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, 4

Ethereum an open software platform based on blockchain technology that enables developers to build and deploy decentralized applications 1, 5

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

side chains are emerging mechanisms enable secure transfer of digital assets and tokens between blockchains. Attached to the parent blockchain using a two-way peg, sidechains are a separate blockchain and enables interchangeability of assets at a predetermined rate [1]. 4

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. Background

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 [2] 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 [3]. 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) [4].

Decentralized Applications

Blockchain technology is revolutionizing the internet by establishing trust in shared data. [5]. Additionally, transactions recorded on the blockchain are practically impossible to remove or change.

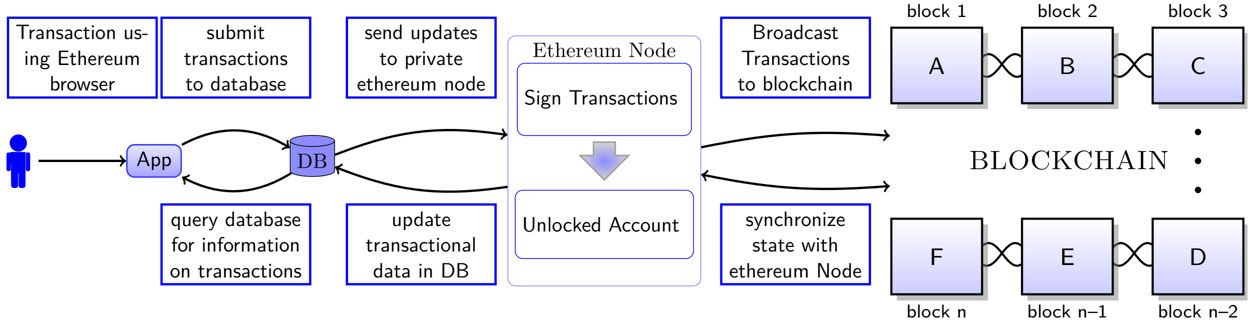

Transaction using Ethereum browser
submit transactions to database
send updates to private ethereum node
Ethereum Node
Sign Transactions
Unlocked Account
Broadcast Transactions to blockchain
block 1
block 2
block 3
A
B
C
App
DB
BLOCKCHAIN
query database for information on transactions
update transactional data in DB
synchronize state with ethereum Node
F
E
D
block n
block n–1
block n–2

Figure 1: An example of server-blockchain architecture in a DAPP.

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.

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

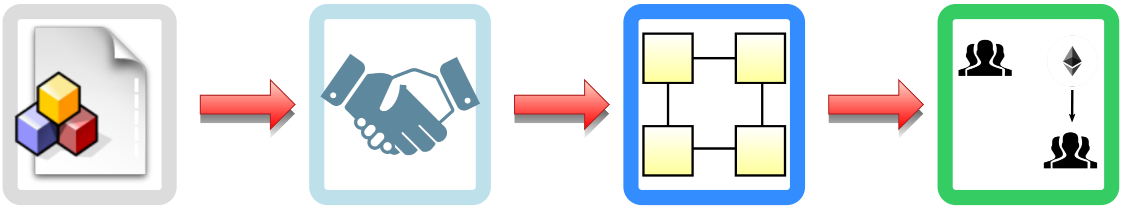
Public and Private Keys **In a blockchain** system, any key holder can use their private key to sign a piece of data. This results in a signature. In a Dapp, this can be used for:

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.

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 [4]. 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 [7].

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 [8]. Scrutinizing smart contracts and reducing bugs in production code is essential. An example of smart contract is available in Appendix A 1.



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 2: Illustrating how a smart contract works

1.2. Objective

The prominence of cryptocurrency and decentralized applications suggests usage of smart contracts will experience explosive growth.

Problem

Currently commonplace transactions require days to process and for parties verify correctness. For example to purchase houses, a plethora of steps are required, one must interactive with lawyers, real-estate agents, home inspector, buy insurance and shop for a mortgage.

Purpose

Leveraging existing blockchain technologies can automatic the majority of steps and cut out the middlemen, resulting in buyers conversing directing with sellers.

1.3. Aims

The aims of this project are to develop a decentralized blockchain system that:

1. Reduce cost of transactions by at least 50% from removing middlemen.
2. Improve transparency in software systems through augmented accessibility and understandability.
3. Increased security and greater enforceability of contractual obligations.

