Decentralization of Identities in Blockchain

C&ESAR 2024, November, Rennes, France

Authors

Ahmed Abid - Pierre Alain
ENSSAT, Université de Rennes
IRISA, Université de Rennes

OUTLINE

1

Problematic

The main challenges in secure decentralized identity management.

2

Solution 1 & Limitations

Encryption management with smart contracts.

3

Solution 2 & Limitations

Data referencing using IPFS.

OUTLINE

4

Solution 3 & Limitations

Integration of unikernels with Albatross.

5

Summary Comparative Analysis & Conclusion

- Comparing the three solutions.
- Wrap-up and next steps.

Introduction

Comment garantir que les données restent sous le contrôle des utilisateurs, malgré les vulnérabilités des algorithmes de chiffrement découvertes après l’enregistrement dans la blockchain?

GROWING COLLECTION OF PERSONAL DATA

PRIVACY AND SECURITY ISSUES

ROLE OF BLOCKCHAIN

State of the Art
Blockchain Context and Data Security

- **INCREASING USE OF BLOCKCHAIN**

  Blockchain technology is increasingly used for secure storage of sensitive data.

- **KEY PROPERTIES OF BLOCKCHAIN**

  Immutability and decentralization make blockchain a strong candidate for identity management.

- **REFERENCE TO LITERATURE**

  Crosby et al. (2016) demonstrated the use of blockchain for secure data storage, highlighting its immutability.

State of the Art

Challenges in Decentralized Identity Management

- **LIMITATIONS OF TRADITIONAL METHODS**

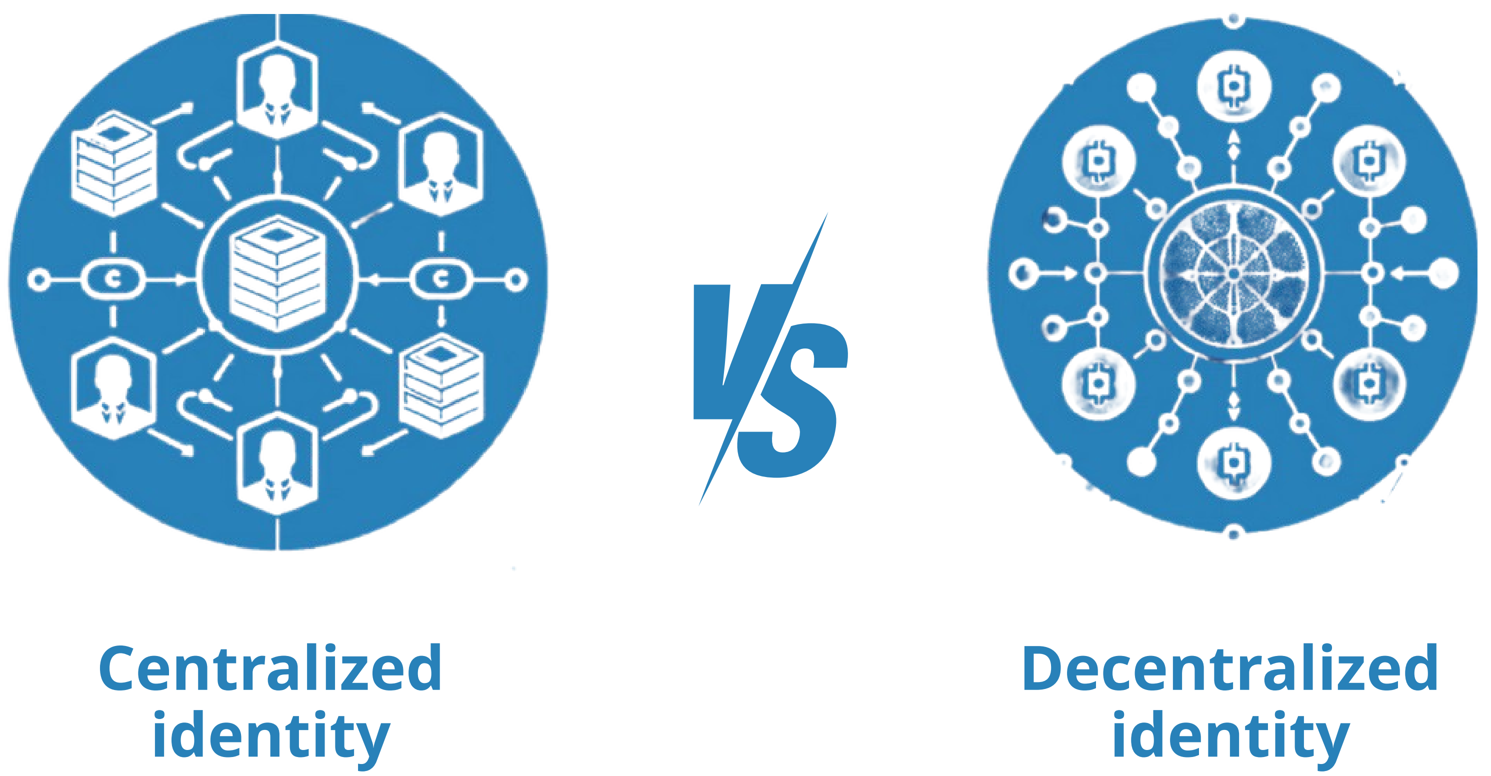
  Traditional identity management methods are prone to vulnerabilities and lack privacy guarantees.

- **IMPORTANCE OF DECENTRALIZED VERIFICATION**

  Antonopoulos and Wood (2017) highlighted the significance of decentralized identity verification in ensuring security.

- **SSICHAIN EXAMPLE**

  Kim and Laskowski (2018) proposed SSIChain, a decentralized identity management framework leveraging blockchain.



State of the Art

Technologies: IPFS

- **IPFS FOR DECENTRALIZED STORAGE**

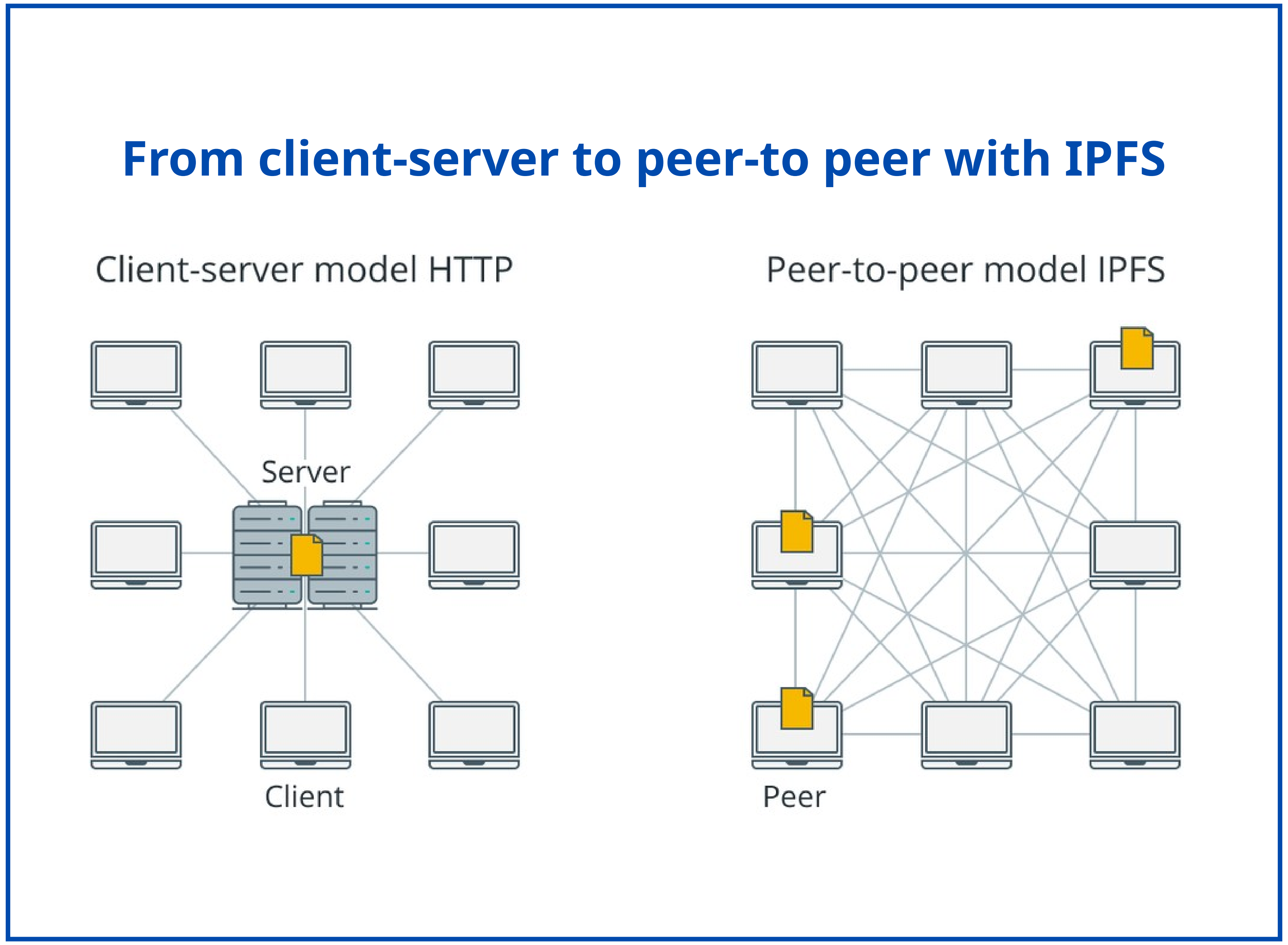
  IPFS (InterPlanetary File System) is used for decentralized data storage, avoiding single points of failure.

