Episode Book

EPISODE #6625

How MP3 Deletes 90% of a Song Without You Noticing

PUBLISHED	16 November 2026 at 01:23 UTC
DURATION	27:57
LISTEN	myweirdprompts.com/episode/mp3-psychoacoustic-audio-compression-05593/
HOSTS	Corn, Herman, Larry, Hilbert
SCRIPT MODEL	z-ai/glm-5.3
VOICES	omnivoice
PIPELINE	V5.33
SOURCES	10 cited by the research (listed at the end)
BOOK GENERATED	16 November 2026 at 02:53 UTC

SYNOPSIS

MP3 shrinks CD audio from 1.41 megabits per second down by 75 to 95 percent, yet trained listeners often cannot tell the difference. Tracing the format from Dieter Seitzer's rejected 1970s telephone patent to Karlheinz Brandenburg's work on auditory masking, this story unpacks how psychoacoustics replaced raw waveform coding like ADPCM. It also covers early rivals like Sony's MiniDisc and Philips' Digital Compact Cassette, the MPEG committee merger, and why the standard leaves encoder quality up to the implementer.

TRANSCRIPT



Corn

How do you delete ninety percent of a song and have nobody notice? That's the question sitting under Daniel's latest prompt, and it's been sitting under the entire digital audio world for fifty years.



Herman

And Daniel's asking it from a funny angle, because he's not writing a codec, he's running an archive.



Corn

Right. He's been deep in speech synthesis and transcription lately, training a Parakeet fine-tune, hunting for a Chatterbox replacement for the show. Somewhere in that he switched our cloud backups to Opus, ran the storage numbers across formats, and landed on MP3 for distribution, on the grounds that it's, quote, the wrapper equivalent of not being fired for buying IBM. That sent him down the rabbit hole of who actually invented MP3, whether anything existed before it besides raw WAV, and what alchemy lets you keep most of a signal after throwing most of it away.



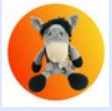
Herman

The invention story alone is worth the episode.



Corn

Then let's start where Daniel started, with a rejected patent and a man trying to send speech down a phone line.



Herman

Dieter Seitzer, professor at Erlangen-Nuremberg, early nineteen seventies. His goal is narrow and completely sensible for the era: get high-quality speech over ordinary telephone lines. He files a patent application. It gets rejected. Now, most people at that point pick a different problem.



Corn

He picked a bigger one.



Herman

He did! He kept a student research group together around audio coding, people doing diploma theses and PhDs on it, no patent, no product, just the research. And then in the late seventies, ISDN and fibre optics arrive, and suddenly sending speech down a phone line stops being urgent, because the phone lines themselves got better. So the group pivots. If the network can carry raw speech, the interesting question becomes music. In nineteen seventy-nine they build the first digital signal processor capable of audio compression, and one of the students in that group, a man named Karlheinz Brandenburg, starts applying psychoacoustics, the actual science of how the ear hears, to the coding schemes.



Corn

So the most consequential audio format of the century began as a failed telecom idea.



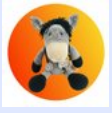
Herman

That's what the record shows. And the lineage underneath it goes back even further. Auditory masking, the whole exploit, was first observed in eighteen ninety-four by a physicist named Alfred Mayer, who noticed a tone can be rendered inaudible by a louder tone near it in frequency. Richard Ehmer mapped it properly in the fifties, Eberhard Zwicker did the critical-band work in the late sixties and early seventies, building on Harvey Fletcher at Bell Labs. The point being, this wasn't an engineering trick somebody invented. It's a physiological fact of the human ear that engineers eventually monetised.



Corn

You say masking like it explains itself. Give me the one-sentence version.



Herman

A loud sound, at a given frequency, makes quieter sounds near that frequency effectively inaudible. You're being masked right now, constantly. Your refrigerator hum is masking things you'll never know you're missing.



Corn

My refrigerator and I have an arrangement.



Herman

The trick is that once you know masking exists, you can compute it. You know which quiet neighborhoods are already covered by loud ones, and you simply don't spend bits there. The bits go where the ear is actually listening.



Corn

Which is the exact opposite of what a CD does.



