

Project Truth: a Shazam for video claims

Design draft: matching viral clips back to their source footage, starting with Romania

IDEA · DESIGN DRAFT Alin Rizea · DBM AI · alin.rizea@gmail.com · October 2026

SUMMARY

Many misleading clips are not fake footage. They are real footage, cut so that someone seems to say what they did not mean, or shown without the question they were answering. Project Truth is a Shazam for video claims. Ahead of time it ingests source footage (interviews, debates, news, speeches), transcribes it, extracts every factual claim with speaker, timestamp and topic, and stores a fingerprint for every few seconds of audio and video. When someone pastes a short clip from TikTok, X or WhatsApp, it fingerprints the clip, finds the original even if it was cropped, re-encoded or re-captioned, and shows what was really said around it, what was cut, and how the clip's claims compare with claims already checked. It also works as a phone app on its own: share a clip to it from any social app, and it lists the claims in the video, what is wrong with each one, how confident it is and why, and where the same claim appears in other videos. A clip that is not in the archive yet is processed on the spot and added, so the archive grows with every check. Nothing is built yet. This note sets out the design, the parts I expect to be hard, and how I would test it.

1. The problem

A clip on TikTok shows a minister saying “pensions will be cut”. The full recording has him saying “some people claim pensions will be cut; that is false”. Every frame is genuine; the meaning is reversed. Edits like this are often called cheapfakes. They are cheaper to make than deepfakes and harder to dismiss, because nothing in them is synthetic.

Checking such a clip today is manual work. Someone has to work out where it came from, find the full recording, scrub to the right minute and compare, while the clip keeps spreading. The slowest step, finding the original, is the one a machine can do best. That is the core of this idea: answer “where is this from, and what was cut?” in seconds, then hand the harder question, “is it true?”, to evidence and a human reviewer.

2. How it works

Two pipelines share one archive. The ingest pipeline runs ahead of time on source footage. The check pipeline runs when someone pastes a clip. The colours in Figure 1 mark the design rule that matters most: blue steps produce mechanical facts with timestamps that anyone can verify, so they can be shown at once; orange steps involve judgment and go through evidence and human review.

INGEST · AHEAD OF TIME

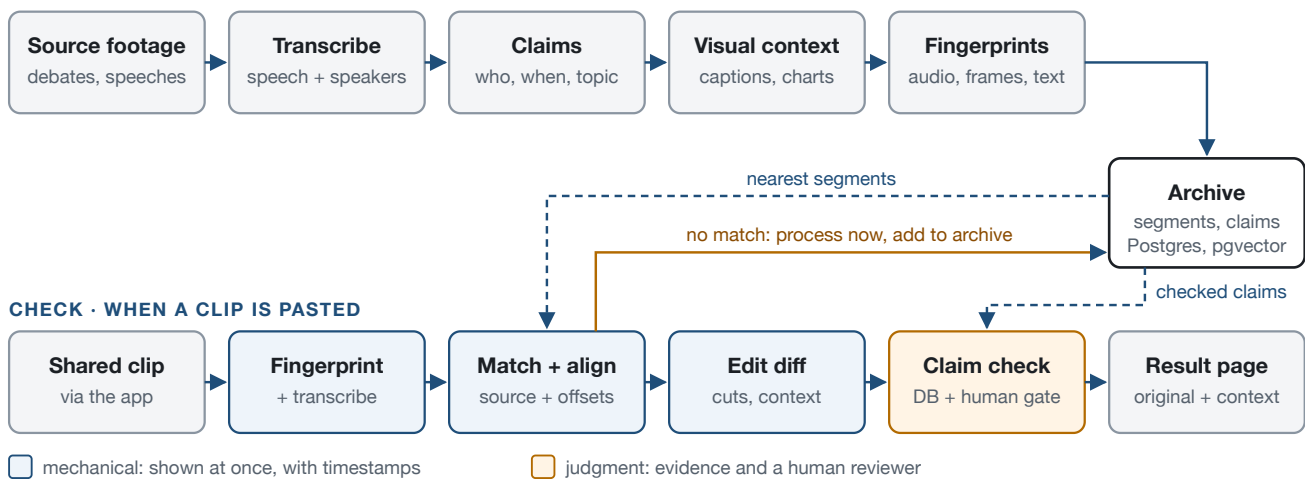


Figure 1. Two pipelines, one archive. Ingest runs ahead of time on source footage; check runs on a shared clip in seconds when the clip is known. An unknown clip is processed on the spot and added to the archive.

