SAFELAYER

TrustedX - Signature Platform

Whitepaper

CONTENTS

Introduction

The electronic signature is becoming strategically vital as part of the process in improving efficiency and the effort to do away with the paper format. EU Directive 1999/93/EC establishes a common legal framework for the use of electronic signatures, making them as legally valid as handwritten signatures.

Safelayer security solutions are based on PKI technology and digital certificates. This technology has been widely adopted for a range of applications, including for implementing recognized and/or advanced electronic signature. PKI technology is also used to provide certificate-based authentication methods, which are characterized by the high level of trust that they offer.

Safelayer's electronic signature solutions are noted for their flexibility in integrating electronic signature and authentication mechanisms in corporate processes and for their design for service-oriented architectures. They provide different levels of modularity and scalability, supporting different configurations according to the scenario that they are used in:

- **Corporate environments.** Configuration for the quick deployment of the electronic signature in business environments. Employees can directly sign from their desktops and the corporate applications by simply writing files to network folders.
- **Trust service providers.** Configuration for a range of different environments and for providing e-signature related trust services. Offers great integration flexibility via SOAP (OASIS DSS and WS-Security), RESTful and Java APIs Web services.

TrustedX is a Safelayer product that provides a complete set of security mechanisms that can be accessed as services or via specific integration modules in certain applications and environments. At the system level, TrustedX provides a **platform of centrally-managed PKI security services** and a system for auditing the electronic signature and authentication mechanisms used by the applications.

The most important benefits of TrustedX include:

- **Proven interoperability and support for the standards**

Wide support for the e-signature standards: CMS, PKCS #7, CAdES, S/MIME, XML-DSig, XAdES, PDF Signature and PAdES.

Safelayer has taken part in all the PlugTests organized by the ETSI (European Telecommunications Standards Institute) since the first edition in 2007.

- **Available in hardware and virtual appliance format**

TrustedX is the first product in its category distributed in physical (for Safelayer-approved hardware) and virtual (for virtual-machine environments) appliance format.

The TrustedX product contains all the software needed for its installation, maintenance and administration.

- **Flexible integration and investment protection**

It comes with specific integration APIs and modules and supports industry-adopted integration mechanisms, including SOAP (using the OASIS DSS and WS-Security standards) and REST.

It includes configurable semantic interpretation functions that are used in the processing and assessing of the trust of information, which means that the applications do not require additional logic or complexity.

- **Scalability in functionality and features**

The platform can be extended with the digital signature custody service and/or services that provide data protection and encryption key management.

New authentication mechanisms can be incorporated using agents and credential validation can be delegated to third-parties with RADIUS or LDAP/AD. It also supports identity federation via SAML.

The appliance format is optimized for high-availability environments and improving features. The system can be extended to improve performance.

- **Third-party certification and references**

TrustedX is the only Common Criteria EAL4+ certified product in its category and Europe's most complete technology solution in the validation tools and services category.

Safelayer is a developer of specialized products with multiple projects to its name and diverse partners both nationally and internationally that endorse the strength of its solutions.

For more information on TrustedX, visit:

- http://www.safelayer.com
- http://labs.safelayer.com

CHAPTER1

The TrustedX Platform

As explained above, TrustedX is a PKI security services platform that includes electronic signature and authentication services. It:

- Allows separating the electronic signature and certificate validation mechanisms from the applications and provides centralized management.
- Supports a complete set of digital signature formats and managing multiple CAs, providing the interoperability and federation of PKI domains.
- Includes advanced time-stamping and digital signature verification functions and the option for storing file signatures as per ETSI standards.
- Capacity for implementing server-based signature models. TrustedX allows centrally managing user and application keys.
- Supports corporate repositories (LDAP/AD) and existing authentication mechanisms (LDAP and RADIUS) as well as SAML federation.
- Can be extended with data protection functions, including data encryption and the centralized management of encryption keys.
- Provides flexibility in integrating applications. It comes with plug-ins for applications and incorporates watched folders and a range of APIs for integrating in different environments.
- Is based entirely on policies (e.g., authentication, authorization, electronic signature policies) and administration roles.
- Has a log and auditing system that is extendable and easy to integrate with SIEM tools.
- Is Common Criteria EAL4+ certified, providing the maximum security guarantee.

TrustedX functions can be grouped as follows:

- **Object and entity management.** This service manages platform entities (e.g., users) and objects. External repositories, such as user LDAP/AD, databases, files and HSMs, can be added for protecting private keys.
- **Authentication and authorization.** Supports authentication mechanisms based on digital certificates and passwords. New authentication mechanisms can be incorporated using agents and credential validation can be delegated to third-parties with RADIUS or LDAP/AD. It also supports identity federation via SAML.
- **Certificate validation.** Includes functions for validating digital certificates and analyzing their fields. Supports OCSP, CRL and customized (e.g., database query, accessing the @firma platform) methods for querying the status of certificates.
- **Generation of electronic signature**. Server-signature service for generating electronic signatures in different standard formats for electronic documents, including email and Web messaging. Formats with multiple electronic signatures and electronic signatures with time-stamps are supported.

- **Electronic signature verification**. Service for verifying signatures in different formats for electronic documents, including email and Web messaging. Formats with multiple electronic signatures, electronic signatures with time-stamps and long-term signatures are supported.
- **Time-stamp generation and verification**. Functions for requesting the generation and verification of time-stamps ond ata using the OASIS DSS protocol.
- **Electronic signature update.** Service for extending the lifetime of electronic signatures by maintaining the cryptographic reliability. Functions for incorporating the certificate chain, data on the status of the digital certificates at the time of signing and a time-stamp are provided.
- **Electronic signature custody.** Optional module for autonomously maintaining the lifetime of electronic signatures by interacting with the non-repudiation service and managing the electronic signature metadata.
- **Data protection.** Entails two services that provide (i) the encryption methods for protecting documents, email and web messaging and (ii) the custody of encryption keys and the control of access to them. These services are optional and are not covered in this document. For more information on them, see Safelayer's "TrustedX - Encryption Key Management" white paper.
- **Auditing and accounting.** Securely and uniformly centralizes log data on access control and the consumption of platform services. The log system supports incorporating specific entries, which facilitates management with third-party tools.

Some of the services listed above are common to all platform configurations. The basic services are the entity and object management service, the authentication and authorization service, and the auditing and accounting service. Other services are incorporated in the platform configuration depending on the scenario of use and usually include:

- **Electronic signature.** In addition to the basic services, this includes the digital certificate validation service, the electronic signature generation and verification service, and the signature update service. The electronic signature custody service can also be incorporated.
- Encryption key management. In addition to the basic services, this includes the digital certificate validation service and the data protection services. This configuration of the TrustedX platform is not described in this document. For more information on this configuration, see Safelayer's "TrustedX - Encryption Key Management" white paper.

Architecture

In a service-oriented architecture (SOA), TrustedX's role is to provide specialized services, in this case, PKI security services. When any part of the business process needs to generate, verify, update or archive a signature or validate its certificate, the operation in question is requested from TrustedX via one of its interfaces.

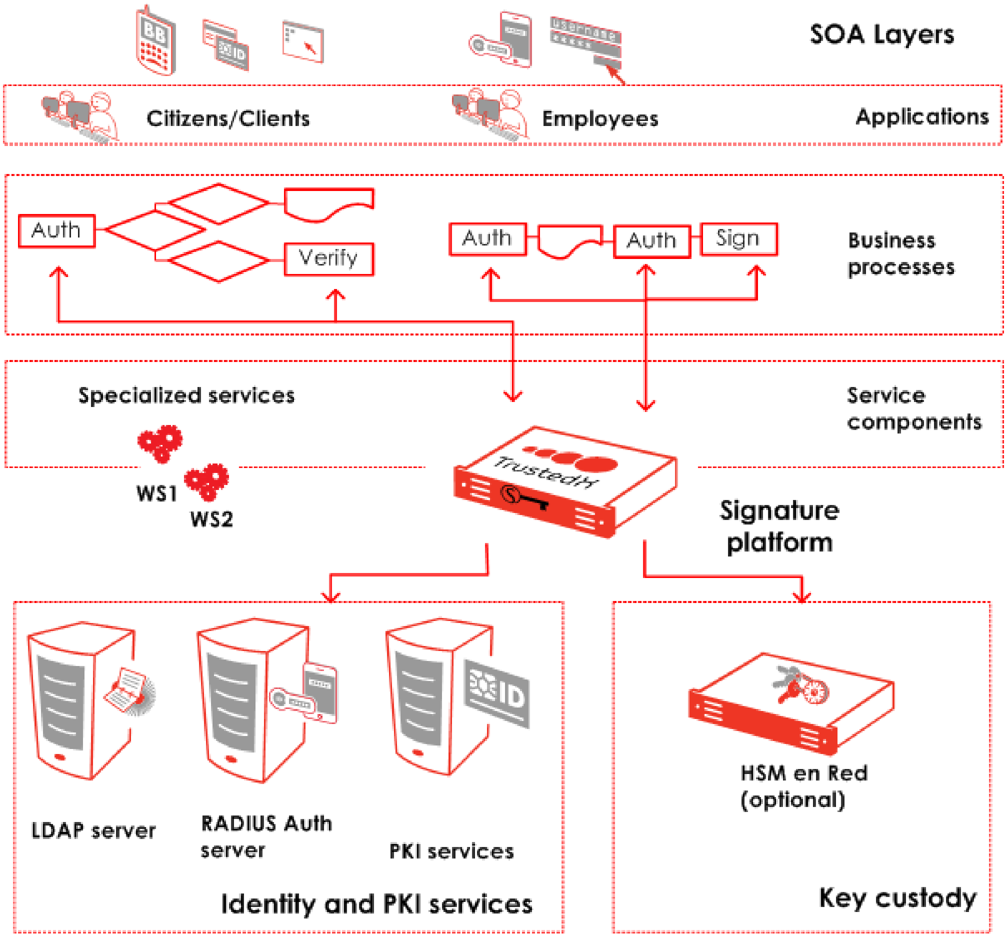


Figure 1-1. TrustedX's position in the SOA architecture.

