MICHIGAN STATE UNIVERSITY

Project Plan

Automatic Resume Verification

The Capstone Experience

Team Yello

Giorgio Maroki
Ryan Nagy
Nathaniel Hagan
Brandon Burt
Wan Kim

Department of Computer Science and Engineering
Michigan State University

Fall 2017

From Students…
…to Professionals

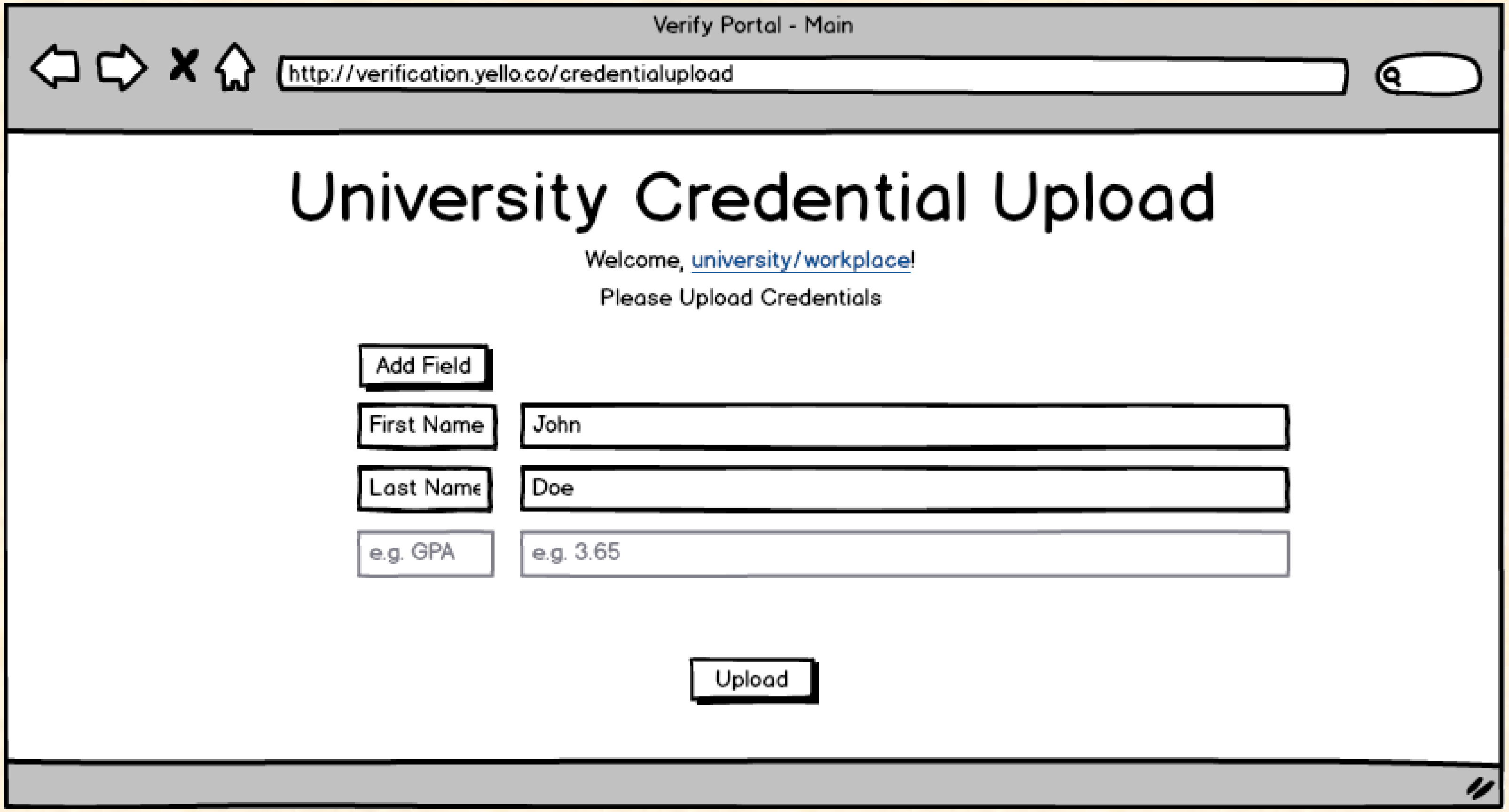
Functional Specifications

- Automatically verifying resume credentials.
- Submit credentials to a web application.
- Utilizes blockchain technologies
- Protects user data through one-way hashing.

