

Data Flow Diagram

AWS S3 to Labellerr to Annotator

How your data travels from your Amazon S3 bucket, through the Labellerr annotation platform, to an annotator's browser — and back on export.

Scope	Datasets connected via an AWS S3 connection
Processing region	us-central1
Platform egress address	35.224.11.247
Date	14 August 2026

Data Flow Diagram: AWS S3 → Labellerr → Annotator

Scope: Datasets connected to Labellerr through an AWS S3 connection, covering import, delivery to annotators, model-assisted labelling, optional search indexing, and export write-back.

1. Summary

What Labellerr reads from your bucket depends on the data type. For images, audio and text, Labellerr reads metadata only. Only video is read in full, and the frames it produces are written back into your own bucket.

The annotator's browser reads file content directly from your bucket. Labellerr issues a short-lived, cryptographically signed URL, and the browser uses it to fetch the object from AWS. Labellerr does not proxy, relay or copy that content. Two different network identities therefore appear in your S3 access logs.

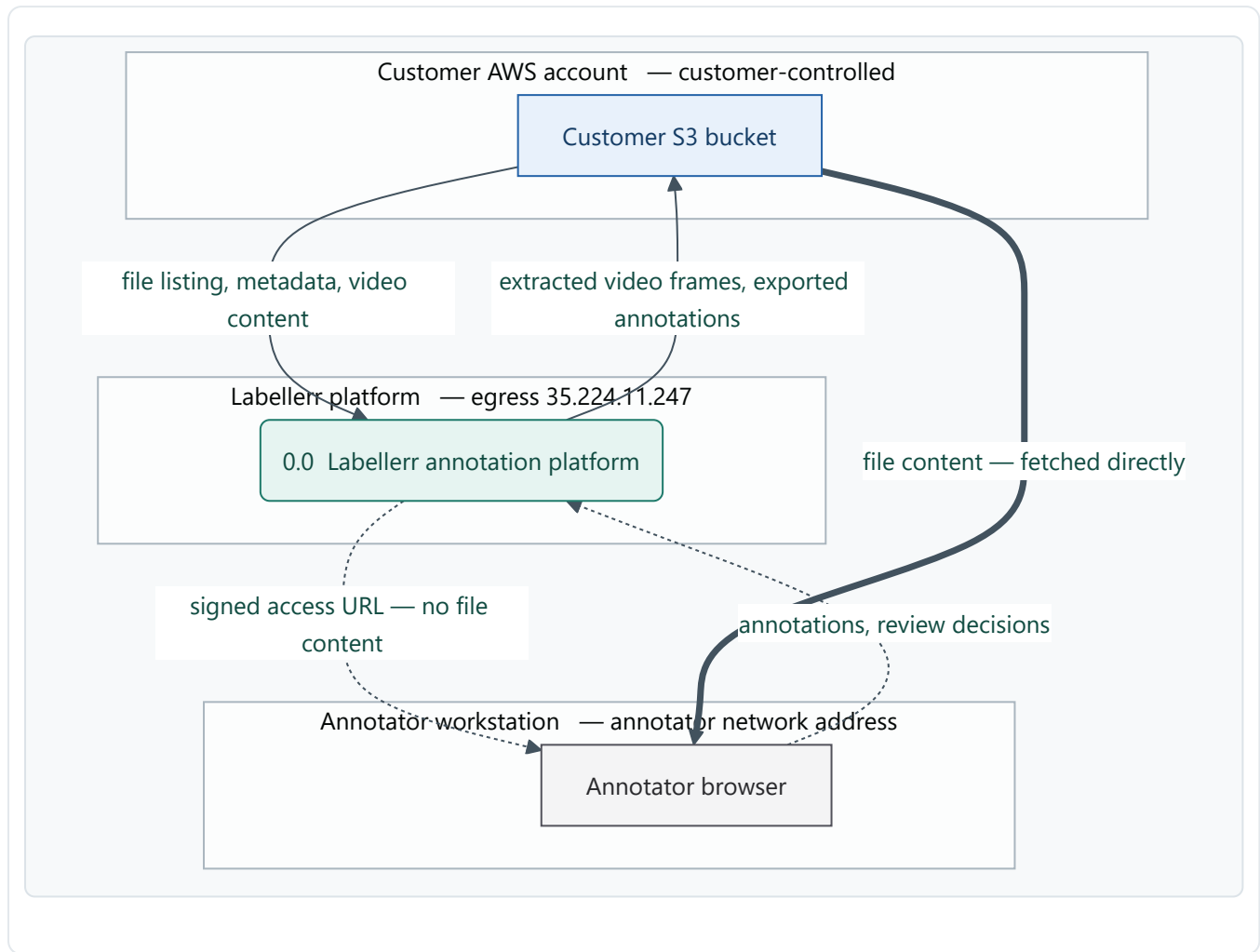
The processing tier holds nothing. All processing runs on ephemeral serverless instances whose working storage is memory-backed and destroyed when the instance is torn down.

