Exploring ADSS Server Signing Services

ADSS Server is a multi-function server providing digital signature creation and signature verification services, as well as supporting other infrastructure services including Time Stamp Authority (TSA) services, OCSP, XKMS and SCVP validation services and certification services.

This solution sheet discusses various ways in which ADSS Server can apply digital signatures to documents and data. It explains how one or more "signing profiles" can be created and configured to add very effective management controls to the signing service. Using "Client Manager" to authenticate business applications and defining which "signing profiles" and signing keys can be used ensures that ADSS Server is a trustworthy system that can tightly control corporate signing keys for one or more applications, for example invoice production, order approval, payroll applications and so on.

Creating Digital Signatures

ADSS Server supports various types of digital signature including PDF, XML, PKCS#7, CMS and S/MIME. A large number of ETSI XAdES or CAdES or PAdES formats are supported. ADSS Server enables digital signatures to be created in a number of ways:

- Server-side corporate signing - using corporate signing keys held on the server
- Server-side personal signing - using end-user signing keys held on the server
- Client-side personal signing - using end-user signing keys held on the desktop / browser

The server can use software keys or keys held on hardware security modules (HSMs) or even USB tokens or smartcards. End-users can use Windows CAPI soft credentials or USB crypto-tokens or smartcards or roamed credentials – these are downloaded on demand to the end user browser memory and unlocked by the user.

ADSS Server can also be requested to enhance basic signatures generated by end-users and add timestamp and validation information to create ETSI long-term signatures.

Server-Side Signing using Corporate Keys

ADSS Server provides signing services for any registered business application. The application either uses ADSS Client SDK or direct OASIS DSS web services calls to the ADSS Server Signing Service and supplies the document or hash of the document to be signed. The signing profile is an optional parameter that identifies the type of signature to be produced. Other parameters can identify the desired signing key, reason for signing, etc.

Having identified the calling application and checked its permissions, ADSS Server then signs the document according to identified signature type and returns the signed object to the application. ADSS Server can manage multiple signing keys and certs and thus the signing service can sign documents using unique keys for each individual corporate entity or application.

The signature may be attached, embedded (for example inside a PDF) or detached (the signature object exists on its own, separate from the original document).

Server-Side Signing with End-User Keys

Where the document or data needs to be signed by end users ADSS Server can create and hold keys for these users. Once again a business application, typically a web-application, is used to interact with the user, show them the data they are being asked to sign and request them to confirm their intent to sign. The application passes this instruction to ADSS Server and this signs according to the identified signing profile and user key. The signed data is passed back to the application.

The diagram below shows such an example, where an end-user is asked to sign a transaction by the business application. In such cases the business application, perhaps a CRM, HR or ECM application, must have previously authenticated the user (often these details are known already) and registered the end-user with ADSS Server using the ADSS Certification Service to generate a key pair and obtain a signing certificate:

Watched Folder Server-Side Signing

Another scenario is where the business application, in this case the Ascertia Auto File Processor (AFP), automatically retrieves documents for signing from an input folder, requests signatures from ADSS Server and deposits the signed documents in an output folder. Multiple input and output folders and multiple signing profiles can be supported:
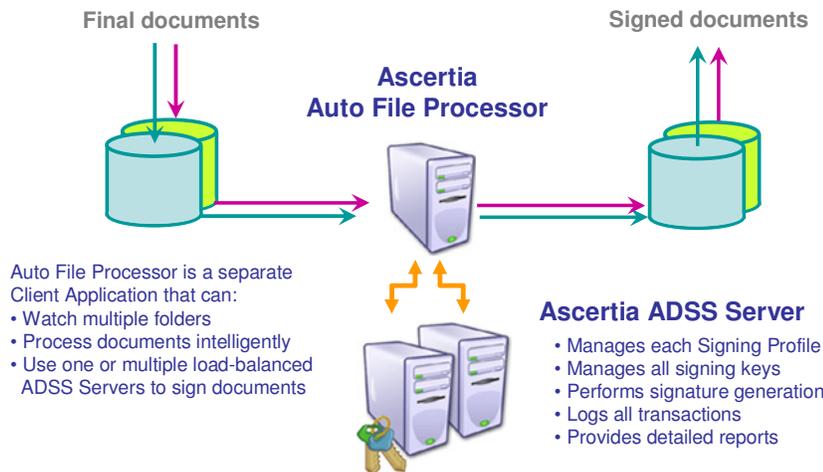

