

—● Softwarebeheer in Linux met Flatpak ●—

Hans Lunsing

Met dank aan Isja Nederbragt voor suggesties met betrekking tot leesbaarheid en begrijpelijkheid

Flatpak is een nieuw systeem voor de verspreiding en het beheer van software. Het maakt niet uit in welke Linux-distributie je het toepast, en wordt daarom een universeel systeem genoemd. Met Flatpak worden enkele problemen van de gebruikelijke distributiespecifieke systemen opgelost.

Verspreiding van software

Software kan op diverse manieren worden verspreid. De meest directe methode is gewoon via een website van de maker van de software. Bij Linux is het echter gebruikelijk om de software te verspreiden via zgn. *repositories* ofwel softwaremagazijnen. Dat zijn plekken op het internet waar softwarepakketten worden verzameld en opgeslagen.

Van oudsher heeft vrijwel elke Linux-distributie zijn eigen *repositories*. De erin opgenomen software wordt door de Linux-distributie heel specifiek voor de verschillende versies van de distributie samengesteld en beheerd. Hij is dan ook gegarandeerd virus- en malwarevrij. De ervoor benodigde broncode wordt door de maker van de software aangeleverd. De Linux-distributie heeft een zgn. pakketbeheersysteem, waarmee software vanuit de *repositories* kan worden geïnstalleerd en zonodig bijgewerkt. Eenmaal geïnstalleerd kan de software ook weer via dat beheersysteem worden verwijderd.

De software in de *repositories* is door de Linux-distributie geselecteerd en bestaat vrijwel helemaal uit opensource-software. Daarnaast is er vaak nog enige gratis closedsource-software die voor de werking van een systeem nodig kan zijn, zoals drivers. Software die niet is geselecteerd zal door de makers van die software vooral via hun eigen website of via een ontwikkelsite als github of sourceforge¹ worden aangeboden. Zo'n ontwikkelsite biedt ontwikkelaars van opensource-software de mogelijkheid hun software te ontwikkelen, te plaatsen en te verspreiden. Aanbieders van commerciële closedsource-applicaties voor Linux zullen niet in een Linux-repository worden opgenomen en zijn dan ook aangewezen op andere middelen van verspreiding, zoals via de eigen website.

Aanbieders van niet-geselecteerde Linux-software, zowel opensource als closedsource, hebben echter een probleem. Alle software in een versie van een Linux-distributie is nauwkeurig op elkaar afgestemd. Een externe applicatie (toepassing ofwel programma) moet daar wel in passen. Dat zou kunnen betekenen dat voor elke ondersteunde versie van elke distributie een pakket met de applicatie zou moeten worden samengesteld. Dat is onwerkbaar. Aanbieders van opensource-software volstaan daarom nogal eens met het aanbieden van de broncode, die de gebruiker zelf maar tot een werkende applicatie moet compileren. Waar dat niet gebeurt, of waar dat niet kan, zoals bij closedsource-software, zal de applicatie maar voor enkele belangrijke distributies en versies daarvan ter beschikking worden gesteld, of wordt er maar helemaal van een Linux-versie afgezien.

Daarom zijn nu universele systemen voor de verspreiding en het beheer van software in opkomst. Ook deze systemen bieden software aan via één of meer *repositories*, maar die

software kan in elke Linux-distributie via een eigen pakketbeheersysteem worden geïnstalleerd en vervolgens gebruikt. De makers van de software behoeven maar één pakket met de applicatie samen te stellen en naar zo'n repository te uploaden om van een zo groot mogelijke verspreiding onder Linux-gebruikers verzekerd te zijn. Zo wordt het voor hen veel aantrekkelijker om hun applicatie niet alleen voor Windows en MacOS, maar ook voor Linux aan te bieden.

De belangrijkste universele systemen zijn Flatpak en Snap. Flatpak is volledig opensource en niet aan een leverancier gebonden. Snap is weliswaar deels opensource, maar is voor de verspreiding van software afhankelijk van Canonical, dat het systeem ontwikkelde en in beheer heeft. Dat is de belangrijkste reden dat steeds meer Linux-distributies Flatpak ondersteunen. Ook de desktopomgevingen KDE en Gnome ondersteunen Flatpak als geprefereerde en aanbevolen wijze van software distributie. Zij denken er over te gaan samenwerken bij de uitbouw van de Flathub-appstore². Het lijkt er dan ook op dat Flatpak voor wat betreft de distributie van software voor Linux een belangrijke rol zal gaan spelen. De Linux-distributies Ubuntu en zijn officiële varianten zoals Kubuntu, ondersteunen het eigen Snap in plaats van Flatpak. Ondersteuning van Flatpak kan echter wel worden toegevoegd³.

Programmabeheer

Vrijwel alle Linux-distributies kennen grafische applicaties om vanuit de *repositories* gemakkelijk software te installeren, vervolgens te beheren en eventueel te de-installeren. Deze kunnen door zowel de Linux-distributie als de desktopomgeving in die distributie worden aangeboden. De desktopomgevingen KDE en Gnome hebben een eigen applicatie die resp. 'Ontdekken' en 'Gnome Software' heet. Linux Mint heeft 'Programmabeheer', Ubuntu en officiële varianten, zoals Kubuntu en afgeleiden, zoals Zorin, hebben 'Software', openSUSE heeft 'YaST Software'. Linux-distributies met KDE en Gnome hebben soms meerdere applicaties voor programmabeheer, zowel die van KDE of Gnome als hun eigen applicatie. Voortaan zal ik al deze applicaties scharen onder de term 'programmabeheer'.

