Sind wir eigentlich schon tot? Wir haben es gecheckt.

Die Uno-Reverse Card zu Deathbyclawd ;)

2026-04-08

(english below)

Deutsch

Vor ein paar Wochen machte ein Tool die Runde, das in der SaaS-Welt für einige Unruhe gesorgt hat: deathbyclawd.com — der selbsternannte "SaaSpocalypse Survival Scanner". Die Prämisse ist so simpel wie brutal: Du gibst die URL deines Produkts ein und bekommst einen "Death Score" — wie wahrscheinlich es ist, dass Claude dein gesamtes Produkt durch eine einzige Markdown-Datei ersetzen kann.

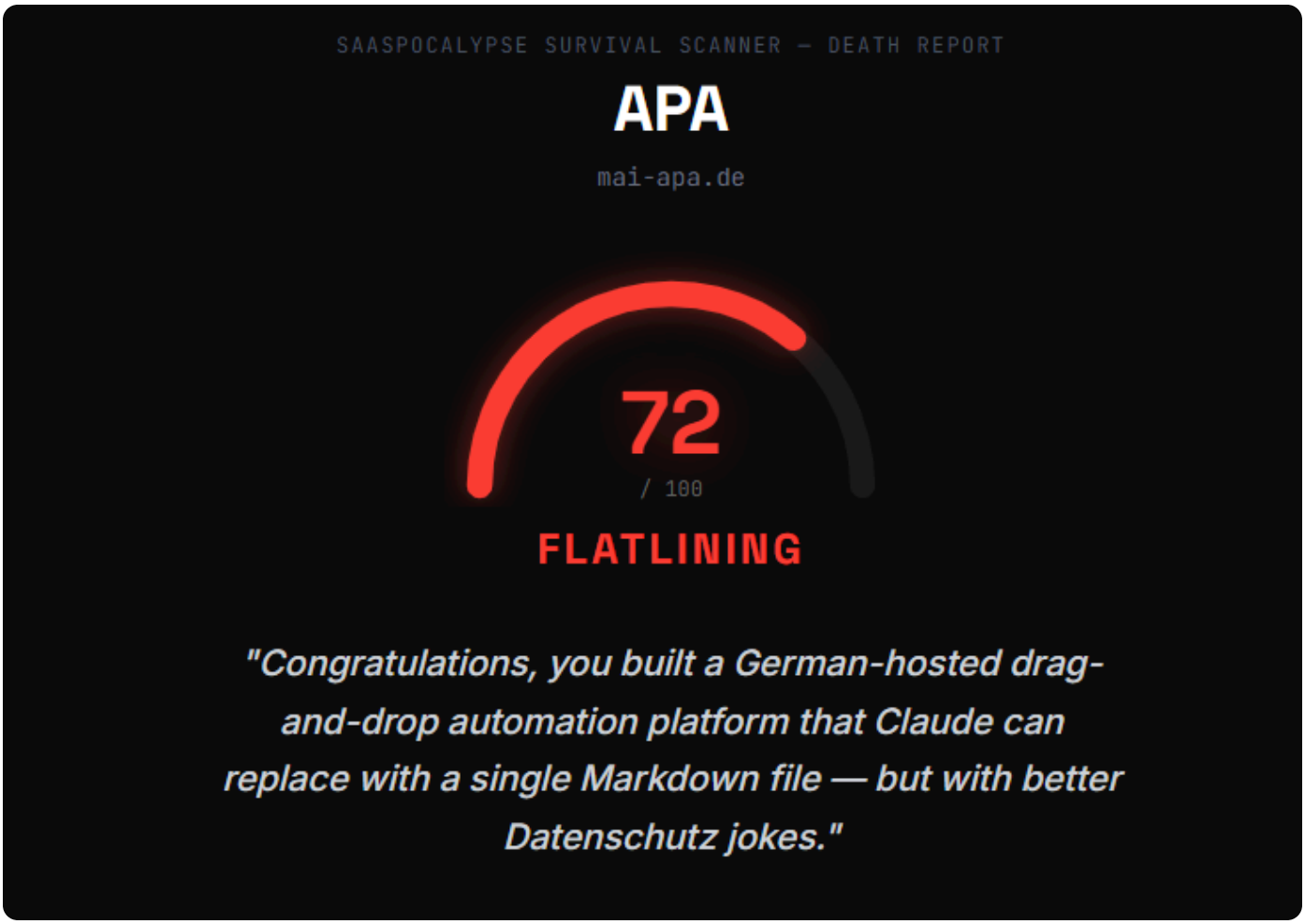
- Intuit? Sweating.
- Adobe? Sweating.
- Microsoft? Immortal — aber mit dem Hinweis: "You can't replace Microsoft with a .md file, but Microsoft is absolutely trying to replace YOU with one."

