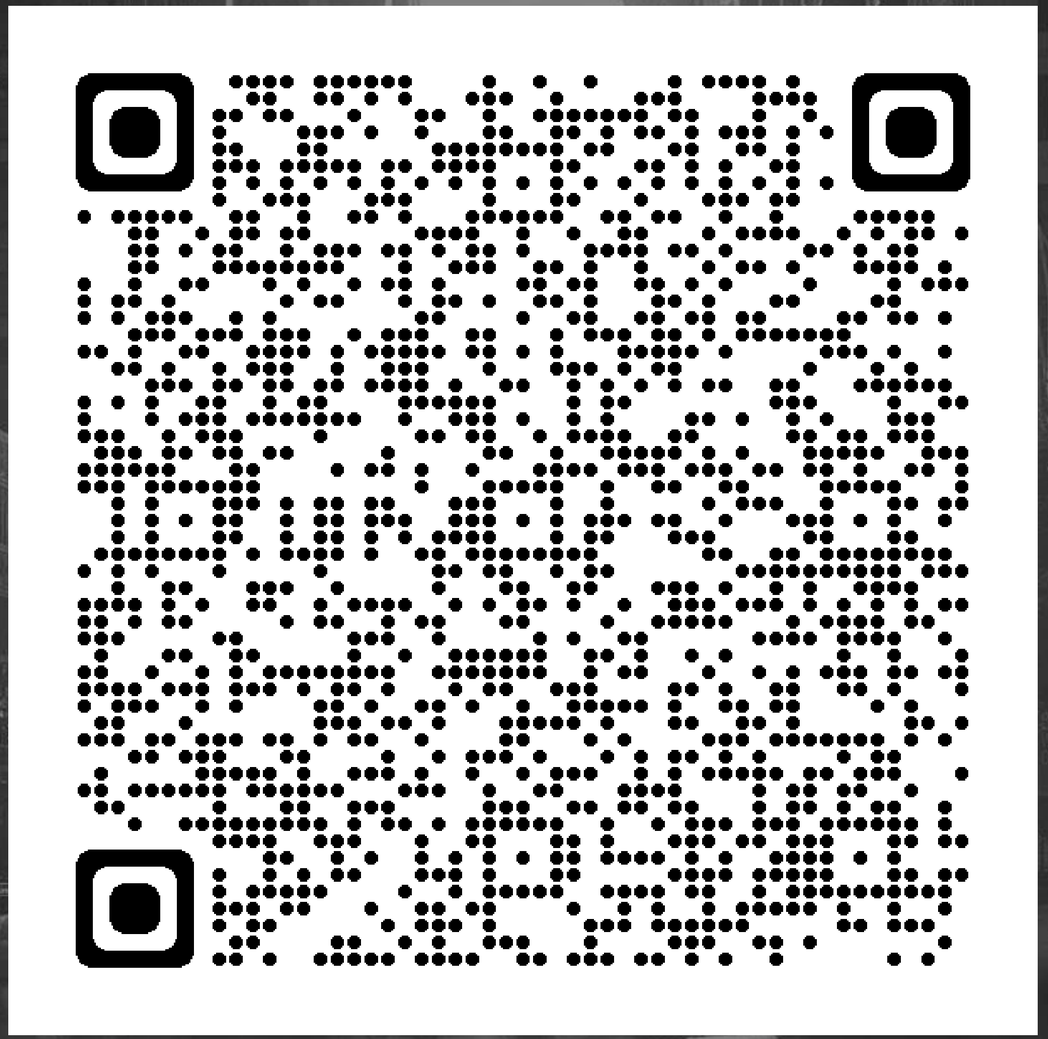
GET THE SLIDES NOW ➔



Build Web3 Applications on a Next Generation “Blockchain”

Introduction to Hedera

Ed Marquez – Hedera

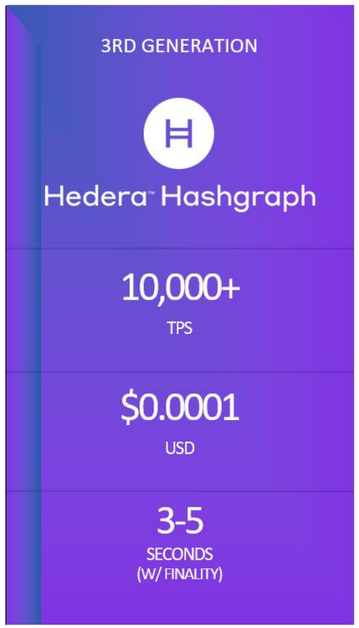
Viv Diwakar – The HBAR Foundation

/ed-marquez | /vivdiwakar

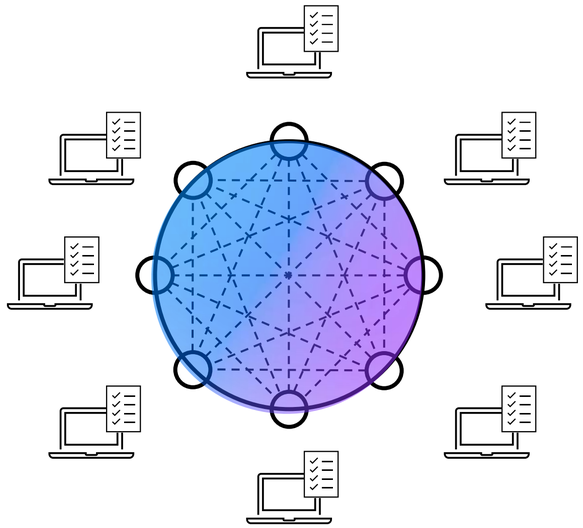
@ed_marquez | @HBAR_foundation

hedera.com

In this session, you will learn how the **Hedera network functions**, how you can **start development**, and **where to learn more**



Introduction



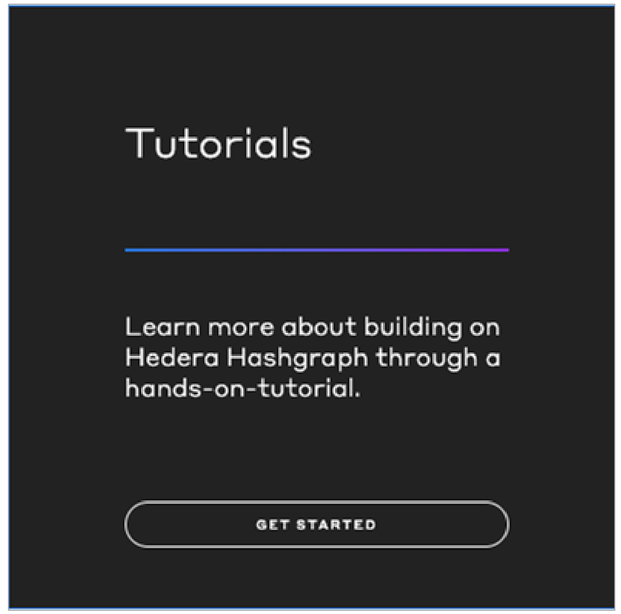
Understand
the Hedera Network

CONSENSUS SERVICE

TOKEN SERVICE

SMART CONTRACTS

Start Developing on
Hedera



Get Resources and
Learn More

hedera.com

The web evolves continuously and now it is becoming more decentralized

Pre-1970

Analog Era

Paper documentation

1970

Digital Era

Mainframes & computers

- Speed

2000

Internet Era

Interconnected computers

- Speed
- Access & collaboration

2009

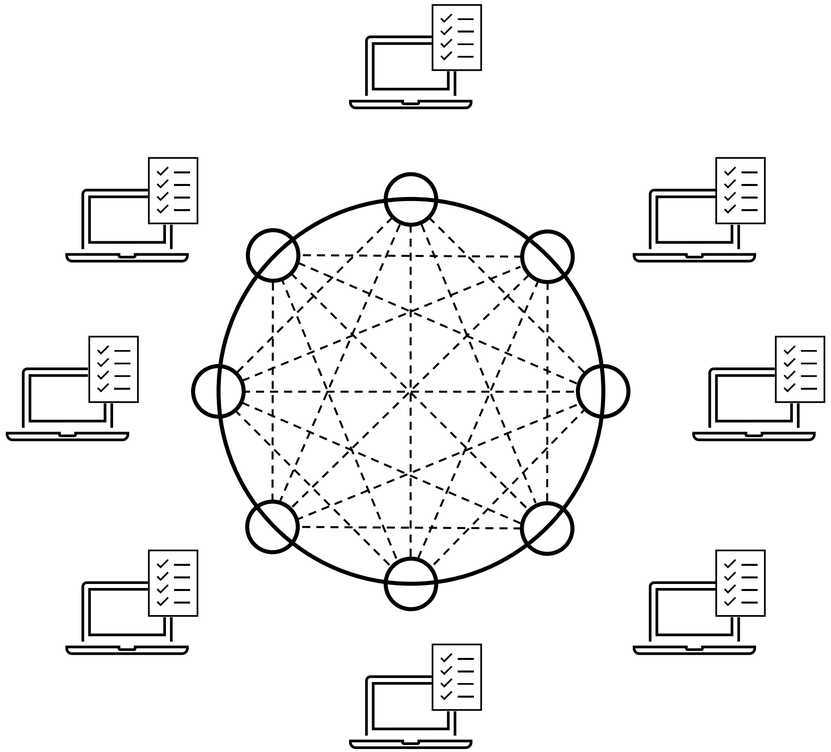
Trust Era

Public blockchain & distributed ledgers

- Speed
- Access & collaboration
- Trust

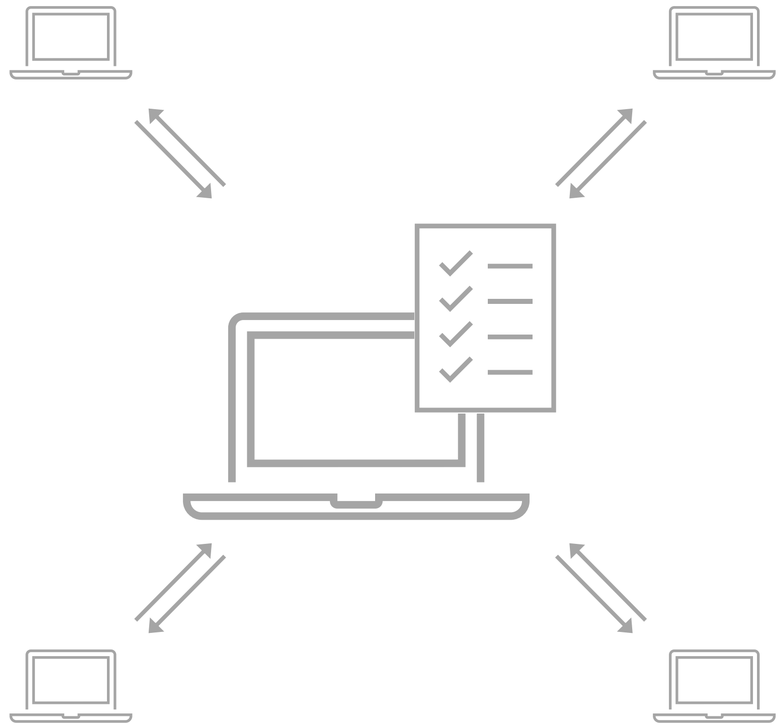
Distributed ledgers are a key component of Web 3.0 because of their qualities

DISTRIBUTED LEDGER



Entire network records and validates each transaction

CENTRALIZED LEDGER



Single authority verifies, records, and executes transactions

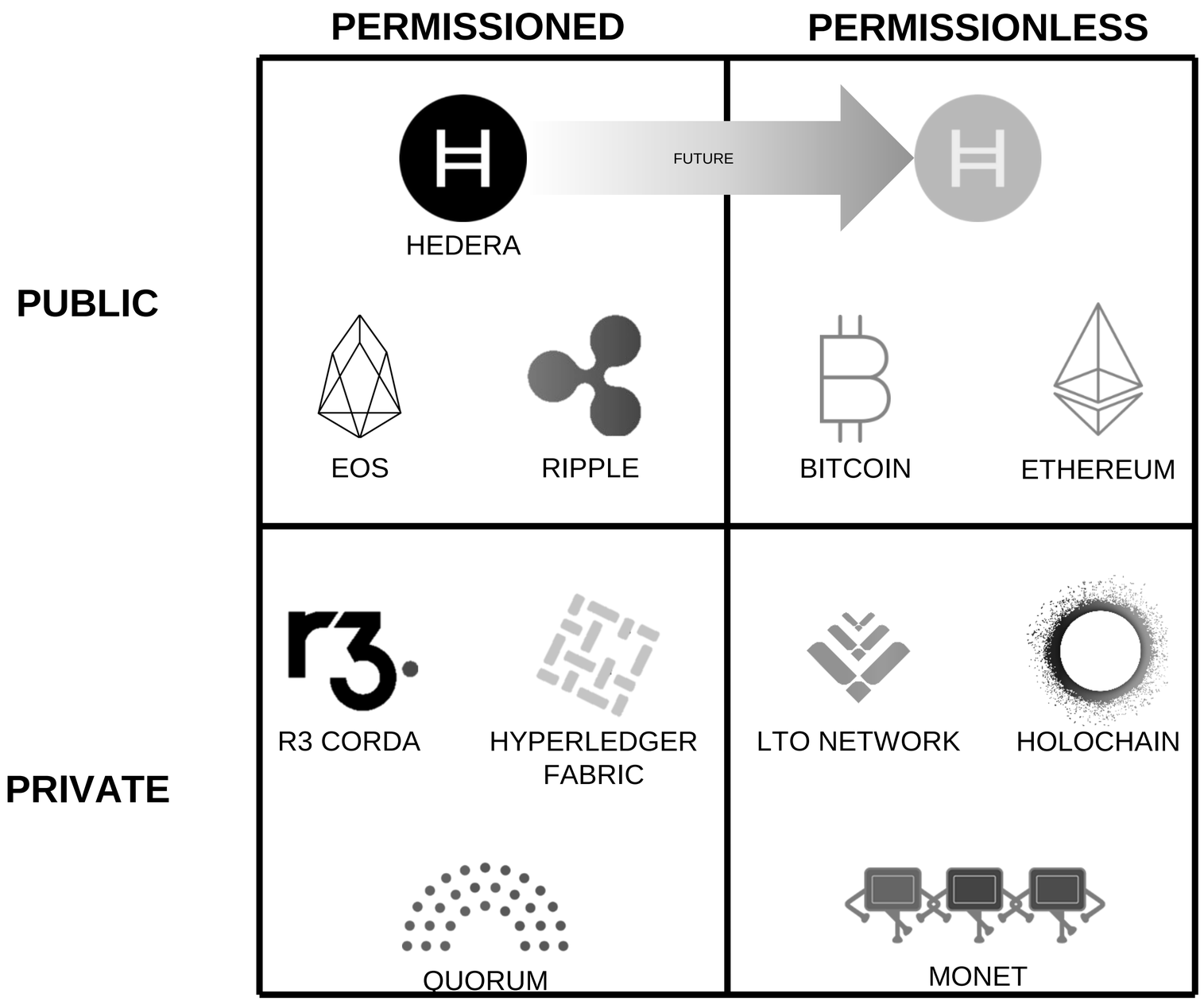
Distributed ledgers are a key component of Web 3.0 because of their qualities

No central point of failure to attack

Reliant on strong cryptography to prove data integrity and tamper resistance

Rules for the ledger are determined by a governing concept of some sort

Require a consensus mechanism to determine the rules for adding new transactions to the state of the ledger


PERMISSIONED
PERMISSIONLESS
PUBLIC
PRIVATE
HEDERA
FUTURE
EOS
RIPPLE
BITCOIN
ETHEREUM
R3 CORDA
HYPERLEDGER FABRIC
LTO NETWORK
HOLOCHAIN
QUORUM
MONET

Hedera is a third-generation public distributed ledger

	1ST GENERATION	2ND GENERATION	3RD GENERATION
	Bitcoin Blockchain	Ethereum Blockchain	Hedera™ Hashgraph
TRANSACTIONS PER SECOND	3+ TPS	12+ TPS	10,000+ TPS
AVERAGE TRANSACTION FEE	$22.57 USD	$19.55 USD	$0.0001 USD
AVERAGE TRANSACTION CONFIRMATION	10-60 MINUTES	10-20 SECONDS	3-5 SECONDS (W/ FINALITY)

- Cryptocurrency transactions. For Hedera, range shown for transactions not requiring a transaction record, but can receive a transaction receipt.
- Avg. Bitcoin tx fee from 6/26/20 - 9/24/20 from https://blockchair.com/bitcoin/charts/average-transaction-fee-usd?interval=3m
- Avg. Ethereum tx fee from 6/26/20 - 9/24/20 from https://blockchair.com/ethereum/charts/average-transaction-fee-usd?interval=3m

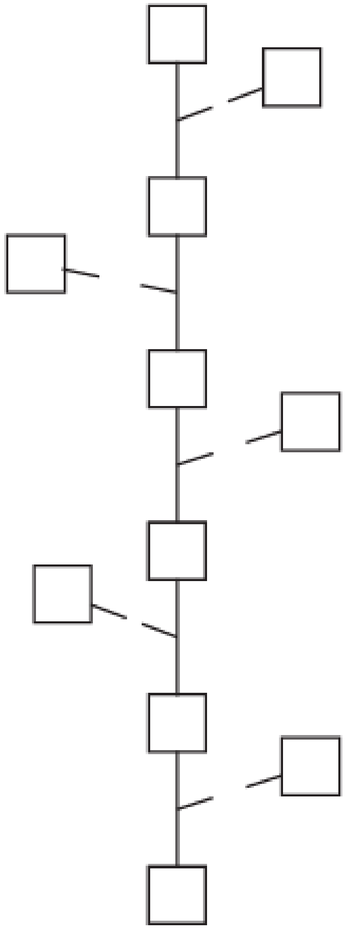
3rd Generation Public Ledgers

	polygon	Algorand		
Transactions per second	6,500* (claimed)	1,000† (approximate)	12+	**10,000+**
Average Fee (USD)	$0.0020 (variable)	0.001 Algo (fixed)	$19.55 (variable)	**$0.0001^ (fixed)**
Time to Confirmation	5 - 10 seconds (leader block creation)	5 seconds (leader block creation)	10 - 20 seconds (leader block creation)	**3-5 seconds (with finality)**
Energy use per transaction	90+ kWh	0.00534 kWh	102+ kWh	**0.00017 kWh**

THE HBAR FOUNDATION

* Actual TPS closer to 53 over a rolling 30 day period - https://thecoinweekly.com/not-so-fast-are-eth-killers-as-speedy-as-they-claim
^ Hedera Transaction fees - https://hedera.com/fees
† https://www.algorand.com/resources/algorand-announcements/algorand-2021-performance

TRADITIONAL BLOCKCHAINS



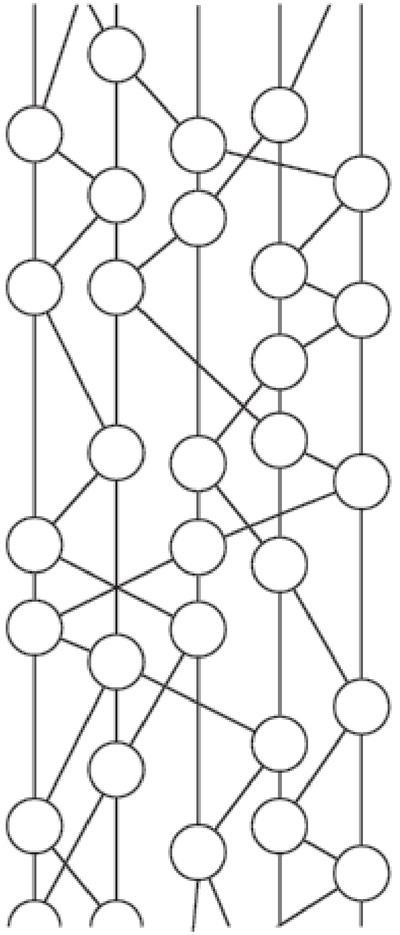
Proof-of-Work (PoW) typically has heavy electrical/computational requirements

No guaranteed consensus, just a “good enough” probability

Inefficient - “stale” blocks are pruned

Can be difficult to scale beyond ~1,000 tps

HASHGRAPH



Hashgraph is an **open-source** consensus algorithm and data structure

Uses a directed acyclic graph (DAG) and novel inventions

- Gossip about gossip
- Virtual voting

Hedera currently supports 10,000+ cryptocurrency transactions per second

Finality within 3-5 seconds

TRADITIONAL BLOCKCHAINS

HASHGRAPH

14,000+ DEVELOPERS

Attending global hackathons, meetups, and active in Discord.

