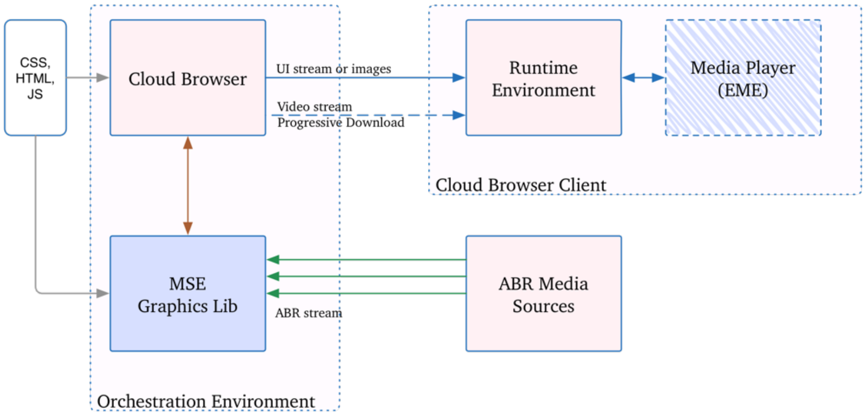
MSE Concepts for Cloud Browser

Alexandra Mikityuk (Deutsche Telekom), Ronen Mizrahi (TVersity)

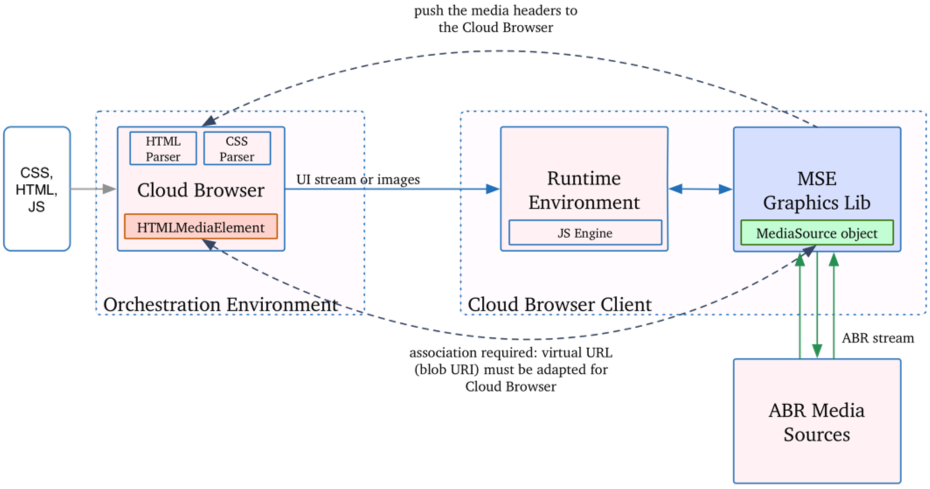
Document status: draft, in progress

Use Case 1: MSE Cloud Termination (full server-side processing)

UC	MSE Cloud Termination
Extends	none
Precondition	Cloud Browser client do not support MSE
Description	▪ Cloud Browser initiates a session with the Cloud Browser client ▪ Cloud Browser client requests a web page that uses MSE for video delivery ▪ Video is being delivered to the MSE component of Cloud Browser ▪ Cloud Browser terminates MSE and delivers the video to the client (progressive download) ▪ Delivery to the Cloud Browser client: • The video stream and the UI/Apps might be combined into one single video stream that is then delivered to the client; • The Video and UI/Apps streams might be also delivered separately to the client which then has to combine both these streams itself to present them to the end user in a unified form. ▪ Player • Player functionality is part of Cloud Browser • Player is required on the client side, if EME scenario considered  CSS, HTML, JS Cloud Browser UI stream or images Video stream Progressive Download Runtime Environment Media Player (EME) Cloud Browser Client MSE Graphics Lib ABR stream ABR Media Sources Orchestration Environment
Actors Server	▪ Cloud Browser ▪ MSE ▪ Graphics Lib
Actors Client	▪ rte ▪ Media Player (if required for EME) as part of the Cloud Browser client functionality
Dependencies	EME concepts for Cloud Browser
Addressed devices	Legacy STBs, low-end STBs, STBs with no MSE support
Architectures	This UC could be mapped on the following architectures ▪ Single Stream Local Player ▪ Single Stream Cloud Player

