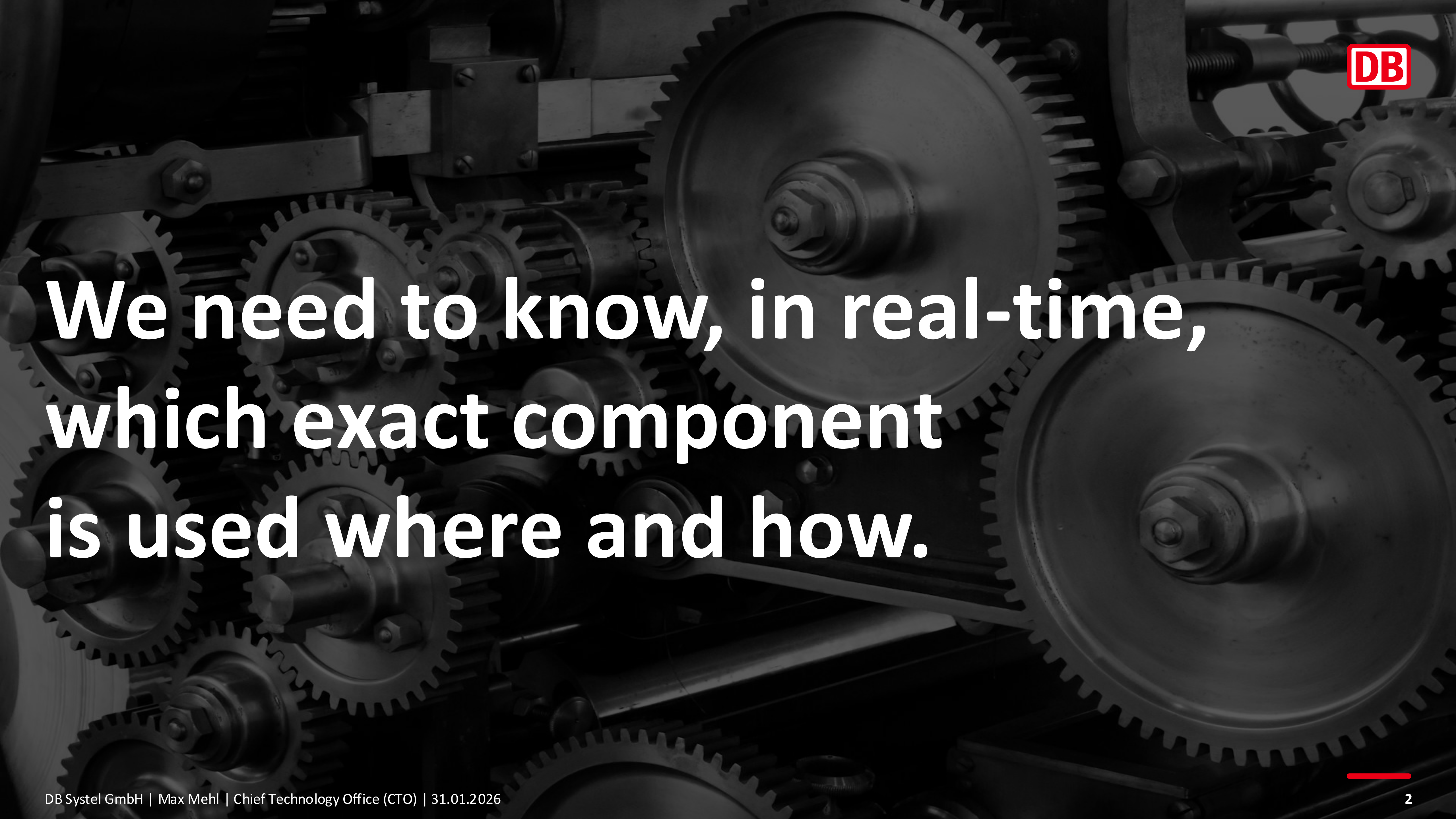
Software Supply Chain Strategy at Deutsche Bahn

