



DIGITALE
SOUVERÄNITÄT
FÜR EUROPÄISCHE
SOFTWAREENTWICKLER!



Digitale Souveränität

Für Europäische Softwareentwickler

Siegfried GÖSCHL
siegfried.goeschl@asciifish.net

ASCIIFISH GmbH

Siegfried GÖSCHL

- » Java Backend Developer
- » Open Source Evangelist
- » Mitgründer der **Java User Group Austria** in 2001
- » Hilft beim **Java Meetup Vienna & DevFest Vienna**



Was Ist Digitale Souveränität für
Softwareentwickler?!

Unabhängigkeit von teuren
proprietären Tools und/oder
Konzernen durch strategische
Technologieentscheidungen

A black and white photograph of a person from the chest up. They are wearing a dark t-shirt and a VR headset. Their eyes are closed or heavily shadowed. In front of them is a computer monitor displaying a dark, abstract scene with some glowing points. The overall mood is mysterious and focused.

DEVELOPER ONBOARDING

Fikitives Developer Onboarding

- » Linux und Macs werden nicht unterstützt - ist Company Policy
- » Es kann nur Outlook sein - wegen Exchange Server
- » Microsoft Word & Powerpoint für das tägliche Arbeiten
- » Technische Dokumente in Confluence und Sharepoint

Fikitives Developer Onboarding

- » Bug Tracking in JIRA
- » Open Source Programmierung vertraglich ausgeschlossen
- » Firmenlizenz für Oracle Datenbanken
- » Produktion wird von On-Prem auf Google Cloud migriert

DIGITAL DEPENDENCIES



DIGITAL DEPENDENCIES EVERYWHERE

HERE'S YOUR LICENSE RENEWAL



Forbes

Microsoft Can't Keep EU Data Safe From US Authorities

By [Emma Woollacott](#), Senior Contributor. © Emma Woollacott is a freelance jo...

Follow Author

Published Jul 22, 2025 at 07:56am EDT

Trump befiehlt – Microsoft sperrt E-Mail-Account?

News

19. Mai 2025 • 4 Minuten

[Cloud Computing](#) [E-Mail](#) [IT-Strategie](#)

Berichten zufolge hat sich Microsoft den Dekreten des US-Präsidenten gefügt und Karim Khan, Chefankläger am Internationalen Strafgerichtshof, das E-Mail-Konto gesperrt. Doch das Unternehmen dementiert.



Wen US-Präsident Donald Trump einmal aufs Korn nimmt, hat nichts zu lachen.

Quelle: Jonah Elkowitz/Shutterstock

Verwandte Inhalte

NEWS-ANALYSE

IT-Führungskräfte verschwenden die Hälfte ihrer Cloud-Ausgaben

Von Evan Schuman

1. Okt. 2025 • 4 Minuten

[Budget](#) [Cloud Computing](#)

KOMMENTAR

13 Gründe gegen SQL

Von Peter Wayner

29. Sept. 2025 • 9 Minuten

[Big Data](#) [Entwickler](#) [IT-Strategie](#)

NEWS

Amazon und SAP kooperieren beim Thema Datensouveränität

Von John E. Dunn

25. Sept. 2025 • 4 Minuten

[Amazon Web Services](#) [Cloud Computing](#)

TIPP

Und Jetzt?!

*Wer etwas will, findet Wege. Wer etwas
nicht will, findet Gründe.*

— Götz Werner, Gründer dm Drogerie Markt

[□ 653 Postings](#)**ABMARSCH**

Microsoft wird ausgemustert: Bundesheer wechselt zu Libreoffice

Bei der Verwendung der Open-Source-Software gehe es nicht ums Sparen, sondern um "digitale Souveränität"

18. September 2025, 17:00

[□ 653 Postings](#)[⌚ Später lesen](#)

Österreichs Bundesheer stellt auf LibreOffice um

Alle Heise-Foren > heise online > Kommentare > Österreichs Bundesheer stellt... > Ich bin schockiert ...



Beitrag



Threads

Anmelden und mitdiskutieren



cd

mehr als 1000 Beiträge seit 07.01.2000

19.09.2025 17:49

Ich bin schockiert ...

