* COIS 3860H : Internship Course

Blockchain

Vitalik Buterin

Ethereum
- Cryptocurrency : Ether (Eth)
  - > no need to give personal details
  - > no financial services needed
  - > peer to peer network
  - > no govt has control - decentralization

concept of shared memory

analogy - court system (resolving disputes)

1. Smart Contracts
2. EVM
3. Ethereum Blockchain
4. Ether
5. Proof of work.

* DeFi → Decentralised Finance
  DAOs → Decentralized Autonomous organizat
  DEX → Decentralized Exchanges. (no centralization)

Uniswap
- 100% on chain market maker — sets price, offers to trade

PoolTogether — no loss lottery system

Similar to ~~ablidit~~ JS.

Solidity : you writing smart contracts.
Python : blockchain

JSON format to represent blocks.

```
{
  "number" : 1234567,        nonce?
  "hash" : "0xabc123...",
  "parentHash" : "0xdef456...",   ← hash of previous block
  "miner" : "0xab1c23...",
  ...,
  "transactions" : [...]
}
```

smallest

ether → wei

$\frac{1}{10^{18}}$ → 1

ether → 10^9 gwei

```
In [2]: Web3.toWei(1, 'ether')
Out[2]: 1000000000000000000
```

```
w3.eth.accounts[]
w3.eth.getBalance(w3.eth.accounts[])
w3.eth.getBlock('latest')
```

```
In [10]: tx_hash = w3.eth.sendTransaction({
    'from': w3.eth.accounts[0],
    'to': w3.eth.accounts[1],
    'value': w3.toWei(3, 'ether')
})
```

3)

```
w3.eth.getTransaction(tx_hash)
```

Blockchain: digital accounting book
↳ to record transaction.
↳ an asset moves from one person to another.
↳ digital representation.

cryptographic algorithm (1) hash

(2) decentralization → security

can be tangible / intangible

mining: process of adding transactions to the large distributed public ledger of existing transactions known as blockchain

Key elements of blockchain

1. Distributed ledger technology
2. Immutable records
3. Smart contracts.

* tamper-resistant
* trust.
* no duplicates/copy.
* transparency

• a new block is added if there are any changes made.

blockchain types

① Public
② Private
③ Permissioned
④ consortium

semi-private.

can be in tightly-controlled & preferable for compliance & regulatory measures.

SHII ways for cyberattacks & fraud
① Code exploitation
② Stolen keys
③ Employee computer hacked.

how fraudsters attack blockchain technology?

① phishing attack
② Routing attacks
③ Sybil attacks
④ 51% — (most difficult?)

PRIVATE BLOCKCHAIN

* secure, resilient infrastructure
* business & governance risks
  - financial implications.
  - reputational factors
  - compliance risks

Hyperledger Fabric

* Permissioned network - decentralized trust.
* Confidential transactions
* Pluggable architecture
* Easy to get started

B2B transactions
↓
to maintain privacies of transactions from different businesses.

	Ethereum	Hyperledger
Purpose	• B2C businesses & generalized applications	• B2B businesses
confidentiality	transparent even the network	confidential transactions
mode of peer participation	public / private & permissionless network	private & permissioned network
consensus mechanism	PoW algorithm: consensus is reached by mining — proof-of-work (miners)	pluggable consensus Algorithm: no mining required
programming language	smart contracts written in Solidity	chaincode written in Golang ⇒ JS, node.JS
cryptocurrency	Ether	no built-in cryptocurrency

Blocks

1. data in the block
2. 32-bit whole number called nonce — randomly generated when block is created, which then generates a block header hash
3. hash — 256 bit number wedded to the nonce. Must start with a huge number of zeroes. (i.e. be extremely small)

NONCE: number only used once

unique

calculated from

the number miners are solving for → (25 Jan meeting) find the nonce that will get the required hash

