Hedera Hashgraph's Public Ledger & Governance Framework | Leemon Baird

March 12, 2018

You know the consensus. You know that you know the consensus. You know that everyone else is going to agree with your consensus. Guaranteed mathematically – that's where the byzantine fault tolerance purely asynchronous all comes in. You do this with zero communication. You get it for free. In a fraction of a second. That's hashgraph. — Leemon Baird

INTRODUCTION

Leemon Baird is the Co-founder and CTO of Swirlds Inc. With over 20 years of technology and startup experience, he has held positions as a Professor of Computer Science at the Air Force Academy, Adjunct Professor at multiple other prestigious universities, and as a senior scientist in several labs. He has been the co-founder of several startups, including two identity-related starts-ups with successful exits. He received his Ph.D. in Computer Science from Carnegie Mellon University faster than any student in school history (2 years, 9 months), has multiple patents, and over 100 publications in peer-reviewed journals on computer security, machine learning, and mathematics. He regularly keynotes on these topics at conferences.

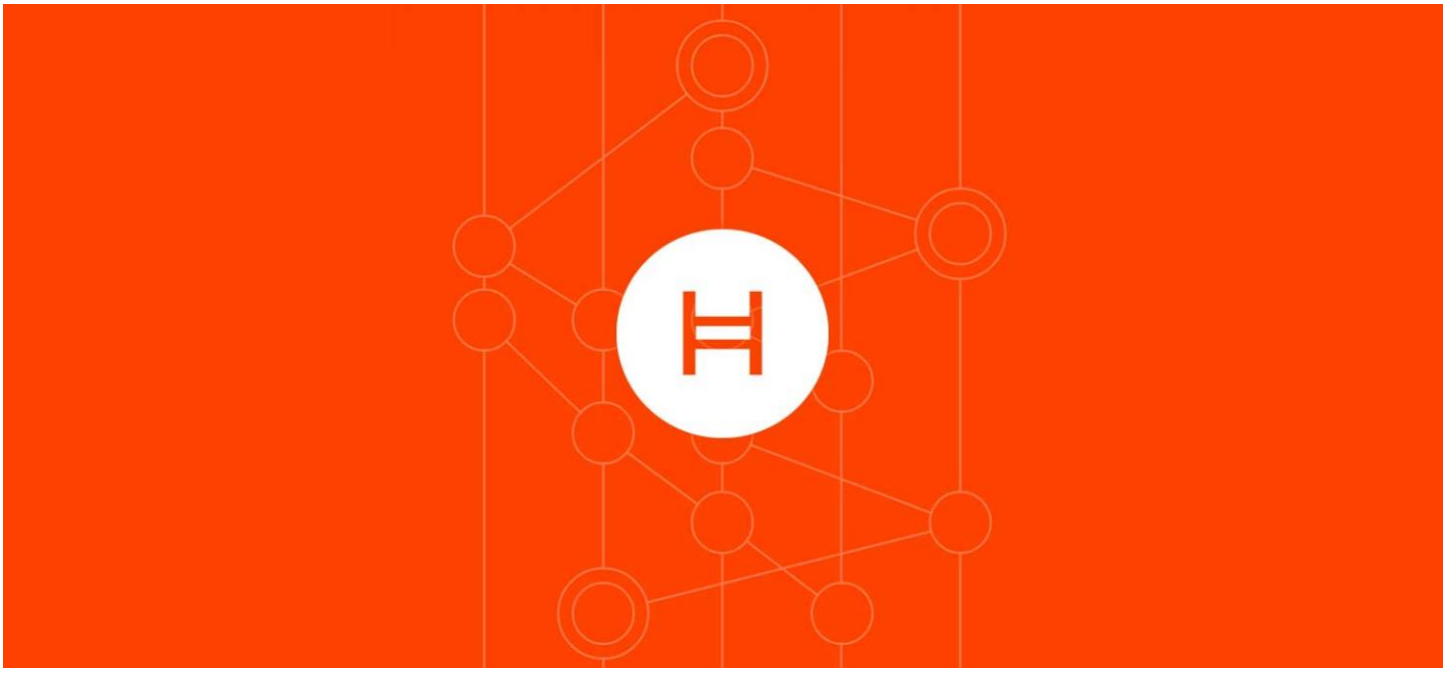
WHY DO I CARE?

DISCLAIMER ⚠ I'M AN INVESTOR IN HEDERA

From the moment I first read the Swirlds white paper detailing the Hashgraph consensus protocol in mid-September of 2017, the most important question I wanted to ask Leemon was the only one he was not prepared to answer: "Are you releasing a public ledger using hashgraph, and if so, how are you prepared to do it?"

