SRA: The Security Reference Architecture for Blockchains:

Towards a Standardized Model for Studying Vulnerabilities, Threats, and Defenses

Ing. Ivan Homoliak, Ph.D.

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

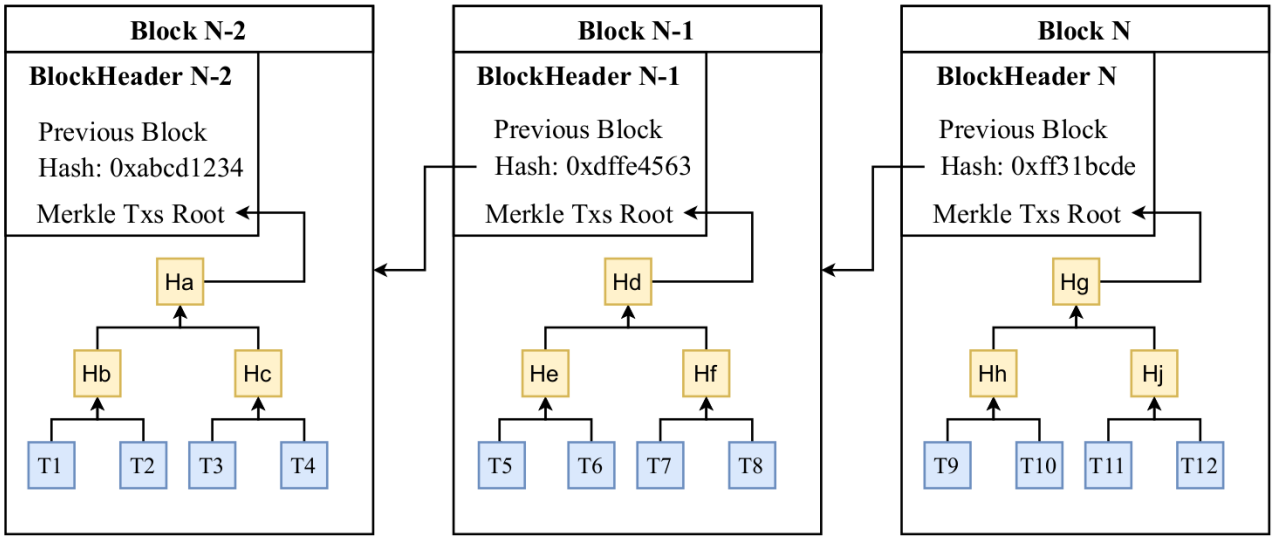
30th May 2022

Missing standardized model to study security threats and mitigation techniques on the blockchains

Background

What is a blockchain?

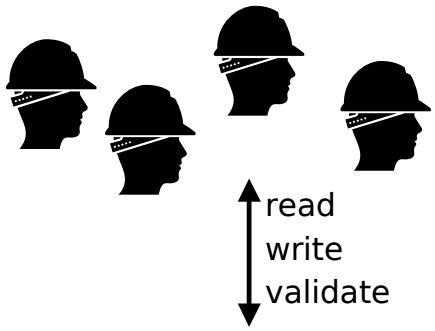
- **General-purpose global DB (a ledger)**
  - append-only
  - replicated by each participant (i.e., validating node)
  - modified by txs
    - batched in blocks
    - each tx signed by a client/user
  - each block references the predecessor by H(header)

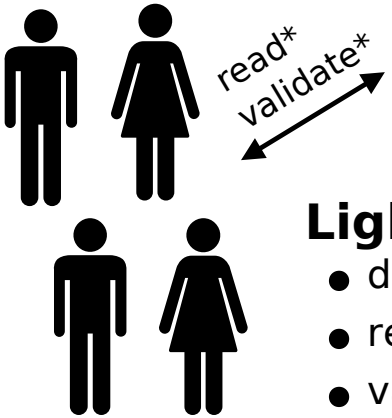


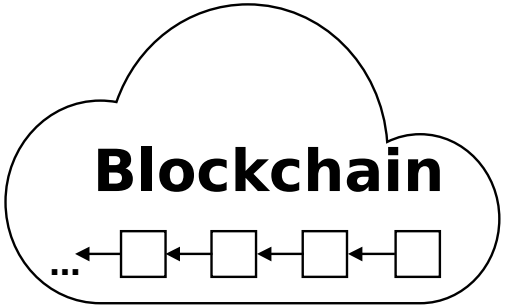
Background - Involved Parties

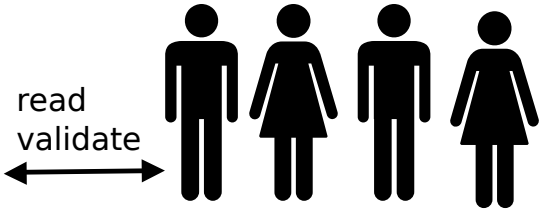
Consensus Nodes

- disseminate txs and blocks
- read blockchain
- write to blockchain
- validate blockchain









Lightweight Nodes

- disseminate own txs
- read blockchain (partially)
- validate blockchain (partially)

Validating Nodes

- disseminate txs and blocks
- read blockchain
- validate blockchain

Why Blockchains?

- **Problems of Centralized Information Systems (IS)**
  - Single-point-of-failure (e.g., HW faults, DDoS to a service)
  - Censorship of clients => no proofs
  - Limited availability (less than 100%)
  - No integrity guarantees on stored data
    - tampering with the data
    - "proofs" about the content are weak => e.g., at courts
    - weak auditability (e.g., incidents)
  - No transparency

Why Blockchains?

- **Decentralization**
  - Avoids single-point-of-failure
  - Extremely high availability (~100%)
- **Censorship-resistance**
  - All requests eventually processed
- **Immutability**
  - Append-only design
- **Auditability**
  - Correctness of each tx and block can be validated
- **Transparency**
  - Txs are visible

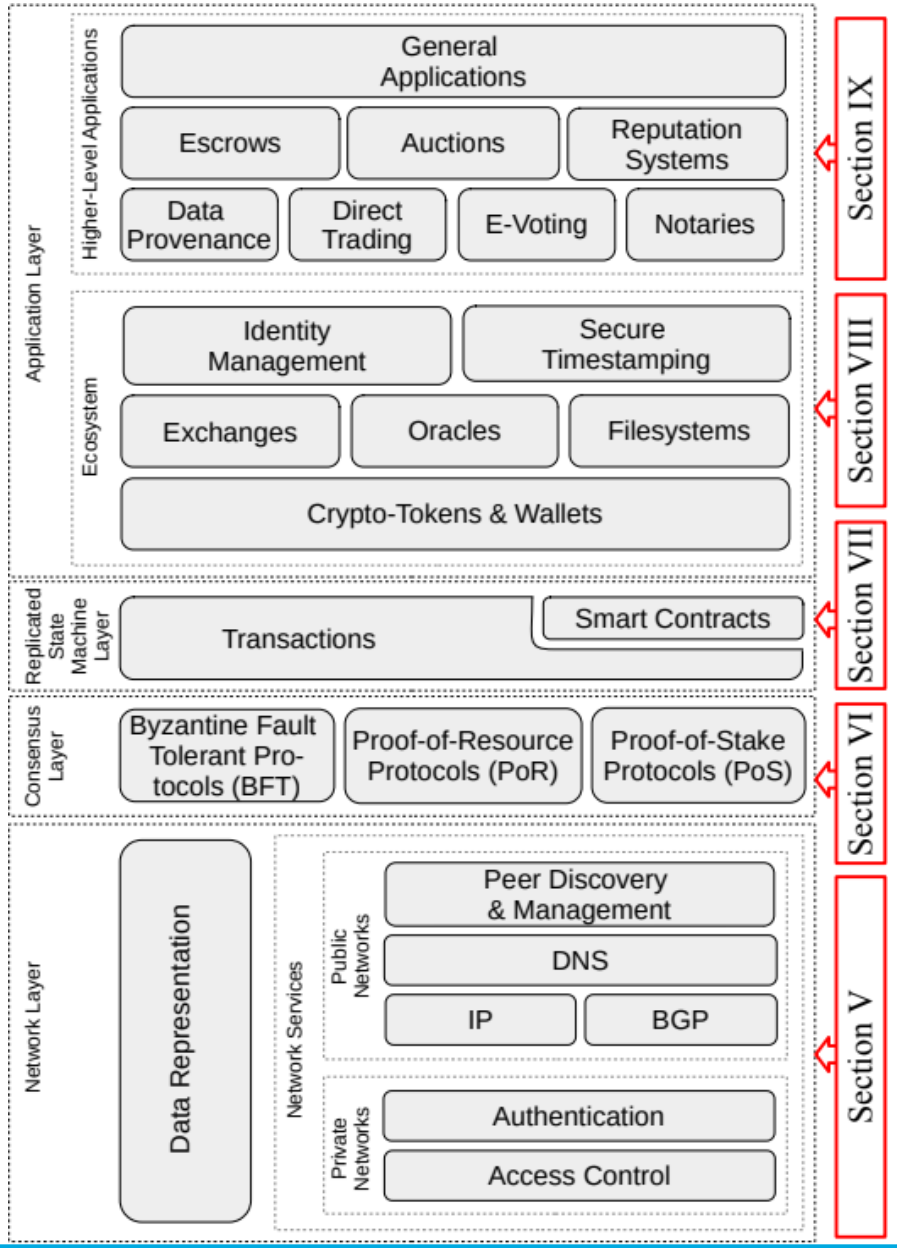
Types of Blockchains

- **How a new node enters a blockchain?**
  - **Permissionless blockchains** (Proof-of-Resource)
    - anyone can join (w/o permission)
    - to prevent Sybil attacks, nodes establish their "virtual identities" by running a Proof-of-Resource protocol (consensus power of a node is proportional to its resources allocated).
  - **Permissioned** (Proof-of-Authority)
    - a new node has to obtain permission to join from a centralized or federated authority(ies)
    - nodes usually have equal consensus power (i.e., one vote per node)
  - **Semi-Permissionless** (Proof-of-Stake)
    - new node has to obtain some form of permission (i.e., stake)
    - such permission can be given by any consensus node
    - consensus power of a node is proportional to the stake

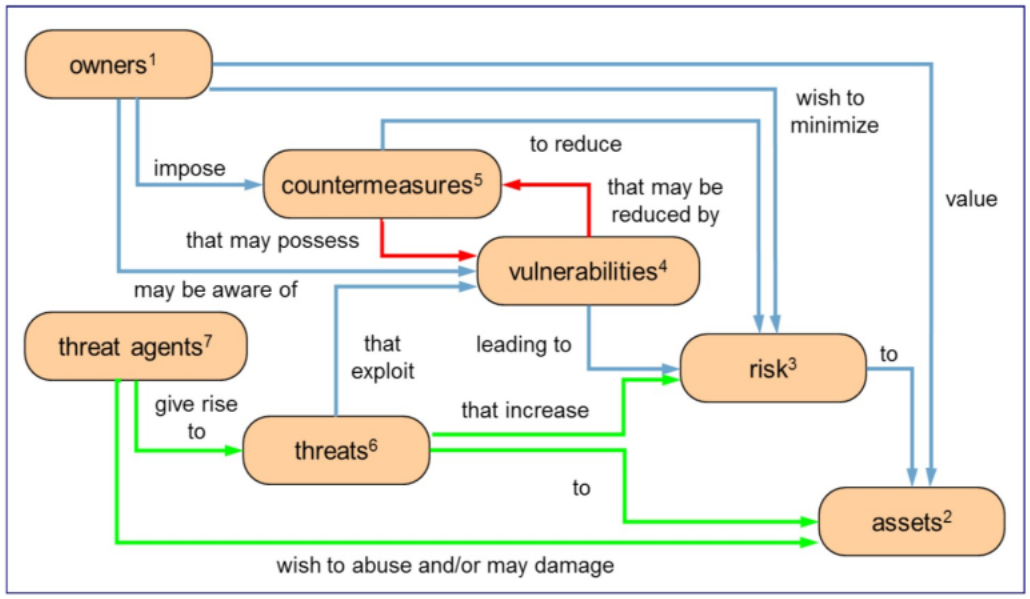
SRA for Blockchains

SRA for Blockchains: Top-Down View

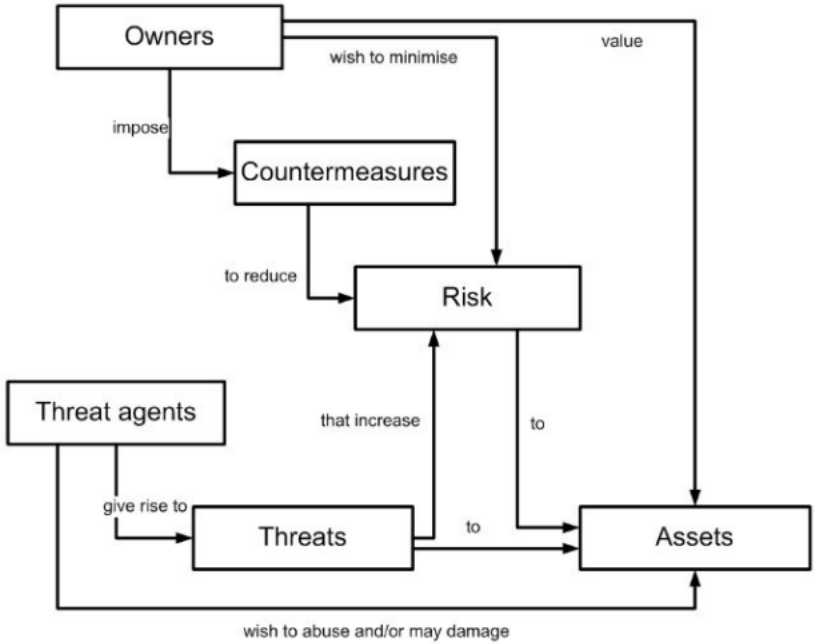
- **Layered / stacked model**
  - Similar to ISO/OSI model
  - 4 layers
  - Focus on security & privacy
  - Incentive mechanisms part of consensus layer
  - Surveys **reasonable** blockchain applications



