# Title of Culminating Experience

A Choose an item. submitted to the faculty of  
San Francisco State University  
In partial fulfillment of  
the requirements for  
the Degree

Click to choose Degree  
In  
Click to choose Major: Concentration

by  
  
First Middle Last Name  
  
San Francisco, California  
  
Click to choose Semester 20XX

Copyright by  
First Middle Last Name  
20XX

## Certification of Approval

I certify that I have read Title of Culminating Experience by First Middle Last Name, and that in my opinion this work meets the criteria for approving a thesis submitted in partial fulfillment of the requirement for the degree (Example) Master of Science in Biology: Conservation Biology at San Francisco State University.

Name of the professor,Choose degree

Click to choose the faculty rank,

Thesis Committee Chair

Name of the professor,Choose degree

Click to choose the faculty rank

Name of the professor, Choose degree

Click to choose the faculty rank

Name of the professor,Choose degree

Click to choose the faculty rank

## Abstract

Begin your abstract here.

Insert a section break at the end of the Abstract by placing your cursor after the final punctuation and going to Layout > Breaks > Next Page (under Section Breaks). For corresponding screenshots, please consult [Microsoft’s support page](https://support.microsoft.com/en-us/office/insert-a-section-break-eef20fd8-e38c-4ba6-a027-e503bdf8375c).

## Preface

Including a Preface for your work is optional. If you do not have one, please delete this page.

[Insert a **page break** at the end of the Preface by placing your cursor after the final punctuation and going to Insert > Page Break.}

## Acknowledgements

Normally the Acknowledgements section gives you the opportunity to thank those to whom you are indebted for assistance in the project. This should be done briefly and in good taste.

[Insert a **page break** at the end of the Acknowledgements by placing your cursor after the final punctuation and going to Insert > Page Break.]

## Table of Contents

[Introduction 1](#__RefHeading___Toc3155_3930847570)

[Motivation 1](#__RefHeading___Toc3173_3930847570)

[Solution 1](#__RefHeading___Toc3175_3930847570)

[Background and Related Work 2](#__RefHeading___Toc3177_3930847570)

[Security Frameworks 2](#__RefHeading___Toc3179_3930847570)

[Transport Layer Security 2](#__RefHeading___Toc3181_3930847570)

[Access Control 2](#__RefHeading___Toc3183_3930847570)

[Blockchain 2](#__RefHeading___Toc3185_3930847570)

[Ethereum 2](#__RefHeading___Toc3187_3930847570)

[Proof-of-Stake 2](#__RefHeading___Toc3189_3930847570)

[Smart Contract 2](#__RefHeading___Toc3191_3930847570)

[Minimal Proxy Contract 2](#__RefHeading___Toc3193_3930847570)

[Sign in with Ethereum 2](#__RefHeading___Toc3195_3930847570)

[Design and Architecture 3](#__RefHeading___Toc3197_3930847570)

[User Stories 3](#__RefHeading___Toc3199_3930847570)

[System Requirements 3](#__RefHeading___Toc3201_3930847570)

[Architecture 3](#__RefHeading___Toc3203_3930847570)

[Implementation 4](#__RefHeading___Toc3205_3930847570)

[UML Diagram 4](#__RefHeading___Toc3207_3930847570)

[Database Schema 4](#__RefHeading___Toc3209_3930847570)

[Code Structure 4](#__RefHeading___Toc3211_3930847570)

[Evaluation 5](#__RefHeading___Toc3213_3930847570)

[Setup 5](#__RefHeading___Toc3215_3930847570)

[Results 5](#__RefHeading___Toc3217_3930847570)

[Conclusion and Future Work 6](#__RefHeading___Toc3219_3930847570)

[Conclusion 6](#__RefHeading___Toc3221_3930847570)

[Limitation 6](#__RefHeading___Toc3223_3930847570)

[Future Work 6](#__RefHeading___Toc3225_3930847570)

[References 7](#__RefHeading___Toc3227_3930847570)

[Appendix/Appendices 9](#__RefHeading___Toc3229_3930847570)

[If your work **includes** a List of Tables, List of Figures, and/or List of Appendices, insert a **page break** at the end of the Table of Contents by placing your cursor at the bottom of the page and going to Insert > Page Break.

