SBOM News and Siemens Standard BOM

T. Graf

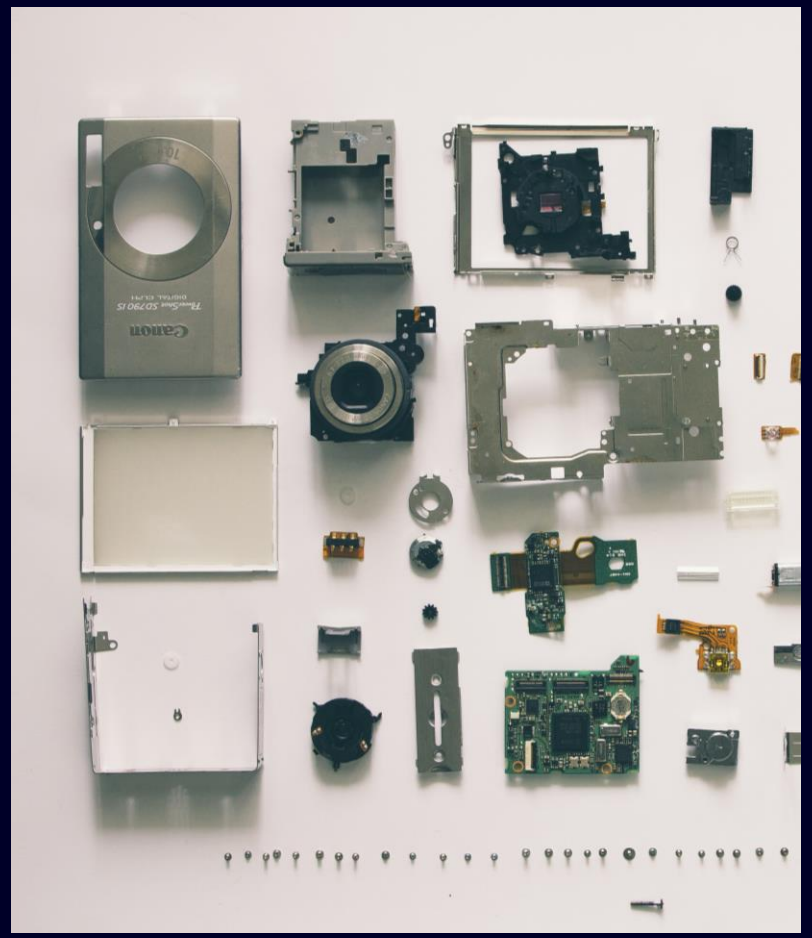
SIEMENS

What is an SBOM - A Software Bill of Materials?

An SBOM is an inventory of components, a list of ingredients that make up a software product. It also ...

- is a formal, machine-readable document
- includes information about the components, especially a unique identifier
- gives the components' hierarchical relationships
- should be comprehensive (or explicitly state where it could not be)
- may include OSS and proprietary software
- can be widely available or access-restricted
- should be generated automatically

The primary purpose of an SBOM is to uniquely and unambiguously identify components and their relationships to one another.



https://pxhere.com/en/photo/969803 (CC0)

SIEMENS

What Type of SBOM – More Details

A good analogy for an SBOM is the nutrition info we have for food

But what exactly do we want to know or which type of SBOM do we like to have:

- Source: created from the development environment, source files, and included dependencies
- Build: generated as part of the process of building the software
- Analyzed: generated through analysis of artifacts after its build. Often this is referred as “3rd party” SBOM
- Deployed: SBOM provides an inventory of software that is presented on a system
- Runtime: SBOM generated through instrumenting the system running the software, also known as “dynamic” SBOM

Example: Yocto Build SBOM (SPDX): 158 MB

Nutrition Facts

8 servings per container	
Serving size	**2/3 cup (55g)**
Amount per serving **Calories**	**230**
	% Daily Value
Total Fat 8g	10%
Saturated Fat 1g	5%
Trans Fat 0g	
Cholesterol 0 mg	0%
Sodium 160 mg	7%
Total Carbohydrate 37g	13%
Dietary Fiber 4g	14%
Total Sugars 12g	
Includes 10g added sugars	
Protein 3g	
Vitamin D 2mcg	10%
Calcium 230 mg	20%
Iron 6 mg	45%
Potassium 253mg	6%

SIEMENS

SBOMs Are Created with a Specific Use Case In Mind

License Compliance

Use SBOMs to ensure that all obligations from OSS and other licenses are met.

