Automated Contract Ledgers for AI Agents

Next-Gen Smart Payments

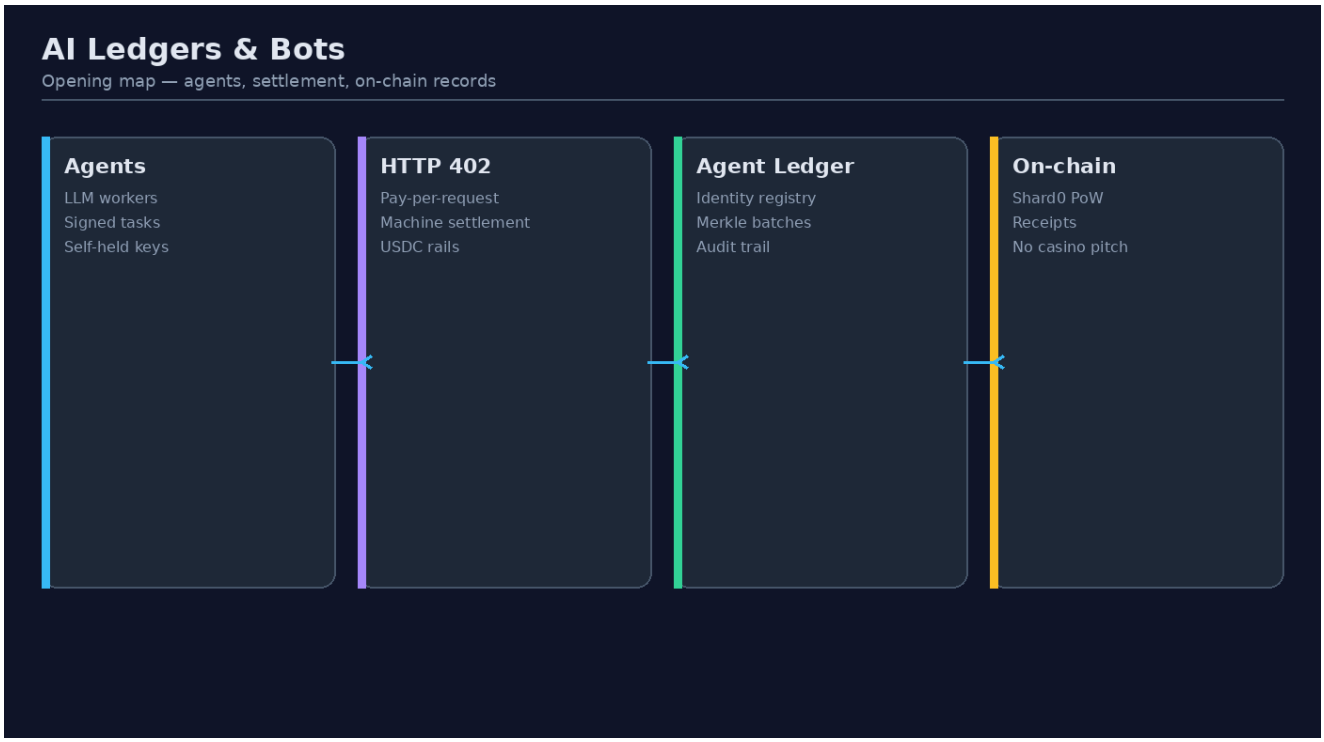
Ying Willam Zhang · SCDO Laboratory · 2026 · v2



Disclaimer: This primer is for education only — not investment, legal, or tax advice.

1. Bitcoin (starting from AI ledgers and bots)

> SCDO Lab · *Automated Contract Ledgers for AI Agents* · *Next-Gen Smart Payments* · v2
> Chinese: `01-bitcoin.md`
> We open with **AI agent contract ledgers** and **bot per-call commerce**: why machines need verifiable books and keys—then return to Bitcoin as the peer-cash / PoW origin.
> Not investment advice.



Opening: AI Ledger, bots, then Bitcoin

When an **AI agent** calls APIs, pays stablecoins, or writes work digests to a ledger, it needs:

1. **Self-held keys** (who signs is accountable);
2. **Checkable trails** (Agent Ledger / contract events);
3. **Micropayments** (HTTP 402 / x402, ~1¢ intuition).

None of that appeared from nowhere—it stands on Bitcoin’s tradition of **public ledgers** + **private-key signatures** + **PoW anti-forgery**. This chapter explains Bitcoin in plain language as the foundation for later SCDO five-chain, mining, bots, and smart-payments chapters.

SCDO Lab · Web3 Primer for Newcomers v2 · English edition
华语版 / Chinese: `01-bitcoin.md`
This chapter explains in plain English: what Bitcoin is, why the internet alone cannot move “unique money,” how nodes and mining cooperate, keys and wallets, farms and pools.
Not investment advice. Price stories are history, not recommendations.

Opening: say it in human words

Bitcoin (BTC) is a **digital money system with no company boss and no central clearing bank**. Many people worldwide run the same open-source software (nodes) and jointly maintain a public transaction ledger—the **blockchain** people talk about is, in Bitcoin’s case, that continuously extended chain of ledger pages.

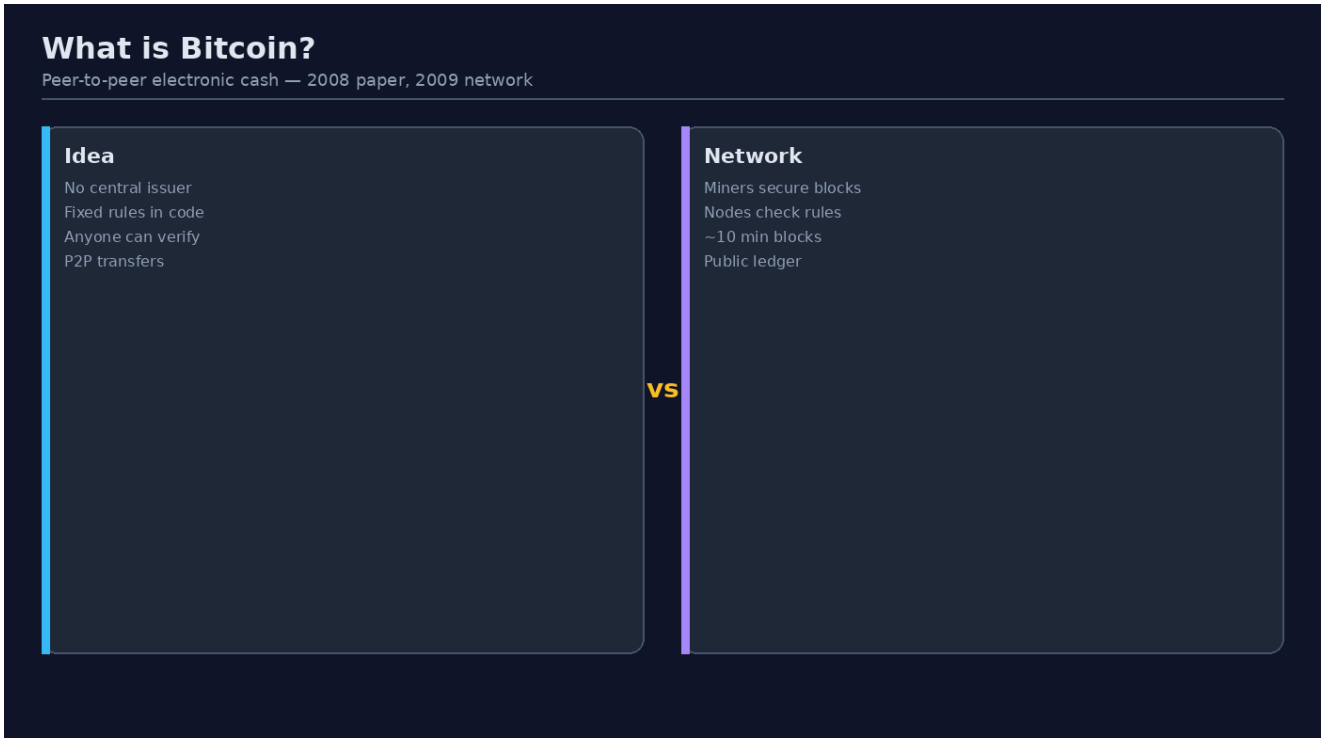
Think of three stacked pieces:

1. **A rule set** (protocol and software): who may write the ledger, how money moves, how supply is issued.
2. **A globally shared ledger**: transfers are public and checkable, without binding your legal name.
3. **A bookkeeping reward**: whoever packages a valid new block under the rules may earn newly issued bitcoin plus fees—that is mining incentive.

Supply has a rule-bound ceiling (commonly about **21 million** coins). New coins enter circulation on a schedule through mining; roughly every four years a “halving” cuts the block subsidy. No central bank can mint more by decree; changing the rules means persuading enough network participants—which is extremely hard in practice.

Five blocks below: double-spending, how it works, why it stays special, consensus basics, keys/wallets, farms/pools.

What is Bitcoin?



In 2008 a white paper under the name Satoshi Nakamoto proposed peer-to-peer electronic cash; the network launched in 2009. Over more than a decade it saw booms, crashes, fork fights, and regulatory storms—and became many people’s first blockchain lesson.

The first question for a newcomer is not “will it moon,” but:

> **The internet already moves photos and files. Why did moving “money” used to require WeChat Pay, Alipay, or a bank as middleman—and what puzzle does Bitcoin claim to complete?**

The internet’s “missing piece”? — Double-spending

Sending a photo online is easy: after you send it, your copy usually remains. You sent a **copy**.

Money cannot work that way. If I “send” you 100 units and can still spend the same 100, two people are spending one pot—that is **double-spending**.

Pure copy-paste internet transfer cannot solve double-spending. In real life we use **intermediated ledgers**:

- When you send 100 via a payment app, the company updates its central database: my balance −100, yours +100.
- Bank wires and card networks are similar: a trusted institution (or set of them) has the final say.

Intermediaries are useful—and costly: onboarding, compliance checks, freezes, outages, fees, data on someone else’s servers. You trust an **institution**, not math and public rules themselves.

Bitcoin’s aim: without depending on a single giant bookkeeper, make “I pay you 1 BTC” eventually mean the whole network agrees **I have less and you have more**, and history is hard to rewrite quietly. The ledger tech underneath is often called a building block of a “value internet.” The slogan is a bit marketing, but the direction is right: the internet shipped information transfer; Bitcoin ships **scarce digital value transfer**.

Two boundaries:

- Bitcoin is not “fully anonymous cash.” The ledger is public; addresses are like pen names. If an address ties to your identity, trails can be analyzed. More accurate term: **pseudonymous**.
- “No intermediary” means **settlement and bookkeeping need not go through one payment company**; you may still buy via exchanges or custodial wallets—those layers have their own risks.

How does Bitcoin work? — A public game

Picture a **voluntary large-scale collaboration game** (rules live in software):

1. **Anyone can join or leave**: with a networked computer you can download node software and store (or verify) the ledger; quit when you want.
2. **Writing blocks is competitive and costly**: roughly every ten minutes the network produces a new block. Miners race on hard-to-guess, easy-to-check computations—**PoW (Proof-of-Work)**. More hashpower raises short-term odds; nobody is guaranteed the next second.
3. **Transfers are broadcast**: your transaction propagates; miners select txs into blocks. Nodes check signatures, unspent funds, and double-spends.
4. **Changing rules or history is extremely hard**: rewriting confirmed history or forcing issuance changes needs overwhelming hashpower and social consensus; most participants reject invalid branches.

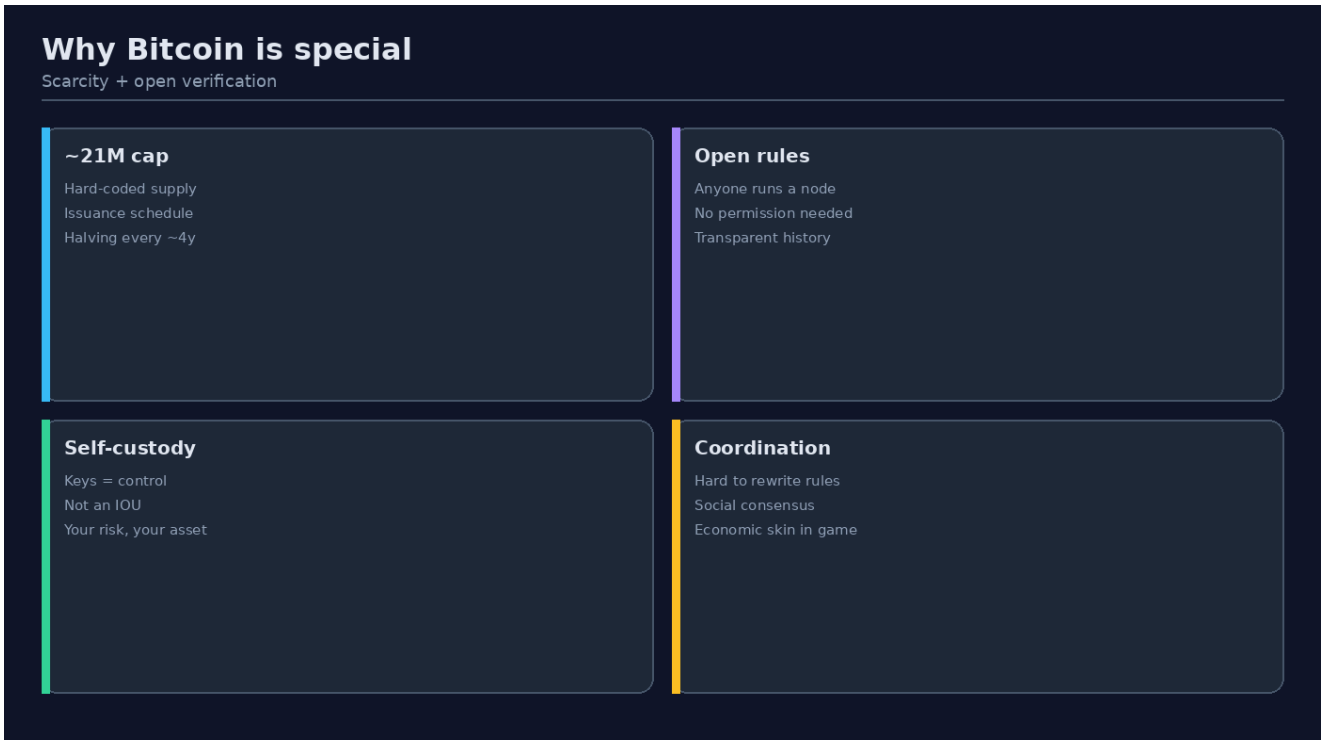
Why burn electricity forever? Successful blocks earn a **block subsidy** (new coins) and user **fees**. Incentives make strangers secure the network: attacking is often too expensive; following rules can pay—that is Bitcoin welding **cryptography** + **economics**.

Versus a payment app in one line:

- App: one company edits two balances.
- Bitcoin: thousands of machines write the change into a shared ledger under common rules, with layered confirmation.

Versus “game points”: points are a company database row the vendor can edit or shut down; bitcoin balances are unspent outputs on a public ledger controlled by your private key—there is no HQ support that can one-click restore them.

Why Bitcoin is still special



(The original manuscript pasted long institutional report translations here. v2 **does not reprint**. Below is our short judgment frame.)

Thousands of tokens exist. Why do many people still keep Bitcoin in its own column? Not because it "only goes up," but because its design trade-offs differ from many other chains.

1. Network and coin are bound together

"Bitcoin" means both the **network** (the protocol people run) and the **coin** (the native unit). You can copy the code and start another chain—that is **another network and another coin**, not automatically "canonical Bitcoin." You also cannot losslessly copy BTC onto arbitrary other chains and still call it the same BTC (bridges are a different trust model; later chapters).

2. More like a digital-age monetary commodity than company equity

No dividends, no CEO earnings call. What people emphasize is transferability, divisibility, verifiable scarcity, and hard unilateral inflation. It is more often discussed as a candidate **store of value / settlement asset** than as "the next unicorn stock." That does not guarantee it wins—it means the **evaluation frame differs**: do not force PE-style valuation onto it.

3. Scarcity rules are hard to change because of social consensus + incentives, not only a line of code



The ~21M cap is in the rules; code can theoretically change. What makes the cap “hard” is holders, miners, exchanges, wallets, and node operators—most refuse dilution. Changing the cap fights their own economic interest; coordination cost is huge. Newcomers should learn: **security comes from cryptography and from people unwilling to collude to rewrite the rules.**

4. “Improving Bitcoin” is usually a different trade-off, not a free upgrade

Later public chains often push speed, smart contracts, and fees harder—usually paying in **decentralization, security, or simplicity**. Bitcoin stays conservative: limited scripting, slow upgrades, in exchange for long-run censorship resistance and “few knives.” Other projects can serve other needs (contracts, games, DeFi); they need not annihilate each other. If your goal is “the most conservative digital scarce asset,” your scorecard should differ.

5. By 2026 the outside world also changed

Institutional rails (e.g. spot ETFs in some markets), custody, and compliant trading change **how ordinary people touch BTC**, not the ledger’s own rules. Easier on-ramps do not mean coins on an exchange equal “you control Bitcoin”—custody risk is separate. On scaling, the base layer stays restrained; smaller high-frequency needs lean on Lightning-style layer-two ideas (Chapter 4 develops the Ethereum side; Bitcoin is similar: L1 for stability, L2 for speed).

One line: Bitcoin is special because it minimizes “who decides” and prioritizes scarcity and security; other chains are special because they prioritize programmability and performance. Which path depends on the problem you solve.

Consensus: PoW, PoS, DPoS

Newcomers keep hitting these acronyms. Do not fear the English: **consensus = the rules by which mutually distrusting parties decide who writes the next ledger page and whether it is valid.**

Family vote on a trip: majority rules—that is one consensus. Blockchains must bookkeep globally with possible adversaries, so rules are harder and often bind real rewards and penalties.

A beginner triad (more variants exist; these three unlock the news):

PoW (Proof-of-Work)

- **Who writes:** race on hashpower; first valid solution roughly wins the block.
- **Pros:** relatively open entry; attacks cost real hardware and electricity; Bitcoin uses PoW.
- **Cons:** energy and hardware arms races; home laptops no longer mine BTC economically.
- **Plain:** whoever keeps paying verifiable “labor” (compute) gets higher odds of bookkeeping rewards.

PoS (Proof-of-Stake)

- **Who writes:** stake coins as collateral; selection/rotation under rules; misbehavior may be slashed.
- **Pros:** usually far less electricity; confirmations can be designed faster.
- **Cons:** “rich get more say” concerns; staking, liquidity, and penalties are more complex.
- **2026 note: Ethereum completed The Merge in 2022; mainnet is PoS, not PoW mining.** Old “mine ETH with GPUs” articles are history. Bitcoin remains PoW; both paths coexist.

DPoS (Delegated Proof-of-Stake)

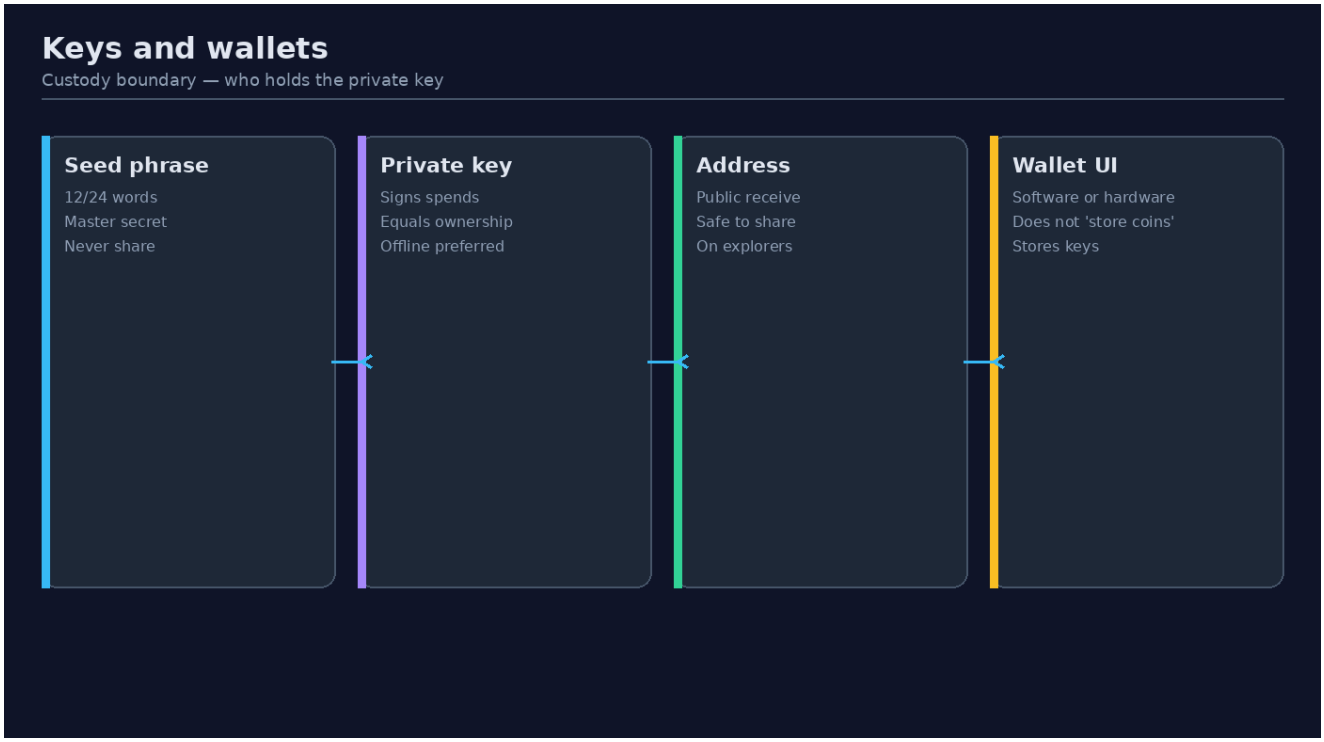
- **Who writes:** like shareholders electing a board—token holders vote a small set of producers.
- **Pros:** fewer nodes, faster coordination, often higher throughput.
- **Cons:** more like an elected committee; usually weaker decentralization than large PoW/PoS; watch vote-buying and oligopoly.

Wrap-up

No perfect consensus—only trade-offs among security, decentralization, and speed (a simplified “impossible triangle”). When reading a project ask: who may propose blocks?

what does attack cost? can ordinary users run verifying nodes? Bitcoin’ s choice is clear—**favor security and decentralization; accept modest base-layer throughput.**

Private keys, accounts, and wallets



Three things closest to users: **private key, address (account), wallet software**. Intuition first; few formulas.

Public-key crypto (intuition)

Modern cryptography can generate a key pair:

- **Private key:** your secret, used to **sign** (“I authorize this spend”).
- **Public key:** derived from the private key; others **verify** signatures but cannot easily reverse to the private key.

In Bitcoin: the private key controls coins; the network checks math without you broadcasting the private key.

What is a private key?

Essentially a **random number from a vast space** (often encoded as a character string). Whoever holds it can spend the corresponding coins.

Therefore:

- **Leak ≈ lose coins**, often irreversibly.

- **Lose with no backup ≈ coins still on-chain, forever unspendable by you**.
- Do not screenshot to albums, send in chat apps, or type into unknown "custody" pages.

What is a Bitcoin account / address?

Daily "Bitcoin account" usually means an **address**: characters derived from hashing the public key (formats differ). Others send to your address; the chain records payments among addresses.

More precisely Bitcoin uses UTXO: your "balance" is the sum of **unspent outputs**, not a single bank-style field. Beginner line: **addresses receive; private keys authorize spends.**

What is a wallet?

A wallet mainly **holds keys, shows balances, builds and signs transactions**.
Coins are on-chain; the wallet holds keys—not a pile of coin files.

Type	Trait	Fit
Hot wallet	Online phone/browser/desktop apps	Small daily amounts
Cold wallet	Keys offline (hardware, offline signing)	Larger long-term holds
Custodial wallet	Exchange etc. holds for you	Convenience; you take platform risk

First setup often shows a **seed phrase**: words that derive keys. The seed is a master key—anyone with it can empty the wallet. Write offline, multiple backups, never type it into networked forms.

You sign locally, then broadcast; after miner inclusion and confirmations, the recipient may treat it as received.

Mining farms and pools

Mining evolved from "laptop hobby" into a specialized industry. Three words often get mixed:

1. Mining machines

Specialized compute. Evolution line to remember:

CPU → GPU → FPGA → ASIC (application-specific chips)

Bitcoin today is dominated by ASICs: they mainly do mining hashes, far more efficient than PCs. Home PCs mining BTC is generally uneconomic.

2. Farms

Places that pack machines with power, cooling, and ops. Why concentrate?

- **Electricity** is often the largest cost—sites chase cheap stable power;
- Professional ops beat basement hobbyist reliability;
- Noise, heat, and compliance are hard at home.

Farms are a **physical and ops** business: host machines, keep uptime.

3. Pools

Mining is like buying lottery tickets: small hashpower may wait long for a "winning ticket" (a block), so income is lumpy. Miners **logically pool** hashpower:

- The pool assigns work;
- Contributors share rewards by valid shares (payout schemes vary);
- Small miners get smoother cash flow without running full solo block workflows.

Plain: **the farm is the factory; the pool is the project lead.**
Machines can sit in farm A while hashpower points to pool B—physical hosting and logical payouts can be chosen separately.

Large pools raise centralization concerns. Healthy networks prefer dispersed pools and migratable miners. If you do not mine, remember: **a large share of network security budget comes from hashpower competing for rewards.**

Chapter wrap

Five takeaways:

1. The internet copies information well; **money cannot be copied**, or you get double-spending.
2. Centralized payments rely on institutional ledgers; Bitcoin relies on a **globally replicated rule ledger** + **incentives**.
3. Bitcoin is deliberately conservative on scarcity and censorship resistance; **it is a different exam from other public chains**.
4. **Private key** = **ownership**; a wallet is a keychain, not the bank vault itself.
5. Machines/farms/pools industrialize PoW security; you can still run a verifying node without mining.

Next chapter leaves "one money system" for the larger **Web3** basket: when accounts, contracts, identity, and apps move on-chain, what newcomers actually meet—and which parts are foam versus usable skill.

(End of Chapter 1 · v2 English edition · SCDO Lab)

2. Web3

> SCDO Lab · Web3 Primer for Newcomers v2 · English illustrated edition
> Full bilingual book: Chinese `/learn/web3/` · English `/learn/web3/en/`
> 华语版 / Chinese: `02-web3.md` · Figures: `assets/ch02/`
> Chapter 1 covered Bitcoin as one digital-cash system. This chapter zooms out: what people mean by **Web3**, how it differs from everyday apps, what actually ships in 2026, and how a newcomer can start safely.
> **Not investment advice. Not a hype manifesto.**

Opening: turn the foam down

Talk about Web3 and you usually hear three voices:

1. **Cure-all**: it will replace Big Tech, banks, and governments—tomorrow.
2. **All scam**: anything with a chain or a token is a rug.
3. **Jargon maze**: DAO, DApp, TVL, account abstraction… three minutes and people close the tab.

We pick a fourth: **engineering and product**.
Web3 is not a company and not a single chain. It is an attempt to use **blockchains** + **cryptographic keys** + **open protocols** so that parts of the internet move from "you must trust this platform" toward "rules are public and critical actions are verifiable." What already works is concrete; what is oversold is also concrete. After this chapter you should tell them apart.