TrustedX is based on an infrastructure formed by elements including PKI and identity services, repositories, and HSMs, some of which are necessary for its operation. The following are the main elements in this infrastructure:

- **Repositories**

The platform requires an SQL database with JDBC support for managing the system configuration, the entities, the policies and the storage of log data.

The platform can be configured to send log data to a syslog server, which enables integration with SIEM systems.

The platform can be integrated with LDAP/AD servers and databases that contain data on the users of the organization without there being the need, therefore, to perform any type of mass copy or import of user data.

TrustedX can process files stored in network folders (NFS or SMB/CIFS) by other applications to perform electronic signature and encryption operations on them.

Lastly, for the archiving of digital signatures and documents, the platform can be connected to a document management system (DMS/ECM) such as Alfresco or to a Fixed Content Storage system accessible via the API XAM (eXtensible Access Method) such as EMC Centera.

- **PKI and identity services**

In the certificate validation and signature verification processes, TrustedX obtains revocation information on the certificates involved by accessing the trusted VAs using OCSP or by retrieving the required CRLs via HTTP or LDAP.

In addition, the validation connectors can integrate any validation procedure into TrustedX. Only certificates with certification paths that have one of the trusted CAs registered in the platform are considered valid.

Likewise, in signature update processes, both when they are triggered by demand (client request) and when they are performed automatically (archived signature custody), TrustedX obtains the required time-stamps by accessing one or more trusted TSAs via TSP (RFC 3161).

Lastly, TrustedX also supports adding authentication mechanisms using protocols such as RADIUS (e.g., in connection with a token management system that implements one-time use keys) and LDAP (e.g., Microsoft Active Directory).

- **HSM (hardware security modules)**

The user keys can be stored in one or more HSM devices that TrustedX accesses via a PKCS #11 driver.

Integration Modules

TrustedX comes with modules for implementing a range of integration strategies for both corporate and electronic signature service provider scenarios. Depending on the scenario, one or more of the following modules will be used:

- **Virtual card:** Integrates seamlessly in desktop applications. Users transparently access the keys held by TrustedX and only need to know one password.
- **Watched folders:** TrustedX watches the content of designated network folders and executes a series of actions on all the files placed in them. Suitable for users and applications. Signing is as straightforward as copying and pasting to a network folder.
- **Signature services:** A set of APIs for accessing the TrustedX services provides a complete range of integration options for adaptations to different scenarios:
- **SOAP/WS:** Using the OASIS DSS standard as an access protocol for Web services.
- **REST/WS, SOAP/WS:** Using TrustedX's integration gateway, which supports configuring traffic and data processing with an XML pipeline language.
- **Java APIs:** Provides the straightforward integration of the signature services in Java applications.

The platform comes with a **Java Applet** for scenarios that require integrating the electronic signature in Web environments.

The Use Scenarios appendix on page 26 contains use case examples and their architecture that illustrate the different uses of the TrustedX integration modules.

CHAPTER 2

TrustedX Platform Services

This chapter describes the functionality of the TrustedX signature and authentication services.

Entity and Object Management

This service manages the entities (users, applications and trusted third parties) and the objects used by the platform in its operation, including the entities and objects stored in the corporate repositories (e.g., a directory of users) alongside those managed by the platform.

- It is a basic service used by the rest of the TrustedX services (i.e., the authentication and authorization service and the auditing and accounting service). You manage this service in the TrustedX graphical administration console.
- It can be used by other corporate services as its functionality can be made available as a SOAP/WS service.
- It provides uniform access to the data added regardless of the system that manages its physical storage.
- It makes it possible to perform a group of operations (Read, Insert, Update, Delete, Search and Count) using XPath expressions.

The service interacts with a set of external systems. As explained above, these systems include:

- **SQL databases.** TrustedX requires an external database for storing the platform's configurations and policies; the data of the trusted entities, the users and the applications; and the event log.
- **LDAP/AD directories.** TrustedX can add the users of one or more corporate directories by accessing the attributes, digital certificates and authentication mechanisms.
- **Authentication servers.** TrustedX supports adding external authentication servers via RADIUS.
- **Cryptographic devices (HSM).** TrustedX supports HSMs that comply with the PKCS#11 standard. It uses them for protecting cryptographic keys. If there is no HSM, the cryptographic material is protected in an SQL database or in the platform's file system.
- **Document managers (DMS).** TrustedX supports using document managers for storing documents and protecting electronic signatures.
- **Network file system (NFS or SMB/CIFS).** TrustedX can manage files stored in network folders (watched folders) and carry out electronic signature and encryption operations on them.
- **Document archiving systems (FCS)**. TrustedX supports the long-termstorage of electronic documents(Fixed Content Storage) for the archiving of documents and the custody of electronic signatures.
- **Trusted services (CA, VA, TSA and SSA).** TrustedX accesses the services and Web resources of trusted third-parties, e.g., by downloading CRLs and accessing validation and time-stamp authorities and SAML identity providers.

Entity Management

TrustedX makes a distinction between end and trusted entities.

- End entities are users and applications. Each end entity is identified with a distinguished name and has a set of attributes. Its keys are protected by TrustedX.
- Trusted entities provide certification or electronic signature related services. TrustedX can recognize certification (CA), validation (VA) and time-stamping (TSA) authorities and SAML identity providers (SAML IdP).

Using TrustedX's graphical console, you can manage the recognition of the trusted entities used in the system policies. You can define the trust level given to each of these entities.

For end entities, TrustedX provides a management system for internal users and applications while also letting you add user entities stored in corporate LDAP/AD repositories. TrustedX usually manages the application entities itself and makes use of the corporate directory to manage user entities.

You can manage end entities using TrustedX's graphical console or the Web API that interacts with the service via an application integrating it. The system provides:

- Management differentiated by entity type (users and applications). TrustedX entities can be grouped to simplify granting permissions (i.e., by defining the authorization policies) so that when permissions are assigned to one group, they are assigned to all the entities in the group.
- The option to distinguish between static and dynamic groups. Static groups are defined by extension (i.e., by the exhaustive listing of all the members in the group), and the dynamic groups by compression (i.e., by defining the condition for belonging to the group).
- The dynamic groups do not require that their members be registered in TrustedX; they do not need a local identity in the platform. In this case, in the authentication TrustedX assigns identifiers to users that do not have local identities for the carrying out of the ensuing control processes (e.g., logging consumption).
- The option to distinguish between organizational dynamic groups, dynamic groups based in X.509 templates and dynamic groups based in queries on the XML view of the entity and object management service.
    - For organizational groups, the condition for belonging to the group is defined using the attributes of the distinguished name that the entity presents during authentication (Organization, Organizational Unit, Locality, Country and Domain Component).
    - For groups based on X.509 templates, the condition for belonging to the group is defined using the certificate found to be owned by the entity during authentication.
    - For groups based on queries, the condition for belonging to the group is defined as an XPath expression evaluated on the EP (i.e., the XML view of all the information registered in platform) that can be defined in terms of the distinguished name accredited to the entity during authentication.
- The capability to define groups formed by groups of end entities and to manage them as roles so as to simplify the granting of permissions (i.e., the defining of authorization policies). So, by assigning a role to a group (i.e., declaring that a group belongs to a group or groups), all the permissions of the role (of the group of groups) are assigned to all the entities in the group. In this way, TrustedX supports implementing role based access control, for authorizing the consumption of its services and those of other applications.

Authentication and Authorization

This service provides the authentication and authorization infrastructure used by TrustedX to control access to the other services. The authentication and authorization service can also be used from any application that integrates the service using the SOAP/WS interface provided.

This service is based on generating, providing and validating SAML assertions (of the authentication, authorization and attributes types). It provides a secure token system for the consumption of services and supports federation with other SAML authorities. It works as follows:

1. A SAML authentication assertion is generated for the entities that use one of the authentication mechanisms recognized by the platform and present the appropriate credentials (e.g., user name and password).
2. The entity can include the assertion in the consumption request messages that it sends to either a TrustedX service or an external service that trusts TrustedX.
3. The services can authenticate the entity successively without having to request the entity's credentials again (single sign-on).
4. TrustedX supports federation, which means validating assertions generated by third parties is supported. To do this, all that is required is that the SAML authority in question is registered in TrustedX.

TrustedX supports authentication mechanisms based on digital certificates and passwords. In addition, new mechanisms can be added with agents or by delegating the credential validation to third-parties via RADIUS or LDAP/AD. There are three scenarios:

- **Internal authentication mechanism:** Credential validation is performed by TrustedX's authentication and authorization services (e.g., username and password, client certificates received in TLS/SSL connections established directly with TrustedX, and digital signatures).
- **External authentication mechanisms:** The authentication and authorization service receives unvalidated credentials and delegates their validation to an external authentication validator (RADIUS, LDAP, Active Directory). An example of this case is authentication in TrustedX through the validation of one-time passwords (OTP) by accessing a RADIUS authentication server.
- **External authentication agent:** Credentials are validated by an external authentication agent. TrustedX recognizes the agent, which provides the client's identity. This allows using the authentication functions of the applications through which the signature services are requested.

