James Armstrong

Writing Samples - Senior Technical Writer

Document Information

Issue Date	2019-01-01
Document ID	james-armstrong-sample-2019-0101
Summary	Provides a collection of writing samples authored by James Armstrong.

Contents

Documentation Changes

This section provides an example of the format I developed to enable a cumulative change logging that is easily understood by customers.

December 14th, 2018 Documentation Release

This topic lists documentation changes in the December 14, 2018 Server Troubleshooting Guide documentation release for Ericsson Mediaroom Server 3.0 FP5.

Troubleshooting Common Components

- PlayReady Policy Handler (p. 1133)

  Added new component with two new scenarios:

  - "Failure to Get PlayReadyPolicy Configuration".
  - "Undefined Policy Profile ID".
- IngestionService (p. 968)

  Added new scenario "No Target Profile Subscriber Groups".

August 15th, 2018 Documentation Release

This topic lists documentation changes in the August 15, 2018 Server Troubleshooting Guide documentation release for Ericsson Mediaroom Server 3.0 FP4.

Troubleshooting Ericsson Mediaroom Server Data Services

- Ericsson Mediaroom Data Services Central Partition Monitoring Scenarios (p. 941)

Added new scenario “Failed to load intent parser”.

June 2nd, 2017 Documentation Release

This topic lists documentation changes in the June 02, 2017 *Server Troubleshooting Guide* documentation release for Ericsson Mediaroom Server 3.0 FP1.

Using the Documentation

- Ericsson Mediaroom Server Troubleshooting Documentation (p. 44)

  Added section explaining contents of the three separate files included in Mediaroom Server Troubleshooting documentation.

Troubleshooting VOD

- VOD Playback - HEVC Video Underruns (p. 257)

  Added troubleshooting content for a scenario where video under-runs are observed while playing back an HEVC VOD asset. (TFS 3056820)

Troubleshooting DVR Scheduler

- Cloud DVR Mediaroom Data Services Monitoring Scenarios (p. 615)

  Added new topic and monitoring scenarios.

- Cloud DVR Branch and Service Group Monitoring Scenarios (p. 636)

  Added new topic and monitoring scenarios.

Troubleshooting Live Anytime

- Live2VOD Recording Server (p. 819)

Moved flow diagram information to "Live Anytime" in Functional Architecture (Ericsson Mediaroom Server 3.0 Reference Architecture Help).

December 15, 2015 Documentation Release

This topic lists documentation changes in the December 15, 2015 *Server Troubleshooting Guide* documentation release for Ericsson Mediaroom Server 3.0.

Using the Documentation

- Ericsson Mediaroom Server Troubleshooting Documentation (p. 44)

  Added section explaining contents of the three separate files included in Mediaroom Server Troubleshooting documentation.

- Detailed Data Flow Diagrams (p. 45)

  Added pointer to detailed data-flow diagrams (no longer in this documentation set).

Troubleshooting VOD

- Use RolesSetup and RepairGrants (p. 261)

  Deleted reference to ReportingStoreDB.

Troubleshooting EPG

- Troubleshooting Listings (p. 504)

  Changed the “EPG subsystem” to “Listings subsystem”.

- Introduction to Troubleshooting Listings (p. 505)

  Moved detailed information from this section, for inclusion in “Listings” in *Mediaroom Reference Architecture*.

- Listings Updater Application (p. 520)

Added note: "NOTE For Server 3.0, the Listings Updater application on the Server-Facing Branch and the Livebackend Controller machines will exit immediately in each run if "Listings Sync" is enabled (Enabled is set to true in ListingsSyncConfiguration, which means to generate the listings data in MDS)."

Troubleshooting DVR

- Introduction to Troubleshooting DVR Scheduler (p. 581)

  Changed the "EPG subsystem" to "Listings subsystem".

Troubleshooting Notifications

- serverlayout.xml (p. 665)

  Removed references to:

  - searchWS
  - tsMonitorPublic
  - vodCatalogWS
  - vodMapServerWS
  - Removed sessionKeyAuthority_KeyGenerator.xml topic.

Troubleshooting Reference

- KeyMonitor (p. 725)

  Added new component.

- ListingsSyncService (p. 517)

  Added new component.

- MDSListingsWS Web Service (p. 999)

  Added new component.

- MdsSlabGenerator (p. 1020)

  Added new component.

- TimeShiftComponents (p. 857)

Added new component.