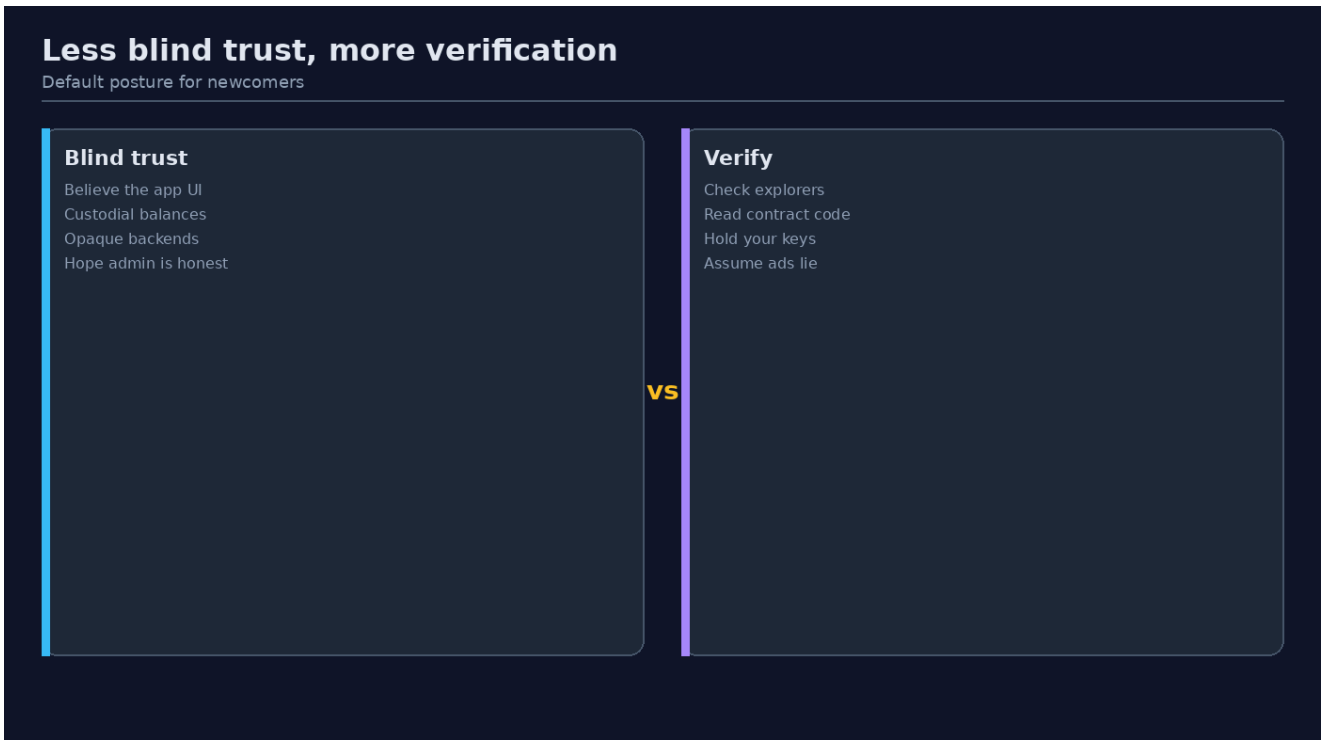
Naming note: people mix **Web3** and **Web 3.0**. Early texts said Web 3.0; spoken English today usually says **Web3**. We use Web3 below. It is not the older "Semantic Web 3.0" idea—if someone says 3.0, ask which lineage they mean.

What is Web3, really?

One-sentence version

Web3 ≈ using blockchain and related tech so users can hold digital assets and credentials themselves, and use online apps with less dependence on a single platform that "has the final say."

Shorter: **less "trust this company," more "you can check."**



This direction was articulated in the mid-2010s by people around early Ethereum (long interview reprints from the old manuscript are **not** copied here). The point is engineering: reduce blind trust in authorities; bake verifiability into protocols. We agree with the direction—and must add a 2026 reality check: **eliminating every intermediary does not exist**. Exchanges, browsers, phone OSes, DNS, and stablecoin issuers are still intermediaries. Web3 changes **which slice** need not trust a single ledger-keeper; it does not put you in a vacuum.

Why bother?

Mainstream internet (Web2) is extremely useful: social, shopping, payments, video—a few giants cover most of daily life. The pain sits on top of that convenience:

- **Accounts live elsewhere**: bans, rule changes, opaque ranking—you leave or accept; taking data with you is hard.
- **Money and points live elsewhere**: skins, balances, creator payouts can vanish if a service shuts down or rewrites terms.
- **High trust, thin evidence**: you believe “the backend will honor the blog post”; individuals rarely audit the database.

Web3 aims at **ownership and control**—signatures prove “I authorized this”; public ledgers prove “this state update happened.” That is not automatically a fair society; it is one more technical option.

Less blind trust, more verification

“Trust” here is a systems word, not advice to distrust friends:

> You must assume a black box is honest, always online, and will not rewrite the rules—without independent evidence.

Bitcoin transfers: the network can check double-spends.
Smart-contract swaps: you (or tools you trust) can read code and on-chain results, not only marketing copy.

If your coins sit on a centralized exchange, you are back to "trust that company's ledger." **The tool can be Web3 while the habit stays Web2.** Newcomers often miss that.

Can it break monopolies?

Vision decks love "break Big Tech lock-in." Honest answer: **weaken lock-in in places; overnight overthrow is unlikely.**

- Cryptography lets ordinary people encrypt and self-custody—harder for a platform to unilaterally erase on-chain balances (custody and legal process are separate).
- Traffic, fiat on/off-ramps, app stores, and licenses remain highly centralized.
- Many "Web3 apps" are still ordinary websites: seize the domain or kill the API and UX dies.

A saner goal: **critical assets and identity should not live only in one company's database**. Companies can still ship great UI without unilaterally wiping your on-chain rights.

What does decentralization mean?

We use a checkable definition:

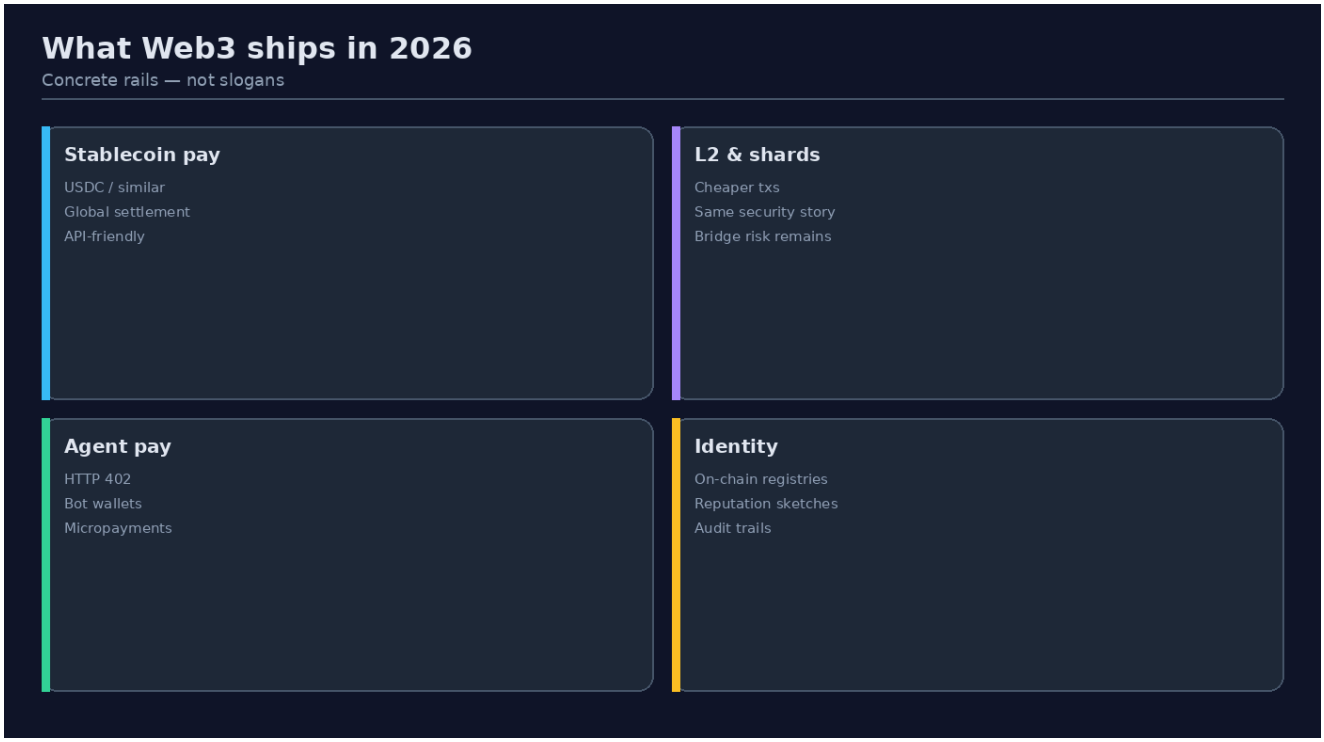
Is there a single point that can halt the system, rewrite history, or move your assets without your consent?



- **Fully decentralized**: almost never exists in pure form.
- **Decentralized enough**: attacks/failures are expensive; alternatives exist; users can run nodes or switch wallets.
- **Decentralization theater**: tokens are on-chain, but upgrades, sequencers, bridges, and oracles are a small club—look closely.

Decentralization ≠ lawlessness. Compliance, tax, and fraud rules still apply in most jurisdictions. Read it as **reducing single-party domination in tech**, not as a free-fire zone.

What does the Web3 world look like in 2026? (no moon math)



You can already touch:

Layer	What you may meet
Assets	BTC, ETH, stablecoins, many tokens; some people access via exchanges or (in some markets) ETFs
Apps	DeFi swap/lend, NFT credentials, game items, on-chain social protocols
Gateways	Browser/mobile wallets, exchange apps, growing smart wallets / account abstraction
Infrastructure	Public chains, Layer 2s, bridges, oracles, indexers

Still hard or unfinished:

- Mass-market UX with almost no seed-phrase learning curve (improving, not done);
- Seamless multi-chain without bridge risk;
- Replacing super-app social and payments habits wholesale with on-chain apps.

Honest 2026 takeaway: Web3 moved from "only miners and geeks" to "real financial and credential infrastructure," but it has **not** fully replaced Web2. Treat it as an **incremental toolbox**, not a religion.

Common myths (clear the mines first)

1. **"On-chain = decentralized and safe."**
   Contracts can have upgrade backdoors; bridges can be multisigs; frontends can phish. A chain guarantees **execution under current rules**, not that rules favor you or that you clicked the real site.

2. **“Token + NFT = successful Web3 startup.”**
   Tokens coordinate incentives—or sell air. Real users, cash flow, or credential demand beat “we launched a coin.”

3. **“Decentralized apps ignore the law.”**
   Builders, issuers, fiat ramps, and marketing still meet finance and consumer rules in many places. Narrative ≠ compliance homework.

4. **“I installed a browser wallet, so I’m Web3.”**
   If funds sit on an exchange, you never open an explorer, and you approve every signature—habits are still Web2. Tools ≠ literacy.

5. **“AI + chain = money printer bot.”**
   In 2026, agents can call APIs and even pay; oversized permissions also steal and fail faster. Automation needs **limits, allowlists, audit logs**—why we talk about Agent Ledger: explain what the machine did before chasing speed.

How should ordinary people participate?

Old titles screamed “the wave is here.” We ask instead: **if you are curious, how do you learn cheaply and safely?**

Web1 → Web2 → Web3 in three lines



Skip the long timeline; remember the user role:

1. **Web1 (mostly read)**: portals and static pages—you are a reader.
2. **Web2 (read/write + platforms)**: you post and trade, but accounts, fans, data, and ranking live on platforms.
3. **Web3 (read/write + own)**: you still use apps, but key assets and some identity can be key-held; rules prefer protocols/contracts.

"User-controlled data/assets" is directionally right **only if** you truly hold keys and understand what you sign.

Why learn at all?

Even without going "all in":

- Digital ownership fights will grow (tickets, memberships, creator splits, game items);
- Institutions already experiment with chains and on-chain assets;
- Scammers borrow Web3 jargon—**ignorance is how you get farmed**.

Not worth it: FOMO "you're late," betting rent money, or letting someone else "operate for you" (that ≈ handing them the keys).

Practical differences vs Web2

Dimension	Typical Web2	Typical Web3
Login	Phone / SSO	Address + signature (email/social login also emerging)
Assets	Platform balance	On-chain tokens / NFTs (or still custodied on exchanges)
Rule changes	Company blog	Often governance or upgrade paths you can observe
Mistakes	Support ticket	Usually no rollback; you own the risk
Transparency	Opaque DB	Public ledger; readability needs tools

The sharpest UX shock is rarely ideology: **undo is hard.** Wrong approvals and wrong addresses usually stay wrong. Chapter 9 covers security.

What can it deliver? (sorted by proven-ness)

Comparatively solid:

- **Programmable payments / DeFi Lego**: swaps, lending, collateral—useful, with liquidation and contract risk.
- **Public credentials (NFTs, etc.)**: tickets, badges, memberships, items that prove holding.
- **Another rail for value transfer**: where lawful, as a complement—not a magic passport.

Early and noisy:

- On-chain social replacing super-apps;

- Metaverse land narratives;
- “Everyone DAOs away the company.”

SCDO Lab, lightly:
What tends to matter is whether **verifiable records and incentives serve real work**—e.g. when an AI agent acts for you, are there auditable ledgers and permission bounds? We call that Agent Ledger thinking: machines may automate; accountability must not blur. Later chapters return to this; no product brochure here.

A seven-day newcomer path (even before depositing much)



1. Separate **self-custody** from **exchange custody**.
2. Install a reputable wallet; try a **tiny** amount you can lose (or an official testnet).
3. Learn to read address, network, and explorer status.
4. Read one signature popup: `approve` can be dangerous.
5. Back up the seed offline; tell no one.
6. Treat “guaranteed profit / insider allocation” DMs as high risk by default.
7. Pick one track—Bitcoin, Ethereum+L2, or security habits—beat chasing twenty airdrops.

Laws differ by jurisdiction. Follow yours. This book offers no evasion tips.

Web3 glossary (compact)

The old manuscript had a hundred overlapping entries. Keep a **post-chapter survival set**; details unfold later.

Infrastructure

- **Blockchain / public chain**: open ledger and execution environment (Ch. 3).
- **Node**: machine that runs software to store or verify the ledger.
- **Consensus**: who proposes blocks and how cheaters are punished (PoW/PoS; Ch. 1).
- **Layer 2 (L2)**: scaling under an L1 security umbrella (Ch. 4).
- **Smart contract**: on-chain program that runs when conditions are met (Ch. 3).
- **Oracle**: feeds off-chain data into contracts—often a trust point.
- **Bridge**: moves assets/messages across chains—historically accident-prone.

Accounts & assets

- **Wallet**: tool that holds keys and creates signatures.
- **Private key / seed phrase**: control itself.
- **Address**: public receive/identity label (pseudonymous).
- **Token**: transferable on-chain unit; fungible (e.g. ERC-20) or NFT (e.g. ERC-721).
- **Stablecoin**: token aiming to track fiat etc.—check reserves and issuer risk.
- **Gas / fees**: payment for inclusion/execution.

Apps & orgs

- **DApp**: frontend + contracts (decentralization varies).
- **DeFi**: decentralized finance apps (Ch. 5).
- **DAO**: on-chain coordination experiments; quality varies wildly.
- **TVL**: value locked—useful, gameable, never enough alone.
- **NFT**: non-fungible credential ≠ "must be a valuable picture."
- **Account abstraction / smart wallets**: login, recovery, sponsored gas (in progress).
- **AI agent**: program that calls tools/contracts; permissions and audit beat chat skill.

Security mantras

- **Allowance**: permission for a contract to move your tokens—over-approval is a theft magnet.
- **Phishing**: fake sites, airdrops, "support" asking you to sign or export seeds.
- **Self-custody vs custody**: you hold keys vs someone else—convenience ↔ risk.

Figure index

File	Role
`assets/ch02/fig01-web1-web2-web3.png`	Web1→2→3 roles
`assets/ch02/fig01-web1-web2-web3-zh.png`	Chinese caption strip
`assets/ch02/fig02-trust-vs-verify.png`	Blind trust vs verify
`assets/ch02/fig02-trust-vs-verify-zh.png`	Chinese caption strip

File	Role
`assets/ch02/fig03-decentralization-spectrum.png`	Decentralization spectrum
`assets/ch02/fig04-web3-2026-ships.png`	2026 shipped vs still hard
`assets/ch02/fig05-newcomer-path.png`	Newcomer 7-day path

Original SCDO Lab diagrams; no third-party brand marks.

Chapter wrap

Five takeaways:

1. Web3’s core promise is **verifiability and self-custody**, not guaranteed riches.
2. Web1 read, Web2 platform read/write, Web3 tries **read/write + ownership**—reality is hybrid.
3. For decentralization, ask **where the single point is**.
4. 2026 has real pieces (assets, DeFi, L2s, credentials) and loud unfinished stories.
5. Newcomer skill #1 is **keys and signatures**; projects come second.

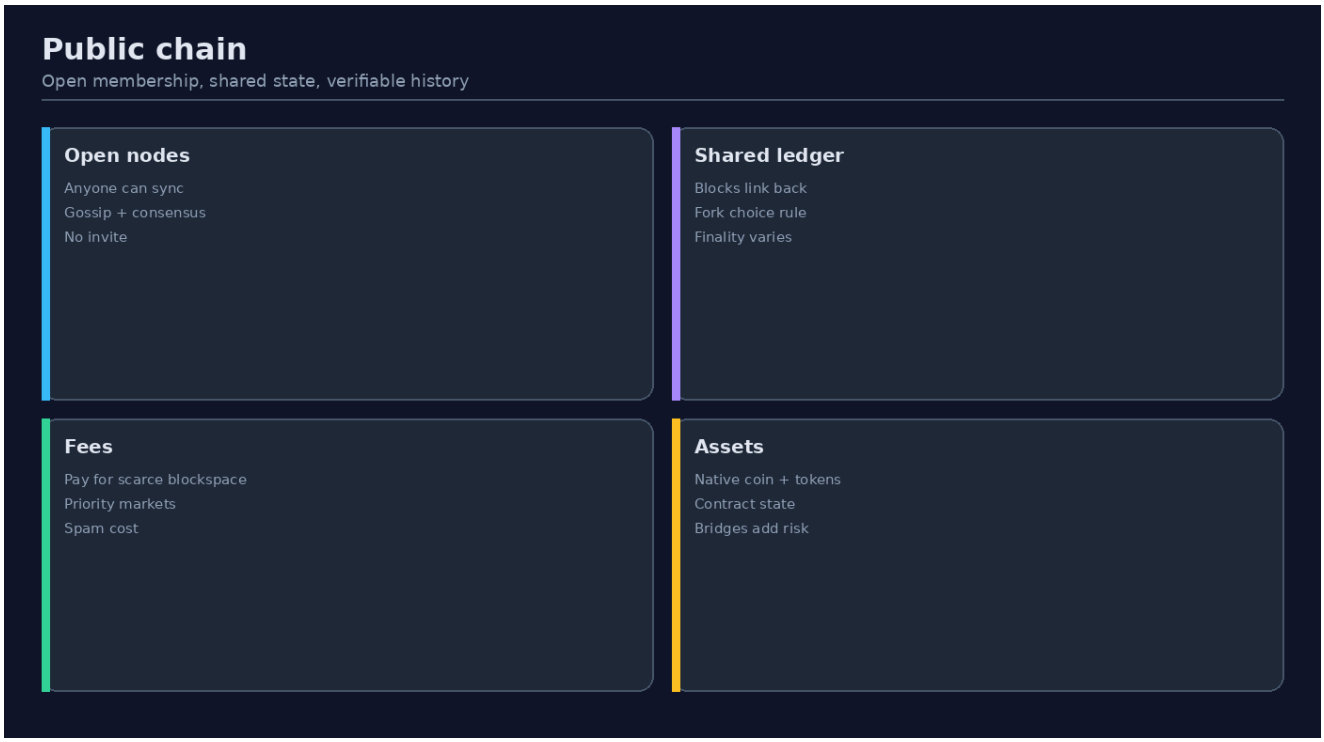
Next: **public chains**—what they are, why Ethereum matters, smart contracts and token standards, and a 2026 map without a hype mega-list.

(End of Chapter 2 · v2 English illustrated edition · SCDO Lab)

3. Public chains

SCDO Lab · Web3 Primer for Newcomers v2 · English edition
华语版 / Chinese: `03-public-chains.md`
Chapter 2 framed Web3 as a “verify + self-custody” toolbox. This chapter covers the **operating system** under it: public blockchains.
Focus: what public chains are, why Ethereum matters, smart contracts and ERC standards, how the difficulty bomb became history, and how to read the 2026 L1 map.
No project promo lists. No multiple narratives. Not investment advice.

What is a public chain?



Plain definition

Public blockchain = an open network: in principle anyone can read the ledger, submit transactions, and run a node (or light client) under the rules—no single company may unilaterally say “today the ledger looks like this.”

Three common shapes:

Type	Who participates	Typical use
Public chain	Open by default	Bitcoin, Ethereum, etc.

Type	Who participates	Typical use
Consortium chain	Invited institutions	Enterprise settlement, supply-chain pilots
Private chain	One organization	Internal experiments, permissioned ledgers

When newcomers say “go on-chain,” they almost always mean a **public chain** (or a Layer2 under a public chain’s security—next chapter).

Is a public chain like an “OS”?

Common metaphor: public chain ≈ Web3 operating system—accounts, verification, transaction storage, and (if present) program execution; wallets, DeFi, games, and social grow on top.

Useful—but do not mythologize:

- Real OSes rarely rewrite consensus weekly; chain upgrades need community coordination and sometimes **hard forks**.
- “Open” ≠ “free and delightful”: Gas, congestion, bridges, and cross-chain risk exist.
- Security comes from cryptography + economic incentives + sufficiently dispersed validators/miners—not from a white-paper sentence.

Checklist of usual properties

When reading a new chain, ask:

1. **Openness:** permissionless read/write? how hard is running a node?
2. **Consensus and security budget:** PoW hashpower or PoS stake? how are attackers punished?
3. **State and execution:** transfers only, or general smart contracts?
4. **Native asset:** what pays fees? are inflation/deflation rules clear?
5. **Client diversity:** one team’s client only? single-point risk?
6. **Ecosystem and exit:** many apps? can users migrate assets if they dislike the rules?

No perfect score. Bitcoin is extreme on simple/conservative/hard-to-change; Ethereum is extreme on programmable ecosystem; many new L1s push speed and fees harder—and often have shorter battle history on decentralization.

What is Ethereum?



From Bitcoin to a “programmable world computer”

Bitcoin showed scarce digital assets can move on a global ledger without a bank.
Its scripting is deliberately weak—more like a single-purpose calculator.

Ethereum’s core idea: a **distributed computer**—nodes jointly maintain state; anyone can deploy programs (smart contracts) that execute under rules. Native coin **ETH** pays Gas and, in the PoS era, is staked.

Teaching shorthand (not an official hierarchy):

- Bitcoin ≈ blockchain 1.0 (digital cash / settlement asset)
- Ethereum ≈ blockchain 2.0 (smart-contract platform)

Much value and innovation still sits on Ethereum and its Layer2s; other L1s compete on performance, languages, modularity, and more.

Vitalik and early story (minimal)

Ethereum was advanced by **Vitalik Buterin** and others: ideas and crowdfunding around 2013–2014; **2015** mainnet launch. Skip gossip. Remember: it aimed at “programmable” from day one—hence the later tree of ICOs, DeFi, NFTs, L2s (good and bad).

Versus Bitcoin (2026)

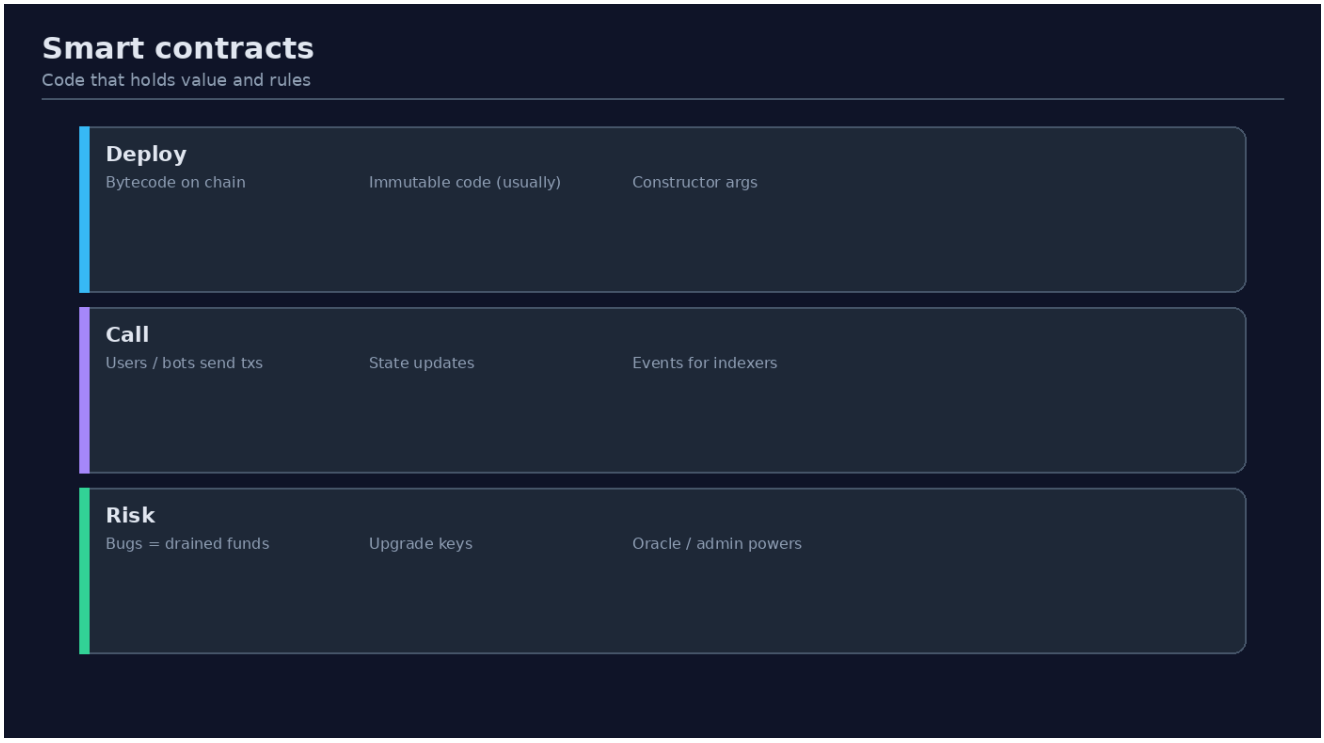
	Bitcoin	Ethereum (2026)
Main role	Robust digital scarcity and settlement	General contract platform + settlement layer
Consensus	Still **PoW** mining	**PoS after The Merge (2022-09)**
Programming	Limited script	EVM contracts (Turing-complete, Gas-bounded)
Accounts	UTXO	Account balances + contract accounts
Fees	BTC fees	ETH Gas (can be high; many users move to L2)
Upgrade pace	Extremely conservative	Roadmap exists; still needs broad consensus

Critical fact: Ethereum completed **The Merge**. Mainnet no longer mines ETH with GPUs/ASICs; validators stake ETH. Treat “Ethereum miners GPU-mine ETH” or “the difficulty bomb is forcing migration now” as outdated.

What is the EVM?

EVM (Ethereum Virtual Machine) executes contract bytecode. Many other chains claim “EVM compatibility”: similar tools/languages (e.g. Solidity), familiar addresses and wallets. Compatibility lowers migration cost; it does **not** automatically equal Ethereum-grade security or decentralization.

What are smart contracts?



Definition

A **smart contract** is an on-chain program: when a transaction triggers it and conditions hold, it updates state by code (transfers, minting, permissions).
“Smart” is not AI—it means **automatic, verifiable execution**. “Contract” does not automatically equal legal force—that depends on jurisdiction and extra arrangements.

Plain examples:

- Crowdfund: release funds only if the goal is met; else refund.
- Swap: exchange token A for B by formula (classic DeFi).
- Credential: mint a unique NFT as a ticket.

Once deployed, code is usually transparent (whether anyone reads it is another matter). Upgrade proxies and admin keys weaken “immutability”—always ask: **who can change the rules?**

Powers and pits

Powers: less dependence on a company manually executing the backend; strangers can coordinate under public rules.
Pits:

- Bugs get exploited (history is long);
- Bad oracle prices blow up systems;
- Wrong allowances hand over the keys;
- Expensive Gas or congestion makes great contracts unusable—why L2s rose.

Smart contracts turn public chains into app platforms—and why security needs its own chapter.