If your work **does not include** a List of Tables, List of Figures, and/or List of Appendices, insert a **section break** at the end of the Table of Contents by placing your cursor at the bottom of the page and going to Layout > Breaks > Next Page (under Section Breaks). For corresponding screenshots, please consult [Microsoft’s support page](https://support.microsoft.com/en-us/office/insert-a-section-break-eef20fd8-e38c-4ba6-a027-e503bdf8375c).]

## List of Tables

[If your work **includes** a List of Figures and/or List of Appendices, insert a **page break** at the end of the List of Tables by placing your cursor at the bottom of the page and going to Insert > Page Break.

If your work **does not include** a List of Figures and/or List of Appendices, insert a **section break** at the end of the List of Tables by placing your cursor at the bottom of the page and going to Layout > Breaks > Next Page (under Section Breaks). For corresponding screenshots, please consult [Microsoft’s support page](https://support.microsoft.com/en-us/office/insert-a-section-break-eef20fd8-e38c-4ba6-a027-e503bdf8375c).]

## List of Figures

[If your work **includes** a List of Appendices, insert a **page break** at the end of the List of Figures by placing your cursor at the bottom of the page and going to Insert > Page Break.

If your work **does not include** a List of Appendices, insert a **section break** at the end of the List of Figures by placing your cursor at the bottom of the page and going to Layout > Breaks > Next Page (under Section Breaks). For corresponding screenshots, please consult [Microsoft’s support page](https://support.microsoft.com/en-us/office/insert-a-section-break-eef20fd8-e38c-4ba6-a027-e503bdf8375c).]

## List of Appendices

[Appendix A: List of California State University Campuses 9](#__RefHeading___Toc3231_3930847570)

[Appendix B: Abbreviations of California State University Campuses 10](#__RefHeading___Toc3233_3930847570)

## 

[Insert a **section break** at the end of the List of Appendices by placing your cursor at the bottom of the page and going to Layout > Breaks > Next Page (under Section Breaks). For corresponding screenshots, please consult [Microsoft’s support page](https://support.microsoft.com/en-us/office/insert-a-section-break-eef20fd8-e38c-4ba6-a027-e503bdf8375c).]

## Introduction

Nowadays, electronic health records (EHRs) are controled by healthcare providers stored in their respective central databases. While EHRs offer many benefits, including improved patient care and increased efficiency, there are still difficulties in exchanging data while maintaining privacy and integrity.

### **Motivation**

#### Data Protection Legislation

The legislation on healthcare data protection varies depending on the locale. Two major pieces of legislation come to mind in terms of health care data, the EU General Data Protection Regulation (GDPR) and the US Health Insurance Portability and Accountability Act (HIPAA). Hotizontal reach (the public and provate domains the legislation regulate), vertical attributes (what data custodian behaviours the legislation regulate, and enforcement are the three key properties to measure data protection efforts. The GDPR provides comprehensive protection across both horizontal and vertical dimensions.

The GDPR provides fines and other penalties for organizations that fail to comply with its requirements, but it does not provide criminal penalties. HIPAA, with the addition of Health Information Technology for Economic and Clinical Health Act (HITECH), provides fines for non-compliance.

### Solution

In this project, the goal is to improve interoperability and to preserve privacy of EHRs using the blockchain technology.

## **Background and Related Work**

### **Security Frameworks**

#### ***Transport Layer Security***

#### Access Control

### Blockchain

### Ethereum

#### Proof-of-Stake

In a Proof-of-Stake (PoS) consensus, validators create new blocks and receive coins in return. Users can stake their coins to participate as a validator node. The amount of stake affects the chances of a node being selected. To prevent the system being biased towards the wealthiest node, additional mechanisms are added to the selection process. For example, some PoS system uses a Coin Age Selection mechanism where it takes in the factor of how long the coins of a node has been staked. The system will lean towards selecting the nodes with coins that are staked for longer. Once the node is selected the age of the coins will be reset, so there will be a cooldown period before the node is selected again.

