Interledger Smart Contracts for Decentralized Authorization to Constrained Things

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

CryBlock 2019 – IEEE INFOCOM 2019

29 April 2019, Paris

EU H2020 SOFIE: Secure Open Federation for Internet Everywhere

Motivation and goal

- Why constrained IoT environments ?
- Why (not) blockchains ?
- Goal: investigate options for integrating blockchains with authorization to constrained IoT devices with different cost/functionality tradeoffs
- Key challenges
  - Transaction cost and delay
  - Fully decentralized solution
  - Ensuring that IoT devices actually provide promised access

Addressed in this paper

Single public ledger not enough

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 blockchains? 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

vsiris@aueb.gr

Two baseline models

- Immutable recording of transactions and events
  - Cryptographically link authorization grants to blockchain payments
  - Record hashes of authorization messages exchanged on blockchain

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

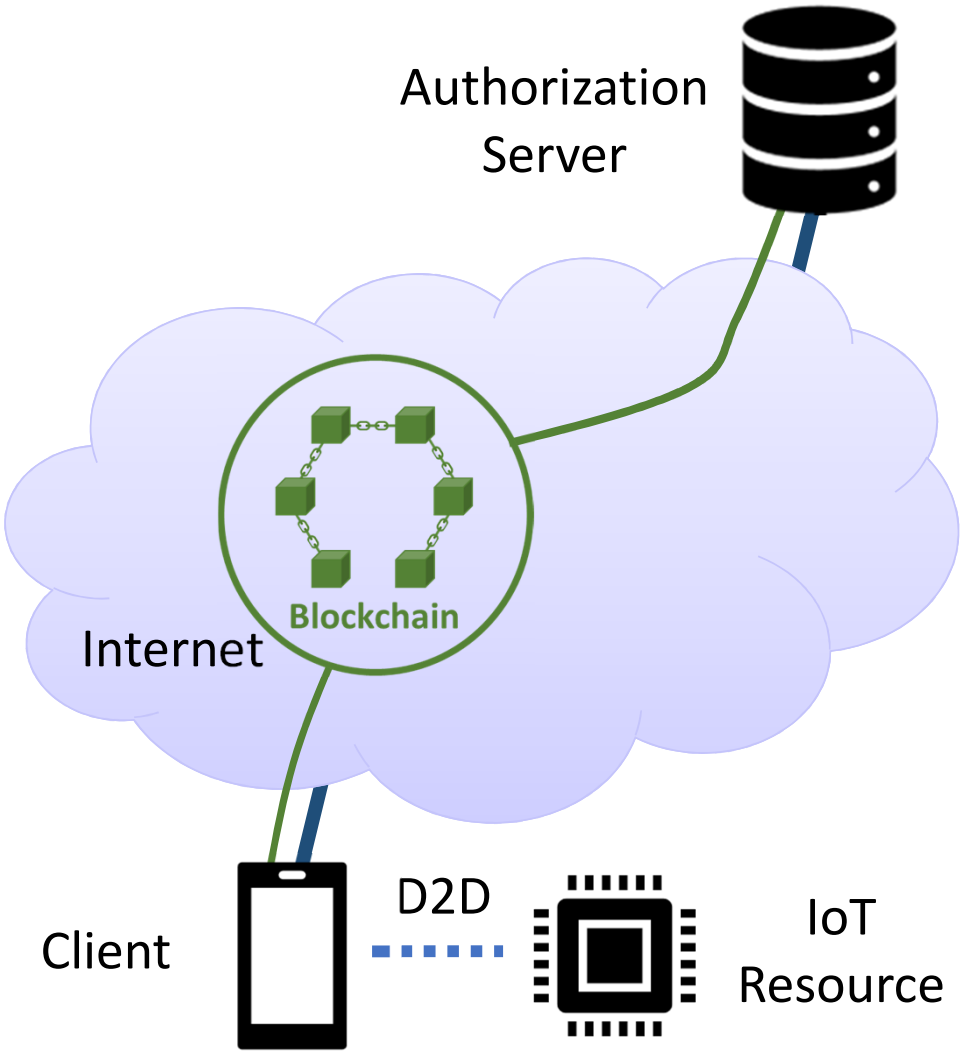
- Transparent and trusted execution of authorization logic
  - More expressive than above
  - Policies can involve IoT events recorded on blockchain
  - Can benefit from blockchain's high availability
  - But more expensive

Model 2: Smart contract handling authorization requests and encoding policies

vsiris@aueb.gr

Assumptions

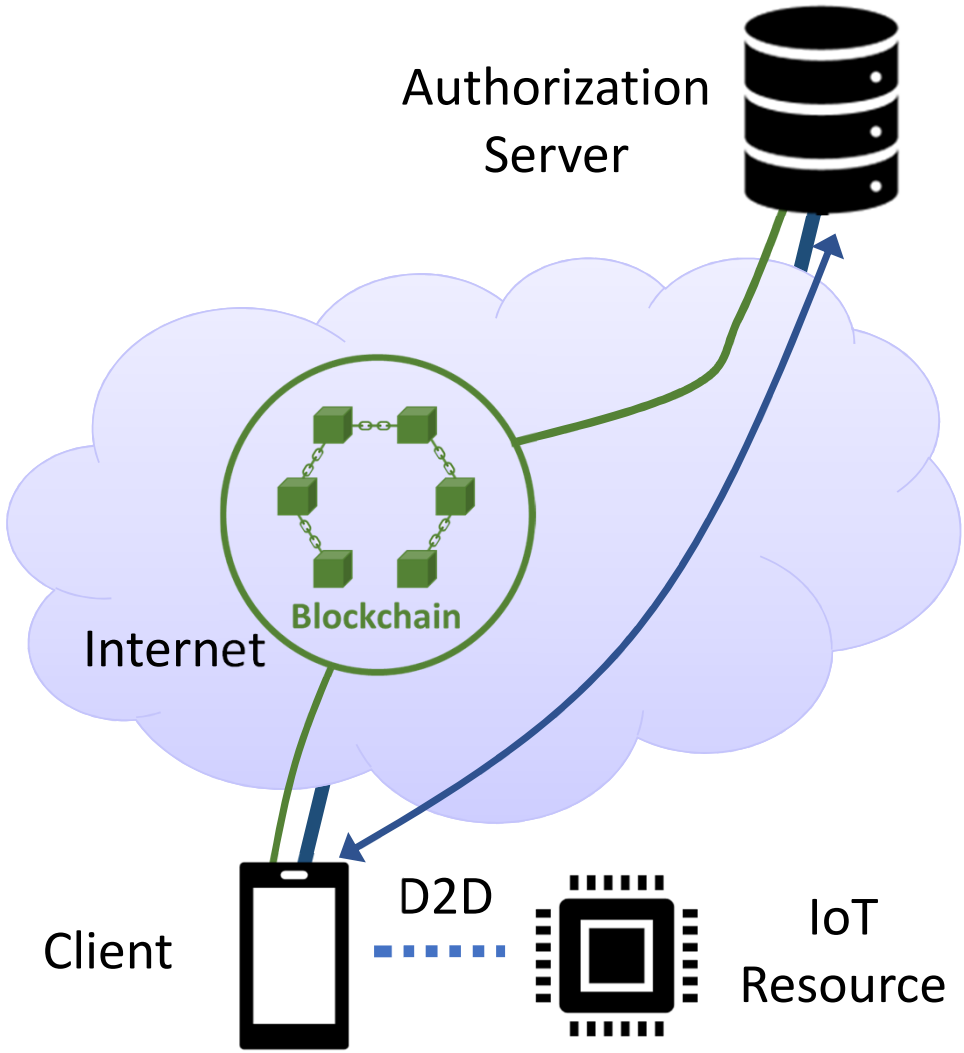
- IoT resource has limited processing, storage and only D2D connectivity
  - *Previous work assumes IoT devices always connected and interact directly with blockchain*
- Authorization Server (AS) handles requests on behalf of IoT resource
  - OAuth 2.0 authorization framework
  - Based on access tokens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



