# The Protocol Was Not the Pizza

We learned to detect AI before we learned to evaluate AI-assisted work.

DENNIS HEDEGREEN ANALYSE V1.0

HTTPS://HEDEGREENRESEARCH.COM/ARTICLES/THE-PROTOCOL-WAS-NOT-THE-PIZZA/

I brought a pizza project into a developer forum and asked people to attack the architecture.

The project was Pizza4People. The public proof was deliberately small: a client could discover a restaurant node through a registry, fetch the menu directly from that node, and send the order directly to it. The registry helped two endpoints find each other, but it was not supposed to own the order, payment, customer account, or restaurant data.

At the time, I was explicit about what did not exist yet. There was no production payment layer, no delivery system, no finished CVR binding, and no claim that this was ready to run a real restaurant.

I asked for technical feedback, especially on node identity and registry design.

The first thing some readers inspected was something else.

One reply said the project looked like “Claude Design” at first glance.

I answered:

> Codex ;)

Another reader went further. The work was “AI slop,” they wrote. “Not original thought.”

The interesting part is that the first observation was basically correct. AI was all over the work. It was in the prose, code, interface work, research assistance, and iteration loop. Anyone who smelled AI had detected something real.

The harder question is what that detection actually told them.

## The smell is information

AI has made the surface of competence cheap.

A polished project page used to be weak evidence that somebody had spent time building and explaining something. It was never proof of quality, but effort left traces. Today a person can produce a large repository, fluent documentation, illustrations, landing pages, and confident technical prose extraordinarily quickly.

Readers adapt.

They look at tone, formatting, vocabulary, visual style, repository shape, even sentence rhythm, and ask whether the thing in front of them was generated. That is not irrational. When production becomes cheap and attention remains scarce, people need triage.

We already read through labels before we read through claims. Left. Right. Institutional. Activist. Expert. Amateur. Those labels do not settle whether a particular statement is true, but they change where we look for failure and how much attention we are willing to spend.

AI is becoming another pre-reading label.

Sometimes that label may be enough for the decision at hand. A forum may decide it does not want low-effort generated posts. A reader may simply decide not to spend ten minutes inspecting another polished prototype. If the question is provenance — whether somebody really wrote a passage they claimed to write unaided — AI involvement can be directly relevant.

Nobody owes every AI-assisted object a forensic audit.

The mistake begins when the triage signal is asked to answer questions it cannot answer.

“Looks AI-generated” can justify suspicion.

It cannot, by itself, establish that the builder does not understand the work, that nothing original happened, that the claims were not checked, or that the architecture is bad.

AI smell may tell us where inspection should begin — or whether we want to inspect at all.

It cannot tell us what the inspection would find.

## Six questions compressed into one

The public conversation around AI-assisted work often collapses several different questions into one:

**Was AI used?**

That question is easy to ask and increasingly easy to answer. In my case the answer was yes.

But it is not the only question that matters.

There is **provenance**: where did the text, code, evidence, and ideas come from?

There is **agency**: who set the problem, constraints, trade-offs, and direction?

There is **comprehension**: does the person presenting the work actually understand the system and its failure modes?

There is **verification**: what was checked rather than merely generated?

There is **responsibility**: who owns the claims, errors, corrections, and consequences?

And there is **substance**: does the thing hold up even if we stop caring how it was produced?

These are not six scores for a dashboard. They are diagnostic questions.

A project can be openly AI-assisted while having strong human agency, strong comprehension, serious verification, clear responsibility, and useful substance.

A completely human-written project can fail all of those tests except provenance.

AI changes the urgency of the inspection. It does not perform the inspection for us.

That distinction matters to me because my own workflow is not well described by pretending I am a traditional lone writer who occasionally uses autocomplete.

I am often closer to an editor than a sole reporter.

I choose the question. I decide what the project is allowed to claim. I set evidence boundaries. I send work to models. I reject drafts. I ask for new research passes. I decide which technical direction is acceptable and which one violates the purpose of the system. I make the publication decision, and I own the correction if the result is wrong.

Models participate materially. Sometimes they write a lot. Sometimes they find things I did not know. Sometimes they propose architecture. Sometimes their proposal is better than mine.

That does not make the workflow magically human-authored because I clicked Accept.

Accepting AI output creates responsibility. It does not, by itself, create authorship.

Authorship lives closer to the decisions.

And editorial authority lives in the ability to commission, constrain, reject, revise, publish, and correct.

That is a harder standard than counting who typed the most characters.

## Then someone inspected the object

The Reddit thread became much more useful when the criticism moved from the smell to mechanisms that could actually fail.

One reader asked the obvious market question:

If Wolt and Just Eat are valuable mainly because they bring customers, how does a protocol solve customer acquisition?

It does not.

My answer was that customer acquisition could be a separate service layer. That remains part of the P4P direction, but the criticism exposed something important: separating a function architecturally does not prove that the resulting market will adopt it.

Another reader asked a sharper economic question: why would anyone build on an open layer without owning the valuable connection?

I answered with TCP/IP, Linux, Red Hat, Canonical, and the distinction between infrastructure and service businesses.

That answer contained a real principle, but the analogy was carrying too much weight. The existence of profitable businesses on open technical infrastructure does not prove that a restaurant-ordering commons can bootstrap demand, fund support, or attract providers.

That remains an open problem.

Then the thread got technical.

Who hosts the restaurant node?

Who hosts a registry?

Why does the node sign itself? A signature can show control of a key, but who establishes that the key belongs to the restaurant the node claims to represent?

That criticism landed directly.

I replied that self-signing was not enough. The signed announcement could establish continuity — that the same key was controlling the node over time — but it could not certify the real-world restaurant. Root trust and CVR binding were separate problems, and they were not built.

