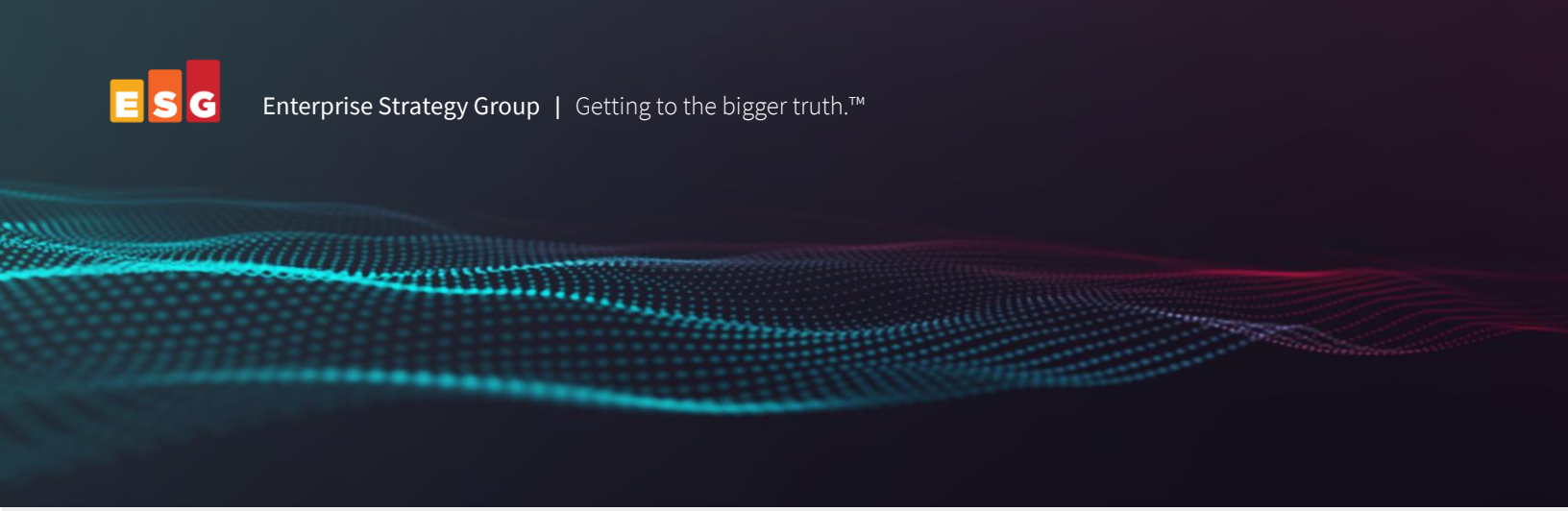


ESG WHITE PAPER

Aiming for Cloud-native, Containerized Workloads? Without a Service Mesh, You're Heading for Scaling Problems.

Using Enterprise Versions of the Istio Open Source Service Mesh to Bridge the Gap Between On-premises and Cloud-native Applications and Solve Scaling, Security, and Resilience Problems

