SES
Platform Services

Media Platform Solutions | AS-20501

Novaspread-S

Version:

0.53

Date issued:

23 August 2015

STRICTLY CONFIDENTIAL

Important Notice

This document has been produced by SES Platform Services GmbH (SES PS). Certain product names or brand names may be trademarks or designations of their respective owners.

Liability/Copyright

SES Platform Services GmbH

Beta Straße 1-10

D-85774 Unterföhring

Germany

Cooperation

This document has been developed in cooperation with:

TARA Systems GmbH

Gmunder Str. 53

81379 München

Germany

SES
Platform Services

TABLE OF CONTENT

1 Introduction 5

1.1 Purpose of document 5
1.2 Document history 5
1.3 References 5
1.4 Terminology 5

2 Scope 6

2.1 Context 6
2.2 Scope of document 6
2.3 Features and functions of Novaspread-S 6

3 Architecture 8

3.1 Overall architecture 8
3.2 Overview on Multiscreen-Server 8
3.3 Overview on Frontend-Manager 8
3.4 Provided interface 9
3.4.1 Overview 9
3.4.2 Required usage of provided interface 10
3.4.3 Details on CASS information 10
3.4.4 Management of SAT>IP servers 11
3.5 Required interface 12
3.5.1 Overview 12
3.5.2 OS interface 13
3.5.3 Host interface and Tuner interface 13
3.5.4 CA interface 16
3.5.5 DRM interface 16
3.5.6 System interface 17
3.5.7 Control interface 17
3.5.8 HbbTV interface 17
3.6 Open source libraries 17
3.7 Configuration data 17
3.7.1 Path 17
3.7.2 Device description 17
3.7.3 Icons 17

4 Integration 18

4.1 Integration of CASS and DRM-S 18
4.2 Integration of Novaspread-S 18
4.2.1 Requirements on Multiscreen-Servers 18
4.2.2 Interprocess communication 18
4.3 Porting process for Novaspread-S 18
4.3.1 Overview 18
4.3.2 Platform adaptation 18
4.3.3 Software integration 19

1 INTRODUCTION

1.1 PURPOSE OF DOCUMENT

This document describes the architecture and porting process of the software component Novaspread-S which is used for the Multiscreen product of SES Platform Services (SPS).

1.2 DOCUMENT HISTORY

Version	Date	Author	Changes
0.53	2015-08-23	Harald Molina-Tillmann	Working draft

1.3 REFERENCES

[1] SPS; "AS-20001: Multiscreen”
[2] SPS; "AS-20502: Novaspread-S: Reference manual"
[3] SPS; "AS-20503: Novaspread-S: OS interface"
[4] SES; "SAT>IP Operator Extensions"; v0.9.2

1.4 TERMINOLOGY

Abbreviations, acronyms, expressions, and notations are defined in document [1].

2 SCOPE

2.1 CONTEXT

Novaspread is a software package that implements essential features which are required in the scope of the Multiscreen product of SES Platform Services (SPS). Multiscreen and Novaspread are introduced in document [1].

The server component of Novaspread (referred to as Novaspread-S) must be integrated in Multiscreen-Server devices.

2.2 SCOPE OF DOCUMENT

This document covers the following:

- Features and functions of Novaspread-S
- Architecture of Novaspread-S as part of a Multiscreen-Server
- Overview on the interface provided by Novaspread-S
- Overview on the interface that must be implemented by the Manufacturer of a Multiscreen-Server
- Functionalities which must be implemented by the Manufacturer of a Multiscreen-Server
- Implementation guidelines
- Porting process

The interfaces are defined in detail in the reference manuals (documents [2] and [3]).

2.3 FEATURES AND FUNCTIONS OF NOVASPREAD-S

Streaming client

- Scan home network for SAT>IP server
- Provide virtual tuners for accessing SAT>IP LNBs
- Unpack incoming RTP streams and feed the transport stream to the middleware

Streaming server

- Advertise the Multiscreen service in the home network
- Manage incoming RTSP requests according to SAT>IP
- Restrict outgoing streams by evaluating broadcasted usage rules
- Provide tuning data, PIDs, transcoding data, and transcryption data to middleware
- Solve hardware resource conflicts between several Multiscreen-Clients
- Configure and initiate license generation
- Manage transport stream delivered by middleware
- Send RTP stream
- Send signal info via RTCP

Data server

- Prevent unauthorized access to the Multiscreen service by using TLS client authentication
- Manage pairing of Multiscreen-Server and Multiscreen-Client
- Provide list of events in regard to the Multiscreen-Server to the Multiscreen-App
- Manage incoming HTTP/HTTPS requests
- Transfer license and other data to Multiscreen-Client
- Forward simulated remote control events to middleware
- Forward remote control commands to middleware

Implementation of the OS interface for Linux (optional)