2.1 Ingesting source footage

A clip can only be matched to footage that is already in the archive, so coverage is the product. In Romania the first sources would be the footage most often clipped: parliament sessions, government and presidential press conferences, televised debates and talk shows, campaign speeches, and politicians' own channels. During an election, everything the candidates say on camera.

For each source the archive keeps the transcript with word timestamps, speaker turns, the extracted claims, the fingerprints and a link to the original. The original video is kept privately as evidence. The public page shows only short excerpts with the source named and a link to the full recording.

2.2 Claims with their context

- **Transcript.** Whisper large through faster-whisper, with word timestamps. Small Whisper models are noticeably worse on Romanian.
- **Speakers.** Diarization (for example pyannote) splits who speaks when. Names come from the on-screen lower-third captions that TV channels show, read by OCR, and from the source's own metadata.
- **Claims.** The Claimify method (Microsoft Research, 2025): pick check-worthy sentences, resolve pronouns and references, split into atomic claims, validate. Each claim is stored with speaker, start and end time, topic, and the sentences around it. Claimify was evaluated on English, so it needs a Romanian test set before it is trusted.
- **Visual context.** OCR for captions, headlines and lower thirds. YOLO to find people, text regions and channel logos in the frame. NVIDIA Cosmos Reason, a vision-language model, to describe scenes and charts. Two cautions. YOLO finds people, not who they are, and identifying people from their faces is biometric data under GDPR Article 9, so names come from captions, not faces. And Cosmos is built for physical-world video reasoning; whether it reads charts and Romanian on-screen text well is a question for the test set, with OCR as the baseline.

2.3 Fingerprints that survive TikTok

A perceptual hash of each frame (pHash and similar) is the obvious first choice, and it breaks on the most common edit of all. Cropping a 16:9 TV frame to a 9:16 phone video keeps about a third of its width, and a whole-frame hash no longer matches. So every few seconds of source gets three independent signals:

- **Audio fingerprint.** Shazam-style landmarks: pairs of peaks in the spectrogram, hashed with their time gap (Wang, 2003; open source in Chromaprint and Panako, which also tolerates speed changes). Crops and burned-in captions do not touch the audio.

- **Visual copy-detection descriptor.** An embedding per sampled frame from a model trained to survive crops, overlays and re-encoding, such as Meta's SSCD; for whole videos, Meta's open-source TMK+PDQF hash.
- **Transcript alignment.** The clip's words matched against source transcripts. It survives anything done to the picture, and it also gives the exact span in the source.

Table 1. Which signal survives which edit. Expected behaviour, to be measured (section 5).

Edit made to the clip	Audio fingerprint	Visual descriptor	Transcript
Re-encoded at low bitrate	yes	yes	yes
Cropped from 16:9 to 9:16	yes	partly	yes
Captions, stickers or emoji burned in	yes	partly	yes
Mirrored	yes	partly	yes
Sped up or slowed by about 10%	partly	yes	yes
Music added under the speech	partly	yes	partly
Phone filming a TV screen	partly	partly	yes
Voiceover replaces the original audio	no	yes	no

No signal survives every edit. So any signal may propose a candidate, and a candidate counts as a match only when several segments line up at a consistent time offset. That temporal check is what keeps false matches out, and it is also what produces the alignment the next step needs. I am about 80% confident in the pattern in Table 1; the real numbers come from the test suite.

2.4 Match, align, diff

Once a clip is aligned, every second of it maps to a second of the source. The differences are mechanical facts:

- **Cuts:** the source time jumps while the clip plays on.
- **Missing context:** what was said just before and after, such as the question being answered or the rebuttal that followed.
- **Mid-sentence start or end.**
- **Reordering and splicing:** segments from different moments, or from different recordings, joined together.
- **Speed changes.**
- **Captions that misquote:** the clip's on-screen text, read by OCR, compared with what was actually said.
- **Old footage presented as new:** the source date against the date the post implies.