In terms of the authorization, the service allows or denies access to any TrustedX service requested (signature, encryption, etc.). So, the service constitutes the policy decision point and the policy enforcement point of the authorization control that TrustedX implements to protect access to all its services. The service also provides data on the resource access rights that can be used as the policy decision point by any application that wants to centrally base its authorization decisions (i.e., its policy enforcement point) in auditable and manageable policies.

Lastly, the authentication and authorization service also provides an information service on entity attributes that other applications can use to implement Attribute Based Access Control as occurs in some identity federation cases (e.g., a federated system wants to know the attribute for the role of the user in an organization; not the local identity of the user).

Signature Generation

The signature generation service signs data and documents centrally on a server. OASIS DSS is the access protocol used for the service. For request and response messages, this service's interface complies with the OASIS Digital Signature Service (DSS) specification. OASIS deliberately defined the messages of this protocol openly, which means that the final profiling of the structure has to be done in each scenario. For the TrustedX signature generation service, the service access messages are defined according to the type of signature generated:

- **CMS/PKCS#7 profile:** for generating PKCS#7, CMS and CAdES signatures, either from the data to be signed or its hash. Single (data signature) and multiple (signature for already signed data) signatures are supported. In terms of multiple signatures, both sequential (signature of a signature) and parallel type signatures are supported. Detached (external) and attached (enveloping) signatures are also supported. Using the data hash to generate the signature is also supported.
- **XML-DSig/XAdES profile:** for generating XML-DSig and XAdES signatures, either from the data to be signed or its hash. Generating enveloping and enveloped signatures is supported, both in the same document as the signed data. The generation of detached signatures is also supported. After obtaining these signatures, the client can insert them in the same document as the signed data or a different one.
- **PDF/PAdES profile:** for signing PDF documents as per the signature format defined by Adobe in [PDFRef]. It also supports signing PDF documents as per the profiles defined in parts 2 and 3 of PAdES (PAdES Basic, PAdES-BES and PAdES-EPES). Future releases of TrustedX will also support parts 4 and 5 of PAdES (PAdES Long-Term and PAdES for XML content).
- **S/MIME profile:** for signing emails so that the resulting messages have the S/MIME v2 (RFC 2361) or S/MIME v3 (RFC 2633) format. The enveloping (Content-Type: application/pkcs7-mime) and detached (Content-Type: multipart/signed) signatures are supported.
- **WS-Security profile:** for signing SOAP messages (i.e., the body element) so that the resulting signature (an XML-DSig signature) is added to its header, as defined in [WSS]. Therefore, this profile can be used to protect the authenticity and integrity of the SOAP messages of any Web service.
- **Raw (PKCS #1) profile:** for generating PKCS #1 signatures.

Key Management

The keys used by entities to generate their signatures are stored in keystores protected by TrustedX. The keys can be protected with an HSM, which you can manage in two ways:

- **In the graphical administration console's GUI.** In this case, this management can only be carried out by users in the Security Officers group.
- **By accessing the key management service.** In this case, Security Officers and the subjects of the keystores can manage the keystores. However, only an end entity can manage its own keystore (via the key management service).

The key management service is based on the W3C [XKMS] specification, specifically, in the part of the specification in which the protocol for registering information on the public keys is defined (XML Key Registration Service Specification).

By integrating the virtual card in corporate registration systems, keys and certificates can be also provided to end entities and stored in their keystores. The same operation and procedures for putting the keys and certificates on the physical cards apply.

Signature Verification

The signature verification service verifies document and data signatures and also validates certificates, centrally, on the server. For request and response messages, the interface of this service complies with the OASIS Digital Signature Service (DSS) specification. OASIS deliberately defined these messages openly, which means that the final profiling of the structure has to be done in each scenario. For the TrustedX signature verification service, the service access messages are defined according to the type of signature to be generated:

- **CMS/PKCS#7 profile:** for verifying signatures in PKCS#7, CMS and CAdES formats (BES, EPES, ES-T, ES-C, ES-X Long and ES-A formats). The verification of single signatures (with only one signer) and multiple signatures (with several signatures in parallel) is supported. Detached (external) and attached (enveloping) signatures are also supported.
- **XML-DSig/XAdES profile:** for verifying signatures in XML-DSig and XAdES formats (BES, EPES, ES-T, ES-C, ES-X Long and ES-A formats). The verification of enveloping, enveloped and detached signatures is supported.
- **PDF/PAdES profile:** for verifying the signatures contained in a PDF document, both ISO 32000-1 signatures and PAdES signatures of any of the profiles defined in ETSI TS 102 778-2 and ETSI TS 102 778-3 (PAdES Basic, PAdES-BES and PAdES-EPES). Note that there is no specific profile for verifying PAdES signatures; the PDF-Signature profile is used to verify any signature on a PDF document.
- **S/MIME profile:** for verifying the signature of an email message in S/MIME v2 (RFC 2361) or S/MIME v3 (RFC 2633) format, for both enveloping (Content-Type: application/pkcs7-mime) and detached (Content-Type: multipart/signed) signatures.
- **Certificate profile:** for verifying the signature of a certificate and checking its validity status.
- **WS-Security profile:** for verifying (as per the criteria defined in [WSS] the XML-DSig signatures in the header of a SOAP message.

Certificate Validation

Verifying a signature always requires validating the signer certificate and the certificates of its certification path. In addition, TrustedX regards the validation of a certificate as a 'profile' or a certain type of signature verification. This means a certificate may be validated as a result of the verification of a signature or because a user requests it. In any case, the validation might require downloading and querying CRLs and accessing one or more VAs to determine if the certificate in question has been revoked.

The signature verification service supports checking the status of a certificate using any combination of the above mechanisms. That is, both by querying the CRLs obtained (downloaded) by HTTP, LDAP or that are stored in the file system and by accessing one or more VAs using OCSP protocol. Customized validation connectors can also be used to integrate in the platform any additional (non-standard) mechanism that supports checking the revocation status of the certificates.

Semantic Interpretation of Signatures

One feature of the signature verification service is that as well as specifying that a signature or a certificate is valid or not, it offers richer semantics. A signature or certificate that is found to be valid is assigned the level of assurance that its validity deserves (low, medium, high, very high, etc.) and a trust label that places this level of assurance in a given context (corporate, finance, public sector). The service can also include any data that the platform has on the signer and the trust entities that were used in the verification, meaning that it includes more or less data depending on the verification policy used. In short, TrustedX supports configuring what signer and trusted entities details are included in the verification service responses.

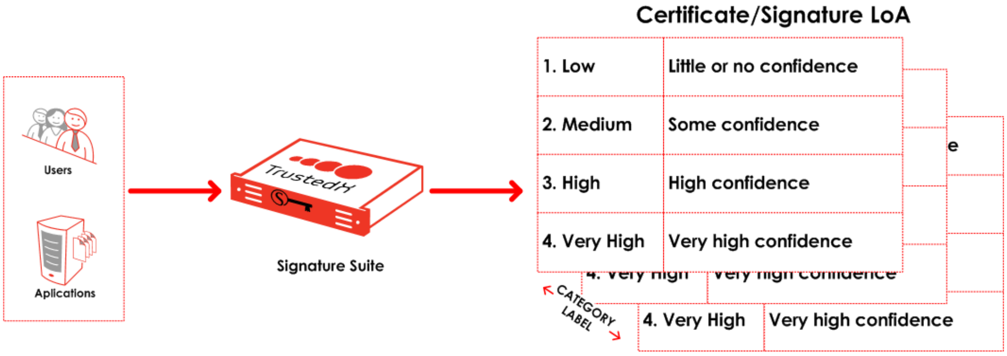


Figure 2-1. Quantification and qualification of the trust in a signature or certificate.

Non-repudiation

The assurances provided by the signature must be preserved and not be left to become invalid when a signer's certificate expires or is revoked. It must be possible to repeat a signature verification in the future, which requires compiling the evidences for being able to repeat this verification (signature time-stamp, signer's certificate, certification chain, CRLs, OCSP responses) and also preserving their cryptographic strength (and that of the original signature) by obtaining successive time-stamps on all of the data.

The signature update (DR) service adds non-repudiation evidences (time-stamps, certificates, CRLs, OCSP responses) to the signatures. This update is performed straight after the signatures have been verified. This way, the service only updates valid signatures. This is achieved by adding the evidences that are required at different times to the signatures to counter the effects of the threats that may invalidate them (e.g., the signer certificate expiring).

Consequently, the signature update service does the following, depending on how and when it is invoked:

- **Adds a time-stamp (ES-T).** This proves the existence of the signature at the moment that the (ES-T) stamp is generated, which serves as a base for being able to preserve the signature over time.
- **Adds all the certificates, CRLs and OCSP responses that were used to verify the signature and a time-stamp that proves the existence of all this validation data at the moment the stamp is generated (ES-X Long, ES-A).** This guarantees being able to repeat the signature verification in the future (e.g., in an arbitration process). The signature is valid for as long as the new time-stamp is. This means that the signature remains valid (and that its verification can be repeated) after the signer certificate expires or is revoked and also after any of the certificates in the certification path expire or are revoked.
- **Adds an archive time-stamp that extends the validity of the signature (ES-A).** This keeps the signature valid while the new time-stamp is valid. So, the signature is still valid after the previous time-stamp becomes invalid or any of the cryptographic algorithms used in the building of the signatures are broken (e.g., the hash function the signer used to sign the data is successfully cryptanalyzed).

The signature update service updates ETSI CAdES and XAdES signatures (BES, EPES, ES-T, ES-C, ES-X Long and ES-A formats). In future releases, TrustedX will support the update of PAdES signatures (ETSI TS 102 778) so that, using the PAdES-LTV profile, the signatures of a PDF can conserve their validity over very long periods.