- » Manage IP sockets
- » Provide system time and timer
- » Manage files
- » Provide memory allocation
- » Support for multi-threading (Thread, Mutex, Semaphore)

3 ARCHITECTURE

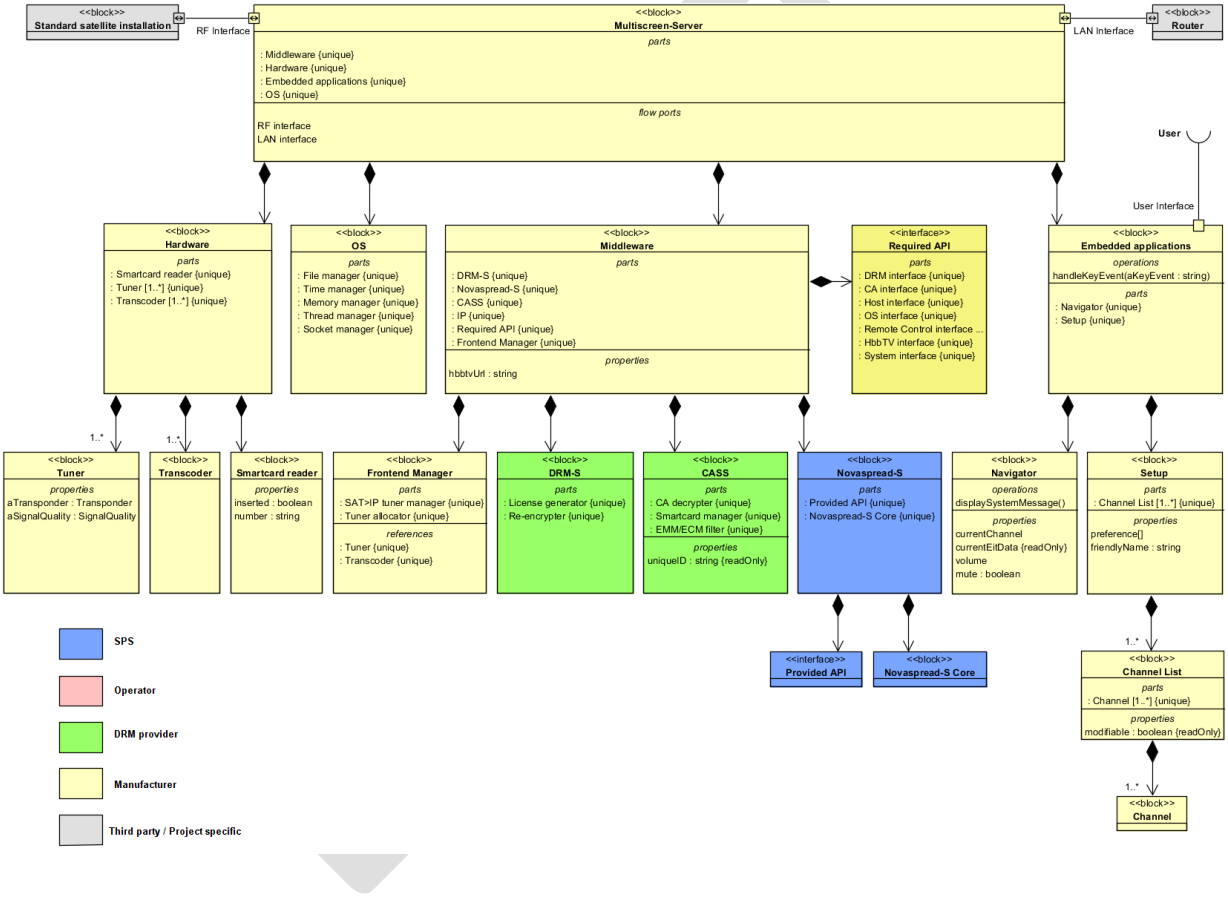
3.1 OVERALL ARCHITECTURE

The overall architecture of Multiscreen is described in document [1].

3.2 OVERVIEW ON MULTISCREEN-SERVER

The following block definition diagram summarizes the components which must be integrated in the Multiscreen-Server.

Note 1: The light yellow blocks are to be understood as hardware components or functional components. The internal architecture and technical implementation is not defined here and is solely at the discretion of the Manufacturer.



Detailed and further requirements on a reference Multiscreen-Server are defined in Annex A of document [1].

