

Safety Case as Data:

Towards a machine readable safety SBOM for Linux



Linux Plumbers 2026

05.10.2026

Nicole Pappler, AlektoMetis

Kate Stewart, Linux Foundation



Agend

a

SPDX 3.1 Orientation

Work Product Models for the Safety Case

Feedback and Suggestions



Whoami – Nicole Pappler

Safety Consultant at AlektoMetis



Professional History:

Been working in production maintenance, automotive, ECU software development

All my projects had some safety criticality

Started to focus on Functional Safety about 15 years ago at TÜV SÜD

Currently:

Tech consulting as part of AlektoMetis

Supporting my customers regarding Functional Safety, Security & compliant use of open source

Involved in some open source projects:

- Zephyr (Functional Safety Manager)

- ELISA (Medical & Systems Group)

- FuSa for SPDX Profile Group

- OpenChain (3rd party certification with TÜV SÜD)

What else?

Contact handle at GitHub, Discord, etc: @nicpappler

System Package Data eXchange

SPDX Overview

- Open standard, providing a format to describe system elements in both machine and human readable way and enable exchange of metadata information between organizations
- SPDX Project consists of the
 - SPDX Specification,
 - SPDX License List, and
 - SPDX tools and libraries
- SPDX 2.3 -> ISO/IEC 5962:2021
- SPDX 3.0 ISO/IEC ballot passed, will be formalized by ISO this year incorporated AI & Data related information

Timeline of SPDX Evolution - Use Case by Use Case

I AM THE
Cavalry
Legislation



NTIA:
Software
Transparency
begins



3T-SBOM:
CONOPs for
Tool-to-Tool
SBOM



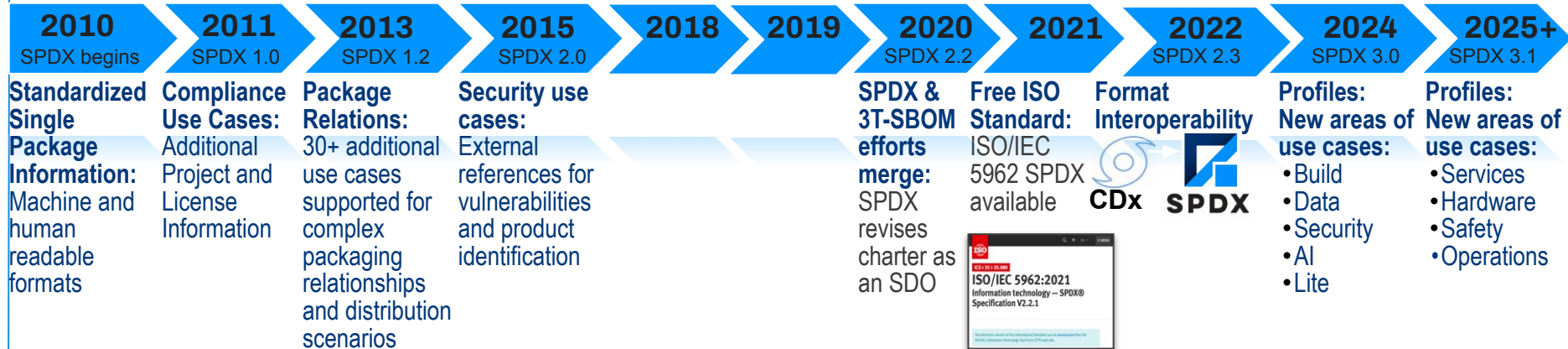
**Transition of
SBOM work
to DHS**



**Executive
Order 14028**

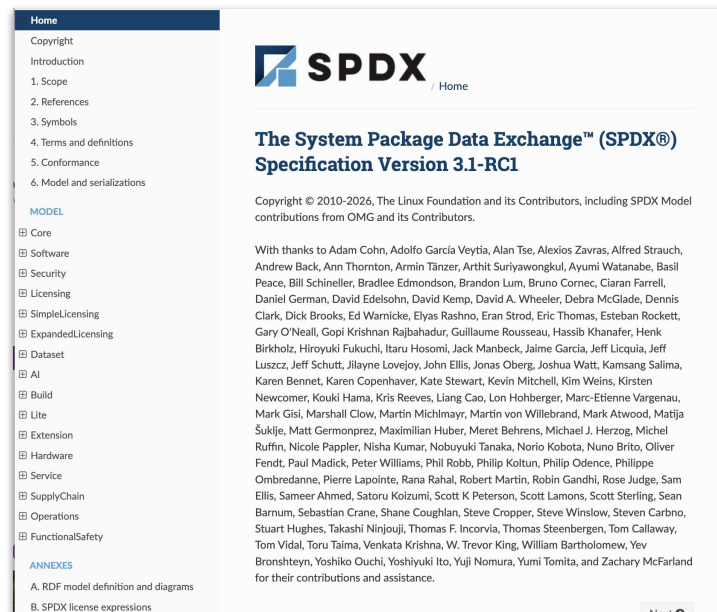


**EU Cyber
Resilience Act**



SPDX 3.1 Release Goals:

- a **knowledge graph** not just a flat file (although it can export them)
- profiles capture the **full supply chain lifecycle**
- provides **Evidence Traceability for Compliance**
- has more Tools to help with **automating the metadata capturing**
- providing more **guides & examples to Help Manufacturers for compliance** such as **EU CRA**



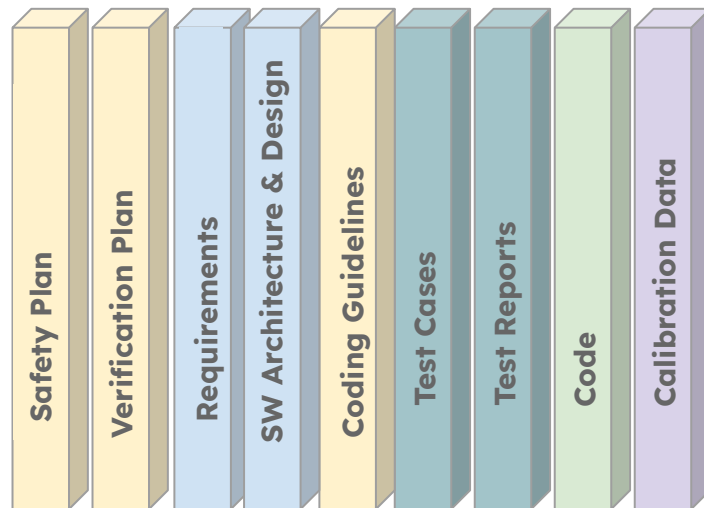
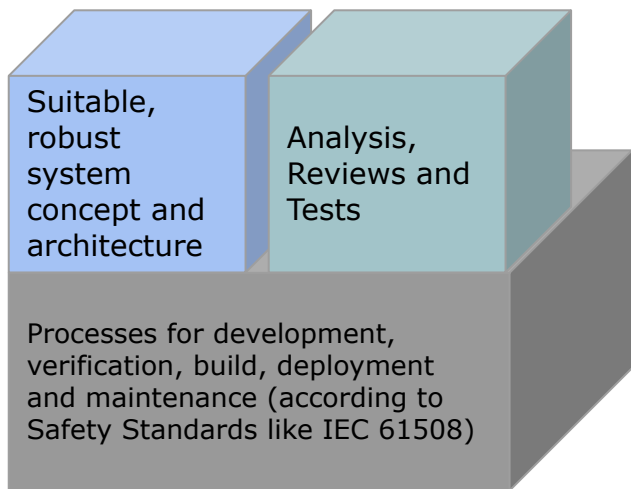
What is FuSa documentation aiming for?



