Open Source Software onder controle

By Marcel Kornegoor

Agenda

- Welkom!
- Wie ben ik?
- Over AT Computing
- Wat is Open Source Software en hoe werkt het?
- Vijf risicogebieden

About me

Marcel Kornegoor

Co-owner & director @ AT Computing

- 15 years of experience in IT.
- Huge fan of open source software.
- Likes to share knowledge.
- Co-founder of DOSBA.
- Husband, father, cyclist and nerd by night.

AT Computing — Who We Are & What We Do

AT Computing is a **Dutch open-source IT consultancy and training company** that helps organizations design, implement, and manage reliable, automated IT environments.

What We Do

- **Training & Education** — Hands-on courses in Linux, Ansible, Kubernetes, Python, and more.
- **Consultancy** — Independent advice and support for open-source infrastructure and automation.
- **Staffing & Expertise** — Experienced specialists available to strengthen IT teams.
- **Custom Solutions** — Tailor-made training and implementation trajectories for specific needs.

Why Choose AT Computing?

- Deep expertise in **open-source technologies**.
- **Independent** — no vendor lock-in or sales agenda.
- Trusted partner since **1985**, known for quality and practical knowledge.

Proud member of THE LINUX FOUNDATION AUTHORIZED TRAINING PARTNER

Wat is Open Source Software?

- Meer dan "de broncode is (publiek) beschikbaar"
- Vier fundamentele vrijheden (Free Software Foundation):
  - Bestuderen
  - Aanpassen
  - Distribueren / verkopen
  - Gebruiken
- ~70-90% van alle software wereldwijd is open source software

"Free as in freedom"

(not free beer)

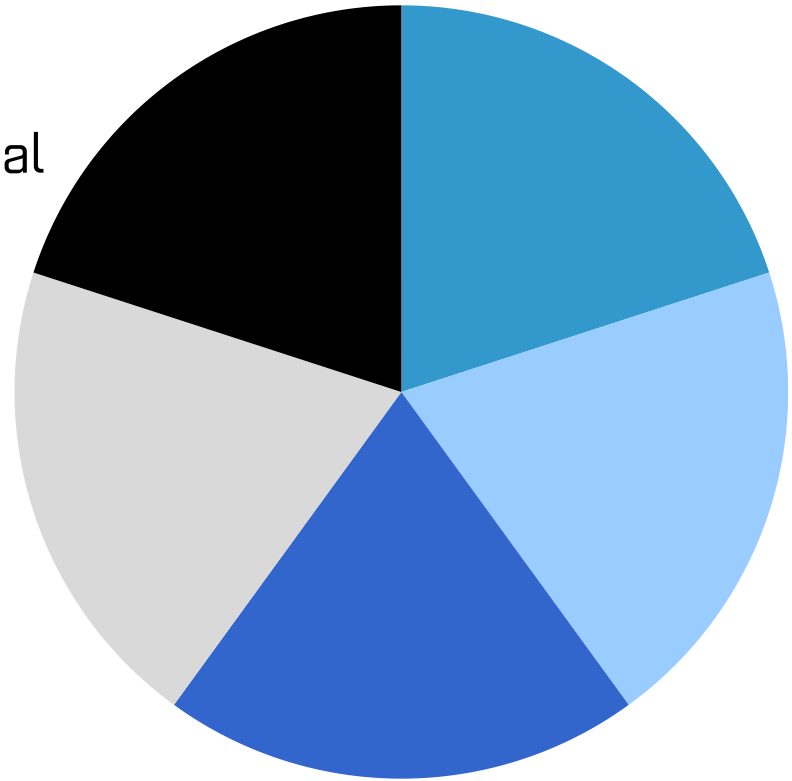
https://www.fsf.org/

Hoe werkt Open Source Software?

