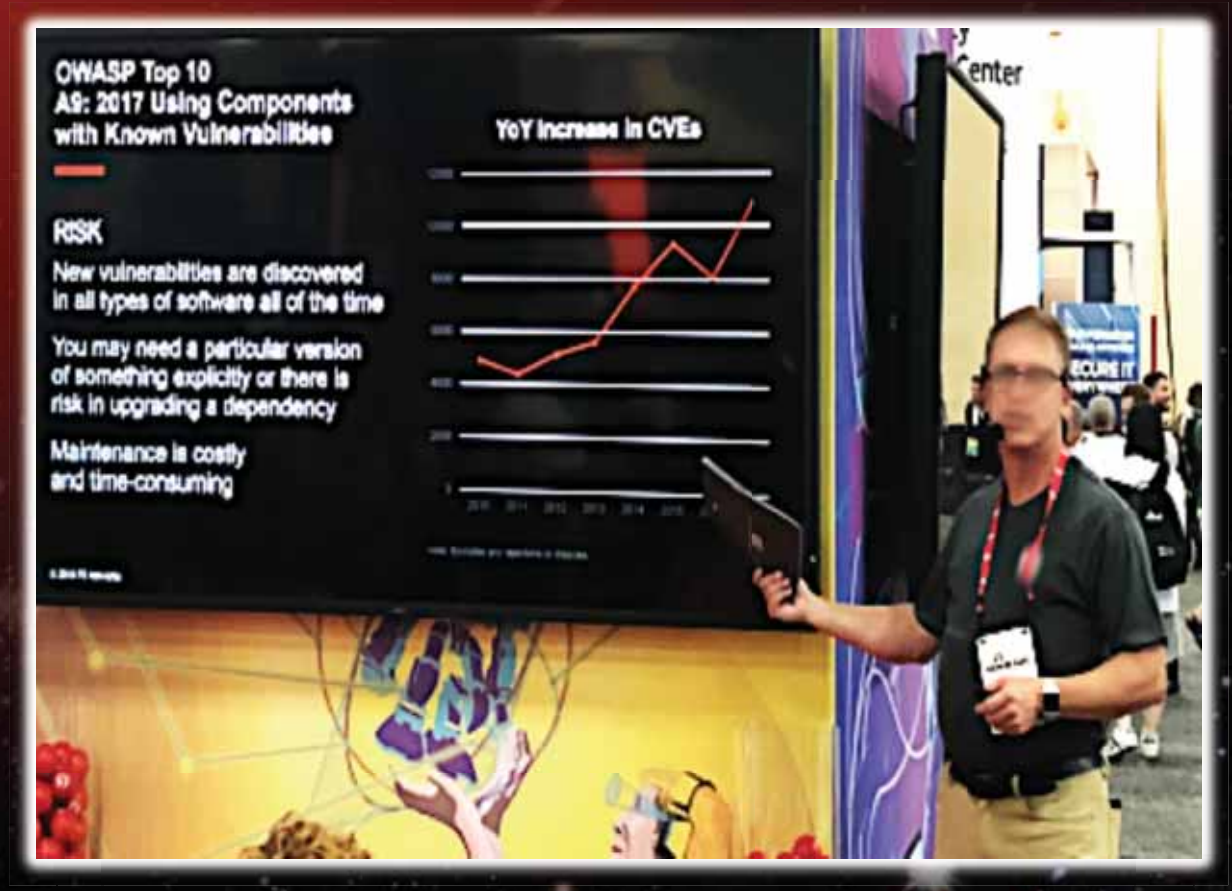
Software Bill of Materials (SBOM) standardization

Robert A. Martin
Sr. Secure Software & Technology Prin. Eng.
Trust & Assurance Cyber Technologies Dept.
Cyber Solutions Technical Center



OMG Technical Meeting | Long Beach, CA

MITRE

All types of Enterprises are Incorporating SW & SW-Enabled Things…

Medical

Buildings

Aeronautics

Manufacturing

Energy

Shipping

Vehicles

MITRE

These Changes Go Well beyond Traditional Information Technology…

Water Treatment

Status & Health Monitoring

Smart Munitions

Remote Management

Oil & Gas

Hydro Power & Dam Mngt

MITRE

Secure Behavior

Reliable Behavior

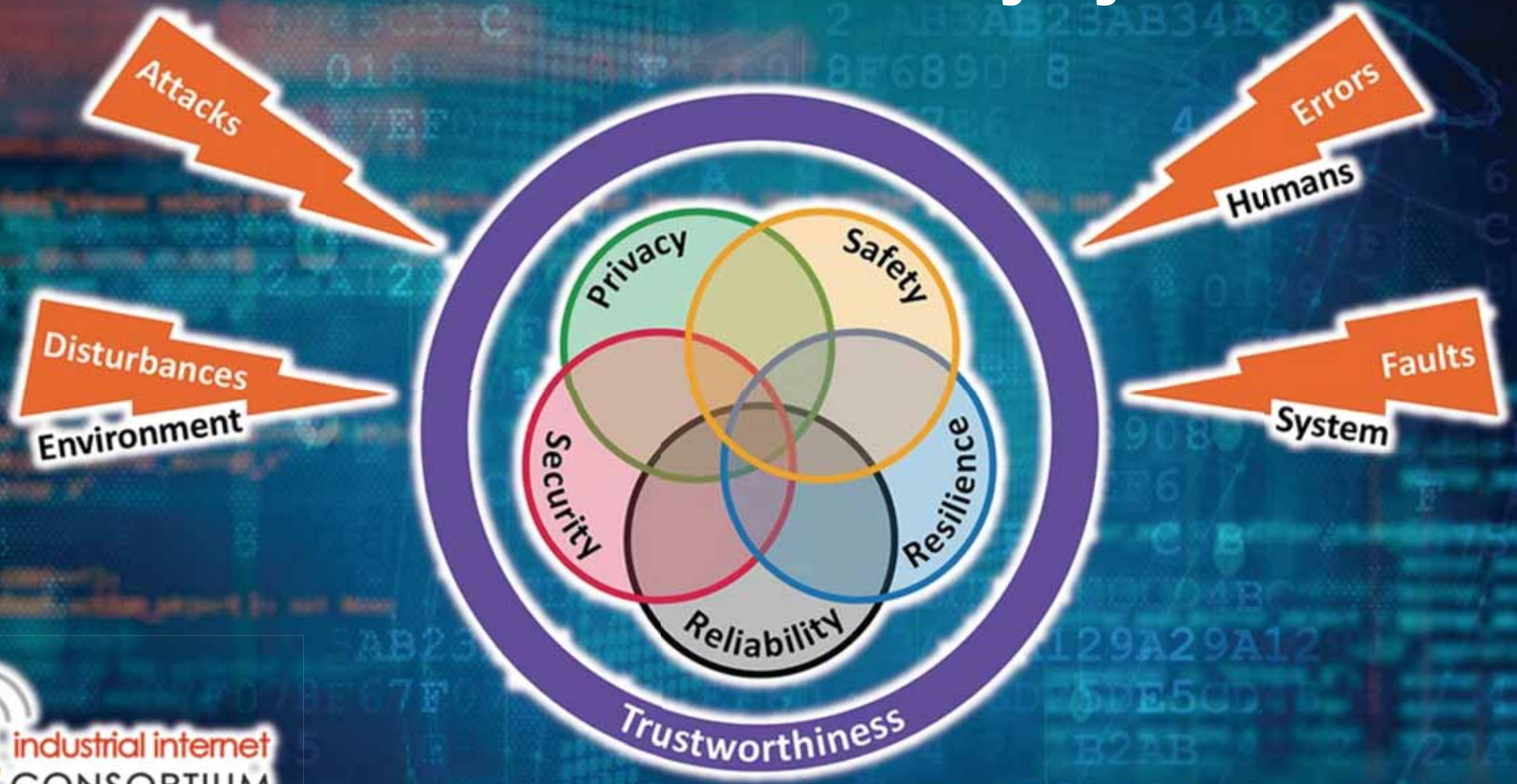
Safe Behavior

Resilient Behavior

Privacy Expectations

MITRE

– Need Assurance of More Than Security –
Need Assured Trustworthy Systems



industrial internet CONSORTIUM

MITRE

Pervasiveness of connected SW & SW-enabled capabilities requires supply chain security skills / new awareness of SW risks

IT Risk → **Operational Risk**

Loss of data or capability → **Loss of safety or reliability** → **Loss of property or lives**

Scratch Built Software → **Assembled Software**

Majority of products built with no 3rd Party dependencies

Use of open source and 3rd party libraries, modules, frameworks, and services
Multi-party software updating/patching

