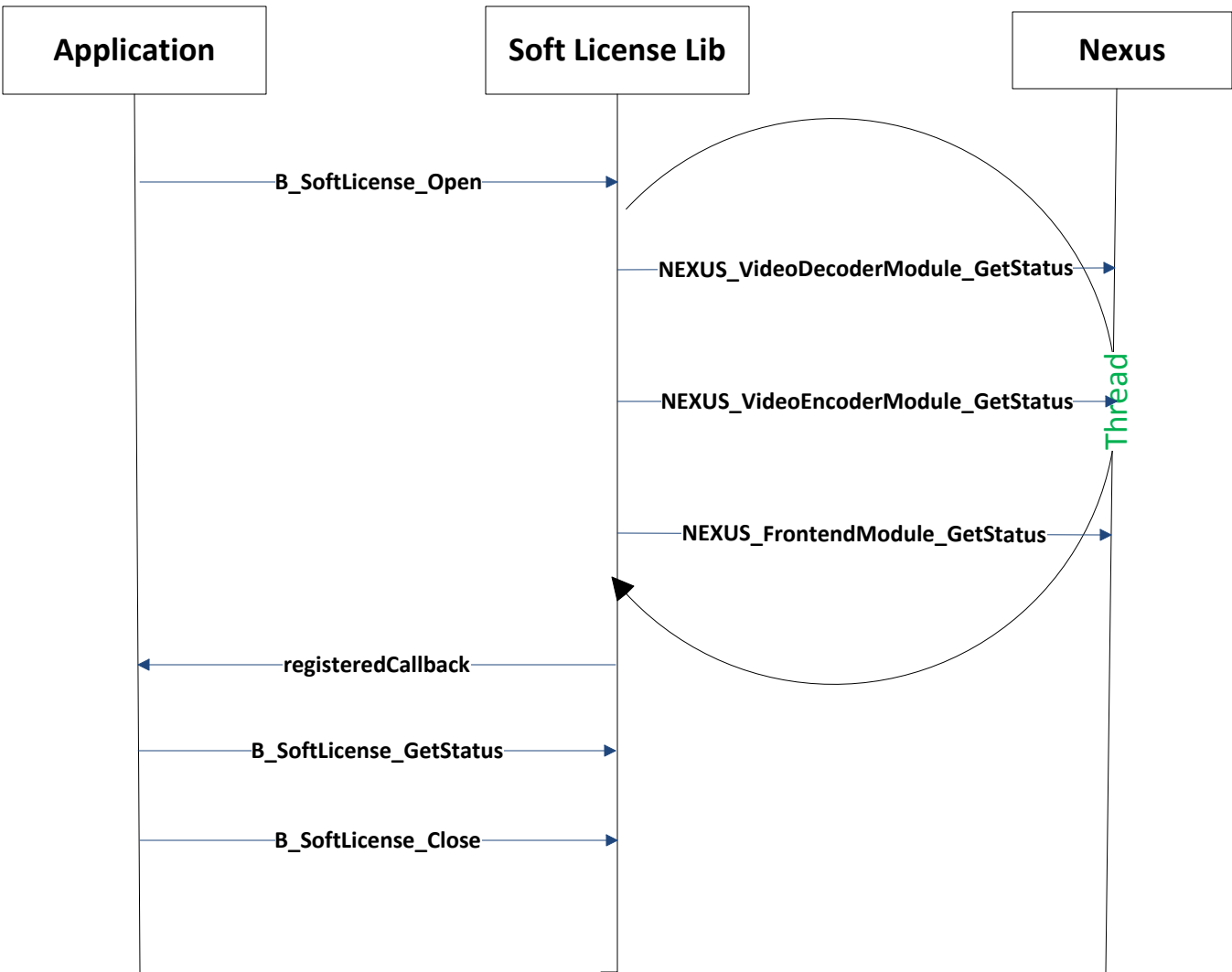


Soft Licensing Method



New Library : Soft License

Overview:

Soft License library would be a nxclient or uniprocess nexus application library. User application shall link to this library to get UHD decode, transcode and number of tuners usage through a callback it registers in B_SoftLicense_Open. This library would spin off a thread internally in B_SoftLicense_Open which will keep polling usage status from video decoders, video encoders and tuners periodically. Custom polling period can be set via B_SoftLicense_Open API in B_SoftLicense_OpenSettings. Default polling period can be obtained using B_SoftLicense_GetDefaultOpenSettings.

When the thread in soft license library detects usage of UHD decode, transcode and 8 tuners for the 1st time, the app provided registered callback will be invoked. Upon receiving the callback from library, app shall call B_SoftLicense_GetStatus to check what changed and accordingly take appropriate action like storing the usage info in a persistent storage file.

B_SoftLicense_Close can be used by the application to kill the thread spun off in B_SoftLicense_Open.

It's recommended for the app to read the persistent storage info on usage before invoking the soft license library. If the persistent storage info on usage maintained by app says all the features are used in the past, then there would be no need to check the usage statistics with the soft license library.

APIs:

```
typedef struct B_SoftLicense *B_SoftLicenseHandle;

typedef void (*B_SoftLicense_Callback)(void * appContext);

typedef struct B_SoftLicense_Status
{
    bool ultraHdDecode;
    bool transcode;
    unsigned numTuners;
}B_SoftLicense_Status;

typedef struct B_SoftLicense_OpenSettings
{
    unsigned pollingPeriod; /* in milli-seconds */
    B_SoftLicense_Callback registeredCallback;
    void * appContext;
}B_SoftLicense_OpenSettings;

void B_SoftLicense_GetDefaultOpenSettings(B_SoftLicense_OpenSettings *pOpenSettings);

B_SoftLicenseHandle B_SoftLicense_Open(B_SoftLicense_OpenSettings *pOpenSettings);

NEXUS_Error B_SoftLicense_GetStatus(B_SoftLicenseHandle handle,B_SoftLicense_Status *pStatus);

NEXUS_Error B_SoftLicense_Close(B_SoftLicenseHandle handle);
```

New Nexus APIs

```
void NEXUS_VideoDecoderModule_GetStatus(NEXUS_VideoDecoderModuleStatus *pStatus );

void NEXUS_VideoEncoderModule_GetStatus(NEXUS_VideoEncoderModuleStatus *pStatus);

void NEXUS_FrontendModule_GetStatus(NEXUS_FrontendModuleStatus *pStatus );
```