OSI2019

Blockchain Workshop

WORKSHOP OBJECTIVES

Lecture/Demo

A. Blockchain and Ethereum fundamentals

B. Intro to DApps and examples

C. Solidity programming basics

D. Overview of DApps tools and ecosystem

E. How to design DApps? Best practices

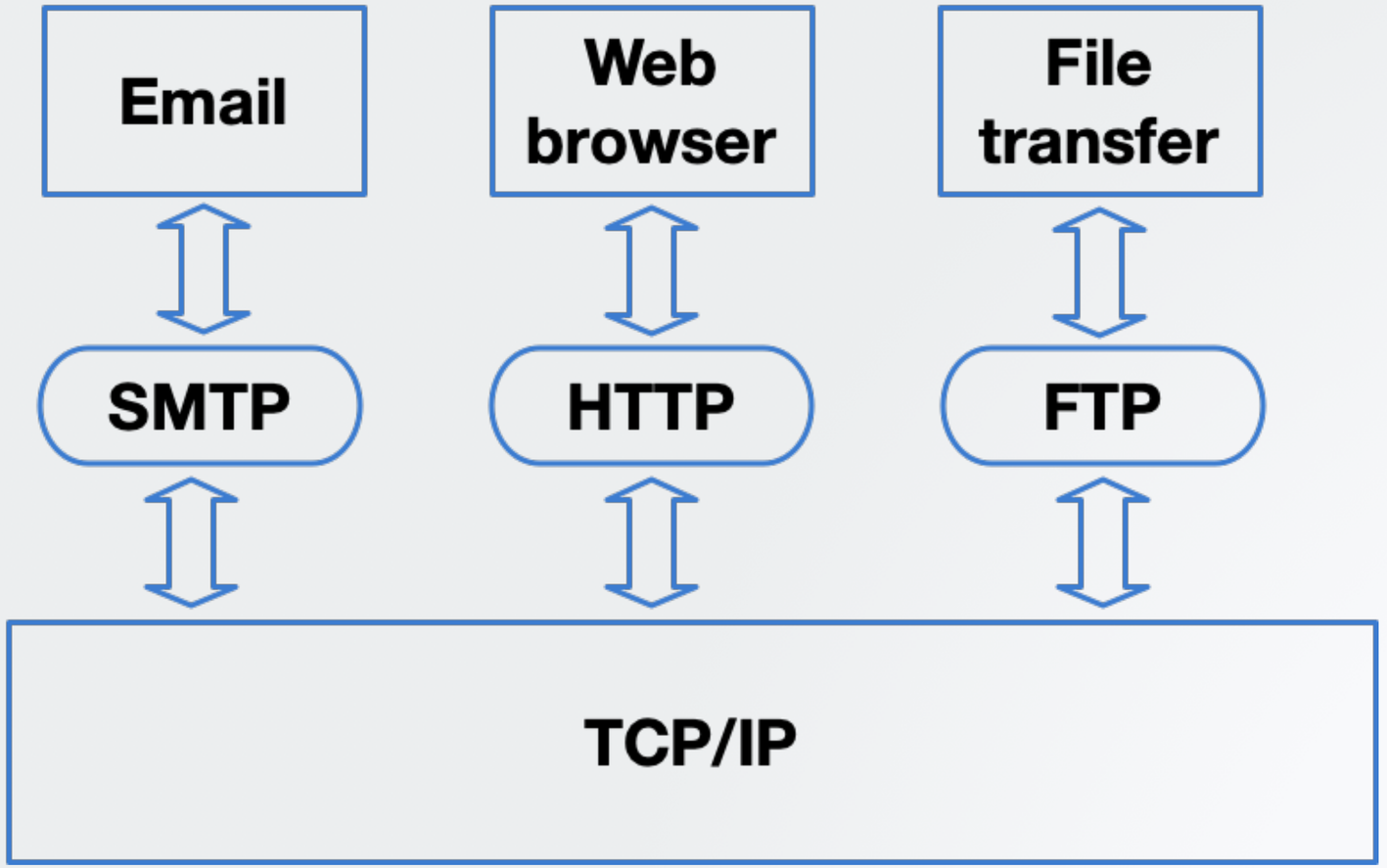
F. Live coding/demo of building a DApp

Hands-on coding exercise and Q&A

G. Build and run a DApp

What are the limitations of Internet?

PROBLEM 1: VALUE TRANSFER OVER INTERNET



PROBLEM 2: LACK OF DISTRIBUTED TRUST

Centralised trust

Intermediated Trust

P2P Trust

Internet solved the problem of information, distribution and communication. It has not solved the problems of transferring value and distributed trust.

OSI-2019

INTERNET TECH HAS NOT SOLVED THESE PROBLEMS

Double Spend

Store , prove possession and transmit value

MAINTAIN SHARED RECORDS WITHOUT INTERMEDIARIES

How can transactions be made anonymously without intermediaries

What is Blockchain?

Blockchain is not technology, it is a philosophy

Decentralisation

Decentralization of control

Disintermediation of transactions

Shared trust

Censorship-resistance

Immutability, Tamper-proof, unalterable truth

Seizure-resistance

Human rights, privacy, civil liberty, freedom of expression, security, empowerment