By Paul Nashawaty, ESG Senior Analyst

September 2021

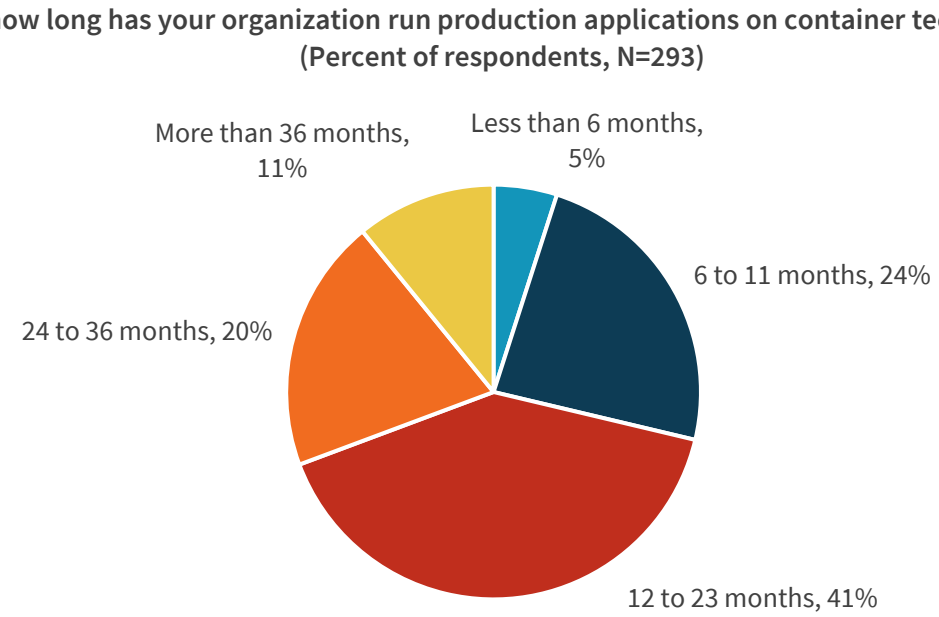
This ESG White Paper was commissioned by Solo.io and is distributed under license from ESG.

Contents

Cloud-driven Containerization Creates Major Scaling Problems

As businesses look to drive operational efficiencies alongside workload portability and security, they are increasingly moving both code development and production applications to the cloud. As part of this process, they are reworking existing workloads and creating new applications as collections of microservices. These non-monolithic applications are also known as cloud-native applications, and they can be distributed across multiple execution venues—both on-premises and in public clouds.

Figure 1. Significant Movement of Production Applications to Containers over the Last Two Years



Source: Enterprise Strategy Group

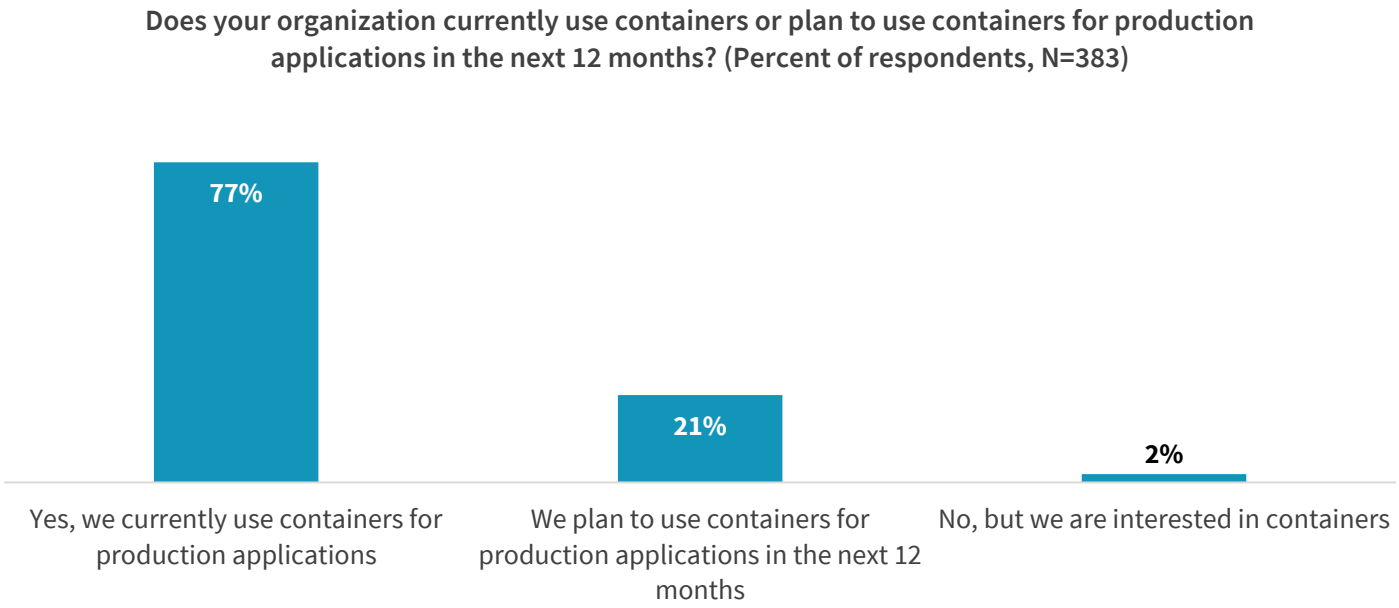
Service Mesh Increase in Popularity

Service meshes are becoming more mainstream and widely adopted across the industry.

