SBoM Practices Working Group: 2/20/19 Summary

Practices working group's objective is anthropology: capture current practice/uses of Software Bill of Materials (SBoMs) or similar artifacts. Understanding the current state of practice across an ecosystem helps us chart the path between *where we are* and *where we want to be*.

TL;DR of the "Why":
Capture, share, and improve current pockets of SBoM value toward driving greater system value

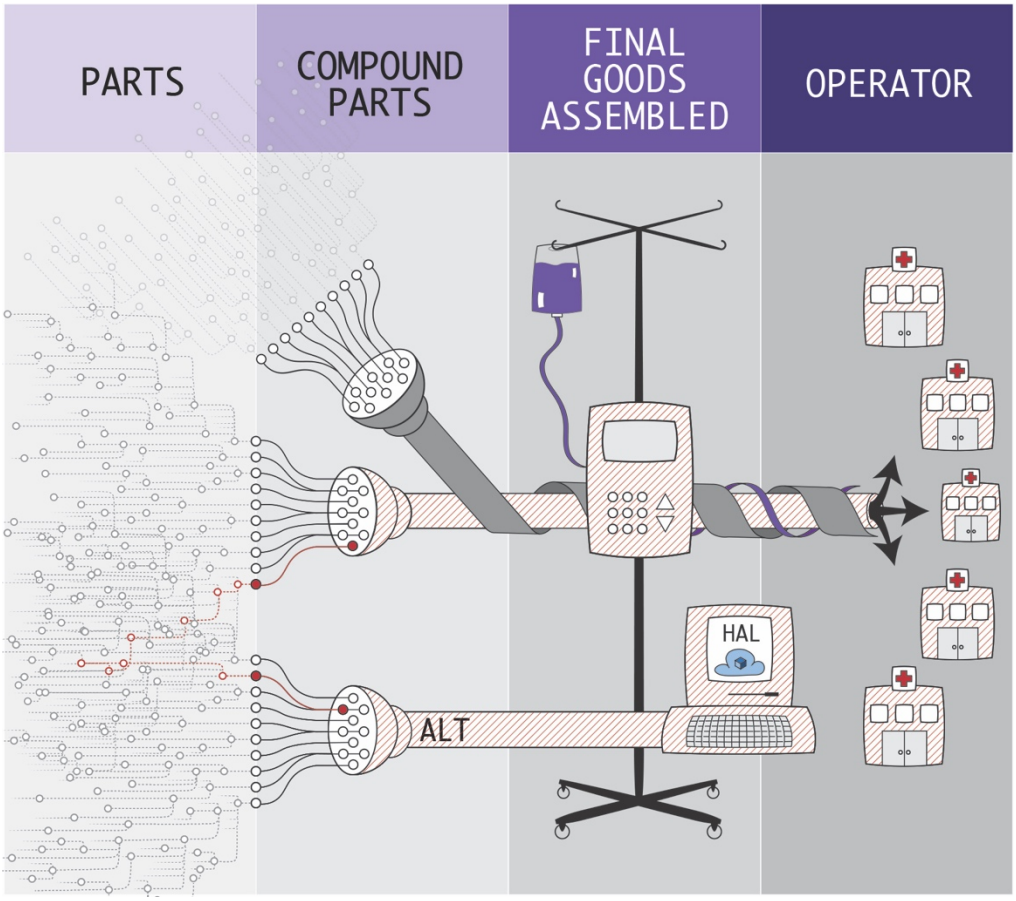
Modern Software has a supply chain – we just don't (universally) manage it like one. There are pockets of existing (and maturing) Bill of Materials practices *within* stakeholders. We're capturing and sharing use cases and state of practice across stakeholders and sectors with an eye toward near-term and long-term improvements. We also aim to capture and articulate opportunities for greater utility and value via a system view *across* these chains.

Opportunities:
Today's SBoMs enable *good* use cases. SBoMs contain similar information, but are not (yet) uniform, not pervasive, and could be even *better* with future enhancements. To unleash full potential, our belief is that NTIA overall could advance the state of practice 3 ways: Harmonization, Amplification, and Extension/Innovation.

1. *Harmonization (near term)* : more standard, machine readable presentation output of available data – from build tools and Software Composition Analysis (SCA) tools, and the like
2. *Amplification (near term)* : Once harmonized and machine readable/ingestible, greater adoption of SBoM production will unlock the value across stakeholders for greater percentages of their digital infrastructure.
3. *Extension/Innovation (longer term)* : There are types of information that are not currently captured, but which could both unlock net-new use cases, and enhance or accelerate existing use cases. The Standards Working Group is exploring those in greater detail.

