KURENTO

Aarya Tapaswi

INTRO DUCTION

- multimedia server package
- used to develop advanced video applications for WebRTC platforms
- Open Source project
- It provides a set of capabilities and APIs (Application Programming Interfaces) that enable developers to incorporate features like **video streaming, video conferencing, and real-time communication** into their web or mobile applications.

FEATURES

1. Media pipeline

what?

a media pipeline is a logical chain of interconnected media processing elements. Each element in the pipeline performs a specific operation or task on the incoming media stream. These elements can include things like capturing video from a webcam, encoding and decoding media, applying filters or effects, mixing multiple media streams, and more.

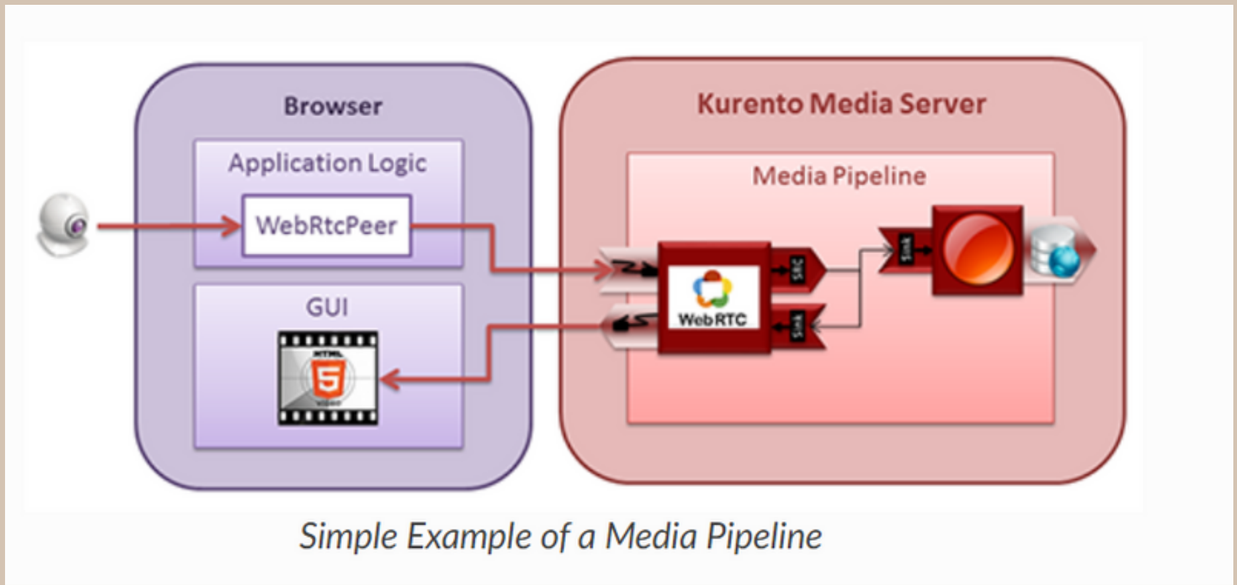
why?

Application developers use Kurento to control a so-called Media Pipeline with the desired Media Elements, effectively forming a fully customized architecture that is tailored to their needs.

how?

there can be MPEs for capturing video, audio, or both from a source, encoding the media, applying filters, or sending the media to an output destination.

you might start with a "WebRtcEndpoint" to capture media from a web browser, then connect it to an "RtpEndpoint" for RTP (Real-time Transport Protocol) streaming, and finally connect it to a "PlayerEndpoint" to display the media in a player.



Simple Example of a Media Pipeline

FEATURES

MEDIA ELEMENTS

- In Kurento, Media Elements are the fundamental building blocks for creating media processing pipelines.
- Each Media Element performs a specific action or function on a media stream.
- These Media Elements are designed to be self-contained "black boxes," abstracting away the low-level details and complexities of media processing for the application developer.