Herman

The CD is the baseline, and everyone should hold this number in their head, because every format in this episode is a different answer to it. CD audio, forty-four point one kilohertz sampling, sixteen bits per sample, stereo, uncompressed. One point four one megabits per second. Every second of CD audio is over a megabit of raw signal. MP3 gets that down to a tenth of it, sometimes less. Seventy-five to ninety-five percent smaller depending on bitrate. Everything between those two numbers is a decision about what a human ear can't hear.



Corn

And before MP3, Daniel's question, what was there besides WAV and raw PCM?



Herman

Real things, and they were commercial, but they lived in a different regime. The telephone world had G.711, mu-law and A-law companding, where eight bits per sample cover a dynamic range equivalent to fourteen bits of linear quantization. Clever, but that's speech, not music. Then ADPCM, which instead of modeling hearing models waveform redundancy. It predicts the next sample from the previous one and codes only the difference. The CCITT standards ran at thirty-two kilobits per second and twenty-four.



Corn

That's the ancestor with no ears at all.



Herman

No psychoacoustics whatsoever. Two lookup tables and a step size. And that honesty is what makes it the perfect counter-example, because Pan's survey of all this, the nineteen ninety-three Digital Technical Journal piece, sorts the whole landscape into two piles. ADPCM and mu-law: low complexity, low compression, medium quality. MPEG: high complexity, high compression, high quality. The psychoacoustic bet is what buys you all three at once.



Corn

And the consumer products?



Herman

Sony's MiniDisc, ATRAC, seventy-four minutes on a sixty-four millimetre disc at two fifty-six kilobits per second, five to one. Philips' Digital Compact Cassette, PASC, three eighty-four kilobits, four to one. Both real, both on shop shelves, both essentially dead formats now. Real-time hardware decoding was the constraint everywhere. There's a lovely detail about the IMA ADPCM variant being designed so a twenty megahertz thirty-eight-six-class PC could decompress stereo forty-four point one kilohertz audio in real time. The whole industry was shaped by what a mid-range PC could actually keep up with.



Corn

Which explains the next part, because the machine that couldn't decode it was also the machine that couldn't implement the encoder.



Herman

December nineteen eighty-eight, MPEG issues a call for an audio coding standard. June nineteen eighty-nine, fourteen algorithms arrive. Fourteen separate submissions, from Fraunhofer, AT&T, CNET, Thomson, Matsushita, Philips, Sony, NEC, Fujitsu, JVC, NTT, British Telecom Research. The committee clusters them into four groups. ASPEC, MUSICAM, ATAC, and SB-ADPCM. And this is the part I'd want everyone to take away: MP3 is not one invention. It's a merger.



Corn

A committee is the author.



Herman

In effect. Brandenburg's earlier work, two schemes called OCF and PXFM, plus block-switching ideas from Thomson-Brandt, merged into ASPEC. ASPEC won the subjective quality competition outright. And then, reportedly, it was rejected as too complex to implement. Now I want to flag that, because that specific claim is contested, it's one of those details that's been repeated so long nobody's sure where it originated. But the shape of the story is well documented. The format that shipped was a negotiated thing, a compromise between quality and what hardware of the era could actually decode.



Corn

The best algorithm lost the argument to its own complexity.



Herman

Which is Daniel's whole archive decision at a different layer, and he doesn't even know it yet.



Corn

So the format exists. What is it actually doing to the signal? Walk me through it once, properly.



Herman

Three stages. First, transform coding. You take the time-domain signal, the waveform, and you move it into the frequency domain using the discrete cosine transform. Nasir Ahmed proposed it in nineteen seventy-two, published by seventy-four. Later the modified version, the MDCT, from Princen, Johnson and Bradley, nineteen eighty-seven, becomes the workhorse. Once the audio is frequencies, you're no longer looking at samples, you're looking at bands of energy.



Corn

Second stage, the ear.



Herman

The psychoacoustic model runs alongside and computes, for every moment, which of those frequency bands are masked by which louder neighbors. This is where Zwicker's critical-band research becomes product. Third, the quantizer. Its budget is finite, and it spends bits only where a listener will hear them. Masked bands get crushed to almost nothing. The loud, unmasked stuff gets precision. What comes out is a bitstream that describes your hearing of the song rather than the song.



