

Stap voor stap je vrijheid terug met open source

Stap voor stap je vrijheid terug met open source

**Een praktische gids om alle software te vervangen door
open source, van pc en smartphone tot server**

Ir. Erik Mols

24 mei 2026

Brave New Books

Stap voor stap je vrijheid terug met open source

Colophon

Dit document is ge-typeset met KOMA-Script en \LaTeX , gebruikmakende van de kaobook class.

Publisher

Eerste druk, mei 2026 door Brave New Books

ISBN: 9789465466637

De Geallieerden kwamen ons bevrijden van de Duitse bezetter. Ze bleven hangen met de dollar. Ze bezetten ons met hun software. Ze probeerden onze politiek te beïnvloeden met hun socials. Ze vergaten dat vrijheid een Europese uitvinding is.

– Erik Mols

Voorwoord

Toen ik op mijn dertiende voor het eerst het computerlokaal van het St. Odulphuslyceum in Tilburg betrad, wist ik niet dat ik op een bijzondere plek stond. Op nog geen 200 meter van waar nu *OS-Sci* gevestigd is, stonden daar machines die voor veel Nederlandse scholen nog toekomstmuziek waren: *Commodore 64's*, *Vic 20's*, een *Commodore PET*, en zelfs een *Philips-computer*. Elke vrijdagmiddag was het lokaal voor ons open, en konden we urenlang coderen — of, zoals het vaak ging, elkaars code overtypen. We groeiden op in een tijd waarin delen de norm was: muziek van lp's werd overgetapt op cassettebandjes, en software? Die kopieerden we gewoon. Er was nog geen term als *open source*, maar de mentaliteit was er wel: *wat je had, deelde je*.

Die mentaliteit droeg ik met me mee, van de kelder van de VD — waar we urenlang de opgestelde computers in beslag namen — tot december 1983, toen ik met een gespaard bedrag (en een flinke duw van mijn ouders) mijn *eerste eigen Commodore 64* in huis haalde. Officieel een Sinterklaascadeau voor mijn jongere broer en mij, maar al snel bleek dat ik de enige was die er urenlang op programmeerde. Die machine zou me zeven jaar lang vergezellen, tot ik in 1990 mijn eerste *80286* kocht. De wereld was toen al aan het veranderen: waar hardware ooit geleverd werd met technische tekeningen en onderdelenlijsten, was dat niet meer vanzelfsprekend. *Open hardware maakte plaats voor gesloten systemen*. Zelfs de beroemde brief van Bill Gates aan computerhobbyisten — waarin hij afrekende met het gratis kopiëren van software — was al geschiedenis. Toch bleven wij, opgegroeid in een cultuur van delen, gewoon doen waar we goed in waren: *software kopiëren, aanpassen, en doorgeven*.

In 1992 kocht ik een *80486*, en mijn oude *80386* — die ik zelf had ge-upgraded van een *80286* — kreeg een nieuw leven met *Linux*. Zonder dat ik het doorhad, was ik het domein van de open source binnengestapt. Een wereld die later, tijdens mijn studies *biologie* en *bio-informatica*, alleen maar groter zou worden. Want in dat vakgebied was (en is) open source software niet zomaar een optie — het is de *standaard*.

Terugkijkend op die 42 jaar voelt het als een *onrechtvaardigheid* dat ons

zelfbeschikkingsrecht over de spullen die we met onze eigen centjes hebben gekocht, langzaam is afgenomen. Sterker nog: er bestaat hardware waar in de voorwaarden staat dat het *openen van het apparaat* de rechten van de ontwikkelaar schendt. Dat is niet alleen onredelijk — het is *onacceptabel*.

Dit boek is mijn antwoord daarop. Het is een *praktische gids* om stap voor stap je vrijheid terug te pakken, zonder dat je het hele traject hoeft af te leggen dat ik de afgelopen vier decennia heb doorlopen. Want die vrijheid? Die verdien je. Die hoort bij *jou*.

