From digital assets to a world computer

Session 8 · 28 September 2026

Bitcoin: an asset and an accounting mechanism

Bitcoin defines a digital asset together with the rules that account for it.

- Holding and receiving the asset requires **no registration**.
- Only the holder of the corresponding keys can **spend** it.
- Nobody can **prevent** a valid transaction from eventually being included.
- The system has **no single operator** whose withdrawal stops it.
- New units are **issued** by a fixed, public schedule.

Keys instead of accounts

A Bitcoin identity is a cryptographic key pair, created by the user alone.

- A private key is essentially a **random number**. Coin tosses or dice would suffice.
- The public key and the address are **derived** from it by computation.
- No institution, network connection, or permission is involved.
- A valid signature can only be produced with the private key, and anyone can verify it.

Compare a bank account: a balance in euros requires the bank to open an account and establish an identity first.

A system that keeps running

File-sharing networks such as BitTorrent offer a comparable **liveness** property.

- A file stays available as long as **one participant** keeps offering it.
- Removing it would require every holder to stop.

Bitcoin obtains a similar property for an accounting system: it keeps operating as long as some participants keep validating and producing blocks.

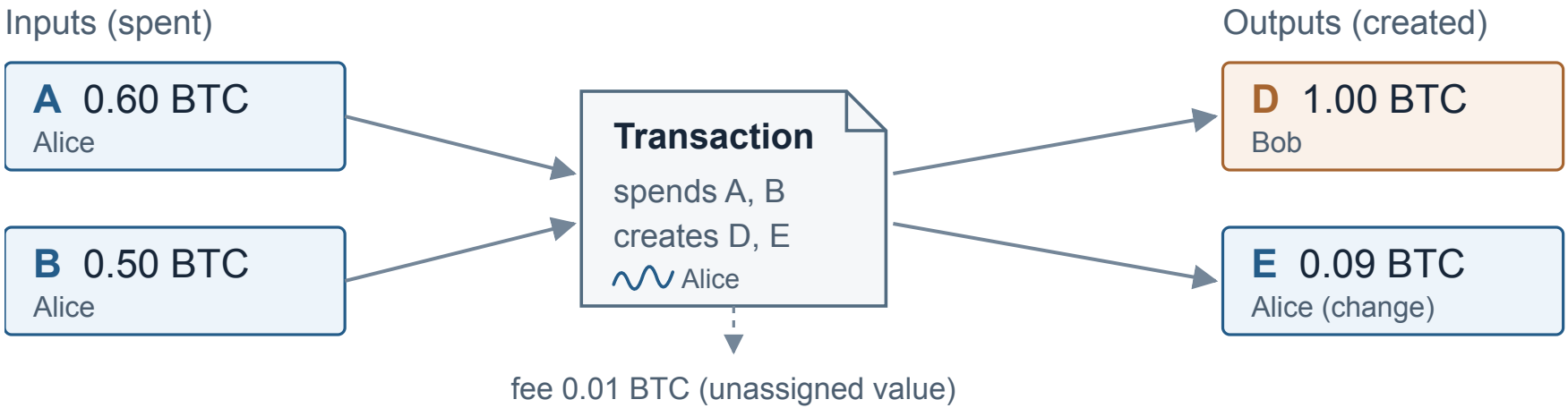
The state: a set of unspent outputs

At any moment, Bitcoin's **state** records how many bitcoins exist and who can spend them.

Output	Amount	Spendable by, simplified
A	0.60 BTC	Alice's address
B	0.50 BTC	Alice's address
C	0.30 BTC	Carol's address

Each row is an **unspent transaction output (UTXO)**: a "bag" of bitcoins with an owner. A wallet computes a balance by adding the rows it can spend.

A transaction spends outputs and creates new ones



The transaction names its inputs, its outputs, and carries Alice's **signature**. The fee is what the outputs leave unassigned.

The administrator: a short program

Applying a transaction to the state takes a few steps.

