XML schema, which defines the structure

[20]We were forced to define an own schema, since existing schemata were not able to meet our requirements.

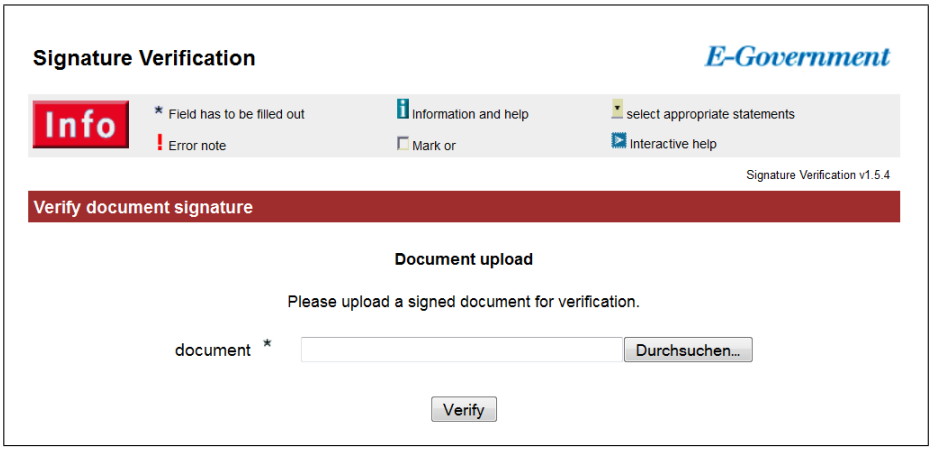

Figure 3: Web front-end of the signature-verification tool.

of a signature-verification request that is accepted by the tool's Web-service interface, is shown in Listing 1.

According to the defined XML schema, a signature-verification request consists of two XML elements. The *Document* element is mandatory and contains the signed file to be verified. The second element is optional and can be used to identify the signed file. When a schema-compliant SOAP request is received, a signature-verification operation as shown in Figure 4 is triggered.

Listing 1: VerifyDocumentRequest XML schema.

```
<xsd:element name="VerifyDocumentRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Document"
        type="xsd:base64Binary"/>
      <xsd:element name="FileID"
        type="xsd:token"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

The first step of the whole verification process consists of several preprocessing operations, like XML schema validation and BASE64 decoding. Afterwards, the provided document's format is determined using the tool's Format Detection Engine. Subsequently, the document's electronic signatures are verified using the tool's Signature Verification Engine.

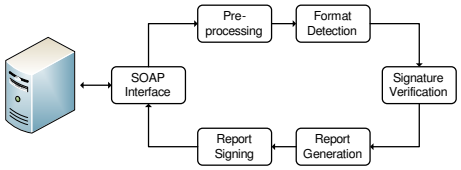

Figure 4: Web service based document verification process.

Results of the format-detection step and the signature-verification step are collected to generate a verification report. Relevant parts of this verification report are shown in Listing 2. The report's *FileInfo* element contains the *FileID* of the verified file, the detected document format, and the computed hash value of the document. Signature-verification results generated by the Verification Engine are embedded in the report's *SignatureInfo* element.

Listing 2: VerificationReport XML Schema.

```
<xsd:complexType name="VerificationReportType">
  <xsd:sequence>
    <xsd:element name="FileInfo"
      type="tns:FileInfoType"/>
    <xsd:element name="SignatureInfo"
      type="tns:SignatureInfoType"/>
  </xsd:sequence>
</xsd:complexType>
```

Listing 3: VerifyDocumentResponse XML Schema.

```
<xsd:element name="VerifyDocumentResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="VerificationReport"
        type="tns:VerificationReportType"/>
      <xsd:element name="Signature"
        type="dsig:SignatureType"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Finally, the generated verification report is electronically signed in order to ensure its authenticity and integrity. The signed report is returned to the sender of the SOAP based verification request. Listing 3 shows relevant parts of the XML schema that specifies the structure of this SOAP response. The response consists of two parts, the *VerificationReport* and a *Signature* element, which contains the electronic signature of the verification report. This sig-

nature is created according to the XMLDSig standard using the enveloped-signature scheme.