ERC-20 and ERC-721: why standards?



ERC (Ethereum Request for Comments) are community interface standards so wallets, exchanges, and explorers know how to show balances and transfer—without custom software per project.

ERC-20: fungible tokens

- Units are interchangeable: like one dollar bill, or game gold.
- Interfaces cover name, symbol, supply, transfer, allowance (`approve`/`transferFrom`), etc.
- Uses: governance tokens, stablecoins, points, LP receipts, and more.
- **Allowances are powerful and dangerous:** after you approve a contract, a malicious one may drain tokens. Manage allowances; never blind-sign.

ERC-721: non-fungible tokens (NFTs)

- Each tokenId is **unique**: like a title number, a concert seat, a collectible.
- Uses: art/avatars, tickets, memberships, game items, achievements...
- NFT ≠ “must be a valuable picture”; it is an **on-chain unique credential**. Value depends on rights, community, and liquidity—most go to zero or ignore; that distribution is normal.

Quick contrast

	ERC-20	ERC-721
Fungibility	Homogeneous	Each distinct
Balance meaning	Quantity	Which IDs you hold

	ERC-20	ERC-721
Typical pitfalls	Inflation, allowance drains	Fake floors, phishing mints

ERC-1155 mixes fungible and non-fungible in one contract; beginners can hold 20 and 721 first. Other chains have analogous standards under different names.

Appendix: Ethereum’s “difficulty bomb” — finished history

Some old articles still write the bomb in present tense as if The Merge never happened. **Treat this as history class.**

What it was

The **Difficulty Bomb** in PoW-era Ethereum made mining difficulty rise exponentially after certain epochs, slowing blocks until PoW mining became uneconomic. One goal: **force migration to PoS**, reduce endless delay, and lower the appeal of stubbornly staying on the old chain.

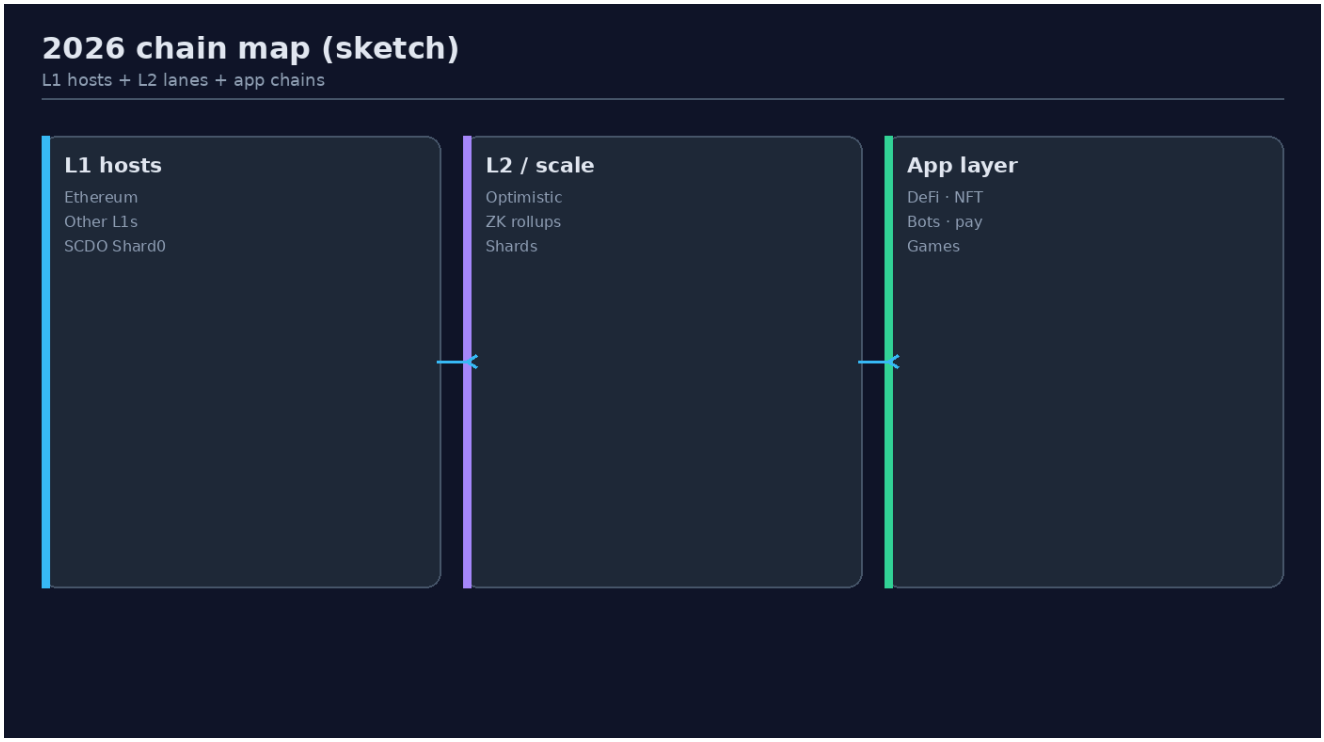
In practice the bomb was **delayed** many times to match the roadmap; hard-fork parameter tweaks were normal—not a sci-fi “detonate and destroy the network” plot.

How it ended

September 2022, The Merge: execution layer joined the beacon chain; consensus became **PoS**. Mainnet PoW mining ended; the bomb’s job as a migration stick was essentially over. PoW chains that remain are **other networks** (e.g. certain fork narratives)—do not confuse them with today’s Ethereum mainnet.

When you read 2017–2021 pieces about “defusing the bomb” or “Ice Age,” file them as archives—**not current ops guides**.

Innovation directions and a 2026 map (selective)



We do **not** reprint “watch sixty new chains / 100× bull lists.” Keep **map coordinates**, not a recommendation list.

Ethereum vs new L1s: competing trade-offs, not one fill-in-the-blank

- **Ethereum path:** L1 for security and settlement; much activity on **Layer2**; deep ecosystem, mature tools, long battle testing.
- **Independent L1 path:** new VMs, consensus, higher throughput or lower fees; must cold-start liquidity and security reputation.
- **EVM-compatible chains:** lower migration cost; differentiate via niches—judge security and decentralization case by case.

There is no single “next Ethereum.” More often: **multi-chain coexistence** + **bridges and intent routers**—treat bridge risk as high by default.

Layer 1 / 2 / 3 (intuition)

- **Layer1:** base chain with its own consensus and security root (Bitcoin, Ethereum, other independent public chains).
- **Layer2:** scales under L1 data/security (especially Ethereum Rollups)—next chapter.
- **Layer3:** more application-specific or recursive scaling talk; definitions still evolve—know “further layers exist.”

Technical routes (name-level)

1. **High-performance / parallel L1s:** low latency, high TPS; often higher hardware/node requirements.

2. **Move and asset-oriented languages:** resource-type safety to reduce some asset bugs (language is not a silver bullet).
3. **Modular stacks:** split execution, settlement, data availability—flexible; complexity and assumptions rise.
4. **Vertical app chains:** games, DeFi, privacy—liquidity and cold-start remain hard.
5. **Bitcoin side ecosystems:** inscriptions, L2 experiments—different exam from "smart-contract L1s," same broad universe.

How to look in 2026 (not a 60-name list)

Ask questions instead of memorizing lists:

- Where do money and users mainly settle?
- Are new apps more often on L1 or L2?
- Are clients and validator/miner distributions healthy?
- Whom do you trust when bridging?
- Do regulation and stablecoin rails affect real use?

SCDO Lab, lightly: the endgame is less "one chain left" and more **verifiable records as default infrastructure**—winners fit security, fees, developer experience, and real business. We care whether actions are auditable and permissions constrained (Agent Ledger-class questions), not which narrative is loudest this year.

A ten-minute hands-on with a public chain

Theory sticks better after one touch (use tiny amounts you can lose, or official testnets):

1. Open the chain's **block explorer**; find one of your txs: success/fail, Gas, from/to.
2. Open a popular token contract page: holders, verified source, upgrade proxy?
3. Ask: am I on **mainnet L1** or some **L2/sidechain**? what unit prices fees?
4. If bridging: assume the bridge can fail; understand lock/mint before large amounts.

Finish those four and you are closer to a real user than someone who only forwards "chain ranking" screenshots.

Chapter wrap

1. Public chains are open shared ledgers and (often) programmable execution—the Web3 base.
2. Ethereum mainstreamed smart contracts; **it is PoS**; much activity is on L2.
3. ERC-20 / ERC-721 let wallets and markets share token semantics; allowances and speculation risks remain.
4. The difficulty bomb is a PoW→PoS historical device, not 2026 news.
5. Do not memorize mega lists; read maps with openness, security, execution, and ecosystem questions.

Next: **Layer2**—why Ethereum stacks another layer, sidechain vs Rollup, and short cards for Optimism / Arbitrum / ZK families in 2026—instead of an encyclopedia.

(End of Chapter 3 · v2 English edition · SCDO Lab)

4. Layer2

SCDO Lab · Web3 Primer for Newcomers v2 · English edition
华语版 / Chinese: `04-layer2.md`
Public chains (especially Ethereum) that stay secure and open get crowded—and Gas gets expensive. Layer2 moves most execution to a “second layer” and anchors results back to L1.
The old manuscript had nearly a hundred pages of scaling encyclopedia—v2 **compresses** to concepts + short cards. Old names like “zkSync2.0 / zkSnyc” are corrected. **Not investment advice.**

Why scale?



A chain chasing three goals at once fights itself (simplified “impossible triangle”):

1. **Security** (attacks are expensive)
2. **Decentralization** (many can verify)
3. **High throughput, low fees** (many cheap txs per second)

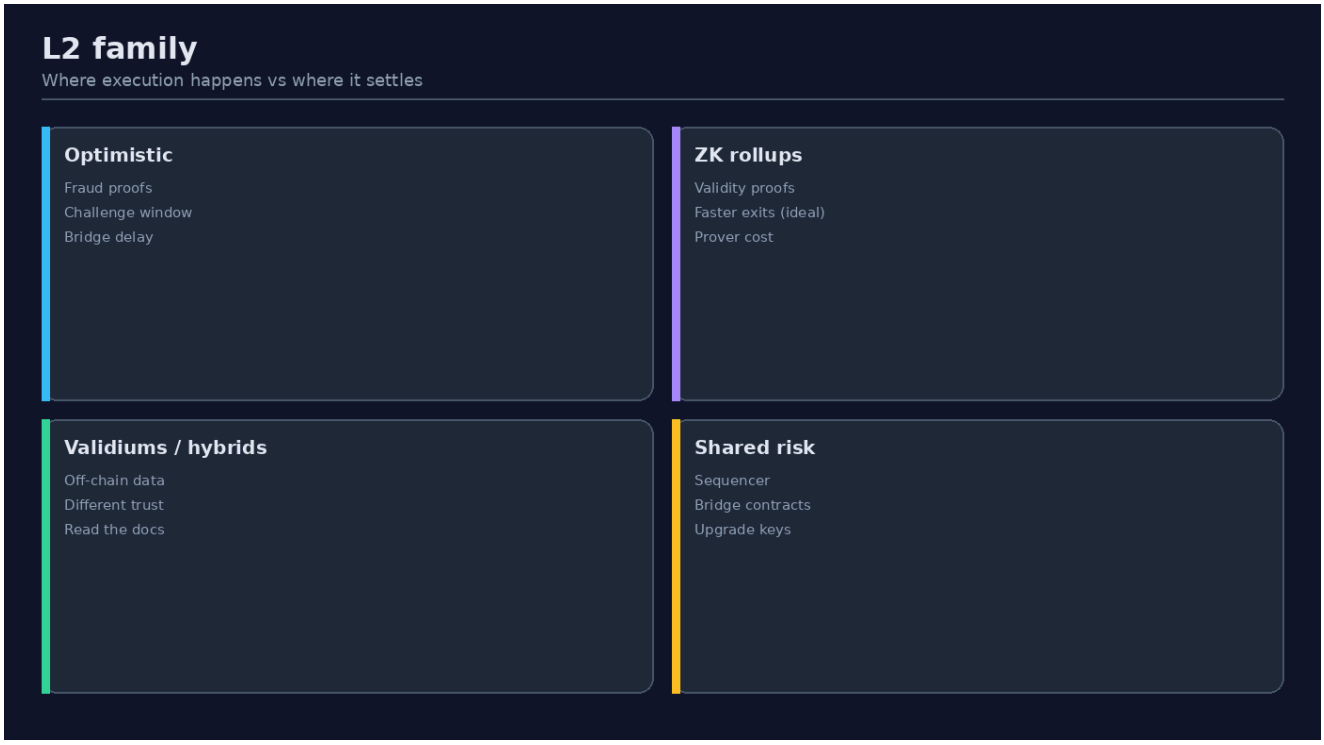
When Bitcoin and Ethereum mainnet prioritize the first two, frequent L1 use hurts: queues, cost, poor UX.
Scaling = let more people afford use while sacrificing as little security and verifiability as possible.

Two broad directions:

Direction	Approach	Examples
On-chain scaling	Change L1 itself (bigger blocks, sharding, etc.)	Historic block-size fights; parts of Ethereum’s roadmap
Off-chain / Layer2	Most execution on L2; periodically settle to L1	Rollups, some channel designs

In 2026 the common Ethereum-user path is: **trust L1 for assets and settlement; do daily ops on L2.**

Layer2, sidechains, Rollups: family relations



Layer2 (L2) in plain words

An **execution layer hanging under some L1’s (often Ethereum’s) security assumptions**: txs are processed on L2; critical data or proofs go to L1 so users can (in ideal designs) still rely on mainnet rules to constrain L2.

Not every self-labeled “L2” has the same safety grade—ask: **if things break, can L1 data/proofs recover funds?**

Sidechains

An independent chain with its own consensus and validators; bridges move assets to/from a parent chain.
Security mainly rests on the sidechain, not automatic Ethereum-grade inheritance.
Historical game-sidechain hacks remind newcomers: cheap ≠ equally safe.

Ecosystems like Polygon include sidechain-leaning designs and lines evolving toward Rollups / stronger security inheritance—**judge the product, not only the brand.**

Rollups

Mainstream Ethereum scaling narrative:

1. Execute many txs on L2;
2. Post compressed data (or commitments) to L1;
3. Use a mechanism so L2 cannot cook the books—fraud proofs or validity proofs.

Two schools:

Type	How correctness is enforced	Newcomer intuition
Optimistic Rollup	Assume honesty; challengers can dispute in a window (fraud proofs)	Withdrawals to L1 often wait a challenge period
ZK Rollup	Each batch comes with a zero-knowledge validity proof	Proof generation is complex; withdrawal logic differs from OP

State/payment channels (e.g. Bitcoin Lightning-style ideas) are “off-chain scaling cousins”: high-frequency small payments among relatively fixed parties—not the same product shape as general smart-contract Rollups. We skip the old chronological long-form.

Choosing a path: a sketch



Ask four questions:

1. Do I need **cheap interaction**, or **maximum Ethereum security inheritance**?
2. Can I accept withdrawal delay (common on some Optimistic paths)?
3. Are the bridge and official entry the domains I verified? (phishing hotspot)
4. Is the app actually deployed on that L2, or only "claimed support"?

Cross-L2 / cross-chain bridges are high risk by default: locked assets, minted wrappers, multisigs or validator sets—large hacks are historical fact. Bridge less; for large amounts, batch and test small first.

Representative short cards (2026, not a ranked buy list)

Map only—**not recommendations**. Names and product lines change; trust official docs.

Arbitrum (Optimistic family)

- **What:** a mature Optimistic Rollup ecosystem on Ethereum with strong EVM compatibility.
- **You will meet:** many DeFi/apps also deployed; Gas cheaper than mainnet.
- **Watch:** mainnet vs Arbitrum networks; official vs fake bridges; challenge-period withdrawal expectations.

Optimism (Optimistic family) and Superchain narratives

- **What:** another major Optimistic camp; emphasizes open stacks and many OP-stack chains.
- **You will meet:** apps, governance, and bridging UX similar-but-not-identical to Arb.
- **Watch:** “same tech stack” ≠ “same chain, same bridge, same risk.”

Starknet (ZK / STARK family)

- **What:** ZK path using STARK proofs; language/tooling historically differed from pure Solidity worlds (EVM/DX keep improving).
- **You will meet:** learning curve; wallets/tools may differ from Arb/OP.
- **Watch:** account model and official entries before large deposits.

zkSync (ZK family; names corrected)

- **Naming:** manuscript forms like “zkSync2.0” / “zkSnyc” are outdated or typos. Use the project’s **current official product names** (and network names in docs), verified against explorers and wallet network lists.
- **What:** a ZK scaling path toward Ethereum aiming at validity proofs and better fee UX.
- **Watch:** do not mix mainnet / testnet / product lines; airdrop rumor zones are phishing-dense.

Other words you will see

- **Base and other app / OP-stack chains:** often tied to an ecosystem or distribution channel—still check what security they inherit.
- **Validium / DA trade-offs:** cheaper by putting data off-chain; security assumptions change—advanced topic; beginners learn “cheaper often swaps assumptions.”
- **Shared / decentralized sequencers:** censorship resistance and single points; still evolving in 2026.

Daily L2 habits (more useful than chasing narratives)

1. **Correct network in the wallet:** “missing” funds are often wrong network.
2. **Bridges and DApps only from official/verified entries**; search ads may be fakes.
3. **Test tiny** before scaling; note your address and tx hashes.
4. **Manage allowances:** malicious `approve` works on L2 too.
5. **Read withdrawal rules before exiting to L1:** time, fees, official bridge path.
6. **Do not conflate sidechains, Rollups, and custodial “one-click cross-chain”** — underlying trust parties differ.

One SCDO Lab link

Scaling makes automated agents (AI agents) able to afford on-chain actions—and makes **mistakes and drains happen faster**. Even cheap L2s need: spend limits, auditable logs, revocable permissions—why Agent Ledger talk stresses: speed serves business; traces serve accountability.

Deeper: where data lives, who sequences, how funds return to L1

Data availability (DA)

For others to check L2 honesty, enough data must be obtainable. Data on Ethereum inherits more security and costs more; elsewhere is cheaper with different assumptions. Beginners need not memorize every acronym—remember the trade-off: **cheaper often means more extra honesty assumptions.**

Sequencers

Most L2s today still have a **centralized or semi-centralized sequencer** ordering txs and giving fast confirmation. Smooth UX comes with:

- Downtime → temporarily painful;
- Censorship → some txs delayed;
- MEV → ordering itself has value.

Long-term roadmaps discuss decentralized and shared sequencing; 2026 is still “product UX first, decentralization catching up.” Treat the sequencer as a **visible single point**, not pretend it is absent.

Exits and bridges

- **Official bridges:** clearer rules; Optimistic withdrawals may wait challenges.
- **Third-party / liquidity bridges:** faster, extra trust and contract risk.
- **CEX deposit/withdraw:** convenient; essentially custodial accounting.

Choice depends on size, urgency, and whom you trust. There is no free lunch of “fast + zero trust + zero learning.”

Bitcoin side paths and “other L2s”

Ethereum L2s are loudest, but scaling ideas are not exclusive. Bitcoin has Lightning-style channels; other L1s have their own L2s or app chains. Cross-ecosystem naming is messy—some call every sidechain an L2. **Judge by security inheritance, not marketing labels.**

How to read L2 news (don’t be a fan)

Force five questions:

1. Evidence system: Optimistic, ZK, or none?
2. Where is transaction data published?
3. Who holds upgrade keys; is there a timelock?
4. If assets stick, what social/technical exit exists?
5. Are fake official sites phishing lately?

Muscle memory beats memorizing TVL ranks.

Fee intuition

L2s cut **execution and congestion cost**, not magic free rides. Congestion, complex contracts, allowances, and bridges can still spike. Habit: check estimates before send; split large moves.

One line for builders (not a tutorial)

If you ship apps: document which L2 users enter from, how the official bridge is described, and how wrong-network mistakes are blocked. Many “lost coins” are UX failures. Teaching users to pick the network is part of product duty.

A newcomer’s day (story)

Suppose you hold a little ETH on mainnet, want a DeFi app, and Gas scares you off. You add an L2 network (or your wallet already has it), copy the bridge address from official docs—checking every letter of the domain. You bridge ~$5 equivalent first: minutes later (or longer), the explorer shows L2 arrival. You do a small swap; Gas is clearly cheaper. You note: mainnet and L2 addresses often match, but **balances are per network**; wrong network feels like “coins vanished.” After playing, you decide whether to bridge back and budget wait time and fees. The day did not make you rich; it built muscle memory better than ten AMAs.

Unknown signature? Refuse. DM “your bridge failed, fix here”? Treat as scam. Official bridge congested? Wait rather than jump to a strange fast bridge on day one. Keep that paranoia—L2s cheapen txs and also cheapen fraud’s marginal cost.

Advanced: one ecosystem may have many chains and similar testnet/mainnet names. Habit: **check chainId / explorer URL before signing.**

FAQ (L2)

Q: Are L2 coins the same as mainnet?
A: Balances are per network; addresses often match, but different networks are different ledger views.

Q: Why do withdrawals sometimes take long?
A: Some Optimistic paths have challenge windows; third-party bridges trade time for different trust.

Q: Is a sidechain an L2?
A: Not necessarily. Judge whether security depends on the parent’s data/proofs—not ads.

Q: Which L2 should I pick?
A: Where your app lives, whether the bridge is official, client/community maturity—not a single TVL screenshot.

Q: Are “zero Gas” campaigns safe?
A: Sponsored Gas can be account abstraction—or phishing wrapping; still read signatures.

Values inside scaling debates

Surface fights are about throughput; deeper fights are values: who may sequence? who may upgrade contracts? can users force-exit via L1 if the sequencer misbehaves? Different answers yield different products. Some accept a bit of centralization for silky payments; others prefer costlier maximal independence from one company. No unique right answer—but marketing slogans as security proofs are wrong answers.

Ethereum scaling history rotated state channels, Plasma, sidechains, Optimistic Rollups, ZK Rollups through the spotlight. Losers were not always “scam tech”—often engineering/UX lagged, or security models failed mainstream acceptance. Today’s winning narrative may rewrite in three years. This book uses cards, not a final ranking. You will change phones and default L2s; migration cost is assets and habits, not loyalty.

For institutions, custodians may bundle account abstraction + sponsored Gas + allowlisted contracts—another trust model: convenient, partially Web2 again. Education’s goal is not forbidding convenience; it is making you **name whom you trust**. If you cannot name them, do not deposit more than insomnia money.

Builders/ops: screenshots that say “pick the right network,” disable dangerous buttons on wrong networks—beats ten threads. L2s cut on-chain fees, not communication fees—and communication fees are the real Gas of most user asset accidents.

Shards: another relative of scaling



Layer2 is “another floor on top.” **Sharding** splits one large chain into parallel lanes: accounts and transactions are partitioned; cross-shard moves use dedicated bookkeeping (e.g. Classic **Debt**).

Idea	One line
Local blocks	Produced and confirmed inside a shard
Cross-shard	Sender shard records first; recipient shard credits after checks (details vary)
vs Rollups	Shards are often peer chains in one family; Rollups usually settle back to one L1

SCDO layout (2026, educational): Classic **Shards 1–4** use **ZPoW**; a peer **Shard0 EVM (Ethash, chainId 5680)** hosts contracts and Agent Ledger. Details: Chapter 10 (SCDO) and Chapter 11 (Mining). Here you only need the map: **shard ≠ sidechain ≠ Rollup**.

Chapter wrap

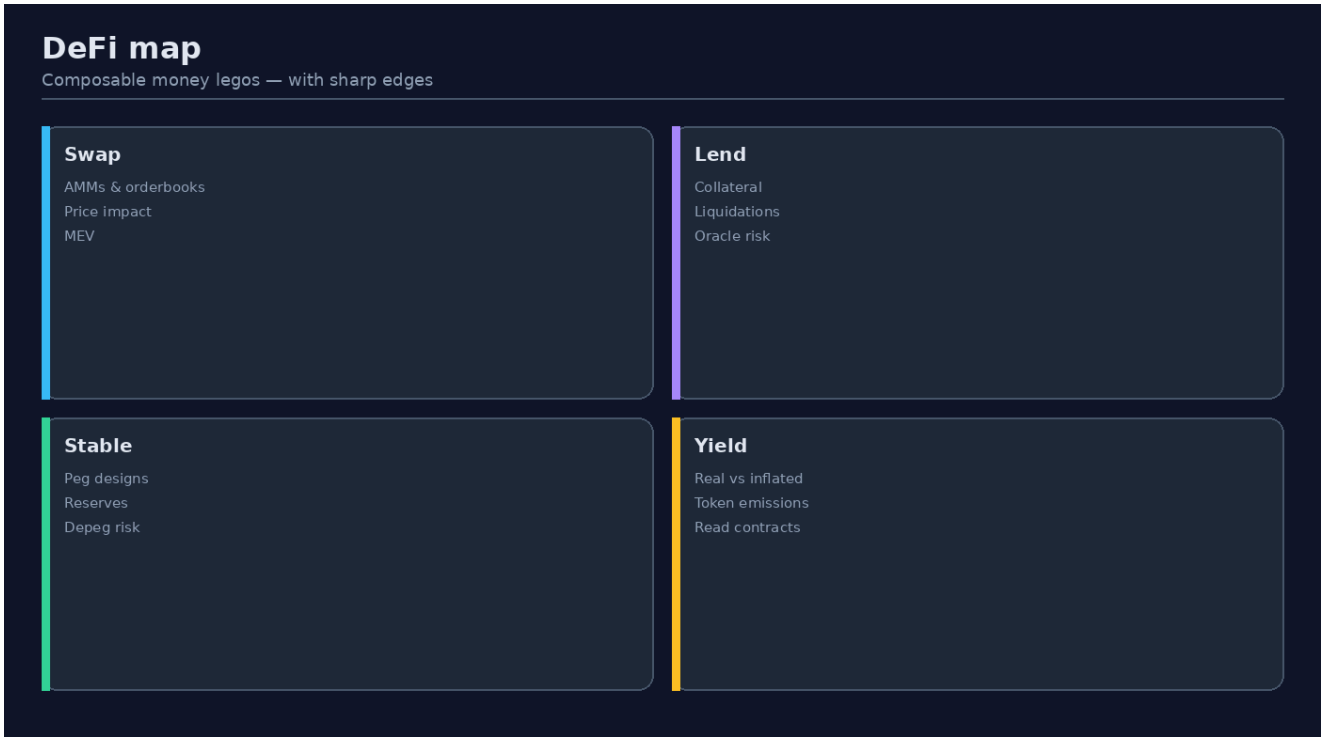
1. Crowded L1s need L2s; in 2026 much Ethereum activity is on L2.
2. Sidechain ≠ Rollup; whether security inherits L1 is the soul question.
3. Optimistic uses challenges; ZK uses validity proofs—UX and withdrawals differ.
4. Memorize cards, not encyclopedias; bridges and fake sites kill faster than “wrong narrative.”
5. Next: **DeFi**—how swaps, lending, and market-making work on these L1/L2s, and where risk lives.

(End of Chapter 4 · v2 English edition · SCDO Lab)

5. DeFi

> SCDO Lab · Web3 Primer for Newcomers v2 · English edition
> 华语版 / Chinese: `05-defi.md`
> DeFi = decentralized finance: smart-contract Lego for swaps, lending, market-making, and more. Useful—and scary.
> No protocol white-paper dumps; do not worship TVL. **Not investment advice, and not a yield tutorial.**

What is DeFi?



DeFi (Decentralized Finance): on public chains (and L2s), open or verifiable smart contracts offer pieces of traditional finance—swaps, lending, liquidity provision for fees, derivatives—while reducing “must open an account and trust one broker’s backend.”

You usually need:

- A self-custody wallet (or a custodial entry whose risks you understand);
- Some native coin for Gas;
- Literacy about what a signature authorizes.

Where it came from

Bitcoin proved on-chain assets; Ethereum contracts made money rules programmable. Late 2010s through 2020’s “DeFi Summer,” AMMs and lending exploded; then bubbles, hacks, stablecoin storms, regulatory pressure. By **2026** DeFi is calmer: more institutional and RWA (real-world asset) talk—but **contract risk and user-ops risk never left**.