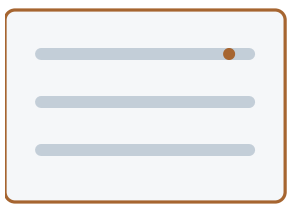
1. Check that every input exists and is unspent.
2. Check the signature against each input’s owner.
3. Check that outputs plus fee do not exceed inputs.
4. Remove the inputs from the set and add the outputs.

These rules could be written in about a hundred lines of Python.

Thought experiment: the same rules on my server

Accounting rules
check signature · spend · create

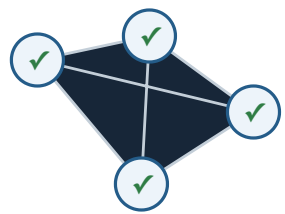
On a server I operate



one operator

- ✕ it can be switched off
- ✕ its code can be changed
- ✕ a transaction can be ignored

On the Bitcoin network



many independent nodes

- ✓ no single off switch
- ✓ rule changes need wide agreement
- ✓ anyone may mine

Identical rules, run by a single operator, lose the properties that make Bitcoin interesting.

Three properties of the special computer

Property	On a server I operate	On the Bitcoin network
Liveness	stops when I stop it	runs while someone mines
Fixed rules	I can change the code	changes need node adoption
Open access	I can ignore a transaction	anyone can include it

Bitcoin is a **simple program** running on a **computer with unusual properties**.

What was new in 2008?

- Digital signatures existed since the 1970s.
- Ledgers and account-keeping rules were long established.
- The new element was a way to run such rules on a computer **nobody controls**.

If the computer is the innovation, a question follows: **can it run programs other than Bitcoin's accounting rules?**

Why digitize?

Digital information and processes are more efficient to handle.

- **Store**: thousands of books fit on a USB stick.
- **Transmit**: a copy reaches the other side of the world in seconds.
- **Process**: counting a word across a library takes milliseconds.
- **Automate**: applications and procedures no longer need a queue at a counter.

Digitization also enables applications that had no physical equivalent.

The credibility problem

A digital process runs on a computer controlled by someone.

- Users must **trust** that the published rules are the rules actually executed.
- Whoever controls the computer could change the result **without visible traces**.
- In daily life, other mechanisms hold operators accountable: law, audits, reputation.

When the stakes are high enough, that trust is not available, and the process stays analog.

Example: electronic voting

Proposals to vote through an app recur, and meet strong opposition.

- **Verification**: how can voters check that every vote was counted as cast?
- **Secrecy**: the operator of the server could link votes to voters.
- Publishing the source code does not show which code actually ran.

Raised in class: anonymity, the risk of rigging, and accessibility.

Other credibility problems

Content ranking. Public debate takes place on platforms whose ranking rules are opaque. Whoever controls them decides who sees which message.

AI training and inference. A model may decide on a loan. Was it trained on the declared data and algorithm? Does the answer really come from that model?

In each case one would want to verify that a **known program** was executed **as written**.

Generalizing Bitcoin

Vitalik Buterin asked whether anyone could run their own programs on a Bitcoin-like computer.

- Attempts to extend Bitcoin itself did not succeed, for technical and cultural reasons.
- He proposed a new blockchain, **Ethereum**, built around general programs.
- Programs uploaded to it inherit the properties of the underlying network.

What Ethereum shares with Bitcoin

- A native asset, **ether (ETH)**, playing the role of bitcoin.
- A shared **state**, updated by **transactions**.
- Transactions are collected and put in an agreed **order** by consensus.
- Consensus was initially **proof of work**, like Bitcoin's.

Ethereum later moved to **proof of stake**, the topic of a later session.

Account-based state

Ethereum’s state is a table of **accounts**, not of unspent outputs.

