



## PUBLIEKE SOFTWARE OPEN TENZIJ

### Over dit document

Dit document is het resultaat van een verkenning naar

- hoofdpunten voor wie open source software wil inkopen
- belangrijke ingrediënten om een ICT inkoopbeleid op te stellen gericht op het verwerven van onafhankelijkheid als overheidsorganisatie en
- verwijzingen naar overzichten en achtergronden bij deze hoofdpunten.

De hoop is dat dit playbook overheden kan inspireren tot vernieuwende manieren van aanbesteden van ICT, die bijdragen aan belangrijke doelen van deze tijd: continuïteit, gezonde marktwerking, samenwerking en ketensamenwerking, innovatie, bedrijvigheid, veiligheid, transparantie en vooral hetgeen waarop al deze doelen rusten: onafhankelijkheid.

### Versie

Versie 2.0 12 mei 2021

### Inhoud

Over dit document.....	1
1.Publieke Software Open Tenzij, waar gaat het over?.....	2
1.1.Wat is open source software?.....	3
2.Open source, waarom zou je het willen?.....	6
2.1.Continuïteit.....	6
2.2.Leveranciersonafhankelijkheid.....	6
2.3.Marktwerking.....	7
2.4.Samenwerking en ketensamenwerking.....	7
2.5.Innovatie en bedrijvigheid.....	8
2.6.Veiligheid.....	8
2.7.Rechtsstatelijke argumenten.....	9
2.8.Alles komt samen in het primaire proces.....	9
3.Publieke software aanbesteden, wat is het probleem?.....	11
3.1.Het probleem volgens de literatuur.....	11
3.2.Wat ervaren mensen als het probleem?.....	14
3.3.Zicht op antwoorden.....	17
4.Op welke vragen kennen we het antwoord?.....	18
4.1.Wat is open source in de context van aanbesteden.....	18
4.2.Vragen over eigenaarschap en het beschikbaar stellen van code en principiële en juridische vragen rond mededinging.....	22
5.Strategische vragen.....	23
5.1.Concrete strategieën.....	24
6.Naar een andere manier van denken.....	27
Credits.....	32
Licentie.....	32
Literatuur.....	33



## PUBLIEKE SOFTWARE OPEN TENZIJ

### 1. Publieke Software Open Tenzij, waar gaat het over?

*Het gaat in dit document over de drempels die overheden ervaren bij het voldoen aan al lang bestaande voorkeur om open source software de voorkeur te geven en de nieuwe beleidslijn om overheidssoftware actief vrij te geven.*

Op 17 april 2020 verscheen een [beleidsbrief](#) van staatssecretaris Knops van Binnenlandse Zaken en Koninkrijksrelaties over het vrijgeven van de broncode van overheidssoftware. De staatssecretaris roept overheden op om de broncode van overheidssoftware actief vrij te geven.

Het vrijgeven van rechten betekent uiteraard dat overheden die rechten moeten hebben. Daarom is dit onderwerp direct gekoppeld aan het aanbesteden van software. Maar wellicht zijn er ook andere drempels en vragen die overheden ervaren bij het vrijgeven van de overheidssoftware, het opstellen van een open source beleid of de al lang bestaande beleidslijn om open source software de voorkeur te geven.

Dit document bevat allereerst een verkenning van dergelijke mitsen en maren. Onzekerheden die overheden nu ervaren zijn zoveel mogelijk ingedeeld in antwoorden, verwijzingen en het benoemen van de ruimte om hieraan een eigen invulling te geven. De antwoorden vormen ook een “playbook” een stappenplan voor wie overheidssoftware open source wil maken of overheidssoftware open source wil aanbesteden.

*Dit document is een eerste versie van een playbook voor publieke software:*

- *voor wie strategisch wil nadenken over software in het publiek belang*
- *voor wie een open source beleid wil opstellen voor zijn organisatie*
- *voor wie publieke software wil aanbesteden of open wil maken*



## PUBLIEKE SOFTWARE OPEN TENZIJ

### 1.1. Wat is open source software?

*Vrije en open source software is software waarbij je niet alleen het recht krijgt om de software te gebruiken, maar ook om die te begrijpen, te wijzigen en te kopiëren.*

Open source software is software met een licentie die is bedacht om de gebruiker meer rechten te geven. Dit maakt een andere manier van werken en ontwikkelen mogelijk. In grote lijnen zijn er vijf categorieën van licenties.

alle vrijgeven <	rechten onder auteursrecht			> alle behouden
wettelijk	rechten op de code in licentie			
publiek domein	toegeeflijk	beschermend	propriëtair	bedrijfsgeheim
vrij van alle restricties	behoudt auteursrecht, maar verleent licentienemer alle rechten	toegeeflijk, maar met restricties om de software open te houden bij wijzigingen	verleent enkel(e) recht(en)	behoudt alle rechten
	vrije en open source licenties			

Bron: Mark Webbing<sup>1</sup> vertaald en met toevoegingen

Wanneer het auteursrecht is verlopen of wanneer een auteur expliciet afstand doet van zijn auteursrecht spreken we over *publiek domein*. Er zijn dan geen restricties. Aan de andere kant van het spectrum staan nieuwe zeer restrictieve licenties bij diensten via de cloud of software-as-a-service. De software is dan *bedrijfsgeheim* en kan alleen gebruikt worden via het Internet. *Propriëtaire* licenties zijn de licenties die de meeste mensen zullen kennen. Je mag de software doorgaans alleen uitvoeren en gebruiken. Deze categorie is zeer divers. Zowel de bekende commerciële licenties vallen hieronder, als ook allerlei bijzondere varianten als shareware, freeware, adware, open core<sup>2</sup> en dergelijke. Dergelijke software is in privaat bezit en de broncode ervan staat onder private controle.

Daarnaast zijn er twee soorten van open source licenties. Toegeeflijke licenties<sup>3</sup> zijn licenties die (vrijwel) alle rechten verlenen aan de licentienemer. Een gebruiker mag de software dus ook begrijpen. Dit betekent dan ook dat de broncode moet worden

- 1 Webbing werkte als advocaat, hoogleraar en juridisch adviseur van Red Hat, een tot de recente overname door IBM beursgenoteerd bedrijf dat één van de Linux distributies uitbrengt.  
[https://web.archive.org/web/20140122163130/http://www.redhat.com/fi/summitfiles/presentation/May31/Open%20Source%20Dynamics/Troan\\_OpenSourceProprietyPersp.pdf](https://web.archive.org/web/20140122163130/http://www.redhat.com/fi/summitfiles/presentation/May31/Open%20Source%20Dynamics/Troan_OpenSourceProprietyPersp.pdf)
- 2 Businessmodel waarbij een 'community' versie wel een open source licentie wordt geleverd en een 'enterprise' versie onder een propriëtaire licentie. De enterprise versie is dan niet open source.
- 3 Dit type open source licenties wordt ook wel BSD-licenties genoemd, naar de universiteit van Californië, Berkeley, die zo'n licentie voor het eerst gebruikte.



## PUBLIEKE SOFTWARE OPEN TENZIJ

meegeleverd. Hiernaar verwijst het begrip 'open source'. En een gebruiker mag deze code ook wijzigen en kopiëren. En wat een gebruiker bij toegankelijke licenties ook mag is een wijziging aanbrengen en de gewijzigde software weer uitbrengen onder een propriëtaire licentie. Apple baseert de kern van zijn besturingssystemen op softwareprojecten onder toegankelijke open source licenties en brengt ze vervolgens met toevoegingen opnieuw uit onder een eigen (propriëtaire) licentie<sup>4</sup>.

Hier zit het verschil met de andere soort open source licenties. Deze vereisen dat software onder een open source licentie publiek blijft en niet later onder een restrictievere licentie uitgebracht kan worden.

De Europese Unie heeft een beschermende licentie uitgebracht, de European Union Public Licence ([EUPL](#)), die zowel verenigbaar is met de lokale wetgeving van de 27 lidstaten als met populaire open source licenties.

Het belangrijke verschil met propriëtaire licenties is dat open source software niet onder private controle staat. Het verschil is dus niet per se dat de broncode op een openbare codebibliotheek als *GitHub* staat, dat de software gratis gebruikt kan worden of via Internet gedownload. Het verschil tussen publieke en private controle ligt in het verlenen van bepaalde rechten of vrijheden. Daar verwijst 'vrij' in vrije en open source software naar.

*Er zijn twee belangrijke groepen open source licenties*

- *toegankelijke licenties, die het mogelijk maken om de software opnieuw uit te brengen onder private controle*
- *beschermende of publieke licenties, die de software onlosmakelijk verbinden met de licentie om te zorgen dat de code publiek blijft*

<sup>4</sup> Aan veel open source projecten draagt Apple ook bij: <https://developer.apple.com/opensource/>

**PUBLIEKE SOFTWARE OPEN TENZIJ**

Verleende rechten	Publiek domein	Toegeeflijk	Publiek	Propriëtair	Restrictief
Rechtenvoorbeeld	Wettelijk	BSD	GPL	Windows EULA	Bedrijfsgeheim
Softwarevoorbeeld	SQLite	Apache webserver	GNU/Linux	Windows	Buienradar
Auteursrecht behouden	Nee	Ja	Ja	Ja	Ja
Recht om uit te voeren	Ja	Ja	Ja	Ja	Nee
Recht om te begrijpen	Ja	Ja	Ja	Nee	Nee
Recht om te wijzigen	Ja	Ja	Ja	Nee	Nee
Recht om te kopiëren	Ja	Ja	Ja	Nee	Nee
Recht om te verspreiden onder restrictievere voorwaarden	Ja	Ja	Nee <sup>5</sup>	Nee	Nee

Overzicht rechten onder diverse categorieën licenties. Bron: Widlak, A.C., 2021, Volwassen Digitale Overheid, gebaseerd op een indeling van Mark Webbing, deels aangepast en uitgebreid met sublicentiëren en voorbeelden.

De bovenstaande tabel illustreert de verschillen nogmaals. Zowel publiek domein als toegeeflijke open source licenties bieden de mogelijkheid om code opnieuw uit te brengen onder restrictievere voorwaarden. Beschermende of publieke licenties verlenen alle rechten onder licentie, behalve de mogelijkheid om de code opnieuw uit te brengen onder beperktere voorwaarden. Propriëtaire en zeer restrictieve licenties verlenen slechts enkele rechten.

<sup>5</sup> Vaak is er wel de mogelijkheid om te kiezen voor een latere versie van een beschermende licentie. Daarin kunnen extra restricties opgenomen zijn zoals bijvoorbeeld een verbod op “tivoization” in GPL v3. <https://en.wikipedia.org/wiki/Tivoization> Eigen restricties toevoegen echter blijft onmogelijk.



## 2. Open source, waarom zou je het willen?

*Open source software biedt andere rechten en daarmee andere mogelijkheden. Voor de een is het een middel om betere marktwerking te organiseren, voor de ander gaat het om rechtsstatelijke argumenten, zoals transparantie en voor weer een ander om organisatorische doelen.*

Open source software biedt andere mogelijkheden, maar die mogelijkheden moeten uiteraard praktisch georganiseerd worden willen ze werkelijkheid worden. Eerst staat de vraag centraal welke doelen je als overheidsorganisatie zou kunnen nastreven.

### 2.1. Continuïteit

Open source software geeft iedereen het recht die software te begrijpen en te wijzigen. In praktische zin betekent dit dat je die software kunt beschouwen als permanent onderdeel van je eigen organisatie en dat je meerdere aanbieders kunt organiseren rond de software die je als organisatie gebruikt. Dat is belangrijk, omdat software in de praktijk als een infrastructuur is voor organisaties. Via software werken mensen samen en organisaties delegeren beslissingen aan software. In het ideale geval behoud je daarom die software en bouw je daarop voort, ook wanneer je van aanbieder wisselt. Dat zorgt voor continuïteit.