- Iedereen met een goed idee kan een project beginnen
- Uitgangspunt is vrijheid: delen en hergebruiken
- "Community" (gemeenschap) helpt bij het verder ontwikkelen/verbeteren/promoten
- Staat of valt bij geven en nemen
- Biedt een alternatief voor vendor-gebaseerde, niet vrije (closed/proprietaire) software
- Kan in veel gevallen kosteloos worden gebruikt, maar is niet gratis
- Diverse grote, internationale projecten met vele ontwikkelaars en sponsoren

https://www.linuxfoundation.org/

Vijf belangrijke risicogebieden


Strategic & Commercial
License & Legal
Governance & Shadow IT
Security & Vulnerability
Operational

Aanpak in dit webinar

Per risico-cluster

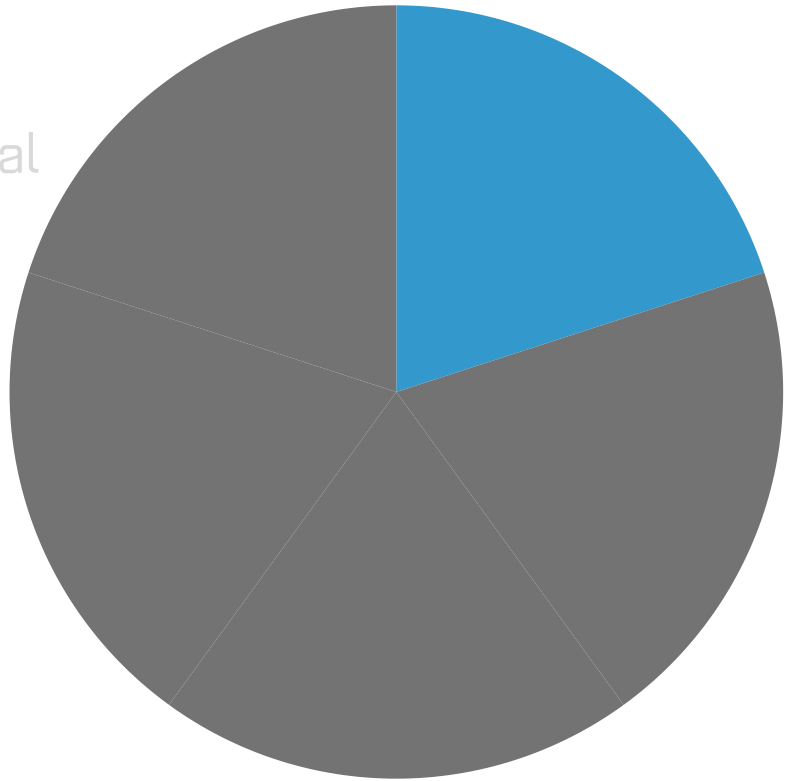
- welk(e) risico('s) kun je lopen?
- welke beheersmaatregelen kun je treffen?

vrijwaring: dit is slechts een overview van een uurtje

iedere situatie is anders

er zijn nog meer risico’s…

…en ook nog meer beheersmaatregelen


License
&
Legal
Security
&
Vulnerability
Operational
Governance
&
Shadow IT
Strategic
&
Commercial

De risico’s

- Er zijn veel verschillende OSS licenties in omloop (incompatibiliteit)
- Gebruik van copyleft licentie in een proprietair product (zoals GPL)
- Gebrek aan inzicht in licentiestructuur (complexe bundeling)
- onjuiste/ontbrekende attributie
- Meestal geen enkele vorm van garantie bij schade