100+ APPLICATIONS

In production on Hedera Mainnet, since open access on Sept. 2019

>4,000,000+ TXS/DAY

Far surpassing the daily transaction volume of Ethereum.

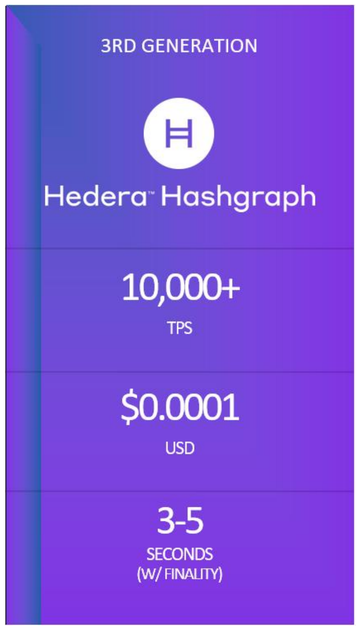
>2,300,000,000+ TXS TO DATE

Most used public DLT surpassing total transaction volume of Ethereum.

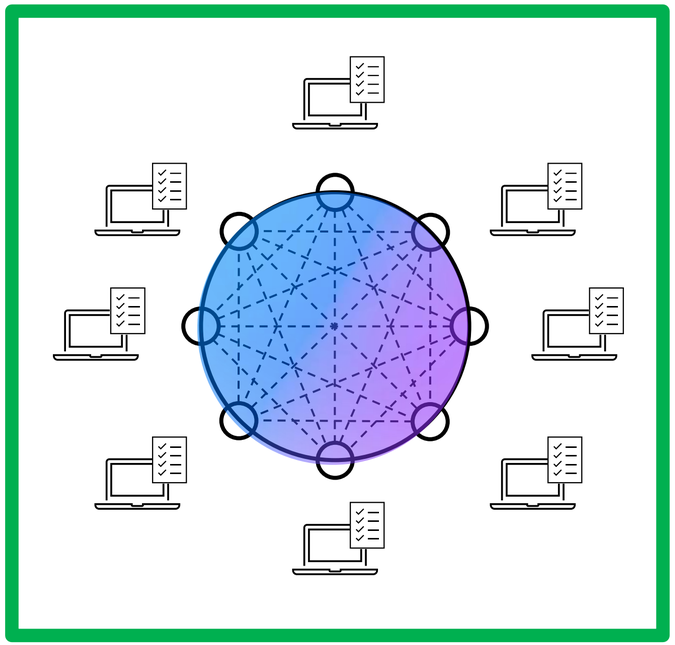
>700,000+ TOTAL ACCOUNTS ON MAINNET

49,500 accounts created in February 2022 alone.

In this session, you will learn how the **Hedera network functions**, how you can **start development**, and **where to learn more**



Introduction



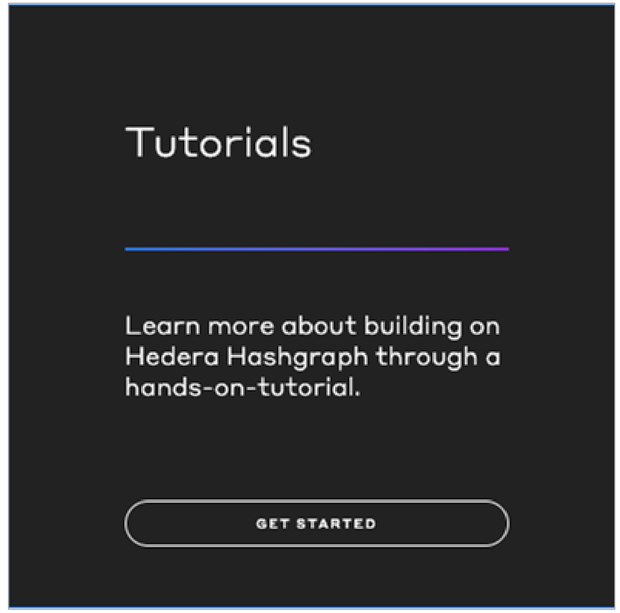
Understand
the Hedera Network

CONSENSUS SERVICE

TOKEN SERVICE

SMART CONTRACTS

Start Developing on
Hedera



Get Resources and
Learn More

END USERS

INDIVIDUALS

ENTERPRISES

GOVERNMENTS

DEVICES

APPLICATION USE CASES

DATA INTEGRITY

TOKENIZATION

DECENTRALIZED IDENTITY

SUPPLY CHAIN

PAYMENTS

CONSENSUS SERVICE

TOKEN SERVICE

SMART CONTRACTS

PRIMARY NETWORK SERVICES

HEDERA GOVERNING COUNCIL

HASHGRAPH CONSENSUS

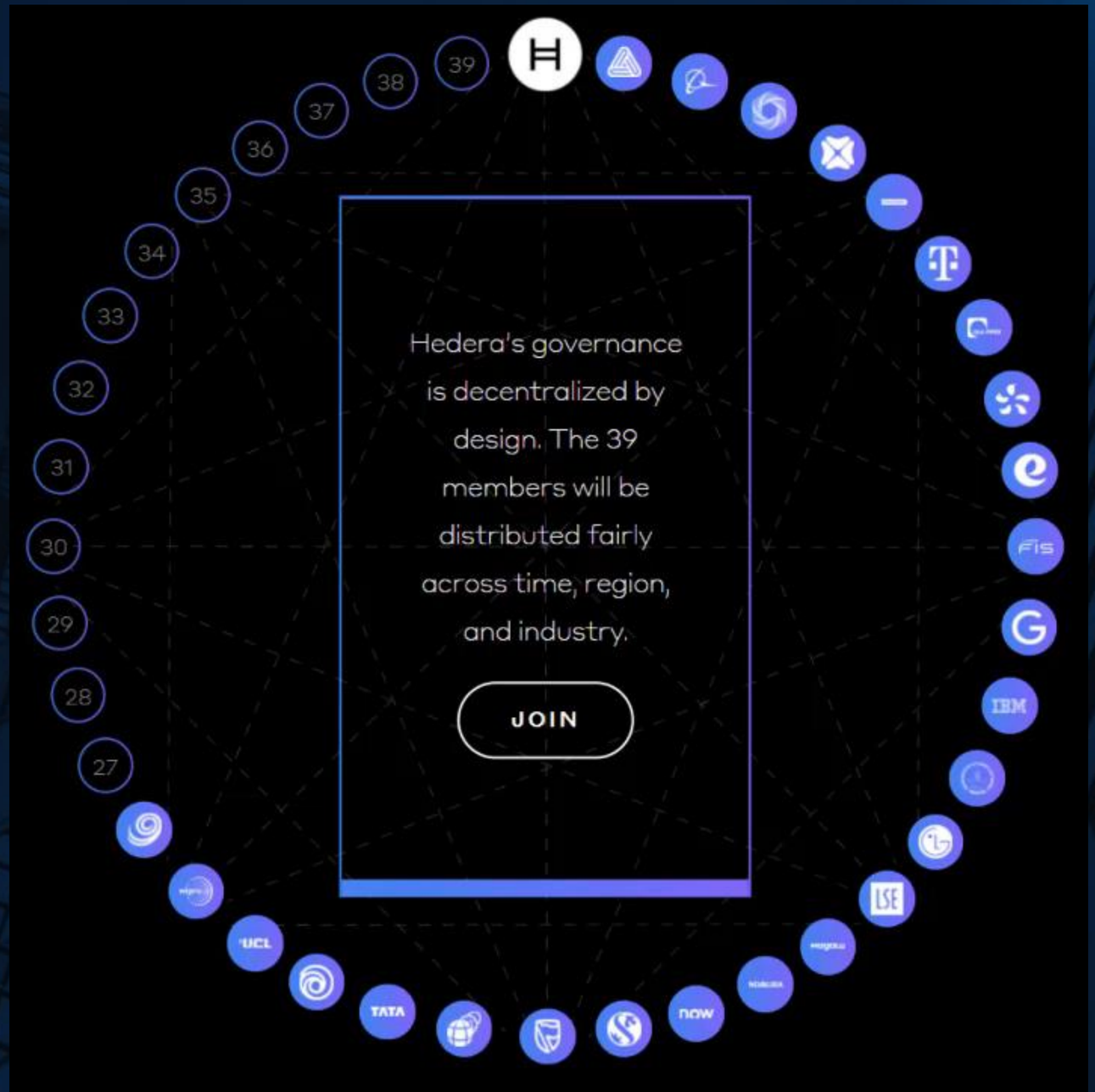
HEDERA MAINNET & MIRROR NODES

Hedera Governing Council

Up to **39** leading global organizations

20+ unique industries across **5** continents, touching all major markets

3-year maximum term, with up to 2 consecutive terms



2.6% influence per member (equal vote)

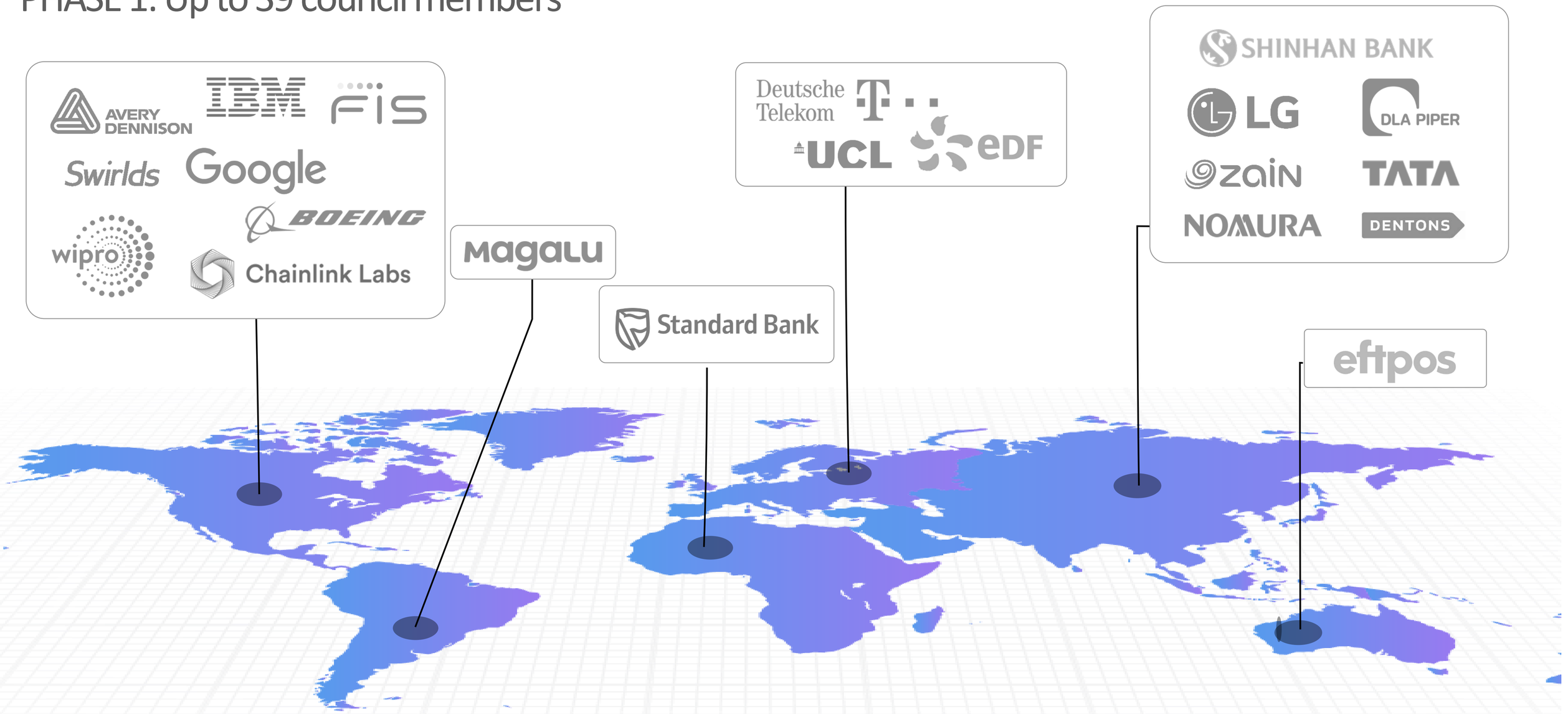
7 committees: Membership, Marketing, Finance, Audit, Technical Steering/Product Pricing, Legal & Regulatory

Every member required to run a network node


Chainlink Labs
IBM
LG
BOEING
wipro
DENTONS
Building the future together
zain
Standard Bank
UCL
Deutsche Telekom
TATA COMMUNICATIONS
NOMURA
eDF
HEDERA
GOVERNING COUNCIL
magalu
Swirlds
eftpos
DLA PIPER
SHINHAN BANK
AVERY DENNISON
FIS
Google

