Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 1 of 117 PageID #: 63

# Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 2 of 117 PageID #: 64

Claim Chart U.S. Patent No. 11,196,566 (the "'566 Patent") Coinbase	
<u>Claim 1</u>	<u>Coinbase Products &amp; Services</u> Payment of block rewards to new Validator Nodes under Proof of Stake
A computing device for processing a transaction between a first client device, and a second client device via a transfer mechanism, the transfer mechanism comprising a decentralized digital currency, the computing device comprising:	<ul> <li>The Computing Device   Facilitator includes:</li> <li>the Coinbase (Owned, managed) Ethereum Validator Full Nodes; and</li> <li>the Coinbase (Owned, managed) Ethereum supporting Archive Nodes and Light Nodes; and</li> </ul> Client Device The First Client is an <u>active</u> Coinbase (Owned, managed) Ethereum Validator Full Nodes. The Computing Device and the First Client are the same device.
	The Second Client is a <u>prospective</u> Validator Node, which will include any prospective Coinbase (Owned, managed) Ethereum Validator Nodes.
	The Coinbase (Owned, managed) Ethereum Validator Nodes facilitates value transfer to newly activated Validator Nodes (as Second Client) based on work performed in producing blocks securely. The Beacon Chain upgrade brings proof-of-stake consensus to Ethereum. For this, active participants - known as validators – are required to propose, verify, and vouch for the validity of blocks.
	The Patent allows for, and the Claims do not prevent, the Computing Device from being, or including, the First Client or Second Client. This is detailed in the Patent description [0055].
	FIG. 1 (see Figure 16) depicts a typical embodiment for practicing the invention—especially for use with a distributed transfer mechanism—where the clients, transfer mechanism, facilitator, and data source are distinct participants. However, the depicted arrangement is not the only one contemplated by the invention. In an alternate embodiment, the facilitator provides some or all aspects of the transfer mechanism. In another embodiment, the facilitator comprises some or all aspects of a client. For example, part or all of a client's data store, the ability to initiate or accept offers, etc., could be "embedded" in the facilitator, thereby enabling the facilitator to operate as a client itself (e.g., one controlled by the owners of the facilitator). In yet another embodiment, the facilitator comprises the data source. Many configurations are contemplated by the invention are possible, and will become apparent to one skilled in the art.

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 3 of 117 PageID #: 65

	$\begin{array}{c c} \hline F \rightarrow \varphi \gamma \rightarrow \chi \\ \hline Data \\ Source \\ \hline 130 \\ \hline 130 \\ \hline 120 \\ \hline 140 \\ \hline 140 \\ \hline 140 \\ \hline 120 \\ \hline 140 \\ \hline 140 \\ \hline 140 \\ \hline 150 \\ \hline 1$
• a memory for storing a first asymmetric key pair, the first asymmetric key pair comprising a first private key and a first public key;	<ul> <li>The First Client is an <u>active</u> Coinbase (Owned, managed) Ethereum Validator Full Nodes. The Computing Device and the First Client are the same device.</li> <li>These consist of a computer hardware/software combination to run, namely: <ul> <li>Memory (RAM), used in the computing device (such as a computer, server or server cloud instance).</li> <li>Transaction record sector (stores transactions and data that haven't been submitted to the blockchain yet)</li> <li>a first key pair sector which is generated and stored on the device (typically)</li> <li>The asymmetric key pair generated and/or stored consists of a first private key and a first public key and is stored on the device (typically)</li> </ul> </li> </ul>
	Recommended hardware requirements for running a node. <u>https://launchpad.ethereum.org/en/checklist</u> A description of the Validator Node's keys is described in the following link. <u>https://kb.beaconcha.in/ethereum-2-keys</u>
	<ul> <li>The validator signing key consists of two elements:</li> <li>1. Validator private key</li> <li>2. Validator public key</li> <li>The purpose of the validator private key is to actively sign on-chain (ETH2) operations such as block proposals and attestations. Therefore, these keys have to be held in a hot wallet.</li> </ul>

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 4 of 117 PageID #: 66

		Example Validator Client (Prysmatic) installation guide shows key management.
		https://docs.prylabs.network/docs/install/install-with-script/
•	a network interface for receiving terms, the terms comprising:	The First Client is an <u>active</u> Coinbase (Owned, managed) Ethereum Validator Full Nodes. The Computing Device and the First Client are the same device.
		The Ethereum Network requires network connectivity in order to achieve PoS consensus.
		https://launchpad.ethereum.org/en/checklist
		https://github.com/ethereum/consensus-
		specs/blob/dev/specs/phase0/validator.md#block-proposal
		https://github.com/ethereum/consensus-
		specs/blob/dev/specs/phase0/validator.md#attesting
		https://github.com/ethereum/consensus-
		specs/blob/dev/specs/phase0/validator.md#broadcast-attestation
		https://github.com/ethereum/consensus- specs/blob/dev/specs/phase0/validator.md#broadcast-aggregate
		spees/0100/dev/spees/phase0/vandatof.ind#010adeast-aggregate
0	at least one of a first principal	First principle data
0	data or a second principal	The initial deposit staking amount to participate as an Eth2 Validator in
	data;	the Ethereum Network.
		The Beacon Chain upgrade brings proof-of-stake consensus to Ethereum. For this, we need active participants - known as validators to propose, verify, and vouch for the validity of blocks. In exchange honest validators receive financial rewards. Importantly, as a validator you'll need to post ETH as collateral - in other words, have some fund at stake. The only way to become a validator is to make a one-way ET transaction to the deposit contract on the current Ethereum chain.
		https://launchpad.ethereum.org/en/overview
		https://github.com/ethereum/consensus-
		specs/blob/dev/specs/phase0/beacon-chain.md#deposits
	a volovovo to at 1t f	First data source
0	a reference to at least one of a first data source or a second	<b>First data source</b> The Beacon Chain reward and penalty algorithm.
	data source; and	The Deacon chain reward and penalty algorithm.
		https://consensys.net/blog/codefi/rewards-and-penalties-on-ethereum-
		<u>20-phase-0/</u>
		https://kb.beaconcha.in/rewards-and-penalties#block-reward
		https://github.com/ethereum/consensus-
		specs/blob/dev/specs/phase0/beacon-chain.md#attestations
0	an expiration timestamp;	At the beginning of each epoch (every 32 slots, except GENESIS), several things happen, including

	<ul> <li>Justification and finalization of the chain</li> <li>Assignment of rewards and penalties to attesters</li> <li>Update of the validator registry</li> <li>The special slashing penalty (see above), and</li> <li>Some final updates (computing effective balances, resets, etc)</li> <li>https://consensys.net/blog/codefi/rewards-and-penalties-on-ethereum-20-phase-0/</li> <li>https://github.com/ethereum/consensus-specs/blob/dev/specs/phase0/beacon-chain.md#epoch-processing</li> </ul>
• a computer processor coupled to the memory and the network interface, the computer processor configured to:	The First Client is an <u>active</u> Coinbase (Owned, managed) Ethereum Validator Full Nodes. The Computing Device and the First Client are the same device. These consist of a computer hardware/software combination to run.
• read the first private key from the memory;	The active Coinbase (Owned, managed) Ethereum Validator Full Nodes is expected to propose a SignedBeaconBlock at the beginning of any slot during which is_proposer(state, validator_index) returns True. To propose, the validator selects the BeaconBlock, parent, that in their view of the fork choice is the head of the chain during slot - 1. The validator creates, signs, and broadcasts a block that is a child of parent that satisfies a valid beacon chain state transition. <u>https://github.com/ethereum/consensus-</u>
	specs/blob/dev/specs/phase0/validator.md#block-proposal <b>Preparing for a BeaconBlock</b> To construct a BeaconBlockBody, a block (BeaconBlock) is defined with the necessary context for a block proposal:
	Slot Set block.slot = slot where slot is the current slot at which the validator has been selected to propose. The parent selected must satisfy that parent.slot < block.slot. Note: There might be "skipped" slots between the parent and block. These skipped slots are processed in the state transition function without per-block processing.
	<b>Proposer index</b> Set block.proposer_index = validator_index where validator_index is the validator chosen to propose at this slot. The private key mapping to state.validators[validator_index].pubkey is used to sign the block.
	<b>BLS public key</b> Validator public keys are <u>G1 points</u> on the <u>BLS12-381 curve</u> . A private key, privkey, must be securely generated along with the resultant pubkey. This privkey must be "hot", that is, constantly available to sign data throughout the lifetime of the validator.

	https://github.com/ethereum/consensus- specs/blob/dev/specs/phase0/validator.md#bls-public-key
• compute a first cryptographic signature from the first private key;	The active Coinbase (Owned, managed) Ethereum Validator Full Nodes is expected to propose a SignedBeaconBlock at the beginning of any slot during which is_proposer(state, validator_index) returns True. To propose, the validator selects the BeaconBlock, parent, that in their view of the fork choice is the head of the chain during slot - 1. The validator creates, signs, and broadcasts a block that is a child of parent that satisfies a valid beacon chain state transition.
	https://github.com/ethereum/consensus- specs/blob/dev/specs/phase0/validator.md#block-proposal
	<b>Preparing for a BeaconBlock</b> To construct a BeaconBlockBody, a block (BeaconBlock) is defined with the necessary context for a block proposal:
	Slot Set block.slot = slot where slot is the current slot at which the validator has been selected to propose. The parent selected must satisfy that parent.slot < block.slot. Note: There might be "skipped" slots between the parent and block. These skipped slots are processed in the state transition function without per-block processing.
	<b>Proposer index</b> Set block.proposer_index = validator_index where validator_index is the validator chosen to propose at this slot. The private key mapping to state.validators[validator_index].pubkey is used to sign the block.
	Signature signed_block = <pre>SignedBeaconBlock(message=block, signature=block_signature), where block_signature is obtain</pre>
	<pre>def get_block_signature(state: BeaconState, block: BeaconBlock, privkey: int) -&gt; BLSSignature: domain = get_domain(state, DOMAIN_BEACON_PROPOSER, compute_epoch_at_slot(block.slot)) signing_root = compute_signing_root(block, domain) return bls.Sign(privkey, signing_root)</pre>
• create an inchoate data record comprising:	The SignedBeaconBlock, as produced by the Validator (as the block producer), containing the deposit of a prospective Validator that is yet to be added to the list of registered Validators.
• a commit input for receiving a commit data from a commit transaction;	The 32ETH deposit stake to instantiate an Eth2 Validator.
• one or more output data obtained from at least one of the first principal data or the second principal data	The prospective Validator's reward or penalty. This is implied as these are rules of the network.
the second principal data, and a value data from at least one of the first data	The Beacon Chain reward and penalty algorithm uses predefined values and formulas, along with the effective staking balance of the participating nodes to calculate the reward or penalty for the Validators.

source or the second data source; and	https://consensys.net/blog/codefi/rewards-and-penalties-on-ethereum- 20-phase-0/
	https://kb.beaconcha.in/rewards-and-penalties#block-reward
	https://github.com/ethereum/consensus- specs/blob/dev/specs/phase0/beacon-chain.md#attestations
• the first cryptographic signature; and	The Validator (as the block producer) is expected to propose a SignedBeaconBlock at the beginning of any slot during which is_proposer(state, validator_index) returns True. To propose, the validator selects the BeaconBlock, parent, that in their view of the fork choice is the head of the chain during slot - 1. The validator creates, signs, and broadcasts a block that is a child of parent that satisfies a valid beacon chain state transition.
	https://github.com/ethereum/consensus- specs/blob/dev/specs/phase0/validator.md#block-proposal
	Signature signed_block = SignedBeaconBlock(message=block, signature=block_signature), where block_signature is obtai
	<pre>def get_block_signature(state: BeaconState, block: BeaconBlock, privkey: int) -&gt; BLSSignature: domain = get_domain(state, DOMAIN_BEACON_PROPOSER, compute_epoch_at_slot(block.slot)) signing_root = compute_signing_root(block, domain) return bls.Sign(privkey, signing_root)</pre>
	Note that the signed block includes the Eth1 data containing the 32ETH deposit stake to instantiate a Validator.
	Eth1 Data The block.body.eth1_data field is for block proposers to vote on recent Eth1 data. This recent data contains an Eth1 block hash as well as the associated deposit root (as calculated by the get_deposit_root() method of the deposit contract) and deposit count after execution of the corresponding Eth1 block. If over half of the block proposers in the current Eth1 voting period vote for the same eth1_data then state.eth1_data updates immediately allowing new deposits to be processed. Each deposit in block.body.deposits must verify against state.eth1_data.eth1_deposit_root.
	https://github.com/ethereum/consensus- specs/blob/dev/specs/phase0/validator.md#eth1-data
• publish the inchoate data record to at least one of the first client device or the second client device,	As the Computing Device and the First Client are the same device (Coinbase (Owned, managed) Ethereum Validator Node), the inchoate data record is already known by the First Client. As such the inchoate data record is considered to be published to the First Client on creation of the inchoate data record by the Validator Node.
	A validator is expected to propose a <u>SignedBeaconBlock</u> at the beginning of any slot during which is _proposer(state, validator_index) returns True. To propose, the validator selects the BeaconBlock, parent, that in their view of the fork choice is the head of the chain

	Proposer index
wherein the inchoate data record is used by at least one of the first client device or the second client device to create a complete data record and to create the transaction by broadcasting the complete data record for transmitting and receiving among network participants in the computer network for recording in the distributed ledger,	As the Computing Device and the First Client are the same device (Coinbase (Owned, managed) Ethereum Validator Node), the inchoate data record is already signed by the First Client. The signed inchoate data record (as a complete data record) is broadcast and recorded on the Ethereum Network. The SignedBeaconBlock, in order to achieve consensus, is required to be signed by the Ethereum Network as the First Client (the Eth2 Validators) as either the Block Producer, Attester or Aggregator's.
transaction between the first client device and the second client device without the need for a trusted central authority, wherein at least one of the first client device or the second client device signs the inchoate data record and saves a copy of the inchoate data record on at least one of the first client device or the second client device; and	<ul> <li>authority.</li> <li>From the Ethereum yellow paper.</li> <li>2.1. Value. In order to incentivise computation within the network, there needs to be an agreed method for transmitting value. To address this issue, Ethereum has an intrinsic currency, Ether, known also as ETH and sometimes referred to by the Old English <sup>-</sup>D.</li> <li>https://ethereum.github.io/yellowpaper/paper.pdf</li> <li>As the Computing Device and the First Client are the same device (Coinbase (Owned, managed) Ethereum Validator Node), the inchoate data record is already signed by the First Client. The signed inchoate data record (as a complete data record) is broadcast and recorded on the Ethereum Network.</li> <li>A validator is expected to propose a <u>SignedBeaconBlock</u> at the beginning of any slot during which is _proposer(state, validator_index) returns True. To propose, the validator selects the BeaconBlock, parent, that in their view of the fork choice is the head of the chain during slot - 1. The validator creates, signs, and broadcasts a block that is a child of parent that satisfies a valid <u>beacon chain state</u> transition.</li> <li>https://github.com/ethereum/consensus-specs/blob/dev/specs/phase0/validator.md#block-proposal</li> </ul>
wherein the decentralized digital currency comprises a distributed ledger that enables processing the	that is a child of parent that satisfies a valid <u>beacon chain state</u> <u>transition</u> . <u>https://github.com/ethereum/consensus-</u> <u>specs/blob/dev/specs/phase0/validator.md#block-proposal</u> Ether is used as the decentralised currency. The Ethereum network maintains a distributed ledger without the need for a trusted central

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 9 of 117 PageID #: 71

Set block.proposer_index = validator_index where validator_index is
the validator chosen to propose at this slot. The private key mapping to
state.validators[validator_index].pubkey is used to sign the block.
https://github.com/ethereum/consensus-
specs/blob/dev/specs/phase0/validator.md#proposer-index
Attesting
A validator is expected to create, sign, and broadcast an attestation
during each epoch. The committee, assigned index, and assigned slot
for which the validator performs this role during an epoch are defined
by get committee assignment(state, epoch, validator index).
https://github.com/ethereum/consensus-
specs/blob/dev/specs/phase0/validator.md#attesting
Construct aggregate
<b>Construct aggregate</b>
If the validator is selected to aggregate (is_aggregator()), they construct an aggregate attentation via the following
construct an aggregate attestation via the following. Collect attestations seen via gossip during the slot that have an
equivalent attestation data to that constructed by the validator. If
len(attestations) > 0, create an aggregate attestation: Attestation with
the following fields.
ine jouowing fields.
https://github.com/ethereum/consensus-
specs/blob/dev/specs/phase0/validator.md#construct-attestation
def get_aggregate_signature(attestations: Sequence[Attestation]) ->
BLSSignature:
signatures = [attestation.signature for attestation in attestations]
return bls.Aggregate(signatures)
The Coinbase (Owned, managed) Ethereum Validator Node uses the
SignedBeaconBlock (as a complete data record) along with the
broadcasted Aggregated Attestations to update its record of the Beacon
Chain. Finality is achieved if two epochs in a row are justified.
ETH2 uses Casper Proof of Stake, specifically, something called a
"finality gadget". The ETH2 finality process is defined as follows:
1. If $> 2/3rds$ of validators vote correctly on the chain head
during an epoch, we call the last epoch justified
2. <i>If two epochs in a row are justified, the current_epoch - 2 is</i>
considered <b>finalized</b>
https://hackmd.io/@prysmaticlabs/finality#How-Finality-Works-in-
ETH2
The Coinbase (Owned, managed) Ethereum Validator Node as the First
Client process the Beacon Block and records the new Validator in the

# Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 10 of 117 PageID #: 72

	Registry on identifying a new deposit in the Deposit Contract. The Beacon Block is saved by each Eth2 Client. <a href="https://github.com/ethereum/consensus-">https://github.com/ethereum/consensus-</a>
wherein the at least one of the	specs/blob/dev/specs/phase0/beacon-chain.md#deposits         The new Validator Node, as the Second Client, can view the status of
computing device, the first client device, or the second client device verifies the recording of the complete	its activation from <u>https://beaconcha.in/</u>
data record in the distributed ledger by observing an external state.	Deposited Pending Active Exited
	This validator has been processed by the beacon chain and is currently waiting activated. It will approximately be activated on <b>Feb 3</b> , <b>2021</b> , <b>6</b> : <b>21 AM</b> during epoch Make sure your nodes and your client is up and running <i>before</i> the countdown rezero.
	18 days 11 hr 45 min 25 sec
	Once a deposit is made to the Ethereum Beacon Chain, it will join a waiting queue before joining the network.
	All Ethereum 2.0 deposits have two delays before going into the waiting queue:
	<ol> <li>Eth1 data inclusion delay: The Beacon Chain follows Eth1 with some delay to make sure the Eth1 data is finalized.</li> <li>Eth1 data voting delay: The Beacon Chain validators vote on the deposits to process every 4 hours.</li> </ol>
	You can check more details about your validator, including the estimated time to join the network, at <u>https://beaconcha.in/</u> .

# Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 11 of 117 PageID #: 73

Claim Chart U.S. Patent No. 11,196,566 (the "'566 Patent") Coinbase	
<u>Claim 2</u>	<u>Coinbase Products &amp; Services</u> Payment of block rewards to new Validator Nodes under Proof of Stake
The device of claim 1, where:	The Computing Device   Facilitator consists of:
the computer processor is configured to obtain the one or more output data based on:	• the Coinbase (Owned, managed) Ethereum Validator Full Nodes; and
buscu on.	• the Coinbase (Owned, managed) Ethereum supporting Archive Nodes and Light Nodes; and
	Client Device
	The First Client is the active Coinbase (Owned, managed) Ethereum Validator Full Nodes. The Computing Device and the First Client are the same device.
	The Second Client is the newly activated Validator Node, which will include any prospective Coinbase (Owned, managed or offered) Ethereum Validator Nodes.
	The Coinbase (Owned, managed) Ethereum Validator Nodes facilitates value transfer to newly activated Validator Nodes (as Second Client) based on work performed in producing blocks securely. The Beacon Chain upgrade brings proof-of-stake consensus to Ethereum. For this, active participants - known as validators – are required to propose, verify, and vouch for the validity of blocks.
the first principal data; and the value data from the first data source.	The Coinbase (Owned, managed) Ethereum Validator Full Nodes processes the validator rewards at the end of each epoch. This is calculated from the Beacon Chain reward and penalty algorithm (Data Source) and considers the staked principle.
	https://consensys.net/blog/codefi/rewards-and-penalties-on-ethereum- 20-phase-0/
	https://kb.beaconcha.in/rewards-and-penalties#block-reward
	Each Validator account is updated with the reward/penalty. The Beacon Block is updated accordingly.
	<pre>def process_rewards_and_penalties(state: BeaconState) -&gt; None:     # No rewards are applied at the end of `GENESIS_EPOCH` because rewards are for work done in the previous epoch     if get_current_epoch(state) == GENESIS_EPOCH:         return</pre>
	rewards, penalties = get_attestation_deltas(state) for index in range(len(state.validators)): increase_balance(state, ValidatorIndex(index), rewards[index])

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 12 of 117 PageID #: 74

decrease_balance(state, ValidatorIndex(index), penalties[index])
https://github.com/ethereum/consensus- specs/blob/dev/specs/phase0/beacon- chain.md#process_rewards_and_penalties
<pre>def process_slashings(state: BeaconState) -&gt; None: epoch = get_current_epoch(state) total_balance = get_total_active_balance(state) adjusted_total_slashing_balance = min(sum(state.slashings) * PROPORTIONAL_SLASHING_MULTIPLIER, total_balance) for index, validator in enumerate(state.validators): if validator.slashed and epoch + EPOCHS_PER_SLASHINGS_VECTOR // 2 == validator.withdrawable_epoch: increment = EFFECTIVE_BALANCE_INCREMENT # Factored out from penalty numerator to avoid uint64 overflow penalty_numerator = validator.effective_balance // increment * adjusted_total_slashing_balance penalty = penalty_numerator // total_balance * increment</pre>
decrease_balance(state, ValidatorIndex(index), penalty) <a href="https://github.com/ethereum/consensus-specs/blob/dev/specs/phase0/beacon-chain.md#slashings">https://github.com/ethereum/consensus-specs/blob/dev/specs/phase0/beacon-chain.md#slashings</a>

Claim 3	aim Chart U.S. Patent No. 11,196,566 (the "'566 Patent") Coinbase <u>Coinbase Products &amp; Services</u>
	Payment of block rewards to new Validator Nodes under Proof of Stake
The device of claim 1,	The Computing Device   Facilitator consists of:
where the computer	• the Coinbase (Owned, managed) Ethereum Validator Full Nodes; and
processor is further configured to:	• the Coinbase (Owned, managed) Ethereum supporting Archive Nodes and Light Nodes; and
	Client Device
	The First Client is the active Coinbase (Owned, managed) Ethereum Validator Full Nodes. The Computing Device and the First Client are the same device.
	The Second Client is the newly activated Validator Node, which will include any prospective Coinbase (Owned, managed) Ethereum Validator Nodes.
	The Coinbase (Owned, managed) Ethereum Validator Nodes facilitates value transfer to newly activated Validator Nodes (as Second Client) based on work performed in producing blocks securely. The Beacon Chain upgrade brings proof-of-stake consensus to Ethereum. For this, active participants - known as validators – are required to propose, verify, and vouch for the validity of blocks.
compute a second cryptographic	The Voluntary Exit is initiated and signed from the Validator Nodes Client CLI.
signature from the	Teku Client
first private key;	teku voluntary-exitvalidator-keys= <key_dir>:<pass_dir></pass_dir></key_dir>
	<key_file>:<pass_file>[,<key_dir>:<pass_dir>   <key_file>::<pass_file>]</pass_file></key_file></pass_dir></key_dir></pass_file></key_file>
	https://docs.teku.consensys.net/en/latest/Reference/CLI/Subcommands/Voluntary-Exit/
	Lighthouse Client
	In order to initiate an exit, users can use the lighthouse account validator exit command. Thekeystore flag is used to specify the path to the EIP-2335 voting keystore for the validator.
	Thebeacon-node flag is used to specify a beacon chain HTTP endpoint that
	confirms to the Eth2.0 Standard Beacon Node API specifications. That beacon
	node will be used to validate and propagate the voluntary exit. The default
	value for this flag is http://localhost:5052. Thenetwork flag is used to specify a particular Eth2 network (default is
	mainnet).
	Thepassword-file flag is used to specify the path to the file containing the password for the voting keystore. If this flag is not provided, the user will be
	prompted to enter the password.
	After validating the password, the user will be prompted to enter a special exit phrase as a final confirmation after which the voluntary exit will be published to the beacon chain.
	https://lighthouse-book.sigmaprime.io/voluntary-exit.html
create an another	A Signed Voluntary Exit message is created.
inchoate data record comprising:	
ccora comprising.	1