The opportunity to become a seed investor in Hedera had never been discussed or presented to me prior to the very end of 2017. My coverage of Hashgraph, beginning with my September 2017 interview with Leemon Baird, my October 2017 panel at the Assemblage, and my December 2017 follow-up, on-camera interview with Leemon, all happened *before* I became an investor in the company. My enthusiasm for the project was 100% organic, and it was aided, in no small part, by the enormous amount of time I spent alone with Leemon. His passion and love for his work is infectious. Like many who came before me, I caught the bug.

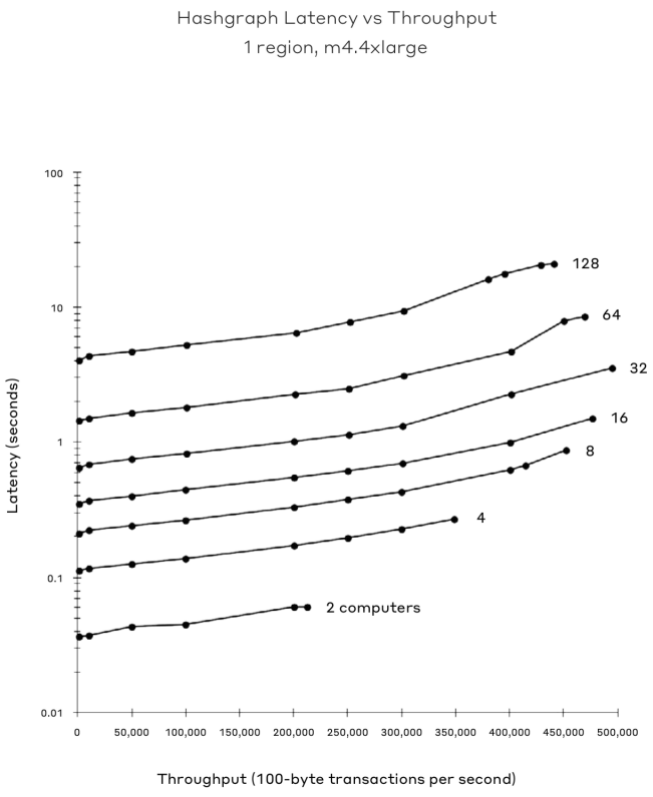
I've been waiting for this moment to speak with Dr. Baird publicly about what may one day be seen as the most revolutionary technology since the commercialization of the Internet in the early 1990s. I want audiences (developers, speculators, entrepreneurs, business executives, government officials, & regulators) to appreciate what he has invented and what the implications are for their businesses, economies, countries, and societies.

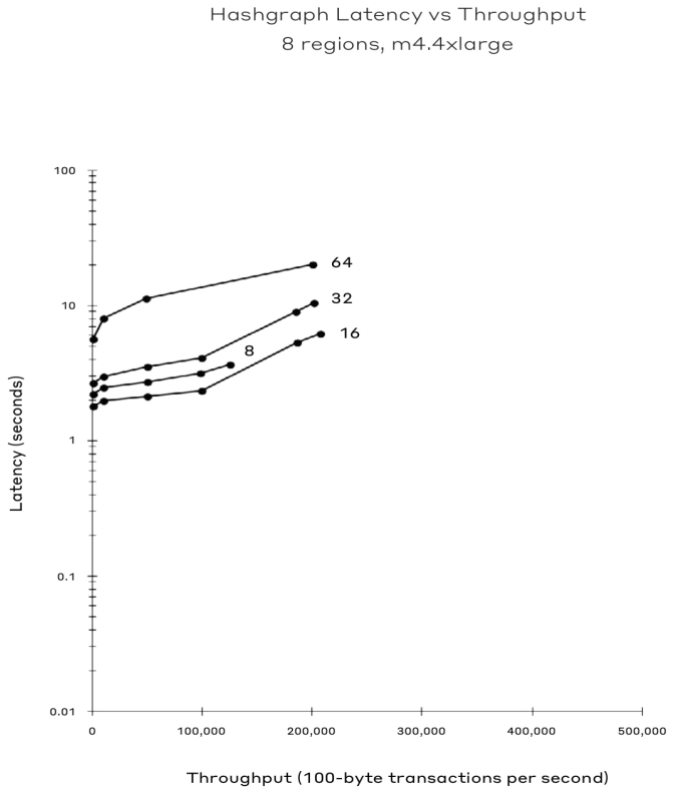


I began five years ago working on the math of it and I kept convincing myself it's impossible. You can't get the strong security guarantees without a voting system, but a voting system is too slow. And, if you try a hybrid system then you get all of the vulnerabilities again – because of the leader that's mixed in. I just kept convincing myself it was impossible, but it kept gnawing at me for years – I couldn't stop. Eventually, I realized you can just add a couple of bytes to each message and suddenly you know the entire history – then you can do virtual voting. But, it was a while before I figured it out. – Leemon Baird

