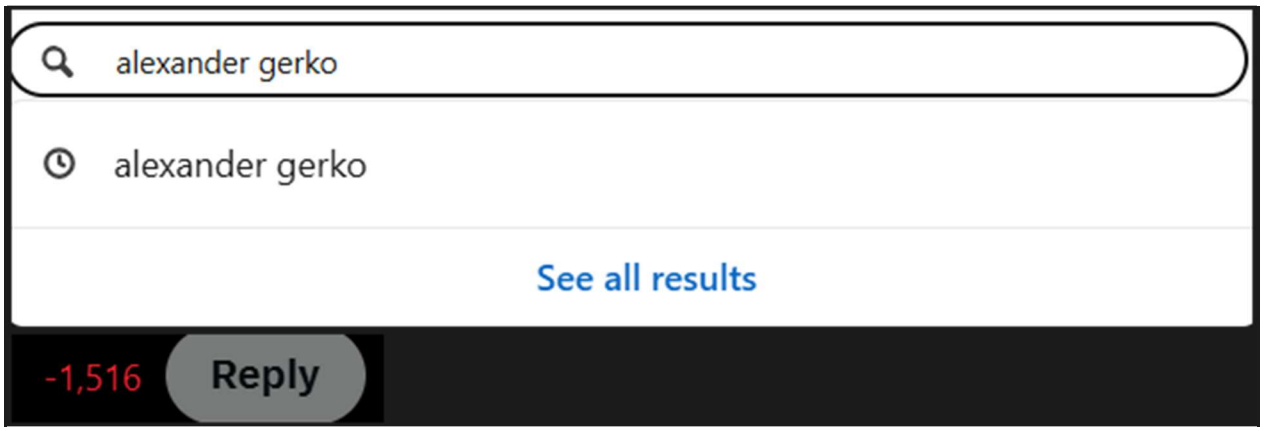

alexander gerko
alexander gerko
See all results
-1,516
Reply

Post Analytics

Calvin Thomas @TestTempAI · 1m

Replying to @VitalikButerin

@VitalikButerin

1 1 0

Impressions
1

Engagements
2