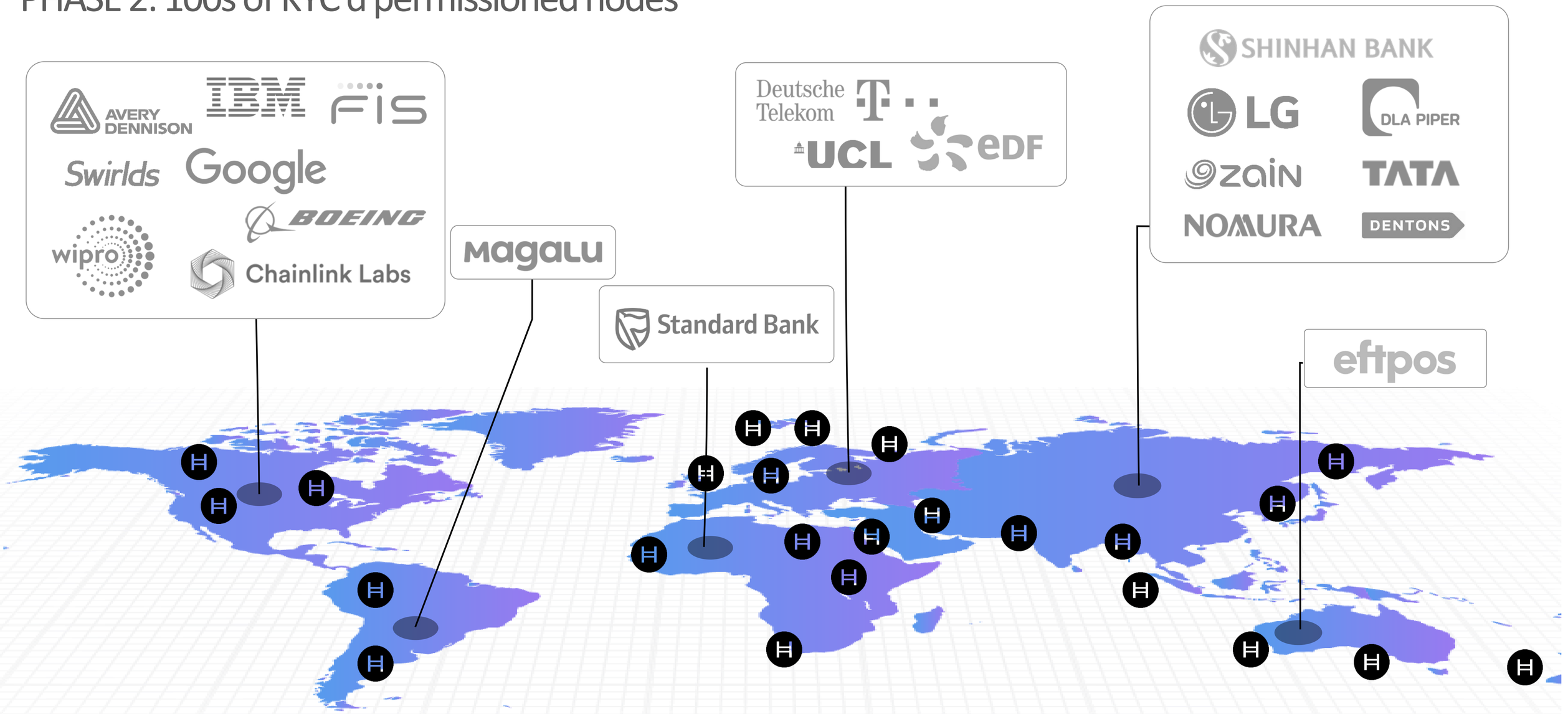
NETWORK GROWTH OVER TIME

PHASE 1: Up to 39 council members



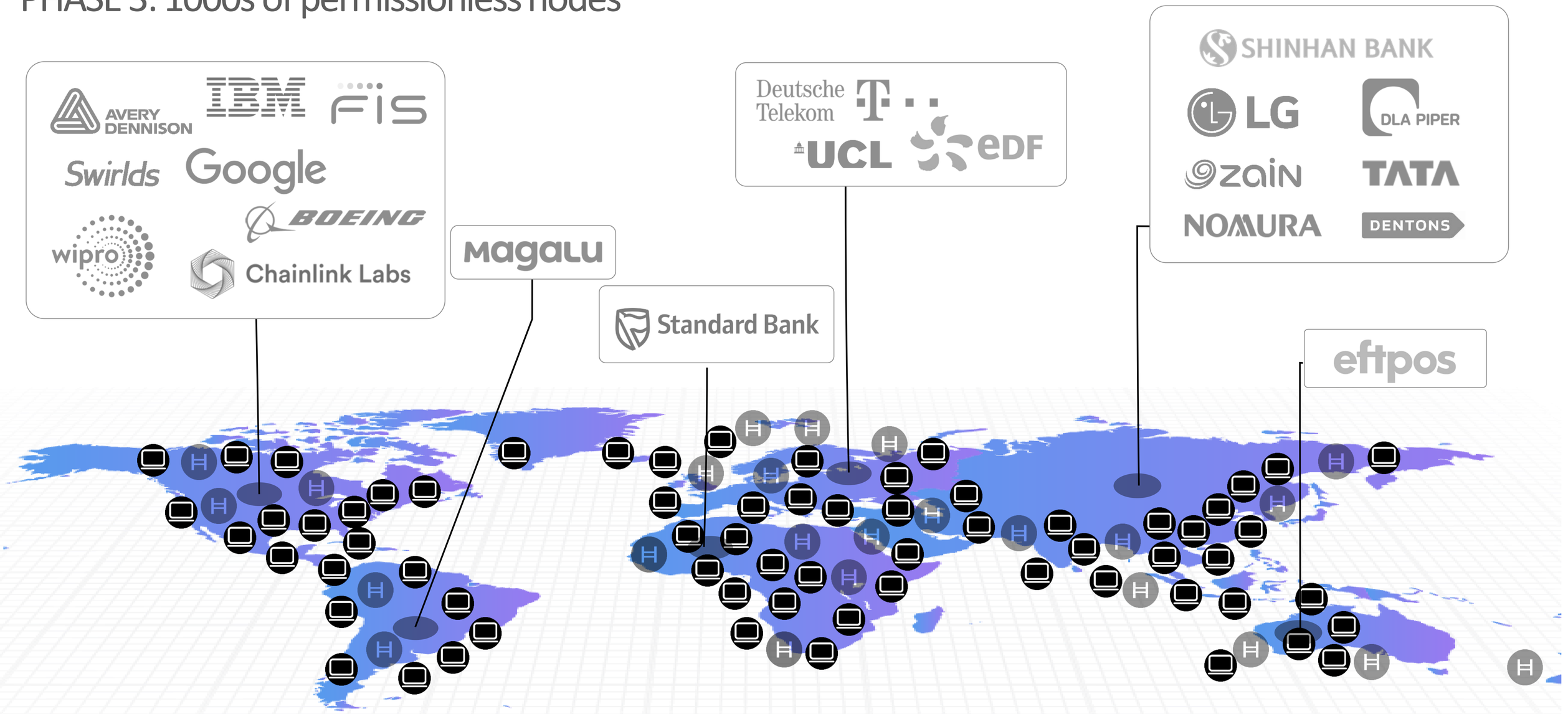
NETWORK GROWTH OVER TIME

PHASE 2: 100s of KYC'd permissioned nodes

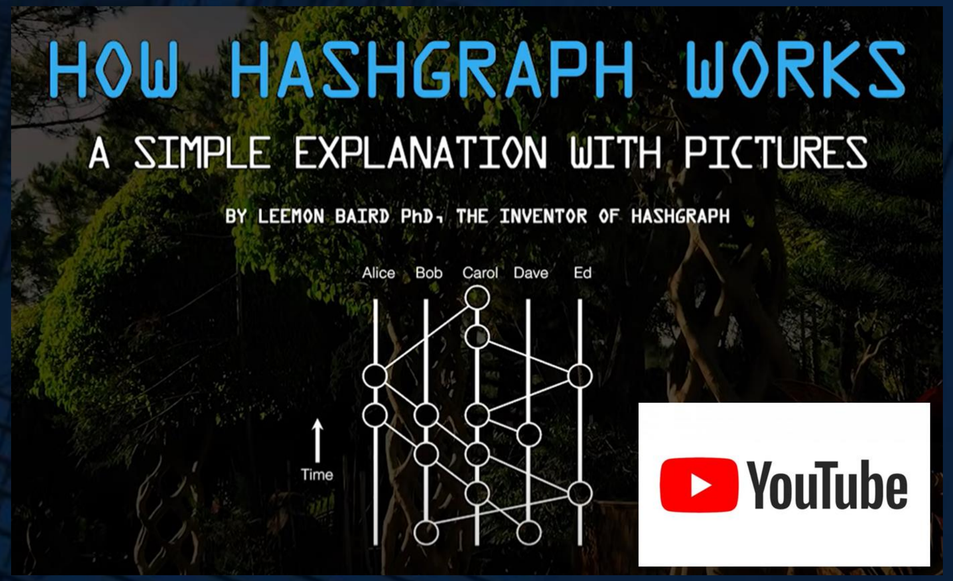


NETWORK GROWTH OVER TIME

PHASE 3: 1000s of permissionless nodes

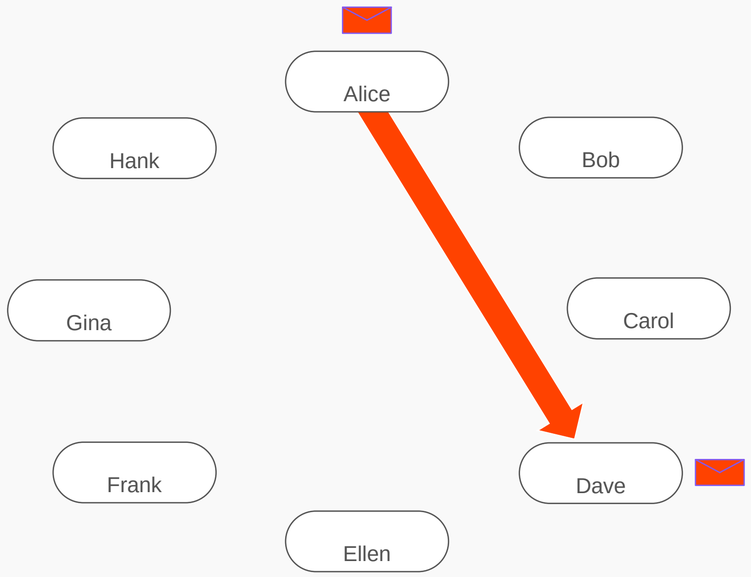


A brief overview of how hashgraph works

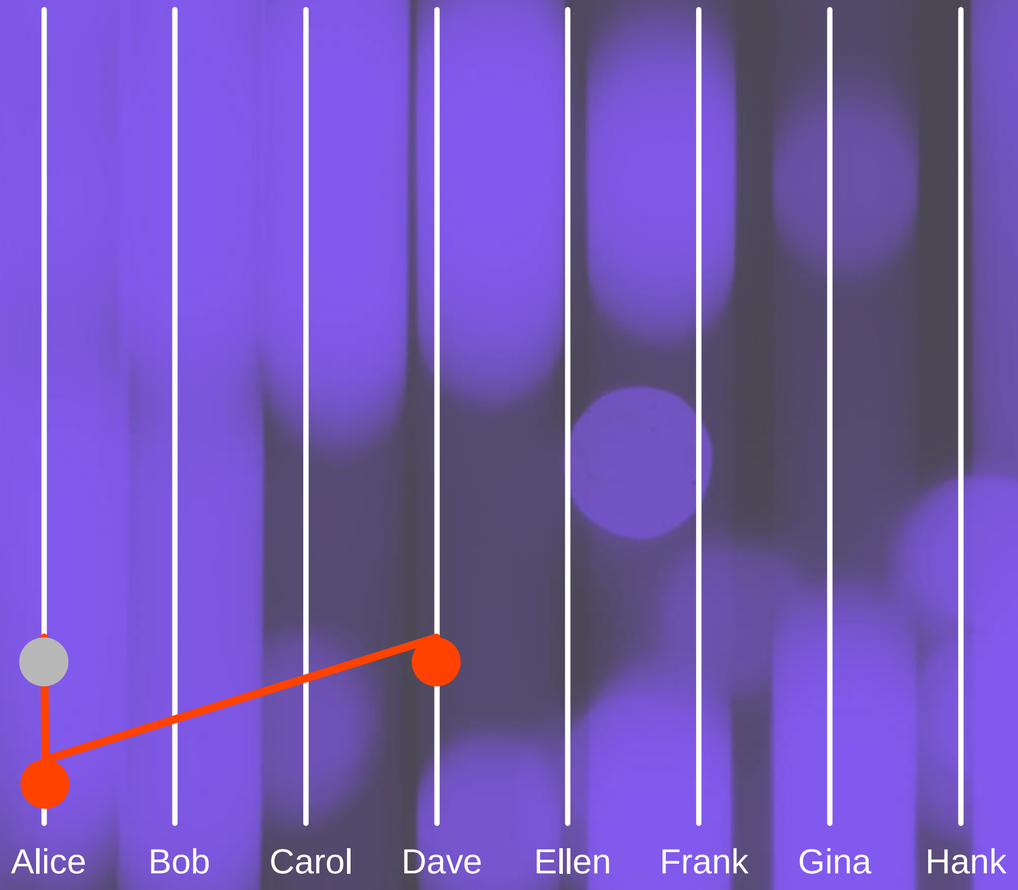


hedera.com

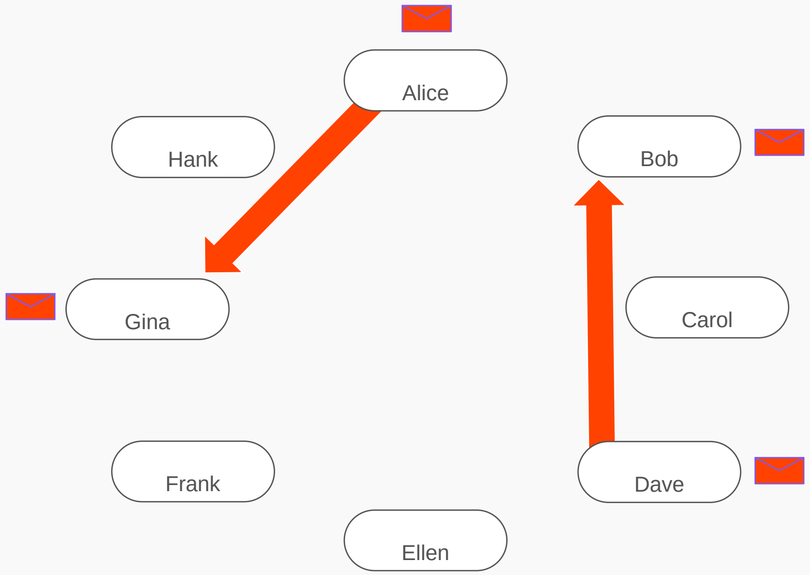
Gossip about gossip



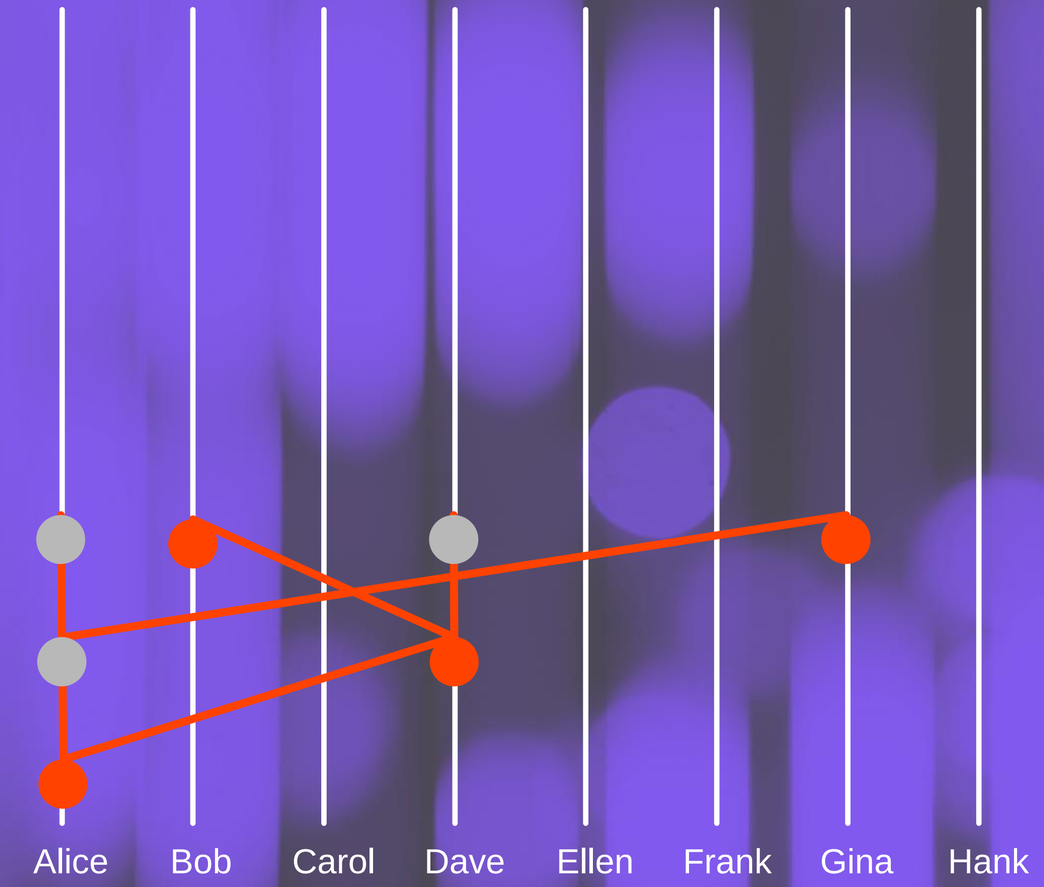
Gossip about Gossip



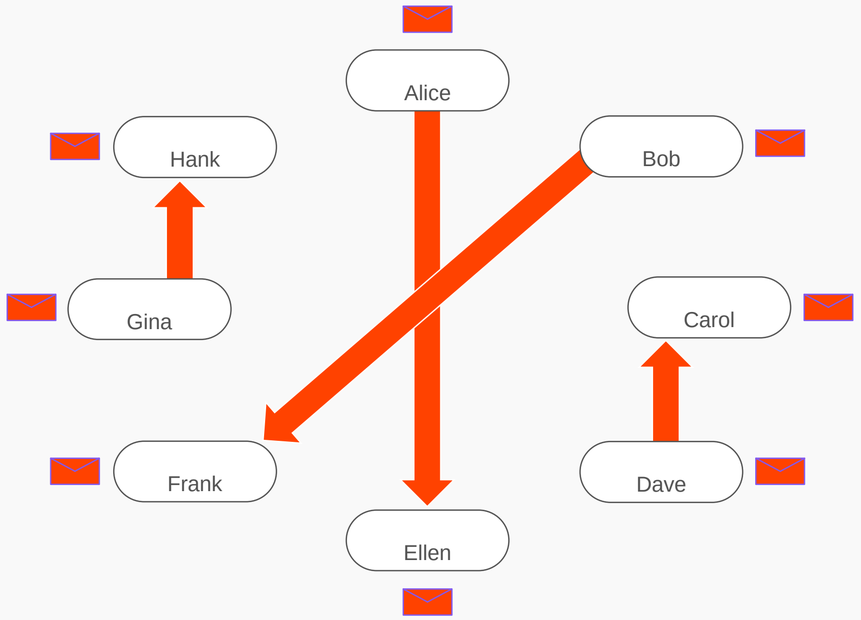
Gossip about gossip



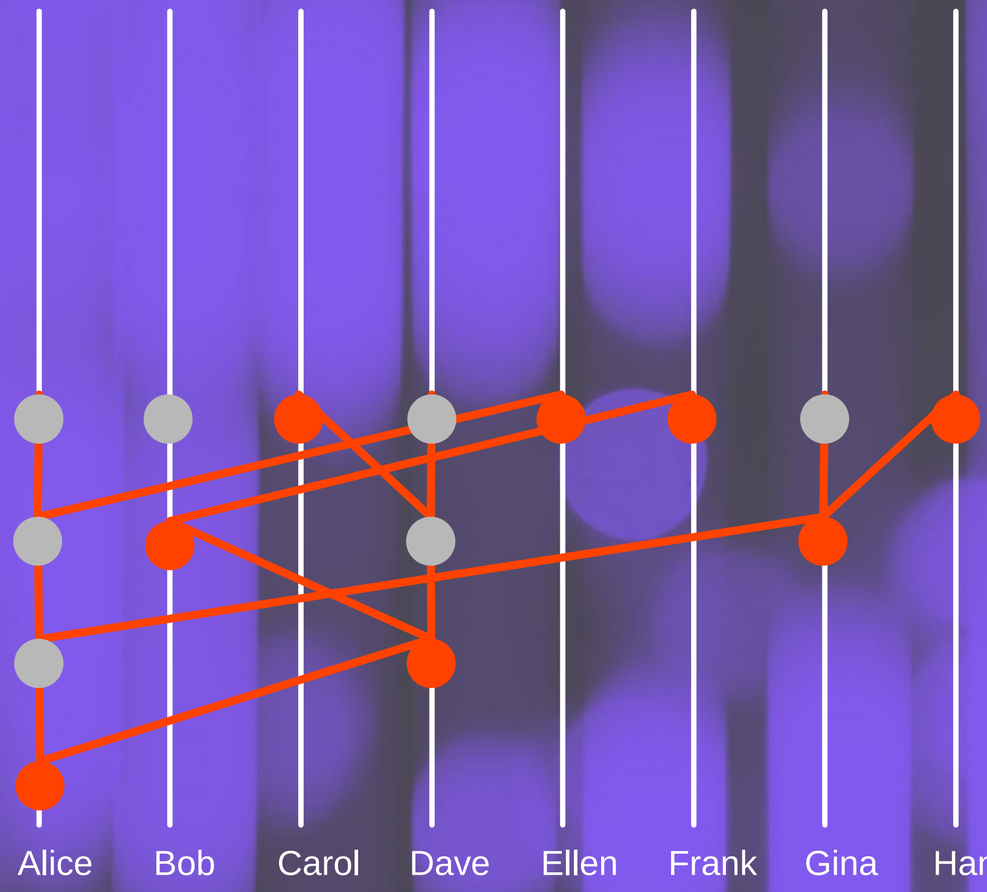
Gossip about Gossip



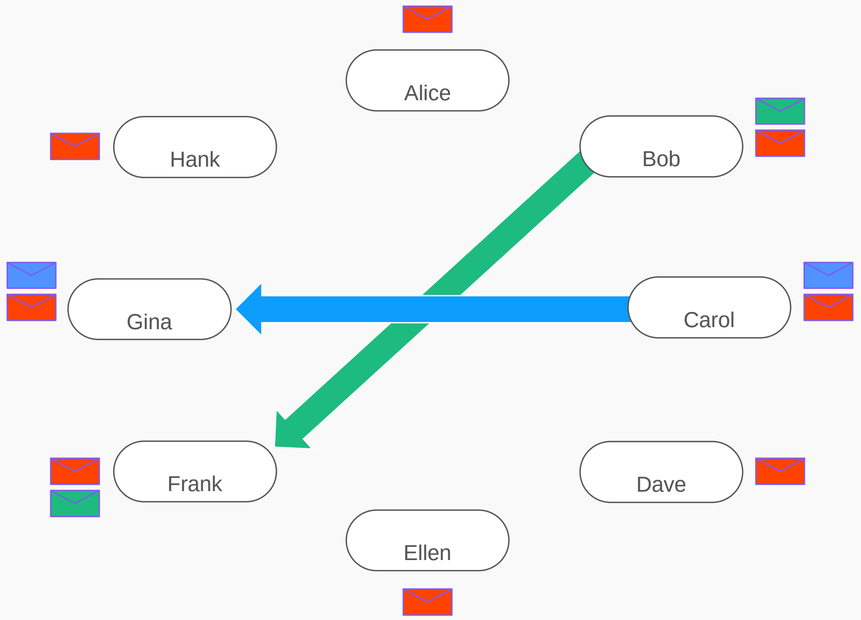
Gossip about gossip



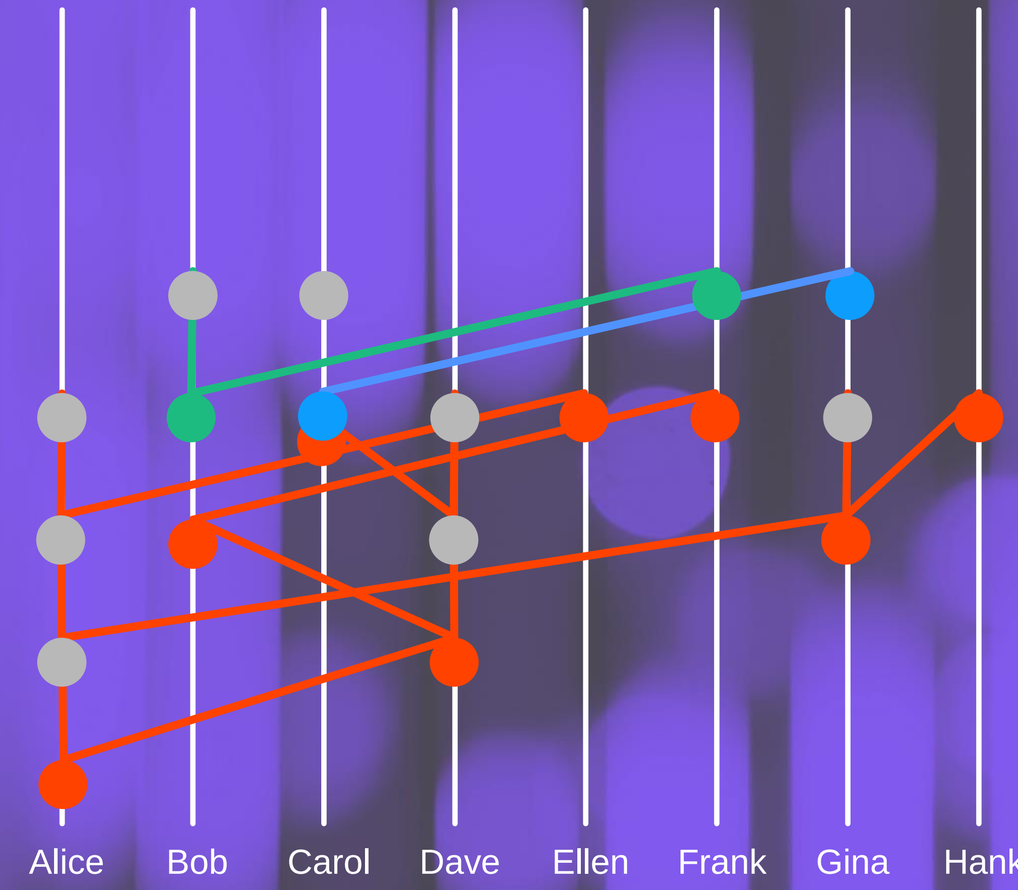
Gossip about Gossip



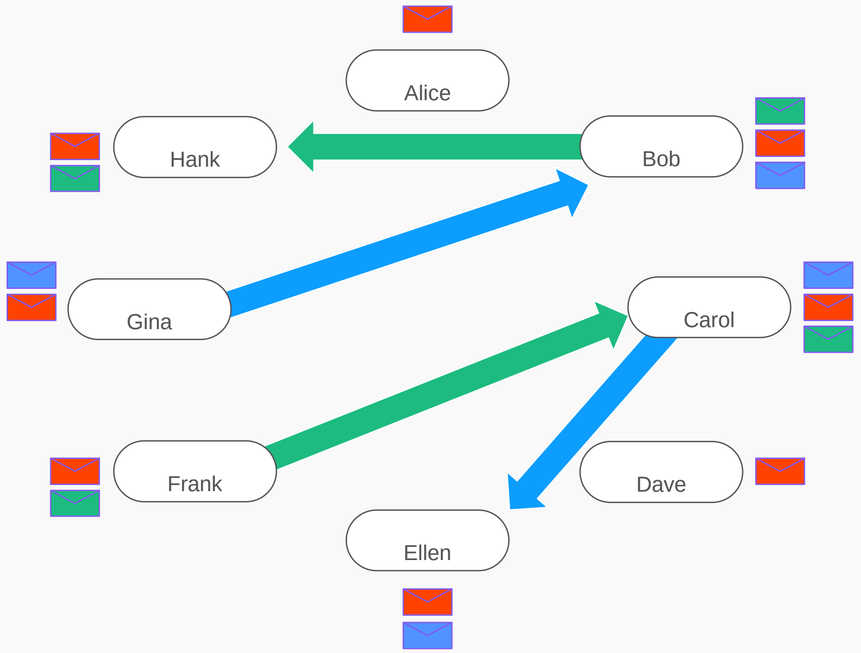
