Introducing the Software Bill of Materials Specification

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

07 October 2020
The Supply Chain for Software-Enabled Capabilities is Opaque
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 deliverables mid-November 2019

• More info or to join: afriedman@ntia.doc.gov
Roles and Benefits for SBOM Across the Supply Chain

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

Introduction
The Software Supply Chain
About this document: Goals and Methodology

Perspective: Produce Software
Reduce unplanned, unscheduled work
Reduce code bloat
Adequately understand dependencies within broader complex projects
Know and comply with the license obligations
Monitor components for vulnerabilities
End-of-life (EOL)
Make code easier to review
A blacklist of banned components
Provide an SBOM to a customer

Perspective: Choose Software
Identify potentially vulnerable components
A more targeted security analysis
Verify the sourcing
Compliance with policies
Aware of end-of-life components
Verify some claims
Understand the software’s integration
Pre-purchase and pre-installation planning
Market signal

Perspective: Operate Software
Organization can quickly evaluate whether it is using the component
Drive independent mitigations
Make more informed risk-based decisions
Alerts about potential end-of-lifes
Better support compliance and reporting requirements
Reduce costs through a more streamlined and efficient administration

Ecosystem, Network Effects, and Public Health Benefits of SBOM
Accelerated Vulnerability Management
Enabling Adoption of SBOMs

- Agriculture and Food
- Energy
- Transportation
- Chemical Industry
- Postal and Shipping

- Water
- Public Health
- Telecommunications
- Banking and Finance
- Key Assets

Source Code & Package Repositories

Integrated Development Environments (IDEs)

Cloud Tools

Frameworks

Build Choreography

Software Composition Analysis

Tools & Capabilities for Software

Product & Service Suppliers

End Users in Industry, Government, and Commerce

Sectors

Key Assets

Banking and Finance

Telecommunications

Public Health

Water

Tool-to-Tool SBOM Exchange Standard effort

© 2020 The MITRE Corporation. All rights reserved. Approved for Public Release; Distribution Unlimited. Case No: 20-01465-20
Ecosystem of SW Development, Integration, and Management Tools
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, Autotest, Bamboo, Bitrise, Buildkite, Buildroot, CircleCI, CMak, 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)

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, Kuatee, 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, Insignia, GrammaTech, SourceClear, Sonatype, Snyk, and WhiteSource
Table 1: IDEs in Use Survey

Table 2: Frameworks in Use Survey

Table 3: Cloud Tools in Use Survey

Table 4: Package Repositories in Use Survey

Table 5: Build & Build Choreography Capabilities in Use Survey

Table 6: Software Composition Analysis Capabilities in Use Survey
**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**

**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)*

**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**

- Company A
  - A1
  - A2
  - A3
  - **A6**
    - *Provenance* (Chain of Custody) of A6 includes Company C and Company B
  - **Process P1**
  - **Pedigree** (Lineage) of A6 includes the processes P1 and P2 and other artifacts used to create A6

- Company B
  - **A6**
  - **Process P2**

- Company C
  - **A6**

---

**Separating Pedigree & Provenance**

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

- **Provenance** (Chain of Custody)
  - Company C → **A6**
  - Company B → **A6**

- **Pedigree** (Lineage)
  - **A6** → **P2**
  - **A6** → **P1**
  - **A6** → **A4**
  - **A6** → **A5**
  - **A4** → **A1**
  - **A4** → **A2**
  - **A4** → **A3**
The Path to Code Provenance at Uber

April 17, 2019

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**.
**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 away
- Flexible, enforced policies for what code is allowed to land in production releases

**What are we protecting against?**
1. Lazy / shortchanging insider
2. Malicious insider
3. Engineer laptop controlled by malicious outsider
4. Build / deploy infrastructure attacked by malicious outsider
Usage Scenarios for Tool-to-Tool SBoM

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

Supply Chain Sequence Integrity
Usage Scenarios and Tool-to-Tool SBOM candidate elements

