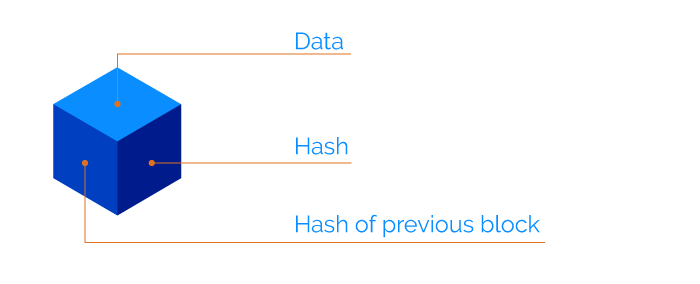
Blockchain je lanac blokova koji sadrže informacije.

Blockchain je distribuirana knjiga koja je dostupna svima.



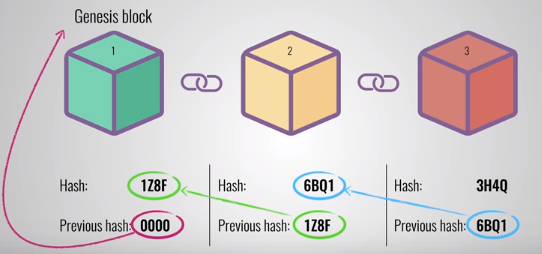
Slika 1. od cega se sastoji block.

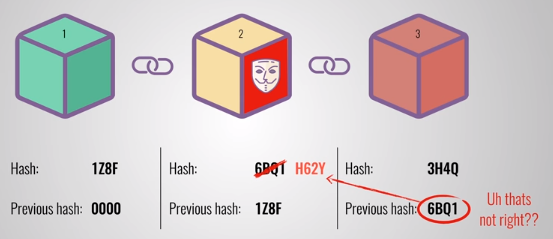
Vrsta podataka (data) koji se spremaju unutar bloka ovise o vrsti blocchaina. Npr BitCoin sadrži:

* From
* To
* Amount

Hash se može zamisliti kao otisak prsta. Hash jednoznačno označava blok. Dakle hash je jedinstven. Generira se na osnovu podataka pohranjenih u bloku i ukoliko se promjeni bilo koji podatak unutar bloka, automaski se mijenja i hash bloka.

Svaki blok sadrži hash prethodnog bloka.

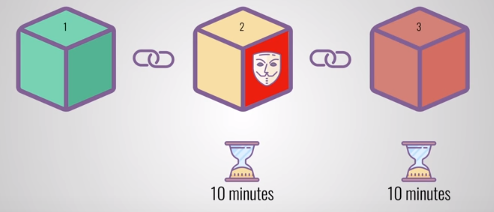




Ukoliko npr u drugom bloku promjenimo neki od podataka, njegov hash se također mijenja što rezultira da 3. blok više ne sadrži ispravan hash 2. bloka te se cijeli lanac kida.

Ovo se može riješiti tako da se ponovno odredi hash svakog bloka počevši od onog mijenjanog pa nadalje. Moderna računala su veoma snažna i mogu proračunati oko 100 000 hashova po sekundi.

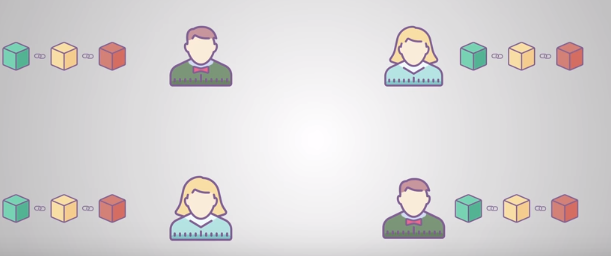
Stoga da bi se onemogućilo to blockchainovi imaju nešto što se zove „proof of work“. To je mehanizam koji usporava nastanak novih blokova. U bitCoina treba otprilike 10 min za stvorit novi blok.



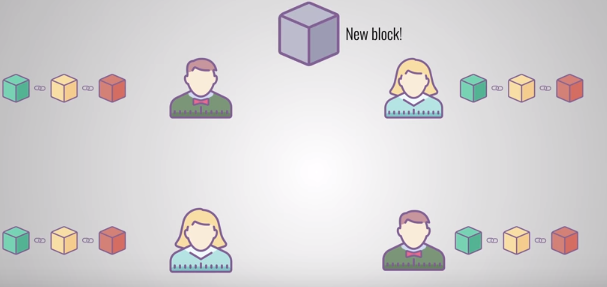
Dakle ukoliko želimo promjeniti blok i sve sljedeće trebat će nam oko 10 minuta za svaki blok.

