MIDDLEWARE SYSTEMS
RESEARCH GROUP
MSRG.ORG

Please do not forget to provide your feedback via the course evaluation.

Deconstructing Blockchains: Concepts, Systems and Applications

PLATFORMS AND APPLICATIONS

Blockchain Platforms

ETHEREUM

HYPERLEDGER

ETHEREUM

Managing entity: Ethereum Foundation

- Major players: Deloitte, Toyota, Microsoft, …

Enable decentralized applications (Dapps) *et al.*

Open-source, flexible, general platform

- Permisionless (public) ledger, proof-of-work-based(alternative mechanisms are work in progress)
- Cryptocurrency: 1 Ether = 1e18 Wei (~150 USD, 2020/4)
- Smart contracts: Solidity, Remix (Web IDE), Truffle (Dev./Test), *Viper* (programming language to build Dapps)
- Ethereum Virtual Machine (EVM)

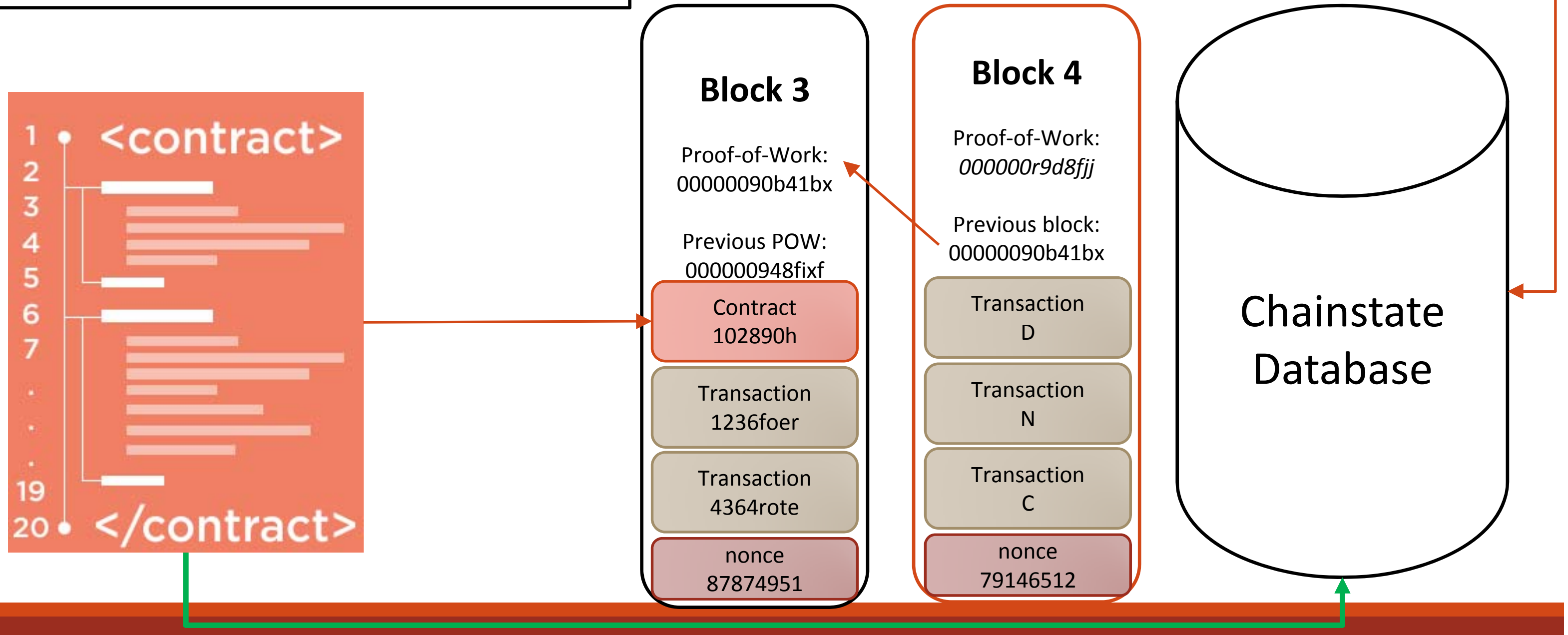
Smart Contracts

- *Contracts* are programs, compiled into bytecode to execute on EVMs
- Contracts have internal storage