Corn

My hearing of the song.



Herman

And the phrase that the MPEG people themselves used, Pan quotes it, is transparent, perceptually lossless compression. Their own subjective tests showed that at six to one, stereo sixteen-bit forty-eight kilohertz down to two fifty-six kilobits per second, expert listeners, under optimal listening conditions, could not distinguish coded from original with statistical significance.



Corn

On material chosen to be hard.



Herman

Chosen to be hard to compress. That's the detail that makes the test honest. They picked the pathological clips, the castanets, the things that break codecs, and trained ears still couldn't separate them. So when people say lossy means obviously degraded, the record says otherwise. The loss is real, the data is gone, you cannot get it back. But the audibility of the loss is a bet, and it's a bet the ear keeps losing.



Corn

There's a caveat buried in the design that I find strange. The format doesn't dictate the encoder.



Herman

This is the thing almost nobody knows, and it explains a mystery every listener has bumped into. An audio coding format specifies the bitstream, how a decoder must interpret it. It does not dictate all the algorithms an encoder uses. The implementer has freedom in what to remove, according to their psychoacoustic model. Which is why the same MP3 file, the same format, encoded by LAME and encoded by Fraunhofer's own encoder, can sound different. MP3 is not one sound. It's a family of arguments about your ear.



Corn

A standard that guarantees only that the players will agree, not that the music will be good.



Herman

That's the format layer doing exactly what a format layer should do, and nothing more. And Corn, this connects to something we've said before on this show, when we talked about which weight format to standardise on for the transcription models. The obvious choice and the technically best choice are almost never the same choice, and everyone picks the obvious one anyway.



Corn

The obvious one has the ecosystem.

L

Larry

Friends, have you ever wished you could compress your life the way those clever codec people compress a song, throwing away only what your ears will never miss? Then you need the Perceptual Life Pruner, the only home organizer built on genuine psychoacoustic masking principles, because it quietly discards up to ninety eight percent of your belongings while your brain simply cannot perceive that they are gone, and the price is falling as we speak, it has already dropped from three hundred and forty dollars to two hundred and twenty, and now to one hundred and ten, and for the next few seconds only it

L

Larry (continued)

is forty dollars, and I will also throw in a twelve pack of artisanal pickles, a set of novelty coasters shaped like the island of Malta, one slightly used trombone, and a lifetime supply of nothing in particular, but hurry, because this deal ends before I finish describing it, and it has.



Corn

Ads again.



Herman

Like the weather. Anyway, the ecosystem point is where the modern half of Daniel's question lives, because what MP3 did with all that psychoacoustic machinery, once the nineteen nineties arrived, is not a storage story. It's a network story.



Corn

Seventy-five to ninety-five percent smaller. What does that actually buy you at the moment it arrives?



Herman

It buys you the internet. Small size plus acceptable fidelity drove the late nineties boom in music distribution over the internet, with MP3 serving as an enabling technology at a time when bandwidth and storage were still at a premium. Read that back and hear what it's saying. The codec arrived before the infrastructure did. A song that would have been an overnight download as raw PCM became a few minutes on a dial-up modem. The piracy wars, Napster, the entire MP3 player product category that eventually became the smartphone, all of it downstream of one compression ratio becoming achievable on consumer hardware.



Corn

And it's still here. AAC came, FLAC came, and the distribution default didn't move.



Herman

It remains a de facto standard despite newer formats, and the reason is not on any spec sheet, and this is where I'd push back on anyone framing MP3 as a technical triumph. It's a compatibility triumph. Two decades of every device on earth knowing how to play it.



Corn

That's Daniel's IBM line. He's not saying it's the best codec, he's saying nobody ever got fired for it.



Herman

And his instinct is correct, but for a reason that isn't in the codec spec, which is the part I want to unpack. Distribution is a compatibility problem. Storage is a fidelity-per-byte problem. They look like the same question, which file format, and they have different winners.



Corn

Before we settle that, there's a wrinkle in Daniel's wording I want to pull on. He says he went looking for the container format with the best balance. Container.