- Rich and complete information preferred
- Source code required for all components(!)
- Used internally

Security Vulnerability Monitoring

Use SBOMs to enable monitoring of security vulnerabilities as they emerge.

- Slightly different fields required, such as CPEs
- Can include build tools and test frameworks
- Source code not needed
- Used internally

Regulatory

Use SBOMs to comply with regulations like U.S. EO14028 or the E.U. Cyber Resilience Act.

- Only strictly required content to minimize attack surface
- Source code not needed
- Published

All use cases have in common that the SBOM must be accurate and complete, including all transitive dependencies.

Software Bills Of Materials Are About Interoperability

Source Ecosystems
Libraries, Open-Source Software, Transitive Dependencies

Partner Companies
Components provided by partners may come with their own SBOMs.

3rd Party Vendors
Software Products, e.g. PostgreSQL, Nginx, …

Siemens Business Units
Our own internally developed Inner Source components have SBOMs.

Container Images
Built and/or consumed, Base Images, Layers

… and more
The list goes on.

SIEMENS

1 Nearly every team is different

- No silver bullet
- No universal automation approach

2 We need to have a set of tools

- In order to simplify SBOM generation / software license automation for our developers
- Inner source or open source

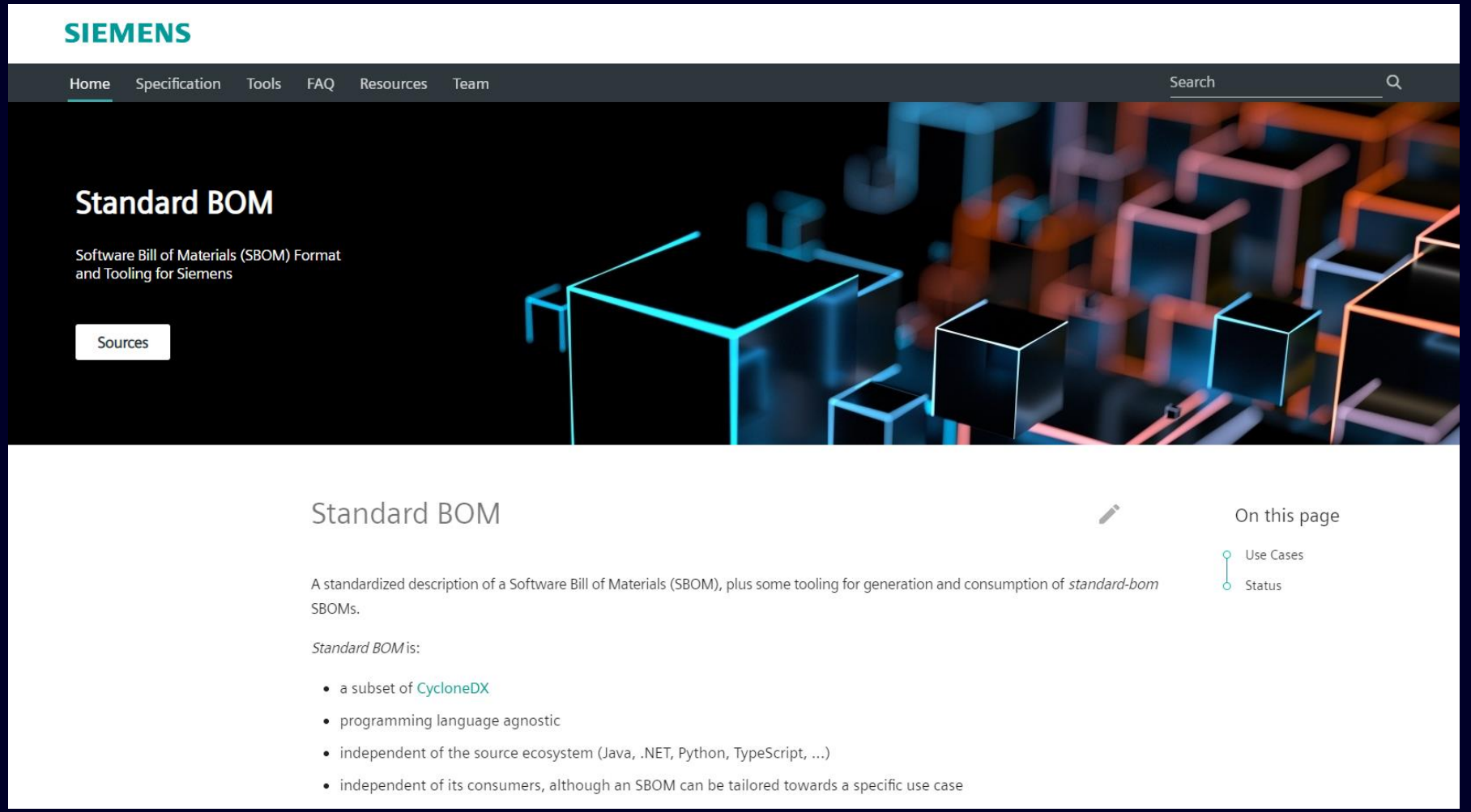
3 We need to have simple to use libraries

