# Visual eID – Problem Statement

## The basic problem

The basic problem is that X.509 certificates used to identify entities in electronic exchange do not have any meaningful visual representation.

One could say that the generic function V(x) is missing, where V is the visualization of any certificate x. V(x) represents a function that can take any certificate x and create a visual representation of it which, when presented to a human, would reveal at least the following essential aspects of the certificate:

- **Type of certificate**, such as web service provider identity, code signing author identity, personal EU Qualified Certificate ID, etc.
- **The Issuer**. The identity and possibly trademark of the entity who issued this certificate and stands behind it.
- **The subject**. The entity identified through this certificate.

## Who needs this?

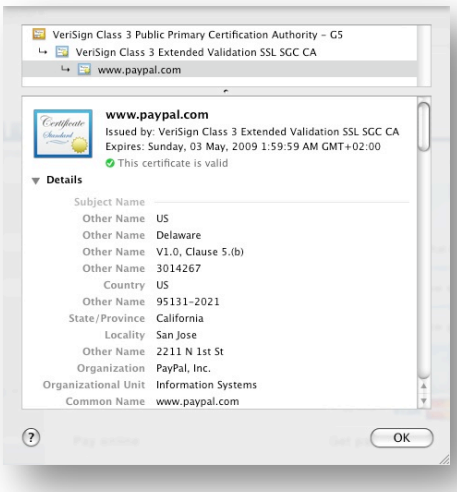
Humans need this information in some form when they come in contact with either certificates directly, or the consequence of using them. Some cases are:

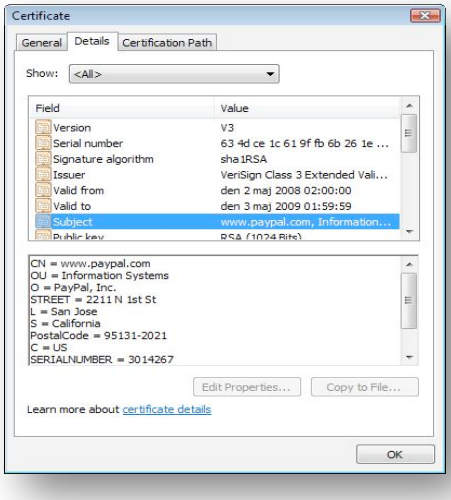
- A person establishes contact with an authenticated entity, such as a web server, access point, government service, software provider. The user is to be informed about what entity that was authenticated.
- A person consumes information that has been authenticated. This might be a signed e-mail, a signed document, a signed form, a signed contract. The person is to be informed about who the signer is.
- A person is requested to authenticate to a service, or to sign some information. The user is further requested to select an appropriate certificate (instrument of identification). The user is to understand both that the selected certificate is appropriate, but also what type of personal data he/she is sharing by using this certificate.

It is **impossible** to provide these functionalities in the absence of a V(x) function.

## Why does not current certificate viewers qualify as V(x) function?

- Contextualized viewers that are designed to view only a specific type of certificates are static and not generic. The better job they do, the more they need to be locked to a very specific certificate profile. If the profile changes, the code of the viewer needs to be updated.
- Certificates do not contain specific information about the data they contain. This lack of information makes it impossible to create V(x). In particular, the following information is missing:
    - **Certificate type** (e.g. Is this a personal ID, Web ID etc) is derived from a set of complex identifiers such as EKU, Policy OID, Key usage and Qualified Certificate Statements etc. The V(x) function would need to learn them all and then subjectively create a display name for each type.
    - C**ertificate issuer**, **subject name** fields as well as **Subject Alt Name**, carry data with undefined display names. What is stored in "commonName" may not be a "Common name" but something else, like a DNS name. countryName, may be country of residence, country of citizenship or something else. serialNumber can be virtually anything. There are further no general agreement about basic display names for these fields and even less how these display names are to be translated to multiple languages. This is why we generally present just their abbreviations in current generic UI (e.g. CN=, O=, C= etc).
    - Any **graphical elements** that would enhance the V(x) function such as images representing aspects of the certificate type, issuer or subject.
- Generic UI that simply dump technical content (our current "detail" certificate viewers) does not qualify as V(x). They are useful for experts and educated users but not for the general population, and they still lack vital information for the reasons above. I.e. What information is carried in serialNumber attributes?



## Two basic solutions: Complete certificate image or structured display metadata?

To successfully solve this problem, the certificate must provide more data. The easiest way to do this without exploding the size of certificates, is to reference an external hashed "V(x)" file from the certificate. This can be done through the extension mechanisms defined in RFC 3709.

Two basic choices are available for the type of referenced data:

1. A structured file (ASN.1 or XML) that carries all necessary data (localized display names, graphical elements and optionally placement and scaling information).
2. By allowing the issuer to generate a complete visual representation of the whole certificate and provide that through a well defined format such as PDF/A or SVG.

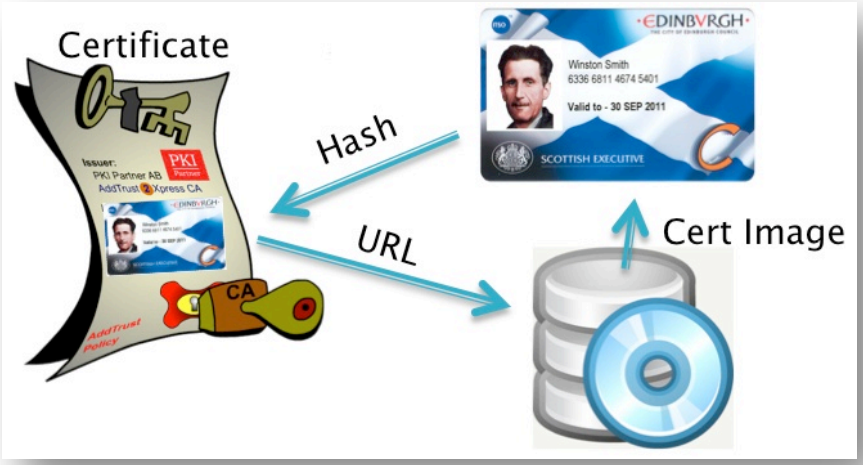



Fig. Option 2)

The problem with option 1) is that it might be too complex. It would take many years to standardize and even longer to adopt. Many situations where this is needed, the simpler option 2) is both adequate and a lot easier for all parties to implement. It would further allow the V(x) output to be consistent over multiple applications and platforms.

As a result of this problem and its need for a solution, a standard is being developed in IETF that specifies option 2, allowing any image format but specifying how to reference an image file.