1

Thinking in Layers and aspects

- Functional vs Non Functional LAYER
- Application vs Implementation ASPECTS
- Analogy of Mobile Phone
- Integrity
  - Data Integrity
  - Behavioral Integrity
  - Security

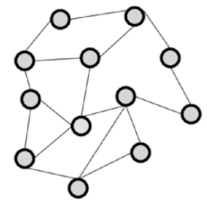
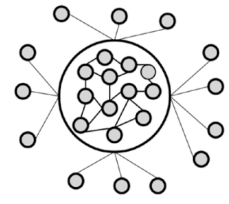
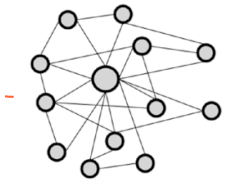
Table 1-1. Example of Mentally Layering a Mobile Phone

Layer	Functional Aspects	Nonfunctional Aspects
Application	Taking photos	The graphical user interface looks beautiful
	Making phone calls	Easy to use
	Sending e-mails	Messages are sent fast
	Browsing the Internet	
	Sending chat messages	
Implementation	Saving user data internally	Store data efficiently
	Making a connection to the nearest mobile connector	Saving energy
		Maintaining integrity
	Accessing pixels in the digital camera	Ensure user privacy

2

Seeing the Big Picture

Purpose of Blockchain:
To achieve and maintain integrity in distributed systems

- Architectures
  - Centralized
    
  - Distributed
    
    - ADVANTAGES
      - Higher computing power
      - Cost reduction
      - Higher reliability
      - Ability to grow naturally
    - DISADVANTAGES
      - Coordination overhead
      - Communication overhead
      - Dependency on networks
      - Higher program complexity
      - Security issues
  - Hybrid
    - 
      Looks Centralized but is Decentralized
    - 
      Looks Decentralized but is Centralized
- Payment System
  

Layer	Functional Aspects	Nonfunctional Aspects
Application	Deposit money	The graphical user interface looks beautiful
	Withdraw money	Easy to use
	Transfer money	Transfer of money is done fast
	Monitor account balance	System has many participants
Implementation	?	Available 24 hours a day
		Fraud resistant
		Maintaining integrity
		Ensure user privacy

Recognizing the Potential

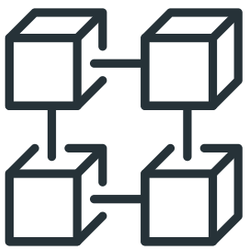
Napster
example of a P2P system

Centtralized P2P

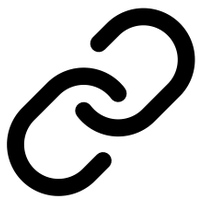
Middlemen of
immaterial goods and services
face a disruption risk

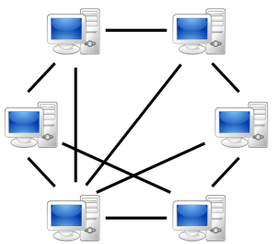
P2P system

Nodes that make their
computational resources
directly available to another



Blockchain





Purely distributed
P2P systems

Why Blockchain ?
Tool for achieving and
maintaining integrity in
distributed systems

Herding cats problem

4

Discovering the Core Problem

Trust

nonfunctonal aspect of the system

Systems that are Safe, consistent, correct and free of corruption and errors

Integrity

given in advance and will increase or decrease

Threats

Techinical Failures

Malicious Peers

How to achieve it ?

Depends on knowledge about the number of peers

Depends on the knowledge about the trustworthiness of the peers

Achieve Integrity and Trust in a Distributed system

5

Disambiguating the Term

Data Structure

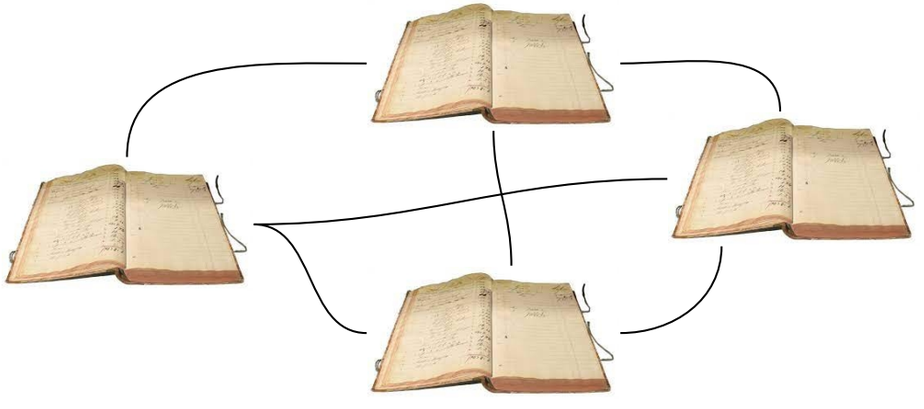
Algorithm

sequence of instructions
that negotiates the informational
content of many blockchain data structures

Suite of Technologies

- blockchain
  data structure
- blockchain
  algorithm
- Cryptographic
  technologies
- Security technologies