AFP is an intelligent Java application using ADSS Client SDK to handle interactions with ADSS Server. It is controlled by an XML configuration file that defines which folders to monitor, which file types to look for, which ADSS Servers to communicate with, how to handle issues that might arise and more.

Client-Side Signing

In this scenario, end-users hold their signing keys locally either within the Windows key store, or on smartcards or USB tokens, or they might use roamed credentials. In all cases the documents or data they are shown will be signed locally within the browser.

Ascertia's ADSS GoSign Applet has been created specifically to allow this scenario without requiring the installation of client-side signing software. The document to be signed may exist locally or be provided by the server (in full form or only its hash). The document can also be hashed locally and the hash is signed by GoSign applet within the end-user browser. ADSS GoSign can embed the signature OR ADSS Server can do this to create the final signed document. A key part of the process is using ADSS Server to verify the end-user signature and certificate. The following diagram illustrates this process:
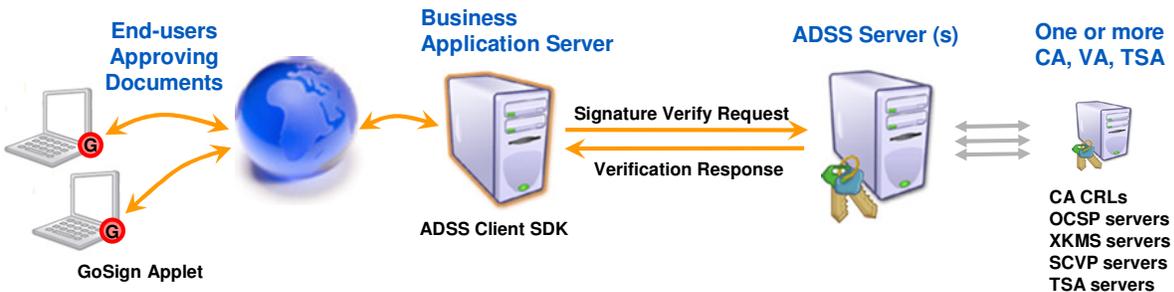
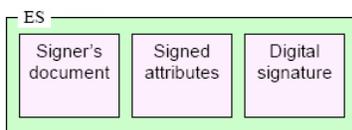## Client-Side Signing with End-User Roamed Keys and Certificates

This scenario is a variation on the above but in this case the end-user's keys are held within a secure key container on ADSS Server and only downloaded to ADSS GoSign Applet as required when the application needs the user to sign. The key container is never stored locally on the client machine and only accessed from within the GoSign applet. This avoids the need to have locally held signing keys and installing smartcard/USB drivers, thereby providing the functionality to digitally sign from any supported system.

## Client-Side Signing with Server Signature Enhancement

A further variation on the above is where the end-user's signature is enhanced as part of the verification request to ADSS Server, which adds timestamp and validation information to extend the existing signature into a 'long-term' signature enabling historic review and verification.

