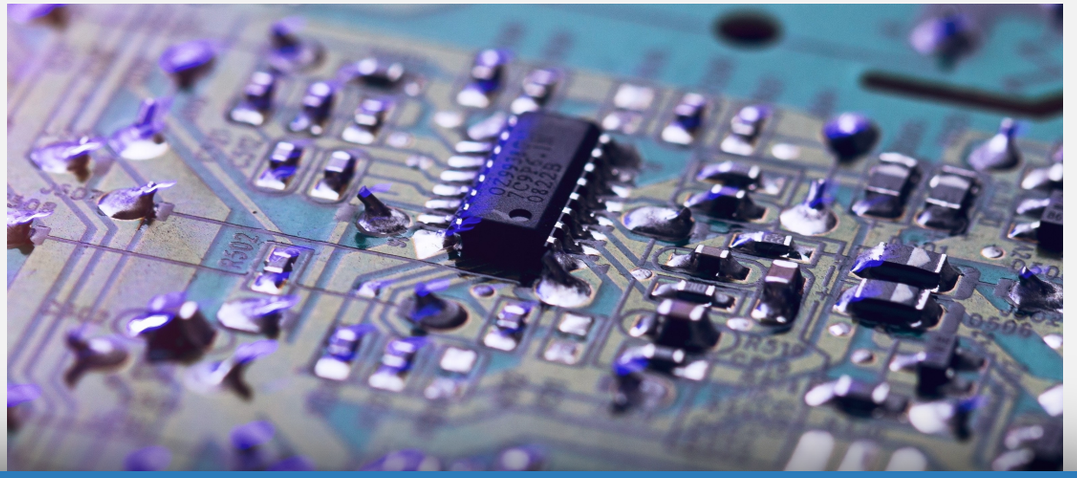
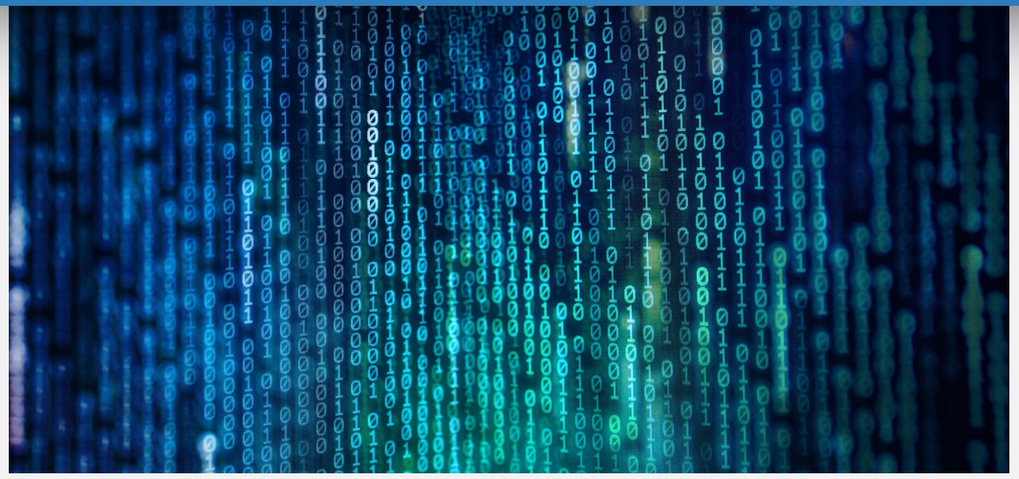


U.S. DEPARTMENT OF ENERGY | OFFICE OF Cybersecurity, Energy Security, and Emergency Response

CyTRICS™ Cyber Testing for Resilient Industrial Control Systems

SBOM Use Cases

Hannah Pearson-Kleinheider



Overview

- Use cases
  - What questions do you want to answer?
  - How will SBOMs provide value to your organization?
- Considerations for different use cases
  - Level of detail or “depth”
  - What information to gather
  - SBOM format and fields used
- Example: CyTRICS use of SBOMs

So … what are SBOMs good for?

