IoT Resource Access utilizing Blockchains & Trusted Execution Environments

Vasilios A. Siris

joint work with D. Dimopoulos, N. Fotiou, S. Voulgaris, G.C. Polyzos

Mobile Multimedia Laboratory

Athens University of Economics and Business, Greece

vsiris@aueb.gr

Global IoT Summit

17-21 June 2019, Aarhus, Denmark

EU H2020 SOFIE: Secure Open Federation for Internet Everywhere

Motivation and challenges

- Why constrained IoT environments ?
- Why use blockchains & TEEs ? Which type of blockchain ?
- Goal: identify and quantify tradeoffs in terms of **transaction cost, transaction delay, trust, and privacy**

Challenges

- Transaction cost and delay
- Fully decentralized solution
- Ensuring that IoT devices actually provide promised access
- Constrained client devices & constrained IoT resource devices

Single public ledger not enough

Addressed in this paper

Blockchain interaction with real world is a challenge

vsiris@aueb.gr

Why constrained IoT environments?

- Because many IoT devices are constrained in terms of
  - processing and storage resources
  - network connectivity

  } Reducing usage also ***reduces power consumption*** & ***security threats***

Scalability of IoT systems ***can be addressed*** by utilizing device-to-device communication

Device-to-device technologies ***exist*** and are ***becoming mature***

New challenge: how to achieve ***trusted*** device-to-device communication

vsiris@aueb.gr

Why/which blockchain? Features …

- **Decentralized trust,** i.e. no single trusted third party
  - Public ledgers: *wide-scale decentralized trust*
  - Permissioned ledgers: *degree of trust* determined by permissioned set
- **Immutability**
  - related to first point, majority of nodes need to agree to change state
- **Transparency**
  - not only a feature but a *requirement* for decentralized trust
  - tradeoff with *privacy*
- **Availability**, through *decentralized storage and execution*
  - can be achieved other ways

Two baseline models

- Immutable recording of transactions and events
  - Cryptographically link authorization grants to blockchain payments
  - **Record hashes** of authorization messages exchanged on blockchain
- Transparent and trusted execution of authorization logic in **smart contract**
  - More expressive than above
  - Policies can involve IoT events recorded on blockchain
  - Can benefit from blockchain's high availability
  - But more expensive

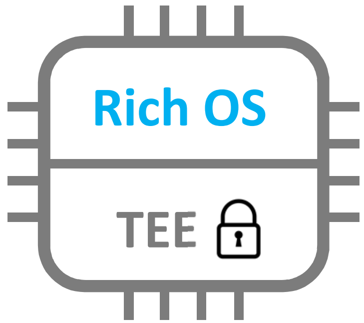
Model 1: Authorization grants linked to blockchain payments and **hashes** recorded

Model 2: **Smart contract** handling authorization requests and encoding policies

vsiris@aueb.gr

Trusted Execution Environment - TEE

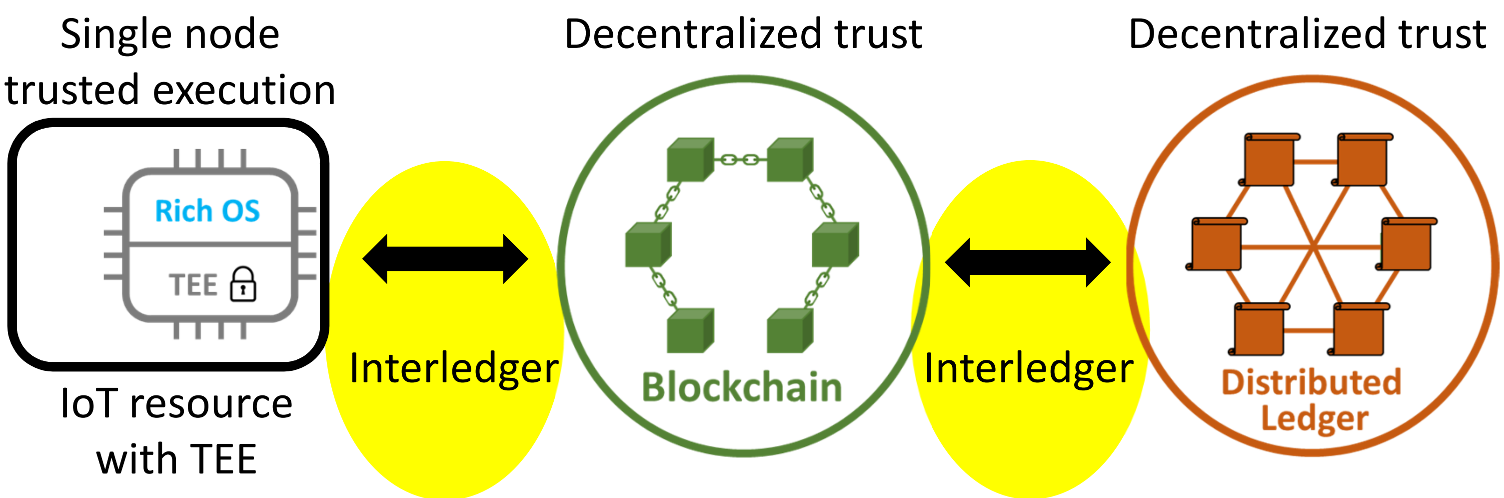
- TEEs provide a secure environment for executing code and storing data, ensuring confidentiality and integrity
- A TEE runs in isolation and in parallel to normal or “rich” OS
- TEE more flexible than TPM (Trusted Platform Modules)
- TEE environments:
  - ARM TrustZone: widely deployed in mobile devices and micro-controller devices
  - Intel Software Guard eXtension (SGX)
  - Keystone open-source secure enclaved for Risc-V



