940 Blanshard Street
Victoria, British Columbia
V8W 3E6
Susan Fiddler
Co-op Coordinator
Faculty of Engineering
University of Victoria
P.O. Box 1700
Victoria, B.C.
V8W 2Y2

Dear Susan,

Please accept the accompanying Work Term Report entitled "Determining uses of JIRA and Confluence at OFI IBM."

This report is the result of work completed at the SOME GENEriC ORGANIZATION. During my first work
term as a University of Victoria student, I used charts and tables to display information about issue, complied documentation for critical applications in a wiki and researched add-ons to extend functionality. In the course of work, I gained exposure to a technical environment, and learned how software can integrate together.

Through the course of the term, I was given the opportunity to learn much agile software development, testing applications, and software products. I feel that this knowledge will be helpful in future work terms, and in my career.

I would like to thank my manager, MISTER MAN, for his patience and good judgement, as well as the RANDOM FOLK who were always willing to help.
Sincerely,

David Li

Table of Contents

List of Figures

List of Tables

Summary

Glossary

blockchain A blockchain is a digitized, decentralized, public ledger of all cryptocurrency transactions. 1

Dapp Decentralized Application have backend code running on a decentralized peer-to-peer network and not controlled by a single entity. In a decentralized application transactions are versified through consensus of multiple users. 1, 3

Ethereum an open software platform based on blockchain technology that enables developers to build and deploy decentralized applications 1, 3

HyperLedger group of open-source blockchain technologies started by the Linux Foundation 1

HyperLedger Composer permissioned blockchain that defines assets, business rules, and participants, and access controls for existing roles and types of transactions. 1

smart contract computer program that directly controls transfer of digital currencies or assets under predefined conditions and used to automatic transactions on the blockchain. These transactions are trackable and irreservable. 1

1. Introduction

1.1. History of Cryptocurrency

Table 1 Timeline of Cryptocurrency

2008	Bitcoin White Paper
2009	Bitcoin Genesis Block
2013	1 BTC = $ 31 USD
2013	Ethereum White Paper
2015	Ethereum Genesis Block
2015	HyperLedger starts
2017	Over 1000 different cryptocurrencies
2018	AWS Blockchain Templates

In 2008 bitcoin white paper [1] described a way to solve the double spending problem without a centralized body using blockchain. Although, the value of bitcoin (BTC) has grown exponentially, high computational and energy consumption in mining and slow performance [2]. Released in July 30, 2015, Ethereum, an open-source platform based on blockchain technology, distinguishes itself from bitcoin through faster transactions, unlimited processing capability for smart contracts, and its network is optimized to support Decentralized Application(DApp) [3].

1.2. Decentralized Applications

Blockchain technology is revolutionizing the internet by establishing trust in shared data. [4]. Additionally, transactions recorded on the blockchain are practically impossible to remove or change. A decentralized application, or DApp are deployed on peer to peer networks such as Ethereum or on the cloud [1]. A decentralized system (peer to peer) has many advantages over a conventional centralized network including no single points of failure, cheaper distribution (servers are expensive), faster upload speeds.

Types of Blockchains

— **Public blockchains** are large distributed networks that are run through a native token such as bitcoin or ether. Anyone can participate and the community maintains its open-source code. The two largest public blockchains are Ethereum and Bitcoin.

— **Permissioned blockchains** define role based access control for individuals in the network and uses native tokens. HyperLedger Composer, an open-source framework for permissioned blockchains, is used for smart contracts and for blockchain application development [5]. One use case is an accounting system that calculates payment, while hiding that information from unrelated organizations.

[1] Amazon recently started offering blockchain templates on AWS.

— **Private blockchains** membership is tightly controlled and lacks a native token. Useful for consortiums with trusted associates and exchanging confidential information, however, less powerful because it is supported by limited private resources.

Public and Private Keys **In a blockchain** system, any key holder can use their private key to sign a piece of data. This results in a signature. In a Dapp, this can be used for:

1. Recovering the public key (ethereum account address) of the Author.
2. Verify if the raw data is identical using the Author's public key.

1.3. Smart Contracts

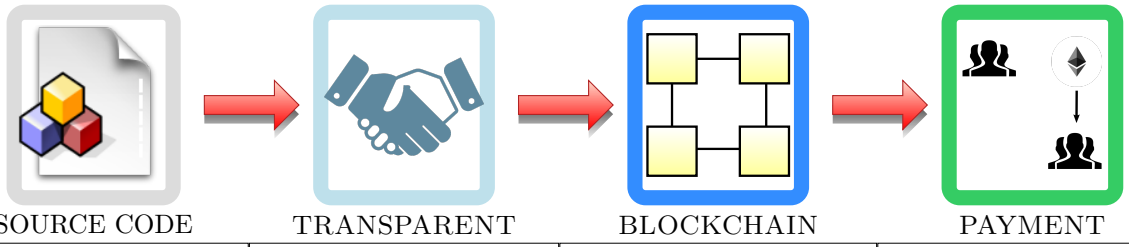
Traditional legal contracts are written to represent the contracting parties. In a smart contract, self-executing source code is used to automatic transactions that are publicly available on the blockchain. Furthermore, smart contracts allow buyers and sellers exchange money, property, shares, or anything of value in a transparent, conflict-free way while avoiding the services of a middleman. This allows validation of complex transactions swiftly while maintaining transparency. Although the benefits of using smart contracts are obvious, legal enforceability is difficult because "no central administering authority to decide a dispute" exists [6]. In addition, irreversible and immutable transactions are a disadvantage that hackers can exploit. For example, an amateur coder killed the contract that allowed users to transfer Ether for the Parity Ethereum Wallet, rendering 150 to 300 million dollars completely useless [7]. This highlights the need to scrutinize smart contracts and the frequency of bugs in production level code.