Standard for Threat-Risk Assessment



ISO/IEC15408:1999

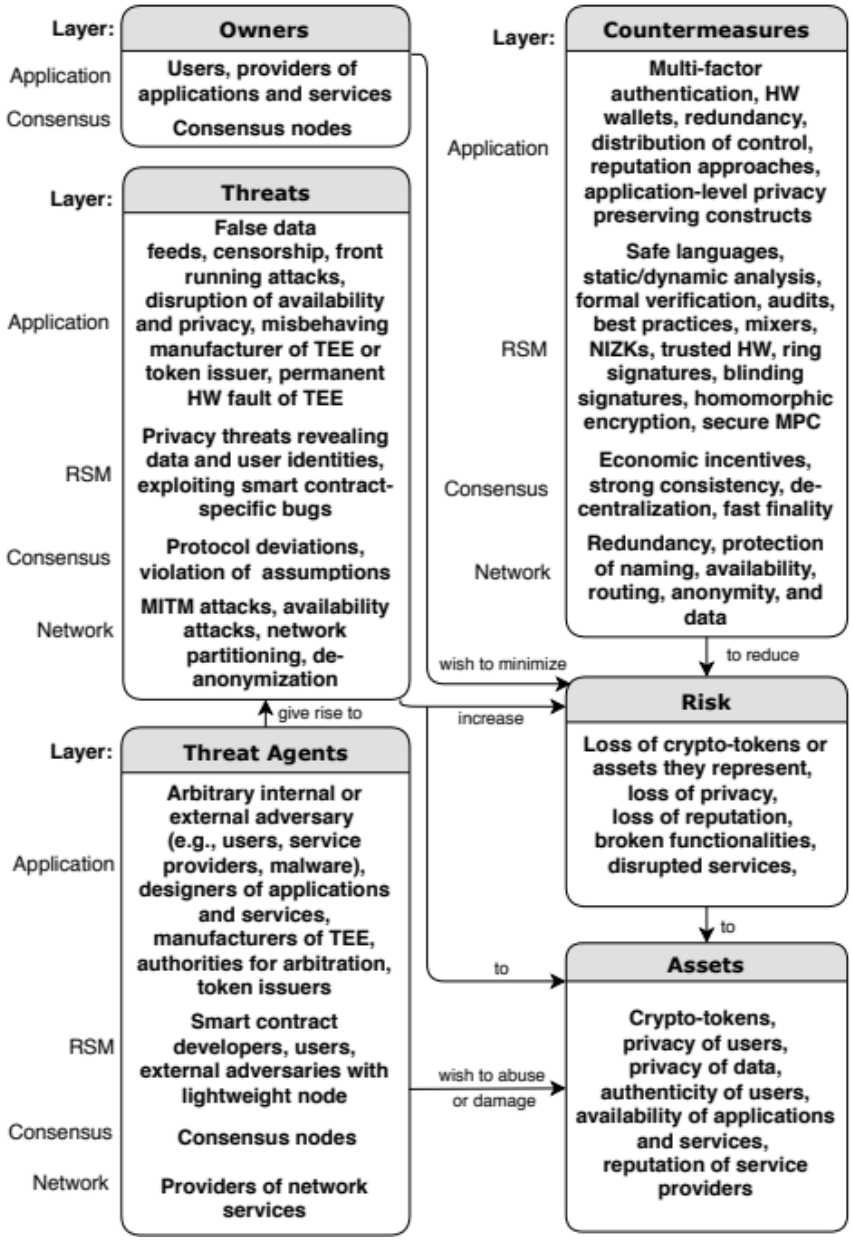


ISO/IEC15408:2017

- Captures the essence of a system's security
- Built before invention of blockchains
  - Misses publicly trust-less systems

Threat Risk Assessment Model for Blockchains

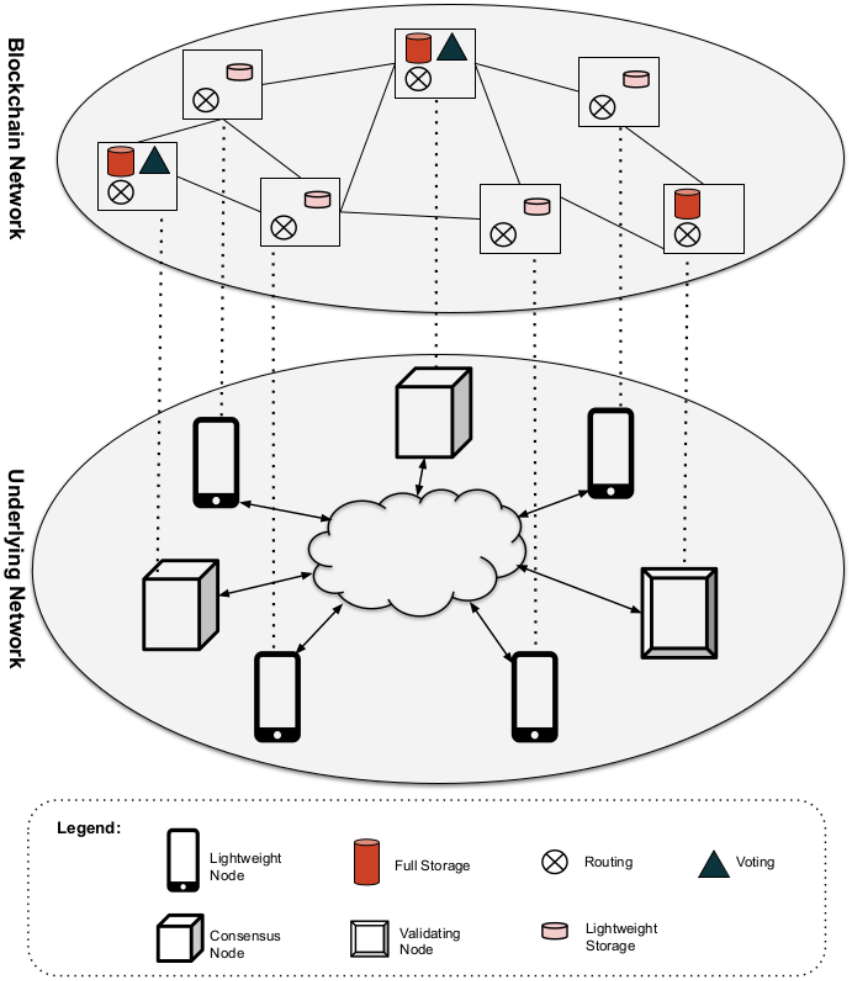
- Accommodates blockchains
- Projects SRA to threat/risk assm.
- We will expand it into VTD graphs
  - Vulnerability
  - Threat
  - Defense



Network Layer

Network Layer

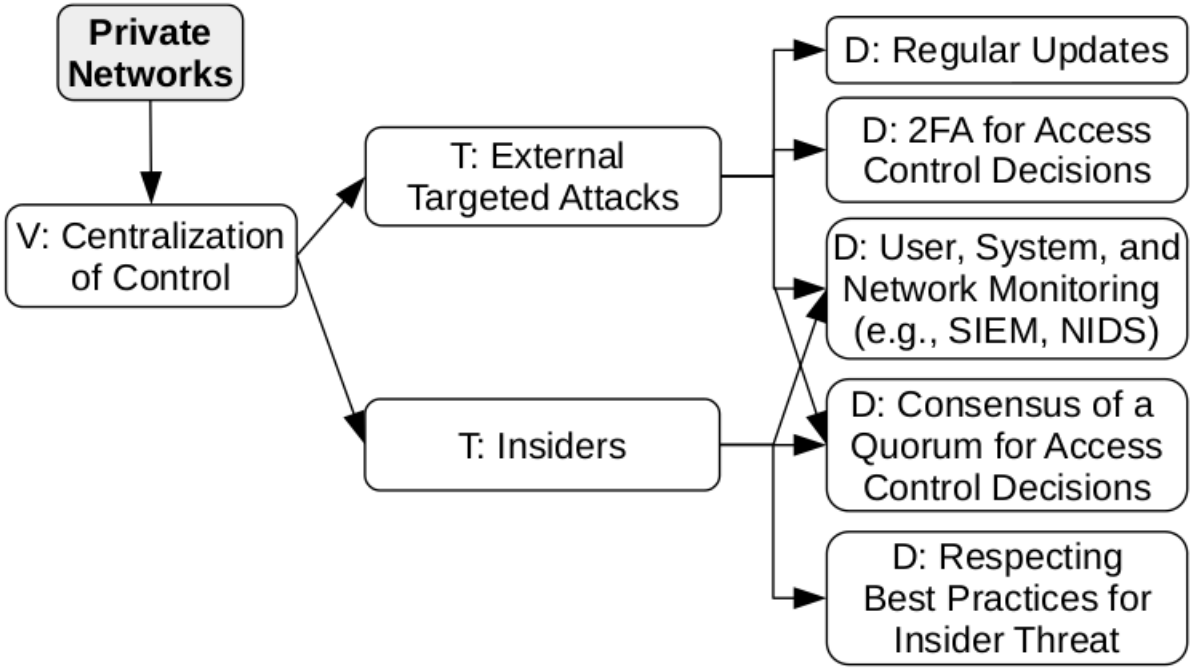
- **Blockchains introduce p2p overlay networks** built on top of other networks
  - Inherit security and privacy issues
- **Divided into 2 sub-planes**
  - **Data representation**: cryptography ensuring data integrity, authentication, and optionally privacy, confidentiality, anonymity, non-repudiation, and accountability
  - **Network services**: peer management and discovery, domain name resolution (i.e., DNS) and routing protocols
- **Accessibility**
  - Can be viewed as permission to enter
  - Splits networks to **private** and **public**



Private Networks

- **Pros**
  - Access control is achieved by centralized authentication of users
  - Data privacy is ensured by permissioned settings
    - User identities might be revealed in a private group
  - Full control over routing paths and physical resources
    - Regulation of the network topology w.r.t. requirements
  - Fine-grained authorization controls
    - The security principle of minimal exposure
    - Mitigate insider threat attacks
- **Cons**
  - VPN is required to communicate between private networks spread over different geographical locations
  - Suitable only for permissioned/restricted blockchains