Model

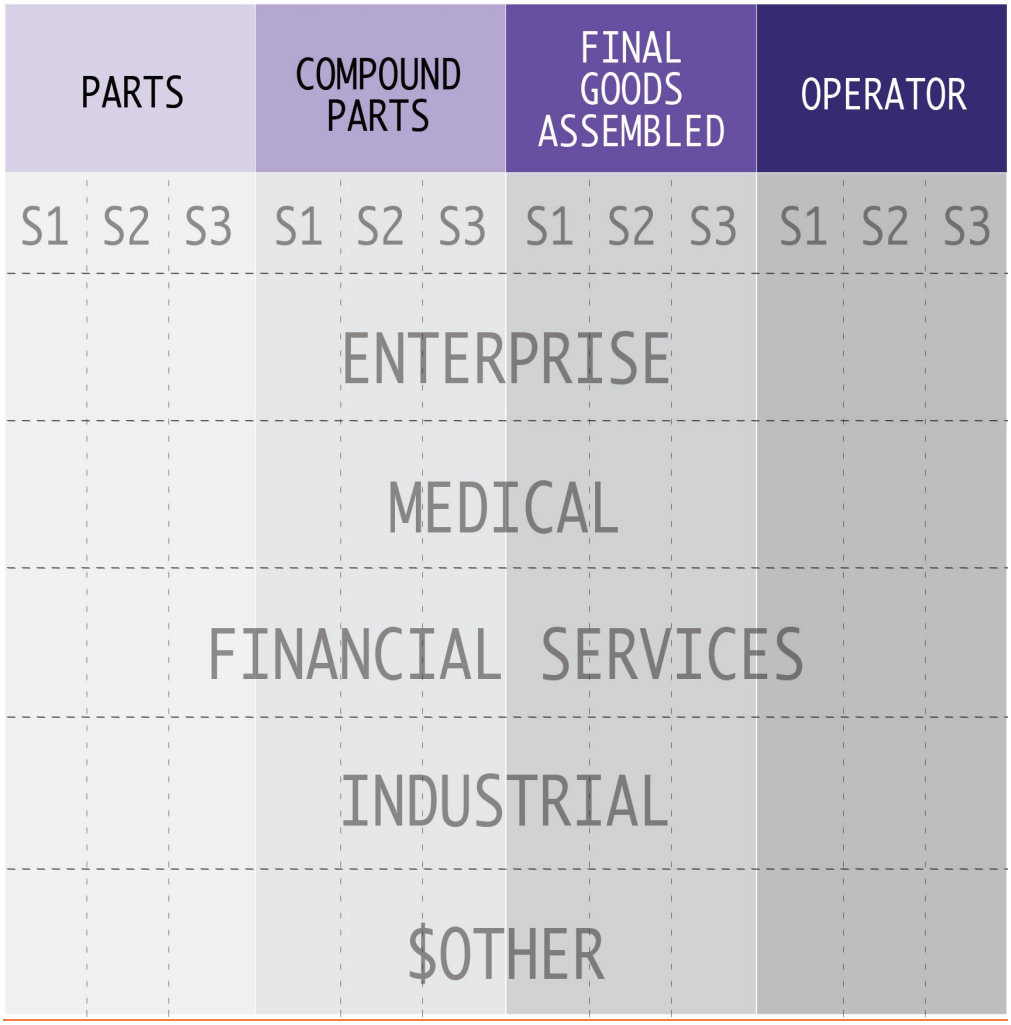
Our group developed a model to contextualize and capture the practices to ensure a consistent and organized approach that would be both broadly applicable and reflective of real-world scenarios. This model provides for input from different stakeholders in the SBoM process, including those that develop them and those that use them in different contexts. These "entities" (Parts, Compound Parts [1..n], Final Good Assemblers, Operators) and their relationship to each, other are shown in the following chart:



Entity Types:

- **Parts**: The smallest, identifiable, 'atomic' "Parts" are built. E.g. A small logging project.
- **Compound Parts**: These "Parts" are combined into "Compound Parts"/Projects. Sometime there are plural levels of "Compound Parts". E.g. A large project like Apache Struts 2 – or a commercial remote connectivity platform.
- **Final Good Assembled**: Ultimately these "Parts" are pulled together into a "Final Assembled Good". These are procured/acquired, by an entity who then "Operates" that good – for the life of the deployment.
- **Operator**: This SBoM Artifact enables various use cases in the Operational environment. E.g. a Bank who purchases, deploys, and operates software. E.g. a Hosptial who procures Bedside Infusion Pumps, deploys the latest version during a GoLive rollout to replace 1/3 of their current devices, and then safely operates them through their retirement – maintaining vigilance for ransomware attacks like SamSam – to answer: "Am I affected?" "Where am I affected?" To take steps to remediate or mitigate.

A Part may be an SBoM of one. Complex Compound Parts may pull together dozens to hundreds into their SBoM. A "Final Assembled Good" aggregates upstream Parts and dependencies, adds their own coding, and produces a final SBoM for that product version. Operators evaluate/consume these good with their SBoM.



While these Stakeholder Entity Type Columns are common across sectors, we know different industries are at different maturity levels and use varied vocabulary. As such, we've sought to interview and capture unique clusters against such a model above. Within an entity, we also encounter three repeating use case categories which we've marked as S1, S2, and S3 (inspired by Deming and his Toyota Supply Chains):

1. Supplier Selection (S1): An analysis of which parts suppliers with which to enter into a relationship
2. Supply Selection (S2): At Implementation/Deployment, which optimal (e.g. least vulnerable) version/batch of supply to use from those suppliers.
3. Supply Vigilance (S3): Ongoing monitoring for new-new vulnerabilities, attacks, recalls, issues – that may affect your operational dependence and/or compensating mitigations.

These S1/S2/S3 use cases categories can often involve different persona's within the entity – so we have captured those Personas within Entity interviews.