vsiris@aueb.gr

Exploiting Trusted Execution Environments

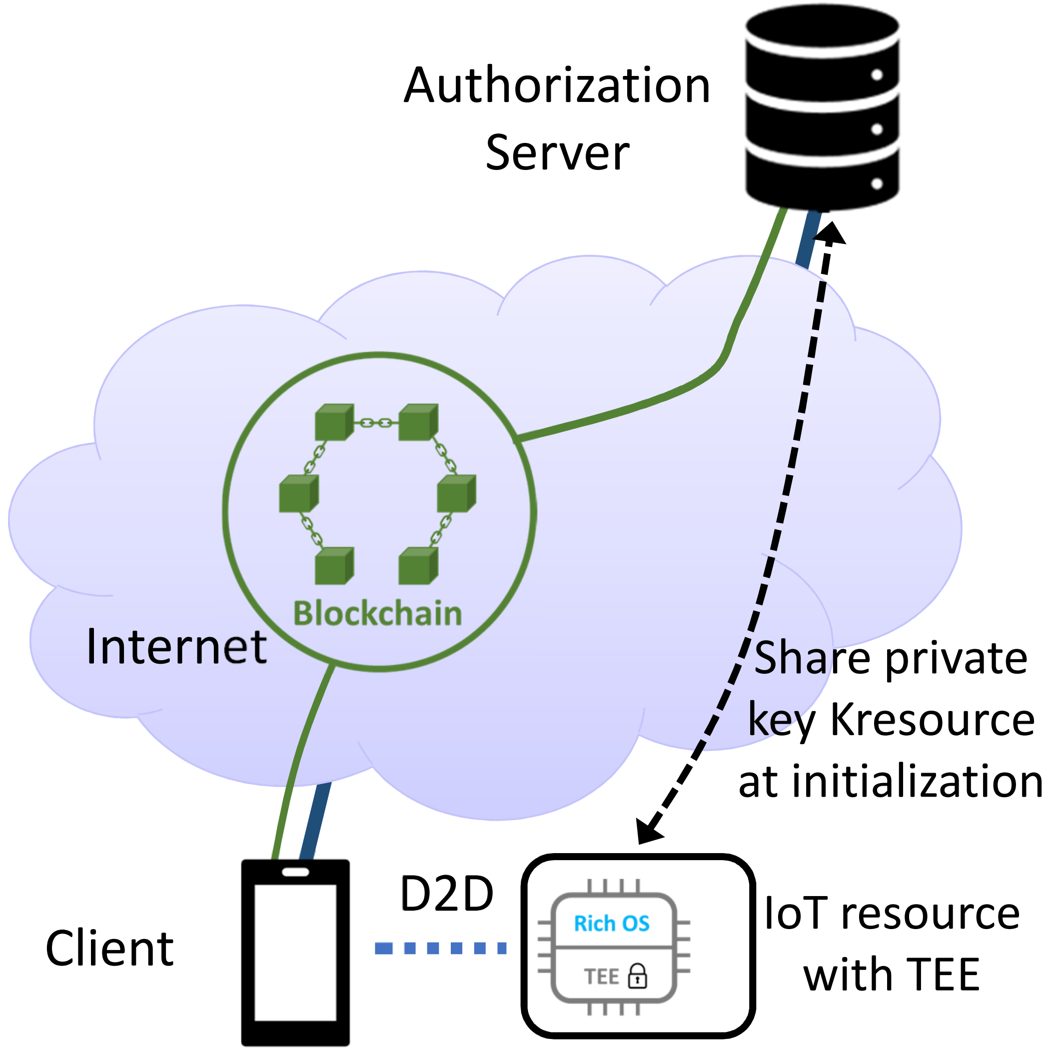
- TEEs provide trusted execution & storage in a **single node**
  - help ensure that IoT resource actually provide promised access
- Blockchains and Distributed Ledgers provide **decentralized trust**
- DLT and TEE guarantees provided only for transactions inside particular DLT/TEE
- Goal: **securely link transactions & events across ledgers and TEEs**



