CISA SBOM-A-RAMA

DEC 15TH PRESENTATIONS

SBOM@cisa.dhs.gov
Dec 15, 2021

JOSH CORMAN

@JOSHCORMAN
@IAMTHECAVALRY

HISTORY OF SBOM

(SEE VIDEO)

Hello, world!

SBOM

Roles and Benefits

CISA SBOM-A-RAMA

12/15/2021

Audra Hatch


R 1 3 5
2 4

Produce

The person or organization that creates a software component or software for use by others

[write/create/assemble/package]

Produce

The person or organization that creates a software component or software for use by others

[write/create/assemble/package]

Choose

The person or organization that decides the software, products, and/or suppliers for use

[purchase/acquire/source/select/approve]

Produce

The person or organization that creates a software component or software for use by others

[write/create/assemble/package]

Choose

The person or organization that decides the software, products, and/or suppliers for use

[purchase/acquire/source/select/approve]

Operate

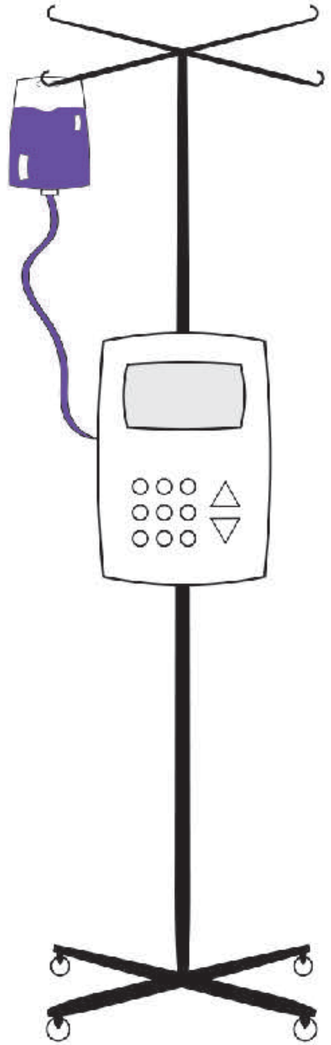
The person or organization that operates the software component or software

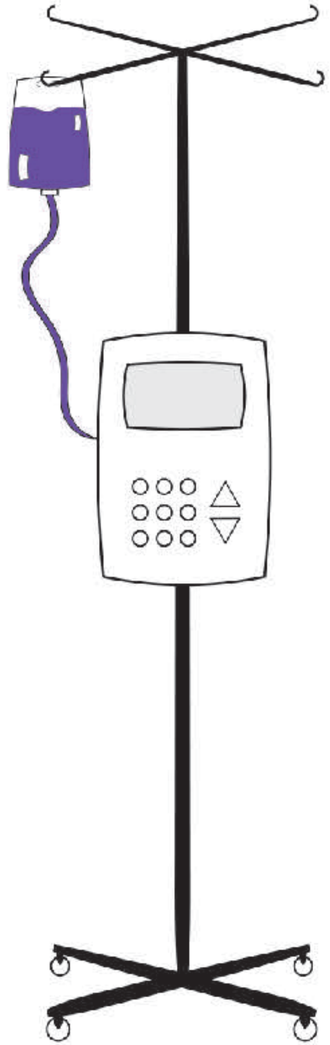
[uses/monitor/maintain/defend/respond]

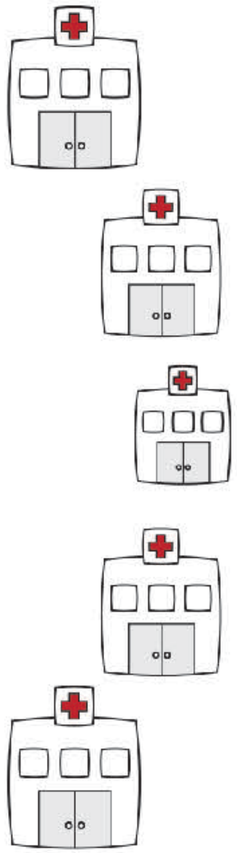
Benefits	Produce	Choose	Operate
Cost	Less unplanned, unscheduled work	A more accurate total cost of ownership	More efficient administration
Security Risk	Avoid known vulnerabilities	Easier due diligence	Faster identification and resolution. Know if and where specific software is affected.
License Risk	Quantify and manage licenses and associated risk	Easier due diligence	More efficient, accurate response to license claims
Compliance Risk	Easier risk evaluation. Identify compliance requirements earlier in lifecycle	More accurate due diligence, catch issues earlier in lifecycle	Streamlined process
High Assurance	Make assertions about artifacts, sources, and processes used	Make informed, attack-resistant choices about components	Validate claims under changing and adversarial conditions

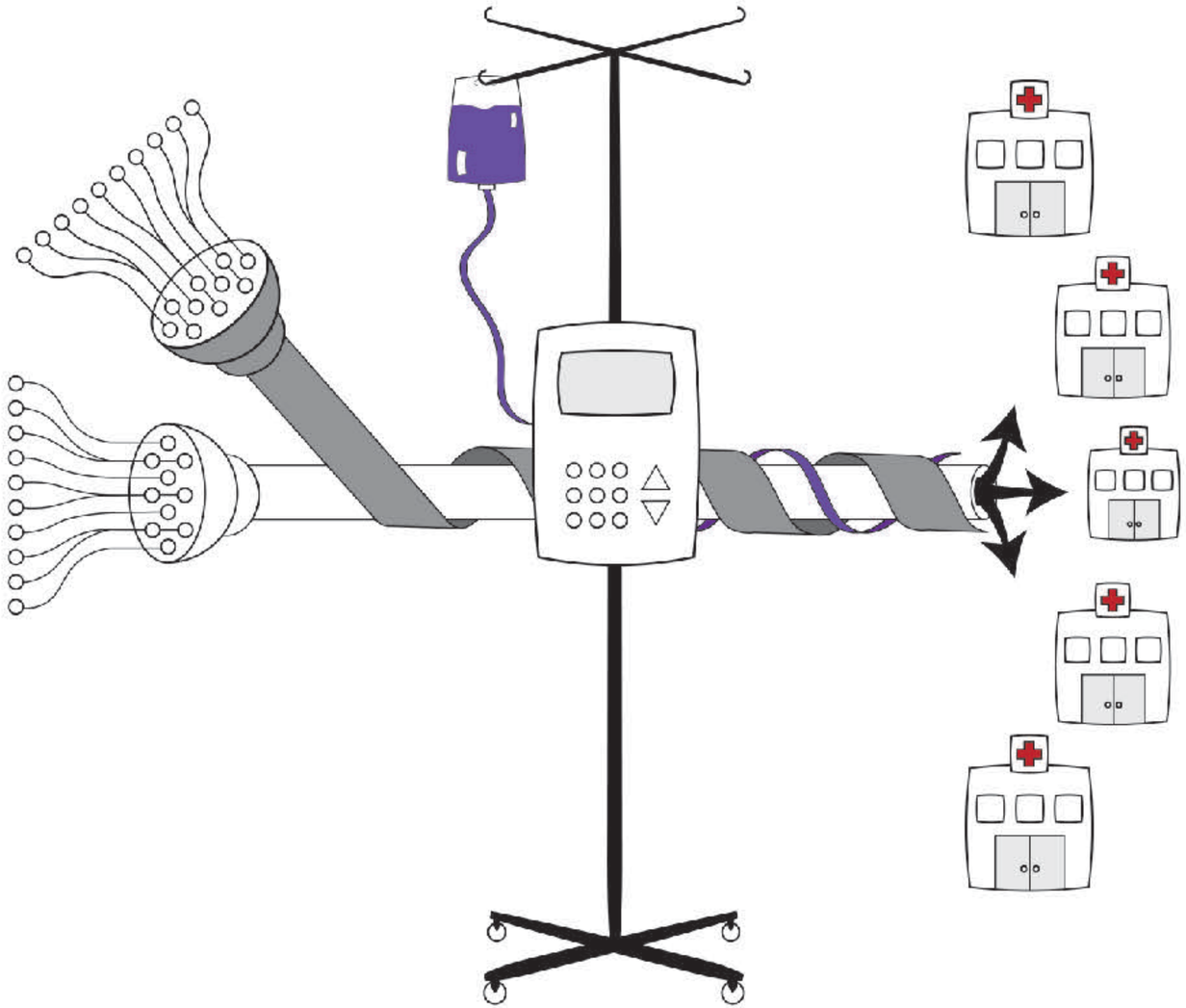
Benefits	Produce	Choose	Operate
Cost	**Less unplanned, unscheduled work**	**A more accurate total cost of ownership**	More efficient administration
Security Risk	**Avoid known vulnerabilities**	**Easier due diligence**	**Faster identification and resolution. Know if and where specific software is affected.**
License Risk	Quantify and manage licenses and associated risk	Easier due diligence	More efficient, accurate response to license claims
Compliance Risk	Easier risk evaluation. Identify compliance requirements earlier in lifecycle	More accurate due diligence, catch issues earlier in lifecycle	Streamlined process
High Assurance	Make assertions about artifacts, sources, and processes used	**Make informed, attack-resistant choices about components**	Validate claims under changing and adversarial conditions

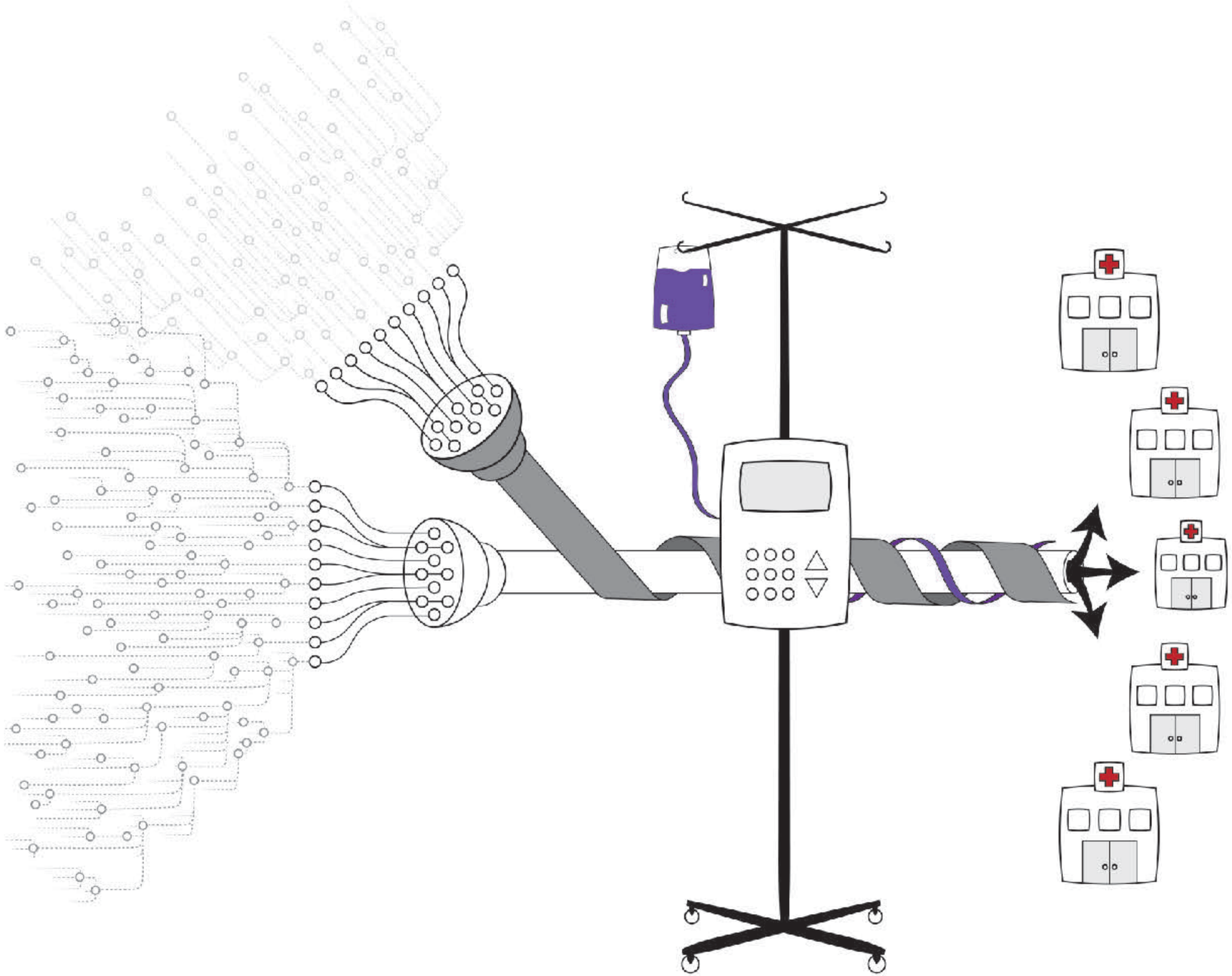
Ecosystem, Network Effects, and Public Health Benefits

