# OAuth 2.0 meets Blockchain for Authorization in Constrained IoT 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

IEEE 5th World Forum on Internet of Things
15-18 April 2019, Limerick, Ireland

EU H2020 SOFIE: Secure Open Federation for Internet Everywhere

# Motivation and goal

- Why constrained IoT environments ?
- Why (not) blockchains ?
- Two proposals for integrating blockchains with authorization to constrained IoT devices with different cost/functionality tradeoffs
- First step in identifying next challenges
  - Transaction cost and delay
  - Fully decentralized solution
  - Ensuring that IoT devices actually provide promised access

*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

Reduction also for *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

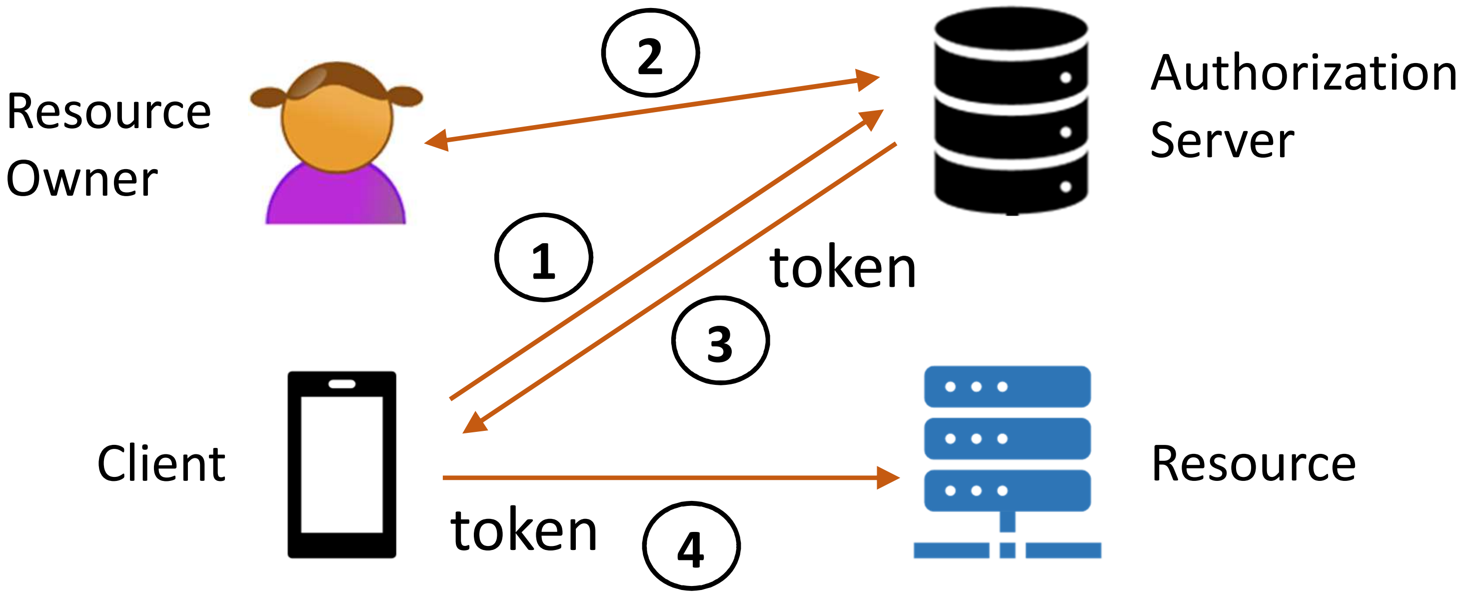
# Why blockchains? Blockchain features

- **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

# Problem OAuth 2.0 addresses

- OAuth 2.0 Authorization Framework: RFC 6749 (10/2012)
- How can client obtain access to a protected resource?
  - Authorization offloaded to separate entity (Authorization Server)
  - With resource owner's consent
  - Based on access tokens




vsiris@aueb.gr

# OAuth 2.0 assumptions

- Client, Resource, Authorization Server, and Resource Owner are
  - always connected and online
  - resource capable
- ACE (Authentication and Authorization for Constrained Environments) IETF Working Group tries to address above issues by adding
  - CoAP versus HTTP
  - More *efficient encoding*: CBOR binary versus JSON-based JWT
  - *Symmetric* versus public/private for *self-contained access tokens*
  - *Proof-of-Possession (PoP)* key together with access token
  - Authorization based on *resource owner policies*

