

How GStreamer is used in WebKit

GStreamer Conference

October 2026



Contents

- WebKit basics
- GStreamer in WebKit
 - Regular video playback
 - Media Source Extensions video playback
 - Encrypted Media Extensions
 - WebAudio
 - MediaStream
 - WebRTC

WebKit basics

WebKit code tips

- **Client class** pattern
 - Lower layer gets access to a subset of an upper layer class
 - Upper layer class shows itself as a client of the lower layer class by implementing the Client interface
- **Private class** pattern
 - Hides platform specific code to multiplatform code
 - There can be multiple platform specific implementations, sometimes at runtime

WebKit GStreamer code tips

- **Private class** pattern
 - Hides implementation code
 - Eg: WebKitWebSrc & WebKitWebSrcPrivate
- GStreamer code can be called from **multiple threads**
 - **DataMutex** & **DataMutexLocker** helps you organize lock granularity in multithread code
 - Thread asserts give hints about which threads the methods are supposed to run

GStreamer in WebKit

Regular video playback

WebKit regular video

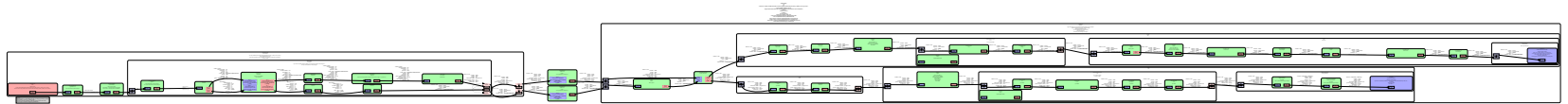
- **HTMLMediaElement W3C spec**
- Most relevant classes involved:
 - **HTMLMediaElement**: Implements the spec
 - **MediaPlayer**: Multiplatform player behaviour
 - **MediaPlayerPrivateGStreamer**: GStreamer specific internal implementation.
 - Manages a **playbin**-based pipeline.
 - **WebKitWebSrc**: GStreamer source element for regular video playback.

WebKit regular video classes

Regular video pipeline

- **Playbin** autoplug several relevant elements:
 - UriDecodeBin
 - **WebKitWebSrc**
 - DownloadBuffer
 - DecodeBin
 - **Demuxer** (QtDemux, MatroskaDemux, ...)
 - A/V parsers
 - A/V decoders
 - **PlaySink**
 - Audio sink (AutoAudioSink → PulseSink)