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND

https://mit-license.org/

https://choosealicense.com/

De beheersmaatregelen

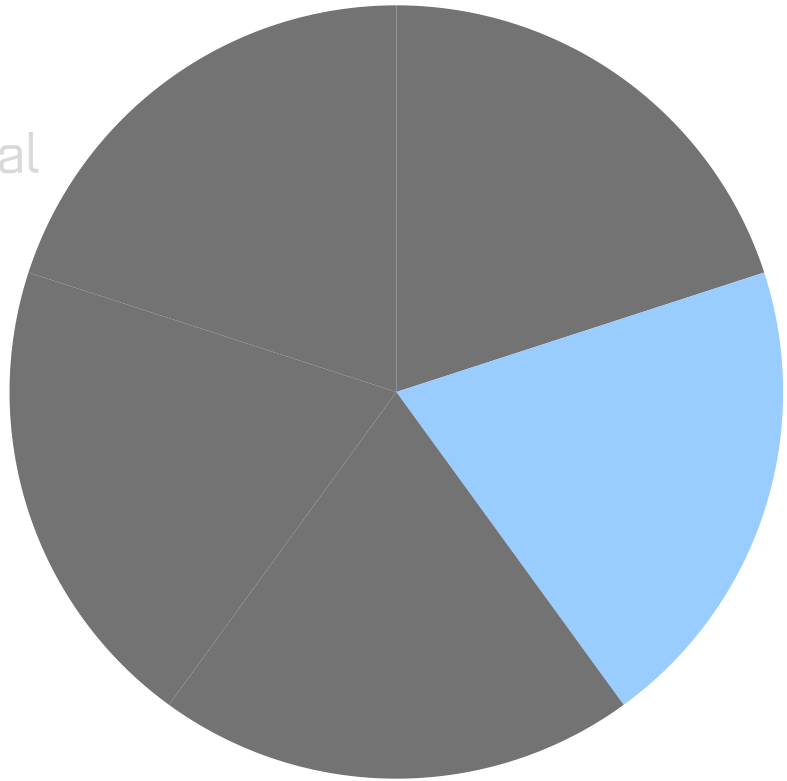
- Voer (automatische) licentie-analyse/scanning uit
- Denk na over het business model (kan de broncode echt niet worden gepubliceerd?)
- Onderhoud een SBOM (Software Bill Of Materials)
- Stel een lijst op met toegestane licentievormen

https://www.aikido.dev/blog/top-open-source-license-scanners

AT COMPUTING

Strategic & Commercial

License & Legal



Governance & Shadow IT

Security & Vulnerability

Operational

De risico’s

- Al snel enorm veel afhankelijkheden (dependencies)
- Gebruik van componenten met bekende kwetsbaarheden (CVE’s)
- Niet altijd snel een update beschikbaar
- Malafide packages (supply chain attack)

Voor deze presentatie (Slidev) was installatie van 600+ javascript/npm packages nodig…

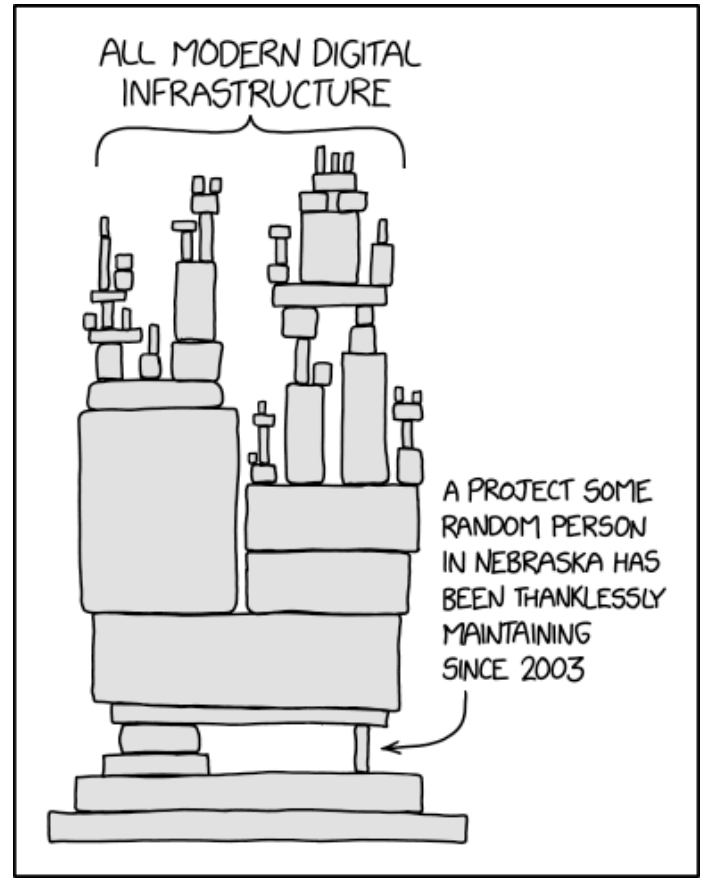
https://owasp.org/www-project-open-source-software-top-10/

https://sli.dev/

Voorbeeld: Log4Shell (Log4J)