Detail expands
0

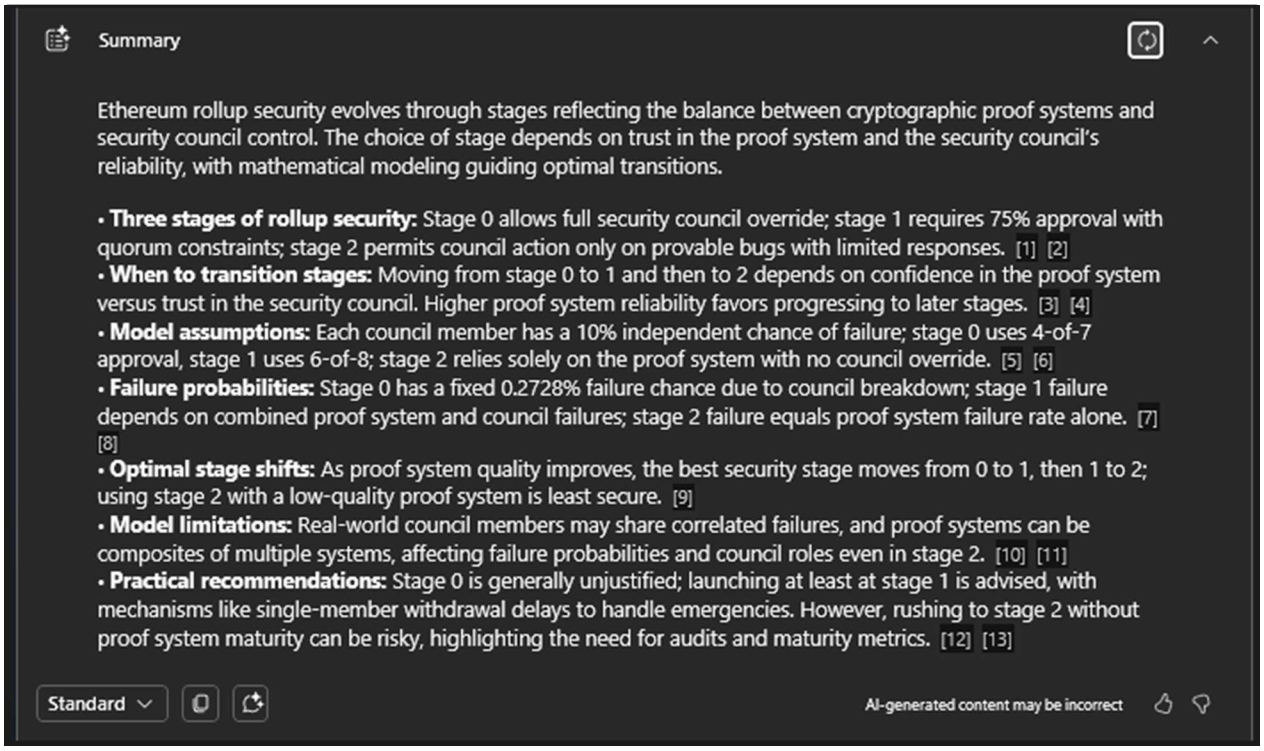
Profile visits
0

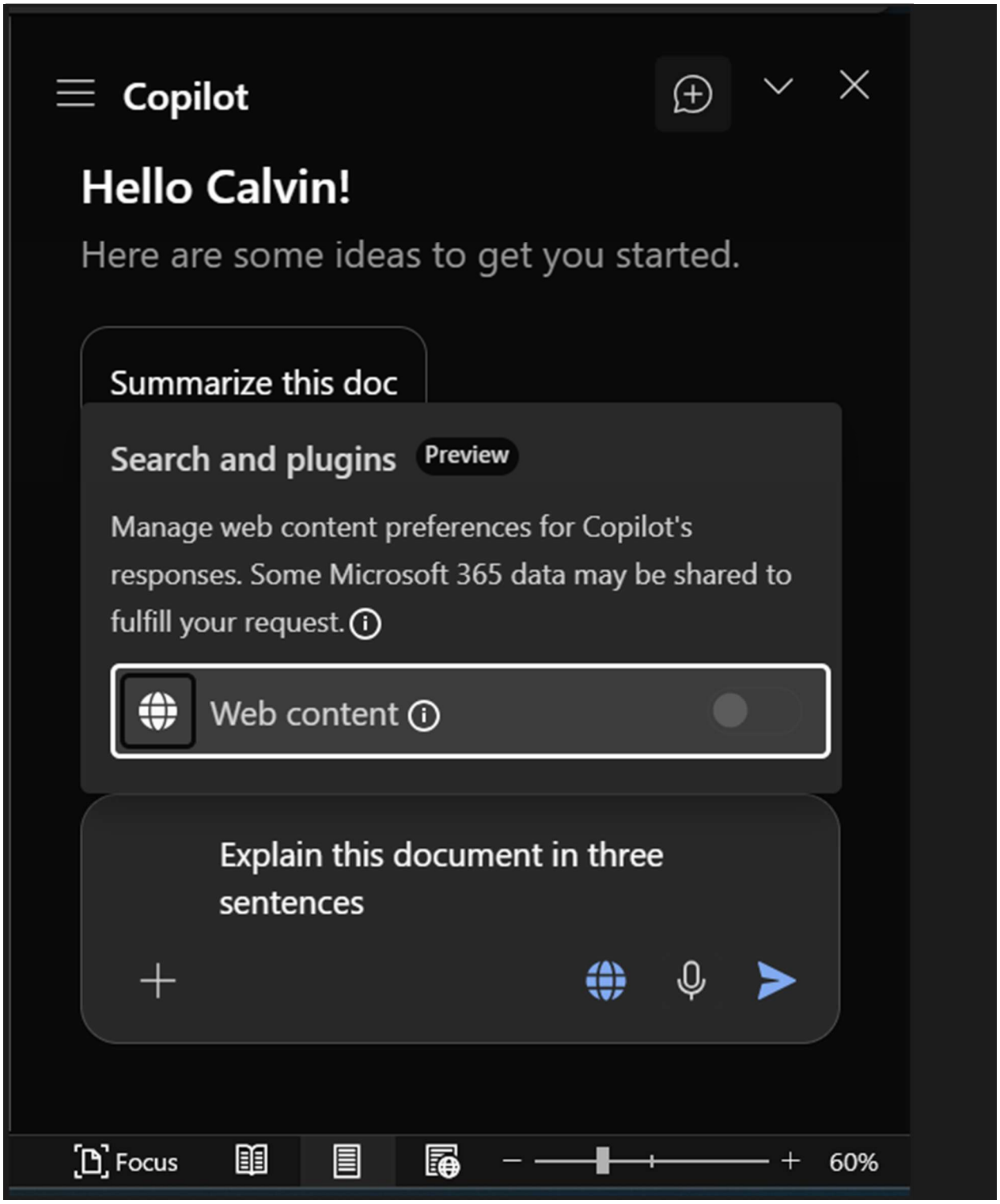
Nothing to see here - yet

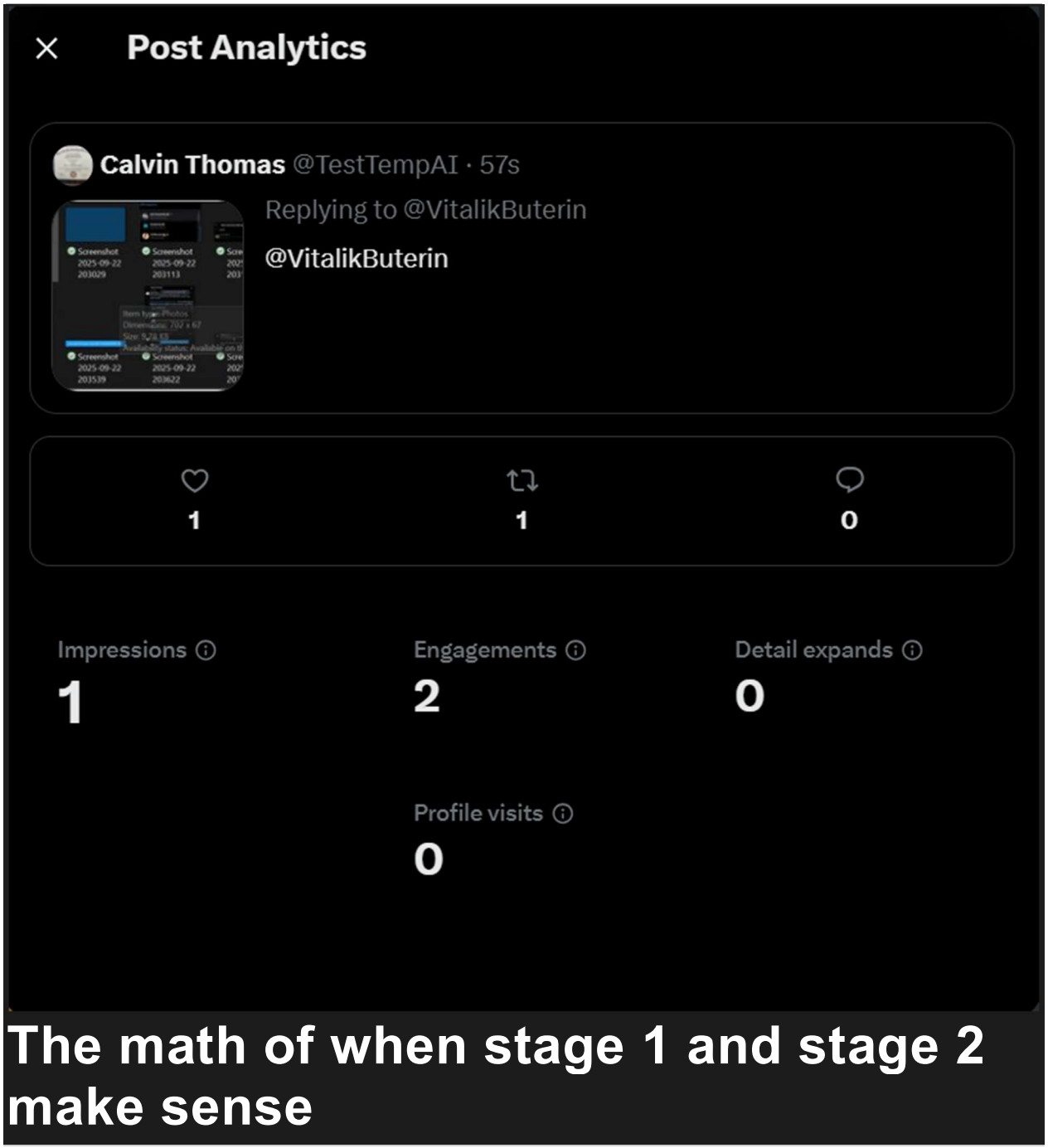
Once more data is collected, you'll see that information here.