Drugi način zaštite je distribuiranost.

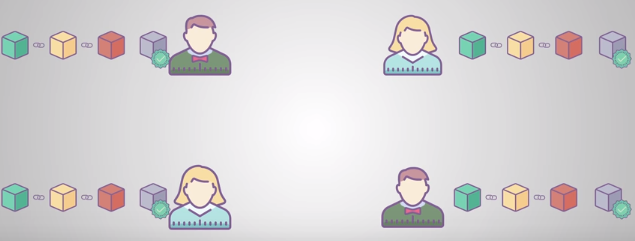
Blockchainovi koriste P2P network i svakom je moguće sudjelovati. Svaki sudionik ima svoju kopiju blockchaina. Sudionici koriste to da provjere da je sve uredu.



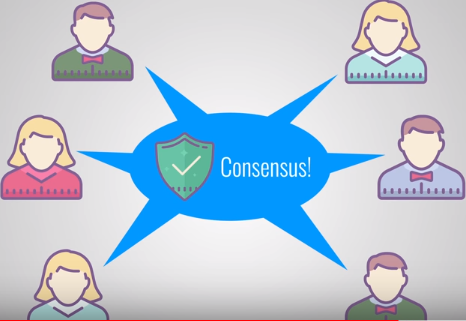
Pogledajmo sada što se događa kada neko stvori novi blok.



Blok je tada poslan svima na mreži.



Svaki čvor tada provjeri ispravnost bloka. Ukoliko je blok ispravan, svaki čvor doda blok na kraju svog blockchaina.



Svi čvorovi stvore konsenzus, tj. Složnost oko toga koji je blok ispravan a koji ne.

Blockchainovi stavno napeduju. Jedna od inovativnosti je „pametni ugovor“ (smart contracts). To su programi pohranjeni untar blockchaina koji omogućavaju npr. Automatsku dodjelu novaca ovisno o određenim uvjetima.

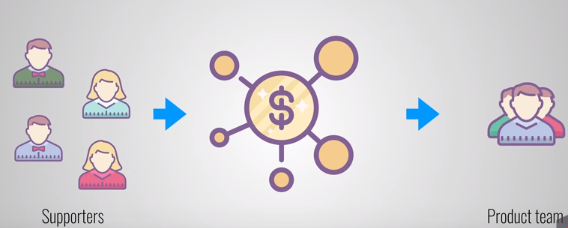
SMART CONTRACTS

Pametni ugovori su kao i obični ugovori u stvarnom svijetu. Razlika je u tome što su pametni ugovori potpuno „digitalni“. To su mali računalni programi koji su pohranjeni unutar blockchaina.

Uzmimo za primjer kickstarter, jedna od najpopularnijih platformi za prikupljanje sredstava.

Korisnici mogu otići na kickstarter platformu, pokrenuti projekt, postaviti koliko novca im treba za projekt i početi prikupljati novac od ljudi koji vjeruju u taj projekt.

Kickstarter je u biti „treća strana“ koja se nalazi između projekt tima i osoba koje podupiru i financiraju projekt.



To znači da obe strane moraju vjerovati kickstarteru da će raspodjeliti sredstva pošteno i kako je dogovoreno. Sponzori projkta će u počeku financirati projekt u koji vjeruju te ako projekt uspije će dobiti dio novca od zarade po dogovorenim principima.

Ovakvu platformu možemo napraviti uz pomoć smart contracta. Koritenjem smart contracta nije nam potrebna „treća strana“ kao što je kickstarter.



Dakle, sponzori projekta prebace novac na smart contract.



Ukoliko projekt uspije smart contract automatski šalje novac projekt timu.

