Perfect Streamer

Release 2.0.0.206

Perfect Soft

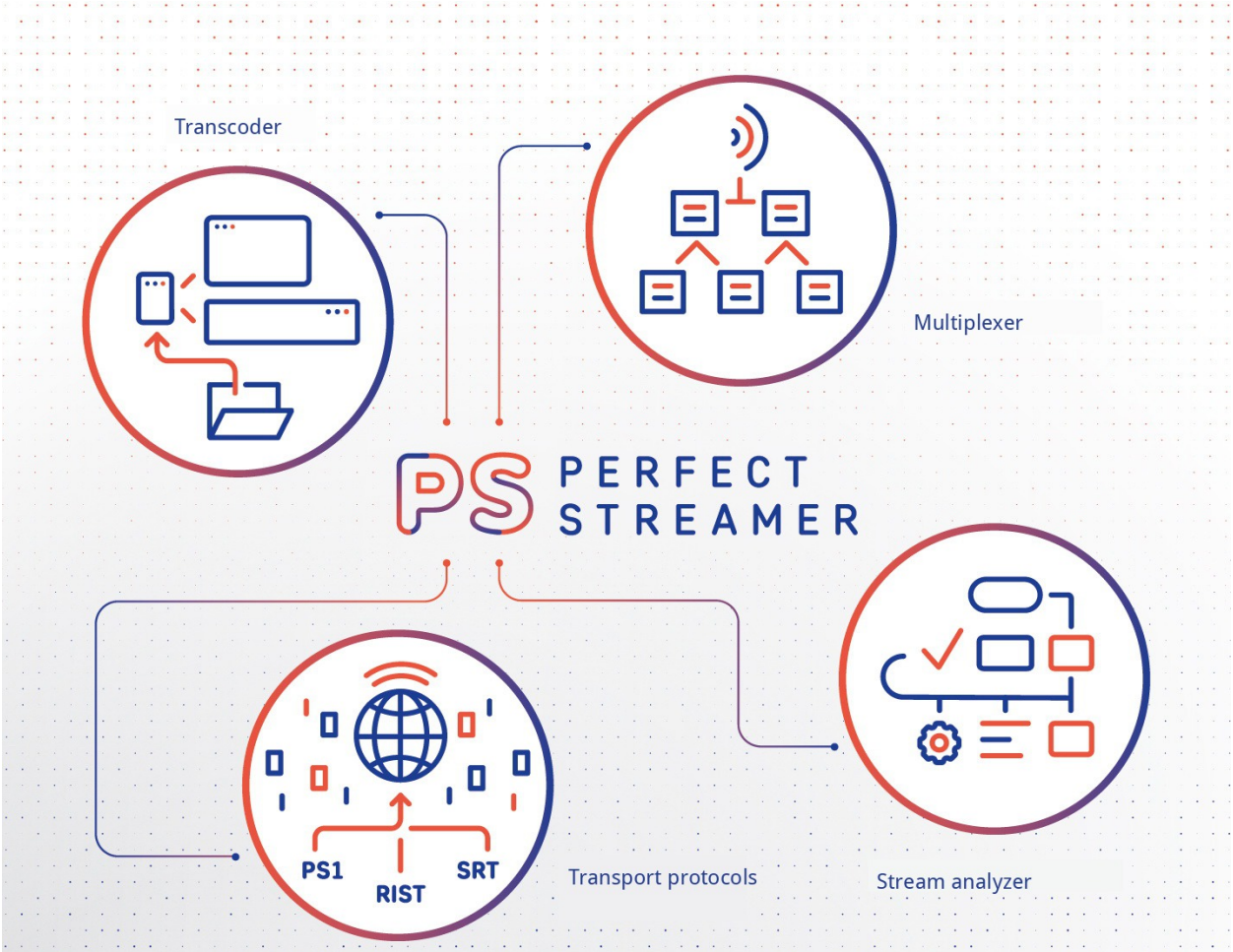
29.07.2026

Inhaltsverzeichnis

KAPITEL 1

Perfect Streamer — Benutzerhandbuch

1.1 Zweck



Perfect Streamer ist eine Server-Software für die zuverlässige Zustellung von MPEG-TS-Strömen über das öffentliche Internet mit Paketverlusten und Verzögerungen sowie für den Empfang, die Verarbeitung, die Multiplexierung, die Transkodierung und die OTT-Verteilung von Fernsehkanälen. Das Produkt richtet sich an Rundfunkveranstalter und Telekommunikationsbetreiber: Backbone-Übertragung von Kanälen zwischen Betreibern, DVB-to-IP-Kopfstationen, IPTV/OTT-Plattformen und Verteilnetze.

Perfect Streamer wird als einzige Anwendung für Linux x86-64 mit integrierter Weboberfläche und HTTP-API ausgeliefert. Eine einzige Installation führt alle Stufen der Verarbeitungskette aus: Empfang, Verarbeitung und Remultiplexierung, Transkodierung, Übertragung zwischen Knoten, OTT-Verteilung und Archivierung (DVR), Überwachung und vieles mehr.

1.1.1 Übertragung von Strömen zwischen Knoten

Für jeden Strom werden ein Sender-Server (**Sender**) und ein oder mehrere Empfänger (**Receiver**) konfiguriert. Für die Übertragung von Strömen zwischen Sender und Empfänger sowie für die Kommunikation mit Fremdgeräten stehen vier **Peer**-Protokolle zur Verfügung:

- **PS1** (Perfect Stream) — ein selbst entwickeltes, UDP-basiertes Protokoll, das nach dem Prinzip Automatic Repeat reQuest (ARQ) mit selektiver Neuübertragung arbeitet. Geringer Ressourcenbedarf; Übertragung von Strömen mit hoher Bitrate, einschließlich eines vollwertigen MPTS. Besonders effizient in Punkt-zu-Mehrpunkt-Konfigurationen, ein Sender und viele Empfänger.
- **SRT** — ein offenes, weit verbreitetes Protokoll mit guten Eigenschaften zur Kompensation von Paketverlusten; Listener/Caller-Modi; kompatibel mit Fremdgeräten und Cloud-Diensten.
- **RIST** — ein offenes, auf RTP/RTCP basierendes Protokoll. Es arbeitet nach dem ARQ-Prinzip ohne ACK, nur mit NACK, was eine hohe Effizienz gewährleistet; Profile Simple und Main.
- **Pro-MPEG / RTP+FEC** (Pro-MPEG COP3, auch bekannt als SMPTE 2022-1/2) — RTP mit Vorwärtsfehlerkorrektur (FEC) ohne Rückkanal. Die Vorteile sind die geringe Latenz und die Kompatibilität mit professioneller Rundfunktechnik; die Nachteile sind der ständige zusätzliche Datenverkehr und die schwache Wiederherstellung bei großen Paketverlusten.

Alle Peer-Protokolle unterstützen AES-Verschlüsselung. Eine ausführliche Beschreibung der Protokolle und die Planung der Übertragungskanäle finden Sie im Abschnitt *Planung und Datenübertragungsprotokolle*.

Neben den Peer-Protokollen werden Standard-Ein- und -Ausgänge unterstützt: UDP (Unicast/Multicast), RTP, HLS/HTTP, RTMP/RTMPS, Datei/Gerät, benannte Pipe (pipe) und externe Anwendung (std); zusätzlich nur als Eingänge — TCP und RTSP.

1.1.2 Hauptfunktionen

- *Verarbeitung ohne Transkodierung* — Listen erlaubter und gesperrter PIDs, Neuzuordnung von PIDs und automatische PID-Anordnung des Programms, Bearbeiten der SDT und der Audiosprachen, Ändern der PNR, Entfernen nicht benötigter Tabellen (SDT/CAT/NIT/TDT, Teletext, Untertitel), Angleichung der PAT/PMT-Intervalle an TR 101 290; Bitratenmodi, einschließlich CBR mit Stuffing bis zu einer DVB-kompatiblen konstanten Bitrate.
- *MPTS-Multiplexer* — Zusammenstellung eines Multiplex aus einzelnen Programmen mit PSI/SI-Generierung und einem TR 101 290-konformen Ausgang.
- *Demultiplexer* — Extraktion einzelner Programme aus einem MPTS in eigenständige SPTS.
- *Transkoder* — Hardware-Transkodierung auf NVIDIA-GPU und Intel VPL, Software-Transkodierung auf der CPU; mehrere Encoder aus einem einzigen Decoder (zum Beispiel für adaptive Bitrate).

- *Analysator* — TR 101 290-Konformitätsprüfung (DVB) für SPTS und MPTS, PCR-Drift-Detektor, T-STD-Puffermodell, tiefgehende Stromanalyse, SCTE-35-Analysator, CBR/VBR-Detektor.
- *EPG und EIT* — Erfassung von EIT aus dem Rundfunk in die EPG-Datenbank, XMLTV-Import, Generierung eines korrekten EIT (present/following und schedule) in den ausgehenden Strom, erzwungene Generierung der SDT.
- *EPG-Server* — Bereitstellung des vollständigen XMLTV unter `/xmltv` mit Einschränkung des Kanalsatzes nach Login, für OTT-middleware.
- *Testströme* — integrierter Testsignalgenerator.
- *Meshwork* — Zusammenschluss von Servern zu Domänen auf Basis des Gossip-Protokolls dezentraler Netze: Live-Status der Knoten, ein gemeinsames Verzeichnis der verwendeten Adressen und Ports (Ressourcenbibliothek), Replikation von Alerts über die Domäne.
- Sitzungsabrechnung und Zugriffskontrolle — Verbindungslimits pro Benutzer, ACL pro Kanal, Prefix/Suffix-Login zur Integration mit middleware. Autorisierung von Client-Sitzungen in einer externen Abrechnung — Prüfung jeder Sitzung beim Verbindungsaufbau und periodische Reautorisierung mit Entzug des Zugriffs mitten in der Sitzung (Lebenszyklus im RADIUS-Stil: Start / Interim / Stop); ein universeller HTTP connector zur Anbindung der Abrechnung des Betreibers. Die Abrechnung erfolgt getrennt für jedes Protokoll — HLS/DASH für OTT und PS1/SRT für das Peering von Strömen.
- *Weboberfläche* — Verwaltung aller Funktionen, Administratorrollen, adaptive Profile für Arbeitsplatzrechner, Tablet und Telefon.
- Automatische TLS-Zertifikate (ACME) — automatische Ausstellung und Erneuerung von Zertifikaten, Hot-Reload von TLS; HTTPS und HTTP/3 „out of the box“.
- *Analysator* und Überwachung — TR 101 290-Kontrolle (DVB) am Eingang, einheitliches Alerting mit Zustellung per Email, Telegram oder an ein externes Skript, Kanalmosaik.

Außerdem verfügbar: Quellenredundanz mit automatischer Umschaltung; Integration mit Überwachungssystemen (Zabbix, Grafana, Prometheus, InfluxDB, Nagios); Integration mit externer Abrechnung zur Autorisierung von Client-Sitzungen; Sicherung sowie Export/Import von Einstellungen.

1.1.3 Für DVB-Rundfunkveranstalter

Für Kopfstationen des Satelliten-, Kabel- und terrestrischen Rundfunks (DVB-to-IP, Remultiplexierung, Contribution zwischen Standorten) stehen zur Verfügung:

- *Empfang von DVB-Karten* — DVB-S, DVB-S2, DVB-T, DVB-T2 und DVB-C mit vollständigem Satz an Abstimmparametern; LNB- und DiSEqC-1.0-Steuerung, gemeinsame Nutzung eines LNB durch mehrere Tuner.
- Transponder-Scanner und Blindscan — Durchlauf der Reference-Listen (satellites.xml, cables.xml, terrestrial.xml) mit Erfassung von PSI/SI, Aufbau der Struktur Multiplex → Programm und Messung von SNR, Pegel und BER; Suche nach nicht standardmäßigen Transpondern ohne Reference-Datei.
- Entwürfelung — hardwarebasiert per CI/CAM (EN 50221) mit einem in den DVB-Empfänger integrierten CAM sowie softwarebasiert per BISS-1 / BISS-E (Schlüssel nach PNR oder PLP, Aktualisierung im laufenden Betrieb). Bereinigung des bedingten Zugriffs am Ausgang — Entfernen von ECM/EMM, CAT und CA-Deskriptoren aus der PMT, um einen sauberen FTA-Strom zu erhalten.
- T2-MI-Dekapselung (ETSI TS 102 773) — gleichzeitige Ausgabe des äußeren DVB-S/S2-Multiplex und aller eingebetteten T2-MI-Träger: Empfang von regionalem DVB-T2 über Satellit ohne separates Gateway.

- Signalmonitor (femon) — kontinuierliche Messung von Lock, Pegel, SNR, BER und UCB mit Verlaufsdiagrammen über die Zeit.
- *Demultiplexer* — Extraktion eines einzelnen Programms (nach PNR) aus einem empfangenen Transponder in ein eigenständiges SPTS.

1.1.4 Für OTT-Rundfunkveranstalter

Für IPTV/OTT-Betreiber und Verteilplattformen stehen zur Verfügung:

- *OTT-Auslieferung* — HLS, Low-Latency HLS und MPEG-DASH über CMAF; Auslieferung über HTTPS und HTTP/3 (QUIC); adaptive Multi-Bitrate (ABR) in einer einzigen master playlist; GOP/IDR-ausgerichtete Segmentierung; roher MPEG-TS over HTTP für TS-Clients.
- WebVTT-Untertitel — Dekodierung von DVB-Teletext und DVB-Untertiteln in WebVTT für OTT-Player, auch im Archiv.
- *Netzwerk-DVR* — Aufzeichnung jedes OTT-Kanals in ein Festplattenarchiv parallel zur Auslieferung; Wiedergabe des Archivs über dieselben HLS/DASH-URLs (Timeshift und VOD), nahtloser Übergang Archiv → Live, EPG-basiertes Catch-up, adaptives VOD und CMAF-Archiv, Multi-Festplatten-Speicher mit Retention und automatischer Bereinigung.
- *Transkoder und ABR-Leiter* — ein Decoder speist mehrere Encoder mit eigener Auflösung, Bitrate und eigenem Codec (1toN); hardwarebasiert auf der GPU (NVIDIA, Intel VPL) oder softwarebasiert auf der CPU.
- CDN-Bereitschaft — inhaltsadressierbare Segmente mit immutable-Headern, normalisierter cache-key und CORS für die Skalierung hinter einem Reverse-Proxy oder CDN.

1.2 Installation

Perfect Streamer wird aus Paketen installiert; für die Familien RHEL und Debian stehen Repositorys zur Verfügung. Nach der Installation *aktivieren Sie den Dienst und nehmen Sie die Erstkonfiguration vor*.

1.2.1 Systemanforderungen

- **Perfect Streamer** läuft unter dem Betriebssystem Linux. Die Hauptanforderung ist die GLIBC-Version >= 2.17.
- Netzwerkschnittstellen, die der Streamer-Dienst verwendet, müssen über eine statische Konfiguration verfügen.
- Die Installer-Skripte verwenden das Dienstprogramm **sudo**, d. h. es muss im System installiert sein.

Für die Red-Hat- und Debian-Familien stehen Installationspakete und Repositories zur Verfügung. Unterstützt werden RHEL-Version 7 und höher (CentOS usw.) sowie RPM-kompatible Distributionen — zum Beispiel openSUSE Leap (siehe die Anmerkung am Ende des Abschnitts *Installation auf Systemen der RHEL-Familie*). Debian-basierte Systeme (Ubuntu usw.) müssen über den Dienst systemd verfügen.

Ungefähre Hardwareanforderungen: 1 Kern mit 2.4 GHz und 1 GB RAM je 200 Mbit/s Datenverkehr. Diese Schätzung ist ungefähr und hängt von den verwendeten Protokollen und den Diensteinstellungen ab.

1.2.2 Installation auf Systemen der RHEL-Familie

Repository für RHEL 7 einrichten:

```
$ sudo yum install yum-utils
$ sudo yum-config-manager --add-repo=http://repo.pstreamer.tv/pub/pstreamer/pstreamer.
↪repo
```

Oder für RHEL 8+:

```
$ sudo yum config-manager --add-repo=http://repo.pstreamer.tv/pub/pstreamer/pstreamer.
↪repo
```

Paket installieren:

```
$ sudo yum -y install pstreamer
```

Paket aktualisieren:

```
$ sudo yum -y update pstreamer
```

Alle Pakete entfernen:

```
$ sudo yum -y remove pstreamer aksusbd
```

Bemerkung: **openSUSE** (Leap, aktuelle stabile Version) ist ein RPM-basiertes System. Es verwendet dasselbe Repository wie RHEL, die Operationen werden jedoch mit dem Paketmanager **zypper** ausgeführt:

```
$ sudo zypper addrepo http://repo.pstreamer.tv/pub/pstreamer/pstreamer.repo
$ sudo zypper --gpg-auto-import-keys refresh
$ sudo zypper install pstreamer
```

Aktualisierung — `sudo zypper update pstreamer`, Entfernung — `sudo zypper remove pstreamer aksusbd`.

1.2.3 Installation auf Systemen der Debian-Familie

Repository installieren:

```
$ sudo wget http://repo.pstreamer.tv/pub/deb/dists/pstreamer/pstreamer.list -O /etc/
↪apt/sources.list.d/pstreamer.list
$ sudo apt-get update
```

Paket installieren:

```
$ sudo apt-get install pstreamer
```

Paket aktualisieren:

```
$ sudo apt install pstreamer
```

Alle Pakete entfernen:

```
$ sudo apt-get remove pstreamer aksusbd
```

1.2.4 Dateien und Dienste

/usr/local/bin/pss
: Ausführbare Datei.

/opt/pss/config/pss.properties
: Globale Einstellungen, Protokolle, Ordnerpfade usw. Nach Änderungen den Dienst neu starten.

/opt/pss/config/pss.json
: Haupt-Einstellungsdatei. Wird automatisch erstellt und aktualisiert. Beim Start versucht der Dienst, genau diese Datei zu laden.

/opt/pss/config/pss_back.json
: Sicherungskopie der vorherigen funktionierenden Konfiguration. Wird durch die Aktion **Einstellungen wiederherstellen** in der Weboberfläche automatisch gespeichert und als erste Ausweichoption verwendet, wenn die Haupt-*pss.json* nicht ausgewertet werden kann.

/opt/pss/config/pss_default.json
: Standard-Einstellungsdatei. Wird mit dem Paket ausgeliefert und als letzte Ausweichoption verwendet, wenn weder die Haupt-*pss.json* noch *pss_back.json* geladen werden können. Aus ihr wird auch beim allerersten Start die betriebsbereite *pss.json* erzeugt: Die Datei legt den Port der Weboberfläche 8808 sowie das Konto *admin* / *admin* fest.

/opt/pss/config/bad/
: Archiv beschädigter *pss.json*-Dateien. Wenn die Haupt-Einstellungsdatei beim Start nicht ausgewertet werden kann, wird sie mit einem Namen der Form *pss_YYYYMMDD_HHMMSS.json* hierher verschoben. Das Verzeichnis wird automatisch erstellt. Näheres siehe Abschnitt *Verhalten beim Start und bei Konfigurationsfehlern*.

/opt/pss/data
: Ordner zur Datenablage. Wird automatisch erstellt und aktualisiert. Kann in der Datei mit den globalen Einstellungen geändert werden.

/usr/lib/systemd/system/pss.service
: systemd-Unit-Datei des Dienstes.

/var/log/pss
: Ordner für die Protokollausgabe. Kann in der Datei mit den globalen Einstellungen geändert werden.

Der Dienstname lautet **pss**. Wird unter dem Benutzer **pss** ausgeführt.

Während der Installation wird das begleitende Paket **aksusbd** des Schutzsystems installiert; es enthält die Dienste **hasplmd** und **aksusbd**.

Verhalten beim Start und bei Konfigurationsfehlern

Beim Start versucht der Dienst nacheinander, die Einstellungsdateien aus dem Ordner */opt/pss/config* zu laden:

1. *pss.json* — Haupt-Einstellungsdatei.
2. *pss_back.json* — Sicherungskopie der vorherigen funktionierenden Konfiguration.
3. *pss_default.json* — mit dem Paket ausgelieferte Standardeinstellungen.

Es wird die erste erfolgreich geladene Datei verwendet. Wenn alle drei Dateien fehlen oder beschädigt sind, startet der Dienst mit leeren Einstellungen; in diesem Fall sind eine manuelle Bearbeitung des Administratorpassworts und ein Neustart erforderlich.

Strukturell beschädigte Einstellungsdatei. Wenn *pss.json* einen JSON-Syntaxfehler, unbekannte Schlüssel, einen falschen Werttyp oder eine Verletzung der Eindeutigkeit enthält (zum Beispiel zwei Streams mit derselben *id*), verschiebt der Dienst die beschädigte Datei in das Archiv */opt/pss/config/bad/* unter dem Namen *pss_YYYYMMDD_HHMMSS.json* (Datum und Uhrzeit des Starts). Danach setzt der Dienst seine Ladeversuche in der üblichen Reihenfolge fort und speichert die funktionierende Konfiguration aus *pss_back.json* oder *pss_default.json* erneut in *pss.json*. Die Einzelheiten (Schlüsselname, Fehlerbeschreibung, Dateiname im Archiv) werden in das Betriebsprotokoll geschrieben.

In das Archiv gelangt nur die Haupt-*pss.json*. Die Dateien *pss_back.json* und *pss_default.json* werden bei Beschädigung nicht archiviert — die Protokolleinträge reichen für die Diagnose aus, und die Dateien selbst bleiben an ihrem Platz und können manuell korrigiert werden.

Wenn zum Zeitpunkt des nächsten Ladens in */opt/pss/config/bad/* bereits eine Datei mit demselben Zeitstempel vorhanden ist (zum Beispiel bei schnellen Neustarts innerhalb derselben Sekunde), wird sie überschrieben.

Numerische Werte außerhalb des zulässigen Bereichs. Wenn in der Einstellungsdatei ein numerischer Wert vorkommt, der kleiner als der zulässige Mindestwert oder größer als der zulässige Höchstwert für diesen Parameter ist, verwirft der Dienst die Datei nicht vollständig. Stattdessen wird eine Warnung in das Protokoll geschrieben, die den Parameternamen, den gelesenen Wert und die angewendete Grenze angibt, und der Wert selbst wird auf die nächstgelegene zulässige Bereichsgrenze (den Mindest- oder Höchstwert) gesetzt. Nach Abschluss des Ladevorgangs speichert der Dienst *pss.json* automatisch mit den bereits korrigierten Werten erneut, sodass diese Warnungen beim erneuten Start nicht mehr erscheinen.

Dieses Verhalten gilt nur beim erstmaligen Laden der Einstellungsdatei. Beim Ändern der Einstellungen über die Weboberfläche oder die externe API werden Werte außerhalb des zulässigen Bereichs weiterhin mit einem Fehler abgelehnt — ohne automatische Korrektur.

1.2.5 Transcoder

Installation der Transcoder für Perfect Streamer.

Verfügbare Pakete:

- **pstreamer-tcsw** — Transcodierung auf der CPU (Software).
- **pstreamer-tcnv** — Transcodierung auf einer NVIDIA-GPU.
- **pstreamer-tcivpl** — Transcodierung auf einer Intel-GPU (Intel VPL).

Die Hauptvoraussetzung ist GLIBC-Version >= 2.28.

Der Transcoder läuft unter AlmaLinux 8.9. Der Intel-VPL-Transcoder läuft auf Systemen mit AlmaLinux 10 (RHEL 10) und Ubuntu 24.04.

Welche Intel-Hardware für die Transkodierung geeignet ist, welche Laufzeitumgebungen und Codecs auf den verschiedenen Grafikgenerationen verfügbar sind und wie sich prüfen lässt, dass die Hardware erkannt wurde, beschreibt der Abschnitt *Intel-VPL-Transcoder: unterstützte Hardware*.

RHEL 8+

1. pstreamer installieren.
2. Für NVIDIA die Repositorys installieren und das System aktualisieren (falls sie noch nicht installiert wurden):

```
sudo dnf config-manager --add-repo https://developer.download.nvidia.com/compute/cuda/
↪repos/rhel9/x86_64/cuda-rhel9.repo
sudo dnf clean all
sudo dnf update -y
reboot
```

3. NVIDIA Encoder.

- CUDA installieren:

```
sudo dnf -y install cuda-toolkit-12-5
```

- Den Treiber installieren (Variante auswählen):

Legacy

```
sudo dnf -y module install nvidia-driver:latest-dkms
```

New

```
sudo dnf -y module install nvidia-driver:open-dkms
```

Nach der Installation muss die Maschine unbedingt neu gestartet werden:

```
reboot
```

Nach dem Neustart die Funktion des Treibers prüfen:

```
nvidia-smi
modprobe nvidia
sudo lsmod | grep nvidia
```

oder

```
modprobe nouveau
sudo lsmod | grep nouveau
```

4. Intel VPL Encoder.

- Installation des Intel-Treibers und von Intel VPL (RHEL 10):

```
dnf install -y intel-gpu-firmware
dnf install -y https://mirrors.rpmfusion.org/free/el/rpmfusion-free-release-$(rpm -E
↪%rhel).noarch.rpm https://mirrors.rpmfusion.org/nonfree/el/rpmfusion-nonfree-
↪release-$(rpm -E %rhel).noarch.rpm
dnf install -y intel-media-driver
dnf install -y intel-vpl-gpu-rt
```

5. Die Transcoder-Pakete installieren.

- CPU-Software-Methode:

```
sudo dnf install -y pstreamer-tcsw
```

- NVIDIA GPU:

```
sudo dnf install -y pstreamer-tcnv
```

- Intel VPL GPU:

```
sudo dnf install -y pstreamer-tcivpl
```

Ubuntu 22/24

1. pstreamer installieren.
2. NVIDIA Encoder.

- CUDA-Toolkit Version 12.5 installieren:

```
wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64/cuda-
↪keyring_1.1-1_all.deb
sudo dpkg -i cuda-keyring_1.1-1_all.deb
sudo apt-get update
sudo apt-get -y install cuda-toolkit-12-5
```

- Den Treiber installieren (Variante auswählen):

legacy kernel module flavor:

```
sudo apt-get install -y cuda-drivers
```

oder

open kernel module flavor:

```
sudo apt-get install -y nvidia-driver-555-open
sudo apt-get install -y cuda-drivers-555
```

Nach der Installation muss die Maschine unbedingt neu gestartet werden:

```
reboot
```

Nach dem Neustart die Funktion des Treibers prüfen:

```
nvidia-smi
```

Für den Betrieb des Transcoders ist CUDA-Version 12.5 erforderlich. Auf dem System kann bereits eine andere CUDA-Version installiert sein. Außerdem kann bei einer Systemaktualisierung CUDA auf eine neuere Version aktualisiert werden. Dies beeinträchtigt den Betrieb nicht; sie müssen nicht entfernt werden, da verschiedene CUDA-Versionen in separaten Verzeichnissen installiert werden.

3. Intel VPL Encoder (Ubuntu 24.04).

Die zentralen Intel-VPL-Komponenten stammen aus dem offiziellen Intel-Graphics-Repository — in den Standard-Repositorys von Ubuntu sind sie älter und unterstützen keine modernen GPUs.

- Das Intel-Graphics-Repository hinzufügen:

```
sudo apt-get update
sudo apt-get install -y ca-certificates gnupg wget
wget -qO - https://repositories.intel.com/gpu/intel-graphics.key | sudo gpg --yes --
↪dearmor --output /usr/share/keyrings/intel-graphics.gpg
echo "deb [arch=amd64 signed-by=/usr/share/keyrings/intel-graphics.gpg] https://
↪repositories.intel.com/gpu/ubuntu noble unified" | sudo tee /etc/apt/sources.list.d/
↪intel-gpu-noble.list
sudo apt-get update
```

- Die Intel-VPL-Laufzeitumgebung und den Treiber installieren:

```
sudo apt-get install -y libvpl2 libmfxgen1 intel-media-va-driver-non-free
```

- Optional können die Diagnosewerkzeuge installiert werden:

```
sudo apt-get install -y libvpl-tools
```

- Der Benutzer **pss**, unter dem der Dienst läuft, muss den Gruppen **render** und **video** angehören (Zugriff auf die Geräte */dev/dri/*):

```
sudo usermod -aG render,video pss
sudo systemctl restart pss
```

- Prüfen, dass die GPU sichtbar ist (bei installierten libvpl-tools):

```
vainfo --display drm --device /dev/dri/renderD128
vpl-inspect
```

In der Ausgabe von *vainfo* muss die Zeile „Driver version: Intel iHD …“ vorhanden sein, und *vpl-inspect* muss die Hardware-Implementierung auflisten (AccelerationMode VAAPI).

4. Die Transcoder-Pakete installieren.

- CPU-Software-Methode:

```
sudo apt-get install -y pstreamer-tcsw
```

- NVIDIA GPU:

```
sudo apt-get install -y pstreamer-tcnv
```

- Intel VPL GPU (Ubuntu 24.04):

```
sudo apt-get install -y pstreamer-tcivpl
```

Dateien und Installationsprüfung

Die Transcoder-Pakete installieren folgende Dateien:

- */usr/local/bin/tcsw* — ausführbare Datei des SW-(CPU-)Transcoders.
- */usr/local/bin/tcnv* — ausführbare Datei des NVIDIA-(GPU-)Transcoders.
- */usr/local/bin/tcivpl* — ausführbare Datei des Intel-VPL-(GPU-)Transcoders.
- */opt/pss/config/pss_tc_sw.properties* — Startkonfigurationsdatei für den SW-(CPU-)Transcoder.

- */opt/pss/config/pss_tc_nv.properties* — Startkonfigurationsdatei für den NVIDIA-(GPU-)Transcoder.
- */opt/pss/config/pss_tc_ivpl.properties* — Startkonfigurationsdatei für den Intel-VPL-(GPU-)Transcoder.

Die Installation der Transcoder-Pakete startet den Dienst **pss** neu. Die Installation der Transcoder kann im Abschnitt *About* der Weboberfläche geprüft werden: Es wird die Version der Transcoder oder ein Fehler angezeigt.

Andere Betriebssysteme auf Basis von Debian und RHEL

Um den Transcoder pstreamer-tcnv auf anderen Betriebssystemen als Ubuntu 22/24 und RHEL 8+ zu installieren, verwenden Sie den Konfigurator zur Auswahl der erforderlichen CUDA- und Treiberversion auf der NVIDIA-Website:

https://developer.nvidia.com/cuda-toolkit-archive

Bei der Auswahl jeder CUDA-Version stehen Informationen darüber zur Verfügung, für welche Betriebssystemversion sie geeignet ist. Die unterstützte Architektur ist x86_64. Wenn Sie eine ältere Betriebssystemversion benötigen, prüfen Sie deren Unterstützung in älteren CUDA-Versionen. Die Hauptvoraussetzung für das Betriebssystem ist die Unterstützung von GLIBC-Version >= 2.28.

Der NVIDIA-Treiber muss aus dem Repository installiert werden, das für Ihre Betriebssystemversion auf der Seite der ausgewählten CUDA-Version angeboten wird.

Die Unterstützung von NVIDIA-Grafikkarten für die Funktionalität des Transcoders pstreamer-tcnv kann auf der NVIDIA-Website in der Video Encode and Decode Support Matrix geprüft werden. Auf dieser Seite steht eine Matrix zur Unterstützung von Videoformaten für Decoder und Encoder sowie weitere Merkmale zur Verfügung.

Entfernen der alten CUDA-Version und des Treibers

Falls eine falsche CUDA- und Treiberversion installiert wurde, kann es erforderlich sein, eine andere Version zu installieren, wobei die installierte Version zuvor entfernt werden muss.

https://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html?highlight=uninstall#removing-cuda-toolkit

Entfernen von CUDA

RHEL:

```
dnf remove "cuda*" "*cublas*" "*cufft*" "*cufile*" "*curand*" "*cusolver*" "*cusparse*
↪" "*gds-tools*" "*npp*" "*nvjpeg*" "nsight*" "*nvvm*" "*nvptx*"
```

Debian/Ubuntu:

```
apt remove --autoremove --purge "*cuda*" "*cublas*" "*cufft*" "*cufile*" "*curand*"
↪"*cusolver*" "*cusparse*" "*gds-tools*" "*npp*" "*nvjpeg*" "nsight*" "*nvvm*"
↪"*nvptx*"
```

Entfernen des Treibers

Ubuntu 22/24:

```
apt remove --autoremove --purge -V \
   cuda-compat\* \
   cuda-drivers\*  \
   libnvidia-cfg1\* \
   libnvidia-compute\* \
   libnvidia-decode\* \
   libnvidia-encode\* \
   libnvidia-extra\* \
   libnvidia-fbc1\* \
   libnvidia-gl\* \
   libnvidia-gpucomp\* \
   libnvidia-nscq\* \
   libnvsdm\* \
   libxnvctrl\* \
   nvidia-dkms\* \
   nvidia-driver\* \
   nvidia-fabricmanager\* \
   nvidia-firmware\* \
   nvidia-headless\* \
   nvidia-imex\* \
   nvidia-kernel\* \
   nvidia-modprobe\* \
   nvidia-open\* \
   nvidia-persistenced\* \
   nvidia-settings\* \
   nvidia-xconfig\* \
   xserver-xorg-video-nvidia\*
```

RHEL 9:

```
dnf module remove --all nvidia-driver
dnf module reset nvidia-driver
```

RHEL 10:

```
dnf remove nvidia-driver\*
```

1.3 Erster Start und Aktivierung

1.3.1 Temporäre Aktivierung und Start

Die temporäre Version unterliegt keiner Begrenzung der Anzahl der Streams. Nach der Installation ist der Dienst **pss** inaktiv, da für den Start eine Aktivierung erforderlich ist. Für die temporäre Aktivierung das Skript ausführen:

```
$ /opt/pss/tools/activate.sh
```

Der Dienst **pss** wird gestartet und für den Autostart konfiguriert. Der erfolgreiche Start kann überprüft werden:

```
$ systemctl status pss
```

Bei Problemen die Protokolle einsehen:

```
$ tail -n 20 /var/log/pss/main.log
$ journalctl -u pss -n 20
```

1.3.2 Anfangseinstellungen

Über einen Browser mit der Weboberfläche verbinden:

http://host:8808

Benutzername: *admin*

Passwort: *admin*

Es sind zwei Weboberflächen verfügbar: die neue (Haupt-)Oberfläche unter dem Pfad */admin* und die klassische unter dem Pfad */classic*.

Standard-Ports der Dienste:

Dienst	HTTP	HTTPS
Weboberfläche und HTTP API	8808	43981
OTT-Verteilung (HTTP-Server)	43972	43982
EPG-Server	43973	43983

Unmittelbar nach der Installation ist nur der HTTP-Port der Weboberfläche verfügbar — `8808`. Er ist in der mitgelieferten Datei *pss_default.json* festgelegt, aus der beim ersten Start die betriebsbereite *pss.json* erzeugt wird; weitere Dienste beschreibt diese Datei nicht, daher werden die übrigen Ports der Tabelle erst nach der Konfiguration des jeweiligen Dienstes geöffnet und HTTPS erst nach dem Aktivieren von TLS und der Installation eines Zertifikats. Fehlt der Schlüssel `web-server.bind-port` in den Einstellungen, verwendet der Dienst den eingebauten Ersatzwert `43971`.

Unbedingt Login und Passwort des Administrators ändern — in der Administratorliste der Weboberfläche. Den Konten werden Rollen zugewiesen: **Admin** (Vollzugriff), **Restricted admin** (Anzeigen und Pausieren von Streams), **Viewer** (nur Anzeigen).

Bei Bedarf den Port der Weboberfläche ändern — in den Einstellungen des Webservers. Die Änderung wird nach einem Neustart des Dienstes wirksam; der Neustart kann über die Weboberfläche durchgeführt werden.

Die Aktivierungsdaten können im Abschnitt *About* der Weboberfläche überprüft werden.

1.3.3 Dauerhafte Aktivierung

Das Schutzsystem unterstützt Software- (SL) und USB- (HL) Schlüssel. Für die dauerhafte Aktivierung:

1. Im Abschnitt *About* der Weboberfläche die C2V-Daten für die Aktivierungsanfrage abrufen und an den Anbieter senden.
2. Die vom Anbieter erhaltenen V2C-Daten aus einer Datei oder aus der Zwischenablage im selben Abschnitt der Weboberfläche eingeben.
3. Im selben Abschnitt die Aktivierungsdaten überprüfen.

Damit Perfect Streamer den USB-Schlüssel bei aktiver Testlizenz erkennt, muss der Dienst **pss** neu gestartet werden — über die Neustartfunktion in der Weboberfläche oder mit dem Befehl `sudo systemctl restart pss`. Läuft die Frist der Testlizenz ab, wechselt der Dienst selbst auf den dauerhaften USB-Schlüssel.

1.3.4 Beim Ablauf der Frist der Jahres- oder Testlizenz

Wenn der Dienst **pss** nicht gestartet werden kann, den Befehl eingeben:

```
$ journalctl -u pss -n 20
```

Der folgende Fehler bedeutet, dass die Lizenz von Perfect Streamer abgelaufen ist:

```
LDK Protection System: Feature has expired (H0041)
```

1.4 Planung und Datenübertragungsprotokolle

Der Kern von **Perfect Streamer** ist die zuverlässige Zustellung von MPEG-TS-Streams zwischen Knoten über ein öffentliches Netz mit Paketverlust und Verzögerungen. Dieser Abschnitt beschreibt das Übertragungsmodell, die Transportprotokolle und wie ein Kanal geplant wird: Auswahl eines Protokolls sowie Berechnung von Bandbreite, Latenz und Ports. Detaillierte Einstellungen einzelner Felder finden Sie in der kontextbezogenen Beschreibung der Weboberfläche (*Weboberfläche*), und die Streamverarbeitung auf einem Knoten im Abschnitt *Streamer: Implementierungsdetails*.

1.4.1 Übertragungsmodell: Stream, Sender, Receiver, Peer

Die Arbeitseinheit ist ein Stream (**Stream**), ein einzelner MPEG-TS-Kanal. Für jeden Stream werden ein Sender-Server (**Sender**) und ein oder mehrere Empfänger (**Receiver**) konfiguriert; eine Sender-Empfänger-Kopplung wird als **Peer** bezeichnet.

Die Konfiguration läuft auf Listen von Eingängen (**input**) und Ausgängen (**output**) für jeden Stream hinaus:

- auf dem Sender ist **input** die MPEG-TS-Quellen, **output** die Übertragung an die Empfänger;
- auf dem Empfänger ist **input** der Empfang des Streams vom Sender.

Mehrere Eingänge in der Liste bieten Quellenredundanz, mehrere Ausgänge die gleichzeitige Verteilung eines einzelnen Kanals an viele Empfänger und über verschiedene Protokolle.

Bei Protokollen, bei denen der Empfänger die Verbindung initiiert, verbindet sich ein Empfänger hinter NAT selbst mit dem Sender — eine Weiterleitung eingehender Ports auf seiner Seite ist nicht erforderlich.

1.4.2 Peer-Protokolle für zuverlässige Übertragung

Für die Übertragung zwischen Sender und Empfänger stehen vier Peer-Protokolle zur Verfügung. Sie gliedern sich nach dem Prinzip der Verlustkompensation in zwei Klassen:

- **ARQ** (Automatic Repeat reQuest) — verlorene Pakete werden auf Anforderung des Empfängers erneut übertragen. Erfordert einen Rückkanal und einen Puffer auf dem Empfänger; die Wiederherstellungstiefe ist konfigurierbar. Zu dieser Klasse gehören PS1, SRT und RIST.

- **FEC** (Forward Error Correction) — dem Stream werden fortlaufend redundante Daten hinzugefügt, die eine Wiederherstellung von Verlusten ohne Rückkanal ermöglichen. Zu dieser Klasse gehört Pro-MPEG / RTP+FEC.

PS1 (Perfect Stream)

Ein Eigenentwicklungsprotokoll auf UDP-Basis. Es arbeitet nach dem ARQ-Prinzip mit selektiver Neuübertragung. Es zeichnet sich durch geringen Ressourcenverbrauch aus und überträgt Streams mit hoher Bitrate, einschließlich vollständigem MPTS.

Der Empfänger initiiert die Verbindung, daher erfordert der Sender eine Authentifizierung: Empfänger werden als Peer mit Login und Passwort eingetragen oder per IP autorisiert. Die Latenz wird durch den Puffer des Empfängers bestimmt (Fenster für Jitter-Beseitigung und Neuübertragung); der Puffer des Senders muss größer sein als die Puffer der Empfänger. Beim Wechsel der aktiven Quelle auf dem Sender verbinden sich die Empfänger nicht neu — die Unterbrechung wird durch reguläre Neuübertragung ohne sichtbaren Neuaufbau der Verbindung behoben.

PS1 ist ein proprietäres Protokoll und funktioniert nur zwischen Perfect-Streamer-Knoten. Es ist besonders effizient in einer Punkt-zu-Mehrpunkt-Konfiguration: ein Sender und viele Empfänger.

SRT

Ein offenes Protokoll auf UDP-Basis (UDT). Es ist weit verbreitet und kompensiert Paketverluste gut. Es unterstützt die Modi listener und caller, was Kompatibilität mit SRT-Geräten von Drittanbietern und Cloud-Diensten bietet.

Im listener-Modus auf dem Sender verbinden sich mehrere Empfänger mit einem einzelnen Stream; zur Autorisierung werden Empfänger als Peer eingetragen, eine Autorisierung per IP ist verfügbar. Die initiierende Seite (caller) kann sich sowohl auf dem Empfänger als auch auf dem Sender befinden. Die Bandbreitenreserve für die Neuübertragung wird in Prozent über der Stream-Bitrate festgelegt.

Auf dicht belegten Empfangsknoten mit einer großen Anzahl von SRT-Eingängen im caller-Modus kann die integrierte Pufferung des SRT-Empfängers (TSBPD) deaktiviert werden: Die Synchronisation übernimmt dann der Jitter-Puffer des Knotens (*Synchronisation*), was die CPU-Last spürbar senkt.

RIST

Ein offenes Protokoll auf RTP/RTCP-Basis. Es arbeitet nach dem ARQ-Prinzip ohne ACK, nur mit NACK, was hohe Effizienz gewährleistet. Es verwendet unicast und multicast.

Die Profile Simple und Main sind implementiert: Simple belegt zwei aufeinanderfolgende UDP-Ports (der Basisport muss gerade sein), Main multiplext die Daten in einen einzelnen Port. Mehrere Pfade (Peers) mit gewichteter Lastverteilung werden unterstützt — für Redundanz und Aggregation. Bei multicast können es viele Empfänger sein, mit Autorisierung per IP.

Pro-MPEG / RTP+FEC (SMPTE 2022-1/2)

Zustellung von MPEG-TS über RTP mit Vorwärtsfehlerkorrektur (FEC), ohne Rückkanal. Das Protokoll ist auch als Pro-MPEG COP3 bekannt. In PSS ist eine zweidimensionale FEC-Matrix implementiert (Zeilen und Spalten, SMPTE 2022-2).

Der Strom von RTP-Paketen wird in eine Matrix vorgegebener Größe gruppiert, entlang deren Zeilen und Spalten FEC-Pakete gebildet werden. Je kleiner die Matrix, desto besser die Wiederherstellung, aber desto höher der konstante Overhead. FEC-Kanäle belegen zwei zusätzliche Ports (`port`+2 und `port`+4), was bei der Platzierung mehrerer Streams auf einem Host oder in einer multicast-Gruppe zu berücksichtigen ist.

Die Vorteile sind niedrige feste Latenz und Kompatibilität mit professioneller Sendetechnik. Die Nachteile sind konstanter zusätzlicher Verkehr und schwache Wiederherstellung bei hohen Verlusten (über ~0,2 %), daher ist das Protokoll für verwaltete Kanäle mit geringen Verlusten und nicht für das „schwere“ öffentliche Internet ausgelegt.

1.4.3 Auswahl des Übertragungsprotokolls

Protokoll	Prinzip	Rückkanal	Latenz	Kompatibilität
PS1	ARQ über UDP	erforderlich	konfigurierbar	nur zwischen Perfect-Streamer-Knoten
SRT	ARQ über UDP (UDT)	erforderlich	konfigurierbar	offen; SRT-Geräte von Drittanbietern und Cloud
RIST	ARQ (NACK) über RTP/RTCP	erforderlich	konfigurierbar	offen; RIST-Geräte von Drittanbietern; unicast und multicast
Pro-MPEG / RTP+FEC	FEC über RTP	nicht erforderlich	niedrig, fest	offen (SMPTE 2022-1/2); professionelle Sendetechnik

Auswahlkriterien:

- ein Kanal zwischen Perfect-Streamer-Knoten, hohe Bitrate oder vollständiges MPTS, Punkt-zu-Mehrpunkt-Verteilung — **PS1**;
- Kompatibilität mit Geräten von Drittanbietern oder Cloud, flexible Einstellung der Neuübertragungsreserve — **SRT**;
- ein offener RTP-basierter Transport, multicast, Lastverteilung über mehrere Pfade — **RIST**;
- ein verwalteter Kanal mit geringen Verlusten, minimale Latenz ohne Rückkanal, professionelle Sendetechnik — **Pro-MPEG / RTP+FEC**.

1.4.4 Planung von Bandbreite, Latenz und Ports

ARQ-Protokolle (PS1, SRT, RIST). Verluste werden durch Neuübertragung kompensiert, daher ist Folgendes vorzusehen:

- einen Rückkanal vom Empfänger zum Sender;
- eine Bandbreitenreserve über der nominalen Bitrate — Verlustspitzen verursachen Spitzen im Neuübertragungsverkehr;
- einen Puffer auf dem Empfänger für die Kanallatenz. Je größer der Puffer, desto tiefer die Verlustwiederherstellung, aber desto höher die Ende-zu-Ende-Latenz; als Richtwert dienen mehrere

RTT-Werte. Streams mit niedriger Bitrate erfordern einen zeitlich größeren Puffer, damit genügend Pakete in das Wiederherstellungsfenster passen.

FEC-Protokoll (Pro-MPEG / RTP+FEC). Redundante Pakete werden fortlaufend übertragen, unabhängig von den tatsächlichen Verlusten, daher ist der Overhead fest und hängt von der Größe der FEC-Matrix ab. Ein Rückkanal ist nicht erforderlich, die Latenz ist niedrig und fest, aber die Wiederherstellung ist begrenzt — dieser Transport ist für Kanäle mit geringen Verlusten.

Ports. Jeder Lauschpunkt benötigt einen eindeutigen UDP-Port innerhalb des Hosts oder der Gruppe. Beachten Sie zusätzlich, dass das RIST-Simple-Profil zwei aufeinanderfolgende Ports belegt (den geraden Basisport und den nächsten) und Pro-MPEG den Basisport plus zwei FEC-Ports (`port`+2 und `port`+4). In einem Meshwork-Netz hilft ein gemeinsamer Katalog belegter Adressen und Ports, Kollisionen zu vermeiden (siehe *Meshwork*).

1.4.5 Stream-Verschlüsselung

Alle Peer-Protokolle unterstützen AES-Verschlüsselung mit einer gemeinsamen Passphrase, die auf beiden Seiten eingegeben wird. Für SRT wird zusätzlich die Schlüssellänge gewählt (128, 192 oder 256 Bit). Die Verschlüsselung ändert die Stream-Bitrate nicht.

Für Pro-MPEG ist die Verschlüsselung eine nicht standardmäßige Erweiterung, daher wird bei ihrer Aktivierung die Kompatibilität mit Software und Geräten von Drittanbietern nicht garantiert.

1.4.6 Weitere Eingänge und Ausgänge

Neben den Peer-Protokollen stehen Standardtransporte für Empfang und Übertragung zur Verfügung:

Protokoll	Eingang	Ausgang	Anmerkung
UDP	ja	ja	unicast/multicast, SSM, Schnittstellenauswahl; bis zu 7 TS-Pakete pro Datagramm
RTP	ja	ja	Wiederherstellung der Reihenfolge vertauschter Pakete
TCP	ja	—	Client-Modus; Empfang dort, wo UDP blockiert ist
HLS / HTTP	ja	ja	beim Empfang wird die Variante der adaptiven Wiedergabeliste anhand einer konfigurierbaren Nummer gewählt (standardmäßig die erste); Eingang nur für SPTS
RTSP	ja	—	IP-Kameras und RTSP-Server; Auto-Remux zu MPEG-TS; nur für SPTS
RTMP / RTMPS	ja	ja	Veröffentlichung an RTMP-Empfänger von Drittanbietern; H.264 und HEVC; nur für SPTS
file / device	ja	ja	Aufnahme in eine TS-Datei und Ausgabe an ein Gerät (einschließlich SDI); Schleifenwiedergabe aus einer Datei
pipe (FIFO)	ja	ja	Brücke zu einem externen Prozess über eine benannte Pipe
std (externe Anwendung)	ja	ja	Brücke zu nicht standardmäßigen Protokollen über eine Konsolenanwendung

Veröffentlichung und Empfang von RTMP sind in *RTMP-Veröffentlichung und -Empfang* beschrieben; RTSP, HLS/HTTP-Eingang, Dateien und Geräte, benannte Pipes und externe Anwendungen — in *Sonstige Ein- und Ausgänge*. Detaillierte Einstellungen jedes Transports finden sich im Abschnitt *Weboberfläche*. Die OTT-Verteilung (HLS, LL-HLS, DASH) ist gesondert in *OTT und DVR* beschrieben.

1.4.7 Quellenredundanz und Verteilung

Redundanz. Für einen Stream können mehrere Eingänge festgelegt werden; aktiv ist immer nur einer. Beim Ausfall des aktiven Eingangs schaltet der Stream automatisch auf den nächsten in der Liste um und kann bei Wiederherstellung zur priorisierten Quelle zurückkehren. Dadurch lassen sich die primäre und die Reservequellen eines Kanals im Voraus planen; detaillierte Einstellungen der Eingangsumschaltung sind in *SPTS-Streams* beschrieben.

Verteilung. Mehrere Ausgänge für einen einzelnen Stream ermöglichen es, einen einzelnen Kanal gleichzeitig an viele Empfänger und über verschiedene Protokolle zu verteilen.

1.4.8 Anforderungen an den Eingangsstream

- Konformität mit ISO/IEC 13818-1, Single Program (SPTS) oder Multi Program Transport Stream (MPTS).
- Die Zusammensetzung der Spuren richtet sich nach dem gewählten Inhaltstyp des SPTS-Streams, siehe *SPTS-Streams*.
- Verwürfelte (verschlüsselte) Streams werden unterstützt, siehe *SPTS-Streams*.
- Zur Synchronisation muss der Stream gültige PCR-Marken enthalten.

Filterung, Modifikation und Bitratenmodi des Eingangsstreams sind in *SPTS-Streams* beschrieben, die Besonderheiten des Multiplex in *MPTS-Streams*.

1.5 Schnellstart

Durchgängiges Szenario: vom installierten Knoten bis zum laufenden Stream mit Auslieferung. Es wird vorausgesetzt, dass Perfect Streamer bereits installiert ist (*Installation*), der Dienst gestartet und aktiviert wurde und Sie sich an der Weboberfläche angemeldet haben (*Erster Start und Aktivierung*, Zugriff und Rollen — *Zugriff und Rollen*).

1.5.1 Schritt 1. Stream anlegen

1. Öffnen Sie die Ansicht „Streams“ (*Streams*) und klicken Sie auf „+ Neuer Stream“.
2. Belassen Sie im Fenster „Neuer Stream“ (*Neuer Stream*) den Typ SPTS — ein einzelnes Programm — und legen Sie den „Stream-Name“ fest (2–32 Zeichen: lateinische Buchstaben, Ziffern, Bindestrich, Unterstrich). Der Typ kann nach dem Anlegen nicht mehr geändert werden.
3. Klicken Sie auf „Erstellen“ — der Stream-Editor wird geöffnet (*Stream konfigurieren*).

Ein neuer Stream wird angehalten angelegt — das ist normal: zuerst die Konfiguration, dann der Start.

1.5.2 Schritt 2. Eingang hinzufügen

Klicken Sie im Editor auf der Registerkarte „Eingänge“ auf „Eingang hinzufügen“. Für den typischen UDP/multicast-Empfang wählen Sie den Typ udp und geben die Gruppenadresse sowie den Port an — die Felder Address (multicast group) und Port; die Feldbezeichnungen entsprechen denen der Oberfläche. Die vollständige Liste der Empfangsprotokolle und deren Auswahl ist im Abschnitt *Planung und Datenübertragungsprotokolle* beschrieben.

Steht keine reale Quelle zur Verfügung, verwenden Sie den Testgenerator — einen Eingang des Typs test-stream (*Teststreams*; das Paket des Software-Transcoders muss installiert sein, *Transcoder*): Er erzeugt ein Testbild mit Ton ohne externes Signal.

Es können mehrere Eingänge vorhanden sein — der erste in der Liste ist der Haupteingang, die übrigen sind Reserveeingänge (*Quellenredundanz*).

1.5.3 Schritt 3. Starten und prüfen

Neue Objekte werden angehalten angelegt — heben Sie deren Pause auf: mit der Pausenschaltfläche in der Eingangszeile auf der Registerkarte „Eingänge“ („Eingang fortsetzen“) und mit der Pausenschaltfläche des Streams in dessen Listenzeile oder in der Kopfzeile des Editors („Stream fortsetzen“). Der Stream geht in den Zustand „Auf Sendung“ über.

Prüfen Sie den Empfang im Statistikfenster des Streams (*Stream-Statistik*):

- Status und aktiver Eingang — in der Kopfzeile des Fensters;
- Abschnitt „MPEG-TS-Prüfer“ — alle Attribute müssen gesetzt sein (*Gültigkeitsprüfung*);
- Eingangsbitrate und Diagramme;
- das Bild des Streams ist im Abschnitt „Snapshot“ des Statistikfensters sowie in der Ansicht „Mosaik“ sichtbar (*Mosaik*).

1.5.4 Schritt 4. Stream ausliefern

Variante A — OTT (HLS). Setzen Sie auf der Registerkarte „OTT“ des Stream-Editors „HLS-Dienst“ auf den Wert OTT / HLS und speichern Sie. Die URL-Vorlage erscheint im Statistikfenster (Abschnitt „Auslieferungs-URLs“) — kopieren Sie sie und ersetzen Sie die Teile login/password durch die Zugangsdaten eines Client-Logins, das in der Ansicht „Benutzer / Logins“ (*Benutzer / Logins*) angelegt wurde; danach lässt sich die URL im Player öffnen. Auslieferungsmodi, DASH und Low-Latency HLS sind in *OTT und DVR* beschrieben.

Variante B — Transportausgang. Fügen Sie auf der Registerkarte „Ausgänge“ einen Ausgang des benötigten Protokolls hinzu (UDP, SRT, PS1, RIST und weitere — *Planung und Datenübertragungsprotokolle*), geben Sie die Adresse des Empfängers an und heben Sie die Pause des Ausgangs auf. Es können mehrere Ausgänge vorhanden sein — der Stream wird dann gleichzeitig an alle Empfänger gesendet.

1.5.5 Wie geht es weiter

- Auswahl der Übertragungsprotokolle und Planung des Auslieferungsschemas — *Planung und Datenübertragungsprotokolle*.
- Verarbeitung des Streams auf dem Knoten: Filterung, Bitrate, Analyse, Transcodierung — *Streamer: Implementierungsdetails*.
- Archivaufzeichnung und Wiedergabe (DVR) — *DVR: Archiv des Streams*.
- Zusammenschluss von Knoten zu einem Netzwerk und Überwachung des Standorts — *Meshwork*.
- Referenz zu allen Ansichten der Weboberfläche — *Weboberfläche*.

1.6 Meshwork

Meshwork ist ein Mechanismus zur Zusammenfassung von Perfect Streamer-Servern zu einem einheitlichen Netzwerk aus Knoten. Knoten derselben Domain finden sich automatisch und tauschen Dienstinformationen über das für dezentrale Netzwerke charakteristische gossip-Protokoll aus. Die Zusammenfassung der Knoten bietet drei Möglichkeiten:

- **Live-Status der Knoten.** Auf jedem Knoten ist im Admin-Bereich der Zustand aller übrigen Knoten des Netzwerks sichtbar: wer in Verbindung steht, wer schweigt, wer hinter NAT steht und wann der letzte Kontakt war. Es ist nicht nötig, jeden Server einzeln zu öffnen. Ein Knoten, der keine Daten mehr austauscht, geht in den Zustand offline über — dies ist sofort auf allen Knoten sichtbar.
- **Gemeinsamer Ressourcenkatalog (Bibliothek).** Die auf den Knoten laufenden Ein- und Ausgänge UDP / RTP / Pro-MPEG / RIST / PS1 / SRT gelangen in einen gemeinsamen Katalog, der auf allen Knoten der Domain sichtbar ist: So lassen sich bequem eine freie Adresse und ein freier Port auswählen, Multicast-Kollisionen vermeiden und feststellen, wo der benötigte Stream bereits gesendet oder empfangen wird. Innerhalb der Domain wird der Peer zwischen den Knoten per Autologin aufgebaut — ohne manuelle Eingabe von Logins und Passwörtern auf beiden Seiten.
- **Replikation der Alerts über die Domain.** Die Alerts eines Knotens sind auf allen Knoten derselben Domain sichtbar: Dem Diensthabenden genügt ein einziges Fenster, um den gesamten Standort zu überwachen. Externe Benachrichtigungen müssen nur auf einem Knoten der Domain konfiguriert werden — die Benachrichtigungen werden dann für alle Knoten dieser Domain zugestellt.

Meshwork ist eine Schicht für Sichtbarkeit und Abstimmung: Es überträgt keine Streams zwischen Servern und ändert die Broadcast-Konfiguration nicht automatisch. Das Zustellungsschema bauen Sie nach wie vor selbst, und Meshwork hilft, es zu planen und zu beobachten.

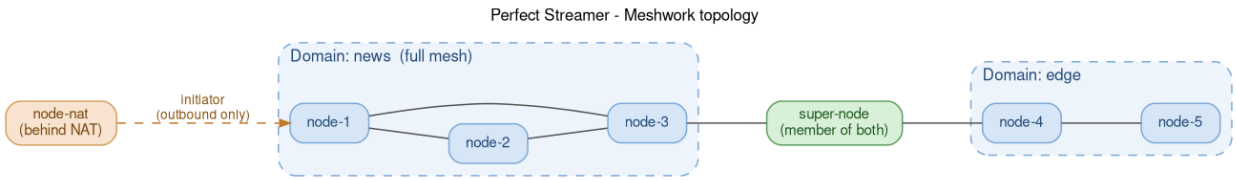


Abb. 1: Meshwork-Topologie: Knoten derselben Domain bilden ein vollständiges Netz direkter Verbindungen; ein Knoten hinter NAT verbindet sich als Initiator; eine super-node gehört zu zwei Domains, verbindet diese aber nicht miteinander.

1.6.1 Zentrale Begriffe

Knoten
: Ein einzelner Perfect Streamer-Server im Netzwerk. Jeder Knoten hat einen kurzen eindeutigen Namen, an dem ihn die anderen erkennen. Ohne Namen und Signatur-Secret nimmt ein Knoten nicht am Netzwerk teil.

Domain
: Eine logische Gruppe von Knoten und zugleich eine **Isolationsgrenze**. Innerhalb einer Domain finden sich Knoten automatisch, teilen einen gemeinsamen Ressourcenkatalog und replizieren Alerts. Zwischen verschiedenen Domains gibt es keinen Austausch — die Domains sind unabhängig.

Automatische Erkennung (mesh)
: Es genügt, den Knoten dieselbe Domain und mindestens eine Verbindung zuzuweisen — danach finden sie sich selbst, ohne jeden Nachbarn manuell aufzuzählen. Innerhalb der Domain entsteht ein vollständiges Netz direkter Verbindungen: Alle Knoten der Domain werden zu direkten Nachbarn, Zwischenketten werden nicht gebildet.

  Die Verbindung wird über HTTP eingetragen, der weitere Austausch erfolgt jedoch über HTTPS. Für ein öffentliches Netzwerk ist konfiguriertes HTTPS zwingend erforderlich; für ein internes Netzwerk (intranet address) ist HTTPS nicht erforderlich.

Explizite Verbindung
: Ein Peer kann manuell angegeben werden — mit Adresse, Namen und einem gemeinsamen Secret. Eine solche Verbindung funktioniert immer, auch über einfaches HTTP, und erfordert keine gemeinsame Domain. Sie eignet sich für die punktuelle Verbindung zweier Standorte oder das Anbinden eines Knotens hinter NAT.

Super-node
: Ein einzelner Knoten kann gleichzeitig zu mehreren Domains gehören. Er sieht die Ressourcen und Knoten jeder seiner Domains, verbindet die Domains aber nicht miteinander — die Isolation bleibt erhalten. In der Netzwerkkarte wird ein solcher Knoten als `super` gekennzeichnet.

Alarme (Alerts)
: Der integrierte Alerting-Dienst überwacht Streams, die Hardware des Knotens, den DVB-Empfang und die DVR-Aufzeichnung und löst Alerts aus, wenn etwas die festgelegten Grenzen überschreitet. In einem Netzwerk mit aktivierter Replikation ist auf jedem Knoten ein zusammengefasster Satz von Alerts der gesamten Domain sichtbar (siehe *Benachrichtigungen (Alerter)*).

Aufbau eines Meshwork-Netzwerks

Die Konfiguration von Meshwork läuft darauf hinaus, dass jeder Knoten seinen eigenen Namen und seine Domain kennt und mindestens einen erreichbaren Peer hat. Anschließend baut die Domain die übrigen Verbindungen selbst auf. Alle Aktionen werden im Web-Adminbereich durchgeführt; das Ändern der Meshwork-Einstellungen erfordert die Administratorrolle. Die genauen Felder und ihre Position sind im Abschnitt *Weboberfläche* beschrieben.

Anforderungen und Erreichbarkeit

- **Netzwerk-Erreichbarkeit.** Ein Knoten muss in der Lage sein, eine TCP-Verbindung zum administrativen Webserver eines Peers herzustellen. Der Port wird der Adresse des Peers („Host:Port“) entnommen oder dem öffentlichen Port, den der Peer über sich selbst meldet (*Knotenidentität*) — es ist immer der Port, auf dem sein Webserver konfiguriert ist, und bei aktiviertem HTTPS auf dem Peer dessen TLS-Port. Die Werte für eine frische Installation sind in *Anfangseinstellungen* aufgeführt; der benötigte Port eines Peers muss von den anderen Knoten aus geöffnet sein.
- **HTTPS für die automatische Erkennung über das Internet.** Wenn automatisch erkannte Domain-Knoten über öffentliche (Internet-)Adressen kommunizieren, müssen sie untereinander eine sichere Verbindung (HTTPS) herstellen, um die Authentizität des jeweils anderen über das offene Netz sicher zu bestätigen. Aktivieren Sie HTTPS daher zumindest auf den Knoten, über die die Erkennung erfolgt. Die Aktivierung von HTTPS wird nach einem **Neustart** des Servers wirksam. Wenn Knoten nur über öffentliche Adressen erreichbar sind und keiner per HTTPS läuft, werden direkte automatisch erkannte Verbindungen nicht authentifiziert: Der Knoten ist dennoch sichtbar (über Relay von einem Peer), seine direkte Verbindung wird jedoch mit einer Ursache gekennzeichnet (siehe *Netzwerkkarte und Ressourcenkatalog*).
- **Lokales Netzwerk — HTTPS nicht erforderlich.** Wenn Domain-Knoten über private Adressen eines einzigen vertrauenswürdigen Netzwerks kommunizieren (`10.x.x.x`, `172.16`–`31.x.x`, `192.168.x.x` sowie `127.x.x.x` und `169.254.x.x`), ist eine sichere Verbindung für die automatische Erkennung nicht erforderlich — die Knoten bestätigen die Authentizität auch über einfaches HTTP. Ein Knoten wird anhand seiner IP-Adresse als lokal erkannt, geben Sie Peers im lokalen Netzwerk daher gezielt über IP-Adressen und nicht über Domainnamen an. HTTPS kann dabei aktiviert werden — es wird verwendet, sofern verfügbar.
- **Eine explizite Verbindung funktioniert auch über HTTP** — die HTTPS-Anforderung gilt für sie in keinem Fall.
- **Name und Secret sind obligatorisch.** Ein Knoten nimmt nur dann am Netzwerk teil, wenn für ihn ein Name und ein Signatur-Secret festgelegt sind. Ohne Namen oder ohne Secret nimmt ein Knoten nicht an Meshwork teil.
- **Namensbeschränkungen.** Ein Knotenname umfasst 2 bis 16 Zeichen, ein Domainname ist höchstens 16 Zeichen lang; erlaubt sind lateinische Buchstaben, Ziffern und die Zeichen `_`, `-`, `.`. Leerzeichen und andere Zeichen sind nicht zulässig (zum Beispiel lässt sich die Domain „Alex Home“ nicht festlegen — verwenden Sie `Alex_Home`).

Knotenidentität

In den Servereinstellungen wird festgelegt, als was sich der Knoten im Netzwerk darstellt:

- **Knotenname** — ein kurzer, eindeutiger Name, an dem die anderen den Knoten erkennen. Ein leerer Name bedeutet, dass der Knoten nicht am Netzwerk teilnimmt.
- **Notiz** — eine beliebige Beschriftung des Knotens (zum Beispiel „Moskau, Rack 3“), wird im Adminbereich zur Übersicht angezeigt und hat keinen Einfluss auf den Netzwerkbetrieb.
- **Domain** — Knoten mit derselben Domain finden sich automatisch, teilen sich einen gemeinsamen Ressourcenkatalog und replizieren Alerts. Ein leerer Wert deaktiviert die automatische Erkennung (der Knoten kann nur über die explizite Peer-Liste arbeiten).
- **Öffentliche Ports** (HTTP und HTTPS) — die Ports, die der Knoten den Peers für die Rück-Erreichbarkeit bei Adressübersetzung (NAT) mitteilt. Sind sie nicht festgelegt, meldet der Knoten seine eigenen Webserver-Ports.
- **Zusätzliche Domains** (super-node-Modus) — der Knoten nimmt an der automatischen Erkennung jeder aufgeführten Domain gleichberechtigt mit der Hauptdomain teil. Die Ressourcen bleiben je Domain isoliert — der Knoten vermischt sie nicht.

Peers und Secrets

Hier wird festgelegt, mit welchen Peers der Knoten eine direkte Verbindung unterhält und wie die Seiten einander überprüfen. Anfragen zwischen Knoten werden mit einem gemeinsamen Secret signiert, daher kommt ohne übereinstimmende Secrets keine Verbindung zustande. Die Länge jedes Secrets beträgt 16 bis 128 Zeichen.