Private Networks - VTD Graph

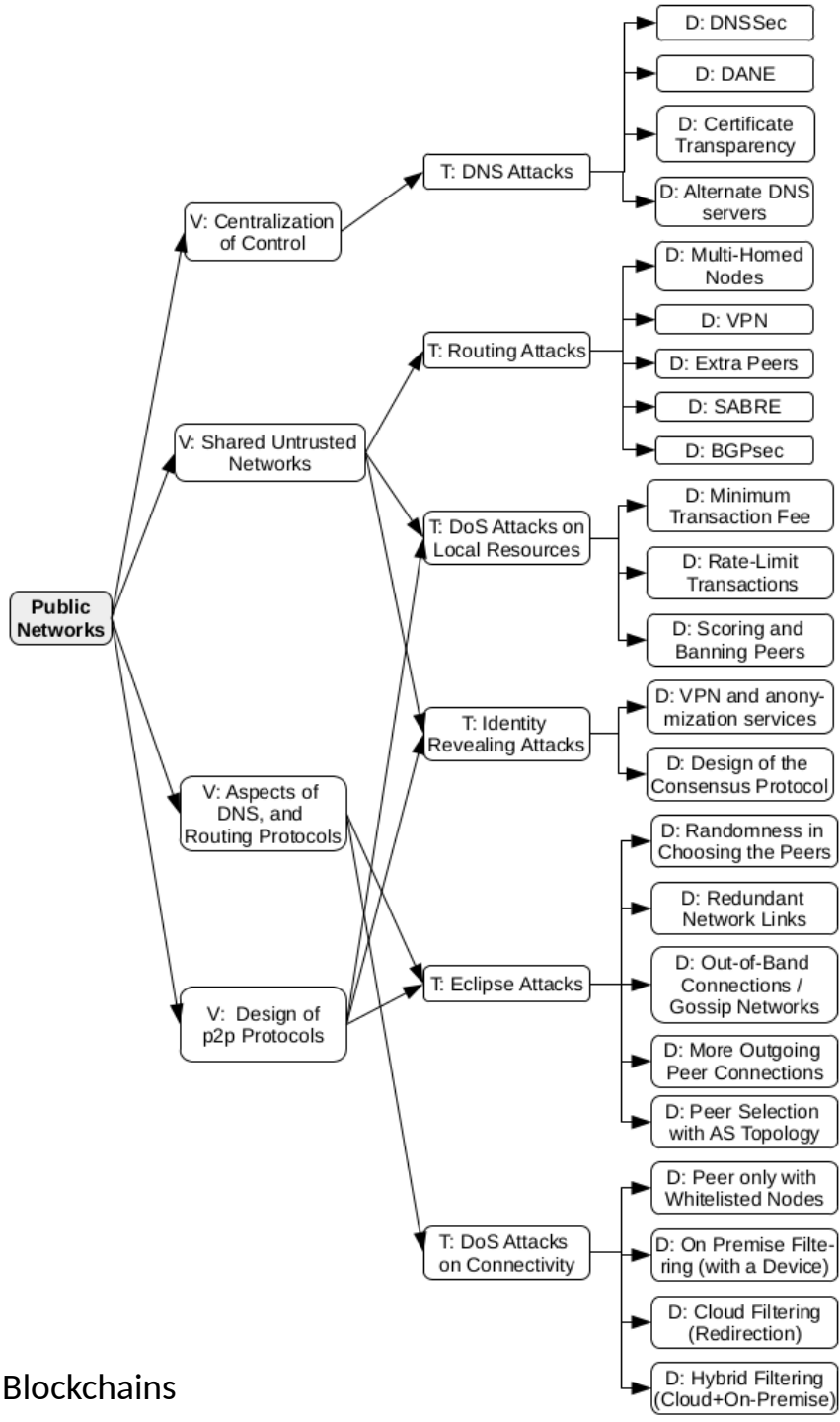


Public Networks / the Internet

- **Pros**
  - **High availability and decentralization** (geographical dispersion)
  - **Openness and low entry barrier**
    - Technology interoperability (e.g., using TCP/IP), economic factors (e.g., low cost of broadband connection), and societal factors (e.g., resistance to regulations)
- **Cons**
  - **Single-point-of-failure**
    - DNS with its hierarchy, IP addresses, and autonomous systems managed by centralized parties (ICANN/IANA)
  - **External attackers**
    - (1) Resources under attacker control (e.g., botnets, DNS and BGP servers), (2) stolen or masqueraded identities (e.g., IP addresses in an eclipse attack or route manipulation), (3) MITM attacker (i.e., eavesdropping and spoofing), (4) the exploitation of common network vulnerabilities, (5) revealing secrets (e.g., de-anonymizing peers)
  - **Distribution of infrastructure is not uniform** => overall latency increased
    - Might result into loss of created blocks (wasting consensus power)

Public Networks - VTD Graph

- **DNS Attacks**
  - Peers obtained from a hard-coded list of DNS seeders => DNS cache poisoning
- **Routing attacks** - traffic route diversions, hijacking, or DoS attacks
  - May lead to network partitioning
- **Eclipse attacks** aim to hijack all connections of a node to its peers
  - Arise from threats on DNS and routing, or result from vulnerabilities in p2p protocols
- **DoS attacks on connectivity** of consensus nodes may result in a loss of consensus power (and rewards)
- **DoS attacks on resources** (e.g., memory and storage) may reduce the peering and consensus capabilities
  - Penny-flooding - of the network with low fee txs can cause memory pool depletion => crash
- **Identity revealing -** linking the IP with address in tx



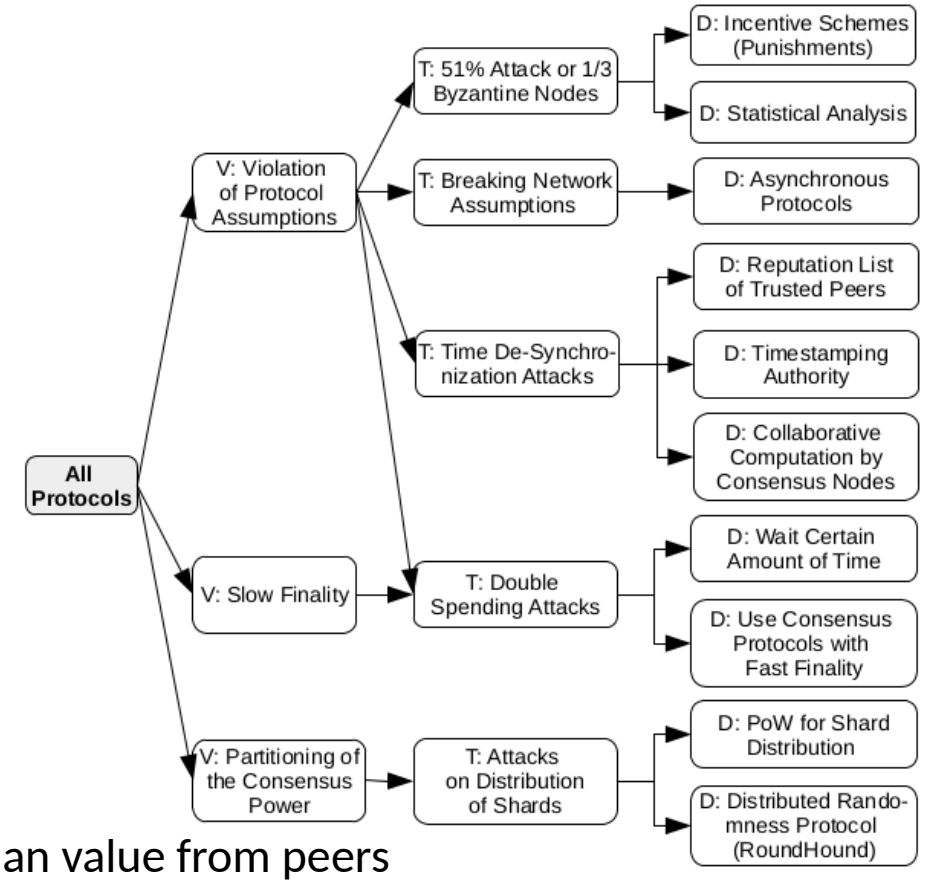
Network Layer - Summary of Options

Layer	Category in a Layer	Pros	Cons
Network Layer	Private Networks	• low latency, high throughput • centralized administration, ease of access control • privacy of data, privacy of identities • meeting regulatory obligations • resilience to external attacks	• VPN is required for geographically spread participants • suitable only for permissioned blockchains • insider threat at nodes with administrative privileges
	Public Networks	• high decentralization • high availability • openness & low entry barrier (low cost of broadband connection, resistance to regulations)	• high and non-uniform latency • single point-of-failure (DNS, IP, and ASes are managed by centralized parties) • external adversaries (botnets, compromised BGP/DNS servers) • stolen identities

Consensus Layer

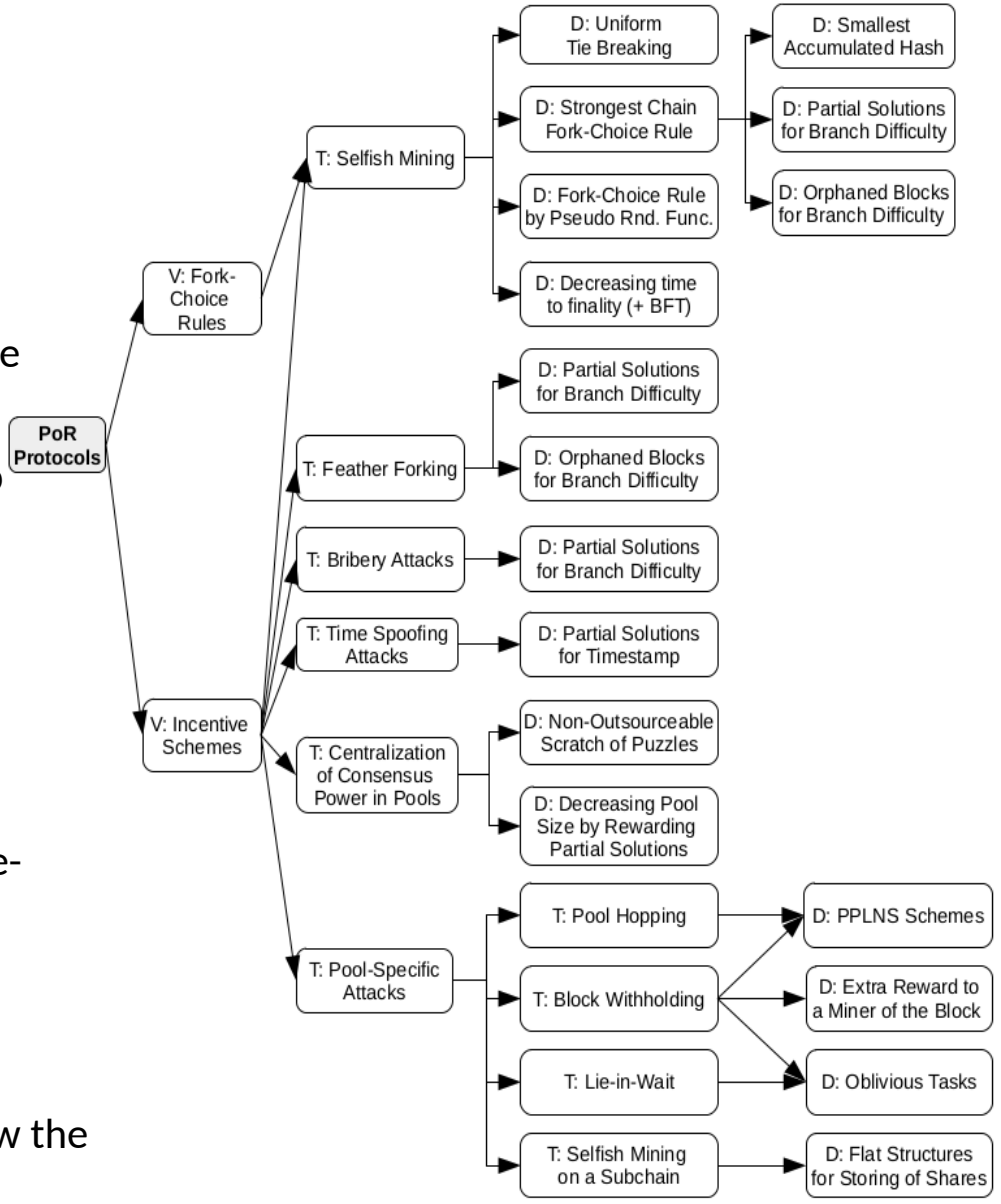
Consensus Layer

- **Deals w ordering of txs**
- **Three categories of protocols**
  - PoR protocols
  - PoS protocols
  - BFT protocols
- **Generic attacks** (to all protocols)
  - Centralization of consensus power
  - Breaking network assumptions
  - Time de-synchronization
    - Network time computed as the median value from peers
    - Attacker can slow down the victim node's network time
    - When victim creates a block, it can be discarded due to time constraints
  - Double spending
  - Attacks on shards - attacker might obtain a majority of consensus power in a shard



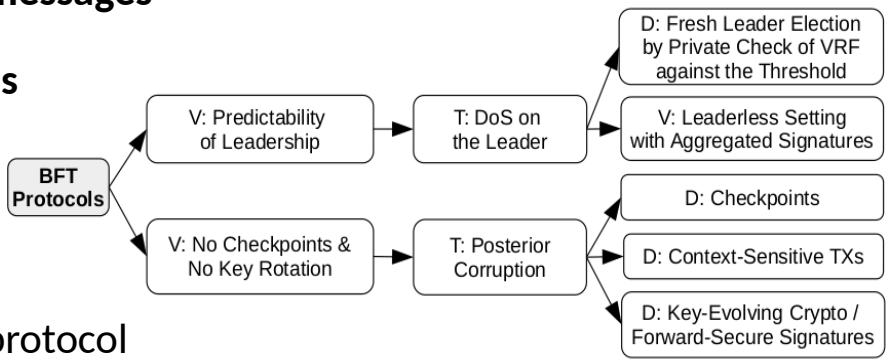
Consensus Layer - PoR Protocols

- **Selfish mining (**w 33% mining power)
- **Feather forking**
  - The attacker creates incentives for rational miners to collectively censor certain txs
    - Before a mining round begins, she announces that she will not extend the block with blacklisted txs
  - Rational nodes prefer to join the censorship to avoid the potential loss
  - Not profitable for the attacker
- **Bribery attacks**
  - Offering of direct rewards to miners
  - Consensus nodes might be bribed to double-spend or reorder txs in a block (enabling transaction front-running)
- **Time spoofing attacks**
  - Target a time-based difficulty computation w the intention to decrease the difficulty