vsiris@aueb.gr

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

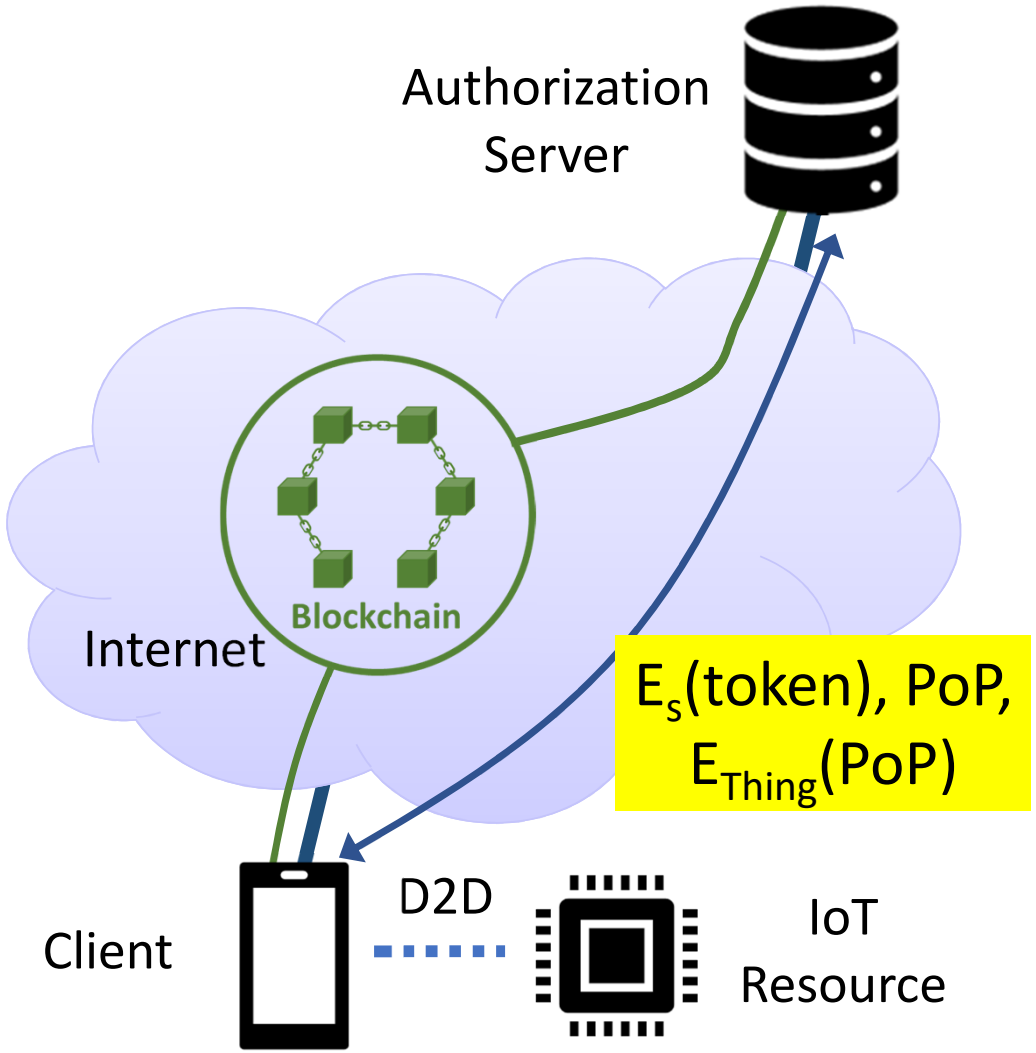
- Client and AS communicate directly as in OAuth 2.0



vsiris@aueb.gr

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

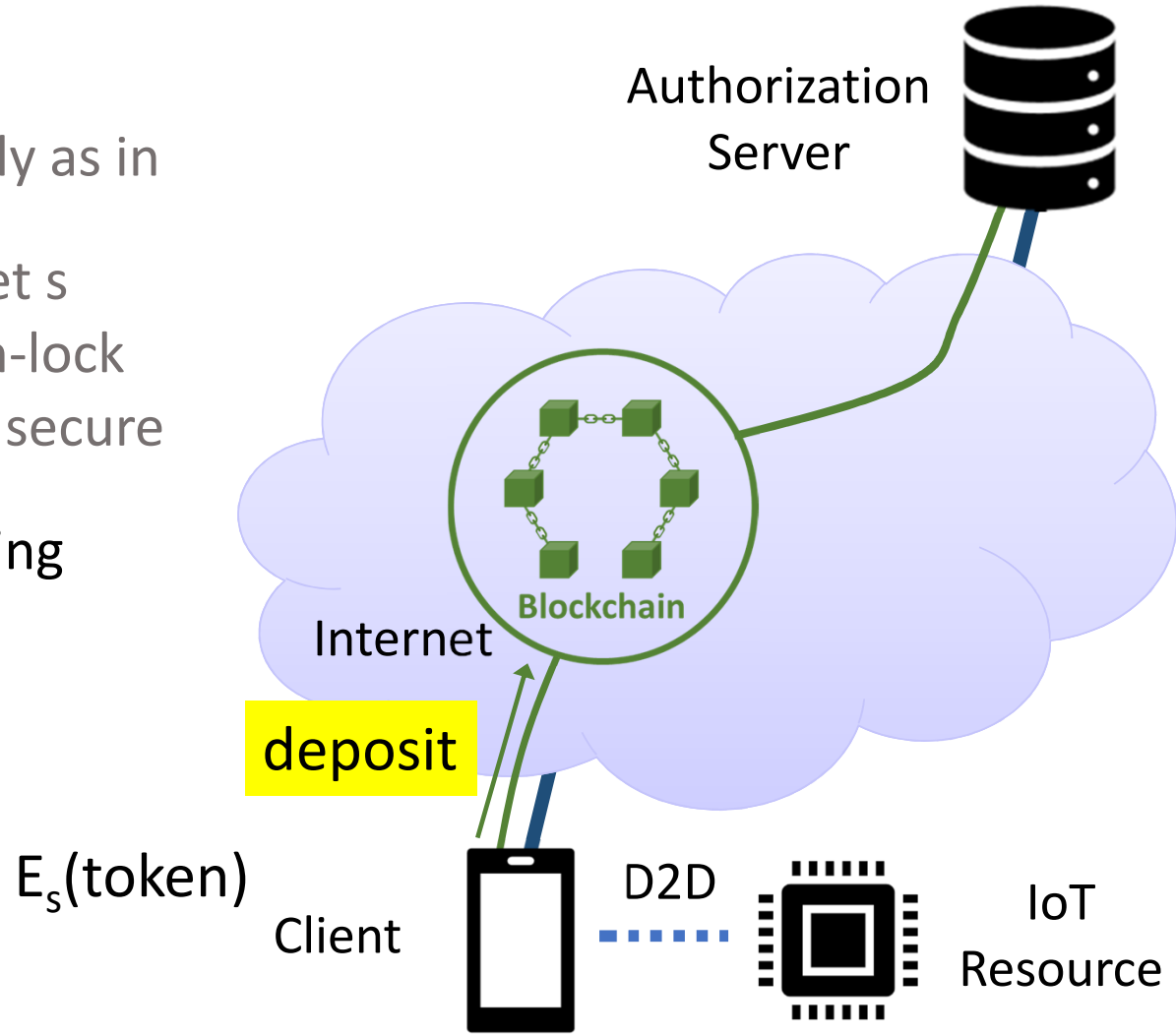
- Client and AS communicate directly as in OAuth 2.0
- Access token encrypted with secret s
- Secret s related to payment's hash-lock
- Proof-of-Possession (PoP) used to secure client-IoT resource D2D link



vsiris@aueb.gr

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

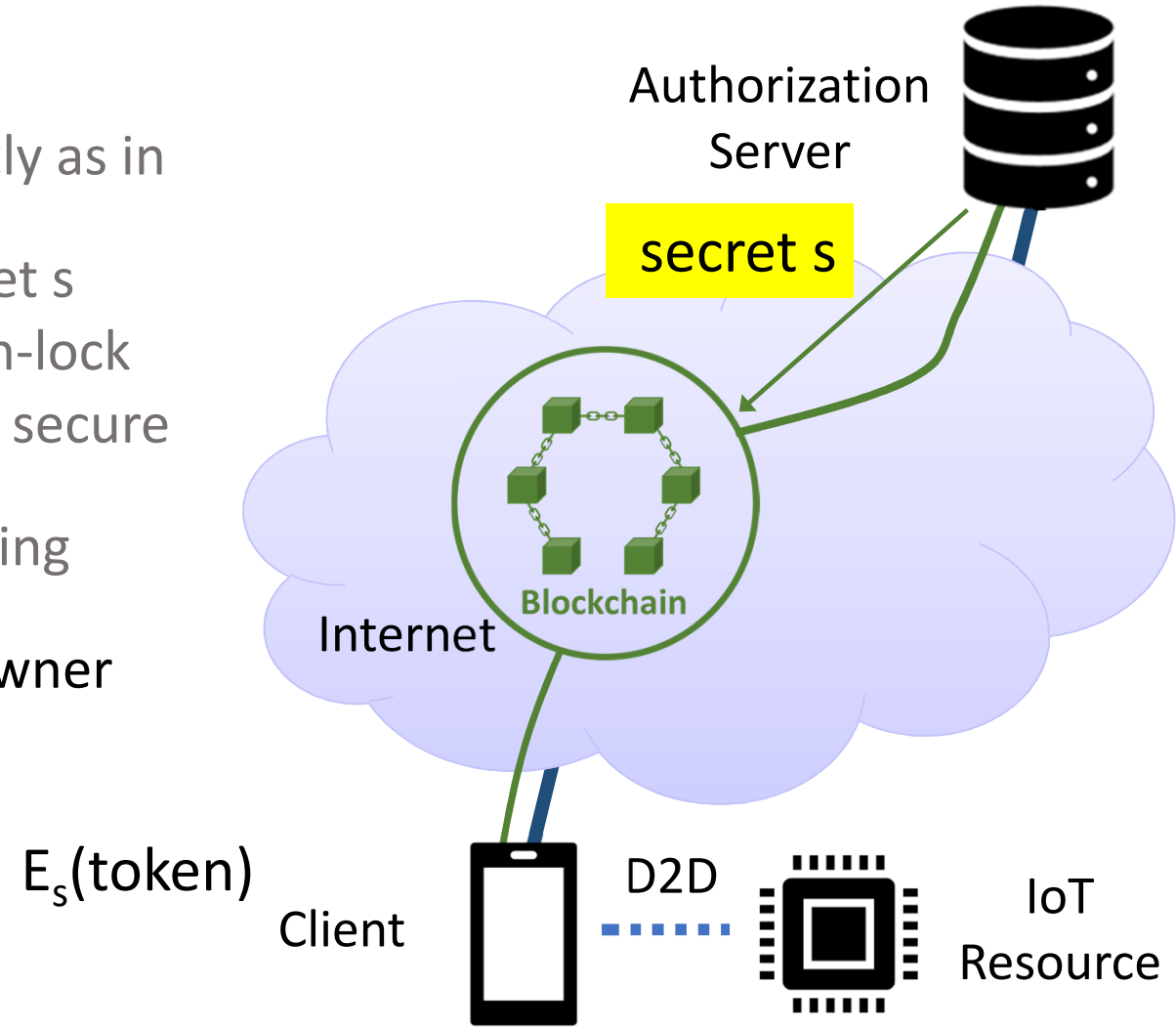
- Client and AS communicate directly as in OAuth 2.0
- Access token encrypted with secret s
- Secret s related to payment's hash-lock
- Proof-of-Possession (PoP) used to secure client-IoT resource D2D link
- Client deposits amount for accessing resource