- Supply Chain security
- Vulnerability management
- Risk assessment
- License management
- … and more!

SBOM Depth in General

- Varies depending on use case
- From most-to-least detail required:
  - Supply Chain security
  - Vulnerability management
  - Risk assessment
  - License management
- Why?
  - It all depends on what questions you want to answer …

Considerations for SBOM Depth

- Most important aspect: **accuracy**
  - BOM needs to be correct in order to be useful
  - Completeness (more depth and detail) is desirable, but never at the cost of accuracy
- Try exercises testing your ability to obtain answers from your BOMs
  - “Suppose some instances of component XYZ are compromised. What information is needed to identify the compromised components?”

Example: CyTRICS

- Use cases
  - Supply Chain security
  - Vulnerability assessment
- Either way, a high degree of detail is required!
  - Ability to identify compromised components
  - Find common components across an industry
    - Which components are sufficiently widespread that a single vulnerability could have significant impact?
  - Connect known vulnerabilities in components to systems that use those components

Final Thoughts

- Let your use case inform your data collection
- SBOMs are a tool that provides business value
  - Incident response
  - Risk assessment and mitigation
  - Customer support
  - Patching and vulnerability mitigation decisions
- Start using SBOMs now so you have them when you need them!

Questions?

hannah.kleinheider@inl.gov

STAMP: Software Trace of a Manufacturing Process/Product

Dr. Gabriela Ciocarlie, Dr. Dongyan Xu, Dr. Ananth Grama, Michael Focosi

Nov 15th, 2023

Preliminaries: The Software Bill of Materials (SBOM)

- A Software Bill of Materials (SBOM) is a formal record containing the details and supply chain relationships of various components used in building software
- Section 10(j) of EO 14028 defines an SBOM as a “formal record containing the details and supply chain relationships of various components used in building software”
- These components, including libraries and modules, can be open source or proprietary, free or paid, and the data can be widely available or access-restricted

[Source: `ntia.gov/sbom`]

Preliminaries: SBOM Structure

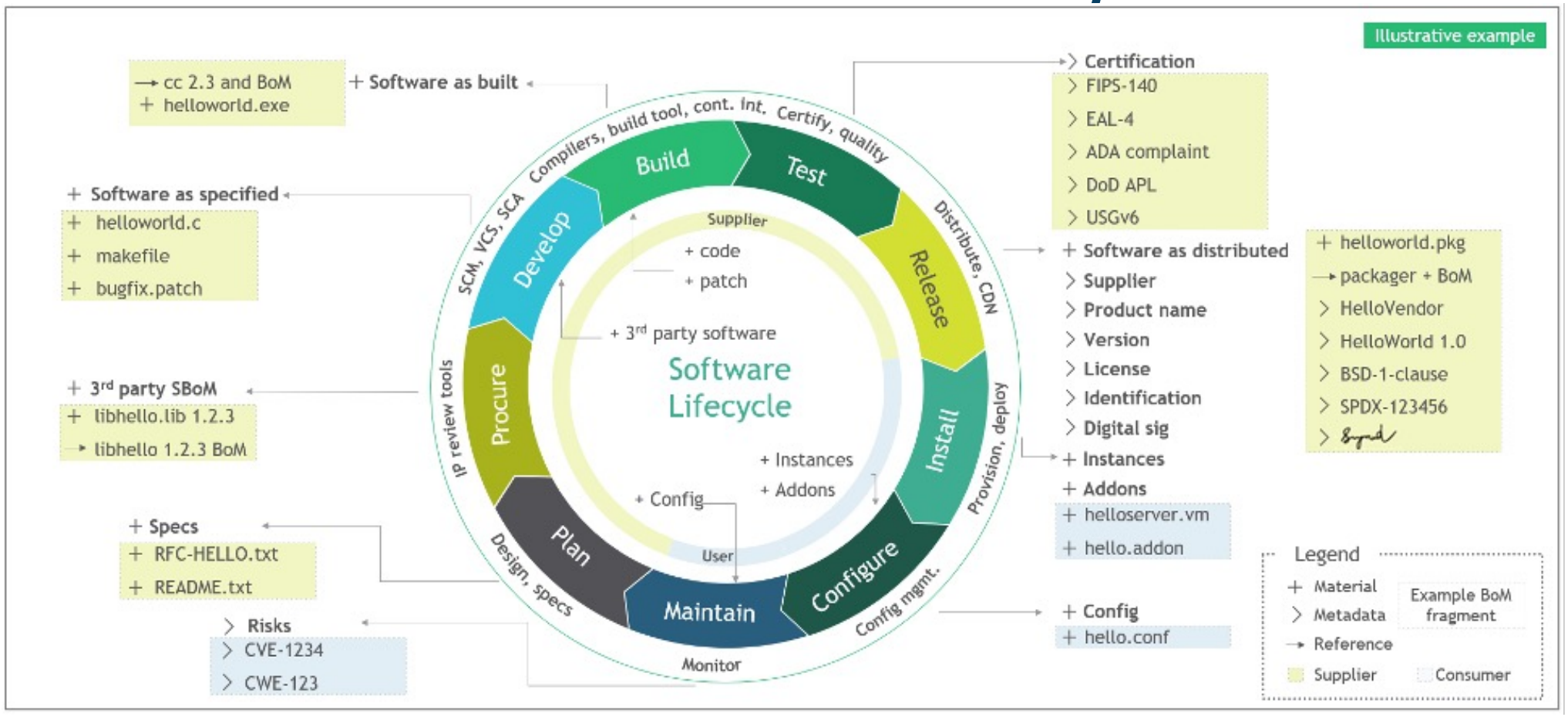
- The primary purpose of an SBOM is to uniquely and unambiguously identify components and their relationships to one another
- Baseline information may be augmented with several other fields to enable application-specialization