### 2.2. Leveranciersafhankelijkheid

De mogelijkheid om van aanbieder te wisselen, zonder van software te wisselen, maakt je minder afhankelijk van een specifieke leverancier. Dat kan in specifieke gevallen potentieel moeilijke en dure migratietrajecten voorkomen. Behalve wisselen van aanbieder, is het ook mogelijk om meerdere aanbieders *tegelijk* aan dezelfde software te laten werken. Dat kan in eenzelfde rol, waarbij meerdere aanbieders gelijktijdig werken aan uitbreidingen, verbeteringen, het wegwerken van fouten of in de werkzaamheden die nodig zijn om software beschikbaar te maken als een dienst via de cloud. En dat kan ook in complementaire rollen, waarbij de ene leverancier nieuwe onderdelen ontwikkelt, een andere aanbieder die integreert en een derde aanbieder die onderhoudt. Omdat je de ontwikkeling op zo'n wijze kunt organiseren, organiseer je tevens dat meerdere partijen inhoudelijke kennis hebben van de software. Je organiseert dat aanbieders elkaars documentatie testen en meer zaken die het reëel maken dat verschillende aanbieders elkaars werk kunnen overnemen. Dit betekent ook dat juridische waarborgen, zoals Escrow<sup>6</sup>, niet nodig zijn. Zulke waarborgen zijn dan georganiseerd - opgenomen in de werkwijze - en daarmee niet juridisch, maar praktisch geborgd.

<sup>6</sup> Een broncode-escrow is een juridische overeenkomst. De leverancier van software deponert de broncode bij een derde partij. Deze derde partij geeft de broncode vrij in bijzondere gevallen, zoals faillissement.



## PUBLIEKE SOFTWARE OPEN TENZIJ

### 2.3. Marktwerking

Open source schept de mogelijkheid om kennis te nemen van de werking van de software. Naarmate meer aanbieders die kennis ook daadwerkelijk hebben, omdat een bredere groep aanbieders betrokken is bij de software, is het beter mogelijk om een marktordening te organiseren die daadwerkelijk tot marktwerking leidt. Naarmate overheden meer code actief delen, is het aantrekkelijker voor bedrijven om te investeren in kennis van die code. Er zit veel overlap in de code van overheidsorganisaties en het werk daaraan. Het kan een doel op zich zijn om code te kunnen delen, bijvoorbeeld omdat dit een kostenbesparing kan opleveren. Maar breed gebruik van dezelfde code maakt die ook interessanter voor bedrijven. Zij kunnen met die kennis op meerdere plekken aan de slag<sup>7</sup>. Ook dat is een stap op weg naar een marktordening die daadwerkelijk tot marktwerking leidt. Uiteraard vergt dit een strategie<sup>8</sup> waarin actief beheer van de software is afgestemd op het actief organiseren van een bredere groep van aanbieders.

### 2.4. Samenwerking en ketensamenwerking

Ongehinderd samenwerken over organisatiegrenzen heen is een belangrijke kern van wat open source software mogelijk maakt. Samenwerking tussen organisaties is vandaag ook altijd een kwestie van samenwerking tussen softwaresystemen ofwel *interoperabiliteit*<sup>9</sup>. Ook de technische transparantie draagt daaraan bij. Hoe beter je weet hoe iets werkt, hoe makkelijker het is om daarop aan te sluiten. Dat geldt voor overheden onderling, maar dit maakt het natuurlijk ook voor bedrijven makkelijker om aan te sluiten op systemen en te anticiperen. Wanneer overheden onderling samenwerken en één van hen gebruikt software die onder private controle staat kan dit onderling aansluiten lastig maken. Dit schept ruimte voor leveranciers om private belangen boven het belang van samenwerking of het belang van de keten te stellen. Zij kunnen dan samenwerking op het niveau van software onmogelijk maken. Dit is inefficiënt vanuit overheidsperspectief, omdat meerdere partijen dan in hun eigen cocon aan hetzelfde probleem moeten werken, maar uiteraard voordelig vanuit het perspectief van een aanbieder.

7 De investering in het kennismaken van de interne werking van de software verschuift hiermee ook deels van de kopende partij (de overheid), naar de marktpartijen.

8 In beleidstermen: de regierol en governance zijn essentieel.

9 In beleidsstukken staat op dit punt vaak dat open source software geen voorwaarde is voor interoperabiliteit en open standaarden wel. Dat is niet helemaal juist. Maar belangrijker is dat dit beeld doet geen recht aan de praktijk. Open standaarden en open source software zijn verschillende werelden. Open standaarden hebben hun oorsprong bij grote bedrijven, ontwikkelen zich langzaam en hebben een verre tijdshorizon. Ze ontstaan indien meerdere bedrijven voldoende belang hebben om na jarenlange onderhandelingen een specificatie te implementeren. Open source software ontwikkelt zich sneller en heeft een meer pragmatische insteek. Open source software gebruikt open standaarden als ze er zijn en anders is een onofficiële microstandaard snel ontwikkeld. Open standaarden en open source software versterken elkaar in termen van interoperabiliteit, mede door deze verschillen. Een open standaard is er niet altijd. En open standaarden zijn - soms moedwillig - niet altijd juist geïmplementeerd. Open source software maakt dergelijke problemen op een pragmatische manier en op een redelijke termijn oplosbaar, omdat de implementatie transparant is. Dat maakt tevens de ontwikkeling van een open standaard en de toetsing of de werking juist is makkelijker.

## PUBLIEKE SOFTWARE OPEN TENZIJ

### 2.5. Innovatie en bedrijvigheid

Het is niet slechts voor onderwijs en wetenschappelijk onderzoek van groot belang dat de broncode van software gepubliceerd beschikbaar is. Inmiddels rust de gehele IT industrie in elke sector op open source software<sup>10</sup>. De enorme snelheid waarmee vandaag een idee in een concrete toepassing kan worden gegoten, kan alleen bij de gratie van de brede beschikbaarheid van open source componenten. Of het nu gaat om databases, front-end frameworks of modules, de mogelijkheid om in relatief korte tijd een complete toepassing te realiseren zou voor slechts een fractie van de ontwikkelaars open staan zonder open source software. De investeringen in tijd en geld zouden anders simpelweg te hoog zijn. In de praktijk zien we dan ook dat een groot en steeds groter deel van de commerciële software open source componenten bevat. Volgens een jaarlijkse audit<sup>11</sup> van inmiddels meer dan 1500 commerciële softwarepakketten bevatte elk open source componenten en gemiddeld 75% van de code bestond uit open source software. Ook commerciële software is inmiddels in de praktijk voor het grootste deel open source software. Starten als bedrijf zonder een basis of aanvulling in open source software is nauwelijks meer reëel, maar met open source software kan een startend bedrijf meeliften op miljoenen regels code die zich al in de praktijk hebben bewezen<sup>12</sup>.

*Starten als bedrijf zonder een basis in open source software is nauwelijks meer reëel, maar met open source software kan een startend bedrijf meeliften op miljoenen regels code die zich al in de praktijk hebben bewezen*

- *Ook commerciële software is in de praktijk voor het grootste deel open source software: gemiddeld 75%*
- *Software in elke sector rust op open source software.*
- *Open source software is het fundament van de gehele IT industrie*

### 2.6. Veiligheid

Openheid en veiligheid in de betekenis van cybersecurity lijken voor veel mensen intuïtief tegengesteld. Het intuïtieve argument is dat een fout die onbekend is, niet misbruikt kan worden. Het tegenargument is dat het beter is om in alle openheid een systeem te bedenken waarvan je kunt beredeneren dat het veilig is. Cryptografie is het ultieme voorbeeld daarvan. Je demonstreert de veiligheid en juist de consensus onder kritische

10 Het gaat om 100% in marketing en tech, 98% in de gezondheidszorg, 97% in de financiële dienstverlening en 92% in de retail en e-commerce van de software die rust op open source software. OSSRA, 2021.

11 Black Duck by Synopsys brengt jaarlijks het OSSRA rapport uit, een analyse van veiligheidsaspecten van open source software. De editie van 2021 moet worden aangevraagd (<https://www.synopsys.com/software-integrity/resources/analyst-reports/open-source-security-risk-analysis.html>) Dit is de editie van 2020: <https://www.synopsys.com/content/dam/synopsys/sig-assets/reports/2020-ossra-report.pdf>.

12 Zie ook: Edo Plantinga, Johan Groenen, Hilbrand Bouwkamp, Ad Gerrits, Winst op (tien) punten <https://ibestuur.nl/nieuws/winst-op-tien-punten>





## PUBLIEKE SOFTWARE OPEN TENZIJ

kenners geeft je vertrouwen. Empirisch<sup>13</sup> blijken veelgebruikte open en gesloten systemen<sup>14</sup> elkaar weinig te ontlopen in termen van de ernst en aantallen kwetsbaarheden. De onderzochte open systemen publiceren die kwetsbaarheden wel sneller en lossen ze veel sneller op. Het open source concept “vele ogen maken fouten schaars” gaat in die zin op.

In de praktijk echter gebruiken organisaties niet alleen grote en goed onderhouden systemen. Dat geldt voor open en gesloten software. En wanneer organisaties zelf functionaliteit laten ontwikkelen, is die niet altijd onderdeel van een brede discussie. Die praktijk is waarschijnlijk veel belangrijker voor de veiligheid. Wie software gebruikt, wil daarom zeker stellen dat iemand die software onderhoudt en kwetsbaarheden oplost, of dit nu een (impliciet) onderdeel is van de licentiekosten bij gesloten software - waarbij je de leverancier moet vertrouwen - of van een (expliciet) onderhoudscontract bij open source software, waarbij je dit zelf kunt controleren.

### 2.7. Rechtsstatelijke argumenten

Naarmate we meer beslissingen delegeren aan software, is die software ook meer bepalend voor het maatschappelijk resultaat. Daarmee krijgt software een meer publiek karakter. Dit geldt voor software waarmee we individuele administratieve besluiten nemen, maar steeds vaker ook voor software die op basis van statistiek conclusies trekt, zoals systemen die gebruik maken van big data en kunstmatig intelligente systemen. Als software bepaalt wanneer het stoplicht op groen staat en of en wanneer dit gebeurt voor kolonnes van vrachtwagens, dan ligt er een publieke waarde-afweging besloten in de software. Aan de software bij individuele beslissingen kunnen we zien hoe de wet in de praktijk wordt uitgevoerd. Het belangrijkste motief is hier transparantie. Niemand zou een land open en vrij noemen, wanneer de wet niet gekend kan worden. Naarmate meer beslissingen gedelegeerd worden aan software geldt dat ook voor overheidssoftware.

### 2.8. Alles komt samen in het primaire proces

Al deze motieven komen samen in het primaire proces van overheidsorganisaties. Uit het bestuursrecht kan nog geen onverkorte en algemene plicht worden afgeleid tot technische transparantie, maar dit komt wel al voor, voor de procespartijen<sup>15</sup>. Als software onder private controle staat, brengt dat een risico met zich mee voor wie niet voorbereid is om aan dergelijke wettelijke eisen te kunnen voldoen. Of die transparantie ook in praktische zin vorm krijgt is niet alleen afhankelijk van de vraag of een brede professionele gemeenschap kennis *kan* nemen van de software, maar ook van de mate waarin dat feitelijk gebeurt. Het

<sup>13</sup> Schryen, 2009, 2010, 2011

<sup>14</sup> Er is een open en gesloten browser, email client, webserver, officepakket, besturingssysteem en database management systeem vergeleken.