vsiris@aueb.gr

Assumptions

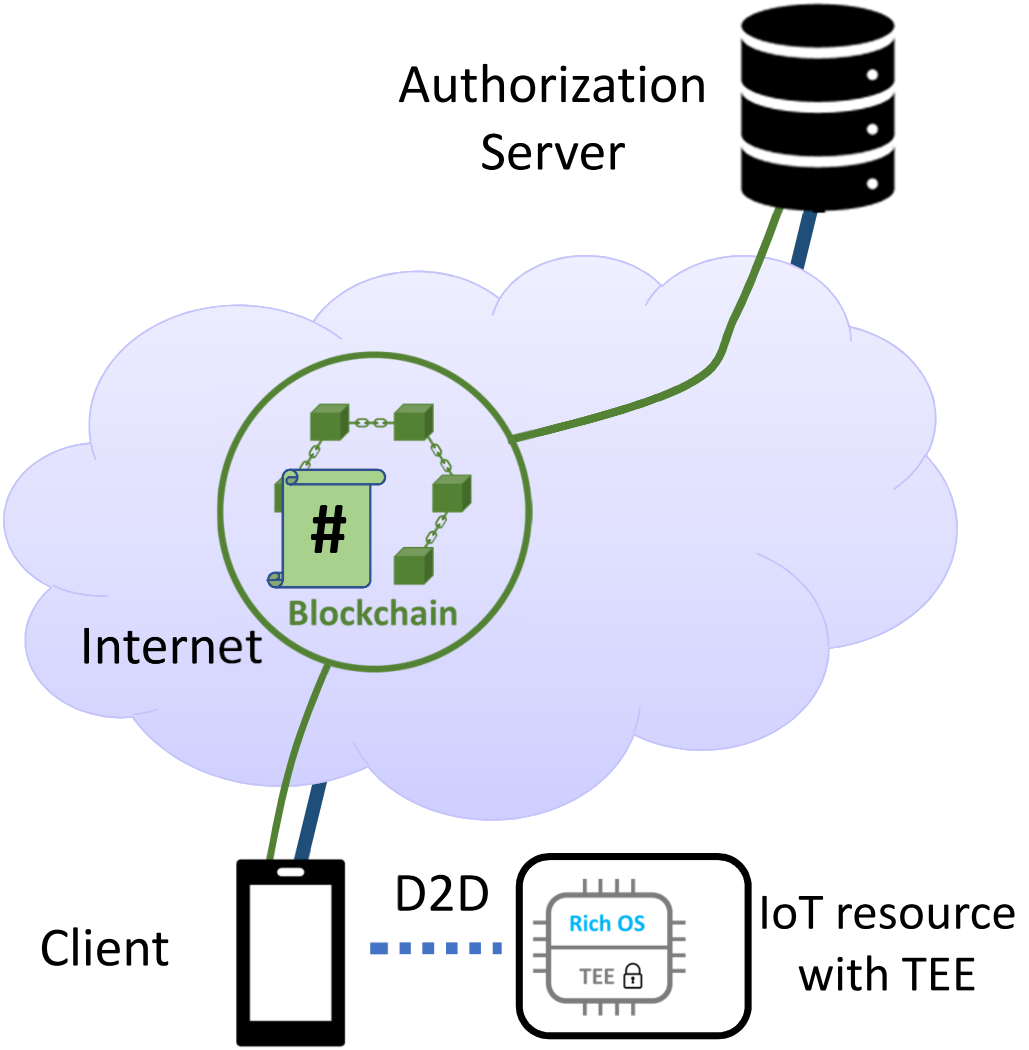
- IoT resource has limited processing, storage and only D2D connectivity
  - *Previous work assumes IoT devices always connected and interact directly with blockchain*
- IoT resources has TEE
- Authorization Server (AS) handles requests on behalf of IoT resource
  - OAuth 2.0 authorization framework
  - Based on access tokens
  - AS and resource share private key KResource
- Client and AS always connected and can interact with blockchain