[Source: ntia.gov/sbom]

Baseline Component Information
Author Name
Supplier Name
Component Name
Version
Component Hash
UID
Relationship

CYMANII

Preliminaries: SBOMS and Software Lifecycle



Example of Software Life Cycle and Bill of Materials Assembly (source `https://www.nist.gov/itl/executive-order-14028-improving-nations-cybersecurity/software-security-supply-chains-software-1#_ftn1`)

CYMANII

Preliminaries: SBOM Tools

- A number of tools provide support for identifying software entities and conveying associated metadata (producing, consuming, and transforming software entity metadata)
  - NIST Software Identification Tags (SWID Tags, https://csrc.nist.gov/projects/Software-Identification-SWID/),
  - CycloneDX (https://cyclonedx.org/)
  - Software Package Data Exchange (SPDX, https://spdx.github.io/spdx-spec/v2.3/)
  - Supply Chain Levels for Software Artifacts (SLSA, https://slsa.dev/)

Important Note: "SBOMs and the improved transparency that they are meant to provide for federal acquirers are a complementary, not substitutive, capability. Federal acquirers that are unable to appropriately ingest, analyze, and act on the data that SBOMs provide will likely not improve their overall C-SCRM posture." EO 14028

Preliminaries: SBOM Benefits

SBOMs are primarily used by three primary groups:

- **Software developers** use SBOMs to assist in the building and maintenance of their software, including upstream components
- **Software procurers** use SBOMs to inform pre-purchase assurance, negotiate discounts, or plan implementation strategies
- **Software users/operators** use SBOMs to inform vulnerability management and asset management, to manage licensing and compliance, and to quickly identify software or component dependencies and supply chain risks

Preliminaries: SBOM Interoperability

- The Manufacturer Disclosure Statement for Medical Device Security (MDS) provides medical device manufacturers with a means for disclosing to healthcare providers the security-related features of their medical devices
  - **The SBOM section of the MDS was created with these parallel efforts in mind**
- OpenC2 is a standardized language for the command and control of cybersecurity
  - **OpenC2 has commands for obtaining the SBOM of a device, for analyzing the SBOM, and for taking appropriate actions based on the analysis (e.g. connect, patch, sandbox, or block)**
- Manufacturer Usage Descriptions (MUD) describe IoT devices, their capabilities, and their needs
  - **An extension to those descriptions can inform local deployments on how to find an SBOM by pointing to a URL, indicating appropriate local mechanisms, or indicating a point of contact for further information**
- DBOM is a common backbone for attestation sharing including data such as SBOMs among supply chain partners

