Video platform interface

We call the video platform several times during the publication process to:

- provision a slot for the new video
- upload the actual binary
- know when the new video is actually available on the platform, or collect any publication error
- query different metadata (more below)

Additionally, a call to a delete endpoint is ready to use but not wired to any workflow yet.

Actual calls

1. Creation and initial upload

This step usually executes in sequence the provisioning of a slot for the future upload, then uploads the binary content itself.

Slot provisioning

We call the platform to inform it that a new video is coming next. The platform replies to acknowledge the request and send information to continue the process.

Request parameters:

- Title
- Description

Response parameters:

- Identifier of the new slot
- Upload URL if applicable (on some platforms, it could be inferred from the identifier and a known URL pattern)

Content upload

Once the platform gives a new identifier, hence acknowledging it is ready to receive a new video content, the upload begins.

Request parameters:

- Identifier (somehow - probably part of the called URL)
- Binary data, of MIME type video/mp4

Response parameters:

- None if OK, could be an error response

2. Monitoring of the processing on the platform

As platforms often re-encode the binary in different formats for convenience, player compatibility or bandwidth optimization as well as preparing thumbnail(s), we need to monitor the progression of these long-running processes to know when the video actually becomes available.

Request parameters:

- Identifier (somehow - probably part of the called URL)

Response parameters:

- Current status
- Error message or code if something went wrong

The status could at least be one of:

- Pending
- Ready
- Error

Alternatively, the status may indicate a more precise unavailability reason and void the need for a separate error message.

The monitoring endpoint will be repeatedly called with an exponential back-off timing strategy until reaching a final status (= not pending).

3. Metadata collection

Once the video is available on the platform, we will collect some metadata before closing the publication procedure on our side.
Here represented as 2 separate calls because we will use it this way to maximize compatibility, the data may be provided by a single endpoint.

Video information

Request parameters:

- Identifier (somehow - probably part of the called URL)

Response parameters:

- video launch URL (mp4 or m3u8 playlist, suitable for HLS)
- video width
- video height
- video duration

Thumbnail information

Request parameters:

- Identifier (somehow - probably part of the called URL)

Response parameters:

- thumbnail image URL (jpeg or PNG)
- image width
- image height

4. Removal

As said before, we would like to get basic control on the video lifecycle from our back office in the future. This deletion endpoint is also part of the requirements for testing purposes.

Request parameters:
- Identifier (somehow - probably part of the called URL)

Response parameters:
- None if OK, could be an error response

Security

The API endpoints should be protected by a standard method. We expect to provide an **Authorization** header containing a token.
The token generation may either be using credentials on a dedicated endpoint (preferably of OAuth2 standard) or a permanent pre-shared secret, like an API key.