

# Cross Domain Configuration Management - Die Grundlage für Informationsmanagement im Engineering

# Die Komplexität in der Entwicklung von Produkten steigt rasant

## **Produkte: Mehr Komplexität**

- Produkte übernehmen immer anspruchsvollere Aufgaben
- Der Grad der Vernetzung von Informationssystemen steigt (System of Systems)

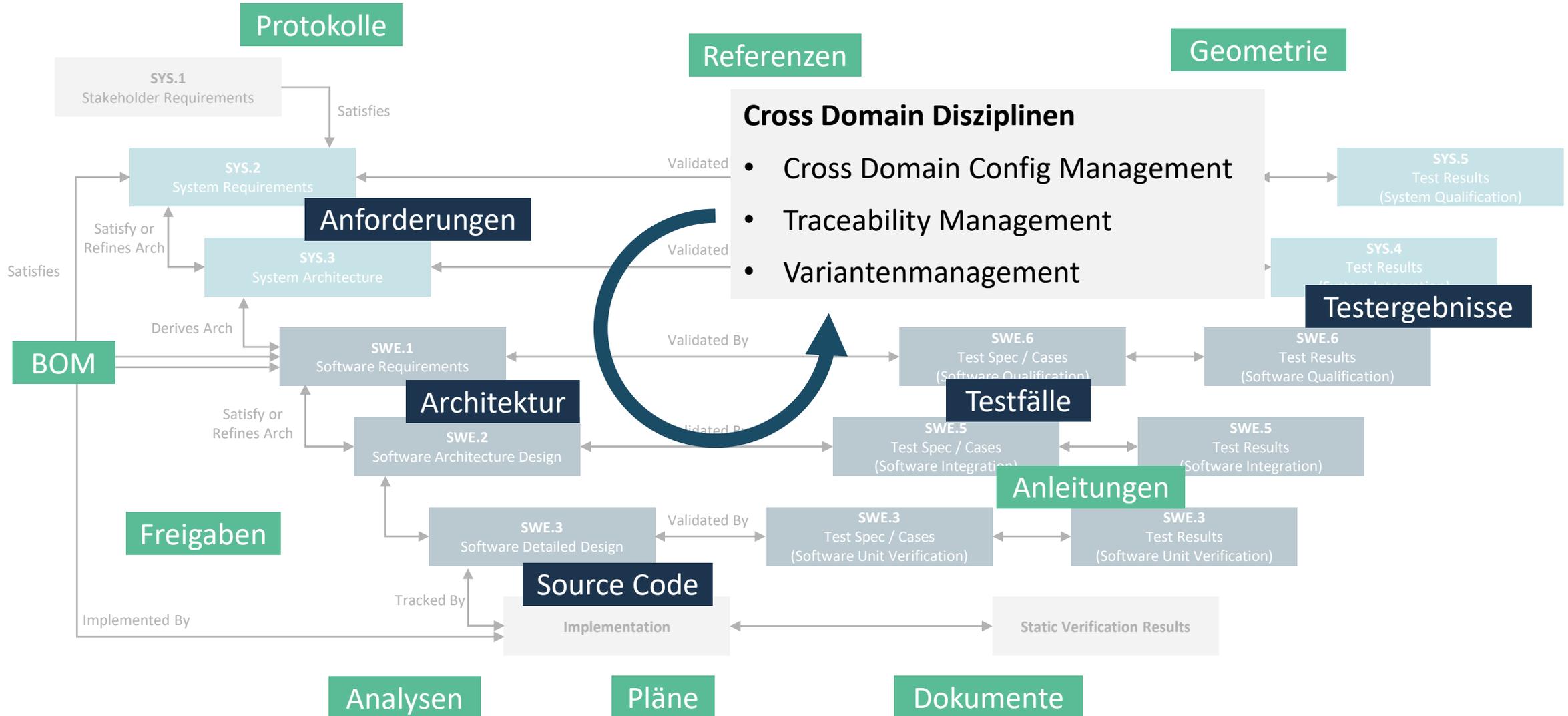
## **Werkzeuge: Mehr Vielfalt**

- MBSE verteilt Informationen auf mehr Werkzeuge
- Immer spezialisierte teils proprietäre Werkzeuge
- Regulierung fordert Traceability von der Anforderung bis in den Code