vsiris@aueb.gr

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

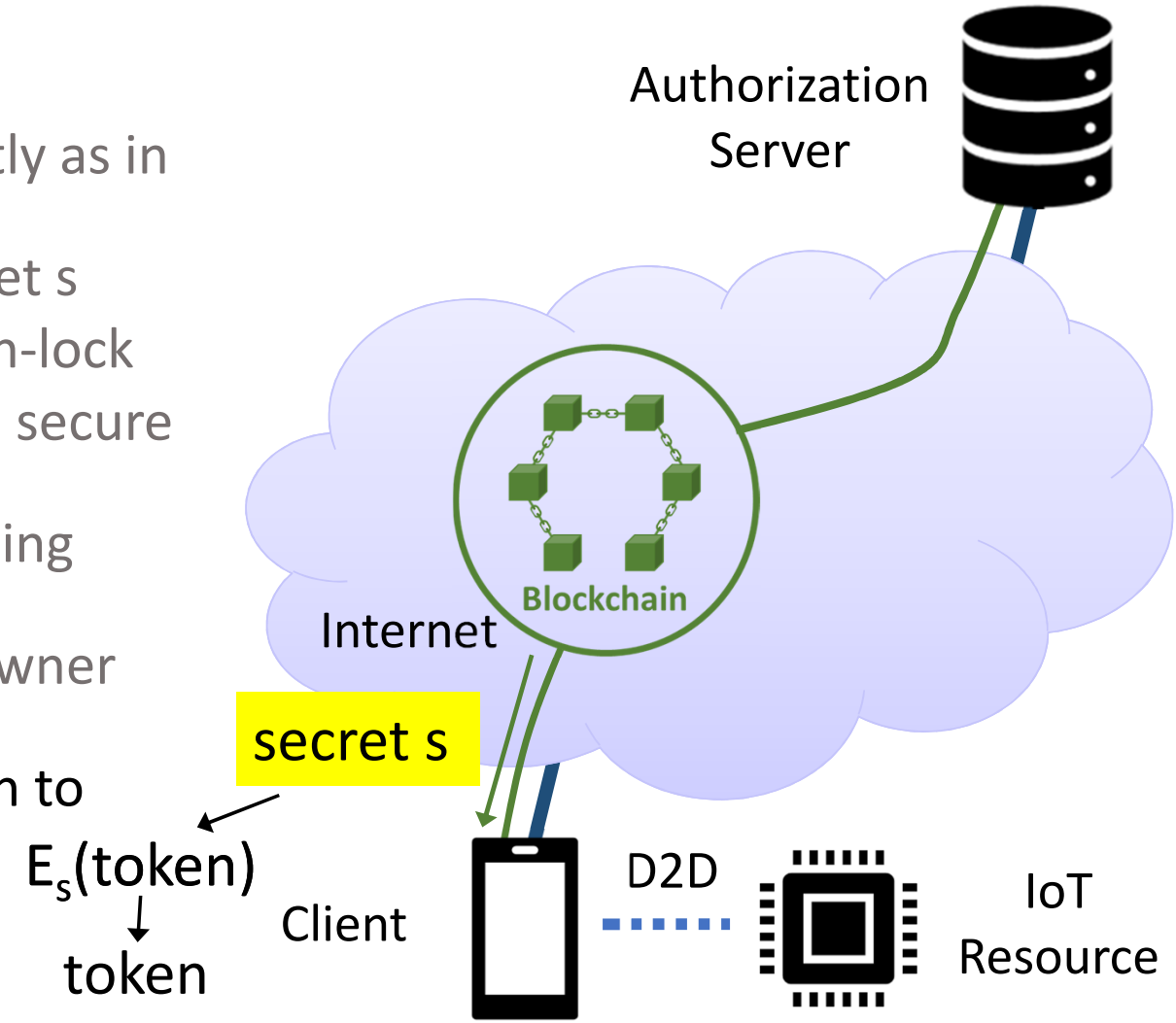
- Client and AS communicate directly as in OAuth 2.0
- Access token encrypted with secret s
- Secret s related to payment's hash-lock
- Proof-of-Possession (PoP) used to secure client-IoT resource D2D link
- Client deposits amount for accessing resource
- Deposit transferred to resource owner when s revealed on blockchain



vsiris@aueb.gr

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

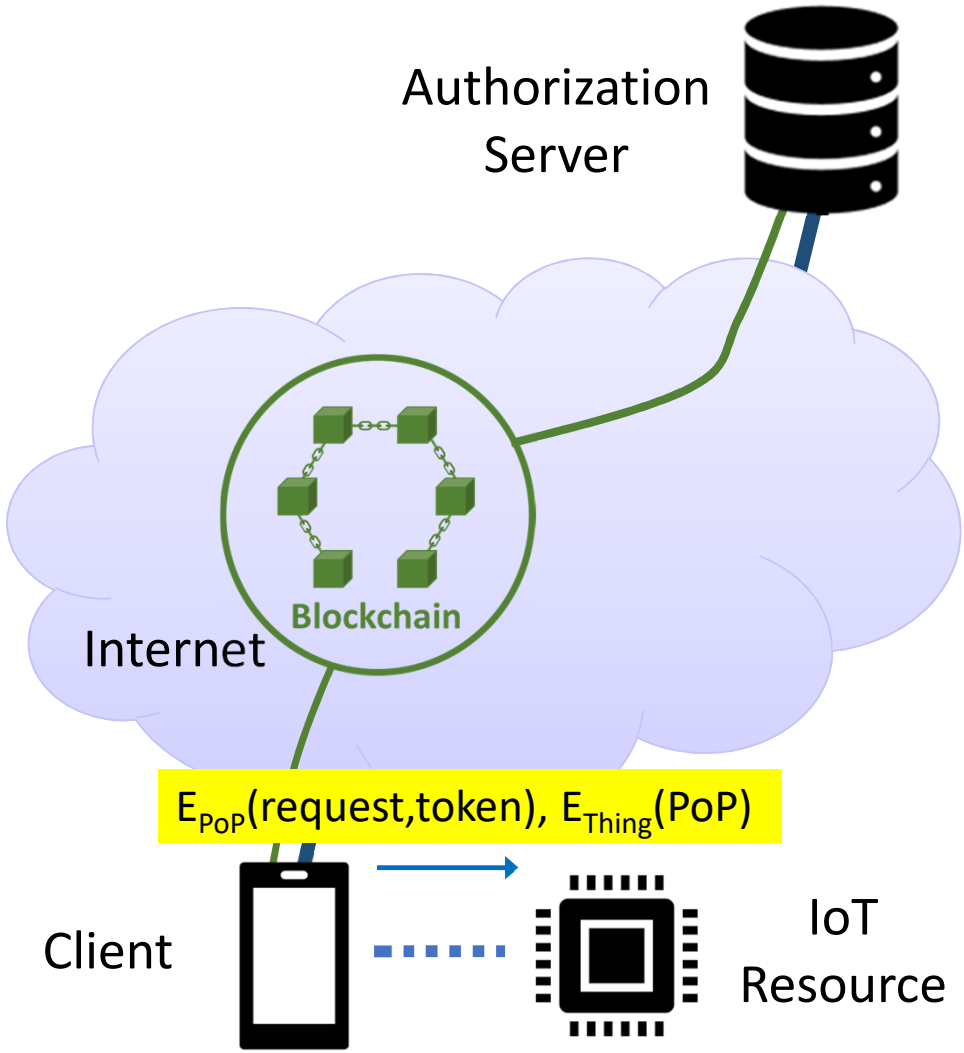
- Client and AS communicate directly as in OAuth 2.0
- Access token encrypted with secret s
- Secret s related to payment's hash-lock
- Proof-of-Possession (PoP) used to secure client-IoT resource D2D link
- Client deposits amount for accessing resource
- Deposit transferred to resource owner when s revealed on blockchain
- Client reads secret s on blockchain to decrypt access token