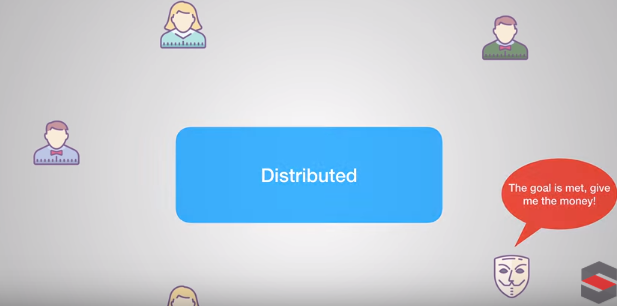


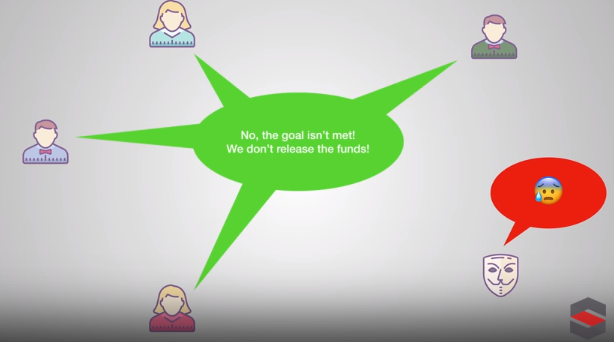
A ukoliko projekt ne uspije, novac se automatski vraća nazad sponzorima.

Budući da je smart contract pohranjen unutar blokcchaina sve je distribuirano što znači da ni jedna strana ne može kontrolirati novac.

Ali čekaj malo, zašto da vjerujemo smart contractu?

Zbog toga što su smart contracti pohranjeni u blockchainu. To znači da kada se jednom napiše više ga nitko ne može promjeniti. Ukoliko neko promjeni smart contract, tada se mijena i hash bloka i puknut će lanac te će ostali sudionici reći da to nije ispravno te će se taj neispravni blockchain odbaciti.





Odavde proizlazi moguća velika primjena smartcontracta. ( Banke, razni ugovori...).

Najveći blockchain koji podržava smart contracte je Ethereum. Smart contracti za ethereum se programiraju u programskom jeziku Solidity.

BlockCHAIN DEMO – paralelno demonstratirati sa gornjim dijelom.

<https://anders.com/blockchain/>

1. Hash  
   Hash izgleda kao neka random gomila brojeva i slova. To je u stvari fingerprint koji za uneseni Data daje jedinstveni hash. Ne postoji isti unos koji daje isti hash. Za isti unos dobije se isti hash.   
   Neovisno koliko je dug uneseni data uvijek ćemo dobiti hash iste duljine.
2. Blok  
   Hash počinje određenim brojem nula (ranije spomenuti proofofwork). Treba određeno vrijeme da se nađe odgovarajući hash. To se postiže tako da se redom povećava Nonce (broj pokušaja) dok se ne dobije valjani hash.

<https://anders.com/blockchain/public-private-keys/>

Public/Private keys

Što sprječava nekog da unese nečiji public key i samo sebi šalje novac?

Stoga postoji private key. Za uneseni private key se generira public key. Private key mora ostati skriven i nikome to ne smijete reći. Public key s druge strane kažete svima.

Na osnovu signature-a ( potpisa) može se provjeriti je li netko stvarno generirao te podatke s tim i tim public keyom. Na osnovu potpisa moguće je provjeriti je li ta osoba kojoj pripada taj public key stvarno vlasnik odgovarajućeg private key-a.

Sada umjesto običnog teksta, amo napraviti transakciju.

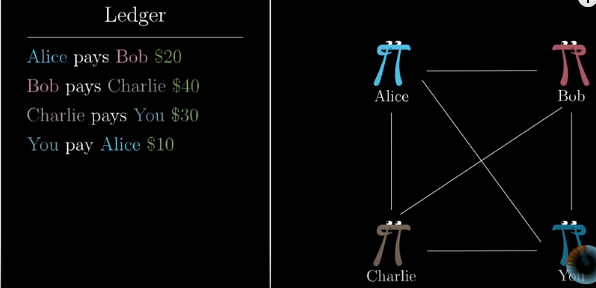
Ako netko pokuša promjeniti npr poslani iznos u transakciji s recimo 2 dolara na 20 i stisne mine (da nađe odgovarajući hash), signature će i dalje biti crven. Razlog je što ta osoba nema pristup private key-u od pošiljatelja i ne može pogoditi pravi potpis.

How bitcoin actually work?

