Introductie

In deze inleiding beantwoorden we de belangrijkste vraag die elke auteur moet beantwoorden: Waarom zou iemand dit boek moeten lezen? Of meer specifiek: Waarom zou iemand nog een boek over blockchain moeten lezen? Lees verder en u zult zien waarom dit boek geschreven is, wat u wél en wat u niet van dit boek verwachten kunt, voor wie het geschreven is en hoe het in elkaar zit.

Waarom nog een boek over de blockchain?

De blockchain heeft veel aandacht gekregen in de publieke discussie en de media. Sommige enthousiastelingen beweren dat de blockchain de grootste uitvinding is sinds de opkomst van internet. Vandaar dat er de afgelopen jaren veel boeken en artikelen zijn geschreven over de blockchain. Als je echter meer wilt weten over hoe de blockchain werkt, raak je verdwaald in een universum van boeken die ofwel de technische details even aanroeren of de onderliggende technische concepten op een heel abstract niveau behandelen. In het eerste geval blijf je met een ontevreden gevoel achter omdat het voorbijgaat aan de technische details die je nodig hebt om de blockchain te leren waarderen, en in het laatste ook, maar nu omdat de uitleg kennis vereist die je juist wilt verwerven.

Dit boek dicht de kloof die bestaat tussen enerzijds puur technische boeken over de blockchain en anderzijds de literatuur die zich vooral bezighoudt met specifieke toepassingen en discussies over de verwachte economische impact of visies op de toekomst.

Dit boek is geschreven omdat een conceptueel begrip van de technische basis van de blockchain noodzakelijk is om specifieke blockchainapplicaties te begrijpen, business cases van blockchainstart-ups te evalueren of de discussie over de verwachte economische impact te kunnen volgen. Zonder waardering van de onderliggende concepten is het niet mogelijk de waarde en de potentiële impact van de blockchain in het algemeen te

beoordelen of de toegevoegde waarde van specifieke blockchainapplicaties te begrijpen. Dit boek concentreert zich op de onderliggende concepten, omdat gebrek aan kennis van een nieuwe technologie ertoe kan leiden dat u zich laat meeslepen door de hype en later teleurgesteld achterblijft omdat u er onrealistische, niet-onderbouwde verwachtingen van had.

Met dit boek leert u de concepten van de blockchain op een niet-technische, beknopte en begrijpelijke manier kennen. Het behandelt de drie grote vragen die rijzen bij de introductie van een nieuwe technologie: Wat is het? Waarom hebben we het nodig? Hoe werkt het?

Wat u niet hoeft te verwachten van dit boek

In dit boek doen we geen morele uitspraken over de toepassing van de blockchain. Hoewel cryptocurrency's in het algemeen en Bitcoin in het bijzonder prominente toepassingen van de blockchain zijn, behandelt dit boek de blockchain als een algemene technologie. We hebben voor deze benadering gekozen om de generieke aspecten en technische patronen van de blockchain uit te lichten in plaats van ons te richten op een specifieke en beperkte toepassingscasus. Vandaar dat dit boek:

- niet specifiek over Bitcoin of een andere cryptocurrency gaat;
- niet louter gaat over één specifieke blockchainapplicatie;
- niet gaat over het bewijzen van de wiskundige grondslagen van de blockchain;
- niet gaat over het programmeren van een blockchain;
- niet gaat over de juridische gevolgen en implicaties van de blockchain;
- niet gaat over de sociale, economische of ethische impact van de blockchain op onze samenleving en de mensheid in het algemeen.

Sommige van deze punten worden op de betreffende plaatsen in dit boek echter wel terloops aangeroerd.

Wat u kunt verwachten van dit boek

Dit boek geeft op een niet-technische manier uitleg van de technische concepten van de blockchain, zoals transacties, hashwaarden, cryptografie, datastructuren, peer-to-peersystemen, gedistribueerde systemen, systeemintegriteit en gedistribueerde consensus. De didactische aanpak van dit boek is gebaseerd op vier elementen:

- gespreksstijl;
- geen wiskundige formules;
- opeenvolgende stappen door het probleemdomein;
- gebruik van metaforen en analogieën.

