SBOM FRAMING GROUP - REPORT OUT

CO-CHAIRS:

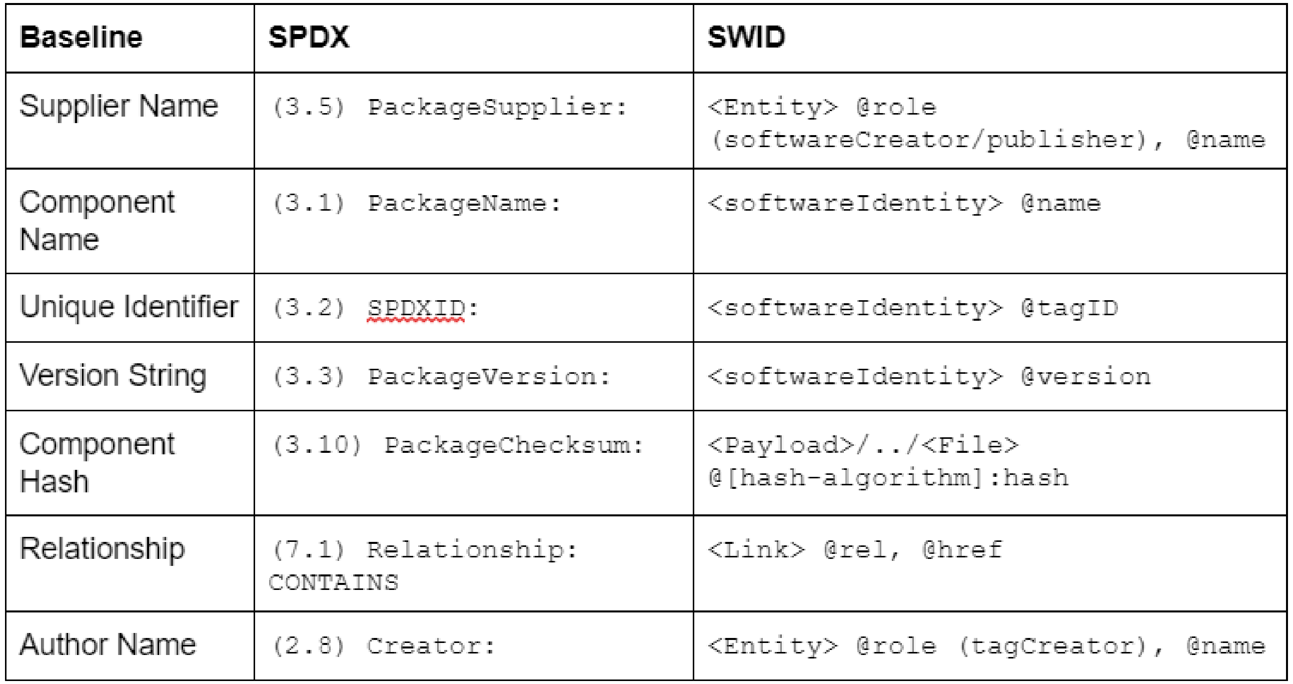
ART MANION & MICHELLE JUMP

Section 2: What is an SBOM?

A baseline SBOM was defined with a minimum set of information.

1. Author Name (author of SBOM)
2. Supplier Name (supplier of component)
3. Component Name
4. Version String
5. Component Hash (cryptographic hash of component)
6. Unique Identifier (to help identify components)
7. Relationship (inherent to design of SBOM. Default is “includes”)

Mapping SBOM Baseline to Existing Formats



Baseline	SPDX	SWID
Supplier Name	`(3.5) PackageSupplier:`	`<Entity> @role (softwareCreator/publisher), @name`
Component Name	`(3.1) PackageName:`	`<softwareIdentity> @name`
Unique Identifier	`(3.2) SPDXID:`	`<softwareIdentity> @tagID`
Version String	`(3.3) PackageVersion:`	`<softwareIdentity> @version`
Component Hash	`(3.10) PackageChecksum:`	`<Payload>/../<File> @[hash-algorithm]:hash`
Relationship	`(7.1) Relationship: CONTAINS`	`<Link> @rel, @href`
Author Name	`(2.8) Creator:`	`<Entity> @role (tagCreator), @name`

Table 1: Mapping baseline component information to existing formats

Section 2: Component Relationships

1. Unknown. This is the default. There is not yet any claim, knowledge, or assertion about upstream components. Immediate upstream components are not currently known and therefore not yet listed, or there may not be any upstream components. This default value implies the open-world ontological assumption.
2. Root. There are no immediate upstream relationships. As defined by the supplier, the component has no subcomponents.
3. Partial. There is at least one immediate upstream relationship and may or may not be others. Known relationships are listed.
4. Known. The complete set of immediate upstream relationships are known and listed.