2. System context

Property	Value
Labellerr processing region	us-central1 (Google Cloud)
Labellerr outbound network identity	35.224.11.247 (static egress address)
Annotator network identity	The annotator's own corporate or home network address
Transport	TLS 1.2+ on every flow shown
Authentication to your bucket	An IAM user in your AWS account, created and controlled by you

Every Labellerr service and job egresses through a single fixed network address. Any request Labellerr makes to your bucket originates from 35.224.11.247 and can be allow-listed or audited on that basis.

3. Level 0 — Context

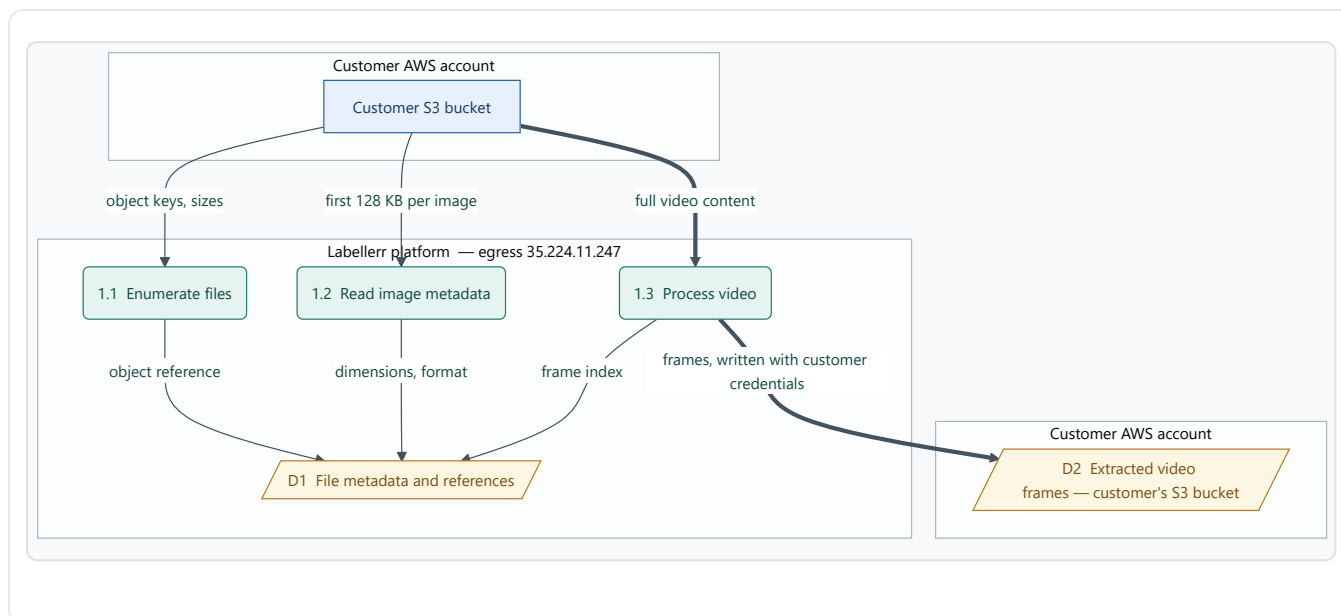


Legend. Thick solid arrow = file content. Thin solid arrow = data Labellerr reads or writes. Dashed arrow = control data only, carrying no file content.

File content does not travel through Labellerr on its way to the annotator.

4. Level 1 — Data import

What happens when a dataset is created against an S3 connection.



What is read, by data type

Data type	Operations against your bucket	Does file content leave your bucket?
Image	<code>ListBucket</code> , then <code>GetObject</code> with <code>Range: bytes=0-131071</code>	No — only the first 128 KB is read, solely to determine dimensions and format
Audio	<code>ListBucket</code> only	No — no object read is performed
Text / documents	<code>ListBucket</code> only	No — no object read is performed
Video	<code>ListBucket</code> , <code>GetObject</code> (full), <code>PutObject</code> (frames)	Read into instance memory for frame extraction; frames are written back to your own bucket

D1 stores a **reference** to each object — its `s3://bucket/key` path — together with size, type and derived metadata. It does not store the object.