... dass hier das österreichische Bundesheer - von manchen Grundwehrdienern auch liebevoll "Idiotenverein" genannt - hier tatsächlich eine Vorreiterrolle spielt. *Die Wöd stöt nimma long **)

cd

*) Die Welt steht nicht mehr lange, frei nach Nestroy.

Bundesheer Migration

- » In 2020 war klar, daß Microsoft Office in die Cloud wandert
- » In 2023 wurde deutsches Unternehmen mit Support und externer Entwicklung beauftragt
- » Es wurden 16.000 Mitarbeiter auf LibreOffice umgeschult
- » Änderungen / Erweiterungen gehen zurück an LibreOffice
- » LibreOffice wird von einer Stiftung betrieben - The Document Foundation

Strategien zur Verbesserung der digitalen Souveränität in der Softwareentwicklung

- » Priorisierung von Open Source
- » Reduktion technologischer Abhängigkeiten
- » Eigene Infrastruktur & Datenkontrolle

Priorisierung Von Open Source

- » Einsatz freier Entwicklungsumgebungen & Tools
- » Freie Datenbanken (PostgreSQL, MariaDB) statt kommerzieller Varianten (Oracle, MSSQL)
- » Verwendung von Open Source Frameworks (Spring Boot, Quarkus, Django, Angular, React)
- » Prüfung der Lizenz bei Verwendung von Open Source notwendig

Reduktion Technologischer Abhängigkeiten

- » Offene Standards (z. B. OpenAPI, SQL, OAuth) statt proprietärer Technologien
- » Cloud-Strategie prüfen: Multi-Cloud oder Hybrid-Ansätze statt nur AWS, Azure oder GCP
- » Einsatz von Container (Docker, Podman) und Orchestrierung (Kubernetes) in der Cloud

Eigene Infrastruktur & Datenkontrolle

- » Verwendung offener Datenformate (XML, Markdown, ODT, EML)
- » LibreOffice & NextCloud statt Microsoft
- » Self-Hosting wichtiger Systeme (z. B. Git, CI/CD, Issue-Tracker)
mit Tools wie GitLab CE, Gitea, Jenkins, SonarQube
- » Datenhaltung bevorzugt auf eigener Hardware oder in souveränen
europäischen Clouds

Open Source Lizenzen

Ein Überblick

Open Source Lizenzen

Aspekt	Permissive Lizenz	Copyleft Lizenz
Weiterverwendung	Quellcode darf in propriäre (geschlossene) Software integriert werden.	Quellcode darf nur in freie Software integriert werden, die ebenfalls offen bleibt.
Pflichten bei Weitergabe	Nur Urheberrechtshinweis und Lizenztext beibehalten.	Modifizierte Versionen müssen unter derselben Lizenz veröffentlicht werden.
Kommerzielle Nutzung	Uneingeschränkt erlaubt, auch in geschlossenen Produkten.	Erlaubt, aber Quellcode muss offen gelegt werden, wenn verteilt.
Beispiele	MIT, BSD, Apache 2.0	GPL, LGPL, MPL, EPL, GNU Affero GPL
Zielsetzung	Maximale Verbreitung und Flexibilität.	Sicherstellung, dass Software immer frei bleibt .