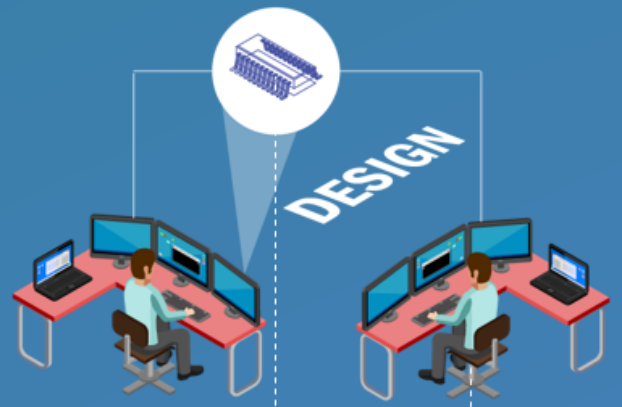
CyManII's Cyber-Physical Passports

- Collect **provenance information** across the automation and supply chain networks with **non repudiation**
- Provide a **multiple-perspective** view of manufacturing processes and products across **cyber, physical, and energy** dimensions
- Enable **supply chain verification, analytics, and integrity checks** with privacy-preserving policies in place

Offer the "know how/when/where" of manufacturing products

CYMANII

The CPP follows a product along its lifecycle.



Cyber-Physical Passport Types

Product-Item Passport

- Product-Item UUID
- Authenticating Agent ID, Encryption Key
- Product Passport ID
- Machine Passport ID
- Physical Attributes Data (dimension, geometry)
- Product Manufacturing Data (time-series data on product item, process, environment and energy)
- Component List (Product-Item UUIDs of components-BoM)
- Human-Agent UUID, Process Parameters
- Analysis-Result UUID
- Product-Item Passport Hash

Human-Agent Passport

- Human-Agent UUID
- Authenticating Agent ID, Encryption Key
- Human Agent Role/Access
- Human-Agent Passport Hash

Machine Passport

- Machine UUID
- Authenticating Agent ID, Encryption Key
- Machine Parameters, State, Ownership, Location
- Location
- Machine-Type UUID
- Machine Passport Hash

Analysis Passport

- Analysis UUID
- Authenticating Agent ID, Encryption Key
- Analysis Specification
- Analysis Result
- Software UUID
- Analysis Passport Hash

Machine-Type Passport

- Machine-Type UUID
- Authenticating Agent ID, Encryption Key
- Machine Specifications
- Software UUID
- Machine-Type Passport Hash

Product Passport

- Product UUID
- Authenticating Agent ID, Encryption Key
- Product Owner UUID
- Product Design (e.g. CAD Design) UUID
- Product Software UUID (Compiler, GCode)
- Product Descriptor
- Functional Descriptor
- Product Passport Hash

Software Passport

- Software UUID
- Authenticating Agent ID, Encryption Key
- Software Version, Configuration, Dependencies, Parameters
- Software Passport Hash

Product-Design Passport

- Product-Design UUID
- Authenticating Agent ID, Encryption Key
- Product-Design Owner UUID
- Product-Designer UUID
- Product Design
- Version Information
- Software UUID (e.g. Fusion 360)
- Product-Design Passport Hash

Passports have the same schema but capture different relevant information

CYMANII


From CPP to STAMP…

STAMP: Software Trace of a Manufacturing Process/Product

- A STAMP or its associated part is **a complete trace of all software components** that impact the function or structure of a manufactured part, process, or operation

- A STAMP is substantially more powerful than an SBOM, since the latter is defined **for a software object**, whereas the former is defined for **a physical artifact**

This difference has significant implications for how STAMPs are constructed, manipulated, and analyzed

