Hierarchical multi-blockchain architectures for autonomous management of medical data and services

Mike Burmester and Xiuwen Liu
Florida State University

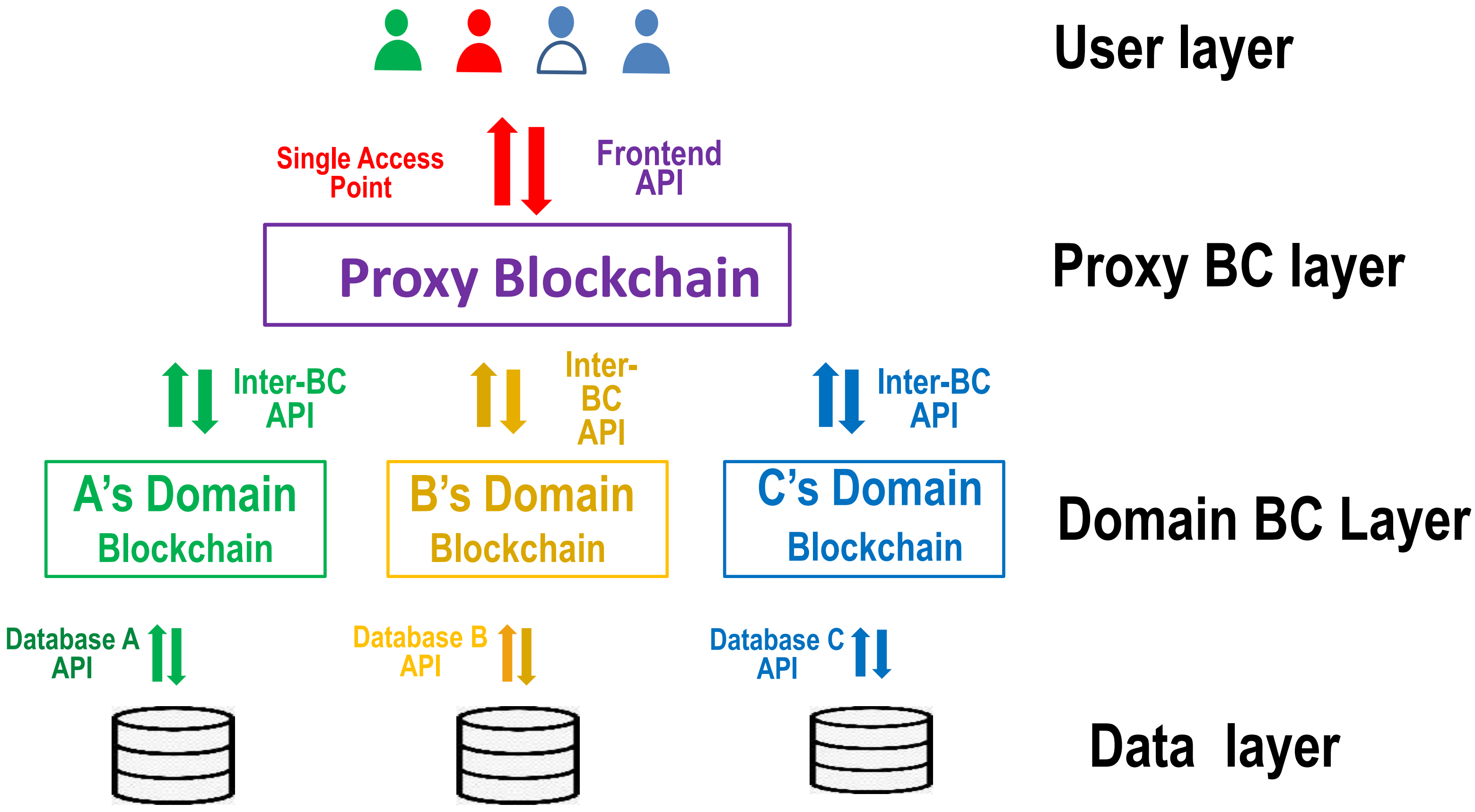
1. Challenges

- Managing and sharing remotely generated private data/services is a challenging task.
  - Centralized **cloud based** solutions provide interoperability & scalability, but make strong trust assumptions
  - Decentralized **blockchain** solutions provide security and independent trust management, but typically do not allow dynamic changes of underlying trust domains
- The challenge is to design architectures that achieve a good balance for:
  - **Dynamic trust management** for multi-authority & multi-domain applications
  - **Flexible fine-grain access control policy enforcement** at the at the domain and cross-domain level
  - **A global source of trust** for an immutable forensics-by-design auditing mechanism