Traditional Computers → **Software Enabled Everything**

Servers
Desktops
Laptops
Tablets
Switches

databases
office apps
e-mail
browsers
Routers

Healthcare
Aeronautics
Smart Energy
Oil & Gas
Microgrids

Implantable Medical
Smart Manufacturing
Water Treatment
Hydro Power
Smart Cities

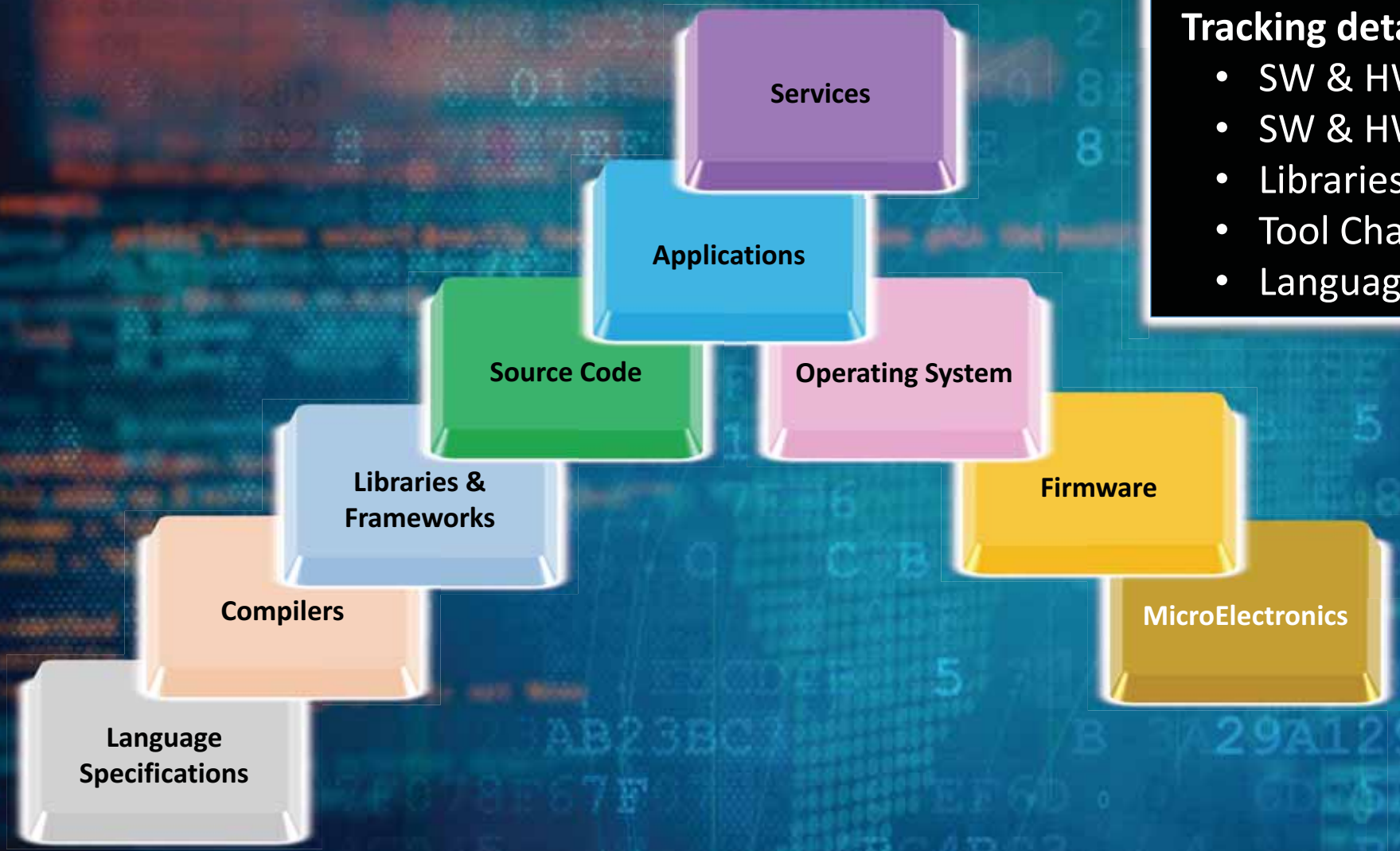
Smart Munitions
Intelligent Vehicles
Intelligent Shipping
Dam Management
Building Management
Autonomous Systems

MITRE

For Software-Enabled IIoT Version Control is Crucial

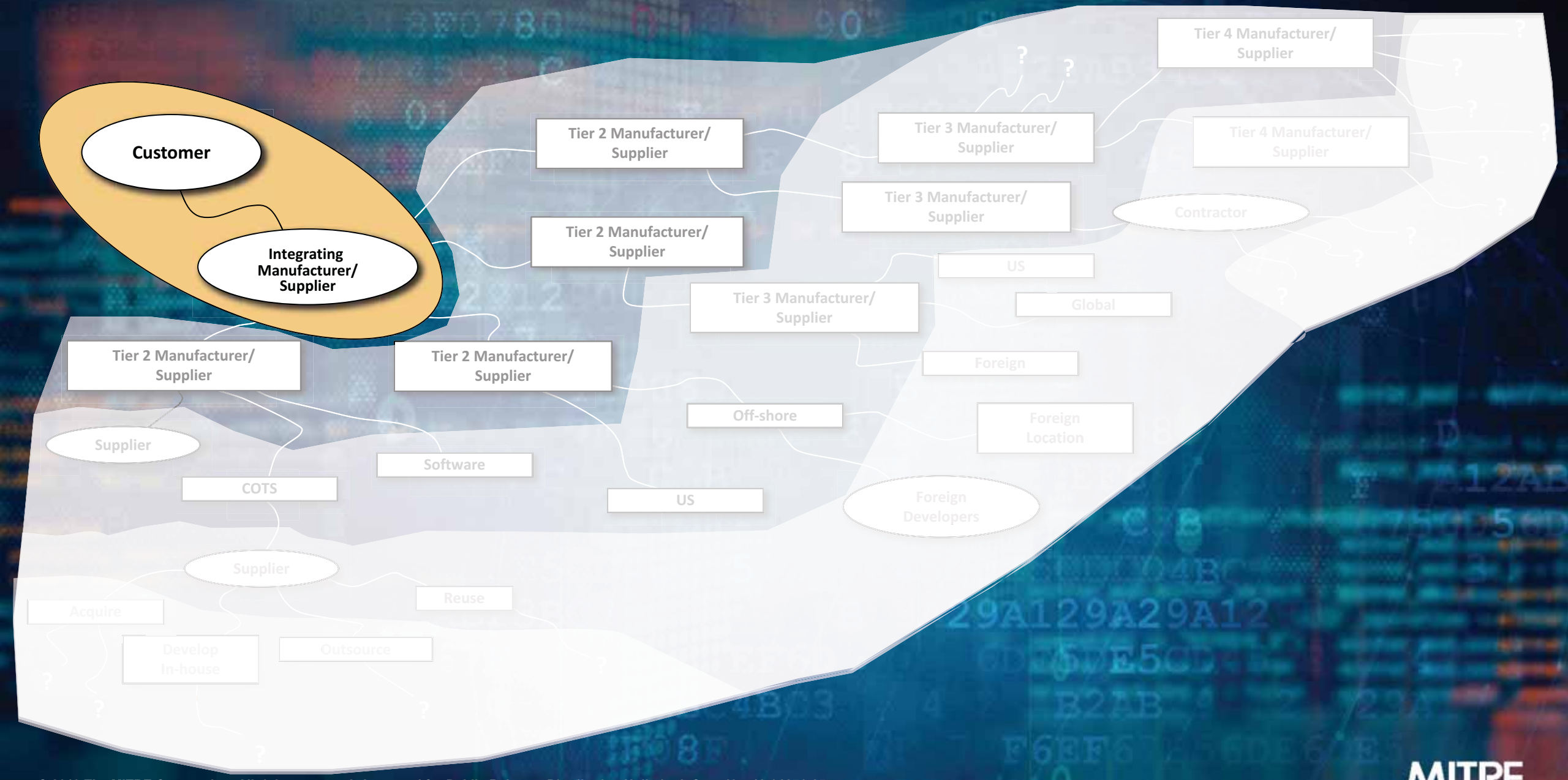
Tracking details for SW & HW components

- SW & HW Part numbers/names
- SW & HW versions
- Libraries & Frameworks Used
- Tool Chain Used/Flags/Options
- Languages & versions used



MITRE

The Supply Chain for Software-Enabled Capabilities is Opaque



MITRE

Market Transparency through "Software Bill of Materials"

- **Third party components are a known systemic risk.**
  - **Transparency can drive tools and behavior to document risk, support mitigations, and drive better SW development practices.**
- **NTIA at Commerce launched an open, community-driven, cross-sector "multistakeholder process" to promote software component transparency.**
  - **Understand the problem and define basics of SBOM**
  - **Develop use cases across sectors on how such data can be used, today and in the future.**
  - **Guidance on how to use existing standards to implement SBOM**
    - **Software ID tags (SWID)**
    - **Software Package Data Exchange (SPDX)**
- **First phase deliverable mid-November 2019**
- **More info or to join: afriedman@ntia.doc.gov**

MITRE

NTIA Transparency Phase 1 Final Products

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

NTIA Multistakeholder Process on Software Component Transparency
Framing Working Group
2019-11-12

Éamonn Ó Muiri
https://flic.kr/p/46dsiz
https://creativecommons.org/licenses/by/2.0/legalcode

Roles and Benefits for SBOM Across the Supply Chain

NTIA Multistakeholder Process on Software Component Transparency
Use Cases and State of Practice Working Group