Regular video pipeline



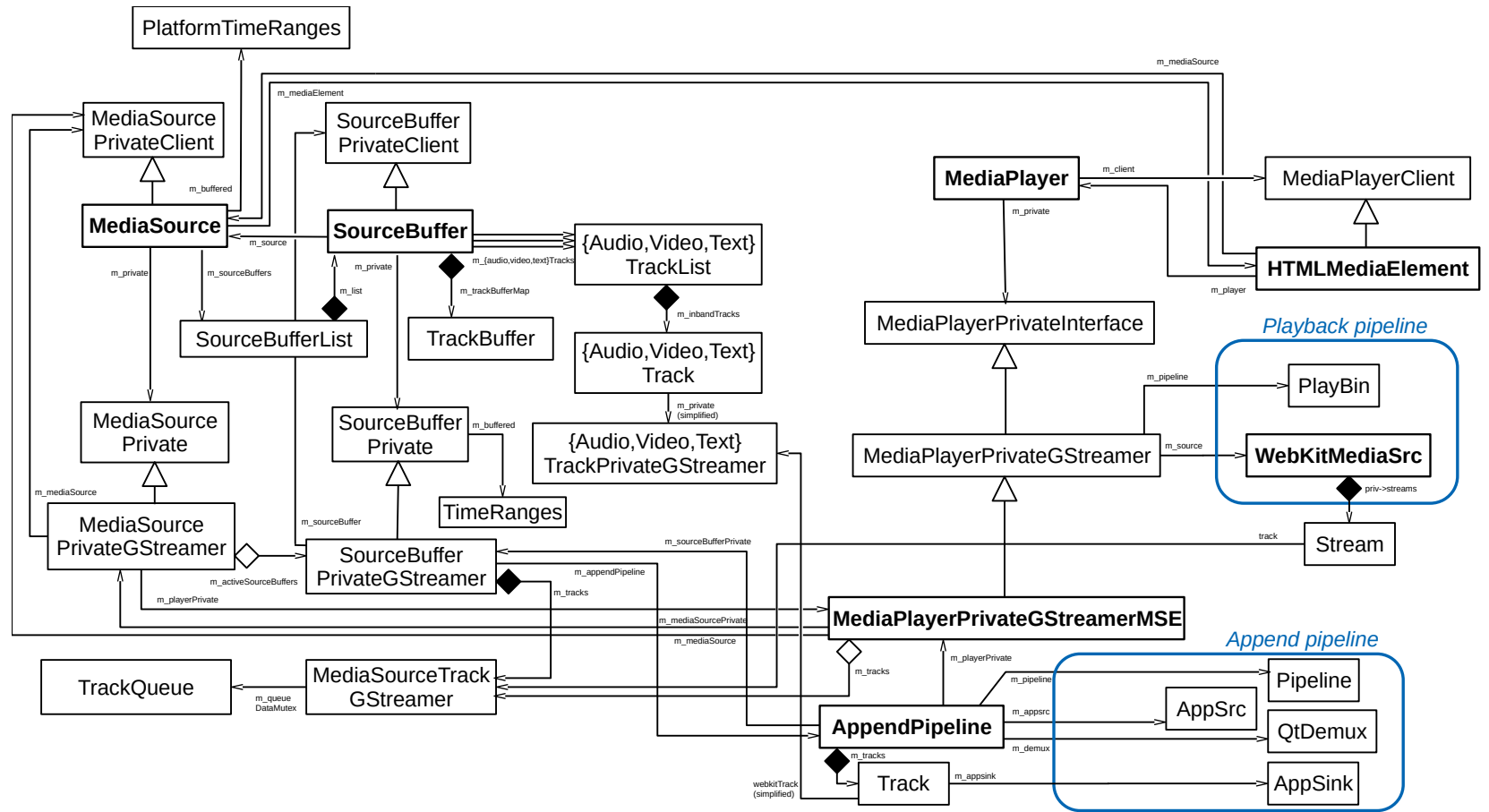
GStreamer in WebKit

Media Source Extensions (MSE)

Media Source Extensions

- Purpose: Allow JavaScript to feed data to media streams for playback
- **MSE W3C spec**
- Most relevant new classes involved:
 - **MediaSource, SourceBuffer**
 - **MediaSourcePrivateGStreamer, SourceBufferPrivateGStreamer, MediaPlayerPrivateGStreamerMSE**
 - **AppendPipeline**

WebKit MSE video classes



Append pipeline

- **One** append pipeline **per SourceBuffer**
- **Preprocesses** raw mp4/webm data appended to the SourceBuffer
- AppendPipeline exposes streams to the SourceBuffer as **Tracks**

Playback pipeline

- Same pipeline as in the regular player, but uses **WebKitMediaSrc**
- SourceBuffer cooperates with the player to **enqueue** media data to the playback pipeline
- SourceBuffer places data in the **TrackQueues** (main thread)
- **WebKitMediaSrc** picks the data from the TrackQueue (streaming thread)