- When there are no specific tools
- When teams need customization
- Inner source or open source

SIEMENS

A common SBOM format and tooling for Siemens would be nice!

https://sbom.siemens.io



SIEMENS

What Is This Siemens Standard BOM?

The Siemens Standard BOM is a standardized SBOM format with tooling for Siemens.

It is

- a subset of OWASP CycloneDX
- programming language agnostic (It's just JSON)
- independent of the source ecosystem (Java, .NET, Python, TypeScript, ...)
- independent of its consumers, although an SBOM can be tailored towards a specific use case (for example, it works with different Siemens software clearing toolchains)

SIEMENS

Why Should We Have Standard BOM Rather Than Plain CycloneDX?

- Standard BOM is a proper subset of CycloneDX
- SBOM components are presented in *list form*, not as a tree
- Custom properties are not random free-form Strings as per CycloneDX, but elements from the Siemens Property Taxonomy for CycloneDX. CycloneDX reserves a `siemens` namespace for Standard BOM.
- *Component Sources* can be specified
- A Standard BOM Package bundles the SBOM document with any referenced files, such as component sources or binary archives

```
"properties" : [ {
    "name" : "siemens:direct",
    "value" : "true"
  }, {
    "name" : "siemens:filename",
    "value" : "commons-codec-1.13.jar"
  }, {
    "name" : "siemens:primaryLanguage",
    "value" : "Java"
  }, …
],
```

```
"externalReferences" : [
  {
    "type" : "distribution",
    "url" : "file:sources/a11bdc0e8f…a35c23e197498d/log4j-api-2.11.2-sources.jar",
    "comment" : "source archive (local copy)",
    "hashes" : [ … ]
  }, {
    "type" : "distribution",
    "url" : "https://repo.maven.apache.org/maven2/…/log4j-api-2.11.2-sources.jar",
    "comment" : "source archive (download location)",
    "hashes" : [ … ]
  }, …
```

