



Joomla!®
WORLD CONFERENCE

TUF for extension developers

extending
The Update Framework (TUF)
from Joomla! CORE to third-party Extensions

Rüdiger Schultz
Schultz IT Solutions
Vienna, Austria

Schultz it Solutions
swiss servicequality for
austrian and international enterprises

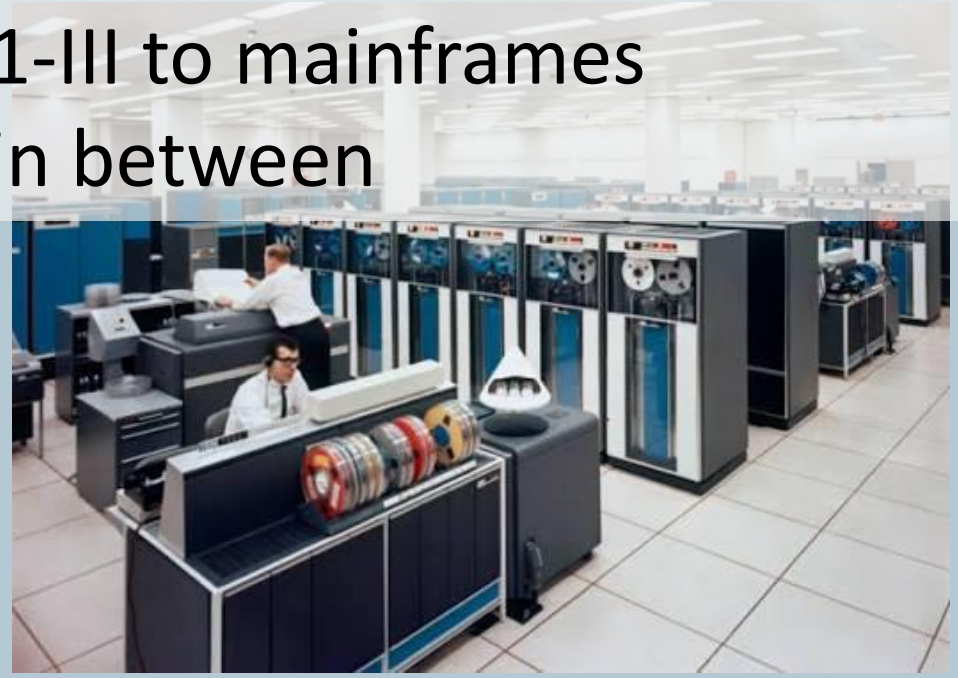


Rüdiger Schultz

Schultz IT Solutions

Building with Joomla since 2009 — this project sits on ~50 years of writing software, from the TI-51-III to mainframes and almost everything in between





TUF in Joomla CORE since 5.1 automated since 5.4/6.0

Joomla 5.1 introduced signed, later automated Core updates for minor, patch and security releases, allowing Joomla installations to be kept up to date reliably and without manual intervention – an important step towards improving the security and maintainability of Joomla websites.

Automated Core Updates use The Update Framework (TUF), a secure software update distribution system designed to prevent supply chain attacks when automatically updating Joomla core.

Schultz it Solutions
swiss servicequality for
austrian and international enterprises

