Smart Contracts: The Visual Guide

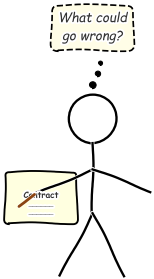
A Story in Pictures

Prof. Dr. Jörg Osterrieder

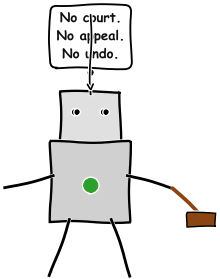
BSc Blockchain, Crypto Economy & NFTs

Spring 2026

Can Code Replace a Lawyer?



Developer / User



Smart Contract (EVM)

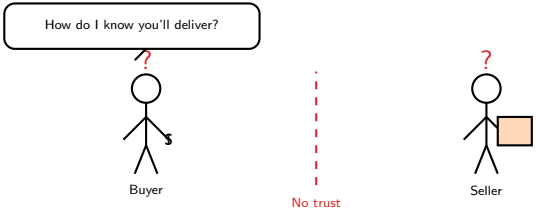
This lecture tells the story of how code can enforce deals automatically — no courts, no paperwork.

What Will You Learn Today?

1. **Why** traditional contracts fail online
2. **How** smart contracts enforce deals automatically
3. **What** you can build and what risks to watch

This visual guide complements the full Ethereum & Smart Contracts lecture. Go deeper after you understand the story.

Can You Trust a Stranger Online?



Online, strangers have no reason to trust each other. No handshake, no eye contact, no shared community.

Why Do Traditional Deals Break Online?

Three problems:

1. **Slow** — lawyers draft contracts over weeks
2. **Expensive** — legal fees run hundreds to thousands
3. **Not global** — different countries, different laws, different courts

The core question:

"What if a computer program could enforce your deal — instantly, cheaply, and anywhere in the world?"

That program is called a **smart contract**.

Smart contracts were first described by Nick Szabo in 1994 — long before blockchain existed.

Can a Blockchain Do More Than Store Value?

Bitcoin vs Ethereum: Six Dimensions

Dimension	Bitcoin (BTC)	Ethereum (ETH)
Scripting	Limited (Script)	Turing-Complete (Solidity)
State Model	UTXO	Account-Based
Purpose	Value Transfer	Programmable Logic
Throughput	~ 7 TPS	~ 15 TPS
Launch Year	2009	2015
Consensus	Proof of Work	Proof of Stake

Bitcoin is a calculator: it can add and subtract balances. Ethereum is a computer: it can run any program.

What If Code Could Enforce Your Deal?

Traditional contract:

1. Lawyer drafts agreement
2. Both parties sign
3. Courts enforce if someone cheats
4. Takes weeks, costs thousands

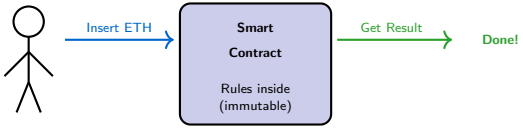
Smart contract:

1. Developer writes code
2. Code deploys to blockchain
3. Code executes automatically
4. Takes seconds, costs cents

No lawyer, no judge, no waiting.

A smart contract is "self-enforcing" — it does exactly what the code says, every time, with no human intervention.

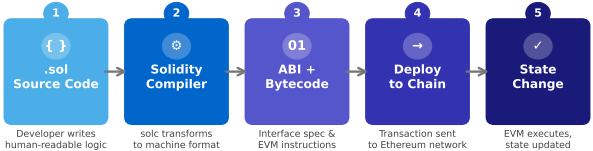
Meet the Vending Machine



A smart contract is like a vending machine: insert your input, get your output. No cashier, no negotiation, no trust needed.

How Does Code Become a Contract?

From Solidity to State Change



Every smart contract follows this exact pipeline — no exceptions

Solidity (human-readable) compiles to bytecode (machine-readable), which the Ethereum Virtual Machine executes.

Who Lives on Ethereum?

Externally Owned Accounts (EOA):

- Controlled by people with private keys
- Like your personal wallet
- Can send transactions and hold ETH

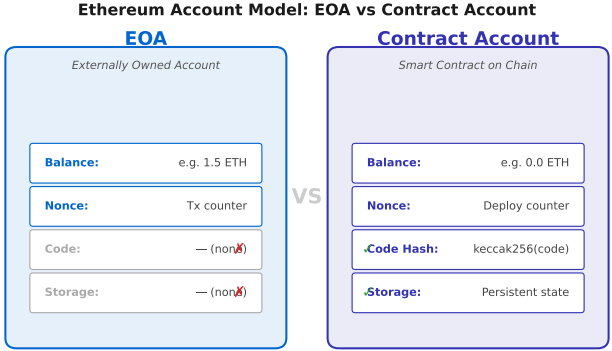
Contract Accounts:

- Controlled by code, not people
- Like programs that live on the blockchain
- Can hold funds, enforce rules, and call other contracts

Every smart contract has an address, just like your wallet. But nobody holds its private key — the code is in charge.