Listing 1: Smart contract in Solidity

```
// pragma is used to specifc
   version of Solidity
pragma solidity ^0.4.16;
contract HelloWorld {
 uint256 counter = 5;
 //state variable we assigned
    earlier
 address owner = msg.sender;
 //set owner as msg.sender
function add() public {  //
   increases counter by 1
    counter++;
 }
 function subtract() public { //
    decreases counter by 1
  counter--;
 }
 function getCounter() public
    constant returns (uint256) {
  return counter;
 }
 function kill() public { //self-
    destruct function,
   if(msg.sender == owner) {
    selfdestruct(owner);
   }
}

// Fallback function
function () public payable {

 }
```



SOURCE CODE	TRANSPARENT	BLOCKCHAIN	PAYMENT
self-executing code that faciliates agreement between parties	terms of contract are available on the public ledger	event triggers the execution of the contract (redeeming a ticket)	accounts are updated, parties recieve amounts based on contracts

Figure 1: Illustrating how a smart contract works

2. Discussion

As shown in Figure 2 Dapp, a user's transactions on the application is publicly broadcasting to the blockchain. Implementing architecture for blockchain applications [2] adds an third layer to the standard client-server architecture [3]. Disadvantages of blockchain data storage include difficult retrieving relevant information (without an abstraction layer, the entire blockchain or a single transaction is returned), users will experience latency before transactions are validated, [4] and writing to the blockchain is relatively expensive compared to traditional systems. Usage of interfaces such as the JSON RPC and/or cloud hosting solutions serve as a abstraction layer allowing databases to load publicly available data on the blockchain for more efficient user interactions with that information.

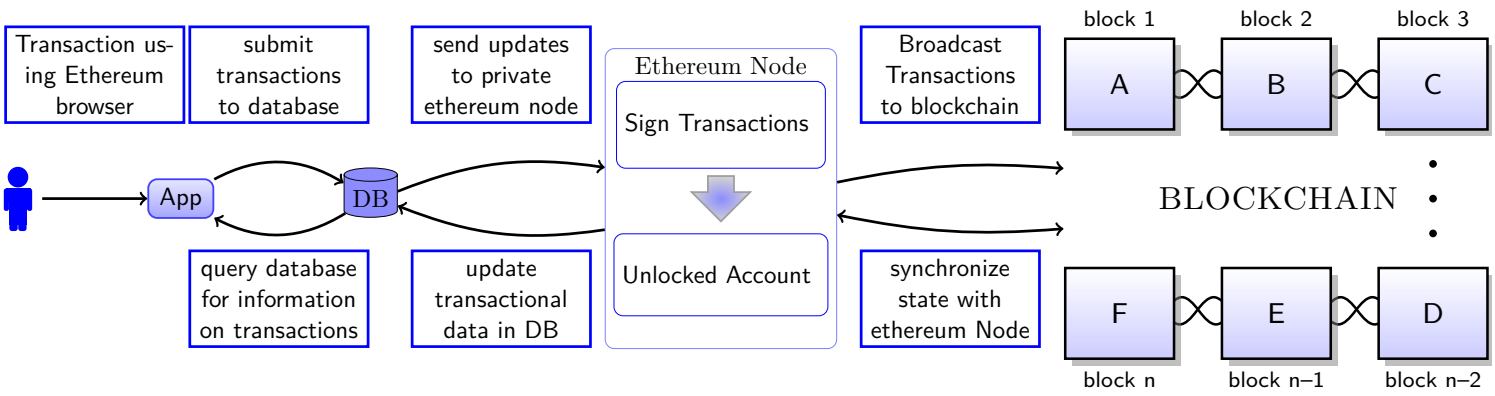


Figure 2: An example of server-blockchain architecture in a DAPP.

[2] Although, server blockchain architecture with an abstraction layer resemble traditional applications, other approaches are available such as offline signing with a public node, and client-blockchain in serverless apps and leveraging cloud infrastructure.

[3] **Signing Transactions**: One approach involves interacting with the JSON RPC interface of the Ethereum node from the application to perform all blockchain operations.

[4] For bitcoin, it takes 10 minutes before blocks of transactions are validated (mining process)

Table 2: Sample Decision Matrix for designing a blockchain system

	Existing Systems	BlockChain Systems		
Criteria	Centralized	Public	Permissioned	Private
speed and latency				
scale and volume				
security and immutablity				
storage capacity				
transparency				
Total				

3. Conclusion

4. Recommendations

5. References

Cited References

[1] *Bitcoin White Paper*. [Online] Available: https://bitcoin.org/bitcoin.pdf. Accessed April 25, 2018.