Die Peer-Liste besteht aus Einträgen der Form „Adresse — Name — Secret“:

- **Adresse** eines Peers — der „Host:Port“ seines Adminbereichs. Ein Eintrag darf ohne Adresse sein: Ein solcher Peer wird von diesem Knoten nie kontaktiert (er wählt sich nur selbst ein) und wird anhand von Namen und Secret erkannt — dies wird für einen Knoten hinter NAT verwendet, der nur eine Verbindung initiieren kann.
- **Name** eines Peers muss exakt mit dem Namen übereinstimmen, der auf dem Peer selbst festgelegt ist.
- **Gemeinsames Secret** der Verbindung mit diesem Peer muss an beiden Enden identisch sein.

Neben der Peer-Liste besitzt ein Knoten sein **eigenes Signatur-Secret**: Damit signiert der Knoten seine ausgehenden Anfragen. Ein Peer überprüft die Signatur mit dem Secret, das er für diesen Knoten speichert, daher muss das eigene Secret des Knotens mit dem Secret übereinstimmen, das bei jedem Peer über ihn hinterlegt ist.

Bemerkung: Eine praktische Regel: Legen Sie für einen einfachen Standort ein gemeinsames Secret fest und verwenden Sie es sowohl als eigenes Signatur-Secret auf jedem Knoten als auch als Secret in jedem Peer-Eintrag — dann authentifiziert sich jedes Knotenpaar auf dieselbe Weise. Für eine strengere Trennung können Sie pro Verbindung ein separates Secret führen.

Schnellstart

Variante A. Zwei Knoten direkt (explizite Verbindung)

Das einfachste Meshwork sind zwei manuell verbundene Server.

1. Legen Sie auf jedem Knoten einen eindeutigen kurzen Namen fest (zum Beispiel `node1` und `node2`).
2. Fügen Sie auf jedem Knoten einen Peer-Eintrag hinzu: die Adresse seines Adminbereichs, seinen Namen und das gemeinsame Secret.
3. Legen Sie auf jedem Knoten das eigene Signatur-Secret fest. Am einfachsten ist es, ein einziges gemeinsames Secret für beide Knoten zu führen.
4. Öffnen Sie die Netzwerkkarte im Adminbereich. Innerhalb weniger Sekunden sehen beide Knoten einander im Zustand „online“.

Eine Domain muss dabei nicht festgelegt werden, aber ohne gemeinsame Domain erhalten Sie nur die gegenseitige Sichtbarkeit der Knoten. Der gemeinsame Ressourcenkatalog und die Alert-Replikation funktionieren nur zwischen Knoten mit derselben Domain — falls Sie sie benötigen, legen Sie für die Knoten eine gemeinsame Domain fest (Variante B).

Variante B. Eine Domain mit automatischer Erkennung

Bei mehr als zwei Knoten ist es bequemer, ihnen eine gemeinsame Domain zuzuweisen, als jede Verbindung manuell aufzulisten.

1. Legen Sie auf jedem Knoten einen eindeutigen Namen und die gemeinsame Standort-Domain fest.
2. Wenn Domain-Knoten über das Internet kommunizieren (über öffentliche Adressen), aktivieren Sie HTTPS auf mindestens ein bis zwei Domain-Knoten und starten Sie diese neu (siehe *Anforderungen und Erreichbarkeit*). In einem lokalen Netzwerk kann dieser Schritt übersprungen werden.
3. Verbinden Sie die Knoten über einen „Einstiegspunkt“: Geben Sie auf jedem Knoten mindestens einen bereits funktionierenden Peer an (Adresse, Name, gemeinsames Secret), wie in Variante A. Die übrigen Domain-Knoten findet der Knoten selbst.
4. Öffnen Sie die Netzwerkkarte. Innerhalb weniger Sekunden erscheinen darauf alle Domain-Knoten: die explizit angegebenen als permanent, die automatisch gefundenen als automatisch erkannt.

Knoten hinter NAT

Ein Knoten in einem privaten Netzwerk, der von außen nicht direkt erreichbar ist, kann als **Initiator** vollwertig am Netzwerk teilnehmen. Er stellt selbst eine Verbindung zu einem erreichbaren Peer her, und über diese eine Verbindung fließen die Daten in beide Richtungen — der Knoten meldet sich sowohl selbst als auch erhält Informationen über andere. Für einen solchen Knoten genügt es, mindestens einen erreichbaren Peer anzugeben.

Informationen über einen NAT-Knoten verbreiten sich automatisch über die Domain: Ein Peer, der mit ihm online ist, leitet seinen Zustand per Relay an die anderen weiter. Bei entfernten Peers wird ein solcher Knoten als über Relay erhalten angezeigt, jedoch im Zustand „online“.

Damit ein Knoten hinter NAT auch für eingehende Verbindungen erreichbar ist (zum Beispiel als Einstiegspunkt der Domain dient), konfigurieren Sie die Veröffentlichung des öffentlichen Ports (siehe *Knotenidentität*) und leiten Sie diesen Port auf dem Router/der Firewall an den administrativen Webserver des Knotens weiter. Auf Seiten der Peers wird ein solcher Knoten als Eintrag „Name + Secret ohne Adresse“

geführt: Der Peer versucht nicht vergeblich, die private Adresse zu erreichen, sondern erkennt den Knoten anhand von Namen und Secret, wenn dieser sich selbst verbindet.

Typische Probleme

- **Die Knoten sehen einander nicht.** Prüfen Sie: Bei beiden ist ein Knotenname festgelegt; jeder hat ein Signatur-Secret und es stimmt mit dem überein, das beim Peer über ihn hinterlegt ist; die Adresse des Peers ist korrekt und sein Admin-Port ist vom anderen Knoten aus geöffnet.
- **Ein Peer ist im Zustand „offline".** Der Knoten hat das Wartefenster überschritten: Prüfen Sie das Netzwerk und die Funktionsfähigkeit seines Webservers. Stellen Sie bei einem Knoten hinter NAT sicher, dass er die Verbindung zu einem erreichbaren Peer selbst initiiert oder dass Portveröffentlichung und Portweiterleitung konfiguriert sind.
- **Ein automatisch erkannter Peer ist mit einer Ursache gekennzeichnet, die direkte Verbindung funktioniert nicht.** In der Domain lässt sich die Authentizität der Knoten nicht bestätigen. Wenn sie über das Internet kommunizieren, aktivieren Sie HTTPS auf mindestens einem Domain-Knoten und starten Sie ihn neu; prüfen Sie in einem lokalen Netzwerk, dass Peers über IP-Adressen und nicht über Domainnamen angegeben sind. Über eine explizite Verbindung (permanente Knoten) funktioniert die Verbindung auch ohne HTTPS.
- **Falsche NAT-Kennzeichnung.** Wenn Knoten über Domainnamen und nicht über IP angegeben sind, lässt sich die NAT-Kennzeichnung möglicherweise nicht anhand der Adresse ermitteln — dies beeinträchtigt den Betrieb nicht. Verwenden Sie für eine zuverlässige NAT-Diagnose IP-Adressen.
- **Der Domainwechsel hat die Karte „zurückgesetzt".** Das Ändern der Domain oder der Menge zusätzlicher Domains baut die Mitgliedschaft neu auf: Automatisch erkannte Verbindungen und der gemeinsame Katalog werden vorübergehend zurückgesetzt und neu aufgebaut. Das ist normal.

Netzwerkkarte und Ressourcenkatalog

Meshwork bietet zwei Übersichtsseiten im Admin-Bereich: eine Karte der Netzwerkknoten und einen gemeinsamen Katalog der belegten Netzwerkressourcen. Beide werden automatisch aktualisiert und fassen domänenweit Informationen zusammen, die andernfalls von jedem Server einzeln von Hand gesammelt werden müssten.

Netzwerkkarte

Die Karte zeigt die Zusammensetzung des Netzwerks — eine Zeile pro bekanntem Knoten, einschließlich desjenigen, von dem aus Sie sie betrachten. Die Knoten prüfen etwa alle 5 Sekunden die Verbindung untereinander, und die Ansicht aktualisiert sich von selbst. Ist Meshwork nicht konfiguriert, bleibt die Liste leer.

Was die Zeile eines Knotens zeigt:

- **Knotenname** und die Adresse seines Admin-Bereichs.
- **Eigener Knoten** — die Zeile des Knotens, auf dem Sie die Karte betrachten; sie ist besonders gekennzeichnet und wird immer als „online" angezeigt.
- **Verbindungstyp** — dauerhaft (ein Knoten aus der expliziten Peer-Liste) oder automatisch erkannt (über eine gemeinsame Domain per automatischer Erkennung gefunden).

- **Status** — „online“ oder „offline“. Kommen etwa 15 Sekunden lang keine Bestätigungen von einem Knoten, wird er als „offline“ markiert. Ein Neustart eines Peers wird automatisch erkannt, daher ist ein kurzes Flackern des Status direkt nach seinem Neustart normal.
- **Letzter Kontakt** — wann zuletzt eine erfolgreiche Verbindung bestand.
- **Domain** — die Domain oder Domains des Knotens. Eine super-node hat mehrere, und der Knoten ist zusätzlich als `super` gekennzeichnet.
- **NAT** — ein Hinweis darauf, dass auf dem Weg zum Knoten eine Adressübersetzung erkannt wurde.
- **Relay / Über** — werden die Informationen über Erreichbarkeit und NAT eines Knotens nicht direkt, sondern über einen benachbarten Knoten bezogen, wird die Zeile als „Relay“ markiert, und im Feld „über“ wird der Name dieses Peers angezeigt. So werden etwa Knoten hinter NAT dargestellt, die dieser Knoten nicht direkt erreichen kann. Sobald eine direkte Verbindung zustande kommt, hat sie Vorrang vor den weitergeleiteten Informationen.
- **Grund** — erscheint, wenn ein automatisch erkannter Peer direkt erreichbar ist, sich aber nicht authentifizieren lässt. In der Regel bedeutet dies, dass die Knoten über öffentliche Adressen kommunizieren, in der Domain jedoch kein Knoten mit aktiviertem HTTPS vorhanden ist, über den die Knoten die Echtheit des jeweils anderen bestätigen (siehe *Anforderungen und Erreichbarkeit*). Die Kennzeichnung bedeutet, dass die **direkte** Verbindung nicht funktioniert; der Knoten selbst kann dabei über Relay weiterhin „online“ bleiben. In einem lokalen Netzwerk (Knoten mit privaten IP-Adressen) tritt diese Kennzeichnung nicht auf.

Gemeinsamer Ressourcenkatalog

Eine separate Seite im Admin-Bereich zeigt einen verteilten Katalog der belegten Netzwerkressourcen — alle auf den Knoten der Domain laufenden UDP- / RTP- / Pro-MPEG- / RIST-Eingänge und -Ausgänge. Dies ist eine domänenweite Liste der belegten Adressen und Ports; damit lässt sich bequem eine freie Ressource auswählen und erkennen, wo welcher Stream bereits genutzt wird.

Für jeden Eintrag zeigt der Katalog das Protokoll und die Richtung (Eingang/Ausgang), den Quellknoten, die Domain, die Adresse/Gruppe/den Port, das VLAN, bei Bedarf die Quelle für source-specific multicast und eine Notiz.

Derselbe Katalog beantwortet auch die umgekehrte Frage — wer sonst noch mit der Adresse eines bestimmten Ein- oder Ausgangs arbeitet: neben der Adresse befindet sich die Schaltfläche „In der Bibliothek anzeigen“ (*Bibliothek — dieser Feed*).

Die Veröffentlichung wird über zwei zusätzliche Felder am Eingang/Ausgang selbst gesteuert (sie werden nur bei den Transporten angezeigt, die in den Katalog gelangen — UDP / RTP / Pro-MPEG / RIST / PS1 / SRT; die Bezeichnungen entsprechen der Oberfläche): Peer note (meshwork library) — eine öffentliche Notiz für den Katalog (ist sie leer, wird die gewöhnliche Notiz des Eintrags verwendet), und Peer private — der Ausschluss des Eintrags von der Veröffentlichung.

Verbindungstyp PS1 / SRT

Neben UDP / RTP / Pro-MPEG / RIST gelangen auch *PS1*- und *SRT*-Verbindungen in den Katalog. Eine solche Verbindung hat zwei Seiten: die rufende (baut die Verbindung zu einer entfernten Adresse auf) und die lauschende (nimmt die Verbindung an). Für die rufende Seite zeigt der Katalog den Verbindungstyp an:

- **intern** — der Eingang/Ausgang ist so konfiguriert, dass er sich als Domain-Mitglied verbindet (Domänen-Auto-Login aktiviert): Er verbindet sich mit einem anderen Knoten des Netzwerks und bestätigt seine Zugehörigkeit zur Domain;
- **extern** — es gibt keine Bestätigung als Domain-Mitglied: Dies ist eine gewöhnliche Verbindung zu einem beliebigen Endpunkt. Selbst wenn sich unter der angegebenen Adresse ein Knoten des Netzwerks befindet, gilt die Verbindung ohne Domänen-Auto-Login als extern — der Typ spiegelt wider, wie die Verbindung konfiguriert ist, und nicht, ob der Empfänger zum Netzwerk gehört.

Dies ist eine Hinweiskennzeichnung: Sie beeinflusst die Übertragung selbst nicht, hilft aber, bestätigte domäneninterne Verbindungen von externen zu unterscheiden. Bei den lauschenden Seiten und bei der UDP- / RTP-Familie wird der Typ nicht angezeigt.

Benachrichtigungen (Alerter)

Der Alerter ist ein integrierter Benachrichtigungsdienst. Er überwacht den Zustand von Streams, der Node-Hardware, des DVB-Empfangs und der DVR-Aufnahme und löst **Alerts** (Vorfälle) aus, wenn etwas die festgelegten Grenzen überschreitet. Die Liste aktiver Alerts ist im Admin-Bereich sichtbar; in einem Netzwerk mit aktivierter domänenweiter Replikation zeigt jeder Node einen zusammengefassten Alert-Satz für die gesamte Domain.

Der Alerter funktioniert auch auf einem einzelnen Node — Meshwork wird nur für die Replikation von Alerts über die Domain benötigt.

Wie ein Alert aufgebaut ist

- Jeder Vorfall hat eine **stabile Kennung**: das erneute Auftreten desselben Problems ist derselbe Vorfall, kein neuer.
- Ein Vorfall durchläuft Zustände: **ausgelöst** (erstmals bemerkt), **aktualisiert** (erneut mit einer wesentlichen Änderung bemerkt), **behoben** (das Problem ist weg). Der aktive Satz sind die Vorfälle, die noch nicht behoben wurden.
- Jeder Alert hat einen **Schweregrad** (in aufsteigender Reihenfolge: Information, Warnung, Fehler, kritisch, erfordert Administrator-Eingriff), einen **Code** (die numerische Art des Vorfalls — siehe Katalog unten), eine **Quelle** (Stream, Node, Storage), einen **Quell-Node** (wer den Alert ausgelöst hat) und einen menschenlesbaren Objektnamen.
- Die meisten Alerts sind **pegelbasiert**: der Dienst bewertet die Bedingung periodisch neu und löst den Alert selbst aus oder behebt ihn. Ein Teil der Alerts, die einen Administrator-Eingriff erfordern, ist **haftend** — sie werden nicht von selbst behoben, bis die Ursache beseitigt ist (sie werden manuell behoben, siehe *Alerts anzeigen und beheben*).

Die Liste aktiver Alerts wird im Speicher gehalten und beim Neustart des Dienstes geleert.

Steuerung auf Stream-Ebene. Jeder Stream hat einen Schalter, der seine Alerts vollständig stummschaltet (zusammen mit den Alerts seiner Inputs und Outputs), und eine Alert-Verzögerung: eine Quelle, die sich innerhalb der festgelegten Zeit erholt, löst keinen Alert aus — das glättet kurzzeitiges Flattern.

Alerter-Einstellungen

Die Einstellungen befinden sich im Benachrichtigungsbereich und erfordern die Administrator-Rolle. Die genauen Felder sind im Abschnitt *Weboberfläche* beschrieben; nachfolgend, was konfiguriert werden kann.

- **Hauptschalter** des Benachrichtigungsdienstes (standardmäßig aktiviert).
- **Domänenweite Replikation** (standardmäßig aktiviert): ein Node teilt seine Alerts mit Nodes derselben Domain und empfängt deren Alerts — jeder Node zeigt einen zusammengefassten Satz. Nur wirksam, wenn eine Domain festgelegt ist.
- **Replikation von System-Alerts** (standardmäßig deaktiviert): Alerts über die Node-Hardware (Host-CPU und -Speicher, Streamer-Prozess und Transcoder, Netzwerk- und GPU-Auslastung, Worker-Threads und das DVB-Frontend — Codes 16–23 und 40–45) sind standardmäßig lokal; aktivieren Sie dies, damit sie ebenfalls über die Domain repliziert werden. Storage-Alerts (Codes 24–25) bleiben immer lokal.
- **Stream-Schwellenwerte**: Timeout bei fehlenden Daten (Standard 15 s), minimal zulässige Bitrate (Prüfung standardmäßig deaktiviert), Schwellenwert für Continuity-Counter-Fehler über ein 10-s-Fenster (Warn- und Fehlerpegel).
- **Node-Ressourcen-Schwellenwerte** — Paare „Warnung / Fehler“ in Prozent für Host-CPU und -Speicher, den Streamer-Prozess und Transcoder, die Auslastung von Netzwerkschnittstelle und GPU sowie die Füllung des DVR-Storage. Die Host-Schwellenwerte sind höher angesetzt als die des Streamer-Prozesses und der Transcoder, damit eine einzelne Komponente früher warnt, als die gesamte Maschine gesättigt wird. Der Wert 0 deaktiviert den entsprechenden Pegel.
- **DVB-Frontend-Schwellenwerte** — Signalpegel, SNR/Qualität, Bitfehlerrate und unkorrigierte Blöcke; werden auf einen Adapter angewendet, der das Signal erfasst hat (Lock).
- **DVR-Aufnahme-Schwellenwerte** — die Anzahl aufeinanderfolgender fehlgeschlagener Schreibvorgänge, der Multiplikator für ein Aufnahme-Stocken, der Anteil der Archiv-Lücken und die Lese-/Schreibzeit der Chunks auf die Festplatte.

Katalog der Alert-Codes

Der Code ist die numerische Art des Vorfalls, praktisch für Gruppierung und Lokalisierung. Die Spalte „Replikation“ gibt an, ob der Alert über die Domain repliziert wird (bei aktivierter Replikation) oder für den Node lokal bleibt.

Streams: Zustand und Arbeitszyklus (Quelle — Stream)

Code	Ereignis	Replikation
1	ein Input oder Output ist in den Fehlerzustand übergegangen	domänenweit
2	nicht behebbarer Arbeitszyklus-Fehler, manueller Reset erforderlich (erfordert Administrator-Eingriff)	domänenweit
3	ein laufender Stream liefert länger als das Timeout keine Daten	domänenweit
4	die Stream-Bitrate bleibt unter dem Minimum	domänenweit
13	der Stream ist pausiert: kein nutzbarer Input mehr vorhanden (erfordert Administrator-Eingriff)	domänenweit
14	ein Input oder Output hat sich selbst geparkt und erholt sich beim Wiederholen nicht (erfordert Administrator-Eingriff)	domänenweit
27	der Stream läuft auf dem Backup-Input — eine Umschaltung ist erfolgt	domänenweit
39	hohe CPU-Auslastung durch den Worker-Thread des Streams	domänenweit

Transportstrom-Konformität (TR 101 290-Analyzer) (Quelle — Stream; alle domänenweit repliziert)

Code	Ereignis
5	wiederholte PCR-Diskontinuitäten (TR 101 290, 2.3)
6	PCR-Wiederholintervall größer als 40 ms (2.3a)
7	PAT-Wiederholintervall größer als 500 ms (1.3)
8	PMT-Wiederholintervall größer als 500 ms (1.5)
9	PTS-Wiederholintervall größer als 700 ms (2.5)
10	PCR-Genauigkeit schlechter als 500 ns (2.4)
11	T-STD-Puffer-Über- oder -Unterlauf beim Video (3.3)
12	PCR-Drift größer als 30 ppm (historisch: der Alert wird nicht mehr ausgelöst, die Drift wird als Metrik angezeigt — *Fortlaufende Messungen*)
15	Continuity-Counter-Fehler über dem Schwellenwert innerhalb eines kurzen Fensters (1.4)
46	CRC-Fehler in PSI-Tabellen (2.2)
47	das transport_error_indicator-Flag ist gesetzt (2.1)
48	Wiederholintervall der SI-Tabelle überschritten (SDT / EIT / TDT / NIT)
49	Gesamturteil: der Stream entspricht nicht TR 101 290

Ein Teil der Analyzer-Codes erscheint nur bei aktivierten Analyseoptionen (Tiefenanalyse, Analyse von PTS-Diskontinuitäten, Analyse des T-STD-Puffers) und gilt nur für Streams mit konstanter Bitrate; das Gesamturteil (49) wird nur bei zuverlässig konstanter Bitrate gefällt. Näheres im Abschnitt *Analysator*.

Node-Ressourcen und -Hardware (Quelle — System)

Code	Ereignis
16	Host-CPU-Auslastung über dem Schwellenwert
17	Host-Speichernutzung über dem Schwellenwert
18	CPU des Streamer-Prozesses über dem Schwellenwert
19	Speicher des Streamer-Prozesses über dem Schwellenwert
20	Gesamt-CPU der Transcoder über dem Schwellenwert
21	Gesamtspeicher der Transcoder über dem Schwellenwert
22	Auslastung der Netzwerkschnittstelle über dem Schwellenwert (pro Schnittstelle)
23	GPU-Auslastung über dem Schwellenwert (pro Gerät)
24	DVR-Storage-Füllung über dem Schwellenwert (pro Storage)
25	das Dateisystem des DVR-Storage ist nicht verfügbar (erfordert Administrator-Eingriff)
40	hohe CPU-Auslastung durch den Worker-Thread des DVB-Adapters
41	das DVB-Frontend hat das Signal (lock) verloren
42	DVB-Signalpegel unter dem Schwellenwert
43	SNR/Qualität des DVB-Signals unter dem Schwellenwert
44	DVB-Bitfehlerrate (BER) über dem Schwellenwert
45	unkorrigierte DVB-Blöcke über dem Schwellenwert

Alle Alerts dieser Gruppe sind standardmäßig lokal. Die Codes 16–23 und 40–45 können mit dem Schalter für die Replikation von System-Alerts über die Domain repliziert werden; die Codes 24–25 bleiben immer lokal. Die Schwellenwert-Alerts des DVB-Frontends (42–45) werden nur auf einem Adapter ausgewertet, der das Signal erfasst hat; der Code 41 wird beim Verlust des Locks ausgelöst. BER (44) ist standardmäßig deaktiviert, da die Rohskala vom Empfänger abhängt (siehe *DVB-Empfänger*).

Meshwork und Lebenszyklus (Quelle — Netzwerk/System; lokal)

Code	Ereignis
26	ein Netzwerk-Node ist verstummt — bestätigt die Verbindung nicht mehr
28	der Streamer-Prozess hat den Start abgeschlossen (kurzlebige Benachrichtigung)
37	zwei Nodes mit demselben Namen in einer Domain (erfordert Administrator-Eingriff)

DVR-Aufnahme-Gesundheit (Quelle — Stream; lokal)

Code	Ereignis
29	DVR-Segment- oder Index-Schreibvorgänge schlagen fehl — das Archiv wird nicht geschrieben
30	die DVR-Aufnahme ist ins Stocken geraten: Input ist vorhanden, aber ein neues Segment wird lange nicht gespeichert
31	das DVR-Archiv ist beeinträchtigt — Lücken in der jüngsten Abdeckung
32	der Index des DVR-Archivs ist beschädigt (erfordert Administrator-Eingriff)
50	hohe Lese-/Schreibzeit von DVR-Chunks auf die Festplatte

Konfiguration, Transcoder, OTT und Zertifikate (Quelle — System/Stream)

Code	Ereignis	Replikation
33	die Hauptkonfiguration konnte beim Start nicht geladen werden, der Dienst startete mit der Ersatzkonfiguration (erfordert Administrator-Eingriff)	domänenweit
34	im low-latency HLS/DASH-Modus wurden Audiospuren in einem nicht unterstützten Codec verworfen (unterstützt werden AAC und AC-3)	domänenweit
35	das HTTPS-Zertifikat konnte über den integrierten ACME-Client nicht ausgestellt oder erneuert werden	domänenweit
36	ein konfigurierter Transcoder konnte beim Start nicht geladen werden — keine ausführbare Datei oder kein Gerät (erfordert Administrator-Eingriff)	domänenweit
38	der Stream erfordert einen Transcoder (Software, NVIDIA oder Intel VPL), der auf diesem Node nicht verfügbar ist (erfordert Administrator-Eingriff)	lokal

Conditional-Access-Modul (CI/CAM, EN 50221) (Quelle — System; lokal)

Code	Ereignis
51	das CAM-Modul wurde entfernt (erfordert Administrator-Eingriff)
52	kein Abonnement: das Modul meldet, dass das Programm nicht bezahlt ist
53	CI/CA-Fehler — ein nicht behebbarer Kanal- oder Conditional-Access-Fehler

Alerts anzeigen und beheben

Der Admin-Bereich hat einen Benachrichtigungsabschnitt mit einer Liste von Alerts. Er kann nach Satz (nur aktive / nur behobene / alle), nach Schweregrad, nach Quelltyp und -kennung, nach Code, nach Domain und nach Zeit gefiltert werden, außerdem kann nach Nachrichtentext gesucht werden. Standardmäßig werden sowohl aktive als auch kürzlich behobene Einträge angezeigt (das Übergangsprotokoll), daher kann die Gesamtzahl der Zeilen die Zahl der aktiven Vorfälle übersteigen — für den aktuellen aktiven Satz wählen Sie den Modus „nur aktive“.

Sie müssen den Zustand nicht manuell abfragen: der Admin-Bereich empfängt Änderungen am aktiven Satz in Echtzeit und aktualisiert die Liste selbst, sobald ein Alert ausgelöst, aktualisiert oder behoben wird.

Pegelbasierte Alerts werden automatisch behoben, wenn die Bedingung verschwindet. Haftende Alerts, die einen Administrator-Eingriff erfordern (zum Beispiel ein Fehler beim Laden der Konfiguration, Nichtverfügbarkeit des Storage, ein harter Arbeitszyklus-Fehler), werden nicht von selbst behoben: nach Beseitigung der Ursache wird ein solcher Vorfall manuell behoben (bestätigt). Das Beheben wirkt auf dem Node, der den Alert ausgelöst hat: wenn Sie im Netzwerk einen fremden (domänenweit replizierten) Vorfall sehen, müssen Sie ihn auf dem Quell-Node beheben (er ist in der Zeile angegeben).

Domänenweite Replikation von Benachrichtigungen

Bei aktivierter Replikation und festgelegter Domain zeigt jeder Node einen zusammengefassten aktiven Satz für die gesamte Domain: die eigenen Alerts des Nodes plus die Alerts der Peers derselben Domain. Jede Zeile gibt den Quell-Node an, sodass ersichtlich ist, auf welchem Node genau das Problem aufgetreten ist.

- **Domänenweit repliziert:** Stream-Alerts und Transportstrom-Konformitäts-Alerts, Umschaltung auf den Backup-Input, Lade-Fehler von Konfiguration und Transcoder, OTT-Audio-Verwerfung, ACME-Zertifikatsfehler.
- **Bleiben lokal:** Ressourcen-Alerts über Hardware und Storage, „Node verstummt“, Kollision von Node-Namen, Benachrichtigung über den Dienststart, Alerts zur DVR-Aufnahme-Gesundheit und Alerts des Conditional-Access-Moduls. System-Hardware-Alerts (Codes 16–23 und 40–45) können mit dem Schalter für die Replikation von System-Alerts auf Replikation umgestellt werden.

Den Alert „Netzwerk-Node verstummt“ löst jeder Node selbstständig über den Peer aus, der aufgehört hat zu antworten — daher lösen die Peers eines einzelnen ausgefallenen Nodes einen solchen Alert unabhängig voneinander aus.

Externe Alert-Zustellung

Neben der Liste im Browser können Alerts an einen externen Befehl, in einen Telegram-Chat und per E-Mail (SMTP) zugestellt werden. Jeder Kanal ist unabhängig und standardmäßig deaktiviert und führt die Zustellung auf einem eigenen Hintergrund-Thread durch, sodass ein langsamer oder nicht erreichbarer Empfänger die anderen Kanäle und den Dienst selbst nicht aufhält.

Jeder Kanal hat einen Mindest-Schweregrad für die Zustellung (standardmäßig Warnung): jeder auf dem Node sichtbare Alert (eigener oder domänenweit replizierter) ab diesem Pegel wird zugestellt, und zwar sowohl beim Auslösen als auch beim Beheben. Die Verbindungsparameter jedes Kanals sind im Abschnitt *Weboberfläche* beschrieben.

Typische Probleme

- **Alerts der Peers sind nicht sichtbar.** Prüfen Sie, dass auf den Nodes die domänenweite Replikation aktiviert und eine gemeinsame Domain festgelegt ist. System-Hardware-Alerts sind standardmäßig lokal — aktivieren Sie die Replikation von System-Alerts, wenn Sie diese ebenfalls domänenweit sehen möchten.
- **Ein haftender Alert verschwindet nicht, nachdem die Ursache beseitigt wurde.** Beheben Sie ihn manuell auf dem Quell-Node.

1.7 Streamer: Implementierungsdetails

Dieser Abschnitt beschreibt, wie ein Knoten Streams verarbeitet: die SPTS- und MPTS-Pipelines, den Analysator, den Demultiplexer und den Multiplexer, Teststreams, OTT-Auslieferung und DVR, Transcoder, DVB-Empfang, RTMP-Veröffentlichung und -Empfang, die Arbeit mit Dateien und externen Anwendungen sowie EPG. Das allgemeine Übertragungsmodell und die Protokolle sind im Abschnitt *Planung und Datenübertragungsprotokolle* beschrieben, die Steuerung über die Weboberfläche im Abschnitt *Weboberfläche*.

1.7.1 SPTS-Streams

SPTS (Single Program Transport Stream) ist ein MPEG-TS-Stream mit einem einzigen Programm; er ist die grundlegende Verarbeitungseinheit von Perfect Streamer. Der Stream-Typ — SPTS oder MPTS — wird bei der Erstellung gewählt und kann später nicht mehr geändert werden. Für einen SPTS-Stream wird zusätzlich der Inhaltstyp gewählt (Werte wie in der Oberfläche): Default (audio + video), Audio only (Ton ohne Bild, Radiokanäle) oder Video only.

Verwürfelte (verschlüsselte) Streams werden unterstützt: Die Verwürfelung wird automatisch erkannt, und die Prüfungen der Elementarströme werden für diese Zeit ausgesetzt. Die Einstellung „Verschlüsselter Kanal“ erklärt den Stream als dauerhaft verwürfelt und deaktiviert die automatische Erkennung.

Die allgemeinen Anforderungen an den Eingangsstream sind in *Anforderungen an den Eingangsstream* aufgeführt, die Empfangs- und Sendeprotokolle im Abschnitt *Planung und Datenübertragungsprotokolle*. Die Verarbeitung von Multiplexen ist in *MPTS-Streams* beschrieben. Streams werden in der Weboberfläche angelegt und konfiguriert (*Streams*).

Verarbeitungspipeline

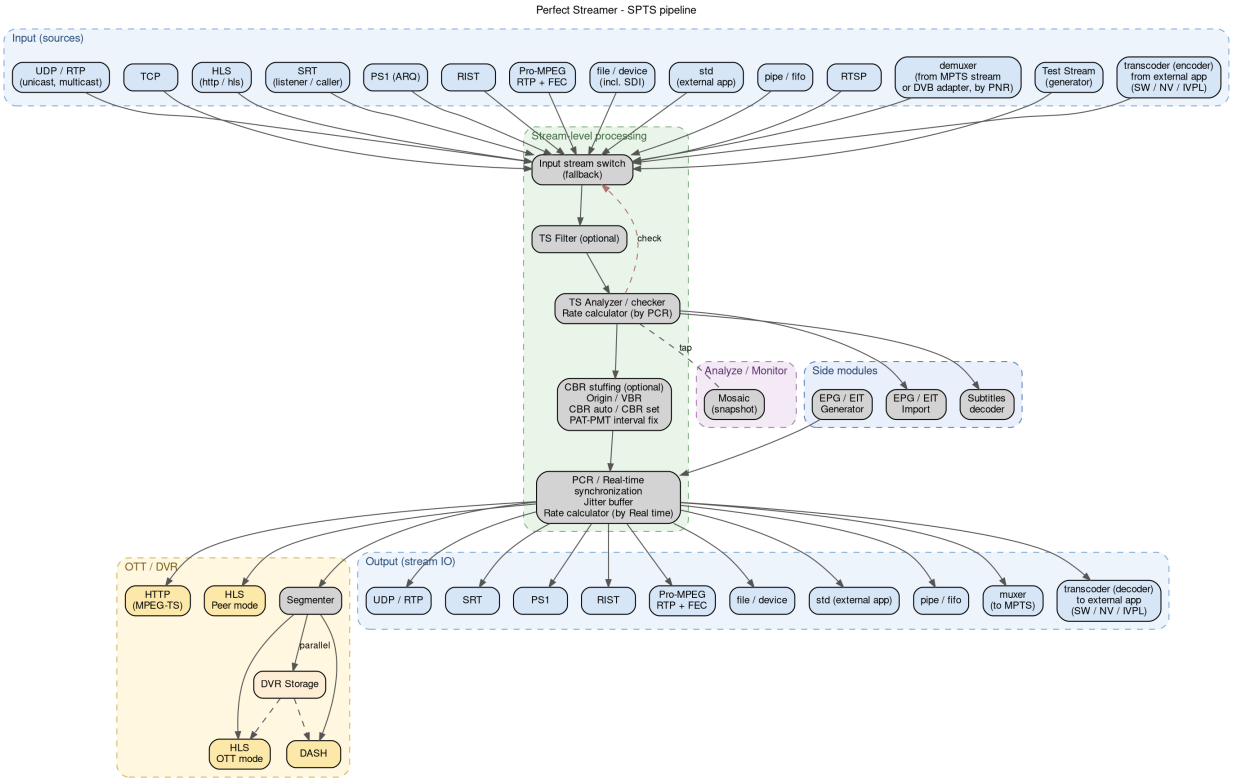


Abb. 2: Pipeline eines SPTS-Streams: Eingänge, Verarbeitung auf Stream-Ebene (Eingangsumschalter, TS-Filter, Analysator, Bitratenangleichung, Synchronisierer), das Subsystem OTT / DVR und die Ausgänge.

Alle Eingänge eines Streams laufen im Eingangsumschalter zusammen, anschließend durchläuft der Stream die aufeinanderfolgenden Verarbeitungsstufen:

1. **Eingangsumschalter** — wählt die aktive Quelle aus und schaltet den Stream bei einer Störung auf die Reserve um (*Quellenredundanz*).

2. **TS-Filter** — optionale Filterung und Modifikation des MPEG-TS (*MPEG-TS-Filterung*, *Modifikation des Streams*).
3. **Analysator** — fortlaufende Gültigkeitsprüfung und Vermessung des Streams (*Analysator*); auf sein Urteil hin erklärt der Eingangsumschalter eine Quelle für gestört. Die Daten des Analysators nutzen außerdem das Mosaik (*Mosaik*, wird über die Einstellung „Mosaik erzeugen“ deaktiviert), der EPG-Import und der Untertiteldecoder.
4. **Bitratenangleichung** — optionales Auffüllen (Stuffing) auf eine konstante Bitrate und Korrektur des PAT/PMT-Intervalls (*Bitratensteuerung*).
5. **Synchronisierer** — Paketausgabe im Takt der PCR mit Kompensation des Netzwerk-Jitters (*Synchronisation*). An derselben Stelle fügt der EIT-Generator die Tabellen des Programmführers ein (*EIT-Generator*).

Vom Ausgang des Synchronisierers wird der Stream an alle Ausgänge verteilt: Transportprotokolle (*Peer-Protokolle für zuverlässige Übertragung*), Aufzeichnung in eine Datei, MPTS-Multiplexer (*Multiplexer*), Transcoder (*Transcoder*) sowie das Subsystem OTT / DVR (*OTT und DVR*). Als Quelle eines SPTS-Streams kann auch ein einzelnes Programm eines Multiplexes dienen — ein Demultiplexer-Eingang (Typ demuxer) extrahiert es anhand der Programmnummer aus einem MPTS-Stream oder von einem DVB-Empfänger (*Demultiplexer*).

Quellenredundanz

Ein Stream kann mehrere Eingänge besitzen, aktiv ist jedoch immer nur einer. Die Reihenfolge der Eingänge in der Liste bestimmt die Priorität: Der erste ist der Haupteingang, die übrigen sind Reserveeingänge. Wird der aktive Eingang als gestört erklärt, schaltet der Stream auf den nächsten Eingang der Liste um; nach dem letzten Eingang wird der Durchlauf beim ersten fortgesetzt. Ein pausierter Eingang wird beim Umschalten übergangen.

Über eine Störung entscheidet der Analysator: Ein Eingang gilt als gestört, wenn der Stream die Gültigkeitsprüfung nicht besteht oder länger als das eingestellte Timeout keine Daten eintreffen. Der Betrieb des Streams auf einem Reserveeingang wird durch einen Alarm festgehalten (*Benachrichtigungen (Alerter)*).

Die Einstellungen für das Umschalten befinden sich im Reiter „Stream“ des Stream-Editors (*Stream konfigurieren*):

„Stream-Timeout (s)“
: Die Zeit ohne gültigen Eingangsstream, nach der auf den nächsten Eingang umgeschaltet wird. Standardmäßig 30 Sekunden.

„Prüfintervall (s)“
: Der Zeitraum, in dem die Quellen erneut geprüft werden. Standardmäßig 60 Sekunden.

„Haupteingang erneut prüfen“
: Während der Stream auf einem Reserveeingang läuft, werden die in der Liste höher stehenden Eingänge im Prüfintervall erneut geprüft. Sobald eine höher priorisierte Quelle wieder gültig ist, kehrt der Stream zu ihr zurück. Standardmäßig aktiviert.

MPEG-TS-Filterung

Standardmäßig wird der Stream unverändert durchgereicht. Die Filterung wird auf zwei Ebenen angewendet: die PID-Regeln jeweils einzeln an jedem Eingang, die Tabellenfilter auf Stream-Ebene.

Die PID-Regeln werden im Reiter „MPEG-TS PID“ des Eingangsfensters konfiguriert (*Editor für Eingang und Ausgang*):

„PIDs annehmen (Whitelist)“
: Die Liste der erlaubten PIDs. Ist die Liste leer, ist alles erlaubt, was nicht durch die Blacklist verboten ist.

„PIDs verwerfen (Blacklist)“
: Die Liste der verbotenen PIDs. Eine PID, die gleichzeitig in beiden Listen steht, wird verworfen.

Nach der Filterung wird die PMT anhand der tatsächlichen Zusammensetzung der Elementarströme neu erzeugt. Die PID-Regeln werden vor der automatischen PID-Anordnung (*Automatische PID-Anordnung*) angewendet, sofern diese aktiviert ist.

Die Filter auf Stream-Ebene befinden sich im Reiter „MPEG-TS“ des Stream-Editors, Abschnitt „Filter“ (standardmäßig alle deaktiviert):

„Nicht benötigte Tabellen bereinigen“
: Entfernt Dienstdaten, die nicht zum Programm gehören, sowie Elementarströme unbekannter Typen; Ströme bekannter Typen (Video, Ton, Teletext, Untertitel) und die Tabellen PAT/PMT bleiben erhalten. Einzelne Tabellen (SDT, EIT, NIT, TDT, CAT) werden über eigene Filter gesteuert.

„CAT / ECM / EMM löschen“
: Entfernt die Tabellen und Nachrichten der Systeme für bedingten Zugriff (CAS) einschließlich der CA-Deskriptoren aus der PMT — zum Beispiel, um die Rückstände des CAS nach der Entwürfelung zu beseitigen.

„EIT löschen“
: Entfernt die Programmführer-Tabellen der Quelle. Standardmäßig wird die EIT unverändert durchgereicht.

„NIT löschen“, „TDT löschen“
: Entfernen die Netzwerk- und Zeitstempeltabellen der Quelle.

„Teletext löschen“, „Untertitel löschen“
: Entfernen die Teletext- und DVB-Untertitelströme zusammen mit ihren Einträgen in der PMT.

„Ursprüngliche SDT löschen“ (Abschnitt „SDT“)
: Entfernt die Service-Description-Tabelle der Quelle — zum Beispiel bei der Weitergabe des Streams zur weiteren Multiplexierung. Bei der Vergabe eines eigenen Servicenamens (*Modifikation des Streams*) wird die ursprüngliche SDT automatisch ersetzt; dieser Filter muss dann nicht zusätzlich aktiviert werden.

Bemerkung: Das Entfernen der obligatorischen Tabellen (SDT, NIT, TDT, EIT, CAT) verletzt die Anforderungen von TR 101 290 an deren Vorhandensein im Stream — dies ist eine bewusste Entscheidung des Betreibers, die zum Beispiel bei der Weitergabe des Streams zur weiteren Multiplexierung angebracht ist. Darüber hinaus verändert jede Filterung die Paketzusammensetzung des Streams; zur Auswirkung auf die PCR-Genauigkeit siehe *Bitratensteuerung*.

Modifikation des Streams

Eigene Servicedaten werden im Abschnitt „SDT“ des Reiters „MPEG-TS“ festgelegt. Die Erzeugung der Tabelle aktiviert das Feld „Servicename“ (ebenso die aktive Erzeugung der EIT aus EPG): Enthält die Quelle keine SDT, wird die Tabelle erzeugt; ist eine vorhanden, wird sie durch die erzeugte ersetzt. Die Felder „Providername“ und „Sprache“ füllen die entsprechenden Felder der erzeugten Tabelle. Die Einstellung „Netzwerk-ID“ legt den Netzwerkidentifikator für die erzeugten EIT-Tabellen fest, wenn die Quelle keinen eigenen Netzwerkidentifikator meldet; ist in der Quelle ein Identifikator vorhanden, wird dieser verwendet.

Die Programmnummer wird im Abschnitt „Programmnummer“ neu vergeben: Aktivieren Sie „Programmnummer ändern“ und legen Sie „Neue Programmnummer“ fest. Die Tabellen PAT und PMT werden mit der neuen Nummer neu erzeugt; SDT und EIT werden ebenfalls neu erzeugt.

„PAT/PMT-Intervall fixieren“ — erzeugt PAT/PMT neu und fügt sie in einem Intervall gemäß TR 101 290 ein. Anzuwenden bei Quellen mit seltenen oder unregelmäßigen Tabellen.

Die Neuzuordnung von PIDs und Sprachen erfolgt am Eingang, im Reiter „MPEG-TS PID“ des Eingangsfensters:

„PIDs neu zuordnen“
: Paare „Von PID“ / „Zu PID“ — Ersetzen der Identifikatoren der Elementarströme, zum Beispiel um den Stream an den PID-Plan des Netzwerks anzupassen. Bei aktivierter automatischer PID-Anordnung (*Automatische PID-Anordnung*) bleiben diese Paare wirkungslos.

„Audiosprache neu zuordnen“
: Paare „Audio-PID“ / „Neue Sprache“ — Ersetzen der Sprache der Tonspur in der PMT.

Die Erzeugung der EIT aus einer externen EPG-Quelle wird im Abschnitt „EIT aus EPG erzeugen“ des Reiters „Stream“ aktiviert: Es werden „EPG-Quelle“ und „EPG-Kanal“ ausgewählt, und in den Stream werden EIT-Tabellen mit dem Programmführer eingefügt (*EIT-Generator*). Die umgekehrte Operation ist die Einstellung „EIT in die EPG-Datenbank extrahieren“, die den Programmführer aus dem Stream in die EPG-Datenbank des Knotens exportiert (*EPG/XMLTV-Import*).

Für Eingänge, die den ursprünglichen Transportstrom empfangen, steht eine softwareseitige BISS-Entwürfelung zur Verfügung: Am Eingang wird ein BISS-1- oder BISS-E-Schlüssel hinterlegt, und der Stream wird vor der Analyse und der weiteren Verarbeitung entschlüsselt. Der Empfang verwürfelter Services von einem DVB-Empfänger (CAM, BISS auf Adapterebene) ist in *DVB-Empfänger* beschrieben.

Automatische PID-Anordnung

Standardmäßig werden die Identifikatoren der Elementarströme unverändert von der Quelle in den Ausgangsstream übernommen; beim Umschalten auf einen Reserveeingang ändert sich die PID-Anordnung am Ausgang daher zusammen mit der Quelle. Die automatische Anordnung vergibt die PIDs selbst, nach ein und derselben Regel für jeden Eingang: Der PID-Plan des Ausgangsstreams bleibt unverändert, unabhängig davon, welche Quelle gerade aktiv ist.

Der Modus steht nur für SPTS zur Verfügung und wird im Abschnitt „MPEG-TS“ des Reiters „MPEG-TS“ des Stream-Editors aktiviert:

„Automatische PID-Anordnung“
: Aktiviert den Modus.

„Basis-PID“
: Der erste PID des Fensters: von 32 bis 8127, Standardwert 32. Die untere Grenze legt das Fenster außerhalb des für die PSI/SI-Tabellen reservierten Bereichs, die obere hält das gesamte Fenster unterhalb des PIDs der Nullpakete.

Das Programm wird in ein Fenster aus 64 PIDs umnummeriert, das beim Basis-PID beginnt:

PID	Zweck
Basis	PMT
Basis + 1	Video
danach	Tonspuren
danach	die übrigen Elementarströme: Teletext, DVB-Untertitel, Daten, SCTE-35
danach	PCR, sofern er nicht in einem der aufgeführten Ströme übertragen wird
danach	ECM: zuerst nach den Deskriptoren auf Programmebene, danach nach den Deskriptoren der einzelnen Ströme
vom Ende des Fensters abwärts	EMM

Innerhalb jeder Gruppe bleibt die Reihenfolge der ursprünglichen PMT erhalten. Enthält das Programm kein Video, wird die Anordnung verdichtet: Unmittelbar nach der PMT folgt die erste Tonspur. Die EMM werden vom Ende des Fensters aus vergeben, da die CAT unabhängig von der PMT ausgewertet wird und früher eintreffen kann — ihre Adressen dürfen nicht von der Anzahl der Elementarströme abhängen.

Die Tabellen PAT, PMT und CAT werden für die neue Anordnung neu aufgebaut. SDT, NIT, EIT und TDT bleiben unverändert: Sie enthalten keine PID-Verweise.

Was beim Aktivieren des Modus zu beachten ist:

- Der Modus erzwingt die MPEG-TS-Filterung, sodass PIDs, die nicht in PMT und CAT deklariert sind, nicht mehr nach außen gelangen. Ein Stream, der auf den Durchlauf undeklarierter PIDs angewiesen war, verliert diese.
- Die manuelle PID-Neuzuordnung an den Eingängen ist wirkungslos und wird ignoriert; darüber wird eine Warnung ins Protokoll geschrieben. Die Weboberfläche blendet diese Felder aus, solange der Modus aktiv ist; die gespeicherten Paare gehen nicht verloren und kehren beim Deaktivieren zurück.
- Die Annahme- und Ablehnungslisten für PIDs bleiben wirksam und werden vor der Umnummerierung angewendet: Einen Slot erhalten nur die verbliebenen Elementarströme, und im Fenster bleiben keine Lücken (*MPEG-TS-Filterung*).
- Auf den Transkoder-Eingang (*Transcoder*) und den Eingang des Testgenerators (*Teststreams*) wirkt der Modus nicht: Dort erstellt der Encoder bzw. der Generator den PID-Plan, weshalb deren eigene PMT-PID-Felder verfügbar bleiben.
- Passt das Programm nicht in das Fenster, findet überhaupt keine Umnummerierung statt — die Anordnung bleibt die ursprüngliche und eine Warnung geht ins Protokoll; ein halb umnummerierter Stream wird nicht ausgegeben.
- Eine Änderung an einer der beiden Einstellungen wird im laufenden Betrieb übernommen: Der Stream selbst wird nicht neu gestartet, seine Ein- und Ausgänge werden jedoch neu aufgebaut — einige Sekunden für die Neusynchronisation.

In der Stream-Statistik zeigt der Block „Medieninformation — Quelle“ die am Eingang eingetroffenen PIDs, „Medieninformation — Ergebnis“ und die Bitratentabelle nach PID dagegen die vergebenen; bei aktiviertem Modus ist eine Abweichung zwischen beiden normal.

Die automatische PID-Anordnung hat nichts mit der PID-Vergabe im MPTS-Multiplexer (*PID-Zuweisung*) zu tun — das ist eine eigene Einstellung des muxer-Eingangs.

Bitratensteuerung

Der Modus der Bitratensteuerung wird über das Feld „Bitratenmodus“ im Reiter „MPEG-TS“ festgelegt (Werte wie in der Oberfläche):

Original (Standard)
: Der Stream wird unverändert durchgereicht; die PCR-Marken der Quelle werden nicht neu berechnet.

VBR
: Entfernt die NULL-Pakete der Quelle, wodurch die Bitrate minimiert wird. Geeignet, wenn der Stream ausschließlich für die OTT-Ausstrahlung oder den Transit verwendet wird. Die PCR-Marken werden dabei nicht neu berechnet, daher ist die PCR-Genauigkeit nach TR 101 290 am Ausgang nicht gewährleistet.

CBR (auto stuffing)
: Gleicht die Bitrate durch Einfügen von NULL-Paketen (stuffing) auf einen konstanten Wert an. Die Zielbitrate wird automatisch ermittelt: Sie wird einige Sekunden nach dem Einschalten festgelegt, bei anhaltender Überschreitung angehoben und schrittweise gesenkt, wenn die Spitze des Inhalts lange unterhalb des Ziels bleibt. Ein Wechsel des Ziels unterbricht die Ausgabe des Streams nicht.

CBR (explicit stuffing)
: Ebenso, jedoch wird die Zielbitrate explizit über das Feld „Stuffing-Bitrate (kbps)“ festgelegt. Legen Sie den Wert mit einer Reserve oberhalb der tatsächlichen Bitrate des Inhalts fest.

In den CBR-Modi werden die PCR-Marken anhand der tatsächlich übertragenen Bytes neu berechnet, daher erfüllt der Ausgang die Anforderungen von TR 101 290 an die PCR-Genauigkeit. Zusätzlich wird bei aktivem Stuffing am Ausgang die Wiederholrate der PCR auch bei langen Videoframes innerhalb der Norm gehalten. Der Analysator misst in diesem Fall genau den ausgegebenen, angeglichenen Stream (*Analysator*).

Wann CBR erforderlich ist. Jede Operation, die Pakete einfügt oder entfernt — die Tabellen- und PID-Filter, die Erzeugung von SDT und EIT, die Korrektur des PAT/PMT-Intervalls, der Wechsel der Programmnummer sowie der VBR-Modus selbst — verschiebt die Lage des Inhalts gegenüber seinen PCR-Marken. Wenn der Empfänger des Streams die Konformität mit TR 101 290 verlangt, zum Beispiel bei der DVB-Ausstrahlung, aktivieren Sie einen CBR-Modus: Das Stuffing maskiert diese Änderungen durch die Neuberechnung der PCR. Für einen Stream hinter dem Transcoder aktivieren Sie den CBR-Modus am empfangenden Stream (*PCR und Bitrate des Ausgangsstreams*).

Der aktuelle Zustand des Stuffings — die Zielbitrate und die Einfügerate der NULL-Pakete — wird in den Daten des aktiven Eingangs auf der Stream-Seite angezeigt (*Streams*, Abschnitt „Stuffing“; der Abschnitt ist nur sichtbar, solange das Stuffing aktiv ist).

Synchronisation

Der Synchronisierer gibt die Pakete im Takt der PCR-Marken der Quelle an die Ausgänge aus — gültige PCR sind daher für den Eingangsstream zwingend erforderlich. Zwischen Empfang und Ausgabe wird der Stream in einem Jitter-Puffer verzögert, der die Ungleichmäßigkeit des Paketeintreffens aus dem Netz ausgleicht.

Der Puffer wird für jeden Eingang separat konfiguriert (Eingangsfenster, *Editor für Eingang und Ausgang*; erweiterte Einstellungen, Bezeichnungen wie in der Oberfläche):

Auto jitter buffer
: Automatische Wahl der Verzögerung anhand des Eingangstyps: Für TCP- und HLS-Eingänge wird ein vergrößerter Puffer verwendet, für die übrigen der Standardpuffer. Standardmäßig aktiviert.

Jitter buffer (ms)
: Die manuell eingestellte Puffergröße, wenn der automatische Modus deaktiviert ist (standardmäßig 500 ms). Erhöhen Sie den Wert, wenn der Puffer in einem instabilen Netz regelmäßig leerläuft.

PCR discontinuity window (ms)
: Das Fenster des zulässigen PCR-Sprungs (standardmäßig 1000 ms). Ein Sprung über das Fenster hinaus gilt als Unterbrechung des Streams und löst eine harte Neusynchronisation aus.

Eine langsame Abweichung zwischen dem Takt der PCR der Quelle und der Uhr des Knotens beseitigt der Drift-Kompensator: Die Korrektur wird gleitend in Mikroverschiebungen angewendet, ohne harte Neusynchronisation und ohne Unterbrechung der Ausgabe. Die Steuerung befindet sich im Reiter „Analyse“ des Stream-Editors: „Kompensation der Synchronisationsdrift“ (standardmäßig aktiviert) und „Weiches Fenster der Synchronisationsdrift (ms)“ — die Totzone, innerhalb derer keine Korrektur angewendet wird (standardmäßig 500 ms). Die harte Neusynchronisation bleibt der Rückfallmechanismus für echte Unterbrechungen des Streams.

1.7.2 MPTS-Streams

MPTS (Multi Program Transport Stream) — ein MPEG-TS-Stream mit mehreren Programmen (Diensten); jedes Programm besitzt eine eindeutige Nummer (PNR). Haupteinsatzgebiete sind die DVB-Ausstrahlung und der Transit fertiger Multiplexe. Der Streamtyp — SPTS oder MPTS — wird beim Anlegen gewählt und kann später nicht mehr geändert werden (*Neuer Stream*).

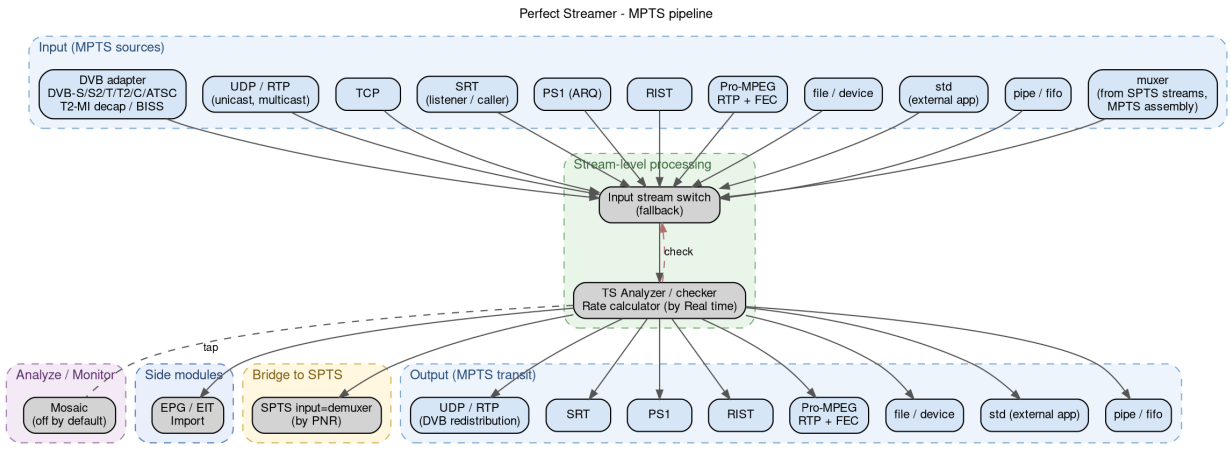


Abb. 3: Pipeline eines MPTS-Streams: Transit des Multiplexes ohne Änderung seiner Zusammensetzung; einzelne Programme werden über den Demultiplexer in SPTS-Streams verarbeitet.

Transit des Multiplexes

Ein MPTS-Stream ist ein Transitstream: Der Multiplex wird unverändert weitergegeben. Filterung und Modifikation des MPEG-TS (*MPEG-TS-Filterung*, *Modifikation des Streams*), die Bitratenmodi (*Bitratensteuerung*), die OTT-Auslieferung und die DVR-Aufzeichnung (*OTT und DVR*) gelten für einen MPTS-Stream nicht. Die konstante Bitrate und die Zusammensetzung eines eigenen Multiplexes werden bei dessen Aufbau durch den Multiplexer festgelegt (*Multiplexer*); ein einzelnes Programm wird verarbeitet, nachdem es in einen SPTS-Stream ausgekoppelt wurde (*Demultiplexer*).

Quellen eines MPTS-Streams sind die Netzwerktransporte, die einen fertigen Multiplex empfangen (UDP / RTP, SRT, PS1, RIST, Pro-MPEG, TCP, Datei und weitere — Abschnitt *Planung und Datenübertragungsprotokolle*), der DVB-Empfänger (*DVB-Empfänger*) und der Multiplexer (*Multiplexer*). Einprogramm-Quellen — HLS, RTSP, RTMP, Demultiplexer, Transcoder, Testgenerator — stehen für MPTS nicht zur Verfügung. Die Ausgänge sind Transportausgänge, einschließlich der Ausspielung des Multiplexes in ein DVB-Verteilnetz; OTT-Ausgänge besitzt ein MPTS-Stream nicht.

Die Quellenredundanz arbeitet ebenso wie bei SPTS-Streams: mehrere Eingänge nach Priorität, Stream-Timeout, Prüfintervall und Rückkehr auf den Haupteingang (*Quellenredundanz*).

Analyse nach Programmen

Der Analysator eines MPTS-Streams erfasst jedes Programm einzeln sowie den Multiplex in der Summe: Bitrate, Zustand und Kontinuitätsfehler, Verwürfelung, Medieninformationen zu jedem Programm sowie die Kennungen des Multiplexes (TS ID und Netzwerkkennung). Die Summenzähler der Programme — insgesamt, unverwürfelt, verwürfelt und fehlerhaft — werden im Statistikfenster des Streams ausgegeben (*Stream-Statistik*); in der Streamliste sind MPTS-Streams mit dem Symbol MPTS gekennzeichnet und in einer eigenen Gruppe zusammengefasst. Ein verwürfeltes Programm gilt als gültig und geht nicht in den Fehlerzähler ein.

Der TR 101 290-Monitor (Reiter „Analyse“, Einstellung „TR 101 290-Monitor“) fällt für MPTS ein einziges Urteil über den gesamten Multiplex: Transportfehler, Kontinuität über alle PIDs, PCR-Wiederholung und PCR-Sprünge über alle Programme, Vorhandensein der Tabellen PAT/PMT und SI (NIT, SDT, EIT, TDT). Das Fenster des zulässigen PCR-Sprungs wird über die Eingangseinstellung „PCR discontinuity window (ms)“ festgelegt — wie bei SPTS-Streams (*Synchronisation*). Die vertieften SPTS-Metriken — PCR-Genauigkeit, Analyse des Videopuffers T-STD, PTS-Intervalle — gelten nicht für den Multiplex als Ganzes: Koppeln Sie ein Programm zur detaillierten Analyse in einen eigenen SPTS-Stream aus (*Demultiplexer*, *Analysator*).

Das Mosaik ist für einen MPTS-Stream standardmäßig abgeschaltet (Einstellung „Mosaik erzeugen“); bei Aktivierung wird für jedes Programm des Multiplexes ein Miniaturbild erzeugt. Das bedeutet eine spürbare zusätzliche Last — auf ausgelasteten Servern wird das Einschalten des Multiplex-Mosaiks nicht empfohlen.

BISS-Entschlüsselung

An den Netzwerk- und Dateieingängen eines MPTS-Streams (UDP / RTP, SRT, RIST, Pro-MPEG, TCP, PS1, Datei und ähnlichen) steht die softwareseitige BISS-Entschlüsselung je Programm zur Verfügung: Festgelegt werden Paare aus „Programmnummer — Schlüssel“ (BISS-1 oder BISS-E), und jedes angegebene Programm wird vor der Analyse und der weiteren Übertragung entschlüsselt. Die Entschlüsselung auf Ebene des DVB-Adapters — CAM und BISS, einschließlich der Schlüssel für T2-MI — ist in *DVB-Empfänger* beschrieben.

EPG des Multiplexes

Der im Multiplex enthaltene Programmführer wird mit der Einstellung „EIT in die EPG-Datenbank extrahieren“ in die EPG-Datenbank des Knotens übernommen (*EPG/XMLTV-Import*). Auf einem MPTS-Transitstream findet keine EIT-Erzeugung statt: Die EIT-Tabellen der Programme überträgt der Multiplexer beim Aufbau aus den Quellstreams (*Multiplexer*, *EIT-Generator*).

1.7.3 Analysator

Der integrierte Analysator prüft und vermisst fortlaufend den aktiven Eingang jedes Streams. Die Ergebnisse werden von mehreren Teilsystemen des Knotens genutzt: Das Gültigkeitsurteil steuert die Eingangsumschaltung (*Quellenredundanz*), Metriken und Diagramme werden im Statistikfenster des Streams ausgegeben (*Stream-Statistik*), die zusammenfassenden Badges in der Stream-Liste, und Verstöße lösen Alarme aus (*Benachrichtigungen (Alerter)*). Die Daten des Analysator-Decoders nutzen das Mosaik, der EPG-Import und der Untertiteldecoder (*Verarbeitungspipeline*).

Die grundlegenden Prüfungen und Messungen sind stets aktiviert. Die vertieften Analysen werden über eigene Einstellungen im Reiter „Analyse“ des Stream-Editors aktiviert — jede von ihnen erzeugt zusätzliche

CPU-Last und wird nur bei Bedarf eingeschaltet. Die kumulativen Zähler werden mit der Schaltfläche „Statistik zurücksetzen“ im Statistikfenster gelöscht.

Gültigkeitsprüfung

Der Eingangsstream wird stufenweise geprüft: Einrasten der TS-Synchronisation, Empfang der Tabellen PAT und PMT, Vorhandensein von PCR-Marken, Auswertung der Audio- und Videospuren bis hin zur Dekodierung eines Keyframes. Das Ergebnis wird im Statistikfenster im Abschnitt „MPEG-TS-Prüfer“ angezeigt — die einzelnen Attribute TS, PAT, PMT, PCR, Audio, Audio ES, Video, Video ES. Die Prüfung wird im Stream-Prüfintervall wiederholt; ein Eingang gilt als gestört, wenn der Stream die Prüfung nicht bestanden hat oder länger als das Stream-Timeout keine Daten eintreffen — dann erfolgt die Umschaltung auf den Reserve-Eingang (*Quellenredundanz*).

Ein verwürfelter Transport ist ein gültiger Zustand (*SPTS-Streams*): Die Transport- und die PSI-Ebene werden weiterhin geprüft, und sobald offene Daten auftreten, wird die vollständige Analyse automatisch wieder aufgenommen.

Fortlaufende Messungen

Ohne zusätzliche Einstellungen vermisst der Analysator:

- **Bitrate** — die Rate des Eingangsstreams (bei SPTS anhand der PCR-Marken, bei MPTS gegen die Echtzeit) sowie die Ausgangsrate nach dem Synchronisierer; Diagramm „Bitrate“ im Statistikfenster.
- **Bitrate je PID** — die aktuelle Rate und die Anzahl der Pakete pro Sekunde für jede PID des aktiven Eingangs; die Tabelle im Block „Analysator“ des Statistikfensters. Der PID-Typ stammt aus der Auswertung der Streamtabellen (PAT, PMT, `video`, `aac` und ähnliche); die PID der NULL-Pakete (8191) ist mit `NULL` gekennzeichnet, ein Stream, dessen Typ sich nicht bestimmen ließ, dagegen mit einem Strich. Die Messung erfolgt nur bei SPTS: ein MPTS-Stream verfügt bei keiner Multiplexquelle über diese Tabelle, und die Zusammensetzung sowie die Raten der Programme zeigt die Multiplextabelle (*Analyse nach Programmen*).
- **Kontinuität und Transport** — Continuity-Counter-Fehler CC (fensterbezogen und kumulativ), Transportfehler (TEI), Zähler verwürfelter Pakete; Diagramm „Kontinuität & Verschlüsselung“.
- **PCR-Intervall** — mittleres und maximales Intervall zwischen den PCR-Marken. Bei aktiviertem TR 101 290-Monitor gilt der strenge Grenzwert von 40 ms, ohne ihn der Grenzwert von ISO/IEC 13818-1 von 100 ms.
- **PCR-Jitter** — der Rückstand des Ausgabe-Schedulers hinter dem PCR-Takt; Diagramm „PCR-Jitter“.
- **PCR-Drift** — das systematische Weglaufen der Taktfrequenz der Quelle gegenüber der Echtzeit, in ppm. Im Unterschied zum Jitter kennzeichnet die Drift die Stabilität des Encoder-Quarzes und baut sich über Minuten auf; ein belastbares Urteil bildet sich über einem Fenster ab 300 Sekunden. Die Toleranz nach ISO/IEC 13818-1 beträgt ±30 ppm; eine anhaltende Überschreitung wird im Statistikfenster hervorgehoben („PCR-Drift“), getrennt vom TR 101 290-Urteil: Eine Quarzdrift ist ein Problem des Encoders auf der Quellenseite und keines der Transportintegrität.
- **PCR-Genauigkeit** — der Fehler beim Einfügen der PCR-Marken gegenüber dem tatsächlichen Sendezeitpunkt; die Toleranz nach TR 101 290 beträgt ±500 ns. Die Kennzahl ist nur für Streams mit konstanter Bitrate sinnvoll (siehe Modus-Detektor weiter unten).

Multiplex-Modus-Detektor. Der Analysator klassifiziert die Quelle anhand der Regelmäßigkeit der PCR gegenüber der Paketposition automatisch als CBR oder VBR; das Urteil wird mit Hysterese festgehalten, um häufige Umschaltungen auszuschließen. Ein belastbares Urteil wird durch ein Badge CBR oder VBR in der Stream-Liste angezeigt; solange die Datenlage nicht ausreicht, erscheint kein Badge. VBR ist kein Defekt: Es

ist ein normaler Modus, in dem die nur für CBR sinnvollen Prüfungen (PCR-Genauigkeit, T-STD-Videopuffer) automatisch nicht bewertet werden.