Limitations

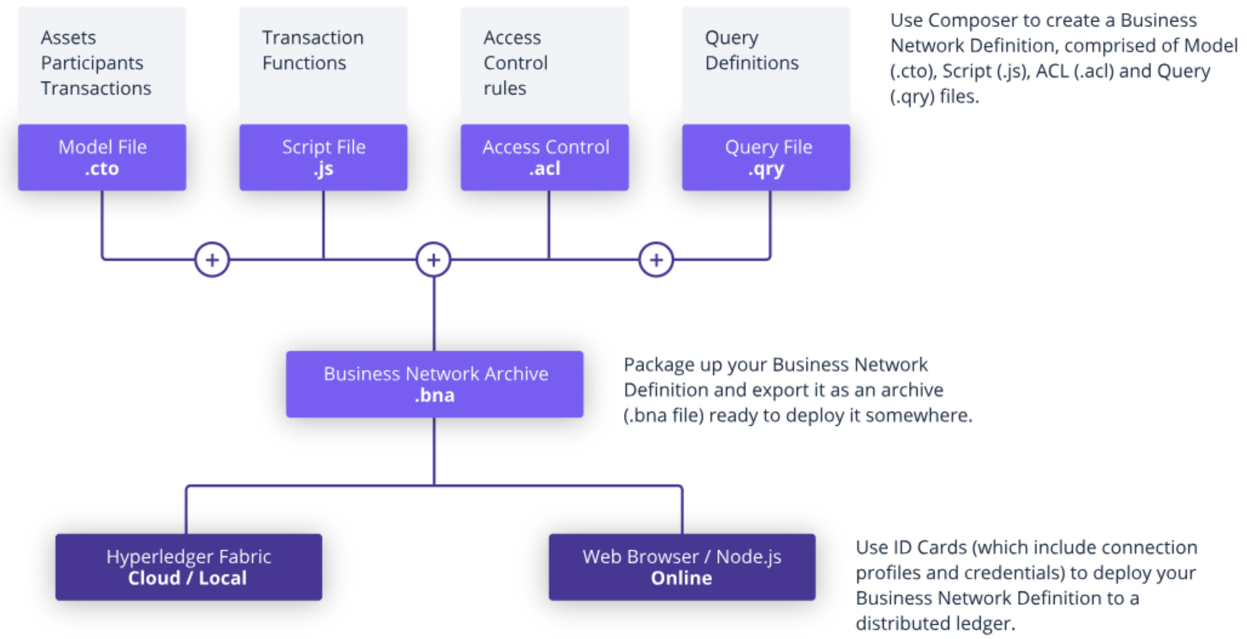
The regulatory uncertainty and impact of future regulations on blockchain technologies such as smart contracts will not be investigated. In addition, criminal usage of cryptocurrencies to avoid taxation and legal repercussions are beyond the scope of this report. In addition the impacts of quantum computing altering the validity of modern cryptography algorithms will not be investigated.

2. Discussion

2.1. Potential Solutions

— **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. They are open for anyone to participate at any level and have open-source code that their community maintains.

— **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 [6]. One use case is an accounting system that calculates payment, while hiding that information from unrelated organizations.



Use Composer to create a Business Network Definition, comprised of Model (.cto), Script (.js), ACL (.acl) and Query (.qry) files.

Package up your Business Network Definition and export it as an archive (.bna file) ready to deploy it somewhere.

Use ID Cards (which include connection profiles and credentials) to deploy your Business Network Definition to a distributed ledger.

Figure 3: Architecture of Hyperledger composer

— **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. Large organizations such as governments will likely use these extensively.

2.2. Initial Assessment

Determining which platform is best for smart contracts should be done using a weighted decision matrix, based on the particular application. For internal processes such as supply chains, a private blockchain makes sense (data cannot be changed) and cryptographic auditing with known identities (public keys). For a trustless system that verifies every transaction, using a public blockchain is essential. In comparison, role-based access control is feasible by using a permissioned blockchain.

Despite the slow speed of the public blockchain, innovations such as side chains enable quick transactions and are used in decentralized game development [9]. A permissioned blockchain allows role based access control which is essential in business applications. One example is to prevent unrelated parties from viewing other's data. 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.

Table 2: Sample Decision Matrix for designing a blockchain system

	Existing Systems	BlockChain Systems		
Criteria	Centralized	Public	Permissioned	Private
speed and latency	5	7	7	6
scalability	5	9 8	7	4
security and immutablity	3	7	8.5	9
storage capacity	4	9	9	6
transparency	3	9	7	5
Total	21	41.6	38.5	30

2.3. Engineering Analysis

Security of Smart Contracts

Blockchain transactions are secure because that are immutable and decentralized. However, exploiting bugs in smart contracts are financially devastating [8] as fraudulent transactions cannot be reverted. Disconnects between software developers and security experts has resulted in 3 out of 4 "applications produced by software vendors fail to meet OWASP Top 10 standards" [12]. Although blockchain technologies increase underlying security and reliability, exploiting poorly coded and insecure smart contracts remains a major risk, and releasing open-source code allows hackers exploit flaws in the codebase before corrective processes are applied.

