ÉTS

Le génie pour l'industrie

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

Université du Québec

fuseelab.github.io

Research Directions of FUSÉE Lab

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE, MONTRÉAL, CANADA

UNIVERSITY OF QUÉBEC

CREDENCE WORKSHOP 2019

ÉTS Montréal

Engineering school in Montréal, QC, Canada
- 5 departments

Department of Software and IT Engineering
- Two streams: Soft Eng. and IT Eng.
- Masters: course-based and thesis-based
- PhD program

Constituent of the University of Quebec
- Provincial network of public universities

French is the language of instruction
- English is the language of research
- Agreement with other Montreal universities for courses

Focus: industrial research
- Emphasis on partnership with local companies
- Startup incubator: CenTech

fuseelab.github.io

FUSÉE Laboratory

Established in 2017

- Fast, unified, scalable: event processing and event messaging

3 domains of research:

- Practical blockchain & DLT
- Expressive publish/subscribe middleware
- Networked game engines
- 1 Postdoc, 4 PhDs, 9 Masters

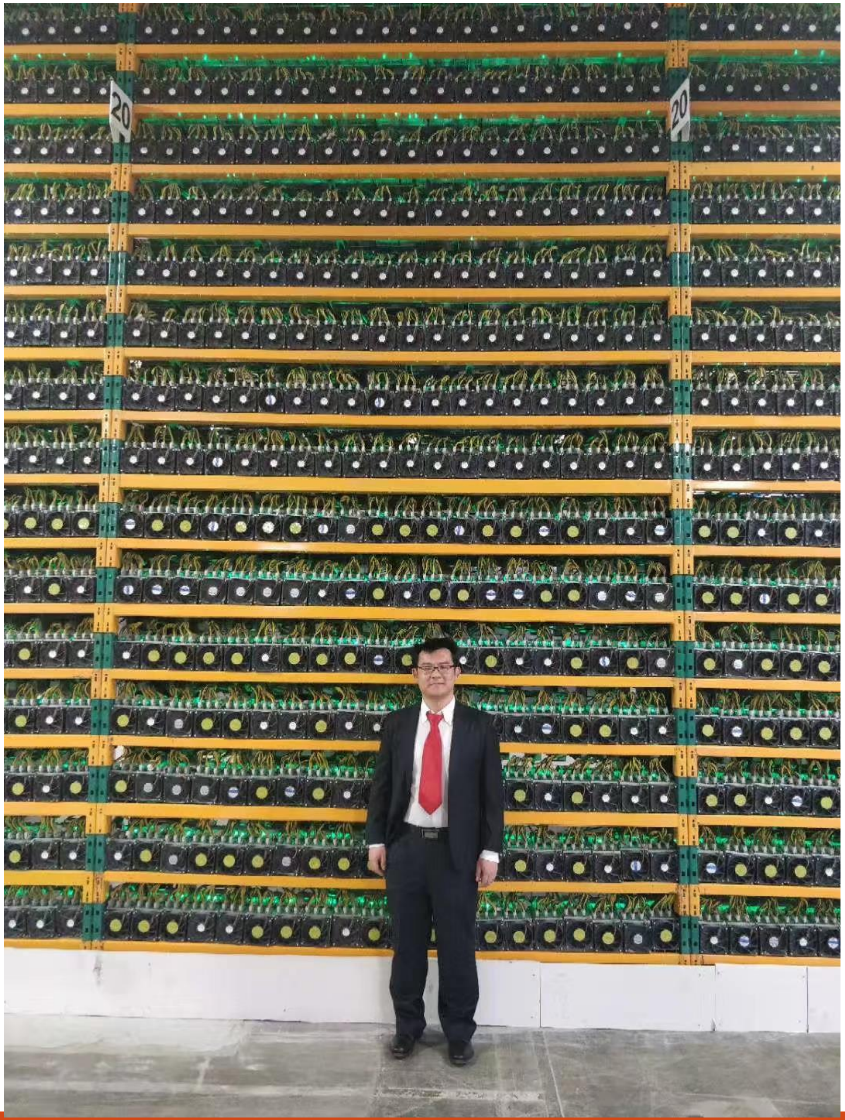
Teaching 3 courses in French:

- Foundations of distributed systems (undergrad.)
- Middleware and distributed applications (undergrad.)
- Decentralized applications and systems (grad.)