Vertiefte Analysen

Drei voneinander unabhängige Analysen werden im Reiter „Analyse“ des Stream-Editors aktiviert (alle sind standardmäßig deaktiviert):

„Tiefenanalyse“
: Versetzt den Analysator in den fortlaufenden Modus der Stream-Auswertung (spürbare CPU-Last) und ergänzt:

  - die Intervalle der Tabellen PAT und PMT (Empfehlung — höchstens 500 ms);
  - die Intervalle der Keyframes (GOP) und der IDR-Frames. Für Player, die zu einem beliebigen Zeitpunkt einsteigen, wird ein GOP von höchstens 1 Sekunde empfohlen. Stimmt das IDR-Intervall ungefähr mit dem Keyframe-Intervall überein, hat der Stream ein closed-GOP — jedes GOP beginnt mit einem vollständigen Einstiegspunkt; fehlt die IDR-Kennzahl, hat der Stream ein open-GOP;
  - das Intervall von PAT bis zum Keyframe — von ihm hängt ab, wie schnell die Wiedergabe bei einem Einstieg zu einem beliebigen Zeitpunkt startet;
  - den Video-Codec-Steckbrief (Stream-Typ, Profil und Level, Auflösung, Scan-Typ, VBV-Parameter) und die Parameter der Audiospuren;
  - Daten zur Bereitschaft für die Werbeeinblendung: GOP-Struktur, Random-Access-Marken (RAI), SCTE-35-Ereignisse und Splice-Punkte. Die Schwelle der Vorabbenachrichtigung über einen Splice-Punkt wird über die Einstellung „Splice-Punkt-Benachrichtigung bei (Frames)“ festgelegt (eine erweiterte Einstellung);
  - die Überwachung des Vorhandenseins der SI-Tabellen (SDT, EIT, NIT, TDT) und Fehler in der Struktur der Elementarströme.

„PCR/PTS-Lücke analysieren“
: Die Streuung zwischen der PCR und den PTS/DTS-Marken je Elementarstrom. Eine zu große Streuung verursacht Probleme bei Playern mit kleinem Synchronisationspuffer. Zusätzlich wird das Wiederholintervall der Video-PTS-Marken gemessen (Indikator 2.5 des TR 101 290-Berichts, Grenzwert 700 ms).

„T-STD-Videopuffer analysieren“
: Das Modell des Referenzdecoder-Puffers T-STD (ISO/IEC 13818-1): die Prüfung, dass der Videopuffer weder überläuft noch leerläuft. Unterstützt werden MPEG-2, H.264/AVC und HEVC; die Abflussrate passt sich der tatsächlichen Videobitrate an. Bewertet wird nur bei Streams mit konstanter Bitrate.

Die Ergebnisse werden im Statistikfenster in den Abschnitten „Tiefenanalyse“ und „Analysator“ ausgegeben. Die Einstellungen „Trace“ und „MPEG-TS-Trace“ aktivieren die ausführliche Protokollierung der Stream-Prüfung — nur zur Diagnose; lassen Sie sie nicht dauerhaft eingeschaltet.

TR 101 290-Monitor

Die Einstellung „TR 101 290-Monitor“ (Reiter „Analyse“) aktiviert die fortlaufende Überwachung der Konformität des Streams mit dem Standard ETSI TR 101 290. Der Monitor ist leichtgewichtig — die aufwendige Auswertung der Elementarströme schaltet er nicht ein — und arbeitet unabhängig von den vertieften Analysen.

Das Ergebnis ist ein einziges Urteil für den Stream: „Gut“, „Schlecht“ oder „Prüfung läuft…“ für die Dauer der Datensammlung. Das Urteil wird durch ein farbiges Badge TR-290 in der Stream-Liste und im Statistikfenster angezeigt; ein Klick öffnet den Bericht „TR 101 290“ — eine Tabelle der Indikatoren, gruppiert nach den Prioritäten des Standards (Priorität 1 — für die Dekodierung erforderlich, Priorität 2 — für die Überwachung empfohlen, Priorität 3 — anwendungsabhängig), mit den gemessenen Werten und den Grenzwerten.

Besonderheiten der Bewertung:

- Ein Verstoß geht nur dann in das Urteil ein, wenn er anhaltend ist (in mindestens 30 der letzten 60 Sekunden beobachtet) — kurze Ausschläge lassen die Anzeige nicht aufleuchten. Gesondert verfolgt wird die Abnahmestufe: Ein Verstoß, der ununterbrochen 300 Sekunden andauert, ist das Kriterium für die technische Abnahme eines Kanals.
- Ein Teil der Indikatoren erfordert zusätzliche Analysen. Zeilen, die von den Einstellungen „PCR/PTS-Lücke analysieren“ und „T-STD-Videopuffer analysieren“ abhängen, werden bei deaktivierter Einstellung im Bericht als deaktiviert gekennzeichnet und beeinflussen das Urteil nicht. Die numerischen PAT/PMT-Intervalle und die Überwachung des Vorhandenseins der SI-Tabellen werden nur bei aktivierter „Tiefenanalyse“ gemessen — ohne sie bleiben die entsprechenden Zeilen ohne gemessene Werte.
- Die PCR-Genauigkeit und der T-STD-Videopuffer werden erst bewertet, nachdem der Modus-Detektor eine konstante Bitrate der Quelle bestätigt hat — und sie werden bis zum Zurücksetzen der Statistik oder bis zu einer Änderung der Einstellungen der Bitratenangleichung weiter bewertet, auch wenn sich das Urteil des Detektors später ändert. Bei Streams, die nie als CBR bestätigt wurden, werden diese Zeilen nicht bewertet.
- Bei aktiver Bitratenangleichung vermisst der Monitor genau den ausgegebenen, angeglichenen Stream (*Bitratensteuerung*).
- Für einen MPTS-Stream wird ein einziges Urteil für den gesamten Multiplex gefällt (*Analyse nach Programmen*).

Anhaltende Verstöße lösen zusätzlich Alarme mit Analysator-Codes aus — die vollständige Liste ist in *Katalog der Alert-Codes* aufgeführt. Die Zähler des Monitors und die kumulierte Statistik werden mit der Schaltfläche „Statistik zurücksetzen“ zurückgesetzt.

Reklamationsassistent

Weist ein Kanal anhaltende TR 101 290-Verstöße oder eine PCR-Drift auf, kann der Bediener einen fertigen Aufgabentext für einen KI-Chat abrufen, aus dem dieser ein förmliches Reklamationsschreiben an den Stream-Anbieter verfasst. Die Anfrage erfolgt über die HTTP-API des Knotens:

```
GET /data/stream/<id>/ai-complaint-prompt
```

Die Antwort ist ein Text mit der Auflistung der aktiven Verstöße, den gemessenen Werten, den Grenzwerten des Standards und der Auswirkung jedes Fehlers auf den Decoder des Empfängers. Der Name des Streams, seine Kennungen und die Adressen der Quelle werden bewusst nicht in den Text aufgenommen — an ihrer Stelle steht ein Platzhalter, den der Bediener selbst ausfüllt, damit keine Betriebsdaten an einen fremden KI-Dienst übermittelt werden. Die Funktion steht nur für SPTS-Streams zur Verfügung.

1.7.4 Demultiplexer

Ein Demuxer-Eingang (Typ demuxer) extrahiert ein einzelnes Programm aus einem Multiplex und wandelt es in einen gewöhnlichen SPTS-Stream um. Nach der Extraktion steht dem Programm die gesamte Verarbeitung der SPTS-Pipeline zur Verfügung (*SPTS-Streams*): Filterung und Modifikation, Transkodierung, OTT-Auslieferung und DVR-Aufzeichnung. Dies ist der einzige Weg, einen empfangenen Multiplex programmweise zu verarbeiten (*MPTS-Streams*).

Um ein Programm anzubinden, wird im SPTS-Stream ein Eingang des Typs demuxer hinzugefügt und Folgendes ausgewählt (die Feldbezeichnungen entsprechen denen der Oberfläche):

- **Quelle** (Felder Source kind und Source id) — ein MPTS-Stream desselben Knotens oder ein DVB-Adapter (*DVB-Empfänger*).
- **Programm** (Feld PNR) — die Programmnummer. Bei einer MPTS-Stream-Quelle, die aktiv und bereits analysiert ist, wird die Nummer aus der Liste der erkannten Dienste mit ihren Namen ausgewählt; bei einem DVB-Adapter und bei einer noch nicht analysierten Quelle wird die PNR manuell eingegeben.

Besonderheiten:

- Der Demultiplexer erhält einen bereits descrambelten Multiplex: die BISS-Schlüssel werden am Eingang des MPTS-Streams festgelegt (*BISS-Entschlüsselung*), das CAM- und BISS-Descrambling auf Adapterebene dagegen in den Einstellungen des DVB-Adapters (*DVB-Empfänger*).
- Die Quelle ist gegen Löschen geschützt: solange mindestens ein Demultiplexer auf einen MPTS-Stream oder einen DVB-Adapter verweist, kann dieser nicht gelöscht werden — der Knoten meldet, welche Objekte darauf verweisen.
- Für innere T2-MI-Multiplexe wird eine zusammengesetzte Programmnummer verwendet, die die Nummer des Trägers und die PNR innerhalb dieses Trägers kombiniert: Trägernummer × 65536 + PNR (zum Beispiel Träger 1, PNR 7 → 65543; Träger 0 ist der äußere Multiplex, gewöhnliche PNR). Der zusammengesetzte Wert wird in das Feld PNR eingetragen. Der Empfang von T2-MI ist in *DVB-Empfänger* beschrieben.

Derselbe Multiplex kann gleichzeitig vollständig (als MPTS-Stream mit Transportausgängen) und in Teilen — als einzelne SPTS-Streams über Demultiplexer — ausgeliefert werden.

1.7.5 Multiplexer

Der Multiplexer setzt einen MPTS-Stream aus den einzelnen SPTS-Streams des Knotens zusammen und erzeugt einen vollwertigen DVB-Multiplex: Er generiert die PSI/SI-Tabellen, führt einen Ausgabefahrplan nach dem T-STD-Modell und gleicht die Bitrate durch Einfügen von NULL-Paketen aus. Bei vorgegebener Zielbitrate entspricht der Ausgang TR 101 290 und eignet sich zur Speisung von Modulatoren und von DVB-Verteilnetzen.

Aufbau des Multiplexes

1. Einen Stream vom Typ MPTS erstellen (*Neuer Stream*).
2. Ihm einen Eingang vom Typ muxer hinzufügen — er ist der einzige und enthält die Einstellungen des gesamten Multiplexes (siehe unten).
3. Die Programme anbinden — auf eine der beiden folgenden Weisen:
    - über den Programmeditor des Multiplexes: Schaltfläche „Programme bearbeiten…“ am muxer-Eingang, anschließend „Programm hinzufügen“ und Auswahl des Quell-Streams;

- von der Quellenseite: dem SPTS-Stream einen Ausgang vom Typ muxer hinzufügen und den Ziel-MPTS-Stream angeben.

Eine neue Anbindung wird pausiert angelegt — das Programm erscheint mit einem Pausensymbol in der Liste; nach Prüfung der Einstellungen wird die Pause aufgehoben.

4. Für alle Programme des Multiplexes wiederholen.

Die Teilnehmer eines Multiplexes sind gewöhnliche SPTS-Streams: Die gesamte Verarbeitung (*SPTS-Streams* — Filterung, Modifikation, Transcodierung) erfolgt vor dem Zusammensetzen am Quell-Stream. Ein SPTS-Stream kann gleichzeitig direkt ausgeliefert werden und Teil eines Multiplexes sein.

Parameter des Multiplexes

Werden am muxer-Eingang des MPTS-Streams festgelegt (Bezeichnungen wie in der Oberfläche):

Transport stream id, Network id, Network name
: Die Kennungen und der Netzwerkname des Transportstroms — sie gelangen in die generierten PAT/SDT/NIT-Tabellen.

Country code (ISO 3166 alpha-3)
: Der Ländercode — wird in der TOT-Tabelle (Zeitzone, lokaler Zeitversatz) und in den regionalen NIT-Deskriptoren verwendet.

Output bitrate (kbps)
: Die Zielbitrate des Multiplexes. Sie wird mit Reserve oberhalb der Summenbitrate aller Programme festgelegt. Der Wert 0 deaktiviert den Ausgleich — die Summenbitrate folgt einfach den Eingangsströmen, ohne Auffüllung mit NULL-Paketen (Stuffing) und ohne T-STD-Modell.

SDT provider name
: Der Anbietername — gemeinsam für alle Dienste des Multiplexes (erweiterte Einstellung).

Parameter der Programme

Werden am muxer-Ausgang des Quell-Streams festgelegt (Bezeichnungen wie in der Oberfläche); der Programmeditor steuert Zusammensetzung, Reihenfolge und Pause sowie — bei abgeschalteter automatischer PID-Zuweisung — die Bearbeitung der PIDs:

SDT service name (empty = stream name)
: Der Dienstname in der SDT-Tabelle; bei leerem Wert wird der Name des Quell-Streams verwendet.

PNR (0 = stream id)
: Die Programmnummer im Multiplex; 0 — es wird die Kennung des Quell-Streams verwendet.

Logical channel number (0 = PNR)
: Die logische Kanalnummer (LCN) für die Empfänger; 0 — stimmt mit der PNR überein.

Service type (0 = auto, ETSI)
: Der Diensttyp gemäß der ETSI-Klassifikation: 0 — wird anhand der Zusammensetzung des Programms (SD/HD, Codec, Radio) automatisch bestimmt, andernfalls manuell festgelegt.

Die Reihenfolge der Programme in den Tabellen des Multiplexes wird durch Ziehen der Zeilen im Programmeditor geändert.

PID-Zuweisung

Der Modus der PID-Zuweisung wird mit dem Schalter „Automatic PID mapping“ am muxer-Eingang gewählt:

- **Eingeschaltet** (Vorgabe) — jedem Programm werden neue PIDs aus dem automatischen Bereich zugewiesen; Kollisionen sind ausgeschlossen.
- **Ausgeschaltet** — die ursprünglichen PIDs der Quell-Streams bleiben erhalten. Das wird zum Beispiel bei der Migration eines laufenden Multiplexes benötigt, damit die Empfänger keinen erneuten Suchlauf verlangen. Einzelne PIDs lassen sich bei Bedarf im PID-Editor des Programms ändern (Schaltfläche „PID“ im Programmeditor): die PMT PID und die Ziel-PIDs der Elementarströme; Duplikate innerhalb eines Programms lässt der Editor nicht zu. Für die Eindeutigkeit der PIDs zwischen den Programmen sorgt in diesem Modus der Bediener — bei einer Kollision schreibt der Knoten eine Warnung in das Protokoll, der Multiplex arbeitet jedoch weiter. Reservierte PIDs der Quelle (kleiner als 32) können nicht beibehalten werden — diesen wird eine neue PID zugewiesen, mit einer Warnung im Protokoll.

Ist bei einem beteiligten Stream die automatische PID-Anordnung aktiviert (*Automatische PID-Anordnung*), zeigt der Programm-PID-Editor die von ihr bereits vergebenen Identifikatoren statt der ursprünglichen an und weist darauf hin: Zuordnungspaare, die vor dem Aktivieren der Anordnung eingetragen wurden, treffen nicht mehr zu, und solche Elementarströme erhalten ihre PID automatisch. Kritisch ist deshalb das Aktivieren der Anordnung bei einem Stream, der bereits Teil des Multiplexes ist, nicht die Einrichtung von Grund auf.

Der Moduswechsel wird im laufenden Betrieb übernommen, ohne Neustart des Streams.

Erzeugung der Tabellen

Der Multiplexer bildet eigene PSI/SI-Tabellen: PAT, die PMT jedes Programms, SDT (Dienst- und Anbietername), NIT (Netzwerkkennung und Netzwerkname, Dienstliste, LCN) sowie TDT/TOT (Zeitstempel). Die CAT-Tabelle wird gebildet, wenn die Programme Daten von Systemen für bedingten Zugriff enthalten.

Die EIT-Tabellen (Programmführer) erzeugt der Multiplexer nicht, sondern überträgt sie aus den Quell-Streams — darunter auch eine EIT, die an der Quelle aus einem externen EPG erzeugt wurde (*EIT-Generator*).

Überwachung des Multiplexes

Der Zustand des zusammengesetzten Multiplexes wird im Statistikfenster des MPTS-Streams angezeigt (*Stream-Statistik*): Ausgangs-, Ziel- und Eingangsrate, Anzahl der Programme, die Programmtabelle mit Bitrate und Zustand jedes Programms, die TS- und Netzwerkkennungen sowie der Zustand des Schedulers und der Auffüllung.

Übersteigt die Summenbitrate der Programme die Zielbitrate, wird die Warnung „Bitrate-Überlast — Quellprogramme überschreiten die Mux-Zielrate“ ausgegeben. Ein unterdimensionierter Multiplex verliert Pakete der Programme — Ton- und Bildaussetzer sowie Kontinuitätsfehler bei den Empfängern; bei anhaltender Überlast ist auch der Verlust von PSI-Tabellen möglich. Erhöhen Sie die Zielbitrate oder verringern Sie die Bitrate der Programme (zum Beispiel durch Transcodierung der Quellen, *Transcoder*).

Die Konformität des Multiplexes mit TR 101 290 wird vom Monitor des Analysators überwacht (*Analyse nach Programmen*).

1.7.6 Teststreams

Der Testsignalgenerator erzeugt rein softwareseitig einen vollwertigen SPTS-Stream mit Testbild und Tonsignal — eine externe Quelle ist nicht erforderlich. Er dient als Sendeersatz während einer Störung der Hauptquelle, zur Prüfung von Übertragungswegen und Empfängern sowie für Einricht- und Belastungsprüfungen.

Eine Testquelle ist ein Eingang des Typs test-stream eines SPTS-Streams (für MPTS nicht verfügbar). Die Erzeugung übernimmt das Modul der Software-Transkodierung — das entsprechende Paket muss auf dem Knoten installiert sein (*Transcoder*). Da es sich um einen gewöhnlichen Eingang handelt, gelten die allgemeinen Regeln der Redundanz: Der Generator wird als letzter Eintrag in die Eingangsliste gesetzt — er übernimmt den Sendebetrieb automatisch, sobald alle Hauptquellen gestört sind; die Rückkehr auf einen höher priorisierten Eingang erfolgt nach den allgemeinen Regeln der erneuten Prüfung (*Quellenredundanz*).

Der erzeugte Stream durchläuft die übliche Verarbeitungspipeline (*Verarbeitungspipeline*): Er lässt sich analysieren, über Transportprotokolle und per OTT verteilen, im DVR aufzeichnen sowie dem Multiplexer und dem Transkoder zuführen.

Einstellungen des Generators

Sie werden am Eingang test-stream vorgenommen (die Bezeichnungen entsprechen der Oberfläche; ein Teil davon gehört zu den erweiterten Einstellungen):

Video
: Video encoder — der Codec: MPEG-2, H.264 (Standard) oder HEVC; Video bitrate (kbps) — die Videobitrate (Standard 3000); Width und Height — die Auflösung (Standard 1920 × 1080); Frame rate — die Bildrate (Standard 25); GOP interval — die GOP-Länge in Bildern; Video pattern — die Art des Testbilds; Width PAR / Height PAR — das Pixelseitenverhältnis.

Ton
: Audio encoder — der Codec: MPEG-1 Layer II (Standard) oder AAC; Audio waveform — die Form des Tonsignals; Audio frequency (Hz) — die Frequenz des Tons; Audio volume — die Lautstärke (0…1).

Einblendung
: Enable overlay — aktiviert die Einblendung über dem Testbild (standardmäßig aktiviert); Overlay text — beliebiger Text, beispielsweise der Name des Kanals; Time format (strftime) — das Format der eingeblendeten Uhrzeit, standardmäßig Stunden:Minuten:Sekunden.

Transportstrom
: Total bitrate (kbps) — die Gesamtbitrate des Streams mit Auffüllung durch NULL-Pakete (Stuffing) auf einen konstanten Wert, 0 — ohne Auffüllung; PNR, PMT PID, Video PID, Audio PID — die Programmnummer sowie die Kennungen der PMT-Tabelle und der Elementarströme, um den Teststream in den PID-Plan des Netzes einzupassen.

Der Bildschirm „Test-Streams“ der Weboberfläche (*Test-Streams*) befindet sich in Vorbereitung; Testeingänge werden im Stream-Editor konfiguriert (*Editor für Eingang und Ausgang*).

1.7.7 OTT und DVR

Das OTT-Subsystem liefert Streams über HTTP-basierte Protokolle aus: HLS (über MPEG-TS), MPEG-DASH und Low-Latency HLS (über CMAF — fragmentiertes MP4) sowie MPEG-TS over HTTP. HTTPS und HTTP/3 (QUIC) werden unterstützt. Die Auslieferung wird auf der Registerkarte „OTT“ des Stream-Editors (*Stream konfigurieren*) aktiviert und steht ausschließlich SPTS-Streams zur Verfügung; ein Multiplex-Programm wird durch den Demultiplexer in einen SPTS-Stream ausgekoppelt (*Demultiplexer*).

DVR ist ein dauerhaftes Archiv des Streams auf der Festplatte. Es wird parallel zur OTT-Auslieferung geschrieben — ein separater Aufzeichnungsdienst ist nicht erforderlich — und wird über dieselben URLs wiedergegeben wie die Live-Ausstrahlung; unterschiedlich sind nur die Abfrageparameter (*DVR: Archiv des Streams*).

Auslieferungsmodi

Die progressive MPEG-TS-Auslieferung über HTTP wird mit der Einstellung „HTTP-Dienst“ (Wert Live) aktiviert: kontinuierliche Übertragung des ursprünglichen Transportstroms innerhalb einer einzigen HTTP-Antwort.

Die segmentierte Auslieferung wird über die Einstellung „HLS-Dienst“ gesteuert (Werte wie in der Oberfläche):

Peering / HLS
: Einfache Segmentierung. Der Modus ist für die Übertragung eines Streams zwischen Perfect Streamer-Knoten vorgesehen: der ausliefernde Knoten gibt HLS über MPEG-TS aus, der empfangende Knoten verbindet sich mit einem Eingang vom Typ HLS. WebVTT-Untertitel und DRM sind in diesem Modus nicht verfügbar, eine DVR-Aufzeichnung ist nicht konfigurierbar.

OTT / HLS
: Segmentierung, die auf einen schnellen Start der Player bei der OTT-Ausstrahlung optimiert ist (höhere CPU-Last). Ausgeliefert wird HLS über MPEG-TS; die GOP-Ausrichtung der Segmente ist wirksam (*Segmentierer*).

OTT / LL-HLS / DASH
: Auslieferung über CMAF: der Stream erzeugt fMP4-Segmente, auf deren Grundlage MPEG-DASH und Low-Latency HLS ausgeliefert werden. Bei aktivierter Einstellung „Veraltete MPEG-TS-HLS-Chunks“ (standardmäßig aktiviert) wird zusätzlich gewöhnliches HLS über MPEG-TS ausgeliefert; ihre Deaktivierung spart Speicherplatz und CPU. MPEG-DASH und Low-Latency HLS sind ausschließlich in diesem Modus verfügbar.

Low-Latency HLS zerlegt die Medienwiedergabeliste in Teilsegmente (Parts): der Player beginnt die Wiedergabe, ohne ein vollständiges Segment abzuwarten; dabei kommen das blockierende Neuladen der Wiedergabeliste und der Preload-Hinweis zum Einsatz. Die Zieldauer eines Parts wird mit der Einstellung „Zieldauer des LL-Parts (ms)“ festgelegt (erweiterte Einstellung; standardmäßig 500 ms, wird im laufenden Betrieb übernommen und muss kleiner als das minimale Chunk-Intervall sein). Für eine präzise niedrige Latenz enthält das DASH-Manifest die Zuordnung der Medienzeit zu UTC, und die CMAF-Segmente führen Zeitstempel des Produzenten mit.

Bemerkung: Die CMAF-Verpackung überträgt ausschließlich Audiospuren in AAC und AC-3; Spuren in anderen Codecs werden verworfen, worüber ein Alarm ausgelöst wird (*Benachrichtigungen (Alerter)*).

Segmentierer

In den OTT-Modi wird der Stream anhand der PAT/PMT-Tabellen und der Keyframes analysiert, und die Segmente werden nach dem Kriterium eines schnellen Player-Starts geschnitten. Der Schnitt wird über die Einstellungen „Min. Chunk-Intervall (s)“ und „Max. Chunk-Intervall (s)“ gesteuert: die Analyse beginnt beim minimalen Intervall, und wird kein Schnittpunkt gefunden, so wird das Segment beim maximalen Intervall zwangsweise abgeschlossen.

„GOP-ausgerichtete Segmente“ (standardmäßig aktiviert; wirksam im Modus OTT / HLS) — der Segmentierer richtet die Segmentgrenzen an den Random-Access-Punkten aus und unterscheidet zwischen einem IDR- und einem gewöhnlichen I-Frame. Bei Quellen mit Closed-GOP beginnt jedes Segment garantiert mit SPS/PPS/IDR — einem vollständigen Einstiegspunkt; der „Rest“ des vorangehenden GOP wird abgeschnitten. Damit entfällt das Schwarzbild zu Beginn von VOD-Sitzungen, und die Auslieferung entspricht den Anforderungen von HLS (RFC 8216). Der Typ der GOP-Struktur der Quelle ist an den Metriken des Analysators ablesbar (*Vertiefte Analysen*). Wegen der Ausrichtung kann die tatsächliche Segmentdauer das minimale Intervall überschreiten — die in der Wiedergabeliste angegebene Zieldauer gibt das tatsächliche Maximum wieder.

URLs und Autorisierung

Die Auslieferung wird vom Streaming-HTTP-Server des Knotens bedient (*Streaming-HTTP-Server*). Eine URL wird nach folgendem Schema gebildet:

```
http://host:port/hls/<Stream>/<Login>/<Passwort>/index.m3u8
http://host:port/hls/<Stream>/<Login>         — Autorisierung per Token
http://host:port/hls/<Stream>/                — Autorisierung per IP
```

Anstelle von `/hls/` wird `/dash/`, `/llhls/` oder `/http/` eingesetzt; für adaptive Bundles — `/hls/adaptive/` und entsprechende (*Adaptive Bundles*). `<Stream>` ist die Kennung des Streams oder der Name, der mit der Einstellung „Stream-Name in der URL“ festgelegt wurde. Unautorisierter Zugriff ist untersagt: Logins und Berechtigungen der Clients werden in der Weboberfläche konfiguriert (*Benutzer / Logins*).

Fertige URL-Vorlagen mit den aktuell aktivierten Auslieferungsmodi werden im Statistikfenster des Streams angezeigt — Abschnitt „Auslieferungs-URLs“ mit Schaltfläche zum Kopieren (*Stream-Statistik*). Aktive Client-Sitzungen sind auf dem Bildschirm „Clients“ sichtbar (*Clients*).

HTTP/3 (QUIC)

Der Streaming-HTTP-Server kann die OTT-Routen (HLS, DASH, LL-HLS, MPEG-TS over HTTP) über QUIC ausliefern. HTTP/3 wird in den Einstellungen des Streaming-HTTP-Servers aktiviert (*Streaming-HTTP-Server*); es wird dasselbe Zertifikat wie für HTTPS verwendet, HTTPS selbst muss jedoch nicht aktiviert sein — QUIC lauscht getrennt und liest die Zertifikatsdateien direkt. Genau diese werden benötigt: ohne Zertifikat und Schlüssel startet der HTTP/3-Listener nicht.

Browser-Clients wechseln über die standardmäßige Alt-Svc-Ankündigung zu QUIC: die Ankündigung wird auf ausdrückliche Anforderung des Clients ausgegeben — mit dem Parameter `?h3` an der Master-URL. Kommt die QUIC-Verbindung nicht zustande (geschlossener UDP-Port, nicht vertrauenswürdiges Zertifikat), verbleibt der Client transparent auf HTTPS — die Ausstrahlung wird nicht unterbrochen. Administrative Anfragen werden ausschließlich über TCP bedient. Low-Latency HLS und DASH über QUIC werden inkrementell ausgeliefert — die Parts gehen an den Client, sobald sie bereitstehen.

WebVTT-Untertitel

Die Einstellung „Teletext-Untertitel in WebVTT“ (standardmäßig aktiviert, wirksam in den OTT-Modi) wandelt die Untertitel DVB Teletext / DVB Subtitling des Eingangsstroms in WebVTT-Spuren um: der HLS-Master-Wiedergabeliste wird eine Untertitelspur hinzugefügt, dem DASH-Manifest ein Text-Adaptationset. Das Vorhandensein von Untertiteln in der Quelle ist in den Medieninformationen im Statistikfenster ersichtlich. Die Untertitel werden parallel zu den Mediensegmenten im DVR archiviert und stehen bei der Archivwiedergabe zur Verfügung; durch DRM werden sie nicht verschlüsselt. Zur Diagnose dient die Einstellung „Untertitelmeldungen protokollieren“ (erweiterte Einstellung).

Adaptive Bundles

Ein adaptives Bundle fasst mehrere OTT-Streams desselben Kanals mit unterschiedlichen Bitraten unter einer einzigen HLS-Master-Wiedergabeliste oder einem DASH-Manifest zusammen — der Player wechselt selbsttätig zwischen den Varianten entsprechend der verfügbaren Bandbreite.

Ein Bundle wird als Stream vom Typ „Adaptiv“ angelegt (*Neuer Stream*, Bundle-Editor — *Adaptives Bundle*). Die Teilnehmer sind SPTS-Streams mit aktiviertem OTT-Modus: OTT / HLS für adaptives HLS über MPEG-TS, OTT / LL-HLS / DASH für adaptives DASH und Low-Latency HLS. Für jeden Teilnehmer lässt sich die Bitrate manuell festlegen; beim Wert 0 wird die gemessene Bitrate übernommen.

Die Master-URLs des Bundles werden nach den Schemata `/hls/adaptive/`, `/dash/adaptive/`, `/llhls/adaptive/` mit denselben Autorisierungsvarianten gebildet. Ein einem Client für ein adaptives Bundle gewährter Zugriff erstreckt sich automatisch auf alle darin enthaltenen Streams.

Inhaltsschutz (DRM)

Die OTT-Auslieferung eines Streams kann verschlüsselt werden. Die Verschlüsselung erfolgt einmalig, im Moment der Segmentbildung: die Live-Auslieferung und das DVR-Archiv erhalten dieselben verschlüsselten Daten; das Archiv wird in verschlüsselter Form auf der Festplatte gespeichert. Die Einstellungen sind im Abschnitt „Inhaltsschutz“ der Registerkarte „OTT“ zusammengefasst; die Übernahme einer beliebigen davon startet die OTT-Auslieferung des Streams neu.

Modi (Feld „DRM“, Werte wie in der Oberfläche):

HLS AES-128
: Vollständige Verschlüsselung der gesamten TS-Segmente; die Entschlüsselung übernimmt jeder standardkonforme HLS-Player. Erfordert „HLS-Dienst“ = OTT / HLS.

CENC cenc (DASH), CENC cbcs (DASH)
: Frameweise Verschlüsselung nach MPEG Common Encryption (ISO/IEC 23001-7) — Schema cenc (AES-CTR) oder cbcs (musterbasiertes AES-CBC, Ökosystem Apple FairPlay). Sie erfordern den Modus OTT / LL-HLS / DASH mit deaktivierter Auslieferung der TS-Chunks; die Auslieferung erfolgt ausschließlich über MPEG-DASH.

Jeder Schutzmodus erfordert einen deaktivierten „HTTP-Dienst“ — die progressive Auslieferung umgeht die Verschlüsselung. Eine unverträgliche Kombination von Einstellungen weist der Server unter Angabe des Grundes zurück.

Schlüssel und Schlüsselzustellung:

- „Inhaltsschlüssel (32 hex)“ ist der Verschlüsselungsschlüssel (für statisches AES-128 und für CENC). „Schlüssel-ID (CENC KID)“ ist die Kennung des CENC-Schlüssels; bei leerem Feld wird die Kennung deterministisch aus der Kennung des Streams abgeleitet.

- „Schlüsselzustellung“ (AES-128, Werte wie in der Oberfläche): Built-in (PSS serves the key) — der Server gibt den Schlüssel selbst über eine Sitzungs-URL mit derselben Autorisierung wie die Segmente aus; External key server — das Schlüssel-Tag in der Wiedergabeliste verweist auf „Schlüssel-URI (EXT-X-KEY)“, die integrierte Zustellung wird abgeschaltet. Bei CENC gibt der Server den Playern keine Schlüssel aus — die Schlüsselzustellung übernimmt die DRM-Infrastruktur (Lizenzserver).
- Schlüsselrotation (nur AES-128): „Schlüsselquelle“ = Derived (rotating), „Rotationsgeheimnis“ (mindestens 16 Zeichen) und „Schlüsselfenster (Minuten)“. Der Schlüssel jedes Fensters wird mit einer Einwegfunktion aus dem Geheimnis berechnet; das DVR-Archiv lässt sich über beliebig viele Rotationen hinweg wiedergeben — die Fensternummer wird aus dem Aufzeichnungszeitpunkt des Segments rekonstruiert.

Warnung: Ein Wechsel der „Schlüsselidentität“ eines Streams — des Inhaltsschlüssels, des Geheimnisses oder der Rotationseinstellungen — macht ein zuvor aufgezeichnetes verschlüsseltes Archiv unlesbar; die Weboberfläche verlangt eine Bestätigung solcher Änderungen. Die Schlüssel werden in der Serverkonfiguration im Klartext gespeichert — beschränken Sie den Zugriff auf die Konfigurationsdateien und die Sicherungskopien.

DVR: Archiv des Streams

Die Aufzeichnung wird auf der Registerkarte „OTT“ im Abschnitt „DVR“ aktiviert (sichtbar in den OTT-Modi):

„DVR-Speicher“
: Der Speicher, in den das Archiv geschrieben wird (*DVR-Speicher*). Der Wert „Keiner“ bedeutet, dass die Aufzeichnung deaktiviert ist.

„Aufbewahrung (Stunden)“
: Archivtiefe des Streams, von 1 Stunde bis 90 Tage. Ältere Segmente werden durch die reguläre Bereinigung entfernt.

„Geschütztes Minimum (Stunden)“
: Untere Schutzschwelle: die Bereinigung entfernt niemals Aufzeichnungen, die jünger als dieser Wert sind, auch nicht bei knappem Speicherplatz auf der Festplatte.

Die Aufzeichnung beginnt automatisch, sobald der Stream in den Betriebszustand übergeht. Das Lösen der Bindung an den Speicher hält die Aufzeichnung an (die Dateien verbleiben auf der Festplatte, die Archivwiedergabe wird beendet); bei einem Wechsel des Speichers beginnt die Aufzeichnung von Neuem, die Dateien werden nicht übertragen. Destruktive Änderungen bestätigt die Weboberfläche mit dem Dialog „DVR-Änderungen bestätigen“.

DVR-Speicher

Die Speicher werden auf dem Bildschirm „DVR-Speicher“ konfiguriert (*DVR-Speicher*). Für jeden werden der Pfad zum Verzeichnis (nach dem Anlegen unveränderlich), die Grenze der Festplattenbelegung in Prozent und — in den erweiterten Einstellungen — das Bereinigungsintervall, die Karenzzeit und die Abschneideportion bei Überfüllung, die Notfallreserve an Speicherplatz sowie die Hysterese festgelegt. Pro Festplatte wird ein Speicher empfohlen: mehrere Speicher auf einer gemeinsamen Festplatte konkurrieren um den freien Speicherplatz. Das Entfernen eines Speichers aus der Konfiguration löscht die Dateien nicht von der Festplatte.

Die Archivbereinigung arbeitet auf mehreren Ebenen:

- nach Tiefe — Segmente, die älter als der Wert „Aufbewahrung (Stunden)“ des jeweiligen Streams sind;

- nach Speicherplatz — bei anhaltender Überschreitung der Grenze der Festplattenbelegung werden die ältesten Segmente anteilig über alle Streams hinweg entfernt, jedoch keine, die jünger als das geschützte Minimum sind;
- Notfallbereinigung — bei kritischem Speicherplatzmangel wird die Aufzeichnung angehalten, bis wieder freier Speicherplatz verfügbar ist;
- die Bereinigung verwaister Dateien innerhalb der Archive aufgezeichneter Streams erfolgt automatisch; Archivverzeichnisse, die von gelöschten Streams zurückgeblieben sind, werden ausschließlich manuell entfernt — mit der Schaltfläche „Verwaiste Archive bereinigen“ auf dem Bildschirm „DVR-Monitor“ (*DVR-Monitor*).

Der Zustand der Speicher und der aufgezeichneten Streams — Füllstand, Archivgröße, Schreib- und Leselatenzen, aktive Bereinigungsaufgaben — wird auf demselben Bildschirm „DVR-Monitor“ angezeigt. Das Archiv eines einzelnen Streams lässt sich mit der Schaltfläche „DVR-Archiv löschen“ im Statistikfenster des Streams unwiderruflich entfernen; die Aufzeichnung wird dabei fortgesetzt.

Archivwiedergabe (VOD)

Das Archiv wird über dieselben URLs wiedergegeben wie die Live-Ausstrahlung HLS / DASH — unterschiedlich sind nur die Abfrageparameter:

`t=<Zeit>`
: Beginn des VOD-Fensters (Unix-Zeit, Sekunden). `t=0` — ab dem Anfang des Archivs. Das Vorhandensein des Parameters `t` aktiviert den VOD-Modus.

`d=<Sekunden>`
: Dauer des VOD-Fensters. Ohne den Parameter (oder mit `d=0`) — bis zum aktuellen Zeitpunkt. Wirksam nur zusammen mit `t`.

`epg=<Zeit>`
: VOD nach dem Programmführer: der Server ermittelt das zum angegebenen Zeitpunkt aktive EPG-Ereignis und übernimmt dessen Beginn und Dauer als Fenstergrenzen. Erfordert die Bindung des Streams an eine EPG-Quelle — die Felder „EPG-Quelle“ und „EPG-Kanal“ auf der Registerkarte „Stream“ (*Modifikation des Streams*, *EIT-Generator*); die Parameter `t` und `d` werden dabei ignoriert. Für einen Programmkatalog genügt der Oberfläche die Zeit des Ereignisses — die genauen Grenzen berechnet der Server.

Besonderheiten der Wiedergabe:

- Die Fenstergrenzen werden normalisiert: eine Anfrage vor dem Anfang des Archivs wird auf das erste verfügbare Segment nachgezogen; ein vollständig außerhalb des Archivs liegendes Fenster liefert eine leere, aber gültige Wiedergabeliste.
- Die HLS-Wiedergabeliste für VOD ist geschlossen (mit Abschlusskennzeichen); das DASH-Manifest ist statisch. Aufzeichnungslücken werden in DASH als eigene Perioden abgebildet — die Player spulen ohne besondere Einstellungen über die Grenze hinweg.
- Erreicht ein Client die rechte Grenze des Fensters, so werden die fehlenden Segmente aus dem Live-Speicher des Streams ausgeliefert — ohne Umleitungen und ohne erneute Autorisierung.
- Eine offene VOD-Sitzung schützt ihr Fenster bis zum Schließen der Sitzung vor der regulären Bereinigung; die Notfallbereinigung und das geschützte Minimum haben Vorrang.
- WebVTT-Untertitel werden auch aus dem Archiv wiedergegeben (*WebVTT-Untertitel*).
- Adaptive Bundles unterstützen dieselben VOD-Parameter. In das DASH-Manifest gelangen nur die Varianten mit angebundenem DVR-Speicher; in die HLS-Master-Wiedergabeliste gehen alle Varianten

ein, doch Varianten ohne Archiv beantworten eine Archivanfrage mit einem Fehler, und der Player überspringt sie.

1.7.8 Transcoder

Transcoder codieren Video und Ton eines Streams um: Sie senken die Bitrate für OTT oder für einen Multiplex, ändern Codec und Auflösung und bereiten die Varianten adaptiver Bundles vor. Jeder Transcoder läuft als eigener Prozess des Knotens; das Schema „eins zu vielen“ wird unterstützt — aus einem Decoder lassen sich mehrere Ausgangsstreams mit unterschiedlichen Codiereinstellungen gewinnen. Transcodierung steht ausschließlich SPTS-Streams zur Verfügung; der Quellstream muss sowohl Video als auch Ton enthalten.

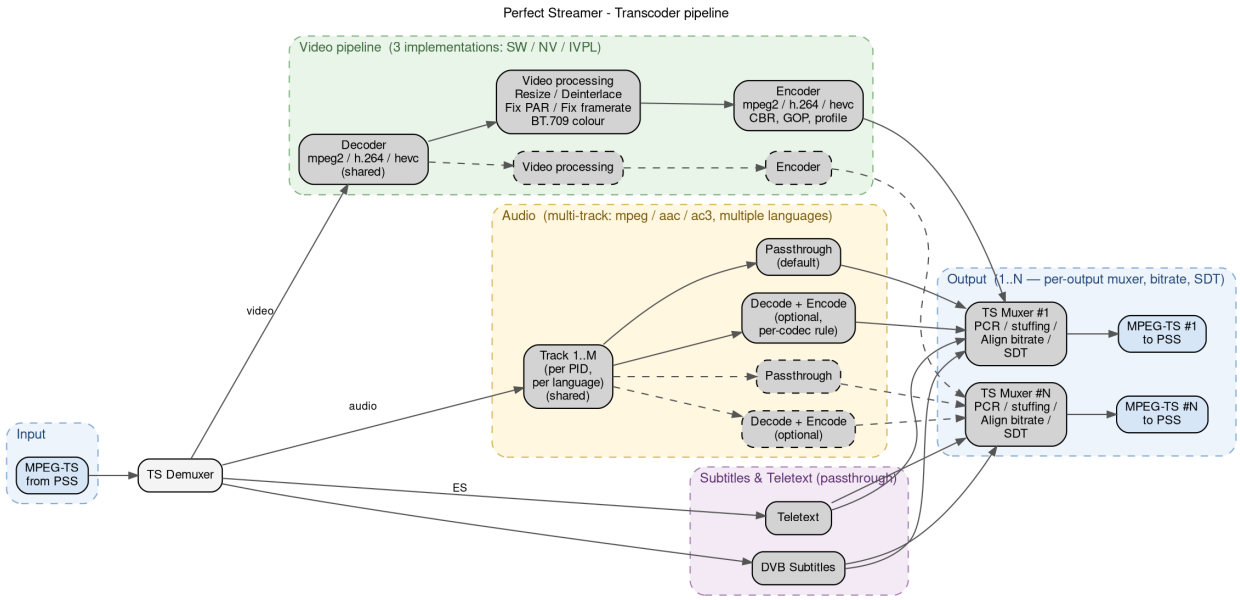


Abb. 4: Architektur des Transcoders: ein Decoder (gemeinsam) und N unabhängige Zweige „Verarbeitung + Encoder“; die Tonspuren werden getrennt vom Video verarbeitet, Untertitel und Teletext werden durchgereicht.

Verschaltung

Ein Transcoder verbindet zwei oder mehr Streams:

1. Am Quellstream wird ein **Ausgang vom Typ transcoder** hinzugefügt — das ist der Decoder. In seinen Einstellungen wird das Backend der Transcodierung ausgewählt (siehe unten).
2. Am Ergebnisstream wird ein **Eingang vom Typ transcoder** hinzugefügt — das ist der Encoder. In seinen Einstellungen wird der laufende Quell-Decoder ausgewählt (Feld Decoder).
3. Schritt 2 wird für jede weitere Variante wiederholt — alle nutzen denselben gemeinsamen Decoder.

Danach ist der Ergebnisstream ein gewöhnlicher SPTS-Stream: Er kann über OTT ausgeliefert, in den DVR aufgezeichnet oder dem Multiplexer zugeführt werden.

Backends und Codecs

Das Backend wird am Decoder ausgewählt (Feld Transcoder backend, die Werte wie in der Oberfläche):

Video pass-through
: Das Video wird ohne Umcodierung durchgereicht, transcodiert wird ausschließlich der Ton. Der Modus wird vom Software-Transcoder ausgeführt — das Paket der Software-Transcodierung muss installiert sein (*Transcoder*).

Software (CPU)
: Software-Transcodierung auf dem Prozessor.

NVidia (GPU)
: Hardware-Transcodierung auf Grafikprozessoren von NVIDIA.

Intel VPL (GPU)
: Hardware-Transcodierung auf Grafikprozessoren von Intel.

Für die GPU-Backends wird das Gerät über das Feld GPU device selector ausgewählt (erweiterte Einstellung) — relevant auf Servern mit mehreren Karten. Die Transcoder-Pakete werden separat installiert (*Transcoder*); die installierten Backends, deren Versionen und die gefundenen Geräte werden auf dem Bildschirm „Über“ angezeigt (*Über das System*).

Codecs des Decoders — MPEG-2, H.264, HEVC; Ton — MPEG (Layer 1–3), AAC, AC-3. Codecs des Encoders — H.264 und HEVC (der Encoder von NVIDIA unterstützt kein MPEG-2; MPEG-2 codieren die Backends Software und Intel VPL); Ton — MPEG Layer II und AAC.

Besonderheiten des Decoders:

- Zeilensprungstreams (interlace) werden am Eingang und am Ausgang unterstützt; bei H.264 und HEVC wird das Format interlace alternate (getrennte Halbbilder) in interleaved umgewandelt;
- für HEVC wird das Profil Main10 mit BT.709 (SDR) und BT.2020 (HDR) unterstützt; der HEVC-Encoder verwendet stets das Profil Main mit BT.709;
- eine variable Bildrate (VFR) wird in eine konstante umgewandelt;
- Streams ohne IDR-Bilder werden unterstützt.

Einstellungen des Decoders

Sie werden am Ausgang vom Typ transcoder des Quellstreams festgelegt (Bezeichnungen wie in der Oberfläche; alle nachfolgend aufgeführten sind erweiterte Einstellungen):

Fix PAR (N/D), Fix framerate (N/D)
: Hinweise zur Korrektur der Daten des Eingangsstreams — sie wandeln den Stream nicht um, sondern berichtigen Parameter, wenn die Quelle sie nicht oder falsch meldet. Fix PAR — das Seitenverhältnis des Pixels als Bruch (zum Beispiel 16/9 für Wide SD); Fix framerate — die Bildrate als Bruch, falls sie im Stream fehlt (PAL — 25/1, NTSC — 30000/1001, Film — 24/1).

Convert to BT.709
: Konvertierung der SD-Farbformate (BT.470-2 PAL, SMPTE 170M NTSC) nach BT.709.

Watchdog timeout
: Timeout des Watchdog-Neustarts des Transcoder-Prozesses.

Der Transcoder-Prozess führt stets ein ausführliches Protokoll — eine separate Einstellung für die Ablaufverfolgung ist nicht erforderlich (*Betriebsüberwachung*).

Einstellungen des Encoders

Sie werden am Eingang vom Typ transcoder des Ergebnisstreams festgelegt:

Encoder
: Videocodec: MPEG-2, H.264 (Standard) oder HEVC.

Video profile (H.264)
: Codierprofil von H.264: Main oder High.

Video bitrate (kbps)
: Bitrate des Videos. Die Codierung erfolgt stets im CBR-Modus; die Gesamtbitrate des Streams fällt durch die Tonspuren höher aus.

Resize
: Änderung der Bildgröße: proportionale Verkleinerung um den Faktor 2 oder 4; Anpassung an SD — die Varianten SD, SD PAL und SD NTSC, alle mit dem Breitbildverhältnis 16:9; Hochskalierung von SD auf HD; Vorgabe der Breite oder der Höhe (die jeweils andere Seite wird proportional nachgerechnet). Eine beliebige Größe wird nicht unterstützt — sie würde die Proportionen verzerren.

Deinterlace
: Umwandlung von Zeilensprungvideo in Vollbildvideo (standardmäßig aktiviert; bei der Hochskalierung von SD auf HD wird sie automatisch angewendet).

Einstellungen für Qualität und Struktur des Streams (erweiterte Einstellungen):

GOP interval
: Intervall der Keyframes in Bildern. Standardmäßig 25 — eine Sekunde bei 25p; empfohlen, wenn Player zu einem beliebigen Zeitpunkt auf den Stream zugreifen.

B-frames, Lookahead
: Sie erhöhen die Codierqualität; die empfohlenen Werte sind 3 beziehungsweise 20–50 Bilder.

Speed preset
: Voreinstellung der Codiergeschwindigkeit von 1 bis 7: je kleiner der Wert, desto höher die Qualität und desto größer die Last. Standardmäßig 4.

Framerate resample
: Ausdünnung der Bildrate um den Faktor 2 oder 4.

Encoder PMT PID, PAT/PMT interval (90kHz)
: PID der PMT-Tabelle und Wiederholintervall der Tabellen PAT/PMT im Ausgangsstream — falls der Stream in den Netzplan eingepasst werden muss.

Unverträgliche Parameterkombinationen sind möglich — die Fehler sind im Protokoll des Transcoders sichtbar (*Betriebsüberwachung*).

Tonverarbeitung

Standardmäßig werden alle Tonspuren ohne Umcodierung vom Eingang an den Ausgang durchgereicht. Überzählige Spuren werden durch die PID-Filter am Eingang des Streams entfernt (*MPEG-TS-Filterung*).

Die Umcodierung des Tons wird über Regeln getrennt für jeden Quellcodec konfiguriert (erweiterte Einstellungen): Eine Spur kann unverändert durchgereicht, umcodiert (MPEG — nach AAC; AAC — nach MPEG Layer II; AC-3 — nach MPEG Layer II oder AAC) oder entfernt werden (Wert Skip). Zusätzlich werden die Bitrate des Tons (Audio bitrate, standardmäßig 128 kbit/s) und die Lautstärke des 5.1-Downmix (5.1 downmix volume) festgelegt. Bleibt nach Anwendung der Regeln im Ausgangsstream keine einzige Tonspur übrig, meldet der Encoder einen Fehler im Protokoll.

PCR und Bitrate des Ausgangsstreams

Der Multiplexer des Transcoders erzeugt neue, konsistente PCR-Marken. Standardmäßig ist der Ausgangsstream VBR: Die Gesamtbitrate folgt der tatsächlichen Bitrate von Video und Ton, und die PCR-Genauigkeit nach TR 101 290 ist auf der Empfangsseite nicht gewährleistet. Ist der transcodierte Stream für die DVB-Ausstrahlung bestimmt, aktivieren Sie den CBR-Modus am Empfängerstream — das Stuffing gleicht die Bitrate an und berechnet die PCR neu (*Bitratensteuerung*).

Betriebsüberwachung

Der Bildschirm „Transcoder“ (*Transcoder*) zeigt alle Instanzen an: Jede Zeile ist eine eigene Instanz (ein Decoder und die mit ihm verbundenen Encoder) mit Zustand, Backend-Typ, Betriebsdauer, CPU-Last und Speicherbelegung. Das Fenster „Transcoder-Instanz“ (*Transcoder-Instanz*) zeigt die Quelle, die Liste der Encoder mit den Parametern des Ausgangsvideos, die Ressourcen des Prozesses sowie die Protokolle des aktuellen und des vorherigen Starts — die erste Anlaufstelle bei der Diagnose von Störungen.

Die Auslastung der GPU-Geräte wird auf dem Bildschirm „System-Monitor“ angezeigt (*System-Monitor*). Das Überschreiten der Schwellenwerte für CPU, Speicher und GPU-Auslastung sowie die Nichtverfügbarkeit des konfigurierten Backends lösen Alarme aus (*Katalog der Alert-Codes*).

1.7.9 DVB-Empfänger

Perfect Streamer arbeitet mit den im System installierten DVB-Adaptern: unterstützt werden die Standards DVB-S, DVB-S2, DVB-T, DVB-T2, DVB-C und ATSC. Zusätzlich sind die T2-MI-Dekapselung (ETSI TS 102 773) sowie die Entschlüsselung BISS-1 / BISS-E und CI/CAM implementiert. Grundvoraussetzung ist ein korrekt installierter und funktionierender Adaptertreiber; der Bildschirm „DVB-Adapter“ (*DVB-Adapter*) ist verfügbar, sobald im System DVB-Geräte erkannt werden. DVB-ASI-Geräte werden anders angebunden — über einen Datei-Eingang des Streams auf den Gerätepfad, ohne einen DVB-Adapter-Eintrag.

Adapter

Ein Adapter ist ein Eintrag, der einen physischen Tuner (das Paar „Adapter/Gerät“ im System) an die Empfangsparameter bindet; ist das Signal eingerastet, liefert der Adapter den MPTS-Multiplex des Transponders. Ein Adapter wird im Fenster „DVB-Adapter hinzufügen“ (*DVB-Adapter-Editor*) hinzugefügt:

- **Modus**: Stream — Empfang des Streams; Frontend monitor — ausschließlich Signalüberwachung, ohne den Stream zu lesen (der veraltete Wert Scanner wird nicht verwendet). Der Modus und das „Übertragungssystem“ werden bei der Anlage festgelegt und sind danach unveränderlich.
- **Hardware** — der Tuner wird aus der Liste der im System erkannten Tuner ausgewählt; belegte Tuner sind gekennzeichnet.
- **Abstimmparameter** — hängen vom Übertragungssystem ab: Frequenz (für Satellit in MHz, für terrestrischen Empfang und Kabel in kHz), „Symbolrate, ksym/s“, „FEC“, „Modulation“, „Spektrale Inversion“, „Stream-ID (ISI/PLP)“ — die Stream-Kennung eines DVB-S2-Multistreams oder die DVB-T2-PLP auf Tuner-Ebene; für den terrestrischen Empfang — „Bandbreite“, „Übertragungsmodus“, „Schutzintervall“, „Hierarchie“.

Für Satellitensysteme wird der Abschnitt „LNB“ konfiguriert: die Oszillatorfrequenzen (universelles Ku — 9750/10600, Umschaltung bei 11700 MHz; für zirkulare Polarisation und das C-Band gelten eigene Werte), „Polarisation“ (der Adapter steuert die Speisung des Konverters: 18 V — horizontal, 13 V — vertikal), „DiSEqC-Port“ und „22-kHz-Ton erzwingen“.

„LNB-Sharing“ deaktiviert die Steuerung von Speisespannung, Polarisation und Ton — diese werden von einem anderen Empfänger gesetzt, der den Konverter besitzt. Der Modus wird verwendet, wenn mehrere Tuner über einen Verteiler an ein und demselben LNB angeschlossen sind (empfangen werden können nur die vom Besitzer gewählte Polarisation und das gewählte Band), sowie um verschiedene PLPs desselben Transponders auf getrennte Adapter zu verteilen.

Entschlüsselungsschlüssel und die CAM-Programmliste werden im laufenden Betrieb übernommen; Änderungen an den Abstimmparametern stimmen das Frontend neu ab. Die Einstellung „EPG exportieren“ schreibt die Programmvorschau des Multiplex in die EPG-Datenbank des Knotens (*EPG/XMLTV-Import*).

Anbindung an Streams

Der empfangene Multiplex wird auf zwei Arten genutzt:

- **vollständig** — ein Eingang vom Typ dvb an einem MPTS-Stream: der gesamte Transponder geht in den Transit oder in das Remultiplexing (*MPTS-Streams*);
- **als einzelne Programme** — ein Demultiplexer-Eingang an SPTS-Streams (*Demultiplexer*): das Programm wird anhand der PNR aus der Liste der erkannten Dienste ausgewählt.

In der Programmliste des Adapters ist ersichtlich, welche Programme bereits genutzt werden: die Markierung „Aktiv“ mit einem Zusatz — als einzelnes Programm extrahiert (SPTS) oder als Teil des vollständigen Multiplex gelesen (MPTS). Solange mindestens ein Stream auf den Adapter verweist, kann er nicht gelöscht werden.

Scan

Um die Empfangsparameter nicht von Hand eingeben zu müssen, ist ein Transponder-Scanner eingebaut (Fenster „DVB-Scan“, *DVB-Scan*). Der Scanner geht die Transponder des ausgewählten Satelliten oder des regionalen Bands durch, rastet auf jeden einzelnen ein, sammelt die PSI/SI-Tabellen und stellt eine Liste der Multiplexe mit ihren Programmen zusammen: PNR, Dienstname, Anbieter, Verschlüsselungskennzeichen und die wichtigsten PIDs. Jeder gefundene Multiplex lässt sich mit einer einzigen Schaltfläche in einen fertig konfigurierten Adapter überführen; es können auch mehrere zugleich ausgewählt und hinzugefügt werden.

Besonderheiten:

- Frequenzquelle ist der „Katalog“ (Verzeichnisse der Satelliten und der regionalen Bänder) oder der „Blind-Scan“ über ein synthetisiertes Frequenzraster mit optionalem Bereich und optionaler Schrittweite;
- für Satellit wird eine „LNB-Voreinstellung“ gewählt (universelles Ku, zirkulare Polarisation, C-Band) oder es werden eigene Oszillatorfrequenzen angegeben;
- der Scanner belegt den physischen Tuner vollständig — erforderlich ist ein Tuner, der weder vom System noch von anderen aktiven Einträgen belegt ist; ein pausierter Eintrag gibt seinen Tuner frei, daher kann ein laufender Adapter vorübergehend pausiert und für den Scan genutzt werden; freie und belegte Tuner sind in der Liste ersichtlich;
- in T2-MI steigt der Scanner nicht hinein: die Dekapselungsparameter werden erst in den Einstellungen des angelegten Adapters festgelegt.

T2-MI-Dekapselung

T2-MI (T2-Modulator Interface, ETSI TS 102 773) ist ein Format für den Transport von DVB-T2-Strömen über Verteilstrecken, üblicherweise satellitengestützte DVB-S/S2 (Multistream eingeschlossen): der äußere Transponder führt einen oder mehrere T2-MI-Träger, von denen jeder BBFRAME mit einer oder mehreren PLPs kapselt. Nach der Dekapselung wird der innere MPEG-TS mit seinen Programmen und PSI/SI-Tabellen gewonnen.

Perfect Streamer arbeitet im Multi-Carrier-Modus: ein einzelner Adapter liefert gleichzeitig den äußeren Multiplex und sämtliche dekapselten T2-MI-Träger. Einstellungen (Abschnitt „T2-MI“):

„T2-MI-Modus“
: Off — die Dekapselung ist deaktiviert, der äußere Stream wird unverändert weitergegeben (wird T2-MI im Multiplex erkannt, weist der Knoten im Protokoll darauf hin); Manual — die Dekapselung ist stets aktiv; Auto — die Träger werden anhand der PMT-Tabellen automatisch erkannt, fehlen diese, arbeitet der Adapter als gewöhnlicher Multiplex.

«T2-MI PID»
: Vorabanlage eines Trägers auf einer bekannten PID, bevor die Tabellen eintreffen; die übrigen Träger werden dennoch automatisch erkannt.

«T2-MI PLP»
: Die PLP-Nummer, die aus jedem Träger extrahiert wird; einen trägerspezifischen Wert gibt es nicht — für verschiedene PLPs werden getrennte Adapter mit LNB-Sharing konfiguriert. Treffen keine BBFRAME mit der angegebenen PLP ein, schreibt der Knoten die Liste der beobachteten PLPs in das Protokoll; dieselbe Liste ist in der Adapter-Statistik zu sehen. Das Feld „T2-MI-TSID-Filter (−1 = beliebig)“ ist reserviert.

Die Programme der inneren Multiplexe erscheinen in der gemeinsamen Programmliste des Adapters mit der Kennzeichnung ihrer PLP. Um ein solches Programm an einen SPTS-Stream anzubinden, wird eine zusammengesetzte PNR verwendet — Trägernummer × 65536 + PNR innerhalb des Trägers (*Demultiplexer*); Träger 0 ist der äußere Multiplex. Dienstliche Duplikate und Testfeeds innerhalb der Träger werden automatisch mit „[test]“ markiert und wirken sich nicht auf die Zustandsprüfung aus.

Entschlüsselung

BISS. Ein Software-Entschlüssler für BISS-1 / BISS-E (Abschnitt „BISS-Entschlüsselung“). Festgelegt werden Paare aus „Slot — Schlüssel“; auf einem Adapter können mehrere Entschlüssler gleichzeitig aktiv sein:

- Slot — die Programmnummer (PNR) im äußeren Multiplex: alle PIDs des Programms werden entschlüsselt;
- Slot — die PLP eines T2-MI-Trägers: der Schlüssel wird vor der Dekapselung auf den gesamten Träger angewendet. Für verschlüsselte Multistream-Ströme ist dies zwingend — ohne Schlüssel verwirft die Dekapselungsstufe jedes verschlüsselte Paket.

Das Schlüsselformat ergibt sich aus der Länge: 12 Hex-Ziffern — BISS-1 (die Prüfbytes werden automatisch berechnet), 16 — BISS-E. Schlüssel werden im laufenden Betrieb übernommen, ohne Neustart des Adapters. Ob ein Schlüssel wirkt, zeigen die Zähler in der Adapter-Statistik: steigt „entschlüsselt“ — der Schlüssel funktioniert; steigt „Schlüssel fehlt“ — der Schlüssel ist nicht gesetzt oder auf der falschen Ebene gesetzt; steigt „Fehler“ — der Schlüssel ist falsch.

CI/CAM. Ein Adapter mit installiertem CAM-Modul entschlüsselt ausgewählte Programme über das Common Interface (Abschnitt „CI/CAM-Entschlüsselung“): die Programmnummern werden aufgeführt, das Modul entschlüsselt sie unmittelbar im Empfangspfad. Die Liste wird im laufenden Betrieb übernommen und ist vom Übertragungssystem unabhängig. Der Zustand des Moduls, die unterstützten CA-Systeme und der Entschlüsselungsstatus jedes Programms sind in der Adapter-Statistik und im Bildschirm „Hardware“

(*Hardware*) ersichtlich; das Entnehmen des Moduls, ein fehlendes Abonnement und CA-Fehler lösen Alarme aus (*Katalog der Alert-Codes*).

„CA aus entschlüsselter Ausgabe entfernen“ (erweiterte Einstellung) räumt die CAS-Tabellen und -Meldungen aus den bereits entschlüsselten Programmen — analog zum Stream-Filter „CAT / ECM / EMM löschen“ (*MPEG-TS-Filterung*).

Signalüberwachung

In der Adapterliste sind Signaleinrastung, Signalstärke, SNR, BER und die Eingangsbitrate jedes Adapters ersichtlich. Das Statistikfenster des Adapters (*Adapter-Statistik*) zeigt den Signalverlauf (Signalstärke, SNR, BER, unkorrigierte Blöcke), die aktuellen Frontend-Kennwerte, die Programmliste mit Mediendaten, die Zähler der T2-MI-Träger und der Entschlüssler, den Zustand des CAM-Moduls sowie das Protokoll; die kumulativen Zähler werden mit der Schaltfläche „Statistik zurücksetzen“ gelöscht. Zur Überwachung der Antenne ohne Lesen des Streams wird der Adapter im Modus Frontend monitor angelegt.

Verlust der Einrastung, ein Abfall von Pegel oder Qualität des Signals sowie eine Zunahme von BER-Fehlern und unkorrigierten Blöcken lösen Alarme mit konfigurierbaren Schwellen aus (*Benachrichtigungen (Alerter)*, die Schwellen stehen in den Einstellungen des Alerters).

Empfangspuffer. „Demux-Puffer (Blöcke)“ (erweiterte Einstellung) legt die Größe des Empfangs-Ringpuffers in Blöcken zu je 64 KB fest. Der Vorgabewert 512 (32 MB) deckt Volltransponder-Szenarien in DVB-S/S2 mit mehreren Verbrauchern ab; auf schwach ausgelasteten oder eingebetteten Systemen kann er verringert werden. Bei Pufferüberläufen (der Zähler ist in der Statistik, die Meldung im Protokoll zu sehen) erhöhen Sie den Puffer oder verringern Sie das Volumen der empfangenen Daten — verzichten Sie beispielsweise auf das Lesen des vollständigen Multiplex, wenn er nicht benötigt wird. Berücksichtigen Sie den Speicherbedarf: jeder Adapter reserviert „Blöcke × 64 KB“.

Zugriffsrechte auf Geräte

Werden die DVB-Adapter in Perfect Streamer nicht angezeigt, obwohl sie im System vorhanden sind, erteilen Sie dem Dienst die Rechte an den DVB-Geräten. Legen Sie die Datei `/etc/udev/rules.d/99-dvb-permissions.rules` mit der folgenden Zeile an:

```
SUBSYSTEM=="dvb", GROUP="video", MODE="0660"
```

Fügen Sie anschließend den Dienstbenutzer der Gruppe video hinzu und wenden Sie die Rechte an:

```
sudo usermod -aG video pss
sudo chown -R root:video /dev/dvb/*
sudo systemctl restart pss
```

Nach einem Neustart des Systems setzt die udev-Regel die Rechte automatisch.

1.7.10 RTMP-Veröffentlichung und -Empfang

Perfect Streamer arbeitet mit RTMP und RTMPS als Client: Er veröffentlicht einen SPTS-Stream an einen externen Empfangspunkt (den Ingest einer Videoplattform oder einen beliebigen RTMP-Empfänger) und empfängt einen Stream von einer RTMP-Quelle auf Anforderung (Pull). Der Empfang eingehender Veröffentlichungen (Push-Ingest, „wie bei Videoplattformen“) wird nicht unterstützt — verwenden Sie für solche Szenarien eine Brücke über eine externe Anwendung (*Sonstige Ein- und Ausgänge*). RTMP steht nur SPTS-Streams zur Verfügung.

Veröffentlichung (Ausgang vom Typ rtmp)

Der Ausgang wird auf der Registerkarte „Ausgänge“ des Streams hinzugefügt (die Feldnamen entsprechen der Oberfläche):

URL (rtmp(s)://…)
: Die Adresse des Empfangspunkts einschließlich des Streamschlüssels, so wie ihn die empfangende Plattform ausgibt.

Bind interface
: Die lokale Schnittstelle der ausgehenden Verbindung.

Client timeout
: Das Verbindungs-Timeout, standardmäßig 10 Sekunden.

Allow HEVC (Enhanced RTMP)
: HEVC wird über die Erweiterung Enhanced RTMP veröffentlicht (standardmäßig aktiviert; erweiterte Einstellung). Für Empfänger ohne Unterstützung von Enhanced RTMP schalten Sie sie aus — dann wird nur H.264 veröffentlicht.

Verify TLS certificate
: Für RTMPS: Überprüfung des Serverzertifikats (standardmäßig deaktiviert; erweiterte Einstellung).