→ http and unknown website

→ continue doing what we decide for 2 weeks

nonce → generates accepted hash

Node.js

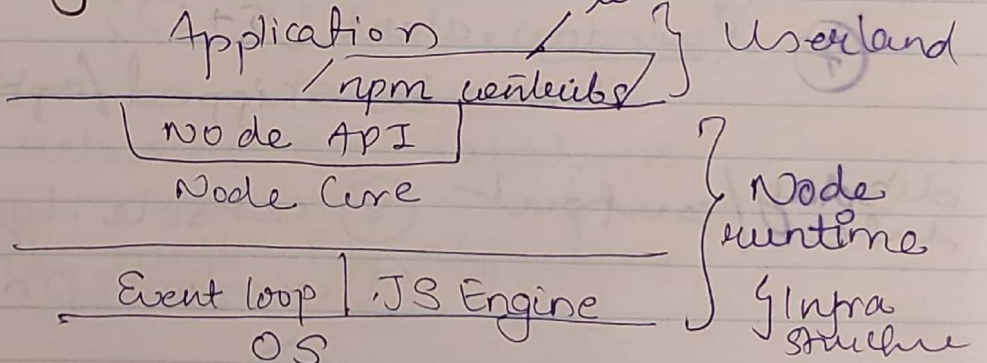
JS → popular programming language & it runs in any web browser with a good web browser.

an interpreter & environment for JS with some specific useful libraries which JS programming can be used seperately.

→ developing server-side & networking applications

EVENT LOOP

1. Timers phase: setInterval() setTimeout()
2. Poll phase
3. check phase: setImmediate()



javascript – program the behaviour of web pages.

(HTML) methods :- ① getElementById() → "demo"

② .innerHTML = 'aaaa' ; → to change the text/content

③ onclick

④ .src = ' ' → import external file

change in CSS ⑤ • style.fontSize

• display = "none"; — hiding HTML elements + displaying in a style like block.

js code is inserted in <script> </script> tags

— using external script src = "myscript.js"

creating a function function myFunction() { }

calling a function ⑥ onclick = "myFunction()"

use several script tags to add several script files to one page

⑦ window.alert(5+6); (can be skipped / optional)

display/output ⑧ console.log()

⑨ window.print()

declaring variable ⑩ var x, y, z;

keywords
- break
- continue
- debugger
- do .. while
- for
- function
- if else
- return
  ↳ exits a function
- switch
- try ... catch
- var

// single line comment
/* */ multi line comment

jump out of loop and start at top

terminates a switch, loop.

JS ⇒ case-sensitive
⇒ no hyphens
⇒ camel case firstCase
⇒ pascal case ~~firstCase~~ FirstName
⇒ Underscore
⇒ names can begin $ & _ also

* if no value is assigned to a variable then the value is undefined.

Bitwise Operators

(1×0)	&	AND
(1+0)	|	OR
	~	NOT
	^	XOR → return 1 when the numbers are different
	<<	zero fill left shift
	>>	Signed right shift
	>>>	zero fill right shift

0 ^ 1 = 1
0 ^ 0 = 0
1 ^ 1 = 0

(11) typeof returns the datatype.

* datatype of null is object.

common HTML events

onchange ——— HTML element has been changed
onclick — user clicks an HTML element
onmouseover — hover over HTML element
on mouseout — stops hover...?
onkeydown — keyboard key is pressed
onload — page has loaded.

Node.js

require(' '); → to load modules

http.createServer(— method turns my computer into a HTTP server

function(req, res){

req → contains info about the HTTP request that raised event

res → use res to send back the desired HTTP response

res.writeHead(200)
- sends a response header to the request
- status code (3-digit HTTP status code) 404

create HTTP server

```
var http = require('http');
http.createServer(function(req, res){
  res.writeHead(200, {'Content-Type':
                      'text/plain'});
  res.end('Hello world');
}).listen(8080);
```

message

8080 → HTTP default listener

Website : innov-edu.ca

Ethereum ← Distributed ledger Tech
- is an application.
- runs Smart Contracts on the EVM for applications that are attributed to being decentralized & are for mass consumption.

transaction → fees → spike 9.331 – 10.50 (1 day)
→ transparent transactions

→ public / private without permission
⇓
anyone can spin up a node, other nodes only acknowledge its existence but not share any data.

→ consensus required
↳ Proof-of-Work algorithm

can end up with large no. of participants.

is private in public network possible??

incentives.

cost of execution - gas

lines of codes.

Hyperledger
- Sawtooth
- Iroha
- Indy
- Burrow

designed to foster
- support pluggable implementations of components delivering high degrees of confidentiality, resilience & scalability

resilience: capacity to recover quickly from difficulties.

scalability: capacity to be changed in scale/size.

* more confidentiality
* better trust

?? (*) closer network??
* permissioned private network.

users must be granted both network & application access just to connect.