Section 2: SBOM Examples

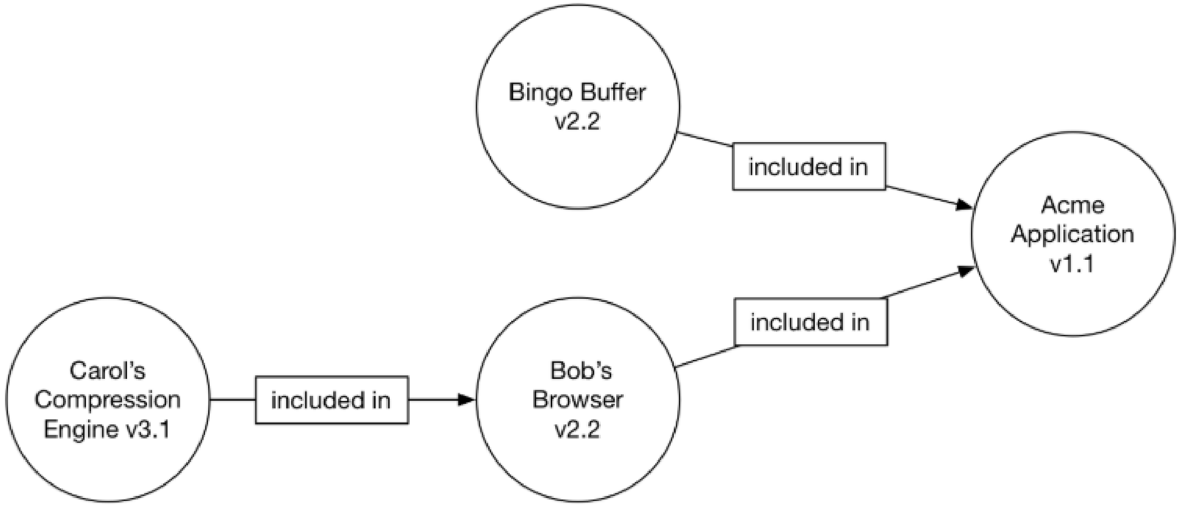
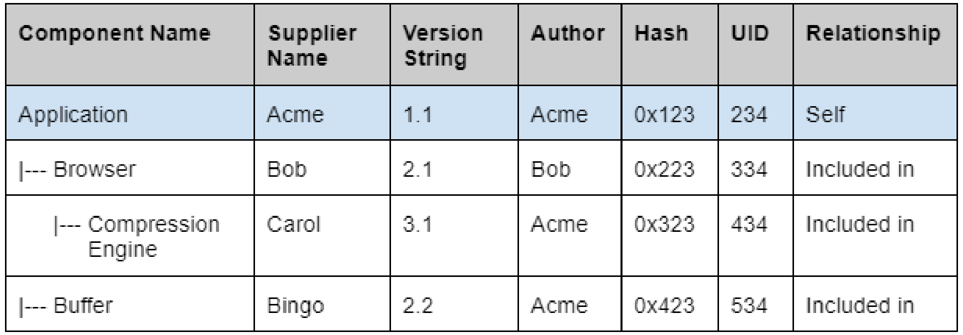


Figure 1: Conceptual SBOM tree



Component Name	Supplier Name	Version String	Author	Hash	UID	Relationship
Application	Acme	1.1	Acme	0x123	234	Self
|--- Browser	Bob	2.1	Bob	0x223	334	Included in
|--- Compression Engine	Carol	3.1	Acme	0x323	434	Included in
|--- Buffer	Bingo	2.2	Acme	0x423	534	Included in