vsiris@aueb.gr

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

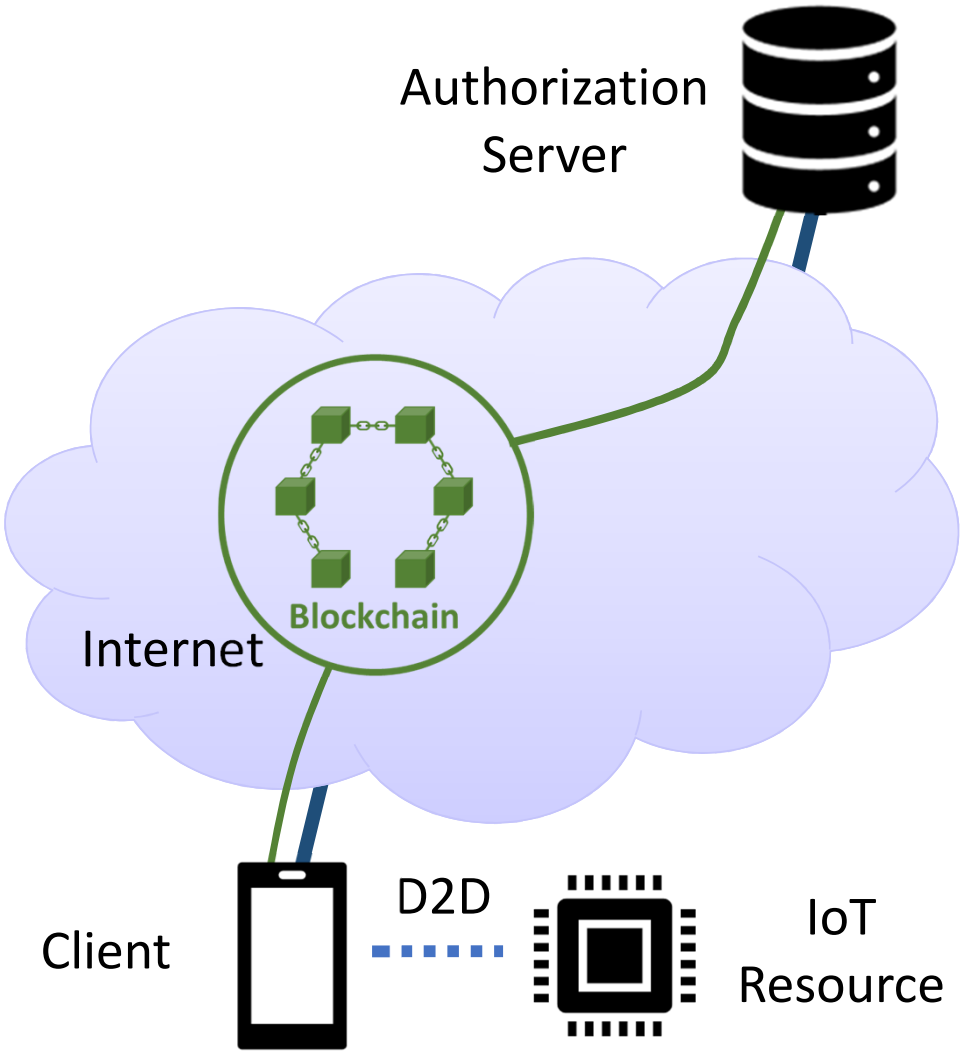
- Client and AS communicate directly as in OAuth 2.0
- Access token encrypted with secret s
- Secret s related to payment's hash-lock
- Proof-of-Possession (PoP) used to secure client-IoT resource D2D link
- Client deposits amount for accessing resource
- Deposit transferred to resource owner when s revealed on blockchain
- Client reads secret s on blockchain to decrypt access token



vsiris@aueb.gr

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

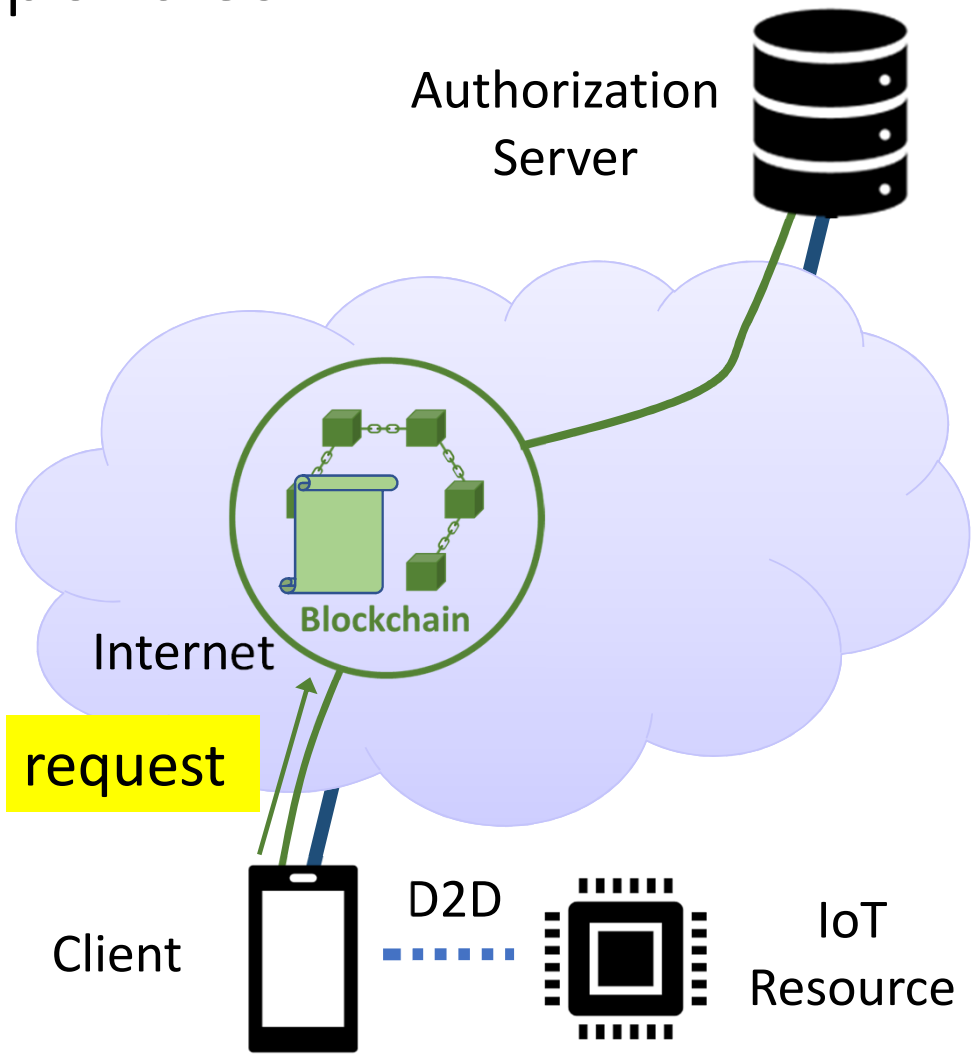
- Client and AS communicate directly as in OAuth 2.0
- Access token encrypted with secret s
- Secret s related to payment's hash-lock
- Proof-of-Possession (PoP) used to secure client-IoT resource D2D link
- Client deposits amount for accessing resource
- Deposit transferred to resource owner when s revealed on blockchain
- Client reads secret s on blockchain to decrypt access token
- Hash of messages exchanged between client and AS recorded on blockchain



vsiris@aueb.gr

Model 2: Smart contract handling authorization requests and encoding policies

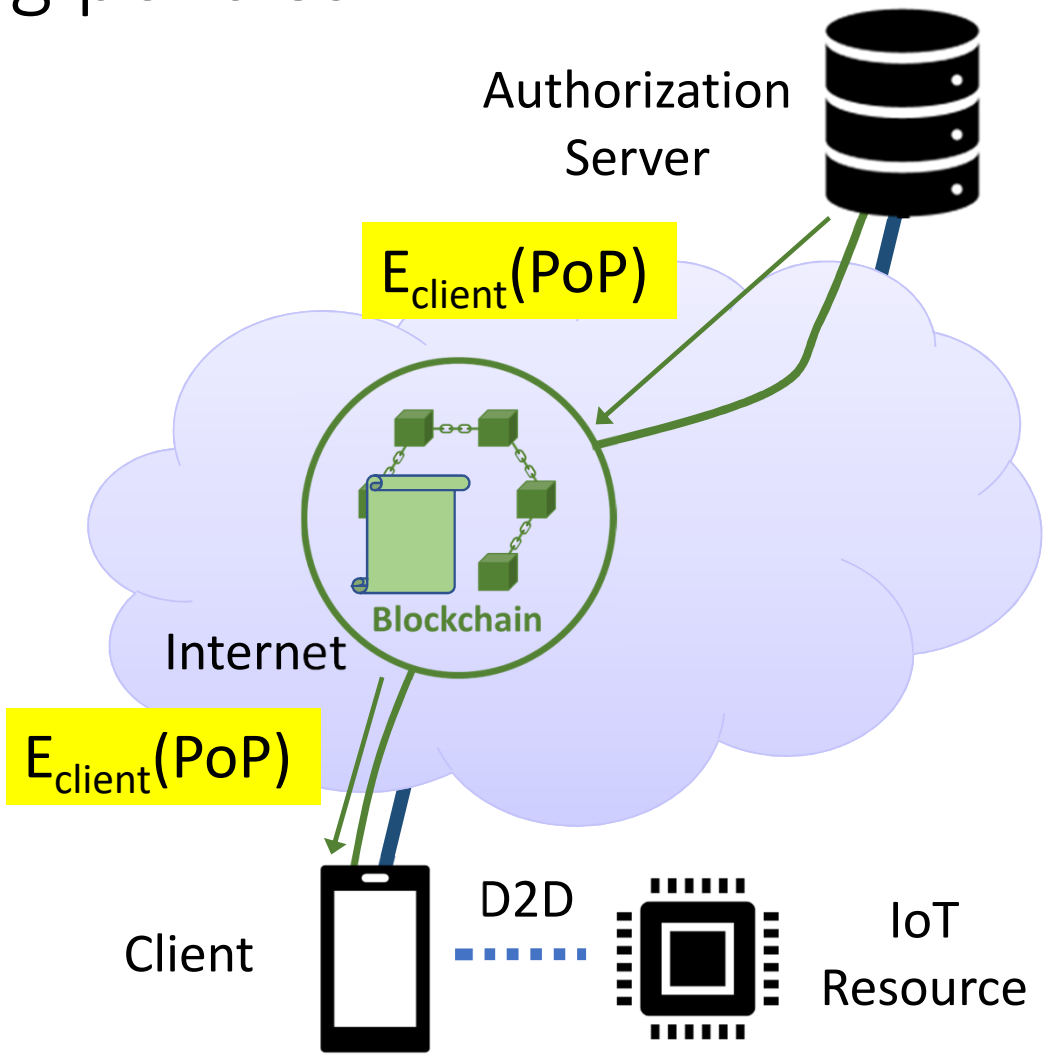
- Client sends authorization request to Smart Contract