MEDIA ELEMENT TYPES

Input Endpoints

- Media Sources are Media Elements that capture or generate media streams from various input devices or sources.
- They serve as the starting point of a media processing pipeline, providing raw media data.
- Examples of Media Sources include:
  - WebRtcEndpoint: Captures media streams from web browsers using WebRTC, making it a source for audio and video input.
  - PlayerEndpoint: Acts as a source for pre-recorded media, allowing playback of existing audio and video files.
  - HttpEndpoint: Fetches media streams from external HTTP sources, such as URLs.

Fliters

- Filters: Media Elements in charge of transforming or analyzing media.
- Hence there are filters for performing operations such as mixing, muxing, analyzing, augmenting, etc.

MEDIA ELEMENT TYPES

Hubs

- Media Objects in charge of managing multiple media flows in a pipeline.
- A Hub contains a different HubPort for each one of the Media Elements that are connected.
- Depending on the Hub type, there are different ways to control the media.
- For example, there is a Hub called Composite that merges all input video streams in a unique output video stream, with all inputs arranged in a grid.

Output Endpoints

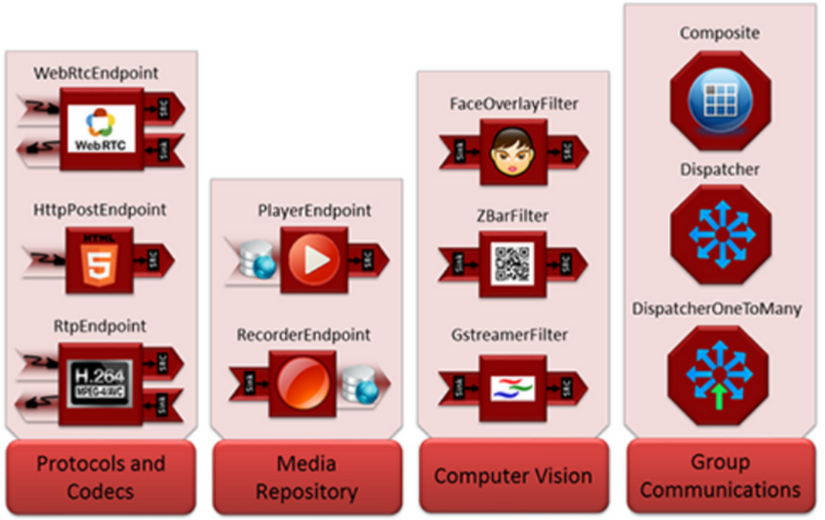
- Media Elements capable of taking a media stream out of the Media Pipeline.
- there are several types of output endpoints, specialized in files, network, screen, etc.

Examples

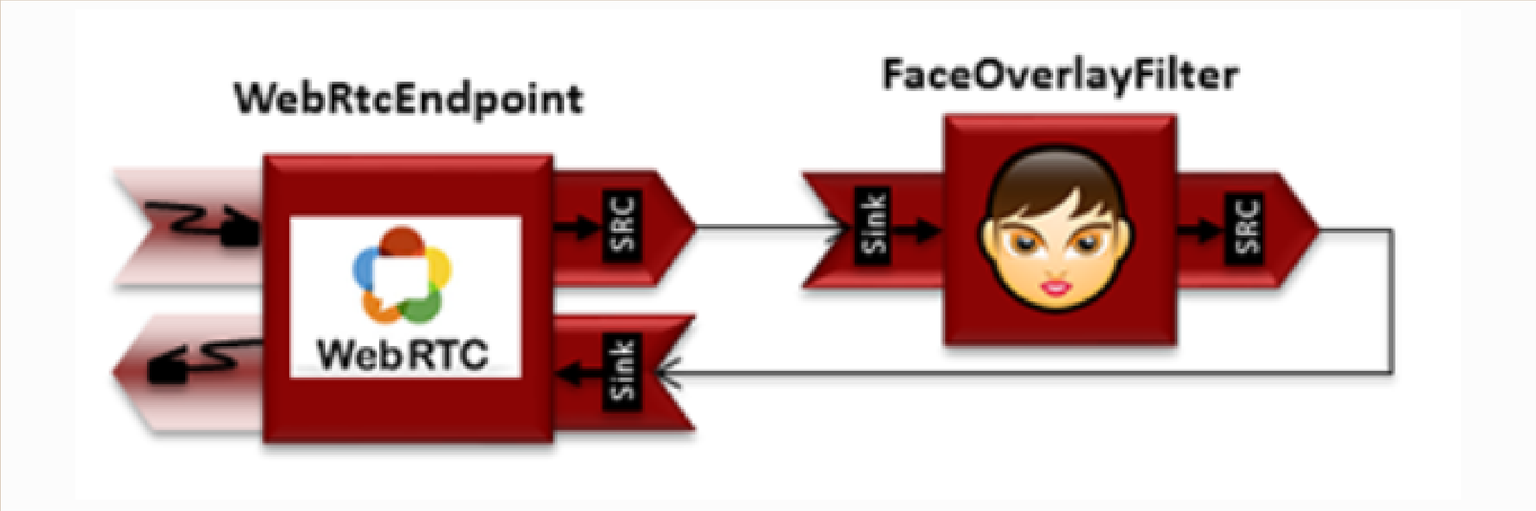
- The **WebRtcEndpoint** is able to send and receive WebRTC media streams.
- The **PlayerEndpoint** can be used to consume media from RTSP, HTTP, or local sources.