Diagrams

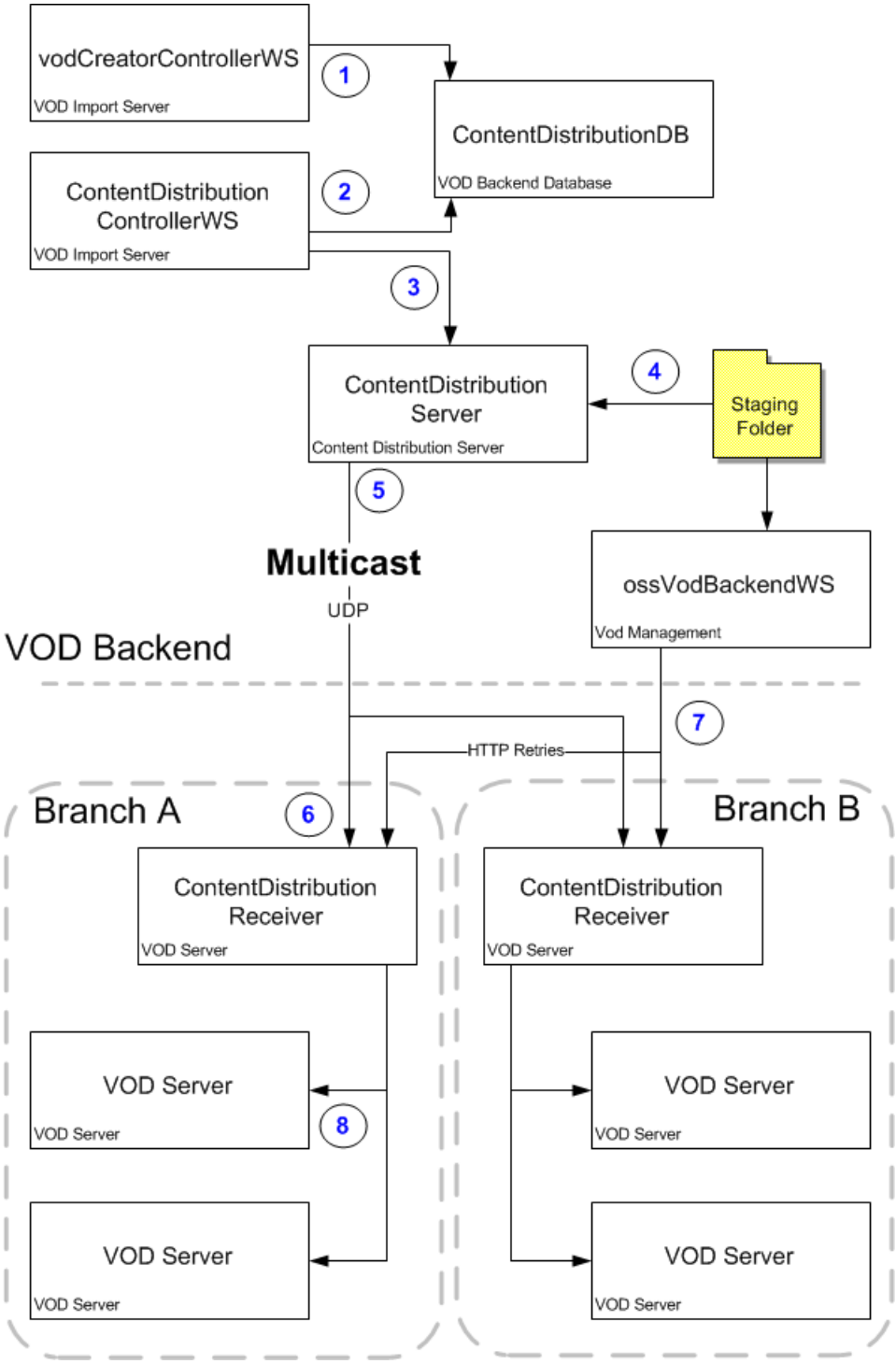
This section demonstrates design principles I developed to create logical, useful diagrams that enable customers to quickly understand data-flows between system components. The primary audience was system administrators tasked with troubleshooting issues.

Deploy VOD Asset to Branch Detailed Data Flow: Multicast Asset Distribution (Mediaroom Pre-Server 2.1 SP1)

Multicast Asset Distribution (MVD) uses multicast in the deployment of VOD assets from the backend to the branches. This method of multicast asset distribution results in a reduction of bandwidth requirement from the backend.

Note This topic applies only to Ericsson Mediaroom Server 2.1, prior to the SP1 updates to MVD. For Ericsson Mediaroom Server 2.5 MVD system flow, see Deploy VOD Asset to Branch Detailed Data Flow: Multicast Asset Distribution (p. 268).

The following diagram shows details of the deploy VOD asset to branch phase, with the addition of MVD enabled. The text that follows describes each numbered step of the diagram.



Deploy VOD asset to branch detailed data flow: multicast asset distribution

1. After an asset is imported by the VOD Creator Station, the VODCreatorControllerWS web service (vodCreatorControllerWS in the diagram)

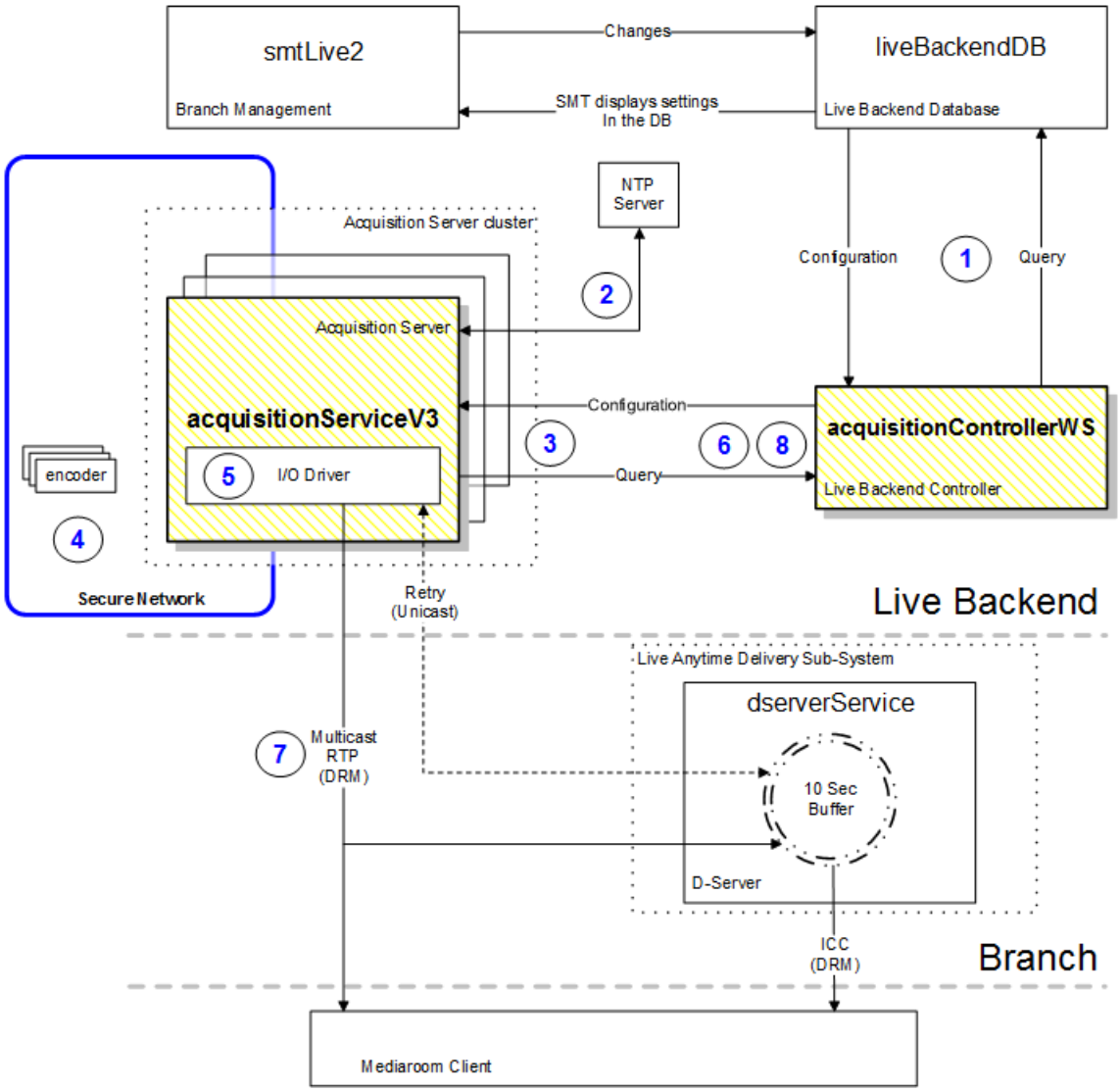
adds a new multicast asset distribution task to the ContentDistributionDB database.