3.3 OVERVIEW ON FRONTEND-MANAGER

The Frontend-Manager of the Multiscreen-Server implements several functions which are required for Multiscreen streaming services:

- » Management of hardware resource conflicts as defined in document [1] and in section 3.5.3.2 of the document at hand
- » Usage of SAT>IP streams provided by Novaspread-S (see section 3.4.4)
- » Adaptation of PSI/SI tables in the scope of transcoding and transcrypting as defined in document [4]

3.4 PROVIDED INTERFACE

3.4.1 OVERVIEW

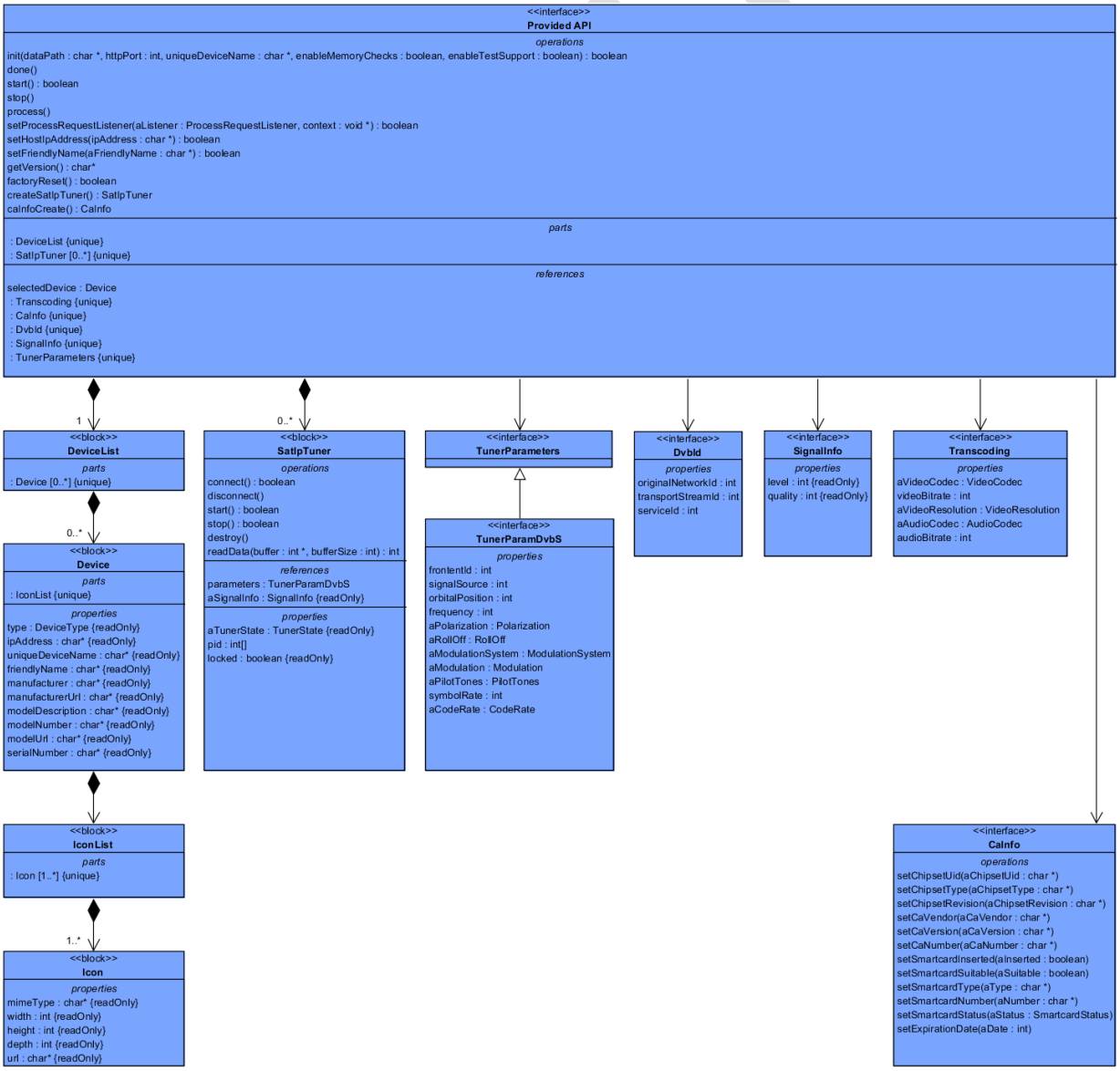
Novaspread-S provides an interface that is accessed by the middleware. This interface is introduced in this section and specified in detail in document [2].

The following topics are covered:

- Controlling the life cycle of Novaspread-S
- Management of SAT>IP server devices in the home network
- Provision of virtual tuners to get data from a connected SAT>IP LNB
- Configuration of items for information on the CASS and an inserted smartcard

Furthermore, data structures used by the required interface (see section 3.5) are provided.

The following diagram illustrates the general architecture and the most crucial functions and properties of the provided interface.



