7. Identity and access management

<< Prev Next >>

7.1 Do you have an enterprise Identity and Access Management roadmap?

	Yes	No	In the process of creating one
UBC			✓
Michigan			✓
Cornell		✗	
Georgetown	✓		
Ohio State	✓		
UMUC	✓		
UofT			✓
MIT	✓		
UW-Madison	✓		
Washington			✓
UC-Irvine	✓		
Colorado			✓
Indiana	✓		

7.2 When applications invoke services on behalf of a user, are requests represented as coming from the user?

	This is not a goal	This is an architecture goal but it is only sometimes implemented	This is generally implemented for services in the local domain	This is generally implemented for services in the local domain and in the cloud	Other
Ohio					
UMUC		✓			
UofT		✓			
MIT			✓		
UW-Madison		✓			
Washington		✓			
UC-Irvine		✓			
Colorado		✓			
Indiana		✓			

7.3 When applications invoke services, how do services authenticate the requests?

	Locally developed solution for mutual authentication	An n-tier solution such as Shibboleth ECP or CILogon	Other
UMUC			WS-Security via SAML assertions
UofT		✓	
MIT	✓		
UW-Madison	✓		
Washington		✓	
UC-Irvine	✓		
Colorado	✓		
Indiana	✓		

7.4 After requests are authenticated, do services access another service to determine what the requestor is authorized to do?

	This is not a goal	This is an architecture goal but it is only sometimes implemented	This is consistently implemented for services in the local domain	This is consistently implemented for services in the local domain and in the cloud	Other
UMUC	✓				
UofT		✓			
MIT		✓			
UW-Madison		✓			
Washington		✓			
UC-Irvine			✓		
Colorado		✓			
Indiana		✓			

7.5 More generally, how do you manage trust between distributed components ?

UofTOn an "as needed" basis. We have more work to do in this area.
MITpoint to point at the moment
UC IrvineSSL, system username/passwords, and PGP key exchange.
ColoradoCurrently application specific service accounts are created. Goal to move to Cert based AuthN and externalized AuthZ

Indiana
We manage trust between components using a combination of digitally signed web service messages using public-private key pairs and mutual trust. We also utilize oauth in certain cases when invoking services. Elsewhere we also use simple username/password authentication to services.

<< Prev Next >>