- Contracts execute when triggered by a transaction (or by another contract)
- Execution time is limited by *gas*

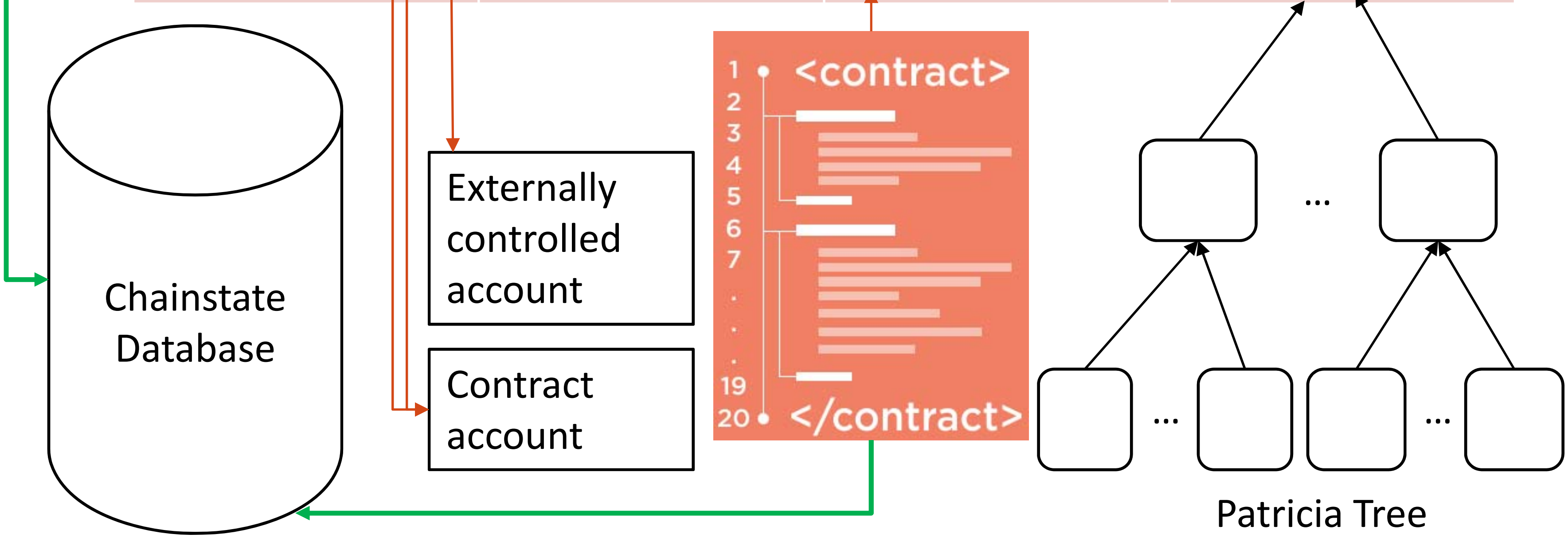
Example: Land registry

Wallet ID	Held Titles
99823428347	34356,324324
98217981623	677343,4444
90987344755	994,38842,439