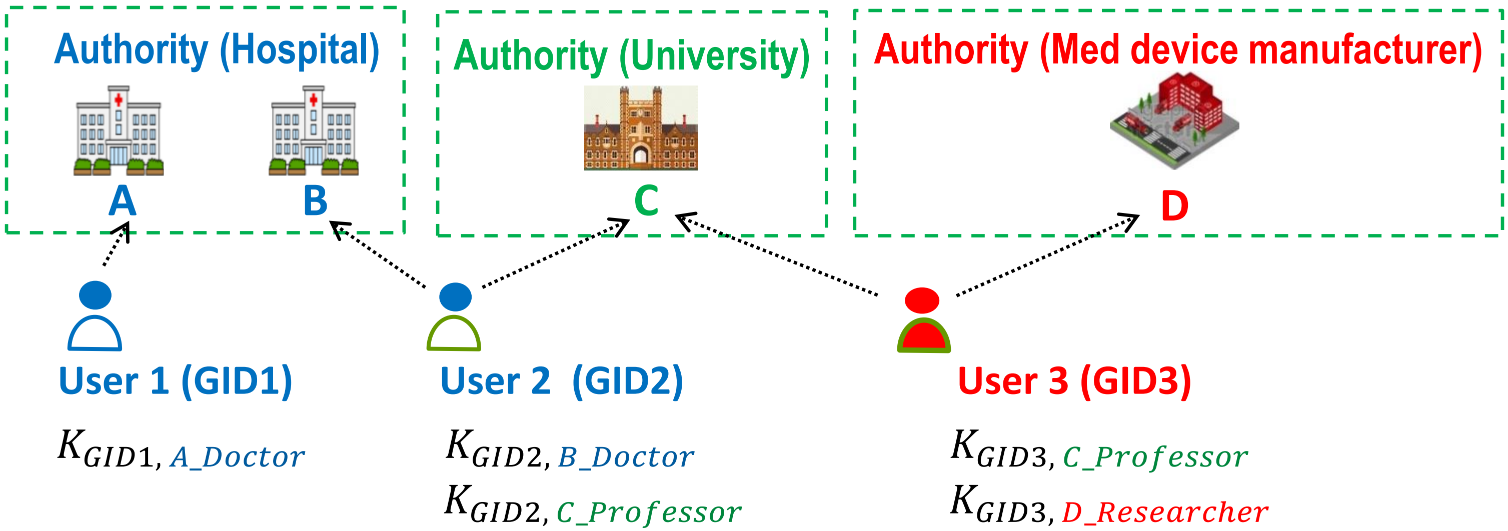
An architecture for secure management of multi-authority domain resources

- We propose a secure and flexible architecture for managing multi-authority domain resources that combines:
  - Hierarchical multi-blockchains with
  - Multi-Authority Ciphertext Policy Attribute Based Encryption (**MA-CP-ABE**).
- This architecture can be used for autonomous management of medical services and data, involving mutually untrusted stakeholders and patients.
- Security features:
  - Blockchains are used to ensure the integrity of transactions by creating an immutable forensics-by-design ledger
  - Blockchain services are implemented through Smart Contracts
  - MA-CP-ABE enables fine grain access control in which users combine attributes issued by different authorities to access system resources