Time-stamps

TrustedX uses the profiles of the signature generation and verification services that implement the OASIS DSS protocol to provide the option for requesting the generation of time-stamps on data and its subsequent verification. To issue time-stamps, the system requires the availability of a trusted TSA (RFC 3161) service. Verifying time-stamps also entails checking the trust in the TSA issuer.

Signature Custody

The signature custody service implements a document and signature archiving service. Archiving is the storing of documents and signatures in repositories along with their metadata over long periods of time. The service also supports retrieving archived documents, signatures and metadata; verifying archived signatures; and updating them (on request) with non-repudiation evidences.

Preservation of Evidence Value and Signature Cryptographic Strength

In the specific case of the signatures, as well as supporting archiving, retrieving, verifying and updating on request, the signature custody service carries out an active custody that, via automatic updates, supports maintaining signature validity over an indefinite period of time (long-term signatures). The signature custody service is capable of maintaining the validity of the signatures it stores beyond the expiry of the signer certificate. It can even do this when the signer certificate is revoked after the signature was archived (i.e., the system protects against attempts to repudiate archived signatures by malicious signers).

Multiple Document and Signature Format Support

Any type of document can be stored in the archive (PDF, Word, XML, HTML, image, videos, etc.). For signatures, the custody service supports all the formats that the CAdES and XAdES specifications define based on the CMS and XML-DSig standards (i.e., BES, EPES, ES-T, ES-C, ES-X Long and ES-A formats). In addition, future versions will support updating PDF signatures as per the PAdES specification.

Multiple System Support for Archiving

With respect to the repositories that can be used, the signature custody service supports archiving documents and signatures in document management systems that are accessible via HTTP/WebDAV (e.g., Alfresco), and in Fixed Content Storage systems accessible via XAM (eXtensible Access Method) such as EMC Centera, which provide additional integrity mechanisms such as retention policies. In addition, the policies of the service can be configured so that, with each of them, the archiving is done in a different repository and/or with different properties

Auditing and Accounting

All TrustedX services send the log records to the same (extendable) set of repositories, normally databases and/or syslog servers. The connection with these repositories and the granularity of the events logged in them can be managed with the shell.

The platform comes with appenders or modules for the sending of log records. For example, it provides one for saving log records to any JDBC-accessible database and another for sending them to a syslog server. In addition, customized appenders can be installed for sending log data to repositories that use specific communication protocols or that require that log records be sent in a certain format.

Log records can be labeled with an accounting policy identifier that facilitates their subsequent use. In fact, the accounting policies allow keeping a track of the number of authentications (sessions started) and authorizations (services consumed) performed by users. These figures are important for statistical monitoring and billing.

Log records stored on database can be browsed using the graphical administration console or via the platform's information integration service.

CAPÍTULO 3

Integrating Applications

One of the main advantages of the TrustedX platform is its ease of integration owing to its great flexibility and its high degree of standardization.

TrustedX functions can be accessed:

- As Web services.
- From watched folders.
- From desktop applications using the virtual card.
- From a document or content manager using a specific plug-in.

TrustedX also has an integration gateway that, through the use of pipelines, allows customizing the Web interface for accessing the platform's signature functions.

Integration by Programming SOAP, REST and HTTP Interface Access

Both new and existing applications can integrate the access to the signature functions through straightforward development in which only the access to one or more Web services needs to be programmed and in which the developer does not require a knowledge of signature formats, certificates, CRLs, CAs, VAs, TSAs, keys management, etc. This access can consist of exchanging SOAP messages with TrustedX's native Web services interface, which is described using WSDL. Thus, the programming can be done using tools (Axis, .NET, JAX-WS, etc.) that generate access libraries (Java, .NET) to Web services automatically using the WSDL definition. For Java applications, as an alternative, the access library for the TrustedX services that comes with the platform can be used.

Alternatively, pipelines can be defined in the integration gateway to implement a customized interface (SOAP, REST or simply HTTP) for accessing TrustedX's services. To complete the integration, the accesses to the Web resources defined in the integration gateway need to be programmed into the client application, for which the appropriate client APIs can be used (JAX-WS, JAX-RS, etc.). This approach includes programming in TrustedX and, therefore, concentrates even further the complexity on the server-side and makes integrating signature services in new and existing applications more straightforward. Under this approach, for example, validating a certificate becomes a simple matter of sending the certificate to a URL of a HTTP message that contains the certificate, encoded in base64, in the Body of this message.

Integration Java API

TrustedX's integration Java API is developed on Axis and provides access to the signature services using the different classes implemented. These classes are for the requests and responses that can be sent to the different services.

This API supports implementing client applications without having to deal with Axis's complexity while allowing access to Axis structures that might be necessary for advanced use. Thus, it provides very simple Java applications with very little code and resolves the complexity involved in writing SOAP requests.

Another major benefit of TrustedX's integration Java API is that it facilitates handling large documents and signatures.

Java Applet for Web Environments Safelayer's OpenSignX applet, which supports local signing (i.e., not centralized) in the client, is a complement to TrustedX's server signing functionality. It is for scenarios in which the signature is generated with a key that is stored on a cryptographic card (a smart card) or a similar device that the subject (the user) holds, or in a software keystore managed by the operating system of the workstation or laptop.

OpenSignX can be used to sign both Web forms and files in the local file system.

Integration via Reading and Writing Files in Watched Folders

For many applications, the only integration mechanism possible is that of writing and reading files in folders. Thus, the only way, for example, to add an electronic signature to data generated by an application might be for the application to write the data to a file and then copy the file to a network folder so that a signature application can read it, generate the data signature, write the signature and data to another file, and copy this file to another network folder. TrustedX supports this integration mechanism with its watched folder functionality. TrustedX uses this functionality to check the content of watched folders periodically and process the files it finds in them by executing pipelines of the integration gateway (SmartGateway). The result of each pipeline execution is a file in a network folder from which the client application can collect the file.

Integration via Virtual Card

Applications are often designed to access signature and authentication functions via predefined cryptographic interfaces. Browsers (Firefox, Internet Explorer), for example, usually access these functions using the PKCS #11 or Microsoft CAPI interface.

These interfaces isolate the applications from the specific characteristics of the cryptographic devices. This means that you avoid having to make changes to the applications if the cryptographic (token) provider is changed. Where the cryptographic provider or token implements the same interface (PKCS #11 or Microsoft CAPI), it does not matter whether it is a provider implemented exclusively on software or hardware (e.g., a smart card) or from one manufacturer or another.

The way that these applications use the cryptographic mechanisms generally requires a distributed management of the user keys as the keys are usually stored in the system on which the application is run (token software) or on a cryptographic card held by the user (token hardware). In neither case can the use of these mechanisms be governed with corporate-level policies; nor can a centralized log on key usage be kept that can be consulted at any time.

TrustedX's virtual card is a library that implements the functions of the PKCS #11 and Microsoft CAPI interfaces by accessing the TrustedX services. In other words, it is a library that provides access to a remote cryptographic token. Having a PKCS #11 or Microsoft CAPI interface means that it can be immediately integrated with all applications that are compatible with this interface. When the application invokes (on behalf of a user) one of the virtual card's functions (e.g., signature), the card executes a code requesting that TrustedX performs the operation. This means that the storage and use of user keys can be centralized on an HSM, the use of the cryptographic mechanisms can be made subject to corporate-level policies, and a single log that collects the data on each of these uses can be compiled.

The virtual card provides the same functionality as a physical card (a smart card) but without really being a card. This amounts to a significant reduction in costs as the organization does not have to give a physical card to all the users and supply them with a card reader. However, it behaves as a physical card does and, therefore, gives users a secure way of using their keys as these keys are stored centrally in TrustedX and, as a result, are isolated from the operating system in which they are used. Furthermore, key access control, which for physical cards is performed by verifying a static PIN, is done using TrustedX's authentication and authorization service for the virtual card, which supports the use of one-time PINs (OTPs) or PINs that combine a static part (something you know) with a dynamic part generated using a given device (something you have).

The virtual card, both for devices accessible via PKCS #11 and Microsoft CAPI, can be integrated into the corporate face-to-face registration authorities (KeyOne LRA). Thus, after the user’s credentials have been entered (e.g., name and password that the user registered in TrustedX with), the registration system can request that TrustedX generates the user keys in question. Once the corresponding certificate is issued, the user can use the keys in any system in which the virtual card is installed as if the keys were on a physical card.

CHAPTER4

Administering the TrustedX Platform

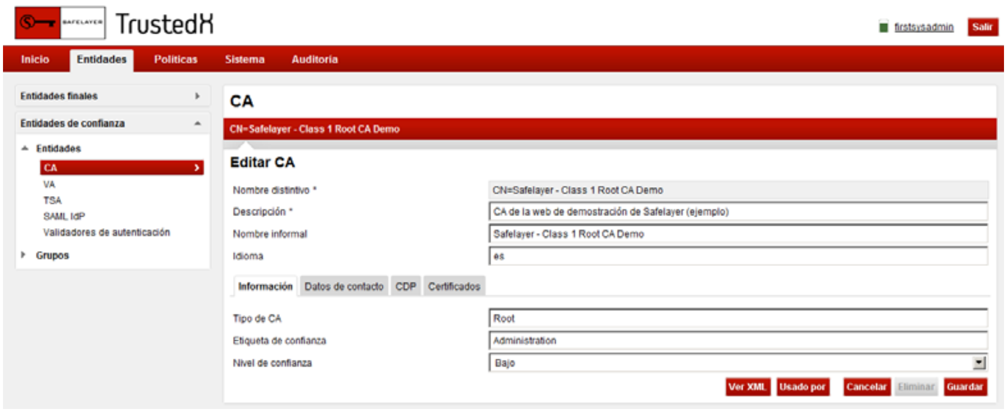
TrustedX administration comprises two clearly-differentiated domains. On the one hand, there is the administration of the system configuration and the access to the log records generated by the services. On the other hand, you have the administration of the 'appliance', i.e., of the execution platform on which TrustedX operates:

- The first type of administration is done using a Web application that forms part of the system. It has a graphical interface in which the TrustedX configuration can be managed and the log records can be browsed.
- The second type involves using an application called the command-line administration console (or the shell), which is accessed via the physical terminal of the appliance or from a remote terminal connected via SSH. See *Command Shell*, page 22.