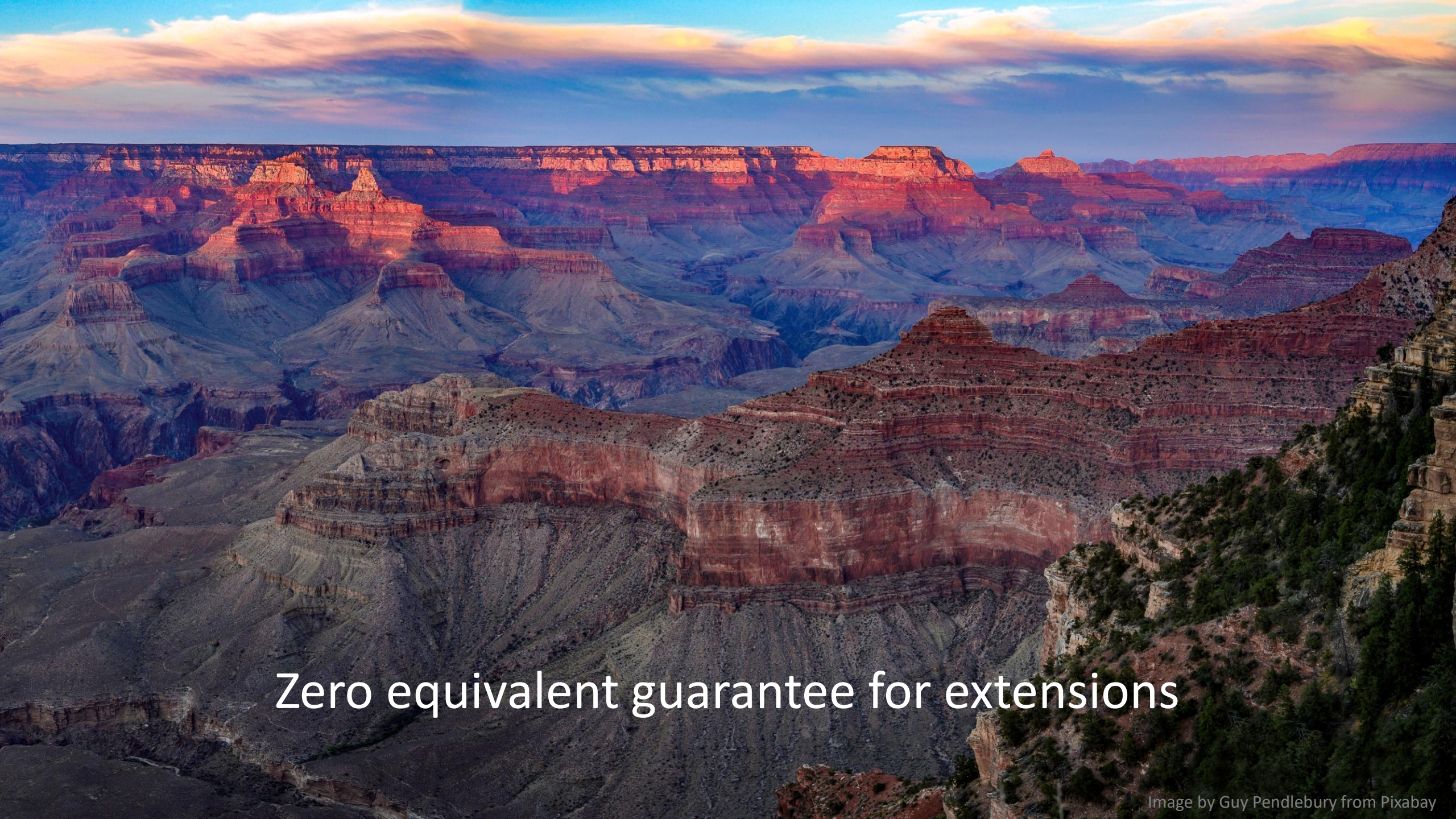


Image by Guy Pendlebury from Pixabay

It has happened before

June 2024 — WordPress.org

- **5 plugins backdoored** (e.g. Social Warfare, Blaze Widget)
- **35,000+ sites** received the update through the normal channel
- malicious code created rogue admin accounts and injected SEO spam
- **root cause:** compromised developer passwords

With signed packages

- **a stolen password alone ships nothing** — the signing key is separate from the account
- sites verify every update before installing; a wrong or missing signature is refused
- a compromised update server can't push code either
- Joomla core has this today — extensions don't

Sources: Wordfence and BleepingComputer, June 2024

Schultz it Solutions
swiss servicequality for austrian and international enterprises

