Media Platform Solutions | AS-20501

Novaspread-S

DRAFT

Version:

1.0

Date issued:

31 May 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

Cooperation

This document has been developed in cooperation with:

TARA Systems GmbH

Gmunder Str. 53

81379 München

Germany

TABLE OF CONTENT

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
1.0	2015-05-31	Harald Molina-Tillmann	Working release
		Manfred Schmidt	
		Georg Kamjunke	

1.3   REFERENCES

[1]   SPS; "AS-20001: Multiscreen"; v1.0
[2]   SPS; "AS-20502: Novaspread-S: Reference manual"

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
- » Porting process

The interfaces are defined in detail in document [2].

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

Linux library

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

Note 1: If the Multiscreen-Server is based on a different operating system, the functions of the Linux library must be implemented by the Manufacturer according to the OS interface defined in document [2].

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 2: 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.
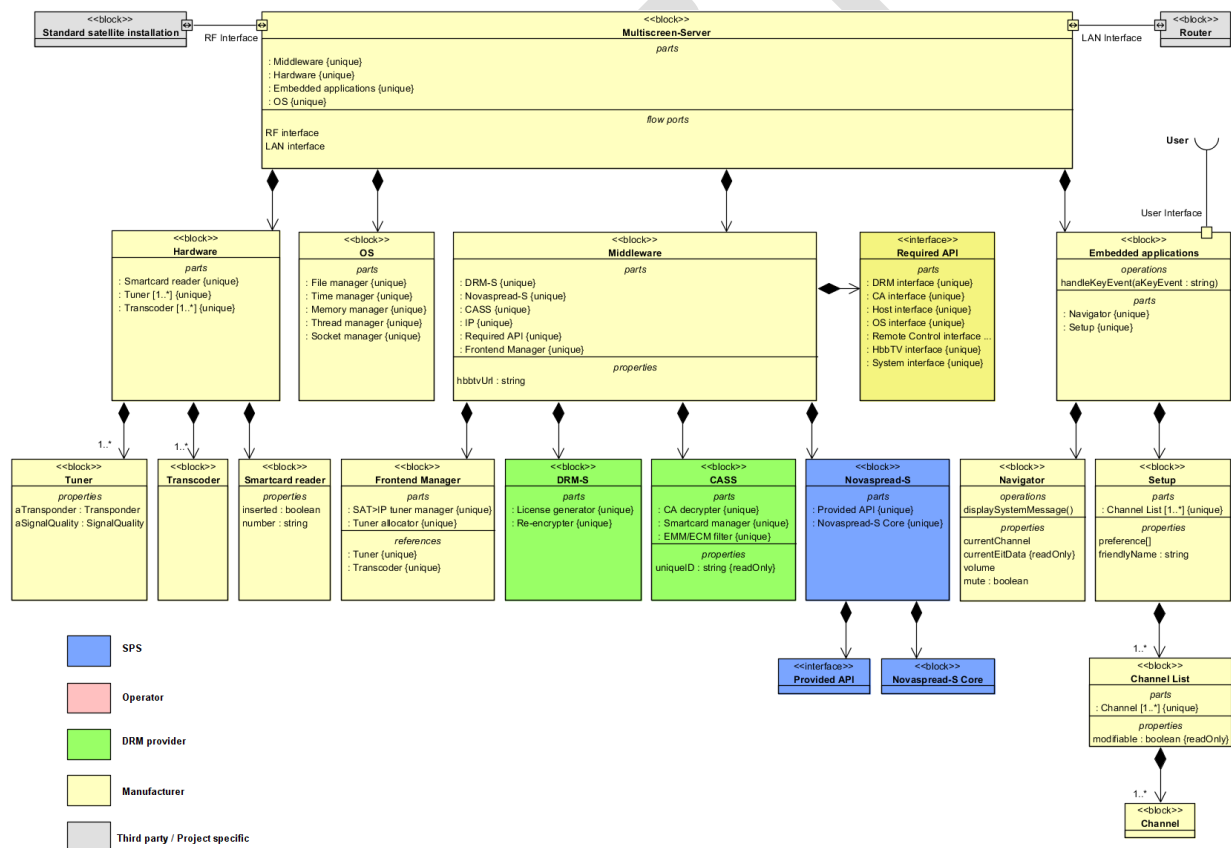
Note 3: Detailed and further requirements on a reference Multiscreen-Server are defined in Annex A of document [1].