We chose the sectors listed as representations; they are not meant to be comprehensive and we expect other sectors to be able to use this model equally as well. Using this model, members of the working group interviewed people in different parts of the software supply chain.

Stakeholder/Entity Interviews (to date)

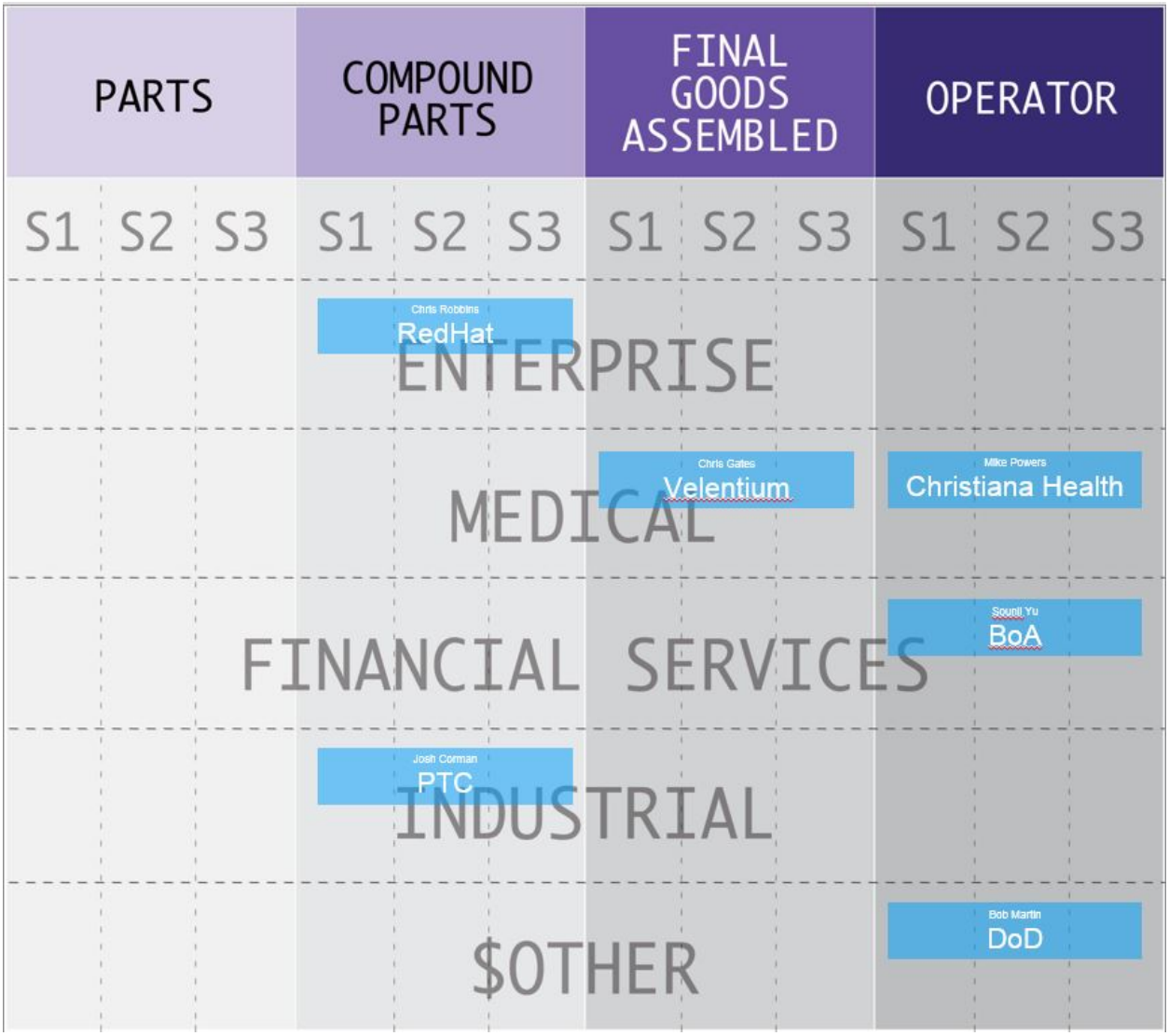
The working group's efforts to conduct interviews has resulted in the following information being collected to date. As shown, there have currently been no interviews conducted with any stakeholder that aligns to the "Parts" element of the model, and this remains an area of direct and immediate need.

Anyone interested in being interviewed for any of the stakeholder categories should sign at the below URL on the "Signup" tab.

https://docs.google.com/spreadsheets/d/1I5r8H4kyMVdyhyclYKB7tKrXIc1EIE0mmnU0x9hU7hk/edit?ts=5c3e3021#gid=2044634760