Design Specifications

- The first web application will be where credential submission takes place.
- University/workplace receives a transaction ID that's passed along to student.
- Student uploads resume to first web app along with transaction IDs received from university/workplace, receives modified resume file that can be automatically verified
- The second web application will mimic Yello's internal application for job postings.
- After uploading a resume received from the first web application, the second web app automatically verifies the credentials listed on the resume

Screen Mockup: Verification Web App



Verify Portal - Main

http://verification.yello.co/credentialupload

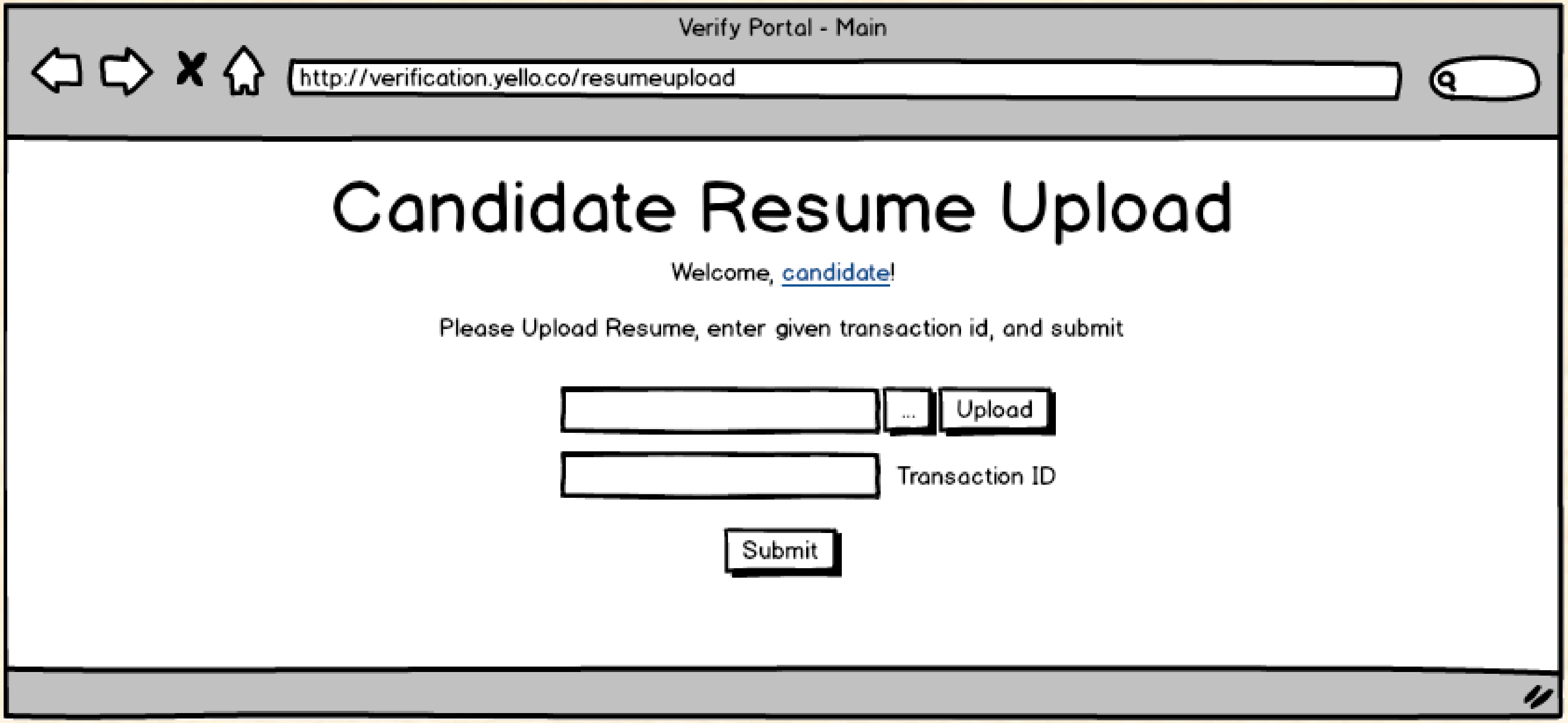
University Credential Upload

Welcome, university/workplace!

Please Upload Credentials

First Name	John
Last Name	Doe
e.g. GPA	e.g. 3.65

Screen Mockup: Verification Web App



Verify Portal - Main

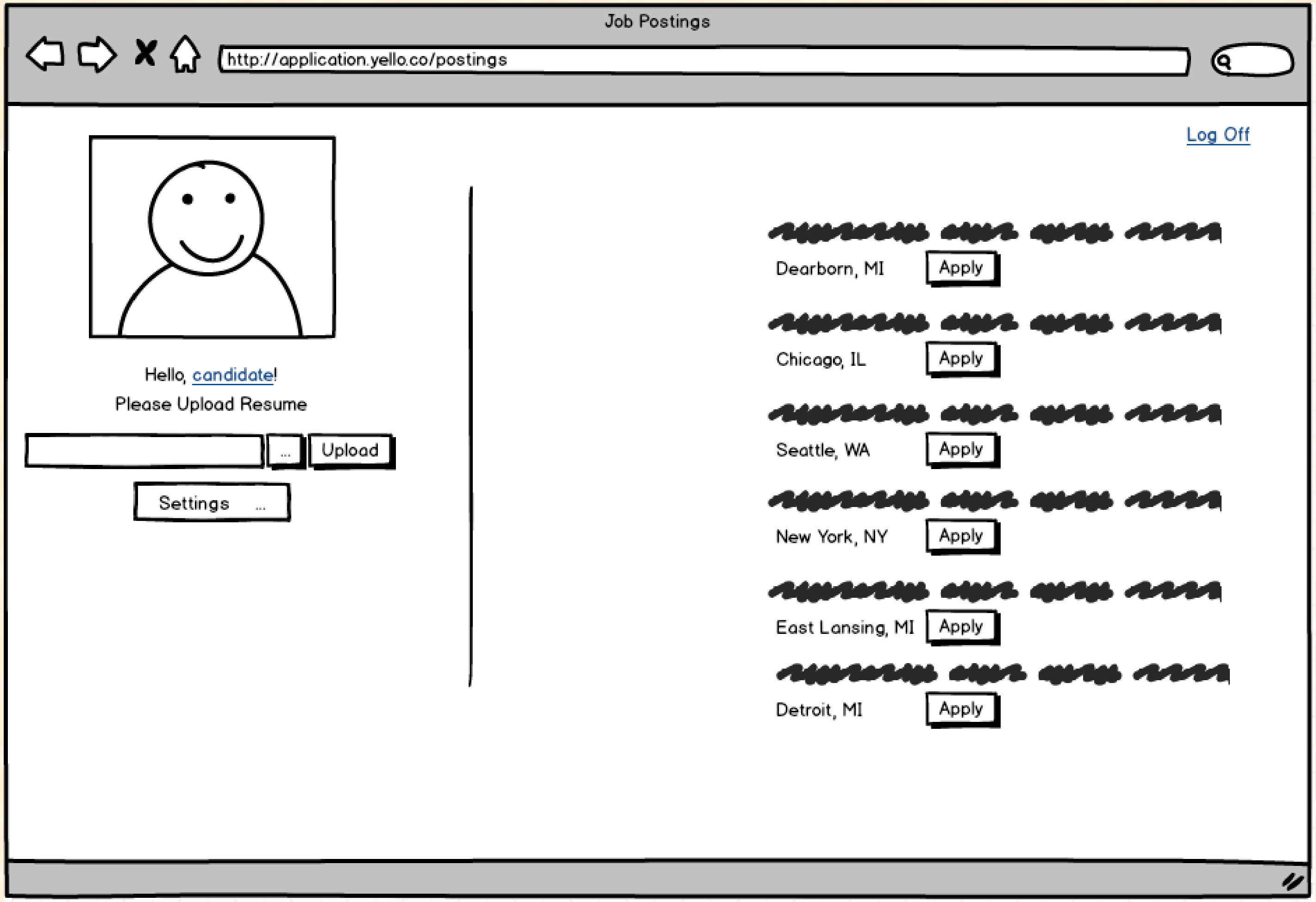
http://verification.yello.co/resumeupload