Lizenz	Typ (Copyleft?)	Kurzbeschreibung	Vorteile	Nachteile	Typische Einsatzszenarien
MIT License	Nein (permissiv)	Sehr einfach, erlaubt uneingeschränkte Nutzung und Modifikation bei Erhalt des Copyright-Hinweises.	- Extrem einfach - Kompatibel mit anderen Lizenzen - Weit verbreitet	- Keine Pflicht zur Offenlegung - Proprietäre Nutzung erlaubt	◆ JavaScript-/Python-Bibliotheken ◆ Startups & akademische Projekte
Apache License 2.0	Nein (permissiv)	Enthält Patentklauseln und ist für Unternehmen rechtlich sicherer als MIT.	- Patent-Schutz - Unternehmensfreundlich - Rechtlich klar	- Etwas längerer Lizenztext	◆ Cloud-/Enterprise-Projekte ◆ Firmen-Open-Source (Google, Apache Foundation)
BSD 3-Clause	Nein (permissiv)	Ähnlich MIT, aber mit zusätzlicher Namensverwendungsbeschränkung.	- Minimaler Verwaltungsaufwand - Kommerzfreundlich	- Kein Schutz vor proprietärer Nutzung	◆ Systemnahe Software (z. B. FreeBSD) ◆ Embedded & Netzwerktools
GPLv3 (GNU General Public License)	Ja (starkes Copyleft)	Erzwingt Offenlegung und Weitergabe unter gleichen Bedingungen.	- Garantiert Offenheit - Fördert Community-Kollaboration	- Eingeschränkte Kompatibilität - Für Firmen oft zu restriktiv	◆ Community-Projekte ◆ Betriebssysteme (Linux) ◆ Open-Hardware-Software
LGPLv3 (Lesser GPL)	Ja (schwaches Copyleft)	Offenlegungspflicht nur bei Änderungen an der Bibliothek selbst.	- Gute Balance zwischen Offenheit & Nutzung - Ideal für Libraries	- Juristisch anspruchsvoll	◆ Bibliotheken & Frameworks ◆ Middleware-Komponenten
MPL 2.0 (Mozilla Public License)	Ja (schwaches Copyleft)	Copyleft gilt nur auf Datei-Ebene; erlaubt proprietäre Kombinationen.	- Feingranulares Copyleft - Rechtlich ausgewogen	- Weniger verbreitet	◆ Browser-nahe Projekte (Firefox) ◆ Unternehmenssoftware mit Open-Core
EPL 2.0 (Eclipse Public License)	Ja (schwaches Copyleft)	Unternehmensfreundlich; weit verbreitet in Java-Ökosystemen.	- Klar und präzise - Geeignet für Firmenkollaborationen	- Weniger bekannt außerhalb Eclipse	◆ Java-Ökosystem ◆ Unternehmenssoftware ◆ Open-Source-Plattformen

Datenbanken

Vendor Lock-in Vermeidung

Datenbanken

Datenbank	Typ	Vorteile	Nachteile
Oracle Database	Kommerziell	<ul style="list-style-type: none">- Sehr leistungsfähig & hochskalierbar- Umfassende Enterprise-Features (RAC, Data Guard, Partitionierung)- Hohe Stabilität & Sicherheit- Weltweit verbreitet in Großunternehmen- Umfangreicher Hersteller-Support	<ul style="list-style-type: none">- Sehr hohe Lizenz- und Supportkosten- Komplexes Lizenzmodell- Proprietär → starker Vendor Lock-in- Administration komplex
Microsoft SQL Server (MSSQL)	Kommerziell	<ul style="list-style-type: none">- Gute Integration in Microsoft-Ökosystem (Windows, Azure, PowerBI)- Leistungsfähige BI- und Reporting-Features- Stabil, weit verbreitet im Enterprise- Gute Tools für Entwickler (SSMS, Profiler)	<ul style="list-style-type: none">- Proprietär, Vendor Lock-in- Lizenzkosten hoch (Enterprise Edition)- Primär auf Windows/Linux- Abhängigkeit von Microsoft-Strategie
MySQL (Community Edition)	Frei (Open Source, gehört Oracle)	<ul style="list-style-type: none">- Weit verbreitet, große Community- Einfach zu installieren & zu administrieren- Gute Performance für Webanwendungen- Viele Hosting-Anbieter unterstützen MySQL nativ- Kostenlos in Community Edition	<ul style="list-style-type: none">- Kommerzielle Versionen mit erweiterten Features proprietär- Weniger standardkonform als PostgreSQL- Eingeschränkte Enterprise-Funktionalitäten- Teilweise Abhängigkeit von Oracle

Datenbanken

Datenbank	Typ	Vorteile	Nachteile
PostgreSQL	Frei (Open Source)	<ul style="list-style-type: none">- Vollständig Open Source & kostenlos- Sehr leistungsfähig, stabil & erweiterbar- Starke Standardkonformität (SQL, ACID)- Erweiterungen (PostGIS, TimescaleDB, FDWs)- Plattformunabhängig, große Community	<ul style="list-style-type: none">- Weniger „Out-of-the-box“-Enterprise-Tools als Oracle- Komplexere Administration im Vergleich zu MySQL/MariaDB- Kein zentraler Hersteller-Support (nur Drittanbieter)
MariaDB	Frei (Open Source, Fork von MySQL)	<ul style="list-style-type: none">- Komplett Open Source- Kompatibel zu MySQL (Drop-in Replacement)- Flexibel durch viele Storage Engines- Schnell & leichtgewichtig- Plattformübergreifend	<ul style="list-style-type: none">- Kleinere Community als MySQL/Postgres- Weniger Enterprise-Features- Weniger verbreitet im Enterprise-Sektor- Für extreme Skalierung nicht so optimiert wie Oracle/PG