Purely distributed P2P
system of ledgers that utilize
blockchain tech suite



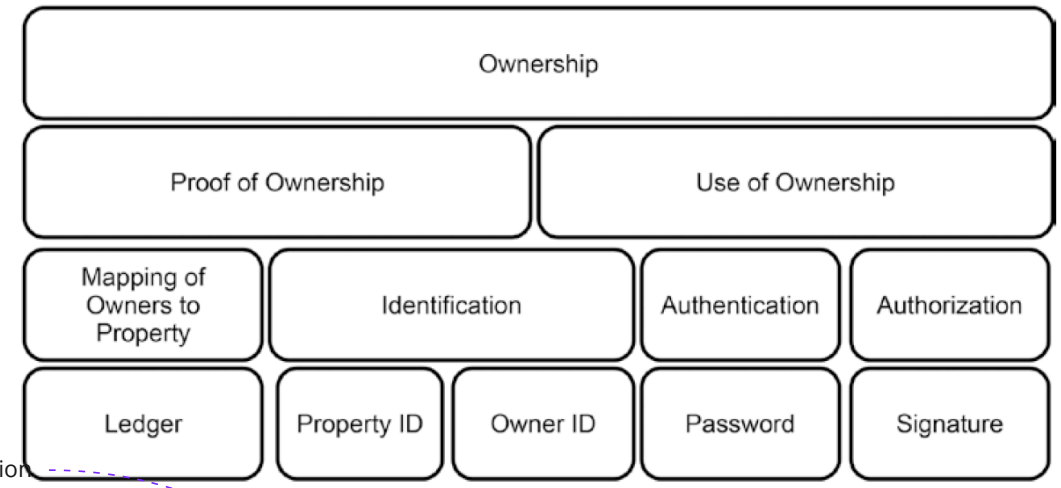
Managing
Ownership

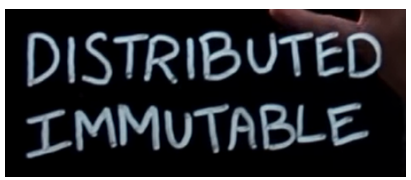
Understanding the Nature of Ownership

Independent Witnesses

How to prove that the apple in your bag is actually one from a different store that you have already purchased ?

Concepts of Ownership





Security

- Identification
- Authentication
- Authorization

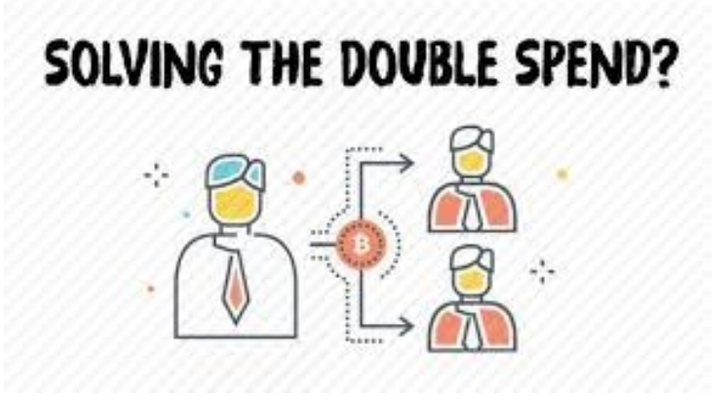
Principles of Ledger

- Conflicting forces of transparency vs privacy
- Having single ledger is RISKY

You walk in to a store and want to buy wine? What are the steps that take place before you walk away with wine ?

7

Spending Money Twice

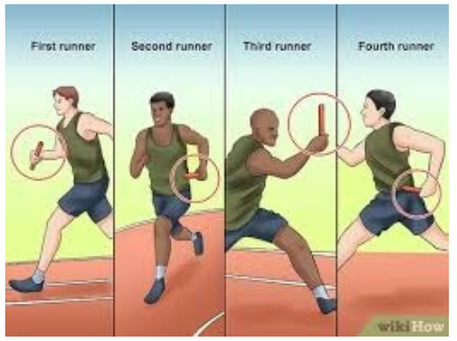


- Double Spend Problem
  - Problem caused by copying digital goods
  - Problem that may appear in distributed P2P ledgers
  - Violated integrity of purely distributed P2P
- Data consistency problem across distributed ledgers