Table 2: Conceptual SBOM table

Section 2: SBOM Examples

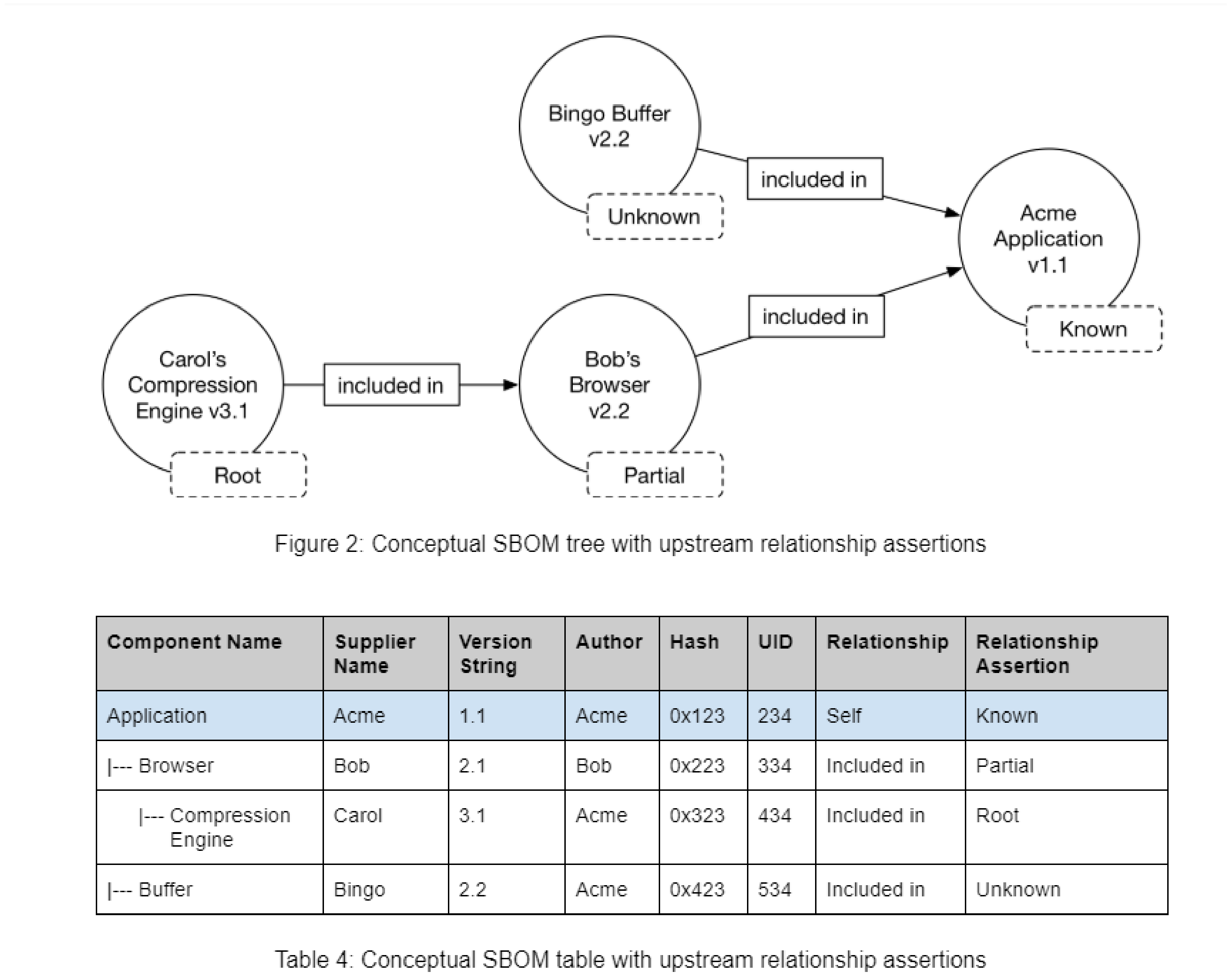

Bingo Buffer v2.2
Unknown
included in
Acme Application v1.1
Known
included in
Carol's Compression Engine v3.1
Root
included in
Bob's Browser v2.2
Partial

Figure 2: Conceptual SBOM tree with upstream relationship assertions

Component Name	Supplier Name	Version String	Author	Hash	UID	Relationship	Relationship Assertion
Application	Acme	1.1	Acme	0x123	234	Self	Known
|--- Browser	Bob	2.1	Bob	0x223	334	Included in	Partial
|--- Compression Engine	Carol	3.1	Acme	0x323	434	Included in	Root
|--- Buffer	Bingo	2.2	Acme	0x423	534	Included in	Unknown

Table 4: Conceptual SBOM table with upstream relationship assertions

Section 4: Processes

SBOM creation (how, when) and exchange

Network rules for participants: Define components, create and provide SBOMs

Roles and perspectives (produce, choose, operate), from Practices WG

Use cases and applications: Vulnerability management, IP and license management, high assurance