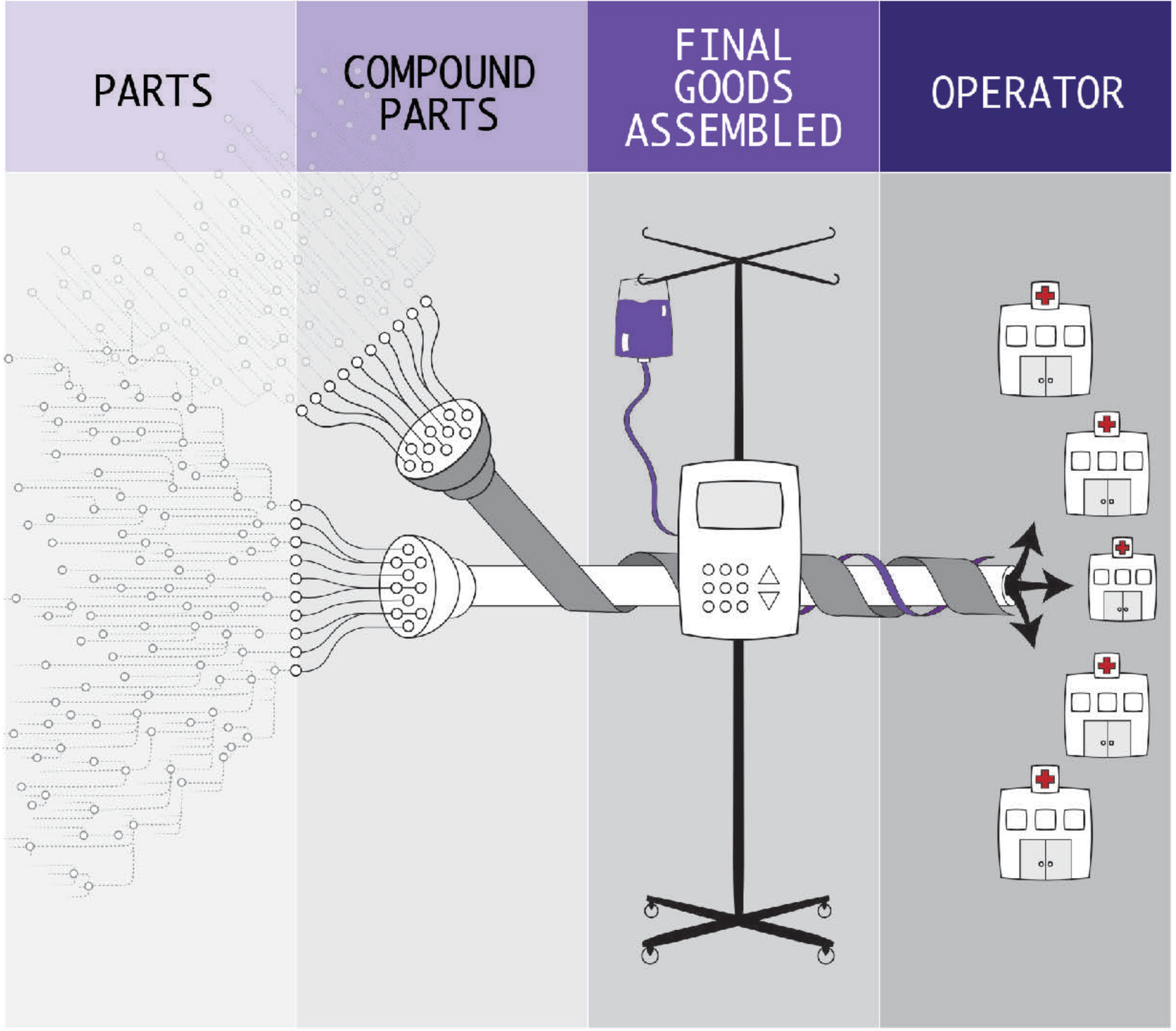


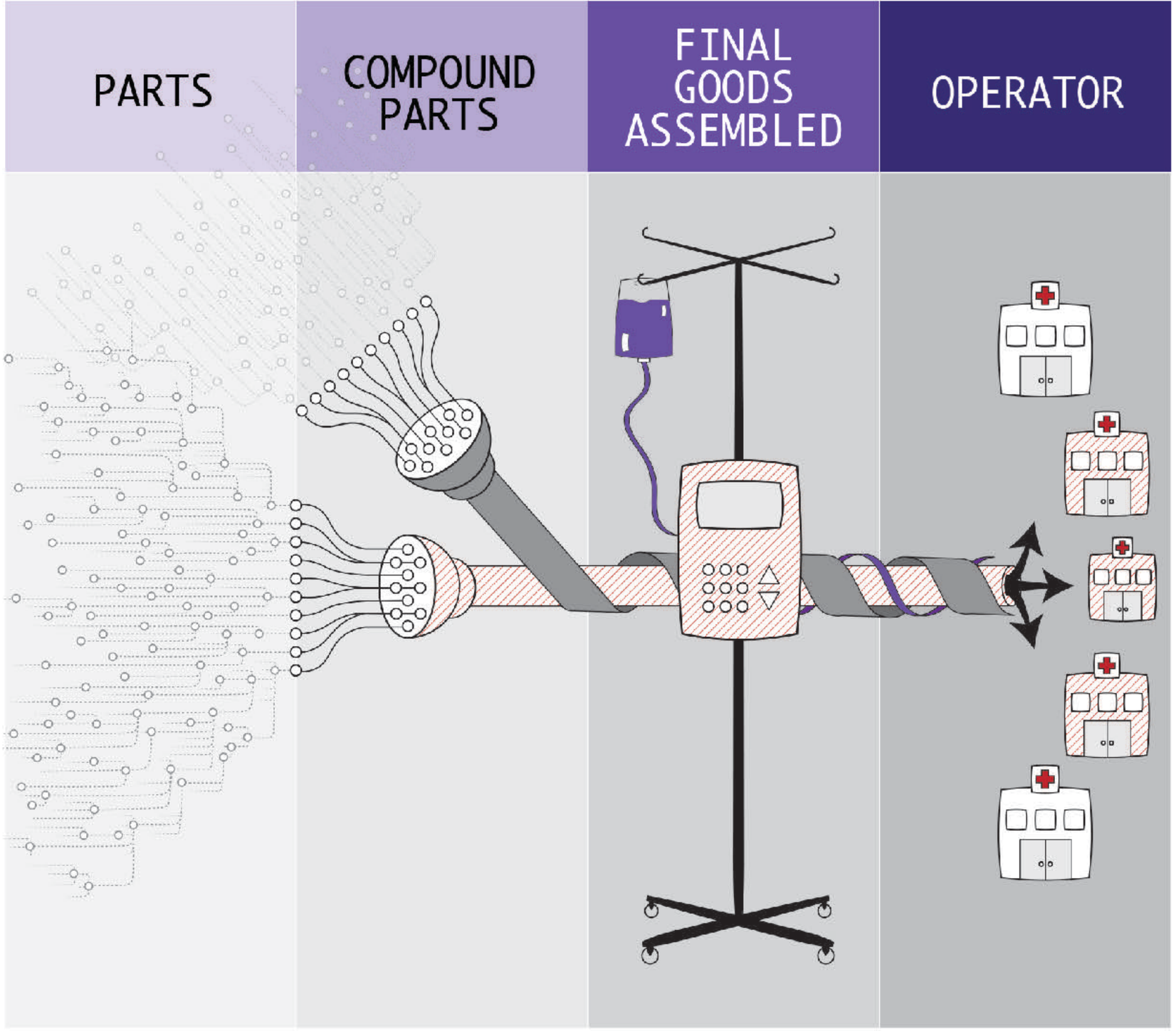


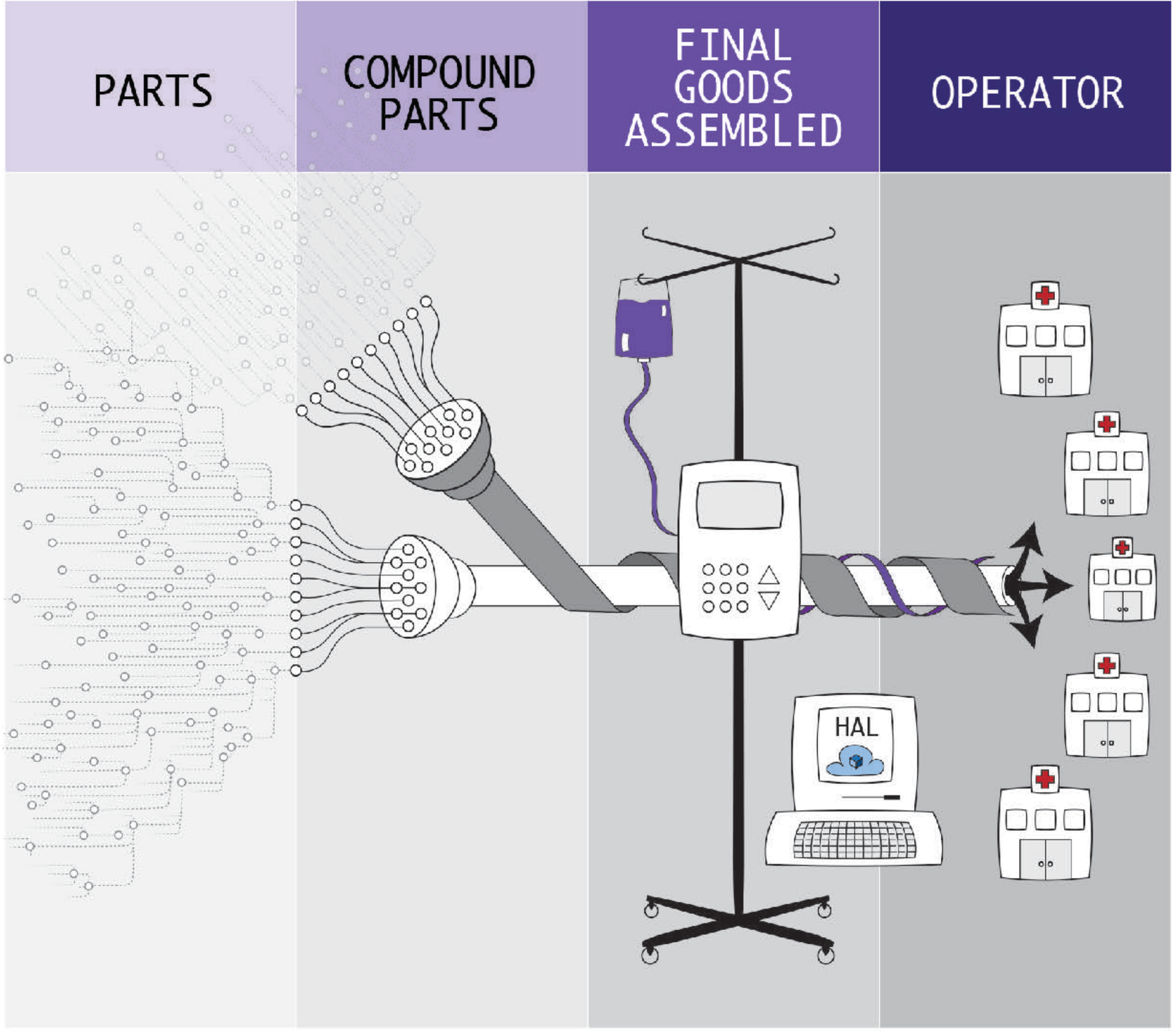


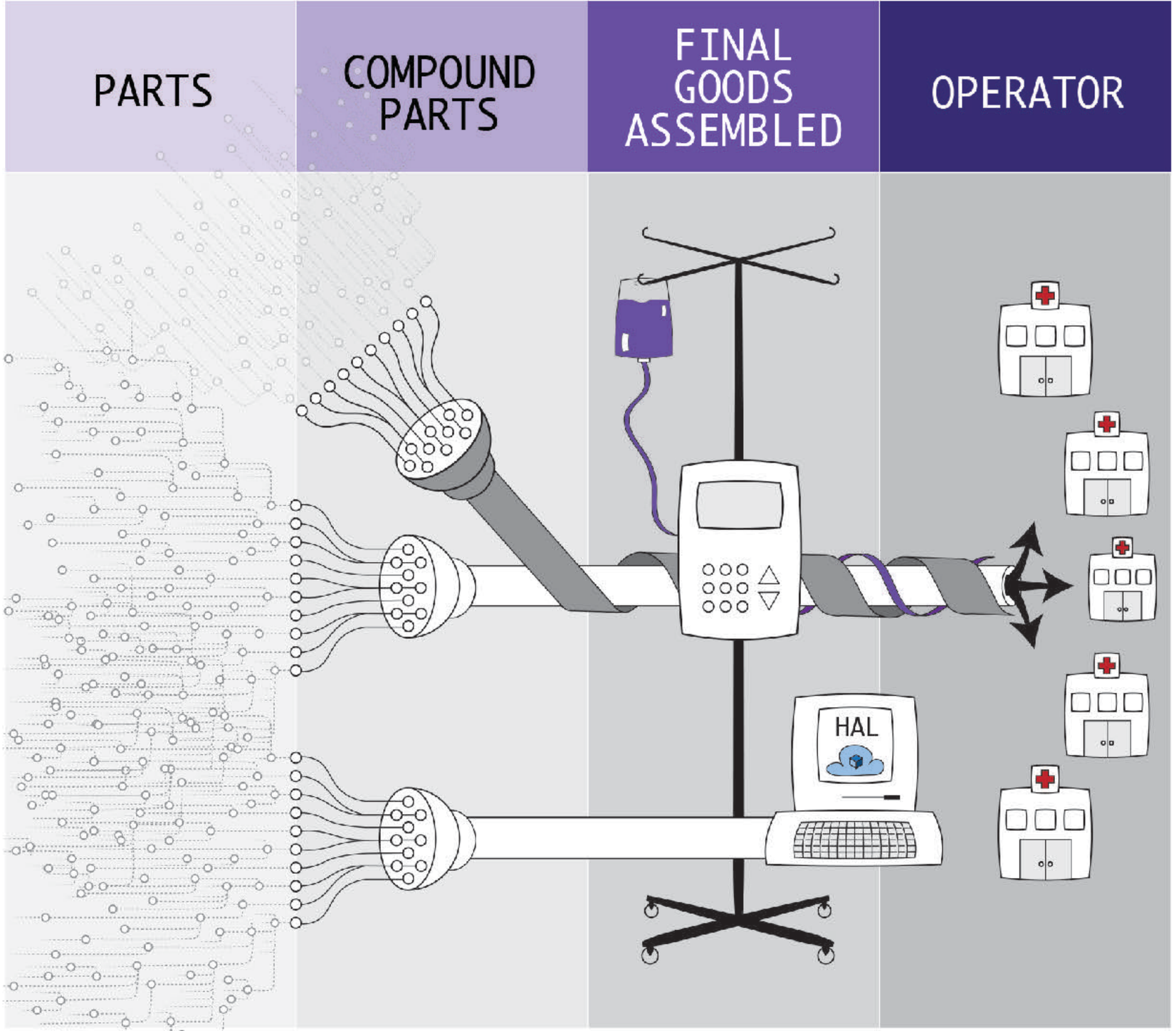


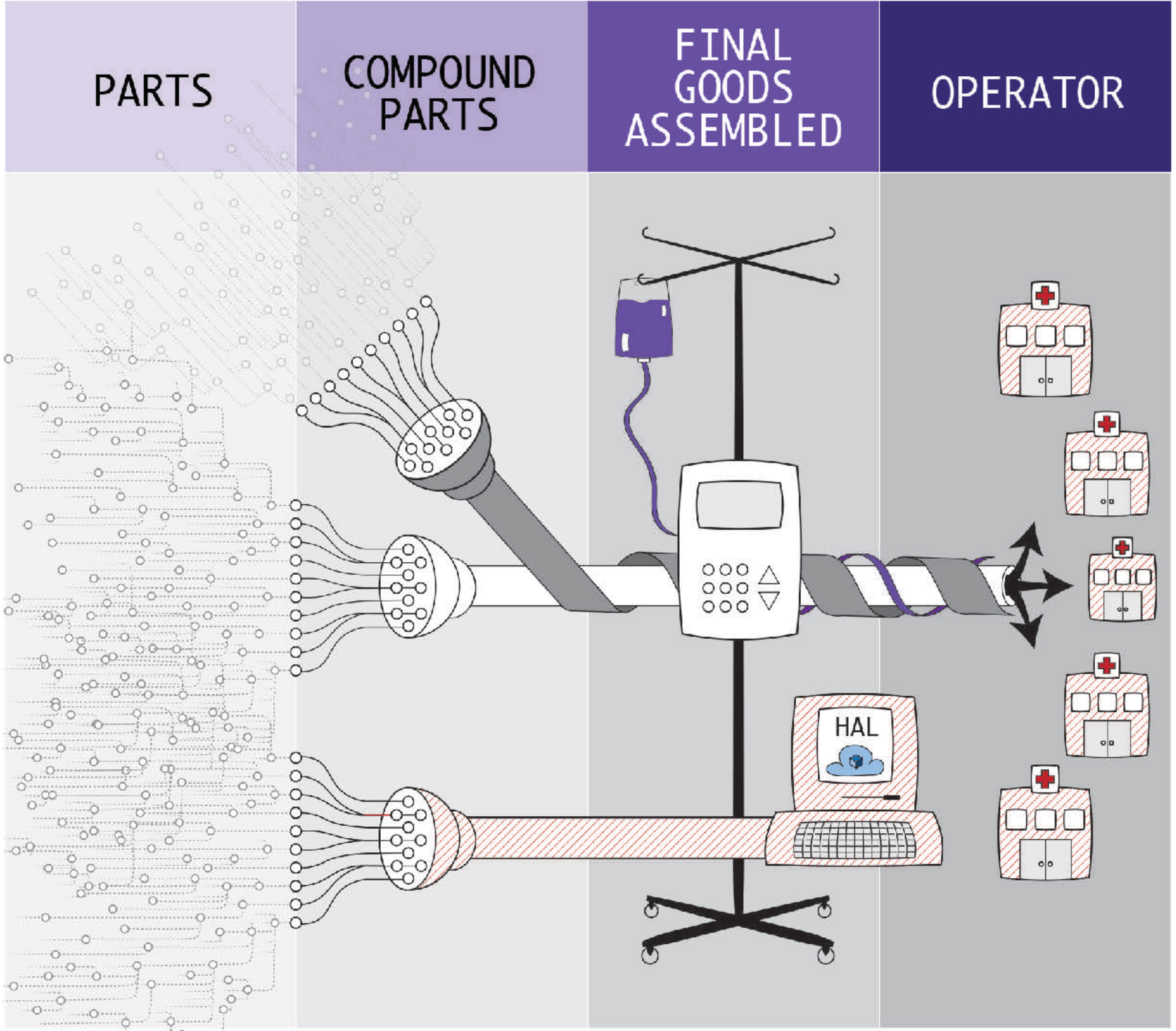


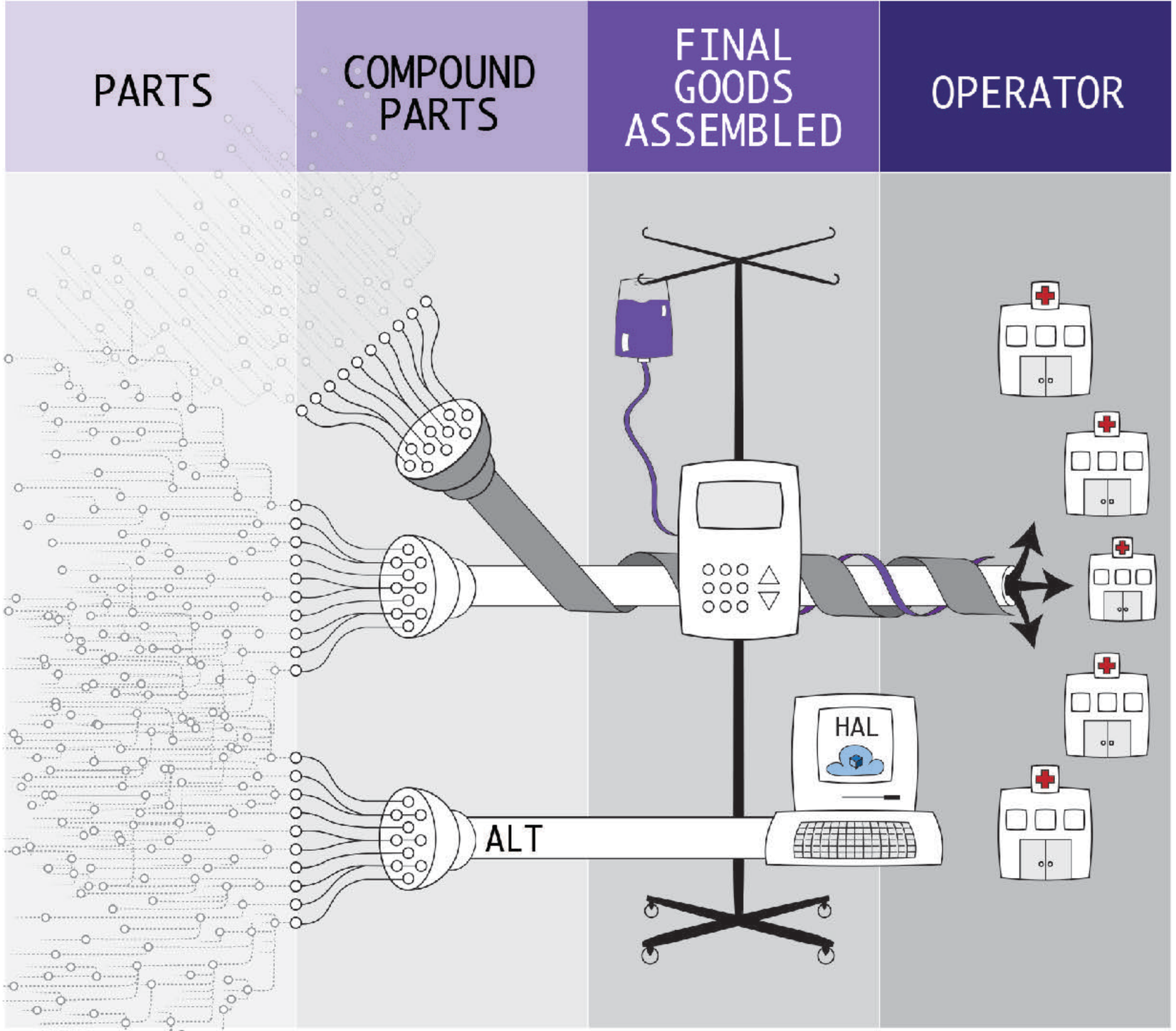

PARTS
COMPOUND PARTS
FINAL GOODS ASSEMBLED
OPERATOR

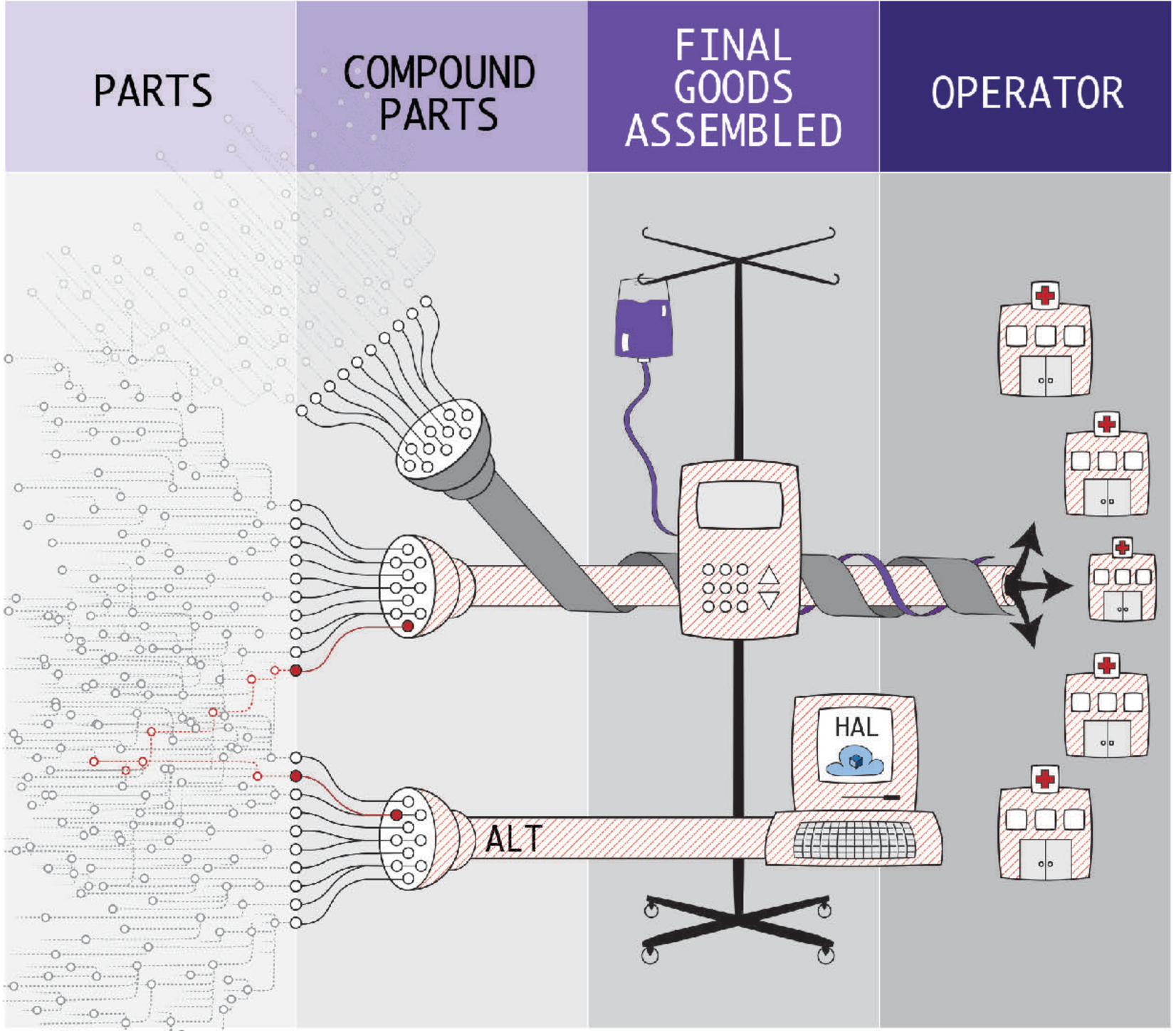

PARTS
COMPOUND PARTS
FINAL GOODS ASSEMBLED
OPERATOR


PARTS
COMPOUND PARTS
FINAL GOODS ASSEMBLED
OPERATOR
HAL


PARTS
COMPOUND PARTS
FINAL GOODS ASSEMBLED
OPERATOR
HAL

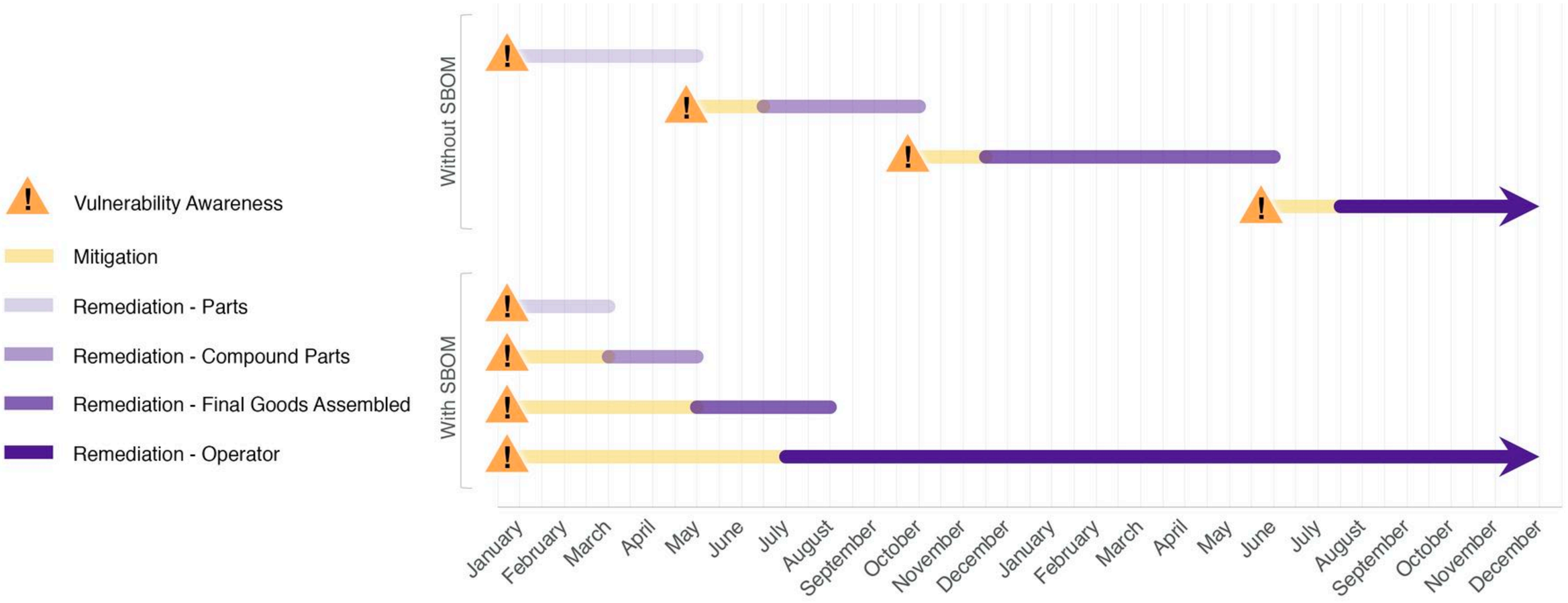

PARTS
COMPOUND PARTS
FINAL GOODS ASSEMBLED
OPERATOR
HAL


PARTS
COMPOUND PARTS
FINAL GOODS ASSEMBLED
OPERATOR
HAL
ALT

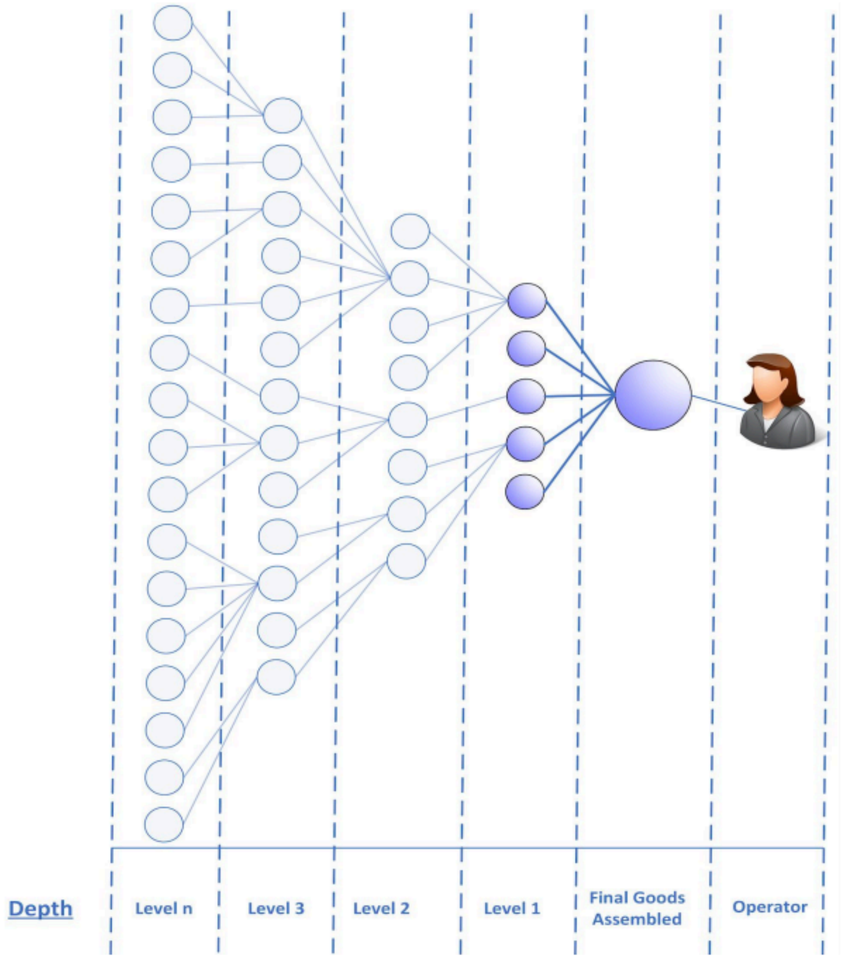

PARTS
COMPOUND PARTS
FINAL GOODS ASSEMBLED
OPERATOR
HAL
ALT

Time to Remediation Case Studies

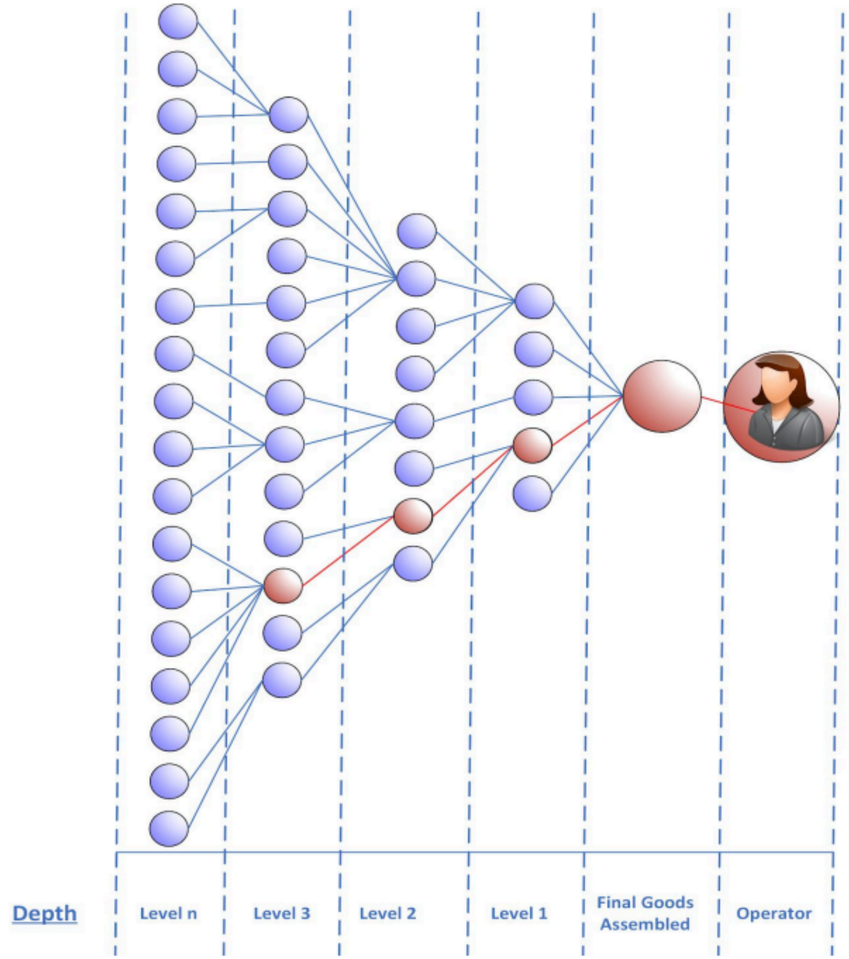
Without and With SBOM



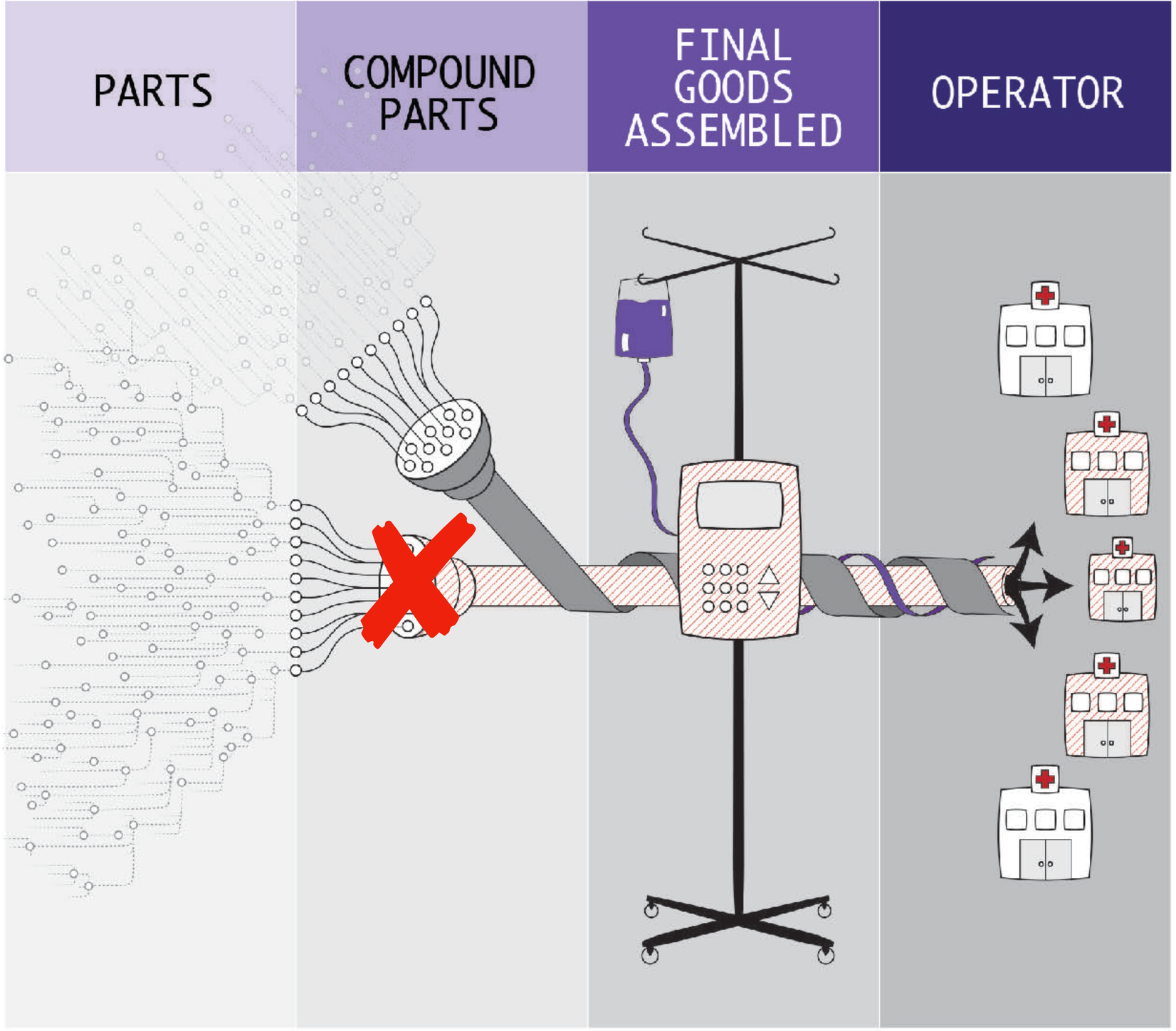
Depth vs. Effectiveness

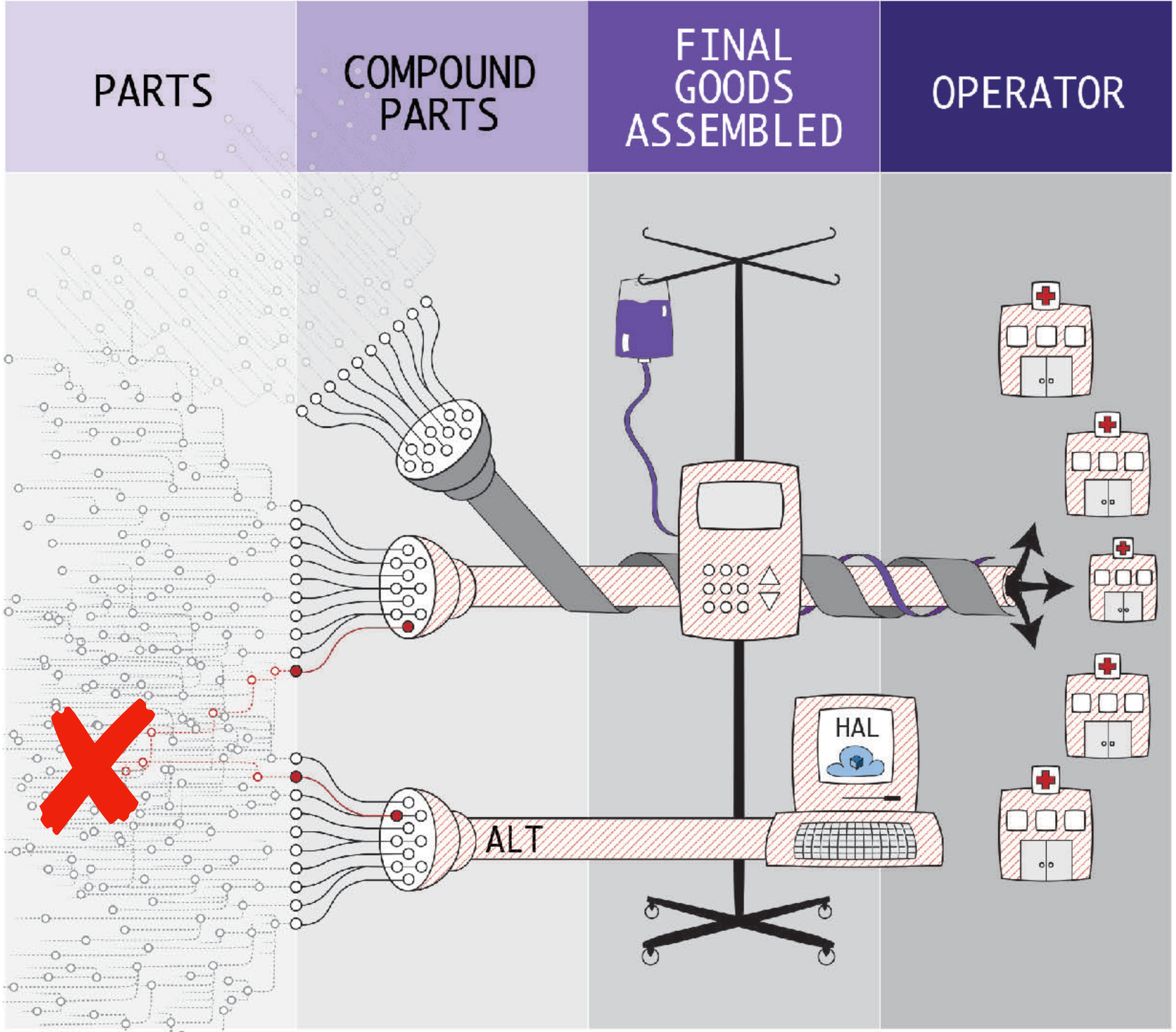


Limited visibility enables less awareness of risk



More complete visibility enables more complete awareness of risk


PARTS
COMPOUND PARTS
FINAL GOODS ASSEMBLED
OPERATOR


PARTS
COMPOUND PARTS
FINAL GOODS ASSEMBLED
OPERATOR
HAL
ALT

Parts

Compound Parts

Final Goods Assembled

Operators & Consumers

Entire Supply Chain

Parts

Compound Parts

Final Goods Assembled

Operators & Consumers

Supply Chain

Cross Sector

Thank you!

References

- NTIA Software Bill of Materials Website
  https://www.ntia.gov/sbom
- Roles and Benefits for SBOM Across the Supply Chain
  https://www.ntia.gov/files/ntia/publications/ntia_sbom_use_cases_roles_benefits-nov2019.pdf

