Blockchain Business Development

Smart Contract Development

CBS, DIKU

Copenhagen, Denmark

30/9/2020

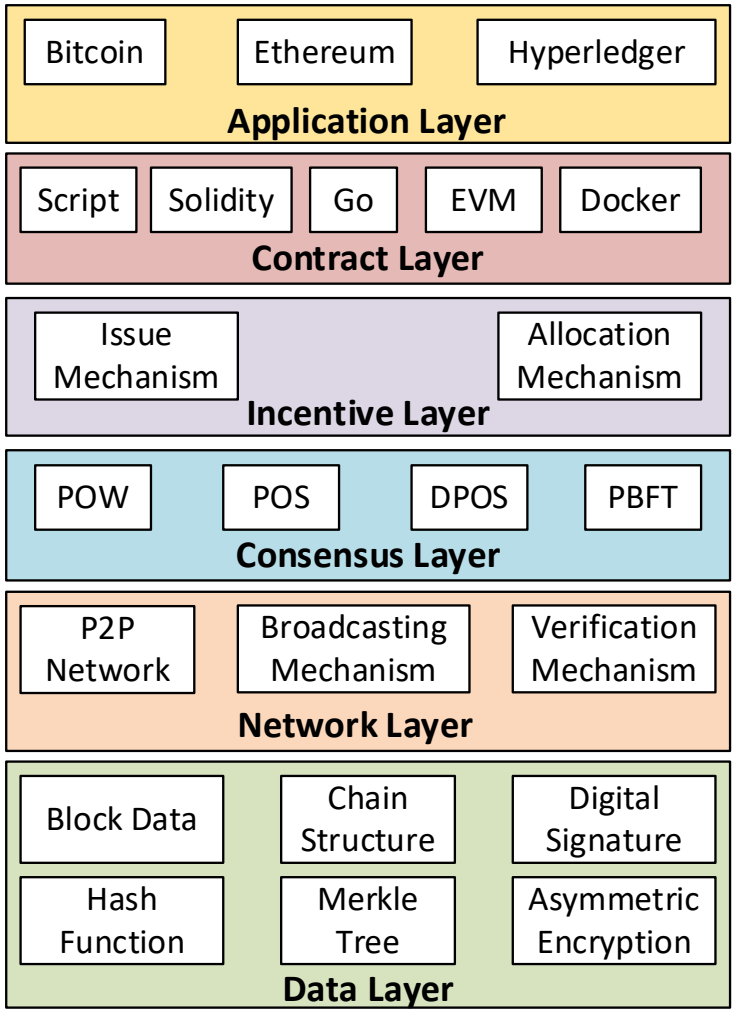
Boris Düdder

UNIVERSITY OF COPENHAGEN

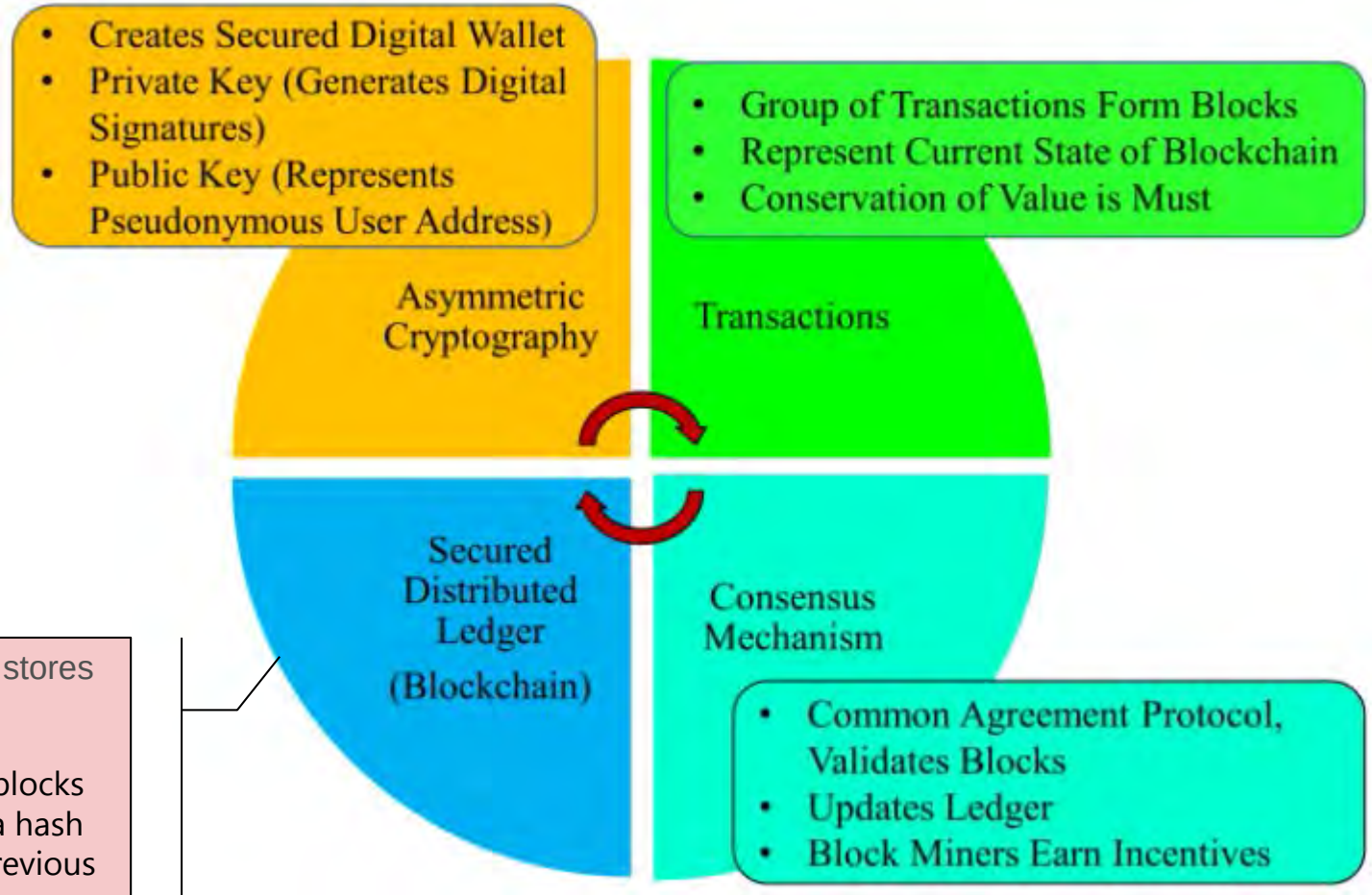
Comparison of Popular Blockchain Platforms

	Ethereum	Hyperledger fabric	EOS	Corda	MultiChain	OpenChain
Industry focus	Cross-Industry	Cross-Industry	Cross-Industry	Financial Services	Financial Services	Digital Asset Management
Governance	Ethereum developers	Linux foundation	ECAF	R3	CoinPrism	Linux foundation
Mode of operation	Public, private	Permissioned, private	Public	Permissioned	Permissioned, private	Permissioned
Consensus	POW	PBFT	DPOS	Pluggable Framework	POW	Partionned Consensus
Supported Language	Solidity	Go, Java	C++/Rust/ Python/Solidity	Kotlin, Java	C, C++, Python, JavaScript	JavaScript
Currency	Ether	None	EOS token	None	None	None
Transaction fee	Ether	None	None	None	None	None
Privacy	No	Yes	No	Yes	Yes	Yes

Blockchain Architecture: Six Layers



Core Components of Blockchain



- A data structure stores transaction data
- Transactions are assembled into blocks
- Each block has a hash pointer to the previous block

Puthal D, Malik N, Mohanty S P, et al. Everything you wanted to know about the blockchain: Its promise, components, processes, and problems[J]. IEEE Consumer Electronics Magazine, 2018, 7(4): 6-14.

