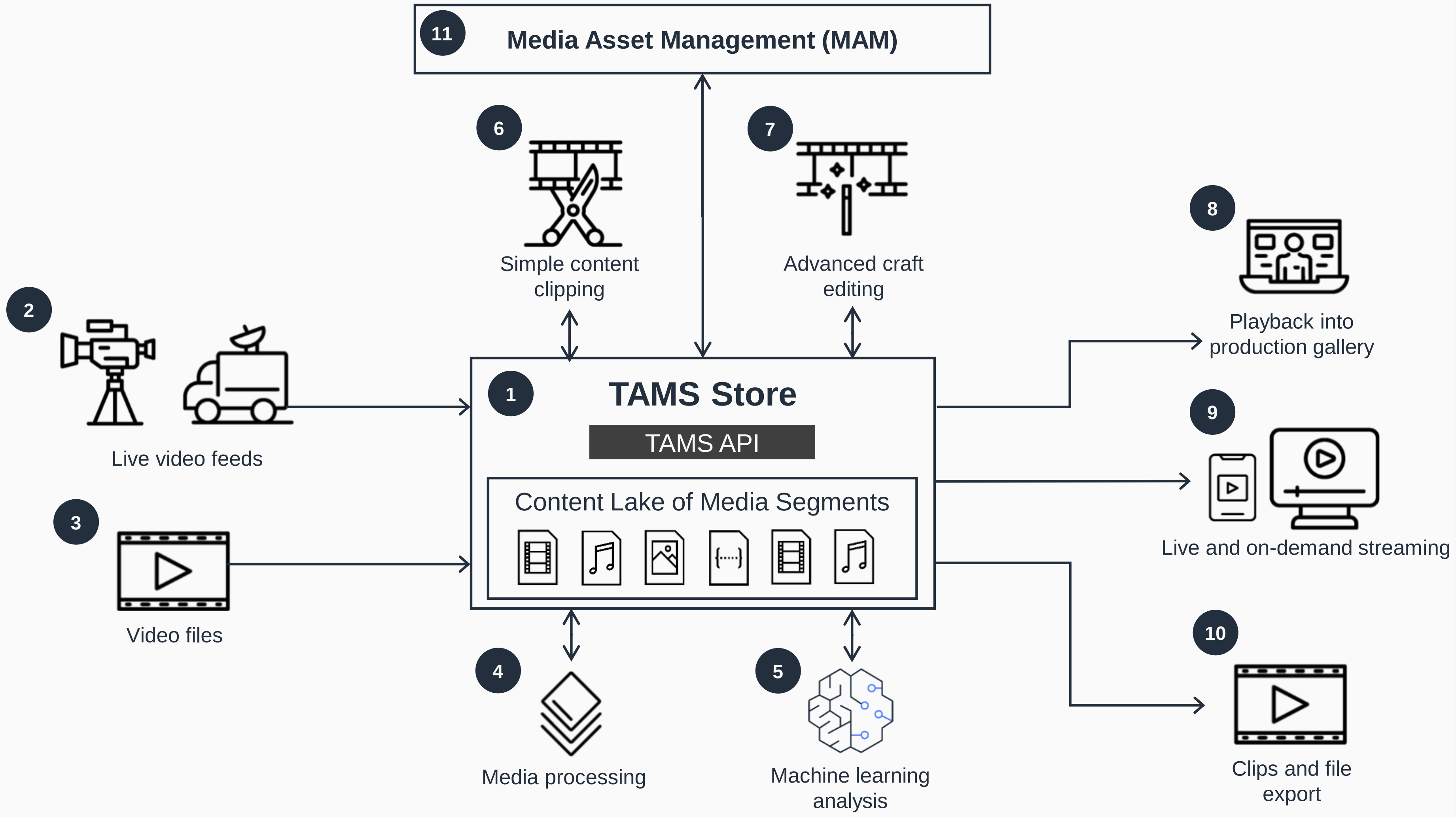
Guidance for Cloud-Native Fast-Turnaround Media Workflows on AWS

TAMS concept overview

This architecture diagram shows the concept about how a Time Addressable Media Store (TAMS) sits at the core of a fast-turnaround workflow for processing live or near-live video and audio content.