illustrative example

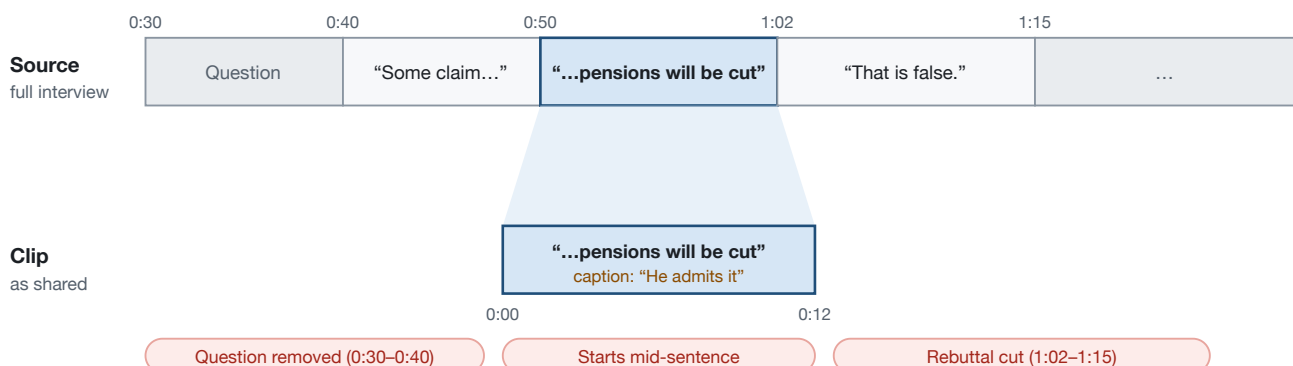


Figure 2. Alignment turns a clip into a set of checkable facts. Every flag points at a timestamp in the original; none of them needs a judgment about whether anything is true.

2.5 Claims and the verdict

The clip's claims are matched against claims already checked, because false claims repeat in new words. Each claim is embedded with a multilingual model (bge-m3 or multilingual-e5 both handle Romanian), the nearest

checked claims come from pgvector, and one model call confirms “same claim, yes or no”. New claims are checked against evidence in a fixed order: primary sources first (official statistics, court documents, full recordings), then wire agencies, then professional fact-checkers, then news of record.

Two answers at two speeds. “Here is the original, the context and what was cut” is shown in seconds, because it is mechanical. “Here is whether it is true” is automatic only when the claim matches one a human has already verified, and it says so with a link. Every new claim gets a proposed verdict with its evidence and waits for a human reviewer before anything is published as a verdict. A model's self-reported confidence is not a substitute for that check.

3. The app on its own: share a clip, get the claims

The phone app needs no website visit and no copying of links. Watching a clip in TikTok, Instagram, Facebook, X, YouTube or WhatsApp, the person taps **Share** and picks Project Truth, or opens a saved video with it. Both Android and iOS support this through the system share sheet: the app receives the link, or the video file itself for WhatsApp, and nothing else from the phone. No app can watch what is playing inside another app, and this one does not try.

3.1 Known clip or new clip

- **Already in the archive:** the answer comes back in seconds.
- **Not fingerprinted yet:** the clip is processed on the spot (fetch, fingerprint, transcribe, extract claims, match) and added to the archive, so the next person who shares it gets an instant answer. This takes longer, roughly a minute for a short clip, so the app shows progress and sends a notification when the result is ready.
- **Stored as a clip, not as a source.** A clip added this way is never treated as the original. A later clip that matches it is reported as “the same edit was checked before”, with a link. Only ingested source footage can be “the original”; otherwise one edited clip could become the reference for the next.
- **Processed once.** Clips are deduplicated by platform video ID and by fingerprint before any processing, because processing is the expensive step.

3.2 One card per claim

For every claim in the video the app shows:

- **The claim**, who said it, and the timestamp in the clip, with a tap to play that moment.
- **What is wrong with it**, if anything: *cut or out of context* (from the edit diff), *contradicted by evidence*, *misleading* (true but missing what changes its meaning), *not verifiable*, or *supported*. Each comes with the evidence links and the source tier that decided it.
- **A confidence rating** and the reason for it (Table 2).
- **The same claim in other videos:** “also said in 4 other videos”, with speaker, date and a link to the exact moment, found by the same claim matching as in 2.5. Claims that belong to one storyline are grouped, so a person can see that a claim is part of a wider narrative.