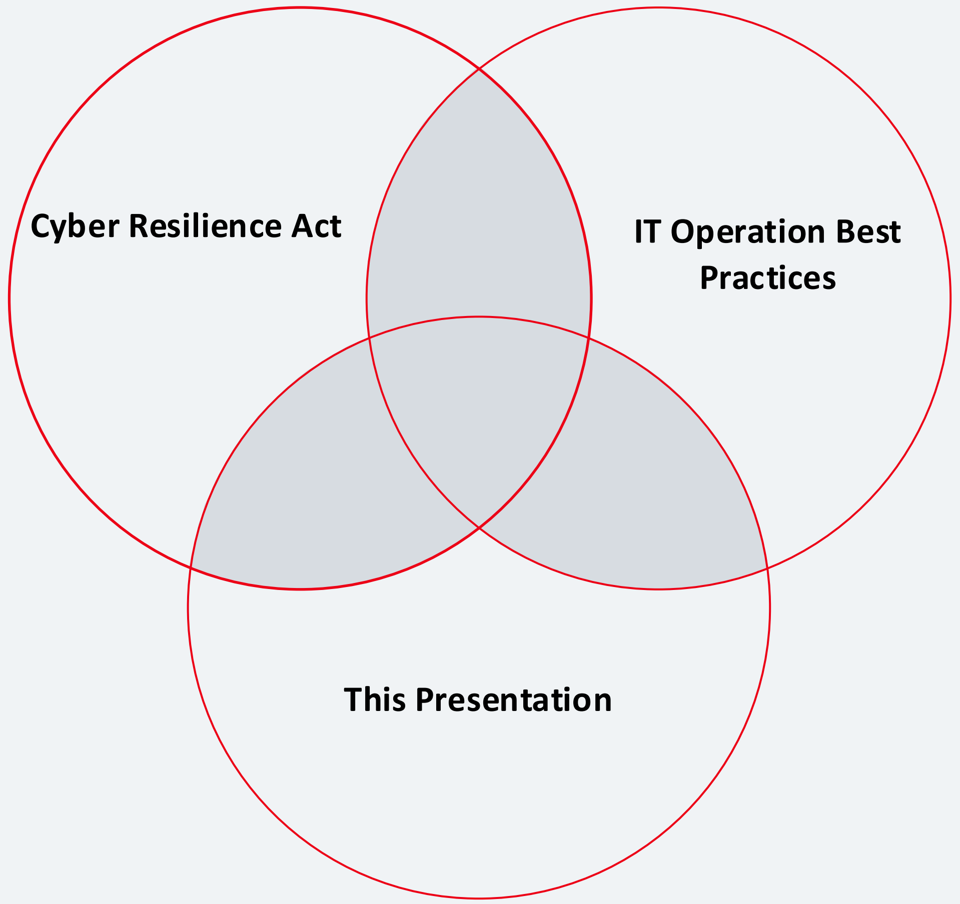
From Operational Need to Concrete Implementation

DB Systel GmbH | CTO Team | Max Mehl | FOSDEM 2026 | 31.01.2026



We need to know, in real-time, which exact component is used where and how.

Intersection of CRA, Our Daily IT Operations, and This Session



Deutsche Bahn's Business is Trains, not Software

But its IT is equally large

Our Core Business

Transporting people and goods.

- 5,1 million train travelers / day
- 60,970 km of tracks
- 5,700 train stations
- 22,500 trains / day
- 180 million tons of freight / year

Complex Organization

A large and diverse organization keeps our core business running every day.

- 220,000+ employees
- 500+ professions
- Hundreds of subsidiaries

Digitalization is Essential to Scale in the Future

Without IT – and Open Source – no train would be able to run.

- 7,000+ IT applications/services
- 10,000+ IT professionals
- 20,000+ virtual machines
- 40,000+ containers
- 60,000+ repositories
- 100,000+ OSS components

Example: DB Navigator for information and ticketing

The essential entry point for most travelers.

- 23 million users per month
- 170 million travel information requests per month

Supply Chain Management in Context of CRA

CRA is the context for what I'll be talking about today, not the trigger

In our understanding, CRA consists of 4 activity areas

- General principles of secure software and products ← we do that
- Professional handling of software vulnerabilities ← we also do that
- **Transparency of software supply chains, SBOMs** ← that's a new challenge, focus of today
- Information to users, conformity assessments ← out of scope for today, but interesting

SBOMs weren't new to us

- Originated from Open Source license compliance
- Previously missing: alignment of different governance systems and regulation

We didn't adopt SBOMs *because* of regulation — regulation validated the direction.

Transparent Supply Chains: Easier Said than Done

At DB, we have the most diverse sourcing streams for IT

Build software

- For ourselves (services, internal)
- For external customers (you)
- Ranging from operating systems for displays in trains, to services, to apps on your phones

Buy software

- Local
- On-premise
- SaaS
- Bundled in hardware (like trains)

Operate software

- On-premise
- Cloud (VM and containers)
- Edge (embedded)

Which software components are where? And in which state and context?