CASSIE CROSSLEY

PRODUCT & SYSTEMS SECURITY DIRECTOR AT SCHNEIDER ELECTRIC

PRACTITIONER PERSPECTIVE

(SEE VIDEO)

JENNINGS ASKE

CISO, NEW YORK-PRESBYTERIAN HOSPITAL

PRACTITIONER PERSPECTIVE

(SEE VIDEO)

Framing Software Component Transparency: Establishing a Common Software Bill of Materials (SBOM)

https://ntia.gov/files/ntia/publications/ntia_sbom_framing_2nd_edition_20211021.pdf

https://tinyurl.com/5n9b45sv

Art Manion <amanion@cert.org>

What is an SBOM?

An SBOM is a formal, machine-readable inventory of software components and dependencies, information about those components, and their hierarchical relationships. These inventories should be comprehensive – or should explicitly state where they could not be. SBOMs may include open source or proprietary software and can be widely available or access-restricted.

Global model: SBOM elements, baseline attributes, processes, terminology

Attribute	SPDX	CycloneDX	SWID
Author Name	(2.8) Creator:	metadata/authors	<Entity> @role (tagCreator), @name
Timestamp	(2.9) Created:	metadata/timestamp	<Meta>
Supplier Name	(3.5) PackageSupplier:	metadata/supplier components/publisher	<Entity> @role (softwareCreator/publisher), @name
Component Name	(3.1) PackageName:	metadata/component/name components/name	<softwareIdentity> @name
Version String	(3.3) PackageVersion:	components/version	<softwareIdentity> @version
Component Hash	(3.10) PackageChecksum: (3.9) PackageVerificationCode:	components/hashes	<Payload>/../<File> @[hash-algorithm]:hash
Unique Identifier	(2.5)SPDX Document Namespace (3.2) SPDXID:	serialNumber components/bom-ref	<softwareIdentity> @tagID