eg news, elections

Social Good

Financial inclusion

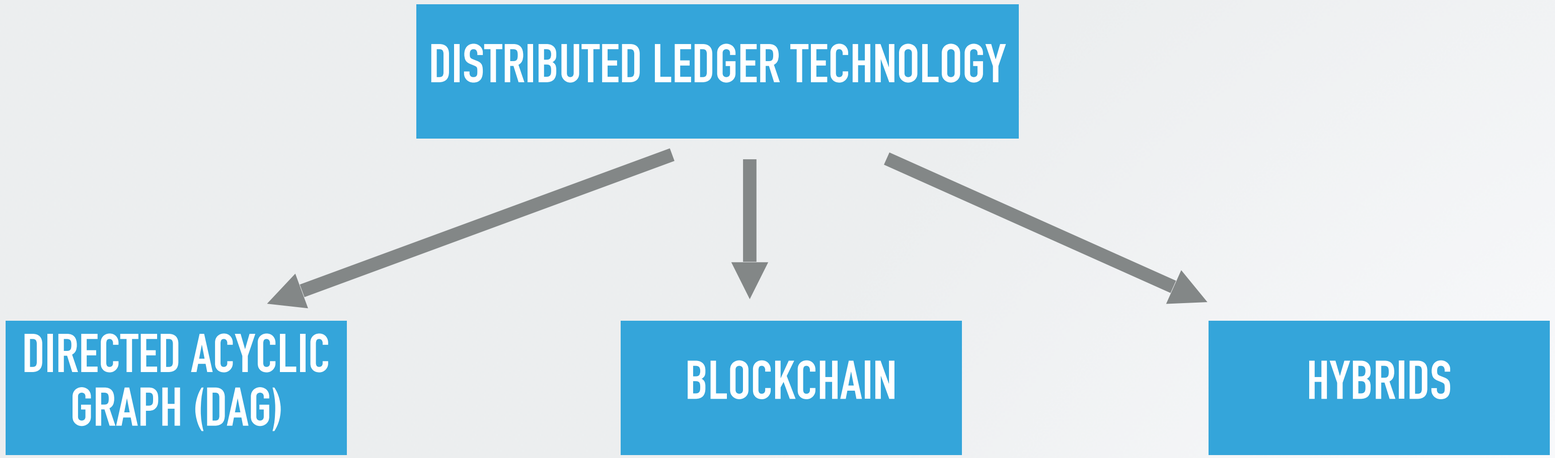
Collective ownership- DAOs, utility tokens, Better distribution of incentives, cooperatives

Let’s get down to fundamentals

BLOCKCHAIN VS DATABASE

Blockchain	*Database*
Only insert & read operations	*CRUD operations*
Full replication of block on every peer	*Master-slave and multi-master configurations*
Majority of peers agree on the outcome of transactions	*Dstributed transactions (2-phase commit)*
Anybody can validate transactions across the network	*Integrity constraints*
Designed for security of transactions	*Designed for performance of transactions*

TYPES OF DISTRIBUTED LEDGERS



What problems does blockchain solve?

OSI-2019

WHAT PROBLEM DOES DLT SOLVE?

Aha Moment

01

INTRODUCES SCARCITY TO DIGITAL ASSETS

Solves the double spend problem. Blockchain enables assets to be identified & owned uniquely and transferred by rightful owner through transactions

02

ENABLES TRUST WITHOUT INTERMEDIARIES

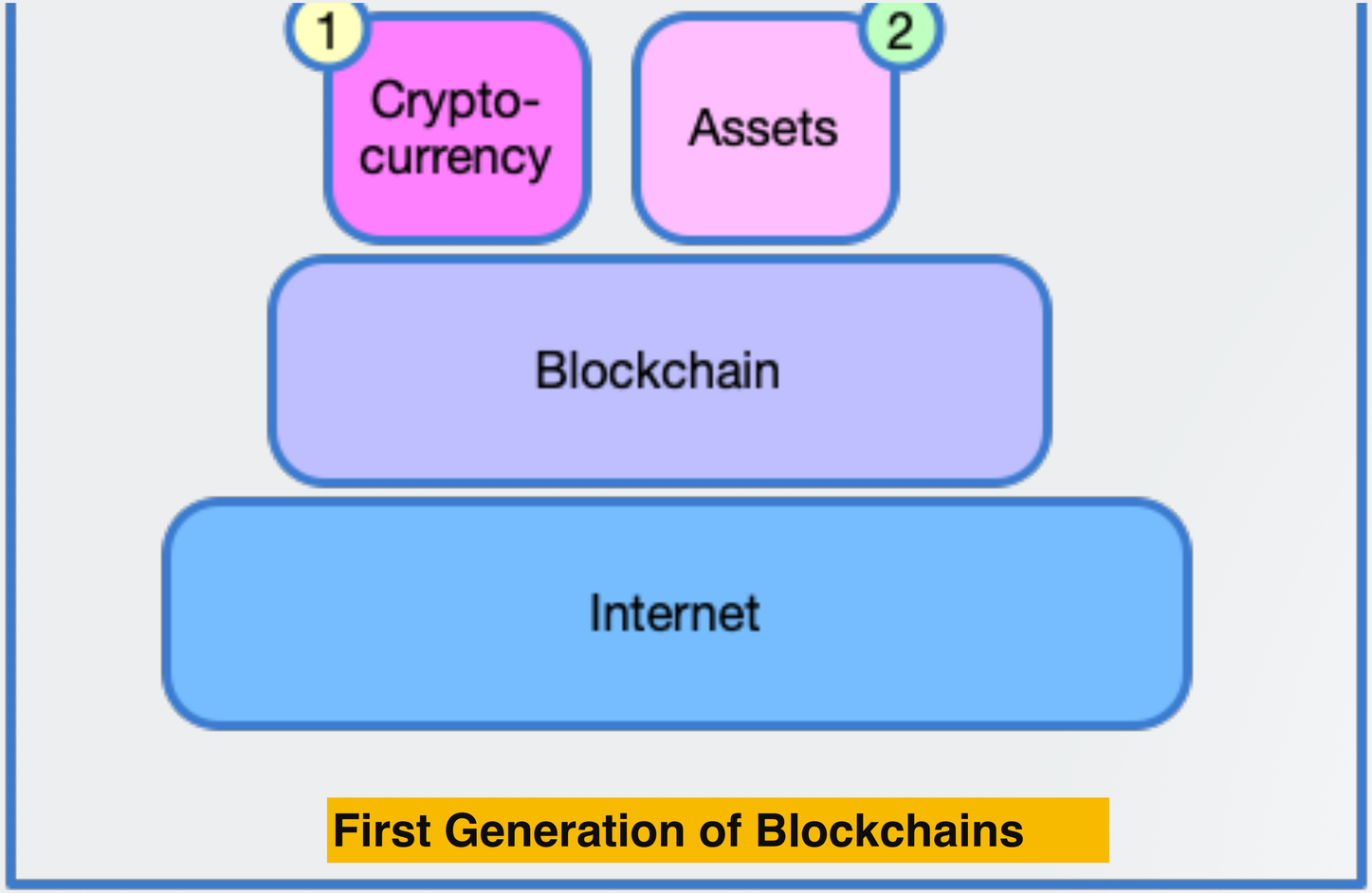
Practically all commercial online transactions are fulfilled through intermediaries. Blockchain enables peer-to-peer transactions without central intermediaries through digital machine consensus.