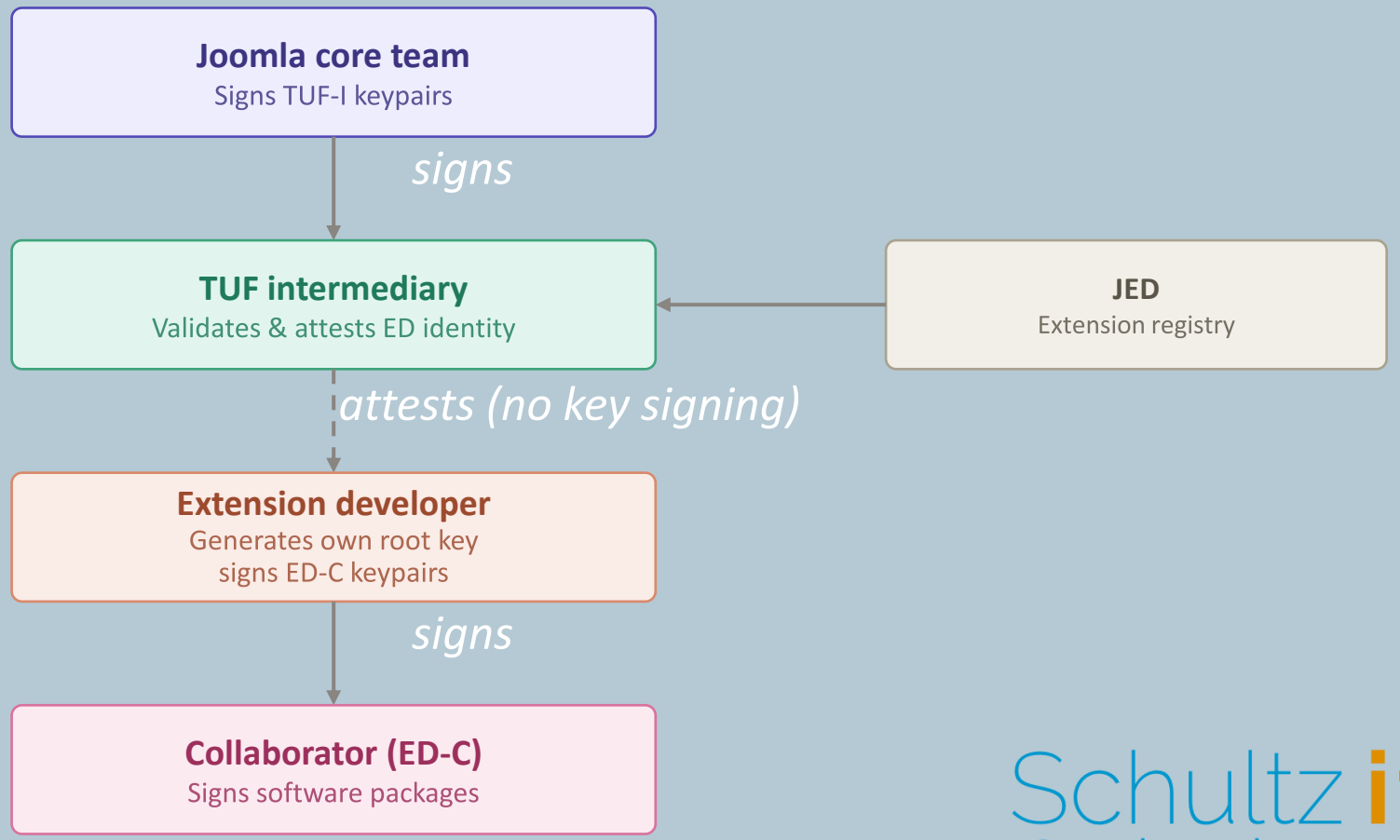
TUF for Extensions

TUF for Extensions (TUF4E) brings signed, automatically updatable packages to third-party extensions — without making Joomla core the bottleneck for every developer.

