THE ASF

Who Do You Trust? Open Source at Scale Security, Governance & Supply-Chain-Realität in der Apache Software Foundation

Dr. Richard Zowalla

12. OWASP Stammtisch Heilbronn, 28.01.2026

Whoami – Richard Zowalla

- Apache Software Foundation Member
- Committer / PMC in Apache Storm (Chair), Apache StormCrawler (Chair), Apache TomEE (PMC), Apache OpenNLP (PMC), Apache Geronimo, …
- 7+ Jahre in Open Source ;)
- Kein Security Background; Software Engineer
- Wiss. Mitarbeiter @ Fraunhofer IAO / KODIS & Hochschule Heilbronn

Disclaimer

Die in diesem Vortrag geäußerten Meinungen und Einschätzungen stellen meine persönliche Sicht dar.

Sie spiegeln weder eine offizielle Position der Apache Software Foundation (ASF) noch die Sichtweise meiner aktuellen oder früheren Arbeitgeber wider.

Ich spreche ausschließlich als Privatperson.

Agenda

1. Who Do You Trust?
2. Apache Software Foundation
3. Sicherheit in ASF-Projekten
4. Supply-Chain Risiken
5. Fazit & Takeaway

Who Do You Trust?

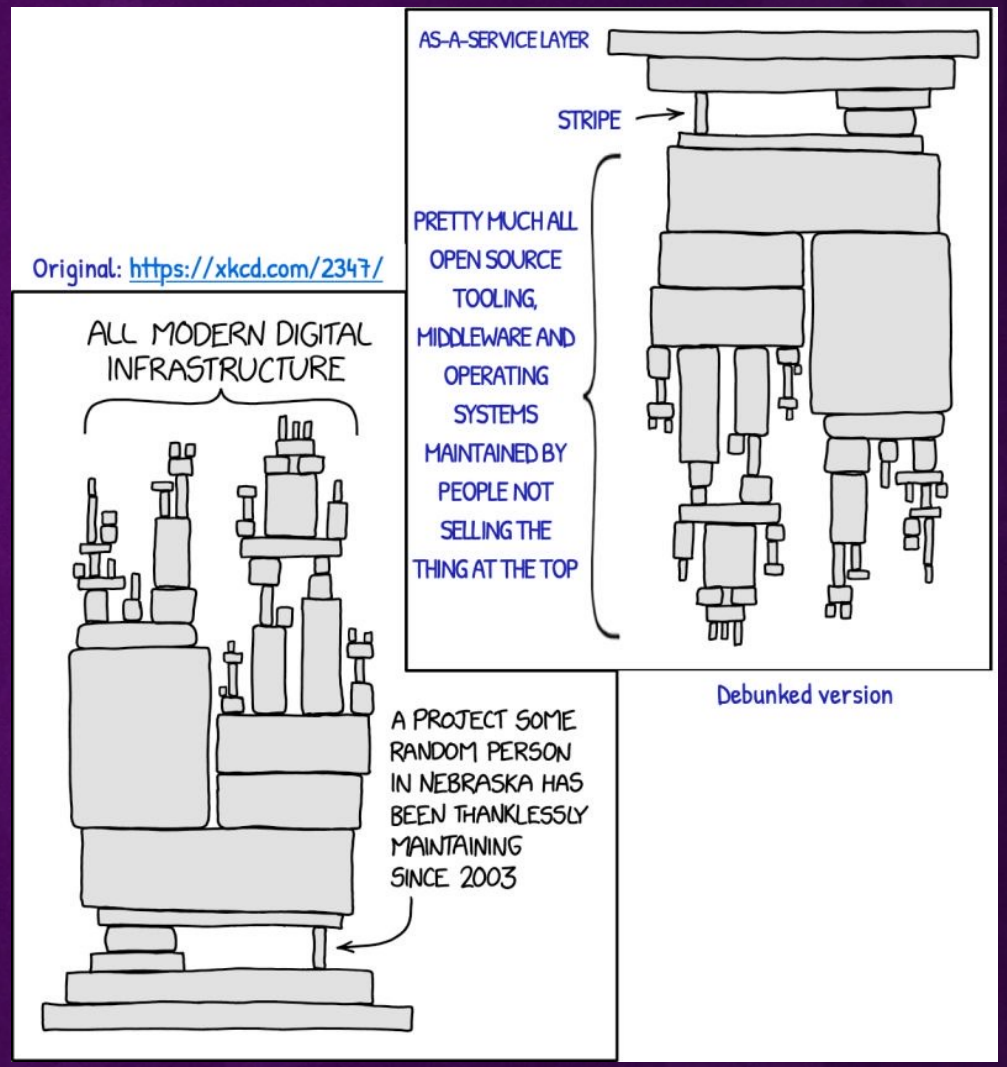
Wer von Euch vertraut Open Source Software?
Und warum (nicht)?