Das Veröffentlichungsformat ist ein einzelnes Programm: ein Video in H.264 oder HEVC und höchstens eine AAC-Tonspur. Eine mehrspurige Quelle wird durch PID-Filter am Streameingang eingeschränkt (*MPEG-TS-Filterung*); Tonspuren in anderen Formaten werden verworfen. Ist die Spurbelegung mit RTMP nicht kompatibel, findet keine Veröffentlichung statt — der Grund wird in der Ausgangsstatistik angezeigt (*Stream-Statistik*).

Empfang (Eingang vom Typ rtmp)

Ein Eingang vom Typ rtmp holt einen Stream von einer RTMP-Quelle über deren URL: Der Knoten verbindet sich als Client, der empfangene Stream wird nach MPEG-TS remultiplexiert und durchläuft die übliche SPTS-Pipeline (*Verarbeitungspipeline*). Die Einstellungen sind dieselben: URL (rtmp(s)://…), Bind interface, Client timeout und Verify TLS certificate für RTMPS.

1.7.11 Sonstige Ein- und Ausgänge

Einzelheiten zur Konfiguration der Ein- und Ausgänge, die in der Protokollübersicht (*Weitere Eingänge und Ausgänge*) kurz aufgeführt sind: Empfang von RTSP-Quellen und über HLS/HTTP, Arbeit mit Dateien und Geräten, benannte Pipes und die Brücke zu externen Anwendungen. Die Feldbezeichnungen entsprechen der Oberfläche.

RTSP-Eingang

Empfang eines Videostreams von RTSP-Quellen — IP-Kameras und RTSP-Servern. Der Knoten öffnet eine RTSP-Sitzung, liest die Elementarströme und remultiplexiert sie in den integrierten MPEG-TS; anschließend durchläuft der Stream die reguläre SPTS-Pipeline. Enthält die Quelle keine Tonspuren, wird eine stille Tonspur MPEG-1 Layer II hinzugefügt — die Pipeline benötigt mindestens eine Tonspur. RTSP ist eine Einzelprogrammquelle und steht nur SPTS-Streams zur Verfügung.

Einstellungen:

- URI (rtsp://…) — die vollständige Adresse der Sitzung; Login und Password — die Zugangsdaten, falls der Server eine Autorisierung verlangt;
- Transport — der RTP-Transport innerhalb von RTSP: UDP (Standard; geringe Latenz, empfindlich gegenüber Verlusten), TCP (durchquert NAT und Firewalls) oder HTTP-Tunnel (zum Durchqueren von HTTP-Proxys);
- User-Agent und Cookies (erweiterte Einstellungen) — für Server mit nicht standardmäßiger Autorisierung;
- Client timeout — das Zeitlimit für Öffnen und Lesen, standardmäßig 10 Sekunden.

HLS-/HTTP-Eingang

Empfang von MPEG-TS über HTTP: eine gewöhnliche HLS-Playlist oder eine kontinuierliche HTTP-Übertragung. Der Modus wird standardmäßig anhand des Inhaltstyps der Antwort automatisch bestimmt; bei Bedarf wird er explizit festgelegt. Bei einer adaptiven Playlist wird standardmäßig die erste Variante der Master-Playlist verwendet; die Nummer der Variante lässt sich explizit angeben (erweiterte Einstellung). Unterstützt werden die Autorisierung mit Login und Passwort, Cookies und ein eigener User-Agent sowie Playlists, die auf die Playlist einer neuen Sitzung umleiten. Der Eingang steht nur SPTS-Streams zur Verfügung; für MPTS verwenden Sie UDP / RTP / TCP, eine Datei oder einen DVB-Adapter.

Dateien und Geräte

Der Typ file dient sowohl als Eingang als auch als Ausgang:

- **Ausgang** — Aufzeichnung in eine TS-Datei (beispielsweise für die spätere Diagnose mit externen Analysatoren) oder Ausgabe an ein Gerät: jedes Gerät in `/dev`, einschließlich SDI- und ASI-Karten. Für ein Gerät wird das Merkmal Is device node aktiviert.
- **Eingang** — zyklische Wiedergabe aus einer TS-Datei oder Aufnahme von einem Gerät (beispielsweise DVB-ASI-Empfang — *DVB-Empfänger*).

Der Pfad wird im Feld File path angegeben — der vollständige Pfad zur Datei oder zum Gerät.

Benannte Pipes (pipe)

Lesen aus einer bestehenden benannten Pipe (FIFO) und Schreiben in diese. Im Unterschied zum Typ std startet der Knoten keinen Kindprozess: Das externe Quell- oder Empfängerprogramm muss unabhängig gestartet werden und die Pipe auf seiner Seite geöffnet halten. Das Feld FIFO path (empty = auto) legt den Pfad zu einer bestehenden FIFO fest (die beispielsweise mit dem Befehl `mkfifo` erzeugt wird); bei leerem Wert erzeugt der Knoten die FIFO selbst und vergibt den Pfad automatisch (der Pfad ist in der Statistik des Eingangs bzw. Ausgangs sichtbar). Verfügbar für SPTS und MPTS.

Externe Anwendungen (std)

Der Typ std ist eine Brücke zu Protokollen, die von den integrierten Mitteln nicht unterstützt werden: Der MPEG-TS-Stream wird über die Standard-Ein-/Ausgabeströme einer Konsolenanwendung eines Drittanbieters empfangen und übertragen; diese Anwendung startet und überwacht der Knoten selbst (und startet sie bei Beendigung neu).

Vertrag der Anwendung:

- **Eingang** — die Anwendung schreibt MPEG-TS nach stdout; Diagnosemeldungen sind ausschließlich nach stderr auszugeben — stdout ist dem Stream vorbehalten;
- **Ausgang** — die Anwendung liest MPEG-TS von stdin; die Einstellung Packets per write legt die Bündelung der Schreibvorgänge fest (1–16 TS-Pakete pro Vorgang, erweiterte Einstellung).

Einstellungen: Command (absolute path) — der absolute Pfad zur Anwendung; Arguments — die Befehlszeile (Anführungszeichen erhalten Leerzeichen innerhalb eines Arguments, ein umgekehrter Schrägstrich maskiert Zeichen); Env names und Env values — Paare von Umgebungsvariablen, die an die Anwendung übergeben werden.

1.7.12 EPG/EIT

Der Knoten führt eine EPG-Datenbank — den Programmführer, der aus den empfangenen Streams und aus externen XMLTV-Quellen zusammengestellt wird. Die Daten der Datenbank werden von mehreren Subsystemen genutzt: Erzeugung von EIT-Tabellen in SPTS-Streams (*EIT-Generator*), Wiedergabe des DVR-Archivs anhand des Programmführers (*Archivwiedergabe (VOD)*) und Bereitstellung des Programmführers an externe Systeme — Set-Top-Boxen und OTT-Middleware (*EPG für OTT-Middleware*).

EPG/XMLTV-Import

Die EPG-Datenbank wird aus drei Arten von Quellen befüllt:

- **EIT aus empfangenen Streams** — die Stream-Einstellung „EIT in die EPG-Datenbank extrahieren“ (SPTS und MPTS, *Modifikation des Streams*);
- **EIT von DVB-Adaptern** — die Adaptereinstellung „EPG-Export“ (*DVB-Empfänger*);
- **externe XMLTV-Quellen** — über einen Link auf eine Webressource oder aus einer lokalen Datei.

Das aus Streams und von DVB-Adaptern extrahierte EIT wird in der gemeinsamen integrierten Quelle „Stream-Import“ gesammelt; jede externe XMLTV-Quelle führt eine eigene Datenbank. Die Kanalkennung ist innerhalb ihrer eigenen Quelle eindeutig; Kanalnamen und Ereignistexte werden in mehreren Sprachen gespeichert.

Konfiguration der Quellen

Die Quellen werden auf dem Bildschirm „EPG-Einstellungen“ (*EPG-Einstellungen*) konfiguriert. Die Registerkarte „Import“ enthält die allgemeinen Parameter: „Import aktivieren“, „Verlauf aufbewahren, Tage“ (Aufbewahrungstiefe für vergangene Ereignisse) und „Automatische Bereinigung verwaister Kanäle“ (Entfernen von Kanälen, zu denen keine Ereignisse mehr vorhanden sind).

Externe XMLTV-Quellen werden auf der Registerkarte „Quellen“ hinzugefügt: Name, „URL / Pfad der Quelle“, „Aktiviert“, „Aktualisierungsintervall, s“ (standardmäßig einmal pro Stunde), „Inhaltskodierung“ (automatische Erkennung oder erzwungenes GZip), Zugangsdaten und Cookies für zugangsbeschränkte Webquellen, „Heruntergeladene Kopie speichern“ für die Diagnose.

EPG-Datenbank

Die gesammelte Programmvorschau wird auf dem Bildschirm „EPG-Datenbank“ (*EPG-Datenbank*) eingesehen. Auf der Registerkarte „Datenbank“ wird die Quelle ausgewählt; Kanalsuche und ein Filter nach Kanalsatz stehen zur Verfügung; der Programmplan eines Kanals wird tageweise durchgeblättert, die laufende Sendung ist hervorgehoben. Für jeden Kanal lässt sich Folgendes ändern:

- „Kanalname“ — der Name für den XMLTV-Export;
- „Zeitzone“ — falls die Ereigniszeiten beim Import nicht an UTC gebunden waren;
- „Symbol-URL“ — das Kanalsymbol;
- „Kanalsätze“ — Zugehörigkeit zu den Sätzen für die Auslieferung an externe Abnehmer (*EPG für OTT-Middleware*).

Manuelle Änderungen können bei der nächsten Aktualisierung der Quelle überschrieben werden — die Weboberfläche weist darauf hin.

Die Registerkarte „Status der Quellen“ zeigt den Verlauf des Imports: Zeitpunkt der letzten und der nächsten Aktualisierung, Anzahl der Kanäle und Ereignisse, Fehler; die Schaltfläche „Jetzt aktualisieren“ startet die Aktualisierung der Quelle sofort, ohne den Zeitplan abzuwarten.

EIT-Generator

Der EIT-Generator fügt in den SPTS-Stream die Tabellen des Programmführers aus der EPG-Datenbank des Knotens ein — sowohl die laufende/folgende Sendung als auch den Sendeplan. Damit lässt sich ein vollwertiges DVB-Signal für Streams erzeugen, deren Quelle keine EIT führt: Der Programmführer wird aus einer beliebigen Quelle der EPG-Datenbank übernommen (*EPG/XMLTV-Import*).

Die Erzeugung wird im Abschnitt „EIT aus EPG erzeugen“ des Reiters „Stream“ im Stream-Editor aktiviert: Es werden „EPG-Quelle“ und „EPG-Kanal“ ausgewählt (*Modifikation des Streams*). Führt die Quelle keine SDT, so wird zusammen mit der EIT automatisch auch die Dienstbeschreibungstabelle erzeugt. Der Zustand des Generators wird im Statistikfenster des Streams angezeigt — Abschnitt „EIT-Erzeugung“: Anzahl der Ereignisse, Zeitpunkt der letzten Aktualisierung und Fehler der EPG-Quelle.

Besonderheiten:

- Das Einfügen der Tabellen verändert die Paketzusammensetzung des Streams: Aktivieren Sie für die Konformität mit TR 101 290 am Ausgang den CBR-Modus (*Bitratensteuerung*).
- Führt die Quelle bereits eine eigene EIT, so kann diese mit dem Filter „EIT löschen“ entfernt und durch die erzeugte ersetzt werden (*MPEG-TS-Filterung*).

- Beim Aufbau eines Multiplexes überträgt der Multiplexer die EIT der beteiligten Programme — einschließlich der erzeugten; die Kennungen der Tabellen werden an den Plan des Multiplexes angeglichen (*Erzeugung der Tabellen*).
- Der vom Generator eingefügte Programmführer wird auch für die Wiedergabe des DVR-Archivs anhand eines Ereignisses genutzt (*Archivwiedergabe (VOD)*).

EPG für OTT-Middleware

Der Sendeplan aus der EPG-Datenbank wird externen Systemen — Set-Top-Boxen, Anwendungen und OTT-Middleware — auf zwei Wegen bereitgestellt: als vollständiges XMLTV-Dokument über den dedizierten EPG-Server und als Tagesprogramm eines Kanals im JSON-Format über die HTTP-API des Knotens.

XMLTV-Auslieferung

XMLTV wird von einem separaten integrierten EPG-Server ausgeliefert. Er wird im Bildschirm „EPG-Server“ (*EPG-Server*) aktiviert: „EPG-Server aktivieren“, HTTP-Port (Standardwert 43973), bei Bedarf HTTPS mit Zertifikat (Standardport 43983) und „Hostname“.

Der Zugriff erfolgt über Benutzerkonten (Registerkarte „Benutzerkonten“ des Bildschirms „EPG-Einstellungen“, *EPG-Einstellungen*): Login und Passwort, „Kanalsatz“, „Ablaufdatum“ der Gültigkeit, „Zugelassene IP“ sowie das Kennzeichen „Angehalten“. Der Umfang der Auslieferung wird durch den Kanalsatz bestimmt:

- Sätze werden auf der Registerkarte „Kanalsätze“ angelegt;
- Kanäle werden in der EPG-Datenbank (*EPG-Datenbank*) in Sätze aufgenommen; ein Kanal kann mehreren Sätzen angehören;
- einem Benutzerkonto wird genau ein Satz zugewiesen — ohne ihn bleibt die Auslieferung leer.

Das Dokument ist unter dem Pfad `/xmltv` verfügbar (sowie `/xmltv.gz` — die komprimierte Datei); die Autorisierung erfolgt per HTTP Digest oder über Anfrageparameter. Die Sendezeiten werden mit der Zeitzone des Kanals ausgegeben; in das Dokument gelangen nur laufende und künftige Ereignisse. Wurde derselbe Kanalbezeichner aus verschiedenen Quellen empfangen, wird der Kanal nur einmal ausgegeben; über die Auslassung informiert das Protokoll.

Tagesprogramm im JSON-Format

Für Middleware-Schnittstellen liefert der Knoten das Tagesprogramm eines Kanals mit einer einzigen Anfrage an die HTTP-API im JSON-Format (Autorisierung wie bei den übrigen API-Anfragen):

```
GET /data/epg/channel?src=<Quelle>&ch=<Kanal>&lang=<Sprache>&t=<Zeit>
```

Zurückgegeben wird die Liste der Ereignisse des Tages (nach UTC), in den der Zeitpunkt `t` fällt: Anfang, Ende, Titel und Beschreibung jeder Sendung in der angeforderten Sprache; fehlt eine Übersetzung, wird die Standardsprache verwendet, anschließend eine beliebige verfügbare. Ein leerer Tag ist eine gültige leere Liste. Die Antworten werden vom Server zwischengespeichert und beim Eintreffen neuer EPG-Daten automatisch aktualisiert.

Der integrierte Cache ist auf etwa tausend gleichzeitige Clients ausgelegt; bei größeren Installationen wird empfohlen, sowohl die XMLTV- als auch die JSON-Auslieferung hinter einem zwischenspeichernden Reverse-Proxy oder einem CDN zu betreiben.

Die Wiedergabe des DVR-Archivs anhand eines Sendeplan-Ereignisses (Parameter `epg` in der Wiedergabe-URL) ist in *Archivwiedergabe (VOD)* beschrieben.

1.8 Weboberfläche

Die Weboberfläche ist das wichtigste Mittel zur Verwaltung und Überwachung eines Perfect Streamer-Knotens. Dies ist der Abschnitt der kontextsensitiven Hilfe: Seine Struktur entspricht der Struktur der Weboberfläche, und jeder Bildschirm sowie jedes wichtige Dialogfenster verfügt über einen eigenen Abschnitt, auf den die Schaltfläche zum Aufruf der Hilfe führt.

Der Abschnitt beschreibt den Zweck jedes Bildschirms und seinen Bezug zur übrigen Dokumentation. Die Bedeutung einzelner Felder und die Regeln zu ihrem Ausfüllen werden unmittelbar in der Oberfläche angegeben (Kurzhinweise an den Feldern), während die Konzepte hinter den Einstellungen in den verlinkten thematischen Abschnitten des Handbuchs behandelt werden.

1.8.1 Zugriff und Rollen

Die Weboberfläche wird im Browser über die Adresse des Knotens und den Port des Webservers geöffnet: standardmäßig `http://<Adresse>:8808` (oder `https://<Adresse>:43981` bei aktiviertem TLS). Es sind zwei Oberflächen verfügbar: die neue (Haupt-)Oberfläche und die klassische unter dem Pfad `/classic`. Die Erstanmeldung, die Ports der Dienste und die Änderung des Standardpassworts sind im Abschnitt *Anfangseinstellungen* beschrieben.

Die Anmeldung erfolgt mit Login und Passwort (Digest-Authentifizierung). Die Rechte werden durch die Rolle des Kontos bestimmt:

Admin
: Vollzugriff: Konfiguration von Streams und Diensten, Verwaltung, Anzeige.

Restricted admin
: Anzeige und Pausieren — eines Streams, eines einzelnen Eingangs oder Ausgangs davon und eines DVB-Adapters.

Viewer
: Nur Anzeige des Zustands: Eine Pausenschaltfläche gibt es bei dieser Rolle überhaupt nicht.

Die Rolle bestimmt auch den Umfang der Bildschirme. Nur dem Administrator stehen „Benutzer / Logins“ und „EPG-Einstellungen“, die gesamte Gruppe „Verwaltung“ sowie im „Monitoring“ die Punkte „Logs“ und „Alarme“ zur Verfügung; die übrigen Bildschirme öffnen sich in allen drei Rollen. Die Einschränkung wirkt nicht nur im Menü: Bei einem Direktlink auf einen gesperrten Bildschirm erscheint die Seite „Kein Zugriff“. Innerhalb der allgemein zugänglichen Bildschirme sind die Aktionen, die die Konfiguration ändern, nicht deaktiviert, sondern werden schlicht nicht angezeigt — eine Rolle ohne Rechte sieht sie nicht.

Den Alarm-Feed gibt der Knoten nur an den Administrator aus. Deshalb bleiben in den beiden anderen Rollen alle Alarmanzeigen der Oberfläche leer: die Alarmmarkierungen in den Stream-Zeilen, die Punkte an den Graphen und Karten, die Zustandsfarbe nach Alarm. Die Beobachtung funktioniert dabei weiterhin — geschlossen ist lediglich der Alarmkanal (*Alarme*).

Konten und Rollen werden auf dem Bildschirm *Webserver & Konten* verwaltet.

Anmelden

Das Anmeldeformular wird vor dem Laden der Anwendung angezeigt und fragt Login und Passwort ab. Nach erfolgreicher Anmeldung wird der zuletzt angezeigte Bildschirm geöffnet. Die Änderung des Standardpassworts ist bei der ersten Anmeldung verpflichtend (siehe *Anfangseinstellungen*).

1.8.2 Aufbau der Oberfläche

Links befindet sich das nach Abschnitten gruppierte Navigationsmenü, oben die Leiste mit den allgemeinen Elementen. Das Menü arbeitet in zwei Bereichen, die in der oberen Leiste umgeschaltet werden:

- **Knoten** — Verwaltung und Überwachung des aktuellen Knotens. Drei Menügruppen:
  - *Monitoring* — Zustand der Streams, Clients, Hardware und Dienste;
  - *Konfiguration* — Benutzer und EPG;
  - *Verwaltung* — Serverdienste, Speicher, Lizenz, Wartung.
- **Meshwork** — konsolidierte Bildschirme des Knotennetzes: Karte, Verbindungen und gemeinsamer Ressourcenkatalog (siehe *Meshwork* und Abschnitt *Meshwork*).

Die Zusammensetzung des Menüs hängt von der Rolle und davon ab, was auf dem Knoten vorhanden ist. Ein Eintrag, für den keine Rechte bestehen, wird nicht angezeigt. „DVB-Adapter“ und „Hardware“ erscheinen anhand der erkannten Geräte, nicht anhand der konfigurierten. Die übrigen bedingten Einträge hängen nicht von der Hardware ab, sondern von den Objekten selbst: „DVR“ setzt mindestens einen Speicher voraus, „Transcoder“ mindestens eine Transcoder-Instanz, „Test-Streams“ einen laufenden Testeingang.

Mit Listen wird in der gesamten Oberfläche gleich gearbeitet. In den Konfigurationslisten — Ein- und Ausgänge eines Streams, Logins, Billing-Server, Quellen, Kanalsätze und EPG-Konten, DVR-Speicher, Konten des Webservers — öffnet ein Klick an beliebiger Stelle der Zeile deren Editor, ebenso wie die Schaltfläche „Bearbeiten“ am Zeilenende; ein Klick auf ein verschachteltes Element (Pause, Löschen, Bibliotheksschaltfläche) führt nur dessen Aktion aus. In den Beobachtungslisten klappt ein Klick auf die Zeile die Details auf, und die aufgeklappte Zeile ist durch Hervorhebung und einen Streifen am linken Rand gekennzeichnet. Die Bedienung über die Tastatur ist in *Bedienung über die Tastatur* beschrieben.

Allgemeine Leiste

In der oberen Leiste stehen neben dem Bereichsumschalter zur Verfügung: die Uptime-Anzeige, die Alarmglocke (nur bei der Administrator-Rolle; führt zum Bildschirm *Alarme*), die Liste der Tastenkürzel, die Schaltfläche zum Aufruf der Hilfe, die Sprachauswahl, der Umschalter für das Öffnen benachbarter Knoten, der Umschalter des Designs und die Abmeldung. Der Umschalter für das Öffnen von Knoten zeigt seinen aktuellen Wert an — „Neuer Tab“ oder „Aktueller Tab“ (Kurzhinweis „Knoten öffnen in“) — und wirkt überall dort, wo die Oberfläche auf den Admin-Bereich eines anderen Knotens führt.

1.8.3 Bedienung über die Tastatur

Die Oberfläche lässt sich ohne Maus bedienen. Der erste Druck auf Tab bringt auf jedem Bildschirm die Schaltfläche „Zum Inhalt“ hervor: Sie verlagert den Fokus in den Bildschirmbereich und überspringt dabei die obere Leiste und das Menü. Das Element unter dem Fokus ist stets von einem Rahmen umgeben; bei der Arbeit mit der Maus erscheint der Rahmen nicht.

Vier Tastenkombinationen wirken auf jedem Bildschirm:

Tasten	Aktion
?	Das Fenster „Tastenkürzel“ öffnen.
/	Den Cursor in das Suchfeld des Bildschirms setzen. Das Feld gibt es auf den Bildschirmen „Streams“, „Clients“, „EPG-Datenbank“, „Logs“ und „Pipeline“; auf den übrigen bleibt die Taste dem Browser überlassen, und dessen eigene Seitensuche arbeitet.
Alt+N	In das Navigationsmenü wechseln, auf den aktuellen Punkt.
Alt+M	Zum Inhalt des Bildschirms wechseln.

Solange der Cursor in einem Eingabefeld steht und solange ein Dialogfenster geöffnet ist, wirken diese Kombinationen nicht: Die Zeichen werden eingegeben, und die Tastatur gehört dem Fenster. Alt+N und Alt+M werden anhand der Tastenposition erkannt und funktionieren daher auch auf Belegungen, bei denen Alt+N ein diakritisches Zeichen erzeugt.

Eine Datenliste ist ein einziger Tab-Halt: Man geht einmal hinein und bewegt sich darin weiter mit den Pfeiltasten. Die Tasten ↑ und ↓ wechseln zwischen den Zeilen, Home und End — zur ersten und zur letzten, Enter oder die Leertaste öffnen oder klappen eine Zeile auf, → und ← klappen sie auf und zu. Tab innerhalb der Liste läuft die Schaltflächen der aktuellen Zeile ab und verlässt danach die Liste vollständig; die Cursorposition wird gemerkt, auch nach einem Mausklick. Bloßes Bewegen des Cursors öffnet nichts: Für die Aktionen in der Kopfzeile wird eine Zeile mit Enter ausgewählt.

Wo die Reihenfolge der Zeilen per Ziehen festgelegt wird, erfolgt dasselbe Verschieben über die Kombinationen Alt+↑ und Alt+↓ — in der Liste „Streams“, in der Liste „Billing-Server“ (*Benutzer / Logins*) und im Fenster „Programme“ (*Programme*). Die neue Reihenfolge wird sofort gespeichert; bei einer Ablehnung kehrt die Zeile an ihren Platz zurück. Anders als das Ziehen steht das Verschieben per Tastatur auch auf einem schmalen Bildschirm zur Verfügung.

Auf der Zeichenfläche eines Graphen (*Pipeline*, *Topologie*, *Node-Peers*) laufen die Pfeiltasten nicht durch Zeilen, sondern verschieben das Bild; + und – ändern den Maßstab, 0 passt den gesamten Graphen ein.

In jedem Dialogfenster schließt Esc das Fenster, Tab läuft im Kreis durch dessen Elemente und führt nicht aus dem Fenster heraus; nach dem Schließen kehrt der Fokus auf das Element zurück, von dem aus das Fenster geöffnet wurde.

Das Fenster „Tastenkürzel“ wird mit der Taste ? und über die Schaltfläche in der allgemeinen Leiste geöffnet. Auf einem schmalen Bildschirm gibt es die Schaltfläche nicht, die Kombinationen selbst funktionieren jedoch: Bei angeschlossener Tastatur wird das Fenster über die Taste aufgerufen. Es führt die Kombinationen in vier Gruppen auf — „Überall“, „In der Liste“, „In einer verschiebbaren Zeile“ und „Im Dialog“ — und erinnert daran, dass unter macOS statt Alt die Taste ⌥ Option verwendet wird. Die Tastenbezeichnungen im Fenster werden nicht übersetzt.

1.8.4 Funktionsweise der kontextsensitiven Hilfe

Die Hilfe wird aus der Oberfläche aufgerufen und öffnet den Abschnitt, der dem aktuellen Kontext entspricht:

- für einen **Bildschirm** wird die Hilfe über die allgemeine Schaltfläche zum Aufruf der Hilfe geöffnet und führt zum Abschnitt dieses Bildschirms;
- für ein **Dialogfenster** wird die Hilfe über die Schaltfläche „?“ im Fenster selbst aufgerufen und führt zum Abschnitt dieses Fensters.

Jeder Bildschirm und jedes wichtige Fenster besitzt einen eigenen Hilfeabschnitt, daher führt der Übergang aus der Oberfläche stets zur Beschreibung genau dessen, was auf dem Bildschirm geöffnet ist.

Monitoring

Die Gruppe **Monitoring** im Bereich „Knoten“ fasst die Beobachtungsansichten des aktuellen Knotens zusammen: Zustand der Streams, Clients, Hardware und Dienste. Die Ansichten werden automatisch aktualisiert. Die meisten von ihnen zeigen ausschließlich Daten an, ein Teil erlaubt jedoch auch die Verwaltung des Objekts: „Streams“ — Streams anlegen und konfigurieren, „DVB-Adapter“ — einen Adapter hinzufügen, konfigurieren und scannen, „EPG-Datenbank“ — einen Kanal bearbeiten. Die Konfiguration der Dienste, der Konten und der Speicher ist in die Gruppen *Konfiguration* und *Verwaltung* ausgelagert.

Die zentrale Ansicht der Gruppe — „Streams“ — wird in einem eigenen Abschnitt *Streams* behandelt.

Knotenübersicht

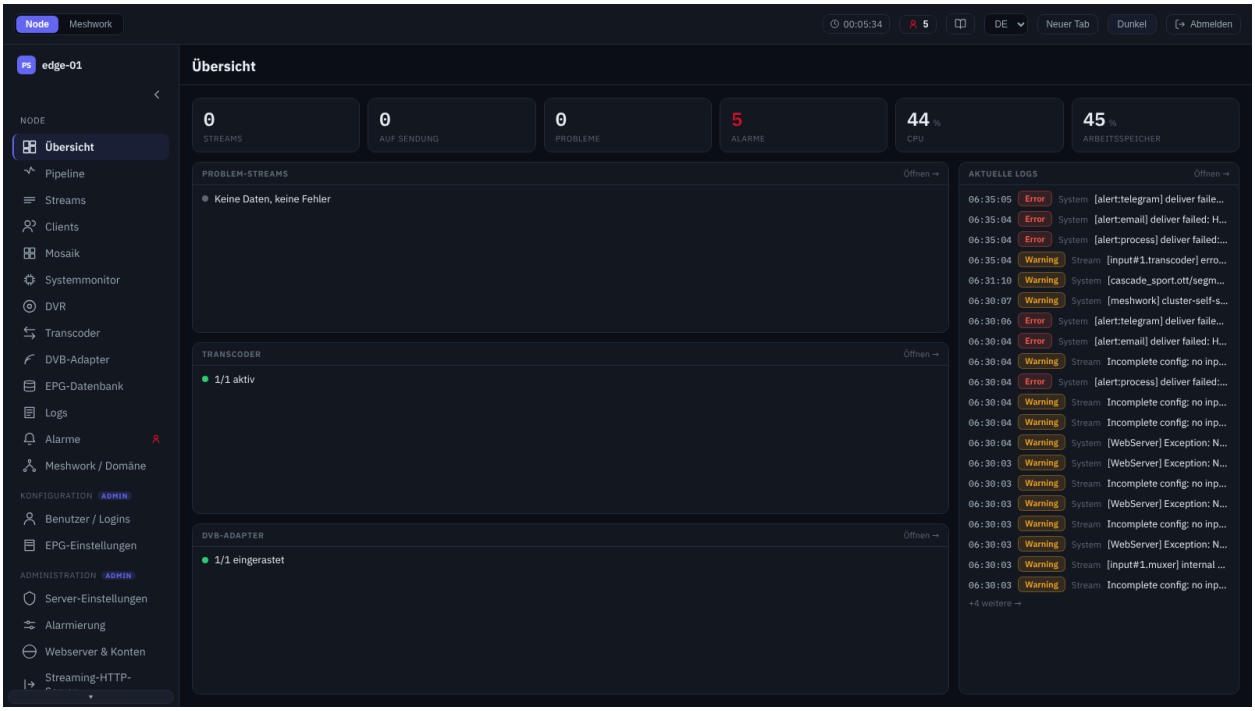


Abb. 5: Knotenübersicht.

Das Start-Dashboard des Knotens. Die obere Leiste enthält die Kacheln der Kennwerte: Anzahl der Streams und davon „Auf Sendung“, Zähler „Probleme“, aktive „Alarme“ (in der Administratorrolle), CPU-

und Speicherauslastung. Darunter folgen die Panels der auffälligen Teilsysteme: „Problematische Streams“, „Transcoder“, „DVB-Adapter“ und (für den Administrator) „Neueste Logs“; jedes davon führt die fehlerhaften Objekte oder eine kurze Zusammenfassung „normal“ auf; aus der Überschrift des Panels führt der Übergang zur zugehörigen Ansicht. Ganz unten liegt die Leiste „Favoriten“ mit den angehefteten Signaldiagrammen der DVB-Adapter und Mosaik-Kacheln (das Anheften erfolgt auf den jeweiligen Ansichten). Die Bewertung der Knotenauslastung im Detail — auf der Ansicht *System-Monitor*.

Pipeline

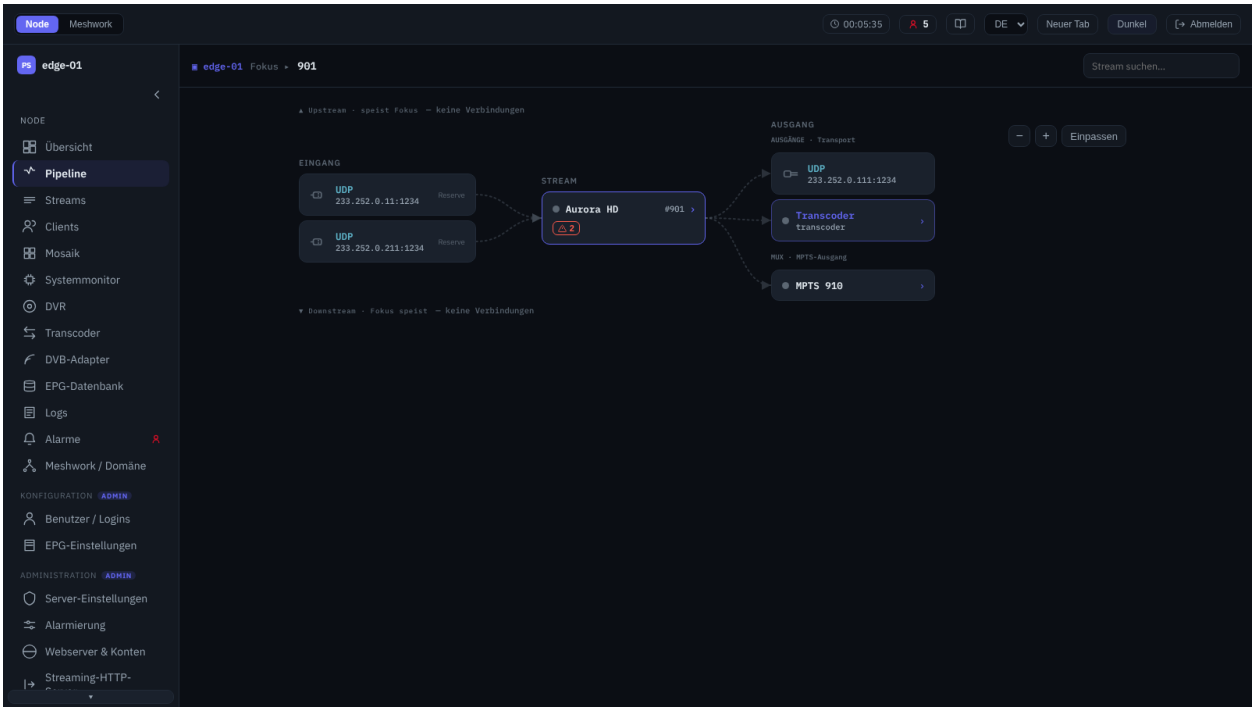


Abb. 6: Pipeline-Graph: Einzelstream (SPTS).

Der Verarbeitungsgraph einer einzelnen ausgewählten Entität: wie Quellen, Stream und Ausgänge zu einer durchgehenden Pipeline verbunden sind (EINGANG → STREAM → AUSGANG). Die Fokusansicht richtet sich nach dem Typ: SPTS- oder MPTS-Stream (mit Anzeige der eingehenden und ausgehenden Programme des Multiplexes; Liste der Streams — *Streams*), *Adaptive Bundles*, Transcoder (*Transcoder*) oder DVB-Adapter (*DVB-Adapter*). Zwischen den Entitäten wird über die Nachbarpfeile, über die Namenssuche oder aus anderen Ansichten gewechselt; der aktuelle Fokus wird in der Adresse gehalten (ein Direktlink kann weitergegeben werden). Ein eigener Block fasst die lokalen Weiterleitungen des Knotens (Peer) zusammen, und die „Türen“ zu den Nachbarknoten öffnen deren Admin-Oberfläche im selben Fokus. Die Karten des Transcoders — sowohl die des Eingangs als auch die des Ausgangs — führen in den Graphen dieses Transcoders; beim Eingang muss dafür ein Decoder festgelegt sein. Die Karten der Ein- und Ausgänge in den Transporten des gemeinsamen Domain-Katalogs verfügen über die Schaltfläche „In der Bibliothek anzeigen“ (*Bibliothek — dieser Feed*). Der Graph zeigt nur aktive Streams und IO; pausierte Ein- und Ausgänge werden darin nicht dargestellt. Das Übertragungsmodell ist in *Übertragungsmodell: Stream, Sender, Receiver, Peer* beschrieben, die Verarbeitung eines Streams auf dem Knoten — in *Streamer: Implementierungsdetails*, MPTS-Multiplexe — in *MPTS-Streams*.

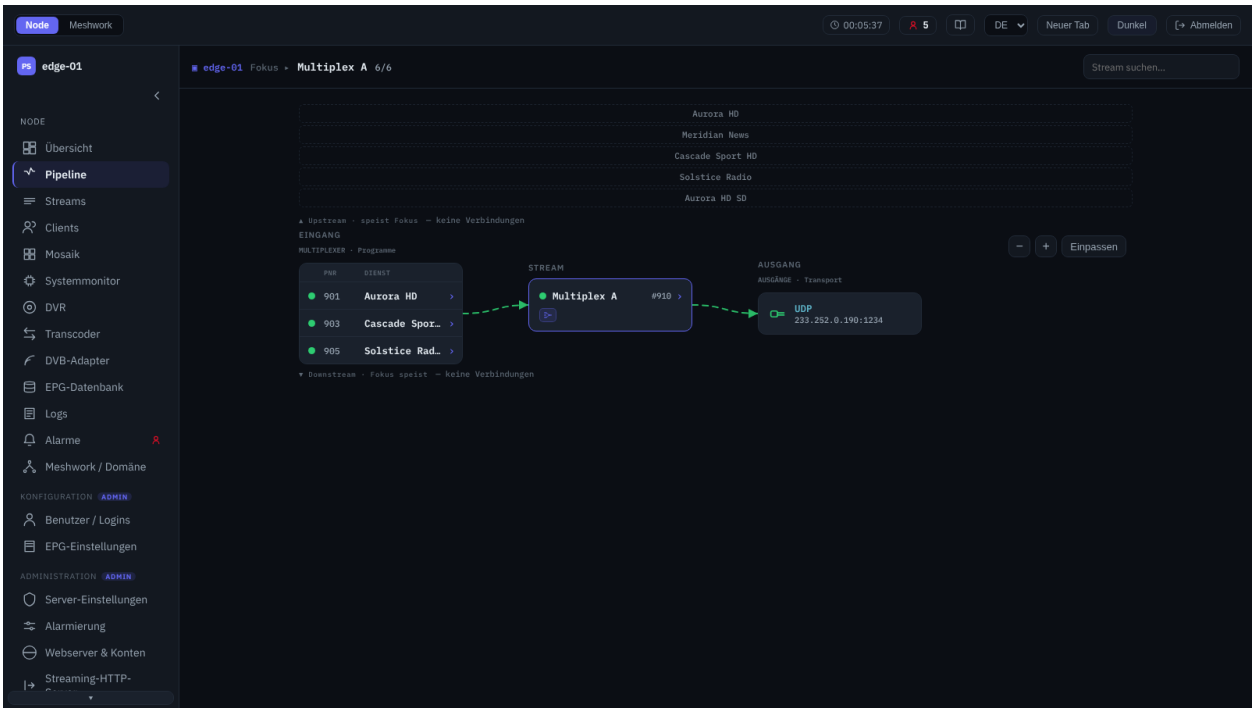


Abb. 7: Pipeline-Graph: MPTS-Multiplexer.

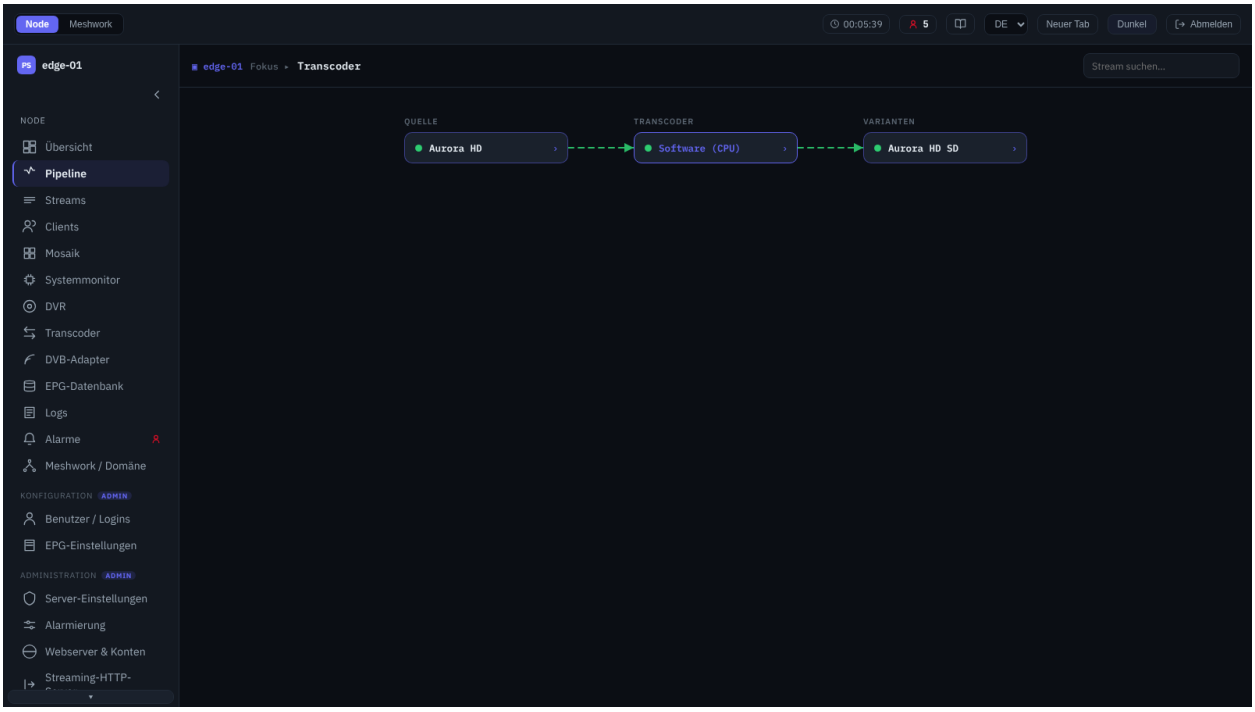


Abb. 8: Pipeline-Graph: Transcoder.

Clients

Liste der verbundenen Empfänger, aufgeteilt in Registerkarten nach Auslieferungsprotokoll: „HTTP / OTT“, „SRT“, „PS1“ und „RIST“; jede Registerkarte trägt einen Live-Zähler der aktiven Sitzungen. Angezeigt werden die betrieblich wichtigen Spalten: Client (mit Herkunftskennzeichen — „Lokaler Login“ oder „Externes Billing“), Adresse, Stream, Login, für SRT — Modus (caller/listener), Verluste und RTT, für OTT — Sitzungstyp und „Qualität“ der Auslieferung, Verbindungszeitpunkt. Über die Liste arbeitet die Suche (Name, Login, Adresse), lange Listen werden auf Seiten aufgeteilt. Das Aufklappen einer Zeile öffnet den vollständigen Satz der Diagnosezähler der Sitzung, der in Echtzeit aktualisiert wird. Dem Administrator steht die Aktion „Trennen“ zur Verfügung — sie beendet alle Sitzungen des ausgewählten Logins. Die Übertragungsprotokolle sind in *Peer-Protokolle für zuverlässige Übertragung* beschrieben, die OTT-Auslieferung — in *OTT und DVR*, die Konten der Empfänger — auf der Ansicht *Konfiguration* (*2. Verhalten der Clients*).

Mosaik

Eine Vorschauwand: Live-Miniaturbilder der Einzelbilder — je eine Kachel für jeden aktiven SPTS-Stream und für jedes Programm eines aktiven MPTS (die Programme des Multiplexes sind mit einem Symbol gekennzeichnet). Das Raster passt sich der Bildschirmbreite an, die Kachelgröße wird über das Element „Größe“ (1, 1/2, 1/3) umgeschaltet. Pausierte und gestoppte Streams erscheinen nicht im Mosaik. Ein Klick auf ein Miniaturbild öffnet das vergrößerte Bild; mit dem Stern lässt sich die Kachel in den „Favoriten“ auf der Ansicht *Knotenübersicht* anheften.

Originalbild

Fenster zur vergrößerten Ansicht des ausgewählten Einzelbildes in der Originalauflösung — für die detaillierte visuelle Prüfung des Bildes. Innerhalb des Fensters kann durch die benachbarten Mosaik-Kacheln geblättert werden.

System-Monitor

Zustand des Servers in Form von Karten: „CPU“ (Gesamtauslastung und Anteil des PSS-Prozesses, Anzahl der Kerne), „Speicher“ (belegt und Anteil von PSS, Umfang in GB), „Netzwerk“ (je Schnittstelle — Empfang/Senden und Kanalauslastung, mit Einbettung der VLANs unter den physischen Adapter) sowie Karten der Hardwarebeschleunigung (NVIDIA — GPU/Speicher/Decoder/Encoder, Intel VPL). Darunter folgen die Diagramme der Kennwerte im Zeitverlauf. Auf Knoten mit DVB-Tunern werden in einem eigenen Abschnitt die Diagramme der Signalqualität der Frontends ausgegeben (*Signalüberwachung*). Hilft, die Lastreserve einzuschätzen; Empfehlungen zur Abstimmung der Ressourcen — in *Optimierung der Knotenleistung*, die externe Erfassung von Metriken — in *Anbindung externer Überwachungssysteme*.

DVR-Monitor

Beobachtung der DVR-Aufzeichnung nach Speichern (nur Anzeige). Für jeden Speicher — Füllstand relativ zum Bereinigungsziel (Balken mit Zielmarke), Volumina (belegt, frei, gesamt, Archivgröße aufgeschlüsselt nach TS/MP4), Kennzeichen „Speicherplatzdruck“, aktuelle Hintergrundaufgabe und Anzahl der gebundenen Streams; die aktiven Alarme des Speichers werden daneben ausgegeben. Das Aufklappen zeigt zeilenweise die Streams des Archivs (Aufzeichnung läuft / Leerlauf, Vorhaltetiefe, angesammeltes Volumen, Lese-/Schreiblatenzen) und die Wartungsdiagnose (Einsammeln verwaister Verzeichnisse, Kürzung bei Speicherplatzdruck). Für einen Stream öffnet sich das Fenster *Stream-Statistik*; dem Administrator stehen die Bereinigung des Archivs eines einzelnen Streams sowie der allgemeine Befehl „Verwaiste Archive

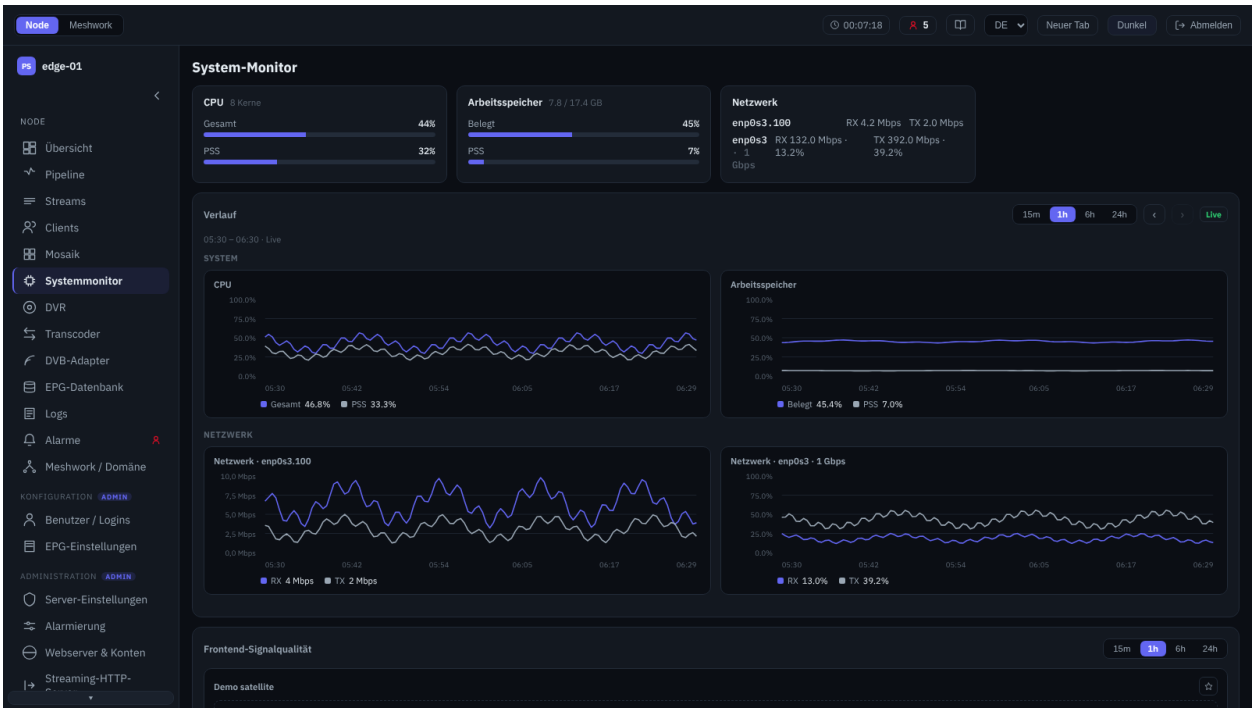


Abb. 9: System-Monitor.

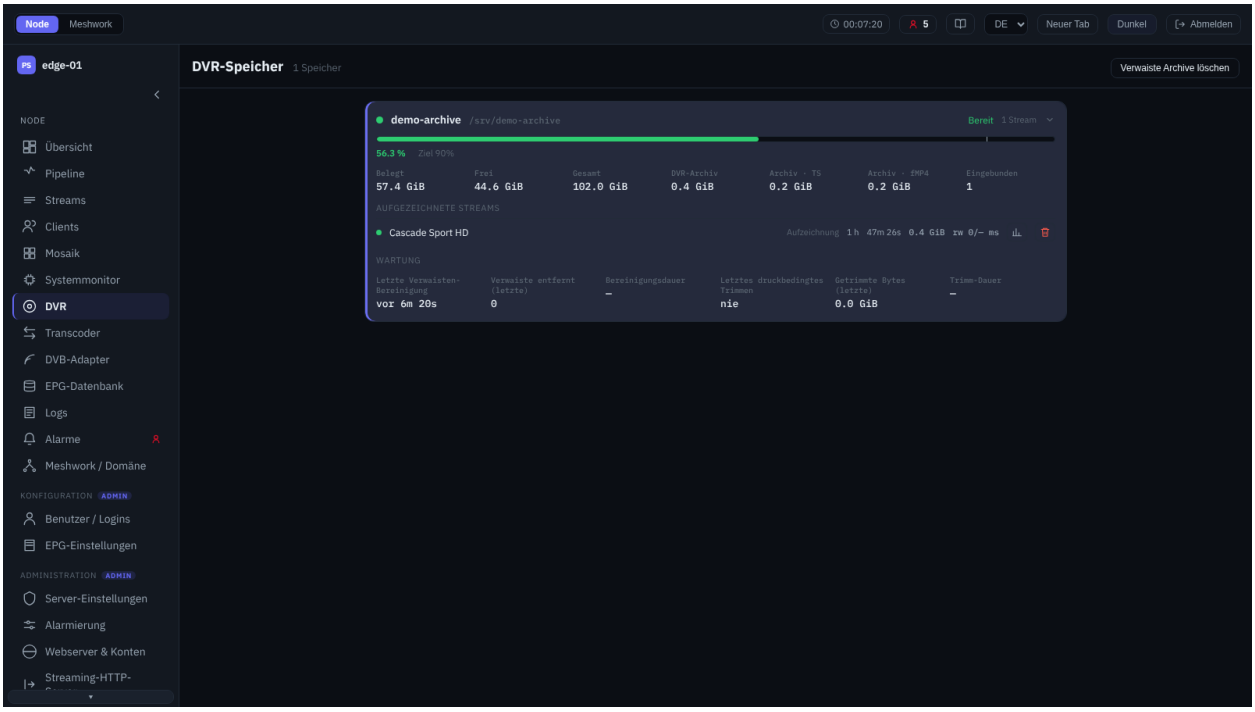


Abb. 10: Monitor der DVR-Speicher.

bereinigen“ zur Verfügung. Die Konfiguration der Speicher selbst — auf der Ansicht *DVR-Speicher*; Aufzeichnung und Wiedergabe des Archivs sind in *OTT und DVR* und *DVR: Archiv des Streams* beschrieben.

Transcoder

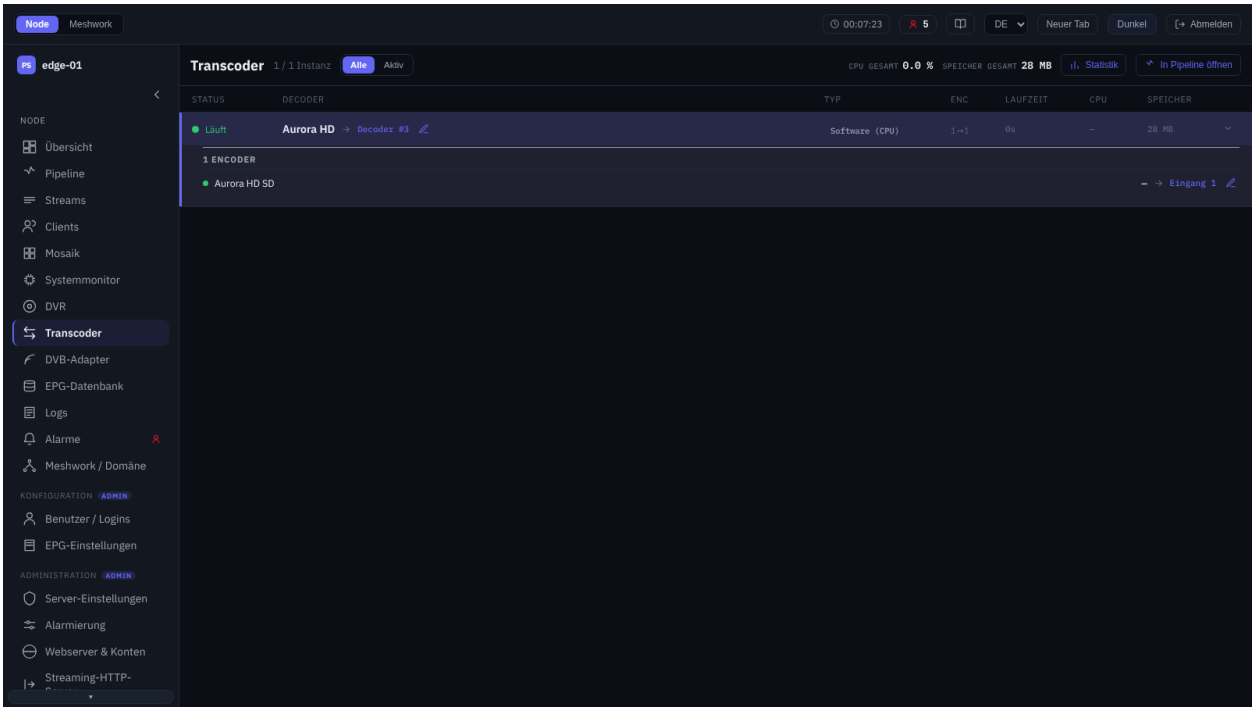


Abb. 11: Bildschirm „Transcoder“.

Liste der aktiven Transcoder-Instanzen des Knotens mit dem Filter „Alle / Aktive“. In der Kopfzeile — die summarische Auslastung „CPU gesamt“ und „Speicher gesamt“ sowie die Aktionen für die ausgewählte Instanz: „Statistik“ und „In der Pipeline öffnen“ (*Pipeline*). Je Zeile — Zustand, Decoder (Quelle), Typ, Anzahl der Encoder, Laufzeit, CPU und Speicher; das Aufklappen zeigt die Encoder mit den Parametern des Ausgangs (Videocodec, Auflösung, Bildrate/Abtastverfahren, Audiocodecs, tatsächliche Bitrate). Die Marken der Pipeline öffnen die Statistik des zugehörigen Streams. Dem Administrator steht beim Decoder und bei jedem Encoder eine Konfigurationsschaltfläche zur Verfügung: sie öffnet den Editor des zugehörigen Streams über der Liste (*Stream konfigurieren*). Möglichkeiten und Konfiguration der Transcodierung — in *Transcoder*, die Beobachtung — in *Betriebsüberwachung*, die Installation von Paketen und Treibern — in *Transcoder*.

Transcoder-Instanz

Die Karte eines einzelnen Transcoders — ein Statistikfenster, das aus der Kopfzeile der Ansicht oder aus dem zentralen Knoten des Transcoder-Graphen geöffnet wird. Sie zeigt die Quelle (Decoder), die Liste der Encoder (1 → N), die Ressourcen (CPU, Laufzeit, PID, Speicher) und das Log des Prozesses: letzter Exit-Code, aktuelles und vorheriges Log der Starts (mit Kopierfunktion) — für die Diagnose. Beim Aufruf über einen Link klappt das Fenster zugleich die Zeile dieser Instanz in der dahinterliegenden Liste auf und hebt sie hervor, sodass der Bediener nach dem Schließen des Fensters auf ihr bleibt; ist die Zeile durch den Modus „Aktive“ ausgeblendet, wird die Liste nicht zu ihr gescrollt.

DVB-Adapter

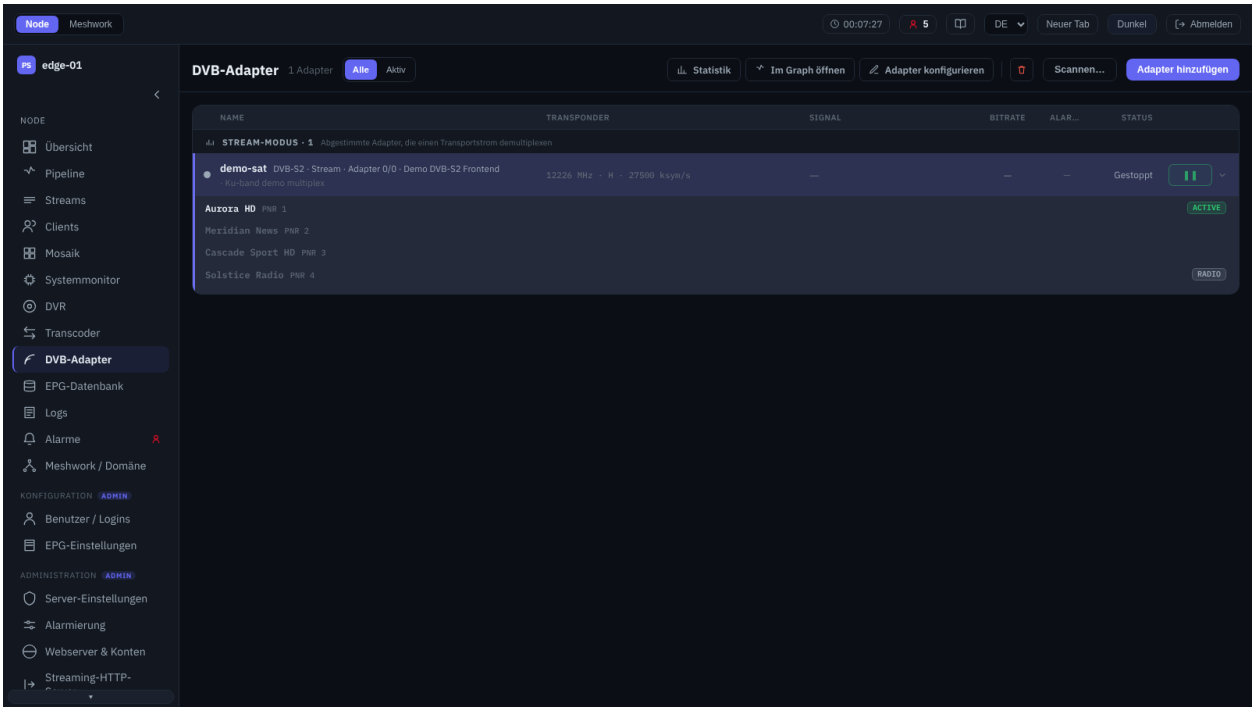


Abb. 12: Bildschirm „DVB-Adapter“.

Eine einheitliche Ansicht des DVB-Empfangs: die Konfiguration der Adapter, angereichert um den Live-Zustand (Signalerfassung, Qualität, Bitrate). Die Zeilen sind nach Modus gruppiert — „Modus Stream“ (Empfang in einen Stream) und „FE Monitor“ (Überwachung eines von einer anderen Anwendung abgestimmten Frontends); in den Spalten — Name, Transponder, Signal, Bitrate, Alarme und Zustand; in der Zeile — ein Umschalter für die Pause. Die Aktionen für den ausgewählten Adapter sind in der Kopfzeile untergebracht: „Statistik“, „Im Graph öffnen“ (*Pipeline*) sowie (für den Administrator) „Adapter konfigurieren“ und „Adapter löschen“. Ebendort — der Filter „Alle / Aktive“, „Adapter hinzufügen“ und „Scannen…“ (für den Administrator). Der DVB-Empfang ist in *DVB-Empfänger* beschrieben, die Beobachtung der Adapter — in *Signalüberwachung*.

DVB-Adapter-Editor

Fenster zum Hinzufügen und Konfigurieren eines DVB-Adapters. „Modus“ und „Übertragungssystem“ werden beim Hinzufügen festgelegt und sind danach unveränderlich — sie bestimmen den Umfang der Einstellungen. Die Hardware wird aus der Liste der erkannten Frontends ausgewählt (siehe *Hardware*). Für den Modus Stream stehen die Abstimmparameter zur Verfügung (Frequenz, Symbolrate, FEC, Modulation und weitere — je nach Übertragungssystem), der Abschnitt LNB für den Satellitenempfang, „T2-MI“, „BISS-Entschlüsselung“ und „CI/CAM-Entschlüsselung“. Das Stummschalten der Alarme des Adapters und der Abschnitt „Erweitert“ (erweiterte Parameter) sind in beiden Modi verfügbar; der Modus FE Monitor liest das Signal, ohne den Tuner umzustimmen. Empfang und Descrambling sind in *Adapter*, *Entschlüsselung* und *T2-MI-Dekapselung* beschrieben.

DVB-Scan

Assistent für die Kanalsuche auf einem freien Tuner (für den Administrator). Er läuft in Schritten ab: Auswahl von Tuner und Übertragungssystem, Auswahl der Quelle — „Katalog“ (Satellit, Kabel oder terrestrisch mit LNB-Parametern) oder „Blindscan“ über einen Frequenzbereich; anschließend werden die Transponder durchlaufen, mit Fortschrittsanzeige und Sammlung der gefundenen Multiplexe, zu denen jeweils die Liste der Dienste gehört. Gefundene Multiplexe lassen sich einzeln hinzufügen (dabei öffnet sich das Konfigurationsfenster des Adapters mit vorbelegten Parametern) oder mehrere auswählen und auf einmal hinzufügen; die Namen werden automatisch gebildet und auf Übereinstimmungen geprüft. Die Kanalsuche ist in *Scan* beschrieben.

Adapter-Statistik

Fenster mit den Live-Parametern des ausgewählten Adapters. Es enthält: die Zustandsübersicht und die Abstimmparameter, die aktiven Alarme, die Verlaufsdiagramme des Signals (Pegel, SNR, BER, UCB), den Abschnitt „Metriken“ (Frontend, Erfassungsstatus, Frequenz, SNR, Signalpegel, Fehlerzähler, für den Modus Stream — Bitrate), die Liste der Programme, „T2-MI-Träger“, „BISS-Entschlüssler“, „CI/CAM-Modul“, „Profiler“ (Auslastung des Verarbeitungsthreads, Überlauf des DVR-Rings) und das Log des Adapters. Dem Administrator steht „Statistik zurücksetzen“ zur Verfügung. Dient der Ausrichtung der Antenne und der Kontrolle der Empfangsstabilität (*Signalüberwachung*).

Test-Streams

Die Ansicht der Test-Streams — Erzeugung von Dienst-Streams zur Prüfung der Signalwege (*Teststreams*). Der Abschnitt befindet sich in Vorbereitung.

EPG-Datenbank

Die gesammelten Programmdaten und der Zustand ihrer Quellen. Die Ansicht enthält die Registerkarten:

Datenbank
: Ansicht der angesammelten Programmdaten: Auswahl der Quelle, Liste der Kanäle mit Suche und Filter nach Sets, Programmplan des ausgewählten Kanals nach Tagen (Umschalten von Tag und Sprache, Hervorhebung der laufenden Sendung). Dem Administrator steht die Bearbeitung des Kanals zur Verfügung — Name, Zeitzone, Symbol, Zugehörigkeit zu Sets.

Zustand der Quellen
: Importstatus je Quelle: Zustand, Zeitpunkt der letzten und der nächsten Aktualisierung, Anzahl der Kanäle und Ereignisse, Fehler; dem Administrator steht die erzwungene Aktualisierung einer Quelle zur Verfügung.

Import und Verarbeitung des EPG sind in *EPG-Datenbank* und *EPG/XMLTV-Import* beschrieben; die Quellen und ihre Konfiguration — in *Konfiguration der Quellen* und auf der Ansicht *EPG-Einstellungen*.

Logs

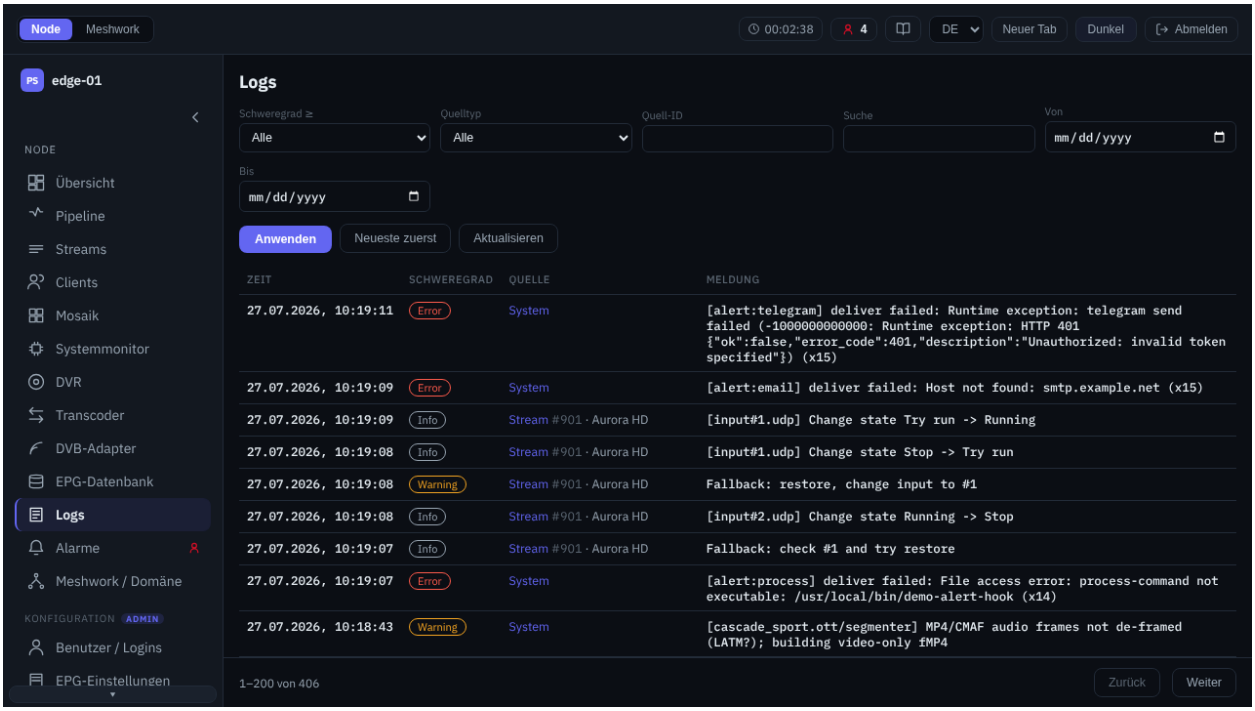


Abb. 13: Logs.