Video processing closes inside your account

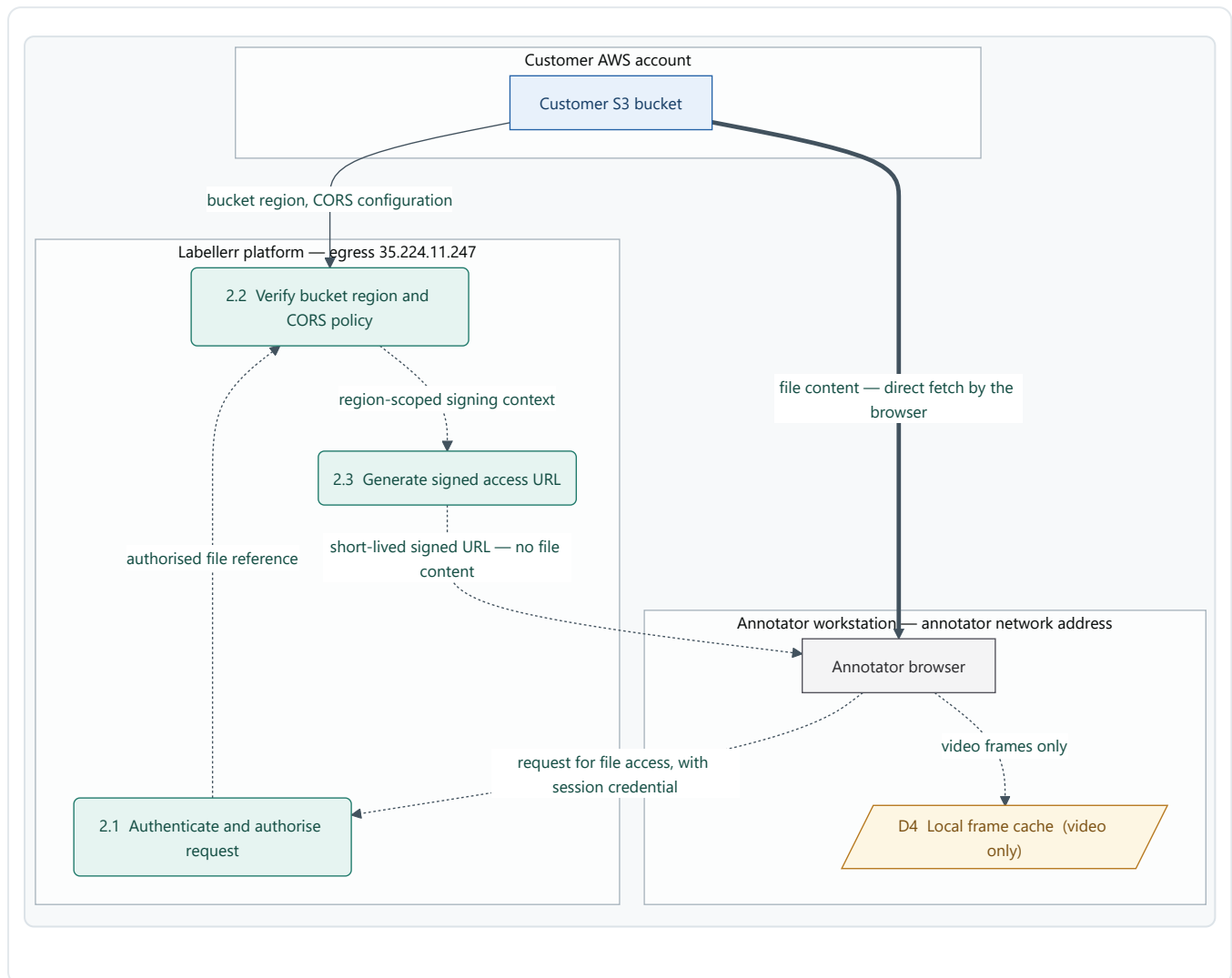
Video is the only data type whose content is read in full, and the loop completes within your own AWS account:

1. Labellerr reads the video from your bucket into the processing instance's memory.
2. Frames are extracted in memory.
3. Frames are written **back into your own bucket**, authenticated with your own credentials — the same IAM user you created for the connection.
4. Temporary working files are deleted, and the instance is torn down and its memory released.