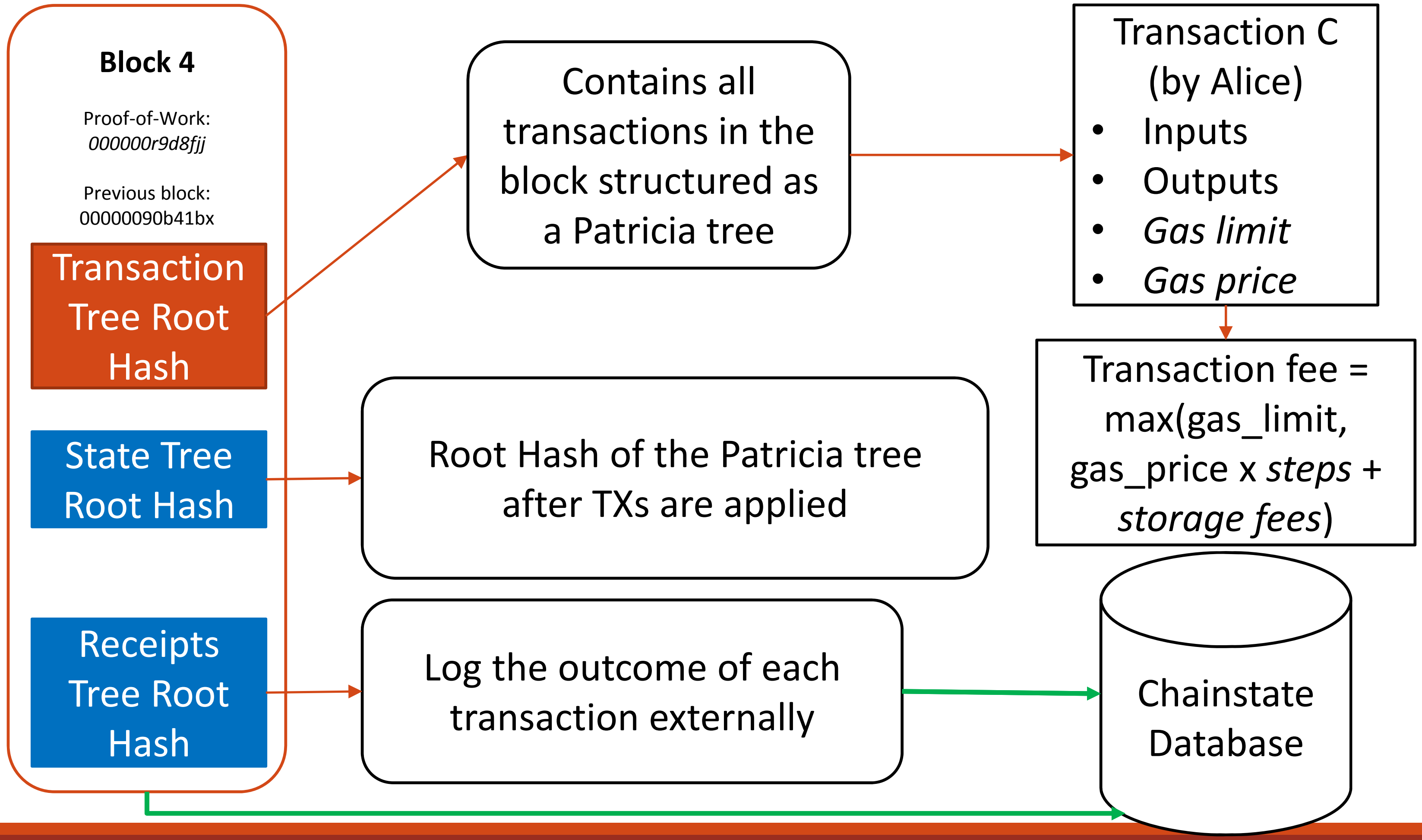


Account State (“World State”)

Wallet ID	Balance	Code Hash	Internal State
99823428347	45.12	-	99554HGJ
98217981623	1123.332	9ERU12T4	3453ADFG
90987344755	9.3444	0490CNDJ	132GJR4



Execution



Block 4

Proof-of-Work:
000000r9d8fjj

Previous block:
00000090b41bx

Transaction Tree Root Hash

State Tree Root Hash

Receipts Tree Root Hash

Contains all transactions in the block structured as a Patricia tree

Transaction C (by Alice)