Third Time's the Charm…

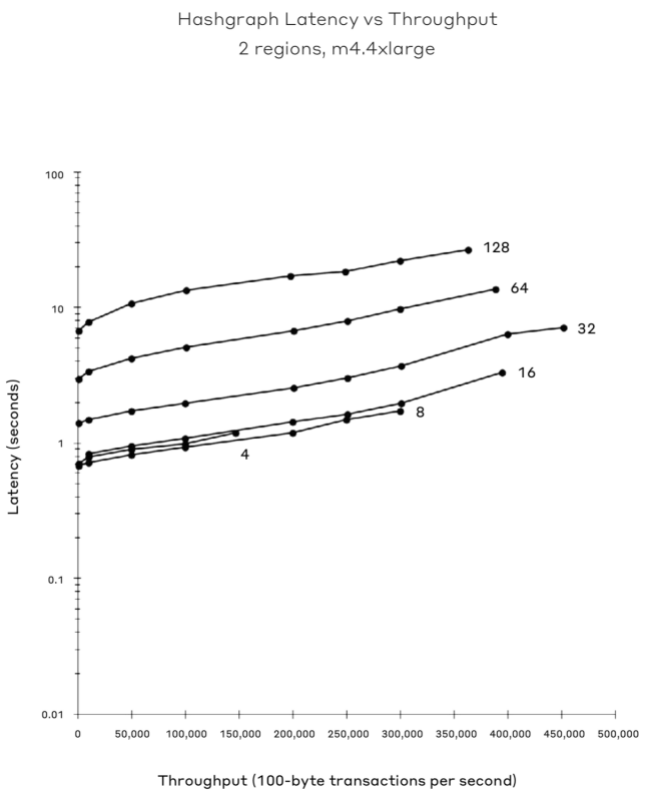
1. **Vision —** In your vision statement, you emphasize being open, fast, fair, and stable, as essential qualities for the successful deployment of a public ledger. I want to go through each of these. Let's start with openness. Hedera is what you call "open review." What is the distinction between open source, which is what these blockchain protocols use, and open review? What is the reason that you have chosen this model of software development? ***the community is technically able to copy the entire source code, but they cannot deploy it as the legitimate version of Hedera (talk later about disincentivizing forks)***
2. **Performance (fast) —** We will get back into open review as a matter of our conversation on stability, and I also want to ask you towards the end, what your goal is – what you hope to achieve with your approach to governance – but let's move to performance – how your ledger performs in some of the areas that are most crucial to the building of distributed applications. We're going to get into security later, which is arguably the most important part of the conversation, but let's talk about performance. In our prior conversations and in our panel with Mance Harmon and other members of the founding team, it was Hashgraph's low latency and its high throughput that I think most excited people. At the time, we were comparing apples to oranges, because Hashgraph was only operating in permissioned environments and some of the other blockchain based protocols are non-permissioned. In the case of Hedera, this is a fully distributed public ledger that relies on proxy staking and database sharding in order to replicate those low levels of latency and high throughput in a trustless environment with as little centralization as possible. You have conducted tests with as many as 128 nodes operating across a variety geographies.





The results show a moderate tradeoff between latency (time to consensus) and throughput (TPS/transaction volume) – in other words, for a fixed number of computers operating in a single shard, increasing the number of transactions you can process in any given second also moderately increases the time to consensus. Where the tradeoff is more pronounced is – not surprisingly – in latency with respect to the number of nodes operating on the network, as well as the distribution of those nodes in space – i.e. the more nodes, the longer it takes to reach network wide consensus within a shard and the further apart the nodes in any given shard, the longer it also takes to reach consensus.

a. How do you feel these results will hold up for the public ledger?
b. Will Hedera's software attempt to maximize throughput and minimize latency by processing some types of transactions on shards operating in fewer regions and/or with fewer nodes, depending on what the particular application or use cases are? (ex: latency is not as important when transferring title of your home, but it is when processing a CC transaction). *** Will get into security implications of such a solution later ***
c. Your results do not account for the time it takes to process transactions. For example, if a great deal of processing power is needed to verify hundreds of thousands of digital signatures per second, that could slow down the network, could it not? How would processing time affect performance? Do you foresee hardware implementations that could significantly mitigate this?
d. What about the fact that not every node has access to the same amount of bandwidth for communicating information across the network. How would this affect performance?
e. What types of applications do you believe can be reliably built on top of Hedera within the first year, and is it even possible to give an answer to that given the fact that you can't ordain how the size of the network?



