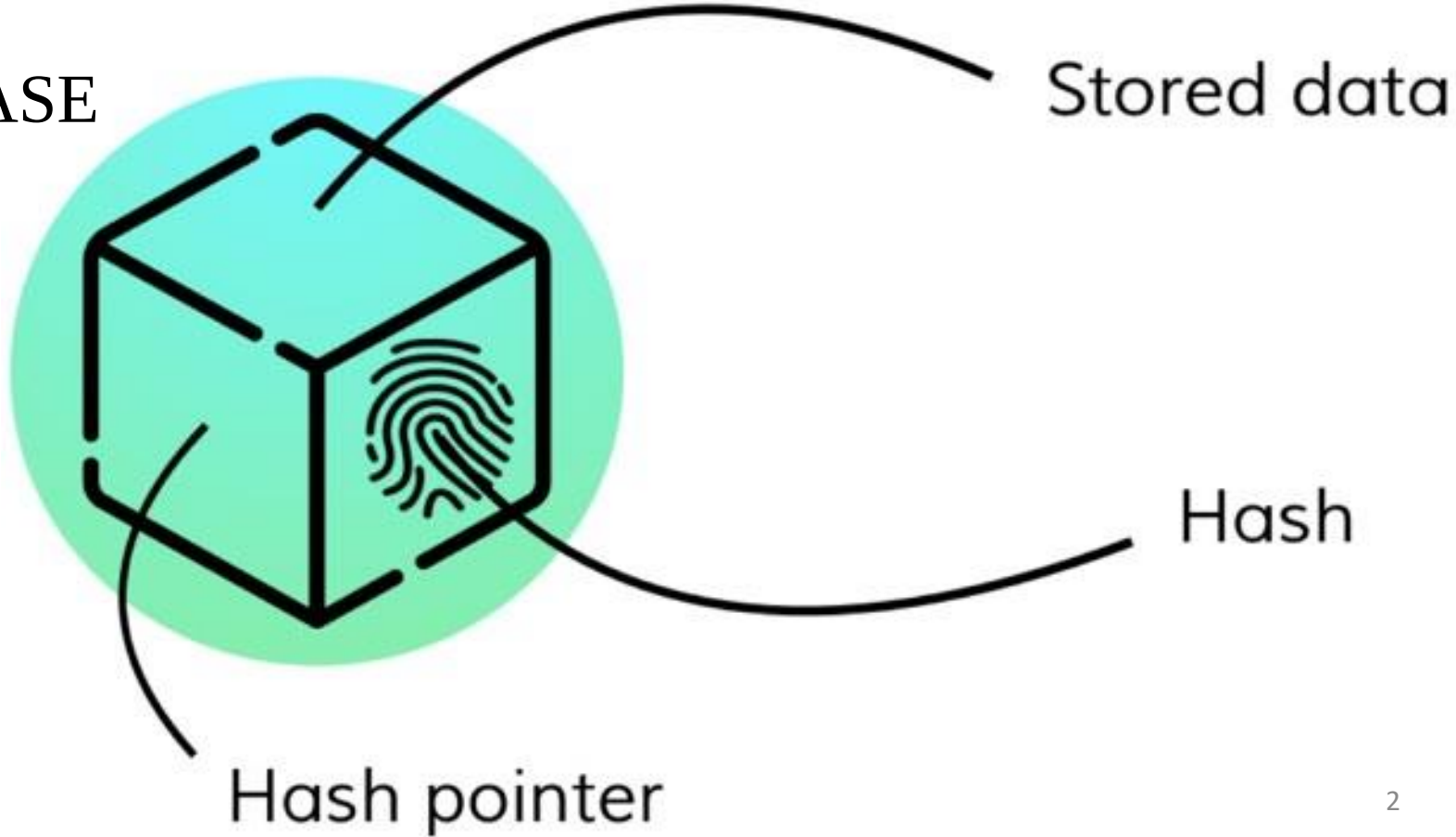
UNIT 3

BLOCKCHAIN DATA

Lecturer: Ph.D Lê Quang Huy

CONTENTS



1. DATA

1.1. DATA

1.2. DATA TYPE

1.3. DATA STRUCTURES

1.4. MERKEL TREE

1.5. DATA PROCESSING

1.6. INFOMATION

1.1. DATA

Data is:

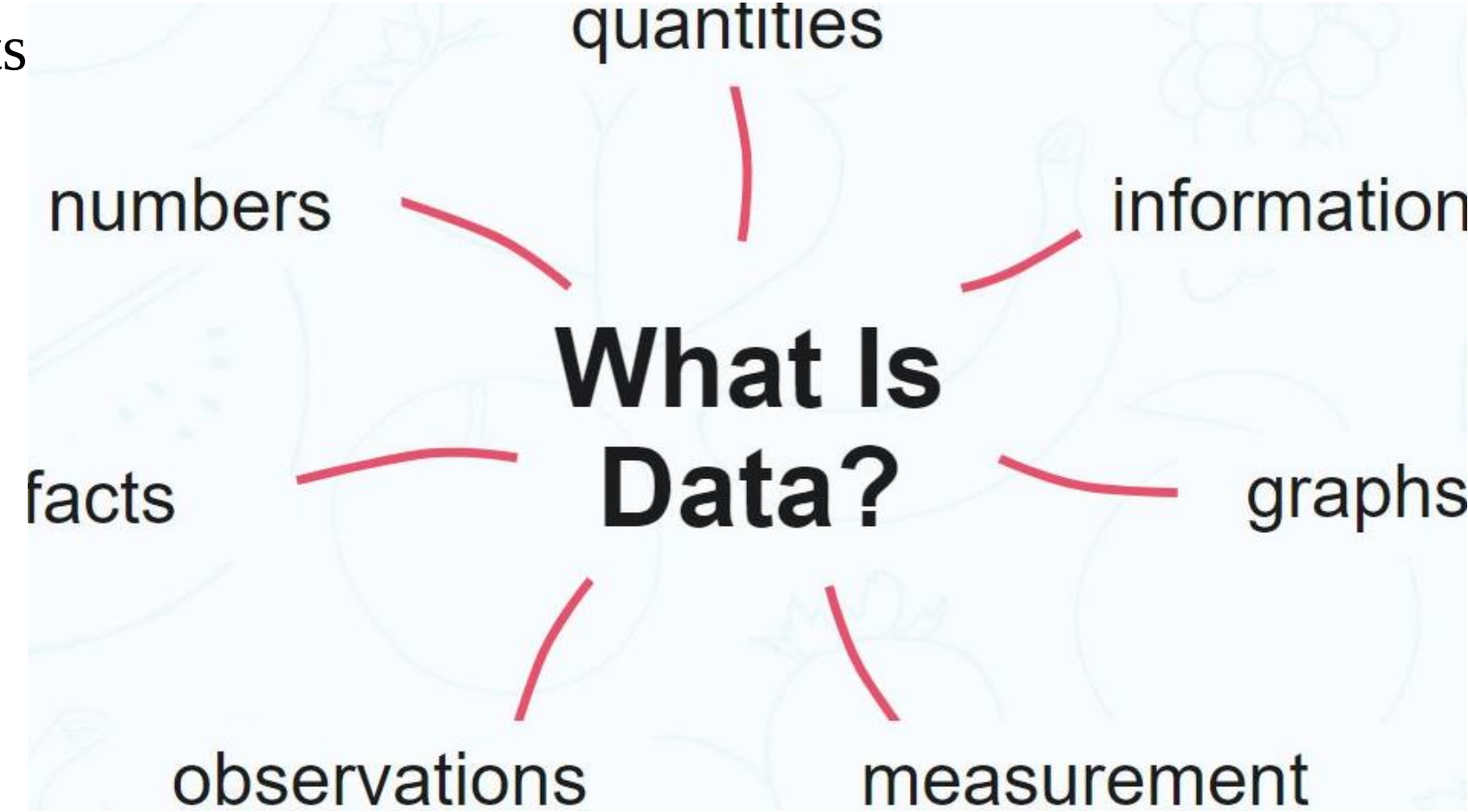
- a collection of discrete values
- sequences of symbols.
- raw and unorganized facts

Data represent:

- abstract ideas
- concrete measurements

Data

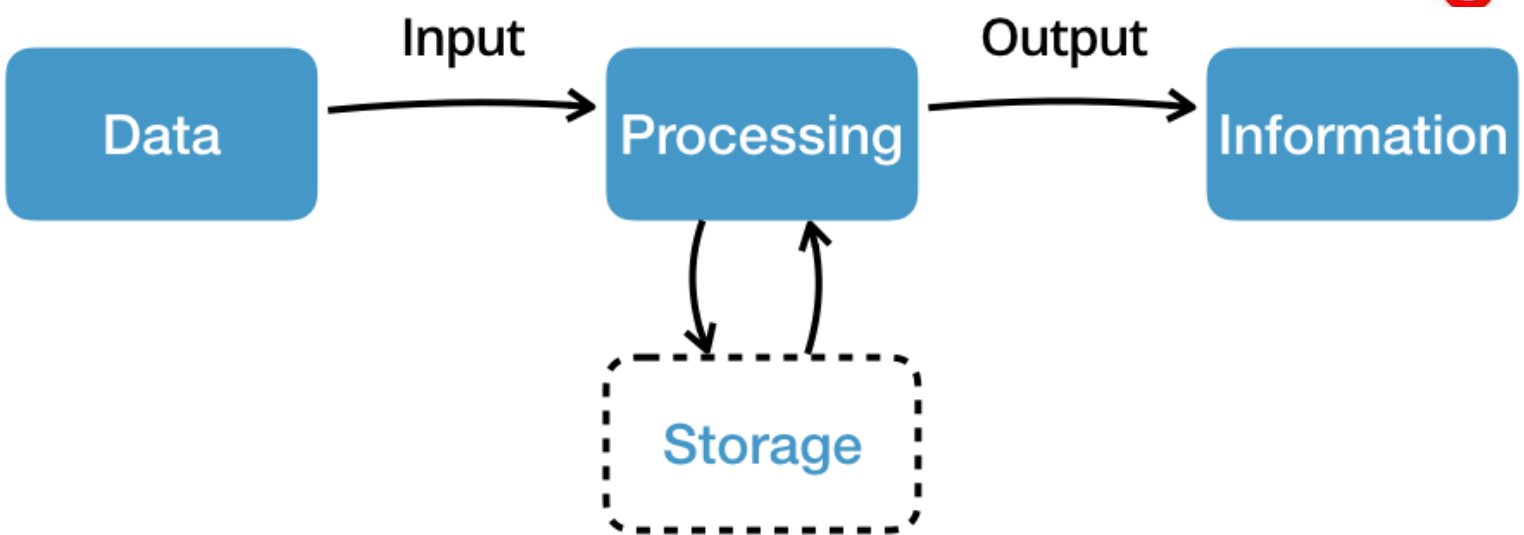
- has to be processed
- make it meaningful



1.1. DATA

Data components:

- Attribute: property/characteristic
- Attribute values: value (type).
- Object: are described as collection of attributes
- Data set: collection of objects



Attributes

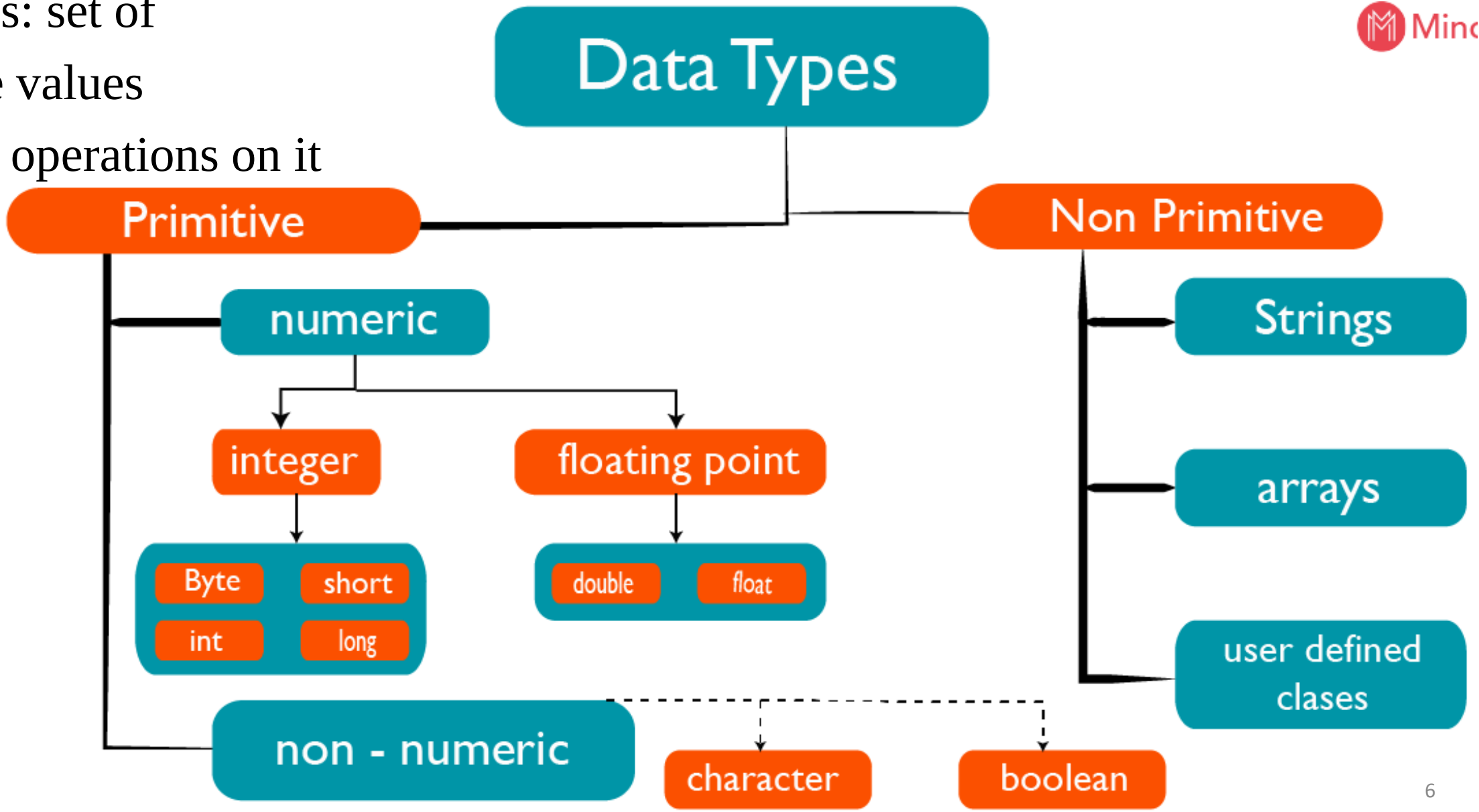
Objects

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

1.2. DATA TYPE

Data types: set of

- possible values
- allowed operations on it



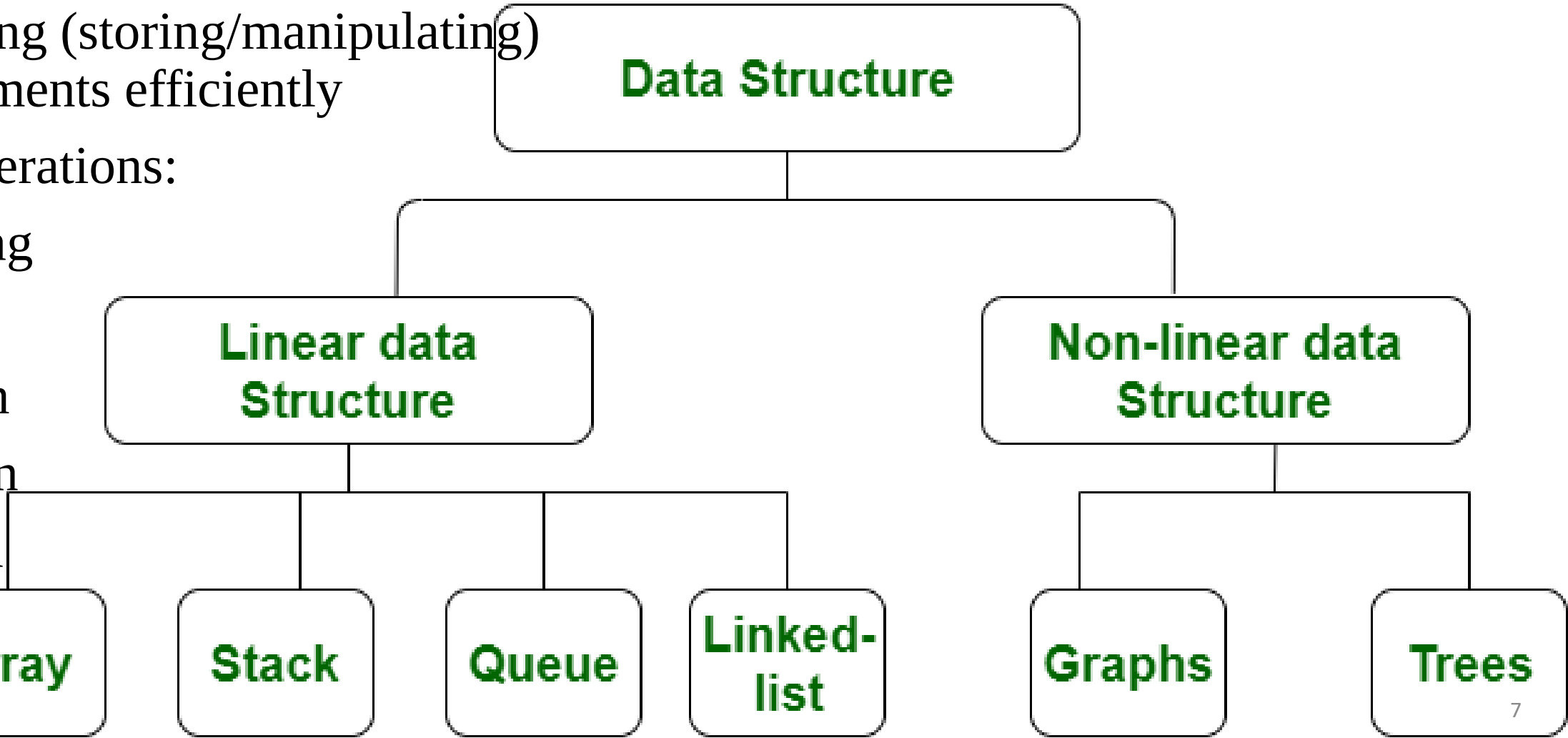
1.3. DATA STRUCTURES

Data structures:

- collection of elements of data type
- organizing (storing/manipulating) data elements efficiently

Major Operations:

- Searching
- Sorting
- Insertion
- Updation
- Deletion

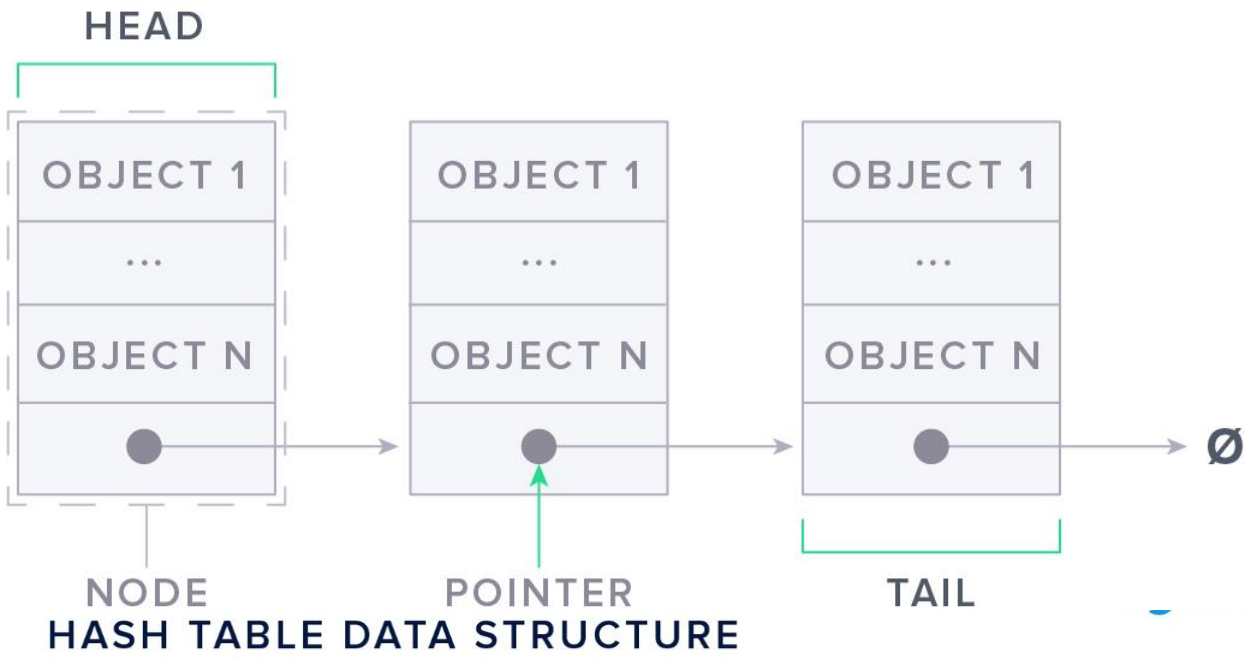


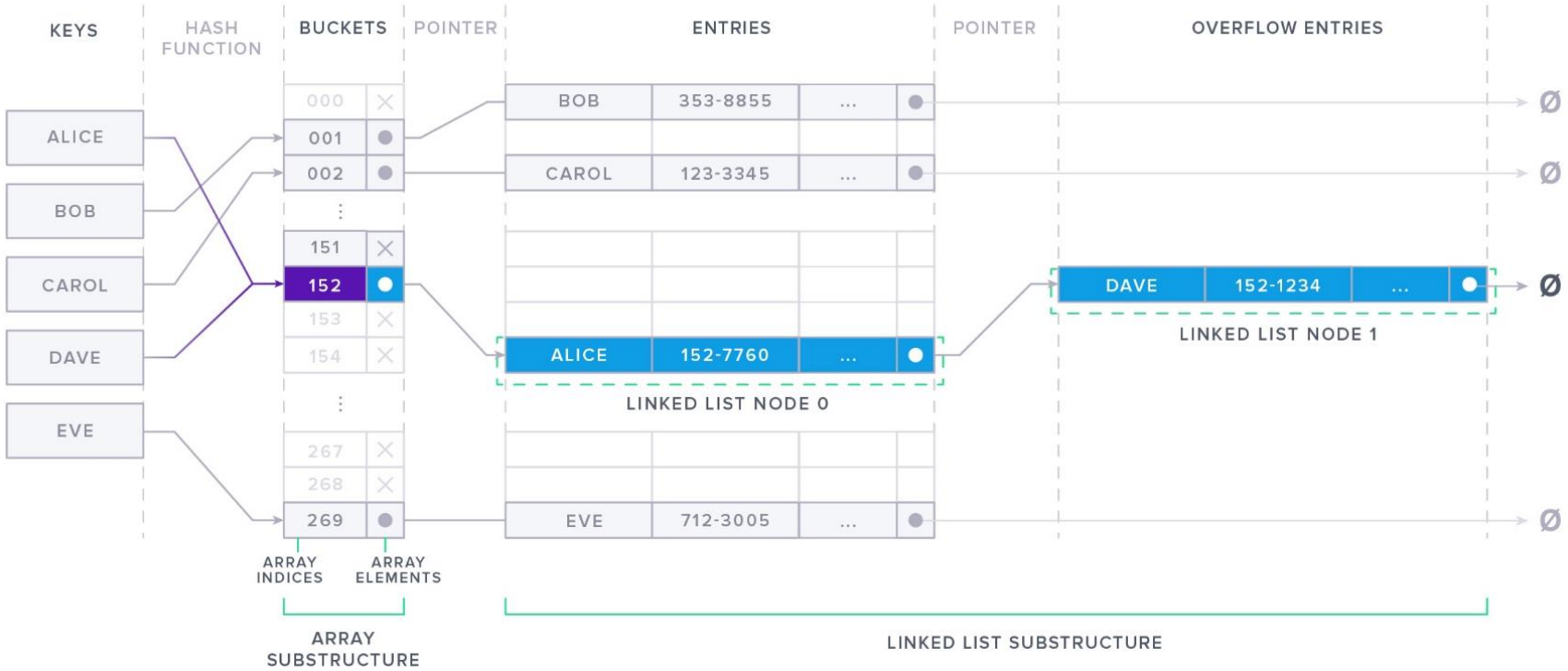
1.4. MERKEL TREE

LINK LIST

- Concept
- Use
- Benefits

HASH TABLE





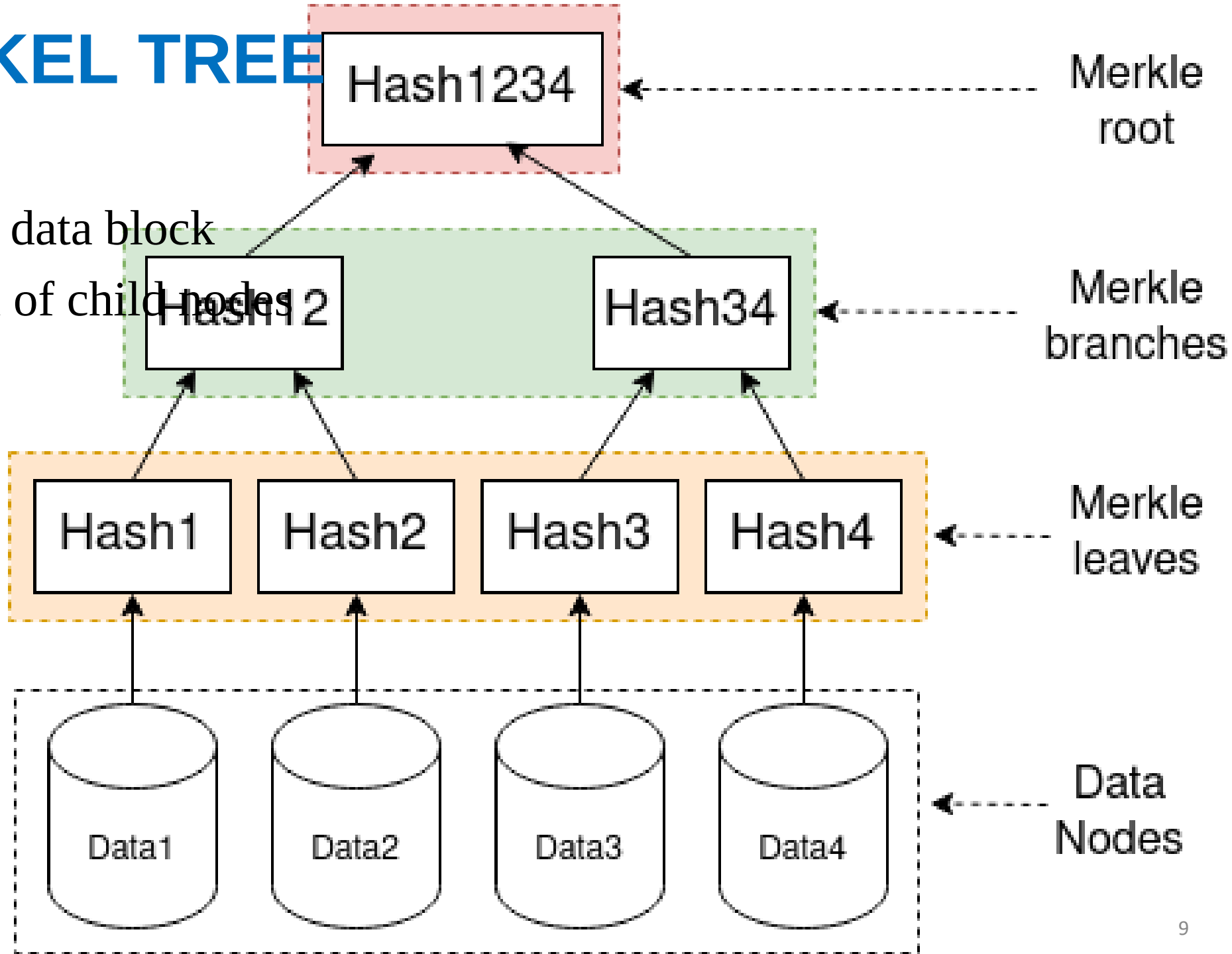
1.4. MERKEL TREE

Merkle Tree is a tree

- leaf node: hash of a data block
- non-leaf node: hash of child nodes

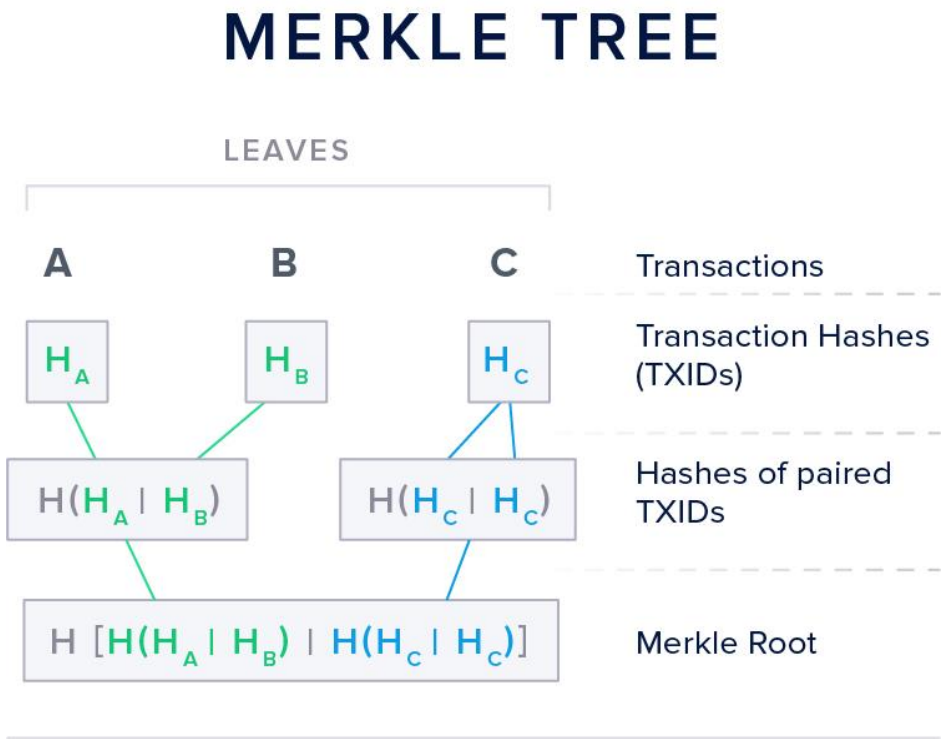
Operations:

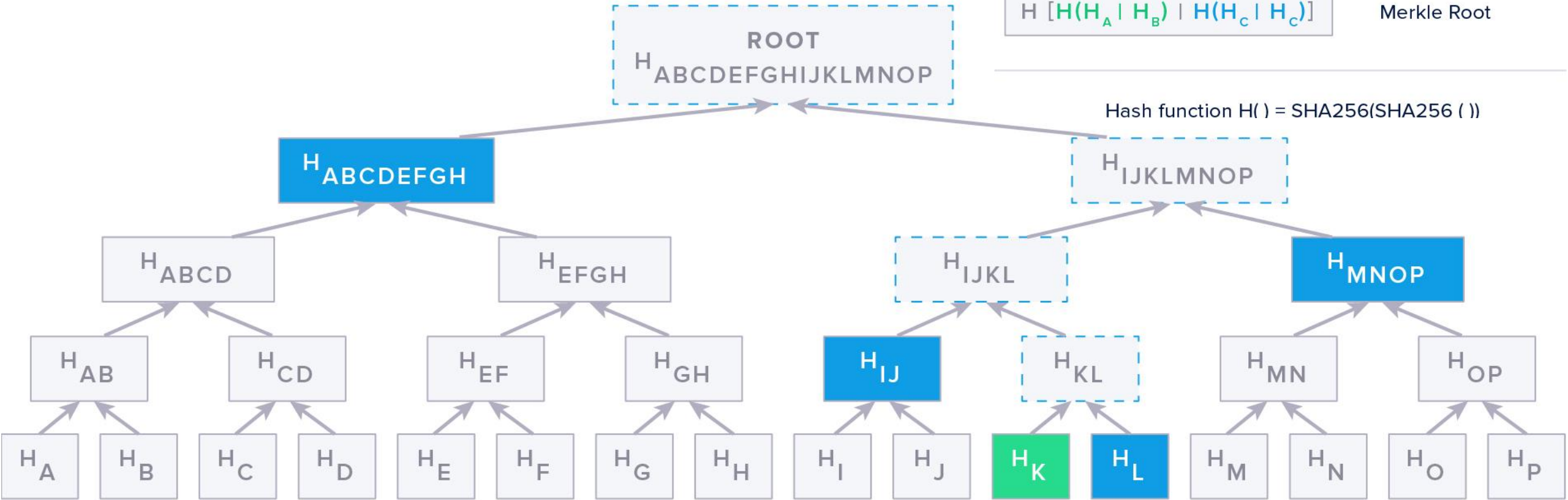
- Search O(log N)
- Traverse O(n)
- Insertion O(n)
- Deletion O(n)



1.4. MERKEL TREE

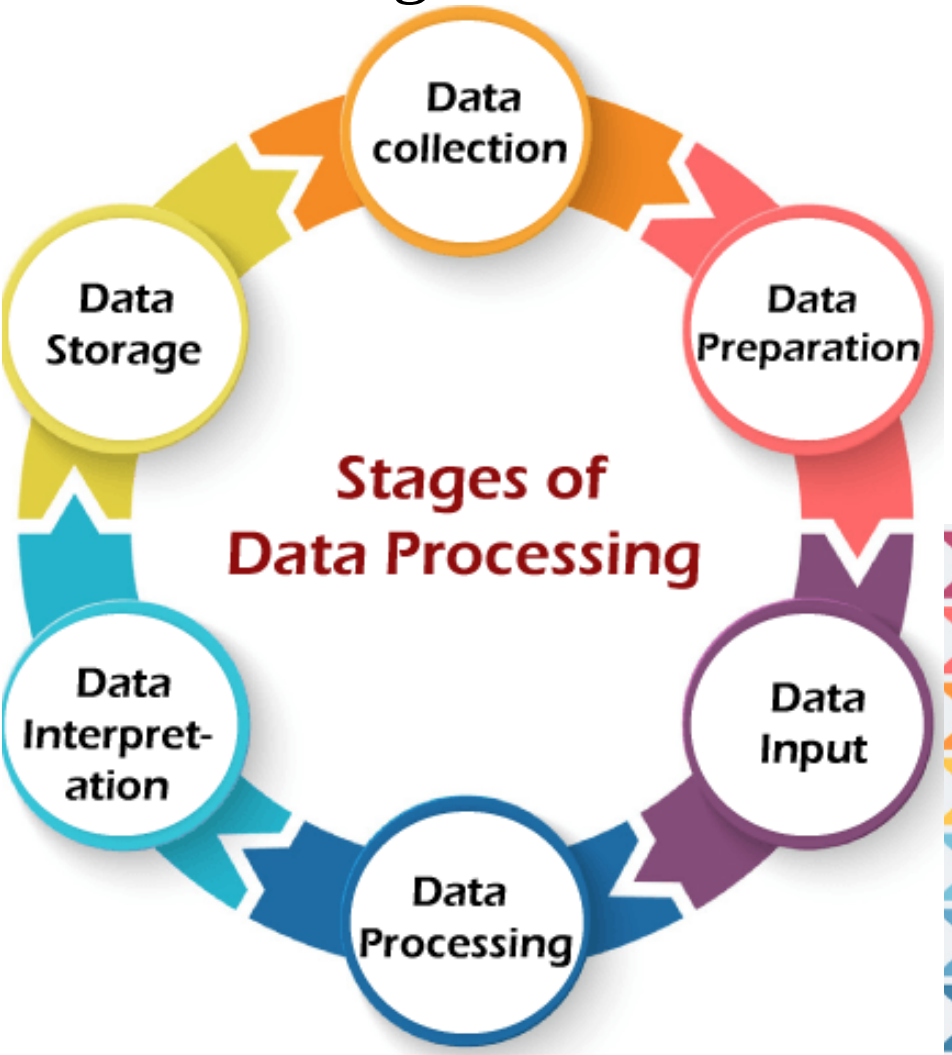
Proof of Membership

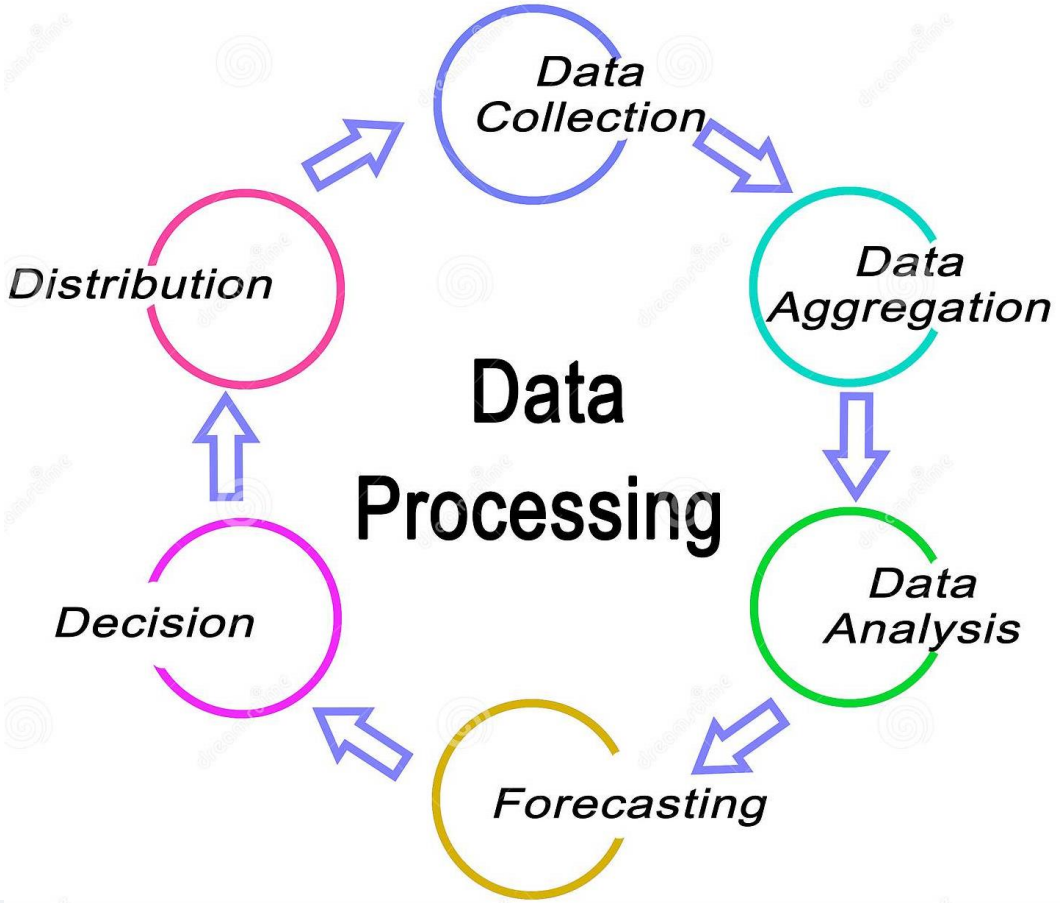




1.5. DATA PROCESSING

- collecting raw data
- translating into usable information

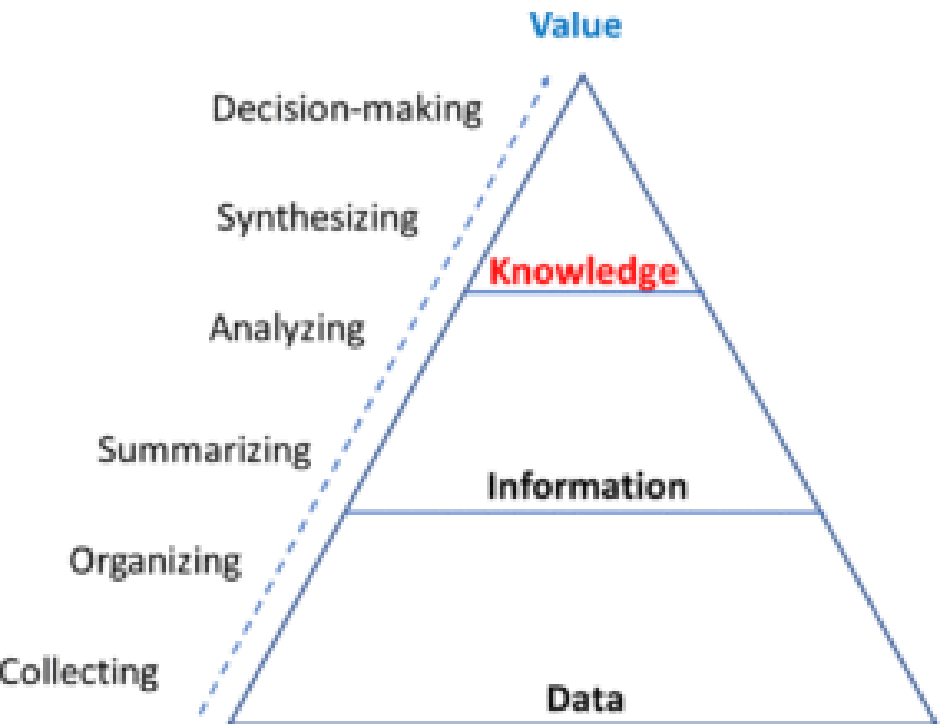


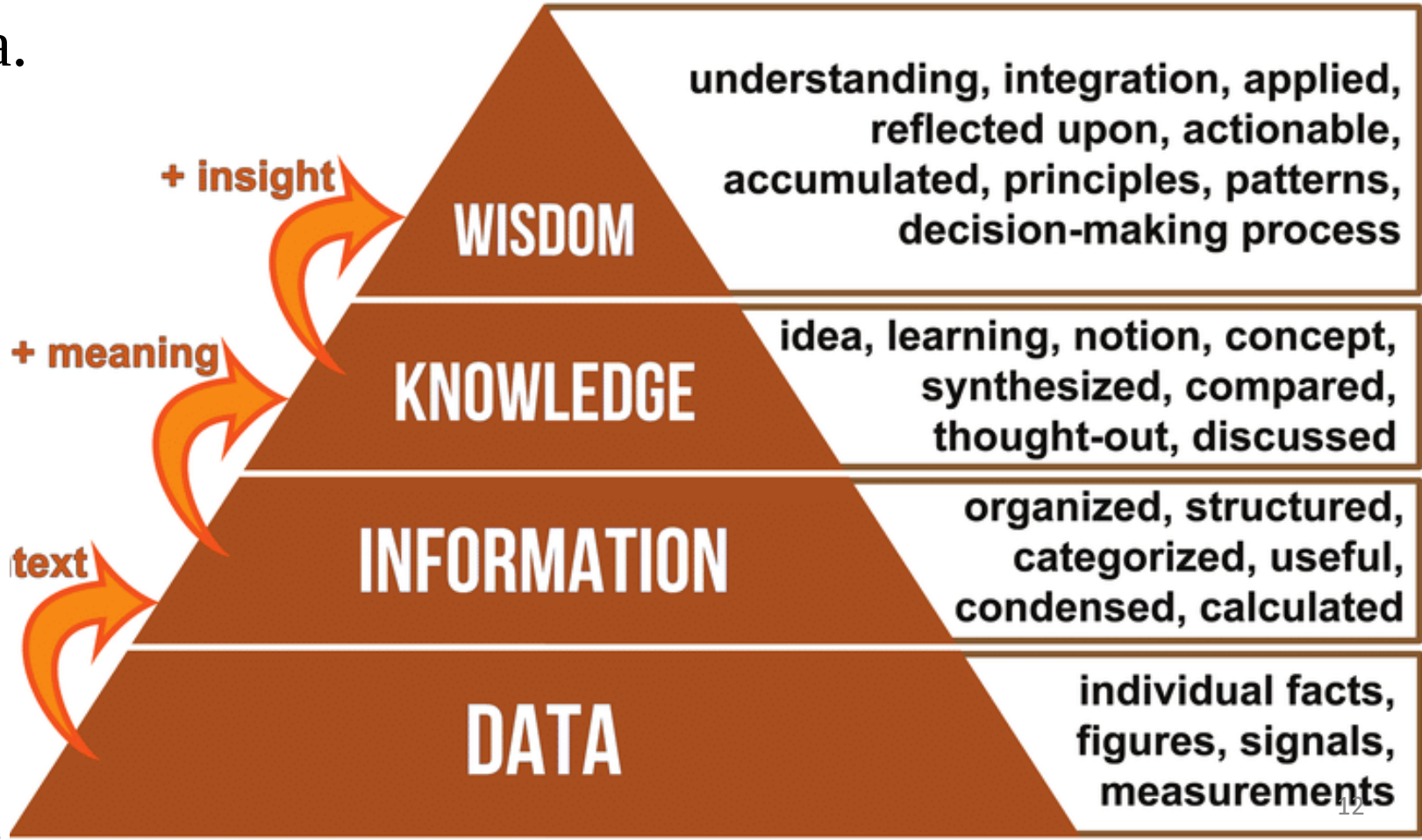


1.6. INFOMATION

Information is

- created when: data are processed, organized, or structured to provide context and meaning
- is essentially processed data.