Relationship	(7.1) Relationship: DESCRIBES CONTAINS	Dependencies compositions	<Link> @rel, @href

Table 1: Mapping baseline component information to existing formats

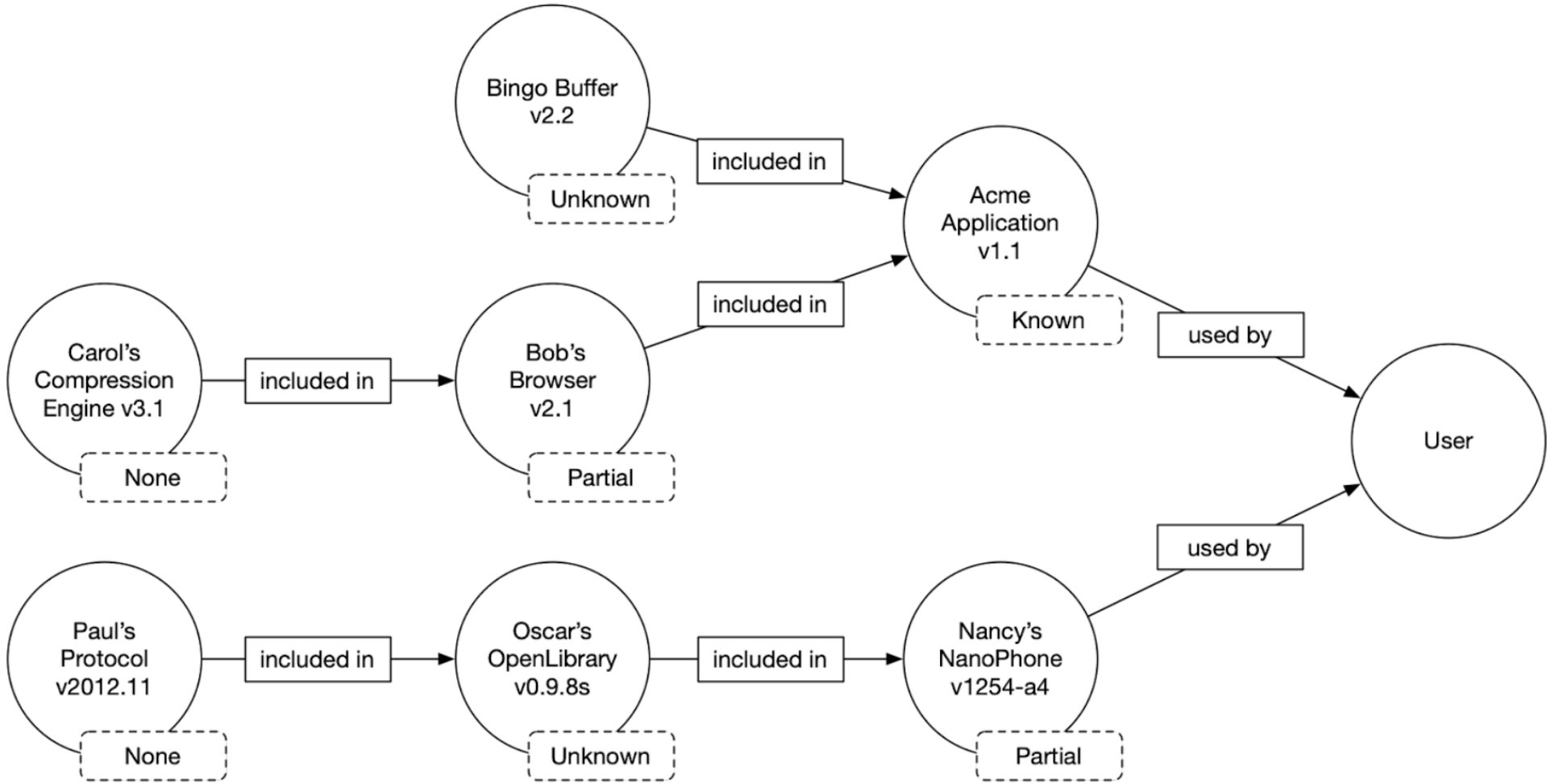


Figure 3: User graph with two supply chains

Component Name	Supplier Name	Version String	Author	Hash	UID	Relationship	Relationship Assertion
Application	Acme	1.1	Acme	0x123	234	Primary	Known
|--- Browser	Bob	2.1	Bob	0x223	334	Included in	Partial
|--- Compression Engine	Carol	3.1	Acme	0x323	434	Included in	None
|--- Buffer	Bingo	2.2	Acme	0x423	534	Included in	Unknown

Table 4: Conceptual SBOM table with upstream relationship assertions

Component Name	Supplier Name	Version String	Author	Hash	UID	Relationship	Relationship Assertion
NanoPhone	Nancy	v1254-a4	Nancy	0x523	237	Primary	Partial
|--- OpenLibrary	Oscar	0.9.8s	Nancy	0xA23	394	Included in	Partial
|--- Protocol	Paul	2012.11	Nancy	0xB53	934	Included in	None

Table 6: Conceptual SBOM table representation for Nancy’s NanoPhone

Summary SBOM Processes

- Define components, produce, maintain, provide SBOM
  - SBOM generation happens around build, package, deployment
- Seek, request, require SBOM from upstream suppliers
  - Sector-specific regulation, acquisition
- If upstream SBOM unavailable, make one up
- Choose existing formats and exchange mechanisms

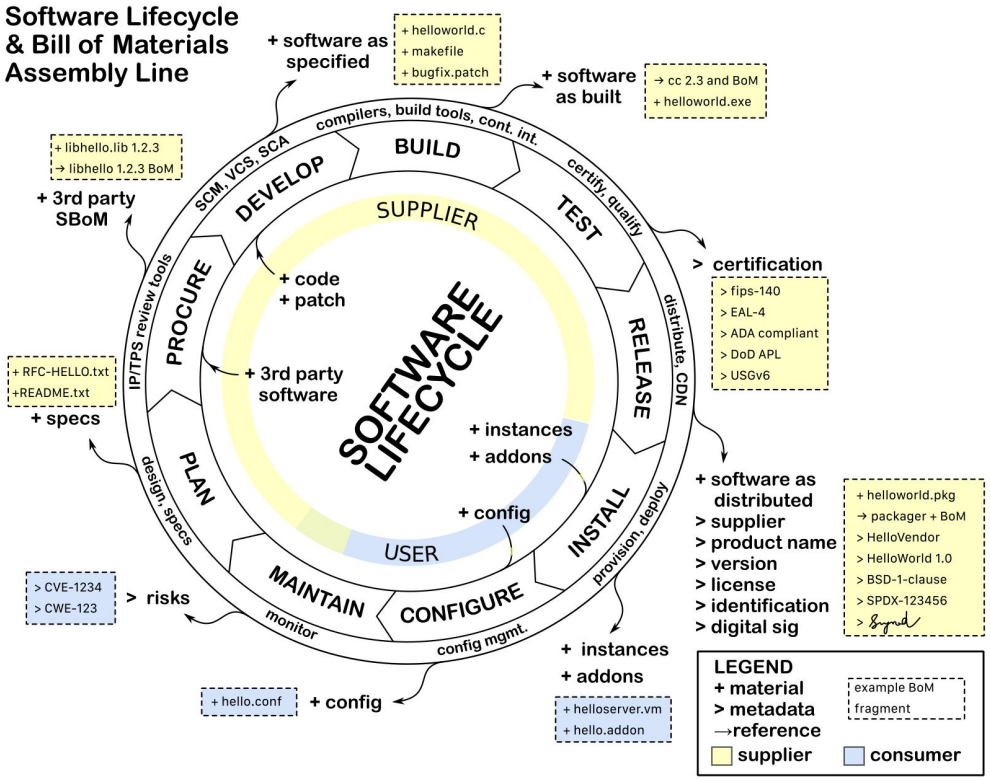
Source: Pierre Metivier (https://flic.kr/p/rdHr) CC BY-NC / Desaturated from the original

Dec 15, 2021

Tooling Taxonomy Overview

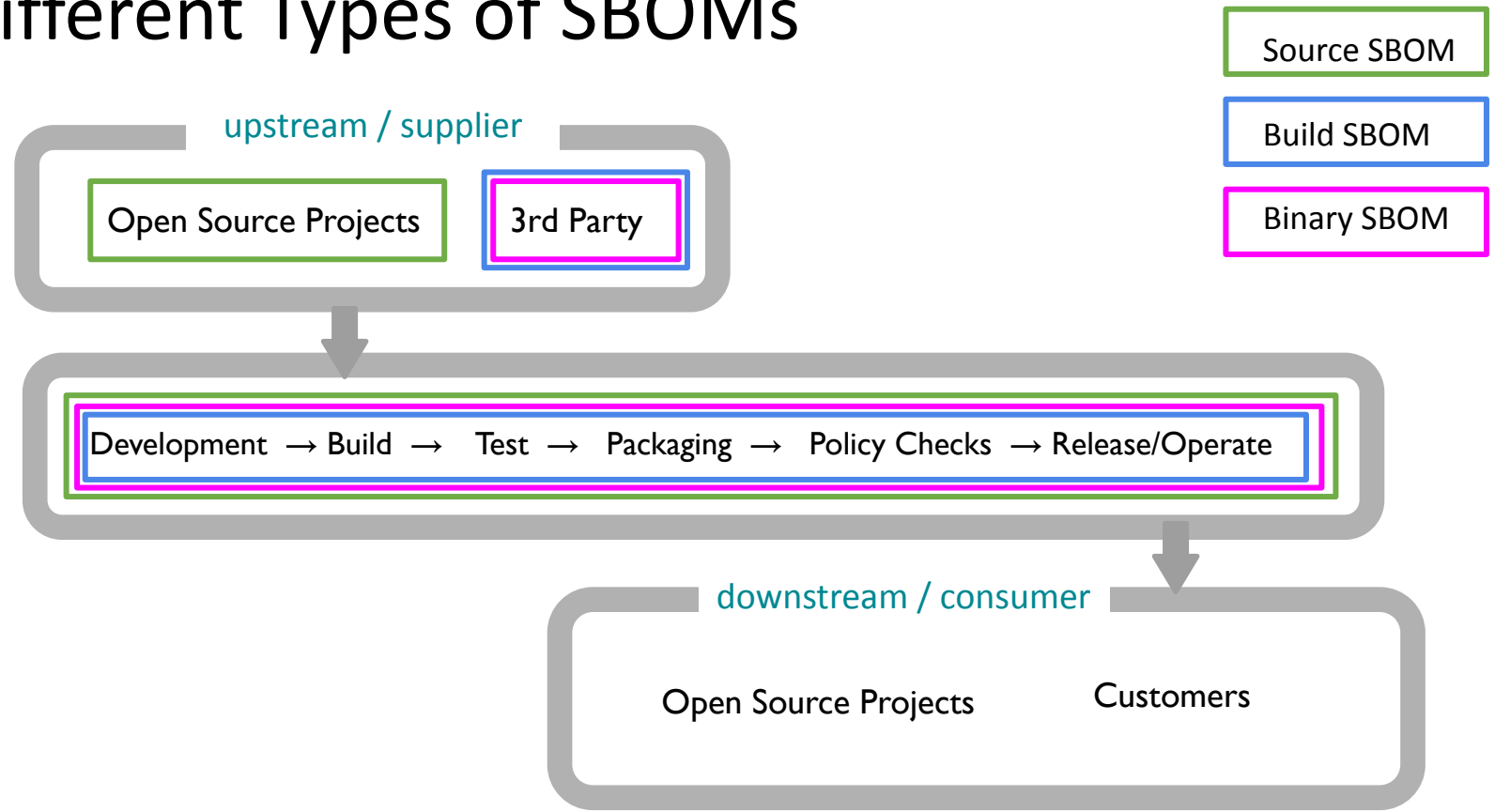
Kate Stewart
stewart@linux.com
The Linux Foundation

When should an SBOM be used?