That is not AI criticism.

That is protocol criticism.

A little later came an even cleaner question.

If registries are replaceable, how does the client know which registry to use in the first place?

In the proof implementation, it was hardcoded.

I said so.

The bootstrap mechanism was not solved, and I accepted “decentralised web API” as a fair description of the proof at that stage.

That comment contained more information about the system than any judgement about whether the page looked generated.

It identified a missing mechanism.

It could change the architecture.

## The criticism that wins gets to change the work

The current P4P repository is more explicit about several of these boundaries than the project I posted into that thread.

The public proof remains deliberately narrow. It still claims only that a restaurant node can be discovered without a marketplace middleman owning first contact, while menu fetch and order submission go directly from client to node. It explicitly says the signing key is not a finished trust or certification system.

The architecture now states the distinction in even plainer language:

> control of a node key does not certify the restaurant.

It also describes a broader registry direction in which no single registry is required forever, with scoped umbrella, vertical, country, and local registries and mirror/federation work in the prototype.

That does not mean the Reddit criticism has been “solved.”

A mature federation is not the current public proof.

Restaurant adoption remains open.

The economics of provider businesses remain open.

Real-world verification remains a later trust layer.

And a controlled live restaurant pilot is still a different gate from a public protocol proof.

That is exactly why specific criticism is more valuable than generic hostility.

A useful criticism does not need to be polite.

It needs to identify a claim, a mechanism, a contradiction, missing evidence, or a failure condition clearly enough that something can change.

“This smells generated” tells me something about trust.

“Self-signing proves key control, not restaurant identity” tells me what the system does not prove.

Those are different information objects.

## The builder was part of the failed interface

There is an emotionally convenient version of this story.

I brought a valid but unusual idea to a forum. People saw AI and failed to understand it.

That version is too easy.

If I wanted architectural criticism but readers repeatedly thought I was showing them another takeaway app or startup pitch, then the presentation itself was part of the failure.

A public does not owe a project the category the builder had in mind.

If the protocol is buried under a product-looking surface, the category error is partly an interface bug.

If the page describes more system than the proof demonstrates, the page is ahead of the evidence.

If AI assistance makes the language feel detached from the person responsible for the work, that person has to restore a visible chain of judgement.

The builder is also inside the field test.

That lesson has survived much better than the temptation to blame the room.

## Better criticism, not softer criticism

None of this is an argument that people should become less suspicious of AI-assisted work.

I think the opposite is probably necessary.

When output becomes abundant, verification becomes more important.

When models can produce convincing explanations instantly, comprehension matters more.

When code can appear faster than a person can inspect it, runnable proofs and explicit limitations matter more.

When authorship becomes distributed across people and models, responsibility needs to become more visible, not less.

The answer is not to demand that every reader patiently evaluate every generated object across six dimensions.

The answer is to be precise about which judgement we are actually making.

“I do not want to spend attention on this” is a legitimate decision.

“This forum does not want this kind of generated material” can be a legitimate moderation rule.

“This person does not understand the system” is a different claim.

“The architecture is wrong” is a different claim.

“There is no original judgement here” is a different claim.

Those claims need evidence that reaches the thing they claim to know.

A smoke detector can tell you that something deserves attention.

It cannot write the fire report.

## Back through the same door

The original thread changed P4P.

Not because every criticism was correct.

Not because the forum was a court.

Because a forum can be a sensor, and some of what it sensed survived inspection.

I took those reactions out of the fast thread and into a slower format where I could separate provenance, agency, comprehension, verification, responsibility, and substance. Some comments became weaker when separated from tone. Others became stronger.

Publishing that analysis only here would be too convenient.

This is my site. I control the format, the length, the surrounding context, and the final edit.

So this article is going back to the same place the first test happened.

Not to ask anyone to admit I was right.

Not to prove that Reddit was wrong about AI.

To run the test again.

If the distinction between AI smell and substantive evaluation is wrong, I want to know where it breaks.

If the six questions collapse in a way I have missed, say how.

If I have selected the criticism too conveniently, identify what I excluded.

If P4P still contains the same architectural failure under better documentation, point to it.

The first thread improved the project because some people were willing to be specific.

Don’t be nicer this time.

**Be more precise.**

---

## Source note

The primary field source is the original r/dkudvikler thread (https://www.reddit.com/r/dkudvikler/comments/1sz5esh/jeg_byggede_en_open_protocol_proof_for_direkte/), “Jeg byggede en open protocol proof for direkte restaurant-bestilling — Just Eat lukker i morgen i DK.” The article uses the thread as evidence of public reactions and of the state of the proof described at the time.

Current P4P claims are checked against the public repository's PROOF.md (https://github.com/hedegreen/P4P/blob/main/PROOF.md) and ARCHITECTURE.md (https://github.com/hedegreen/P4P/blob/main/ARCHITECTURE.md). The current proof explicitly distinguishes node-key control from restaurant certification and keeps federation, trust, real restaurant operation, and mature module interoperability outside the narrow `v0.1` proof claim.

## Production note

This article was developed through an AI-assisted editorial workflow. I set the question, evidence boundaries, revisions, and publication decision. AI systems participated materially in research, drafting, technical reading, and editing. I remain responsible for the claims and corrections.

---

RELATION MEMORY

RETURNS TO Når en platform forlader markedet · Returns to the P4P question through the limits of an AI-assisted public protocol proof.

NEARBY Day 0.5 — Don't Trust It Yet · Both pieces argue that inspectable limitations and correction matter more than a convincing surface.

---

CANONICAL URL: HTTPS://HEDEGREENRESEARCH.COM/ARTICLES/THE-PROTOCOL-WAS-NOT-THE-PIZZA/
PDF VIEW GENERATED: 2026-08-25T16:44:16.893Z