Planning the Blockchain

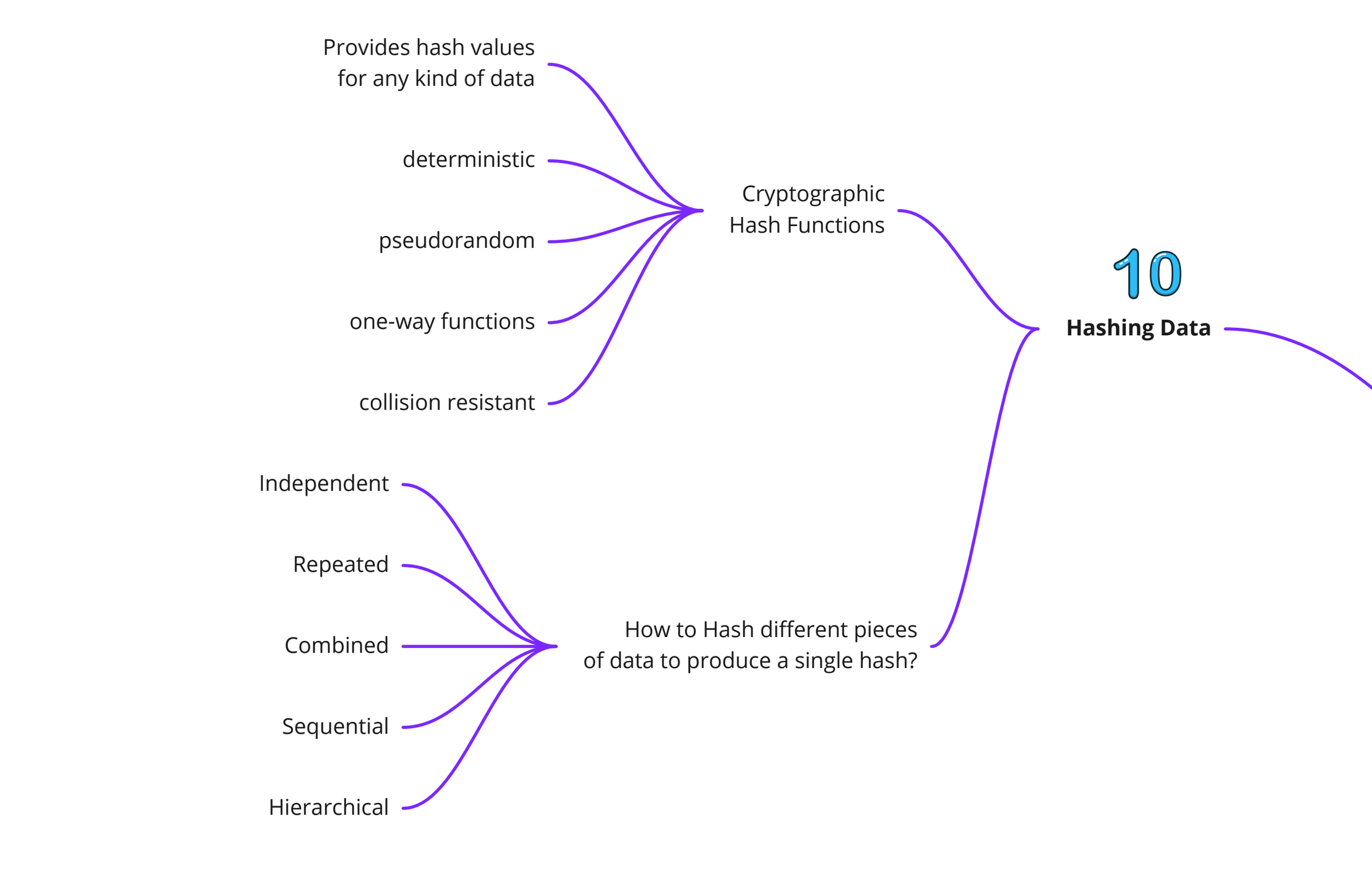
- Describing Ownership
- Protecting Ownership
- Storing Transaction Data
- Preparing Ledgers to be distributed in an untrustworthy environment
- Distributing the ledgers
- Adding New transactions to the ledgers
- Deciding which ledger represents the truth

9 Documenting Ownership



Analogy
Relay Race - Runners passing Baton

- Two ways to describe Ownership
  - Inventory Data
  - Transaction Data
- Two Step process
  - Describing the transfer of ownership
  - Maintaining the history of transfers
- Importance of Ordering
- Integrity of Transaction History
  - Formal Correctness
  - Semantic Correctness
    - Meaning and Intended effect
  - Authorization


Provides hash values for any kind of data
deterministic
pseudorandom
one-way functions
collision resistant
Cryptographic Hash Functions
10
Hashing Data
Independent
Repeated
Combined
Sequential
Hierarchical
How to Hash different pieces of data to produce a single hash?

11

Hashing in the Real World

Data being referred to has not changed since the reference was created

Hash References as Cloak Room Tickets

Cloak Room Ticket that points to a Coat Hook

Point to other data

Cloak Room tickets that reference other tickets

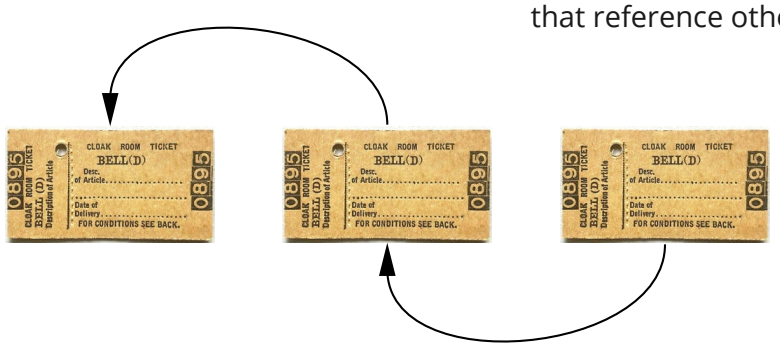
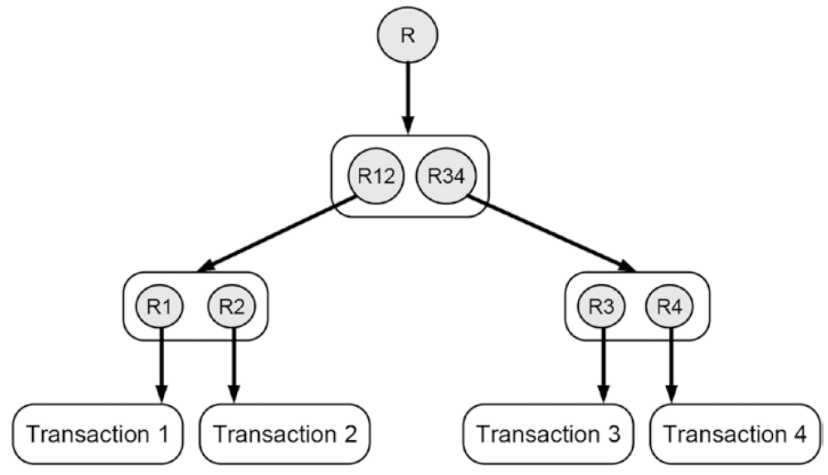