<sup>15</sup> Beginselen als 'technische transparantie' zijn nog geen onderdeel van het bestuursrecht en bestaande beginselen zijn nog niet door de rechter uitgelegd op een wijze dat ze technische transparantie volledig afdekken. In specifieke gevallen (niet onverkort) voor procespartijen (niet algemeen) komt dit wel al voor. Wolswinkel, J. (2020). Willekeur of algoritme? Laveren tussen analoog en digitaal bestuursrecht. Tilburg University.



**KAFKA**  
BRIGADE

## PUBLIEKE SOFTWARE OPEN TENZIJ

ontstaan van een meer transparante markt en meer transparante overheidssoftware versterken elkaar. Meer gebruik van dezelfde code binnen de overheid versterkt dit ook en maakt de overheid bovendien overzichtelijker. En ook dat is weer een stap naar een professionele gemeenschap met kennis van de interne werking van de overheid.

Uiteraard zijn deze motieven in het algemeen geen staande praktijk, maar een streven. Echter geen enkele opgave kan zinnig of succesvol zijn, zonder ideaal.



### 3. Publieke software aanbesteden, wat is het probleem?

*Het verlies van onafhankelijkheid is uiteindelijk de kern van het probleem. Alle andere vragen zijn een worsteling met de vraag hoe dit probleem te adresseren.*

De behoefte om een verantwoordelijkheid als geheel te delegeren aan een (één) leverancier via een contract is vaak het uitgangspunt van denken. Een oplossing waarbij het gebruik van de software hetgeen is dat je koopt en al het andere zoveel mogelijk de verantwoordelijkheid is van de leverancier lijkt dan het ideaal. *Hoe* de leverancier levert is in deze lijn van denken hetgeen waarover de koper zo min mogelijk wil nadenken. Hij betaalt als het ware om daarover niet te hoeven nadenken. Dit kennen we ook wel als de behoefte aan “ontzorgen”.

Denken in termen van “ontzorgen” heeft een aantal implicaties, met name:

- Wat er nodig is om de dienst te leveren is voor de koper een *black box*. De koper heeft immers geen kennis van de activiteiten die de leverancier onderneemt om de dienst te leveren.
- Garanties en zekerheden zijn *juridisch* van aard. De nadruk ligt niet op de activiteiten die de leverancier onderneemt om die garanties en zekerheden te bieden, maar op de verantwoordelijkheid wanneer dit resultaat uitblijft.

Zolang de relatie voortduurt en de leverancier de dienstverlening in alle opzichten naar tevredenheid levert, ervaart de koper een zeker gemak. Dit gemak zit in de eenvoud van een enkele transactie en de eenvoud van specificatie van een resultaat zonder de wijze van organisatie die daarbij hoort. Dit verandert wanneer wijzigingen nodig zijn.

#### 3.1. Het probleem volgens de literatuur

De inkoop van ICT leidt nu tot afhankelijkheden. In de literatuur<sup>16</sup> heet de situatie waarin organisaties hun onafhankelijkheid verliezen ‘lock-in’. Dit is een sterke vorm van zogenaamde pad-afhankelijkheid, waarbij keuzes uit het verleden een voorspelbare en disproportioneel grote invloed hebben op keuzes later. Er zijn vijf<sup>17</sup> zelfversterkende mechanismen die daarin een rol spelen.

<sup>16</sup> Een bespreking hiervan waarop deze paragraaf mede is gebaseerd is te vinden in Paapst, 2013:20-30,92-95

<sup>17</sup> Woerdman, 2004:26, beschrijft er vier, waaraan Paapst, 2013:161 nog een vijfde toevoegt op basis van onderzoek in Nederland.



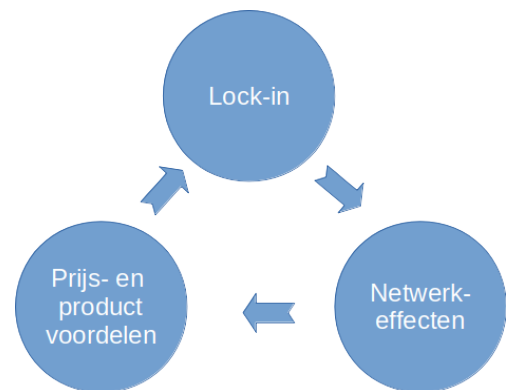
## PUBLIEKE SOFTWARE OPEN TENZIJ

### Leereffecten

Bij leereffecten gaat het om de ervaring die mensen in de organisatie hebben met een systeem. Competentie met bestaande technologie maakt een overstap naar nieuwe en betere technologie lastig<sup>18</sup>.

### Netwerk- en coördinatie-effecten

Bij netwerk- en coördinatie effecten gaat het om een diversiteit aan schaalvoordelen. Het bekendste netwerk-effect is in essentie het fenomeen dat een goed meer waarde krijgt als er meer gebruikers zijn. Het huidige categorie-management en strategisch leveranciersmanagement zijn daar voorbeelden van. Daarbij bundelen overheden hun inkoopkracht om zo ofwel financiële voordelen te verkrijgen (categoriemanagement) of aanpassing van de voorwaarden aan de Nederlandse wetgeving (strategisch leveranciersmanagement). Deze behoefte is begrijpelijk, maar leidt ook tot een netwerkeffect. In ruil voor korting of andere voorwaarden bestendigt en versterkt een leverancier zijn dominante positie<sup>19</sup> en tevens vaak de lock-in van de koper. Voor individuele organisaties wordt het lastiger om af te wijken, omdat een aankoopproces bij de dominante leverancier is gestroomlijnd. Het is relatief kostbaarder om af te wijken. De dominante leverancier is goedkoper, omdat meerdere partijen gezamenlijk een afspraak maken met hem en dit bovendien ook zijn transactiekosten verlaagt. Er zijn meer netwerk- en coördinatie effecten<sup>20</sup>.



Illustratie van de zelfversterkende relatie tussen lock-in en netwerkeffecten.

### Verzonken kosten

Bij verzonken kosten gaat het om de investeringen die reeds zijn gedaan voor het huidige systeem. Er zijn sterke psychologische mechanismen die maken dat verzonken kosten keuzes voor de toekomst beïnvloeden, al is dat vaak niet rationeel<sup>21</sup>.

### Overstapkosten

Overstapkosten zijn de kosten en risico's om een alternatief te integreren in de organisatie. Feitelijk zijn deze kosten exit-kosten, die horen bij de bestaande oplossing. (zie ook kader hieronder)

### Objectieve compatibiliteit

Naast de bovenstaande mechanismen uit de literatuur blijkt in Nederland ook objectieve compatibiliteit een rol te spelen bij de overstap naar (open source) alternatieven. Dit is de

18 Swanson, 1994:1082

19 Er zijn meer effecten te noemen, zoals de informatiepositie, de relatie en de erkenning van voorwaarden en importantie.

20 Zie bijvoorbeeld Windrum, 1999.

21 [https://en.wikipedia.org/wiki/Sunk\\_cost#Fallacy\\_effect](https://en.wikipedia.org/wiki/Sunk_cost#Fallacy_effect)



## PUBLIEKE SOFTWARE OPEN TENZIJ

situatie dat de technische verwevenheid zodanig sterk is dat organisaties zich gedwongen voelen bepaalde keuzes te maken<sup>22</sup>.

### **Hoe zit dit in Nederland?**

*In Nederland promoveerde Mathieu Paapst in 2013 op (onder meer) dit vraagstuk. Bij de overstap naar een (open source) alternatief blijken leereffecten, verzonken kosten, overstapkosten en objectieve compatibiliteit een rol te spelen.*

*Overstapkosten zouden geen rol moeten spelen. Het zijn feitelijk exit-kosten, die horen bij de bestaande oplossing, mede omdat bij een nieuwe aanbesteding er weer sprake moet zijn van een level playing field. Toch geven respondenten<sup>23</sup> aan dat dit de belangrijkste reden is om te kiezen voor een voorkeursleverancier.*

Uit deze afhankelijkheden vloeien andere problemen voort, die je ook kunt zien als de inverse van alle motieven uit het tweede hoofdstuk of als een drempel of barrière om dergelijke doelstellingen te bereiken. Ook het streven om interoperabiliteit te bereiken via open standaarden kan hierdoor gehinderd worden en wordt dit ook<sup>24</sup>.

Uiteraard is het verkrijgen van een voorkeurspositie in de markt en een machtspositie ten opzichte van kopers, onder meer via het tot stand brengen van lock-in precies de kern van de strategie van veel ICT-bedrijven<sup>25</sup>. Het maximaal “wegzetten” van een verantwoordelijkheid, waardoor de wijze waarop de leverancier de dienst levert ongrijpbaar wordt, schept daarvoor de ruimte. Dit maakt vervolgens dat wisselen van leverancier vraagt om een veelomvattende exit-strategie. Waar afhankelijkheid aanvankelijk slechts een juridische kwestie was, moet zo'n exit-strategie de praktische afhankelijkheid adresseren die daaruit is voortgekomen:

- *De infrastructuur als geheel is geen onderdeel van de eigen organisatie.* Wanneer onderdelen van de infrastructuur niet transparant verplaatsbaar zijn, niet toegankelijk of niet inwisselbaar én in privaat bezit, dan is er afhankelijkheid in praktische zin. Onderdelen staan dan feitelijk onder private controle.
- *Kennis over de infrastructuur ontbreekt, waarin alsnog geïnvesteerd moet worden.* Een exit-strategie vereist dat helder is wat precies overgedragen moet worden aan een andere partij. Het ontbreken van volledig inzicht daarin brengt risico's met zich mee voor de continuïteit en kosten om dit inzicht alsnog te verkrijgen.

22 Paapst, 2013:93

23 Paapst, 2013:93

24 Paapst, 2013:134 e.v.

25 Bv. Schmidt & Schnitzer, 2003, Maurer & Scotchmer, 2006

## PUBLIEKE SOFTWARE OPEN TENZIJ

Het verlies van onafhankelijkheid is uiteindelijk de kern van het probleem. Het heroveren daarvan is een politiek-bestuurlijk belang dat de basis vormt voor zowel organisatiedoelen als continuïteit, wendbaarheid en risicomitigatie als publieke doelen, zoals gezonde marktwerking, cyberveiligheid en transparantie.

### 3.2. Wat ervaren mensen als het probleem?

Organisaties zijn vaak al zoekend naar manieren om meer onafhankelijkheid en wendbaarheid te vinden. Virtualisatie<sup>26</sup> en het opdelen van diensten in microservices<sup>27</sup> zijn bijvoorbeeld technische middelen die organisaties daarvoor inzetten, die het tevens mogelijk maken om meer onderdelen onafhankelijk van elkaar in te kopen. De aandacht voor open source software is onderdeel van die zoektocht. Maar ook daar waar organisaties open source software inzetten is er niet als vanzelf sprake van een ideale situatie. De inzet van open source software op zich is niet gelijk aan het verwerven van onafhankelijkheid. In gesprekken met inkopers, architecten en strategen komen in grote lijnen vijf soorten problemen naar voren rond de inzet en inkoop van open source software:

1. misbegrip van wat open source software is en verwarring tussen product en recht
2. vragen over eigenaarschap en het beschikbaar stellen van code
3. strategische vragen over aanbesteding
4. juridische en principiële vragen over mededinging
5. gebrek aan een integrale visie op de relatie tussen de ontwikkeling van software en de (makkelijke) inzet daarvan via inkoop

#### ad 1) misbegrip en verwarring rond open source

De begripsverwarring over open source software is heel begrijpelijk. Enerzijds is open source slechts een juridische kwestie en gaat het om rechten en niet om een product. Anderzijds identificeren we specifieke software met de licentie waaronder die beschikbaar is. Is het nu een product of is het nu een licentie bij een product? Tenslotte is er nog zoiets als dual-licensing, waarbij een leverancier een product onder verschillende licenties aanbiedt en soms bepaalde diensten alleen beschikbaar zijn bij een bepaalde licentie.