## **Prozesse: Mehr Tempo**

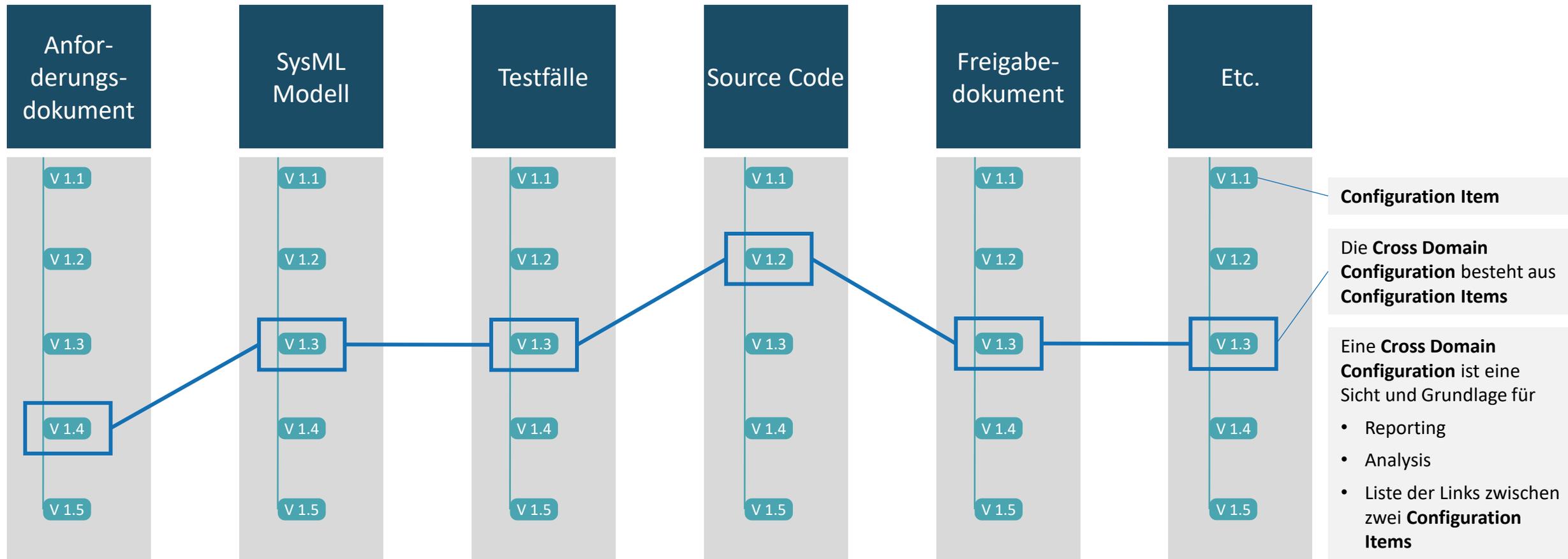
- Modernes Product-Line Engineering
- Parallelentwicklung von Varianten und Versionen
- Maximierung der Wiederverwendung von Inhalten

# Die Kontrolle des "V" erfordert deshalb holistische Ansätze in mehreren Disziplinen



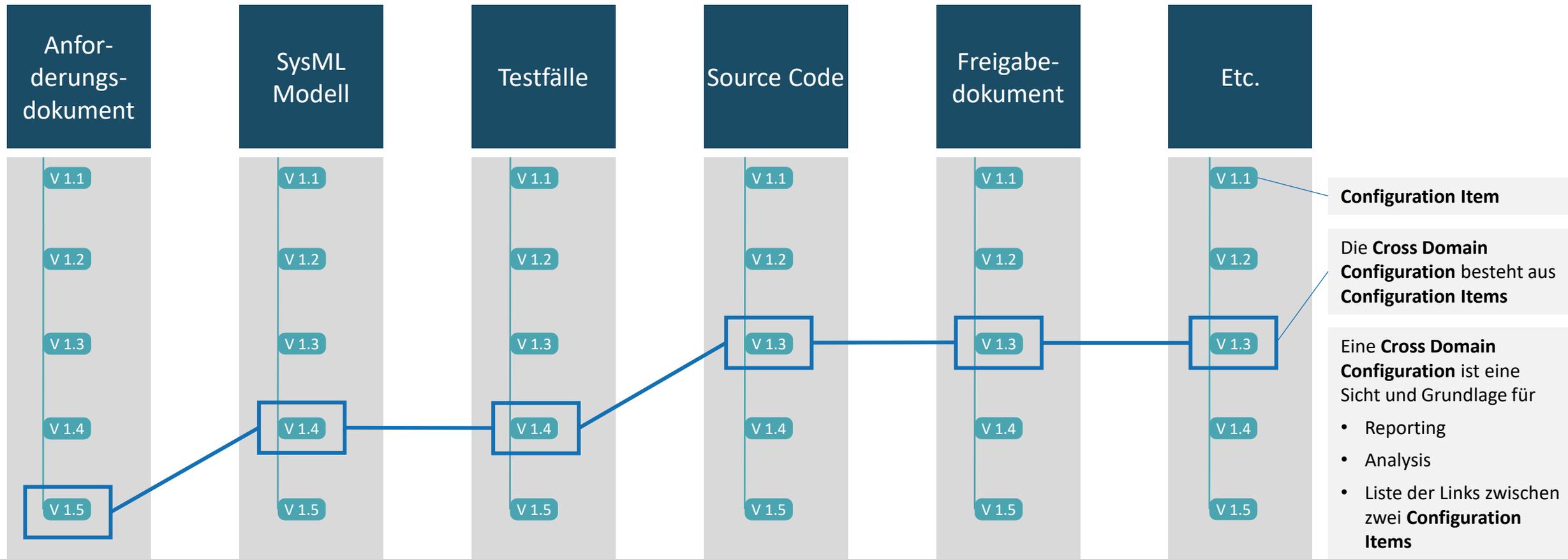
# Versionierte Engineering Informationen sind über viele Werkzeuge und Domänen verteilt

