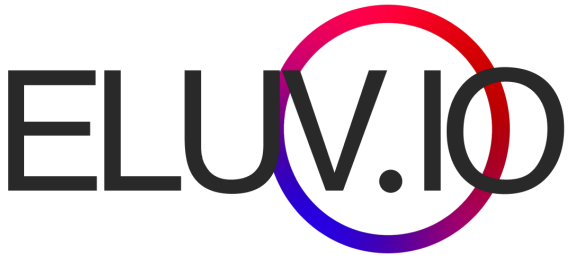


Core Audio/Video Streaming and Reporting Features

Last Updated November 2023

Table of Contents

Introduction

The Eluvio Content Fabric is an open and programmable global content streaming, delivery and storage network running the Content Fabric Protocol. The Fabric provides source to consumer content distribution globally including low latency 4K live streaming, premium video on demand, and composed (FAST) channels to standard video (DASH/HLS players) including all functions of audio and video transcoding, packaging and streaming distribution. Content encryption, industry-standard DRM, forensic watermarking, and authorization are built-in to the protocol and require no additional technology. The approach is dramatically *simpler, faster performing, and more efficient* than legacy CDNs and Media Clouds.

The platform is fully open with a complete API for video publishing, streaming, and content distribution, and provides a suite of applications and open source tools for publishers to manage premium video supply chains end-to-end. It also provides built in ML tagging, search and extensible content insertion and other server side content personalization functions. The programmable nature allows Eluvio (and others) to add specific features to the A/V pipeline and other capabilities with no forklift upgrades, and quickly.

For details on the Content Fabric Technology and Media Features please see https://live.eluv.io/content-fabric/technology and https://live.eluv.io/features/details. This product spec summarizes the key features and capabilities for premium video distribution.

Audio/Video Streaming Features

Core Features

The Fabric's native just-in-time audio/video pipeline in the core protocol creates and serves the playout manifest and playable streaming segments in its software dynamically (and on any node) such that any specific manifest capabilities can be readily added to the pipeline. The approach is also component based so different variants can be generated on demand.

The current production AV pipeline supports creation of standard DASH/HLS CMAF-packaged ABR manifests (fMP4, 1 frame per fragment) and playable segments (2 seconds) with the following playout features:

- Live, Delayed Live, VOD Playout
- Low Latency Live - 2 seconds "live edge" pullback globally
- Subtitles/captions,
- Multiple audio tracks,
- Live Multiview Switch for personalized views
- Forensic watermarking using NAGRA A/B
- DRM (Apple Fairplay, Google Widevine), and Clear Key
- Content Encryption (CBCS and CENC)
- Discontinuities for inserted content / play across content boundaries, e.g. ad insertion
- Dynamic Content Selector for personalized insertion of content
- SCTE cues read, stored, and transmitted - can be added as needed to the output manifest
- Automatic Content Verification through Fabric Hashes

Audio Details

The Fabric currently supports ingest of multi-channel surround audio with AAC or Dolby AC3 & E-AC3 source encoding and AAC playout. We are working with Dolby on adding support for AC4 and Dolby Atmos targeting release in Q4 2023.