Safety Architecture and Documentation -> Safety Case

You are sure you have a solid structure and concept

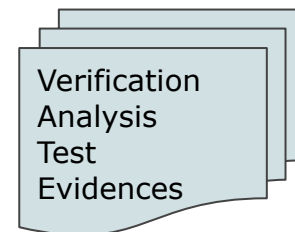
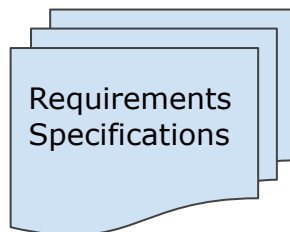
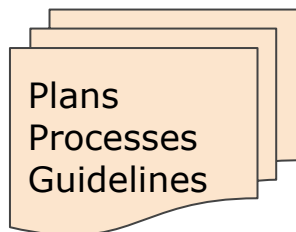
... and you have the evidence that you most probably have implemented everything as planned



FuSa documentation structure

All FuSa related documentation is part of the Safety Case!

Think of all these documents as part of the release - each document is part of the Bill of Material, as is each screw, each microcontroller and each piece of software!



Data Structure of current FuSa projects...



.pdf, .docx,
QMS System,
Wikis

Plans
Processes
Guidelines

One or more
repos, git or
svn based

Code,
Build data,
executables

Zoo of lifecycle
management
systems,
.pdf, .docx

Requirements
Specifications

Verification
Analysis
Test
Evidences

Zoo of lifecycle
management
systems and test
tools,
.pdf, .docx, .xls,
html, code ...

Data Structure of current FuSa projects...

.pdf, .docx,
QMS System,
Wikis

Plans
Processes
Guidelines

One or more
repos, git or
svn based

Code,
Build data,
executables

Traceability breaks
between tools, between
configurations, etc,
impossible to keep up
during updates and
product variants

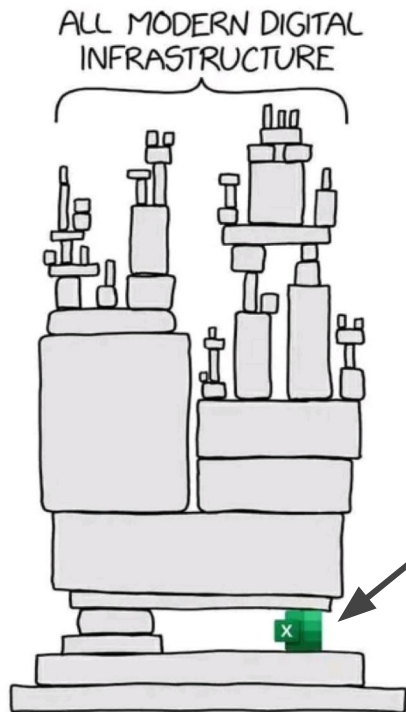
Zoo of lifecycle
management
systems,
.pdf, .docx

Requirements
Specifications

Verification
Analysis
Test
Evidences

Zoo of lifecycle
management
systems and test
tools,
.pdf, .docx, .xls,
html, code ...