2. The ContentDistributionControllerWS web service checks the ContentDistributionDB database for new tasks.
3. If the ContentDistributionDB database has a new task, the ContentDistributionControllerWS web service creates a job request manifest and sends it to an active Content Distribution Server.
4. The Content Distribution Server copies the asset files from the Staging Folder.
5. The Content Distribution Server sends the asset files to the multicast address.
6. The Content Distribution Receiver joins the multicast stream and receives the asset.

   Note The Content Distribution Receiver (also known as the Receiver client) gets the manifest by joining the manifest spool, which is on a separate multicast address. After the Receiver client receives the manifest, it starts saving the data, including the manifest file, to disk.

7. The Receiver clients can get the missing blocks through unicast retries, with an HTTP range request.
8. The Content Distribution Receiver copies the asset to the other VOD Servers in the cluster.

Normal Data Flow Between Acquisition Server and AcqCtrl Web Service



Acquisition Server and AcqCtrl web service data flow

1. When an Acquisition Server queries the AcqCtrl web service (role name acquisitionControllerWS) for configuration information, the AcqCtrl web service

reads source configuration data from the LiveBackend database and builds a table of Acquisition Servers that acquire live services and their assigned encoder multicast streams. Similarly, when an Acquisition Server requests keys (typically, because of a restart of the operating system), the AcqCtrl web service retrieves them from the LiveBackend database and returns them to the Acquisition Server.

Configuration details, including assignment of services, DRM parameters, and the IP addresses of associated encoders, are transferred to the individual Acquisition Servers. When an Acquisition Server starts up, it is assigned an "active" or "backup" role. If it is an active Acquisition Server, it continues to establish connections with the encoder, as described in the next phase, Phase 2. Process Services (p. 177).

If the Acquisition Server is assigned a "backup" role, it begins listening with its egress adapter to sample service streams from assigned Acquisition Servers, as described in Acquisition Server Failover Mode (p. 191).

When a new Acquisition Server is added to the live backend, the AcqCtrl web service directs it to start listening to a specific encoder IP address (source of MPEG-2 transport stream).

2. The Acquisition Server first establishes a stable NTP clock before transmission of any services, or prior to contacting the AcqCtrl web service. The total service startup time is less than 5 seconds, which enables fast recovery and support for redundancy. The 5-second services startup time assumes that the NTP clock is initialized, and that all services, keys and configurations are loaded to the Acquisition Server.

   Note As long as a stable NTP clock is not established, the Acquisition Server continues to query the NTP server in an effort to establish a stable clock. The Acquisition Server does not initialize other services until this task is complete.

   The Ericsson NTP client software synchronizes its clock to an NTP server clock by communicating through a network connection. The Acquisition Server uses its management interface (default) with the NTP server. Thus, the accuracy of the synchronization is limited by the quality of the network environment connected to the management interface. Computers linked with a high-quality network connection with moderate traffic will synchronize faster than will those with poor connections.

3. The Acquisition Server contacts the AcqCtrl web service and requests service configurations, keys, and DRM parameters. AcqCtrl web service retrieves the information as in step 1 and returns it to the Acquisition Server.

   The AcqCtrl web service is a stateless, cacheless "middleman" between the Acquisition Servers and the LiveBackend database. An AcqCtrl web service cannot initiate a connection to an Acquisition Server. All communication between an AcqCtrl web service and its Acquisition Servers are HTTP connections initiated by

an Acquisition Server. The AcqCtrl web service does not communicate to the LiveBackend database on its own behalf, but always because of a query posed by an Acquisition Server.

4. Following the operating system startup, the active Acquisition Server begins receiving and processing all assigned services. The Acquisition Server uses its ingress adapter to capture an unencrypted MPEG-2 transport stream from the encoder using the multicast IP address from the service configuration. The Acquisition Server can capture multiple MPEG-2 transport streams from multiple encoders.

   Encoders can deliver multi-program transport streams (MPTS). The full-screen and the PIP are commonly broadcast on the same multicast address from the same encoder, but are differentiated by the program number in the transport stream.

5. After the Acquisition Server joins the encoder's multicast stream, the I/O driver creates Mediaroom live TV services from the streaming content. Each MPEG-2 transport stream (an incoming, ingress source stream) from an encoder is assigned to a single Acquisition Server service.