Here the user's original basic signature (depicted directly below) is extended with the addition of timestamp and certificate revocation information (following diagram):

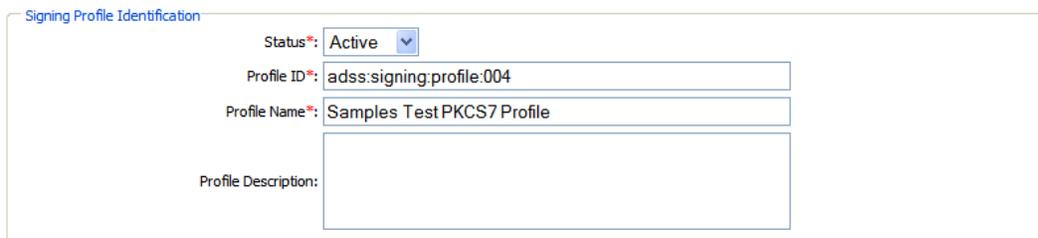Basic Signature Format   Long-term Signature Format:
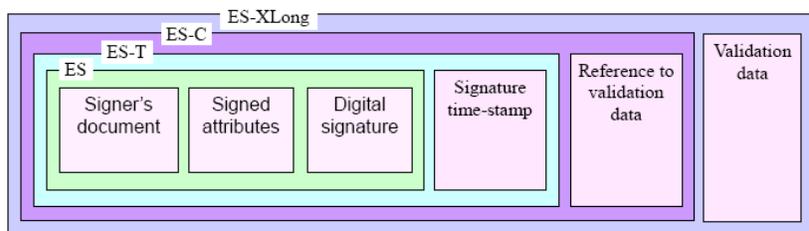
ADSS Server Signing Profiles

As mentioned earlier signing profiles are used by ADSS Server to configure the digital signature formats and appearances that are produced. Once a signing profile is configured by a suitably privileged ADSS Server operator and a business application is enabled to access it, then signing requests can reference this signing profile in the web service request message. This has the huge benefit of removing all signature configuration management from business applications. Any number of signing profiles can be configured to suit different application needs. The main attributes that can be configured in a signing profile are:

- The Signing Profile identifier
- The digital signature format
- The signature type
- The default signing certificate

Signing Profile Identifier:

Signing profiles are uniquely identified within ADSS Server via a system defined Profile ID:
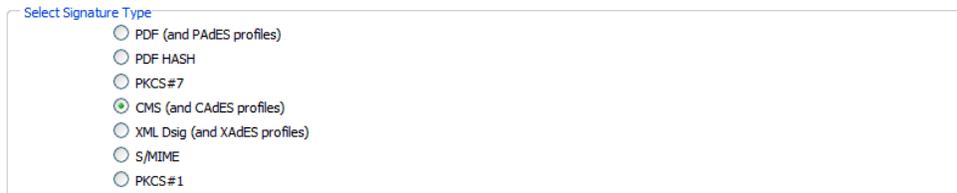
```
Signing Profile Identification
                    Status*:  Active ▼
                  Profile ID*:  adss:signing:profile:004
               Profile Name*:  Samples Test PKCS7 Profile

          Profile Description:
```

Signature Type:

The most important element of a signing profile is the type of signature to be generated. ADSS Server enables the main signature type to be selected from a list of signature categories:

```
Select Signature Type
   ○ PDF (and PAdES profiles)
   ○ PDF HASH
   ○ PKCS#7
   ● CMS (and CAdES profiles)
   ○ XML Dsig (and XAdES profiles)
   ○ S/MIME
   ○ PKCS#1
```

In most cases, the choice of signature type is then further refined depending upon the category chosen above. For example for PDF (and PAdES profile) signatures a choice can be made between Basic PDF signatures, PDF signatures with embedded timestamps and long term PDF signatures with embedded timestamp and revocation status information:

```
Signature settings
   Select the signature format type
      ● Basic PDF signature
      ○ PDF signature with embedded timestamp
      ○ Long-term PDF signature with embedded timestamp and revocation status info
```