- Inputs
- Outputs
- *Gas limit*
- *Gas price*

Transaction fee = max(gas_limit, gas_price x *steps* + *storage fees*)

Root Hash of the Patricia tree after TXs are applied

Log the outcome of each transaction externally

Chainstate Database

HYPERLEDGER

Managing entity: Hyperledger Consortium

- Major players: IBM, NEC, Intel, R3, …

Enterprise blockchains

- Permissioned ledger (private and consortium networks)
- Smart contracts in general purpose language(s)
- Open-source, configurable, pluggable consensus
- World state on CouchDB, LevelDB, *et al*.

Hyperledger is a family of projects

- Fabric: PBFT Consensus *et al.*
- Sawtooth: Proof-of-elapsed time (using Intel SGX)
- Composer: Smart contract language and development tool
- Cello: Blockchain-as-a-Service framework

HYPERLEDGER

Key **differentiators**

- Assumes a more trusted environment than Bitcoin/Ethereum
- Requires authentication to partake in business network
- Dozens of peers that manage distributed ledger (not 1000s)
- No cryptocurrency, no tokens (could be build on top)
- No proof-of-work-based consensus (traditional consensus)
- No mining, no intrinsic inventive mechanisms

Intended **use cases**

- Trade finance (tracking financial transactions and goods)
- Supply chains, logistics (tracking goods, assets, etc.)
- Cross-border trade
- Inter governmental information exchange
- Health-care networks (provider, insurer, laboratory, end-user)

Chaincode Example

Digital Rights Management for Music (DRM)

The DRM chaincode has a function ‘play()’ which:

- Reads an artwork
- Reads the royalty related to that artwork
- Increments a count to track royalty payments
- Writes the new count

Chaincode Example

Digital Rights Management for Music (DRM)

DRM chaincode function ‘play()’:

```
async play(ctx, artWorkId) {
    const metadata = await ctx.stub.getState(artWorkId);
    let royaltyManagementAsset = await
    ctx.stub.getState(metadata.royaltyManagementId);
    royaltyManagementAsset.incrementPlayCount();
    await ctx.stub.putState(metadata.royaltyManagementId,
          royaltyManagementBuffer);
}
```

Chaincode Example

Digital Rights Management for Music (DRM)