Containerization of code (and Kubernetes) are the major enablers of this change from monolithic to microservices-based software, which is happening quickly. In fact, in a recent ESG research survey, 41% of respondents stated that their organizations have run production applications on containers for the past 12 to 23 months.[1] This number represents the middle of the pack. Some organizations are moving more quickly and others more slowly. Overall, containerization is now approaching universal adoption, with 77% of organizations currently running production applications in containers, and another 21% are planning to do so within the next twelve months.[2]

[1] Source: ESG Research Report: *The Maturation of Cloud-native Security*, May 2021.
[2] Ibid.

Figure 2. Containers Are Underpinning Microservices-based Cloud-native Applications



Source: Enterprise Strategy Group

Major Challenges

As the delivery of cloud-native applications and services grows, the communication of data between application services must also evolve. Modern organizations delivering cloud-based applications are often faced with challenges when it comes to at-scale deployments of microservices across multiple on-premises server clusters and public clouds. This is especially true when the deployments are diverse and comprised of a mix of containers running on bare-metal servers; containers running within virtual machines or cloud instances managed by Kubernetes; or applications that have been built using multiple, heterogenous development tools—as they often are. One of the problems is the need to manage new data traffic patterns between microservice containers as application demands grow, which requires sophisticated routing of requests to maintain overall application effectiveness. Another problem is the difficulty of ensuring security in such complex environments.

Service Meshes Address Distributed Application Problems

As applications transform into collections of distributed microservices, new patterns of data traffic between microservices will not work without sophisticated routing.

Addressing the Challenge with Service Meshes

Service mesh software provides a dedicated infrastructure layer that delivers scalable and secure communication in microservices architectures. A service mesh incorporates a sidecar proxy (such as Envoy) into the data control plane that allows microservices to effectively transfer information as needed. The data control plane also manages authentication, security, and network resiliency and provides data insights for applications. However, the value of service meshes extends far beyond infrastructure, as it also enables organizational lines of business to become more agile, innovative, and productive with the adoption of GitOps and CI/CD approaches to modern application development. By providing a standardized method of delivering applications, service meshes make IT organizations more cost-effective and reduce risks when deploying applications.

The Business Case for Service Meshes

In order to survive and maintain their competitive edge, businesses are looking to be agile and innovative while enabling faster deployments of applications and more flexible use of hybrid on-premises and public cloud infrastructure. Achieving this while also improving security and reliability can be a daunting challenge, even on a good day.

For DevOps teams, the business value offered by a service mesh is realized in the CI/CD pipeline. According to research completed by ESG, 46% of IT enterprise IT organizations have already integrated build, test, and testing tools into their DevOps CI/CD pipeline. These organizations are also exploring GitOps as a Kubernetes-native strategy. Many more organizations have also realized the importance of this integration and plan to integrate soon. When deploying microservices-based applications, a service mesh programmatically implements policies concerning routing, failover, and security. By allowing application developers to focus on the development of the applications themselves and separating out the logic of the service mesh from the application code, this provides a faster time to value for application delivery. It also boosts security by implementing security policies in-line, leveraging automated processes that are repeatable and consistent and can be applied to any application. Indeed, without an enterprise-grade service mesh, production applications and data actually may be at risk.