Siemens Standard BOM is OPEN-SOURCE

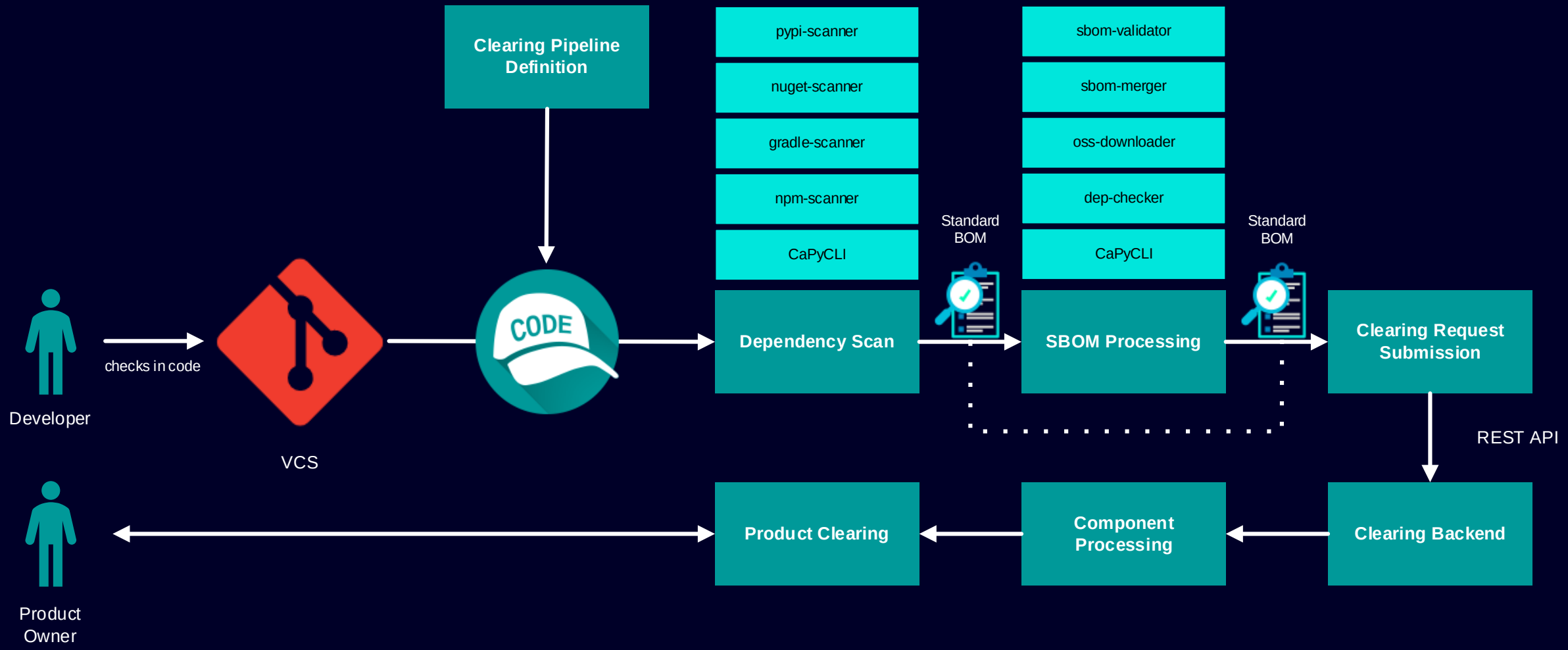
SIEMENS

Example:
BOM Entry for Java Library

```
{
  "type": "library",
  "author": "Henri Yandell <bayard@apache.org>, Tim OBrien …",
  "group": "commons-codec",
  "name": "commons-codec",
  "version": "1.13",
  "purl": "pkg:maven/commons-codec/commons-codec@1.13?type=jar",
  "description": "The Apache Commons Codec package contains …",
  "hashes": [ … ],
  "licenses": [ {
    "license": {
      "name": "Apache License, Version 2.0",
      "url": "https://www.apache.org/licenses/LICENSE-2.0.txt"
    } } ],
  "externalReferences": [ …
    {
      "type": "distribution",
      "url": "file:sources/2…d/commons-codec-1.13-sources.jar",
      "comment": "source archive (local copy)",
      "hashes": [ … ]
    }, {
      "type": "website",
      "url": "https://commons.apache.org/proper/commons-codec/"
    },
    {
      "type": "vcs",
      "url": "https://github.com/apache/commons-codec"
    }
  ],
  "properties": [
    {
      "name": "siemens:direct",
      "value": "true"
    }, {
      "name": "siemens:primaryLanguage",
      "value": "Java"
    }, {
      "name": "siemens:thirdPartyNotices",
      "value": "Apache Commons Codec\nCopyright 2002-2019 The ↲
        Apache Software Foundation\nThis product includes software ↲
        developed at\nThe Apache Software Foundation ↲
        (https://www.apache.org/).\nsrc/test/org/apache/commons↲
        /codec/language/DoubleMetaphoneTest.java\ncontains test …"
    }
  ],
  "copyright": "Copyright 2002-2019 The Apache Software …",
  "bom-ref": "pkg:maven/commons-codec/commons-codec@1.13?type=jar"
}
```


SIEMENS

Standard BOM is Great For Automated Pipelines Which Need SBOMs
Example: Software License Compliance



SIEMENS

Standard BOM Profiles

U.S. EO14028 and the E.U. Cyber Resilience Act require us in the near future to provide SBOMs for customers. Main reason is an improved security vulnerability handling.