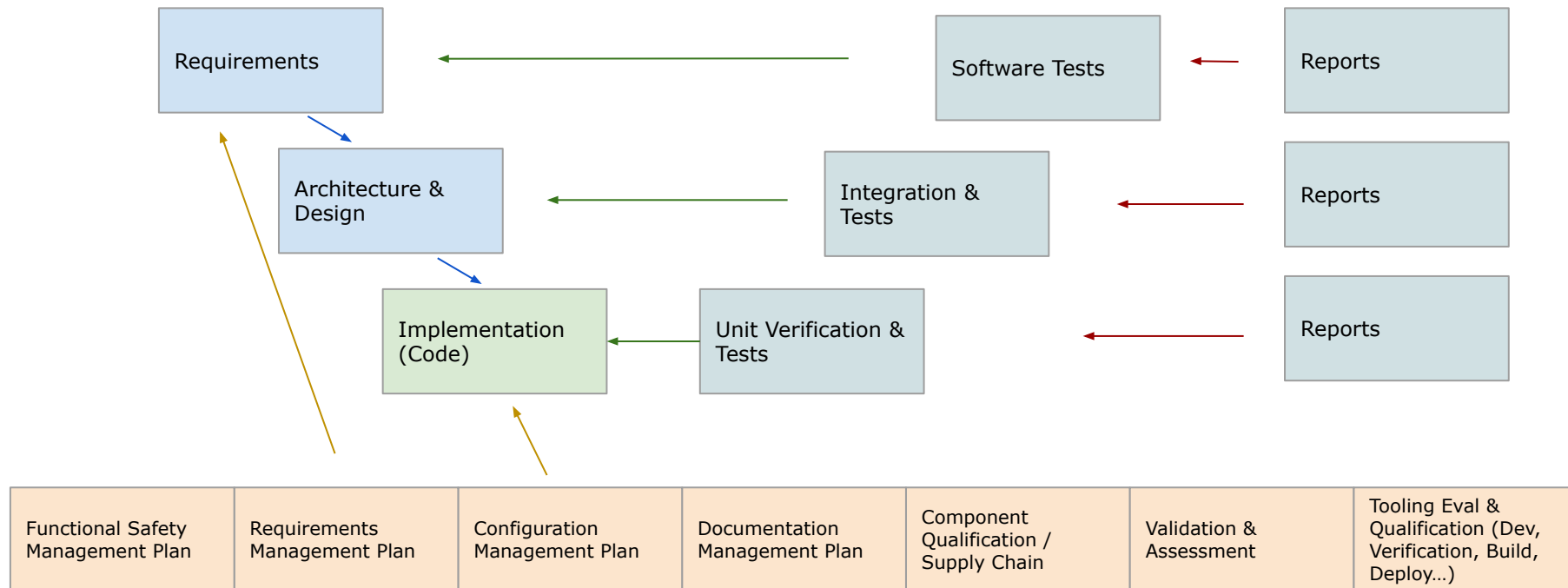
No 1 Safety Information Exchange Format



`draft_2005TemplateSafetyCase_thisproject_final_forTraceingv06.xls`

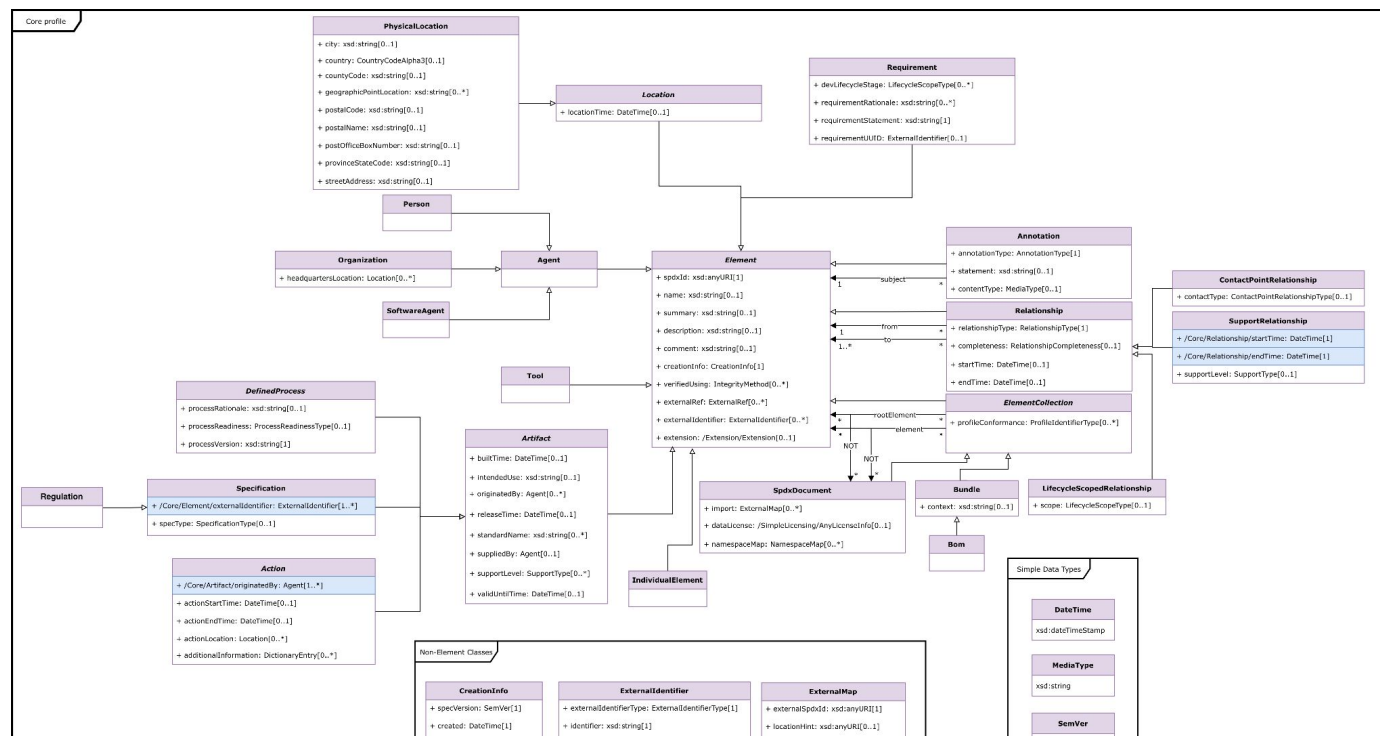
Knowledge Graph in a FuSa Project

Sequence model becomes knowledge model



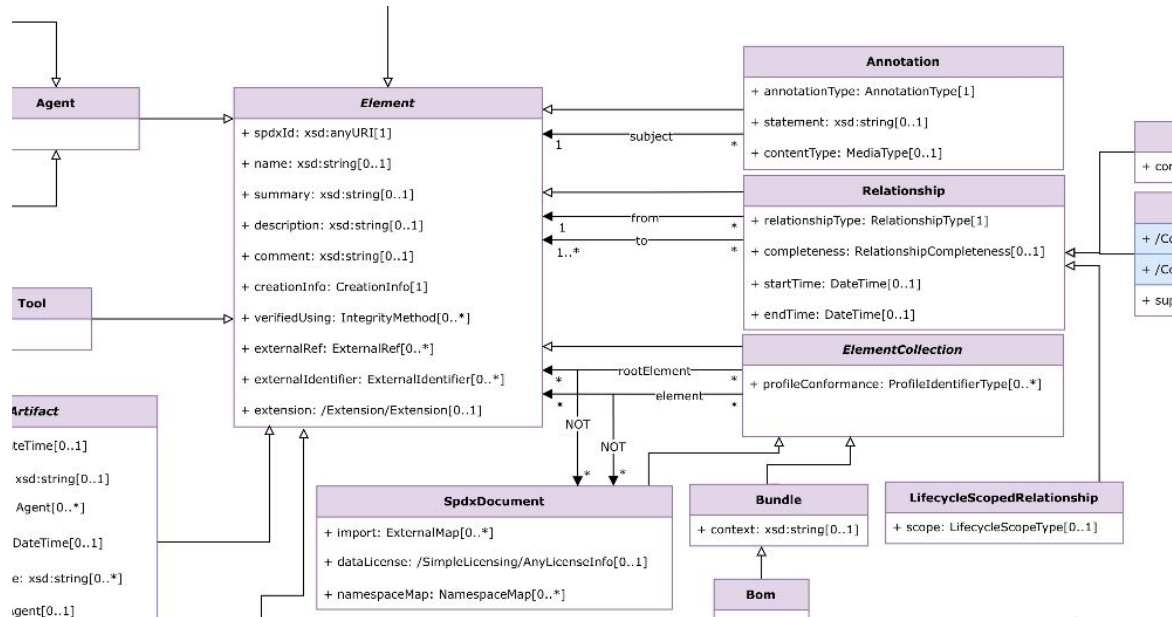
SPDX 3.1-rc1 core model

Core



SPDX 3.1-rc1 Element class

Core



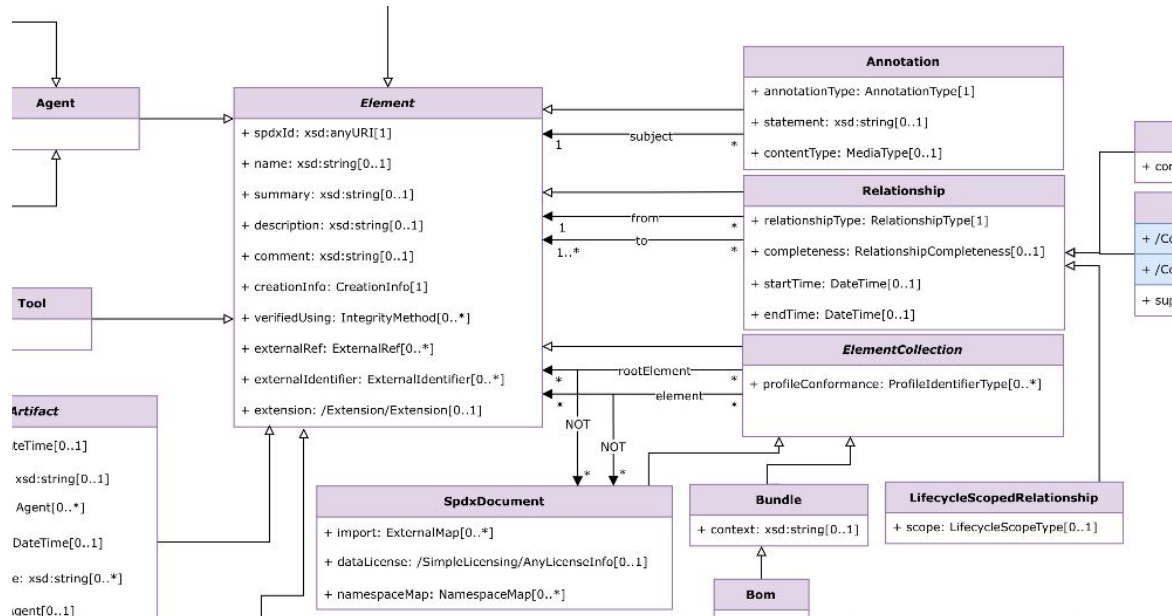
Element:

Basic class, includes e.g. information on

- Creation (who, when)
- ID, name
- description

SPDX 3.1-rc1 Relationship class

Core



Relationship:

Class describing dependencies, like

- hasEvidence
- hasSpecification
- verifiedBy
- traceToDetail

A relationship can be complete or incomplete

SPDX FuSa Use Cases



Use Case 1: Exchange FuSa related information in the supply chain

- Complete project information as Safety Case
- Planning of Functional Safety Management and (Safety) Concept
- Obligations for integration and operation

Use Case 2: Traceability within the project

- Process application and guidelines to implemented work products
- Specifications, requirements, architecture, safety analysis, code, tests and test reports

Use Case 3: Support of automated assessments

- Standardized assessment interfaces

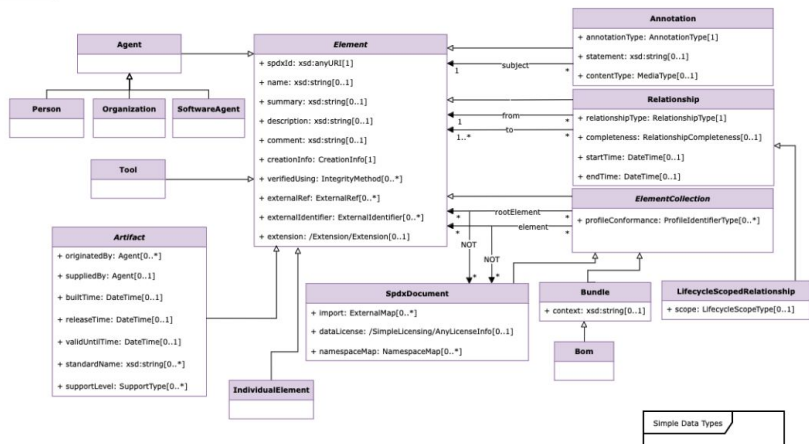
Use Case 4: Support of automated assessments

- Easy identification of components within a system that run safety related loads

Why not use this instead?



Core profile



Core Enumerations			
RelationshipType	ExternalRefType	AnnotationType	ExternalIdentifierType
Meta authoredBy [Element -> Element] describes [Element -> Element] modifiedBy [Element -> Element] other [Element -> Element] comment	altDownloadLocation altMetaType binaryArtifact browser buildMeta buildSystem certificationReport chat componentAnalysisReport documentation dynamicAnalysisReport echinotic exportControlAssessment funding issueTracker license mailingList mavenCentral metrics npm nugget other privacyAssessment productMetadata purchaseOrder qualityAssessmentReport releaseHistory releaseNotes riskAssessment runtimeAnalysisReport securedForwardAttestation securityAdversary securityAdversaryModel securityFile securityOther securityPenTestReport securityPolicy securityThreatModel socialMedia sourceArtifact staticAnalysisReport support vcs vulnerabilityDisclosureReport vulnerabilityExploitabilityAssessment	other review	ExternalIdentifierType cpe22 cpe23 cve email getid other packageId1 securityOther shwid urid uridScheme
RelationshipType	ExternalRefType	AnnotationType	ExternalIdentifierType
Structure contains [Element -> Element]			
Behavioral configure [Element -> Element] decompileTo [Element -> Element] dependsOn [Element -> Element]			
Pedigree copiedTo [Element -> Element] expandTo [Artifact -> Artifact] generates [Artifact -> Artifact] hasAddedFile [Element -> Element] hasDatafile [Element -> Element] hasDeletedFile [Element -> Element]			
Provenance ancestorOf [Element -> Element] availableFrom [Element -> Element] descendantOf [Element -> Element] variant [Artifact -> Artifact]			
Serialization serializesInArtifact [SpdxDocument -> Artifact]			
Build hasDependencyManifest [Element -> Element] hasDistributionArtifact [Element -> Element] hasDocumentation [Element -> Element] hasDynamicLink [Element -> Element] hasExample [Element -> Element] hasInput [Build -> Element] hasOutput [Build -> Element] hasOptionalComponent [Element -> Element] hasOptionalDependency [Element -> Element] hasOutput [Build -> Element] hasPrequisite [Element -> Element] hasProvidedDependency [Element -> Element] hasRequirement [Element -> Element] hasSpecification [Element -> Element] hasStaticLink [Element -> Element] hasTest [Element -> Element] hasTestCase [Element -> Element] hasVariant [Element -> Element] invokeBy [Element -> Agent] packageBy [Element -> Element] patchBy [Element -> Element] useTool [Element -> Element]			
Licensing hasConcludedLicense [SoftwareArtifact -> AnyLicenseInfo] hasDeclaredLicense [SoftwareArtifact -> AnyLicenseInfo]			
Security affects [Vulnerability -> Element] doesNotAffect [Vulnerability -> Element] exploitCreateBy [Vulnerability -> Agent] findBy [Vulnerability -> Agent] foundBy [Vulnerability -> Agent] hasAssessmentFor [Vulnerability -> Element] hasAssociatedVulnerability [Artifact -> Vulnerability] publishedBy [Vulnerability -> Agent] reportedBy [Vulnerability -> Agent] republishedBy [Vulnerability -> Agent] underInvestigationFor [Vulnerability -> Element]			
AI Dataset hasEvidence [Element -> Element] testedOn [Element -> Element] trainedOn [Element -> Element]			
	HashAlgorithm adler32 blake2b256 blake2b384 blake2s256 blake3 cryptaldilithium cryptalryber falcon md2 md4 md5 md6 other sha1 sha224 sha256 [default] sha384 sha512 sha12 sha1_256 sha1_256 sha1_384 sha1_512		
		RelationshipCompleteness complete [default] incomplete noAssertion	LifecycleScopeType build design development other runtime test
			ProfileIdentifierType ai build core dataset expandedLicensing extension lile security simpleLicensing software
		PresenceType no noAssertion yes	SupportType deployed development endOfSupport limitedSupport noAssertion noSupport support