[2] Saeed Elnaj. *The Problems With Bitcoin And The Future Of Blockchain*. [Online] Available: https://www.forbes.com/sites/forbestechcouncil/2018/03/29/the-problems-with-bitcoin-and-the-future-of-blockchain. Accessed May 06, 2018.

[3] *Ethereum White Paper*. [Online] Available: https://github.com/ethereum/wiki/wiki/White-Paper. Accessed April 25, 2018.

[4] Tiana Laurence. *Blockchain For Dummies*. 1st ed. For Dummies: Computers. For Dummies, 2017. ISBN: 1119365597,9781119365594.

[5] *Hyperledger Composer Overview*. [Online] Available: https://www.hyperledger.org/wp-content/uploads/2017/05/Hyperledger-Composer-Overview.pdf. Accessed May 27, 2018.

[6] Norton Rose Fulbright and R3. *Can smart contracts be legally binding contracts?* [Online] Available: http://www.nortonrosefulbright.com/files/norton-rose-fulbright-r3-smart-contracts-white-paper-key-findings-nov-2016-144554.pdf. Accessed May 29, 2018.

[7] Thijs Maas. *Yes, this kid really just deleted 300 MILLION by messing around with Ethereums smart contracts.* [Online] Available: https://hackernoon.com/yes-this-kid-really-just-deleted-150-million-dollar-by-messing-around-with-ethereums-smart-2d6bb6750bb9. Accessed May 29, 2018.

General References

[8] *Full Stack Hello World Voting Ethereum Dapp TutorialPart 1*. URL: https://medium.com/@mvmurthy/full-stack-hello-world-voting-ethereum-dapp-tutorial-part-1-40d2d0d807c2.

[9] mjhm. *Hello World Dapp*. https://github.com/mjhm/hello_world_dapp. 2018.

[10] *Bitcoin mining*. [Online] Available: https://www.investopedia.com/terms/b/bitcoin-mining.asp. Accessed May 27, 2018.

[11] Daniele Magazzeni and McBurney, and William Nash. "Validation and verification of smart contracts: a research agenda". English. In: *COMPUTER* 50.9 (Sept. 2017), pp. 50–57. ISSN: 0018-9162. DOI: 10.1109/MC.2017.3571045.

A. Code Listings

Listing 2: Example of Solidity Code

```
pragma solidity 0.4.16;

contract TestContract {

  string private myString = "foo";

  function getString() constant returns (string) {
      return myString;
  }

  function setString (string _string) {
      myString = _string;
  }
}
```

Listing 3: Caesar Cipher for CSC 111

```
/*
 *  Author:    David Li
 *  UVicID:    V00818631
 *  Date:      Oct 69, 2014
 *  Lecture:   Assignment 7
 *  File name: V00818631A7P3.c
 *  Description: Reading, writing, and encoding files
 */
```

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <ctype.h>
#include <string.h>
#include <time.h>

#define INPUTFILENAME ("A7_2014_P3_TestingSherlock.txt")
#define OUTPUTFILENAME ("A7_2014_P3_SherlockEncoded.txt")
#define MAXSWAP (4)



void processFile(FILE* ifp, FILE* ofp, int seed) {
   char line[25];
   char word[25];
   printf("Begin process file\n");
    // your code goes here
   int rn1,rn2;
   int j;
   char tmp;

   while(!feof(ifp)) {

      fscanf(ifp,"%s",line);
      strcpy(word,line);

      if(strlen(line) > 3) {
         for(j=0; j<MAXSWAP; j++) {
            rn1 = rand() % (strlen(line) -2);
            rn2 = rand() % (strlen(line) -2);
            tmp = line[rn1 + 1];
            line[rn1 + 1] = line[rn2 + 1];
            line[rn2 + 1] = tmp;
         }
         if(strcmp(word,line) == 0) {
            tmp = line[1];
            line[1] = line[2];
            line[2] = tmp;
         }
      }

      if(!feof(ifp))
         fprintf(ofp,"%s ",line);
```

```
   }
   printf("End process file\n");
} /* ProcessFile */


int main(void) {
   printf("Welcome to Sherlock Holmes\n\n");
   unsigned int seed = (unsigned int)time(NULL);
   srand(seed);
   FILE *ifp;
   FILE *ofp;
   ifp = fopen(INPUTFILENAME, "r");
   if (ifp == NULL) {
      printf("Cannot open input file %s\n", INPUTFILENAME);
      exit(EXIT_FAILURE);
   } /*if*/
   ofp = fopen(OUTPUTFILENAME, "w");
   if (ofp == NULL) {
      printf("Cannot create output file %s\n", OUTPUTFILENAME);
      exit(EXIT_FAILURE);
   } /*if*/
   processFile(ifp, ofp,seed);
   fclose(ofp);
   fclose(ifp);
   printf("\nWe encoded Sherlock Holmes\n");
   return EXIT_SUCCESS;
} /*main*/
```