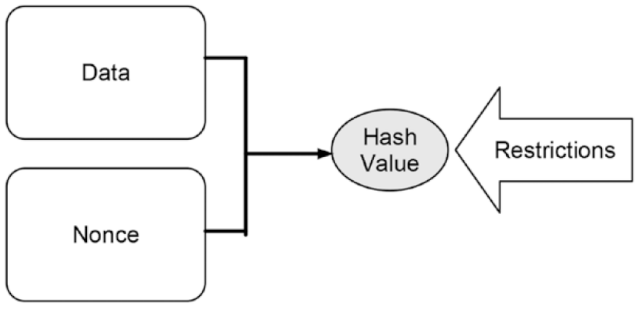
Figure 11-4. Data linked together in a chain-like fashion

Chain

Merkel Tree

Two ways to store hash references





Hash Puzzle

Higher the difficulty level, the higher the # of zeros needed in the restrictions

Hash values uses :

Compare Data
Detect whether the data has been altered
Refer to data
Store a collection of data
Create computationally intensive tasks

Anyone can put letters in to it but only the owner with the key can access the letters

Mailbox analogy

12

Protecting User Accounts

Cryptography

Symmetric

Asymmetric

Creating the Keys

Using the Keys

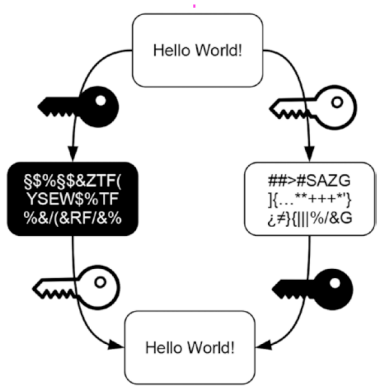
Public to Private

Private to Public

Usage

Identify Accounts

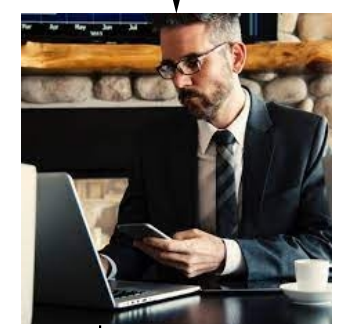
Authorize Transactions



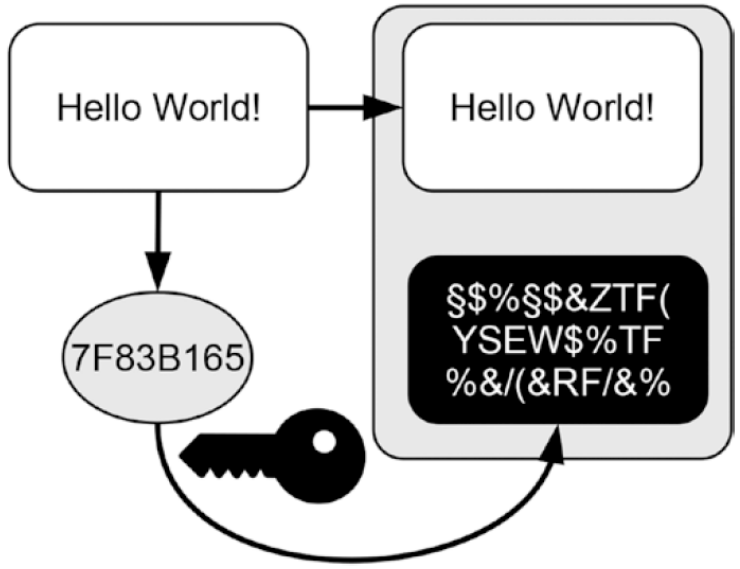
Identify user accounts

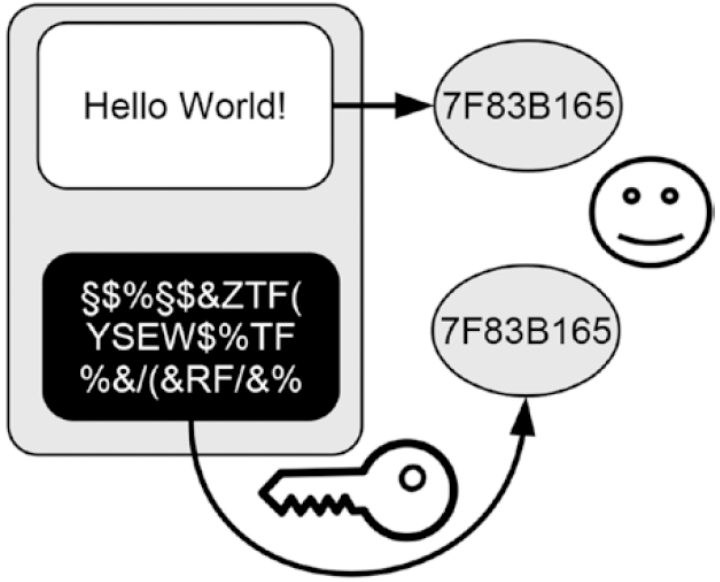
PUBLIC

Proof that the rightful owner is doing the transaction



PRIVATE





Creating a Signature

Verifying data using the Signature

Identifying Fraud using the Signature

13

Authorizing Transactions

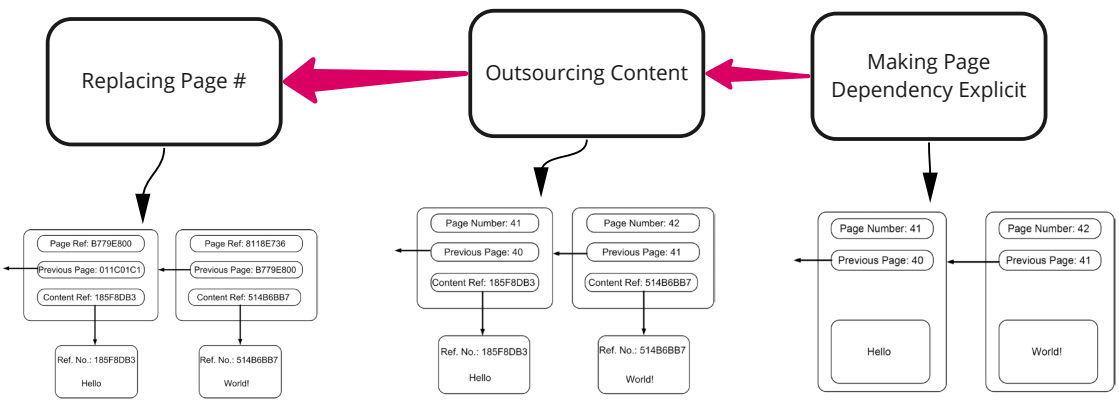
1. Create Hash value of the message
2. Encrypt the hash value
3. Send message and encrypted hash value

Card Catalogue Metaphor

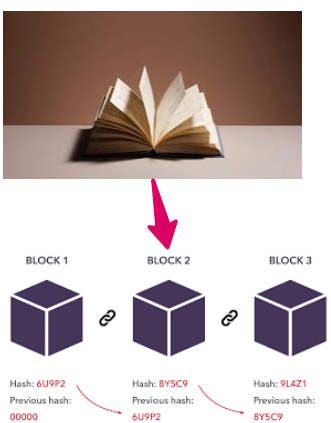