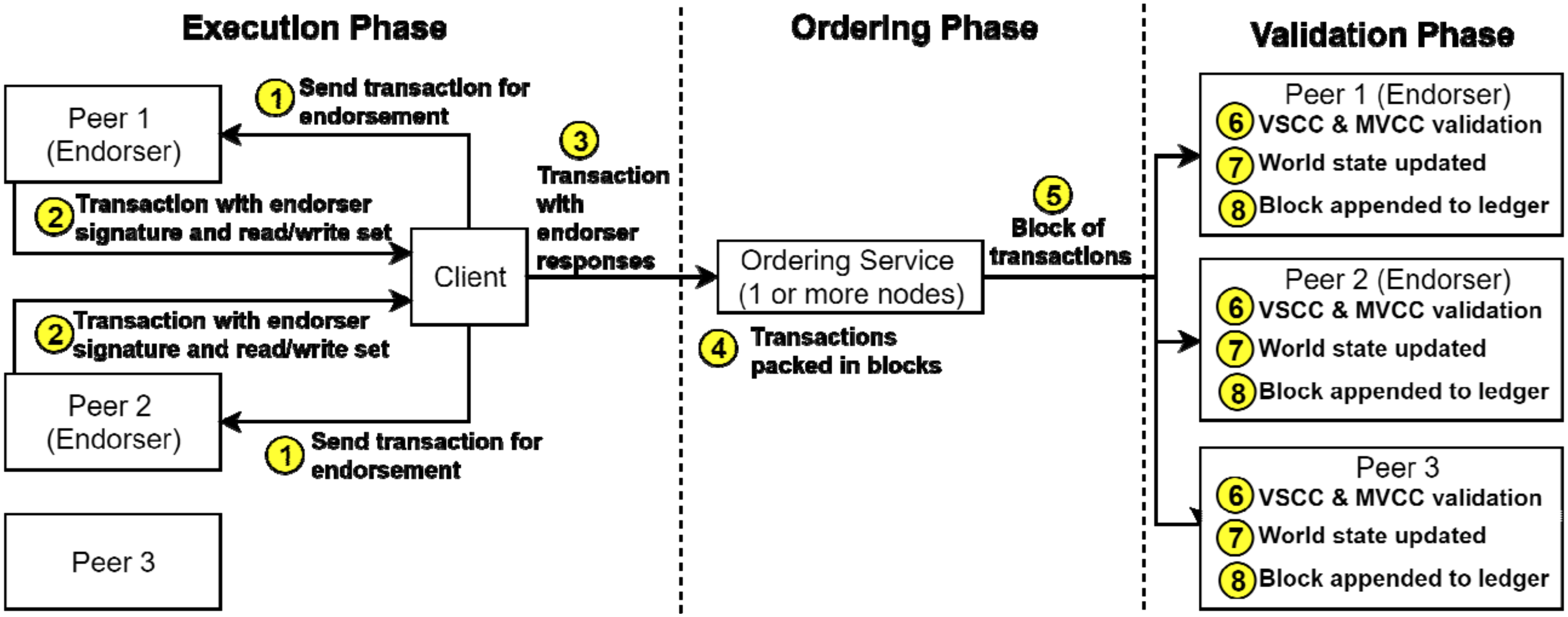
play(context, 04672033) generates this **read-write set**:

```
{"namespace":"drm","rwset":{
"reads":[
{"key":"04672033","version":{"block_num":"5","tx_num":"8"}},
{"key":"554266330","version":{"block_num":"5","tx_num":"8"}}],
"range_queries_info":[],
"writes":[
{"key":"04672033","is_delete":false,"value":"{\"docType\":\"royaltyMa
nagement\",\"perPlayRoyalty\":0.0031611628296938066,\"allRightHold
er\":[{\"ipiName\":\"44350234880\",\"share\":0.7245692636304071},{
\"ipiName\":\"28085045037\",\"share\":0.10356757729154009},{\"ipiN
ame\":\"88061101255\",\"share\":0.17186315907805283}],\"playCount
\":1}"}],
"metadata_writes":[]},"collection_hashed_rwset":[]}
```

Read set with key and version

Write set with key and value

Transaction Flow in Fabric


Execution Phase
Ordering Phase
Validation Phase
Peer 1 (Endorser)
1 Send transaction for endorsement
2 Transaction with endorser signature and read/write set
Client
3 Transaction with endorser responses
2 Transaction with endorser signature and read/write set
Peer 2 (Endorser)
1 Send transaction for endorsement
Peer 3
Ordering Service (1 or more nodes)
4 Transactions packed in blocks
5 Block of transactions
Peer 1 (Endorser)
6 VSCC & MVCC validation
7 World state updated
8 Block appended to ledger
Peer 2 (Endorser)
6 VSCC & MVCC validation
7 World state updated
8 Block appended to ledger
Peer 3
6 VSCC & MVCC validation
7 World state updated
8 Block appended to ledger