Strukturiertes Ereignis-Log des Knotens (für den Administrator verfügbar) mit Filtern und seitenweiser Ausgabe. Die Auswahl erfolgt nach Schweregrad („Schweregrad ≥“), nach Typ und Kennung der Quelle, nach Meldungstext und nach Datumsbereich; die Sortierreihenfolge ist umschaltbar (neueste / älteste zuerst). In den Spalten — Zeit, Schweregrad, Quelle und Meldung; eine Quelle vom Typ Stream öffnet ihr Fenster *Stream-Statistik*, andere Quellen führen auf die jeweilige Fachansicht. Dient der Diagnose und der Aufarbeitung von Störungen.

Die Quelle eines Eintrags wird durch den Objekttyp bezeichnet, bei Streams zusätzlich durch Nummer und Namen: `Stream #43 · orig CBC HD`; ist kein Anzeigename gesetzt, wird der Stream-Name eingesetzt. Mit demselben Namen samt Nummer sind die Streams in der Auswahlliste des Filters „Quell-ID“ beschriftet. Einträge, die von einem Ein- oder Ausgang eines Streams stammen, werden dem Stream selbst zugerechnet und zeigen dessen Namen; der konkrete Ein- bzw. Ausgang wird am Anfang des Meldungstextes genannt.

Alarme

Zusammenfassende Ansicht der geltenden Alarme: die eigenen (des aktuellen Knotens) oder die der gesamten Domain — umgeschaltet über den Bereich in der Kopfzeile. Die Ansicht zeigt nur den aktiven Satz — ein aufgehobener Alarm verschwindet aus der Liste. In den Spalten — Schweregrad, Quelle, Meldung und Alter; eine Quelle vom Typ Stream öffnet ihr Fenster *Stream-Statistik*. In einer Domain aus mehreren Knoten wird in der Zeile der Ursprungsknoten angezeigt (mit Übergang in dessen Admin-Oberfläche). Alarme der Stufe Administratoreingriff sind mit der Schaltfläche „Zurücksetzen“ (manuelle Bestätigung) versehen, die auf dem eigenen Knoten wirkt. Bei solchen Alarmen ist unter der Quelle angegeben, wo die Ursache zu beseitigen ist; handelt es sich um einen Eingang oder einen Ausgang, dient die Zeile dem Administrator als Link und öffnet den Editor des besitzenden Streams (*Stream konfigurieren*). Ein Alarm, der von einem anderen Knoten stammt, bietet keinen Link — die Stream-Nummern hat jeder Knoten für sich. Das Alarmmodell,

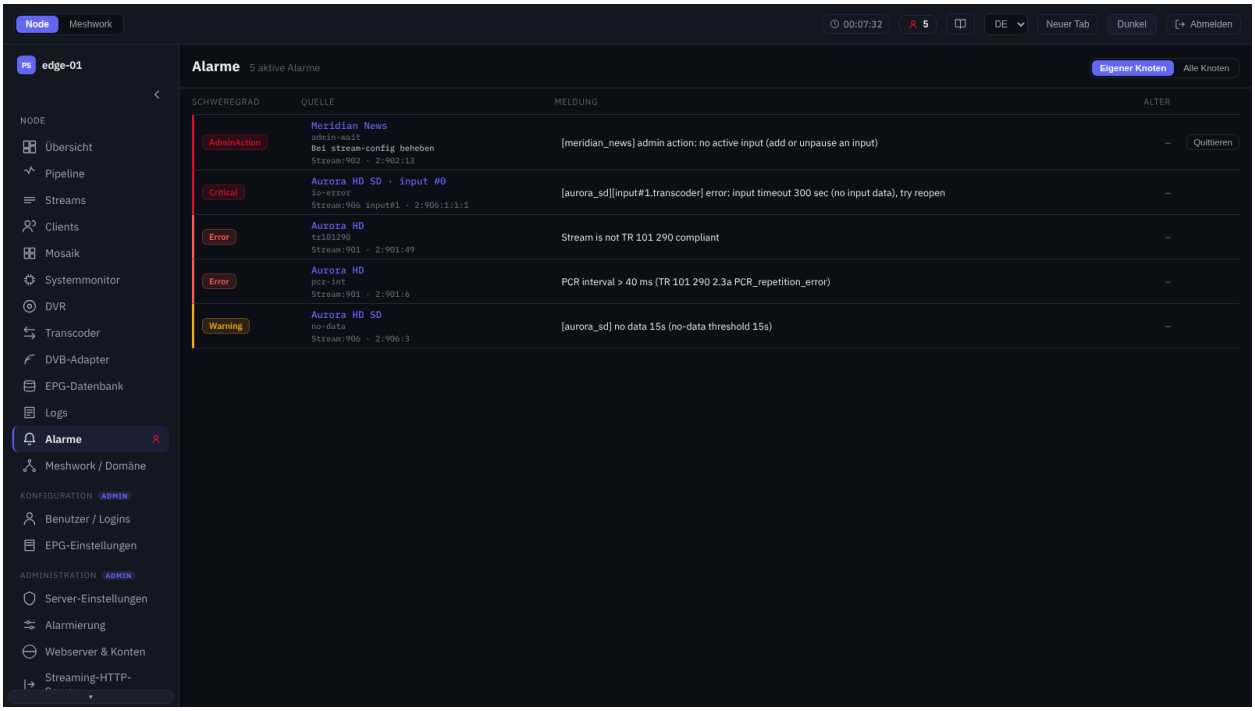


Abb. 14: Aktive Alarme.

dessen Auswertung und die vollständige Liste der Codes sind in *Benachrichtigungen (Alerter)* und *Katalog der Alert-Codes* aufgeführt; die Konfiguration der Schwellwerte und der externen Zustellung — auf der Ansicht *Alarmierung*.

Meshwork / Domain

Sicht auf das Meshwork-Netz aus der Perspektive des aktuellen Knotens. In der Kopfzeile — Name des Knotens, dessen Anmerkung, die beobachtete externe Adresse („gesehen als …“) und die verknüpften Domains; sind keine Domains vorhanden, wird „Keine verknüpften Domains“ ausgegeben.

Der Rumpf der Ansicht ist die Peer-Tabelle: die von diesem Knoten aus sichtbaren Knoten der Domain mit den Spalten „Typ“ („Konfiguriert“ oder „Replikation“), „Status“ („Aktiv“ / „Ausgefallen“), „Knoten“, „Domain“, „Adresse“, „NAT“ und „Letzter Kontakt“. Zuerst sind die konfigurierten Nachbarn aufgeführt, dann die automatisch gefundenen, innerhalb jeweils nach Knotennamen. Die Zeile des aktuellen Knotens ist mit „dieser Knoten“ gekennzeichnet, ein Super-Knoten ist gesondert markiert, Keep-alive-Fehler durch die Markierung „Beeinträchtigt“; ein Knoten hinter NAT zeigt „Internes Netzwerk“ anstelle der nicht erreichbaren privaten Adresse.

Der Pfeil am Ende der Zeile klappt die Einzelheiten des Peers auf: die öffentlichen Zugangspunkte (HTTP und HTTPS), den genauen Zeitpunkt des letzten Kontakts, die Telemetrie des Kontakts (Verzögerung, Anzahl der Fehlschläge in Folge und insgesamt, letzter Fehler), die gemeldete und die beobachtete Adresse sowie den Nonce der Instanz; von hier aus wird per Schaltfläche die Admin-Oberfläche des Nachbarknotens geöffnet. Die netzweite Sicht auf die Peers der Domain bietet *Node-Peers*; den gemeinsamen Ressourcenkatalog der Domain zeigt die eigene Ansicht *Bibliothek*.

Die Begriffe und die Übersichtsansichten des Netzes sind im Abschnitt *Meshwork* beschrieben (Karte — *Netzwerkkarte und Ressourcenkatalog*, Katalog — *Gemeinsamer Ressourcenkatalog*, Peers — *Peers und Secrets*); die Ansichten des Bereichs „Meshwork“ — in *Meshwork*.

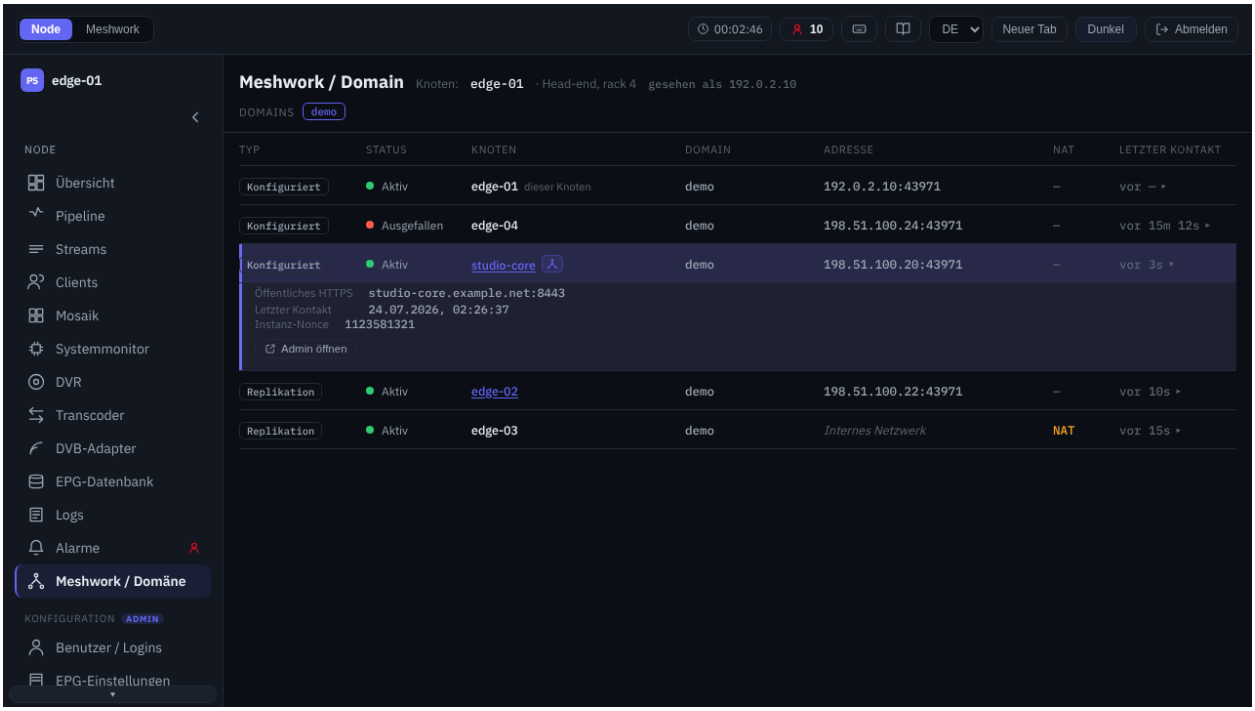


Abb. 15: Meshwork-Domäne: Node-Peers.

Streams

Der Bildschirm **Streams** ist der zentrale Arbeitsplatz des Knotens: eine einheitliche Liste der konfigurierten Streams mit ihrem Zustand, dem aktiven Eingang, der Bitrate, „Uptime“ und dem Zähler der Continuity-Fehler „CC“. Jede Zeile lässt sich per Klick aufklappen und zeigt die Eingänge und Ausgänge des Streams mit ihrem Zustand, ihrer Redundanzpriorität, Notizen und aktiven Alerts; ein einzelner Eingang oder Ausgang kann von hier aus angehalten und wieder aufgenommen werden, und die Schaltfläche „In der Bibliothek anzeigen“ neben der Adresse öffnet die Liste der übrigen Punkte der Domain an dieser Adresse (*Bibliothek — dieser Feed*). Der Begriff des Streams und das Übertragungsmodell werden in *Übertragungsmodell: Stream, Sender, Receiver, Peer* beschrieben; die SPTS-Verarbeitung — in *SPTS-Streams*, der MPTS-Multiplex — in *MPTS-Streams*.

Die Liste ist nach Typen gruppiert: einzelne Streams (SPTS) bilden eine gemeinsame Liste, darunter folgen die Gruppen „MPTS-Streams“ und „Adaptive Streams“ (ABR-Bundles). Oberhalb der Liste befinden sich die Navigations- und Steuerelemente:

Suche und Filter
: Das Feld „Stream suchen…“ sucht nach Name, Notiz und Eingangs-/Ausgangsadresse und hebt die gefundene Zeile hervor, ohne die Liste zu verkürzen. Der Filter „Tags“ schränkt die Liste anhand der Stream-Markierungen ein, der Schalter „SPTS / MPTS / Adaptiv“ blendet die Typgruppen ein oder aus, und das Segment „Alle / Aktive“ blendet angehaltene Streams aus.

Aktionen für den ausgewählten Stream
: Die Kopfzeile enthält die Aktionen, die auf den aufgeklappten (ausgewählten) Stream oder das ABR-Bundle angewendet werden: „Stream-Statistik“ (*Stream-Statistik*), „In der Pipeline öffnen“ (*Pipeline*), „Stream konfigurieren“ (*Stream konfigurieren*) und „Stream löschen“. Die Schaltfläche „+ Neuer Stream“ öffnet das Fenster zum Anlegen. Der Administrator kann die Zeilen per Drag-and-drop umsortieren — nur unter den SPTS-Streams und nur, wenn die Liste vollständig angezeigt wird: ohne Filter nach Tags, im Modus „Alle“ und mit allen aktivierten Typen. Dasselbe Verschieben erfolgt über die Kombinationen `Alt+↑` und `Alt+↓` (*Bedienung über die Tastatur*).

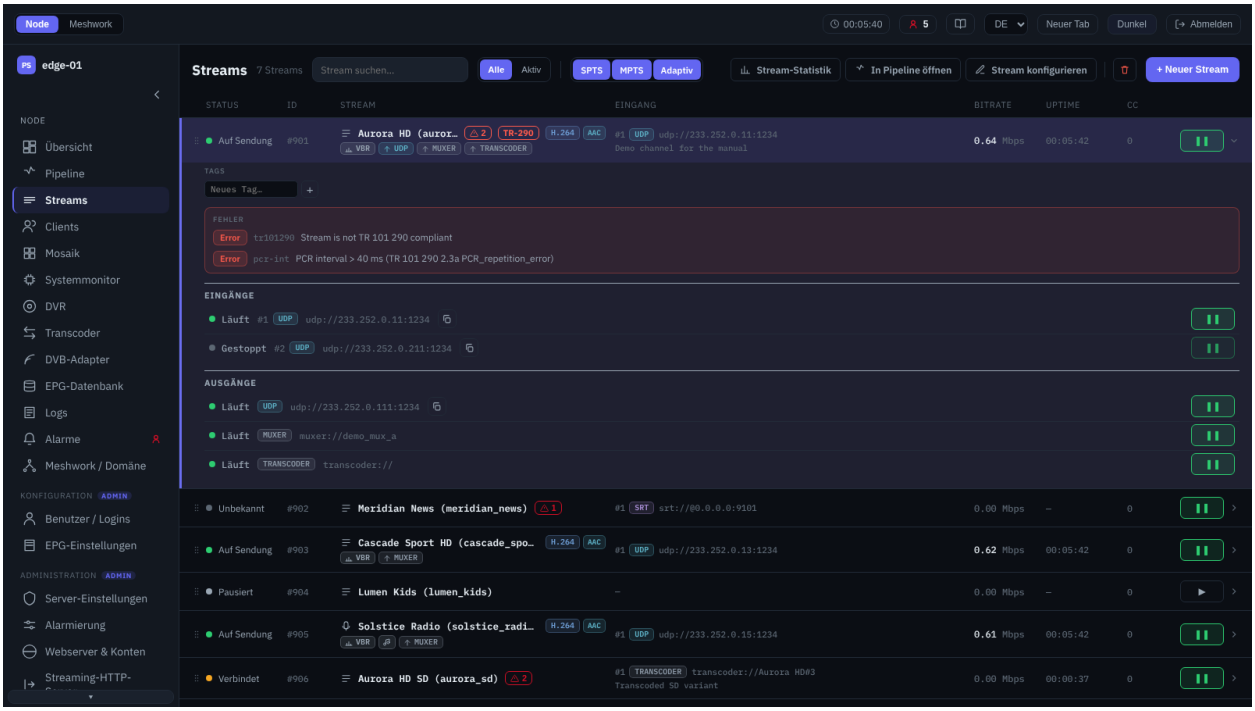


Abb. 16: Bildschirm „Streams“: Streamliste mit aufgeklappter Zeile.

Zustand und Alerts
: Eine farbige Zustandsanzeige, die Merkmals-Badges des Streams (CBR/VBR, Verwürfelung, DRM-Schutz, Betrieb am Reserve-Eingang, nur Audio oder nur Video, Codecs, TRACE) und die Markierung aktiver Alerts. Alerts sind an den Stream und an seine Eingänge/Ausgänge gebunden (*Benachrichtigungen (Alerter)*); der TR 101 290-Bericht und der Abschnitt der Tiefenanalyse werden direkt aus der Zeile geöffnet.

Die einzelnen Fenster des Bildschirms sind nachfolgend beschrieben.

Neuer Stream

Das Fenster zum Anlegen eines Streams. Der Typ wird über einen Schalter gewählt: einzelnes Programm (**SPTS**), Multiplex (**MPTS**) oder adaptives Bundle (**Adaptiv**); für SPTS wird zusätzlich der „Inhaltstyp“ festgelegt. Nach dem Anlegen eines SPTS- oder MPTS-Streams öffnet sich sofort der Editor (*Stream konfigurieren*). Für ein adaptives Bundle wird kein eigener Editor geöffnet — die Zusammensetzung der Teilnehmer und deren Bitraten werden direkt in diesem Fenster festgelegt, da das Bundle aus bereits konfigurierten OTT-Streams aufgebaut wird. Die Unterschiede zwischen SPTS und MPTS sind in *SPTS-Streams* und *MPTS-Streams* beschrieben, adaptive Bundles — in *Adaptive Bundles*.

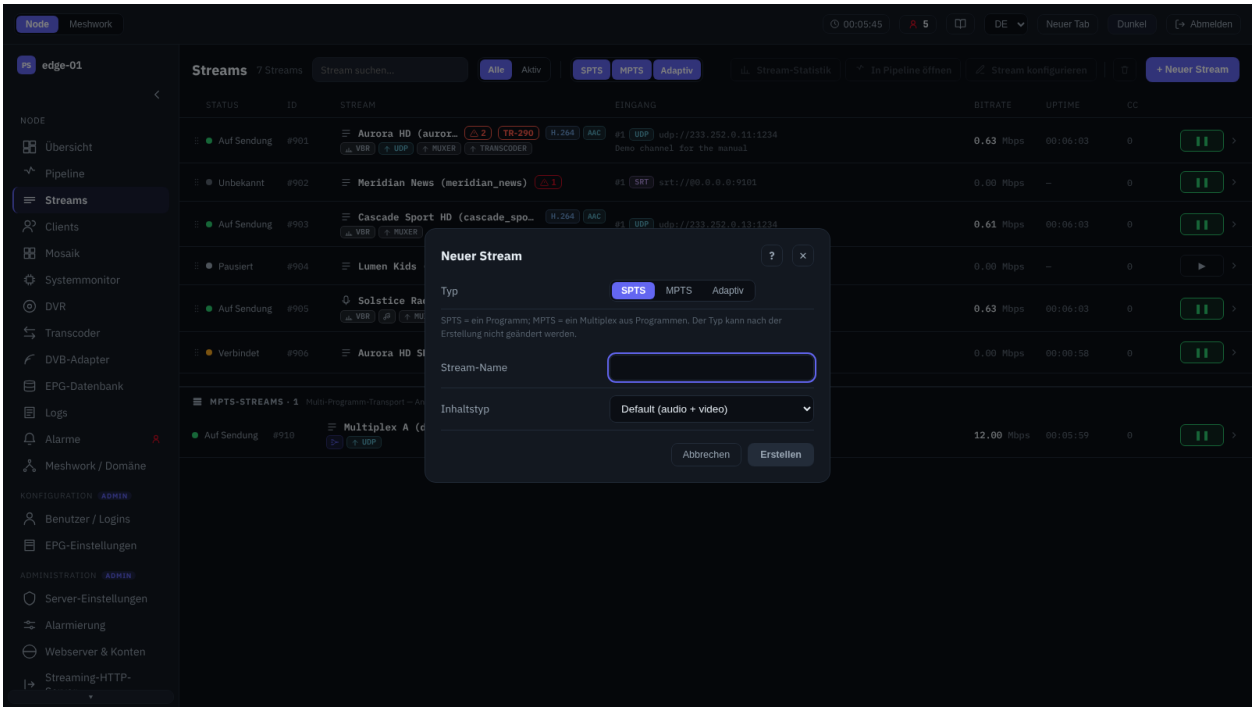


Abb. 17: Dialog „Neuer Stream“.

Stream konfigurieren

Der Haupteditor des Streams. Die Einstellungen sind auf Registerkarten gruppiert; für MPTS-Streams wird die Registerkarte „OTT“ nicht angezeigt.

Stream
: Identifikation (Name, Anzeigename, für SPTS — der Stream-Typ, Notiz) und Verhalten des Streams: Timeout, Prüfintervall, erneute Prüfung des Haupteingangs, Mosaik-Erzeugung (*Mosaik*), EIT-Extraktion in die EPG-Datenbank (*EPG-Datenbank*) und das Verwürfelungsmerkmal für SPTS. Für SPTS wird hier ebenfalls die EIT-Erzeugung aus der ausgewählten EPG-Quelle und deren Kanal konfiguriert (*EIT-Generator*).

Eingänge
: Die Liste der Stream-Quellen mit ihrer Priorität und Redundanz (*Quellenredundanz und Verteilung*). Das Hinzufügen und Ändern eines Eingangs öffnet den Editor für Eingang und Ausgang (*Editor für Eingang und Ausgang*); ein einzelner Eingang kann angehalten werden. Der Eingang `muxer` wird nur in einer leeren Liste angeboten: Bei einem MPTS-Stream wird er als erster hinzugefügt, und solange er vorhanden ist, ist die Schaltfläche „Eingang hinzufügen“ mit dem Hinweis „Der Eingang muxer muss der einzige Eingang des Streams sein.“ deaktiviert — eine Redundanz der Eingänge gibt es beim Multiplexer nicht. Ein MPTS-Stream, der einen fertigen Multiplex über das Netz empfängt, unterliegt dieser Einschränkung nicht: Seine Eingänge sind gewöhnlich und werden redundant ausgelegt.

Ausgänge
: Auslieferung des Streams über die Übertragungsprotokolle (*Peer-Protokolle für zuverlässige Übertragung*) und die Standardtransporte (*Weitere Eingänge und Ausgänge*). Jeder Ausgang wird über denselben Editor hinzugefügt und bearbeitet. Der Ausgang `ps1` kann bei einem Stream nur einmal vorkommen — beim Hinzufügen des nächsten steht dieser Typ nicht mehr in der Liste; die übrigen Typen sind wiederholbar.

MPEG-TS

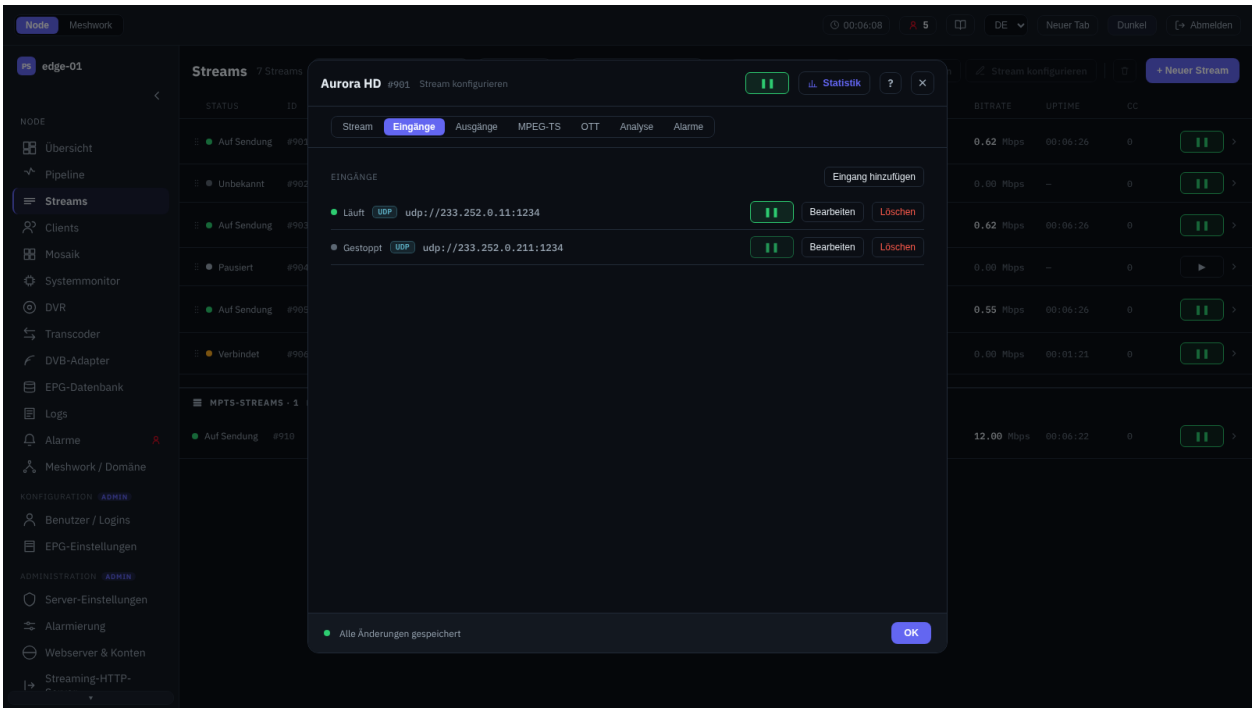


Abb. 18: Stream-Editor, Registerkarte „Eingänge“.

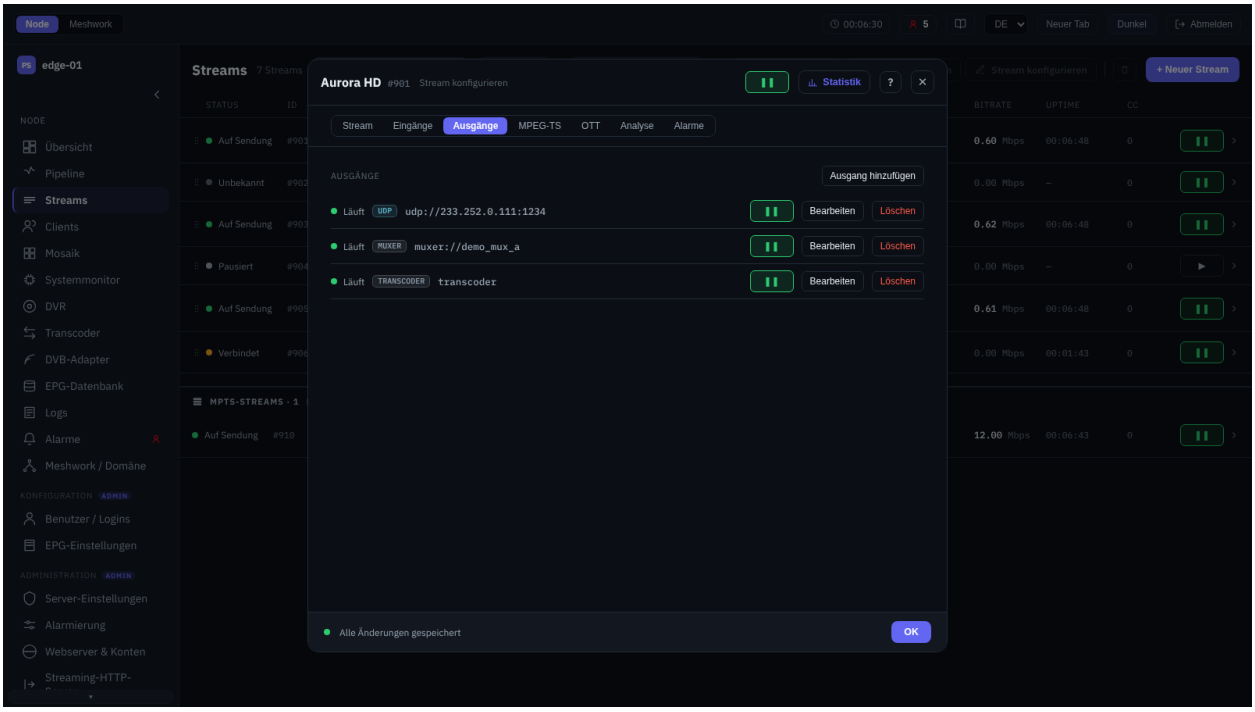


Abb. 19: Stream-Editor, Registerkarte „Ausgänge“.

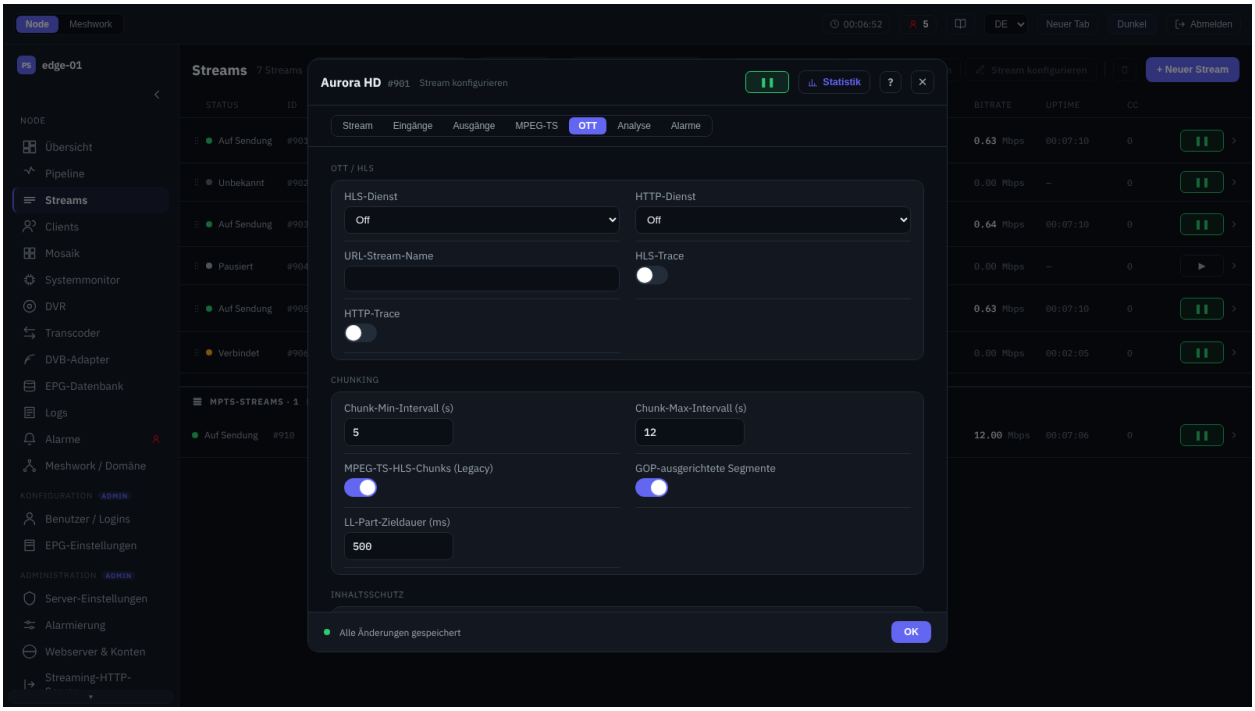


Abb. 20: Stream-Editor, Registerkarte „OTT“.

Der Bitratenmodus des ausgehenden Transportstroms (*Bitratensteuerung*) und — für SPTS — die „Netzwerk-ID“, die Fixierung des PAT/PMT-Intervalls, die Filterung der Diensttabellen (CAT/ECM/EMM, EIT, NIT, SDT, Untertitel, TDT, Teletext; *MPEG-TS-Filterung*), das Überschreiben der SDT und der Programmnummer (*Modifikation des Streams*) sowie die automatische PID-Anordnung des Programms und der Basis-PID seines Fensters (*Automatische PID-Anordnung*).

OTT
: OTT-Auslieferung des Streams: die Dienstmodi HLS und HTTP (*Auslieferungsmodi*), der Stream-Name in der URL, die Extraktion von Teletext-Untertiteln nach WebVTT, die Aufteilung in Chunks und niedrige Latenz (*Segmentierer*), die DVR-Aufzeichnung in den ausgewählten Speicher (*OTT und DVR*) und der DRM-Inhaltsschutz — HLS AES-128 oder CENC (*Inhaltsschutz (DRM)*). Nur für SPTS.

Analyse
: Qualitätskontrolle des Transportstroms: ausführliches Tracing, Tiefenanalyse der Elementarströme, TR 101 290-Monitor, Analyse von PCR/PTS-Sprüngen und des T-STD-Videopuffers, Kompensation der Synchronisationsdrift (*Analysator*).

Alarme
: Der übergreifende Schalter für die Alerts des Streams und die Verzögerung bis zum Auslösen von Alerts über Fehler seiner Eingänge und Ausgänge (*Benachrichtigungen (Alerter)*); die Schwellenwerte werden auf Knotenebene festgelegt (*Alarmierung*).

Das Speichern von Änderungen, die ein aufgezeichnetes DVR-Archiv betreffen, ist zusätzlich durch eine Bestätigung abgesichert (*DVR-Änderungen bestätigen*). Die einzelnen Fenster des Editors sind nachfolgend beschrieben.

Editor für Eingang und Ausgang

Das Fenster zum Hinzufügen und Ändern eines Eingangs oder Ausgangs eines Streams. Der Transporttyp wird beim Anlegen gewählt und ist danach unveränderlich (zum Wechsel des Transports wird der Eingang oder Ausgang neu angelegt). Die Parameter sind auf Registerkarten verteilt:

Eigenschaften
: Adresse, Port, Bindung an eine Netzwerkschnittstelle und die Verbindungsparameter des gewählten Transports. Peer-Protokolle und Standardtransporte sind in *Planung und Datenübertragungsprotokolle* beschrieben; die Verschlüsselung — in *Stream-Verschlüsselung*. Für PS1- und SRT-Eingänge steht die automatische Anmeldung über die Meshwork-Peer-Autorisierung zur Verfügung, mit der Möglichkeit, auf manuelles Login und Passwort umzuschalten. Die Adresse lässt sich aus der Zwischenablage einfügen („URL einfügen“) oder als Quelle aus dem Domain-Katalog auswählen („Aus Bibliothek wählen…“, *Bibliotheksquellen*). Für einen Transcoder-Eingang werden hier ebenfalls der Decoder und die Encoding-Parameter festgelegt, einschließlich des Modus zur Änderung der Bildgröße (*Transcoder*).

MPEG-TS PID
: Auswahl und Neuzuordnung von PIDs an einem Ein- oder Ausgang: PID-Annahme (Whitelist), PID-Ablehnung (Blacklist), Neuzuordnung von PIDs und der Sprache der Tonspuren. Solange für den Stream die automatische PID-Anordnung (*Automatische PID-Anordnung*) aktiviert ist, wird die Gruppe der PID-Neuzuordnung an den Eingängen ausgeblendet und bleibt wirkungslos; die Annahme- und Ablehnungslisten arbeiten weiter wie bisher. Am muxer-Ausgang bleibt die PID-Neuzuordnung erhalten — sie gehört zum Multiplexer (*PID-Zuweisung*) —, doch als Ausgangs-PIDs in ihren Paaren gelten nun die von der automatischen Anordnung vergebenen PIDs.

BISS
: BISS-Entwürfelungsschlüssel für eine verwürfelte Quelle — ein Schlüssel für SPTS oder je ein Schlüssel pro Programm (PNR) für MPTS (*BISS-Entschlüsselung*).

Bibliotheksquellen

Auswahl eines Eingangs aus dem gemeinsamen Ressourcenkatalog der Meshwork-Domain: eine Liste der auf den Knoten der Domain bereits ausgelieferten Streams desselben Transports, aus der sich eine Quelle anbinden lässt, ohne die Adresse manuell einzugeben. Die Liste wird nach Quellknoten und Text gefiltert und ist in Seiten aufgeteilt; Quellen derselben Domain in Peer-Protokollen sind mit dem Merkmal „Peer“ gekennzeichnet (*Gemeinsamer Ressourcenkatalog*). Nicht zu verwechseln mit dem Fenster *Bibliothek — dieser Feed*: hier wird über den Katalog eine Quelle ausgewählt, dort wird angezeigt, wer sonst noch mit einer bereits festgelegten Adresse arbeitet.

Bibliothek — dieser Feed

Das Fenster beantwortet die Frage „wer sonst noch mit dieser Adresse arbeitet“. Geöffnet wird es durch die Schaltfläche „In der Bibliothek anzeigen“ neben der Adresse eines Ein- oder Ausgangs: in der aufgeklappten Zeile der Streamliste, in den Ein- und Ausgangslisten des Stream-Editors, auf den Ein- und Ausgangskarten der Pipeline (*Pipeline*) und im Statistikfenster — beim aktiven Eingang und im Abschnitt „Ausgänge“. Der Untertitel des Fensters ist die Adresse des Punktes, von dem aus es aufgerufen wurde.

Die Schaltfläche gibt es nur bei den Transporten, die in den gemeinsamen Katalog der Domain gelangen: UDP, RTP, Pro-MPEG, RIST, PS1 und SRT (*Gemeinsamer Ressourcenkatalog*). Ein Transcoder-, Muxer-, Demuxer-, Datei- oder Testgenerator-Eingang hat keine Einträge im Katalog und daher auch keine Schaltfläche — das ist kein Anzeichen einer Störung.

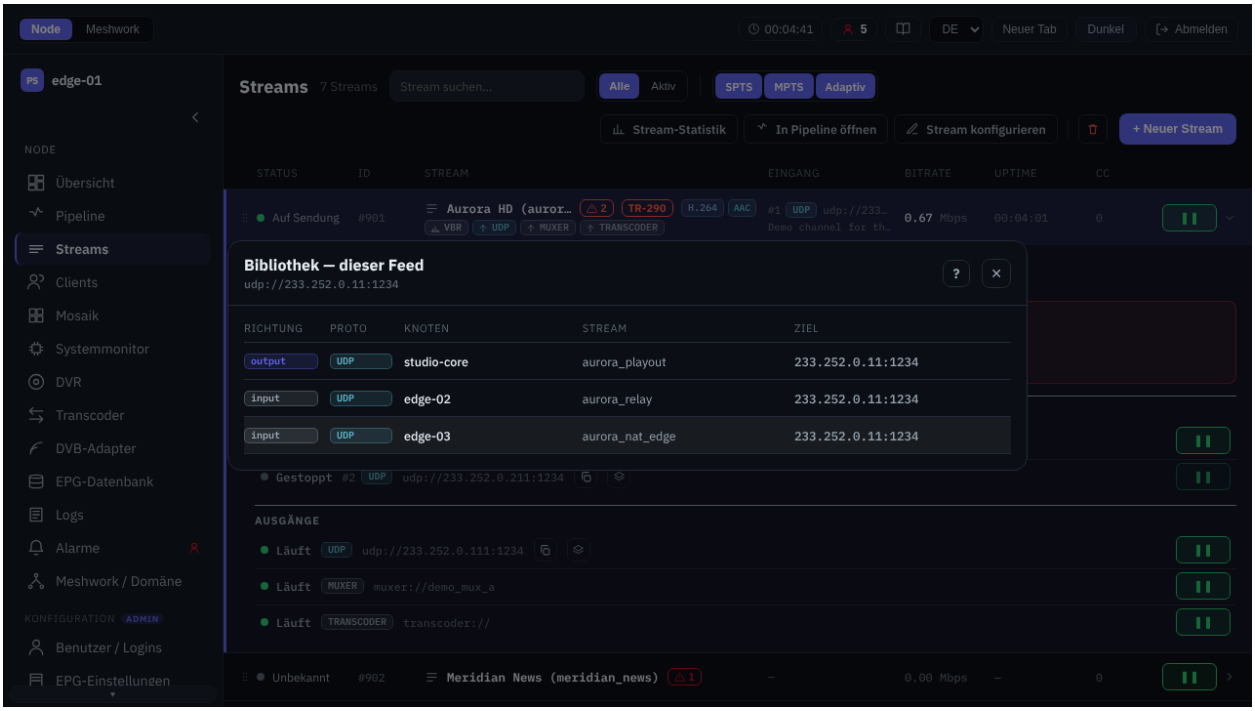


Abb. 21: Das Fenster „Bibliothek — dieser Stream“: beim Eingang eines Streams werden die Quelle (`output`) und zwei weitere Abnehmer (`input`) derselben Adresse angezeigt.

Im Fenster sind die übrigen Punkte der Domain aufgeführt, die mit derselben Adresse arbeiten. Die eigenen Ein- und Ausgänge dieses Streams gelangen nicht in die Liste, einschließlich seiner Reserveeingänge: der Bediener sieht sie ohnehin. Ein anderer Stream desselben Knotens, der auf derselben Adresse lauscht, ist dagegen ein gewöhnlicher Nachbar und wird gleichrangig mit den übrigen angezeigt.

Die Spalte „Richtung“ zeigt die Rolle des Punktes: `output` — er liefert diesen Stream aus (die Quelle), `input` — er empfängt ihn (ein Abnehmer); die Quellen sind zuerst aufgeführt. Die übrigen Spalten sind dieselben wie im gemeinsamen Katalog: „Proto“, „Knoten“, „Stream“ und „Ziel“ (Adresse und Port). Hinter der Adresse wird die Kennzeichnung „verifiziert“ ausgegeben: dieses Paar hat der Knoten durch die Domain-Autorisierung bestätigt und nicht aus übereinstimmenden Adressen abgeleitet — beim Betrieb über NAT stimmen die Adressen der Seiten ohnehin nicht überein. Die Kennzeichnung erhält höchstens eine Zeile je lauschendem Socket: der Knoten benennt einen Vertreter aus den bestätigten Sitzungen, und dieser wechselt, während Clients sich verbinden und trennen. Das Fehlen der Kennzeichnung bedeutet daher nicht, dass die Verbindung unecht ist.

Ein Klick auf eine Zeile öffnet den Admin-Bereich ihres Knotens bei der Statistik ihres Streams. Die Zeile des eigenen Knotens wird an Ort und Stelle geöffnet, die Zeile eines Nachbarn gemäß der Einstellung „Knoten öffnen in“ der oberen Leiste (*Allgemeine Leiste*); Strg oder erzwingt einen neuen Tab. Die Zeile eines Knotens, der nicht erreichbar oder dem Netz unbekannt ist, hat keine Aktion und trägt den Kurzhinweis „Knoten nicht erreichbar“.

Die Übereinstimmung wird nach Protokoll und vollständiger Adresse — Gruppe und Port — berechnet, bei UDP, RTP, Pro-MPEG und RIST zusätzlich nach VLAN und SSM-Quelle. Ein Punkt auf derselben Gruppe, aber in einem anderen VLAN oder mit einer anderen SSM-Quelle gelangt daher nicht in die Liste, ebenso wenig ein Punkt eines anderen Transports an derselben Adresse. Bei PS1 und SRT ist eine Übereinstimmung der Adressen überhaupt nicht erforderlich: die rufende Seite benennt den Partner über den Hostnamen, und dieser Name wird mit der Zusammensetzung der Domain abgeglichen — deshalb kann die Adresse einer so gefundenen Zeile ganz anders aussehen. Auf dieselbe Weise wird auch das bestätigte Paar gefunden, über die Namen von Knoten und Stream, die die lauschende Seite nennt.

Die beiden Meldungen über das Fehlen von Nachbarn bedeuten Unterschiedliches:

Meldung	Was sie bedeutet
„An dieser Adresse sind keine weiteren Punkte registriert.“	Der Punkt ist im Katalog angekündigt und hat keine Nachbarn.
„Dieser Punkt ist nicht in der Bibliothek angekündigt (angekündigt werden nur laufende Netzwerk-Ein- und -Ausgänge).“	Zu dem Punkt selbst gibt es keinen Eintrag im Katalog: der Stream läuft nicht, der Ein- oder Ausgang ist angehalten, wurde gerade erst gestartet oder ist durch das Feld Peer private von der Veröffentlichung ausgeschlossen (*Gemeinsamer Ressourcenkatalog*). Die Nachbarn fehlen in diesem Fall nicht — sie sind unbekannt.

Denselben Katalog in seiner Gesamtheit zeigt die Ansicht *Bibliothek*.

Programme

Der Programmeditor des Multiplex (MPTS), der aus dem Konfigurationsfenster des Muxer-Eingangs geöffnet wird: die Programmzusammensetzung des Ausgangsstroms, deren Reihenfolge in der PSI und die PID-Zuweisung. Ein Programm wird aus den konfigurierten SPTS-Streams hinzugefügt oder entfernt; die Reihenfolge wird per Drag-and-drop geändert, auf einem schmalen Bildschirm über die Kombinationen `Alt+↑` und `Alt+↓` (*Bedienung über die Tastatur*); ein einzelnes Programm kann angehalten werden. Ist die automatische PID-Zuweisung deaktiviert, werden für jedes Programm die PMT-PID und die PIDs der Elementarströme bearbeitet. Die Zusammenstellung des Multiplex ist in *Multiplexer* beschrieben, die Verteilung von Programmen und PIDs — in *Parameter der Programme* und *PID-Zuweisung*, Transit und Analyse von MPTS-Strömen — in *MPTS-Streams*.

DVR-Änderungen bestätigen

Eine Warnung vor dem Speichern von Stream-Änderungen, die ein bereits aufgezeichnetes DVR-Archiv betreffen: das Abschalten der Aufzeichnung, der Wechsel zu einem anderen Speicher, die Verkürzung der Aufbewahrungsdauer oder des geschützten Minimums sowie der Wechsel des DRM-Schlüssels bei einem Stream mit Aufzeichnung (zuvor aufgezeichnete verschlüsselte Segmente werden dadurch unzugänglich). Das Fenster listet auf, was genau betroffen sein wird, und verlangt eine Bestätigung; unbedenkliche Änderungen (das erstmalige Anbinden eines Speichers, eine Verlängerung der Aufbewahrungsdauer) werden ohne Rückfrage gespeichert. Aufzeichnung und Archivaufbewahrung sind in *OTT und DVR* und *DVR: Archiv des Streams* beschrieben.

Stream-Statistik

Ausführliche Statistik des Streams in einem separaten Fenster, in Echtzeit aktualisiert. Über einen Link von einem anderen Bildschirm geöffnet, zeigt das Fenster diesen Stream zugleich in der dahinterliegenden Liste: Es hebt den Filter auf, der ihn verborgen hat, klappt die Zeile auf und scrollt die Liste zu ihr. Die Abschnitte im Einzelnen:

Zustandsübersicht
: Der aktuelle Zustand und die Meldung des Streams, die Merkmals-Badges, der aktive Eingang und die Anzahl der Ausgänge, der Bitratenmodus, das Format und die Codecs des Bildes, die Links der OTT-Auslieferung und ein Live-Standbild (*Mosaik*). Ebenfalls hier — das Symbol des TR 101 290-Urteils, das den vollständigen Bericht öffnet, und die Schaltfläche „In der Bibliothek anzeigen“ beim aktiven Eingang (*Bibliothek — dieser Feed*).

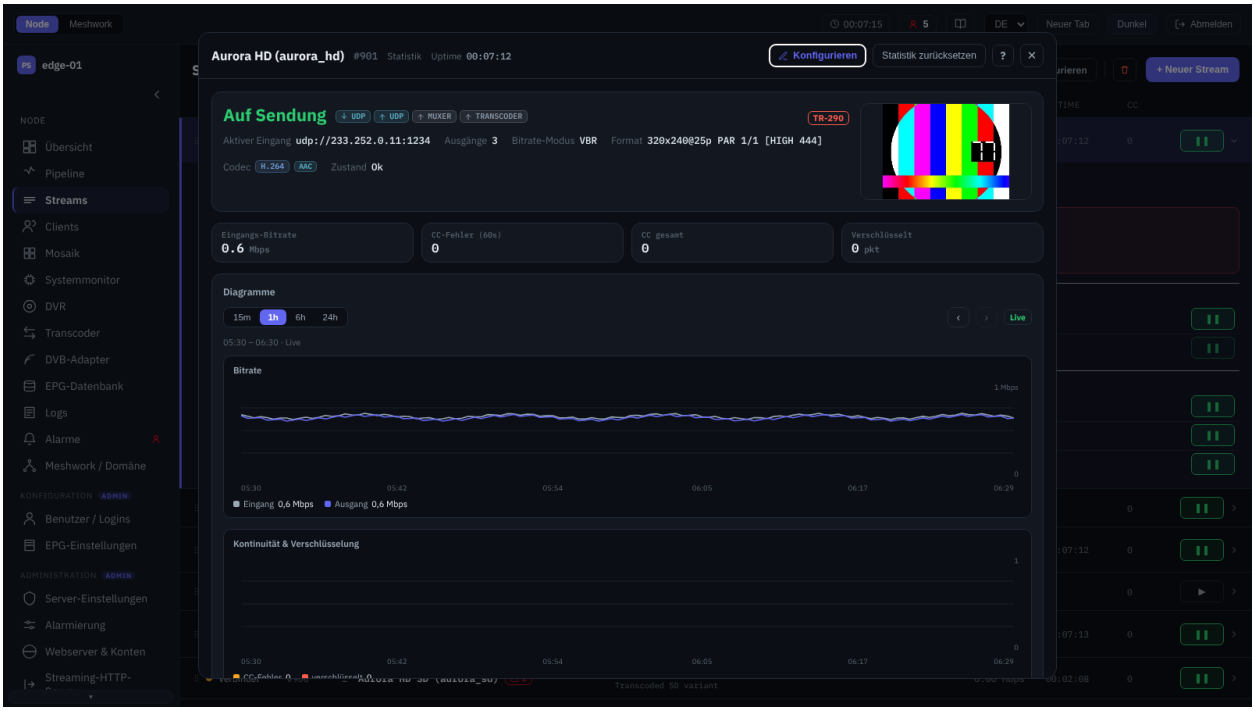


Abb. 22: Fenster der Stream-Statistik.

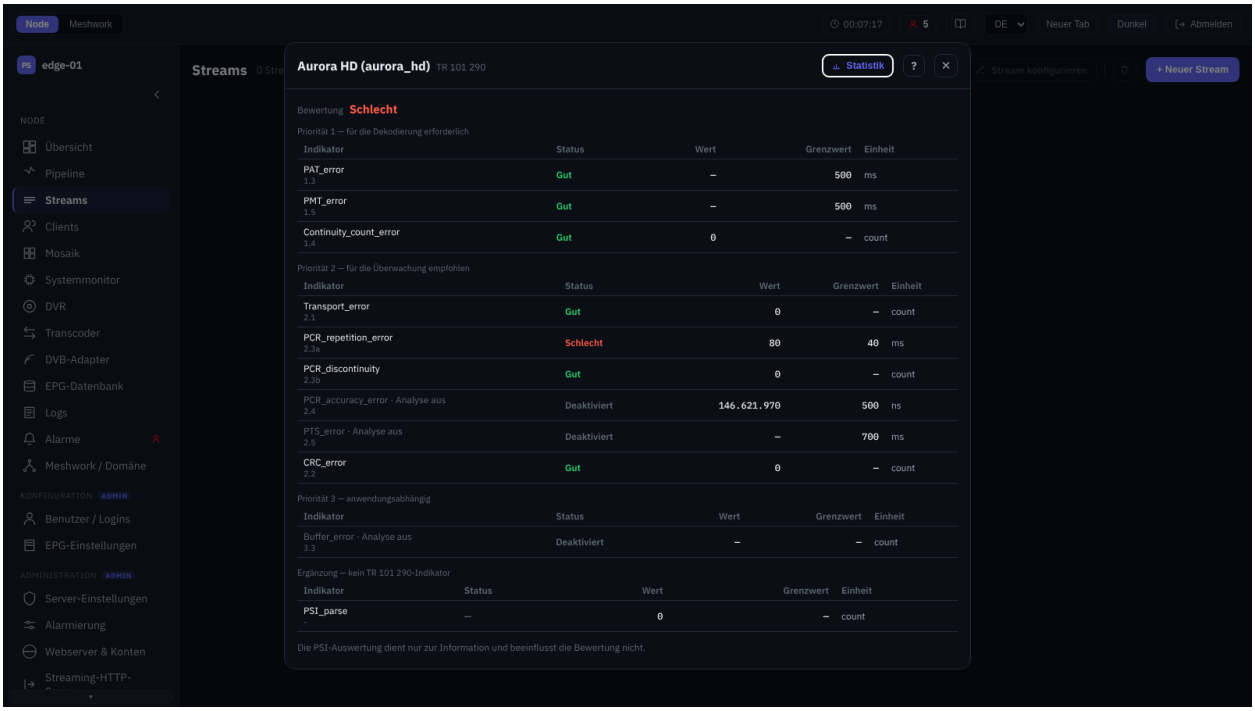


Abb. 23: Analysebericht TR 101 290.

Multiplex / KPI
: Für SPTS — die wichtigsten Zähler (Eingangsbitrate, CC-Fehler im Fenster und insgesamt, verwürfelte Pakete); für MPTS — die Multiplex-Tabelle mit den Summenraten und dem Zustand der Programme (*Analyse nach Programmen*).

Diagramme
: Zeitreihen: Bitrate, Continuity und Verwürfelung, PCR-Jitter, Paketverlust — mit Auswahl von Fenster und Maßstab.

MPEG-TS-Prüfung
: Eine Schnellprüfung der Struktur (TS, PAT, PMT, PCR, Audio/Video und deren ES), das Gesamturteil nach TR 101 290 und die Merkmale der Programmsignalisierung (*Gültigkeitsprüfung*).

Eingangsdiagnose
: Aufklappbare Blöcke: „Medieninformation“, „Eingangsverbindung“, BISS-Entschlüsselung, EIT-Erzeugung, „Analysator“ (PCR- und Synchronisationsmetriken und darunter die PID-Tabelle des aktiven Eingangs mit den Spalten „PID“, „Typ“, „Bitrate“ und „Pakete/s“), Stuffing und „Tiefenanalyse“ — GOP-Struktur, T-STD-Puffer, Videocodec-Steckbrief, PTS/DTS-Sprünge, SCTE-35 (*Fortlaufende Messungen*, *Vertiefte Analysen*). Die Zusammensetzung des Abschnitts hängt vom Eingang ab: bei einem Muxer-Eingang gibt es den Abschnitt überhaupt nicht — er besitzt keinen Socket, und die Programme zeigt die Multiplextabelle; beim Netzwerkempfang eines fertigen MPTS bleibt „Eingangsverbindung“ mit den Kennwerten des eigenen Transports erhalten, während „Medieninformation“, „Analysator“ und „Tiefenanalyse“ ausgeblendet sind — Messungen zu einem einzelnen Programm gibt es bei einem solchen Eingang nicht. Die BISS-Entschlüsselung wird nur bei SPTS angezeigt: bei MPTS werden die BISS-Zähler je Programm geführt (*BISS-Entschlüsselung*). Die Blöcke EIT-Erzeugung und Stuffing erscheinen von selbst, wenn der Stream EIT erzeugt oder NULL-Pakete hinzufügt.

Ausgänge
: Telemetrie jedes laufenden Ausgangs: Zustand, Ziel, Bitrate, Zähler des gesendeten Datenverkehrs und transportspezifische Kennwerte (PRO-MPEG FEC, PS1, RIST, SRT, RTMP). Die Ausgänge in den Transporten des gemeinsamen Domain-Katalogs verfügen über die Schaltfläche „In der Bibliothek anzeigen“ (*Bibliothek — dieser Feed*).

OTT / DVR
: Die Vorlagen der Auslieferungs-URLs, die Metriken des Chunk-Rings und des DVR-Archivs, die Diagramme der Archivabdeckung und der Untertitel-Repliken (*OTT und DVR*, *Archivwiedergabe (VOD)*).

Profiler und Stream-Protokoll
: Die Auslastung der Verarbeitungs-Threads (Anteil eines Kerns) und das strukturierte Ereignisprotokoll des Streams.

Der TR 101 290-Bericht lässt sich auch als separates Fenster öffnen; der Analysator insgesamt ist in *Analysator* beschrieben, der TR 101 290-Monitor — in *TR 101 290-Monitor*. Dem Administrator stehen das Zurücksetzen der kumulativen Statistik und das Löschen des DVR-Archivs zur Verfügung.

Adaptives Bundle

Das Konfigurationsfenster des adaptiven Bundles (ABR): der Name des Bundles und der Name in der URL, die Zusammensetzung der beteiligten Streams und deren Bitraten („0“ — automatisch). Teilnehmer können nur SPTS-Streams mit aktivierter OTT-Auslieferung sein; sie werden direkt im Fenster hinzugefügt und entfernt, die Bitrate wird für jede Variante festgelegt. Das Bundle wird über eine einzige Master-Playlist veröffentlicht, anhand derer der Client die Qualität auswählt. Die adaptive Auslieferung ist in *Adaptive Bundles* beschrieben, die DVR-Aufzeichnung für OTT — in *OTT und DVR*.

Konfiguration

Die Gruppe **Konfiguration** im Bereich „Knoten“ fasst die Einstellungen zusammen, die nicht zu den Serverdiensten gehören: die Zugangskonten und das Subsystem des elektronischen Programmführers (EPG). Die Bildschirme der Gruppe stehen der Administrator-Rolle zur Verfügung.

Benutzer / Logins

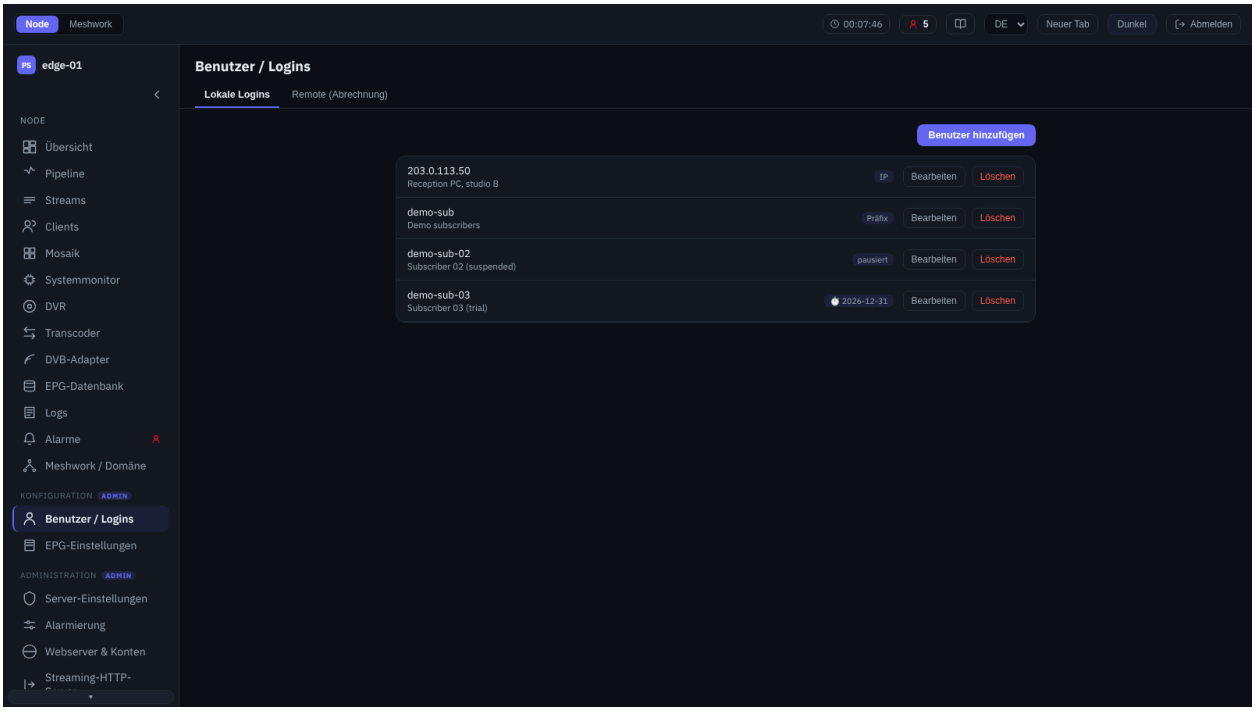


Abb. 24: Benutzer und Logins.

Verwaltung des Zugriffs der Stream-Empfänger auf den Knoten. Der Bildschirm enthält die Registerkarten:

Lokale Logins
: Konten der Stream-Empfänger — der Clients der OTT-Auslieferung und der Peer-Protokolle. Für jedes werden Login und Passwort oder „Konto nach IP-Adresse“ (Feld „IP-Adresse / CIDR“; einzelne Adresse und Bereiche werden unterstützt), „Erlaubte Streams“ (leere Liste — alle sind erlaubt), die Grenzwerte gleichzeitiger Verbindungen je Protokoll (PS1, SRT, HTTP/HLS), „Ablaufdatum“, Pause und „Präfix-Login“ für die Integration mit Middleware (*12. Integration mit Middleware*) festgelegt. Die Anmeldekonten der Weboberfläche und ihre Rollen werden auf dem Bildschirm „Webserver & Konten“ (*Webserver & Konten*) konfiguriert, die Rollen sind in *Zugriff und Rollen* beschrieben.

Extern (Billing)
: Autorisierung der Stream-Empfänger über ein externes System: anstelle der lokalen Prüfung wird der Login durch eine HTTP-Anfrage an einen Fremdserver bestätigt. Oben auf der Registerkarte — die Karte „Billing-Dienst“ mit dem Parameter „Intervall der Billing-Anfragen (s)“. Darunter — die geordnete Liste „Billing-Server“; bei der Prüfung eines Logins werden die Server in der Reihenfolge der Liste abgefragt (die Reihenfolge wird per Ziehen festgelegt, auf einem schmalen Bildschirm über die Tastenkombinationen `Alt+↑` und `Alt+↓`, siehe *Bedienung über die Tastatur*). Für einen Server werden „Name“, „URL“, Passwort, „Zeitüberschreitung (s)“, „Sitzungsabrechnung“, „Login-Präfix“ und „Login-Suffix“ festgelegt, während der „Konnektor“ mit dem Wert „Universal (HTTP)“ den Austausch

vollständig beschreibt — die Vorlage der HTTP-Anfrage (Query, Body, „Content-Type“, Header mit Vorlagenersetzungen von Login, Passwort, IP usw.) und die Auswertung der Antwort („Antwortformat“ — HTTP-Statuscode, JSON-Body oder XML-Body — mit den Pfaden zum Erfolgskennzeichen und zum Ablehnungsgrund); „Tracing“ aktiviert die Protokollierung des Austauschs. Der „Präfix-Login“ als integrierte Alternative zum externen Billing ist in *12. Integration mit Middleware* beschrieben.

Benutzer: hinzufügen und bearbeiten

Fenster zum Anlegen und Bearbeiten eines Kontos eines Stream-Empfängers; es wird über die Schaltfläche „Benutzer hinzufügen“ oder „Bearbeiten“ in einer Listenzeile auf der Registerkarte „Lokale Logins“ geöffnet. Felder:

- „Login“ und „Passwort“ — ein gewöhnliches Konto; beim Bearbeiten belässt ein leeres Passwort das aktuelle. Das Kennzeichen „Konto nach IP-Adresse“ ersetzt das Paar „Login — Passwort“ durch das Feld „IP-Adresse / CIDR“ (einzelne Adresse oder Bereich), und „Präfix-Login“ aktiviert den Modus der Integration mit Middleware (*12. Integration mit Middleware*).
- „Erlaubte Streams“ — die Streams und adaptiven Bundles, zu denen das Konto zugelassen ist; eine leere Liste erlegt keine Einschränkungen auf — alle sind erlaubt.
- „Max. Verbindungen“ — separate Grenzwerte gleichzeitiger Verbindungen für PS1, SRT und HTTP/HLS.
- „Ablaufdatum“ und „Angehalten“ — planmäßige oder vorübergehende Abschaltung des Zugriffs ohne Löschen des Eintrags.
- „Anzeigename“ und „Notiz“ — für die eigene Verwaltung; sie wirken sich nicht auf den Zugriff aus.

Die Anmeldekonten der Weboberfläche und ihre Rollen werden nicht hier, sondern auf dem Bildschirm „Webserver & Konten“ (*Webserver & Konten*) festgelegt.

EPG-Einstellungen

Konfiguration des elektronischen Programmführers: woher die Daten stammen und wie sie ausgeliefert werden. Die Ansicht des gesammelten Programms ist auf einen separaten Bildschirm „EPG-Datenbank“ (*EPG-Datenbank*) ausgelagert. Der Bildschirm enthält die Registerkarten:

Import
: Allgemeine Parameter des Imports des Programmführers: „Import aktivieren“, „Verlauf aufbewahren, Tage“ (Aufbewahrungstiefe vergangener Ereignisse), „Automatische Bereinigung verwaister Kanäle“ sowie „URL der XMLTV-Quelle“ und „Name der XMLTV-Quelle“ der externen Hauptquelle. Der EPG-Import insgesamt ist in *EPG/XMLTV-Import* beschrieben.

Quellen
: Liste der externen XMLTV-Quellen. Für jede — „Name“, „URL / Pfad der Quelle“, das Kennzeichen „Aktiviert“, „Aktualisierungsintervall, s“, „Inhaltskodierung“ (automatische Erkennung oder erzwungenes GZip), Zugangsdaten und Cookies für geschlossene Webressourcen sowie „Heruntergeladene Kopie speichern“ für die Diagnose (*Konfiguration der Quellen*).

Kanalsätze
: Benannte Sätze, die Kanäle für die Auslieferung an externe Abnehmer gruppieren. Auf der Registerkarte wird ein Satz angelegt (Name und Notiz); seine Zusammensetzung — die Kanalzugehörigkeit — wird in der EPG-Datenbank festgelegt (*EPG-Datenbank*). Die Anwendung der Sätze bei der Auslieferung ist in *EPG für OTT-Middleware* beschrieben.