Cards are stored in the order in which books are added to the library

14

Storing Transaction Data



Transformed Book	Blockchain-Data-Structure[1]
A page in the ordering catalog	A block header
The whole ordering catalog	The chain of block headers
The reference number of a page in the ordering catalog	The cryptographic hash value of a block header
The reference number to the preceding page	The cryptographic hash value of the preceding block header
Content	Transaction data
A content page	A Merkle tree containing transaction data
Reference to the content page	The root of the Merkle tree that contains transaction data
The mental unit of a page of the ordering catalog and its corresponding content page	One block of the blockchain-data-structure
The whole ordering catalog and all content pages together	The blockchain-data-structure



Adding a new block is easy but changing the block somewhere in the chain is very elaborate

15
Using the Data Store

Knitting Metaphor

Adding New Transaction Data

Detecting Changes

Radical all-or-nothing approach to changing data

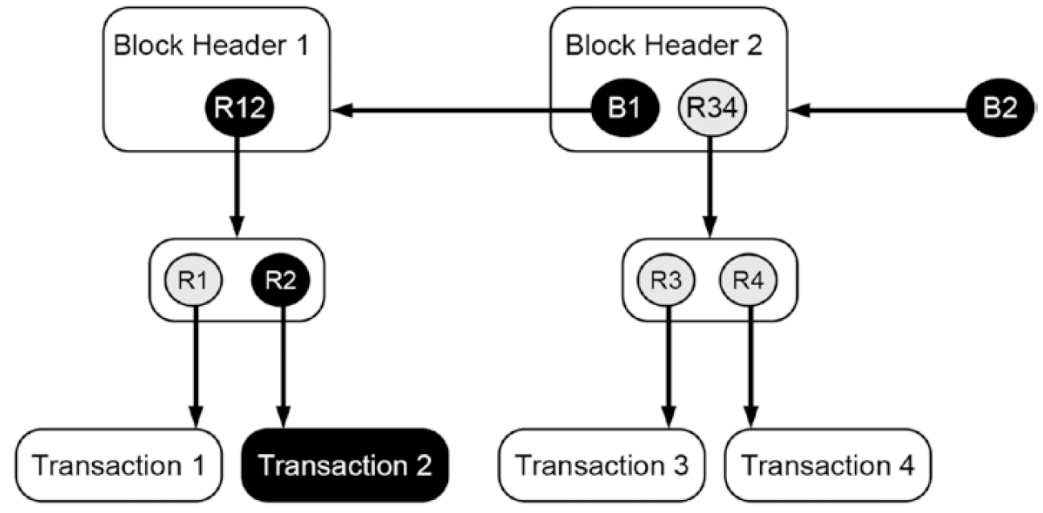


Figure 15-10. Changing a transaction orderly includes changing all subsequent hash references

Changing Family Tree Metaphor

Making Manipulations Stand out

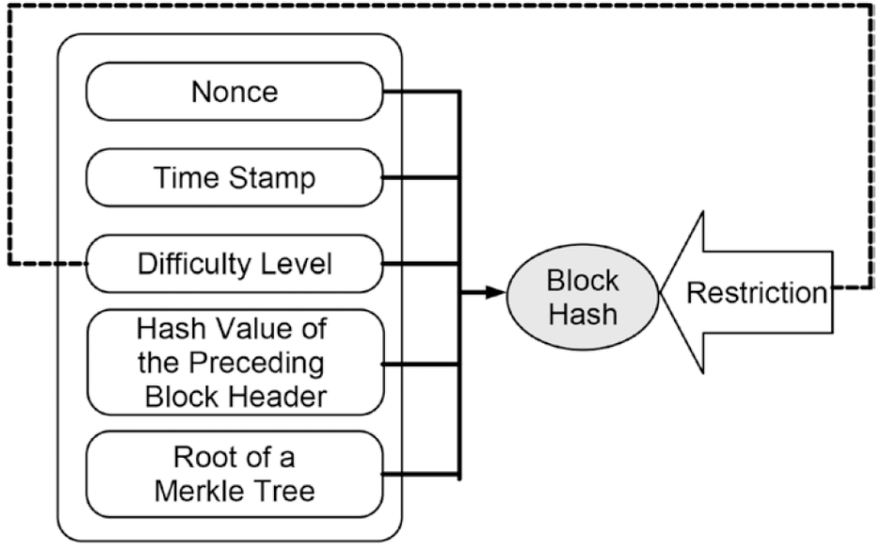
Enforce rewriting history for embedding changes