Candidate Resume Upload

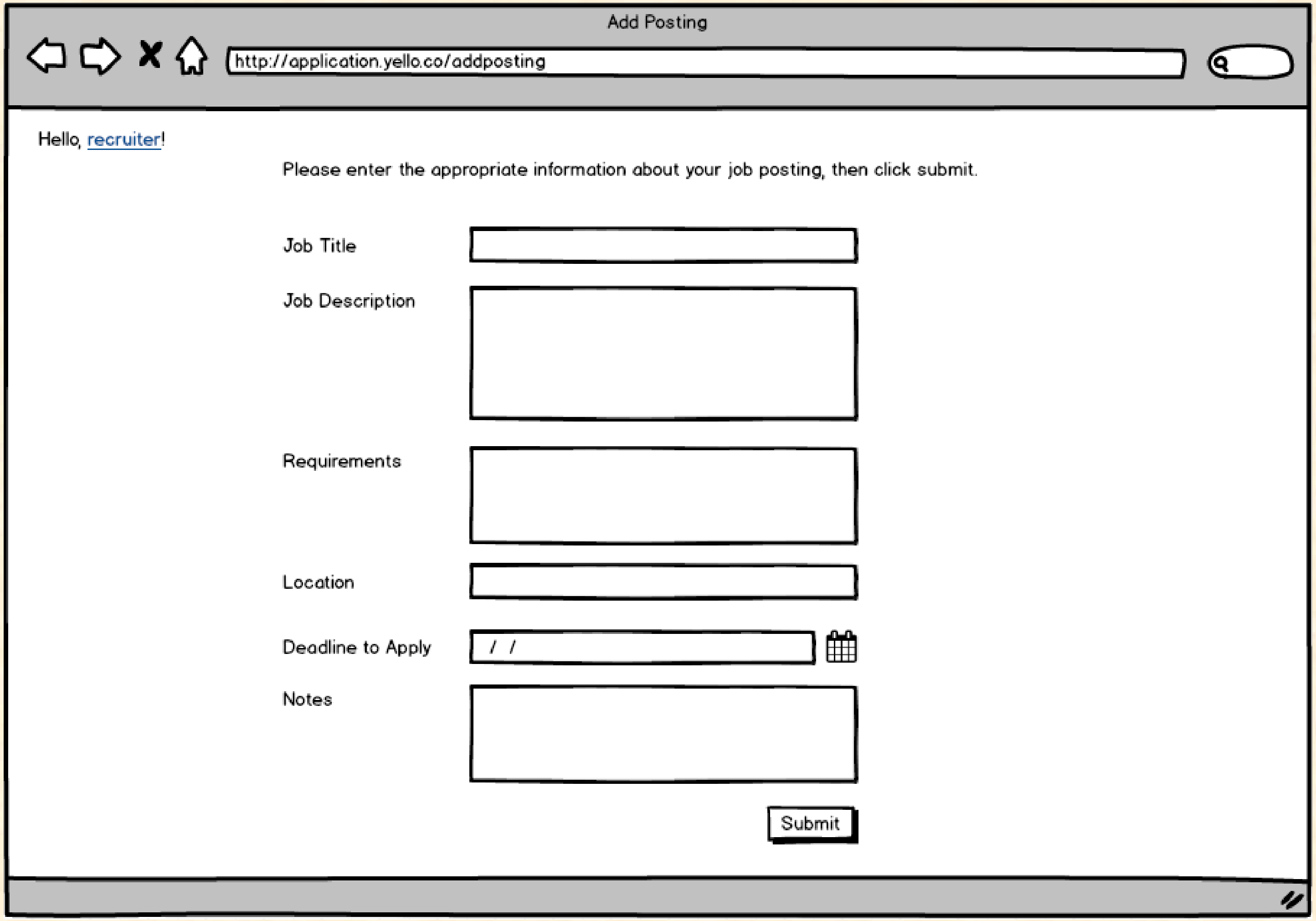
Welcome, candidate!