11 Media Asset Management (MAM)
6 Simple content clipping
7 Advanced craft editing
8 Playback into production gallery
2 Live video feeds
1 TAMS Store
TAMS API
Content Lake of Media Segments
9 Live and on-demand streaming
3 Video files
4 Media processing
5 Machine learning analysis
10 Clips and file export

1. All media is stored within a Time Addressable Media Store (TAMS). This holds chunked media on object storage with an API to provide the link between content and the media essence.
2. Live video feeds are uploaded as small chunks of media and registered with the TAMS to provide the effect of growing content.
3. File-based content can be uploaded natively or chunked prior to import as required.
4. Content can be processed in near real-time, generating additional versions, such as proxies, triggered through notifications from the storage system.
5. Content analysis can occur asynchronously and in near real-time, enabling rapid access to the outputs of machine learning and artificial intelligence (AI/ML) models. Examples include live subtitling, highlights generation, and content logging.
6. Simple clip-based editing can be performed, and the resulting edits can be published back to the TAMS, referencing the original content.
7. Craft editing can access content from the TAMS and publish back only new segments.
8. Content from the store can be played back as a real-time video stream into production galleries of linear channel playout facilities.
9. TAMS API can be easily converted into HTTP Live Streaming (HLS) manifests to enable live or on-demand content streaming from the TAMS.
10. Clips and files can be exported from the store for alternative purposes, such as the rapid distribution of content onto social media platforms.
11. The Media Asset Management (MAM) system maintains references to the content stored within the TAMS, along with the associated rich editorial and time-based metadata.

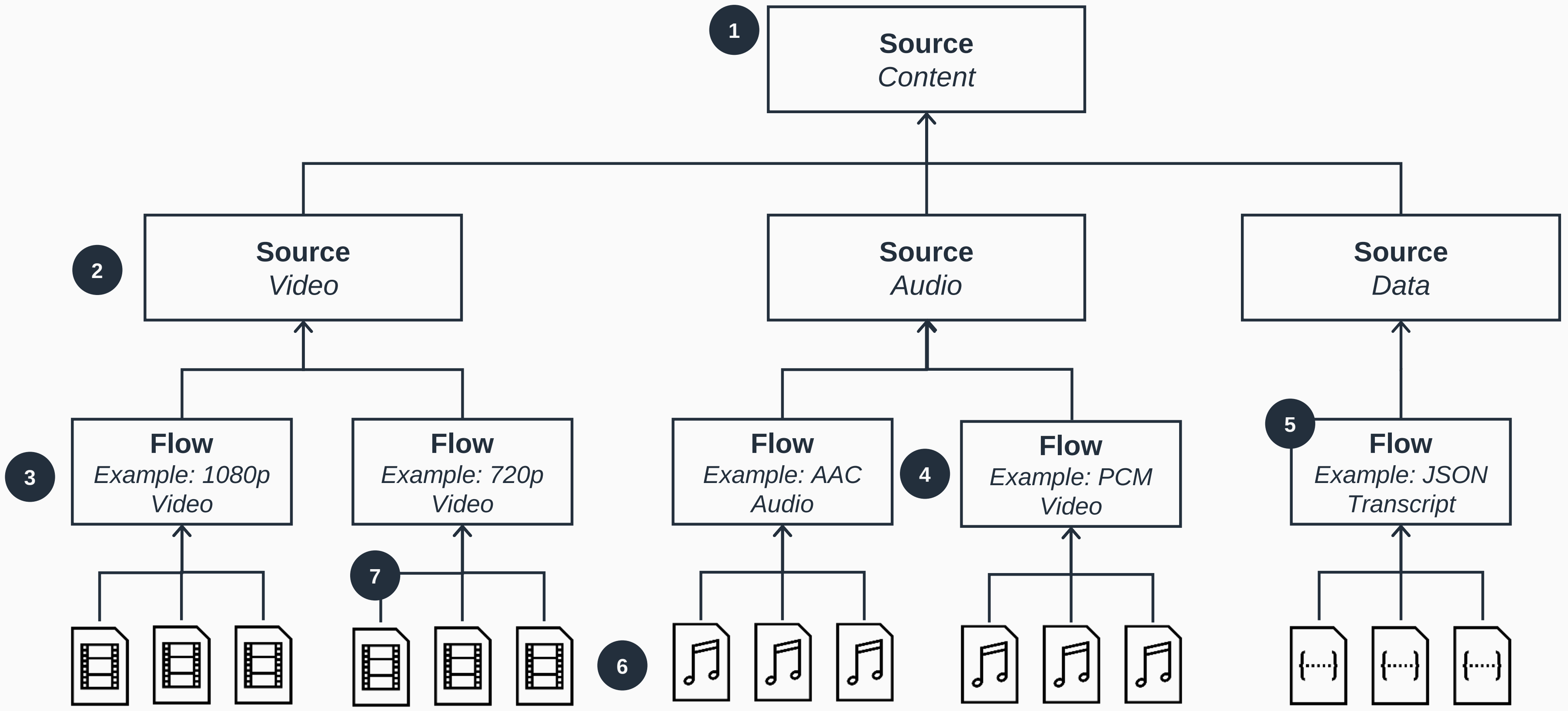
Reviewed for technical accuracy March 6, 2025