SBOMs As a Common Methodology to Tackle Challenges

SBOMs are not a means by itself, but a standardised method to support several needs

- Establishing Open Source license compliance
- Checking for known security vulnerabilities
- Assessing quality of used components
- Supporting strategic decisions on ecosystem engagement and investments
- Understanding the (distribution of) use of components, frame-works, and ecosystems
- Satisfying regulatory and internal governance needs
- Lowering the barrier for integration and processing with other services and tools

SBOMs must become shared infrastructure.

VEX is a Perfect Match for SBOMs

VEX as a perfect match

- Standardized way to make a statement on the status of a known vulnerability detected in one's supply chain
- Match CVE to component found in an SBOM
- Track status information throughout involved processes and tools, avoid duplicated work for teams
- Allow manufacturers to communicate their interpretation of affection status to us

Reality: integrate a new underlying standard beneath existing processes and tools → challenging in large organizations

To be effective, VEX and SBOMs must be thought together.

Creating an SBOM Strategy and Architecture from Scratch

Challenges

- Size and diversity of the organization
- Various software sourcing models
- DB's different roles and requirements
- Many stakeholders and user groups
- Preset tools and processes
- Limited resources of teams
- Pressure of time, e.g. by the CRA

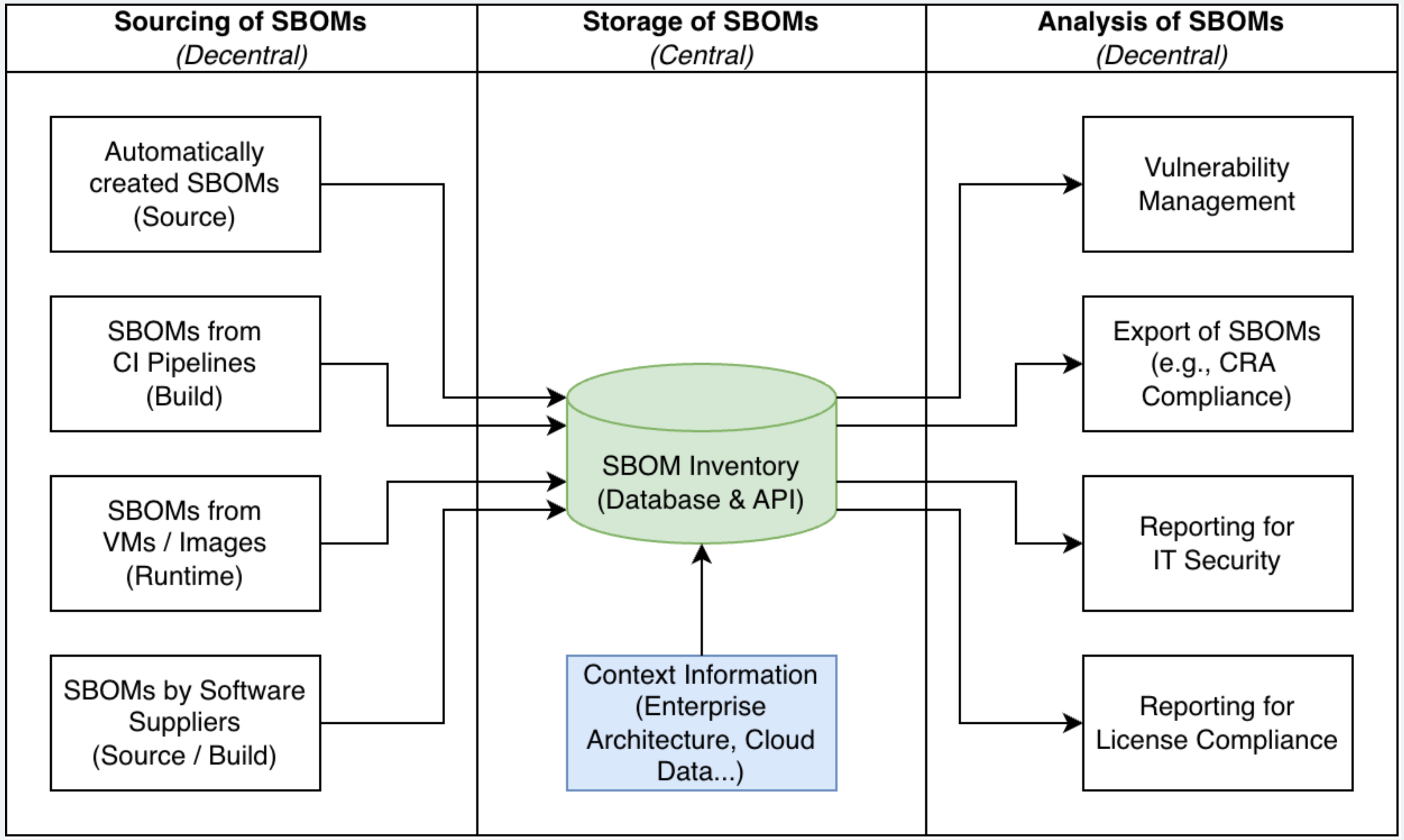
Procedural principles

- Small, interdisciplinary group, consisting of volunteers
- Iterate quickly, gather feedback continuously
- Do not talk in tools, but capabilities
- Focus on existing needs of the organization, not abstract recommendations with all the bells and whistles
- Think big, expect incremental realization
- Document progress and material organization-public

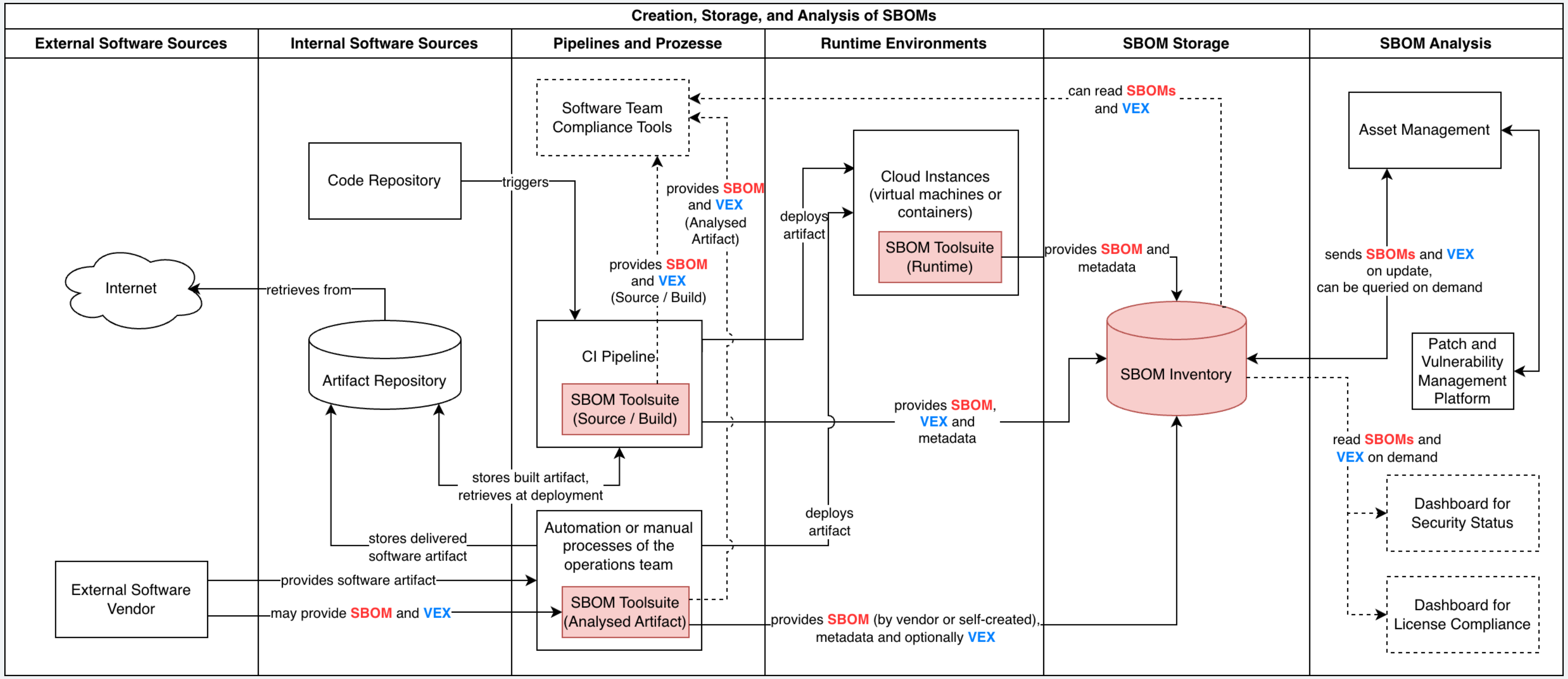
Technical and architectural principles

- Consider all sourcing and SBOM types incl. VEX
- Modularity
- Open standards and interfaces
- Central storage of SBOMs
- Decentral sourcing and analysis of SBOMs