## Alle Autorenwerkzeuge erzeugen versionierte Inhalte

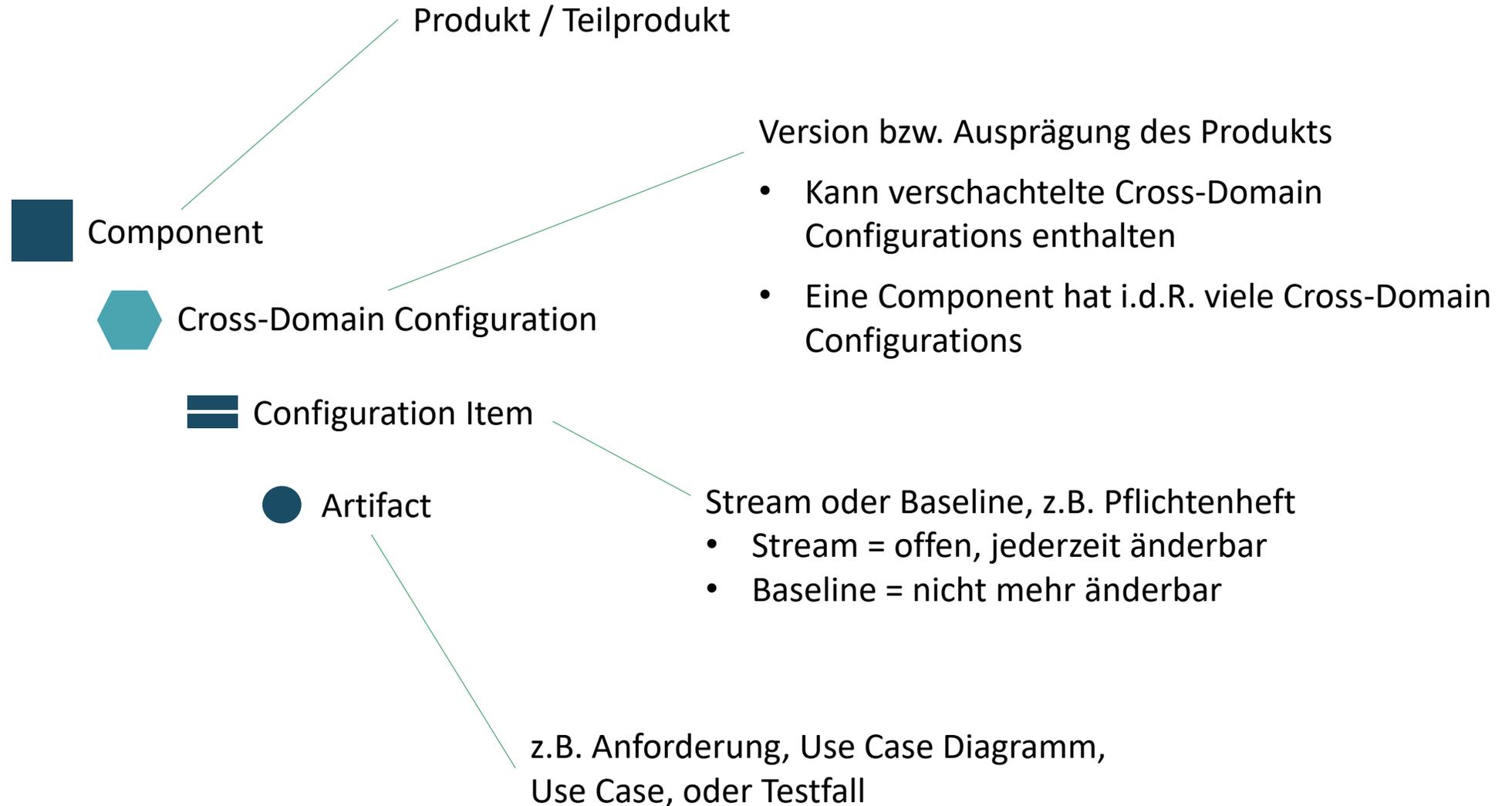


# Versionierte Engineering Informationen sind über viele Werkzeuge und Domänen verteilt

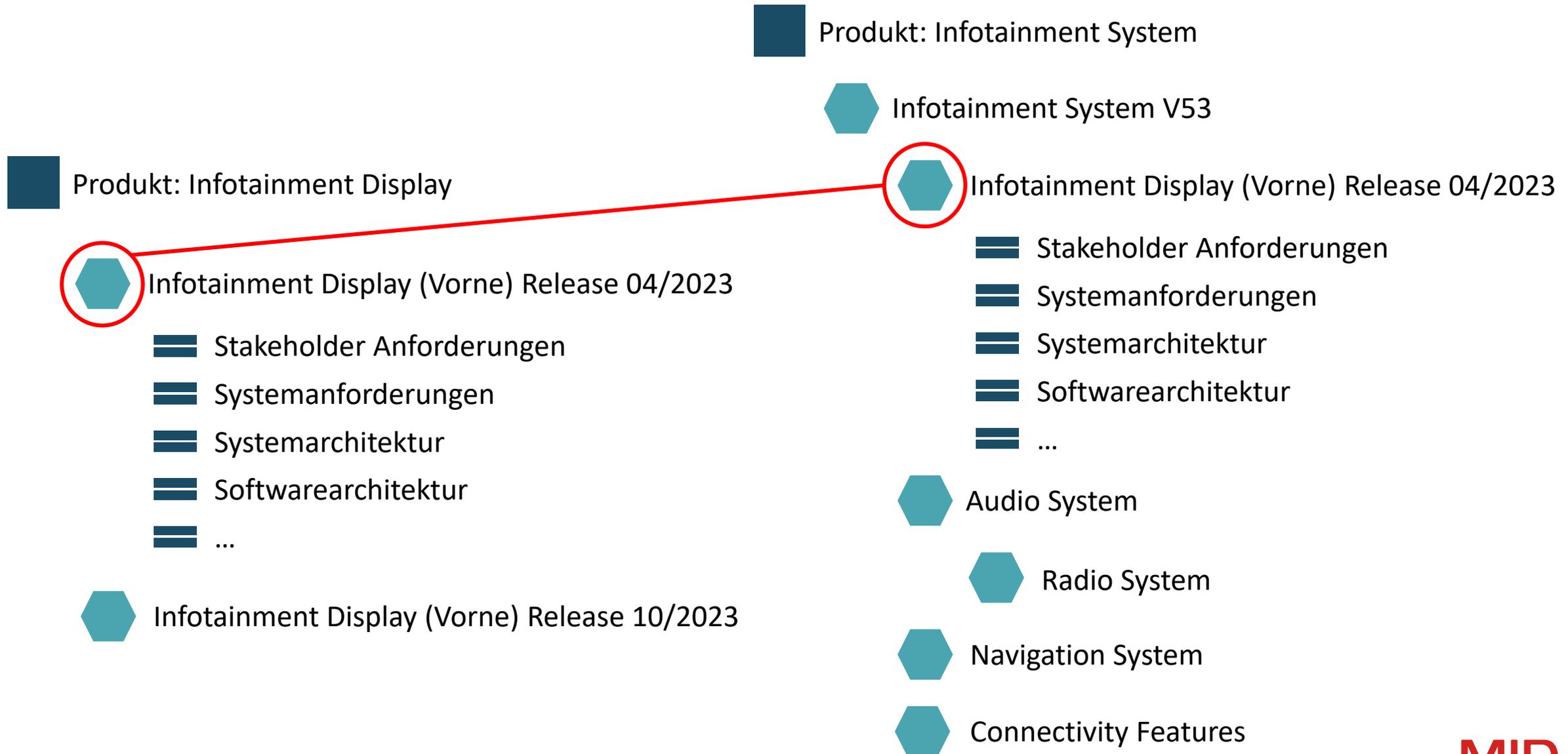
## Alle Autorenwerkzeuge erzeugen versionierte Inhalte



# Beispiel für eine Cross Domain Configuration



# Beispiel für eine Cross Domain Configuration





Tools

Aktuell nimmt kein Vendor die Rolle des Integrators ernsthaft wahr

## ★ Lone Stars

### Jama Software

- Jama Connect

### Contact Software

- CIM Database

### Open Source

- GIT
- Eclipse

### Zuken

- Genesis

### Polarion Inc.

- Polarion

### NoMagic

- Cameo Systems Modeler

### Sparx

- Enterprise Architect

### The Mathworks

- Matlab Simulink

### Vector

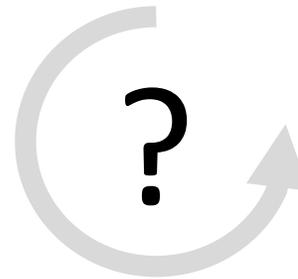
- PREEvision

### IBM

- DOORS Classic

### Intland

- Codebeamer



## Cross Tool Config Management Cross Tool Traceability Management

## ★ Proprietäre Tools

- Autorenwerkzeuge
- Datenquellen

## ★ Plattformen

### IBM ELM

- DOORS Next
- Engineering Test Mgmt
- Engineering Workflow Mgmt
- Rhapsody
- Global Config Management

### Dassault 3DEXperience

- Cameo Systems Modeler
- Enovia
- CATIA
- Etc.