Please Upload Resume, enter given transaction id, and submit

Screen Mockup: Application Web App



Screen Mockup: Application Web App



Add Posting

http://application.yello.co/addposting

Hello, recruiter!

Please enter the appropriate information about your job posting, then click submit.

Job Title

Job Description

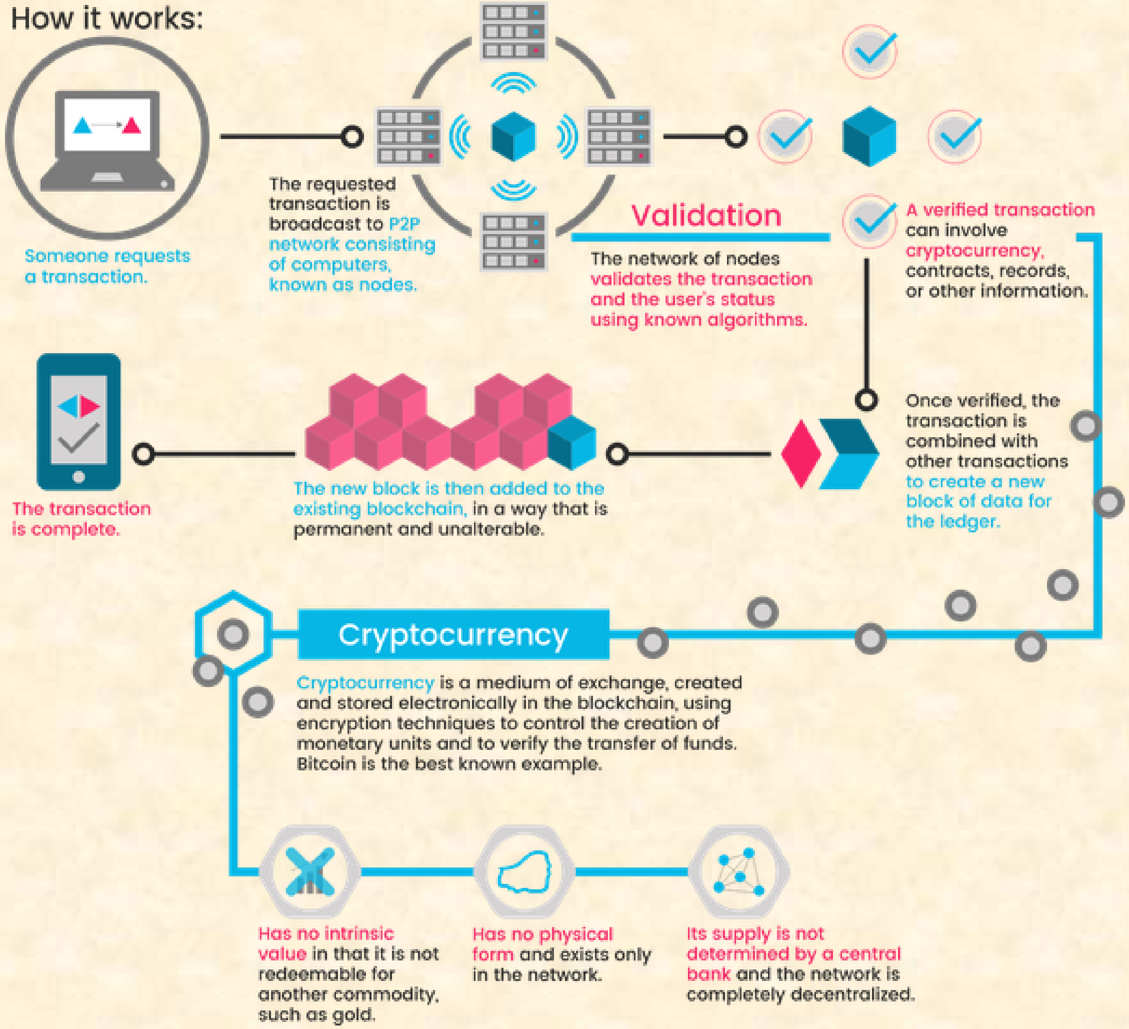
Requirements

Location

Deadline to Apply / /

Notes

Blockchain



How it works:

Someone requests a transaction.

The requested transaction is broadcast to P2P network consisting of computers, known as nodes.

Validation

The network of nodes validates the transaction and the user's status using known algorithms.