SBOMs are typically (small) parts of STAMPs

CYMANII

STAMP vs. SBOM

- An SBOM is defined for a **software object**, therefore all provenance links manifest in the object (either as code or library calls)
- A STAMP is defined for a **physical artifact, process, or operation**. The artifact may itself have software components for which the SBOM describes the software provenance
- The STAMP of a physical artifact also includes all software used in its **design, design compilation, the software controlling the machine used to manufacture the part**, etc.
- Since a part may be an assembly of sub-parts, the STAMP of a part **includes references to STAMPs** associated with its subparts

Linkages in SBOMs are explicit in the software target, while for STAMPs are established across physical objects

CYMANII

STAMPS: Construction and Data Structures

- Since linkages are established through part/process/operational composition, the core data structure much establish a linkage between an artifact and its STAMP, along with all of the constituent subparts
  - The core data structure can then be used to construct the complete STAMP for a given part
- **STAMPS in our system are also represented as CPP passports** (in addition to all of the physical artifacts)
- Our CPP system also provides **linkage across parts and sub-parts**, and can therefore be used to construct complete STAMPs

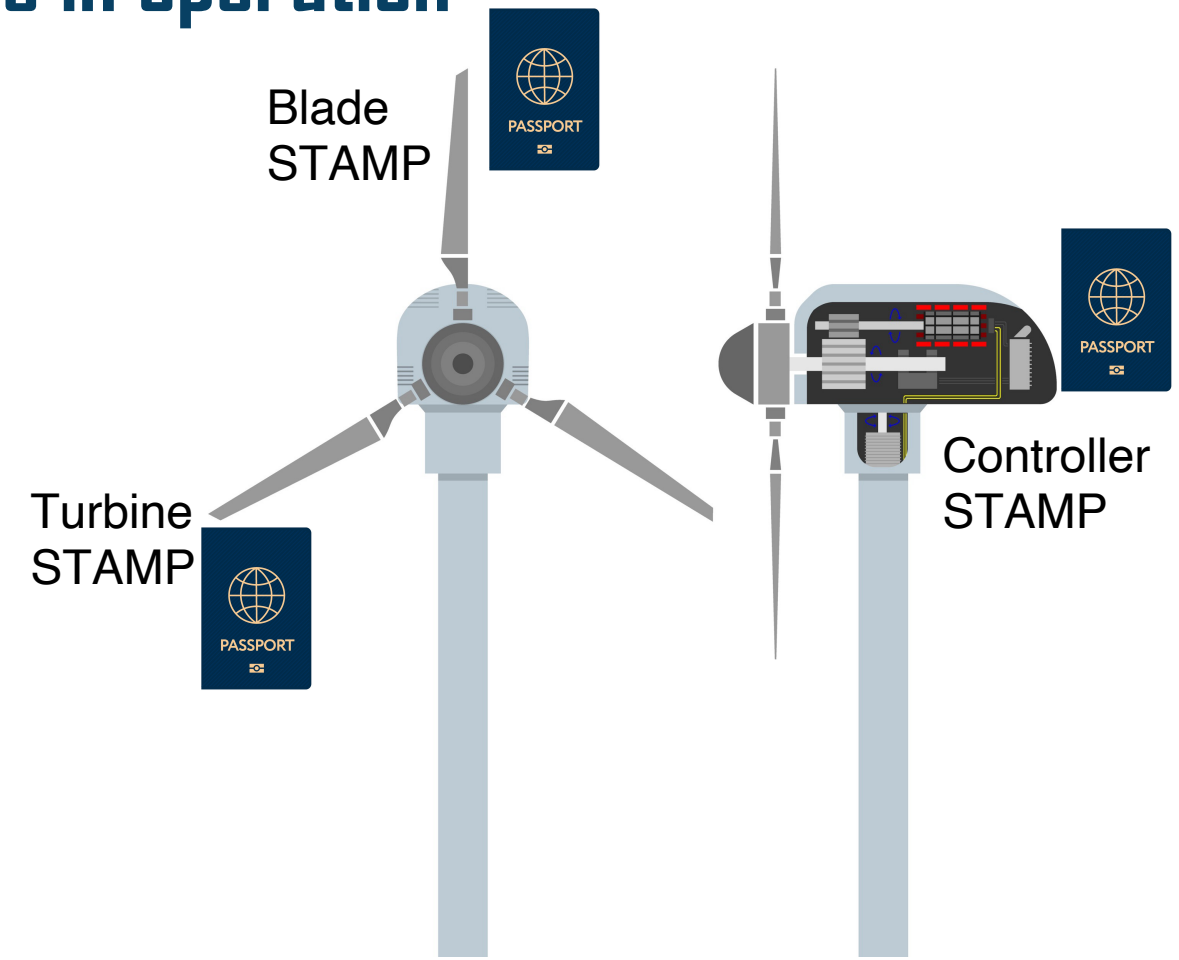
STAMPS: Uses and Benefits

- Since STAMPs provide a comprehensive software trace of an artifact, they can be used to analyze the impact of software on the structure and function of a given part
  - **This includes security as well as design flaws**
- A direct application of STAMPs is in **root-cause analysis** of an observed structural or functional anomaly in a part
- Since STAMPs impact structure and function, they can be associated with functional **digital twins** or **structural part descriptions to enable complex analyses**

Use Case: Controller on a turbine in operation

- A **STAMP for a turbine** links to the **STAMP of its constituent blades,** even though there may be no software link between the **controller for the turbine** and any **software associated with a blade**

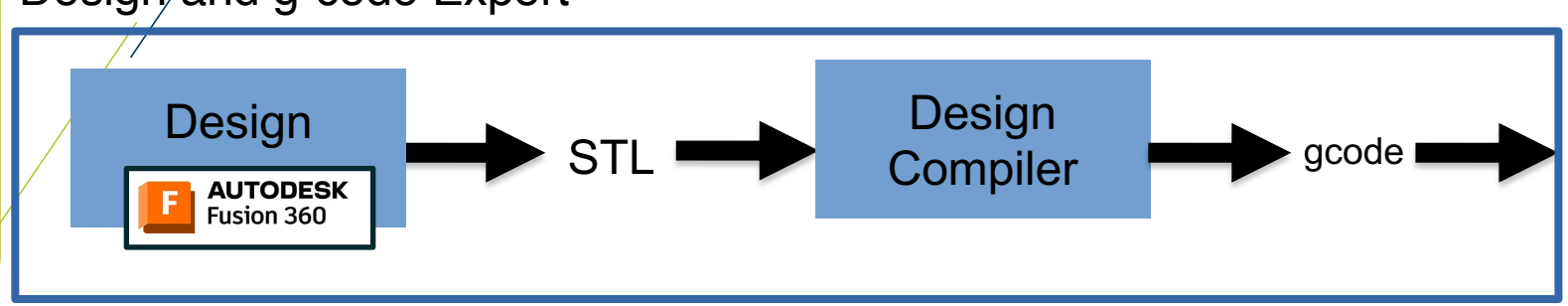
- An SBOM captures a subset of the STAMP information, focusing more on software linkage, **while STAMP provides additional process- or product-based linkage**

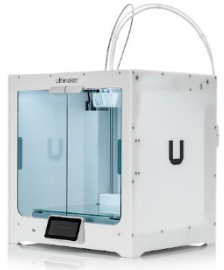


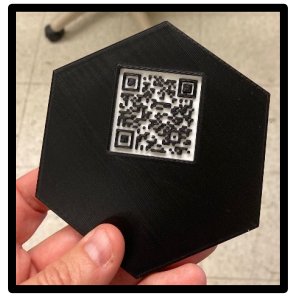
CYMANII

STAMPs in CPP Manufacturing Environment

Design and g-code Export







SOFTWARE (STAMP)