Das Tool ist Satire (und zugegebenermaßen ziemlich witzig & durchaus relevant). Aber wie Analyst Anand Iyer auf X schrieb: "It's a gag, but the underlying taxonomy it accidentally produces is actually useful." Der ernste Kern: Viele SaaS-Produkte sind im Kern UI-Wrapper um Logik, die ein LLM heute direkt ausführen kann. Die Frage, ob das auf das eigene Produkt zutrifft, ist keine rhetorische.

Also haben wir es einfach ausprobiert. Natürlich wollten wir wissen: *Sind wir schon tot?*

Das Ergebnis: Das Tool hat... nichts gefunden.

Kaum erkennbare Logik. Kaum crawlbare Struktur. Klinisch flatlining.



Kurze Ehrlichkeit: Tiefes Lachen beim Lesen unserer letzten Worte *"But... but we have a data center in Frankfurt! FRANKFURT! Claude doesn't even have a Impressum!"* - es ist tatsächlich verdammt witzig (try it out...).

Dann aber setzt der Realitäts-Check ein.

Und hier liegt die eigentliche Pointe — die man nicht verpassen sollte.

deathbyclawd ist im Kern ein Claude-basiertes Tool. Es analysiert, was Claude aus einer URL lesen kann. Was es nicht lesen kann, existiert für es nicht. Das Tool urteilt also: "Kein Befund — wahrscheinlich nichts Relevantes da."
Aber genau das ist der Denkfehler, den wir mit APA täglich bekämpfen: LLMs treffen Annahmen dort, wo sie keine Informationen haben. Sie generieren, wo sie eigentlich nachfragen müssten. Sie sehen eine leere Oberfläche und schlussfolgern: tot.
Ein Architekt würde fragen: *Warum ist da nichts?*

deathbyclawd demonstriert live seinen eigenen blinden Fleck — und dieser blinde Fleck ist exakt der Grund, warum APA existiert.

Was das mit unserer Architektur zu tun hat

deathbyclawd funktioniert, indem es die Oberfläche eines Produkts scannt — Struktur, Logik-Muster, erkennbare Architektur. Produkte mit hohem Death Score haben eines gemeinsam: Ihre Kernlogik liegt offen oder die offen liegende Logik ist sehr knapp bemessen. Sie sitzt in der UI, ist crawlbar, ist replizierbar. Genau das macht sie angreifbar.
Dass das Tool bei uns ins Leere läuft, hat einen einfachen Grund: Es gibt bei mai-apa.de keine Oberfläche, unter der die Logik liegt. Die Logik ist vollständig von der Oberfläche getrennt.

APA (Application Programming Application) ist kein UI-Builder, der Code generiert, während du zusiehst. Es ist ein semi-deterministisches Orchestrierungssystem aus tausenden voneinander abhängigen Prompt-Ketten — jede davon darauf ausgelegt, spezifische Fehlermodi von LLMs in der Softwaregenerierung abzufangen, bevor sie sich durch den Stack fressen.

Frontend, Backend, Datenbank, API-Schicht: alles dekoppelt. Rollenlogik, Freigabeprozesse, Schnittstellen, Sonderfälle — alles modelliert, bevor eine einzige Zeile Code generiert wird. Das ist kein Designprinzip aus dem Lehrbuch. Es ist das Ergebnis von hunderten gescheiterten Generierungsläufen, aus denen wir gelernt haben, wo LLMs systematisch versagen — und wie man das verhindert.
Ein Surface-Scan findet das nicht. Weil es nicht an der Oberfläche liegt.