   The acquisition subsystem performs the following operations:

   - Acquires PIPs directly from encoders.
   - Generates boundary keys, rotates the keys that are used for encryption on content boundaries as dictated by OSS and DRM, and stores the keys in a database. After the live TV service is deployed on the branch, a polling mechanism keeps the keys up to date at the branch.
   - Encrypts video and audio elementary streams (ES) (full-screen video and audio, secondary audio, and PIP video).

   A service can be assigned one or more full-screen services and one, multiple, or no PIP services. The processing of the ingress stream into the multi-streamed service is accomplished at the (NDIS) I/O driver level, which runs in kernel mode. The I/O driver is also responsible for encrypting packets.

   Note If the I/O driver becomes inoperable, the only recovery is to reboot the server.

6. Acquisition Servers periodically query the AcqCtrl web service for configuration updates from the LiveBackend database. Configuration details, including assignment of services and the IP addresses of associated encoders, are transferred to the individual Acquisition Servers.

7. A separate multicast address is assigned to individual full-screen streams and PIP streams. In normal operation, each packet in the RTP stream is encrypted with keys generated by the Acquisition Server; these keys are also called "boundary keys."

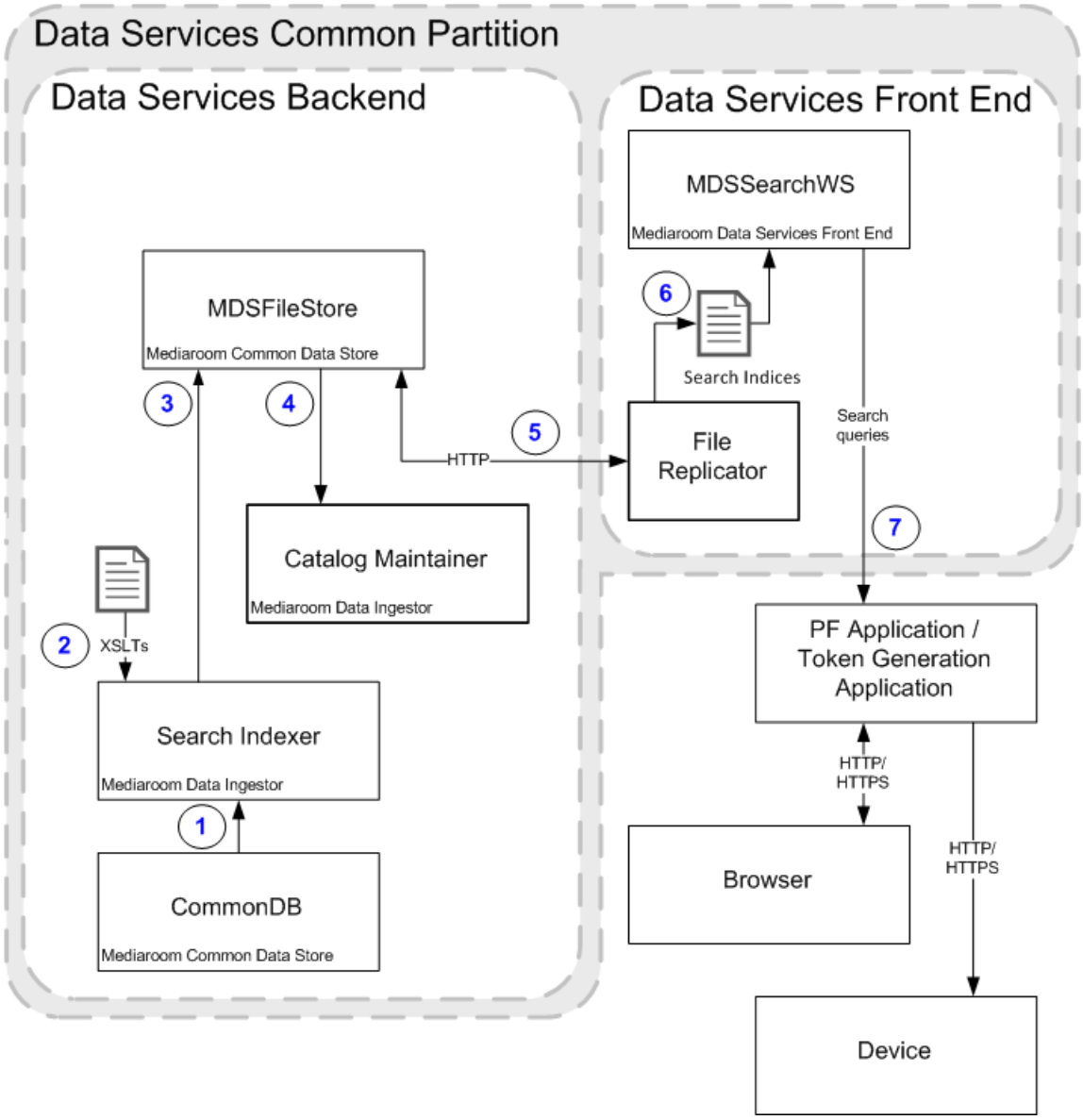
The acquisition subsystem performs the following operations:

- Encapsulates streams into RTP for multicast delivery to the live TV delivery subsystem and, depending on the configuration, to Mediaroom clients through the service provider's multicast-enabled network.
- Marks the RTP stream with the appropriate Macrovision analog content-protection control bits. The control bits instruct the Mediaroom client to add analog content protection to the outgoing analog live TV stream.
- Finds and marks RAP points in RTP headers.

8. Configuration details, including assignment of services and the IP addresses of associated encoders, are transferred to the individual Acquisition Servers.

Search: Detailed Data Flow

The following figure shows the functioning of search in the Data Services subsystem. Each number in the figure corresponds to a numbered item in the list that appears after the figure.