This builds on reusing the TUF client/library Joomla core already ships — with a small core change to the extension updater.

Schultz it Solutions
swiss servicequality for
austrian and international enterprises

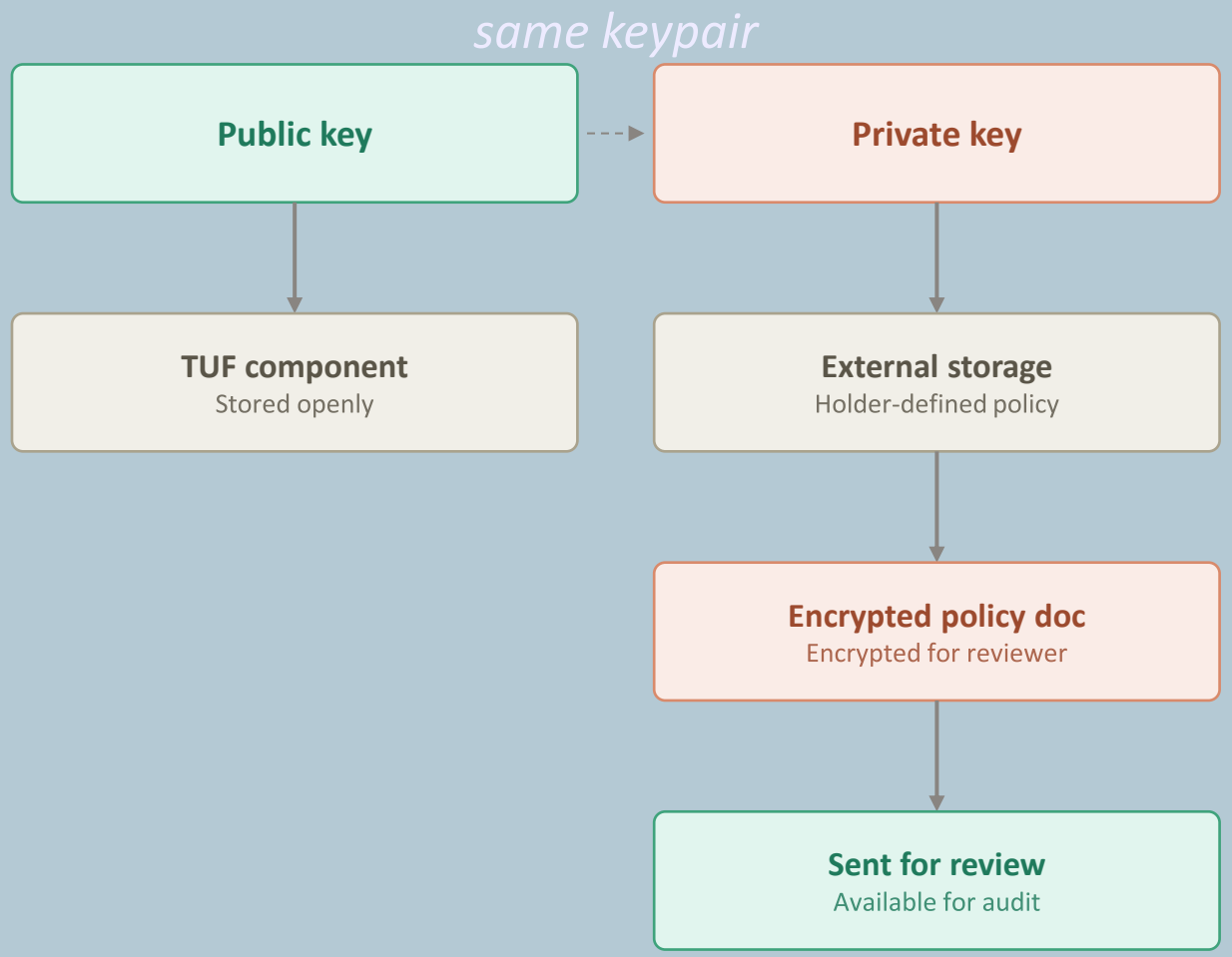
The Trust Model



TUF-I : TUF Intermediary
JED : Joomla Extension Directory
ED : Extension developer
ED-C : Collaborator (team member) with ED