The implemented SOAP based Web-service interface provides third-party applications a common interface for on-the-fly signature verifications. Third-party applications can use this interface to efficiently carry out verification operations on different document and signature formats. This way, the provided solution facilitates the implementation of electronic signature based applications by encapsulating and providing common functionality.

5.3 Use Case 3: Test Framework

The growing number of document and signature formats rapidly increases the complexity for developers and providers of the signature-verification tool. The situation is even more complicated by the fact that verification results are not only influenced by the input document and the implementation of the verification tool, but also by the tool's configuration. For instance, the validity of an electronic signature depends to a large extent on root and intermediate certificates that are configured in the tool's trust stores. Hence, extensive tests are not only required during development, but also after deployment. This use case describes how the tool's improved architecture is used to implement a comprehensive test framework that allows both developers and operators to verify the correct behavior of the tool.

The implemented test framework makes use of the tool's I/O Engine and implements an appropriate I/O Plug-in. Furthermore, the test framework makes use of a database containing signed test documents and corresponding expected verification results. By feeding the test documents as input into the signature-verification tool and comparing the obtained results with the stored expected results, the functional integrity of the tool can be verified. Figure 5 gives an overview of the test framework's general architecture.

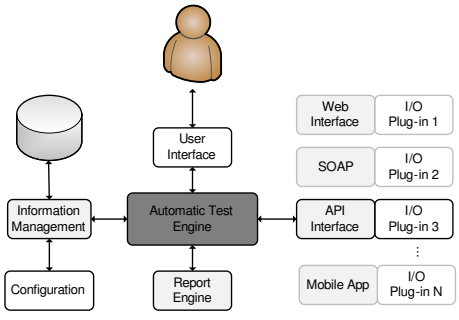

Figure 5: Test Framework with sub-modules and interconnections.

The main part of the test framework is the *Automatic Test Engine*, which controls the whole automatic verification and test process. The Automatic Test Engine makes use of several additional sub-modules. These sub-modules implement an *User Interface*, an *Information Management* module, and a *Report Engine*. As interconnection to the signature-verification tool, we make use of the signature-verification tool's I/O Engine and its plug-in based architecture.

The Information Management sub-module is used to manage the signed test documents and the corresponding expected results. We use a file system based method to manage the different files depending on their document type and signature format. The expected results are stored in XML files. Listing 4 shows the XML schema, which defines the internal structure of these files.

Each *file* element represents the expected verification result of a test document by using a set of child elements. The *name* element represents the name of the signed test document. The number of signatures contained in the test documents is represented by the *numberofsignatures* element. For each signature, the expected verification result is stored in a separate *signatures* element. For comprehensive functionality tests, the test-document database also contains documents without signatures and documents with incorrect signatures. All incorrect documents are marked with the *executable* flag. XML files with expected verification results are stored in all directories and subdirectories of the test-document hierarchy.

Listing 4: XML schema of the verification result database.

```
<complexType name="files">
  <sequence>
    <element name="path"
      type="string"/>
    <element name="file"
      type="tns:file"/>
  </sequence>
</complexType>
<complexType name="file">
  <sequence>
    <element name="name"
      type="string"/>
    <element name="executable"
      type="boolean"/>
    <element name="numberofsignatures"
      type="string"/>
    <element name="signatures"
      type="tns:signature"/>
  </sequence>
</complexType>
```

Besides the Information Management sub-module, the Report Engine represents another important component of the test framework. The Report Engine implements the entire report func-

tionality. A report basically contains the number of tested documents and the number of documents, in which the signature-verification result matches the expected verification result. If the verification result of a test document does not match the expected result, a detailed report is generated. This report contains a detailed error description and shows, which part of the signature verification has caused the problem. By default, the report is rendered as HTML document and presented to the user via the user interface. Additionally, the Report Engine can generate an XML based report or a textual report.

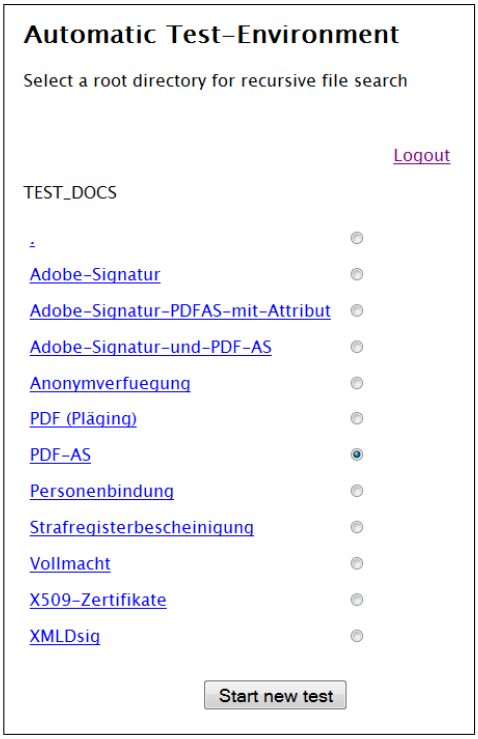

Figure 6: Web based user interface of the Test Engine after login.

Developers and operators of the signature-verification tool can interact with the test framework by using a Web based User Interface. Figure 6 illustrates the user interface after a successful log-in. The shown list represents the directory hierarchy, in which the test documents are organized. The user can select one type of documents (i.e. one directory) from the test-document database by selecting the corresponding radio button. Afterwards, the test operation can be started using the *Start new Test* button. The Automatic Test Engine controls the entire test process and uses the Information Management sub-module to get all documents from the selected directory. Every document is verified by the signature-verification tool. Obtained verification results are compared with the expected verification results. The result of this comparison is stored and processed by the Report Engine. When all selected documents have been tested, the generated HTML report is shown in the Web based user interface. Figure 7 shows an excerpt from a test result, in which 457 test documents are in use. Obtained test results can be stored or alternatively a new test run can be started.

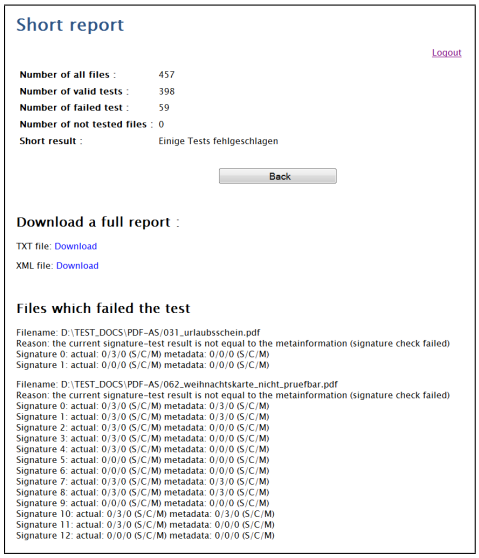

Figure 7: Web based user interface of the Test Engine with test report.

6 CONCLUSIONS

The secure and reliable verification of electronic signatures is an integral component of security-sensitive applications from various fields of application such as e-government, e-banking, or e-business. The capability to assess the validity of electronic signatures can be important for both end users and service providers. In this paper we have presented a new solution for the verification of electronic signatures. The presented solution features a single point of contact for the verification of various document and signature formats and relies on a modular architecture that facilitates future extensions of the solution's functionality. Although the presented solution has been developed to meet the special requirements of the Austrian legal framework, its general architectural design and implementation is also applicable in other contexts.

We have demonstrated the practical applicability and flexibility of the presented architectural design by implementing solutions for different use cases. These use cases cover the use of the presented solution by end users through a Web based user interface, the pro-

vision of the solution's functionality through a well-defined SOAP based Web-service interface, and the realization of a comprehensive test framework that assists is assessing the correct functionality of our solution. The realization of further use cases such as the implementation of mobile smartphone apps that make use of the presented signature-verification tool is regarded as future work.

Due to its modular architecture, the presented solution is dynamically extensible especially with respect to new document formats and communication interfaces. This distinguishes the presented solution from other signature-verification tools that are available on the market. A conducted survey has revealed that these tools are typically limited to certain document and signature formats, or to certain communication interfaces. The presented solution removes these limitations and thereby contributes to the security, usability, and efficiency of present and future electronic-signature based applications.

REFERENCES

Adobe Corporation (2008). Document management - Portable document format Part 1: PDF 1.7.

Bray, T., Paoli, J., Sperberg-McQueen, C., Maler, E., Yergeau, F., and Cowan, J. (2006). Extensible Markup Language (XML) 1.1 (Second Edition). http://www.w3.org/TR/2006/REC-xml11-20060816/.

ETSI TS 101 903 (2010). Electronic Signatures and Infrastructures (ESI); XML Advanced Electronic Signatures (XAdES) V1.4.2.

European Commission (2011). European Commission Decision, Establishing minimum requirements for the cross-border processing of documents signed electronically by competent authorities under Directive 2006/123/EC of the European Parliament and of the Council on services in the internal market, notified under document C(2011) 1081, 2011/130/EU. http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2011:053:0066:0072:EN:PDF.

Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and Berners-Lee, T. (1999). Hypertext transfer protocol – http/1.1. http://www.ietf.org/rfc/rfc2616.txt.

Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J.-J., and Nielsen, H. F. (2007). Soap version 1.2 part 1: Messaging framework. http://www.w3.org/TR/soap12-part1/.

Housley, R. (2009). Cryptographic Message Syntax (CMS). http://www.ietf.org/rfc/rfc5652.txt.

Leitold, H., Posch, R., and Rössler, T. (2009). Media-break resistant eSignatures in eGovernment-An Austrian experience. In Dimitris Gritzalis, J. L., editor, *Emerging Challenges for Security, Privacy, and Trust - 24th IFIP SEC*, volume IFIP AICT 297 of *IFIP Advances in Information and Communication Technologies*, pages 109 – 118. Springer.

Leitold, H., Posch, R., and Rössler, T. (2010). Reconstruction of electronic signatures from eDocument printouts. *Computers and Security*, 29(5):523 – 532. Challenges for Security, Privacy and Trust.

Leitold H., Hollosi A., P. R. (2002). Security Architecture of the Austrian Citizen Card Concept. In *Proceedings of 18th Annual Computer Security Applications Conference (ACSAC'2002), Las Vegas, 9-13 December 2002. pp. 391-400, IEEE Computer Society, ISBN 0-7695-1828-1, ISSN 1063-9527.*, pages 391–400.

OASIS (2007). Digital Signature Service Core Protocols, Elements, and Bindings Version 1.0. http://docs.oasis-open.org/dss/v1.0/oasis-dss-core-spec-v1.0-os.pdf.

Ramsdell, B. and Turner, S. (2010). Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification. http://tools.ietf.org/html/rfc5751.

RSA Laboratories (1993). PKCS#7: Cryptographic Message Syntax Standard. ftp://ftp.rsasecurity.com/pub/pkcs/ascii/pkcs-7.asc.

Stranacher, K. and Kawecki, T. (2012). Interoperable Electronic Documents. In Scholl, Flak, Janssen, Macintosh, Moe, Sbø, and Wimmer, editors, *Electronic Government and Electronic Participation - Joint Proceedings of Ongoing Research and Projects of IFIP EGOV and IFIP ePart 2012*, volume 39 of *Informatik*, pages 81 – 88. Trauner.

The European Parliament and the Council of the European Union (2000). Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures. http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2000:013:0012:0020:EN:PDF.

The European Parliament and the Council of the European Union (2006). Directive 2006/123/EC of the European Parliament and of the Council of 12 December 2006 on services in the internal market. http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2006:376:0036:0068:en:PDF.

World Wide Web Consortium (2008a). Web Content Accessibility Guidelines (WCAG) 2.0. http://www.w3.org/TR/WCAG/.

World Wide Web Consortium (2008b). XML Signature Syntax and Processing (Second Edition). http://www.w3.org/TR/xmldsig-core/.

Zefferer, T., Tauber, A., Zwattendorfer, B., and Knall, T. (2011). Secure and Reliable Online-Verification of Electronic Signatures in the Digital Age. In Bebo White, P. I. and Santoro, F. M., editors, *Proceedings of the IADIS International Conference WWW/INTERNET 2011*, pages 269 – 276.