Similarly for CMS /XML signatures (and CAdES /XAdES signature profiles) the choice is made between the various supported signature types. For CMS and CAdES these are:
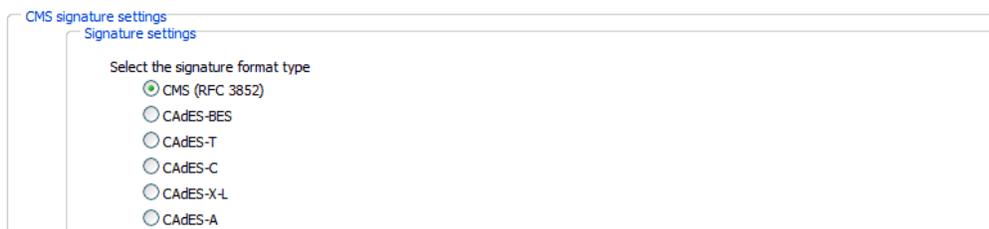
```
CMS signature settings
   Signature settings
      Select the signature format type
         ● CMS (RFC 3852)
         ○ CAdES-BES
         ○ CAdES-T
         ○ CAdES-C
         ○ CAdES-X-L
         ○ CAdES-A
```

For XML (and XAdES profiles) the choice becomes:

XML signature settings
Signature settings

Select the signature format type
- ⦿ XML DSig (W3C)
- ◯ XAdES-BES
- ◯ XAdES-T
- ◯ XAdES-C
- ◯ XAdES-X (Type2)
- ◯ XAdES-X-L
- ◯ XAdES-A

Signature Format:

The signature format allows the signature /document relationship to be defined, "Enveloping" means the signature wraps around the document, "Enveloped" means the signature is embedded within the document, and "Detached" means the signature is provided as a separate object to the original document:

Signature type settings

Signature/Document Relationship: Enveloping ▾

Signing Certificate:

It is possible to configure a default certificate to be used with this signing profile. This certificate can be overridden in the request thus allowing business applications to use one signing profile with the range of certificates that are available to it.

Default Signing Certificate

Default Signing Certificate (overridable): samples_test_signing_certificate ▾ [View Certificate]

This is also where the type of certificate that is acceptable for signing purposes is defined. In many cases a Key Usage of "non-repudiation" is desirable or even mandatory.

CAdES and XAdES Profiles

For CAdES and XAdES profiles, ADSS Server supports Explicit Policy-based Electronic Signatures (ES-EPES). Here a Signature Policy Object ID, a Signature Policy URI and Signature Policy User Notice can be specified:

Explicit policy based electronic signature (EPES) settings

☑ Add Signature Policy Identifier (Creates Explicit Policy-based Electronic Signatures)

Signature policy object ID: 1.2.3.4.5
☑ Add Signature Policy URI
http://www.ascertia.com
☑ Add Signature Policy User Notice (Restriction 200 characters)
test policy

Also for CAdES and XAdES profiles, depending upon which format type has been chosen, a Time Stamp Authority (TSA) can be selected.

In addition, as shown below, the profile may result in validation data being added into the signature. These types of signature stop local time settings being an issue and simplify the task of future signature verification:
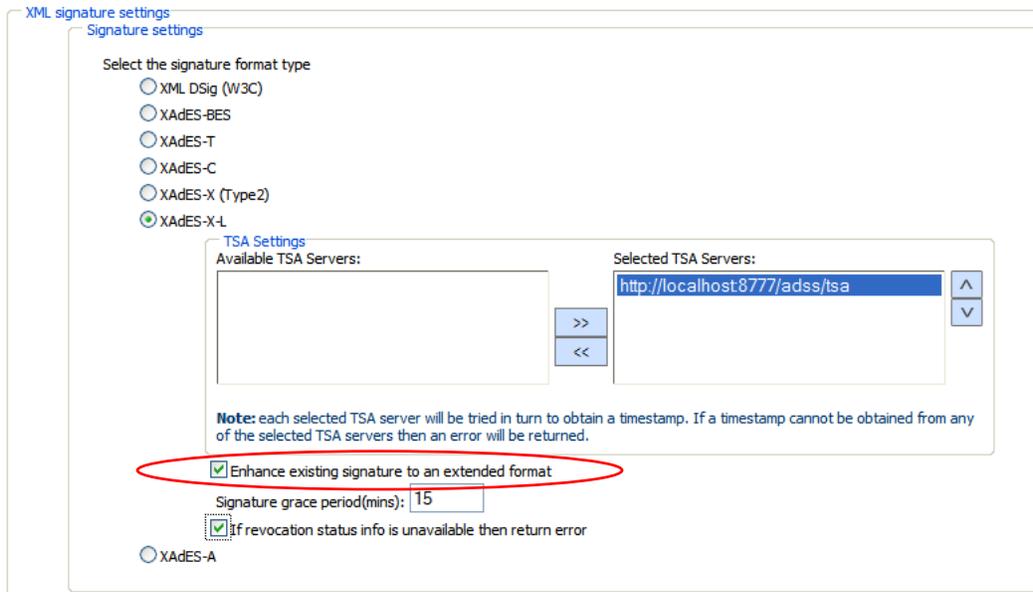
The grace period field is a very useful element for those projects that require this capability. In addition if revocation information is not available, e.g. a back-end OCSP responder fails to reply, then a setting enables the signing operation to fail and return an error condition. This is vital as many other products simply do not enforce such policies and "assume" its okay to continue without this information. It will be expensive and awkward to discover this some time later!