Who Do You Trust? (1/3)

- **Open Source Security & Risk Analysis Report** [1] aus 2025:

  97% (N=1.658) aller gescannten, kommerziellen Code-Bases enthalten Open-Source Komponenten.

- Gesamtvolumen des Open Source Software-Markts liegt nach einer Harvard Business School Studie bei ca. 8,8 Billionen USD [2].



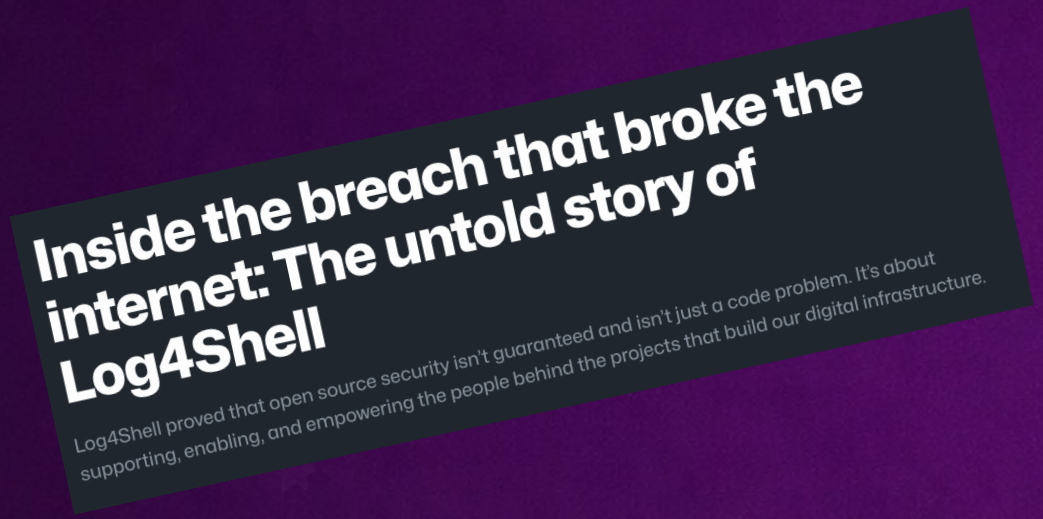
src: https://www.linkedin.com/posts/ylechelle_opensource-opensource-upsidedown-activity-7317317987990986752-clLW

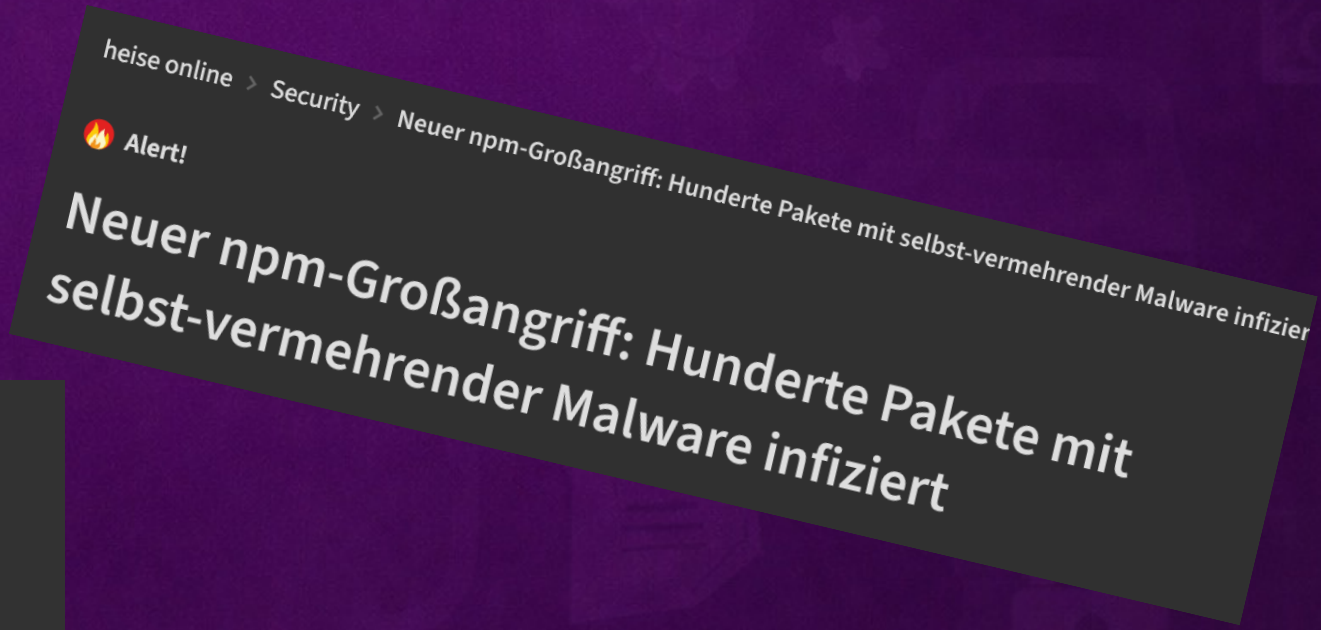
[1] https://www.blackduck.com/content/dam/black-duck/en-us/reports/rep-ossra.pdf
[2] https://www.hbs.edu/ris/Publication%20Files/24-038_51f8444f-502c-4139-8bf2-56eb4b65c58a.pdf