However, the benefits of service meshes go beyond application development, as they also include improved resilience and overall application performance and a solution to the scaling issues inherent in microservices-based architectures. These latter benefits are delivered via the same automated implementation of policies and configurations that result in standardized application deployments. Service meshes also provide visibility into networks that can speed diagnosis of connectivity, performance, and security issues. Enterprise versions of Istio—or versions of Istio that have been bolstered with third-party overlay software add features such the ability to split data traffic during testing or canary deployments of applications. Deliver on security and compliance requires solutions that comply with Federal Information Processing Standards (FIPS). For some, ARM architectures can cover specialized processing requirements of large organizations and edge startups alike.

Istio as Preferred Mesh

Istio is an open source service mesh that has emerged as the clearly preferred approach in the market and is currently the most comprehensive offering available. One of its most obvious strengths is that it is cloud- and Kubernetes-native or was architected at inception to work with Kubernetes, unlike older offerings which have been inadequately retrofit to modern designs and requirements. Kubernetes orchestration of containers and microservices decouples the application from the infrastructure, delivering portability of microservices for flexibility. Organizations using Kubernetes to manage microservices on-premises and/or in public clouds—whether or not it is distributed—can realize value from using a service mesh to accelerate Kubernetes adoption. A mesh eases the integration of containers into existing environments and provides orchestration capabilities for modern deployments.

Istio manages deployments, resilience, and security in Kubernetes environments. As an open source tool, Istio provides uniform, efficient, and reliable ways to secure and connect the application's logic. The software also monitors services while load balancing and authenticating without adding service code.

- Istio is extensible and customizable with WebAssembly (Wasm) to build advanced filters and policies.
- The Istio control plane addresses the challenge of achieving consistency in policies and security across single or multiple clusters, whether located on-premises, in public clouds, or in both locations.
- Istio most often rides in an Envoy Proxy open-source sidecar with each Kubernetes pod, an efficient placement.

- Istio allows organizations to add applications and deploy clusters more rapidly and with greater levels of security, resilience, and scalability.

- While Istio is Kubernetes-native, it can be used to handle server clusters, virtual machines, or other endpoints outside of Kubernetes, while supporting legacy apps (such as those running on SOAP) that haven't been modernized (yet).

You may be wondering what the alternative approaches to Istio are. There are alternatives—however, they come with a cost. One alternative approach to consider is to use service meshes supplied by cloud service providers. These meshes can only be used with the service provider's specific cloud, making them fine for organizations that are not looking to scale, don't need advanced features, or institute multi-cloud or distributed cloud environments but unsuitable for any other customers. Another alternative would be to use older API infrastructure offerings that were built pre-Kubernetes. These offerings primarily focus on network monitoring and static routing. It is critical for most organizations with growing microservices workloads to understand that a Kubernetes-native service mesh will provide the growth and capabilities they need across containers as well as virtual machines.

Enhanced Open Source Istio – Build or Buy?

Modern IT organizations are heavily dependent on open source software. In a research study completed by ESG, 72% of organizations stated that 26% to 75% of their organization's code base was pulled in from open source. Kubernetes and the Cloud Native Computing Forum landscape are the main focus for open source projects, and vendors are aware that customers are looking to them to support these initiatives.

However, the use of open source code introduces challenges. Who will provide technical support for the code? Can organizations keep up with versions and compatibility, and is the code feature-complete for enterprise requirements? How fast are Common Vulnerability Exposures (CVEs) resolved? These and other questions apply to Istio just as much as they apply to other open source software. Like some other open source tools, Istio is updated more frequently than conventionally created software (with quarterly releases), and support for older versions is limited, preventing IT organizations from working at their own pace on updates. And while Istio is a preferred solution because of its more robust functionality compared to other API infrastructure and service meshes (and its cloud- and Kubernetes-native architecture), there are many desirable functions that are not yet incorporated into the community edition of Istio. Just some examples of the latter include support for federated multi-cluster management, serverless functions, FIPS compliance, and tight integration with external authentication tools.

> **The Community Edition of Istio Needs More Capabilities**
>
> There are many desirable functions that are not yet incorporated into the community edition of Istio.

The Build Option

One way to address these issues is for organizations to develop their own enhancements to Istio. This approach is not for the faint-hearted, and before organizations adopt this approach, they should ask themselves:

- Is the organization aware of the most current best practices in this technology sector, which will be essential to de-risk the project and deliver initiatives on-time?