Visible and Invisible PDF Signatures

When a PDF document is signed, the existence and validity of the signature can be verified by opening the document with a PDF viewer such as Adobe Reader or Ascertia's PDF Sign&Seal or ADSS GoSign Applet products.

The signature to the right shows the detail of an invisible signature when viewed through PDF Reader.

Invisible signatures are useful when securing documents where no space exists for a digital signature appearance.

ADSS Server and GoSign Applet have the ability to add a visible signature appearance into the PDF document. This has advantages because it looks to less technical recipients as though the document is conventionally and formally signed and yet also secures the document against change. A visible trust status icon is also very useful to end-users.
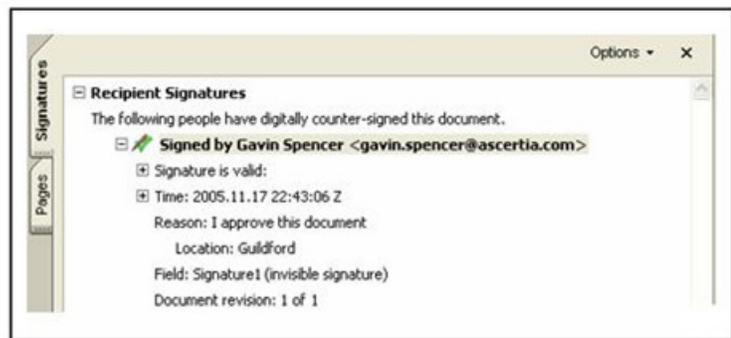
Signature Appearance Details:

ADSS Server has a sophisticated set of controls to determine how signature appearances should be displayed. These options include the dimensions of the signature, where it should be placed on the PDF document, if labels are to be displayed, whether the reason for signing, location, date and time and contact details are shown or not. Hand-signature and company logos can be applied and engineering seal layouts are also supported.

Local languages are of course supported for both labels and the text content.

Authenticating Business Applications

Business applications are the clients that call the ADSS Server. They can be authenticated using the following three techniques depending on the level of security desired:

- Registering a business application within the ADSS Client Manager and assigning it an "Originator ID". The business application must then use this "Originator ID" in its service request messages in order for it to be authenticated successfully by ADSS Server.

- The service request messages can be sent over an SSL connection with client authentication enabled.

- The business application can sign the XML service request message using a request signing certificate.

The business application's certificates must be registered within ADSS Server Client Manager:
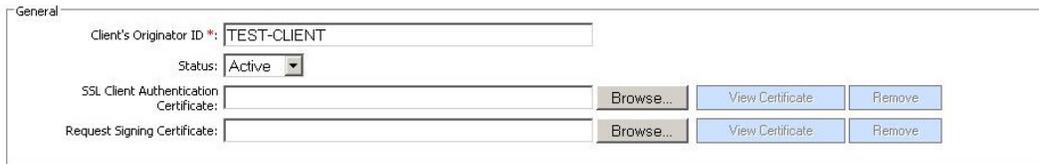

Originator ID alone, or Originator ID + SSL, or Originator ID + SSL + signature can be used to provide strong levels of client authentication.