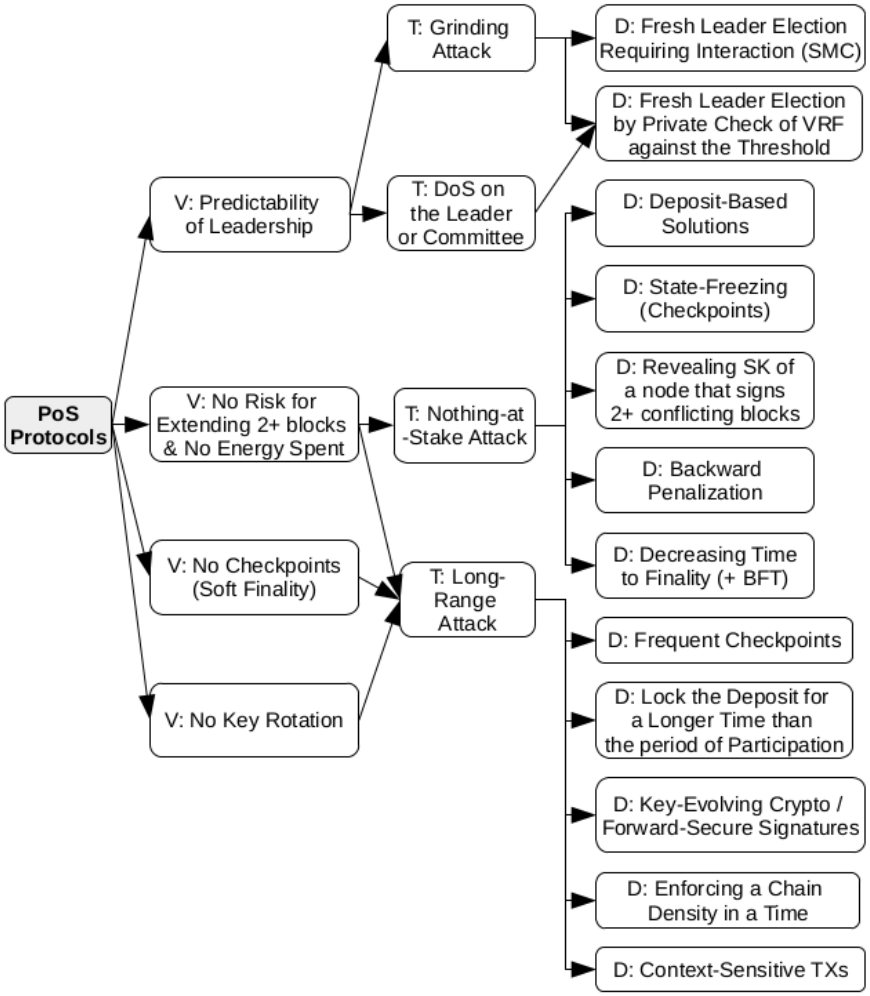
Consensus Layer - BFT Protocols

- **Assume a fully connected topology and broadcasting messages**
- **Mostly intended for (private) permissioned blockchains**
  - Run by trusted participants
- **DoS on a leader**
  - Assumes the node whose goal is to sabotage the protocol
  - Leader of the round is known before the round starts
  - The leader can be DoSed => a restart of the round
- **Posterior corruption** (a.k.a., long-range in PoS)
  - The adversary steals/buys private keys of 2/3 possibly “retired” consensus nodes and then rerun the consensus protocol
  - Violation of protocol assumptions, in which the adversarial consensus power reaches 2/3
  - Discussed mainly in PoS



Consensus Layer - PoS Protocols

- **Nothing-at-stake**
  - A node can extend 2+ conflicting blocks w/o risking its stake => increase its chance to be rewarded
  - Increases # of forks and thus time to finality
- **Grinding attack**
  - If the leader is known before the round starts, then the attacker can bias this process to increase her chances of being selected in the future
  - E.g., if a PoS protocol takes only h(prev. block) for the next leader election, the current leader might bias it
- **DoS on a leader/committee**
  - Equivalent in BFT
- **Long-range attack**
  - Equivalent of posterior corruption in BFT



Consensus Layer - Summary of Options

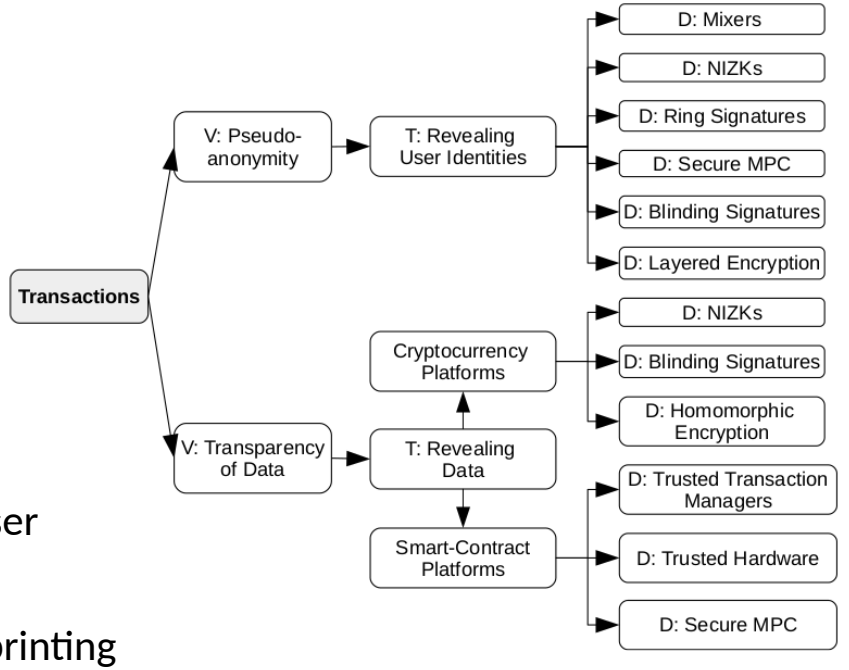
Layer	Category in a Layer	Pros	Cons
Consensus Layer	**PoR**	• high cost of overriding the history of blockchain • high scalability	• high operational costs • low throughput • low finality
	BFT	• high throughput (with a small number of nodes) • fast finality	• low scalability • high communication complexity • limited number of nodes (efficient use only in permissioned blockchains)
	PoS	• energy efficiency	• PoS specific attacks and issues • supports only semi-permissionless setting • slow finality
	PoS+BFT	• energy efficiency • high scalability • probabilistic security guarantees • lower communication overheads than BFT	• some PoS specific attacks • supports only semi-permissionless setting
	PoR+BFT	• high scalability • fast finality	• spending some scarce resources
	PoR+PoS (i.e., PoA)	• high scalability	• spending some scarce resources • some PoS specific attacks • slow finality

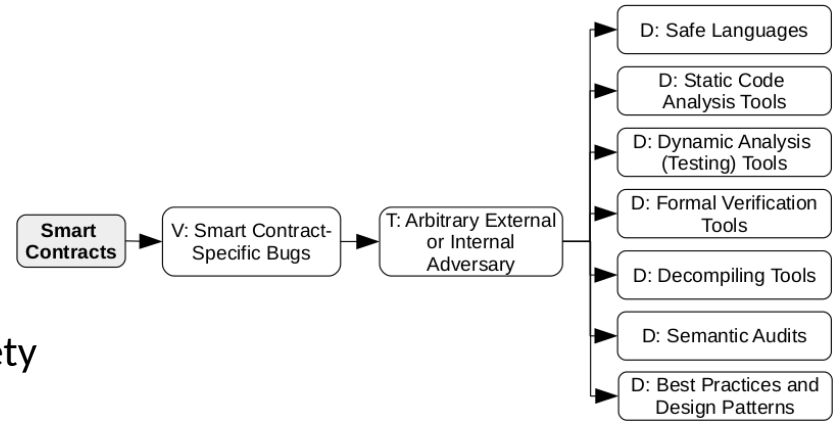
Replicated State Machine (RSM) Layer

(Transactions & Smart Contracts)

RSM Layer

- **Deals w interpretation of txs**
- **Two categories**
  - 1) Transaction threats
  - 2) Smart contract bugs
- **1) Transaction threats**
  - **Privacy threats to user identity**
    - A network-eavesdropping adversary can link user identities with IP addresses
    - Network analysis, address clustering, tx fingerprinting
  - **Privacy of data**
    - By default data in blockchain are public
- **2) Smart Contract bugs**
  - Turing-complete languages
  - Turing-incomplete languages
    - A small attack surface and the emphasis on safety
    - Limited expressiveness





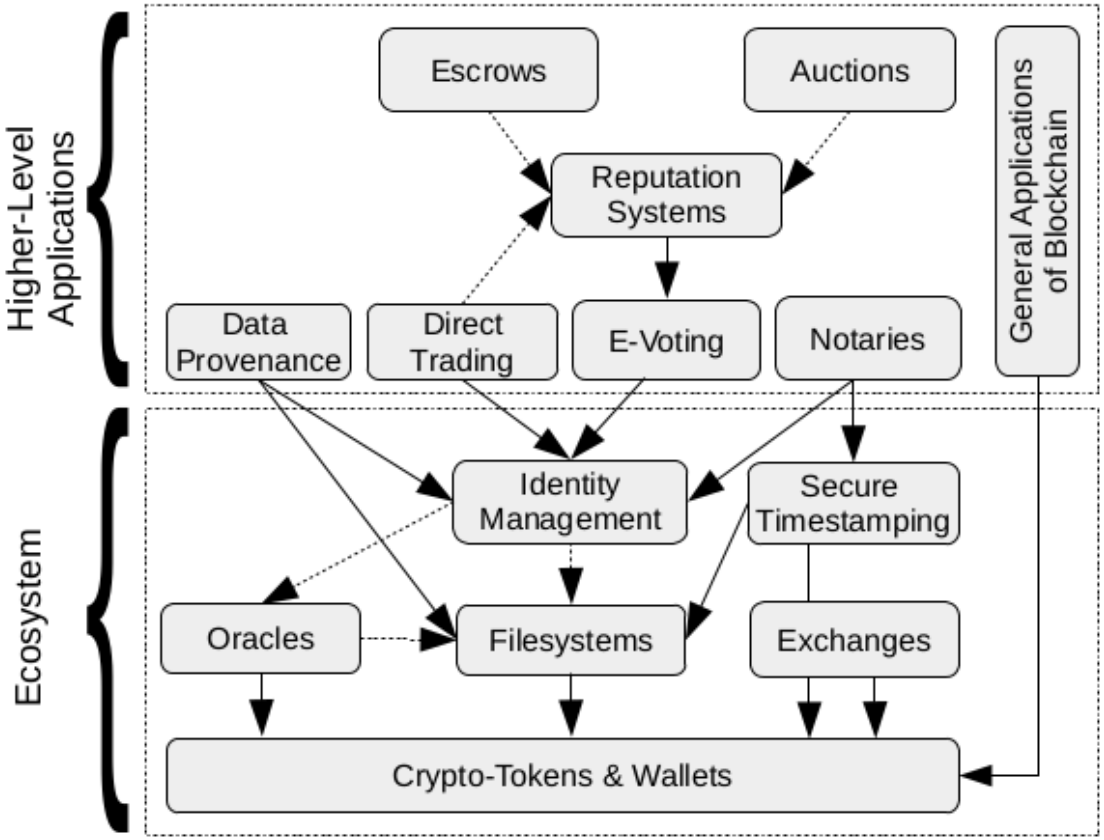
RSM Layer - Summary of Options

Layer			Category in a Layer	Pros	Cons
Replicated State Machine Layer	Transactions	Protecting Identities	**Standard Approach**	• fast processing • ease of verification	• identities are only pseoudonymous and can be traced to IPs • all data of transactions are publicly visible
			Standard Approach + Mixers	• privacy identity protection of users in a group • ease of verification	• additional complexity, in some cases unlinkability by the mixer or involved parties in a group • all data of transactions are publicly visible
			NIZKs and Ring-Signatures	• identities are anonymized to the extend of the group	• additional computation overheads for running the schemes
			MPC Blinding Signatures, Layered-Encryption	• unlinkability for all involved parties	• additional computation overheads for running the schemes
		Protecting Data	**NIZKs, Blinding Signatures, Homomorphic Encryption**	• privacy of data in cryptocurrency platforms	• additional computation overheads for running the schemes
			Trusted Transaction Managers, Trusted Hardware, MPC	• privacy of data in transactions of smart contract platforms	• additional computation overheads for running the schemes
	Smart Contracts		**Turing-Complete Languages**	• smart contracts may contain an arbitrary programming logic	• wide surface for making the programming bugs that often results in vulnerabilities
			Turing-Incomplete Languages	• small attack surface and emphasis on safety	• the programming logic serves only for limited purposes