Legal vs. Smart Contract

- A smart contract is more like a vending machine (Automated processing, trust reduction, unambiguous…)
- Follow predetermined rules

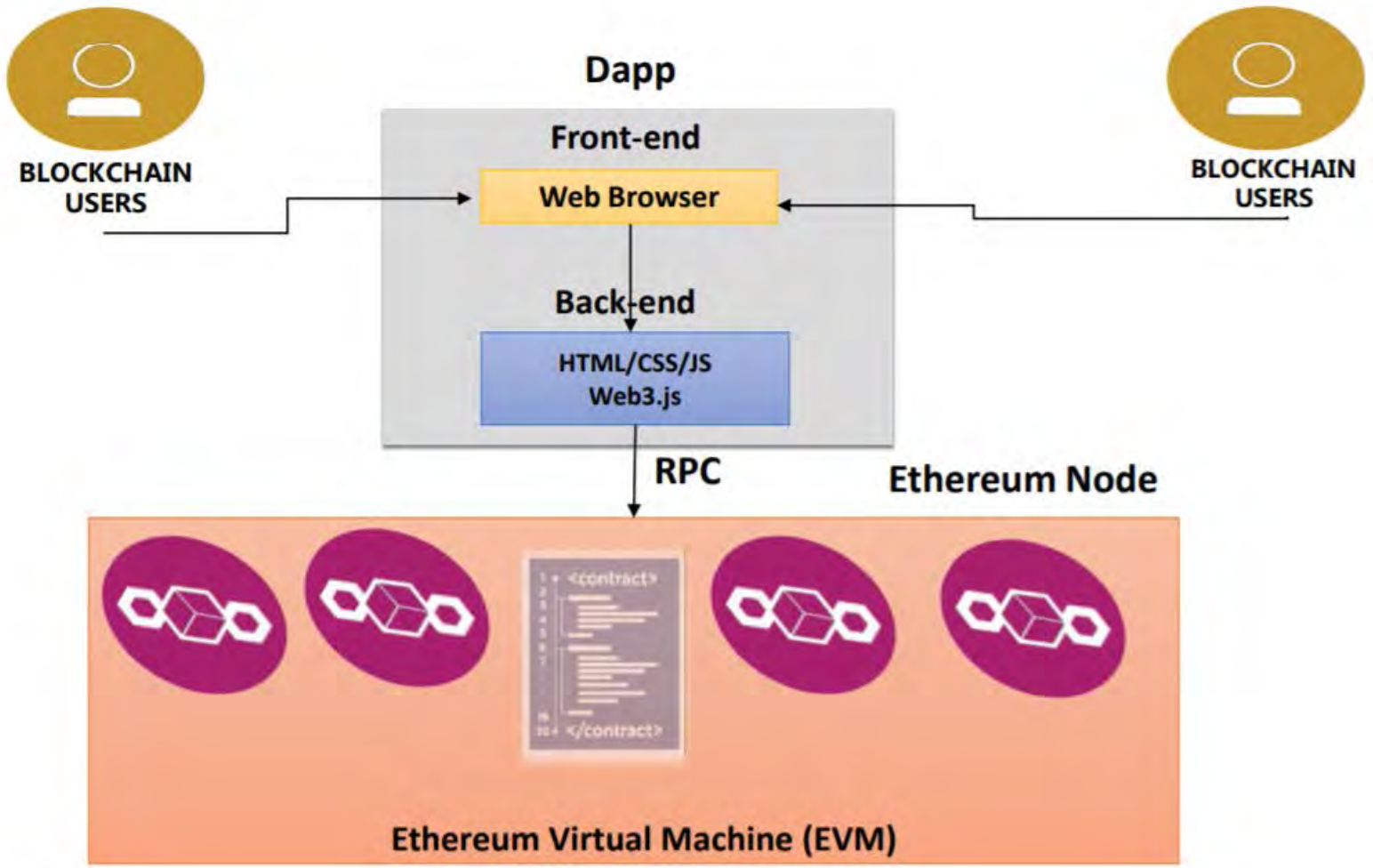
Legal contracts	**Smart contracts**
Good at subjective (ie. requiring human judgement) claims	Good at objective (ie. mathematically evaluable) claims
High cost	Low cost
May require long legal process	Fast and automated
Relies on penalties	Relies on collateral/security deposits
Jurisdiction-bound	Potentially international (“a-legal”)