Survey of Existing SBOM Formats and Standards

Credi

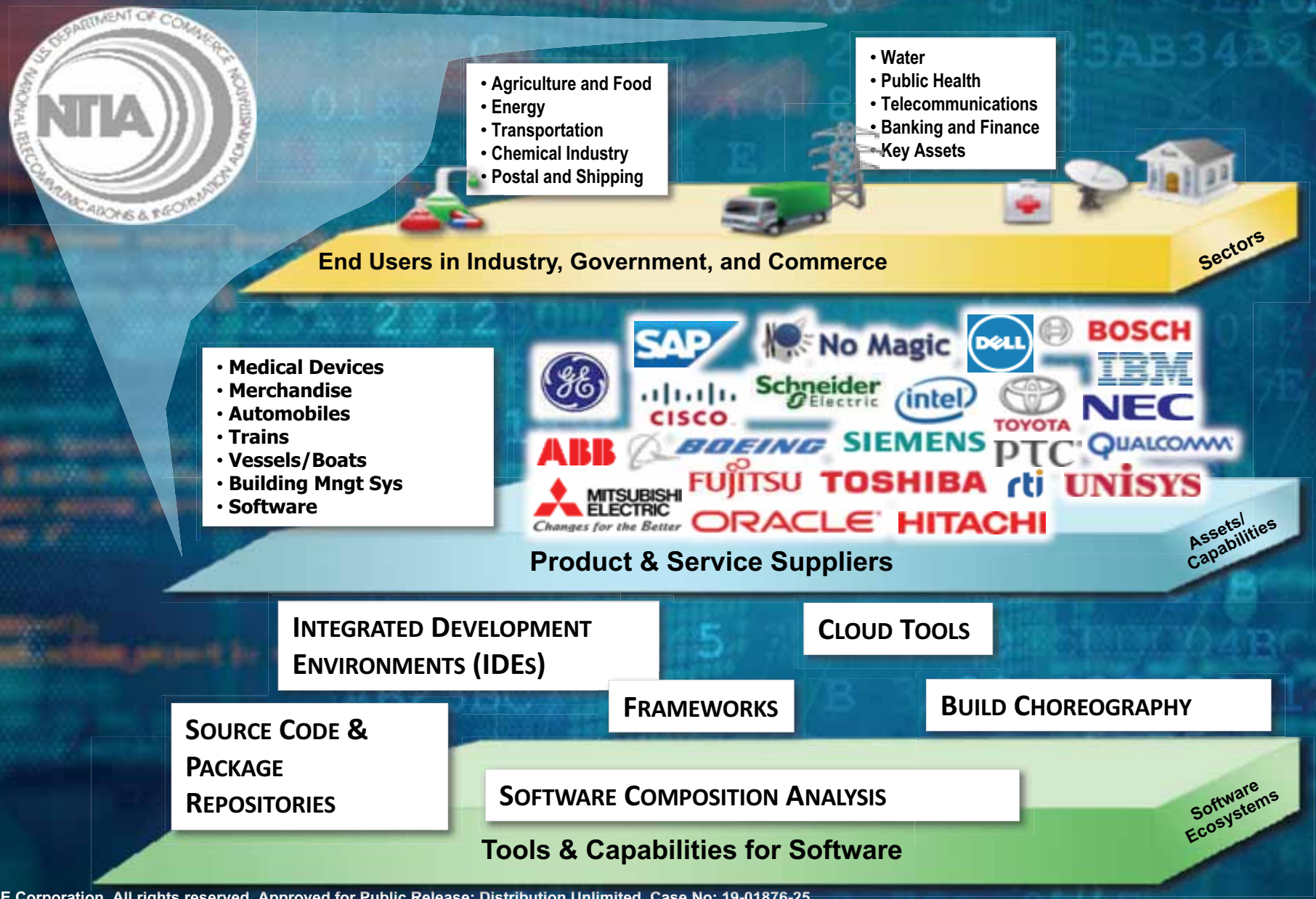
NTIA Multistakeholder Process on Software Component
Standards and Formats Working Group
Final Version - 20191025

SOFTWARE COMPONENT TRANSPARENCY: HEALTHCARE PROOF OF CONCEPT REPORT

Drafted as part of a process convened by the National Telecommunications and Information Administration

October 1, 2019

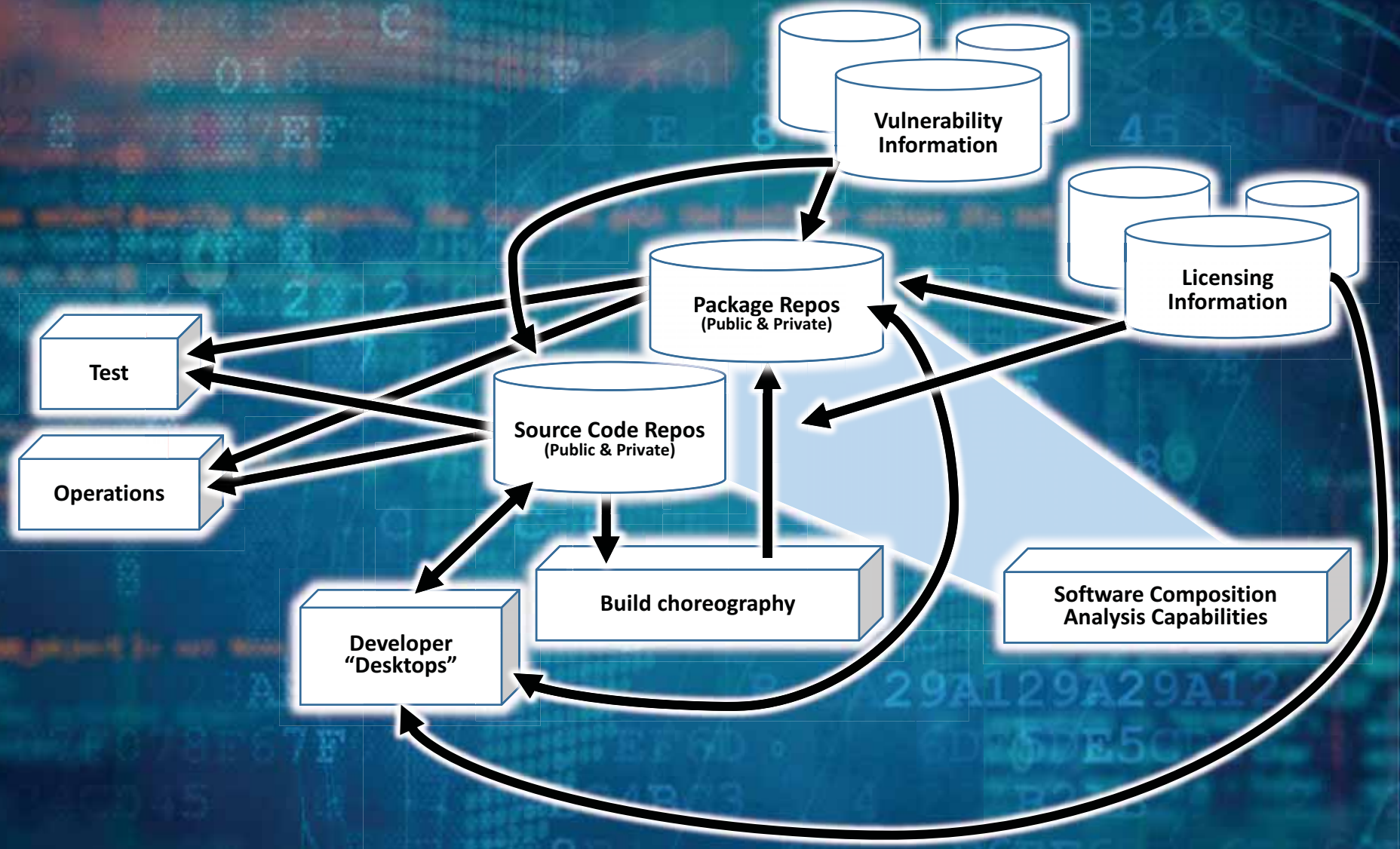
Lowering Adoption Hurdles for SBOMs



Tool-to-Tool SBOM Exchange Standard effort

MITRE

Ecosystem of SW Development, Integration, and Management Tools

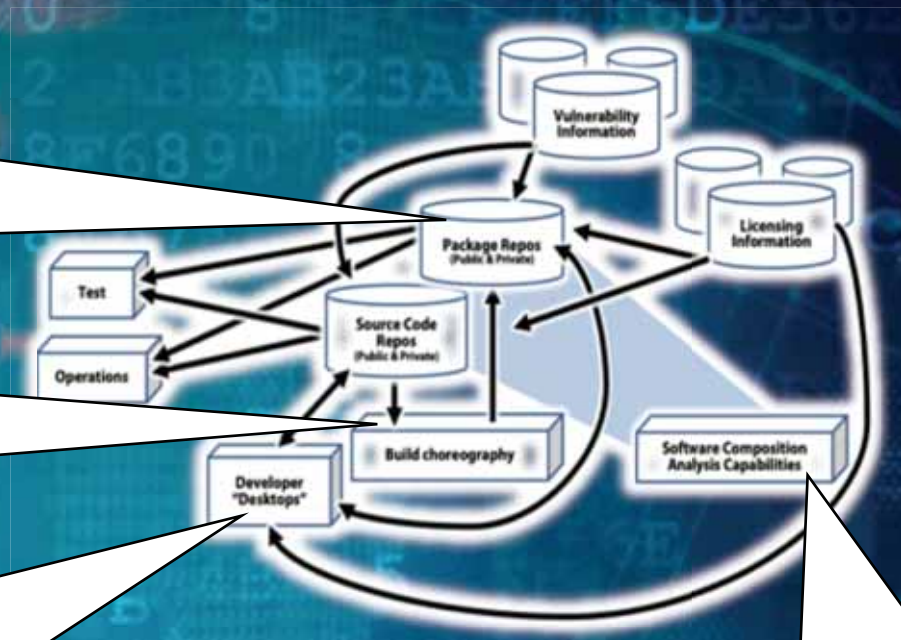


MITRE

SW Development, Integration, and Management Tools

Source Code & Package Repositories

Amazon ECR, Assembla, Azure Container Registry, Beanstalk, Bitbucket, Codebase, Docker, GitHub, GitLab, Glitch, Google Container Registry, JFrog Artifactory, JFrog Xray, inedo, Kubernetes, Launchpad, Maven, Nexus (Sonatype), Phabricator, ProjectLocker, Repository Hosting, Savannah, SourceForge, SourceRepo, Subversion, and Unfuddle



Build & Build Choreography Capabilities

Ansible, Autorabit, Bamboo, Bitrise, Buildkite, Buildroot, CircleCI, CMake, CruiseControl, Final builder, GCC, Gitlab CI, GoCD, Integrity, Jenkins, Strider CD, TeamCity, Terraform, Travis CI, Urbancode, and Vagrant

Developer Desktops (Embedded, Web, Cloud, Desktops/Servers)

IDEs: Android Studio, AppCode, Atom, BlueJ, CLion, Cloud9 IDE, Code Blocks, CodeCharge Studio, CodeLobster, CodePen, DataGrip, Eclipse, GoLand, IDLE, IntelliJ IDEA, LINX, Microsoft Visual Studio, MPLAB, NetBeans, PhpStorm, Pycharm, Rider, RubyMine, Spiralogics Application Architecture, WebStorm, Xcode, and Zend Studio