Content for SPDX 3.1 RCs



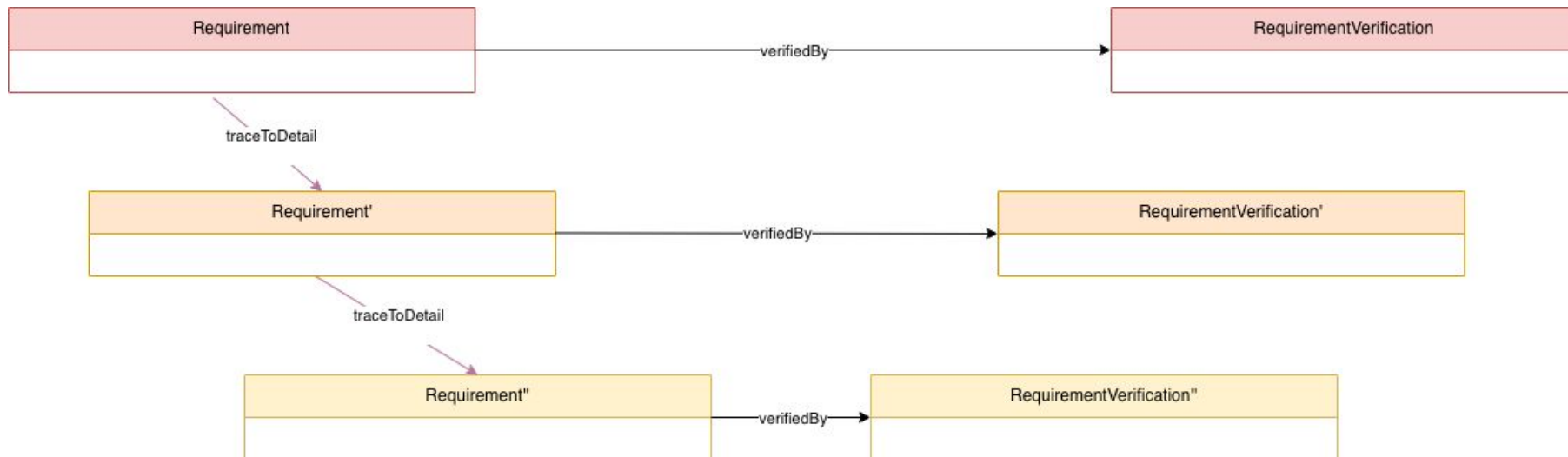
In this release candidate:

- Specification Class
- Requirement Class
- traceToDetail entry to Relationship Types to connect hierarchies of requirements
- EvaluationResult Class
- evidenceRelationship to link evidence to evaluation result
- Role
- Assumption (Assumption of Use)
- Need
- Analysis Trigger and System Impact Analysis

Already in 3.0: Requirement, RequirementVerification

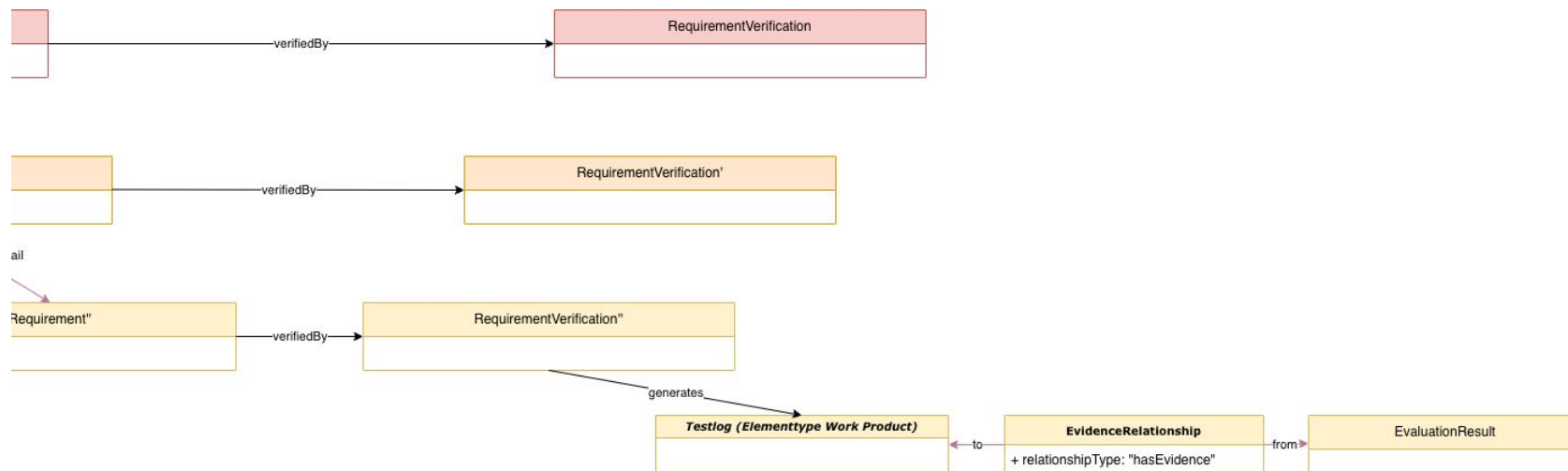
Requirement Relationships

RC SPDX 3.1



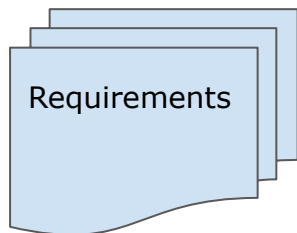
Classes for WPs - EVIDENCE

RC SPDX 3.1

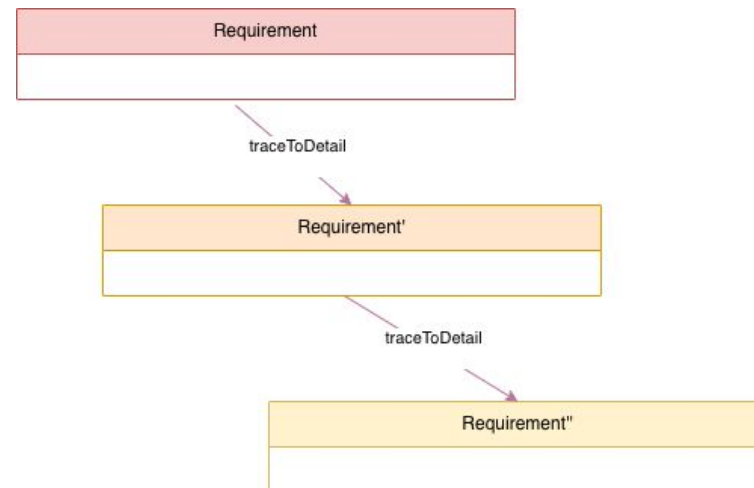
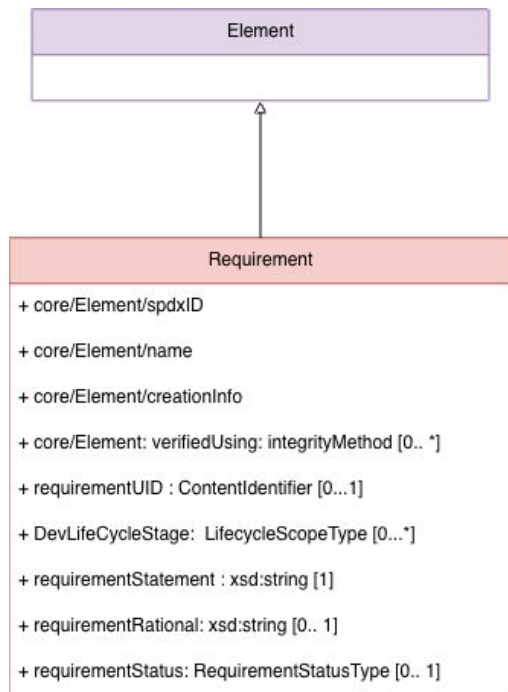