Konten
: Zugriff externer Abnehmer auf den integrierten EPG-Server — den XMLTV-Download. Für ein Konto

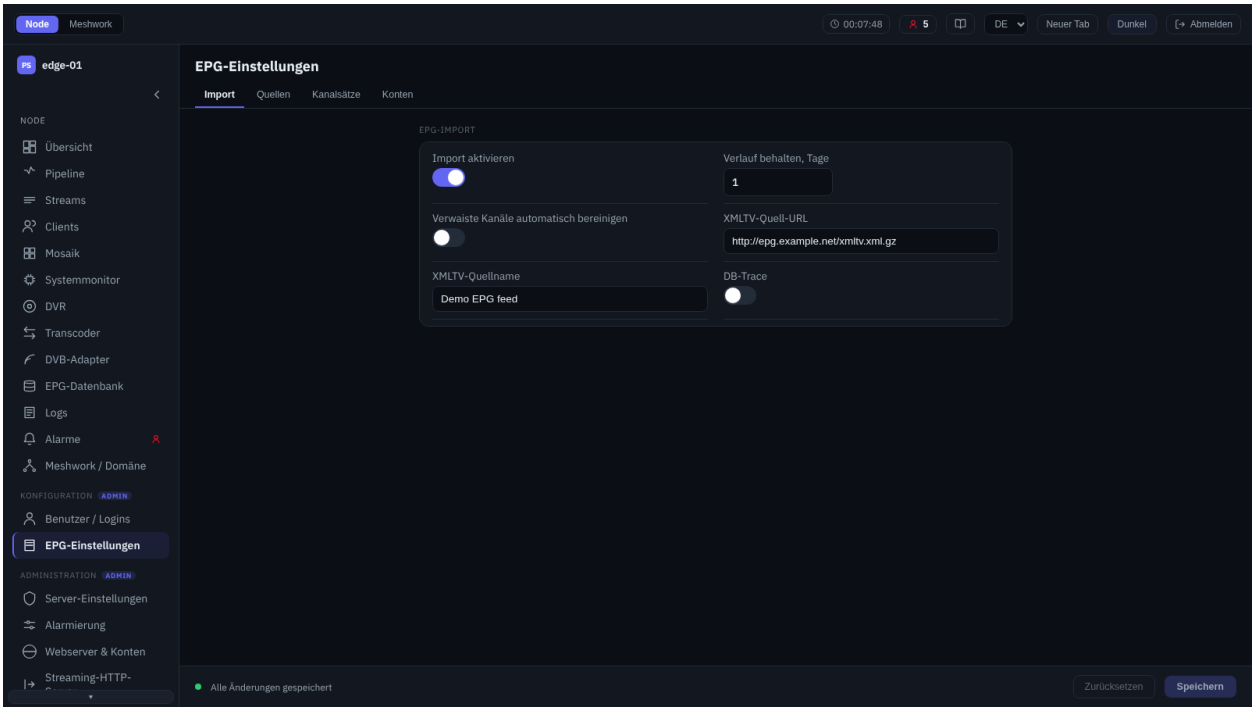


Abb. 25: EPG-Einstellungen (Import).

werden Login und Passwort, „Kanalsatz“ (bestimmt die Zusammensetzung der Auslieferung; ohne Satz ist die Auslieferung leer), „Ablaufdatum“, „Erlaubte IP“ und das Kennzeichen „Angehalten“ festgelegt. Die XMLTV-Auslieferung ist in *XMLTV-Auslieferung* beschrieben.

Das EPG-Subsystem insgesamt — die Erzeugung der EIT und die Auslieferung für OTT-Middleware — ist in *EPG/EIT* beschrieben.

Verwaltung

Die Gruppe **Verwaltung** im Bereich „Knoten“ umfasst die Serverdienste des Knotens, die Speicher, die Hardware, die Lizenz und die Wartung. Die Bildschirme der Gruppe stehen der Administrator-Rolle zur Verfügung. Die Erstkonfiguration des Knotens ist in *Erster Start und Aktivierung* beschrieben.

Server-Einstellungen

Allgemeine Parameter des Knotens. Der Bildschirm fasst mehrere Einstellungsgruppen unter einer gemeinsamen Speicherleiste zusammen:

Server
: Servername, Protokollierungsstufe (global und separat für die Bibliotheken SRT, RIST und FFmpeg), Aufbewahrungsdauer der Protokolle, Tiefe und Tracing der Datenbankstatistik, Größenbegrenzung der Protokolldatenbank.

Mosaik
: Knotenweiter Schalter für die Mosaik-Erzeugung; das Mosaik selbst — *Mosaik*.

Meshwork
: Aktivierung der Teilnahme des Knotens am Meshwork und dessen Identifikation: Knotenname, Notiz,

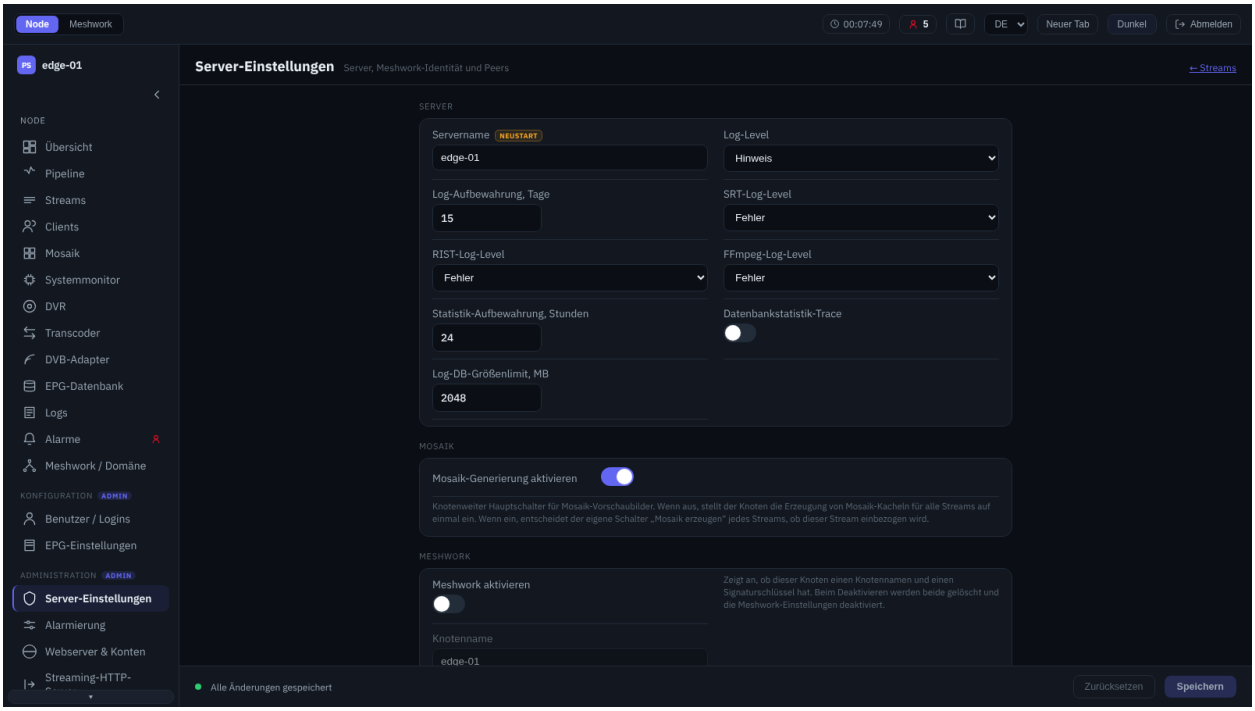


Abb. 26: Server-Einstellungen.

Meshwork-Domain sowie die öffentlichen HTTP- und HTTPS-Ports und zusätzliche Domains — die Adressen, unter denen der Knoten von außen erreichbar ist (*Knoten hinter NAT*). Das Identitätsmodell — *Knotenidentität*.

Meshwork-Peers
: Liste der Peer-Knoten und das gemeinsame Signaturgeheimnis für den gesicherten Austausch (*Peers und Secrets*).

Meshwork-Kontakt (erweitert)
: Parameter der Peer-Abfrage: Anzahl der Worker-Threads und Kontakt-Timeout.

Die Erstkonfiguration des Knotens — in *Anfangseinstellungen*; die Bereitstellung des Meshwork — in *Aufbau eines Meshwork-Netzwerks*.

Alarmierung

Auslöseschwellen der Alarme des Knotens und die Regeln ihrer Replikation. Neben dem globalen Einschalten der Alarmierung werden festgelegt:

- Stream-Schwellenwerte — Timeout ohne Daten, minimale Bitrate, CC-Fehlerzähler (Paare „Warnung / Fehler“);
- eine eigene Warnung zur CPU-Auslastung durch den Arbeits-Thread, der einen Stream oder einen DVB-Adapter bedient (in Prozent);
- Ressourcen-Schwellenwerte — CPU und Speicher des Hosts, CPU und Speicher von Prozess und Transcoder, Netzlast, GPU, Festplattennutzung (in Prozent, Paare „Warnung / Fehler“);
- Lese-/Schreibzeit der DVR-Speicher;

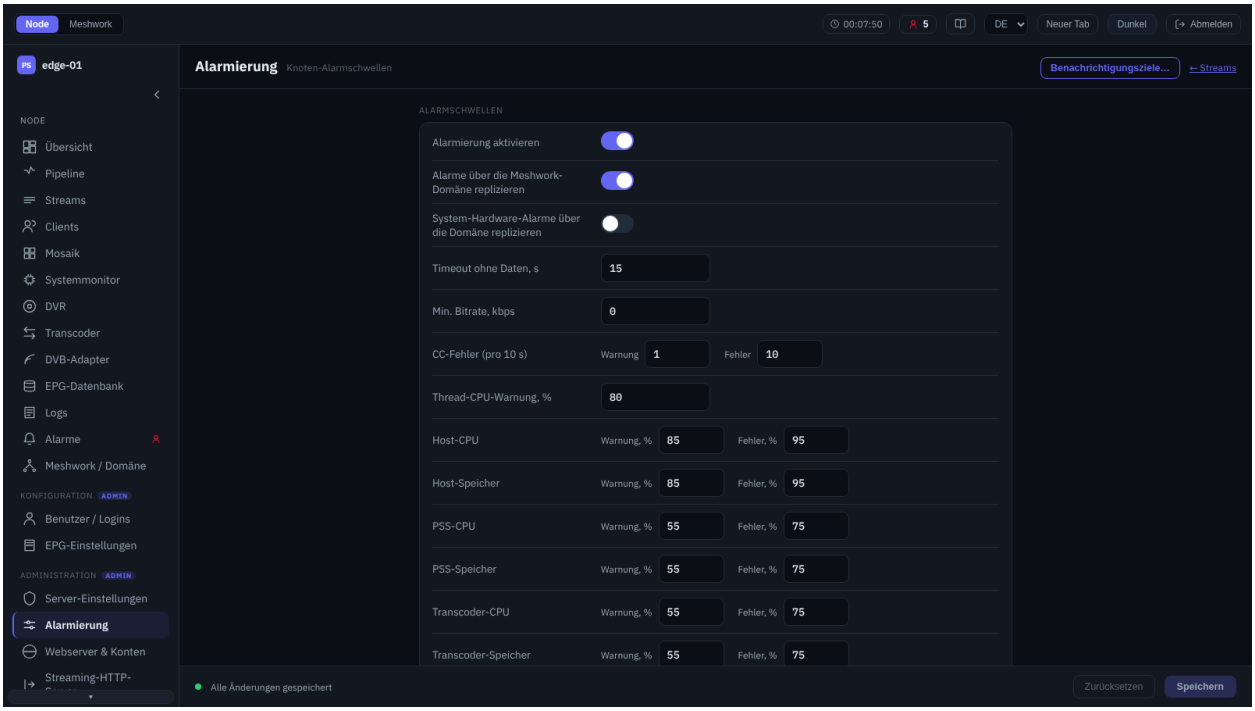


Abb. 27: Alarmierungseinstellungen.

- bei vorhandenem DVB-Empfang — Schwellenwerte der Signalqualität der Frontends (Signalpegel, SNR, Bitfehlerrate, unkorrigierbare Blöcke).

Die Replikation der Alarme über die Meshwork-Domain und, getrennt davon, der System-Hardware-Alarme wird ebenfalls hier aktiviert (*Domänenweite Replikation von Benachrichtigungen*). Die externe Zustellung wird in einem eigenen Fenster konfiguriert (*Alarmziele*).

Das Alarmmodell ist in *Wie ein Alert aufgebaut ist* beschrieben, die Schwellenwerte — in *Alerter-Einstellungen*. Die Ansicht der aktiven Alarme — auf dem Bildschirm *Alarme*.

Alarmziele

Das Fenster der externen Alarmzustellung wird vom Bildschirm *Alarmierung* aus geöffnet. Die Kanäle sind auf Registerkarten verteilt; für jeden wird ein eigener Schwellenwert der minimalen Ereignisschwere festgelegt:

Prozess
: Ausführung eines externen Befehls bei einem Alarmereignis mit vorgegebenem Timeout.

Telegram
: Versand an Telegram: Bot-Token und Liste der Chat-IDs.

E-mail
: Versand über SMTP: Host und Port des Servers, TLS-Modus (STARTTLS, implizites TLS oder ohne Verschlüsselung), Prüfung des Serverzertifikats, Zugangsdaten, Absender- und Empfängeradressen.

Die Zustellungsregeln sind in *Externe Alert-Zustellung* beschrieben.

Webserver & Konten

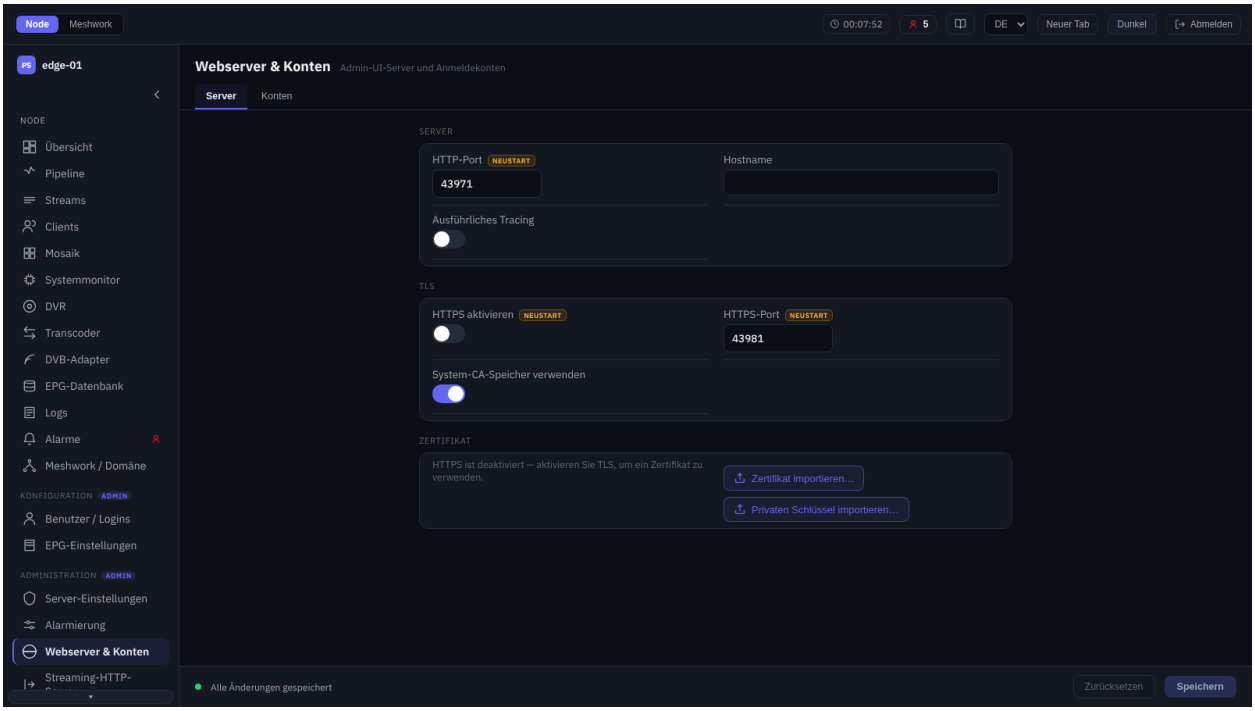


Abb. 28: Webserver & Konten.

Konfiguration des Webservers der Verwaltungsoberfläche und der Anmeldekonten. Der Bildschirm enthält die Registerkarten:

Server
: HTTP-Port und Hostname, Aktivierung von HTTPS mit eigenem Port und Auswahl des CA-Speichers, ausführliches Tracing. Solange HTTPS deaktiviert ist, sind „HTTPS-Port“ und die Auswahl des CA-Speichers ausgegraut und mit dem Hinweis „HTTPS ist deaktiviert — aktivieren Sie TLS, um das Zertifikat zu verwenden.“ versehen; die Felder reagieren auf den Schalter selbst, ohne das Speichern abzuwarten, und die eingegebenen Werte gehen nicht verloren. Darunter — die Angaben zum aktuellen TLS-Zertifikat (Subjekt, Aussteller, Gültigkeitsdauer, SAN) mit den Aktionen zum Import des Zertifikats und des privaten Schlüssels sowie zum Neuladen von TLS; bei deaktiviertem HTTPS zeigt die Leiste anstelle der Angaben denselben Hinweis, der Import bleibt jedoch verfügbar.

Konten
: Lokale Anmeldekonten der Oberfläche und ihre Rollen — Admin, Restricted admin, Viewer (*Zugriff und Rollen*). Für jedes werden Login, Passwort und Rolle festgelegt; Einträge können hinzugefügt, geändert und gelöscht werden. Beim Anlegen ist das Passwort verpflichtend; beim Ändern wird das Passwortfeld leer angezeigt, und bleibt es leer, so bleibt das bisherige Passwort erhalten.

Die automatische Ausstellung von Zertifikaten — auf dem Bildschirm *HTTPS-Zertifikate*; das Neuladen von TLS ohne Neustart — ebenfalls in *Wartung*. Die Ports der Dienste und die Erstkonfiguration sind in *Anfangseinstellungen* beschrieben.

Streaming-HTTP-Server

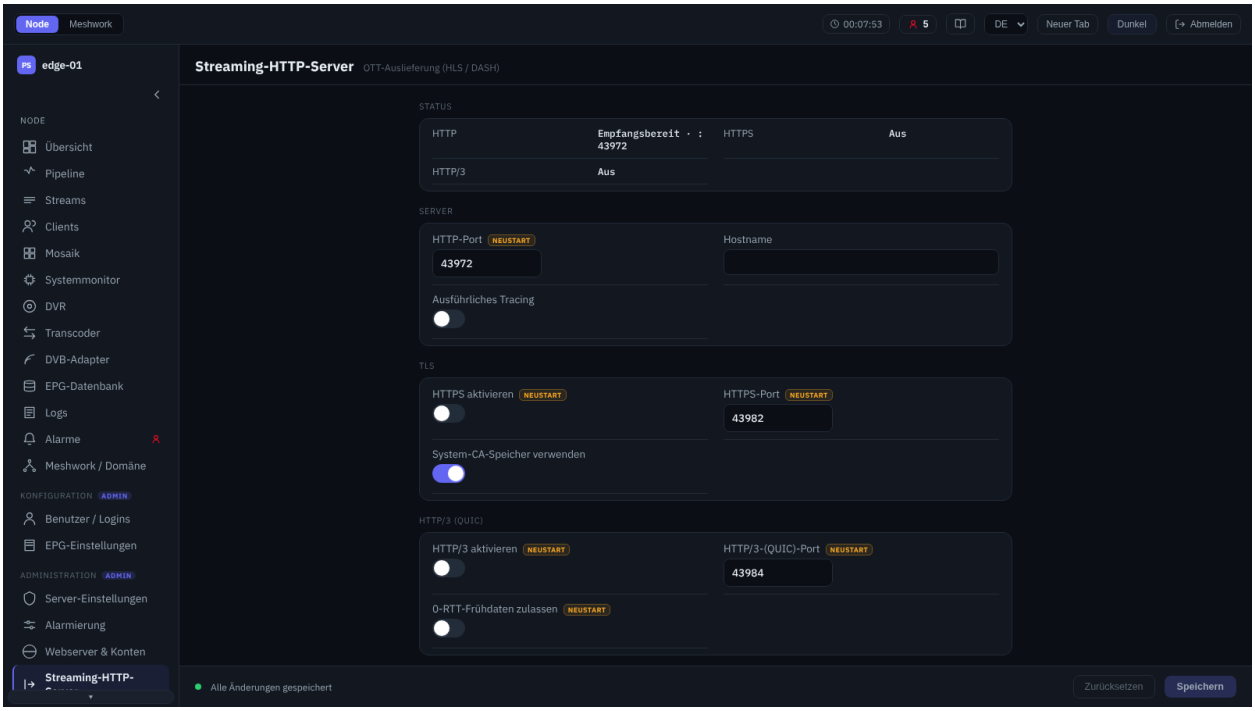


Abb. 29: Streaming-HTTP-Server.

Konfiguration des integrierten HTTP-Servers für die Stream-Auslieferung (OTT-Auslieferung, HLS/DASH-Playlists und -Segmente). Oben — der Status der Listener (HTTP, HTTPS, HTTP/3). Konfiguriert werden:

Server
: HTTP-Port, Hostname, ausführliches Tracing.

TLS
: Aktivierung von HTTPS mit eigenem Port und Auswahl des CA-Speichers. Port und Speicher sind ausgegraut, solange HTTPS deaktiviert ist.

HTTP/3 (QUIC)
: Aktivierung von HTTP/3 auf einem eigenen UDP-Port und Erlaubnis früher Daten (0-RTT); Port und 0-RTT sind ausgegraut, solange HTTP/3 selbst deaktiviert ist. Von HTTPS hängt dieser Block nicht ab: HTTP/3 ist ein eigener Listener, er nimmt die Zertifikatsdateien direkt und arbeitet auch bei deaktiviertem HTTPS. Genau das zeigt der Block „Status“: „HTTPS“ kann auf „Aus“ stehen, während „HTTP/3“ läuft.

Darunter — die Leiste des aktuellen TLS-Zertifikats mit Import und Neuladen von TLS. Die Leiste verfolgt nur HTTPS, daher bleibt sie bei deaktiviertem HTTPS auch dann stumm, wenn dasselbe Zertifikat über HTTP/3 ausgeliefert wird. Die Auslieferung ist in *OTT und DVR* beschrieben, der HTTP/3-Transport — in *HTTP/3 (QUIC)*.

EPG-Server

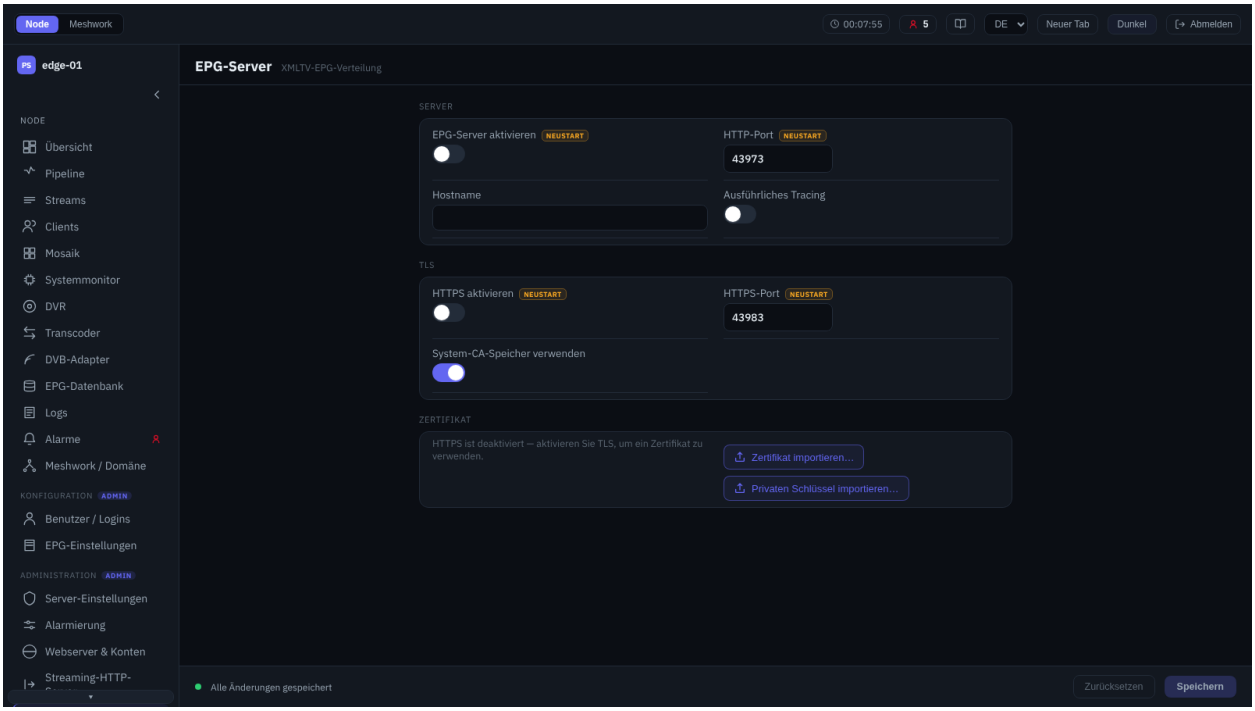


Abb. 30: EPG-Server.

Konfiguration des Dienstes zur EPG-Auslieferung (XMLTV) an externe Abnehmer — Middleware und Anwendungen. Festgelegt werden die Aktivierung des Dienstes, Port, Hostname und ausführlicher Trace, HTTPS mit eigenem Port und CA-Speicher; zudem steht die Leiste des TLS-Zertifikats zur Verfügung. Der HTTPS-Port und der CA-Speicher sind ausgegraut, solange der Dienst deaktiviert oder HTTPS deaktiviert ist. Die Zugangskonten für EPG und die Kanalsätze werden separat auf dem Bildschirm *EPG-Einstellungen* konfiguriert. Die Integration mit Middleware ist in *EPG für OTT-Middleware* beschrieben.

HTTPS-Zertifikate

Automatische Ausstellung und Erneuerung von TLS-Zertifikaten über ACME (Let's Encrypt). Oben — der Status: Zustand der Ausstellung, Zeitpunkt des letzten Laufs, das ACME-Konto und die Liste der ausgestellten Zertifikate mit ihren Fristen. Konfiguriert werden:

Automatische Ausstellung
: Aktivierung und Kontakt-E-Mail. Die Aktivierung bedeutet die Zustimmung zu den Bedingungen der Zertifizierungsstelle.

Erweitert
: URL des ACME-Verzeichnisses, Port der HTTP-01-Prüfung, Vorlauffrist der Erneuerung und die Parameter der externen Kontobindung (EAB).

Die Aktion „Jetzt ausstellen / erneuern" startet die Ausstellung unmittelbar. Das Hochladen eigener Zertifikate erfolgt auf den Bildschirmen der entsprechenden Dienste (*Webserver & Konten*, *Streaming-HTTP-Server*, *EPG-Server*). Die Aktivierung von TLS ist in *Anfangseinstellungen* beschrieben.

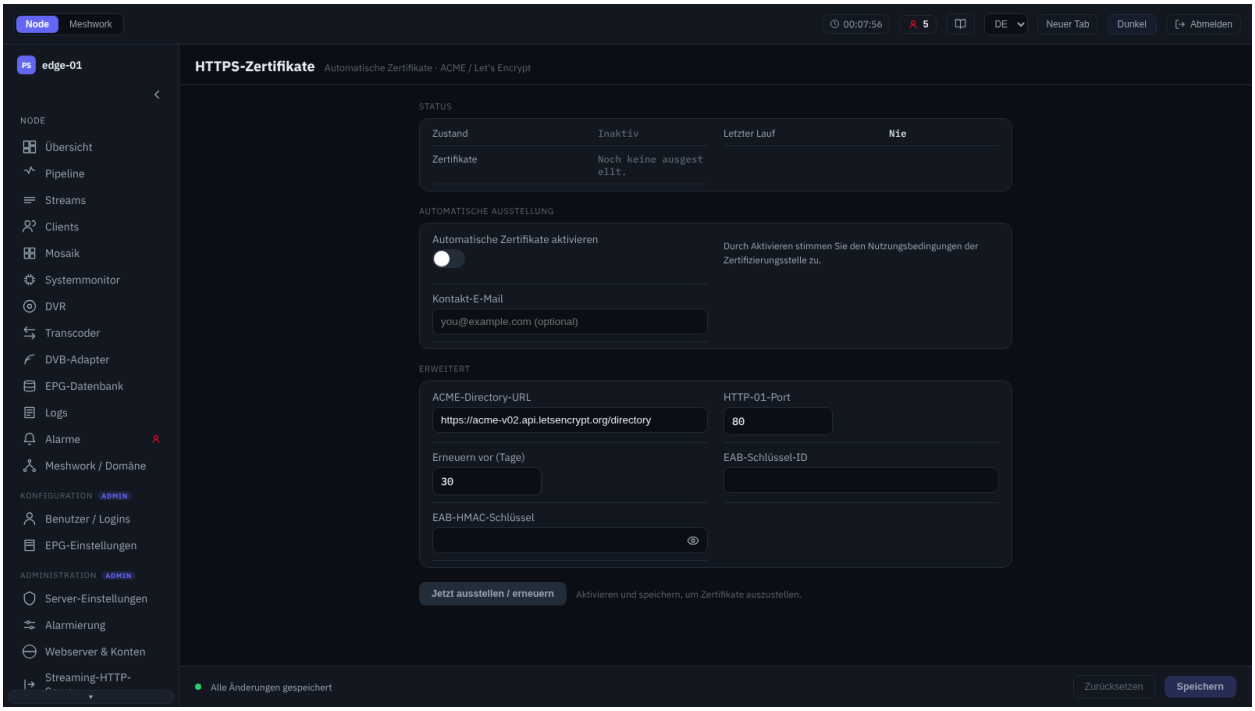


Abb. 31: HTTPS-Zertifikate (ACME).

DVR-Speicher

Verzeichnisse des DVR-Archivs. Speicher können hinzugefügt, geändert und gelöscht werden; für jeden werden Name, Verzeichnispfad (nach dem Anlegen unveränderlich), Notiz und die maximale Festplattennutzung festgelegt. Eine separate Gruppe steuert das Verhalten bei Platzmangel: Bereinigungsintervall, Verzögerung und Kürzung bei Überlauf, Notreserve und Bereinigungshysterese. Eine Senkung des Nutzungslimits kürzt das älteste Archiv; das Löschen eines Speichers belässt die Dateien auf der Festplatte, die angebundenen Streams verlieren jedoch den Zugriff auf VOD. Die aktuelle Belegung der Festplatten wird auf dem Bildschirm *DVR-Monitor* beobachtet. Aufzeichnung und Wiedergabe des Archivs sind in *DVR-Speicher* und *OTT und DVR* beschrieben.

Wartung

Serviceoperationen des Knotens, nach Abschnitten gruppiert:

Knoten
: Regulärer Neustart des Knotens mit Bestätigung — unterbricht kurzzeitig alle Streams.

TLS-Zertifikate
: Erneutes Einlesen des TLS-Zertifikats von der Festplatte ohne Neustart — jeweils separat für den Webserver, den Streaming-HTTP-Server und den EPG-Server.

Konfigurationssicherung
: Herunterladen eines gzip-Abbilds der Knotenkonfiguration.

Wiederherstellung
: Hochladen einer zuvor gespeicherten Kopie: die Konfiguration wird ersetzt, der Knoten startet neu.

Dateien und Dienste sind in *Dateien und Dienste* beschrieben.

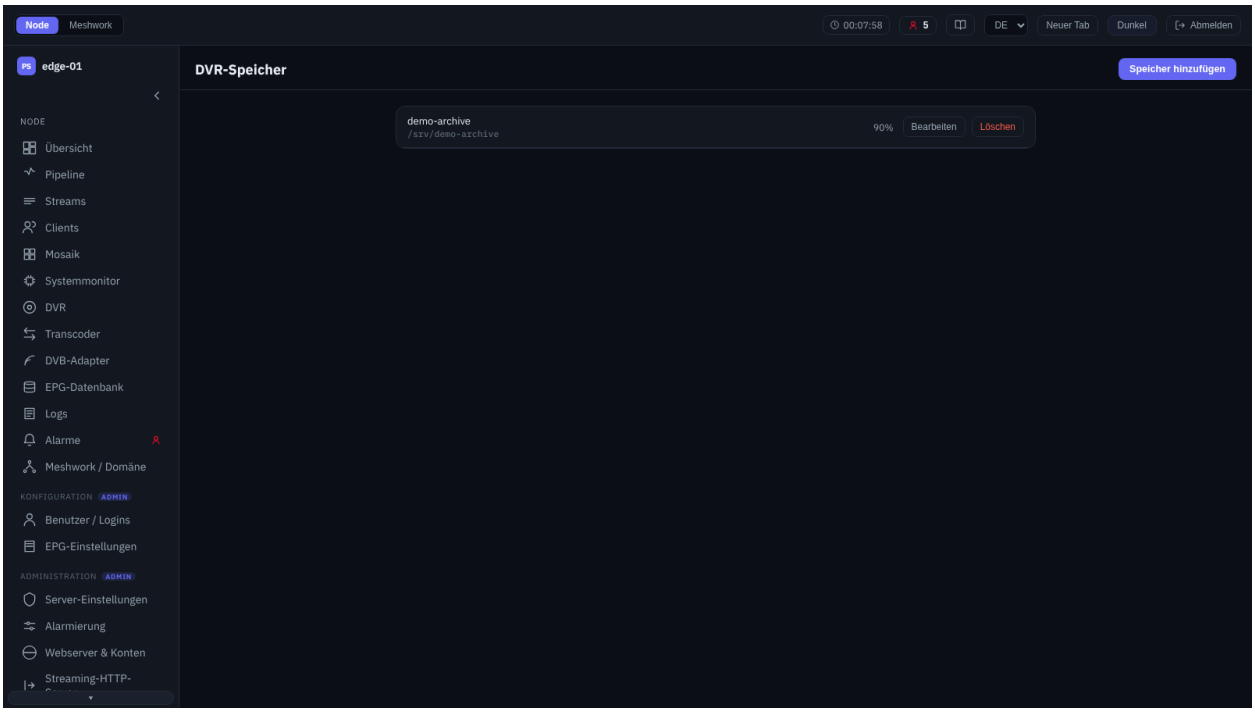


Abb. 32: DVR-Speicher.

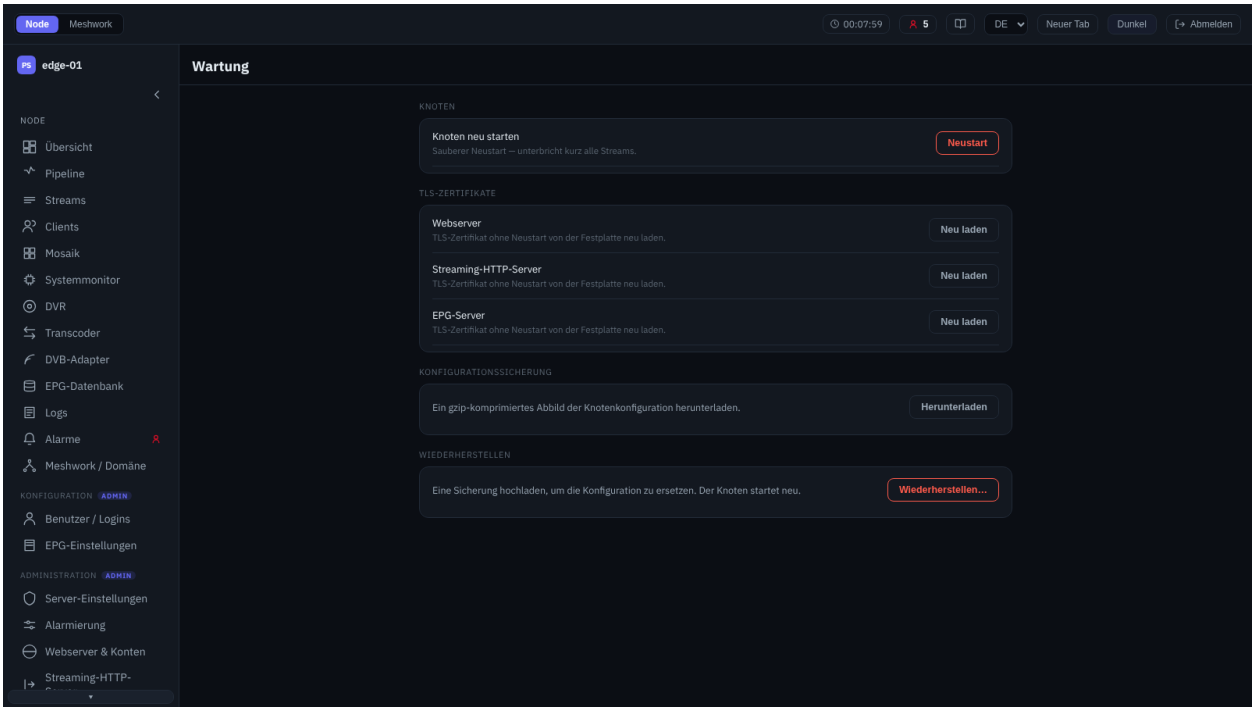


Abb. 33: Wartung.

Hardware

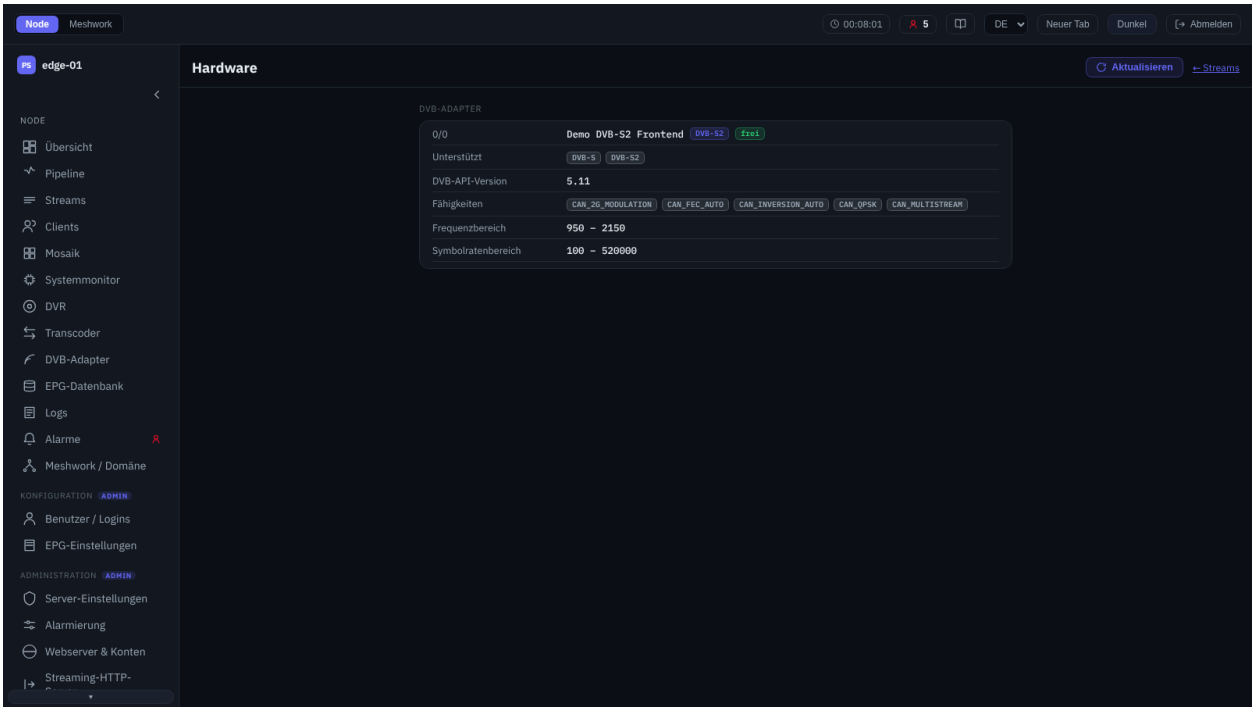


Abb. 34: Hardware.

Die erkannte Empfangs-Hardware, die dem Knoten zur Verfügung steht (schreibgeschützt; die Schaltfläche zum Aktualisieren liest die Liste erneut ein). Zwei Blöcke:

DVB-Adapter
: Frontends des DVB-Empfangs: Bezeichnung, unterstützte Übertragungssysteme, DVB-API-Version, Fähigkeiten, Frequenz- und Symbolratenbereiche, Belegtkennzeichen.

CAM-Module
: CI/CAM-Slots: das eingesetzte Modul und dessen Modell, Typ und Zustand des Slots, die Kennungen der CA-Systeme und der Abonnementstatus (Entschlüsselung) je Programm.

Der DVB-Empfang ist in *DVB-Empfänger* und *Adapter* beschrieben, die Arbeit mit CAM und die Entschlüsselung — in *Entschlüsselung*. Die erkannten Transcoding-Geräte werden auf dem Bildschirm *Über das System* angezeigt.

Lizenz

Angaben zur Lizenz des Knotens: Kunde, Lizenztyp, Grenzwerte für die Anzahl der Streams und Peers (0 — ohne Beschränkung), Demo-Modus, Beginn der Gültigkeit und lizenzierte Zeit. Bei Verwendung eines Hardware-Schlüssels wird der HASP-Block (Kennung und Typ des Schlüssels) angezeigt. Die dauerhafte Aktivierung mit SL/HL-Schlüsseln ist in *Dauerhafte Aktivierung* beschrieben; das Verhalten beim Ablauf der Lizenz — in *Beim Ablauf der Frist der Jahres- oder Testlizenz*.

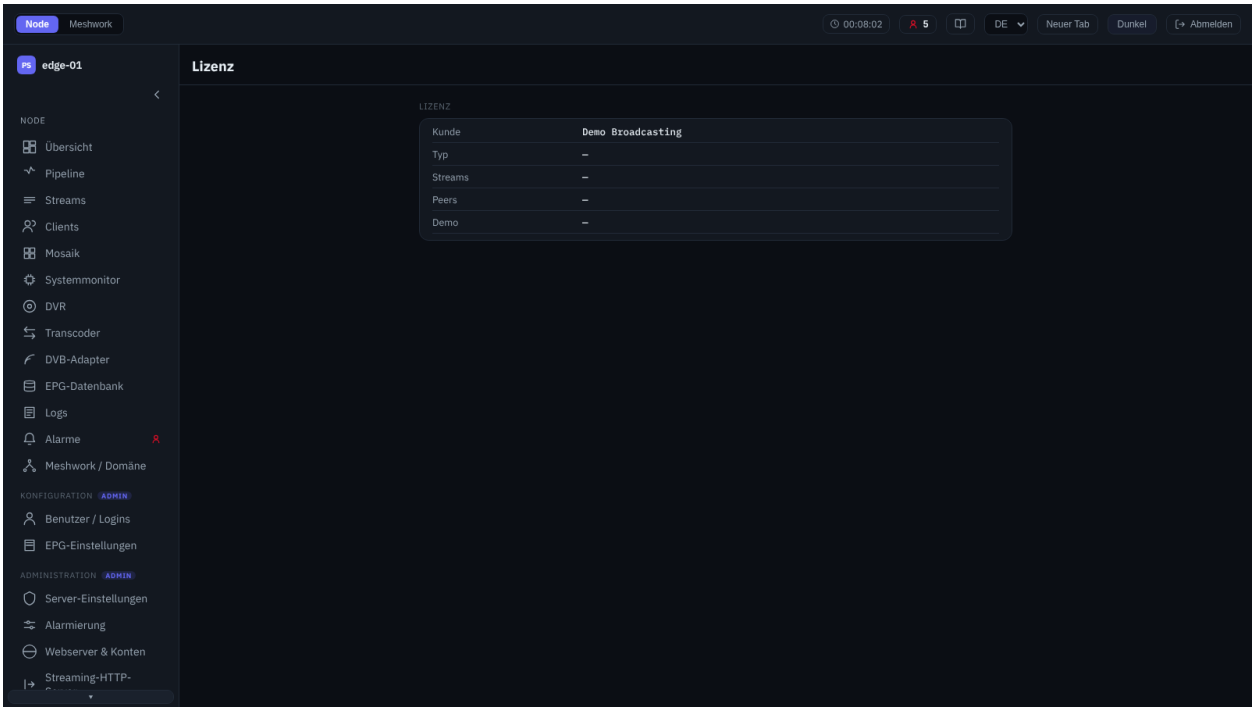


Abb. 35: Lizenz.

Über das System

Eine Systemübersicht, nach Abschnitten gruppiert:

Lizenz
: Kurzangaben zur Lizenz — Einzelheiten auf dem Bildschirm *Lizenz*.

Transcoder
: Die erkannten Transcoding-Backends: Typ, Version des ausführbaren Moduls, Bereitschaft und die Liste der Geräte. Die Installation der Transcoder-Pakete — in *Transcoder*, die Auswahl des Backends — in *Backends und Codecs*.

Build und Bibliotheken
: Die PSS-Build-Version und die Versionen der integrierten Bibliotheken, der Build-Typ.

Identifikation
: Servername, Knoten und dessen Notiz, das aktuelle Konto, Betriebszeit und Startzeitpunkt, die Beschränkung nach Quell-IP.

Der Aufruf der Dokumentation ist auch über die gemeinsame Oberflächenleiste möglich (*Allgemeine Leiste*).

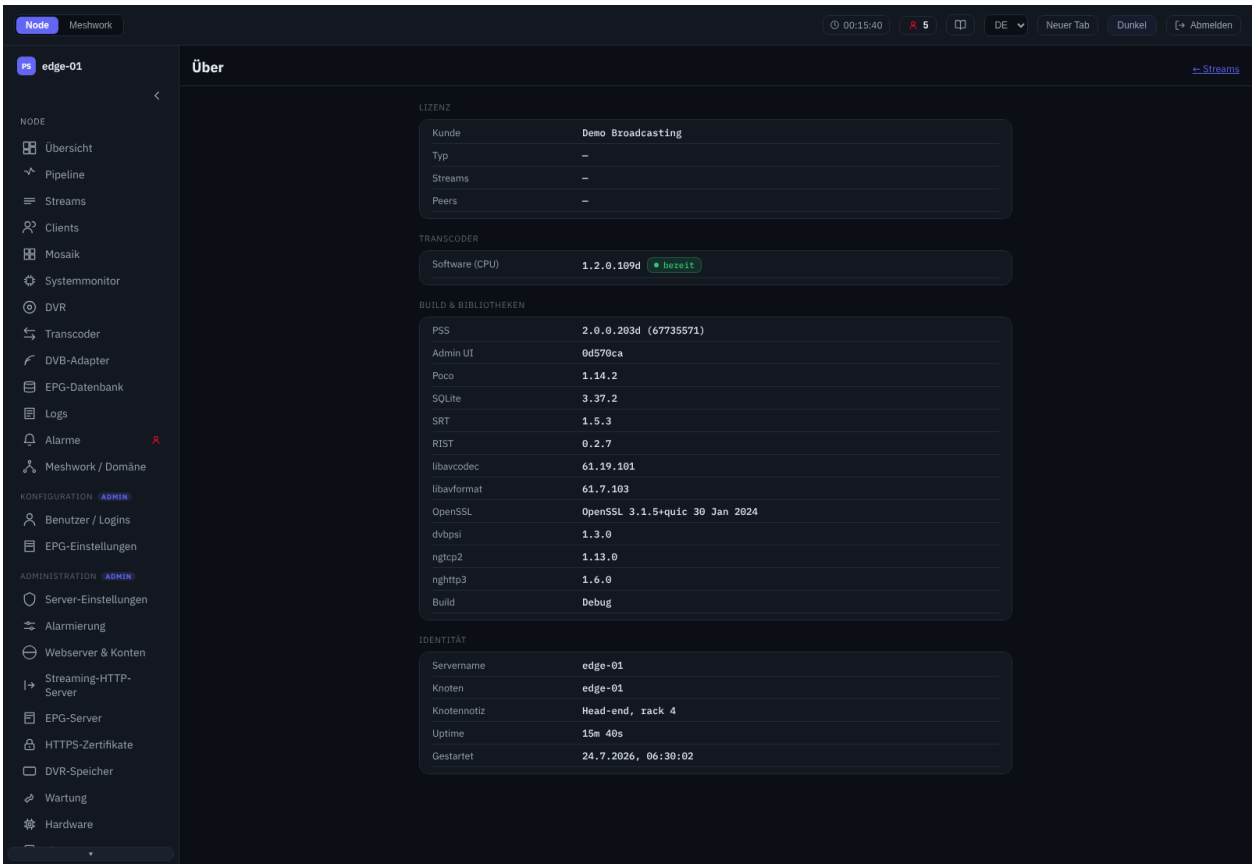


Abb. 36: Über das System.

Meshwork

Der Bereich **Meshwork** (der Bereichsumschalter in der oberen Leiste) fasst die Übersichtsseiten des Knotennetzwerks der Domain zusammen: die Domain-Übersicht, die Transporttopologie, die Zusammensetzung der Peers, die einzelnen Transport-Links und den gemeinsamen Katalog der Netzwerkressourcen. Dies ist die Sicht „von oben“ auf das gesamte Netzwerk, im Unterschied zu den Seiten des Bereichs „Node“, die einen einzelnen Knoten zeigen.

Die Begriffe des Netzwerks, die Domains, die automatische Erkennung von Peers, der Betrieb hinter NAT und die Alarme sind im Abschnitt *Meshwork* beschrieben.

Meshwork-Übersicht

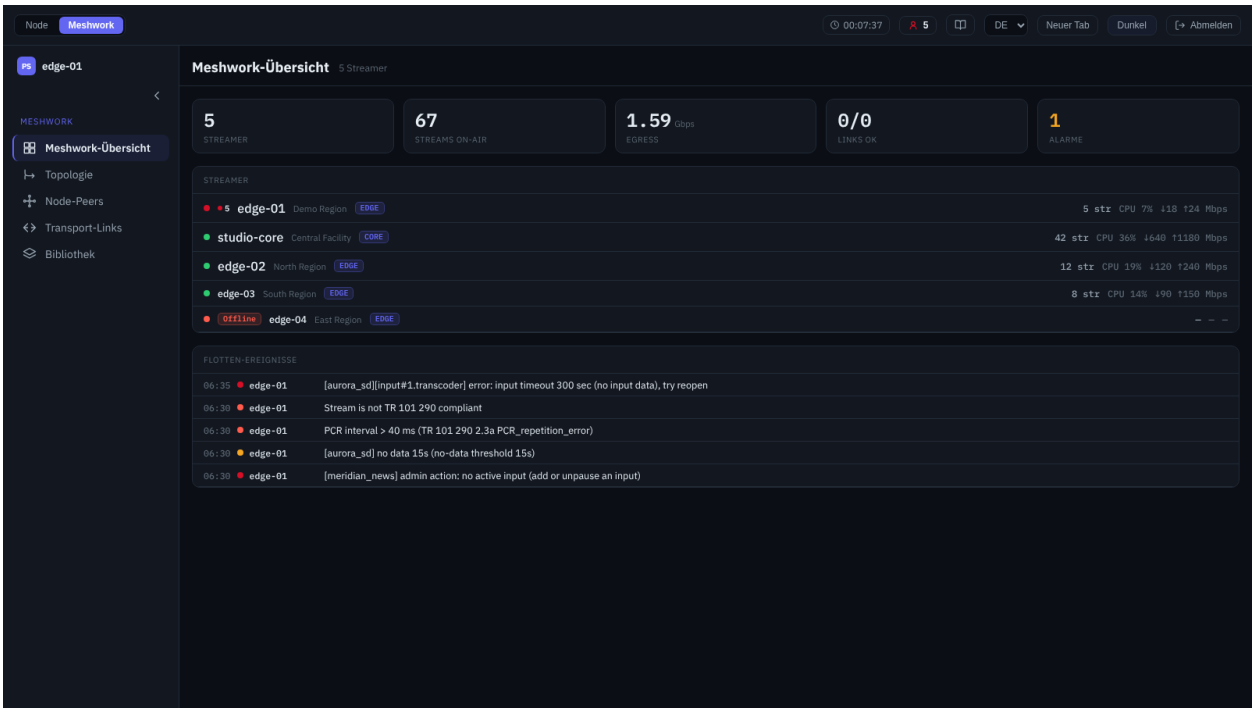


Abb. 37: Meshwork-Übersicht.

Die Übersichtsseite der Domain: alle Knoten des Netzwerks und die wichtigsten Kennzahlen der Domain in einem Fenster. In der Kopfzeile steht die Anzahl der Knoten, darunter folgen die Kennzahlenzeile und die Listen.

Die Kennzahlenzeile vereint die Kacheln „Streamer“ (Anzahl der Knoten), „Streams on-air“, „Egress“ (gesamte ausgehende Bandbreite, Gbps), „Links OK“ (fehlerfreie Transport-Links im Verhältnis zu ihrer Gesamtzahl) und „Alarme“. Die Kachel „Alarme“ zählt die fehlerhaften Knoten und die fehlerhaften aktiven Links und wird bei einem Wert ungleich null hervorgehoben.

Das Panel „Streamer“ enthält eine Zeile je Knoten: den Zustandsindikator und bei einem fehlerhaften Knoten daneben zusätzlich eine Textmarkierung („Beeinträchtigt“ oder „Offline“); den Knotennamen, die Region und die Rolle, die Anzahl der Streams auf Sendung, die CPU-Auslastung und den Verkehr (Ein-/Ausgang). Die Zeile eines Knotens mit erreichbarem Admin-Bereich führt zu dessen Seiten im Bereich „Knoten“. Für die Rolle des Administrators wird neben dem Zustand ein Zähler der aktiven Alarme des Knotens angezeigt.

Das Panel „Flotten-Ereignisse“ ist für den Administrator sichtbar und fasst die aktiven Alarme aller Knoten zusammen: die Zeit, den Knoten, auf den sich das Ereignis bezieht (ein Link auf dessen Admin-Bereich), und den Text. Das Modell des Netzwerks ist in *Meshwork* beschrieben, die Alarme in *Benachrichtigungen (Alerter)*.

Topologie

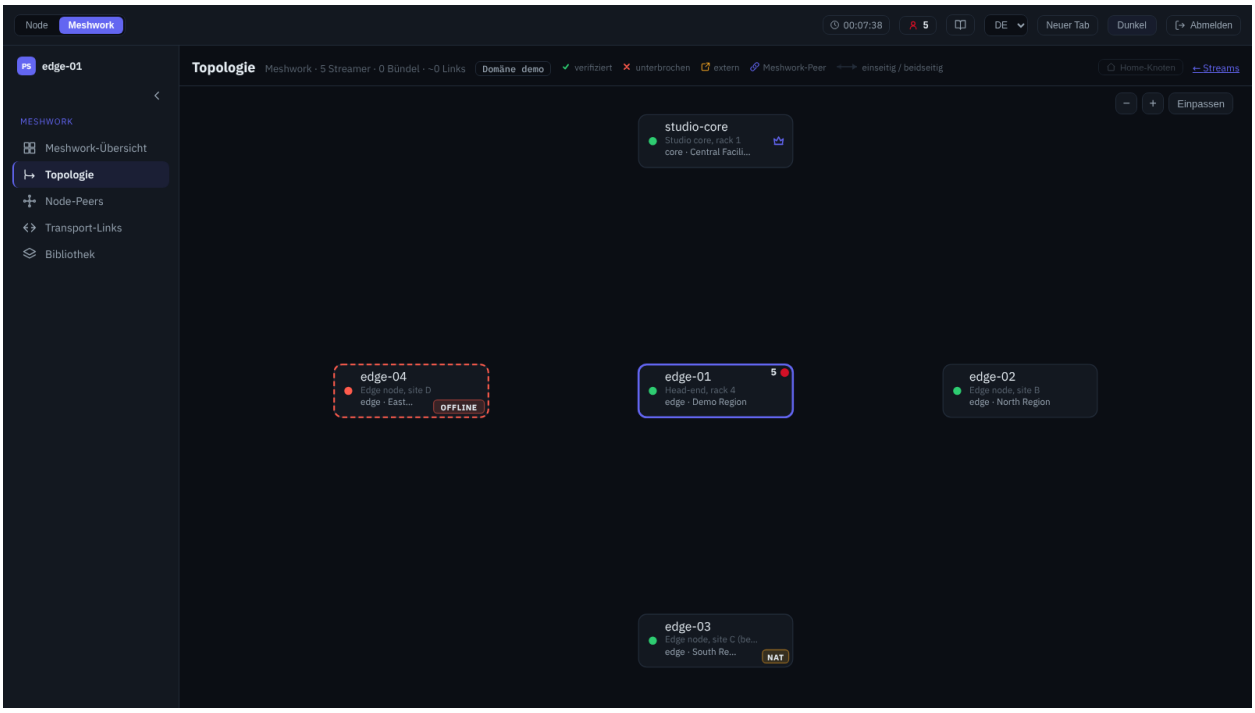


Abb. 38: Topologie-Graph der Flotte.

Der Transportgraph der Domain: die Knoten des Netzwerks und die Transport-Links zwischen ihnen, zu Bündeln zusammengefasst (mehrere gleichgerichtete Links derselben Richtung werden als eine einzige Verbindung dargestellt). Die Seite ist graphbasiert; in der Kopfzeile stehen die Anzahl der Knoten, Bündel und Links, die Domain des verbundenen Knotens, die Verbindungssymbole und die Schaltfläche „Home-Knoten“, die den Mittelpunkt des Graphen auf den aktuellen Knoten zurücksetzt.

Der Graph zeigt, welche Knoten verbunden sind, wo eine Verbindung verifiziert und wo sie abgeleitet ist, welche Verbindungen extern sind (außerhalb der Domain) und welche Knoten hinter NAT stehen. Die Symbole unterscheiden eine verifizierte Verbindung, eine unterbrochene (Peer offline), eine externe Verbindung, eine Verbindung zwischen Meshwork-Peers und die Richtung (einseitig oder beidseitig). Ein Knoten kann einen Zähler der aktiven Alarme tragen. Ein Klick auf einen Knoten verschiebt den Mittelpunkt des Graphen auf ihn (ein erneuter Klick öffnet dessen Admin-Bereich), ein Klick auf eine Verbindung öffnet *Transport-Links* mit einem Filter auf das gewählte Bündel. Die Netzwerkkarte und ihre Symbole sind in *Netzwerkkarte und Ressourcenkatalog* beschrieben, die Felder der Knoten in *Netzwerkkarte*, der Betrieb hinter NAT in *Knoten hinter NAT*.

Node-Peers

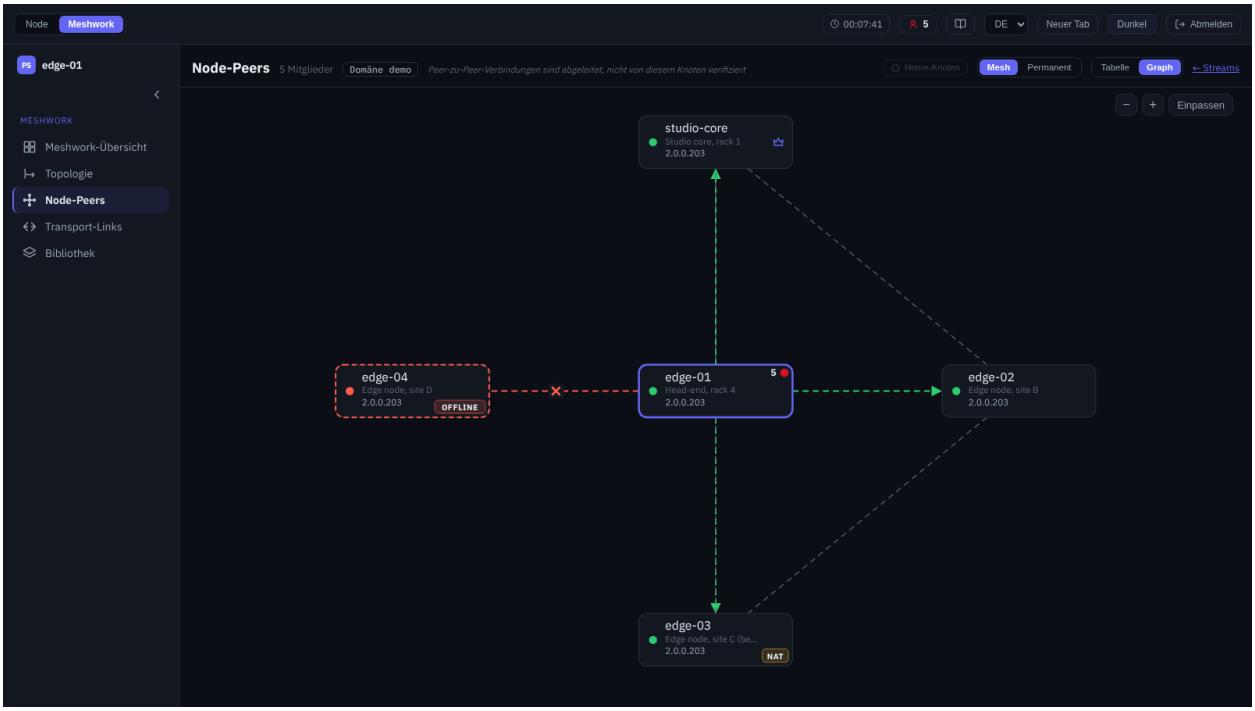


Abb. 39: Node-Peers-Graph (mesh).

Die Zusammensetzung der Domain und die Verbindungen zwischen ihren Teilnehmern. Verfügbar sind der Umschalter des Graphbereichs „Mesh / Permanent“ und der Umschalter der Darstellung „Tabelle / Graph“ (standardmäßig Graph).

Im Graphmodus zeigt „Mesh“ alle Teilnehmer der Domain mit dem verbundenen Knoten im Mittelpunkt: den zuverlässigen Keep-alive-Stern von diesem Knoten aus sowie die abgeleiteten — von diesem Knoten nicht verifizierten — Verbindungen zwischen den übrigen Paaren. „Permanent“ erstellt den Graphen des konfigurierten Peerings: die Knoten und die gerichteten Verbindungen aus den Peer-Listen jedes Knotens (der Pfeil zeigt die Richtung der Konfiguration, beidseitig bei gegenseitiger). Die Schaltfläche „Home-Knoten“ setzt den Mittelpunkt auf den aktuellen Knoten zurück.

Im Tabellenmodus sind die Teilnehmer mit den Spalten „Knoten“, „Adresse“, „Typ“ („Permanent“ oder „Auto“), „Version“, „Status“ („Aktiv“ / „Ausgefallen“, mit der Markierung „Beeinträchtigt“ bei Keep-alive-Fehlern), „NAT“ und „Zuletzt gesehen“ aufgeführt. Ein Knoten hinter NAT zeigt „Internes Netzwerk“ anstelle der nicht erreichbaren privaten Adresse. Der Knotenname ist ein Link auf dessen Admin-Bereich (ein nicht erreichbarer Peer wird als einfacher Text ausgegeben); in der Zeile befindet sich eine Schaltfläche zum Öffnen des Admin-Bereichs. Die Zeile des aktuellen Knotens ist gekennzeichnet und klappt das Panel „Meshwork-Verkehr“ auf — die Aufschlüsselung des ausgehenden Verkehrs nach Anteilen („Basis“, „Peers“, „Bibliothek“, „Alarme“), die TX/RX-Rate und die Zähler des Austauschs. Die Felder der Zeilen sind in *Netzwerkkarte* beschrieben, die Konfiguration der Peers in *Peers und Secrets*, der Betrieb hinter NAT in *Knoten hinter NAT*. Die Sicht auf die Peers aus der Perspektive des aktuellen Knotens bietet die Seite *Meshwork / Domain*.

Transport-Links

Die einzelnen Transport-Links zwischen den Knoten der Domain — eben jene Links, die in der „Topologie“ zu Bündeln zusammengefasst sind. In der Kopfzeile stehen die Kennzahlen „Links“ (gesamt), „Gesund“ (im Verhältnis zur Gesamtzahl) und „Verifiziert“. Beim Wechsel aus dem Graphen der „Topologie“ öffnet sich die Tabelle mit einem Filter auf das gewählte Bündel; der Filter wird über den Link „ alle Links“ aufgehoben.

Jede Zeile beschreibt einen Link: den Zustandsindikator, die Richtung (Spalte „Link“, Knoten → Knoten), „Proto“ (PS1, SRT und weitere Transportkanäle), „Stream“, „Identität“ („verifiziert“ oder „abgeleitet“, mit der Kennzeichnung „Reserve“ für einen Reserve-Link), „Bereich“ („intern“, „extern“, „lokal“ oder „unbekannt“) und die Telemetrie — „TX→RX Mb/s“, „RTT“ und „Verlust“. Der Bereich wird von der Seite übernommen, die sich selbst verbindet, daher lautet er bei Verbindungen über UDP, RTP, Pro-MPEG und RIST „unbekannt“. Die Telemetriewerte erscheinen, sofern die entsprechenden Daten vorliegen, andernfalls wird „—“ angezeigt; lange Listen werden auf Seiten aufgeteilt. Der Verbindungstyp PS1 / SRT ist in *Gemeinsamer Ressourcenkatalog* beschrieben, die Protokolle selbst in *Peer-Protokolle für zuverlässige Übertragung*.