Programmabeheer is eenvoudig in het gebruik, maar verbergt heel wat complexiteit. Er liggen een of meer zgn. pakketbeheersystemen aan ten grondslag, die ten doel hebben de te installeren applicatie te voorzien van alle benodigde basis- en hulpssoftware, in het bijzonder softwarebibliotheken (*libraries*) die door de applicatie wordt gebruikt.

We kunnen twee soorten pakketbeheersystemen onderscheiden: *specifieke* en *universele*.

Specifieke pakketbeheersystemen

Specifieke pakketbeheersystemen werken met pakketten die specifiek voor een versie van een distributie zijn samengesteld. Specifieke systemen zijn veelal door een bepaalde Linux-distributie ontwikkeld, en worden door die distributie en ervan afgeleide distributies gebruikt. Er zijn echter ook dis-

tributies die een elders ontwikkeld systeem gebruiken. Een voorbeeld daarvan is SUSE Linux / openSUSE, dat het systeem van Red Hat gebruikt. De twee belangrijkste en meest gebruikte specifieke pakketbeheersystemen zijn:

- Het Debian systeem. De Debian-softwarepakketten hebben een naam met extensie `.deb`. Dit systeem wordt gebruikt door Debian en alle ervan afgeleide distributies, zoals Ubuntu en daarvan afgeleiden zoals Kubuntu en Linux Mint. De programma's voor pakketbeheer zijn `apt` (voor de CLI⁴) en `synaptic` (voor de GUI⁵).
- Het Red Hat systeem (RPM, ofwel Red Hat Package Manager). De RPM pakketten hebben een naam met extensie `.rpm`. Dit systeem wordt gebruikt door Red Hat Enterprise Linux (RHEL) en afgeleide distributies zoals Fedora, CentOS, AlmaLinux, en onafhankelijk daarvan ook door SUSE Linux Enterprise Desktop (SLED) en -Server (SLES) en afgeleide distributie openSUSE. In RHEL etc. is het programma voor pakketbeheer `yum`, opgevolgd door `dnf` (allebei voor de CLI). Er is een grafische schil genaamd `yumex`. In SLED etc. is het `zypper` (voor de cli) en `yast sw_single` (voor de GUI).

Voor applicaties benodigde basis- en hulpsoftware die ook door andere applicaties kan worden gebruikt, wordt apart verpakt. Het belangrijkste voorbeeld hiervan zijn gedeelde bibliotheken (*shared libraries*). Zo blijven de pakketten zo klein mogelijk, en wordt heel wat schijf- en geheugenruimte bespaard. Elk pakket bevat meta-informatie met onder andere een lijst van pakketten waarvan het pakket in kwestie afhankelijk is. Het pakketbeheersysteem zorgt ervoor dat bij installatie van een applicatie aan zijn afhankelijkheden wordt voldaan door de benodigde basis- en hulpsoftware ook te installeren, als die al niet geïnstalleerd is. Natuurlijk kan ook basis- en hulpsoftware afhankelijk zijn van andere basis- en hulpsoftware. Ook al is een verzameling pakketten specifiek voor een versie van een distributie, toch kunnen distributies die van een bepaalde distributie zijn afgeleid vaak ook de pakketten van de ouderdistributie gebruiken. Zo gebruikt Linux Mint allerlei pakketten van Ubuntu. Ook veel Debian-pakketten zullen bruikbaar zijn. Maar deze Debian-pakketten zijn onbruikbaar op RPM-systemen, en omgekeerd. En pakketten voor RHEL, Fedora, etc., zijn veelal niet bruikbaar in SLED en openSUSE, en omgekeerd. Dat komt omdat veel pakketten anders zijn ingedeeld en anders heten.

Wat zijn de voor- en nadelen van deze specifieke beheerssystemen? Voordelen zijn:

- 1) Dank zij het feit dat applicaties softwarebibliotheken delen wordt efficiënt met schijf- en geheugenruimte omgegaan. Bovendien blijven de pakketten daardoor relatief klein en zal bij de installatie van een applicatie alleen datgene nog moeten worden gedownload dat nog niet in het systeem aanwezig is.
- 2) Pakketten worden onderhouden en getest door de distributie. Daardoor zijn ze allemaal goed op elkaar afgestemd, en zullen ze geen virussen en malware bevatten. Let wel dat dit laatste niet per se hoeft te gelden voor pakketten van derden, zoals bijvoorbeeld de pakketten uit Personal Package Archives (ppa's) bij Ubuntu en afgeleiden.

Maar er zijn ook nadelen:

- 1) Zoals bij de verspreiding van software al aan de orde kwam, is het voor leveranciers van applicaties een probleem dat voor elke nog ondersteunde versie van elke distributie een pakket zou moeten worden samengesteld. In het bijzonder geldt dat voor commerciële leveranciers, omdat hun software niet via de opensource-repositories van de distributies kan worden aangeboden. Dat is zo veel werk dat ze, als ze al pakketten leveren, dat alleen doen voor op zijn hoogst enkele distributies, in het bijzonder die welke ook commercieel aan de weg timmeren zoals

Red Hat, openSUSE en Ubuntu. Dit probleem zou wel eens een belangrijke rol kunnen spelen bij het feit dat allerlei commerciële software, denk aan Adobe Photoshop, wel voor Windows en MacOS wordt geleverd, maar niet voor Linux. Het is gewoon te veel werk voor een relatief geringe opbrengst.