A verified transaction can involve cryptocurrency, contracts, records, or other information.

Once verified, the transaction is combined with other transactions to create a new block of data for the ledger.

The new block is then added to the existing blockchain, in a way that is permanent and unalterable.

The transaction is complete.

Cryptocurrency

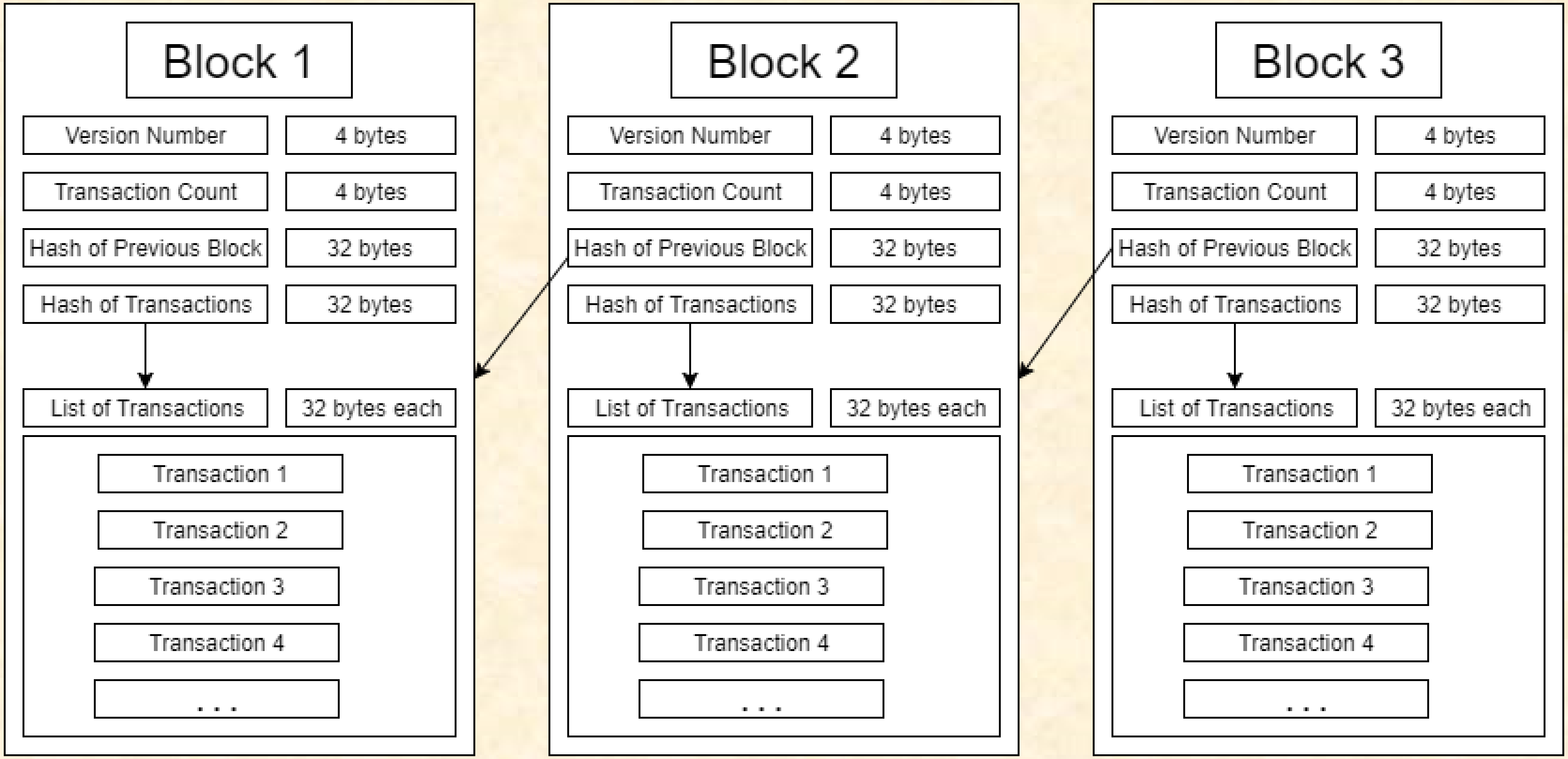
Cryptocurrency is a medium of exchange, created and stored electronically in the blockchain, using encryption techniques to control the creation of monetary units and to verify the transfer of funds. Bitcoin is the best known example.