Ethereum has two types of citizens: humans (EOA) and programs (contracts). They interact through transactions.

How Do People and Programs Coexist?



People send transactions to contracts, contracts call other contracts. The EVM executes everything deterministically.

What Does It Cost to Run Code on a World Computer?

The gas analogy:

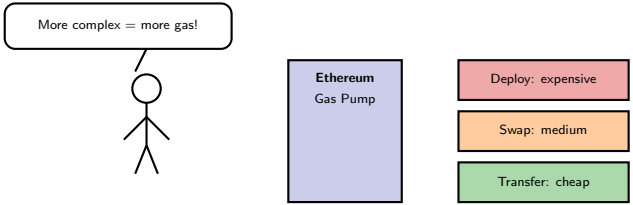
- Every operation costs "gas" — like fuel for a car
- More complex code = more gas
- You set a maximum you are willing to pay

Why gas exists:

- Prevents infinite loops (code that never stops)
- Pays validators for their work
- Makes spam attacks expensive

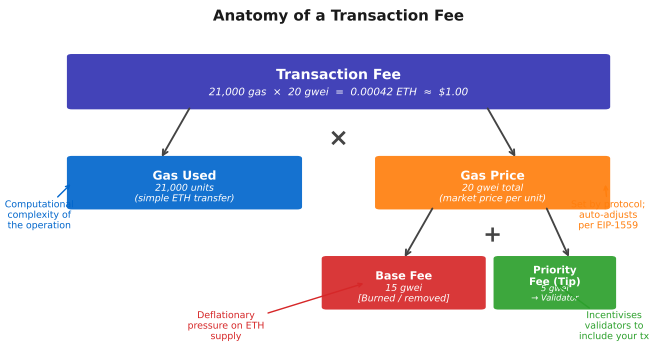
Without gas, anyone could deploy code that runs forever and freeze the entire network. Gas is Ethereum's self-defense.

Welcome to the Gas Station



A simple ETH transfer costs about 21,000 gas. Deploying a new contract can cost millions. Complexity has a price.

Why Do You Pay Gas Fees?



- Gas = fuel that powers every operation on the world computer
- You pay per unit of computation, not per dollar amount transferred

Since EIP-1559, part of each fee is burned (destroyed) and part goes to the validator who processes your transaction.

What Can You Build With Smart Contracts?

Financial tools:

- **Tokens** — create your own currency (ERC-20)
- **NFTs** — unique digital items (ERC-721)
- **DeFi** — lending, trading, insurance without banks

Beyond finance:

- **DAOs** — organizations governed by votes, not bosses
- **Games** — own your items, trade freely
- **Supply chain** — track products from factory to shelf

If a deal can be expressed in rules, a smart contract can enforce it.

The Ethereum ecosystem has over 4,000 active dApps (decentralized applications) built on smart contracts.

Pick Your Token Type

ERC-20

Fungible tokens

Like money — every unit is identical. Used for currencies, governance votes, and loyalty points.

ERC-721

Unique tokens

Like art — every piece is one-of-a-kind. Used for collectibles, deeds, and identity.

ERC-1155

Multi-tokens

Like a gaming inventory — mix of unique and stackable items in one contract.

ERC = Ethereum Request for Comment. These standards let wallets and marketplaces understand any token automatically.

Which Token Standard Do You Need?

Token Standards: ERC-20 vs ERC-721 vs ERC-1155

	ERC-20	ERC-721	ERC-1155
Fungibility	Fungible	Non-Fungible	Both
Batch Ops	No	No	Yes
Gas / Transfer	~65,000 gas	~85,000 gas	~40,000 (batch)
Use Case	Currencies Stablecoins	Art Collectibles	Gaming Items
	Low	Medium	High

Choose ERC-20 for currency, ERC-721 for unique items, ERC-1155 for games. Most projects use just one standard.

What Is the Catch?

Three risks you must know:

1. **Bugs are permanent** — once deployed, code cannot be changed. A flaw stays forever.
2. **Code is public** — anyone can read your contract, including hackers looking for exploits.
3. **Immutable means immutable** — there is no "undo" button, no customer support, no refund.

The tradeoff:

Traditional contracts are *flexible but slow*.
Smart contracts are *fast but rigid*.

You trade human judgment for mathematical certainty.