Source: NTIA's [Survey of Existing SBOM Formats and Standards]

Different Types of SBOMs



Taxonomy for Classifying SBOM Tools

Category	Type	Description
Produce	Build	SBOM is automatically created as part of building a software artifact and contains information about the build
	Analyze	Analysis of source or binary files will generate the SBOM by inspection of the artifacts and any associated sources
	Edit	A tool to assist a person manually entering or editing SBOM data
Consume	View	Be able to understand the contents in human readable form (e.g. picture, figures, tables, text.). Use to support decision making & business processes
	Diff	Be able to compare multiple SBOMs and clearly see the differences (e.g. comparing two versions of a piece of software)
	Import	Be able to discover, retrieve, and import an SBOM into your system for further processing and analysis
Transform	Translate	Change from one file type to another file type while preserving the same information
	Merge	Multiple sources of SBOM and other data can be combined together for analysis and audit purposes
	Tool support	Support use in other tools by APIs, object models, libraries, transport, or other reference sources

More details in: https://www.ntia.gov/files/ntia/publications/ntia_sbom_tooling_taxonomy-2021mar30.pdf

Information to Collect per Tool

Tool Template

Support	Produce, Consume, Transform
Functionality	
Location	Website: Source:
Installation instructions	
How to use	
Versions Supported	

Example: FOSSology

Support	Produce (Analyze, EditI), Consume(View,Diff,Import), Transform(Translate, Merge, Tool Support)
Functionality	FOSSology is an open source license compliance software system and toolkit allowing users to run license, copyright and export control scans from a REST API. As a system, a database and web UI are provided to provide a compliance workflow. As part of the toolkit multiple license scanners, copyright and export scanners are tools available to help with compliance activities.
Location	Website: https://www.fossology.org/ Source: https://github.com/fossology
Installation instructions	https://www.fossology.org/get-started/
How to use	https://www.fossology.org/get-started/basic-workflow/
Versions Supported:	SPDX 2.1, SPDX 2.2

Collecting the Tools...

- Google docs for collecting **tools** in the three formats (open source and commercial offerings)
  - **SWID:** http://tiny.cc/SWID
  - **SPDX:** http://tiny.cc/SPDX
  - **CycloneDX:** http://tiny.cc/CycloneDX

- Desire to move to neutral GitHub location to allow a more open process and wider set of visible reviews.
  - Anyone can nominate tool to be added to a list
  - Point to evidence of producing, consuming or transforming of SBOM documents to get tool on the list (this includes participating in Plugfest)

Translating between SBOM Formats & File Types

- SwiftBOM: (SPDX(.spdx), SWID(.xml), CycloneDX(.xml,.json))
  - Demo at: https://democert.org/sbom/
  - Source code at: https://github.com/CERTCC/SBOM/tree/master/sbom-demo
- SPDX online tools: (SPDX (.spdx, .json, .yaml, .rdf, .xml, .xls))
  - Demo at: https://tools.spdx.org/app/
  - Source code at: https://github.com/spdx/spdx-online-tools
- CycloneDX CLI: (CycloneDX (.xml, .json), SPDX(.spdx))
  - Source code at: https://github.com/CycloneDX/cyclonedx-cli

Where to find more info on tools:

- **CycloneDX:** https://cyclonedx.org/tool-center/
- **SPDX:** https://spdx.dev/resources/tools/

Possible Next Steps:

- Plugfests in 2022 (Consumers, ???)
- **Case studies** of organization adoption of tools & **reference tooling workflows**

SPDX: Overview

William Bartholomew (@iamwillbar)
Principal Security Strategist, Microsoft
Core Profile Lead, SPDX

Mission

The mission of SPDX is to develop and promote **open standards** for communicating software bill of material information (SBOM), including **provenance**, **license**, **security**, and **other related information**.

SPDX

Background

- Born out of a need to exchange OSS component and license information
- Recently celebrated its 10th birthday
- SPDX 2.2 became ISO standard this year (ISO/IEC 5962:2021)
- Open weekly working group and monthly general meetings

SPDX Supporters

anchore, Bitergia, Canvass Labs, Chain Guard, Cisco, FOSSA, Foundries.io, Google, here, Hitachi Inspire the Next, intel, Microsoft, MITRE, nexB, Scania, Siemens, snyk, Source Auditor, Synopsys, Texas Instruments, Tidelift, TNG Technology Consulting, vmware, WNDRVR, wipro, Xilinx

SPDX

Features

Multiple formats (JSON, YAML, RDF/XML, Tag/Value)

Flexible for different use cases

Rich open source licensing expressions

Describe complex relationships

Cross-document references

Example

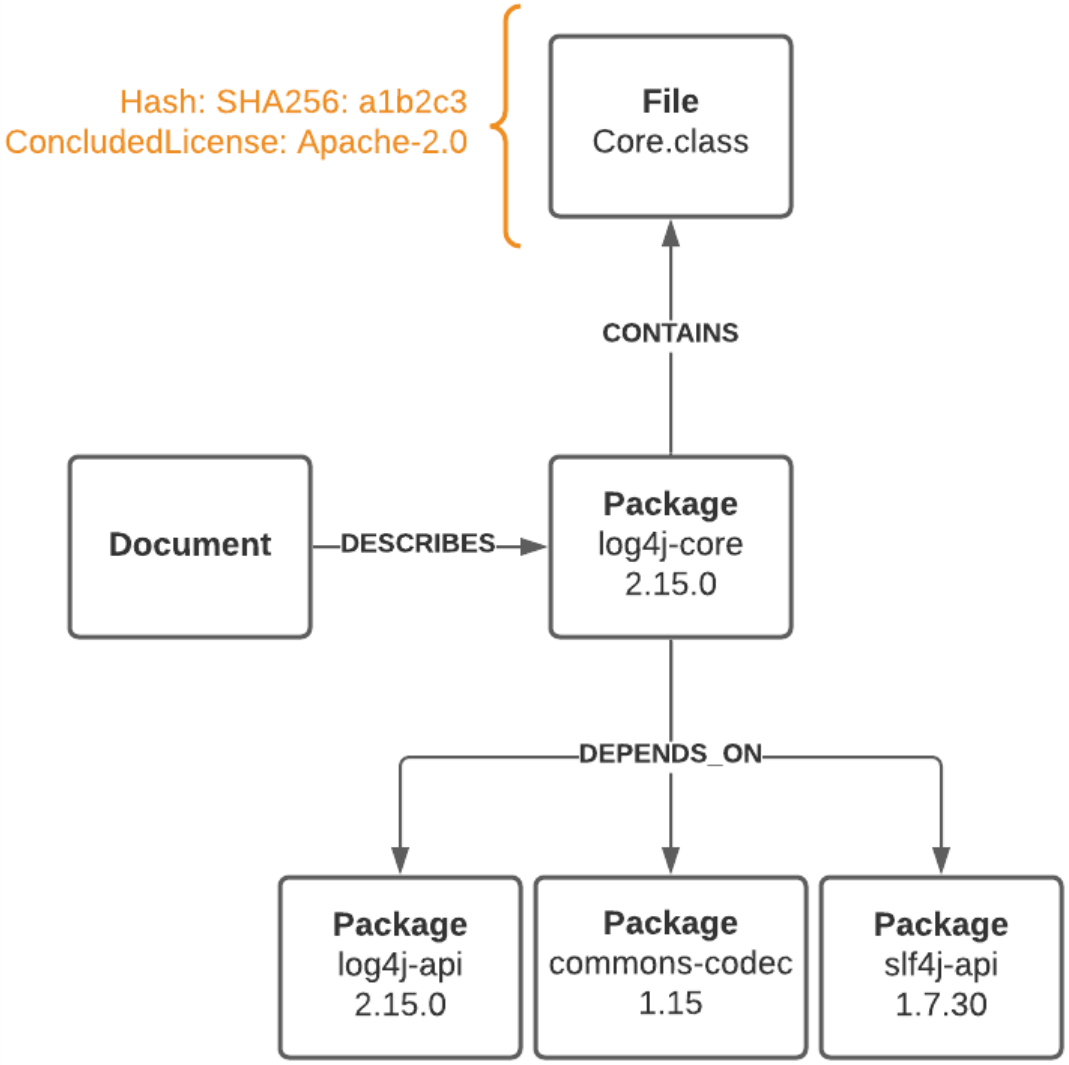
```
SPDXVersion: SPDX-2.2
DataLicense: CC0-1.0
SPDXID: SPDXRef-DOCUMENT
DocumentName: hello
DocumentNamespace: https://swinslow.net/spdx-
examples/example1/hello-v3
Creator: Person: Steve Winslow (steve@swinslow.net)
Creator: Tool: github.com/spdx/tools-golang/builder
Creator: Tool: github.com/spdx/tools-golang/idsearcher
Created: 2021-08-26T01:46:00Z

PackageName: hello
SPDXID: SPDXRef-Package-hello
PackageDownloadLocation: git+https://github.com/swinslow/spdx-
examples.git#example1/content
FilesAnalyzed: true
PackageVerificationCode: 9d20237bb72087e87069f96afb41c6ca2fa2a342
PackageLicenseConcluded: GPL-3.0-or-later
PackageLicenseInfoFromFiles: GPL-3.0-or-later
PackageLicenseDeclared: GPL-3.0-or-later
PackageCopyrightText: NOASSERTION
```

```
FileName: /build/hello
SPDXID: SPDXRef-hello-binary
FileType: BINARY
FileChecksum: SHA1: 20291a81ef065ff891b537b64d4fdccaf6f5ac02
FileChecksum: SHA256:
83a33ff09648bb5fc5272baca88cf2b59fd81ac4cc6817b86998136af368708e
FileChecksum: MD5: 08a12c966d776864cc1eb41fd03c3c3d
LicenseConcluded: GPL-3.0-or-later
LicenseInfoInFile: NOASSERTION
FileCopyrightText: NOASSERTION

FileName: /src/hello.c
SPDXID: SPDXRef-hello-src
FileType: SOURCE
FileChecksum: SHA1: 20862a6d08391d07d09344029533ec644fac6b21
FileChecksum: SHA256:
b4e5ca56d1f9110ca94ed0bf4e6d9ac11c2186eb7cd95159c6fdb50e8db5a823
FileChecksum: MD5: 935054fe899ca782e11003bbae5e166c
LicenseConcluded: GPL-3.0-or-later
LicenseInfoInFile: GPL-3.0-or-later
FileCopyrightText: Copyright Contributors to the spdx-examples
project.

Relationship: SPDXRef-DOCUMENT DESCRIBES SPDXRef-Package-hello
Relationship: SPDXRef-hello-binary GENERATED_FROM SPDXRef-hello-
src
```

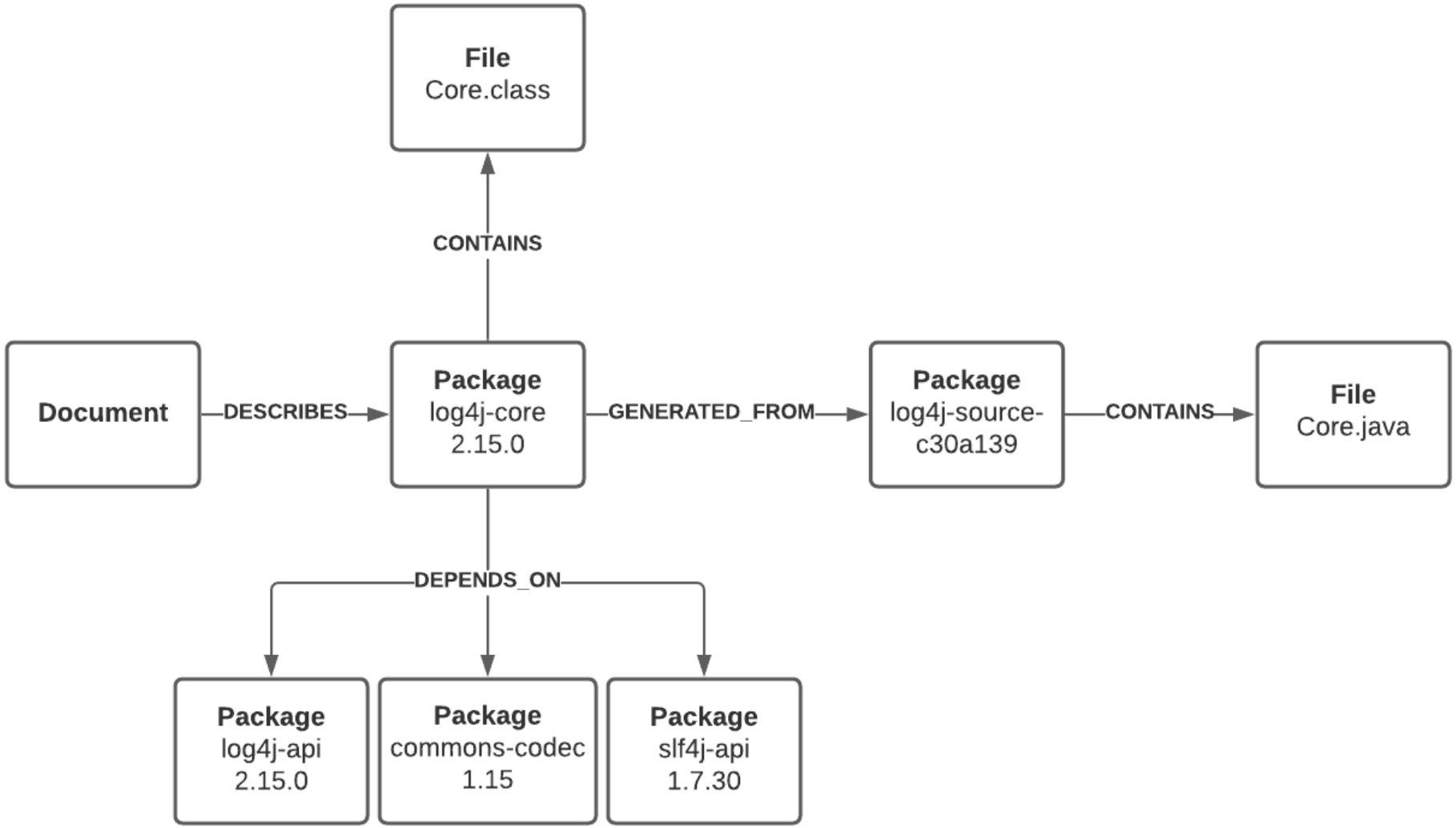
SPDX


Creator: Person: William (...)
Created: 2021-11-12T13:29:00Z
Document
DESCRIBES
Package
log4j-core
2.15.0
PackageSupplier: Organization: Apache Software Foundation
DeclaredLicense: Apache-2.0
ExternalRef: SECURITY cpe23Type cpe:2.3:a:apache:log4j:2.15.0:*:*:*:*:*:*:*
ExternalRef: PACKAGE-MANAGER purl pkg:maven/org.apache.logging.log4j/log4j-core@2.15.0

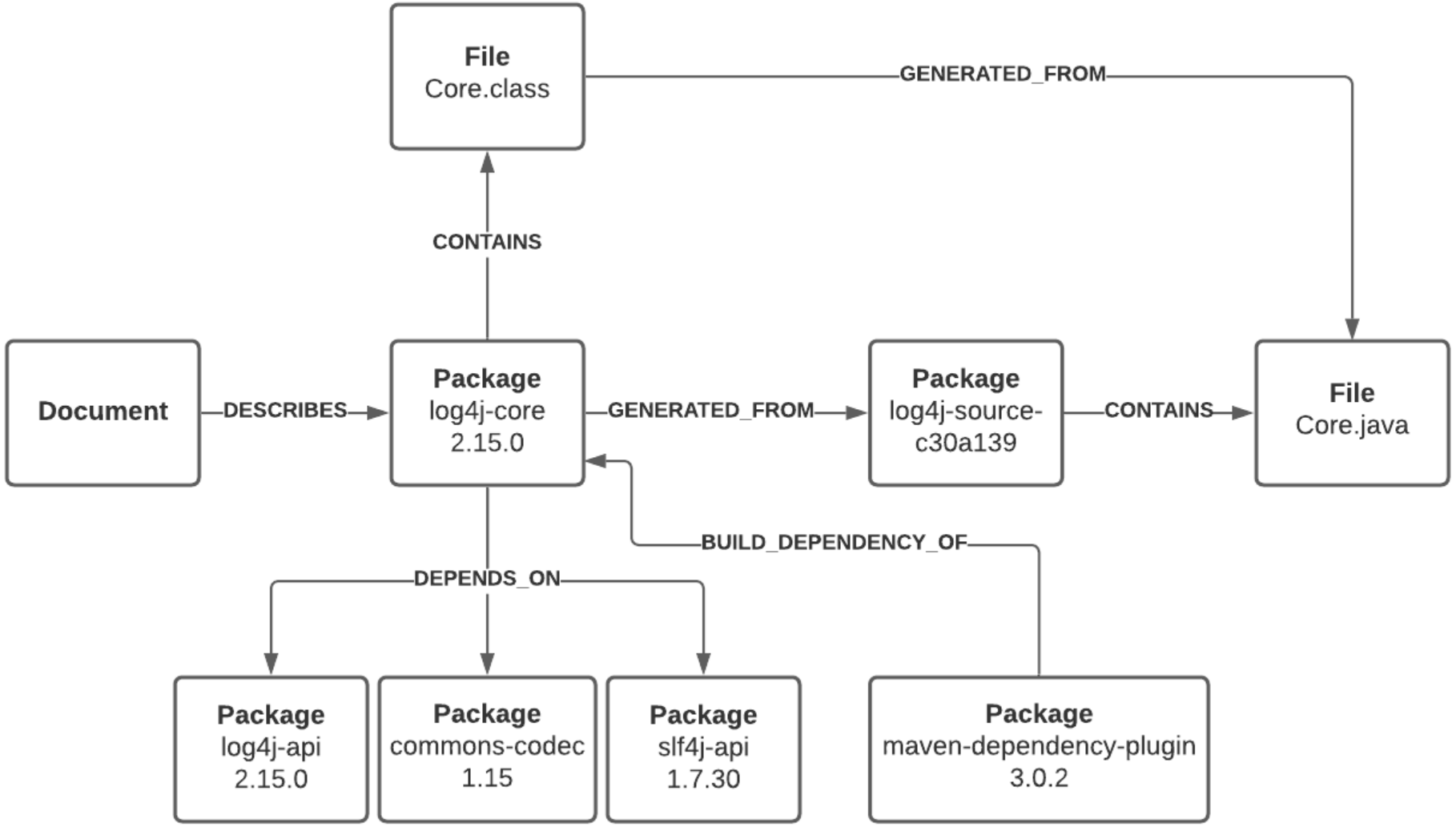
SPDX


Hash: SHA256: a1b2c3
ConcludedLicense: Apache-2.0
File
Core.class
CONTAINS
Document
DESCRIBES
Package
log4j-core
2.15.0
DEPENDS_ON
Package
log4j-api
2.15.0
Package
commons-codec
1.15
Package
slf4j-api
1.7.30

