

**System Design**

R&D - VoteChain Team

April 3, 2018

**Project Coordinator:** Petteri Kaskenpalo

Lam Quang Vu

**Client:** Hoang Le Minh

**Supervisor:** Le Ngoc Son

**Team Members:** Nguyen Trong Tri

Pham Huynh Tri Minh

Contents

[INTRODUCTION 2](#_Toc510541758)

[Angular (Front-End) 3](#_Toc510541759)

[Ethereum (Blockchain Framework) 3](#_Toc510541760)

[Smart-Contract (Back-End) 4](#_Toc510541761)

[OBJECTIVES AND FEATURES 5](#_Toc510541762)

[Key benefits 5](#_Toc510541763)

[Features of Product 6](#_Toc510541764)

[Functional Scope 6](#_Toc510541765)

[SYSTEM ARCHITECTURE DIAGRAM 7](#_Toc510541766)

[UML DIAGRAMS 8](#_Toc510541767)

[Use Case Diagram 8](#_Toc510541768)

[Activity Diagram 9](#_Toc510541769)

[Sequence Diagram 10](#_Toc510541770)

[SYSTEM DESIGN 13](#_Toc510541771)

# INTRODUCTION

The objective of Blockchain Voting Project is seeking for the potential of blockchain technology, the advantages and disadvantages in order to start training associates in Bosch about Knowledge, Technique and Development.

Voting Application is a web-application to illustrate the potential of blockchain in 3 categories: Robustness, Security and Immutability.

Therefore, the objectives of this project are as follows:

* **Objective 1:** Create a runnable private network at Bosch, at least 11 computers.
* **Objective 2:** Create a vote room and cast the ballot.
* **Objective 3:** Create an account and login by ID instead of Address.
* **Objective 4:** Prototype a website-application using Angular.
* **Objective 5:** Restrict one-person-one-account.
* **Objective 6:** Provide 2 types of a vote room: Public and Private.
* **Objective 7:** Check history of each user.
* **Objective 8:** Provide multiple Question and Answer.
* **Objective 9:** Documents to prove the potential of blockchain.

## Angular (Front-End)

[](https://angular.io/docs)Angular is a platform that makes it easy to build applications with the web. Angular combines declarative templates, dependency injection, end to end tooling, and integrated best practices to solve development challenges. Angular empowers developers to build applications that live on the web, mobile, or the desktop.

Bosch aims to start using Angular in their project so they suggest using Angular to build front-end. Moreover, many associates have experiments about it; they can give a hand if the progress is stuck with angular troubles.

## Ethereum (Blockchain Framework)

[](https://www.ethereum.org/)

Ethereum is software running on a network of computers that ensures that data and small computer programs called smart contracts are replicated and processed on all the computers on the network, without a central coordinator. The vision is to create an unstoppable censorship-resistant self-sustaining decentralized world computer.

## Smart-Contract (Back-End)



Smart contracts represent special algorithms for the automation of contracts including deal-making processes. Such a contract allows people to sell realties, exchange shares, money, documents or any proprietary. Smart Contracts are self-executing contractual states, stored on the blockchain, which nobody controls and therefore everyone can trust.

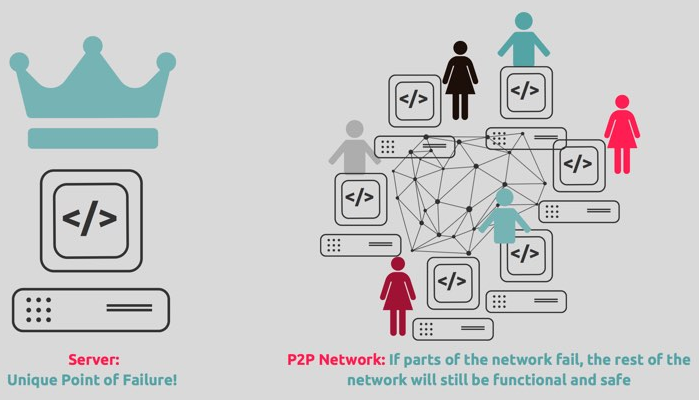
The key properties of smart contracts are:

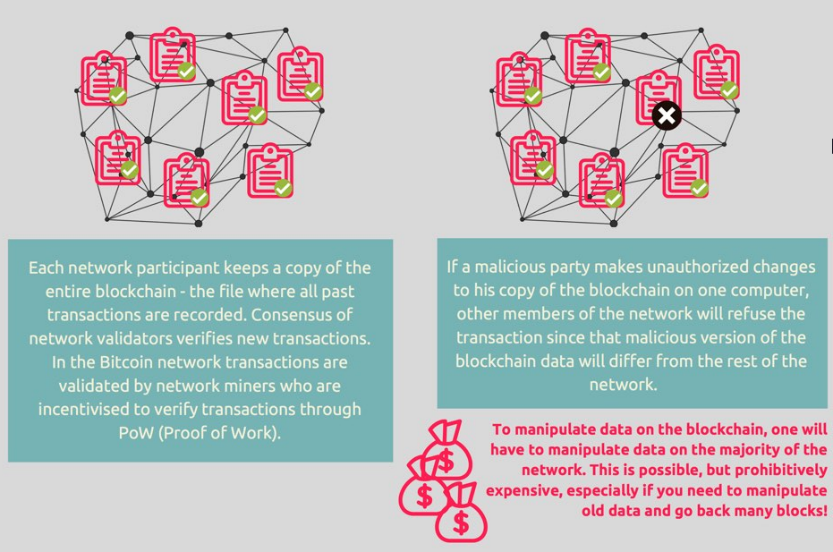
* Autonomy
* Decentralization
* Auto-sufficiency

# OBJECTIVES AND FEATURES

## Key benefits

Blockchain technology is appropriate for applications that focus on: **Decentralization**, **Security** and **Trust**.

 **-** **Decentralization**: Everyone who joins the network has to download the whole ledger and keep it up-to-date after a period of time (10 minutes in Bitcoin, 15 seconds in Ethereum). That means whenever there are new transactions, the ledgers must be updated to make them the same. There is no single point of failure in the network. Since someone ledger is lost or damaged, they can easily recover the ledger by downloading from the others. (**Robustness**)

 **- Security**: When a user requests a transaction, they have to broadcast to all nodes in the network and put it in a block. After being validated by other nodes, the block is added into the database like a chain of blocks. At this time, everyone will update their own database and there is no way to change or remove the transaction have made **(Immutability)**. If the database is not match with many other nodes, the ledger has to be synchronized with the others in the network in order to commit new transactions.

**- Trust:** Since everybody has a same database, users can access and check their transactions **(Transparency)**. That means no one can change or tamper the data behind the owner back. There is technique called Merkle Tree to detect which block has been changed.