C	aim Chart U.S. Patent No. 11,196,566 (the "'566 Patent") Coinbase
<u>Claim 3</u>	Coinbase Products & Services
	Payment of block rewards to new Validator Nodes under Proof of Stake
a commit input for receiving the	The Voluntary Exit is initiated and signed from the Validator Client.
commit data from	Teku Client
the commit	teku voluntary-exitvalidator-keys= <key dir="">:<pass dir=""></pass></key>
transaction;	<key_file>:<pass_file>[,<key_dir>:<pass_dir> </pass_dir></key_dir></pass_file></key_file>
	<key_file>:<pass_file>]</pass_file></key_file>
	https://docs.teku.consensys.net/en/latest/Reference/CLI/Subcommands/Voluntary-Exit/
	Lighthouse Client
	<i>In order to initiate an exit, users can use the lighthouse account validator exit command.</i>
	The <i>investore</i> flag is used to specify the path to the EIP-2335 voting keystore for the validator.
	Thebeacon-node flag is used to specify a beacon chain HTTP endpoint that
	confirms to the Eth2.0 Standard Beacon Node API specifications. That beacon
	node will be used to validate and propagate the voluntary exit. The default
	value for this flag is http://localhost:5052.
	Thenetwork flag is used to specify a particular Eth2 network (default is
	mainnet).
	Thepassword-file flag is used to specify the path to the file containing the
	password for the voting keystore. If this flag is not provided, the user will be
	prompted to enter the password.
	After validating the password, the user will be prompted to enter a special exit phrase as a final confirmation after which the voluntary exit will be published to the beacon chain.
	https://lighthouse-book.sigmaprime.io/voluntary-exit.html
a refund output comprising a	The Voluntary Exit message contains a reference to the ValidatorIndex which is a pointer to the account. The refund output is the account balance.
refund data;	https://github.com/ethereum/consensus-specs/blob/dev/specs/phase0/beacon-
	chain.md#voluntary-exits
	def process_voluntary_exit(state: BeaconState, signed_voluntary_exit:
	SignedVoluntaryExit) -> None:
	voluntary_exit = signed_voluntary_exit.message
	<pre>validator = state.validators[voluntary_exit.validator_index] # Verify the validator is active</pre>
	assert is active validator(validator, get current epoch(state))
	# Verify exit has not been initiated
	assert validator.exit_epoch == FAR_FUTURE_EPOCH
	# Exits must specify an epoch when they become valid; they are not valid before then
	assert get_current_epoch(state) >= voluntary_exit.epoch
	# Verify the validator has been active long enough
	assert get_current_epoch(state) >= validator.activation_epoch +
	SHARD_COMMITTEE_PERIOD
	# Verify signature
	domain = get_domain(state, DOMAIN_VOLUNTARY_EXIT, voluntary_exit.epoch)
	<pre>signing_root = compute_signing_root(voluntary_exit, domain) assert bls.Verify(validator.pubkey, signing_root, signed_voluntary_exit.signature)</pre>
	assort ofs. v offig(vandator.puokey, signing_root, signed_voluntary_exit.signature)

	aim Chart U.S. Patent No. 11,196,566 (the "'566 Patent") Coinbase
<u>Claim 3</u>	<u>Coinbase Products &amp; Services</u> Payment of block rewards to new Validator Nodes under Proof of Stake
	# Initiate exit
	initiate_validator_exit(state, voluntary_exit.validator_index)
the second cryptographic	The Signed Voluntary Exit message contains the signature from the Validator.
signature; and	https://github.com/ethereum/consensus-specs/blob/dev/specs/phase0/beacon-
	chain.md#signedvoluntaryexit
	class SignedVoluntaryExit(Container): message: VoluntaryExit signature: BLSSignature
a lock time; and	One of the main design decisions of the Ethereum 2 project is performing the roll-out of the system over several phases. This decision has a serious impact on voluntary exits. Even though validators are able to perform an exit in Phase 0 and Phase 1, they will have to wait until Phase 2 to be able to withdraw. This means their staked funds will be frozen until withdrawals are available, which should be around 2 years after Mainnet launch.
	https://docs.prylabs.network/docs/wallet/exiting-a-validator/
publish the another inchoate data record to at least one of the first client device or the second	As the Computing Device and the First Client are the same device (Coinbase Ethereum Validator Node), the inchoate data record is already signed by the First Client. As such the inchoate data record is considered to be published to the First Client on creation of the inchoate data record by the Validator Node.
device or the second client device.	The signed inchoate data record (as a complete data record) is broadcast and recorded on the Ethereum Network. The following is an example output message from the Lighthouse Client.
	Successfully published voluntary exit for validator 0xabcd Voluntary exit has been accepted into the beacon chain, but not yet finalized. Finalization may take several minutes or longer. Before finalization there is a low probability that the exit may be reverted. Current epoch: 29946, Exit epoch: 29951, Withdrawable epoch: 30207 Please keep your validator running till exit epoch Exit epoch in approximately 1920 secs.
	https://lighthouse-book.sigmaprime.io/voluntary-exit.html

# Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 16 of 117 PageID #: 78

Claim 7	ent No. 11,196,566 (the "'566 Patent") Coinbase <u>Coinbase Products &amp; Services</u>
A system for processing a transaction between a first client device and a second client device via a transfer mechanism the system comprising a computing device, the first client device, the second client device, and	Payment of block rewards to new Validator Nodes under Proof of Stake
<ul> <li>the transfer mechanism.</li> <li>7. a. the computing device comprising for storing a first asymmetric key pair, the first asymmetric key pair comprising a first private key and a first public key;</li> </ul>	<ul> <li>The Computing Device   Facilitator consists of: <ul> <li>the Coinbase (Owned, managed) Ethereum Validator Full Nodes; and</li> <li>the Coinbase (Owned, managed) Ethereum supporting Archive Nodes and Light Nodes; and</li> </ul> </li> <li>Client Device <ul> <li>The First Client is an <u>active</u> Coinbase (Owned, managed) Ethereum Validator Full Nodes. The Computing Device and the First Client are the same device.</li> </ul> </li> </ul>
	<ul> <li>These consist of a computer hardware/software combination to run, namely:</li> <li>Memory (RAM), used in the computing device (such as a computer, server or server cloud instance).</li> <li>Transaction record sector (stores transactions and data that haven't been submitted to the blockchain yet)</li> <li>a first key pair sector which is generated and stored on the device (typically)</li> <li>The asymmetric key pair generated and/or stored consists of a first private key and a first public key and is stored on the Eth Node</li> </ul>
	Recommended hardware requirements for running a node. <u>https://launchpad.ethereum.org/en/checklist</u> A description of the Validator's Node keys is described in the followin link. <u>https://kb.beaconcha.in/ethereum-2-keys</u>
	<ul> <li>The validator signing key consists of two elements:</li> <li>1. Validator private key</li> <li>2. Validator public key</li> <li>The purpose of the validator private key is to actively sign on-chain (ETH2) operations such as block proposals and attestations. Therefore these keys have to be held in a hot wallet.</li> </ul>

		Example Validator Client (Prysmatic) installation guide shows key management.
		https://docs.prylabs.network/docs/install/install-with-script/
ii.	a first network interface for receiving terms, the terms comprising.	The First Client is an <u>active</u> Coinbase (Owned, Managed) Ethereum Validator Full Nodes. The Computing Device and the First Client are the same device.
		The Ethereum Network requires network connectivity in order to achieve PoS consensus.
		https://launchpad.ethereum.org/en/checklist
		https://github.com/ethereum/consensus-
		specs/blob/dev/specs/phase0/validator.md#block-proposal
		https://github.com/ethereum/consensus-
		specs/blob/dev/specs/phase0/validator.md#attesting
		https://github.com/ethereum/consensus-
		specs/blob/dev/specs/phase0/validator.md#broadcast-attestation
		https://github.com/ethereum/consensus- specs/blob/dev/specs/phase0/validator.md#broadcast-aggregate
		specs/biob/dev/specs/phaseo/validatol.mu#bioadcast-aggregate
A.	<i>At least one of a first principal data or second principle data;</i>	<b>First principle data</b> The initial deposit staking amount to participate as a Eth2 Validator in the Ethereum Network.
		The Beacon Chain upgrade brings proof-of-stake consensus to Ethereum. For this, we need active participants - known as validators - to propose, verify, and vouch for the validity of blocks. In exchange, honest validators receive financial rewards. Importantly, as a validator you'll need to post ETH as collateral - in other words, have some funds at stake. The only way to become a validator is to make a one-way ETH transaction to the deposit contract on the current Ethereum chain.
		https://launchpad.ethereum.org/en/overview
		https://github.com/ethereum/consensus-
		specs/blob/dev/specs/phase0/beacon-chain.md#deposits
В.	a reference to at least one of a first data source or a second data source; and	<b>First data source</b> The Beacon Chain reward and penalty algorithm.
	secona ada source, ana	https://consensys.net/blog/codefi/rewards-and-penalties-on- ethereum-20-phase-0/
		https://kb.beaconcha.in/rewards-and-penalties#block-reward
		https://github.com/ethereum/consensus-
		specs/blob/dev/specs/phase0/beacon-chain.md#attestations

	C. an expiration timestamp,	At the beginning of each epoch (every 32 slots, except GENESIS), several things happen, including: <ul> <li>Justification and finalization of the chain</li> <li>Assignment of rewards and penalties to attesters</li> <li>Update of the validator registry</li> <li>The special slashing penalty (see above), and</li> <li>Some final updates (computing effective balances, resets, etc)</li> </ul> <li><a href="https://consensys.net/blog/codefi/rewards-and-penalties-on-ethereum-20-phase-0/">https://consensys.net/blog/codefi/rewards-and-penalties-on-ethereum-20-phase-0/</a> </li> <li><a and="" between="" block.<br="" href="https://github.com/ethereum/consensus-specs/blob/dev/specs/phase0/beacon-chain.md#epoch-specs/blob/dev/specs/phase0/beacon-chain.md#epoch-specs/blob/dev/specs/phase0/beacon-chain.md#epoch-specs/blob/dev/specs/phase0/beacon-chain.md#epoch-specs/blob/dev/specs/phase0/beacon-chain.md#epoch-specs/blob/dev/specs/phase0/beacon-chain.md#epoch-specs/blob/dev/specs/phase0/beacon-chain.md#epoch-specs/blob/dev/specs/phase0/beacon-chain.md#epoch-specs/blob/dev/specs/phase0/beacon-chain.md#epoch-specs/blob/dev/specs/phase0/beacon-chain.md#epoch-specs/blob/dev/specs/phase0/beacon-chain.md#epoch-specs/blob/dev/specs/phase0/beacon-chain.md#epoch-specs/blob/dev/specs/phase0/beacon-chain.md#epoch-specs/blob/dev/specs/phase0/beacon-chain.md#epoch-specs/blob/dev/specs/phase0/beacon-chain.md#epoch-specs/blob/dev/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/specs/sp&lt;/th&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;iii.&lt;/td&gt;&lt;td&gt;a first computer processor&lt;br&gt;coupled to the first memory and&lt;br&gt;the first network interface, the&lt;br&gt;first computer processor&lt;br&gt;configured to:&lt;/td&gt;&lt;td&gt;processing&lt;br&gt;The First Client is an active Coinbase (Owned, Managed) Ethereum&lt;br&gt;Validator Full Nodes. The Computing Device and the First Client are&lt;br&gt;the same device.&lt;br&gt;These consist of a computer hardware/software combination to run.&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;A. read the first private key from&lt;br&gt;the first memory;&lt;/td&gt;&lt;td&gt;The Ethereum Validator Full Node is expected to propose a SignedBeaconBlock at the beginning of any slot during which is _proposer(state, validator_index) returns True. To propose, the validator selects the BeaconBlock, parent, that in their view of the fork choice is the head of the chain during slot - 1. The validator creates, signs, and broadcasts a block that is a child of parent that satisfies a valid beacon chain state transition.&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;td&gt;https://github.com/ethereum/consensus-&lt;br&gt;specs/blob/dev/specs/phase0/validator.md#block-proposal&lt;br&gt;Preparing for a BeaconBlock&lt;br&gt;To construct a BeaconBlockBody, a block (BeaconBlock) is defined&lt;br&gt;with the necessary context for a block proposal:&lt;br&gt;Slot&lt;br&gt;Set block.slot = slot where slot is the current slot at which the validator&lt;br&gt;has been selected to propose. The parent selected must satisfy that&lt;br&gt;parent.slot &lt; block.slot.&lt;br&gt;Note: There might be " parent="" skipped"="" slots="" the="">These skipped slots are processed in the state transition function without per-block processing. Proposer index Set block.proposer_index = validator_index where validator_index is the validator chosen to propose at this slot. The private key mapping to state.validators[validator_index].pubkey is used to sign the block. BLS public key Validator public keys are G1 points on the BLS12-381 curve. A</a></li>
--	-----------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	resultant pubkey. This privkey must be "hot", that is, constantly available to sign data throughout the lifetime of the validator.
	https://github.com/ethereum/consensus- specs/blob/dev/specs/phase0/validator.md#bls-public-key
B. compute a first cryptographic signature from the first private key;	The Ethereum Validator Full Node is expected to propose a SignedBeaconBlock at the beginning of any slot during which is_proposer(state, validator_index) returns True. To propose, the validator selects the BeaconBlock, parent, that in their view of the fork choice is the head of the chain during slot - 1. The validator creates, signs, and broadcasts a block that is a child of parent that satisfies a valid beacon chain state transition.
	https://github.com/ethereum/consensus-
	specs/blob/dev/specs/phase0/validator.md#block-proposal
	<b>Preparing for a BeaconBlock</b> To construct a BeaconBlockBody, a block (BeaconBlock) is defined with the necessary context for a block proposal:
	Slot Set block.slot = slot where slot is the current slot at which the validator has been selected to propose. The parent selected must satisfy that parent.slot < block.slot. Note: There might be "skipped" slots between the parent and block.
	These skipped slots are processed in the state transition function without per-block processing.
	Proposer index
	Set block.proposer_index = validator_index where validator_index is the validator chosen to propose at this slot. The private key mapping to state.validators[validator_index].pubkey is used to sign the block.
	Signature
	<pre>signed_block = SignedBeaconBlock(message=block, signature=block_signature), where block_signature is obtai</pre>
	<pre>def get_block_signature(state: BeaconState, block: BeaconBlock, privkey: int) -&gt; BLSSignature: domain = get_domain(state, DOMAIN_BEACON_PROPOSER, compute_epoch_at_slot(block.slot)) signing_root = compute_signing_root(block, domain) return bls.Sign(privkey, signing_root)</pre>
C. create an inchoate data record comprising:	The SignedBeaconBlock, as produced by the Ethereum Network's Validator, containing the deposit of a prospective Validator that is yet to be added to the list of registered Validators.
a. a commit input for receiving a commit data from a commit transaction;	The 32ETH deposit stake required to instantiate a prospective Eth2 Validator.
b. one or more outputs obtained from at least one of the first principal data or the	The prospective Validator's reward or penalty. This is implied as these are rules of the network.
second principal data, and a value data from at least one	The Beacon Chain reward and penalty algorithm uses predefined values and formulas, along with the effective staking balance of the participating nodes to calculate the reward or penalty for the Validators.

of the first data source or the second data source; and	https://consensys.net/blog/codefi/rewards-and-penalties-on-         ethereum-20-phase-0/         https://kb.beaconcha.in/rewards-and-penalties#block-reward         https://github.com/ethereum/consensus-         specs/blob/dev/specs/phase0/beacon-chain.md#attestations
c. the first cryptographic signature; and	The Ethereum Network Validator is expected to propose a SignedBeaconBlock at the beginning of any slot during which is_proposer(state, validator_index) returns True. To propose, the validator selects the BeaconBlock, parent, that in their view of the fork choice is the head of the chain during slot - 1. The validator creates, signs, and broadcasts a block that is a child of parent that satisfies a valid beacon chain state transition. <u>https://github.com/ethereum/consensus-</u> specs/blob/dev/specs/phase0/validator.md#block-proposal Signature signed_block = SignedBeaconBlock(message=block, signature=block_signature), where block_signature is obtai def get_block_signature(state: BeaconState, block: BeaconBlock, privkey: int) -> BLSSignature: domain = get_domain(state, DMAIN_BEACON_PROPOSER, compute_epoch_at_slot(block.slot)) signing_root = compute_signing_root(block, domain) return bls.Sign(privkey, signing_root)
	Note that the signed block includes the Eth1 data containing the 32ETH deposit stake to instantiate a Validator.
	Eth1 Data The block.body.eth1_data field is for block proposers to vote on recent Eth1 data. This recent data contains an Eth1 block hash as well as the associated deposit root (as calculated by the get_deposit_root() method of the deposit contract) and deposit count after execution of the corresponding Eth1 block. If over half of the block proposers in the current Eth1 voting period vote for the same eth1_data then state.eth1_data updates immediately allowing new deposits to be processed. Each deposit in block.body.deposits must verify against state.eth1_data.eth1_deposit_root.
	https://github.com/ethereum/consensus- specs/blob/dev/specs/phase0/validator.md#eth1-data
D. publish the inchoate data record to at least one of the first client device or the second client device;	As the Computing Device and the First Client are the same device (Coinbase (Owned, managed) Ethereum Validator Node), the inchoate data record is already known by the First Client. As such the inchoate data record is considered to be published to the First Client on creation of the inchoate data record by the Validator Node.
	A validator is expected to propose a <u>SignedBeaconBlock</u> at the beginning of any slot during which is_proposer(state, validator_index) returns True. To propose, the validator selects the BeaconBlock,

7. b. the first client comprises: i. a second memory for storing a second asymmetric key pair, the second asymmetric key pair comprising a second private key and a second public key.	<ul> <li>parent, that in their view of the fork choice is the head of the chain during slot - 1. The validator creates, signs, and broadcasts a block that is a child of parent that satisfies a valid beacon chain state transition.</li> <li>https://github.com/ethereum/consensus-specs/blob/dev/specs/phase0/validator.md#block-proposal</li> <li>The Computing Device   Facilitator consists of: <ul> <li>the Coinbase (Owned, managed) Ethereum Validator Full Nodes; and</li> <li>the Coinbase (Owned, managed) Ethereum supporting Archive Nodes and Light Nodes; and</li> </ul> </li> <li>Client Device <ul> <li>The First Client is an active Coinbase (Owned, managed) Ethereum Validator Full Nodes.</li> </ul> </li> <li>The Facilitator and the First Client are the same device. Thus, the clause is not assessable as it is covered by the above clauses 7a.i.</li> </ul>
ii. a second network interface; and	The Facilitator and the First Client are the same device. Thus, the clause is not assessable as it is covered by the above clauses 7a.ii.
iii. a second computer processor coupled to the second memory and the second network interface, the second computer processor configured to:	<u>The Facilitator and the First Client are the same device.</u> Thus, the clause is not assessable as it is covered by the above clauses 7a.iii.A-E.
A. read the second private key from the second key pair sector;	The Facilitator and the First Client are the same device. Thus, the clause is not assessable as it is covered by the above clauses 7a.iii.A.
B. read the inchoate disbursement transaction record,	The Facilitator and the First Client are both considered to be the same device. Thus, the clause is not assessable.
C. compute a second cryptographic signature from the second private key	The Facilitator and the First Client are the same device. Thus, the clause is not assessable as it is covered by the above clauses 7a.iii.B.
<ul> <li>D. create a complete data record comprising:</li> <li>I. the commit input,</li> </ul>	The Facilitator and the First Client are the same device. Thus, the clause is not assessable as it is covered by the above clauses 7a.iii.C.
II. the output data,	<u>The Facilitator and the First Client are the same device.</u> Thus, the clause is not assessable as it is covered by the above clauses 7a.iii.C.
III. the first cryptographic signature, and	The Facilitator and the First Client are the same device. Thus, the clause is not assessable as it is covered by the above clauses 7a.iii.C.

# Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 22 of 117 PageID #: 84

IV. the second cryptographic signature,	The Facilitator and the First Client are the same device. Thus, the clause is not assessable as it is covered by the above clauses 7a.iii.C.
E. create a transaction by submitting the complete data record to the transfer mechanism.	The Facilitator and the First Client are the same device. Thus, the clause is not assessable as it is covered by the above clauses 7a.iii.D.
7. c. the second client comprises: i. a third memory for storing a third asymmetric key pair, the third asymmetric key pair comprising a third private	The Second Client is a prospective Validator Node, which will include any prospective Coinbase (Owned, managed, offered) Ethereum Validator Nodes.These consist of a computer hardware/software combination to run,
key and a third public key.	<ul><li>namely:</li><li>Memory (RAM), used in the computing device (such as a</li></ul>
	<ul> <li>computer, server or server cloud instance).</li> <li>Transaction record sector (stores transactions and data that haven't been submitted to the blockchain yet)</li> </ul>
	<ul> <li>a first key pair sector which is generated and stored on the device (typically)</li> </ul>
	• The asymmetric key pair generated and/or stored consists of a first private key and a first public key and is stored on the device
	Recommended hardware requirements for running a node.
	https://launchpad.ethereum.org/en/checklist
	A description of the Validator Node's keys is described in the following link.
	https://kb.beaconcha.in/ethereum-2-keys
	The validator signing key consists of two elements:
	<ol> <li>Validator private key</li> <li>Validator public key</li> </ol>
	The purpose of the validator private key is to actively sign on-chain (ETH2) operations such as block proposals and attestations. Therefore, these keys have to be held in a hot wallet.
	Example Validator Client (Prysmatic) installation guide shows key management.
	https://docs.prylabs.network/docs/install/install-with-script/
ii. <i>a third network interface;</i> <i>and</i>	The newly activated Validator requires network connectivity in order to contribute to PoS consensus.
	https://launchpad.ethereum.org/en/checklist
	https://github.com/ethereum/consensus- specs/blob/dev/specs/phase0/validator.md#block-proposal

	https://github.com/ethereum/consensus-
	specs/blob/dev/specs/phase0/validator.md#attesting
	https://github.com/ethereum/consensus-
	specs/blob/dev/specs/phase0/validator.md#broadcast-attestation
	https://github.com/ethereum/consensus-
	specs/blob/dev/specs/phase0/validator.md#broadcast-aggregate
<i>iii. a third computer processor</i>	The newly activated Validator is required to sign (by reading the locally
coupled to the third memory	stored private key) produced blocks, attestations and aggregated
and the third network	attestations.
interface, the third computer	https://github.com/othoroum/conconsus
processor configured to read the third private key from	https://github.com/ethereum/consensus- specs/blob/dev/specs/phase0/validator.md#signature
memory; and	https://github.com/ethereum/consensus-
memory, and	specs/blob/dev/specs/phase0/validator.md#aggregate-signature
	https://github.com/ethereum/consensus-
	specs/blob/dev/specs/phase0/validator.md#aggregate-signature-
	1
wherein the at least one of the first	As the Computing Device and the First Client are the same device
client device or the second client	(Coinbase (Owned, managed) Ethereum Validator Node), the inchoate
device signs the inchoate data record	data record is already signed by the First Client. The signed inchoate data record (as a complete data record) is broadcast and recorded on the
and saves a copy of the inchoate data record on at least one of the first client	Ethereum Network.
device or the second client device,	
,	A validator is expected to propose a SignedBeaconBlock at the
	beginning of any slot during which is _proposer(state, validator_index)
	returns True. To propose, the validator selects the BeaconBlock,
	parent, that in their view of the fork choice is the head of the chain
	during slot - 1. The validator creates, signs, and broadcasts a block
	that is a child of parent that satisfies a valid beacon chain state transition.
	https://github.com/ethereum/consensus-
	specs/blob/dev/specs/phase0/validator.md#block-proposal
wherein the transfer mechanism	Ether is used as the decentralised currency. The Ethereum network
comprising a decentralized digital	maintains a distributed ledger without the need for a trusted central
currency that comprises a distributed	authority.
ledger that enables processing the transaction between the first client	
device and the second client device	From the Ethereum yellow paper.
without the need of a trusted central	2.1. Value. In order to incentivise computation within the network,
authority,	there needs to be an agreed method for transmitting value. To address
	this issue, Ethereum has an intrinsic currency, Ether, known also as
	ETH and sometimes referred to by the Old English $$ D.
	https://ethereum.github.io/yellowpaper/paper.pdf
wherein the transaction is created by	As the Computing Device and the First Client are the same device
broadcasting the complete data record	(Coinbase (Owned, managed) Ethereum Validator Node), the inchoate

for transmitting and receiving among network participants in the computer network for recording in the distributed ledger, and	data record is already signed by the First Client. The signed inchoate data record (as a complete data record) is broadcast and recorded on the Ethereum Network.
aistributeu teuger, and	The SignedBeaconBlock, in order to achieve consensus, is required to be signed by the Ethereum Network as the First Client (the Eth2 Validators) as either the Block Producer, Attester or Aggregator's.
	<b>Proposer index</b> Set block.proposer_index = validator_index where validator_index is the validator chosen to propose at this slot. The private key mapping to state.validators[validator_index].pubkey is used to sign the block. https://github.com/ethereum/consensus- specs/blob/dev/specs/phase0/validator.md#proposer-index
	Attesting A validator is expected to create, sign, and broadcast an attestation during each epoch. The committee, assigned index, and assigned slot for which the validator performs this role during an epoch are defined by get_committee_assignment(state, epoch, validator_index). https://github.com/ethereum/consensus- specs/blob/dev/specs/phase0/validator.md#attesting
	<b>Construct aggregate</b> If the validator is selected to aggregate (is_aggregator()), they construct an aggregate attestation via the following. Collect attestations seen via gossip during the slot that have an equivalent attestation_data to that constructed by the validator. If len(attestations) > 0, create an aggregate_attestation: Attestation with the following fields.
	https://github.com/ethereum/consensus- specs/blob/dev/specs/phase0/validator.md#construct-attestation
	<pre>def get_aggregate_signature(attestations: Sequence[Attestation]) -&gt; BLSSignature:     signatures = [attestation.signature for attestation in attestations]     return bls.Aggregate(signatures)</pre>
	The Coinbase (Owned, managed) Ethereum Validator Node uses the SignedBeaconBlock (as a complete data record) along with the broadcasted Aggregated Attestations to update its record of the Beacon Chain. Finality is achieved if two epochs in a row are justified.
	<ul> <li>ETH2 uses Casper Proof of Stake, specifically, something called a "finality gadget". The ETH2 finality process is defined as follows:</li> <li>3. If &gt; 2/3rds of validators vote correctly on the chain head during an epoch, we call the last epoch justified</li> <li>4. If two epochs in a row are justified, the current_epoch - 2 is considered finalized</li> </ul>

# Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 25 of 117 PageID #: 87

	https://hackmd.io/@prysmaticlabs/finality#How-Finality-Works-in- ETH2
	The Coinbase (Owned, managed) Ethereum Validator Node as the First
	Client process the Beacon Block and records the new Validator in the Registry on identifying a new deposit in the Deposit Contract. The Beacon Block is saved by each Eth2 Client.
	https://github.com/ethereum/consensus- specs/blob/dev/specs/phase0/beacon-chain.md#deposits
wherein at least one of the computer device, the first client device, or the second client device verifies the	The new Validator Node, as the Second Client, can view the status of its activation from https://beaconcha.in/
recording of the complete data record in the distributed ledger by observing an external state	Deposited Pending Active Exited
	This validator has been processed by the beacon chain and is currently waiting to be activated. It will approximately be activated on <b>Feb 3</b> , <b>2021</b> , <b>6</b> :21 AM during epoch <b>14272</b> . Make sure your nodes and your client is up and running <i>before</i> the countdown reaches zero.
	18 days 11 hr 45 min 25 sec
	Once a deposit is made to the Ethereum Beacon Chain, it will join a waiting queue before joining the network.
	All Ethereum 2.0 deposits have two delays before going into the waiting queue:
	<ol> <li>Eth1 data inclusion delay: The Beacon Chain follows Eth1 with some delay to make sure the Eth1 data is finalized.</li> <li>Eth1 data voting delay: The Beacon Chain validators vote on the deposits to process every 4 hours.</li> </ol>
	You can check more details about your validator, including the estimated time to join the network, at https://beaconcha.in/.
	https://intercom.help/stakefish/en/articles/4799068-when-will-my- validator-join-the-network-and-be-activated

Claim 8 Coinbase Products & Services			
	Payment of block rewards to new Validator Nodes under Proof of Stake		
The system of claim	The Computing Device   Facilitator consists of:		
7, where the first	• the Coinbase (Owned, managed) Ethereum Validator Full Nodes; and		
computer processor is further configured to:	• the Coinbase (Owned, managed) Ethereum supporting Archive Nodes and Light Nodes; and		
	Client Device		
	The First Client is the active Coinbase (Owned, managed) Ethereum Validator Full Nodes. The Facilitator and the First Client are the same device.		
	The Second Client is the newly activated Validator Node, which will include any prospective Coinbase (Owned, managed) Ethereum Validator Nodes.		
compute a third cryptographic	The Voluntary Exit is initiated and signed from the Validator Nodes Client CLI.		
signature from the	Teku Client		
first private key;	teku voluntary-exitvalidator-keys= <key_dir>:<pass_dir>           <key_file>:<pass_file>[,<key_dir>:<pass_dir>           <key_file>::<pass_file>]         https://docs.teku.consensys.net/en/latest/Reference/CLI/Subcommands/Voluntary-Exit/</pass_file></key_file></pass_dir></key_dir></pass_file></key_file></pass_dir></key_dir>		
	Lighthouse ClientIn order to initiate an exit, users can use the lighthouse account validator exit command.Thekeystore flag is used to specify the path to the EIP-2335 voting keystorefor the validator.Thebeacon-node flag is used to specify a beacon chain HTTP endpoint thatconfirms to the Eth2.0 Standard Beacon Node API specifications. That beaconnode will be used to validate and propagate the voluntary exit. The defaultvalue for this flag is http://localhost:5052.Thenetwork flag is used to specify a particular Eth2 network (default is		
	mainnet). Thepassword-file flag is used to specify the path to the file containing the password for the voting keystore. If this flag is not provided, the user will be prompted to enter the password. After validating the password, the user will be prompted to enter a special exit phrase as a final confirmation after which the voluntary exit will be published to the beacon chain.		
	https://lighthouse-book.sigmaprime.io/voluntary-exit.html		
create another inchoate data record comprising:	A Signed Voluntary Exit message is created.		
a commit input for receiving the	The Voluntary Exit is initiated and signed from the Validator Client.		
commit data from	Teku Client taku volumtum, avit, validator kaus=		
the commit transaction;	teku voluntary-exitvalidator-keys= <key_dir>:<pass_dir>   <key_file>:<pass_file>[,<key_dir>:<pass_dir>  </pass_dir></key_dir></pass_file></key_file></pass_dir></key_dir>		
<i>u unsue non</i> ,	<key file="">::<pass file="">]</pass></key>		
	https://docs.teku.consensys.net/en/latest/Reference/CLI/Subcommands/Voluntary-Exit/		

	Lighthouse Client
	In order to initiate an exit, users can use the lighthouse account validator exit command. Thekeystore flag is used to specify the path to the EIP-2335 voting keystore for the validator.
	Thebeacon-node flag is used to specify a beacon chain HTTP endpoint that confirms to the Eth2.0 Standard Beacon Node API specifications. That beacon node will be used to validate and propagate the voluntary exit. The default value for this flag is http://localhost:5052.
	Thenetwork flag is used to specify a particular Eth2 network (default is mainnet). Thepassword-file flag is used to specify the path to the file containing the
	password for the voting keystore. If this flag is not provided, the user will be prompted to enter the password.
	After validating the password, the user will be prompted to enter a special exit phrase as a final confirmation after which the voluntary exit will be published to the beacon chain.
	https://lighthouse-book.sigmaprime.io/voluntary-exit.html
a refund output comprising a refund data; and	The Voluntary Exit message contains a reference to the ValidatorIndex which is a pointer to the account. The refund output is the account balance.
	https://github.com/ethereum/consensus-specs/blob/dev/specs/phase0/beacon- chain.md#voluntary-exits
	def process_voluntary_exit(state: BeaconState, signed_voluntary_exit: SignedVoluntaryExit) -> None:
	<pre>voluntary_exit = signed_voluntary_exit.message validator = state.validators[voluntary_exit.validator_index] # Verify the validator is active</pre>
	assert is _active_validator(validator, get_current_epoch(state)) # Verify exit has not been initiated
	<pre>assert validator.exit_epoch == FAR_FUTURE_EPOCH # Exits must specify an epoch when they become valid; they are not valid before then assert get_current_epoch(state) &gt;= voluntary_exit.epoch</pre>
	# Verify the validator has been active long enough
	assert get_current_epoch(state) >= validator.activation_epoch + SHARD_COMMITTEE_PERIOD
	# Verify signature domain = get_domain(state, DOMAIN_VOLUNTARY_EXIT, voluntary_exit.epoch)
	<pre>signing_root = compute_signing_root(voluntary_exit, domain) assert bls.Verify(validator.pubkey, signing_root, signed_voluntary_exit.signature) # Initiate exit</pre>
	initiate_validator_exit(state, voluntary_exit.validator_index)
the third cryptographic	The Signed Voluntary Exit message contains the signature from the Validator.
signature; and	https://github.com/ethereum/consensus-specs/blob/dev/specs/phase0/beacon- chain.md#signedvoluntaryexit
	class SignedVoluntaryExit(Container): message: VoluntaryExit signature: BLSSignature

publish the another	As the Computing Device and the First Client are the same device (Coinbase Ethereum
inchoate data	Validator Node), the inchoate data record is already signed by the First Client. As such
record to at least	the inchoate data record is considered to be published to the First Client on creation of
one of the first client	the inchoate data record by the Validator Node.
and the second	
client.	The signed inchoate data record (as a complete data record) is broadcast and recorded on
	the Ethereum Network.
	The following is an example output message from the Lighthouse Client.
	Successfully published voluntary exit for validator 0xabcd Voluntary exit has been
	accepted into the beacon chain, but not yet finalized. Finalization may take several
	minutes or longer. Before finalization there is a low probability that the exit may be
	reverted. Current epoch: 29946, Exit epoch: 29951, Withdrawable epoch: 30207 Please
	keep your validator running till exit epoch. 25551, whindrawable epoch. 56267 Hease
	https://lighthouse-book.sigmaprime.io/voluntary-exit.html

# Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 29 of 117 PageID #: 91

Claim Chart U.S. Pate	nt No. 11,196,566 (the "'566 Patent") Coinbase
<b><u>Claim 1</u></b> A computing device for processing a transaction between a first client device, and a second client device via a transfer mechanism, the transfer mechanism comprising a decentralized digital currency	<u>Coinbase Products &amp; Services</u> Payment to Validator Node from a transaction (that incurs a transaction fee) on the Solana Network.
The computing device comprising:	<ul> <li>The Computing Device   Facilitator consists of:</li> <li>the Coinbase (Owned, managed or offered) Solana Validator Full Nodes; and</li> <li>the Coinbase (Owned, managed or offered) Solana supporting Archive Nodes and Light Nodes; and</li> <li>the Coinbase (Solana compatible) wallets;</li> <li>Where both the Coinbase Nodes and the Coinbase Solana compatible end user wallet device are networked to have direct or indirect communication with each other.</li> </ul>
	The Computing Device and the First Client are considered to be the same device. Four (4) instances have been identified that represent various implementations that exist.
	• The Computing Device and the First Client are the same device where the First Client runs Coinbase Wallet software to create, sign and submit transactions to the Solana Node for processing.
	• The Computing Device and the First Client are the same device where the First Client is a Client Browser or Command Line Interface on a Coinbase Solana Node used to create, sign and submit transactions to the Coinbase Solana Node for processing.
	• The Computing Device and the First Client are the same device where the First Client utilises Coinbase Private Key management mechanism in order to sign and submit transactions to the Coinbase Solana Node for processing.
	• The Computing Device and the First Client are the same device where the First Client is a Coinbase Validator Node with a vote account key pair to sign voting transactions.
	The Patent allows for, and the Claims do not prevent, the Computing System from being, or including, the First Client. This is detailed in the Patent description [0055].
	FIG. 1 (see Figure 16) depicts a typical embodiment for practicing the invention—especially for use with a distributed transfer mechanism—where the clients, transfer mechanism, facilitator, and data source are distinct participants. However, the depicted arrangement is not the only one contemplated by the invention. In an alternate embodiment,

	the facilitator provides some or all aspects of the transfer mechanism. In another embodiment, the facilitator comprises some or all aspects of a client. For example, part or all of a client's data store, the ability to initiate or accept offers, etc., could be "embedded" in the facilitator, thereby enabling the facilitator to operate as a client itself (e.g., one controlled by the owners of the facilitator, or on behalf of a third party who has entrusted control to the facilitator). In yet another embodiment, the facilitator comprises the data source. Many configurations are contemplated by the invention are possible, and will become apparent to one skilled in the art.
<ul> <li>a memory for storing a first asymmetric key pair, the first asymmetric key pair comprising a first private key and a first public key;</li> </ul>	<ul> <li>The First Client, as part of the Computing Device, need a computer hardware/software combination to run, namely:</li> <li>Memory (RAM), used in the computing device (such as a computer or mobile phone).</li> <li>Transaction record sector (stores transactions that haven't been submitted to the blockchain yet) kept via the crypto software wallets</li> </ul>
	<ul> <li>Where the First Client runs Coinbase Wallet software to sign transactions, contains: <ul> <li>a first key pair sector which is generated and stored in the wallet software</li> <li>The asymmetric key pair generated and/or stored consists of a first private key and a first public key – all found and manipulated via the wallet software.</li> <li>The wallet software connects via the public key or the key pair, and authorizes (signs) the transaction with the private key of the key pair.</li> </ul> </li> </ul>
	<ul> <li>Where the First Client is a Coinbase Solana Node Client Browser or Command Line Interface to transaction creation, contains: <ul> <li>a first key pair sector which is generated and stored on the device</li> <li>The asymmetric key pair stored consists of a first private key and a first public key.</li> </ul> </li> </ul>
	<ul> <li>Where the First Client utilises Coinbase Private key management in order to sign transactions, contains: <ul> <li>a first key pair sector which is stored in the key management vault/software</li> <li>The asymmetric key pair generated and/or stored consists of a first private key and a first public key – all found and manipulated via the key management vault/software.</li> <li>The key management vault/software connects via the public key or the key pair, and authorizes (signs) the transaction with the private key of the key pair.</li> </ul> </li> </ul>
	<ul> <li>Where the First Client is a Coinbase Validator Node with a vote account key pair to sign voting transactions contains</li> <li>a first key pair sector which is generated and stored on the device</li> </ul>

• The asymmetric key pair stored consists of a first private key and a first public key.
Source Code https://github.com/solana- labs/solana/blob/master/validator/src/main.rs#L2999
The Coinbase Wallet product serves as the network interface usage implementation, we see the following.
The app prompts you to connect to "Connect my Coinbase Wallet" Note that the mobile data interface is set to airplane mode, which disables all network data i/o coming into and out of the mobile device (reference the 'plane' icon in the upper RHS corner).
Use of coinbase.com account linkage feature is subject to coinbase.com User Agreement and Privacy Policy.
Connect my Coinbas Upon attempting to "Connect my Coinbase" with the data interface disabled, a blank screen results, as follows:

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 32 of 117 PageID #: 94

7:44 ₪ 🛢 🖬 • 🛛 📲 🛧 55% 🛔 ← Sign in
The Coinbase Wallet app interfaces with nodes and servers in the Coinbase infrastructure to query information or connect to a client. To fetch data like user identification verification credentials, token swap rates, token balances, and other information from the Coinbase infrastructure servers, nodes and smart contracts, such infrastructure responds to queries through the Coinbase Wallet app, the network interface, Once the network interface is enabled, as shown by the wi fi and
Once the network interface is enabled, as shown by the wi-fi and cellular service icons in the upper right-hand corner that have replaced the airplane mode symbol, the interface for entering and submitting user credentials is available after querying Coinbase's servers. The interface actually states that "you will be redirected to <u>https://wallet.coinbase.com</u> (their website) after the authorization"

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 33 of 117 PageID #: 95

7:47 ▲ 4 때 • 송 세 종 세 동 4 54% à
<ul> <li>← Sign in</li> <li>coinbase Ξ</li> </ul>
by Coinbase Sign in to your Coinbase account to authorize <b>Coinbase Wallet</b> (Android)
You will be redirected to https://wallet.coinbase.com after the authorization.
Email you@example.com
Password SIGN IN
Once the Coinbase Wallet app is connected, it is able to query Coinbase infrastructure and populate the app with a plethora of information.
Choosing the "Send" option as an illustration, we can opt to send SOL to any address.

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 34 of 117 PageID #: 96

Confirm send         Image: Serum       \$0.10         Image: Output Description       0.0813 SRM         Image: Image: Output Description       0.0813 SRM         Image: Output Description       Serum         Image: Output Description       26vuBUdwkP	0.0833 SRM         94.13 available         I       2         I       2         4       5         7       8         0       -         Continue       -         Image: Serum       -         Image	÷	Send Cr	урто						
\$4.18 available         Image: Serum S	\$4.18 available         Image: Serum       \$4.18         1       2       3         4       5       6         7       8       9         .       0       ←         Continue       Continue	Max	\$0	.1						
Your balance     3.396 SRM       1     2     3       4     5     6       7     8     9       .     0     ←       Continue     Continue	Your balance       3.396 SRM         1       2       3         4       5       6         7       8       9         .       0       +         Continue       Continue									
4       5       6         7       8       9         .       0       €         Continue         Continue         the Coinbase Wallet app populates the app with prices, quantities ansaction fees.         Confirm send         Image: Serum       \$0.10         0.0813 SRM       0.0813 SRM         Image: Serum       \$0.0013 SRM         Image: Serum       \$0.0913 SRM         Image: Serum       \$0.001 JunkeP         Network       Solana mainnet         Network fee       \$0.001 JunkeP	4       5       6         7       8       9         .       0       €         Continue         continue         the Coinbase Wallet app populates the app with prices, quantities ansaction fees.         the Coinfirm send         Image: Serum       \$0.10         Outside Serum       \$0.10         Image: Serum       \$0.10         Image: Serum       \$0.10         Image: Serum       \$0.0013 SRM         Image: Serum       \$0.10         Image: Serum       \$0.10         Image: Serum       \$0.0013 SRM         Image: Serum       \$0.0013 SRM         Image: Serum       \$0.001 SOL									
7       8       9         .       0       ←         Continue       Continue         te Coinbase Wallet app populates the app with prices, quantities insaction fees.         ★       Confirm send         Serum       \$0.10         0.0913 SRM       0.0913 SRM         ↓       ©wallettest1         \$p2cRjXgbg       June         Allet used       26vuBUdwkP         etwork       Solana mainnet         twork fee       \$0.00 • 0.000005 SOL	7       8       9         0       ←         Continue         Continue         te Coinbase Wallet app populates the app with prices, quantities insaction fees.         Confirm send         Serum       \$0.10         0.0913 SRM       0.0913 SRM         Confirm send       \$0.0913 SRM         Serum       \$0.0913 SRM	1	2		3					
. 0 ← Continue The Coinbase Wallet app populates the app with prices, quantities ansaction fees. Confirm send Serum 0.0813 SRM 0.0813 SRM Wallet used 26VuBUdwkP Network Solana mainnet Network fee ① \$0.00 0.000005 SOL	0 ← Continue the Coinbase Wallet app populates the app with prices, quantities ansaction fees. <b>Confirm send</b> <b>Confirm send</b> <b>Serum</b> <b>Solution</b> <b>Serum</b> <b>Solution</b> <b>Serum</b> <b>Solution</b> <b>Serum</b> <b>Solution</b> <b>Solution</b> <b>Serum</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b> <b>Solution</b>	4	5		6					
Continue the Coinbase Wallet app populates the app with prices, quantities ansaction fees. ★ Confirm send \$0.10 0.0813 SRM U Wallet used 26/UBUdwkP Network Solana mainnet Network fee ● \$0.00 • 0.000005 SOL	Continue the Coinbase Wallet app populates the app with prices, quantities ansaction fees. ★ Confirm send \$0.10 0.0813 SRM 0.0813 SRM Wallet used 26vuBUdwkP Network Solana mainnet Network fee ● \$0.00 • 0.000005 SOL	7	8		9					
The Coinbase Wallet app populates the app with prices, quantities ansaction fees.	he Coinbase Wallet app populates the app with prices, quantities ansaction fees. Confirm send Serum Solido Serum Solido Serum Solido Serum Solido Serum Solido Serum Solido Serum Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solido Solid		0		÷					
Confirm send   Image: Serum   0.0813 SRM   Image: Serum   0.0813 SRM   Image: Serum   Image: Serum  <	Image: Confirm send   Image: Confirm send   Image: Serum   0.0813 SRM   Image: Confirm send   Image: Serum   Serum   0.0813 SRM   Image: Confirm send   Image: Serum    Serum   Serum		Contin							
Confirm send         Image: Serum       \$0.10         0.0813 SRM         Image: Serum       26vuBUdwkP         Network       Solana mainnet         Network fee       \$0.00 + 0.000005 SOL	Image: Confirm send     Image: Confirm send     Image: Serum   0.0813 SRM   Image: Optimized   Image: Sp2cRjXgbg     Wallet used   26vuBUdwkP   Network   Solana mainnet   Network fee ●			ue						
← Confirm send Confirm send Serum \$0.10 0.0813 SRM Wallettused 26vuBUdwkP Network Solana mainnet Network fee € \$0.00 + 0.000005 SOL	←       Confirm send         ●       \$0.10         0.0813 SRM         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●			lue						
Serum       \$0.10         0.0813 SRM         Image: Serum	Serum       \$0.10         0.0813 SRM         Image: Serum	The Coi	inbase Wall		opulate	s the	app	with p	orices, qu	uantitie
Image: Serum       0.0813 SRM         Image: Serum       Serum         Image: Serum       26vuBUdwkP         Network       Solana mainnet         Network fee Image: Solon + 0.000005 SOL	Serum 0.0813 SRM   Serum 0.0813 SRM   Sp2cRjXgbg   Wallet used 26vuBUdwkP Network Solana mainnet Network fee Soloo • 0.000005 SOL	transact	inbase Wall tion fees.	et app p	opulate	s the	app	with p	prices, qu	uantitio
Image: Wallet used mained       26vuBUdwkP         Network fee Image: Solana mainnet       \$0.00 • 0.000005 SOL	Image: Wallet used       26vuBUdwkP         Network       Solana mainnet         Network fee Image: \$0.00 • 0.000005 SOL	ransact	inbase Wall tion fees.	et app p	opulate	s the	app	with p	prices, qu	uantitie
● 5p2cRjXgbg         Wallet used       26vuBUdwkP         Network       Solana mainnet         Network fee ●       \$0.00 • 0.000005 SOL	● 5p2cRjXgbg         Wallet used       26vuBUdwkP         Network       Solana mainnet         Network fee ●       \$0.00 • 0.000005 SOL	ansact	inbase Wall tion fees. Confirm	et app p	\$0.10	s the	app	with p	prices, qu	uantitie
Wallet used     26vuBUdwkP       Network     Solana mainnet       Network fee •     \$0.00 • 0.000005 SOL	Wallet used     26vuBUdwkP       Network     Solana mainnet       Network fee •     \$0.00 • 0.000005 SOL	ansact	inbase Wall tion fees. Confirm	et app p	\$0.10	s the	app	with p	prices, qu	uantitie
Network Solana mainnet Network fee Soloo + 0.000005 SOL	Network Solana mainnet Network fee ● \$0.00 • 0.000005 SOL	ransact ←	inbase Wall tion fees. Confirm Serum @wallettest1	et app p send	\$0.10	s the	app	with p	orices, qu	uantitie
Network fee ● \$0.00 • 0.000005 SOL	Network fee  \$0.00 • 0.000005 SOL	eansact	inbase Wall tion fees. Confirm Serum @wallettest1	et app p send	\$0.10	s the	app	with p	prices, qu	uantitie
		ansact ←	inbase Wall tion fees. Confirm Serum @wallettest1 5p2cRjXgbg	et app p send 0.08	<b>\$0.10</b> 3 SRM	s the	app	with p	prices, qu	uantitie
Total cost \$0.10	Total cost \$0.10	← ←	inbase Wall tion fees. Confirm Serum @wallettest1 5p2cRjXgbg	et app p send 0.08 26vut	<b>\$0.10</b> 3 SRM 3UdwkP	s the	app	with p	prices, qu	uantitie
		Ansact	inbase Wall tion fees. Confirm Serum @wallettest1 5p2cRjXgbg	et app p send 0.08 26vut Solana	<b>\$0.10</b> I3 SRM BUdwkP	s the	app	with p	prices, qu	uantitie
		Ansact	inbase Wall tion fees. Confirm Serum @wallettest1 5p2cRjXgbg sed	et app p send 0.08 26vut Solana	<b>\$0.10</b> 3 SRM BUdwkP 1 mainnet	s the	app	with p	prices, qu	uantitie
		Ansact	inbase Wall tion fees. Confirm Serum @wallettest1 5p2cRjXgbg sed	et app p send 0.08 26vut Solana	<b>\$0.10</b> 3 SRM BUdwkP 1 mainnet	s the	app	with p	prices, qu	uantitie
		Ansact	inbase Wall tion fees. Confirm Serum @wallettest1 5p2cRjXgbg sed	et app p send 0.08 26vut Solana	<b>\$0.10</b> 3 SRM BUdwkP 1 mainnet	s the	app	with p	prices, qu	uantitie
		Ansact	inbase Wall tion fees. Confirm Serum @wallettest1 5p2cRjXgbg sed	et app p send 0.08 26vut Solana	<b>\$0.10</b> 3 SRM BUdwkP 1 mainnet	s the	app	with p	prices, qu	uantitie
		Ansact	inbase Wall tion fees. Confirm Serum @wallettest1 5p2cRjXgbg sed	et app p send 0.08 26vut Solana	<b>\$0.10</b> 3 SRM BUdwkP 1 mainnet	s the	app	with p	prices, qu	uantitie
Confirm	Confirm	Ansact	inbase Wall tion fees. Confirm Serum @wallettest1 5p2cRjXgbg sed	et app p send 0.08 26vuf Solana	<b>\$0.10</b> 3 SRM BUdwkP 1 mainnet	s the	app	with p	prices, qu	ıantitie

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 35 of 117 PageID #: 97

Wall	let ap	p access to the	plane mode turned on to deprive the Coinbase Coinbase infrastructure results in a very one attempts to perform the send transaction	
Max	•	<b>\$0.813 SRM</b> \$4.18 available		
		<b>rum</b> ur balance	<b>\$4.18</b> 3.396 SRM	
	1	2	3	
	4	5	6	
	7	8	9	
		0	<b>←</b>	
		Continue		
An e	error	dialog is given a	after the "continue" button is depressed.	