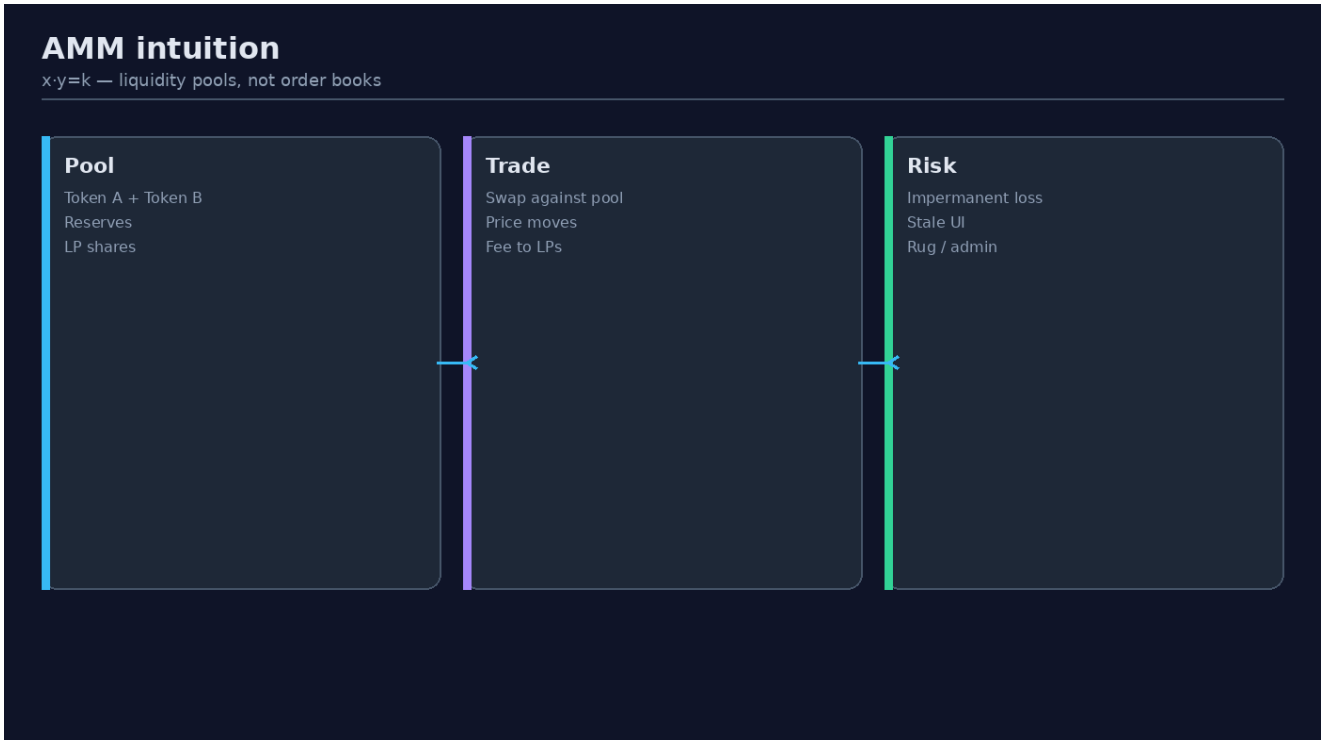
Traits (double-edged)

Trait	Benefit	Cost
Permissionless access	Wallet → try	Scammers deploy fake contracts too
Composability	Protocols call each other like Lego	Risk contagion; one pit topples a stack
Transparency	Parameters and positions inspectable	Transparent ≠ you understand; MEV can sandwich you
Global 24/7	No market close	No support rollback; wrong sign stays wrong

What still blocks mainstream adoption?

UX (Gas, bridges, seeds), compliance (stablecoins and on/off-ramps), security (hacks and phishing), education (impermanent loss and liquidation jargon). Do not believe “DeFi already replaced banks.”

Uniswap: AMM for newcomers



What it is

Uniswap is a flagship **DEX** protocol in the Ethereum ecosystem: instead of a centralized order book matching buyers and sellers, it uses **liquidity pools** + **pricing formulas** (automated market makers, AMMs).

Plain:

- People deposit two tokens into a pool (become LPs);
- Traders swap A for B; the pool adjusts price by algorithm;
- LPs take **impermanent loss** risk from price moves and may earn trading fees.

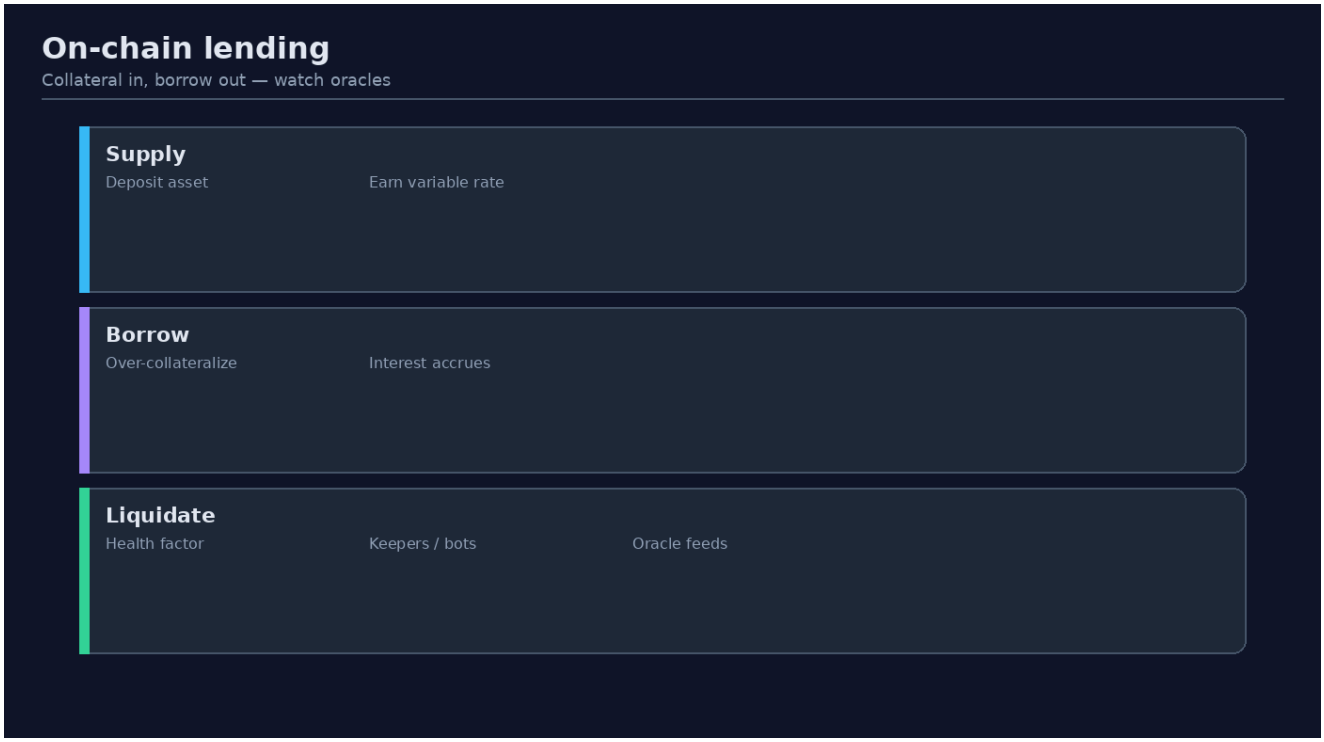
Versus CEXes

- You often trade while self-custodying keys (frontends still phish);
- No classic “deposit to exchange account” layer (unless you already sit on an exchange);
- Price forms from pools and arbitrageurs; extreme markets can produce huge slippage.

Mental model when using

1. Correct network (Ethereum L1 or an L2 deployment).
2. Clear pair, slippage, fee tier.
3. First token approval to a router—be cautious; smaller allowances when possible.
4. Tiny test swap; large size may split or seek better routes.
5. Versions and fee tiers change—trust the UI and explorer of the moment, not memorized numbers.

Lending: Aave-class protocols (with risks)



What they do

Like an “on-chain pawnshop + money market”:

- **Depositors** supply assets and earn interest;
- **Borrowers** post collateral and borrow other assets;
- If collateralization fails on price moves, **liquidation** sells collateral to keep the system solvent.

Aave is a long-standing major lending protocol (peers exist). Experiments like GHO are ecosystem extensions—newcomers learn collateralized borrow/repay before stablecoin issuance details.

Risks you must understand

1. **Liquidation:** collateral falls or debt rises; liquidations add penalties.
2. **Smart-contract risk:** bugs, oracle manipulation.
3. **Rates and utilization:** when money is scarce, rates spike.
4. **Nested asset risk:** collateral that is itself another DeFi receipt stacks risk.
5. **Ops risk:** wrong network, wrong allowance, misread health factor.

Flash loans: borrow-and-repay in one tx for arbitrage/liquidation—advanced tools, not “uncollateralized forever loans for the masses.” “Risk-free flash-loan riches” is usually a scam or hidden complexity.

Curve and stable pools (compact)

Curve-class protocols excel at low-slippage swaps among **similar assets** (various stablecoins, related derivatives) and grew complex reward/governance/bribe markets (historically Convex and peers).

Newcomer takeaways:

- Stables can depeg; pools can lose;
- “Stable” is a design goal, not a law of physics;
- High yields often come from token incentives—when incentives end, APY collapses.

Skip memorizing full bribe-war history; advanced readers can dig later.

LP and token logic

Providing liquidity is not “bank deposit interest”:

1. You deposit assets and receive an LP receipt (often another token).
2. Traders pay fees; you share by share.
3. When relative prices move hard, what you withdraw may underperform simply holding—**impermanent loss (IL)**.
4. If farm rewards stack on top, count reward-token dump pressure into real yield.

Mindset: first ask “is this pool sensible without reward tokens,” then “how long can rewards last.”

How to read TVL

TVL (Total Value Locked): sum of asset valuations locked in a protocol’s contracts. Media loves rankings.

How to read:

- **Useful:** scale and relative heat;
- **Not gospel:** recursive collateral can inflate, prices distort, competitors game numbers;
- **Disaggregate:** what assets? free exit? audits and multisig posture?

One line: **TVL is one thermometer, not a full medical report.**

Synthetic assets (principles)

Synthetics: use collateral and oracles to tokenize exposure to some price (stocks, FX, crypto indexes).

Usual pieces:

1. Collateral;
2. Debt / synthetic position;
3. Price feeds (oracles);
4. Liquidation and stability mechanisms.

The hard part is always **information input**: chains do not know real-world prices unless someone feeds them. Wrong or manipulated feeds make synthetics absurd. Dai, sUSD, and peers trade different assumptions—compare decentralization, censorship resistance, and reliance on real-world custody; do not compare slogans only.

Ten newcomer anti-pit rules

1. Cross-check official sites and contract addresses from multiple sources.
2. Minimize allowances; revoke regularly.
3. Do not max leverage if you do not understand health factors.
4. Stablecoins: reserves and issuer risk matter more than a “USD” ticker.
5. Assume high APR is unsustainable.
6. Someone should clearly explain upgrade rights, multisigs, timelocks.
7. Separate pits: principal loss, IL, liquidation penalties, token → zero.
8. Exchange “support” and Telegram DMs are not official.
9. Tax and local compliance are your responsibility.
10. Use sims/tiny size as tuition—not rent money.

Stablecoins: DeFi’s blood—and a risk source

Without a relatively stable unit of account, lending and trading shake. Rough types:

- **Fiat-backed / custodial:** company + bank-account narratives; watch reserve transparency and freeze policy.
- **Crypto-collateralized:** more on-chain collateral; complex markets; extremes can depeg.
- **Algorithmic / hybrid:** history of blow-ups; “algorithm guarantees forever $1” raises alarms.

In DeFi, a “USD-named token” ≠ bank dollars. Ask: who can freeze? where are reserves? how do oracles price?

MEV and your fill price

Miners/validators/searchers may front-run, sandwich, or reorder. Wallet “expected fill” is not guaranteed. Protections include sensible slippage, private tx rails (advanced), avoiding

huge sweeps of thin pools. You need not become an MEV researcher—know the **public mempool is not a charity ball**.

Governance tokens and the “voting power illusion”

Many protocols issue governance tokens. Holding ≠ you alone control the treasury. Look at quorums, delegation, multisigs, emergency switches. Treating gov tokens as “dividend stocks” disappoints; treating them as **coordination tools** + **speculative assets** is more honest.

Abstract ops flow (no specific frontend)

1. Dedicated browser profile and wallet.
2. Confirm network and contracts (multi-source).
3. Tiny swap; confirm receipt.
4. If LPing: stress IL at ±30%/±50% price moves.
5. If lending: set liquidation alerts; do not max.
6. After ops, review allowances; revoke unused.
7. Bookkeep (future you will thank you for taxes).

Regulation and geography (principles)

Jurisdictions differ on DEX frontends, stablecoins, and staking ads. Some regions change frontend access; some require KYC entries. This book does not teach “how to dodge local rails”—only: **tech is global; law is local.**

When people say “smart money”

On-chain whale addresses are visible; that does not mean you should copy-trade. Fake deposits, scam pads, and wash trades can perform. Analytics are tools; without situational awareness they become reverse indicators.

Composition risk: when Lego falls

Composability means your yield may nest four layers—deposit receipts re-collateralized, farmed, rewards sold into another debt. Each layer has contract and oracle risk. Bull markets call this “capital efficiency”; bears call it “dominoes.”

Practical: draw your position dependency graph. If unclear, cut leverage and layers. Protocol “safety modules” / insurance funds buffer some risk—they are not infinite policies. “One-click strategy vaults” outsource complexity to strategists—you also trust them and their contracts.

Another hidden risk is **liquidity illusion:** high TVL with circular same-asset staking; real exits slip hard. Probe depth with sims or tiny size before large exits.

Treat DeFi as a toolbox: swap when you need to swap, lend when you need to lend, then reduce exposure. Chasing max APY all day usually sells volatility and tail risk to yourself.

FAQ (DeFi)

Q: Is impermanent loss losing principal?
A: It is an opportunity gap vs “just hold”; asset quantities change on exit; extremes hurt.

Q: Audited = safe?
A: Audits reduce risk; they do not erase it—also watch permissions and dependencies.

Q: Where does high APY come from?
A: Often token subsidies or high risk; when subsidies stop, so does the rate.

Q: Can I just hold stables for interest?
A: Still contract, issuer, and depeg risk; rates also move.

Q: Wallet shows fake tokens?
A: Ignore; do not click fake “claim” pages.

Liquidity, price discovery, and where newcomers sit

AMMs let anyone LP—and move market-making risk from desks to retail LPs. You earn fees (and maybe rewards) and pay adverse selection: informed flow trades when it hurts you. Thin pools amplify impact; deep pools may thin your fee share per trade. Fee tiers and concentrated ranges (where supported) are specialist work—novices start with full-range tiny size or simply swap.

Lending markets hand rates to supply/demand algorithms. High utilization raises deposit rates and borrowing costs—that is a feature. Danger is longing collateral while borrowing another asset; flash crashes liquidate faster than your fingers. Treat health factor as a seatbelt, not a high-score challenge.

Yield aggregators auto-compound. They save time and amplify contract/strategy risk. Ask: if the strategist vanishes, can you still exit? timelocks? admin mint rights? Write answers in notes—more responsible than APY screenshots on social media.

DeFi turns part of global capital markets into 24/7 permissioned Lego. It does not erase risk; it reallocates risk and information. You see on-chain data; counterparties see your positions. Transparency is double-edged: auditable and huntable.

Chapter wrap

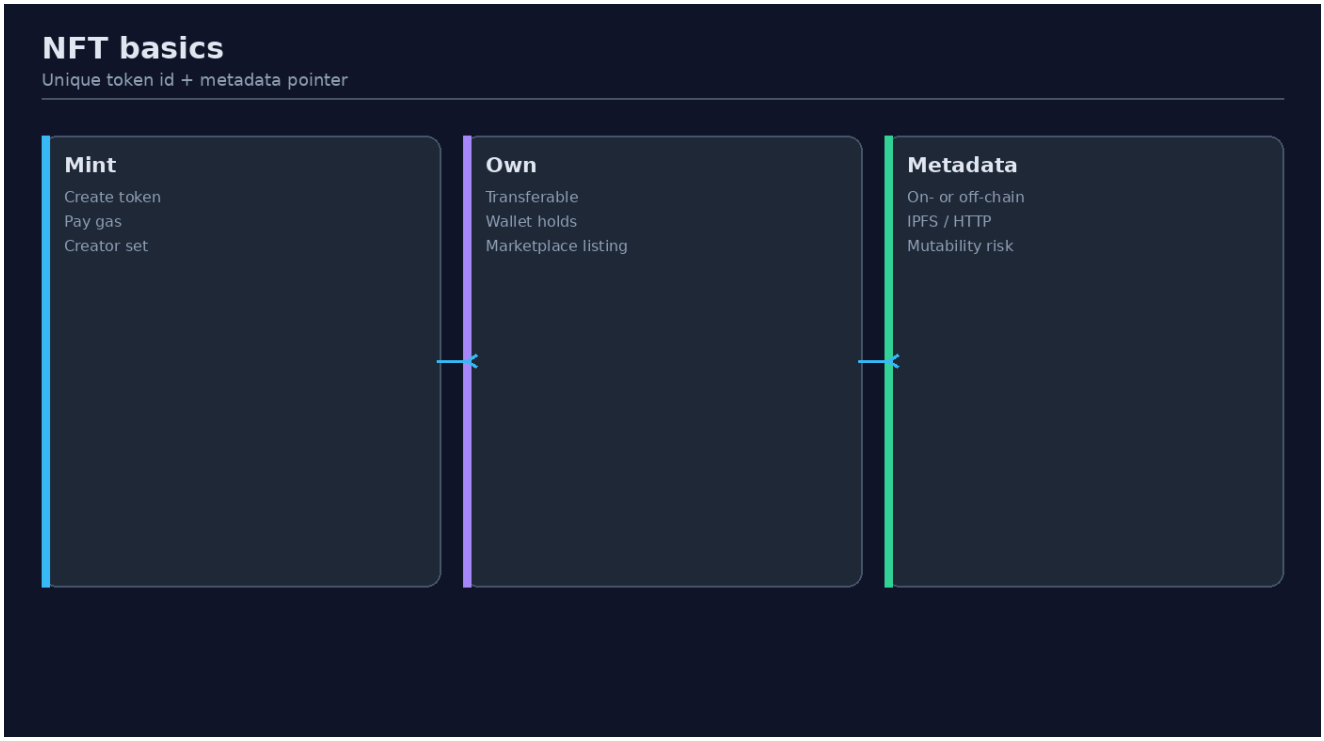
1. DeFi is on-chain financial Lego; core is contracts and collateral liquidation—not a magic printer.
2. Uniswap teaches AMMs; lending teaches liquidation; Curve teaches "stable isn't always stable."
3. LPs face impermanent loss; TVL can mislead.
4. Synthetics live on oracles and collateral, not pretty names.
5. Next: **NFT and the metaverse**—pull unique credentials out of financial Lego and cool the 2021 foam.

(End of Chapter 5 · v2 English edition · SCDO Lab)

6. NFT and the metaverse

> SCDO Lab · Web3 Primer for Newcomers v2 · English edition
> 华语版 / Chinese: `06-nft-metaverse.md`
> NFTs are **on-chain non-fungible credentials**; the metaverse is a larger, fuzzier narrative basket.
> Turn down 2021–22 hype; Sandbox / Decentraland are **cases**, not "must-buy leaders." **Not investment advice.**

What is an NFT?



NFT (Non-Fungible Token): each token has a unique ID and cannot be swapped 1:1 indistinguishably like ERC-20. Common Ethereum standards: **ERC-721**, also **ERC-1155**, etc.

Plain: an NFT is more like an **on-chain title/badge number** proving "this contract's this ID currently belongs to this address." It does **not** automatically mean:

- You own copyright in the image file (often separate);
- The project will operate forever;
- Floor price only goes up.

Infrastructure you will actually meet

- **Issuance contracts:** mint rules, royalty interfaces, upgradeability.

- **Metadata:** name, image links—on-chain, IPFS, or a central server (server dies → image breaks).
- **Marketplaces:** display and trade frontends—also fake sites.
- **Wallets:** show NFTs, sign listings and approvals.

Uses that actually make sense (rough reliability order)

1. **Tickets and access:** on-chain tickets reduce fakes; still watch scalpers and fake mints.
2. **Membership and permissions:** hold NFT X to enter a community / unlock features.
3. **Game items and cosmetics:** tradable and portable in the ideal case; bad economies still collapse.
4. **Certificates and achievements:** courses, contributions, attendance—issuer reputation matters.
5. **Collectibles / art:** culture and speculation coexist; most go to zero—normal market distribution.

Bitcoin-ecosystem inscriptions and peers also “carve / attach unique data to chain assets”—standards and risk models differ from Ethereum NFTs; point only here.

What the bubble taught



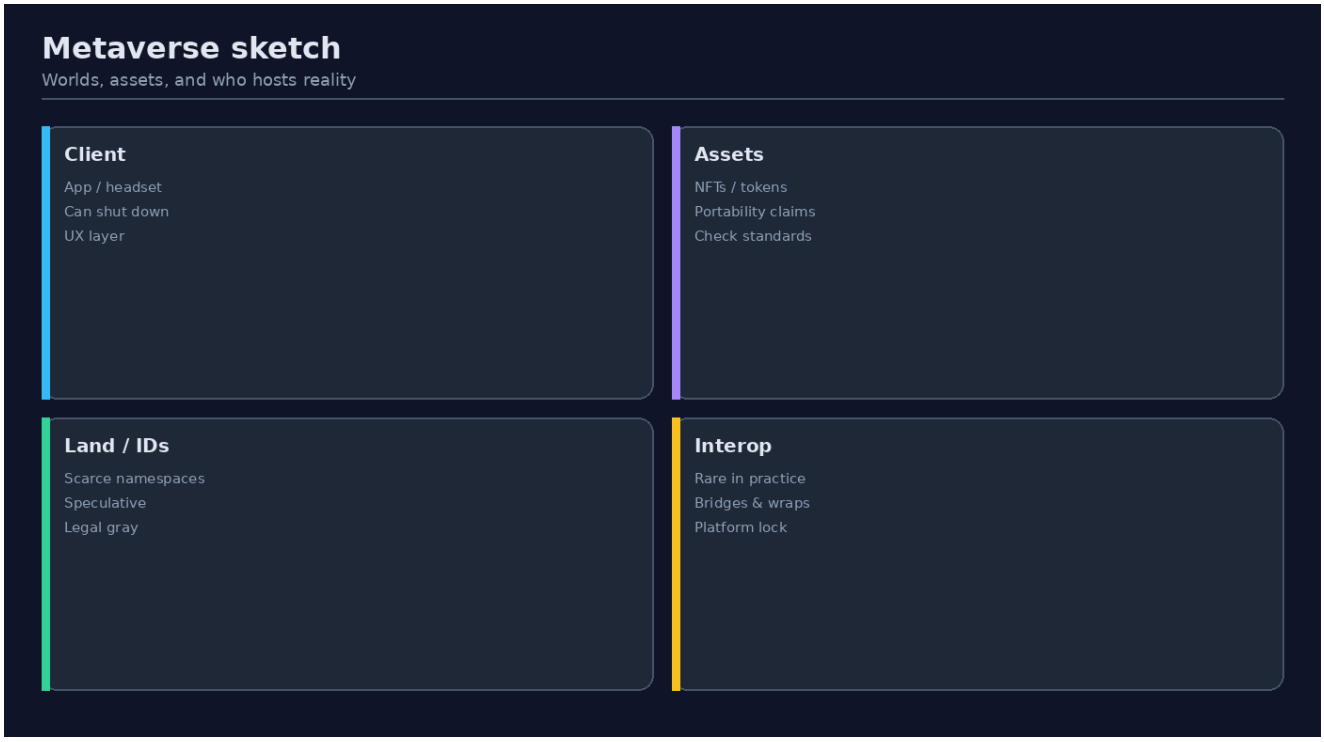
Around 2021: avatar JPEGs, sky-high floors, celebrity shills, everyone minting. After the drawdown:

- **One-way liquidity:** easy to sell in bulls; no bids in bears.
- **Royalties and marketplace rules change:** creator cuts are not physics.

- **Wash trading** can fake floors and volume.
- **Phishing mints**, fake support, stolen approvals zero you faster than “wrong fundamentals.”
- **IP chaos:** people sell rights they never had.

In 2026, healthier questions: **whose problem does this credential solve? If speculation stops, does it still exist?**

Metaverse: concept, volume down



Concept

No single definition. Marketing often means: persistent shared virtual spaces + identity + economy + interactive content. Tech may include game engines, social, creator tools, payments, sometimes blockchain items and land.

Commonly listed traits (not promises)

- Persistent “world” feel;
- Editable / creatable;
- Digital identity and social;
- Economic systems (may include on-chain assets);
- Links to real scenes (events, brands, remote work).

Blockchain is not a sufficient condition for a metaverse: many virtual worlds run fully centralized. On-chain parts mainly offer **verifiable digital ownership and possible**

interoperability— "possible" matters; cross-world portability remains hard.

Relation to NFTs

NFTs often play land deeds, outfits, tickets. Without a living world and players, deeds are quiet JSON. Conversely, good games need not be on-chain first; ask whether anti-fraud, markets, and open economies truly require a chain.

More realistic directions

Brand virtual launches, remote concerts, training sims, industrial digital twins—often **do not** require buying "on-chain land" first. Equating the metaverse with "buy land, get rich" is the most misleading newcomer simplification.

Cases: The Sandbox and Decentraland (not leader shills)

Both early combined **tradable virtual land/experiences** with crypto wallets and were once called "metaverse representatives" in media.

How to read cases:

Dimension	What to observe
Product	Are people still creating and gathering inside?
Assets	Are LAND-class NFTs for in-world production or pure speculation?
Company / foundation	Who sets rules? Who expands the map?
Cycle	Hype rides markets and narratives—drawdowns are normal
Lesson	Virtual real-estate valuation is fragile; traffic acquisition is costly

v2 is explicit: learning samples, not a "2026 must-buy pair." Many peer worlds exist—some more chain-game, some more UGC platforms. Pick worlds by content and community, not headline count.

Newcomer checklist

1. Separate **collecting preference** vs **speculation**; treat the latter as able to go to zero.
2. Before mint: contracts, supply, free-mint traps (allowance drains).
3. Ask hosting: IPFS? centralized URL?
4. Ignore "floor-bot guaranteed profits."
5. Metaverse land: ask "will I use it" before "will it pump."
6. AI art + NFT: provenance and originality disputes grow—read licenses.

Metadata and “broken image” problems

Buying an NFT often buys: a tokenId + a metadata pointer. If images live on the project’s server, shutdown breaks images. IPFS / on-chain data resist censorship better but are not infinite free disk. Ask: **is metadata frozen? what is the backup plan?**

Creator royalties were once treated as default; some markets later made them optional—creator/platform bargaining continues. Do not write “forever royalties” into financial fantasies.

Royalties, licenses, copyright



On-chain transfer ≠ automatic commercial copyright. Read the project license (derivatives? commercial use?). AI-generated mints muddy copyright further—litigation and platform policy still evolve. Read terms before collecting or commercial use.

Identity NFTs and “soulbound”

Some credentials are non-transferable (or hard to transfer) for reputation, eligibility, attendance. Upside: less pure speculation. Downside: privacy and discrimination risk (soulbound to an address can be tracked). Good designs consider **minimal disclosure, revocation, refresh**.

Metaverse landing checklist

If your team builds a “world” rather than a story:

1. Frame rate and feel good enough?
2. Creator tools usable by ordinary people?
3. Economy runs without token subsidies?
4. Identity useful across contexts—or only land JPGs?
5. Moderation, child safety, harassment handling planned?

If unanswered, it is still marketing metaverse.

Practical brand / ticketing fits

In 2026 success more often looks like: concert anti-counterfeit tickets, membership year cards, digital keepsakes—**small clear rights**, not “buy land, become a digital landlord.” SCDO view: credentials serve verifiable events, not empty real-estate myths.