Frameworks: .NET, Angular, Ansible, Apache Spark, ASP.NET, Bootstrap, Chef, Cordova, CryEngine, Django, Drupal, Express, Flask, Flutter, Hadoop, HTML5 Builder, Laravel, Node.js, Pandas, Puppet, React Native, React.js, Ruby on Rails, Spring, TensorFlow, Torch/PyTorch, Unity D, Unreal Engine, Visual Online, Vue.js, and Xamarin

Cloud Tools: Azure, AWS CodeBuild, Cloud Foundry, Google Cloud Build, Kwatee, Pivotal, and Red Hat

Software Composition Analysis:

Black Duck Software Composition Analysis (Synopsys), CAST Highlight (CAST Software), Finate State, FlexNet Code Insite (Flexera), Ion Channel, Insignary, SourceClear, Sonatype, Snyk, and WhiteSource

MITRE

Usage Scenarios for Tool-to-Tool SBoM

- **Refer, Transfer or Purchase** (definition of what it is)
- **Proper and Legal** (conditions about its use)
- **Pedigree** (history of how it was produced)
- **Known Sw Vulns** (known fixes are applied to it)
- **Provenance** (chain of custody of it)
- **Assurance** (safe-secure-resilient)
- **Integrity** (cryptographic basis of unalteredness)
- **SBoM of a SW Service** (SBoM of sw delivering service)

Supply Chain Sequence Integrity

MITRE

Provenance and Pedigree

DEFINITIONS

- *Provenance**
  1. The origin, or source of something
  2. The history of ownership of a valued object, or work of art, or literature

- *Pedigree**
  1. A register recording a line of ancestors
  2. An ancestral line : lineage
     The origin and the history of something; broadly : background, history

CONFUSION

- *Many use "Provenance" for both meanings.*

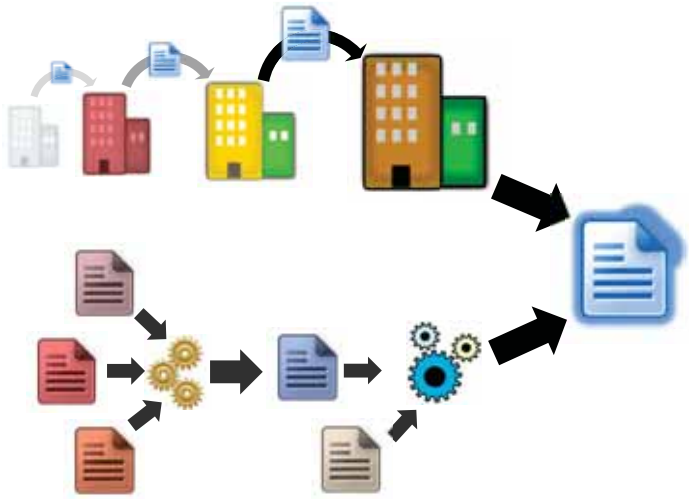
  The provenance of a piece of data is both the custodianship as well as the lineage of processing and/or derivation that led to the piece of data.

**Definitions (from Merriam-Webster.com)*

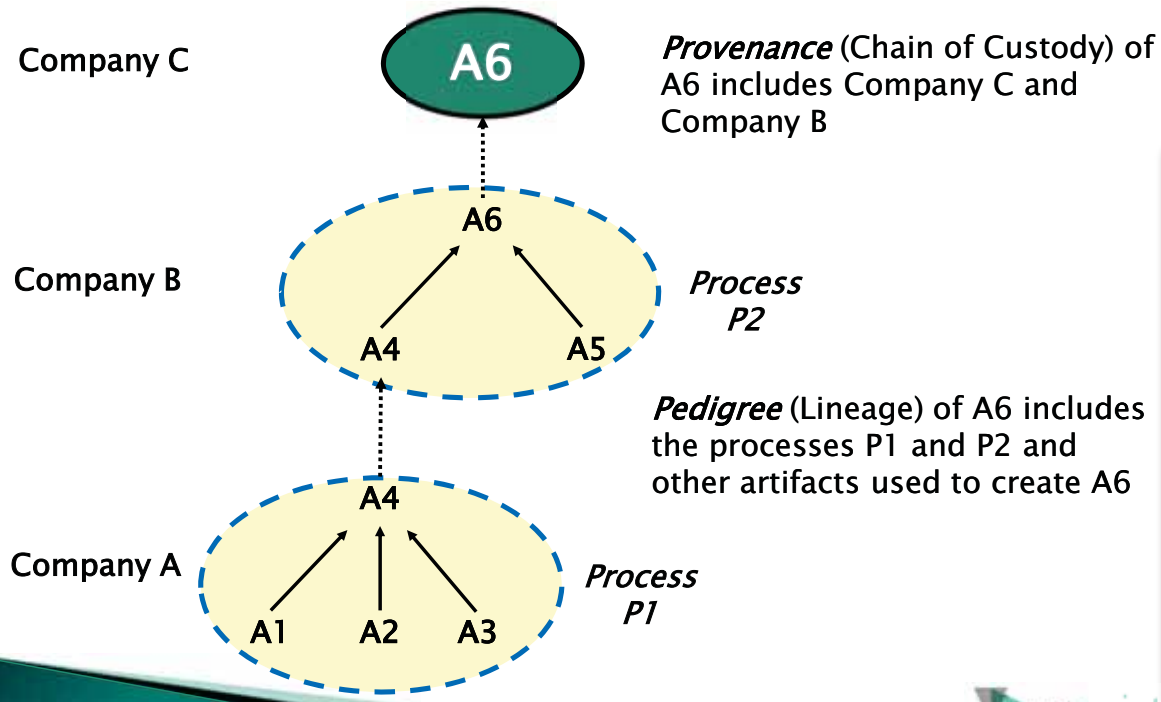
Separating Provenance and Pedigree

Provenance
Captures ***chain of custody*** of an Artifact, Document or Record

Pedigree
Captures the ***history*** of how an Artifact or Document *was produced or derived*

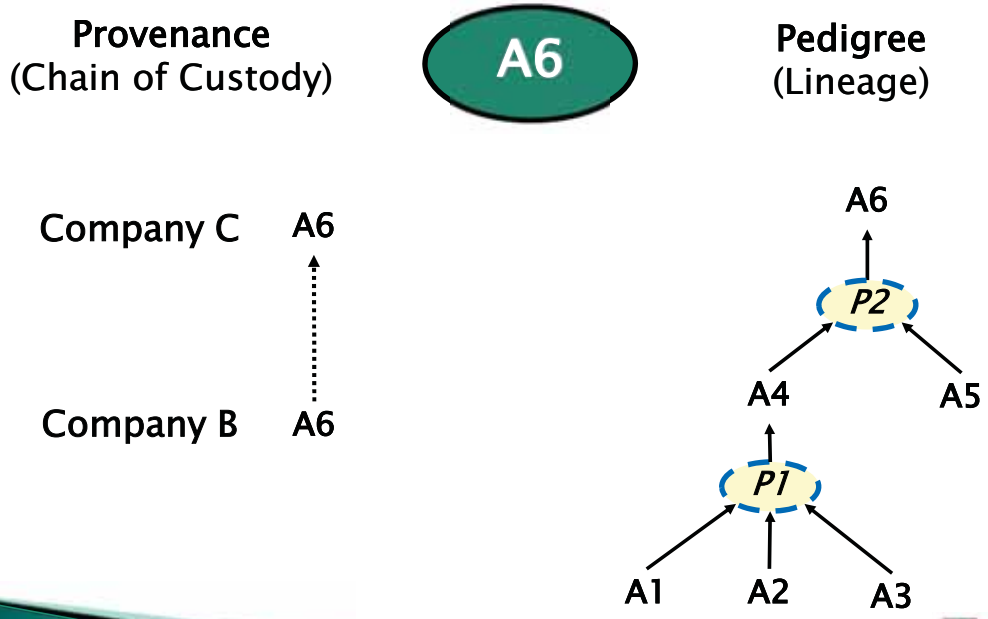


Combined Pedigree & Provenance



Separating Pedigree & Provenance

Provenance and *Pedigree* provide a basis on which to reason about the *trustworthiness* of an artifact or document



The Path to Code Provenance at Uber

April 17, 2019

Uber

Code Provenance

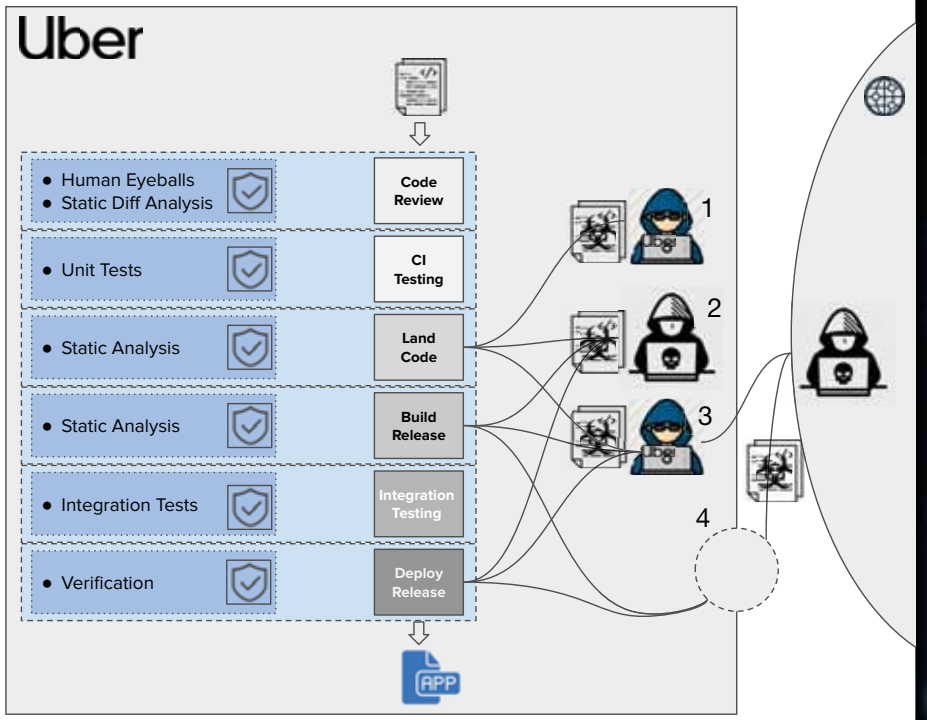
Ensuring we have a **verifiable attestation** of the **origin of all code** running in production so that we can have a **root of trust** as we move forward to **defining** and **enforcing** a collection of **policies** throughout the different stages of the **software development process**.