AWS Reference Architecture

Guidance for Cloud-Native Fast-Turnaround Media Workflows on AWS

TAMS data structure

This architecture diagram shows the high-level data structure represented in the TAMS API specification. This diagram establishes the connection between the content that a user would be aware of and the actual media essence, which is stored in multiple formats and segments on the object storage system.



1. In the TAMS data structure, the parent source is equal to the actual content that a user interacts with. This parent source could be an editorial version of the content or a clip.
2. The secondary level source within the data structure allows for the aggregation of the various media types, such as video, audio, or data, into a cohesive collection.
3. The "flow" represents the technical manifestation of the content. This construct contains all the technical metadata necessary to describe the underlying media segments, such as bitrate, resolution, and frame rate.
4. Multiple flows can exist for a single piece of content, enabling the representation of different formats, such as HD and proxy, to coexist.
5. Flow types include video, audio, and data, allowing the different content types to be referenced in the store.
6. Segments are held on object storage and referenced in the TAMS API. The only interaction between the store and the segments occurs during deletion management.
7. The segments are linked to a flow and exist within the context of that flow's virtual timeline. A time range format, expressed as Epoch time plus nanoseconds, is used to represent the position of each segment along the timeline.

Notes

A segment can be referenced in one or more flows, allowing the reuse of segments between content without duplication at the storage layer.

The TAMS maintains only the metadata required for the storage and referencing of the media content. The rich metadata should be managed within separate systems, such as the Media Asset Management system.