Application Layer

Application Layer

- **Functionality-oriented categorization** of the applications running on or utilizing the blockchain
- Hierarchy in inheritance of security aspects across categories of the application layer
- Dotted arrows represent application-specific and optional dependencies`



App. Layer

Application Category	Subcategory	Pros	Cons
Wallets	Server-Side Hosted Wallets	• simplicity of control for end-users • no storage requirements for end-users	• keys stored at the server, susceptibility to the theft of keys by external or internal attacks • single-point-of-failure, availability attacks
	Client-Side Hosted Wallets	• simplicity of control for end-users • no storage requirements for end-users • keys stored locally	• single-point-of-failure, availability attacks • possibility of key theft by malware • possibility of tampering attacks
	Self-Sovereign Wallets	• keys stored locally or in a dedicated hardware device	• moderate storage requirements for end-users • more difficult control for end-users • extra device to carry in the case of hardware wallet
Exchanges	Centralized Exchange	• a high throughput and speed of operations • the simplicity of control for end-users • low costs for exchange transactions • trading of obscure crypto-tokens	• risk of insider threat due to centralization • external threats to exchange infrastructure • overheads for secure storage of secrets • a fee specified by the operator
	Direct Cross-Chain Exchange	• fairness of the exchange • no fee to any operator	• costs for 4 transactions of the atomic swap • user has to find the counter-order on her own • counter-orders might not exist • a lower throughput than in a centralized exchange • a higher complexity for end-users
	Cross-Chain DEX	• fairness of the exchange • order matching made by DEX • trading of obscure crypto-tokens	• costs for 4 or 6 transactions of the atomic swap • a lower throughput than in a centralized exchange • a fee specified by the operator
	Intra-Chain DEX	• fairness of the exchange • uniform finality for every pair • a high speed of operations	• a limited number of pairs that are specific to the target platform • a fee specified by the operator • costs for smart contract execution
Oracles	Prediction Markets	• early (close to accurate) estimation of the future event's result • decentralization	• possible conflict of interest • a limited set of data specific to a few events • a long time to obtain a final result, especially in the case of disputes
	Centralized Data Feeds	• wide range of data • fast provisioning time • handling of private parameters of requests • censorship evidence	• centralization (accidentally or intentionally wrong data) • availability issues
	Oracle Networks	• decentralization • wide range of data • fast provisioning time	• unsupported private parameters of requests • publicly visible data and requests
Filesystems	Fully Replicated FSs with Ledger	• a high availability • accountability and auditability	• a high storage overheads and operational costs • a high price
	Partially Replicated FSs with Ledger	• reasonably high availability • accountability and auditability • a lower price than in a full replication	• attack vectors specific to partial replication
	Partially Replicated FSs without Ledger	• reasonably high availability • a lower price than in a full replication	• a lack of native accountability and auditability • low durability due to a lack of incentives for storage
	Centralized Storage of Off-Chain Data	• a low price • accountability and auditability	• a low availability

Thank You for Your Attention !

Questions?

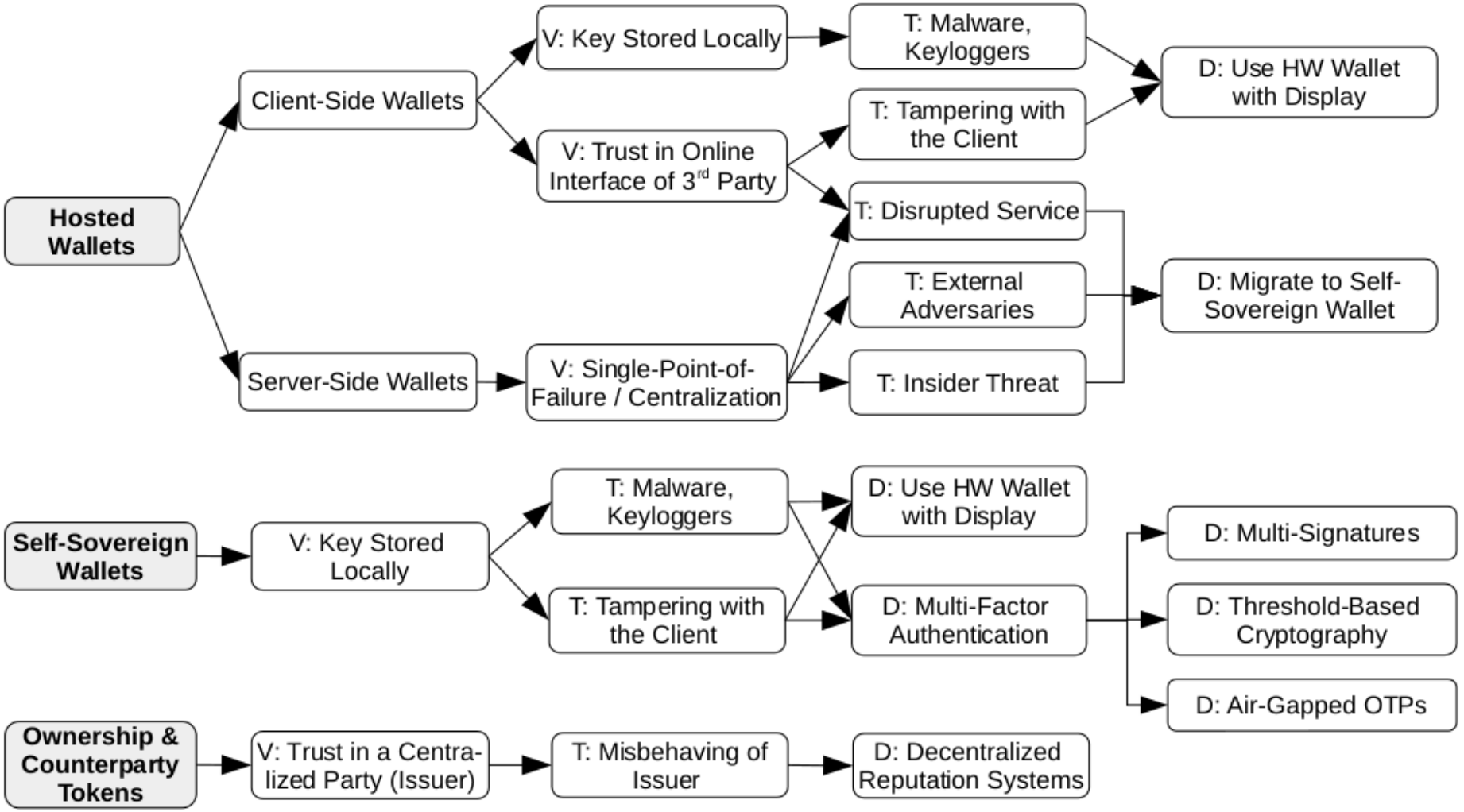
- **Full paper**
  - Homoliak, I., Venugopalan, S., Reijsbergen, D., Hum, Q., Schumi, R., & Szalachowski, P. (2020). The security reference architecture for blockchains: toward a standardized model for studying vulnerabilities, threats, and defenses. IEEE Communications Surveys & Tutorials, 23(1), 341-390.
- **Recordings from Blockchains and Decentralized Application course** from FIT@BUT: https://video1.fit.vutbr.cz/av/records-categ.php

Application Layer:
Ecosystem Applications

Application Layer - Crypto-Tokens & Wallets

- **Tokens**
  - Cryptocurrencies with native tokens
  - Counter-party tokens provide owners with rights against a third party
  - Ownership/colored tokens enable transferring of (physical/virtual) asset ownership
- **Self-Sovereign wallets**
  - Users store private keys locally and directly interact with the blockchain using the keys
  - They verify the inclusion of their transactions by SPV/light client SW
  - **Types**
    - SW wallets - store keys in PC
    - Hardware wallets - store keys in a sealed storage and expose only signing functionality
    - Smart contract wallets - functional and security customization, may utilized above
- **Hosted wallets**
  - Require a centralized party - an interface for interaction with the wallet/blockchain
  - **Types:**
    - Server-side wallets - have full control over private keys (exchanges - Coinbase, Binance, Okex, etc.)
    - Client-side wallets - keys are stored in the user's browser,

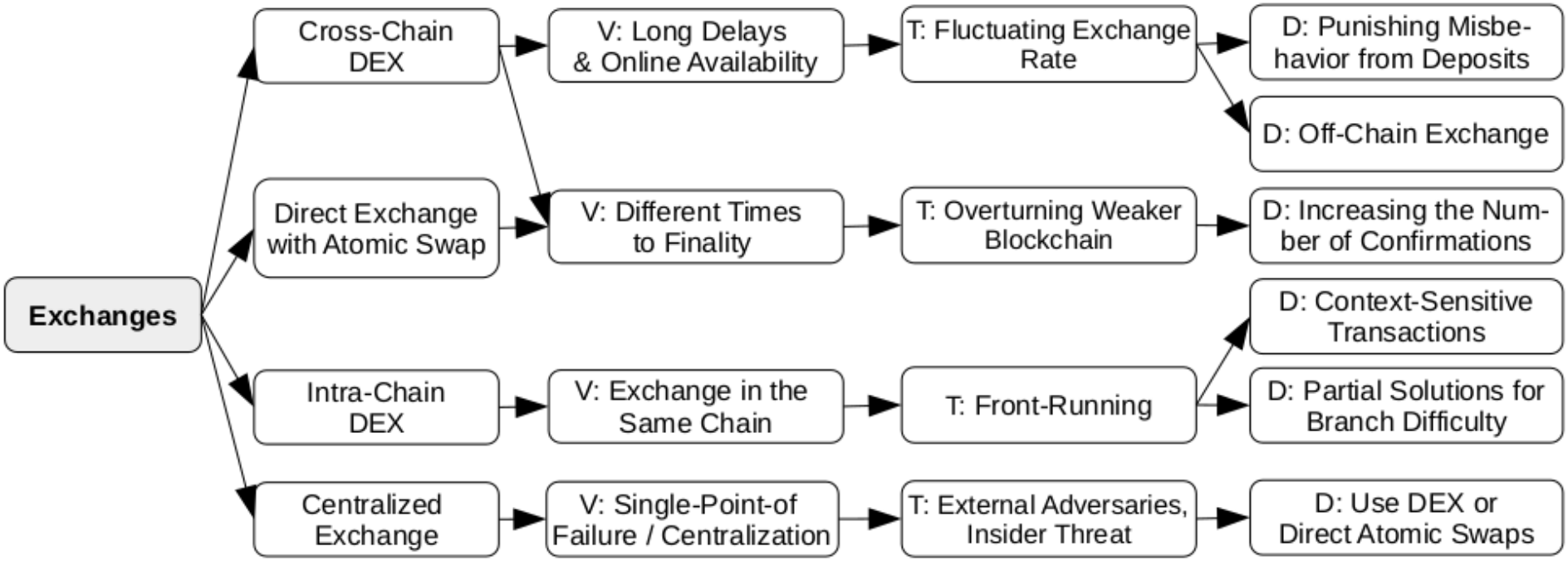
Application Layer - Crypto-Tokens & Wallets



Application Layer - Exchanges

- **Direct cross-chain exchange with atomic swap**
  - 2 parties owning crypto-tokens in two different blockchains
  - Enables conditional redemption of the funds in the 1st blockchain upon revealing the hash pre-image (i.e., secret) that redeems the funds on the 2nd blockchain
  - 2 Hashed Time-Lock Contracts (HTLC) deployed by 2 parties in 2 blockchains (requires 4 txs)
- **Cross-chain DEX** (Decentralized Exchange)
  - There might not exist a contra-party exchanging the opposite pair within swap
  - DEXes facilitate the process of matching the existing orders, act as a contra-party or intermediary
    - For obscure crypto-tokens (with no matching counter-order) DEX serves as a counter-party
  - The users match the orders, reward DEX, and afterward perform an atomic swap on their own
- **Intra-Chain DEX** (e.g., for ERC20 tokens of a single chain)
  - Users post buy&sell offers on the blockchain, and smart contracts perform matches and executions
  - Expensive due to gas fees => only trades executed on-chain, while orders and matching is off-chain
  - 0x protocol (EtherDelta), Automated market maker w deposited reserves (Euler, Bancor, Uniswap)
- **Cross-Chain Communication**
  - Generalized cross-chain exchange - interoperability of applications running on different blockchains

