

WASP (Web Activated Signature Protocol)

Application Note #1 – Signature Validation and Multiple Signatures

WASP was designed to be a “thin client” browser add-in. This among many things, excluded local signature validation. The major reason for that is that in an information system environment (which is the sole target for WASP), a validation server can perform the validation *once* (typically immediately after the receipt of a signature), and save the result in a database table for future references.

This scheme is also a prerequisite for Identrus-style (“pay-per-view”) certificates as such certificates cannot easily be validated locally due to trust network specific validation software and contracts.

A further advantage with off-loading validation exclusively to the information system layer, is that you can avoid warnings due to expired certificates when looking at old signatures, as you rather return the result of the original validation rather than repeating it. The need for *end-users* to study, or even know what a “certification path” is, seems marginal.

In addition, the information system approach eliminates the need to download root certificates for other persons’ certificates *making the user’s private keys and certificates the only required local unique resource.*

Signatures are always detached using WASP due to the reasons listed on page 3.

The following pages show how an information system can cope with user display of signatures, as well as supporting multiple signatures, giving users an almost “PKI-free” world in spite of actually using PKI.



Company Registration/Update Form

Current user: [John Smith](#)

Company file: [ACME Corporation, 567845-4534](#)

File status: *Updated, awaiting signatures*

Show history...

Currently signed by: [Mary Donahue](#), [Steve Miller](#)

Missing signatures: [John Smith](#), [Yoki Masaki](#)

To company registration wizard. Removes previous signatures

Revise company file...

Sign company file...

Help

WASP Workflow application sample

To signature application (WASP). Shows the completed company file and requests a signature. Returns to this view after the signature has been performed or is cancelled

First name: **John**
Last name: **Smith**
Citizen code: **19750710-1513**
Address: **etc. etc**

Shows the current company file

To list of previous filings

Signed: **10-Nov-2004 10:32**
First name: **Mary**
Last name: **Donahue**
Citizen code: **19701110-1612**
Address: **etc. etc**

Optional: Certificates etc.

Send request via e-mail
First name: **Yoki**
Last name: **Masaki**
Citizen code: **19500524-6413**
Address: **etc. etc**

[John Smith](#) Hyperlink



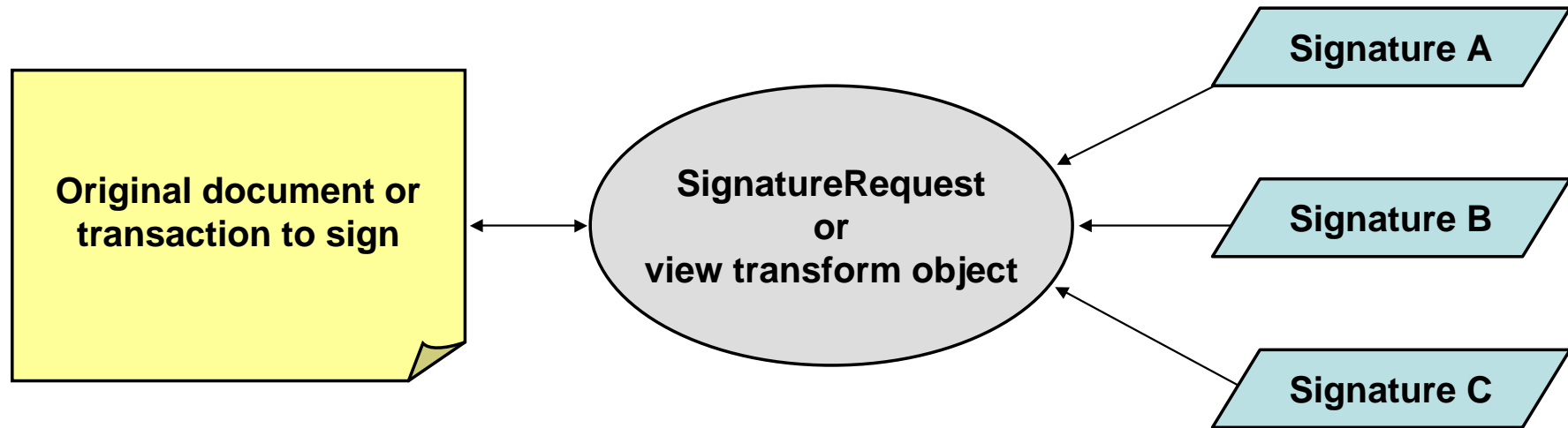
Pop-up window



Replaces current window

The information system approach to multiple signatures:

A document and a set of associated *detached* signatures



The primary advantage with this scheme (besides not having to download and process signatures of other signers in the client environment), is that documents are not “clobbered” with signature data. Signatures and WASP `SignatureRequest` (or view transform) objects are typically stored in separate database tables (*which also enables a straightforward way of adding digital signatures to existing systems*). The relation between signers with respect to authority then becomes an information system issue only, not requiring any specific code on the client side in order to differentiate between an authorization “counter signature” and a “peer signature”. This makes sense as a “genuine” counter signature (like in CMS), in itself does not have any specified semantics like that the outermost signature is the most authorized etc.