



OBSIDIAN NETWORK

The Proof of Time Blockchain

White Paper

Obsidian Seal Coin (OBS) · Mainnet · Chain ID 7777

The ledger, its consensus, its economy and the products built on it

DOCUMENT

White Paper, version 1.1

DATE

24 October 2026

RELEASE

Obsidian Core 2.0.4 · Protocol 2.0.2

PUBLISHER

Emolux Labs

mineobs.network

app.mineobs.network

wallet.mineobs.network

api.mineobs.network



CONTENTS

Contents

1 Abstract	4
2 Introduction	5
2.1 Design principles	5
2.2 What makes Obsidian different	5
3 System architecture	6
3.1 The four networks	7
3.2 Official links	7
4 Proof of Time consensus	8
4.1 The chain's own clock	8
4.2 The producer set and the schedule	8
4.3 Fork choice	9
4.4 Equivocation and suspension	9
4.5 PoT Difficulty, Time-Rate and checkpoint finality	9
5 Cryptography and identity	10
5.1 Wallet custody	10
6 The ledger	11
6.1 Blocks	11
6.2 Transactions	11
6.3 Gas	12
6.4 State, supply and genesis	12
7 Mining	13
7.1 The reward schedule	13
7.2 Account-verified claims	14
7.3 Replay safety	14
7.4 Withdrawals	14
8 Economics	15
8.1 Issuance	15
8.2 What things cost	15
8.3 ONS revenue and the 90 / 10 split	16
9 Obsidian Name Service	17
9.1 Ownership operations	17
9.2 Resolution	18



10 ONS Marketplace	19
10.1 Guarantees	19
11 Node runners	20
11.1 Becoming a node runner	20
11.2 How a node is measured	20
11.3 Settlement	20
12 The Obsidian product suite	21
12.1 The Platform	21
12.2 The Web App	21
12.3 The Wallet	22
12.4 The Browser Extension	22
12.5 Obsidian Node	22
12.6 Obsidian Core and the command line	23
12.7 The explorer and the edge layer	23
13 Security design	24
13.1 In the protocol	24
13.2 In the products	24
14 Developer reference	25
14.1 Read routes	25
14.2 Write routes	25
14.3 Signing a transaction	26
Appendix A Protocol parameters	27
Appendix B Glossary	28
Appendix C Document information	29
Version history	29



OVERVIEW

1 Abstract

Obsidian Network is a Proof of Time blockchain with a fixed maximum supply of **21,000,000 OBS**. Its native coin, Obsidian Seal Coin (ticker **OBS**), moves on a public ledger that anyone can read, verify and run. Mainnet is live on chain ID **7777** with the address prefix **obs**.

The network is built for ordinary people, most of them on a phone. A person creates a wallet in the browser, claims a small protocol-defined reward every four hours, sends and receives OBS, registers a human-readable **.obs** name, trades names on a built-in marketplace, and can run a node that is paid from the protocol's only revenue stream. Each of those actions is chain state: it is recomputed by every node and rejected if it breaks a rule. None of it lives in a company database.

Consensus is **Proof of Time**. The right to produce a block comes from a deterministic schedule over registered, live nodes and from the chain's own clock, not from a race of computation. There is no hash puzzle, no block nonce, and no advantage in computing faster. Hashing and signatures protect integrity and identity; time is what orders the chain.

This paper describes the network and every product in the Obsidian ecosystem as they exist today: the ledger and its rules, the mining and reward economy, the Obsidian Name Service and its Marketplace, node-runner rewards, and the five ways people use the network: the Platform, the Web App, the Wallet, the browser Extension and the Obsidian Node desktop application.

At a glance	
Network	Obsidian Network · Mainnet is live
Coin	Obsidian Seal Coin (OBS), a native coin of its own chain
Consensus	Proof of Time: deterministic producer schedule gated by protocol time
Maximum supply	21,000,000 OBS, enforced on every issuance
Block time	5 seconds (target)
Chain ID / address prefix	7777 / obs1...
Issuance	Mining claims, plus a one-time 100,000 OBS genesis allocation. No other path exists.
Fees	Gas of 0.02% of the amount (maximum 0.01 OBS) to the Mining Pool; 0.05 OBS for a .obs name
Revenue	ONS fees only: 90% to the Node Runner Reward Pool, 10% to the treasury wallet
Release	Obsidian Core 2.0.4, protocol 2.0.2, open source under Apache-2.0
Official links	mineobs.network · app.mineobs.network · wallet.mineobs.network · api.mineobs.network
Genesis id and live network details	Visit the main web page at https://mineobs.network



WHY OBSIDIAN EXISTS

2 Introduction

Most blockchains ask a participant to commit capital or machinery. Obsidian asks for something every person already has: time. The core activity on the network is a claim a person makes from a phone every four hours, six times a day. That design has exactly one hard problem, which is deciding **what time it is without letting the participant decide**.

Proof of Work sidesteps the problem by making block production expensive. Obsidian solves it directly. The chain carries its own clock, derived from block timestamps, and every rule that matters (claim windows, name expiry, reward periods and transaction expiry) is evaluated against that clock. A participant's device clock, a website's clock, a CDN's clock and a node operator's clock cannot move it.

2.1 Design principles

- **The chain is the source of truth.** Names, balances, claims and node registrations are ledger state. Websites, caches and databases only display it.
- **Users hold their keys.** Wallet keys are created in the user's browser or device and are never transmitted. Nodes see only public addresses and finished, signed transactions.
- **One network, one set of rules.** Every product reaches the chain through the same node RPC and builds transactions with the same encoder the node uses.
- **Prices belong to the protocol.** Every fee is denominated in OBS and fixed by consensus. No external price feed can change what anything costs.
- **Participation is open and measured, not self-reported.** A node is rewarded for uptime and behaviour that other nodes attested; there is no field in which a node reports its own performance.
- **Determinism everywhere.** Reward amounts, schedules, splits and eligibility are exact integer computations that every node reproduces identically.

2.2 What makes Obsidian different

Property	Obsidian Network
Right to produce a block	The deterministic producer schedule, once protocol time has advanced. Not computation spent in a race.
Headline network metric	Time-Rate: verified blocks and transactions per minute of protocol time
Entry cost to run a node	None. Registration needs no gas, no deposit and no wallet signature.
Mining	A signed claim that the protocol clock permits: one every four hours, six per day, per wallet
Names	A .obs name maps to one wallet address as consensus state, with transfer, recovery and a marketplace
Application data	First-class chain state: names, claims, listings, escrow and the node registry
Native coin	OBS is the native coin of the Obsidian chain. It is not a token on another blockchain.