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 previous 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

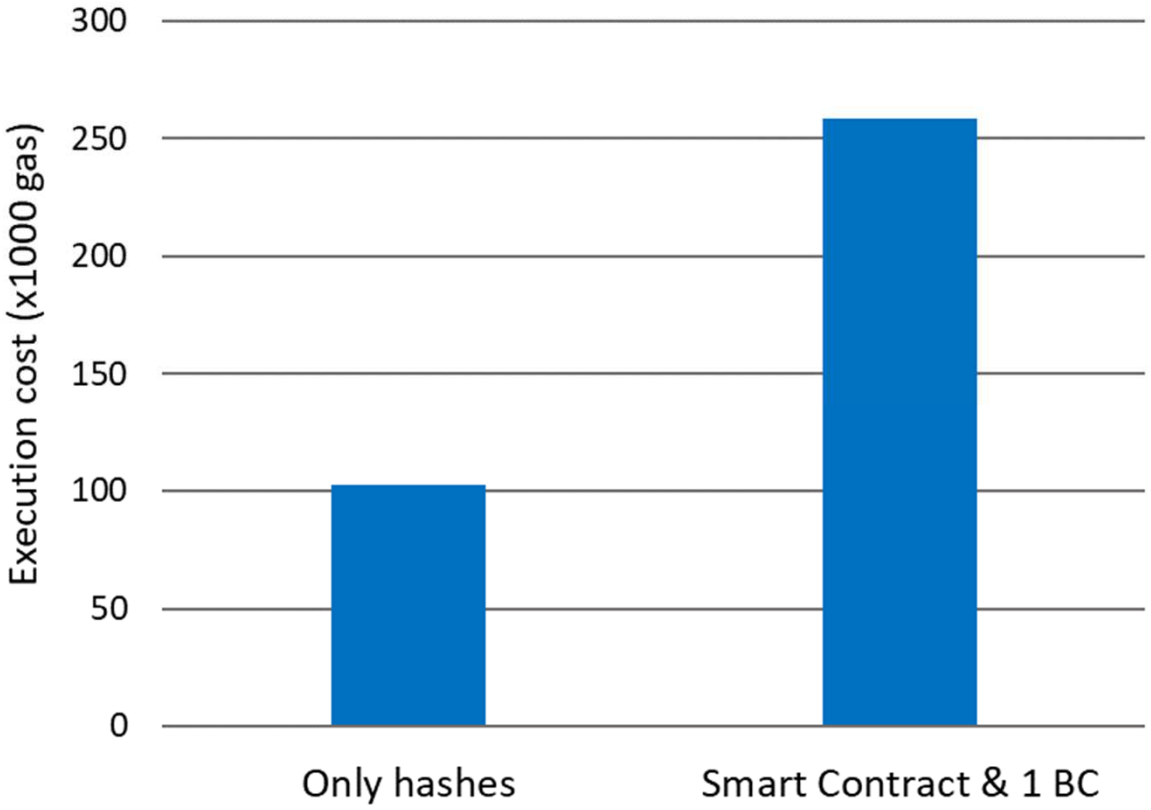
Implementation

- Deployed local node connected to Rinkeby and Ropsten public Ethereum testnets
  - Private chain is a local Ethereum network
- Smart contract written in Solidity with Remix web-based editor
- Web3.0 to interact with Rinkeby and Ropsten blockchains
- Authorization server based on open PHP implementation of OAuth 2.0
- CBOR (Concise Binary Object Representation) Web Token (CWT)
  - More efficient than JSON Web Token (JWT) encoding

vsiris@aueb.gr

Single blockchain results: execution cost

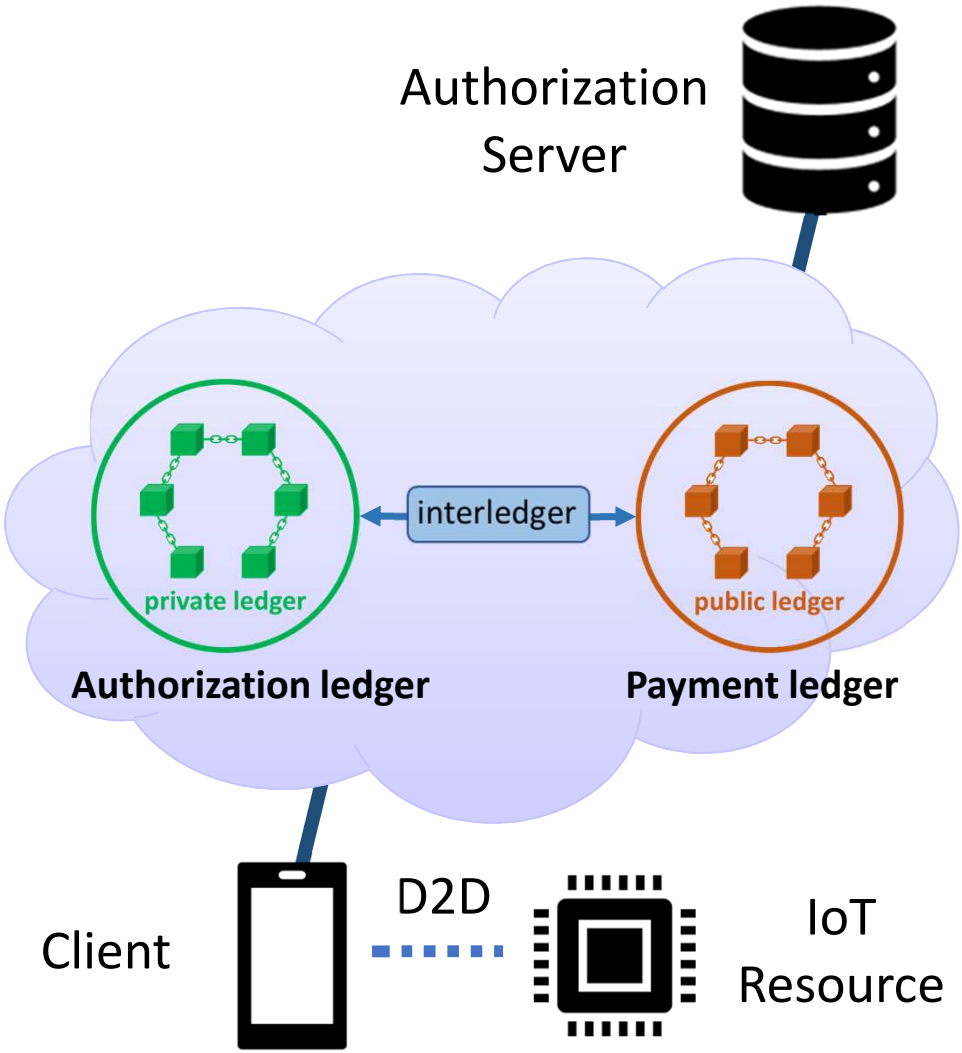
- Smart contract requires 2.5 times EVM gas compared to simply recording hashes
- Only write transactions cost gas
  - Reading data has zero cost
- Quantifies cost for higher functionality of smart contracts
  - Authorization policies & logic



