Introduction to Smart Contracts: A Visual Introduction

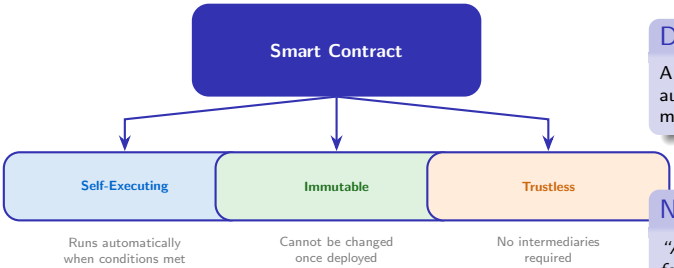
Mini-Lecture

Prof. Dr. Joerg Osterrieder

University Lecture Series

March 5, 2026

What Is a Smart Contract?



Definition

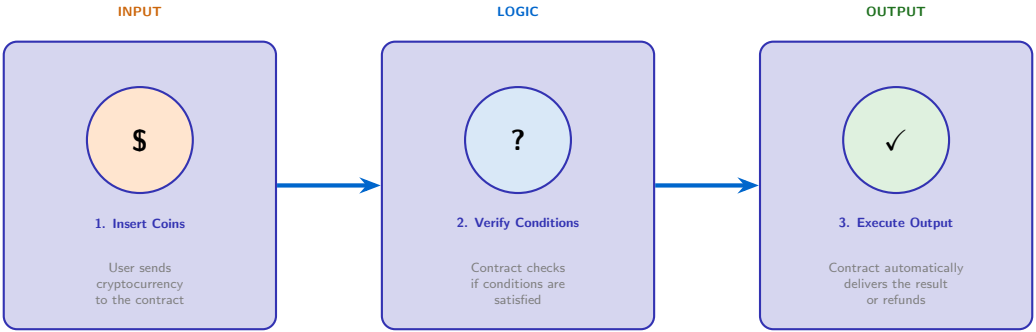
A **smart contract** is a program stored on a blockchain that automatically executes when predetermined conditions are met—no middlemen, no delays, no trust required.

Nick Szabo (1996)

"A smart contract is a set of promises, specified in digital form, including protocols within which the parties perform on these promises."

Nick Szabo coined the term "smart contract" in 1994, long before Ethereum made them practical on a global scale.

The Vending Machine Analogy



Nick Szabo used the vending machine as the simplest example of a "smart contract": deterministic rules, no negotiation, automatic execution.

Smart Contracts vs Traditional Contracts

Feature	Traditional Contract	Smart Contract
Execution	Manual enforcement by parties or courts	Automatic, self-executing code
Intermediaries	Lawyers, banks, notaries	None required
Speed	Days to weeks for settlement	Seconds to minutes
Cost	High fees for legal and processing	Gas fees only (often \$0.01–\$50)
Transparency	Private, restricted access	Public, auditable on-chain
Modification	Amendments possible	Immutable once deployed
Jurisdiction	Bound by national laws	Borderless, code-is-law
Dispute Resolution	Courts and arbitration	On-chain logic or DAO governance

Smart contracts eliminate intermediaries and automate enforcement, but trade flexibility and legal recourse for determinism and speed.

Key Smart Contract Platforms

Ethereum

Solidity / Vyper
EVM-based
Largest ecosystem
$400B+ TVL

Most Mature
Est. 2015

Solana

Rust / Anchor
65,000 TPS
Low fees (**$**0.00025)
Parallel execution

Fastest
Est. 2020

Cardano

Plutus / Haskell
eUTXO model
Formal verification
Peer-reviewed

Most Rigorous
Est. 2017

Polkadot

ink! / Rust
Parachain model
Cross-chain
Shared security

Most Interoperable
Est. 2020

Ethereum dominates smart contract adoption, but Solana, Cardano, and Polkadot offer distinct trade-offs in speed, rigor, and interoperability.