- 2) Het onderhouden en bijwerken van pakketten door een distributie is zoveel werk dat distributies er vaak voor kiezen om nieuwere versies van applicaties alleen te leveren in nieuwere versies van de distributie. Als gebruiker blijf je dan zitten met oudere versies van applicaties, tenzij je overstapt op de nieuwere versie van de distributie. Of je zo'n overstap zou willen maken alleen maar om van een bepaalde applicatie een nieuwere versie te kunnen gebruiken is de vraag. En je kunt dan een probleem hebben als je van een andere applicatie de oudere versie zou willen blijven gebruiken.
- 3) Dat alle pakketten ter wille van gedeeld gebruik van bibliotheken nauwkeurig op elkaar zijn afgestemd heeft ook een belangrijk nadeel. Als je van een applicatie een oudere of nieuwere versie zou willen installeren, kan het zijn dat die versie een oudere of nieuwere versie van gedeelde bibliotheken nodig heeft. Maar die biedt het systeem helaas niet. En mocht het bij installatie van zo'n afwijkende versie van een applicatie toch lukken om de benodigde gedeelde bibliotheken op te halen en te installeren, dan kunnen al geïnstalleerde applicaties worden geconfronteerd met de foute versie van een of meer gedeelde bibliotheken.

Ook kan het zo zijn dat je een applicatie gebruikt die niet door de distributie, maar door derden geleverd is. Dan is het mogelijk dat die applicatie niet meer werkt als je op de volgende versie van de distributie over gaat, omdat hij afhankelijk is van een bepaalde versie van een bibliotheek die in de nieuwe versie van de distributie niet meer beschikbaar is.

Dit probleem staat wel bekend onder de naam 'hel van afhankelijkheden' (*dependency hell*).
- 4) Elk pakket dient met rootrechten te worden geïnstalleerd. Dat betekent dat een pakket bij installatie van alles met je systeem kan doen. Zolang een pakket uit een vertrouwde bron komt, zoals de repositories van de distributie, is dat geen probleem, maar een pakket uit een onbekende bron zou malware kunnen installeren zonder dat je het in de gaten hebt.

Universele pakketbeheersystemen

Om aan de nadelen van specifieke beheerssystemen tegemoet te komen hebben in de loop van de tijd verscheidene initiatieven het licht gezien om een universeel pakketbeheersysteem te ontwikkelen. De volgens zo'n systeem samengestelde pakketten zou je in principe in elke versie van iedere distributie moeten kunnen installeren en gebruiken. Met pakketten van de bestaande specifieke beheerssystemen is dat, zoals we zagen, heel vaak niet mogelijk. Op het ogenblik hebben we de keus uit een aantal van deze systemen, waarvan de belangrijkste zijn:

Flatpak (www.flatpak.org) is ontwikkeld door een onafhankelijke gemeenschap van vrijwilligers en ondersteunende organisaties, en geheel opensource.

Een volgens dit systeem samengesteld pakket met een applicatie noemen we een flatpak. Meerdere repositories zijn mogelijk.

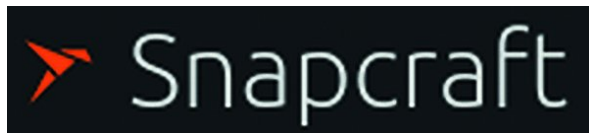
De centrale repository met bijna 2300 applicaties is Flathub (flathub.org/nl), maar onder meer de desktopomgevingen KDE en Gnome hebben ook eigen repositories. Vrijwel alle erin



Flatpak

aangeboden applicaties zijn grafisch, er zijn er maar enkele voor de terminal (opdrachtregel). Ongeveer 10% van de beschikbare applicaties zijn propriëitair⁶. Voorbeelden daarvan zijn Opera (web-browser), Microsoft Edge, Skype, Steam (platform voor games) en VuSCAN (scanner software). Er zijn verschillende distributies waarin de ondersteunende software van Flatpak standaard geïnstalleerd is, onder meer Fedora, Linux Mint en Zorin. Dat geldt niet voor Ubuntu en zijn officiële varianten zoals Kubuntu, want daarin is standaard Snap geïnstalleerd. Een enkele opdracht (flatpak update) is voldoende om alle flatpaks op een systeem bij te werken naar de nieuwste versie.

Snap (snapcraft.io) is ontwikkeld door Canonical (de onder-neming achter Ubuntu).



Het pakketformaat en de ondersteunende software zijn opensource, maar er is maar één centrale repository (de *Snap Store*: snapcraft.io) onder de hoede van Canonical, waarvan de broncode niet openbaar is. Een volgens dit systeem samengesteld pakket met een programma noemen we een snap. Snap was oorspronkelijk bedoeld voor cloud-, server- en IoT-tools en applicaties, maar ondersteunde later ook desktop-applicaties. Het totaal aantal applicaties is enkele duizenden (eind 2018 al ca. 4000), maar het aantal desktop-applicaties lijkt achter te blijven bij dat van Flatpak. Alleen Ubuntu en zijn officiële varianten hebben de ondersteunende Snap-software standaard geïnstalleerd. Snaps (Snap-applicaties) worden standaard automatisch bijgewerkt, maar gebruikers kunnen aangeven wanneer en hoe dit moet gebeuren, en het tegenhouden voor geselecteerde snaps.

AppImage (appimage.org) bestaat al sinds 2004 onder de naam *Klik*, werd vanaf 2011 *PortableLinuxApps* genoemd en vanaf 2013 *AppImage*.

Het is ontwikkeld door Simon Peter en is geheel opensource. De centrale repository is AppImages (appimage.github.io/apps) met circa 1400 applicaties.