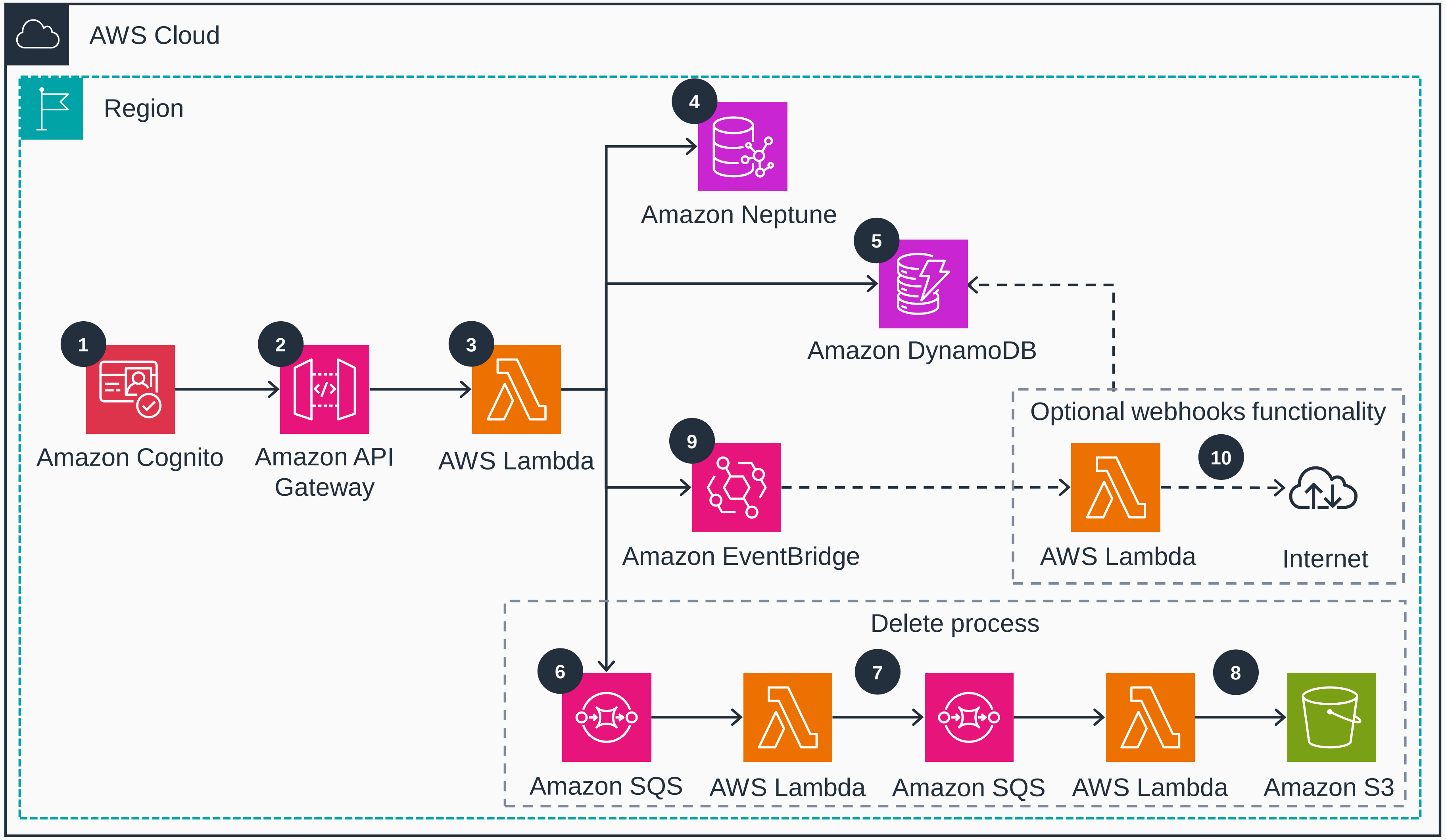
Reviewed for technical accuracy March 6, 2025

AWS Reference Architecture

Guidance for Cloud-Native Fast-Turnaround Media Workflows on AWS

AWS open source TAMS API

This architecture diagram shows the components and data flows within the AWS open source implementation of the TAMS API.