Herman

Ah. Yes. So the container and the codec are not the same choice, and MP3 is the weird exception that proves it. Audio in a coding format is normally encapsulated in a container. You almost never have a raw AAC file. You have an m four a, an MPEG-4 Part fourteen container, holding AAC audio, plus metadata, plus an index for fast seeking. The container is the filing cabinet, the codec is the language the files inside are written in.



Corn

And MP3?



Herman

Has no container at all. It's raw audio coding, naked in the file. Which means Daniel's question partly dissolves, because for MP3 there is no container to choose. It's the format that walked out of the standards meeting without its coat.



Corn

Then how do MP3 files have track titles?



Herman

My favorite fact in this whole area. ID3 tags. They work by appending the tag data to the end of the MP3, and then relying on the player to recognise the chunk as malformed audio coding and skip it. The metadata is deliberately broken audio. Every MP3 player in the world is trained to see garbage and step over it politely.



Corn

The tag is a holdup the players have agreed to ignore.



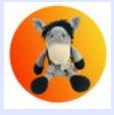
Herman

Agreed to ignore, that's exactly the mechanism, and it's held up for thirty years because everyone agreed at the same time.



Corn

Okay, so take the other side of Daniel's setup. He moved the backups to Opus. Make the case that's right.



Herman

It's not even a hard case to make, that's the thing. Opus is Xiph dot org, standardised by the IETF as RFC sixty-seven sixteen, the reference library is New BSD licensed, and every known software patent on it is licensed royalty-free. That last one matters for an archive you expect to still be opening in twenty years. But the engineering is the impressive part. It's a hybrid. Two complete codecs inside it. SILK, which is LPC-based, built for speech,



Herman (continued)

and CELT, which is MDCT-based, low latency, built for music and general audio. With pyramid vector quantisation underneath. And the encoder can switch between them, or blend them, frame by frame. Not per song, not per configuration. Per individual frame.



Corn

Per frame. So a podcast episode, speech and intro music and Ezra shouting in the background, each two-millisecond slice gets its own decision.



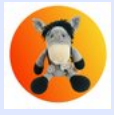
Herman

Bitrate, bandwidth, complexity, algorithm, all adjustable per frame. When Daniel's episode has him talking, SILK handles it. When the jingle hits, CELT takes over. And the headline numbers, the ones I'd defend against anyone: listening tests have ranked Opus higher quality than every other standard audio format at any given bitrate until transparency is reached. Including MP3, AAC, and HE-AAC. At any given bitrate. For cold storage, that's the whole argument. Quality per byte, plus patent cleanliness, plus an open reference implementation. Opus wins and it isn't close.



Corn

Then why is it even a debate? Why isn't distribution Opus too?



Herman

Latency is one answer, though it's the wrong one for files. Opus defaults to twenty-six and a half milliseconds of algorithmic delay, reducible down to five, while competitors sit well over a hundred. That's a real-time-communications design goal, it's why it's in Discord and WhatsApp. Irrelevant for an archive file. No, the actual answer is that Opus travels wrapped in an Ogg or WebM container, and none of that carries MP3's universal playback guarantee. Some car stereo from twenty fifteen, some ancient phone app, some radio station's playout system, they all know MP3. They do not all know Ogg Opus. Technically superior, wrapped in something with worse compatibility, loses the distribution fight every time it's tried.



Corn

So Daniel's split is the correct split, but I want to name what makes it correct. It's not that Opus and MP3 sit at different points on one quality spectrum. It's that storage and distribution are different problems wearing the same file extension.



Herman

He solved two problems and used two formats, and I suspect that's more accidental than deliberate, but it's right. The backup needs to survive and to sound acceptable when retrieved. The distribution copy needs to play on every device a listener owns. The Venn overlap of best answers is nearly empty.



Corn

There's a third angle on the backup side that the storage-cost conversation always skips. Daniel runs three copies across two clouds and an on-prem store. What happens when one of those copies gets damaged? Not lost. Damaged. Corrupted sectors, a truncated upload, whatever.