Authorising Business Applications to use Signing Profiles

When registering business applications within ADSS Server Client Manager it is possible to configure which of the existing signing profiles (as discussed above) are to be made available for this business client:
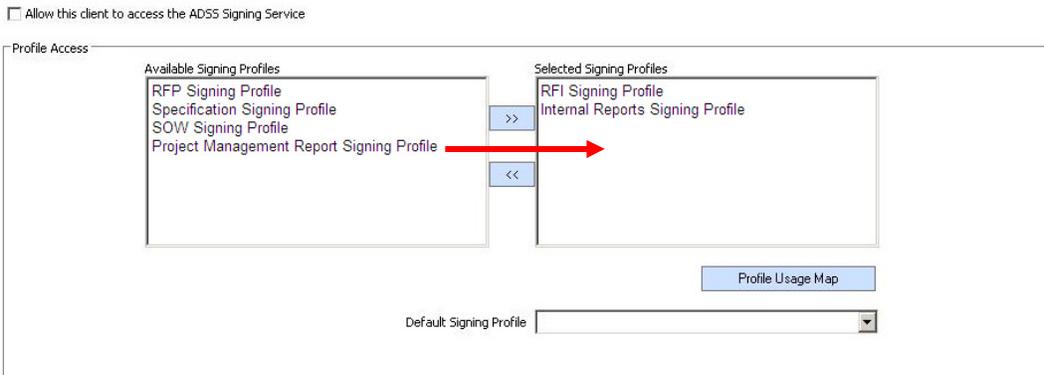
The first checkbox within the Client Manager defines whether the business application can make signing service requests. Authorised administrators can simply select an available profile and

---

move it across to the selected list by clicking on the '>>' button. This updates the profiles assigned to this business application.

In a similar way administrators can also control which set of server held keys and certificates are available to this business application for use with the signing profile.

In summary it is possible to restrict which business applications can make signature creation service requests, which signing profiles they can reference, which document signing keys they can use, the format of the signatures to be applied and where and how these will appear. Therefore ADSS Server not only provides strong authentication of client applications but also strong authorisation checks to ensure client applications privileges on the system are controlled.

Business Application Development

Business Applications can be developed for particular projects but there are also a number of pre-prepared applications developed by Ascertia e.g. the Auto File Processor (AFP).

When a new Business Application is required this can be written in Java or C# (.Net) as ADSS Client SDKs are provided in both languages. Alternatively, the underlying XML/SOAP protocols may be implemented directly if desired. For high performance applications, such as Auto File Processor, communication with ADSS Server using the optimised HTTP/S interface is recommended to avoid the overheads associated with web-services.