Ondersteunende software is niet nodig. Een AppImage-pakket hoeft niet echt te worden geïnstalleerd, maar kan naar believen ergens worden geplaatst, bijvoorbeeld in de map `/usr/local/bin`, en uitvoerbaar worden gemaakt. AppImages zijn daarom gemakkelijk overdraagbaar (*portable*). Elke AppImage bevat alle hulpsoftware (met name bibliotheken) die nodig is om de applicatie te draaien en deelt geen hulpsoftware met andere AppImages. Desondanks nemen AppImages minder ruimte op de schijf in dan je zou denken omdat de software erin is gecomprimeerd. AppImages moeten door de gebruiker zelf aan het menu worden toegevoegd. Er is geen mechanisme om AppImages bij te werken. Dat kan alleen handmatig door de nieuwe versie van zo'n AppImage te downloaden en de oude er door te vervangen. Eigenlijk is AppImage dan ook geen echt pakket-beheersysteem. Het beheer wordt immers volledig aan de gebruiker overgelaten.

Archief voor installatie in map /opt. Sommige aanbieders van software leveren een archiefpakket (bijvoorbeeld `.tar.gz`), dat alle basis- en hulpsoftware bevat die voor de applicatie nodig zijn, ook al is die software mogelijk al elders op het systeem aanwezig. Zo'n pakket kan onder de naam van de applicatie in de map `/opt` worden uitgepakt, waarna aan de map `/usr/local/bin` een symbolische link naar het hoofdprogramma kan worden toegevoegd. Ook moet zo'n

applicatie door de gebruiker nog aan het menu worden toegevoegd. Voorbeelden van software die zo wordt aangeboden zijn Firefox, Seamonkey, Thunderbird. Dit soort software is dan ook voor vrijwel alle Linux-distributies geschikt. Het moge duidelijk zijn dat dit net als AppImage eigenlijk geen pakketbeheersysteem is.

Zoals we al zagen wordt Flatpak al door heel wat Linux distributies ondersteund. Snap blijft beperkt tot Ubuntu en zijn officiële varianten zoals Kubuntu. AppImage kan een rol spelen als leverancier van overdraagbare applicaties. Al deze systemen kunnen, indien gewenst, naast elkaar worden gebruikt.

In het vervolg van dit artikel zullen we ons richten op Flatpak. Snap en AppImage laten we voorshands buiten beschouwing.

Flatpak

Flatpak mag dan een universeel pakketbeheersysteem zijn, de mate waarin het door de vele verschillende distributies wordt ondersteund verschilt nogal. In Linux Mint vanaf versie 18.3 is Flatpak-ondersteuning volledig ingebouwd en kunnen de Flatpak-applicaties, net als de regulier verpakte applicaties, via Programmabeheer worden geïnstalleerd, bijgewerkt en verwijderd. Dat geldt ook voor Zorin, zij het dat daarin met het Gnome Software-beheerprogramma wordt gewerkt. Als voorbeelden geef ik verder Fedora, waarin alleen de Flathub-repository nog moet worden geïnstalleerd, MX Linux vanaf versie 18 waarin Flathub nog moet worden geactiveerd, en Manjaro waarin de ondersteuning nog moet worden geactiveerd. En zo zijn er nog verscheidene andere distributies waarin ondersteuning, al dan niet volledig geactiveerd, is ingebouwd. In Ubuntu en zijn officiële varianten, zoals Kubuntu, is standaard alleen Snap geïnstalleerd en wordt door Programmabeheer ondersteund. De webbrowsers Firefox is daarin sinds kort zelfs opgenomen als snap en niet meer als gewoon distributiepakket. Wat in een 36-tal Linux-distributies, waaronder ook Ubuntu en Kubuntu, voor ondersteuning en activering van Flatpak moet worden gedaan, blijkt uit de instructies op de Flathub-website (flathub.org/nl/setup).

Flatpaks hebben een aantal belangrijke voordelen ten opzichte van de normale distributiespecifieke pakketten:

- 1) Voor ontwikkelaars hebben flatpaks het grote voordeel dat ze hun applicatie maar één keer hoeven in te pakken en het pakket maar één keer hoeven te uploaden, naar Flathub. Flatpaks draaien immers op elke distributie waarin de Flatpak-hulpsoftware is geïnstalleerd. Dit was in feite het belangrijkste doel van de opzet van Flatpak.

- 2) Flatpaks zijn om twee redenen veiliger dan de reguliere pakketten. Ten eerste hoeven ze niet door root te worden geïnstalleerd, maar kan dat door gewone gebruikers worden gedaan. De applicatie krijgt bij installatie dus geen beheerderspermissies.

- 3) Ten tweede draaien de applicaties geïsoleerd van het reguliere systeem in een sandbox (zie verderop), waardoor ze alleen toegang hebben tot datgene wat ze nodig hebben en wat verder nog toegestaan wordt.

- 4) Gebruikers kunnen applicaties installeren die niet door hun Linux-distributie worden geleverd, of nieuwere versies van applicaties waarvan hun Linux-distributie alleen een oudere versie levert.

- 5) Geen problemen met afhankelijkheden. Als je een oude applicatie wilt installeren die een oude bibliotheek nodig heeft, kan dat in principe zonder andere applicaties of je systeem te hinderen.