2. BLOCKCHAIN DATA

2.1. IDENTITY & ADDRESS

2.2. ACCOUNTS

2.3. TRANSACTIONS

2.4. BLOCKS

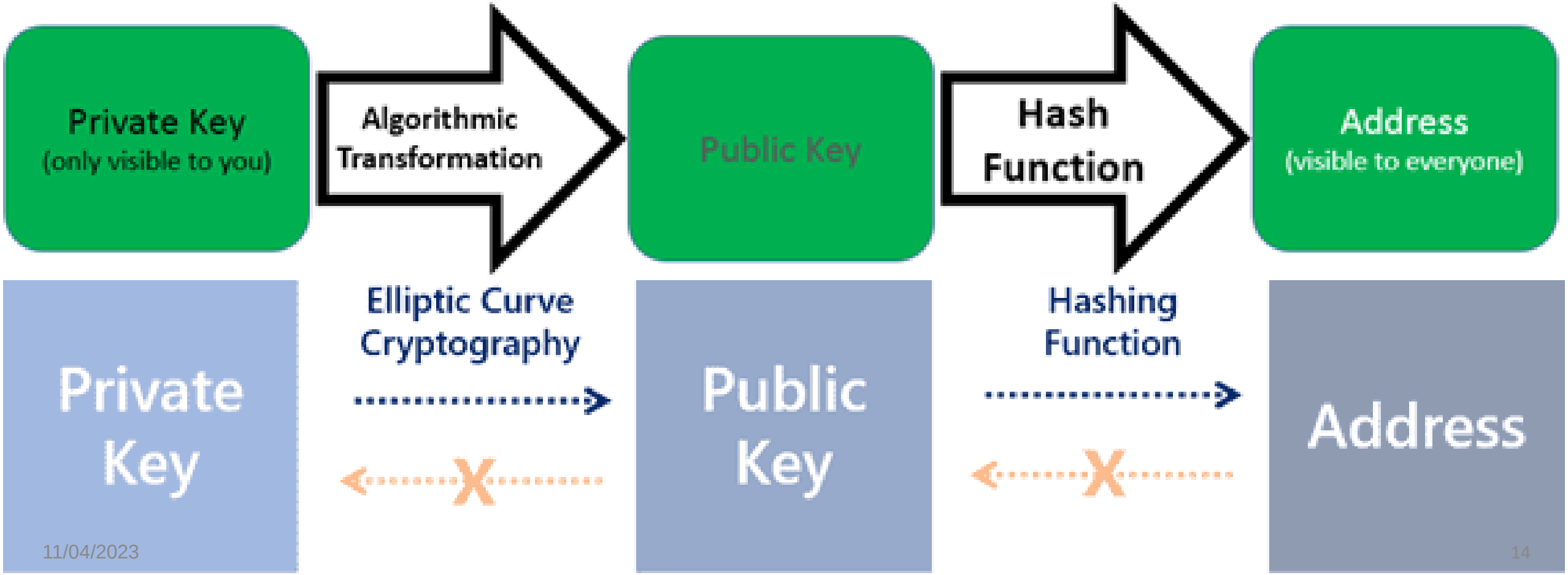
2.5. BLOCKCHAIN

2.6. BLOCKCHAIN STATES

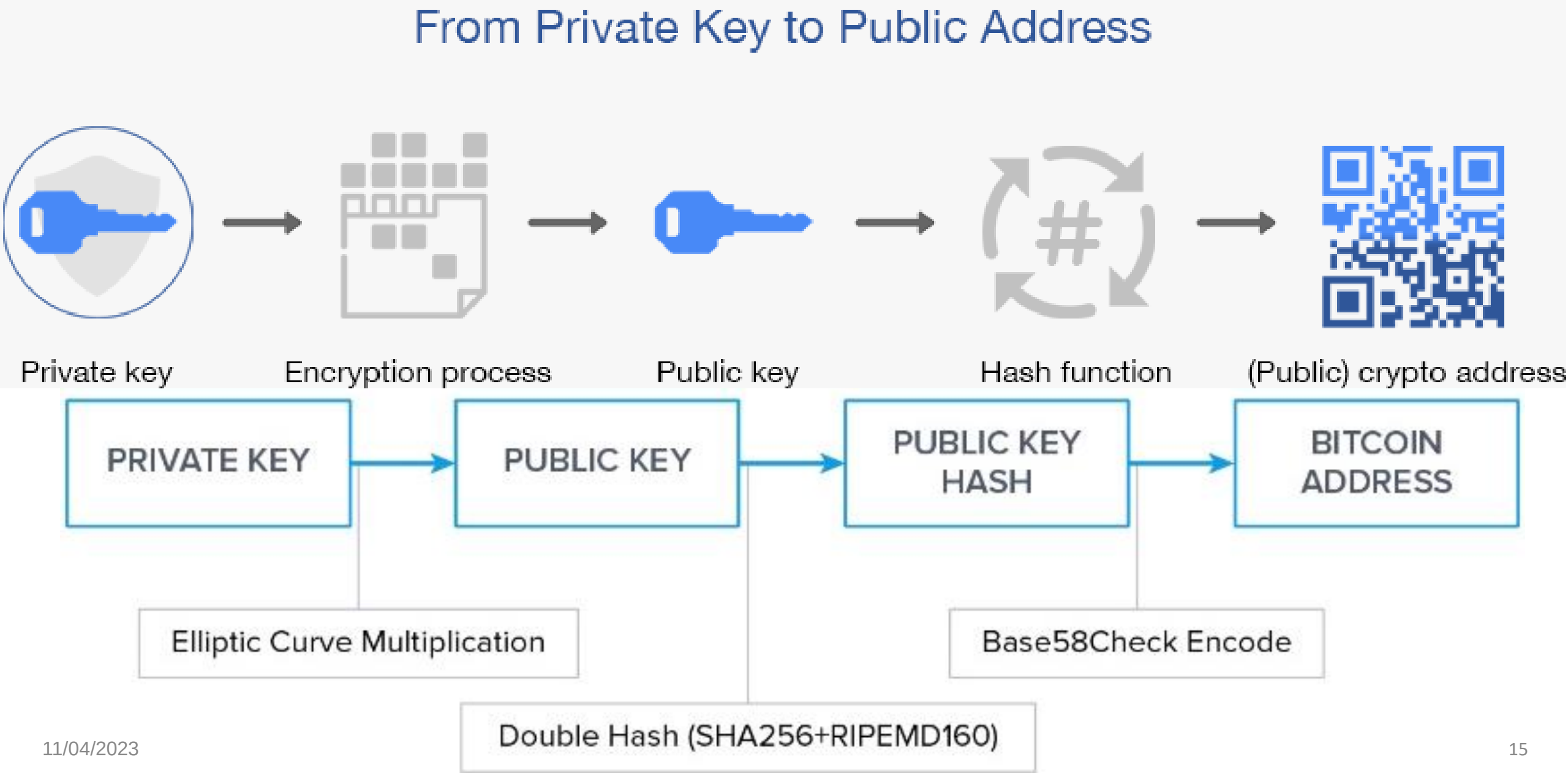
2.7. TRANSACTIONS ACCOUNTING MODEL

2.1. IDENTITY & ADDRESS

- Address: Identify an entity
- Keypairs: sign (signature): party/owner aggree to do



2.1. IDENTITY & ADDRESS



2.2. ACCOUNTS

Account

- is a record
- tracks the financial activities of asset, liability, equity revenue, expense

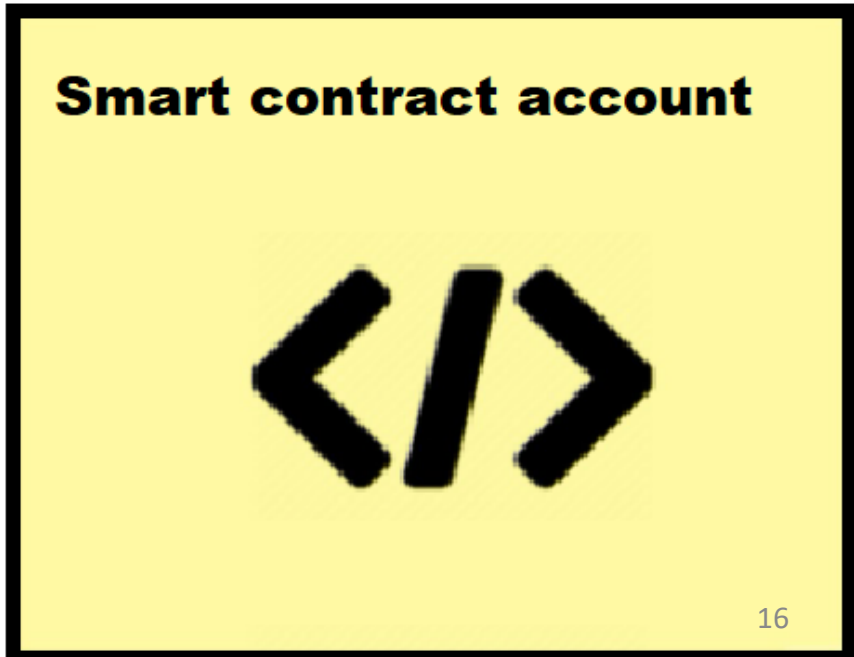
Blockchain account.

- represents assets as balances within accounts similar to bank accounts

Ethereum Accounts

➡ **Accounts are classified into two main types:**

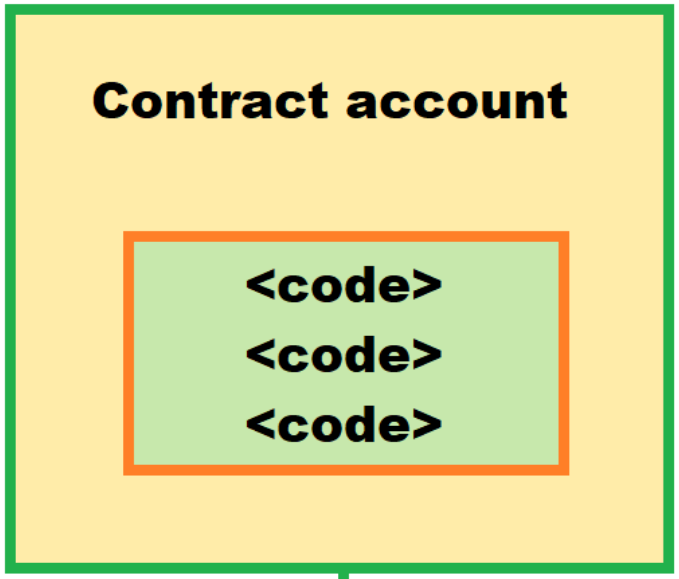
1. **Externally Owned Accounts**
2. **Contract Accounts**



2.2. ACCOUNTS

- Account is identified by address (derived from public key - EOA account)

Externally Owned Account Vs Contract Account



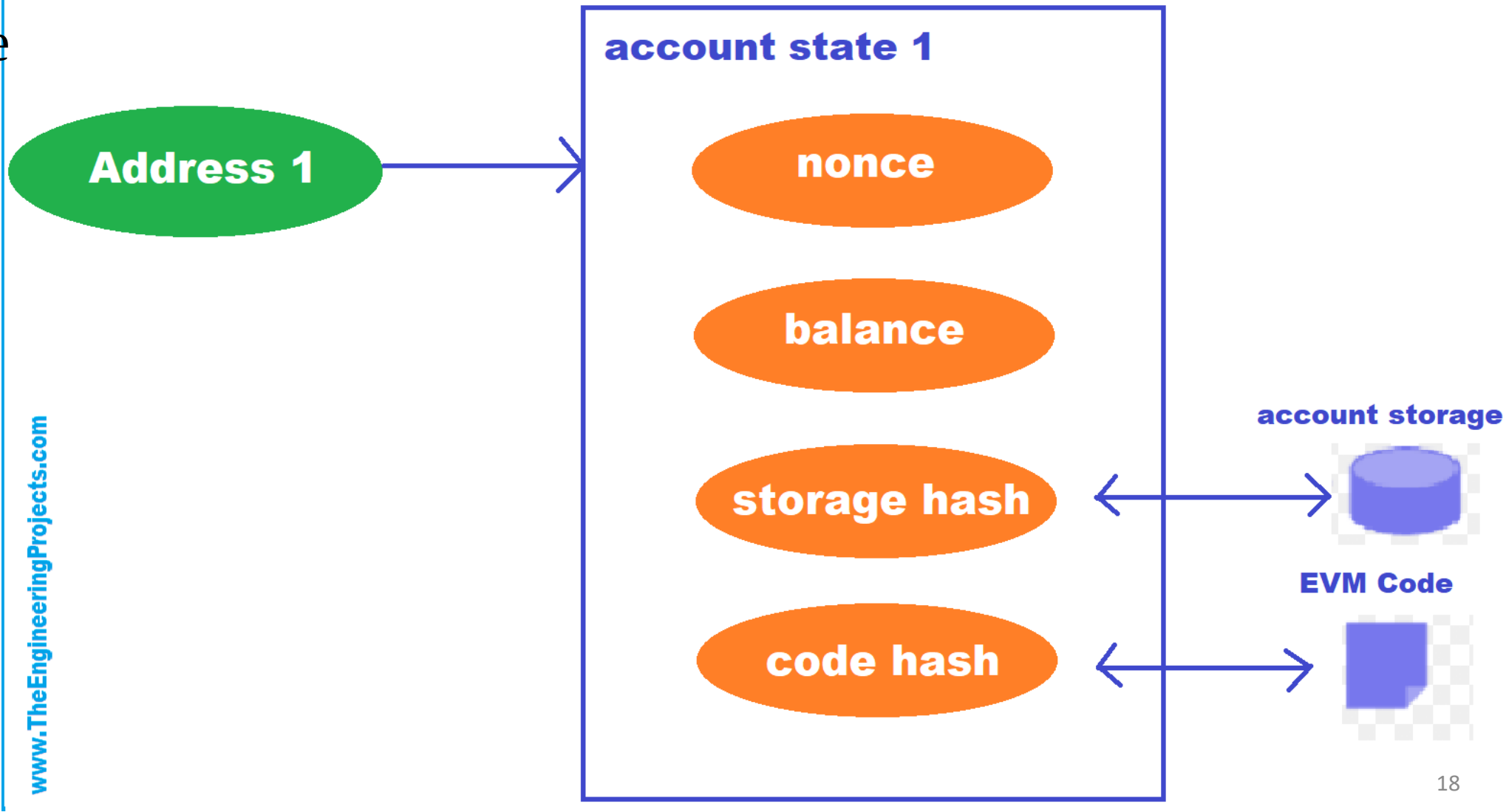
Blockchain account type:

- Private key controlled user accounts
- Contract code-controlled accounts

2.2. ACCOUNTS

- public key acts as the identity of the EOA account

Fields of an Ethereum Account



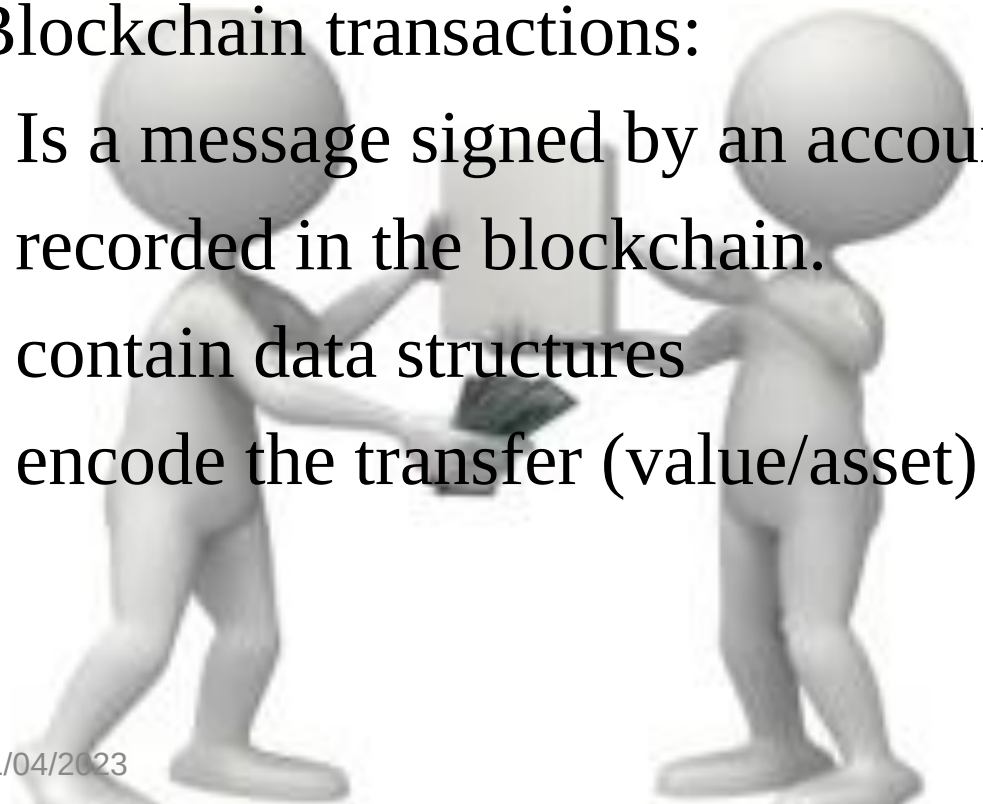
2.3. TRANSACTIONS

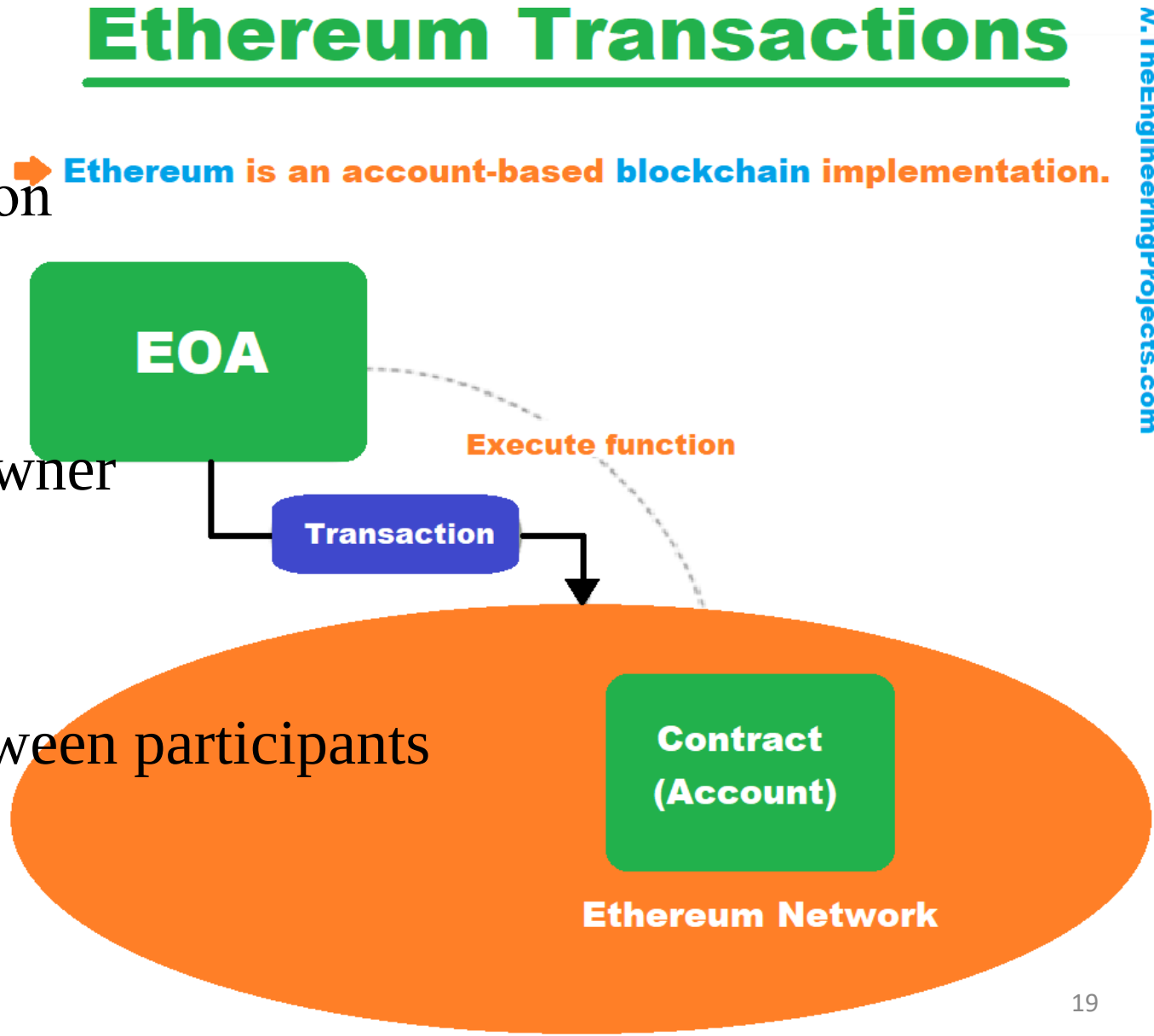
Transaction: events change in

- Financial position/assets, liabilities
- Owner's equity of person/organization
- Unit of actions