MITRE

Code Provenance

What are we protecting against?

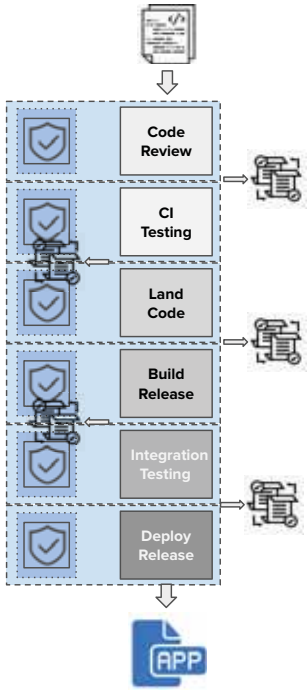
1. Lazy / shortcutting insider
2. Malicious insider
3. Engineer laptop controlled by malicious outsider
4. Build / deploy infrastructure attacked by malicious outsider



Code Provenance

What do we get out of all this?

- "Chain of custody" for all code landing in production releases
- Enabling response in the event that anything goes awry
- Flexible, enforced policies for what code is allowed to land in production releases



MITRE

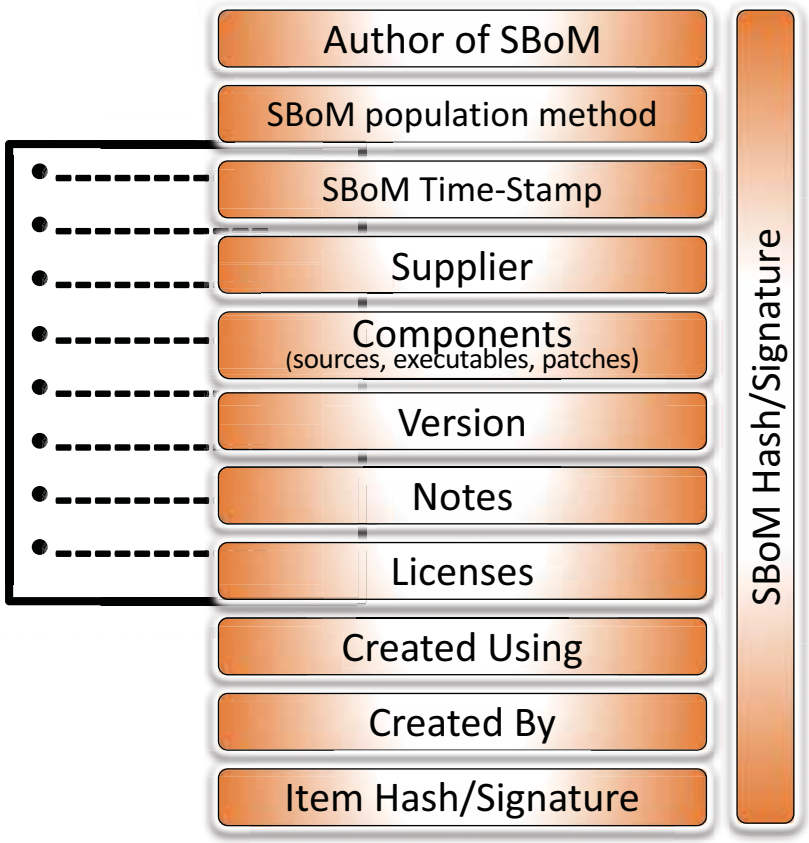
Usage Scenarios and Tool-to-Tool SBOM candidate elements

CISQ
Consortium for IT Software Quality

Usages

1. Refer, Transfer or Purchase (definition of what it is)
2. Pedigree (history of how it was produced)
3. Provenance (chain of custody of it)
4. Integrity (cryptographic basis of unalteredness)
5. Intellectual Property Constraints
6. Known SW Vulns (known fixes are applied to it)
7. Assurance (secure-safe-resilient)
8. SBoM of a SW Service (SBoM of sw delivering service)
9. Supply Chain Sequence Integrity

SBoM elements



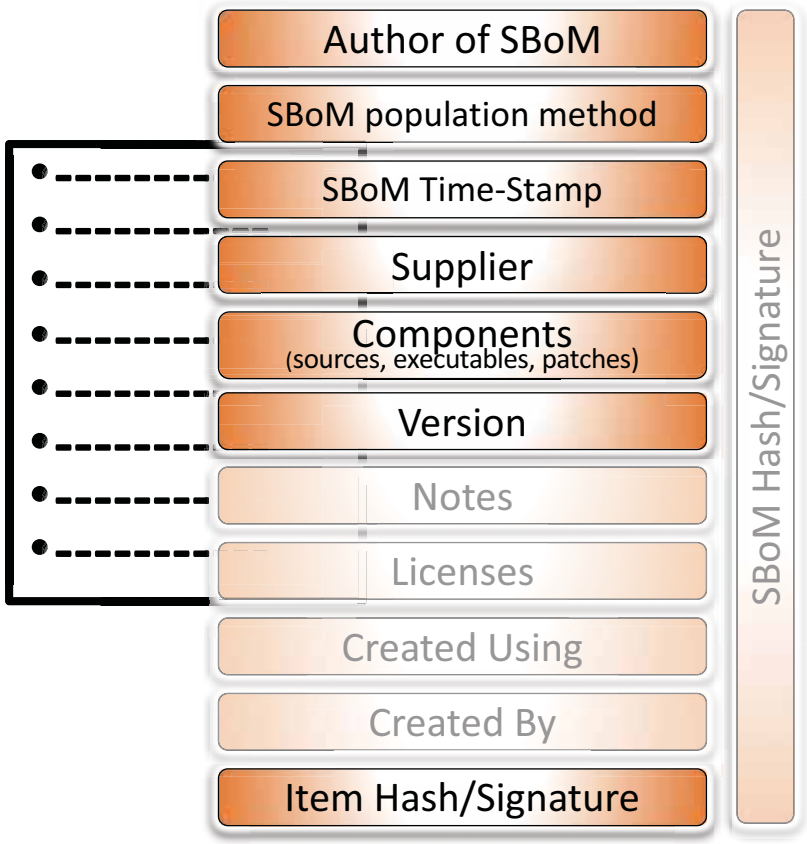
Correlated Info

CISQ
Consortium for IT Software Quality

Usages

1. **Refer, Transfer or Purchase** (definition of what it is)
2. Pedigree (history of how it was produced)
3. Provenance (chain of custody of it)
4. Integrity (cryptographic basis of unalteredness)
5. Intellectual Property Constraints
6. Known SW Vulns (known fixes are applied to it)
7. Assurance (secure-safe-resilient)
8. SBoM of a SW Service (SBoM of sw delivering service)
9. Supply Chain Sequence Integrity

SBoM elements



Correlated Info

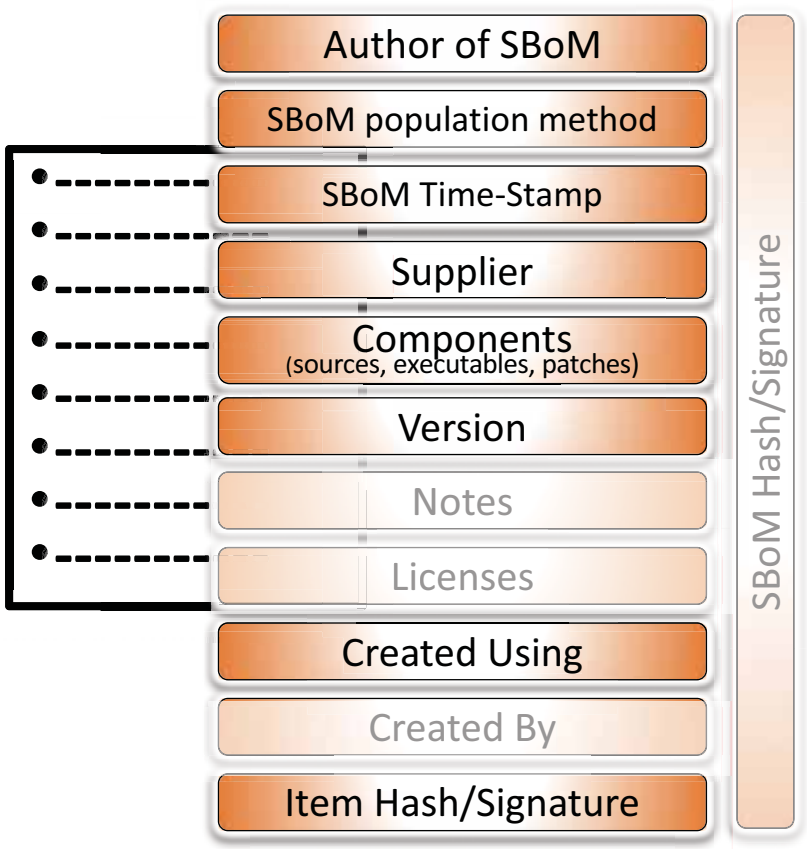
None

CISQ
Consortium for IT Software Quality

Usages

1. Refer, Transfer or Purchase (definition of what it is)
2. **Pedigree (history of how it was produced)**
3. Provenance (chain of custody of it)
4. Integrity (cryptographic basis of unalteredness)
5. Intellectual Property Constraints
6. Known SW Vulns (known fixes are applied to it)
7. Assurance (secure-safe-resilient)
8. SBoM of a SW Service (SBoM of sw delivering service)
9. Supply Chain Sequence Integrity