Brute force cracking of private keys

A ethereum key, which is randomly selected 256 digits [4], is very difficult to hack. A simple calculation illustrates the impracticalities of brute forcing for a 256 bit key. Assuming that a 1 exaflop (10^{18} calculations per second) 15 megawatts supercomputer [10] is used, electricity costs are 0.1326 per kWH [11].

$$\text{Possible number of private keys:} = 2^{256} = 1.1569 \times 10^{77}\text{decryptions} \tag{1}$$

$$10^{18}\frac{\text{decryptions}}{\text{second}} \times \frac{3.154 \times 10^{7} second}{1 year} = 3.154 \times 10^{25}\frac{\text{decryptions}}{year} \tag{2}$$

$$\text{Time to decrypt a 256 bit key} = \frac{1.1569 \times 10^{77}}{3.154 \times 10^{25}}\text{years} = 3.66804 \times 10^{51} years \tag{3}$$

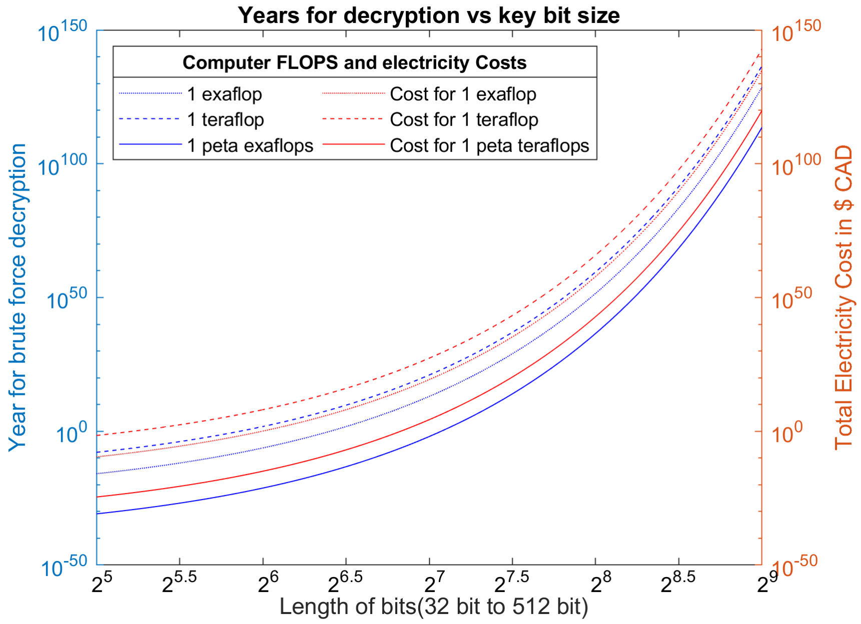


Figure 4: Difficulty of brute forcing a 256-bit key for different computers

Scalability is the biggest issue facing blockchain technology, because storage is expensive, and public blockchains are trustless, every transaction is verified [13].

Enforcability of Smart Contracts

If contractual fulfillment are delegated to code, it becomes paramount to ensure that such code contains no errors. Oftentimes smart contracts run on top of a blockchain, so the code of the smart contract would be neither incorruptible nor secure. When self-enforcement guarantees performance and subsequent human intervention is disallowed, modification of a smart contract is impossible, logically, its code must be perfect.

Lastly, it must be acknowledged that self-enforcement may be limited to transactions where off-chain events are computationally verifiable. Just like not all contractual obligations can be represented in code, not all off-chain events can be captured as computer-readable data or measured with objective criteria.

Performance of Smart Contracts

Statistically, every computer program coding error ("bugs"). It is practically impossible to ensure perfect performance of a smart contract. Immutable, self-executing smart contracts may protect the transaction from unexpected human interactions but introduce the risk of performance being affected by coding errors. Given that the smart contract may execute incorrectly due to a coding error, the practical difficulty of writing bug-free code, alleviating and reducing risk from vulnerabilities is essential.

Technical writings suggest that such direct coding of smart contracts would force lawyers to be more precise and structured in describing the rights and obligations of the parties [14]. Since most lawyers are unlikely to become programmers and vice-verse, converting existing legal documents into smart contracts is undesirable. In order to create smart contracts, obligations must be reported in a detailed manner and provide objective criteria for execution.

Transparency and Speed of Blockchain Transactions

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, [2] and writing to the blockchain is relatively expensive compared to traditional systems. Smart contracts are useful because they cannot be changed by the parties involved in the transactions, yet, in some cases such as criminal activity, the ability to reverse transactions or freeze accounts is extremely desirable. For example EOS, a centralized blockchain platform, was criticized for freezing accounts without due process and community backing. [15].