Classes for WPs - REQUIREMENT

RC SPDX 3.1

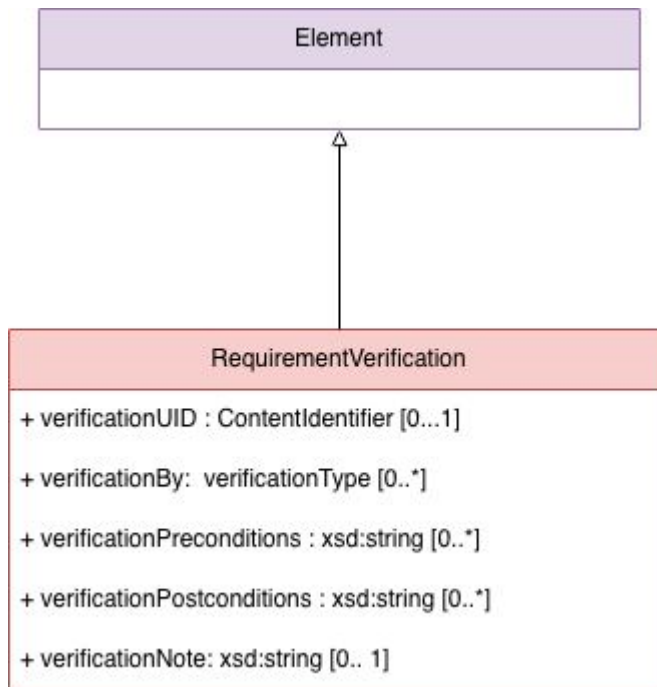
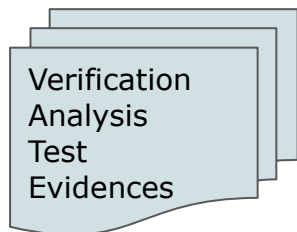


REQUIREMENT class



Classes for WPs - VERIFICATION

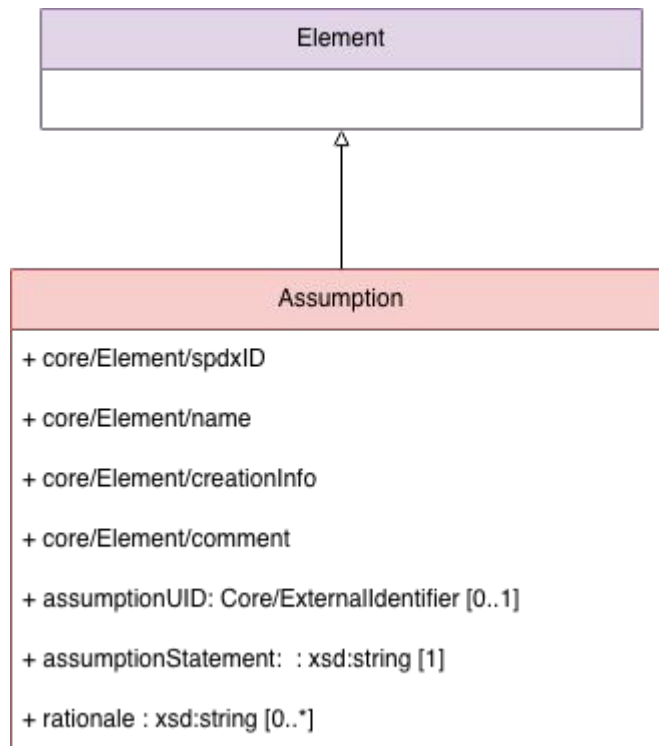
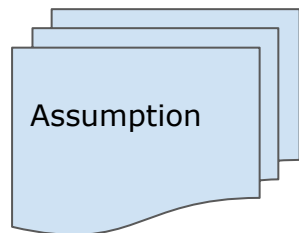
RC SPDX 3.1



enum: verificationType
Review
Inspection
Analysis
Test
Demonstration
Assessment
Audit
other

Classes for WPs - Assumption

SPDX 3.1 RC 2



Example:

Assumption UID: FuSa_A001

Assumption Statement: The *component* is integrated into a system that provides it with input values with ASIL D integrity.

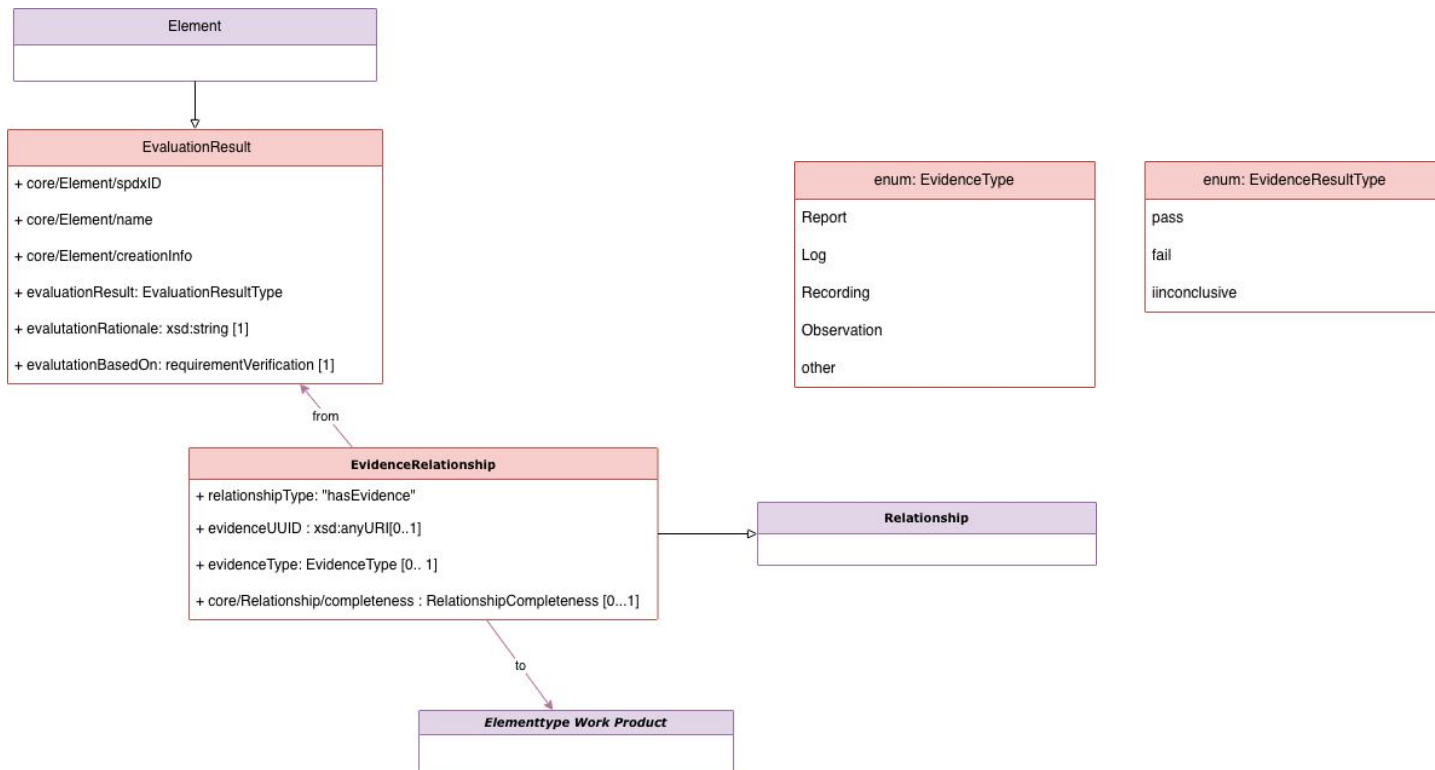
Assumption rationale: The *component* cannot check the integrity of provided input values beyond basic range checks. (Garbage in in garbage out)