Website: http://fuseelab.github.io

Current role of Quebec in crypto?

Cryptocurrency mining

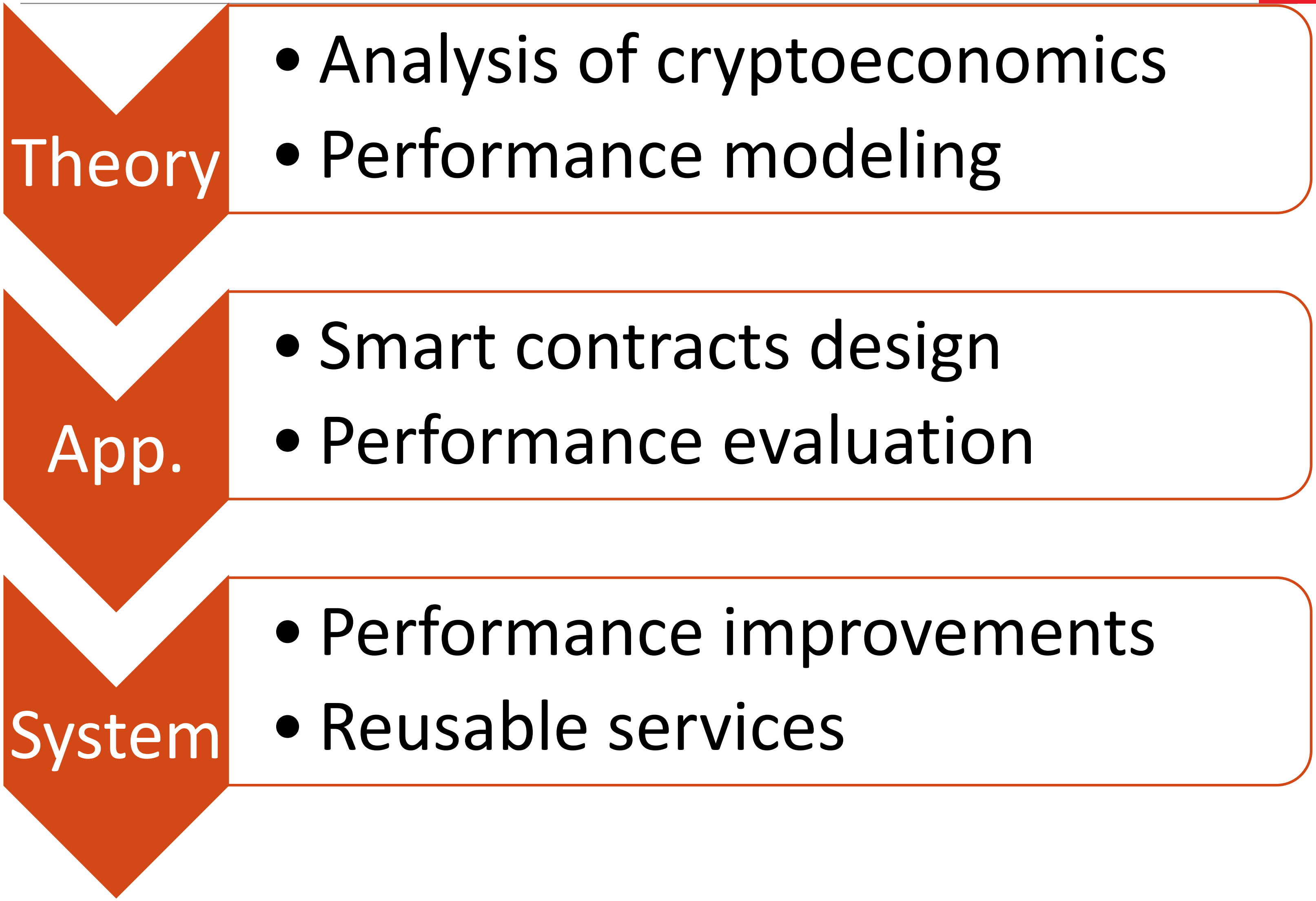


Bitfarms™

Main objective for blockchain

"Demonstrate the ***applicability*** and improve the ***utility*** of distributed ledger technologies (DLTs) for a wide variety of ***future applications***, primarily accomplished by delivering technical innovations to raise the ***performance and scalability*** of core blockchain systems"

Overview of blockchain research



Theory

- Analysis of cryptoeconomics
- Performance modeling

App.

- Smart contracts design
- Performance evaluation

System

- Performance improvements
- Reusable services

Research projects

RESEARCH COLLABORATION OPPORTUNITIES

GDPR-compliant data collection

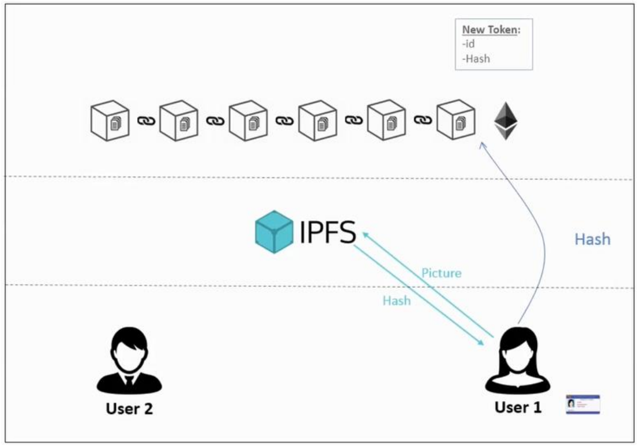
Target domains: IoT sensors, mobile data, model training

Data tokenization model

- Adaptation of ERC-721 (non-fungible tokens) and UTXO
- Record consent, access control, and data integrity

Smart contract implementation

- Solidity for Ethereum
- C# for Hyperledger Fabric



Off-chain data storage

- Integration with IPFS, MongoDB
- Challenge: how to support *right of erasure*?

Data availability problem

Sharding in Ethereum 2.0: Serenity

- Idea: Split into 1000+ public shard chains
- On-boarding of validators using Proof-of-Stake
- Problem 1: Requires constant shuffling of validators
- Problem 2: Requires constant synchronization of shard data

