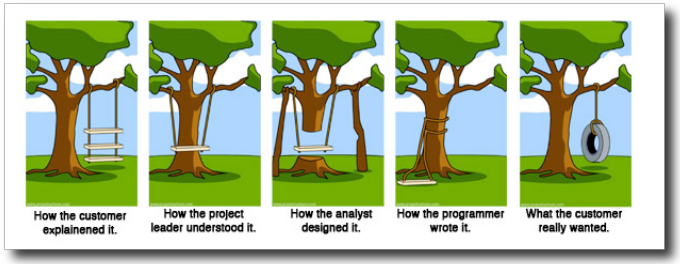
The blockchain technology: benefits, types, and challenges

Roman Vitenberg
UNIVERSITY OF OSLO

Motivation

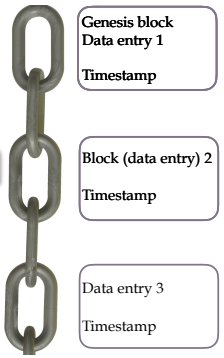
- Many blockchain applications with very different requirements
- Different types of blockchain, with different benefits and guarantees
- Matching between the requirements and technology is important



A way to arrange your data

Corruption-resistant sequence of data entries shared over a network by multiple parties

- Protects against
  - **Corruption** of data in any single entry
  - **Deletion** of entries (once there stays forever)
  - Only allows adding entries to the end of the sequence
- Makes compaction of storage essential to any solution
- It is a **sequence**, not a collection of data entries
  - The order is strictly enforced and corruption-resistant
  - Important for related entries (conflicting transactions)
  - Performance cost for ordering unrelated entries



Types of blockchain: Private

- There is a **single** enterprise or organisation that
  - **Writes** all the data
  - Manages **read** access permissions for individual users
- Centralised (one writer) but the data can still be distributed across multiple servers within the same organisation
- Benefits
  - **Corruption-resistance** (but there are many alternative solutions)
  - Marketing benefits thanks to blockchains being popular

Types of blockchain:

- Public: everyone in the world can read or write
- Consortium: there is an agreed set of participating organisations that write the data and manage read access
  - Consortium can be updated but every participant must be known to each other participant
- Public and consortium blockchains are collectively called decentralised

Main benefit of decentralised blockchains

Enable parties who do not fully trust each other to form and maintain consensus about the existence, status and evolution of a set of shared facts

- Without a trusted intermediary
  - Can be **expensive**
  - Requires access to your confidential data
  - Can be vulnerable to attacks
- Requires (and implements) an **agreement** among the participants to update the data and order these data updates
  - Called a **consensus protocol**

Protection against mistrusted parties

- Protects against possible misbehaviours
  - data corruption and deliberate falsification
  - giving a priority to "desirable" updates and starving undesirable ones
  - compromised machine that belongs to a participant
- Works only if majority of participants are "honest" (no 51% attack)
  - No colluding majority coalition
  - Less than majority is compromised
- Requires additional mechanisms to
  - Protect against hiding specific data entries and limiting read access to specific users
  - Provide data confidentiality and privacy of writes or reads

Smart contracts

A programmatic intermediary that runs on a blockchain

- Multiple benefits
  - cheaper than a human intermediary
  - open and transparent program that fulfils the contract and does nothing else
    - does not peak into your data
  - highly resistant to attacks

Challenge: requires that participant's database be compatible with blockchain

Consortium vs Public blockchains

- The consortium can **change** blockchain **rules**, **revert** data **updates**
  - Flexible, yet the power can alienate customers, depending on the app
- Well-known consortium makes a 51% attack less likely
  - Thus, a simpler and more **efficient consensus protocol** can be used
- Smaller scale and enterprise servers make updates **cheaper**
- Fast proprietary networks make updates **faster**
- Controlled read permissions make it easier to provide **data confidentiality** yet make it less **transparent**
- No easy cooperation with enterprises outside of consortium
  - Change of consortium is a heavyweight operation
  - **Integration** is painful

Storage techniques for blockchain

- Partition the data and use multiple smaller blockchains instead of a single big one
- Archive old data entries if the data are not in use any longer
- Store only newer entries on most servers while only a small number of servers keep all the entries
- Store most data off-chain with the hash being on-chain
- Partition the blockchain and store different data entires on different servers
- Partition the off-chain data and store on different servers
- Within the same enterprise or belonging to different organisations
- IPFS can be used in this context

My pet challenges

- Improved consensus protocols suited for blockchains
  - Protocols used in public blockchains allow < 10 updates per second and take up to 10 minutes for an update to complete
  - No standards for consortium blockchains but known protocols do not scale beyond 30 participants
- Solutions for data partitioning and exchange protocols that guarantee data availability
  - IPFS as a package is much better suitable for public blockchains
- Partitioning into multiple blockchains and data consistency across them
- Reliability of the solution and other Quality of Service

Potential and expertise in the area

- Area where companies have the need while research institutions have the know-how
  - Ethereum's foundation blog discusses individual publications by Cornell and University of Singapore in detail
  - IPFS was initially conceived by Juan Benet while in Stanford
  - The peaks of activity at industry and research coincide!
- 15-20 years of experience in consensus protocols, data storage and availability, fault-tolerance, data consistency, P2P data exchange
- ~7 years of experience in security and privacy in distributed systems (secure P2P, Tor)
- Related commercial projects while working for IBM
  - IBM corporate award for the data availability component in WebSphere
- Collaboration with world-leading research centres in this area and with Norwegian companies in adjacent areas (privacy and security in other contexts)