Media Processing Lambda – Functional Guide

Hosting Infrastructure · https://www.mooplugins.com/docs/video-lesson-activity/hosting-infrastructure/media-processing-lambda-functional-guide/

Purpose of this Lambda

The Media Processing Lambda is responsible for **orchestrating video processing workflows** for Video Lesson Activity.

It reacts to media uploads, submits transcoding jobs, tracks job lifecycle events, records processing state, and exposes reliable status information for external systems such as Moodle.

This Lambda does **not** handle subtitles directly. Subtitle processing is handled by a separate, decoupled Lambda.

Role in the overall architecture

The Media Processing Lambda acts as the **control layer** between raw media uploads and processed video outputs.

Its responsibilities include:

- Detecting new media uploads
- Submitting MediaConvert jobs
- Tracking job state changes
- Persisting job metadata and status
- Notifying downstream workflows when required

It ensures that Moodle can determine **when a video is ready**, **what outputs exist**, and **whether processing succeeded or failed**.

Event sources

The Lambda responds to two categories of events.

Media upload events

Triggered when a new media file is uploaded to the input storage location.

These events are used to:

- Read file-level metadata
- Determine processing requirements
- Initiate a transcoding job

Metadata may include:

- Site or tenant context
- Requested output formats
- Subtitle preferences
- Downloadable video options

MediaConvert job lifecycle events

Triggered when the transcoding service updates the job state.

These events are used to:

- Track processing progress
- Detect completion, failure, or cancellation
- Perform state persistence and downstream notifications

Lifecycle handling is idempotent and resilient to duplicate events.

Transcoding job creation

When a new media upload is detected:

- File metadata is extracted and validated
- A MediaConvert job is created with:

    - Adaptive streaming outputs
    - Optional downloadable formats
    - Thumbnail and preview generation
- Contextual metadata is embedded into the job request

Embedding context allows the Lambda to reconstruct processing intent during later lifecycle events without relying on external lookups.

Output organization

Generated outputs are written to storage using a **predictable and structured layout**.

This may include:

- Streaming assets for playback
- Downloadable video files (when enabled)
- Thumbnails and preview images
- Lightweight metadata describing the processing context

Consistent organization ensures that:

- Outputs are discoverable without scanning storage
- External systems can fetch assets deterministically
- Downstream workflows remain decoupled

Job lifecycle handling

When MediaConvert emits lifecycle events:

- Job metadata and current status are extracted
- Output locations are validated
- Ownership and context are confirmed
- Processing state is persisted

The Lambda records:

- Current job status

- Final outcome (success or failure)
- Associated output locations

This allows Moodle to reliably:

- Display processing progress
- Detect readiness for playback
- Handle error or retry scenarios

Persistent job state tracking

Processing state is stored in a persistent data store designed for:

- Lookup by job identifier
- File-based or content-based queries
- Historical state inspection

This data enables external systems to:

- Poll for completion status
- Avoid querying AWS services directly
- Recover safely from retries or partial failures

State persistence is lightweight and optimized for frequent reads.

Optional subtitle workflow triggering

Subtitle generation is **not part of the core transcoding path**.

If subtitle processing is enabled:

- A notification is emitted with the required context
- Subtitle processing runs in a separate Lambda
- Resulting subtitle files are stored alongside media outputs

This separation ensures that:

- Video availability is not blocked by transcription delays

- Subtitle failures do not affect video playback
- Each workflow can scale independently

Storage usage accounting

To support reporting and quota management:

- Output sizes are calculated per media item
- Folder-level or domain-level usage summaries are generated
- Usage data is stored as metadata

This avoids expensive storage scans while still providing accurate usage information.

Security and access model

The Media Processing Lambda operates under a scoped IAM role with permissions limited to:

- Required storage locations
- MediaConvert job creation and monitoring
- Event and notification handling
- Persistent state updates

All access is:

- Role-based
- Auditable
- Restricted to required resources

No shared credentials are used.

Design characteristics

This Lambda is designed to be:

- Event-driven
- Horizontally scalable
- Resilient to retries and partial failures

- Idempotent across repeated events
- Decoupled from subtitles and delivery concerns

Each responsibility is isolated to reduce operational risk.

Implementation summary

At a functional level, the Media Processing Lambda:

- Responds to media uploads and job lifecycle events
- Creates transcoding jobs with embedded context
- Tracks processing state reliably
- Persists status for external consumption
- Triggers optional downstream workflows without coupling
- Supports scalable and secure video processing for Moodle