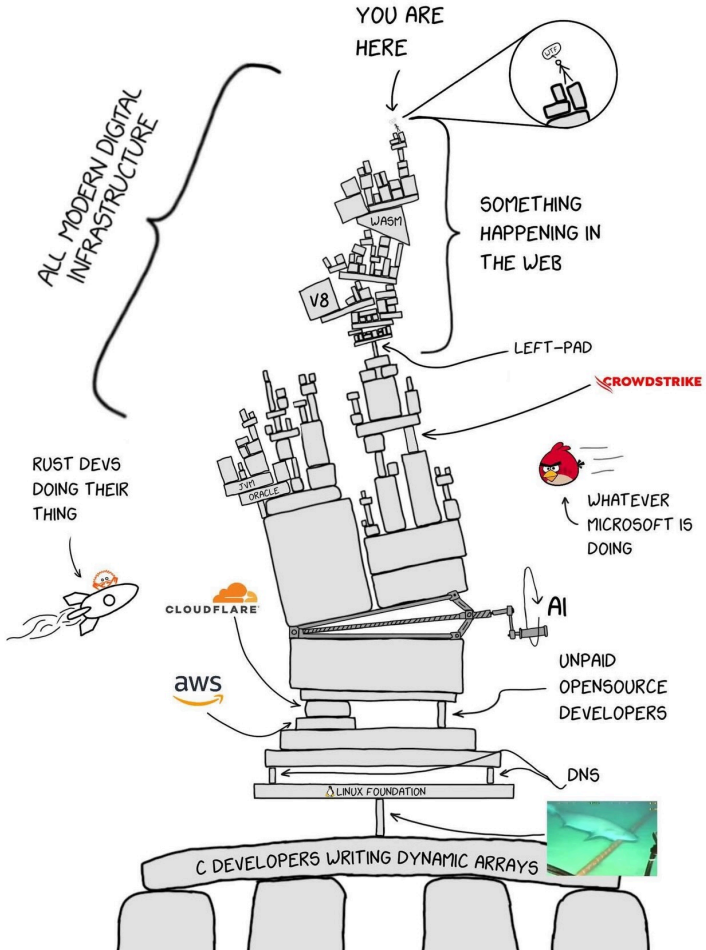
- Kritiek veiligheidslek
- CVE-2021-44228
- "It’s like sugar. It’s in everything"
- CVSS score: 10 (uit 10)
- 1 relatief klein component
- Klein team van ontwikkelaars (project van Apache Foundation)
- Nog steeds niet alle software is gepatcht (ruim 4 jaar na dato)

Brieven van hele dure advocaten hielpen he-le-maal niets



https://www.cve.org/CVERecord?id=CVE-2021-44228

https://xkcd.com/2347/



De beheersmaatregelen

- Dependency scanning (bijv in CI/CD pipeline)
- CVE scanning
- Implementeer een patch management beleid
- Gebruik een interne Artifact Repository
- Threat modeling voor kritieke componenten
- Betaal een security audit/pentest voor het open source project of deel je bevindingen!