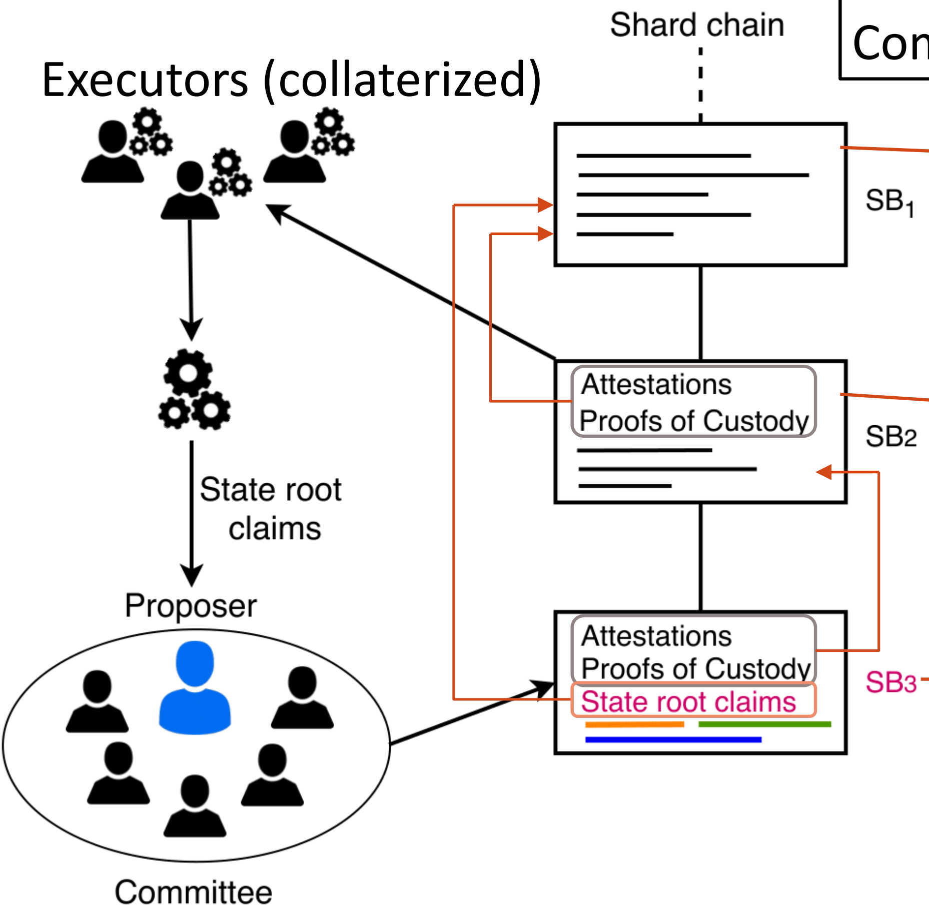
Solution: Stateless consensus

- Problem 3: Requires guarantees and incentives for all shard data to **stay available**

Sel et al. Towards Solving the Data Availability Problem for Sharded Ethereum. SERIAL 2018.

Delayed state execution

Executors are shuffled every 3 days
Committee shuffled every slot (24 seconds)