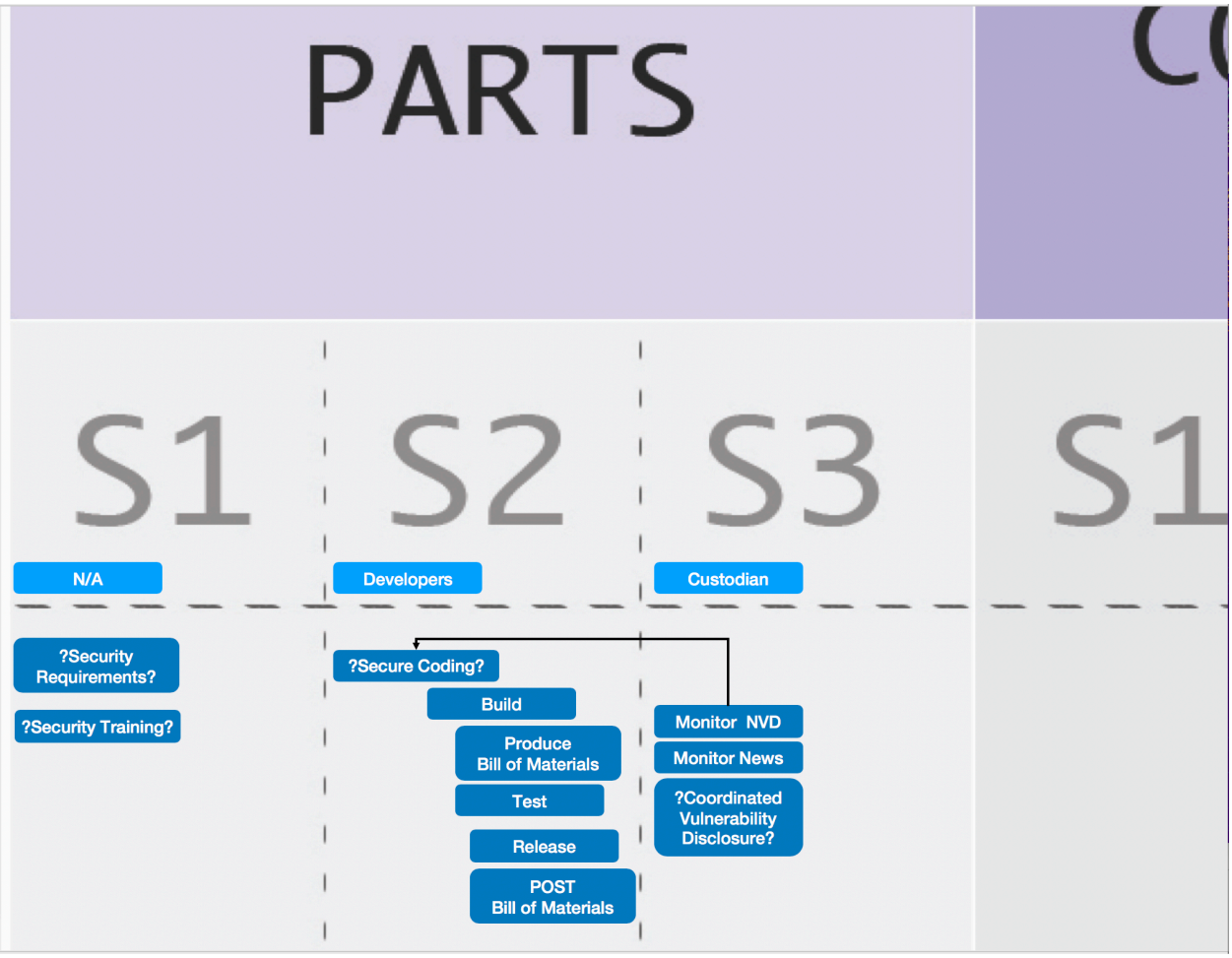
The full results of the interviews discussed below can also be found at this location.

The following chart shows how the model is used in each of the cases presented:



Parts (0 ‘formal’ interviews):

