

Infrastructure - Task #3963

Story # 1906 (Closed): Review, revise, and update architecture documentation

Remove choice in MNStorage.update() and misc other changes

2013-09-03 16:04 - Roger Dahl

Status:	Closed	Start date:	2013-09-03
Priority:	Normal	Due date:	
Assignee:	Dave Vieglais	% Done:	100%
Category:	Documentation	Estimated time:	0.00 hour
Target version:	Maintenance Backlog	Story Points:	
Milestone:	None		
Product Version:	0		

Description

The docs state:

The Member Node MUST check or set the values of SystemMetadata.obsoletes and SystemMetadata.obsoletedBy so that they accurately represent the relationship between the new and old objects. If the client sets these values and they are incorrect, then an InvalidSystemMetadata MUST be raised.

I think we are inviting bugs by leaving it up to the client to chose if obsoletes should be set in the sysmeta for the new object, because both clients and Member Nodes are likely to be tested with only one of scenarios. The docs also seem to say that it's ok for the client to set SystemMetadata.obsoletedBy, which is, of course, not the case. We should also add the requirement for non-branching update trees and state which exception should be returned if branching is attempted. So, I think it would be cleaner to specify a single way for this to work, instead of the two alternatives, and to describe it explicitly in terms of postconditions and preconditions. I think it would be ok for us to change the description on this since there are no signature or schema changes, and since we are the only ones that have developed Tier 4 nodes.

The wording could be something like:

Preconditions:
The Member Node MUST raise Exceptions.InvalidSystemMetadata if the client has set SystemMetadata.obsoletes or SystemMetadata.obsoletedBy for the new object.
The Member Node MUST raise Exceptions.InvalidRequest if SystemMetadata.obsoletedBy is already set on the old object.

Postconditions:
- SystemMetadata.obsoletes is unchanged on the old object.
- SystemMetadata.obsoletedBy on the old object is set to the PID of the new object.
- SystemMetadata.obsoletes on the new object is set to the PID of the old object.
- SystemMetadata.obsoletedBy is unset on the new object.

Roger

History

#1 - 2013-09-04 17:39 - Roger Dahl

Discussed on standup today and it was decided that the behavior of the call will not be changed (it's too late). However, we will make the docs more explicit. I'll take a stab at that.

#2 - 2013-12-13 20:15 - Roger Dahl

- Status changed from New to Closed
- translation missing: en.field_remaining_hours set to 0.0
- Product Version set to 0

#3 - 2014-10-02 17:21 - Dave Vieglais

- Target version set to Maintenance Backlog