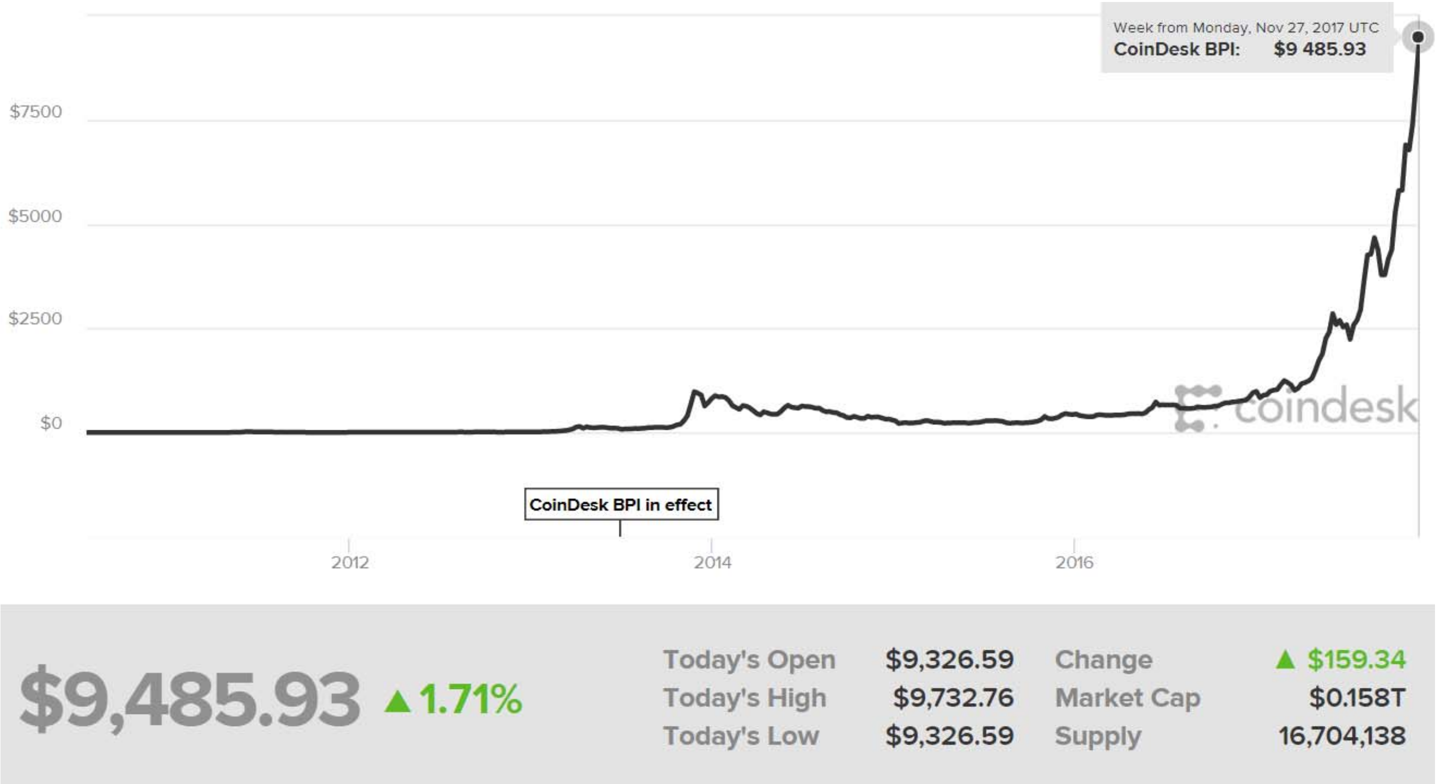
E-O-V Model

Blockchain Applications

1.0, 2.0, 3.0 GENERATIONS

IMPACT

Blockchain 1.0: Currency



$9,485.93 ▲1.71%

Today's Open	$9,326.59	Change	▲ $159.34
Today's High	$9,732.76	Market Cap	$0.158T
Today's Low	$9,326.59	Supply	16,704,138

Bitcoin cryptocurrency (2008)

Blockchain 2.0: Decentralized Apps (DApps)

ETHEREUM

DApps are applications built on blockchain platforms using smart contracts (e.g. Ethereum)