Development Phases of Blockchain Applications

- Analysis phase: collect and analyze the requirements, identify the entity/parties involved (roles and relationship)
- Design phase: model the entity attributes as state variables and interactions as functions, capture the constraints and dependencies
- Implementation phase: implement the smart contract and system business logic (decentralized applications, DApps)

Karamitsos I, Papadaki M, Al Barghuthi N B. Design of the blockchain smart contract: A use case for real estate[J]. Journal of Information Security, 2018, 9(3): 177-190.

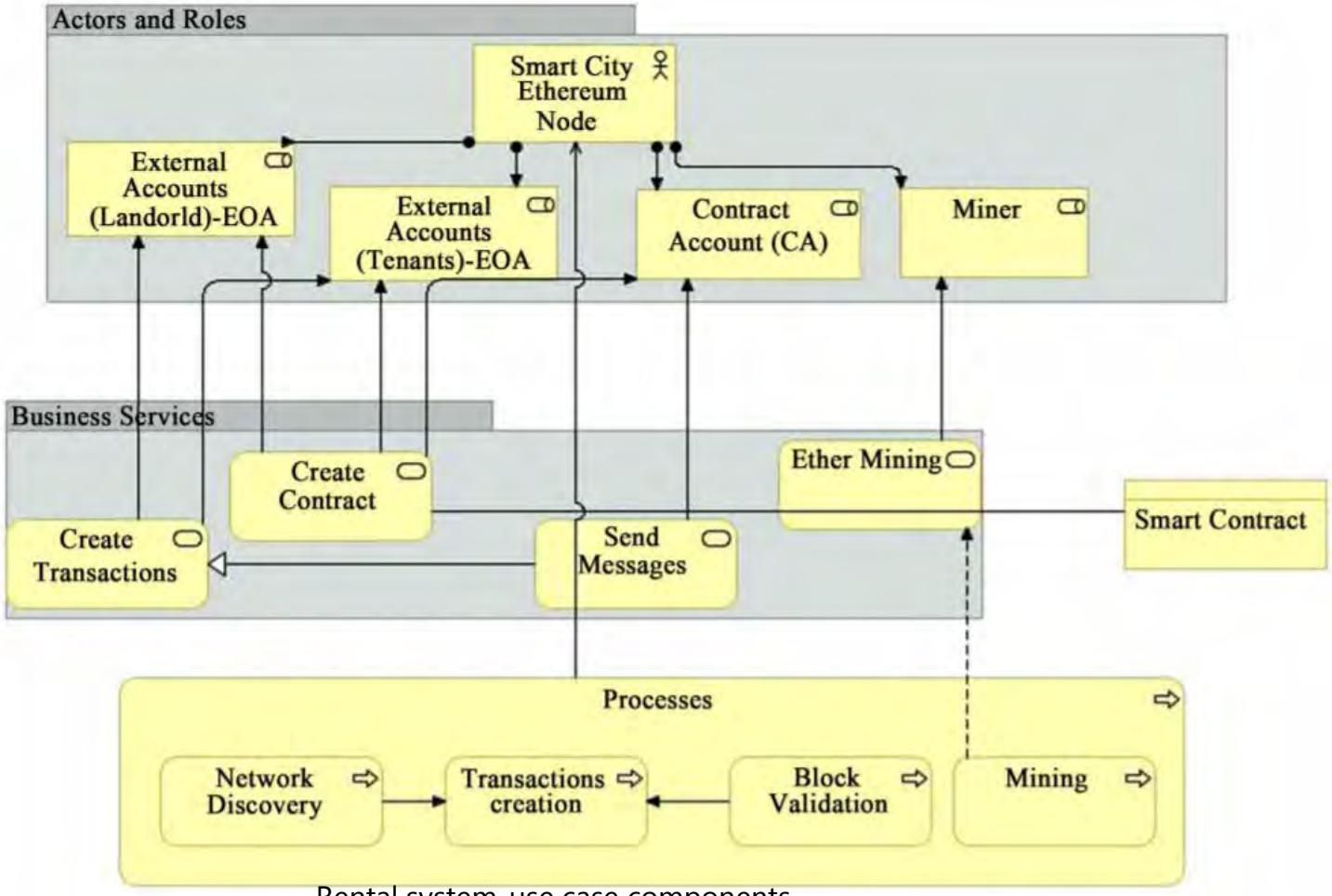
DApp Structure in Ethereum Blockchain



Karamitsos I, Papadaki M, Al Barghuthi N B. Design of the blockchain smart contract: A use case for real estate[J]. Journal of Information Security, 2018, 9(3): 177-190.

Design Methodology for Smart Contract

- Ethereum node setup
  - Actors and roles
  - Business service functions
  - Processes