Classes for WPs - EVIDENCE

RC SPDX 3.1

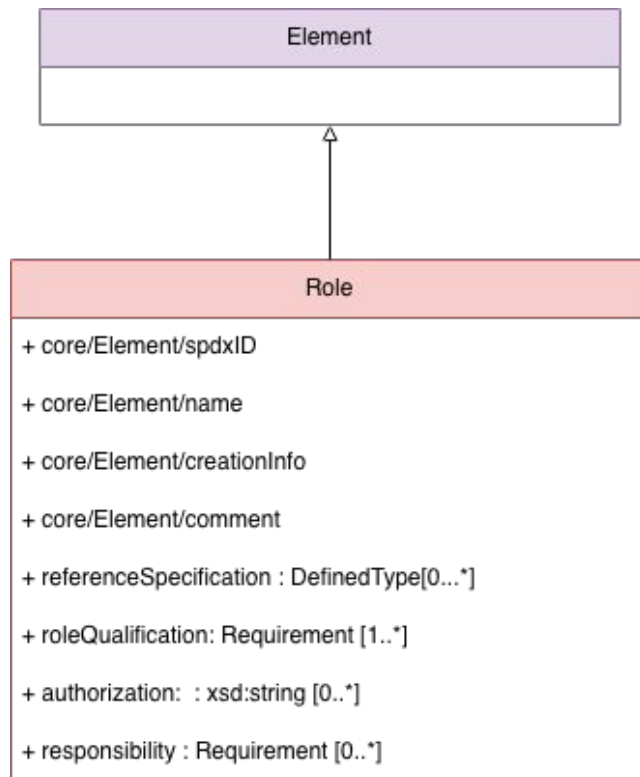
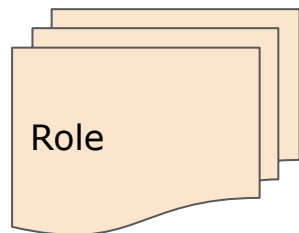


Evidence
(test reports,
build logs,
etc.)



Process related classes - Role

SPDX 3.1 RC 2



Example for Safety Manager Role:

Role Name: FuSa Manager

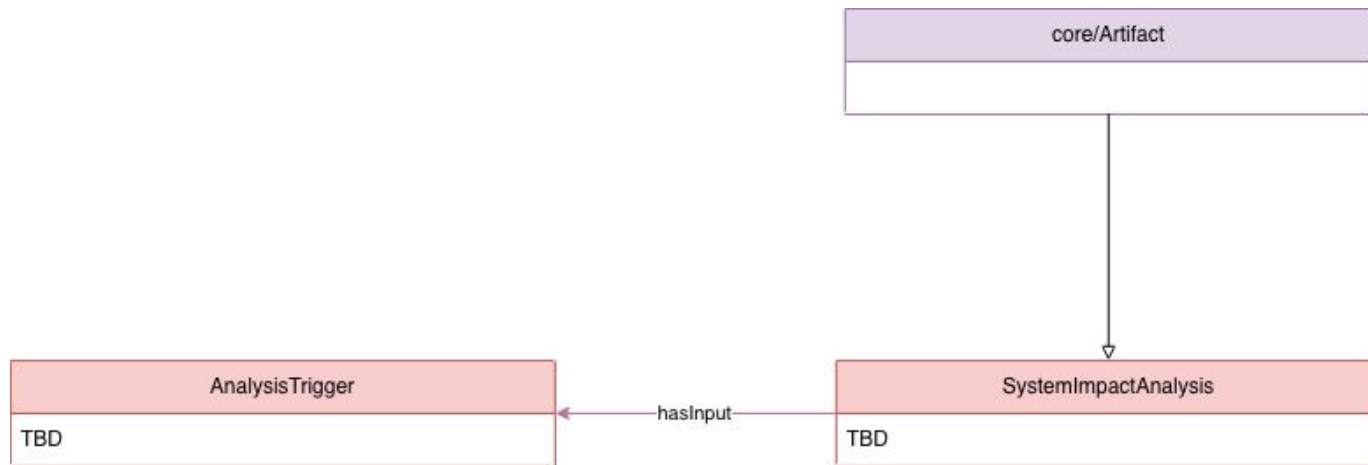
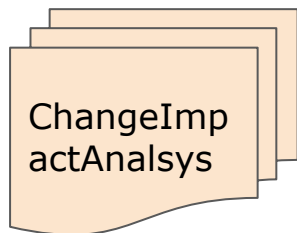
Role Qualification: [Engineering Degree, Systems Engineering experience, FuSa training, FuSa project experience,..]

Role Authorization: [approve safety requirements, approve verification and validation, approve release, ...]

Role Responsibility: [create and maintain safety plan, prioritizes safety related product changes, onboard new safety team members, ...]