"The practical consequence […is…] that for the first time, a way for one Internet user to transfer a unique piece of digital property to another internet user, such that the transfer is guaranteed to be safe, and secure, everyone knows the transfer has taken place, and no-one can challenge the legitimacy of the transfer. The consequences of this breakthrough are hard to overstate.

-Marc Andreessen

Generations of Blockchain

FIRST GENERATION OF BLOCKCHAINS



Distributed, immutable database

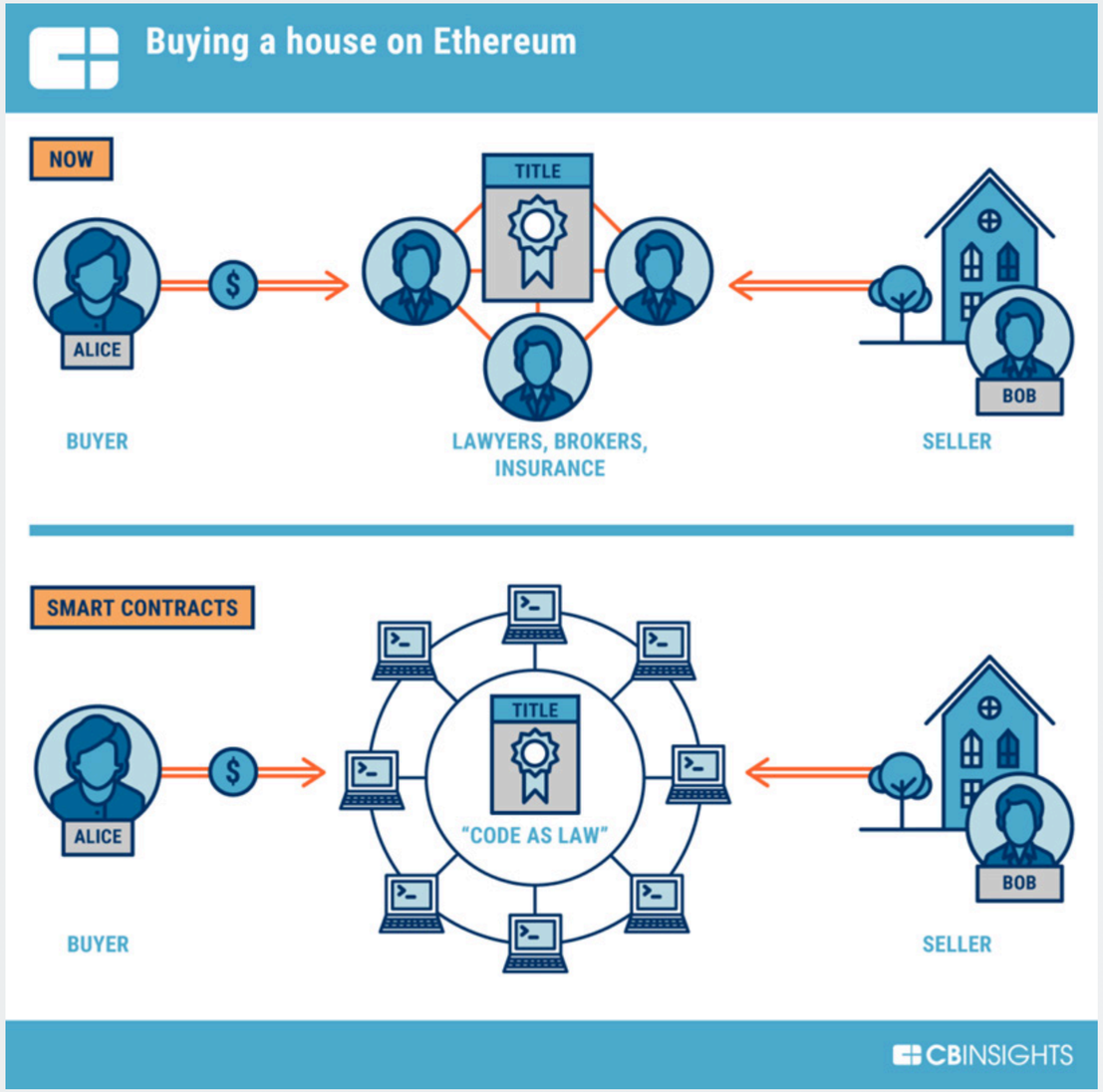
Peer to peer

Assets are anything of value

Tangible- money, real estate, diamonds

Intangible- Financial instruments, identity

SECOND GENERATION OF BLOCKCHAINS- SMART CONTRACTS