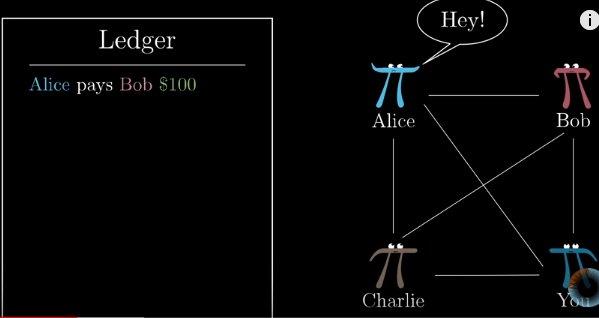
Zamislimo da imamo neku javnu knjigu u kojoj pišemo sve transakcije.

Nek imamo protokol:

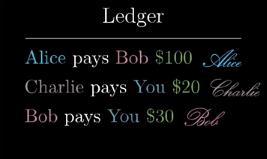
* Svako može dodati transakciju u javnu knjigu
* Nitko ne smije uplatiti nekome više nego što posjeduje



Što sprječava nekoga da samo unosi transakcije koje njemu donose zaradu? Npr Bob stavi da mu Alice treba platit 100 dolara, bez da Alice to odobri.



Problem se rješi uz malo kriptografije tj. Digitalnog potpsa.



Ideja se sastoji od toga da postoji nešto što Alice i drugi mogu dodati pored transakcije što označava kako su oni vidjeli i odobrili transakciju.

Vjerojatno se sada javlja pitanje: A što nas sprječava da samo kopiramo nečiji signature? Ključ je u tome što se signature generira na osnovu private key-a i ostalog sadržaja bloka ( data). Dakle samo osoba s private key-om može generirati ispravan signature (primjer gore public/private key).

Dakle svaka osoba ima par private/public key. Private key držimo za sebe, a public key kažemo svima.

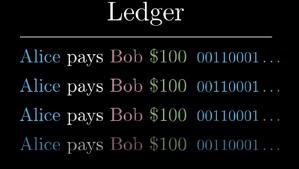
Dakle generiranje potpisa možemo zamisliti kao funkciju koja kao parametre prima Message ( sadržaj poruke) te private key. Funkcija vraća jedinstveni signature. Message kao parametar znači da niko ne može uzeti vaš signature i kopirati ga na neku drugu poruku. Npr. Pošaljete nekom 20 dolara i taj neko ide promjeniti da umjesto 20 piše 200. NE IDE! Razlog je što je promjenio sadržaj poruke što znači da nema ispravan signature (potpis) i blok neće bit valjan.

Postoji funkcija koja provjerava ispravnost signature-a (potpisa) koja za parametre uzima Message, Signature i public key. Verify funkcija vraća true ili false.

Signature se sastoji od 256 bitova što znači da će bit gotovo nemoguće pogoditi signature „na sreću“ bez da se posjeduje private key.

256 bitova znači da postoji 2256 mogućih kombinacija signature-a. To je jako jako velik broj(od 78 znamenki) što znači da je gotovo nemoguće pogoditi taj broj.

To sve teži zaključku da stvarno osoba koja ima odgovarajući private key može generirati ispravan signature.



Što sprječava nekog da nakon prve potvrde kada se dobije potvrdni potpis samo kopira tu transakciju n puta? Trik je u tome što svaka transakcija nosi svoj ID što znači da signature nikad neće bit isti.

Sada dolazimo do pojma decentraliziranosti

Dakle svaki sudionik ima svoju kopiju „javne knjige“ (Ledger). Kada neki sudionik želi napraviti transakciju broadcast-a to svim ostalim sudionicima na mreži koji to čuju i upišu u svoju javnu knjigu.

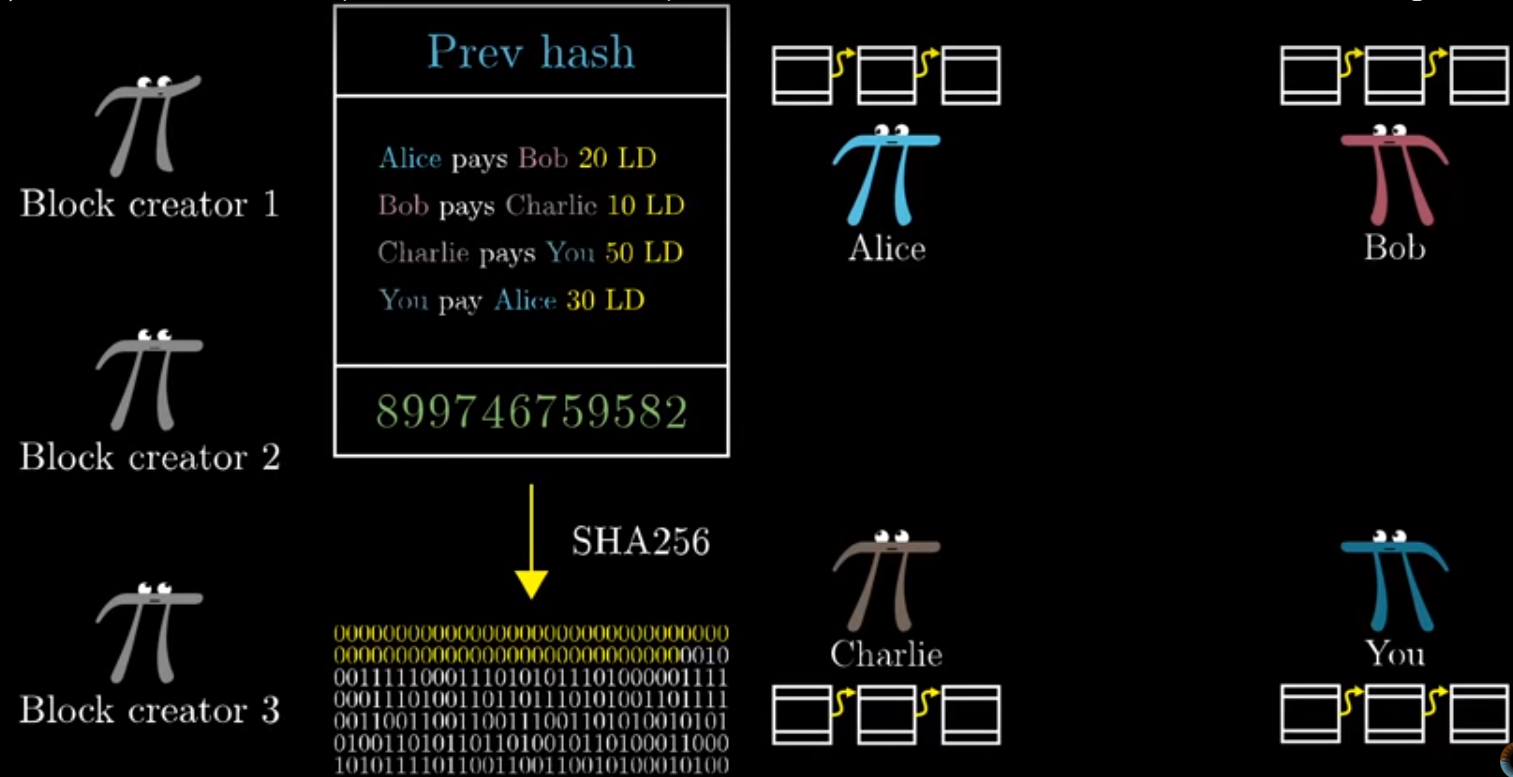
Međutim ukoliko se ne učini nešto više, ovo je veoma loš sistem. Kako biti siguran da se svi slažu koja je javna knjiga ispravna? Kako biti siguran da svi ostali slušaju transakcije koje dolaze? Npr neko vama uplati 20 dolara i kako biti siguran da su i svi ostali sudionici „čuli“ da vam je poslano 20 dolara te da će te tih 20 dolara moći kasnije koristiti.

Problem je rješen tako da se vjeruje onoj javnoj knjizi (Ledger) koja ima najviše Computation worka (računalnog rada). Dakle računalni rad se uzima kao mjera kome vjerovati. Vjeruje se uvijek onom zapisu koji ima više računalnog rada 🡪 proof of work!!! 🡪 ranije spomenuti.

Sha256 je kriptografska hash funkcija što znači da je nemoguće otkriti parametar s kojim je generirana ukoliko imamo sha256 hash kodnu riječ. Dakle nije moguće otkriti u suprotnom smjeru o kojem podatku je riječ. OVO SVE BOLJE OBJASNITI NA PRIMJERU. Jedina opcija je nasumično pogađanje...

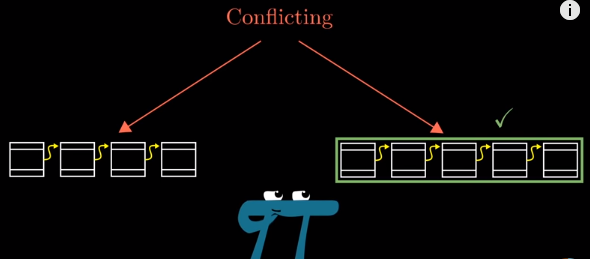
Hash da bi bio prihvatljiv mora počinjat s određenim brojem nula.

Dakle slušaju se sve transakcije koje se potom stavljaju u blok te se traži valjan hash za taj blok. Nakon što se nađe hash, taj se blok šalje svim sudionicima na mreži da ga dodaju na kraju svoje kopije blockchaina.



Block creatori se nazivaju Mineri (rudari). Sve što oni rade je da slušaju sve transakcije i probaju otkriti valjani hash prije ostalih minera. Miner koji prvi otkrije valjani hash dobije nagradu. Nagrada je određeni broj coina koji se prikupi od takse koja se uzme od svake transakcije.

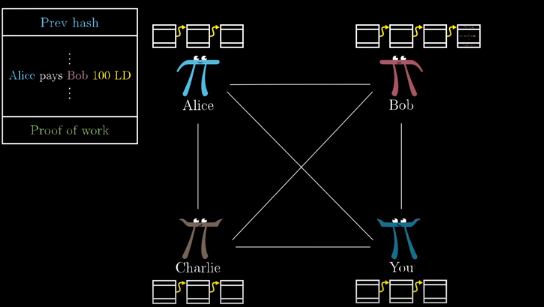
Svi ostali sudionici koji samo žele raditi transakcije ne slušaju na sve transakcije nego samo na blokove koji se emitiraju od strane block minera.



Dakle ključ svega ovoga je ukoliko čujemo dva različita tj.konfliktna blockchaina uvijek vjerujemo onom koji je duži jer je za njega izdvojena veća količina računalne snage.

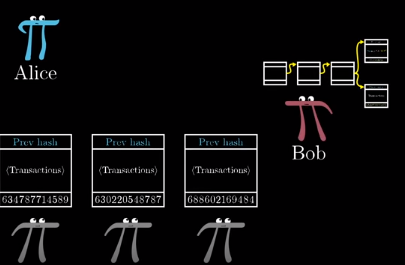
Pokazat ćemo primjer gdje ćemo pokušati „prevariti“ ovaj sistem.

Npr Alice će poslati Bobu 100 dolara bez da emitira svoju transakciju svim ostalim sudionicima.

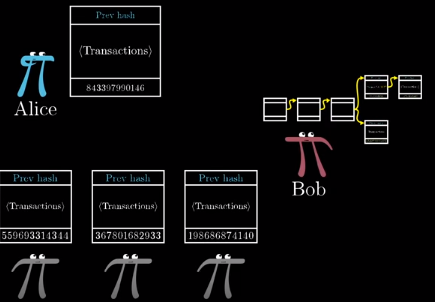


Bob doda taj blok na kraj svog lanca. Dakle svi ostali sudionici ne znaju za tu transakciju i vjeruju da Alice još ima tih 100 dolara. Da bi bilo moguće da blok pošalje samo Bobu, treba pogoditi odgovarajući hash prije svih Minera koji rade pogađanje hasha. To je sasvim moguće. Moguće je da se dogodi da Alice pogodi pravi hash prije svih ostalih minera i pošalje Bobu transakciju bez da ostali sudionici vide tu transakciju.

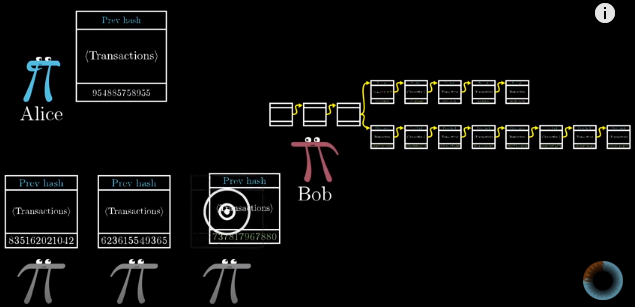
Poanta je u tome što će Bob i dalje slušati emitirane blokove od Minera te će mu se u jednom trenutku granati njegov lanac.



Trenutno ne zna koji je lanac ispravan jer oba imaju isto blokova tj. Računalne snage uloženo. Da bi Alice uspjela u svom naumu mora i dalje pogađati hash svih ostalih blokova prije svih drugih rudara.