3.3   PROVIDED API

3.3.1   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

Furthermore, data structures used by the required API (see section 3.4) are provided.

The following diagram illustrates the general architecture and the most crucial functions and properties of the provided API.
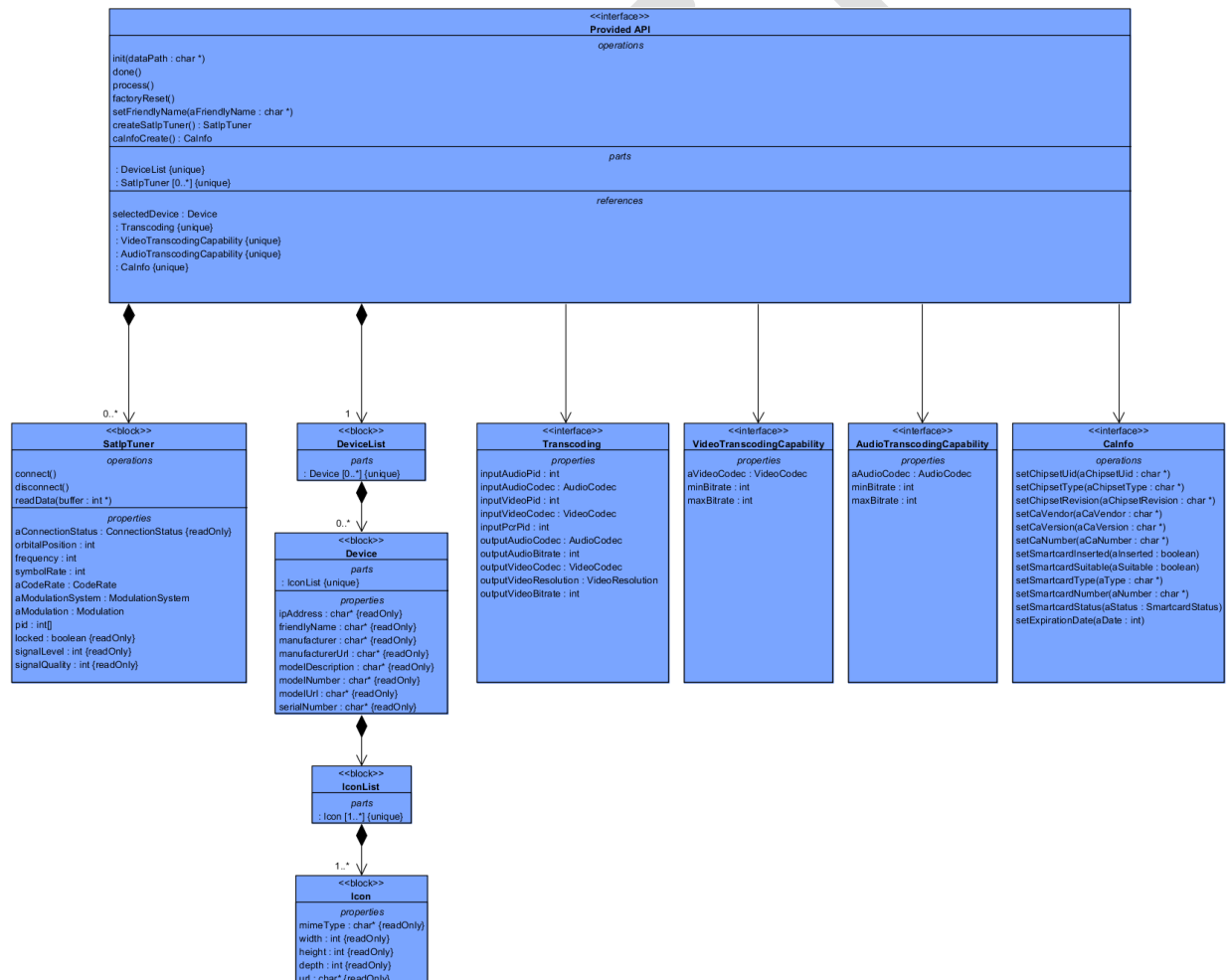
3.3.2 DETAILS

3.3.2.1 Controlling Novaspread-S

The middleware must initialise and shutdown and regularly update Novaspread-S.

3.3.2.2 Factory reset

As part of a factory reset, the middleware must inform Novaspread-S.