# Benefits from utilizing blockchain for authorization

- 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

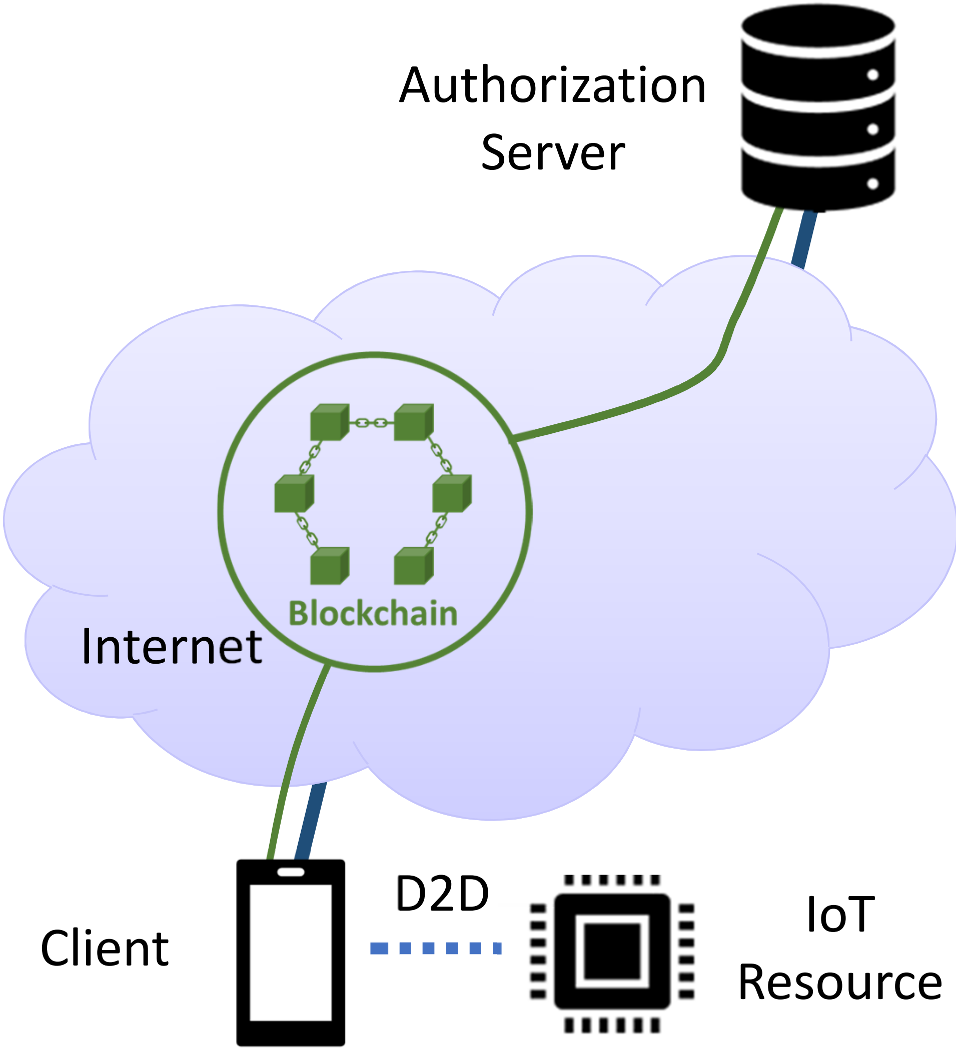
- Transparent and trusted execution of authorization logic
  - More expressive than above
  - Policies can involve IoT events recorded on blockchain
  - Can benefit from blockchain's high availability
  - But more expensive

Model 2: Smart contract handling authorization requests and encoding policies

vsiris@aueb.gr

# Assumptions

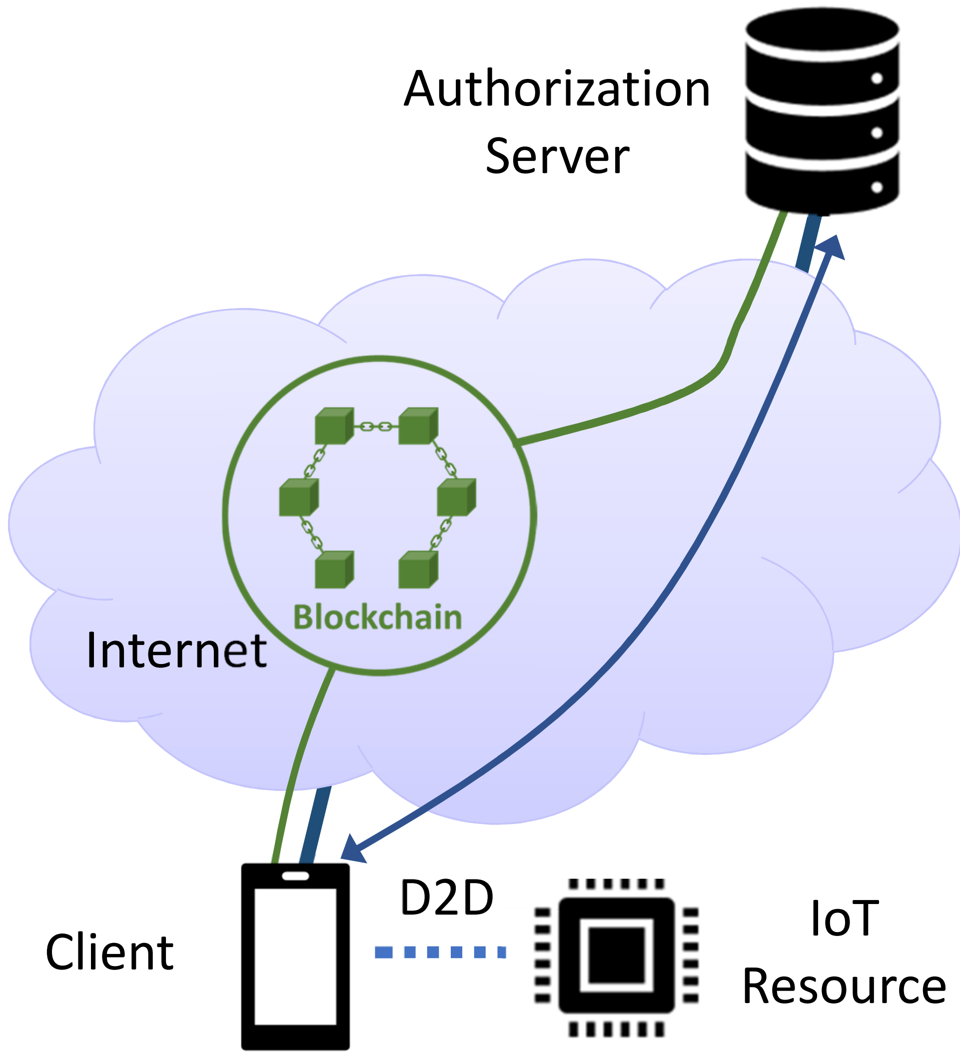
- IoT resource has limited processing, storage and only D2D connectivity
- Authorization Server handles requests on behalf of IoT resource
- Client and AS always connected and can interact with blockchain