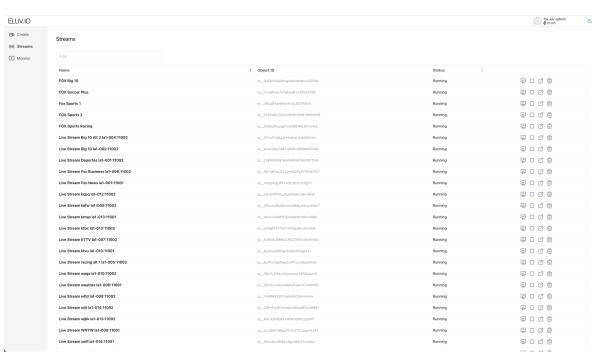
Captions/Subtitles Details

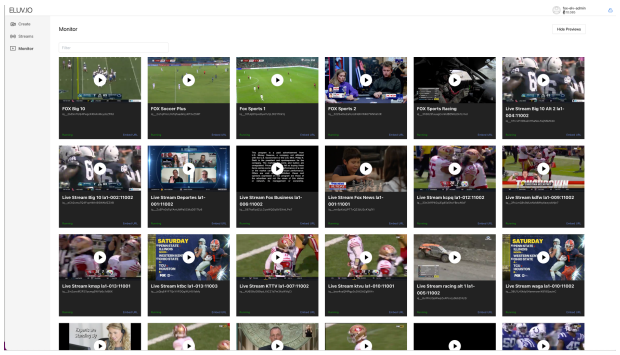
The Fabric ingest supports all major subtitling formats in its ingest tooling and automatically conforms/converts those to the WebVTT standard for Fabric storage and playout. The current ingest automation supports STL, ITU, WebVTT and can add other

formats such as EBU-TT/SMPTE-TT. This applies to VoD playout now and can be added to Live playout.

Stream Monitoring and Quality of Service

For actively streaming content the Fabric includes a live stream management console that displays the current status and health of all running streams and provides start/stop/reset control. It also provides a comprehensive live playout console for viewing and playing out any stream within a tenancy with filter options.

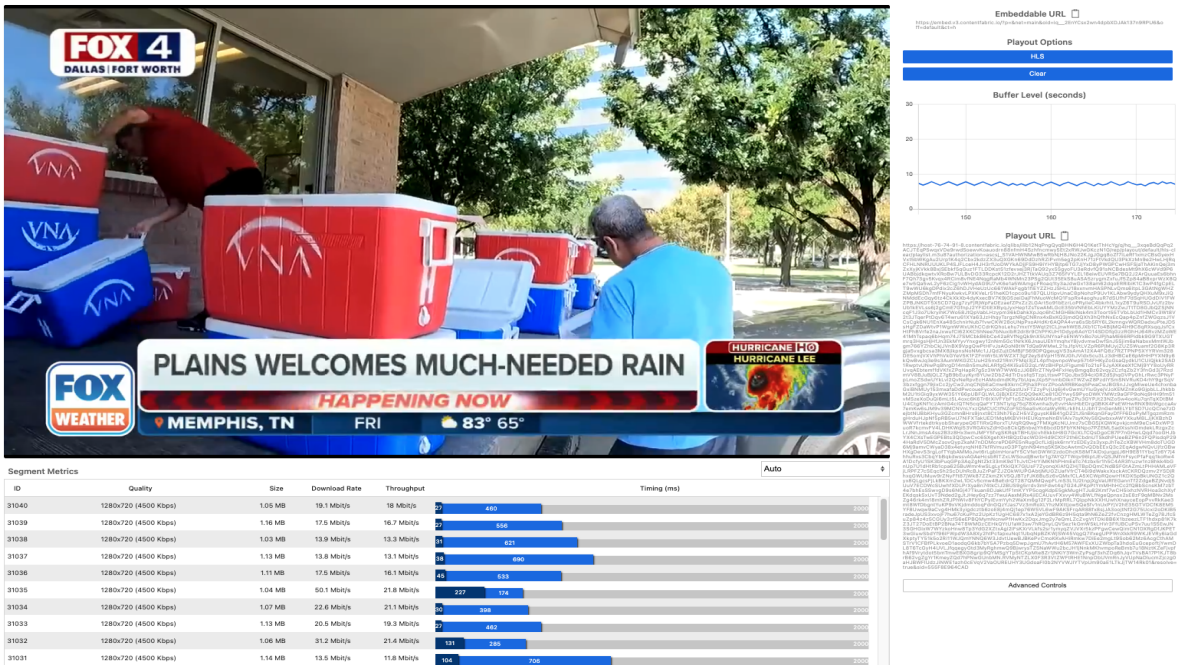




Client/Player QoS Reporting

The per-stream drilldown allows for interactively viewing the time-to-first byte, segment arrival time, and player buffer statistics for the viewing client:

This data can be easily reported by 3rd party developers using any player by following the open source sample “stream sample” that does this for the HLS.js player.