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

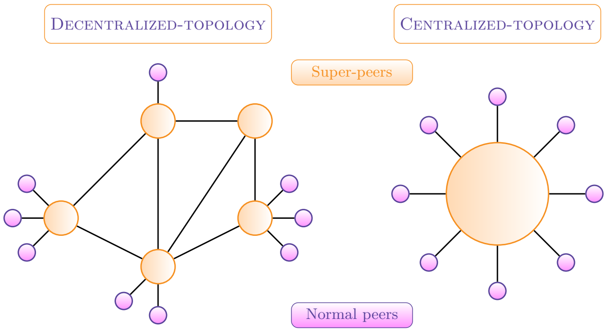


Figure 5: Decentralized structure vs Centralized Structure

3. Conclusion

Decentralized systems are inherently more reliable, resilient against brute force attacks, and are more transparent. Fulfillment of smart contracts is limited due to off-chain information, logistical challenging obtaining impartial or truthful inputs from stakeholders, and verification of transaction completion involving real-world assets. This suggests effectiveness of smart contracts is currently limited to digital assets rather than for physical assets. In addition, translating legal-binding contracts to code is challenging because the majority of programmers lack a legal background.

Despite shortcomings that involve transactional steps that are computational impossible to verify (package is delivered), performance of smart contracts is superior as transactions are tamper-proof. Additionally, reduction of ambiguity from usage of smart contracts is highly probable because only one interpretation is possible. Coding errors or bugs exist in every non-trivial application, therefore transactions through smart contracts intrinsically carry some risk. This implies protocols and prior real-world agreements are necessary to migrate risk when executing smart contracts. Furthermore, increased precision and detail for transactional inputs and outputs required for creating smart contracts would benefit all parties.

As shown in Figure 4 only computers with significant high processing power (quantum computers) can decrypt a 256 bit ethereum key cost-effectively. This indicates trying to hacking private keys is not a worthwhile venture. Overall, smart contracts in conjunction with blockchain technologies allow for users to have a unique digital presence, securely transfer ownership of assets and avoid key shortcomings of centralized IT systems.

4. Recommendations

5. References

Cited References

[1] Adam Back Matt Corallo. *Enabling Blockchain Innovations with Pegged Sidechains*. [Online] Available: https://blockstream.com/sidechains.pdf. Accessed May 31, 2018.

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

[3] 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.

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

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

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

[7] 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.

[8] 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.

[9] Adam Back Matt Corallo. *Loom Network sdk for developers*. [Online] Available: https://medium.com/loom-network/loom-sdk-for-developers-using-an-indexing-layer-for-lightning-fast-dapp-performance-b17f8ba25a3c. Accessed May 31, 2018.

[10] Robert F. Service. “Design for U.S. exascale computer takes shape”. In: *Science* 359.6376 (2018), pp. 617–618. ISSN: 0036-8075. DOI: 10.1126/science.359.6376.617. eprint: http://science.sciencemag.org/content/359/6376/617.full.pdf. URL: http://science.sciencemag.org/content/359/6376/617.

[11] *Residential Rates*. [Online] Available: https://app.bchydro.com/accounts-billing/rates-energy-use/electricity-rates/residential-rates.html. Accessed June 06, 2018.

[12] *How Do Vulnerabilities Get Into Software?* Tech. rep.

[13] Vitalik Buterin. *Ethereum scalability research and development subsidy programs.* [Online] Available: https://blog.ethereum.org/2018/01/02/ethereum-scalability-research-development-subsidy-programs/. Accessed June 28, 2018.

[14] Stephen Wolfram. *Computational Law, Symbolic Discourse and the AI Constitution.* [Online] Available: http://blog.stephenwolfram.com/2016/10/computational-law-symbolic-discourse-and-the-ai-constitution/. Accessed July 07, 2018.

[15] *EOS Emergency Actions.* [Online] Available: https://bitcoinexchangeguide.com/eos-undergoes-emergency-actions-after-block-producers-freeze-accounts-without-community-input/. Accessed June 26, 2018.

General References

[16] *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.

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

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

[19] 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.

[20] *What are sidechains.* [Online] Available: https://hackernoon.com/what-are-sidechains-1c45ea2daf3. Accessed May 31, 2018.

[21] P. Smith et al. "Network resilience: a systematic approach". In: *IEEE Communications Magazine* 49.7 (2011), pp. 88–97. ISSN: 0163-6804. DOI: 10.1109/MCOM.2011.5936160.

A. Code Listings

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 {

     }
```