Address	Balance	Nonce	Code
0x12…a4 (Alice)	10 ETH	30	—
0xab…c7 (Bob)	2 ETH	6	—
0x9f…e1	0.5 ETH	1	contract code

A balance is read from a single row instead of being summed over many outputs.

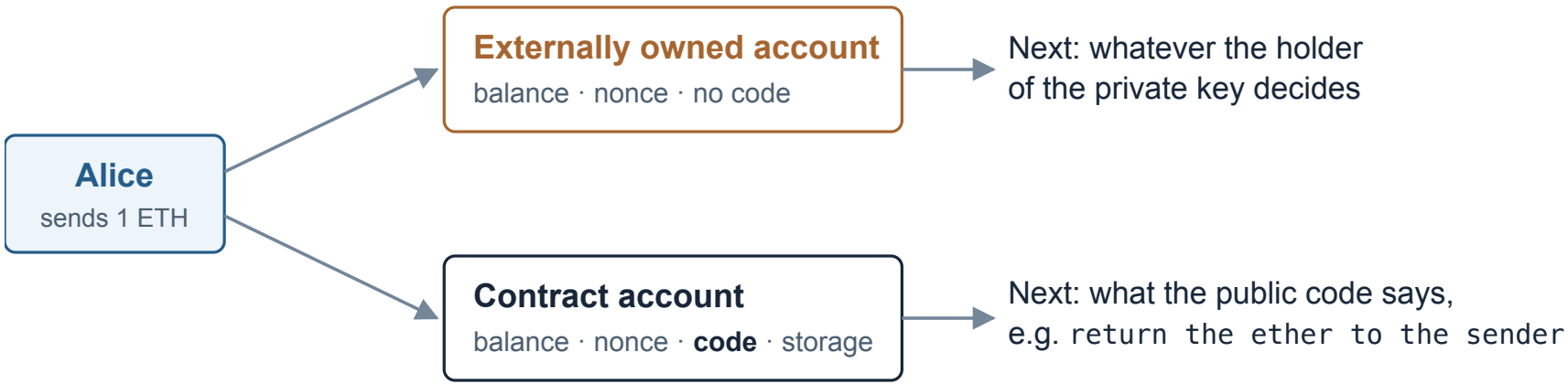
A transfer transaction

To send 3 ETH to Bob, Alice signs a transaction with these fields:

Field	Example
to	Bob’s address
value	3 ETH
nonce	30
fee limits	maximum gas and price
signature	made with Alice’s private key

The sender’s address is recovered from the signature. Executing the transaction moves 3 ETH from Alice’s row to Bob’s.

Two kinds of account



An **externally owned account (EOA)** is controlled by a private key. A **contract account** is controlled by its code.

Smart contracts

A **smart contract** is an account whose behavior is determined by code stored in the state.

- The code is **public**: anyone can read what it does.
- The code **cannot be changed** after it is created.
- Its execution follows the rules of the network, like a Bitcoin transfer.
- **Anyone** can create one.

An election tally written as a contract would let everyone check that the published result comes from the published code.

Hands-on: a wallet on a test network

1. Create a new browser profile and install **MetaMask** from metamask.io.
2. Choose **Create a new wallet**. Decline the optional social login.
3. Set a password, then write down the **secret recovery phrase** offline.
4. Under *All networks* → *Manage networks*, enable **Show test networks**.
5. Select **Sepolia**. The new account has a balance of 0.

Sepolia works like Ethereum, but its ether has **no monetary value**.

Why a password, if there are keys?

- The wallet stores the private key on the computer.
- The **password encrypts** it, so someone with access to the device or its memory cannot read it directly.
- The **recovery phrase** regenerates the keys on another device. Whoever has it controls the funds.
- **Social login** stores a recoverable backup through a third-party account, trading self-custody for convenience.

Before next session

- Submit your Sepolia address through the course form, under any nickname.
- Next session: receiving test ether, a block explorer, and the first smart contract.

Activities on Sepolia leave public, verifiable traces. They will be used to award **badges** to participating addresses.