	- Double Stream Cloud Player
Gaps	none
Notes	none

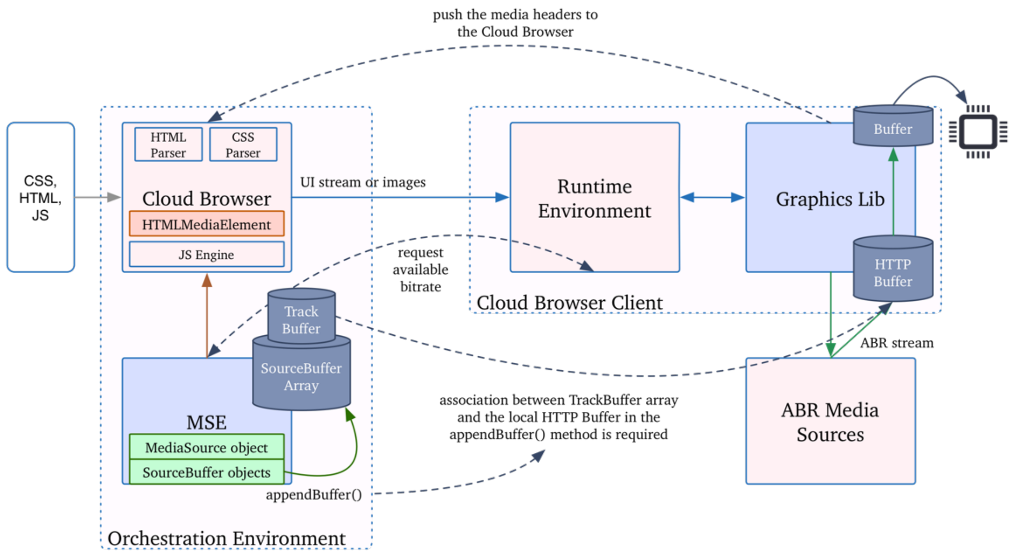
Use Case 2: MSE Client termination (full client-side processing)

UC	MSE client termination
Extends	none
Precondition	Cloud Browser (CB) client supports MSE
Description	- CB initiates a session with the CB client (session id, user id) - The CB client requests a web page that uses MSE for video delivery - CB parses the html and css data, sets the HTMLMediaElement, forwards the mpd manifest data to the CB client - The CB client requests the mpd file and parses it - The CB client creates mediaSource and associates it with HTMLMediaElement (scr for the HTMLMediaElement is MediaSource, currently associated through the virtual URL blob URI) - The CB client loads the init video segment and requests the video chunk, dependent on the available bitrate - Bitrate estimations run on the CB client - The CB client renders and displays the video - CB delivers UI/Apps the stream to the CB client - The CB client overlaps the UI/Apps video stream with ABR video, being played back locally  push the media headers to the Cloud Browser CSS, HTML, JS HTML Parser CSS Parser Cloud Browser HTMLMediaElement Orchestration Environment UI stream or images Runtime Environment JS Engine MSE Graphics Lib MediaSource object Cloud Browser Client association required: virtual URL (blob URI) must be adapted for Cloud Browser ABR stream ABR Media Sources
Actors Server	- Cloud Browser
Actors Client	- rte - MSE - Graphics Lib - JS engine
Dependencies	MSE
Addressed devices	STBs with MSE support
Architectures	This UC could be mapped on the following architectures - Double Stream Local Player
Gaps	- Manipulation with media file headers are not allowed by the MSE spec - HTMLMedia Element is currently not associated with the MediaSource object that is executed