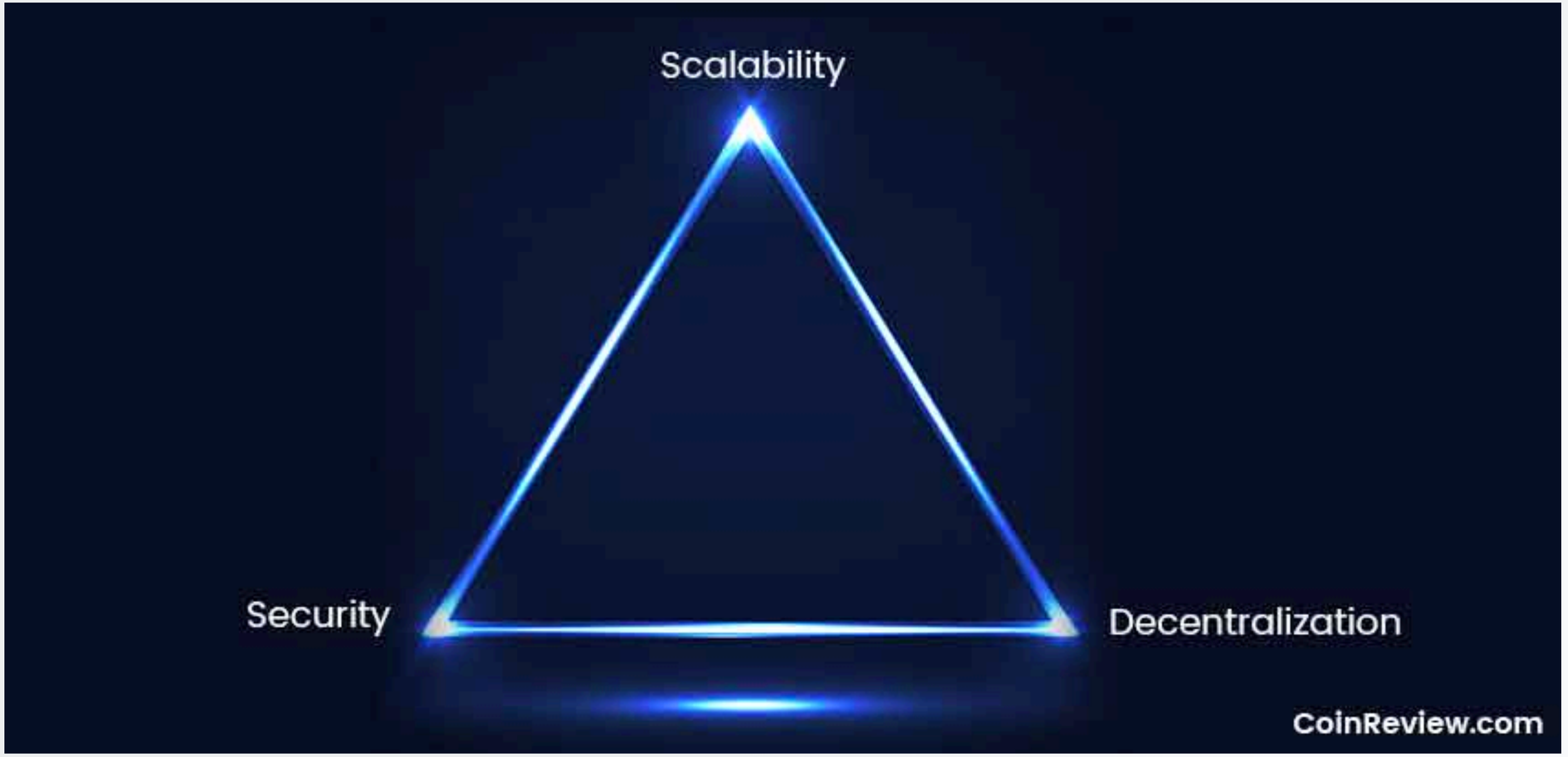
Programmable blockchain

Distributed Apps (DApps)

Automate value transfers based on business rules

OSI-2019

GENERATION 2 - BLOCKCHAIN TRILEMMA



DRAWBACKS OF GEN 2 BLOCKCHAINS

- Very new technology
- Industry best practices are still being formed
- Scalability , transaction cost and power consumption
- Still learning good and bad use cases
- Stigma of blockchain - dark web, ICO scams
- Interoperability

Areas of blockchain research

LIST OF RESEARCH AREAS

- Ricardian Contracts
- Blockchain interoperability
- Privacy Coins
- Stable Coins
- Security Tokens
- Non-Fungible Tokens
- Identity management
- DLT in Supply chains
- Decentralised financial services
- DLT with IOT