Who Do You Trust? (2/3)

- Open Source Software bedeutet aber (leider) oftmals:
  - Unbezahlte Entwickler*innen
    - Arbeiten finden als Hobby in der Freizeit statt
  - Kein Hersteller
  - Kein SLA
  - Diffuse oder unvollständige Lizenzwahl
- **Vertrauensfrage** (auch bzgl. Abhängigkeiten)

Who Do You Trust? (3/3)

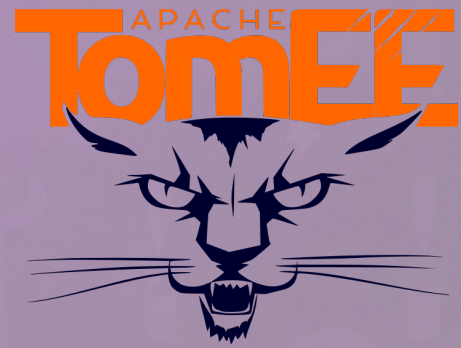




The ASF

Wer kennt die ASF?
Wer hat schon Software(bibliotheken) der ASF eingesetzt?

ASF-Projekte in modernen Infrastrukturen



und viele mehr …

Was ist die ASF?

- The ASF = The Apache Software Foundation
- Virtuelle, weltweite Open Source Organisation
- Community von Freiwilligen
- Formell: US 501(c3) charity (non-profit)
- Organisatorisches Rahmengerüst für 200+ Projekte

Die Mission der ASF

- **„Provide software for the public good“**

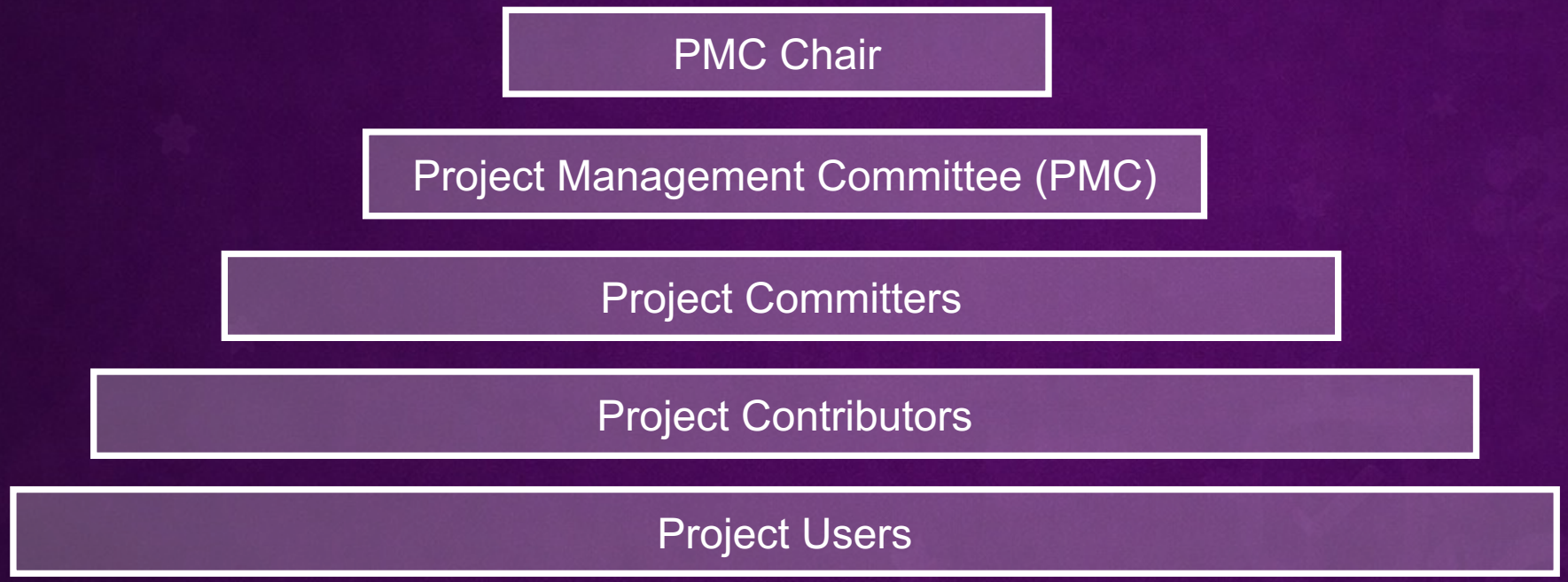
- **Schutz** individueller Personen vor Rechtsverfolgung im Kontext eines Projekts der ASF