- **ENHANCED CONFIDENTIALITY VIA SUBMARINE**

  The Submarine method hides data until it is ready to be revealed, ensuring exclusivity.

- **REFERENCE TO BENET AND DYLAN**

  Benet (2014) and Dylan (2021) explored the use of IPFS for secure data storage and the Submarine method for enhancing confidentiality.



State of the Art

Remaining Challenges and Open Questions

- **DATA IMMUTABILITY**

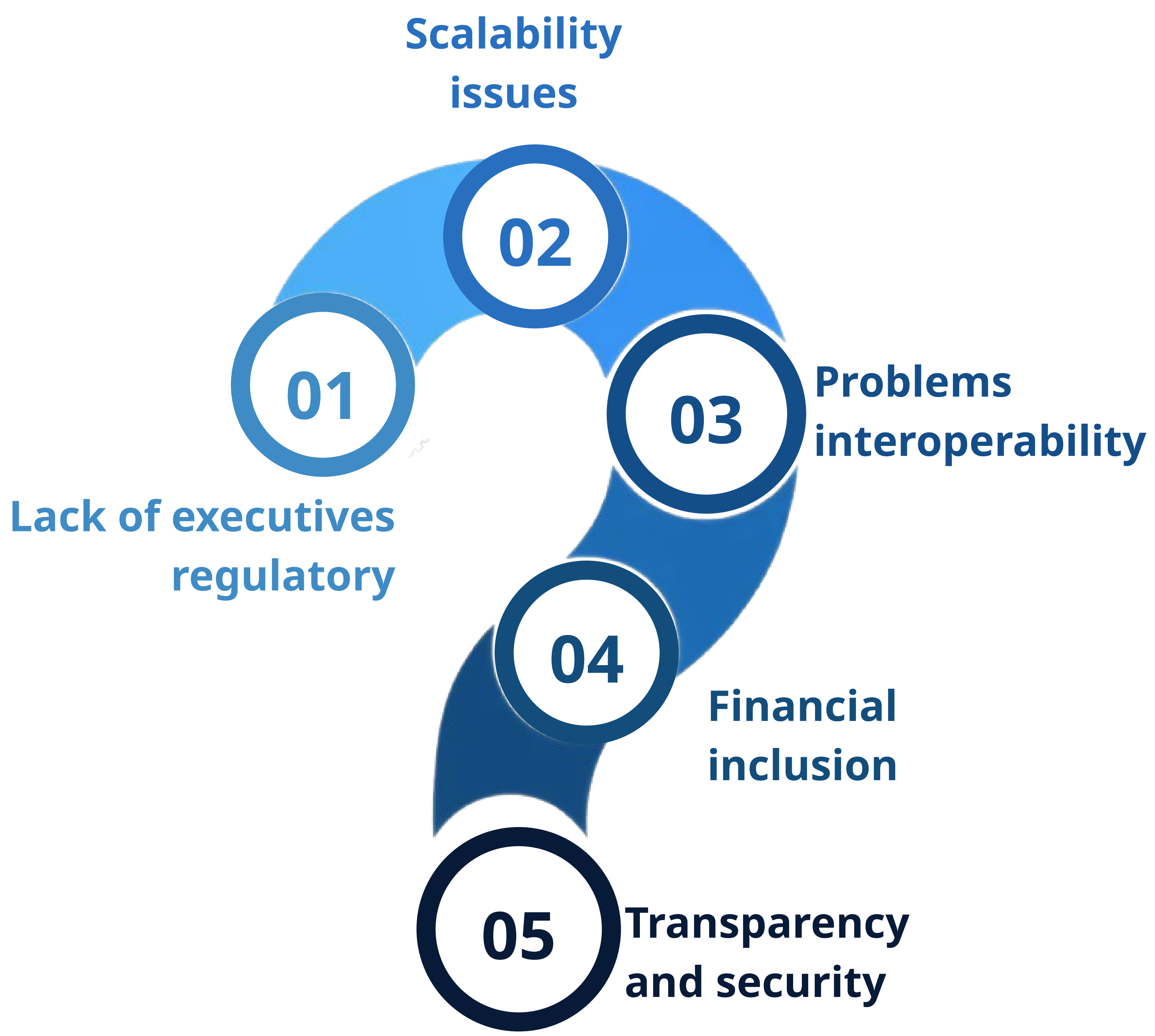
  Once data is stored in a blockchain, it becomes immutable, which poses issues if vulnerabilities are discovered
  in encryption algorithms.

- **INTEROPERABILITY AND IDENTITY MANAGEMENT**

  How can these decentralized identity systems be integrated with existing solutions while maintaining security ?

- **ENCRYPTION ALGORITHM UPDATES**

  A major challenge is how to update or replace encryption algorithms without compromising data integrity.



Proposed Framework for Identity Management

- **UNIFIED APPROACH:**

  A general framework for decentralized identity management to ensure security and flexibility in encryption.

  Our proposed framework addresses both data confidentiality and user control, by leveraging multiple technologies like blockchain, IPFS, and unikernels.

- **MAIN COMPONENTS:**

  Blockchain for data integrity.
  IPFS for flexible data referencing
  Albatross and unikernels for secure, isolated execution.

Solution 1

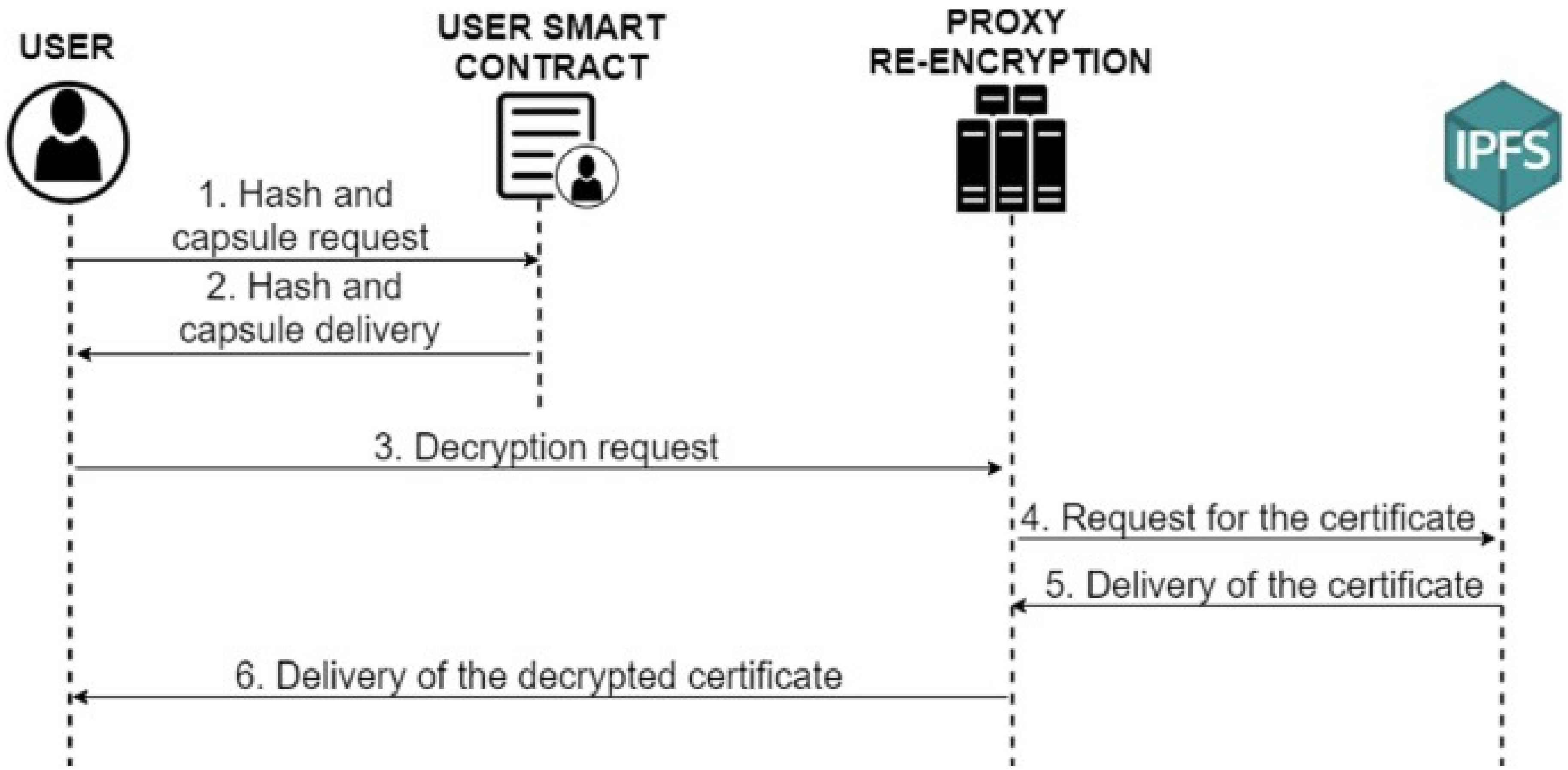
Encryption Management via Smart Contracts