Introduction to Ethereum

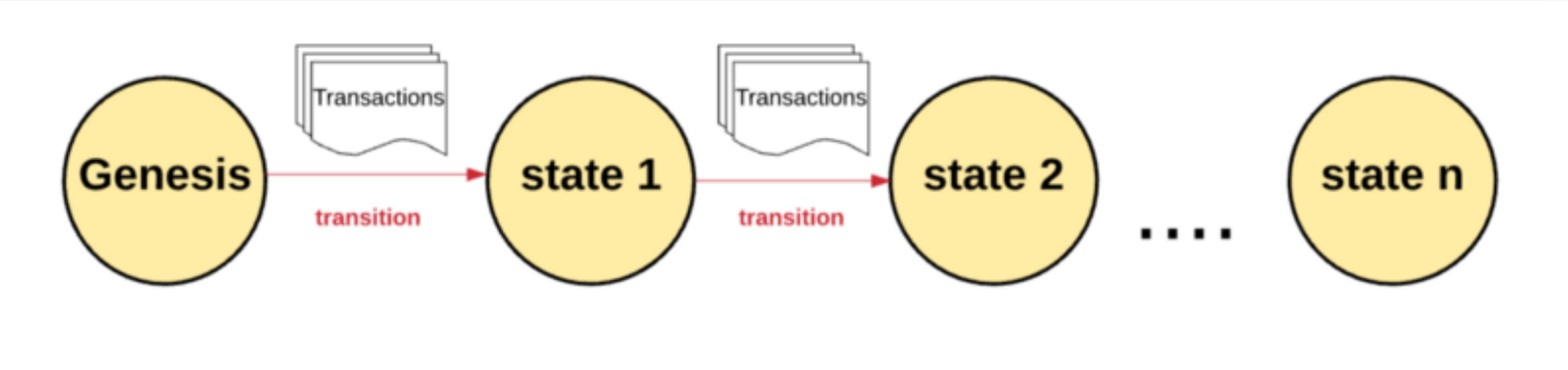
OSI-2019

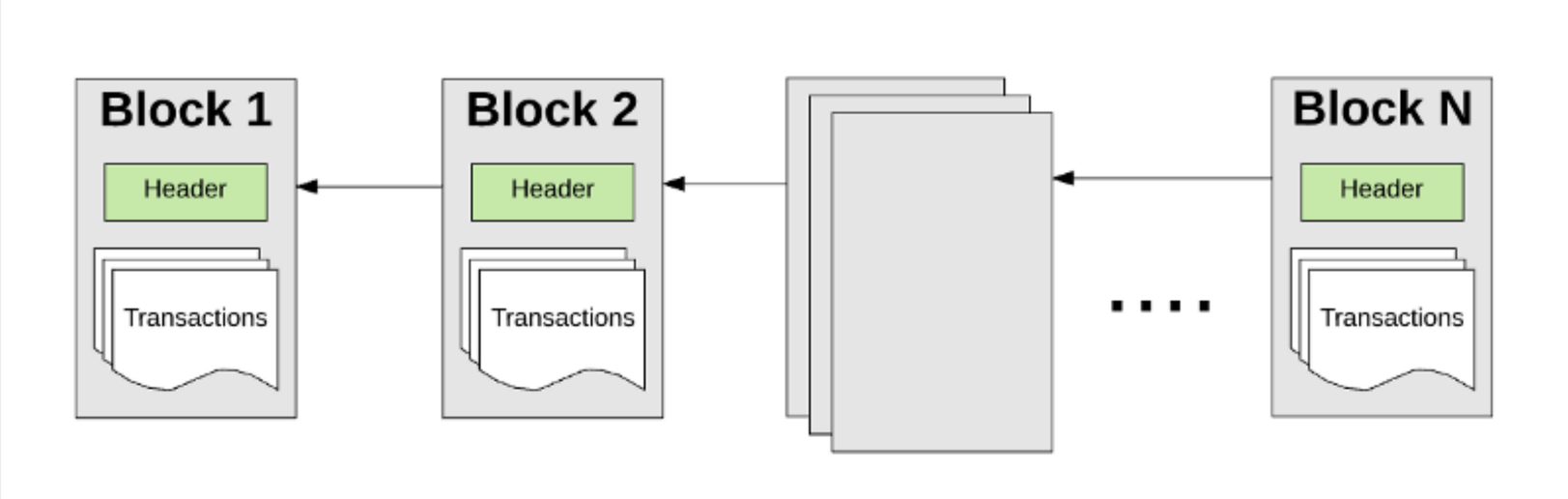
ETHEREUM BASICS

Key concepts in Ethereum

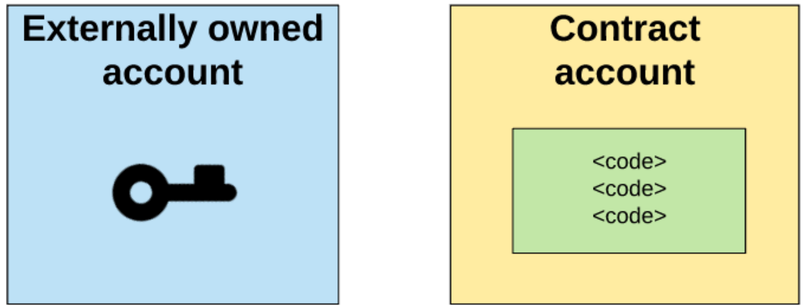
- Blockchain
- Ethereum
- State machine
- Types of accounts
- Smart Contracts
- Cryptography
- Gas
- Transactions
- Blocks
- Mining
- Proof of Work (Ethash)
- ERC20 tokens

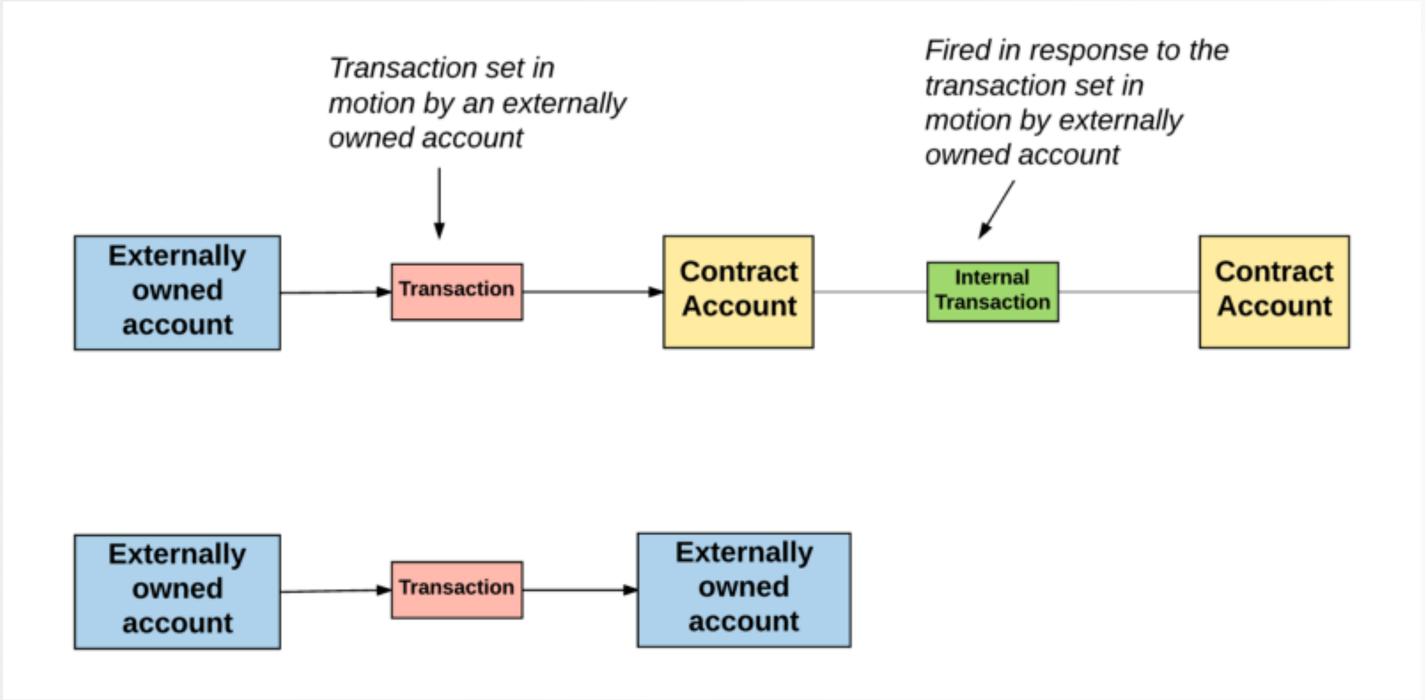
Ethereum is referred to a world-computer