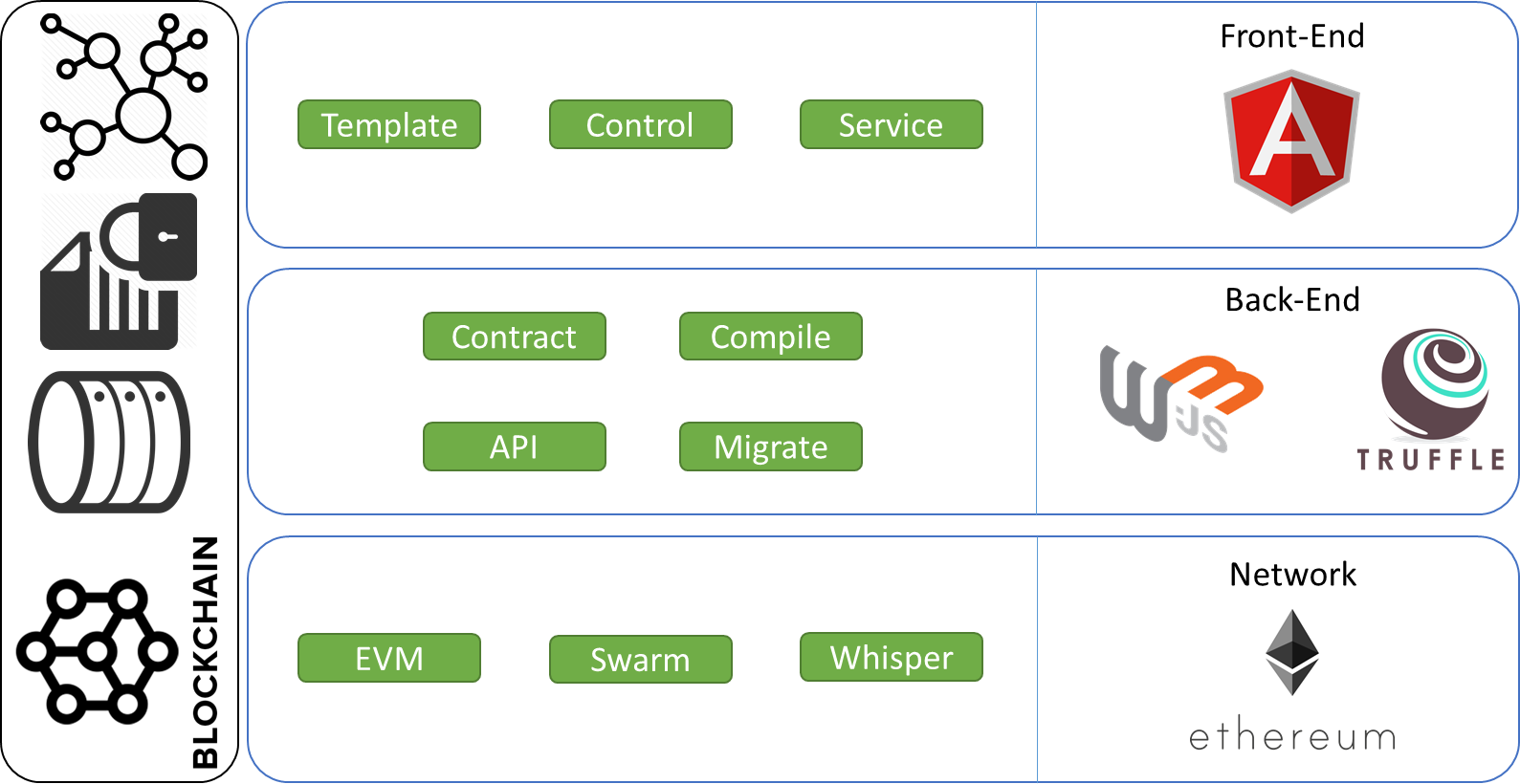
## Features of Product

* Create Vote:
  + - Information: Title, Detail, Date Create, Date End, Description and Time Remaining.
    - Multiple Question and Answer.
    - Types: Public and Private.
    - Restriction: Anyone can vote in Public and only in the invited list can join and take action in Private; One-person-one-vote.
    - Close the room automatically.
* Login with ID:
  + - Able to create/login account with ID instead of Address.
    - Confirm by login when submitting transaction.
* Check History:
  + - Show everything the user had done since he/she joins in the network. (Create, Vote)
* Check Result:
  + - Appear on only if the vote is closed, in general and detail.

## Functional Scope

* Product:
  + Runnable under private network, at least 10 nodes.
  + All feature work correctly without error.
* Document:
  + Research about Blockchain technology.
  + Research about Ethereum platform.
  + Ethereum Technique.
  + Environment.
  + User Guide.
  + UI/UX Design.
  + System Design.
  + Step-By-Step development in Ethereum.