### Siemens

- TeamCenter
- Polarion
- NX
- Modeling Solution

### PTC

- RV&S
- Codebeamer
- Modeler
- Creo
- Onshape
- Windchill

### Atlassian

- Confluence
- Jira
- Bitbucket

### Ansys

- Etc.

# Wie wird Cross Domain Configuration Management heute gemacht?

## MS Excel

- Manuell
- Keine Hierarchische Abbildung von CDCs
- Keine Wiederverwendung
- Nicht Tool-“connected”:  
Referenzierung auf Configuration Items
- Produktübergreifende Baselines müssen manuell erstellt werden
- Schwächen hinsichtlich Compliance

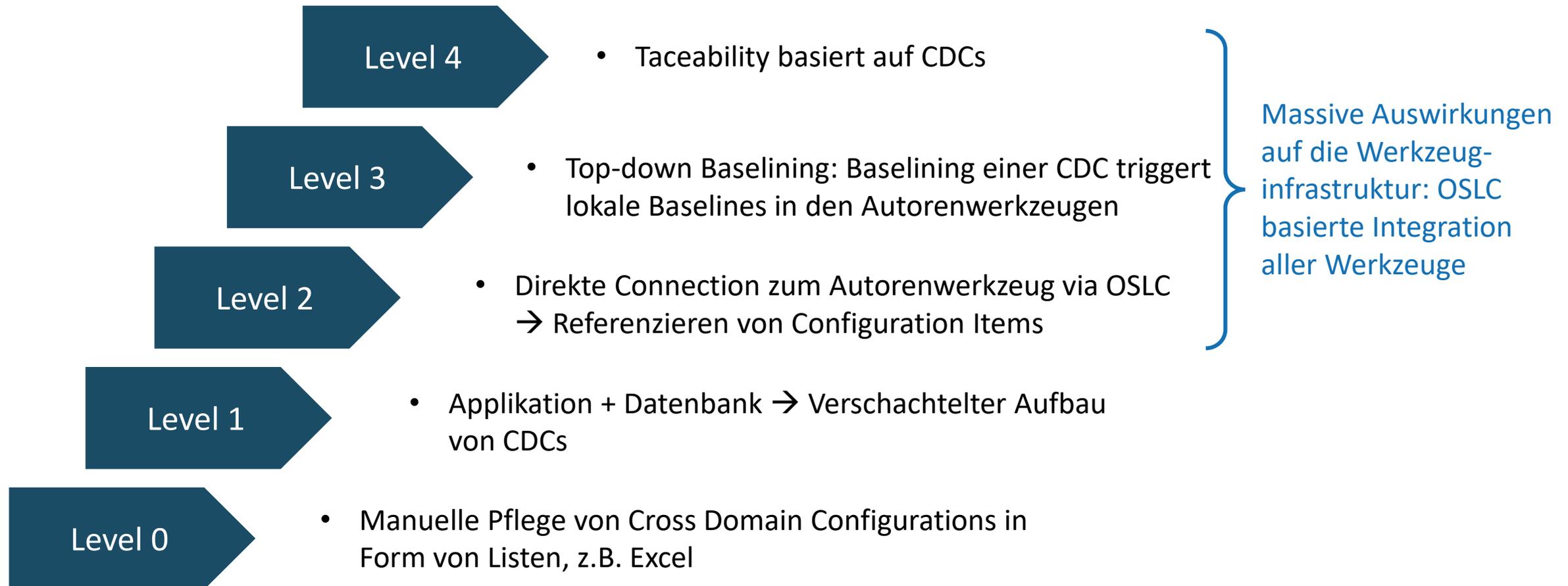
## IBM ELM (“Jazz”)

- “Global Config Management” ist integriert mit den ELM Autorenwerkzeugen
- GCs können hierarchisch aufgebaut und verschachtelt sein
- Traceability und Reporting funktionieren auf Basis von Global Configurations
- Erweiterbarkeit durch OSLC gegeben, aber selten genutzt

## Gesucht: Ein Autorenwerkzeug für Cross Domain Configurations

- Tool-/Plattform- / Vendor-neutral
- Standard-basiert: OSLC Standard
- Wiederverwendbarkeit von CDCs
- Verschachtelung von CDCs
- Integration mit allen wichtigen Autorenwerkzeugen
  - Auch non-engineering Tools wie Sharepoint, DMS, etc.
- Konfigurierbares System von Business Objects
- Integrierbar mit Workflow Management
- Integrierbar mit Umsystemen (SAP, etc.)