v.2.0.16985
Aug. 24, 2023

SBOM

NAME	VERSION	TYPE
AcApp Module	24.2.53.DM.11	dotnet
AcDimRes DLL	24.0.36.D.680	dotnet
AcFdBlockEdit Resources	24.0.49.DM.184	dotnet
AutoCAD Mechanical Common	24.0.52.0.0	dotnet
AutoCAD Mechanical Common	27.0.52.F.21	dotnet
AutoCAD component	24.0.49.DM.184	dotnet
AutoCAD component	24.1.38.DM.2	dotnet
AutoCAD component	24.1.39.0.0	dotnet
AutoCAD component	24.2.0.0.0	dotnet
AutoCAD component	24.2.53.DM.11	dotnet
Autodesk Hardcopy component	16.2.53.DM.11	dotnet
Autodesk component	24.2.0.0.0	dotnet
Heidi® DWF Driver Resource DLL	16.2.0.0.0	dotnet
Scene Resources	24.2.53.DM.11	dotnet
Visual LISP resource DLL	24.0.0.0.0	dotnet
acdb24res.dll	24.2.0.0.0	dotnet
node	14.18.3	binary
setuptools	60.10.0	python

SOFTWARE (STAMP)

v.5.5.0
Oct 23, 2023

SBOM

NAME	VERSION	TYPE
Automat	20.2.0	python
EnricoMi/publish-unit-test-result-action	v1	github-action
PyQt6	6.4.2	python
PyQt6-NetworkAuth	6.4.0	python
PyQt6-NetworkAuth-Qt6	6.4.2	python
PyQt6-Qt6	6.4.2	python
PyQt6-sip	13.4.1	python
SecretStorage	3.3.3	python
Twisted	21.2.0	python
actions-ecosystem/action-add-labels	v1	github-action
actions/cache	v3	github-action
...	...	...

SOFTWARE (STAMP)

v.8.2.0
May15, 2023

SBOM

NAME	VERSION	TYPE
Click	7.0	python
Flask	1.0.2	python
Flask-SQLAlchemy	2.1	python
Flask-Script	2.0.6	python
Jinja2	2.11.3	python
MarkupSafe	1.1.0	python
Pillow	5.4.1	python
PyGObject	3.30.4	python
SQLAlchemy	1.2.18	python
Werkzeug	0.14.1	python
actions/checkout	v1	github-action
aniso8601	4.1.0	python
certifi	2019.11.28	python
chardet	3.0.4	python
colorama	0.3.7	python
cycler	0.10.0	python
flask-restplus	0.10.1	python
idna	2.6	python
importlib-metadata	1.6.0	python
inotify	0.2.10	python
itsdangerous	0.24	python
...	...	...

PRODUCT ITEM

SOFTWARE (STAMP)

CYMANII

Supply Chain Transport and Tree Visualization with Software Passports

DESIGN

DESIGN FILE

GCODE FILE

PRODUCT

PRODUCT ITEM

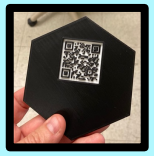
TIMESERIES

TRANSPORT

MACHINE

SOFTWARE (STAMP)

Supplier A

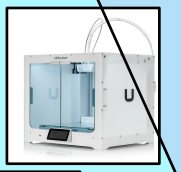


TRANSPORT

SOFTWARE

DataHub API

PRODUCT ITEM



TIMESERIES

SOFTWARE

Data Connector

MACHINE

SOFTWARE

3D Printer Firmware

PRODUCT

GCODE FILE

SOFTWARE

G-code compiler

DESIGN

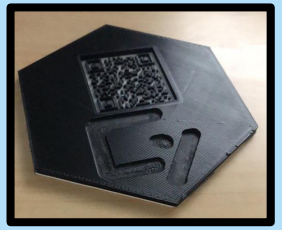
SOFTWARE

AUTODESK Fusion 360

DESIGN FILE

Supplier B

UUID: …8EC1B5



TIMESERIES

SOFTWARE

Data Connector

MACHINE

SOFTWARE

CNC Controller

PRODUCT

GCODE FILE

SOFTWARE

G-code compiler

DESIGN

SOFTWARE

AUTODESK Fusion 360

DESIGN FILE

CYMANII

Thank you!

gabriela.ciocarlie@cymanii.org
dongyan.xu@cymanii.org
ananth.grama@cymanii.org

CYMANII