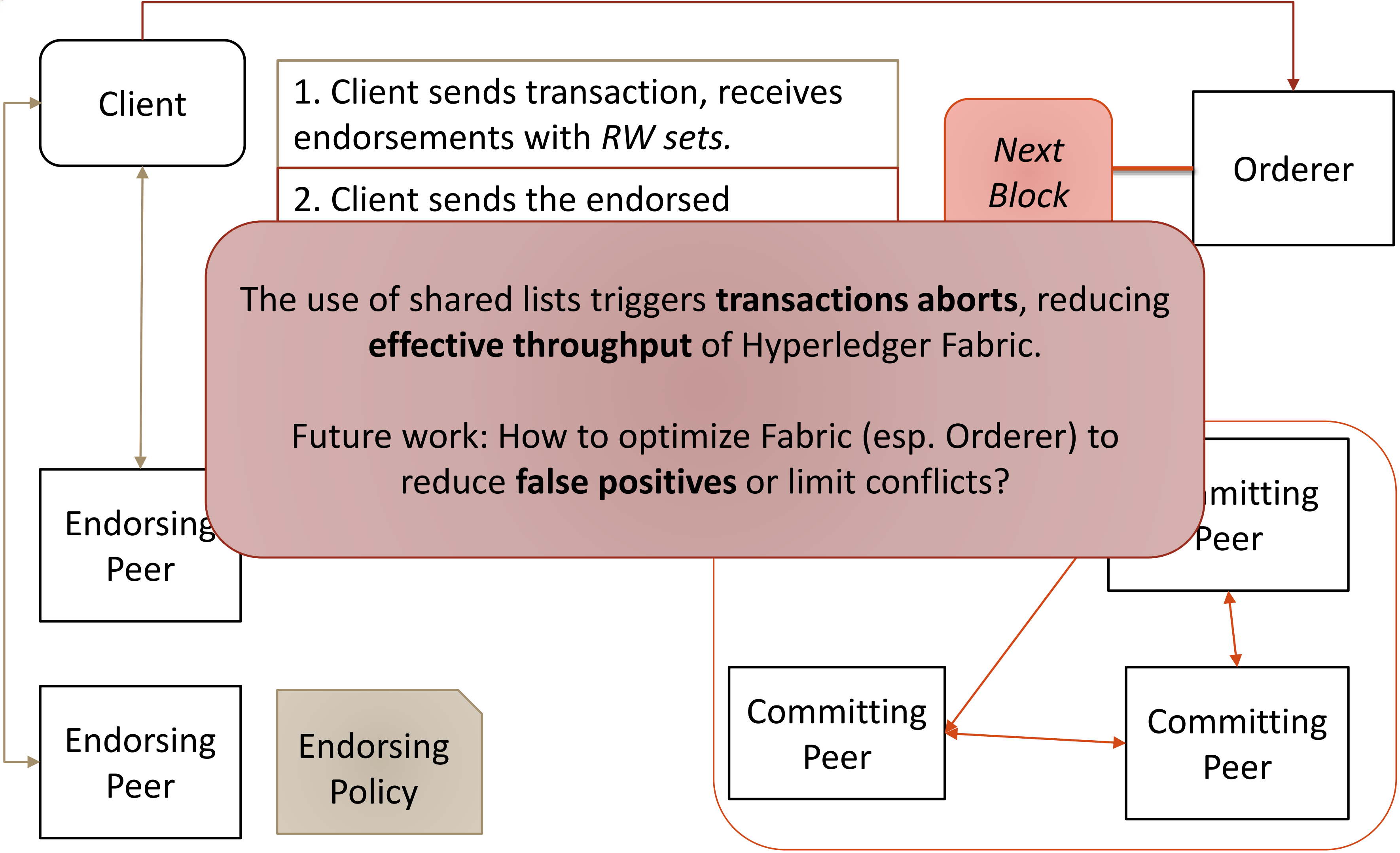


Contains final transaction order, aggregated from executors’ suggestions

Attestations contain Proofs of Custody that committee will keep transaction data from SB1 for 1 month