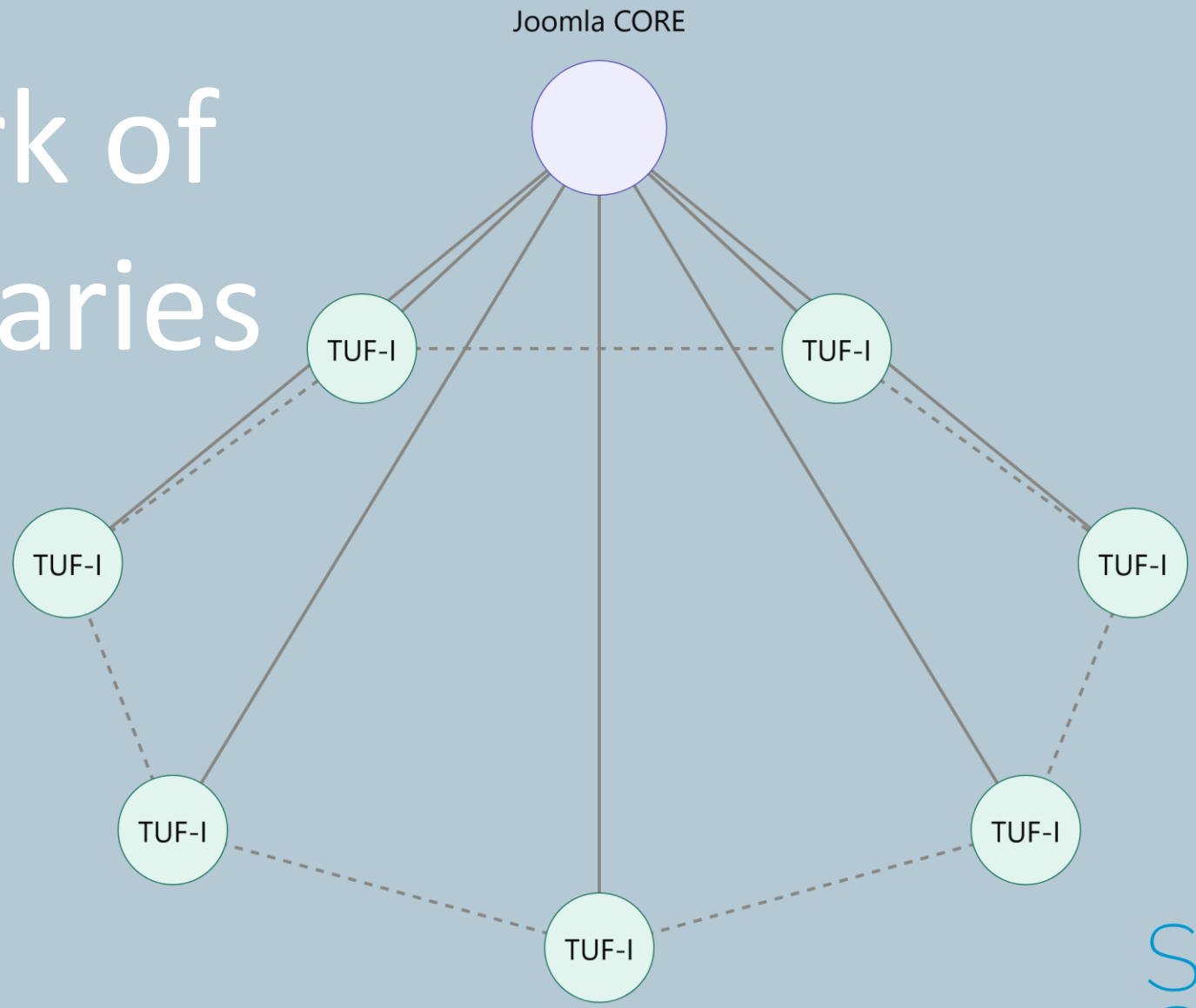
Schultz it Solutions
swiss servicequality for
austrian and international enterprises

