

WBC HLD review

1. perceived benefits – lower latency, particularly on wide area networks
2. order definitions alphabetically
3. define file system object
4. define epoch
5. QAS is what?
6. Add a requirement that “grants prevent unexpected ENOSPC conditions”
7. Add a requirement that recovery will lead to well define results
8. I am not so comfortable with functional specification. Shouldn't it mention:
 - a. Fid handling
 - b. Grants
 - c. Quota mechanisms
 - d. Prototypes of exported functionality
9. Local sequentiality – for security should this include all preceding operations on any ancestors in the namespace as well?
10. Doesn't writing out data independently also introduce a security concern?
Granted, it's there also without the wbc but the window is getting bigger.
11. To avoid ongoing negative lookups, how do you transfer full directory content to the client? (use case: “make bzImage”)
12. Versioning is very important. How are versions handled, how is a partial reintegration completed? (the client needs to know the versions to which the previous integration was applied perhaps?)
13. What is changed in the llite module, or are all changes below it?
14. This solutions needs to work well on clients with many many CPUs and eliminate the disadvantages of a single threaded client
15. All exported API's must be added to the HLD in the in functional specification
16. Detailed recovery descriptions must be added; epochs is not the only use case probably (e.g. networking can fail and come back).
17. If you run out of memory locally, do you push out the changelog? Not mentioned in 4.4.
18. Note that there are many server interactions that do not require writeout, such as a lookup or getting more fid sequences.