- The **RecorderEndpoint** can store media streams into a local or remote file system.
- The **FaceOverlayFilter** is a simple Computer Vision example that detects people's faces on the video streams, to add an overlay image on top of them.

MODULES



Some Media Elements provided out of the box by Kurento



Media Sources:

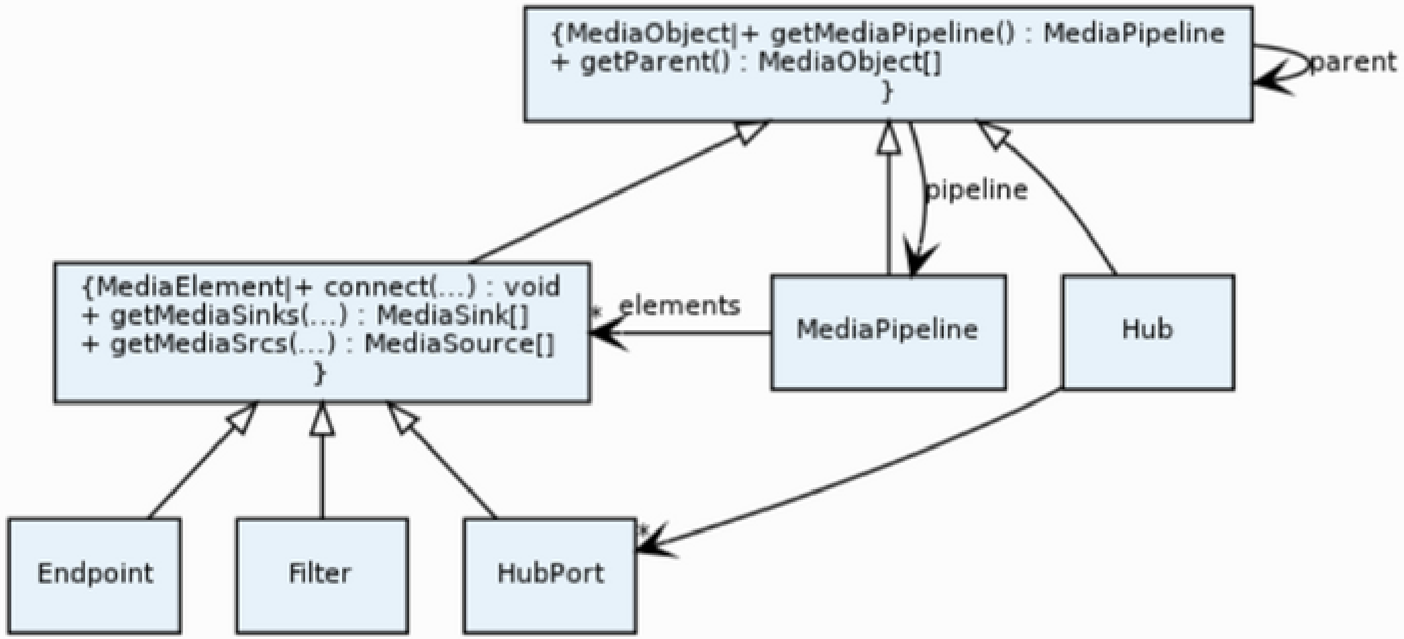
- Media Sources are Media Elements that capture or generate media streams from various input devices or sources.
- They serve as the starting point of a media processing pipeline, providing raw media data.
- Examples of Media Sources include:
  - WebRtcEndpoint: Captures media streams from web browsers using WebRTC, making it a source for audio and video input.
  - PlayerEndpoint: Acts as a source for pre-recorded media, allowing playback of existing audio and video files.
  - HttpEndpoint: Fetches media streams from external HTTP sources, such as URLs.