- Bereitstellung einer Organisationsumgebung / Rahmengerüst für die **herstellerunabhängige** Entwicklung von Software

Projekte

- **Herz** der ASF sind ihre Top-Level Projekte (TLPs)
- ASF stellt als Organisation **zentrale Dienste**n bereit:
  - Infrastruktur (IDM, CI/CD, Mail, …)
  - Marketing, Legal & **Security**
  - Beratung und Erfahrung im Bereich **Governance** und **Community Building**

Struktur eines Top Level Projekts

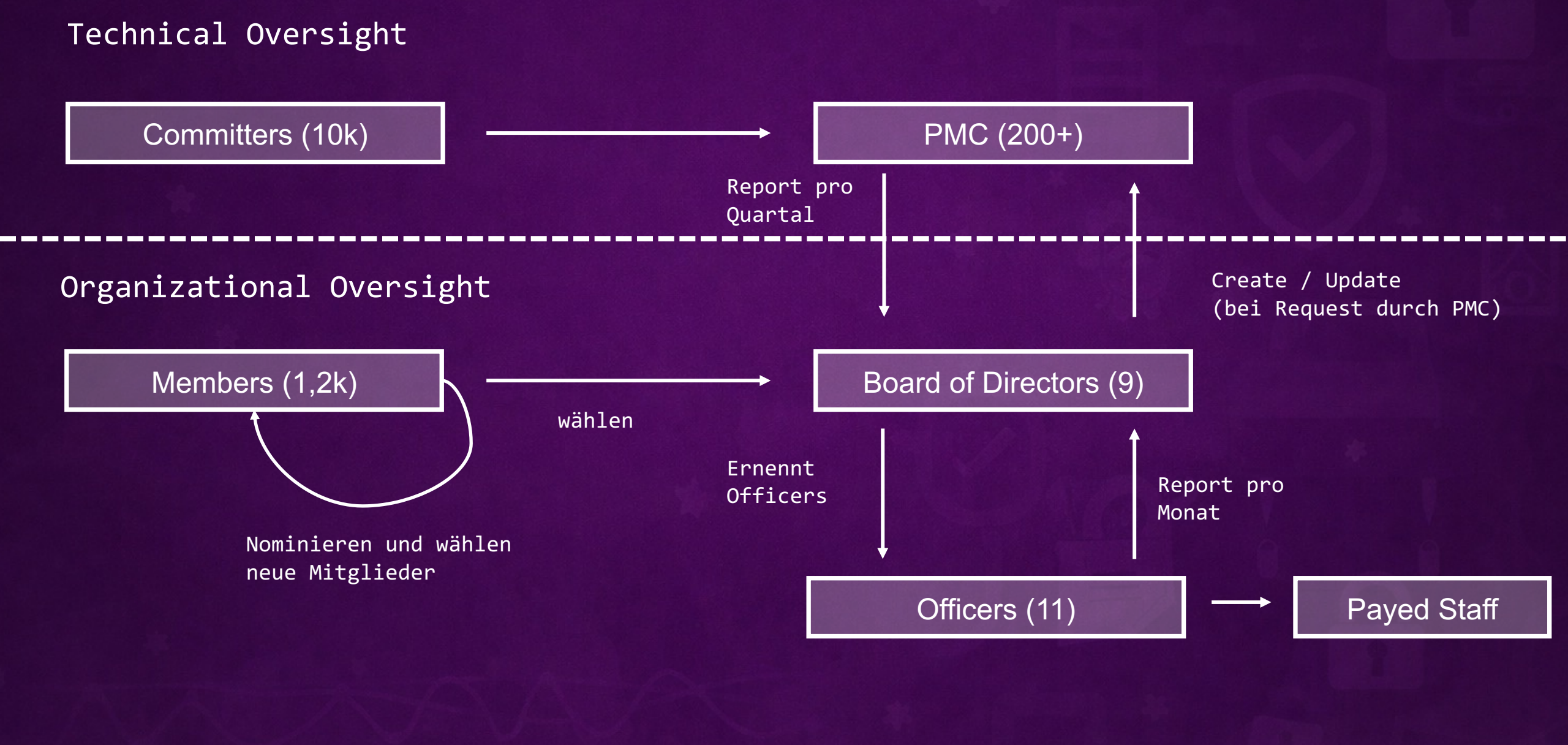


Total: 10k+

Total: 689k+

Total: Milliarden

Struktur der ASF



Wie funktioniert das?