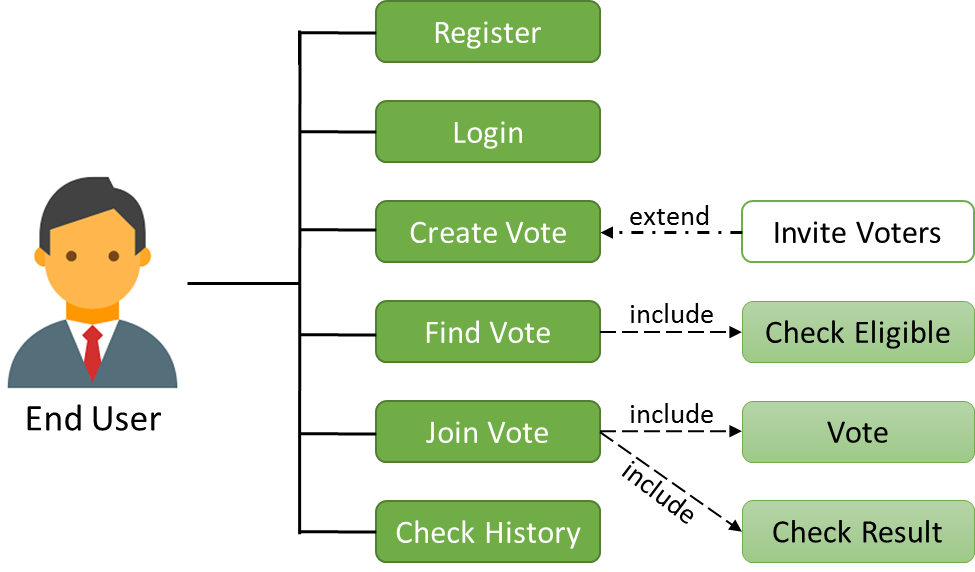
# SYSTEM ARCHITECTURE DIAGRAM



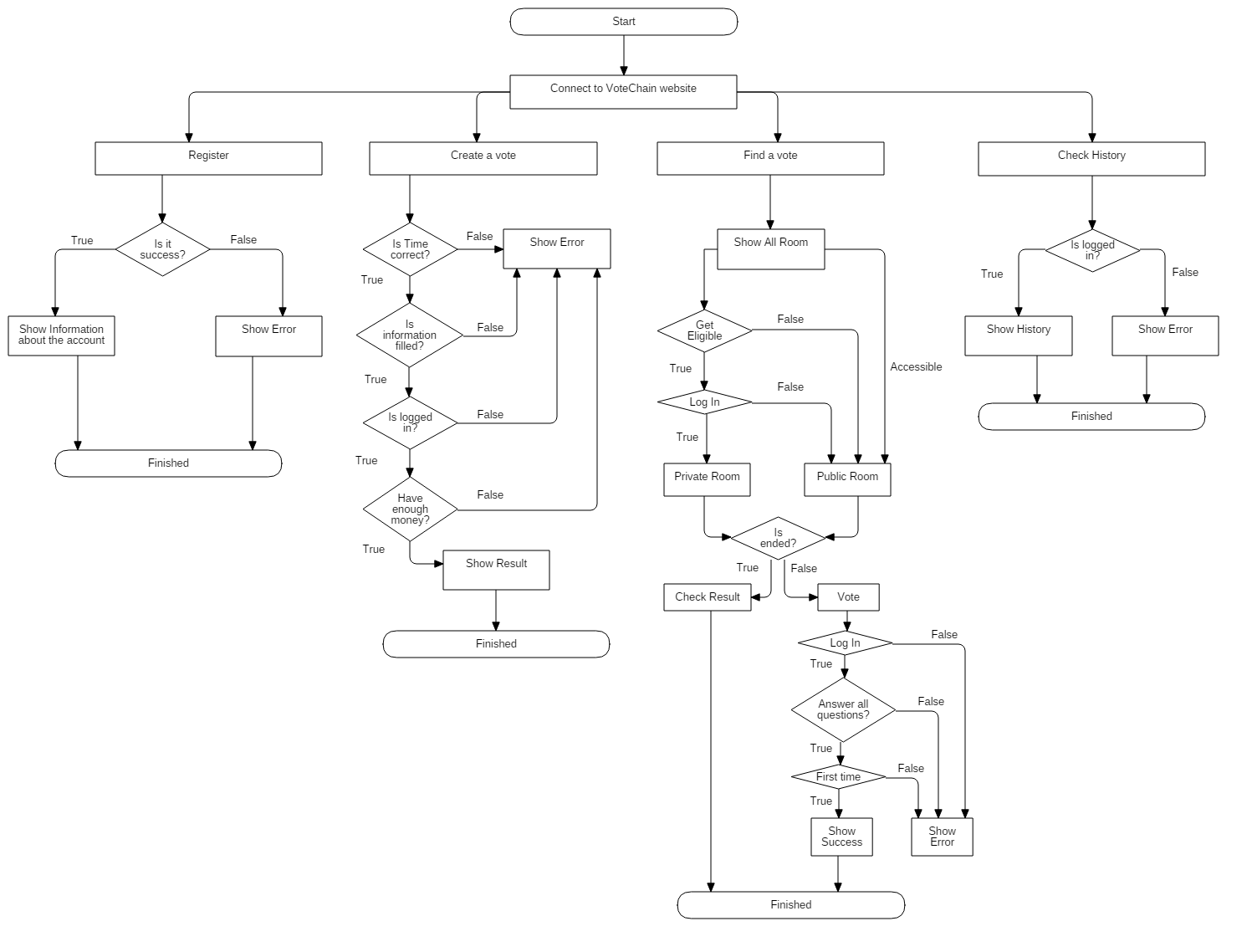
Layer architecture

# UML DIAGRAMS

## Use Case