Gespreksstijl

Dit boek is opzettelijk in gespreksstijl geschreven. Om het voor niet-technische lezers toegankelijk te houden wordt er geen wiskundig of computerwetenschappelijk jargon gehanteerd. Het boek introduceert en verklaart de noodzakelijke terminologie om deel te kunnen nemen aan het gesprek en om andere publicaties over de blockchain te kunnen begrijpen.

Geen wiskundige formules

Belangrijke elementen van de blockchain zoals cryptografie en algoritmen zijn gebaseerd op complexe wiskundige concepten, die op hun beurt weergegeven worden in veeleisende en soms angstaanjagende wiskundige notatie en formules. In dit boek maken we echter bewust geen gebruik van wiskundige notatie of formules om onnodige complexiteit voor niet-technische lezers te vermijden.

Incrementele stappen door het probleemdomein

De hoofdstukken van dit boek worden om een goede reden *stappen* genoemd. Ze vormen een leerpad waarlangs de kennis over de blockchain stapsgewijs wordt opgebouwd. De volgorde van de stappen is zorgvuldig gekozen. Ze behandelen de grondbeginselen van software-engineering en geven uitleg over de terminologie en waarom de blockchain

noodzakelijk is, en zetten de afzonderlijke concepten van de blockchain en zijn interacties uiteen. Door de afzonderlijke hoofdstukken te benoemen als stappen benadrukken we hun afhankelijkheid van elkaar en het didactische doel. Ze staan in een logische volgorde; het zijn geen hoofdstukken die onafhankelijk van elkaar gelezen kunnen worden.

Gebruik van metaforen en analogieën

Elke stap waarin een nieuw concept wordt geïntroduceerd begint met een illustratief verhaal door te verwijzen naar een situatie uit de werkelijkheid. Deze metaforen dienen vier belangrijke doeleinden. Ten eerste bereiden ze de lezer voor op een nieuw technisch concept. Ten tweede verminderen de metaforen de mentale hindernis om een nieuw, nog onontgonnen terrein te ontdekken door een technisch concept te verbinden met een gemakkelijk te begrijpen scenario uit het dagelijkse leven. Ten derde maken metaforen het mogelijk nieuwe concepten te leren door middel van overeenkomst en analogie. En ten slotte bieden metaforen vuistregels voor het onthouden van nieuwe concepten.

Hoe dit boek is gestructureerd

Dit boek bestaat uit 25 stappen die gegroepeerd zijn in vijf hoofdfasen. Deze vormen tezamen een leerpad waarlangs u stap voor stap uw kennis van de blockchain vergroot. De stappen behandelen enkele grondbeginselen van software-engineering, geven uitleg van de vereiste terminologie en waarom de blockchain nodig is, en gaan in op de afzonderlijke concepten van de blockchain en hun interacties. Verder wordt gekeken naar blockchain-toepassingen en lopend onderzoek en ontwikkeling.

Fase I: Terminologie en technische grondslagen

In de stappen 1 tot en met 3 worden de hoofdbegrippen van software-engineering en de terminologie die nodig is om de volgende stappen te begrijpen, uiteengezet. Aan het einde van stap 3 hebt u een overzicht van de fundamentele grondbeginselen gekregen. Bovendien hebt u nu een globaal overzicht van waar de blockchain zich ergens bevindt.

Fase II: Waarom de blockchain nodig is
In de stappen 4 tot en met 7 leggen we uit waarom de blockchain nodig is, welk probleem hij oplost, waarom de oplossing van dit probleem belangrijk is en welk potentieel de blockchain bezit. Aan het einde van stap 7 hebt u een dieper inzicht gekregen in het probleemdomein van de blockchain, in welke omgeving deze de meeste waarde heeft en waarom hij eigenlijk nodig is.

Fase III: Hoe de blockchain werkt
De derde fase vormt de kern van dit boek omdat hierin wordt uitgelegd hoe de blockchain van binnen in elkaar zit. De stappen 8 tot en met 21 leiden u langs 15 verschillende technische concepten die samen de blockchain vormen. Aan het einde van stap 21 hebt u een goed begrip van alle belangrijke concepten van de blockchain, hoe ze apart van elkaar werken en hoe ze op elkaar inwerken en met elkaar de grote machine vormen die de blockchain heet.

