XTRMX – DATA SHEET

GENERAL DESCRIPTION

XTRMX provides a simple framework for creating collaborative applications. The framework allows sharing data and especially media content, between multiple workstations in real-time. The shared content may be either local (e.g. on a local hard drive) or remote (e.g. on an accessible s3 repository),. The workstations may be spread out globally.

SYSTEM ARCHITECHTURE

The system is comprised of multiple client devices and a server machine.

An encoding component on the client device encodes the media's frames in a low-latency manner to transmit them on demand, either to the server or peer-to-peer to another client.

In a multi-device session, the stream might require re-transcoding into multiple formats in real-time. In that case, the stream will be transmitted to the server, re-transcoded in realtime to the required multiple-target-formats, and transmitted downstream to the clients.

The processing engine is fragmented in the sense that whatever processing is required – transcoding, transformations, analysis – it might be executed on any of the devices participating in the session.

KEY BENEFITS

Optimized quality, responsiveness & load-balance:
Distributing the signal processing among any number of those machines allows a trade-off between these characteristics:

- Quality: The processing can be executed on the media-hosting machine, thereby resulting in the highest quality.
- Responsiveness: The processing can be executed on the manipulation-triggering machine, thereby resulting in the highest responsiveness.
- Load-balance: The processing can be executed on the machine with the best adjusted capabilities (in terms of resources), thus optimizing the system resource-consumption.

Many-to-many streaming & control:
Users at different locations may manipulate media together, with no need to upload/download, and no device-related restrictions. A change done by any user is reflected in real-time to all others.

Media Content Location:
The media may be hosted on any accessible UNC or URL address – including local devices & S3-web-storage. As long as the address is accessible from a client workstation, this content may be added into an XTRMX session.

Browser and native based:
The client-side engine may be executed from the browser itself – in which case **no installation is required**, or with an XTRMX native extension, where an installation is required. The two flavors have an impact on the transcoding performance and on supported formats as detailed below.

TECHNICAL OVERVIEW

Supported video source codec / container:

Browser-based	Native-based
VP8 / WebM	VP8 / WebM
VP9 / WebM	VP9 / WebM
H264 / Mp4	H264 / Mp4
MPEG-4/Mp4	MPEG-4/Mp4
	Elementary H264
	Elementary H265
	ProRes*

supported audio source container / codec:

Browser-based	Native-based
PCM / WAV	PCM / WAV
Vorbis / WebM	Vorbis / WebM
Vorbis / Ogg	Vorbis / Ogg
Mpeg1L3/MP3	Mpeg1L3/MP3
AAC / MP4	AAC / MP4
FLAC / Ogg	FLAC / Ogg
	Mpeg1L2

Supported images:

Browser-based	Native-based
Jpeg	Jpeg
WebP	WebP
GIF	GIF
PNG	PNG
TIFF	TIFF

Incoming/outgoing streaming resolution compatibility with the device:
The following table presents XTRMX streaming capability benchmarks with a 29.97fps stream – both for incoming and outgoing streams – as derived from the workstation's specification. During a session, as the environmental conditions may change, the streaming codec parameters may be automatically adapted.
XTRMX works with several codecs, depending on the device capabilities and the bandwidth:

- A low-latency variation of an H265-compliant codec
- A low-latency variation of an H264-compliant codec
- An in-house implementation of inter-jpeg compression

	Browser-based encoding	Browser-based decoding	Native encoding	Native decoding
352X288 (CIF)	All devices	All devices	PC, Android, IOS*, MAC*, iPad*	PC, Android, IOS*, MAC*, iPad*
480x640 (SD)	>= i5 PC	All devices	PC, Android, IOS*, MAC*, iPad*	PC, Android, IOS*, MAC*, iPad*
704X576 (4CIF)	>= i7 PC	>= i5 PC	PC, Android, IOS*, MAC*, iPad*	PC, Android, IOS*, MAC*, iPad*
1280X720 (HD)	-	-	PC, MAC*	PC, Android, IOS*, MAC*, iPad*
1920X1080 (FHD)	-	-	PC, MAC*, Kepler/Maxwell/Pascal NVIDIA GPU	PC, Android, IOS*, MAC*, iPad*, NVIDIA GPU
3840X2160 (4K)	-	-	PC, MAC*, Kepler/Maxwell/Pascal NVIDIA GPU	PC, Android, IOS*, MAC*, iPad*, NVIDIA GPU

Server-side specification:
If required, the server can transcode the incoming streams into the formats required by the workstations, and distribute the streams to the target terminals. The capabilities of the servers are measured by the **number of concurrent sessions** it can run, given the required target formats, and associated with the server hardware specification:

Resolution/ Server specifications	i7, GTX1070	Xeon E5, K520	2 Xeon E5, 4 K520
4CIF	8	6	17
HD	5	2	7
FHD	2	1	2
HD Multiple targets	5	1	6
FHD Multiple targets	1	0	1

Bitrate:
The stream bit rate was measured at 29.97fps with an H265-based codec**.**

Resolution	Mbps
352X288 (CIF)	0.44
480x640 (SD)	0.9
704X576 (4CIF)	1.1
1280X720 (HD)	2.3
1920X1080 (FHD)	8
3840X2160 (4K)	28

Latency:
If the appropriate hardware settings for the required streaming format are set, then the low latency codec allows 40-60 milliseconds of latency for the transcoding chain. On top of that, the trip from one workstation to the other (possibly via XTRMX service) has to be added.