Making adding data computationally expensive

Hash Puzzle

16

Protecting the Data Store



17

Distributing the Data Store Among Peers

Gossip Metaphor

Message Delivery
- Not guaranteed to arrive at the addresses
- May arrive more than once
- May arrive in a different order

How to handle hurdles ?
- Use Gossip style
- Messages have digital fingerprint
- Timestamps

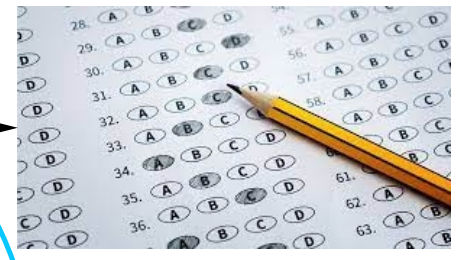
Communication
- Keep existing connections alive
- Establishing new connections
- Distributing new information

18

Verifying and Adding Transactions

Outsourcing company that grades MCQs

Answer Sheets + Solutions available with every evaluator



Metaphor

1. All answer sheets to be marked, the solutions as well as all marked answer sheets are available to all contractors at any time through the company's software system.
2. Only the first contractor who marks an answer sheet correctly receives one dollar as a reward.
3. If a contractor finds out that another contractor marked an answer sheet incorrectly, the contractor who made the mistake has to repay the compensation and the one who found and corrected the mistake will receive the compensation instead.

Components

- Validation Rules
  - Transaction Data
  - Headers
- Reward
- Punishment
- Competition
  - Speed
  - Quality
- Peer Control

Peer Control

1. Evaluating a new block that was created and submitted by one of the peers
2. Trying hard to be the next node that creates a new block that in turn has to be evaluated by all others