THE NETWORK AND ITS PRODUCTS

3 System architecture

Obsidian is a family of products around one chain. The node is the reference implementation of the protocol; every other product is a client of its RPC. None of them is a second blockchain, and none holds keys or balances on the user's behalf.

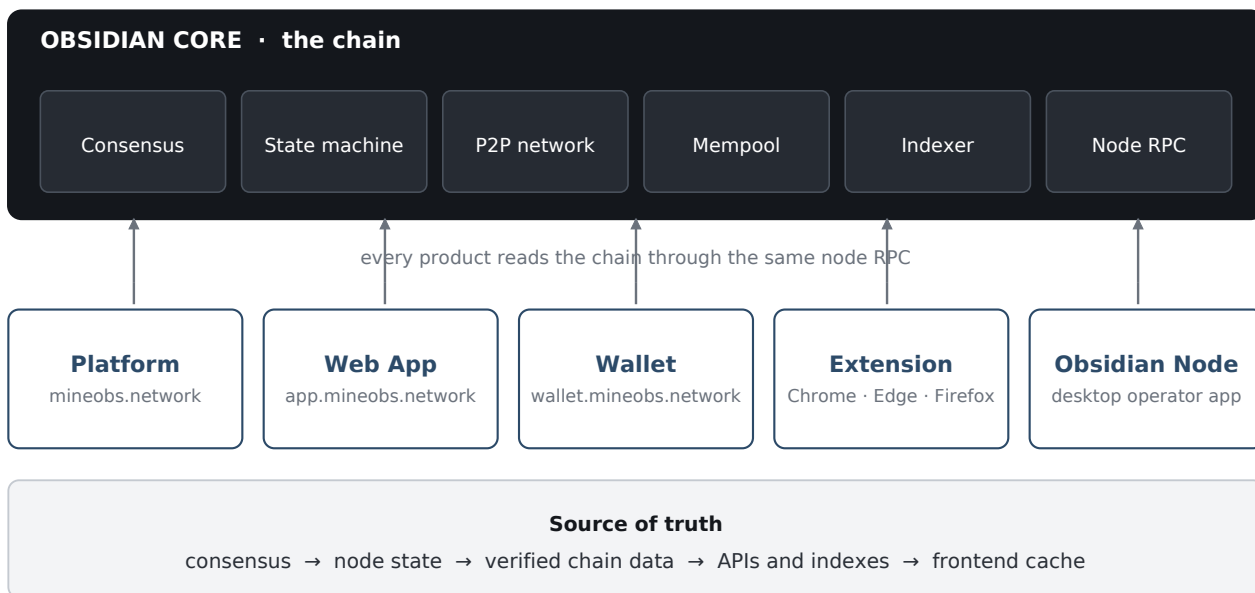


Figure 1. Every Obsidian product reads and writes the chain through the same node RPC.

Component	Role
Obsidian Core	The node. Validates and serves the chain: block production, peer-to-peer networking, mempool, RPC (default port 8630 on mainnet), indexer and the state machine.
Platform (mineobs.network)	The account system and the public sites: Mine OBS, Wallet, Explorer, ONS and Marketplace, Node Runners, Developer.
Web App (app.mineobs.network)	The Obsidian client in the browser: account, mining, wallet, explorer, names and an API console. Shares accounts with the Platform.
Wallet (wallet.mineobs.network)	A non-custodial browser wallet: create or import, receive, send, activity, encrypted backup.
Browser Extension	The Obsidian client as a Manifest V3 extension for Chrome, Edge and Firefox, with claim alerts.
Obsidian Node	The desktop application (Windows, macOS, Linux) that runs a real node with a local operator dashboard.
Edge layer	A Cloudflare worker that caches and forwards read traffic. It signs nothing and decides nothing.



3.1 The four networks

Obsidian runs four isolated networks from one codebase. Mainnet carries real value; the others exist for rehearsal and development.

Network	Chain ID	Address prefix	RPC / P2P port	Purpose
mainnet	7777	obs1...	8630 / 8631	The live network and real value
testnet	7778	tobs1...	18630 / 18631	Public rehearsal
staging	7779	sobs1...	28630 / 28631	Release candidates
devnet	7780	dobs1...	38630 / 38631	Local development

Networks are separated by genesis document, genesis id, hash, parameters hash and chain ID. A node rejects a block, transaction or peer from another network and refuses to start on a data directory written by a different one. Address prefixes differ on every network, so a mainnet address cannot be mistyped into a test-network transaction.

3.2 Official links

These are the official addresses of Obsidian Network. Use only these to reach the network's products.

Product	Official address	Purpose
Main web page	https://mineobs.network	The home of the network: Mine OBS, wallet, explorer, ONS and Marketplace, node runners, developer resources and the account system
Web App	https://app.mineobs.network	The full Obsidian client in a browser or on a phone
Wallet web app	https://wallet.mineobs.network	The non-custodial browser wallet
API	https://api.mineobs.network	The official network API

Genesis id and live network details

For the genesis id and other current network information, visit the main web page at <https://mineobs.network>.



HOW BLOCKS ARE ORDERED

4 Proof of Time consensus

Obsidian's consensus identity is **PROOF_OF_TIME**. A block is valid when it is produced by the node whose turn it is, carries a timestamp the chain's clock accepts, and passes every state rule. How much computation anyone spent is irrelevant.

	Proof of Work	Obsidian Proof of Time
What buys the right to produce a block	Computation spent in a race	The producer schedule, once protocol time has advanced
Headline metric	Hash rate	Time-Rate: verified blocks and transactions per minute
"Difficulty" means	How many hashes a block costs	PoT Difficulty: how block spacing compares with the 5-second target
Fork choice	Most accumulated work	Checkpoint prefix, fixed PoT Weight, height, then lowest block hash
Does a faster machine help?	Yes	No

4.1 The chain's own clock

The clock is the **median timestamp of the last 11 blocks**. A median is used rather than a maximum or a mean so a single absurd timestamp, honest or hostile, cannot move it. Every block timestamp must satisfy three rules:

BLOCK TIMESTAMP RULES

```
timestamp > median time past of the last 11 ancestors
timestamp >= parent timestamp + 1 second
timestamp <= the validating node's own clock + 10 seconds
```

The first two clauses are computed entirely from chain data, so every node reaches the same verdict. A node's local clock is used only to reject a block that claims the future. It can never make a backdated block valid.

4.2 The producer set and the schedule

Any node can become a producer by registering with its own identity key. Registration is free: no gas, no deposit and no wallet signature. Every registered node that is mature and live joins one deterministic schedule.