vsiris@aueb.gr

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

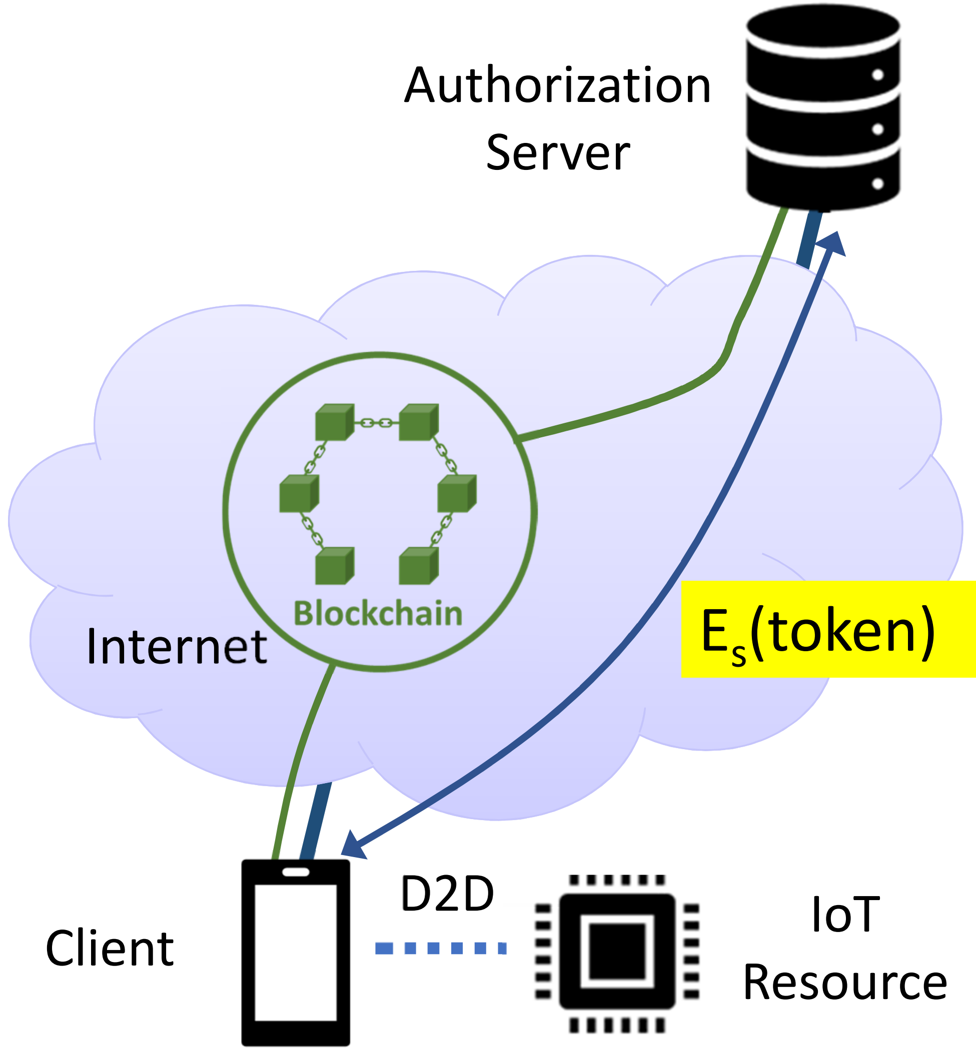
- Client and AS communicate directly as in OAuth 2.0




vsiris@aueb.gr

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

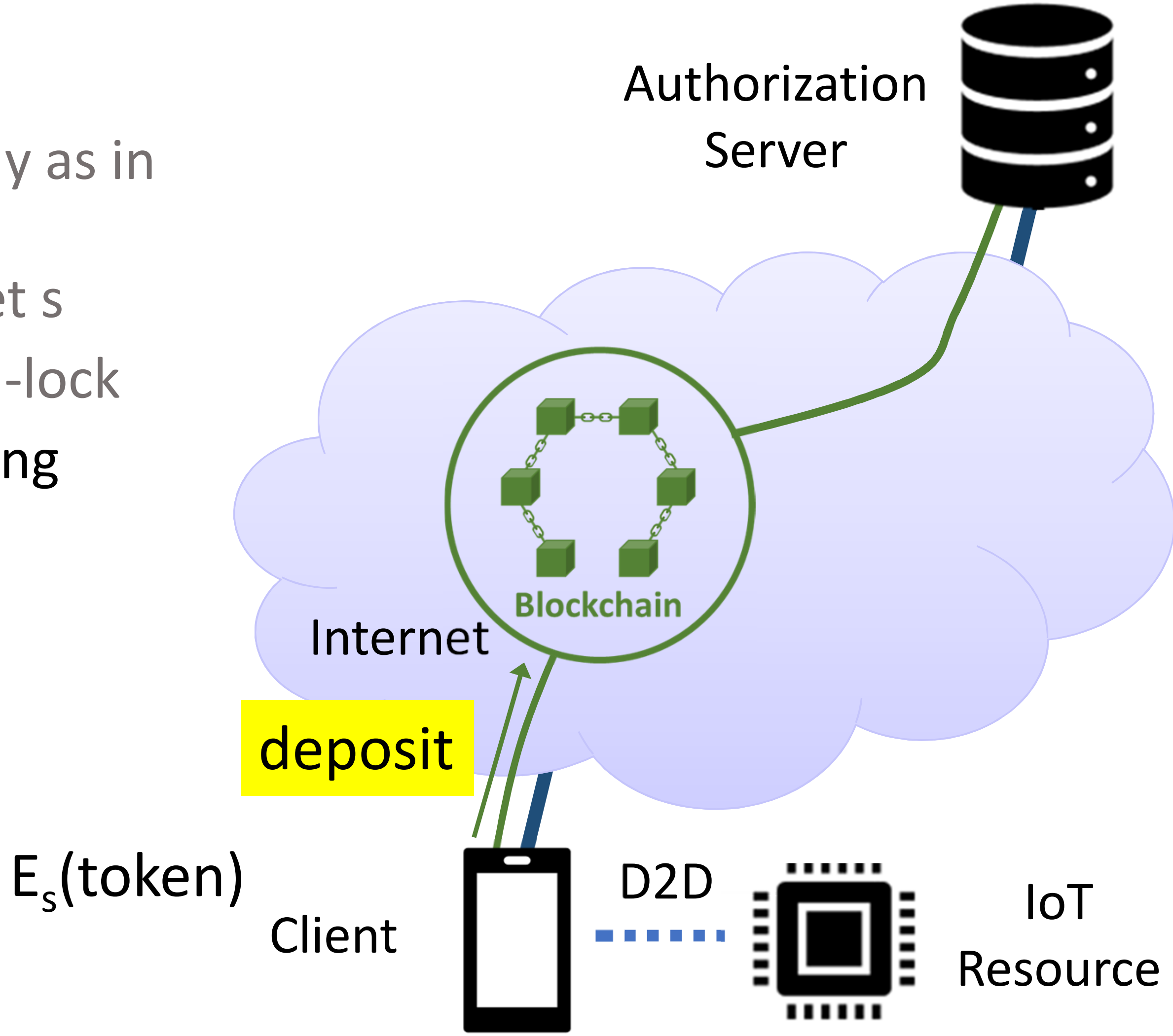
- Client and AS communicate directly as in OAuth 2.0
- Access token encrypted with secret s
- Secret s related to payment's hash-lock




