Extending UNICORE 5 Authentication Model by Supporting Proxy Certificate Profile Extensions

Kate Stamou,
Dr.Fredrik Hedman,
KTH
Contributions from Anthony Iliopoulos

UNICORE Summit 2007
Rennes, France

Overview

- Motivation: near-term authentication interoperability between UNICORE and the other middleware systems. OMII-EU scope

- Goal: integration and support of proxy X.509 certificate profiles as main tokens of authentication

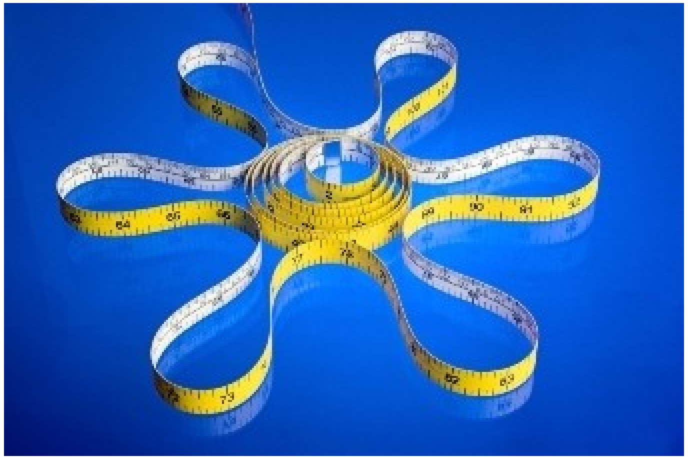
- Existing UNICORE 5 Authentication Model

- Implementation of Proposed Extension

- Validation Testing

- Conclusion

Motivation



Motivation

·Immediate and direct interoperability issues need to be addressed, as many different middleware systems have to co-exist in large grid settings

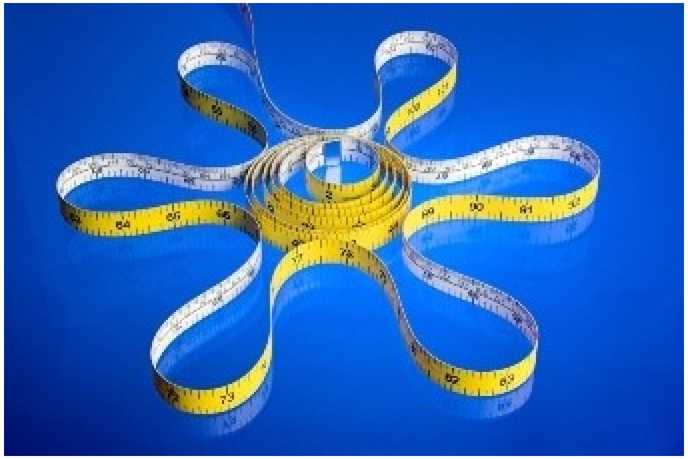
·Introduced by OMII-EU, as one of its initial Milestones with relevance to Security for Month12

· Currently most grid middleware systems use proxy X.509 certificates as main tokens of authentication (and authorization)

· Compatibility in a transparent and lightweight manner

·Main Use Case: Enable existing user credentials from other grid middleware systems to be used by the UNICORE system (Uniform authentication token between dominant middleware systems)

Goal: Support of Proxy X.509 Certificate Profiles

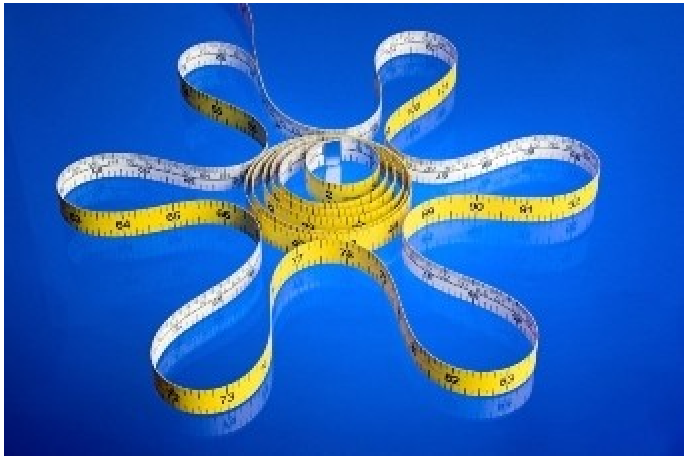


Goal: Support of Proxy X.509 Certificate Profiles

- Extend the UNICORE authentication model in order to include optional authentication verification of proxy certificates

