Organizations of all sizes these days need to concern themselves with managing their software supply chain. The fact is that many contemporary applications consume third-party software / services in the way of application programming interfaces (APIs) and / or via Web services (e.g., service-oriented architecture: SOA). This reliance on external parties within an enterprise's software ecosystem will introduce additional risks to the organization.

To manage this growing risk, organizations must treat these APIs and / or Web services as part of their supply chain. While logistical professionals frequently manage traditional supply chains, this new paradigm crosses multiple disciplines (i.e., logistics, procurement, software development, legal, risk management, development operations [DevOps], and / or information security) and therefore needs to be treated differently. Specific concerns that organizations should focus on, include: application / software security, asset management, and / or intellectual property (IP) usage rights.
While the software supply chain involves multiple concerns, it should be the developers, DevOps, and software security professionals who are the most involved. These individuals should first implement a software supply chain management (S-SCM) program. To accomplish this, an organization needs to do the following:

- Catalog the APIs and Web services used across the enterprise.
- Run a security review / audit on said software.
- Cross-reference the use of said software versus its IP licensing allowance.
- Plan end of life (EOL) with DevOps for proper care and maintenance.
- Update contracts and / or service-level agreements (SLAs) with S-SCM provisions.

The thought points mentioned above require some heavy lifting. However, this program can be brought to life within six (6) months or less with the assistance of executive sponsorship, adequate system documentation, cataloging software, security assessment tools, and with an accessible subject matter expert (SME) on IP usage / licensing. It should be noted that both project management office (PMO) and disaster recovery (DR) documentation may help with the cataloging, and that some software packages (e.g., OpenLogic Exchange) already exist for identifying open-source software assets. Also, note that some companies (e.g., eBay's VeRO) have embraced a community mentality for vetting / cataloging IP usage within their enterprise.

Beyond cataloging the usage of third-party software, an organization must assess this software from a security and IP usage standpoint as well. While many organizations use static and dynamic application security scanners (SAST / DAST) to review their custom software, third-party libraries are often overlooked or omitted from these scans. To remediate this oversight the S-SCM program should explicitly review the security and allowed IP usage of these APIs / services on a quarterly basis. Beyond reviewing the allowable IP usage and security of these libraries, an organization should plan for the EOL of the software as well. As has been observed with Adobe releasing the care and maintenance of its Flex technology to Apache, companies and organizations will not continue to support their software artifacts forever. Therefore, it is incumbent upon the organization that the S-SCM program works in conjunction with DevOps to plan for the eventual EOL of these software packages. Finally, as many organizations have third-parties (e.g., independent software vendors, ISVs) build custom or modified applications for them, it is essential that the Procurement function builds S-SCM into the contracts and SLAs used for these engagements. Such measures reduce risk and protect the enterprise's software ecosystem.

As enterprises rely more on third-parties via open-source APIs for software artifacts it is essential that

these organizations build an S-SCM program. By incorporating S-SCM thought leadership into SLAs, as well as planning for the eventual EOL of these packages, the company will confidently and successfully engage in proper risk management. However, this effort is reliant upon a catalog of used APIs and that requires effort with or without the use of a cataloging tool.