ADSS GoSign Applet Deployment

When using GoSign applet, intelligent software components and a framework are provided to integrate the solution within the Business Application to achieve the desired workflow.
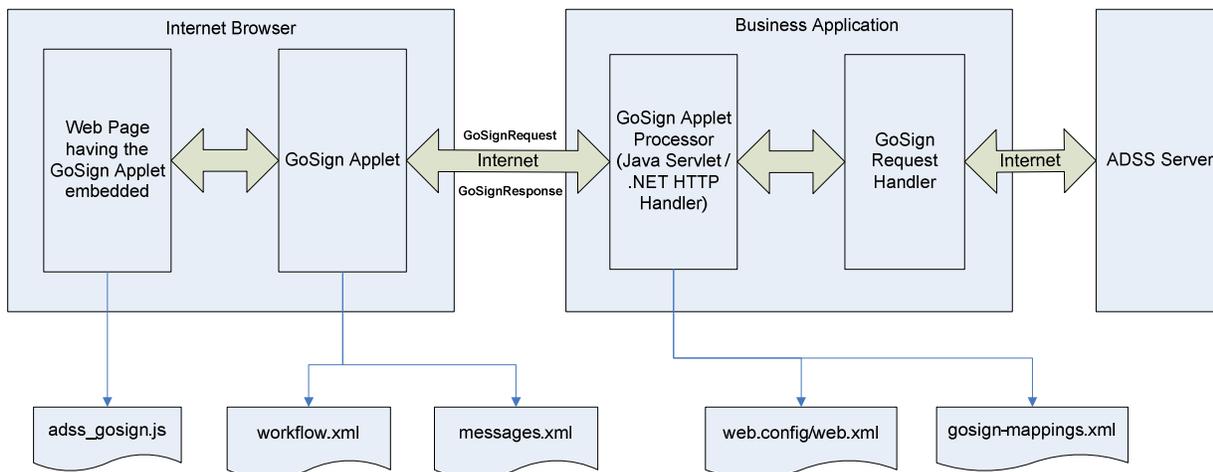
JavaScript files are used to control the user's web page and workflow files specify the signing workflow. Request handlers can be written in Java or C# and these call the ADSS Client SDK to interface to ADSS Server.

Authorised Signing

ADSS Server provides support for getting multiple people to sign and authorise the use of a special corporate signing key held on the server. Imagine a set of high value documents that need to be approved before final submission or publication. In this scenario the business application is responsible for getting a request for corporate sign-off (an authorisation control file) signed by one or more individuals. The authorisation control file contains both the documents to be signed and the signatures of the approvers.

The authorising signatures are checked at the ADSS Server by reference to an Authorisation Profile specified within the Signing Profile:

The Authorisation Profile specifies the authorising certificates and the number of approvers required. 'M' out of 'N' schemes can be implemented in this way:


Further details about Authorised Signing can be found in the solution sheet 'Authorising Corporate or Qualified Signatures'.

Signing and Encryption

GoSign Applet also supports the encryption of documents after signing e.g. as part of a secure local storage or upload process for any web-application, for example e-tendering solutions.

In this scenario, GoSign Applet is used to create an XML signature and then encrypts the payload element within the signed XML structure. Later the document can be decrypted by ADSS Server at the request of the business application using decryption keys held at the server.
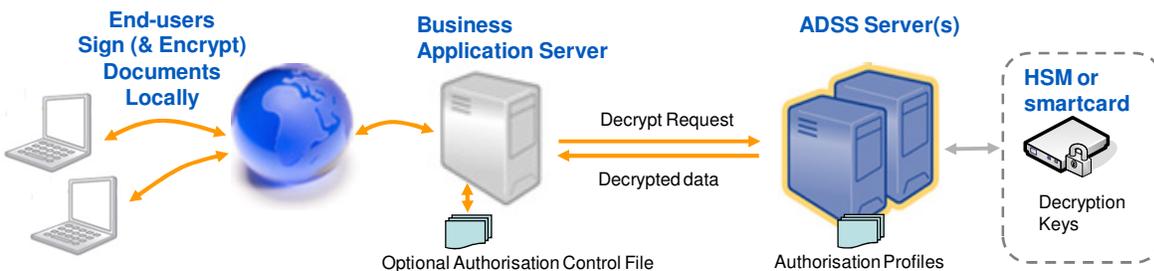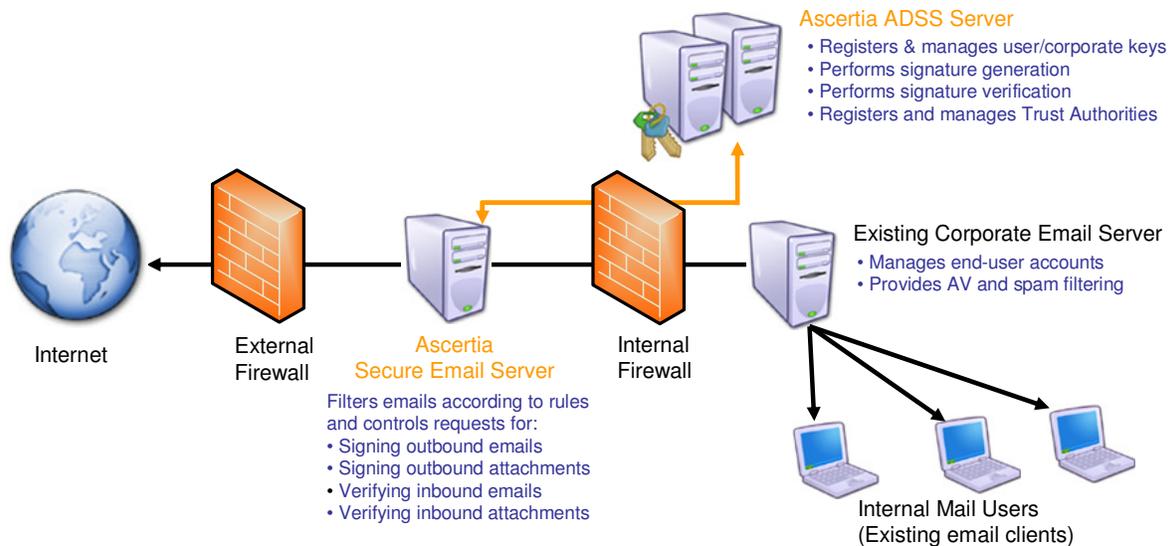