vsiris@aueb.gr

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

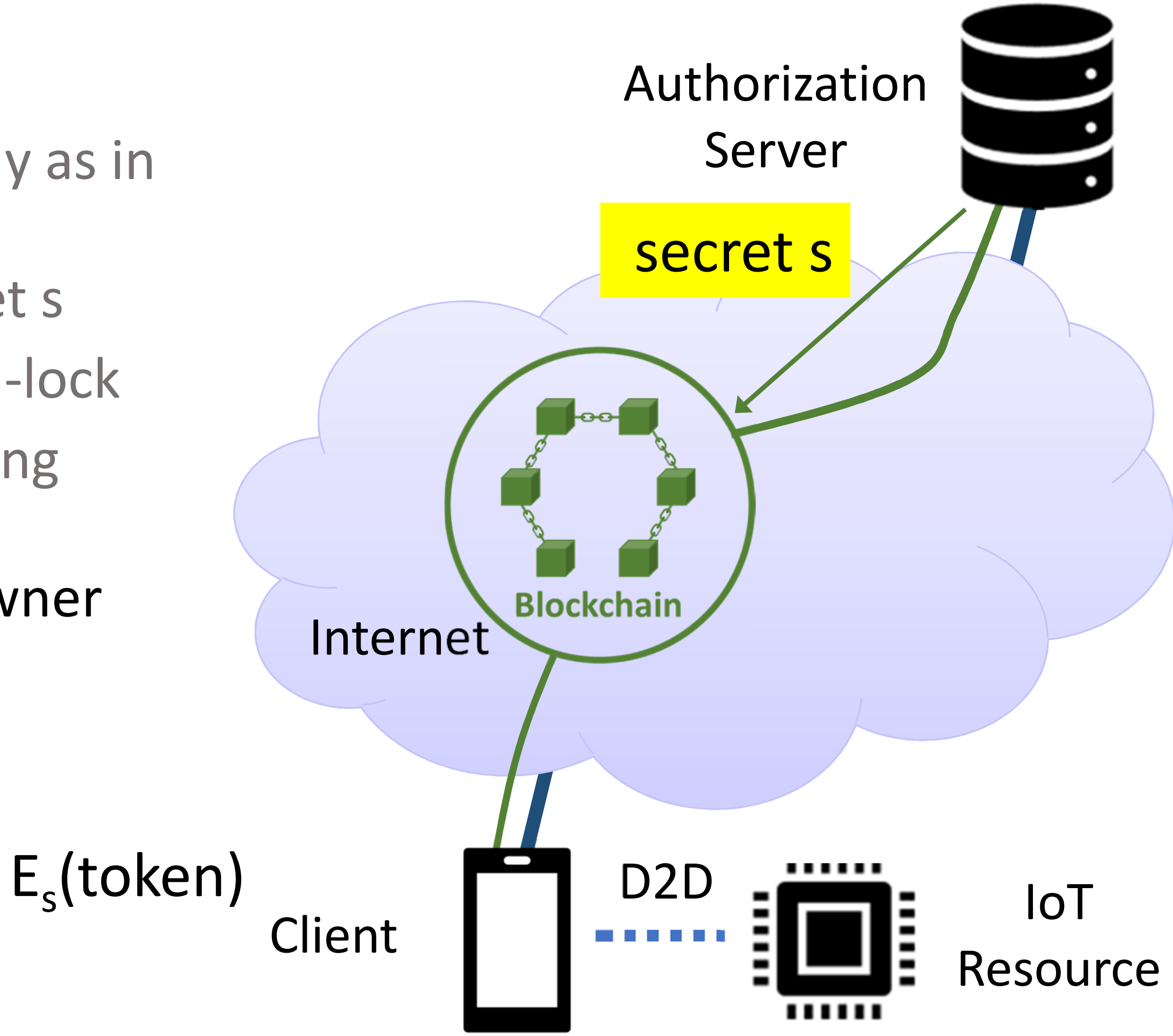
- Client and AS communicate directly as in OAuth 2.0
- Access token encrypted with secret s
- Secret s related to payment's hash-lock
- Client deposits amount for accessing resource