3.4.2 REQUIRED USAGE OF PROVIDED INTERFACE

Trigger	Functions to be called

Boot process finished	`NovaspreadServerInit()`
	`NovaspreadServerSetFriendlyName()`
	`NovaspreadServerSetProcessRequestListener()`
	`NovaspreadServerSetHostIpAddress()` (optional)
	`NovaspreadServerStart()`

Home network access method has changed (LAN / WLAN) OR Multiscreen-Server got new address from DHCP server	`NovaspreadServerStop()`
	`NovaspreadServerSetHostIpAddress()`
	`NovaspreadServerStart()`

User or system has initiated a factory reset	`NovaspreadServerFactoryReset()`

User has changed friendly name of Multiscreen-Server	`NovaspreadServerStop()`
	`NovaspreadServerSetFriendlyName()`
	`NovaspreadServerStart()`

ProcessRequestListener called by Novaspread-S	`NovaspreadServerProcess()`
`NovaspreadCaGetInfo()` called by Novaspread-S	`NovaspreadCaInfoCreate()` *all functions to configure the CaInfo (see below)*
Shut-down initiated	`NovaspreadServerStop()` `NovaspreadServerDone()`

3.4.3 DETAILS ON CASS INFORMATION

The required interface (see section 3.5) includes the function `NovaspreadCaGetInfo()`. If this method is called, the middleware must create a `CaInfo` data item using the provided method `NovaspreadCaInfoCreate()`. The returned `CaInfo` must then be configured using the following provided functions:

```
NovaspreadCaInfoSetChipsetUid()
NovaspreadCaInfoSetChipsetType()
NovaspreadCaInfoSetChipsetRevision()
NovaspreadCaInfoSetCaVendor()
NovaspreadCaInfoSetCaVersion()
NovaspreadCaInfoSetCaNumber()
NovaspreadCaInfoSetSmartcardInserted()
NovaspreadCaInfoSetSmartcardSuitable()
NovaspreadCaInfoSetSmartcardType()
NovaspreadCaInfoSetSmartcardNumber()
NovaspreadCaInfoSetSmartcardStatus()
NovaspreadCaInfoSetExpirationDate()
```

3.4.4 MANAGEMENT OF SAT>IP SERVERS

As soon as Novaspread-S is initialized, it starts scanning the home network for SAT>IP servers. The following should be implemented by the Manufacturer:

- » Retrieve the current list of found devices
- » Retrieve the data of all devices in the list
- » Optional: Provide a user dialogue to select one of the found devices
- » Connect to one of the found devices

Once a SAT>IP server device has been selected, the middleware should use it as follows:

- » Create a virtual SAT>IP tuner
- » Connect the tuner to the currently selected SAT>IP server
- » Set physical tuning data and the IDs of the required elementary streams (PIDs)
- » Start tuner
- » Regularly read and process the data provided by the SAT>IP tuner

SAT>IP tuners can be used for both Multiscreen services and standard services of the Multiscreen-Server.