After avatar culture

Avatar NFTs were social currency: Discord access, allowlists, recognition. After the wave, the plainer question remains—**how to verify digital identity cheaply** without KYC every time. Tickets, memberships, event badges, course completion can be NFTs or verifiable credentials. Success becomes: smooth redemption, hard forgery, users completing flows with custody or smart-wallet help while understanding custody costs.

For artists: on-chain auctions lower geography barriers and bring platform cuts and plagiarism. Protect with watermarking, provenance, counsel—not “mint first, think later.” For brands: small credential experiments beat airdropping hundreds of thousands of unused images.

If you still use “metaverse,” prefer “persistent multiplayer online world + optional on-chain assets.” Honest wording filters half the fundraising talk.

FAQ (NFT)

Q: Rising floor = quality?
A: Not necessarily; wash trades or transient liquidity.

Q: Listed on a marketplace = I own copyright?
A: Usually not automatic copyright transfer.

Q: Are free mints always losses?
A: Not always—but often allowance phishing; read wallet prompts.

Q: Can I buy metaverse land?
A: Treat as entertainment/experiment budget—not must-have housing.

Q: How do I send an NFT to a friend?
A: Confirm network and address; test small.

Market structure and creator bargaining power

NFT market competition once centered on fees and royalty toggles. Creators want secondary cuts; traders want low friction; platforms want share. Policy flip-flops followed. For creators, stabler income often comes from primary sales, commissions, brand deals—not assumed perpetual royalties. For collectors, separate: community ticket, art preference, or greater-fool speculation. They can overlap; loss psychology differs—stacking them invites self-deception.

When museums try on-chain collection IDs, the point is often provenance and education—not floors. That use pulls NFTs back toward "credential." Games and tickets likewise. Off the trading-screen dopamine, long-term value depends on offline/online rights delivering: entry, redeem, upgrade. Read the rights sheet before buying; no sheet → treat as decoration.

On metaverse land: buildability, traffic, and interoperability decide stage vs wasteland. Slicing land into many NFTs helps speculation; it does not auto-produce audiences. Audiences come from content and social; content from tools and incentives; if incentives are only inflationary tokens, audiences leave. Budget levels and live ops usually beat "two more virtual street corners."

From speculation screens back to usage rights

When volume falls, surviving NFT uses almost all bind **usage rights**: door scan works, ticket redeems, membership unlocks, in-game gear works. Design to that north star—healthier than rarity tables. Issuers should publish rights duration, whether rights move on transfer, failure/refund rules; collectors should treat those as pre-purchase reading, not post-hoc slogans.

Technically, dynamic NFTs (updatable metadata) fit game state and membership tiers—and introduce central switches: who updates? are updates logged on-chain? Static NFTs are simpler but hard to express growth. Choice is situational.

Cross-chain NFTs and "wrap bridges" add mapping errors and fake-bridge risk. If rights work on one network you understand, do not bridge for multi-chain flex. Teaching users with "which city's theater holds the ticket" often beats explaining cross-chain messaging.

For teams like SCDO that emphasize verifiable records, NFT/credentials should prioritize **proof that events happened** (compete, settle, achieve)—not new land illusions. Proofs should be verifiable, expirable, privacy-aware—trade-offs belong in product docs, not slogans.

Compliance display and minors

When NFTs enter schools, museums, or youth events, content ratings, privacy consent, and guardian authorization hit walls before chain tech. Educational credentials should avoid hard-binding public addresses to real identities; selective disclosure is better. Commercial activity must handle invoices, withdrawal rights, and false advertising— "guaranteed forever appreciation" language is a minefield. Bring counsel into design; cheaper than deleting posts later.

Cross-jurisdiction digital loans for exhibitions need clear authorization and minting rights so one tokenId does not carry two rights narratives. Standards are unfinished; until then contracts are the main language, chain the attachment.

Chapter wrap

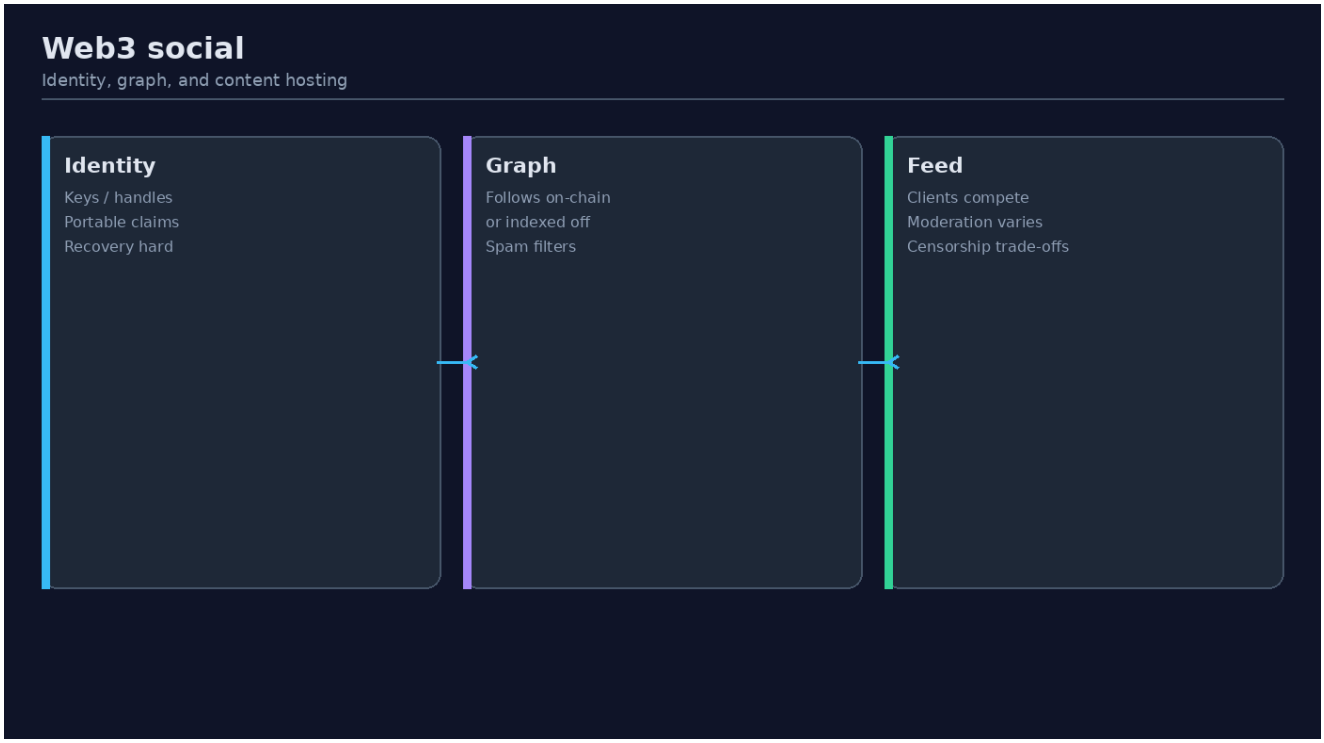
1. NFT = unique on-chain credential; use cases beat avatar speculation.
2. Bubbles proved: high valuations without real demand do not last.
3. Metaverse is a basket; blockchain is an optional part.
4. Sandbox / Decentraland are cases, not must-buy lists.
5. Next: **decentralized social**—when posts and follow graphs want protocols, what federation, Lens, and Nostr each compromise.

(End of Chapter 6 · v2 English edition · SCDO Lab)

7. Decentralized social

SCDO Lab · Web3 Primer for Newcomers v2 · English edition
华语版 / Chinese: `07-social.md`
When social wants to “go protocol,” the fight is not another App skin—it is **whether relationships and content are portable, who can ban you, where data lives**.
Compresses the old two long essays; clarifies **federation ≠ fully decentralized**.
Not investment advice.

What is Web3 social?



Web2 social: accounts live on platforms; fans and posts mostly lock to platforms; companies change rules.
Web3 social tries to put some identity, follow graphs, and content indexes on **open protocols or user-controllable storage**, so apps become replaceable “clients.”

Ideal: change frontend, keep your social graph.
Reality: UX, moderation, discovery, and anti-spam remain hard; many products are still “Web2 with wallet login.”

Tiny history

From forums and blogs to giant feeds, centralization brought growth and lock-in/censorship fights. After blockchains heated up, “on-chain social” experiments appeared: follow graphs on-chain, NFTs as tickets, tokens for posting—some shipped products; some only shipped charts. In 2026 the calm read separates protocol layers (graphs, relays) from app layers.

How to enter (low risk)

1. Start as a reader: which protocol does this product use?
2. Experiment with tiny funds or empty accounts; do not bind your main wallet’s large assets first.
3. Separate **on-chain writes** (costly, slow, permanent) vs **off-chain storage** + **on-chain pointers**.
4. Assume exits, shutdowns, and protocol migrations can happen—test portability.

Stack layers: infrastructure / middleware / apps / tools

Layer	Role	How newcomers touch it
Infrastructure	Public chains/L2s, identity, storage, availability	Usually indirect
Middleware	Social-graph protocols, indexing, notifications, permissions	Decides “can I move”
Apps	Clients and creator studios you open daily	The thing you swipe
Tools	Analytics, anti-phishing, cross-client import	Safety and migration

Choose products backwards: is the app nice? → is middleware open? → is infrastructure costly, where are censorship points?

Three routes: Lens / Mastodon / Nostr



Lens-class: on-chain social graph protocols

- **Idea:** follow/post relations as protocolized as possible so many clients share a graph.
- **Feel:** often wallet signatures; on/near-chain ops affect fees and pace.
- **Watch:** open protocol ≠ one frontend is decentralized; NFT/token incentives can inject speculative noise.

Mastodon: the Fediverse

- **Idea:** many independent servers (instances) interoperate with shared protocols—like email domains.
- **Critical clarification: federation ≠ no admins**. Each server still has people setting rules, bans, and storing your data. You "change post offices," not "abolish post offices."
- **Fit:** people who dislike one global corporate moderator and will choose a community server.
- **Cost:** discovery is fragmented; server risk (admin malice or shutdown).

Nostr: key identity + relays

- **Idea:** identity ≈ key pair; messages to multiple relays; diverse clients.
- **Feel:** more "crypto-native"; spam and discovery depend on clients and relay quality.
- **Watch:** lost private key ≈ lost identity; relays can refuse service—censorship resistance is a degree question.

Contrast table

	Lens-class	Mastodon federation	Nostr
Identity	Often wallet / profile NFT	Server account	Public/private keys
Moving	Depends on protocol + clients	Server export / interoperability	Clients easier; maintain your relay list
Decentralization point	Chain and governance assumptions	Many servers, not zero admin	Many relays, not automatic permanence
Main friction	Signatures, fees, complexity	Server choice and fragmentation	Spam and key security

Farcaster and similar "sufficiently decentralized + strong product" routes mix on-chain identity with off-chain messages—when reading docs, grab **data availability and censorship points**.

Trends and boundaries (2026)



Worth watching: verifiable credentials in social (attendance, contribution, weak anti-Sybil signals); client competition; better wallet-signing UX.
Do not fantasize: full replacement of super-apps next year; an unmoderated high-quality global square appearing automatically; token incentives alone growing healthy public discourse.

Security: "support" DMs, signature approvals, fake airdrops are dense in social scenes—Chapter 9 generalizes.

SCDO, lightly: if social graphs serve real communities, they need auditable permissions and anti-abuse—not only points. Agent auto-posting needs budgets and logs even more.

Anti-spam and Sybils: the tax of open protocols

Open write invites spam and farming. Web2 uses central algorithms and humans; Web3 social often uses stake, invites, CAPTCHAs, client filters, reputation scores. Each introduces new centers or new discrimination. **A fully unmoderated global square** and **high-quality conversation** rarely coexist—products must choose explicitly.

Content storage strategies

- Full text on-chain: expensive, permanent, hard to delete (a feature in some cases).
- Off-chain storage + on-chain hash: cheaper; depends on storage networks or companies.
- Relay-only: faster; durability follows relay policy.

Creators should ask: which copy is the “canonical” one you care about?

Honest creator economics

Token tips, fan tokens, content NFTs all tell stories. Sustainable creator income still mostly comes from sponsorships, memberships, ads, live events—on-chain tools are amplifiers, not automatic financial freedom. Locking fans into one speculative coin collapses communities when the coin collapses.

Privacy

Pseudonymous addresses can still be clustered. On-chain social graphs make follows more transparent—and easier to surveil. When you need privacy: separate identity addresses, reduce on-chain social binding, learn advanced ZK/mixnet tools (learning cost applies).

Product selection flow

1. Do I want “anti-giant” or “a more fun feed”?
2. Can I export or recreate relationships in a second client?
3. If keys are lost, is social recovery possible?
4. In moderation fights, where do I stand—choose servers/protocols that match values.
5. Did I connect my main wallet to a high-risk experimental client?

Crossings with news / governance

DAO votes, forums, Snapshot-class tools often link social identity. Watch **vote buying, farming, concentrated delegation**. On-chain governance is not automatic democracy—only a countable rule set.

Moderation, values, and the “protocol neutrality” myth

Protocols claim neutrality; clients still filter; relays still refuse; servers still ban. Users want **predictable rules** and **migratable exits**, not abstract absolute neutrality. Joining a federated server or a client’s default blocklist is already taking a side—normal. What matters: are rules public, abusive, and can you leave with relationships?

Another reality: celebrities and institutions may still prefer more centralized, better-moderated products for brand safety. Web3 social will not auto-become the only public sphere; it will be one among many. Accept that, and debate stops being a religious war.

Builders: ship high-quality tools for small communities before “global replace.” Users: experiment on alts; bind main identity carefully.

FAQ (social)

Q: Wallet login = decentralized?
A: Only a login method; content and moderation may still be centralized.

Q: Server dies—then what?
A: In federation, backups/migration matter; choose servers carefully.

Q: Lost Nostr private key?
A: Identity is hard to recover; multi-device strategy and social recovery if the client supports it.

Q: Connect main wallet to social DApps?
A: Prefer dedicated addresses.

Q: On-chain posting too expensive?
A: Most designs put content off-chain or on L2.

Product metrics for protocol social

Do not measure Web3 social only by token price. More informative: how many daily addresses are real interaction, client-switch rates, migration success, abuse rates, creator

repeat-post rates, whether people use the identity for real-world ticket checks. Hard to measure ≠ ignorable—if you cannot measure, do not claim “already replaced.”

Governance communities that mix forums and votes must design against bribery and apathy: delegation, optimistic governance, one-address-one-vote vs one-person-one-vote each have pits. Bind social identity too tightly to voting power → bought accounts; too loosely → Sybils. No perfect formula—only revised rules with public revision logs.

For ordinary people, a practical strategy: **keep main social where you already know the network; experiment with alts and dedicated wallets**. Raise binding only after a protocol proves migration and moderation. Moving every social graph onto an experimental protocol is like putting every deposit into an experimental contract—brave, mismatched risk.

Client competition and open graphs

If graphs truly live on open protocols, clients should compete like email clients: better notifications, anti-spam, creation tools. In reality liquidity may still concentrate in one or two frontends because discovery has strong network effects. Open protocols lower the legal/technical ceiling of lock-in; they do not erase economic incentives to lock. As a user, occasionally try a second client—exercise for exit rights.

Developers worry about index cost and write cost: full-chain indexes are expensive; self-built indexes become new centers. Public community indexes, verifiable sync, light-client proofs are ongoing engineering. You need not build them—but understand when your timeline “can’t see latest” whether it is relay, index, or censorship. Misclassify the fault and you may recklessly rotate seeds or approve junk.

Finally, when social mixes with money stimuli, relationships financialize: follows become investments, replies become market moves. Sensible separation—e.g. dedicated “finance identity” vs “chat identity”—reduces personality eroded by price. Not a sermon; mental-health risk management.

Modular identity: from one giant profile to composable proofs

The likelier future is not “one decentralized Facebook wins,” but composable identity modules: login, reputation proofs, verifiable credentials, encrypted inbox from different protocols; frontends orchestrate. UX still decides winners. SCDO’s verifiable-record emphasis fits the “reputation/achievement” module—not rewriting the entire feed. Builders: pick a narrow cut and go deep. Users: avoid binding every module to one hot wallet.

Crisis comms and rumors

Decentralized social in crises is fast and messy: official and fake statements run in parallel; screenshots leave context. Projects should pre-stage multi-channel authenticity (on-chain notes, known-key signed notices, mirrors). Users should wait for verifiable statements instead of instantly forwarding spicy screenshots. That habit protects community assets better than any anti-phishing plugin.

Patch summary

Protocol social’ s decisive factors are migration cost and anti-abuse—not slogans. Alts first, binding later; second client before full migration.

Chapter wrap

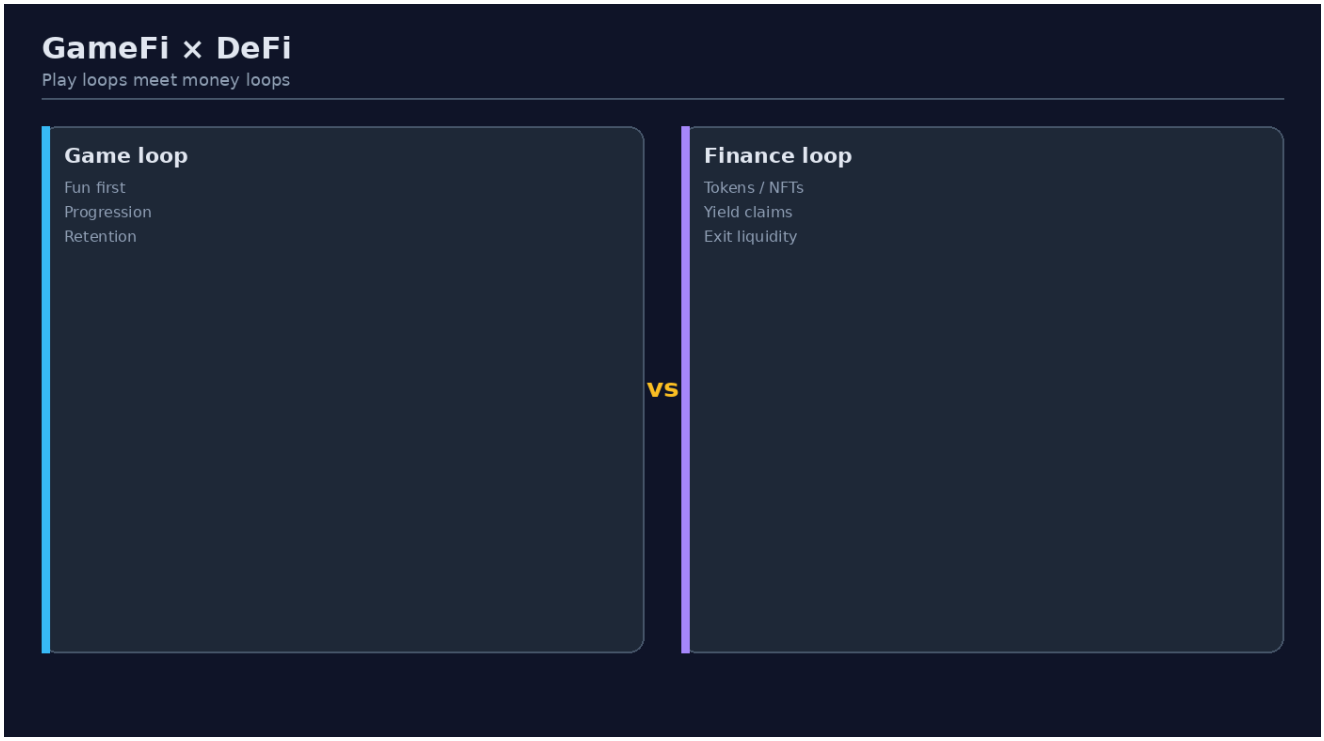
1. Web3 social fights for portability and fewer single points—not another skin.
2. Read in layers: pretty apps ≠ open protocols.
3. Federation has admins; Nostr relies on keys and relays; on-chain graphs have fees and governance assumptions.
4. 2026 is still an experiment field; start with alts.
5. Next: **GameFi**—motives for on-chain game economies, failure modes, industry map—without “explosive growth” hype.

(End of Chapter 7 · v2 English edition · SCDO Lab)

8. GameFi

SCDO Lab · Web3 Primer for Newcomers v2 · English edition
华语版 / Chinese: `08-gamefi.md`
GameFi ≈ games + financial incentives (token/NFT economies). Fun and "earn" often fight.
No "explosive growth" tone.
Not investment advice.

How GameFi relates to DeFi



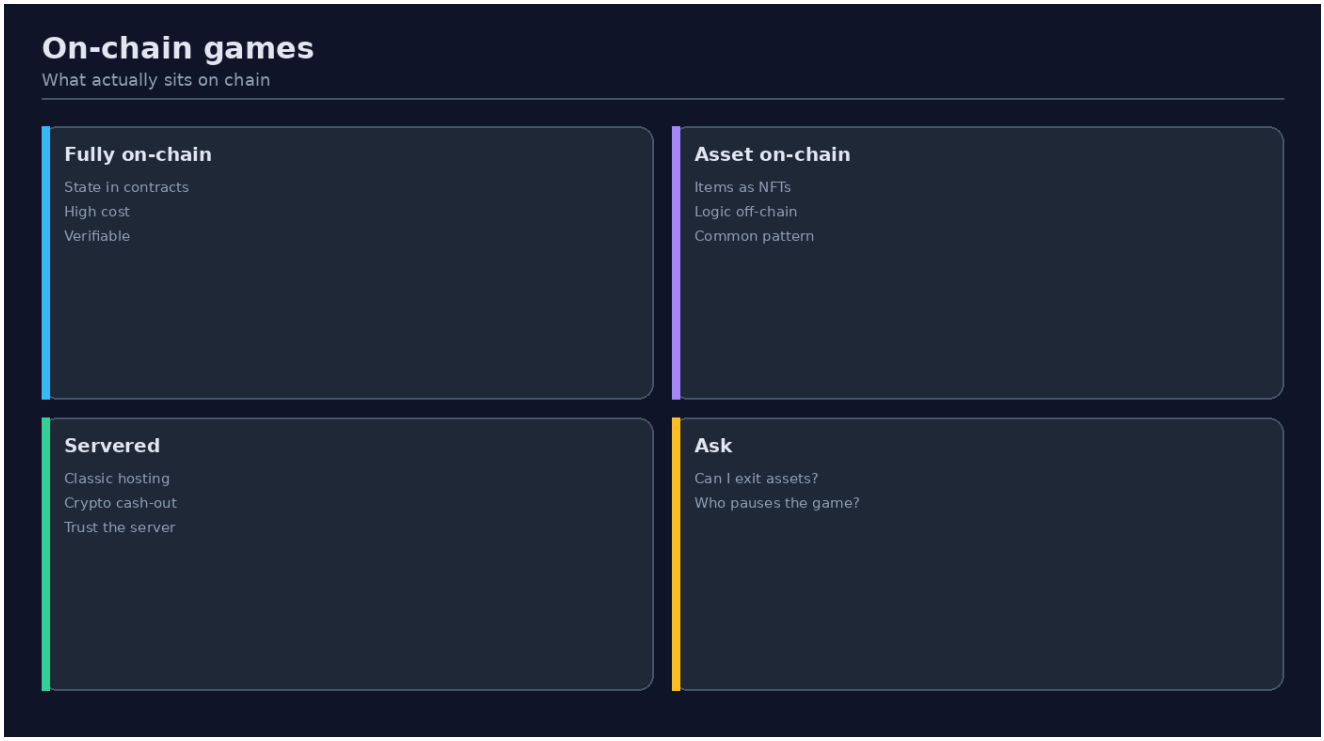
- **DeFi:** financial Lego itself (swaps, lending, market-making).
- **GameFi:** game loops (quests, combat, progression, land) drive people to hold, spend, and trade tokens/NFTs; under the hood often reuses DeFi pools, markets, and bridges.

Plain: DeFi is the exchange and pawnshop; GameFi wants "play while participating in an economy." When it fails, it looks like a game but the core is high-inflation Ponzi—telling them apart matters.

Tiny origin

Early chain games started from collectibles and simple play; a "play-to-earn" phase pulled in labor-style players—and collapses. The industry later talked more about **Play-and-earn** / **Play-to-own**, off-chain play + on-chain assets, and "fun first, tokens later."

Why put games on-chain?



Reasons that can make sense:

1. **Portable assets:** items/characters are not solely one server DB’ s truth (ideal case).
2. **Transparent trading:** secondary market rules more public (fakes and phishing remain).
3. **Open economies:** third parties build tools and markets (also bots and extraction).
4. **Verifiable randomness / settlement:** some genres need publicly checkable outcomes.

Reasons that do not:

- “On-chain automatically more fun”；
- “Issuing a token cold-starts eternal inflation prosperity”；
- “Metaverse land must appreciate.”

Common types

Type	Trait	Risk sketch
Labor / P2E	Time → tokens	Token dump; not enough newcomers → collapse
Collectible competitive	NFT characters/skins + matches	Floor bubbles, match quality
UGC sandbox	User content / land	Traffic and tooling barriers

Type	Trait	Risk sketch
Off-chain play + on-chain settlement	Feel in traditional clients	Bridge and custody complexity
Casino skins	High-odds “games”	Regulation and fairness questions

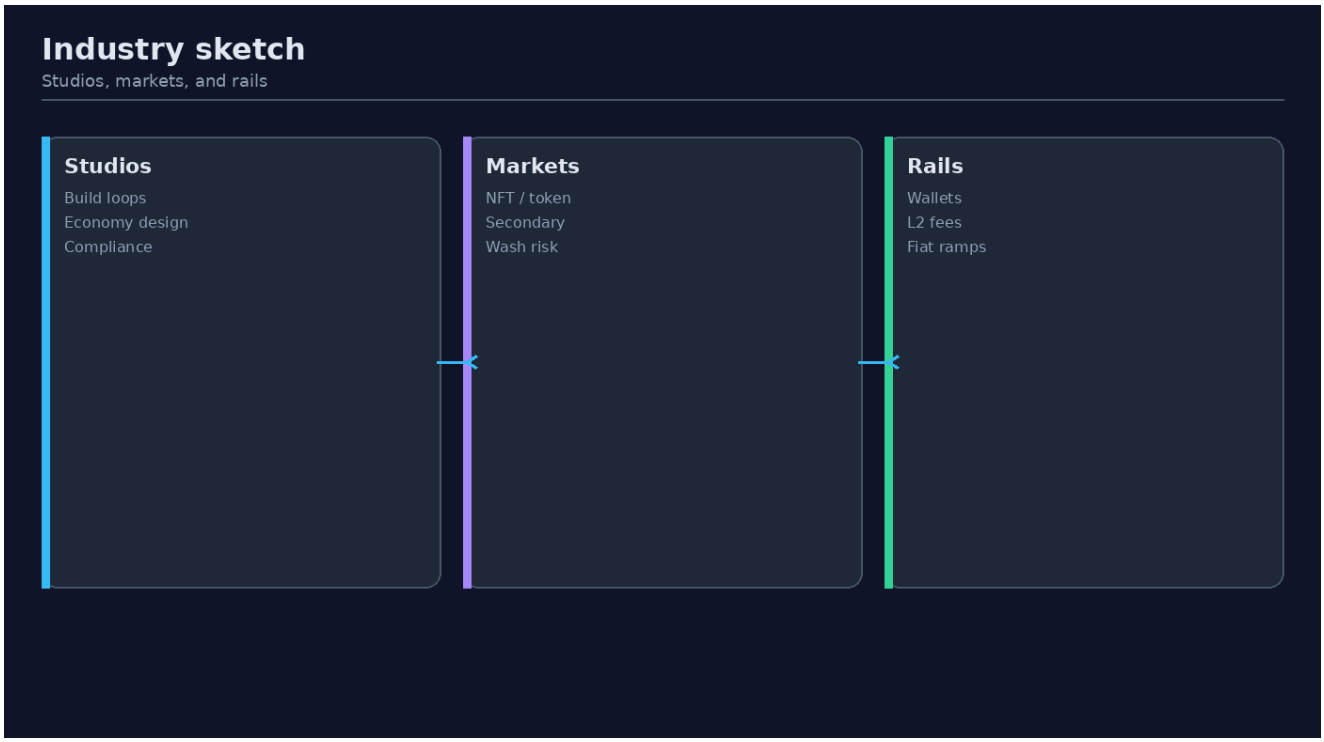
Sustainability: watch death spirals first

Healthy games: fun → voluntary spend/trade → economic balance.
Fragile GameFi: reward tokens → sell pressure → prices crash → players leave → only charts remain.