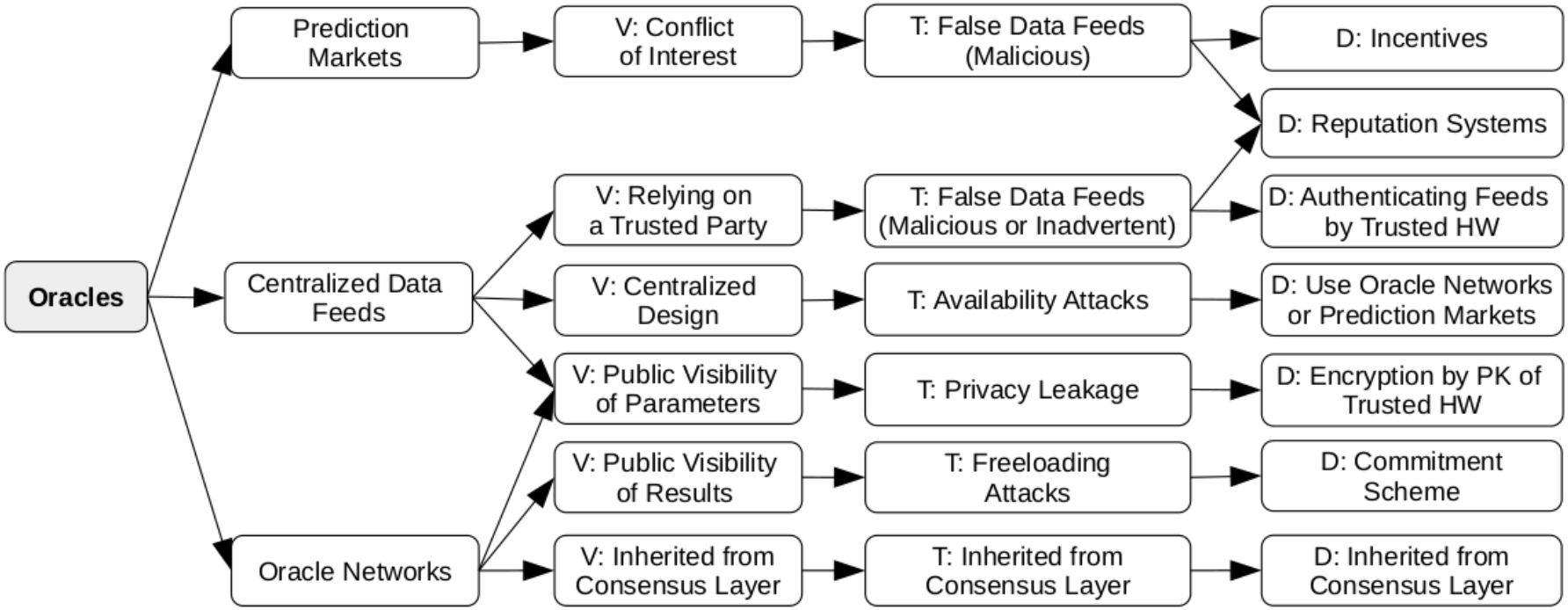
Application Layer - Exchanges



Application Layer - Oracles

- Trusted entities that provide plausible data reflecting the state of the world beyond the blockchain
- **Desired properties**
  - Authenticity: data produced by content providers agreed by the consumers of the data
  - Integrity: Content providers should guarantee the correctness of the newly created data and publicly prove their consistency with the past
  - Confidentiality: input parameters may contain confidential or private data. Therefore, an oracle should support such parameters and their handling
  - Availability: Since the execution of dependent smart contracts relies on data feeds delivered by oracles, they need to provide high availability
- **Prediction markets** - individuals accurately wager on outcomes serving as data feed (Augur, Gnosis)
  - Outcomes are provided by either a centralized reporter or a quorum of reporters
- **Centralized data feeds** - provide data from a centralized source (Oraclize, TownCrier, PDFS)
  - Auditable virtual machines, trusted computing, smart contracts
- **Oracle networks** internally run a consensus protocol for decentralized agreement on data (ChainLink, Witnet)
  - Oracle providers have reputation => the higher reputation, the higher the chance the node produces a block (Witnet) or will be selected as to the quorum by a contract (ChainLink)

Application Layer - Oracles



- **Freeloading attacks**
  - an oracle provider might copy a publicly visible value provided by other oracles w/o any effort.

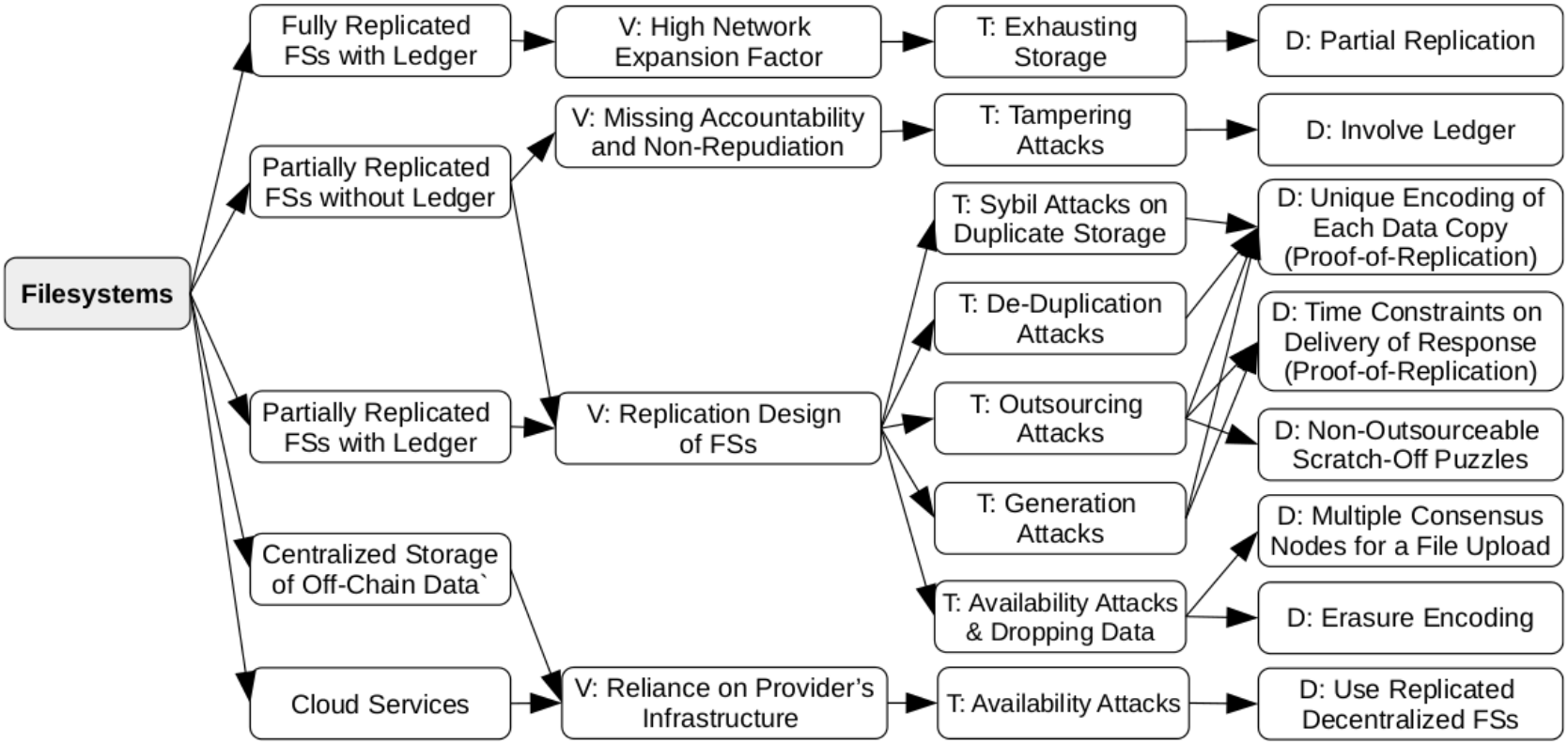
Application Layer - Filesystems

- Distributed data storage infrastructure that borrows ideas from p2p storage systems + additionally incentivizing data preservation by tokens
  - Often embed functionality into consensus layer
- **Fully replicated FS w ledger** (e.g., Namecoin or OP_RETURN)
  - A naive approach - stores the full content of data at the blockchain
  - Very high data durability (availability of data) as well as network expansion factor (storage overhead)
- **Partially replicated FS w ledger** (e.g., Permacoin, Storj, and KopperCoin)
  - Decrease the costs while preserving reasonable durability (use often erasure encoding)
- **Partially replicated FS w/o ledger** (e.g., IPFS, Swarm)
  - Distributed hash tables (DHT) - a decentralized data lookup with key:data mappings, in which the set of nodes storing the data is unambiguously determined by the key associated with the hash of data
  - IPFS does not contain any incentives and the availability of the data is dependent on its popularity
    - No blockchain but optional BitSwap ledger that logs data transfers with other nodes
- **Centralized storage of off-chain data**
  - On-chain integrity proofs and off-chain data
  - Single-point-of-failure => availability issues

Application Layer - Filesystems

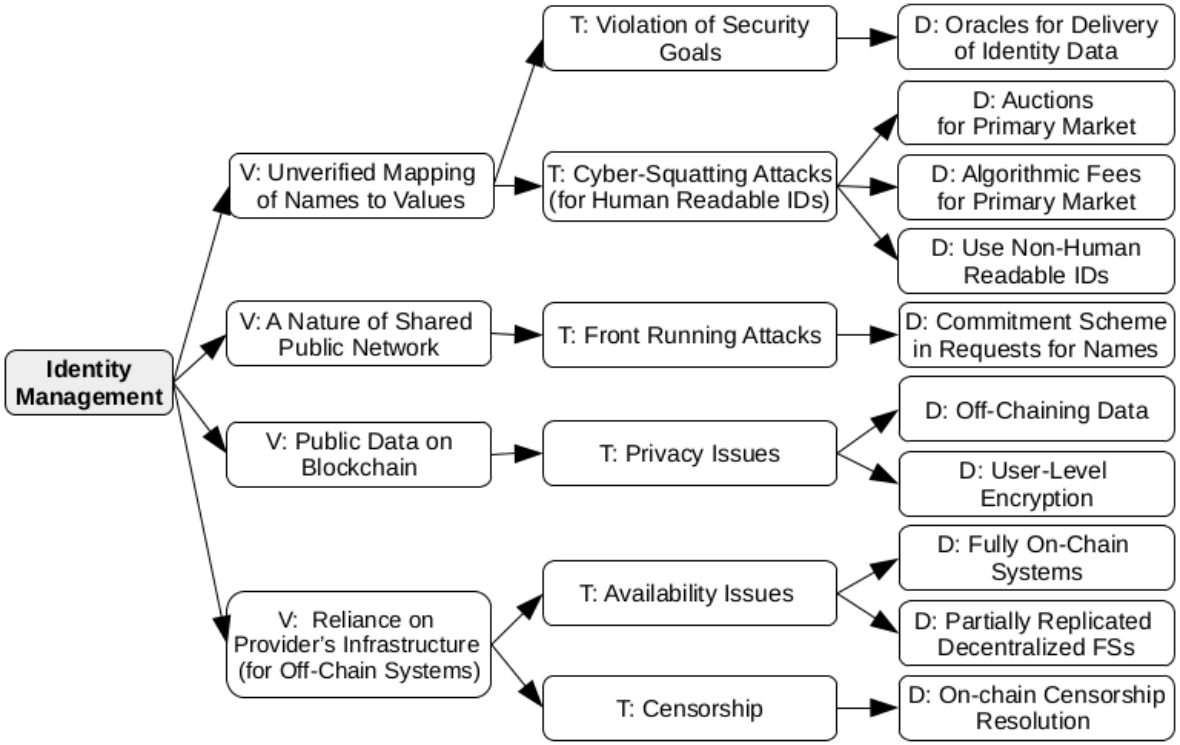
- Security Threats (recap)
  - **Sybil attack**
    - A malicious node claims the storage of multiple copies of the same data
  - **De-duplication attack**
    - More nodes may collude to claim that each of them is storing an independent copy of the data, while only one stores them
  - **Outsourcing attack**
    - A malicious consensus node claims the storage of more data than it can physically store while relying on data retrieval from outsourced data providers
  - **Generation attack**
    - A malicious node can re-generate the previously uploaded data upon request