Diagram

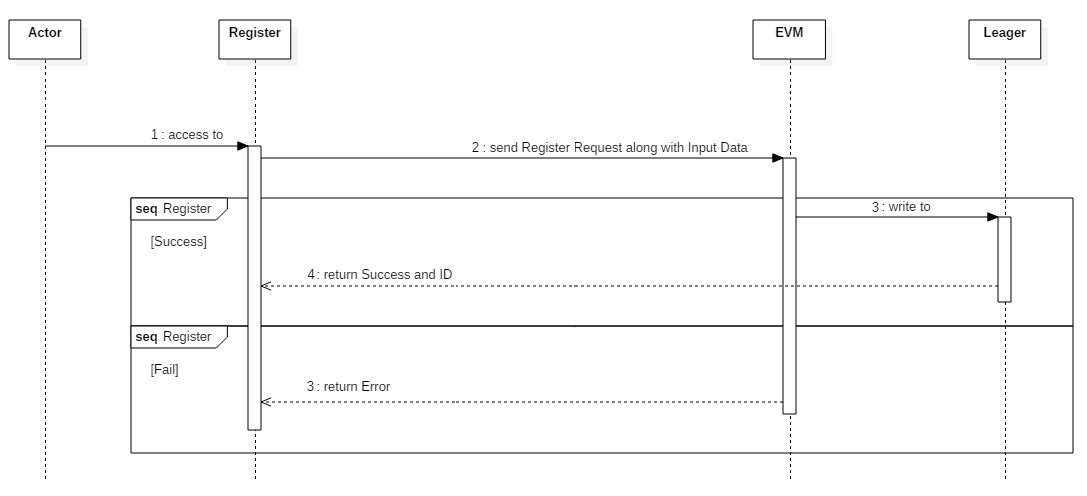


## Activity Diagram

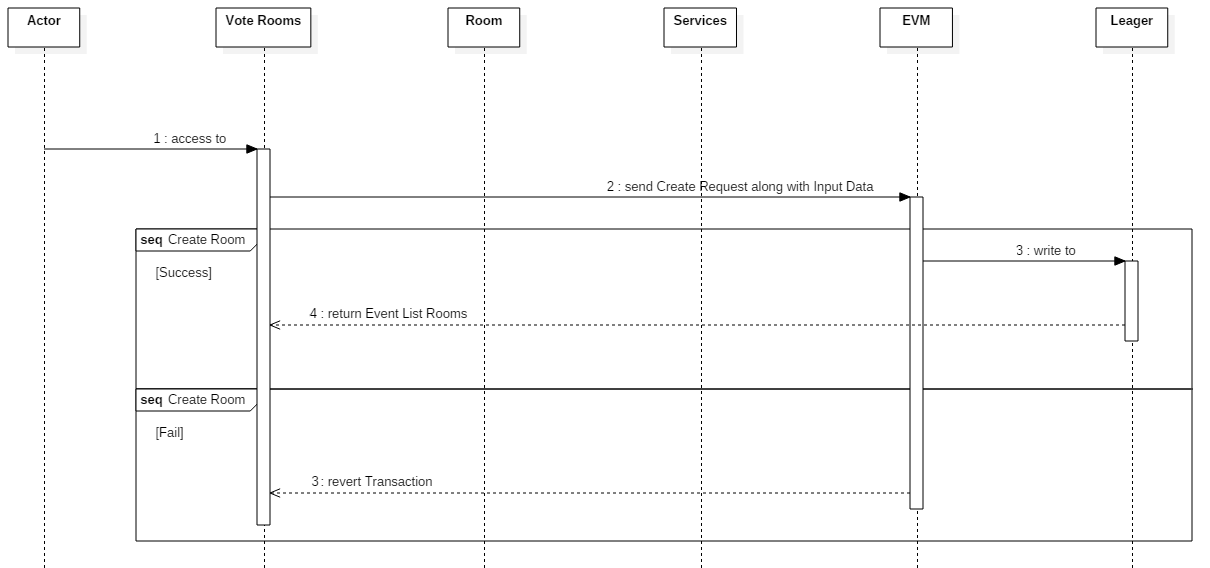


## Sequence Diagram

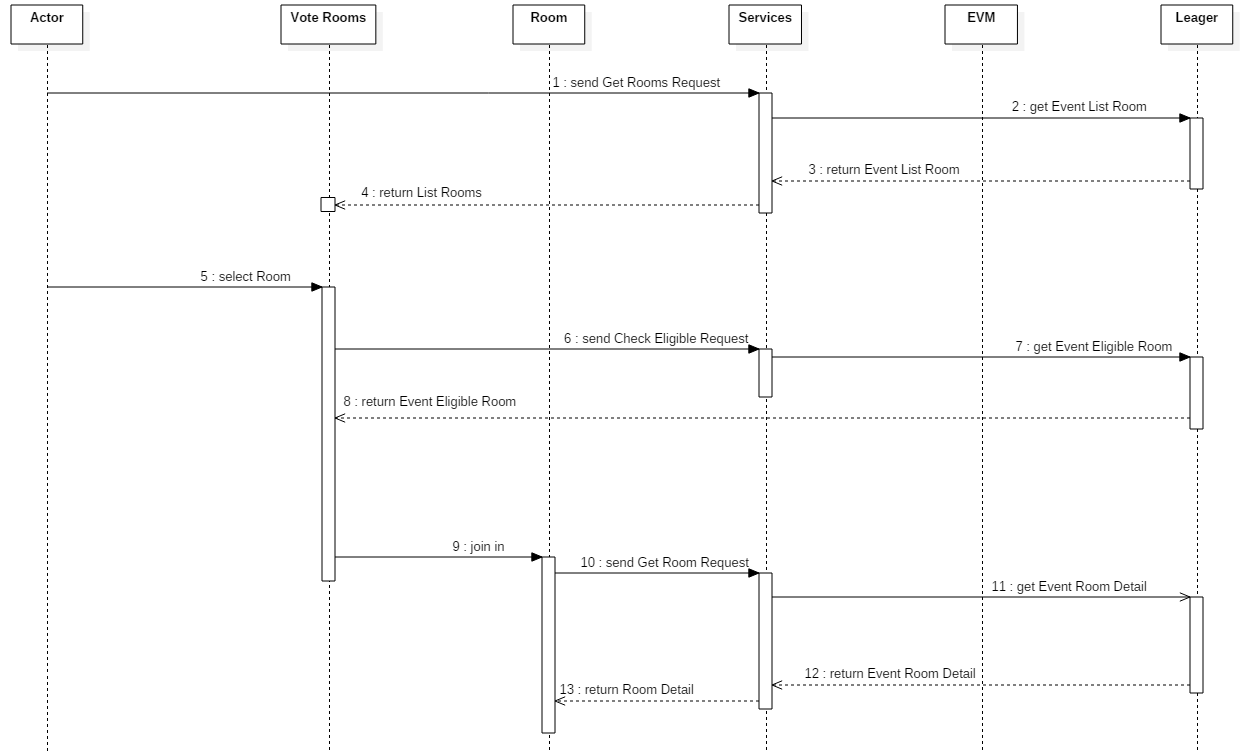