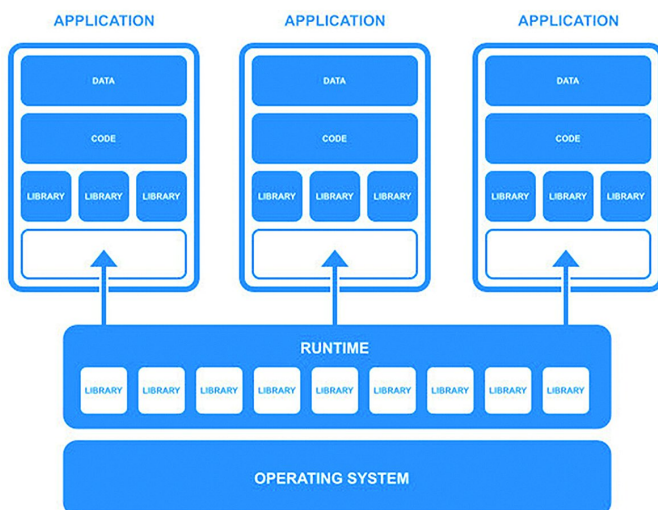
Flatpaks hebben echter ook enkele nadelen, zij het dat deze m.i. van minder belang zijn:

- 1) Het Flatpak-systeem is gescheiden van het reguliere systeem van de distributie, en dat betekent dat de basis- en hulpsoftware die de applicaties nodig hebben en kunnen delen (de zgn. runtimes, zie verderop) ook binnen het Flatpak-systeem moeten worden geïnstalleerd. Dat heeft tot gevolg dat gebruik van flatpaks tot extra ruimtegebruik leidt, hoewel dat per applicatie wel meevalt als voor meerdere applicaties steeds dezelfde runtimes worden gebruikt. Als er meerdere runtimes en daarvan meerdere versies nodig zijn kan het extra ruimtegebruik vrij hoog zijn.⁷
- 2) Flatpaks zijn er vrijwel alleen voor grafische programma's, niet voor tekstgeïntendeerde programma's voor de opdrachtregel, zoals tools voor een server. Voor de meeste gewone gebruikers zal dit echter geen probleem zijn.
- 3) Flatpaks volgen mogelijk niet je eigen thema voor het uiterlijk van het bureaublad. Of veel gebruikers dit een belangrijk nadeel vinden staat te bezien. Wel is het mogelijk om thema's vanuit Flathub te installeren en applicaties te dwingen zo'n thema te volgen.

Eerst komt nu een aantal basisbegrippen aan de orde, waardoor duidelijker zal worden hoe Flatpak werkt. Het gaat om *runtimes* met basis- en hulpsoftware, *sandboxes* voor een van het host⁸-systeem geïsoleerde omgeving, *poorten* voor uitwisseling van data, enz., met het host-systeem, en *remotes* als opslagplaatsen van software (repositories). Daarna gaan we in op het beheer en gebruik van Flatpak.

Runtimes

Vrijwel elke applicatie is afhankelijk van basis- en hulpsoftware, in het bijzonder bibliotheken. Bij Flatpak is deze software ondergebracht in zgn. '*runtimes*'. Elke applicatie is gebouwd op basis van een runtime, die in zijn afhankelijkheden⁹ voorziet en geïnstalleerd moet zijn om de applicatie te kunnen draaien. Er kunnen meerdere runtimes, en zelfs meerdere versies van een runtime, op een host-systeem worden geïnstalleerd. Runtimes worden gedeeld tussen applicaties. Bij installatie van een applicatie zal Flatpak van de benodigde runtimes alleen de nog ontbrekende mee-installeren. Dat betekent dat de eerste installatie van een Flatpak-applicatie weliswaar meer ruimte zal vergen dan wanneer de applicatie op de standaard distributiespecifieke wijze zou worden geïnstalleerd, maar Flatpak-applicaties die daarna worden geïnstalleerd en dezelfde runtimes gebruiken hebben daardoor geen extra ruimte nodig.



Werking flatpak

De belangrijkste runtimes zijn die van Freedesktop, Gnome en KDE. De Freedesktop runtime levert essentiële bibli-

otheken en diensten zoals X11 en Wayland (het grafische systeem), GLib en Gtk (grafische bibliotheken), PulseAudio (geluidsserver) en D-Bus (voor communicatie tussen processen). Daarnaast zijn er uitbreidingen van deze runtimes, zoals vertalingen en codecs.

Het is mogelijk dat een applicatie basis- en hulpsoftware gebruikt die niet in zijn runtime is opgenomen. In dat geval kan deze software samen met de applicatie worden gebundeld. Dat geeft ontwikkelaars van applicaties flexibiliteit met betrekking tot de basis- en hulpsoftware die zij gebruiken. Zo kunnen bijvoorbeeld ook versies van deze software worden gebundeld die van de runtime afwijken.

Sandboxes

Elke applicatie wordt gedraaid in een van het systeem geïsoleerde omgeving, die de '*sandbox*' wordt genoemd.

Elke sandbox bevat een applicatie en zijn runtime. Standaard heeft een applicatie alleen toegang tot zijn sandbox. De maker van een applicatie zorgt er voor dat hij de nodige permissies heeft voor toegang tot hulpbronnen als geluid, netwerk en bestanden. De gebruiker kan permissies uitbreiden of inkrimpen. Met de applicatie *Flatseal* is dat heel gemakkelijk. Flatseal wordt aangeboden op Flathub en kan worden geïnstalleerd met de opdracht:



```
flatpak install flathub com.github.tchx84.Flatseal
```

maar ook vanuit Programmabeheer mits Flatpak-ondersteuning is ingebouwd.

Sommige elementen binnen een sandbox dienen toegankelijk te zijn voor het host-systeem. Deze worden '*exports*' genoemd omdat ze als het ware vanuit de sandbox worden geëxporteerd. Voorbeelden zijn het *.desktop* bestand¹⁰ en het pictogram van de applicatie.