Gossip about gossip



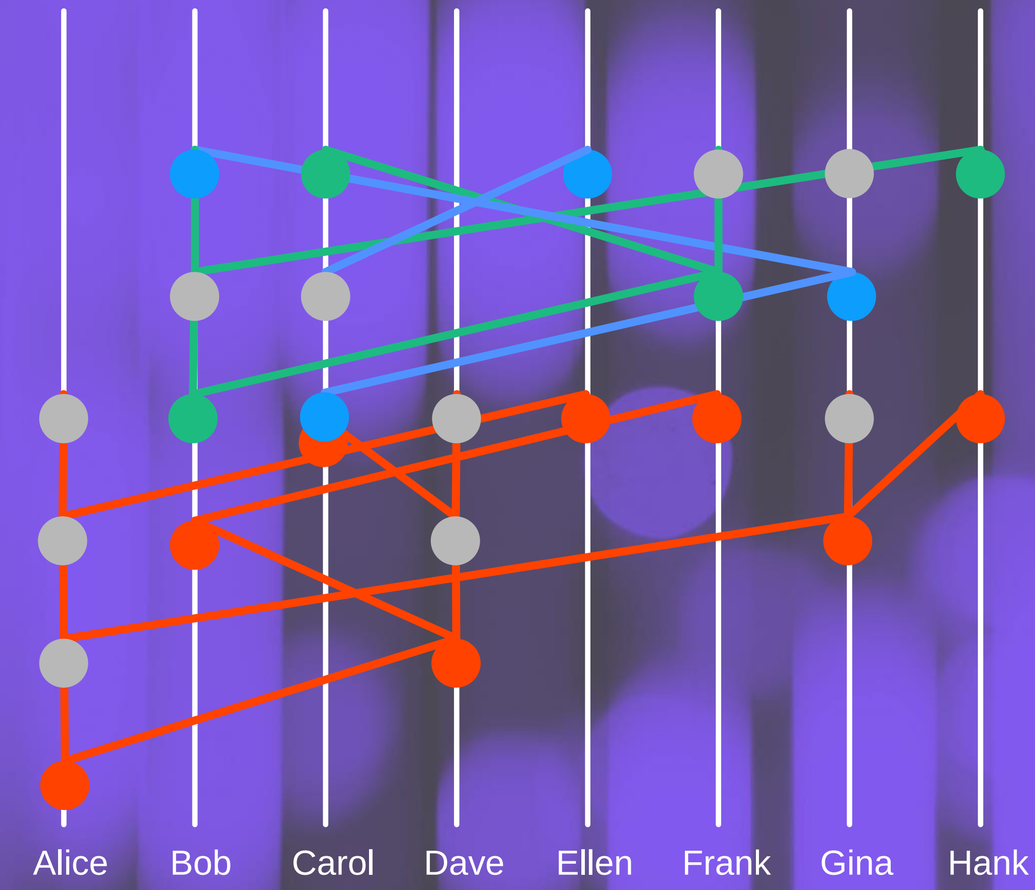
Gossip about Gossip



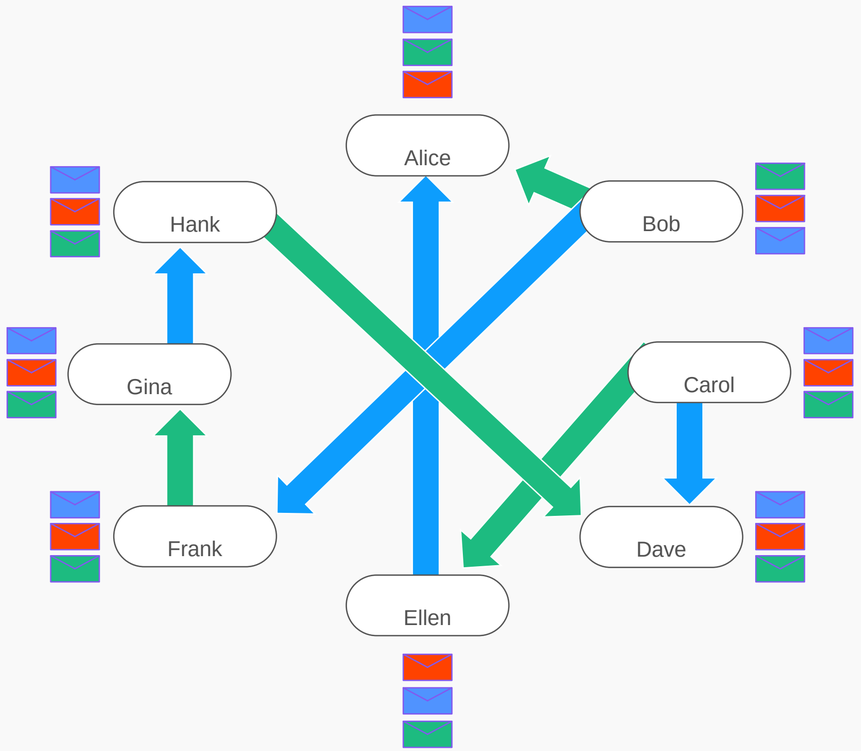
Gossip about gossip



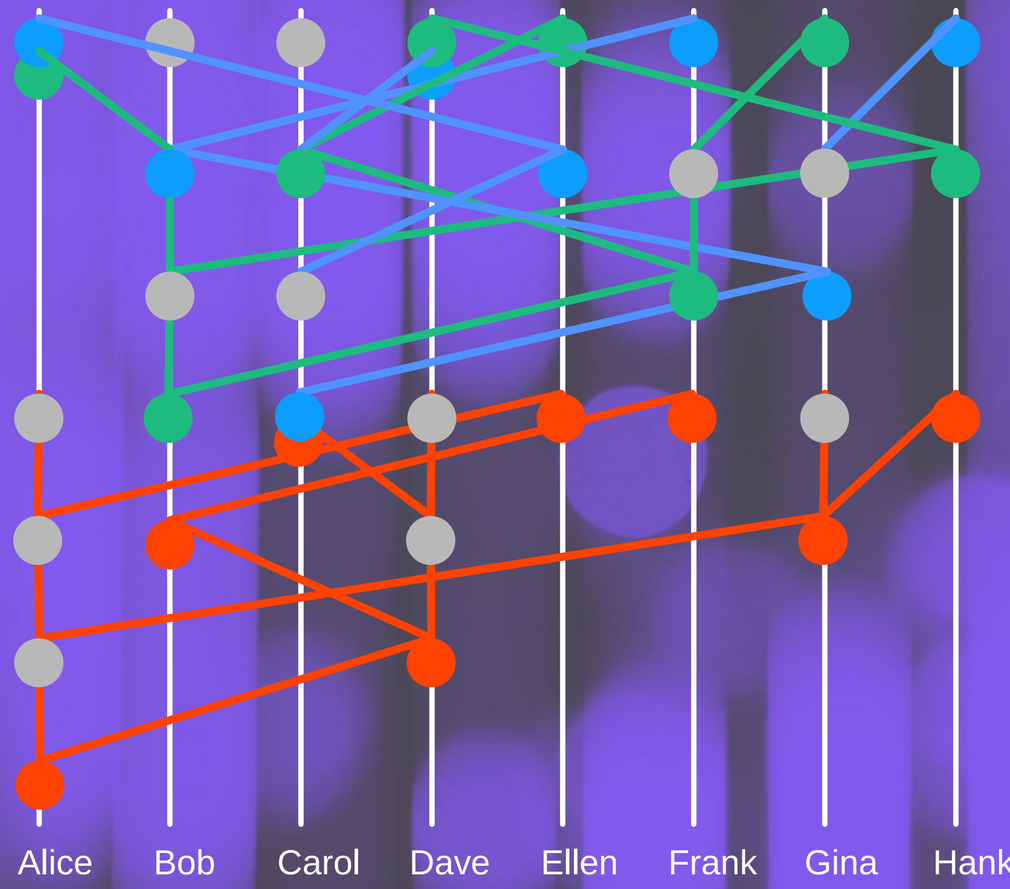
Gossip about Gossip

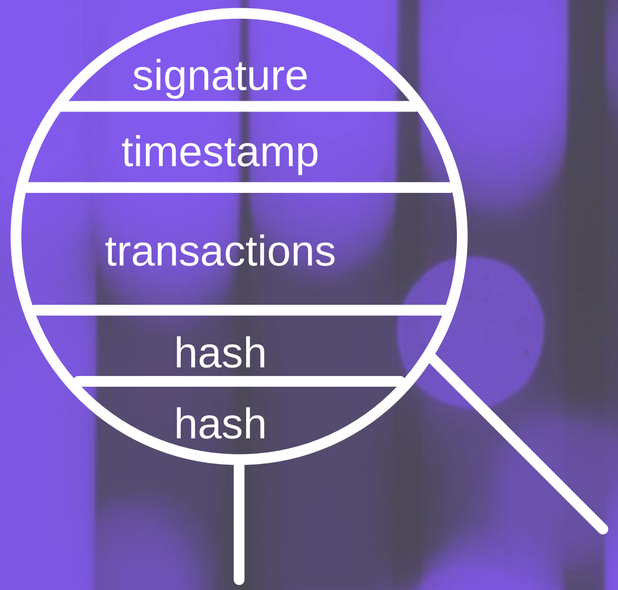


Gossip about gossip



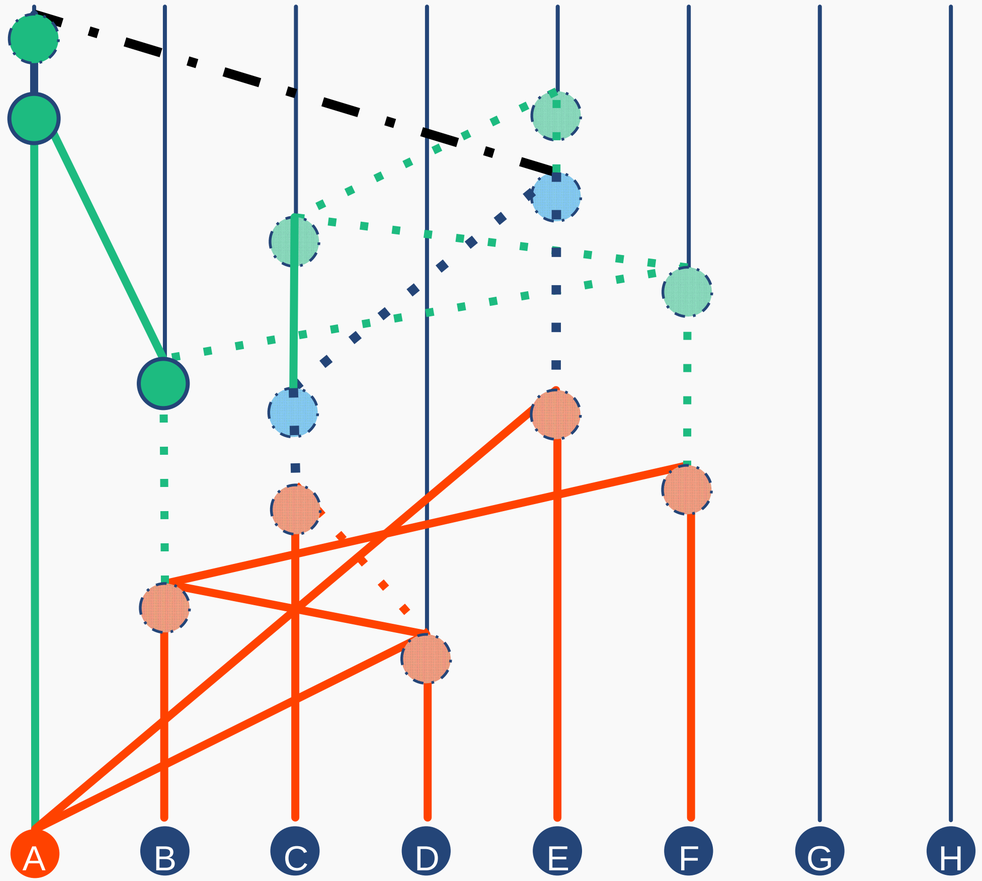
Gossip about Gossip

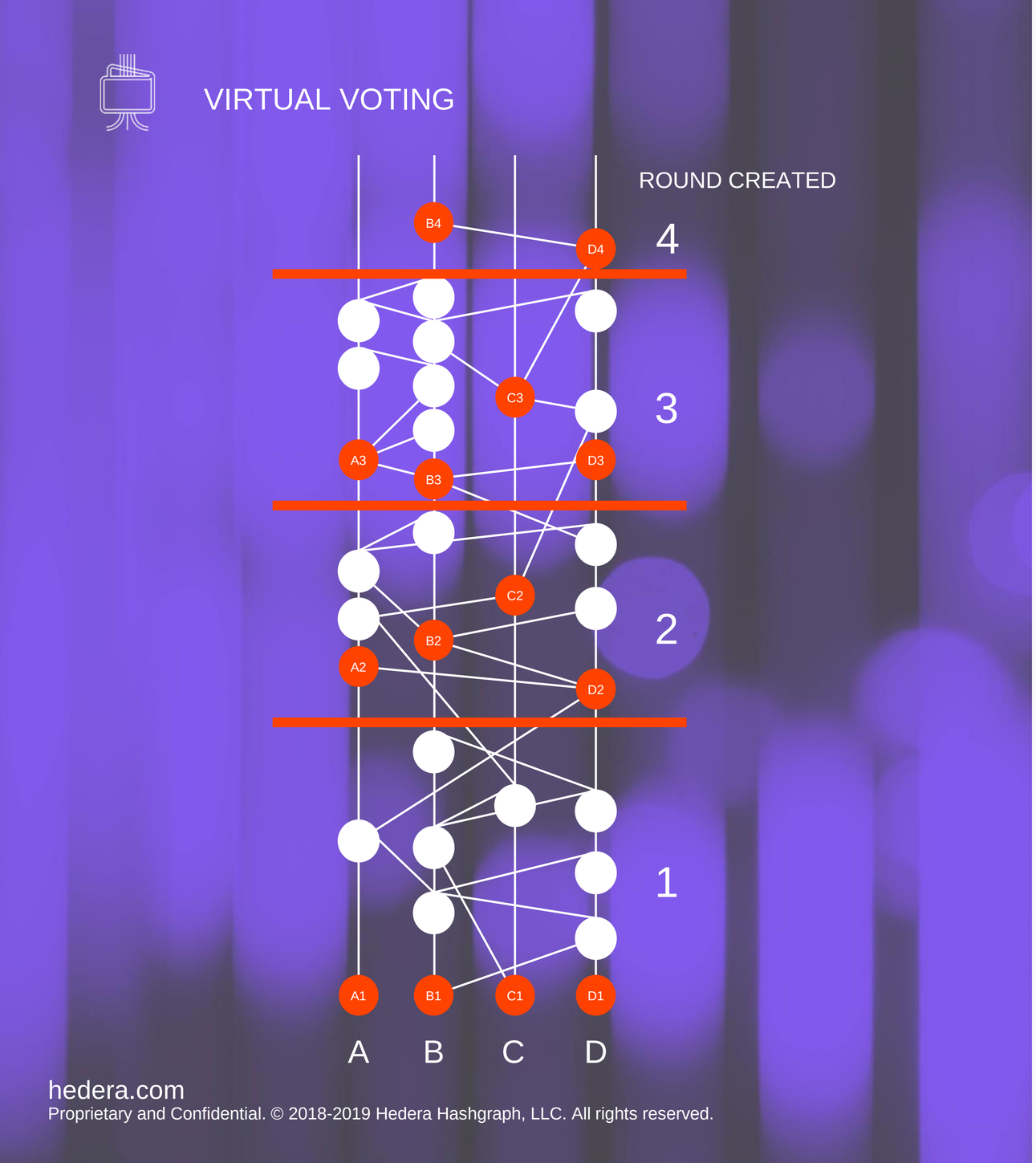




Share events unknown to each other during gossip

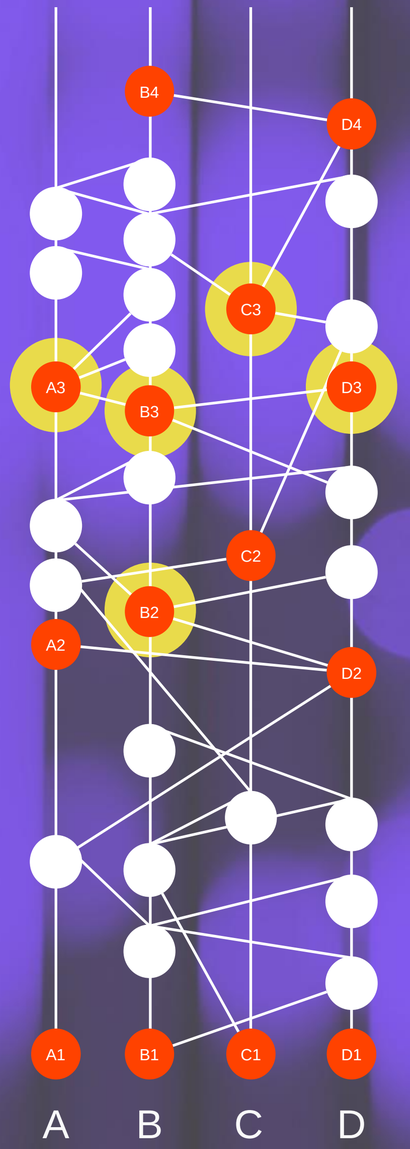
Gossip about Gossip





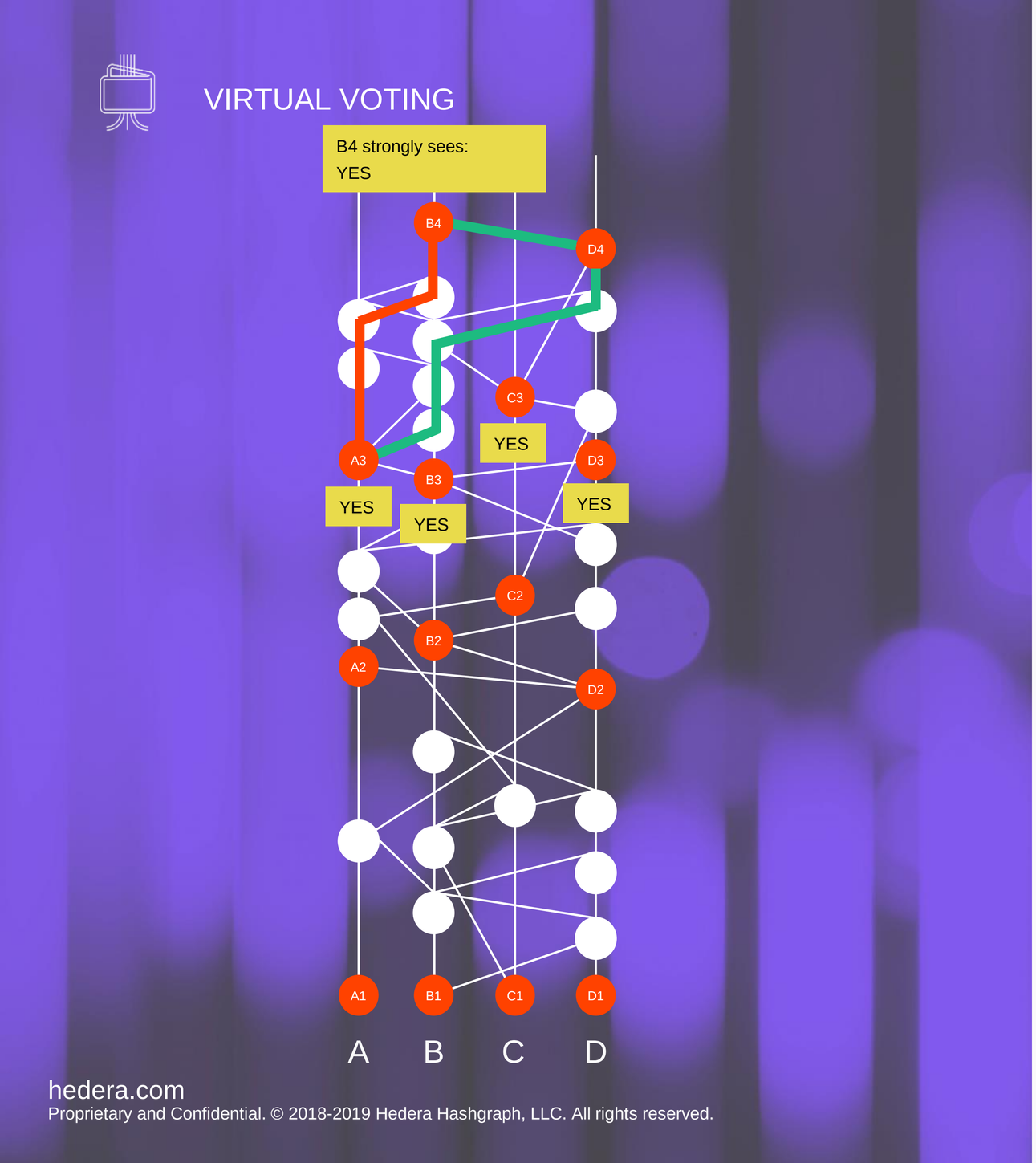
Rounds and witnesses

- Round: Created when the supermajority of witnesses in the previous round can be strongly seen
- Witness: First event in a round for a given node



Famous witnesses

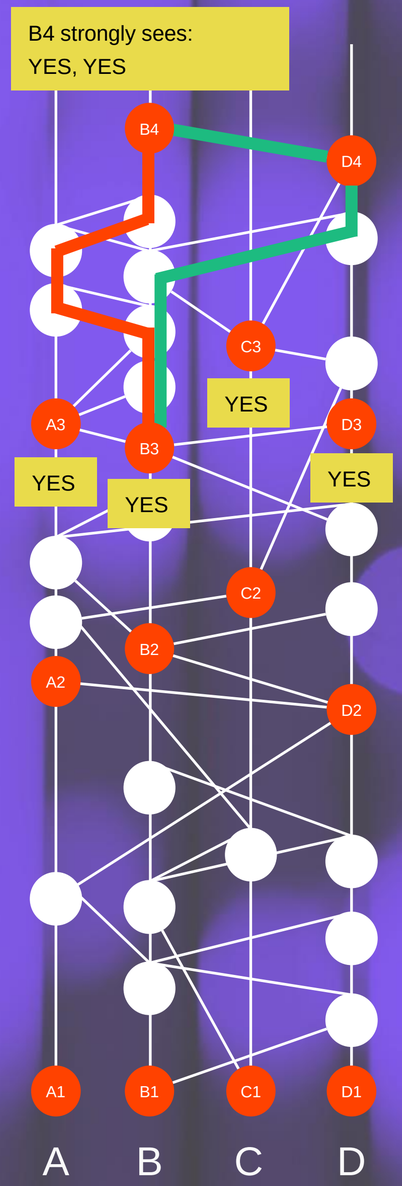
- Famous witnesses: means lots of people see it in the next round



Counting votes