Labellerr retains no copy of the video or its frames. The frames are your objects, in your bucket, under your own lifecycle and retention policy.

5. Level 2 — Serving file content to the annotator

A decomposition of `2.0 File access`.



The two planes

Control plane — originates from `35.224.11.247`. Labellerr confirms the requesting user is authorised, reads your bucket's region and CORS configuration, then generates a signed URL. URL generation is a **local cryptographic operation**: the signature is computed from your credentials without contacting AWS. The only requests Labellerr makes to your bucket on this path are `GetBucketLocation` and `GetBucketCors`.

Data plane — originates from the annotator's network address. The browser receives the signed URL and fetches the object from AWS itself. Labellerr is not in this path. The endpoints that serve the URL return JSON only and carry no file content, and the Labellerr web application is a static front-end bundle that never handles file content.

Three consequences:

1. Object reads in your S3 access logs will show **annotator network addresses**, not Labellerr's.
2. Your bucket must carry a CORS rule permitting `GET / HEAD` from `https://*.labellerr.com`. Without it, browsers block the fetch. Labellerr verifies this rule is present before issuing URLs.

3. Access URLs are **short-lived bearer credentials**. They are scoped to a single object and to read-only access, and are issued only after the requesting user's session is authorised.

Client-side caching — video only

To keep video scrubbing responsive, the annotation tool stores decoded video frames in the browser's local IndexedDB store, budgeted at up to 1 GB per video, evicted on a least-recently-used basis. This persists across page reloads.

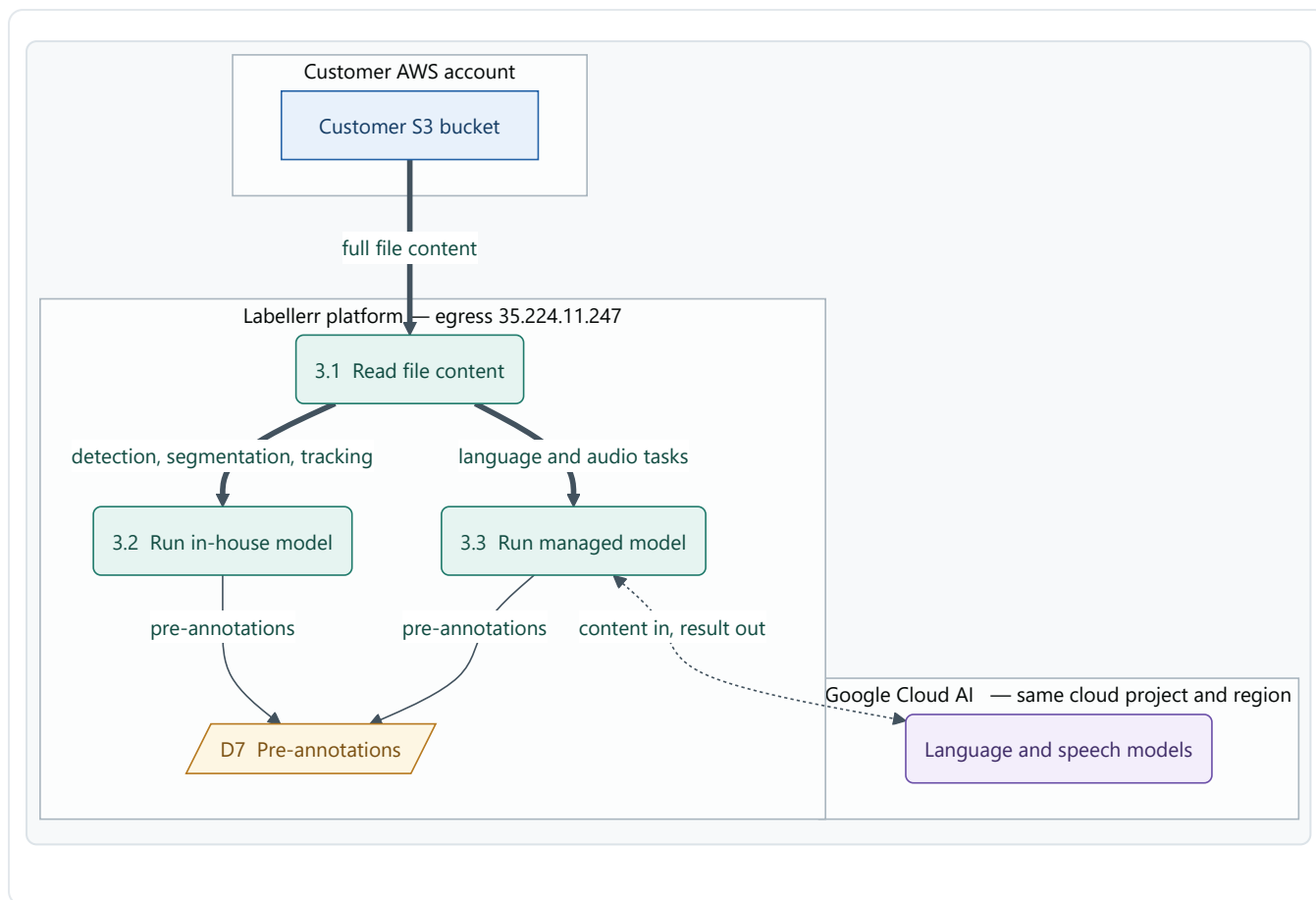
This applies to video only. Images are held in page memory for the lifetime of the browser tab and are never written to browser storage. Audio and text are not cached client-side at all. There is no service worker and no use of the browser Cache API for file content.