The **Eluvio Embed Player** built upon the open source HLS.js player also includes MUX analytics support for reporting comprehensive player side quality of service statistics such as playback failures, startup time, rebuffering and video quality.

Fabric Side QoS Reporting

The Fabric operational reporting continuously monitors the entire just-in-time transcoding pipeline on every node including incoming part arrival, transcoding throughput, and new segment output availability. This data is available through the Fabric APIs and will be exposed through the tenant reporting dashboard currently under development (ETA Q4 2023).

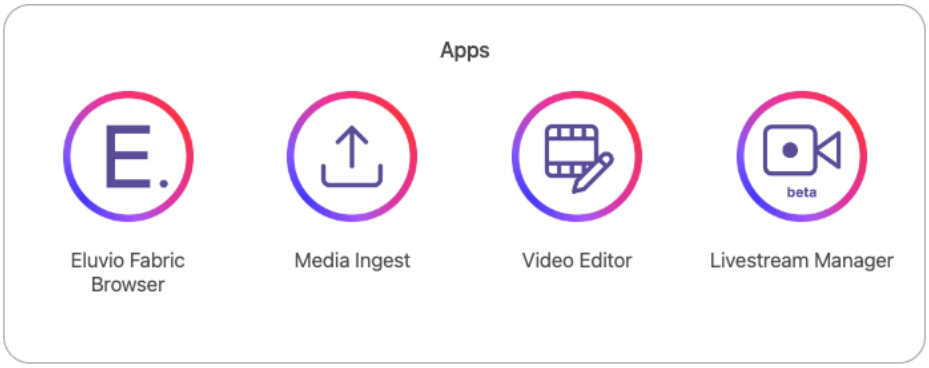
Media Ingest

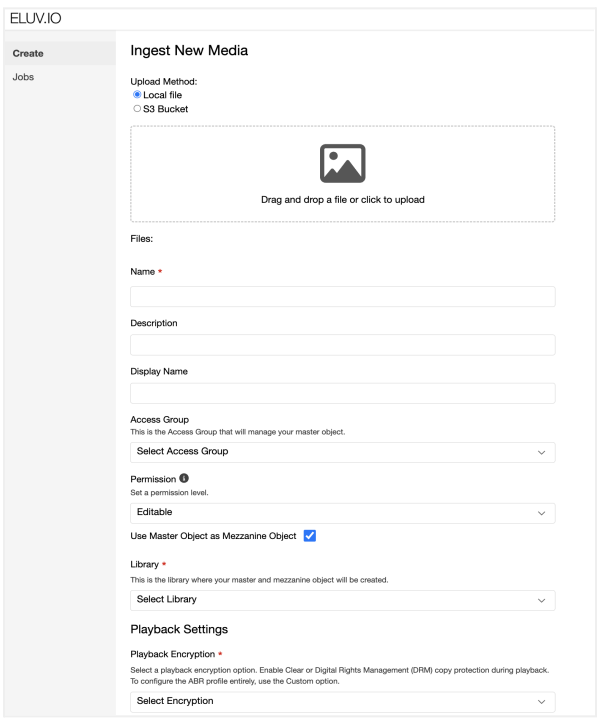
The Fabric provides automatic and component-based ingest of all master format file and live video content, automatically creating globally playable and serviceable content mezzanine objects on ingest. Ingest supports all professional master video formats (Apple ProRes, Avid DNxHD, IMF, JPEG-2000) and can be extended for custom codecs. A master object is also created storing references to the source (which can be remote, e.g. on cloud or in Fabric). The automation APIs support retrieval from Glacier as well as Amazon S3. The bit rate and resolution of the mezzanine content object is configurable and is unlimited for visually lossless options (e.g. 100 Mbps+).