- **DYNAMIC ENCRYPTION DURING DATA INSERTION**

  Data is encrypted dynamically during insertion into the blockchain, ensuring confidentiality.

- **MANAGEMENT VIA SMART CONTRACTS:**

  Smart contracts allow for the encryption algorithm to be updated without altering data integrity.



Solution 1

Practical Usage Scenario

- **LONG-TERM ENCRYPTION ADAPTATION:**

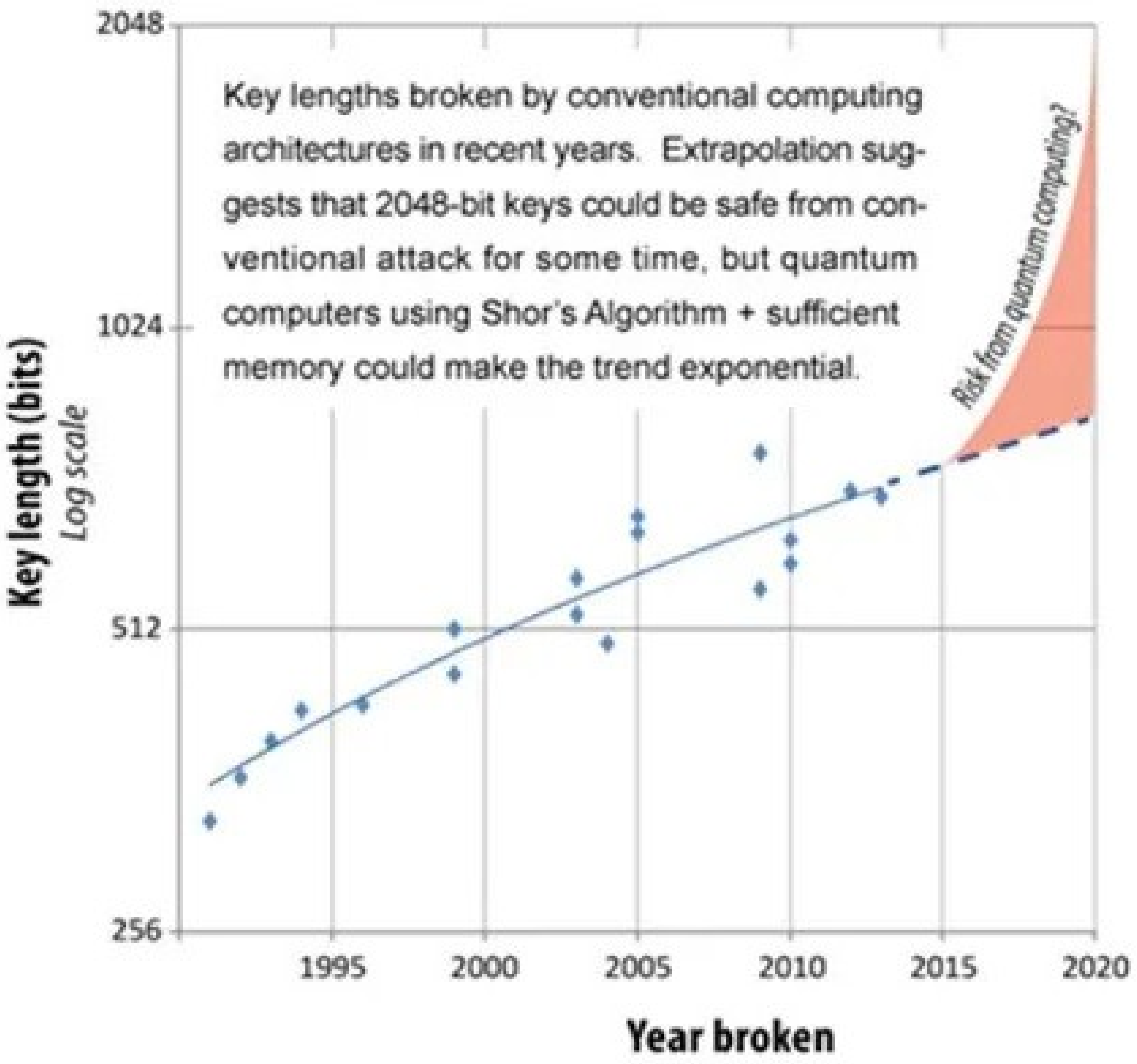
  Imagine a scenario five years from now, where the encryption algorithm currently in use is compromised.

- **DYNAMIC RE-ENCRYPTION:**

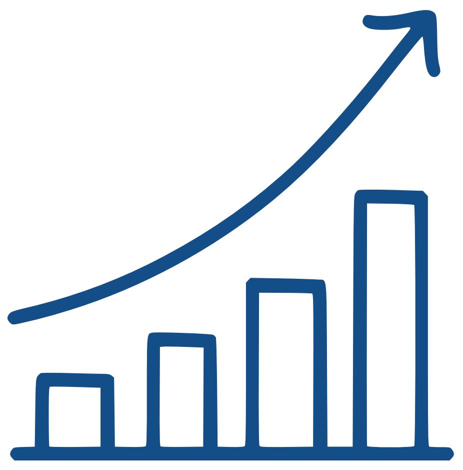
  The smart contract allows the encryption algorithm to be updated to a more secure version without altering data integrity or disrupting operations.

- **REAL-WORLD IMPLICATIONS:**

  This ensures the longevity of data security in a constantly evolving cryptographic landscape.



Solution 1
Limitations



- **COST AND SCALABILITY:**
  Updating encryption requires a significant amount of computing resources.

- **TRANSPARENCY ISSUE:**
  Historical transactions remain visible even if encryption is updated.