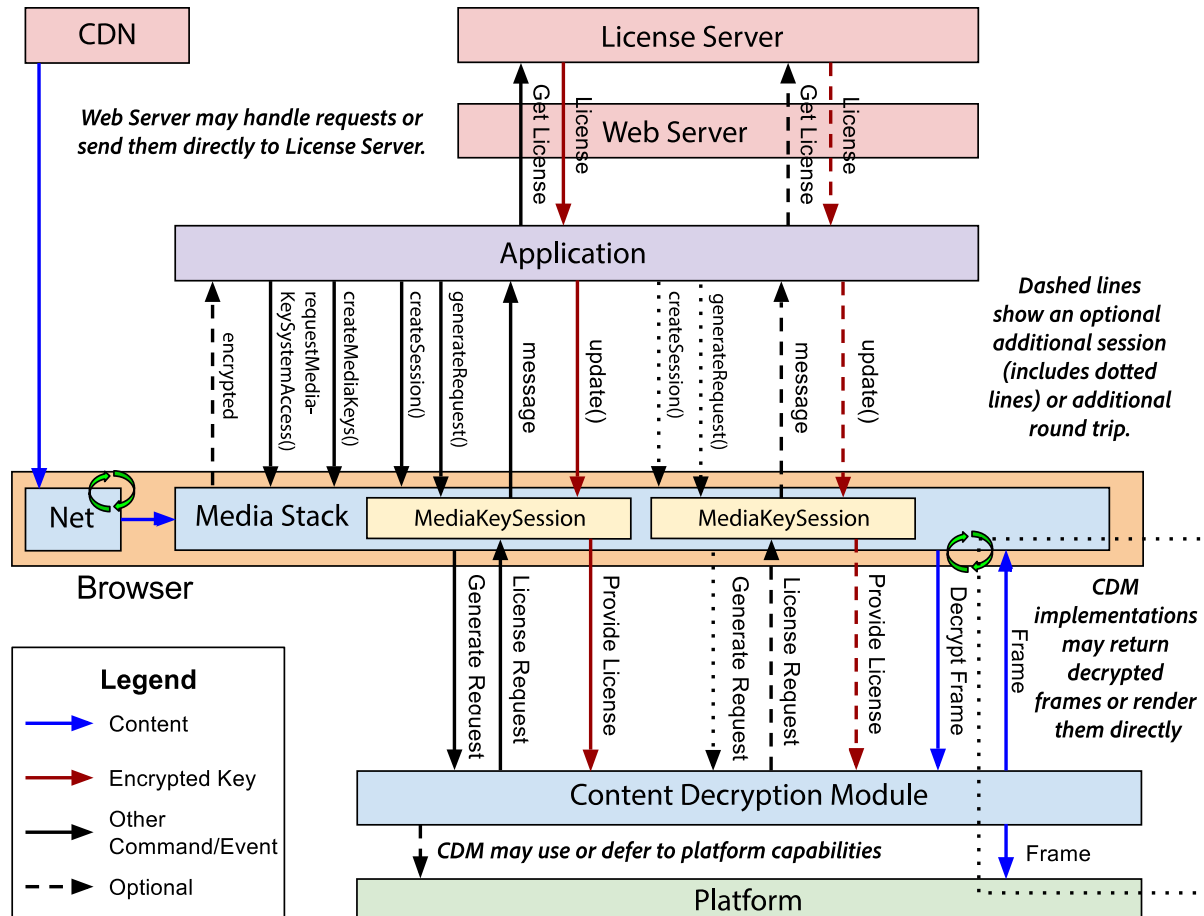
GStreamer in WebKit

Encrypted Media Extensions (EME)

EME

- Purpose: Enforce distribution rights by using encrypted video.
- **EME W3C spec**
- Most relevant new classes involved:
 - **NavigatorEME, MediaKeySystemAccess**
 - **MediaKeys[ystem]**
 - **MediaKeySession**
 - **CDM, CDMPrivate, CDMInstance, CDMInstanceSession** (also **proxies** for multithread/multiprocess interaction)