Tier	Who is in the producer set
Tier 1 (normal)	Registered, not banned, mature (registered at least 64 blocks ago), live (a sign of life within 2 days of protocol time) and not suspended.
Tier 2	Used only if Tier 1 is empty: every registered, non-banned node, so a chain whose producers vanished can continue.
Open	Used only when no registered node exists at all: the founding state of a fresh chain. It ends with the first accepted registration.

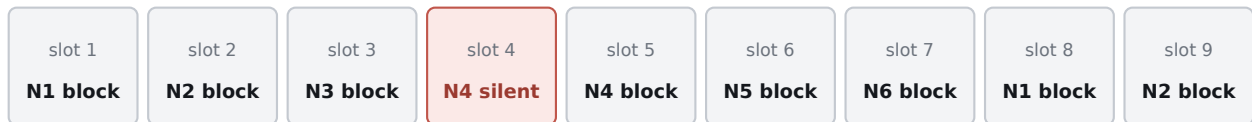


The set is sorted by producer address, so registration order and arrival time carry no weight. For each slot exactly one producer is named, derived from the parent state and the two header timestamps alone:

THE SCHEDULE

```
round(parent, block) = max(0, floor((block.timestamp - parent.timestamp) / 5) - 1)
producer(height, round) = set[(height + round) mod |set|]
```

Producer set, sorted by producer address · $\text{producer}(\text{height}, \text{round}) = \text{set}[(\text{height} + \text{round}) \bmod n]$



protocol time (one slot = 5 seconds)

a silent producer costs one slot, never the chain

After a skipped slot the turn passes to the next node in the set. Three consecutive misses suspend a node for 10 minutes, doubling on each repeat up to 24 hours. Suspension costs turns and reward weight; no funds are ever confiscated.

Figure 2. A silent producer is skipped by the next round. The chain keeps its 5-second pace.

4.3 Fork choice

A branch must contain the checkpointed prefix. Among such branches the greatest cumulative **PoT Weight** wins (every valid block adds exactly one unit), then the greatest height, then the **lowest block hash**. The final rule makes the choice fully deterministic: two honest nodes holding the same candidates choose the same head. Transaction volume, peer arrival order and header-hash grinding buy no authority. A node holds at most 8 competing candidates per height and refuses reorganisations deeper than 256 blocks.

4.4 Equivocation and suspension

A producer that signs two different blocks on the same parent has equivocated. Any registered node can submit both signed headers; every node re-verifies both signatures and, if valid, the identity is **banned permanently** and its reward wallet is released. A ban needs two valid signatures from the producer's own key, so nobody can forge one. A producer that is merely offline is suspended for a deterministic back-off (10 minutes, doubling to a 24-hour ceiling) and returns by itself.

4.5 PoT Difficulty, Time-Rate and checkpoint finality

PoT Difficulty is recomputed from chain history every 720 blocks and reports how actual block spacing compares with the 5-second target. It is a published measurement, not an acceptance rule.

Time-Rate replaces hash rate: verified blocks and transactions per minute of protocol time over a one-day window. Both are served by the node's /pot route.

The protocol also derives a **producer-diversity checkpoint** from chain history alone. A block is checkpointed once enough distinct, established producers have built on it. No votes are cast, no message is gossiped for it, and it is committed in the state root, so every node computes the same answer.



KEYS, ADDRESSES AND SIGNATURES

5 Cryptography and identity

Purpose	Primitive
Transaction, block and node signatures	secp256k1 ECDSA, 64-byte compact signatures, deterministic (RFC 6979), low-S normalised
Address	Bech32 of a 20-byte hash (RIPEMD-160 of SHA-256 of the compressed public key) with a per-network prefix; 42 characters on mainnet
Block ids, Merkle roots, state roots, PoT Weight	SHA-256
Recovery phrase	24 BIP-39 words (256 bits of entropy); derivation path m/44'/7777'/0'/0/index
Browser wallet vault	PBKDF2-SHA256 (600,000 iterations) and AES-256-GCM; passphrase of at least 12 characters
Node identity	A secp256k1 key created on the node; the node id is the 20-byte hash of its public key; the keystore is script and AES-256-GCM
Domain separation	Every signed message type has its own tag, for example OBSIDIAN:TX_SIGNING:v1. A signature made for one purpose is never a valid transaction.

Hashing proves integrity and identity. It is not a competition: computing faster earns nothing, which is the whole content of "Proof of Time, not Proof of Work".

5.1 Wallet custody

Keys are created in the browser from the WebCrypto secure random generator. They are never derived from an email address, username or account id, and no function in the wallet library sends key material over the network. The stored vault holds only the key-derivation parameters, salt, IV and ciphertext plus public labels. The browser wallet builds transaction bodies with the same encoder module the node uses, so a signature produced in a browser verifies on a node.

One recovery phrase produces the same key pair on all four networks; only the address prefix and checksum change. Transactions commit to the chain ID, so a signature made on one network is useless on another.



BLOCKS, TRANSACTIONS AND STATE

6 The ledger

6.1 Blocks

Parameter	Value
Target interval	5 seconds
Maximum block size / transactions	2 MiB / 2,000
Median-time-past window	11 blocks
Allowed future timestamp drift	10 seconds (two slots)
Maximum ordinary reorganisation depth	256 blocks
Header nonce	None. There is no nonce field and no difficulty target in the header.

A header commits to the height, parent hash, timestamp, cumulative PoT Weight, transactions root, events root, state root, parameters hash, protocol version, producer and the producer's signature. A block with one invalid transaction is rejected as a whole.

6.2 Transactions

TRANSACTION ENVELOPE

```
protocolVersion | chainId | sender | nonce | type | gas | body | validUntil | memo | signature
```

Bodies use a deterministic binary codec, so one transaction has exactly one byte representation. The maximum signed transaction is 16 KiB, a memo is at most 256 bytes, and a transaction may emit at most 64 events. Replay protection is the pair (sender, nonce), the chain ID in every signed transaction, and validUntil, which may be at most 240 blocks (20 minutes) ahead. The transaction id is a hash of the canonical bytes, so it is known before broadcast.

Id	Type	Purpose
1	PAYMENT	Transfer OBS. Gas goes to the Mining Pool.
2	ONS	Register, renew, transfer, accept, update, recover and configure a .obs name
7	MINING_CLAIM	Claim the protocol mining reward
9	TREASURY	A treasury action, signed by the treasury wallet only
10	GOVERNANCE	Protocol governance transactions
11	NODE_REGISTRY	Node registration, wallet binding, heartbeats, attestations, fault and equivocation reports
13	GATE_AUTHORITY	Rotate or revoke the mining gate issuer keys, approved by a majority of the current keys
14	ONS_MARKET	List, re-price, delist, buy, offer, accept, decline and cancel an offer on a .obs name

An unknown transaction type is always a hard rejection, never a silent no-op.