Graphical Administration Console

The graphical administration console is a Web application for administering and accessing all the information TrustedX handles using a browser. The administration functions this application implements include:

- **Management of end-entities:** for registering users, applications and services as end entities and managing their data. It also supports defining groups of end entities.
- **Management of trusted entities:** for managing the certification authorities, validation authorities and time-stamping authorities (*Figure 5-1*) that the platform trusts.



CA

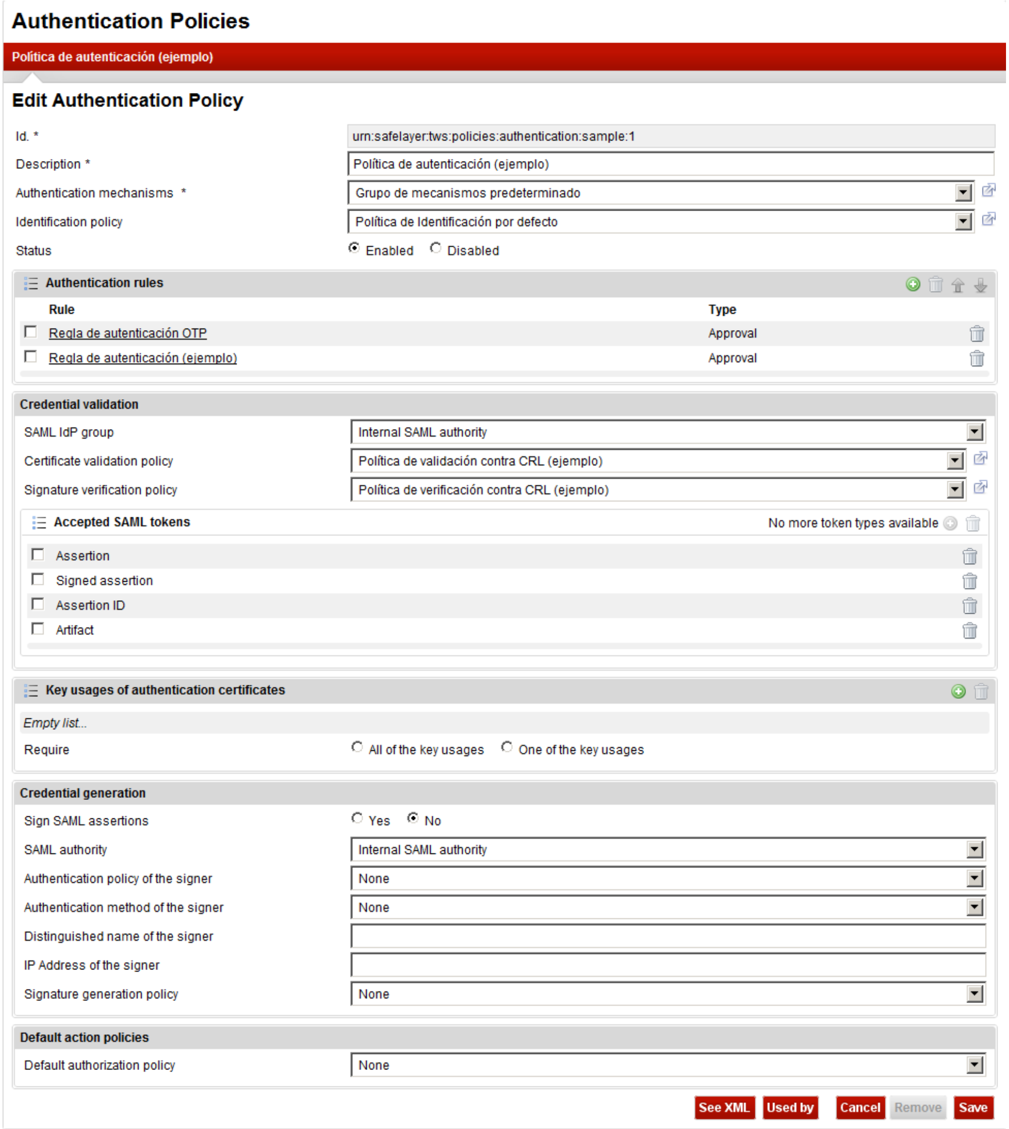
CN=Safelayer - Class 1 Root CA Demo

Editar CA

Nombre distintivo *	CN=Safelayer - Class 1 Root CA Demo
Descripción *	CA de la web de demostración de Safelayer (ejemplo)
Nombre informal	Safelayer - Class 1 Root CA Demo
Idioma	es
Tipo de CA	Root
Etiqueta de confianza	Administration
Nivel de confianza	Bajo

Figure 4-1. Management of trusted entities.

- **Management of authentication and authorization policies:** for defining the authentication and authorization policies for controlling access to the end entities of the TrustedX services (Figure 5-2.)



Authentication Policies

Política de autenticación (ejemplo)

Edit Authentication Policy

Id. *	urn:safelayer:tws:policies:authentication:sample:1
Description *	Política de autenticación (ejemplo)
Authentication mechanisms *	Grupo de mecanismos predeterminado
Identification policy	Política de Identificación por defecto
Status	● Enabled ○ Disabled

Authentication rules

	Rule	Type
□	Regla de autenticación OTP	Approval
□	Regla de autenticación (ejemplo)	Approval

Credential validation

SAML IdP group	Internal SAML authority
Certificate validation policy	Política de validación contra CRL (ejemplo)
Signature verification policy	Política de verificación contra CRL (ejemplo)

Accepted SAML tokens No more token types available

- □ Assertion
- □ Signed assertion
- □ Assertion ID
- □ Artifact

Key usages of authentication certificates

Empty list...

Require ○ All of the key usages ○ One of the key usages

Credential generation

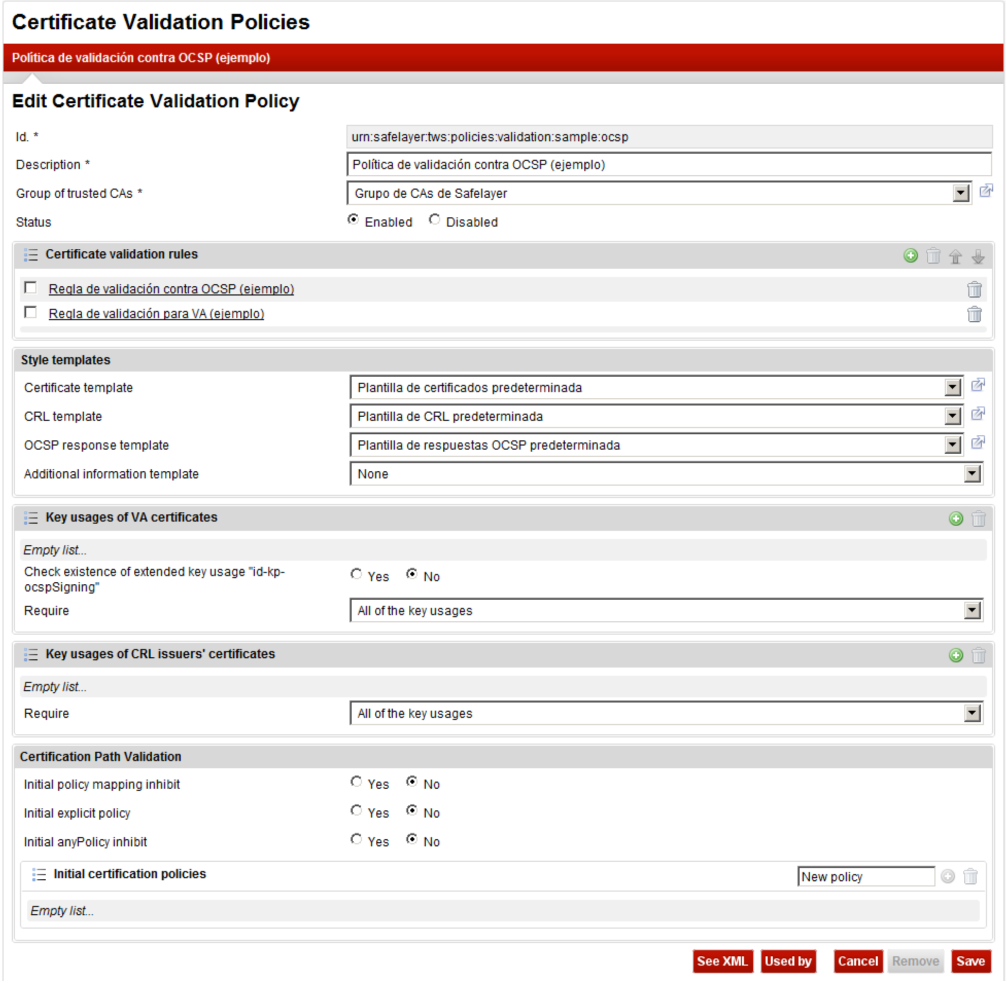
Sign SAML assertions	○ Yes ● No
SAML authority	Internal SAML authority
Authentication policy of the signer	None
Authentication method of the signer	None
Distinguished name of the signer	
IP Address of the signer	
Signature generation policy	None

Default action policies

Default authorization policy	None

Figure 4-2. Management of authentication and authorization policies.