Om hierin niet te verdwalen is het belangrijk om hier een zeer scherp onderscheid te maken. Dat is met name relevant, omdat het bij een aanbesteding veel uitmaakt wat het perspectief is dat je gebruikt. Vergelijk je producten, diensten bij producten, diensten

<sup>26</sup> Virtualisatie simuleert een computeromgeving. Dit kan het makkelijker maken om een geheel van software op virtuele hardware te verplaatsen naar een andere fysieke omgeving.

<sup>27</sup> Dit is een variant op het idee om software als een dienst aan te bieden. Een applicatie wordt dan opgezet als een verzameling van kleinere diensten, die communiceren via het breed gebruikte HTTP-protocol. Ook dit kan het makkelijker maken om onderdelen onafhankelijk van elkaar te verplaatsen en/of onder te brengen op een andere (fysieke) omgeving.



## PUBLIEKE SOFTWARE OPEN TENZIJ

verpakt als producten of vergelijk je leveranciers? Zonder scherp onderscheid ben je al snel verdwaald.

### ad 2) vragen over eigenaarschap en het beschikbaar stellen van code

Zowel kleinere als grote organisaties hebben soms twijfel over het eigenaarschap van code. Ook dit is begrijpelijk. Zolang code alleen in eigen huis gebruikt wordt, is een zorgvuldige registratie van de totstandkoming ervan niet iets dat je als noodzaak ervaart. Wanneer de vraag komt om die code actief beschikbaar te stellen, lijkt die vraag ineens veel relevanter. Soms gaat het om slordigheid. Er worden bijvoorbeeld voorwaarden vermeld uit een jaar waarin geen nieuwe voorwaarden zijn opgesteld. Soms om onduidelijkheid. “Een deel van de code is door eigen mensen ontwikkeld, een deel door mensen die zijn gedetacheerd en een deel ook door bedrijven die daarvoor zijn ingehuurd. Hebben wij nu in alle gevallen alle rechten op de code?” Eén organisatie deed daarbij de suggestie om een speciaal soort auteursrecht in te stellen, zodat altijd als code bij of voor de overheid is ontwikkeld helder is hoe de rechten liggen<sup>28</sup>. In de praktijk komt het ook voor dat organisaties betrokken programmeurs of leveranciers een zogenaamde Contributor License Agreement (CLA) laten tekenen, waarmee zij zeer ruim gebruiksrechten overdragen, aan bijvoorbeeld een stichting, waarin zij tevens de governance van de software regelen<sup>29</sup>.

### ad 3) strategische vragen over aanbesteding

In het algemeen gaan die vragen over aanbesteding<sup>30</sup> in de context van (de ontwikkeling van) open source software. De belangrijkste vragen bij aanbesteding gaan over de strategie om een bepaald resultaat te verkrijgen. In veel gevallen zijn deze vragen geformuleerd als een klacht. Daarbij zijn er veel varianten op de klacht “ik krijg niet de partijen die ik wil”. Het lukt niet om kleinere bedrijven te laten meedoen of te selecteren. Grote bedrijven zijn goed in het aanbestedingsproces, maar delegeren vervolgens het werk met grote marges aan kleine partijen. Om deze redenen ontwijken organisaties ook het aanbestedingsproces en huren mensen in (“sourcen”) of doen zij dit via organisaties waarvan zij aangeven dat het “unique selling point” is dat het geen aanbesteden heet, maar “inbesteden”.

28 Algemene voorwaarden, zoals [ARVODI](#) en [ARBIT](#), leggen de rechten volledig bij de overheid via inkoopvoorwaarden, niet via de wet. Bij fouten of bij een mengvorm - inkoop, detachering, eigen mensen - worden organisaties onzeker over hun rechten. Daarnaast kun je als organisatie veel eisen in algemene voorwaarden, maar zijn organisaties gehouden redelijke alternatieven tijdens het inkoopproces in overweging te nemen.

29 Dit heeft ook een nadeel, namelijk dat je hiermee inboet op de drempelloze werking van open source licenties. Hoewel open source licenties formeel contracten zijn, werken ze als algemene regels vanwege het gebrek aan barrières zoals transactiekosten. Er is daarom alle reden om de noodzaak tot dergelijke barrières met wetgeving te voorkomen.

30 Richtlijn 2004/18/EG is vastgelegd in de Aanbestedingswet: <https://wetten.overheid.nl/BWBR0032203/2019-04-18>  
Memorie van toelichting: <https://zoek.officielebekendmakingen.nl/kst-34329-3.html>



## PUBLIEKE SOFTWARE OPEN TENZIJ

Sommige betrokkenen ervaren als probleem dat hun organisatie niet als eenheid weet te handelen tegenover marktpartijen die over het algemeen de voorkeur hebben voor een betaald licentiemodel.

En diverse betrokkenen zijn op zoek naar strategieën om een marktordering te bereiken, waarbij een groep van “geschikte” bedrijven - technische bedrijven, de uitvoerders zelf in plaats van het bedrijf dat zich specialiseert in opdrachten verwerven - beschikbaar is. Of naar het organiseren van meerdere partijen rond software die zij hebben laten ontwikkelen. Of naar een strategie om überhaupt bedrijven te organiseren rond bestaande software.

### ad 4) principiële en juridische vragen rond mededinging

In enkele gevallen is de vraag of het beschikbaar stellen van overheidssoftware niet botst met de Wet markt en overheid<sup>31</sup>. Soms zijn dit algemene “zorgen”, waarbij vooral behoefte is aan zekerheid. De vraag om een expliciete uitzondering - die nu in de maak lijkt te zijn<sup>32</sup> - zodat volledig zeker is dat hier geen risico's liggen, is vaker gesteld. In één geval was de vraag meer van principiële aard. Daarbij ging het om een door de organisatie ontwikkelde toepassing waarvan soortgelijke toepassingen commercieel worden geëxploiteerd. Deze toepassing gaf bovendien niet alleen toegang tot de eigen gegevens van de organisatie, dan was deze jurist gerustgesteld, maar ook tot gelijksoortige gegevens in het algemeen. Afhankelijk van de directe bruikbaarheid van de software, zonder aanbieder, zou dit inbreuk kunnen maken op een reëel commercieel belang.

### ad 5) gebrek aan integrale visie op de relatie tussen ontwikkeling en exploitatie

Op allerlei plaatsen ontwikkelen overheden software die onder een open source licentie wordt gepubliceerd. Het gaat daarbij soms om modules, soms om hele toepassingen en het gaat daarbij om software die soms met ingehuurd mensen is gemaakt, soms door een bedrijf dat is ingehuurd. De vraag is vervolgens: En nu?

Er zijn respondenten die serieus en met succes bezig zijn geweest om de toekomst van een toepassing zeker te stellen, door te zorgen dat meerdere leveranciers betrokken zijn en de governance zeker is, ook in het geval de organisatie zelf ooit niet meer de belangrijkste

31 Opgenomen in de Mededingingswet <https://wetten.overheid.nl/BWBR0008691/2019-01-01>

32 Zie hiervoor onder meer de Kamerbrief “Open, tenzij” van april 2020: “ik [ga] samen met de staatssecretaris van Economische Zaken en Klimaat kijken naar de mogelijkheden voor een uitzondering voor open source software in de Wet markt en overheid. Een uitzondering zorgt ervoor dat overheden vrij zijn om software als open source [software] ter beschikking te stellen en niet langer voor situaties, waarbij de Wet markt en overheid geldt, per geval elke vrijgave zoals hiervoor beschreven afzonderlijk dienen te motiveren.”

<https://www.rijksoverheid.nl/documenten/kamerstukken/2020/04/17/kamerbrief-inzake-vrijgeven-broncode-overheidssoftware>





## PUBLIEKE SOFTWARE OPEN TENZIJ

drijvende kracht of financier kan zijn achter het softwareproject<sup>33</sup>. Maar in veel gevallen is de vraag juist hoe vanuit een situatie waarin een organisatie eigen code maakt, openstelt of doorontwikkelt gewerkt kan worden naar een situatie waarin commerciële partijen dit op een laagdrempelige wijze aanbieden aan leden, mede-overheden of andere geïnteresseerde partijen.

### 3.3. Zicht op antwoorden

Op een deel van de vragen hierboven is het antwoord helder en bekend. Op een ander deel van de vragen is het antwoord niet volledig en uitputtend bekend en beschreven, maar kunnen wel bestaande strategieën worden benoemd op hoofdlijnen en achtergrondinformatie daarbij. En met name de laatste vraag - een integrale visie op de relatie tussen software ontwikkelen en makkelijk kunnen inkopen - nodigt ook uit tot de verkenning van andere manieren van aanbesteden. Om dat goed te kunnen doen is het nodig dat er een globaal beeld is van de juridische ruimte waarbinnen andere werkwijzen kunnen worden beproefd. Hieronder wordt achtereenvolgens besproken op welke vragen we het antwoord kennen, waar de ruimte zit om anders in te kopen, bestaande concrete strategieën en tenslotte een andere manier van denken die richting kan geven aan een andere manier van inkopen en de relatie legt met de techniek zelf.

<sup>33</sup> Een voorbeeld is OpenConext dat voornamelijk in opdracht van SURF (<https://www.surf.nl/>) op basis van bestaande software door meerdere partijen wordt (door)ontwikkeld. SURF biedt de dienst aan onder de naam SURFconext voor haar leden. Ook door andere overheidspartijen zetten deze software in (aantal ministeries, DICTU, en straks mogelijk VNG). In een eerder project (MediaMosa) heeft SURF ook de governance georganiseerd. Mogelijk gaat dit ook hier gebeuren wanneer het aantal gebruikers groeit. Er is overigens een Nederlandse non-profit organisatie - The Commons Conservancy (<https://commonsconservancy.org/>) - die de governance van technologische projecten in het publieke belang als dienst aanbiedt.



## 4. Op welke vragen kennen we het antwoord?

### 4.1. Wat is open source in de context van aanbesteden

Zoals hierboven al aangegeven, begripsverwarring over open source software is veelvoorkomend en heel begrijpelijk. Het volgende citaat illustreert dit:

*“Open source, dat is eigenlijk gewoon een manier om aanbestedingen te omzeilen. Het is gratis, dus je hoeft het niet in te kopen. En allemaal mensen in onze organisatie hebben dat ook gedaan. Nu hebben we recent ontdekt dat wij allemaal code in onze productiesoftware hebben zitten, waarvoor je opeens toch moet betalen bij een bepaald aantal gebruikers. En soms ook hele rare dingen, zoals dat je iemand elke dag koffie moet brengen. En wat nog erger is: er blijken nu allemaal kwetsbaarheden in te zitten. En die kwetsbaarheden publiceren ze nog op Internet ook!”*

#### 1. “Het is gratis dus je hoeft het niet in te kopen”

Juist:

Het is juist dat iedereen de vrijheid heeft om de broncode zelf te gebruiken voor elk doel, zonder daarvoor een financiële vergoeding te geven. En dus hoeft de broncode zelf niet aanbesteed te worden<sup>34</sup>.

Onjuist:

Net als alle software zijn er allerlei werkzaamheden nodig om open source software te implementeren, op een toegankelijke manier aan te bieden en te onderhouden. De software als *product* hoeft niet te worden ingekocht. De *dienstverlening* die daarbij nodig is moet wel worden aanbesteed, tenzij dit volledig met eigen mensen gebeurt.

#### 2. “waarvoor je opeens toch moet betalen” (...) “rare eisen”

Onjuist:

Wat gratis is, is niet altijd open source software. Juist als blijkt dat de software onder private controle staat, is er geen sprake van open source software. Dat blijkt hier uit het feit dat de eigenaar eisen stelt. De eigenaar kan die eisen stellen, omdat er sprake is van private controle en dus niet van vrije en open source software.