vsiris@aueb.gr

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

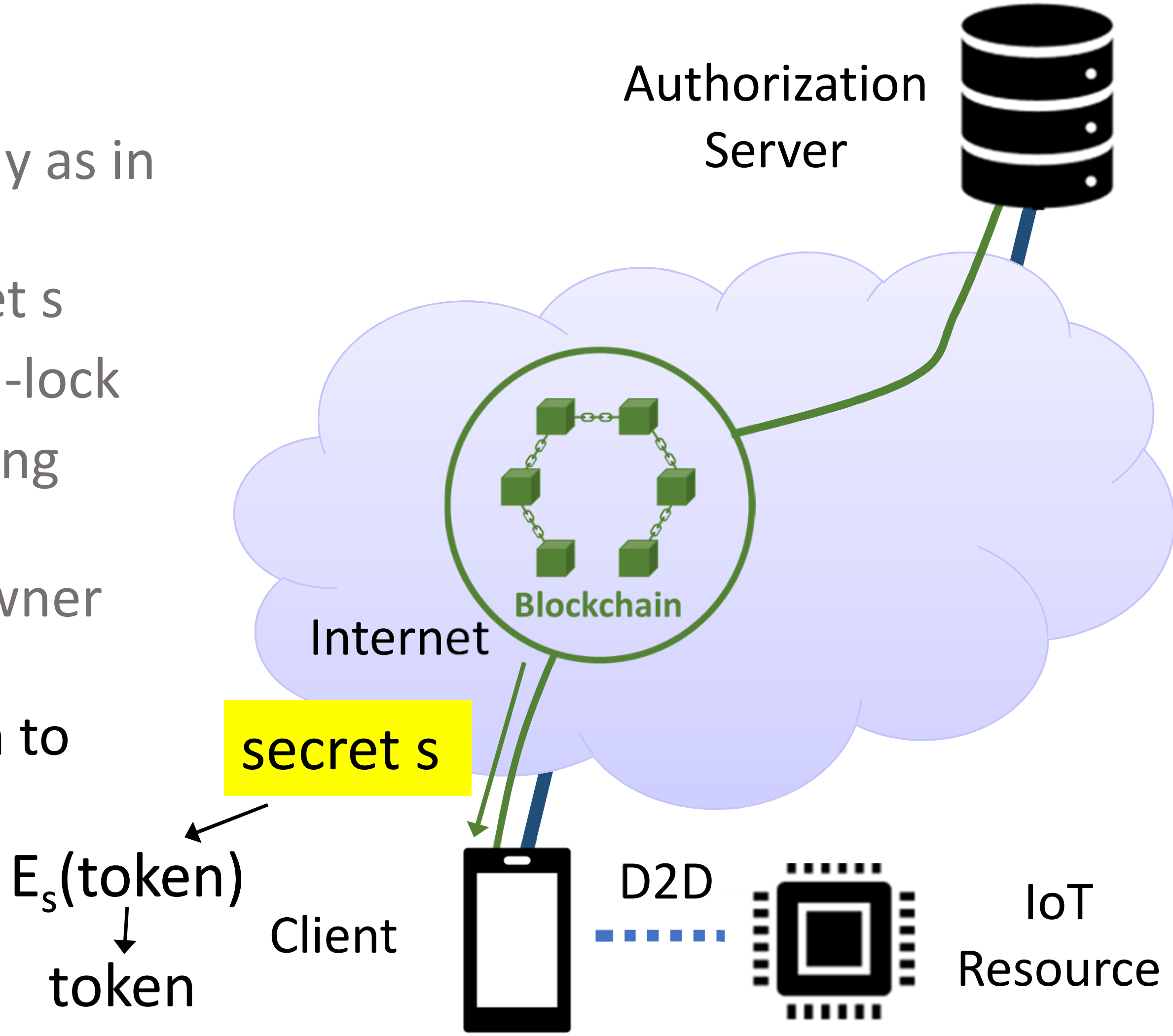
- Client and AS communicate directly as in OAuth 2.0
- Access token encrypted with secret s
- Secret s related to payment's hash-lock
- Client deposits amount for accessing resource
- Deposit transferred to resource owner when s revealed on blockchain




vsiris@aueb.gr

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

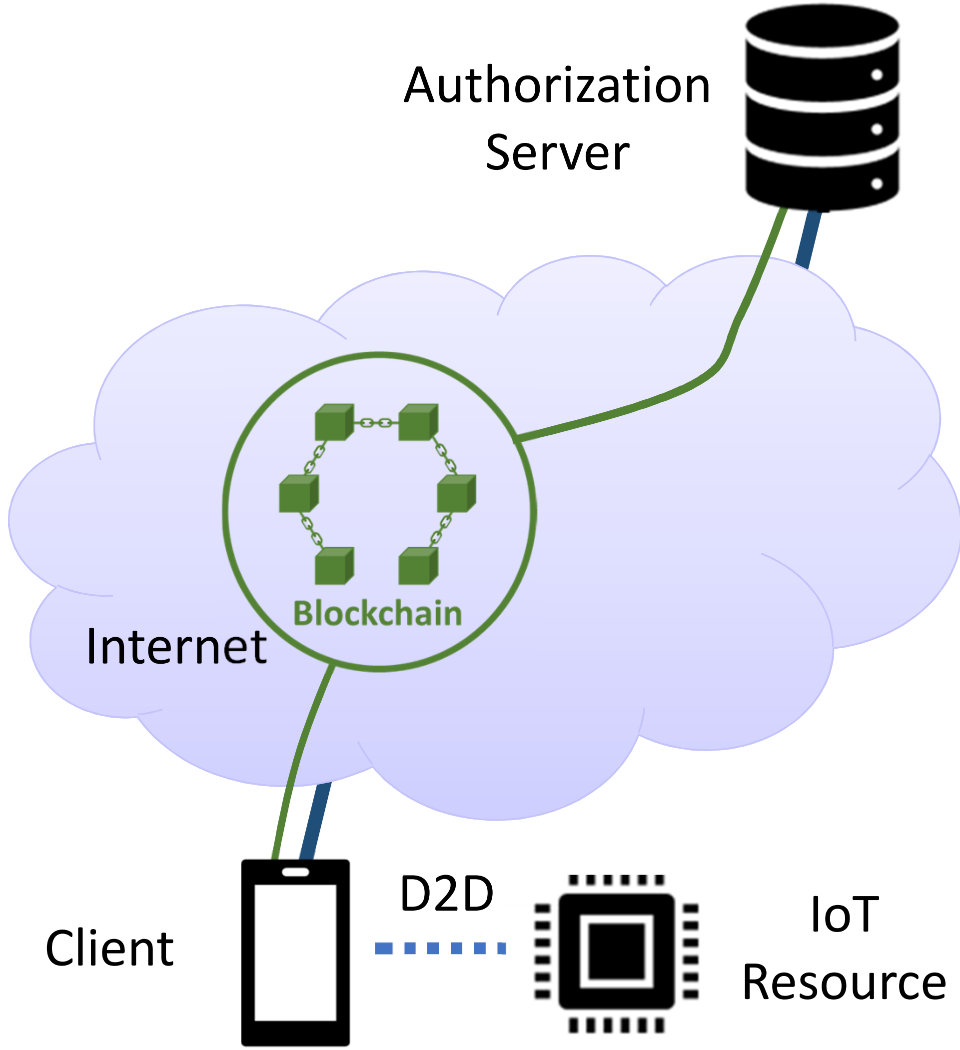
- Client and AS communicate directly as in OAuth 2.0
- Access token encrypted with secret s
- Secret s related to payment's hash-lock
- Client deposits amount for accessing resource
- Deposit transferred to resource owner when s revealed on blockchain
- Client reads secret s on blockchain to decrypt access token