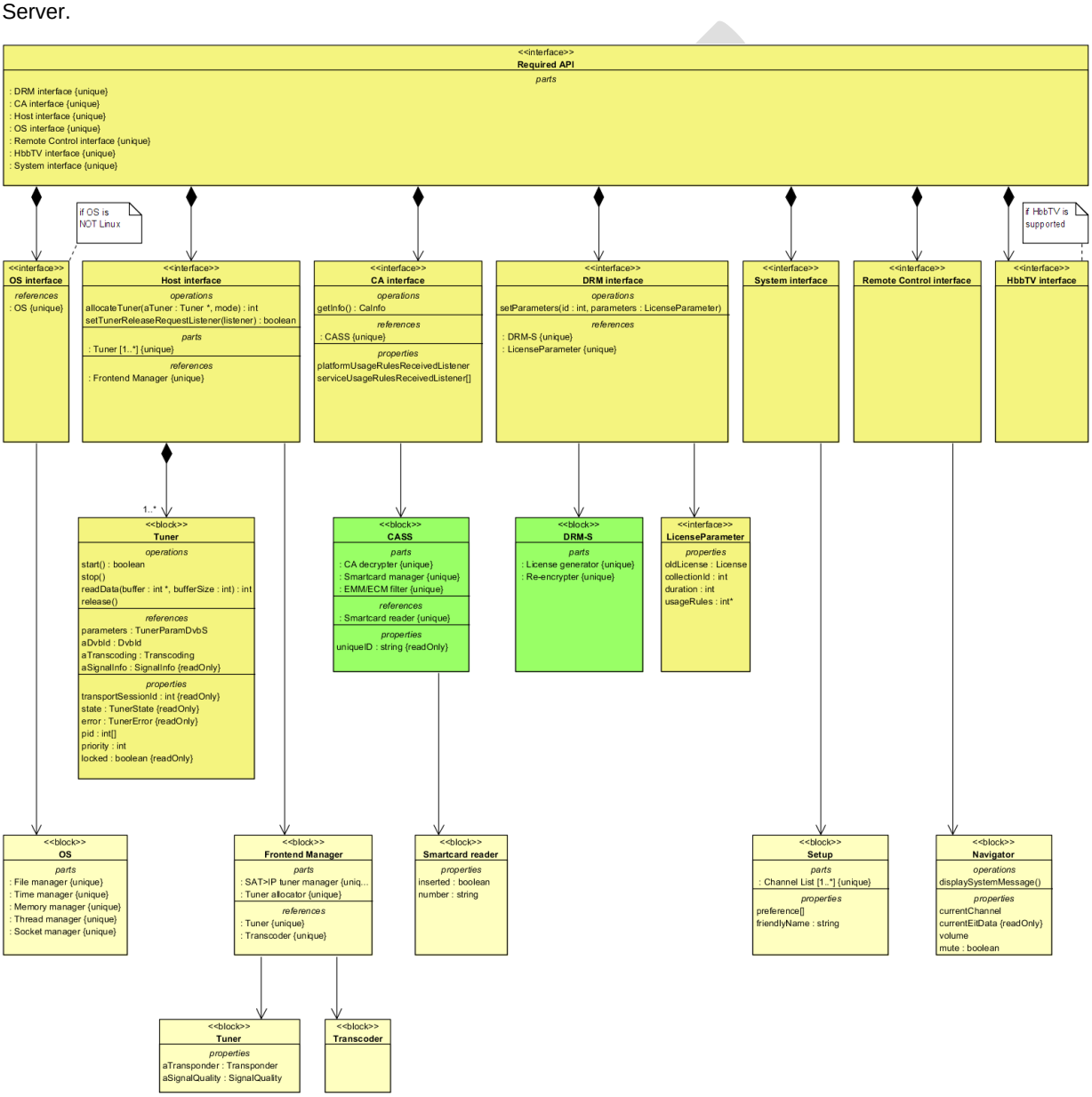
3.5 REQUIRED INTERFACE

3.5.1 OVERVIEW

The integration of Novaspread-S requires the implementation of several interfaces. These interfaces are introduced in this section and specified in detail in the following reference manuals:

» OS interface: AS-20503 (document [3])
» All other interfaces: AS-20502 (document [2])

The following diagram illustrates the general architecture and the most crucial functions and properties of the required interface. It also shows the dependencies on other software and hardware blocks of the Multiscreen-Server.



3.5.2 OS INTERFACE

This interface covers several basic functions of an operating system. It is specified in document [3].

The following topics are covered:

- Memory management
- File management
- System time and timer
- Socket management
- Multi-thread management

For platforms based on **LINUX** these functions are already implemented and provided as an additional module of Novaspread-S and **NEED NOT** to be implemented by the Manufacturer. If the Manufacturer does not want to use the provided implementation, or if the platform is based on a different operating system, the OS interface must be implemented by the Manufacturer itself.

3.5.3 HOST INTERFACE AND TUNER INTERFACE

3.5.3.1 Overview

These interfaces provide access to the frontend manager.

The following topics are covered:

- Tuner allocation
- Setting up transcoding
- Reading signal quality
- Reading transport streams

3.5.3.2 Tuner management

Concept

The virtual tuners of Novaspread-S and the concept for solving hardware resource conflicts are described in document [1].