The Hierarchical Multi-Blockchain architecture
-- an overview --



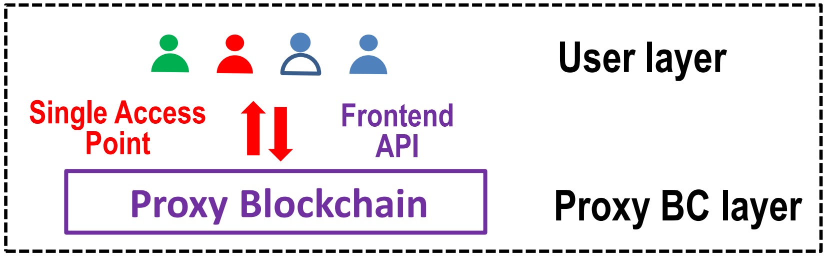
Flexible Access Control based on
Multi-Authority Ciphertext Policy Attribute Based Encryption (MA-CP-ABE)



MA-CP-ABE encryption and decryption

- GP: global parameters, GID_U = global identity of user U, $\mathcal{P}$ set of policies
- PK_X, SK_X: public/private key pair issued by authority X
- **MA-CP-ABE Encryption**: $ct = Enc(m, \mathcal{P}, GP, \{PK\})$ the MA-CP-ABE encryption of m = message using the public keys of the authorities involved
- **MA-CP-ABE Decryption:** users use their combined attribute keys: $m = Dec(ct, GP, \{K_{U,attr}\})$

2. Building blocks of the architecture

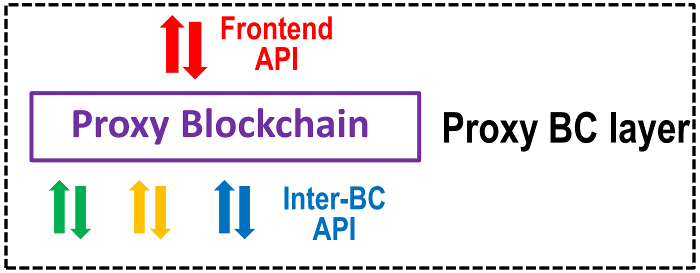


A. Frontend API

Provides *a single point of entry* that cannot be bypassed by users or authorities and is implemented by a distributed 2-step decryption procedure in which:

- each data item d_i is initially encrypted with a symmetric key k_i: $c_i = AES(k_i, d_i)$, and then
- the key k_i is MA_ABE encrypted based on the access policies for d_i .

Building blocks: Hierarchical Blockchains

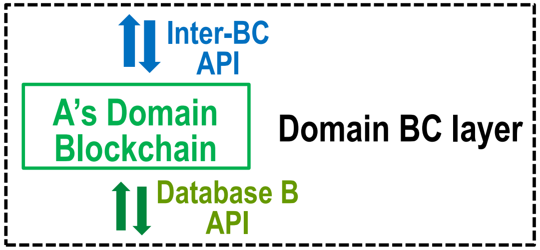


B. Proxy Blockchain

receives user requests via the Frontend API and implements 3 main services:

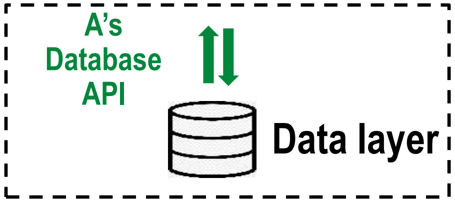
- Handles requests accompanied with attribute certificate(s) and validates the certificate (handled by a *Proxy Smart Contract*)
- Validates the attributes assigned to the user (handled by a *Trust Management Smart Contract*)
- Creates a log for each incoming request (handled by a *Logging Smart Contract*)

Building blocks: Hierarchical Blockchains



C. The Authorities Domain Blockchains

receive user requests only via the Inter-BC API and enforce the access control policies of the particular authorities by checking that the user attributes are sufficient for the specific request (handled by a ***Access Control Smart Contract***)