Dit boek is geschreven in de trend van alles kan, niets moet. Het is een handleiding en onderbouwing hoe jij stapsgewijs onze wereld kunt verbeteren. Dit boek bevat geen stap voor stap handleidingen over hoe je, je computer of telefoon overzet. In dit boek laat ik zien dat het kan. Je vind in dit boek dus geen handleidingen hoe je, je Windows 10 machine omzet naar Linux, we laten wel zien welke belangrijke stappen je daarbij dient te zetten en waar en hoe je hulp kunt krijgen. Dat zelfde geldt voor je telefoon of tablet. We vertellen globaal hoe en waarmee, maar niet stapsgewijs. Mede ook omdat software behoorlijk vaak veranderd, er er weinig dingen zo irritant zijn als screenshots in een boek, die niet matchen met de huidige realiteit.

Inhoudsopgave

Preface	v
Inhoudsopgave	vii
I Grondslagen	1
1 Wat is open source en open hardware?	3
1.1 Definities en kernwaarden	3
1.2 Geschiedenis: Van Stallman tot moderne toepassingen . .	8
1.3 Verschil tussen open source, free software, en open hardware	9
1.4 Voorbeelden: Linux, Arduino, Lomiri, en andere projecten	11
1.5 Waarom open source meer is dan alleen software	12
2 De redenen om over te stappen	17
2.1 Vrijheid: Geen vendor lock-in, controle over je eigen data .	17
2.2 Kosten: Besparen op licenties en abonnementen	18
2.3 Privacy en veiligheid: Waarom open source vaak veiliger is	19
2.4 Duurzaamheid: Langere levensduur van hardware	20
2.5 Ethische overwegingen: Vermijden van gesloten systemen	21
2.6 Casus: Vergelijking tussen gesloten vs. open alternatieven	22
II Voorbereiding	25
3 Voorbereiden op de overstap	27
3.1 Mentale voorbereiding: Omgaan met angst voor verandering en verwachtingen managen	27
3.2 Doormodderen met Windows en macOS	29
3.3 Technische voorbereiding	30
3.4 Hulpmiddelen: Overzicht van tools en bronnen voor een soepele overstap	36

3.5	Checklist: Een stappenplan om klaar te zijn voor de overstap	37
III	Praktijk	39
4	De Computer	41
4.1	Hardware: De belofte van open hardware	42
4.2	Besturingssysteem	46
4.3	Kantoorsoftware	52
4.4	Multimedia	54
4.5	E-mail en communicatie	55
4.6	Gaming	55
4.7	Back-up en beveiliging	56
4.8	Casus: Voorbeeld van een volledige overstap	57
5	De smartphone	63
5.1	Besturingssystemen	63
5.2	Apps	67
5.3	Privacy: Instellingen om tracking te minimaliseren	70
5.4	Casus: Vergelijking tussen Android vs. Ubuntu Touch	71
6	De Server	75
6.1	Besturingssystemen	75
6.2	Webhosting	76
6.3	Cloud-diensten	78
6.4	Databases	80
6.5	Beveiliging	81
6.6	Automatisering	83
6.7	Casus: Thuis-server voor back-ups, media-streaming, of persoonlijke cloud	86
IV	It's the Community ...	99
7	Gemeenschap en bijdragen	101
7.1	Waarom bijdragen? De kracht van open-source gemeenschappen	102

7.2	Hoe draag je bij?	103
7.3	Lokale initiatieven	106
7.4	Onderwijs: Hoe open source een rol kan spelen in educatie	109
7.5	Casus: Succesverhalen van bijdragers en gemeenschappen	111
8	Toekomstvisie	115
8.1	De impact van open source	115
8.2	Uitdagingen	118
8.3	Jouw visie	122
8.4	Oproep tot actie: Hoe lezers zelf kunnen bijdragen aan een open toekomst	125

Deel I

Grondslagen

Wat is open source en open hardware?

1

Inleiding

Open source en open hardware zijn meer dan alleen technische concepten, ze representeren een *filosofie* van vrijheid, samenwerking, en transparantie. In dit hoofdstuk verkennen we de definities, de geschiedenis, en de kernwaarden die deze bewegingen drijven. We kijken ook naar het verschil tussen open source, free software, en open hardware, en waarom deze principes verder reiken dan alleen software.