Poorten

Via zgn. poorten (*portals*) kunnen applicaties vanuit de sandbox een wisselwerking aangaan met het *host*-systeem.

Zo krijgt de applicatie veilig toegang tot benodigde bestanden, gegevens en diensten zonder dat het nodig is de sandbox-permissies uit te breiden. Dat maakt het bijvoorbeeld mogelijk om bestanden te openen via een bestandskeuzediagram, of te printen.



Naamgeving

Namen van flatpaks, zowel applicaties als runtimes, hebben dezelfde structuur als de domeinnaam van de ontwikkelaar op het internet, maar dan omgekeerd. Ze heten 'Application ID's'. Zo'n ID begint dus met het hoogste domeinniveau (zoals 'nl', 'com' of 'org'), daarna op niveau twee het eigenlijke domein (zoals 'compusers'), en vervolgens op niveau drie de eigenlijke naam van de applicatie of runtime, eventueel gevolgd door nadere aanduidingen. Al deze niveaus worden van elkaar gescheiden door een punt. De naam van de applicatie of runtime begint meestal met een hoofdletter, maar verwarrend genoeg niet altijd.

Zo heet de browser Firefox *org.mozilla.firefox*, de browser Edge *com.microsoft.Edge*, de browser Chrome *com.goog-*

le. *Chrome*, en het fotobeheerprogramma *Digikam* org.kde.digikam.

De runtime voor KDE-applicaties heet *org.kde.Platform*, en die van GNOME *org.gnome.Platform*. Een derde belangrijke runtime is *org.freedesktop.Platform* voor opensource grafische applicaties in het algemeen. Runtimes kunnen extensies hebben, die op de vierde en hogere niveaus van de naam worden aangegeven, ook vaak beginnend met een hoofdletter. Zo is er in de meeste gevallen een extensie genaamd *Locale* met vertalingen van de runtime. Een voorbeeld is *org.kde.Platform.Locale*.

Remotes

Flatpak-applicaties en -runtimes worden verspreid via magazijnen (*repositories*) op het internet. Flatpak noemt ze 'remote repositories' of kortweg 'remotes', met de betekenis van 'opslagplaatsen op afstand'. De standaard algemene remote is Flathub, maar het is mogelijk toegang tot meerdere remotes te configureren. Zo hebben de desktopomgevingen GNOME en KDE voor hun applicaties eigen remotes. Eenmaal geconfigureerd kan de inhoud van een remote worden geïnspecteerd en doorzocht, en kunnen er applicaties en runtimes uit worden geïnstalleerd.

In Flathub, maar mogelijk ook in andere remotes, zijn in veel gevallen ook oudere versies van applicaties beschikbaar, maar installatie daarvan is minder eenvoudig en gaat via enkele bijzondere opdrachten in de terminal¹¹.

Wanneer een applicatie of runtime wordt bijgewerkt worden van de nieuwe versie alleen de verschillen met de geïnstalleerde versie gedownload en vervolgens toegepast. Dat maakt het proces van bijwerken heel efficiënt.

Flathub

Flathub is de centrale algemene opslagplaats van Flatpak-software.



Ontwikkelaars kunnen hun softwarepakketten hier uploaden, en gebruikers kunnen ze downloaden en installeren. Na installatie van het flatpak-beheerprogramma moet toegang tot Flathub worden geactiveerd door het toe te voegen aan de lijst van toegankelijke remotes. Dat gaat met de volgende opdracht:

```
flatpak remote-add --if-not-exists flathub \
  https://dl.flathub.org/repo/flathub.flatpakrepo
```

Het bestand *flathub.flatpakrepo* bevat een aantal relevante gegevens over de remote *flathub*, zoals de naam, de URL, de homepage en de GPG-sleutel.

Op de website van Flathub (flathub.org) heeft een deel van de aangeboden applicaties (ca 30%) een vinkje achter zijn naam staan. Dat houdt in dat de applicatie door de oorspronkelijke ontwikkelaar op Flathub is geplaatst, of door een derde die door die ontwikkelaar is goedgekeurd. De identiteit van de ontwikkelaar is door Flathub geverifieerd. Dat geeft meer zekerheid met betrekking tot de herkomst van de applicatie.

Andere remotes

De desktopomgevingen GNOME en KDE hebben eigen remotes voor de allernieuwste ontwikkelversies van hun applicaties. Ze kunnen worden toegevoegd met resp. de opdrachten:

```
flatpak remote-add --if-not-exists gnome-nightly \
  https://nightly.gnome.org/gnome-nightly.flatpakrepo
```

```
flatpak remote-add --if-not-exists kdeapps \
  https://distribute.kde.org/kdeapps.flatpakrepo
```

Deze remotes hebben niet zo'n fraaie grafische interface als Flathub. Een lijst van de door hen aangeboden applicaties wordt verkregen met de opdracht:

```
flatpak remote-ls --app {naam van remote}
```

en natuurlijk ook in het programmabeheer als Flatpak-ondersteuning is ingebouwd.

Let wel dat het kan gaan om ontwikkelversies die mogelijk nog niet stabiel zijn. De reguliere stabiele versies zijn te vinden op Flathub. De reguliere- en ontwikkelversie van een applicatie kunnen zonder probleem naast elkaar worden geïnstalleerd. Daarnaast kan ook de door de Linux-distributie geleverde versie van de applicatie nog zijn geïnstalleerd.