WIP - Impact Analysis

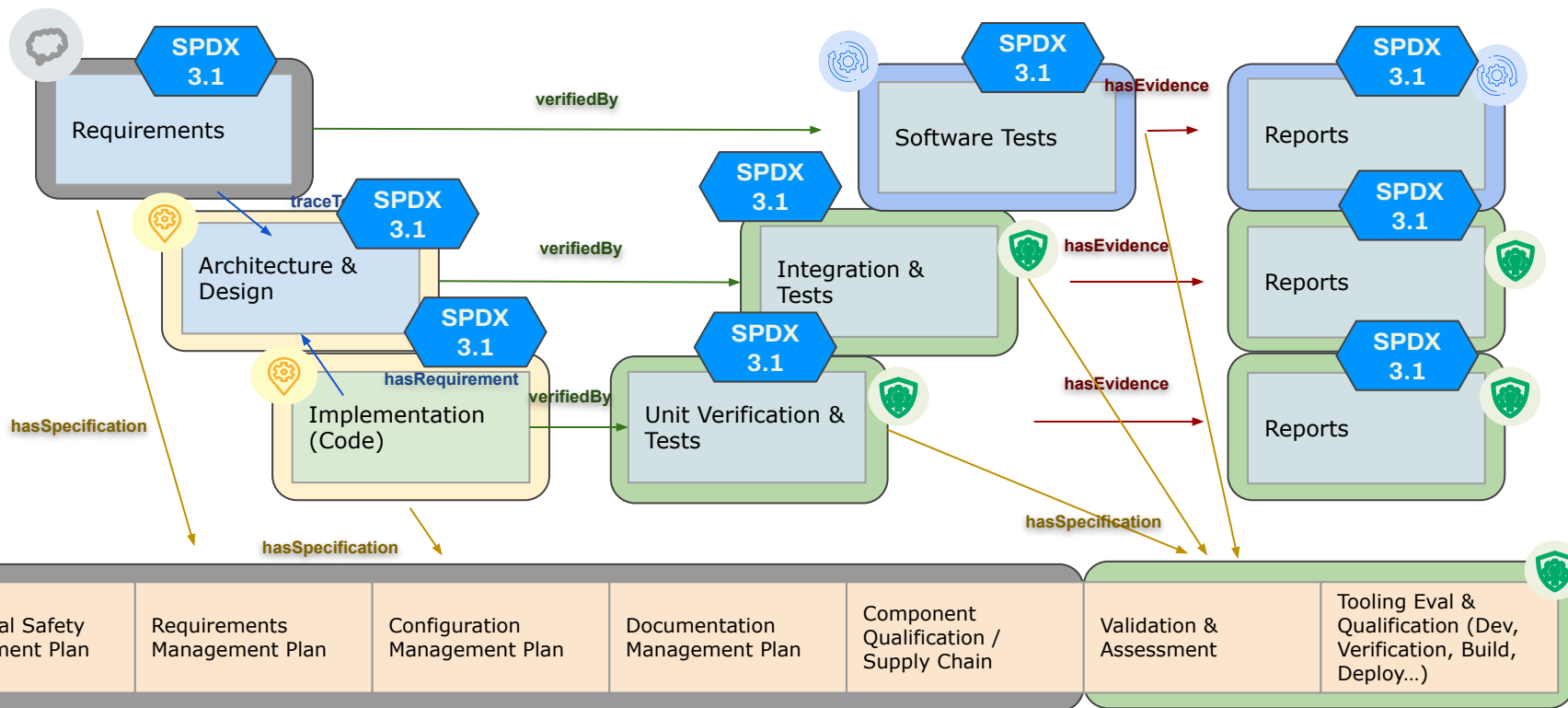
SPDX 3.1 RC 2



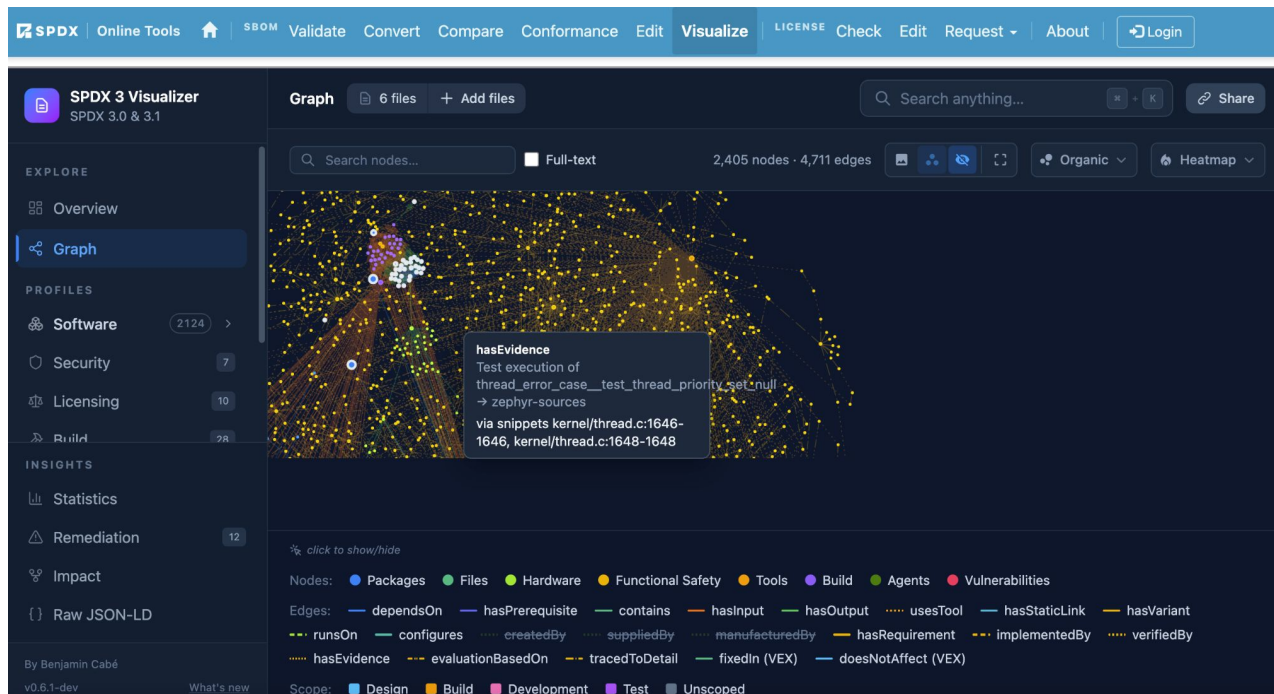
Example use cases:

Impact Analysis based on a CVE, bug report, structural refactorng etc.

Knowledge Graph in a FuSa Project



SDPX Safety SBOM & SPDX Visualizer



https://tools.spdx.org/app/spdx3_viz/

Conclusions (so far)



- What we started for Functional Safety is universally valid for other System Engineering flavours
- Enabling standardized, automated format to exchange safety case documentation
- Tailored SBOMs for design phase, dev phase (source SBOM), runtime and deployed phase
- Reproducible impact analysis
- Tool agnostic information exchange: standardized Safety SBOM
- Compliance as code works as expected

... to be continued



Talk to us:

nicole@alektometis.com

kstewart@linuxfoundation.org

[Mailing List](#)

[Weekly meeting Friday 18:00 CET/CEST](#)