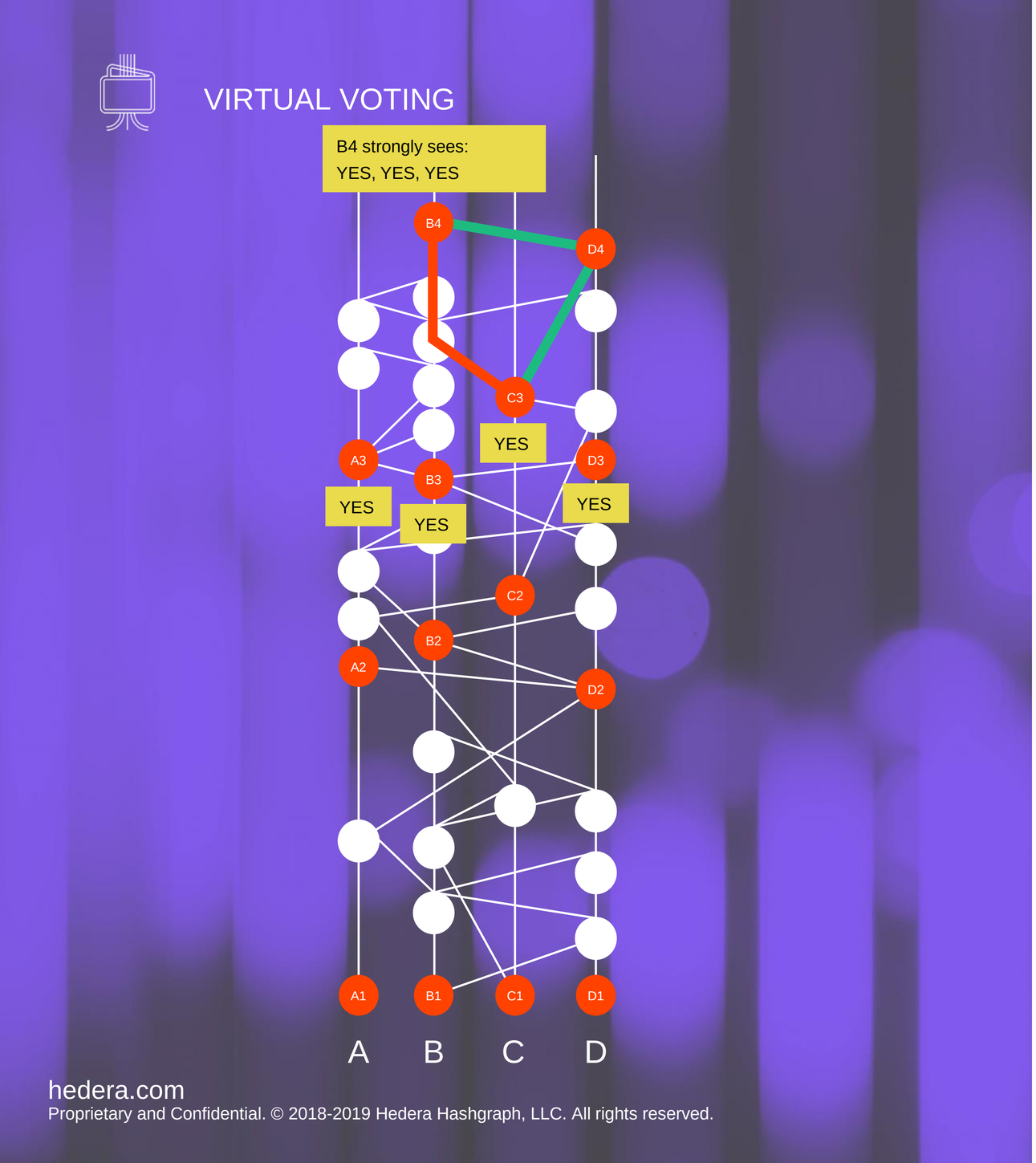
- Strongly seen: To strongly see someone, you have to see them through a supermajority. That means more than two-thirds of the population (or stake as we will see later).

VIRTUAL VOTING



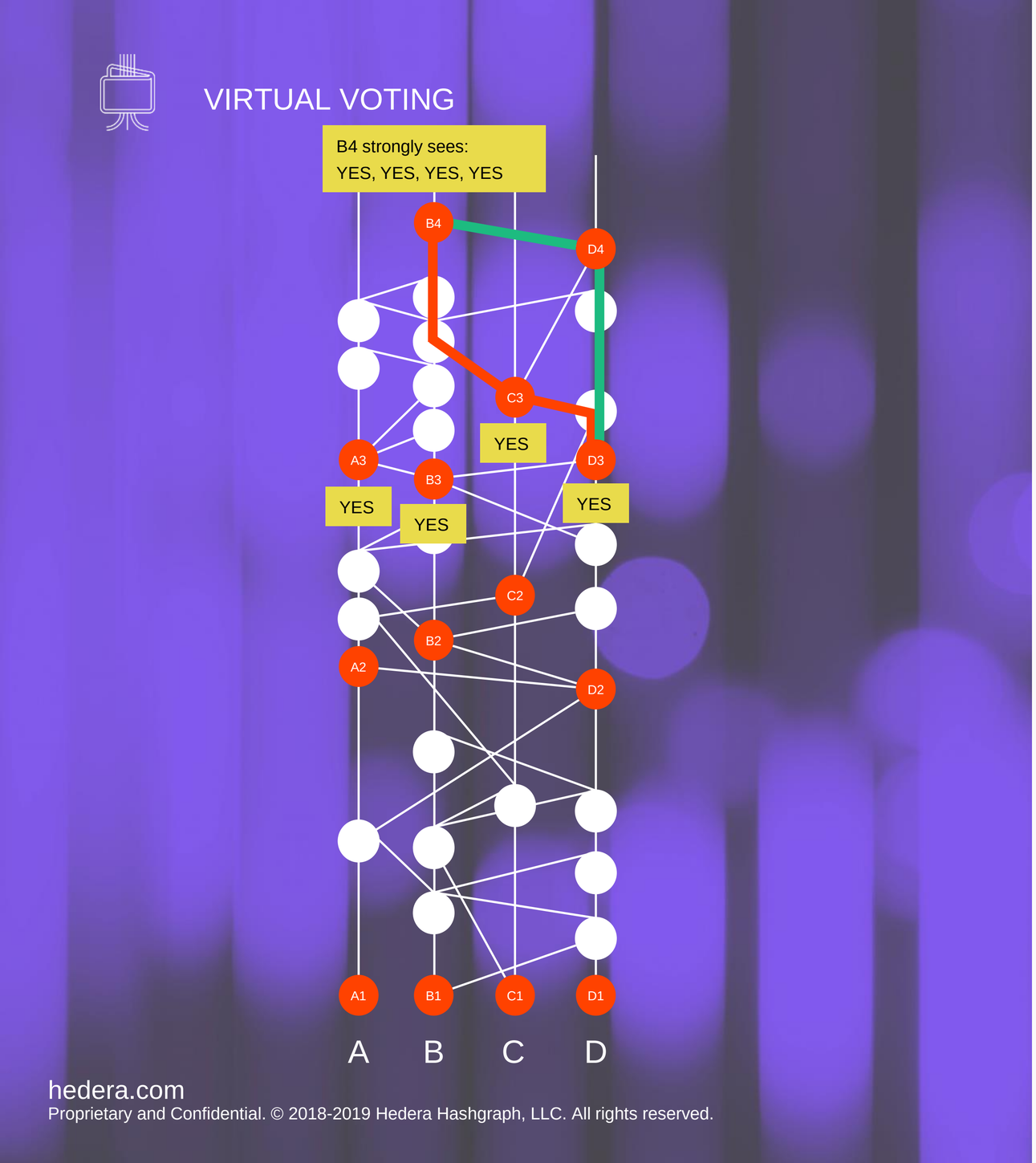
Counting votes

- Strongly seen: To strongly see someone, you have to see them through a supermajority. That means more than two-thirds of the population (or stake as we will see later).



Counting votes

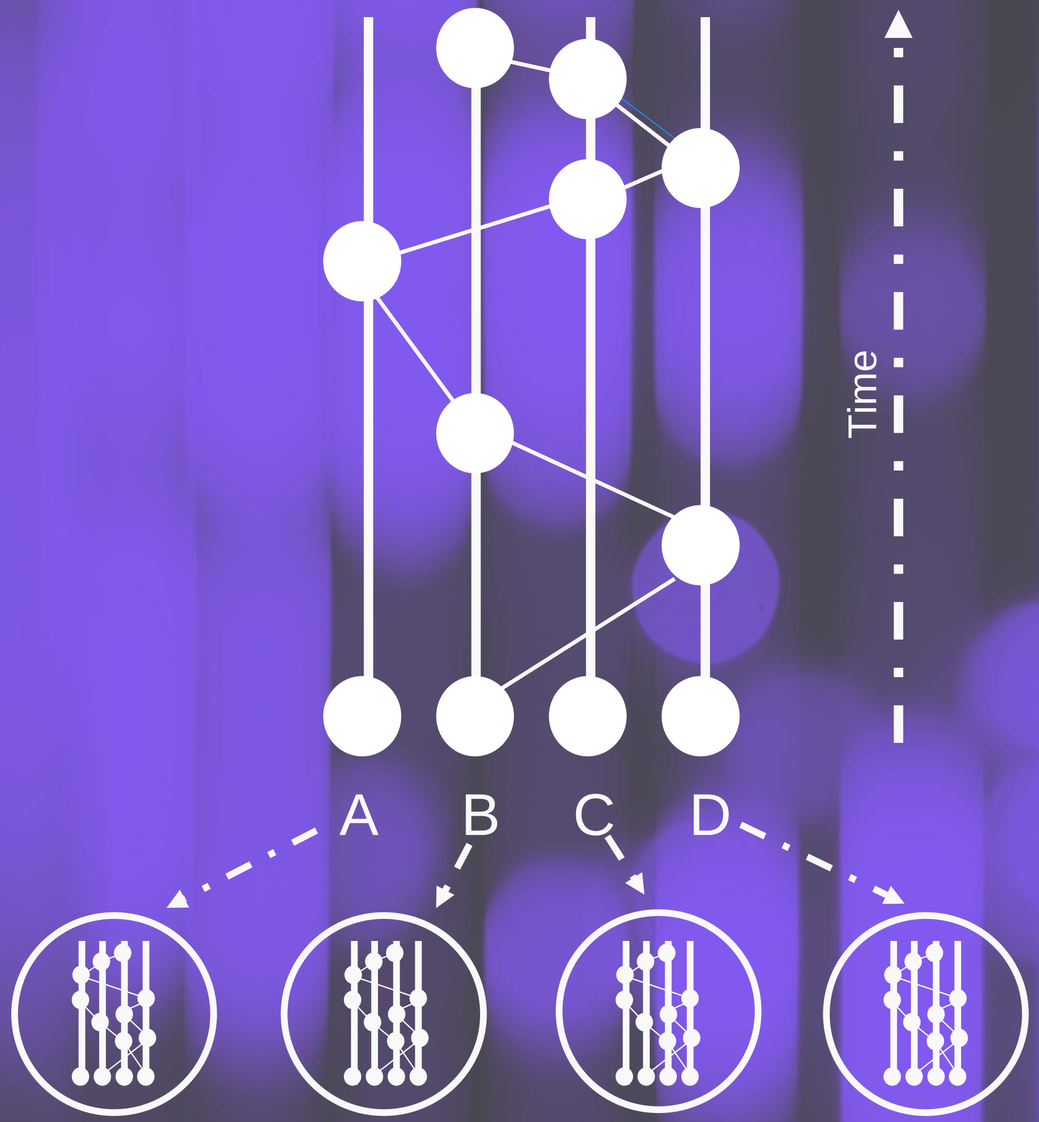
- Strongly seen: To strongly see someone, you have to see them through a supermajority. That means more than two-thirds of the population (or stake as we will see later).



Counting votes

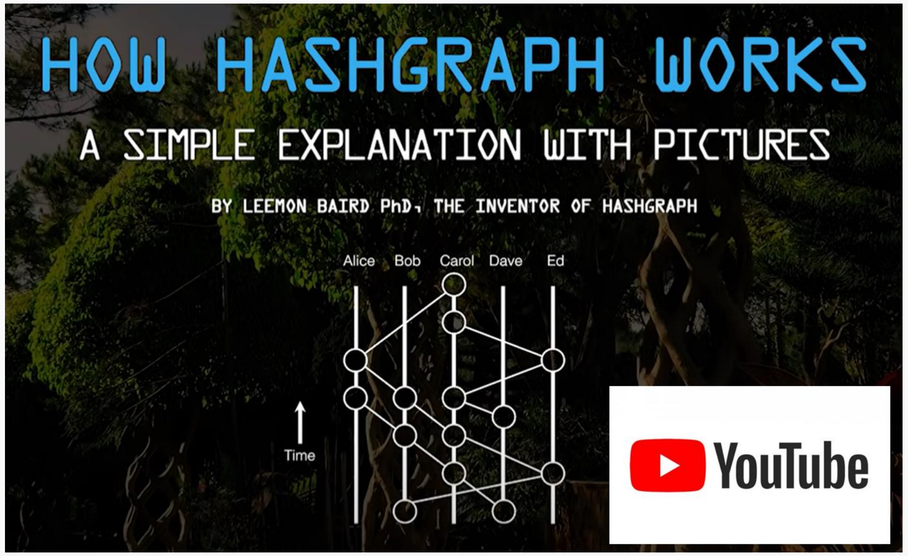
- Strongly seen: To strongly see someone, you have to see them through a supermajority; that means more than two-thirds of hbar staked.