vsiris@aueb.gr

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

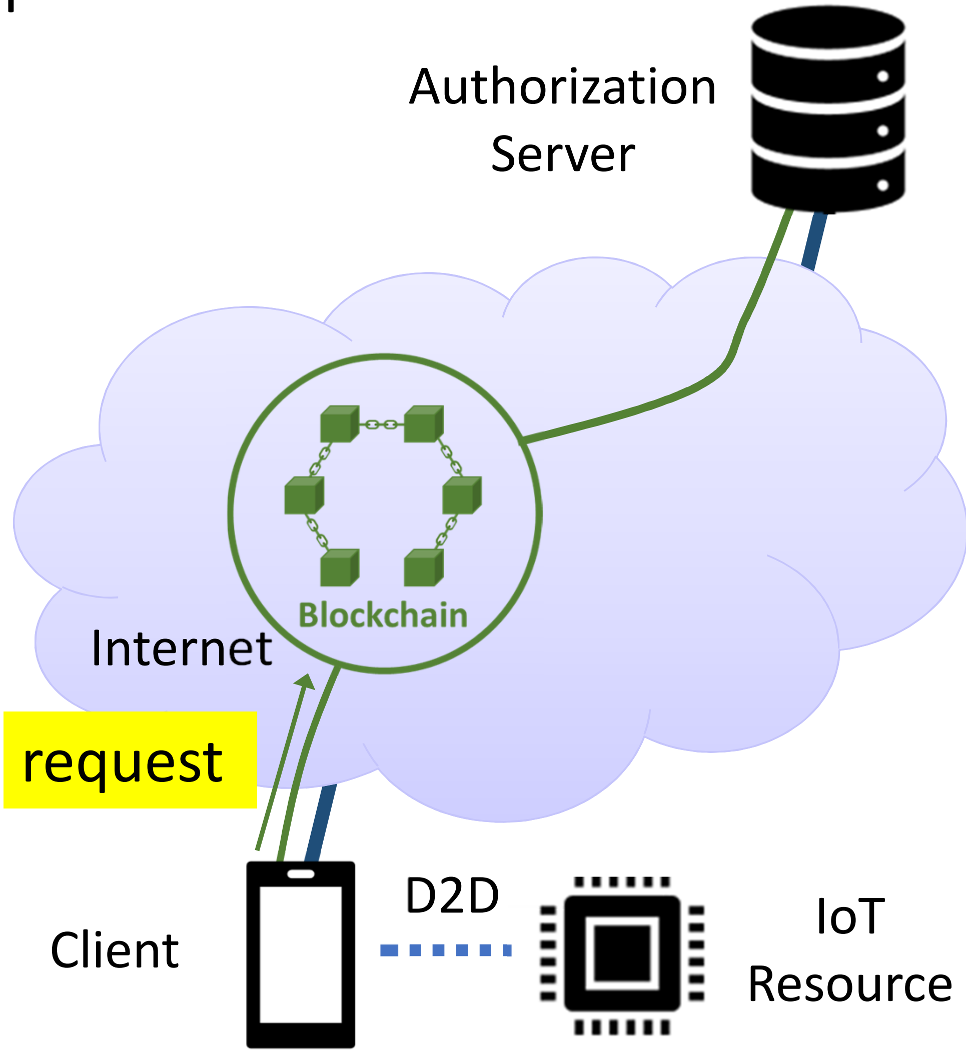
- Client and AS communicate directly as in OAuth 2.0
- Access token encrypted with secret s
- Secret s related to payment's hash-lock
- Client deposits amount for accessing resource
- Deposit transferred to resource owner when s revealed on blockchain
- Client reads secret s on blockchain to decrypt 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