Standard

AI-generated content may be incorrect


Summary
Ethereum rollup security evolves through stages reflecting the balance between cryptographic proof systems and security council control. The choice of stage depends on trust in the proof system and the security council's reliability, with mathematical modeling guiding optimal transitions.
• Three stages of rollup security: Stage 0 allows full security council override; stage 1 requires 75% approval with quorum constraints; stage 2 permits council action only on provable bugs with limited responses. [1] [2]
• When to transition stages: Moving from stage 0 to 1 and then to 2 depends on confidence in the proof system versus trust in the security council. Higher proof system reliability favors progressing to later stages. [3] [4]
• Model assumptions: Each council member has a 10% independent chance of failure; stage 0 uses 4-of-7 approval, stage 1 uses 6-of-8; stage 2 relies solely on the proof system with no council override. [5] [6]
• Failure probabilities: Stage 0 has a fixed 0.2728% failure chance due to council breakdown; stage 1 failure depends on combined proof system and council failures; stage 2 failure equals proof system failure rate alone. [7] [8]
• Optimal stage shifts: As proof system quality improves, the best security stage moves from 0 to 1, then 1 to 2; using stage 2 with a low-quality proof system is least secure. [9]
• Model limitations: Real-world council members may share correlated failures, and proof systems can be composites of multiple systems, affecting failure probabilities and council roles even in stage 2. [10] [11]
• Practical recommendations: Stage 0 is generally unjustified; launching at least at stage 1 is advised, with mechanisms like single-member withdrawal delays to handle emergencies. However, rushing to stage 2 without proof system maturity can be risky, highlighting the need for audits and maturity metrics. [12] [13]
Standard
AI-generated content may be incorrect


Copilot
Hello Calvin!
Here are some ideas to get you started.
Summarize this doc
Search and plugins
Preview
Manage web content preferences for Copilot's responses. Some Microsoft 365 data may be shared to fulfill your request.
Web content
Explain this document in three sentences
Focus
60%



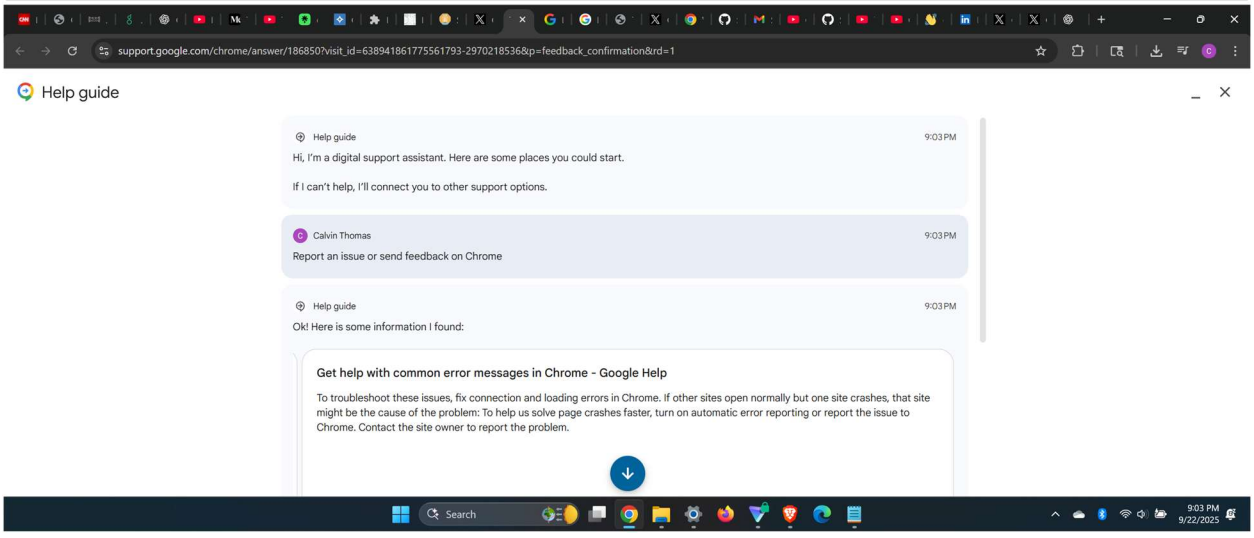
The math of when stage 1 and stage 2 make sense

2025 May 06See all posts

(L2) solution, achieves Stage 1 decentralization with a security council where at least 26% of members are independent, preventing unilateral control and aligning with research from the Ethereum Foundation showing distributed governance reduces single-point failures (Ethereum Research, 2023).

