**Hashgraph** is a [distributed ledger technology](https://en.wikipedia.org/wiki/Distributed_ledger) that has been described as an alternative to [blockchains](https://en.wikipedia.org/wiki/Blockchain). The hashgraph technology is currently [patented](https://en.wikipedia.org/wiki/Patent), and the only authorized ledger is Hedera Hashgraph. The native [cryptocurrency](https://en.wikipedia.org/wiki/Cryptocurrency) of the Hedera Hashgraph system is HBAR.

Unlike blockchains, hashgraphs do not bundle data into blocks or use miners to validate transactions. Instead, hashgraphs use a "gossip about gossip" protocol where the individual nodes on the network "gossip" about transactions to create [directed acyclic graphs](https://en.wikipedia.org/wiki/Directed_acyclic_graph) that time-sequence transactions.[[1]](https://en.wikipedia.org/wiki/Hashgraph#cite_note-1) Each "gossip" message contains one or more transactions plus a [timestamp](https://en.wikipedia.org/wiki/Timestamp), a [digital signature](https://en.wikipedia.org/wiki/Digital_signature), and [cryptographic hashes](https://en.wikipedia.org/wiki/Cryptographic_hash_function) of two earlier events. This makes Hashgraph form an asynchronous [Byzantine Fault](https://en.wikipedia.org/wiki/Byzantine_fault)-Tolerant (aBFT) [consensus algorithm](https://en.wikipedia.org/wiki/Consensus_(computer_science))

.

Hashgraph has been described as a continuation or successor to the [blockchain](https://en.wikipedia.org/wiki/Blockchain) concept, which provides increased speed, fairness, low cost, and security constraints.[[5]](https://en.wikipedia.org/wiki/Hashgraph#cite_note-5) The Hedera white paper co-authored by Baird explained that "at the end of each round, each node calculates the shared state after processing all transactions that were received in that round and before," and it "digitally signs a hash of that shared state, puts it in a transaction, and gossips it out to the community

<https://101blockchains.com/hashgraph/>

<https://github.com/hashgraph/hedera-sdk-java>

The programming language used by the Hashgraph includes **LISP and Java**. The core is written in these two programming languages. However, it is inclined towards JVM language such as Scala, Java, etc. with the use of the SDK that the Hashgraph offers.

**Three key features of Hashraph**

Hashgraph has three key features that make it an excellent choice for different projects. In the white paper, it describes itself as secure, fair, and fast. To understand each of these features, let’s discuss them below.

**Secure:**The consensus algorithm does offer a secure way of handling transactions and ensure that an event is covered correctly. The order is what matters in Hashgraph, and the Hashgraph make sure that no malicious actor can fiddle with the data accuracy or the order in which the events are connected with each other. This way, it protects the network from both double spending problem as well as a 51% attack. It also effectively utilizes the resistant [hash function and digital signatures](https://101blockchains.com/hashing-and-digital-signature-in-blockchain/). Once a transaction is committed, it cannot be reversed or changed. After all, it uses ABFT(Asynchronous Byzantine Fault Tolerant).

**Fair:** The fairness concept surrounds the idea of being fair to all the nodes in the network. It defines fairness by stating that an attacker will not be able to learn which two new transactions will make it to the consensus order. However, it is not clear how it can provide fairness to the Hashgraph. Apart from the white paper definition, the Hashgraph team also clarified through social media platforms that fairness works well if the majority of nodes know about the transaction. This can lead to a problem if an attacker gets hold of the 2/3rd of the participants. He can easily re-order the events without impacting on the fairness of the network. There is also no requirement of Hashgraph mining for the nodes.

**Fast:**[Gossip methods](http://www.cs.cornell.edu/home/rvr/presentations/gossip/ppframe.htm) are considered fairly fast. This is true in case of the Hashgraph’s gossip protocol. The events spread across the network fast considering that it is all about “gossip-about-gossip.” This also means that less information required to be propagated over time. Hashgraph also utilizes virtual voting, which makes it more efficient. But if we take into consideration that each node require entire Hashgraph, the size of the inbound should increase over time. For now, we do not know how it will impact the performance of the network. Theoretically, Hashgraph TPS can reach 5,00,000

### **Hedra Hashgraph Architecture**

Hedra Hashgraph architecture is a three-layered architecture. It consists of the Internet Layer(Bottom), Hashgraph Consensus Layer(Middle), and Services Layer(Top). Let’s discuss each layer briefly.

* Internet Layer: The layer takes care of the communication between computers on the internet. It deploys TCP/IP connections with TLS encryption.
* Hashgraph Consensus Layer(Middle): The middle layer contains the nodes which participate in the network. These [nodes](https://101blockchains.com/blockchain-nodes/) take part in the consensus method using the Hashgraph [consensus algorithm](https://101blockchains.com/consensus-algorithms-blockchain/) and gossip protocol.
* Services Layer: The topmost layer has their its own subgroups – File Storage, [Cryptocurrency](https://101blockchains.com/what-is-a-cryptocurrency/), and Hashgraph [Smart Contracts](https://101blockchains.com/smart-contracts/).

The nodes earn the cryptocurrency for taking participating in the network. It is a native currency and ensures that the users get their incentive for participating.

The file storage, on the other hand, is Merkle-based. Moreover, if you are a developer, then you can also use [Solidity](https://101blockchains.com/solidity-tutorial/) as it is supported by the Hedra. Lastly, it offers smart contract support on the top of the network — giving you the ability to create scalable dApps