# Reifegrade für Cross-Domain Configuration Management



# Was ist eigentlich der OSLC Standard?

- OSLC = Open Services for Lifecycle Collaboration
- Standard zur Verbesserung der Tool Interoperabilität
- Seit 2013 Mitglied in der Open Standard Organization OASIS
- Wichtige Bestandteile:
  - OSLC Core
  - OSLC Configuration Management
  - OSLC TRS (Tracked Resource Set)
- Primäre Nutzung: Sicherstellung der Traceability zwischen Artefakten unterschiedlicher Werkzeuge

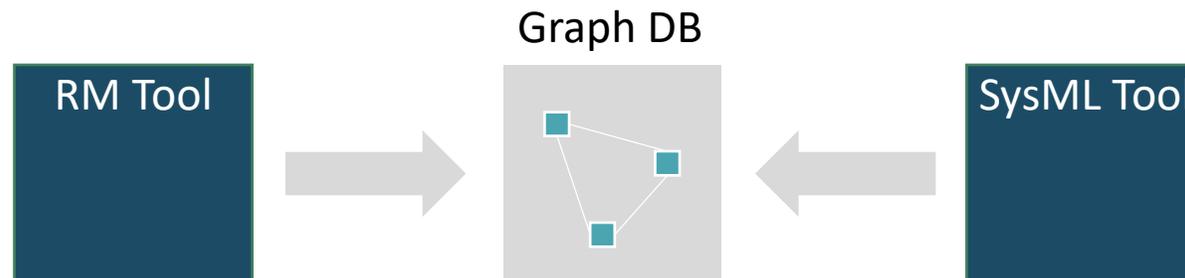
# OSLC: Referencing Content - Instead of Synchronizing it

**Synchronize** one silo into the other and leverage tool internal link mechanisms



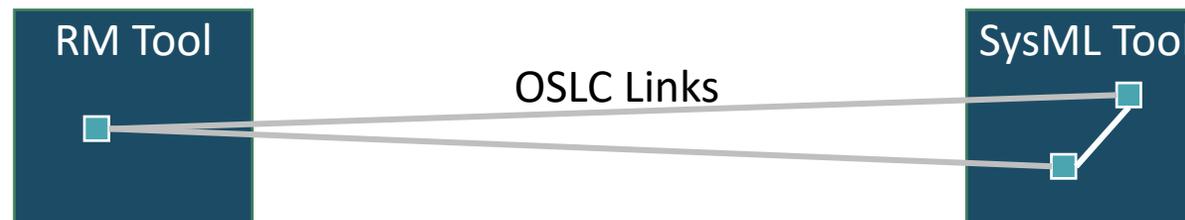
- Too little flexibility e.g. for variant management
- Security Issues
- Works only for two models

**Synchronize** models and links into one data source and manage them there



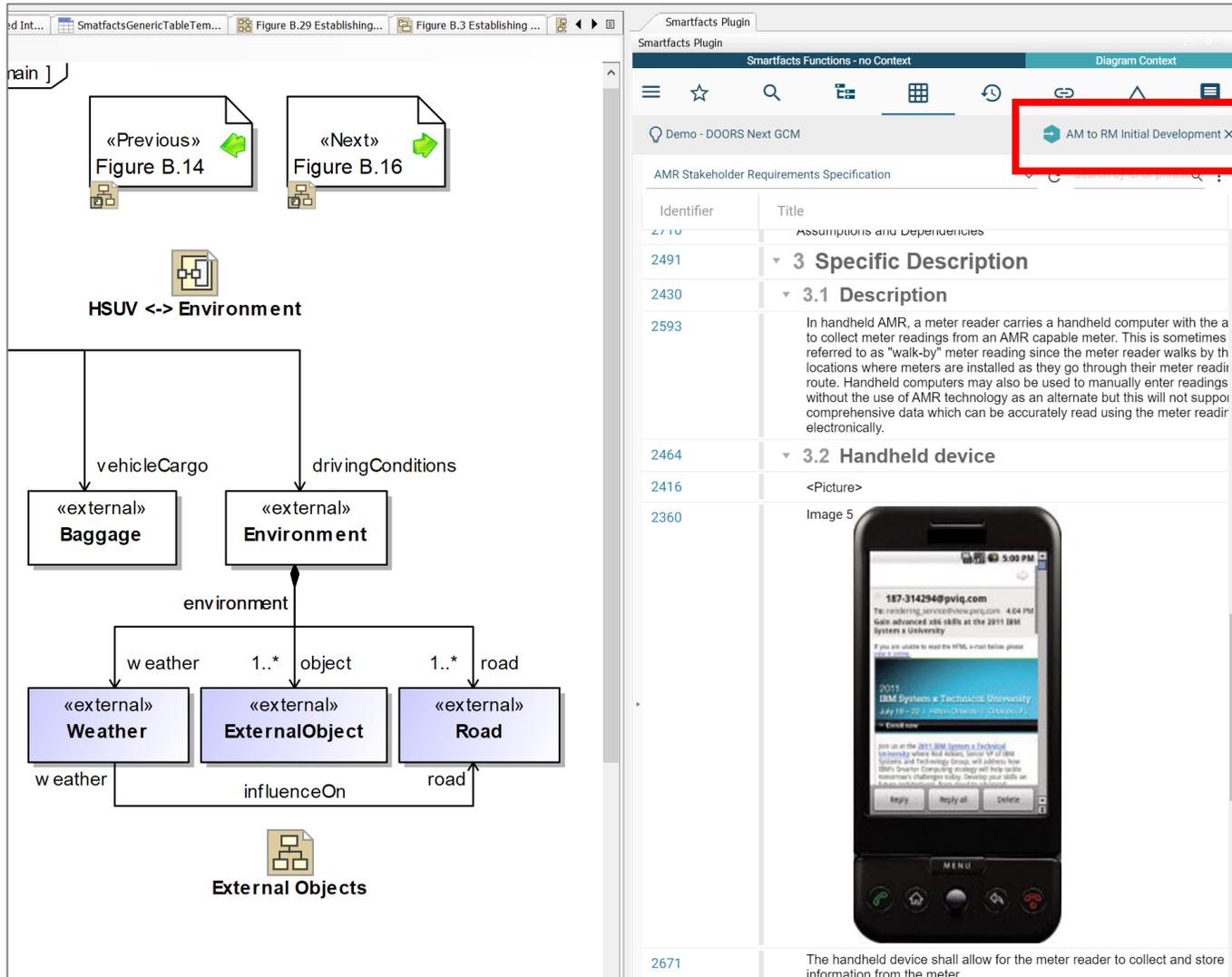
- Bad behavior in model branch and merge