Initialisation

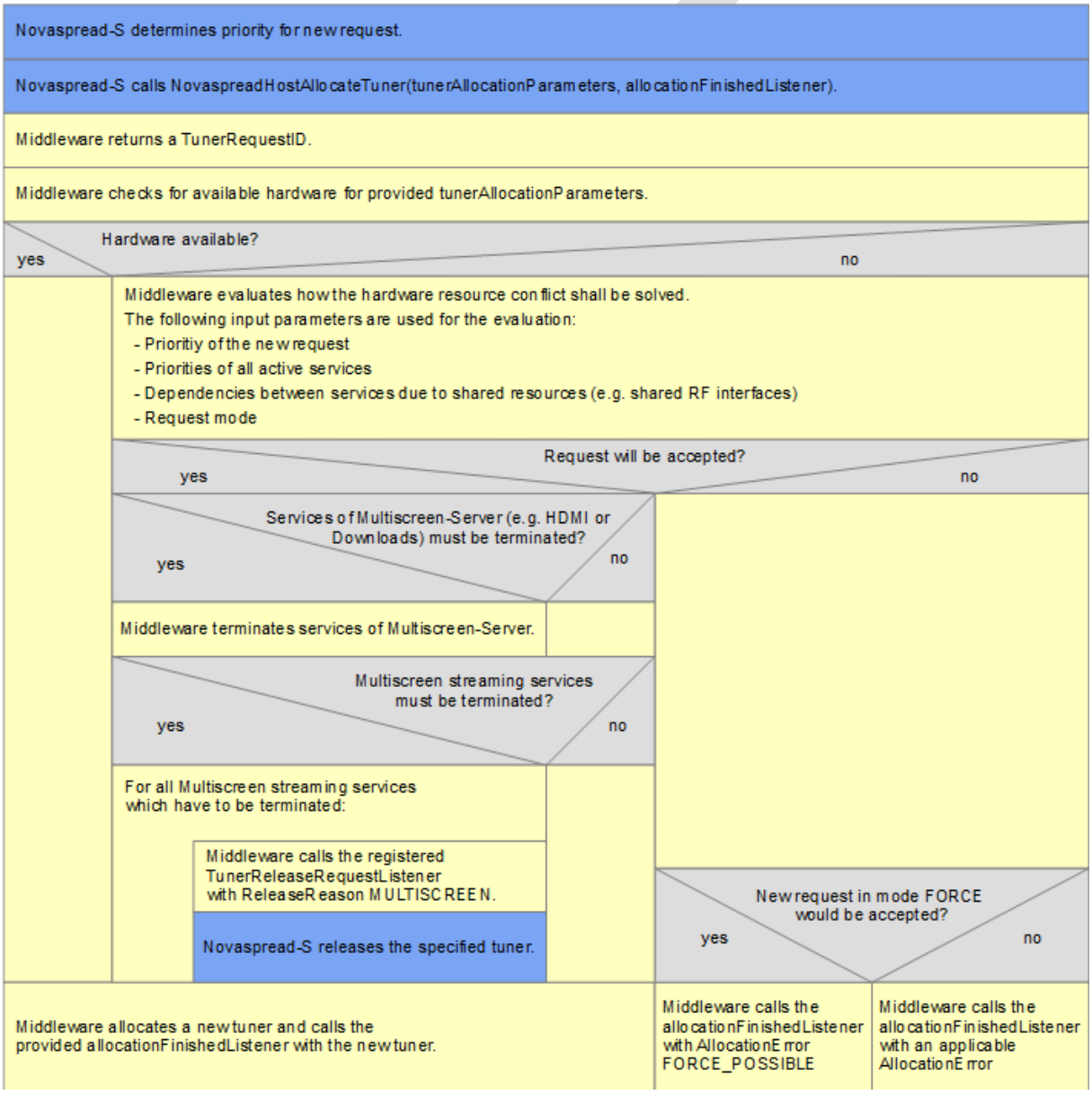
As part of the initialisation process, Novaspread-S registers a callback function for tuner release requests by calling the required function `NovaspreadHostSetTunerReleaseRequestListener()`. The middleware calls this function if virtual tuners of Novaspread-S must be released due to a hardware resource conflict.

Request for Multiscreen stream

If Novaspread-S receives a stream request from a Multiscreen-Client, it tries to allocate a tuner from the middleware by calling the required function `NovaspreadHostAllocateTuner()`. For this, Novaspread-S provides the following:

- Physical tuning data
- Demand for transcoders and transcryptors
- Priority
- Request mode
- Function to be called once the allocation is finished

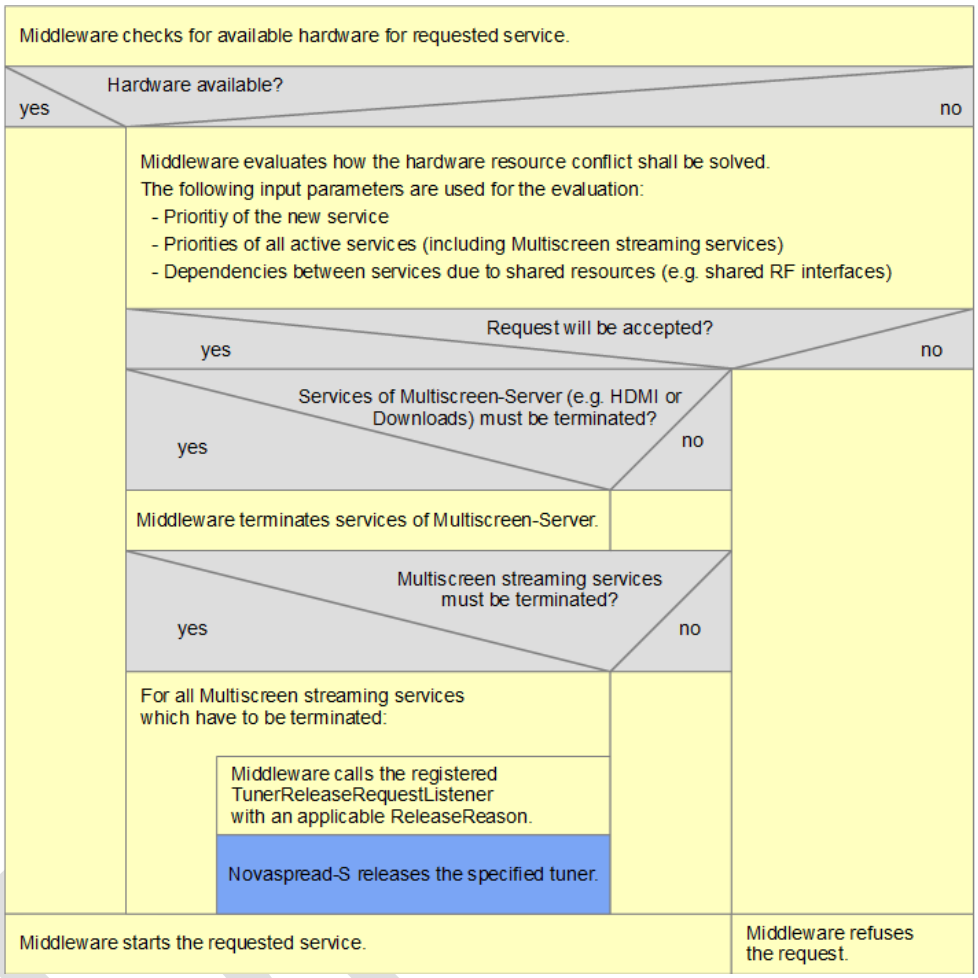
The following structogram shows how a tuner allocation and a possible hardware resource conflict shall be handled by the middleware and Novaspread-S:



Novaspread-S may cancel an unfinished tuner allocation process by calling the required function `NovaspreadHostCancelAllocation(requestId)`.

Request for service of Multiscreen-Server

The following structogram shows how the middleware shall handle a possible hardware resource conflict if a service of the Multiscreen-Server itself is requested or is going to start (e.g. new TV programme selected for HDMI output or programmed recording starting soon). The method follows the same concept as for managing a new request for a Multiscreen stream.



Releasing unused tuners

Novaspread-S is responsible for releasing tuners which are not longer used by calling the required function `NovaspreadTunerRelease()`.

3.5.3.3 Streaming

Once a tuner is allocated, the PIDs are set by calling `NovaspreadTunerSetPids()`. Streaming is started by calling `NovaspreadTunerStart()`. The middleware must then provide transport stream packets which are read by Novaspread-S by calling `NovaspreadTunerReadData()`. If the delivery of transport stream packets has been interrupted (i.e. `NovaspreadTunerReadData()` has returned 0), the middleware must call a registered `DataAvailableListener` function as soon as new packets become available.

3.5.3.4 Transcoding

Novaspread-S can request a transcoding for an allocated tuner by calling `NovaspreadTunerSetTranscoding()`.

If transcoding is requested, the middleware must meet the requirements on "PSI/SI usage and adaptations for transcoded services" as defined in document [4].

3.5.4 CA INTERFACE

This interface provides communication with the conditional access subsystem (CASS).

CASS and smartcard information

The middleware must implement the function `NovaspreadCaGetInfo()` to return information on the CASS and the inserted smartcard. This function itself creates and configures the required data item by using functions of the provided interface as described in section 3.4.3.

Usage rules

The Operator can restrict the number of concurrent streams using broadcasted usage rules (see document [1]). To evaluate this, Novaspread-S registers listener methods which must be called by the middleware if the usage rules have changed.

3.5.5 DRM INTERFACE

This interface provides access to DRM-S.

The following topics are covered:

- Control re-encryption sessions
- Manage licenses

The middleware must implement the function `NovaspreadDrmSetParameters()` to configure re-encryption. The function requires the following parameters:

- Session ID (to associate the re-encryption with a tuner)
- Collection ID /which is required for DRM-S)
- Old license (if available to check if this old license is still valid)
- Requested duration of the new license
- Usage rules

The function returns the new license, which Novaspread-S will send to the requesting Multiscreen-Client.

If re-encryption is requested, the middleware must meet the requirements on "PSI/SI usage and adaptations for transcrypted services" as defined in document [4].

3.5.6 SYSTEM INTERFACE

This interface provides access to system information and user setup data.

3.5.7 CONTROL INTERFACE

This interface provides functions which allow Multiscreen-Clients to control the Multiscreen-Server.

3.5.8 HBBTV INTERFACE

If the Multiscreen-Server includes HbbTV, this interface is used to launch HbbTV applications.

This interface **NEED NOT** to be implemented if the Multiscreen-Server does not include an HbbTV compliant application framework.

3.6 OPEN SOURCE LIBRARIES

Novaspread-S uses the following open source libraries:

- OpenSSL
- libupnp

3.7 CONFIGURATION DATA

3.7.1 PATH

Novaspread-S requires a path for storing and reading configuration files inside the local file system of the Multiscreen-Server. This path is set by the Multiscreen-Server using the function `NovaspreadServerInit()`.

Within the path Novaspread-S expects to find configuration files for the device description and for icons as defined in the following sections.

Novaspread-S uses the path to store and read dynamic configuration data, too.

3.7.2 DEVICE DESCRIPTION

Within the path Novaspread-S expects to find a file named "deviceDescription.xml".

Novaspread-S uses this file as starting point to build the actual device description used for UPnP advertisement.

(The format of the file will be described in the next version of this document.)