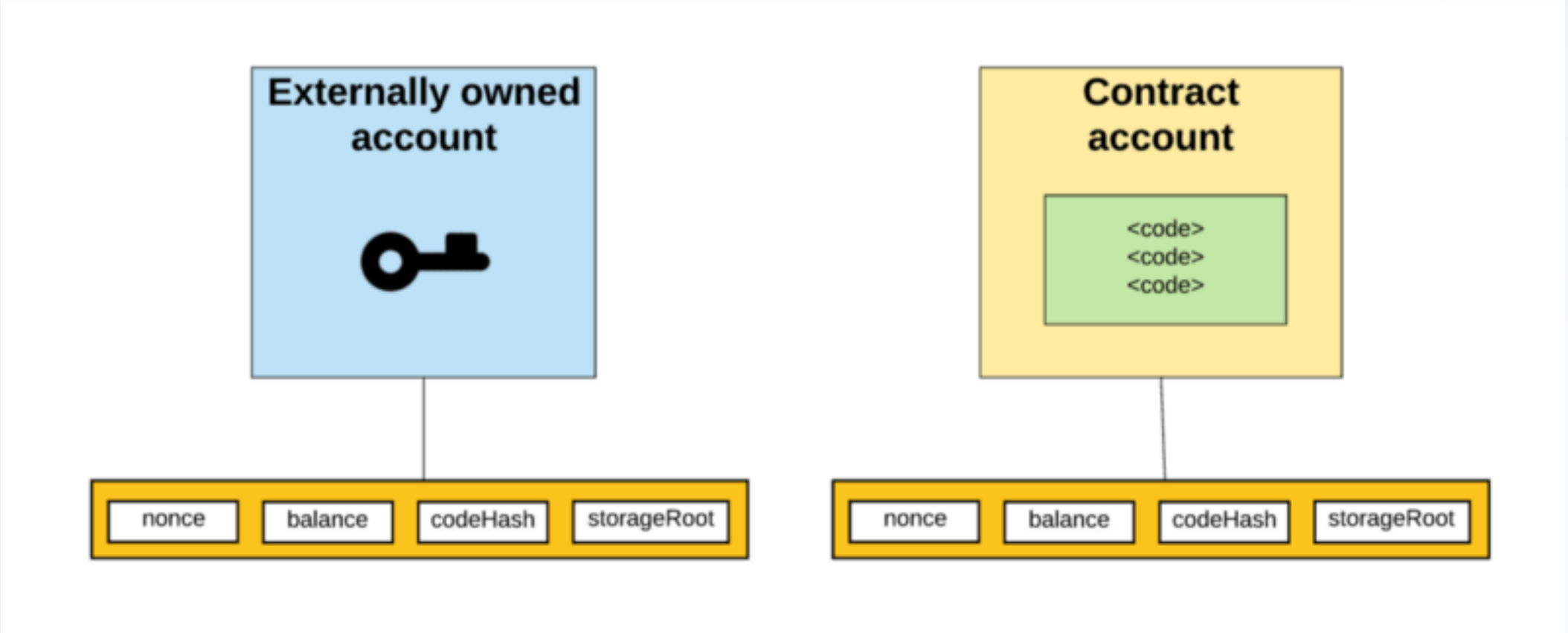


Externally owned accounts vs Contract accounts



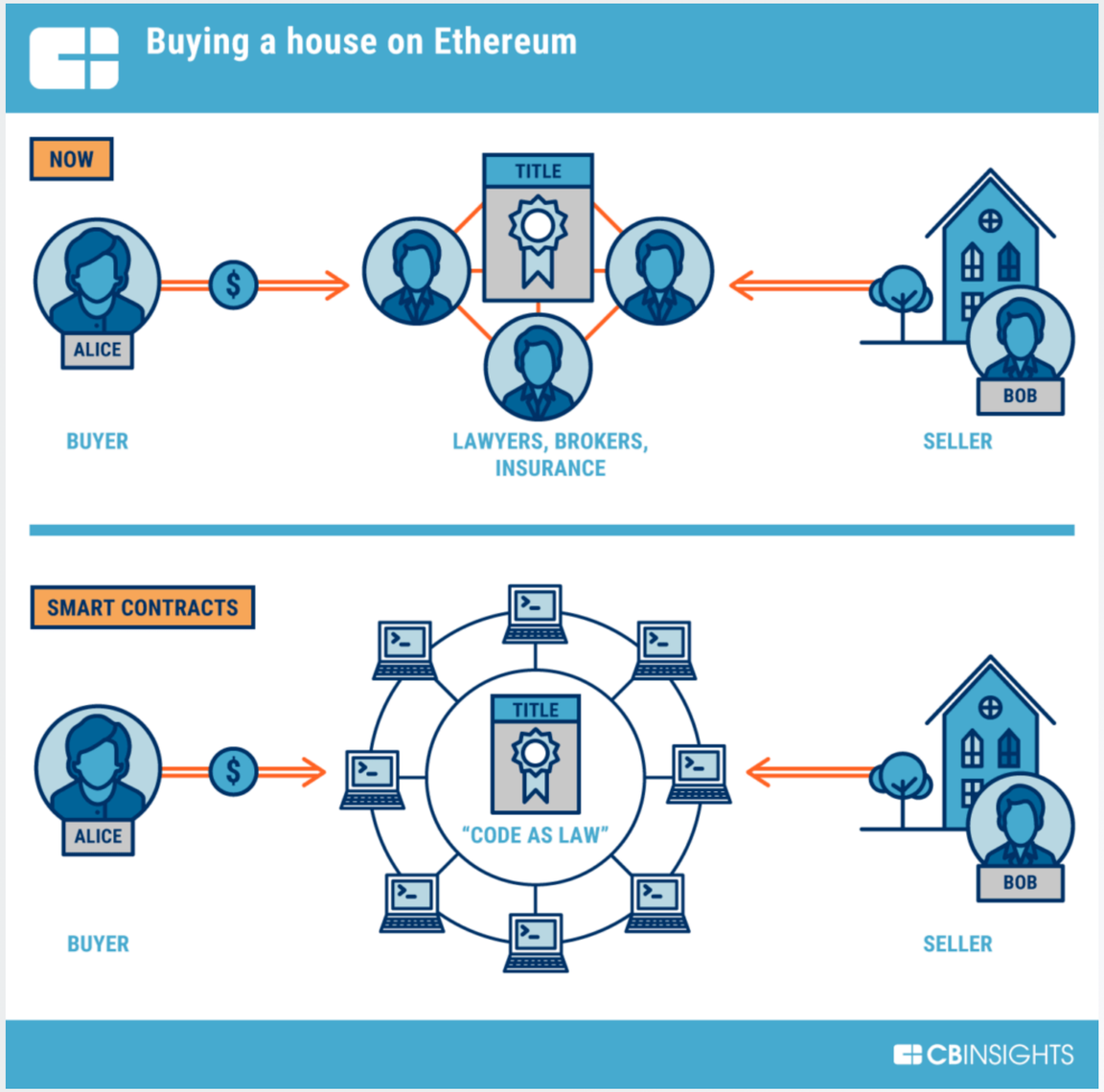


Account State



OSI-2019

ETHEREUM SMART CONTRACTS



ETHEREUM GAS

ETHEREUM CURRENCY UNITS

Units in Ethereum		
Unit	**Number per ETH**	**Most appropriate uses**
Ether (ETH)	1	Currently used to denominate transaction amounts (eg 20 ETH) and mining rewards (5 ETH)
finney	1,000	
szabo	1,000,000	Currently the best unit for the cost of a basic transaction, eg 500 szabo
Gwei	1,000,000,000	Currently the best unit for Gas Prices eg 22 Gwei
Mwei	1,000,000,000,000	
Kwei	1,000,000,000,000,000	