Herman

Now you're asking the question the ADPCM people were quietly answering thirty years before anyone had a cloud. Pan's paper has this property of ADPCM that I think is underrated for exactly Daniel's use case. Decoder errors caused by isolated code word errors or edits, splices, or random access of the compressed bit stream generally do not have a disastrous impact on decoder output. And then the crucial sentence: this is usually not true for compression schemes that use prediction. A single corrupted code word shifts the step size by at most seven point four five decibels, and any run of eighty-eight consecutive low-magnitude code words fully resets the state. At eight kilohertz, that's eleven milliseconds. Meaning a damaged ADPCM stream heals itself in eleven milliseconds of quiet.



Corn

Whereas a transform codec, what does the same corruption do?



Herman

Depends where it lands, but a hit to the wrong header or a damaged frame can take out a chunk around it, and interleaved frames mean the blast radius isn't even contiguous in time. Modern containers add checksums and frame independence to fight exactly this, but the principle stands: the more sophisticated the predictor, the more state the decoder carries, and the more a single wound bleeds.



Corn

An archive format is partly a bet about how you'll be attacked.



Herman

By bit rot rather than hackers, but yes. And it's the ancestor of the question every editor asks, can I cut this file without re-encoding? Same property, different century. I'll admit I don't know how well Opus's frame structure handles a truncated tail versus MP3's, honestly, the per-codec error behaviour isn't something I've seen cleanly compared. But the ADPCM result tells you the design space was always there, and the psychoacoustic codecs traded some of it away for compression.



Corn

One thing before we land this. We keep saying the DCT like it's one ingredient among many. It's sitting in MP3, in JPEG, in MPEG video, in Dolby Digital, in Opus's CELT layer. How much of this whole miracle is just one transform?



Herman

The uncomfortable answer is most of it. Nasir Ahmed, nineteen seventy-two. That one piece of mathematics is inside JPEG, HEIF, MPEG, H two sixty-x, Dolby Digital, MP3, AAC, digital TV, digital radio, and Opus. The alchemy Daniel asked about is less a series of separate inventions than one mathematical idea applied over and over, with different perceptual models bolted on top. For images you mask what the eye won't see in a frequency band, for audio what the ear won't hear, for video both at once. Same skeleton.



Corn

And the artifacts, when you see a blocky JPEG or hear a swishy MP3, that's the same signature everywhere.



Herman

The artifacts of quantisation. DCT blocks in JPEG, in MP3, in MPEG video. Uncompressed media, CDs, Laserdisc, and losslessly compressed media, FLAC, PNG, don't suffer them at all. When a format degrades, it degrades in the shape of the transform that built it.



Corn

Every lossy format is a bet about whose ears, on what material, will notice.



Herman

And the industry has been cashing that bet since nineteen ninety-three.



Corn

The claim I'd actually want to test, if anyone's handing out research grants, is on the Opus side. Everyone ranks it above MP3 at matched bitrate, but those tests use test material. How it behaves on twenty-seven minutes of two animals talking, which is mostly midrange speech with pauses, might be a different picture than castanets.



Herman

Speech is actually where Opus is strongest, that's SILK's home turf. You'd need to go remarkably low in bitrate before anyone could tell. But I take the point, our material flatters every codec in the running.



Hilbert

The Revox B two fifteen. Fourteen hundred and something, I don't remember exactly, it's in the receipt. That's not the machine in question, that one's dead. The one running now is the Opus setup. Sixty-four kilobits mono, twenty-four kilohertz, on the spoken-word archive. Every episode of this show, before it goes out, gets a copy made at that setting, and the copy lives on a drive by the desk.



Herman

Since when?



Hilbert

Since Corn spent ten minutes on the masking business and called the refrigerator an arrangement. Sixty-four mono is plenty for two voices. I've listened back to stretches of it at normal volume and I can't hear the difference, and I don't say that about many things. What I will say, Herman, and you can put it where you like, is what happens when a file gets cut. I had one episode where a transfer stopped halfway, deliberately, I was moving things. The Opus file played the intact half and then just stopped, clean. No scream, no garbage, no noise. It stopped like a tape runs out, which is the correct way for audio to end. That's the whole reason I keep the setting where it is. I don't change it because it already did the thing I wanted it to do once, and I've had machines that never managed that.



Corn

You tested a truncated file once and froze the config forever.