6.3 Gas

Every value-bearing transaction pays gas in OBS, on top of the amount, to the Mining Pool:

GAS RULE

```
gas = clamp( floor(amount x 0.02%), 0.000001 OBS, 0.01 OBS )
```

Amount sent	0.001 OBS	1 OBS	10 OBS	50 OBS	100 OBS or more
Gas	0.000001 OBS	0.0002 OBS	0.002 OBS	0.01 OBS	0.01 OBS

The floor (0.000001 OBS) is also the minimum transfer, so every transaction that creates state pays the Mining Pool and spam funds miners instead of burdening them.

6.4 State, supply and genesis

State is a Merkleised map: accounts (balance, nonce), the Mining Pool, the treasury, ONS records, marketplace listings and escrow, and the node registry. Each block commits to a state root and an events root, and every node verifies both after applying the block. Applying a block is a pure function of the parent state and the block, and a candidate state is committed only after its root matches.

THE SUPPLY INVARIANT, CHECKED ON EVERY BLOCK

```
sum(balances) + mining pool + locked escrow and node holdings = total supply  
total supply <= 21,000,000 OBS
```

The one-time **genesis allocation of 100,000 OBS** is awarded atomically to the first protocol-valid mining claim on a chain. That wallet becomes the on-chain treasury wallet. A new account starts with 0 OBS, and no administrator, mint key or alternative issuance path exists.



CLAIMING OBS, ONE SIGNED CLAIM AT A TIME

7 Mining

Mining on Obsidian is a **protocol claim**. The node decides whether a wallet is eligible, how much the claim pays and which claim id is next; the browser only signs and submits. Nothing in an interface can create a claim a node would not accept.

Parameter	Value
Claim interval	4 hours of protocol time per wallet
Claims per cycle	6 per 24 hours
Reward per claim at genesis participation	0.00016666666666666666 OBS (0.001 OBS per wallet per day)
Reduction	0.5% per 100,000 active miners, compounding, in exact integer arithmetic
Floor	0.0002 OBS per wallet per day, never crossed
Active miner	A wallet with at least one valid claim in the last 30 days
Claims per wallet per block	1

7.1 The reward schedule

The reward per claim falls as participation rises, so the schedule distributes the 21,000,000 OBS supply over a very long horizon. The values below are computed from the consensus rule with the same integer arithmetic every node uses.

OBS per wallet per day (six claims)

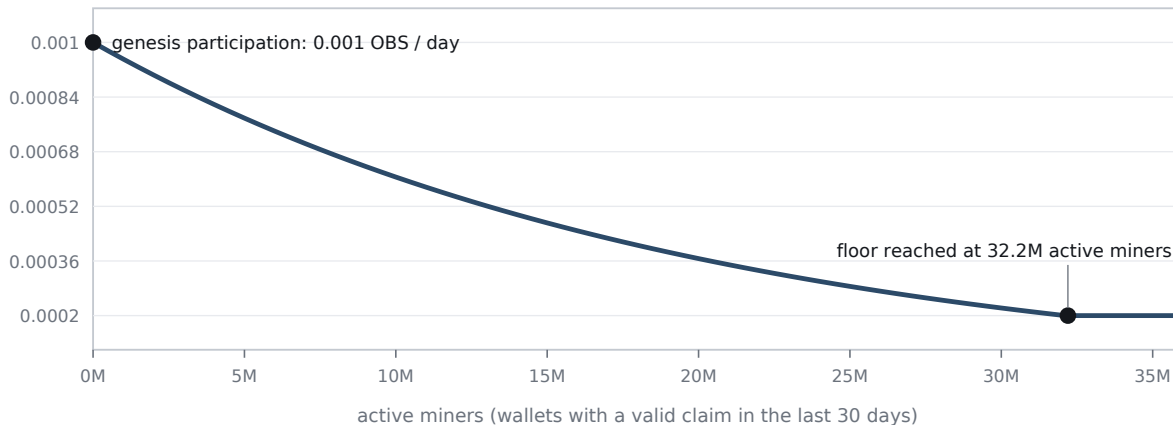


Figure 3. Daily reward per wallet against active miners. The floor is reached at 32.2 million active miners.

Active miners	Reward per day (6 claims)	Reward per claim
Below 100,000	0.001 OBS	0.00016666666666666666 OBS
1,000,000	0.000951110130465771 OBS	0.000158518355077628 OBS
5,000,000	0.000778312557068642 OBS	0.00012971875951144 OBS
10,000,000	0.000605770436490728 OBS	0.000100961739415121 OBS
20,000,000	0.000366957821726167 OBS	0.000061159636954361 OBS



Active miners	Reward per day (6 claims)	Reward per claim
32,200,000 and above	0.0002 OBS (floor)	0.00003333333333333333 OBS

Rewards are settled from the Mining Pool first. Only the remainder is newly issued, so gas recycled into the pool pays real rewards before any new OBS is created.

7.2 Account-verified claims

A claim is valid when it carries a short-lived **gate certificate** issued by the Obsidian account system for the signed-in account's one linked wallet. Mining therefore requires a signed-in account and a connected wallet. The certificate is valid for 15 minutes and for one claim id only. The issuer keys are committed in genesis and can be rotated on chain by approvals from a strict majority of the current keys. The chain still decides eligibility and timing from the wallet's own history; the gate decides only that the claim came through the account system.



The chain, not the interface, decides eligibility and timing from the wallet's own claim history and the block time.

A claim is a signed MINING_CLAIM transaction. The device clock is never consulted.

Figure 4. The path of one claim, from sign-in to a block.

7.3 Replay safety

Each claim carries a unique claim id derived by the node, a per-wallet claim sequence and the wallet's next nonce. One claim per wallet per block means two tabs, two devices or two nodes cannot double-claim; a mined claim cannot be mined again. Eligibility is derived from the chain head timestamp, so a device with the wrong clock can display a wrong countdown but cannot produce an eligible claim.

7.4 Withdrawals

Rewards land in the wallet that claimed them. Moving them is an ordinary payment with standard gas. There is no lock-up.



SUPPLY, FEES AND REVENUE

8 Economics

The protocol has no sale, no premine, no administrator and no mint authority. OBS is issued by exactly two rules, and every fee is paid in OBS at a price fixed by consensus.