wei	1,000,000,000,000,000,000	The base indivisible unit used by programmers

ETHEREUM NETWORK UPGRADE

Network Upgrade History

Name	Block	Date
Frontier	1	2015-07-30
Frontier Thawing	200000	2015-09-07
Homestead	1150000	2016-03-14
DAO Fork	1920000	2016-07-20
Tangerine Whistle	2463000	2016-10-18
Spurious Dragon	2675000	2016-11-22
Byzantium	4370000	2017-10-16
Constantinople	7280000	2019-02-28

DApps & Examples

CRITERIA FOR DAPPS

There are four key criteria that enable an application to be decentralized:

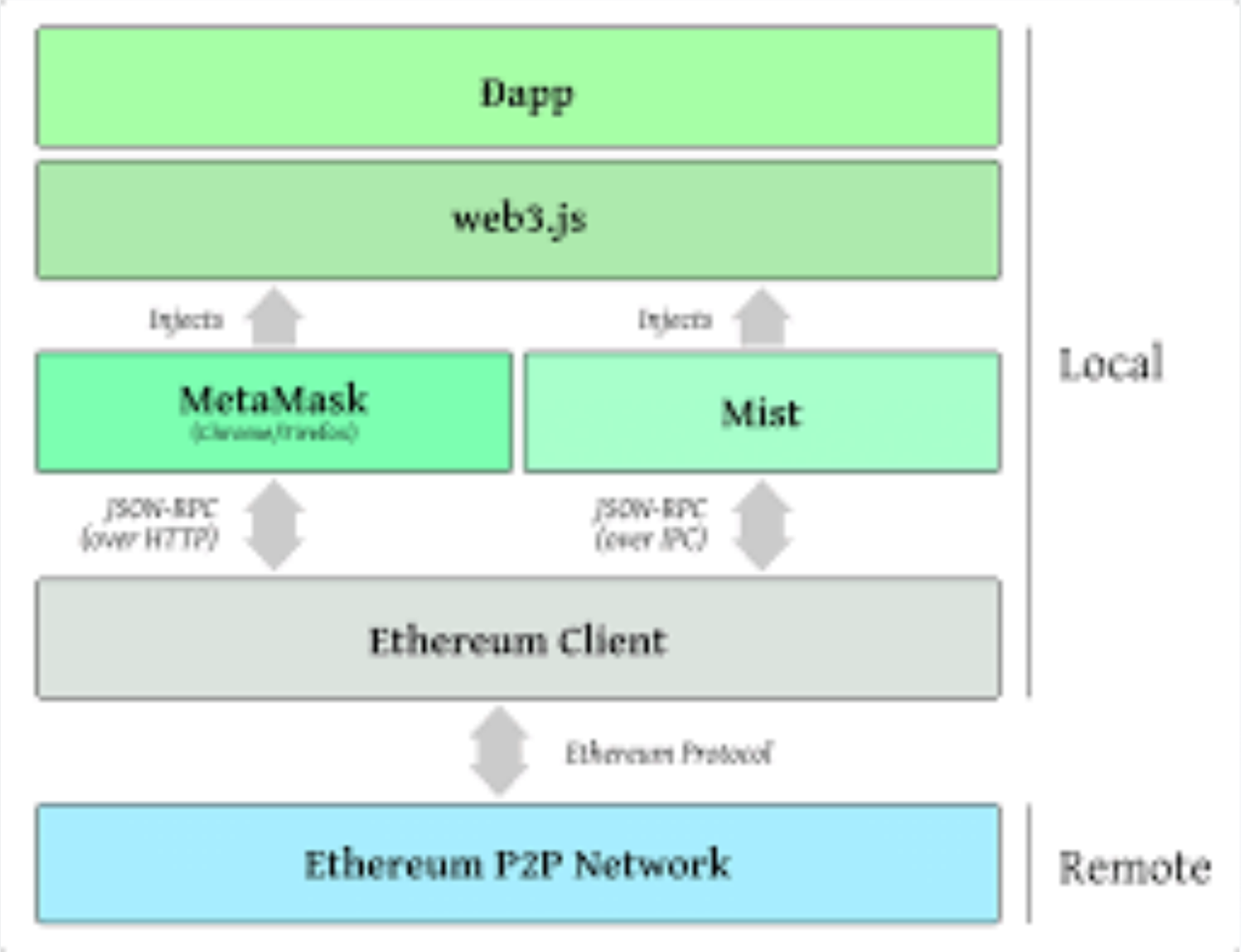
- **Open source:** The application's code must be available for public scrutiny. Changes to the protocol decided by consensus.