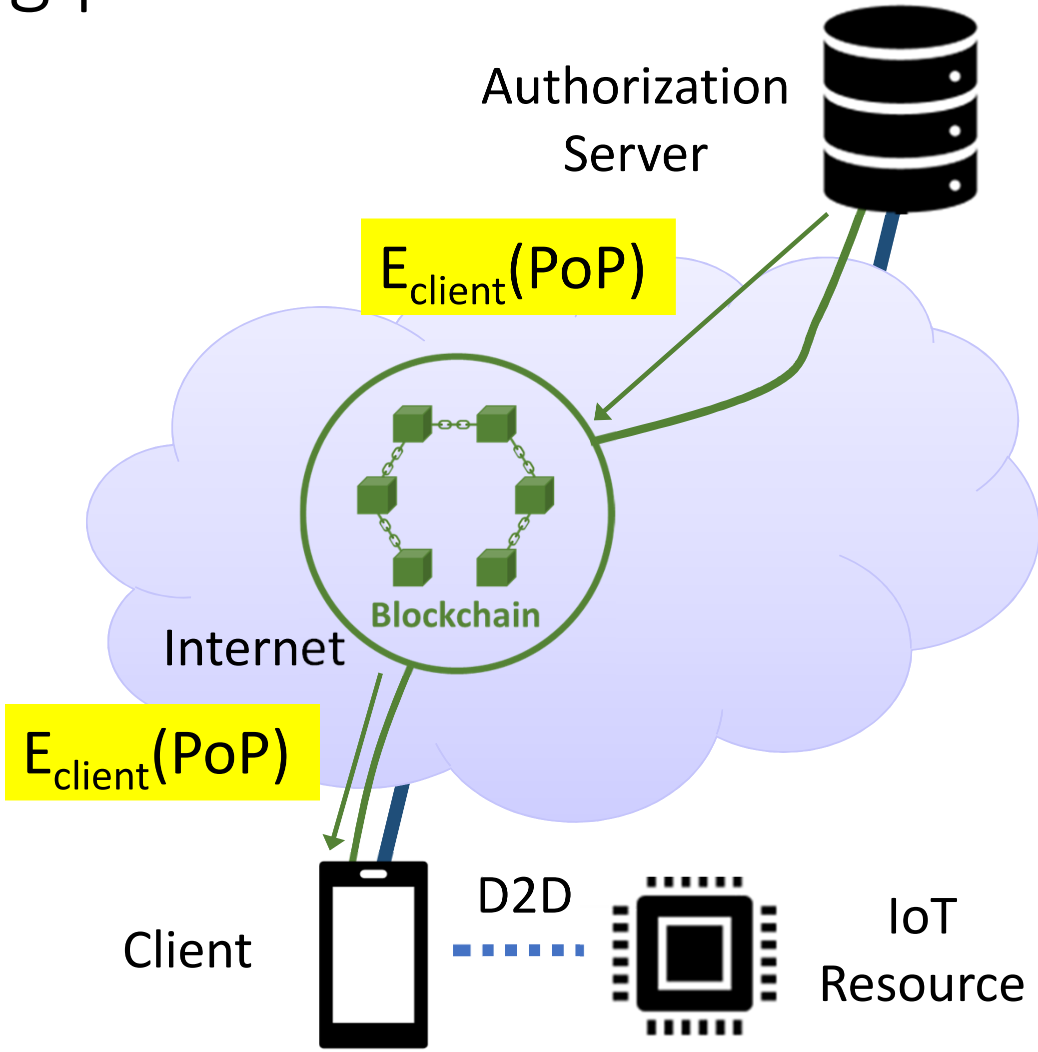
- Client sends authorization request to Smart Contract




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



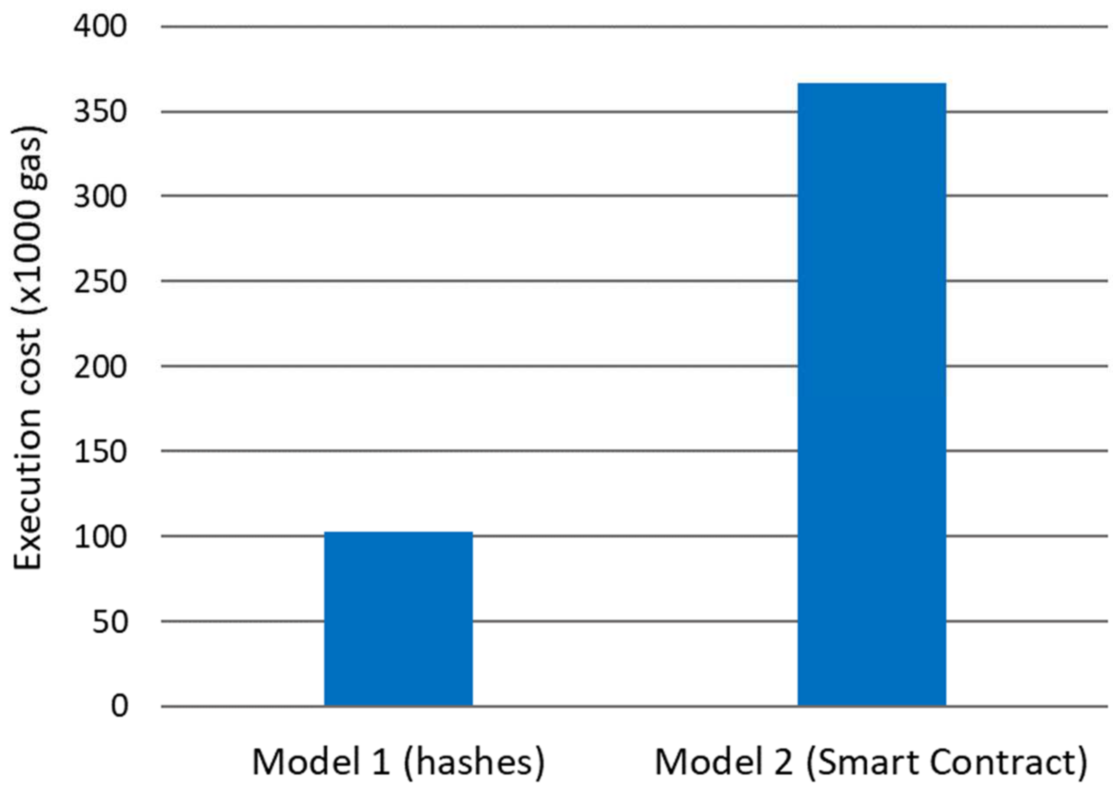

vsiris@aueb.gr

# Implementation

- Deployed local node connected to Rinkeby public Ethereum testnet
- Smart contract written in Solidity with Remix web-based editor
- Web3.0 to interact with Rinkeby blockchain
- Authorization server based on open PHP implementation of OAuth 2.0