Application Layer - Filesystems



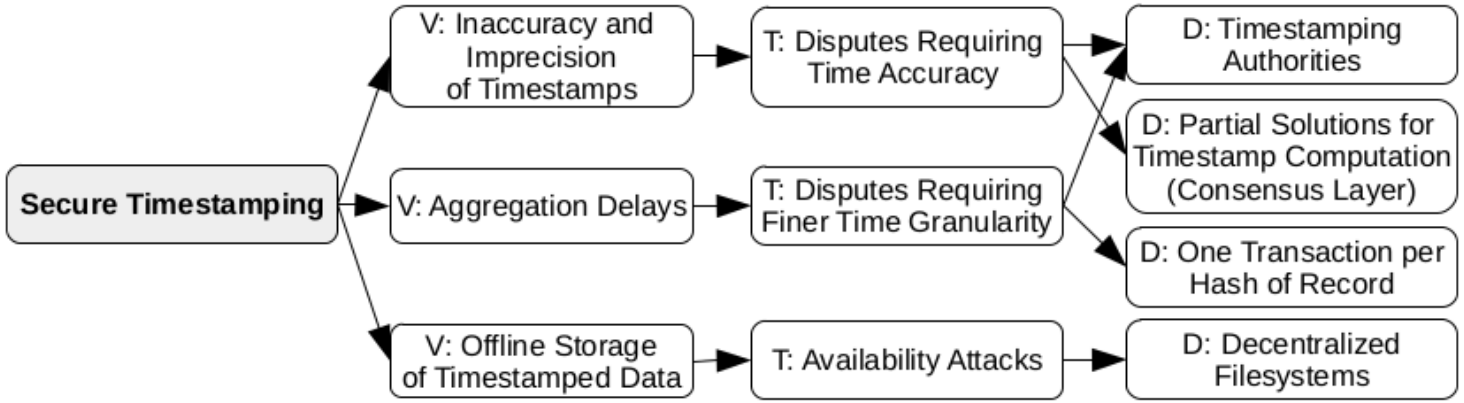
Application Layer - Identity Management

- **Binding identities of entities to their public keys**
  - Public Key Infrastructure (PKI)
- **Security goals**
  - Accurate registration - The user must be unable to register an identity that she does not own
  - Identity retention - The user must be unable to impersonate an identity already registered
  - Censorship resistance - The user must be able to register any identity that she owns.



Application Layer - Secure Timestamping

- The task is to **prove that some data existed prior to some point in time**
  - Also referred to as proof-of-existence
- **Blockchain serves as a trusted notary** that enables such proofs (since it provides immutability)
  - Blockchain “does not understand” the semantics of data => cannot verify or certify them
- Simple examples: CommitCoin, STAMPD, Bitcoin.com Notary, OriginStamp
- Aggregating data to Merkle trees - OpenTimestamps, POEX.IO

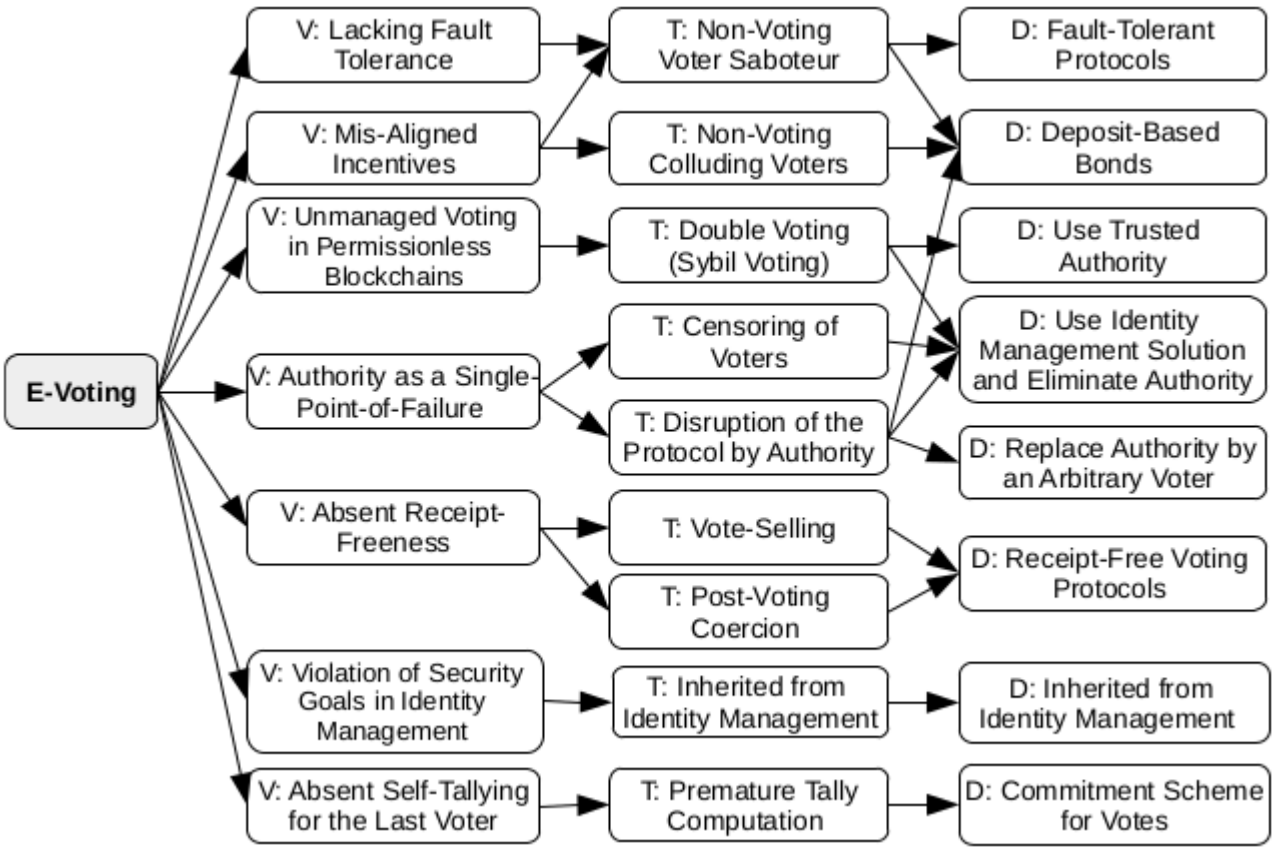


Application Layer:
Higher Level Applications

Application Layer - E-Voting

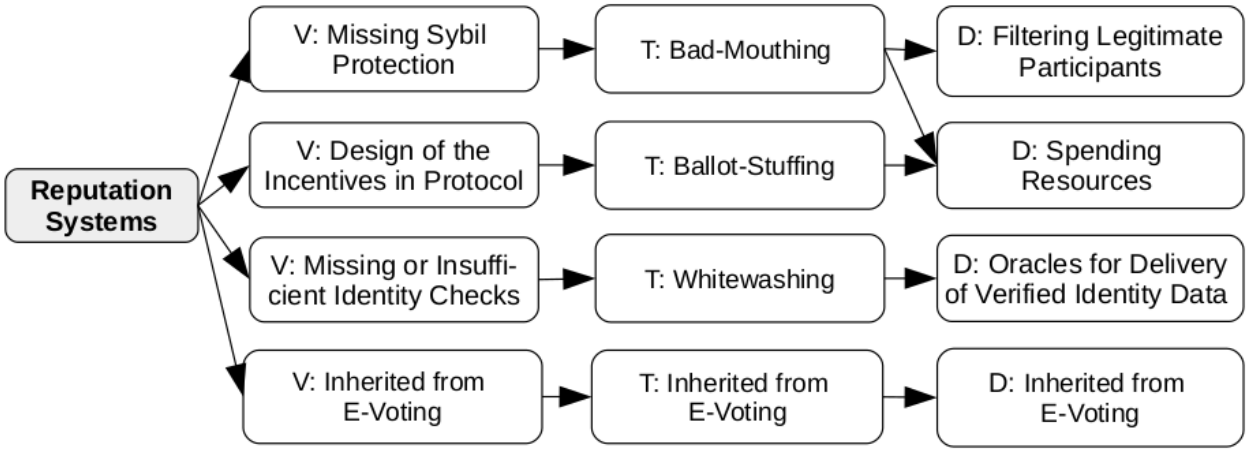
- Desirable properties in e-voting applications
  - **Perfect Ballot Secrecy**: finding partial tally before the voting finishes is possible only if all voters collude
  - **Fairness**: the tally can be computed only when all participants cast their vote
  - **Public Verifiability**: any public observer can verify the validity of all votes and final tally. This is achieved by using a public bulletin board (e.g., blockchain). A consequence is dispute-freeness, i.e., the result of the voting is indisputable
  - **Self-Tallying**: once the voting finished, anyone can compute the final tally. This property together with fairness ensures that the last voter is unable to compute the tally before casting her vote
  - **Fault Tolerance/Robustness**: a voting protocol is able to recover from a fixed number of faulty voters who do not vote or whose vote is invalid
  - **Receipt-Freeness**: a participant is unable to supply a receipt of her vote after casting the vote. The goal is to prevent vote-selling and post-election coercion
- **The main advantages of using the blockchain for e-voting**
  - Immutability, public verifiability, enforcing protocol rules by the smart contract, and higher availability
- Usually a multiparty computation (MPC) si executed by the voters
  - Voting involves an interaction among participants
  - Less robust to fault tolerance – if voters drop out midway, a recovery round has to be initiated (overhead)

Application Layer - E-Voting



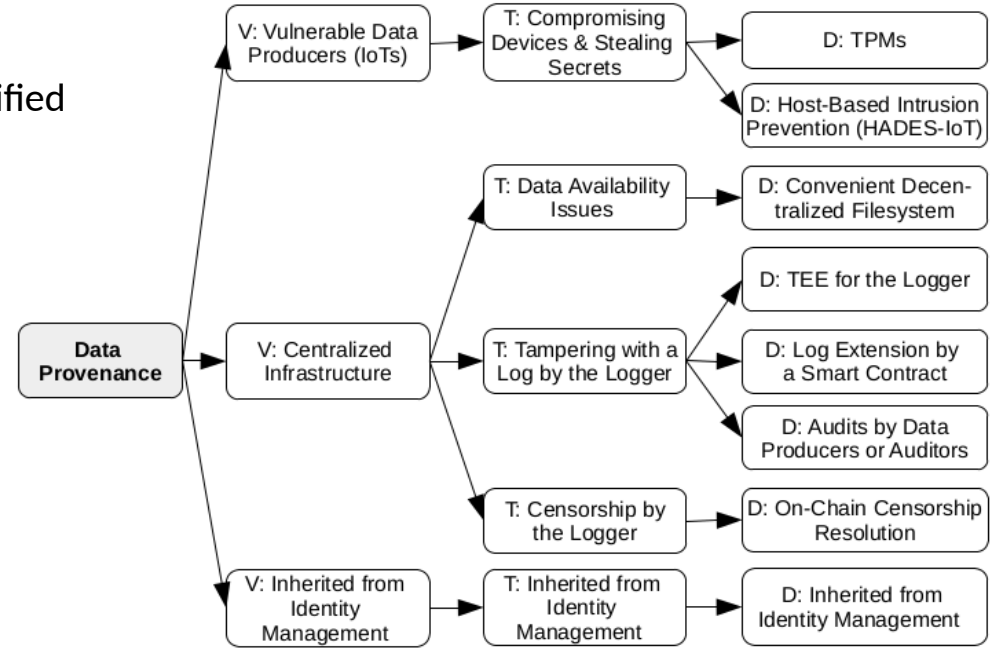
Application Layer - Reputation Systems

- The reputation is usually quantified based on the voting of parties/users with verified identities
- **1) Rating by Arbitrary Participants**
  - An arbitrary legitimate participant can rate a product/service that she has bought/consumed
- **2) Rating by Several Selected Participants**
  - A number of selected participants can vote on the authenticity of individual records (e.g., accreditation)
- **Security threats**
  - Bad-mouthing - the customer (e.g., competitor) lies about the product or service
  - Ballot-stuffing - the service provider might increase her reputation by herself
  - Whitewashing - the service provider creates a new service w a neutral reputation, which is unlinkable to her previous service (w negative reputation)