Register



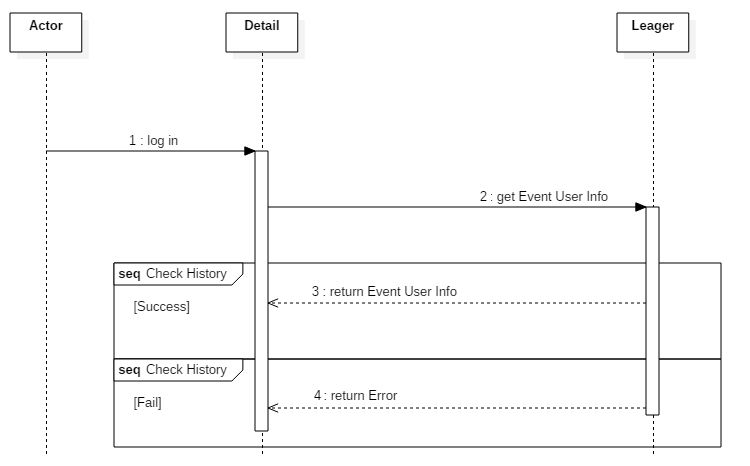
Create Room



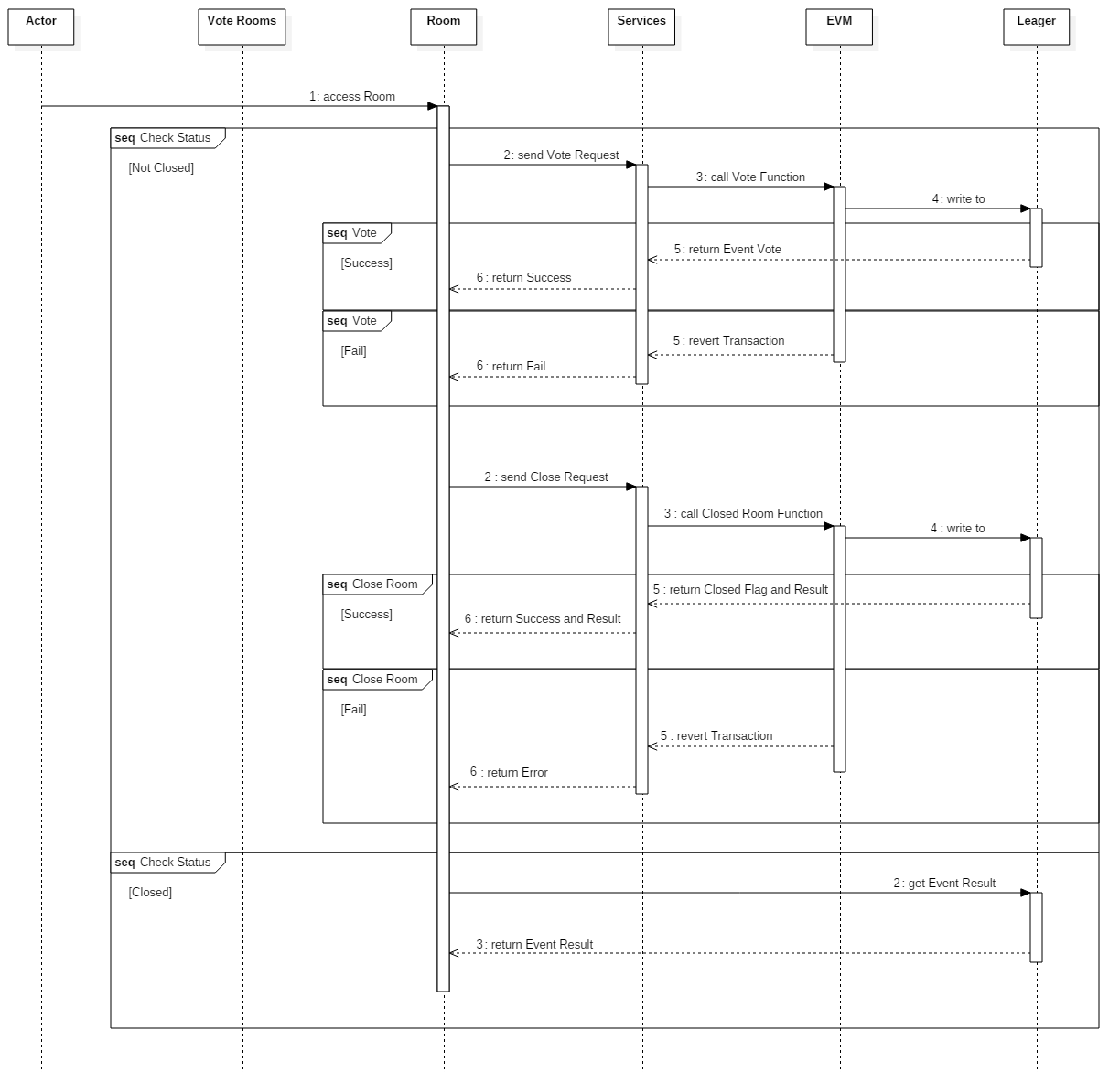
Get Rooms and Find Eligible Room