Warum das gerade jetzt für Unternehmen relevant ist

Die SaaSpocalypse, die deathbyclawd satirisch beschreibt, **ist real — aber sie trifft nicht alle gleich.**

Was gerade passiert: Unternehmen experimentieren massenhaft mit Vibe-Coding-Tools, die schnell etwas Aussehendes produzieren. Das Problem ist nicht die Geschwindigkeit. Das Problem ist, was danach kommt: Code ohne Architektur, Logik ohne Struktur, Prototypen, die niemand warten kann — und Sicherheitslücken, die niemand sieht, weil sie von Anfang an eingebaut wurden. Studien schätzen, dass KI-generierter Code heute bereits für einen signifikanten Teil der wachsenden globalen technischen Schulden verantwortlich ist.
Der Unterschied zwischen einem Prompt und einem Framework ist genau das: Ein Prompt

liefert eine Antwort. Ein Framework liefert eine Architektur, die standhält, wenn das Unternehmen wächst, die Anforderungen sich ändern und die erste Euphorie über den schnellen MVP verflogen ist.

Genau dafür ist APA gebaut — nicht für den schnellen Demo-Moment, sondern für Software mit langfristigem Planungshorizont.

Fazit - Die Uno-Reverse Karte

Wenn ein Claude-basiertes Tool unsere Architektur nicht lesen kann — dann ist das vielleicht der klarste Beweis, den wir je für unsere Arbeit bekommen haben.
Danke, deathbyclawd. Wirklich.

> *Quellenreferenz: https://www.youtube.com/watch?v=WfjGZCuxl-U&t=2s*
> *Hashtags: #deathbyclawd*

English

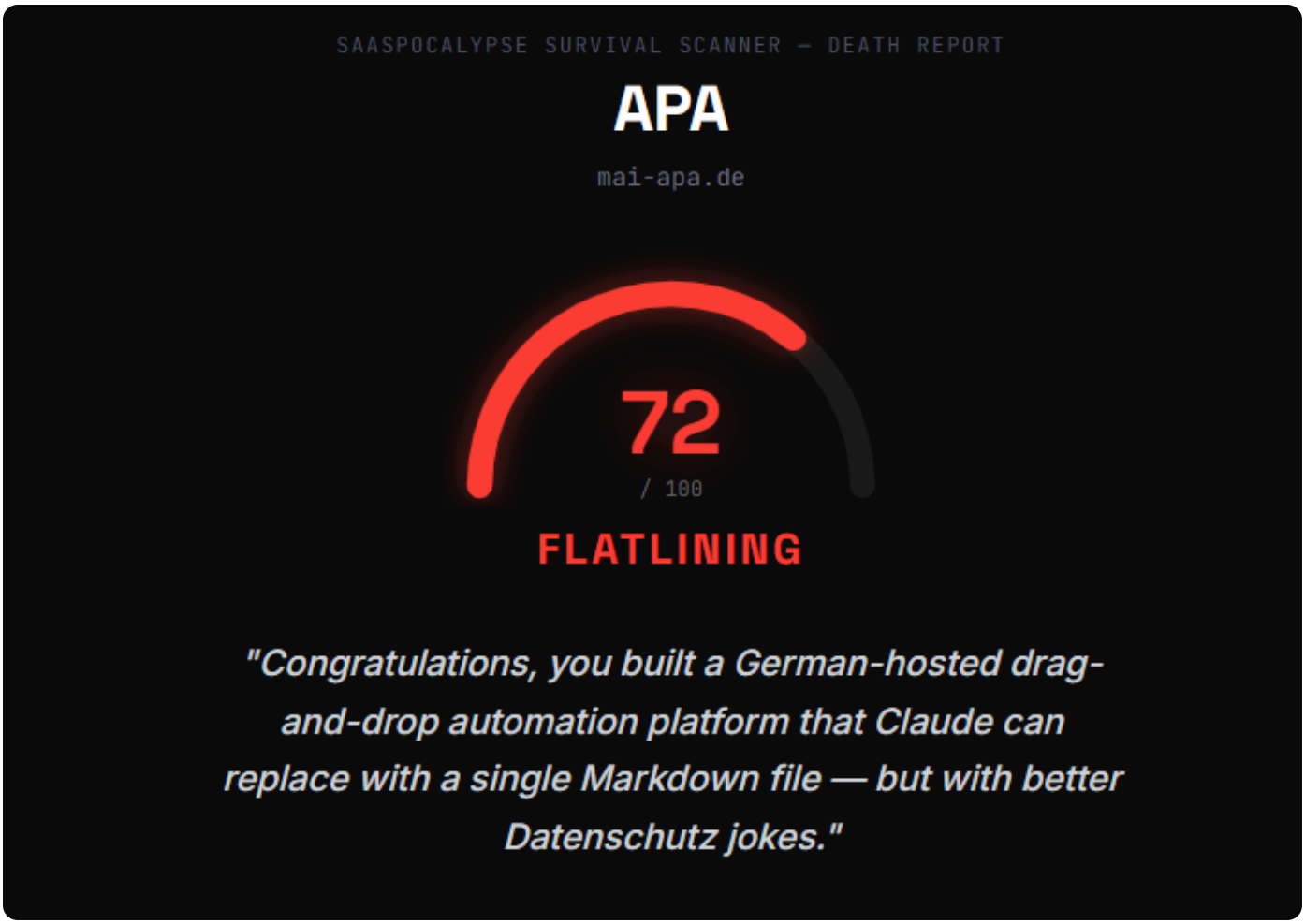
A few weeks ago, a tool went viral in the SaaS world that caused quite a stir: deathbyclawd.com — the self-proclaimed "SaaSpocalypse Survival Scanner".
The premise is as simple as it is brutal: you enter your product's URL and get a "Death Score" — the likelihood that Claude could replace your entire product with a single Markdown file.