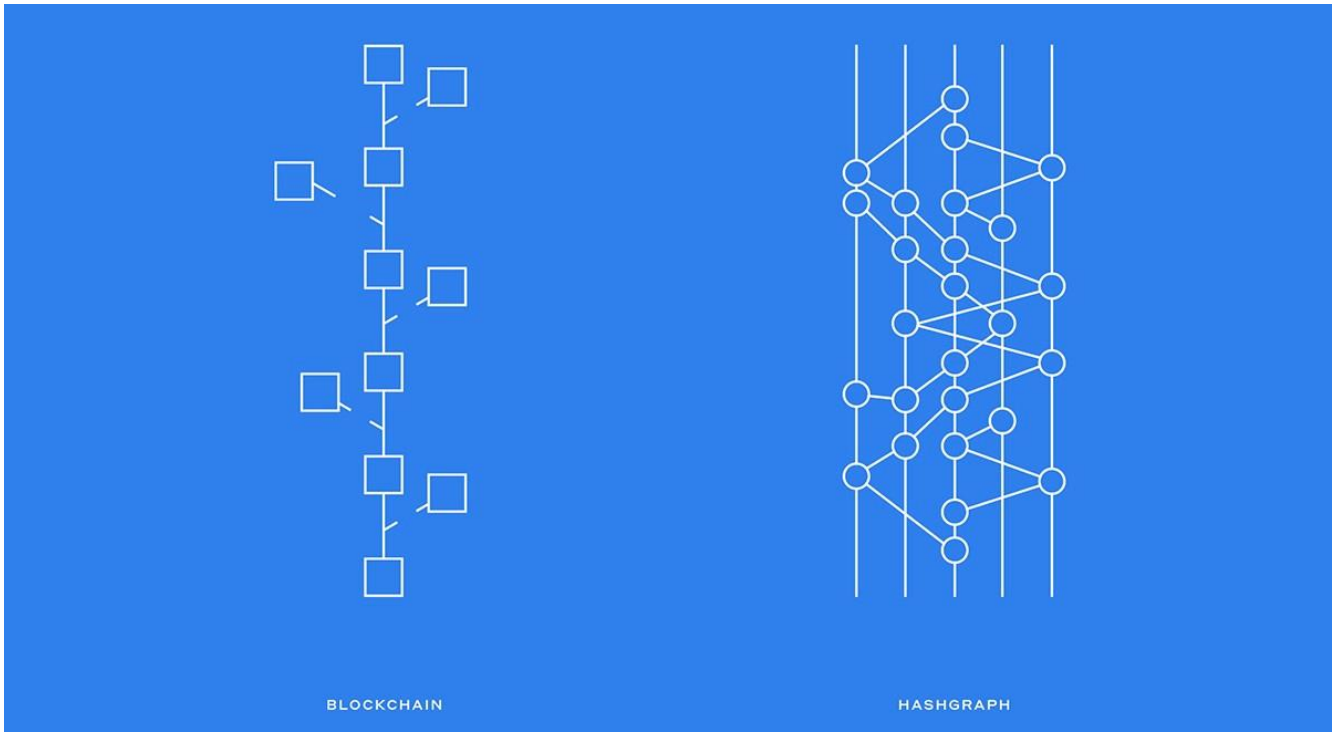
3. **Fairness (fair)** — Let's talk about fairness, which is entirely a function of the consensus layer. Your consensus protocol sets you apart among your competitors in this domain. Currently, how is fairness done for non-permissioned blockchain databases? (leaders are elected/win lottery to compile the next block in the chain) How does this differ from Hashgraph? Why is this important? How does consensus timestamping work? How does the ability to reach consensus on the order of transactions impact the types of applications that can be built on Hedera, which would otherwise not work on ledgers that don't have fairness in ordering?
4. **Governance Model (stable)** — As stated earlier, being "stable" features prominently in your vision statement as a primary goal that you are committed to for Hedera. We are going to get into technical controls and regulatory compliance in a bit, but how does "stability" figure into your designs for governance? You have mentioned before that public platforms are "at risk of devolving into chaos." Can you elaborate on what you mean?
    a. **Council Governance** — Can you explain the thinking behind the architecture of your governance structure? Where did the number of 39 governing council members come from? Can you tell us who they are/why not? What are the criterion for choosing who the members will be? You make

a point to emphasize that the council comprise of broad expertise across a multitude of industries and geographies, as well as legal and regulatory expertise. Why? What are the term limits and is there a constitution for governance? What is the role of the council members? What is the role of the board?

b. **Consensus Model** — This "concerns the process by which the nodes reach a consensus on the order of transactions in the platform. The model is designed to *prevent consolidation of power over consensus*. It prevents collusion by a few to attack the system such as by counterfeiting the cryptocurrency, modifying the ledger inappropriately, or influencing the consensus order of transactions." Is consensus model governance another way of stating that the consensus protocol and the larger architecture of the public ledger feature prominently in any conversations about governance and fairness in network access?

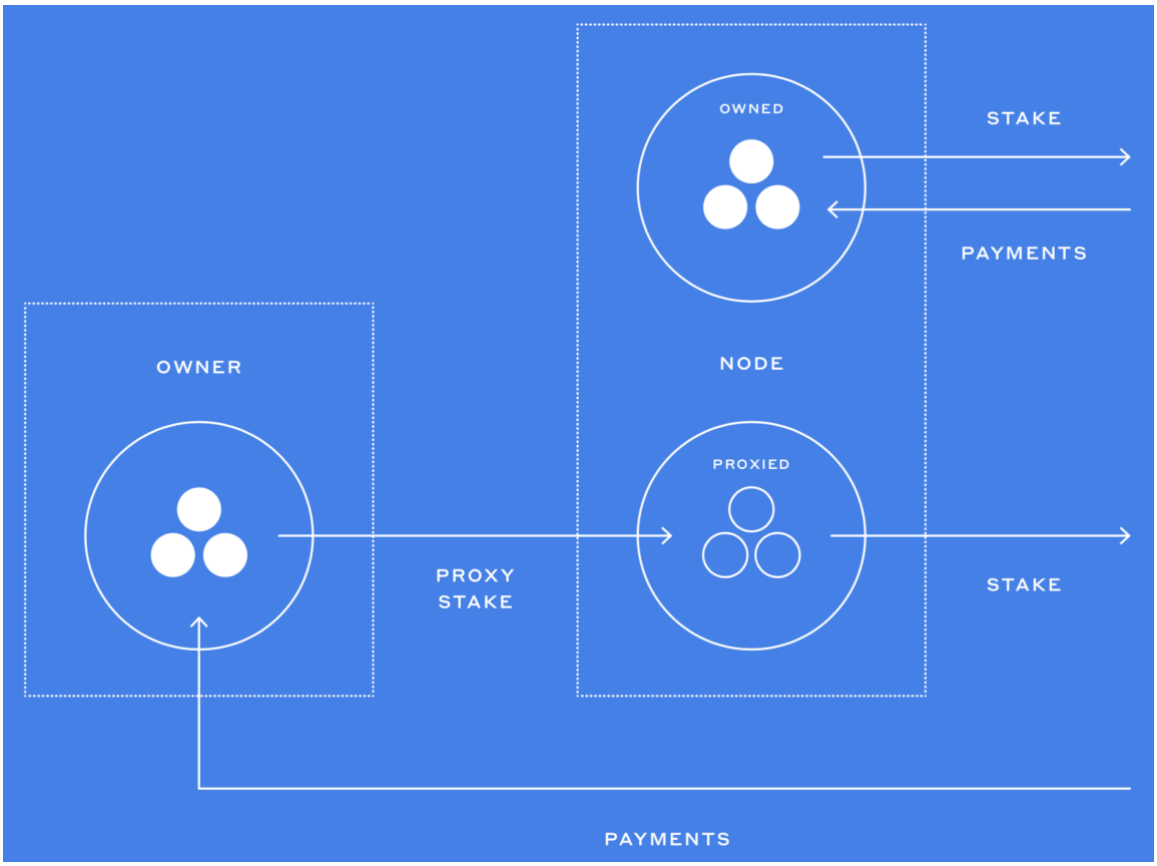
5. **Controls (*no forking*)** — Hedera features both technical, as well as legal controls, meant to help maintain the integrity and stability of the ledger. In terms of technical controls,
   a. first, the Swirlds technology ensures that software clients validate the pedigree of the Hedera hashgraph prior to use through a shared state mechanism.
   b. second, you make it possible for the Hedera governing body to not only specify the software changes to be made to network nodes, but also to ensure precisely when those changes are adopted, and to guarantee that they are.