Collaborators:
There might be up to 10 concurrent collaborators in each session**.**

Security:

User Authentication
Users are registered to the system via a scheme that associates cardinals with unique, JWT-encrypted tokens.

Authorization-Roles
Each token belongs to a group, which in turn is associated with a set of administrator-configured authorization-roles. According to those roles, the user may or may not generate a new session.

Session-invitation
The user who generated the session may invite other collaborators to join the session. In order to invite a collaborator to join the session, the session-owner may choose one of two invitation-alternatives:

1) Authenticated invitation: The collaborator must be pre-registered to the system, and session-joining will be accomplished via the user's cardinals.*
2) Non-Authenticated invitation: The invitation is sent by mail to a collaborator (who is not necessarily registered to the system). The mail contains a link. Clicking on that link generates a session-specific, time-limited, JWT-encrypted, unique token, which is therefore non-transferable.

Encryption
In order to encrypt the session, the stream and data are both SSL-encrypted.
On top of the SSL encryption, the system uses an authenticated encrypted scheme (GCM, CCM, EAX)*.

Watermark
In order to protect the media stream further, the administrator may configure a watermark to be burned on the stream – to identify either the streaming source, the browsing client, or both.

Load balancing: In many cases, some collaborators use devices that are weak- considering the type of manipulations they wish to apply. In such cases, XTRMX API will transparently apply the manipulation by using another (capable) collaborator's device, or by a dedicated server. This characteristic allows the system also to trade-off responsiveness and quality.

Permissions: It is common for a collaborative application to have different permission types for different collaborators. Each user is associated with a group, which in turn is associated with a set of roles that restrict the user's manipulations.

Robustness: Sessions' manipulations and assets references are saved within the server's database. Upon abnormal termination (due to network failure, workstation-abnormal shutdown etc.), the session is recovered based on the database image.

Saving the Outcome: In case the media content of a session has to be transmitted to the same repository (commonly in order to be rendered), the original media will be uploaded in the background, when the network usage is idle. Since this process may be lengthy, it might extend beyond the period of the session itself.

In order to render, the server registers all changes made by the collaborators, so it may be applied on the uploaded high-resolution content when its upload has completed.

APPLICATIVE USAGE

XVIEW

xView is a professional review tool intended for the broadcast and the post-production industries. xView allows multiple users to review video content together, in real-time, regardless of their physical locations.

Collaborative Transport: Transport control usage (play, pause, scrub, frame+, frame-) is collaborated among the users. As one user manipulates the transport, all collaborators are updated in real-time.

Frame Accurate, Random-access stream: Transport control is frame-accurate, allowing frame-by-frame stepping without drop-frames.

Content Location: Reviewed video content is either remote (hosted in the cloud), or local. In both cases no preliminary upload/download is needed.

Frame Annotation: Textual and graphical frame-referenced annotations can be added.

Live Chat: Voice and textual chat are available.

Premiere integration: Allows reviewing and manipulating premiere content by another collaborator in a simultaneous, realtime manner. The premier and the browser may manipulate the EDL collaboratively, and use the xView panel – annotation and chat – through a premiere panel as well as from the browser.

XSTUDIO

xStudio allows multiple users to design and set-up a full virtual studio set collaboratively, in real-time, regardless of their physical locations, using media content originating from multiple sources.

Source: Any of the supported video, audio or image assets may reside either on a web-account or on the users' local devices. Wherever it's hosted, all collaborators may manipulate the entire sum of assets.

Filters: These are a vast array of filters and manipulations, to be used on any asset. Among them are position, scale, angle, warp, hue, opacity, contrast, brightness, z-indexing, chroma-keying, zooming and more.

The filters may be set by multiple users simultaneously, in a realtime fashion.

Selection: xStudio allows different users to select different assets and therefore set up multiple areas of the virtual studio simultaneously.

Ingest: Allows live capturing using either a browser or native-based components, from an Android, PC, MAC, iPad* and iPhone*.

High resolution streams can be captured from bluefish* or blackmagic* devices.

The live stream can be browsed and manipulated while capturing flows, by multiple users in a realtime, simultaneous, manner.

Outgoing streaming: The output stream can be routed either to the multiple collaborators or to a video-streaming bluefish* or blackmagic* device.

MXAPI

MX API is a JavaScript and C/C++ framework that aims to simplify the creation of collaborative applications. The framework allows sharing data (and media content in particular), between multiple workstations in real-time. The shared content may be either local (e.g. on a local hard drive) or remote (e.g. on an accessible s3 repository), and the workstations may be spread out globally.

Custom Data-Model: MX API allows the user to define a data-model, using the user's semantics. The data model is dynamically compiled into an API that will be used to modify the data-model and get notified once it has changed, in a collaborated, realtime, multi-user session.

Local and Remote MVC: The API encapsulates the need to update both local views and remote views that are bound to a given attribute in the data-model. This allows a given machine to update all its views that are bound to a given attribute, and all the views that correspond to that attribute in other machines, in one API call.

Immediate Local Updates: The API is internally aware of which views are remote (views that reside on another machine), and which views are local (views that reside on the same machine). This distinction allows the system to update all local views immediately (that is- synchronously, with no roundtrip to the server), therefore the user experience in this sense is optimal.

Media realtime collaboration: MX API allows multiple users to manipulate multiple media assets, regardless of where the content is hosted (on local devices, on the network or in the cloud) or where the users are, in a simultaneous, realtime fashion.

3rd party integration: The SDK allows the translation of 3rd party data models into the MX API data model, thustransforming any SDK-exposing application into a realtime-collaborated application, including media-realtime collaboration.