Usage of ISAR in the EB corbos Toolkit

Dr. Oliver Pajonk

October 5, 2026



About me

- 20 years of (more or less) professional experience in software development, across industries and roles.
- About 10 years in automotive.
- Started my "OSS career" with an ancient Linux flavor, S.u.S.E. Linux 4.3.
- Principal Engineer in the EB corbos Linux for Safety Applications Team.



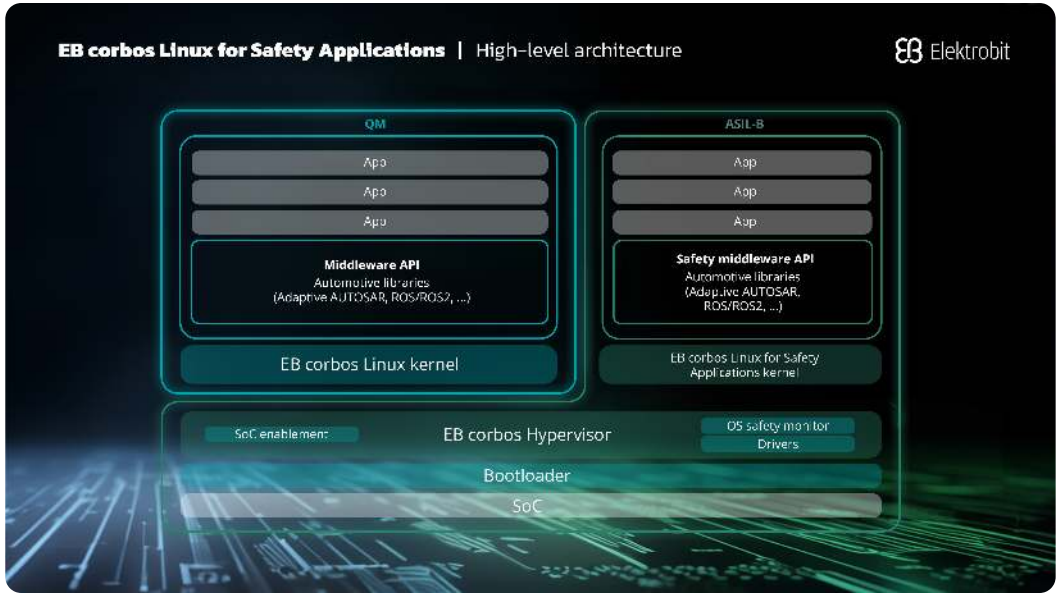
© Elektrobit 2026

Elektrobit builds automotive software

Elektrobit develops embedded software for automotive ECUs and software-defined vehicles.

EB corbos Linux for Safety Applications

- Adds a hypervisor-based safety layer to open-source Linux.
- An OS safety monitor intercepts and validates kernel actions, supporting high-integrity workloads alongside QM applications.



High-level architecture: Elektrobit. The ASIL B / SIL 2 statement is based on a positive technical assessment feasibility report from TÜV Nord.

© Elektrobit 2026

Motivation

- Get to know the community & get the community to know us 🙂
- Exchange feedback and have discussions
- Share what we do and check if something could be worth upstreaming



Photo: Alena Darmel / Pexels

© Elektrobit 2026

ISAR in the EB corbos Toolkit

- **EB corbos Toolkit**: tooling around Debian-based embedded Linux products
  - develop, build, run multi-VM system images (incl. hypervisor), export SDKs, and applications using them
- Supporting Linux integration for
  - embedded systems,
  - focus on automotive,
  - especially **Linux for Safety Applications**, but also "regular" Linux
- ISAR: build engine we use and extend
- EB layers describe product packages, images, and machine integration.
- `kas` pins the ISAR revision (currently: v1.1), applies a downstream patch series, generally orchestrates tasks

