So You Want to Use the Raw XML Element for a Genesis II Calling Context as the Context for a Genesis II Grid Tool

Mark Morgan

Abstract

This document details the steps that need to be taken in order to implement the Unicore/Genesis II security interoperability work around discussed on conversations leading up to 28 April 2011. In essence, the work around is to have the Unicore endpoint extract the calling context information from the SOAP headers without requiring them to understand its contents, to then write that information out to the file system, and then call a Genesis II client tool (which will in turn use the context information) to do correctly delegated file staging.

On the Genesis II Client Side (Not the Delegatee, but rather the BES Activity Creator)

For delegation to happen correctly, the Genesis II client must delegate the calling context to the correct certificate. This means that the correct security headers needs to be placed in to an EPR for the target Unicore BES container. This has so far been done manually using the **mint-epr** tool that comes with Genesis II.

On the Unicore Side

When the target SOAP message arrives at the Unicore BES endpoint, the Unicore system must find the SOAP header element whose QName is **{http://vcgr.cs.virginia.edu/Genesis-II}calling-context** and that element must be written in its entirety as a stand-alone XML document into a file called user-**combined.xml** located in any directory that the developers wish to use.

Prior to launching any Genesis II command line tools that would use this information, the following must be true:

- An environment variable called **GENII_USER_DIR** must be set which indicates the full path (relative would work, but I recommend absolute) to the directory that contains the appropriate **user-combined.xml** file.
- An environment variable called **GENII_UNICORE_DELEGATEE_DIR** must be set which indicates the full path (relative would work, but I recommend absolute) to the directory that contains:
    - A PKCS12 keystore called **delegatee.pfx** that contains the X.509 certificate and associated private key for the credential to which the Genesis II calling context was delegated.
    - A Java properties file called **delegatee.properties** that contains information describing how to access the certificate and private key contained in the **delegatee.pfx** file described above.

Format of the delegatee.properties Properties File

The **delegatee.properties** properties file has the following properties that must be defined correctly:

Property Name	Required?	Purpose
delegatee.keystore.password	yes	Gives the password to the entire keystore
delegatee.key.password	no	Gives the password to the private key for the delegatee credential. If absent, the keystore password is used.
delegatee.certificate.alias	no	Gives the alias for the X.509 certificate stored in the keystore that will be used. If absent **and** the keystore contains exactly one certificate, then that sole certificate is automatically used.
delegatee.key.alias	no	Gives the alias for the private key store in the kesytore that will be used. If absent, then the alias used for the X.509 certificate is used.