Data Services Search Detailed Data Flow

1. The search indexer component of the DataGeneratorService role periodically polls the CommonDB database for changes and pulls updates when it finds new data.

   The search indexer component generates a full rebuild of VOD and EPG data when there are changes in EPG data or if there are channel map changes from the branch. If there are no EPG data changes but only VOD data changes, a full rebuild is not triggered, and a smaller version containing only changed items, called a "delta," is generated. By default, deltas are generated every five minutes if changes are present. A separate index is built for each language supported in the Mediaroom Data Services Front End machine.

   For configuration details, see "MdsCommonDataAccessConfig" in the *Data Services Configuration Reference* (Mediaroom Server Operations Help).

2. The search indexer component of the DataGeneratorService both applies the XSLT files that are used to extract data to index and pulls out attributes that are exposed by the search function.

3. The search indexer component stores the updated search indexes, including some grant data, in the MDSFileStore (location is configurable). The VOD and Branch Ingestor components of the IngestionService, the search indexer component of the DataGenerator service, and the FragmentGeneratorService all write files to a folder specified in their respective configuration files on the machine hosting the MDSFileStore role (typically, this is the Mediaroom Common Data Store machine).

4. The Catalog Maintainer periodically checks the share on the MDSFileStore for new files. If the Catalog Maintainer finds a new version of a file (that is, search indexes or poster files), it updates its manifest with the new changes.

   When the manifest is updated, downstream clients of the Catalog Maintainer can discover files that have changed by referencing the updated manifest. Downloads of complete files are done rather than just partial file changes.

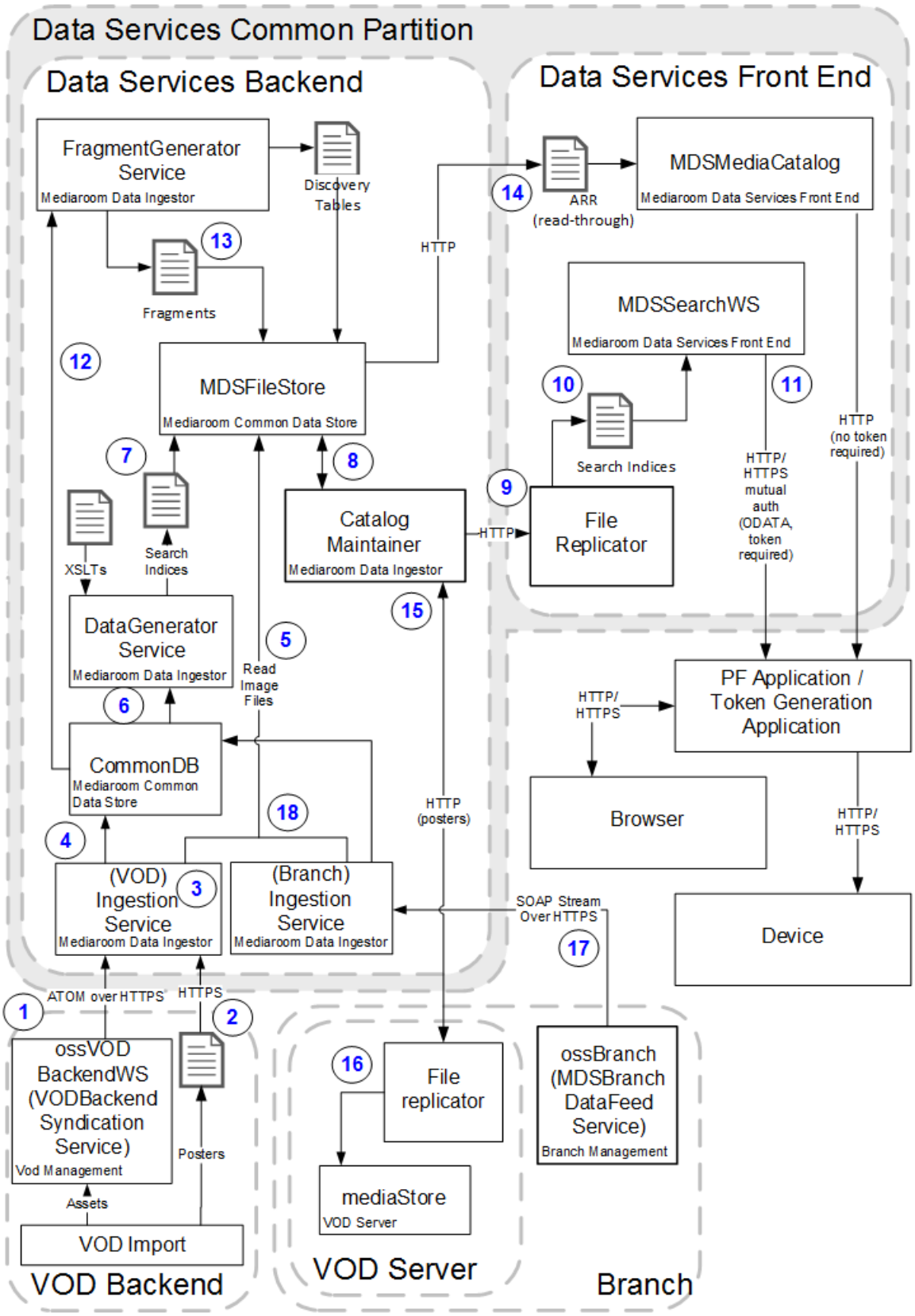
5. The File Replicator pulls the data from the MDSFileStore share via HTTP through the File Replicator virtual directory. In this way, downstream clients can get changes to the search indexes instead of having to download new copies each time there is a change.

6. The File Replicator in the Mediaroom Data Services Front End machine stores the new search indexes in a local folder where the MDSSearchWS web service can reference them.

7. After MDSSearchWS web service loads the new search indexes into memory, they are available to Mediaroom PF applications. Changing search indexes is transparent to the Mediaroom PF applications; that is, the Mediaroom PF applications do not recognize any versioning of the search indexes.

