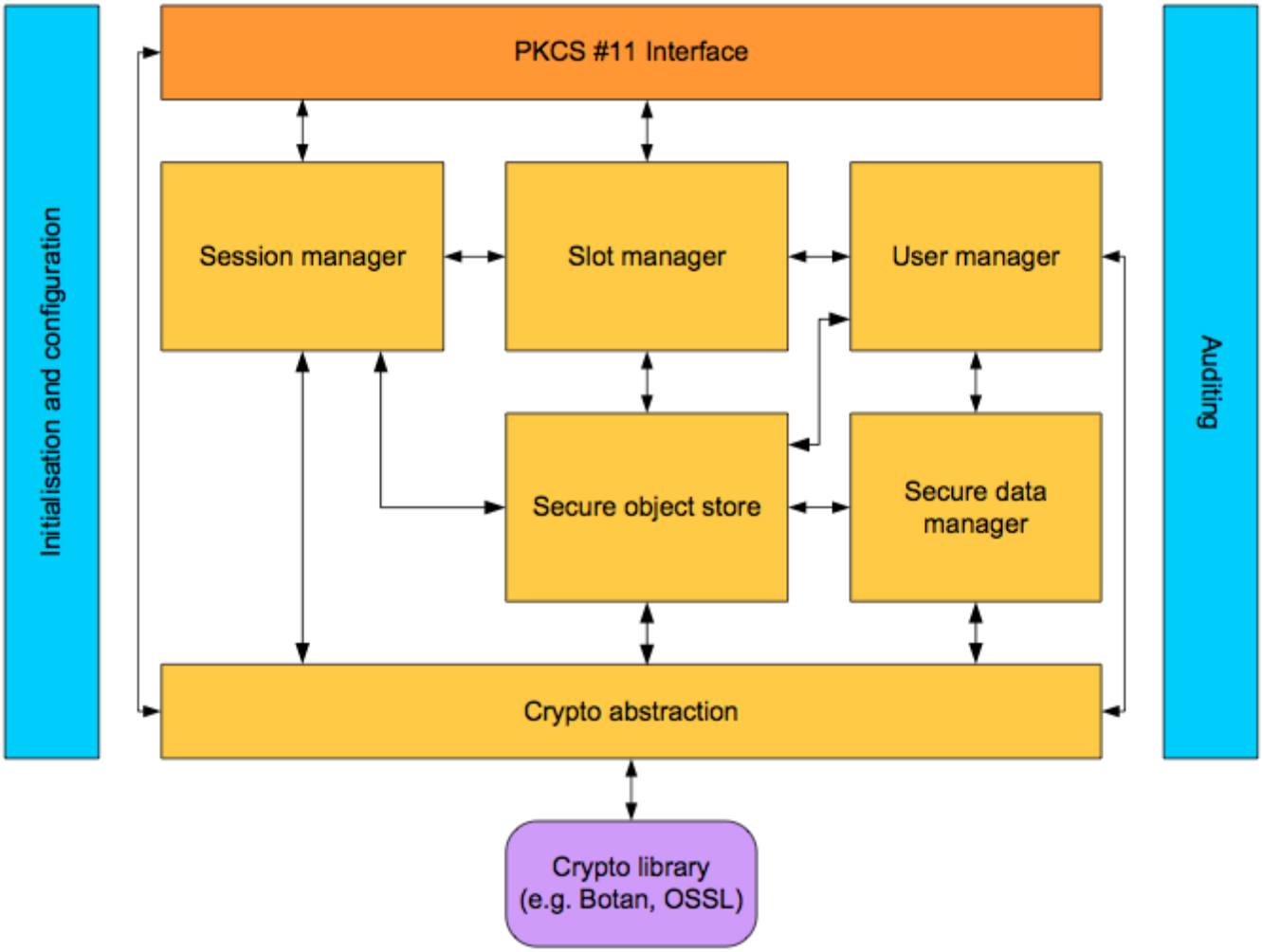
v2 Design

This section contains design specifications and design decisions w.r.t SoftHSM v2. It is currently not a complete in-depth design but rather a specification of key components, their interfaces and important design principles and decisions.