SBoM elements



Correlated Info

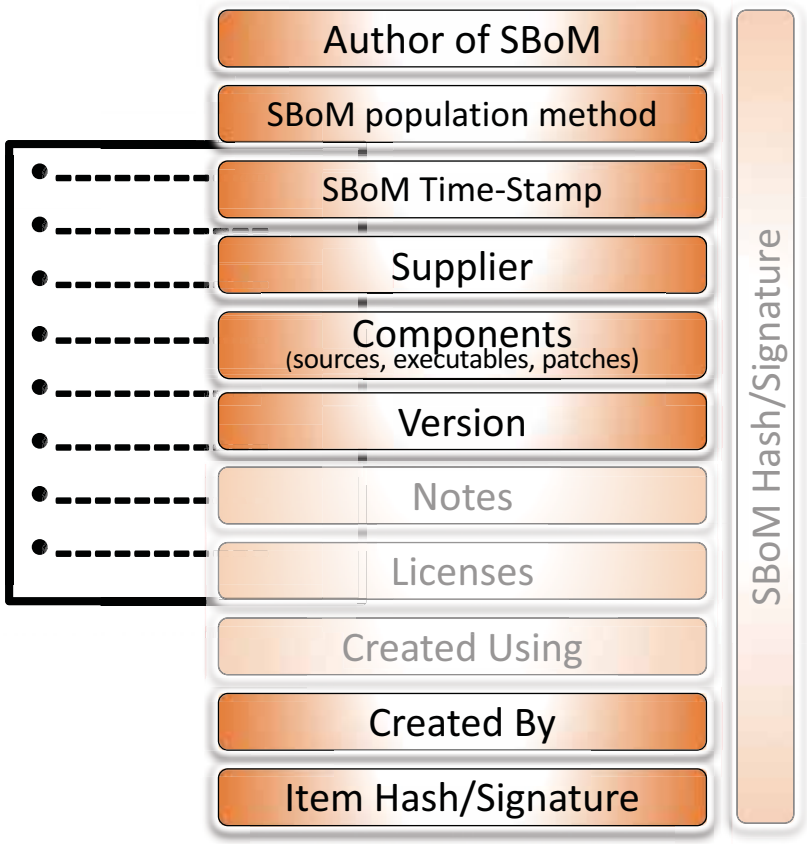
None

CISQ
Consortium for IT Software Quality

Usages

1. Refer, Transfer or Purchase (definition of what it is)
2. Pedigree (history of how it was produced)
3. **Provenance (chain of custody of it)**
4. Integrity (cryptographic basis of unalteredness)
5. Intellectual Property Constraints
6. Known SW Vulns (known fixes are applied to it)
7. Assurance (secure-safe-resilient)
8. SBoM of a SW Service (SBoM of sw delivering service)
9. Supply Chain Sequence Integrity

SBoM elements



Correlated Info

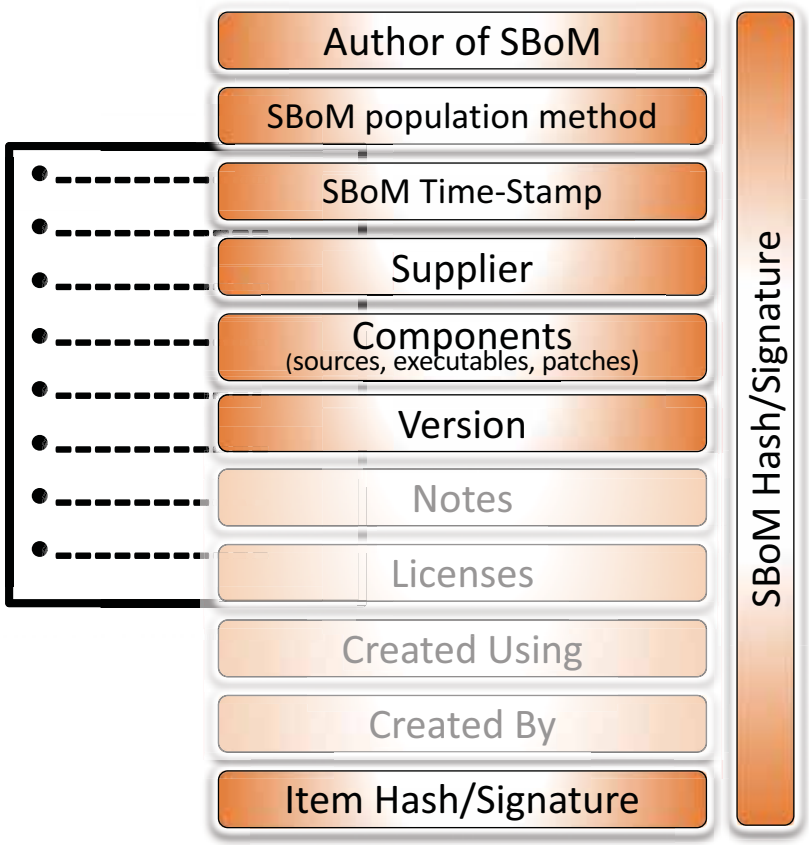
None

CISQ
Consortium for IT Software Quality

Usages

1. Refer, Transfer or Purchase (definition of what it is)
2. Pedigree (history of how it was produced)
3. Provenance (chain of custody of it)
4. **Integrity (cryptographic basis of unalteredness)**
5. Intellectual Property Constraints
6. Known SW Vulns (known fixes are applied to it)
7. Assurance (secure-safe-resilient)
8. SBoM of a SW Service (SBoM of sw delivering service)
9. Supply Chain Sequence Integrity

SBoM elements



Correlated Info

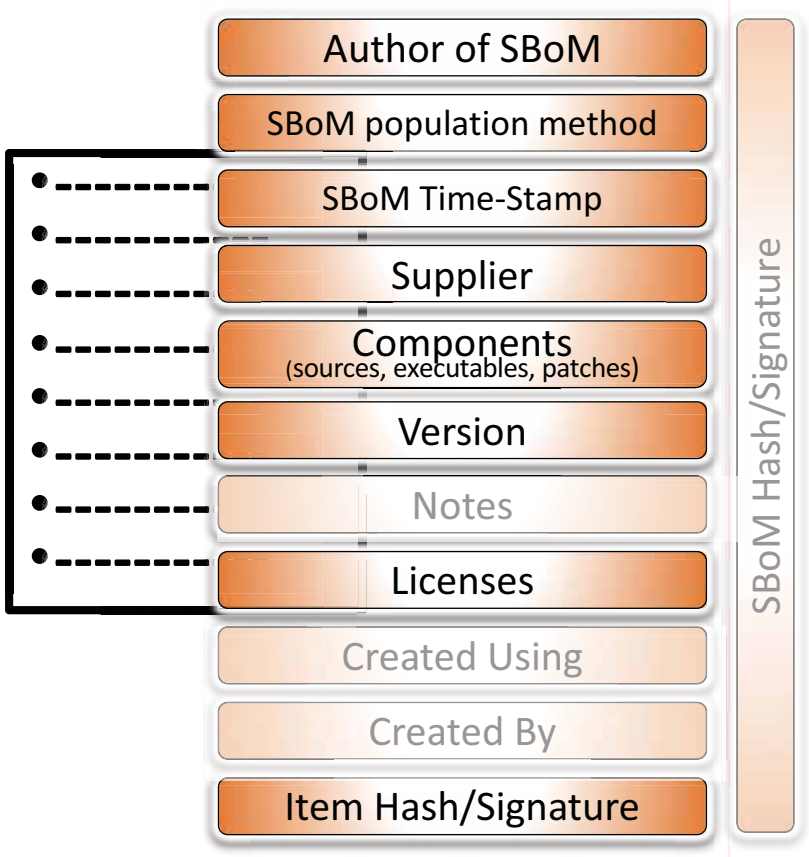
None

CISQ
Consortium for IT Software Quality

Usages

1. Refer, Transfer or Purchase (definition of what it is)
2. Pedigree (history of how it was produced)
3. Provenance (chain of custody of it)
4. Integrity (cryptographic basis of unalteredness)
5. **Intellectual Property Constraints**
6. Known SW Vulns (known fixes are applied to it)
7. Assurance (secure-safe-resilient)
8. SBoM of a SW Service (SBoM of sw delivering service)
9. Supply Chain Sequence Integrity

SBoM elements



Correlated Info

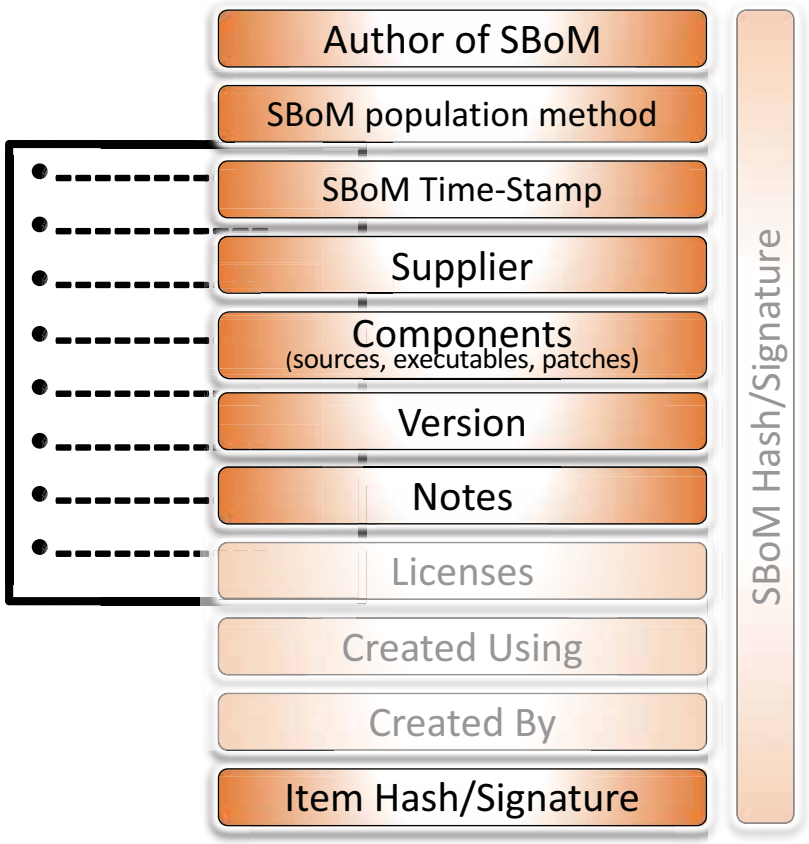
None

CISQ
Consortium for IT Software Quality

Usages

1. Refer, Transfer or Purchase (definition of what it is)
2. Pedigree (history of how it was produced)
3. Provenance (chain of custody of it)
4. Integrity (cryptographic basis of unalteredness)
5. Intellectual Property Constraints
6. **Known SW Vulns (known fixes are applied to it)**
7. Assurance (secure-safe-resilient)
8. SBoM of a SW Service (SBoM of sw delivering service)
9. Supply Chain Sequence Integrity