Our Mental Model of SBOM Lifecycle Consists of Three Phases


Sourcing of SBOMs
(Decentral)
Storage of SBOMs
(Central)
Analysis of SBOMs
(Decentral)
Automatically created SBOMs (Source)
SBOMs from CI Pipelines (Build)
SBOMs from VMs / Images (Runtime)
SBOMs by Software Suppliers (Source / Build)
SBOM Inventory (Database & API)
Context Information (Enterprise Architecture, Cloud Data...)
Vulnerability Management
Export of SBOMs (e.g., CRA Compliance)
Reporting for IT Security
Reporting for License Compliance

The SBOM Blueprint is Our Guiding Star


Creation, Storage, and Analysis of SBOMs
External Software Sources
Internal Software Sources
Pipelines and Prozesse
Runtime Environments
SBOM Storage
SBOM Analysis
Internet
External Software Vendor
Code Repository
Artifact Repository
retrieves from
triggers
stores built artifact, retrieves at deployment
stores delivered software artifact
provides software artifact
may provide SBOM and VEX
Software Team Compliance Tools
provides SBOM and VEX (Analysed Artifact)
provides SBOM and VEX (Source / Build)
CI Pipeline
SBOM Toolsuite (Source / Build)
Automation or manual processes of the operations team
SBOM Toolsuite (Analysed Artifact)
deploys artifact
Cloud Instances (virtual machines or containers)
SBOM Toolsuite (Runtime)
deploys artifact
provides SBOM, VEX and metadata
provides SBOM (by vendor or self-created), metadata and optionally VEX
can read SBOMs and VEX
provides SBOM and metadata
SBOM Inventory
Asset Management
sends SBOMs and VEX on update, can be queried on demand
Patch and Vulnerability Management Platform
read SBOMs and VEX on demand
Dashboard for Security Status
Dashboard for License Compliance

Last updated: March 2025

Implementation of Architectural Blueprint by Prioritized Increments

- Given the preconditions, implementation cannot happen overnight
- Prioritization based on identified risks, external requirements, and pragmatism

Results

- Focus on Source/Build SBOMs for software developed in-house
- Onboard as many teams as possible
- Low-threshold drop-in solutions for CI pipelines and their templates
- Increase SBOM Quality, especially licenses and metadata → but balance quality vs quantity
- Teams: Integration into compliance portal
- Governance: Enable basic central insights, no shiny dashboards
- Focus on Happy Paths, do not consider all edge cases from the start

Future Steps and Improvements

- Runtime SBOMs from VMs and containers
- Easier ingestion of SBOMs delivered by vendors
- Support of OT and low-level IT close to hardware

Central Oversight Makes Supply Chain Dimensions Transparent

79,943 SBOMs analyzed from Source and Build stages

1,855 enterprise applications covered by the analyzed SBOMs

104,904 packages in use, most of them Open Source

52,115 internal repositories covered

244 dependencies on average per code project

7.7% of our code projects contain the most-used dependency

Challenge: turn data into actionable items.

Last updated: January 2026

Tools Don’t Integrate Themselves – It’s People

To establish SBOMs and related tools/processes as a core methodology, we need to take all users with us:

- High adoption > perfection
- Pipelines and tools > dashboards
- Automation > manual processes
- Incremental improvements > Big bang release
- User feedback > top-down governance
- Open Source > Inner Source > Blackboxes

Concrete actions

- Heavy use of open source tools to which we contribute upstream
- All development, issue tracking and planning Inner Source, prospectively partly Open Source
- API and automation by default
- Regular open office hours for all users of the related tools and services: see new features, answer questions, provide direct feedback to developers
- Resulting findings are risk-based to not overload teams and help them prioritize

Governance Owners Can Tip the Scales

- Work together and align. Do not think in silos (e.g. Open Source compliance vs IT security)
- Take load off operative teams: automation, compliance by default, risk-based approaches
- Clear expectations to implementation, support agile development, consider 80% solutions

Challenges:

- Keep oversight of different working streams, stakeholders (internal and external)
- Identify mismatches, misunderstandings, and blockers for progress

Take-aways and Call to Action

Main take-aways

1. SBOMs are a common methodology, beyond individual needs
2. Think big, implement incrementally
3. Modularity > monoliths
4. Delight your users

Call to Action

1. Internalize knowledge and skill about such core technologies
2. But collaborate and share in the open
3. Do not reinvent the wheel

Thank you!

Max Mehl
Open Source Governance & Strategy

max.mehl@deutschebahn.com
@mxmehl@mastodon.social
@mxmehl

Join the Follow-up Session!

Sun, 12:00 @ SBOM devroom (UD2.208)

“Deutsche Bahn's Approach to Large-Scale SBOM Collection and Use”