Where public / private keys live



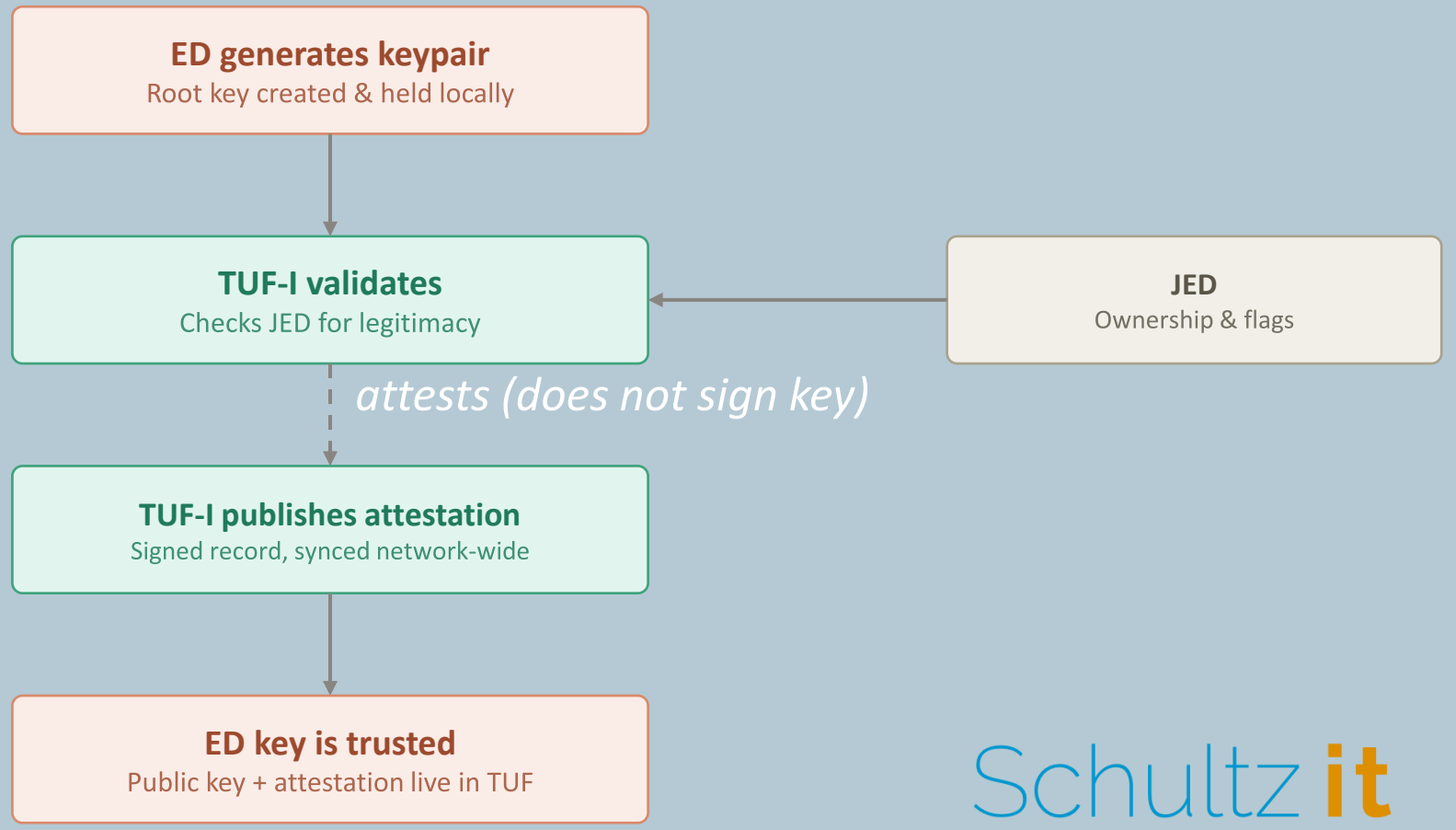
Schultz it Solutions
swiss servicequality for
austrian and international enterprises