- Methodik zur Entwicklung von Software
- Methodik zum Betreiben einer Community
- Methodik zum Betreiben einer Organisation

The Apache Way

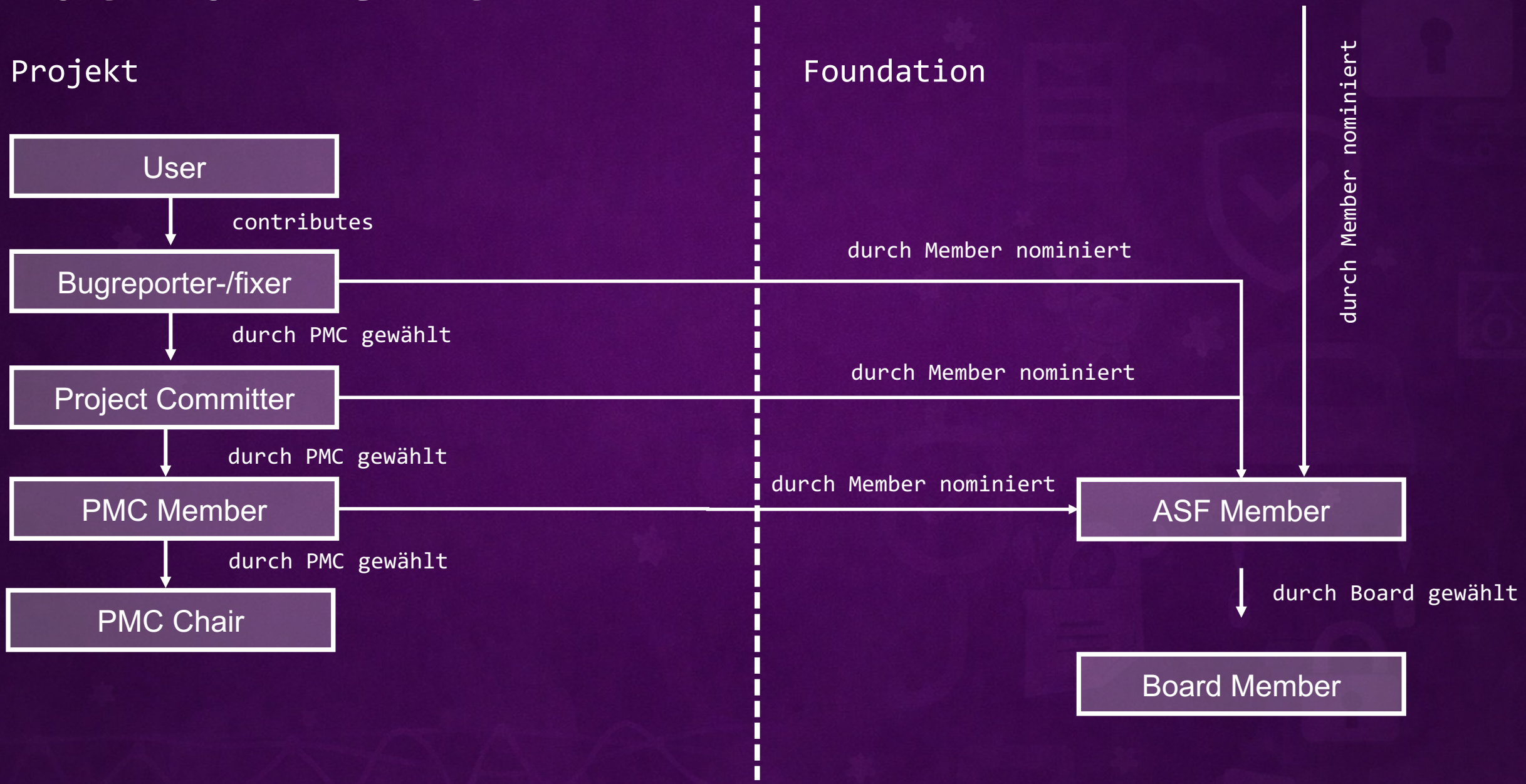
- **Leistung (merit)**
  - Verdientes Vertrauen: Aufstieg/Rechte durch Engagement