EME Stack Overview



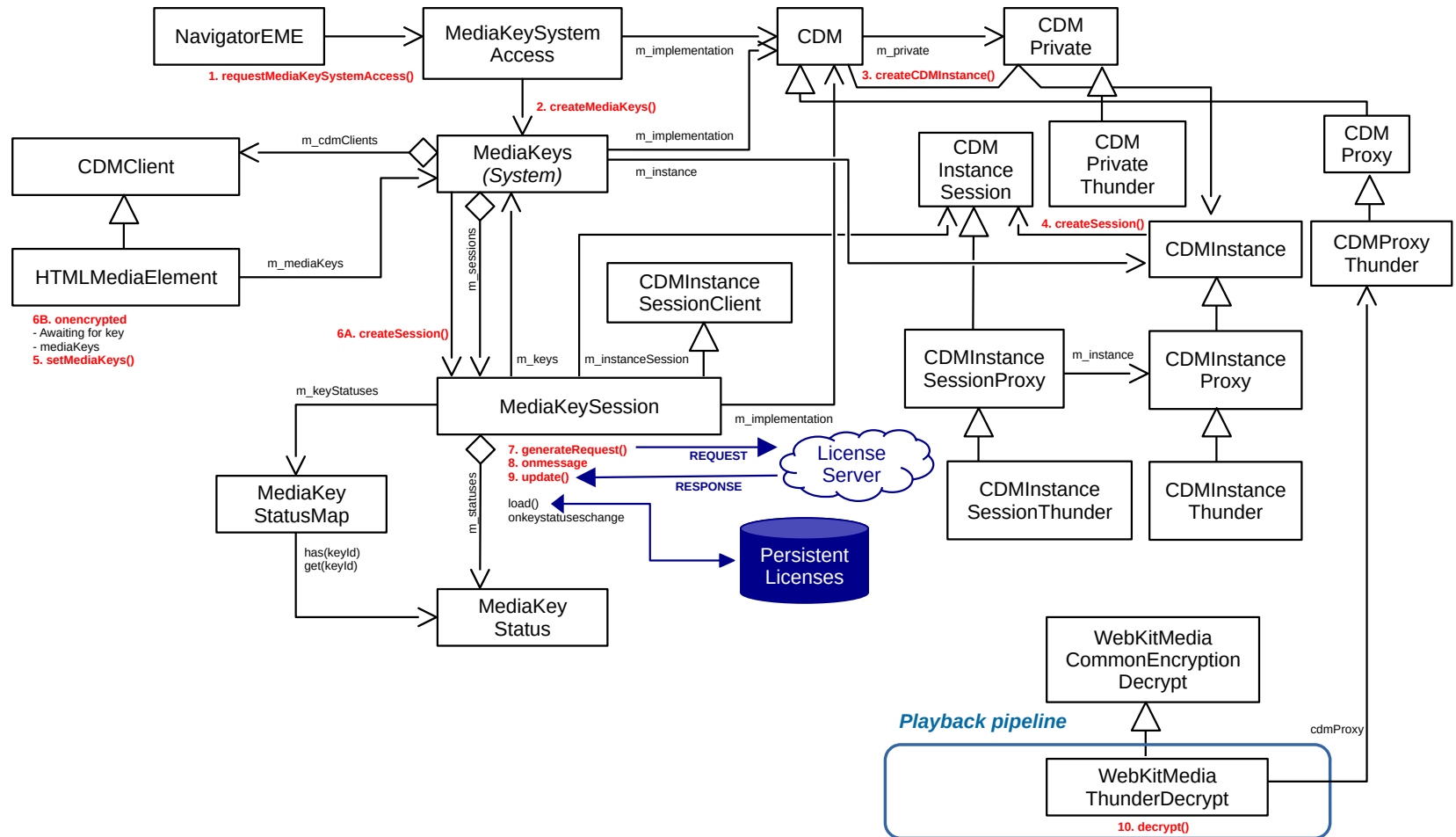
EME implementations in WebKit

- ThunderCDM
 - An evolution of OpenCDM
 - Used by RDK companies on set-top-boxes
 - Requires proprietary plugins, in cooperation with vendors
- SparkleCDM
 - API compatible with ThunderCDM

Thunder EME

- Connects to an external **ThunderCDM** process, developed by Comcast.
- Able to decrypt **PlayReady** and **Widevine**.
- Derived classes:
 - **CDM** → **CDMProxy** → **CDMProxyThunder**
 - **CDMPrivate** → **CDMPrivateThunder**
 - **CDMInstance** → **CDMInstanceProxy** → **CDMInstanceThunder**
 - **CDMInstanceSession** → **CDMInstanceSessionProxy** → **CDMInstanceSessionThunder**

WebKit + Thunder classes



GStreamer in WebKit

WebAudio

WebAudio

- Purpose: Processing and synthesizing audio in web applications using a routing graph.
- **WebAudio spec**
- Example:

A few relevant classes

- **AudioContext**: Contains the graph of elements.
- **AudioNode**: Audio sources, outputs and processing nodes. Some implemented using **GStreamer**:
 - **DefaultAudioDestinationNode**: Plays (renders) audio.
 - **MediaElementAudioSourceNode**: Takes audio played by HTMLMediaElement.
 - **AudioBufferSourceNode**: Takes audio from a memory buffer.
 - Others: Use **FFT** provided by **GStreamer**.

GStreamer in WebKit

MediaStream

MediaStream

- Purpose: Manages flows of audio/video related data.
- **Media Capture and Streams spec**
- **Source** of streams:
 - Local media capture devices enumerated by **MediaDevices**.
 - Canvas & MediaElement, using `.captureStream()`
 - WebAudio, from **MediaStreamAudioDestinationNode**
 - WebRTC, using **RTCPeerConnection**

GStreamer in WebKit

WebRTC

WebRTC

- Purpose: To stream captured A/V streams and arbitrary data between browsers without requiring an intermediary.
- **WebRTC spec**
- A/V streams: **MediaStream**
- Arbitrary data: **RTCDataChannel**
- Many moving parts:
 - ICE, STUN and TURN to manage connectivity.
 - SDP for client capability negotiation.

Features using GStreamer

- All the part of A/V streams, using **MediaStream**.
- WebRTC itself has ~~two~~ **one** implementations
 - **Using libWebRTC**: GStreamer implements multimedia **management** and video **encoding/decoding**.
 - ~~**Using GstWebRTC**: GStreamer implemented the protocols.~~
 - Used the **GstWebRTCBin** element and other features from the **GstWebRTC** module from the



Questions?