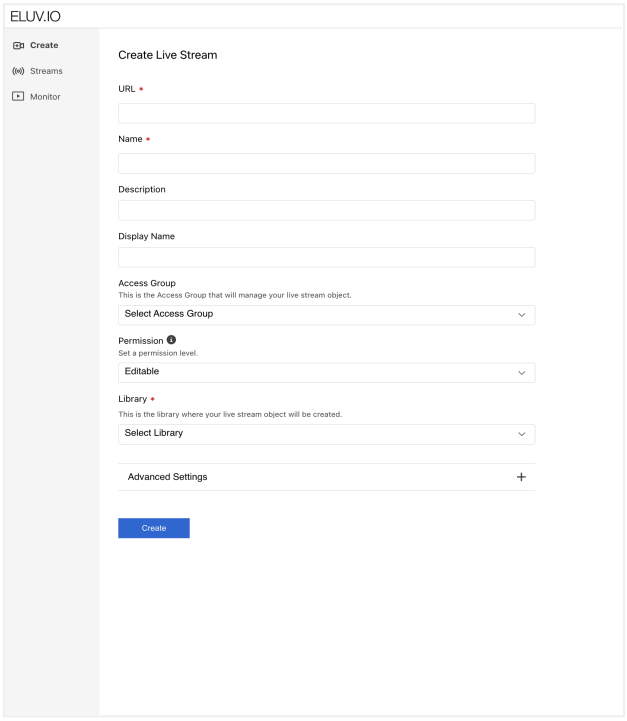
Live stream ingest supports automatic ingest of H.264/AVC, H.265/HEVC live streaming content packaged as MPEG-TS/UDP or RTMP/TCP, 30-50fps+, SD-4K UHD High Bandwidth.

The Fabric provides two browser apps for creating/ingesting master file and live streams that return playable, embeddable URLs and has comprehensive APIs and an automation engine for advanced ingest.

- The **Media Ingest App** provides drag-and-drop automatic ingest for any file based video and returns a playable URL.
- The **Livestream Manager App** automatically creates playable live streams and returns embeddable URLs for immediate sharing and playout globally.
- Comprehensive ingest APIs and CLI are available here: https://hub.doc.eluv.io/guides/ingestingmedia/workflow-examples/



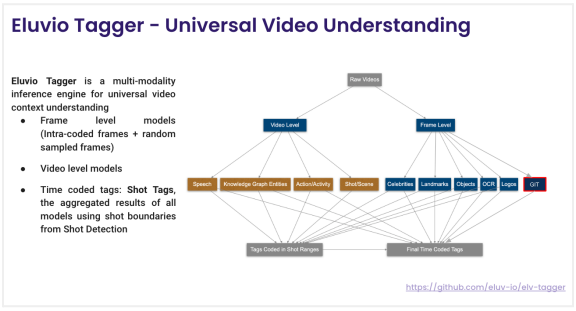


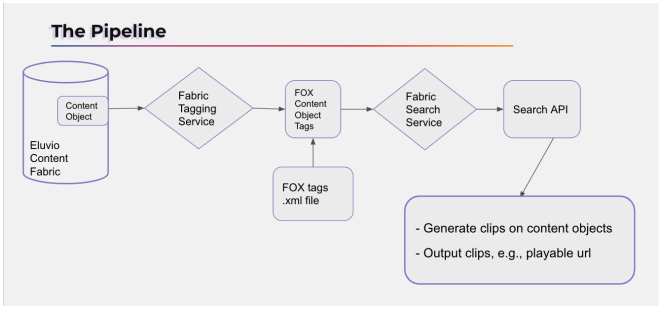


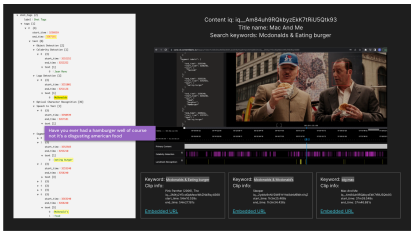
Automatic Speech to Text through AI/Deep Learning