   Can you explain how this is done? People are technically able to copy and reproduce the ledger, correct? Do you foresee any challenges in validating the software run by untrusted participants (nodes) on the network? What does it mean that software updates will be "automatically" updated? Does this mean that nodes have to change their settings from allowing for automatic updates to making them manual? Are there legitimate reasons for a node needing time to make an update? Can you explain how forking is pre-conditionally disincentivized?
6. **Regulatory Compliance** — "The Hedera technical framework includes an Opt-In Escrow Identity mechanism that gives to users a choice to bind verified identities to otherwise anonymous cryptocurrency accounts, and thereby provide to governments with the oversight necessary to ensure regulatory compliance." You state elsewhere in your paper that you soon expect governments to "require comparable visibility and oversight into public ledger financial transactions that they have now



for traditional banking and financial applications." Do you think that governments have largely taken a hands off approach, at least in the West, because current DLT's don't actually pose a meaningful threat to the existing industries, like the banking industry? How are you accommodating for this outcome, and do you expect to have to make further changes to your software in order to accommodate further?

7. **Security** — This is going to be the most complicated part of our conversation. Security, in a permissioned network, though perhaps not straightforward, makes securing a public ledger look like a walk in the park by comparison.
    a. **Proxy Staking** — Can you explain what proxy staking is and how/why it is being used in Hedera? You do not require bonding. Can you explain the difference between your model and a PoS or DPoS network that requires bonding (CD or Time Deposit) vs. one that does not require bonding (Checking or Demand Deposit). You do not punish nodes, stake weighs the votes of the node in virtual voting, and nodes receive payments proportional to their stake.
        i. **Threat Scenario (economies of scale)** — Although proxy staking does not exhibit the same economies of scale that have led to centralization in Bitcoin, where a few mining pools control a majority of the network hash power, there still do seem to be economies to scale, particularly if specialty hardware and very robust broadband connections can improve the performance of shards in which such nodes participate. Are there any risks you foresee that could lead to centralization of proxy stakes, which could in turn lead to the creation of shards with a few nodes controlling 1/3 of the staking power? Wouldn't that make the entire shard more vulnerable to collusion or even DDoS?



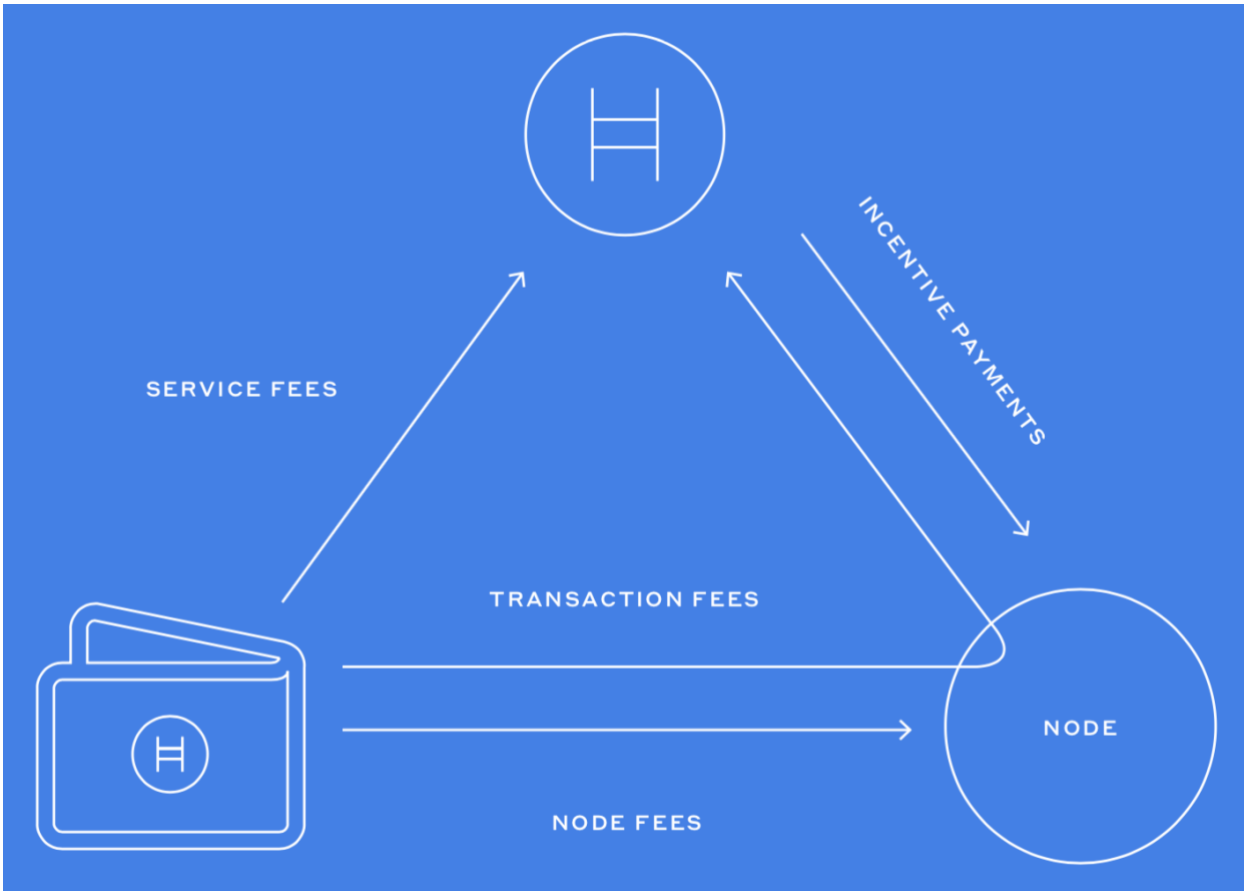
        ii. **Threat Scenario (follow the fat nodes)** — What about an attacker sniffing out (playing the equivalent of follow the leader) the top 100 or 1000 nodes in the network by amount of stake proxied and attacking them whenever they join a shard or whenever they join up with other nodes within a shard to comprise more than 1/3 of that shard's staking power?