vsiris@aueb.gr

Model 3: Combine public & private ledgers

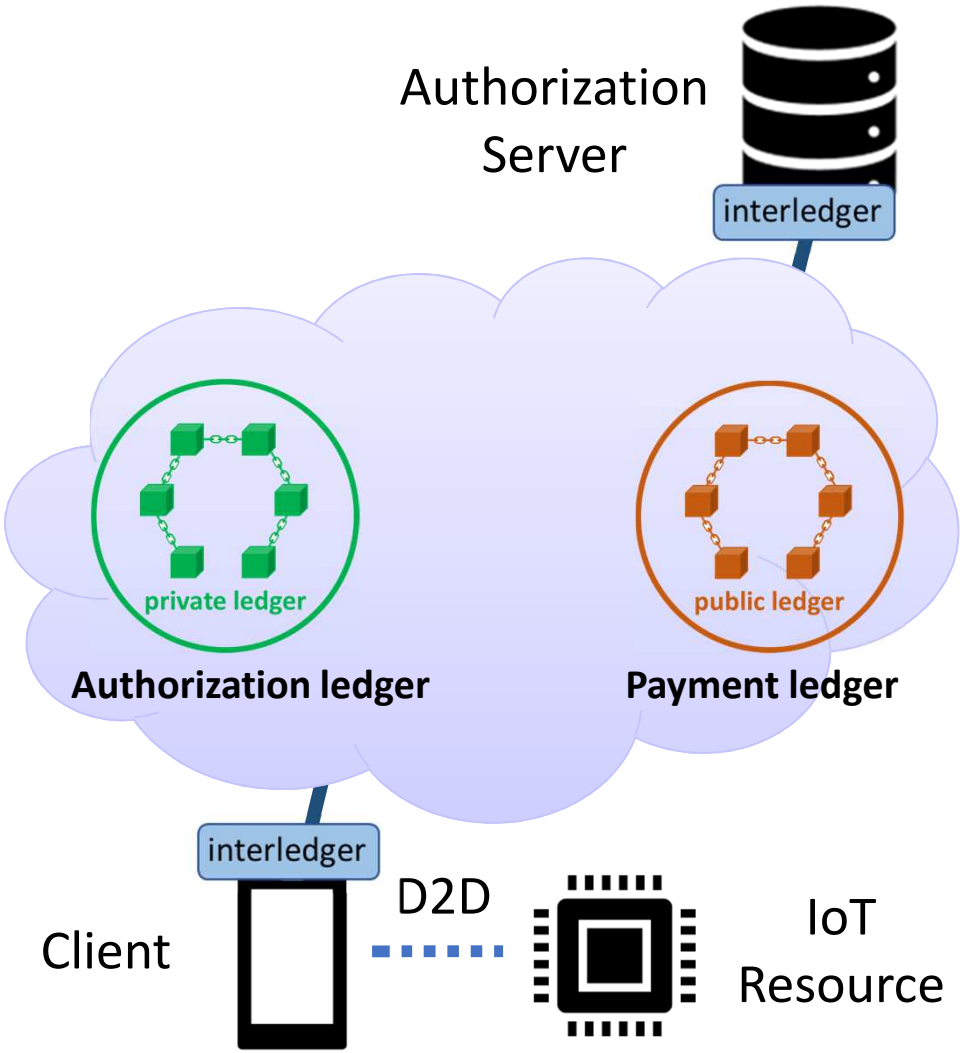
- Public ledger
  - High cost and high delay
  - Payments
- Private/permissioned ledger
  - Low/zero cost and low delay
  - Authorization functionality
- Interledger operation
  - Required: Atomicity of transactions on private and public ledgers
  - How? Hash-lock and time-lock contracts



vsiris@aueb.gr

Model 3: Combine public & private ledgers

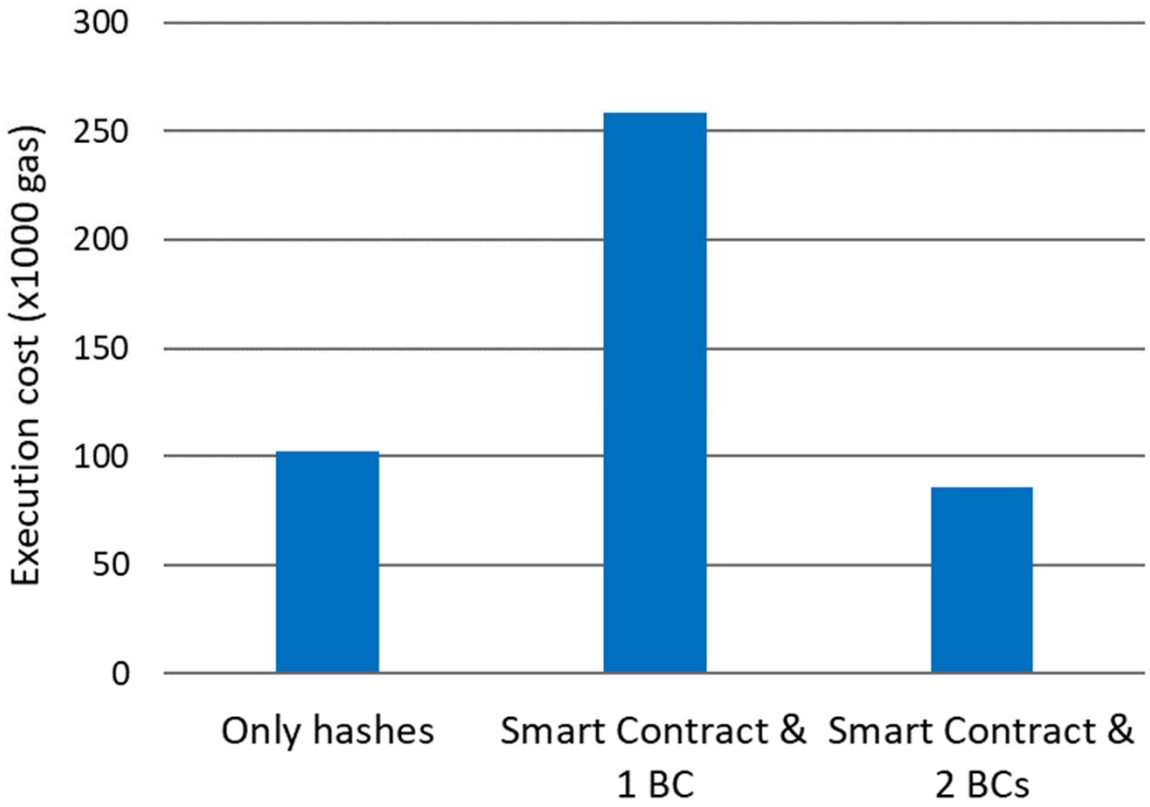
- Public ledger
  - High cost and high delay
  - Payments
- Private/permissioned ledger
  - Low/zero cost and low delay
  - Authorization functionality
- Interledger operation
  - Required: Atomicity of transactions on private and public ledgers
  - How? Hash-lock and time-lock contracts
  - Who? Client or AS (both incentivized)



vsiris@aueb.gr

Two blockchain results: execution cost

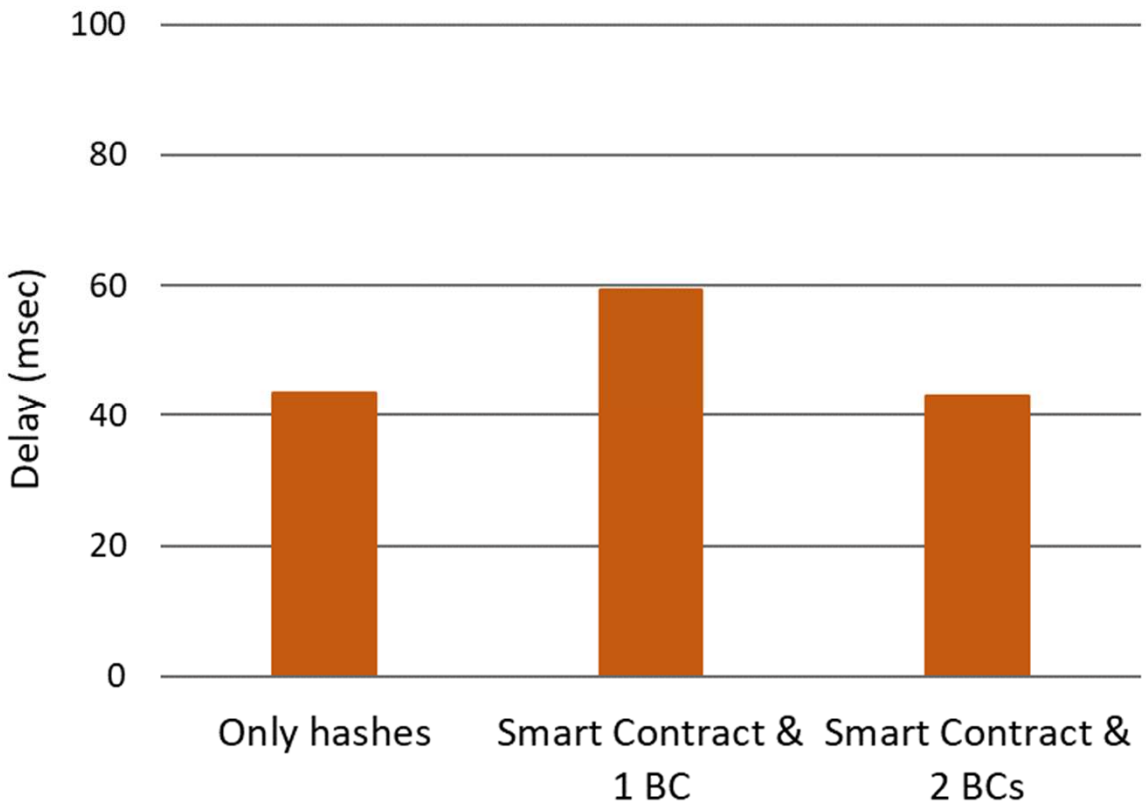
- Two blockchains achieve lower cost compared to one
  - Only payment transaction on public ledger
- Tradeoffs
  - Two ledgers: trust and transparency for authorization transactions determined by permissioned node set
  - Public ledger: wide-scale decentralized trust and transparency