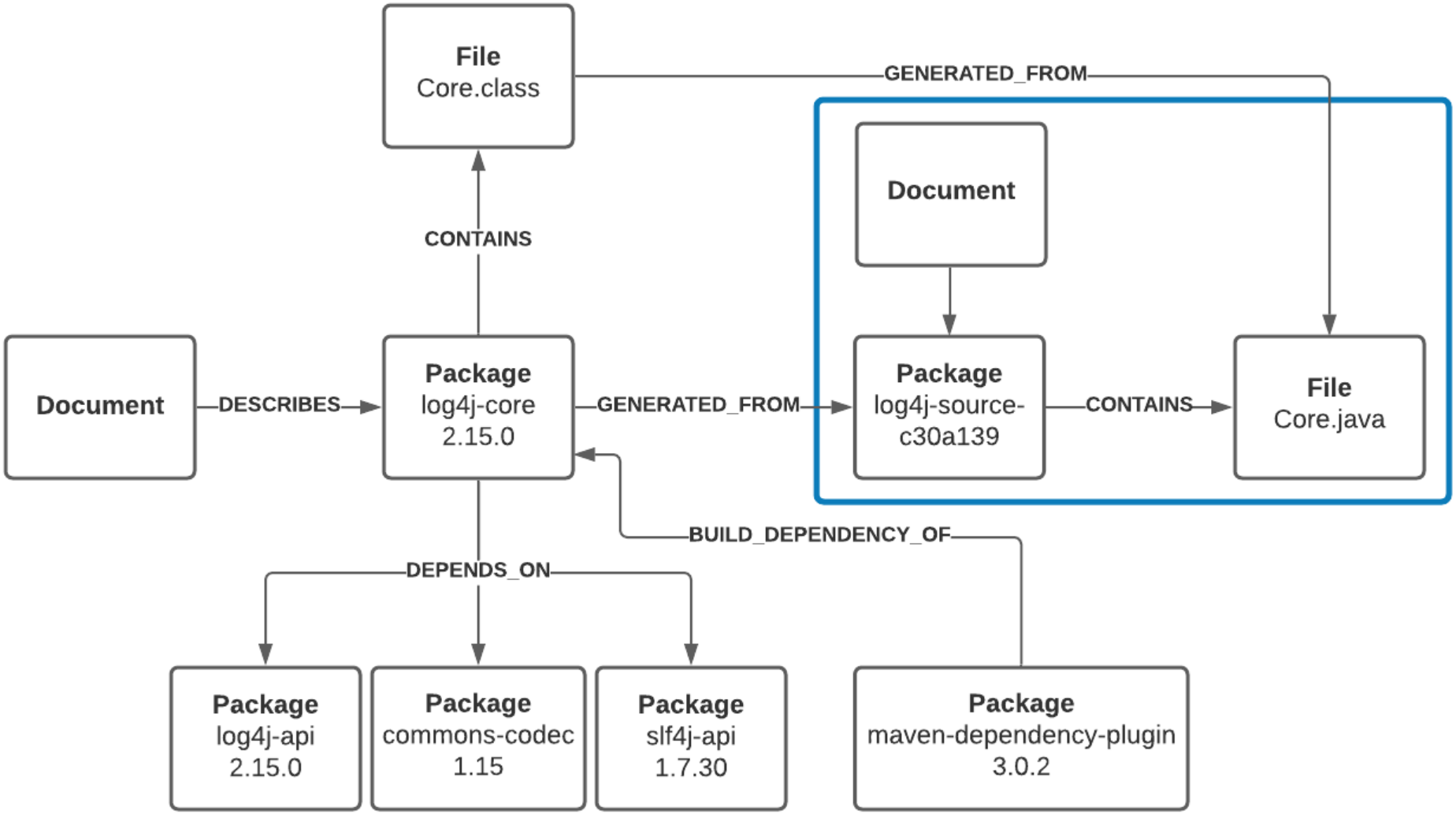
SPDX


File
Core.class
CONTAINS
Document
DESCRIBES
Package
log4j-core
2.15.0
GENERATED_FROM
Package
log4j-source-
c30a139
CONTAINS
File
Core.java
DEPENDS_ON
Package
log4j-api
2.15.0
Package
commons-codec
1.15
Package
slf4j-api
1.7.30

SPDX


File
Core.class
GENERATED_FROM
CONTAINS
Document
DESCRIBES
Package
log4j-core
2.15.0
GENERATED_FROM
Package
log4j-source-
c30a139
CONTAINS
File
Core.java
BUILD_DEPENDENCY_OF
DEPENDS_ON
Package
log4j-api
2.15.0
Package
commons-codec
1.15
Package
slf4j-api
1.7.30
Package
maven-dependency-plugin
3.0.2

SPDX


File
Core.class
GENERATED_FROM
Document
CONTAINS
Document
DESCRIBES
Package
log4j-core
2.15.0
GENERATED_FROM
Package
log4j-source-
c30a139
CONTAINS
File
Core.java
BUILD_DEPENDENCY_OF
DEPENDS_ON
Package
log4j-api
2.15.0
Package
commons-codec
1.15
Package
slf4j-api
1.7.30
Package
maven-dependency-plugin
3.0.2

SPDX

SPDX 3.x

Split specification into profiles

- Core (Artifact, Relationship, Collection, IntegrityMethod, Identity, …)
- Software (Package, File, Snippet, SBOM, …)
- Licensing

Minimize required fields

New profiles

- Defects (Vulnerability, …)
- Usage

Support scenarios beyond software

- Hardware
- Services
- Data

SPDX

Next steps

- Learn https://spdx.dev/
- Participate https://spdx.dev/participate/
- Use https://spdx.dev/resources/tools/

SPDX

CycloneDX

Software Bill of Materials Standard

Patrick Dwyer

@coderpatros

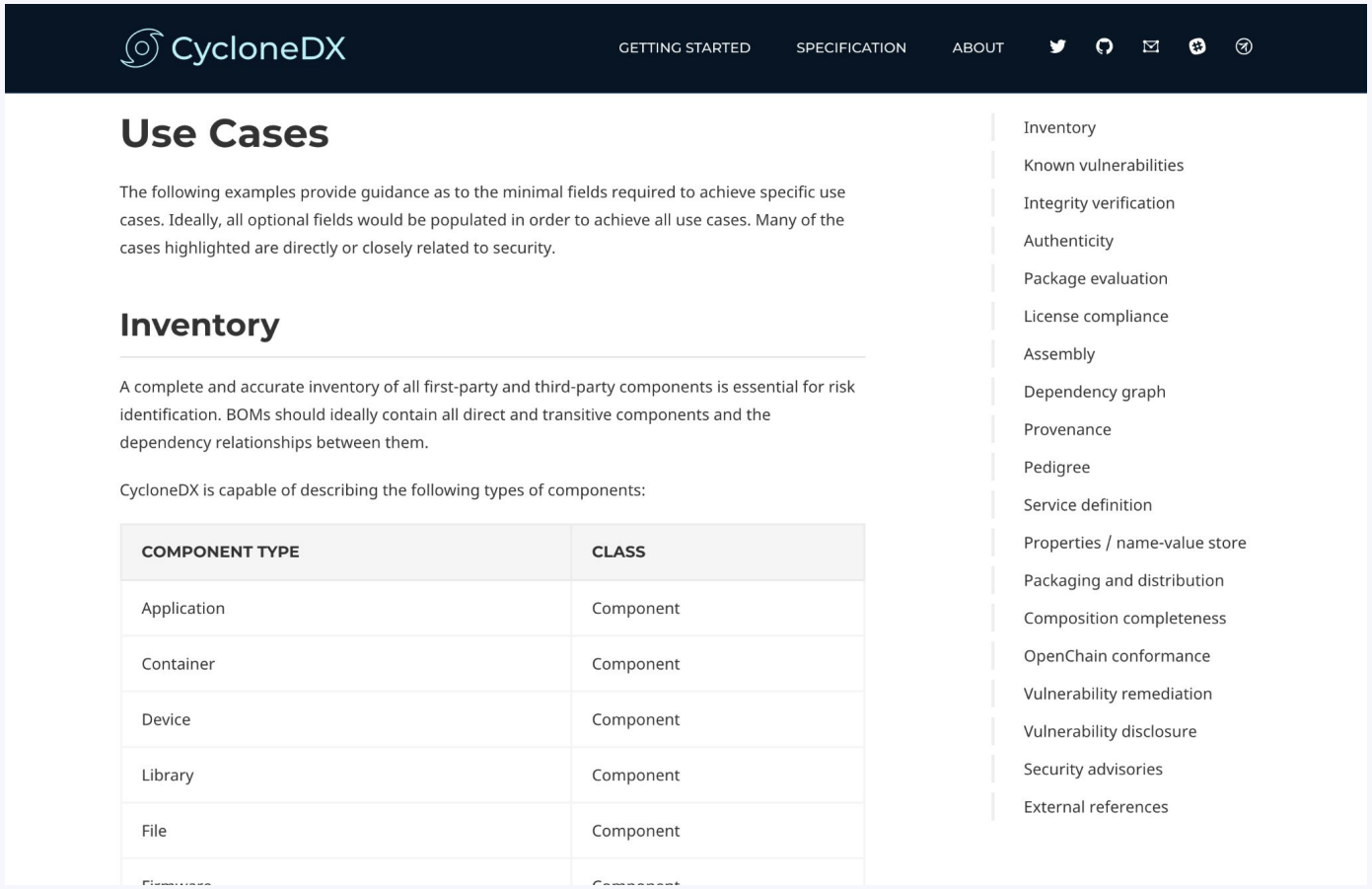
patrick.dwyer@owasp.org

- Co-Leader of OWASP CycloneDX
- Contributor to multiple SBOM related projects and tools
- OSS Maintainer
- Software Development Lead (Government)

Introducing CycloneDX

- Flagship OWASP standards project
- Lightweight, simplicity over complexity - easy to implement and adopt
- Optimized for highly automated processes
- Purpose built as a BOM format for cybersecurity use cases
- Designed in May 2017
- Initial release March 2018
- Yearly releases since
- Formal governance and standards process
- Recommended by multiple world government agencies
- Large and growing industry and vendor support
  - https://cyclonedx.org/about/supporters/
- Estimated to be in use at 100k organizations

Use Case Examples



A collection of common use cases achievable with CycloneDX along with concrete examples in XML and JSON.

https://cyclonedx.org/use-cases/

BOM Metadata

```
{
  "bomFormat": "CycloneDX",
  "specVersion": "1.3",
  "serialNumber": "urn:uuid:3e671687-395b-41f5-a30f-a58921a69b79",
  "version": 1,
  "metadata": {
    {
      "timestamp": "2020-04-13T20:20:39+00:00",
      "tools": [ … ],
      "authors": [ … ],
      "manufacture": { … },
      "supplier": { … },
      "component": { … }
    }
  }
}
```

Component Inventory

```
{
  …
  "components": [
    {
      "type": "library",
      "group": "org.apache.logging.log4j",
      "name": "log4j2-core",
      "version": "2.14.1"
    }
  ]
}
```

Supports:

- Applications
- Libraries
- Frameworks
- Containers
- Operating systems
- Firmware
- Devices
- Files
- Services

Known vulnerabilities

```
{
  …
  "components": [
    {
      "type": "library",
      "group": "org.apache.logging.log4j",
      "name": "log4j2-core",
      "version": "2.14.1",
      "cpe": "cpe:2.3:a:apache:log4j:2.14.1",
      "purl": "mvn:org.apache.logging.log4j/log4j-core@2.14.1",
      "swid": { … }
    }
  ]
}
```

Integrity

```
{
  …
  "components": [
    {
      "type": "library",
      "group": "org.apache.logging.log4j",
      "name": "log4j2-core",
      "version": "2.14.1",
      "hashes": [
        {"alg": "SHA3-512", "content": "…" }
      ]
    }
  ]
}
```

Authenticity

- XML Signature
- JSON Web Signature (JWS)
- JSON Signature Format (JSF)
- Digital signatures can be applied to a BOM or to an assembly within a BOM
- Signatures can be external to the BOM or enveloped (included within)

Component Pedigree

```
"pedigree": {
  "ancestors": [
    {
      "type": "library",
      "group": "org.apache.logging.log4j",
      "name": "log4j2-core",
      "version": "2.14.1"
    }
  ],
  "patches": [
    {
      "type": "backport",
      "diff": { …, "resolves": [{ "type": "security", "id": "CVE-2021-44228", … }] }
  ]
}
```

Provenance

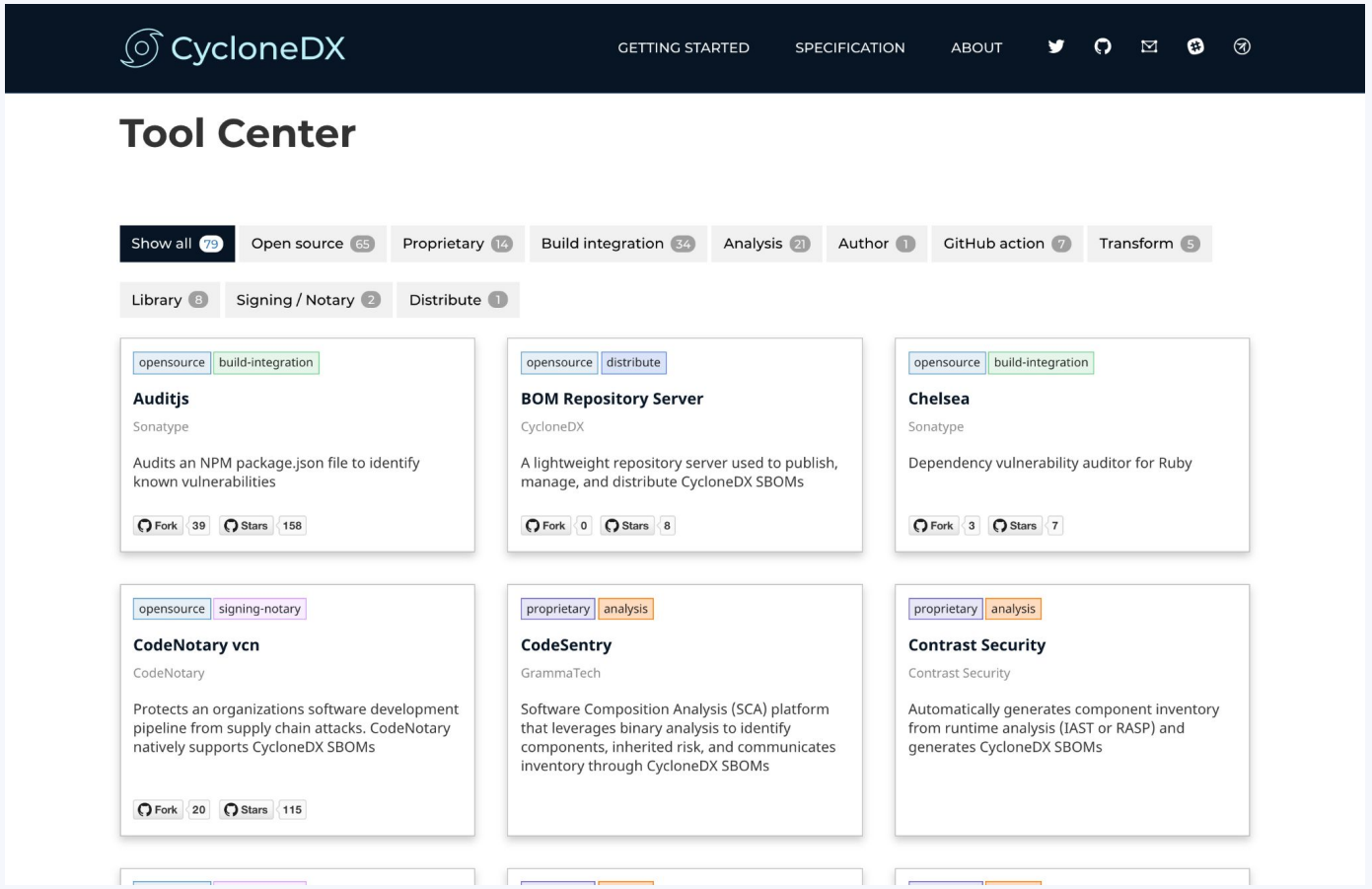
- Component downloaded location
- Supplier
- Author
- Publisher

Composition

- Assemblies
- Dependency graph
- Completeness
  - complete
  - incomplete
  - first-party/third-party
  - unknown

and many, many more...