Is there anything to prevent a single node from joining a shard where it would have 1/3 staking power? Is there a limit on how many nodes can stake 1/3 of the shard?

iii. **Threat Scenario 3 (smaller shards)** — In permissioned networks Hashgraph completely eliminates the possibility of DDoS attacks, because there is no single leader or validator – the entire community is seeing and voting on all the transactions almost simultaneously. But what about in cases where we need very high levels of throughput AND low levels of latency (like a massively multiplayer online role playing game or a stock market). Might that not require shards comprised of a smaller number of nodes making the network vulnerable to a 1/3 attack on all its members? Is there a threshold tolerance for DDoS (number of independent nodes required to reasonably prevent DDoS)?

iv. **Mitigation, Deterrence & Defense** — Will your software be able to identify malicious actors over some number of rounds of voting and purge them from the network? If so, how does this factor into your security?

8. **Sharding** — Can you explain sharding? What is it? What are the challenges you have faced in designing your particular implementation with Hedera? What are some meaningful differences between the challenges one would face in trying to shard a blockchain database vs. one using Hashgraph for consensus? How do you manage communication and consensus for applications that are using multiple shards concurrently?
9. **Incentives** — There are three, distinct fees that are generated by using Hedera. Can you walk us through the different types of fees – Node Fee, Transaction Fee, and Service Fee – and help us understand how to think about these in the context of the network's operation.
    a. **Node Fee** — A client can use the services of the platform by contacting a node, which will submit transactions on the client's behalf. For example, if a client wants to transfer cryptocurrency from their account to another, they will contact a node, and give it the signed transaction. The node