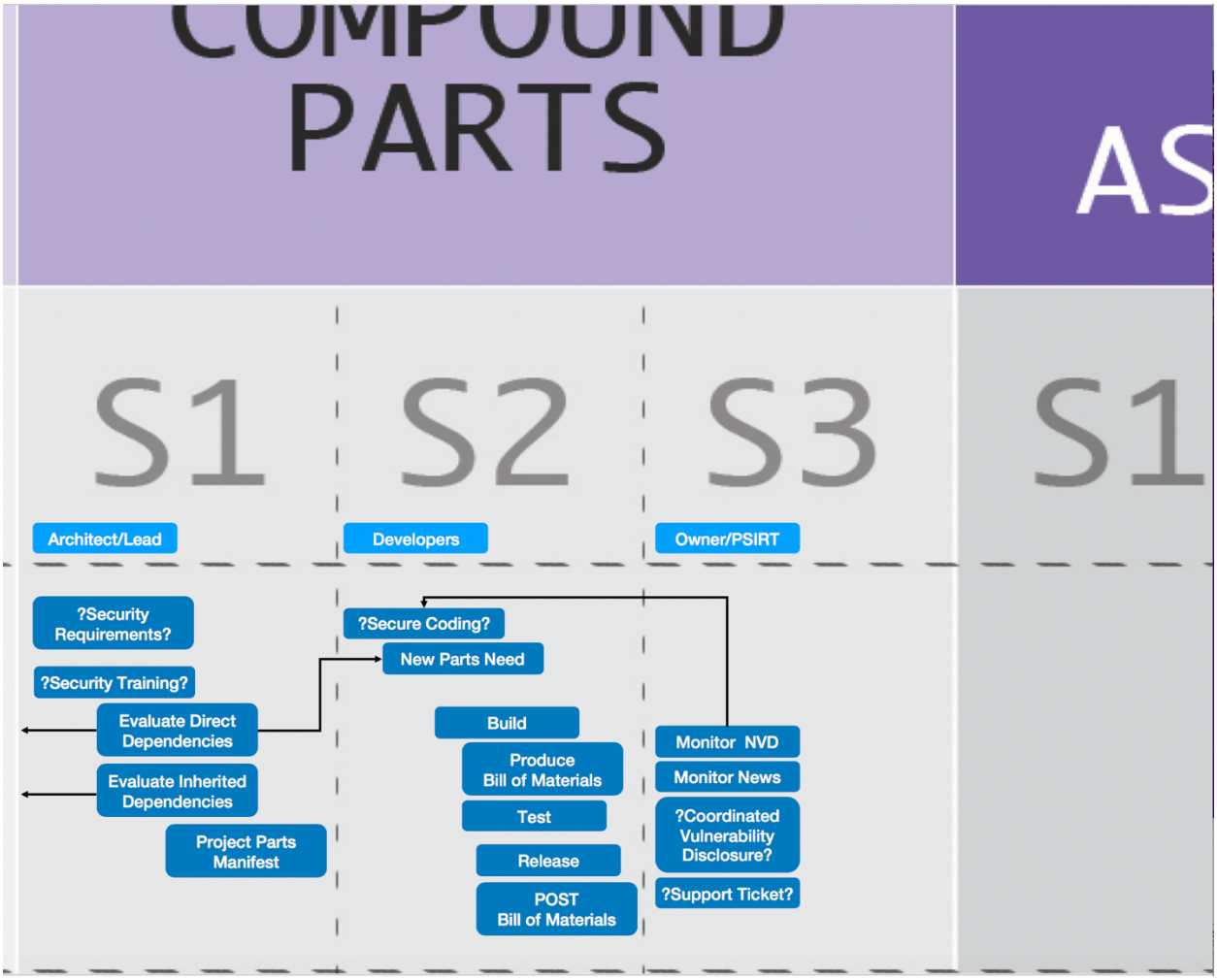
- As noted, the working group needs formal interview subjects for this entity/stakeholder role. Given that the “Parts” element resides at the earliest stage in the process, any analysis of SBoM practices is incomplete without information here that validates the model.



Compound parts (2 interviews):

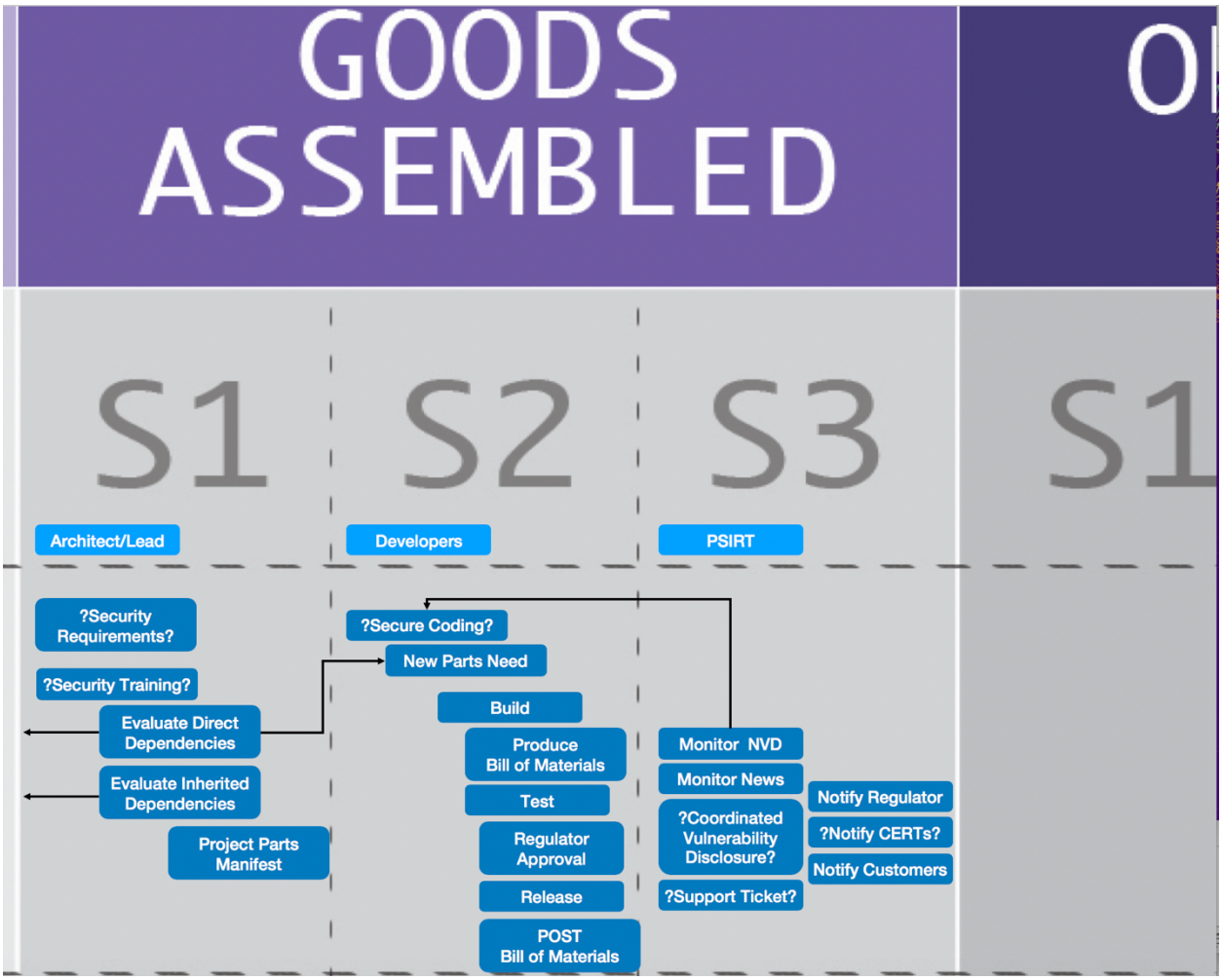
This 1st provider of compound parts is a “curator” of software that takes an active role in making sure it ships high-quality products, and its customers trust it to carefully vet components and apply good judgment. When considering a component for inclusion, this provider evaluates the quality of the code (including its upstream dependencies) and its maturity in handling security issues. In some cases, this provider embeds itself into the component’s development community and submits patches. In some cases it applies its own patches to the component before inclusion. All dependencies are described in a “manifest” that the provider (generally) writes and ships along with packaged components. Manifests are machine readable and maintained carefully as components change. The SBoM for the compound part is easily derived from components via the package manager.