State root claims contain final state root hash for SB1, in case of conflict, request witness data and slash offenders

Hyperledger Fabric: MVCC



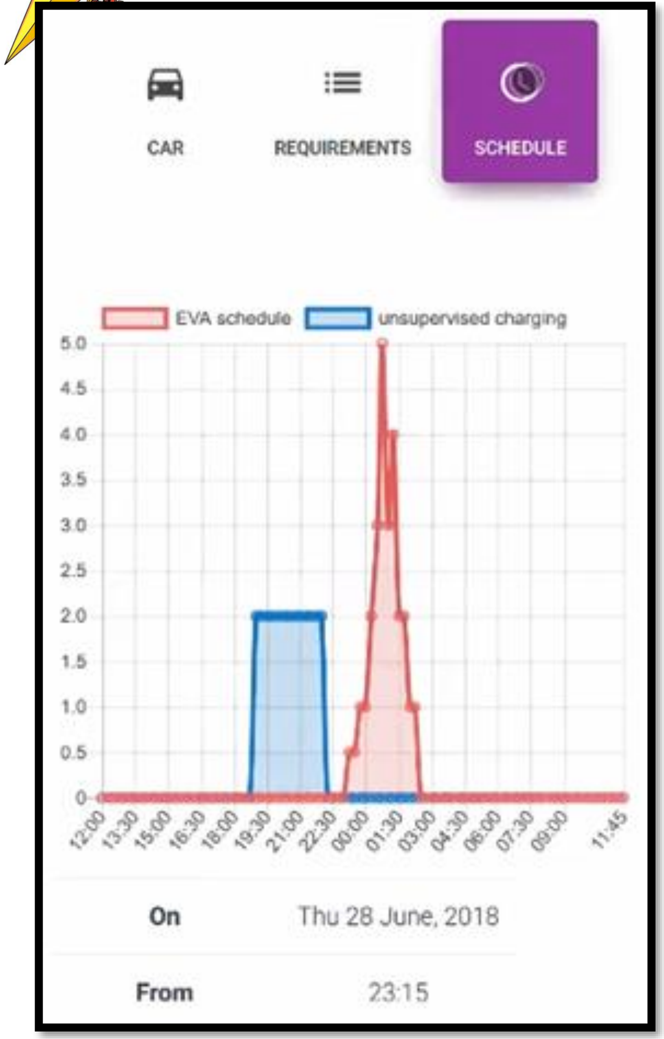
1. Client sends transaction, receives endorsements with *RW sets.*