Fase IV: Beperkingen en hoe deze te overkomen
In de stappen 22 en 23 concentreren we ons op de belangrijkste beperkingen van de blockchain, verklaren we de redenen daarvoor en schetsen we mogelijke manieren om deze te overwinnen. Aan het einde van stap 23 begrijpt u waarom het oorspronkelijke idee van de blockchain, zoals uitgelegd in de voorgaande stappen, mogelijk niet geschikt is voor grootschalige commerciële toepassingen. Bovendien gaan we in op de wijzigingen om deze beperkingen te overwinnen en hoe deze wijzigingen de eigenschappen van de blockchain hebben veranderd.

Fase V: Gebruik van de blockchain, samenvatting en hoe verder
In de stappen 24 en 25 bekijken we hoe we de blockchain in de harde werkelijkheid kunnen toepassen en welke vragen we ons moeten stellen bij de keuze van een blockchainapplicatie. In deze fase wordt ook ingegaan op lopend onderzoek en ontwikkeling. Aan het einde van stap 25 hebt u een goed onderbouwd inzicht in de blockchain gekregen, kunt u beter overweg met ingewikkelde teksten en gaat u misschien een actieve rol spelen in de lopende discussie over de blockchain.

Begeleidend materiaal

De website **www.blockchain-basics.com** biedt begeleidend materiaal voor enkele stappen in dit boek.

Over de auteur

Daniel Drescher is een ervaren bankprofessional die posities bekleedde in de elektronische effectenhandel bij verschillende banken. Zijn recente activiteiten waren gericht op automatisering, machine learning en big data in het kader van de effectenhandel. Daniel behaalde onder meer een doctoraat in econometrie aan de Technische Universiteit van Berlijn en een MSc in software engineering aan de Universiteit van Oxford.

Over de technisch redacteur

Laurence Kirk raakte na een succesvolle carrière als schrijver van financiële *low-latency* applicaties voor de City of London gefascineerd door de mogelijkheden van het gedistribueerde grootboek. Hij verhuisde naar Oxford om te studeren voor zijn masters en richtte Extropy.io op, een consultancybureau dat met start-ups werkt aan de ontwikkeling van applicaties op het Ethereum-platform. Gepassioneerd door gedistribueerde technologie, werkt hij nu als ontwikkelaar, promotor en leraar van Ethereum.

Inhoud

Fase 1

Terminologie en technische grondslagen

In deze fase leggen we de belangrijkste concepten van software-engineering uit. Bovendien bekijken we hoe we op een georganiseerde en gestandaardiseerde wijze kunnen praten over technologie. In deze leerfase worden ook de concepten software-architectuur en -integriteit geïntroduceerd, en hoe die zich verhouden tot de blockchain. Aan het einde van deze fase hebt u inzicht gekregen in het doel en de mogelijkheden van de blockchain.

Stap 1

Denken in lagen en aspecten

Systemen analyseren door ze te scheiden in lagen en aspecten

In deze stap leggen we met de introductie van een georganiseerde en gestandaardiseerde manier van communiceren over technologie de basis van ons leerpad langs de blockchain. In deze stap wordt uitgelegd hoe je een softwaresysteem analyseert en waarom het belangrijk is om een softwaresysteem te beschouwen als een samenstel van lagen. Deze stap illustreert bovendien wat je kunt winnen door op deze manier naar de verschillende systeemlagen te kijken en hoe we met deze aanpak inzicht kunnen krijgen in de blockchain. En ten slotte geeft deze stap een korte inleiding in het concept en het belang van software-integriteit.

De metafoor

Hebt u een mobiele telefoon? Ik denk het wel, want de meeste mensen hebben er nu minstens één. Hoeveel weet u af van de verschillende draadloze communicatieprotocollen voor het verzenden en ontvangen van gegevens? Wat weet u van elektromagnetische golven die ten grondslag liggen aan mobiele communicatie? Nou, de meesten van ons weten niet veel af van deze details, want u hoeft ze niet te kennen om een mobiele telefoon te gebruiken. Bovendien hebben de meesten van ons geen tijd om dat allemaal te leren. In ons brein delen we de mobiele telefoon gewoon op in wat we moeten weten en wat we kunnen negeren en gewoon voor lief nemen.