vsiris@aueb.gr

Model 1: Authorization grants linked to blockchain payments and hashes recorded

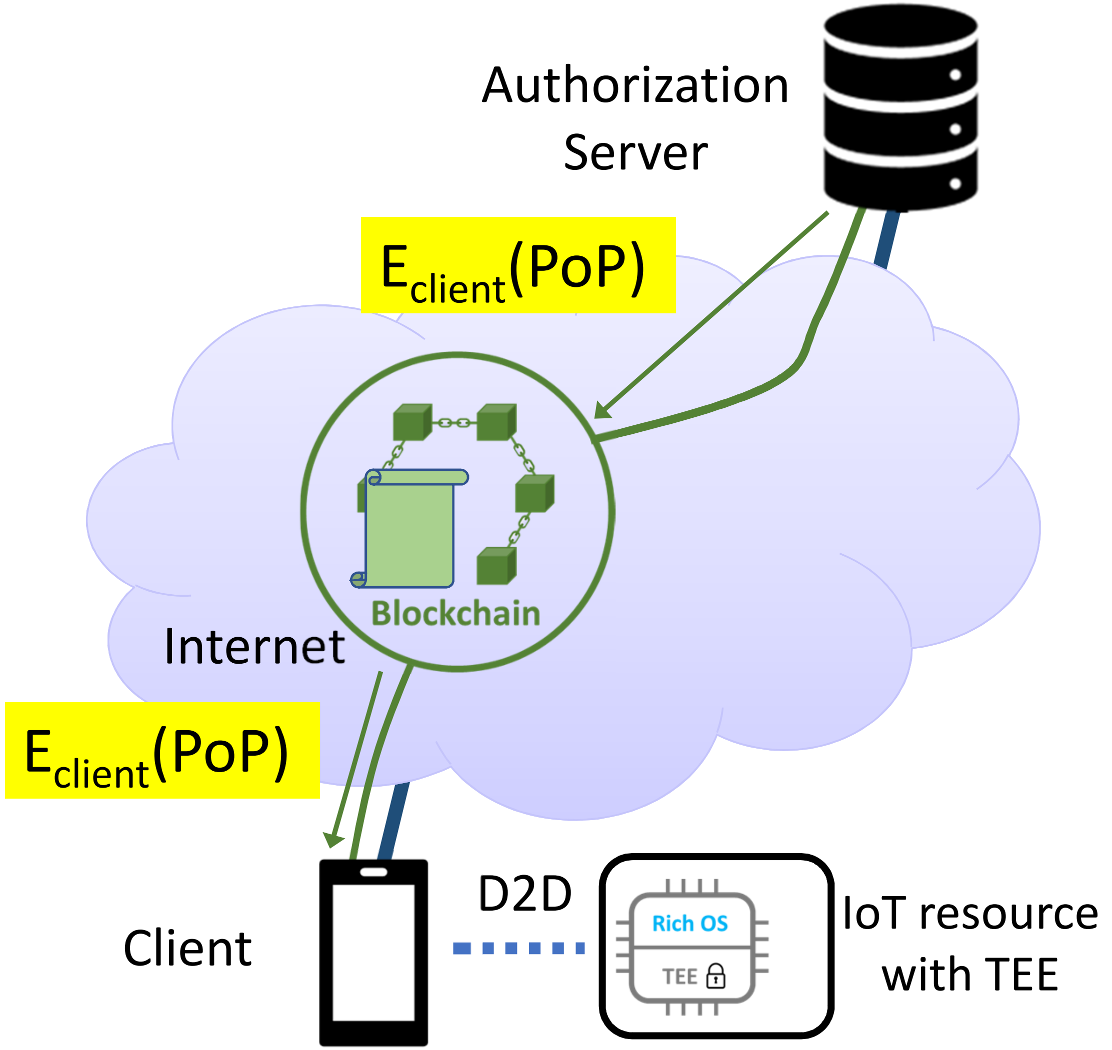
- Client and AS communicate directly as in OAuth 2.0
- Proof-of-Possession (PoP) used to secure client-IoT resource D2D link
- Client submits token secured using PoP to IoT resource
- IoT resource's TEE selects secret s and h=Hash(s)
- IoT resource sends s encrypted to AS and sends h to client and AS
- Client deposits amount for accessing resource
- Deposit transferred to resource owner when s revealed on blockchain
- Client reads secret s on blockchain and sends to IoT resource to obtain access
- Hash of messages exchanged between client and AS recorded on blockchain



