EIP 792

The standard for Arbitra(tion/ble) smart contracts.

A Primer on Blockchains

They:

A Primer on Blockchains

They:

- Are distributed data stores that can be used to store any sort of data (e.g. Bitcoin stores balances for a bunch of addresses.).

A Primer on Blockchains

They:

- Are distributed data stores that can be used to store any sort of data (e.g. Bitcoin stores balances for a bunch of addresses.).
- Can be edited as long as certain rules are followed (e.g. No double spending, no spending of others' funds, etc.).

A Primer on Blockchains

They:

- Are distributed data stores that can be used to store any sort of data (e.g. Bitcoin stores balances for a bunch of addresses.).
- Can be edited as long as certain rules are followed (e.g. No double spending, no spending of others' funds, etc.).
- Guarantee provable data integrity, commonly through different consensus protocols based on `cryptography + economics = cryptoeconomics`.

Smart? Contracts

They:

Smart? Contracts

They:

- Are turing complete, stateful programs that get sent in a special deployment `TX` and become immutable.

Smart? Contracts

They:

- Are turing complete, stateful programs that get sent in a special deployment `TX` and become immutable.
- Have their own address to which you can send `TX`s with some input data and value and have some logic executed on-chain.

Smart? Contracts

They:

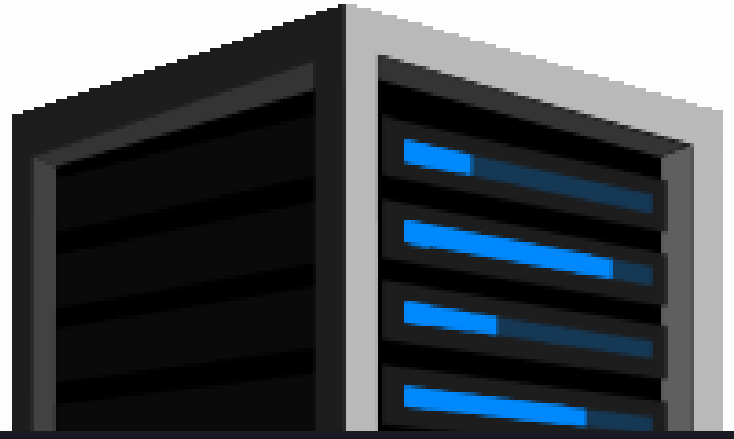
- Are turing complete, stateful programs that get sent in a special deployment `TX` and become immutable.
- Have their own address to which you can send `TX`s with some input data and value and have some logic executed on-chain.
- Guarantee provable data and state transition integrity through the chain they live on, in our case, Ethereum.

A Note on Smart Contract Design

A Note on Smart Contract Design

Traditional Back Ends:

- Do most of the heavy lifting for the front end, because computation is cheap and faster than on the client.



A Note on Smart Contract Design

Traditional Back Ends:

- Do most of the heavy lifting for the front end, because computation is cheap and faster than on the client.

Smart Contracts Back Ends:

- Leave most of the heavy lifting for the front end, because computation is expensive and slower than on the client.