Blockchain transactions:

- Is a message signed by an account owner
- recorded in the blockchain.
- contain data structures
- encode the transfer (value/asset) between participants





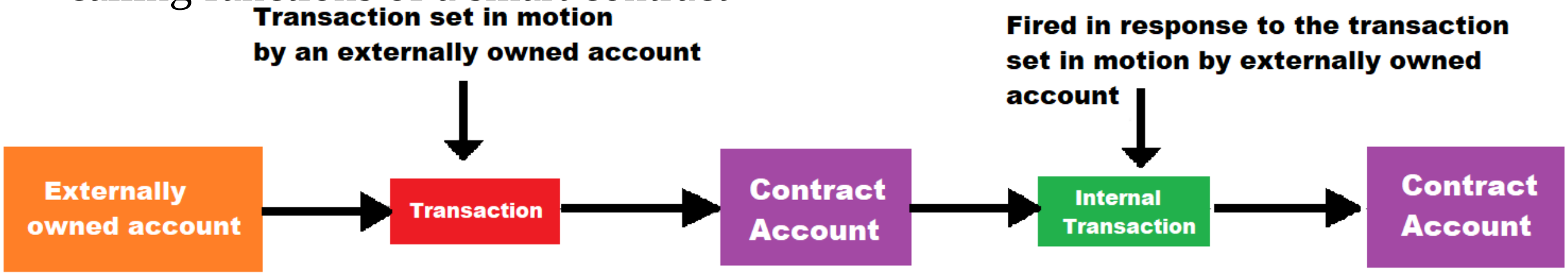
2.3. TRANSACTIONS

Blockchain transactions: perform any action:

- transferring value

➡ **Transaction is the way the external world interacting with the Ethereum network.**

- calling functions of a smart contract



2.3. TRANSACTIONS

Types of Transaction

The transactions can be classified into three general types.

- Funds Transfer
- Contract Deployment
- Function Execution

Structure of a Transaction

Transaction

- Recipient
- Nonce
- Gas Price
- Gas limit
- Value
- Data
- Signature

2.3. TRANSACTIONS

Funds Transfer

This transaction has a value associated with it but does not have any data.

Contract Deployment

➡ Whenever a contract is deployed on the ethereum network, this transaction takes place.

Smart Contract → Compile → Bytecode and ABI → Deploy → Contract Address

➡ A block is actually the building block or the key element of a blockchain.

2.4. BLOCKS

➡ Blocks contain transactions, each block contains a different number of transactions.

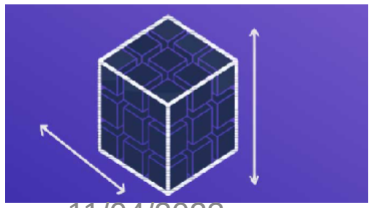
Block:

Block

- **Data:** "Blockchain Data Structure"
- **Previous Hash:** 0234ABED4
- **Hash:** 0234ABED4

Block Identification

Block Height



Block Hash

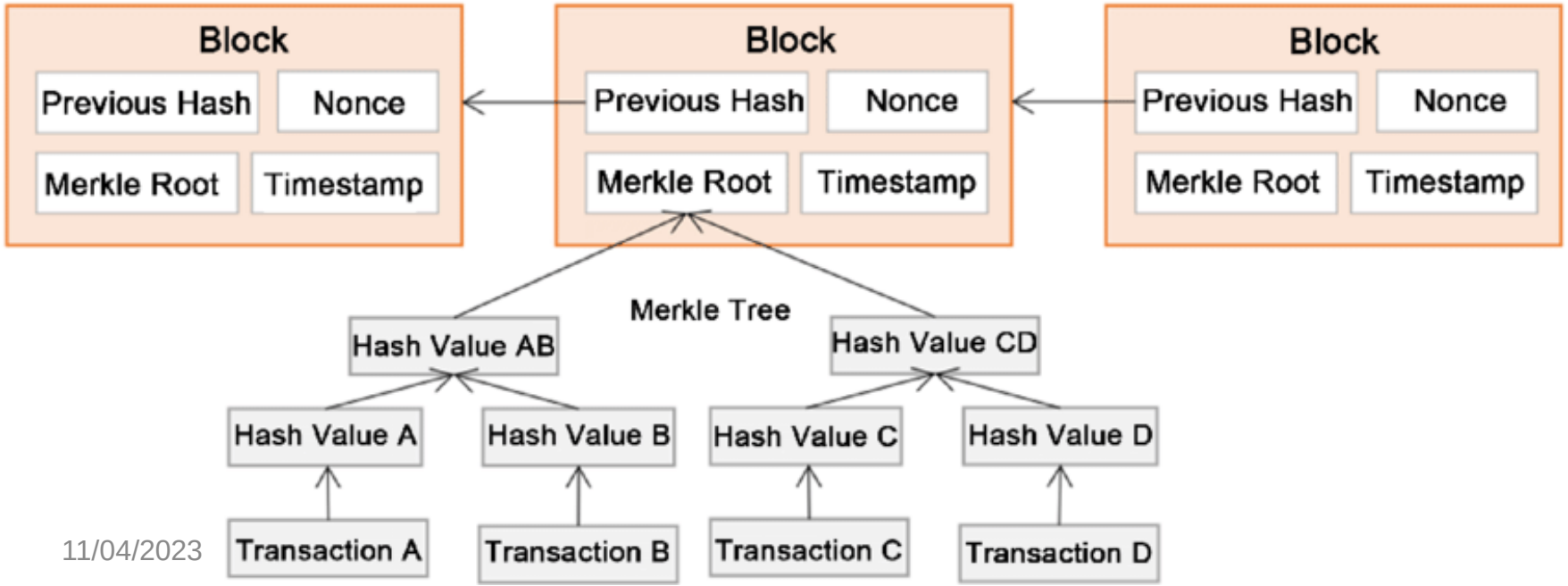
Hash
4f90feb5c78gbf7a5dge667cafgfc4fgb6fce74agf53735236d2gf7f03dc687a

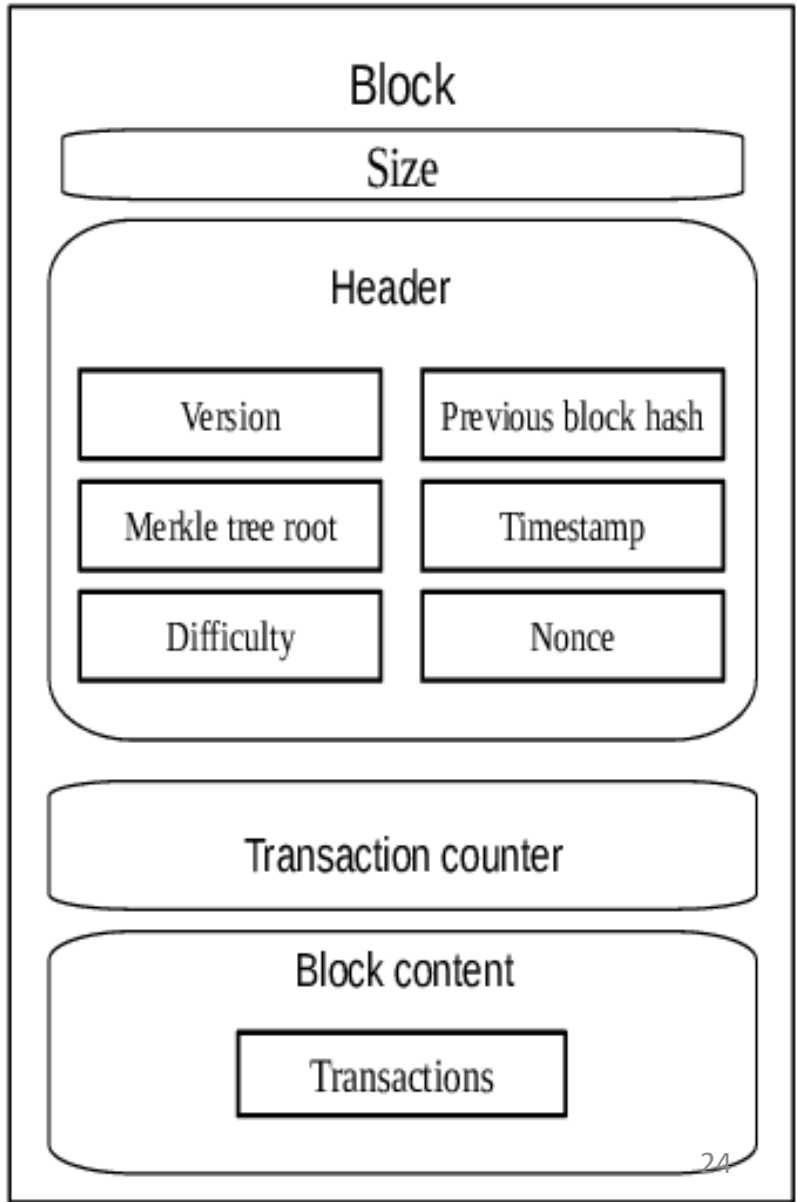
Properties of a Block

- Miner Property
- Total Difficulty Property
- gasLimit Property
- Difficulty Property
- Nonce Property
- gasUsed Property
- Hash Property
- Transactions Property
- Number Property