Bibliothek

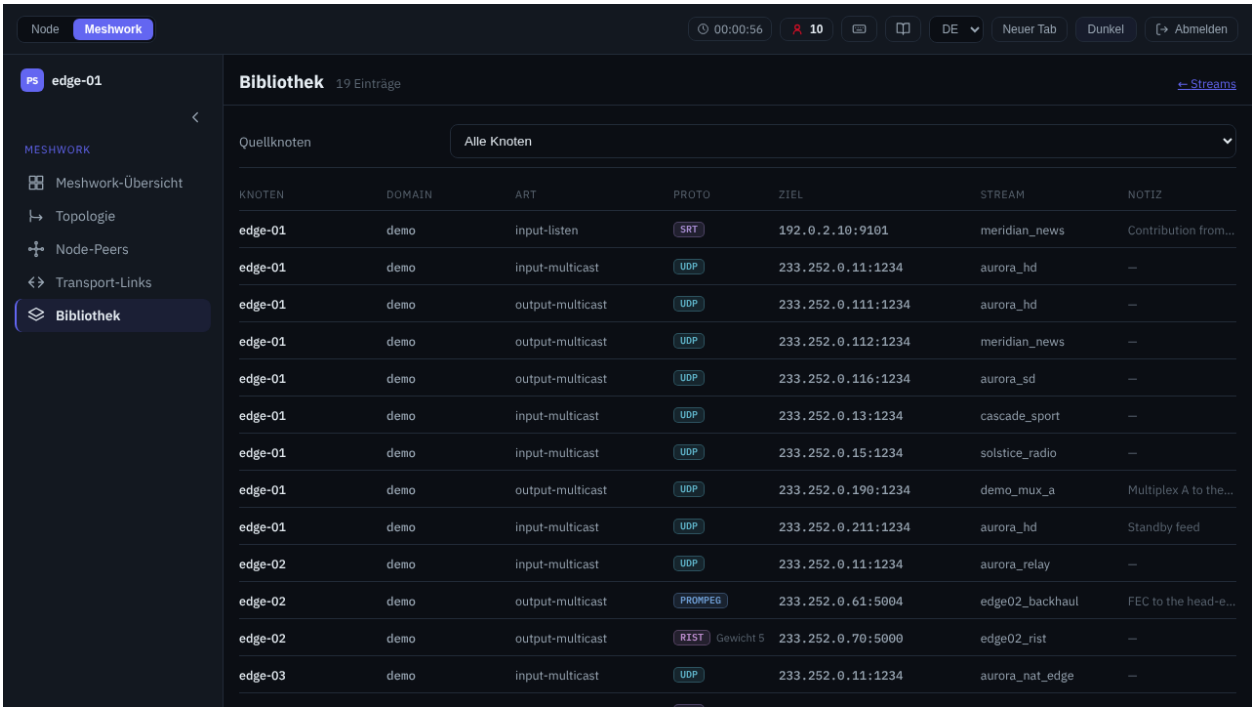


Abb. 40: Bibliothek der Domänenressourcen.

Der gemeinsame Katalog der belegten Netzwerkressourcen der Domain: die Ein- und Ausgänge der Knoten über die Protokolle UDP, RTP, Pro-MPEG, RIST sowie PS1 und SRT. Der Katalog wird automatisch aus den Ankündigungen der Knoten gebildet und ist schreibgeschützt. Anhand von ihm werden freie Adressen und Ports ausgewählt und die benötigten Streams im Netzwerk aufgefunden. In der Kopfzeile stehen die Gesamtzahl der Einträge und der Link „ Streams“.

Über der Liste befindet sich der Filter „Quellknoten“: standardmäßig „Alle Knoten“, in der Werteliste stehen die Knoten, die gerade etwas ankündigen. Der Filter wählt die Einträge eines einzelnen ankündigenden Knotens aus und ändert die Anzahl der Einträge in der Kopfzeile nicht. Die Zeilen sind nach Knoten, Domain und Adresse geordnet.

Die Spalten: „Knoten“ — der ankündigende Knoten, „Domain“, „Art“ (Richtung und Anbindungsweise des Punktes: `output-multicast`, `input-caller`, `output-listen` und ähnliche), „Proto“, „Ziel“ (Adresse und Port), „Stream“ (Name oder Nummer) und „Notiz“. Der Adresse werden der Verbindungsbereich („lokal“, „intern“ oder „extern“; dasselbe Merkmal gibt es bei den Transport-Links — *Transport-Links*), die SSM-Quelle (`SSM <Adresse>`) und die VLAN-Nummer (`VLAN <n>`) beigestellt. Der Verbindungsbereich wird nur bei den Punkten PS1 und SRT ausgegeben, die sich selbst mit der angegebenen Adresse verbinden (`*-caller`): „lokal“ — eine Adresse desselben Knotens, „intern“ — ein durch die Domain-Autorisierung bestätigter Peer, „extern“ — alle übrigen. Bei RIST wird neben dem Protokoll das Gewicht des Links (`Gewicht <n>`) ausgegeben.

Ein Klick auf eine Zeile öffnet den Admin-Bereich ihres Knotens bei der Statistik ihres Streams: die Zeile des eigenen Knotens wird an Ort und Stelle geöffnet, die Zeile eines Nachbarn gemäß der Einstellung „Knoten öffnen in“ der oberen Leiste (*Allgemeine Leiste*); Strg oder erzwingt einen neuen Tab. Die Zeile eines Knotens, der nicht erreichbar oder dem Netz unbekannt ist, hat keine Aktion und trägt den Kurzhinweis „Knoten nicht erreichbar“. Der Kurzhinweis der Zeile zeigt den Zeitpunkt der Ankündigung des Eintrags.

Der Katalog und die Regeln zur Auswahl der Adressen sind in *Gemeinsamer Ressourcenkatalog* beschrieben; die Sicht aus der Perspektive eines einzelnen Ein- oder Ausgangs bietet *Bibliothek — dieser Feed*.

1.9 Zusätzliche Materialien

Anhänge für den Betrieb: Skalierung der OTT-Auslieferung über Caching-Proxys und CDN, Anbindung externer Überwachungssysteme, Optimierung des Knotens für große Installationen sowie Anforderungen an die Intel-Hardware für die Hardware-Transkodierung.

1.9.1 Caching und CDN für die OTT-Auslieferung

Anhang für den Betrieb großer Installationen: wie das Caching der OTT-Antworten von Perfect Streamer aufgebaut ist und wie einem Knoten ein cachender Reverse Proxy oder ein CDN vorgeschaltet wird. Auslieferungsmodi, URL-Formate und Autorisierung sind im Hauptabschnitt *OTT und DVR* beschrieben.

Der Server erzeugt Antworten dreier Kategorien, die sich in der Lebensdauer des Inhalts und in der Eignung für das Caching durch zwischengeschaltete Knoten (Reverse Proxy, CDN, Client-Cache) unterscheiden.

1. Caching-Modell

1.1. Ressourcen und HTTP-Header

Ressource	URL	Content-Type	Cache-Control
TS-Segment (HLS)	`/h<sess>/<keyHex>.ts`	`video/mp2t`	`public, max-age=60, immutable`
VTT-Segment (Untertitel)	`/h<sess>/sub/<pid>/<keyHex>.vtt`	`text/vtt; charset=utf-8`	`public, max-age=60, immutable`
fMP4-Segment (DASH / LL-HLS)	`/h<sess>/init.mp4`, `/h<sess>/<key|N>.m4s`	`video/mp4`	`public, max-age=60, immutable`
DASH MPD	`/h<sess>/index.mpd`	`application/dash+xml; charset=utf-8`	`public, max-age=1`
HLS master	`/hls/<stream>/<login>/<pass>/index.m3u8`	`application/vnd.apple.mpegurl`	`public, max-age=1`
HLS media	`/h<sess>/index.m3u8`, `/h<sess>/sub/<pid>/index.m3u8`	`application/vnd.apple.mpegurl`	`public, max-age=1`
302 Redirect	`/dash/<stream>/<login>/<pass>/index.mpd`	—	`no-cache, no-store`
Raw TS	`/http/<stream>/<login>/<pass>`	`video/mp2t`	nicht gesetzt; wird nicht gecacht

1.2. Eigenschaften der Segmente

Der hexadezimale Segmentbezeichner in der URL (`<keyHex>` in den Pfaden `/h<sess>/<keyHex>.ts`) wird als CRC64 aus der Startzeit des Segments und der Stream-ID gebildet und ist global eindeutig. Die URL des Segments adressiert unveränderlichen Inhalt — bei wiederholten Anfragen derselben URL wird ein identischer Bytestrom zurückgegeben (solange das Segment innerhalb des gleitenden Fensters bleibt).

Die Direktive `immutable` unterdrückt die bedingte Revalidierung durch den Client (`If-None-Match`, `If-Modified-Since`). Der Wert `max-age=60` gewährleistet die Kompatibilität mit einem typischen `timeShiftBufferDepth=40s`.

CMAF-fMP4-Segmente (`.m4s`) und die gemeinsame `init.mp4` für DASH / Low-Latency HLS werden analog adressiert und nach demselben Modell gecacht (`immutable`, `max-age=60`). Teilsegmente (Parts) von LL-HLS sind Byte-Range-Anfragen innerhalb derselben `.m4s` und bilden daher keinen eigenen Cache-Eintrag.

1.3. Eigenschaften der Manifeste

`max-age=1` begrenzt die obere Schranke der Veralterung des Inhalts im Cache auf eine Sekunde. Zusammen mit `proxy_cache_lock on` (nginx) werden Anfragespitzen auf das Manifest zu einer Anfrage pro Sekunde an den Origin zusammengefasst.

1.4. Variabilität des Inhalts

Bei `absPath=0` (Standardwert, der URL-Parameter `a` fehlt) enthalten die HLS-Media-Manifeste und das DASH-MPD keinen Sitzungsbezeichner im Rumpf. Der Inhalt des Manifests ist zwischen Sitzungen identisch, die zu derselben (stream, param)-Kombination gehören. Das erlaubt einem Reverse-Proxy-Cache, den Eintrag bei normalisiertem Cache-Schlüssel sitzungsübergreifend wiederzuverwenden.

Bei `absPath=1` (URL-Parameter `a=1`) enthält der Rumpf des Manifests absolute URLs einschließlich Schema, Host und Sitzungsbezeichner. Der Inhalt wird sitzungsspezifisch, eine sitzungsübergreifende Wiederverwendung des Caches ist nicht möglich.

2. Verhalten der Clients

Client	URL der Manifest-Aktualisierung	Auswirkung auf die Anzahl der Sitzungen
HLS-Player über die Media-Playlist	`/h<sess>/index.m3u8`	Eine Sitzung je Wiedergabevorgang
DASH-Player über die Root-URL	`/dash/<stream>/.../index.mpd` (Wiederholungszyklus)	Wird durch Session Reuse behandelt (siehe 3.2)
Browser-Player (MSE)	`/h<sess>/...` über `<Location>` / Session-URL	Eine Sitzung je Wiedergabevorgang

3. Spezielle Mechanismen

3.1. HTTP 302 Redirect für DASH

Eine Anfrage der Form `/dash/<stream>/<login>/<pass>/index.mpd` liefert die Antwort `302 Found` mit dem Header `Location: /h<sess>/index.mpd`. Der Antwortrumpf ist leer. Autorisierung und Zuweisung der Sitzung erfolgen bei der Verarbeitung des Redirects.

Clients, die das Caching des Redirects unterstützen, greifen in den folgenden Anfragen direkt auf die Session-URL zu. Clients, die es nicht unterstützen, wiederholen die Redirect-Anfrage. Der Aufwand für die erneute Verarbeitung des Redirects beschränkt sich auf die Authentifizierungsprüfung und die Operationen des Session Reuse.

3.2. Session Reuse für DASH

Bei der Verarbeitung einer Anfrage `/dash/.../index.mpd` desselben Logins an denselben Stream (mit demselben Adaptive-Merkmal) findet der Server eine bereits bestehende DASH-Sitzung und gibt deren Bezeichner erneut zurück. Es wird keine neue Sitzung angelegt und kein Slot des Limits gleichzeitiger Verbindungen verbraucht. Wiederverwendet werden nur Sitzungen im gleichen Wiedergabemodus: eine Live-Sitzung und eine VOD-Sitzung des Archivs (Parameter `t`/`d`/`epg`) werden nicht zusammengeführt.

Gilt nur für DASH. Für HLS ist kein eigener Reuse-Mechanismus erforderlich: HLS-Clients aktualisieren die Media-Playlist über die Session-URL und legen nicht bei jeder Aktualisierung eine neue Sitzung an.

3.3. Wiederverwendung von Segmenten zwischen Sitzungen

Der Pfad `/h<sess>/<keyHex>.ts` hängt bei der Auflösung von `<keyHex>` zum Inhalt nicht von `<sess>` ab: `<keyHex>` identifiziert das TS-Segment innerhalb des Streams global eindeutig. Nginx mit normalisiertem Cache Key (der das Präfix `/h<sess>/` abschneidet) bedient alle Anfragen desselben `<keyHex>` aus einem einzigen Cache-Eintrag, unabhängig davon, welche Clients sie gestellt haben.

Die Deduplizierung ist nur auf content-addressed Namen anwendbar — auf Segmente mit einem 16-stelligen hexadezimalen Bezeichner (`.ts`, `.m4s` mit `<keyHex>`, `.vtt`). Nummerierte Segmente von Live-DASH (`<Nummer>.m4s`), `init.mp4` und die Manifeste sind nur innerhalb ihrer Sitzung eindeutig — ihr Cache-Schlüssel muss das Präfix `/h<sess>/` beibehalten, sonst liefert der Cache den Inhalt eines Streams an die Zuschauer eines anderen. Für DASH sorgt Session Reuse für die Deduplizierung der Manifeste zwischen Clients desselben Logins (siehe 3.2).

4. Anfrageparameter

Parameter	Standardwert	Wirkung
`a`	`0`	`1` — absolute URLs in den Manifesten; `0` — relative
`s`	`40`	`timeShiftBufferDepth` in Sekunden
`m`	`40`	Minimale Fensterlänge für die Ausgabe des Manifests
`v`	`6` für die OTT-Modi, `3` für Peering / HLS	`#EXT-X-VERSION` in HLS (wird von DASH ignoriert); ein explizit in der URL angegebener Wert überschreibt den Standardwert
`h3`	nicht vorhanden (off)	Opt-in für HTTP/3 (QUIC): veranlasst den Server, `Alt-Svc` in der Antwort auszugeben; sticky auf der Session-URL. Siehe *HTTP/3 (QUIC)*

Die Änderung eines Parameters über den Query String aktualisiert die in der Sitzung gespeicherten Werte beim nächsten Wiederöffnen der Sitzung. Die Parameter der Archivwiedergabe `t`, `d` und `epg` sind in *Archivwiedergabe (VOD)* beschrieben.

5. Lastcharakteristika

Die Last auf dem Origin skaliert mit der Anzahl gleichzeitig betrachteter unterschiedlicher Streams. Eine steigende Anzahl von Clients, die denselben Stream betrachten, erhöht die Anzahl der Anfragen an den Origin nicht, sofern ein Reverse-Proxy-Cache und ein normalisierter Cache Key vorhanden sind.

Szenario	Frequenz der Origin-Anfragen (Richtwert)
1 Client auf Stream X	MPD: 0.4 req/s, segment: 0.2 req/s
N Clients auf denselben Stream X (Cache aktiviert)	MPD: 1 req/s, segment: 0.2 req/s
N DASH-Clients im Wiederholungsmodus auf einem Stream	MPD: 1 req/s (mit `proxy_cache_lock`)
N Clients auf N unterschiedliche Streams	MPD: 0.4·N req/s, segment: 0.2·N req/s

Segmente werden global über content-addressed Namen dedupliziert; für die Deduplizierung der Manifeste zwischen Clients sorgen Session Reuse (DASH) und der Cache innerhalb der Sitzung.

6. Nginx als cachender Reverse Proxy

6.1. Basiskonfiguration

```
proxy_cache_path /var/cache/nginx/pss_segments
    levels=1:2 keys_zone=pss_segments:100m
    max_size=20g inactive=30m use_temp_path=off;

proxy_cache_path /var/cache/nginx/pss_manifests
    levels=1:2 keys_zone=pss_manifests:10m
    max_size=256m inactive=5m use_temp_path=off;

upstream pss_backend {
    server 127.0.0.1:43972;
    keepalive 64;
}

map $uri $pss_cache_key {
    "~^/h[0-9a-f]{16}(?<tail>(/[0-9]+)?/[0-9a-f]{16}\.(ts|m4s)|/sub/[0-9]+/[0-9a-f]
↪{16}\.vtt)$"  "stream:$tail";
    default  $uri;
}

server {
    listen 80;
    server_name stream.example.com;

    location ~* "^/h[0-9a-f]{16}(/[0-9]+|/sub/[0-9]+)?/([0-9a-f]+\.(ts|m4s|vtt)|init\.
↪mp4)$" {
        proxy_cache              pss_segments;
        proxy_cache_key          $pss_cache_key;
        proxy_cache_valid        200 60s;
        proxy_cache_valid        404 403 0s;
        proxy_cache_lock         on;
        proxy_cache_use_stale    updating error timeout;
        proxy_cache_revalidate   on;
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
        proxy_ignore_headers      Vary;
        add_header                X-Cache-Status $upstream_cache_status;

        proxy_pass                http://pss_backend;
        proxy_http_version        1.1;
        proxy_set_header          Connection "";
        proxy_buffering           on;
    }

    location ~* "(^/h[0-9a-f]{16}(/[0-9]+|/sub/[0-9]+)?/index\.(m3u8|mpd)$|^/
↪(hls|dash|llhls)/.*\.(m3u8|mpd)$)" {
        proxy_cache               pss_manifests;
        proxy_cache_key           $pss_cache_key;
        proxy_cache_valid         200 1s;
        proxy_cache_valid         404 403 0s;
        proxy_cache_lock          on;
        proxy_cache_lock_timeout  2s;
        proxy_cache_use_stale     updating;
        add_header                X-Cache-Status $upstream_cache_status;

        proxy_pass                http://pss_backend;
        proxy_http_version        1.1;
        proxy_set_header          Connection "";
    }

    location / {
        proxy_pass                http://pss_backend;
        proxy_http_version        1.1;
        proxy_set_header          Connection "";
        proxy_set_header          X-Forwarded-Proto $scheme;
        proxy_set_header          X-Forwarded-Host  $host;
        proxy_buffering           off;
        proxy_read_timeout        3600s;
    }
}
```

Der Origin-Port im Beispiel ist der Standardport des Streaming-HTTP-Servers (43972; HTTPS — 43982). Normalisiert werden nur die content-addressed Namen der Segmente (siehe 3.3); Manifeste, `init.mp4` und nummerierte Segmente werden unter der vollständigen URI ihrer Sitzung gecacht. Die Direktive `proxy_ignore_headers Vary` in der Location der Segmente ist gerechtfertigt: Der Server versieht OTT-Antworten mit dem Header `Vary: Origin`, und ohne sie würde der Cache nach dem Origin-Wert des Clients in Varianten aufgeteilt (siehe 7).

6.2. Zweck der Direktiven

Direktive	Zweck
`proxy_cache_lock on`	Serialisiert die Ausführung der Upstream-Anfragen bei gleichzeitigen Cache Misses auf denselben Schlüssel
`proxy_cache_use_stale updating`	Gibt parallelen Anfragen während der Cache-Aktualisierung die veraltete Kopie zurück
`proxy_cache_revalidate on`	Verwendet `If-Modified-Since` bei einem Cache Miss mit gespeicherter Kopie
`proxy_cache_valid 404 403 0s`	Verbietet das Caching von Autorisierungsfehlern und 404
`keepalive 64` im Upstream	Unterhält einen Pool persistenter Verbindungen zum Origin
`proxy_buffering on`	Für Segmente; aktiviert die Pufferung der Antwort in nginx
`proxy_buffering off`	Für den Abschnitt `/`; deaktiviert die Pufferung (Raw Streaming)

6.3. Berechnung von `max_size` des Segment-Caches

Richtwert: `bitrate × timeShiftBufferDepth × distinct_streams × 2`

Beispiel: 10 Streams × 8 Mbit/s × 40 s × 2 ≈ 800 MB. Es wird empfohlen, eine Reserve von 10x für die Schwankungen der Bitrate vorzusehen.

6.4. TLS-Terminierung

Der Server Perfect Streamer nimmt Verbindungen auf dem HTTP- und dem HTTPS-Port entgegen. Bei TLS-Terminierung auf nginx verwendet der Upstream den HTTP-Port. Die Weitergabe der Header `X-Forwarded-Proto` und `X-Forwarded-Host` ist für die korrekte Bildung absoluter URLs bei `absPath=1` zwingend erforderlich.

```
server {
    listen 443 ssl http2;
    server_name stream.example.com;

    ssl_certificate           /etc/letsencrypt/live/stream.example.com/fullchain.pem;
    ssl_certificate_key       /etc/letsencrypt/live/stream.example.com/privkey.pem;
    ssl_protocols             TLSv1.2 TLSv1.3;
    ssl_session_cache         shared:SSL:10m;
    ssl_session_timeout       1d;

    add_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;

    location ... {
        proxy_pass                http://pss_backend;
        proxy_set_header          X-Forwarded-Proto https;
        proxy_set_header          X-Forwarded-Host  $host;
        proxy_set_header          Host              $host;
        # + caching directives from 6.1
    }
}

server {
    listen 80;
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
    server_name stream.example.com;
    return 301 https://$host$request_uri;
}
```

Bei HTTPS zwischen nginx und Origin kommen die Direktiven `proxy_ssl_verify` und `proxy_ssl_trusted_certificate` zum Einsatz. Für Loopback-Verbindungen ist eine Verschlüsselung überflüssig.

6.5. Multihost

Werden mehrere `server_name` aus einem nginx-Prozess bedient, wird `$host` zur Isolierung der Inhalte in den Cache Key aufgenommen:

```
map $uri $pss_cache_key {
    "~^/h[0-9a-f]{16}(?<tail>(/[0-9]+)?/[0-9a-f]{16}\.(ts|m4s)|/sub/[0-9]+/[0-9a-f]
↪{16}\.vtt)$"  "$host:stream:$tail";
    default  "$host:$uri";
}
```

Die Größe von `keys_zone` wird mit 8000 Schlüsseln/MB berechnet. Für Multihost-Installationen mit Tausenden von Streams wird `keys_zone=...:300m` oder höher empfohlen.

7. Client-seitiges Caching

`Cache-Control: immutable` wird von aktuellen Browsern verarbeitet. Der Client-Cache liefert das Segment bei erneutem Zugriff ohne bedingte Anfrage (auch beim Rückwärtssprung innerhalb des Player-Puffers).

Service Workers können anhand des Inhalts von Cache-Control die Strategie `cache-first` anwenden. DASH-Player nutzen MSE über `SourceBuffer`; ein in den Puffer eingefügtes Segment bleibt ohne erneute HTTP-Anfrage verfügbar, bis es aus dem gleitenden Fenster herausfällt.

Für Cross-Domain-Anfragen gibt der Server `Access-Control-Allow-Origin: *` zusammen mit dem Header `Vary: Origin, Access-Control-Request-Headers` aus. Ein Caching-Proxy, der Vary beachtet, teilt die Einträge nach dem Origin-Wert des Clients in Varianten auf (Browser-Player senden Origin, Konsolen-Clients nicht), was die HIT rate senkt. Da ACAO das universelle „*“ ist, kann in der Location der Segmente gefahrlos `proxy_ignore_headers Vary` ergänzt werden (siehe 6.1).

8. Bereitstellung über ein CDN

Perfect Streamer ist mit einem CDN im Modus Pull-from-Origin kompatibel.

Origin shield. Es wird empfohlen, einen oder mehrere Shield-Knoten zwischen CDN-Edge und Origin zu platzieren, um die Anfragefrequenz am Origin bei globaler Verteilung der Clients zu senken.

Purge. Content-addressed Segmente erfordern keinen Purge. Bei Änderung der Metadaten des Streams (Codec, Auflösung) werden die Manifeste innerhalb von `max-age=1` ohne expliziten Purge aktualisiert.

Cache warming. Bei erwartetem Lastanstieg auf einen bestimmten Stream ist ein Vorwärmen des CDN von mehreren geografischen Standorten aus vor Beginn der Übertragung zulässig.

Geografische Verteilung. Segmente (`max-age=60`) eignen sich gut für geografisch verteiltes Caching. Manifeste (`max-age=1`) lassen eine Auslieferungsverzögerung von bis zu einer Sekunde zu — für Live-Übertragungen ohne niedrige Latenz akzeptabel.

9. Überwachung

9.1. `X-Cache-Status`

Hinzufügen von `add_header X-Cache-Status $upstream_cache_status;` in jeder Location mit Caching. Werte:

Wert	Beschreibung
`HIT`	Antwort aus dem Cache
`MISS`	War nicht im Cache vorhanden, vom Origin bezogen und gespeichert
`EXPIRED`	Abgelaufen, aktualisiert
`UPDATING`	Während der Aktualisierung wurde einer parallelen Anfrage die Stale-Kopie geliefert
`STALE`	`use_stale` hat eine abgelaufene Kopie geliefert (Origin nicht erreichbar)
`REVALIDATED`	Der Origin hat 304 Not Modified zurückgegeben
`BYPASS`	`proxy_cache_bypass` hat ausgelöst

9.2. Format des Access-Logs

```
log_format pss_cache '$remote_addr $status $request_method "$request" '
                     '$body_bytes_sent rt=$request_time ut=$upstream_response_time '
                     'cache=$upstream_cache_status key=$pss_cache_key';

server {
    access_log /var/log/nginx/pss.log pss_cache;
}
```

9.3. Metriken

Das Modul `nginx-vts` exportiert Metriken pro Zone im Prometheus-Format:

```
GET /status/format/prometheus
```

Empfohlene Schwellenwerte für Alarme:

Metrik	Schwellenwert	Mögliche Ursache
Segment HIT rate	< 90% über 5 Minuten	Die Normalisierung des Cache Key ist verletzt; zu kleiner `max_size`
Manifest MISS rate	> 50% über 1 Minute	`proxy_cache_lock` serialisiert die Anfragen nicht
Upstream response time p95	> 500 ms über 1 Minute	Überlastung des Origin
Cache zone fill	> 90% über 10 Minuten	Annäherung an `max_size`, LRU-Eviction steht bevor

Die Auslieferungsqualität auf Knotenseite ist an den Sitzungsmetriken auf dem Bildschirm „Clients“ (*Clients*) ersichtlich.

10. Diagnose

Symptom	Wahrscheinliche Ursache	Lösung
Segment HIT rate niedrig	`Vary: Origin` teilt den Cache in Varianten auf; verletzte Normalisierung in `map`	`proxy_ignore_headers Vary` in der Location der Segmente ergänzen; Regex in der Direktive `map` prüfen
404 auf Segmenten nach dem Verlassen des Fensters	Gecachte 404 bei einem Segment, das aus dem Sliding Window herausgefallen ist	`proxy_cache_valid 404 0s` in der Location segments ergänzen
Verzögerung des Playback-Starts 2–5 s	`proxy_cache_lock_timeout` überschreitet die Target Latency	Auf 1–2 s senken; `proxy_cache_use_stale updating` aktivieren
Manifest wird nicht aktualisiert	`proxy_ignore_headers Cache-Control` ist gesetzt, und `proxy_cache_valid` mit großem TTL ist wirksam	`proxy_cache_valid 200 1s` explizit angeben oder das Ignorieren der Header entfernen
Anstieg von TIME_WAIT am Upstream	`keepalive` fehlt im Upstream-Block	`keepalive 64`, `proxy_http_version 1.1`, `proxy_set_header Connection ""` ergänzen
403 auf `/dash/.../<segment>.m4s` von Konsolen-Clients	Der Client löst relative URLs gegen die Pre-Redirect-URL auf	Der Server gibt `<BaseURL>` mit absolutem Pfad aus; im aktuellen Build kompatibel
Ruckler, häufiges Rebuffering bei entfernten Clients	Niedriger effektiver TCP-Durchsatz durch Slow Start und Idle Restart bei großen RTT (300 ms und mehr)	Tuning des Linux-Netzwerkstacks auf dem Origin: siehe 10.1

10.1. TCP-Tuning des Origin für High-RTT-Clients

Das Problem tritt bei Clients mit großem RTT zum Origin auf (zum Beispiel 300 ms und mehr), wenn die Bitrate des Streams nahe an der Kapazität des Kanals liegt. Symptome im Player sind häufiges Rebuffering und Abbrüche; auf dem Server wirkt der Client dabei normal, Fehler treten nicht auf.

Ursache:

- **TCP slow start.** Jede neue TCP-Verbindung beginnt mit einem Congestion Window von etwa 14 KB und vergrößert es über mehrere RTT. Bei einem RTT von 300 ms dauert das Erreichen des vollen Fensters 2–3 Sekunden. In dieser Zeit wird ein HLS/DASH-Segment von 5 s Dauer (4–6 MB) merklich langsamer als in Echtzeit heruntergeladen.
- **TCP idle restart.** Zwischen den Segmentanfragen legt der Client im HLS-Pull-Modell eine Pause von 4–5 s ein. Standardmäßig setzt der Linux-Kernel nach einer solchen Pause das Congestion Window der Verbindung auf das Initial cwnd zurück (Verhalten `net.ipv4.tcp_slow_start_after_idle=1`). Dadurch beginnt die Keep-alive-Verbindung beim nächsten GET die Übertragung erneut mit Slow Start — selbst in einer bereits eingelaufenen Sitzung.

Erschwerend kommt hinzu, dass das standardmäßige Congestion Control CUBIC lange RTT und Paketverluste auf zwischengeschalteten Netzabschnitten schlecht verkraftet.

Die Lösung sind zwei sysctl-Parameter auf dem Origin:

```
# Keep congestion window across idle pauses inside keep-alive sessions.
sysctl -w net.ipv4.tcp_slow_start_after_idle=0

# Use BBR instead of CUBIC: better behaviour on long-RTT paths
# with mild packet loss; paces sending instead of bursting.
modprobe tcp_bbr
sysctl -w net.ipv4.tcp_congestion_control=bbr
```

Für die dauerhafte Anwendung:

```
cat > /etc/sysctl.d/99-pss-net.conf <<EOF
net.ipv4.tcp_slow_start_after_idle = 0
net.ipv4.tcp_congestion_control    = bbr
EOF
sysctl --system
```

Die Hauptwirkung erzielt der erste Parameter (`tcp_slow_start_after_idle=0`) — er beseitigt unmittelbar den erneuten Slow Start zwischen den Segmentanfragen innerhalb derselben Keep-alive-Verbindung. Der zweite (BBR) bringt zusätzliche Stabilität und gilt für alle neuen Verbindungen. Das Tuning erfordert keinen Neustart von Perfect Streamer. Für HTTP/3-Clients (QUIC) gelten diese Parameter nicht: QUIC arbeitet über UDP mit eigener Überlastungssteuerung.

11. Sicherheit

11.1. Session URL

Eine URL im Format `/h<sess>/...` erfüllt die Funktion eines Sitzungstokens — sie erfordert keine erneute Authentifizierung. Die Lebensdauer ist durch ein Inaktivitäts-Timeout von 60 Sekunden begrenzt; bleibt Aktivität aus, wird die Sitzung automatisch gelöscht.

Anforderungen:

- HTTPS für alle OTT-Pfade (`/hls/`, `/dash/`, `/h<sess>/`) im Produktivbetrieb;
- Die Session ID im Header `Location` der 302-Antwort wird nicht gecacht (`no-cache, no-store`).

11.2. Rate limiting

```
limit_req_zone $binary_remote_addr zone=dash_top:10m  rate=5r/s;
limit_req_zone $binary_remote_addr zone=hls_top:10m   rate=5r/s;
limit_req_zone $binary_remote_addr zone=llhls_top:10m rate=5r/s;

server {
    location /dash/ {
        limit_req zone=dash_top burst=20 nodelay;
        proxy_pass http://pss_backend;
    }
    location /hls/ {
        limit_req zone=hls_top burst=20 nodelay;
        proxy_pass http://pss_backend;
    }
    location /llhls/ {
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
        limit_req zone=llhls_top burst=20 nodelay;
        proxy_pass http://pss_backend;
    }
}
```

Die Session-URL (`/h<sess>/`) erfordert kein Rate Limiting — die Verarbeitung ist günstig, die Antworten werden gecacht.

11.3. Caching von Fehlerantworten

```
proxy_cache_valid 200 60s;
proxy_cache_valid 301 302 0s;
proxy_cache_valid 404 403 0s;
proxy_cache_valid any 1s;
```

Verbietet das Caching von Redirects (eindeutige sess in Location) sowie von Antworten mit Autorisierungsfehlern oder fehlender Ressource.

11.4. Beschränkung des Netzwerkzugriffs auf den Origin

Der Port des Streaming-HTTP-Servers (standardmäßig 43972; 43982 für HTTPS) muss für externen Verkehr gesperrt sein. Zulässige Konfigurationen:

1. Perfect Streamer an `127.0.0.1` binden (bei lokalem nginx)
2. Firewall-Regel:

```
iptables -A INPUT -p tcp -m multiport --dports 43972,43982 ! -s 10.0.0.0/8 -j DROP
```

12. Integration mit Middleware

12.1. Modell prefix-login

Perfect Streamer unterstützt die Delegierung der Benutzeridentifikation an ein Middleware-System über den Mechanismus prefix-login — eine eingebaute Alternative zur vollständigen Integration mit einem externen Billing-System (Konten, die von einem externen System geprüft werden, werden auf dem Bildschirm „Benutzer / Logins“ auf der Registerkarte „Extern (Billing)“ konfiguriert, *Benutzer / Logins*).

Beim lokalen Login wird das Präfix-Merkmal aktiviert, woraufhin der Server URLs mit einem Login der Form `<prefix><Abonnenten-ID>` akzeptiert:

```
/dash/test1/sub42/xxx/index.mpd
/hls/test1/sub43/xxx/index.m3u8
```

Hier ist `sub` das konfigurierte Login-Präfix, und `42`/`43` sind die Abonnentenbezeichner der Middleware; das Passwort ist gemeinsam.

12.2. Statistik nach Abonnenten

In der Liste der Clients trägt jede Sitzung sowohl den konfigurierten Login als auch den tatsächlichen URL-Login des Abonnenten (match-login), über den die Middleware die Sitzungen ihren eigenen Teilnehmern zuordnet. Die Daten sind auf dem Bildschirm „Clients“ (*Clients*) sichtbar.

12.3. Einschränkungen von prefix-login

- **Gemeinsames Passwort.** Alle Abonnenten eines prefix-Pools verwenden denselben Passwortwert. Eine Kompromittierung des Passworts gewährt Zugriff auf jedes `<prefix><Zeichenfolge>`.
- **Zugriffsgranularität.** Die Liste der erlaubten Streams gilt für den gesamten prefix-Pool. Zugriff auf der Ebene einzelner Abonnenten bietet die Integration mit einem externen Billing-System.
- **Passwortrotation.** Eine Änderung des Passworts trennt alle aktiven Abonnenten. Für einen schrittweisen Wechsel ist die vorübergehende Nutzung zweier prefix-Logins erforderlich.

1.9.2 Anbindung externer Überwachungssysteme

pss-metrics — universeller Metrik-Exporter für Perfect Streamer

Ein einzelnes CLI-Skript in Python 3, das Statistiken über die HTTP-API des PSS-Webservers abruft und daraus Ausgaben für die gängigsten Überwachungssysteme erzeugt:

- Zabbix (UserParameter, Low-Level Discovery, zabbix_sender trapper)
- Prometheus (Text-Expositionsformat für den textfile collector)
- InfluxDB / Telegraf (line protocol oder JSON für den exec input)
- Generisches JSON für beliebige Skripte und Zustandsprüfungen im Nagios-Stil

Der Exporter ist eine einzelne, eigenständige Datei ohne Fremdabhängigkeiten; er verwendet ausschließlich die Standardbibliothek von Python 3.6+ (*urllib*, *xml.etree*, *json*, *argparse*).

Dateien

```
pss-metrics.py                    main CLI (executable)
userparameter_pss.conf.example    UserParameter template for Zabbix
```

Installation

Der Exporter wird als */opt/pss/monitoring/pss-metrics.py* ausgeliefert. Stellen Sie sicher, dass Python 3.6 oder neuer installiert ist:

```
# RHEL / Rocky / AlmaLinux
yum install -y python3

# Debian / Ubuntu
apt-get install -y python3
```

Zusätzliche Pakete sind nicht erforderlich.

Konfiguration

Standardmäßig verbindet sich *pss-metrics* über die Loopback-Adresse mit dem Knoten und ermittelt den Port des Webservers aus */opt/pss/config/pss.json* (oder */opt/pss/config/pss_default.json*); lässt sich der Port aus keiner der beiden Dateien lesen, wird *http://127.0.0.1:43971* verwendet. Die Parameter lassen sich über Umgebungsvariablen, die Datei */etc/pss-metrics.conf* (Format *Schlüssel=Wert*) oder Kommandozeilenflags überschreiben. Priorität: CLI > env > Datei > Standardwerte. Ausnahme ist *PSS_URL*: Ein Wert aus der Datei kann durch die automatische Porterkennung überschrieben werden; setzen Sie die URL daher über eine Umgebungsvariable oder ein Kommandozeilenflag.

Unterstützte Variablen:

```
PSS_URL          full URL, e.g. http://10.0.0.1:8808          (auto by default)
PSS_USER         web server login (if authorization is enabled)
PSS_PASS         web server password
PSS_TIMEOUT      HTTP timeout, in seconds                      (default 5)
PSS_CACHE_DIR    cache directory                               (default /run/pss-
↪metrics)
PSS_CACHE_TTL    cache TTL, in seconds                         (default 10)
PSS_CA_BUNDLE    path to CA bundle for HTTPS
PSS_INSECURE     1 — disable TLS certificate verification
PSS_VERBOSE      1 — log requests to stderr
```

Cache: Jeder Lauf führt innerhalb des TTL-Fensters höchstens ein HTTP GET pro Endpunkt aus. Bei TTL=10 s ergeben selbst Hunderte von UserParameter-Prüfungen pro Minute nur etwa 6 HTTP-Anfragen pro Minute an jeden API-Endpunkt.

Schnellstart

Zustandsprüfung (exit codes im Nagios-Stil):

```
pss-metrics.py health
# OK: total=42 running=15 stopped=27 unhealthy=0 version=<Version>
```

Zabbix-LLD-Erkennung:

```
pss-metrics.py discover streams
pss-metrics.py discover inputs --running-only
pss-metrics.py discover outputs
```

Abruf einer einzelnen Metrik (zur Verwendung in einem Zabbix-UserParameter):

```
pss-metrics.py get summary.running
pss-metrics.py get stream.10031.bitrate
pss-metrics.py get input.10031.1.speed1
pss-metrics.py get output.10031.1.speed
pss-metrics.py get sysmon.cpu.self-usage
pss-metrics.py get server.server-version
```

Vollständiger Export:

```
pss-metrics.py dump --format=json
pss-metrics.py dump --format=prometheus
pss-metrics.py dump --format=influx
pss-metrics.py dump --format=zabbix-trapper --zabbix-host=streamer-01
```

Metrikpfade

pss-metrics get erwartet einen durch Punkte getrennten Pfad. Eine leere Ausgabe bedeutet „kein Wert“ (die Metrik existiert beispielsweise nur für laufende Streams).

```
server.<attr>                 e.g. server.server-version, server.uptime
summary.<key>                 total | running | stopped | unhealthy |
                              input_bitrate_kbps | output_bitrate_kbps
sysmon.cpu.<attr>             self-usage | total-usage | cores
sysmon.memory.<attr>          self-usage-kb | available-kb | total-kb
sysmon.netbw.<iface>.<attr>   rx-bw | tx-bw  (interface name as in XML)
stream.<id>.<attr>            any <stream> attribute
input.<sid>.<iid>.<attr>      any <input> attribute
output.<sid>.<oid>.<attr>     any <output> attribute
```

Nützliche Attribute je Stream (aus /data/stream/detail):

```
stream:  state, state-str, bitrate, thread-usage, mpts
input:   speed1, recv-bytes, recv-packets, recv-err,
         stat-disc, stat-disc1, stat-scrambled, stat-scrambled1,
         health-state-good, health-status, check-status
output:  speed, sent-bytes, sent-packets, sent-err, uri, type
```

Integration mit Zabbix

Es werden zwei Szenarien unterstützt — wählen Sie das für Ihre Umgebung passende aus.

1) Statischer UserParameter + LLD (Zabbix agent v1 / v2)

   Kopieren Sie *userparameter_pss.conf.example* nach */etc/zabbix/zabbix_agentd.d/pss.conf*, starten Sie *zabbix-agent* neu und importieren Sie auf dem Server eine Vorlage mit LLD-Prototypen, die die Schlüssel *pss.discover* verwenden. Beispiel für Zuordnungen:

   ```
   UserParameter=pss.discover[*],/opt/pss/monitoring/pss-metrics.py discover $1
   UserParameter=pss.get[*],/opt/pss/monitoring/pss-metrics.py get $1
   UserParameter=pss.health,/opt/pss/monitoring/pss-metrics.py health
   ```

   Auf dem Zabbix-Server:

   ```
   Discovery rule key:        pss.discover[streams]
   Item prototype keys:       pss.get[input.{#STREAM_ID}.1.speed1]
                              pss.get[stream.{#STREAM_ID}.bitrate]
                              pss.get[summary.unhealthy]
   ```

2) Trapper (push) über zabbix_sender

   Führen Sie den Aufruf zeitgesteuert aus (cron / systemd) und leiten Sie die Ausgabe in eine Pipe:

   ```
   /opt/pss/monitoring/pss-metrics.py dump --format=zabbix-trapper \
       --zabbix-host="$(hostname)" \
     | zabbix_sender -z zabbix.example.com -i -
   ```

Integration mit Prometheus

Zwei Varianten.

a) Textfile collector (empfohlen für One-shot-Umgebungen).

   Führen Sie den Export periodisch über einen systemd-Timer oder cron aus:

```
*/1 * * * * /opt/pss/monitoring/pss-metrics.py dump --format=prometheus \
    > /var/lib/node_exporter/textfile_collector/pss.prom.$$ \
    && mv /var/lib/node_exporter/textfile_collector/pss.prom.$$ \
          /var/lib/node_exporter/textfile_collector/pss.prom
```

   node_exporter liefert die Datei über *–collector.textfile.directory* aus.

b) Direktes Scraping über einen kleinen Wrapper (zum Beispiel *socat + pss-metrics dump*) oder über einen beliebigen HTTP-Proxy eines Drittanbieters Ihrer Wahl.

Integration mit Telegraf / InfluxDB

Telegraf *inputs.exec*:

```
[[inputs.exec]]
  commands = ["/opt/pss/monitoring/pss-metrics.py dump --format=influx"]
  interval = "10s"
  timeout  = "5s"
  data_format = "influx"
```

Für den JSON-Parser verwenden Sie *–format=json* und konfigurieren *data_format = „json“* unter Angabe der Feldpfade.

HTTPS und Authentifizierung

Der Exporter ist für den Betrieb auf dem Knoten selbst ausgelegt: Anfragen von der lokalen Adresse nimmt der PSS-Webserver ohne Autorisierung entgegen. Die Variablen *PSS_USER*/*PSS_PASS* werden als HTTP Basic übergeben — sie sind nützlich, wenn der Zugriff auf die API über einen vorgelagerten Proxy mit Basic-Autorisierung veröffentlicht wird; die eigene Digest-Autorisierung des PSS-Webservers durchläuft der Exporter nicht.

Zugriff über HTTPS:

```
PSS_URL=https://streamer.example.com:43981 pss-metrics.py health
```

Selbstsignierte Zertifikate: Setzen Sie *PSS_INSECURE=1* (nicht empfohlen) oder geben Sie *PSS_CA_BUNDLE=/path/to/ca.pem* an.

Rückgabecodes

pss-metrics folgt der Nagios-Konvention:

```
0  OK
1  WARNING       (e.g. running streams report unhealthy)
2  CRITICAL      (PSS is unreachable)
3  UNKNOWN       (invalid arguments / internal error)
```

get gibt eine leere Zeile aus und endet mit dem Code 0, wenn die angeforderte Entität nicht vorhanden ist — das entspricht der Erwartung von Zabbix, dass ein leerer Wert als „NOT_SUPPORTED“ und nicht als Ausfall des Agenten gewertet wird.

Diagnose

```
pss-metrics.py -v health        # log every HTTP request to stderr
pss-metrics.py --cache-ttl=0 …  # bypass cache while debugging
rm -rf /run/pss-metrics         # purge cache
```

1.9.3 Optimierung der Knotenleistung

Empfehlungen für Installationen mit einer großen Anzahl von Streams, bei denen CPU oder Arbeitsspeicher knapp werden.

CPU-Last

- Aktivieren Sie die Filterung und Modifikation von MPEG-TS (*MPEG-TS-Filterung*) nur dort, wo sie tatsächlich benötigt werden: Standardmäßig sind alle Filter deaktiviert, und jeder aktivierte Filter fügt dem heißen Pfad des Streams zusätzliche Verarbeitung hinzu. Dasselbe gilt für die Tiefenanalysen (*Vertiefte Analysen*).
- Das Mosaik dekodiert die Bilder aller Streams. Es lässt sich vollständig in den Servereinstellungen deaktivieren, selektiv bei einzelnen Streams („Mosaik erzeugen“) oder seltener aktualisieren, indem „Prüfintervall (s)“ erhöht wird. Beachten Sie: Das Prüfintervall ist gemeinsam — es steuert ebenso die erneute Prüfung der Quellen und die Rückkehr auf den Haupteingang (*Quellenredundanz*), sodass seine Erhöhung auch die Reaktion der Redundanz verlangsamt.
- Die aktuelle Auslastung nach Streams ist im Statistikfenster (Abschnitt „Profiler“) und auf dem Bildschirm „System-Monitor“ (*System-Monitor*) ersichtlich.

Arbeitsspeicher

Perfect Streamer gibt nicht genutzten Arbeitsspeicher regelmäßig an das Betriebssystem zurück, und in der mitgelieferten systemd-Unit ist die Anzahl paralleler Speicherzuteilungs-Arenen standardmäßig begrenzt (Umgebungsvariable `MALLOC_ARENA_MAX`). Dies bremst das allmähliche Anwachsen des residenten Speichers beim Betrieb mit Dutzenden von Streams; das Anwachsen an sich ist keine Folge eines Speicherlecks. Der Wert muss in der Regel nicht manuell geändert werden.

Datenbanken für Statistik und EPG

Fehler durch Überlauf der Warteschlange der Statistikdatenbank oder der EPG-Datenbank treten bei unzureichender Geschwindigkeit des Datenträgers auf: Die Datenbanken liegen auf einem langsamen Medium, oder das System ist überlastet.

Der Speicherort der Datenbanken wird durch den Parameter `data-dir` der Konfigurationsdatei `pss.properties` festgelegt (standardmäßig das Konfigurationsverzeichnis). Mögliche Lösungen:

1. Die Datenbanken mit dem Parameter `data-dir` auf einen schnellen Datenträger (SSD) verlagern. Die Variante mit `/tmp` nutzt den Arbeitsspeicher des Systems: Sie erfordert eine Abschätzung des freien Speichers und eine Anpassung der Aufbewahrungsdauer der Statistik; beim Neustart gehen die Daten verloren.
2. Die Detaillierung der Statistik verringern — Parameter `dbstat-detail` (Intervall in Sekunden, standardmäßig 5; zulässig bis 30).

1.9.4 Intel-VPL-Transcoder: unterstützte Hardware

Welche Intel-Hardware für die Hardware-Transcodierung geeignet ist, welche Codecs auf den verschiedenen Grafikgenerationen verfügbar sind und wie sich prüfen lässt, dass die Hardware erkannt wurde. Die Installation der Transcoder-Pakete ist im Abschnitt *Transcoder* beschrieben, die Konfiguration des Transcoders selbst — in *Transcoder*.

Kurz gefasst: erforderlich ist eine integrierte oder dedizierte Intel-Grafik ab Gen9. Der Transcoder ermittelt, welche Intel-Laufzeitumgebung auf dem Rechner vorhanden ist, und bindet sich an sie an; ein und dieselbe ausführbare Datei `tcivpl` arbeitet sowohl auf neuer als auch auf alter Hardware.

Zwei Intel-Laufzeitumgebungen

Laufzeitumgebung	Hardware
oneVPL (VPL v2)	Gen12 und neuer: Tiger Lake, Rocket Lake, Alder Lake (einschließlich Alder Lake-N: N100, N305), Raptor Lake, Arc
Media SDK v1 (MSDK v1)	Gen9–Gen11: Skylake, Kaby Lake, Coffee Lake, Comet Lake, Ice Lake

Die Auswahl der Laufzeitumgebung erfolgt automatisch und erfordert keine separate Konfiguration: der VPL-Dispatcher listet die Implementierungen auf, die für das ausgewählte Gerät `/dev/dri/renderD<N>` gefunden wurden, und der Transcoder nimmt die erste für den Hardwarebetrieb geeignete. Eine eigene Regel „VPL v2 bevorzugen“ gibt es im Transcoder nicht — für Gen9–Gen11 existiert die neue Laufzeitumgebung schlicht nicht, dort findet sich daher immer Media SDK v1, während auf Gen12 und neuer üblicherweise nur oneVPL installiert ist.

Beachten Sie dies bei der Auswahl der Hardware: die Installationsverfahren des Transcoders (*Transcoder*) installieren ausschließlich die Laufzeitumgebung VPL v2, sind also auf Gen12 und neuer ausgelegt. Für Gen9–Gen11 wird die Laufzeitumgebung Media SDK v1 separat installiert, und in neuen Distributionen kann sie fehlen (siehe unten).

Codecs

Laufzeitumgebung	Dekodierung	Codierung
VPL v2 (Gen12+)	H.264, HEVC, MPEG-2	H.264, HEVC, MPEG-2
MSDK v1 (Gen9–Gen11)	H.264, MPEG-2	H.264, MPEG-2

Was fehlt:

- **HEVC wird auf MSDK v1 nicht unterstützt** — weder Dekodierung noch Codierung: in Media SDK v1 wird HEVC über eine Plugin-Schnittstelle eingebunden, die der oneVPL-Dispatcher nicht bereitstellt. Für HEVC ist ein Rechner mit Gen12 oder neuer erforderlich. Bei einem Versuch schreibt der Transcoder eine eindeutige Fehlermeldung — `HEVC decode is not supported on the Intel Media SDK v1 runtime` oder `HEVC encode is not supported on the Intel Media SDK v1 runtime`.
- AV1, VP9 und VC-1 werden von keiner der beiden Laufzeitumgebungen unterstützt — im Rundfunk werden sie nicht benötigt.

Die drei aufgeführten Codecs sind das Maximum, mit dem der Transcoder arbeiten kann. Auf VPL v2 legt er keine eigenen Beschränkungen auf, daher bestimmt die Laufzeitumgebung den tatsächlichen Codec-Satz eines konkreten Adapters; was auf dem Rechner verfügbar ist, zeigt `tcivpl --hw-list` (siehe unten). Die allgemeine, backend-unabhängige Codec-Matrix des Transcoders ist in *Backends und Codecs* aufgeführt.

Eingeschränkte Atom-Hardware: nicht unterstützt

Plattformen mit Atom-Kernen vor Gen12 — Apollo Lake, Gemini Lake, Elkhart Lake, Jasper Lake (Celeron und Pentium der Serien J und N: J4125, J6412, N5105 und ähnliche) — sind für die Transcodierung nicht geeignet.

Ihre Media-Engine ist eingeschränkt: eine vollwertige Hardware-Codierung fehlt, und eine MPEG-2-Codierung gibt es überhaupt nicht. Die Dekodierung kann auf solchen Chips funktionieren, doch es fehlt die Möglichkeit zu transcodieren — verwenden Sie den Software-Transcoder (Paket **pstreamer-tcsw**) oder andere Hardware. Der Transcoder sortiert solche Hardware nicht im Voraus aus: er startet und bricht bei der Initialisierung des Encoders mit einem Fehler ab.

Eine wichtige Ausnahme: **Alder Lake-N (N100, N305 und ähnliche) sind ebenfalls Atom-Kerne, ihre Grafik ist jedoch eine vollwertige Gen12, und sie werden unterstützt.**

Gesondert zu beachten: alte Atom-Prozessoren besitzen kein AVX, während ein Teil der fertigen Intel-Pakete mit AVX gebaut ist. In diesem Fall stürzt der Transcoder-Prozess mit `Illegal instruction` ab — die Instruktionen führt die in den Prozess geladene Bibliothek des Intel-Treibers oder der Intel-Laufzeitumgebung aus, nicht der Code des Transcoders selbst.

Was installiert sein muss

Komponente	RHEL / AlmaLinux / Rocky	Debian / Ubuntu
VA-API-Treiber **iHD** (erforderlich)	`intel-media-driver`	`intel-media-va-driver-non-free`
Laufzeitumgebung **VPL v2** (Gen12+)	`intel-vpl-gpu-rt`	`libmfxgen1`
Laufzeitumgebung **MSDK v1** (Gen9–Gen11)	`intel-mediasdk`	`libmfx1`
VA-API, DRM	`libva`, `libdrm`	`libva2`, `libva-drm2`, `libdrm2`

Es wird genau eine Laufzeitumgebung benötigt — diejenige, die zur Hardware passt. Der iHD-Treiber ist in jedem Fall erforderlich: auf dem alten `i965` gelingt die VA-API-Initialisierung zwar, der Decoder kommt jedoch nicht hoch, und eine gesonderte Meldung darüber gibt es nicht — der Treiber ist nur im Protokoll beim Start des Streams sichtbar. Unter Debian und Ubuntu wählen Sie genau die Variante `intel-media-va-driver-non-free`: in `intel-media-va-driver` ist ein Teil der Codecs entfernt.

Die Paketnamen hängen davon ab, woher sie installiert werden. Die angegebenen Debian-Namen stammen aus dem Intel-Graphics-Repository, das gemäß dem Installationsverfahren hinzugefügt wird (*Ubuntu 22/24*); im eigenen Repository von Ubuntu heißt dieselbe Laufzeitumgebung VPL v2 `libmfx-gen1.2`, und die Versionen sind dort deutlich älter.

Der Intel-VPL-Dispatcher (in den Distributionen — `libvpl2`) wird als Bestandteil des Transcoder-Pakets ausgeliefert: er wird nach `/opt/pss/lib/libvpl.so.2` installiert, der Pfad wird im Verzeichnis `/etc/ld.so.conf.d/` eingetragen. Der Dispatcher wählt jedoch nur die Laufzeitumgebung aus — ist auf dem Rechner keine passende Laufzeitumgebung vorhanden, fehlt die Möglichkeit zu transcodieren.

Die Laufzeitumgebung MSDK v1 wurde aus den neuen Distributionen entfernt: `intel-mediasdk` ist in EPEL 9 vorhanden, in EPEL 10 jedoch nicht; `libmfx1` ist in Debian 12 und Ubuntu 24.04 vorhanden, in den folgenden Ausgaben aber nicht mehr. Auf solchen Systemen muss die Laufzeitumgebung für Hardware der Generationen Gen9–Gen11 separat installiert werden.

Das Transcoder-Paket zieht weder den Treiber noch die Laufzeitumgebung nach sich, daher bedeutet eine erfolgreiche Installation für sich genommen noch nicht, dass die Hardware-Transcodierung funktioniert — das wird separat geprüft. Beachten Sie außerdem, dass der Dienstbenutzer Zugriff auf die Geräte `/dev/dri` haben muss (*Ubuntu 22/24*).

Wie geprüft wird

```
/usr/local/bin/tcivpl --hw-list
```

Der Befehl gibt XML mit der Liste der Implementierungen aus, die der VPL-Dispatcher gefunden hat — in einer Zeile, ohne Umbrüche. Für jede sind der Name der Laufzeitumgebung (`device-name`), die API-Version, die Codec-Listen von Decoder und Encoder, der Adapter und das Merkmal der Eignung für den Hardwarebetrieb angegeben. Beispiel einer Ausgabe auf Gen9–Gen11 (zur besseren Lesbarkeit mit Umbrüchen dargestellt):

```
<vpl mfx-version="2.15">
  <device id="0" device-name="mfxhw64" api-version="1.35" legacy="1"
          dec-codecs="h264,mpeg2" enc-codecs="h264,mpeg2"
          vendor-id="8086" device-id="4555" drm-render="128"
          supported="1"/>
</vpl>
```

Wie die Ausgabe zu lesen ist:

- `legacy="1"` — die Implementierung meldet eine API-Version unter 2.0, es handelt sich also um die alte Laufzeitumgebung Media SDK v1; ohne dieses Merkmal ist es oneVPL 2.x. Darauf und auf `api-version` ist in erster Linie abzustellen: der Wert von `device-name` kommt unverändert vom Dispatcher (in der Praxis — `mfxhw64` für Media SDK v1 und `mfx-gen` für oneVPL).
- `supported="1"` — die Implementierung ist für den Hardwarebetrieb geeignet. Ein Eintrag ohne dieses Merkmal ist gleichbedeutend mit seinem Fehlen.
- Für Media SDK v1 lautet die Codec-Liste stets `h264,mpeg2` — das ist der Satz, den der Transcoder auf dieser Laufzeitumgebung unterstützt, und nicht das Ergebnis einer Abfrage der Hardware. Für oneVPL werden die Listen aus dem übernommen, was die Laufzeitumgebung meldet.

- `mfx-version` — die API-Version der Header, mit denen der Transcoder gebaut wurde, und nicht die Version der installierten Laufzeitumgebung.

Eine leere Liste (ein Element <vpl> ohne ein einziges <device>) bedeutet, dass es auf diesem Rechner keine Hardware-Transcodierung geben wird: die Laufzeitumgebung oder der Treiber ist nicht installiert, die GPU stammt nicht von Intel, oder es besteht kein Zugriff auf `/dev/dri`. Die Meldung `No vpl libraries found` ist ein gesonderter Fall: die Dispatcher-Bibliothek selbst wurde nicht gefunden.

Der Befehl listet die Implementierungen nur auf und wählt nichts aus. Welche von ihnen verwendet wird, entscheidet sich beim Start des Streams — anhand des in den Einstellungen des Transcoders angegebenen Geräts. Beim Start des Streams schreibt der Transcoder zwei Zeilen in das Protokoll: die ausgewählte Laufzeitumgebung mit der API-Version sowie die VA-API-Version zusammen mit dem Namen des Treibers. An der zweiten Zeile ist zu erkennen, dass tatsächlich der Treiber iHD arbeitet und nicht `i965`.

Die erkannten Transcodierungs-Backends, deren Versionen und die gefundenen Geräte werden ebenfalls auf dem Bildschirm „Über“ (*Über das System*) angezeigt, die GPU-Auslastung — auf dem Bildschirm „System-Monitor“ (*System-Monitor*).

1.10 FAQ

Antworten auf die Fragen, die im Betrieb am häufigsten auftreten: Kompatibilität von SRT mit Software von Drittanbietern und Multicast-Empfang bei hohen Bitraten.

1.10.1 SRT: Autorisierung mit Login und Passwort in Software von Drittanbietern

Zwischen zwei Perfect-Streamer-Knoten wird die SRT-Autorisierung regulär über die Felder von Eingang und Ausgang konfiguriert (*SRT*). Das Zusammenspiel mit anderer Software ist nicht garantiert: Der Parameter stream ID ist nicht standardisiert und in verschiedener Software unterschiedlich umgesetzt.

Der Kompatibilitätsmechanismus ist überall derselbe: Die empfangende Seite nimmt die stream ID des sich verbindenden Clients und sucht damit ein Konto in der Liste der lokalen Logins (*Benutzer: hinzufügen und bearbeiten*). Übereinstimmen muss das Feld „Login“ — vollständig mit dem Wert der stream ID. Zum Beispiel bei der URL

```
srt://Stream_IP:port?streamid=!#::u=1234567890,password=1234567890
```

wird in das Feld „Login“ Folgendes eingetragen

```
!#::u=1234567890,password=1234567890
```

Die Syntax der stream ID ist für Perfect Streamer ohne Bedeutung: Die Zeichenkette wird nicht in Teile zerlegt, sondern als Ganzes verglichen. Es gibt drei Ausnahmen — Zeichen, die der Knoten selbst auswertet:

- `|` — Trennzeichen: Der Teil der Zeichenkette vor dem ersten solchen Zeichen gilt als Login, der Rest als Passwort des Kontos;
- `*` — es ist keine stream ID gesetzt; der Client wird über die Adresse autorisiert, also über ein Konto mit dem Merkmal „IP-Adress-Konto“;
- `@` am Anfang der Zeichenkette ist der dienstinternen Autorisierung zwischen den Knoten einer Meshwork-Domain vorbehalten.

Perfect Streamer begrenzt die Länge der stream ID nicht — die Begrenzung kommt von der SRT-Bibliothek und beträgt 512 Byte einschließlich des abschließenden Nullzeichens. Werte, die länger als etwa 500

Zeichen sind, sollten nicht übertragen werden: Eine zu lange Zeichenkette bricht entweder die Verbindung ab oder wird leer und ohne Fehlermeldung übertragen.

Verschiedene Software bildet die stream ID auf ihre eigene Weise; funktioniert der Empfang über die URL nicht, aktivieren Sie daher am SRT-Ausgang die erweiterte Einstellung „Trace“. Im Protokoll des Streams, der die Verbindung annimmt, erscheinen dann die Adresse des Clients, die empfangene stream ID und der Grund der Ablehnung — an dieser Zeile ist zu erkennen, was die Gegenseite tatsächlich gesendet hat. Anschließend lässt sich die stream ID korrigieren: zum Beispiel überzählige Zeichen am Anfang der Zeichenkette entfernen, die die Übergabe des Werts verhindern.

1.10.2 Empfehlungen für die Arbeit mit UDP-Multicast

Aufgabe. UDP-Multicast mit einer Gesamtbitrate von mehreren hundert Mbit/s (1 Gbit/s und mehr) auf einem einzigen Server stabil empfangen.

Problem. Beim Empfang über Netzwerkkarten mit RJ-45-Schnittstelle setzen mit steigendem Verkehr oberhalb von einigen hundert Megabit zunehmende Verluste von Multicast-Paketen ein. Eine Feinabstimmung der Karte hilft nicht — in aktuellen Versionen der Betriebssysteme werden bereits die optimalen Werte verwendet. Karten mit besseren Chips beseitigen das Problem ebenso wenig, insbesondere jenseits von 500 Mbit/s. Auch das Bonding zweier Karten hilft nicht: Eine Multicast-Gruppe trifft immer auf einer physischen Leitung und in einer Empfangswarteschlange ein, aggregiert wird daher die Gesamtbandbreite und nicht die Reserve eines einzelnen Streams.

Lösung. Die Betriebserfahrung zeigt, dass für den Empfang und die Übertragung von UDP-Multicast 10-Gigabit-Netzwerkkarten mit SFP+-Schnittstelle eingesetzt werden sollten. Eine günstige Karte der Klasse Intel X520-DA1/2 (Intel 82599ES) genügt — in unseren Tests beseitigt sie die Paketverluste bei Multicast-Verkehr oberhalb von 1 Gbit/s. Zusätzlich sinkt die CPU-Last spürbar: Die Verarbeitung der Interrupts und der Empfangswarteschlangen ist auf solchen Karten günstiger.

Beachten Sie bei der Diagnose: Der Zähler der Empfangsfehler `recv-err` in der Statistik des Eingangs zählt nur Lesefehler des Sockets und den Synchronisationsverlust an der Grenze eines TS-Pakets. Verluste durch einen Überlauf des Empfangspuffers des Kernels erreichen die Anwendung nicht und gehen in diesen Zähler nicht ein — sie sind über den Befehl `netstat -su` (Zeile `RcvbufErrors`) und über die Continuity-Fehler im Analysator des Streams sichtbar (*Analysator*).

1.10.3 Empfehlungen zur Netzwerkkonfiguration für Multicast

Der Abschnitt betrifft die Netzwerkeingänge UDP / RTP und Pro-MPEG. Auf den SRT-Empfang wirken sich diese Parameter nicht aus: Die SRT-Bibliothek legt die Puffergröße selbst fest, während die Verzögerung und die Reserve für die Neuübertragung am Eingang selbst konfiguriert werden (*SRT*).

Die Größe des Empfangspuffers des Sockets wird über die erweiterte Eingangseinstellung „Socket buffer (bytes)“ festgelegt. Standardmäßig ist sie null — der Knoten fordert nichts an, und der Socket erhält die Puffergröße, die im System über den Parameter `net.core.rmem_default` vorgegeben ist. Sobald ein Wert ausdrücklich gesetzt ist, greift die Obergrenze `net.core.rmem_max`: Der Kernel kürzt die Anforderung stillschweigend auf diesen Grenzwert und meldet die Kürzung weder im Protokoll noch in der Statistik. Deshalb wird die Obergrenze angehoben, bevor die Puffergröße am Eingang gesetzt wird.

Die Parameter werden in einer eigenen Datei in `/etc/sysctl.d/` festgelegt:

```
cat > /etc/sysctl.d/99-pss-net.conf <<EOF
net.core.rmem_default=8388608
net.core.rmem_max=16777216
EOF
```

Anwenden:

```
sudo sysctl --system
```

Der Wert `rmem_max` ist nur eine Obergrenze; Speicher wird dafür nicht reserviert. Der Wert `rmem_default` gilt für alle UDP-Sockets des Systems; ist eine derart globale Größe unerwünscht, lassen Sie ihn unverändert und setzen Sie „Socket buffer (bytes)“ nur an den Eingängen, die ihn tatsächlich benötigen. Als Richtwert für die Berechnung dient die Bitrate des Streams multipliziert mit der zulässigen Pause bei der Bedienung des Sockets: 8 MB decken etwa 64 ms bei 1 Gbit/s ab.

1.10.4 Flussonic und SRT

Beispiel einer SRT-URL für den Empfang durch die Software Flussonic:

`srt://Stream_IP:port?streamid=flussonic`

Dabei ist **streamid** der Login des Clients in der Software Flussonic, der im Abschnitt „Konfiguration — Peer-Einrichtung“ festgelegt wird.

Beim Hinzufügen eines neuen Clients genügt es, nur den Login anzugeben; in der Test-URL ist „flussonic“ angegeben. Die Software Flussonic erzeugt beim Betrieb mit SRT automatisch eine *streamID*, und wenn in der URL auf der Empfangsseite der Login *streamID* nicht angegeben wird, wird der Stream nicht empfangen und Perfect Streamer kann ihn nicht ausliefern.

Auf dieselbe Weise lässt sich die *streamID* im Format aus Login und Passwort festlegen: Legen Sie in Perfect Streamer ein Konto an, dessen Feld „Login“ gleich `!#::u=1234567890,password=1234567890` ist, und tragen Sie in Flussonic die folgende URL ein:

```
srt://Stream_IP:port?streamid=!#::u=1234567890,password=1234567890
```

Die *streamID* lässt sich bei Perfect Streamer auch im Format einer IP-Adresse angeben — dazu wird beim Konto das Merkmal „IP-Adress-Konto“ aktiviert und in das Adressfeld eingetragen:

- `192.168.1.1` — eine einzelne IP;
- `192.168.1.1-192.168.1.254` — ein IP-Bereich.

In beiden Fällen sieht die URL für den Empfang über IP in Flussonic so aus: `srt://Stream_IP:port?streamid=*`

Der Stream wird an die IP-Adresse des Clients gebunden: Der Empfang über diese URL ist nur von der angegebenen IP-Adresse (oder aus dem angegebenen Bereich) aus möglich.

