

ATIX

The Linux & Open Source Company

Welcome

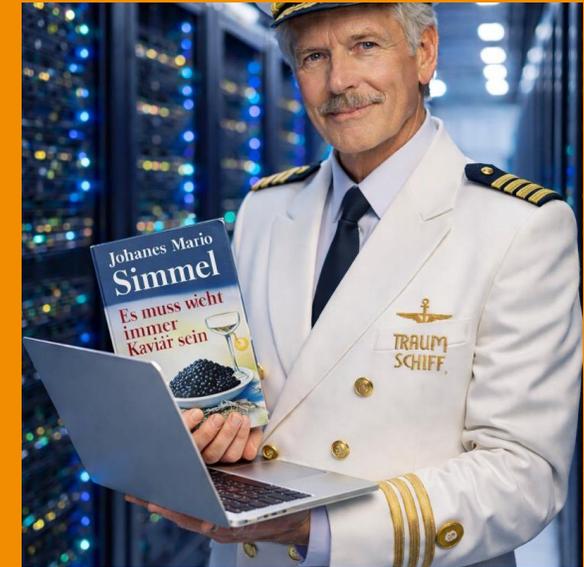




The Linux & Open Source Company

It Doesn't Always Have to Be Caviar:

Enterprise Alternatives and the Challenge of Default Stacks



About me

Amir Zahirovic

- 5+ years of experience
- DevOps/Platform Engineer
- AWS certified, CKA & CKS
- Kubernetes ecosystem



About me

Gergely Szalay

- 15+ years in OPS and DevOps
- Linux, Application Support
- Kubernetes, Container

- 2 Children
- Reading, Series
- Cats





Agenda

- What is this with the caviar?
- How did we find the topic?
- The setup we have delivered
- The mindset behind IT decisions
- The caviar metaphore
- The caviar problem
- Enterprise alternatives

How much did your last enterprise software cost?
Now ask yourself: how much of it did you actually
use?

The caviar metaphor

Caviar:

- Expensive & status driven
- Chosen for reputation, not taste
- Small portions (underutilized)
- Signals sophistication



How did we find the topic?

The case with the customer:

- Kubernetes: Cloud or on-prem? Online or air-gapped?
- Image registry and Git
- Containers: Complexity and Security
- Data: aggregation or database?



How did we find the topic?

The actual setup:

- Kubernetes: on-prem, via KubeSpray
 - o Straightforward configuration and recovery
- Image registry and Git: Gitea
 - o Small footprint
 - o Easy maintenance
 - o CI/CD



How did we find the topic?

The actual setup:

- Containers: Podman
 - o No daemon
 - o Rootless setup
 - o Clarity in logs
 - o Built-in Kubernetes approach
 - o Docker compatibility

How did we find the topic?

The actual setup:

- Data: PostgreSQL and TimescaleDB
 - Robust SQL AND NoSQL-like time-series database
 - SQL experience: backup, security
 - Effective data handling, smaller footprint
 - Grafana Integration

The mindset behind the scenes

Why?

"How much did your last enterprise software purchase cost?"

"How much does it cost, solving someone else's problem?"

- The way to solve our problems
- The price we pay

The mindset behind the scenes

"Can I find the right solution?"

"Where does this come from?"

- Search the interwebs, and find always the same answers
- Copy-paste or own experience?
 - *Five reasons to do some obscure thing I'm suggesting*
 - *Five reasons to do the exact opposite*
 - *The very same list about that certain topic, again*
 - *Why you should stop being rational, right now*
- Click-baiting vs. useful help?
 - *Human or AI?*

The mindset behind the scenes

Lightweight Linux for Old Computers ▾

write me an article in the style of a nerd-oriented tech site, suggesting five lightweight Linux distributions for weak/dated computers. Give it a click-baiting title, please.

◆ Gedankengang anzeigen ▾

TITLE: Your Potato PC is Begging for Mercy: 5 Lightweight Linux Distros to Resurrect That Ancient Rig

Let's be real. You have one.

Maybe it's a Core 2 Duo ThinkPad collecting dust in the closet beneath a pile of obsolete micro-USB cables. Maybe it's that netbook from 2010 that you swear you're going to turn into a dedicated Pi-hole controller someday.

Right now, it's useless. It probably tries to boot Windows 7, hyperventilates for twenty minutes, and then hard locks when you open a second Chrome tab. Modern operating systems—looking at you, Windows 11, you demanding bloatlord—have absolutely zero respect for limited clock cycles and meager RAM allotments.

But we don't send silicon to the graveyard just because it can't run Electron apps smoothly. We are enthusiasts. We optimize. We deploy the penguin.

If you want to squeeze every last drop of compute power out of aging hardware, you need to ditch the cruft and embrace minimalism. We're talking about sub-1GB RAM usage at idle, Window Managers instead of Desktop Environments, and kernels stripped bare.

Grab that dusty laptop, find a USB drive, and prepare to flash. Here are five lightweight distributions that will make your potato PC feel surprisingly crisp.