Design questions:

1. Without token rewards, would anyone still play?
2. Do sinks keep up with sources?
3. Do guilds/studios monopolize production?
4. Are assets utility or pure speculation tickets?

2026 industry map (schematic, no name-dropping shills)

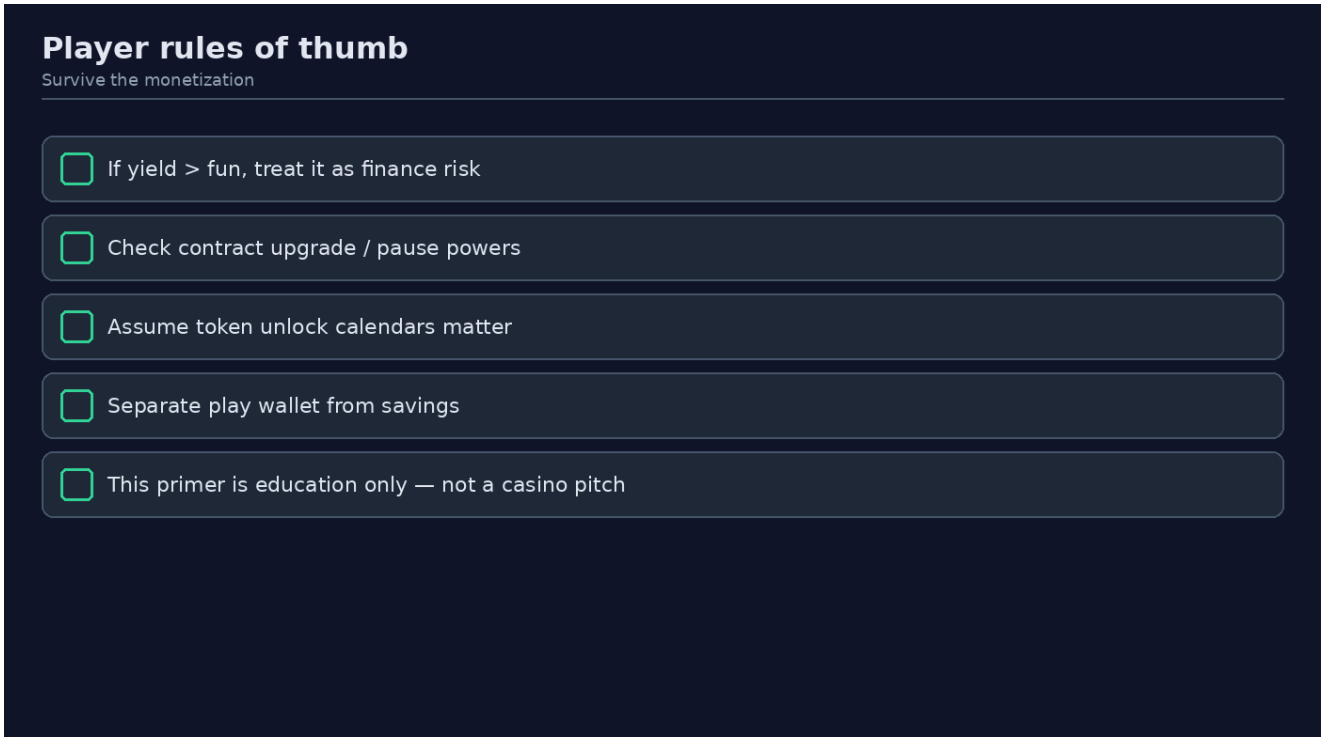



```
Players / guilds
    ↓
Game clients & engines ← publishers / studios
    ↓
On-chain assets (NFTs/tokens) ← markets, bridges, wallets
```

```
    ↓
Public chains / L2s / app chains
    ↓
Liquidity & financial services (swaps, lending—be careful leveraging “earn”)
```

Periphery: analytics, anti-cheat, security audits, compliance, creator tools.
Name lists go stale; learn roles, not last year’s “must-play” charts.

Newcomer rules



1. Try play before talking earn.
2. Stress-test P2E income as able to go to zero.
3. Approve game contracts sparingly; dedicated burner wallets.
4. Watch “guaranteed daily yield” and “insider blind boxes.”
5. Mind local gambling and token-game regulation.

Economy design vocabulary

- **Source:** quest rewards, drops, mining.
- **Sink:** repair, crafting, tickets, burns.
- **Soft/hard caps:** is inflation bounded?
- **Seasons:** periodic resets ease permanent inflation.
- **Dual/multi-token:** separate governance and game currencies—complex, can confuse players more.

Good design is city traffic: inflows and outflows; bad design is flyer-style money printing.

Guilds, studios, and labor

In P2E eras guilds organized labor, split devices and accounts. Efficient—and turns individuals into pyramid bottoms. Before any “team coaching,” clarify contracts: who holds keys, how revenue splits, who owns security on exit.

Common technical architectures

1. **Fully on-chain:** transparent but costly/slow—strong settlement, weak twitch feel.
2. **Big off-chain world + on-chain assets:** better feel; trust servers + bridges.
3. **App chains:** custom performance; build your own security and liquidity.

No silver bullet. Ask what claim path exists if theft or shutdown happens.

Regulatory radar

Some designs approach gambling or securities issuance. Definitions differ by country. Before launch or mass acquisition, get legal advice—this book only educates; it does not provide evasion structures.

Healthy participation checklist

- Budget: entertainment money, not rent.
- Wallet: dedicated.
- Time: anti-addiction; P2E easily becomes another job.
- Info: official channels; Discord account theft is common.
- Exit: predefine “stop after X% drawdown.”

From player to builder boundaries

Studios: ship a fun 30-minute vertical slice before token design. Economy advisors and gameplay designers at the same table—not tokens first, gameplay later. Guilds: bake security training into policy; never require members to hand over seeds. Investors (high risk): treat GameFi tokens like high-volatility early-venture substitutes—not wages.

Content pipelines (art, levels, live ops) do not vanish because of chains. Blockchains replace some asset/settlement assumptions. Ignore content cost and you are left with price charts in a browser.

If SCDO-related experiments exist, sell fair competition, verifiable scores, and entertainment itself—not “guaranteed earn.” Business-development pitches stay out of this book.

FAQ (GameFi)

Q: Can P2E still work?
A: Treat as high-risk short labor; default the yield curve sloping down.

Q: Game NFT floor crashed?
A: First ask if anyone still plays; prices without gameplay rarely recover.

Q: Guild wants my seed phrase?
A: Refuse—hard red line.

Q: Must chain games have tokens?
A: No; Web2 accounts + optional on-chain assets first is fine.

Q: How to tell if a studio is serious?
A: Update cadence, playable builds, economic transparency—not roadmap adjectives.

Content pipelines, anti-cheat, and the livestream era

If chain games depend on streamers for acquisition, economies get “exposure pulses”: a wave arrives, then dumps. Seasons and regional servers can smooth pulses but also fragment liquidity. Anti-cheat is harder when bounties exist: bots, multi-boxing, studio farming push device fingerprints or KYC—which collide with “permissionless” narratives. Teams must decide early: fear studios more, or fear ideology users lost to KYC? Pleasing both usually disappoints both.

Art/music copyright, mod ecosystems, UGC revenue share decide whether sandboxes live. Automatic on-chain royalty splits sound nice; they still meet fiat tax and child-safety rules. Old game-industry problems do not vanish under Solidity. Chains are good at hard settlement and open markets; bad at making levels automatically fun.

Review mantra: open the white paper for the “fun loop diagram” first—if missing, walk away; if token allocation appears before fun loops, it is usually a financial product in a game costume.

Seasons, regions, and experiment ethics

Seasons let designers reset inflation but may punish long-term collectors; mitigations include cross-season cosmetics, soulbound achievements, or inheritable record credentials. Regional servers cut latency and compliance complexity but split liquidity and guild maps.

When publishing economic experiments, act like drug trials: hypothesis, sample, failure criteria, rollback policy. Treating players as guinea pigs without notice scores ethical and commercial demerits.

Esports-style chain games must handle prize custody and cheat disputes: multisig or contract for prizes? centralized refs? how are VODs and RNG seeds published? Fail those and pretty tokens are merch only.

External analyses of P2E-labor-heavy regions often ignore FX, power, and device depreciation. If your model needs forever-cheap labor somewhere, the model itself is fragile. More sustainable: local players pay for fun; labor output is fringe—not the reverse.

Data, matchmaking, spectating

Unfair matchmaking kills competitive chain games; verifiable RNG reduces “black-box drop” suspicion but does not fix matchmaking Elo itself. Spectating needs delay, caster rights, highlight copyrights—almost all off-chain. Minting “verifiable scorecards” for players and viewers is where chain can add; forcing every input frame on-chain is a performance disaster. Choosing the on-chain boundary marks mature teams.

Milestones from demo to long ops

Internal milestones: playable Demo (no token) → small testnet economy → mainnet limited assets → open economy. Each step needs stop conditions: retention below threshold, inflation above threshold, major security incidents. Public milestones filter partners who only want “one-click token.” Long ops need support, anti-cheat shifts, event calendars—budget those FTEs or “sudden update silence” rises. On-chain treasury transparency can be monthly: revenue, burns, balances—browser-checkable. Transparency does not guarantee success; opacity almost guarantees distrust.

Token allocation and unlock calendars

Read GameFi projects by allocation first: team, investors, ecosystem, play-earn budgets; whether unlocks tie to content milestones. Front-loaded unlock dumps are normal risk. “Fair launch” narratives without calendars still need liquidity and market-making scrutiny. Treat allocation tables as arsenals that can hurt player experience—designers use carefully; players enter carefully.

Chapter wrap

1. GameFi grafts games and finance; fun and earn often conflict.

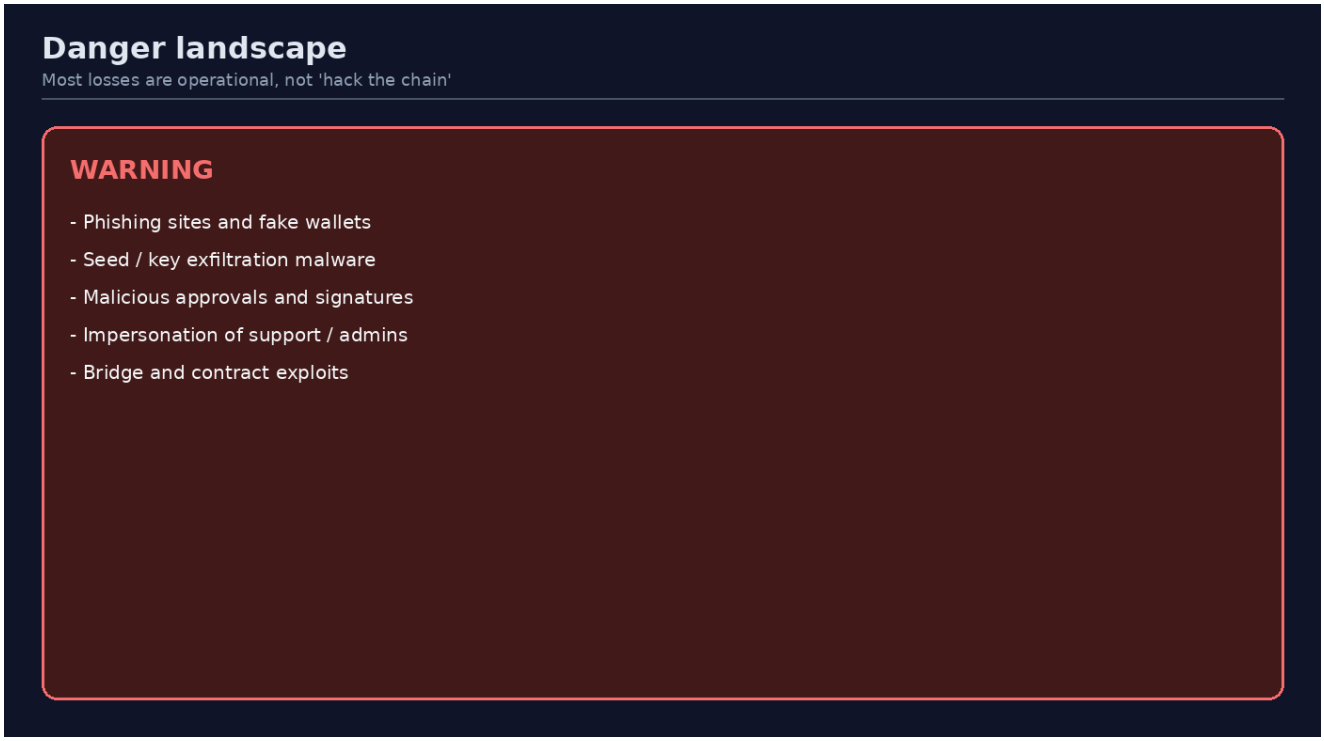
2. Chains serve assets and transparency; they do not automatically raise fun.
3. Watch inflation and sinks; do not believe eternal P2E.
4. In 2026 read industry roles, not stale charts; respect compliance and education boundaries.
5. Next: **security**—anti-theft, phishing, and allowances in the dark forest—more life-saving than chasing airdrops.

(End of Chapter 8 · v2 English edition · SCDO Lab)

9. Security always on

SCDO Lab · Web3 Primer for Newcomers v2 · English edition
华语版 / Chinese: `09-security.md`
On-chain life is often called a “dark forest”: assume someone wants to trick your signature, allowance, or seed export.
No third-party vendor long-form dumps; no frozen “top 11 plugin install” lists. Principles and habits. **Not investment advice.**

Why especially dangerous?



1. **Hard to reverse txs:** wrong often stays wrong.
2. **Signatures are powerful:** one blind sign can empty a wallet.
3. **Fake sites are cheap:** ads, cloned UI, lookalike domains.
4. **Social engineering upgrades:** fake support, fake jobs, romance, fake AI assistants.
5. **Composition risk:** bridges + allowances + malicious tokens + phishing chain together.

Goal is not to scare you off—it is **muscle memory**.

Attacker map

Type	Looks like	Defense gist
Phishing sites	Search ads / fake links that look real	Type official URLs, bookmarks, check domain/certs
Fake airdrop/mint	“Free claim” but needs approve	Read allowance size and contract
Fake wallets/extensions	Clone apps	Official channels only; verify publisher
Fake support	DM “help refund/unfreeze”	Officials never ask for seed first
Address poisoning	Similar addresses in history	Check prefix/suffix every time; use address books
Malicious RPC / Wi-Fi	Network hijacks	Avoid strange networks for sensitive ops; hardware wallets
Contract backdoors	Team or dependency malice	Limits, split wallets; audits are not gospel
NFT scams	Stolen art, fake floors, quote signatures	Do not random-sign; understand listing signatures

NFT-specific pits

- Signatures that look like “offers” actually authorize transfers;
- Free mints demanding unlimited allowances on major stables;
- Discord overnight filled with stolen-account links.

Phishing’s common cards

Urgency, fear, greed. Anything “countdown must sign” gets a cold pause first.

Defense principles (more important than plugin lists)

1. **Seeds / private keys never typed online**, never sent to anyone, never cloud-album screenshots.
2. **Large cold, small hot**; games/experiments on burn addresses.
3. **Minimize allowances**; revoke regularly with tools you verified yourself.
4. **Bookmark official sites**; do not click the first search ad.
5. **Hardware wallets** for savings-scale assets.
6. **Simulate/preview** before send if the wallet supports it; refuse if unclear.
7. **Split holdings:** one address should not do everything.
8. **Assume DMs are fake**—especially “support.”
9. **Upgrade and backup drills:** restore-test small amounts before device swaps.
10. **Law and platform process:** recovery paths differ by region; set expectations early.

How to choose anti-phishing tools (principles, not install ranks)

- Open source or at least explains detection logic?
- Permissions oversized (read every page)?
- Updates frequent; community reports of false/missed positives?
- Asks for your seed? **Discard immediately.**
- Domain cross-checked against docs and git?

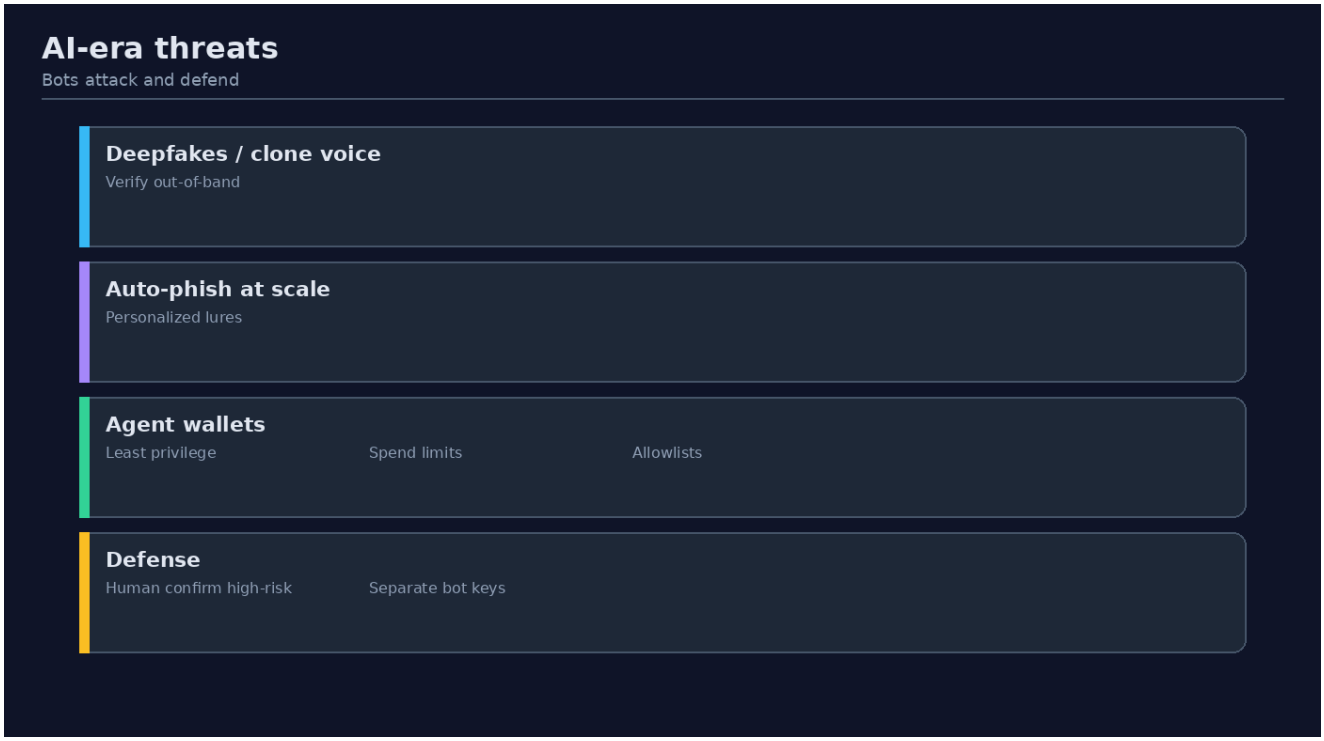
Tools assist; **they do not replace reading signatures**.

Risk classes (principle paraphrase)

Chain/contract layer: bugs, oracles, bridges, upgrade keys.
Client layer: fake wallets, malicious extensions, clipboard malware.
Human layer: greed, fear, blind KOL trust.
Ops layer: rugs, insider multisig, fake audits.

Response: limits, separation of duties, timelocks, multisigs, monitoring, limited insurance—and most importantly, **do not join what you do not understand**.

One AI-era line



Automated agents can walk steps on your signing path. Give agents intern-level permissions: budgets, allowlisted contracts, revocable, full logs. Aligns with SCDO’ s auditable-record emphasis—**speed must not erase traceability**.

After an incident (principles)

1. **Immediately:** disconnect affected devices; revoke allowances from a clean device; move remaining funds to a new-seed wallet.
2. **Evidence:** tx hashes, phishing domains, chat screenshots.
3. **Police / platforms:** follow local and platform process; do not assume recovery.
4. **Postmortem:** which blind sign? which search ad?
5. **Do not** click the next “asset recovery” scam chain.

Family and team

Shared treasuries: multisig and role split—propose, approve, execute separated. One person in all roles is a single point. Train: support-script recognition, hardware ops, emergency contact tree.

Minimal developer security set

- Checklists and peer review;
- Testnets and invariant fuzzing;
- Limit upgrade powers and timelocks;
- Dual oracle sources;
- Emergency pause and disclosure channels;
- Do not copy mystery “get rich” contracts.

Audit reports are not immunity cards: scope, version, and whether findings were fixed matter.

Physical and supply chain

Fake hardware wallets, pre-backdoored used devices, repair implants—large holders buy official, verify packaging, generate seeds themselves. Rooted/jailbroken phones lower hot-wallet assumptions.

Psychological safety

Shame keeps victims silent; scammers win more. A culture where “almost got phished” is discussable beats five more plugins.

Printable weekly checklist

Weekly:

- [ ] Review allowances; revoke unused;
- [ ] Confirm bookmarks still hit real domains;
- [ ] Update wallet and OS patches;
- [ ] Check backup hardware/paper still readable;
- [ ] Rehearse one “fake support” script.

Monthly:

- [ ] Tiny restore drill to a clean device;
- [ ] Inventory which addresses hold experimental contracts;
- [ ] Read one recent major hack postmortem (learn patterns, not imitation).

Yearly:

- [ ] Consider hardware firmware upgrades (official channels);
- [ ] Family emergency: contacts know you never ask them for codes, and shared “no-transfer passphrase.”

Tape the list to the desk—beats bookmarking a hundred security essays.

FAQ (security)

Q: Typed seed into a fake site?
A: Treat as compromised; move assets to a brand-new seed; abandon the old address.

Q: Does hardware make me 100% safe?
A: Cuts hot-end risk a lot; does not stop you signing malicious txs yourself.

Q: Wallet on public Wi-Fi?
A: Prefer not; if must, hardware confirm + tiny size.

Q: Family asks remote help to transfer?
A: Anti-scam passphrase; video confirm.

Q: Revokes too expensive?
A: Prioritize high-risk unlimited allowances; wait for cheaper fee windows when you can.

Org threat model (small teams too)

Even a three-person studio can write a one-page threat model: asset classes (treasury, hot wallets, domains, social accounts, repos), adversaries (external hackers, phishing, insiders, vendors), worst impact, existing controls, gaps. Stolen domains/social for fake

announcements often harvest fans faster than contract bugs. MFA, least privilege, hardware keys for admin accounts—high ROI, low cost.

Repo dependency poisoning, CI leaking keys, testnet seeds reused on mainnet—classic own-goals. Checklist: dedicated secret storage; short-lived CI tokens; test/prod isolation; dedicated machines for contract deploys. Pre-write disclosure templates: when found, scope, what users should do, whether compensation exists. Silence destroys the only thing pricier than money—trust.

Individuals shrink “org-level” to family: who inherits wallets? is there a careful estate note (that does not become a leak)? do emergency contacts know you never ask them for verification codes? Awkward topics—more controllable than on-chain inheritance fights.

Simulated attack drills (yearly)

A trusted friend runs a half-day “red team”: phishing pages, fake signatures, fake support scripts (agree: no real large transfers). Fix process, not friendships afterward. Companies can hire outsiders; individuals can use family meetings. Findings: confusable domains, missing bookmarks, hardware confirms ignored into blind signing.

Also drill “lost phone”: biometric second gates on hot wallets? exchange withdrawal allowlists? can social accounts be recovered via backup email? Security is a process, not graduating by buying one device.

If AI assistants manage calendars or even draft txs, never hand seeds or unlimited permissions to cloud chat. Treat assistants as strangers on the internet—useful and dangerous—until you can attach audit logs and budgets to their actions.

Travel and cross-border

Carry hardware carefully; split paper backups; never put seeds in checked luggage; avoid sketchy public charge ports (charge attacks); hotel safes ≠ safe. If exchange allowlists bind a home phone number, ensure 2FA still works while roaming. Drill before the airport. Follow local declaration and FX rules—this book does not discuss evasion.

Three-second rule before every signature

Force three seconds: 1) which domain? 2) does the contract want my money or only a verify-sign? 3) did I initiate this? If any answer fails, refuse. Teach new teammates this rule—blocks many reflex thefts. Pair with wallet simulation (if any) and hardware screen confirms for defense in depth. Security is a speed bump, not a brake that ruins the trip; slowing down lets you drive farther.

Clipboard and QR codes

Mobile malware can swap copied addresses; always re-check on-screen prefix/suffix. QR stickers can be faked; large payments use allowlisted address books. Desktops can have malicious extensions watching clipboards—audit extension permissions strictly.

Chapter wrap

1. Assume adversaries are phishing; reversals are hard.
2. Keep seeds secret, minimize allowances, type official URLs.
3. NFTs and airdrops are hot zones.
4. Choose plugins by principles, not stale charts.
5. Next: **closing**—learning path, disclaimer, light SCDO extension, how updates happen.

(End of Chapter 9 · v2 English edition · SCDO Lab)

10. SCDO: sharded PoW, sites, and how newcomers try it

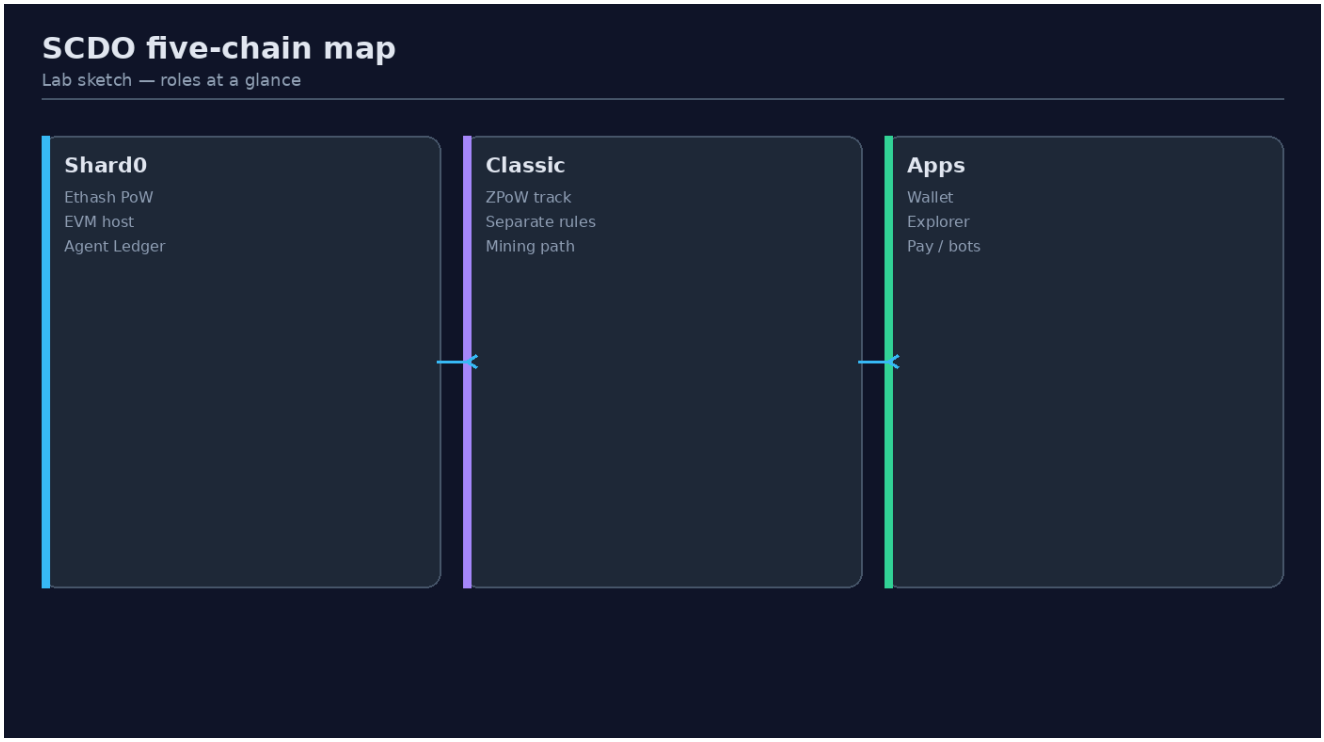
> SCDO Lab · Web3 Primer for Newcomers v2 · English
> Chinese: `10-closing.md`
> Chapters 1–9 gave a general map; this chapter focuses on **SCDO**: what it is, how five chains share work, where the site/explorer/wallet/Agent Ledger live, how to read compliance, and how to try small steps.
> **Not investment advice.**

What is SCDO?