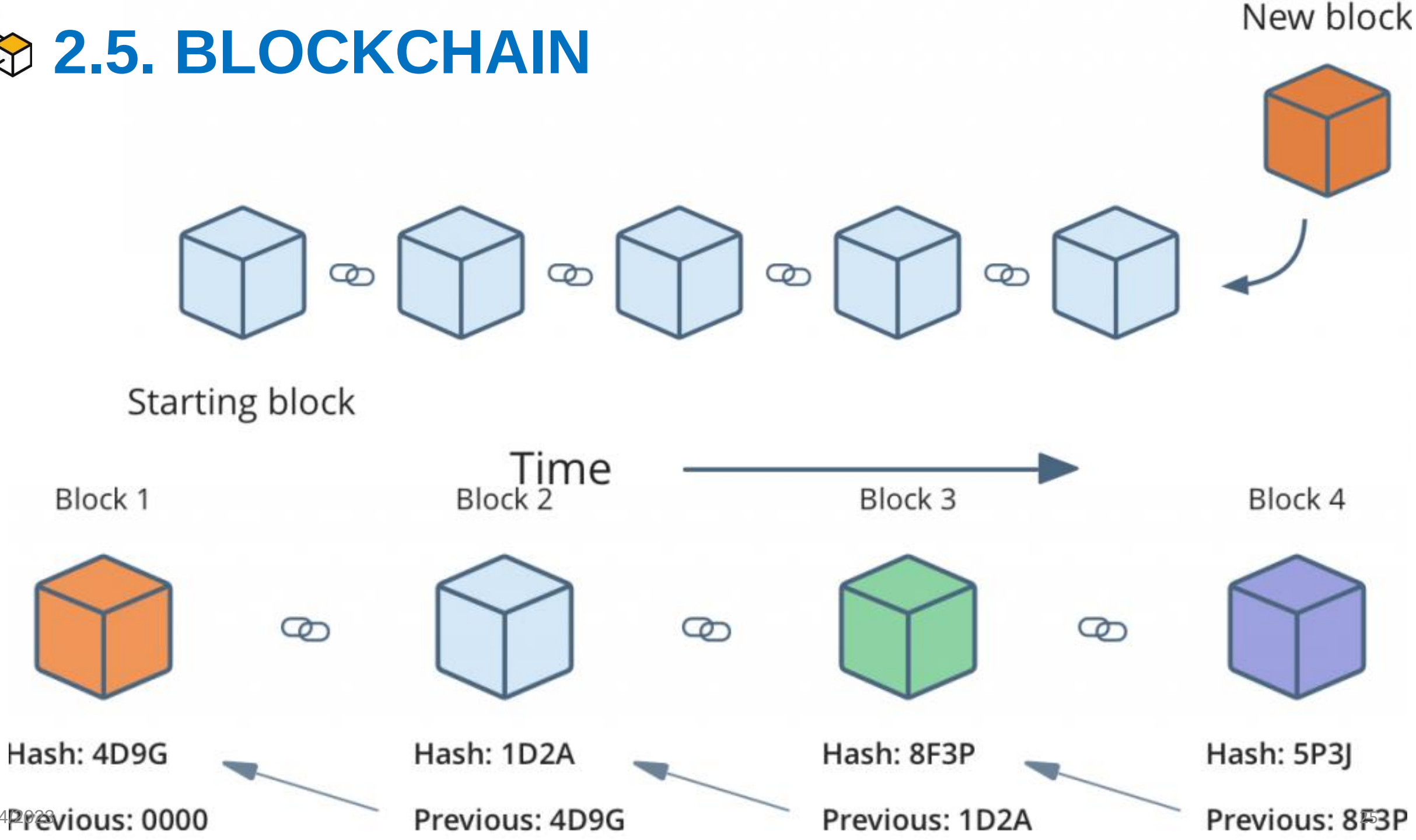
2.4. BLOCKS

Blockc data structures

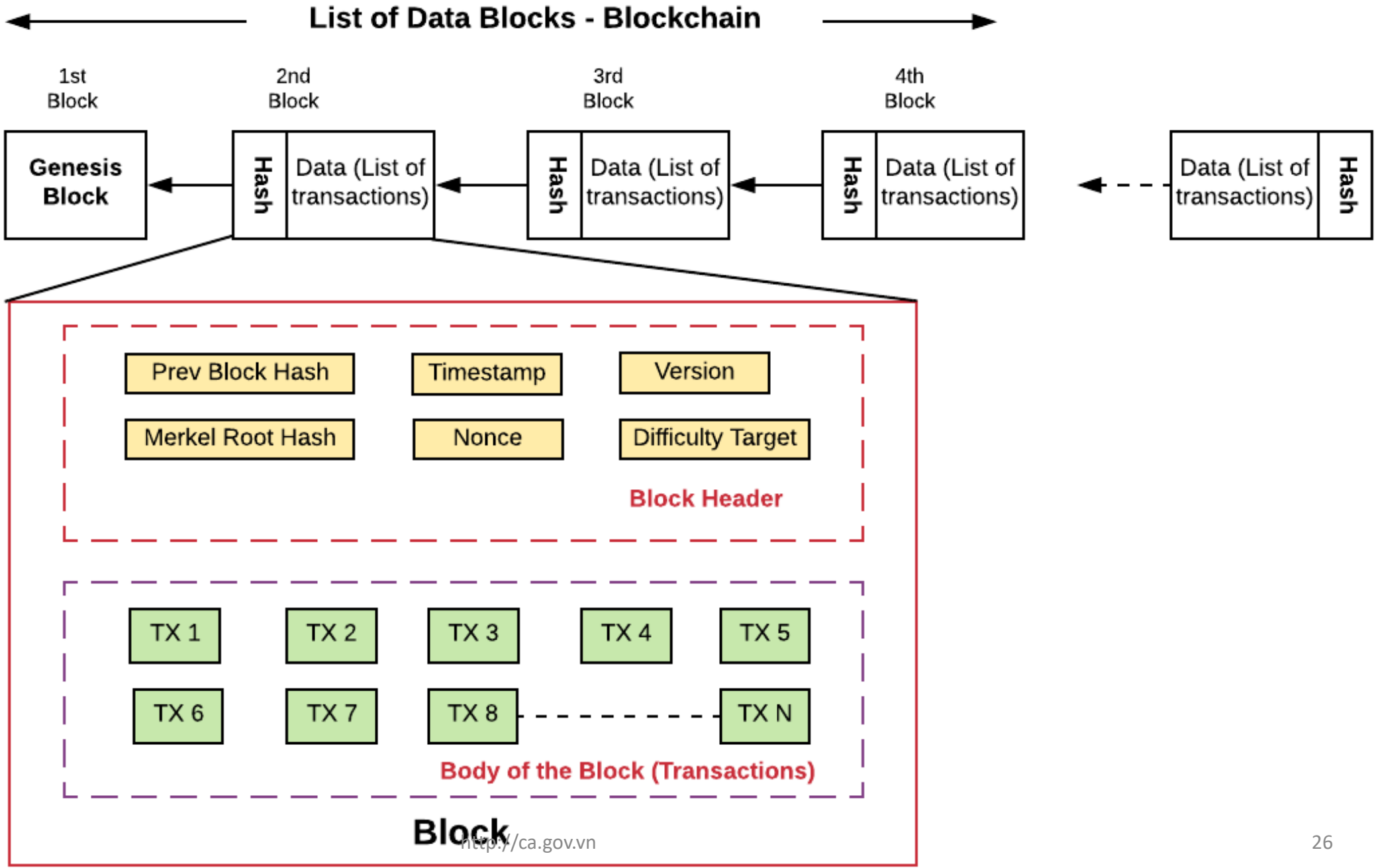




2.5. BLOCKCHAIN

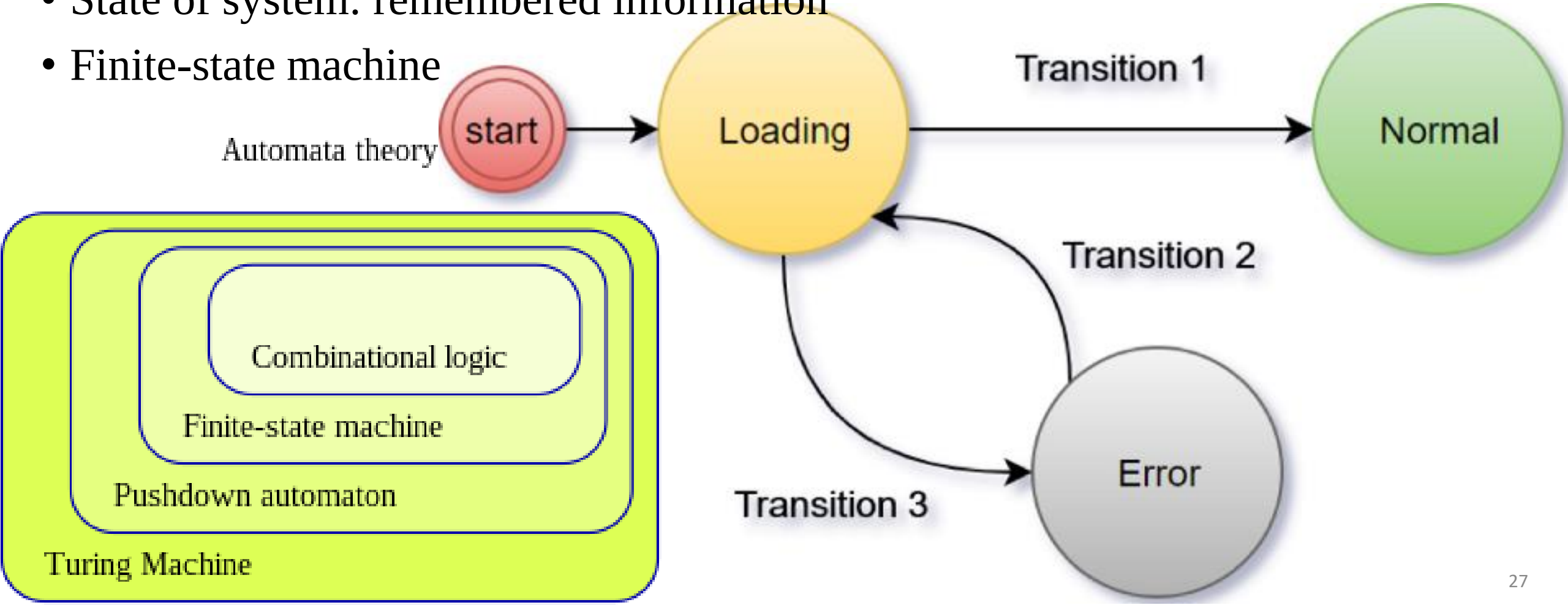


2.5. BLOCKCHAIN

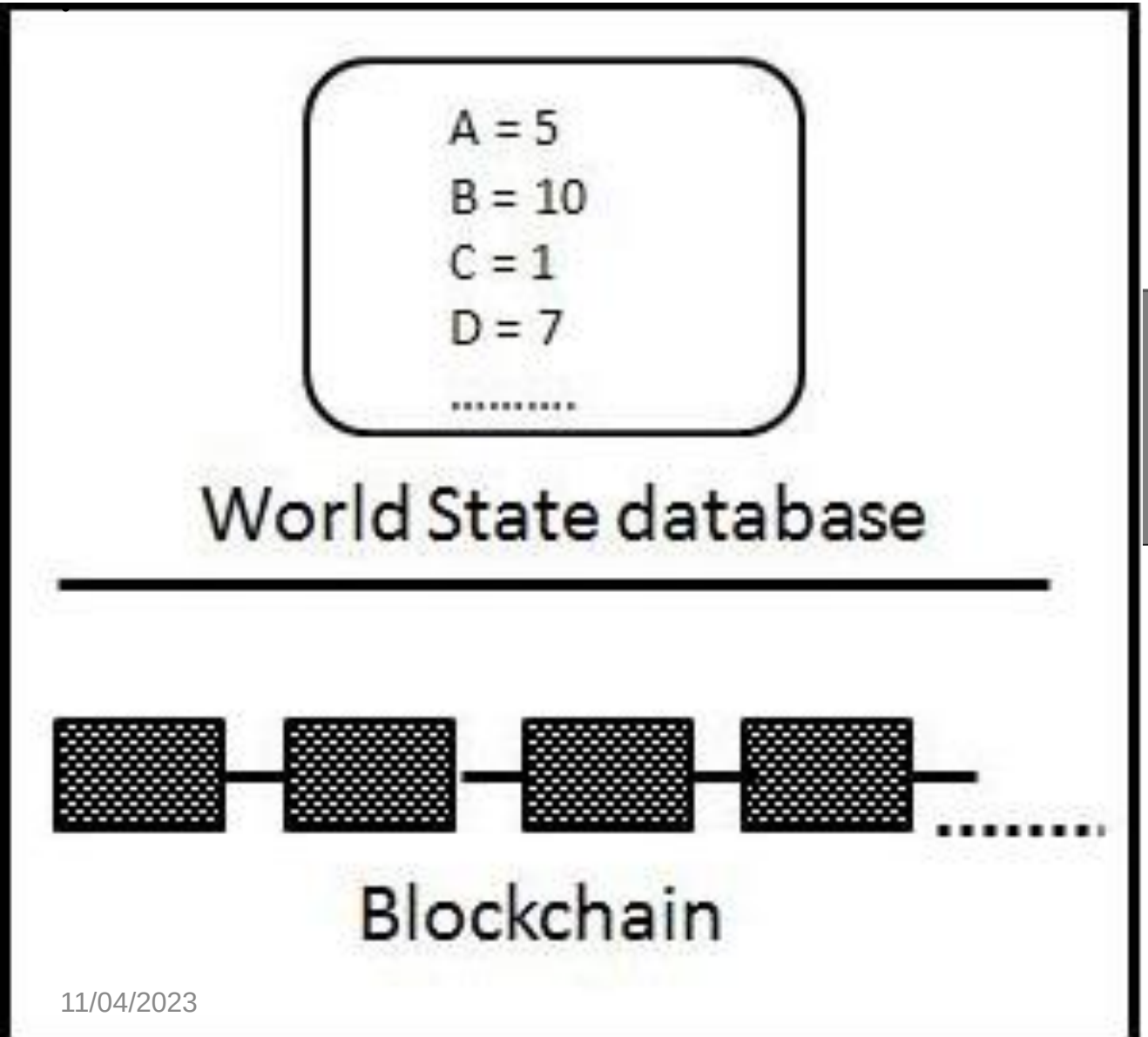


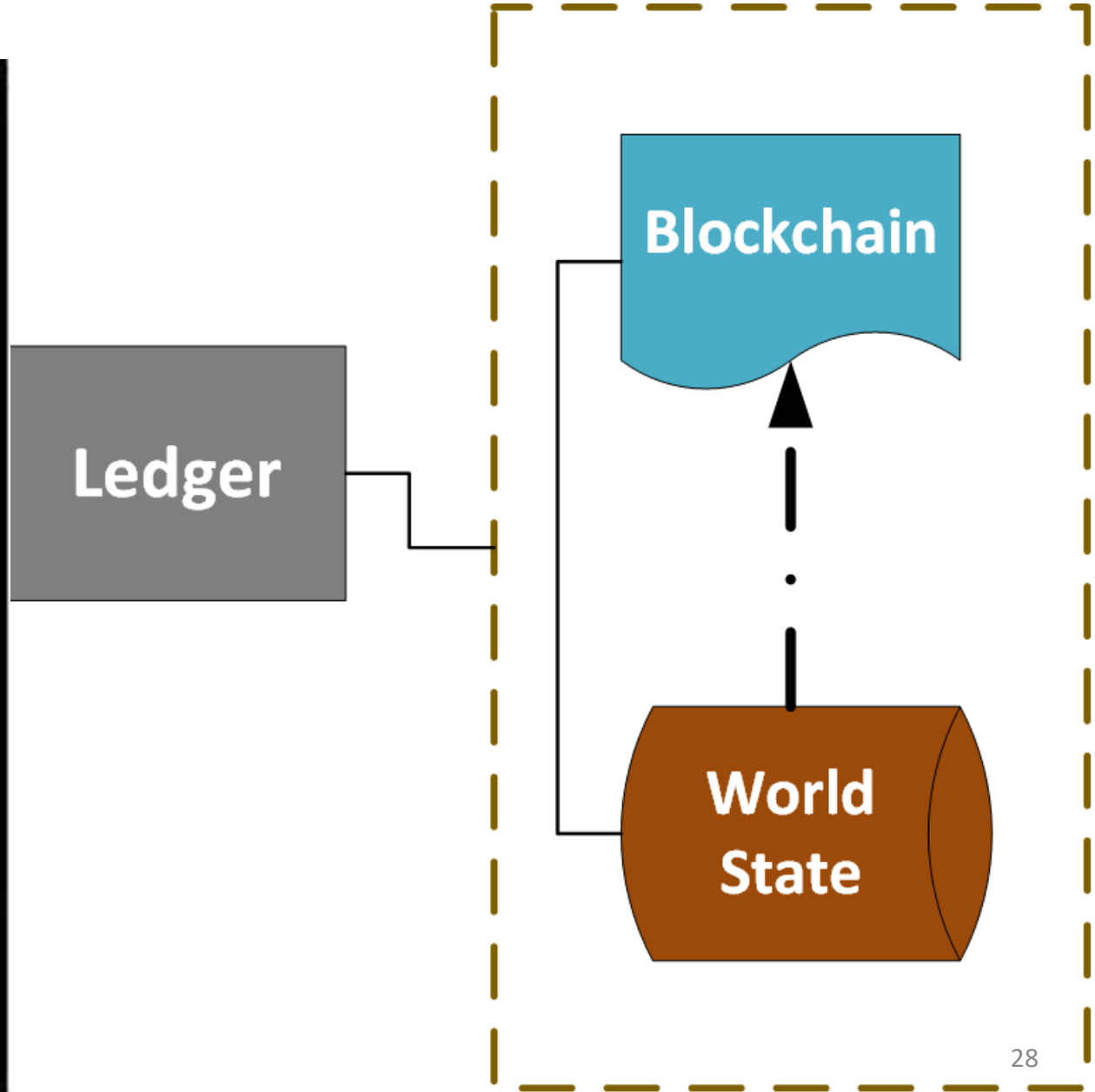
2.6. BLOCKCHAIN STATES

- State: particular condition that someone/something is in at a specific time.
- Stateful system: able remember preceding events/user interactions
- State of system: remembered information
- Finite-state machine

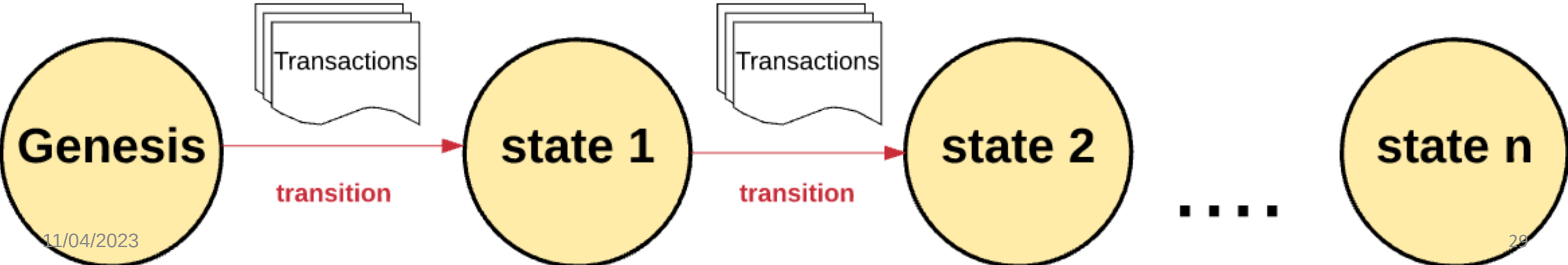


2.6. BLOCKCHAIN STATES

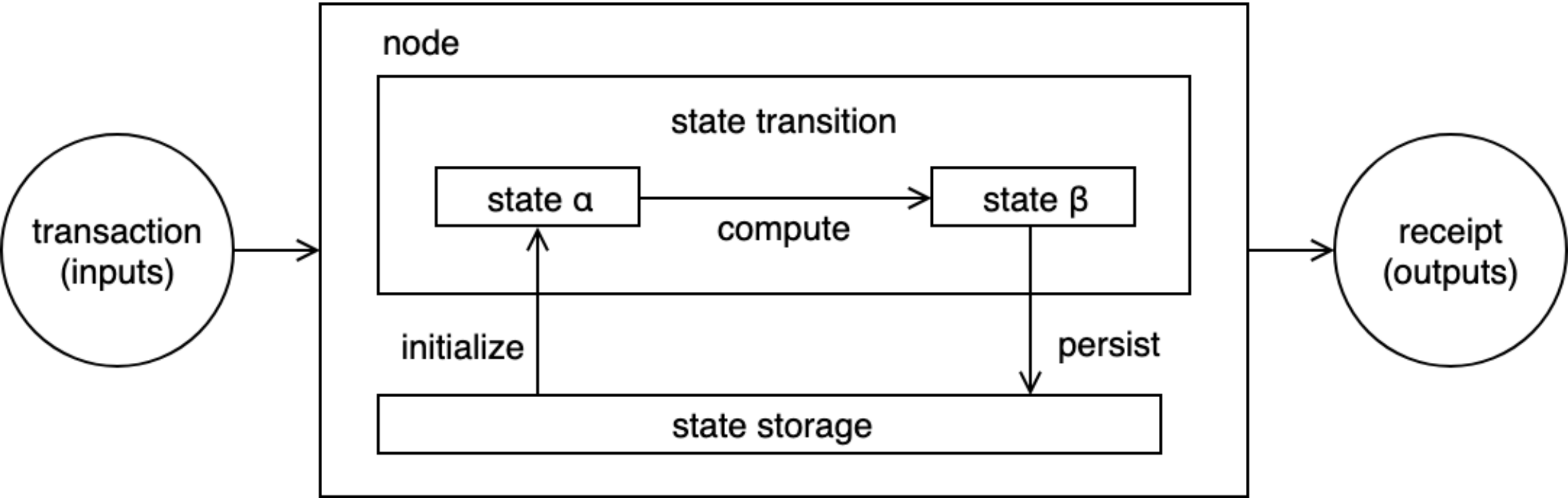




2.6. BLOCKCHAIN STATES



2.6. BLOCKCHAIN STATES



2.7. TRANSACTIONS ACCOUNTING MODEL

Triple entry accounting

How are records stored?

Conventional Accounting

v.s.

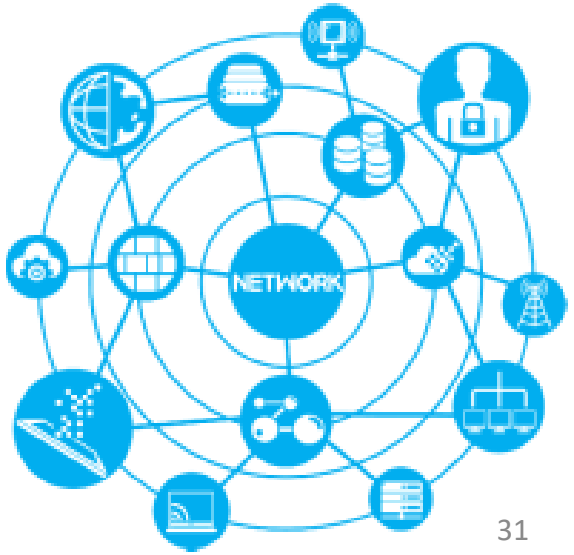
Blockchain Accounting

↓

Centralized ledger

↓

Distributed ledger technology



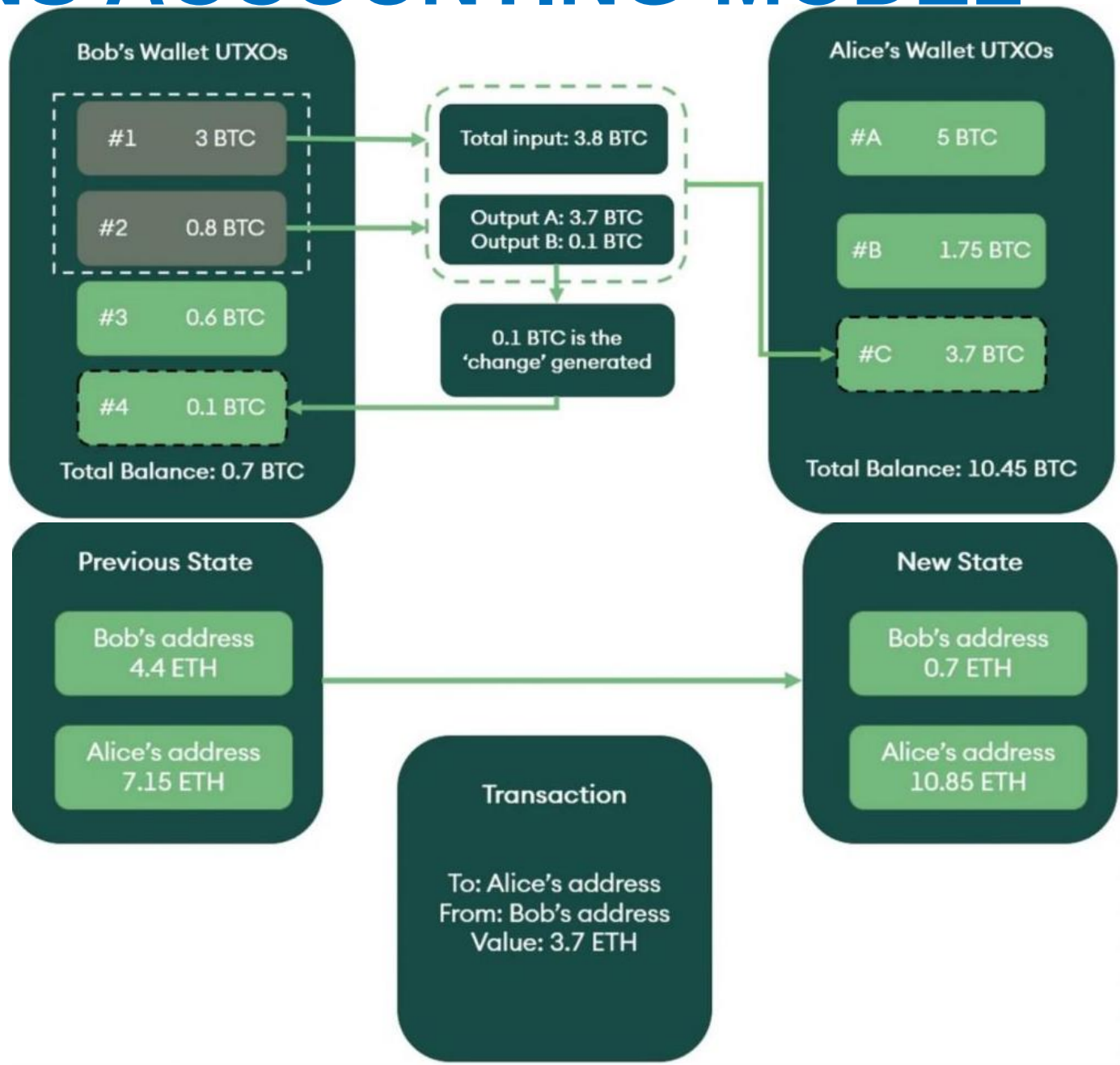
2.7. TRANSACTIONS ACCOUNTING MODEL

Unspent Transaction Output (UTXO) model:

- Analogy with physical fiat currency
- Represents a global blockchain state through all spent and unspent transaction outputs.
- based entirely on individual transactions, grouped in blocks

Account model:

- represents assets as balances within accounts, similar to bank accounts
- state of system is updated based on previous transactions.

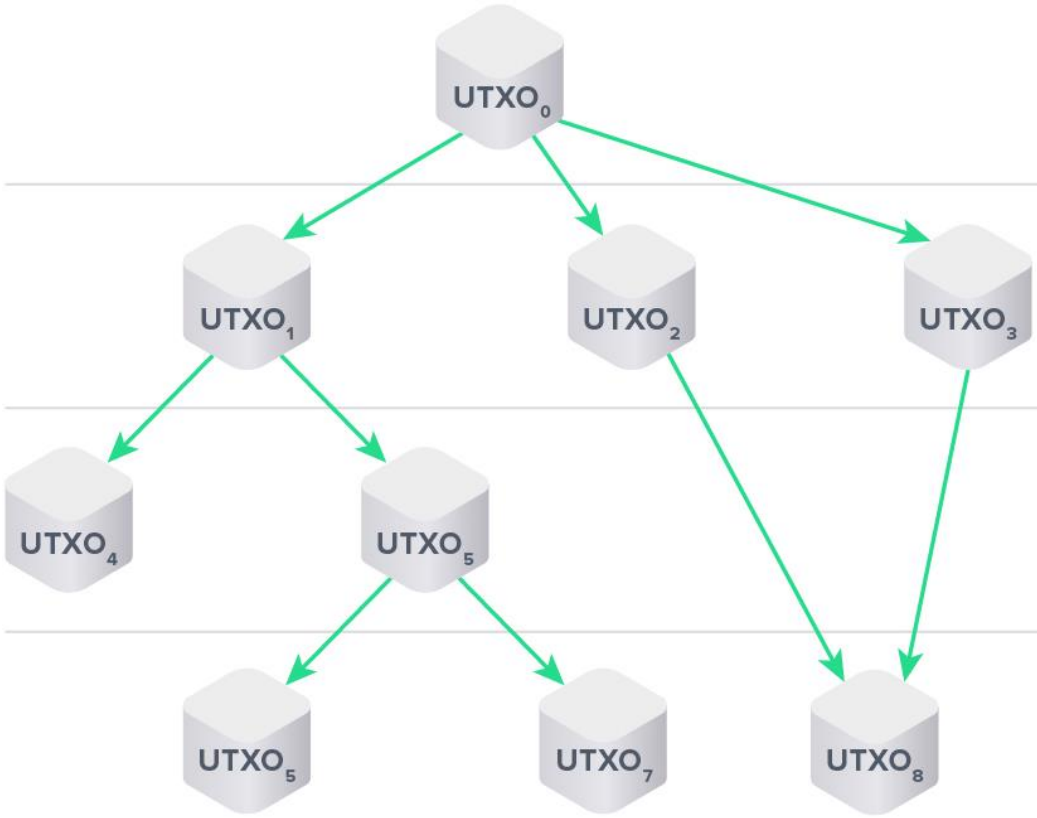


2.7. TRANSACTIONS ACCOUNTING MODEL

RECORDING THE STATE OF THE SYSTEM

•

UTXO Model



Directed graph of assets (UTXOs) moving between users

	Account Model	
State n	Account A	*Balance* t_0
State (n+1)	Account A	*Balance* t_1
	Account B	*Balance* t_1
State (n+2)	Account A	*Balance* t_2
	Account B	*Balance* t_2
State (n+3)	Account A	*Balance* t_3
	Account B	*Balance* t_3
	Account C	*Balance* t_3

Database of network states

2.7. TRANSACTIONS ACCOUNTING MODEL

	UTXO Model	Account/Balance Model
Explicit balance		✓
Verification efficiency		✓
Smart rules support	✓	
Divisibility	Indivisible	Divisible

	State Type	User account	Scalability	Security	Decentralisation	Smart Contract
UTXO	Local and deterministic state.	Private-Key Wallet address	Outputs can be easily processed in parallel.	Public key transactions	Implement strong Nakamoto-Style consensus	Not expressive. E.g. Bitcoin Script.
Account	Global shared state.	Account with balance & state.	Layer 2 off-chain solutions.	Public key transaction/ Merkle trees	Implement strong Nakamoto-Style consensus	Very expressive. E.g. Solidity.
eUTXO	Local and deterministic state. Maintain contract state.	Private-Key Wallet address	Outputs can be easily processed in parallel.	Public key transactions	Implement strong Nakamoto-Style consensus	More expressive than UTXO model.