3.7.3 ICONS

Within the path Novaspread-S expects to find a file named "icons.xml". This file defines the location and the dimension of icons that are provided for UPnP advertisement.

(The format of the file will be described in the next version of this document.)

The image files are stored in the local file system.

4 INTEGRATION

4.1 INTEGRATION OF CASS AND DRM-S

The integration of the CASS and DRM-S is defined by the DRM-Provider and out of scope of this document.

Some code is required to map the API of the CASS to the CA interface defined in this document. In the same way some code is required to map the API of DRM-S to the DRM interface defined in this document. SPS provides sample source code and support for this mapping in cooperation with the DRM-Provider. However, the Manufacturer is responsible for adaptation and integration and testing.

4.2 INTEGRATION OF NOVASPREAD-S

4.2.1 REQUIREMENTS ON MULTISCREEN-SERVERS

Novaspread-S can be ported on a wide range of DVB-S/S2 compliant receivers. It is independent of the concrete chipset, operating system, and implemented features.

The reference Multiscreen-Server is defined in Annex A of document [1].

4.2.2 INTERPROCESS COMMUNICATION

Novaspread-S communicates with connected clients using different open protocols like HTTP and RTP/UDP. Depending on the DRM-Provider there may be a security requirement to run Novaspread-S in a separate process. In this case the functions of Novaspread-S must be called using some kind of interprocess communication (IPC). IPC as such is not part of Novaspread-S. The Manufacturer is responsible to fulfil the security requirements of the DRM-Provider using their own IPC infrastructure if needed.

4.3 PORTING PROCESS FOR NOVASPREAD-S

4.3.1 OVERVIEW

The porting process comprises three parts:

- Platform adaptation which is driven by SPS
- Software integration which is driven by the Manufacturer
- End-to-end test

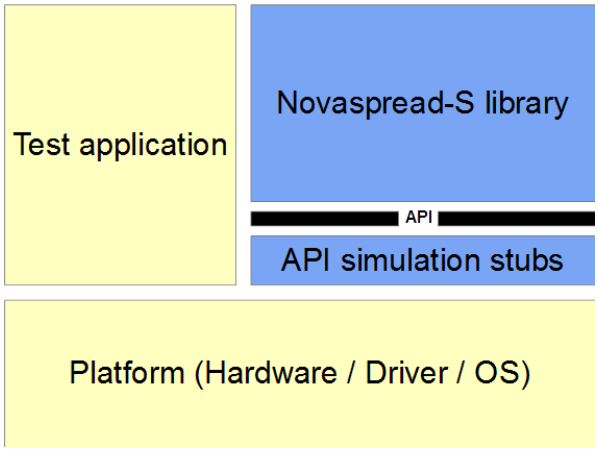
Note 2: Details and contractual provisions are out of scope of this document.

4.3.2 PLATFORM ADAPTATION

For this process the Manufacturer must deliver an environment to test Novaspread-S on its platform. Usually that includes:

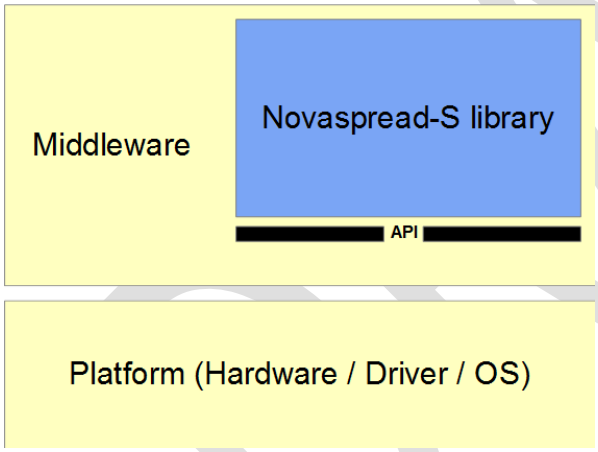
- Samples of the platform including hardware, driver, and operating system
- Possibility to mount an external file system (e.g. via NFS)
- Tool chain to compile the software
- Simple example application
- Description of the platform
- Additional platform requirements (e.g. in regards to interprocess communication)
- Required documentation

SPS will test the Novaspread-S library on the platform by using stubs which simulate the implementation of the required interface defined in this document.



4.3.3 SOFTWARE INTEGRATION

Once the library has been validated on the platform, the Manufacturer integrates this very library into its own middleware. Novaspread-S will interact with the middleware using functions of the middleware via the same required interface.



SPS delivers:

- Tested libraries (including Include files and third-party libraries)
- API Simulation stubs (if requested by Manufacturer)
- Demonstration application for the target platform as source code and as binary file (including build environment and Makefiles)
- Unit test results and code coverage
- Documentation and reference manuals

Published by:

SES Platform Services GmbH

Beta Straße 1-10

85774 Unterföhring

Germany

For more information about SES, visit

www.ses-ps.com or email **info@ses-ps.com**