The 2nd is a commercial, embedded IoT platform used in many final assembled goods. Supply Chain practices are currently used internally but are increasingly requested by customers to enable/meet their customers’/markets’ demands.



Final goods assembler (1 interview):
From the perspective of its customers, and regulatory bodies where applicable, the final goods assembler is responsible for the entire product, including everything preceding it in the supply chain. Documentation of components is a major theme: how did this component get into our product, what do we already know about the risks of including it, and what role does it play? This detailed information is captured in an SBoM artifact. A security architect updates this document at development milestones to ensure that the components are up to date.

Because it bears the ultimate responsibility for its product, the final goods assembler must remain vigilant after the product is released. Someone tasked with security watches public disclosure feeds and fields customers' reports, and when an issue arises with a component that's in the product (according to the SBoM), they assess its impact and decide how to respond.



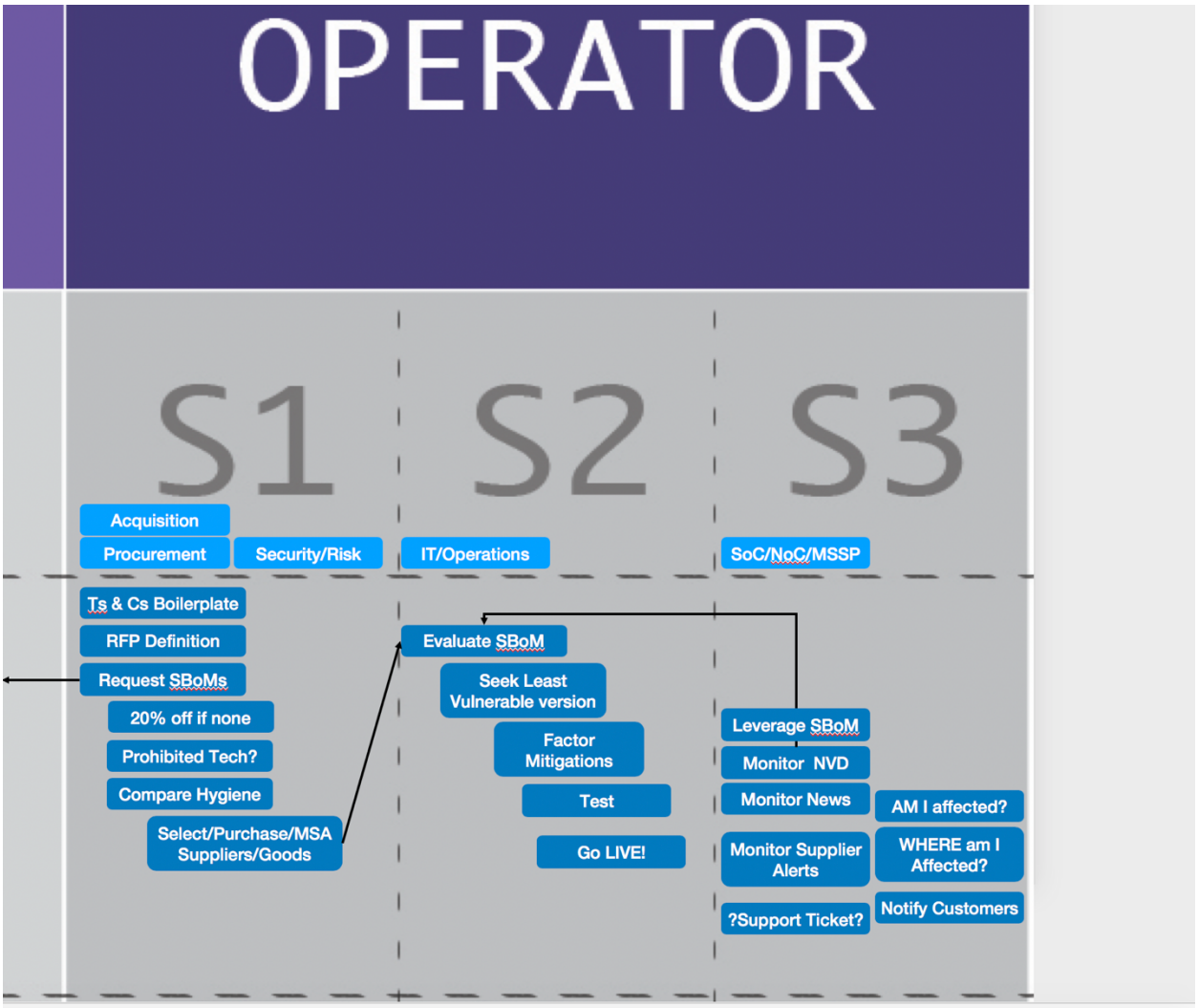
Operator (2 interviews):