3. BLOCKCHAIN DATABASE

3.1. INTRODUCTION

3.2. DATABASE

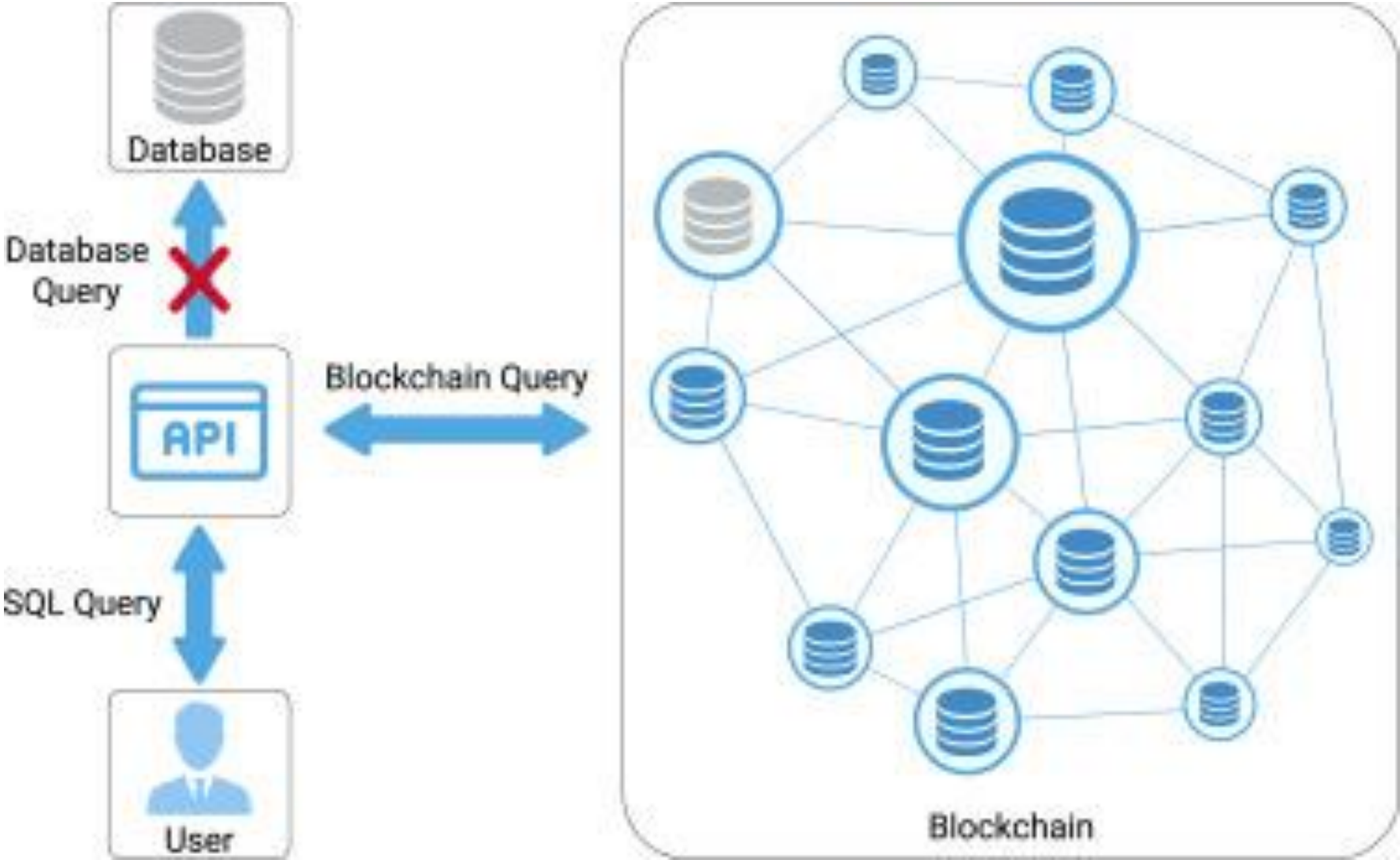
3.3. DATABASE MANAGEMENT SYSTEM

3.4. BLOCKCHAIN DATABASE

3.5. BLOCKCHAIN DATABASE MANAGEMENT SYSTEM

3.6. BLOCKCHAIN DB VS RELATIONAL DB

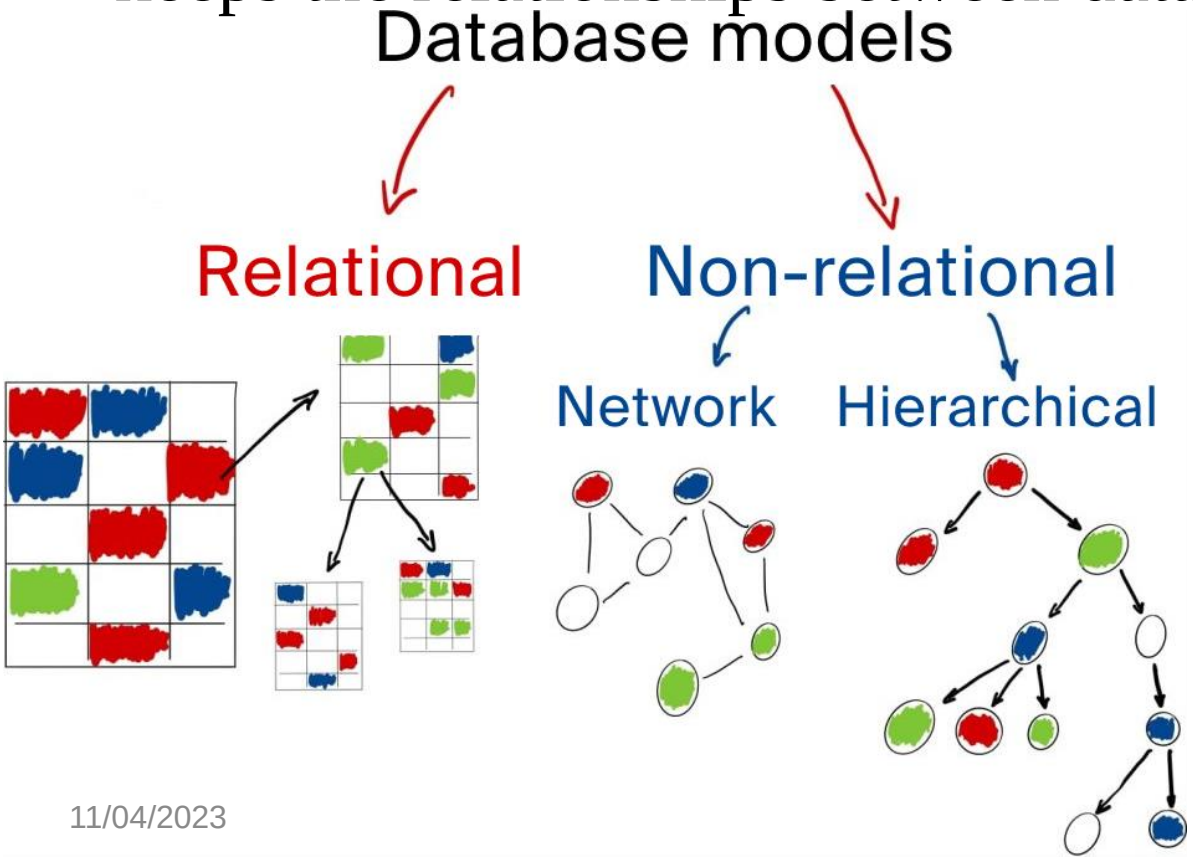
3.1. INTRODUCTION

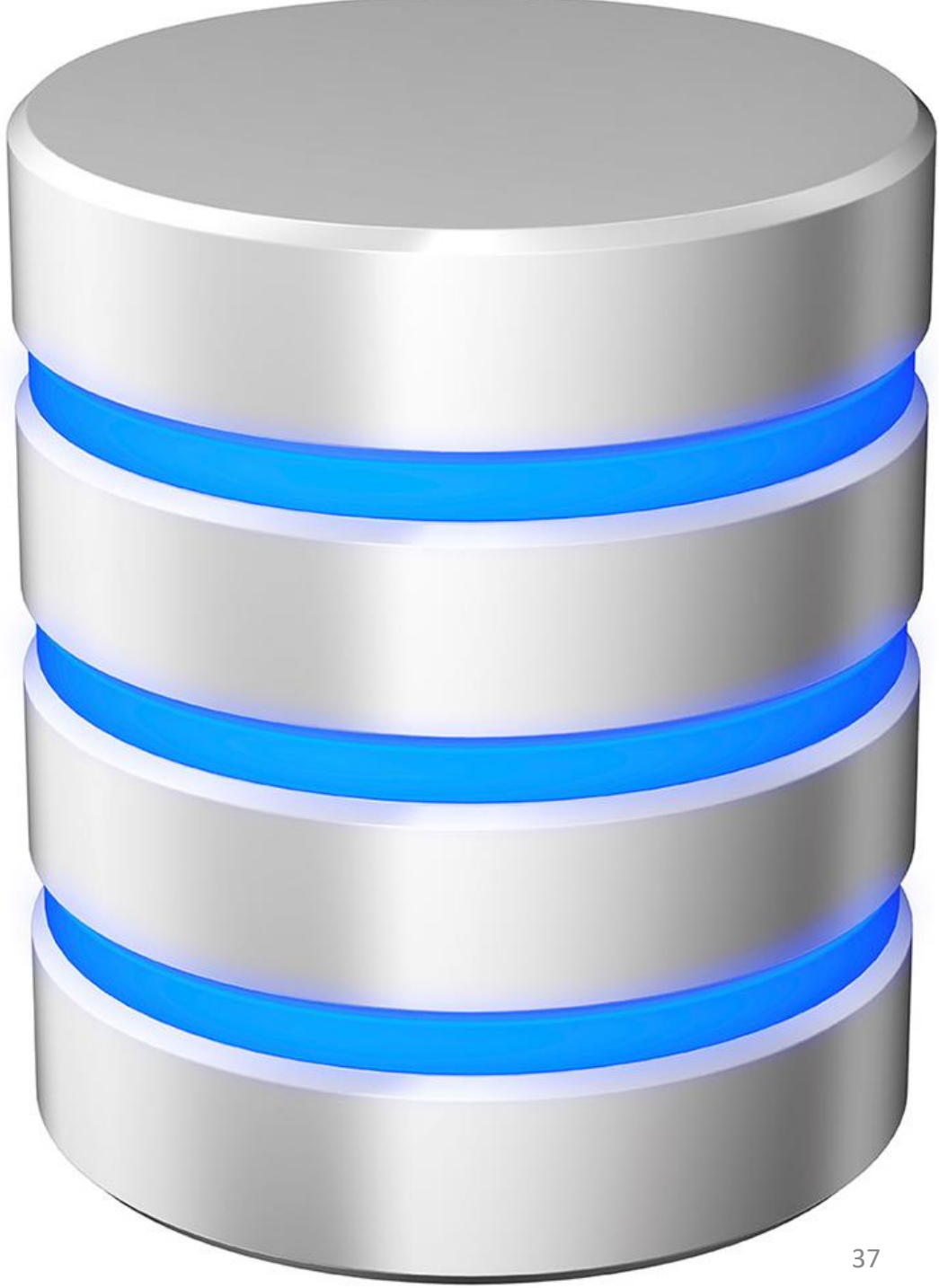


3.2. DATABASE

Database

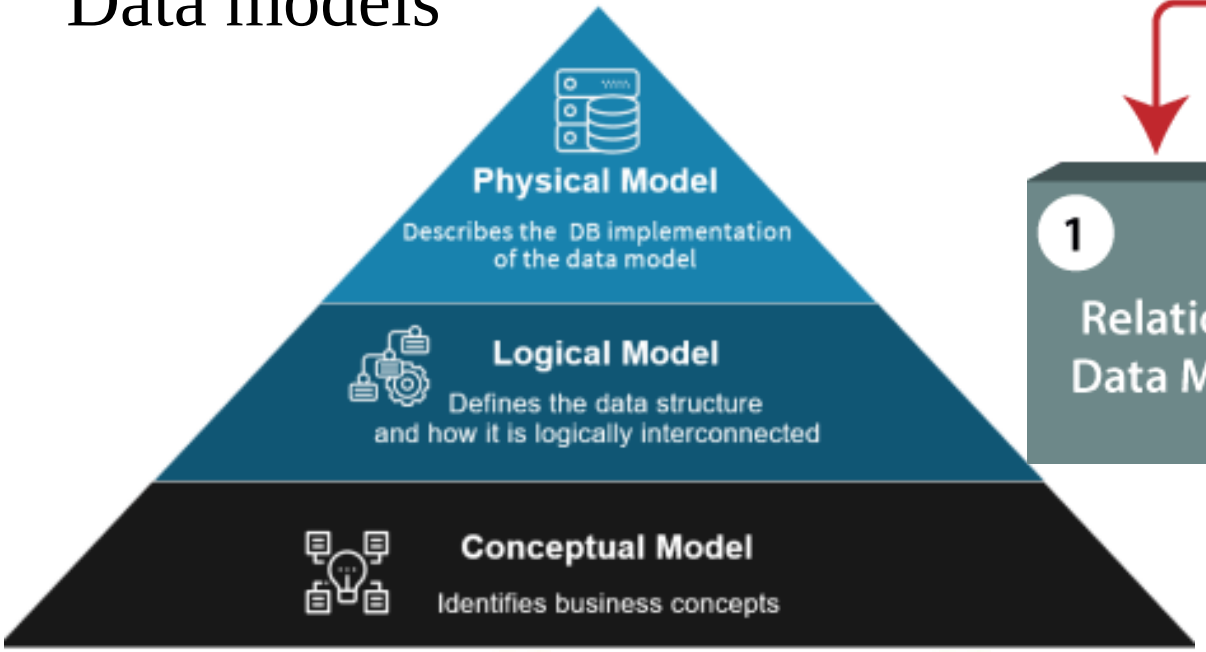
- organized collection of logically data
- stored and accessed electronically
- keeps the relationships between data points