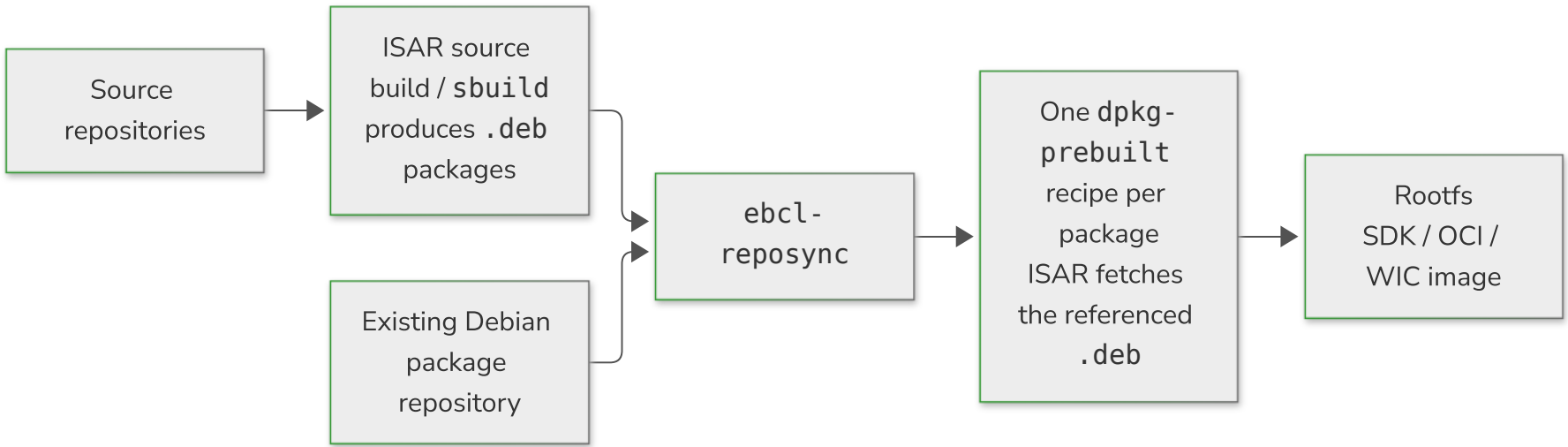


Photo: ThisIsEngineering / Pexels

© Elektrobit 2026

Toolkit Specialty: Per-Package Download Recipes

> EB corbos Toolkit heavily relies on generated `dpkg-prebuilt` recipes.



Version discipline
Strict package-version selection and download correctness.

Separate tooling
`ebcl-reposync` generates the recipes, simplifies usage.

Usefulness upstream? (own assessment)
Likely too specific to upstream as-is.

© Elektrobit 2026

What we use from ISAR

- Toolkit leans on existing ISAR abstractions: recipe/task model, fetchers, dependencies, and `sstate`.
- Debian package build and deployment use `dpkg` and `sbuild`, with `mmdebstrap` for rootfs bootstrap.
- ISAR handles package installation into rootfs, SDK generation and relocation, OCI container creation, and WIC image assembly.
- EB adds mostly layers, configuration, custom classes: recipes, generated package metadata, distro/image policy, and machine support.
- **We patch ISAR and BitBake** where required behavior is not currently provided; the next slides sketch those deltas.
  - Interested if others have related patches or not!



Photo: ThisIsEngineering / Pexels

© Elektrobit 2026

Patches: Correctness and Concurrency

Summary	Details	Usefulness upstream? (own assessment)
Multiarch fixes	Reject `HOST_ARCH`-incompatible native providers; higher `PV` can otherwise override compatible siblings.	High
Buildstats	Handle zero-duration tasks without division by zero.	High
`isar-apt` locks	Keep repository state stable through rootfs, imager, and bootstrap installs; fix race conditions.	High
`PREMIRRORS`	Use the original filename when mapping mirror URLs. Otherwise, using premirrors seems to not work correctly - at least for us.	High
Download cache	Allow copying instead of hardlinking on overlay filesystems; circumvents another race condition.	Medium to high
Repository hygiene	Fix a `.gitignore` pattern that can exclude valid `build*` files. Causes issues only on vendoring.	Medium

© Elektrobit 2026

Patches: Reproducibility and SDKs

Summary	Details	Usefulness upstream? (own assessment)
Reproducible package outputs	Use a fixed `sbuild` path and clamp source `mtimes` to `SOURCE_DATE_EPOCH`.	High
Reproducible OCI images	Normalize rootfs and image/history timestamps.	High
Relocatable SDKs	Relative linker paths; GCC/Clang relocation; respect wrapper settings; use the SDK's own `patchelf`. We ran into "interesting" build issues when using exported SDKs in Bazel.	Medium / unclear
Buildstats	Handle zero-duration tasks without division by zero.	High

© Elektrobit 2026

Features

Summary	Details	Usefulness upstream? (own assessment)
Bootstrap from local `isar-apt`	Let `mmdebstrap` use packages from ISAR's own repository. Could help other package-only builds. Maybe already submitted?	Unclear
"Enterprise buildability"	E.g. change source URLs of `debsbom` so they work with `PREMIRRORS` (required for "no direct access to internet in CI").	Unclear

© Elektrobit 2026

Summary

- We use ISAR and like it!
- We have some possible things to upstream - maybe?
- **Any feedback?** Anyone else interested in "build reproducibility aspects" in general?



Photo: Domingos Henriques / Pexels

© Elektrobit 2026

Contact us

Dr. Oliver Pajonk

Principal Engineer, HPC OS and Middleware
Elektrobit – Driving the future of software

Phone +49 9131 7701-0

Email oliver.pajonk@elektrobit.com

Web elektrobit.com



© Elektrobit 2026