Personal Techstack

- » MacBook Pro
- » Git für die Versionskontrolle
- » Sublime (commercial) & Visual Code als Editor
- » IntelliJ Ultimate als IDE (commercial)
- » DbVisualizer für Datenbankzugriff (commercial)

Personal Techstack

- » Markdown & AsciiDoc für Dokumentation
- » Mermaid für Diagramme
- » Bear für Notizen mit Markdown (commercial)
- » Deckset für Präsentation mit Markdown (commercial)
- » Backendentwicklung mit Java, Maven & Spring



WHAT IS YOUR NEXT MOVE?!



THANK
YOU



Resources

- » Criminal Court: Microsoft's email block a wake-up call for digital sovereignty
- » Österreichs Bundesheer stellt auf LibreOffice um
- » EU cloud provider: Broadcom raised VMware license prices by 800 to 1500 percent
- » Microsoft Can't Keep EU Data Safe From US Authorities
- » US-Provider würden Kundendaten aushändigen

Resources

- » [Trump befiehlt – Microsoft sperrt E-Mail-Account?](#)
- » [The Document Foundation](#)

Unused Content

Übersicht Open Source & kommerzielle Tools

Linux Distributionen

Distribution	Fokus-Bereiche	Vorteile	Nachteile
Ubuntu	Desktop, Server, Cloud	<ul style="list-style-type: none">- Sehr große Community & viele Tutorials- Einsteigerfreundlich- Breite Softwareunterstützung- LTS-Versionen mit 5 Jahren Support	<ul style="list-style-type: none">- Auf Canonical-Ökosystem ausgerichtet- Teilweise konservative Paketversionen- Telemetrie-Diskussionen
Debian	Server, Entwickler, Minimal-Desktop	<ul style="list-style-type: none">- Sehr stabil und zuverlässig- Riesige Paketquellen- Freiheitsfokus (DFSG)- Weit verbreitet als Server-OS	<ul style="list-style-type: none">- Pakete oft älter (konservativ)- Weniger benutzerfreundlich- Längere Release-Zyklen
Fedora	Entwickler, Desktop	<ul style="list-style-type: none">- Sehr aktuell mit neuesten Features- Community-getrieben, von Red Hat unterstützt- Gute Integration moderner Technologien (Wayland, SELinux)	<ul style="list-style-type: none">- Kürzerer Support-Zyklus (~13 Monate)- Manchmal instabiler als Debian/Ubuntu- Starker Fokus auf Red Hat-Ökosystem
RHEL (Red Hat Enterprise Linux)	Enterprise-Server, Cloud, zertifizierte Systeme	<ul style="list-style-type: none">- Enterprise-Support & Zertifizierungen- Sehr stabil & langzeitunterstützt- Weit verbreitet im Unternehmensumfeld	<ul style="list-style-type: none">- Kostenpflichtiger Support- Starke Bindung an Red Hat-Ökosystem- Weniger aktuell bei Desktop-Paketen