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 36 of 117 PageID #: 98

	<pre>7:55 B ■ 4 ·</pre>
	Try again
<ul> <li>at least one of a first principal data or a second principal data;</li> </ul>	First principle data         A transaction fee provided by the Client to pay for sending a transaction.         Note: Before any transaction instructions are processed, the fee payer account balance will be deducted to pay for transaction fees. If the fee payer balance is not sufficient to cover transaction fees, the transaction will be dropped by the cluster. If the balance was sufficient, the fees will be deducted whether the transaction is processed successfully or not. In fact, if any of the transaction instructions return an error or violate runtime restrictions, all account changes *except* the transaction fee deduction will be rolled back.         https://solanacookbook.com/core-concepts/transactions.html#fees         https://jstarry.notion.site/Transaction-Fees-f09387e6a8d84287aa16a34ecb58e239
0 a reference to at least one of a first data source or a second data source; and	First data source The current lampoons per signature Each validator uses signatures per slot (SPS) to estimate network congestion and SPS target to estimate the desired processing capacity of the cluster. The validator learns the SPS target from the genesis config, whereas it calculates SPS from recently processed transactions. The genesis config also defines a target lamports_per_signature, which is the fee to charge per signature when the cluster is operating at SPS target. https://docs.solana.com/implemented-proposals/transaction- fees#congestion-driven-fees

	Note that the Client uses the JSON RPC API to query the cluster for the current fee parameters. getFeeForMessage https://docs.solana.com/developing/clients/jsonrpc- api#getfeeformessage Second data source The portion of the transaction fee payable to the Leader. 50% of each transaction fee is burned, with the remaining fee retained by the validator that processes the transaction. https://docs.solana.com/inflation/terminology#effective-inflation-rate-
o an expiration timestamp;	1. The transaction needs to be performed within a timeframe. Thus the expiration time is implied from the use of the network.
	A transaction includes a recent blockhash to prevent duplication and to give transactions lifetimes. Any transaction that is completely identical to a previous one is rejected, so adding a newer blockhash allows multiple transactions to repeat the exact same action. Transactions also have lifetimes that are defined by the blockhash, as any transaction whose blockhash is too old will be rejected.
	https://docs.solana.com/developing/programming- model/transactions#recent-blockhash
	A blockhash contains a 32-byte SHA-256 hash. It is used to indicate when a client last observed the ledger. Validators will reject transactions when the blockhash is too old.
	https://docs.solana.com/developing/programming- model/transactions#blockhash-format
	Source Code Calculation of blockhash as part of transaction creation. <u>https://github.com/solana-</u> labs/solana/blob/master/client/src/nonblocking/rpc_client.rs#L656
	2. The Leader Validator processes the disbursement amount on receipt of the voting transaction.
	Before any transaction instructions are processed, the fee payer account balance will be deducted to pay for transaction fees. If the fee payer balance is not sufficient to cover transaction fees, the transaction will be dropped by the cluster. If the balance was sufficient, the fees will be deducted whether the transaction is processed successfully or not. In fact, if any of the transaction
	instructions return an error or violate runtime restrictions, all account changes *except* the transaction fee deduction will be rolled back.
	https://solanacookbook.com/core-concepts/transactions.html#fees
	https://jstarry.notion.site/Transaction-Fees- f09387e6a8d84287aa16a34ecb58e239

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 38 of 117 PageID #: 100

	Note the following.
	The description of the patent allows for the terms to define a point in time 'on or after the expiration timestamp or at a time or upon an event as defined by the terms'. In this instance the event is the reception of the transaction for processing.
	Patent references: [0123] 22. On or after the expiration timestamp or at a time or upon an event as defined by the terms, and before the lock time of the complete refund transaction record,
	[0186] 21. On or after the expiration timestamp, or at a time or upon an event as defined by the terms, and before the lock time of the complete refund transaction record,
• a computer processor the memory and the ne interface, the compute processor configured t	<i>twork</i> Computing Device) such as a computer or mobile phone.
i. read the first private k the memory;	

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 39 of 117 PageID #: 101

	The validator identity is a system account that is used to pay for all the vote transaction fees submitted to the vote account. Because the validator is expected to vote on most valid blocks it receives, the validator identity account is frequently (potentially multiple times per second) signing transactions and paying fees. For this reason the validator identity keypair must be stored as a "hot wallet" in a keypair file on the same system the validator process is running. <a href="https://github.com/solana-labs/solana/blob/master/programs/vote/src/vote_transaction.rs#L17">Validator Code Reference</a>
ii. compute a first cryptographic signature from the first private key;	The hardware device running the Coinbase Wallet software on a computer or mobile phone. Coinbase Wallet app requesting vertification to access (read) the private key calculate a signature to autorize transaction.         Image: Confirm send       Image: Confirm send         Image: Confirm send       Image: Confirm send <tr< td=""></tr<>

	The validator identity is a system account that is used to pay for all the vote transaction fees submitted to the vote account. Because the validator is expected to vote on most valid blocks it receives, the validator identity account is frequently (potentially multiple times per second) signing transactions and paying fees. For this reason the validator identity keypair must be stored as a "hot wallet" in a keypair file on the same system the validator process is running.           Validator Code Reference           https://github.com/solana-labs/solana/blob/master/programs/vote/src/vote_transaction.rs#L39
iii. create an inchoate data record comprising:	The First Client as part of the Computing Device creates a transaction message to be sent to the Solana Network. RPC JSON API calls are defined to allow interaction with the Solana Network. The App constructs the following RPC message.
	sendTransaction
	<ul> <li>Object - The transaction object <ol> <li>header - The message header contains three unsigned 8-bit values. The first value is the number of required signatures in the containing transaction. The second value is the number of those corresponding account addresses that are read-only. The third value in the message header is the number of read-only account addresses not requiring signatures.</li> <li>Account addresses - The addresses that require signatures appear at the beginning of the account address array, with addresses requesting write access first and read-only accounts following. The addresses that do not require signatures follow the addresses that do not require signatures follow the addresses that do not require signatures follow the addresses that do not require signatures following. The addresses that do not require signatures following. The addresses that do not require signatures following. The addresses that do not require signatures to indicate when a client last observed the ledger. Validators will reject transactions when the blockhash is too old</li> <li>Instructions - An instruction contains a program id index, followed by a compact-array of opaque 8-bit data. The program id index is used to identify an on-chain program that can interpret the opaque data. The program id index is an unsigned 8-bit index to an account address in the message's array of account addresses. The account address in the message's array of account addresses. The account address in the amessage's array of account addresses. The account address in the amessage's array of account addresses. The account address in the amessage's array of account addresses. The account address in the same array.</li> </ol> </li> </ul>
	Official Documentation Reference https://docs.solana.com/developing/programming-model/transactions https://docs.solana.com/developing/clients/jsonrpc- api#sendtransaction
	<u>RPC Code Reference</u> <u>https://github.com/solana-</u> labs/solana/blob/master/client/src/nonblocking/rpc_client.rs#L128

	https://github.com/solana- labs/solana/blob/master/client/src/nonblocking/rpc_client.rs#L640 https://github.com/solana- labs/solana/blob/master/client/src/nonblocking/rpc_client.rs#L648
	Validator Code Reference <u>https://github.com/solana-</u> labs/solana/blob/master/programs/vote/src/vote_transaction.rs#L11
<ul> <li>a commit input for receiving a commit data from a commit transaction;</li> </ul>	The commit input is the primary account address which will pay the transaction fee.
	Transactions are required to have at least one account which has signed the transaction and is writable. Writable signer accounts are serialized first in the list of transaction accounts and the first of these accounts is always used as the "fee payer".
	Note that the Client can use the JSON RPC API to query the cluster for the current fee parameters. (getFeeForMessage) to determine the total fee payable to ensure that the paying account has the required amount to pay.
	Official Documentation Reference https://docs.solana.com/developing/clients/jsonrpc- api#getfeeformessage
	RPC Code Reference         The transaction sent to the RPC Client contains the Transaction         generated by the App.         https://github.com/solana-         labs/solana/blob/master/client/src/nonblocking/rpc_client.rs#L128         https://github.com/solana-         labs/solana/blob/master/client/src/nonblocking/rpc_client.rs#L640
	The commit input is the primary account address which will pay the transaction fee.
	The validator identity is a system account that is used to pay for all the vote transaction fees submitted to the vote account. Because the validator is expected to vote on most valid blocks it receives, the validator identity account is frequently (potentially multiple times per second) signing transactions and paying fees. For this reason the validator identity keypair must be stored as a "hot wallet" in a keypair file on the same system the validator process is running. https://docs.solana.com/running-validator/vote-accounts#validator-identity
	<u>Validator Code Reference</u> <u>https://github.com/solana-</u> <u>labs/solana/blob/master/programs/vote/src/vote_transaction.rs#L24</u> <u>https://github.com/solana-</u> <u>labs/solana/blob/master/programs/vote/src/vote_transaction.rs#L31</u>

• one or more output data obtained from at least one of the first principal data or the second principal data, and a value data from at least one of the first data source or the second data source; and	<ul> <li>The output data included in the transaction message is as follows.</li> <li>The list of signatures in the message. The Validator leader will use this, along with the clusters current lampoons per signature, to calculate the total transaction fee.</li> <li><u>Official Documentation Reference</u> <u>https://docs.solana.com/implemented-proposals/transaction-fees</u></li> </ul>
	<u>RPC Code Reference</u> The transaction sent to the RPC Client contains the Transaction         generated by the App. <u>https://github.com/solana-</u> labs/solana/blob/master/client/src/nonblocking/rpc_client.rs#L128 <u>https://github.com/solana-</u> labs/solana/blob/master/client/src/nonblocking/rpc_client.rs#L640
	<u>Validator Code Reference</u> <u>https://github.com/solana-</u> <u>labs/solana/blob/master/programs/vote/src/vote_transaction.rs#L36</u> <u>https://github.com/solana-</u> <u>labs/solana/blob/master/programs/vote/src/vote_transaction.rs#L38</u> <u>https://github.com/solana-</u> <u>labs/solana/blob/master/programs/vote/src/vote_transaction.rs#L39</u>
• the first cryptographic signature; and	The hardware device running the Coinbase Wallet software on a computer or mobile phone. Coinbase Wallet app requesting verification to access (read) the private key to calculate a signature to authorize transaction.
	Wallet used26vuBUdwkPNetworkSolana mainnetNetwork fee •\$0.00 • 0.000005 SOLTotal cost\$0.10
	Confirm

## Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 43 of 117 PageID #: 105

	The Coinbase Wallet creates a transaction message to be sent to the Coinbase Solana Node. RPC JSON API calls are defined to allow interaction with the Solana Nodes. The Coinbase Wallet constructs the following RPC message.
	sendTransaction
	Submits a signed transaction to the cluster for processing.
	Before submitting, the following preflight checks are performed:
	The transaction signatures are verified The transaction is simulated against the bank slot specified by the preflight commitment. On failure an error will be returned. Preflight checks may be disabled if desired. It is recommended to specify the same commitment and preflight commitment to avoid confusing behavior. The returned signature is the first signature in the transaction, which is used to identify the transaction (transaction id). This identifier can be easily extracted from the transaction data before submission.
	Official Documentation Reference https://docs.solana.com/developing/clients/jsonrpc- api#sendtransaction https://docs.solana.com/developing/programming- model/transactions#signatures
	https://solanacookbook.com/core-concepts/transactions.html#fees
	RPC Code Reference         https://github.com/solana-         labs/solana/blob/master/client/src/nonblocking/rpc_client.rs#L129         https://github.com/solana-         labs/solana/blob/master/client/src/nonblocking/rpc_client.rs#L813         https://github.com/solana-         labs/solana/blob/master/client/src/nonblocking/rpc_client.rs#L654
	Validator Code Reference https://github.com/solana- labs/solana/blob/master/programs/vote/src/vote_transaction.rs#L38 https://github.com/solana- labs/solana/blob/master/programs/vote/src/vote_transaction.rs#L39
<i>iv.</i> publish the inchoate data record to at least one of the first client device or the second client device,	Given the Computing Device and the First Client are the same device, this clause is not applicable. The First Client as part of the Computing Device does create a transaction message (where the signed inchoate transaction record = complete transaction record).
	Official Documentation Reference https://docs.solana.com/developing/clients/jsonrpc- api#sendtransaction
	RPC Source Code

	https://github.com/solana- labs/solana/blob/master/client/src/nonblocking/rpc_client.rs#L935
	aus/solana/0100/master/enent/src/honolocking/rpc_chent.rs#L935
	Validator Code Reference
	https://github.com/solana-
	labs/solana/blob/master/gossip/src/cluster_info.rs#L1087
wherein the decentralized digital currency comprises a distributed ledger that enables processing the transaction between the first client device and the second client device without the need for a trusted central authority,	A SOL is the name of Solana's native token, which can be passed to nodes in a Solana cluster in exchange for running an on-chain program or validating its output. The system may perform micropayments of fractional SOLs, which are called lamports. https://docs.solana.com/introduction#what-are-sols
wherein the inchoate data record is used by	The Coinbase Wallet creates a transaction message to be sent to the
at least one of the first client device or the second client device to create a complete data record and to create the transaction by broadcasting the complete data record for transmitting and receiving among network participants in the computer network for recording in the distributed ledger,	Coinbase Solana Node. RPC JSON API calls are defined to allow interaction with the Solana Nodes.
	Only the First Client is required to sign the transaction and so once signed the RPC JSON API call ' <b>sendTransaction</b> ' is considered to be a complete data record and is broadcast to the Leader Validator via the Coinbase Solana Node.
	Official Documentation Reference https://docs.solana.com/developing/clients/jsonrpc- api#sendtransaction
	<u>RPC Source Code</u> <u>https://github.com/solana-</u> <u>labs/solana/blob/master/client/src/nonblocking/rpc_client.rs#L935</u>
	Validator Code Reference <u>https://github.com/solana-</u> <u>labs/solana/blob/master/gossip/src/cluster_info.rs#L1087</u>
wherein at least one of the first client device or the second client device signs the inchoate data record and saves a copy of the inchoate data record on at least one of the first client device or the second client device; and	The Coinbase Wallet creates a transaction message to be sent to the Coinbase Solana Node. RPC JSON API calls are defined to allow interaction with the Solana Nodes.
	Only the First Client is required to sign the transaction and so once signed the RPC JSON API call ' <b>sendTransaction</b> ' is considered to be a complete data record and is broadcast to the Leader Validator via the Coinbase Solana Node.
	Official Documentation Reference https://docs.solana.com/developing/clients/jsonrpc- api#sendtransaction
	<u>RPC Source Code</u> <u>https://github.com/solana-</u> <u>labs/solana/blob/master/client/src/nonblocking/rpc_client.rs#L935</u>
	<u>Validator Code Reference</u> <u>https://github.com/solana-</u> <u>labs/solana/blob/master/programs/vote/src/vote_transaction.rs#L38</u>

	https://github.com/solana- labs/solana/blob/master/programs/vote/src/vote_transaction.rs#L39
	A copy of the inchoate data record is (optionally, as described in pgs 11 and 15 of the general description text of the patent) saved on the Leader Validator node.
wherein the at least one of the computing device, the first client device, or the second client device verifies the recording of the complete data record in the distributed ledger by observing an external state.	The Coinbase Wallet shows the transaction history. Transactions History
	2pUa7qKN6u         -\$0.10           Send • 6:36 PM         0.0813 SRM
	2pUa7qKNóu         -\$0.10           Send • 2:18 PM         0.082 SRM
	Ox6237c68b         +\$0.69           Receive • 2:06 PM         1 MATIC
	0x6237c68b         +\$0.33           Receive • 2:06 PM         0.01 AVAX
	EUhg6JbNte         +\$0.62           Receive + 2:06 PM         0.5 RAY           0x6237c68b         +\$5.05
	Keceive • 2:03 PM 0.00249 ETH
	Send * May 13, 2022     0.01 SOL       O     Image: Constraint of the second s
	The Coinbase Wallet shows the transaction details along with a link to the Solana explorer.

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 46 of 117 PageID #: 108

÷	Sent Serum
	0.0813 SRM
	\$0.10
From	AFdowYci5G
То	2pUa7qKN6u
Date	06:36 PM • May 16, 2022
Network	Solana mainnet
Status	Completed
Value	\$0.10 • 0.0813 SRM
Network fee	\$0.00
Total cost	\$0.10
View	on solana explorer
Close 🔒 ex	plorer.solana.com 🗚 🖒
<b>SOLANA</b>	
	⊘ Mainnet Beta
Search for bloc	
transactions, p	rograms, and tokens
Transaction	
Overview	Inspect C Refresh
Signature	凸 37VXwoatrQiTZsK6zZ
Result	
Timestamp	May 16, 2022 at 18:36
Confirmation St	atus FINALIZED
Confirmations	
	> 🖞 🥝

## Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 47 of 117 PageID #: 109

Claim Chart U.S. Patent No. 11	1,196,566 (the "'566 Patent") Coinbase
<u>Claim 2</u>	Coinbase Products & Services
	Payment to Validator Node from a transaction (that incurs a
	transaction fee) on the Solana Network.
The device of claim 1, where: the computer processor is configured to obtain the one or more output data based on:	<ul> <li>The following disbursements are made.</li> <li>1. The transaction fee is withdrawn from the senders account by the Lead Validator as part of processing the transaction and (likely) inclusion into the Block.</li> <li>2. The transaction fees collected from all processed transactions are banked for disbursement at the end of the epoch.</li> <li>3. The transaction fees collected are disbursed to the Lead Validators vote account based on the accounts commission rate with the remainder of the rewards being distributed to all of the stake accounts delegated to that vote account, proportional to the active stake weight of each stake account.</li> </ul>
the first principal data; and the value data from the first data source.	Before any transaction instructions are processed, the fee payer account balance will be deducted to pay for transaction fees. If the fee payer balance is not sufficient to cover transaction fees, the transaction will be dropped by the cluster. If the balance was sufficient, the fees will be deducted whether the transaction is processed successfully or not. In fact, if any of the transaction instructions return an error or violate runtime restrictions, all account changes *except* the transaction fee deduction will be rolled back.         https://solanacookbook.com/core-concepts/transactions.html#fees         https://jstarry.notion.site/Transaction-Fees-f09387e6a8d84287aa16a34ecb58e239         The collected transaction fees (of those transactions included in the Block) by the Lead Validator are: <ol> <li>burnt based on the genesis burnt rate (second data source)</li> <li>written to the Validators vote account for disbursement.</li> </ol> Source Code – Calculation of fee for the individual transaction using the first Data Source         https://github.com/solana-labs/solana/blob/master/runtime/src/bank.rs#L4805         https://github.com/solana-labs/solana/blob/master/runtime/src/bank.rs#L4718
	Source Code – Withdraw transaction fee from senders account https://github.com/solana- labs/solana/blob/master/runtime/src/bank.rs#L4822 https://github.com/solana- labs/solana/blob/master/runtime/src/bank.rs#L6256

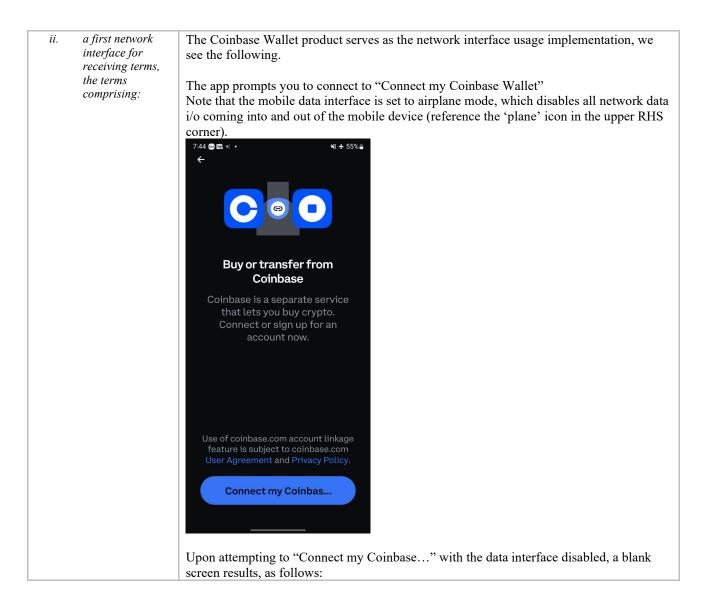
Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 48 of 117 PageID #: 110

Source Code – Burning of a percentage of all fees collected using the
second Data Source
https://github.com/solana-
labs/solana/blob/master/runtime/src/bank.rs#L3280
https://github.com/solana-
labs/solana/blob/master/runtime/src/bank.rs#L3284

### Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 49 of 117 PageID #: 111

	hart U.S. Patent No. 11,196,566 (the "'566 Patent") Coinbase
<u>Claim 7</u>	Coinbase Products & Services
A system for processing	Payment to Validator Node from a transaction (that incurs a transaction fee)
a transaction between a	on the Solana Network.
first client device and a	
second client device via	
a transfer mechanism the	
system comprising a	
computing device, the	
first client device, the	
second client device, and	
the transfer mechanism.	
7. a. the computing device	The First Client, as part of the Facilitator, need a computer hardware/software combinatio
comprising:	to run, namely:
i. a first memory comprising for	• Memory (RAM), used in the computing device (such as a computer or mobile phone).
storing a first	<ul> <li>Transaction record sector (stores transactions that haven't been submitted to the</li> </ul>
asymmetric key pair, the first asymmetric key	blockchain yet) kept via the crypto software wallets
pair comprising	Where the First Client runs Coinbase Wallet software to sign transactions, contains:
a first private	<ul> <li>a first key pair sector which is generated and stored in the wallet software</li> </ul>
key and a first	<ul> <li>The asymmetric key pair generated and/or stored consists of a first private key</li> </ul>
public key;	and a first public key – all found and manipulated via the wallet software.
	• The wallet software connects via the public key or the key pair, and authorizes
	(signs) the transaction with the private key of the key pair.
	Where the First Client is a Coinbase Solana Node Client Browser or Command Line Interface to transaction creation, contains:
	• a first key pair sector which is generated and stored on the device
	• The asymmetric key pair stored consists of a first private key and a first public key.
	Where the First Client utilises Coinbase Private key management in order to sign
	transactions, contains:
	• a first key pair sector which is stored in the key management vault/software
	• The asymmetric key pair generated and/or stored consists of a first private key
	and a first public key – all found and manipulated via the key management
	vault/software.
	• The key management vault/software connects via the public key or the key pair, and authorizes (signs) the transaction with the private key of the key pair.
	Where the First Client is a Coinbase Validator Node with a vote account key pair to sign
	voting transactions contains
	• a first key pair sector which is generated and stored on the device
	• The asymmetric key pair stored consists of a first private key and a first public key
	Source Code
	https://github.com/solana-labs/solana/blob/master/validator/src/main.rs#L2999

#### Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 50 of 117 PageID #: 112



Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 51 of 117 PageID #: 113

7:44 ₪ 🖱 🖬 • 🛛 📲 ★ 55% 🚔 ← Sign in
The Ceichese Wellet and interfaces with nodes and servers in the Ceichese infrastructure
The Coinbase Wallet app interfaces with nodes and servers in the Coinbase infrastructure to query information or connect to a client. To fetch data like user identification verification credentials, token swap rates, token balances, and other information from the Coinbase infrastructure servers, nodes and smart contracts, such infrastructure responds to queries through the Coinbase Wallet app, the network interface,
Once the network interface is enabled, as shown by the wi-fi and cellular service icons in the upper right-hand corner that have replaced the airplane mode symbol, the interface for entering and submitting user credentials is available after querying Coinbase's servers. The interface actually states that "you will be redirected to <u>https://wallet.coinbase.com</u>
 (their website) after the authorization"

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 52 of 117 PageID #: 114

7:47 ▲ ≰ 🛤 • 🛞 ¥ 🖘 ⊒ 54% 🚔 🔶 Sign in
coinbase
by Coinbase Sign in to your Coinbase account to authorize <b>Coinbase Wallet</b> (Android)
You will be redirected to https://wallet.coinbase.com after the authorization.
Email you@example.com Password
SIGN IN
Once the Coinbase Wallet app is connected, it is able to query Coinbase infrastructure and populate the app with a plethora of information.
Choosing the "Send" option as an illustration, we can opt to send SOL to any address.

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 53 of 117 PageID #: 115

	÷	Send Crypto		
	Max	<b>\$0.0813 SRM</b> \$4.18 available	*	
		erum bur balance	<b>\$4.18</b> 3.396 SRM	
	1	2	3	
	4	5	6	
	7	8	9	
	•	0	÷	
T	he Coinba	<b>Continue</b> ase Wallet app popul	ates the app	with prices, quantities and transaction fees.

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 54 of 117 PageID #: 116

	← Confirm send
	Serum \$0.10 0.0813 SRM
	Wallet used 26vuBUdwkP
	Network Solana mainnet
	Network fee ● \$0.00 • 0.000005 SOL
	Total cost \$0.10 Confirm
1	Note below, with the airplane mode turned on to deprive the Coinbase Wallet app access to the Coinbase infrastructure results in a very different outcome when one attempts to perform the send transaction

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 55 of 117 PageID #: 117

Serum         \$4.18           Your balance         3.396 SRM           1         2         3           4         5         6           7         8         9           .         0         €	← Max	Send Crypto \$0.1 0.0813 SRM \$4.18 available	<b>*</b>	
4 5 6 7 8 9 . 0 ←				
7 8 9 . 0 ←		2	3	
. 0 <i>←</i>		5	6	
	7	8	9	
Continue		0	÷	
An error dialog is given after the "continue" button is depressed.				