- **Management of digital signature generation policies:** for defining and making changes to the policies applied for generating electronic signatures.
- **Management of digital signature verification and certificate validation policies:** for defining and making changes to the policies applied for verifying digital signatures and validating public key certificates (*Figure 5-3*).



Certificate Validation Policies

Política de validación contra OCSP (ejemplo)

Edit Certificate Validation Policy

Id. *	urn:safelayer:tws:policies:validation:sample:ocsp
Description *	Política de validación contra OCSP (ejemplo)
Group of trusted CAs *	Grupo de CAs de Safelayer
Status	● Enabled ○ Disabled

Certificate validation rules

□ Regla de validación contra OCSP (ejemplo)

□ Regla de validación para VA (ejemplo)

Style templates

Certificate template	Plantilla de certificados predeterminada
CRL template	Plantilla de CRL predeterminada
OCSP response template	Plantilla de respuestas OCSP predeterminada
Additional information template	None

Key usages of VA certificates

Empty list...

Check existence of extended key usage "id-kp-ocspSigning"	○ Yes ● No
Require	All of the key usages

Key usages of CRL issuers' certificates

Empty list...

Require	All of the key usages

Certification Path Validation

Initial policy mapping inhibit	○ Yes ● No
Initial explicit policy	○ Yes ● No
Initial anyPolicy inhibit	○ Yes ● No

Initial certification policies New policy

Empty list...

Figure 4-3. Management of the certificate validation policies.

- **Management of the configuration of the services:** for defining the configuration of the services in the platform.
- **Management of the configuration of the connections with the repositories:** for defining the configuration of the connections for accessing the different repositories (databases, LDAP services) that the system uses.
- **Management of the configuration of the access to HSM devices:** for defining the configuration used for accessing the HSM devices used by the platform.
- **Log browsing and auditing:** for browsing the events generated by all the service components of the platform.

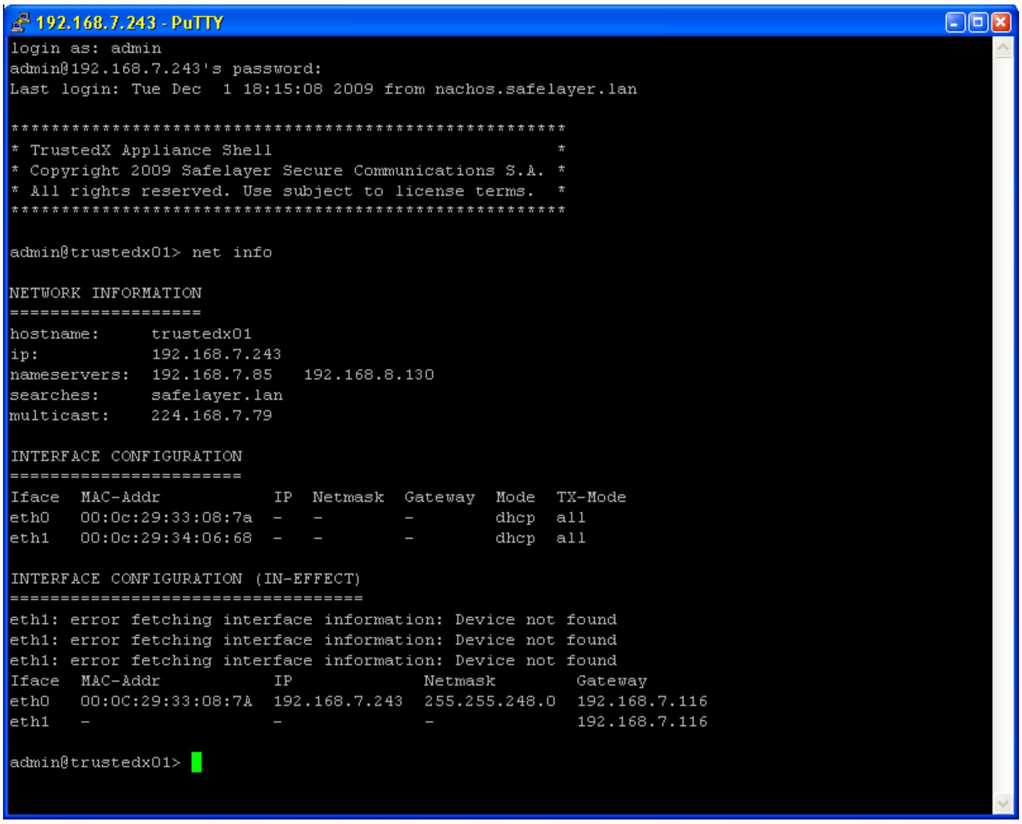
Command Shell

This application, which, as its name suggests, has a command-line user interface, is for administering the system on which TrustedX is run (Figure 5-4). With it you can:

- Install the license file in the appliance's file system.
- Configure the appliance's network interface.

- Install the drivers and define the client configuration so TrustedX can access the elements that make up its operating environment (databases, HSM devices, NTP servers, etc.):

This application's commands are hierarchically organized in a multi-level structure. All the commands have a very similar syntax that can be checked using the help command. Pressing the 'tab' key autocompletes commands and displays their options.



```
login as: admin
admin@192.168.7.243's password:
Last login: Tue Dec  1 18:15:08 2009 from nachos.safelayer.lan

********************************************************
* TrustedX Appliance Shell                             *
* Copyright 2009 Safelayer Secure Communications S.A. *
* All rights reserved. Use subject to license terms.  *
********************************************************

admin@trustedx01> net info

NETWORK INFORMATION
===================
hostname:     trustedx01
ip:           192.168.7.243
nameservers:  192.168.7.85   192.168.8.130
searches:     safelayer.lan
multicast:    224.168.7.79

INTERFACE CONFIGURATION
=======================
Iface  MAC-Addr           IP  Netmask  Gateway  Mode  TX-Mode
eth0   00:0c:29:33:08:7a  -   -        -        dhcp  all
eth1   00:0c:29:34:06:68  -   -        -        dhcp  all

INTERFACE CONFIGURATION (IN-EFFECT)
===================================
eth1: error fetching interface information: Device not found
eth1: error fetching interface information: Device not found
eth1: error fetching interface information: Device not found
Iface  MAC-Addr           IP             Netmask        Gateway
eth0   00:0C:29:33:08:7A  192.168.7.243  255.255.248.0  192.168.7.116
eth1   -                  -              -              192.168.7.116

admin@trustedx01>
```

Figure 4-4. Command-line administration console.

High Availability

TrustedX's services can be deployed in high availability so that they are always accessible. This deployment's architecture is illustrated in Figure 5-5.

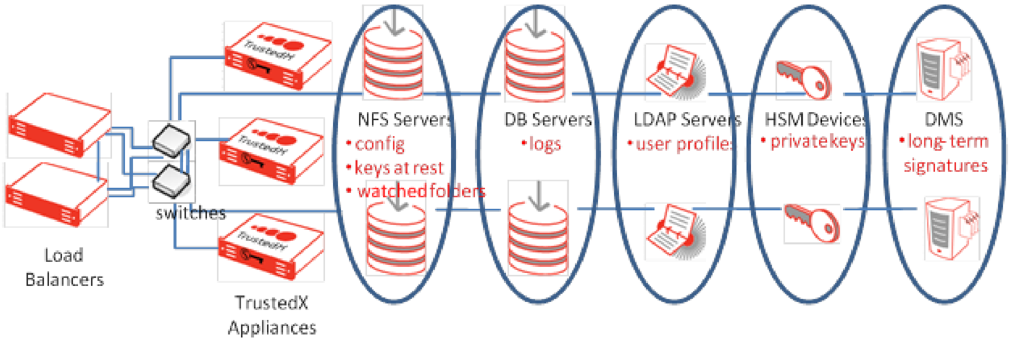


Figure 4-5. TrustedX high-availability deployment.

This architecture has a cluster formed by two or more TrustedX appliances to which a load balancer, also in high availability (e.g., active/passive configuration), distributes the requests received from clients. There is only one copy of the TrustedX configuration, and it is stored on a network file system (e.g., SMB/CIFS, NFS). Each cluster appliance has the network folders containing this configuration mounted on its local file system. This means that the configuration can be managed by accessing the graphical administration console of any of the appliances without it being necessary to subsequently perform any type of synchronization. All the systems and resources (log databases, LDAP servers, HSM devices, DMS systems, etc.) that the TrustedX services access must be in high availability.

Monitoring and Auditing

Monitoring in TrustedX, which is carried out using a SNMP agent, aims to assure the correct operation of the platform (Figure 5-6). The organization's external monitoring product receives traps from this agent in real-time when an exceptional situation occurs. The external monitoring product also sends requests to the SNMP agent (probing) to detect failures during periods of apparent inactivity. With the data obtained via the monitoring product, reports can be created for the organization's IT systems department.

Auditing is performed by sending all the activity that occurs in the platform to an external system (e.g., Splunk) using syslog (Figure 5-6) to generate business and compliance reports.