The post addresses misconceptions about L2 custody, noting that Base's design ensures users can withdraw funds even if the L2 shuts down, supported by smart contract logic on Ethereum L1, a mechanism validated by real-world tests like the Soneium censorship incident earlier in 2025.

The addendum challenges the oversimplified view of decentralization, highlighting that transitioning to Stage 2 (full onchain governance) depends on network maturity, with data from L2BEAT indicating only 15% of L2s have reached this level as of September 2025.



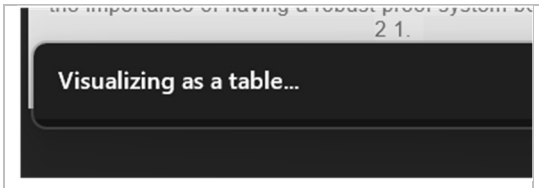
Expanded on from this earlier draft: https://x.com/VitalikButerin/status/1919263869308191017

The three "stages" of Ethereum rollup security can be described in terms of **when a security council is allowed to override trustless (ie. pure cryptographic or game-theoretic) components**:

- **Stage 0: security council has full control**. There may be a proof system (optimistic or ZK) running, but a security council can overturn it with a simple majority vote. Hence, the proof system is "advisory only".
- **Stage 1: security council can override with 75% (at least 6-of-8) approval**. A quorum-blocking subset (ie. >= 3) must be outside the primary organization. Hence, there is a high, but not impassable, barrier to overriding the proof system.
- **Stage 2: security council can only act in case of provable bugs**. Provable bugs could be eg. two redundant proof systems (eg. OP and ZK) disagreeing with each other. And if there are provable bugs, it can only choose between one of the proposed answers: it cannot answer arbitrarily.

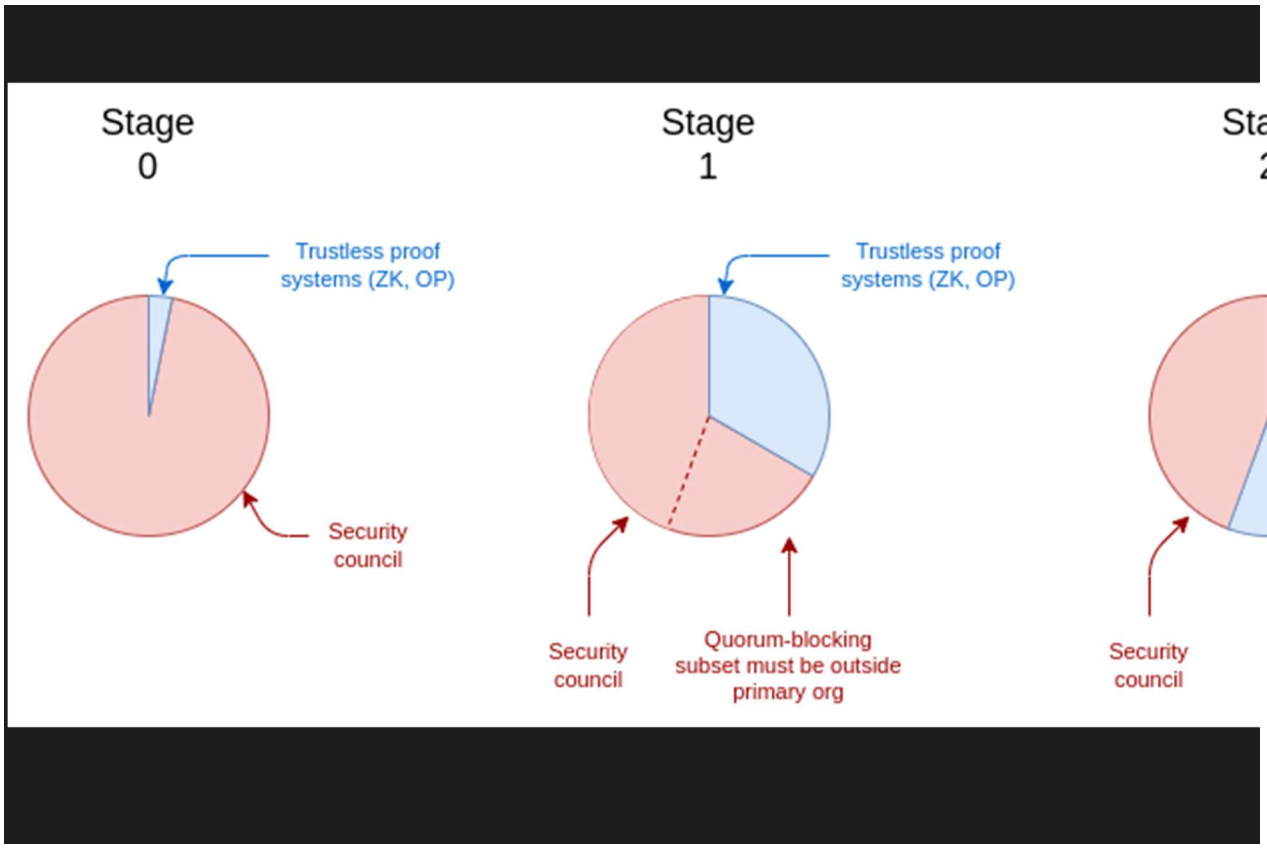
We can model this with a chart showing "what share of the vote" the security council has at each stage:

This document discusses the three stages of Ethereum rollup security and the conditions under which each stage is optimal. It provides a mathematical model to determine when it is best for an L2 to move from stage 0 to stage 1, and from stage 1 to stage 2, based on the probability of the proof system breaking and the security council's ability to override it 1. The document concludes that stage 0 is rarely justified and emphasizes the importance of having a robust proof system before moving to stage 2 1.