vsiris@aueb.gr

Two blockchain results: delay

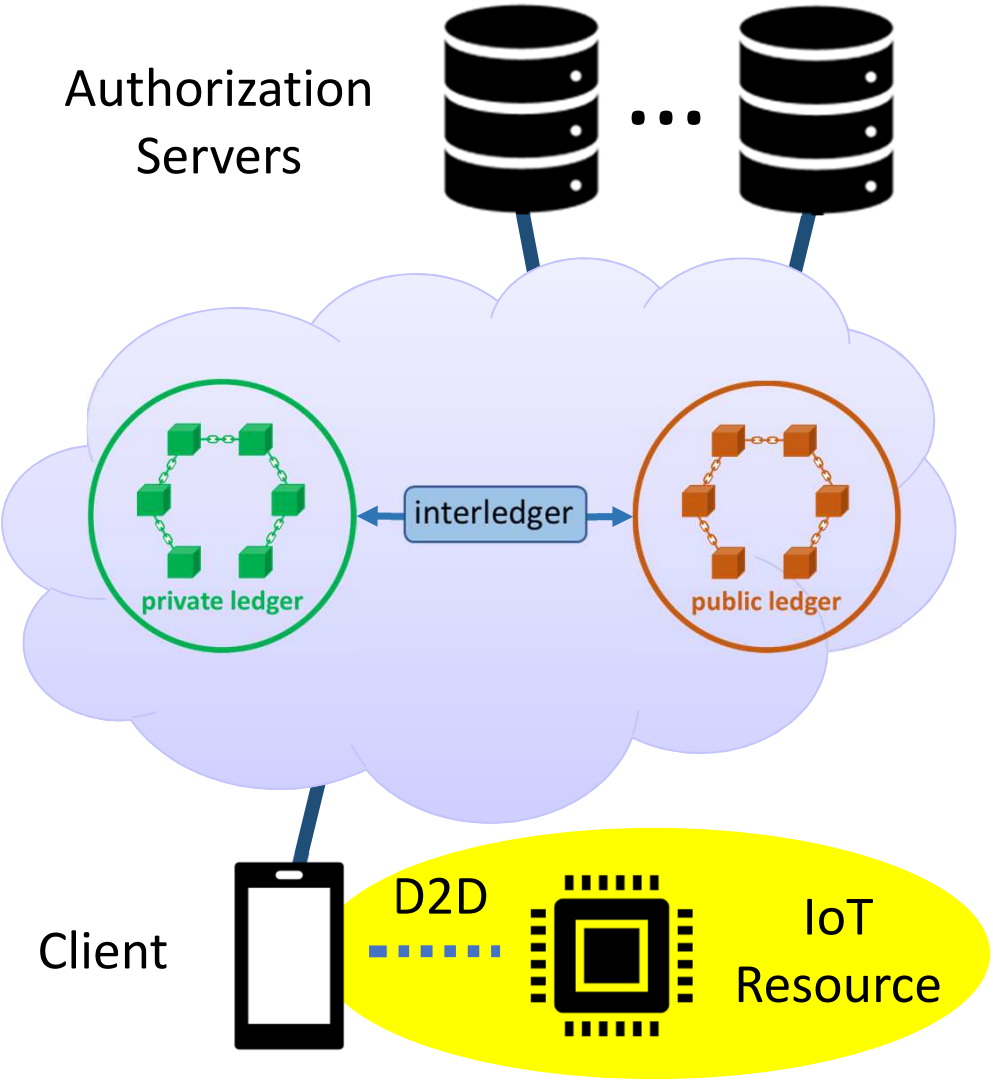
- Two blockchains achieve same delay as one blockchain recording only hashes
  - Only hashes & 2 blockchains: 3 transactions
  - 1 blockchain: 4 transactions



vsiris@aueb.gr

Achieving full end-to-end decentralization

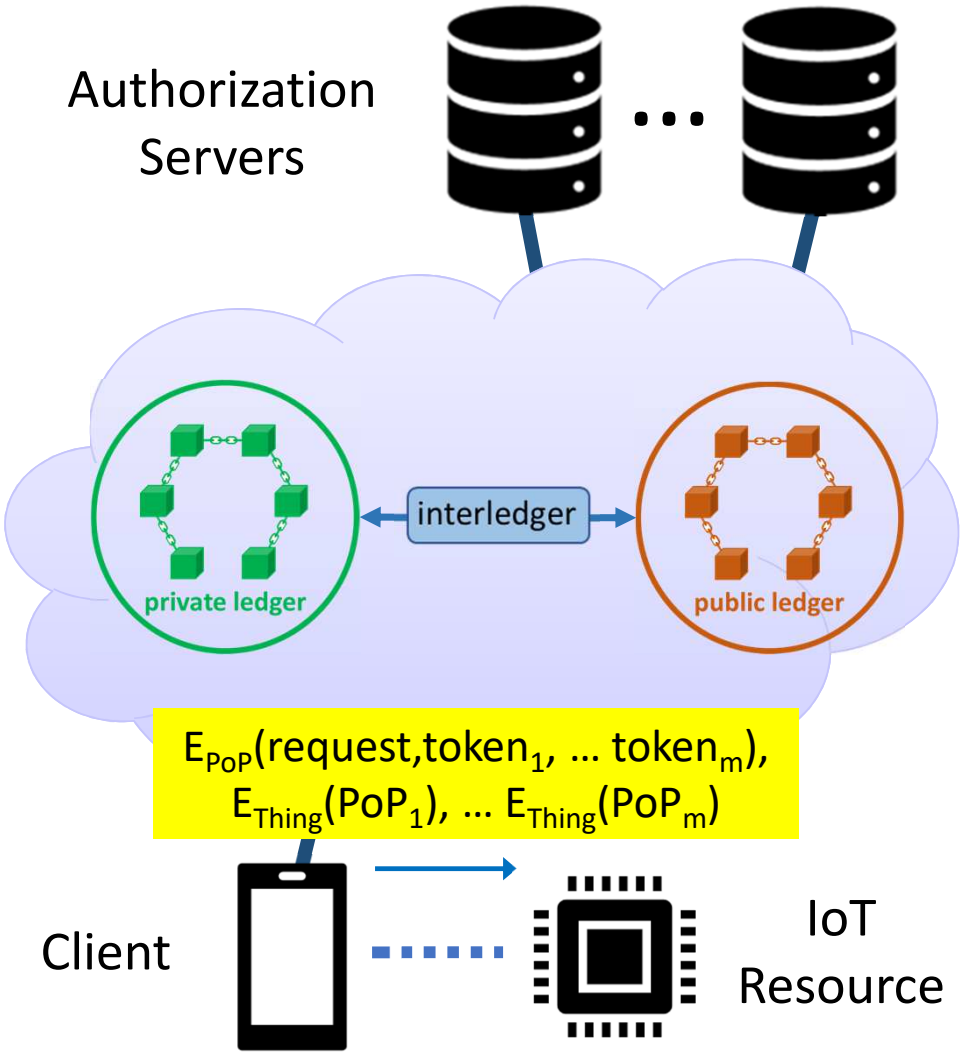
- Solution not fully decentralized
  - ***AS is single point of failure/attack***
- Cannot move AS functionality into blockchain
  - Increases reliability but not privacy
  - AS processes secret information
- Solely adding multiple ASes not enough
  - Can use threshold signatures between ASes and client
  - But, not an end-to-end solution



vsiris@aueb.gr

End-to-end decentralized authorization

- Assume n ASes each transmits its own Proof-of-Possession (PoP) key
- IoT resource requires m-out-of-n PoP keys
- Client and IoT resource XOR m PoP keys
  - PoP = PoP_1 XOR PoP_2 XOR … PoP_m
  - PoP used to secure the client-resource link
- But, still need to reduce amount of data sent to constrained IoT resource