Once a validator is selected, they receive new blocks from peers from the network. The validator then finds the block that contains the valid transactions and block signature, and sends a vote in favor of that block to the rest of the network. To prevent a validator from acting maliciously, the stake to validate a block must be higher than the transaction total in the block. If the validator approved fraudulent transaction, their stake along with the rewards will be burned.

###### Against 51% Attack

51% attack is an attack on a blockchain network by a group of miners who own more than 50% of the computing power. Since blockchain networks run on a majority rules democracy, owning over 50% of the network allows the attackers to reverse transactions, block transactions, changing the transaction history on the blockchain. Figure shows that at the time of writing, if two of the biggest mining pools merged, they would have control to the bitcoin network. In a PoS consensus, an attacker would need to own over 50% of the circulating supply of the network. Depending on the value of the currency for the network, attempting 51% attack on a PoS network would be infeasible.

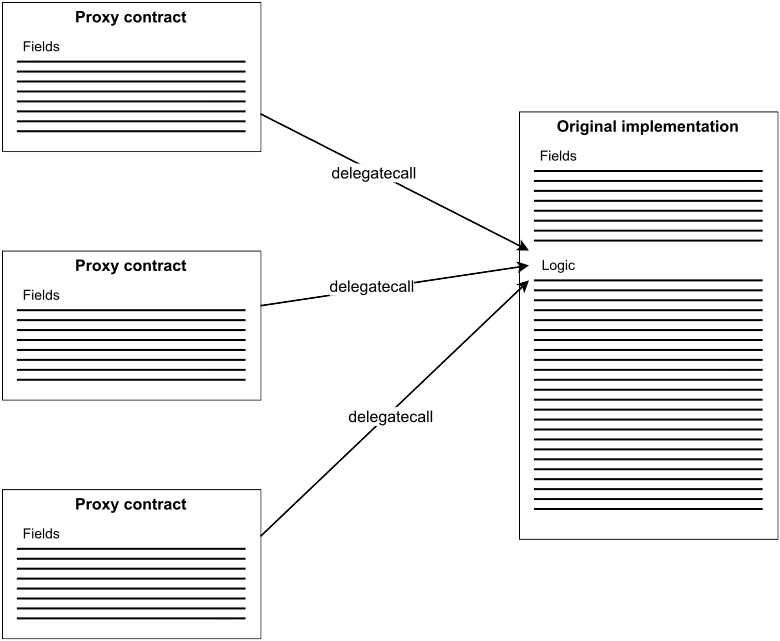
###### Decentralization

In a PoW consensus where miners with more computing power are more likely to win the race to create new blocks, participating in the mining process can be infeasible for most users. The cost of mining also increases as new blocks are added to the blockchain in PoW. Therefore, PoW consensus tends towards centralization. PoS consensus does not require large amount of computing power to create new blocks, which allows more users to setup validator nodes without purchasing expensive hardware. Therefore, a PoS system tends towards decentralization.

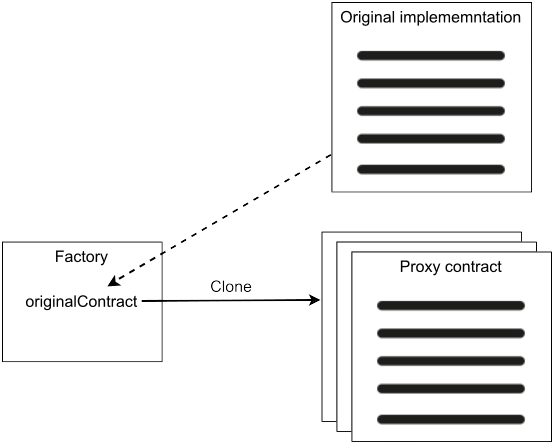
#### Smart Contract

#### Minimal Proxy Contract

Minimal Proxy Contract, also known as Clones, is proposed in Ethereum Improvement Proposals EIP-1167 which allows developers to simply and cheaply clone contract functionality in an immutable way. Deploying new contracts to the blockchain is an expensive operation. Certain contracts such as user profile, game session are required to be deployed multiple times, which causes an application to be expensive to use.