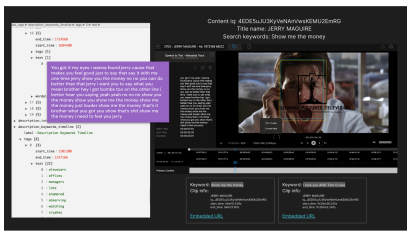
The Fabric includes an advanced and highly accurate automatic speech to text AI model in its AI/ML Tagger Services that can be run on any playable content objects in the Fabric. This model is currently in use for on demand content **but executes fast enough for live content as its inference runs about 3X as fast as real time for typical audio/video**.

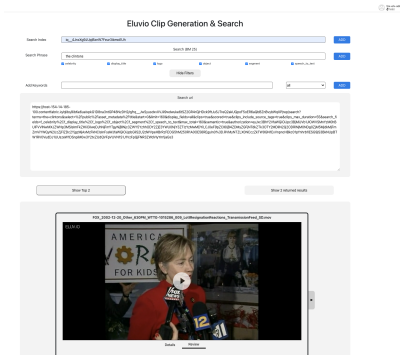
The resulting audio transcription is continuously stored as metadata on the content object with timecode as it is generated. This means that it could be used for automatic transcription on both live and on demand content. The captured metadata can be viewed in the Video Editor (for editorial/curation) and can be consumed/viewed/presented by front end applications such as a player or indexed for search (as in our Clip Search application). See below for results:











Reporting & Analytics

Audience Engagement Analytics

The Fabric provides end-to-end delivery of all audio/video and static content to the end user as a full 'utility' service and thus all viewing/playout/access is recorded continuously across the Fabric for all users. The data is also *tamper proof* and *verifiable* back to the *individual user* session because every viewer session is uniquely authorized and the Fabric operation is recorded by the Node providing that service against its internal blockchain in an authorized and tamper proof manner. All data is collected continuously and provided as an API (REST/JSON and CSV export) for queries by Fabric tenants. Additionally, aggregated daily and monthly reports are available in the Fabric Browser for download.

Fabric reporting includes the following information:

- By Content Object: Number of accesses/views and duration of playout (totals and by bitrate) for all playable content objects (titles, streams, interactive experiences). For downloadable types (e.g. files, images), bytes served per object. *Note that because the Fabric utility only serves what is consumed, the utility totals per object equal the consumption per object. This is different and much more efficient than legacy distribution (cloud-origin-cdn) where the bytes/hours generated may be much larger than consumption.*
- By Viewer Session: Individual user sessions including unique authorization address or wallet address, content object URL accessed, and user agent details including device platform/browser/client IP and client IP to geo resolution.

These core reports can be accessed as exportable CSV or via the tenant reporting API. A detailed spec of the reporting API is provided below.

Per Viewer Session Data

auth_identity: the address of the wallet of the user originating that playout session (if applicable).

auth_addr: the address of the private key to authenticate into the Eluvio fabric browser to originate that playout session (if applicable).

auth_email: the email address for the user originating that playout session (if provided in the authentication token).

client_ip: the IP address originating the playout session.

client_location: the global region from which the session was served.

client_city: the city from which the playout session originated (obtained by IP address lookup).

client_country: the country from which the playout session originated (obtained by IP address lookup).

client_region: the state, province or region from which the playout session originated (obtained by IP address lookup).

product: the application used to as a client for the playout session (extracted from the user agent).

platform: the hardware platform used as a client for the playout session (extracted from the user agent).