1. Linux Mint Xfce Edition

The Vibe: The gateway drug for recovering Windows users.

Why It Rocks: If you want stability without needing to memorize `man` pages just to connect to Wi-Fi, start here. Linux Mint is renowned for its polish. While the standard Cinnamon edition is too heavy for truly weak hardware, the Xfce edition is the sweet spot.

Xfce is a fully-featured Desktop Environment (DE), meaning you get a panel, a start menu, and

Frag Gemini 3

+ 🧰 Tools Thinking-Modus ▾ 🗣️

The mindset behind the scenes

"Enterprise questions, enterprise answers?"

- Does it solve *your* problem?
- Does it match *your* challenges?
- Does it consider *your* situation?
 - Do you have the same setup?
 - Do you *know* those applications?
 - Aren't they *too* clever for your needs?
 - Do they match your compliance rules?
 - Are you a startup?

The mindset behind the scenes

The dangers of low hanging fruits

- Do you understand what you're doing?
- Do you implement products or logic?
- Do you lock yourself to something you don't even need?
- Can you trust those solution providers?
- Buzzwords and clichés vs. the right solutions for your very own challenges
- Security: are you providing the typical surfaces to attack?



Back to the metaphor

Caviar and enterprise

The current state

The enterprise default playbook:

- When in doubt, buy the biggest brand
- "Enterprise ready" over "actually suitable"
- Complexity = sophistication
- More features = better value



THE CAVIAR PROBLEM

Fear-based decision making

- "Nobody ever got fired for buying IBM/SUN/..."
 - Risk aversion over optimal choice
 - Compliance theater
 - CYA culture in procurement

Sales & procurement dynamics

- Vendor relationships influencing technical decisions
- Enterprise sales teams selling dreams
- Long-term contracts locking in pricing
- Buying for your company or their sales department?

The real costs – hidden impact

Hidden costs:

- Development velocity slowdown
- Vendor dependency limiting innovation
- Over-complicated architecture
- Team frustration & talent loss
- Opportunity cost



ThePhoto von PhotoAuthor ist unter CCYSA lizenziert.

Challenges of default stacks

Technical Debt Accumulation

Following best practices from 2010:

- Legacy patterns that made sense... then
- Architecture decision based on old constraints
- "This is how enterprises do it"



ThePhoto von PhotoAuthorist unter CCYSA
lizenziert.

Over-engineering simple problems

Using a tank to kill a mosquito:

- Implementing enterprise patterns for simple use cases
- Six-month projects for two-week problems
- Ceremony > Substance



Vendor Lock-in

"We can't change because..."

- Proprietary APIs and formats
- Specialized skills not transferable
- Data migration nightmares
- Integration dependencies



Performance and bloat

"Why is everything so slow..."

- Generic solutions trying to serve everyone
- Unused features consuming resources
- Heavyweight frameworks
- The "Swiss Army chainsaw" problem



Skills and hiring challenges

The talent problem:

- Niche enterprise skills limit hiring pool
- Younger developers prefer modern stacks
- Training time for complex systems
- Legacy stigma



ThePhoto von PhotoAuthor ist unter CCYSA lizenziert.

ENTERPRISE ALTERNATIVES



Better solutions *do* exist

Open-Source Maturity

Enterprise-grade doesn't require enterprise licensing

- PostgreSQL vs closed source solutions
- Kubernetes vs proprietary orchestration
- Kafka vs proprietary messaging
- Grafana/Prometheus vs commercial monitoring

Cloud-Native Approaches

Rethinking architecture for the cloud era

- Microservices/monoliths
- Serverless
- API-first design



The Pragmatic Stack

Example modern enterprise stack:

- PostgreSQL + Redis (data)
- Kubernetes (orchestration)
- API gateways (Kong, Ambassador)
- Observability (Grafana, Prometheus)
- CI/CD (GitLab, ArgoCD)
- *Total licensing cost: \$0*



ThePhoto von PhotoAuthorist unter CCYSA lizenziert.

Making better decisions

Start with actual needs

Requirements-first thinking:

- What problem are we actually solving?
- What's the simplest solution?
- What are our actual constraints?
- What do we need vs want?



Total Cost of Ownership

Beyond the sticker price:

- Licensing + maintenance (obvious)
- Implementation services
- Training and learning curve
- Operational overhead
- Opportunity cost and tech debt



ThePhoto von PhotoAuthor ist unter CCYSA lizenziert.

When premium IS Worth IT

Legitimate reasons for enterprise solutions:

- Strict regulatory requirements (not always)
- Truly unique capabilities
- Scale beyond OSS maturity
- Risk tolerance and appetite
- Available expertise



Conclusion

Remember:

- Question expensive defaults
- Evaluate actual needs first
- Open-source is enterprise-ready
- Total cost includes more than licensing
- Flexibility and control have value
- Sometimes, the best solution isn't caviar



Questions?

We're happy to answer

Thank You



+ 49 (0)89 452 3538-0



info@atix.de



atix.de

Follow us!