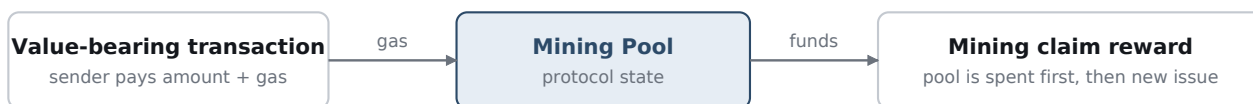
8.1 Issuance

Path	Amount	When
Genesis allocation	Exactly 100,000 OBS, once, ever	To the wallet behind the first protocol-valid mining claim, which becomes the treasury wallet
Mining rewards	Per the schedule in section 7	Per accepted claim, capped by the 21,000,000 OBS supply cap

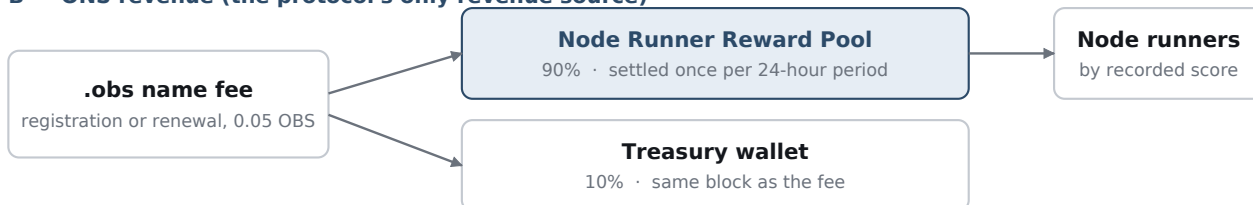
8.2 What things cost

Action	Price	Destination
Gas on any value-bearing transaction	Between 0.000001 OBS and 0.01 OBS (0.02% of the amount)	Mining Pool
.obs name registration	0.05 OBS	ONS revenue, split 90% / 10%
.obs name renewal	0.05 OBS	ONS revenue, split 90% / 10%
Node registration and wallet binding	Nothing	No transfer exists
Name transfer, acceptance, recovery	Nothing	No value moves
Marketplace sale	No commission: the seller receives the full price	Buyer pays ordinary gas to the Mining Pool

A · Gas and mining rewards



B · ONS revenue (the protocol's only revenue source)



C · ONS Marketplace: the seller receives exactly the price; there is no commission. Gas goes to the Mining Pool.

Offers lock the offered OBS in consensus escrow until the owner accepts or declines, or the offer lapses after 7 days.

Figure 5. The three value flows of the protocol.



8.3 ONS revenue and the 90 / 10 split

ONS registration and renewal fees are the protocol's only revenue source. They are split at the moment they are received, inside the state transition, by every node:

EXACT INTEGER SPLIT

```
nodePool = floor(total x 9000 / 10000)
treasury = total - nodePool
```

The treasury takes the remainder, so the two parts always add back to the whole and no unit is lost to rounding. Gas, issuance, mining rewards and the genesis allocation are never ONS revenue. The treasury share is credited in the same block as the fee; the node-runner share accumulates in the Node Runner Reward Pool and is paid once per 24-hour protocol period, by the first block after the period closes.

Each credit is visible on the ledger. The node's address history lists them as ONS_TREASURY_SHARE, ONS_TREASURY_RELEASE and NODE_REWARD entries with amount, block, time and, for a sale, the name and transaction id.

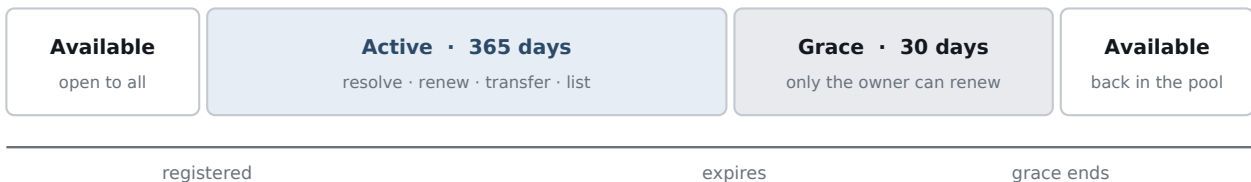


HUMAN-READABLE NAMES AS CHAIN STATE

9 Obsidian Name Service

A **.obs** name maps to exactly one wallet address, and that mapping is blockchain state. Nodes recompute it, reject double registrations and apply transfers as ordinary signed transactions. No DNS, registrar or company sits in the middle.

Parameter	Value
Registration and renewal fee	0.05 OBS each, denominated in OBS with no exchange rate involved
Term	365 days
Grace period after expiry	30 days; only the owner can renew, nobody else can register the name
Length	3 to 63 characters
Reserved names	Protocol-reserved names such as obs, wallet, explorer, node and api can never be registered



A fee of 0.05 OBS buys or renews one year. Names are 3 to 63 characters. A name inside its grace period cannot be registered by anyone else.

Figure 6. The life of a .obs name.

9.1 Ownership operations

A name never changes hands because its owner said so. Every ownership operation is a signed ONS transaction applied in consensus state, atomic, replay-safe and free of gas, since no value moves.

Operation	Who signs	Effect
TRANSFER (offer)	Current owner	Offers the name to one recipient. Nothing changes yet. One open offer at a time; it lapses after 7 days.
ACCEPT_TRANSFER	Offered recipient	Atomically moves the owner, the resolved address and the remaining term to the recipient.
CANCEL_TRANSFER	Owner	Withdraws an open offer.
UPDATE_ADDRESS	Owner	Changes only the address the name resolves to.
RENEW	Current owner	Extends the term. After a transfer the new owner pays all future renewals.
SET_RECOVERY	Owner	Names a recovery address (an empty value clears it).
START_RECOVERY	Recovery address	Proposes a new owner; takes effect only after a 7-day delay.
CANCEL_RECOVERY	Owner	Stops a recovery in flight.
FINALIZE_RECOVERY	Recovery address or new owner	After the delay, if the owner has not cancelled, ownership moves.



A transfer to a mistyped address costs nothing: it is an offer nobody accepts, and it lapses. Any ownership change wipes the previous owner's open offer, pending recovery and recovery address, so a new owner never inherits the previous owner's recovery path. Every recipient or pointer address in a transaction must be in its canonical lowercase spelling, so two spellings of one address can never be two ledger keys.

9.2 Resolution

GET /names/<name> returns the owner, resolved address, registration and expiry times, transfer count and any pending transfer or recovery. Wallets resolve a name from the authoritative on-chain lookup for chain 7777 and show the name, the full address and the network before anything is signed. A listed name does not resolve, and wallets refuse to send to it.