- **Transparenz**
  - Nichts geschieht im Verborgenen,
    alles ist öffentlich einsehbar (Mailinglisten, …)

  - „if it didn‘t happen on the list, it did not happen.“

- **Community**
  - Gemeinsam sind wir stark

Path of Merit



The Apache Way: Community over Code

- Alles basiert auf **Freiwiligkeit**
- Eine gesunde Community heißt neue Mitglieder willkommen, betreut sie und ermutigt sie beizutragen.
- Es ist normal, dass Mitglieder eine Community (vorübergehend) verlassen-
- Gesunde Community → besserer Code, langlebige Projekte

Entscheidungsprozesse

- Konsensentscheidungen:
  - PMC Member haben “binding votes“
  - Non-PMC Member haben „non-binding votes“

- Formal zählen **nur** „binding votes“, aber normalerweise werden „non-binding votes“ trotzdem respektiert.

- Etwas ist entschieden, wenn ein Vote
  - mind. 3 „binding votes“ erhält
  - mehr +1 als -1

Collaborative Development

- Code sollte als Community entwickelt werden
  - Große „Code-Drops“ sind schlecht…
- Development sollte transparent passieren (Git, Email)
- Votes setzten voraus, dass in einem Projekt mind. 3 PMC Member aktiv sind
- Votes dauern üblicherweise mind. 72 Stunden (Zeitzonen, …)
- Diskussionen und Votes passieren ausschließlich auf der Mailingliste des Projekts.