#### 3. “er blijken kwetsbaarheden in te zitten” (...) “die publiceren ze nog op Internet ook”

Juist:

In alle software zitten kwetsbaarheden, ook in open source software. Het is gebruikelijk die kwetsbaarheden - vaak na een periode van responsible disclosure<sup>35</sup> - te publiceren.

Onjuist:

Dit is geen nadeel. Kwetsbaarheden in software is iets waar je naar moet handelen. Daarvoor moeten altijd procedures zijn. Dat kwetsbaarheden worden gepubliceerd

<sup>34</sup> Dit wordt ook wel de ‘downloadmethode’ genoemd, zie OSOSS, 2007:10.

<sup>35</sup> Er zijn diverse procedures om gebruikers eerst de mogelijkheid te geven kwetsbaarheden te herstellen.



## PUBLIEKE SOFTWARE OPEN TENZIJ

en dat er allerlei tooling is om te kijken of die kwetsbaarheden in jouw software zitten is een groot goed: dan kun je ernaar handelen. Alle software bestaat inmiddels deels - en gemiddeld zelfs in meerderheid - uit open source software. Er zal dan ook altijd capaciteit beschikbaar moeten zijn om kwetsbaarheden te verhelpen. Het is vooral te hopen de code die niet open is ook goed gemonitord en onderhouden wordt, wanneer dit impliciet is in de licentiekosten en geen expliciete dienst. Uit onderzoek aan de hand van de [database](#) van [NIST](#) blijkt dat de hoeveelheid en zwaarte van kwetsbaarheden in vergelijkbare propriëtaire en open source software elkaar niet veel ontlopen, maar dat kwetsbaarheden wel eerder opgelost en gepubliceerd worden bij open source software<sup>36</sup>.

- *De broncode van bestaande software, het product, hoeft niet aanbesteed te worden indien dit onder een open source licentie beschikbaar is<sup>37</sup>.*
- *Er is altijd dienstverlening nodig om de software te implementeren, beschikbaar te stellen en te onderhouden. De inkoop van deze dienstverlening staat in beginsel open voor elke leverancier<sup>38</sup>.*
- *Onderhoud, zoals het installeren (of maken) van correcties op de code na het ontdekken van kwetsbaarheden, is dienstverlening die altijd nodig is. Wanneer dit een impliciet onderdeel is van de licentiekosten bij propriëtaire software is het ook verstandig om dit expliciet te maken. Daarmee maak je in een aanbesteding deze dienstverlening niet alleen los van het product en daarmee vergelijkbaar, maar dit maakt tevens verantwoording daarover beter mogelijk.*

Wie open source software inzet krijgt dus ook met aanbestedingen te maken, tenzij alle werkzaamheden met eigen mensen worden uitgevoerd. Een belangrijk verschil is dat er geen sprake is van leveringen, maar alleen van diensten.

Bij het denken over aanbesteden lopen de begrippen levering (product), dienst en leverancier nogal eens door elkaar. Ook worden deze begrippen nogal eens met elkaar vereenzelvigd. De koper is uiteraard vooral bezig met doelmatig inkopen, al is dit niet het belangrijkste doel van de wetgeving. In het aanbestedingsrecht is het belangrijkste principe dat *aanbieders* gelijke kansen krijgen<sup>39</sup>. Daarbij horen beginselen als objectiviteit (iedereen moet op de hoogte kunnen zijn), transparantie (regels en procedure moet helder zijn) en

<sup>36</sup> Schryen, 2009, 2010, 2011.

<sup>37</sup> Dit wordt ook wel de 'downloadmethode' genoemd, zie OSOSS, 2007:10.

<sup>38</sup> Dit blijkt ook wel uit het feit dat zelfs Microsoft nu een diversiteit aan open source software aanbiedt via zijn cloudplatform.

<sup>39</sup> "Die ratio [van elke aanbestedingsprocedure] is dat ondernemers met gelijke kansen in kunnen schrijven op overheidsopdrachten, opdat in vrije concurrentie een optimale prijs-kwaliteitsverhouding voor de overheid tot stand komt. Alle vragen die rijzen dienen in het licht van deze ratio te worden beantwoord." Rb Groningen (vzr), 19 juni 2009, LJN BJ0717.



## PUBLIEKE SOFTWARE OPEN TENZIJ

non-discriminatie (de selectiecriteria moeten objectief zijn). Wanneer je producten inkoop en dit product is software in privaat eigendom, dan staat een gelijke kans voor een aanbieder in praktische zin gelijk aan een gelijke kans voor een product. Dat is simpelweg omdat alleen die aanbieder een bepaald product kan leveren. Bij software onder een publieke licentie is dit anders. Dat betekent niet dat een product onder een open source licentie niet geselecteerd *kan* worden via een aanbestedingsprocedure, al dan niet in vergelijking met propriëtaire software. Dat kan. Maar dit is een keuze. Bij software onder een publieke licentie vallen leverancier en product niet samen.

- *Overheden kunnen vooraf kiezen voor een bestaand product dat beschikbaar is onder een open source licentie en daarbij dienstverlening aanbesteden.*
- *Overheden kunnen gebruiksrechten tot gunningscriterium maken in een aanbesteding en daarmee een voorkeur uitdrukken voor ruime gebruiksrechten.*
- *Overheden kunnen gebruiksrechten als knock-out criterium gebruiken in een aanbesteding<sup>40</sup>.*

Als een koper wil kiezen voor een bestaand open source product is het van belang onderscheid te maken tussen de bestaande software enerzijds en eventuele uitbreidingen. De koper besteedt dan de uitbreiding aan. Ook dit is een dienst. Daarbij kan de koper ervoor kiezen om uren in te kopen of om de *uitbreiding* in te kopen.

Uiteraard gaat dit ook op voor de ontwikkeling van software - al dan niet onder een publieke licentie - die een koper niet reeds op een bestaande basis wil ontwikkelen.

Als een koper niet vooraf wil kiezen voor open source als randvoorwaarde en niet vooraf een keuze wil maken voor een bepaalde bestaande open source architectuur, maar in

<sup>40</sup> Het uitgangspunt is dat hier een *dienst* wordt ingekocht, geen levering. In de discussie over Saas versus on-premise [overwoog de commissie van aanbestedingsexperts](#): “Uitgangspunt is dat een aanbestedende dienst in beginsel vrij is te bepalen wat hij wil inkopen. (...) Bovendien is inherent aan een eis dat ondernemingen die daaraan niet kunnen voldoen, worden uitgesloten van deelname aan de aanbestedingsprocedure. (...) Dat een aanbestedende dienst een eis stelt waar een of meer ondernemingen niet aan kunnen voldoen, betekent derhalve nog niet per definitie dat de aanbestedende dienst in strijd handelt met het beginsel van gelijke behandeling.” Belangrijker is dat de aanbestedende organisatie zorgt dat de eis helder, transparant en objectief is. Een onderbouwing kan worden gevonden in de wens om de code te gebruiken onder bijvoorbeeld een EUPL licentie en het belang om libraries binnen de EUPL te kunnen laten vallen. In Italië heeft de regio Del Piemonte in 2009 een [wet](#) aangenomen waarin zij eigendomsrechten specificieert. Het Italiaanse constitutionele hof heeft zich hierover gebogen (Corte costituzionale della Repubblica Italia, [Uitspraak 122/2010, 22 maart 2010](#)) en stelt dat een juridische leveringsvereiste niet samenhangt met een product of merk en daarmee ook geen inbreuk is op de vrije mededinging.

## PUBLIEKE SOFTWARE OPEN TENZIJ

plaats daarvan een open source oplossing wil selecteren in vergelijking met gesloten oplossingen, dan kan de licentie ook als *selectie criterium* worden gebruikt<sup>41</sup>. Er is dan sprake van een voorkeur voor open source, die meespeelt in de weging. Uiteraard moeten de gunningscriteria duidelijk en objectief zijn, maar een ruimer gebruiksrecht kan een hoger aantal punten worden toegekend dan een beperkter gebruiksrecht. Het gaat dan in elk geval om de vier rechten die publieke licenties kenmerken:

- de gebruiker mag de software zonder beperkingen gebruiken
- de gebruiker mag de broncode inzien
- de gebruiker mag de broncode bewerken
- de gebruiker mag de broncode vrij distribueren.

Een meer uitgebreide publicatie hierover is hier te vinden: [Open voorkeur in een ICT aanbesteding](#)

Tenslotte is het ook nog denkbaar om gebruiksrechten als knock-out criterium te gebruiken in een aanbestedingsprocedure. Dit is geen nuttig instrument om een bestaande open source oplossing te selecteren. Die hoeft immers niet aanbesteed te worden. Als dat het doel is, kan beter een advies of onderzoek worden ingekocht. Dit kan wel heel nuttig zijn bij het inkopen van uitbreidingen of onderhoud.

Het uitgangspunt is dat een aanbestedende organisatie in beginsel vrij is te bepalen wat hij wil inkopen<sup>42</sup>. Wanneer dit een dienst is, zoals het schrijven van code waarmee bestaande software wordt uitgebreid of gewijzigd, kan een organisatie daarop het eigendom eisen. Inkoopvoorwaarden als [ARVODI](#) en [ARBIT](#) stellen zo'n eis. Volgens dezelfde logica kunnen andere rechten geëist worden. In feite is het volledige intellectueel eigendom op te delen in de vier rechten die een open source licentie definieert. De organisatie kan deze vier aspecten van eigendom dan ook eisen voor zichzelf, of voor het publiek waaronder zichzelf. Wat het ook is dat een organisatie eist, tijdens een procedure moet een organisatie redelijke alternatieven overwegen, net zoals een organisatie ook redelijke alternatieven moet overwegen bij de inkoopvoorwaarden ARVODI en ARBIT. Het is daarom ook van belang vooraf te bedenken waarom je als organisatie zo'n eis stelt. De onderbouwing is gelegen in het nut daarvan.

Dat nut kan gezocht worden in alle motieven uit [hoofdstuk 2](#). Daarnaast kan een organisatie dit onderbouwen met een concrete open source strategie voor de organisatie of een architectuur. Daarbij is het in ieder geval belangrijk dat de rechten op (alle) softwarecomponenten met elkaar verenigbaar (compatibel) zijn. Wanneer dit niet het geval is, kan de organisatie de software niet gezamenlijk gebruiken. De organisatie kan bijvoorbeeld gekozen hebben voor de EUPL, waardoor het voor de organisatie van belang is dat alle componenten daarmee verenigbaar zijn<sup>43</sup>.

<sup>41</sup>Paapst, 2009:10-11

<sup>42</sup>Zie bijvoorbeeld: <https://www.commissievanaanbestedingsexperts.nl/advies/advies-475-onduidelijkheid-over-aanbieden-van-saas-oplossing-naast-premise-oplossing>



## PUBLIEKE SOFTWARE OPEN TENZIJ

*Het eisen van publieke gebruiksrechten als knock-out criterium heeft twee voordelen:*

- *Het maakt alles eenvoudig, iedereen kan de software zonder hindernissen delen.*
- *Twijfels over eigendom van de code bij het zelf openbaar maken zijn of zorgvuldige administratie daarvan, zijn geen issue meer als dit een duidelijke lijn is.*

*De praktische vormgeving van zo'n beleidslijn is wel belangrijk, omdat alleen rechthebbenden een licentie-overeenkomst kunnen afdwingen.<sup>44</sup>*

### 4.2. Vragen over eigenaarschap en het beschikbaar stellen van code en principiële en juridische vragen rond mededinging

*Het Ministerie van Binnenlandse Zaken en Koninkrijksrelaties heeft een zeer leesbare handleiding laten schrijven over de juridische aspecten van het ter beschikking (laten) stellen van overheidssoftware als open source software:*

*[Delen en hergebruik van overheidssoftware](#)*

Naast deze handleiding is er [een paper dat nader ingaat op de kernvraag rond mededinging](#). Het kan voorkomen dat het beschikbaar stellen van overheidssoftware een economische activiteit lijkt én geen voorgeschreven overheidstaak van de organisatie. Dan kan de eis om kosten door te berekenen van toepassing lijken en dit botst met het kosteloos ter beschikking stellen van de broncode. In de handleiding worden hiervoor al een aantal uitzonderingsgronden beschreven. Het paper voert hiervoor nog andere gronden aan. Tenslotte is een expliciete uitzondering in Wet markt en overheid in voorbereiding om hierover alle onzekerheid weg te nemen.