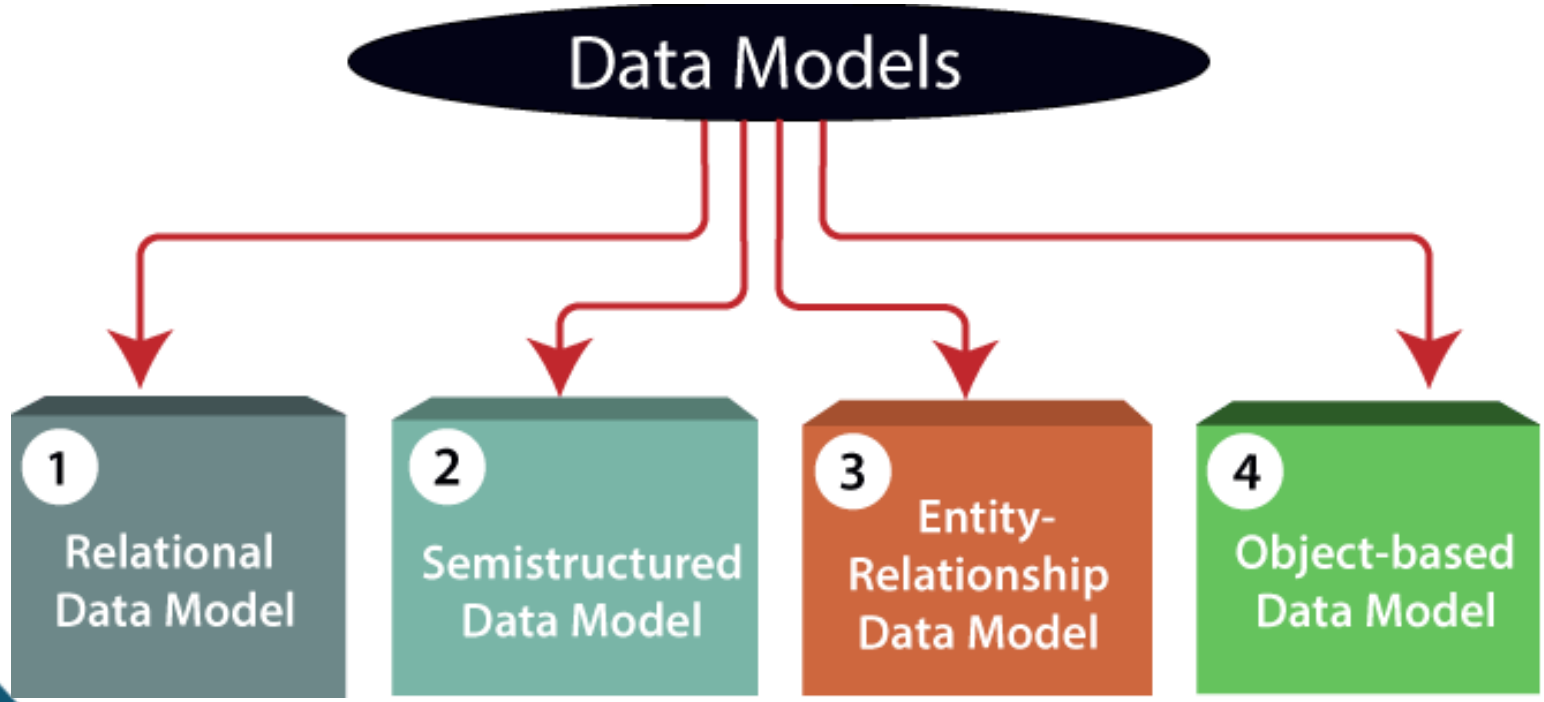


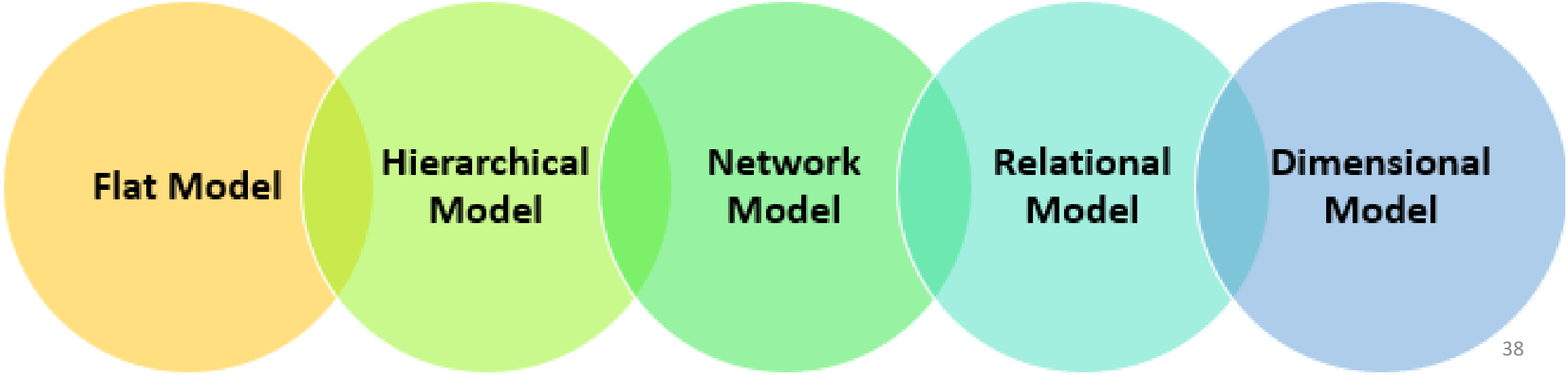


3.2. DATABASE

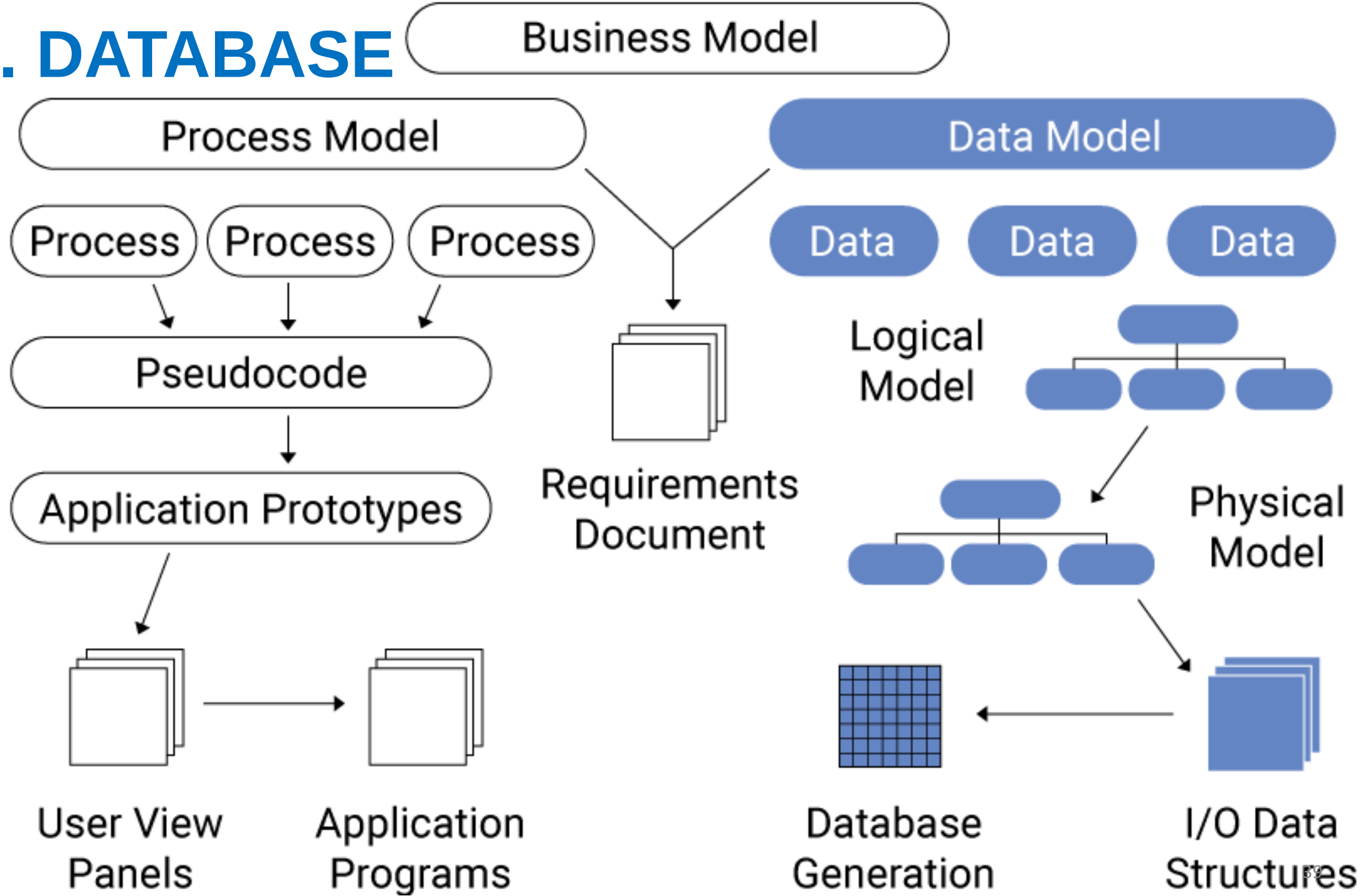
Data models







3.2. DATABASE

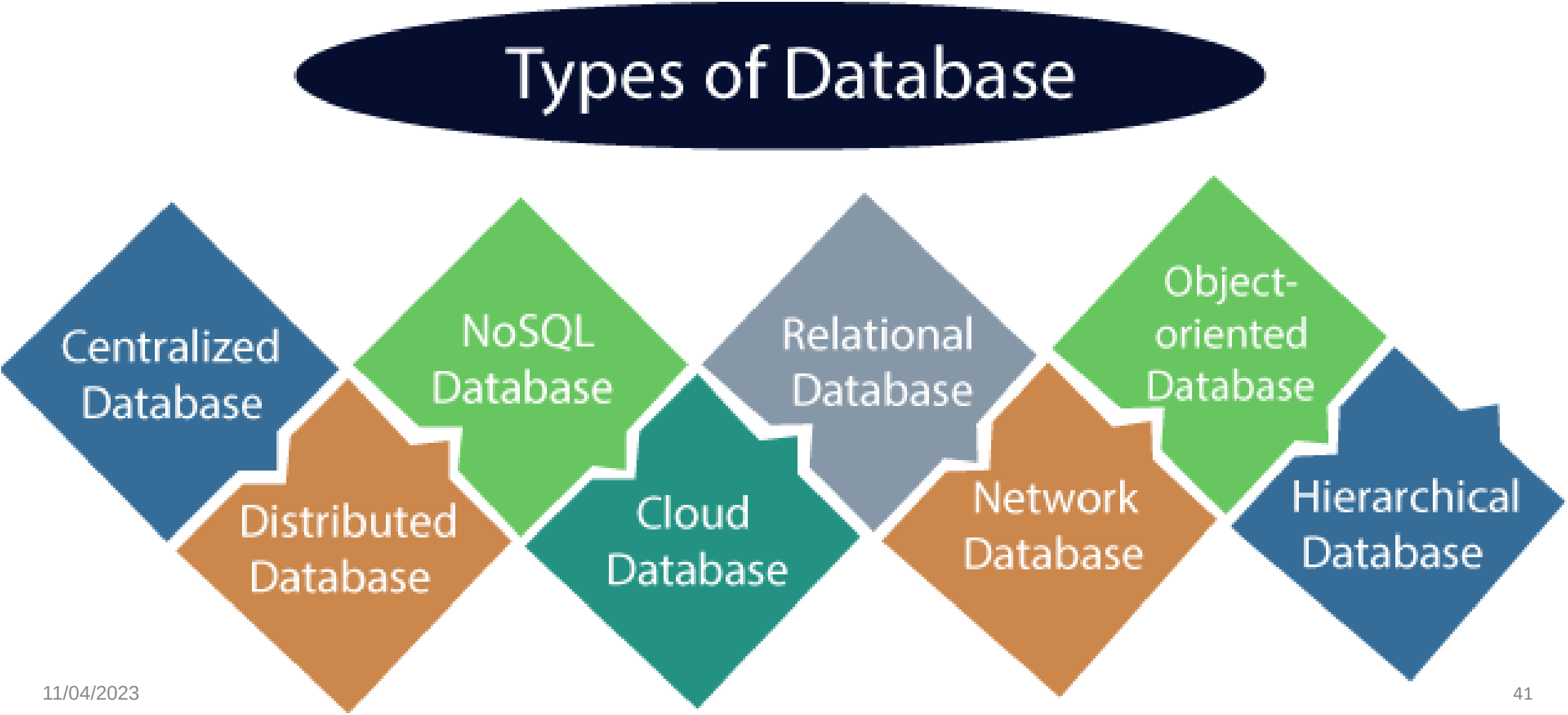

Business Model
Process Model
Data Model
Process
Process
Process
Data
Data
Data
Pseudocode
Logical Model
Requirements Document
Application Prototypes
Physical Model
User View Panels
Application Programs
Database Generation
I/O Data Structures

3.2. DATABASE

Components of a Database

3.2. DATABASE

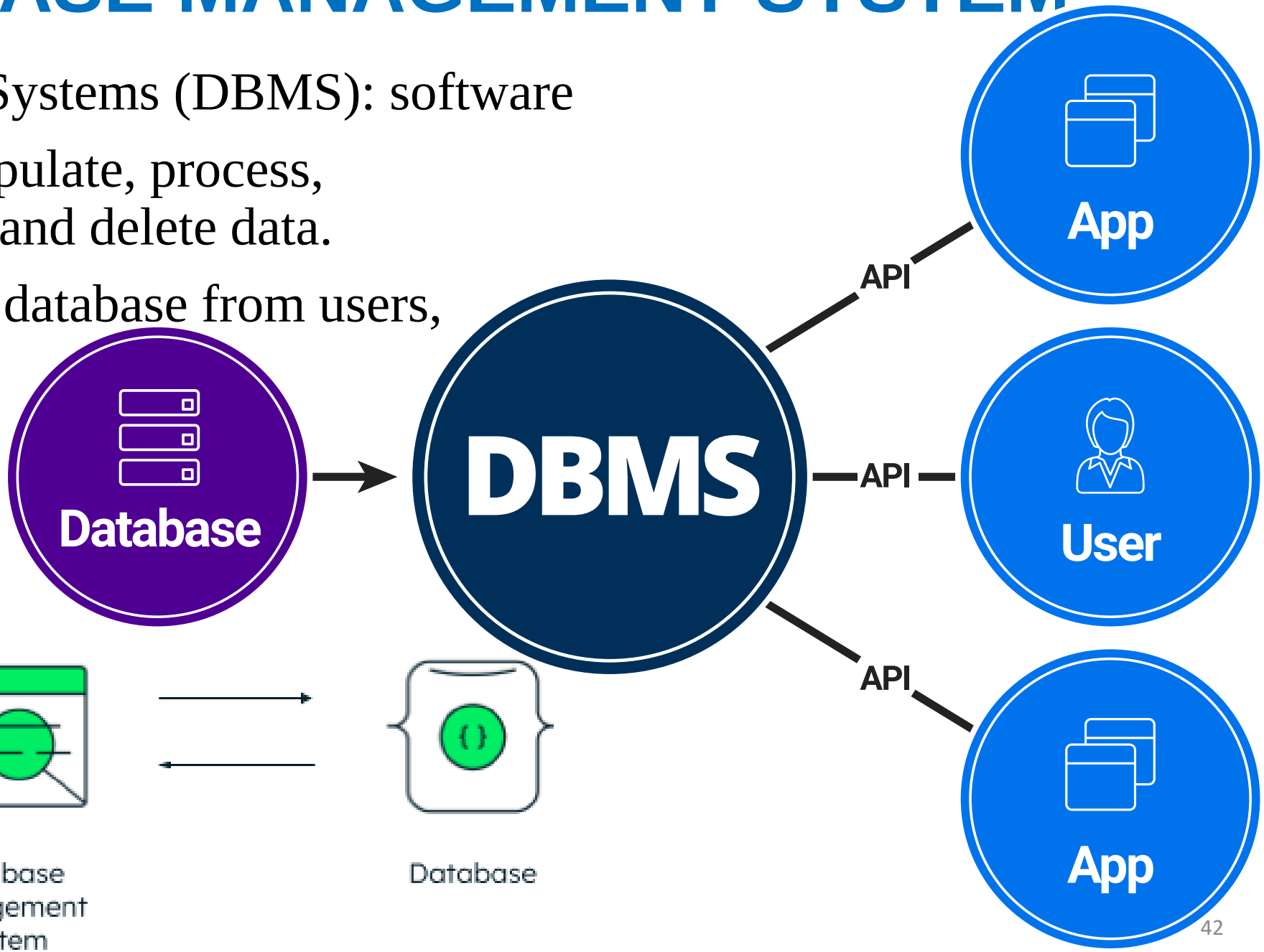
•

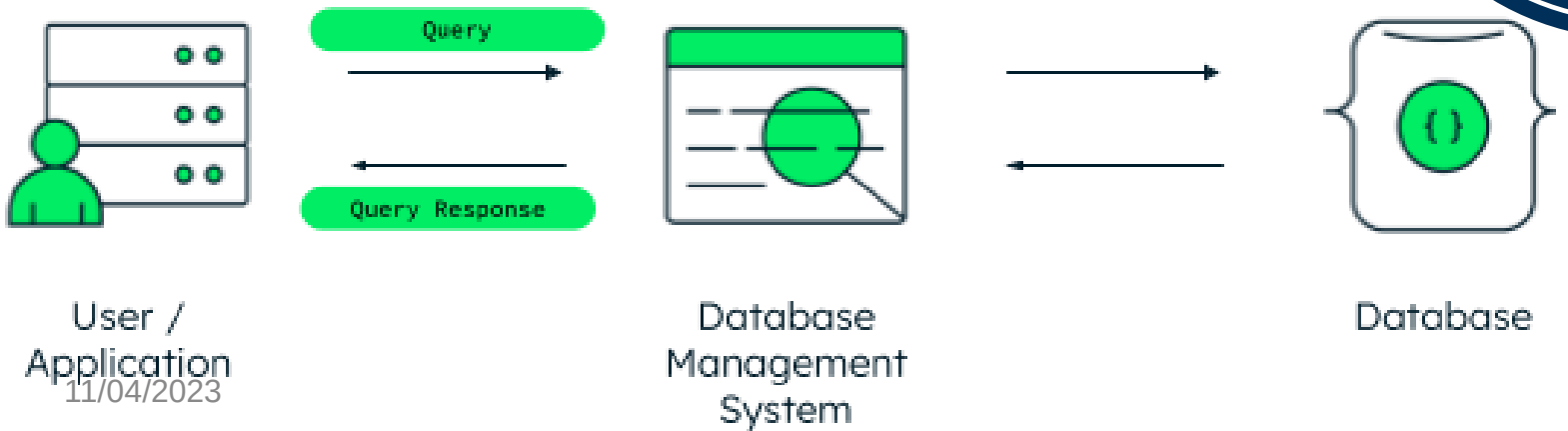

Types of Database
Centralized Database
Distributed Database
NoSQL Database
Cloud Database
Relational Database
Network Database
Object-oriented Database
Hierarchical Database

3.3. DATABASE MANAGEMENT SYSTEM

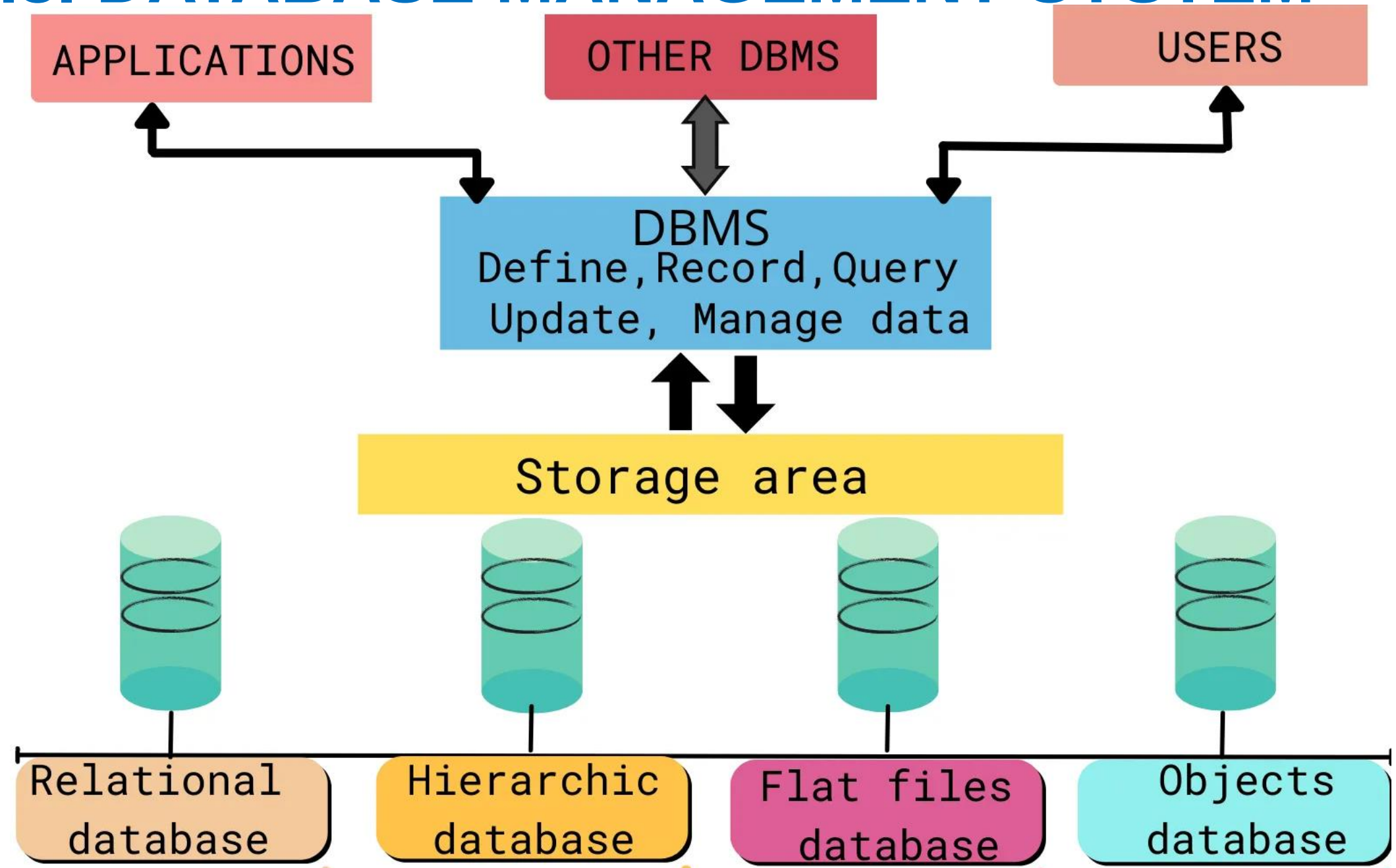
Database Management Systems (DBMS): software

- allows to access, manipulate, process, store, update, archive, and delete data.
- gatekeeper, separating database from users, apps that access data.





3.3. DATABASE MANAGEMENT SYSTEM



3.3. DATABASE MANAGEMENT SYSTEM

DBMS functions:

- Data definition: Creation, modification, removal of define organization of data.
- Update: Insertion, modification, deletion actual data.
- Retrieval: Providing data (information)
- Administration: users, security, performance, control, system failure

Data definitions

Data retrieval

Data manipulation

Access control

Data sharing

Data integrity

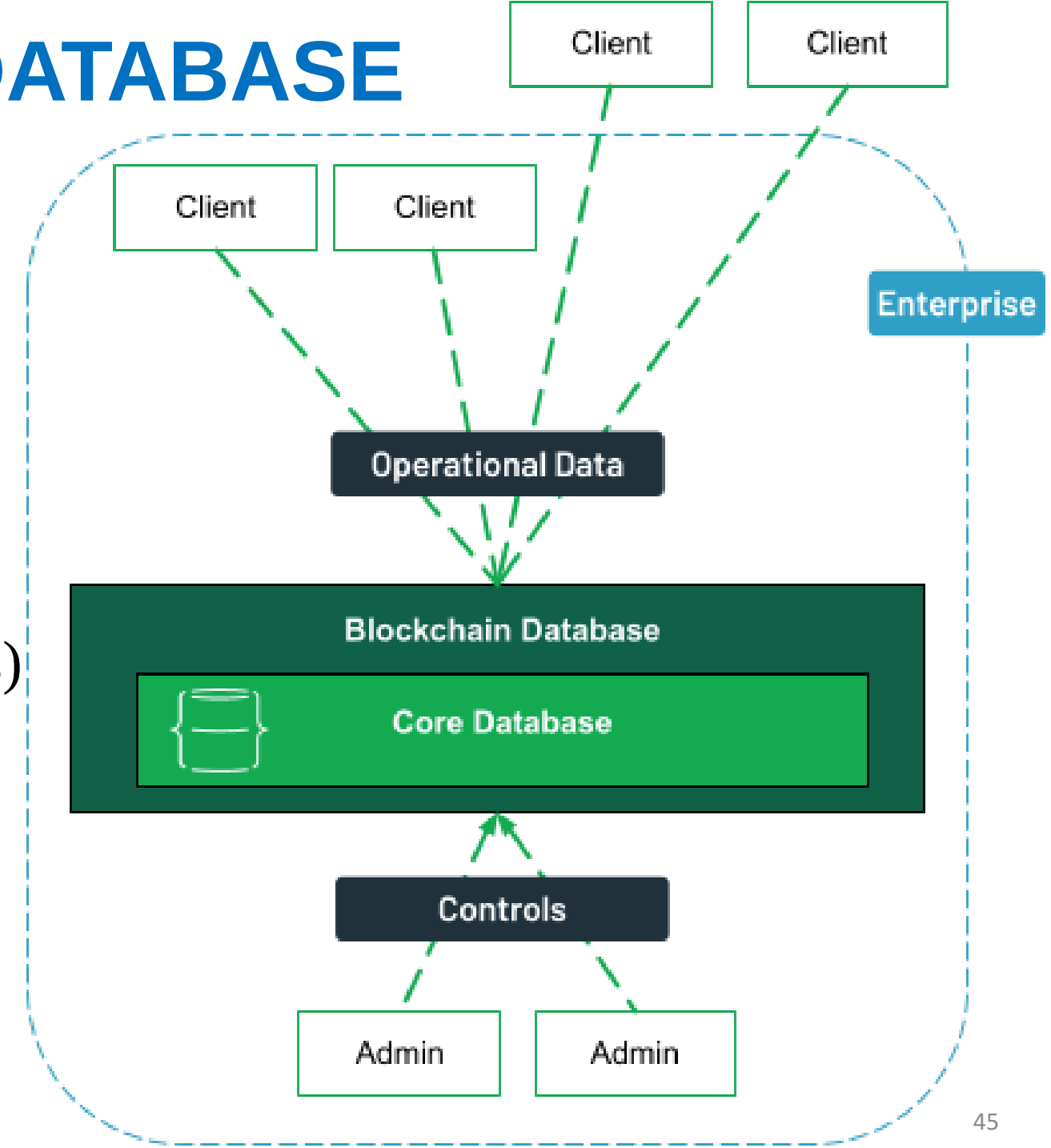
3.4. BLOCKCHAIN DATABASE

Blockchain databases:

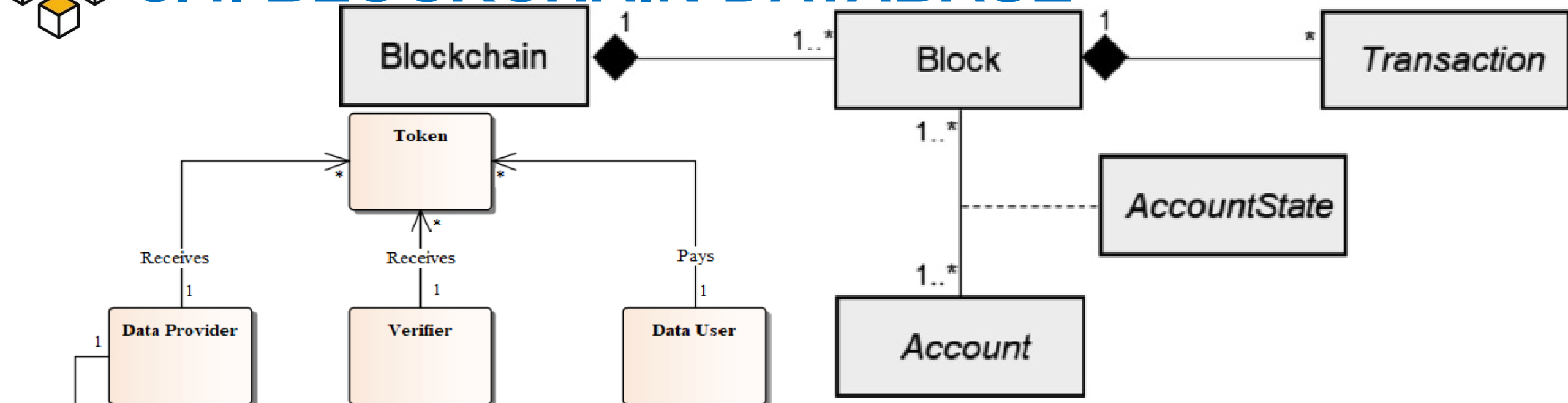
- Organize, store, manage blockchain data structure.