6. AI processing of your content

6.1 Model-assisted labelling

Most model-assisted labelling runs **on Labellerr's own compute**. Object detection, segmentation and video tracking models are loaded and executed inside the Labellerr platform, and your content is not sent to any external service for these operations.

Certain language-model features — auto-labelling driven by a large language model, and audio transcription — are served by managed Google Cloud models. In those cases your file content and a prompt are sent to the model.

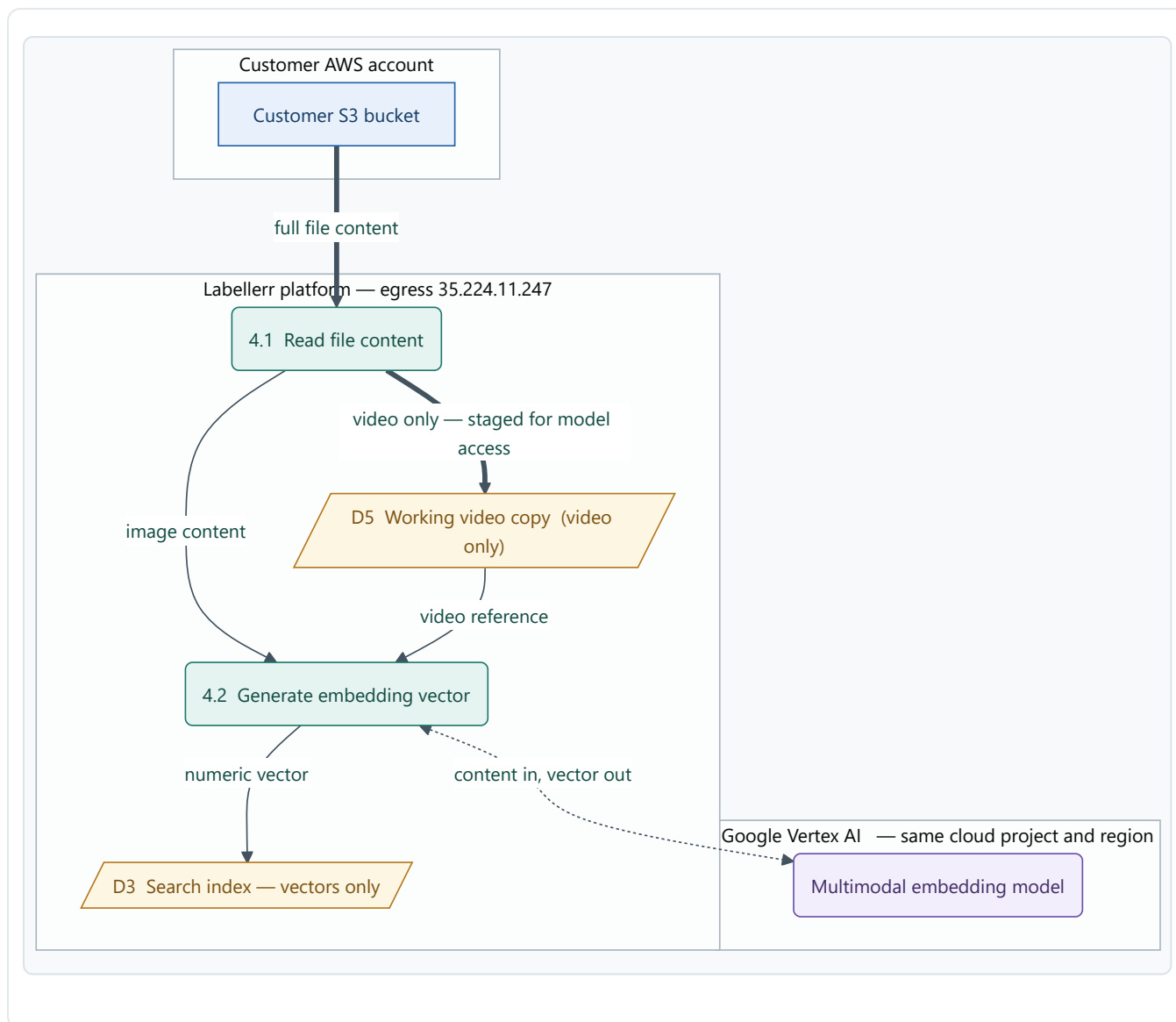


The managed models run inside Labellerr's own Google Cloud project, in the same `us-central1` region, as managed Google Cloud services. Your content is not sent to any external third-party AI provider.

On both paths, content is processed in memory and is not retained by the processing tier. Only the resulting pre-annotations are stored, and you can review, edit or discard them like any other annotation.

6.2 Optional multimodal search indexing

This process runs only when you explicitly enable multimodal indexing for a dataset at import time. It is off by default and is not applied to workspaces or datasets that have not opted in. If it is not enabled, nothing in this section occurs.