https://dev.to/samlan/top-dependency-scanners-a-comprehensive-guide-2kf

https://owasp.org/www-project-dependency-check/

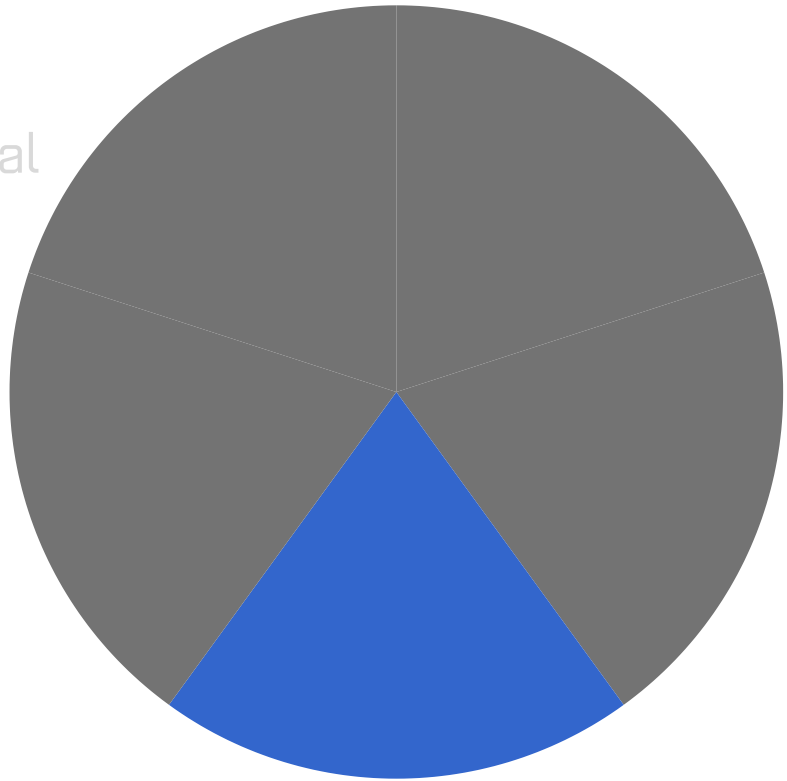
https://www.sonatype.com/products/sonatype-nexus-repository

https://www.harness.io/blog/what-is-artifact-repository

https://owasp.org/www-community/Vulnerability_Scanning_Tools

Strategic & Commercial

License & Legal



Governance & Shadow IT

Security & Vulnerability

Operational

De risico’s

- Eenpitters & weesprojecten
- Onvoorspelbare update-cycli
- Gebrekkige/ontbrekende documentatie
- Gebrek aan ondersteuning
- Onvolwassenheid (bugs/ontwerpfouten)
- Project forking

https://www.reddit.com/r/ProgrammerHumor/comments/1b257b3/fiveyearsofselftaught/

De beheersmaatregelen

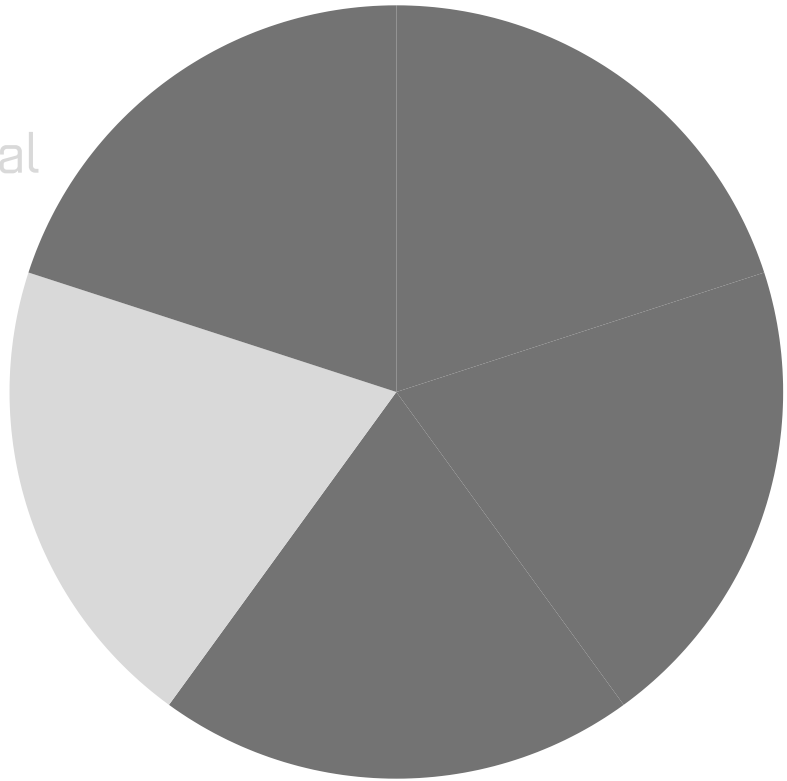
- "Health check" voor gebruik
- Opstellen van een exit strategie
- Interne kennisontwikkeling
- Adopteer een project
- Doneer aan een project
  - Geld
  - Resources (hosting, mensen, kennis)