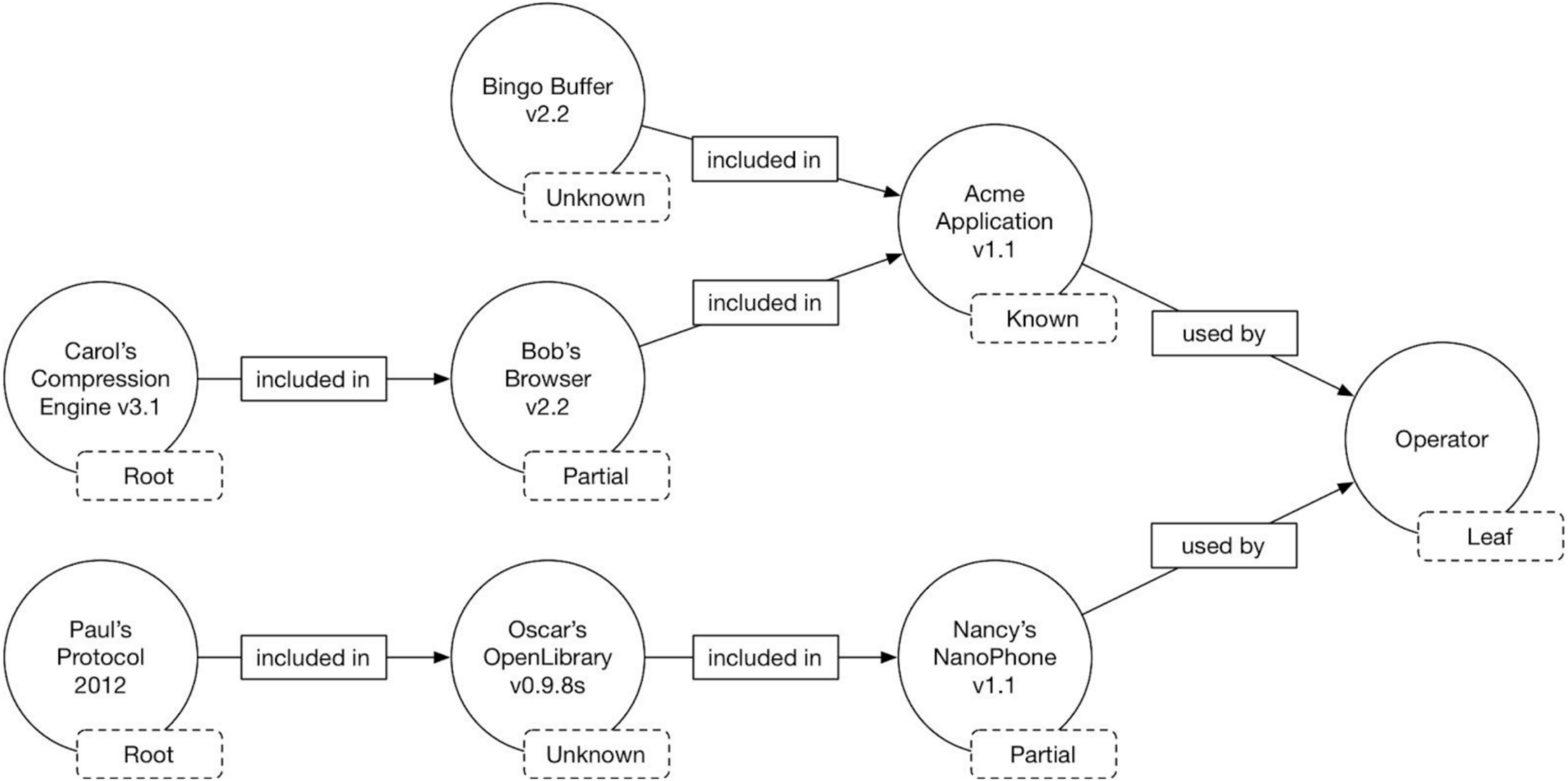


Figure 3: Operator tree with two supply chains

Next Steps

- List of ideas, topics
  - Rationale, effort, importance, owner
  - Collected from last meeting and Framing WG
  - https://tinyurl.com/yx3ufff7
- What elements of SBOM are needed to support capabilities (use cases, applications) and sectors?

Next Steps: Beyond Baseline

Capability	Sector			
	Health Care	ICS/OT	IT operation	Finance
Vulnerability				
IP/license				
High assurance				
Export (import) control				

What other capabilities? Sectors?

Supplier, consumer, operator, particularly in general purpose software

Next Steps: Beyond Baseline