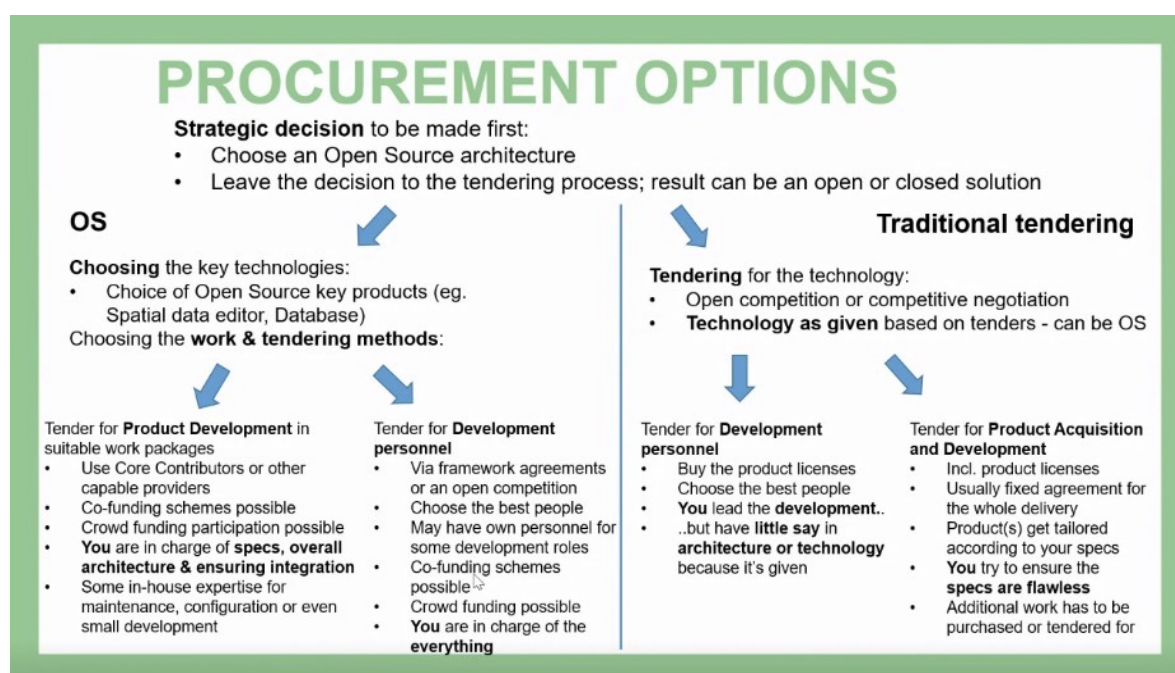
43 Er zijn uiteraard nog veel meer redenen te bedenken - belangen van de organisatie - zoals de eenduidigheid en algemene bekendheid voor medewerkers.

44 Eigendom niet hetzelfde als een licentie. Eigendom wordt in de vorm van auteursrecht gegund via de wet, terwijl een licentie specifieke rechten contractueel overdraagt. Het is relevant te weten bij welke partij het auteursrecht ligt om een licentie-overeenkomst te kunnen afdwingen. Met andere woorden: hoe je dit praktisch vormgeeft doet ertoe. Dat kan door de leverancier zelf de code onder een specifieke licentie te laten publiceren. Dat kan ook door het eigendom onder te brengen in een rechtspersoon waarin je ook een specifieke governance voor de software regelt.



## 5. Strategische vragen

De strategische vragen over aanbesteden van betrokkenen staan in een breder kader van strategische beslissingen, dat door het Finse Kadaster als in onderstaande afbeelding werd geschetst:



## Aanbestedingsopties volgens het Finse Kadaster

Er zijn gevallen waarin een organisatie om pragmatische redenen zal moeten kiezen voor de rechter route. Een open source oplossing is niet altijd reëel. Er kan een situatie zijn waarin er vooral of alleen standaard producten bestaan die onder gesloten licenties beschikbaar zijn. Er kunnen situaties zijn waarin pakketten vragen om allerlei vormen van certificering. En er kunnen situaties zijn waarin een pakket heel omvangrijk is, terwijl je zelf een kleine organisatie bent die individueel aanbesteedt. En dit kan uiteraard ook allemaal tegelijkertijd gelden. Bepaalde primaire processen van de gemeente, zoals die rond de Basisregistratie personen kunnen daarvoor als voorbeeld dienen, al denkt de Vereniging van Nederlandse gemeenten na over alternatieven in de toekomst binnen het project Common Ground en al wil één van de leveranciers zijn software onder een open source licentie uitbrengen.

## PUBLIEKE SOFTWARE OPEN TENZIJ

*[Common Ground](#) beoogt via een goede en langdurige samenwerking tussen gemeenten de informatievoorziening van gemeenten te verbeteren door te werken aan een andere informatie-architectuur. Open source is één van de realisatieprincipes binnen een bredere visie.*

De linkerroute aan de rechterkant is een veelgekozen route, omdat het specificeren - de kern van aanbesteden - in grote projecten vaak als zeer moeilijk of onmogelijk wordt gezien. Werken met ingehuurd personeel maakt dan meer iteratieve ontwikkeling mogelijk. De invloed op de specifieke implementatie is dan groter en makkelijker bij te stellen. De keuze in architectuur of technologie is dan doorgaans al gegeven via het aanbestedingsproces.

Bij de linkerroute kiest de koper vooraf een open source architectuur. Dat wil zeggen dat de koper kiest voor een basis van concrete bestaande applicaties en componenten. Dit kunnen slechts bouwstenen zijn of een reeds vrij compleet framework dat alleen wordt uitgebreid. Ook hier kan gekozen worden voor de inhuur van personeel. De koper managet dan het gehele proces, wanneer de koper de oplossing grotendeels zelf bouwt of wanneer er geen bedrijf (bekend) is dat deze dienst levert bij een bestaand framework. De meest linker route wordt vaak als ideaal gezien, echter tegen het aanbestedingsproces zien mensen op, net als tegen het opdelen van een project in werkpakketten. Dit zou anders zijn als de organisatie al bekend was met een framework en wanneer een groep bedrijven bekend was rond dit framework. Bij de start van een project is dit doorgaans niet het geval. Een oplossing voor dit probleem vraagt een meer beleidsmatige strategische benadering op de lange termijn, waartoe [het volgende hoofdstuk](#) een aanzet geeft.

### 5.1. Concrete strategieën

De vraag naar concrete strategieën is voor betrokkenen een vraag naar:

- zicht op “geschikte” bedrijven, die ook actief benaderd kunnen worden
- hoe de weg loopt van software maken naar betrokkenheid van meerdere partijen
- bekendheid van bestaande oplossingen en ondersteuning daarvan
- hoe de weg loopt van software maken naar eenvoudige levering





## PUBLIEKE SOFTWARE OPEN TENZIJ

### **gov.uk (GDS)**

*In Groot-Brittannië is er een marktplaats van specialisten, die software ontwikkelen en integreren (naast een marktplaats van diensten die via de zogenaamde “g-cloud” van de Britse overheid kunnen worden afgenomen). Dienstverleners moeten zich houden aan een “service standard”, waarvan leveren onder een open source licentie een onderdeel is. Dienstverleners zijn op de marktplaats te vinden met hun specialismen en hun beoordelingen.*

De marktplaats van het Britse GDS<sup>45</sup> lost zeker één probleem op en dat is de zichtbaarheid van en voor bedrijven. Bovendien committeren bedrijven zich aan bepaalde eisen en kwaliteitsstandaarden. Dit heeft *niet* het karakter van een raamcontract; toetreding blijft - op specifieke momenten - mogelijk. De GDS biedt (nog) geen antwoord op de vraag hoe je meerdere leveranciers betreft bij een softwareproject.

[OS2 in Denemarken slaagde erin om meerdere leveranciers te betrekken bij verschillende toepassingen](#). Zij deden dit door actief bedrijven te benaderen om:

- als tweede organisatie bestaande software uit te breiden
- een audit te doen op code van andere organisaties
- software aan te bieden als een dienst

### **Foundation for Public Code<sup>46</sup>**

*In Nederland beoogt de [Foundation for Public Code](#) meerdere leveranciers te betrekken, die verschillende diensten leveren rond het OpenZaak systeem.*

De Amerikaanse Federale overheid vereist<sup>47</sup> dat ten minste 20% van alle maatwerksoftware onder een open source licentie wordt gepubliceerd. Op code.gov participeren 30 federale overheden met de bedoeling de projecten vindbaar en bruikbaar te maken, ook voor niet-Federale organisaties. Er zijn meer dan zesduizend software projecten te vinden die in gebruik zijn bij organisaties van het Federale ministerie van Financiën tot de nationale archieven inclusief een indicatie van de kwaliteit van de code. Ook is er een lijst te vinden van taken die gedaan moeten worden, zoals fouten die verholpen moeten worden en uitbreidingen die gewent zijn.

Hoewel in deze strategieën belangrijke elementen zitten lijkt vooral ook de combinatie nodig van betrokkenheid van bredere groepen dienstverleners, breder gebruik, meer generieke frameworks en laagdrempelig aanbod. Dit zijn elementen die elkaar kunnen

45 Government Digital Service <https://www.gov.uk>

46 De Foundation for Public Code helpt organisaties gezamenlijk hun software te ontwikkelen en onderhouden.

47 Sinds 11 maart 2016 <https://obamawhitehouse.archives.gov/blog/2016/08/08/peoples-code>



## PUBLIEKE SOFTWARE OPEN TENZIJ

versterken. Gebruik in meerdere contexten leidt tot een meer abstracte en configureerbare kern en meer specifieke toevoegingen. Dit maakt software flexibeler en veelzijdiger. Laagdrempelig aanbod is aanbod dat op een makkelijke manier kan worden afgenomen. Het aanbieden van de software als een dienst maakt dit makkelijk en dat stimuleert het gebruik. Hoe meer helder is hoe de software aangeboden kan worden en hoe hierbij diensten geleverd kunnen worden, hoe breder dit interessant is voor leveranciers.

### **Laagdrempelig aanbod**

*De VNG wil "Signalen" - een toepassing om overlast te rapporteren voor burgers - beschikbaar maken met een druk op de knop vanuit de Common Ground appstore.*

Tegelijk brengt dit ook andere vragen met zich mee. Wanneer er niet één eigenaar is, is ook niet evident wie de "standaard distributie" uitbrengt. Binnen bestaande open source communities zijn er doorgaans maar enkele mensen die daarover het laatste woord hebben. Wanneer er meerdere bedrijven rond een softwareproduct zijn gevormd is daarmee niet als vanzelf een community geschapen of een project leader gevonden. En een eenduidige distributie, met goed versiebeheer en tests en waarin alle correcties worden opgenomen is zeer wenselijk voor alle betrokkenen.