Figure Minimal Proxy Contract

Minimal Proxy Contract solves this problem using delegatecall, which allows a contract to execute functions of another contact in the context of the original contract. Developers can first deploy a factory contract that contains the original implementation contract they want to deploy multiple times. To create a copy of the implementation contract, the factory contract create a proxy contract to the implementation contract. This proxy contract only contains the fields of the implementation contract but not the logic. When a function of the proxy contract is called, the proxy contract will make a delegate call to the implementation contract. Since the context of the delegate call is in the proxy contract, the data will be modified in the proxy contract if needed, and the implementation contract will remain unchanged.



#### Sign in with Ethereum

Sign-in with Ethereum is proposed in Ethereum Improvement Proposal EIP-4361**,** which allows users to authenticate with their Ethereum wallet for off-chain services. Signing into off-chain services nowadays often turns to large centralized identity providers (IdPs) through OAuth or SAML. While this method eliminates the need to create new usernames and passwords, users’ identities are controlled by the IdPs, which leads to more aggregation of user data. Since Ethereum already provided users an built-in identifier, Ethereum wallet, sign-in with Ethereum is a self-custodial alternative to centralized IdPs.

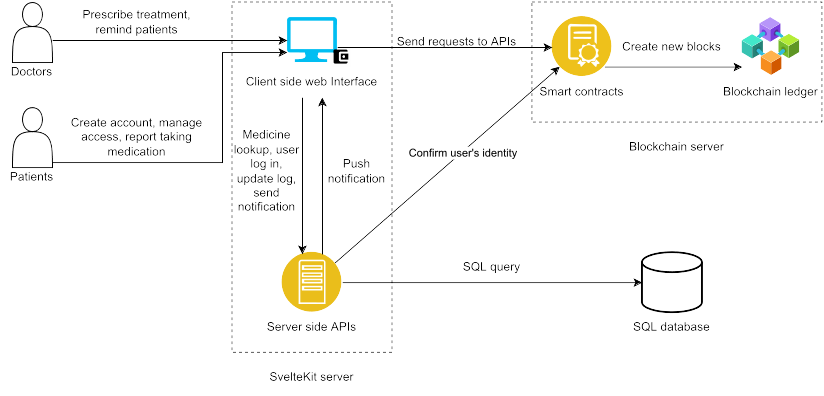
To initiate Sign-in with Ethereum, the Ethereum wallet first provides the user a structured message which contains an address, domain requesting the signing, version of the message, chain-id, uri for scoping, nonce as security mechanism, and issued-at time stamp. The user then sign the message with their private key and send it to the off-chain service. Finally, the off-chain service can validate the user identity with the signature in the message. Figure illustrates the sign-in with Ethereum workflow.