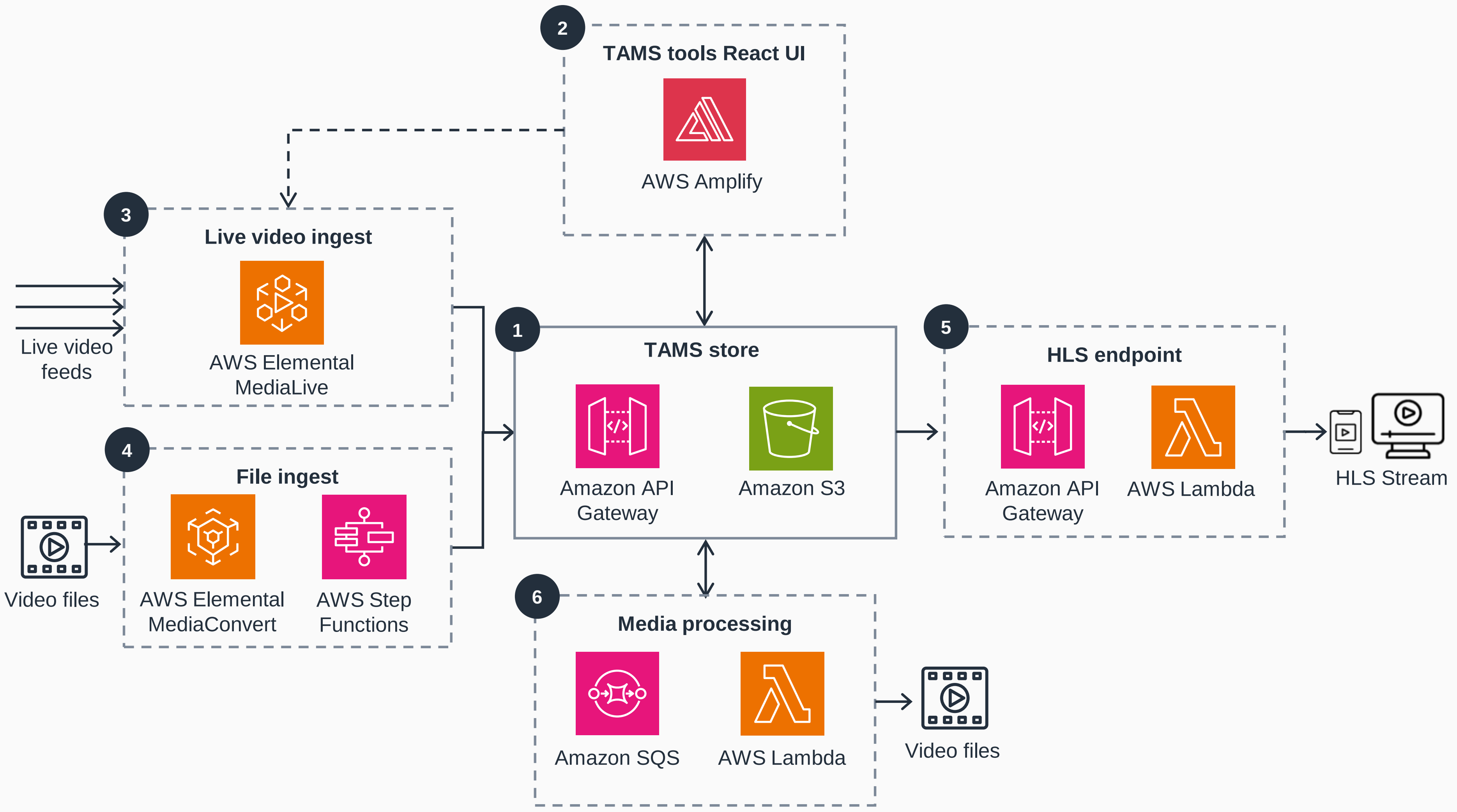
1. **Amazon Cognito** provides user and system-to-system authentication.
2. The API is presented through **Amazon API Gateway**, including the validation of requests using the OpenAPI specification.
3. **AWS Lambda** functions process the API requests. Separate functions exist for the services, sources, flows, segments, and delete request endpoints.
4. Source and flow metadata is stored in an **Amazon Neptune** graph database.
5. Segment metadata is stored in **Amazon DynamoDB** for speed of retrieval.
6. Delete requests are forwarded to **Amazon Simple Queue Service** (Amazon SQS) for asynchronous deletion.
7. A **Lambda** function is responsible for processing delete operations by forwarding the requests to a secondary **Amazon SQS** queue, which then handles the deletion of the corresponding **Amazon Simple Storage Service** (Amazon S3) objects.
8. After the required wait period, a **Lambda** function evaluates delete requests and removes only unused objects from **Amazon S3**.
9. Events from core API functions are sent to **Amazon EventBridge** for subsequent reuse by other systems.
10. An optional **Lambda** function can process webhook requests to external systems according to the specification.

AWS Reference Architecture

Guidance for Cloud-Native Fast-Turnaround Media Workflows on AWS

AWS open source TAMS tools

This architecture diagram demonstrates how the multiple components of the AWS TAMS Tools repository can be used alongside the core AWS open source TAMS implementation.



1. The TAMS Store capability is provided by the AWS open source implementation.
2. A React-based user interface application, deployed through **AWS Amplify**, enables users to navigate the store, view video content through the HLS endpoint, and control the live and file-based ingestion processes.
3. Live video ingestion is facilitated using **AWS Elemental MediaLive**, which creates segments on **Amazon S3** that are subsequently uploaded to the TAMS.
4. File-based import is enabled using **AWS Elemental MediaConvert**, orchestrated by **AWS Step Functions**, to chunk up the media and upload into the TAMS.
5. An HLS endpoint is provided to convert the TAMS native API calls into a set of HLS manifests to allow content to be played back within a web-based HLS player.
6. The media processing workflow uses event notifications from the TAMS to trigger additional post-processing of the ingested content. This includes the extraction of images and the creation of proxy versions using **Lambda**, as well as the export of concatenated files for integration with other systems.

Reviewed for technical accuracy March 6, 2025

AWS Reference Architecture