Linux Distributionen

Distribution	Fokus-Bereiche	Vorteile	Nachteile
CentOS Stream	Server, Cloud, Testumgebungen	<ul style="list-style-type: none">- Rolling-Release nah an RHEL- Stabilität durch RHEL-Upstream- Gut für Testumgebungen von RHEL-Deployments	<ul style="list-style-type: none">- Nicht mehr klassisch "Enterprise-stabil" wie früher CentOS- Kürzere Support-Laufzeiten- Stärker an Red Hat gebunden
openSUSE Leap	Server, Enterprise-Desktop	<ul style="list-style-type: none">- Stabil & für Unternehmen geeignet- YaST als mächtiges Admin-Tool- Kombination aus Community & SUSE-Enterprise	<ul style="list-style-type: none">- Weniger verbreitet als Ubuntu/Debian- Kleinere Community- Leap nicht so aktuell wie Tumbleweed
openSUSE Tumbleweed	Entwickler, Desktop	<ul style="list-style-type: none">- Rolling Release mit neuesten Paketen- Sehr aktuell, Entwicklerfreundlich- Gute Stabilität trotz Rolling-Modell	<ul style="list-style-type: none">- Höherer Wartungsaufwand (häufige Updates)- Nicht ideal für Produktionsserver- Kleinere Nutzerbasis
Arch Linux	Entwickler, Power-User-Desktop	<ul style="list-style-type: none">- Extrem flexibel und minimalistisch- Rolling Release (immer aktuell)- Riesige Community (Wiki!)- Volle Kontrolle für Power-User	<ul style="list-style-type: none">- Komplexe Installation- Wartungsintensiv- Updates können Systeme brechen- Wenig geeignet für Einsteiger
Linux Mint	Desktop (Einsteiger, Umsteiger von Win)	<ul style="list-style-type: none">- Einsteigerfreundlich (Windows-ähnliche Oberfläche)- Auf Ubuntu/Debian basiert- Sehr stabil- Fokus auf Desktop-User	<ul style="list-style-type: none">- Weniger innovativ- Abhängig von Ubuntu/Debian-Ökosystem- Kleinere Entwicklerbasis

Programmiersprachen

Sprache	Abhängigkeit	Vorteile für Souveränität.	Nachteile / Verzicht
Python	Community-getrieben, keine einzelne Firma	Enorm große Community, viele freie Tools, Plattformunabhängig, starker Fokus auf Wissenschaft, KI, DevOps	Teilweise Performance (im Vergleich zu C/Go/Java), viele inkompatible Libraries
Java	Oracle (Historisch), Eclipse Foundation, OpenJDK	Sehr viele Open-Source-Implementierungen (OpenJDK, Eclipse Temurin, Azul, Adoptium), „Enterprise-lastig“ große Community, stabil	Teilweise Oracle-Politik bzgl. Lizenzen, „Enterprise-lastig“
C/C++	ISO-Standard, Compiler vielfach Open-Source (GCC, Clang)	Sehr souverän, extrem viele freie Werkzeuge, Industriestandard	Komplexe Toolchains, steile Lernkurve
Go (Golang)	Google (ursprünglich), aber Open-Source	Einfach, modern, hervorragende Performance, sehr gutes Ökosystem, viele freie Tools	Google hat starken Einfluss (Governance, Roadmap)
Rust	Community & Rust Foundation (unabhängig)	Sehr moderne Sprache, performant, sicher, Open-Source, wachsendes Ökosystem	Noch weniger Enterprise-Tooling, Einarbeitung schwieriger
JavaScript/TypeScript	Starker Einfluss von Microsoft (TypeScript) und Google (Node.js/V8)	Sehr große Entwicklerbasis, viele freie Frameworks, Browser als Plattform	Abhängigkeit von Google (V8) und Microsoft (TS, GitHub/VS Code)
C#/.NET	Trotz .NET Core Öffnung starke Microsoft-Bindung	Große Entwicklerbasis, freie Frameworks	Verlust des "First Class Supports" bei Verwendung anderer Sprachen

Entwicklungsumgebungen

Entwicklungsumgebung	Vorteile	Nachteile
Emacs	<ul style="list-style-type: none">- Extrem anpassbar & erweiterbar (Lisp)- Vollständig Open Source- Plattformunabhängig- Jahrzehntelang bewährt, riesige Community	<ul style="list-style-type: none">- Sehr steile Lernkurve- Bedienung ungewohnt- Mit vielen Erweiterungen ressourcenintensiv
Vim / Neovim	<ul style="list-style-type: none">- Sehr leichtgewichtig & blitzschnell- Läuft auf fast allen Systemen- Tastaturzentrierte Bedienung- Riesige Community, viele Plugins	<ul style="list-style-type: none">- Hohe Einstiegshürde für Neulinge- Weniger komfortabel als moderne IDEs- Kaum integrierte Enterprise-Features
VS Code (Microsoft)	<ul style="list-style-type: none">- Kostenlos & weit verbreitet- Riesiges Erweiterungs-Ökosystem- Git-Integration & Debugging integriert- Plattformübergreifend- Benutzerfreundlich & modern	<ul style="list-style-type: none">- Offizielle Builds enthalten Telemetrie- Gefahr von Vendor Lock-in (Azure/GitHub)- Kein vollständiger IDE-Ersatz- Performance bei sehr großen Projekten eingeschränkt
IntelliJ IDEA Community Edition	<ul style="list-style-type: none">- Kostenlos & Open Source- Sehr benutzerfreundlich- Starke Java- & Kotlin-Unterstützung- Plattformübergreifend- Gute Integration mit Maven/Gradle	<ul style="list-style-type: none">- Eingeschränkter Funktionsumfang (kein Spring, kein Java EE, keine erweiterten Frameworks)- Weniger für Enterprise-Anwendungen geeignet