1.1 Definities en kernwaarden

Wat is broncode

Stel je voor dat je een *Legobouwset* koopt. In de doos zitten niet alleen de steentjes om het model te bouwen (bijvoorbeeld een auto of een kasteel), maar ook de *bouwplaten*: stap-voor-stap instructies die precies laten zien *hoe* je elk onderdeelje moet plaatsen om het eindresultaat te bereiken. Die bouwplaten zijn de *broncode* van je Lego-creatie. Zonder die platen zou je misschien wel de losse steentjes hebben, maar niet weten *hoe* je ze in elkaar moet zetten om het gewenste resultaat te krijgen.

Of denk aan het *geheime recept van Coca-Cola*. Het eindproduct, de frisdrank in je glas, is wat je proeft en ziet. Maar het echte waardevolle deel is het *recept*: de precieze lijst van ingrediënten

1.1 Definities en kernwaarden . . .	3
1.2 Geschiedenis: Van Stallman tot moderne toepassingen . . .	8
1.3 Verschil tussen open source, free software, en open hardware . . .	9
1.4 Voorbeelden: Linux, Arduino, Lomiri, en andere projecten	11
1.5 Waarom open source meer is dan alleen software	12

en de stappen om die ingrediënten te mengen. Dat recept is de *broncode* van Coca-Cola. Zonder dat recept zou je de smaak kunnen kopiëren door te proeven, maar je zou nooit *precies* weten hoe het gemaakt is, en je zou het ook niet kunnen aanpassen.

Broncode vs. uitvoerbare software

Bij software werkt het net zo. *Broncode* is de menselijk leesbare tekst die programmeurs schrijven in een programmeertaal (zoals Python, C++, of Java). Deze code bevat instructies die de computer moet uitvoeren, maar in een vorm die *wij* kunnen begrijpen. Bijvoorbeeld:

```
print("Hallo, wereld!")
```

Listing 1.1: Een eenvoudig voorbeeld van broncode in Python

Dit is een eenvoudig voorbeeld van broncode in Python. Het zegt tegen de computer: *“Toon de tekst ‘Hallo, wereld!’ op het scherm.”*

Maar computers begrijpen geen Python, Nederlands, of enige andere menselijke taal. Ze begrijpen alleen *machinecode*: een reeks enen en nullen (binaire code) die direct door de processor kan worden uitgevoerd. In de onderstaande code, zie je de machine code, die “Hallo Wereld” op het scherm print.

```
01001000 10111000 00000000 00000000 00000000
      00010000 00000000 00000000
01001000 10111111 00000000 00000000 00000000
      00100000 00000000 00000000
11111111 11010000)
```

Listing 1.2: Een eenvoudig voorbeeld van machinecode

Om van broncode naar machinecode te komen, wordt de code *gecompileerd*. Dit is als het ware het proces waarbij de Lego-bouwplaten (bronco-

de) worden omgezet in een afgebouwd model (uitvoerbaar programma), of het recept wordt omgezet in een kant-en-klare fles Coca-Cola.

Waarom is broncode belangrijk?

Wanneer software *gecompileerd* is, is de originele broncode vaak niet meer terug te vinden in het eindproduct. Het resultaat is een bestand vol enen en nullen dat de computer kan uitvoeren, maar dat voor mensen onleesbaar is. Dit is vergelijkbaar met:

- ▶ Een *Lego-model dat al in elkaar gezet is*: je ziet het eindresultaat, maar niet hoe het in elkaar zit.
- ▶ Een *fles Coca-Cola*: je proeft de smaak, maar zonder het recept weet je niet hoe het gemaakt is.

Bij *open source software* is de broncode *wel* beschikbaar. Dit betekent dat iedereen:

- ▶ Kan *lezen* hoe de software werkt (net als de Lego-bouwplaten bekijken).
- ▶ Kan *aanpassen* wat niet bevalt (net als het recept aanpassen om je eigen variant te maken).
- ▶ Kan *leren* van hoe anderen software hebben geschreven (net als koken leren door recepten te bestuderen).

Bij *gesloten software* (zoals veel commerciële programma's) is de broncode *niet* beschikbaar. Je krijgt alleen het afgebouwde model of de fles frisdrank, maar niet de instructies of het recept erachter. Hierdoor ben je afhankelijk van de maker voor updates, reparaties, of aanpassingen.