Deze benadering van technologie is niet beperkt tot mobiele telefoons. We doen dat de hele tijd, bijvoorbeeld wanneer we een nieuwe televisie, een computer, een wasmachine enzovoort leren gebruiken. Deze mentale opdelingen zijn echter heel individueel van aard, omdat wat al dan niet belangrijk wordt geacht afhangt van iemands voorkeuren, de specifieke

technologie en onze doeleinden en ervaringen. Als gevolg hiervan kan uw mentale opdeling van een mobiele telefoon verschillen van mijn mentale opdeling van dezelfde mobiele telefoon. Dit leidt vaak tot communicatieproblemen, met name wanneer ik u probeer uit te leggen wat u moet weten over een bepaalde mobiele telefoon. Vandaar dat een eenduidige opdeling van een systeem het hoofdpunt is bij het lesgeven over en bespreken van technologie. In deze stap leggen we uit hoe een systeem wordt opgedeeld in lagen. Op die manier leggen we de basis voor onze communicatie over de blockchain.

Lagen van een softwaresysteem

In dit boek worden de volgende twee manieren om een systeem op te delen gebruikt:

- applicatie versus implementatie;
- functionele versus niet-functionele aspecten.

Applicatie versus implementatie

Door een mentale scheiding te maken tussen de behoeften van de gebruiker en de inwendige techniek van een systeem komen we tot een scheiding van de applicatielaag en de implementatielaag. Alles wat tot de applicatielaag behoort, heeft te maken met de behoeften van de gebruiker (bijvoorbeeld luisteren naar muziek, foto's maken of hotelkamers boeken). Alles wat tot de implementatielaag behoort, heeft betrekking op hoe we die dingen laten gebeuren (bijvoorbeeld het converteren van digitale informatie naar akoestische signalen, het herkennen van de kleur van een pixel in een digitale camera of het verzenden van berichten via internet naar een boekingssysteem). Elementen van de implementatielaag zijn technisch van aard en worden beschouwd als een middel om een doel te bereiken.

Functionele versus niet-functionele aspecten

Onderscheid maken tussen *wat* een systeem doet en *hoe* het doet wat het doet, leidt tot de opdeling in functionele en niet-functionele aspecten. Voorbeelden van functionele aspecten zijn het verzenden van gegevens via een netwerk, het afspelen van muziek, het maken van foto's en het

manipuleren van afzonderlijke pixels van een foto. Voorbeelden van niet-functionele aspecten zijn een mooie grafische gebruikersinterface, snel draaiende software en een mogelijkheid om gebruikersgegevens privé en veilig te houden. Andere belangrijke niet-functionele aspecten van een systeem zijn beveiliging en integriteit. Integriteit betekent dat een systeem zich gedraagt zoals het bedoeld is en daar komt veel bij kijken, zoals beveiliging en correctheid.[1] Er is een leuke manier om het verschil tussen functionele en niet-functionele aspecten van een systeem te onthouden, namelijk door te verwijzen naar de grammatica van een taal: werkwoorden beschrijven handelingen of wat wordt gedaan, terwijl bijwoorden beschrijven hoe een handeling wordt uitgevoerd. Een persoon kan bijvoorbeeld snel of langzaam lopen. In beide gevallen is de handeling 'lopen' identiek, maar hoe de handeling wordt uitgevoerd, verschilt. Als vuistregel kunt u stellen dat functionele aspecten vergelijkbaar zijn met werkwoorden, terwijl niet-functionele aspecten vergelijkbaar zijn met bijwoorden.

Twee lagen op dezelfde tijd bekijken

Herkenning van functionele en niet-functionele aspecten en opdeling van applicatie- en implementatielaag kan tegelijkertijd worden gedaan, wat leidt tot een tweedimensionale tabel. Tabel 1.1 (zie de volgende pagina) illustreert het resultaat van mentale opdeling in lagen en aspecten van een mobiele telefoon.

Tabel 1.1 verklaart mogelijk de zichtbaarheid (of niet) van specifieke elementen van een systeem voor zijn gebruikers. Functionele aspecten van de applicatielaag zijn de meest in het oog springende elementen van een systeem, omdat ze duidelijke behoeften van de gebruikers dienen. Gebruikers weten daar vaak het meest van. Aan de andere kant worden de niet-functionele aspecten van de implementatielaag zelden gezien als belangrijke elementen van het systeem. We vinden ze vanzelfsprekend.