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 56 of 117 PageID #: 118

	7:55 혐로 ▲ • 책 ★ 52% 🚔
	←
	Something went wrong
	Try again
A. at least one of a	First principle data
first principal data or a second	A transaction fee provided by the Client to pay for sending a transaction.
principal data;	Note: Before any transaction instructions are processed, the fee payer account balance will
	be deducted to pay for transaction fees. If the fee payer balance is not sufficient to cover transaction fees, the transaction will be dropped by the cluster. If the balance was
	sufficient, the fees will be deducted whether the transaction is processed successfully or not. In fact, if any of the transaction instructions return an error or violate runtime
	restrictions, all account changes *except* the transaction fee deduction will be rolled back.
	https://solanacookbook.com/core-concepts/transactions.html#fees
	https://jstarry.notion.site/Transaction-Fees-f09387e6a8d84287aa16a34ecb58e239
B. a reference to at	First data source
least one of a first data source	The current lampoons per signature
or a second data source; and	Each validator uses signatures per slot (SPS) to estimate network congestion and SPS
	target to estimate the desired processing capacity of the cluster. The validator learns the SPS target from the genesis config, whereas it calculates SPS from recently processed
	transactions. The genesis config also defines a target lamports_per_signature, which is the fee to charge per signature when the cluster is operating at SPS target.
	https://docs.solana.com/implemented-proposals/transaction-fees#congestion-driven-fees

	Note that the Client uses the JSON RPC API to query the cluster for the current fee parameters. getFeeForMessage https://docs.solana.com/developing/clients/jsonrpc-api#getfeeformessage Second data source The portion of the transaction fee payable to the Leader. 50% of each transaction fee is burned, with the remaining fee retained by the validator that processes the transaction. https://docs.solana.com/inflation/terminology#effective-inflation-rate-
C. an expiration timestamp,	<ul> <li>1. The transaction needs to be performed within a timeframe.</li> <li>A transaction includes a recent blockhash to prevent duplication and to give transactions lifetimes. Any transaction that is completely identical to a previous one is rejected, so adding a newer blockhash allows multiple transactions to repeat the exact same action. Transactions also have lifetimes that are defined by the blockhash, as any transaction whose blockhash is too old will be rejected.</li> <li>https://docs.solana.com/developing/programming-model/transactions#recent-blockhash</li> <li>A blockhash contains a 32-byte SHA-256 hash. It is used to indicate when a client last observed the ledger. Validators will reject transactions when the blockhash is too old.</li> <li>https://docs.solana.com/developing/programming-model/transactions#blockhash-format</li> <li>Source Code</li> <li>Calculation of blockhash as part of transaction creation.</li> <li>https://github.com/solana-</li> <li>labs/solana/blob/master/client/sre/nonblocking/rpc_client.rs#L656</li> <li>2. The Leader Validator processes the disbursement amount on receipt of the voting transaction fees. If the fee payer account balance will be deducted to pay for transaction fees. If the fee payer balance is not sufficient to cover transaction fees will be deducted whether the transaction is processefully or not. In fact, if any of the transaction instructions return an error or violate runtime restrictions, all account changes *except* the transaction fee deduction will be rolled back.</li> <li>https://solanacookbook.com/core-concepts/transactions.html#fees</li> <li>https://jstarry.notion.site/Transaction-Fees-f09387e6a8d84287aa16a34ecb58e239</li> <li>Note the following.</li> <li>The description of the patent allows for the terms to define a point in time 'on or after the expiration timestamp or at a time or upon an event as defined by the terms In this instance the event is the reception of the transaction for processing.</li> </ul>
	Patent references:

		[0123] 22. On or after the expiration timestamp or at a time or upon an event as defined by the terms, and before the lock time of the complete refund transaction record,
		[0186] 21. On or after the expiration timestamp, or at a time or upon an event as defined by the terms, and before the lock time of the complete refund transaction record,
iii.	a first computer processor coupled to the first memory and the first network interface, the first computer processor configured to:	The hardware device running the App (First Client as part of the Facilitator) such as a computer or mobile phone.
А.	read the first private key from	The hardware device running the Coinbase Wallet software on a computer or mobile phone. Coinbase Wallet app requesting verification to access (read) the private key to
	the first memory	authorize transaction.
		Serum \$0.10 0.0813 SRM
		4
		@wallettest1 5p2cRjXgbg
		Wallet used 26vuBUdwkP
		Network Solana mainnet
		Network fee  \$0.00 • 0.000005 SOL
		Total cost \$0.10
		Confirm
		<u>RPC Source Code</u>
		https://github.com/solana- labs/solana/blob/master/client/src/nonblocking/rpc_client.rs#L654
		Validator Code Reference

	https://github.com/solana-
B. compute a first	labs/solana/blob/master/programs/vote/src/vote_transaction.rs#L17           The hardware device running the Coinbase Wallet software on a computer or mobile
cryptographic signature from	phone. Coinbase Wallet app requesting verification to access (read) the private key to
the first private	calculate a signature to authorize transaction.
key;	← Confirm send
	\$0.10
	Constant Serum 0.0813 SRM
	@wallettest1 5p2cRjXgbg
	SpeckjAgby
	Wallet used 26vuBUdwkP
	Network Solana mainnet
	Network fee ● \$0.00 • 0.000005 SOL
	Total cost \$0.10
	Confirm
	Contraint
	RPC Source Code
	https://github.com/solana- labs/solana/blob/master/client/src/nonblocking/rpc_client.rs#L129
	https://github.com/solana-
	labs/solana/blob/master/client/src/nonblocking/rpc_client.rs#L813 https://github.com/solana-
	labs/solana/blob/master/client/src/nonblocking/rpc_client.rs#L654
	Validator Code Reference
	https://github.com/solana- labs/solana/blob/master/programs/vote/src/vote_transaction.rs#L39
<i>C. create an inchoate</i>	The App (First Client as part of the Facilitator) creates a transaction message to be sent
data record comprising:	to the Solana Network. RPC JSON API calls are defined to allow interaction with the Solana Network. The App constructs the following RPC message.
	sendTransaction
	Object - The transaction object

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 60 of 117 PageID #: 122

	<ol> <li>header - The message header contains three unsigned 8-bit values. The first value is the number of required signatures in the containing transaction. The second value is the number of those corresponding account addresses that are read-only. The third value in the message header is the number of read-only account addresses not requiring signatures.</li> <li>Account addresses - The addresses that require signatures appear at the beginning of the account addresses that do, again with read-write access first and read-only accounts following. The addresses that do not require signatures follow the addresses that do, again with read-write accounts first and read-only accounts following. The addresses that do not require signatures follow the addresses that do, again with read-write accounts first and read-only accounts following.</li> <li>Blockhash - A blockhash contains a 32-byte SHA-256 hash. It is used to indicate when a client last observed the ledger. Validators will reject transactions when the blockhash is too old</li> <li>Instructions - An instruction contains a program id index, followed by a compact-array of account address indexes, followed by a compact-array of opaque 8-bit data. The program id index is used to identify an on-chain program that can interpret the opaque data. The program id index is an unsigned 8-bit index to an account address in the message's array of account addresses. The account address indexes are each an unsigned 8-bit index into that same array.</li> <li>Official Documentation Reference https://github.com/solana-labs/solana/blob/master/client/src/nonblocking/rpc_client.rs#L128 https://github.com/solana-labs/solana/blob/master/client/src/nonblocking/rpc_client.rs#L640 https://github.com/solana-labs/solana/blob/master/client/src/nonblocking/rpc_client.rs#L648</li> <li>Validator Code Reference</li> </ol>
	https://github.com/solana- labs/solana/blob/master/programs/vote/src/vote_transaction.rs#L11
I. a commit input	The commit input is the primary account address which will pay the transaction fee.
for receiving a commit data from a commit transaction;	Transactions are required to have at least one account which has signed the transaction and is writable. Writable signer accounts are serialized first in the list of transaction accounts and the first of these accounts is always used as the "fee payer". Note that the Client can use the JSON RPC API to query the cluster for the current fee parameters. (getFeeForMessage) to determine the total fee payable to ensure that the paying account has the required amount to pay. https://docs.solana.com/developing/clients/jsonrpc-api#getfeeformessage
	<u>RPC Code Reference</u> The transaction sent to the RPC Client contains the Transaction generated by the App. <u>https://github.com/solana-</u> <u>abs/solana/blob/master/client/src/nonblocking/rpc_client.rs#L128</u> <u>https://github.com/solana-</u> <u>abs/solana/blob/master/client/src/nonblocking/rpc_client.rs#L640</u>

## Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 61 of 117 PageID #: 123

	The commit input is the primary account address which will pay the transaction fee. The validator identity is a system account that is used to pay for all the vote transaction fees submitted to the vote account. Because the validator is expected to vote on most valid blocks it receives, the validator identity account is frequently (potentially multiple times per second) signing transactions and paying fees. For this reason the validator identity keypair must be stored as a "hot wallet" in a keypair file on the same system the validator process is running. https://docs.solana.com/running-validator/vote-accounts#validator-identity Validator Code Reference https://github.com/solana-
	labs/solana/blob/master/programs/vote/src/vote_transaction.rs#L24         https://github.com/solana-         labs/solana/blob/master/programs/vote/src/vote_transaction.rs#L31
II. one or more outputs obtained from at least one of the first principal data or the second principal data, and a value data from at least one of the first data source or the second data source; and;	The output data included in the transaction message is as follows.         • The list of signatures in the message. The Validator leader will use this, along with the clusters current lampoons per signature, to calculate the total transaction fee.         Official Documentation Reference         https://docs.solana.com/implemented-proposals/transaction-fees#congestion-driven-fees         RPC Code Reference         The transaction sent to the RPC Client contains the Transaction generated by the App.         https://github.com/solana-         labs/solana/blob/master/client/src/nonblocking/rpc_client.rs#L128         https://github.com/solana-         labs/solana/blob/master/client/src/nonblocking/rpc_client.rs#L640
	Validator Code Reference         https://github.com/solana-         labs/solana/blob/master/programs/vote/src/vote_transaction.rs#L36         https://github.com/solana-         labs/solana/blob/master/programs/vote/src/vote_transaction.rs#L38         https://github.com/solana-         labs/solana/blob/master/programs/vote/src/vote_transaction.rs#L38         https://github.com/solana-         labs/solana/blob/master/programs/vote/src/vote_transaction.rs#L38
III. The first cryptographic signature; and	The hardware device running the Coinbase Wallet software on a computer or mobile phone. Coinbase Wallet app requesting verification to access (read) the private key to calculate a signature to authorize transaction.

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 62 of 117 PageID #: 124

#### EXHIBIT 3

← Cor	nfirm send		
Serum	<b>\$0.10</b> 0.0813 SRM		
@wallet 5p2cRj			
Wallet used	26vuBUdwkP		
Network	Solana mainnet		
Network fee 🖲	\$0.00 • 0.000005 SOL		
Total cost	\$0.10		
	Confirm		
Node. RPC JSON A		essage to be sent to the Coinl ow interaction with the Sola C message.	
sendTransaction			

Submits a signed transaction to the cluster for processing.

Before submitting, the following preflight checks are performed:

The transaction signatures are verified

The transaction is simulated against the bank slot specified by the preflight commitment. On failure an error will be returned. Preflight checks may be disabled if desired. It is recommended to specify the same commitment and preflight commitment to avoid confusing behavior.

The returned signature is the first signature in the transaction, which is used to identify the transaction (transaction id). This identifier can be easily extracted from the transaction data before submission.

Official Documentation Reference https://docs.solana.com/developing/clients/jsonrpc-api#sendtransaction https://docs.solana.com/developing/programming-model/transactions#signatures https://solanacookbook.com/core-concepts/transactions.html#fees

		RPC Code Reference         https://github.com/solana-         labs/solana/blob/master/client/src/nonblocking/rpc_client.rs#L129         https://github.com/solana-         labs/solana/blob/master/client/src/nonblocking/rpc_client.rs#L813         https://github.com/solana-         labs/solana/blob/master/client/src/nonblocking/rpc_client.rs#L654         Validator Code Reference         https://github.com/solana-         labs/solana/blob/master/programs/vote/src/vote_transaction.rs#L38         https://github.com/solana-         labs/solana/blob/master/programs/vote/src/vote_transaction.rs#L38
D.	publish the inchoate data record to at least one of the first client device or the second client device;;	Given the Facilitator and the First Client are the same device, this clause is not necessarily applicable. The App (First Client as part of the Facilitator) does create a transaction message (where the signed inchoate transaction record = complete transaction record) to the Solana Network and the Second Client. <u>Official Documentation Reference https://docs.solana.com/developing/clients/jsonrpc-api#sendtransaction RPC Source Code https://github.com/solana-labs/solana/blob/master/client/src/nonblocking/rpc_client.rs#L935</u>
7. b. the comprise i.	first client es: a second memory for storing a second asymmetric key pair, the second asymmetric key pair comprising a second private key and a second public key.	The Facilitator and the First Client are both considered to be the same device; namely the App or mechanism that is using the Solana Network by sending a transaction. Thus, the clause is not assessable as it is covered by the above clauses 7a.i.
ii.	a second network interface; and	The Facilitator and the First Client are both considered to be the same device; namely the App or mechanism that is using the Solana Network by sending a transaction. Thus, the clause is not assessable as it is covered by the above clauses 7a.ii.
iii.	a second computer processor coupled to the second memory and the second network interface, the second computer processor configured to:	The Facilitator and the First Client are both considered to be the same device; namely the App or mechanism that is using the Solana Network by sending a transaction. Thus, the clause is not assessable as it is covered by the above clauses 7a.iii.A-E.

## Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 64 of 117 PageID #: 126

А.	read the second private key from the second memory;	The Facilitator and the First Client are both considered to be the same device; namely the App or mechanism that is using the Solana Network by sending a transaction. Thus, the clause is not assessable as it is covered by the above clauses 7a.iii.A.
В.	read the inchoate data record	The Facilitator and the First Client are both considered to be the same device; namely the App or mechanism that is using the Solana Network by sending a transaction. Thus, the clause is not assessable.
С.	compute a second cryptographic signature from the second private key;	The Facilitator and the First Client are both considered to be the same device; namely the App or mechanism that is using the Solana Network by sending a transaction. Thus, the clause is not assessable as it is covered by the above clauses 7a.iii.B.
D. I.	create a complete data record comprising: the commit input;	The Facilitator and the First Client are both considered to be the same device; namely the App or mechanism that is using the Solana Network by sending a transaction. Thus, the clause is not assessable as it is covered by the above clauses 7a.iii.C.
II.	the output data;	The Facilitator and the First Client are both considered to be the same device; namely the App or mechanism that is using the Solana Network by sending a transaction. Thus, the clause is not assessable as it is covered by the above clauses 7a.iii.C.
III.	the first cryptographic signature, and	The Facilitator and the First Client are both considered to be the same device; namely the App or mechanism that is using the Solana Network by sending a transaction. Thus, the clause is not assessable as it is covered by the above clauses 7a.iii.C.
IV.	the second cryptographic signature,	The Facilitator and the First Client are both considered to be the same device; namely the App or mechanism that is using the Solana Network by sending a transaction. Thus, the clause is not assessable as it is covered by the above clauses 7a.iii.C.
Е.	create a transaction by submitting the complete data record to the transfer mechanism;	The Coinbase Wallet creates a transaction message to be sent to the Coinbase Solana         Node. RPC JSON API calls are defined to allow interaction with the Solana Nodes.         Only the First Client is required to sign the transaction and so once signed the RPC JSON         API call 'sendTransaction' is considered to be a complete data record and is broadcast to the Leader Validator via the Coinbase Solana Node.         Official Documentation Reference         https://docs.solana.com/developing/clients/jsonrpc-api#sendtransaction         RPC Source Code         https://github.com/solana-labs/solana/blob/master/client/src/nonblocking/rpc_client.rs#L935
		The Leader Validator, as the second client, uses the transaction (completed disbursement transaction record) and other transactions in the TX Pool to create a Block Transaction. The Block is broadcast to all network participants for consensus and recording to the distributed ledger.

	Clients send transactions to any validator's Transaction Processing Unit (TPU) port. If th node is in the validator role, it forwards the transaction to the designated leader. If in the leader role, the node bundles incoming transactions, timestamps them creating an entry, and pushes them onto the cluster's data plane. Once on the data plane, the transactions are validated by validator nodes, effectively appending them to the ledger.
	https://docs.solana.com/cluster/overview#sending-transactions-to-a-cluster
7. c. the second client comprises: i. a third memory	The Validator Note (including the Leader Validator) requires an account (with key pair) in order to sign the produced blocks.
for storing a third	7 System Architecture
asymmetric key	7.1 Components
pair, the third asymmetric key pair comprising a third private key and a third public key;	7.1.1 Leader, Proof of History generator The Leader is an elected Proof of History generator. It consumes arbitrary user transactions and outputs a Proof of History sequence of all the transactions that guarantee a unique global order in the system. After each batch of transactions the Leade outputs a signature of the state that is the result of running the transactions in that order. This signature is signed with the identity of the Leader.
	https://cryptorating.eu/whitepapers/Solana/solana-whitepaper-en.pdf
	Vote Authority#
	The vote authority keypair is used to sign each vote transaction the validator node wants
	to submit to the cluster. This doesn't necessarily have to be unique from the validator
	<i>identity, as you will see later in this document. Because the vote authority, like the validator identity, is signing transactions frequently, this also must be a hot keypair on the same file system as the validator process.</i>
	https://docs.solana.com/running-validator/vote-accounts#vote-authority
	Source Code
	https://github.com/solana-labs/solana/blob/master/validator/src/main.rs#L2999
ii. a third network interface; and	The Validator Nodes require network interface in order for them to participate in Block Production and network consensus.
	Networking#
	Internet service should be at least 300Mbit/s symmetric, commercial. 1GBit/s preferred
	Port Forwarding# The following ports need to be open to the internet for both inbound and outbound
	It is not recommended to run a validator behind a NAT. Operators who choose to do so should be comfortable configuring their networking equipment and debugging any traversal issues on their own.
	Required# 8000-10000 TCP/UDP - P2P protocols (gossip, turbine, repair, etc). This can be limited to any free 12 port range withdynamic-port-range
	https://docs.solana.com/running-validator/validator-regs#networking

# Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 66 of 117 PageID #: 128

iii. a third computer processor	The Solana Network Node running the Validator software.
coupled to the third memory and the third network interface, the third computer processor configured to read the third	https://docs.solana.com/running-validator/validator-reqs
private key from the third	
memory; and wherein the at least one of	The Coinbase Wallet creates a transaction message to be sent to the Coinbase Solana
the first client device or the second client device signs the inchoate data record	Node. RPC JSON API calls are defined to allow interaction with the Solana Nodes.
and saves a copy of the inchoate data record on at least one of the first client device or the second client	Only the First Client is required to sign the transaction and so once signed the RPC JSON API call ' <b>sendTransaction</b> ' is considered to be a complete data record and is broadcast to the Leader Validator via the Coinbase Solana Node.
device of the second client device.	Official Documentation Reference
	https://docs.solana.com/developing/clients/jsonrpc-api#sendtransaction
	<u>RPC Source Code</u> <u>https://github.com/solana-</u> labs/solana/blob/master/client/src/nonblocking/rpc_client.rs#L935
	Validator Code Reference
	https://github.com/solana- labs/solana/blob/master/programs/vote/src/vote_transaction.rs#L38
	https://github.com/solana- labs/solana/blob/master/programs/vote/src/vote_transaction.rs#L39
	A copy of the inchoate data record is (optionally, as described in pgs 11 and 15 of the general description text of the patent) saved on the Leader Validator node.
wherein the transfer mechanism comprising a decentralized digital currency that comprises a	A SOL is the name of Solana's native token, which can be passed to nodes in a Solana cluster in exchange for running an on-chain program or validating its output. The system may perform micropayments of fractional SOLs, which are called lamports.
distributed ledger that enables processing the	https://docs.solana.com/introduction#what-are-sols
transaction between the first client device and the second client device without the need of a trusted central	
authority,	
wherein the transaction is created by broadcasting the complete data record for	The Coinbase Wallet creates a transaction message to be sent to the Coinbase Solana Node. RPC JSON API calls are defined to allow interaction with the Solana Nodes.
transmitting and receiving among network participants in the computer network for recording in the distributed ledger, and	Only the First Client is required to sign the transaction and so once signed the RPC JSON API call ' <b>sendTransaction</b> ' is considered to be a complete data record and is broadcast to the Leader Validator via the Coinbase Solana Node.

# Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 67 of 117 PageID #: 129

	Official Documentation Reference https://docs.solana.com/developing/clients/jsonrpc-api#sendtransaction		
	<u>RPC Source Code</u> <u>https://github.com/solana-</u> <u>labs/solana/blob/master/client/src/nonblocking/rpc_client.rs#L935</u>		
	Validator Code Reference https://github.com/solana-labs/solana/blob/master/gossip/src/cluster_info.rs#L1087		
wherein at least one of the computer device, the first client device, or the second client device verifies the	The Coinbase Wallet shows the transaction history.		
recording of the complete data record in the distributed ledger by observing an external state	Transactions History		
	2pUa7qKN6u         -\$0.10           Send • 6:36 PM         0.0813 SRM		
	2pUa7qKN6u         -\$0.10           Send • 2:18 PM         0.082 SRM		
	0x6237c68b         +\$0.69           Receive • 2:06 PM         1 MATIC		
	0x6237c68b         +\$0.33           Receive • 2:06 PM         0.01 AVAX		
	EUhG6JbN1e         +\$0.62           Receive • 2:06 PM         0.5 RAY		
	0x6237c68b         +\$5.05           Receive • 2:03 PM         0.00249 ETH		
	26vuBUdwkP         -\$0.54           Send • May 13, 2022         0.01 SOL		
	Assets Transactions Browser Settings		
	The Coinbase Wallet shows the transaction details along with a link to the Solana explorer.		

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 68 of 117 PageID #: 130

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 69 of 117 PageID #: 131

Result Success
Search for blocks, accounts, transactions, programs, and tokens Q DETAILS Transaction Overview © Inspect @ Refresh Signature © 37VXwoat r01TZsK6z2V Result Success Timestamp May 16, 2022 at 18:36
transactions, programs, and tokens
Transaction         Overview       Inspect         Signature       Inspect         Result       Success         Timestamp       May 16, 2022 at 18:36
Signature G 37VXwoatrQiTZsK6zZv Result Success Timestamp May 16, 2022 at 18:36
Result Success Timestamp May 16, 2022 at 18:36
Timestamp May 16, 2022 at 18:36
Confirmation Status FINALIZED

# Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 70 of 117 PageID #: 132

Claim Chart U.S. Patent No. 11,196,566 (the "'566 Patent") Coinbase			
Claim 1     Coinbase Products & Services			
A computing device for processing a transaction between a first client device, and a second client device via a transfer mechanism, the transfer mechanism comprising a decentralized digital currency	The transfer of a NFT (on the Ethereum network) from one party to another.		
The computing device comprising:	<ul> <li>The Computing Device   Facilitator consists of:</li> <li>the Coinbase (Owned, managed or offered) Ethereum Validator Full Nodes; and</li> <li>the Coinbase (Owned, managed or offered) Ethereum supporting Archive Nodes and Light Nodes; and</li> <li>the Coinbase (Ethereum compatible) wallets;</li> <li>Where both the Coinbase Nodes and the Coinbase Ethereum compatible end user wallet device are networked to have direct or indirect communication with each other.</li> </ul>		
	<ul> <li>Client Device</li> <li>The Second Client is the end user device that accepts an offer for an NFT. The First Client is the end user device that authorises an NFT to be offered for sale.</li> <li>Three (3) Client instances have been identified that represent various implementations that exist.</li> <li>1. Client is a device running an application that uses a Javascript Ethereum Provider API (as per EIP-1193) such as web3.js or ethers.js. This device will also use a key storage wallet with EIP-191 or EIP-712 signing support (such as Metamask/Coinbase Wallet) to send an NFT transactions. The use of an EIP-1193 based Ethereum Provider API or EIP-191/EIP-712 signing support means the Client is considered to be part of the Ethereum Network. http://eips/ethereum.org/EIPS/eip-191</li> <li>http://eips/ethereum.org/EIPS/eip-1193</li> </ul>		
	<ol> <li>Client representative of Eth Client Browser or Command Line Interface to create or send an NFT. The Client is considered to be part of the Ethereum Network.</li> <li>Client is a device running an application that uses a Javascript Ethereum Provider API (as per EIP-1193) such as web3.js or ethers.js. This device will also use a key storage wallet. This device uses private key management in order to sign transactions</li> </ol>		
	The Patent allows for, and the Claims do not prevent, the Computing System from being, or including, the Client. This is detailed in the Patent description [0055]. FIG. 1 (see Figure 16) depicts a typical embodiment for practicing the invention—especially for use with a distributed transfer mechanism—where the clients, transfer mechanism, facilitator, and data source are distinct participants. However, the depicted arrangement is not the only one		

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 71 of 117 PageID #: 133

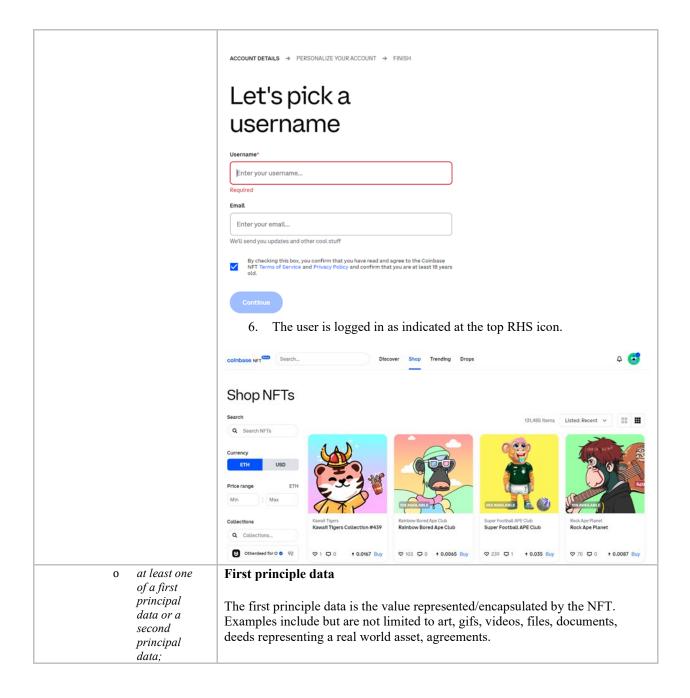
	<ul> <li>contemplated by the invention. In an alternate embodiment, the facilitator provides some or all aspects of the transfer mechanism. In another embodiment, the facilitator comprises some or all aspects of a client. For example, part or all of a client's data store, the ability to initiate or accept offers, etc., could be "embedded" in the facilitator, thereby enabling the facilitator to operate as a client itself (e.g., one controlled by the owners of the facilitator, or on behalf of a third party who has entrusted control to the facilitator). In yet another embodiment, the facilitator comprises the data source. Many configurations are contemplated by the invention are possible, and will become apparent to one skilled in the art.</li> <li>In addition, paragraph 0055 reads as follows:</li> <li>It will become apparent to one skilled in the art that aspects of each of embodiments above may be commingled. For example, the first client could find and retrieve it. As mentioned above, aspects of one or both of the first client and the second client could coincide with the facilitator allowing many of the above steps to be omitted as redundant where the facilitator is entrusted to act as a proxy for or on behalf of one of the first party and the second party. The facilitator could contain aspects of one of the clients, but not the other, in which case the extra-facilitator client would optionally independently validate transaction records it received from the facilitator before signing them, etc. In such embodiments, the facilitator typically comprises a means to control</li> </ul>
	transaction records it received from the facilitator before signing them, etc. In such embodiments, the facilitator typically comprises a means to control aspects of a client it comprises via an interface such as a web-based user
• a memory for storing a first asymmetric key pair, the first asymmetric key pair comprising a first private key and a first public key;	<ul> <li>appends of a criteria in comprises via an interface such as a web-obset aper- interface (UI), an application programmer's interface (API), etc.</li> <li>The Client device hosting the API software/libraries (used by an App), as part of the Computing Device, need a computer hardware/software combination to run, namely: <ul> <li>Memory (RAM), used in the computing device (such as a computer or mobile phone).</li> <li>Transaction record sector (stores transactions that haven't been submitted to the blockchain yet) kept via the crypto software wallets</li> </ul> </li> <li>Where the Client running the App and Wallet software, contains: <ul> <li>a first key pair sector which is generated and stored in the wallet software</li> <li>The asymmetric key pair generated and/or stored consists of a first private key and a first public key – all found and manipulated via the wallet software.</li> <li>The wallet software connects via the public key or the key pair, and authorizes (signs) the transaction with the private key of the key pair.</li> </ul> </li> <li>Where the Client interacts with the Eth Client Browser or Command Line Interface to transaction creation, contains: <ul> <li>a first key pair sector which is generated and stored on the device</li> <li>The asymmetric key pair stored consists of a first private key and a first public key.</li> </ul> </li> </ul>
	• a first key pair sector which is stored in the key management vault/software