Operators would like to have an SBoM with each product they obtain from final goods assemblers. A complete SBoM helps them in several ways:

- If the operator has lists of approved or prohibited components (e.g., outdated software libraries), they can quickly accept or reject products that match these lists, or request changes.
- The operator can compare components' SBoMs with vulnerability information and decide what to do with potentially affected devices. Sometimes they must accept risk and apply whatever mitigating controls they can, and sometimes they can work with suppliers on remediations.

Operators differ in their buying power and therefore how insistent they can be. Some operators can demand changes and then hold up procurement until problems are addressed. Some operators can write contract language that holds suppliers to an update plan. Sometimes suppliers decline to provide information if they don't believe the relationship is in jeopardy; in this case an operator may merely document as much as it can.

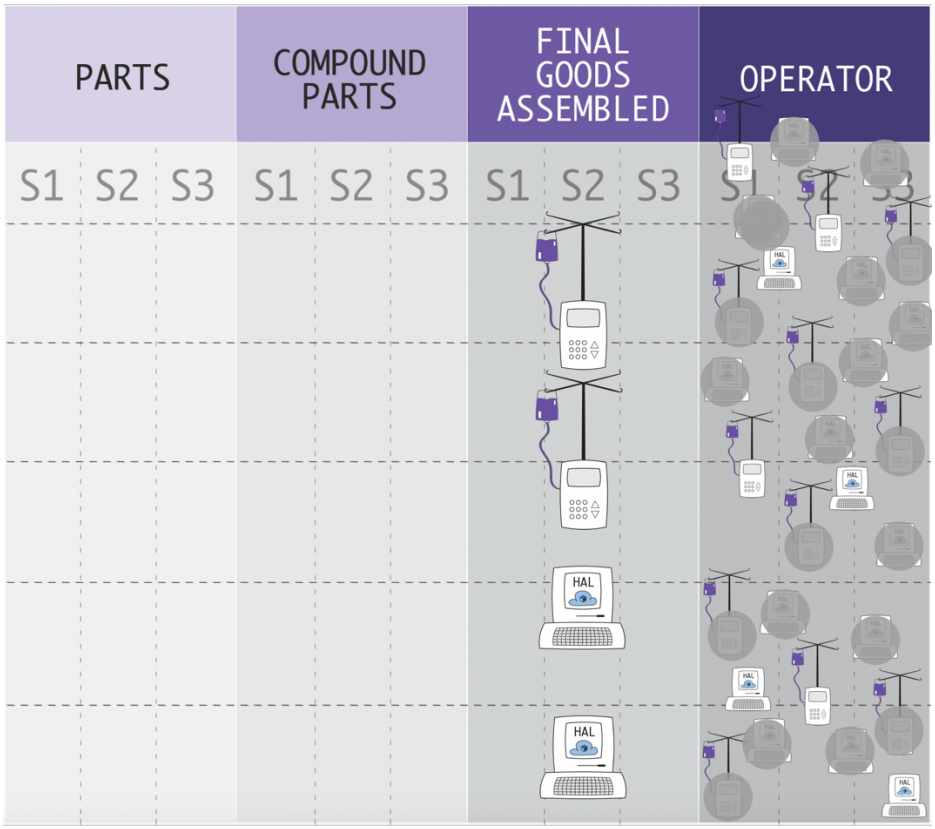
Operators expressed disappointment with the level of specificity they currently receive from suppliers. Sometimes they learn that a device contains a component but cannot finds its version number. Sometimes suppliers refuse to provide information at all or expect operators to find it themselves.