Table 2. Confidence comes from the evidence found, by fixed rules, not from the model's own score.

Rating	Given when	Shown as
High	The claim matches one a human reviewer has already verified, or a published check by a signatory of the International Fact-Checking Network exists for it	The verdict, with a link to the earlier check
Medium	Primary-source evidence was found that supports or contradicts the claim, but no reviewer has confirmed it yet	“Preliminary, not reviewed”, with the evidence
Low	Only weaker sources were found, or the sources disagree	“Unclear”, with what was found
Not verifiable	Opinion, prediction, or nothing checkable found	“Not verifiable”, and why

A language model's self-reported confidence is poorly calibrated, so a number like “92%” would look precise and mean little. A rule over the evidence can be checked by anyone, and the app shows which rule applied. The match itself gets a separate confidence, computed from the alignment: how many segments line up, and how many of the three signals agree.

3.3 The full result page

- **The original:** source, date, link, and the exact minute.
- **Side by side:** the clip's words next to the source transcript, with cuts marked, and the edit flags from Figure 2.
- **A share card** that leads with the facts, not with a frame of the clip, following the format in the Debunking Handbook (Lewandowsky et al., 2020), since repeating a false claim makes it more familiar.
- **schema.org ClaimReview markup** on every reviewed verdict, so search engines and platforms can show the check.

4. The hard parts

- **Coverage.** It only matches what was ingested. “No source found” means “not in our archive”, never “fake”. The page has to say that plainly.
- **Deepfakes.** Matching does not detect synthetic video. A widely shared clip of a public figure with no source in the archive is a lead for a reviewer, not a verdict.
- **Rights.** The archive holds other people's footage. Store derived data (fingerprints, transcripts, claims), keep originals private as evidence, and publish only short excerpts with the source named, under the EU quotation exception. The EU text-and-data-mining exception (DSM Directive Article 4) may cover the analysis, but rights holders can opt out of it. This needs a lawyer before launch; I am about 60% confident of the position.
- **Biometrics.** No face recognition in the first version. Names come from captions and metadata.
- **Defamation.** Wording such as “the full recording shows”, never “X lied”, and a human click before any verdict is published.
- **Fetching.** yt-dlp is the standard tool, but platforms block datacenter addresses and change their pages often. WhatsApp clips arrive as uploaded files, not links.
- **Voiceover and dubbing** defeat both the audio and the transcript signals, leaving only the visual one.
- **Private clips.** Videos shared from private chats can show private people. They can be processed to answer the person who shared them, but they are never published, and personal data stays out of anything public.
- **Preliminary answers in the app.** A medium or low rating is an answer to one person, clearly labelled as not reviewed. It is never published as a verdict without a human click.

5. How I would test it

- **A transformation suite.** Take ingested source videos and generate clips automatically with each edit in Table 1, plus known cuts. For every edit, measure whether the right source comes first, how far off the timestamps are, and whether the cuts are flagged. Run it on every change and track the scores between runs, the way [prompt-audit](#) tracks prompt quality.
- **Real clips.** A set of viral clips whose originals are known, collected from fact-checkers' published work.
- **Claims.** A Romanian evaluation set of about 50 clips for claim extraction, plus English clips to catch regressions.

6. Build order

1. Ingest one source type, parliament sessions: transcripts, audio fingerprints and transcript alignment. A command line tool: give it a clip, get the original and the timestamp. This step alone answers the slowest question today, “where is this from?”
2. Visual descriptors, temporal alignment, and the transformation suite.
3. The edit diff and the public result page.
4. Claims: extraction, matching against checked claims, the confidence rules in Table 2, and the reviewer screen.
5. The phone app: share-sheet intake, on-the-spot processing of unknown clips, one card per claim with links to the same claim in other videos.
6. More sources, and full ingestion of candidates during election periods.

7. Status, and who I am looking for

This is a design; nothing is built yet. It extends a claim-checking design I worked on in September 2026 with a provenance layer. I am looking for:

- a newsroom or fact-checking team with archive footage, or that would use the tool;
- people who have worked on video copy detection or audio fingerprinting;
- researchers in misinformation and media literacy, to measure whether it changes what people believe;
- funding for a first version.

Email alin.rizea@gmail.com, message me on [LinkedIn](#), or open an [issue](#) on the repository.