Resolver-Mania

1. Why do we need all these resolvers?

For security and comfort reasons. In the XML Security package, there exist many kinds of Resolvers for different purposes. Resolvers in this package do the same job as an EntityResolver in the SAX package: retrieve information from the apropriate location and give it to the parser/software who needs it. The reason for offering these different Resolvers is that it should be under complete control of the application which connections to the network are made. In the security area, it wouldn't be a good idea to imediately fetch some documents from the web or make other connections only because you want to verify a Signature. This resolver framework gives the application developer the ability to have total control about the interface from the library to the rest of the world.

2. Types of resolvers

2.1. ResourceResolvers

A ResourceResolver is used by a Reference to retrieve the signed resource from it's location. Different resolvers exist to get signed portions from the XML document in which the signature resides, to make HTTP connections or to fetch files from the local file system.
The concept of a ResourceResolver is very similar to an org.xml.sax.EntityResolver, but in contrast to that Interface, the ResourceResolver is able to de-reference contents *inside* an XML document.

2.2. StorageResolver

A StorageResolver is used by KeyInfo and it's child objects / Elements to retrieve Certificates from storage locations. This approach is used to allow a user to customize the library for use in a specific corporate environment. It's possible to write StorageResolver s who make requests to LDAP servers or to use specificic PKI interfaces.
Bundled with the software come three sample StorageResolver s which can be used for common tasks:

- The KeyStoreResolver is able to retrieve Certificates from a JAVA KeyStore object. This KeyStoreResolver is constructed from an open JAVA KeyStore.
- The SingleCertificateResolver resolves only to a single Certificate. The

SingleCertificateResolver is constructed using this single Certificate.
- The CertsInFilesystemDirectoryResolver is useful for resolving to raw X.509 certificates which reside as separate files in a directory in the filesystem. Such a resolver is needed for verifying the test signatures from Merlin Huges which are bundled in a directory.

StorageResolver s are supplied to the KeyInfo's `addStorageResolver()` method.

Generally, a StorageResolver has only a method to return an Iterator which iterates through the available Certificates.

2.3. KeyResolver

A KeyResolver is used by KeyInfo to process it's child Elements. There exist two general classes of a KeyResolver :

- If a ds:RSAKeyValue or ds:DSAKeyValue or ds:X509Certificate is used inside the ds:KeyInfo, the resolvers can return a public key or Certificate directly without further action, because the key itself is contained inside the ds:Signature.
- If there is only key material identification information like a ds:KeyName or the serial number of the Certificate, the KeyResolver must use the StorageResolvers to query the available keys and certificates to find the correct one.

Of course, there are cross-dependencies: e.g. a KeyResolver named RetrievalMethodResolver uses the ResourceResolver framework to retrieve a public key or certificate from an arbitrary location.