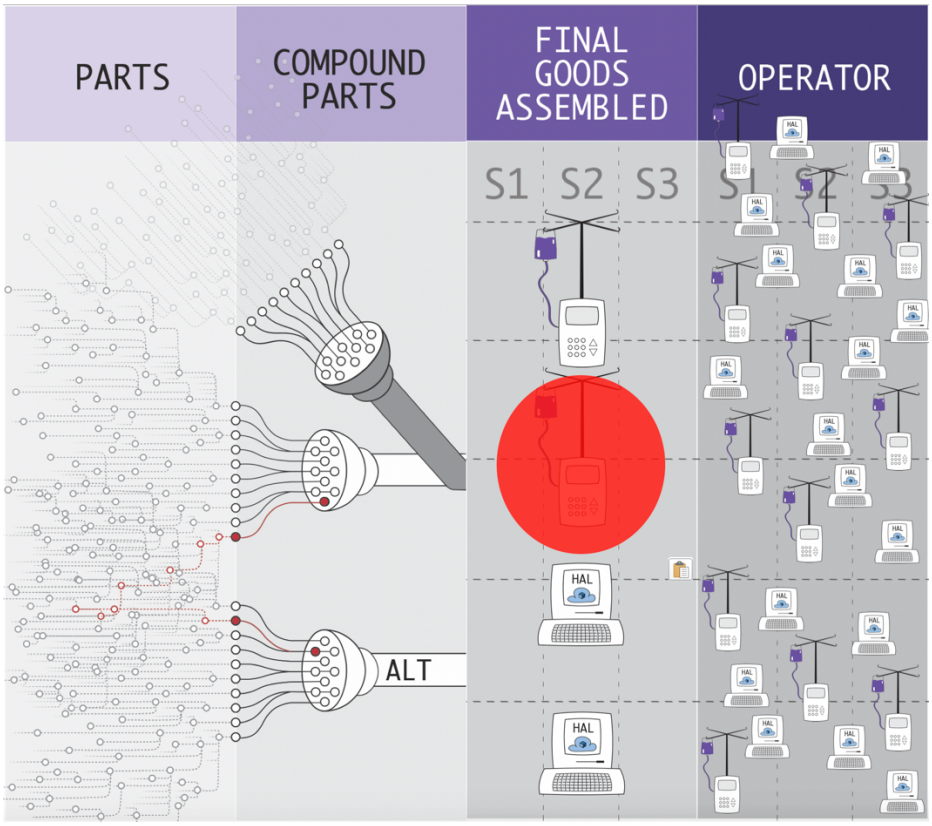
Zooming Out (Near-Term System Value)

Faster/Easier (Harmonization & Standardization):
As discussed, “harmonizing” outputs of existing SBoM could enable machine speed/readable production/ingest across the system and supply chain. This could decrease/eliminate human cutting and pasting, analysis, and delays – at each link in the chain.

Wider & Further (Amplification of Adoption):
If only 10% of your medical devices provide and SBoM, during an attack, your assistance in answering “Am I affected? Where?” is less valuable



When $Supplier goes Out of Business:
Disclosures which depend upon vendor notifications forget vendors or suppliers can go out of business… if an SBoM is provided with each version shipped, the transparency and information persists for all dependents… While updates may not be coming, mitigations can be taken.



Un-Obvious Population Assistance:
If a flaw is in a free open source remote administration project like BusyBox… could it affect anyone via medical devices? Which ones? The answer to that question is largely opaque… One could imagine how such transparency may enhance information sharing entities like ISACs and ISAOs… or those who can help with Recalls (e.g. FDA) - including for vendors long since out of business.

Next Steps

As previously noted, additional interview subjects are the most immediate need. The more input we can receive from real-world stakeholders, the more relevant, and hopefully accurate, the representation of practices will be.

Additionally, we welcome any feedback on the model itself, with a particular emphasis on circumstances where the model may not be able to accurately reflect a real-world situation, and therefore requiring updating.

Appendix:

Sample Interview: Chris Gates – Velentium – Final Goods Assembler – Medical Device

	B	C	D	E	F	G	H	I	J
1	**Security Architect: Medical Device Maker: Chris Gates, Velentium, Final goods assembler**			*Developer*					
2	*In design*			*In implementation*			*In Market*		
3	Supplier Selection	Supplier Selection	Supplier Selection	*SUPPLY Selection*	*SUPPLY Selection*	*SUPPLY Selection*	Supply Vigilance	Supply Vigilance	Supply Vigilance
4									
5	Inventory software we have licenses for or are very familiar with (we plan to use these)			In Development, unplanned needs (E.g. Compression) - Evaluate Alternatives for relative risks. SBoM++;	Aggregate 1..n SBoMs from upstream Suppliers, normalize them, and add our own DIRECT additive development	Vuln-QA-Just Before Pre-Market Submission	Monitor feeds of vulnerabilities to compare against selected parts		
6	Compare/Search for publicly known vulnerabilities in the software we're considering including. Vuln Assessment - snapshot in time. Risk/Reward						Re-evaluate any accepted vulnerabilities		
7	Document acceptance or rejection of software we're considering						Upgrade dependencies and reevaluate impact of known issues with selected parts		
8							Ship an upgrade		
9	Discovery/Enumeration/Inventory						Share information with an ISAO		
10	Vuln Assesment of KNOWN Inventory (Currently Blindspots - COULD be more conprehensive)				PARKING LOT: NOTE: If you took SOURCE... it's YOURS - just like proprietary // Some inteeresting back & forth				
11	Evaluate alternatives								
12									
13									
14	Enumerate direct components						Field customer reports	Confirm exploitability	Report to regulatory
15	Enumerate inherited components						Field CVD (researchers)	Assess impact	Disclose publicly
							Monitor CVE of		