Entwicklungsumgebungen

Entwicklungsumgebung	Vorteile	Nachteile
IntelliJ IDEA Ultimate	<ul style="list-style-type: none">- Vollständige IDE mit allen Enterprise-Features- Breite Unterstützung für Java, Kotlin, Spring, Java EE, Datenbanken, Webframeworks- Sehr gute Code-Analyse & Refactoring- Plattformübergreifend	<ul style="list-style-type: none">- Proprietär & kostenpflichtig- Ressourcenschwer- Abhängigkeit von JetBrains-Lizenzmodell
Eclipse	<ul style="list-style-type: none">- Vollständig Open Source- Riesige Community & viele Plugins- Plattformübergreifend- Starke Java-Integration- Keine Lizenzkosten	<ul style="list-style-type: none">- Oberfläche altmodisch- Performance bei großen Projekten träge- Plugins teils unübersichtlich & qualitativ schwankend
NetBeans (Apache)	<ul style="list-style-type: none">- Vollständig Open Source- Plattformübergreifend- Gute Java- & PHP-Unterstützung- Maven, Git & CI integriert	<ul style="list-style-type: none">- Weniger aktiv entwickelt als Eclipse/IntelliJ- Oberfläche wirkt altbacken- Kleinere Community- Weniger moderne Features
Microsoft Visual Studio	<ul style="list-style-type: none">- Sehr mächtige IDE- Exzellente Debugging- & Profiling-Tools- Optimiert für .NET, C#, Azure, SQL Server- Umfangreiche Enterprise-Features (Architektur, Tests, CI/CD)- Große Community & Ökosystem	<ul style="list-style-type: none">- Proprietär, keine Open-Source-Variante- Hohe Lizenzkosten (außer Community Edition)- Sehr ressourcenintensiv- Volle Funktionalität nur unter Windows- Gefahr von Vendor Lock-in mit Microsoft-Ökosystem

Cloud Technologien

Vendor Lock-in Vermeidung

Cloud VMs & Containers

Feature	AWS	Google Cloud	Azure	Purpose / Benefit
Kubernetes (Managed Service)	EKS (Elastic Kubernetes Service)	GKE (Google Kubernetes Engine)	AKS (Azure Kubernetes Service)	Industry-standard container orchestration
Virtual Machines	EC2	Compute Engine	Virtual Machines	Standard compute abstraction; compatible with Terraform and other IaC tools.
Container Registry (OCI Compliant)	Amazon ECR	Artifact Registry / Container Registry	Azure Container Registry	Follows OCI standard, ensuring container image portability.

Cloud Storage & Datenbanken

Feature	AWS	Google Cloud	Azure	Purpose / Benefit
Object Storage (S3-Compatible API)	S3	Cloud Storage	Blob Storage (S3-compatible via SDKs/tools)	Widely adopted S3 API minimizes migration effort.
Relational Databases (PostgreSQL/MySQL)	RDS (PostgreSQL/MySQL)	Cloud SQL	Azure Database for PostgreSQL/MySQL	Open-source databases ensure data portability.

Cloud Observability & Monitoring

Feature	AWS	Google Cloud	Azure	Purpose / Benefit
OpenTelemetry Support	CloudWatch supports OpenTelemetry	Cloud Operations supports OpenTelemetry	Azure Monitor supports OpenTelemetry	Unified tracing and metrics across vendors.
Prometheus/ Grafana Integration	Managed Prometheus & Grafana	Managed Prometheus & Grafana	Managed Prometheus & Grafana	Portable open-source monitoring stack.
Log Export / Open APIs	CloudWatch Logs export	Cloud Logging export	Log Analytics export	Export to neutral targets for cross-cloud visibility.