3.3.2.3 Friendly name

In two cases the middleware must send the friendly name of the Multiscreen-Server to Novaspread-S:

» As part of the boot process
» After the user has changed the friendly name in the setup menu

3.3.2.4 Management of SAT>IP server devices

As soon as Novaspread-S is initialized, it starts scanning the home network for SAT>IP servers. The following can 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

3.3.2.5 Retrieval of SAT>IP tuners

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

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

3.4 REQUIRED API

3.4.1 OVERVIEW

The integration of Novaspread-S requires the implementation of an interface. This interface is introduced in this section and specified in detail in document [2].

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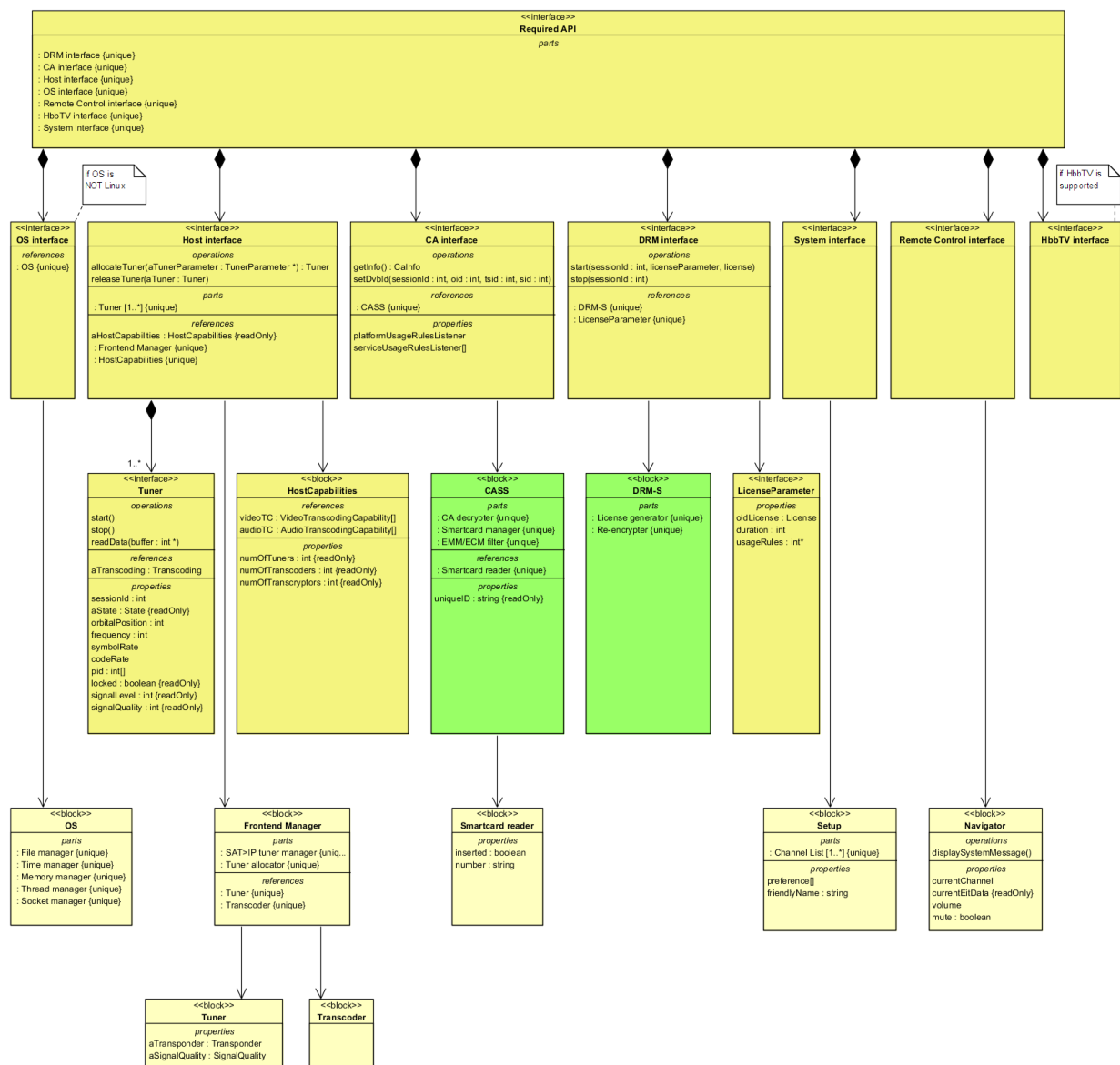
3.4.2 OS INTERFACE

This interface covers several basic functions of an operating system.