OS: the operating system used on the client for the playout session (extracted from the user agent).

referrer: the URL from which the playout session was originated.

id: the version hash of the content object played out.

ip_title_id: the unique ID (if provided in the public metadata) for the object played out in the session.

title: the title of the object being played if provided in the public metadata or the name of the object accessed if not.

offering: the offering being played out. Offerings can be used to differentiate between distributions for a given asset. Some offerings might have a slate, some have it cut off, some play with DRM, some without...

view_date: the UTC date at which the session was initiated (formatted YYYY-MM-DD).

view_start: the UTC time at which the session was initiated (formatted HH:MM:SS.mmmZ).

duration_secs: the duration for the content being watched.

watched_secs: the number of seconds of playout for that session. Note that this number can exceed the duration of the content if the session is looping or if parts of the content are rewinded back and replayed by the end user.

pct_watched: the percentage of the video segments that were actually played.

Per Content Object Fabric Utility ("Usage") Data

object_id: the unique internal ID for that object. That ID can be used to locate the object on the Eluvio content Fabric, using the fabric browser or through the command line.

info.library_id: the unique internal ID assigned to the library in which the content object is located. That ID can be used to locate the library and the objects it contains on the Eluvio content fabric.

info.ip_title_id: ip_title_id is a field in the public metadata of an object used to provide a functional, tenant-provided unique ID for a content object. That field is optional and not always populated.

info.object_name: the name provided at the creation of the content object on the fabric.

info.title: a functional name provided in the public metadata for the object. Typically used to store the content's actual title. That field is optional and not always populated.

info.title_type: a name given to the functional category of the object (example: NTF template, feature...). That field is optional and not always populated.

info.asset_type: used to differentiate primary objects (like a feature), from ancillary objects (like a trailer for a feature). That field is optional and not always populated.

storage: the amount of disk space occupied by the object. That space includes the metadata, attached files and all data stored in attached content parts.

info.active: usage status of the object. Possible values are:
active: indicates the object was played, attached parts or files were downloaded or that non-public fields in its metadata were accessed during the reporting period.
inactive: indicates no accesses whatsoever were made to that object during the reporting period.
poked: indicated that only the public metadata was accessed for that object during the reporting period.

usage.playout.audio_clear_sec: the aggregate number of seconds of audio of that content played without DRM across all audio requests during the reporting period.

usage.playout.audio_drm_sec: the aggregate number of seconds of audio of that content played with DRM across all audio requests during the reporting period.

usage.playout.video_hi_clear_sec: the aggregate number of seconds of video of that content played in 4K or equivalent resolution without DRM across all video requests during the reporting period.

usage.playout.video_hi_clear_sec: the aggregate number of seconds of video of that content played in 4K or equivalent resolution with DRM across all video requests during the reporting period.

usage.playout.video_md_clear_sec: the aggregate number of seconds of video of that content played in HD or equivalent resolution without DRM across all video requests during the reporting period.

usage.playout.video_md_drm_sec: the aggregate number of seconds of video of that content played in HD or equivalent resolution with DRM across all video requests during the reporting period.

usage.playout.video_lo_clear_sec: the aggregate number of seconds of video of that content played in SD or lower resolution without DRM across all video requests during the reporting period.

usage.playout.video_lo_drm_sec: the aggregate number of seconds of video of that content played in SD or lower resolution with DRM across all video requests during the reporting period.

info.master: empty header column used to separate production master related data for readability. The production master is the source media file provided to ingest the content in the Eluvio content fabric. All the following master fields will also be blank unless the object is one of the production masters used during the content ingest process.

info.master.codec.audio: the name of the codec used for the encoding of the source production master audio.

info.master.codec.video: the name of the codec used for the encoding of the source production master video.

info.master.duration: the duration of the production master source content (in seconds).

info.master.resolution: the video resolution of the production master content.

info.mezzanine: empty header column used to separate mezzanine related data for readability. The mezzanine is the playable content object. All the following mezzanine fields will also be blank unless the object is a playable media object.