* no consensus needed
No-Op agreement protocol (PBFT)

* doesn't require crypto for transactions.

Hence, no mining required

develop custom tokens? (for loyalty program)

* no sensitive info?
↓
courses?
copyright?

* the site public to increase the range / base of audience? → courses, loyalty program, token
↳ private ??

B2C.

Ethereum

1) most popular → Solidity — C++ / Python / JS
build smart contracts.
2) Remix — debugging & compiler tool — good for beginners
3) Truffle — DApps IDE.
4) Mist — communicates with Geth.
   - browser interface
   - wallet
   - Serves like a node.
   - Go used to mine Ether.
   - not user friendly hence use with mist.
5) metamask.
   - digital wallet
   - act as browser extension.
6) web3 - library
   - blockchain council (website)
   - can dynamically create abstractions that represent smart contract.

B2B

Hyperledger

1) Java
2) JS
3) Golang.
4) node.js

media.consensys.net
- allows various dApps available on Ethereum network.
- allows user to sync to the blockchain

truffleframework.com
code example

- contract compilation & deployment
- Automated contract code testing
- interactive console to work with built contracts

external script runner that works with contracts that are included

Loyalty Program

→ mentioning metamask somewhere? like how? what? why? maybe?

→ I ~~be~~ am connected to metamask but ~~#~~ on the account page it does not show my address but on Sign up it does.

Ethereum — Metamask wallet

builds the ethereum address.

Extensions

1. Debugger for chrome

Hyperledger

- Docker Engine & Docker Compose
- Nodejs & NPM
- Git
- Python 2.7.x
- JS ↪ SDK is written in JS
- Lisk's

* Golang : most of chaincode is Golang.
* Java SDK
* node.j's

to develop new blockchain applications

chaincode

chaincode interface

Invoke method → by clients to submit transaction proposals.

Init method.
↳ initialization function for chaincode.

→ Run a commercial paper smart contract with the IBM Blockchain VS code extension

node.js
npm
Docker
Docker compose

provides simple API to submit transactions to a ledger or query the contents of a ledger with minimal code.

chaincode

package manager for node.js

puts modules in place so node can find them & manages dependency conflicts intelligently

is currently deployed via a Docker containers.

built-in support for JS, coffeescript, SASS, ES6, JX

Ethereum Developer Tools

Truffle - comply & deploy & work with distributed DApps smart contracts, inject them into web apps, develop front-end for DApps.

IDE
↓
base environment

automated contract testing with Mocha & Chai

↳ integrates compilation, testing & deployment of smart contracts.

Ganache locally deployed Blockchain simulator

↳ features GUI that can simulate blockchain networks & live test smart contracts without requiring to set up real test networks or using to a remote network

like "personal blockchain" — safe & deterministic environment.

Drizzle: assortment of front-end libraries that offer useful components for developing web applications that can seamlessly connect with Smart Contracts

IDE

Remix: open-source tool — helps write Solidity Contracts straight from the browser.

written in JS

supports testing, debugging & deploying of Smart Contracts

IDE to write, test, debug, deploy, analyze Smart Contracts ??? ... } Just for Smart contracts.

Solidity — writing and implementing smart contracts

Test RPC — replaced Ganache

Hyperledger base language { HLF Docker container platform

Go

Golang → VS code.

[C]

like Truffle ? ⇐ Sawtooth Seth

Hyperledger Sawtooth

running EVM smart contracts to Hyperledger Sawtooth Platform

Javascript → writing chaincode.

Java — JAVA SDKs

Node.js — provides a simple API to submit transactions to a ledger or query the contents of a ledger with minimal code.

Docker — helps to pack up an application by using containers with all parts (libraries, other dependencies) & deploy it as one package.

Getting Hyperledger Fabric network details

* IBM blockchain platform : provides the network configuration details as connection profile (in JSON)

* network setup using Docker compose
  - Get appropriate certificates
  - from the crypto config directory along with other network details like IP and port which are defined in the configuration (yaml) file

* network setup using kubernetes
  - Copy the certificates (crypto-config) from ~~other~~ kubernetes pod to your local system, get the public IP of the kubernetes cluster & other network config details from config (yaml) files.