D. The Data layer

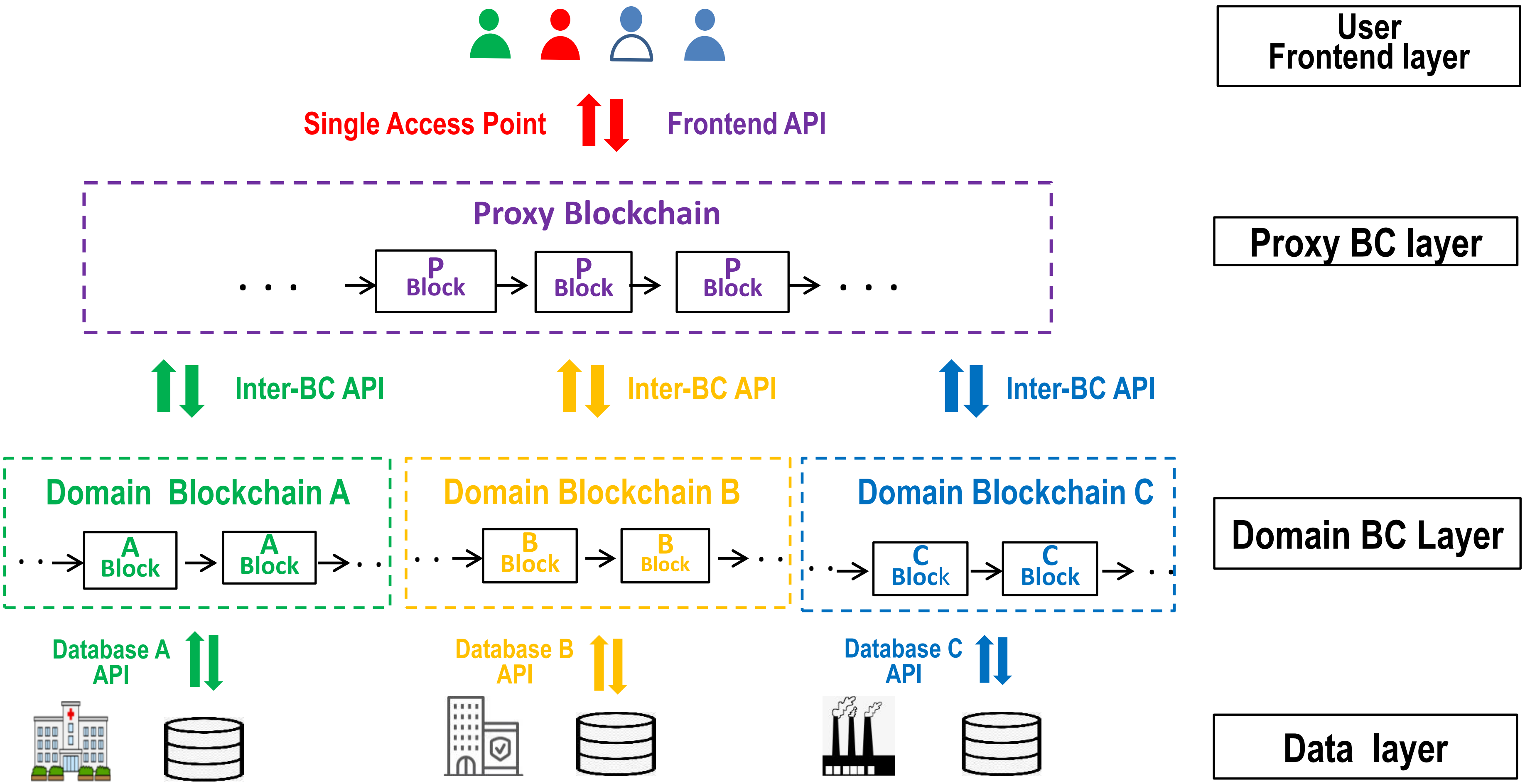
When encrypted data return from the Data Layer, the Database API passes these to a ***Key Store Smart Contract*** that has the relevant domain's attribute public/ private key pair to partially decrypt the data, that is then forwarded to the Proxy BC via the Inter-BC API and eventually to the user via the Frontend API

Building blocks: Smart Contracts

Blockchain services are implemented through smart contracts:

- *Proxy Smart Contract*: integrates the functions of *user validation*, *stakeholder vot*ing and *request forwarding*.
- *Trust Management Smart Contract*: supports trust management services by handling certificates & revocation lists.
- *Access Control Smart Contract*: enforces the predefined access policies.
- *Logging Smart Contract*: enforces *a single point-of-truth* through the registration and retrieval of logs.
- *Key Store Smart Contract*: enforces a *single-point-of access* to the system. This is the only component of the system that can access a domain's attribute private key, needed to access the domain's Database API.