Solution 2

Concept of IPFS and Data Reference

- **ROLE OF IPFS IN DATA MANAGEMENT**

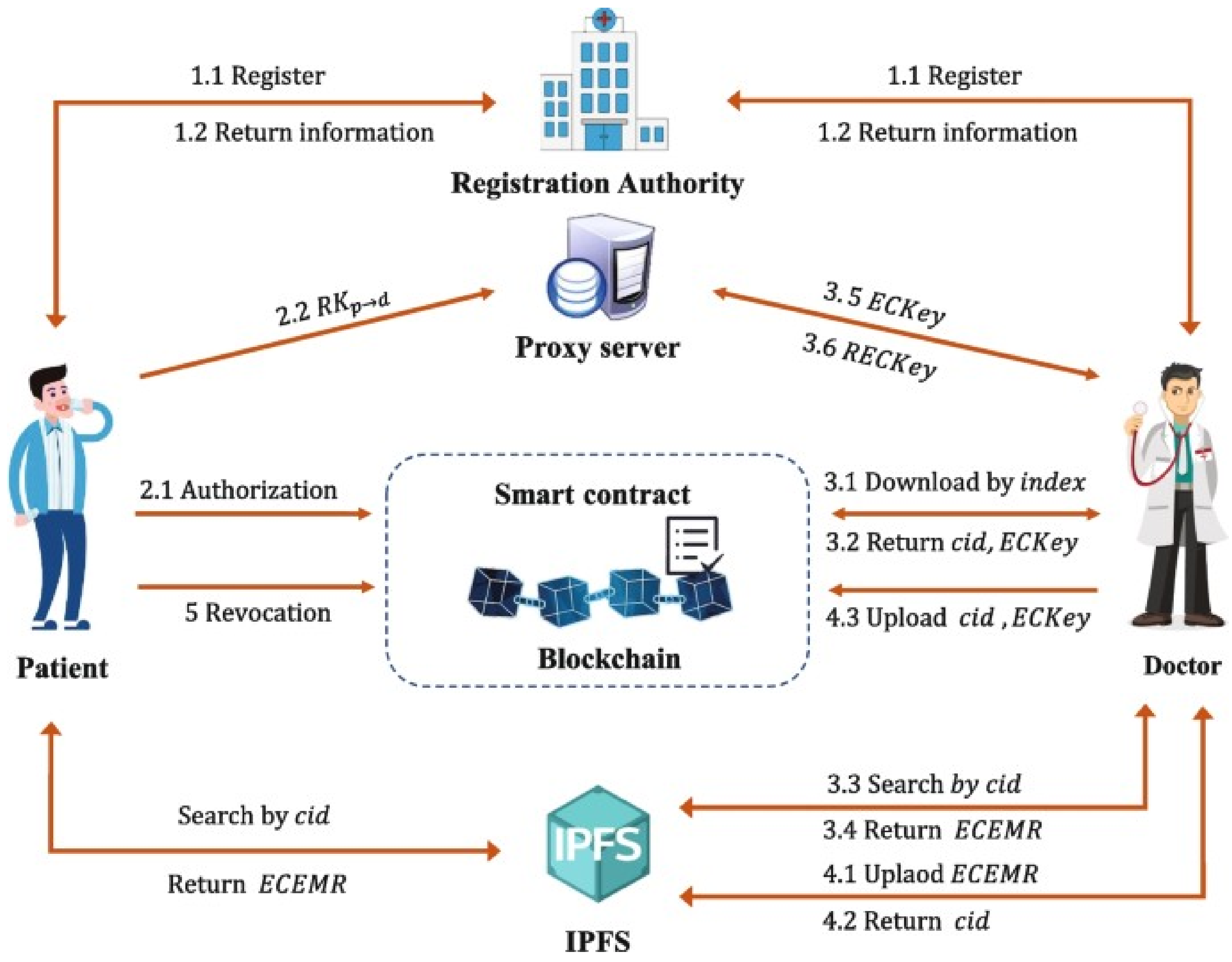
  IPFS is used to reference data rather than store it directly in the blockchain, optimizing scalability.

- **CID:**

  The Content Identifier (CID) is used to track data in a decentralized manner.

- **SEPARATION OF DATA AND ENCRYPTION KEYS**

  Encryption keys are securely stored separately, reducing the risk of exposure.



Solution 2

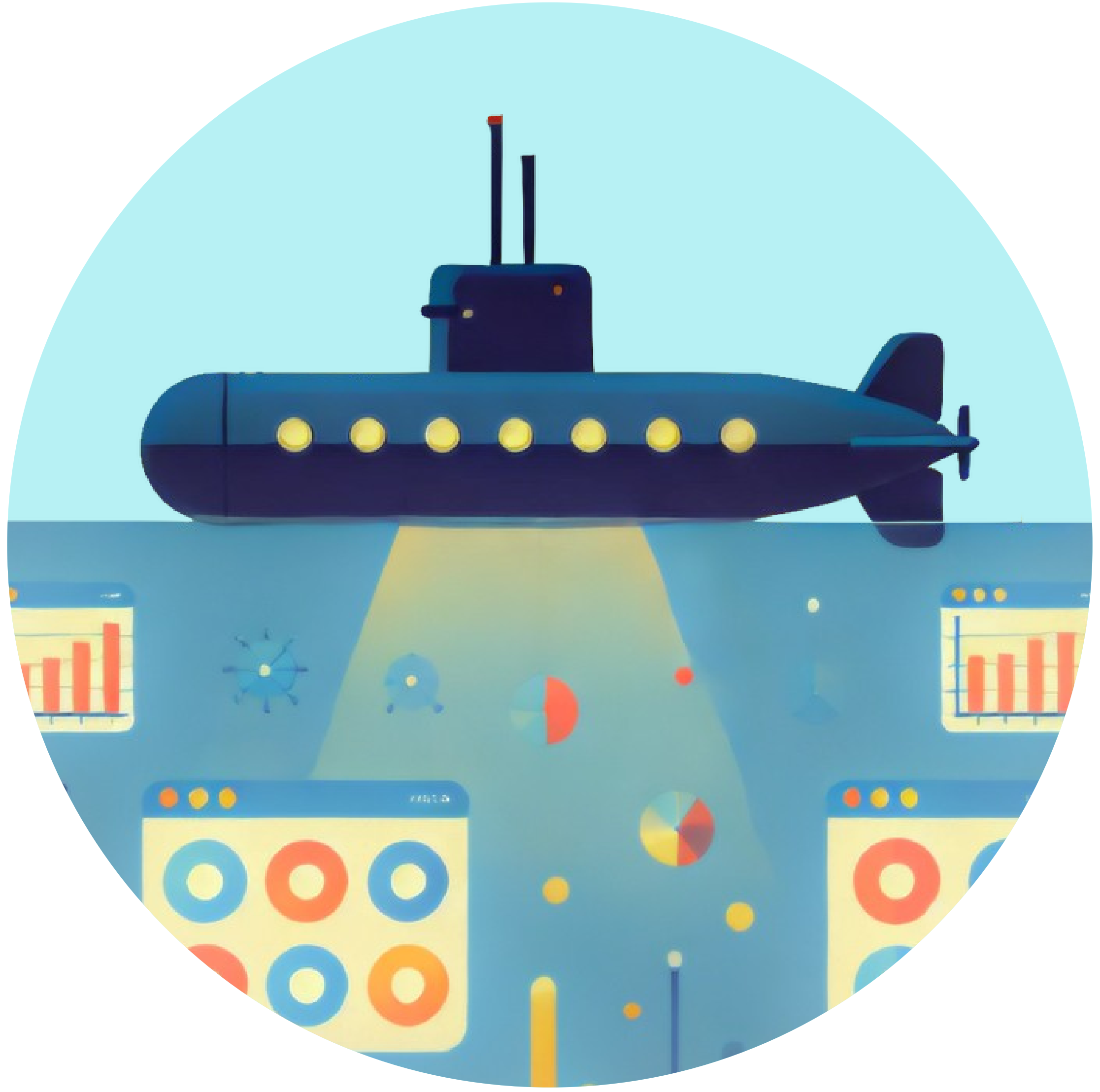
Submarine Method for Enhanced Data Confidentiality

- **SUBMARINE DATA HANDLING**

  Data is hidden initially and only revealed when the user chooses, providing enhanced confidentiality.

- **IMPLEMENTATION IN CONTEXT**

  Submarine method helps maintain exclusivity in sensitive use cases like NFTs or private datasets.