BUYING AND SELLING NAMES ON CHAIN

10 ONS Marketplace

The ONS Marketplace lets the owner of a .obs name sell it for a fixed price in OBS and lets anyone make an offer on a name that is not for sale. It is **protocol**, not an application: a listing, an offer, the OBS locked behind an offer and every ownership change are consensus state that every node recomputes. The interface reads that state and builds transactions the chain validates again.

Operation	Who	Rules
LIST	Owner	The name must be active and not already listed, with no ordinary transfer offer and no recovery in flight. Price at least 0.001 OBS.
UPDATE_PRICE	Owner	The name must be listed by the sender; the new price is at least the minimum.
DELIST	Owner	Allowed during the grace period too. Restores ordinary use.
BUY	Anyone but the seller	The transaction states the price and seller the buyer reviewed; the chain buys only if both still hold.
OFFER	Anyone, on a name that is not listed	The offered OBS is locked in consensus escrow. At most 16 pending offers per name and 32 per maker.
ACCEPT_OFFER	Owner	Atomically pays the owner the full offered amount and transfers the name.
DECLINE_OFFER	Owner	Refunds the escrow to the maker.
CANCEL_OFFER	Maker	Refunds the escrow to the maker.

10.1 Guarantees

- **Atomic sale.** In one state transition the buyer is debited price plus gas, the seller is credited the full price, gas goes to the Mining Pool, the name moves to the buyer and the listing is deleted. A buyer can never pay without receiving the name, and a name can never move without payment.
- **No commission.** The seller of a listing receives exactly the price, and the seller of an accepted offer receives exactly the offered amount.
- **Price protection.** A buyer signs the price and seller they reviewed. If the owner re-prices or delists first, the purchase fails and nothing moves.
- **Listing lock.** While a name is listed, every executor that would use it as an ordinary active name refuses with ERR_NAME_LISTED, and the name does not resolve.
- **Escrow.** Offers lapse after 7 days and their escrow returns to the maker. Settled offers stay in state for 90 days as an audit trail, and the chain's events keep the history permanently.
- **Ownership change clears old rights.** A purchase wipes the old owner's pending transfer, recovery and recovery address.

A wallet's balance reply separates spendable OBS from locked OBS held in offer escrow, with their sum as the total. Public read routes under /market serve listings, offers, name status and history.



INDEPENDENT NODES ARE THE NETWORK

11 Node runners

Obsidian pays for nodes from its only revenue stream. Everything about eligibility, measurement and payout is protocol state and protocol arithmetic: there is no admin panel, allowlist or discretionary allocation.

11.1 Becoming a node runner

The operator pastes an OBS address as the node's reward wallet. That is the whole binding step: no gas, no wallet signature, no deposit. The node then registers itself on chain with its own identity key, which never leaves the operator's machine. A wallet backs at most one node at a time, enforced by a consensus index, and changing a node's wallet is a separate delayed statement (two reward periods) that the current wallet can cancel. The owner of a wallet can release a binding with a signature from that wallet (one period).

11.2 How a node is measured

A node is "up" in an hourly window if the chain contains one of its in-sync heartbeats, or a block it produced, in that window. Peers attest to each other's liveness, and corroborated fault reports reduce a score. A score is built from four components, each computed by every node from chain data:

Component	Weight	What it measures
Uptime	40%	Hourly windows with chain-recorded life, 24 windows per period
Participation	25%	Blocks produced (60%) and attested peer coverage (40%)
Reliability	20%	Starts at full marks; each corroborated fault or invalid attestation removes 20%
Responsiveness	15%	Share of heartbeats reporting a height within 12 blocks of the network

11.3 Settlement

- A reward period is 24 hours of protocol time. The first block after a period closes settles it, so payouts happen with nobody online.
- A node needs at least 50% verified uptime and a score of at least 10% to receive a share.
- One node may take at most 5% of a period's pool, never less than an equal split among eligible nodes, so a small network is paid in full.
- A node needs attestations from $\min(2, n-1)$ distinct peers before its liveness counts as independently verified.
- Allocations below 0.000001 OBS carry to the next period.
- Admission is rate limited, and a new node is schedulable only after 64 blocks.

The platform's Node Runners page shows the revenue totals, the exact 90/10 split, the Reward Pool and every settled period with its per-node payouts. An operator can look up a node id and see the evidence the network recorded and the score it produces.



WHAT YOU CAN USE TODAY

12 The Obsidian product suite

Five products give people access to the network. All five run against the same chain, sign transactions with the same code and show only what a node reported.

Product	Where	For
Main web page	mineobs.network	Mine OBS, wallet, explorer, ONS and Marketplace, node runners, developer tools and the account system
Web App	app.mineobs.network	The full Obsidian client in a browser or on a phone
Wallet	wallet.mineobs.network	A focused non-custodial wallet
Browser Extension	Chrome, Edge, Firefox	The Obsidian client in the toolbar, with claim alerts
Obsidian Node	Windows, macOS, Linux	Running and operating a node, with a local dashboard
API	api.mineobs.network	The official network API (see section 14)

12.1 The Platform

The Platform is the public home of the network and the account system behind mining. Its sites cover Mine OBS, Wallet, Explorer, ONS with the Marketplace, Node Runners and Developer resources. It holds accounts and issues mining certificates; it never holds keys, funds, prices or consensus.

- **Account.** Sign up with a Gmail address, a password, an invitation and multi-factor authentication. Recovery codes are shown once. Each account can invite up to five new accounts, and a new account starts with 0 OBS.
- **Mine OBS.** The claim screen reads eligibility from the node and shows the countdown from protocol time.
- **Explorer.** Blocks, transactions, claims, names and the Mining Pool, read from real nodes.
- **ONS and Marketplace.** Search, register, manage, list, buy and make offers on .obs names with a review screen before every signature.
- **Node Runners.** The reward economics and per-node evidence described in section 11.

12.2 The Web App

The Web App uses the Platform's account system and the node's ledger. An account made in one signs in on the other, with the same password, MFA secret, recovery codes and invitations. A claim from either product is the same claim on the ledger, and the chain prevents a double claim.