vsiris@aueb.gr

Model 2: Smart contract handling authorization requests and encoding policies

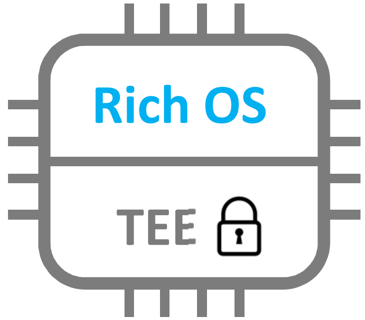
- Client sends authorization request to Smart Contract
- Smart Contract transparently records prices and authorization policies (defined by resource owner)
- As in first model, payments linked to authorization requests
- Unlike previous model: because data on blockchain public need to encrypt part of token with client's public key



vsiris@aueb.gr

Implementation

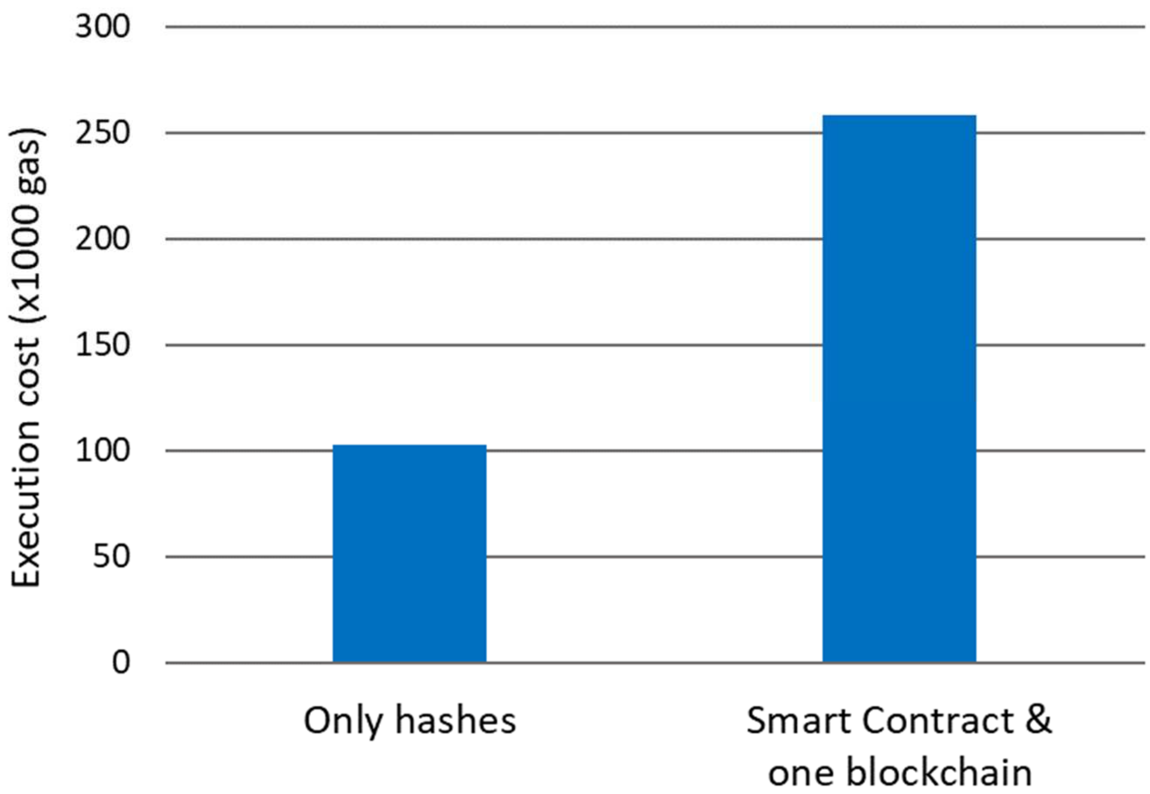
- OP-TEE (Open Portable Trusted Execution Environment) open source port for the Raspberry Pi
  - uses ARM's TrustZone
  - Follows GlobalPlatform TEE system architecture
- IoT resource's TEE
  - Selects secret s used to cryptographically link transactions between TEE and DLT
  - Verifies secret s and access token
  - Guarantees access to IoT resource
- Local Ethereum node running Go-Ethereum connected to Rinkeby public Ethereum testnet
- Smart contracts written in Solidity with Remix web-based editor
- AS based on a PHP implementation of OAuth 2.0 and used web3.js to connect to Rinkeby