1.11 Werkzeuge

Perfect Streamer Toolkit — Kommandozeilenwerkzeuge, die den Streamer ergänzen. Sie sind Bestandteil des Pakets **pstreamer** und befinden sich nach der Installation im Verzeichnis `/opt/pss/tools/`:

- **ts_analyze** — Analysator für MPEG-TS-Transportströme mit Prüfung der Konformität zu TR 101 290 (*TS Analyze Perfect Streamer Toolkit v2.3 — TR 101 290*);
- **mpts_migrate** — Migration der DVB-SI/PSI-Identität eines im Betrieb befindlichen Multiplexes auf Perfect Streamer (*MPTS Migrate Perfect Streamer Toolkit v1.1 — Migration der MPTS-Identität*).

1.11.1 TS Analyze Perfect Streamer Toolkit v2.3 — TR 101 290

Teil des **Perfect Streamer Toolkit** — https://pstreamer.tv

Kommandozeilen-**Analysator für MPEG-TS-Transportströme** mit Konformitätsprüfung nach **ETSI TR 101 290 V1.4.1** und Validierung des T-STD-Puffermodells nach **ISO/IEC 13818-1**.

Liest **UDP multicast/unicast** oder **TS-Dateien**, erkennt die PCR-PIDs automatisch über PAT/PMT und gibt einen ausführlichen oder kurzen Bericht auf stdout aus.

Das Werkzeug ist Bestandteil des Pakets **pstreamer** und liegt nach der Installation unter `/opt/pss/tools/ts_analyze` — eine separate Installation ist nicht erforderlich.

Was geprüft wird

Der Analysator durchläuft jedes TS-Paket und meldet Verletzungen:

- **Priority 1** (Dekodierbarkeit des TS): TS-Synchronisation, Synchronisationsverlust, Vorhandensein und CRC von PAT/PMT, Kontinuitätszähler, Vorhandensein von PIDs
- **Priority 2** (empfohlene Überwachung): Übertragungsfehler-Indikator, CRC-Fehler, PCR-Wiederholung / -Genauigkeit / -Diskontinuitäten, PTS-Intervall, Vorhandensein von CAT
- **Priority 3** (erweiterte Überwachung): NIT/SDT/EIT/TDT-Intervalle, nicht referenzierte PIDs, Überlauf / Unterlauf der T-STD-Puffer

Zusätzlich werden ausgegeben:

- **PCR-Genauigkeit** bis ±500 ns — Regression über Byte-Positionen
- **PCR-Drift** in ppm (nur Live-Modus)
- **T-STD-Puffermodell** für jeden Elementarstrom (Live-Modus)
- Validierung der **SI-Sektionsgrößen** gegen die ISO/EN-Grenzwerte, mit EIT-on-STB-Kompatibilitätswarnungen (>1024 B)
- **UDP IAT** (inter-arrival time) — Jitter-Statistik auf Datagramm-Ebene (nur Live-Modus)

Verwendung

```
ts_analyze [options] <input>
```

Eingänge

Form	Beschreibung
`udp://239.1.1.1:1234`	UDP multicast
`udp://eth0@239.1.1.1:1234`	UDP multicast auf der angegebenen Schnittstelle
`udp://192.168.1.100:1234`	UDP unicast
`udp://lo@127.0.0.1:12655`	UDP unicast auf Loopback (Testquelle)
`/path/to/file.ts`	Lokale TS-Datei

Eine RTP-Kapselung von UDP wird automatisch erkannt und entpackt.

Optionen

Option	Beschreibung	Standard
`-t, --time <sec>`	Analysedauer in Sekunden	`30`
`-s, --short`	Kurzer Zusammenfassungsbericht	–
`-f, --full`	Vollständiger detaillierter Bericht	ja
`-b, --bitrate <Mbps>`	TS-Bitraten-Hinweis für den Dateimodus	`38.8`
`-p, --pcr-pid <pid>`	Nur die angegebene PCR-PID analysieren (dezimal oder `0xHHHH`)	auto
`-l, --pcr-limit <ms>`	Grenzwert für den PCR-Wiederholungsfehler in ms	`40`
`--no-eit`	EIT-Analyse überspringen — P3.7..P3.10 werden als `N/A` ausgewiesen, der EIT-Anteil an P2.2 / an der Summe der Sektionsgrößen entfällt	EIT aktiviert
`--no-nit`	NIT-Analyse überspringen — P3.1, P3.2 werden als `N/A` ausgewiesen, der NIT-Anteil an P2.2 / an der Summe der Sektionsgrößen entfällt	NIT aktiviert
`--no-color`	ANSI-Farben in der Ausgabe deaktivieren	Farben aktiviert
`--xml`	Strukturiertes XML auf stdout (kein stderr)	Text
`-h, --help`	Hilfe anzeigen	–

Beispiele

```
# 30-second TR 101 290 check on a multicast stream
ts_analyze -t 30 udp://239.10.10.1:1234

# short summary, save to log (no color)
ts_analyze -s --no-color -t 60 udp://239.10.10.1:1234 > report.txt

# analyze a TS file
ts_analyze -t 30 recording.ts

# analyze only a single PCR PID
ts_analyze -p 0x0100 -t 30 udp://239.10.10.1:1234

# machine-readable XML for CI / monitoring
ts_analyze --xml -t 30 udp://239.10.10.1:1234 > result.xml

# skip EIT/NIT analysis (e.g. for streams where they are intentionally absent)
ts_analyze --no-eit --no-nit -t 30 udp://239.10.10.1:1234
```

Deaktivieren der EIT- oder NIT-Analyse

In manchen Strömen fehlen EIT oder NIT absichtlich (geschlossene Netze, Laborquellen, reine OTT-Zustellung usw.). Die entsprechenden P3-Prüfungen aus TR 101 290 führen dann stets zu FAIL im abschließenden OVERALL-Gate — das ist reines Rauschen.

Verwenden Sie `--no-eit` und/oder `--no-nit`, um diese Prüfungen auszuschließen:

Flag	Übersprungene Prüfungen	Nebeneffekte
`--no-eit`	P3.7 (EIT actual P/F), P3.8 (EIT other P/F), P3.9 (EIT actual schedule), P3.10 (EIT other schedule)	EIT-Sektionen werden nicht erfasst, daher ist ihr Anteil an P2.2 (CRC) und an der Übersicht der SI-Sektionsgrößen null. Die EIT-on-STB-Kompatibilitätswarnung wird unterdrückt.
`--no-nit`	P3.11 (NIT actual), P3.12 (NIT other)	NIT-Sektionen werden nicht erfasst, daher ist ihr Anteil an P2.2 (CRC) und an der Übersicht der SI-Sektionsgrößen null.

Übersprungene Prüfungen erscheinen im Bericht als `N/A` mit dem Vermerk `disabled (--no-eit)` / `disabled (--no-nit)`, im XML als `applicable="false" result="N/A"`. Im Kurzbericht wird anstelle des Fehlerzählers `NIT=off` / `EIT=off` angezeigt.

Die Flags betreffen ausschließlich die Verarbeitung von EIT (PID `0x0012`) und NIT (PID `0x0010`) — alle übrigen Prüfungen nach TR 101 290 (P1.x, P2.x, SDT, TDT, CAT, T-STD, PCR-Drift, IAT) werden wie gewohnt ausgeführt.

XML-Ausgabemodus (`--xml`)

`--xml` veranlasst den Analysator, ein einziges, in sich geschlossenes UTF-8-XML-Dokument auf **stdout** auszugeben. Sämtliche Nebenausgaben (Banner, „Stream locked“, „PCR PIDs discovered“, die sekündliche Fortschrittsanzeige, die Zusammenfassung der Erfassung, Warnungen zu geringer Dauer) werden unterdrückt; **stderr bleibt leer**, sofern kein tatsächlicher Fehler auftritt (Eingang konnte nicht geöffnet werden, keine PCR-Daten, Strom zu kurz, Abbruch durch ein Signal). ANSI-Farben werden zwangsweise deaktiviert.

Der Exit-Code ist derselbe wie im Textmodus: `0` bei OVERALL=PASS, `65` bei OVERALL=FAIL, dazu die üblichen Fehler- / Signalcodes (1, 2, 3, 130, 143).

Struktur des XML auf oberster Ebene:

```
<?xml version="1.0" encoding="UTF-8"?>
<ts_analyze version="2.3">
  <source>udp://239.1.1.1:5000</source>
  <timestamp>2026-04-30T12:00:00+0300</timestamp>
  <duration_s>30.00</duration_s>
  <packets total="..." null="..."/>
  <ts_bitrate_mbps>20.012</ts_bitrate_mbps>

  <programs>
    <program number="1" pmt_pid="0x0100" pcr_pid="0x0101">
      <es pid="0x0101" stream_type="0x1b" name="H.264/AVC"/>
      <es pid="0x0102" stream_type="0x03" name="MPEG-2 Audio"/>
    </program>
  </programs>

  <tr101290>
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
    <check id="1.1" name="TS Sync Byte Error" applicable="true" errors="0" result=
↪"PASS"/>
    ...
    <check id="2.3" name="PCR Repetition Error"
           applicable="true" errors="0" result="PASS" soft_violations="3"/>
    ...
    <check id="3.4" name="Unreferenced PIDs" applicable="true" count="2" result="INFO
↪"/>
    ...
  </tr101290>

  <si_section_size oversize_total="0" eit_stb_warn_total="12">
    <table name="EIT_actual_pf" max_bytes="1380" std_limit="4096"
           oversize="0" eit_stb_warn="12" result="WARN_STB"/>
    ...
  </si_section_size>

  <iat datagrams="33252" intervals="33251" min_ms="0.002" max_ms="5.009"
       avg_ms="0.150" stddev_ms="0.295" p95_ms="0.872" p99_ms="1.076"
       max_jitter_ms="4.272" gap_threshold_ms="100.0" gaps_over_threshold="0"/>

  <pcr_pids>
    <pcr_pid value="0x0101" sid="1" pcr_count="1500" discontinuities="0"
             estimated_bitrate_mbps="18.750">
      <interval samples="1499" min_ms="19.812" max_ms="20.195"
                avg_ms="20.001" p95_ms="20.102"
                iso_hard_violations="0" tr_soft_violations="0"
                rec_violations="0" result="PASS"/>
      <accuracy samples="1499" min_ns="-148.3" max_ns="201.7" abs_max_ns="201.7"
                avg_ns="2.1" stddev_ns="45.6" p95_ns="102.3"
                violations="0" result="PASS"/>
      <drift measured="true" ppm="0.618" limit_ppm="30"
             verdict_mode="informational" result="PASS"/>
      <tstd overflows="0" underflows="0" max_fill_bytes="34218" result="PASS">
        <es_buffer es_pid="0x0101" stream_type="0x1b"
                   capacity_bytes="3000000" measuring="true"
                   es_bitrate_mbps="15.200" max_fill_bytes="34218"
                   overflows="0" underflows="0"/>
      </tstd>
    </pcr_pid>
  </pcr_pids>

  <unreferenced_pids count="2">
    <pid value="0x01ff"/>
    <pid value="0x0200"/>
  </unreferenced_pids>

  <overall result="PASS"/>
</ts_analyze>
```

Wesentliche Konventionen:

- Alle PIDs werden als `0xHHHH` formatiert (4-stelliger hex-Wert mit dem Präfix `0x`).
- Das Attribut `result` nimmt die Werte `PASS` / `FAIL` / `WARN` / `N/A` / `INFO` / `SKIP` / `ok` / `WARN_STB` an.
- Für nicht anwendbare Prüfungen (Dateimodus, fehlendes Scrambling usw.) ist das Element dennoch vorhanden, mit `applicable="false"` und `result="N/A"`, damit Konsumenten des Schemas eine

stabile Form vorfinden.

- `<drift>` führt `verdict_mode="informational"` für $30\ \text{s} \leq T < 300\ \text{s}$ und `verdict_mode="hard"` für $T \geq 300\ \text{s}$; `result="SKIP"` bei Läufen von weniger als 30 s.
- `<iat>` fehlt bei Läufen im Dateimodus.
- `<overall>` spiegelt dasselbe Gate wider wie die Zeile OVERALL im Textbericht und stimmt mit dem Exit-Code des Prozesses überein.

Die Ausgabe ist wohlgeformtes (well-formed) XML (validierbar mit `xmllint --noout`); sie lässt sich ohne Anpassungen am Parsing direkt an XSLT, Python `lxml` und Ähnliches übergeben.

Lesen des Berichts

Statusfarben

Status	Farbe	Bedeutung
`PASS` / `ok`	grün	Die Prüfung entspricht dem Standard
`WARN` / `WARNING` / `WARN(STB)`	gelb	Weiche Verletzung, ohne Auswirkung auf OVERALL
`FAIL` / `ERROR`	rot	Verletzung des Standards, wirkt sich auf OVERALL aus
`INFO` / `NOTE` / `N/A`	Standardfarbe	Nur informativ

Verwenden Sie `--no-color` bei Umleitung in Logdateien oder in Nicht-ANSI-Terminals.

Minimale Analysedauer

Dauer	Abdeckung	Exit-Code
< 2 s	Unzureichend — die Analyse wird abgelehnt	**3** (Fehler)
2–10 s	P1 + P2 sind belastbar; einigen P3-Prüfungen können die Daten fehlen	0 + WARN
10–30 s	P1 + P2 + der überwiegende Teil von P3; TDT (30 s) wird unter Umständen nicht rechtzeitig erfasst	0 + NOTE
≥ 30 s	Vollständige Abdeckung aller Prüfungen nach TR 101 290	0

Die Standarddauer beträgt **30 Sekunden** und genügt für die vollständige Abdeckung von TR 101 290. Verwenden Sie `-t <sec>`, um sie zu verlängern (etwa für Abnahmeprüfungen des PCR-Drifts) oder zu verkürzen (schnelle Smoke-Prüfungen).

Während des Laufs aktualisiert der Analysator einmal pro Sekunde eine einzeilige Fortschrittsanzeige auf stderr:

```
Progress: 47.3%  (14.2s / 30.0s, 330614 packets)
```

Die Zeile verwendet einen Carriage Return (`\r`), um im Terminal eine einzelne Zeile zu bleiben; leiten Sie stderr um (`2>/dev/null`), um sie zu unterdrücken.

Verdikt zum PCR-Drift — zweistufiges Fenster

Die Toleranz der PCR-Taktfrequenz beträgt nach **ISO/IEC 13818-1 §2.4.2.1** und **ETSI TR 101 290 ±30 ppm**. Der Driftwert im Bericht ergibt sich aus einer linearen Regression der kumulierten PCR-Sekunden gegen die Ankunftszeit nach der Uhr des Messplatzes; der statistische Fehler nimmt mit 1 / T^(3/2) ab, daher muss das Analysefenster lang genug sein, damit das Messrauschen unterhalb der Grenze von ±30 ppm bleibt.

Deshalb macht der Analysator das Verdikt zum Drift von der Analysedauer abhängig:

Fenster	Verdikt bei Überschreitung der Drift-Toleranz	Auswirkung auf OVERALL
`T < 30 s`	**skipped** (das Rauschen dominiert gegenüber der Grenze von ±30 ppm)	keine
`30 s ≤ T < 300 s`	**WARN** — nur informativ	keine
`T ≥ 300 s`	**FAIL**	OVERALL = FAIL, exit 65

`300 s` ist das Abnahmefenster (ein Produktwert; der Standard **ETSI TR 101 290** normiert die Messintervalle nicht) — lang genug, damit sich selbst ein bursthafter Zustellpfad bzw. ein Loopback-Pfad unterhalb von 1 ppm Messrauschen ausmittelt und eine Toleranzüberschreitung den Taktgenerator des Encoders und nicht das Netz widerspiegelt. Der vollständige Bericht zeigt die aktuelle Stufe in der Zeile `Verdict mode` des Blocks PCR DRIFT.

Um ein hartes PASS/FAIL-Verdikt zum Drift zu erhalten, starten Sie den Lauf mit `-t 300` oder länger.

Empfehlungen zur Qualität der Quelle (informativ; die Verdikt-Stufen werden dadurch nicht verändert):

Quelle	Minimales Fenster für ±5 ppm	Für ±2 ppm	Abnahme
Broadcast-Multicast (CBR-Netz, Jitter < 100 µs)	**30 s**	60 s	5 min
Stabiles IP-Netz (Jitter < 200 µs)	**30 s**	2 min	5–10 min
Loopback / bursthafter Sender (UDP unicast auf `lo`)	**5 min**	15 min	30 min
Kalibrierung / Labormessung	—	30 min	1+ Stunde

Beispiele:

```
# Quick PCR drift check on a real broadcast multicast (30 s)
ts_analyze -t 30 -s udp://239.1.1.1:5000

# Reliable check on a loopback source (5 min)
ts_analyze -t 300 -s udp://lo@127.0.0.1:12655

# Lab acceptance (30 min, full report to file)
ts_analyze -t 1800 -f --no-color udp://239.1.1.1:5000 > acceptance.txt
```

Wird derselbe Strom über mehrere kurze Fenster analysiert und schwankt der Driftwert zwischen den Fenstern um mehr als einige ppm, so ist der Zustell-Jitter der Engpass (das Sendetempo des Senders oder das Netz) und nicht der Taktgenerator des Encoders — vergrößern Sie das Fenster.

Exit-Codes

Code	Bedeutung
0	Analyse abgeschlossen, **OVERALL = PASS**
1	Fehler in den Argumenten oder am Eingang
2	Im Strom sind keine PCR-Daten vorhanden
3	Die Dauer des Stroms liegt unter dem Minimum von 2 s, oder es wurden überhaupt keine Daten empfangen (die Quelle sendet nicht, falsche Adresse, kein Multicast-Beitritt erfolgt, Firewall oder falsche Schnittstelle — der Analysator gibt einen Hinweis aus)
65	Analyse abgeschlossen, **OVERALL = FAIL** — Verletzung von TR 101 290 / ISO 13818-1
130	Abbruch durch **SIGINT** (Ctrl+C) — die Analyse wird verworfen, es wird kein Bericht ausgegeben
143	Abbruch durch **SIGTERM** — die Analyse wird verworfen, es wird kein Bericht ausgegeben

65 ist EX_DATAERR aus POSIX <sysexits.h>: „die Eingabedaten sind fehlerhaft“. Verwenden Sie ihn in CI / Überwachung als Gate für die Konformität des Stroms:

```
ts_analyze -s -t 60 udp://239.1.1.1:5000 || {
    case $? in
        65)  echo "stream does not conform — see report" >&2 ;;
        130) echo "interrupted by user" >&2 ;;
        *)   echo "tool error" >&2 ;;
    esac
}
```

Die Codes 130/143 folgen der POSIX-Shell-Konvention `128 + signal_number`, daher stimmt `$?` nach `Ctrl+C` mit dem überein, was bash für jeden durch SIGINT/SIGTERM beendeten Prozess zurückgibt. Bei einem Abbruch gibt der Analysator eine einzelne Zeile auf stderr aus (`Analysis interrupted by signal N — no report produced.`) und überspringt die Erzeugung des Berichts vollständig.

Beispielausgabe

Vollständiger Bericht (Auszug)

```
=========================================================================
  MPEG-TS ANALYZER v2.3 — TR 101 290 FULL REPORT
  Perfect Streamer Toolkit https://pstreamer.tv
=========================================================================
  Source     : udp://239.1.1.1:5000
  Timestamp  : 2026-04-30 12:00:00
  Duration   : 30.00 s
  Packets    : 398936 total, 12045 null
  TS bitrate : 20.012 Mbit/s
-------------------------------------------------------------------------

=========================================================================
  TR 101 290 — PRIORITY 1 (TS decodability)
=========================================================================
  |  1.1   TS Sync Byte                  :        0          PASS
  |  1.4   Continuity Count              :        0          PASS
  |  1.6   PID Absence (5s)              :        0          PASS
  ...
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
=========================================================================
  TR 101 290 — PRIORITY 2 (recommended monitoring)
=========================================================================
  |  2.3   PCR Repetition               :        0          PASS
  |  2.5   PCR Accuracy                 :        0          PASS
  ...

=========================================================================
  OVERALL COMPLIANCE: PASS  — stream is TR 101 290 compliant
=========================================================================
```

Kurzbericht

```
MPEG-TS Analyzer v2.3 | Perfect Streamer Toolkit https://pstreamer.tv
TR 101 290 Summary | udp://239.1.1.1:5000 | 30.0s | 2026-04-30 12:00:00
-------------------------------------------------------------------------------------------
↪--------------
  P1: sync=0 CC=0 PAT=0 PMT=0 PID=0  P2: TEI=0 CRC=0 PTS=0  P3: NIT=0 SDT=0 EIT=0␣
↪TDT=0 unref=2
  IAT: dgrams=33252 avg=0.150 ms max=5.009 ms stddev=0.295 ms p99=1.076 ms gaps>100.
↪000 ms=0
-------------------------------------------------------------------------------------------
↪--------------
PCR PID       SID     Count     Intv max      Jitter max    Drift     Interval ␣
↪Accuracy  T-STD
-------------------------------------------------------------------------------------------
↪--------------
0x0101        1       1500      20.195 ms     76.4 ns       0ppm      PASS      PASS ␣
↪    PASS
-------------------------------------------------------------------------------------------
↪--------------
OVERALL: PASS
```

Anmerkungen

- **Dateimodus**: PCR-Drift, das T-STD-Puffermodell und UDP IAT werden nicht gemessen — sie erfordern eine Echtzeitreferenz. Alle übrigen Prüfungen funktionieren in beiden Modi.
- **Ein einzelner Transportstrom**: Es wird jeweils ein MPTS oder SPTS analysiert.

1.11.2 MPTS Migrate Perfect Streamer Toolkit v1.1 — Migration der MPTS-Identität

Teil des **Perfect Streamer Toolkit** — https://pstreamer.tv

Erfasst die DVB-SI/PSI-Identität eines laufenden Mehrprogramm-MPEG-TS-Stroms (MPTS) und bildet sie auf einer Perfect-Streamer-Instanz (PSS) auf demselben Host nach. Ergebnis: Teilnehmerempfänger (STB / TV) arbeiten nach einer Migration oder einer Umschaltung auf die Reserve **ohne erneuten Kanalsuchlauf** weiter.

Das Werkzeug ist Bestandteil des Pakets **pstreamer** und befindet sich nach der Installation unter `/opt/pss/tools/mpts_migrate` — eine separate Installation ist nicht erforderlich.

Voraussetzungen

Stellen Sie vor dem Start des Dienstprogramms sicher, dass:

- **PSS läuft** auf demselben Host (oder auf einem über `--pss-base` erreichbaren Host). Das Dienstprogramm sucht `pss` in `/proc` und liest `pss.json`, um den Admin-Port zu ermitteln (`43971`, wenn die Konfiguration ihn nicht enthält).
- **Die MPTS-Quelle ist erreichbar**, sofern eine Erfassung vorgesehen ist (Modi 1, 2, save+apply): Die als Positionsargument `<input>` übergebene URL muss einen MPEG-TS-Strom liefern. Für UDP-Multicast müssen IGMP / Firewall den Empfang zulassen; bei Dateien muss der Pfad vorhanden sein.
- **Das Ziel-MPTS ist in PSS bereits eingerichtet**: Das Dienstprogramm legt keine neuen Streams an. Das MPTS-Objekt und mindestens so viele SPTS-Feeder, wie das Inventar Dienste enthält, müssen vorab vorhanden sein. Dienste ohne freien Feeder werden im Dialog angezeigt und können übersprungen werden.
- **Die HTTP-Admin-API ist auf localhost verfügbar** — zum Lesen von `/data/stream` (wird bei verify verwendet) und zum Schreiben von `/config/stream` (apply).

Was migriert wird

Alle für den Empfänger sichtbaren Kennungen auf Transportstrom- und Dienstebene:

- **Transportstrom**: TSID, ONID, network ID, network name, **provider name** (wird als mux-weiter `sdt-provider-name` angewendet, sofern alle Quelldienste ein und denselben Wert aufweisen), Delivery-Deskriptor (Parameter der terrestrischen / Kabel- / Satellitenübertragung), PAT/SDT/NIT-Versionen
- **Je Dienst**: `service_id`, `pmt_pid`, `pcr_pid`, `service_type`, Dienstname, logische Kanalnummer (LCN), free-CA-Kennzeichen, EIT-present- / EIT-schedule-Kennzeichen
- **Elementarströme**: PIDs (es wird ein `identity  remap` angewendet — siehe *Einschränkungen*), Stromtypen, Sprachkennungen
- **Bedingter Zugriff**: CA-Deskriptoren auf Programm- und ES-Ebene

Dienst- / Anbieternamen in DVB-Zeichenkodierungen außerhalb von ASCII (zum Beispiel ISO-8859-5 für Kyrillisch) werden automatisch nach UTF-8 dekodiert.

Für jeden Dienst wird das ES-PID-Remapping als Identitätspaare (`mpegts-pid-old` ≡ `mpegts-pid-new`) für jede PCR- / Video- / Audio- / Teletext- / Daten-PID aufgebaut, sodass die ursprünglichen PIDs in der resultierenden multiplexierten Ausgabe **byte-exact** erhalten bleiben. Ältere Empfänger, die die PMT nach dem ersten Suchlauf zwischenspeichern, arbeiten ohne Neukonfiguration weiter.

Ab PSS 2.0.0.193 aktiviert der Plan am Muxer-Eingang des Ziel-MPTS zusätzlich den regulären Modus zur Beibehaltung der ursprünglichen PIDs (deaktiviertes Automatic PID mapping) und setzt für jeden Dienst die erfasste Programmnummer in der PAT (program order) — die On-Air-Reihenfolge der Programme übersteht die Migration. Die Unterstützung wird anhand der Konfiguration des Ziel-Streamers automatisch ermittelt; auf älteren Versionen werden diese Schlüssel nicht gesendet (das Dienstprogramm gibt eine Warnung aus), und die Identitätspaare bleiben der einzige Mechanismus zur Fixierung der PIDs, während die Programmreihenfolge in der PAT nicht erhalten bleibt.

Anwendungsfälle

- **Failover**: Umschalten der Decoder vom primären MPTS auf das Reserve-MPTS unter Beibehaltung der Kanäle auf der Empfängerseite
- **Hardware-Migration**: Umzug eines laufenden Multiplex von einem PSS-Host auf einen anderen ohne Anweisungen an die Zuschauer
- **Momentaufnahme vor / nach dem Update**: Erfassung des Multiplex vor dem PSS-Upgrade und erneutes Anwenden danach, um bitidentische SI/PSI zu gewährleisten
- **Manuelle Bearbeitung und erneutes Anwenden**: Erfassen, `migrate.json` bearbeiten (Dienste umbenennen, LCN ändern, `service_type` anpassen), erneut anwenden
- **Dry-run-Prüfung**: Ausgabe jedes HTTP POST, der gesendet *würde*, ohne Auswirkung auf PSS

Schnellstart

```
# Capture from a live stream and apply to local PSS in one run
mpts_migrate udp://239.1.1.1:1234

# Capture, save to migrate.json and apply
mpts_migrate -s udp://239.1.1.1:1234

# Capture only — write to file, do not apply
mpts_migrate -o backup.json udp://239.1.1.1:1234

# Apply a previously saved JSON
mpts_migrate -i backup.json

# No arguments — load ./migrate.json and apply
mpts_migrate

# Preview what apply would do, without changes
mpts_migrate -i backup.json --dry-run
```

Arbeitsablauf

1. **Erfassung** — das Dienstprogramm öffnet den Strom, wertet PAT / PMT / SDT / NIT / EIT über `-t` Sekunden aus (Vorgabe 30) und erstellt das Inventar; die Gesamtbitrate des Stroms wird gemessen.
2. **(Optional) Speichern** — mit `-s` oder `-o` wird das Inventar zur Wiederverwendung als JSON geschrieben.
3. **PSS-Ermittlung** — erkennt das laufende PSS über einen `/proc`-Scan und liest `pss.json`, um den Admin-Port zu ermitteln (`43971`, wenn die Konfiguration ihn nicht enthält); `--pss-base http://host:port` umgeht die automatische Ermittlung.
4. **Bestätigung der Zuordnung** — ein interaktiver Dialog fragt ab, wie jeder erfasste Dienst einem vorhandenen SPTS-Feeder zugeordnet werden soll; `--non-interactive` übernimmt die Vorschläge und bricht bei einem Konflikt ab; `--target-mpts <id>` überspringt die Abfrage zur MPTS-Auswahl.
5. **Automatisches Aufheben der Feeder-Pause** — für jeden zugeordneten SPTS / muxer-output sendet das Dienstprogramm `{"pause":false}`, falls der Feeder pausiert war, damit der Multiplexer nach dem Apply tatsächlich Daten erhält.

6. **Automatische Bitratenanpassung** — überschreitet `captured_bitrate × (1 + headroom%)` die `mpegts-output-bitrate` des Ziel-MPTS, hebt das Dienstprogramm dieses Limit auf PSS mit einem einzigen POST an (mit Aufrundung auf die nächsten 1000 kbps). Abschaltbar über `--no-bitrate-adjust`.
7. **Planung** — vergleicht das Inventar mit dem aktuellen `/config/stream`-Baum auf PSS und bereitet HTTP-POST-Anfragen nur für die abweichenden Felder vor. Das ES-PID-Remapping wird als Identitätspaare (`mpegts-pid-old` ≡ `mpegts-pid-new`) erzeugt, damit die multiplexierte Ausgabe jede ursprüngliche PID unverändert beibehält — einschließlich PCR, Video, Audio, Teletext, SCTE-35, DSM-CC. Ab PSS 2.0.0.193 aktiviert der Plan zusätzlich den Modus zur Beibehaltung der ursprünglichen PIDs am Muxer-Eingang des Ziels und setzt die Programmreihenfolge in der PAT (siehe *Was migriert wird*).
8. **Anwenden** — sendet die geplanten POST-Anfragen und schaltet anschließend das MPTS auf pause/unpause, damit PSS die Konfiguration neu einliest; mit `--dry-run` wird der Plan lediglich ausgegeben.
9. **Verify** (standardmäßig aktiviert, auch bei leerem Plan) — erfasst das resultierende MPTS über einen der UDP-Ausgänge von PSS und vergleicht es mit dem Ziel; kritische Abweichungen (TSID, ONID, service_id, name, type, LCN) → Exit-Code 5.

Ein erneuter Durchlauf der gesamten Verarbeitungskette ist **idempotent**: Der zweite Lauf meldet `no changes needed`, wenn PSS bereits dem Inventar entspricht, und verify bestätigt dies mit einer neuen Erfassung.

CLI-Optionen

Modusauswahl

Option	Beschreibung	Standard
`-f, --file <path>`	Pfad zum Migrations-JSON (dient als Standardimport, wenn `-i` fehlt, und als Speicherziel mit `-s`)	`./migrate.json`
`-s, --save`	Erfasstes Inventar in die Migrationsdatei speichern (kombinierbar mit einem Stromeingang; apply wird dennoch ausgeführt)	—
`-o, --output <file>`	Nur Erfassung: Schreiben in die Datei, kein apply	—
`-i, --input <file>`	Nur apply: Datei laden, keine Erfassung	—

Erfassung

Option	Beschreibung	Standard
`-t, --time <sec>`	Maximale Erfassungsdauer	`30`
`-b, --bitrate <Mbps>`	Hinweis zur TS-Bitrate für das Pacing im Dateimodus	`38.8`

Apply / Verify

Option	Beschreibung	Standard
`--target-mpts <id>`	Abfrage zur MPTS-Auswahl überspringen; auf diese stream id auf PSS anwenden	—
`--non-interactive`	Vorschläge des Dialogs automatisch übernehmen; bei einem Konflikt beenden	—
`--dry-run`	Plan der POSTs ausgeben, nichts senden	—
`--pss-base <url>`	Automatische PSS-Ermittlung überschreiben (zum Beispiel `http://host:8808`)	automatisch
`--no-verify`	Erfassung und Diff nach dem Apply überspringen	verify aktiviert
`--verify-time <s>`	Erfassungsfenster für die Verifikation	10
`--no-bitrate-adjust`	`mpegts-output-bitrate` am Ziel-MPTS nicht anheben	Anhebung aktiviert
`--bitrate-headroom <pct>`	Bitratenreserve über dem gemessenen Wert bei der Anpassung	15

Sonstiges

Option	Beschreibung
`-v, --verbose`	Ausführliches Protokoll (jeder HTTP POST und jeder Dialogzweig)
`-h, --help`	Hilfe anzeigen und beenden

Migrationsdatei (`migrate.json`)

Menschenlesbares JSON mit `format_version: 1`. Die Standardablage ist `./migrate.json`; sie wird mit `-f` überschrieben. Beispiel für den Aufbau:

```
{
  "format_version": 1,
  "tool": "mpts_migrate",
  "capture": {
    "source": "udp://239.1.1.1:1234",
    "captured_at_utc": "2026-04-30T08:15:00Z",
    "duration_s": 8.2,
    "packets": 109344
  },
  "transport_stream": {
    "transport_stream_id": 1234,
    "original_network_id": 8442,
    "network_name": "Operator",
    "delivery": { "type": "terrestrial", "frequency_khz": 522000 }
  },
  "services": [
    {
      "service_id": 1, "pmt_pid": 256, "pcr_pid": 256,
      "service_type": 1, "service_name": "Channel 1",
      "provider_name": "MyProvider", "logical_channel_number": 101,
      "program_order": 1,
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
      "free_ca_mode": false,
      "elementary_streams": [
        { "pid": 256, "stream_type": 27, "language": "rus" }
      ]
    }
  ]
}
```

Die Datei kann vor einem erneuten Apply bearbeitet werden: Dienste umbenennen, LCN ändern, `service_type` wechseln, `network_name` anpassen — `mpts_migrate -i migrate.json` sendet nur die geänderten Felder.

Verbindung zu PSS

- **Automatische Ermittlung**: Durchsuchen von `/proc/<pid>/comm` nach `pss`, Lesen der zugehörigen `--config`-Datei, Übernahme von `web-server.bind-port` (`43971`, wenn der Schlüssel fehlt).
- **Manuell**: `--pss-base http://host:port` umgeht die automatische Ermittlung vollständig. Nützlich für ein entferntes PSS oder wenn `--dry-run` einen Plan ohne laufendes PSS erstellen soll.
- Die Admin-REST-API erfordert auf `localhost` keine Authentifizierung.

Verifikation

Wenn `--no-verify` **nicht** angegeben ist (Vorgabe), führt das Dienstprogramm nach dem Anwenden des Plans Folgendes aus:

1. Sucht am Ziel-MPTS einen UDP-Ausgang auf localhost oder fügt vorübergehend einen auf `127.0.0.1:<auto>` hinzu (mit dem Vermerk `added by mpts_migrate for verification`).
2. Erfasst das laufende MPTS über diesen Ausgang für `--verify-time` Sekunden (Vorgabe 10).
3. Vergleicht das erfasste Inventar mit dem Ziel:
   - **Kritische** Abweichung (TSID, ONID, network name, service_id, name, type, LCN) → Exit-Code **5**.
   - **Weiche** Abweichung (PMT PID, Anzahl der ES, Anbietername, Programmreihenfolge in der PAT) → wird als Warnung ausgegeben, Exit-Code 0.
4. Ist das Ziel-MPTS überlastet (Ausgangsbitrate ≥ der konfigurierten `mpegts-output-bitrate`), wird `WARNING: target MPTS is overloaded` ausgegeben — siehe *Bitrate adjust* weiter unten.

Dry-run

`--dry-run` gibt jede HTTP-Anfrage aus, die das Dienstprogramm senden *würde* (Pfad, JSON-Rumpf), sendet jedoch keine davon. Nützlich für:

- Die Durchsicht des Plans mit dem Bediener vor der Übernahme.
- Die Erzeugung reproduzierbarer Änderungssätze in CI / Change-Management.
- Den Betrieb bei nicht erreichbarem PSS (zu kombinieren mit `--pss-base http://host:port`).

Dry-run startet keine Verifikation.

Bitrate adjust

Überschreitet `captured_bitrate × (1 + headroom%)` standardmäßig die `mpegts-output-bitrate` des Ziel-MPTS, hebt das Dienstprogramm dieses Limit auf PSS an, bevor die SI/PSI-Änderungen angewendet werden. Ohne ausreichende Reserve verliert ein überlastetes MPTS Daten niedriger Priorität — typische Symptome sind Tonaussetzer bei Radiodiensten und sporadische EIT-CRC-Fehler. Abschaltbar über `--no-bitrate-adjust`; die Reserve wird über `--bitrate-headroom <pct>` eingestellt (Vorgabe 15).

Exit-Codes

Code	Bedeutung
`0`	Erfolg — apply und (sofern aktiviert) die Verifikation sind fehlerfrei durchgelaufen. Wird auch von einem erfolgreichen `--dry-run` zurückgegeben.
1	Fehler bei Argumenten / Datei / Ermittlung
2	In der Quelle wurde keine PAT gefunden — der Eingang ist kein gültiger MPEG-TS oder das Erfassungsfenster ist zu kurz; wird auch zurückgegeben, wenn der Bediener die Bestätigung der Zuordnung abgelehnt oder den Dialog verlassen hat (apply abgebrochen)
3	Ein oder mehrere HTTP POSTs des Apply sind fehlgeschlagen
4	Apply war erfolgreich, das Ziel-MPTS ist jedoch nicht in den Zustand *Running* zurückgekehrt
5	Die Verifikation ist fehlgeschlagen — der erfasste Strom weicht in kritischen Feldern vom Ziel ab

Einschränkungen und Fallstricke

- **PSS muss das Ziel-MPTS und die Feeder bereits vorhalten**: Das Dienstprogramm legt keine neuen Streams an. Das Ziel-MPTS und mindestens so viele SPTS-Feeder, wie das Inventar Dienste enthält, müssen vorab vorhanden sein; Dienste ohne freien Feeder werden im Dialog übersprungen.
- **Die MPTS-Quelle muss während der Erfassung erreichbar sein** (Modi 1, 2, save+apply): Die als Positionsargument `<input>` übergebene URL muss einen MPEG-TS-Strom liefern; für UDP-Multicast müssen IGMP / Firewall dies zulassen.
- **Das ES-PID-Remapping wird automatisch angewendet**: Für jeden Dienst wird ein Identity-Remap (`mpegts-pid-old` ≡ `mpegts-pid-new`) für jede PCR-/Video-/Audio-/Teletext-/Daten-PID erzeugt, damit die multiplexierte Ausgabe die ursprünglichen PIDs byte-exact beibehält. Das PMT-Layout bleibt von Migration zu Migration stabil — selbst ältere Empfänger (STB vor 2015, Samsung vor der H-Serie, LG vor WebOS 3.0), die PIDs zwischenspeichern, benötigen keinen erneuten Suchlauf.
- **Der Delivery-Deskriptor muss dem tatsächlichen Übertragungsweg entsprechen** — für Empfänger, die sich über die NIT neu abstimmen (DVB-T/T2/C/S). Ein nicht passender `delivery`-Block kann den Empfänger auf eine falsche Frequenz führen.
- **Die Konsistenz der LCN** zwischen primärem und Reserve-MPTS ist für Failover entscheidend — unterscheiden sie sich, verschieben sich die Kanalpositionen in der Empfängerliste nach der Umschaltung.
- **Der Provider name ist in PSS mux-weit gültig** (ein `sdt-provider-name` je MPTS-Muxer-Eingang). Das Dienstprogramm wendet ihn automatisch an, wenn alle erfassten Dienste ein und denselben Wert aufweisen; tragen verschiedene Dienste unterschiedliche `provider_name`-Werte, wird eine Warnung ausgegeben und das Feld bleibt unberührt — die Entscheidung liegt beim Bediener.
- **Kein PSS-Konfigurationswerkzeug**: `mpts_migrate` berührt ausschließlich die SI/PSI-Identitätsfelder, das `pause`-Kennzeichen jedes Feeders, (optional) `mpegts-output-bitrate`

sowie ab PSS 2.0.0.193 die Schlüssel für den PID-Beibehaltungsmodus und die Programmreihenfolge. Encoder, Eingänge, Verschlüsselung, Zeitpläne und Ähnliches konfiguriert es nicht.

- **Auf PSS älter als 2.0.0.193** werden der Modus zur Beibehaltung der ursprünglichen PIDs am Muxer und die Programmreihenfolge in der PAT nicht unterstützt: Das Dienstprogramm gibt eine Warnung aus, die PIDs werden allein durch Identitätspaare fixiert, und die Programmreihenfolge kann nach der Migration von der ursprünglichen abweichen.

Fehlerdiagnose

Symptom	Wahrscheinliche Ursache / Abhilfe
`Error: no PAT seen in stream`	Die Quelle ist kein MPEG-TS, IGMP/Firewall blockiert den Multicast, oder `-t` ist zu kurz
`Error: cannot reach PSS`	`--pss-base http://host:port` verwenden, um die automatische Ermittlung zu umgehen
Apply war erfolgreich, das MPTS bleibt jedoch pausiert	Das PSS-Protokoll prüfen; mit `-v` erneut starten, um den vollständigen Plan der POSTs zu sehen
Die Verifikation meldet eine kritische Abweichung	Ziel und erfassten Strom im JSON vergleichen; meist wurde ein Feeder im Dialog dem falschen Dienst zugeordnet
`WARNING: target MPTS is overloaded`	`mpegts-output-bitrate` auf PSS anheben oder `--bitrate-headroom <higher %>` verwenden; ohne Reserve werden das Audio der Radiodienste und die PSI-Tabellen beschädigt

1.12 Referenzmaterialien

1.13 Versionsverlauf

1.13.1 Version 2.0.0.206 Beta

25.07.2026

- **Meshwork (Multi-Domain)** — mehrere Server werden zu benannten Domains zusammengeschlossen, und die Domain dient als Isolationsgrenze: die Mitgliedschaft der Knoten, der Ressourcenkatalog und die Alerts verbreiten sich nur innerhalb der eigenen Domain, zwischen den Domains wird nichts übertragen. Ein Knoten kann gleichzeitig mehreren Domains angehören, ohne deren Daten zu vermischen. Ausführlich — *Meshwork*.
- **Automatische Knotenerkennung** — es genügt, die Domain und einen einzigen Einstiegspunkt anzugeben: die Knoten tauschen Nachbarlisten aus und bauen selbstständig ein vollvermaschtes Netz direkter Verbindungen auf. Der gemeinsame Domain-Schlüssel wird automatisch erzeugt und ausschließlich über einen geschützten Kanal übertragen — TLS oder ein vertrauenswürdiges Segment des lokalen Netzes.
- **Verteilte Ressourcenbibliothek** — die Ausgänge UDP, RTP, Pro-MPEG und RIST, die Multicast-Eingänge UDP, RTP und RIST sowie die PS1- und SRT-Verbindungen werden im gemeinsamen Katalog der Domain (`/data/library`) veröffentlicht: eine einheitliche Liste der Multicast-Gruppen, Ports, VLANs, SSM-Quellen und Punkt-zu-Punkt-Verbindungen mit Angabe von Knoten und Stream. Am Katalog ist ersichtlich, welche Adressen und Ports bereits belegt sind, und ein neuer Eingang wird

direkt daraus angebunden — durch Auswahl eines Streams, der auf einem anderen Knoten der Domain bereits ausgeliefert wird, ohne die Adresse manuell einzugeben.

- **Übersicht der Domain-Knoten** — zu jedem Knoten werden Status, Zeitpunkt des letzten Kontakts, Rolle, Region, Build-Version, CPU-Auslastung, Datenverkehr je physischer Schnittstelle und die Anzahl der Streams auf Sendung angezeigt, zu jeder Verbindung zwischen Knoten über PS1 oder SRT — Bitrate, RTT, Anteil der Wiederholungen und Verluste sowie das Zustandsurteil.
- **Betrieb hinter NAT** — ein Knoten hinter NAT nimmt vollwertig und ohne Weiterleitung eingehender Ports an der Domain teil: er baut die Verbindung selbst auf, sein öffentlicher Zugangspunkt wird aus der tatsächlich sichtbaren Adresse und dem angekündigten Port abgeleitet, und seinen Zustand geben die Nachbarn einen Hop weiter — der Knoten ist auch für diejenigen sichtbar, die ihn nicht direkt erreichen können.
- **Automatische Anmeldung der Verbindungen innerhalb der Domain** — für PS1- und SRT-Verbindungen zwischen Knoten derselben Domain müssen nicht für jede Verbindung Benutzername und Passwort angelegt werden: auf der rufenden Seite (PS1-Eingang, SRT-Eingang, SRT-Ausgang im caller-Modus) genügt es, die Einstellung „Meshwork peer auth“ zu aktivieren, und die Autorisierung erfolgt über das gemeinsame Domain-Geheimnis. Der bestätigte Partnerknoten und dessen Stream sind in der Ressourcenbibliothek und in der Sitzungsliste sichtbar.
- **Low-Latency HLS und MPEG-DASH über CMAF** — der neue Auslieferungsmodus „OTT / LL-HLS / DASH“: der integrierte Multiplexer erzeugt fragmentiertes MP4 (CMAF), über das MPEG-DASH unter `/dash` und Low-Latency HLS unter `/llhls` ausgeliefert werden; adaptive Bundles werden auch für diese Formate zusammengestellt. Ausführlich — *Auslieferungsmodi*.
- **Geringe Latenz bei LL-HLS** — die Medien-Wiedergabeliste wird in Teile zerlegt, das blockierende Neuladen der Wiedergabeliste und der Preload-Hinweis werden eingesetzt, sodass der Player mit der Wiedergabe beginnt, ohne die Fertigstellung eines vollständigen Segments abzuwarten. Die CMAF-Segmente führen Producer Reference Time mit, das DASH-Manifest kündigt UTCTiming und die Ziellatenz an, und die Einstellung „LL-Part-Zieldauer (ms)“ wird im laufenden Betrieb wirksam, ohne den Stream neu zu starten.
- **HTTP/3 (QUIC)** — HLS, MPEG-DASH, LL-HLS und MPEG-TS over HTTP werden über QUIC auf einem separaten UDP-Port ausgeliefert, mit optionalem 0-RTT; den Wechsel zu QUIC fordert der Client mit dem Parameter `?h3` an. Die administrativen Routen bleiben ausschließlich auf TCP.
- **Segmentausrichtung an IDR** — der Segmentierer unterscheidet ein IDR- von einem gewöhnlichen I-Frame: bei Quellen mit closed-GOP beginnt jedes Segment mit einem vollwertigen Einstiegspunkt, sodass der Player den Stream an jedem beliebigen Segment öffnet; bei Quellen mit open-GOP dient das nächstgelegene I-Frame als Grenze.
- **Inhaltsschutz (DRM)** — die OTT-Auslieferung eines Streams kann verschlüsselt werden: HLS AES-128 mit einem Schlüssel vom Server selbst oder von einem externen Key-Server und mit optionaler Rotation nach Zeitfenstern, oder MPEG Common Encryption (ISO/IEC 23001-7) nach den Schemata cenc und cbcs für CMAF und DASH. Die Verschlüsselung erfolgt einmalig, im Moment der Segmentbildung, daher wird das DVR-Archiv verschlüsselt gespeichert und über beliebig viele Schlüsselwechsel hinweg wiedergegeben. Ausführlich — *Inhaltsschutz (DRM)*.
- **DVR (Netzwerkarchiv)** — jeder OTT-Kanal wird parallel zur Auslieferung auf die Festplatte geschrieben, mit derselben Segmentierung und ohne separaten Rekorder; im Modus geringer Latenz wird das Archiv in zwei Linien geführt, MPEG-TS und CMAF, sodass die Aufzeichnung im selben Container wie die Live-Ausstrahlung verfügbar ist. Ausführlich — *DVR*.
- **Archivwiedergabe (VOD)** — das Archiv wird über dieselben HLS-, DASH- und LL-HLS-URL ausgeliefert wie die Live-Ausstrahlung: `t=<Zeit>` schaltet den VOD-Modus ein und legt den Beginn des Fensters fest, `d=<Sekunden>` dessen Dauer. Die HLS-Wiedergabeliste ist dabei geschlossen, das DASH-Manifest statisch, Aufzeichnungslücken werden als separate Perioden ausgewiesen; adaptive Bundles geben das Archiv mit denselben Parametern wieder.

- **Catch-up über EPG** — statt `t` und `d` genügt es, `epg=<Zeit>` zu übergeben: der Server findet auf dem verknüpften EPG-Kanal selbst die zu diesem Zeitpunkt laufende Sendung und übernimmt deren Beginn und Dauer als Grenzen des Wiedergabefensters.
- **Untertitel im Archiv** — WebVTT-Spuren werden zusammen mit den Segmenten auf die Festplatte geschrieben und in VOD über dieselben URL wiedergegeben; Fenster ohne Untertitelzeilen belegen keinen Speicherplatz auf der Festplatte.
- **DVR-Speicher** — es kann mehrere Speicher geben, jeder mit einer eigenen Füllgrenze und einer eigenen Bereinigungsreihenfolge. Die Archivtiefe („Aufbewahrung (Stunden)“, bis zu 90 Tage) und das geschützte Minimum werden für jeden Stream separat festgelegt, und die Bereinigung nach freiem Speicherplatz rührt weder das geschützte Minimum noch die Fenster offener VOD-Sitzungen an.
- **DVR-Monitor** — ein separater Bildschirm zeigt Zustand und Füllung der Speicher, Umfang und Tiefe des Archivs je Stream sowie die Lese- und Schreiblatenzen, und das Abdeckungshistogramm (`/data/dvrstat`) markiert nicht nur die Lücken im Archiv, sondern auch deren Ursache — Abriss des Eingangs, PMT-Wechsel, Verwürfelung, Auslösen der Bereinigung. Von hier aus werden auch das Archiv eines einzelnen Streams und die Verzeichnisse gelöschter Streams entfernt.
- **Alerter** — der neue Benachrichtigungsdienst: jede Störung wird zu einem Vorfall, der automatisch ausgelöst, aktualisiert und behoben wird, und der deduplizierte aktive Satz zeigt nur das, was gerade geschieht. Der Schweregrad wird über eine einheitliche Skala festgelegt — von rein informativ bis hin zu „erfordert einen Administrator-Eingriff“; ein solcher Vorfall wird durch die Bestätigung des Bedieners behoben. Ausführlich — *Benachrichtigungen (Alerter)*.
- **Katalog der Alert-Codes** — mehr als fünfzig Vorfalltypen in einer einzigen Liste: Streams samt ihren Eingängen und Ausgängen, Verstöße gegen TR 101 290, Ressourcen des Knotens, DVR-Speicher und Zustand der Aufnahme, DVB- und CI/CAM-Empfang, Transcoder, OTT, Zertifikate und Knoten der Domain. Die Auslöseschwellen werden im Benachrichtigungsbereich konfiguriert.
- **Externe Zustellung von Benachrichtigungen** — die Alerts verlassen die Web-Konsole: per E-Mail über SMTP (STARTTLS, implicit TLS, AUTH), per Nachricht an Telegram und durch den Start eines beliebigen Befehls, dem der Vorfall über Umgebungsvariablen und als vollständiges JSON auf der Standardeingabe übergeben wird. Jeder Kanal hat einen eigenen Schalter und einen eigenen Schweregrad-Schwellenwert.
- **Alerts der gesamten Domain** — der Diensthabende sieht die Alerts jedes Knotens von jedem anderen aus: der aktive Satz wird zusammen mit dem Keep-alive repliziert und wird sofort behoben, sobald er an der Quelle verschwunden ist, und in jeder Zeile ist der Ursprungsknoten angegeben. Hardware- und System-Alerts bleiben standardmäßig lokal und werden über eine separate Einstellung auf die Domänenebene angehoben.
- **Protokoll in der Datenbank** — die Meldungen des Dienstes werden in SQLite geschrieben: typisierte Einträge mit Quelle und Schweregrad, Zusammenfassung aufeinanderfolgender Wiederholungen zu einer Zeile mit Zähler, Begrenzung der Aufbewahrung nach Dauer und nach Umfang. Im Unterschied zum aktiven Alert-Satz übersteht das Protokoll einen Neustart.
- **TR 101 290-Monitor** — die fortlaufende Kontrolle der Konformität des Eingangsstreams mit dem Standard: ein einziges Urteil („Gut“, „Schlecht“, „Prüfung läuft…“) und ein strukturierter Bericht nach den Prioritäten 1, 2 und 3, in dem für jeden Indikator der Abschnitt des Standards, der gemessene Wert und der Grenzwert angegeben sind. Ein MPTS-Multiplex wird als Ganzes bewertet — über alle PID, Programme und SI-Tabellen. Der Stream wird automatisch als CBR oder VBR klassifiziert, und die Prüfungen, die nur bei konstanter Bitrate sinnvoll sind, werden auf einer VBR-Quelle nicht bewertet und erzeugen keine Fehlauslösungen. Ausführlich — *TR 101 290-Monitor*.
- **Drift des Referenzoszillators** — das systematische Weglaufen der Taktfrequenz der Quelle wird durch lineare Regression in ppm bei einer Toleranz von ±30 ppm nach ISO/IEC 13818-1 gemessen und als separate Metrik ausgegeben. Ein langsames Weglaufen wird durch sanfte Mikroverschiebungen des

Synchronisationspunkts ausgeglichen („Sync-Drift-Kompensation“, standardmäßig aktiviert), sodass es am Ausgang keine Sprünge gibt.

- **T-STD-Puffermodell** — die Analyse des Videopuffers des Referenzdecoders nach ISO/IEC 13818-1 mit Zählern für Überläufe und Unterläufe für MPEG-2, H.264 und HEVC; die Zeitrechnung erfolgt nach der PCR-Uhr, und die Entleerungsrate richtet sich nach der tatsächlichen Videobitrate. Wird über die Einstellung „T-STD-Videopuffer analysieren“ aktiviert.
- **Analyse für die Werbeeinblendung** — die Auswertung von SCTE-35-Sektionen mit Splice-Ereignissen, Splice-Punkte auf Transportebene mit konfigurierbarer Vorabbenachrichtigung sowie Random-Access-Marken. Die Daten werden bei aktivierter Tiefenanalyse in der Stream-Statistik ausgegeben.
- **Reklamationsassistent** — für einen Stream mit anhaltenden Verstößen wird ein fertiger Aufgabentext für einen KI-Chat erzeugt, aus dem dieser ein förmliches Reklamationsschreiben an den Anbieter des Streams verfasst: die Liste der anhaltenden Verstöße, die gemessenen Werte, die Abschnitte des Standards und die Auswirkung auf den Decoder. Der Aufgabentext wird über die Anfrage `GET /data/stream/<id>/ai-complaint-prompt` ausgeliefert; der Name des Streams und die Adresse der Quelle werden nicht in den Text aufgenommen.
- **CI/CAM (EN 50221)** — ein integrierter Common-Interface-Host: Erkennung des Moduls, Auslesen seines Namens und der Liste der unterstützten CA_system_id, Auswahl der zu entschlüsselnden Programme und Übergabe der CA_PMT für jedes von ihnen. Ein mit dem DVB-Empfänger kombiniertes Modul entschlüsselt die ausgewählten Programme des empfangenen Multiplexes inline, mehrere gleichzeitig; der Zustand der Slots und das Urteil zu jedem Programm sind in der Oberfläche sichtbar.
- **Software-Entschlüssler BISS-1 / BISS-E** — wird nicht nur auf den Empfang von einer DVB-Karte angewendet, sondern auf jeden Stream-Eingang: der Schlüssel wird für das gesamte SPTS oder je Programm des Multiplexes festgelegt und im laufenden Betrieb geändert, ohne den Eingang neu zu starten. Beim Empfang von einer DVB-Karte wird das Ergebnis zusätzlich anhand des Streams selbst geprüft, sodass ein falscher Schlüssel nicht wie eine erfolgreiche Entschlüsselung aussieht, sondern einen eigenen Alert auslöst.
- **Bereinigung des bedingten Zugriffs beim DVB-Empfang** — aus einem von einer DVB-Karte empfangenen Multiplex werden über eine separate Adaptereinstellung die ECM- und EMM-PID entfernt und aus der PMT die CA-Deskriptoren der offenen Programme: weiter im Signalweg läuft ein sauberer FTA-Stream. Bei Verlust des Schlüssels kehrt die Signalisierung des bedingten Zugriffs zurück, damit ein weiter unten in der Kette liegender Empfänger den Zugriff erneut anfordern kann.
- **Transcoder-Telemetrie** — zu jedem Encoder werden die Medieninformationen des transcodierten Ausgangs veröffentlicht: Bildformat, Videocodec, Satz der Audiocodecs und aktuelle Bitrate, und zu jedem Prozess — CPU-Auslastung und belegter Speicher.
- **RTMP und RTMPS** — ein Kanal wird an einen beliebigen RTMP-Empfänger veröffentlicht, einschließlich YouTube und Facebook: Video in H.264 oder HEVC (HEVC — über Enhanced RTMP) und eine AAC-Spur; für `rtmps://` verbindet sich der Knoten als TLS-Client. In der Gegenrichtung holt der Eingang den Stream selbst von einer Quelle `rtmp://` oder `rtmps://` und remultiplexiert FLV nach MPEG-TS.
- **Nahtlose Quellenumschaltung** — beim Wechsel des aktiven Eingangs am Sender verbinden sich die empfangenden PS1-Peers nicht neu: Nummerierung und Zeitstempel bleiben durchgängig, ein Warteschlangenstoß wird durch das Verwerfen der ältesten Pakete abgefangen, und die Lücke wird durch die reguläre Retransmission nachgeholt.
- **Adaptive PS1-Retransmission** — der Empfänger misst die tatsächliche RTT zum Sender und passt die Intervalle der Wiederholungsanfragen selbst daran an; die gemessene RTT, das aktuelle Intervall der erneuten Anfrage und das Merkmal einer zu geringen Latenz werden in der Verbindungsstatistik ausgegeben. Die Latenz des Empfangstunnels wird direkt in Millisekunden festgelegt statt wie bisher als Vielfaches der RTT.
- **Die Zustände „Verbindung wird wiederhergestellt“ und „Admin-Aktion“** — ein Eingang oder Ausgang,

der seine Quelle verloren hat, wird an Ort und Stelle wieder geöffnet und bleibt für die gesamte Dauer der Versuche im Zustand „Verbindung wird wiederhergestellt“, wobei er eine Warnung je Episode auslöst statt einer Meldung bei jedem Versuch. Eine Ursache, die sich durch Wiederholung nicht beheben lässt — ein belegter Port, eine fehlende Schnittstelle, Daten, die kein MPEG-TS sind, ein SRT-Fehlschlag bei der Verschlüsselung —, versetzt der Knoten in den Zustand „Admin-Aktion“ mit einem haftenden Alert, während eine Änderung der Einstellungen sofort wirksam wird.

- **MPTS-Multiplexer: ursprüngliche Identität** — die Programme können ihre ursprünglichen PID behalten, und die Reihenfolge der Programme in PAT, SDT und NIT wird manuell festgelegt, sodass ein Multiplex ohne Änderung der Stream-Identität für die Empfangsgeräte auf **Perfect Streamer** migriert wird. Programme mit TS-Verwürfelung werden mit ihren ursprünglichen Zeitstempeln übernommen.
- **Integration mit externem Billing** — jede Zuschauersitzung wird in einem externen Billing- oder CRM-System autorisiert, durch periodische Re-Autorisierung gehalten und nach ihrem Ende demselben System gemeldet: ein Lebenszyklus Start / Interim / Stop im RADIUS-Stil, ein Widerruf des Abonnements bricht die Sitzung mitten im Betrachten ab. Erfasst sind OTT, das PS1-Peering und die SRT-Ausgänge, und das Billing-System selbst wird über Vorlagen für die Anfrage und die Auswertung der Antwort angebunden, ohne Anpassung der Software.
- **Integrierter ACME-Client (Let's Encrypt)** — Ausstellung und automatische Verlängerung der Zertifikate für den Webserver, den HTTP/OTT-Server und den EPG-Server: ein Zertifikat wird für jeden konfigurierten Hostnamen bestellt, die Verlängerung wird täglich geprüft, bei einem Fehlschlag wird ein Alert ausgelöst. Unterstützt werden eine beliebige Zertifizierungsstelle und die externe Kontobindung, und ein neues Zertifikat wird ohne Neustart des Dienstes wirksam.
- **Neue Weboberfläche** — die Verwaltung des Knotens wurde auf eine neue Konsole umgestellt, die unter der Adresse des Knotens geöffnet wird (Pfad `/admin`): Übersicht, Streams, Systemüberwachung, DVB, DVR, Transcoder, Clients, Protokoll, Alerts und alle Servereinstellungen. Eigene Diagramme zeigen die Topologie der Meshwork-Domain und den Signalweg des ausgewählten Streams von den Eingängen bis zu den Ausgängen. Die frühere Oberfläche bleibt unter `/classic` erhalten. Ausführlich — *Weboberfläche*.
- **Aktualisierung in Echtzeit** — der Knoten liefert seinen Zustand als Ereignisstrom unter `/data/events`: beim Verbinden trifft eine vollständige Momentaufnahme ein, danach werden im Sekundentakt die Frames der Streams, der Systemressourcen, der Clients und der DVB-Adapter aktualisiert, während Alerts und Zustandswechsel als Ereignisse eintreffen. Die Oberfläche und externe Dashboards arbeiten ohne periodische Abfrage.
- **Kontextbezogene Hilfe** — jeder Bildschirm und jedes wichtige Fenster der neuen Oberfläche verfügt über einen Hilfeaufruf, der genau den Dokumentationsabschnitt zu dem öffnet, was gerade geöffnet ist: die allgemeine Hilfeschaltfläche für den Bildschirm, die Schaltfläche „?“ in der Fensterkopfzeile für dieses Fenster. Der Abschnitt wird in der Sprache der Oberfläche geöffnet.
- **Sprachen, Designs und Layouts** — die Oberfläche ist in sechs Sprachen übersetzt (Russisch, Englisch, Deutsch, Französisch, Spanisch, Portugiesisch), unterstützt ein helles und ein dunkles Design und passt die Layoutdichte selbst an Telefon, Tablet, Arbeitsplatzrechner und Videowand an.
- Weitere Verbesserungen und Fehlerbehebungen.

1.14 Roadmap

Der Abschnitt führt die Funktionen auf, die angekündigt sind, aber noch nicht in den aktuellen Lieferumfang eingegangen sind. Für jede ist angegeben, was heute bereits verfügbar ist und was noch nicht — damit sich die Einführungsplanung auf die tatsächlichen Fähigkeiten des Knotens stützt. Umfang und Termine können sich ändern; den aktuellen Stand erfragen Sie beim Anbieter.

1.14.1 Entschlüsselung einzelner Streams über ein CAM

Heute verfügbar. Die Hardware-Entschlüsselung CI/CAM (EN 50221) steht am eigenen DVB-Empfänger des Knotens zur Verfügung: Ein im CI-Slot der Tunerkarte installiertes Modul entschlüsselt ausgewählte Programme des von diesem Tuner empfangenen Multiplex — mehrere Programme gleichzeitig, wobei die Liste im laufenden Betrieb ohne Neustart des Adapters übernommen wird. Der Zustand des Slots, der Name des Moduls, die unterstützten CA-Systeme und das Urteil des Moduls über das Abonnement je Programm werden auf dem Bildschirm „Hardware“ angezeigt; vorgesehen sind Alarme über das Entnehmen des Moduls, ein fehlendes Abonnement und einen Conditional-Access-Fehler (*Entschlüsselung*).

Noch nicht verfügbar. Ein Stream, der nicht zum Multiplex des eigenen Tuners gehört, lässt sich nicht durch ein CAM-Modul führen — einen Pfad „Stream → Modul → Stream“ gibt es nicht. Ein über das Netz oder aus einer Datei empfangener SPTS oder MPTS wird nicht in Hardware entschlüsselt; für solche Quellen steht nur die softwareseitige Entschlüsselung BISS-1 / BISS-E zur Verfügung (*SPTS-Streams*, *BISS-Entschlüsselung*). Ein für diese Betriebsart ausgelegtes Modul erkennt der Knoten und zeigt es im Hardware-Inventar an, führt aber keine Daten hindurch.

1.14.2 Kommerzielle DRM-Systeme für die OTT-Auslieferung

Heute verfügbar. Die OTT-Auslieferung verschlüsselt der Knoten mit eigenen Mitteln: HLS AES-128 — mit Ausgabe des Schlüssels über eine Sitzungs-URL oder mit einem Verweis auf einen externen Schlüsselserver und mit Rotation abgeleiteter Schlüssel nach Zeitfenster — sowie MPEG Common Encryption (ISO/IEC 23001-7) in den Schemata cenc und cbcs über CMAF. Die Verschlüsselung erfolgt einmalig, bei der Bildung des Segments, daher wird das DVR-Archiv verschlüsselt gespeichert und wiedergegeben (*Inhaltsschutz (DRM)*).

Noch nicht verfügbar. Eine fertige Integration mit den Lizenzservern von Widevine, PlayReady und FairPlay gibt es nicht: Das Manifest und das Initialisierungssegment tragen nur die allgemeine Schutzsignalisierung — ohne die Systemkopfdaten dieser DRM-Systeme und ohne die Adresse für den Lizenzbezug. Einen automatischen Schlüsselaustausch mit einem Schlüsselanbieter (CPIX, SPEKE) gibt es ebenfalls nicht: Das Paar „Schlüssel — Schlüsselkennung“ übergibt der Betreiber manuell an seine eigene DRM-Infrastruktur. Das Schema HLS SAMPLE-AES wird nicht unterstützt, und Common Encryption wird ausschließlich über MPEG-DASH ausgeliefert; eine geschützte Auslieferung von HLS über CMAF und von Low-Latency HLS gibt es daher noch nicht.

1.14.3 Signalisierung der Punkte für die Werbeeinblendung in der OTT-Auslieferung

Heute verfügbar. Der Analysator erkennt SCTE-35-Ereignisse und Splice-Punkte und zeigt sie im Bericht an (*Fortlaufende Messungen*). Beim Remultiplexen wird die PID mit der Signalisierung unverändert in den Ausgangsmultiplex übernommen.

Noch nicht verfügbar. Die Grenzen der Werbeblöcke werden nicht in die HLS-Wiedergabelisten und die DASH-Manifeste übernommen: Die Segmente werden am Splice-Punkt nicht geschnitten, dem Player und einem externen System zur Werbeeinblendung wird nichts signalisiert; auch ein Austausch der Werbeblöcke im Stream selbst findet nicht statt. Durch den Transcoder gelangt die Signalisierung nicht — in der transkodierten Kopie des Kanals gehen die Splice-Marken verloren.