Check History



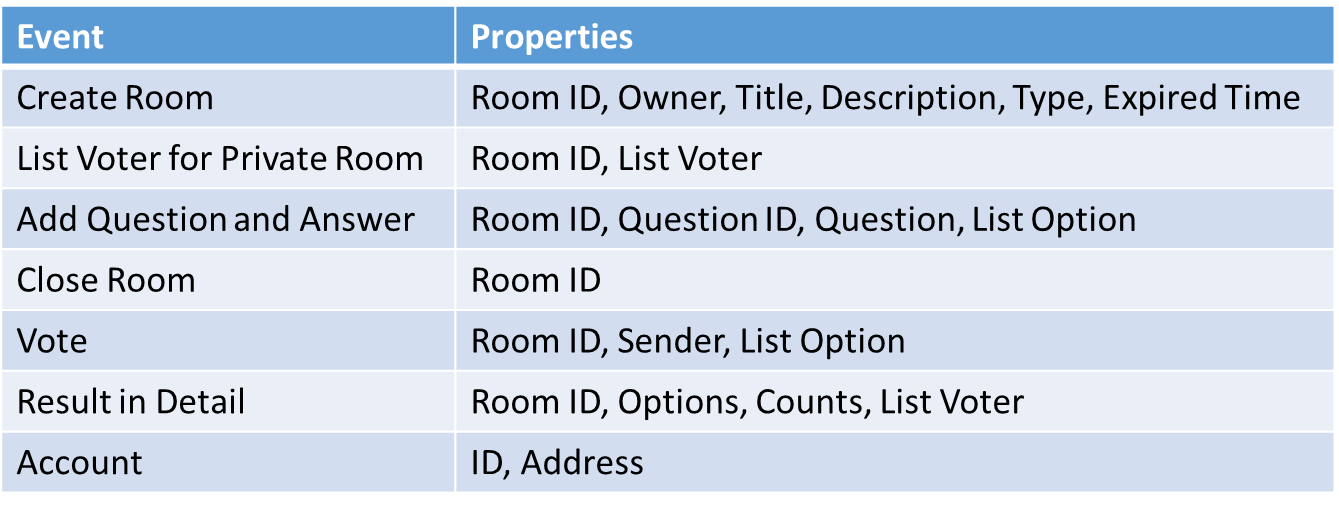
Vote and Check Result



# SYSTEM DESIGN

Blockchain is considered as a Distributed Database and we decided to use Event (such as log in Server-Client with slightly different) to manage and show information to the user (save gas and quickly access).

Event:



Site Map:

