Wat zijn sidechains en hoe werken sidechains?

Onderzoeksdocument Semester 6

Luc Urlings

# Begrippenlijst

|  |  |
| --- | --- |
| **Begrip** | **Uitleg** |
| Mainchain | Hiermee wordt gerefereerd naar het **LISK** mainnet. |
| DApp | Een gedistribueerde applicatie |
| PoW | [Proof of Work](https://en.wikipedia.org/wiki/Proof_of_work) |
| PoS | [Proof of Stake](https://en.wikipedia.org/wiki/Proof_of_stake) |
| DPoS | [Delegated Proof of Stake](https://en.bitcoinwiki.org/wiki/DPoS) |
| Consensus mechanism | Manier voor bepalen van waarheid. Voor bitcoin en (voorlopig nog) bij ethereum wordt dit gedaan door middel van PoW, bij Lisk is dat DPoS |

# Inhoudsopgave

[**Documenten revisies**](#_nlpgoyca73bt) **2**

[**Begrippenlijst**](#_s5i3upe1uzuv) **3**

[**Inhoudsopgave**](#_3e2cz4hbk5al) **4**

[**Inleiding**](#_spywenmvw3yw) **5**

[**Wat is een sidechain?**](#_wl7uchpnsjcv) **6**

[Hoe werkt een sidechain ten opzichte van de mainchain en andere sidechains?](#_867ni7wtqrco) 7

[Sidechain & mainchain communicatie](#_63x5li7abkfi) 7

[2-way peg](#_6e7l24t58wmy) 7

[Federated peg](#_88p78krht7mh) 8

[SPV proof](#_hgedbl77yf4r) 8

[**Zijn er bepaalde eisen die wij aan sidechain owners moeten stellen?**](#_dn77t989xxoy) **9**

[**Hoe en welke data willen we opslaan in een sidechain?**](#_qwh34lbgehbq) **10**

[Hoe sla je data op in een blockchain?](#_9mgt2xpx2k71) 10

[Extra data meesturen](#_qmsscxf7fd7x) 10

[Hoeveelheid data](#_p2l9j1y7f73j) 11

[Gevoelige data opslaan](#_up18nhegg29s) 11

[Het ophalen van data van de blockchain](#_ghq1j03e4dif) 11

[Is er een alternatief voor al deze problemen?](#_9t37c81uzsxg) 11

[Subset meesturen](#_r0g7gwjq12oy) 12

[Welke data opslaan](#_45o7cez8koyl) 13

[**Kunnen we afdwingen dat ons product toegang heeft tot het sidechain netwerk?**](#_4gufd9wu2s5c) **14**

[**Bronnen**](#_gysydvocvqz7) **15**

# Inleiding

In dit document zal er precies uitgelegd worden wat een sidechain is. Daarna zal er uitgezocht worden hoe het werkt en aansluit op de mainchain en / of andere sidechains.

Hierna zullen de volgende vragen beantwoord worden:

* Zijn er bepaalde eisen die wij aan externe sidechains-eigenaars moeten stellen?
* Hoe en welke data willen we opslaan in de sidechains van onze marketplace?
* Kunnen we afdwingen dat ons product toegang heeft tot het sidechain netwerk?

# Wat is een sidechain?

Een sidechain gebaseerd op lisk is een volledig onafhankelijke blockchain die helemaal zelf in te richten is. Dit betekent dat de DApp ontwikkelaar volledig vrij is met wat hij doet met de sidechain en het consensus algorithm.

## Wat zien wij als een (externe) sidechain?

Een sidechain is een blockchain die gebasseerd is op dezelfde source code als (In ons projects geval) Lisk. Verder heeft een sidechain geen link met een andere blockchain.

De term “externe sidechain”, staat voor een sidechain die niet geregistreerd en/of geregistreerd is op de delegate market.

## Hoe werkt een sidechain ten opzichte van de mainchain en andere sidechains?

In dit hoofdstuk zal uitgelegd worden hoe sidechains en de mainchain met elkaar communiceren.

## Sidechain & mainchain communicatie

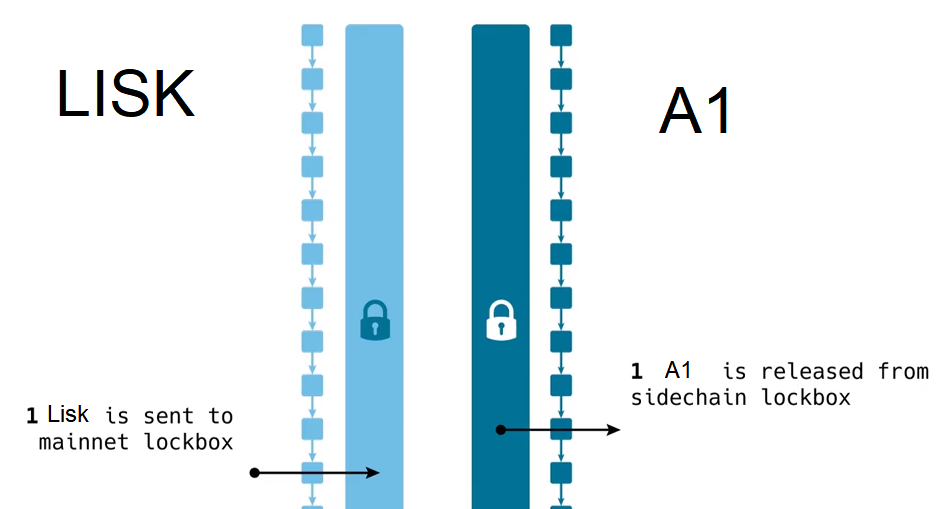
Er zijn 3 manier voor de sidechain en de mainchain om te communiceren in dit hoofdstuk zullen alle drie worden besproken.

<https://blog.rsk.co/noticia/sidechains-drivechains-and-rsk-2-way-peg-design/>

### 2-way peg

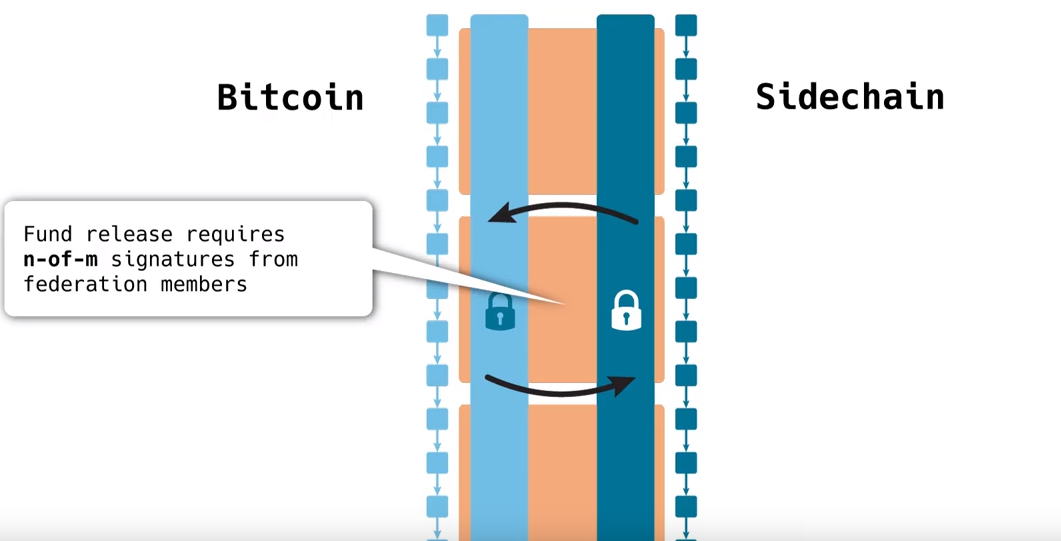
2-way peg is een manier om valuta te verwisselen tussen sidechain en mainchain, dit gebeurt door een centrale partij.  
  
Stel iemand heeft 10 Lisk en wil hier graag 10 A1-coin voor, dan is dit mogelijk door middel van een 2-way peg. Dit houdt in dat de persoon zijn 10 Lisk stuurt naar een entiteit (oftewel lockbox) die deze 10 lisk vast zet, en hier A1-coin voor los laat. Op deze manier heeft er een 2 way peg plaatsgevonden.

Het probleem dat zich hierbij kan voordoen is dat beiden “Lockboxes” door een vertrouwde partij wordt beheerd.. Deze vertrouwde entiteit heeft van beiden munten een volledige node. Hierbij heeft deze vertrouwde partij de mogelijkheid om alle munten te stelen. Dat is ook het grote gevaar.



### Federated peg

Een federated peg lijkt erg veel op een 2-way peg. Het verschil van een federated peg met een 2-way peg is dat in plaats van dat alle vertrouwen in één instantie wordt gezet wordt het verdeeld over meerdere instanties. Dit betekent dus dat de transactie door meerdere instanties met een volledige node geverifieerd moet worden. Hierdoor wordt het risico dat bijvoorbeeld binance geminimaliseerd. Echter het probleem hierbij is dat op het moment dat 1 entiteit meer dan 50% van de nodes in beheer heeft hij alsnog de kracht in handen om de chain met slechte bedoelingen te manipuleren.



### SPV proof

Bij Federated proof is het zo dat het veiliger is dan een 2-way peg, maar er is nog steeds mogelijkheid voor mensen met verkeerde bedoelingen om hun gang te gaan. Daarom is er SPV proof in het leven geroepen, hierbij is het zo dat er een aantal blokken voorbij moeten gaan voordat de betaling pas door gaat.

De technische details worden [hier](https://youtu.be/rzLhw7XI1uo?t=249) uitgelegd.

# 

# Hoe en welke data willen we opslaan in de sidechains van onze marketplace?

## Hoe sla je data op in een blockchain?

De basis transactie ziet er zo uit:

transactions.utils.getTransactionBytes({

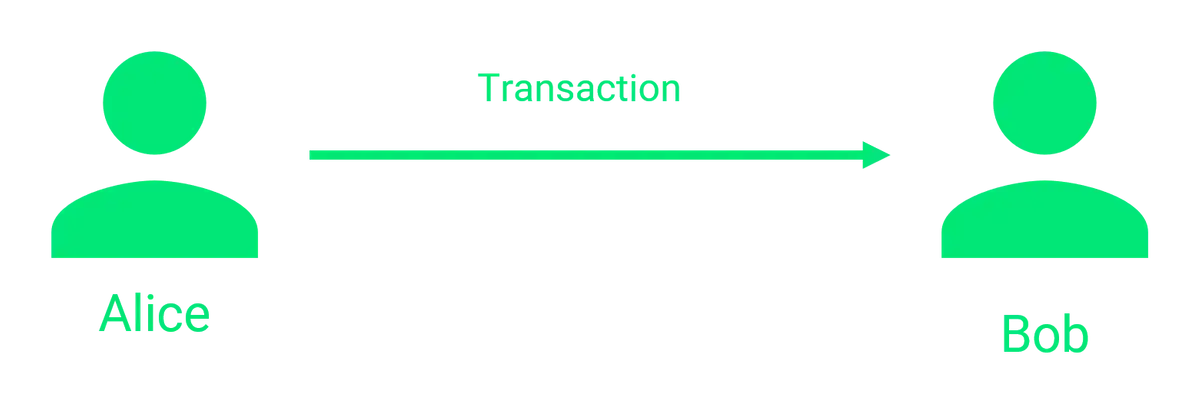
amount: 1000,

senderId: (something),

receiverId: (something)

});

Alice stuurt 1000 geld naar bob. Hierin staat alleen het adres van Alice(verstuurder) en Bob(ontvanger) en de hoeveelheid geld die wordt verstuurd.



### Extra data meesturen

Echter op het moment dat je hier data bij mee wil sturen wordt het wat ingewikkelder.

Bij Lisk is het zo dat er een asset veld meegestuurd kan worden, hier kan dan data in verwerkt worden. Dit ziet er uit als volgt:

transactions.utils.getTransactionBytes({

asset: {

msg: ‘Hello world’

},

senderId: (something),

receiverId: (something)

});

//

### Hoeveelheid data

De hoeveelheid data die opgeslagen kan worden in een transactie is erg gelimiteerd. Vaak laten blockchains maar een paar KB per transactie toe. Daarnaast wordt de transactionfee hoger naarmate de hoeveelheid data hoger is. Dit komt omdat iedere full node in het netwerk deze data moet opslaan.

### Gevoelige data opslaan

Op het moment dat er gevoelige data als persoonlijke data of confidentiële data opgeslagen moet worden in de blockchain zijn er een aantal problemen.

1. Persoonlijke data moet verwijderd kunnen worden op aanvraag van de persoon. Dit is by design niet mogelijk bij een blockchain.
2. Als er confidentiële data opgeslagen wordt in een blockchain is hier het probleem dat deze data tussen alle full nodes gedeeld word. Zeker in een publieke blockchain is dit een probleem. Bij een private blockchain kun je als eigenaar wel nog beslissen wie er toegang toe krijgt.

Een oplossing hiervoor kan zijn dat je de data encrypt. Maar dit geeft dan weer veel moeilijkheden met het distribueren van de private en public key pair.

### Het ophalen van data van de blockchain

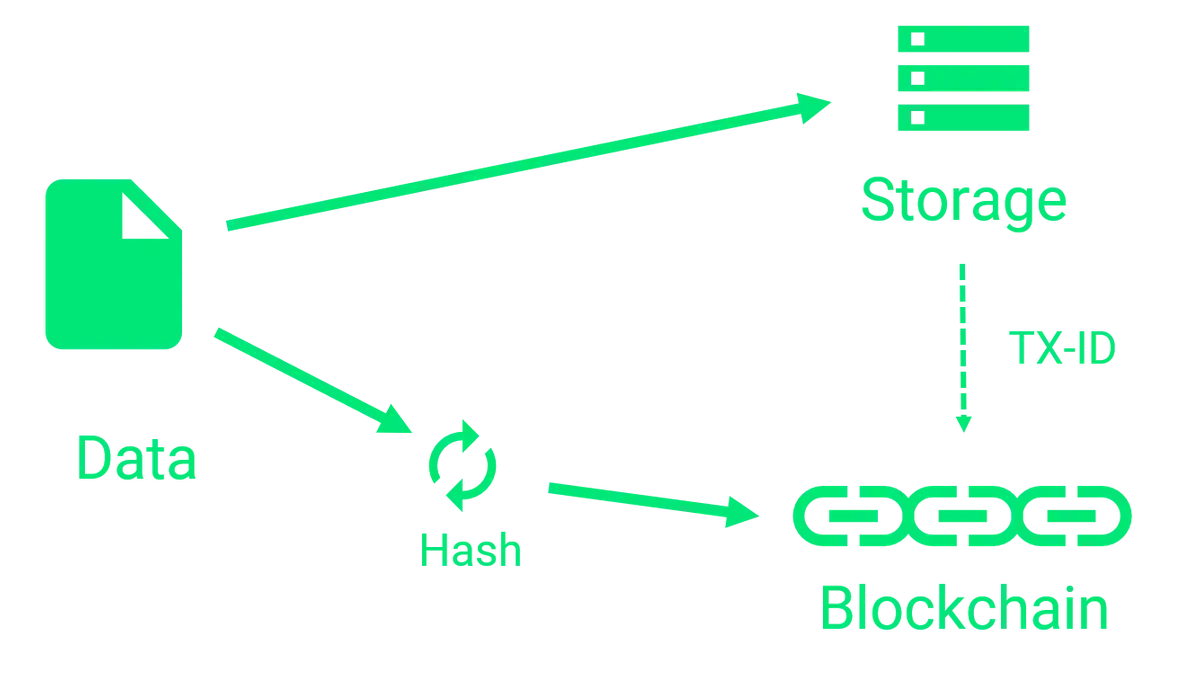
Stel we hebben wat data in de blockchain opgeslagen. Is er dan een manier om hier weer bij te komen?

Een blockchain is geen SQL server waar je simpelweg een select statement maakt. Blockchains staan alleen toe om te zoeken met het ID (hash). Dit maakt het erg moeilijk om data te zoeken.

### Is er een alternatief voor al deze problemen?

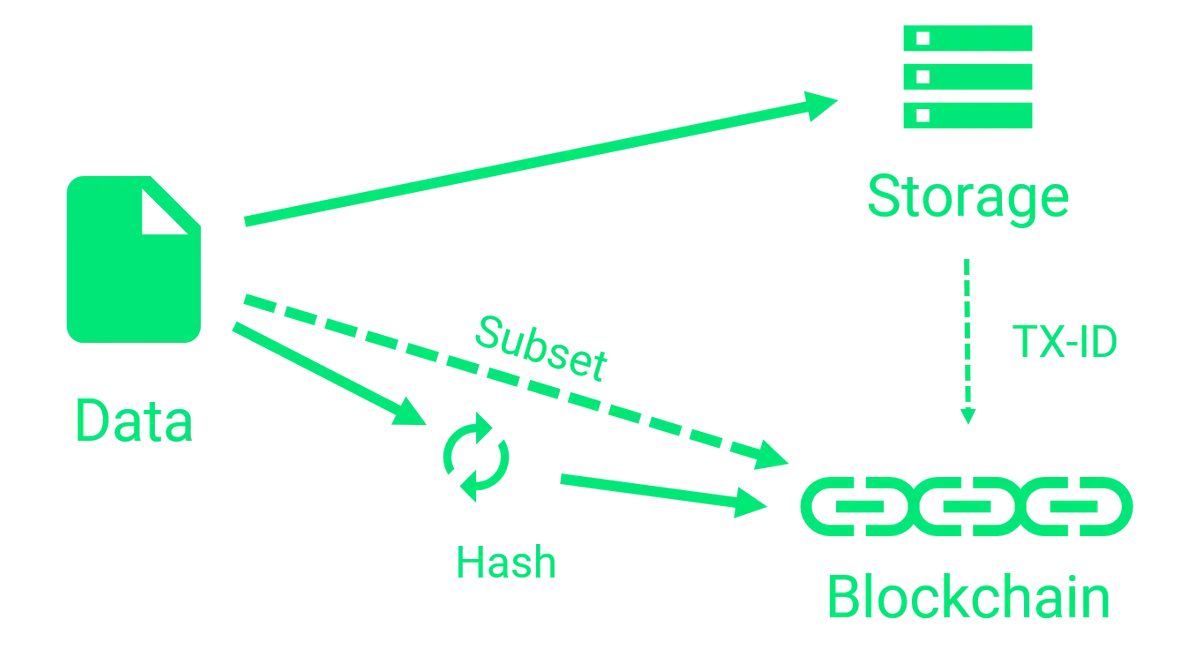
Een oplossing is om naast de blockchain een relationele database te draaien, hierbij creëer je een link tussen de twee door de data payload te hashen en dit op te slaan in de transactie in de blockchain en als transactie id op te slaan in de relationele database met daarbij de daadwerkelijke data. Hiermee los je alle bovengenoemde problemen op:

1. De data in de payload van de transactie wordt niet te groot, een hash kun je zo lang maken als je zelf wil.
2. Hiermee blijft alle data wel herleidbaar en behoud je het voordeel van een blockchain.
3. Het opzoeken van data is een stuk makkelijker, daar kan de relationele database gewoon voor gebruikt worden.
4. Data is verwijderbaar uit de relationele database, dat betekent dat de hash in de blockchain nergens meer naartoe linkt.



### Subset meesturen

Naast de hash is het nog altijd mogelijk om een subset van de data mee te sturen. Dit kan handig zijn wanneer er data wel zichtbaar moet zijn op bijvoorbeeld de blockchain explorer. Dit kan dan simpelweg aan de transactie toegevoegd worden.



## Welke data opslaan

Op het moment is het nog niet duidelijk welke data we gaan opslaan. Daarom neem ik de volgende datasets even aan:

1. persoons- en inloggegevens
2. generieke sidechain en delegate data
3. data over afspraken tussen de sidechains en delegates.

Bij dataset 1 zou heel goed het concept dat in het hoofdstuk hierboven is besproken van pas komen, Bij deze dataset is het zo dat er mogelijk gevoelige data in komt te staan. Hierbij moet het mogelijk zijn om deze te verwijderen. Daarnaast is het niet handig als gegevens van gebruikers direct inzichtelijk zijn via de blockchain explorer. Wel is het zo dat als iemand met niet de juiste autorisatie wachtwoorden probeert te veranderen kan dit automatisch door het systeem uit de blockchain verwijderd worden.

Bij dataset 2 heeft het denk ik niet veel meerwaarde en brengt het alleen maar complexiteit in het systeem. Daarom adviseer ik hier gewoon een relationele database.

Bij dataset 3 is het mogelijk wel handig om een normale blockchain te gebruiken, er zal geen grote hoeveelheid data doorheen stromen en met de blockchain explorer zijn alle contracten/ afspraken tussen een delegate en sidechain inzichtelijk.

# Kunnen we afdwingen dat ons product toegang heeft tot het sidechain netwerk?

Het is momenteel niet duidelijk welke informatie we zouden willen van het sidechain netwerk. Echter zullen er wel een aantal manieren besproken worden om data uit sidechains van gebruikers te verkrijgen.

Wanneer wij als platform delegate data willen van een sidechain is dit direct mogelijk via de API van Lisk. Het enige dat de sidechain-eigenaar dan moet doen is het IP of webadres bij ons invullen en dan kunnen wij daar live data vanaf halen.

Overigens werkt dit niet alleen voor delegates, dit kan op deze manier met alle API calls die in de lisk [API specification](https://lisk.io/documentation/lisk-sdk/reference/api.html) staan.

Op het moment dat we data uit de sidechain willen die niet beschikbaar is door de Lisk API zou een optie zijn om een stukje kant en klare code te geven aan de sidechain-owners en ze dit op een specifieke plek te zetten, op deze manier zetten we een aantal API endpoints open en kunnen we eigen data naar wens opvragen.

# Bronnen

<https://cryptobijbel.be/cryptocurrency/lisk-lsk/>

<https://coincentral.com/what-is-lisk/>

<https://allesovercrypto.nl/blog/lisk-lsk-creeer-eigen-sidechain>

<https://medium.com/@jonathangrosdubois/how-to-run-a-custom-lisk-sidechain-node-99fe9514bc17>

<http://reddit.com/r/Lisk/comments/9cq88o/cost_of_lisk_sidechain_deployment/>

<https://lisk.directory/whitepaper.pdf>

<https://www.reddit.com/r/Lisk/comments/ck3tta/proof_of_concept_sidechain_lisk_directory_a/>

<https://www.reddit.com/r/Lisk/comments/8bqhbo/lisk_has_not_revealed_how_sidechains_will/>

<https://lisk.io/documentation/lisk-sdk/reference/api.html>