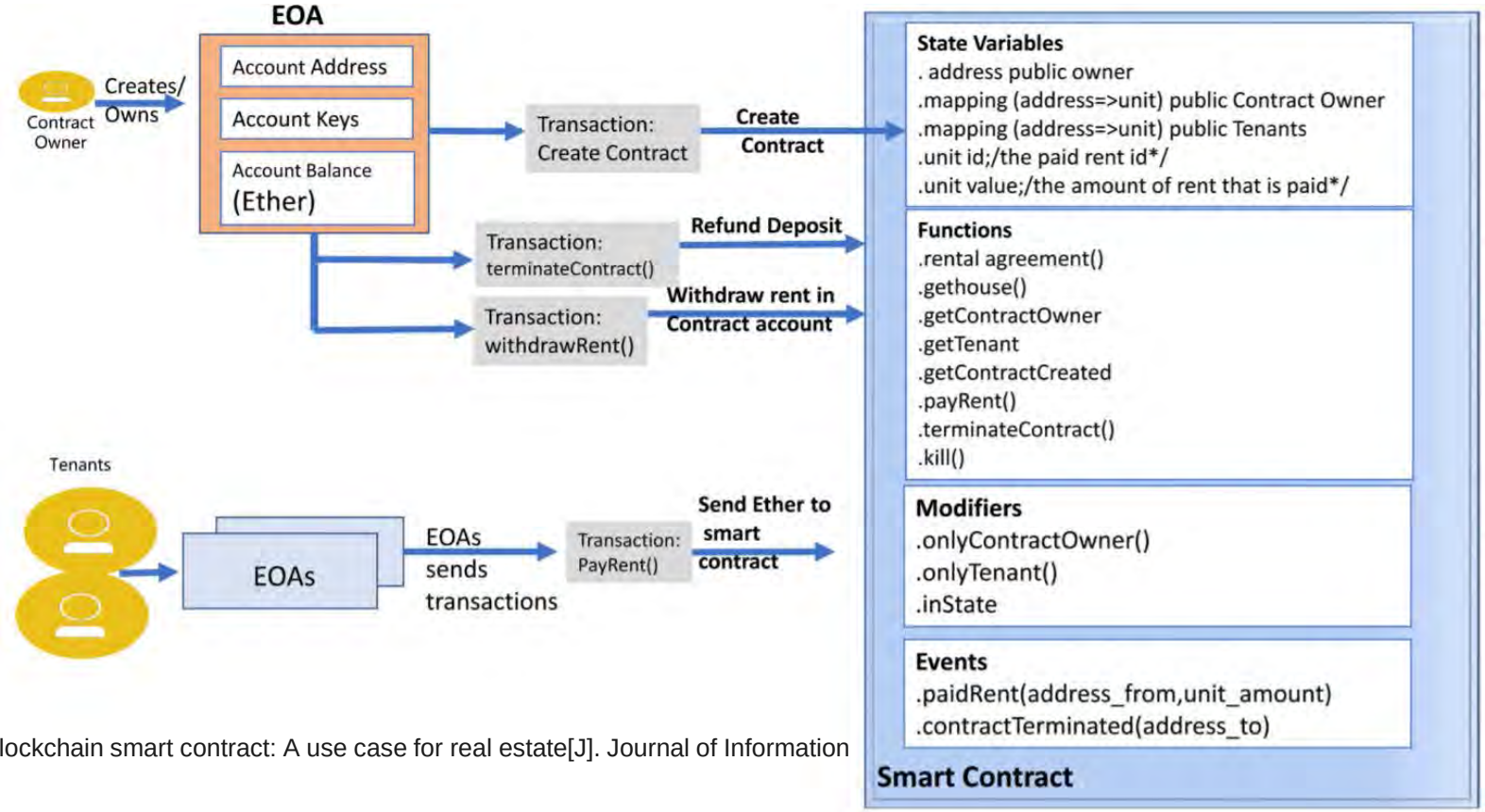


Rental system-use case components

Karamitsos I, Papadaki M, Al Barghuthi N B. Design of the blockchain smart contract: A use case for real estate[J]. Journal of Information Security, 2018, 9(3): 177-190.

Use Case for Real Estate

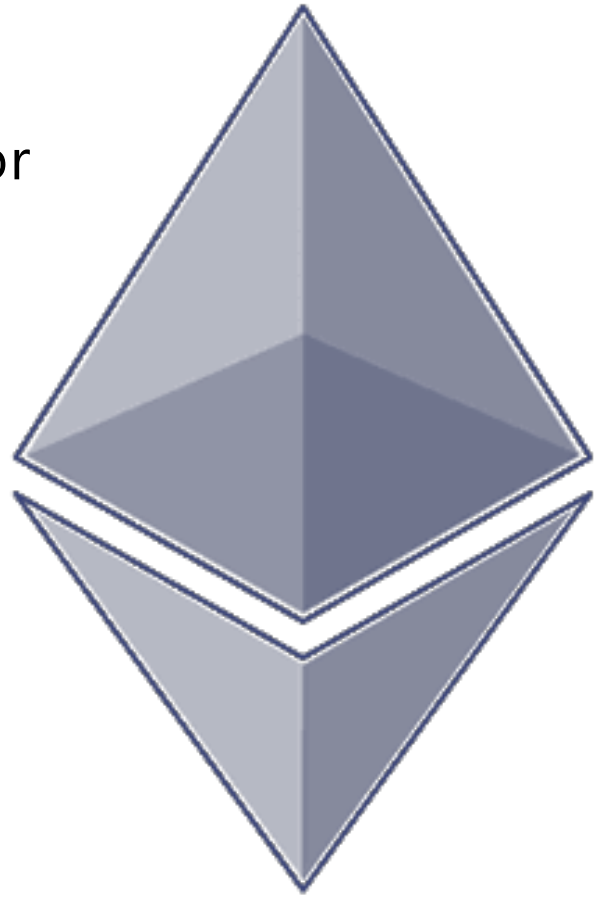
- The real estate acts act landlord for the properties which requires renting a number of residential and business properties using blockchain technology
- **Actors/roles**: external owned account (EOA, including landlord and tenants), contract accounts (CA), and miners
- Smart contract functions
  - Create
  - start
  - Rent_Collection,
  - Terminated
- Smart contract processes
  - Rent contract signature
  - Rental payment
  - Termination rent contract



Karamitsos I, Papadaki M, Al Barghuthi N B. Design of the blockchain smart contract: A use case for real estate[J]. Journal of Information Security, 2018, 9(3): 177-190.

Ethereum

- Blockchain with expressive programming language
  - Programming language makes it ideal for smart contracts
- Why?
  - Most public blockchains are cryptocurrencies
    - Can only transfer coins between users
  - Smart contracts enable much more applications



How Ethereum Works

Two types of account:

- **Normal account** like in Bitcoin
  - has balance and address
- **Smart Contract account**
  - like object: containing (i) code, and (ii) private storage (key-value storage)
  - Code can
    - Send ETH to other accounts
    - Read/write storage
    - Call (ie. start execution in) other contracts

Example

```
1  contract Greetings {
2      string greeting;
3      function Greetings (string _greeting) public {
4          greeting = _greeting;
5      }
6
7      /* main function */
8      function greet() constant returns (string) {
9          return greeting;
10     }
11 }
```