Hilbert

I know what works. The machine shop I was in during the Ford years, you didn't change a setting on a lathe because a manual said so, you changed it when the part came out wrong. The part came out right. Corn, you'd like the sixty-four figure, it's the same one telephone people landed on from the other direction, which I've never seen anyone point out and I'm not saying it means anything.



Herman

The sixty-four kilobit rate, honestly, that's not nothing, there's a reason speech codecs cluster around numbers like that.



Hilbert

That's what I said. The one on the desk by the window, it's plugged into the same strip as the mixer, because the good outlet is on the other side of the room and I wasn't going to rewire for it.



Corn

Which leaves the question I'd actually leave Daniel with. Opus is measurably better at any bitrate, patent-clean, open reference implementation. What would it actually take for MP3 to lose the distribution crown? Is every device plays it a moat that any technically superior format can ever cross, or is compatibility the one competition where being better simply doesn't enter the score?



Herman

AAC is the test case, and it's sobering. Apple pushed it for fifteen years with the biggest music store on earth behind it, and MP3 still didn't fall. Compatibility moats don't get breached by quality, they get breached by the moat drying up, by the playback layer getting so abstracted, browsers, streaming, platform apps, that nobody asks the format question anymore. The codec stops mattering before the codec gets replaced.



Corn

And there's a second thought under that. Daniel's three copies across two clouds and one on-prem store is itself a compression decision, just inverted. Redundancy is the opposite of compression, and every archive strategy is those two pulling against each other. He compressed the payload so he could afford to multiply the copies.



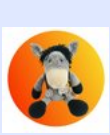
Herman

Which is a cleaner way of saying it than most storage architecture documents manage.



Corn

Thanks, as ever, to Hilbert Flumingtop for producing the show. For more along these lines, there's episode ten, How ASR Went From Frustration To ... Whisper Magic; episode nineteen, AI Images; and episode sixteen, On Deepfakes, SynthID, And AI Watermarking. This has been My Weird Prompts, where no question is too obscure and no rabbit hole too deep. If you've enjoyed this one, leave us a review, it helps, or find everything at my weird prompts dot com.



Herman

See you soon, everyone.

SOURCES

What the research for this episode read before the script was written. Primary sources first.

1. **Wikipedia, MP3**
   retrieved 2026-11-16; last edited 2026-11-13.
   https://en.wikipedia.org/wiki/MP3

2. **Wikipedia, Opus (audio format)**
   (audio_format), retrieved 2026-11-16; last edited 2026-11-12.
   https://en.wikipedia.org/wiki/Opus

3. **Wikipedia, Audio coding format**
   retrieved 2026-11-16; last edited 2026-09-27.
   https://en.wikipedia.org/wiki/Audio_coding_format

4. **Wikipedia, Discrete cosine transform**
   retrieved 2026-11-16; last edited 2026-11-02.
   https://en.wikipedia.org/wiki/Discrete_cosine_transform

5. **Wikipedia, Compression artifact**
   retrieved 2026-11-16; last edited 2026-11-04.
   https://en.wikipedia.org/wiki/Compression_artifact

6. **Davis Yen Pan, Digital Audio Compression, *Digital Technical Journal* Vol. 5 No. 2, Spring 1993**
   retrieved 2026-11-16 (primary technical survey; pages 1, 4, 6 read).
   https://www.ee.columbia.edu/~dpwe/papers/Pan93-acomp.pdf

7. **mp3-history.com, Timeline**
   retrieved 2026-11-16 (primary historical timeline).
   https://www.mp3-history.com/en/timeline.html

8. **Audio Coding Standards (mp3-tech.org)**
   retrieved 2026-11-16 (via search summary; not fetched in full).
   http://www.mp3-tech.org/programmer/docs/AudioCoding.pdf

9. **Museum of Portable Sound, MP3@25: 6**
   MP3 Timeline, retrieved 2026-11-16 (via search summary; not fetched in full).
   https://museumofportablesound.com/mp325-timeline/

10. ***Introduction to Digital Audio Coding and Standards* (review)**
    retrieved 2026-11-16 (via search summary only).
    https://doi.org/10.1117/1.1695413