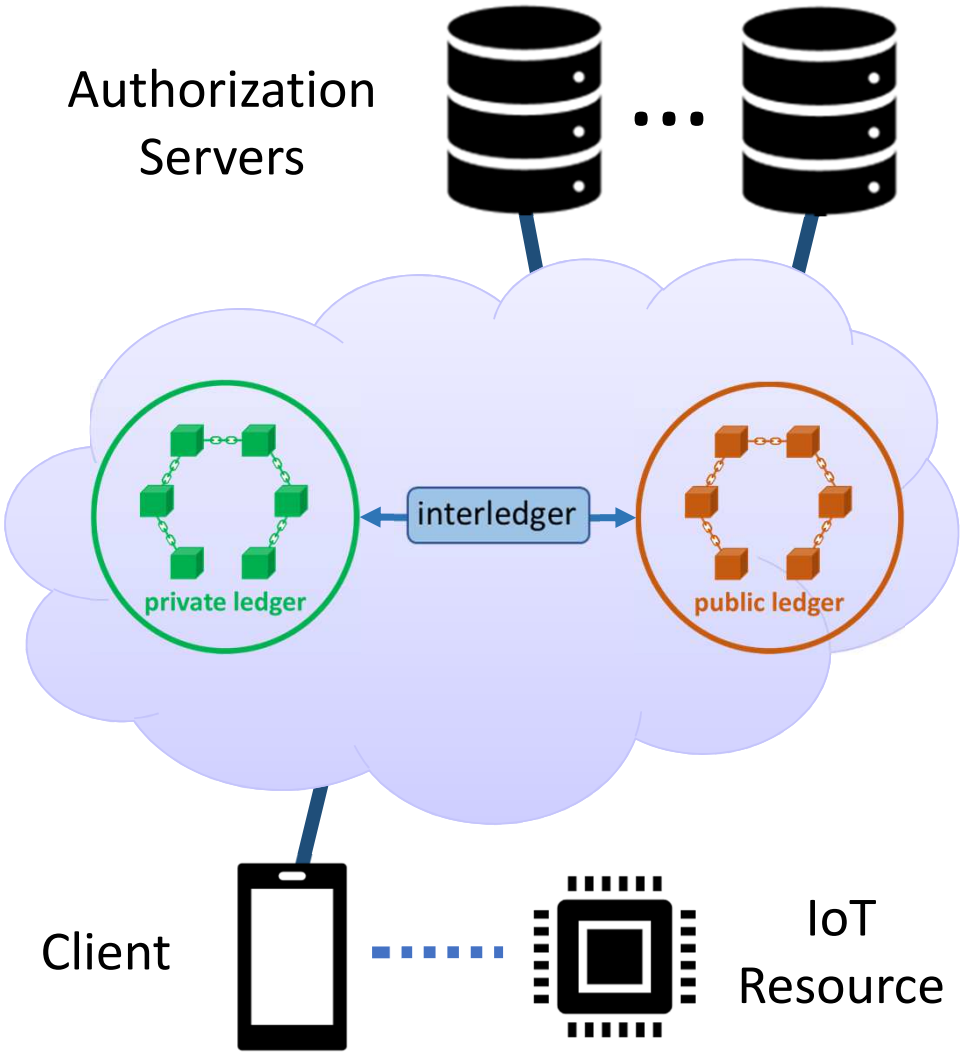


vsiris@aueb.gr

End-to-end decentralized authorization with data reduction

Two mechanisms to reduce data that client sends to IoT resource:

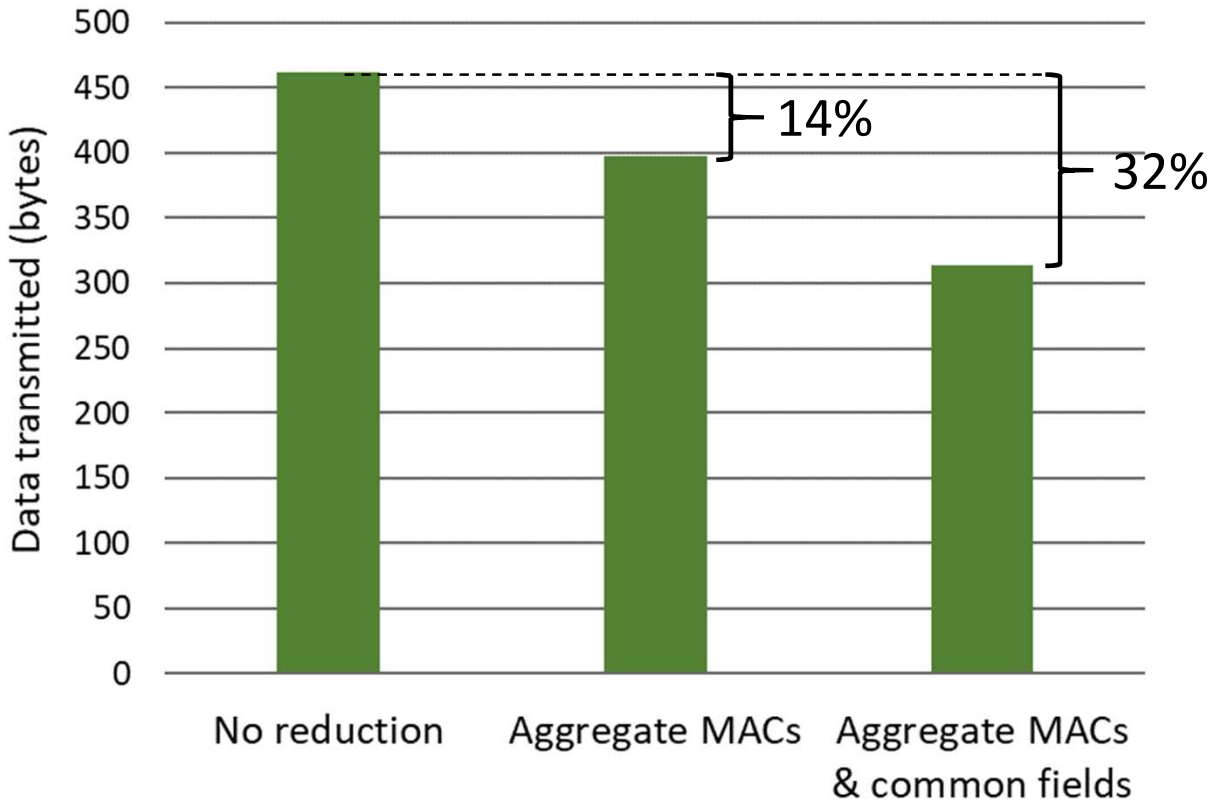
- Aggregate MACs: Client sends XOR of m MACs instead of m individual MACs
- Client sends common access token fields only once



vsiris@aueb.gr

Reduction of client-IoT resource data

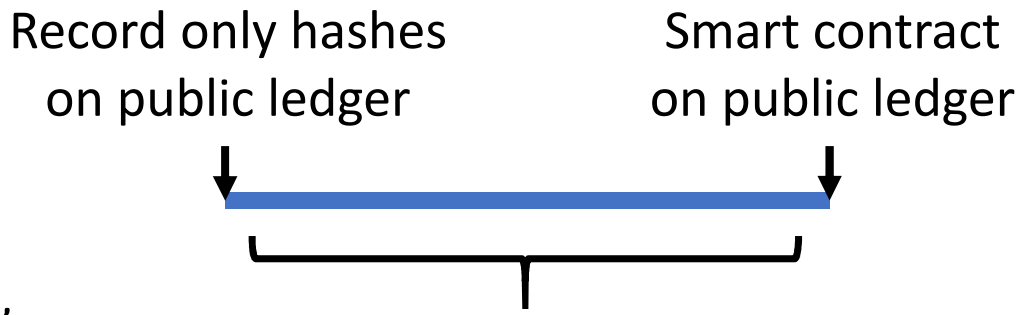
- Assume 3-out-of-n ASes
- Aggregate MACs: 14% reduction
- Common fields: 18% reduction
- Together: 32% reduction
- Gains for more ASes will be higher

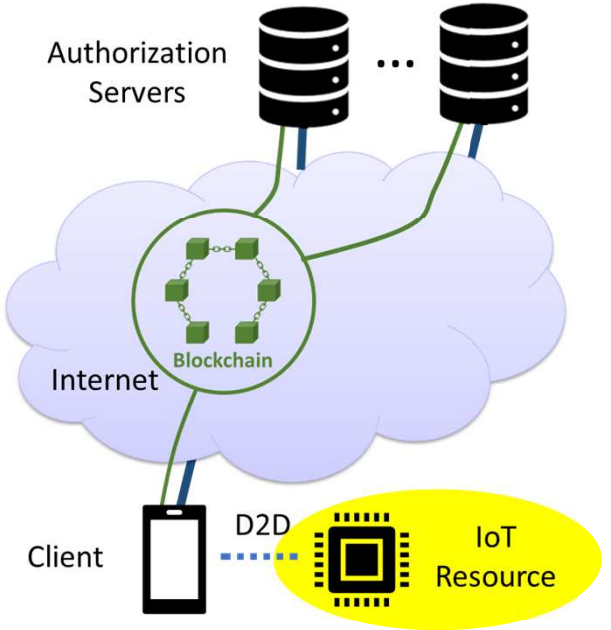


vsiris@aueb.gr

Conclusions

- 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
  - 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

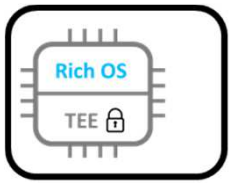




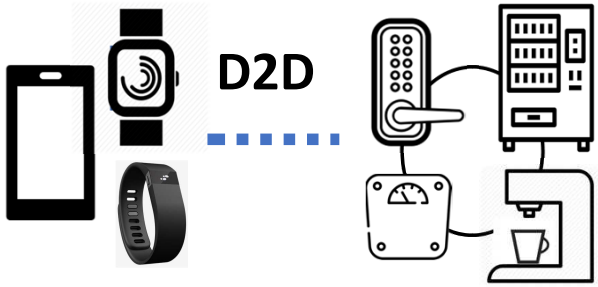
vsiris@aueb.gr

Challenges & ongoing work (cont)

- Trust that resource indeed provides access
  - Trusted Execution Environments (TEEs) such as ARM's TrustZone, Intel's SGX, Keystone (open source RISC V)
- Constrained clients
  - Need client proxy/agent (analogous to AS acting as proxy of IoT resource)



IoT resource with TEE



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

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

"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

SOFIE project

- EU Horizon 2020 funded project
- 1/1/2018 – 31/12/2020
- €4.5M

10 Partners

- Aalto University, Ericsson, Rovio (Finland)
- Guardtime (Estonia)
- AUEB, Synelixis, Optimum (Greece)
- Eng, Asm Terni Spa, Emotion Srl (Italy)

http://www.sofie-iot.eu/