Tool Center



Community effort to establish a marketplace of free, open source, and proprietary tools and solutions that support CycloneDX.

https://cyclonedx.org/tool-center/

In development

- Improved hardware support
- “Vulnerability-Exploitability eXchange” format, aka VEX
- IETF URN namespace registration to deeplink between BOMs
- Schema hardening
- OWASP SBOM Maturity Model
- CycloneDX v1.4 due for release January 2022

Community Participation

- Website (introduction, use cases, tool center, and specification)
  - https://cyclonedx.org/
- GitHub
  - https://github.com/CycloneDX
- Slack
  - https://cyclonedx.org/slack
  - https://cyclonedx.org/slack/invite
- Mailing List
  - https://cyclonedx.org/discussion

Thank You

OWASP

TM

Standing on Shoulders: A Review of Playbooks from NTIA's SBOM Multistakeholder Initiative

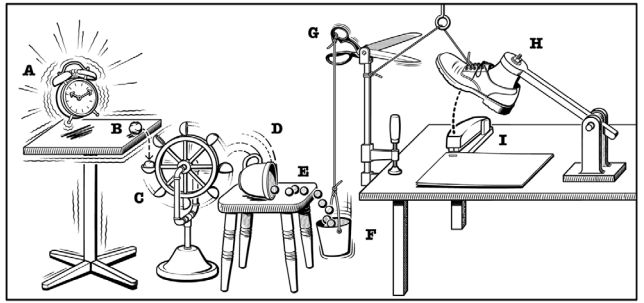
JC Herz
jc.herz@ionchannel.io

Overview: Strategic and Tactical Objectives

- SBOM file formats are established and in commercial use
- BUT: files are a starting point, not the end state
- Operationalizing SBOMs requires workflows to generate and act on the data
- Supplier playbook: Steps to create and provide
- Consumer playbook: Steps to metabolize and use
- Reality: Most suppliers are also consumers, and often vice versa
- Playbooks are technical and business process requirements - not endorsements of specific technological solutions or build/buy decisions.

Supplier Playbook

- SBOM Production: Generalized Process
  1. Identify software components included in a deliverable
  2. Acquire data about components used in a deliverable
  3. Import component data into a structured SBOM format.
  4. Validate SBOM to ensure format is valid and baseline attributes are present.
- Relevant Workflow Differentiators
  - Best-Practice vs. Non-Automated Engineering Processes
  - Build-Time vs. Post-Build SBOMs
- Deliverable (What's in the Box): Applications, Containers, Systems
  - Ex: Operating System, Runtime Dependencies, Installers
- Requires Consensus: External Services



- https://www.ntia.gov/files/ntia/publications/software_suppliers_sbom_production_and_provision_-_final.pdf

Consumer Playbook

- Acquisition of SBOM from a Supplier
  - Contractual procurement of a commercial product
  - Download of commercial closed-source product
  - Contractual procurement of professional services
  - Acquisition of open source software
  - Discovery processes as a device connects to a network
- SBOM Coverage for Software Systems
- Software Entity Resolution
- Third Party Processes and Platforms
- Ongoing Monitoring
- Ideally we should be able to assure critical software as well as we can assure a steak.

Consumer Playbook: https://www.ntia.gov/files/ntia/publications/software_consumers_sbom_acquisition_management_and_use_-_final.pdf
Healthcare POC: https://ntia.gov/files/ntia/publications/howto_guide_for_sbom_generation_v1.pdf

Federal Office
for Information Security

Vulnerability Exploitability Exchange (VEX)

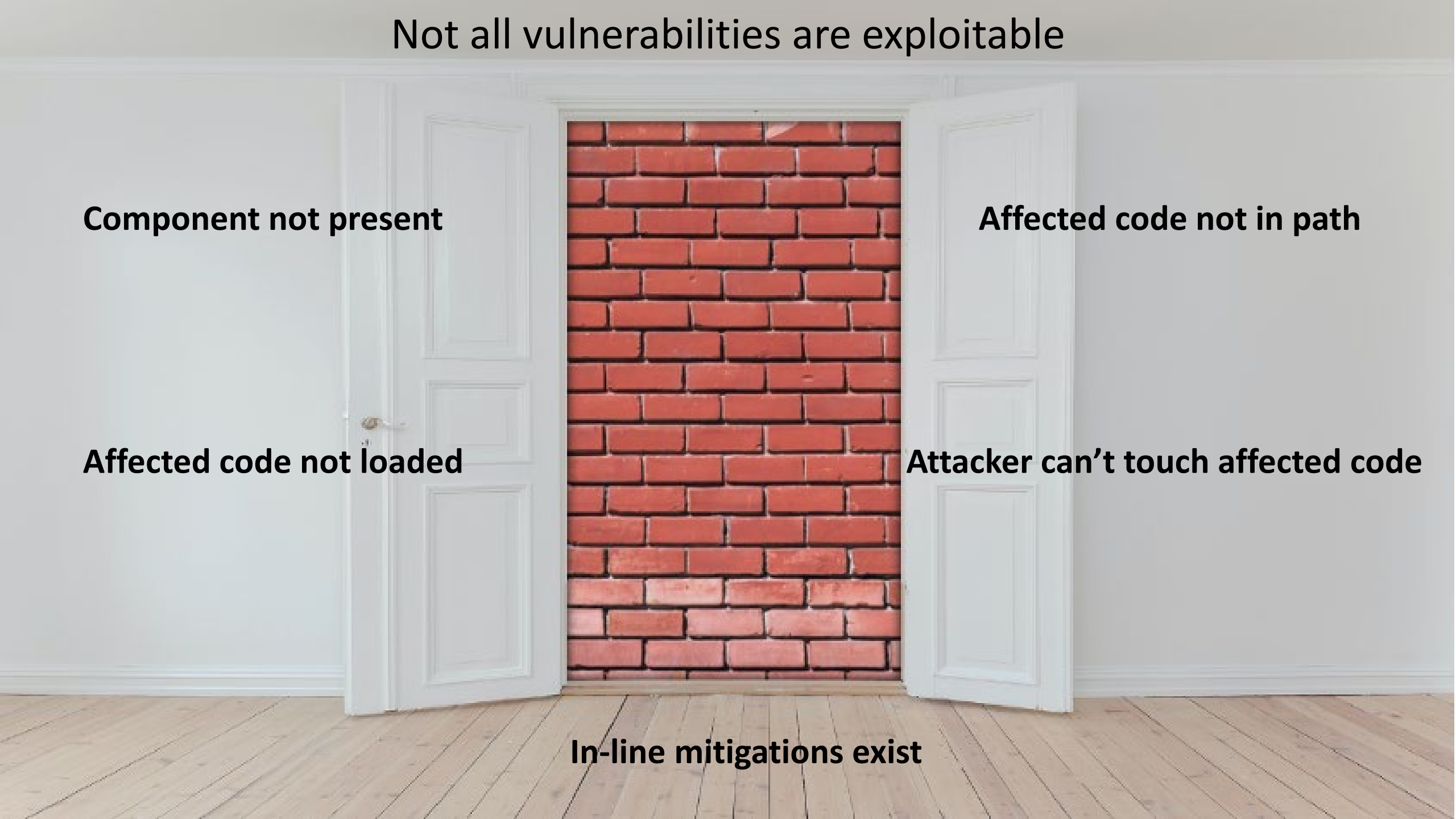
Jens Wiesner
Head of Section
German Federal Office for Information Security (BSI)

We will know about more potential vulnerabilities with SBOM

https://boyter.org/static/books/new2.jpg

Not all vulnerabilities are exploitable

Not all vulnerabilities are exploitable



Component not present

Affected code not in path

Affected code not loaded

Attacker can’t touch affected code

In-line mitigations exist

ZABBIX BLOG

Handy Tips Technical How To Integrations

News Technical

Zabbix NOT AFFECTED by the Log4j exploit

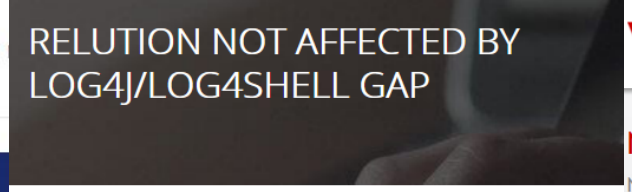
By Arturs Lontons — 2 days ago

A newly revealed vulnerability impacting Apache Log4j 2 versions 2.0 to 2.14.1 was disclosed on GitHub on 9 December 2021 and registered as CVE-2021-44228 with the highest severity rating. Log4j is an open-source, Java-based logging utility widely used by enterprise applications and cloud services. By utilizing this vulnerability, a remote attacker could take control of the affected system.

Zabbix is aware of this vulnerability, has completed verification, and can conclude that the only product where we use Java is Zabbix Java Gateway, which does not utilize the log4j library, thereby **is not impacted by this vulnerability**.

For customers, who use the log4j library with other Java applications, here are some proactive measures, which they can take to reduce the risk posed by CVE-2021-44228:

https://blog.zabbix.com/zabbix-not-affected-by-the-log4j-exploit/17873/



IMPORTANT INFORMATION

SECURITY GAP

Relution is not affected!

LOG4SHELL CVE-2021-44228

Relution

Be safe with Relution

For the critical vulnerability in log4j (CVE-2021-44228), an increase of warning level to red was declared by the German Federal Office for Information Security (BSI) on December 12, 2021. You can read more about this in the official BSI statement

Immediate testing on the Relution system ensured that Relution is not affected by the vulnerability.

- TCPDUMP for checking network connections
- Test with https://log4shell.huntress.com/

Relution is not based on the affected "log4j" framework, but uses Logback. See Log4j2 Vulnerability and Spring Boot

https://relution.io/en/news/security-log4j/

verinice. PRODUCTS .PRO .PARTNER .SUPPORT .FORUM .NE

News

News and Press Releases

verinice not affected by log4j vulnerability

12/13/2021

Last week, a critical vulnerability in the widely used logging library log4j 2 became known. The log4j versions included in the verinice.PRO server are **not** affected by the vulnerability!

The vulnerability is described in this article, among others: Log4Shell: RCE 0-day exploit found in log4j 2, a popular Java logging package and has the CVS number CVE-2021-44228 erhalten.

For more information, see the article in our verinice forum: https://forum.verinice.com/t/verinice-nicht-betroffen-von-log4j-schwachstelle/

However, on a verinice.PRO system there may be other Java applications in Tomcat that have not been installed by the verinice team. Since these applications may contain affected log4j versions, the team recommends including a parameter in the Tomcat configuration that prevents exploitation of the vulnerability in other applications. Again, see our forum post for details: https://forum.verinice.com/t/verinice-nicht-betroffen-von-log4j-schwachstelle/

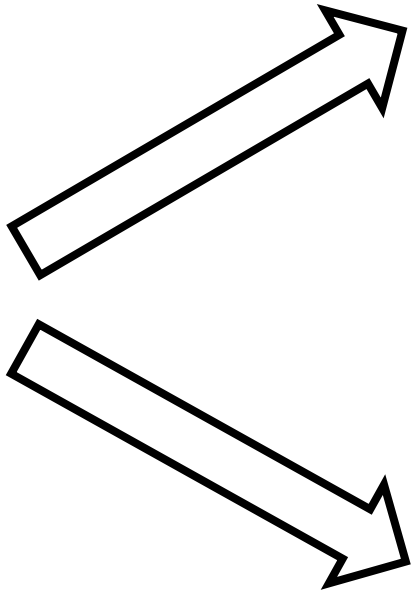
Feel free to contact our team if you have any further questions.

Back

https://verinice.com/en/news/detail/verinice-not-affected-by-log4j-vulnerability

We need a way to communicate that a product is not affected

“exploitable”



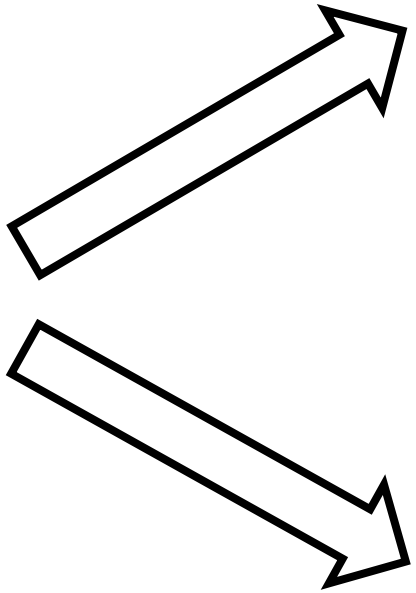
“affected”

Do I need to do anything?

“not affected”

You’re good.

“exploitable”



“affected”

Actions are recommended to remediate or address this vulnerability.
This could include: learning more about the vulnerability and context, and/or making a risk-based decision to patch or apply defense-in-depth measures

“not affected”

No remediation is required regarding this vulnerability.
This could be because the code referenced in the vulnerability is not present, not exposed, compensating controls exist, or other factors.

Required fields for a VEX

Metadata (author, id, timestamp)	
Product id	Product id
Vulnerability ID Vuln details **Product Status** Action statement / Impact statement	Vulnerability ID Vuln details **Product Status** Action statement / Impact statement

Implementing VEX in Common Security Advisory Framework (CSAF)

Common Security Advisory Framework

- Original purpose: automate security advisories to support search and evaluation
- CSAF 2.0
  - JSON format
  - Machine-readable
  - Build with automation in mind
- Standardization through CSAF TC at OASIS Open
- Successor of CSAF CVRF 1.2
- VEX implemented as a profile in CSAF
- VEX is parallel to SBOM (not necessarily in the SBOM)

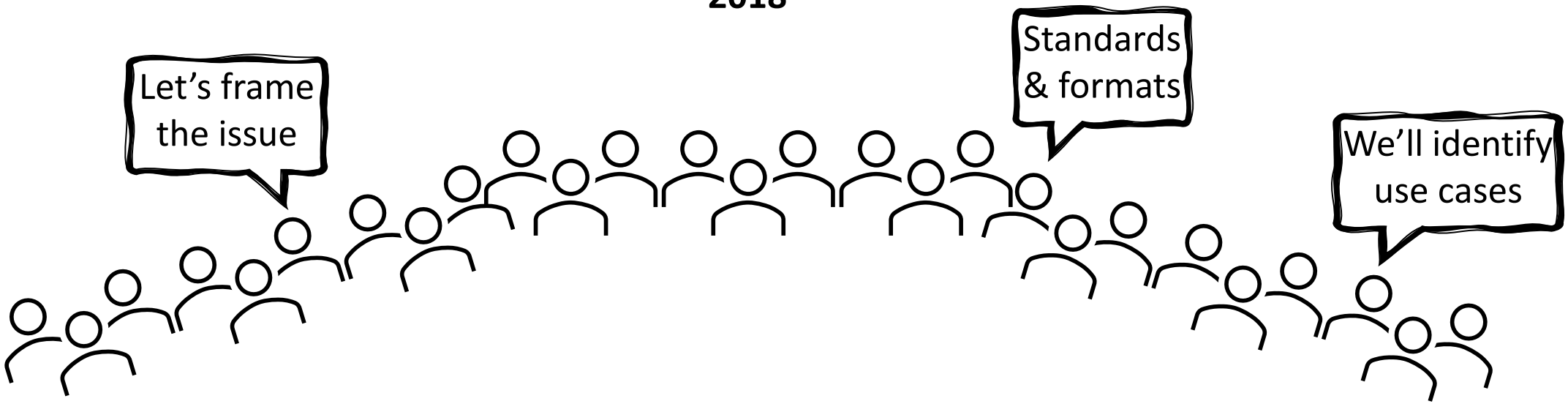
VEX Resources

VEX Overview: ntia.gov/files/ntia/publications/vex_one-page_summary.pdf
CSAF Information: csaf.io
More info on CSAF: www.bsi.bund.de/EN/Topics/Industry_CI/ICS/Tools/CSAF/csaf_node.html
CSAF Editor & Examples: secvisogram.github.io/
Join the VEX working group: sbom@cisa.dhs.gov

SBOM Proof of Concept

HEALTHCARE

SBOM Community
2018



A Healthcare SBOM
Proof of Concept is Born

A Healthcare Proof of Concept: Crawl, Walk, Run

how we set out to prove the viability of generating standardized SBOMs

Phase I Investigate 2018-2019	Custom tools/manual processes to generate SBOMs in standard formats
Phase II Iterate 2020-2021	Expand scope and modify processes based upon findings
Phase III Integrate 2022	Automate with new tools and processes in existing tool chains and systems

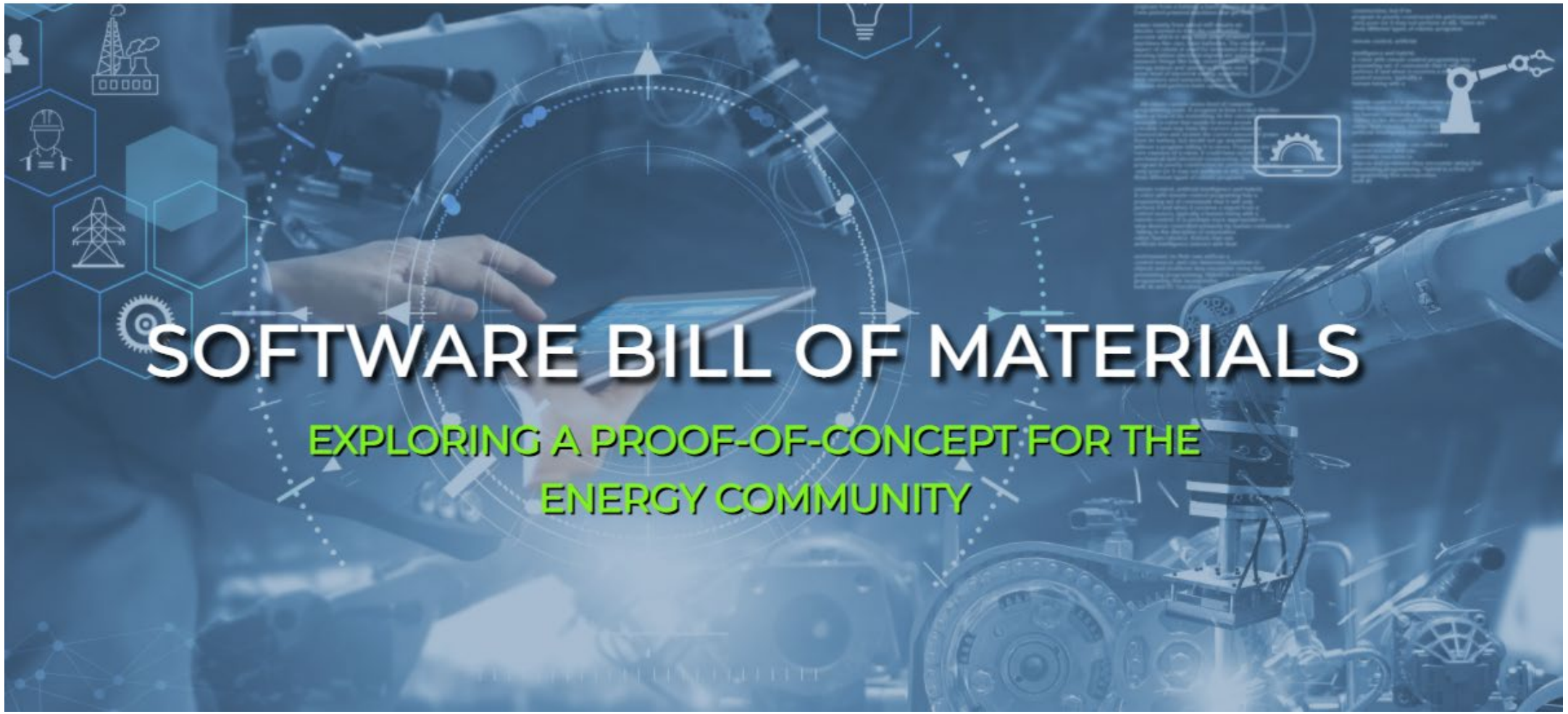
Healthcare Proof of Concept

Goals and accomplishments by phase

Phase I **Investigate** 2018-2019	Exercised primary use cases in creating & ingesting SBOMs for risk mgmt. ✓ Proved the actionable value of component transparency to the consumer ✓
Phase II **Iterate** 2020-2021	More participants, more use cases, more devices, more data, more tools ✓ Proved the viability of standard formats, data, tools; explored context info ✓
Phase III **Integrate** 2022	Drive adoption, expanded participation; real-world scenarios & data Automate SBOM sharing; prove value of context (VEX, support lifetimes)

During Covid!