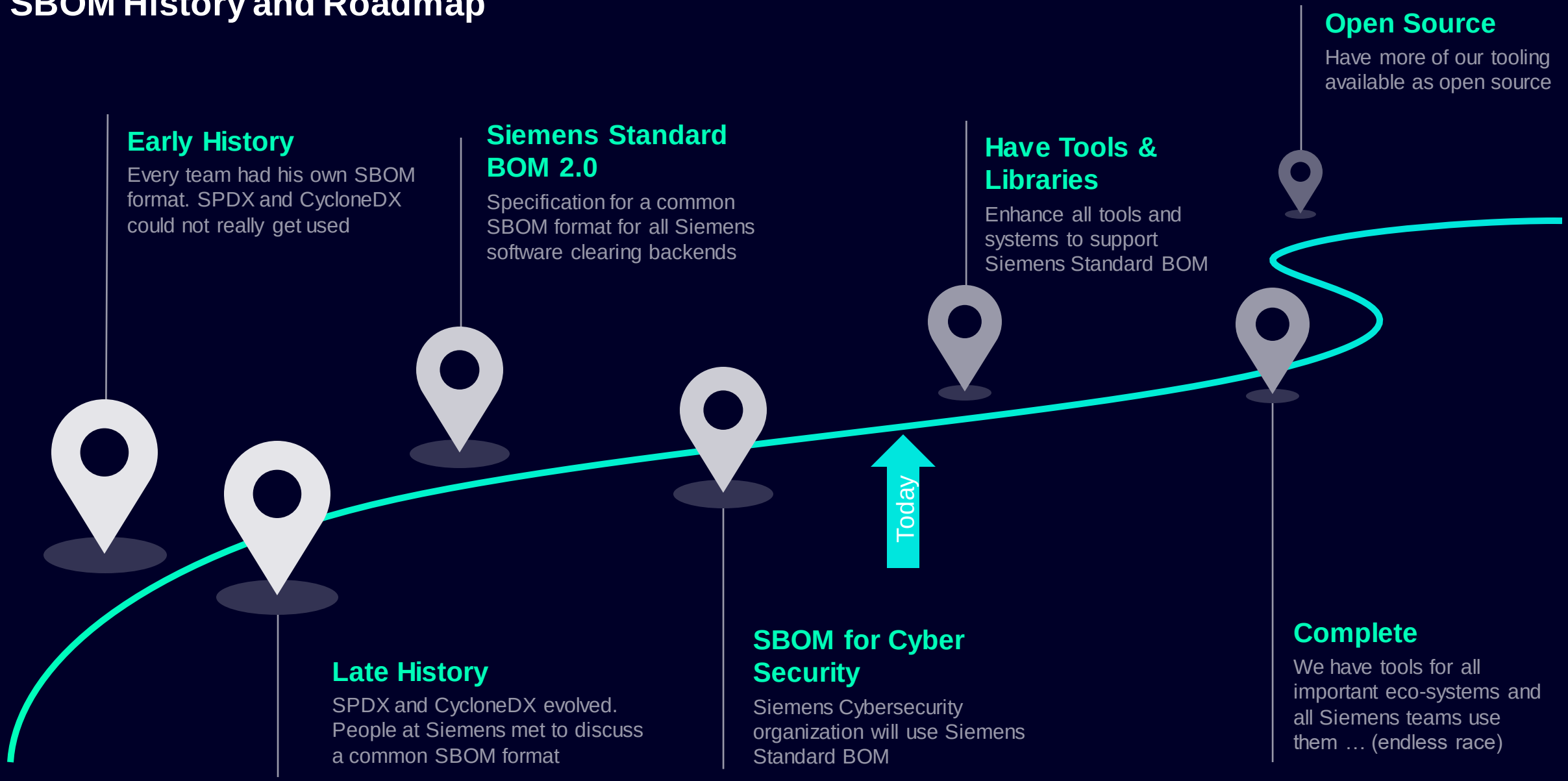
Again we need to decided which information needs to be part of the SBOM. For the Siemens Standard BOM we introduced profiles:

- Clearing – for SBOM used Siemens internally. Here we want as much content as possible, all of which can be helpful in clearing
- External – publicly distributed SBOMs, for example for compliance with regulations. Here, the SBOM content is minimized to reduce a potential attack surface.

Have a look at https://sbom.siemens.io/v2/profiles.html for details.