**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**

- Author of SBom
- SBom population method
- SBom Time-Stamp
- Supplier
- Components (sources, executables, patches)
- Version
- Notes
- Licenses
- Created Using
- Created By
- Item Hash/Signature

**Correlated Info**

© 2020 The MITRE Corporation. All rights reserved. Approved for Public Release; Distribution Unlimited. Case No: 20-01465-20
Usage Scenarios and Tool-to-Tool SBOM candidate elements

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 software delivering service)
9. Supply Chain Sequence Integrity

SBoM elements

- Author of SBOM
- SBOM population method
- SBOM Time-Stmp
- Supplier
- Components (sources, executables, patches)
- Version
- Notes
- Licenses
- Created Using
- Created By
- Item Hash/Signature

Correlated Info

None
Usage Scenarios and Tool-to-Tool SBOM candidate elements

**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**

- Author of SBOM
- SBoM population method
- SBoM Time-Stamp
- Supplier
- Components (sources, executables, patches)
- Version
- Notes
- Licenses
- Created Using
- Created By
- Item Hash/Signature

**Correlated Info**

None
Usage Scenarios and Tool-to-Tool SBOM candidate elements

**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

- Author of SBOM
- SBOM population method
- SBOM Time-Stamp
- Supplier
- Components (sources, executables, patches)
- Version
- Notes
- Licenses
- Created Using
- Created By
- Item Hash/Signature

None
Usage Scenarios and Tool-to-Tool SBOM candidate elements

**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**

- Author of SBOM
- SBOM population method
- SBOM Time-Stamp
- Supplier
- Components
  - (sources, executables, patches)
- Version
- Notes
  - Licenses
  - Created Using
  - Created By
- Item Hash/Signature

**Correlated Info**

None
Usage Scenarios and Tool-to-Tool SBOM candidate elements

**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**

- Author of SBOM
- SBOM population method
- SBOM Time-Stamp
- Supplier
- Components (sources, executables, patches)
- Version
- Notes
- Licenses
- Created Using
- Created By
- Item Hash/Signature

**Correlated Info**

None
Usage Scenarios and Tool-to-Tool SBOM candidate elements

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

- Author of SBOM
- SBOM population method
- SBOM Time-Stamp
- Supplier
- Components (sources, executables, patches)
- Version
- Notes
- Licenses
- Created Using
- Created By
- Item Hash/Signature

Correlated Info

Vulnerability Knowledge Bases
Vulnerability Management Systems

© 2020 The MITRE Corporation. All rights reserved. Approved for Public Release; Distribution Unlimited. Case No: 20-01465-20
Usage Scenarios and Tool-to-Tool SBOM candidate elements

**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**

- Author of SBOM
- SBOM population method
- SBOM Time-Stamp
- Supplier
- Components (sources, executables, patches)
- Version
- Notes
- Licenses
- Created Using
- Created By
- Item Hash/Signature

**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

© 2020 The MITRE Corporation. All rights reserved. Approved for Public Release; Distribution Unlimited. Case No: 20-01465-20
**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**

- Author of SBoM
- SBoM population method
- SBoM Time-Stamp
- Supplier
- **Components**
  (sources, executables, patches)
- Version
- Notes
- Licenses
- Created Using
- Created By
- Item Hash/Signature

**Correlated Info**

Logging SBOMs of Services Used
Usage Scenarios and Tool-to-Tool SBOM candidate elements

### 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

- **Author of SBOM**
- **SBOM population method**
- **SBOM Time-Stamp**
- **Supplier**
- **Components** (sources, executables, patches)
- **Version**
- **Notes**
- **Licenses**
- **Created Using**
- **Created By**
- **Item Hash/Signature**

### Correlated Info

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

© 2020 The MITRE Corporation. All rights reserved. Approved for Public Release; Distribution Unlimited. Case No: 20-01465-20
Launched 24 Sep 2019