Beheer en gebruik van flatpaks

Flatpak-ondersteuning in programmabeheer

In **Linux Mint** is ondersteuning voor Flatpak ingebouwd. Dat houdt onder meer in dat het beheerprogramma *flatpak* is geïnstalleerd en dat Programmabeheer Flatpak ondersteunt. In Programmabeheer (de eigenlijke naam is *mintinstall*) kun je kiezen voor de categorie Flatpak, waarmee je heel gemakkelijk een grafisch overzicht van alle Flatpak-applicaties krijgt. En de Flatpak-applicaties maken natuurlijk ook deel uit van de overige functionele categorieën. Flathub is de enige remote die in Linux Mint is geactiveerd.

In de programmabeheer-applicaties Ontdekken van de desktopomgeving KDE en Gnome Software van dito Gnome kan Flatpak-ondersteuning worden verkregen door de installatie van een plug-in, nl. *plasma-discover-backend-flatpak* of *discover-backend-flatpak* in KDE en *gnome-software-plugin-flatpak* in Gnome. Het is overigens niet gezegd dat dit in elke Linux-distributie met KDE of Gnome werkt, omdat hun programmabeheerapplicatie soms vervangen wordt door de eigen applicatie van de distributie.

Installatie van een applicatie

Als programmabeheer Flatpak ondersteunt kan een Flatpak-applicatie vanuit programmabeheer worden geïnstalleerd, net zoals een door de distributie geleverde applicatie. Anders kan een applicatie ook vanaf de opdrachtregel in een terminal worden geïnstalleerd, en wel zo:

```
flatpak install [naam van de remote] {naam van de applicatie}
```

Het is niet nodig om ook de naam van de remote op te geven. Zonder die naam zoekt flatpak zelf wel uit waar de gewenste applicatie kan worden gevonden. Daarbij moet wel de volledige naam van de applicatie (ofwel de 'Application-ID') worden gegeven, dus met inbegrip van het domein van de ontwikkelaar. Om erachter te komen wat de Application-ID van een gewenste applicatie is kun je naar tekst zoeken die relevant is voor de applicatie, bijvoorbeeld zijn zuivere naam, zonder het domein:

```
flatpak search {tekst}
```

Flatpak toont dan alle gevonden applicaties met de Application-ID, en verder onder meer beschrijving, versie en remote. Een andere manier om er achter te komen is om een lijst van alle beschikbare applicaties op te vragen, en daarin de relevante eruit te lichten:

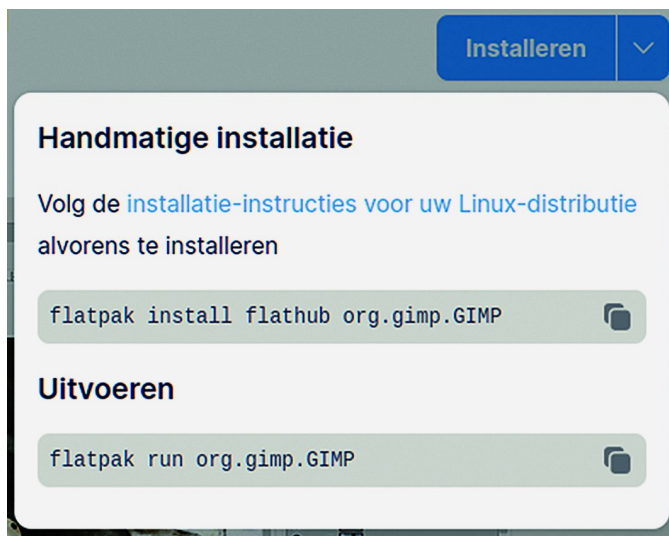
```
flatpak remote-ls | grep {tekst}
```

Dat geeft in principe dezelfde resultaten als het search-commando, zij het zonder beschrijving.

Je kunt de applicatie ook opzoeken op de website van FlatHub (mits hij daar beschikbaar is). Eenmaal gevonden zie je op de pagina van de applicatie een blauwe knop 'Installeren'.



Rechts naast die knop zie je een pijltje naar beneden. Als je daarop klikt toont FlatHub je welke flatpak-opdracht je moet geven om de applicatie te installeren en welke opdracht om de applicatie na installatie te draaien. Zie bijgaande afbeelding.



Installatie vanuit FlatHub

Je zou ook de blauwe knop 'Installeren' kunnen gebruiken om de applicatie te installeren, maar dat heeft wat voeten in de aarde. Onder die knop zit het bestand `[application id].flatpakref`, bijvoorbeeld `org.gimp.GIMP.flatpakref`. Zo'n flatpak-refbestand is een tekstbestand dat de nodige informatie bevat om de applicatie te kunnen downloaden en vervolgens te installeren. Internetbrowsers geven altijd de mogelijkheid het bestand te downloaden en ergens op te slaan. Daarna kan de installatie in de downloadmap vanaf de opdrachtregel worden uitgevoerd, bijvoorbeeld:

```
flatpak install org.gimp.GIMP.flatpakref
```

Uit veiligheidsoverwegingen geven internetbrowsers veelal niet de keuze tussen het openen en het opslaan van een bestand, maar bieden ze alleen opslaan als mogelijkheid. In Firefox is dat standaard ook zo, maar Firefox biedt de mogelijkheid om dat te veranderen. Om ook voor 'Openen' te kunnen kiezen moet je in 'Instellingen' onder 'Toepassingen' kiezen voor 'Vragen of bestanden geopend of opgeslagen moeten worden'. In Linux Mint kun je een flatpak-refbestand dan Openen met 'Software Manager' en in een Gnome-omgeving met 'software-installatie'. Wat er dan gebeurt is dat programmabeheer wordt geopend op de pagina van de te installeren Flatpak-applicatie, van waaruit de applicatie uiteindelijk kan worden geïnstalleerd. Installatie via een omweg dus. In een KDE-omgeving zou hetzelfde moeten kunnen, maar dat is me niet gelukt.