 **This document discusses the three stages of Ethereum rollup**	**Description**	**Optimal Condition**	**Note**  **s**

security and the conditions under which each stage is optimal. It provides a mathematical model to determine when it is best for an L2 to move from stage 0 to stage 1, and from stage 1 to stage 2, based on the probability of the proof system breaking and the security council's ability to override it . The document concludes that stage 0 is rarely justified and emphasizes the importance of having a robust proof system before moving to stage 2 .Stage			
Stage 0	Initial stage of Ethereum rollup security	Rarely justified	Should move to stage 1 when possible
Stage 1	Intermediate stage	Based on probability of proof system breaking and security cou	Mathematical model determines move to stage 2

		ncil's ability to override	
Stage 2	Advanced stage with robust proof system	Move when robust proof of system is available	Emphasized as important



One important question to ask is: **when is it optimal for an L2 to move from stage 0 to stage 1, and from stage 1 to stage 2?**

The only valid reason to not go to stage 2 immediately is that you do not fully trust the proof system - which is an understandable fear: it's a lot of code, and if the code if broken, then an attacker could potentially steal all of the users' assets. The more confidence you have in your proof system (or, conversely, *the less confidence you have in security councils*), the more you want to move towards the right.

It turns out that we can quantify this with a simplified mathematical model. First, let's **list the assumptions**:

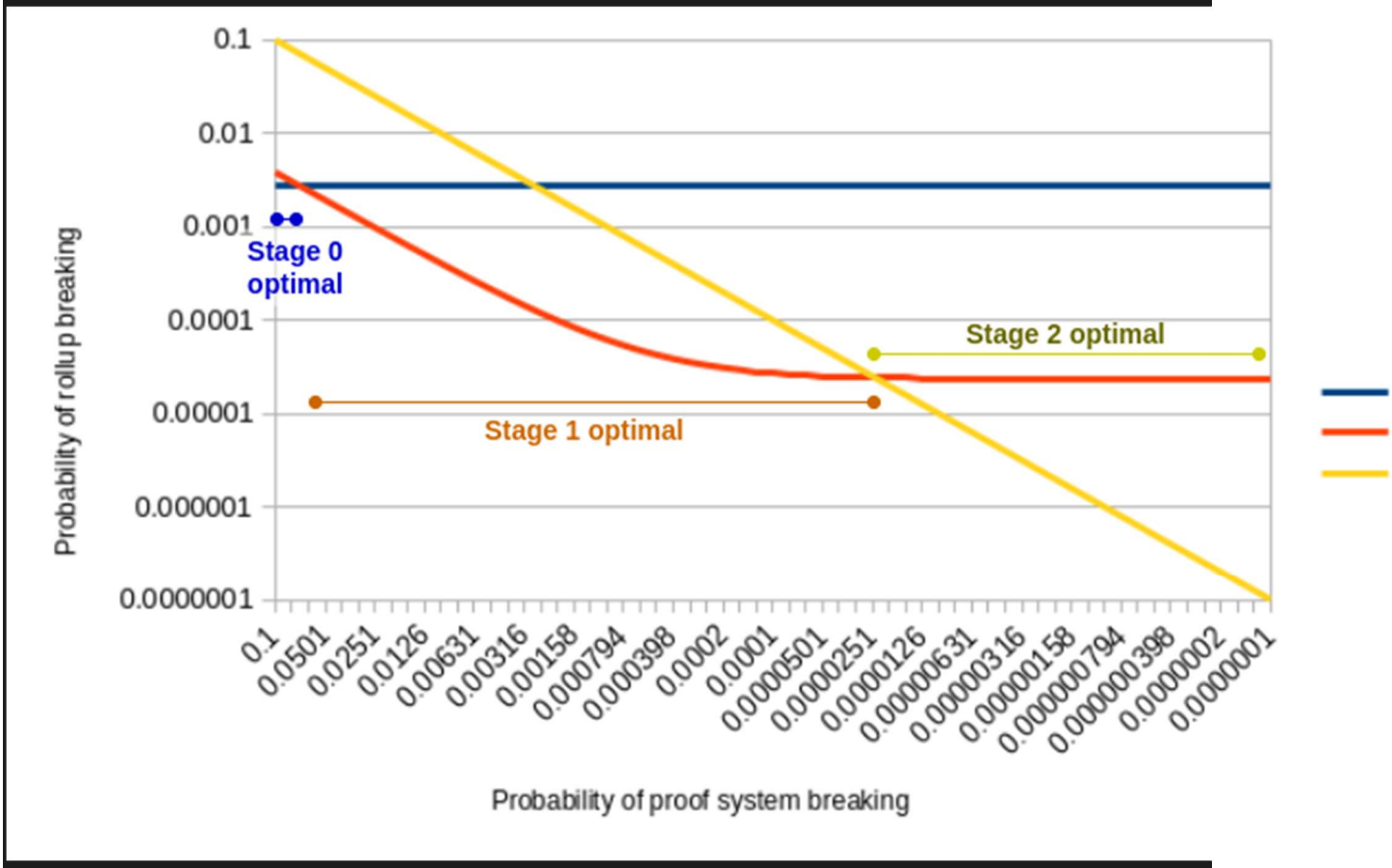
- Each security council member has an independent 10% chance of "breaking"
- We treat liveness failure [refusal to sign or keys inaccessible] and safety failure [signing a wrong thing or keys hacked] as equally likely. In fact, we just assume a single category of "broken" where a "broken" security council member both signs the wrong thing and fails to sign the right thing
- In stage 0, the security council is 4-of-7, in stage 1 it's is 6-of-8.
- We assume a single monolithic proof system (as opposed to a 2-of-3 design where the security council could break ties if the two disagree). Hence, in stage 2 the security council does not matter at all.