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

PARTICIPANTS
- Ken Modeste (UL)
- Philippe-Emmanuel Douziech (CAST)
- Santiago Torres-Arias (in-toto/INU)
- Diannah Gooden (BlackBerry)
- William Cox (Black Duck by Synopsys)
- Steve Lasker (Microsoft - Artifact Storage, Open Container Initiative)
- Brian Russell (Google - CD Foundation)
- Nitesh Bakliwal (Microsoft - Windows)
- Kate Stewart (Linux Foundation - SPDX)
- William Bortholomew (GitHub)
- David Edelson (IBM - GCC)
- Jason Shaver (Microsoft - Developer Division)
- JC Herz (Ion Channel)
- Adam Baldwin (npm, GitHub)
- Thomas Steenbergen (HERE - OSS Review Toolkit)
- Soen Barnum (MITRE)
- Charles Schmidt (MITRE)
- Kriti Pandit (Microsoft - Windows)
- Michael Richardson (Sandeman Software Works Inc.)
- Snihur Poduri (Microsoft - Supply Chains)
- Vijay Chari (Radiometer IDC - Medical Devices)
- Allan Friedman (NTIA)
- Gerald Heidenreich (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)
- Patrick Dwyer (OWASP, CycloneDX)
- Fred Blaise (CloudBlees - Jenkins)
- Mark Galpin (JFrog)
- Gary O’Neal (Source Auditor - SPDX)
- Anna Debenham (Snyk)
- Duncan Sparrell (SFractal Consulting)
- Philippe Ombredanne (nexedi - ScanCode)
- Solomon Rubin (FOSSA)
- Adrian Diglio (Microsoft - Engineering Systems)
- Steve Winslow (Linux Foundation)
- Henk Birkholz (Fraunhofer SIT - IETF RATS and RIM)
- Paul Anderson (GrammoTech)
Table of Contents

Table of Contents 2
1 Problem Statement 4
  1.1 Goals 4
2 What is an SBOM? 6
  2.1 SBOM Elements 7
  2.2 Baseline Component Information 7
    2.2.1 Author Name 7
    2.2.2 Supplier Name 7
    2.2.3 Component Name 8
    2.2.4 Version String 8
    2.2.5 Component Hash 8
    2.2.6 Unique Identifier 8
    2.2.7 Relationship 8
3 Mitigation / Evaluation / 8
<table>
<thead>
<tr>
<th>Name</th>
<th>Owner</th>
<th>Last modified</th>
<th>File size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subgroups</td>
<td>William Bartholomew</td>
<td>May 8, 2020</td>
<td></td>
</tr>
<tr>
<td>Misc</td>
<td>Kay Williams</td>
<td>Feb 3, 2020</td>
<td></td>
</tr>
<tr>
<td>Face to Face Meetings</td>
<td>Kay Williams</td>
<td>Dec 6, 2019</td>
<td></td>
</tr>
<tr>
<td>Modeling</td>
<td>Philippe Emmanuel</td>
<td>Nov 25, 2019</td>
<td></td>
</tr>
<tr>
<td>Meeting Recordings</td>
<td>William Bartholomew</td>
<td>Oct 30, 2019</td>
<td></td>
</tr>
<tr>
<td>User Scenarios</td>
<td>William Bartholomew</td>
<td>Oct 21, 2019</td>
<td></td>
</tr>
<tr>
<td>Specification Drafts</td>
<td>me</td>
<td>Oct 15, 2019</td>
<td></td>
</tr>
<tr>
<td>Whitepaper Versions</td>
<td>me</td>
<td>Sep 29, 2019</td>
<td></td>
</tr>
<tr>
<td>Weekly Meeting Agenda</td>
<td>Kay Williams</td>
<td>Sep 16, 2020</td>
<td></td>
</tr>
<tr>
<td>Working Group Members</td>
<td>Kay Williams</td>
<td>Sep 15, 2020</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Owner</td>
<td>Last modified</td>
<td>File size</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>----------------</td>
<td>------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>Relationship</td>
<td>William Bartholomew</td>
<td>2:48 PM H.B.</td>
<td></td>
</tr>
<tr>
<td>redis-0.3.0.obom.json</td>
<td>William Bartholomew</td>
<td>2:44 PM William Bartho</td>
<td>12 KB</td>
</tr>
<tr>
<td>3T-SBOM Core Model - LATEST.png</td>
<td>William Bartholomew</td>
<td>1:03 PM William Bartho</td>
<td>321 KB</td>
</tr>
<tr>
<td>3T-SBOM Core Model - LATEST.drawio</td>
<td>William Bartholomew</td>
<td>1:03 PM William Bartho</td>
<td>8 KB</td>
</tr>
<tr>
<td>API Artifacts</td>
<td>Adrian Digilo</td>
<td>Sep 17, 2020</td>
<td></td>
</tr>
<tr>
<td>3T-SBOM / SPDX 3.0 Comparison</td>
<td>William Bartholomew</td>
<td>Sep 8, 2020</td>
<td></td>
</tr>
<tr>
<td>Core Specification Draft</td>
<td>William Bartholomew</td>
<td>Aug 17, 2020</td>
<td></td>
</tr>
<tr>
<td>Artifact Type</td>
<td>William Bartholomew</td>
<td>Aug 4, 2020</td>
<td></td>
</tr>
<tr>
<td>SBOM Working Group - Core</td>
<td>William Bartholomew</td>
<td>Jul 29, 2020</td>
<td></td>
</tr>
<tr>
<td>Unique Identifier</td>
<td>William Bartholomew</td>
<td>Jul 7, 2020</td>
<td></td>
</tr>
<tr>
<td>Component Hash</td>
<td>William Bartholomew</td>
<td>Jun 17, 2020</td>
<td></td>
</tr>
<tr>
<td>Component Name</td>
<td>William Bartholomew</td>
<td>Jun 3, 2020</td>
<td></td>
</tr>
<tr>
<td>Component Supplier</td>
<td>William Bartholomew</td>
<td>May 20, 2020</td>
<td></td>
</tr>
<tr>
<td>Document Author</td>
<td>William Bartholomew</td>
<td>May 5, 2020</td>
<td></td>
</tr>
</tbody>
</table>
Lowering Adoption Hurdles for SBOMs and more