### **Governance**

*[The Commons Conservancy](#) is een ideële organisatie die de governance van technologische projecten in het publieke belang faciliteert. De [Foundation for Public Code](#) heeft [een standaard online staan](#) die een leidraad kan vormen voor het beleid dat nodig is voor hergebruik van publieke code.*



## 6. Naar een andere manier van denken

*Overheden zouden niet gehinderd moeten zijn in het uitvoeren van democratische besluiten, omdat hun middelen onder controle staan van private partijen. Het vinden en behouden van onafhankelijkheid vraagt om consequent en strategisch beleid.*

### 6.1. Denken in overdraagbare taken

In alle voorbeelden die in dit onderzoek naar voren kwamen lag de nadruk steeds op het aanbesteden van open source software in plaats van op datgene wat open source software mogelijk maakt: denken in overdraagbare taken. We associëren samenwerking met open source software, omdat er geen (juridische) drempels zijn om de code over te dragen. Leveranciersonafhankelijkheid is vooral lastig te bereiken, omdat er een juridische drempel is in de vorm van privaat eigendom die overdracht in de weg staat. Open source licenties halen die drempel weg en scheppen daarmee een belangrijke voorwaarde, maar nog niet voldoende voorwaarde.

Tegenover het delegeren van verantwoordelijkheden via een juridische weg - dat wil zeggen: inkopen en alle verantwoordelijkheid elders leggen - een andere manier van denken zetten. Wat moeten we organiseren om die verantwoordelijkheid waar te maken? De invalshoek is dan niet een juridische, maar een praktische. Dat neemt niet weg dat er nog altijd behoefte is aan delegeren, aan het “wegzetten” van verantwoordelijkheden. Vandaar dat ik dit karakteriseer als: denken in overdraagbare taken. Wat een organisatie die onafhankelijk wil blijven wil inkopen is een overdraagbare taak. De verantwoordelijkheid van de leverancier is niet het zetten van een handtekening en het accepteren van de verantwoordelijkheid. De verantwoordelijkheid is het opstellen en bijstellen van een navolgbare en herhaalbare procedure en het zichtbaar maken van wat hij doet om het resultaat te bereiken. Daarbij zijn zowel open source software als aanbesteden middelen om het doel van onafhankelijkheid te bereiken.

#### ***Denken in ontzorgen versus denken in overdraagbare taken***

*Je zou in essentie twee manieren van denken over de inkoop van ICT tegenover elkaar kunnen zetten. In het ene geval vraagt de koper om een “totaaloplossing”, waarbij hij de verantwoordelijkheid maximaal delegeert aan de leverancier om daarmee zoveel mogelijk “ontzorgd” te worden. In het andere geval ziet de koper de ICT als onderdeel van de eigen infrastructuur en koopt hij overdraagbare taken in die nodig zijn voor de gebruiks- en kwaliteitseisen die hij daaraan stelt. Waarbij tot de kwaliteitseisen ook behoort welke eisen de rechtsstaat stelt aan de gebruiksrechten.*



## PUBLIEKE SOFTWARE OPEN TENZIJ

Overdraagbaarheid is hier het kernbegrip. De vrijheden die publieke licenties bieden zijn een voorwaarde om langs deze lijn te kunnen denken<sup>48</sup>. De rechten op de software scheppen dan geen juridische afhankelijkheid, waardoor bestaande leveranciers ook geen beperkingen kunnen opleggen aan de inhuur van onderhoud of andere dienstverlening. Dit maakt denken in overdraagbare taken mogelijk.

Deze manier van denken is ook niet alleen van toepassing op de software zelf, maar ook op alle taken daaromheen. Denk daarbij bijvoorbeeld aan een “service level agreement”. Klassiek definieert een zogenaamd “service level agreement” een resultaat, zoals de beschikbaarheid van een dienst, met boetes als regulerend mechanisme. Een alternatief kan zijn om te vragen naar het zichtbaar maken van de activiteiten die de leverancier onderneemt om dit resultaat zeker te stellen en het beschikbaar stellen van de procedure die hij daarvoor instelt en bijstelt. Een onderdeel daarvan kan het maken van een backup zijn, maar ook het regelmatig oefenen op het terugzetten van de backup en zelfs het vanaf nul herinstalleren van de dienst. Zekerheden zijn dan niet (alleen) juridisch van aard, maar (ook) praktisch en tevens overdraagbaar. Deze overdraagbaarheid versterkt bovendien de overdraagbaarheid van andere dienstverlening, want hiermee verkrijgt de koper tevens een beproefd migratiescenario. Zowel de kosten als de risico's van exit en migratie gaan omlaag. Deze dienstverlening kan geheel los staan van het eigenaarschap van de hardware en - indien open source - de software en ook los staan van alle andere dienstverlening. Deze dienst kan ook gelijktijdig, bijvoorbeeld afwisselend, door meerdere leveranciers worden uitgevoerd. Dan is ook de overdracht van de procedure continu beproefd.

*Het inkopen van software as a service of een gespecificeerd product lijkt een overdraagbare taak. In de praktijk blijkt is dit doorgaans niet zo omdat essentiële ingrediënten eigendom zijn van een private partij en essentiële kennis niet praktisch overdraagbaar is. Met het wisselen van een leverancier, wisselt de koper vaak ook van hardware, software, datacenter, databases en meer. Wanneer dit niet direct op een overdraagbare manier is georganiseerd, leidt dit vaak tot moeilijke en dure migratietrajecten. Uit beperkte overdraagbaarheid blijkt de afhankelijkheid van organisaties.*

### 6.2. Ruimte voor experimenteren

In [hoofdstuk 5](#) hebben we gezien dat de Aanbestedingswet veel ruimte biedt. Die ruimte maakt het mogelijk al vanaf de start te denken in overdraagbaarheid, overdraagbare taken

<sup>48</sup> In theorie zou dit ook kunnen wanneer de software volledig in eigen bezit is. Dit is echter theorie. De hoeveelheid ontwikkeltijd en -kosten die dit met zich zou meebrengen is te omvangrijk zelfs voor grote overheidsorganisaties. Het is niet reëel om databases, programmeertalen, besturingssystemen, front-end libraries en alles wat er voor nodig is om tot bruikbare software te komen in eigen huis te ontwikkelen. Uiteraard is het ook mogelijk om door te bouwen op bestaande software in privaat bezit, maar dan staat die basis onder private controle en is die niet overdraagbaar.

## PUBLIEKE SOFTWARE OPEN TENZIJ

en in termen van een meer continu proces van ontwikkeling en onderhoud. Dit geldt voor de lagen die onder ontwikkeling en beheer van software liggen - het datacentrum en de hardware, de virtualisatie van machines en opslag en de omgeving en het systeembeheer - maar dit geldt zeker ook voor de software zelf. Hieronder noem ik enkele mogelijkheden waarin techniek en inkoop samenkomen en wat een voorbeeld kan zijn van een strategie om te komen tot een situatie van onafhankelijkheid.

### 1. Werken naar een framework

Een framework is een verzameling code die steeds breder inzetbaar moet worden en waarmee de organisatie steeds meer verschillende toepassingen kan realiseren. De organisatie zelf én de dienstverleners kunnen hiermee vertrouwd worden. De organisatie kan het framework en de toepassingen in praktische zin beschouwen als eigendom en een eigen infrastructuur, al is het publieke code.

Niet alles hoeft te behoren tot hét of één platform. De keuze om (specifieke) data te ontsluiten via API's maakt dat dit niet tot het platform hoeft te behoren<sup>49</sup>. Zo zijn er meer manieren om onderdelen los te weken uit het platform, zodat ze onafhankelijk ontwikkeld kunnen worden<sup>50</sup>. Echter daar waar de feitelijke ontwikkeling plaatsvindt is een keuze nodig voor een beperkte set aan ontwikkeltalen<sup>51</sup> en bestaande modules en/of toepassingen.

Het kenmerk van een software-framework is dat het de bouwstenen biedt om allerlei toepassingen relatief snel te ontwikkelen, omdat allerlei basisfunctionaliteit *in samenhang* beschikbaar is. Een impliciet framework is een set van modules met basisfunctionaliteit die in samenhang kunnen functioneren, zonder dat al deze modules voor elke toepassing ook geïnstalleerd hoeven zijn. Dit laatste houdt toepassingen licht.

### 2. Specificeren van de spelregels

Door al vanaf het eerste moment te werken naar of met een (impliciet) framework - en niet alleen een toepassing - kunnen al vanaf het eerste moment de voorwaarden ingebouwd worden om het framework én de toepassing(en) breed inzetbaar te maken.

- a. een strategie om het framework enerzijds en toevoegingen die het tot een toepassing maken anderzijds te scheiden
- b. een strategie om zowel het framework als toevoegingen te upgraden, blijvend en onafhankelijk van elkaar, maar op een uniforme wijze

<sup>49</sup> Een (open) interface maakt het mogelijk taken te verdelen en onafhankelijk van elkaar te ontwikkelen.

<sup>50</sup> De keuze om database-onafhankelijk te ontwikkelen, kan ook dat losweken uit het platform bijvoorbeeld.

<sup>51</sup> Er zijn talen die zelf ook volledig open source zijn, waardoor ook daar geen afhankelijkheden binnensluipen. Raku (Perl6) is een voorbeeld van een nieuwe state-of-the-art programmeertaal met een community die bekend is om de centrale beschikbaarheid van modules.



## PUBLIEKE SOFTWARE OPEN TENZIJ

Duidelijke specificatie van waar modules staan, waar toevoegingen staan, waar tests staan, hoe gedocumenteerd wordt en meer.

De spelregels zijn als de ongeschreven regels van een open source community. Door ze expliciet te maken kunnen ze ook een rol spelen bij de inkoop en in het overdraagbaar maken van code.

3. Het publiceren van de spelregels: zo werken wij  
Het specificeren van de werkwijze kan dienen als een voorovereenkomst voor dienstverleners. Iedereen kan diensten leveren, maar dit is de wijze waarop code wordt aangeleverd, aan de koper, maar ook aan andere leveranciers.
4. Het organiseren van dienstverleners rond het framework  
Door al vanaf het begin duidelijk omschreven modules door verschillende leveranciers te laten leveren, raken meerdere leveranciers bekend met het framework en de werkwijze.

Door al vanaf het begin audits door andere dienstverleners te laten uitvoeren, wordt al vanaf het begin de documentatie getest. Het is ook mogelijk om door een andere dienstverlener de integratie te laten doen of de installatie.

Door toepassingen zoveel mogelijk als micro-services op te zetten kunnen al in een vroeg stadium meerdere dienstverleners worden verleid om de software als een dienst aan te bieden. Dit kan ook in een eigen virtuele cloud.

5. Het organiseren van gebruik  
Gebruik, ook door andere organisaties, is ook in het eigenbelang. Hoe breder het gebruik, hoe breder ook het aanbod kan worden van dienstverleners die zich verdiepen in het platform. Bovendien is dit goed voor de veiligheid. Hoe meer mensen de code moeten begrijpen en dus naar de code moeten kijken, hoe beter.
6. Het organiseren van de governance  
Het is van belang om een duidelijke distributie te publiceren<sup>52</sup>. Het moet duidelijk zijn wat de "standaard" distributie is en wat de versie daarvan is. Alleen dan kunnen ook makkelijk specifieke uitbreidingen die het framework tot een toepassing maken goed in verband worden gebracht met een versie waarmee die getest zijn en werken. En alleen dan is helder welke correcties op de code in welke versie zijn verwerkt.

Niet alleen kopers, ook dienstverleners moeten een rol hebben in de governance. Zij zijn veel beter in staat om in te schatten welke elementen uit specifieke uitbreidingen in een meer abstracte vorm onderdeel kunnen worden van het

<sup>52</sup> Dat kunnen er ook meer zijn, en andersom is het is ook mogelijk om te volstaan met een specificatie, maar de beschikbaarheid van ten minste één duidelijke distributie is wel erg praktisch.