For platforms based on **LINUX** these functions are already integrated in Novaspread-S and **NEED NOT** to be implemented by the Manufacturer.

The following topics are covered:

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

3.4.3 HOST INTERFACE AND TUNER INTERFACE

These interfaces provide access to the frontend manager.

The following topics are covered:

- » Hardware capabilities
- » Tuner management
- » Reading transport streams

Capabilities

The middleware must provide its capabilities in regards to reception and transcoding and transcryption.

Hardware allocation

If Novaspread-S receives a stream request from a Multiscreen-Client, it tries to allocate a tuner of the Multiscreen-Server. The middleware must solve a potential hardware conflict as described in document [1]. After that the middleware must inform Novaspread-S using the specified listener method.

Transcoding

A Transcoding (as defined in the provided API) can be assigned to an allocated tuner.

Signal quality

A tuner provides the quality of the DVB signal. Novaspread-S sends this information to connected Multiscreen-Clients.

Streaming

To start streaming, Novaspread-S does the following:

- » Set the IDs of the required elementary streams (PIDs)
- » Start the tuner
- » Read regularly transport stream packets from the tuner

Session ID

Each tuner provides a Session ID which is needed for decryption and re-encryption (see next sections).

3.4.4 CA INTERFACE

This interface provides access to functions of the conditional access subsystem (CASS).

The following topics are covered:

- » CASS and smartcard information
- » Association of tuners with DVB triplets
- » Usage rules

CASS and smartcard information

The middleware must provide a function to return information on the CASS and the inserted smartcard. This function itself can create the required data item by using a function of the API provided by Novaspread-S.

DVB triplet and Session ID

The CASS needs the DVB triplet of the requested service to locate the EMM and ECM of this very service. Therefore, the CA interface includes a function to associate a session ID with a DVB triplet. The Session ID again is associated with a tuner as provided in the Host interface.

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 if the usage rules have changed.

3.4.5 DRM INTERFACE

This interface provides access to DRM-S.

The following topics are covered:

- » Control re-encryption sessions
- » Manage licenses

Re-encryption

The middleware must provide a function to start re-encryption. The function requires the following parameters:

- » Session ID (to associate the re-encryption with a tuner)
- » 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.

3.4.6   SYSTEM INTERFACE

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

3.4.7   CONTROL INTERFACE

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

3.4.8   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.

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

Note 4: Requirements on the interprocess communication must be discussed with the Manufacturer.

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 5: 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 API defined in this document.
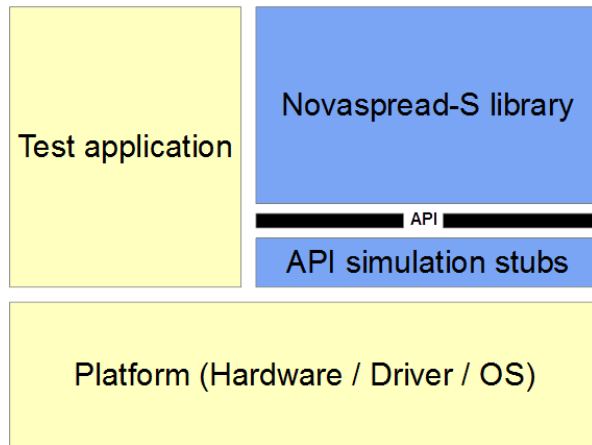
4.3.3   SOFTWARE INTEGRATION

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 API.
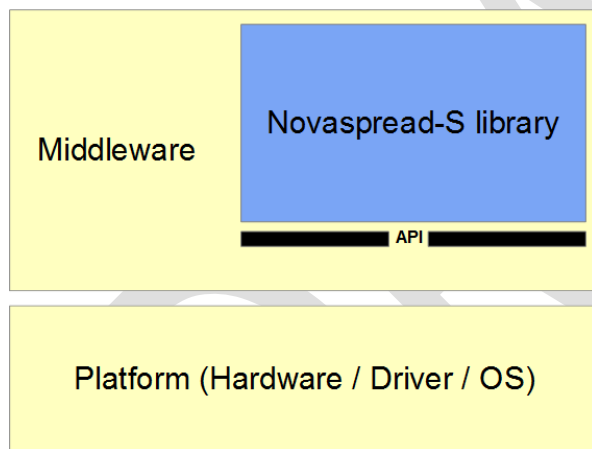
SPS delivers:

- » Tested library
- » API Simulation stubs (if requested by Manufacturer)
- » Reference manual (document [2])