Has no intrinsic value in that it is not redeemable for another commodity, such as gold.

Has no physical form and exists only in the network.

Its supply is not determined by a central bank and the network is completely decentralized.

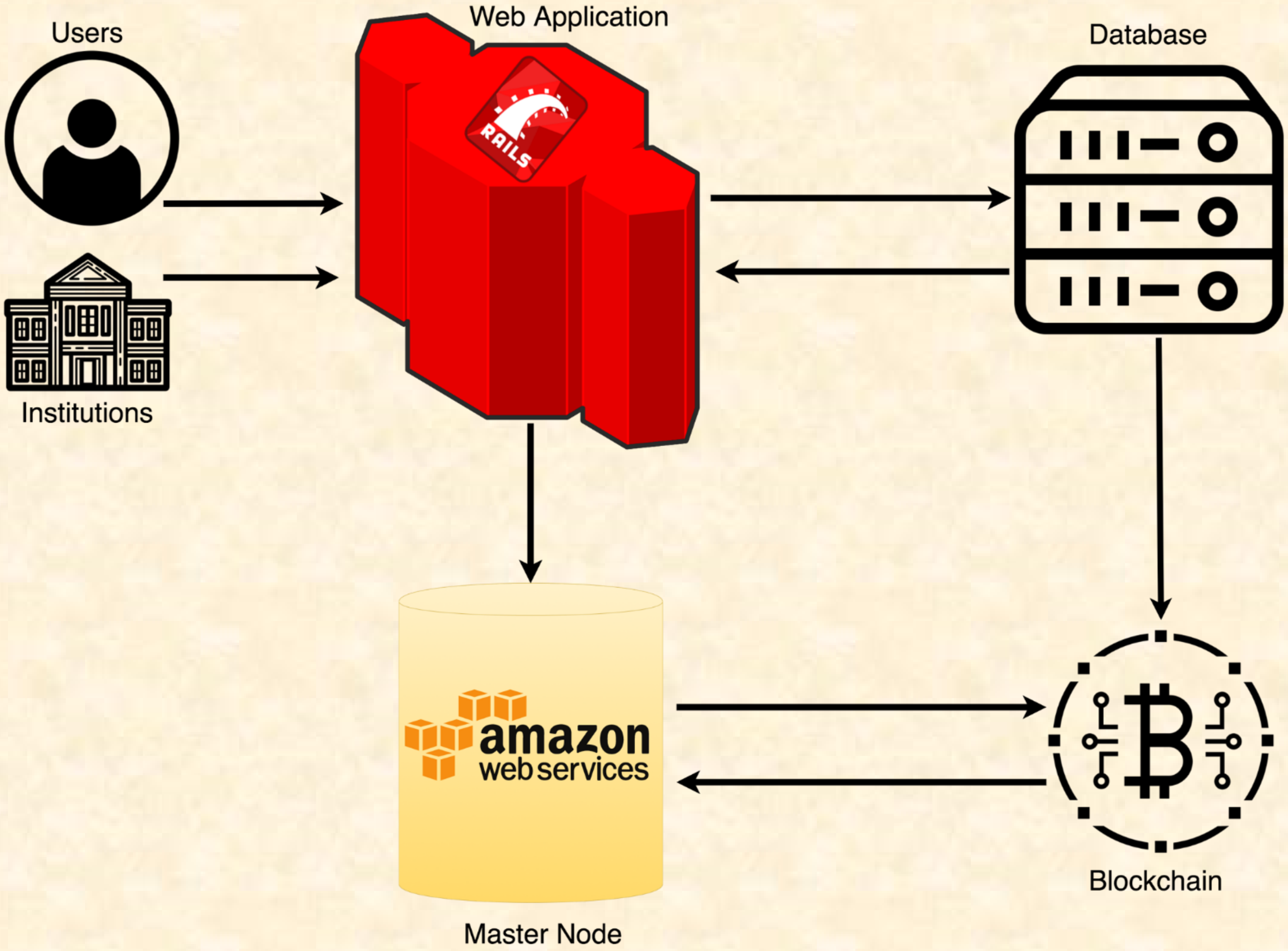
Blockchain


Block 1
Version Number
4 bytes
Transaction Count
4 bytes
Hash of Previous Block
32 bytes
Hash of Transactions
32 bytes
List of Transactions
32 bytes each
Transaction 1
Transaction 2
Transaction 3
Transaction 4
. . .
Block 2
Version Number
4 bytes
Transaction Count
4 bytes
Hash of Previous Block
32 bytes
Hash of Transactions
32 bytes
List of Transactions
32 bytes each
Transaction 1
Transaction 2
Transaction 3
Transaction 4
. . .
Block 3
Version Number
4 bytes
Transaction Count
4 bytes
Hash of Previous Block
32 bytes
Hash of Transactions
32 bytes
List of Transactions
32 bytes each
Transaction 1
Transaction 2
Transaction 3
Transaction 4
. . .

Technical Specifications

- Master Node (Python 3.6)
  - Proof of source
  - New transactions
- Database (Python 3.6, SQLite)
  - SQLite
  - Imports blockchain
- Web Application (Ruby on Rails)
  - 2 web portals
  - AWS Server

System Architecture



System Components

- Hardware Platforms
  - AWS Server Instance (Cloud)
- Software Platforms / Technologies
  - Ruby on Rails
  - SQLite
  - Custom Blockchain
  - Python 3.6

Testing

- Unit Testing for all Python Code
- Stress Test functional network
- Rails testing platform

Risks

- Scalability and Security
  - Proof of source and general security requirements given by the client are not possible to meet.
  - Problems were discussed with the client and goals have been reevaluated to be doable.
- Custom Blockchain
  - Originally the client requested Ethereum be used as the blockchain which is high cost per transaction.
  - A custom built blockchain has been built and demonstrated to client which reduces cost.
- Unfamiliar Web Framework
  - No one in the group is familiar with the web development requirements for this project. Between group members, we have limited experience tying together multiple frameworks.
  - Documentation for Ruby on Rails and the web development process in general have been distributed to the group for study.
- Unfamiliar with API
  - The group as a whole has little to no experience with building APIs. The client has requested a transactional API be built for our platform.
  - By creating a custom blockchain instead of using the Ethereum blockchain, the need for an API has been avoided completely.

Questions?

? ? ? ?

? ?

? ? ?