vsiris@aueb.gr

Results: execution cost

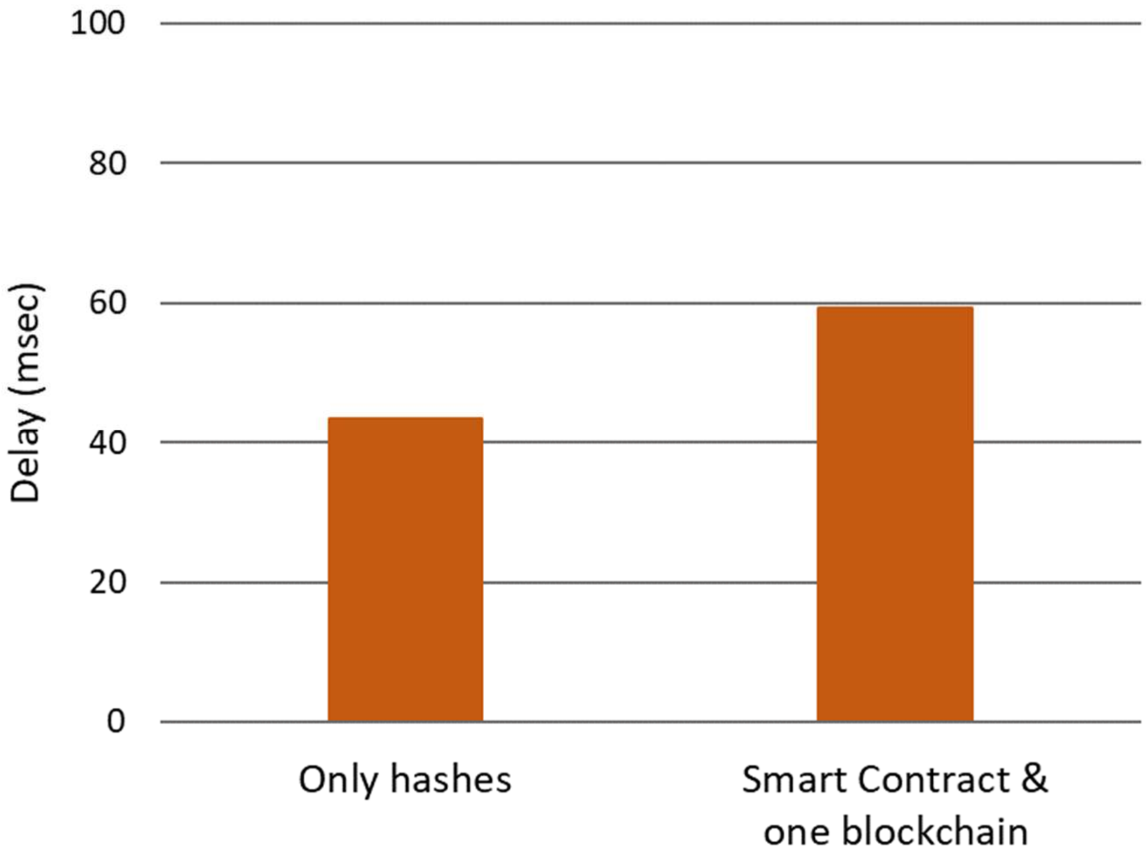
- Smart contract requires 2.5 times EVM gas compared to simply recording hashes
- Only write transactions cost gas
  - Reading data has zero cost
- Quantifies cost for higher functionality of smart contracts
  - Authorization policies & logic