GOSSIP ABOUT GOSSIP



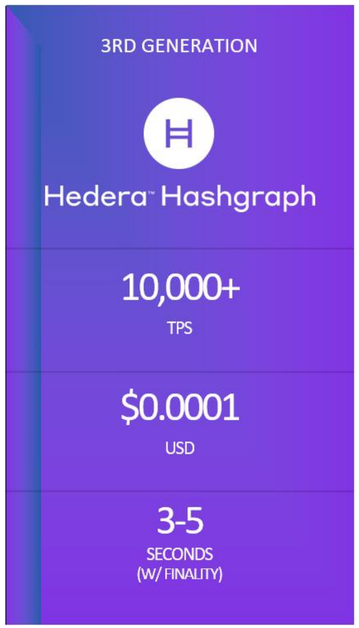
The hashgraph gives each node the entire history of who has talked to who and when

No need to send multiple ballots and receipts over the network as in historical voting models

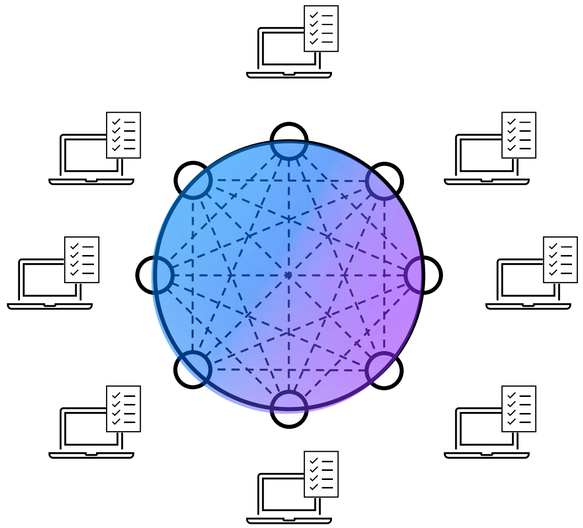


All nodes have the same picture of the hashgraph, except for the very latest events which will eventually sync

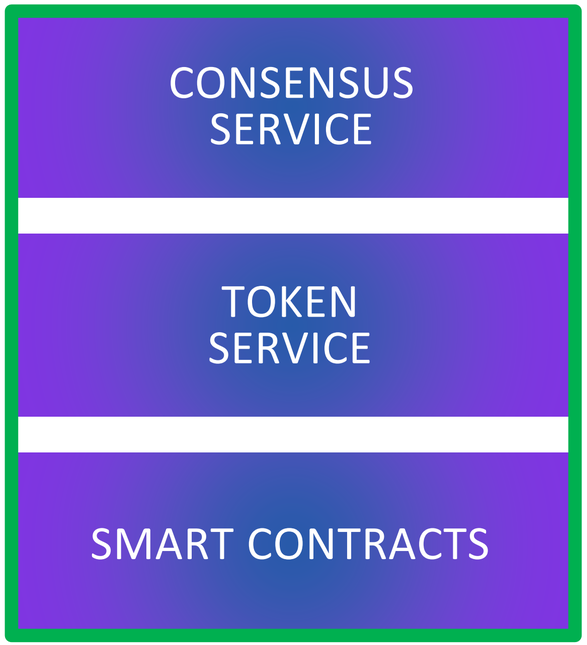
In this session, you will learn how the **Hedera network functions**, how you can **start development**, and **where to learn more**



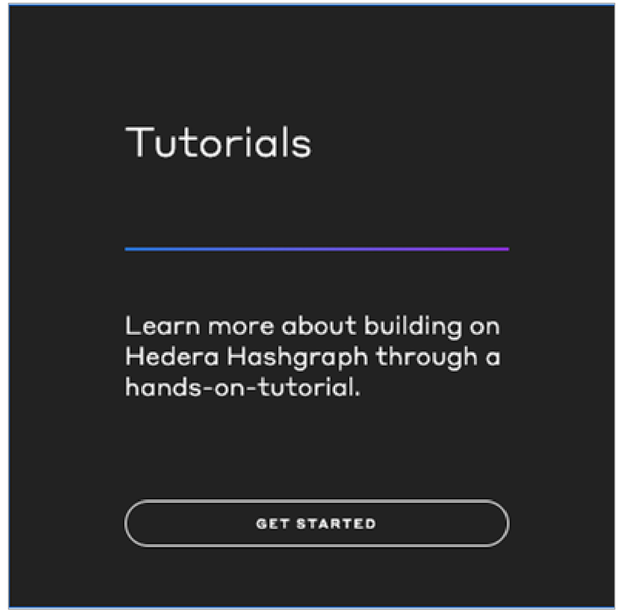
Introduction



Understand
the Hedera Network



Start Developing on
Hedera



Get Resources and
Learn More

END USERS

INDIVIDUALS

ENTERPRISES

GOVERNMENTS

DEVICES

APPLICATION USE CASES

DATA INTEGRITY

TOKENIZATION

DECENTRALIZED IDENTITY

SUPPLY CHAIN

PAYMENTS

TOKEN SERVICE

SMART CONTRACTS

CONSENSUS SERVICE

PRIMARY NETWORK SERVICES

HEDERA GOVERNING COUNCIL

HASHGRAPH CONSENSUS

HEDERA MAINNET & MIRROR NODES

END USERS

INDIVIDUALS

ENTERPRISES

GOVERNMENTS

DEVICES

APPLICATION USE CASES

DATA INTEGRITY

TOKENIZATION

DECENTRALIZED IDENTITY

SUPPLY CHAIN

PAYMENTS

TOKEN SERVICE

SMART CONTRACTS

CONSENSUS SERVICE

PRIMARY NETWORK SERVICES

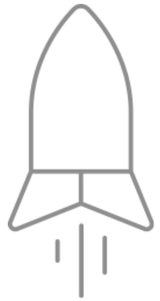
HEDERA GOVERNING COUNCIL

HASHGRAPH CONSENSUS

HEDERA MAINNET & MIRROR NODES

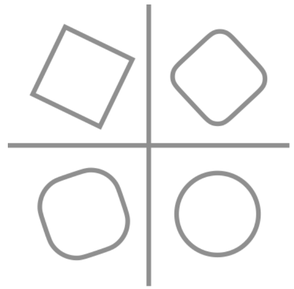
Why build on Hedera?

Hedera helps you meet strict requirements in the following areas…

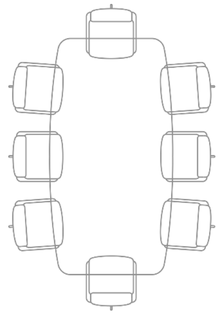


Performance

Security



Stability

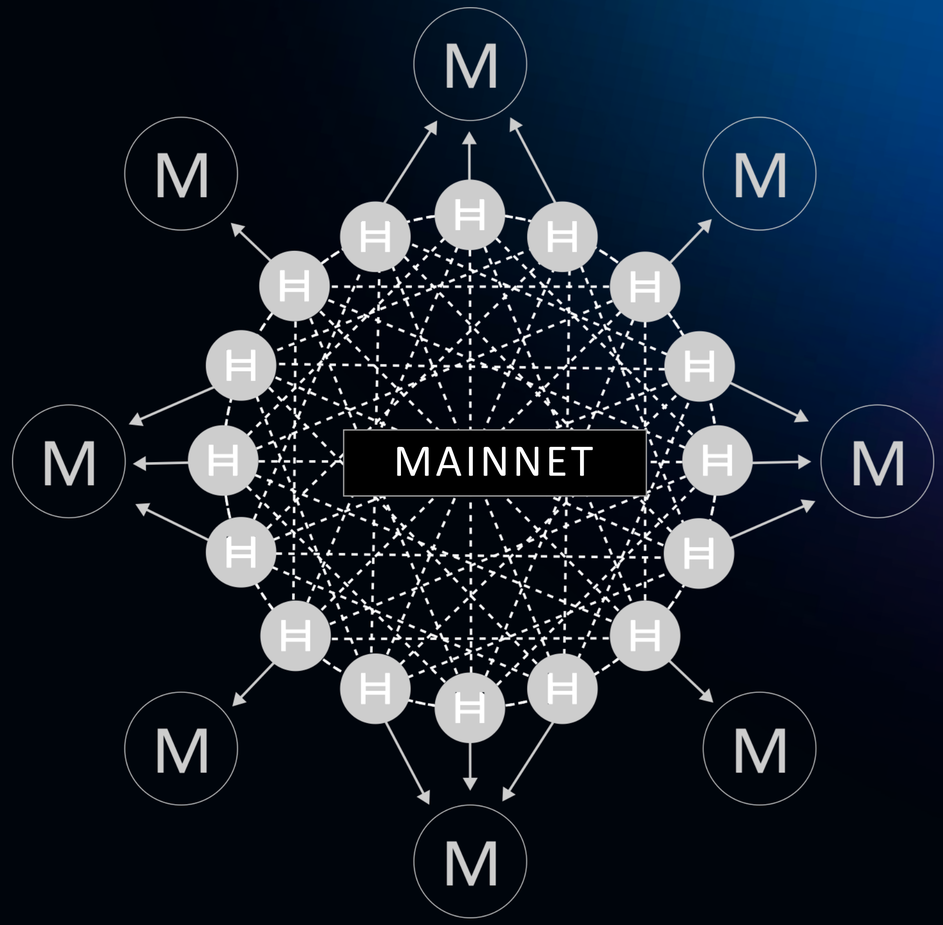


Governance

HEDERA MAINNET & MIRRORNET

MAINNET

- Can submit HAPI (Hedera API) transactions to the Hedera network
- Contributes to consensus on transactions
- Creates events on the Hedera network
- Requires HBAR cryptocurrency payment for transactions & queries



MIRRORNET

- Maintains a history of some or all the Hedera network state and ledger of transactions
- Value-added services (managed read-only node, etc.)
- Enables analytical insight into an application's state / transactions
- Publish and subscribe capabilities

hedera.com

Hedera™ Hashgraph

Cryptocurrency

“Hedera is the only platform we’ve seen that can cope with the volume of split-second transactions that need to take place.”

Jiro Olcott | Director | Power Transition

Scalable transactions

Guaranteed finality in 2-5 seconds

10,000 tps, in a single shard

Used to pay for network fees

HBAR is Hedera’s native coin

Used to pay for each network API call

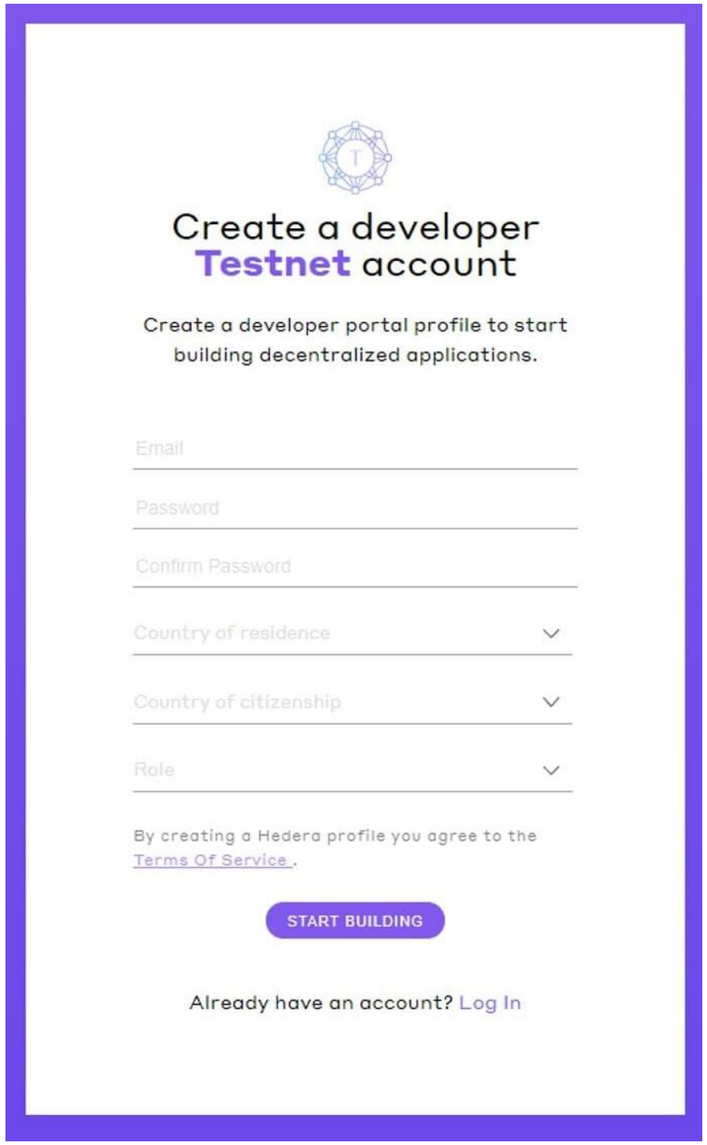
Security Mechanism

Hedera is a proof of stake network

Malicious actors would need to control 1/3rd of the total HBAR supply in order to launch an attack

Connect to a network

Get a testnet account! (https://portal.hedera.com/register)



Update .env file with account details

```
#TESTNET CREDENTIALS
OPERATOR_ID = 0.0.[redacted]
OPERATOR_PBKEY = 302a[redacted]
OPERATOR_PVKEY = 302e[redacted]



TREASURY_ID = 0.0.[redacted]
TREASURY_PBKEY = 302a[redacted]
TREASURY_PVKEY = 302e[redacted]



ALICE_ID = 0.0.[redacted]
ALICE_PBKEY = 302a[redacted]
ALICE_PVKEY = 302e[redacted]
```

portal.hedera.com
OR
docs.hedera.com/docs/mainnet

Configure accounts and client

```
// Configure accounts and client in your main function
const operatorId = AccountId.fromString(process.env.OPERATOR_ID);
const operatorKey = PrivateKey.fromString(process.env.OPERATOR_PVKEY);

const client = Client.forTestnet().setOperator(operatorId, operatorKey);
```

END USERS

INDIVIDUALS

ENTERPRISES

GOVERNMENTS

DEVICES

APPLICATION USE CASES

DATA INTEGRITY

TOKENIZATION

DECENTRALIZED IDENTITY

SUPPLY CHAIN

PAYMENTS

TOKEN SERVICE

SMART CONTRACTS

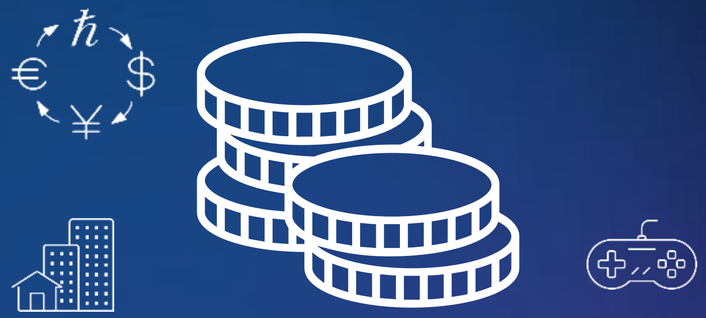
CONSENSUS SERVICE

PRIMARY NETWORK SERVICES

HEDERA GOVERNING COUNCIL

HASHGRAPH CONSENSUS

HEDERA MAINNET & MIRROR NODES



Token Service

“There are a number of challenges in using traditional ordering services for decentralized applications. By exposing this capability, Hedera is making a meaningful contribution to furthering the capabilities of distributed networks.”

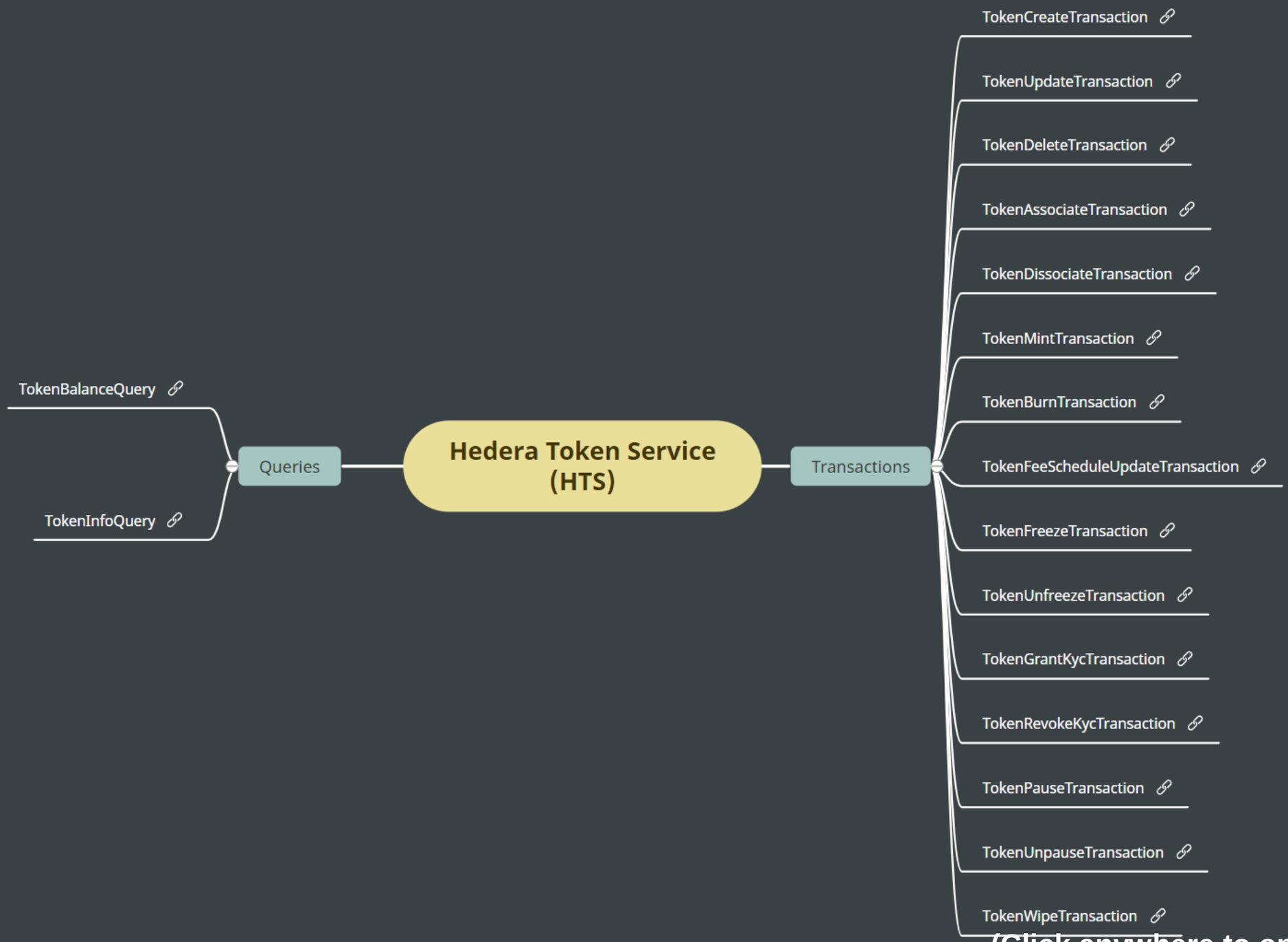
Bryan Gross | Product Manager | IBM Blockchain

Tokenization made simple

Mint and manage fungible and non-fungible tokens without needing to deploy a smart contract