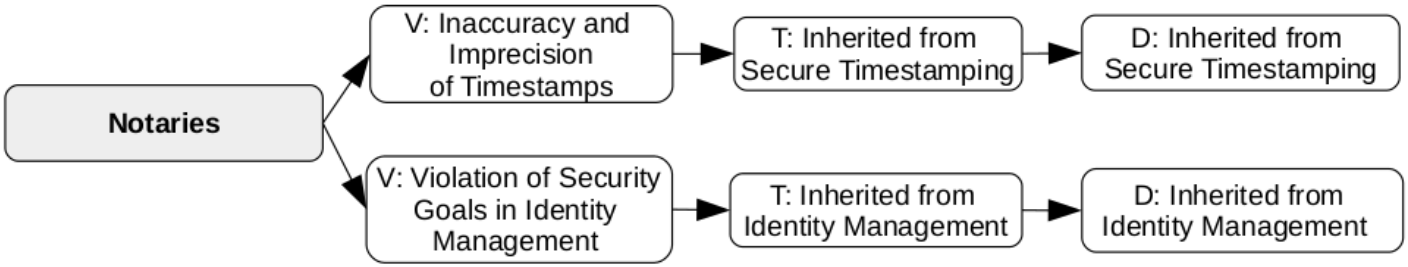
Application Layer - Data Provenance

- **Represents the ownership history of an arbitrary object**
- In the cyber-world, objects are represented by mutable data
  - History must account also for the modifications
  - Use of blockchains has the potential to resolve various issues related to intellectual property, authorship, the validity of certificates or other issued documents
- **E.g. supply-chain management**
  - The goal is to resolve potential issues in the traceability of goods and provenance of associated data
- **E.g., issuance of educational certificates**
  - Issued by institutions whose identities are verified



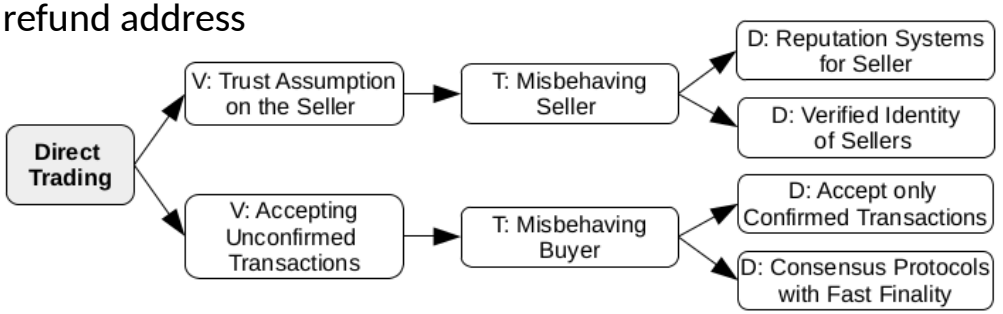
Application Layer - Notaries

- In contrast to timestamping, the role of the notary is not only to prove the existence of documents at certain points in time but also to **vet and certify documents**
- Assume known **verified identities** of involved parties
- **The involved parties may decide whether to store vetted documents**
  - In a database of a notary service provider (e.g., PADVA) or keeping it privately at the client-side (e.g., Blockusign)
- E.g., ADVOCATE is an approach for notarization of agreements about personal data processing in IoT between owners of IoT devices and data processing services – both must co-sign an agreement
- E.g., SilentNotary is a smart contract-based system for self-certifying of files produced by registered users
- E.g., PADVA is a TLS notary service realized as a smart contract-based two-party agreement
  - Notaries obligated to periodically check the validity of PKs in a specified set of certificates



Application Layer - Direct Trading

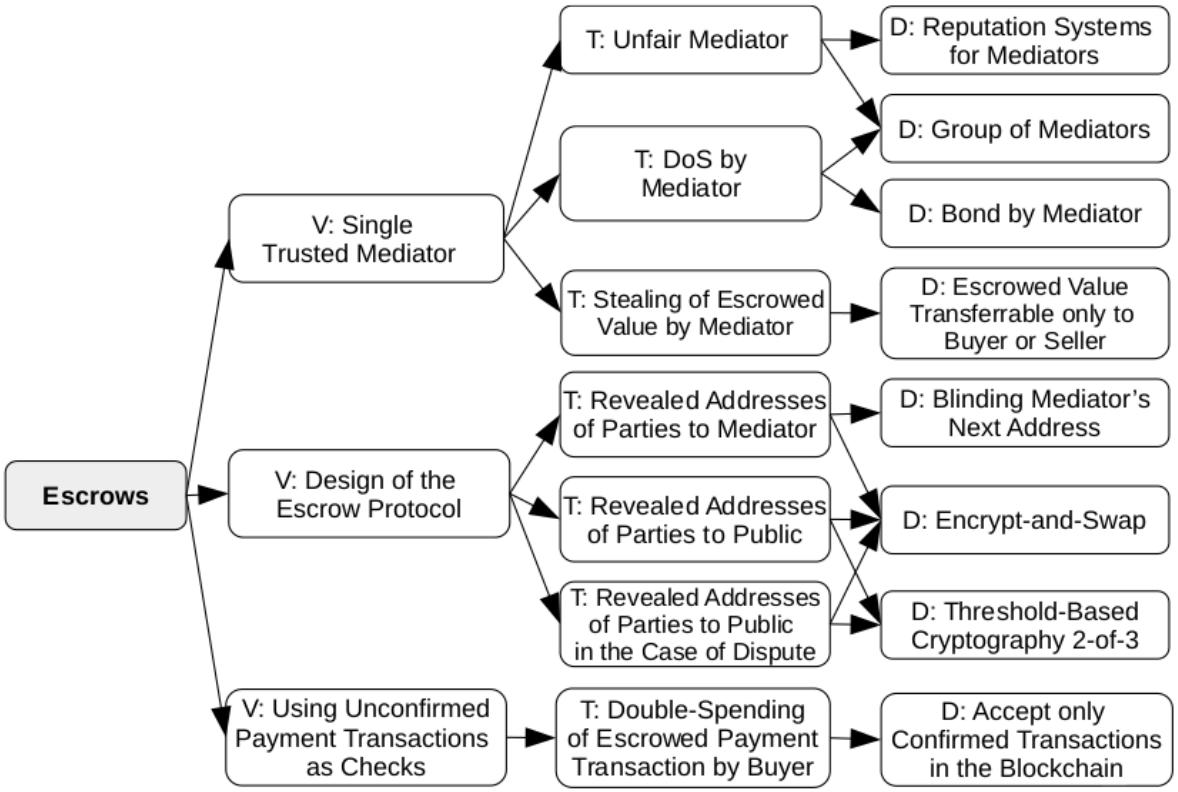
- **Owners want to exchange crypto-tokens they hold for goods outside of the cryptocurrency blockchain**
- **The buyer/seller dilemma -** "Should the buyer trust the seller and pay her before receiving goods or should the seller trust the buyer and ship the goods before receiving the payment?"
- The assumption of a **trusted seller with a verified identity**
- E.g., in BIP-70, the buyer first verifies the authenticity of the seller using its X.509 certificate and then issues a payment transaction
- **Misbehaving seller**
  - The buyer might ask the seller to interrupt the request and get a refund but the seller may misbehave, and thus risk a reputation loss
- **Misbehaving buyer**
  - Double spending
  - Silkroad trader attack - in BIP-70 a malicious buyer might replace her refund address and then ask the seller for a refund. After a refund, the buyer might plausibly deny receipt of a refund (and ask for a refund again) due to missing authentication on the refund address



Application Layer - Escrows

- **The same problem as direct trading but in contrast to it, escrows do not assume a trusted seller**
  - Instead, escrows outsource the trust into the third party, referred to as a **mediator**
  - The mediator might actively participate in the escrow protocol or participate only in the case of a dispute
  - Parties with verified identities and reputation systems to assess these parties and mediators
- **Single mediator protocols**
  - 2-of-3 multi-signatures for splitting the control, threshold-based signatures for improving privacy, and protocols leveraging homomorphic properties of EC to achieve privacy and non-interactiveness
  - Multi-signatures with bonds deposited by a mediator to avoid DoS by the mediator
  - E.g., OpenBazaar is a distributed marketplace that uses smart contract-based escrows with 2-of-3 multi-sig, where the mediator is agreed by the buyer and seller
- **Group-based mediator protocols**
  - Disputes are resolved by a majority vote
  - DoS attack is thwarted as long as the majority of mediators is willing to finish the the protocol

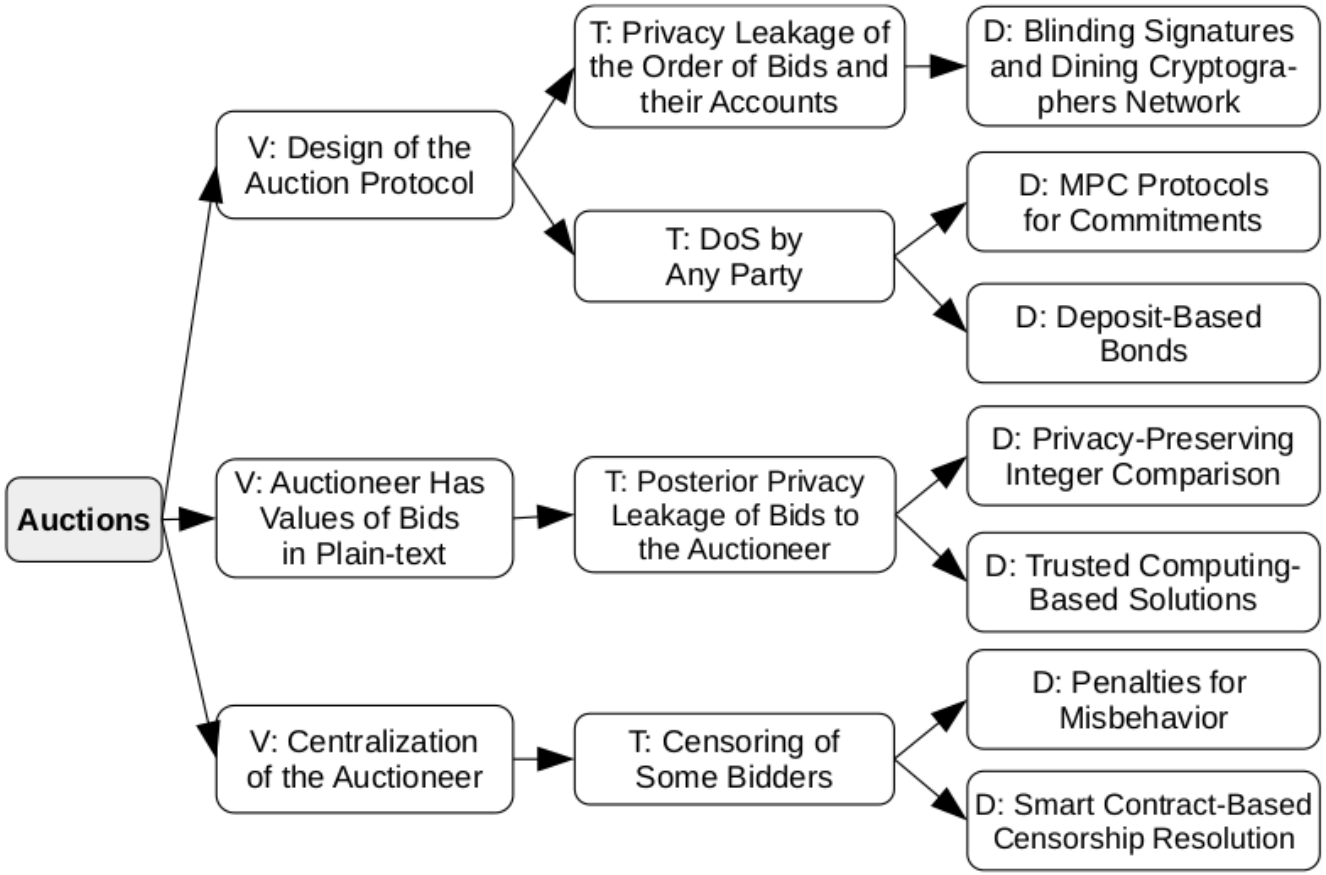
Application Layer - Escrows



Application Layer - Auctions

- **Sellers promote the sale of their goods through blockchain while buyers place bids for them**
- **Desired properties of auctions**
  - **Privacy of bids** ensures that values of particular bids are not revealed to anybody before committing to them
  - **Posterior privacy** ensures that all bids remain private after the auction ends
  - **Publicly verifiable correctness** enables anybody to verify the results of the auction through the blockchain
  - **Resistance against DoS** ensures that no bidder or auctioneer can prematurely abort a protocol w/o being penalized
- Privacy often achieved by 1) homomorphic commitments of sealed bids, 2) zk-SNARKs and its off-chain computation requiring only a single on-chain proof verification, and 3) TEE

Application Layer - Auctions



Thank You for Your Attention !

Questions?

Recordings from Blockchains and Decentralized Application course from FIT@BUT: https://video1.fit.vutbr.cz/av/records-categ.php