	remotely ▪ CB currently can not distinguish the XHR requests from the CB client: whether the CB client requests the data from CB or from the other resources (CDN, Video Networks, etc.)
Reqs	Towards MSE ▪ The mechanism must be identified to enable manipulation with media file headers: headers must be pushed to CB for timestamps manipulations, current video positions, video time, etc. ▪ HTMLMediaElement must be associated with the MediaSource object that is executed remotely (IP, session id, user id, etc.) ▪ XMLHttpRequests must contain an identifier that would identify which kind of resource is requested
Notes	none

Use Case 3: MSE Client-Server processing (distributed MSE processing)

UC	MSE Client-Server processing
Extends	none
Precondition	▪ Cloud Browser (CB) client does not support MSE ▪ Double stream is required
Description	▪ CB initiates a session with the CB client (session id, user id) ▪ The CB client requests a web page that uses MSE for video delivery ▪ CB parses the html and css data, sets the HTMLMediaElement ▪ CB requests the mpd file and parses it ▪ CB creates mediaSource objects and associates it with HTMLMediaElement ▪ mediaSource creates sourceBuffer objects that in turn append media segments into the SourceBuffer array with the appenBuffer method ▪ The CB defines and forwards the media segment URLs to the CB client (with byte range params, video chunks id, etc.) ▪ The CB client loads the init video segment ▪ CB requests the available bitrate data from the CB client to define the following media segment URLs, dependent on the available bitrate ▪ The URLs are forwarded to the CB client ▪ The CB client requests video chunks and pushed the media headers back to CB ▪ Bitrate estimations run on CB that gets the required data from the CB client ▪ The SourceBuffer object at CB queues the media segments in the SourceBuffer array that is associated with the TrackBuffer ▪ The CB client requests the media data and buffers the data in the HTTP Buffer ▪ The appendBuffer method has a control over the HTTP Buffer (TrackBuffer is mapped with the HTTP Buffer) and gives a command which segments from the HTTP buffer go to the GPU for rendering ▪ The CB client renders and displays the video ▪ CB delivers UI/Apps the stream to the CB client ▪ The CB client overlaps the UI/Apps video stream with ABR video, being played back locally

	 push the media headers to the Cloud Browser CSS, HTML, JS HTML Parser CSS Parser Cloud Browser HTMLMediaElement JS Engine UI stream or images request available bitrate Runtime Environment Graphics Lib Buffer HTTP Buffer Cloud Browser Client Track Buffer SourceBuffer Array MSE MediaSource object SourceBuffer objects appendBuffer() association between TrackBuffer array and the local HTTP Buffer in the appendBuffer() method is required ABR stream ABR Media Sources Orchestration Environment
Actors Server	▪ Cloud Browser ▪ JS engine ▪ MSE
Actors Client	▪ rte ▪ Graphics Lib
Dependencies	MSE
Addressed devices	Legacy STBs, low-end STBs, STBs with no MSE support
Architectures	This UC could be mapped on the following architectures ▪ Double Stream Local Player
Gaps	▪ Manipulation with media file headers are not allowed by the MSE spec ▪ TrackBuffer is currently not associated with the remote HTTP Buffer in the appendBuffer() method ▪ CB currently can not distinguish the XHR requests from the CB client: whether the CB client requests the data from CB or from the other resources (CDN, Video Networks, etc.)
Reqs	Towards MSE ▪ The mechanism must be identified to enable manipulation with media file headers: headers must be pushed to CB for timestamps manipulations, current video positions, video time, etc. ▪ TrackBuffer must be associated with the remote HTTP Buffer in the appendBuffer() method (IP, session id, user id, etc.) ▪ XMLHttpRequests must contain an identifier that would identify which kind of resource is requested
Notes	none