SCDO is a peer-to-peer public blockchain system for decentralized value transfer, with public sites and tools maintained by **SCDO Laboratory** (operator **9Y9 PTY LTD**). In 2026 the live network is best read as **five peer chains**:

Chain	Consensus / client	Role (educational)
Shard0 EVM	**Ethash** PoW · core-geth family · chainId **5680**	Contracts, Agent Ledger anchors, Ethereum-like tooling
Classic Shards 1–4	**ZPoW** · go-scdo	Original sharded core; addresses often **1S…**

One line: **Classic keeps scientific-computation-style ZPoW shards; Shard0 uses Ethash for EVM and agent bookkeeping.** They are peer product surfaces—not a simple “main vs deprecated” story.



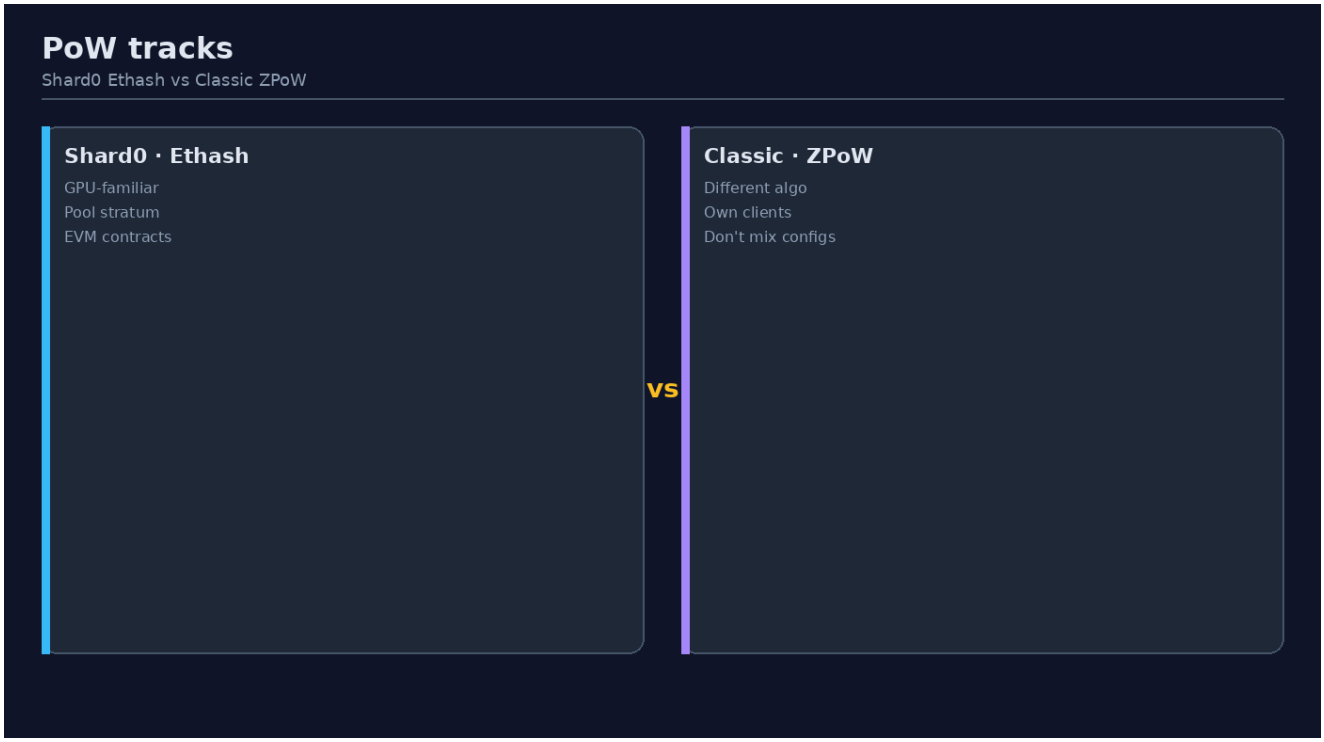
Shards and PoW: two tracks

Classic: shards + ZPoW

- Accounts/nodes partition across Shards 1–4; local txs confirm in-shard; cross-shard uses mechanisms such as **Debt**.
- **ZPoW** targets verifiable scientific-style work (CPU-friendlier design goals vs pure hash races); GPUs are still common in practice.
- Mining detail: Chapter 11.

Shard0: Ethash EVM

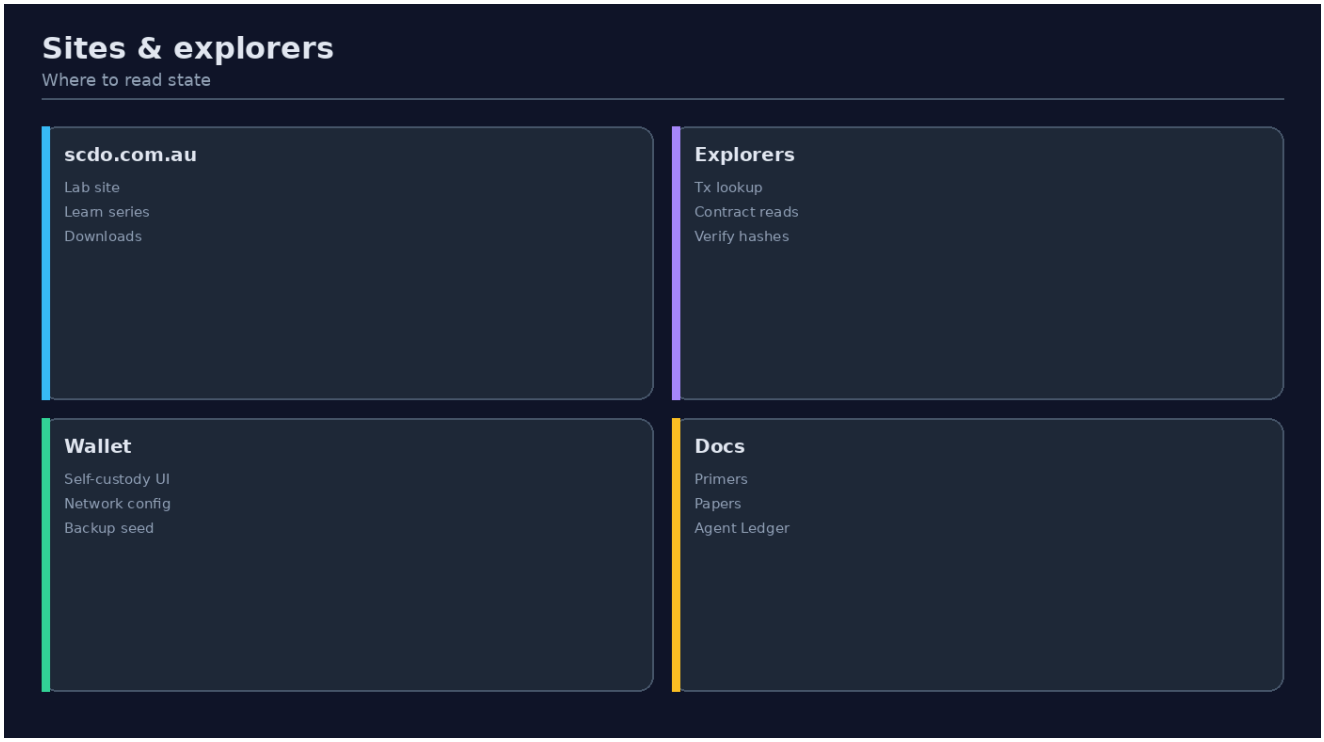
- Pre-merge Ethereum-style **Ethash** epochs; public RPC/explorers follow the official site and scdoscan.io.
- Fits MetaMask-class tools, ERC contracts, and agent identity (e.g. ERC-8004 registries).



Site, explorer, transparency

Entry	Use
scdo.com.au	Home: wallet, Agent Ledger, papers, compliance, learn
scdoscan.io	Explorer and transparency-related pages
`/papers/`	Technical papers / 2026 architecture notes
`/learn/web3/en/`	This primer

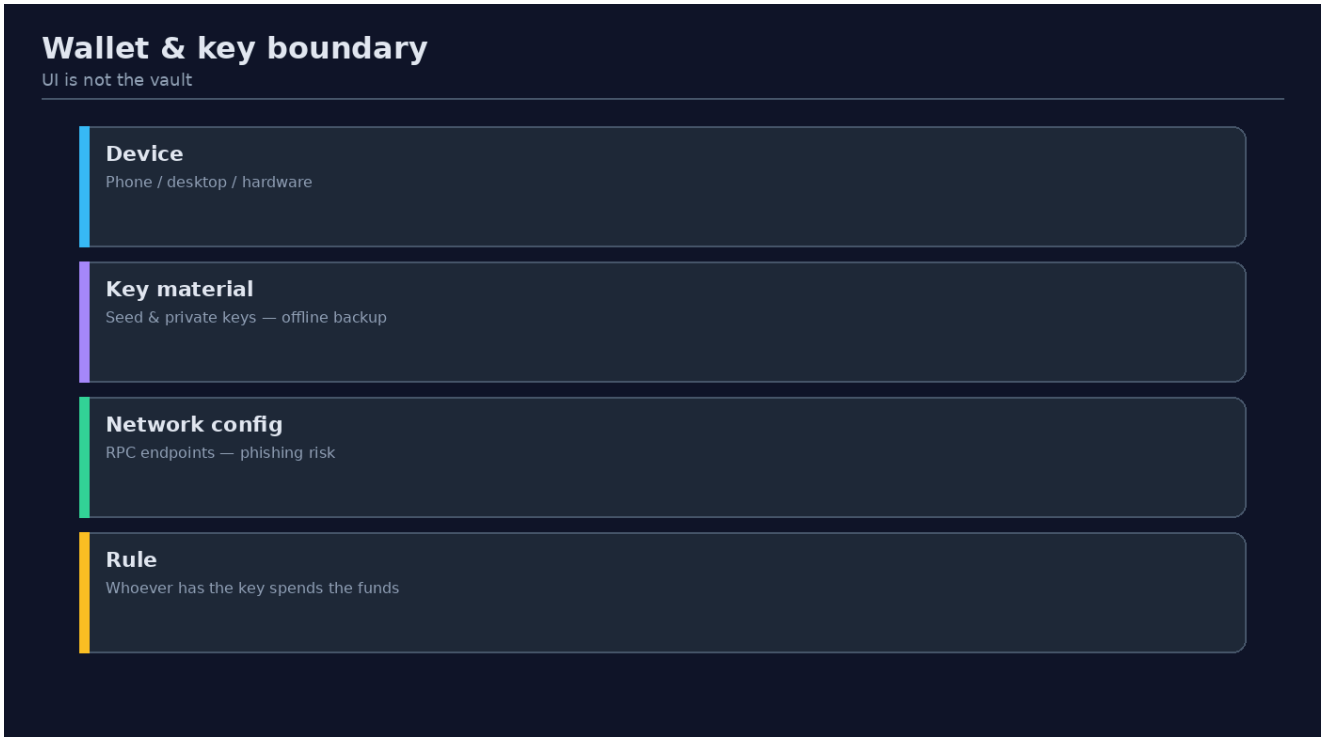
Prefer primary sources over second-hand “it’s over” screenshots.



Wallet (SCDO Wallet)

SCDO Wallet targets a five-chain UI: Classic `1S…` and Shard0 `0x…`; keys stay local per then-current product docs. Headless Linux agents exist for server bookkeeping.

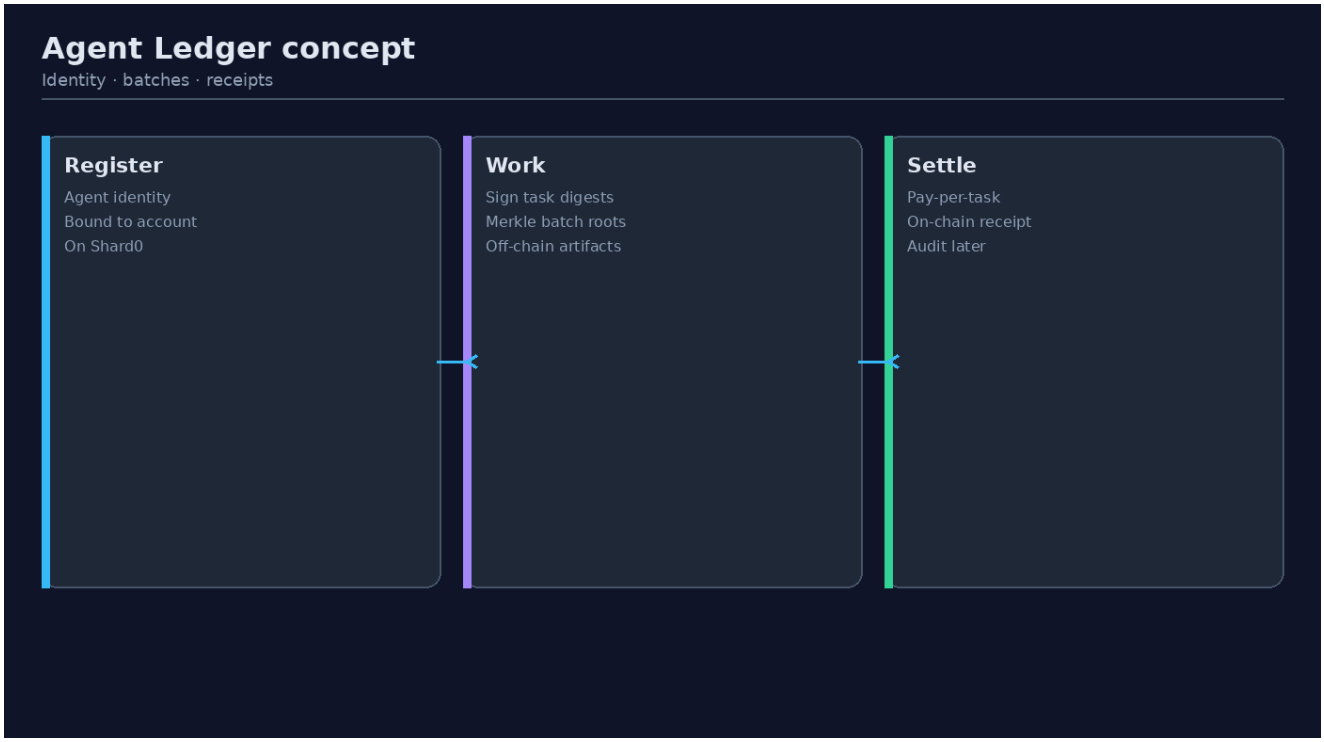
Newcomer rules (same as Ch. 9): tiny amounts first; never paste seeds into chat/cloud notes; download only from official paths.



Agent Ledger: verifiable agent bookkeeping

Agent Ledger lets each AI agent hold its own key, sign tasks/heartbeats, and anchor digests on **Shard0**; optional ERC-8004 identity. Entry: scdo.com.au/agent-ledger/.

It answers **who signed** and **can we check the trail**—not "auto money printer." Payments (x402 / HTTP 402, ~1¢ settle nests) are Chapters 12–13.



Compliance context (high level, not legal advice)

Public materials for operator **9Y9 PTY LTD** include:

- **AUSTRAC** Digital Currency Exchange registration example: **DCE100714503-001** (verify on AUSTRAC's public register; **registration ≠ endorsement**).
- **AFCA** external dispute resolution member example: **124589** (contact the operator first; AFCA may be available if unresolved).

Details: scdo.com.au/compliance.html and regulator pages. This book does not teach evasion.



How newcomers try SCDO (no large size)

1. Read Wallet / Agent Ledger / Papers pages on the official site.
2. Use the explorer for public Shard0/Classic info.
3. If you create a wallet: least privilege, dust-size tests only where lawful.
4. For agents: understand signing/heartbeats before payment pilots.
5. Reread Chapter 9 security.



Link to earlier chapters + business red lines

Chapters 1–9 teach general Web3; this chapter teaches **how to read the SCDO product surface**. Mining, bot commerce, payments, and gambling/compliance angles follow.

Disclaimer

Educational original rewrite — not investment, legal, or tax advice. Assets can go to zero. Obey local law. Third-party names are explanatory, not endorsements. Products follow **then-current** official wording.

Chapter wrap

SCDO = **Classic shards (ZPoW)** + **Shard0 EVM (Ethash)** + **official toolchain (wallet, explorer, Agent Ledger)** + **verifiable compliance disclosure**. Next: Ch. 11 Mining, Ch. 12 Bots×chain, Ch. 13 Payments, Ch. 14 Gambling lens.

(Chapter 10 · SCDO · SCDO Lab)

11. Mining: PoW, miners/pools, GPU/ASIC, and SCDO bookkeeping

> SCDO Lab · Web3 Primer v2 · English · Chinese: `11-mining.md`
> After Ch. 4 shards and Ch. 10 SCDO: PoW basics, miners/pools, hardware paths, and **SCDO’ s two mining tracks** (Shard0 Ethash + Classic S1–4 ZPoW).
> — no entertainment promo. Not investment advice.

What is mining (PoW)?

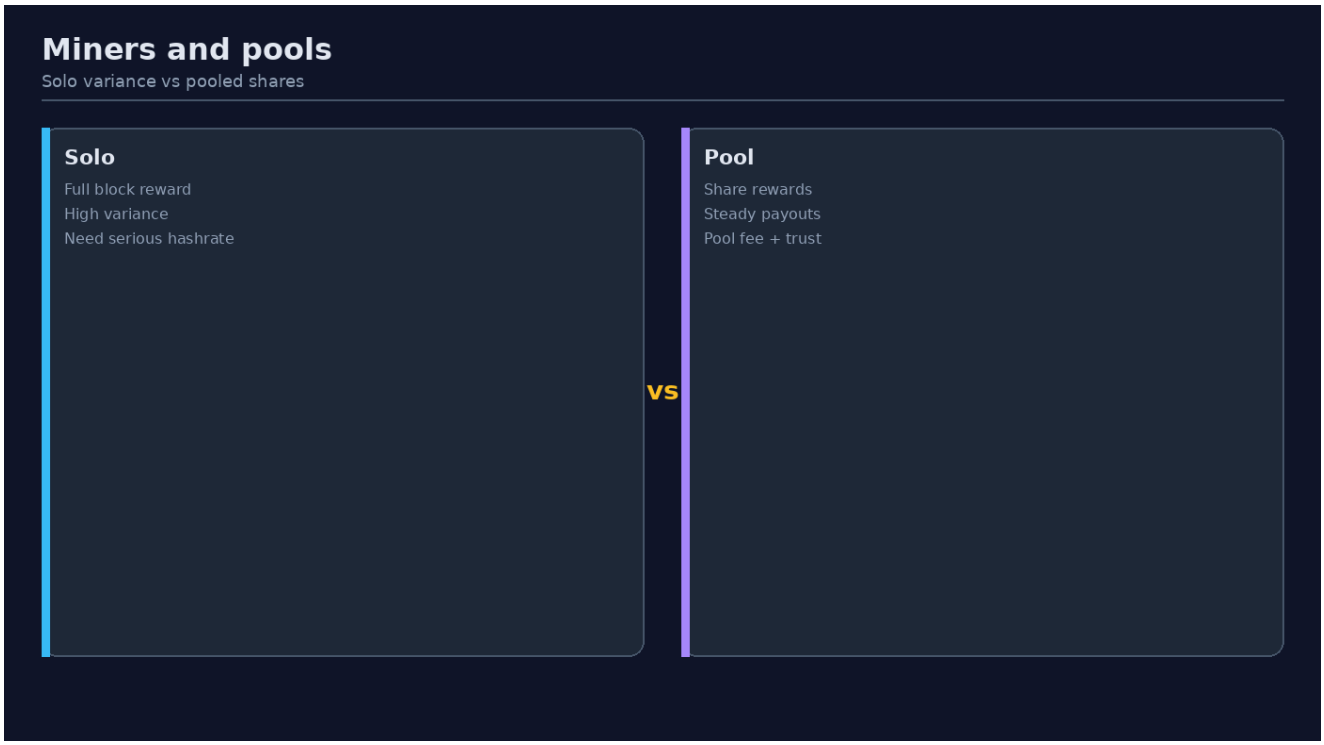
Proof of Work makes nodes compete on a verifiable puzzle; winners produce blocks and earn rewards/fees (rules vary). Mining trades **energy and hardware** for block rights and incentives, and makes history expensive to rewrite.



Miners, full nodes, pools

Role	Job
Full / bookkeeping node	Hold state, verify, may produce blocks
Miner	Search PoW solutions at target difficulty
Pool	Aggregate hashrate; pay by shares (lower solo variance)

Pools cut luck volatility and add **centralization and rule trust**—check fees, payouts, switchability, history.



GPU, CPU, ASIC

- **CPU:** flexible; some scientific-style algorithms.
- **GPU:** workhorse for Ethash-family and many PoWs.
- **ASIC:** algorithm-specific efficiency; prices out casual hardware.

Algorithm choice shapes who can mine: ZPoW vs Ethash imply different hardware ecosystems.



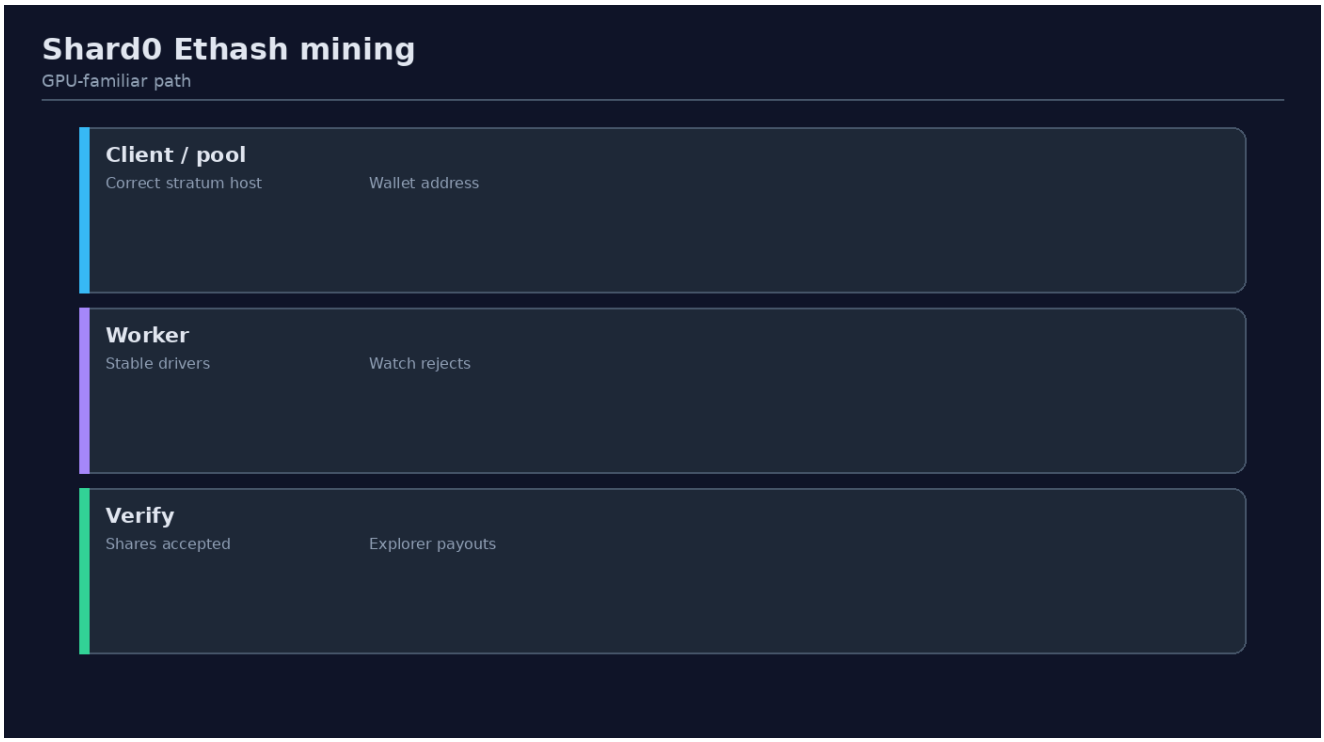
Costs, risks, honest expectations

Mining is a business: power, cooling, downtime, rising difficulty, price swings, pool rules. Losses happen. Policy varies by place. No “guaranteed rig list” here.



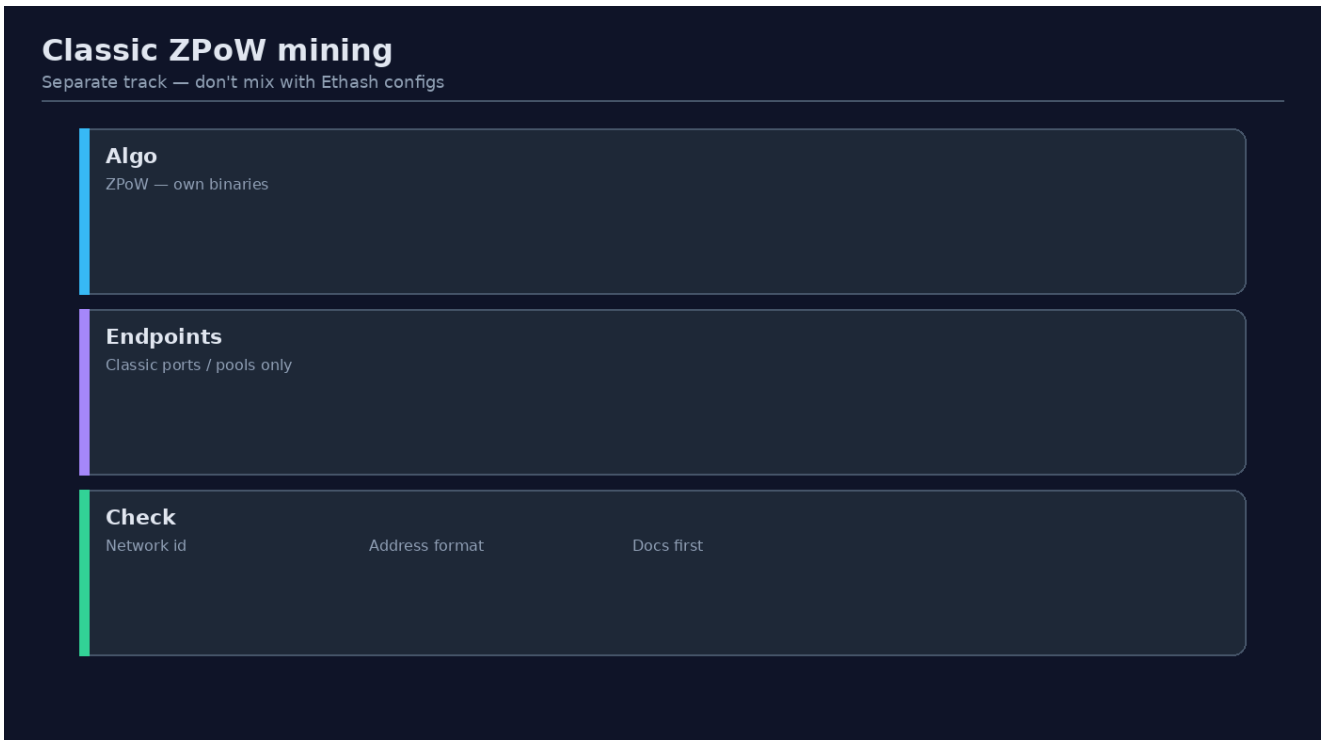
SCDO track A: Shard0 Ethash

- Shard0 EVM · Ethash · chainId 5680.
- Tooling resembles Ethereum-PoW-era GPU stacks; follow official guides and transparency pages.
- Project-operated hashrate may be large—independent bookkeepers help decentralization (per public disclosure).