## **Design and Architecture**

### User Stories

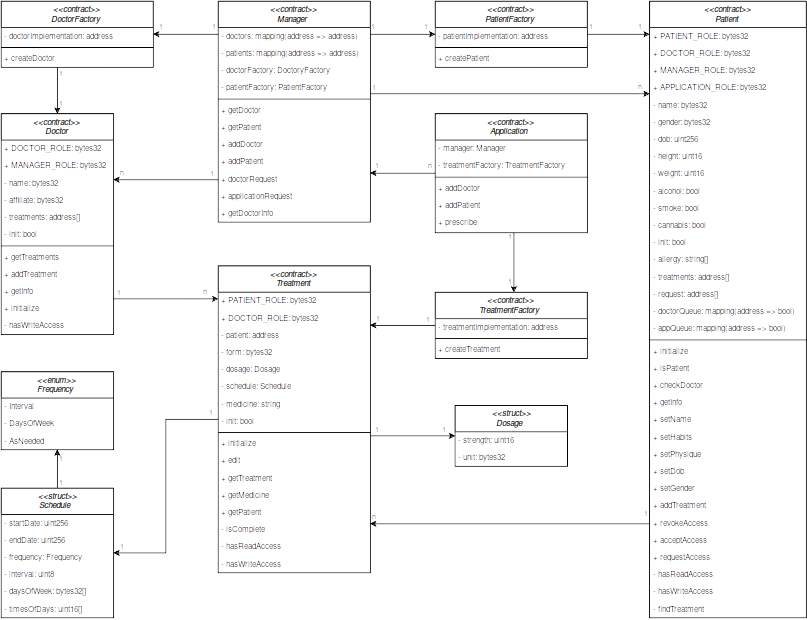
### System Requirements

### Architecture

Figure X shows the key components of our application including SvelteKit server, SQL database server, and Blockchain server. Users interact with the blockchain and server side APIs through client side web interface from SvelteKit server. As mentioned in earlier chapter, blockchain provides integrity and accessibility but it is inefficient. Traditional SQL database has better scalability, but it is prone to data corruption and outages. Our hybrid setup allows us to take advantage of the immutability of blockchain and efficiency of SQL database. Data that require integrity such as patient records and access control list are stored on the blockchain. Other data such as medication lookup and patient report logs are stored in SQL database. Here we are using the Ethereum implementation of blockchain server with smart contracts providing the logic and blockchain ledger as storage. For patient reporting their progress, only the patients' wallet address and the treatments' contract address are stored to preserve privacy. Server side APIs checks the access control on the treatment contract before making query to the SQL database.