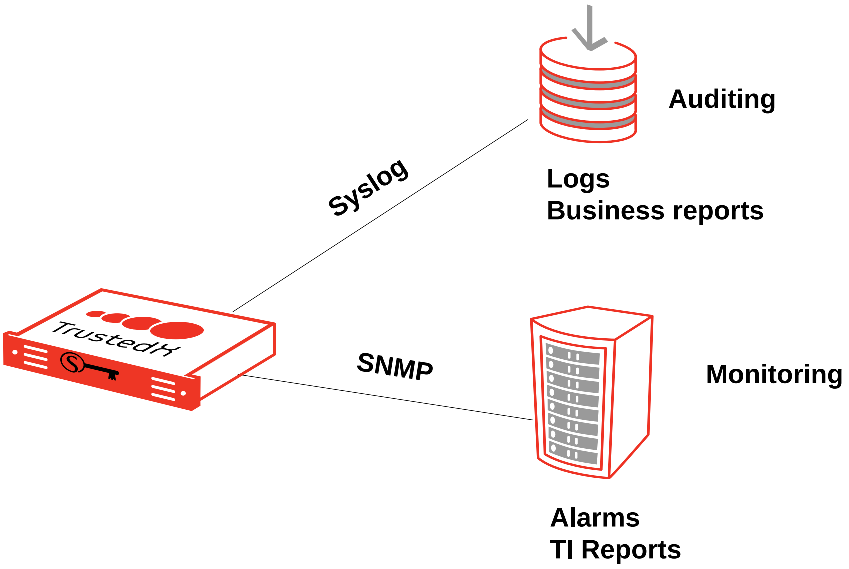
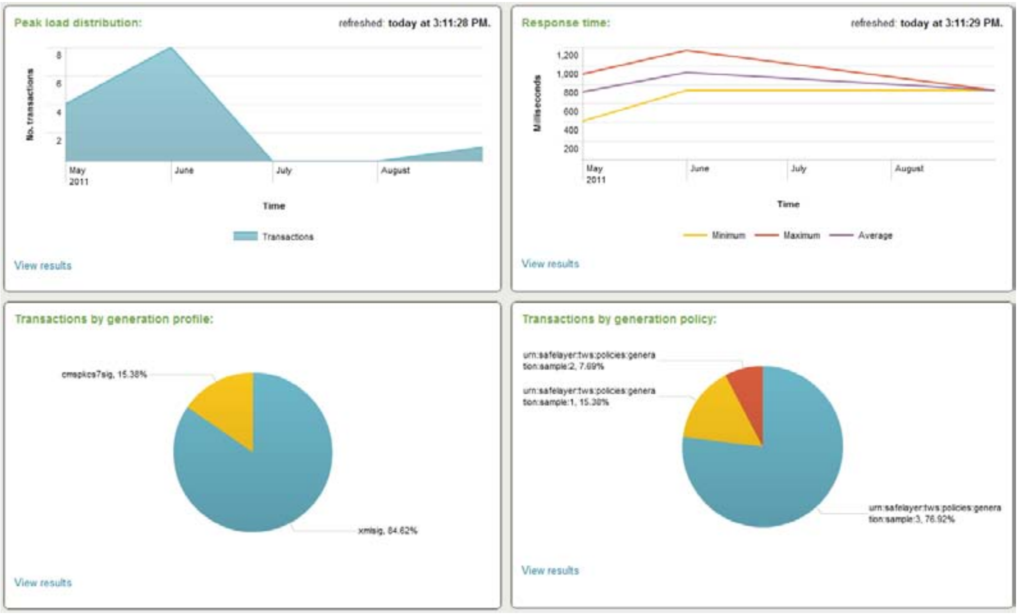


Figure 4-6.TrustedX monitoring and auditing.

Below is an example of a graphical report prepared with Splunk that shows signature generation statistics for TrustedX.



Peak load distribution:

refreshed: **today at 3:11:28 PM.**

Response time:

refreshed: **today at 3:11:29 PM.**

Transactions by generation profile:

Transactions by generation policy:

urn:safelayer:tws:policies:generation:sample:2, 7.69%
urn:safelayer:tws:policies:generation:sample:1, 15.38%
urn:safelayer:tws:policies:generation:sample:3, 76.92%
View results

Figure 4-7. Signature generation service reports.

APPENDIX A

Use Scenarios

This appendix outlines the following three use scenarios for the TrustedX platform in which different platform integration modules are used:

- User signature on server
- Corporate signature on serve
- Signing Web Forms with Card

User Signature on Server

The keys are protected in a centralized repository (optionally, in an HSM) so that employees can remotely use them when they have to perform signature operations. This scenario arises when the organization wants to centrally manage the signature keys of all staff. This allows auditing key use and avoiding that keys are copied in user workstations.

The system supports the following integration modes:

- **User with virtual card:** The user can transparently use the keys in desktop applications (e.g., Explorer, Chrome, Acrobat and Office). This mode requires the TrustedX virtual card and signature services modules.
- **Signing in an application:** User applications require the signing of documents or Web forms. In this case, when the application requires the signature, it sends the document to TrustedX, which manages the user's signature. The TrustedX signature services module is required for this mode.

Validation of signer credentials, which is essential for controlling key access, is performed via TrustedX making use of its capability for integrating with corporate repositories (LDAP, Active Directory) and authentication services (RADIUS). When access is performed using the virtual card, the credential is a static or dynamic PIN. When access is performed in the centralized applications, any other authentication mechanism can be used.

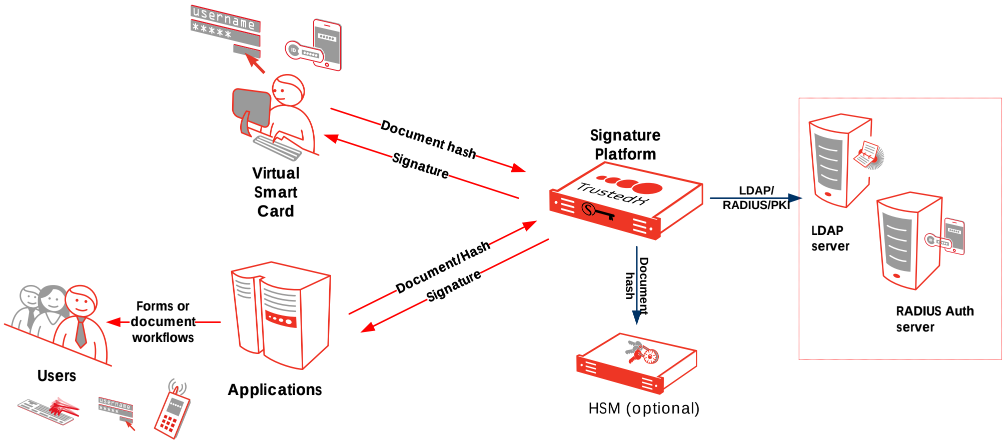


Figure 4-1. User Signature on Server

For this scenario, TrustedX has a signature generation service that, via its graphical console administration and log-record generation, provides effective control and supervision (auditing) of the signing capability of the organization's staff. For example, all requests that do not specify the commitment that the signer wants to acquire with respect to the signed data can be rejected, or the usable signature algorithms can be defined. It can also be established that all signatures must include the policy identifier used to generate it in an attribute signed by this signature.

All applications in which the users have to generate a personal signature must include a mechanism for accessing TrustedX. The applications are relieved of having to manage, as well as all the users' signature keys, this signature function's security configuration (e.g., the applicable policies) and the log records that result from its use. This means that in terms of the signature function, a silo-based deployment is replaced by a centralized SOA-based deployment, with all the advantages in control, coherence and cost saving that this brings.

In the case of the centralized applications (form signatures, document flows with signatures) that users access from multiple terminals that can be of different types, the access to TrustedX needs to be integrated by programming, as it is not normally viable to install the virtual card in all the terminals.

Corporate Signature on Server

This signature is performed using a corporate key, i.e., a key for which the organization is the subject. This key is centrally managed, for example, in a cryptographic hardware module.

This use case is for centrally managing the keys that the organization's applications use for automatically signing on behalf of the organization. This type of key management makes it unnecessary to keep multiple copies of the corporate keys (one per application) and to have to repeat administration procedures for each of them (which can become unmanageable) and facilitates the control (policy enforcement) and auditing of key use.

All applications that have to generate a signature on the organization's behalf must have a mechanism for accessing TrustedX. The applications play no part in managing the signature keys used, the security settings of this signature function (e.g., the applicable policies) or the logging of key use.

TrustedX supports the two following integration modes:

- **Watched folders:** Its low cost and quick set up make watched folder integration ideal for many environments. The TrustedX watched folders module monitors the content of a network folder (NFS or SMB/CIFS), executing a series of actions on the files stored in it. Processed files are put in an outgoing folder, which is also on the network, along with a results report. TrustedX supports defining multiple watched folders and processing pipelines executed for each of them.

- **Signature services:** The signature services module provides a complete set of APIs and Web services that support integrating the mechanisms in applications by using different strategies. This scenario is suitable for different environments that require providing access to all TrustedX's advanced functions, such as SOAP/WS, REST/WS and Java APIs.

This means that in terms of the corporate signature function, a silo-based deployment is also replaced by a centralized service-based deployment, with all the advantages in control, coherence and cost saving that this brings. The programmatic access by the applications to the corporate signature generation function permits automating this function and, as a result, also automating a large part of an organization's data management processes.

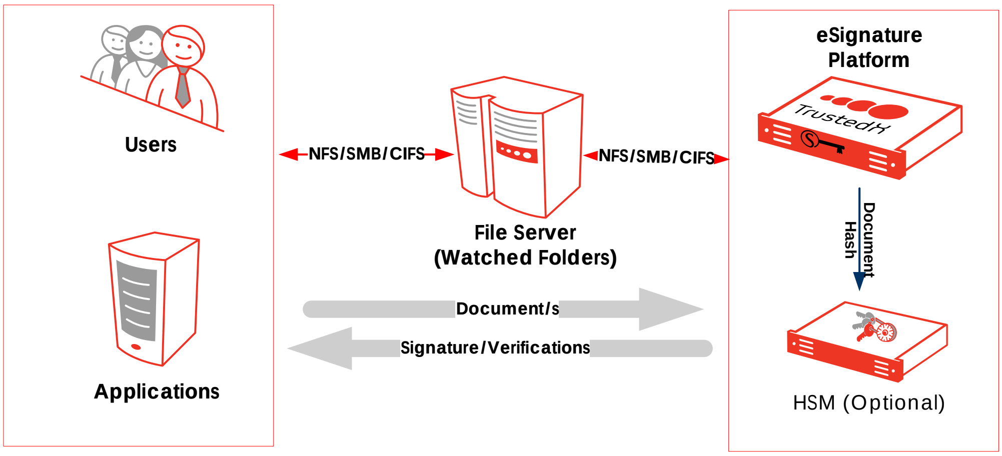


Figure 4-1. Server signature using watched folders.