What has been happening to help?



Starting again in February 2022

https://inl.gov/sbom-poc/

Cooking Classes

- Cooking Class on Making an SBOM, Sept. 22, 2021 - https://youtu.be/Tk4v1IrSNSA
- Cooking Class on Open Source, Oct. 6, 2021 - https://youtu.be/5D0P84ayGpg
- Cooking Class on VEX, Oct 20, 2021 - https://youtu.be/KjMHxeHYgIQ
- Cooking Class on Preparing to Use SBOM's, Nov. 3, 2021 - https://youtu.be/Tqkdb3XvR08
- Cooking Class on Exploring Information in an SBOM, Nov 17, 2021 - https://youtu.be/Qkx7PezvwGM

https://inl.gov/sbom-poc/

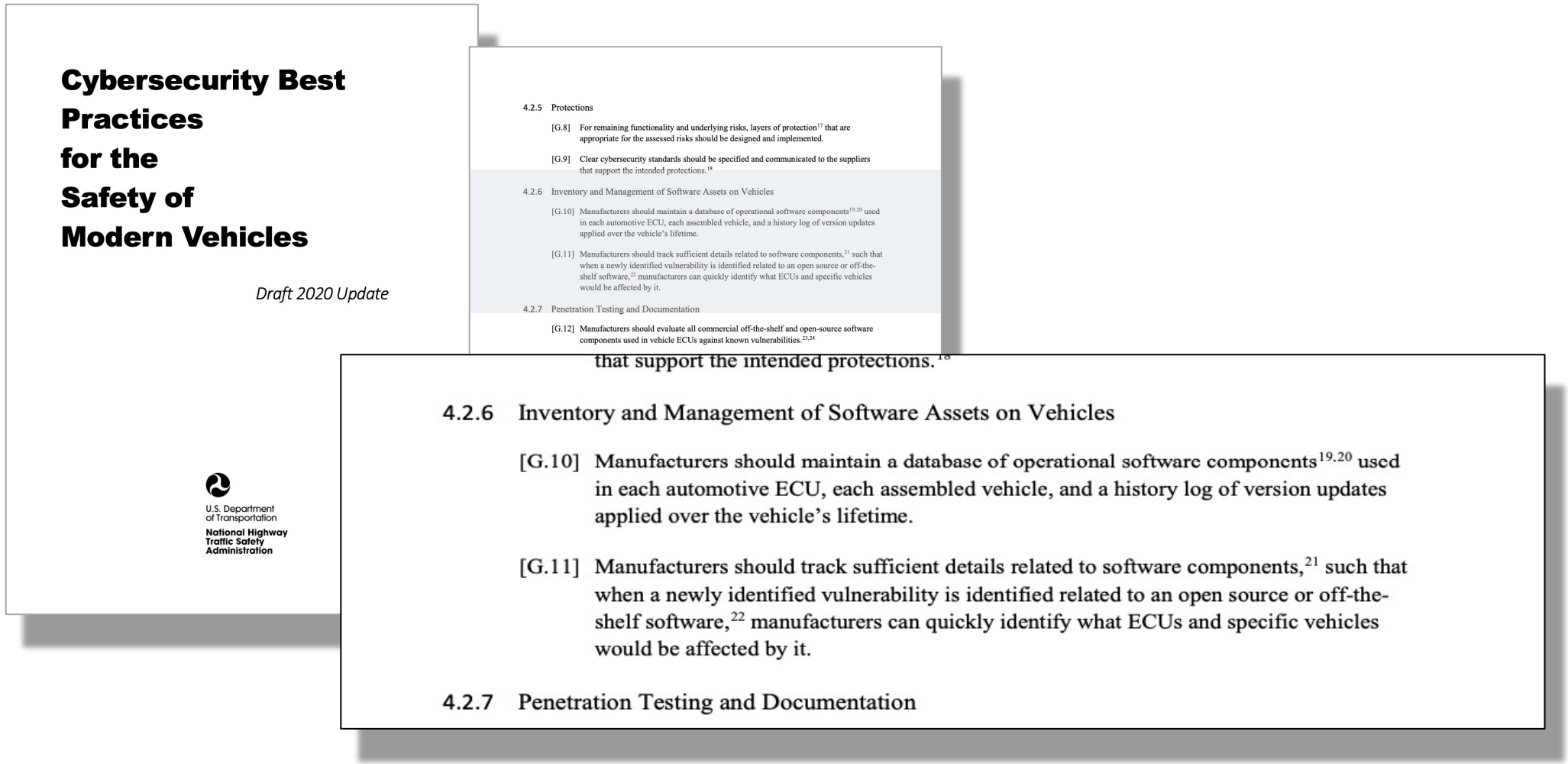
SBOM Work in AutoISAC and the Automotive Industry

Charlie Hart

Senior Analyst, Hitachi America R&D

December 15, 2021

Feb. 2021 - NHTSA – “Cybersecurity Best Practices for the Safety of Modern Vehicles”



Cybersecurity Best Practices for the Safety of Modern Vehicles

Draft 2020 Update

U.S. Department of Transportation
National Highway Traffic Safety Administration

4.2.5 Protections

[G.8] For remaining functionality and underlying risks, layers of protection[17] that are appropriate for the assessed risks should be designed and implemented.

[G.9] Clear cybersecurity standards should be specified and communicated to the suppliers that support the intended protections.[18]

4.2.6 Inventory and Management of Software Assets on Vehicles

[G.10] Manufacturers should maintain a database of operational software components[19,20] used in each automotive ECU, each assembled vehicle, and a history log of version updates applied over the vehicle’s lifetime.

[G.11] Manufacturers should track sufficient details related to software components,[21] such that when a newly identified vulnerability is identified related to an open source or off-the-shelf software,[22] manufacturers can quickly identify what ECUs and specific vehicles would be affected by it.

4.2.7 Penetration Testing and Documentation

[G.12] Manufacturers should evaluate all commercial off-the-shelf and open-source software components used in vehicle ECUs against known vulnerabilities.[23,24]

that support the intended protections.[18]

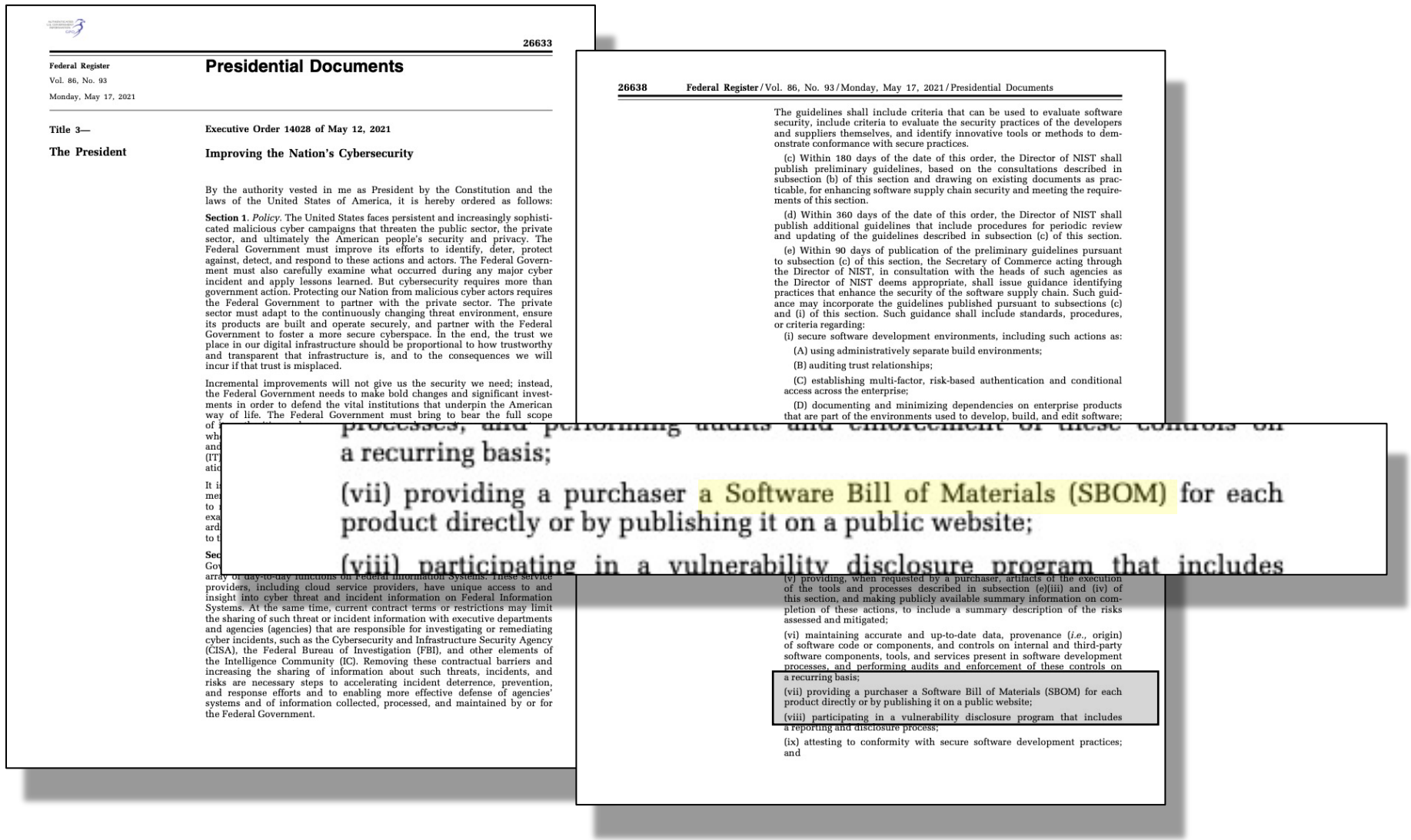
4.2.6 Inventory and Management of Software Assets on Vehicles

[G.10] Manufacturers should maintain a database of operational software components[19,20] used in each automotive ECU, each assembled vehicle, and a history log of version updates applied over the vehicle’s lifetime.

[G.11] Manufacturers should track sufficient details related to software components,[21] such that when a newly identified vulnerability is identified related to an open source or off-the-shelf software,[22] manufacturers can quickly identify what ECUs and specific vehicles would be affected by it.

4.2.7 Penetration Testing and Documentation

May 2021 - Executive Order 14028 - “Improving the Nation’s Cybersecurity”



26633

Federal Register
Vol. 86, No. 93
Monday, May 17, 2021

Presidential Documents

Title 3—

The President

Executive Order 14028 of May 12, 2021

Improving the Nation’s Cybersecurity

By the authority vested in me as President by the Constitution and the laws of the United States of America, it is hereby ordered as follows:

Section 1. *Policy*. The United States faces persistent and increasingly sophisticated malicious cyber campaigns that threaten the public sector, the private sector, and ultimately the American people's security and privacy. The Federal Government must improve its efforts to identify, deter, protect against, detect, and respond to these actions and actors. The Federal Government must also carefully examine what occurred during any major cyber incident and apply lessons learned. But cybersecurity requires more than government action. Protecting our Nation from malicious cyber actors requires the Federal Government to partner with the private sector. The private sector must adapt to the continuously changing threat environment, ensure its products are built and operate securely, and partner with the Federal Government to foster a more secure cyberspace. In the end, the trust we place in our digital infrastructure should be proportional to how trustworthy and transparent that infrastructure is, and to the consequences we will incur if that trust is misplaced.

Incremental improvements will not give us the security we need; instead, the Federal Government needs to make bold changes and significant investments in order to defend the vital institutions that underpin the American way of life. The Federal Government must bring to bear the full scope of

providers, including cloud service providers, have unique access to and insight into cyber threat and incident information on Federal Information Systems. At the same time, current contract terms or restrictions may limit the sharing of such threat or incident information with executive departments and agencies (agencies) that are responsible for investigating or remediating cyber incidents, such as the Cybersecurity and Infrastructure Security Agency (CISA), the Federal Bureau of Investigation (FBI), and other elements of the Intelligence Community (IC). Removing these contractual barriers and increasing the sharing of information about such threats, incidents, and risks are necessary steps to accelerating incident deterrence, prevention, and response efforts and to enabling more effective defense of agencies’ systems and of information collected, processed, and maintained by or for the Federal Government.

26638 Federal Register / Vol. 86, No. 93 / Monday, May 17, 2021 / Presidential Documents

The guidelines shall include criteria that can be used to evaluate software security, include criteria to evaluate the security practices of the developers and suppliers themselves, and identify innovative tools or methods to demonstrate conformance with secure practices.

(c) Within 180 days of the date of this order, the Director of NIST shall publish preliminary guidelines, based on the consultations described in subsection (b) of this section and drawing on existing documents as practicable, for enhancing software supply chain security and meeting the requirements of this section.

(d) Within 360 days of the date of this order, the Director of NIST shall publish additional guidelines that include procedures for periodic review and updating of the guidelines described in subsection (c) of this section.

(e) Within 90 days of publication of the preliminary guidelines pursuant to subsection (c) of this section, the Secretary of Commerce acting through the Director of NIST, in consultation with the heads of such agencies as the Director of NIST deems appropriate, shall issue guidance identifying practices that enhance the security of the software supply chain. Such guidance may incorporate the guidelines published pursuant to subsections (c) and (i) of this section. Such guidance shall include standards, procedures, or criteria regarding:

(i) secure software development environments, including such actions as:

(A) using administratively separate build environments;

(B) auditing trust relationships;

(C) establishing multi-factor, risk-based authentication and conditional access across the enterprise;

(D) documenting and minimizing dependencies on enterprise products that are part of the environments used to develop, build, and edit software;

a recurring basis;

(vii) providing a purchaser a Software Bill of Materials (SBOM) for each product directly or by publishing it on a public website;

(viii) participating in a vulnerability disclosure program that includes

(v) providing, when requested by a purchaser, artifacts of the execution of the tools and processes described in subsection (e)(iii) and (iv) of this section, and making publicly available summary information on completion of these actions, to include a summary description of the risks assessed and mitigated;

(vi) maintaining accurate and up-to-date data, provenance (*i.e.*, origin) of software code or components, and controls on internal and third-party software components, tools, and services present in software development processes, and performing audits and enforcement of these controls on a recurring basis;

(vii) providing a purchaser a Software Bill of Materials (SBOM) for each product directly or by publishing it on a public website;

(viii) participating in a vulnerability disclosure program that includes a reporting and disclosure process;

(ix) attesting to conformity with secure software development practices; and

AutoISAC SBOM Working Group - History

NTIA – July 2018 – November 2021

Hitachi – November 2018 – Dec 2021

AutoISAC Phase 1 – Mar-Jul 2019

Sponsor: Analyst WG

Goal: Ensure NTIA SBOM considers automotive industry issues and opinions

Team: 10 members (includes 3 OEMs)

Objective: Publish concerns to NTIA and advocate for the auto industry

AutoISAC Phase 2 – Nov 2020 – Dec 2021

Sponsor: Supplier Affinity Group

Goal: Agree on best practices among suppliers and propose solution to OEMs

Team: 17 members (1 OEM)

Objectives:

- Unified supplier voice on SBOM adoption to OEMs
- Align with NTIA
- Practical approach with input from OEMs
- Best Practice published in 2021

Preview: Best Practice Guide Proposal

WILL INCLUDE

TLP AMBER distribution (for now)
Substantial overlap with NTIA guidance
Customizations for automotive
Mapping to automotive product lifecycle
Format and operational recommendations
Sharing discussion
Vendor-neutral tool list
Bibliography, training, and reference docs

WILL NOT INCLUDE

Mandatory rules – all points will be recommendations
Usurpation of supplier contracts or requirements
Static guidance – revisions expected during Phase 3 and ongoing

Next Steps

1. **Finalize Best Practice Draft Proposal**
2. **Board of Directors approval**
3. **Phase 3 (Likely)– active exercise – details under discussion**
4. **Future Possibilities (not decided)**
   - Limited production pilot exercise
   - Training program
   - Automation and tool trials
   - DHS/CISA program (NTIA successor)
   - Supply chain integrity exercise
   - Vulnerability management use case and exercise
   - Addition of Vulnerability/Exploitability eXchange (VEX) automation