What you write

What other see on the blockchain

```
6060604052604051610250380380610250833981016040528.....
```

```
PUSH 60
    PUSH 40
    MSTORE
    PUSH 0
    CALLDATALOAD
```

What people get from the disassembler

Solidity Contract with Mapping and Struct

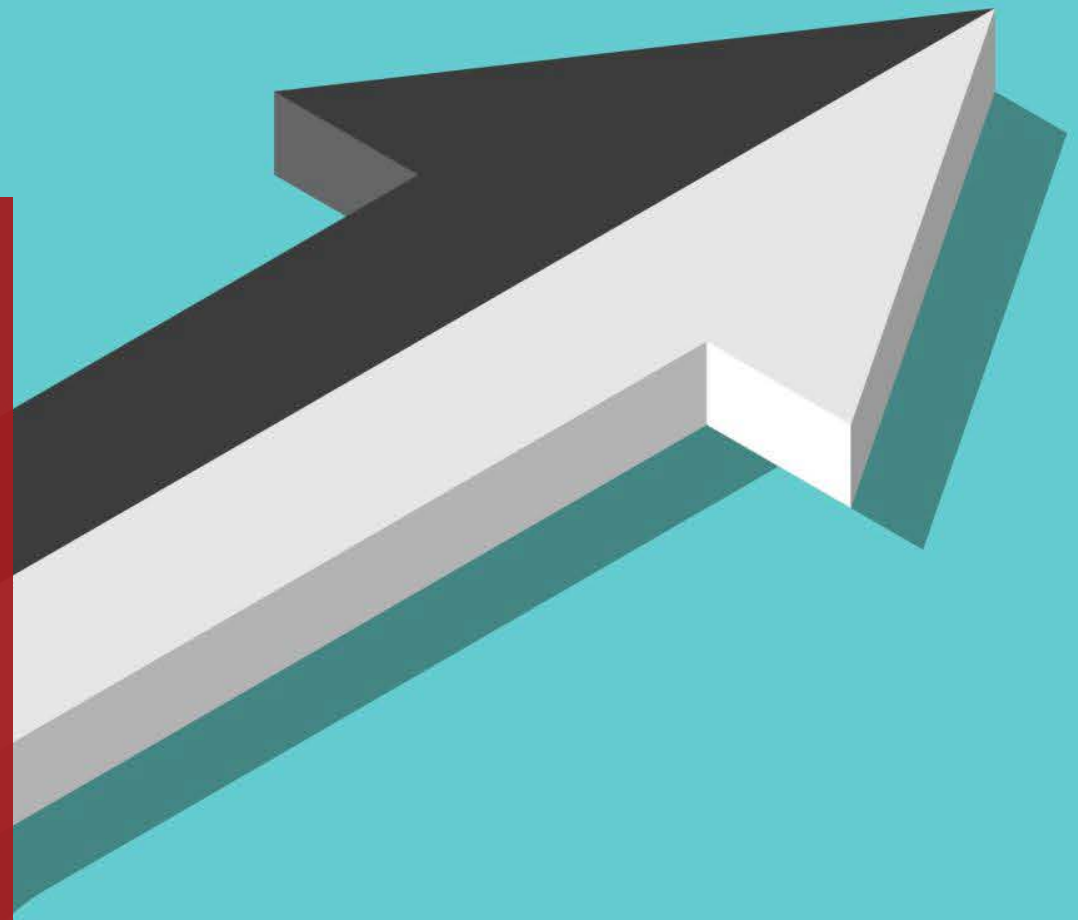
```
pragma solidity 0.5.1;
contract MyContract {

uint256 peopleCount = 0; mapping(uint => Person) public people;

struct Person { uint _id; string _firstName; string _lastName; }

function addPerson(string memory _firstName, string memory _lastName) public {
            peopleCount += 1; people[peopleCount] = Person(peopleCount, _firstName, _lastName);
            }
}
```

- Blockchain Architecture influences application development
- Development phases of blockchain applications
- How to design smart contracts



Next lecture on

Business Architecture

30/9/2020 at 12:35-16:05 online