Signing Web Forms with Card

Sever signing does not allow generating the signature with a key held by the user (e.g., on a cryptographic card, in the browser's keystore). These situations are typical of scenarios in which the organization interacts with external entities using Web forms (e.g., business to business, business to customers, government to citizens).

To cover this need, the TrustedX platform is complemented with a signature applet known as OpenSignX. This applet lets users sign forms and files locally. All they need is a Web browser with the Java plug-in (Figure 5-1). In addition, once the Web server receives the signed form or document, it can access TrustedX's signature update service to add a time-stamp to the signature. Thus, the signature's validity (i.e., its non-reputability) is guaranteed while the time-stamp is valid.

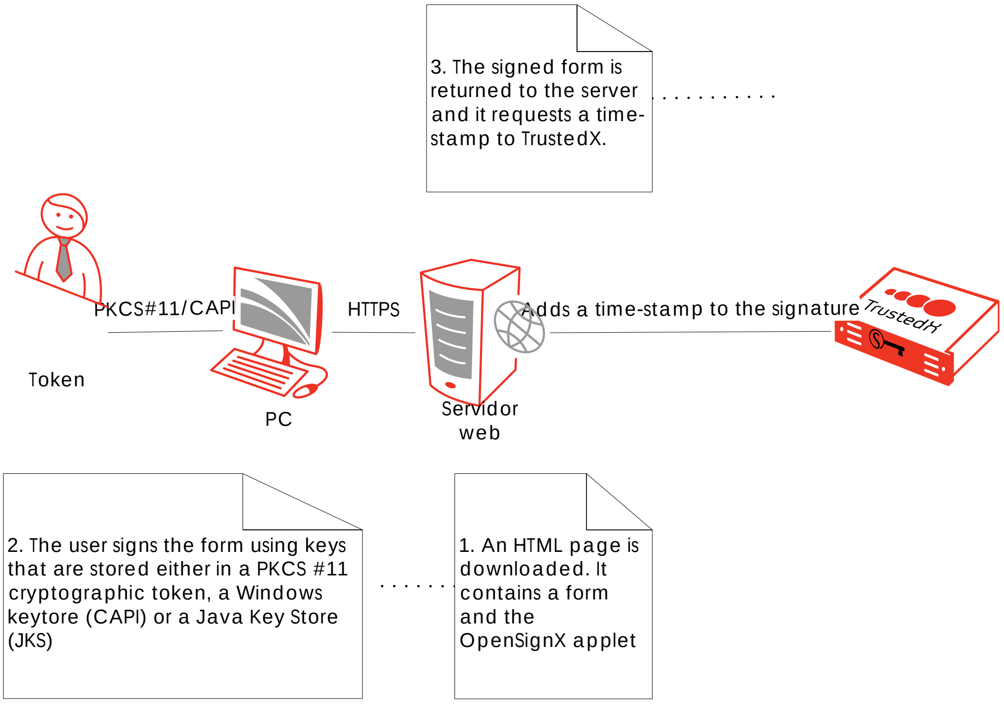

3. The signed form is returned to the server and it requests a time-stamp to TrustedX.
PKCS#11/CAPI
HTTPS
Adds a time-stamp to the signature
TrustedX
Token
PC
Servidor web
2. The user signs the form using keys that are stored either in a PKCS #11 cryptographic token, a Windows keytore (CAPI) or a Java Key Store (JKS)
1. An HTML page is downloaded. It contains a form and the OpenSignX applet

Figure 4-1. Local signing of a Web form.

However, even when user keys are managed centrally (i.e., for corporate users) there is an advantage to using the OpenSignX applet for signing Web forms. In this case (*Figure 5-2*), the applet is used in combination with the virtual card to take advantage of the applet's capability for requesting signature generation from the card, which means that this request does not have to be programmed into the Web server as the virtual card sends it directly to TrustedX.

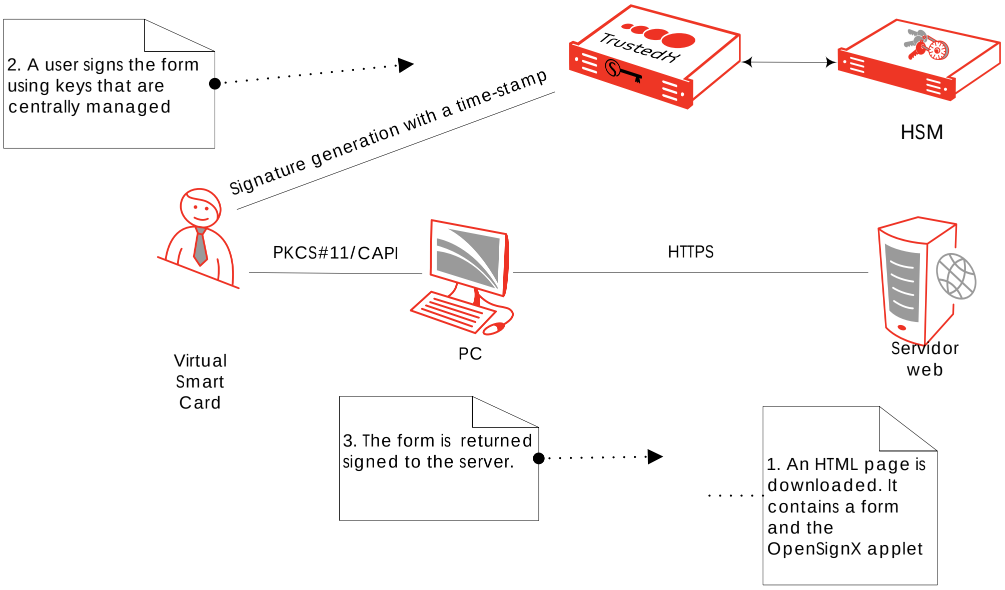

2. A user signs the form using keys that are centrally managed
TrustedX
HSM
Signature generation with a time-stamp
PKCS#11/CAPI
HTTPS
Virtual Smart Card
PC
Servidor web
3. The form is returned signed to the server.
1. An HTML page is downloaded. It contains a form and the OpenSignX applet

Figure 4-2. Web form centralized signature with OpenSignX and virtual card.

APPENDIX B

Supported Signature Standards and Algorithms

This appendix lists the signature standards and algorithms supported by TrustedX.

Standards

TrustedX supports the following standards.

Reference	*Standard*
[CMS]	Cryptographic Message Syntax, IETF RFC 5652
[DSS]	T. Perrin et al. *Digital Signature Service Core Protocols, Elements, and Bindings*. OASIS, Draft 27.
[CAdES]	*Electronic Signatures and Infrastructures (ESI); Electronic Signature Formats.* ETSI TS 101 733 V1.5.1, December 2003
[OCSP]	Online Certificate Status Protocol, IETF RFC 2560
[LDAP]	Lightweight Directory Access Protocol
[PAdES]	PDF Advanced Electronic Signature Profiles. ETSI TS 102 778-1, ETSI TS 102 778-2 y ETSI TS 102 778-3.
[PDFRef]	PDF Reference (sixth edition). Adobe Portable Document Format Version 1.7 or higher.
[PKCS#1]	PKCS #1 v2.1: RSA Cryptography Standard, version 2.1. IETF RFC 3447.
[PKCS#7]	PKCS #7: Cryptographic Message Syntax, version 1.5. IETF RFC 2315
[PKCS#11]	PKCS #11 v2.20: Cryptographic Token Interface Standard
[RADIUS]	Remote Authentication Dial In User Service (RADIUS). IETF RFC 2865
[SAML]	Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V1.1, September 2003
[SMIME2]	S/MIME Version 2 Message Specification, IETF RFC 2311
[SMIME3]	S/MIME Version 3 Message Specification, IETF RFC 2633

Reference	*Standard*
[SOAP]	Simple Object Access Protocol Version 1.1, W3C. May 2001
[SSL/TLS]	Secure Socket Layer / Transport Layer Security
[TSP]	Time-Stamp Protocol, IETF RFC 3161
[X509]	ITU-T Recommendation X509v3
[XAdES]	*XML Advanced Electronic Signatures*. ETSI TS 101 903, March 2006
[XML-DSig]	D. Eastlake et al. *XML Signature Syntax and Processing.*W3C Recommendation, June 2008
[XKMS]	XML Key Management Specification (XKMS 2.0)
[WSDL]	Web Service Description Language (WSDL) 1.1, W3C. March 2001
[WSS]	OASIS Web Services Security: SOAP Message Security 1.1 (WS-Security 2004) February 2006

Signature Algorithms

TrustedX supports the following digital signature algorithms:

- RSA-MD2
- RSA-MD5
- RSA-SHA1
- RSA-SHA256
- RSA-SHA384
- RSA-SHA256

For RSA signatures, the following key sizes are supported:

- Up to 4096 bits in software.
- Up to 4096 bits in HSMs approved by Safelayer (depending on the model).

TrustedX Signature Platform

Safelayer Secure Communications, S.A.

Telephone: +34 93 508 80 90

Fax: +34 93 508 80 91

Web: www.safelayer.com

SAFELAYER