• a network interface for	<ul> <li>private key and a key management</li> <li>The key managem the key pair, and a key of the key pair</li> </ul>	nent vault/software connect nuthorizes (signs) the transa r.	and manipulated via the s via the public key or action with the private	
• a network interface for receiving terms, the terms comprising:	<ul><li>The Client device requires a network interface in order to connect a wallet to the marketplace to offer the NFT for sale.</li><li>1. Click on the 'Sign In' button top RHS.</li></ul>			
	Coinbase NFT Search	Discover Shop Trending Drops	Sign in	
	Shop NFTs	Connect wallet × Use any existing crypto wallet to connect to Coinbase NFT		
	Search Q Search NFTs	Colnbase Wallet >	131,466 items Listed: Recent 🗸 🏢 🏢	
	Currency	💓 MetaMask >		
	Price range ETH Min : Max	WalletConnect >		
	Collections Baby Bored Ape Q. Collections	Create a new Colnbase Wallet	ub Sister Bored Ape Club	
	<ul> <li>C otherdeed for O 2 0 67 0 17 +0.079 Buy 0 0 0 +0.0725 Buy 0 0 0 +0.0079 Buy 0 37 0 5 +0.035 Buy</li> <li>2. Selecting 'Coinbase Wallet' prompts the user to connect to the website.</li> </ul>			
	coinbase NFT Search Disco	ver Shop Trending Drops		
	Shop NFTs	×	Infl.coinbete.com	
	Search Q. Search NFTs	Confirm Wallet Please sign into Colnbase Wallet to	Listed: Recent V Listed: Recent V Listed: Recent V Listed: Recent V	
	Currency ETH USD Price range ETH Content not (isofing	connect to Colhbase NFT	Funds will not leave your wallet until you execute a transaction	
	Min : Max	Content not loading Content not loading	Cannet Tot to	
	Collections Rainbow Bored Ape Club Q Collections	SuperNabob Baby Bored Ape Super Nabob Club Baby Bored Ape	RichApe Club	
	3. Attempting to con	nect without a network inte	erface (as shown with no	
	Wifi connected).	neer whilout a network life		

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 73 of 117 PageID #: 135

Shop NFTs Sere Sere Seren Curreny Trice range Child Callection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collection Collecti	Contern Wallet Presse sign into Concesse Wallet Convect to Concesse Wall Convect to Concesse Wall Content not loading Content not load	Image: state stat
Search     Price range     ETH     User     Price range     ETH     Max     Collections     Collectio	Discour     Shop     Trending     Drops         Connect wallet     ×       Use any wiskling to proper wallet to connect to contense NFT       Image: Contense Wallet	Sister Bored Ape Club
	×	
Currency  Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Currency Curre	Contern net loading Contern net loading Contern net loading Contern net loading Contern net loading Contern net loading Bigen flabio Super flabio	Context foot in Context foot in Context foot in Exclude Rich Ape Club

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 74 of 117 PageID #: 136



Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 75 of 117 PageID #: 137

	Coinbase NFT Search	Discover Shop Trending	Drops		۵ 😅
				Collection	
		K P	Kawali Tigers		
	Current price		Created by		
	0.0167 ETH	\$28.65	0x9125b91d	2	
		Buy now	Attributes		
		lake an offer	Background Gradlent 2	Body Light Red	
	Listing e	nds Sep 17 at 11:07 AM	Face Smile	Hat Viking	
	https://nft.coinbase.com	/nft/ethereum/0x0a6	1tems 52be5f3552c4	outfit 4252f50edcd	75de3aed
	<u>831f9599/439</u>				<u>, e ace aca</u>
	Second principle data				
	The seller will define a the NFT for.	ninimum or market	price that th	ey are willing	g to sell
	Coinbase NFT Search	Discover Shop Trending	Drops		ې چ
	1 4	)	Sell for free 0% Coinbase fees for a lin	nited time	
			Kawaii Tigers #439 Kawaii Tigers is a collectiv		
				to possibly become a Blue-	
			Collection		
		T T	Kawati Tigers		
	Current price		Created by		
	0.0167 ETH	\$28.65	0x9125b91d	8	
		Buy now	Attributes		
		lake an offer	Background Gradlent 2	Body Light Red	
	Listing e	nds Sep 17 at 11:07 AM	Face Smile	Hat Viking	
	https://nft.coinbase.com 831f9599/439	/nft/ethereum/0x0a6	52be5f3552c4	outfit 4252f50edcd	75de3aed
o a reference	First data source				
to at least one of a first					
data source	The NFT smart contract value.	which contains a m	etadata URI	link describe	es the
or a second	, aiuc.				

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 76 of 117 PageID #: 138

1.	
data source;	Metadata URI Link
and	/// @notice A distinct Uniform Resource Identifier
	(URI) for a given asset.
	/// @dev Throws if `_tokenId` is not a valid NFT.
	URIs are defined in RFC
	/// 3986. The URI may point to a JSON file that
	conforms to the "ERC721
	/// Metadata JSON Schema".
	function tokenURI (uint256 _tokenId) external view
	returns (string);
	https://eips.ethereum.org/EIPS/eip-721
	Second data source
	The NFT smart contract which contains a royalty payment fee as per ERC2981
	for consideration of the disbursement function.
	Royalty Info
	/// @notice Called with the sale price to determine
	how much royalty is owed and to /// whom.
	/// Oparam tokenId - the NFT asset queried for
	royalty information
	/// @param_salePrice - the sale price of the NFT
	asset specified by _tokenId
	/// @return receiver - address of who should be sent
	the royalty payment
	/// Creturn royaltyAmount - the royalty payment
	amount for salePrice
	function royaltyInfo(
	uint256 _tokenId,
	uint256 _salePrice
	) external view returns (
	address receiver,
	uint256 royaltyAmount
	https://eips.ethereum.org/EIPS/eip-2981

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 77 of 117 PageID #: 139

	Review purchase det	ails ×	
	Kawaii Tigers Co Kawaii Tigers	ollection #439	
	Pay with crypto wallet	REQUIRES GAS	
	0x56fe3F09 0.5001 ETH   0 wETH	Change	
	List price	<b>0.0167 ETH</b> \$28.65	
	Gas estimate	Shown in wallet	
	Creator collects 10% of all sales.		
	Open wallet to pay $ ightarrow$		
	https://nft.coinbase.com/nft/ethereu 831f9599/439	1m/0x0a62be5f35520	:4252f50edcd75de3aed
	Indicates that 10% of sale goes to t	he Creator.	
o an expiration timestamp;	The approval by the NFT owner to NFT is implied to be infinite or und deadline is mandated.		
	The approve function as defined by	y the EIP-721 is as fo	ollows.
	/// @notice Change or reaf	firm the approv	ed address for an
	NFT /// @dev The zero address address. Throws unless `ms owner, or an authorized op	g.sender` is th	e current NFT
	/// @param _approved The n /// @param tokenId The NE	new approved NFT	
	<pre>function approve(address _ external payable;</pre>		56 _tokenId)

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 78 of 117 PageID #: 140

coinbase NFT Search	Discover Shop Trending	Drops		۵ 😴
14	) 	Sell for free 0% Coinbase fees for a lim	ited time	
		Kawaii Tigers #439	Collection	
2		Kawaii Tigers is a collectio playing on the Ethereum b project with the end goal t Chip NFT project. Bringing Metaverse.	lockchain. A community led o possibly become a Blue-	
		Collection		
	V	Kawati Tigers		
Y I I.		Created by		
Current price	\$28.65	0x9125b91d	2	
Buy	now	Attributes		
Make :	un offer	Background Gradient 2	Body Light Red	
Listing ends Se	ep 17 at 11:07 AM	Face Smile	Hat Viking	
https://nft.coinbase.com/nf	t/ethereum/0x0a	62be5f3552c4	outfit 1252f50edcd	75de3aed
<u>831f9599/439</u>				
Shows an expiration with	the Listing endin	g Sep 17 at 11	:07am.	
Note that the expiration per in this case can be a subset approved address back to the for the terms to define a per a time or upon an event as the reception of the transact	quent all to the ap he NFT owner. To ont in time 'on o defined by the to	pprove function The description or after the exp erms'. In th	on to change n of the pate piration times	the ent allows stamp or at
Patent references: [0123] 22. On or after the as defined by the terms, an transaction record,	1	1	1	
[0186] 21. On or after the as defined by the terms, an transaction record,				
Also note that there is a dr to a specified deadline exp function does have a time	iration. This den	nonstrates that	the 'approv	
https://eips.ethereum.org/H	EIPS/eip-4494			
/// @notice Function /// @param spender d /// @param tokenId d spender on /// @param deadline	the address t the index of a timestamp	o approve the NFT to expiry for	approve the perm	the
/// @param sig a tra	aditional or	EIP-2098 s	ignature	

		<pre>function permit(address spender, uint256 tokenId, uint25 deadline, bytes memory sig) external; The hardware device heating the ADL software(liberrise (used hu or Arr), use </pre>	
•	a computer processor coupled to the memory and the network interface, the computer processor configured to:	The hardware device, hosting the API software/libraries (used by an App as a computer or mobile phone.	
i.	read the first private key from the memory;	The hardware device, hosting the API software/libraries (used by an App), suc as a computer or mobile phone. The App requires the user to authenticate themselves via the Wallet private key either before using the App or for sign individual transactions. The private key is encrypted on the device and accessed for reading (and subsequently used to calculate a transaction signature) by the app/wallet from the passcode/protection mechanism of the app/wallet.	
		🗧 🗧 Coinbase Wallet	
		Confirm transaction	
		coinbase NFT Request from nft.coinbase.com	
		Wallet used         0xA3B0012           Cost         \$12191.27         3.7 ETH	
		Network fee <b>0</b> \$187.73 - \$251.63 🌣	
		Total cost \$12379 - 12442.9	
		Network Ethereum	
		Network fees are high right now. Not paid to Coinbase Learn more	
		Cancel Confirm	
		Coinbase Wallet requesting confirmation to access the private key to authorise and sign the transaction.	
ii.	compute a first cryptographic signature from the first private key;	The hardware device, hosting the API software/libraries (used by an App), such as a computer or mobile phone. The App requires the user to authenticate themselves via a Wallet before using the App.	

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 80 of 117 PageID #: 142

	🗧 🗧 Coinbase Wallet
	Confirm transaction
	coinbase NFT Request from nft.coinbase.com
	Wallet used 0xA3B0012
	Cost \$12191.27 <b>3.7ETH</b>
	Network fee  \$187.73 - \$251.63  \$
	Total cost \$12379-12442.9
	Network Ethereum
	Network fees are high right now. Not paid to Coinbase Learn more
	Cancel Confirm
	Coinbase Wallet requesting confirmation to access the private key to authorise, calculate and sign the transaction.
	eth_SignTransaction
	web3.eth.signTransaction(transactionObject, address [, callback]) Signs a transaction.
	Parameters Object - The transaction data to sign. See web3.eth.sendTransaction() for more.
	String - Address to sign transaction with. Function - (optional) Optional callback, returns an error object as first parameter and the result as second. https://web3js.readthedocs.io/en/v1.5.2/web3-eth.html#signtransaction
iii. create an inchoate data record comprising:	While the Computing Device and the First Client device are the same device, the Inchoate Transaction is created for the NFT seller to approve an account (typically a smart contract account) to transfer ownership of the NFT when the terms are met.
	The App (First Client as part of the Computing Device) creates a transaction message to be sent to the Ethereum Network. RPC JSON API calls are defined to allow interaction with the Ethereum network. The App constructs the following message which is aligned with this standard.
	eth_sendTransaction (EIP1559) Creates new message call transaction or a contract creation, if the data field contains code.

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 81 of 117 PageID #: 143

<ul> <li>Object - The transaction object <ol> <li>from - String Number: The address for the sending account. Uses the web3.eth.defaultAccount property, if not specified. Or an address or index of a local wallet in web3.eth.accounts.wallet.</li> <li>to - String: (optional) The destination address of the message, left undefined for a contract-creation transaction.</li> <li>value - Number String BN BigNumber: (optional) The value transferred for the transaction in wei, also the endowment if it's a contract-creation transaction.</li> <li>gas - Number: (optional, default: To-Be-Determined) The amount of gas to use for the transaction (unused gas is refunded).</li> <li>gasPrice - Number String BN BigNumber: (optional) The price of gas for this transaction in wei, defaults to web3.eth.gasPrice.</li> <li>type - Number String BN BigNumber: (optional) A positive unsigned 8-bit number between 0 and 0x7f that represents the type of the transaction.</li> <li>maxFeePerGas - Number String BN: (optional, defaulted to (2 * block.baseFeePerGas) + maxPriorityFeePerGas) The maximum fee per gas that the transaction is willing to pay in total</li> <li>maxPriorityFeePerGas - Number String BN (optional, defaulted to 1 Gwei) The maximum fee per gas to give miners to incentivize them to include the transaction plans to access</li> <li>data - String: (optional) Either a ABI byte string containing the data of the function call on a contract, or in the case of a contract-creation transaction transaction the initialisation code.</li> </ol> </li> <li>nonce - Number: (optional) Integer of the nonce. This allows to overwrite your own pending transactions that use the same nonce.</li> <li>chain - String: (optional) Defaults to london.</li> </ul>
Returns 1. DATA, 32 Bytes - the transaction hash, or the zero hash if the transaction is not yet available.
https://web3js.readthedocs.io/en/v1.5.2/web3-eth.html#sendtransaction
The App (First Client as part of the Computing Device) initiates a smart contract call populating the data field of the above message.
From: https://eips.ethereum.org/EIPS/eip-721
<pre>/// @notice Change or reaffirm the approved address for an NFT /// @dev The zero address indicates there is no approved address. Throws unless `msg.sender` is the current NFT owner, or an authorized operator of the current owner. /// @param _approved The new approved NFT controller /// @param _tokenId The NFT to approve function approve(address _approved, uint256 _tokenId) external payable;</pre>

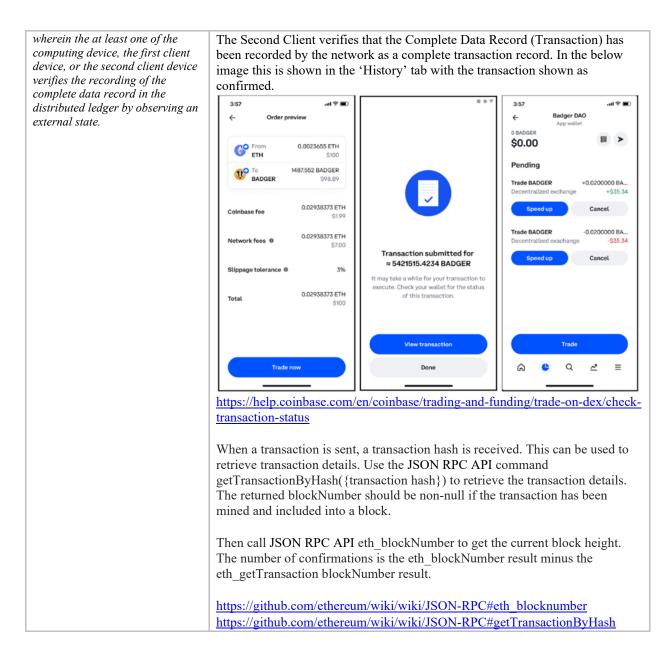
# Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 82 of 117 PageID #: 144

<ul> <li>a commit input for receiving a commit data from a commit transportion;</li> </ul>	The commit input is the offer to sell the NFT indicated by an 'approve' function call.	
transaction;	From: https://eips.ethereum.org/EIPS/eip-721	
	/// @notice Change or reaffirm the approved address for a NFT /// @dev The zero address indicates there is no approved	
	address. Throws unless `msg.sender` is the current NFT owner, or an authorized operator of the current owner. /// @param _approved The new approved NFT controller /// @param _tokenId The NFT to approve	
	<pre>function approve(address _approved, uint256 _tokenId) external payable;</pre>	
• one or more output data obtained from at least one of the first principal data or the	<ul> <li>The output data includes:</li> <li>The signed acceptance of the offered NFT based on the value description of the Metadata URI link.</li> </ul>	
second principal data, and a value data from at least one of the first data source or the second data source; and	• The disbursement amount to the seller and the originator based on the royalty fee definition of the NFT.	
• <i>the first cryptographic signature; and</i>	The hardware device, hosting the API software/libraries (used by an App), so as a computer or mobile phone. The App requires the user to authenticate themselves via a Wallet before using the App.	
	e e Coinbase Wallet	
	Confirm transaction	
	Coinbase NFT Request from nft.coinbase.com	
	Wallet used 0xA3B0012	
	Cost \$12191.27 <b>3.7 ETH</b>	
	Network fee • \$187.73 - \$251.63 \$	
	Total cost \$12379-12442.9	
	Network Ethereum	
	Network fees are high right now. Not paid to Coinbase Learn more	
	Cancel Confirm	
	Coinbase Wallet requesting confirmation to access the private key to authoris calculate and sign the transaction.	

	eth_SignTransaction				
	web3.eth.signTransaction(transactionObject, address [, callback]) Signs a transaction.				
	Parameters Object - The transaction String - Address to sign Function - (optional) Op parameter and the result https://web3js.readthedo	transaction with. tional callback, retu as second.	irns an error	object as fi	rst
<i>iv.</i> publish the inchoate data record to at least one of the first client device or the second client device,	The published Inchoate marketplace. The follow Marketplace.				a 4 💽
		Ň+	Sell for free 0% Coinbase fees for a lim	nited time	
			Kawaii Tigers #439 Kawaii Tigers is a collectio playing on the Ethereum to project with the end goal Chip NFT project. Bringing Metaverse.	on of 3,333 Kawali Tigers slockchain. A community le to possibly become a Blue	
			Collection		
	Current price		Created by		
	0.0167 ETH	\$28.65 Buy now	0x9125b91d	2	*
		lake an offer	Attributes Background Gradient 2	Body Light Red	
		nds Sep 17 at 11:07 AM	Face Smile	Hat Viking	
	1		Items	Outfit	1751 2 1
	https://nft.coinbase.com/ 831f9599/439	/mt/ethereum/0x0a	<u>520e515552C4</u>	<u>+252150eac</u>	<u>a/sdesaed</u>
wherein the decentralized digital currency comprises a distributed ledger that enables processing the transaction between the first client device and the second client device without the need for a trusted central authority,	Ether is used as the dece distributed ledger without From the Ethereum yello 2.1. Value. In order to in to be an agreed method has an intrinsic currency by the Old English <sup>-</sup> D.	ut the need for a tru ow paper. acentivise computat for transmitting val v, Ether, known also	sted central a ion within the ue. To addre. o as ETH and	uthority. e network, i ss this issue	here needs e, Ethereum
	https://ethereum.github.i	io/yellowpaper/pap	er.pdf		
wherein the inchoate data record is used by at least one of the first client device or the second client device to create a complete data	The NFT approved for sale is accepted by the buyer with the safeTransfer function call to transfer ownership to the buyer.				

record and to create the	The App (Second Client as part of the Facilitator) creates a transaction message
transaction by broadcasting the complete data record for transmitting and receiving among network participants in the	to be sent to the Ethereum Network. RPC JSON API calls are defined to allow interaction with the Ethereum network. The App constructs the following message which is aligned with this standard.
computer network for recording in	eth sendTransaction (EIP1559)
the distributed ledger,	Creates new message call transaction or a contract creation, if the data field
	contains code.
	Object - The transaction object
	1. from - String Number: The address for the sending account. Uses the
	web3.eth.defaultAccount property, if not specified. Or an address or index of a local wallet in web3.eth.accounts.wallet.
	2. to - String: (optional) The destination address of the message, left
	undefined for a contract-creation transaction.
	3. value - Number String BN BigNumber: (optional) The value
	transferred for the transaction in wei, also the endowment if it's a contract-creation transaction.
	4. gas - Number: (optional, default: To-Be-Determined) The amount of
	gas to use for the transaction (unused gas is refunded).
	5. gasPrice - Number String BN BigNumber: (optional) The price of
	gas for this transaction in wei, defaults to web3.eth.gasPrice.
	6. type - Number String BN BigNumber: (optional) A positive unsigned 8-bit number between 0 and 0x7f that represents the type of the
	transaction.
	7. maxFeePerGas - Number String BN: (optional, defaulted to (2 *
	block.baseFeePerGas) + maxPriorityFeePerGas) The maximum fee
	per gas that the transaction is willing to pay in total 8. maxPriorityFeePerGas - Number String BN (optional, defaulted to 1
	8. maxPriorityFeePerGas - Number String BN (optional, defaulted to 1 Gwei) The maximum fee per gas to give miners to incentivize them
	to include the transaction (Priority fee)
	9. accessList - List of hexstrings (optional) a list of addresses and
	storage keys that the transaction plans to access
	10. data - String: (optional) Either a ABI byte string containing the data of the function call on a contract, or in the case of a contract-creation
	transaction the initialisation code.
	11. nonce - Number: (optional) Integer of the nonce. This allows to
	overwrite your own pending transactions that use the same nonce.
	12. chain - String: (optional) Defaults to mainnet.
	13. hardfork - String: (optional) Defaults to london.
	Returns
	2. DATA, 32 Bytes - the transaction hash, or the zero hash if the
	transaction is not yet available.
	https://web3js.readthedocs.io/en/v1.5.2/web3-eth.html#sendtransaction
	The App (Second Client as part of the Facilitator) initiates a smart contract call
	populating the data field of the above message.
	From: <u>https://eips.ethereum.org/EIPS/eip-721</u>
	rion. <u>https://etps.culeteuin.org/Dir/3/etp-/21</u>
L	1

	ERC-721 standardizes a safe transfer function safeTransferFrom (overloaded with and without a bytes parameter) and an unsafe function
	<ul> <li>transferFrom. Transfers may be initiated by:</li> <li>The owner of an NFT</li> <li>The approved address of an NFT</li> <li>An authorized operator of the current owner of an NFT</li> </ul>
wherein at least one of the first client device or the second client device signs the inchoate data	The Second Client requires the buyer to authenticate themselves via a Wallet before using the App. The wallet private key is used to sign the transaction for the EVM smart contract call.
record and saves a copy of the	Coinbase Wallet
inchoate data record on at least one of the first client device or the second client device; and	Confirm transaction
	Coinbase NFT Request from
	nft.coinbase.com
	Wallet used 0xA3B0012
	Cost \$12191.27 3.7 ETH
	Network fee 🛛 \$187.73 - \$251.63 🕏
	Total cost \$12379 - 12442.9
	Network Ethereum
	Network fees are high right now. Not paid to Coinbase Learn more
	Cancel Confirm
	<ul> <li>Coinbase Wallet requesting confirmation to access the private key to authorise and sign the transaction.</li> <li>Note that it is not mandatory for the inchoate data record to be saved by the device as it is optional as described in the Patent general description.</li> <li>[0095] 11. The second client creates the complete commit transaction record by similar the inchoate commit transaction record and optionally saves a commit transaction.</li> </ul>
	by signing the inchoate commit transaction record and <u>optionally saves a cop</u> <u>in non-transitory memory</u> , the complete commit transaction record comprising [0111] 15. Optionally, the first client saves a copy of the complete commit transaction record in non-transitory memory.



# Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 87 of 117 PageID #: 149

Claim Chart U.S.	Patent No. 11,196,566 (the "'566 Patent") Coinbase
<u>Claim 2</u>	<u>Coinbase Products &amp; Services</u> The transfer of a NFT (on the Ethereum network) from one party to another.
The device of claim 1, where: the computer processor is configured to obtain the one or more output data based on:	<ul> <li>The Computing Device   Facilitator consists of:</li> <li>the Coinbase (Owned, managed or offered) Ethereum Validator Full Nodes; and</li> <li>the Coinbase (Owned, managed or offered) Ethereum supporting Archive Nodes and Light Nodes; and</li> <li>the Coinbase (Ethereum compatible) wallets;</li> <li>Where both the Coinbase Nodes and the Coinbase Ethereum compatible end user wallet device are networked to have direct or indirect communication with each other.</li> </ul>
the first principal data; and the value data from the first data source.	The Validator node, as part of processing the received complete transaction, disburses the value by sending the NFT (with the metadata link) to the recipient. The Validator node, as part of processing the received complete transaction, disburses the royalty fee to the NFT originator address.
	Metadata URI Link ///@notice A distinct Uniform Resource Identifier (URI) for a given asset. ///@dev Throws if `_tokenId` is not a valid NFT. URIs are defined in RFC /// 3986. The URI may point to a JSON file that conforms to the "ERC721 /// Metadata JSON Schema". function tokenURI(uint256 _tokenId) external view returns (string); https://eips.ethereum.org/EIPS/eip-721
	Royalty Info         /// @notice Called with the sale price to determine how much royalty is owed and to /// whom.         /// @param _tokenId - the NFT asset queried for royalty information         /// @param _salePrice - the sale price of the NFT asset specified by tokenId         /// @return receiver - address of who should be sent the royalty payment         /// @return royaltyAmount - the royalty payment amount for _salePrice         function royaltyInfo(         uint256 _tokenId,         uint256 for oyaltyAmount         ) external view returns (         address receiver,         uint256 royaltyAmount
	); https://eips.ethereum.org/EIPS/eip-2981

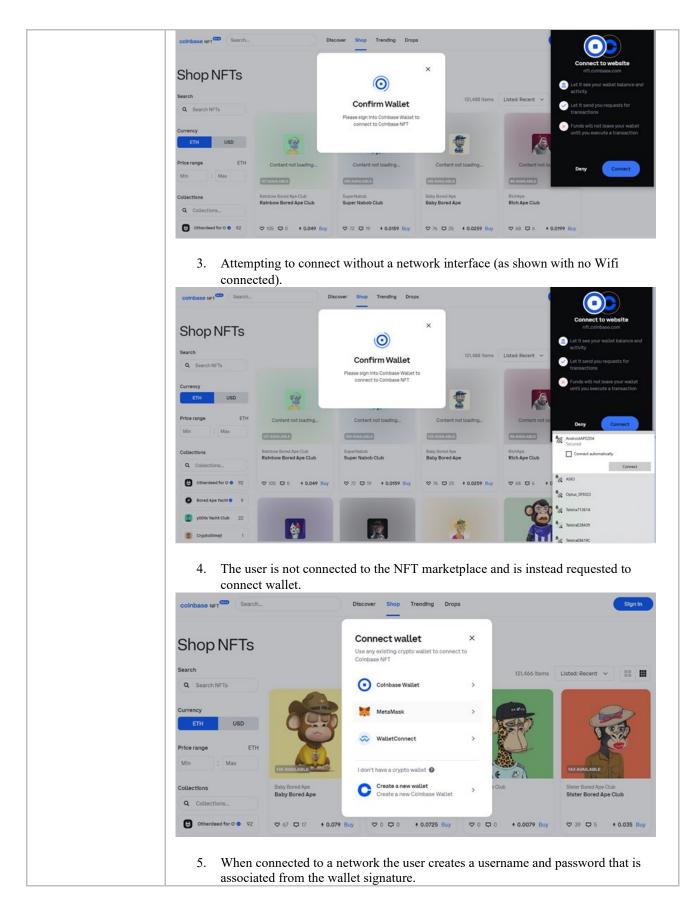
# Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 88 of 117 PageID #: 150

Claim	Chart U.S. Patent No. 11,196,566 (the "'566 Patent") Coinbase
<u><b>Claim 7</b></u> A system for processing a transaction between a first client device and a second client device via a transfer mechanism the system comprising a computing device, the first client device, the second client device, and the transfer mechanism.	<u>Coinbase Products &amp; Services</u> The transfer of a NFT (on the Ethereum network) from one party to another.
7. a. the computing device comprising: i. a first memory comprising for storing a first asymmetric key pair, the first asymmetric key pair comprising a first private key and a first public key;	<ul> <li>The Computing Device   Facilitator consists of: <ul> <li>the Coinbase (Owned, managed or offered) Ethereum Validator Full Nodes; and</li> <li>the Coinbase (Owned, managed or offered) Ethereum supporting Archive Nodes and Light Nodes; and</li> <li>the Coinbase (Ethereum compatible) wallets;</li> </ul> </li> <li>Where both the Coinbase Nodes and the Coinbase Ethereum compatible end user wallet device are networked to have direct or indirect communication with each other.</li> <li>Client Device</li> <li>The First Client is the end user device that accepts an offer for an NFT.</li> <li>The Second Client is the end user device that authorises an NFT to be offered for sale.</li> <li>Three (3) Client instances have been identified that represent various implementations that exist.</li> <li>Client is a device running an App (Coinbase Wallet) that uses a Javascript Ethereum Provider API (as per EIP-1193) such as web3.js or ethers.js. This device will also use a key storage wallet with EIP-191 or EIP-712 signing support (such as Metamask/Coinbase Wallet) to send an NFT transactions. The use of an EIP-1193 based Ethereum Provider API or EIP-191/EIP-712 signing support means the Client is considered to be part of the Ethereum Network. http://eips/ethereum.org/EIPS/eip-1193</li> <li>The Computing Device and the Client are the same device where the Client is a Client Browser or Command Line Interface on a Coinbase Ethereum Node used to create, sign and submit NFT transactions to the Coinbase Ethereum Node for processing.</li> </ul> <li>Client is a device running an App (Coinbase Wallet) that uses a Javascript Ethereum Node reate, sign and submit NFT transactions to the Coinbase Ethereum Node used to create, sign and submit NFT transactions to the Coinbase Ethereum Node for processing.</li>

# Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 89 of 117 PageID #: 151

	<ul> <li>Where the Client running the App (Coinbase Wallet) software, as part of the Facilitator, need a computer hardware/software combination to run, namely: <ul> <li>Memory (RAM), used in the computing device (such as a computer or mobile phone).</li> <li>Transaction record sector (stores transactions that haven't been submitted to the blockchain yet) kept via the crypto software wallets</li> </ul></li></ul>
	<ul> <li>Where the Client running the App (Coinbase Wallet) software, contains: <ul> <li>a first key pair sector which is generated and stored in the wallet software</li> <li>The asymmetric key pair generated and/or stored consists of a first private key and a first public key – all found and manipulated via the wallet software.</li> <li>The wallet software connects via the public key or the key pair, and authorizes (signs) the transaction with the private key of the key pair.</li> </ul></li></ul>
	<ul> <li>Where the Client interacts with the Eth Client Browser or Command Line Interface to transaction creation, contains:</li> <li>a first key pair sector which is generated and stored on the device</li> <li>The asymmetric key pair stored consists of a first private key and a first public key.</li> </ul>
	<ul> <li>Where the Client interacts with App (Coinbase Wallet) with Private key management in order to sign transactions, contains: <ul> <li>a first key pair sector which is stored in the key management vault/software</li> <li>The asymmetric key pair generated and/or stored consists of a first private key and a first public key – all found and manipulated via the key management vault/software.</li> <li>The key management vault/software connects via the public key or the key pair, and authorizes (signs) the transaction with the private key of the key pair.</li> </ul> </li> </ul>
ii. a first network interface for	The Client device requires a network interface in order to connect a wallet to the marketplace to offer the NFT for sale. 1. Click on the 'Sign In' button top RHS.
receiving terms, the terms comprising:	Coinbase with Beach       Discover       Shop Trending Drops       Explore         Shop NFTS       Variable Connect to Coinbase Wallet       Stated Recent V III       Isted Recent V III         Search NFTs       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect         Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect         Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect         Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Connect       Valuet Conn

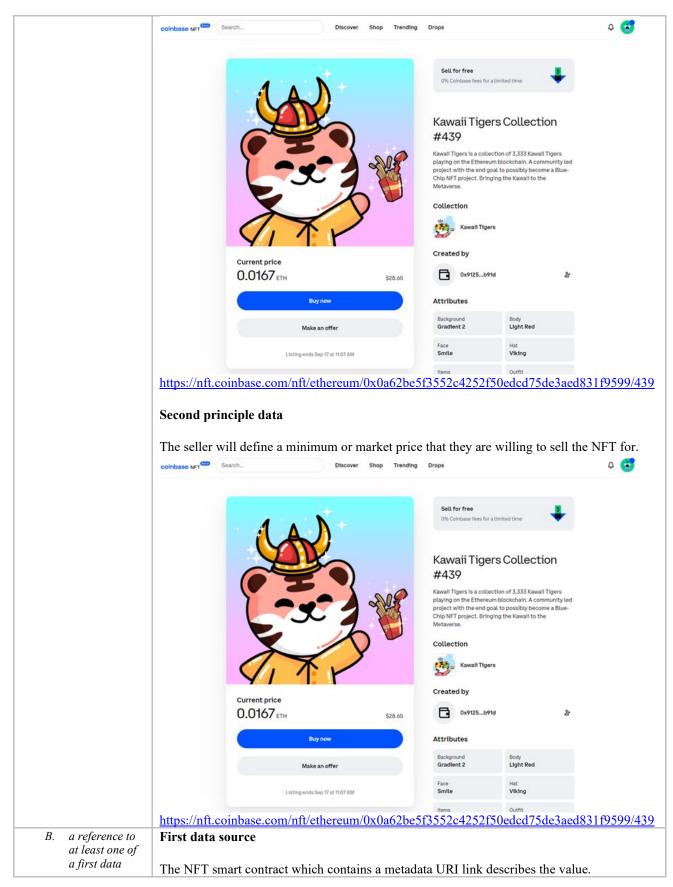
Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 90 of 117 PageID #: 152



Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 91 of 117 PageID #: 153

	coinbase Nr1 <sup>™</sup> Search Discover Shop Trending Drops 
	Shop NFTs
	Search Q. Search NFTs Confirm Wallet Ustadi Recent VI (Add Items Listed: Recent VI (Add VI) (B app or 10- VI) (Add VI) (B app or 10- VI) (Add VI) (B app or 10- VI) (Add VI) (
	Please sign into Corlease Wallet to connect to Colinbase NFT
	ETH USD I Contact out loading Contact and loading Contact and loading Contact and loading
	Price range EIH Content not loading Conten
	Collections Plankow Bored Age Club Super Nabob Club Bady Bored Age Club Club Rathbow Bored Age Club Club Rathbow Bored Age Club
	Contractised for 0 @ 92 \$ 105 \$ 0 + 0.049 Buy \$ 72 \$ 19 + 0.0559 Buy \$ 76 \$ 25 + 0.0259 Buy \$ 0.68 \$ 6 + 0.0599 Buy
	ACCOUNT DETAILS $\rightarrow$ PERSONALIZE YOUR ACCOUNT $\rightarrow$ FINISH
	Let's pick a
	username
	Username*
	Enter your username
	Required
	Enter your email
	We'll send you updates and other cool stuff
	By checking this box, you confirm that you have read and agree to the Coinbase NFT Terms of Service and Privacy Policy and confirm that you are at least 18 years old.
	Continue
	6. The user is logged in as indicated at the top RHS icon.
	Coinbase Net Search Discover Shop Trending Drops A 🖸
	ShanNETa
	Shop NFTs
	Search NFTs
	Collections Kawaii Tigers Collection #439 Rainbow Bored Ape Club Super Football APE Club Rock Ape Planet Q. Collections
	Ctherdeed for O @ 92 O 1 D 0 + 0.0167 Buy O 103 D 0 + 0.0065 Buy O 230 D 1 + 0.035 Buy O 70 D 0 + 0.0087 Buy
A. at least one of a first	First principle data
principal data or a second	The first principle data is the value represented/encapsulated by the NFT. Examples include
principal data;	but are not limited to art, gifs, videos, files, documents, deeds representing a real world asset, agreements.

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 92 of 117 PageID #: 154



Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 93 of 117 PageID #: 155

source or a	
second data	Matadata UDLL intr
source; and	Metadata URI Link /// @notice A distinct Uniform Resource Identifier (URI) for a
	given asset.
	/// @dev Throws if ` tokenId` is not a valid NFT. URIs are
	defined in RFC
	/// 3986. The URI may point to a JSON file that conforms to
	the "ERC721
	/// Metadata JSON Schema".
	function tokenURI(uint256 tokenId) external view returns
	(string);
	https://eips.ethereum.org/EIPS/eip-721
	Second data source
	The NFT smart contract which contains a royalty payment fee as per ERC2981 for
	consideration of the disbursement function.
	Royalty Info
	/// @notice Called with the sale price to determine how much
	royalty is owed and to /// whom.
	/// @param _tokenId - the NFT asset queried for royalty information
	/// @param salePrice - the sale price of the NFT asset
	specified by tokenId
	/// @return receiver - address of who should be sent the
	royalty payment
	/// @return royaltyAmount - the royalty payment amount for
	salePrice
	function royaltyInfo(
	<pre>uint256 _tokenId,</pre>
	<pre>uint256 _salePrice</pre>
	) external view returns (
	address receiver,
	uint256 royaltyAmount
	https://eips.ethereum.org/EIPS/eip-2981

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 94 of 117 PageID #: 156

	Review purchase det	tails ×	
	Kawaii Tigers Co Kawaii Tigers	ollection #439	
	Pay with crypto wallet	REQUIRES GAS	
	0x56fe3F09 0.5001 ETH   0 wETH	Change	
	List price	<b>0.0167 ETH</b> \$28.65	
	Gas estimate	Shown in wallet	
	Creator collects 10% of all sales.  Open wallet to pay   https://nft.coinbase.com/nft/ethereum/	0x0a62be5f3552c4252f50	)edcd75de3aed831f9599/439
C. an expiration	Indicates that 10% of sale goes to the The approval by the NFT owner to sel	Creator. l or allow a transfer of ow	
timestamp,	implied to be infinite as no explicit dea The approve function as defined by the	-	
	<pre>/// @notice Change or reaffirm the approved address for an NFT /// @dev The zero address indicates there is no approved address. Throws unless `msg.sender` is the current NFT owner, or an authorized operator of the current owner. /// @param approved The new approved NFT controller</pre>		
	<pre>/// @param _approved The New /// @param _tokenId The NFT ; function approve(address _app payable;</pre>	to approve	

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 95 of 117 PageID #: 157

Coinbase NFT Search Discover Shop Trendin	ng Drops	۵ 😅		
	Sell for free Ofic Coinbase fees for a limited time			
	Kawaii Tigers Collection #439			
the set	Kawaii Tigers is a collection of 3,333 Kawaii Tigers playing on the Ethereum blockchain. A community led project with the end goal to possibly become a Blue- Chip NFT project. Bringing the Kawaii to the Metaverse.			
	Collection			
	Kawati Tigers			
VITE	20			
Current price	Created by			
0.0167 етн \$28.65	0x9125b91d &			
Buy now	Attributes			
Make an offer	Background Body Gradlent 2 Light Red			
Listing ends Sep 17 at 11:07 AM	Face Hat Smile Viking			
	Items Outfit			
https://nft.coinbase.com/nft/ethereum/0x0a62be5f3552c4252f50edcd75de3aed831f9599/439				
Note that the expiration period can be represent a subsequent all to the approve function to chan owner. The description of the patent allows for the expiration timestamp or at a time or upon ar instance the event is the reception of the transac	nge the approved address back t the terms to define a point in ti in event as defined by the terms.	to the NFT me 'on or afte		
Patent references: [0123] 22. On or after the expiration timestamp the terms, and before the lock time of the comple				
[0186] 21. On or after the expiration timestamp the terms, and before the lock time of the complete				
Also note that there is a draft EIP-4494 which s deadline expiration. This demonstrates that the component albeit undefined and implied.				
https://eips.ethereum.org/EIPS/eip-4494				
/// @notice Function to approve by /// @param spender the address to a /// @param tokenId the index of the /// @param deadline a timestamp exp /// @param sig a traditional or EIP	pprove NFT to approve the spe iry for the permit -2098 signature	nder on		
<pre>function permit(address spender, bytes memory sig) external;</pre>	uint256 tokenId, uint25	6 deadline		

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 96 of 117 PageID #: 158

coupled to the	
first memory and the first	
network	
interface, the	
first computer	
processor	
configured to:	
<i>A. read the first</i>	The hardware device, hosting the API software/libraries (used by an App), such as a
private key from the first memory	computer or mobile phone. The App requires the user to authenticate themselves via the Wallet private key either before using the App or for sign individual transactions. The private key is encrypted on the device and accessed for reading (and subsequently used to calculate a transaction signature) by the app/wallet from the passcode/protection mechanism of the app/wallet.
	Combase Hanet
	Confirm transaction
	COINDASE NFT
	Request from
	nft.coinbase.com
	Wallet used 0xA3B0012
	Cost \$12191.27 3.7 ETH
	Network fee • \$187.73 - \$251.63 🗘
	Total cost \$12379 - 12442.9
	Network Ethereum
	Network fees are high right now. Not paid to Coinbase Learn more
	Cancel Confirm
	Coinbase Wallet requesting confirmation to access (read) the private key to authorise and sign the transaction.
B. compute a	The hardware device, hosting the API software/libraries (used by an App), such as a
first	computer or mobile phone. The App requires the user to authenticate themselves via a Wallet
cryptographic	before using the App.
signature	
from the first	
private key;	

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 97 of 117 PageID #: 159

EXHIBIT 3

	Coinba	ase Wallet	
	Confirm	transaction	
	coinbase NFT Request from nft.coinbase.co	m	
	Wallet used	0xA3B0012	
	Cost	\$12191.27 3.7 ETH	
	Network fee 0	\$187.73 - \$251.63 🌣	
	Total cost	\$12379 - 12442.9	
	Network	Ethereum	
	Network fees are high r Coinbase Learn more	right now. Not paid to X	
	Cancel	Confirm	
	calculate signature and si eth_SignTransaction		(read) the private key to authorise, ss [, callback])
	Parameters Object - The transaction String - Address to sign t Function - (optional) Opt result as second.	ransaction with.	.sendTransaction() for more. error object as first parameter and the tml#signtransaction
C. create an inchoate data record comprising:	While the Computing Inchoate Transaction	Device and the Second Cl is created for the NFT selle	ient device are the same device, the er to approve an account (typically a smart FT when the terms are met.
	Network. RPC JSON	API calls are defined to al	n message to be sent to the Ethereum low interaction with the Ethereum sage which is aligned with this standard.
	eth_sendTransaction Creates new message		ect creation, if the data field contains code.

# Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 98 of 117 PageID #: 160

	<ul> <li>Object - The transaction object <ol> <li>from - String Number: The address for the sending account. Uses the web3.eth.defaultAccount property, if not specified. Or an address or index of a local wallet in web3.eth.accounts.wallet.</li> <li>to - String: (optional) The destination address of the message, left undefined for a contract-creation transaction.</li> <li>value - Number String BN BigNumber: (optional) The value transferred for the transaction in wei, also the endowment if it's a contract-creation transaction.</li> <li>gas - Number: (optional, default: To-Be-Determined) The amount of gas to use for the transaction (unused gas is refunded).</li> <li>gasPrice - Number String BN BigNumber: (optional) The price of gas for this transaction in wei, defaults to web3.eth.gasPrice.</li> <li>type - Number String BN BigNumber: (optional) A positive unsigned 8-bit number between 0 and 0x7f that represents the type of the transaction.</li> <li>maxFeePerGas - Number String BN: (optional, defaulted to (2 * block.baseFeePerGas) + maxPriorityFeePerGas) The maximum fee per gas that the transaction is willing to pay in total</li> <li>maxPriorityFeePerGas - Number String BN (optional, defaulted to 1 Gwei) The maximum fee per gas to give miners to incentivize them to include the transaction plans to access</li> <li>data - String: (optional) Either a ABI byte string containing the data of the function call on a contract, or in the case of a contract-creation transaction the initialisation code.</li> <li>nonce - Number: (optional) Integer of the nonce. This allows to overwrite your own pending transactions that use the same nonce.</li> <li>chain - String: (optional) Defaults to london.</li> </ol> </li> </ul>
	<ul> <li>Returns</li> <li>3. DATA, 32 Bytes - the transaction hash, or the zero hash if the transaction is not yet available.</li> </ul>
	https://web3js.readthedocs.io/en/v1.5.2/web3-eth.html#sendtransaction
	The App (Coinbase Wallet) initiates a smart contract call populating the data field of the above message.
	From: https://eips.ethereum.org/EIPS/eip-721
	<pre>/// @notice Change or reaffirm the approved address for an NFT /// @dev The zero address indicates there is no approved address. Throws unless `msg.sender` is the current NFT owner, or an authorized operator of the current owner. /// @param _approved The new approved NFT controller /// @param _tokenId The NFT to approve function approve(address _approved, uint256 _tokenId) external payable;</pre>
	The commit input is the offer to sell the NFT indicated by an 'approve' function call.
input for receiving a commit data	From: <u>https://eips.ethereum.org/EIPS/eip-721</u>

# Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 99 of 117 PageID #: 161

	from a commit transaction;	<pre>/// @notice Change or reaffirm the approved address for an NFT /// @dev The zero address indicates there is no approved address. Throws unless `msg.sender` is the current NFT owner, or an authorized operator of the current owner. /// @param _approved The new approved NFT controller /// @param _tokenId The NFT to approve function approve(address _approved, uint256 _tokenId) external payable;</pre>
II.	one or more outputs obtained from at least one of the first principal data or the second principal data, and a value data from at least one of the first data source or the second data source; and;	<ul> <li>The output data includes:</li> <li>The signed acceptance of the offered NFT based on the value description of the Metadata URI link.</li> <li>The disbursement amount to the seller and the originator based on the royalty fee definition of the NFT.</li> </ul>
III.	The first cryptographic signature; and	The hardware device, hosting the API software/libraries (used by an App), such as a computer or mobile phone. The App requires the user to authenticate themselves via a Wallet before using the App.

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 100 of 117 PageID #: 162

	Coinbase Wallet	
	Confirm transaction	
	coinbase NFT Request from nft.coinbase.com	
	Wallet used 0xA3B0012	
	Cost \$12191.27 <b>3.7 ETH</b>	
	Network fee • \$187.73 - \$251.63 🌣	
	Total cost \$12379 - 12442.9	
	Network Ethereum	
	Network fees are high right now. Not paid to Coinbase Learn more	
	Cancel Confirm	
	Coinbase Wallet requesting confirmation to access the private key to a transaction.	uthorise and sign the
	eth_SignTransaction web3.eth.signTransaction(transactionObject, address [, callback])	
	Signs a transaction.	
	Parameters Object - The transaction data to sign. See web3.eth.sendTransaction() f String - Address to sign transaction with. Function - (optional) Optional callback, returns an error object as first p result as second. <u>https://web3js.readthedocs.io/en/v1.5.2/web3-eth.html#signtransaction</u>	parameter and the
D. publish the inchoate data record to at least one of the first client device or the second client device;;	The published Inchoate Data Record is the offered NFT for sale on a monotonic following is an example from the Coinbase NFT Marketplace.	arketplace. The

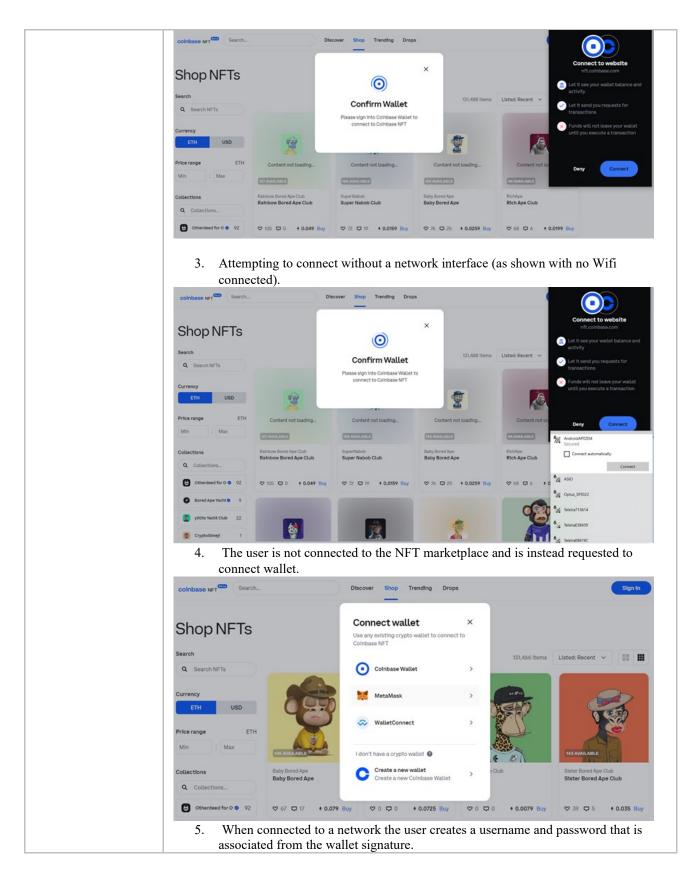
Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 101 of 117 PageID #: 163

	Coinbase NFT Search	Discover Shop Trendin	g Drops		۵ 🔁
	4	A . A +	Sell for free 0% Coinbase fees for a li	errited time	
			Kawaii Tigers #439	s Collection	
		2 2 2		blockchain. A community led to possibly become a Blue-	
			Collection		
		XY	Kawali Tigers		
	Current price		Created by		
	0.0167 ETH	\$28.65	0x9125b91d	2:	
		Buy now	Attributes		
		Make an offer	Background Gradient 2	Body Light Red	
		Listing ends Sep 17 at 11:07 AM	Face Smile	Hat Viking	
	https://nft.coinbase.com	/nft/ethereum/0x0a62be	Items 5f3552c4252f5	outfit 0edcd75de3aed831f9	599/
memory for storing a second asymmetric key pair, the second asymmetric key pair comprising a second private key and a second public key.	<ul> <li>Where both the Coinbast device are networked to</li> <li>Client Device</li> <li>The First Client is the end The Second Client is the end Three (3) Client instance exist.</li> <li>1. Client is a devine the Ethereum Provowill also use a final second to the end to the</li></ul>	Ethereum compatible) we be Nodes and the Coinba have direct or indirect of and user device that accept e end user device that accept e end user device that accept e end user device that accept cer running an App (Coi rider API (as per EIP-11 key storage wallet with	ase Ethereum co communication pts an offer for a athorises an NFT that represent va nbase Wallet) th 93) such as web EIP-191 or EIP	with each other. an NFT. If to be offered for sal arious implementation nat uses a Javascript o3.js or ethers.js. This -712 signing support	le. ns tha s devi (such
	<ul> <li>1193 based Eth Client is consid http://eips/eth http://eips/eth http://eips/eth</li> <li>2. The Computing Client Browser</li> </ul>	Coinbase Wallet) to send nereum Provider API or lered to be part of the E ereum.org/EIPS/eip-1 ereum.org/EIPS/eip-7 ereum.org/EIPS/eip-1 g Device and the Client or Command Line Inte d submit NFT transactio	EIP-191/EIP-71 thereum Networ 91 712 193 are the same de rface on a Coinl	12 signing support mo rk. vice where the Client base Ethereum Node	eans t t is a used

# Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 102 of 117 PageID #: 164

	Ethereum Provider AP	ing an App (Coinbase Wallet) that use PI (as per EIP-1193) such as web3.js o key storage wallet. This device uses pr ctions	r ethers.js This
	Where the Client running the A a computer hardware/software of	pp (Coinbase Wallet) software, as par combination to run, namely:	t of the Facilitator, need
	• Memory (RAM), used phone).	in the computing device (such as a co	omputer or mobile
	Transaction record sec	tor (stores transactions that haven't be ia the crypto software wallets	en submitted to the
	<ul> <li>a first key pair sector w</li> <li>The asymmetric key pair a first public key – all</li> <li>The wallet software compared to the software</li></ul>	pp (Coinbase Wallet) software, conta which is generated and stored in the w air generated and/or stored consists of found and manipulated via the wallet onnects via the public key or the key p	allet software `a first private key and software.
	(signs) the transaction	with the private key of the key pair.	
	transaction creation, contains: • a first key pair sector v	the Eth Client Browser or Command which is generated and stored on the d air stored consists of a first private ke	evice
	<ul> <li>order to sign transactions, conta</li> <li>a first key pair sector v</li> <li>The asymmetric key pair a first public key – all vault/software.</li> <li>The key management v</li> </ul>	which is stored in the key managemen air generated and/or stored consists of found and manipulated via the key ma vault/software connects via the public	t vault/software s first private key and anagement key or the key pair,
ii. a second		the transaction with the private key of twork interface in order to connect a v	· · · · · · · · · · · · · · · · · · ·
network interface; and	to offer the NFT for sale.		valiet to the marketplace
interface, and	1. Click on the 'Sign In'	Discover Shop Trending Drops	Sign In
		_	
	Shop NFTs	Connect wallet × Use any existing crypto wallet to connect to Colnbase NFT	
	Search Q Search NFTs	Colnbase Wallet >	ms Listed: Recent 🗸 🏢
	Currency	MetaMask >	e al la
	Price range ETH	So WalletConnect >	No.
	Collections Baby Bored Ape Baby Bored Ape	I don't have a crypto wallet  Create a new wallet Create a new Colnbase Wallet Cub	Sister Bored Ape Club Sister Bored Ape Club
	Q Collections	Buy ♡0 ◘ 0 + 0.0725 Buy ♡0 ፬ 0 + 0.0079 B	luy ♡ 39 🖵 5 + 0.035 Buy
		/allet' prompts the user to connect to t	

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 103 of 117 PageID #: 165



Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 104 of 117 PageID #: 166

	Colinbase Net Search Discover Shop Trending Drops Signature requested •
	Shop NFTs
	Search Confirm Wallet 131.488 Items Listed: Recent This work 1000 Tem 2000
	Q Search NFTs Please sign into Colvbase Wallet to connect to Colvbase NFT
	ETH USD
	Price range ETH Centent not loading Content not loading Content not loading Content not loading
	Min         Max         Conservation         Conservation         Decry         Sign           Coldections         Rainbox Bond Age         ScoreRainbox         Baty Bond Age         ScoreRainbox
	Ratinbow Bored Ape Club     Super Nabob Club     Baby Bored Ape     Rich Ape Club
	Cotheredeed for 0 © 92 (05 (0 0 + 0.049) Buy (0 72 (0 19) + 0.0559 Buy (0 76 (0 25) + 0.0259 Buy (0 66 (0 6) + 0.0599 Buy
	ACCOUNT DETAILS → PERSONALIZE YOUR ACCOUNT → FINISH
	Let's pick a
	username
	Username*
	Enter your username
	Required
	Email
	Enter your email We'll send you updates and other cool stuff
	By checking this box, you confirm that you have read and agree to the Coinbase NFT Terms of Service and Privacy Policy and confirm that you are at least 18 years old.
	Continue
	6. The user is logged in as indicated at the top RHS icon.
	ShopNFTs
	Search 131,485 items Listed: Recent 🗸 🏢 🏭
	Collections Kawall Tigers Kawall Tigers Collection #439 Rainbow Bored Ape Club Rainbow Bored Ape Club Super Football APE Club Super Football APE Club Rock Ape Planet Rock Ape Planet
	Control Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control Control         Control         Control         Control         Control         Control         Control         Control         Control         Control         Control         Control         Control         Control         Control         Control         <
iii. a second	The hardware device, hosting the API software/libraries (used by an App), such as a
computer processor	computer or mobile phone.
coupled to the second	
memory and	

# Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 105 of 117 PageID #: 167

the second network interface, the second computer processor configured to:		
A. read the second private key from the second memory;	The hardware device, hosting the API software/libraries (used by an App), such as a computer or mobile phone. The App requires the user to authenticate themselves via the Wallet private key either before using the App or for sign individual transactions. The pri key is encrypted on the device and accessed for reading (and subsequently used to calcula transaction signature) by the app/wallet from the passcode/protection mechanism of the app/wallet.	
	Coinbase Wallet	
	Confirm transaction	
	COINDASE NFT	
	Request from	
	nft.coinbase.com	
	Wallet used 0xA3B0012	
	Cost \$12191.27 3.7 ETH	
	Network fee 0 \$187.73 - \$251.63 🌣	
	Total cost \$12379 - 12442.9	
	Network Ethereum	
	Network fees are high right now. Not paid to Coinbase Learn more	
	Cancel Confirm	
	Coinbase Wallet requesting confirmation to access (read) the private key to authorise and sign the transaction.	1
B. read the inchoate data record	The published Inchoate Data Record is the offered NFT for sale on a marketplace. The following is an example from the Coinbase NFT Marketplace.	

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 106 of 117 PageID #: 168

	coinbase NFT	Search	Discover Sho	p Trending	Drops		۵ 😅
		4 A	<b>Å</b> +		Sell for free 0% Coinbase fees for a lin	nited time	
					Kawaii Tigers #439	Collection	
				KAR I	Kawail Tigers is a collection playing on the Ethereum b project with the end goal Chip NFT project. Bringing Metaverse.	blockchain. A community b to possibly become a Blue	
		X	P		Collection Kawali Tigers		
		Current price 0.0167 ETH		\$28.65	Created by	2	8r
		N	Buy now Take an offer		Attributes Background Gradient 2	Body Light Red	
		Listing e	nds Sep 17 at 11:07 AM		Face Smile	Hat Viking Outfit	
	-	oinbase.com/nft					
C. compute a second cryptographic signature from the		are device, hostin r mobile phone. g the App.					
second private key;							

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 107 of 117 PageID #: 169

	🔴 😑 🔹 Coinbase Wallet	
	Confirm transaction	
	coinbase NFT Request from nft.coinbase.com	
	Wallet used 0xA3B0012	
	Cost \$12191.27 3.7 ETH	
	Network fee 0 \$187.73 - \$251.63 🗘	
	Total cost \$12379 - 12442.9	
	Network Ethereum	
	Network fees are high right now. Not paid to Coinbase Learn more	
	Cancel Confirm	
	<ul> <li>Coinbase Wallet requesting confirmation to access (r calculate signature and sign the transaction.</li> <li>eth_SignTransaction</li> <li>web3.eth.signTransaction(transactionObject, address Signs a transaction.</li> <li>Parameters</li> <li>Object - The transaction data to sign. See web3.eth.set String - Address to sign transaction with.</li> <li>Function - (optional) Optional callback, returns an er result as second.</li> <li>https://web3js.readthedocs.io/en/v1.5.2/web3-eth.htm</li> </ul>	[, callback]) endTransaction() for more. ror object as first parameter and the <u>nl#signtransaction</u>
D. create a complete data	The App (Coinbase Wallet) initiates a smart contract above message.	
record comprising: I. the commit	From: <u>https://eips.ethereum.org/EIPS/eip-721</u>	
input;	ERC-721 standardizes a safe transfer function safe and without a bytes parameter) and an unsafe func- be initiated by: • The owner of an NFT	

		• The approved address of an NFT		
		<ul> <li>An authorized operator of the current owner of an NFT</li> </ul>		
II.	the output data;	<ul> <li>The output data includes:</li> <li>The signed acceptance of the offered NFT based on the value description of the Metadata URI link.</li> <li>The disbursement amount to the seller and the originator based on the royalty fee definition of the NFT.</li> </ul>		
III.	the first cryptographic signature, and	The seller has signed the 'approve' smart contract function call to authorise transfer of ownership when the terms are met.		
IV.	the second cryptographic signature,	The buyer signs the transaction. The hardware device, hosting the API software/libraries (used by an App), such as a computer or mobile phone. The App requires the user to authenticate themselves via a Walle before using the App.		
		Coinbase Wallet		
		Confirm transaction		
		coinbase NFT Request from nft.coinbase.com		
		Wallet used 0xA3B0012		
		Cost \$12191.27 <b>3.7ETH</b>		
		Network fee • \$187.73 - \$251.63 🌣		
		Total cost \$12379 - 12442.9		
		Network Ethereum		
		Network fees are high right now. Not paid to Coinbase Learn more		
		Cancel Confirm		
		Coinbase Wallet requesting confirmation to access (read) the private key to authorise, calculate signature and sign the transaction.		
		eth_SignTransaction		
		web3.eth.signTransaction(transactionObject, address [, callback])		

# Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 109 of 117 PageID #: 171

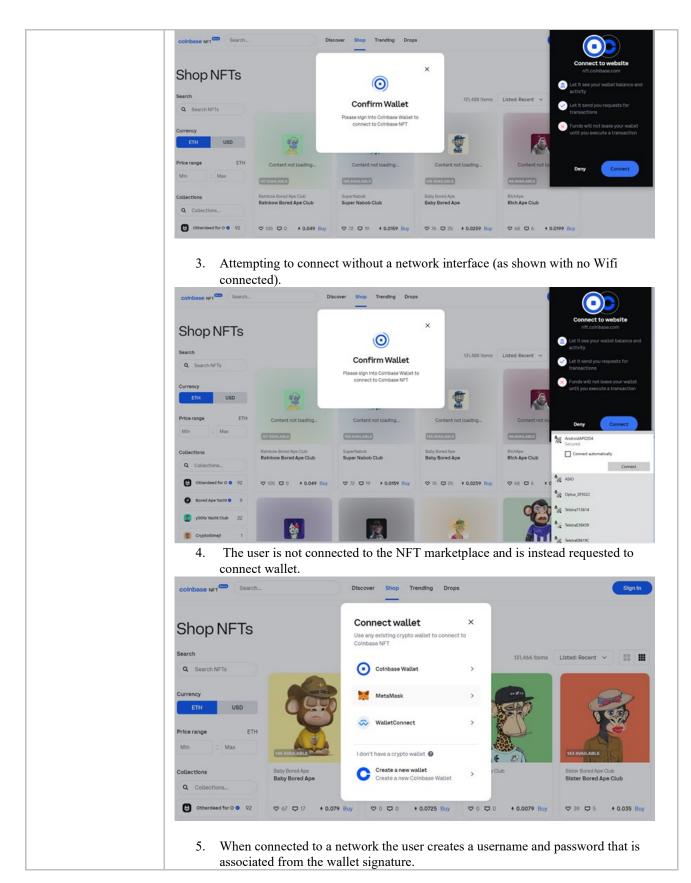
	Signs a transaction.
	Parameters Object - The transaction data to sign. See web3.eth.sendTransaction() for more. String - Address to sign transaction with. Function - (optional) Optional callback, returns an error object as first parameter and the result as second. <u>https://web3js.readthedocs.io/en/v1.5.2/web3-eth.html#signtransaction</u>
E. create a transaction by submitting the complete data record to the transfer mechanism;	The NFT approved for sale is accepted by the buyer with the safeTransfer function call to transfer ownership to the buyer.         The App (Coinbase Wallet) creates a transaction message to be sent to the Ethereum Network. RPC JSON API calls are defined to allow interaction with the Ethereum network. The App constructs the following message which is aligned with this standard. <b>eth_sendTransaction</b> (EIP1559)         Creates new message call transaction or a contract creation, if the data field contains code.         Object - The transaction object
	<ol> <li>from - String Number: The address for the sending account. Uses the web3.eth.defaultAccount property, if not specified. Or an address or index of a local wallet in web3.eth.accounts.wallet.</li> <li>to - String: (optional) The destination address of the message, left undefined for a contract-creation transaction.</li> <li>value - Number String BN BigNumber: (optional) The value transferred for the transaction in wei, also the endowment if it's a contract-creation transaction.</li> <li>gas - Number: (optional, default: To-Be-Determined) The amount of gas to use for the transaction (unused gas is refunded).</li> <li>gasPrice - Number String BN BigNumber: (optional) The price of gas for this transaction in wei, defaults to web3.eth.gasPrice.</li> <li>type - Number String BN BigNumber: (optional) A positive unsigned 8-bit number between 0 and 0x7f that represents the type of the transaction.</li> <li>maxFeePerGas - Number String BN: (optional, defaulted to (2 * block.baseFeePerGas) + maxPriorityFeePerGas) The maximum fee per gas that the transaction is willing to pay in total</li> <li>maxPriorityFeePerGas - Number String BN (optional, defaulted to 1 Gwei) The maximum fee per gas to give miners to incentivize them to include the transaction (Priority fee)</li> <li>accessList - List of hexstrings (optional) a list of addresses and storage keys that the transaction plans to access</li> <li>data - String: (optional) Either a ABI byte string containing the data of the function</li> </ol>
	<ul> <li>call on a contract, or in the case of a contract-creation transaction the initialisation code.</li> <li>11. nonce - Number: (optional) Integer of the nonce. This allows to overwrite your own pending transactions that use the same nonce.</li> <li>12. chain - String: (optional) Defaults to mainnet.</li> <li>13. hardfork - String: (optional) Defaults to london.</li> </ul> Returns <ul> <li>4. DATA, 32 Bytes - the transaction hash, or the zero hash if the transaction is not yet available.</li> </ul>

	The App (Coinbase Wallet) initiates a smart contract call populating the data field of the above message.	
	From: https://eips.ethereum.org/EIPS/eip-721	
	ERC-721 standardizes a safe transfer function safeTransferFrom (overloaded with	
	<ul> <li>d without a bytes parameter) and an unsafe function transferFrom. Transfers mainitiated by:</li> <li>The owner of an NFT</li> </ul>	
	<ul> <li>The owner of an NFT</li> <li>The approved address of an NFT</li> </ul>	
	<ul> <li>An authorized operator of the current owner of an NFT</li> </ul>	
7. c. the second client	The Computing Device   Facilitator consists of:	
comprises: i. a third memory for storing a	<ul> <li>the Coinbase (Owned, managed or offered) Ethereum Validator Full Nodes; and</li> <li>the Coinbase (Owned, managed or offered) Ethereum supporting Archive Nodes and Light Nodes; and</li> </ul>	
third	• the Coinbase (Ethereum compatible) wallets;	
asymmetric key pair, the	Where both the Coinbase Nodes and the Coinbase Ethereum compatible end user wallet device are networked to have direct or indirect communication with each other.	
third asymmetric	device are networked to have direct of indirect communication with each other.	
key pair	Client Device	
comprising a	The First Client is the end user device that accepts an offer for an NFT.	
third private key and a	The Second Client is the end user device that authorises an NFT to be offered for sale.	
third public	Three (3) Client instances have been identified that represent various implementations that	
key;	exist.	
	<ul> <li>1. Client is a device running an App (Coinbase Wallet) that uses a Javascript Ethereum Provider API (as per EIP-1193) such as web3.js or ethers.js. This device will also use a key storage wallet with EIP-191 or EIP-712 signing support (such as Metamask/Coinbase Wallet) to send an NFT transactions. The use of an EIP- 1193 based Ethereum Provider API or EIP-191/EIP-712 signing support means the Client is considered to be part of the Ethereum Network. http://eips/ethereum.org/EIPS/eip-191 http://eips/ethereum.org/EIPS/eip-712 http://eips/ethereum.org/EIPS/eip-1193</li> </ul>	
	2. The Computing Device and the Client are the same device where the Client is a Client Browser or Command Line Interface on a Coinbase Ethereum Node used to create, sign and submit NFT transactions to the Coinbase Ethereum Node for processing.	
	3. Client is a device running an App (Coinbase Wallet) that uses a Javascript Ethereum Provider API (as per EIP-1193) such as web3.js or ethers.js This device will also use a key storage wallet. This device uses private key management in order to sign transactions	
	Where the Client running the App (Coinbase Wallet) software, as part of the Facilitator, need a computer hardware/software combination to run, namely:	
	• Memory (RAM), used in the computing device (such as a computer or mobile phone).	
	Transaction record sector (stores transactions that haven't been submitted to the blockchain yet) kept via the crypto software wallets	

# Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 111 of 117 PageID #: 173

	1			
	<ul> <li>Where the Client running the App (Coinbase Wallet) software, contains: <ul> <li>a first key pair sector which is generated and stored in the wallet software</li> <li>The asymmetric key pair generated and/or stored consists of a first private key and a first public key – all found and manipulated via the wallet software.</li> <li>The wallet software connects via the public key or the key pair, and authorizes (signs) the transaction with the private key of the key pair.</li> </ul> </li> <li>Where the Client interacts with the Eth Client Browser or Command Line Interface to transaction creation, contains: <ul> <li>a first key pair sector which is generated and stored on the device</li> <li>The asymmetric key pair stored consists of a first private key and a first public key.</li> </ul> </li> <li>Where the Client interacts with App (Coinbase Wallet) with Private key management in order to sign transactions, contains: <ul> <li>a first key pair sector which is stored in the key management vault/software</li> <li>The asymmetric key pair generated and/or stored consists of a first private key and a first public key.</li> </ul> </li> </ul>			
	vault/software.	ault/activens compacts via th	a multic leave on the leave noin	
		he transaction with the priva	ne public key or the key pair,	
ii. a third network interface; and	The Client device requires a net marketplace to offer the NFT for 1. Click on the 'Sign In' b	work interface in order to co		
	coinbase NFT Search	Discover Shop Trending Drops	Sign in	
	Shop NFTs	Connect wallet × Use any existing crypto wallet to connect to Coinbase NFT	131,466 lterns Listed: Recent 🗸 🏢	
	Q Search NFTs	Coinbase Wallet >		
	Currency	💓 MetaMask >		
	Price range ETH Min 2 Max	WalletConnect >		
	Collections Baby Bored Ape Q. Collections	Create a new wallet Create a new Coinbase Wallet	Club Sister Bored Ape Club Sister Bored Ape Club	
	Conterdeed for 0 ● 92 ♥ 67 ◘ 17 + 0.079	Buy ♡0 ◘ 0 + 0.0725 Buy ♡0 ◘ 0	0 <b>+ 0.0079 Buy</b> ♡ 39 ◘ 5 <b>+ 0.035 Buy</b>	
	<ol> <li>Selecting 'Coinbase Wallet' prompts the user to connect to the website.</li> </ol>			

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 112 of 117 PageID #: 174



Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 113 of 117 PageID #: 175

	colinbase NeT Search Discover Shop Trending Drops Signature requested •
	Shop NFTs × Messae con
	Search T11.400 Home Linded Based will be good in.
	Search NFTs     Confirm Wallet     Confirm Wal
	Price range ETH Content not loading Content not loading Content not loading
	Min ( Max ( 27.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.04.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00) ( 29.404.00)
	Collections Ranbox Bored Ape Club Dopritable Bary Bored Ape Club Club Bary Bored Ape Club Club Bary Bored Ape Club
	Otherdeed for 0      92     V 105      0     + 0.049     Buy     V 72     10     + 0.0159     Buy     V 72     + 0.0159     Buy     V 75     + 0.0259     Buy     V 65     + 0.0199     Buy
	ACCOUNT DETAILS → PERSONALIZE YOUR ACCOUNT → FINISH
	Let's pick a
	username
	Username*
	Enter your username
	Required
	Email
	Enter your email We'll send you updates and other cool stuff
	By checking this box, you confirm that you have read and agree to the Coinbase
	NFT Terms of Service and Privacy Policy and confirm that you are at least 18 years old.
	Continue
	6. The user is logged in as indicated at the top RHS icon.
	coinbase Net Search. Discover Shop Trending Drops Q 🖸
	ShopNFTs
	Search 131,485 items Listed: Recent V
	Q Search NFTs
	Collections Known Tigers Balakow Bored App Club Super Forball APE Club Rock App Planet
	Collections         Kawall Tigers Collection #439         Rainbow Bored Ape Club         Super Football APE Club         Rock Ape Planet
	C Otherdeed for O © 92 O 1 C 0 + 0.0167 Buy O 103 C 0 + 0.0065 Buy O 239 C 1 + 0.035 Buy O 70 C 0 + 0.0087 Buy
iii. a third computer	The hardware device using the API software/libraries, such as a computer or mobile phone.
processor	
coupled to the third memory	
and the third	

network interface, the			
third			
computer processor			
configured to			
read the third private key			
from the third			
memory; and			
wherein the at least one of the first client device or the second client device signs the inchoate	App. The wallet private	key is used to sign the tran	hemselves via a Wallet before using the saction for the EVM smart contract call.
data record and saves a copy of the inchoate data	Coin	base Wallet	
record on at least one of the first client device or the second client device.	Confirm transaction		
	coinbase NFT		
	Request from		1
	nft.coinbase.c	om	
	Wallet used	0xA3B0012	
	Cost	\$12191.27 <b>3.7 ETH</b>	
	Network fee 0	\$187.73 - \$251.63 🌣	
	Total cost	\$12379 - 12442.9	
	Network	Ethereum	
	Network fees are high Coinbase Learn more	a right now. Not paid to $\times$	
	Cancel	Confirm	
	Coinbase Wallet reques	ting confirmation to access	the private key to authorise and sign the
	transaction.		the private key to autionise and sign the
		atory for the inchoate data r the Patent general description	ecord to be saved by the device as it is ion.
			commit transaction record by signing the
		ction record and <u>optionally</u> insaction record comprising	saves a copy in non-transitory memory, g:
	[0111] 15. Optionally, t in non-transitory memo		of the complete commit transaction record
	2		

# Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 115 of 117 PageID #: 177

wherein the transfer mechanism comprising a decentralized digital currency that comprises a distributed ledger that	Ether is used as the decentralised currency. The Ethereum network maintains a distributed ledger without the need for a trusted central authority. From the Ethereum yellow paper.		
enables processing the transaction between the first client device and the second client device	2.1. Value. In order to incentivise computation within the network, there needs to be an agreed method for transmitting value. To address this issue, Ethereum has an intrinsic currency, Ether, known also as ETH and sometimes referred to by the Old English $\overline{D}$ .		
without the need of a trusted central authority,	https://ethereum.github.io/yellowpaper/paper.pdf		
wherein the transaction is created by broadcasting the	The NFT approved for sale is accepted by the buyer with the safeTransfer function call to transfer ownership to the buyer.		
complete data record for transmitting and receiving among network participants in the computer network for	The App (Coinbase Wallet) creates a transaction message to be sent to the Ethereum Network. RPC JSON API calls are defined to allow interaction with the Ethereum network. The App constructs the following message which is aligned with this standard.		
recording in the	eth_sendTransaction (EIP1559)		
distributed ledger, and	Creates new message call transaction or a contract creation, if the data field contains code.		
	Object - The transaction object		
	1. from - String Number: The address for the sending account. Uses the		
	web3.eth.defaultAccount property, if not specified. Or an address or index of a		
	local wallet in web3.eth.accounts.wallet.		
	2. to - String: (optional) The destination address of the message, left undefined for a		
	contract-creation transaction.		
	3. value - Number String BN BigNumber: (optional) The value transferred for the transaction in wei, also the endowment if it's a contract-creation transaction.		
	<ol> <li>gas - Number: (optional, default: To-Be-Determined) The amount of gas to use for the transaction (unused gas is refunded).</li> </ol>		
	5. gasPrice - Number String BN BigNumber: (optional) The price of gas for this transaction in wei, defaults to web3.eth.gasPrice.		
	<ul> <li>6. type - Number String BN BigNumber: (optional) A positive unsigned 8-bit number between 0 and 0x7f that represents the type of the transaction.</li> </ul>		
	<ol> <li>maxFeePerGas - Number String BN: (optional, defaulted to (2 *</li> </ol>		
	block.baseFeePerGas) + maxPriorityFeePerGas) The maximum fee per gas that the transaction is willing to pay in total		
	<ol> <li>maxPriorityFeePerGas - Number String BN (optional, defaulted to 1 Gwei) The maximum fee per gas to give miners to incentivize them to include the transaction (Priority fee)</li> </ol>		
	9. accessList - List of hexstrings (optional) a list of addresses and storage keys that		
	the transaction plans to access 10. data - String: (optional) Either a ABI byte string containing the data of the function		
	call on a contract, or in the case of a contract-creation transaction the initialisation code.		
	11. nonce - Number: (optional) Integer of the nonce. This allows to overwrite your		
	own pending transactions that use the same nonce.		
	<ol> <li>chain - String: (optional) Defaults to mainnet.</li> <li>hardfork - String: (optional) Defaults to london.</li> </ol>		
	Returns		
	5. DATA, 32 Bytes - the transaction hash, or the zero hash if the transaction is not yet available.		

# Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 116 of 117 PageID #: 178

	https://web3js.readthedocs.io/en/v1.5.2/web3-eth.html#sendtransaction		
	The App (Second Client as part of the Facilitator) initiates a smart contract call populating the data field of the above message.		
	From: https://eips.ethereum.org/EIPS/eip-721		
	<ul> <li>ERC-721 standardizes a safe transfer function safeTransferFrom (overloaded with and without a bytes parameter) and an unsafe function transferFrom. Transfers may be initiated by:</li> <li>The owner of an NFT</li> <li>The approved address of an NFT</li> <li>An authorized operator of the current owner of an NFT</li> </ul>		
wherein at least one of the computer device, the first client device, or the second client device verifies the recording of the complete data record in the distributed ledger by observing an external state	The First Client verifies that the Complete Data Record (Transaction) has been recorded by the network as a complete transaction record. In the below image this is shown in the 'History' tab with the transaction shown as confirmed.		
	3:57       .nl ? I	3:57      II ♥ ■)         ←       Badger DAO         App wallet       0 BADGER         0 BADGER       #0.0200000 BA         Pending       +\$35.34         Speed up       Cancel         Trade BADGER       -0.0200000 BA         Decentralized exchange       -\$35.34         Speed up       Cancel         Trade BADGER       -0.0200000 BA         Decentralized exachange       -\$35.34         Speed up       Cancel         Trade BADGER       -0.0200000 BA         Decentralized exachange       -\$35.34         Speed up       Cancel         Trade       Cancel	
	https://help.coinbase.com/en/coinbase/trading-and-funding/istatus         When a transaction is sent, a transaction hash is received. The transaction details. Use the JSON RPC API command getThe hash}) to retrieve the transaction details. The returned block transaction has been mined and included into a block.         Then call JSON RPC API eth_blockNumber to get the curred confirmations is the eth_blockNumber result minus the eth_result.         https://github.com/ethereum/wiki/wiki/JSON-RPC#eth_blockNumber.	his can be used to retrieve ransactionByHash({transaction Number should be non-null if the ent block height. The number of getTransaction blockNumber	

Case 1:22-cv-01253-UNA Document 1-3 Filed 09/22/22 Page 117 of 117 PageID #: 179

EXHIBIT 3

\_