On this Page

Design overview

The diagram below shows the design overview of SoftHSM v2 and its constituent components:



Below is a description of all the components shown in the diagram.

PKCS#11 interface

The PKCS#11 interface (dark orange, at the top) is just that: an implementation of the PKCS#11 interface version 2.30. It is the only interface layer that SoftHSM v2 exposes to the outside world. All calls from outside the library enter here and are passed on to the other components that make up SoftHSM v2.

Initialisation and configuration

The initialisation and configuration component (shown in blue, at the left hand side of the diagram) is used all through SoftHSM v2 to interact with configuration files and is used during the initialisation of SoftHSM v2 when it is loaded.

Auditing

The auditing component (show in blue, at the right hand side of the diagram) is used all through SoftHSM v2 to provide logging services to other components. The logs can serve as an audit trail for the actions that have been performed by SoftHSM v2. Although the actual levels have not been decided upon yet, the auditing component will very likely provide different log-levels to allow enabling and disabling of certain log message types.

Session manager

The session manager component tracks all PKCS#11 session and the associated state. It also manages session objects.

Slot manager

The slot managers component is responsible for managing all PKCS#11 slots and their associated tokens. All tokens loaded from the secure object store configured in the SoftHSM v2 configuration file are always present in a slot. A design decision was made to always have one extra slot available that contains a blank token. Calling C_InitToken on this slot can be used to create a new token.

User manager

The user manager component is tracks the state of the PKCS#11 user credentials. These consist of the user PIN and the security officer (SO) PIN. The user manager knows per token whether or not the token is logged in.

Secure object store

The secure object store component forms the backend storage of SoftHSM v2. It stores PKCS#11 objects in a directory structure organised as follows:

- There is a top-level directory for the complete secure object store that is configurable in the SoftHSM v2 configuration file
- For each token there is a separate directory; tokens are uniquely identified using a UUID
- Inside the token directory there are separate files for each token. There is also a special file that stores token specific attributes (such as the label, the PINs, etc.)

The secure object store - as it name implies - is capable of storing sensitive attributes of an object securely using the secure data manager (see below).

Secure data manager

The secure data manager derives a key from the user PIN that is used for secure storage of sensitive object attributes. The key derived from the PIN is used to encipher/decipher a token key that is used for the actual encryption of the sensitive data. The benefit of this is that not all objects have to be re-encrypted when the user PIN changes. The token key is stored in memory during the time a token is logged in. To protect it against eavesdropping by snooping the memory of the SoftHSM v2 it is cloaked using a blob of random data that is used to XOR the actual key data. The implementation should be such that it is configurable whether or not a new blob of random data is generated every time the key is used.

Key derivation from the PIN is performed using PKCS#5 methodology for key derivation.

Crypto abstraction

The crypto abstraction forms a layer between an actual cryptogaphic library (such as Botan or OpenSSL) and the SoftHSM v2 core. This extra layer makes it possible to use different cryptographic implementations with SoftHSM v2 . For the moment, two implementations are planned, one that uses Botan and one that use OpenSSL underneath.

Design verification

To verify the design, we have performed a detailed analysis of two use cases and created the corresponding sequence diagrams:

- Use case: Initialising a token
- Use case: Using a key to sign data

The description of the use cases also shows some details on the expected internal implementation of some functions.

Interfaces

There should be clearly defined interfaces between some of the main components in the design specified above. These interfaces make it possible to break down the work on SoftHSM v2 into separate parts and facilitate unit testing. Below is a list of components with links to their interface specifications:

- Secure object store: Interface specification
- User manager: Interface specification
- Secure data manager: Interface specification
- Cryptographic abstraction: Interface specification

Module testing

SoftHSM v2 will incorporate module tests for each component at the interface level of the interfaces specified above; these tests will be implemented using the CPPunit test framework.