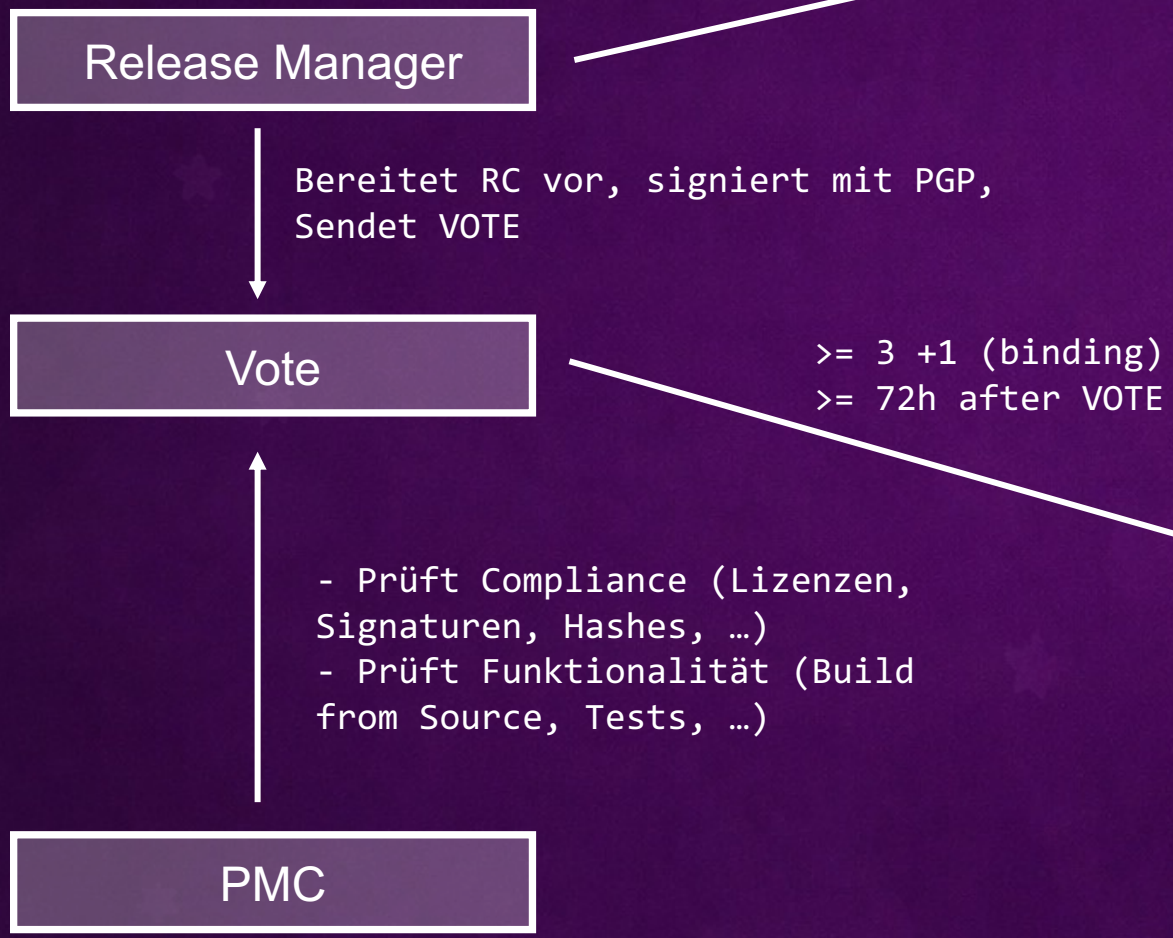
Collaborative Development

- Jeder Committer hat **Schreibrechte** auf der gesamten Codebase
  - „with great power comes …“
- Commit Policies unterscheiden sich je nach Projekt:
  - Review than Commit (RTC)
  - Commit than Review (CTR)
- **Oversight/Verantwortung** liegt beim **PMC des Projekts**, d.h. auch
  - Softwarequalität
  - Sicherheitsaspekte
  - …

Sicherheit?

Wer denkt, dass es eine Pflicht eine MFA gibt?

Release Prozess (1/3)




```
pub   1024D/A46C4CA1 2006-01-05
      Key fingerprint = 9056 B710 F1E3 3278 0DE7  AF34 CBAE BE39 A46C 4CA1
uid                  Matt Hogstrom <hogstrom@apache.org>
sub   2048g/2FD8C3E0 2006-01-05

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2.0.19 (FreeBSD)

mQGiBEO9oSkRBADW6zrFzi3FNVApHusdx/vNRkoCqY6DF0aOKa6fuia3GMt9Cu5z
LmZ2727uSg3ovojI79bcUARatQGPhntfJcdTHZJGn616oEbORBAB8ZyvUON9OkaB
D0oXnmXidJ3QUS93tOdEuDBlQS4B3Lh29zwZgxyfcLFnKqA2DkSiEz+9PwCg28cq
yPVvdghM9blcqoHOtPP/QEUEALTtl+HjseQjoZowPp8aiS1U9G486HovtYxJjtTJ
0Tpdpa3Tt0jqwQqjMQRpoFP67/osAB38wp7GUk+dSKrgmLLXNc0VhsviN5Wv7M+n
Qqb21/OuE+/VY42lFL9karLmQ7wKYFuwe94t/Qtvg07PDAjG+78VU34GJg0FzN81
rKqXA/0bGpkf3zl4z+dR8Kzaz25HymsmfV4EojQ4zsKZUsS4KvvpwcN3EFjHR3E0
aQ+NFhvlu4b0Mspk1jfxdDSYDg1YFiNStChGeE+1boMILDTjxz21zbpvPsPCLQ13
KZU1/ZuKvjRz9ExjlxDTL+n8PCV3B0eaPBtdKlu7BXXOPf6Pg7QjTWF0dCBIb2dz
dHJvbSA8aG9nc3Ryb21AYXBhY2hlLm9yZz6IRQQQEQIABgUCRKLcZwAKCRD1wmAW
```

https://dist.apache.org/repos/dist/release/<TLP>/KEYS

Single Source of Truth 🤡
Keys ggf. ohne Ablaufdatum (Legacy) 🤡

Achtung: Es geht hier nur um das Source Artefakt. Alles andere (Maven Artefakte, Python Wheels, Docker Images, Binaries, etc.) gelten als Convenience Artifacts und sind strenggenommen nicht Gegenstand des Votes.

Unterscheidet sich von TLP zu TLP 🤡

Release Prozess (2/3)

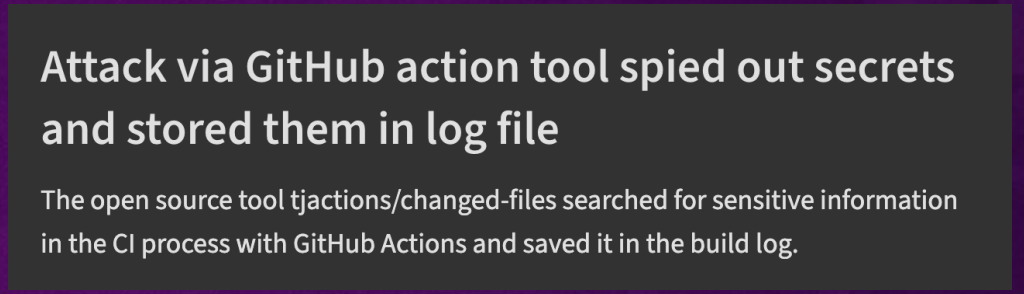
- Grundprinzip beruht auf dem Apache Way: „Merit & Trust“
- Kompromittierung durch Social Engineering möglich (vgl. xz)
- Es gilt zwar 6-Augen Prinzip für einen Release, aber trotzdem ist es z.B. möglich, dass man Maven Pakete alleine ausrollt (weil “Convenience“ und daher nicht von Governance abgedeckt).
  - Es lohnt sich zu prüfen woher/wer die Convenience Artefakte für eine spezifische Plattform bereitstellt
  - Spoiler: Nicht immer das Projekt/ASF