Store the links in the models and **reference** to the other side

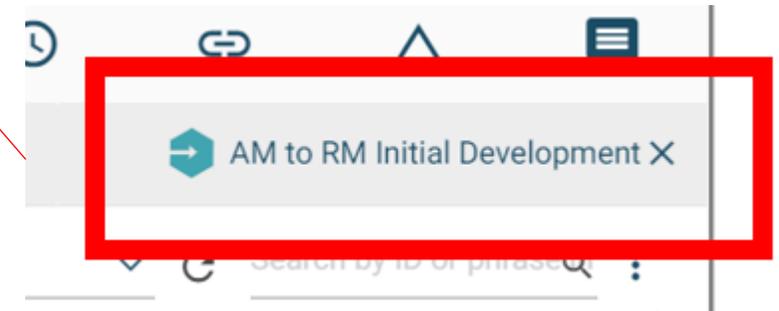


- Flexible
- Scalable
- Secure

# Beispiel: Integration mit Cameo



**Tool-übergreifender Kontext** definiert, in welcher Cross-Domain Configuration gearbeitet wird



# Beispiel Integration Anforderungen und mehr mit Source Code (GIT)

The image shows a development environment with two main windows:

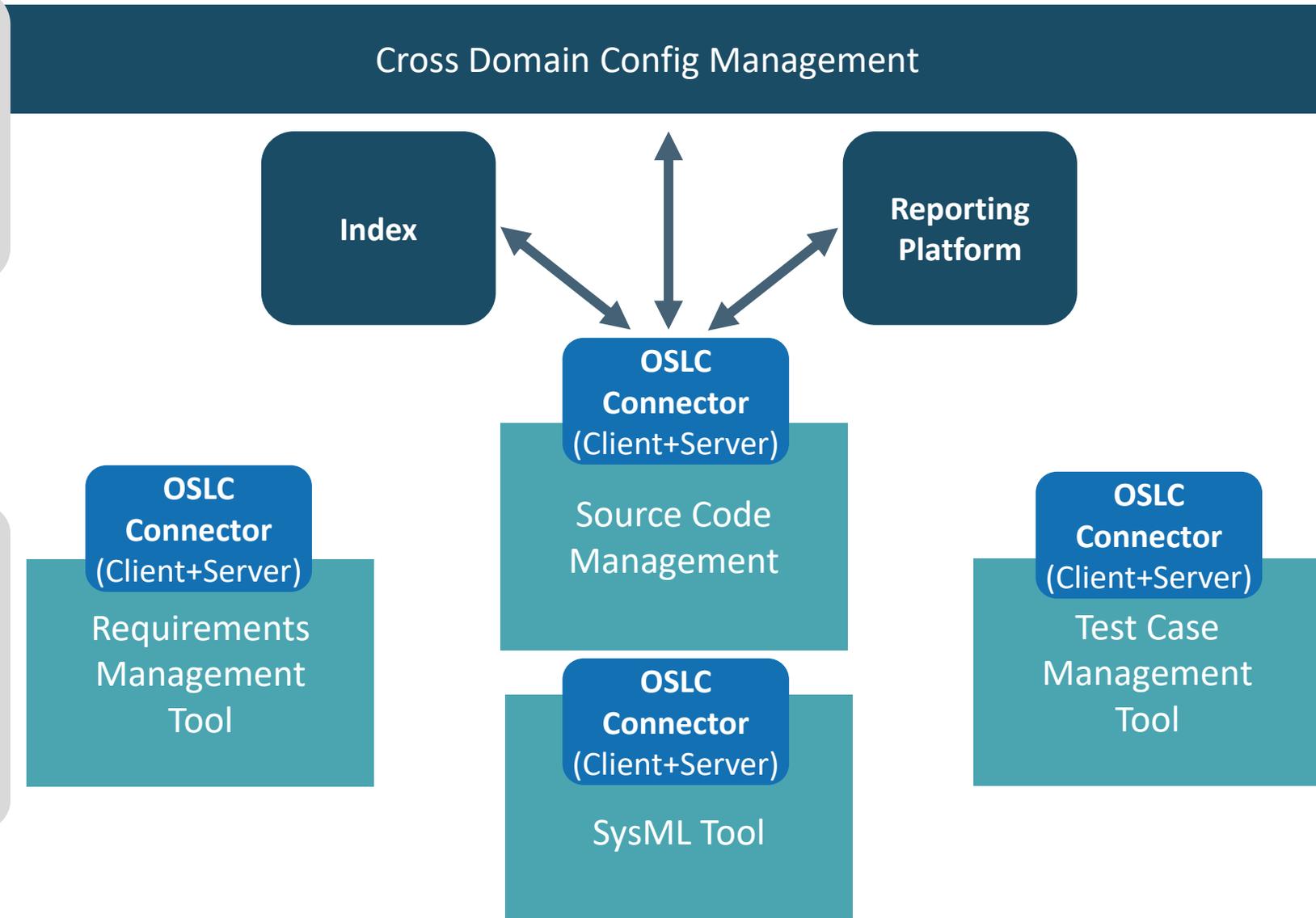
- Source Code Editor:** Displays C++ code for a template class `SwTemp1`. The code includes a `GetMax` function and a constructor that takes two integer values. It uses `@ImplementsRequirement` annotations to link code elements to specific requirements in the Smartfacts tool.
- Smartfacts Modeling Tool:** Shows a project named "AMR Basis Initial Developm...". The tool's interface includes a toolbar with icons for grid, refresh, and link. A list of requirements is displayed, such as "Leashed Pets", "Stray Animals", and "Temperature Operational Limit of the device". A "Global Configuration Definition" section is also visible, containing a "Source Code Integration - V2" entry with a GitHub link and "AMR 2 Requirements Initial Stream".

Red boxes highlight the project name "AMR Basis Initial Developm..." in the Smartfacts toolbar and the "Global Configuration Definition" section.

# Architektur einer Entwicklungsumgebung mit Cross-Domain Configuration Management

- Cross-Domain Configurations erstellen und verwalten
- Lokale Konfigurationen referenzieren
- Analysen und Reports erstellen
- Workflow Management

- CDC muss als globaler Kontext gesetzt werden