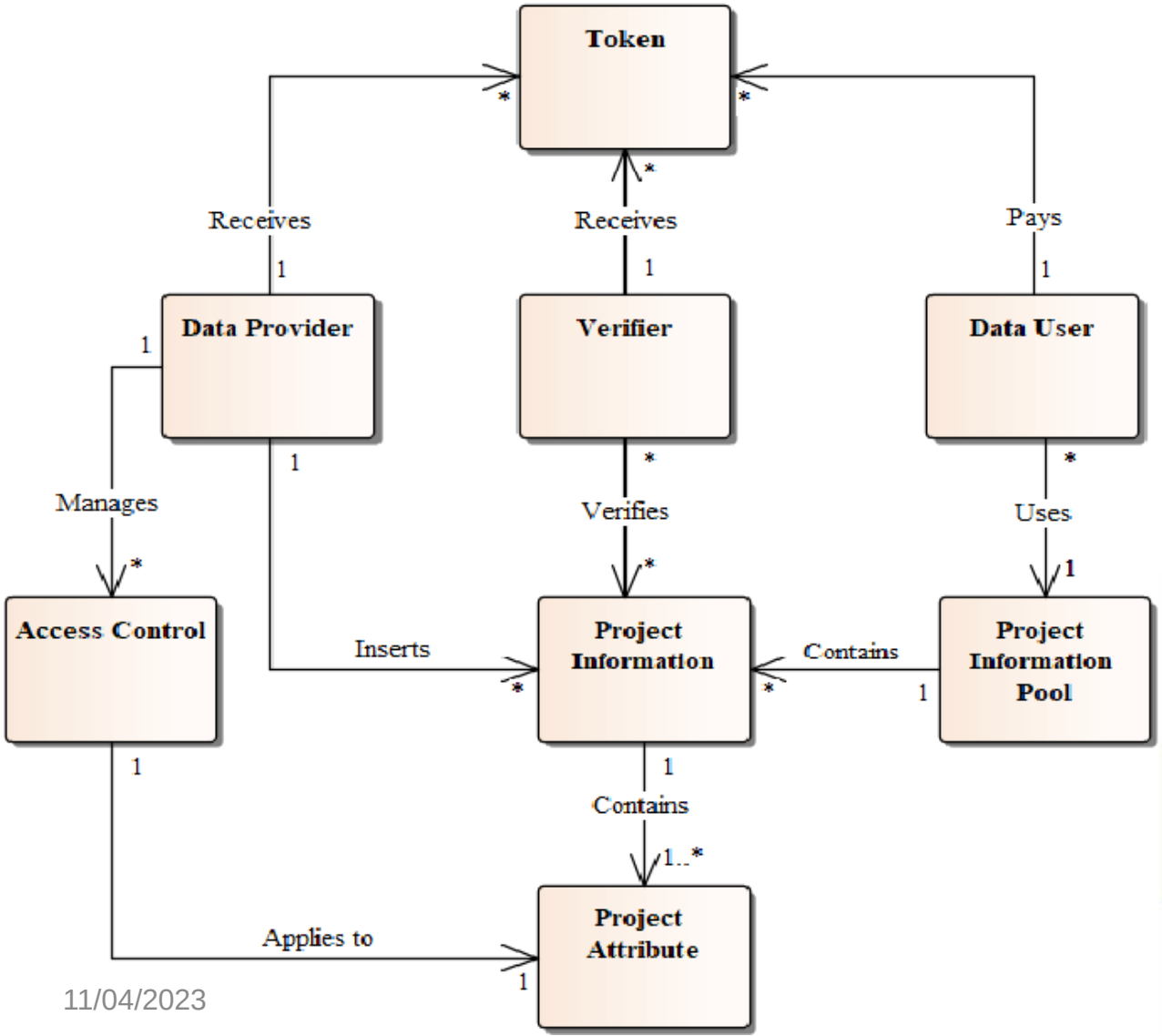
Blockchain data model:

- Data is stored as signed blocks
- Block link to each other (chain of immutable interconnected data entries)

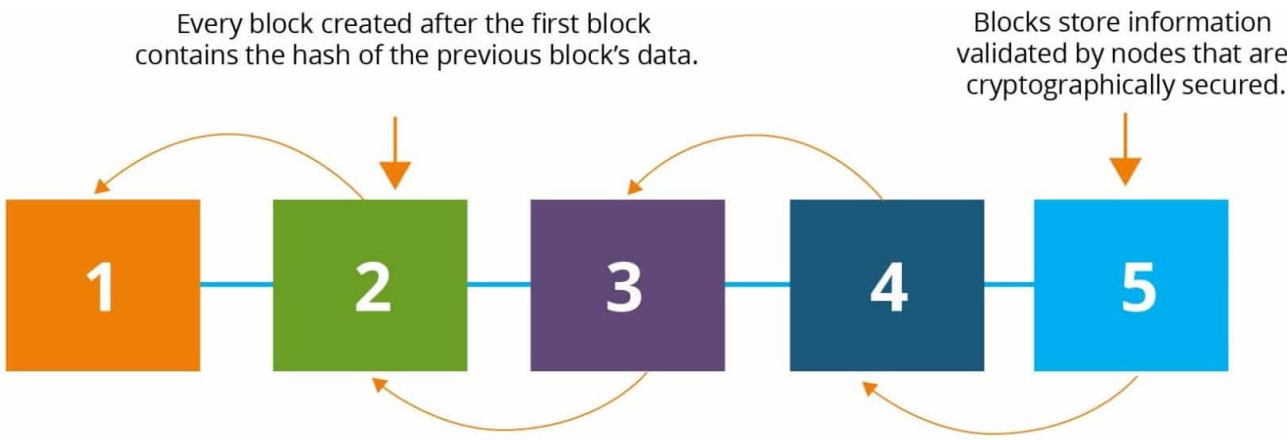


3.4. BLOCKCHAIN DATABASE





Data Modelling for Blockchain

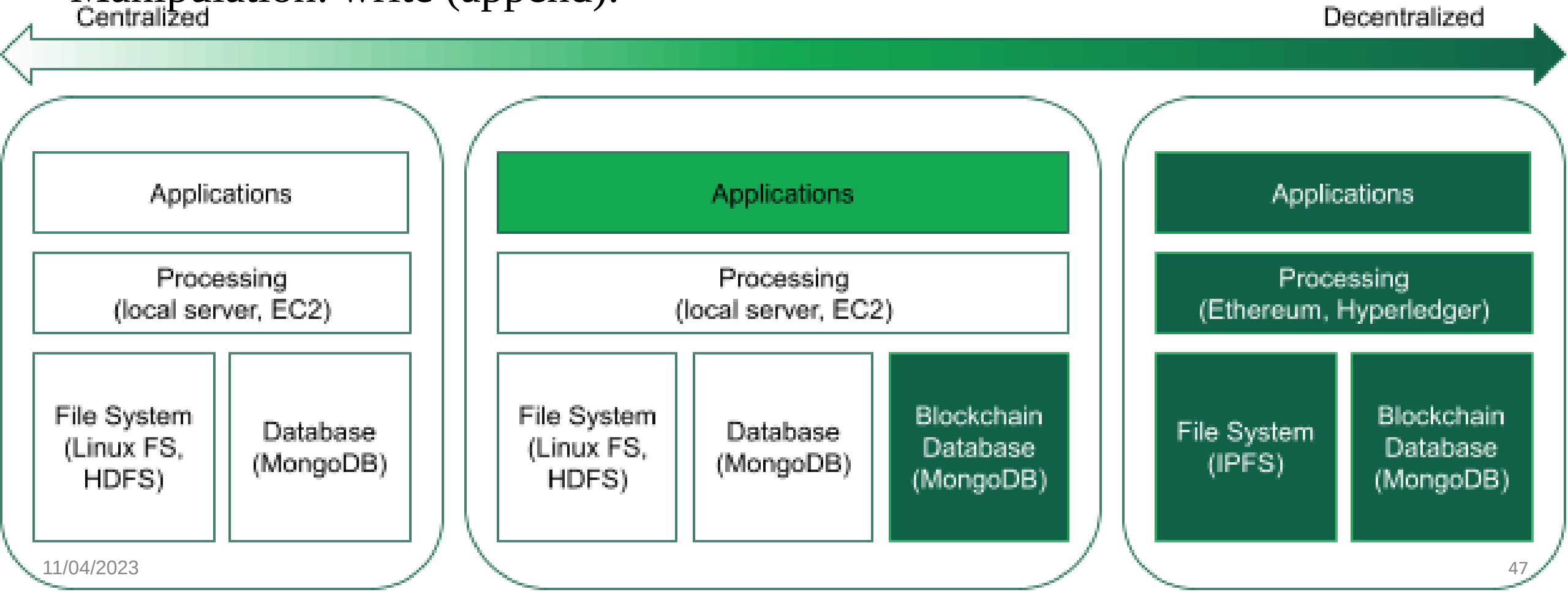


A blockchain is a linked list that consists of pointers.

3.5. BLOCKCHAIN DATABASE MANAGEMENT SYSTEM

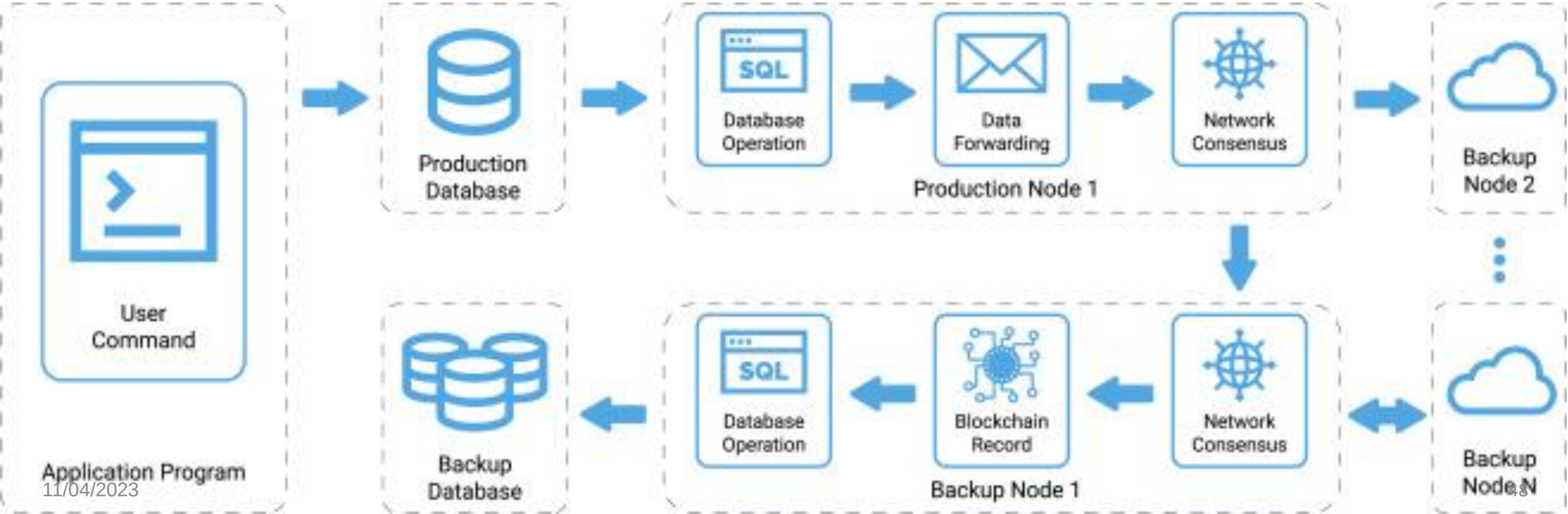
Functions:

- Retrival: read
- Manipulation: write (append).



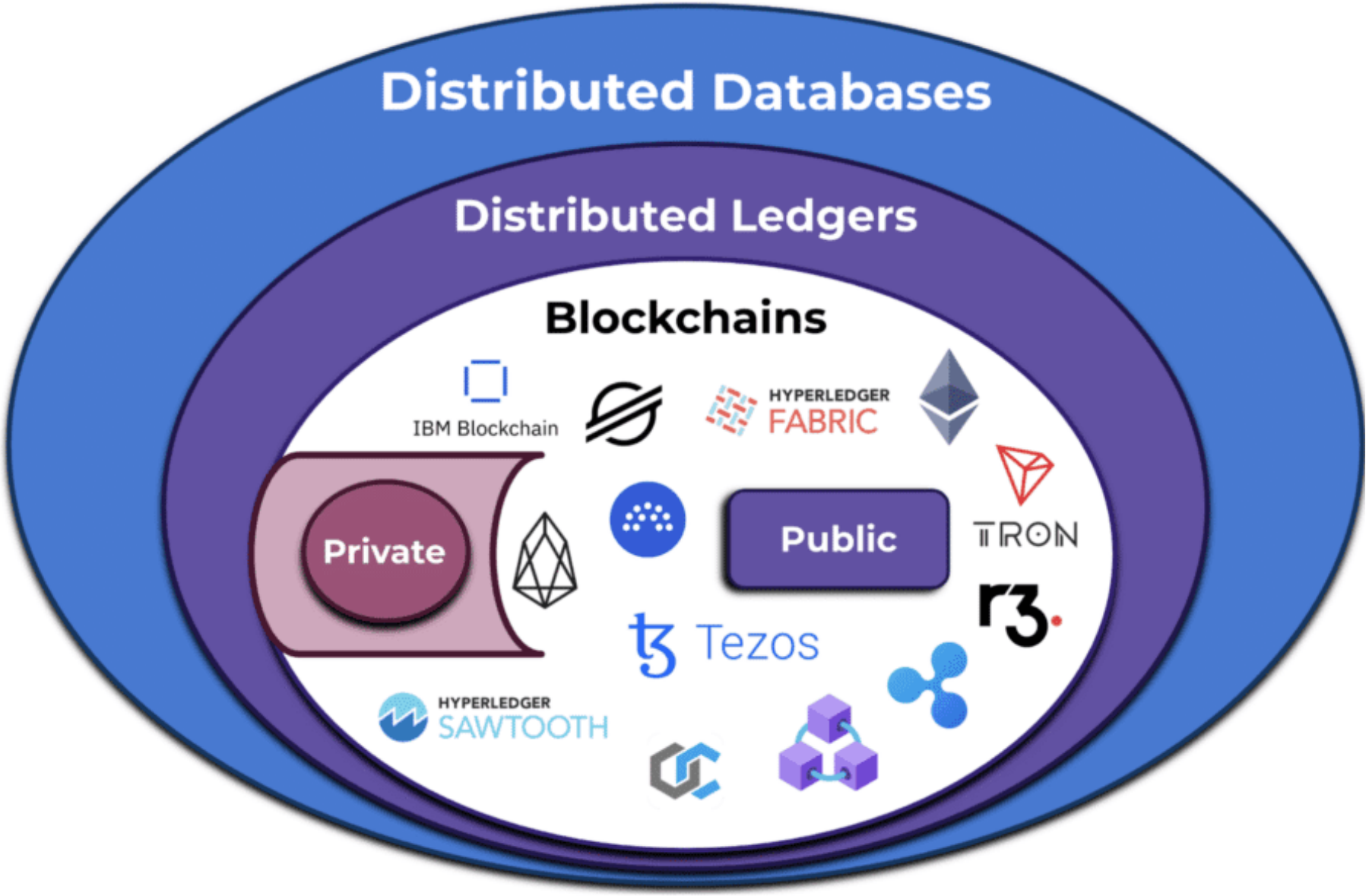
3.5. BLOCKCHAIN DATABASE MANAGEMENT SYSTEM

- Admin:
  - Access control: Address, Identity (Anonimous)
  - Sharing: Decentralization
  - Security: Intergrity, Hashing, Encryption



3.6. BLOCKCHAIN DB VS RELATIONAL DB

Blockchain's Relationship to Databases



3.6. BLOCKCHAIN DB VS RELATIONAL DB

RELATIONAL DATABASE VS. BLOCKCHAIN

	BLOCKCHAIN	RELATIONAL DATABASE
Authority	Decentralized	Centralized
Architecture	Peer-to-peer model	Client-server model
Performance	Relatively slower	Fast
Cost	Costly	Cheap
Data Handling	Only read and write	Create, Read, Update, Delete
Data Integrity	Has data integrity	Doesn't have data integrity
Transparency	Transparent	Non-transparent
Cryptography	✓	X

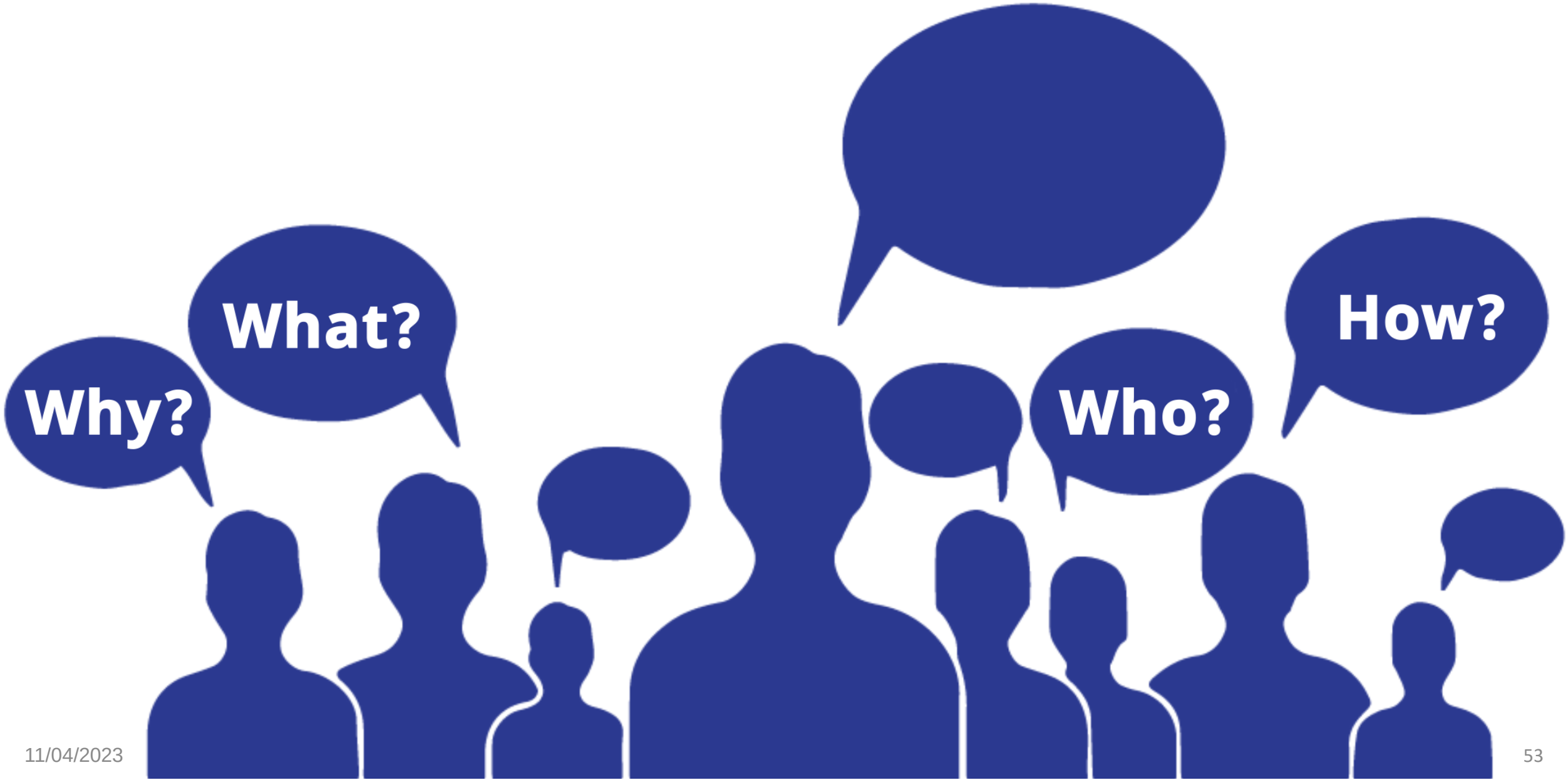
3.6. BLOCKCHAIN DB VS RELATIONAL DB

	Blockchain	Databases
Data Integrity	The blockchain structure makes it virtually impossible for someone to change the data without breaking the chain.	A malicious actor can potentially alter data if necessary measures are not taken.
Transactions	Data can only be read or added to the blockchain.	Data can be created, read, updated, or deleted (CRUD operations).
Querying Performance	The verification methods to ensure data integrity can slow down the querying and general performance of a blockchain.	Databases provide blazing-fast access to the data.
Structure	Blockchains can be fully decentralized and not rely on any central authority.	Databases are centrally managed, and an administrator owns and controls the data.

4. SUMMARY

- Data: data, type, structure, merkel tree, processing, infomation
- Blockchain data: identity, address, account, transaction, block, block of chain, blockchain states, accounting model.
- Blockchain database: database, database management system, blockchain database.

5. DISCUSSION



FINISH