info.mezzanine.codec.audio: the name of the audio codec used for the media streamed out of the fabric (typically AAC).

info.mezzanine.codec.video: the name of the video codec used for the media streamed out of the fabric (typically H264).

info.mezzanine.duration: the duration in seconds of the playable media object.

info.mezzanine.playout_formats.<x>: the supported playout formats, displayed over 1 to 6 columns. Supported formats are a subset of dash-clear, hls-clear, dash-widevine, hls-fairplay, hls-aes128 and hsl-sample-drm.

usage.data_download.bytes: the aggregated amount of downloaded data in bytes for that object.

usage.data_download.occurrences: the number of requests received for data download for that object. The cumulated size of the data served in response to those requests is what is measured in the preceding column (**usage.data_download.bytes**).

usage.error.bytes: the aggregated amount of data served as a result of invalid requests. This data is for error messages.

usage.error.occurrences: the number of invalid requests received for that object. The cumulated size of the error messages served in response to those requests is what is measured in the preceding column (**usage.error.bytes**).

usage.ingest.audio_<codec>: the number of seconds of audio content ingested that were encoded with a certain codec. Note that the amount of audio content and video content for a given data source often do not match as multiple alternate audio can be provided.

usage.ingest.<codec>:.density_large_sec: the number of seconds of 4K or equivalent video content ingested that were encoded with a certain codec.

usage.ingest.<codec>:.density_medium_sec: the number of seconds of HD or equivalent video content ingested that were encoded with a certain codec.

usage.ingest.<codec>:.density_small_sec: the number of seconds of SD or lower video content ingested that were encoded with a certain codec.

usage.ingest.occurrences: the number of transcode requests processed to prepare the mezzanine content of that object. Typically one per video or audio stream.

usage.media.bytes: the number of bytes exchanged with the server when making calls to control the transcoding process.

usage.media.occurrences: the number of calls made to the function used to control the transcoding process.

usage.playout.audio_total_hrs: the total audio playout duration (DRM and clear combined) served for that content object in hours.

usage.playout.bytes: the total number of bytes served for the playout of that content object (audio and video combined).

usage.playout.occurrences: the total number of playout requests served. Note that a request is made for each segment of a video or audio (segments are approximately 2 seconds long). Requests also include manifests describing which segments to download.

usage.playout.video_total_hrs: the total video playout duration (DRM and clear combined) served for that content object in hours.

usage.public_download.bytes: the total number of bytes of public metadata served for this content object.

usage.public_download.occurrences: the total number of public metadata download request received for this object. The sum of the size of the data served in response to those requests is represented in the column **usage.public_download.bytes**.

usage.updated: the number of times the content object was updated during the reporting period.

Content Monetization Analytics

The Fabric provides many content monetization features: direct content sell-through, windowing/authorization, and associated redeemable offers and advertisements through APIs and its turnkey marketplaces and Media Wallet. All of the monetization reporting is included in the tenant reporting API and the elv-live CLI with downloadable reports of the following data:

- Per transaction reports of SKUs sold, associated content objects, price, quantity and purchasing address (wallet) by payment type. Payment types include Stripe, EBANX, Coinbase, Direct Crypto, and ELVD Wallet Balance.
- All Content SKUs per Marketplace Object and the associated content object details including all media, offers, and metadata.
- Owner wallet addresses and associated email addresses (if consented by user) for all authorizing blockchain tokens.
- ELVD (ERC20) wallet balances per wallet including buyers, sellers, and stakeholders such as 3rd party rights holders and advertisers.
- Secondary sales and 3rd party royalties per transaction including buyer and seller addresses, sold listings, price, quantity, creator royalty and purchasing method.
- Wallet addresses that have redeemed claimable offers.

Additionally, Eluvio is creating a comprehensive Analytics Dashboard for primary and Advanced (multi-variable) analysis of purchasing and engagement (in preview now, GA planned for Q4 2023).