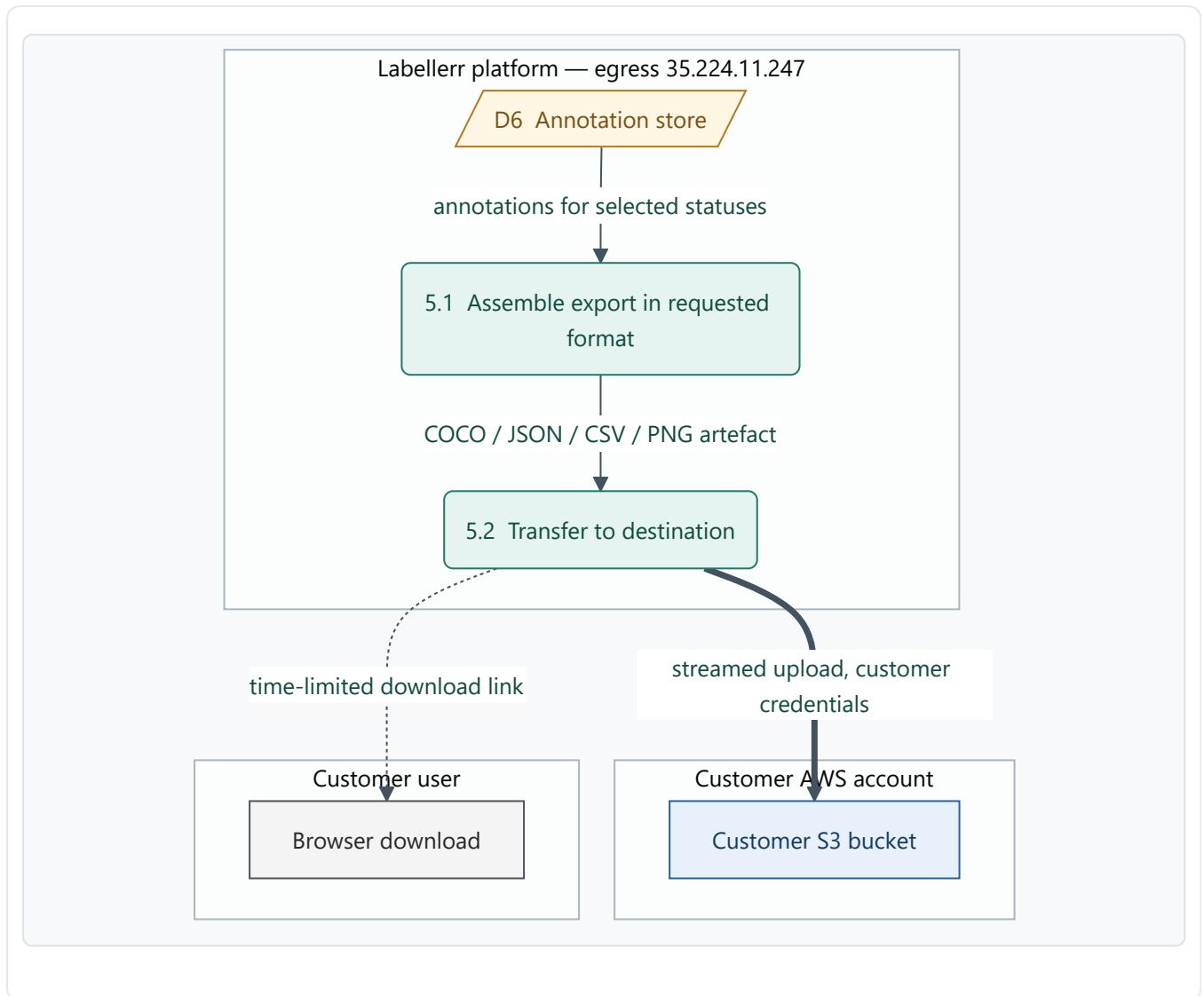


Embeddings are numeric vectors, not content. **D3** holds vector representations used for similarity search, and your original files are not recoverable from them.

The embedding model runs inside Labellerr's own Google Cloud project, in the same `us-central1` region, as a managed Google Cloud service. Your content is not sent to any external third-party API.