2. Client sends the endorsed

The use of shared lists triggers **transactions aborts**, reducing **effective throughput** of Hyperledger Fabric.

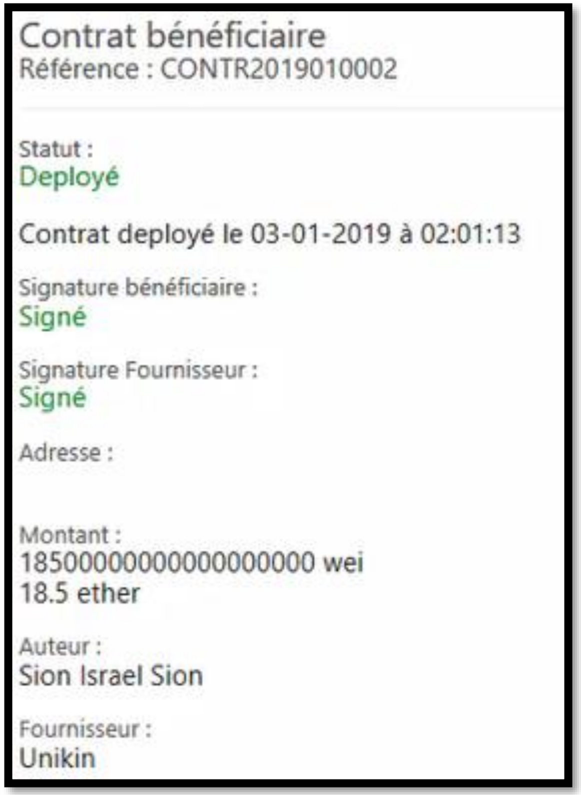
Future work: How to optimize Fabric (esp. Orderer) to reduce **false positives** or limit conflicts?

Other projects



On Thu 28 June, 2018

From 23:15



Contrat bénéficiaire
Référence : CONTR2019010002

Statut :
Deployé

Contrat deployé le 03-01-2019 à 02:01:13

Signature bénéficiaire :
Signé

Signature Fournisseur :
Signé

Adresse :

Montant :
18500000000000000000 wei
18.5 ether

Auteur :
Sion Israel Sion

Fournisseur :
Unikin

Blood donation system using blockchains

Infrastructure for large-scale pool mining

Cross-border remittance

EVA: Electrical Vehicle Aggregation

fuseelab.github.io

Teaching @ ÉTS

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE, MONTRÉAL, CANADA

UNIVERSITY OF QUÉBEC

CREDENCE WORKSHOP 2019

Overview of SYS869

Decentralized systems and applications

- Graduate course with lectures

Objectives:

- Master fundamental concepts related to cryptocurrency and blockchain
- Analyze critically future DApps and systems, understand trade-offs governing blockchains
- Design, develop, and evaluate Dapps and smart contracts

3 credits course

- 13 weeks, ~30 hours of lectures
- 2 exams, and a final project
- Three project types: algorithms, systems, and **DApps**

Lectures content, first half

Lecture 1: Introduction

Lecture 2: Byzantine generals, Nakamoto consensus

Lecture 3: UTXO model, addresses, wallets, script

Lecture 4: Gossiping protocol, Merkle trees, simple payment verification, Bloom filters