# Results: execution cost

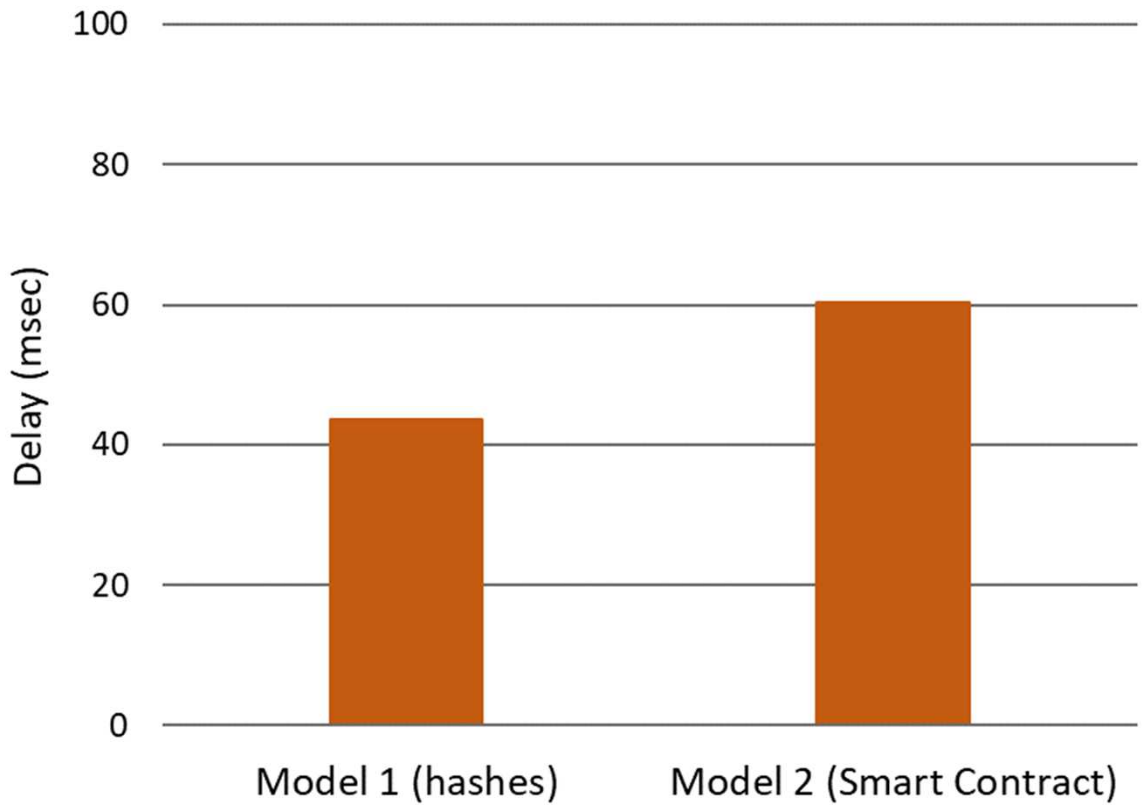
- Smart contract requires almost 3 times EVM gas compared to simply recording hashes
- Only write transactions cost gas
  - Reading data has zero cost
- Quantifies cost for higher functionality of smart contracts
  - Authorization policies & logic




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

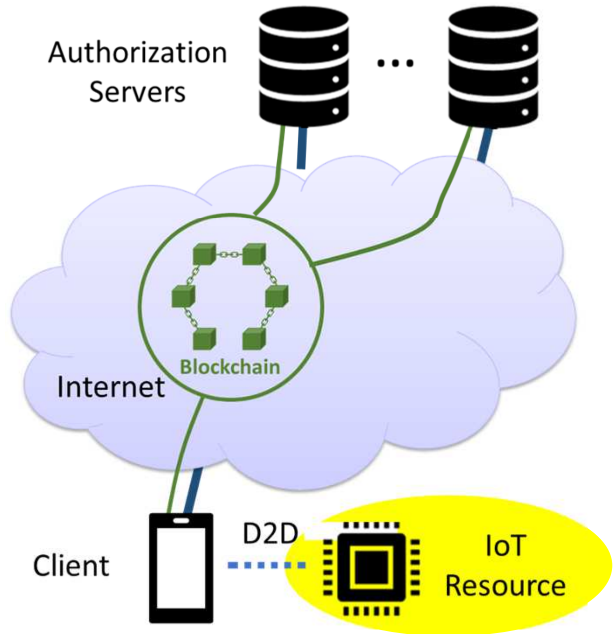
# Challenges & ongoing work

- High cost & delay
  - Due to public ledger
  - Combining public & private/permissioned ledgers can provide different tradeoffs of cost, trust, and privacy
- 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 is not a solution because IoT resource not directly connected to blockchain

Record only hashes on public ledger

Smart contract on public ledger

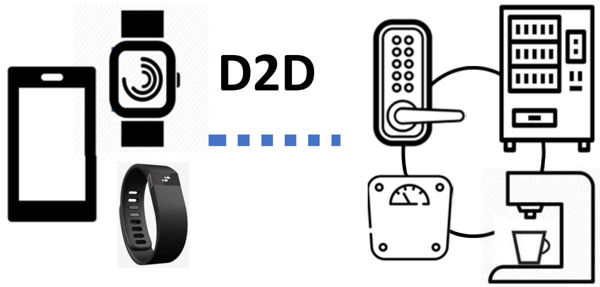
Achieved by combining public with private/permissioned ledger




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)


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/