Less often paths disappear as nature takes back its territory

Unofficial Paths in a Park - Metaphor

Longest Chain criterion

Heaviest Chain criterion

Criterion

Orphan Blocks

Reclaimed Reward

Clarifying Ownership

Reprocessing of Transactions

Growing Common Trunk

Eventual Consistency

Robustness against manipulations

Consequences

19

Choosing a Transaction History

Continuous Voting Schema

Makes Payment to employees

METAPHOR
BreadMaking company pays employees with BREAD

20

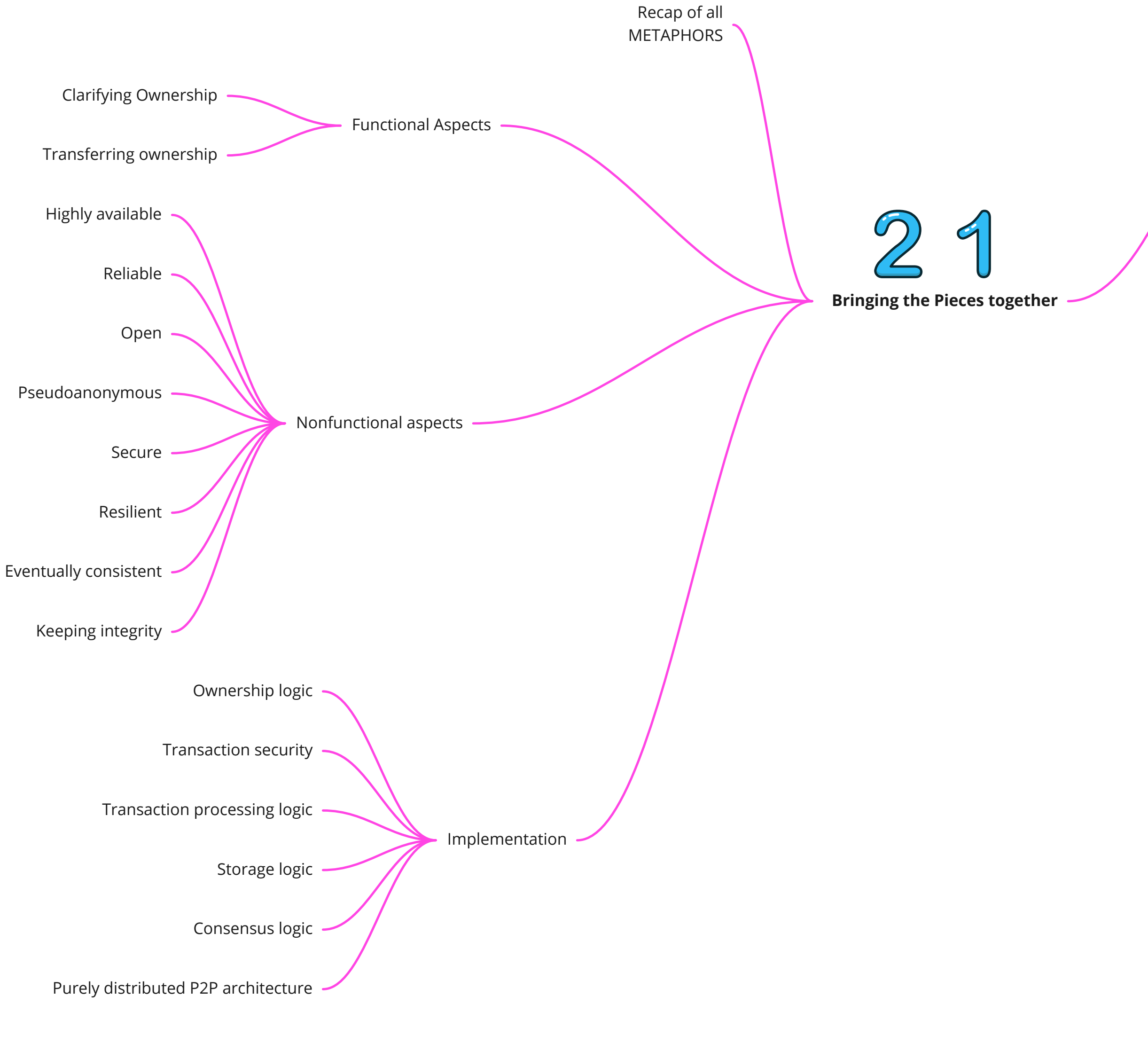
Paying for Integrity

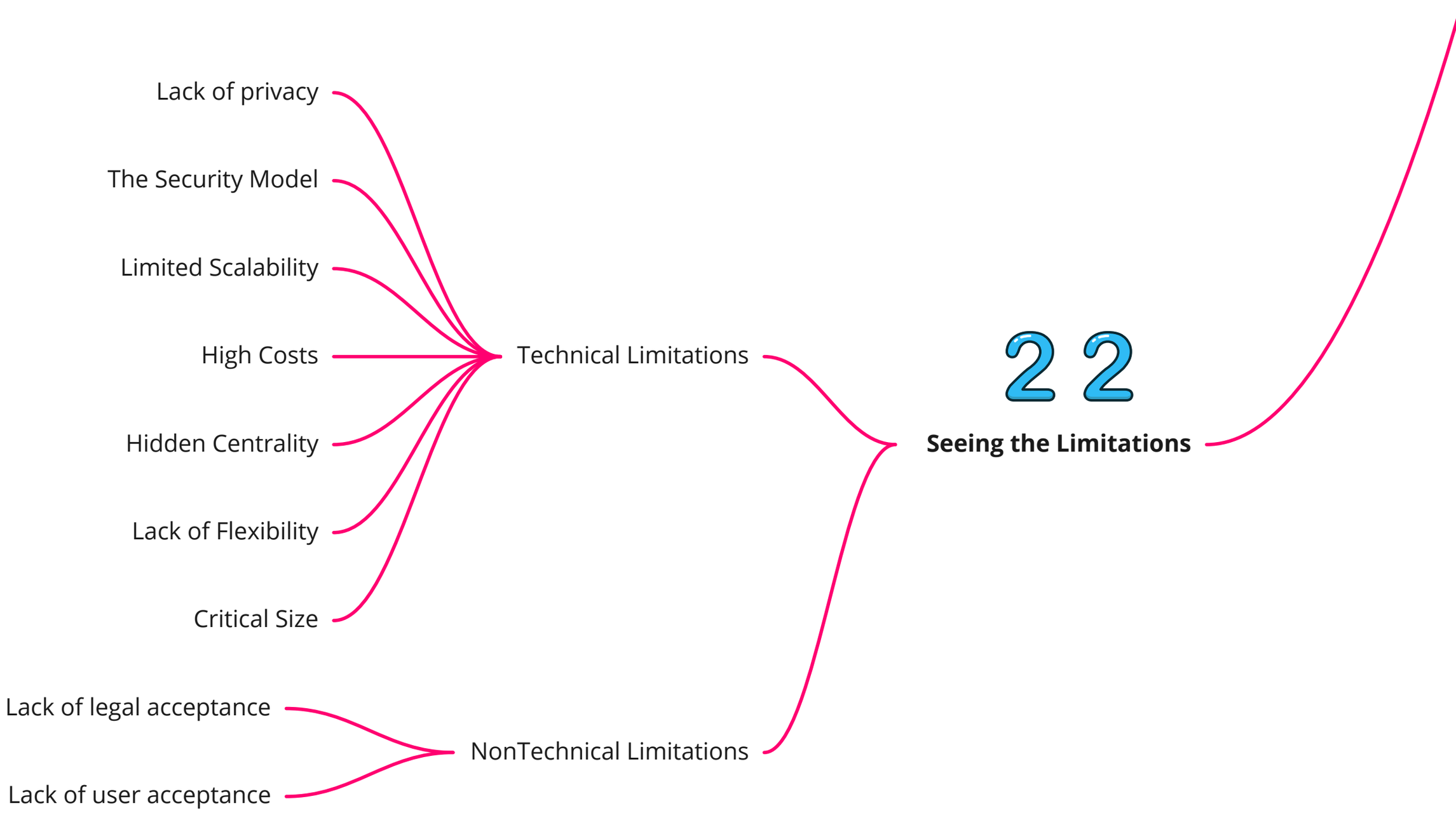
Consequences of choosing an instrument

- Integrity
- Openness
- Distributed nature
- Philosophy

Desirable properties

- Digital form
- Widely accepted
- Not subjected to capital movement restrictions
- Stable value
- Trustworthy
- Decentralized


Recap of all METAPHORS
Clarifying Ownership
Transferring ownership
Functional Aspects
Highly available
Reliable
Open
Pseudoanonymous
Secure
Resilient
Eventually consistent
Keeping integrity
Nonfunctional aspects
21
Bringing the Pieces together
Ownership logic
Transaction security
Transaction processing logic
Storage logic
Consensus logic
Purely distributed P2P architecture
Implementation