Lecture 5: Pool mining, Stratum protocol, pool rewards, pool attacks

Lectures content, second half

Lecture 6: Bitcoin improvements (Lightning Network, P2Pool, SegWit)

Lecture 7: Smart contracts, benefits of blockchain, DAPP methodology

Lecture 8: Ethereum, world state trie, gas, Ethash, GHOST

Lecture 9: DLT trade-offs, Hyperledger, Fabric EOV, MVCC problem

Lecture 10: Seminar course on varied topics, IOTA, Corda, Bitcoin-NG, Ripple, Hashgraph,…

Exercices, first half

A1: Consensus

- Byzantine generals
- Proof-of-work
- 51% attacks

A2: Transations

- UTXO model
- Bitcoin script
- Wallet security

A3: Networking

- Bloom filter
- Merkle trees, SPV
- Block propagation delay

Exercises, second half

A4: Attacks

- Selfish mining
- Pool rewards
- Pool hopping and block withholding

A5: Ethereum

- DAPP scenarios analysis
- Solidity constructs
- GHOST and uncles

A6: Hyperledger

- System trade-offs (DCS)
- Execute-order-validate
- MVCC problem: read-sets and write-sets

Thoughts after first edition (I)

Ordering issue

- Logical order: Bitcoin, then smart contracts, then Ethereum
- Practical order: Smart contracts, Ethereum, then Bitcoin
- Solution: maybe divide the semester into two smaller subprojects (labs)

Student interaction

- Include plenty of leading questions, discussion topics, …
- Course would not translate well to online form

Neutral content delivery

- Promotes critical thinking
- With plenty of time to discuss impact of design decisions

Thoughts after first edition II

Scope of the course

- Chosen systems: Bitcoin, Ethereum, Hyperledger Fabric
- No extensive material on cryptography, distributed systems, game theory
- Essentially cryptography and DS are pre-requistes
- Drawback: cannot explore advanced crypto topics (accumulators, zero knowledge proofs, ECDSA, BLS) or attack analysis

Only highlights of smart contract programming given in class

- Ethereum gas metering
- Hyperledger MVCC
- Students learn the rest in the project

Other courses

LOG736: Foundations of distributed systems (undergrad)

- Loosely follows DS book by Tanenbaum and van Steen
- Clock synchronization, logical clocks
- Coordination, consensus: Paxos, Raft
- State machine replication, consistency models
- CAP theorem
- Byzantine consensus, blockchains
- Final lab on Nakamoto consensus

LOG721: Distributed applications and middleware (undergrad)

- RPC, message queues, publish/subscribe
- MapReduce, Spark mechanisms
- P2P routing, DHT
- Distributed storage: GFS/HDFS, erasure coding, CRDT
- Smart contracts programming with Solidity
- Final lab on DAPP implementation on Ethereum

Backup

MVCC problem: Hyperledger Fabric

Background: HyperPubSub

- Federated publish/subscribe
- Monetization of IoT data streams

Execute-Order-Validate

- Transactions are first executed, then validated
- Stale transactions are aborted

MVCC Problem

- High abort rate leads to a reduction of effective transaction throughput

Possible leads at various layers

- Better chaincode (smart contract) design
- Better ordering service
- Faster propagation of smaller blocks
- Custom logic for resolving conflicts, while respecting endorsements

Other projects

EVA: Electrical Vehicle Aggregation

- Fair and transparent EV scheduling
- https://github.com/i13-msrg/EVA

Blood donation system using blockchains

- Traceability of the blood donation process
- Process validation using smart contracts
- Detailed feedback to the donor

Cross-border remittance

- Actors: migrant workers, families, service providers
- Multi-party transactions
- Issues of cryptocurrency and foreign exchanges

Infrastructure for large-scale mining

- Integration with Stratum (pool mining operator)
- Intra-DC block template dissemination