[1] Chung L, et al. *Non-functional requirements in software engineering*. Vol. 5. New York: Springer Science & Business Media, 2012.

Laag	Functionele aspecten	Niet-functionele aspecten
Applicatie	Fotograferen	De GUI ziet er prachtig uit
	Telefoneren	Gemakkelijk te gebruiken
	E-mails versturen	Berichten worden snel verstuurd
	Websurfen	
	Chatten	
Implementatie	Gebruikersgegevens intern opslaan	Data efficiënt opslaan
	Toegang tot pixels in de digitale camera	Integriteit handhaven
	Verbinding maken met dichtstbijzijnde mobiele connector	Energie besparen Zorgen voor privacy

Tabel 1.1: Voorbeeld van mentale opdeling in lagen van een mobiele telefoon.

Integriteit

Integriteit is een belangrijk niet-functioneel aspect van elk softwaresysteem. Ze heeft drie belangrijke componenten:[2]

- **Gegevensintegriteit** De door het systeem gebruikte en onderhouden gegevens zijn volledig, correct en vrij van tegenstrijdigheden.
- **Gedragsintegriteit** Het systeem gedraagt zich zoals bedoeld en bezit geen logische fouten.
- **Beveiliging** Het systeem kan de toegang tot gegevens en functionaliteit beperken tot alleen bevoegde gebruikers.

De meesten van ons beschouwen de integriteit van softwaresystemen als vanzelfsprekend, omdat we gelukkig meestal communiceren met systemen die hun integriteit behouden. Dit is te danken aan het feit dat programmeurs en softwareontwerpers veel tijd en moeite hebben gestoken in integriteit en het behoud daarvan bij de ontwikkeling van systemen. Misschien zijn we zelfs enigszins verwend geraakt en geven we de softwaremakers die systemen bouwen met een hoge mate van integriteit niet de waardering die ze verdienen. Maar onze gevoelens veranderen zodra we met een systeem werken dat dit niet doet. U wordt u mogelijk geconfronteerd met gegevensverlies en onlogisch softwaregedrag, of merkt u dat vreemden toegang

[2] Boritz JE. IS practitioners' views on core concepts of information integrity. *International Journal of Accounting Information Systems* 6.4 (2005): 260–279.

hebben tot uw privégegevens. In die situaties wordt u boos op uw mobiele telefoon, computer, e-mailsoftware, tekstverwerker of rekenmachine, en vergeet u uw goede manieren! Op zo'n moment realiseren we ons dat software-integriteit iets heel waardevols is. Het is dus geen verrassing dat softwareprofessionals veel tijd besteden aan dit ogenschijnlijk kleine, niet-functionele aspect van de implementatielaag.

Wat volgt

In deze stap hebben we u ingevoerd in enkele algemene principes van software-engineering. We illustreerden met name de begrippen integriteit en functionele versus niet-functionele aspecten alsmede applicatie versus implementatie van een softwaresysteem. Als u deze concepten begrijpt, hebt u ook meer zicht op het grotere gebied dat de blockchain beslaat. In de volgende stap presenteren we u het grotere geheel met behulp van de concepten die in deze stap zijn geïntroduceerd.

Samenvatting

- Systemen kunnen worden geanalyseerd door ze op te delen in:
  - applicatielaag en implementatielaag;
  - functionele en niet-functionele aspecten.
- De applicatielaag richt zich op de behoeften van de gebruiker en de implementatielaag op het laten gebeuren van de dingen.
- Functionele aspecten richten zich op *wat* wordt gedaan en niet-functionele aspecten op *hoe* de dingen worden gedaan.
- De meeste gebruikers zijn geïnteresseerd in de functionele aspecten van de applicatielaag van een systeem, terwijl ze weinig oog hebben voor de niet-functionele aspecten van een systeem, met name die van de implementatielaag.
- Integriteit is een belangrijk niet-functioneel aspect van elk softwaresysteem en kent drie belangrijke elementen:
  - gegevensintegriteit
  - gedragsintegriteit
  - beveiliging
- De meeste softwarefouten, zoals gegevensverlies, onlogisch gedrag of vreemden die toegang hebben tot iemands privégegevens, zijn het resultaat van geschonden systeemintegriteit.