Alice će možda pogoditi hash par puta prije ostalih minera ukoliko ima toliko sreće, ali svakako neće moći održati taj lanac dužim dugoročno. Sve dok Alice nema najmanje 50% računalne snage u odnosu na sve ostale rudare gotovo je nemoguće da se to dogodi. Dakle svi drugi mineri će brže tražiti valjan hash bloka nego Alice sama te će njihov lanac bit duži i Bob će zaklljučiti kako je Alicina grana lanca pokušaj prevare.



To znači da se odmah ne vjeruje na osnovu svakog bloka koji dođe nego da se pričeka još par blokova te se onda zaključi koji je lanac duži odnosno ispravan.

**Decentralized application**

Prvo demo aplikacije.

Sad se vjerojatno pitate: „ok ovo je prejednostavno, zasto mi pokazujes ovo?“. Zato što je razlika između obične aplikacije i ove aplikacije u tome što je ovu aplikaciju nemoguće hakirati.

Nitko ne može mijenjati glasove. Ok možda je moguće hakirati ali potrbno je imati oko 5000 najmoćnijih računala na svijetu da rade zajedno, eto koliko je puno računalne snage potrebno. A to nema nitko na svijetu, što znači da je nemoguće hakirati.

Ova je aplikacija decentralizirana. Mi smo svi programirali centralizirane aplikacije.

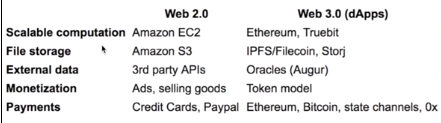
Započnimo s pričom o WWW (world wide web). Početkom 90-tih za stavit neku stranicu na internet niste mogli zakupiti host kao i danas nego ste morali kupit server ( što je u biti super računalo) koje ste stavili u neku prostoriju i na njega pohranili podatke koje ste željeli javno podjeliti svijetu.

Dakle WWW se sastojao od grupe čvorova od kojih svaki sadrži neke podatke vlasnika servera.

Onda su došle ideje u pružanju server usluga. Dakle vlasnici servera su za naknadu pružali korisnicima mogućnost korištenja njihovog servera(zakupa određene količine memorije). Došle je do pojave velikih čvorova koji hostaju puno stranica itd. Svelo se na to da je WWW skoro centraliziran.

Ti su čvorovi Google, Amazon, Instagram, Facebook,... Svi zahtjevi prolaze kroz ove centralne čvorove. A to je loše, veoma loše. Dakle netko posjeduje ogromnu količinu podataka. Nije bitna namjena, je li zlonamjerno ili dobronamjerno, ali sama činjenica da netko posjeduje tolike količine podataka su zastrašujuće. Oni te podatke kasnije mogu prodavati vladi ili nekim tvrtkama, evo kao što je sad bia skandal s Facebook-om. Facebook je najbolji primjer koliko je loše da je netko vlasnik tolike količine podataka. Oni doslovno znaju sve o vama. Npr. Uber može vidjeti u svatkom trenutku sve uber vožnje, te iz tih podataka može npr zaključiti ko noći negdi jednu noć itd.

Dakle zaključak je mi plaćamo da mi njima dajemo podatke.



Steps:

Step 1. setting up environmnet ( ethereumjs-testrpc, web3) – web3 je thin cilient koji nam daje mogućnost interakcije sa smart contractom.

Step 2. creating voting smart contract

Step 3. interacting with the contract via the remix interface ( deploying contract)

Step 4 – creating GUI interface

Ukoliko želimo napraviti bilo koju izmjenu na contractu nakon objave moramo objavit novi contract. Dakle ne možemo modificirat postojeći conract kao kod običnih aplikacija gdje samo editamo kod.

**Dodati:**

* **Solidity**
* **Web 3**
* **Testrpc**
* **remix**
* **Smart c app**
* **Opisati vlastitu aplikaciju sve kako radi u detalje**
* **Private/public key**
* **Tools koje sam koristia (remix)**
* **Metamask – private test network**
* **Browserify (tools)**
* **Koliko su zapravo sigurni smart contracty ? preispitat tvrdnje te donjeti zakljucak hoce li se to u buducnosti poboljsat il ce propast.**
* **Gas – svaka operacija uzima određenu svotu**