Samenvatting installatie

De gemakkelijkste manier om een Flatpak-applicatie te installeren is door dat te doen in programmabeheer, net als alle andere software. Via de browser direct uit FlatHub

installeren is eigenlijk een omweg via programmabeheer of via de opdrachtregel in een terminal. Niet aan te raden dus. Installatie via de opdrachtregel is snel en efficiënt, maar dat is voor de liefhebber.

Bijwerken en verwijderen

Als programmabeheer Flatpak ondersteunt, loopt de update van flatpaks gewoon mee met de update van distributiepakketten. Anders kan het ook heel eenvoudig via de opdrachtregel:

```
flatpak update [namen applicaties of runtimes]
```

Als geen namen van applicaties of runtimes worden genoemd worden alle applicaties en runtimes bijgewerkt.

Hetzelfde geldt voor verwijderen: dat kan via programmabeheer als dat Flatpak ondersteunt, en via de opdrachtregel:

```
flatpak uninstall {naam van applicatie}
```

Uitvoeren van een applicatie

Bij installatie van een applicatie wordt hij in het menu opgenomen. Van daaruit kan hij worden gestart en uitgevoerd. Het kan echter ook vanaf de opdrachtregel:

```
flatpak run {naam van applicatie} &
```

waarin de & er staat om de draaiende applicatie los te maken van de terminal, zodat de terminal niet wacht op het afsluiten van de applicatie.

Bijvoorbeeld:

```
flatpak run org.gimp.GIMP &
```

Het beheerprogramma flatpak

In het voorgaande hebben we al gezien hoe je remotes toevoegt met het beheerprogramma flatpak, maar er kan nog veel meer mee. Geef maar eens de opdracht:

```
flatpak --help
```

Flatpak heeft heel wat opdrachten met betrekking tot applicaties, bestandstoegang, permissies en remotes. Als je voor `--help` de naam van een opdracht geeft, krijg je meer helpinformatie over die opdracht, bijvoorbeeld:

```
flatpak install --help
```

Heel nuttige opdrachten die hierboven nog niet zijn genoemd zijn:

Lijst van geïnstalleerde applicaties en/of runtimes

```
flatpak list
```

Informatie over een geïnstalleerde applicatie of runtime

```
flatpak info {naam app of runtime}
```

Lijst van alle geconfigureerde remotes

```
flatpak remotes {naam app of remote}
```

Lijst van de inhoud (applicaties en runtimes) van een geconfigureerde remote

```
flatpak remote-ls {naam remote}
```

Informatie over een remote applicatie of runtime

```
flatpak remote-info {naam remote}
{naam app of runtime}
```

Tot besluit

Lange tijd was het voor aanbieders van software lastig en tijdrovend om ook voor Linux software te leveren. Debet daaraan waren de specifieke pakketbeheersystemen, die - hoe efficiënt ze verder ook zijn - ertoe leidden dat voor heel wat verschillende Linux-distributies en zelfs van de op-eenvolgende versies daarvan een pakket van de software zou moeten worden gemaakt. Een gevolg daarvan is dat nieuwere versies van software vaak alleen beschikbaar komen in nieuwere versies van de Linux-distributie.

Daarnaast zijn er mogelijk oudere versies van software die trager of zelfs niet verder wordt ontwikkeld, maar nog wel van nut zijn. Zulke software zal vaak oudere bibliotheken nodig hebben, die de Linux-distributie niet meer levert en er ook niet gemakkelijk in zijn te installeren. Universele pakketbeheersystemen beogen die problemen op te lossen. En daarvan lijkt Flatpak de geprefereerde wijze van verspreiding van Linux-software te worden. Steeds meer Linux-distributies ondersteunen Flatpak al, en hetzelfde geldt voor de desktopomgevingen KDE en Gnome.

Probeer Flatpak maar eens. Het kan geen enkel kwaad om flatpaks naast de gewone door de distributie geleverde applicaties te gebruiken.

Links

1. <https://sourceforge.net> resp. <https://github.com>
2. Zie: <https://www.zdnet.com/article/linux-desktop-leaders-unite-behind-flathub-app-store-heres-why/>
3. Zie: <https://flathub.org/nl/setup/Ubuntu>
4. CLI = Command Line Interface ofwel de opdrachtregel
5. GUI = Graphical User Interface ofwel de grafische gebruikersinterface
6. Eigendomsmatige software, dit is software waarvan de exclusieve auteursrechten bij een individu of bedrijf liggen, dat geen toegang tot de broncode geeft en het recht op kopiëren, wijzigen en bestuderen niet verleent of sterk beperkt.
7. Zo blijkt mijn openSUSE-systeem, waarin de meeste applicaties door flatpaks zijn vervangen, 7 à 8 GB meer ruimte in te nemen dan het reguliere openSUSE-systeem. En die extra ruimte komt goed overeen met die van de runtimes.
8. Het *host*-systeem (gastheer) is het (Linux-) besturingssysteem waarin Flatpak draait.
9. In dit verband zijn afhankelijkheden pakketten met bibliotheken en andere hulpsoftware, waarvan de applicatie in kwestie afhankelijk is.
10. Een bestand met de extensie `.desktop` bevat alle gegevens die nodig zijn om de applicatie in het juiste menu te plaatsen, en uit te voeren.
Zie bijvoorbeeld de `.desktop`-bestanden in de map `/usr/share/applications`
11. Zie: <https://www.pragmaticlinux.com/2021/01/how-to-install-flatpak-applications-from-flathub/>