For further details and an example, refer to the GoSign Developers Guide.

Email Signatures

ADSS Server can be used in combination with Ascertia's Secure Email Server to provide an email signing and verification capability. Secure Email Server works together with a company's existing email server to 'post process' outgoing mails, providing various filter options to sign emails and attachments according to sets of configured rules.

Secure Email Server can also be used to pre-process incoming emails and attachments by verifying their signatures. Rules can be set-up on how to handle "untrusted" emails/attachments.

Ascertia ADSS Server
• Registers & manages user/corporate keys
• Performs signature generation
• Performs signature verification
• Registers and manages Trust Authorities

Internet

External Firewall

Ascertia Secure Email Server

Internal Firewall

Existing Corporate Email Server
• Manages end-user accounts
• Provides AV and spam filtering

Filters emails according to rules and controls requests for:
• Signing outbound emails
• Signing outbound attachments
• Verifying inbound emails
• Verifying inbound attachments

Internal Mail Users
(Existing email clients)

For further information about the Secure Email Server refer to the Secure Email Server Datasheet.

Server Integration and Management

ADSS Server offers very easy integration options:

- Easy integration within existing applications for automated signing and verification using web services, HTTP Post, the Auto File Processor (uses one or more watched folders) or the Secure Email Server (automatic email processing)

- Accessibility from anywhere, using HTTP or HTTPS

- Providing high performance and scalability using J2EE architectures

- Offering resilience and availability by using multiple servers

ADSS Server offers excellent management options:

- The administrative interface uses a secure web-browser interface (using client / server SSL) to enable easy local or remote management

- Operators are strongly authenticated and there are fine-grained role-based access controls to allow or deny operators access to ADSS Server functionality. Detailed logging ensures all pre-change and post-change data items are logged.

- Managed services options – multiple users can be managed, authorised for different services and detailed transactional records maintained.

Summary

ADSS Server provides a very rich set of features for supporting different signature types, configuring different signing profiles and linking these to business applications. OASIS and ETSI standards are well supported as are other important standards such as PEPPOL. Given the sophistication of signing options using server-held keys, roamed keys and certificates and local hashing and signing using ADSS GoSign Applet, ADSS Server can be used to solve any signing requirement. ADSS Client SDK, ADSS Auto File processor and ADSS Secure Email Server ensure that signing security services are easy to integrate.

Strong authentication, authorisation and role based controls together with secure logging of operator actions and application request/responses ensure the highest levels of security.

Identity Proven, Trust Delivered

info@ascertia.com   www.ascertia.com