The DAO hack in 2016 exploited a reentrancy bug and drained $60 million. The code worked exactly as written — just not as intended.

How Do You Stay Safe?

Rule 1

Audit the code.

Use only contracts reviewed by professional security firms. No audit = no trust.

Rule 2

Test on a testnet.

Deploy on Sepolia first. Practice with fake ETH before risking real money.

Rule 3

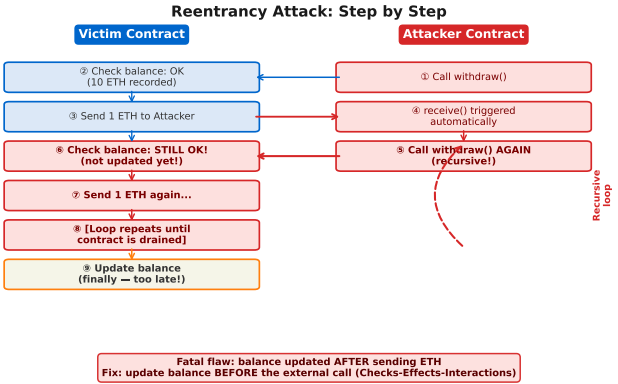
Start small.

Deploy with minimal funds. Scale up only after the contract has been battle-tested.

Every major exploit could have been prevented with a proper audit.

Top audit firms include OpenZeppelin, Trail of Bits, and Consensys Diligence. An audit costs $10K–$100K but can save millions.

What Happens When Code Has a Bug?



A reentrancy bug lets an attacker call a function repeatedly before it finishes, draining funds. This is the most famous smart contract vulnerability.

How Does a Smart Contract Compare to a Paper Contract?

	Paper Contract	Smart Contract
Speed	Days to weeks	Seconds
Cost	Hundreds to thousands	Cents to dollars
Enforcement	Courts, police	Code executes automatically
Reversibility	Can be voided by a judge	Immutable once executed
Access	Requires identity, jurisdiction	Anyone with a wallet

Neither is "better" — they solve different problems. Complex agreements still need legal contracts.

Smart contracts excel at simple, repetitive, rule-based deals. Paper contracts handle ambiguity and human judgment.

Deploy Your First Smart Contract

Three steps, zero cost:

1. **Open Remix IDE** — free browser tool at remix.ethereum.org, no installation
2. **Write a simple contract** — start with a "Hello World" that stores one message
3. **Deploy on Sepolia testnet** — connect MetaMask, click deploy, done

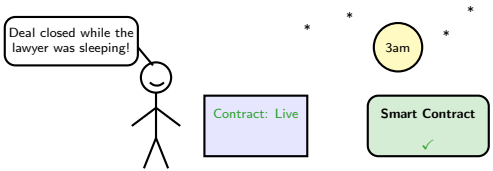
What you will see:

- Your contract gets a unique address
- Anyone can interact with it
- It lives on the blockchain forever
- The entire process takes under 5 minutes

Total cost: $0. Total risk: $0.

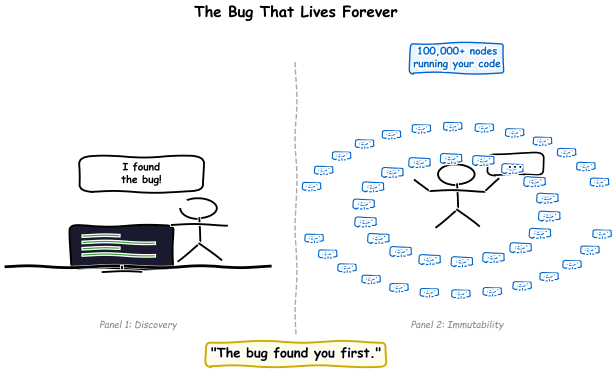
Workshop W01 walks you through this step by step. You will deploy a real contract on Sepolia in 15 minutes.

Celebrate Your First Deal at 3am



Smart contracts work 24/7 — no business hours, no holidays, no coffee breaks. Your deal executes whenever the conditions are met.

Can Anyone Stop This Code?



Once deployed, a smart contract runs as long as the blockchain exists. No government, no company, no individual can shut it down.

Remember These Five Things

1. **Smart contracts** are programs that enforce deals automatically — no middleman needed
2. **Ethereum** is the world computer that runs them — Bitcoin stores value, Ethereum runs code
3. **Gas fees** pay for computation — every operation has a price to prevent abuse
4. **Token standards** (ERC-20, 721, 1155) let you create money, art, and game items
5. **Bugs are forever** — audit, test on a testnet, and start small before going live

Code is law —
make sure the law is correct.

You now understand the story of smart contracts. For Solidity syntax, gas optimization, and security patterns, see the full lecture.