A network of intermediaries



Schultz it Solutions
swiss servicequality for
austrian and international enterprises

ED registration and attestation



ED : Extension developer
TUF-I : TUF Intermediary
JED : Joomla Extension Directory

Schultz it Solutions
swiss servicequality for
austrian and international enterprises

The role of JED

The Joomla Extension Directory (JED) provides the baseline to check the validity of the Extension Developer (ED)

- does ED exist in the directory?
- are there any “red flags” regarding this ED?
- which extensions does ED own?
- do all extensions have a working update server?
- …

Schultz it Solutions
swiss servicequality for austrian and international enterprises

Two components, not one

com_tuf_i — the TUF-Intermediary side

- operated by a TUF-I
- validates ED against the JED
- attests ED identity (issues the signed attestation record)
- publishes & syncs the attestation registry with other TUF-I

com_tuf_ed — the Extension-Developer side

- operated by the ED
- generates & holds the ED's own root key
- issues signed ED-C keypairs
- used by ED-C to sign software packages

Schultz it Solutions
swiss servicequality for austrian and international enterprises

What changes for you

Extension developer

- **once:** create your ED key in com_tuf_ed, get it attested by a TUF-I
- **per team member:** issue an ED-C key
- **per release:** sign the package, publish the signed metadata (process TBD)
- **unchanged:** your build, your update server, your JED listing

Site owner

- **nothing to install:** core's TUF client does the checking
- **signed extensions:** verified before every update; tampered packages are refused
- **Auto-update is opt-in**: per extension – you choose which EDs you trust.
- **Everything else:** updates as today

Schultz it Solutions
swiss servicequality for
austrian and international enterprises

Planned vs. under discussion

Planned approach

- reuses Joomla core's TUF client — no separate updater
- federated trust model (ED holds own root key)
- TUF-I admission must be a JED-validated ED
- component split com_tuf_i / com_tuf_ed
- Key custody policy
- ED vetting with JED support

Ongoing discussions

- package signing process
- automatic update process
- TUF-I ↔ TUF-I sync protocol

Schultz it Solutions
swiss servicequality for austrian and international enterprises