Given these assumptions, and given a particular probability of the proof system breaking, we want to minimize the probability of the L2 breaking.

We can do this with binomial distributions:

- If each security council member has an independent 10% chance of breaking, then the chance that at least 4 of 7 will break is Thus, a stage 0 rollup has a fixed 0.2728% chance of failing.
- A stage 1 rollup can fail if either the proof system fails and the security council gets >= 3 failures so it can't override (probability multiplied by the proof system failure rate), or if the security council gets 6+ failures and can force an incorrect answer by itself (fixed probability)
- The chance that a stage 2 rollup will break is just equal to the probability that the proof system fails

Here it is in graph form:



As conjectured, as proof system quality increases, the optimal stage shifts from stage 0 to stage 1, then stage 1 to stage 2. Doing stage 2 with a stage-0-quality proof system is worst of all.

Now, note that **the assumptions in the above simplified model are very imperfect**:

- In reality, **security council members are not independent**, and have "common mode failures": they could collude, or all get coerced or hacked the same way, etc. The requirement to have a quorum-blocking subset outside the primary organization is meant to mitigate this, but it is still far from perfect.

- The **proof system could itself be a combination of multiple independent systems** (this is what I advocate in https://ethereum-magicians.org/t/a-simple-l2-security-and-finalization-roadmap/23309...). In this case, (i) the probability of a proof system breaking could end up very low, and (ii) even in stage 2, security councils matter, as a matter of tiebreaking.

These two arguments both imply stage 1 and stage 2 are both even more attractive than the chart shows. **If you take the math seriously, stage 0 is pretty much never justified: you should launch at least straight into stage 1**. The main argument that I hear against is: if a critical bug happens, it may be too hard to get 6 of 8 security council members to sign fast enough to fix it. But there is an easy way around this: give any single security council member the permission to delay withdrawals by 1 or 2 weeks, giving everyone else enough time to act.

At the same time, however, **it is a mistake to jump to stage 2 *too* quickly, especially if work to move to stage 2 happens at the expense of work to harden the underlying proof system**. Ideally, data providers like l2beat should show proof system audits and maturity metrics (ideally of the proof system implementation, not the rollup as a whole, so we can reuse) along with the stage. 

It seems you've shared an image with the number "-741" displayed in red on a dark background, alongside a vertical line and some additional elements. This image appears to be unrelated to the original X post by Calvin Thomas (

@TestTempAI

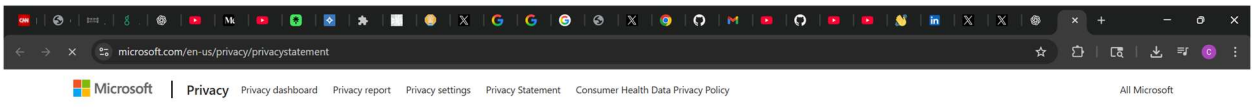
) that mentioned Vitalik Buterin and linked to an email inbox screenshot. Given the context of your initial query and the timestamp (01:53 AM BST, September 23, 2025), let's explore possible interpretations:

- Connection to the Original Post: The "-741" could be an artifact or a misinterpretation of the email inbox image from the original post. However, without a clear link (e.g., a timestamp, message ID, or context from the inbox), it's unlikely to directly relate to the Vitalik Buterin email unless it's a coded reference (e.g., a negative sentiment score, a time offset, or a blockchain-related metric like a block number or gas fee anomaly). The original post's focus was on a potential communication with Buterin, possibly about Ethereum's scaling solutions like sharding, which doesn't immediately suggest a numerical code like "-741."

- Independent Interpretation: The number "-741" could represent something standalone—perhaps a temperature reading (e.g., -74.1°C, an extreme value), a financial loss (e.g., -741 units of currency), or a technical metric (e.g., a negative offset in a system log). The red color and dark background might imply an alert or error state, common in dashboards or monitoring tools. Given the late hour (01:53 AM BST), it could even be a personal observation, like a thermometer reading during a cold night.
- Speculative Tie to Ethereum: Since your interest was in the Buterin-related post, let's hypothesize. In blockchain contexts, negative numbers can occasionally appear in debugging logs or as part of a smart contract's edge case (e.g., a subtraction error or a gas refund exceeding an expectation). However, Ethereum's sharding discussions, as per Buterin's past writings, focus on scalability (e.g., thousands of transactions per second) rather than negative indices. The "-741" might be a red herring unless it's part of a larger puzzle you're assembling.

Without more context, I can't definitively connect this image to the original post. My best guess is that it's either a separate data point you found intriguing or an unintended upload. If this "-741" is meant to relate to the Vitalik Buterin email (e.g., a message number, a timestamp offset from 00:53 UTC, or a coded reference), please provide additional details. Otherwise, it might be worth clarifying its source or intent.Would you like me to search for more information to contextualize "-741" in relation to Ethereum or other topics, or do you have more details to share about this image?