Images from https://en.wikipedia.org/ are public domain

SBOM History and Roadmap



SIEMENS

We collaborate across organizations

Thomas Graf
Principal Key Expert Software Clearing
SI BP

Thomas Jensen
Senior Software Architect
DI PA

Alexander Gschrei
IT Solution Expert - SCP
DI IT

Florian Greinacher
Senior DevOps Engineer
IT APS

Gernot Hillier
Senior Linux Engineer
T CED

SIEMENS


Q&A
time

Resources

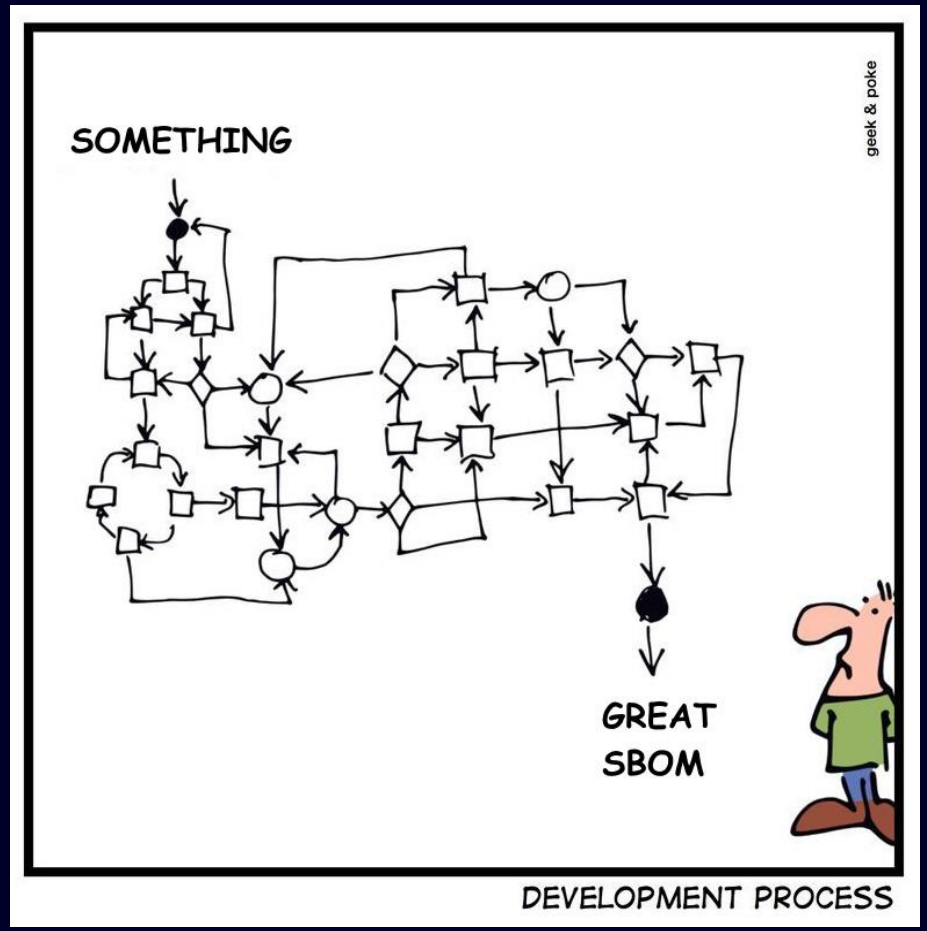
- Siemens Standard BOM: https://sbom.siemens.io
- Code (Inner Source): https://code.siemens.com/sbom
- OWASP CycloneDX: https://cyclonedx.org/
- Siemens Property Taxonomy for CycloneDX: https://github.com/siemens/cyclonedx-property-taxonomy
- CaPyCLI – Clearing Automation Python Command Line Tool https://github.com/sw360/capycli

- https://github.com/CycloneDX/cyclonedx-property-taxonomy/pull/24
- https://github.com/CycloneDX/specification/issues/98
- https://github.com/CycloneDX/cyclonedx-python/pull/534
- https://github.com/CycloneDX/cyclonedx-python-lib/pull/325
- https://github.com/siemens/cyclonedx-property-taxonomy
- https://github.com/anchore/syft/issues/1700
- https://github.com/package-url/purl-spec/pull/57

SIEMENS

Contact

Thomas Graf
Principal Key Expert Software Clearing
SI BP R&D DB SEC 2
Mobile +49 174 19 44 64 4
E-mail thomas.graf@siemens.com



Based on https://geek-and-poke.com/

SIEMENS

BACKUP

SIEMENS

The „Standard BOM Package“

Used for handling file system references in the SBOM.

Objective: Self-contained package

All external references used in the SBOM must be either

- URLs of publicly available resources on the Internet, or
- a relative file system path.

Resources referenced via relative paths become part of the self-contained "Standard BOM Package", which is a ZIP file or file system folder.

```
sbom.json
+--- binaries
|    +--- 77100a62c2e6f04b53977b9f541044d7d722693d
|    |    `--- some-binary.jar
|    +--- 8031352b2bb0a49e67818bf04c027aa92e645d5c
|    |    `--- another-binary.jar
|    `--- (... more ...)
`--- sources
     +--- 6bb10559db88828dac3627de26974035a5dd4ddb
     |    `--- some-binary-sources.jar
     +--- 4d44e4edc4a7fb39f09b95b09f560a15976fa1ba
     |    `--- another-binary-sources.jar
     `--- (... more ...)
```