- **Decentralized:** The application must function on a decentralized blockchain network, which allows for data to be cryptographically stored and distributed in the decentralized ledger, enabling every user on the network to see it.

- **Incentive:** Like blockchain, the application must use and provide specific tokens or digital assets to users on the network to reward them for validating blocks on the chain.

- **Protocol:** The application must run on a protocol, and the community needs to agree on a cryptographic algorithm such as POW or POS.

DAPP CONCEPTUAL ARCHITECTURE



DAPPS EXAMPLES

DAPPs	Description
Golem	Opensource decentralised supercomputer that anyone in world can access. Users can rent out their computing power to other users.
Augur	Prediction market
Melonport	DAPP that provides an entirely new approach to digital assets management.
EtherTweet	Decentralised blogging platform similar to Twitter
Factom	Decentralised record-keeping system
Storj	Decentralised p2p protocol for secure, private cloud storage

Solidity Basics

STRUCTURE OF SMART CONTRACT - 1

Parts of a contract	Sample code

State Variables	`pragma solidity >=0.4.0 <0.7.0;`
	
	`contract SimpleStorage {`
	`  uint storedData; // State variable`
	`  // ...`
	`}`

Functions	`pragma solidity >=0.4.0 <0.7.0;`
	
	`contract SimpleAuction {`
	`  function bid() public payable { // Function`
	`  // ...`
	`  }`
	`}`

STRUCTURE OF SMART CONTRACT - 2

Parts of a contract	Sample code

Events	`pragma solidity >=0.4.21 <0.7.0;`
	
	`contract SimpleAuction {`
	`  event HighestBidIncreased(address bidder, uint amount); // Event`
	
	`  function bid() public payable {`
	`  // ...`
	`  emit HighestBidIncreased(msg.sender, msg.value); // Triggering event`
	`  }`
	`}`

OSI-2019

SOLIDITY DATA TYPES

Data	Sample code
Value types	**Boolean : true or false** **Integers (int, uint): int8 to int256, uint8 to uint256** **Fixed point numbers:** Not full supported **Address/ Address payable:** 20 byte value (size of an Ethereum address) **.** Methods are balance, transfer and send. **Contract types:** Contracts have their own type. **Fixed size byte arrays:** bytes1 … bytes32 **enum:** user-defined type

SOLIDITY DATA TYPES

Data	Sample code

Reference types	**Struct:**
	`struct Funder {`
	`address addr;`
	`uint amount;`
	`}`
	
	Array: Can have compile-time fixed size or dynamic size. Variables of type bytes and strings are special arrays
	
	`uint[] memory a = new uint[](7);`
	
	`bytes memory b = new bytes(len);`
	
	Mapping: Hash tables.
	
	`mapping(address => uint) public balances;`

SOLIDITY INTRODUCTION - DEMO

Switch to Remix editor

https://remix.ethereum.org

Overview of DApps tools and ecosystem

APP TOOLS & ECOSYSTEM - FOUNDATIONAL TOOLS

Tool	Description
Solidity	Most popular smart contract language
Truffle	Most popular smart contract development, testing and deployment framework
Metamask	Chrome extension wallet to interact with DApps
Infura	Ethereum node on the cloud

APP TOOLS & ECOSYSTEM - DEVELOPER TOOLS

Tool	Description
Remix	Web IDE to develop in solidity. Free.
Ganache	Private Test Ethereum network used in dev cycle
Rinkeby, Kovan, Ropsten	Public Test Ethereum networks
Web3.js	Javascript library to communicate with Ethereum node APIs

APP TOOLS & ECOSYSTEM - OTHER TOOLS/LIBRARIES

Tool	Description
OpenZeppelin	Reusable and secure smart contracts in solidity
Ethereum clients	Geth - Go client Pantheon - Java client Parity - Rust client Pyethapp - Python client
IPFS, Swarm	Decentralised storage
Whisper	P2P communication protocol for Apps to communicate with each other

Best practices in DApp design

BEST PRACTICES IN DAPP DESIGN & DEVELOPMENT

Tool	Description
Identity/Authentication	Do not take control of your user’s private keys
Digital signature	Do not sign on behalf of your users in a central server
Storage	Don't put images and large files on blockchain. Use IPFS.
Testing	Deploy DApp on public testnet before mainnet.

BEST PRACTICES IN DAPP DESIGN & DEVELOPMENT

Tool	Description
Oracle	Use oracle to gain access to data stored outside blockchain. Eliminate all sources of randomness.
Access restriction	Restrict access to smart contract by suitable criteria
Emergency stop	Add functionality to stop smart contract execution in case of emergencies
Upgradeability	Use patterns such as proxy delegate and external storage to allow for upgradeability of smart contracts

Live demo - Building a DApp