Release Prozess (3/3)

- **Apache Trusted Releases (ATR)** als Anforderung aus dem Cyber Resilience Act (CRA)
  - https://github.com/apache/tooling-trusted-releases
- Ziel:
  - Reproduzierbarkeit von Builds aus Source Code
  - Automatisierung des Prozesses
  - Verifikation der erzeugten Artefakte

GitHub & GitHub Actions

- Zentrale ASF-weite Allowlist für GH Actions
- Seit den Supply-Chain Angriffen über Tag Overrides, wird Pinning der Hashes zentral enforced.
- **Trotzdem**: Wenig Regulation, unterscheidet sich von TLP zu TLP.



SCA / OWASP Tooling / Security Scanning

- Je nach TLP werden entsprechende Tools im Build Prozess in CI/CD eingesetzt.

- Build = Teil der TLP-Goverance --> nicht reguliert
  - Hinschauen lohnt sich.

- Initiativen in Richtung Reproducable Builds, SBOMs (cyclonedx) sind gestartet, aber noch keine weite Verbreitung in allen TLPs.

  - aktuell v.a. in Apache Logging (log4j)

  - u.a. Finanzierung vom Sovereign Tech Fund

CVE-Prozess

- Geht im Wesentlichen zentral über security@apache.org
  - Security Team triagiert und gibt weiter an PMC.
  - Security Team kümmert sich auch um Follow-up mit PMC
  - Unterstützt bei Klassifikation, etc.
- Bearbeitung und Rückmeldung ist Aufgabe des betroffenen PMC
  - Heterogenität führt dazu, dass das auch manchmal dauern kann…
- **Achtung:** Manche PMC haben eine eigene security@ Liste…

Help Needed!

- IT Security Know-How ist eher unterpräsentiert.

- Mithilfe ist immer erwünscht – bringt Euch ein!
  - Jedes Code Review eines Pull Requests hilft.
  - Jede Verbesserung in Security Hinsicht hilft, auch im Bereich CI/CD, SCA, etc.

- Achtung: Anwendungsfall der Software sollte jedoch klar sein; automatisierte Meldungen über SCA, die aber nur hypothetische Probleme aufdeckt, sind meistens Zeitverschwendung (für die Entwickler*innen des TLP).

MFA?

- Aktuell **keine** MFA-Pflicht für ASF-Systeme (Git, SVN, …) 🤡
- MFA-System auf Basis von Authentik im Aufbau; soll 2026 kommen.
- Code-Systeme sind gemirrored (GitHub <-> ASF eigene Systeme)
  - Für Rechte auf GitHub muss MFA zwingend aktiviert sein.
  - Direktes Schreiben auf ASF-Systeme erfordert **kein** MFA 🤡 🤡

Supply-Chain-Risiken

Erwartung vs. Realität

Drum prüfe wer sich ewig bindet…

- Governance Mechanismen **senken**
  - Risiko für Sabotage,
  - Fehlfunktion oder
  - inkorrekte (Sicherheits)fixes.

- **Trust** und **Transparenz** als zentrale Konzepte
  - Traue ich einem einzelnen Maintainer
  - oder einer Gruppe von Personen mit klaren Prozessen

- Nutzender bleibt **immer** in der Verantwortung:
  - Verifikation (Hashes, Signature, …)
  - Source of Origin

WHO DO YOU TRUST?

FAZIT

Fazit & Takeaway (1/2)

- **Governance** ist **Kern** der Sicherheitsstrategie der ASF.
- **Vertrauen** in OSS entsteht nicht durch SLAs, sondern durch **Transparenz** und **klare Prozesse**.
- Die eigene **Erwartungshaltung** bei der Kommunikation mit OSS-Projekten ins richtige Licht rücken:
  - Meistens unbezahlte Freiwillige, die Bibliotheken/Software in ihrer Freizeit betreuen.

Fazit & Takeaway (2/2)

- Ein Blick in die eigene SW-Abhängigkeitskette ist hilfreich und sollte zwingend in den eigenen SWE-Prozess integriert werden (SBOMs, Lizenzen, …)

- Freiheit schaffen, dass sich Entwickler*innen in der Arbeitszeit mit OSS beschäftigen können.
  - Aufbau von (tiefer) Expertise
  - Möglichkeit Bugs/Features/Probleme schneller zu beheben

- Viele Möglichkeiten beizutragen – gerade im Bereich IT-Security.
  - Traut euch – wir beißen nicht!

Vielen Dank!

FRAGEN?

Kontakt: rzo1@apache.org