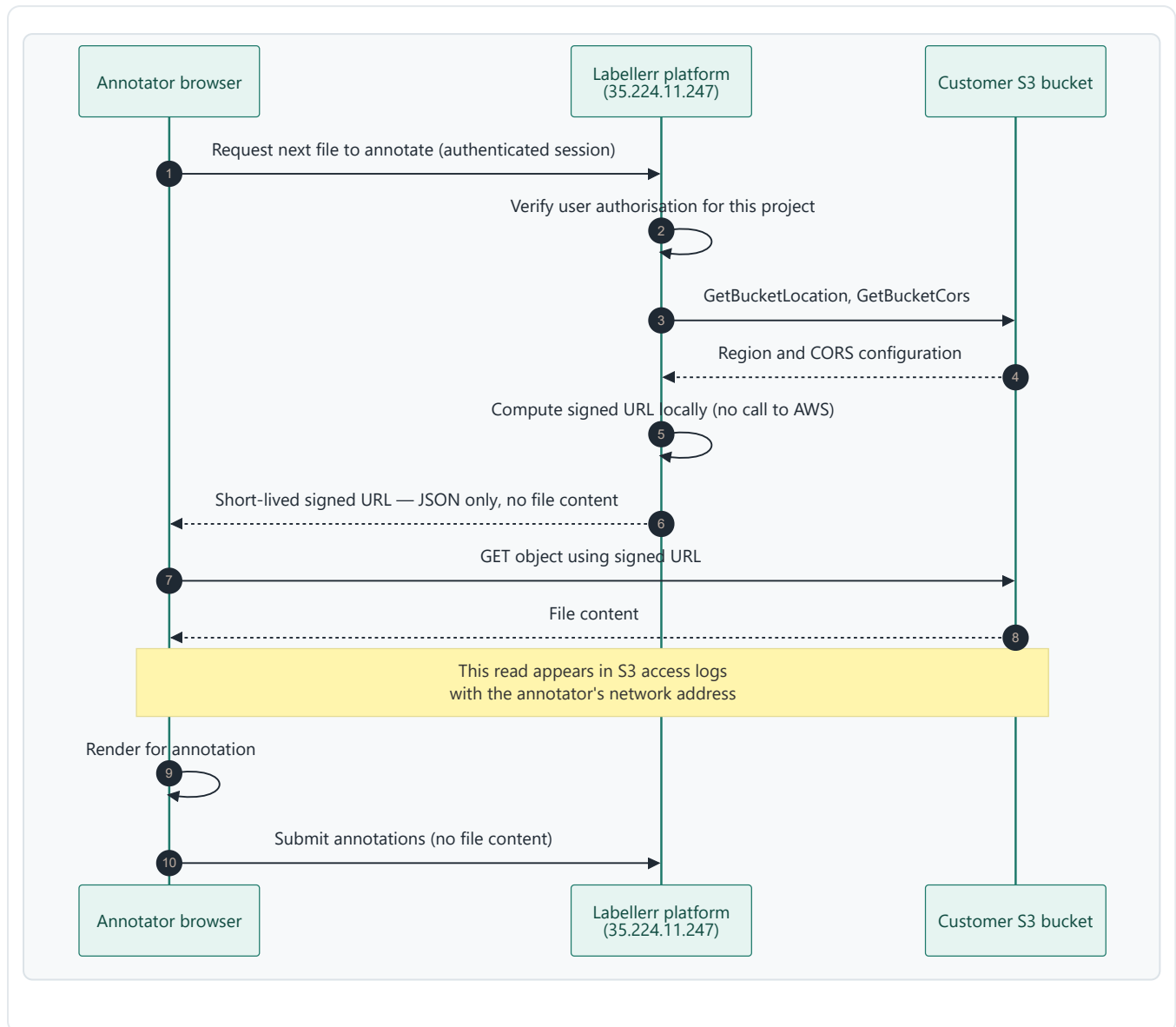
D5 applies to video only. The embedding model can only read video from Google Cloud storage, so a working copy of the video is staged there for the model to read. This is the one point in the S3 data flow where a copy of your content is held on Labellerr infrastructure, and it occurs only for video, and only under an explicit opt-in.