Screen	What it does
Sign up / sign in / recover	One account for the whole ecosystem
Home	Account at a glance: recovery codes, MFA, linked address
Mine	Claim once per four hours; the claim is signed in the browser
Wallet	Create a wallet (24 words, proven before sealing) or import one; Receive shows your address as a QR code; Send can scan a QR into the recipient; history
Explorer	Overview, blocks, claims, names, the reward pool, search



Screen	What it does
ONS	Search, register, manage and list .obs names
API	Run any public read the gateway allows and see the node's own answer
Menu	MFA enrolment, invitations, link your wallet once by signing a challenge, claim alerts, sign out

12.3 The Wallet

The Wallet is a non-custodial browser wallet. It has no chain, node, key format, address format or fee rule of its own: it is a client of the network's own code. It serves one network, named when it starts, and verifies the network's identity before any signature.

Screen	What it does
Create	Password of at least 12 characters, then 24 real BIP-39 words from secure randomness. Nothing is stored until the backup is confirmed.
Import	Recovery phrase (12 or 24 words), a private key checked on the device, or an encrypted backup made by this wallet.
Home	Exact balance from the node, recent transactions, the network card
Receive	Your address as a QR code the app's own decoder reads back; copy and share
Send	Recipient and amount validation, QR scan or photo, live protocol fee
Review	Recipient, amount, fee, total, balance after and network, then the password authorises the signature
Transaction status	Prepared, signed, accepted by the node, in the mempool, in a block with confirmations
Activity and Backup	The node's indexed history; encrypted backup; the phrase after the password, hidden after a minute

12.4 The Browser Extension

The Extension is a Manifest V3 extension for Chrome, Edge and Firefox. It carries the Web App's screens (Home, Mine, Wallet, Explorer, ONS, API and Menu) plus a Node screen with connection state, height, sync, mempool, protocol and genesis id. Every transaction is built, signed and submitted by the same core code the Web App uses. Claim alerts use the browser's notification API, so a due claim can notify you with no page open, and the node, not the extension, decides whether a claim is allowed.

Its permissions are minimal: storage, alarms and notifications. Access to a server is requested per server on a click. It injects no content scripts and no dapp provider, and it runs under a strict content security policy with no eval and no remote code.

12.5 Obsidian Node

Obsidian Node is the desktop application for running and operating a node. It starts the real Obsidian Core as a child process, reads its RPC on the local machine and signs with the core's own key and transaction code. It supports all four networks, each with its own data directory, wallet and settings.

Its screens include Overview, Node, Node Rewards, Wallet, Transactions, Explorer, Logs and Diagnostics, Settings and Help. Wallets are encrypted vaults; the recovery phrase is shown once at creation; the renderer runs sandboxed with context isolation and a strict content security policy; and



logs and diagnostics redact phrases, keys, passphrases and tokens.

12.6 Obsidian Core and the command line

Obsidian Core is the reference node. It needs Node.js 20.10 or newer, one CPU core, 512 MB of memory and a few hundred megabytes of disk to follow the chain. A single command starts a node; the network must always be named, because no default network exists anywhere.

COMMAND LINE

```
node dist/index.js start --network mainnet           # run a node
node dist/index.js start --network mainnet --no-mine # follow and validate only
node dist/index.js health --network mainnet          # what the node sees
node dist/index.js wallet new --network mainnet      # a wallet, offline
node dist/index.js keygen --network mainnet          # the node identity key
node dist/index.js register obs1... --network mainnet # bind the reward wallet
```

12.7 The explorer and the edge layer

The explorer shows blocks, transactions and names read directly from nodes, and it is built on two rules. **It never exposes wallet balances.** Addresses next to a transaction are masked, enough to correlate two transactions by the same author and never enough to look up holdings. **Transaction ids are not wallet addresses.** Ids use their own format, so an address can never be mistaken for a hash.

A Cloudflare worker serves and caches the sites at the edge. It forwards signed transactions untouched, never caches a submission, never invents a response when the origin is unreachable, and holds no key. The chain does not depend on it.



HOW THE PROTOCOL AND THE PRODUCTS PROTECT USERS

13 Security design

13.1 In the protocol

- **Replay protection.** Each transaction binds a sender and nonce, the chain ID and an expiry; a signature from one network is useless on another.
- **Domain-separated signatures.** Every signed message type has its own tag.
- **Deterministic rules.** Consensus parameters are compile-time constants of the protocol version, not operator settings. Changing one requires a documented protocol upgrade.
- **Verified state.** Every block commits to a state root and an events root that every node checks. The supply invariant is checked on every block.
- **Strict canonical encoding.** One transaction has one byte representation; unknown types are rejected; addresses must be in canonical lowercase.
- **Admission limits.** The mempool holds at most 10,000 transactions, 32 MiB and 64 per sender, and requires the account's next nonce, so each sender has one transaction in flight.
- **Equivocation ban.** Two signed blocks on one parent permanently ban a producer, provable by anyone from the two signatures.
- **Network isolation.** Genesis id, parameters hash, chain ID and address prefix keep the four networks apart.

13.2 In the products

- **Non-custodial keys.** Keys never leave the device. Vaults are PBKDF2-SHA256 (600,000 iterations) with AES-256-GCM; the vault envelope is validated before any key derivation.
- **Authenticated accounts.** Passwords are stored as scrypt hashes; sign-in supports TOTP multi-factor authentication and one-time recovery codes; repeated failures are rate limited per account and address.
- **Review before signing.** Every wallet action shows the recipient in full, the amount, the fee and the network, and signs only after the passphrase is entered.
- **Privacy by design.** The explorer masks addresses and answers no balance query; a balance is available only for an address the caller already knows.
- **Request limits on the node RPC.** 240 requests per minute per client address, a 512 KiB body limit and tight header and request timeouts.
- **Strict browser policies.** The apps ship security headers and content security policies; the extension and desktop app disallow eval, inline script and remote code.
- **Display integrity.** Amounts are formatted from exact integers (never floating point) and addresses are never altered for display, so a copied or signed address is always the stored one.



INTEGRATING WITH THE NETWORK

14 Developer reference

The node serves an HTTP API and a JSON-RPC endpoint. Amounts are integers of **seals**: 1 OBS = 10^{18} seals. Reads are unauthenticated. The official API is at <https://api.mineobs.network>; for the genesis id and current network details visit <https://mineobs.network>.