EtherTweet

Decentralized Microblogging

Token Distribution

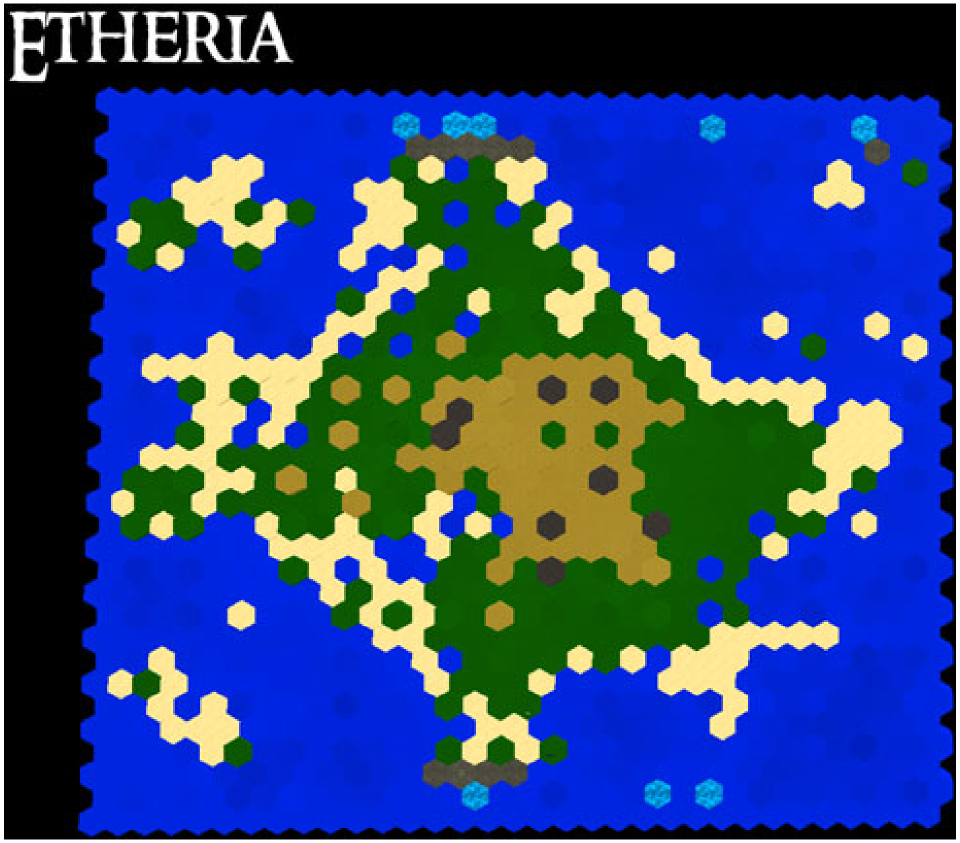
Crowdfunding

alice

Charity donation payment

Blockchain 2.0: Decentralized Apps (DApps)

Forecast market (e.g. betting, insurance)



Decentralized virtual world

Blockchain 3.0: Pervasive Apps

Applications involve entire industries, **public sector**, and IoT.

everledger

Diamonds Provenance

FACTOM

Land Registry in Honduras

BlockchainHealth

Electronic Health Records

VOTEWATCHER

Transparent Voting System

Why Study Blockchains?

Drivers

- Avoid middlemen
- Provide transparency, audit trail
- Eliminate friction during conflicts (non-repudiation)

Research challenges for 1.0:

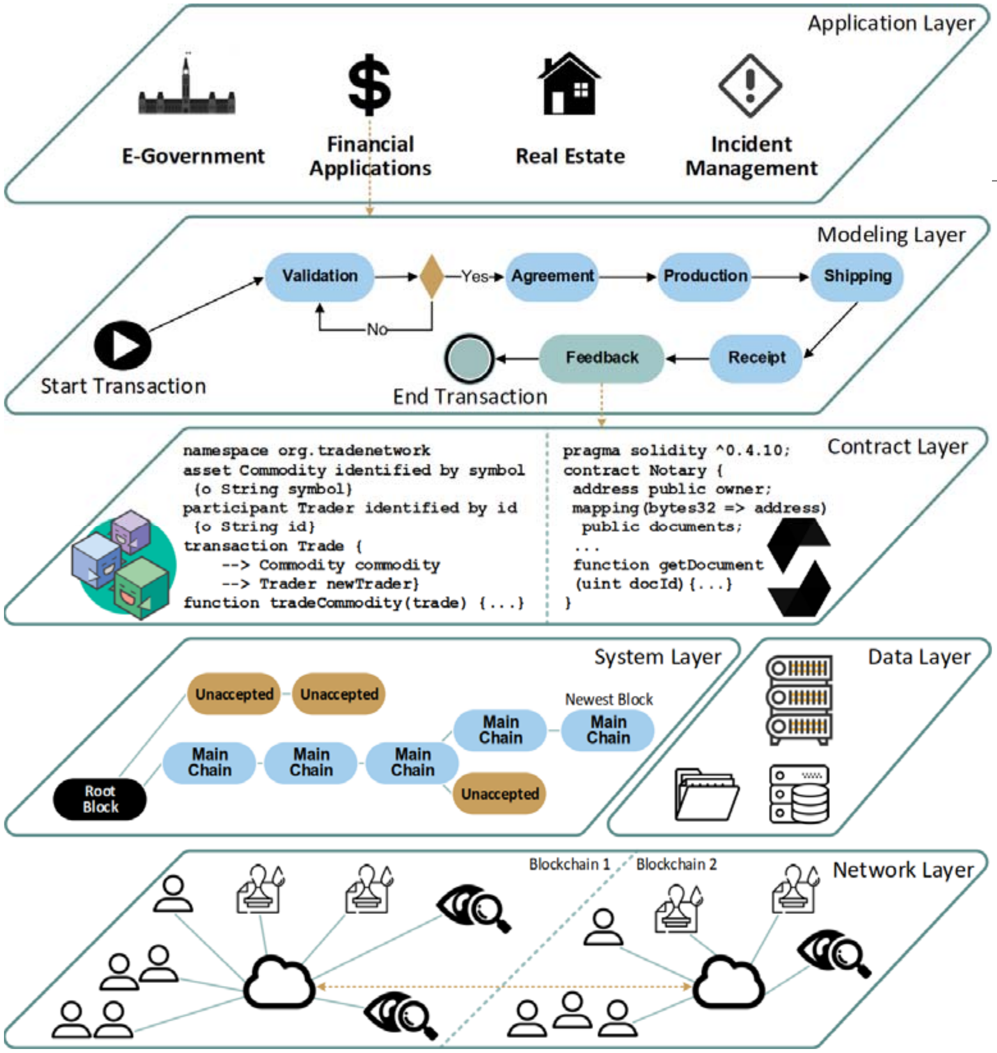
- Identify theoretical **security flaws**
- **Sustainability** of legacy systems

Research challenges for 2.0:

- **Verify** smart contracts
- Create generic middleware **services**

Research challenges for 3.0:

- Develop **scalable and fast** systems
- Guarantee data **privacy**
- Verify **correctness** of data entry points (CPS interface barrier)



```
namespace org.tradenetwork
asset Commodity identified by symbol
 {o String symbol}
participant Trader identified by id
 {o String id}
transaction Trade {
    --> Commodity commodity
    --> Trader newTrader}
function tradeCommodity(trade) {...}
```

```
pragma solidity ^0.4.10;
contract Notary {
 address public owner;
 mapping(bytes32 => address)
  public documents;
 ...
 function getDocument
 (uint docId){...}
}
```

Conclusions

- Blockchains provide *decentralized storage and code execution,* and can be used to combat fraud, avoid redundancy, and provide transparency.
- Blockchains rely on *cryptography* and massive replication using a robust consensus mechanism.
- Blockchains are useful for a wide variety of applications, ranging from cryptocurrency (*1.0*) to health (*3.0*).
- Research directions exist *across the six layers* for all kinds of applications (from 1.0 to 3.0).

The End!

Please do not forget to provide your feedback via the course evaluation.