AT COMPUTING

Strategic & Commercial

License & Legal

Governance & Shadow IT



Security & Vulnerability

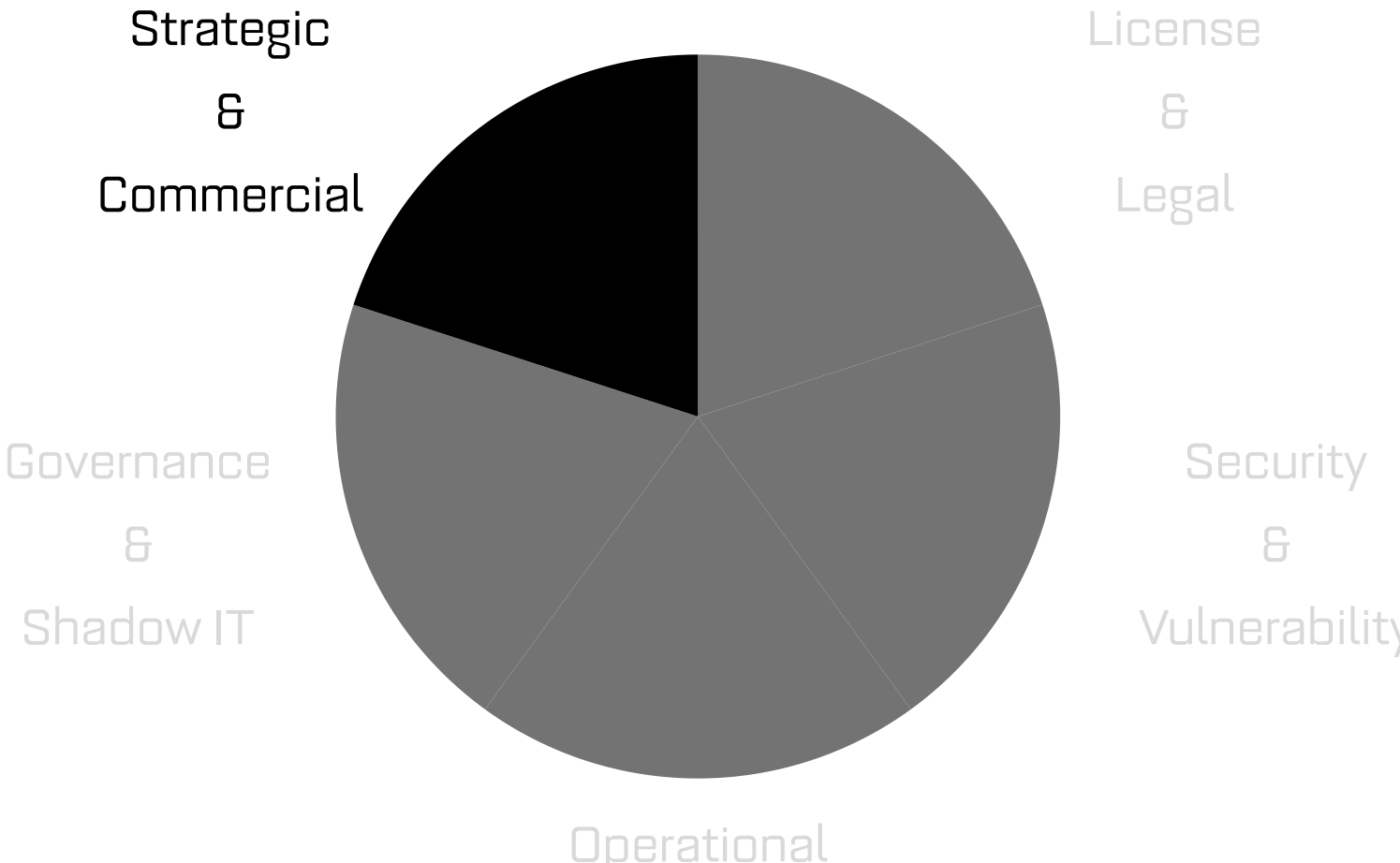
Operational

De risico’s

- Geen zicht op gebruikte componenten (libraries, modules, packages, images)
- Eigenaarschap / verantwoordelijkheid niet belegd
- Tool sprawl / gebrek aan standaardisatie
- Geen centraal overzicht van gebruikte tools
- Technical debt / problemen met onderhoud