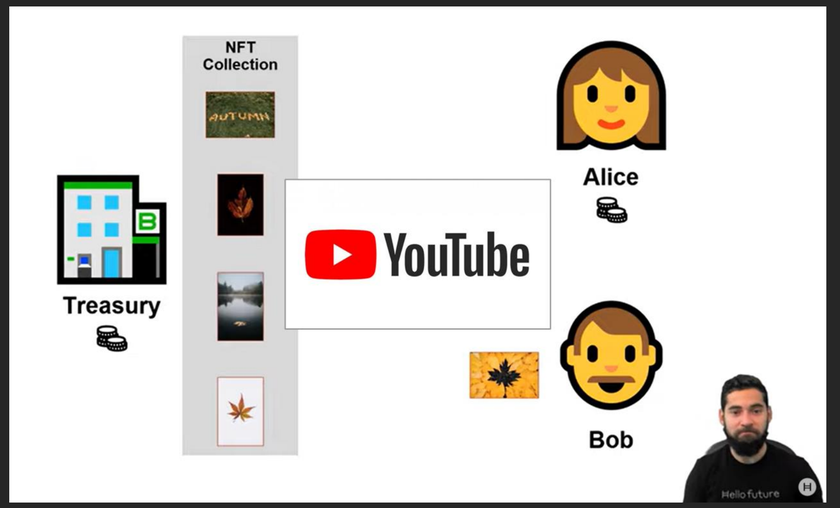
Tokenize natively

High-throughput, compliance configurations, and on-chain programmability



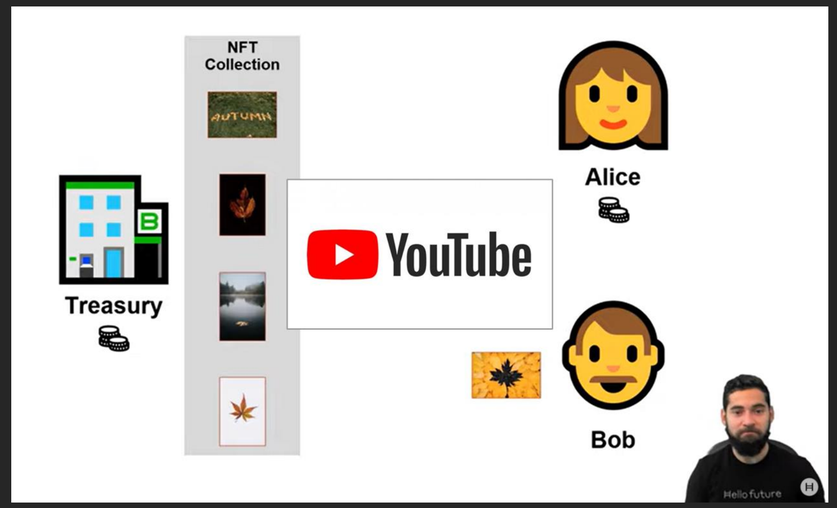
TokenCreateTransaction() | Simple Non-fungible Token (NFT)

```
const { Client, TokenCreateTransaction, TokenType, TokenSupplyType} = require("@hashgraph/sdk");
client = Client.forTestnet().setOperator(operatorId, operatorKey);

async function main() {

    const nftCreate = await new TokenCreateTransaction()
        .setTokenName("Fall Collection")
        .setTokenSymbol("LEAF")
        .setTokenType(TokenType.NonFungibleUnique)
        .setInitialSupply(0)
        .setTreasuryAccountId(treasuryId)
        .setMaxSupply(10)
        .setSupplyKey(supplyKey)
        .freezeWith(client);

    const nftCreateTxSign = await nftCreate.sign(treasuryKey);
    const nftCreateSubmit = await nftCreateTxSign.execute(client);
    const nftCreateRx = await nftCreateSubmit.getReceipt(client);
    const tokenId = nftCreateRx.tokenId;
    console.log(`Created NFT with Token ID: ${tokenId} \n`);
}
main();
```



TokenCreateTransaction() | Customized Non-fungible Token (NFT)

```
const { Client, TokenCreateTransaction, TokenType, TokenSupplyType} = require("@hashgraph/sdk");
client = Client.forTestnet().setOperator(operatorId, operatorKey);

async function main() {

    const nftCreate = await new TokenCreateTransaction()
        .setTokenName("Fall Collection")
        .setTokenSymbol("LEAF")
        .setTokenType(TokenType.NonFungibleUnique)
        .setDecimals(0)
        .setInitialSupply(0)
        .setTreasuryAccountId(treasuryId)
        .setSupplyType(TokenSupplyType.Finite)
        .setMaxSupply(CID.length)
        .setCustomFees([nftCustomFee])
        .setAdminKey(adminKey)
        .setSupplyKey(supplyKey)
        .setPauseKey(pauseKey)
        .setFreezeKey(freezeKey)
        .setWipeKey(wipeKey)
        .freezeWith(client)
        .sign(treasuryKey);

    const nftCreateTxSign = await nftCreate.sign(adminKey);
    const nftCreateSubmit = await nftCreateTxSign.execute(client);
    const nftCreateRx = await nftCreateSubmit.getReceipt(client);
    const tokenId = nftCreateRx.tokenId;
    console.log(`Created NFT with Token ID: ${tokenId} \n`);
}
main();
```



END USERS

INDIVIDUALS

ENTERPRISES

GOVERNMENTS

DEVICES

APPLICATION USE CASES

DATA INTEGRITY

TOKENIZATION

DECENTRALIZED IDENTITY

SUPPLY CHAIN

PAYMENTS

TOKEN SERVICE

SMART CONTRACTS

CONSENSUS SERVICE

PRIMARY NETWORK SERVICES

HEDERA GOVERNING COUNCIL

HASHGRAPH CONSENSUS

HEDERA MAINNET & MIRROR NODES

Hedera Smart Contracts

“the one we’ve already built a use case for

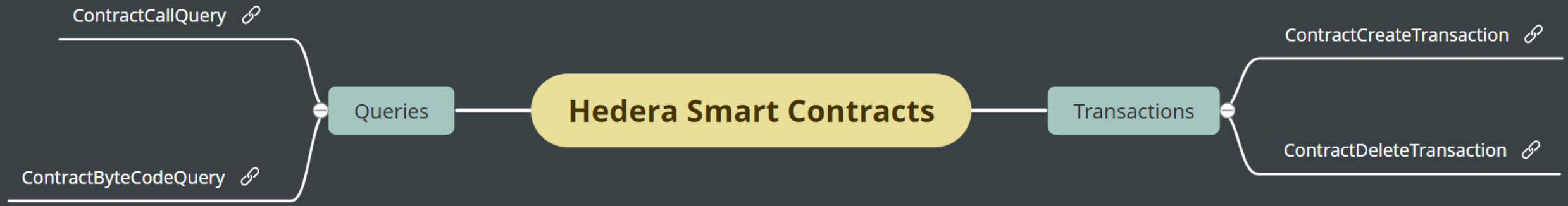
[for our clients] is around tokenization of assets. The origin of that comes from speaking with our commercial real estate clients who have large, commercial real estate assets that are difficult to deal with”

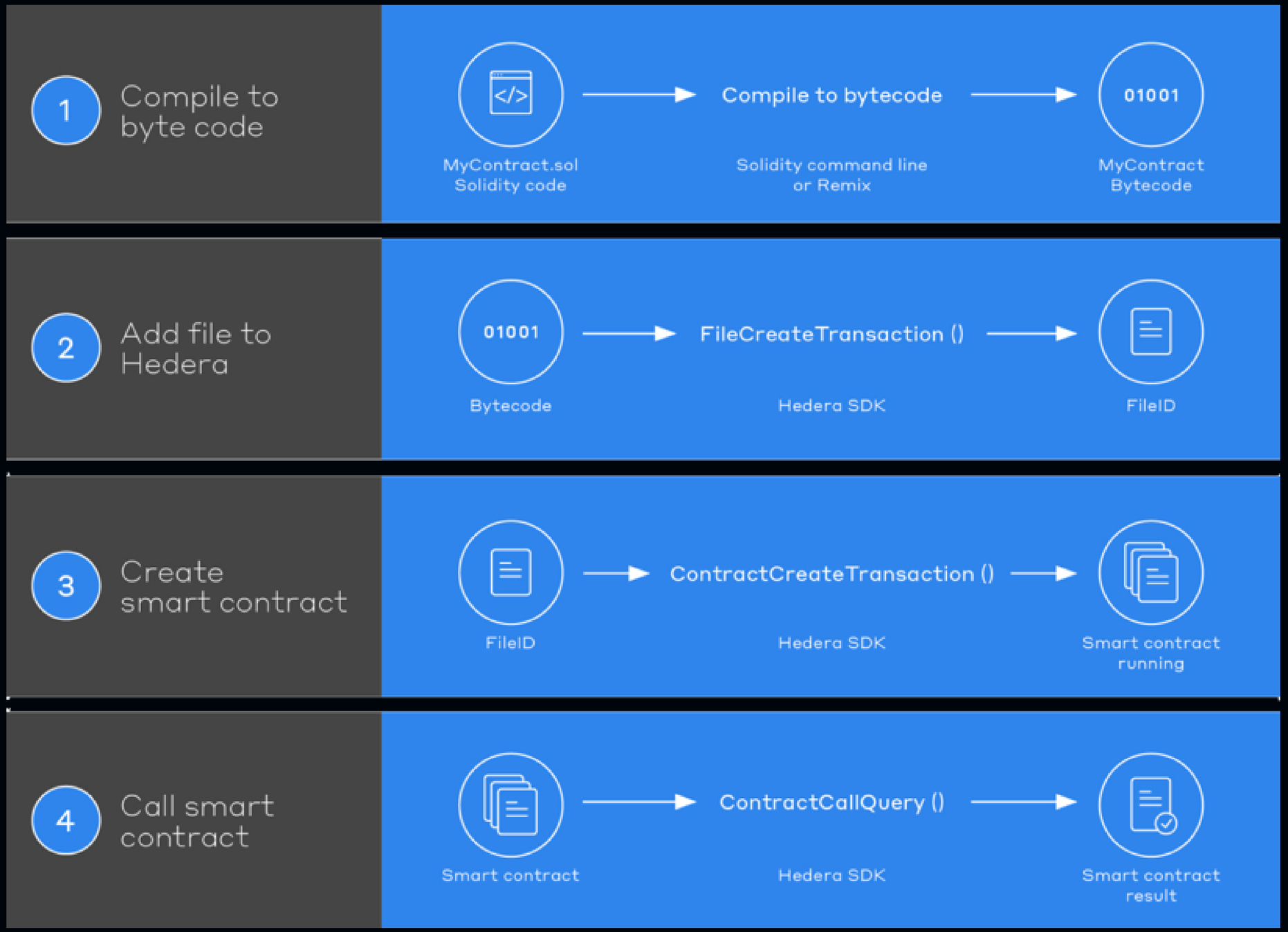
Scott Thiel | Partner & Hedera Governing Council Member | DLA Piper

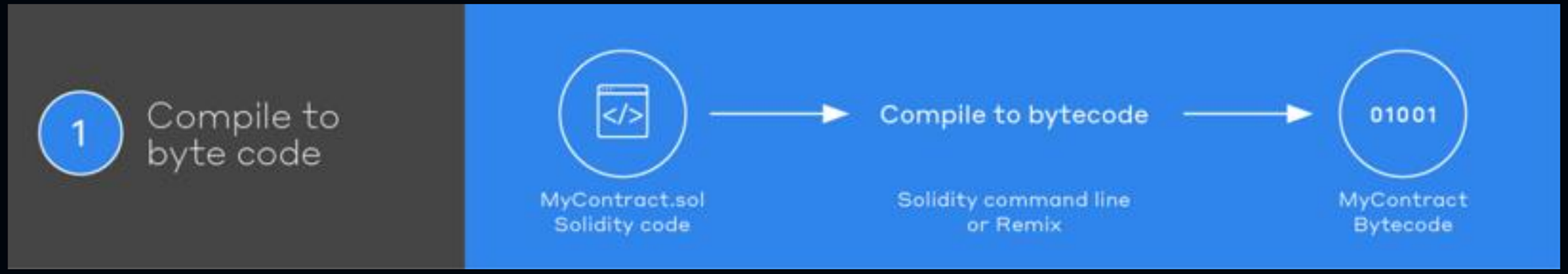
Build Using Solidity
Run smart contracts written in Solidity, unchanged and using existing standards.

Fair Transaction Ordering
Like all services, smart contracts are executed in order received, never the amount of gas paid. No need to pay extra to be included earlier in history.

Admin privileges
Transparently define administrative keys to a contract, allowing owners to make sometimes drastically needed alterations to an otherwise immutable contract…
Or not, it’s up to your implementation!




1
Compile to byte code
MyContract.sol
Solidity code
Compile to bytecode
Solidity command line or Remix
01001
MyContract Bytecode
2
Add file to Hedera
01001
Bytecode
FileCreateTransaction ()
Hedera SDK
FileID
3
Create smart contract
FileID
ContractCreateTransaction ()
Hedera SDK
Smart contract running
4
Call smart contract
Smart contract
ContractCallQuery ()
Hedera SDK
Smart contract result

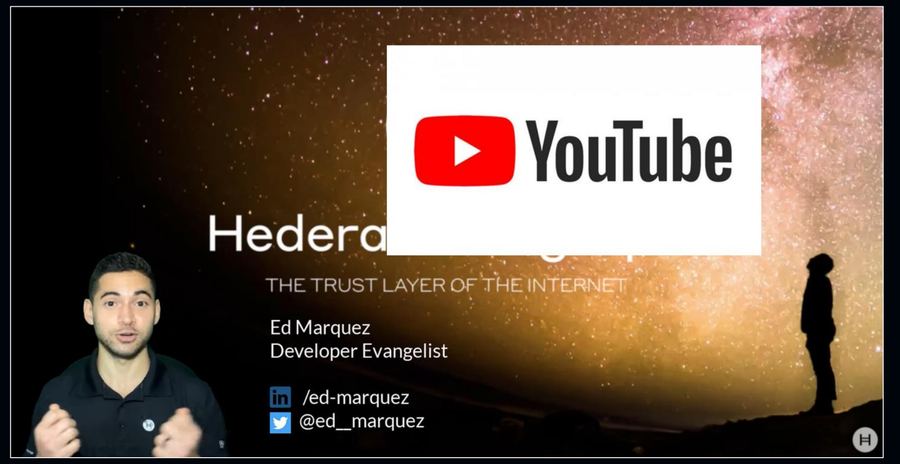


LookupContract.sol

```sol
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.7.0 <0.9.0;


 contract LookupContract{

mapping (string => uint) public myDirectory;

constructor (string memory _name, uint _mobileNumber) public {
    myDirectory[_name] = _mobileNumber;
}

function setMobileNumber(string memory _name, uint _mobileNumber) public{
    myDirectory[_name] = _mobileNumber;
}

function getMobileNumber(string memory _name) public view returns(uint){
    return myDirectory[_name];
}

 }
```

JavaScript SDK

```sol
// Import the compiled contract bytecode
const contractBytecode = fs.readFileSync("LookupContract_sol_LookupContract.bin");
```

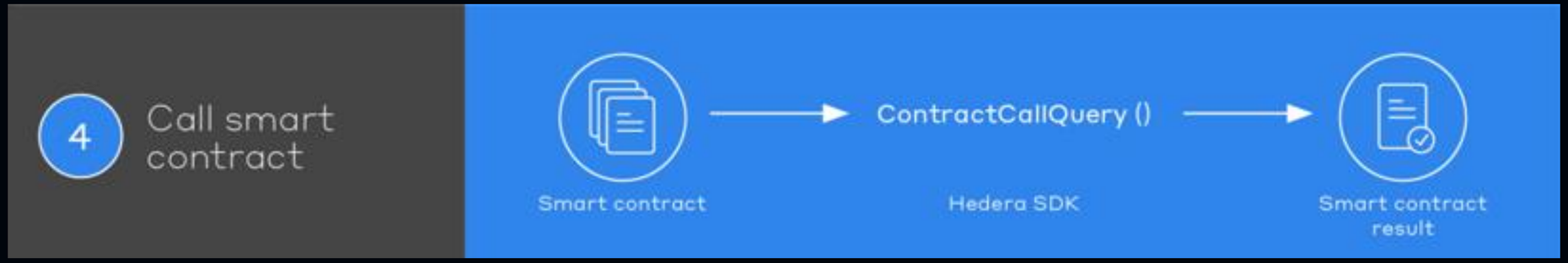


JavaScript SDK

```
// Create a file on Hedera and store the bytecode
const fileCreateTx = new FileCreateTransaction()
    .setContents(contractBytecode)
    .setKeys([operatorKey])
    .freezeWith(client);
const fileCreateSign = await fileCreateTx.sign(operatorKey);
const fileCreateSubmit = await fileCreateSign.execute(client);
const fileCreateRx = await fileCreateSubmit.getReceipt(client);
const bytecodeFileId = fileCreateRx.fileId;
console.log(`- The bytecode file ID is: ${bytecodeFileId} \n`);
```

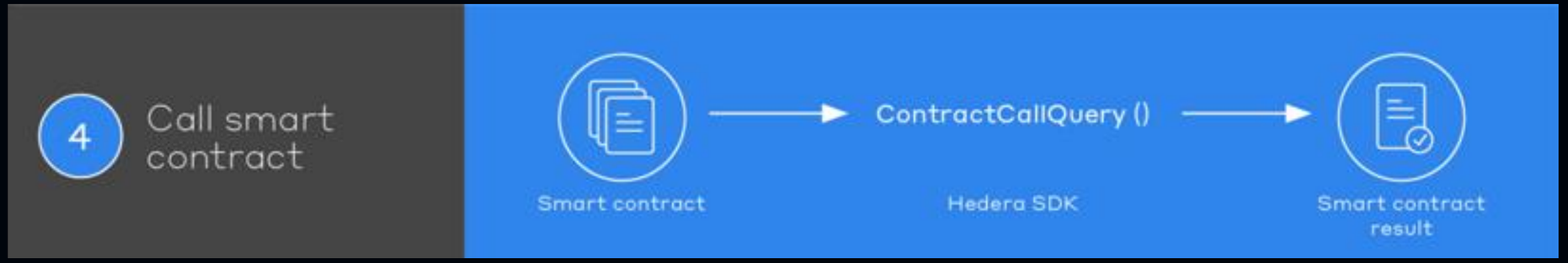
JavaScript SDK

```
// Instantiate the smart contract
const contractInstantiateTx = new ContractCreateTransaction()
    .setBytecodeFileId(bytecodeFileId)
    .setGas(100000)
    .setConstructorParameters(new ContractFunctionParameters().addString("Alice").addUint256(111111));
const contractInstantiateSubmit = await contractInstantiateTx.execute(client);
const contractInstantiateRx = await contractInstantiateSubmit.getReceipt(client);
const contractId = contractInstantiateRx.contractId;
const contractAddress = contractId.toSolidityAddress();
console.log(`- The smart contract ID is: ${contractId} \n`);
console.log(`- The smart contract ID in Solidity format is: ${contractAddress} \n`);
```



JavaScript SDK

```
// Query the contract to check changes in state variable
const contractQueryTx = new ContractCallQuery()
    .setContractId(contractId)
    .setGas(100000)
    .setFunction("getMobileNumber", new ContractFunctionParameters().addString("Alice"));
const contractQuerySubmit = await contractQueryTx.execute(client);
const contractQueryResult = contractQuerySubmit.getUint256(0);
console.log(`- Here's the phone number that you asked for: ${contractQueryResult} \n`);
```




```
// Call contract function to update the state variable
const contractExecuteTx = new ContractExecuteTransaction()
    .setContractId(contractId)
    .setGas(100000)
    .setFunction("setMobileNumber", new ContractFunctionParameters().addString("Bob").addUint256(222222));
const contractExecuteSubmit = await contractExecuteTx.execute(client);
const contractExecuteRx = await contractExecuteSubmit.getReceipt(client);
console.log(`- Contract function call status: ${contractExecuteRx.status} \n`);

// Query the contract to check changes in state variable
const contractQueryTx1 = new ContractCallQuery()
    .setContractId(contractId)
    .setGas(100000)
    .setFunction("getMobileNumber", new ContractFunctionParameters().addString("Bob"));
const contractQuerySubmit1 = await contractQueryTx1.execute(client);
const contractQueryResult1 = contractQuerySubmit1.getUint256(0);
console.log(`- Here's the phone number that you asked for: ${contractQueryResult1} \n`);
```

Console Output

```
- The bytecode file ID is: 0.0.30468526
```

2

```
- The smart contract ID is: 0.0.30468614

- The smart contract ID in Solidity format is: 0000000000000000000000000000000001d0ea06
```

3

```
- Here's the phone number that you asked for: 111111

- Contract function call status: SUCCESS

- Here's the phone number that you asked for: 222222
```

4

END USERS

INDIVIDUALS

ENTERPRISES

GOVERNMENTS

DEVICES

APPLICATION USE CASES

DATA INTEGRITY

TOKENIZATION

DECENTRALIZED IDENTITY

SUPPLY CHAIN

PAYMENTS

TOKEN SERVICE

SMART CONTRACTS

CONSENSUS SERVICE

PRIMARY NETWORK SERVICES

HEDERA GOVERNING COUNCIL

HASHGRAPH CONSENSUS

HEDERA MAINNET & MIRROR NODES

Consensus Service

“There are a number of challenges in using traditional ordering services for decentralized applications. By exposing this capability, Hedera is making a meaningful contribution to furthering the capabilities of distributed networks.”