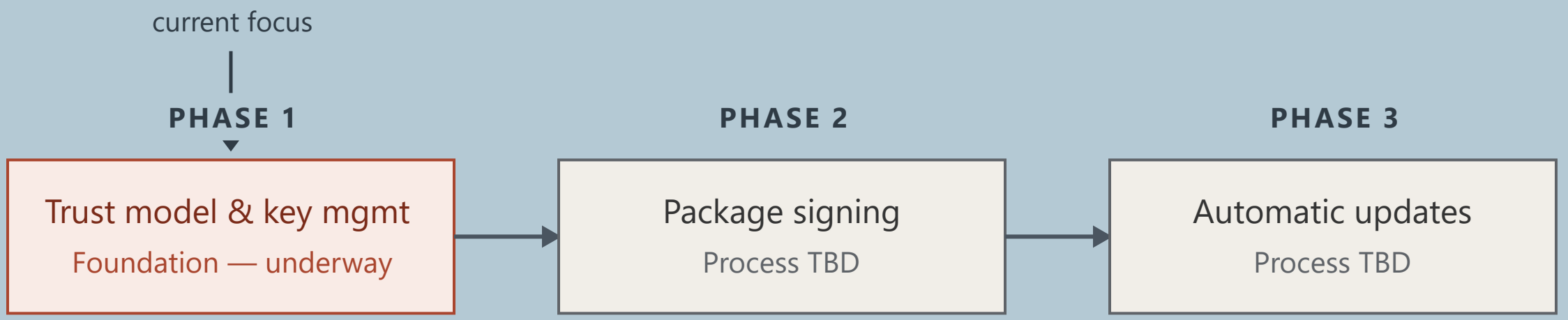
Open questions / your ideas

PHP-TUF
Auto-update opt-in
Commercial extensions
Key custody
Revocation
TUF-I governance
Drupal?
Backend UX
Sync protocol
Transparency log
Core changes
Liability
Metadata hosting
Who runs a TUF-I?
Incentives
Trust on first use
Signing workflow
Rollback attacks
Timestamp role
Key compromise
Key rotation
HSM?
Migration path
Attestation
Threshold signatures
Joining rules
Package signing
Non-JED extensions
ED-C offboarding

Mattermost: joom.la/chat
email: it-solutions@schultz.ch

Schultz it Solutions
swiss servicequality for austrian and international enterprises

Roadmap



colored = current focus · gray = planned, scope still open

Schultz it Solutions
swiss servicequality for
austrian and international enterprises

What comes next?

The TUF for Extensions is not a product yet, we are looking for your input:

- pressure-test the federated trust model.
  Does it hold?
- weigh in on the open questions
  (package signing, sync protocol, automatic updates)
- tell us if you'd want to run a TUF-I
  (once there's something to run)

Mattermost: joom.la/chat
email: it-solutions@schultz.ch

Schultz it Solutions
swiss servicequality for
austrian and international enterprises



Rüdiger Schultz

Software has been part of almost my entire life — my first step at creating Software was a TI-51-III pocket calculator, I later worked on IBM 3090 mainframes, with just about everything in between throughout my career.

I do learn something new every day, which is probably why I still enjoy developing software solutions. I live in Vienna with my two adult sons nearby.

Mattermost: joom.la/chat
email: it-solutions@schultz.ch

Schultz it Solutions
swiss servicequality for
austrian and international enterprises

Appendix

Backup topics from the design discussions — for Q&A, time permitting

Schultz it Solutions
swiss servicequality for
austrian and international enterprises

Why not classic PKI?

Classic X.509 / CSR PKI — not used

- certificate authorities, CSR issuance, certificate chains to validate and renew
- heavyweight machinery for a solo or small-team extension developer to operate
- OpenSSL kept only as a fallback, if X.509 interop is ever needed

TUF's own delegation model — chosen

- native TUF metadata and roles — no certificate chain to issue or renew
- signatures via PHP + libsodium (Ed25519)
- matches the federated trust model already in place (ED holds its own root key)

Schultz it Solutions
swiss servicequality for
austrian and international enterprises

Guarding the TUF-I authority key

In practice

- held in a dedicated KeePass database, kept separate from everyday credentials
- lives only[1] on a fully VeraCrypt-encrypted machine, protected by up-to-date endpoint protection
- loaded only when needed — attesting a new ED key or rotating — never kept resident in a server process

- [1] a VeraCrypt-encrypted backup copy sits offline in a bank safe-deposit box

Considered, not decided

- a separate OS user account just for authority operations — extra isolation, not pursued further
- a hardware security module (HSM) — more assurance than a solo or small TUF-I operator needs today
- a separate VeraCrypt-encrypted container only for the KeePass database

Schultz it Solutions
swiss servicequality for austrian and international enterprises

Can a TUF-I attest its own ED key?

Rule: a TUF-I cannot attest its own operator’s ED key

Prevents a TUF-I from vouching for itself when its operator is also an Extension Developer

If an ED also runs a TUF-I

- that TUF-I cannot sign or attest its own operator's ED key
- closes an obvious conflict-of-interest loophole

They must use a different TUF-I

- any other TUF-I in the network can validate and attest them
- keeps the attestation independent of the applicant

Schultz it Solutions

swiss servicequality for austrian and international enterprises