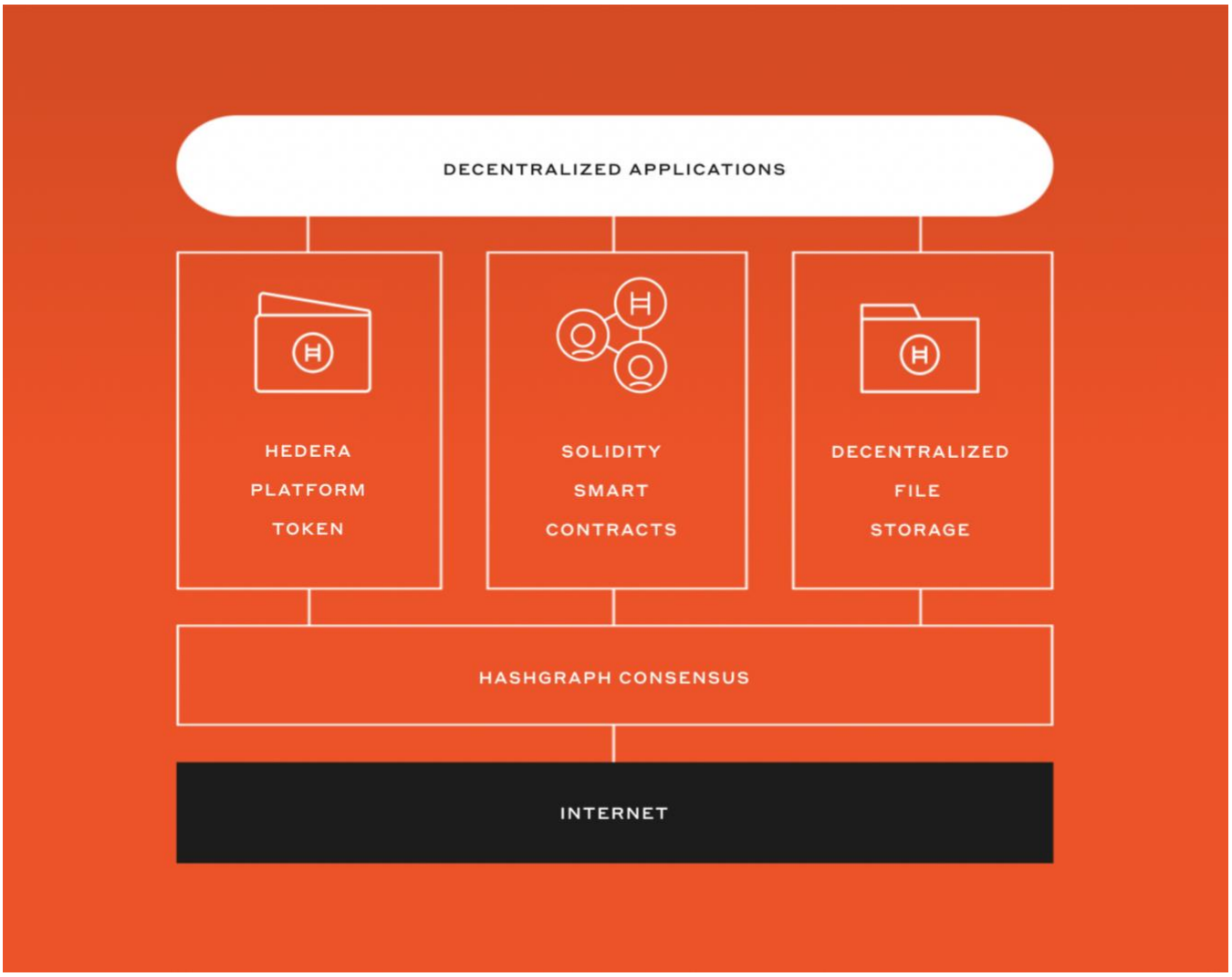


will then put that transaction into the next event it creates, and gossip it out to the network so that it can be entered into consensus. The client reimburses the node for this effort by giving it a node fee. This fee is negotiated between client and node, and can be set by market forces as nodes set their fees. This is the only fee that is not set by Hedera.

b. **Service Fee** — A client will pay a fee for any Hedera service. For example, if a client submits a transaction to store a file in ledger, the fee will be calculated according to a schedule determined by Hedera. This is calculated as a fee per file plus an amount per byte per second that the file will be stored. A single transaction both requests the service and authorizes paying for it. If the client's account has insufficient funds at the point the transaction takes effect in the consensus order, then the client is not charged, and the file is not stored. But if there are sufficient funds, then simultaneously the client is charged and the file is stored.

c. **Transaction Fee** — there is a fee for each transaction handled by the network, to cover the cost to nodes of gossiping it, temporarily storing it in memory, and calculating the consensus on the event containing it. The fee is calculated as an amount per transaction plus an amount per byte within the transaction. When a node includes a transaction in an event that it creates the node will be charged the transaction fee when consensus is reached on that transaction. If the transaction was initiated by a client, that client will compensate the node for that transaction fee the node paid.

10. **Money Supply/Currency Value** — How will Hedera manage the money supply? Will part of the transaction and services fees paid to Hedera be kept in reserve in order to protect the value of the



currency in the event of an unexpected shock/drop in market cap that would make Hedera more vulnerable to a Sybil attack? (similar to FOREX reserves) Where will the value for the currency come from besides speculation? How might fee amounts be used as a mechanism for supporting the currency's value or for making the services more affordable (like fed funds rate)? How will money velocity in the network affect the value of the currency? (MV = PQ)

11. **Services Layer** — Normally, we think of services like file storage and smart contracts as being distributed applications built on top of the stack by third parties. In this case, are you launching the network with these applications built in? Is this analogous to Microsoft launching with MS Office?