- Does the organization want to create and pay for the cost of a team to develop the enhanced functionality, while also managing and support the deployment of the mesh and covering all applications and environments current and future, with benefits delivered to operations organizations and not just operations departments?
- Will the service mesh meet the organization's needs for service level agreements (SLA), and does the organization want the responsibility of supporting those SLAs? Open source Istio, like most open source projects, does not in itself provide SLAs.

Although some organizations will be able to answer yes to these questions, ESG believes they should consult with and consider the software and services offered by vendors that have already learned many lessons working with multiple customers. Those lessons can be painful and can be avoided by relying on the experience of vendors.

The Buy Option

Alternatively, and exactly as with other open source tools, customers can address the support, functionality and other issues related to Istio by deploying "enterprise" or enhanced versions rather than community editions of the software. Choosing the right enhanced version of Istio to meet an organization's needs will not only save development time and resources by providing essential enterprise capabilities out of the box, but also should address:

Consider This:

- Save development time/resources with key capabilities not provided in OSS.
- De-risk projects with guidance and support (development and operations).
- Benefit from tested, validated, and supported integrations and workflows.
- Leverage Istio open source maintenance and support for older versions.

Choosing the right enhanced version of Istio will save development time and resources by providing essential capabilities out of the box.

- Security and compliance requirements (e.g., FIPS, WAF, DLP).
- Federated management and configuration of single or multiple clusters distributed across hybrid on-premises and cloud execution venues.
- Consistent manner of operation across all the organization's operating environments, tested and validated with support the organization's integrations and workflows.
- Integration with the development processes that are already used by the organization or may be used in future.
- Dynamic support for rapid changes in application code. Some large organizations update or change application code every 4 months or more frequently.
- Technical support provided in a way that existing DevOps teams prefer (e.g., private Slack channels, Zoom/phone, or incident systems).
- Extensibility with web assembly (Wasm) to enable advanced filters for security or complex logic.
- Support for previous versions, including Long Term Support (LTS) N-4 because not all organizations are ready to implement quarterly updates. Security patches are important, however, because vulnerabilities can surface after a

release of Istio. When they surface, commercial suppliers of enhanced editions of Istio should guarantee to quickly release patches that can also be backported to earlier versions.

The Bigger Truth

Regardless of where you are in the world of Kubernetes, it is clear that it can become complicated very quickly. From an analyst perspective, we are seeing the networking, configuration, and the architecture continue to be complex, but the good news is that there are vendors in the market to help organizations sort through the complexity. When it comes to service meshes, my hope is that this paper addresses the areas to consider when looking into ways to optimize your modernized cloud-native applications and microservices. Take time in to consider the value of the service mesh. Because a service mesh solves critical challenges for modern applications running in Kubernetes and hybrid-cloud, it is important to consider how the service mesh uses the sidecar model to provide needed functionality consistently alongside microservices. This is key as it is not intrusive to your applications or microservices and separates the connectivity logic from the individual application components. In fact, most services do not realize they are running as a mesh.

A service mesh provides many advantages. However, replacing one service mesh with another can be a complex challenge, especially if your service mesh is incorporated across your applications and microservices. My point of view is that the best way to accomplish your business objectives is to start with a focused project, understand the service mesh deployment, and then incorporate the service mesh into your organization.

An enterprise version of an Istio service mesh offers the fastest time to value as well as a single point of contact for support. As an organization scales, there is no question that a service mesh would be a critical addition to its orchestration layer. Moreover, an enterprise version would offer the most robust path to deployment.

Enterprise Version of Istio Makes the Most Sense

As an organization scales, there is no question that a service mesh would be a critical addition to its orchestration layer. Moreover, an enterprise version would offer the most robust path to deployment.

Enterprise Strategy Group is an IT analyst, research, validation, and strategy firm that provides market intelligence and actionable insight to the global IT community.


www.esg-global.com


contact@esg-global.com


508.482.0188