Lack of privacy
The Security Model
Limited Scalability
High Costs
Hidden Centrality
Lack of Flexibility
Critical Size
Technical Limitations
Lack of legal acceptance
Lack of user acceptance
NonTechnical Limitations
22
Seeing the Limitations

Transparency vs. privacy

Security vs. speed

Conflicting Goals

Public vs. Private blockchains

Permissionless vs Permissioned

Solving Conflicts

Four versions

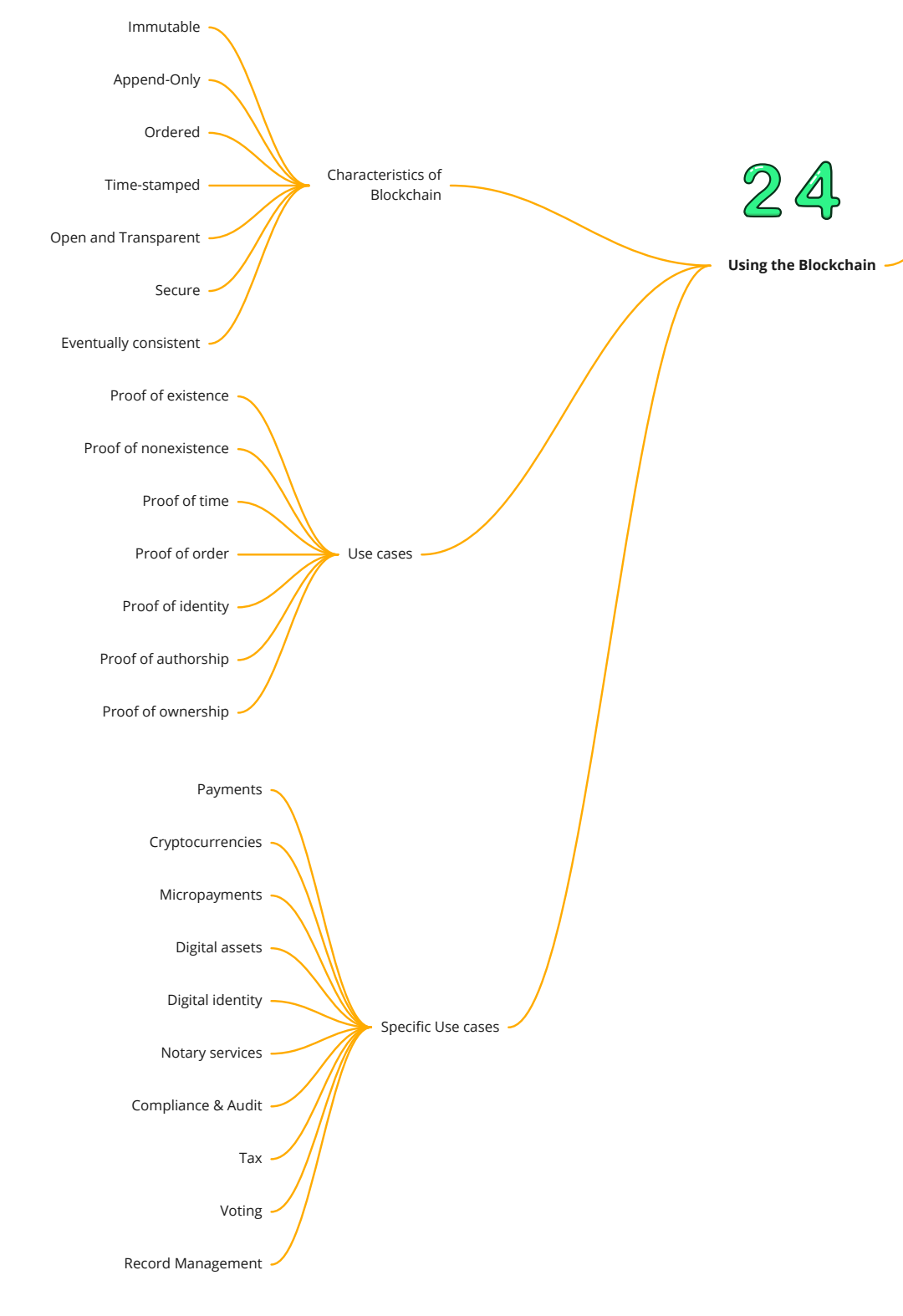
23

Reinventing the Blockchain

Writing Access	Reading Access and Creation of Transactions	
	Everyone	**Restricted**
Everyone	Public & Permissionless	Private & Permissionless
Restricted	Public & Permissioned	Private & Permissioned

Consequences

P2P architecture
Distributed Nature
Purpose


24
Using the Blockchain
Characteristics of Blockchain
Immutable
Append-Only
Ordered
Time-stamped
Open and Transparent
Secure
Eventually consistent
Use cases
Proof of existence
Proof of nonexistence
Proof of time
Proof of order
Proof of identity
Proof of authorship
Proof of ownership
Specific Use cases
Payments
Cryptocurrencies
Micropayments
Digital assets
Digital identity
Notary services
Compliance & Audit
Tax
Voting
Record Management

Decentralization

Non Discrimination

Openness

Universality

Consensus

Tim Berners - Lee vision of Internet

Looks similar to Blockchain's vision

25

Summarizing and Going Further

Access rights

Zero Knowledge Proofs

Privacy

Proof of Stake

Paxos

Raft

Consensus

Transactions

Useful for Smart Contracts

Inventory Data

DAG based

Data structure

Conceptual Evolution