**A screenshot of a black screen

AI-generated content may be incorrect.A black background with white text

AI-generated content may be incorrect.A screenshot of a social media post

AI-generated content may be incorrect.  
The math of when stage 1 and stage 2 make sense**

2025 May 06[See all posts](https://vitalik.eth.limo/index.html)A black and white text on a red background

AI-generated content may be incorrect.

*Expanded on from this earlier draft:*[*https://x.com/VitalikButerin/status/1919263869308191017*](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:

A diagram of a pie chart

AI-generated content may be incorrect.

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](https://en.wikipedia.org/wiki/Binomial_distribution):

* 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:

A diagram of a graph

AI-generated content may be incorrect.

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](https://www.sciencedirect.com/topics/engineering/common-mode-failure)": 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](https://l2beat.com/scaling/summary) 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