Solution 2
Limitations

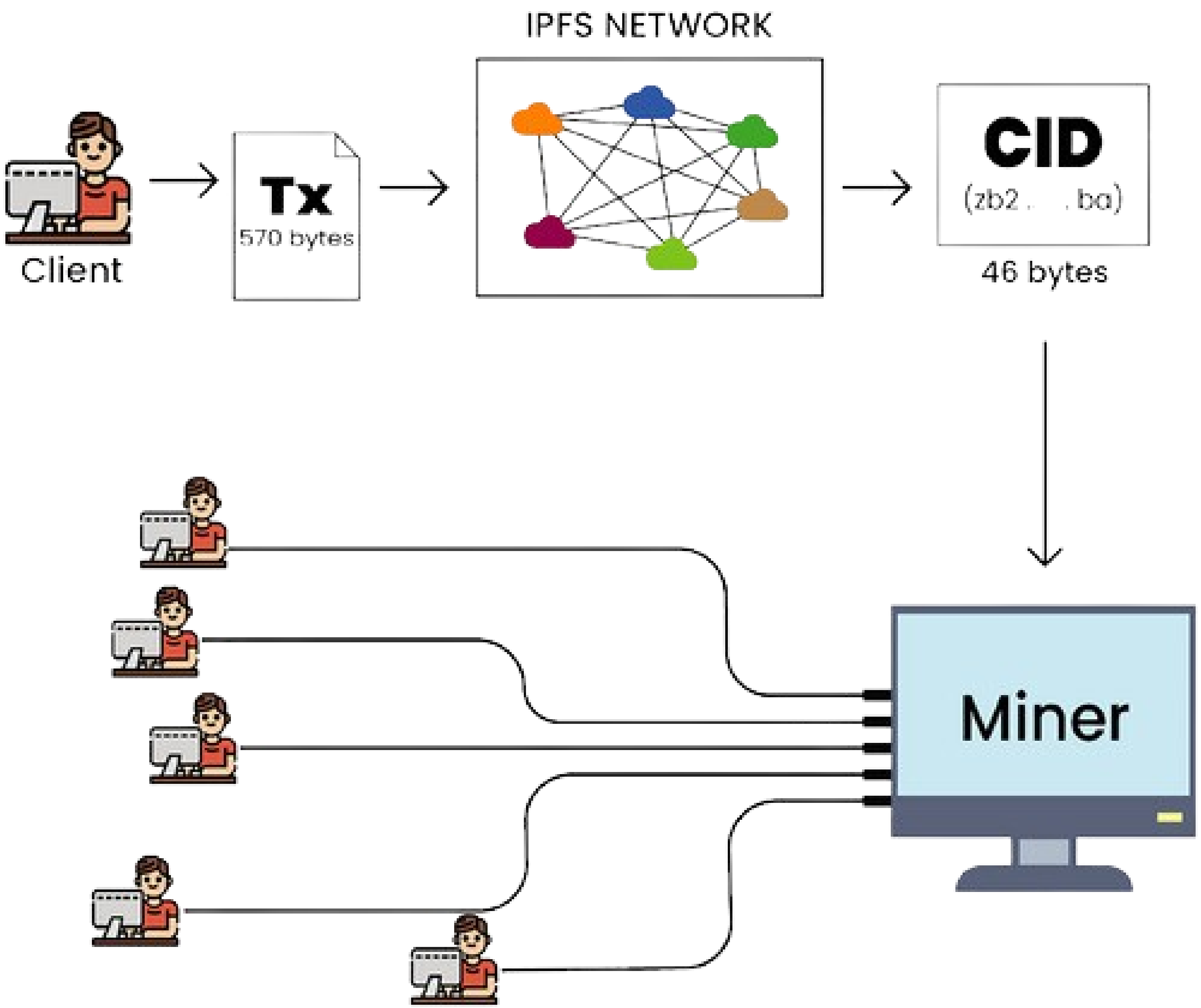
- **VULNERABILITY OF CIDS**

  The CID itself can be compromised, exposing the data.

- **PERFORMANCE ISSUES:**

  IPFS can have latency problems, especially in large-scale environments.

  The data should be shared but the end of the day



Solution 3

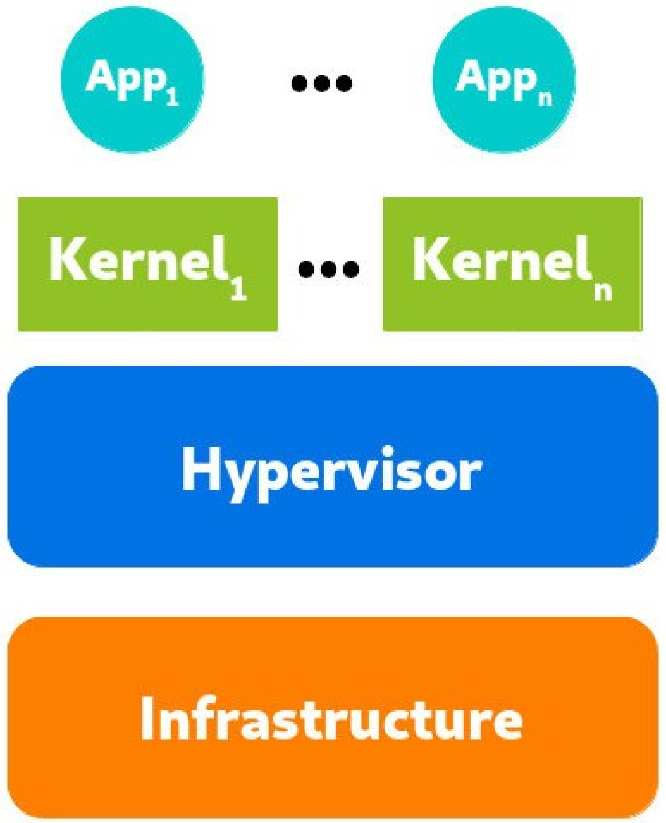
Introduction to Unikernels

- **WHAT ARE UNIKERNELS?**

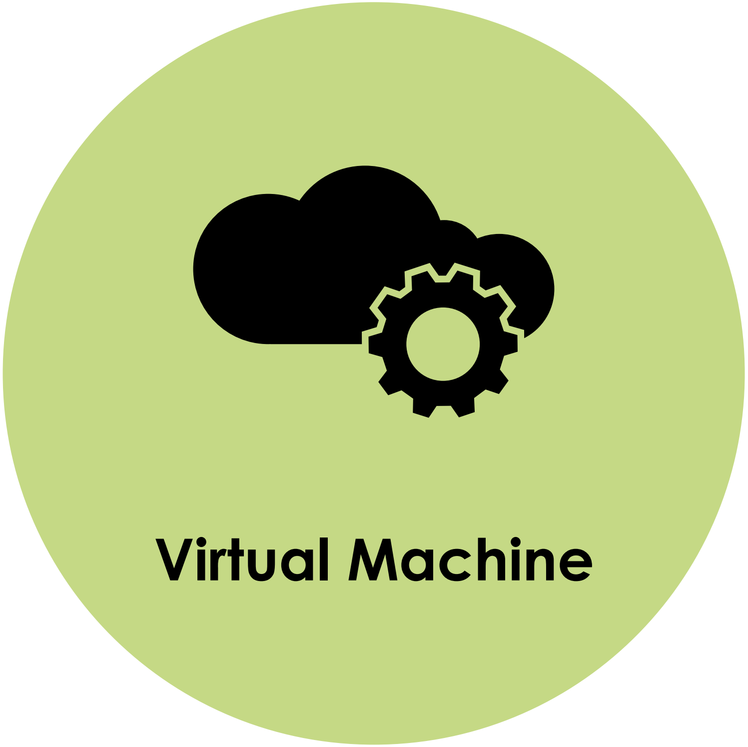
  Unikernels are lightweight operating systems that only integrate the components required for a specific task.

- **MIRAGEOS**

  Based on MirageOS, written in OCaml, offering memory safety guarantees.



Unikernel Architecture



Solution 3
Unipi and Secure Data Access

- **ROLE OF UNIPI**

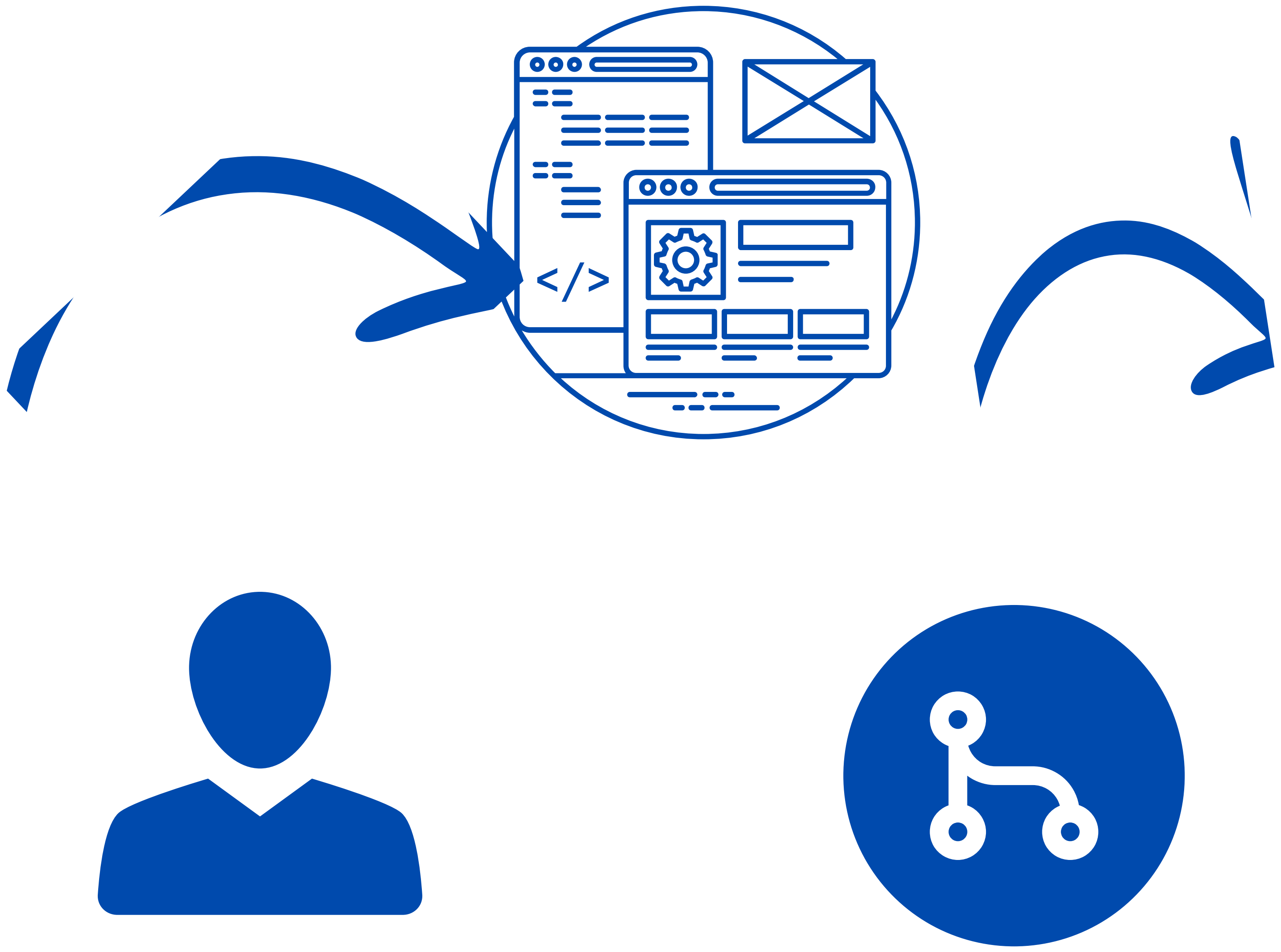
  Unipi acts as a secure gateway to access data from a private repository.