SCDO track B: Classic Shards 1–4 ZPoW

- Four Classic shards; **ZPoW** consensus.
- Address/wallet paths differ (`1S…` vs `0x…`).
- GPUs common in practice; wallet CPU bookkeeping may exist—per then-current official docs.



Newcomer checklist

1. Know which chain and algorithm you mine.
2. Price power and noise; ignore hype screenshots.
3. Triple-check payout addresses; use trusted pools.
4. Verify miner software hashes.



Chapter wrap

PoW = verifiable compute race; pools trade variance for trust points; hardware follows algorithms. SCDO: **Shard0 Ethash EVM track** + **Classic S1–4 ZPoW shard track**. Next: bots and payments.

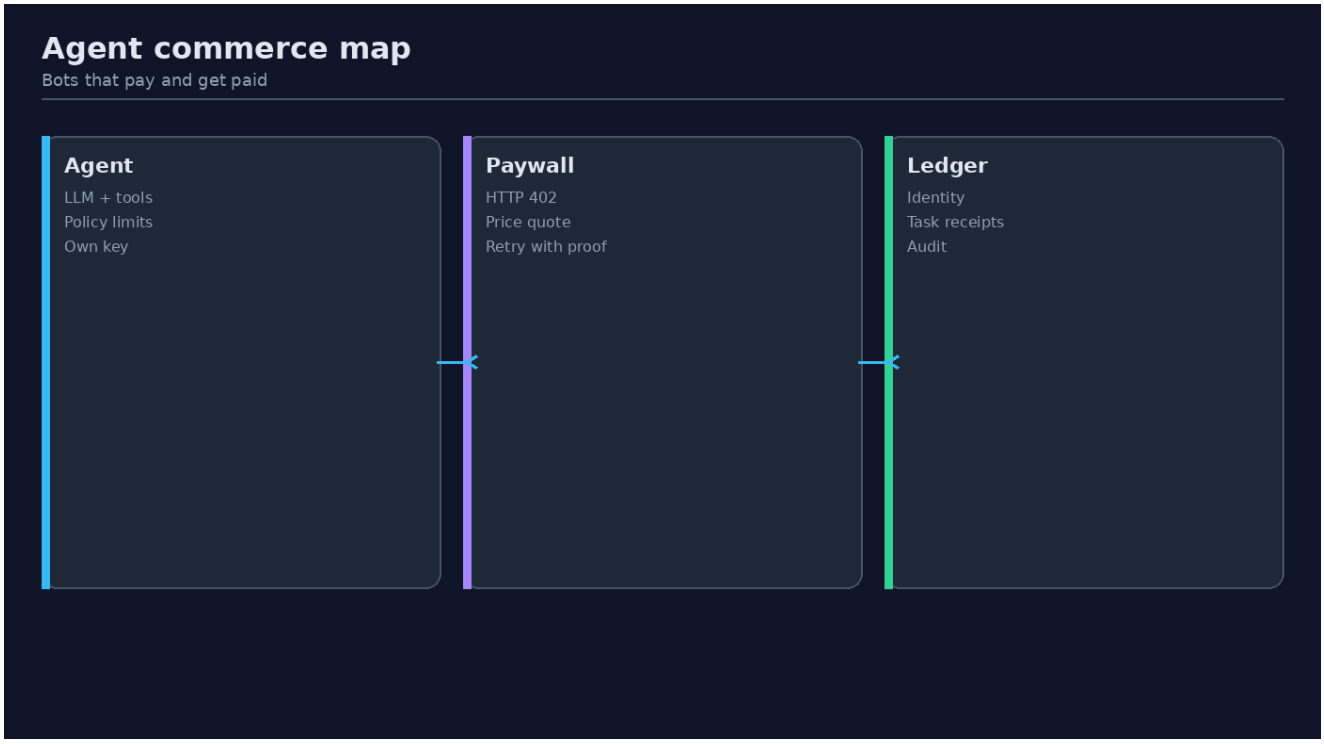
(Chapter 11 · Mining · SCDO Lab)

12. Bots × Blockchain: agent commerce, HTTP 402, SCDO nests

> SCDO Lab · *Automated Contract Ledgers for AI Agents* v2 · ZH: `12-bots.md`
> How bots/agents hold keys, keep books, pay per call; Agent Ledger and ~1¢ settle nests.
> Not investment advice.

Why bots × chains?

Automated agents call APIs, exchange data, hire other agents—they need **identity, permission bounds, auditable trails, micropayments**. On-chain and adjacent rails turn machine commerce from verbal promises into checkable records.



Self-held wallets and ledgers

One key per agent: sign tasks/heartbeats; digests can anchor on SCDO **Shard0**. **Agent Ledger** is the public bookkeeping surface (see Ch. 10). Keys stay on-device per product docs.



HTTP 402 and x402

HTTP 402 Payment Required is the old “pay to proceed” idea; modern **x402** practice wires it to stablecoin per-call pay (SCDO pilots often **USDC on Base**). Fits API calls and agent-to-agent micros—not one-shot buyouts.



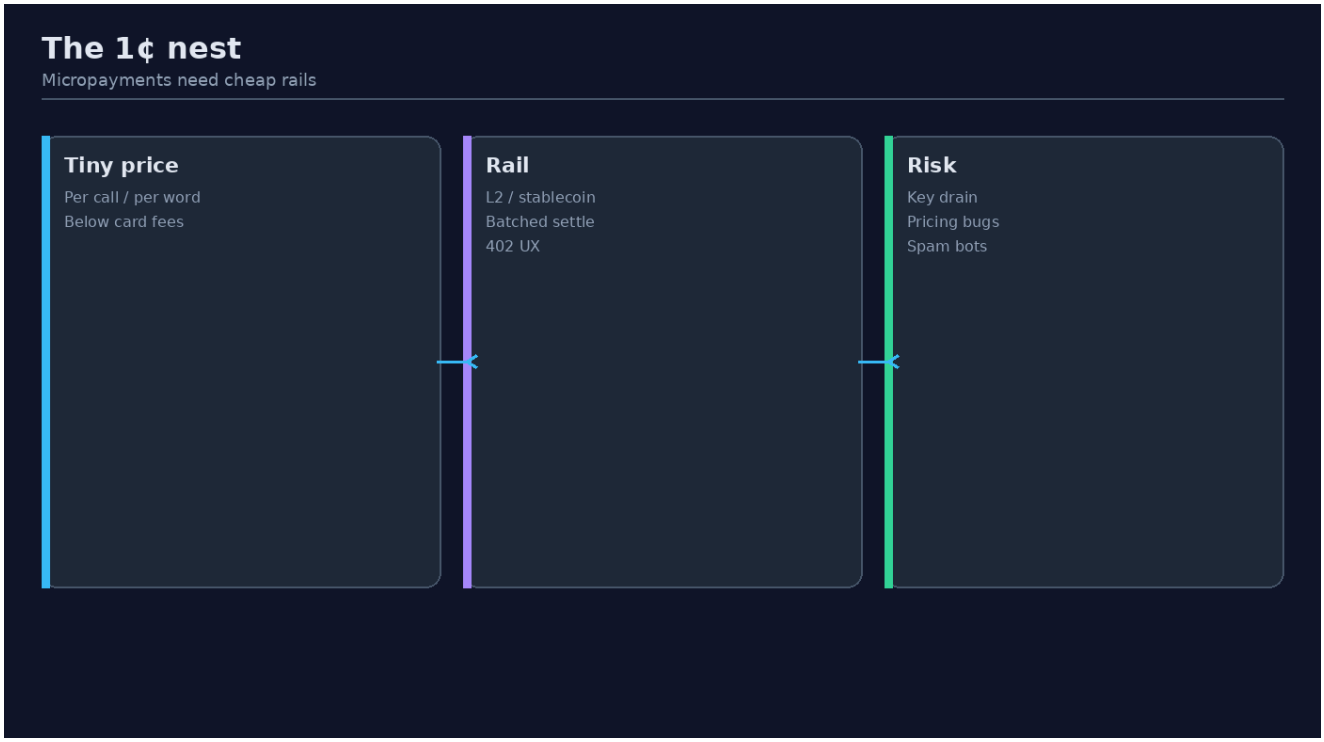
Trading rails and volume illusions

Bots can market-make, arb, crawl—and fake volume. When you see **volume × unit price**, ask: real demand or wash? SCDO stresses verifiable heartbeats/identity; washed volume is not success.



SCDO settle nest (~1¢ intuition)

Think of high-frequency micro-settlement as a **nest**: identity + signed heartbeat + optional pay routes. SoftCenter `scdo_ledger`-style packs are **identity/heartbeat settle-nests**, not a casino lobby and not x402 itself. ~**$0.01** is the intuition: machine commerce must be cheap enough to run.



Newcomer rules

1. Signing/heartbeats before payments.
2. Caps, allowlists, revocable permissions.
3. Separate testnet vs mainnet.
4. Never hand agent seeds to “operators.”



Chapter wrap

Bots × chain = self-custody keys + verifiable ledger + 402/per-call pay + honest metrics. Next: payments and stablecoins.

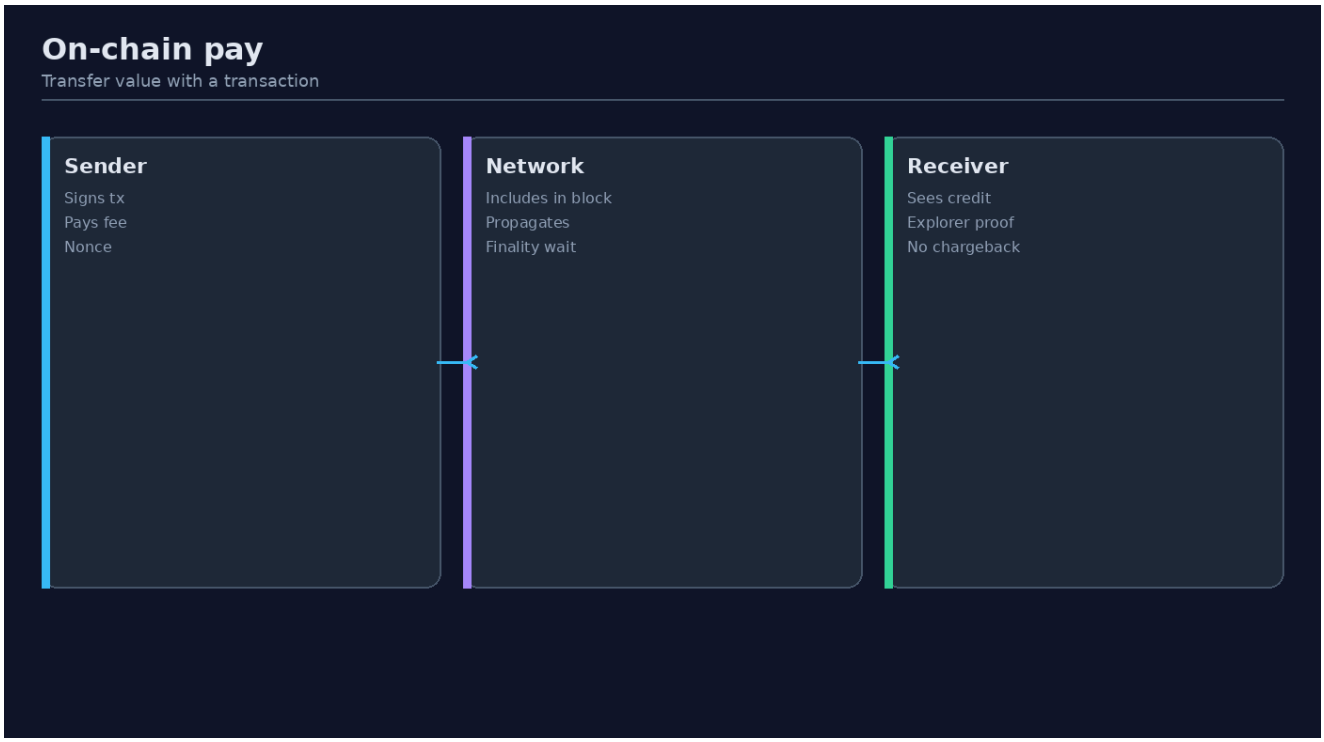
(Chapter 12 · Bots · SCDO Lab)

13. Payments: on-chain pay, stablecoins, x402, SCDO settle shell

> SCDO Lab · *Automated Contract Ledgers for AI Agents* · subtitle *Next-Gen Smart Payments* · ZH: `13-payments.md`
> Not investment advice.

What on-chain pay solves

Turn “I paid” into a **verifiable state change**: amount, asset, payee, time, tx hash. Good for programmatic settlement; not a claim of perfect anonymity or zero regulation.



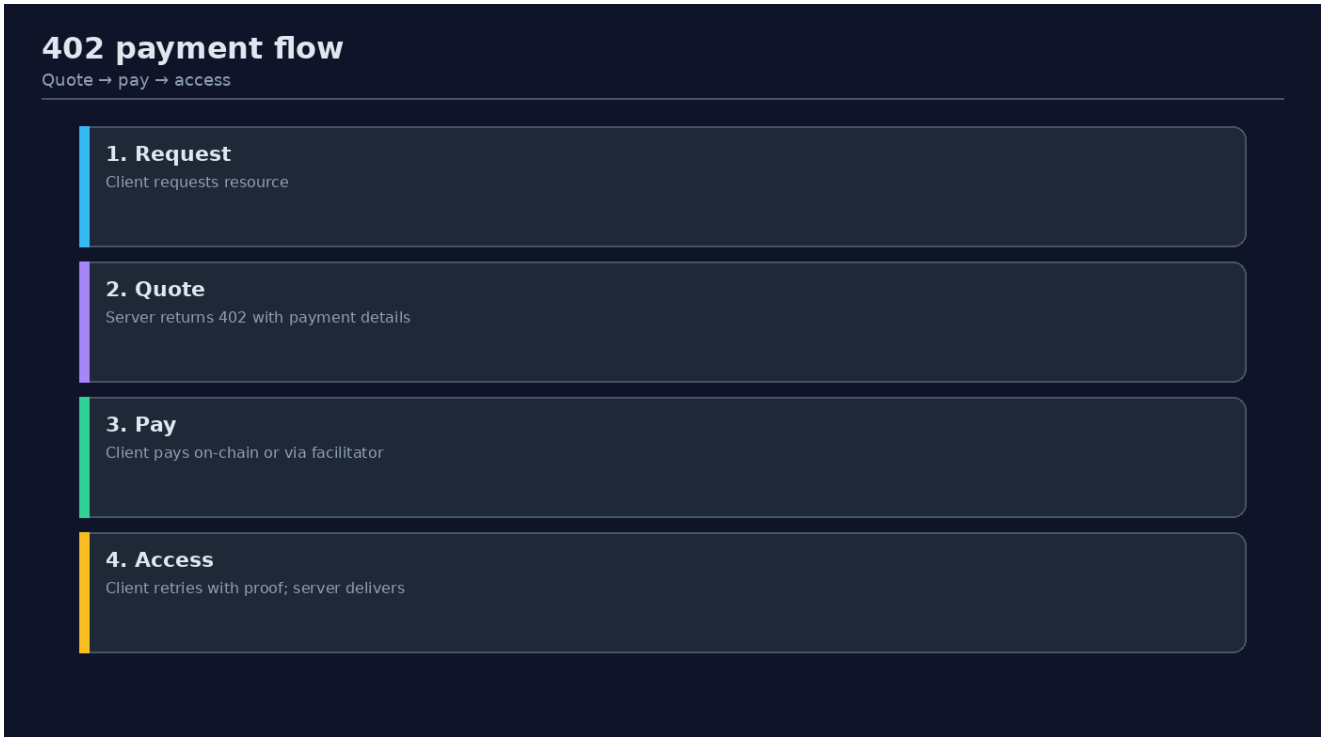
Stablecoins: unit of account and risk

Stablecoins price machine commerce near fiat units. Risks: reserves/depeg, freezes, contract bugs, issuer policy. Treat as a **tool**, not risk-free cash.



x402 / HTTP 402 payment shell

Per-call unlock: request → 402 → pay → retry. SCDO-related practice wires micro **USDC** into agent calls; follow live pilot docs—early stage may change.



SCDO settle shell and ~1¢ nest

Settle shell = thin layer around ledger and pay routes: caps, receipts, anchors. ~**$0.01 nest** is cost intuition—smart payments must support high-frequency micros, not only large wires.



Relation to CEX / DEX

Fiat on/off often via **CEX**; on-chain swaps via **DEX/AMM** (Ch. 5). Agent pay layers reduce clicks but may still depend on issuers and bridges—each hop adds trust assumptions.



Compliance (high level)

Cross-border pay/exchange may trigger local licensing and reporting. In AU context, read operator AUSTRAC DCE / AFCA disclosures (Ch. 10)—**registration ≠ endorsement**.



Chapter wrap

Next-gen smart payments = stablecoin pricing + 402 per-call gates + verifiable ledger shell + honest risk disclosure. Next: gambling fairness/compliance lens .

(Chapter 13 · Payments · SCDO Lab)

14. Gambling lens: on-chain fairness, settle rails, compliance (high level)

> SCDO Lab · educational. Explains verifiable randomness/settlement ideas — **not a gambling tutorial, not a casino funnel**.

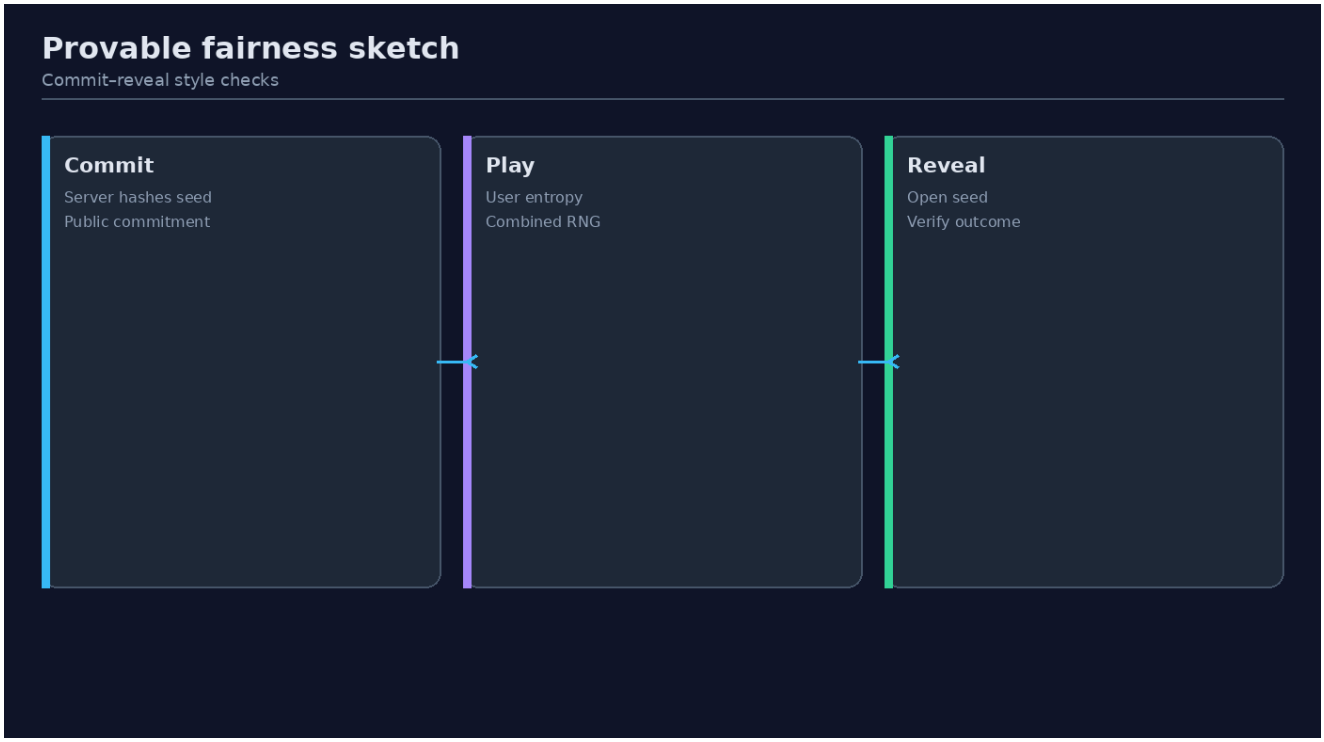
Why this chapter?

Chains often market “provably fair.” Newcomers must separate: **math commitment ≠ legal to play ≠ honest house**. This chapter is a map and red lines—not betting strategy.



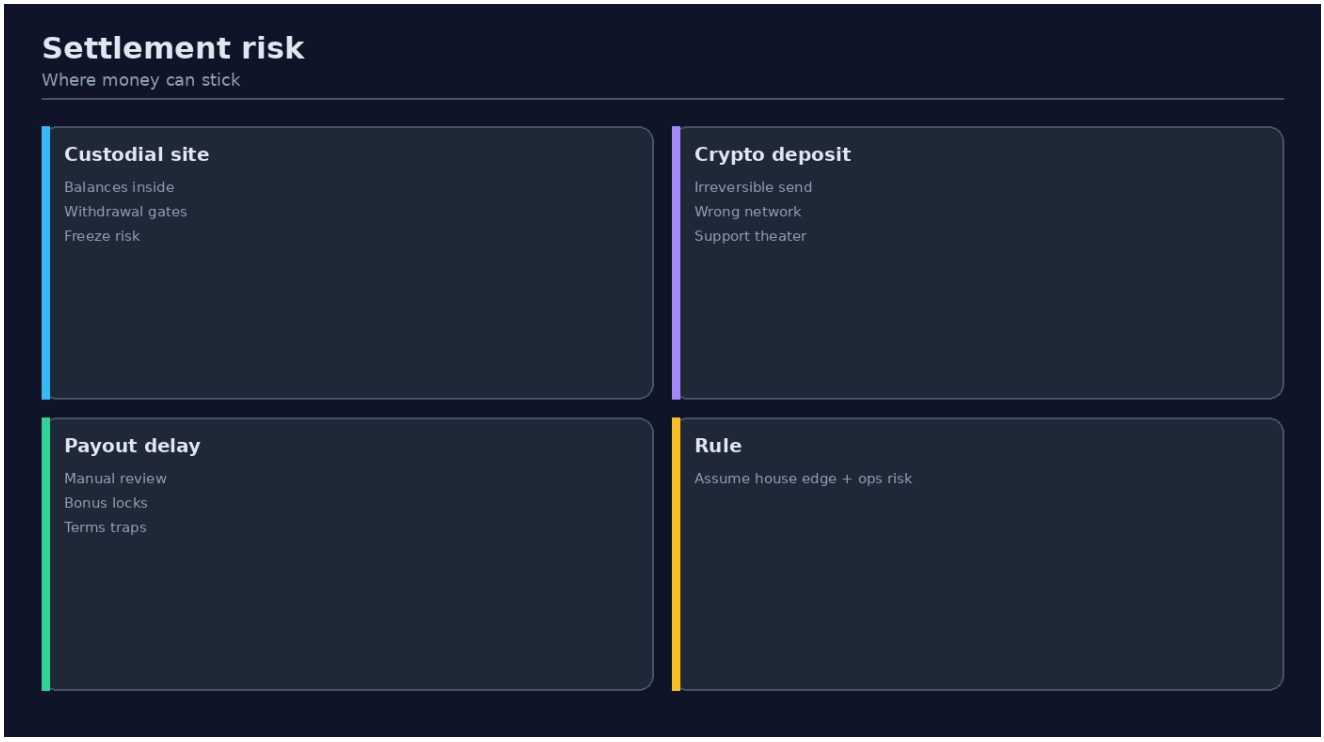
What “on-chain fairness” claims

Common pattern: commit hash → reveal → verify randomness/result. It reduces silent post-hoc edits; it does **not** erase backdoors, oracle games, frontend swaps, withdrawal refusals, or legal risk.



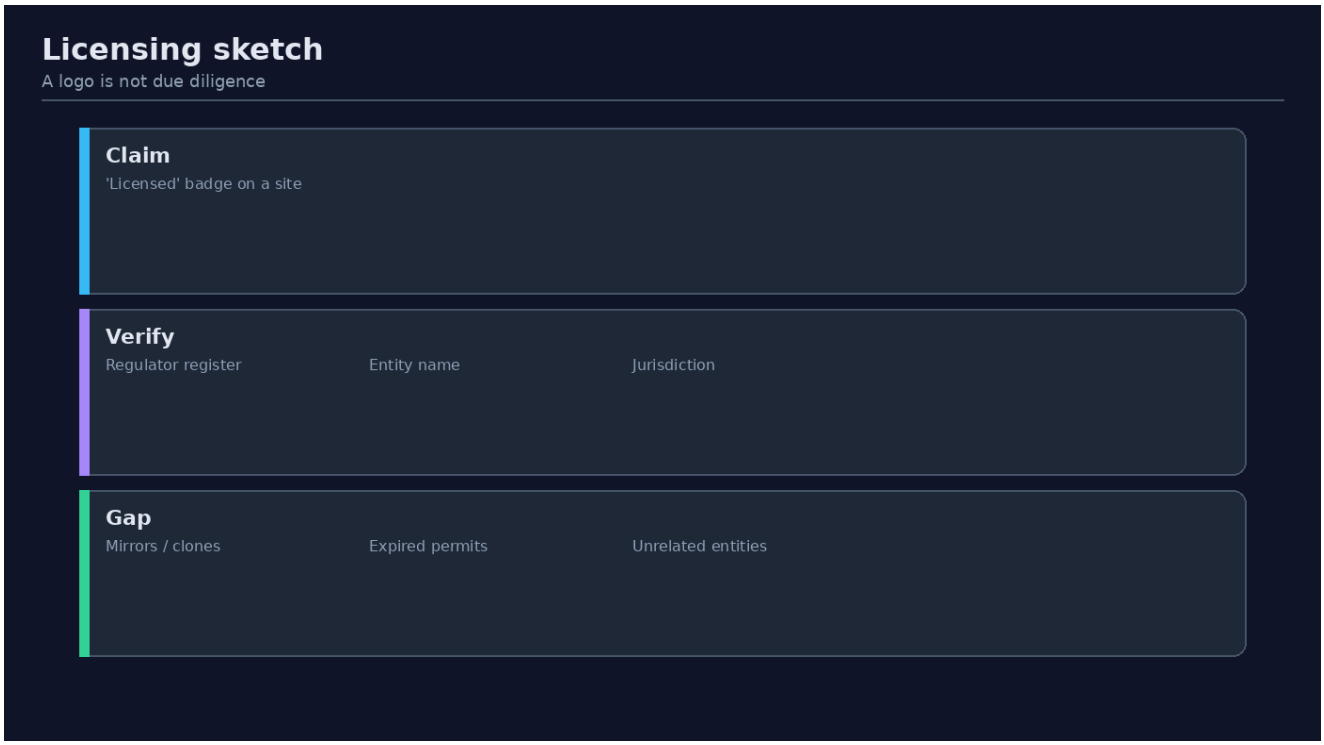
Settle rails

Bets, payouts, escrow, bridges add counterparties. Read them as **high-risk settlement** (same lens as Ch. 13): caps, audits, pause switches, custody.



Licensing (AU high-level sketch)

Crypto exchange registration and gambling licenses are different regimes. SCDO’ s public **AUSTRAC DCE** / **AFCA** materials are exchange-operator disclosure context — **not** “casino license, gamble freely.” Local gambling ads/age rules can be strict.



SCDO games / ledger angle (restrained)

If the site shows game-facing experiments or ledger features, read them as **verifiable-record / permission demos** under local law. Agent Ledger is agent bookkeeping—not a casino backend.



Red-line list

1. Do not teach gambling or funnel players to casinos.
2. Do not present tech demos as playable partners.
3. No evasion guides.
4. Minors / banned jurisdictions: leave.



Chapter wrap

This chapter **de-noises**: unpack fairness marketing, treat settlement as high risk, localize compliance, and refuse casino ads dressed as SCDO tech. Return to verification and safety habits.

(Chapter 14 · Gambling lens · SCDO Lab)