**KAFKA**  
BRIGADE

## PUBLIEKE SOFTWARE OPEN TENZIJ

framework. Het is ook van belang hiervoor financieel duidelijk de ruimte te scheppen. Wanneer een percentage van de ontwikkelkosten jaarlijks als onderhoud voor specifieke uitbreidingen beschikbaar is, wordt het al snel aantrekkelijk om zoveel mogelijk in de standaard distributie - op een centrale plek - onder te brengen.

### 7. Het organiseren van het organiseren

Het is een keuze om de kennis die nodig is voor de organisatie hiervan in eigen huis te halen of ook hiervoor een dienstverlener in te huren. Ook dit kan een roulerende taak zijn tussen dienstverleners.

De elementen hierboven zijn voorbeelden<sup>53</sup> van ingrediënten voor een meer beleidsmatige en strategische visie gericht op onafhankelijkheid van de organisatie, waarbij inkoop en informatie-architectuur samenkomen om dat te realiseren. Onafhankelijkheid is uiteindelijk een politiek-bestuurlijk belang, dat voorwaardelijk is voor alle andere motieven die in het [tweede hoofdstuk](#) naar voren kwamen.

<sup>53</sup> Dit kan uiteraard ook op andere manieren. Dit voorbeeld legt de focus op de organisatie waarin de software moet functioneren. Je kunt ook op een hoger niveau zo'n strategie formuleren. Echter in alle gevallen zal de software ook in een organisatie moeten functioneren en zal rond de eigen organisatie dienstverlening moeten worden georganiseerd. Daarom heb ik geen allianties o.i.d. als voorbeeld genomen, maar de organisatie zelf.



**KAFKA**  
BRIGADE

## PUBLIEKE SOFTWARE OPEN TENZIJ

### Credits

Dit playbook is geschreven door Arjan Widlak van Stichting Kafkabrigade. De inspiratie hiervoor was de “Unconference” *Aligning Public Tech* van Code for NL op 12 maart 2020 in Utrecht. Bij de totstandkoming is allereerst aan de hand van interviews gekeken welke problemen ambtenaren ervaren en welke vragen zij hebben. Daarna is verkend voor welk deel de antwoorden op die vragen bekend zijn, maar ook waar de ruimte ligt om een andere invulling te geven aan het inkopen van software. Tot mei 2021 heeft dit document opengestaan voor commentaar. Velen hebben daarvan gebruik gemaakt door een grotere of kleinere bijdrage te leveren via interviews, commentaar op de tekst en discussies via e-mail, LinkedIn en andere kanalen. Niet alle namen zijn bekend en niet iedereen wilde vermeld worden, maar onder meer deze mensen hebben bijdragen aan dit document:

Ad Gerrits (Gemeente Nijmegen), Arjan Vliem, Ati de Groot (ICTU), Bart Jeukendrup (Delta10), Bert Beentjes (Kadaster), Boris van Hoytema (Foundation for Public Code), Charlotte Heikendorf (OS2), Claus Mullie (Foundation for Public Code), Cora Wils (BZK), Daan Moerman (UBR|HIS), Dabbert Doornbosch (Gemeente Den Haag), Dirk-Willem van Gulik (VWS), Edo Plantinga, Edwin Boender (VNG Realisatie), Edwin Wisse (BZK), Eelco Hotting (Gemeente Haarlem), Ernout van de Waard (Worth), Frank Hooghiemstra (VenJ), Frans Ward (SURF), Hilbrand Bouwkamp, Hillary Buyne (BZK), Inge Wertwijn (Belastingdienst), Ivonne Jansen-Dings (Provincie Zuid Holland), Jani Kylmäaho (National Land Survey of Finland), Jan-Willem Arnold (KNMI), Johan Groenen (Code for NL), John van Dijk (VNG), Kees Koenen (Red Hat), Koos Steenbergen (BZK), Maarten Sprey (RVO), Maarten van Zaane (CJIB), Maarten Zeinstra, Marco Bijkerk (Belastingdienst), Marc van Andel (Kadaster), Marjan Schnetz (Ockham Group), Michiel Schok (SURF), Mike van Dijk, Martin Dias d’Ullois (Logius), Mathieu Paapst (Rijksuniversiteit Groningen), Matthijs Goense (SVB), Mihai Polak (BZK), Nico Rikken (Alliander/FSFE), Peter Reumerman (Procura), Rasmus Frey (OS2), Richard Lennartz (HIS), Rogier Spoor (SURF), Roland van de Boel (Waterschapshuis), Ruud Zwakenberg, Salim Hadri (Milvum), Sanne Wijnhorst (Waterschapshuis), Saskia Hoogwerf (DUO), Seth de Jong (Kadaster), Solaiman Boubkari (UWV), Tamas Erkelens (VWS), Tim de Groot (VWS), Tim Vos-Goedhart (Open State Foundation), Victor Sint Nicolaas, Wouter Welling (BZK)

### Licentie

Naamsvermelding 4.0 Internationaal

<https://creativecommons.org/licenses/by/4.0/legalcode.nl>



## Literatuur

Algemene Rekenkamer, 2012, Europees aanbesteden

(<https://www.rekenkamer.nl/publicaties/rapporten/2012/11/27/europees-aanbesteden>)

Bijlsma, M., De Bijl, P., Kocsis, V., 2009, Competition, innovation and intellectual property rights in software markets, CPB Netherlands Bureau for Economic Policy Analysis

(<https://www.cpb.nl/sites/default/files/publicaties/download/competition-innovation-and-intellectual-property-rights-software-markets.pdf>)

Europese Commissie, 2020, Open source software strategy 2020 - 2023 think open

([https://ec.europa.eu/info/departments/informatics/open-source-software-strategy\\_en](https://ec.europa.eu/info/departments/informatics/open-source-software-strategy_en))

Gartner, 2017, Onderzoek publiceren open source software, in opdracht van Ministerie van Binnenlandse Zaken en Koninkrijksrelaties

(<https://kennisopenbaarbestuur.nl/media/255195/onderzoek-open-source-software.pdf>)

Maurer, S., Scotchmer, S., 2006, Open source software: the new intellectual property paradigm, National bureau of economic research

(<https://www.nber.org/papers/w12148>)

Ministerie van BZK, 2020, Kamerbrief over vrijgeven broncode overheidssoftware

(<https://www.rijksoverheid.nl/documenten/kamerstukken/2020/04/17/kamerbrief-inzake-vrijgeven-broncode-overheidssoftware>)

Ministerie van BZK, 2020, Overwegingen bij Open tenzij en Aanpak open source

(<https://www.rijksoverheid.nl/documenten/publicaties/2020/04/17/overwegingen-bij-open-tenzij-en-aanpak-open-source>)

NOiV, Bijlage 1, Voortgang per actielijn (nog te vinden via archive.org)

OSOR, 2020, Open source software in public procurement, webinar

(<https://joinup.ec.europa.eu/collection/open-source-observatory-osor/event/webinar-oss-public-procurement-knowledge-sharing-success>)

Ososs, 2007, Het verwerven van open source software, BZK.

([https://www.pianoo.nl/sites/default/files/documents/documents/Verwerven\\_van\\_\\_open\\_source\\_software\\_-\\_tekstf-1\\_0.pdf](https://www.pianoo.nl/sites/default/files/documents/documents/Verwerven_van__open_source_software_-_tekstf-1_0.pdf))

Overing, I., 2016, Cloud exit-strategie: hoe zit dat juridisch?, ICTRecht

(<https://www.ictrecht.nl/blog/cloud-exit-strategie-hoe-zit-dat-juridisch>)

## PUBLIEKE SOFTWARE OPEN TENZIJ

Paapst, M., 2009, Open voorkeur in een ICT aanbesteding, Open source jaarboek 2008-2009:133-148) ([https://www.rug.nl/research/portal/nl/publications/open-voorkeur-in-een-ict-aanbesteding\(5181800f-7ea8-49c1-b711-3fe0cf7ae566\).html](https://www.rug.nl/research/portal/nl/publications/open-voorkeur-in-een-ict-aanbesteding(5181800f-7ea8-49c1-b711-3fe0cf7ae566).html))

Paapst, M., 2013, Barrières en doorwerking: Een onderzoek naar de invloed van het open source en open standaarden beleid op de Nederlandse aanbestedingspraktijk. PhD Rijksuniversiteit Groningen. ([https://www.rug.nl/research/portal/nl/publications/barrieres-en-doorwerking\(cb68c60a-dce2-4451-8b43-7c73429e8da8\).html](https://www.rug.nl/research/portal/nl/publications/barrieres-en-doorwerking(cb68c60a-dce2-4451-8b43-7c73429e8da8).html))

Paapst, M., Van der Wees, P., 2020, Hergebruik van overheidssoftware, Tijdschrift voor Staatssteun,2020(4), 182-187. [UDH:TvS/16443]. (<https://www.rug.nl/staff/m.h.paapst/Hergebruik.pdf>)

Thaens, M., IJzerloo, F., 2017, Kenniscentrum open source software, PBLQ in opdracht van Ministerie van Binnenlandse Zaken en Koninkrijksrelaties (<https://kennisopenbaarbestuur.nl/rapporten-publicaties/rapport-kenniscentrum-open-source-software/>)

Schmidt, A., Schnitzer, M., 2003, Public Subsidies for Open Source? Some Economic Policy Issues of the Software Market, SSRN Electronic Journal 16(2) (<https://dx.doi.org/10.2139/ssrn.319081>)

Schmidt, A., Van der Klaauw-Koops, F., Paapst, M., Handreiking Juridische Aspecten, Niov. (nog te vinden via archive.org)

Schryen, G., Rich, E., 2010, Increasing Software Security through Open Source or Closed Source Development? Empirics Suggest that We have Asked the Wrong Question, 43rd Hawaii International Conference on System Sciences, p. 1-10, (<https://doi.org/10.1109/HICSS.2010.228>)

Schryen, G., 2011, Is open source security a myth? Communications of the ACM (<https://doi.org/10.1145/1941487.1941516>)

Swanson, E., 1994, Information Systems Innovation among Organizations, Management Science, p. 1069-1092 (<https://www.jstor.org/stable/2661474>)

Van der Molen, F., 2020, De kunst van het leveranciersmanagement, iBestuur (<https://ibestuur.nl/magazine/de-kunst-van-het%E2%80%86leveranciersmanagement>)

Van Kooten, M.H.A. et all, 2016, Mogelijkheden om de afhankelijkheid van ICT-leveranciers te verminderen, maatregelen en exit-strategieën voor de rijksoverheid, Verdonck, Klooster & Associates, Berenschot, RAND Corporation



**KAFKA**  
BRIGADE

## PUBLIEKE SOFTWARE OPEN TENZIJ

(<https://www.pianoo.nl/sites/default/files/documents/documents/onderzoek-verminder-afhankelijkheid-van-ict-leveranciers-te-verminderen-feb2016.pdf>)

VNG, 2020, Groeipact Common Ground

(<https://commonground.nl/file/download/54477598/Common%20Ground%20groeipact%20versie%201.1%2029%20januari%202020.pdf>)

Widlak, A.C., 2021, Volwassen Digitale Overheid, BoomBestuurskunde Den Haag  
(verschijnt binnenkort)

Windrum, P., 1999, Unlocking a lock-in: towards a model of technological succession.  
Research memoranda (nr. 10), Maastricht: MERIT.

([https://www.researchgate.net/publication/4787602\\_Unlocking\\_a\\_lock-in\\_towards\\_a\\_model\\_of\\_technological\\_succession](https://www.researchgate.net/publication/4787602_Unlocking_a_lock-in_towards_a_model_of_technological_succession))