- **DATA ACCESS THROUGH GIT INTEGRATION**

  Utilizes a private Git repository to store sensitive code or configuration files.

- **DATA RETRIEVAL FLOW**

  Requests are authorized by Unipi, which checks credentials before accessing the data.



Solution 3

Unipi and Certificates for Secure Access

- **CERTIFICATES FOR AUTHENTICATION**

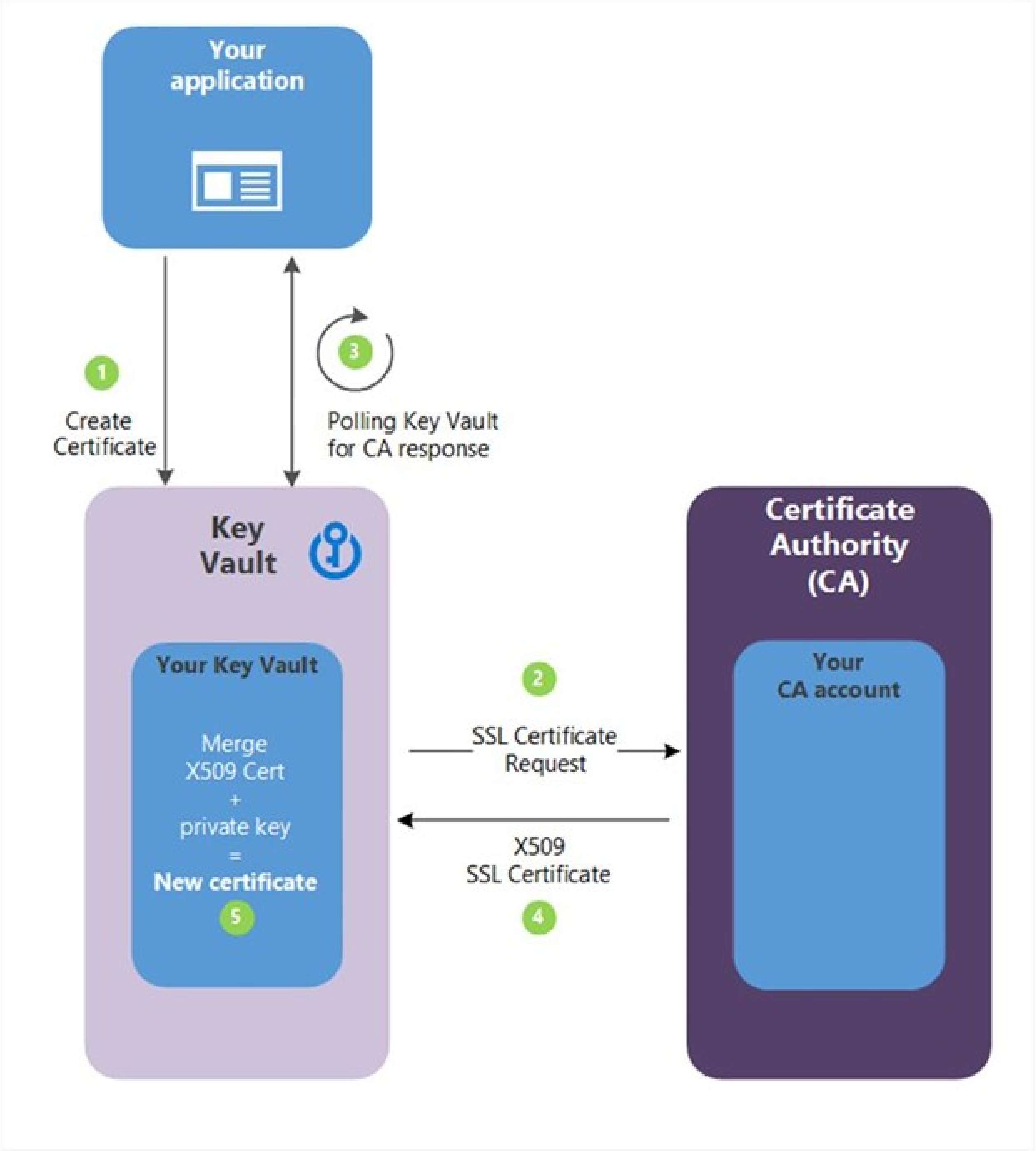
  Private keys and certificates are provided to users for authentication.

- **X.509 CERTIFICATES**

  These certificates validate the user identity and authorize data access.

- **CHAIN OF TRUST**

  A trust chain is established between the certificate authority, Unipi, and Albatross.



Solution 3

Albatross as the Orchestrator

- **ORCHESTRATION WITHOUT COMMAND LINE**

  Albatross provides seamless orchestration of unikernels without manual CLI input.

- **AUTOMATED DEPLOYMENT**

  Users only need to provide the certificate; Albatross handles the rest.

- **SIMPLIFIED USER EXPERIENCE**

  No manual intervention needed, reducing human error.

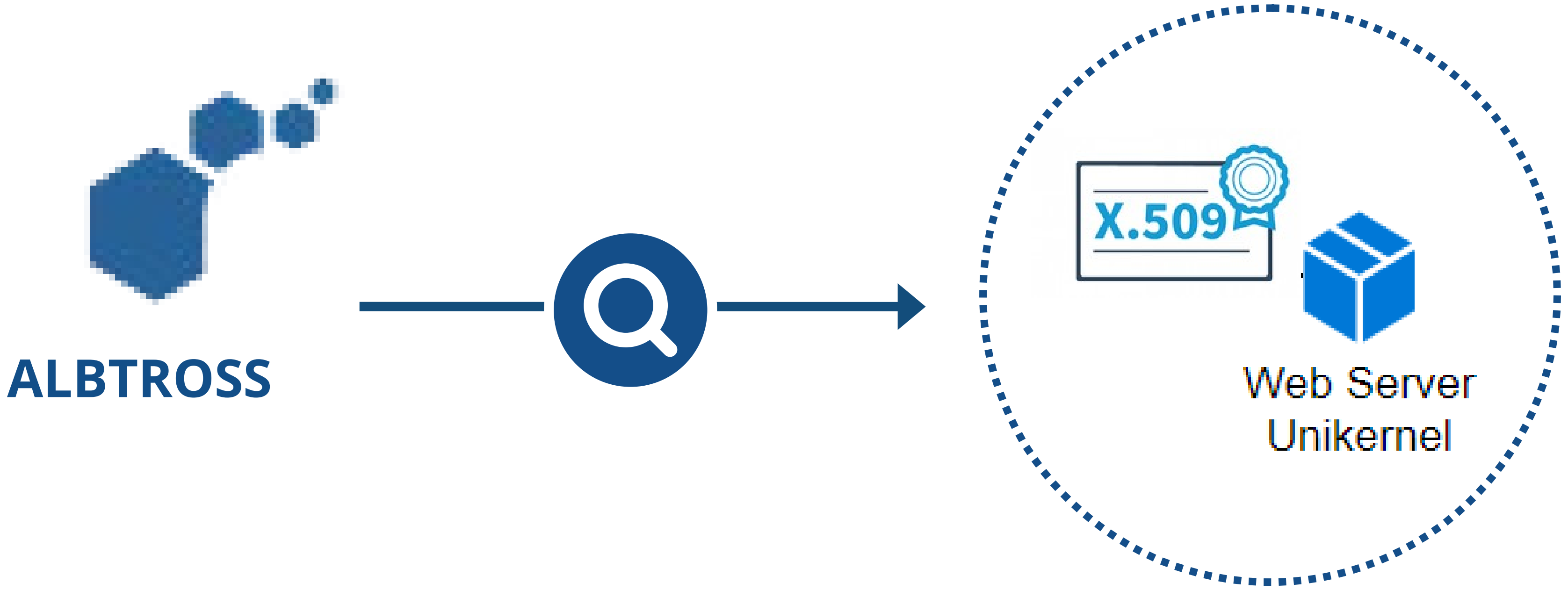
Solution 3

Embedding Unikernels in Certificates

- **EMBEDDING WITH X.509 CERTIFICATES**
  Each unikernel can be embedded within an X.509 certificate, which Albatross can read.