14.1 Read routes

Route	Returns
GET /health, /status	Network, chain ID, genesis id, parameters hash, height, supply, syncing state
GET /params	Protocol parameters including the parameters hash
GET /blocks?limit=, /block/<height hash>	Recent blocks; one full block
GET /tx/<txid>	A transaction with inclusion height
GET /address/<address>?limit=	Activity, mining claims and protocol credits for an address (never a balance)
GET /wallet/<address>/next-nonce	The next nonce to sign with
POST /wallet/balance	Spendable, locked and total balance of an address you already know
GET /mining/schedule, /mining/status?address=	The reward schedule; eligibility for one address
GET /names/<name>, /names	ONS records and a paged list
GET /market, /market/listings, /market/name/<name>	Marketplace summary, listings and the status of one name
GET /pot, /revenue, /supply	Proof of Time state; ONS revenue and split; supply and invariant
GET /nodes/registry, /nodes/rewards, /nodes/status/<id>	Node runners, reward pool and per-node status

14.2 Write routes

Route	Body	Purpose
POST /tx/submit	{ "tx": "<signed hex>" }	Validate and admit to the mempool; returns the transaction id
POST /tx/simulate	{ "tx": "<signed hex>" }	Run against current state without committing
POST /tx/encode	{ "type", "payload" }	Canonical body encoding and its inverse
POST /tx/gas	{ "amountObs": "12.5" }	Deterministic gas quote
POST /wallet/quote	{ "address" }	Balance, next nonce and gas estimate for planning

**EXAMPLES AGAINST A LOCAL NODE (MAINNET RPC PORT 8630)**

```
# next nonce for a sender, then the gas for 12.5 OBS
curl -s http://127.0.0.1:8630/wallet/<address>/next-nonce
curl -s -X POST http://127.0.0.1:8630/tx/gas \
  -H 'content-type: application/json' -d '{"amountObs":"12.5"}'

# submit a signed transaction (hex built and signed locally)
curl -s -X POST http://127.0.0.1:8630/tx/submit \
  -H 'content-type: application/json' -d '{"tx":"<signed hex>"}'
```

14.3 Signing a transaction

A client builds the canonical body with the shared encoder, fills the envelope, hashes it under the signing domain and signs it with secp256k1. The transaction id is known before broadcast. A wallet confirms one transaction before sending the next from the same sender, because the mempool admits only the account's next nonce. The reference signing code is in `obsidian-core/src/transactions` and in the Web App's signing bundle.

Errors use stable codes such as `ERR_BAD_NONCE`, `ERR_WRONG_CHAIN_ID`, `ERR_MINING_TOO_SOON`, `ERR_NAME_LISTED` and `ERR_MARKET_PRICE_MISMATCH`, so clients can react to a rejection precisely.



REFERENCE

Appendix A Protocol parameters

Every value below is a consensus constant of protocol 2.0.2. A node reports the parameters hash in GET /params; for this release it is 12211595855ba405b3d70f6161ef9371 on every network.

Group	Parameter	Value
Supply	Maximum supply	21,000,000 OBS
	Genesis allocation	100,000 OBS, once
	Decimals / smallest unit	18 / 1 seal = 10^{-18} OBS
Blocks	Target interval	5 seconds
	Maximum size / transactions	2 MiB / 2,000
	Median-time-past window	11 blocks
	Future drift allowed	10 seconds
	Maximum reorganisation depth	256 blocks
	PoT Difficulty window	720 blocks
Transactions	Maximum size / memo / events	16 KiB / 256 bytes / 64
	Expiry window	240 blocks
	Minimum transfer	0.000001 OBS
Gas	Rate / minimum / maximum	0.02% / 0.000001 OBS / 0.01 OBS
Mining	Claim interval / claims per cycle	4 hours / 6 per 24 hours
	Reward per claim at genesis participation	0.00016666666666666666 OBS
	Reduction / step / daily floor	0.5% / 100,000 active miners / 0.0002 OBS
	Active-miner window	30 days
Mining gate	Certificate validity / clock skew	15 minutes / 60 seconds
	Maximum issuer keys / changes per update	8 / 8
Producers	Maturity / liveness window	64 blocks / 2 days
	Misses before suspension / first suspension / ceiling	3 / 10 minutes / 24 hours
	Registrations per block / candidates per height	2 / 8
ONS	Registration and renewal fee	0.05 OBS
	Term / grace / name length	365 days / 30 days / 3 to 63
	Transfer offer / recovery delay	7 days / 7 days
Marketplace	Minimum price and offer	0.001 OBS
	Offer lifetime / retention of settled offers	7 days / 90 days
	Pending offers per name / per maker	16 / 32
	Commission	None
Node rewards	Revenue split	90% node pool / 10% treasury
	Period / uptime window	24 hours / 1 hour
	Minimum uptime / minimum score	50% / 10%
	Per-node ceiling / minimum payout	5% of the pool / 0.000001 OBS
	Wallet change delay / reclaim delay	2 periods / 1 period
Accounts	Invitations per account / starting balance	5 / 0 OBS



REFERENCE

Appendix B Glossary

Term	Meaning
OBS	Obsidian Seal Coin, the native coin of the Obsidian chain
Seal	The smallest unit of OBS: 10^{-18} OBS
Proof of Time (PoT)	Consensus in which block production follows a deterministic schedule gated by protocol time
Protocol time	The timestamps committed in block headers, summarised by the median of the last 11 blocks
PoT Weight	Fixed fork-choice weight: one unit per valid block
PoT Difficulty	A published measurement of how block spacing compares with the 5-second target
Time-Rate	Verified blocks and transactions per minute of protocol time
Producer	A registered node scheduled to produce a block in a given slot
Slot / round	A 5-second interval of protocol time / the number of skipped slots before a block
Mining claim	A signed transaction that claims the protocol reward for a wallet
Gate certificate	A short-lived signature authorising one claim for one wallet from a signed-in account
Mining Pool	Protocol state that holds gas and pays mining rewards before new issuance
ONS	Obsidian Name Service: .obs names as chain state
Treasury wallet	The wallet that made the first valid mining claim on a chain; receives 10% of ONS revenue
Node Runner Reward Pool	Protocol state that holds 90% of ONS revenue until each period settles
Escrow	OBS locked by consensus behind a pending marketplace offer



REFERENCE

Appendix C Document information

Item	Detail
Title	Obsidian Network: White Paper
Subtitle	The Proof of Time Blockchain
Coin	Obsidian Seal Coin (OBS)
Version	1.1, 24 October 2026
Describes	Obsidian Core 2.0.4, protocol 2.0.2, Mainnet chain ID 7777
Publisher	Emolux Labs
Main web page	https://mineobs.network
Web App	https://app.mineobs.network
Wallet web app	https://wallet.mineobs.network
API	https://api.mineobs.network
Genesis id and network details	Visit https://mineobs.network
Source code	github.com/EmoluxLabs/Obsidian-Network · Apache-2.0
Contact	emoluxlabs@gmail.com

Version history

Version	Date	Notes
1.1	24 October 2026	Added the official links, including the API; genesis id and live network details point to the main web page
1.0	24 October 2026	First edition of the Obsidian Network White Paper