De beheersmaatregelen

- Zorg voor duidelijk open source beleid/visie (+handhaven)
- Neem bewuste besluiten wat wel/niet te gebruiken
  - Hanteer selectiecriteria
- Stel een heldere architectuur op met voorkeurs-tools
- Maak keuzes! (tool-discussies zullen er altijd zijn)
- Centraliseer goedgekeurde software
  - Artifact Repository
  - Software catalogus
  - ASL (Approved Softare List)


Strategic
&
Commercial
License
&
Legal
Security
&
Vulnerability
Operational
Governance
&
Shadow IT

De risico’s

- Verandering van licentiemodel
  - HashiCorp -> van Mozilla Public License v2.0 naar BSL
  - Docker -> restricties op gratis gebruik
- Verandering van strategie -> MinIO stopt met ondersteuning (publieke) container images
- Verandering van jurisdictie: GitLab + Elastic… 🇳🇱 -> 🇺🇸
- Overnames / investeerders (OwnCloud, Docker, OpenOffice, MySQL, Red Hat, HashiCorp)

De beheersmaatregelen

- "Due diligence" naar leveranciers en geldstromen
- Actief ondersteunen van alternatieven (geld, resources, mankracht)
- Risicospreiding: niet alles op 1 vendor gokken

Omarm de risico’s: je krijgt er veel voor terug!

Risico’s die kansen meebrengen

- Minder vendor lock-in
- Minder afhankelijkheid
- Meer transparantie / controle
- Lagere kosten
- Lagere innovatie-drempel
- Meer voldoening: je draagt bij