- **COMMAND AND AUTHORIZATION BUNDLED**
  Certificate contains the command (e.g., 'start unikernel') and authorization data.

- **FULL AUTOMATION**
  Orchestrator verifies the right to deploy and then executes the command.



Proposed Approach

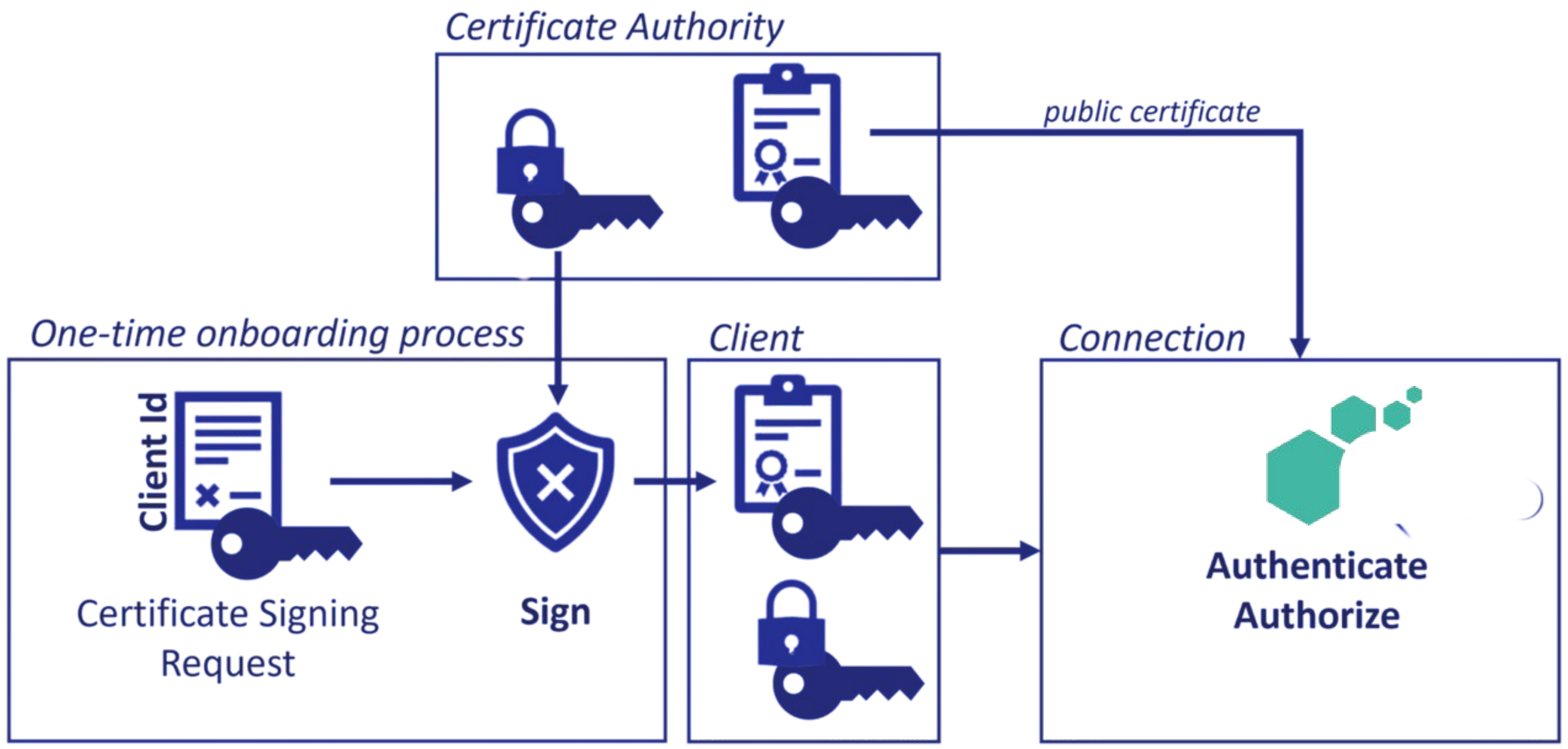
Integration of Virtual Machines for Data Access Control

- **CERTIFICATE CREATION AND SIGNING**

  Upon VM creation, the client generates a unique certificate using a private key.

- **PRIVATE KEY USAGE FOR AUTHENTICATION**

  The private key is provided to the authorized user, allowing secure authentication to Albatross.



Solution 3

Ensuring Data Isolation and Controlled Access

- **CLIENT DATA ISOLATION**

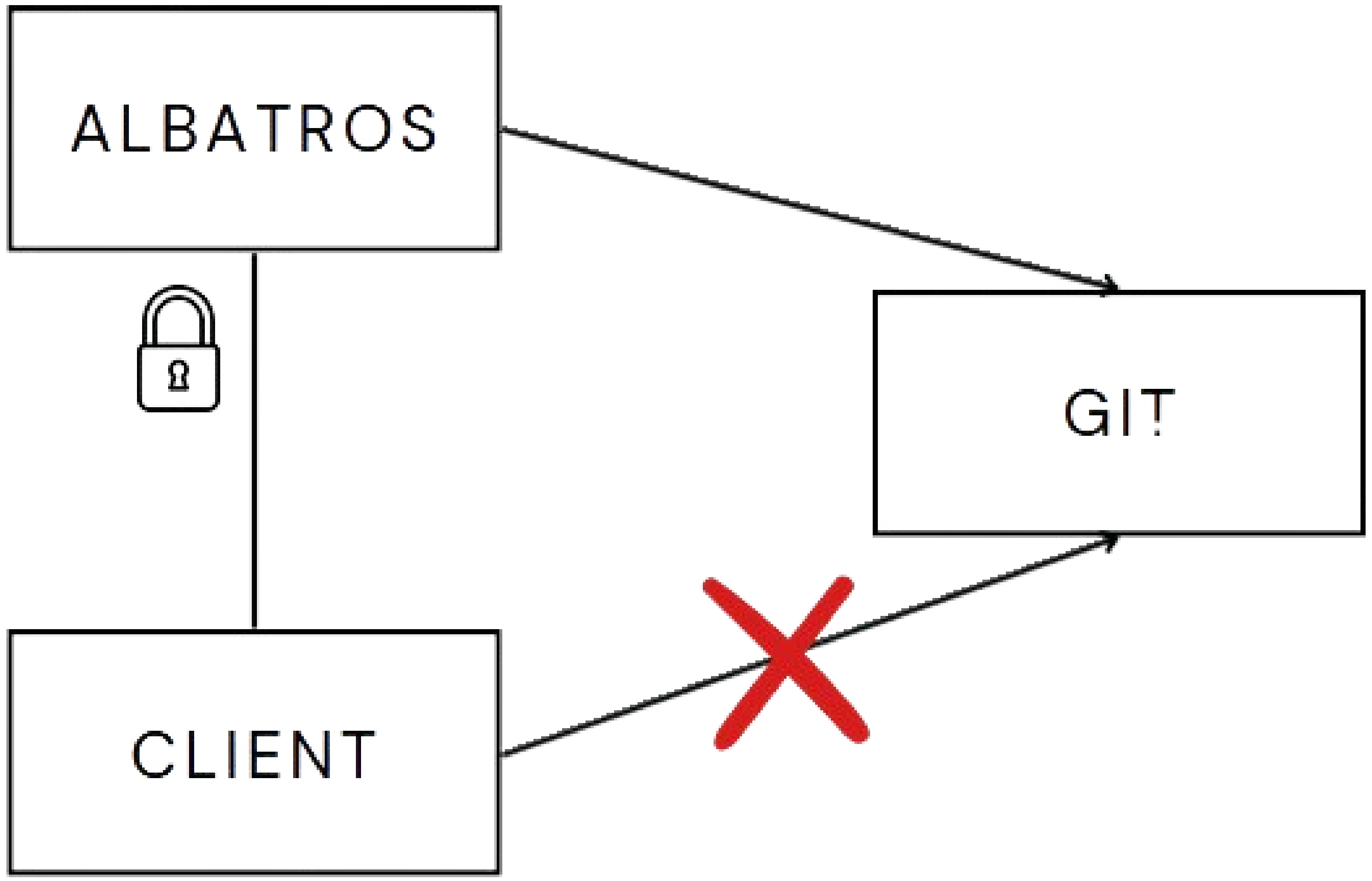
  Clients do not have direct access to sensitive data; they interact through a web interface.

- **SELECTIVE DATA SHARING**

  Only parts of the system are shared, maintaining strict isolation of sensitive information.