vsiris@aueb.gr

Results: delay

- Delay determined by blockchain transaction time
- Smart contract model has four transactions versus three transactions of hash recording model
  - 33% higher delay



vsiris@aueb.gr

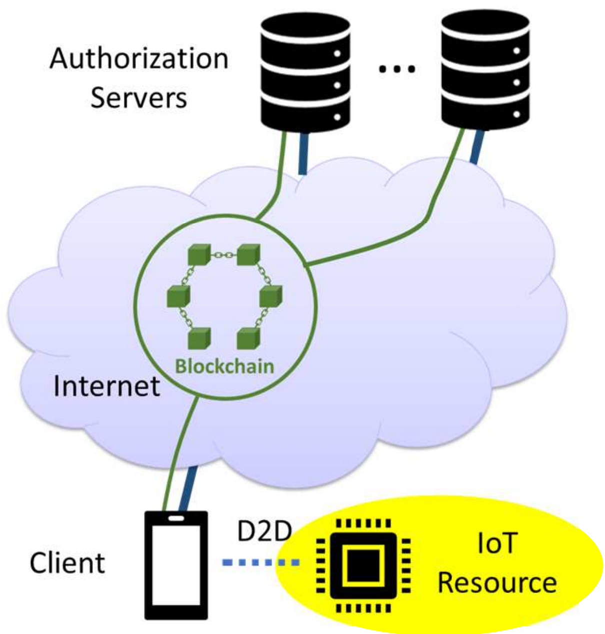
Other challenges

Move smart contract to permissioned ledger and/or only record hashes on public ledger

Smart contract on public ledger

Achieved by combining public with private/permissioned ledger

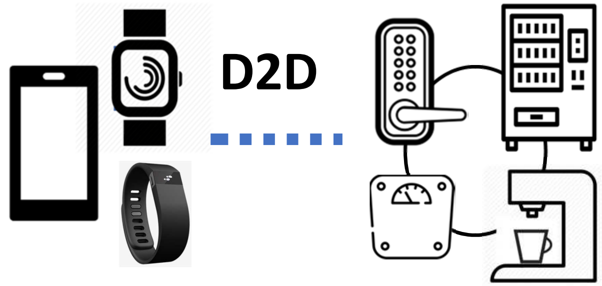
- High cost & delay incurred by blockchains
  - Due to public ledger
  - Combining public & private/permissioned ledgers can provide different tradeoffs of cost, trust, and privacy
  - Off-chain transactions: unidirectional payment channels sufficient for some IoT applications
- Single AS
  - Blockchain advantages are limited to assets & transactions residing in the blockchain
  - Once we traverse blockchain boundaries we loose these benefits
  - Solely adding multiple ASes not a solution because IoT resource not directly connected to blockchain
  - Need processing at client to reduce data & ensure trust with constrained IoT resource



vsiris@aueb.gr

Other challenges (cont)

- Constrained clients
  - Need client proxy/agent (analogous to AS acting as proxy of IoT resource)



Papers – see also https://mm.aueb.gr/blockchains/

"IoT Resource Access utilizing Blockchains and Trusted Execution Environments", Global IoT Summit 2019

"Secure IoT access at scale using blockchains and smart contracts", IEEE IoT-SoS 2019

"Trusted D2D-based IoT Resource Access using Smart Contracts", IEEE WoWMoM 2019

"Smart Contracts for Decentralized Authorization to Constrained Things", CryBlock 2019 workshop at IEEE INFOCOM 2019

"OAuth 2.0 meets Blockchain for Authorization in Constrained IoT Environments", IEEE World Forum on IoT 2019

"Enabling Decentralised Identifiers and Verifiable Credentials for Constrained Internet-of-Things Devices using OAuth-based Delegation", DISS workshop at NDSS 2019

"Bridging the Cyber and Physical Worlds using Blockchains and Smart Contracts", DISS workshop at NDSS 2019

"Interacting with the Internet of Things Using Smart Contracts and Blockchain Technologies", SpaCCS 2018

vsiris@aueb.gr