SifriTaub Review

Naming is done in a good way, a couple of suggestions we would say are to change authenticate to authenticateUser , listBookIds to listAvailableBookIds , waitForBooks to waitForBooksByLoan and userInformation to getUserInformation (use of verb) as it will give more information and be more self-explainable. Overall, there is consistency in naming.

The names of classes and interfaces involved were very informative, and there isn't a notable naming clutter in our eyes.

Actually, signatures of the given methods pretty much guided us on how to correctly implement them, and revealed your intentions in a super clear way.

The size of the API is great – concise and clear – also well encapsulated since it is not coupled with SecureStorage.

This API passes the YAGNI test – no "generic features for future use"

Although checked exceptions are not provided, we see it as an overkill to do so if the thrown exceptions list consists of one or two exceptions, and will make the code look less attractive and clear to the user.

Library used Review (library-316278837-322252685)

We really loved the idea behind abstracting the data to be saved into "block chunks of data" in addition to the support of variable block sizes(block size is sent via a parameter which gives the user more "power") as this is a really smart and convenient solution for saving multiple types of data (and especially instances of classes that we chose to implement) based on their sizes .

We would suggest a couple of changes regarding naming and documentation, it was somewhat hard to read a documentation that is based on a different documentation of a different method, such as blocksMap's documentation in SecureStorageWrapper, it would've been clearer to write in small amount of words instead. Addition naming modifying suggestion would be to change maxFreeBlockFuture to lastWrittenBlockFuture or something like that as it will provide better view on the intentions of using it, also to change saveFreeBlockKey to lastWrittenBlockKey maybe.

The visibility of the methods in the different classes is well formed – only the minimal methods needed to the programmer are public (such s read, write…)

Regarding naming and documentation, in general I would've used more "storage systems language" in both methods or the documentation itself , it will give more information and better view on the intentions of those who wrote this :-) .

Overview – this library is written in a very intuitive way, it's easy to use and we didn't had to modify and change anything, we enjoyed working with it , thank you and great work!!