The Hierarchical Multi-Blockchain architecture
-- details --



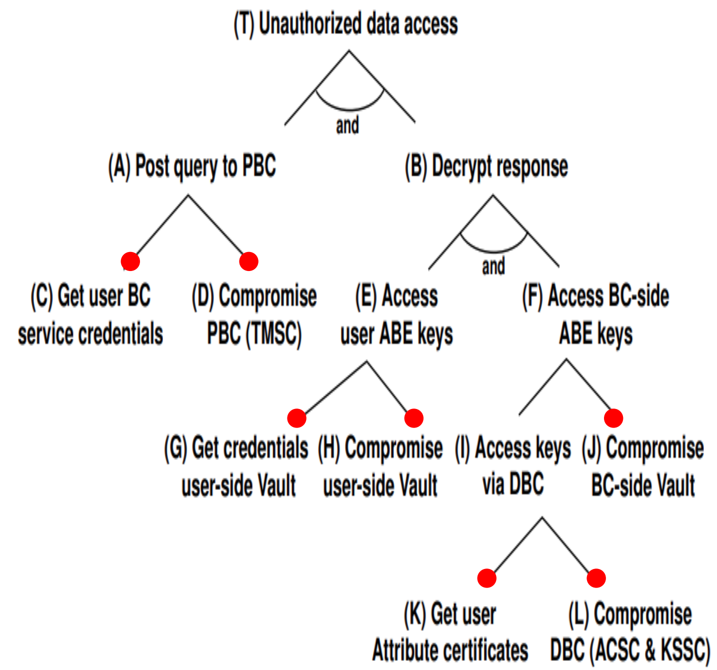
Security

To analyze *unauthorized data access* we use attack trees. For the Hierarchical Blockchain architecture we have 12 nodes:

A, B, <u>C</u>, <u>D</u>, E, F, <u>G</u>, <u>H</u>, I, <u>J</u>, <u>K</u>, <u>L</u> with 7 leafs.

Since we are assuming that there is a *single access point*, attacks must originate at leafs. We identify the following vulnerabilities

- Fully or partially compromised user
- Fully compromised PBC and DBC
- Fully compromised vault
- All entities are partially compromised



In all these case it is shown that the data of non-compromised users is not affected provided their ABE keys are secure

Implementation

Relies on the integration of several technologies

- The Blockchains PBC and DBS are developed on the Hyperledger Fabric platform[1] with Raft[2] the underlying consensus mechanism.
- The orchestration relies on Kubernetes[3] with smart contracts and all APIs executed in Kubernetes Pods. Stakeholders CAs are implemented the Hyperledger Fabric Certificate Authority.
- Hashicorp Vault[4] is used for storing user and authority credential and keys.
- To keep a healthy flow and avoid DOS errors/attacks on Proxy BC a Queue Supervisor is utilized.

1. https://www.hyperledger.org
2. https://www.github.io
3. https://www.kubernetes.io
4. https://www.vaultproject.io

Thanks!

Collaborators:

Vangelis Malamas, Panayotis Kotzanikolaou –University of Piraeus, Greece

George Palaiologos, Dimitris Glynos, CENSUS, S.A. Greece

V Malamas,T Vasaklis, P Kotzanikolaou, M Burmester, & S Katsikas. "A forensics-by-design management framework for medical devices based on blockchain." In *2019 IEEE world congress on services (SERVICES)*, vol. 2642, pp. 35-40. IEEE, 2019.