- Enabling Single Sign-On functionality

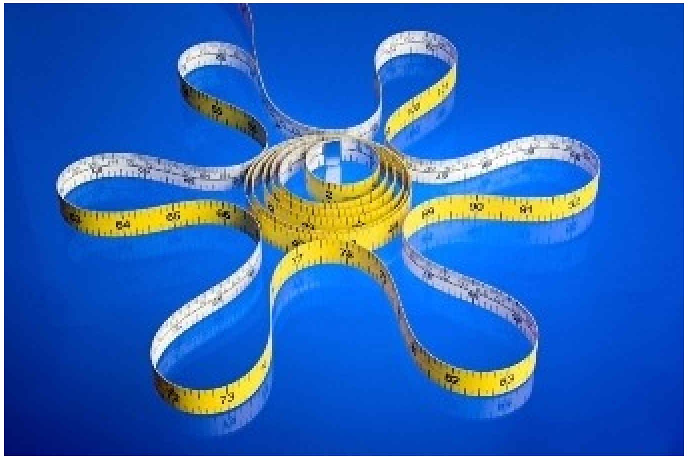
Existing UNICORE 5 Authentication Model



Existing UNICORE 5 Authentication Model

- End-to-end user X.509 certificates as main tokens of authentication
- Gateway acts as central entrance point for task submissions and incoming client authentication
- Static list of trusted CA's kept in gateway configuration
- Authenticated AJO is forwarded to a configured NJS
- No form of restricted privilege delegation

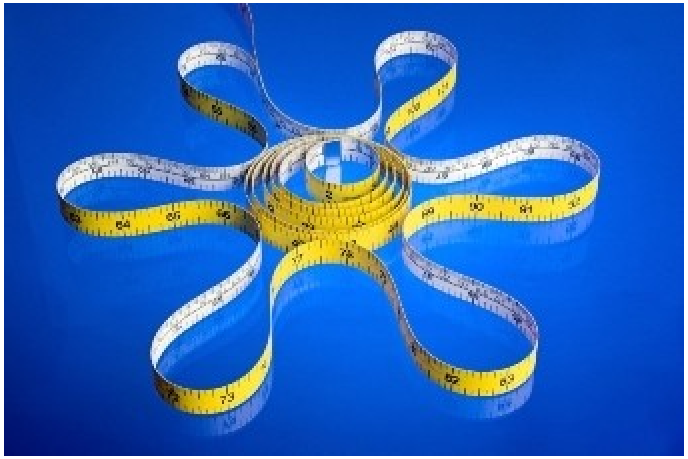
Implementation of Proposed Extension



Implementation of Proposed Extension

- Java API does NOT natively support proxy X.509 profiles
- No third-party libs provide this kind of functionality
- Solution: override the default TrustManager methods
  - Instantiate a default TrustManager covering verification of plain (non-proxy) X.509 certificates
  - On failure, attempt to verify the client certificate chain using an external proxy certificate path validation algorithm (COG-JGlobus provided)
  - Ultimately, if the above fail, client is not-authenticated

Validation Testing



Validation Testing

Low-level development tests conducted with the aid of OpenSSL tool

- OpenSSL Demo CA creation
- Gateway configured to trust certs issued & signed by demo CA
- User cert/key pair issued by the demo CA
- Proxy X.509 cert generated from the above
- Test connections to gateway using OpenSSL client tool with the user cert/key pair as well as the user proxy cert

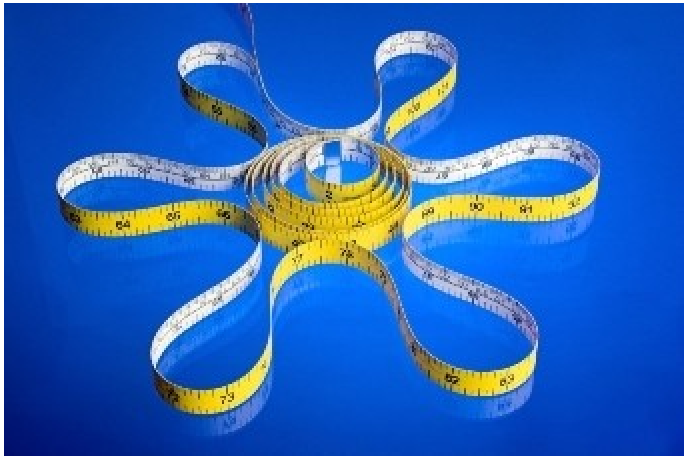
Validation Testing

·Proxy certificates generated using grid-proxy-init (both of Globus & gLite systems)

Following set of tests cases were realized to verify the correctness of the implementation by validating the proper authentication decision:

- Connecting using plain (non-proxy) user cert signed by trusted CA
- Connecting using plain (non-proxy) user cert signed by non-trusted CA
- Connecting using proxy user cert signed by trusted CA
- Connecting using proxy user cert signed by non-trusted CA
- Repeated the above using various styles of proxy-style certs
- Repeated the tests for expired proxy certificates

Conclusion



Conclusion

- Configurable option for UNICORE6

- Everything (demo, packages including src, etc.) can be found at:

  http://www.pdc.kth.se/~kstamou/interop.html

-

- Possible equivalence of dominant authN/authZ modules...

Questions?

Kate Stamou
kstamou@kth.se
Dr. Fredrik Hedman
hedman@kth.se

Thank you.