- secure hashing
- encryption/ decryption
- digital signing

for { Crypto Suite / Abstract class

1. createKeyFromRaw(pem, opts)
2. decrypt(key, cipherText, opts)
3. deriveKey(key, opts)
4. ~~encrypt~~ encrypt(key, plainText, opts)
5. generateEphemeralKey()

② use crypto suite default

Enroll the admin:

org.hyperledger.fabric-ca.sdk.HFCAClient class instance

① → CA, URL, name, pem certificate

③ getUserContext();

④ API

(name, affiliation, setMspId)

Register & Enroll new users

↳ UserContext(), ← Enrollment

Invoke the chaincode.

1. Read user context that was saved
2. create peer, eventhub & orderer references
   - read pem certificates.
3. initialize channel — Channel channel = hfClient.newChannel(channel-name)

   addPeer(peer)
   addEventHub(eventHub)
   addOrderer(orderer)
   initialize()

4. Transaction proposal.
5. Send transaction to the orderer.

Smart Contract language
- JS
- Java
- Go
- TypeScript

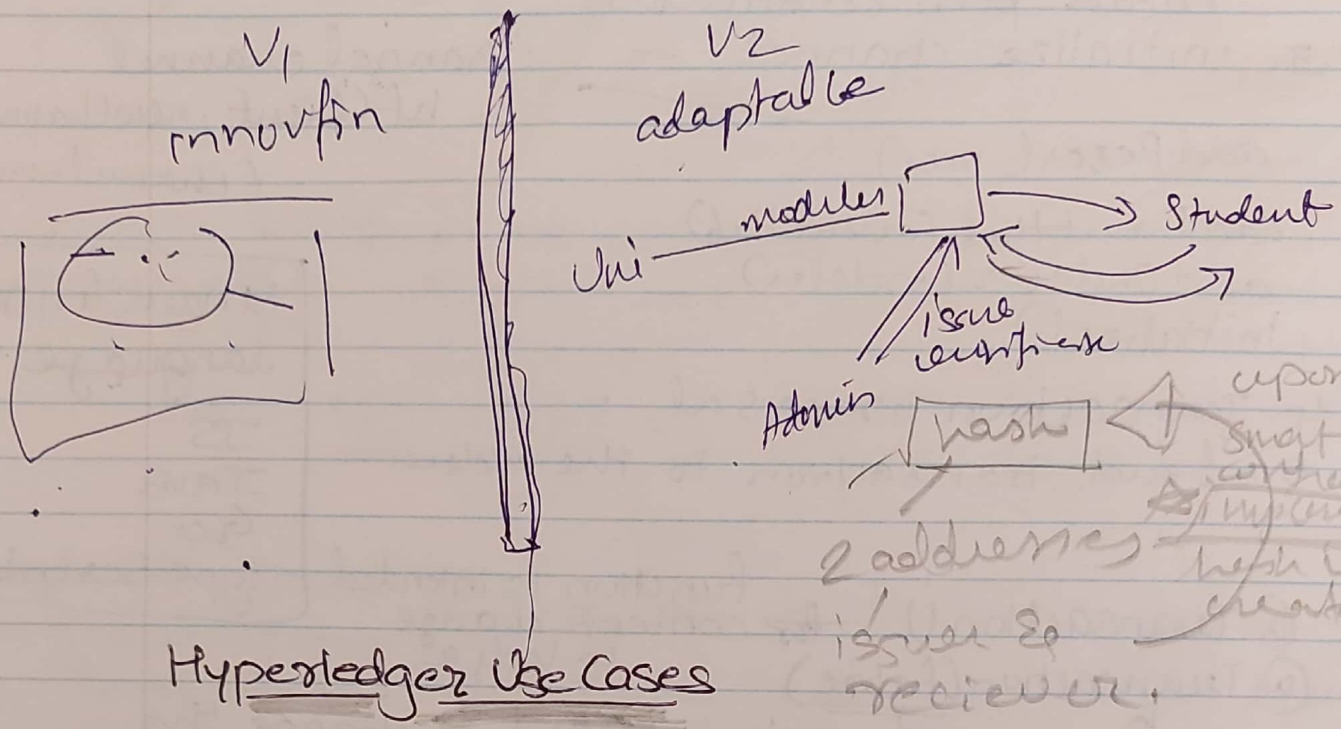
@Transaction() — function is intended for content change in ledger
@Transaction(false) — not intended to change the contents of the ledger.