- Intuit? Sweating.
- Adobe? Sweating.
- Microsoft? Immortal — with the note: "You can't replace Microsoft with a .md file, but Microsoft is absolutely trying to replace YOU with one."

The tool is satire (and honestly very funny). But as analyst Anand Iyer wrote on X: "It's a gag, but the underlying taxonomy it accidentally produces is actually useful." The serious core: many SaaS products are essentially UI wrappers around logic that an LLM can now execute directly. Whether that applies to your own product is not a rhetorical question.
So we tried it. We had to know: **Are we already dead?**

The result: The tool found... nothing.

No recognizable logic. No crawlable structure. Clinically flatlining.



Brief honesty: deep laughter reading our supposed last words — *"But... but we have a data center in Frankfurt! FRANKFURT! Claude doesn't even have an Impressum!"*

Then reality sets in.

And here's the actual punchline — the one you shouldn't miss.

deathbyclawd is, at its core, a Claude-based tool. It analyzes what Claude can read from a URL. What it can't read doesn't exist for it. So the tool concludes: "No findings — probably nothing relevant here."
But that's exactly the failure mode we fight with APA every single day: LLMs make assumptions where they have no information. They generate where they should ask questions. They see an empty surface and conclude: dead.
An architect would ask: *Why is there nothing there?*
deathbyclawd live-demonstrates its own blind spot — and that blind spot is

precisely why APA exists. This isn't a defense. It's product validation by the opposition.

What this has to do with our architecture

deathbyclawd works by scanning a product's surface — structure, logic patterns, recognizable architecture. Products with high Death Scores share one thing: their core logic is exposed. It sits in the UI, is crawlable, is replicable. That's exactly what makes them vulnerable.
The reason the tool finds nothing at mai-apa.de is simple: there is no surface under which logic is hiding. The logic is fully separated from the surface.

APA (Application Programming Application) is not a UI builder that generates code while you watch. It's a semi-deterministic orchestration system of thousands of interdependent prompt chains — each designed to catch specific LLM failure modes in software generation before they propagate through the stack.

Frontend, backend, database, API layer: all decoupled. Role logic, approval workflows, interfaces, edge cases — all modeled before a single line of code is generated. This is not a textbook design principle. It's the result of hundreds of failed generation runs, from which we learned where LLMs systematically break down — and how to prevent it.
A surface scan won't find that. Because it's not at the surface.

Why this matters for companies right now

The SaaSpocalypse that deathbyclawd satirizes is real — but it doesn't hit everyone equally.

What's happening: companies are experimenting en masse with vibe-coding tools that produce something that looks good quickly. The problem isn't the speed. It's what comes after: code without architecture, logic without structure, prototypes nobody can maintain — and security gaps nobody sees because they were built in from the start. Studies estimate that AI-generated code is already responsible for a significant share of the world's growing technical debt.

The difference between a prompt and a framework is exactly this: a prompt delivers an answer. A framework delivers an architecture that holds up as the company grows, requirements shift, and the initial excitement over a fast MVP fades.

That's what APA is built for — not the quick demo moment, but the software that still runs afterward. For companies that want to genuinely digitize their processes, not just build prototypes.

Conclusion - The Uno-Reverse Card

If a Claude-based tool can't read our architecture — that might be the clearest proof of our work we've ever received.
Thank you, deathbyclawd. Genuinely.

> *Quellenreferenz: https://www.youtube.com/watch?v=WfjGZCuxl-U&t=2s*
> *Hashtags: #deathbyclawd*