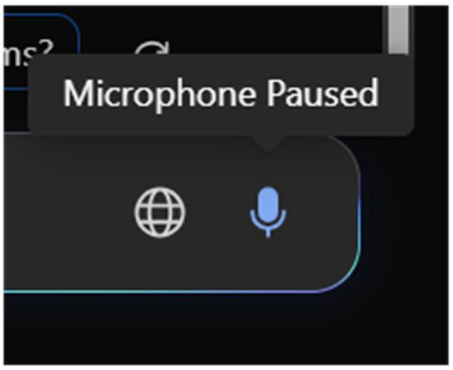
4 web pages



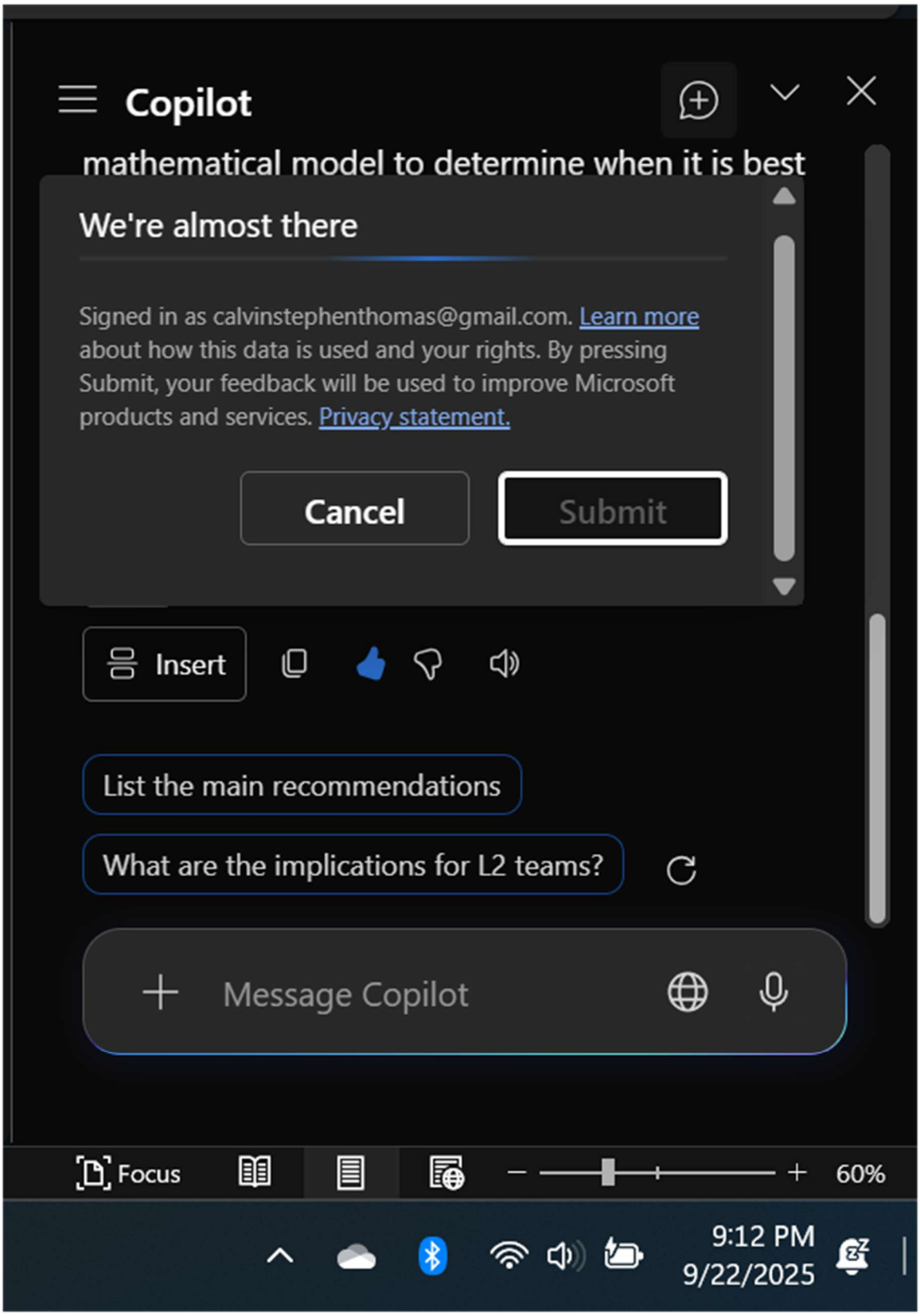
Microsoft Privacy Statement

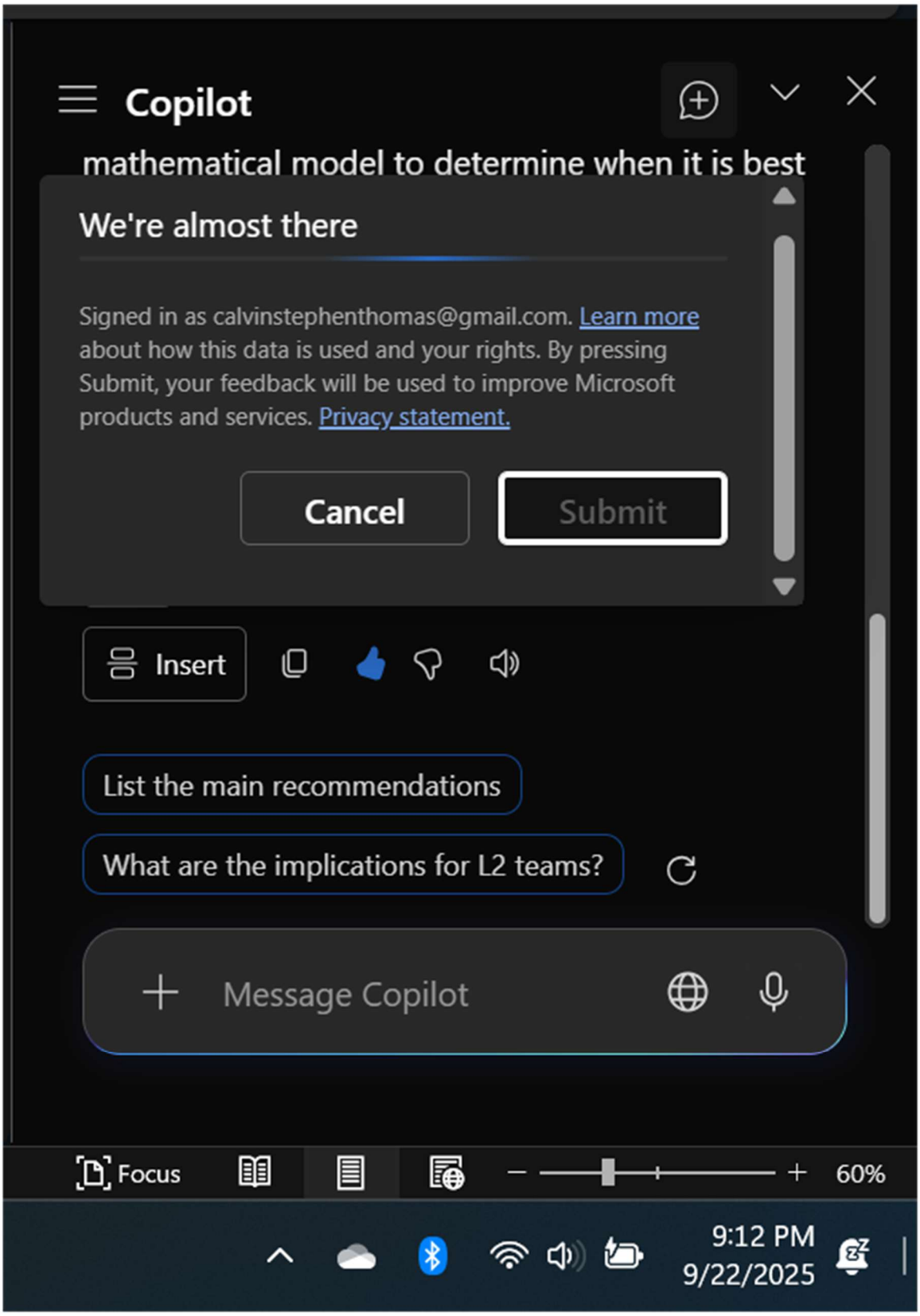
Last Updated: September 2025
What's new?

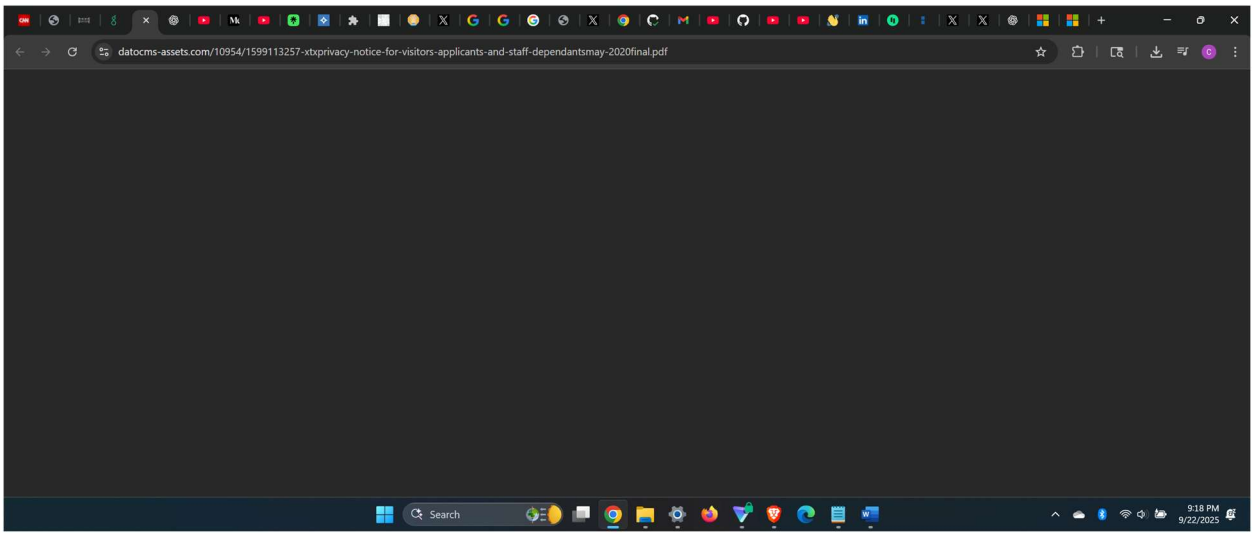
Print Expand All



closer than you cage the elephant lyric embed audio


Copilot
mathematical model to determine when it is best
We're almost there
Signed in as calvinstephenthomas@gmail.com. Learn more about how this data is used and your rights. By pressing Submit, your feedback will be used to improve Microsoft products and services. Privacy statement.
Cancel
Submit
Insert
List the main recommendations
What are the implications for L2 teams?
Message Copilot
Focus
60%
9:12 PM
9/22/2025


Copilot
mathematical model to determine when it is best
We're almost there
Signed in as calvinstephenthomas@gmail.com. Learn more about how this data is used and your rights. By pressing Submit, your feedback will be used to improve Microsoft products and services. Privacy statement.
Cancel
Submit
Insert
List the main recommendations
What are the implications for L2 teams?
Message Copilot
Focus
60%
9:12 PM
9/22/2025


datocms-assets.com/10954/1599113257-xtxprivacy-notice-for-visitors-applicants-and-staff-dependantsmay-2020final.pdf
Search
9:18 PM
9/22/2025