SBoM elements



Correlated Info

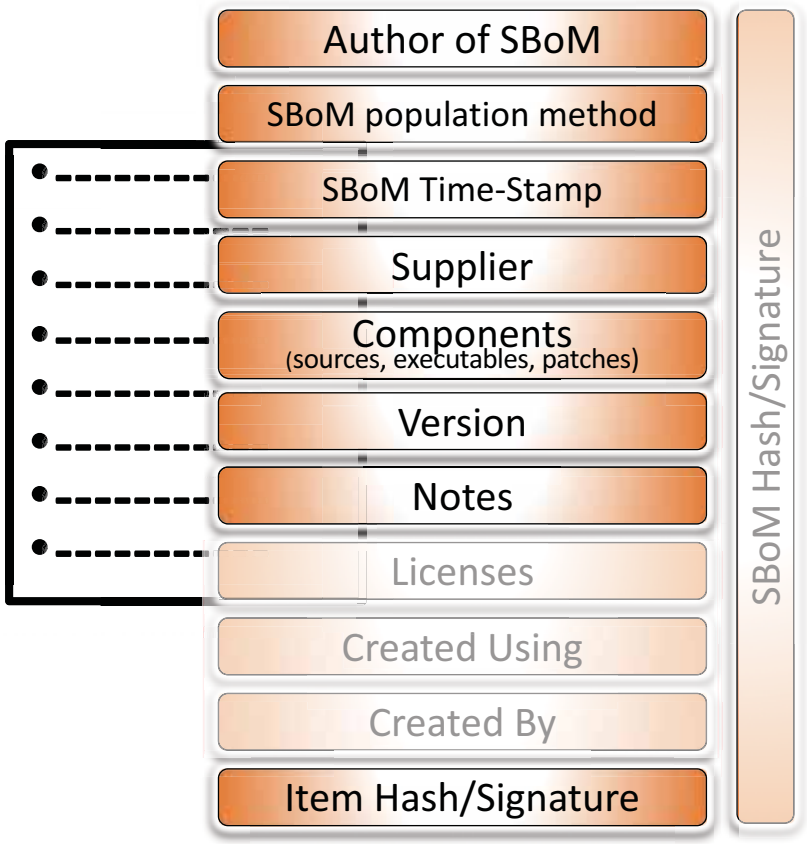
Vulnerability Knowledge Bases
Vulnerability Management Systems

CISQ
Consortium for IT Software Quality

Usages

1. Refer, Transfer or Purchase (definition of what it is)
2. Pedigree (history of how it was produced)
3. Provenance (chain of custody of it)
4. Integrity (cryptographic basis of unalteredness)
5. Intellectual Property Constraints
6. Known SW Vulns (known fixes are applied to it)
7. **Assurance (secure-safe-resilient)**
8. SBoM of a SW Service (SBoM of sw delivering service)
9. Supply Chain Sequence Integrity

SBoM elements



Correlated Info

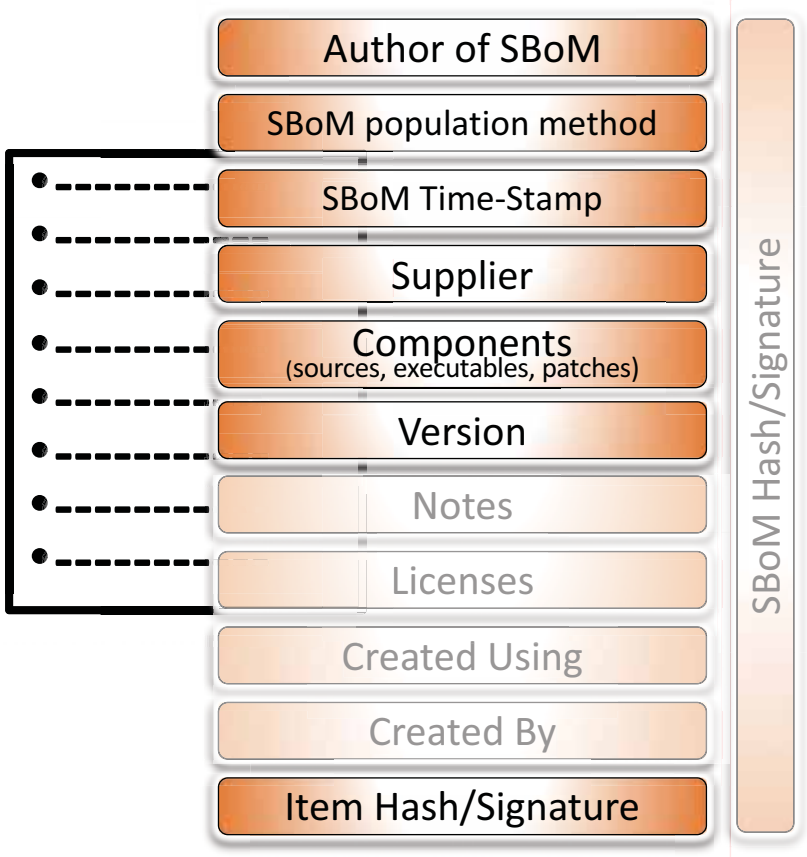
Notes on exploitability of vulns
Vulnerability Knowledge Bases
Weakness Knowledge Bases
Assessment Results
Design Review
Code Review
Attack Surface Analysis
Static Analysis
Dynamic Analysis
Fuzz Testing
Pen Testing
Blue Teaming
Red Teaming
Organized as an Assurance Case

CISQ
Consortium for IT Software Quality

Usages

1. Refer, Transfer or Purchase (definition of what it is)
2. Pedigree (history of how it was produced)
3. Provenance (chain of custody of it)
4. Integrity (cryptographic basis of unalteredness)
5. Intellectual Property Constraints
6. Known SW Vulns (known fixes are applied to it)
7. Assurance (secure-safe-resilient)
8. **SBoM of a SW Service (SBoM of sw delivering service)**
9. Supply Chain Sequence Integrity

SBoM elements



Correlated Info

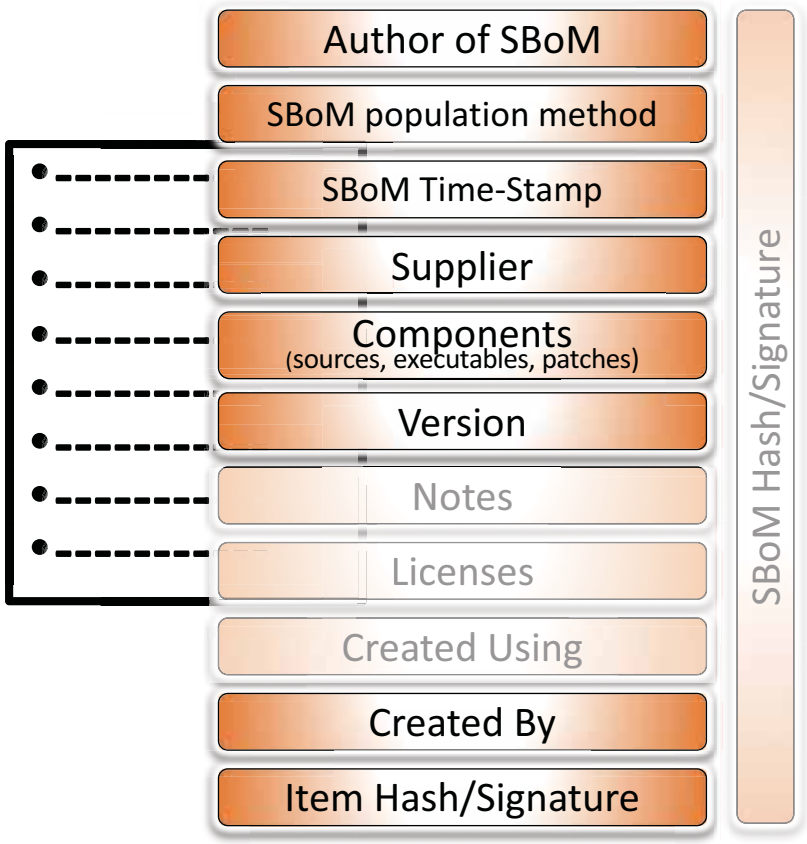
Logging SBOMs of Services Used

CISQ
Consortium for IT Software Quality

Usages

1. Refer, Transfer or Purchase (definition of what it is)
2. Pedigree (history of how it was produced)
3. Provenance (chain of custody of it)
4. Integrity (cryptographic basis of unalteredness)
5. Intellectual Property Constraints
6. Known SW Vulns (known fixes are applied to it)
7. Assurance (secure-safe-resilient)
8. SBoM of a SW Service (SBoM of sw delivering service)
9. **Supply Chain Sequence Integrity**

SBoM elements



Correlated Info

Desired sequence of ordered software supply chain steps, and requirements for each step for a specific project of interest

Launched 24 Sep 2019

CISQ

STANDARDS USE CASES RESOURCES ABOUT CISQ ACTIVE PROJECTS

WORKING GROUP

TOOL-TO-TOOL SOFTWARE BILL OF MATERIALS EXCHANGE

OBJECTIVE

This is a joint working group of CISQ and the Object Management Group (OMG). Defining an exchangeable tool-to-tool software bill of materials (SBOM) metamodel is the primary goal of this working group. The work leverages the efforts of the National Telecommunications and Information Agency's (NTIA's) Software Component Transparency initiative but with a focus on the exchange of SBOMs between and among the software development tools that create, revise, manage, orchestrate, and/or otherwise manipulate software.