OSS Checklist

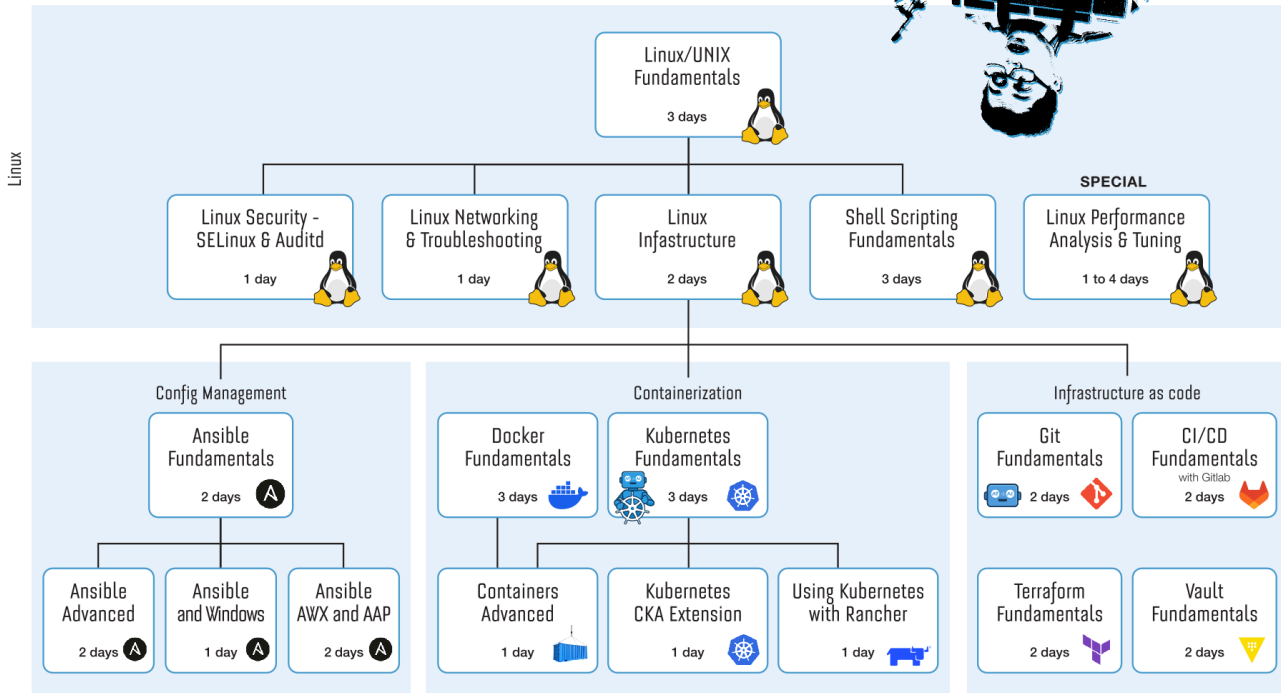
https://atcomputing.nl/oss_checklist

Uniek in 2026: direct te downloaden zonder gegevens achter te laten!

ATYPICAL OPEN SOURCE LEARNING JOURNEY

for Cloud Engineers, DevOps Engineers & SysAdmins

SCAN THE QR CODE FOR A DIGITAL VERSION OF THE LEARNING JOURNEY!



COMPUTING
ATYPICAL OPEN SOURCE GURUS

atcomputing.nl

v.2025

ATYPICAL OPEN SOURCE LEARNING JOURNEY

for Generative AI

SCAN THE QR CODE FOR A DIGITAL VERSION OF THE LEARNING JOURNEY!

Core GenAI

- GenAI for DevOps - Build Your Own LLm Server — 1 day — AI-geletterd
- GenerativeAI & Security — 1 day — AI-geletterd
- Vibe Coding — 1/2 day

Python Programming

For writing your own Gen AI / LLM powered apps or solutions.

- Learn to Program with Python — 5 days
- Deep Learning with Pytorch — 2 days — PyTorch
- Create a REST API in Python with FastAPI — 2 days
- Python for Data Analysis - Introduction to PANDAS — 2 days — pandas

Platform & Containerization

For running your own LLM or GenAI apps.

- Kubernetes Fundamentals — 3 days
- Linux/UNIX Fundamentals — 3 days
- Terraform Fundamentals — 2 days
- Shell Scripting Fundamentals — 3 days
- Docker Fundamentals — 3 days
- Git Fundamentals — 2 days
- CI/CD Fundamentals with Gitlab — 2 days

This training uses AI.

AT COMPUTING
ATYPICAL OPEN SOURCE GURUS

atcomputing.nl

v.2025

ATYPICAL OPEN SOURCE LEARNING JOURNEY

for Software Engineers

SCAN THE QR CODE FOR A DIGITAL VERSION OF THE LEARNING JOURNEY!

Python

- Learn to Program with Python — 5 days
- The Python Programming Language — 3 days
- Python for Data Analysis - Introduction to PANDAS — 2 days
- NumPy — 1 day
- Python Advanced — 4 days
- Create a REST API in Python with FastAPI — 2 days
- Deep Learning with Pytorch — 2 days

SPECIAL

- Vibe Coding — 1/2 day

Other Languages

- The C++ Programming Language — 4 days
  - What's new in C++17 — 1 day
  - What's new in C++20 — 2 days
  - C++ Advanced — 4 days
  - What's new in C++23 — 1 day
  - **SPECIAL**
  - Design patterns for developers — 3 days
- The C Programming Language — 4 days
- The Fortran Programming Language — 4 days

Engineering Toolbox

- Linux/UNIX Fundamentals — 3 days
- Docker Fundamentals — 3 days
- Git Fundamentals — 2 days
- CI/CD Fundamentals with Gitlab — 2 days

This training uses AI.

AT COMPUTING — ATYPICAL OPEN SOURCE GURUS

atcomputing.nl

v.2025

https://atcomputing.nl/leerpaden

Thank You!

Questions?

https://atcomputing.nl

marcel@atcomputing.nl