VOD: Detailed Data Flow

The following figure shows the functioning of VOD data (VOD metadata files and poster files) in the Data Services subsystem. Each number in the figure corresponds to a numbered item in the list that appears after the figure.


Data Services Common Partition
Data Services Backend
Data Services Front End
FragmentGenerator Service
Mediaroom Data Ingestor
Discovery Tables
Fragments
MDSFileStore
Mediaroom Common Data Store
MDSMediaCatalog
Mediaroom Data Services Front End
ARR (read-through)
MDSSearchWS
Mediaroom Data Services Front End
Search Indices
HTTP
HTTP (no token required)
HTTP/ HTTPS mutual auth (ODATA, token required)
Catalog Maintainer
Mediaroom Data Ingestor
File Replicator
XSLTs
Search Indices
DataGenerator Service
Mediaroom Data Ingestor
Read Image Files
CommonDB
Mediaroom Common Data Store
PF Application / Token Generation Application
HTTP/ HTTPS
Browser
HTTP/ HTTPS
Device
HTTP (posters)
(VOD) Ingestion Service
Mediaroom Data Ingestor
(Branch) Ingestion Service
Mediaroom Data Ingestor
SOAP Stream Over HTTPS
ATOM over HTTPS
HTTPS
ossVOD BackendWS (VODBackend Syndication Service)
Vod Management
Posters
Assets
VOD Import
VOD Backend
File replicator
mediaStore
VOD Server
ossBranch (MDSBranch DataFeed Service)
Branch Management
Branch

Data Services VOD Detailed Data Flow

1. VOD assets typically enter the Mediaroom system through the VOD backend pre-import and import process. The VOD Backend Syndication service (a component of the ossVodBackendWS role) monitors the Staging folder in the VOD backend for successfully imported changes and, using ATOM over HTTPS, publishes updates to the VOD asset metadata (assets that were added, modified or deleted) to the Ingestion service in the Data Services Backend machine.
2. After the VOD Ingestion service pulls the VOD asset metadata, the service also pulls poster files from the Staging folder to the VOD backend, using an HTTPS connection.
3. After the poster files are successfully loaded in memory at the VOD Ingestion service, the service references MDSImageProcessorConfig.xml and uses its image-resizing library to resize the posters based on poster type, such as poster, background image, and so on.

   For configuration details, see "MdsImageProcessorConfig" in *Data Services Configuration Reference* (Mediaroom Server Operations Help).
4. The VOD asset metadata is written to the CommonDB database.
5. The resized posters and other associated image files are sent to the MDSFileStore.
6. The search indexer component of the DataGeneratorService role periodically polls the CommonDB database for changes and pulls updates when it finds new data.

   The search indexer component generates a full rebuild of VOD and EPG data when there are changes in EPG data or if there are channel map changes from the branch. If there are no EPG data changes but only VOD data changes, a full rebuild is not triggered, and a smaller version containing only changed items, called a "delta," is generated. By default, deltas are generated every five minutes. A separate index is built for each language supported in the Mediaroom Data Services Front End machine.
7. The DataGeneratorService component stores the updated search indexes, including the grants data, in the MDSFileStore (location is configurable). The VOD and Branch Ingestor components of the IngestionService, the search indexer component of the DataGenerator service, and the FragmentGeneratorService all write files to a folder specified in their respective configuration files on the machine hosting the MDSFileStore role (typically, this is the Mediaroom Common Data Store machine).
8. The Catalog Maintainer periodically checks the share on the MDSFileStore for new files. If the Catalog Maintainer finds a new version of a file (that is, search indexes or poster files), it updates its manifest with the new changes.

When the manifest is updated, downstream clients of the Catalog Maintainer can discover files that have changed by referencing the updated manifest. Downloads of complete files are done rather than just partial file changes.

9. The File Replicator pulls the data from the MDSFileStore share via HTTP through the File Replicator virtual directory.
10. The File Replicator on the Mediaroom Data Services Front End machine stores the new search indexes in a local folder where the MDSSearchWS web service can reference them.
11. After the web service loads the new search indexes into memory, they are available to Mediaroom PF applications.

    Changing search indexes is transparent to the Mediaroom PF applications; that is, the Mediaroom PF applications do not recognize any versioning of the search indexes.
12. The Fragment Generator Windows service reads the CommonDB database and generates fragments. The Fragment Generator Windows service generates fragments for programs, categories, and stations.
13. The Fragment Generator Windows service stores the created fragments in the MDSFileStore (location is configurable).
14. The MDSMediaCatalog role uses ARR to proxy requests from the Mediaroom PF application directly to the MDSFileStore. For details about how ARR works, see step 11 of the topic Fragments: Detailed Data Flow (p. 316).
15. The Catalog Maintainer periodically checks the share on the MDSFileStore for new files. If the Catalog Maintainer finds a new version of a file (that is, search indexes or poster files), the Catalog Maintainer updates its manifest with the new changes.

    When the manifest is updated, clients of the Catalog Maintainer can discover files that have changed by referencing the updated manifest. Downloads of complete files are done rather than just partial file changes.
16. The File Replicator client on the VOD Server copies the file from the Catalog Maintainer to the Media Store folder on the VOD Server.
17. If the VOD asset is retrieved from the branch instead of the VOD backend, the (branch) IngestionService contacts the branch to find out which VOD Server has a copy of the poster file. If the (branch) IngestionService finds the poster file, it copies the file and resizes it by referencing MDSImageProcessorConfig.xml, and uses its image-resizing library to resize the posters based on image type, such as poster, background image, and so on.
18. The (branch) IngestionService stores the resized posters in the MDSFileStore.

Procedures

This section provides examples that demonstrate clarity in writing task-oriented documentation.

Operating the Ericsson Mediaroom Management Pack

This topic describes common procedures for setting up and using Ericsson Mediaroom Management Pack for Operations Manager 2012 R2, and for using the Operations Manager 2012 R2 Operations Console to manage Ericsson Mediaroom.

Viewing Ericsson Mediaroom Status in the Operations Manager 2012 R2 Operations Console

In the Operations Manager 2012 R2 Operations Console, the Monitoring View displays the status of different deployment locations (branch, live backend, VOD backend) and all the active alerts relating to Ericsson Mediaroom. The Monitoring View also displays the status of live TV services.

To view the Monitoring View in the Operations Manager 2012 Operations Console

1. In the **Operations Manager 2012 Operations Console**, click **Monitoring**.
2. In the **Monitoring** pane, expand **Monitoring**, and then expand **Ericsson Mediaroom** to see its contents.

   Note You can customize the fields that appear in this view by right-clicking anywhere in the right pane and clicking **Personalize** view.
3. To view the health state of a live TV service, perform the following:
   a. Select **Live Backend**, then select **Acquisition Server V3 Services**.
   b. In the **Detail View** pane, select the desired service to view details of its health state.
   c. Double-click the component to start **Health Explorer**.

      See Using Health Explorer (p. 86) for more information on **Health Explorer**.

4. To view the health status of the components in a deployment location, perform the following steps:
   a. Select the deployment location (branch, live backend, or VOD backend).
   b. In the **Detail View** pane, select the desired component to view details of its health state.
   c. Double-click the component to start **Health Explorer**.

Showing Ericsson Mediaroom Alerts

You can review and investigate active Mediaroom alerts in the Monitoring pane of the Operations Manager 2012 R2 Operations Console.

To view and investigate a Mediaroom alert

1. In the **Operations Manager 2012 Operations Console**, click **Monitoring**.
2. In the **Monitoring** pane, expand **Monitoring**, and then expand **Ericsson Mediaroom** to see its contents.
3. Click **Active Alerts**.
4. In the **Active Alerts** pane, click the alert to select it.

   Operations Manager 2012 R2 displays the alert details in the **Detail View** pane at the bottom of the screen. The details include which monitor is involved.
5. To view additional information about the alert and what may have caused it, click the **View additional knowledge** link.

   This link opens the **Alert Properties** dialog box and the **Product Knowledge** tab, where you can click the **Ericsson Mediaroom Documentation** link to view the **Product Knowledge** topic.
6. To diagnose and resolve the alert, in the **Actions** pane on the right of the screen, use the links to go to tools and scripts.

   Note On the toolbar, click **Actions** to display the **Actions** pane if the pane is not visible. The **Actions** pane includes links to the settings of the monitor, **Health Explorer**, the controls for **Maintenance Mode**, **Tasks** for the monitor, and additional resources and Help.

Organization

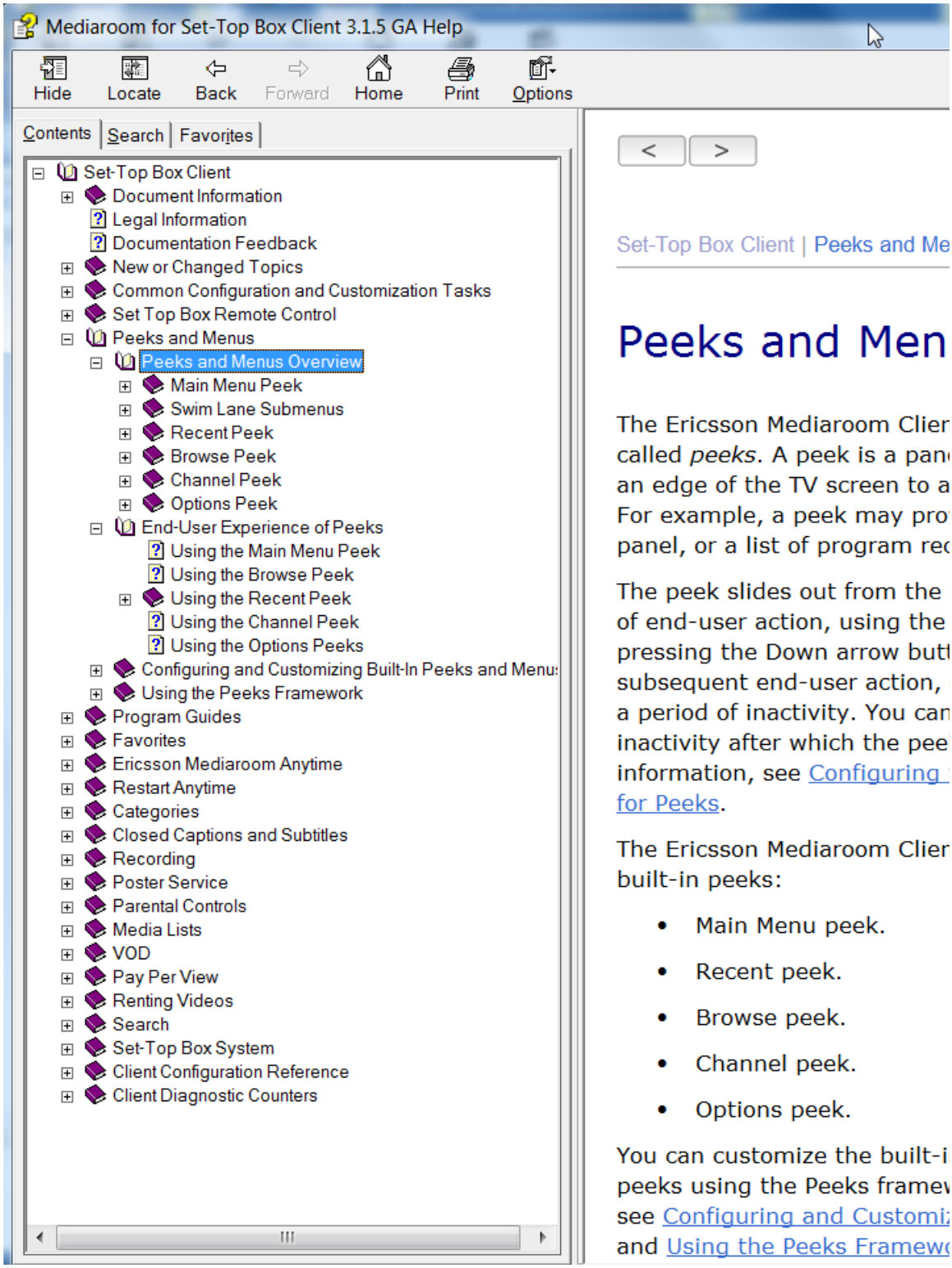
This section provides examples of high-level content organization.

High-Level Content Organization

When I was assigned to take over several documentation sets, I noticed that the top-down organization made it difficult to find content through navigation. I completely reorganized two books using a feature-oriented scheme. The new organization had the following advantages:

- Made locating content very simple and predictable for readers.
- Made finding a location for new content very easy for writers.
- Made the technical review process much easier, as all relevant content for a given feature was located in the same place.

The following diagrams demonstrate the new organizational paradigm that I introduced.


Mediaroom for Set-Top Box Client 3.1.5 GA Help
Hide
Locate
Back
Forward
Home
Print
Options
Contents
Search
Favorites
Set-Top Box Client
Document Information
Legal Information
Documentation Feedback
New or Changed Topics
Common Configuration and Customization Tasks
Set Top Box Remote Control
Peeks and Menus
Peeks and Menus Overview
Main Menu Peek
Swim Lane Submenus
Recent Peek
Browse Peek
Channel Peek
Options Peek
End-User Experience of Peeks
Using the Main Menu Peek
Using the Browse Peek
Using the Recent Peek
Using the Channel Peek
Using the Options Peeks
Configuring and Customizing Built-In Peeks and Menus
Using the Peeks Framework
Program Guides
Favorites
Ericsson Mediaroom Anytime
Restart Anytime
Categories
Closed Captions and Subtitles
Recording
Poster Service
Parental Controls
Media Lists
VOD
Pay Per View
Renting Videos
Search
Set-Top Box System
Client Configuration Reference
Client Diagnostic Counters
Set-Top Box Client | Peeks and Me
Peeks and Men
The Ericsson Mediaroom Clier
called peeks. A peek is a pan
an edge of the TV screen to a
For example, a peek may pro
panel, or a list of program rec
The peek slides out from the
of end-user action, using the
pressing the Down arrow butt
subsequent end-user action,
a period of inactivity. You can
inactivity after which the pee
information, see Configuring
for Peeks.
The Ericsson Mediaroom Clier
built-in peeks:
Main Menu peek.
Recent peek.
Browse peek.
Channel peek.
Options peek.
You can customize the built-i
peeks using the Peeks framew
see Configuring and Customi
and Using the Peeks Framew

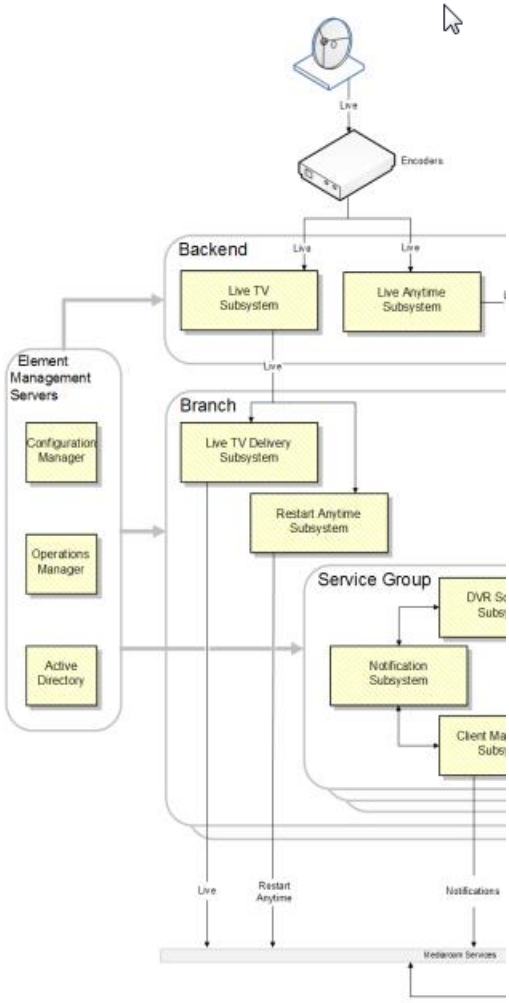
Ericsson Mediaroom Server 3.0 Reference Architecture (ERICSSON CONFIDENTIAL)

Hide Locate Back Forward Home Print Options

Contents Search Favorites

- Mediaroom Reference Architecture
  - Legal Information
  - Documentation Feedback
  - New or Changed Topics
  - Physical Architecture Overview
  - Functional Architecture
    - Using Functional Architecture
    - Subsystem Architecture
    - Live TV
    - DVR
    - Cloud DVR
    - NTP Server
    - Stream Management
    - VOD
    - Restart Anytime
    - Live Anytime
    - Mediaroom Data Services
    - Listings
    - Notification
    - High Priority Notifications
    - Web Service Router
    - Digital Rights Management and Keys
    - Client Devices
    - Client Management and Common Compone
    - User Store Subsystem and Client Store
    - Client and Server Logging
    - Branch Management
    - TV Services Management Tool
    - API Web Services for OSS/BSS
  - Active Directory Domain Services Architecture
  - Network Architecture
  - System Center EMS Architecture

The following diagram shows the bas
major Mediaroom subsystems.



The following table lists and describe
subsystems and major functional co
document.

Mediaroom subsystem / component	Description