SBOMs

Product & Service Suppliers

- End Users in Industry, Government, and Commerce

- Sectors
  - Agriculture and Food
  - Water
  - Energy
  - Public Health
  - Transportation
  - Telecommunications
  - Chemical Industry
  - Banking and Finance
  - Postal and Shipping
  - Key Assets

- Medical Devices
- Merchandise
- Automobiles
- Trains
- Vessels/Boats
- Building Mgmt Sys
- Software

- Asset/ Capabilities

Tools & Capabilities for Software

INTEGRATED DEVELOPMENT ENVIRONMENTS (IDEs)

SOFTWARE COMPOSITION ANALYSIS

INVENTORY MANAGEMENT SYSTEMS

PRODUCT LIFECYCLE MANAGEMENT (PLM) TOOLS

Source Code & Package Repositories

Frameworks

Cloud Tools

License Mgmt

Tools & Capabilities for Hardware

Core BOM

License Profile
Pedigree Profile
Provenance Profile
SW BOM Profile
HW BOM Profile
X Profile
Y Profile
Z Profile

MITRE

Tool-to-Tool SBOM Exchange Standard (3T-SBOM-ES) effort

© 2020 The MITRE Corporation. All rights reserved. Approved for Public Release; Distribution Unlimited. Case No: 20-01465-20
Whitepaper → CISQ → OMG RFC → ISO Std

- Socialize at March 2019 OMG meeting
- Draft SBoM as a Whitepaper in 3-day CISQ SBoM working session at September 2019 OMG meeting
- Prototype draft 3T-SBOM format in tool ecosystem, revise and draft RFC based on prototype results
- Co-submit draft RFC to OMG at Dec 2020 or Mar 2021 meeting
- Mar or Jun 2021 OMG meeting – charter FTF
- Jun or Sep 2021 OMG meeting – approve as OMG Standard
- Sep or Dec 2021 Fast Track to ISO
Questions?