Infrastructure - Task #2478

Bug # 2411 (In Progress): knb MNs and CNs allow self-signed certificates to connect

testConnectionLayer_SelfSignedCert() failing

2012-03-13 19:53 - Ben Leinfelder

Status:	Rejected	**Start date:**	2012-03-13
Priority:	Normal	**Due date:**	
Assignee:	Ben Leinfelder	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:	Release Backlog		
Milestone:	CCI-1.0.0	**Story Points:**	
Product Version:	*		

Description

testConnectionLayer_SelfSignedCert is failing -- it assumes calling the service with a self-signed certificate will throw a ServiceFailure

History

#1 - 2012-03-13 20:02 - Ben Leinfelder

- Status changed from New to Closed

added @Ignore to the test - ping() does not take an auth parameter and should not be tested as such

#2 - 2012-03-27 16:23 - Rob Nahf

- Parent task changed from #2429 to #2411

#3 - 2012-03-27 16:24 - Rob Nahf

- Status changed from Closed to In Progress

rewrote the test to use isAuthorized() instead of ping(). Test still fails, meaning a self-signed cert currently is authorized to read private data.

#4 - 2013-03-25 23:13 - Ben Leinfelder

- Status changed from In Progress to Rejected

- translation missing: en.field_remaining_hours set to 0.0

Looking at the test, I don't believe the client using a self-signed certificate is allowed access to private data. Just because there is no ServiceFailure doesn't mean access has been granted. Only a NotAuthorized exception can conclusively tell us that.
If the Node that is being called does not trust the signer of the self-signed certificate then it will be treated as a public call. We cannot prescribe which CAs any MN chooses to trust and therefore can't really make a test to enforce this.
Perhaps the test should be written to use a *private* object so as to ensure that the access to it is not authorized to a public (e.g., self-signed certificate) user.

#5 - 2014-10-01 22:22 - Dave Vieglais

- Target version set to Release Backlog