Like a bill of materials for physical items, the SBOM is a comprehensive inventory of the software raw materials, subassemblies, parts and components, needed to create a software product. Typically, an SBOM is hierarchical in nature and multi-level.

With today's software creation processes, many of these subassemblies will take the form of third-party components from open source software or other commercial providers. Concerns about the origin and chain of custody can also be captured and conveyed with an SBOM, along with relevant information about the process(es) and choices that the software creation activity underwent that can influence the customers' acceptance and confidence in the software's quality and appropriateness for the intended use by the customer.

TIMETABLE

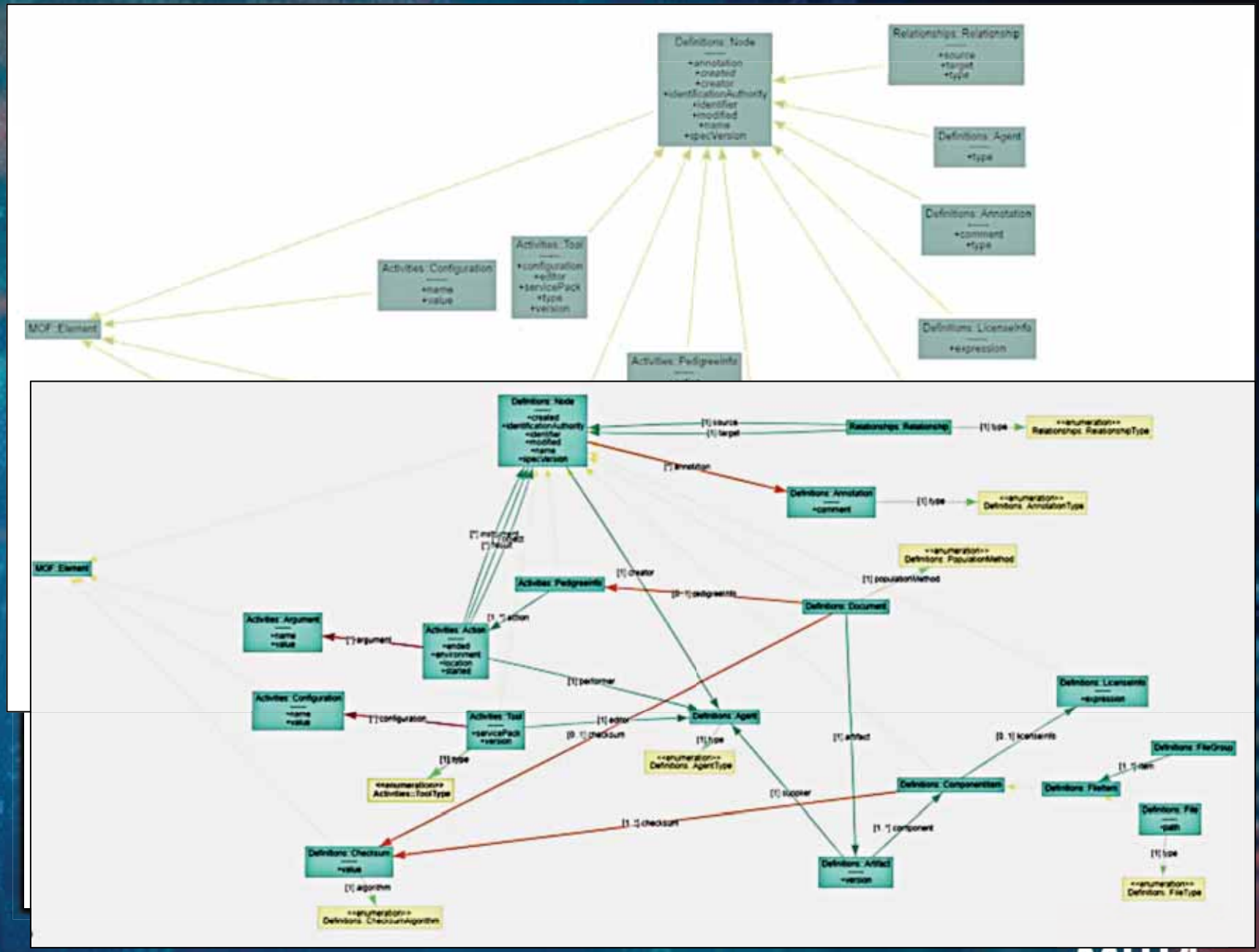
The kick-off of this effort was 24 September 2019 with a three-day workshop on the concept and ideas for a tool-to-tool focused initiative.

CHAIRS

- Bob Martin (MITRE)
- Dr. Bill Curtis (CISQ)
- Kay Williams (Microsoft - Azure - CD Foundation)

PARTICIPANTS

- Philippe-Emmanuel Doutreich (CAST)
- Santiago Torres-Arias (in-toto/NYU)
- David Nalley (BlackBerry - Apache Foundation)
- William Cox (Black Duck by Synopsys)
- Steve Looker (Microsoft - Artifact Storage, Open Container Initiative)
- Brian Russell (Google - CD Foundation)
- Nitesh Bakliwal (Microsoft - Windows)
- Kate Stewart (Linux Foundation - SPDX)
- William Bartholomew (GitHub)
- David Edelsohn (IBM - GCC)
- Jason Shaver (Microsoft - Developer Division)
- Fahad Ahmad (Microsoft - Build Systems)
- JC Herz (Jon Channel)
- Adam Baldwin (npm, Inc.)
- Gerald Heidereich (Microsoft - CloudBuild)
- Dan Lorenc (Google - CD Foundation)
- Jeffrey Martin (White Source Software)
- Michael Muller (CAST)
- Bryan Sullivan (Microsoft - Security and Compliance)
- Brian Fox (Sonatype)
- Steve Springett (OWASP, CycloneDX)
- Fred Blonse (CloudBees - Jenkins)
- Ido Green (JFrog)
- Gary O'Neall (Source Auditor - SPDX)
- Anna Debenham (Snyk)
- Ian Geoghegan (Microsoft - Software Supply Chain Security)
- Duncan Sparrell (sFractal Consulting)



MITRE

Table 1: IDEs in Use Survey

Surveyed Projects →	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Android Studio																							
AppCode																							
Atom																							
BlueJ																							
CLion					X																		
Cloud9 IDE																							
Code Blocks																							
CodeCharge Studio																							
CodeLobster																							
CodePen																							
DataGrip					X																		
Eclipse	X		X			X																	
GoLand					X																		
IDLE								X															
IntelliJ IDEA					X																		
LINX											X	X	X										
Microsoft Visual Studio		X	X				X	X		X													
MPLAB																							
NetBeans			X																				
PhpStorm					X																		
Pycharm					X																		
Rider					X																		
RubyMine					X																		
Spiralogics Application Architecture																							
WebStorm																							
Xcode																							
Zend Studio																							

Table 2: Frameworks in Use Survey

Surveyed Projects →	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Bootstrap										X	X												
HTML5 Builder																							
Microsoft Expression Studio																							
Visual Online																							

Table 3: Cloud Tools in Use Survey

Surveyed Projects →	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Azure							X			X	X												
AWS CodeBuild												X		X									
Cloud Foundry																	X						
Google Cloud Build																							
Kwatee																							

ackage Repositories in Use Survey

Surveyed Projects →	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Amazon ECR														X									
Assembla																							
Azure Container Registry																							
Beanstalk																							
Bitbucket		X	X	X						X	X	X											
Codebase								X			X			X									
Docker			X	X	X				X		X	X	X	X									
GitHub			X				X	X															
GitLab			X	X	X	X			X														
Glitch																							
Google Container Registry																							
JFrog Artifactory					X																		
JFrog Xray					X																		
Inedo																							
[illegible]									X		X		X										
[illegible]											X												
Phabricator																							
ProjectLocker																							
Repository Hosting																							
Savannah																							
SourceForge						X	X																
SourceRepo																							
Subversion	X						X	X															
Unfuddle																							

Table 5: Build & Build Choreography Capabilities in Use Survey

Surveyed Projects →	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Ansible																										
Autorabit																										
Bamboo																										
Bitrise																										
Buildkite																										
Buildroot									X																	
CircleCI																										
CMake									X																	
CruiseControl																										
Final builder																										
GCC									X																	
Gitlab CI																	X									
GoCD																										
Integrity																										
Jenkins		X	X	X	X	X	X		X	X	X		X													
Strider CD																										
TeamCity																										
Terraform																										
Travis CI																										
Urbancode																										
Vagrant																										

Table 4: Software Composition Analysis Capabilities in Use Survey

Surveyed Projects →	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
Black Duck Software Composition Analysis (Synopsys)																		
CAST Highlight (CAST Software)																		
FlexNet Code Insite (Flexera)																		
Ion Channel																		
SourceClear																		
Snyk																		
WhiteSource																		

Whitepaper → CISQ → OMG RFC → ISO Std

- Socialize at Mar19 OMG meeting
- Draft SBoM as a Whitepaper in 3-day CISQ SBoM working session at Sep OMG meeting
- Prototype draft format in tool ecosystem, revise and draft RFC based on prototype results
- Co-submit draft RFC w/CISQ to OMG at ~~Dec19 or~~ Mar20 meeting
- ~~Mar20~~/Jun20 OMG meeting – charter FTF
- ~~Jun20~~/Sep20 OMG meeting - approve as OMG Standard
- ~~Sep20~~/Dec20 Fast Track to ISO

MITRE

Questions?

MITRE