Bryan Gross | Principal Product Manager | IBM Blockchain

Build with native performance

Create applications that demand high throughput 10,000+ txs per second, per shard

Amplify trust

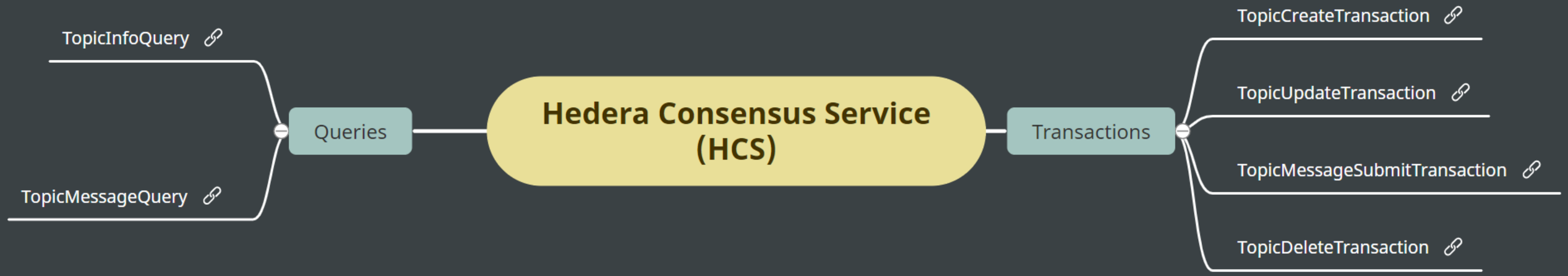
Validate the data in a multi-party database, notary service, or anything that needs transparency with shared infrastructure

Guarantee order

Results in a fair total consensus order, with accurate timestamps, a running hash of order in the topic, along with a state proof that guarantees it is correct

Integrate into existing ecosystems

Open-source integrations with Hyperledger Fabric create new architectures



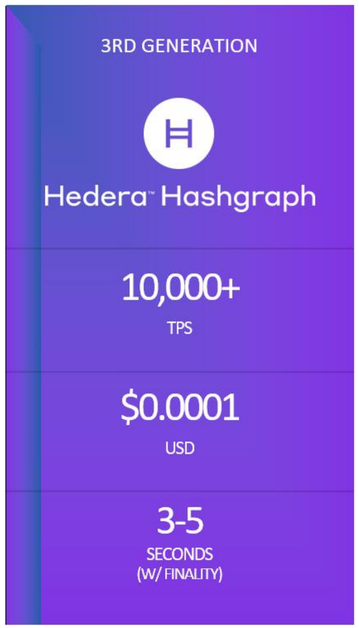
TopicCreateTransaction()

```
const { Client, TopicCreateTransaction } = require("@hashgraph/sdk");
client = Client.forTestnet().setOperator(operatorId, operatorKey);

async function main() {

    tx = await new TopicCreateTransaction().execute(client);
    receipt = await tx.getReceipt(client);
    newTopicId = receipt.topicId;

    console.log(`New HCS topic ID: ${newTopicId}`);
}
void main();
```

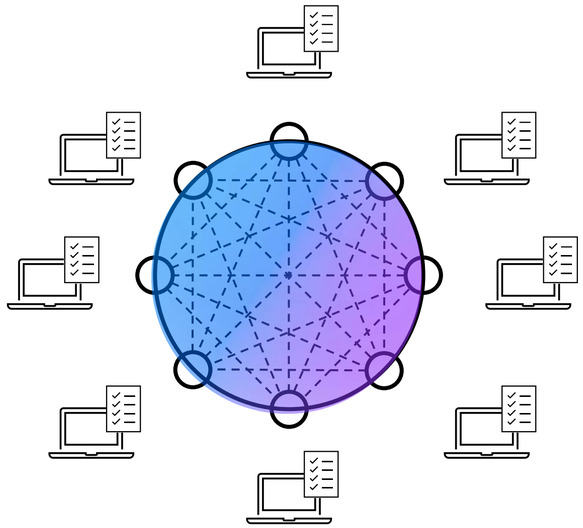
TopicMessageQuery()

```
new TopicMessageQuery()
    .setTopicId(topicId)
    .subscribe(
        client,
        (error) => console.log(`Error: ${error}`),
        (message) => console.log(message.toString())
    );
```

In this session, you will learn how the **Hedera network functions**, how you can **start development**, and **where to learn more**



Introduction



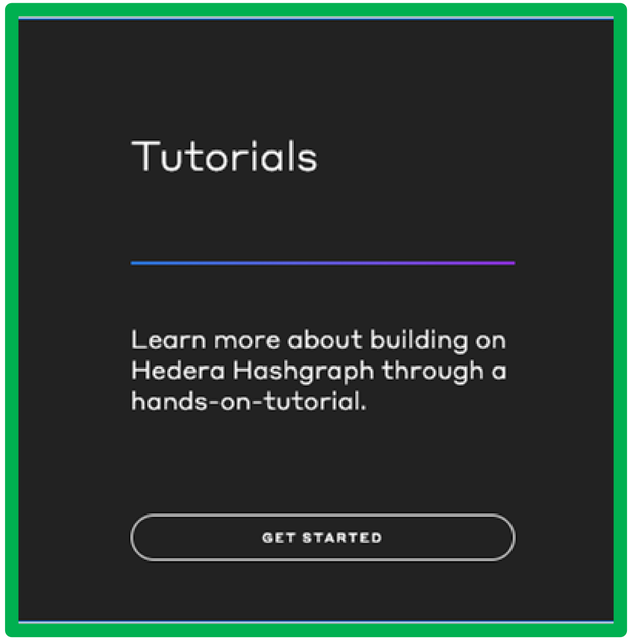
Understand
the Hedera Network

CONSENSUS SERVICE

TOKEN SERVICE

SMART CONTRACTS

Start Developing on
Hedera



Get Resources and
Learn More

Considering use cases

What makes a good decentralized use case?

- Transparent collaboration
- Auditability and attestability
- Censorship resistance
- Composability & compounding
- Inherent lack of trust between parties

BANKS

Standard Bank
NOMURA
Shinhan Bank
DBS

PAYMENTS INFRASTRUCTURE

eftpos Good for Australia
ZIMBALI
FIS
EMTECH

STABLE COINS

Shinhan Bank
eftpos Good for Australia
Standard Bank
USD Coin

MICROPAYMENTS

dropp
Meeco
eftpos Good for Australia
tune.fm

CBDC

UCL
EMTECH
ZIMBALI

WALLETS

MyHbarWallet
coinomi
Meeco

SECURITY TOKENS ISSUERS & EXCHANGES

DLA PIPER
TOKO
Standard Bank
LCX

DEFI & NEOBANKS

Chainlink
akt.
AllianceBlock

SYSTEMS INTEGRATORS

BCW
KrypC

CUSTODY

Hex Trust
BitGo
GK8

TOOLS

armanino
TRM
Google Cloud
Trust Enterprises REST API

KABUTO

RETAIL | DATA INTEGRITY

THE COUPON BUREAU

Develops and maintains the Universal Positive Offer File for manufacturer coupons using the Hedera Consensus Service.

"Only Hedera Hashgraph was able to provide the real-time, tamper-proof logging capabilities that we needed to bring transparency, trust, agnosticism and industry oversight to a platform that connects all coupon industry stakeholders"

BRANDI JOHNSON | CEO

hedera.com

QUICK FACTS

CHALLENGE

Coupon and associated promotional data is fragmented, non-standardized, and rife with fraud. Without the necessary controls, transparency or trust, the industry usage of this promotional vehicle has decreased in recent years.

SOLUTION

The Coupon Bureau can bring its standardization and added efficiencies to the marketplace with transparency and trust. By allowing 3rd party validation and audit capabilities among authorized stakeholders, the industry will be able to have confidence in this centralized, agnostic solution.

MORE INFORMATION

HEDERA.COM/USERS/COUPON-BUREAU

FINANCE | PAYMENTS

eftpos

Micropayments that aims to create a seamless Australian payment experience for web users by providing an alternative to traditional online paywalls and subscriptions.

“By working with Hedera, we are leveraging next generation payments infrastructure technology that can support Australian dollar-based micropayments and open up entirely new ways of conducting business online.”

ROBERT ALLEN | ENTREPENEUR IN RESIDENCE

QUICK FACTS

CHALLENGE

Eftpos needed to facilitate payments and micropayments using a digital dollar that enables fast, secure, and affordable micropayment transactions.

SOLUTION

Eftpos will test the capability of a digital Australian dollar stable coin, enabled by the hedera consensus service, allowing consumers to load a wallet with a few dollars and then pay for web-based content seamlessly.

ADVERTISING | DATA INTEGRITY

AdsDaX

Real-time auditing at scale to eliminate advertising fraud and remove costly intermediaries.

"Hedera has proven it has the scale, speed and reliability to handle all of the live advert tracking data and programmatic event data that we have been handling in recent campaigns."

IAN MULLINS | FOUNDER & CEO

QUICK FACTS

CHALLENGE

The digital advertising industry is rife with fraud and trust issues, while over 75% of fees are placed into the pockets of expensive intermediaries.

SOLUTION

AdsDax is leveraging Hedera to build a platform that will track and verify advertising events and engagement, while providing security, resilience, scalability, and transparency, without the need for costly intermediaries.

MORE INFORMATION

HEDERA.COM/USERS/ADSDAX

INTERNET OF THINGS | SUPPLY CHAIN TRACK & TRACE

EVERYWARE

Digitally track and monitor critical or high-value assets.

“For managing highly critical information, like the COVID-19 vaccine’s temperature, Hedera Hashgraph provides us immutable trust at the speed required for our IoT devices.”

TOM SCREEN | TECHNICAL DIRECTOR

QUICK FACTS

CHALLENGE

Monitoring the temperature of key assets, like the COVID-19 vaccine is of interest to multiple parties, with liability and risk to be had. Everyware wanted a more secure method to manage this device data than a traditional database.

SOLUTION

Everyware uses Hedera Consensus Service to keep an immutable, tamper-proof, and verifiable record of sensor data.

MORE INFORMATION

HEDERA.COM/USERS/EVERYWARE

ENERGY & SUSTAINABILITY | PAYMENTS

power transition

Peer-to-peer energy microgrids built on a stable network with high-throughput, to exchange value.

"The Hedera Token Service has transformed the way our customers can trade electricity and other forms of value across the energy sector."

JIRO OLCOTT | DIRECTOR

hedera.com

QUICK FACTS

CHALLENGE

Peer-to-peer energy trading and microgrid management is becoming a reality with the increased adoption of solar panels and batteries. Power Transition needed a way to enable economically feasible micropayments between disparate parties who require trust.

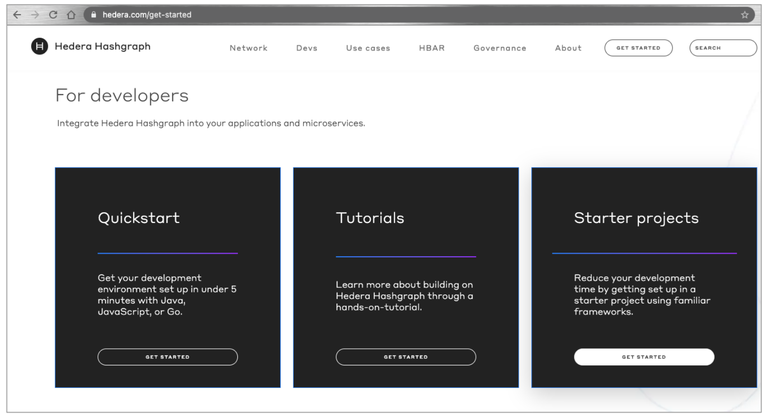
SOLUTION

Hedera was chosen for its network stability, high-throughput, low latency, and predictable fees to power HBAR cryptocurrency micropayments between homes on the Power Transition microgrid.

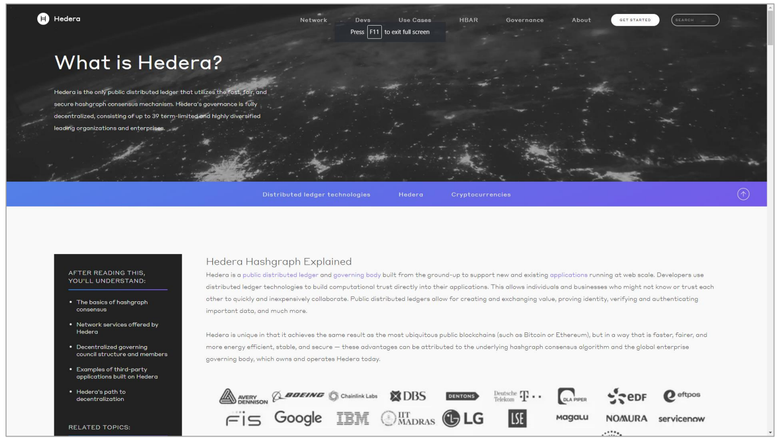
MORE INFORMATION

HEDERA.COM/USERS/POWER-TRANSITION

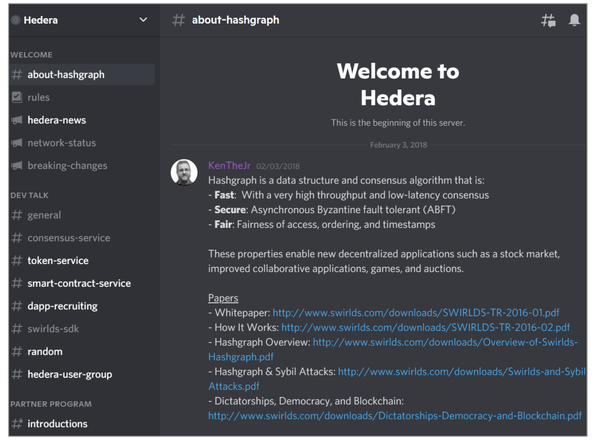
Learn more and start building web 3!



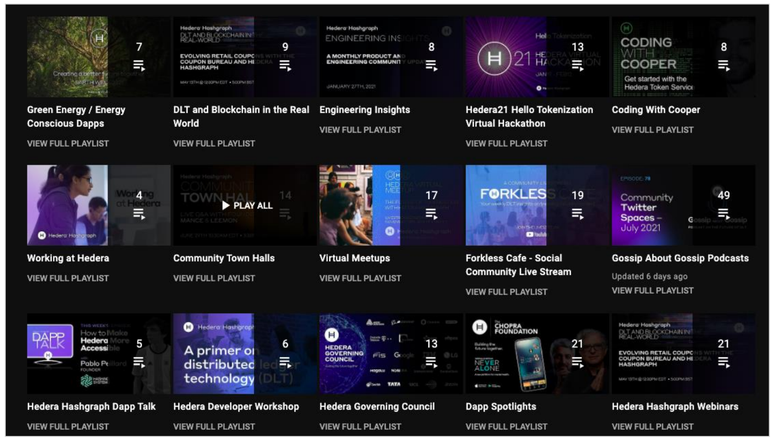
hedera.com/get-started



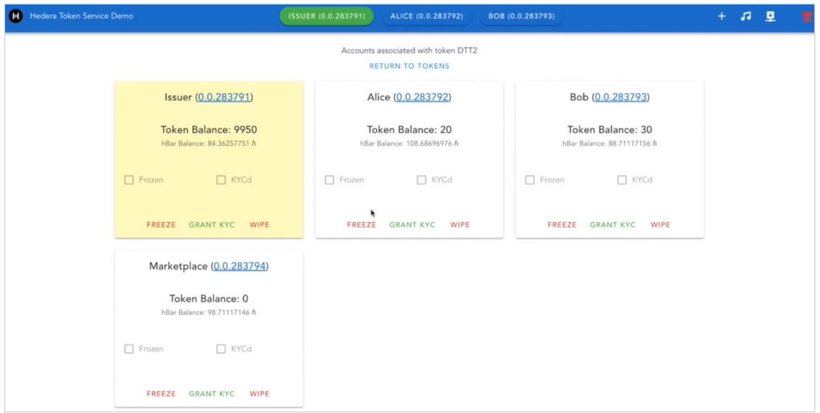
Hedera Learning Center



Join Developer Discord



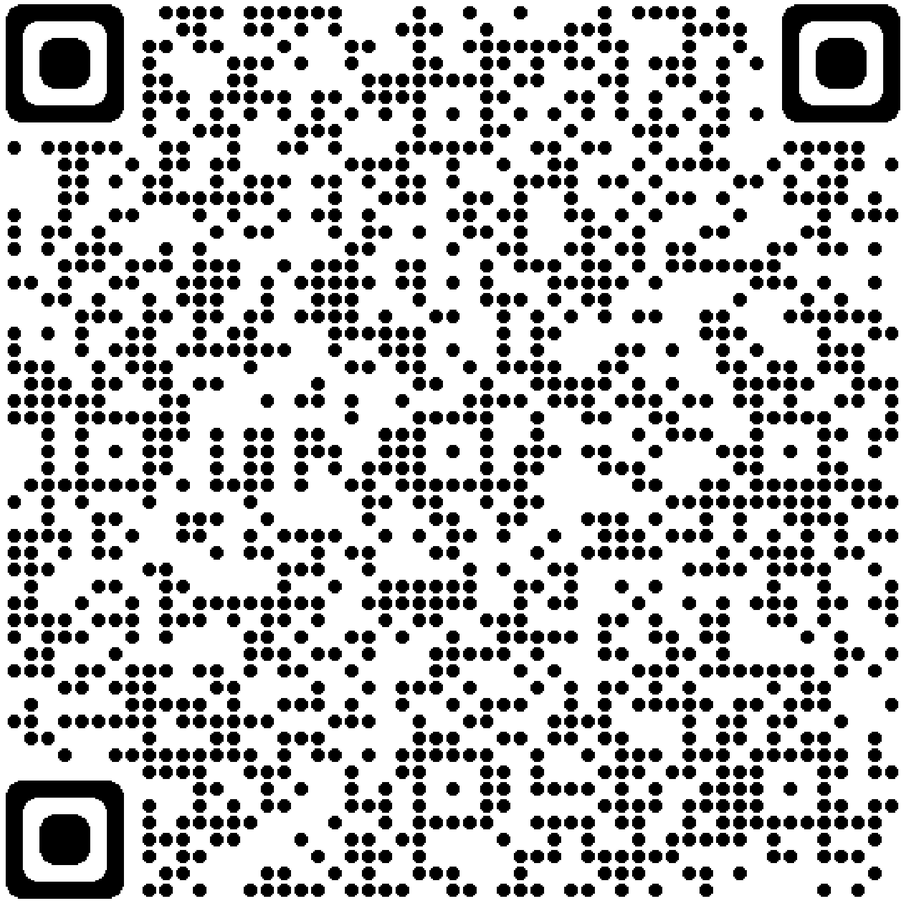
Hedera YouTube Channel



Application Demos

Web 3 is ready for you, and with Hedera, **you are ready for Web 3!**

GET THE SLIDES NOW ➔



Ed Marquez – Hedera

Viv Diwakar – The HBAR Foundation

/ed-marquez | /vivdiwakar

@ed_marquez | @HBAR_foundation