→ University name change
→ full on editable document

☆ Update Again

V1 innovation

V2 adaptable



Hyperledger Use Cases

1☆ A defined use case with projected benefits.

2☆ Identified participants

└ when new participants wants to join their new instance is created.

☆ Students
☆ Universities
☆ Administrators

Assets
☆ loyalty points
☆ courses / videos
☆ address of participants

transaction
☆ earn loyalty points
☆ earn courses/videos
☆ issue certificate
☆ issue certificate
☆ recommend?

upload?

Certification System

front-end ↕ client-side

design HTML, LSS.

logic

JS libraries like Angular, React, Vue, Stencil.

back-end ↕ server-side

old day → database, Logic processing

Instagram?? Youtube??

APIs

only for raw web application

OS

webservers

software stacks

Size: blockchain, ~~biggest~~ upscale in future

Flexible stack

Type of product
- Education place
- Blockchain
- Iot

- .NET
- MEAN
- LAMP

- C#
- JS
- Java
- Go
- Python
- PHP

open source developer platform

Free open source JS stack

~~cost~~

NoSQL database system

binary JSON format

MongoDB

ExpressJS

AngularJS

Node.js

- because all components of MEAN stack support programs that are written in JS.

Server side

LINUX
APACHE
MYSQL
PHP

popular stack

light framework used to build web applications in Node.

JS framework

npm or VSCode

Extensions

metamask

server side JS execution environment.

Helps in building highly scalable & concurrent applications ready

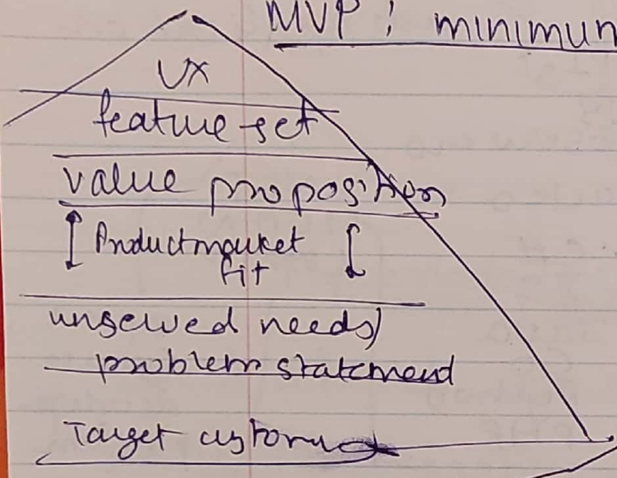
make App Development easy since runs in single process without creating a new thread for every request

+ make website secure.
~~- namechimp~~
- Innov-cdu. /certificate & point towards
ca.
& not ~~ttl~~loyalty
~~- check where the site is hosted~~ ; namecheap-

Design a plan to bring the demo to MVP state. — architecture.

☆ Delightful
☆ Usable
☆ Useful.

MVP : minimum viable product

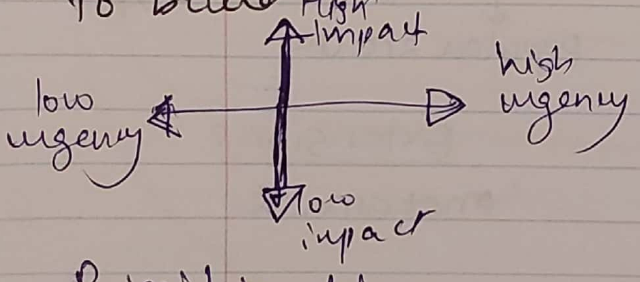


- outline needs
- motivation
- pain points

① Understand Business needs
→ Long term Goal
→ User Personas.
→ Success Criteria
↓
more than single metric.

② Find the opportunities.
→ map out user journeys [BVG]
→ pain & gain
→ opportunity statements.

③ decide what features to build



Prioritization matrix

Blockchain and CryptoCurrency Online Course & Internship.

NFTs

flo — better for NFT rather on Ethereum.

Research on it.

OpenSea — create your own NFTs

Mantic — sidechain

need to be on correct network — create NFTs for free on opensea. — Practice Questions } 2 hrs cours

network effects

Intro to Exchanges & NFTs

company ① blockchain — whitepaper.

token economy — how you distribute your token with community.

② Efforts

Quiz

check all options that are right.