Real-World Applications

DeFi (Decentralized Finance)

- Lending/borrowing (Aave, Compound)
- Decentralized exchanges (Uniswap)
- Yield farming and staking
- **$170B+** total value locked

NFTs (Digital Ownership)

- Digital art and collectibles
- Gaming assets and metaverse
- Music and media royalties
- **$25B+** traded in 2023

Insurance (Parametric)

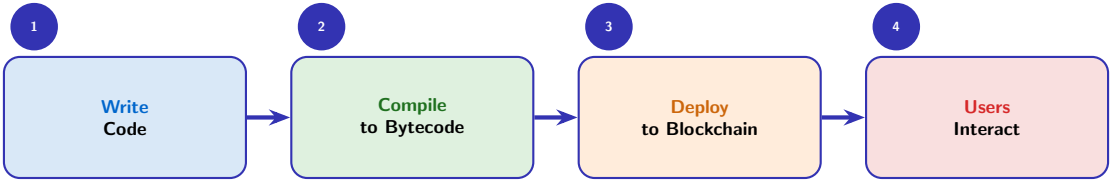
- Automatic claim payouts
- Weather-triggered policies
- Flight delay insurance
- No claims adjusters needed

Supply Chain (Tracking)

- Origin verification
- Automated payments on delivery
- Temperature monitoring (IoT)
- Anti-counterfeiting

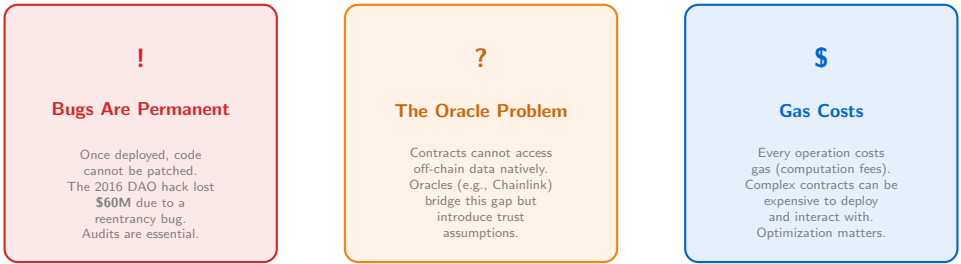
Smart contracts power applications from billion-dollar DeFi protocols to parametric insurance that pays out automatically when conditions are met.

How Smart Contracts Work — Simplified



The lifecycle: write human-readable code, compile to machine bytecode, deploy via a transaction, then anyone can interact with the contract at its address.

Security Considerations and Limitations



Smart contract security is critical: immutability means bugs are permanent, oracles introduce external trust, and gas costs constrain complexity.

Why Smart Contracts Matter

Key Statistics

- ✓ **$170B+** locked in DeFi smart contracts
- ✓ **50M+** unique smart contract addresses on Ethereum
- ✓ **4M+** smart contracts deployed in 2024 alone
- ✓ **$3B+** annual spending on smart contract audits
- ✓ **70%+** of blockchain transactions involve smart contracts

Impact Areas

- ✓ **Finance:** Eliminating intermediaries in lending, trading, insurance
- ✓ **Governance:** DAOs managing billions in community treasuries
- ✓ **Legal:** Self-enforcing agreements reducing court dependency
- ✓ **Identity:** Decentralized identity and credential verification
- ✓ **Real Estate:** Tokenized property and automated escrow

Smart contracts are reshaping industries by automating trust, reducing costs, and enabling programmable agreements at global scale.

Coming Up: The Full Lecture

Prerequisites

- ✓ Basic blockchain concepts (blocks, transactions, consensus)
- ✓ Familiarity with any programming language

Outcomes

- ✓ Read and understand Solidity smart contracts
- ✓ Identify common vulnerabilities and security patterns

The full lecture covers smart contracts from historical foundations through Solidity programming to real-world security and deployment patterns.