- Verifikation ob das aktuell geladene Modell der CDC entspricht
- Berechnung der Liste der Links auf Basis des globalen Kontext



# MID entwickelt Komponenten, um Cross-Domain Config Management Werkzeug- und Vendor-neutral umzusetzen

genOSLC

- OSLC Kompatibilität schaffen für beliebige – auch proprietäre Autorenwerkzeuge mit minimalem Aufwand

Cross-Domain  
Configuration  
Management

- Standardkonforme, konfigurierbare Lösung zum Management von Cross-Domain Configurations inkl. Integration in die Autorenwerkzeuge

Smartfacts  
Link Index

- Link Index basierend auf Cross-Domain Configurations, inkl. Link Validity

Reporting Plattform

- Big Data Plattform zur Erzeugung von Domänen- und werkzeugübergreifenden Analysen, Auswertungen, Reports

Smartfacts  
Speziallösungen

- Speziallösungen z.B. für visuelle Modellierungswerkzeuge oder Traceability in den Source Code

Heute verfügbar

In Entwicklung

# Zusammenfassung

- Engineering Artefakte sind über immer mehr Werkzeuge verteilt. Werkzeugübergreifende Aspekte (Traceability, Varianten, etc.) werden immer wichtiger.
- Das Management von Cross-Domain Configurations ist wichtige Grundlage für alle werkzeugübergreifenden Sichten
- OSLC ist der zentrale Standard, um Cross Domain Config Management zu implementieren
- Die Umsetzung hat massive Auswirkungen auf die Gestaltung von Schnittstellen der Autorenwerkzeuge
- Viele Vendors ziehen inzwischen am gleichen Strang (IBM, PTC, Vector, Siemens) – der Weg ist aber noch weit

# Mehr gerne am Stand von MID

Kontakt:

Christoph Bergner, [c.bergner@mid.de](mailto:c.bergner@mid.de)

Ed Gentry, [e.gentry@mid.de](mailto:e.gentry@mid.de)