- **PROTOCOL FLEXIBILITY**

  Web-based access, but adaptable to other protocols.



Solution 3

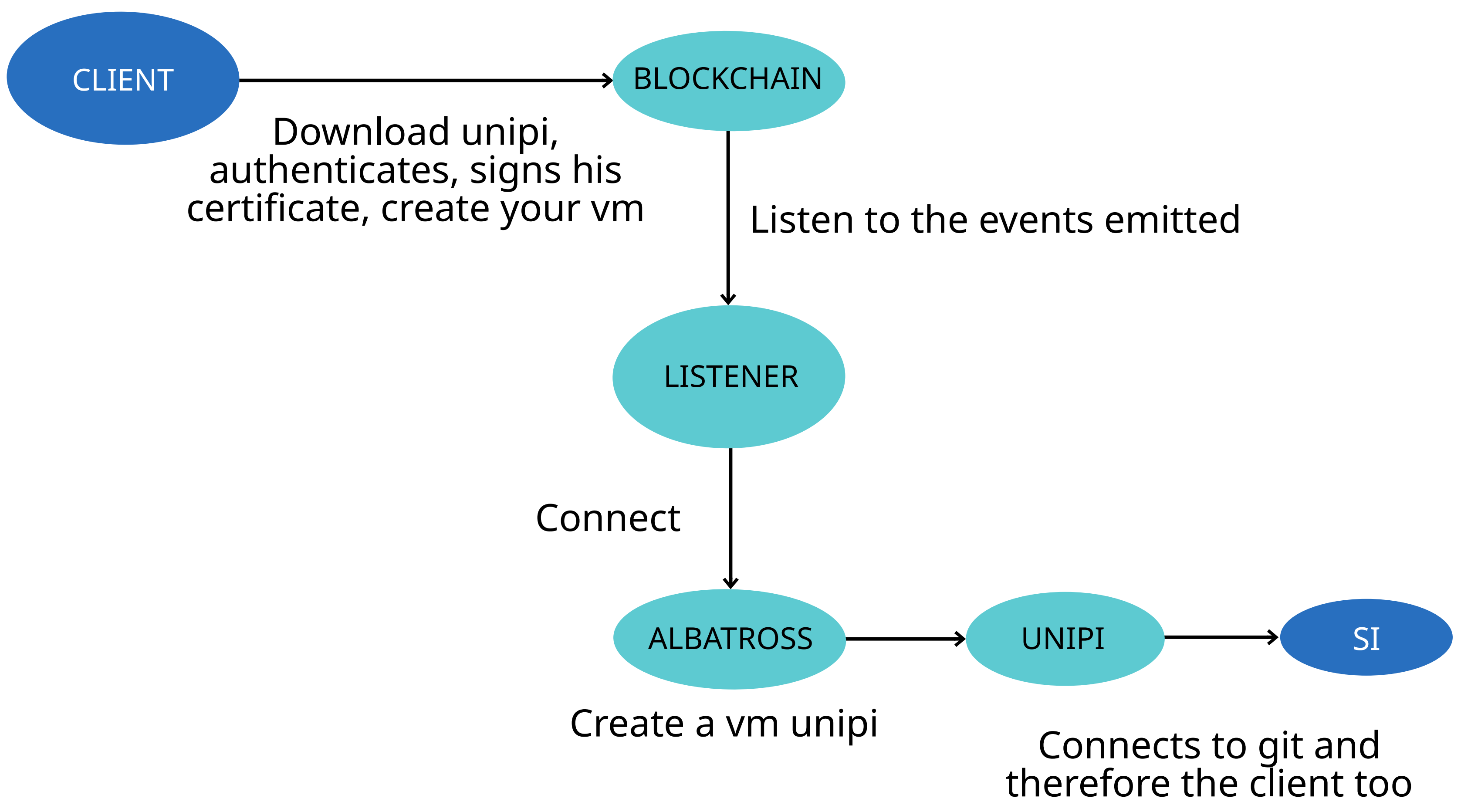
Lifecycle of a Transaction: Launching Unikernels

- **TRANSACTION FLOW**
  The process begins with a request from a client authenticated via certificate.

- **DEPLOYMENT LIFECYCLE**
  Albatross verifies permissions, starts the unikernel, and monitors its lifecycle.

- **LIMITED EXPOSURE**
  Only a portion of the information system is shared—isolated unikernel minimizes the surface area for exposure.

USE CASE : A COMPANY THAT SHARES A PART OF ITS SI WITH A CLIENT



Comparative Analysis

Comparison of Different Approaches

- **THREE APPROACHES COMPARED**

  Data Storage on Blockchain with Encryption Algorithm Modification"

  Use of IPFS and Submarine Method for Data Storage

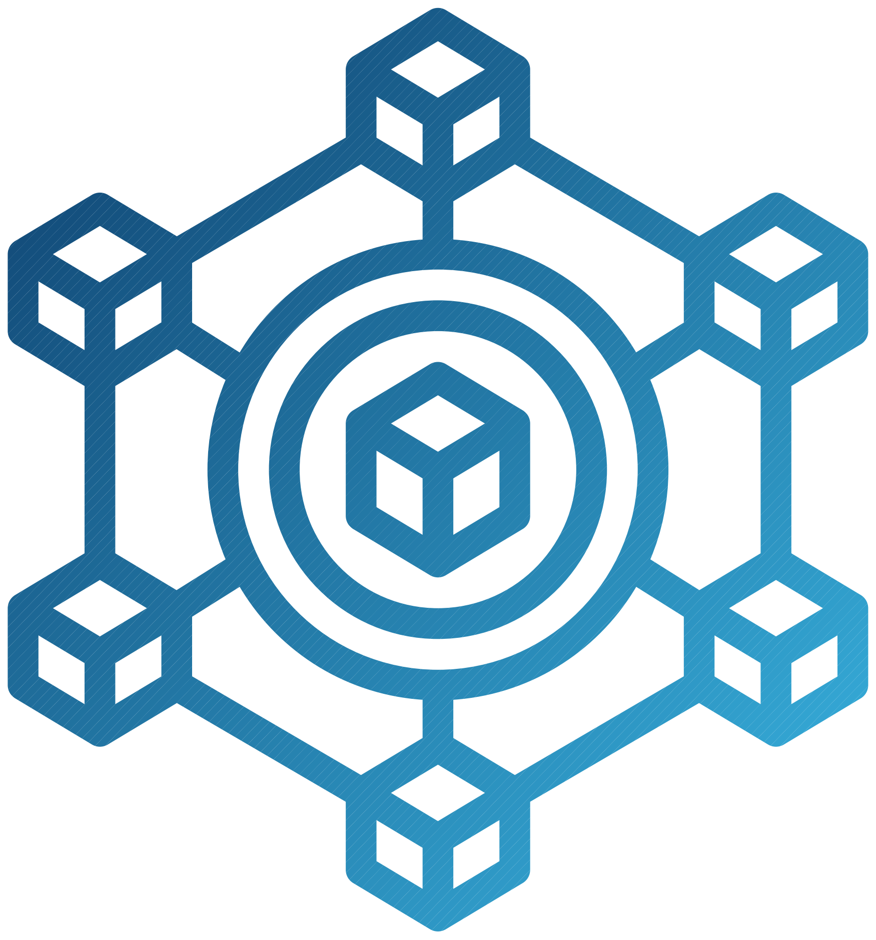
  Albatross and Virtual Machine-Based Data Management

- **STRENGTHS AND WEAKNESSES OF EACH APPROACH**

  Each method provides unique benefits and drawbacks in terms of scalability, security, and flexibility.

Critère	Blockchain	IPFS	Albatross
Scalabilité	ÉLEVÉE	MOYENNE	ÉLEVÉE
Décentralisation	ÉLEVÉE	ÉLEVÉE	ÉLEVÉE
Sécurité	MOYENNE	MOYENNE	ÉLEVÉE
Coût	ÉLEVÉE	MOYENNE	ÉLEVÉE
Fonct. Uniques	Immuabilité	Fichier Décentralisés	Informatique Décentralisée

Future Improvements and Limitations



PERFORMANCE OPTIMIZATION FOR IPFS AND KEY MANAGEMENT AND ENCRYPTION UPDATES:

improve IPFS performance and availability, focusing on reducing latency and increasing data redundancy.
Develop secure key rotation mechanisms to handle changes in encryption algorithms.

COMPLEXITY REDUCTION FOR VM MANAGEMENT:

Automate the configuration of Virtual Machines and improve integration to reduce operational complexity.

Conclusion

Key Insights:

- 1. Blockchain with encryption modification provides strong data integrity but at high cost.
- 2. IPFS with Submarine ensures data confidentiality but requires secure CID management.
- 3. VM integration offers high isolation but adds infrastructure complexity.

Path Forward:

Future improvements focus on optimizing scalability, security, and integration of decentralized data systems.