7. Level 1 — Export write-back



Exports contain **annotation data**, referencing source objects by path — not copies of your source files. You choose the destination: either a download from the Labellerr dashboard, or a direct write into a bucket path you specify. Direct writes are streamed rather than staged, so no intermediate copy is created, and are authenticated with your own credentials.

8. Sequence view — loading one file for annotation



9. Network identities accessing your bucket

Reconcile the following against your S3 server access logs or CloudTrail data events.

Operation on your bucket	When it occurs	Source network identity
ListBucket	Dataset creation and subsequent syncs	35.224.11.247
GetBucketLocation , GetBucketCors	Connection test; before issuing an access URL	35.224.11.247
PutBucketCors	Only if you grant it, during connection setup	35.224.11.247
GetObject , ranged — first 128 KB	Image import, metadata extraction	35.224.11.247
GetObject , full	Video import; model-assisted labelling; multimodal indexing if enabled	35.224.11.247
GetObject , full — annotation display	Every time an annotator opens a file	Annotator's network address
PutObject	Video frames written back; export write-back	35.224.11.247

Expect the large majority of `GetObject` calls to originate from annotator addresses, since every file view is a direct browser fetch. Labellerr does not know or record these addresses beyond normal application access logging, and they will vary with each annotator's network.

10. Data handling summary

Question	Answer
Does Labellerr copy your bucket's contents?	No. Images, audio and text are referenced by path; only metadata is read. Video is read for frame extraction, and the frames are written back to your own bucket.
Is file content stored on Labellerr infrastructure?	Only in one case: a working video copy staged for the embedding model, and only when multimodal indexing is explicitly enabled for that dataset.
Is file content written to disk during processing?	No. All processing runs on serverless instances whose writable filesystem is memory-backed. Content exists only in instance memory and is destroyed when the instance is torn down.
Is file content cached on Labellerr servers?	No. Server-side caching covers annotation records and file metadata, not file content.
Is file content cached in the annotator's browser?	Video frames only, in local browser storage, evicted on a least-recently-used basis. Images are held in page memory only. Audio and text are not cached.
Who can read the signed URLs?	They are issued only to authorised, authenticated users, are scoped to a single object, are read-only, and are short-lived and expire automatically.
Does Labellerr send your data to third parties?	No. Optional embedding generation uses a managed Google Cloud service inside Labellerr's own project and region.
What happens on project deletion?	Annotations, project records and search index entries are purged. Objects in your own bucket, including extracted video frames, remain under your control.

11. Access this connection requires

The IAM permissions below are the complete set Labellerr uses. Setup instructions are in the [AWS S3 connection guide](#).

Import (read): `s3:ListBucket` , `s3:GetObject` , `s3:GetBucketLocation` , `s3:GetBucketCors` , and optionally `s3:PutBucketCors` if you prefer Labellerr to configure CORS rather than setting it manually.

Export and video frame write-back: `s3:PutObject` , and `s3:DeleteObject` where you want Labellerr to be able to replace previously written artefacts.

Required CORS rule — without it, browsers block the direct fetch described in §5:

```
[
  {
    "AllowedHeaders": ["*"],
    "AllowedMethods": ["GET", "HEAD"],
    "AllowedOrigins": ["https://*.labellerr.com"],
    "ExposeHeaders": [],
    "MaxAgeSeconds": 3600
  }
]
```