## Implementation

### UML Diagram



### Database Schema

### Code Structure

## Evaluation

### Setup

### Results

## Conclusion and Future Work

### Conclusion

#### Limitation

### Future Work

## References

1. “Electronic Medical Records in Healthcare,” *HHS.gov*, Feb. 17, 2022. https://www.hhs.gov/sites/default/files/2022-02-17-1300-emr-in-healthcare-tlpwhite.pdf (accessed Jan. 06, 2023).
2. “Electronic Health Record Systems,” *HSS.gov*, Feb. 13, 2020. https://www.hhs.gov/sites/default/files/electronic-health-record-systems.pdf (accessed Jan. 06, 2023).
3. T. H. Friend, S. J. Jennings, and W. C. Levine, “Communication Patterns in the Perioperative Environment During Epic Electronic Health Record System Implementation,” *Journal of Medical Systems*, no. 2, Dec. 2016, doi: 10.1007/s10916-016-0674-3.
4. N. Terry, “Existential challenges for healthcare data protection in the United States,” *Ethics, Medicine and Public Health*, no. 1, pp. 19–27, Jan. 2017, doi: 10.1016/j.jemep.2017.02.007.
5. M. J. Mabee, “Healthcare Data Breaches in South Dakota: Post-Breach Legislation Is Not Enough,” *South Dakota Law Review, vol. 65, no. 3*, 2020, Accessed: Jan. 13, 2023. [Online].
6. D. B. Lafky and T. A. Horan, “Personal health records,” *Health Informatics Journal*, no. 1, pp. 63–71, Mar. 2011, doi: 10.1177/1460458211399403.
7. C. Thapa and S. Camtepe, “Precision health data: Requirements, challenges and existing techniques for data security and privacy,” *Computers in Biology and Medicine*, p. 104130, Feb. 2021, doi: 10.1016/j.compbiomed.2020.104130.
8. T. Li and T. Slee, “The effects of information privacy concerns on digitizing personal health records,” *Journal of the Association for Information Science and Technology*, no. 8, pp. 1541–1554, Apr. 2014, doi: 10.1002/asi.23068.
9. H. Qian, J. Li, Y. Zhang, and J. Han, “Privacy-preserving personal health record using multi-authority attribute-based encryption with revocation,” *International Journal of Information Security*, no. 6, pp. 487–497, Nov. 2014, doi: 10.1007/s10207-014-0270-9.
10. M. A. de Carvalho Junior and P. Bandiera-Paiva, “Strengthen Electronic Health Records System (EHR-S) Access-Control to Cope with GDPR Explicit Consent,” *Journal of Medical Systems*, no. 10, Aug. 2020, doi: 10.1007/s10916-020-01631-5.
11. P. Murray, N. Welch, and J. Messerman, “EIP-1167: Minimal Proxy Contract,” *Ethereum Improvement Proposals*, Jun. 22, 2018. https://eips.ethereum.org/EIPS/eip-1167 (accessed Jan. 13, 2023).
12. W. Chang, G. R. Rocco, B. M. Millegan, and N. Johnson, “EIP-4361: Sign-In with Ethereum,” *Ethereum Improvement Proposals*, Oct. 11, 2021. https://eips.ethereum.org/EIPS/eip-4361 (accessed Jan. 13, 2023).
13. “Ethereum Gas Tracker | Etherscan,” *Etherscan*. https://etherscan.io/gastracker (accessed Jan. 13, 2023).
14. “Pool Stats - BTC.com,” *BTC.com First website for blockchainers*. https://btc.com/stats/pool (accessed Jan. 13, 2023).
15. “Ethereum Price | ETH Price Index and Live Chart- CoinDesk,” *CoinDesk: Bitcoin, Ethereum, Crypto News and Price Data*. https://www.coindesk.com/price/ethereum/ (accessed Jan. 13, 2023).
16. “TESTNET  Sepolia (ETH) Blockchain Explorer,” *Ethereum (ETH) Blockchain Explorer*. https://sepolia.etherscan.io/ (accessed Jan. 13, 2023).
17. “Proof-of-stake (PoS) | ethereum.org,” *ethereum.org*. https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/ (accessed Jan. 13, 2023).
18. “Proof-of-work (PoW) | ethereum.org,” *ethereum.org*. https://ethereum.org/en/developers/docs/consensus-mechanisms/pow/ (accessed Jan. 13, 2023).
19. “Access Control - OpenZeppelin Docs,” *Documentation - OpenZeppelin Docs*. https://docs.openzeppelin.com/contracts/4.x/access-control (accessed Jan. 13, 2023).
20. “Proxies - OpenZeppelin Docs,” *Documentation - OpenZeppelin Docs*. https://docs.openzeppelin.com/contracts/4.x/api/proxy#minimal\_clones (accessed Jan. 13, 2023).
21. OpenZeppelin, “Cheap Contract Deployment Through Clones,” Mar. 02, 2021. https://www.youtube.com/watch?v=3Mw-pMmJ7TA (accessed Jan. 13, 2023).
22. “OWASP Top Ten | OWASP Foundation,” *OWASP Foundation, the Open Source Foundation for Application Security | OWASP Foundation*. https://owasp.org/www-project-top-ten/ (accessed Jan. 13, 2023).

## Appendix/Appendices

Appendix A: List of California State University Campuses

California State University, Bakersfield  
California State University Channel Islands  
California State University, Chico  
California State University, Dominguez Hills  
California State University, East Bay  
California State University, Fresno  
California State University, Fullerton  
Humboldt State University  
California State University, Long Beach  
California State University, Los Angeles  
California State University Maritime Academy  
California State University, Monterey Bay  
California State University, Northridge  
California State Polytechnic University, Pomona  
California State University, Sacramento  
California State University, San Bernardino  
San Diego State University  
San Francisco State University  
San José State University  
California Polytechnic State University, San Luis Obispo  
California State University San Marcos  
Sonoma State University  
California State University, Stanislaus

Appendix B: Abbreviations of California State University Campuses

CSU Bakersfield

CSU Channel Islands

Chico State

CSU Dominguez Hills

Cal State East Bay

Fresno State

Cal State Fullerton

Humboldt State

Cal State Long Beach

Cal State LA

Cal Maritime

CSU Monterey Bay

CSUN

Cal Poly Pomona

Sacramento State

Cal State San Bernardino

San Diego State

San Francisco State

San José State

Cal Poly San Luis Obispo

CSU San Marcos

Sonoma State

Stanislaus State