Media Sinks:

- Media Sinks are Media Elements that receive and process media streams, typically serving as the endpoint of a media processing pipeline.
- They consume and utilize media data sent by other elements for various purposes.
- Examples of Media Sinks include:
  - RtpEndpoint: Represents an endpoint for Real-time Transport Protocol (RTP) streaming, often used for sending media to external systems.
  - RecorderEndpoint: Records and stores media streams to local files or external storage, making it a sink for recorded media.
  - WebSocketSink: Sends media streams over WebSocket connections to be consumed by external applications or clients.

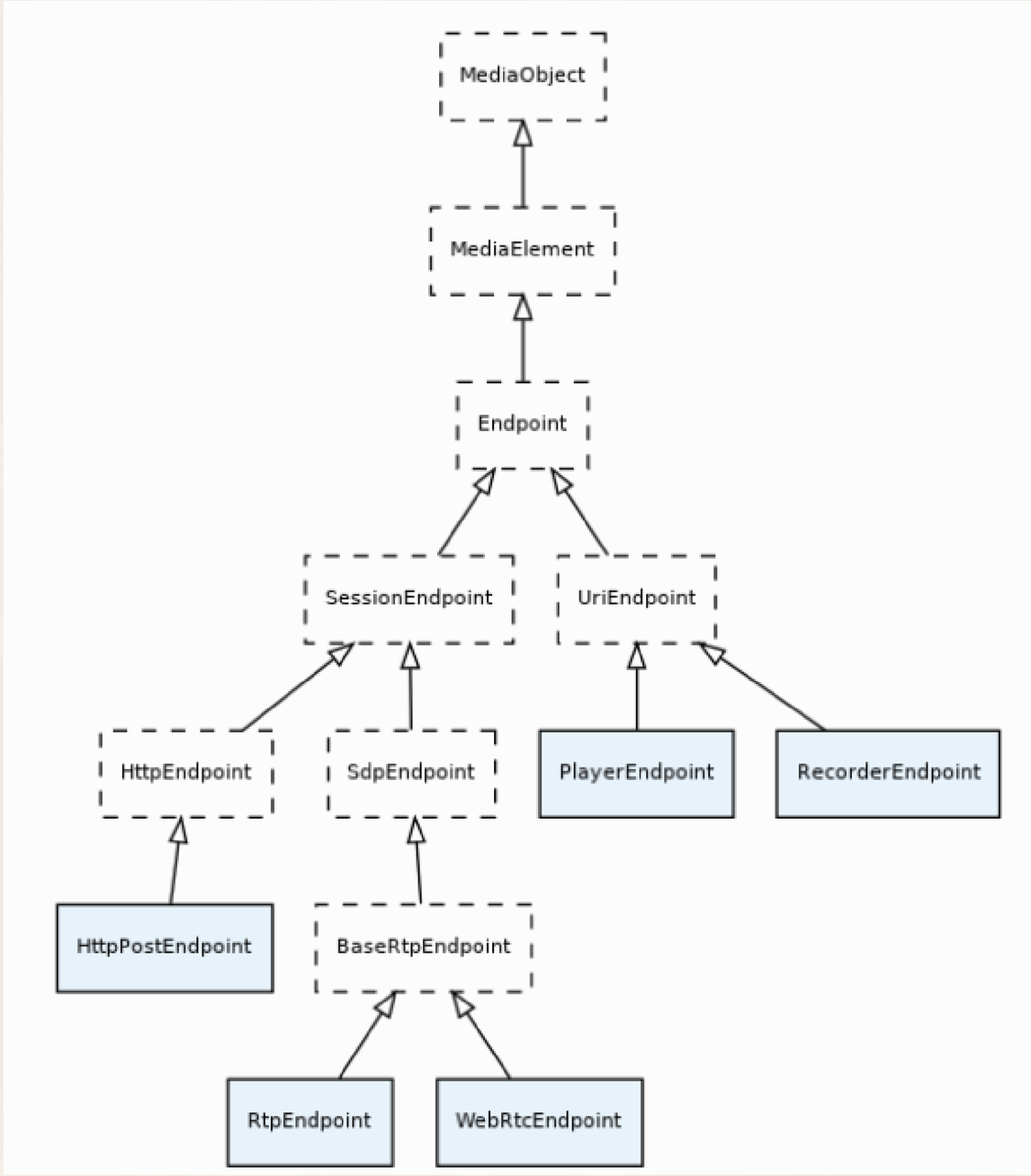
KURENTO

KURENTO API

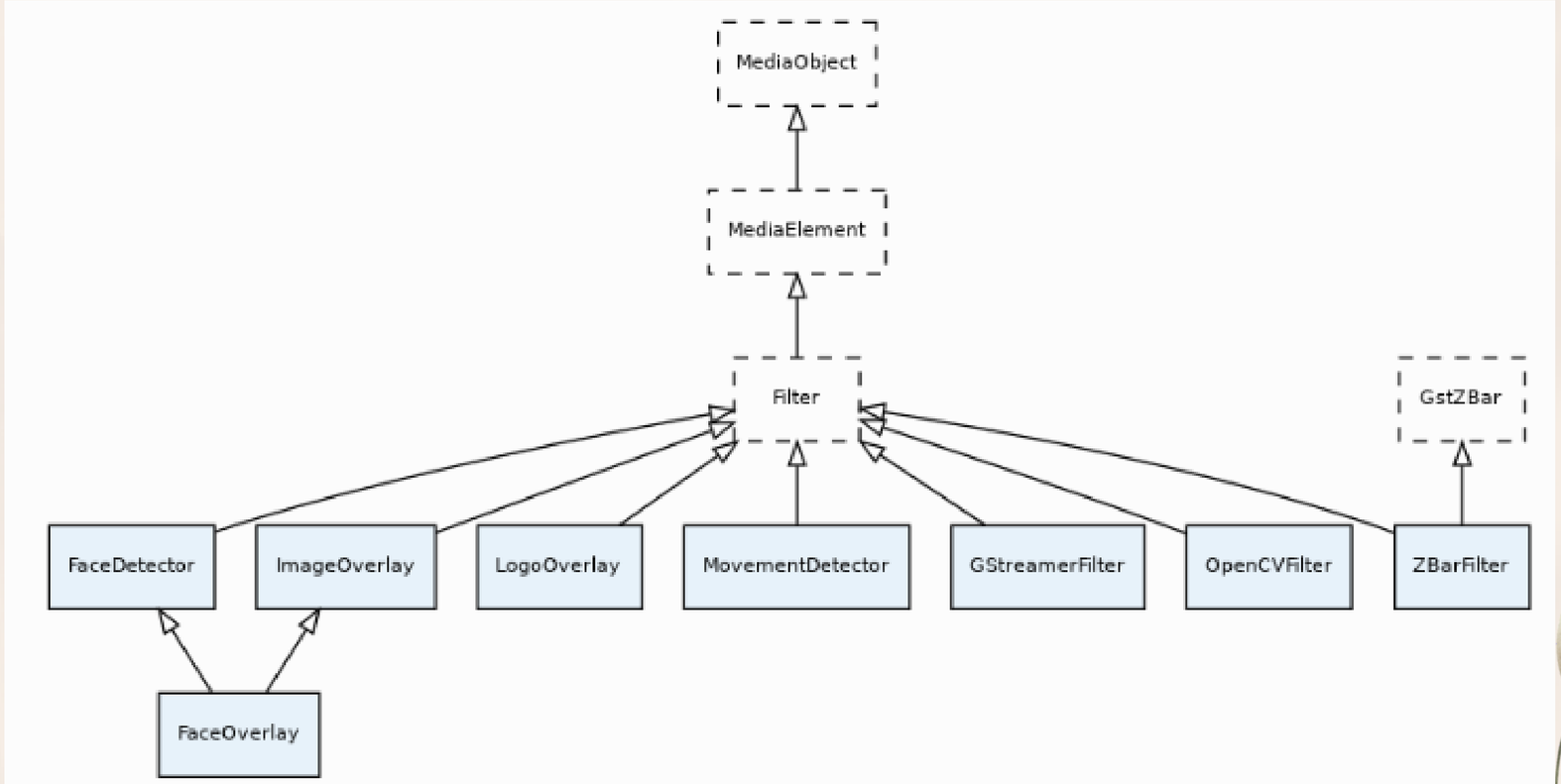


Class diagram of main classes in Kurento API

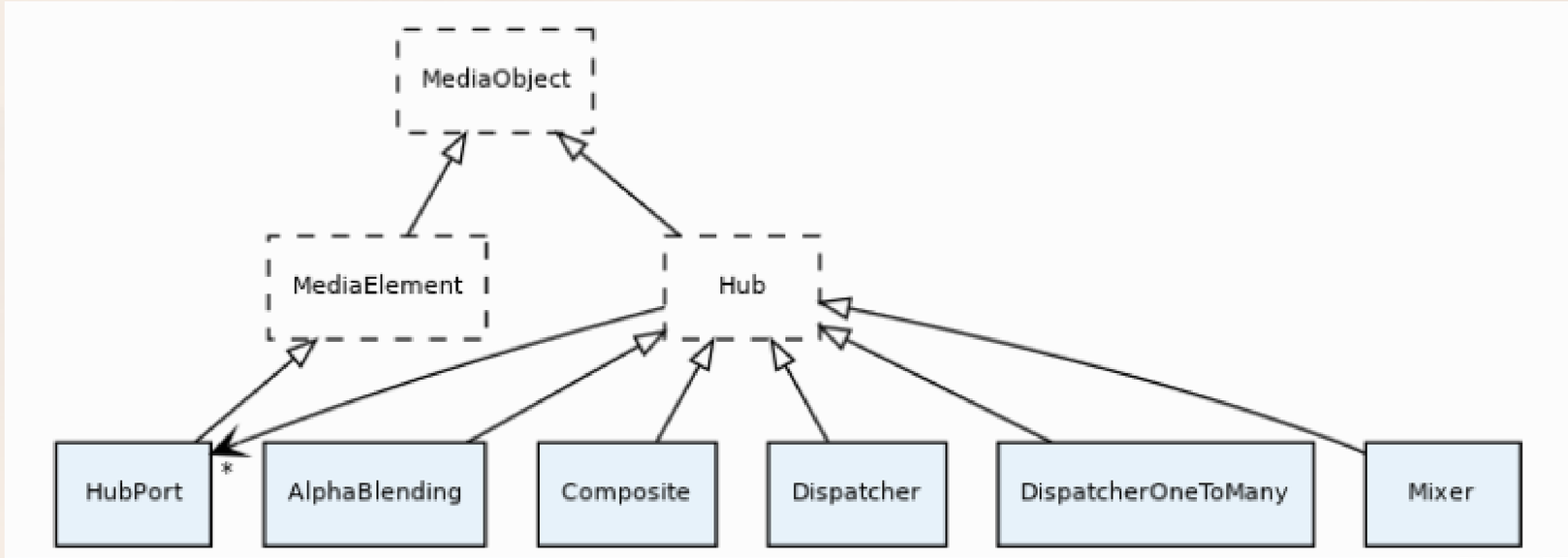
KURENTO ENDPOINTS



KURENTO FILTERS



KURENTO HUBS

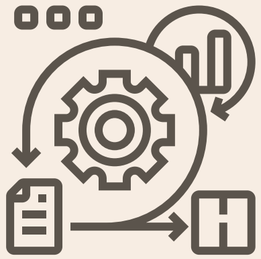


MCU VS SFU

MCU

MCU (Multipoint Control Unit) Topology:

1. Input Endpoints:
   - File Input Endpoint: Kurento doesn't have a specific "File Input Endpoint" as it primarily deals with real-time media. However, you can use custom code to read media from files and inject it into the pipeline.
   - Network Input Endpoint: In an MCU, you can use Kurento's "HttpEndpoint" to fetch media from external network sources.
   - Capture Input Endpoint: Capture Input Endpoints in Kurento can be represented by "WebRtcEndpoint," where each participant's camera and microphone capture their media.
2. Filters:
   - Mixing Filter: Kurento provides a "Composite" Hub that can be used for mixing multiple incoming video streams. It's part of the Hub functionality.
   - Audio Mixing Filter: Audio mixing can be achieved using audio processing elements in Kurento's pipeline.
3. Hubs:
   - Composite Hub: In Kurento, the "Composite" Hub can be used to merge video streams into a single output for MCU-like grid layouts.
4. Output Endpoints:
   - Network Output Endpoint: Kurento's "RtpEndpoint" can be used to send the mixed media stream over the network to participants in an MCU topology.
   - File Output Endpoint: Kurento's "RecorderEndpoint" can be used to record the mixed media to a file for archival purposes.

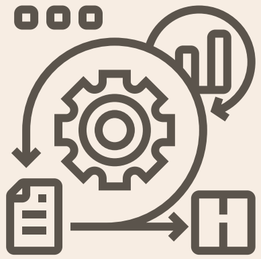


MCU VS SFU

SFU

SFU (Selective Forwarding Unit) Topology:

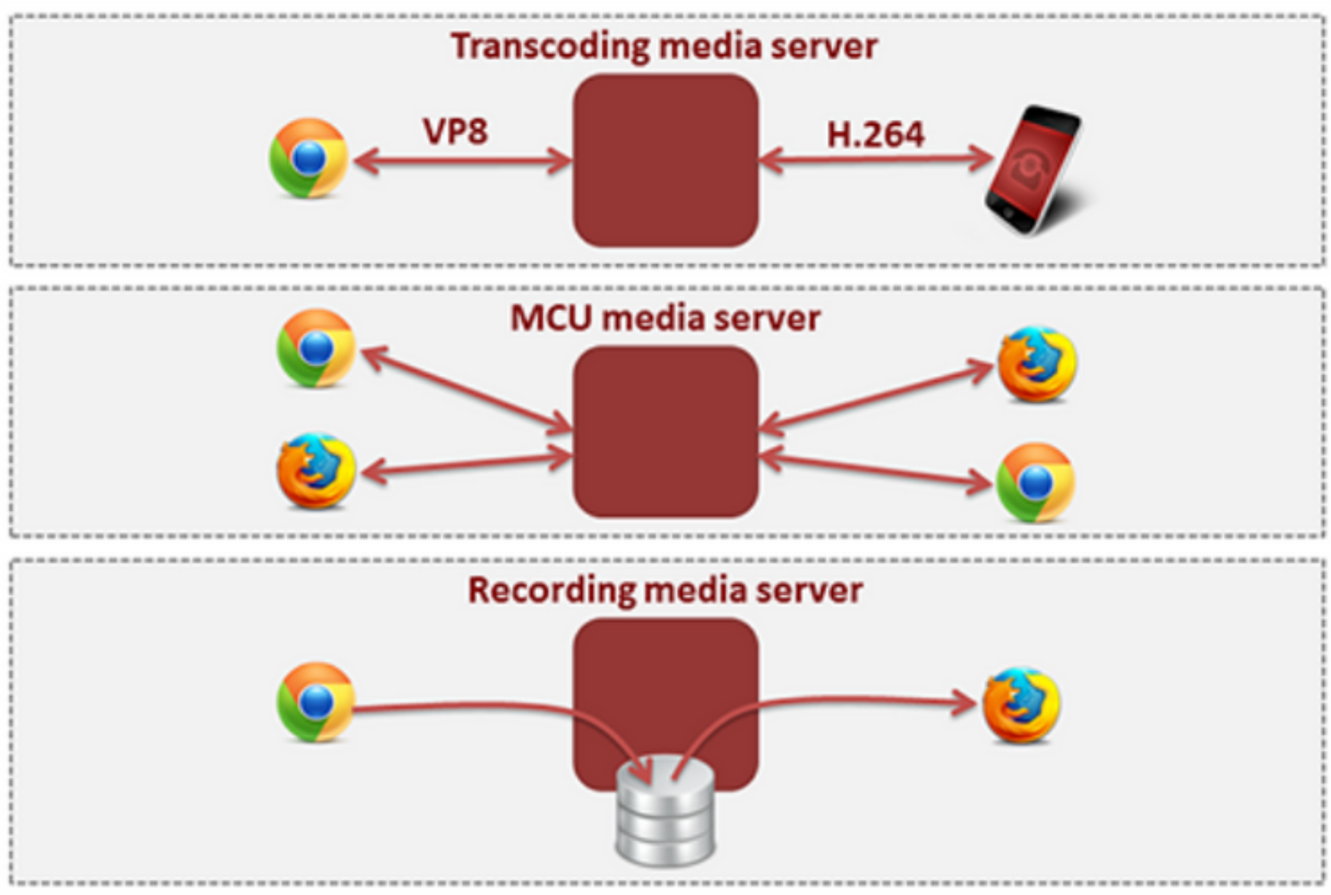
- Input Endpoints:
  - Network Input Endpoint: In an SFU, network input endpoints receive media from remote participants. You can use Kurento's "RtpEndpoint" for this purpose.
  - Capture Input Endpoint: Capture Input Endpoints represent the local participant's media in Kurento, typically done using "WebRtcEndpoint."
- Filters:
  - Selective Forwarding Filter: Kurento doesn't have a specific filter for SFU behavior since SFU functionality is achieved by controlling the routing of media streams dynamically.
- Hubs:
  - SFUs in Kurento typically don't use composite hubs like MCUs do. SFUs focus on selectively forwarding media streams to participants without composite operations.
- Output Endpoints:
  - Network Output Endpoint: Kurento's "WebRtcEndpoint" can be used as a network output endpoint to forward selected media streams to participants based on their subscriptions.
  - File Output Endpoint: Similar to MCUs, Kurento's "RecorderEndpoint" can be used for recording individual media streams in an SFU, but it's less common.



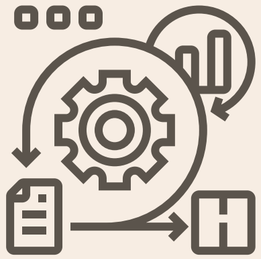
ADVANTAGES

Media servers are capable of processing incoming media streams and offer different outcomes, such as:

- Group Communications: Distributing among several receivers the media stream that one peer generates, i.e. acting as a Multi-Conference Unit (“MCU”).
- Mixing: Transforming several incoming stream into one single composite stream.
- Transcoding: On-the-fly adaptation of codecs and formats between incompatible clients.
- Recording: Storing in a persistent way the media exchanged among peers.



Typical WebRTC Media Server capabilities



ADVANTAGES

It provides the following features:

- Networked streaming protocols, including HTTP, RTP and WebRTC.
- Group communications (both MCU and SFU functionality) supporting media mixing and media routing/dispatching.
- Generic support for filters implementing Computer Vision and Augmented Reality algorithms.
- Media storage that supports writing operations for WebM and MP4 and playing in all formats supported by GStreamer.
- Automatic media transcoding between any of the codecs supported by GStreamer, including VP8, H.264, H.263, AMR, OPUS, Speex, G.711, and more.

What Kurento Media Server adds:

- Flexible processing
- Augmented reality
- Blending
- Mixing
- Analyzing
- Etc.

Kurento Media Server capabilities

OPENVIDU

EXTRA

What?

OpenVidu is an open-source platform for building and deploying video conferencing and real-time communication applications. It is designed to simplify the development of applications that require real-time video and audio communication features, such as video conferencing, online education, telehealth, and more. OpenVidu provides a set of APIs and tools that make it easier for developers to integrate video communication capabilities into their web or mobile applications.

THANK YOU