Embracing a Cloud-Native Approach
MICROSERVICES AND SERVICE MESH
FOR DIGITAL TRANSFORMATION

COMMISSIONED BY

vmware

NOVEMBER 2018

©COPYRIGHT 2018 451 RESEARCH. ALL RIGHTS RESERVED.
About this paper

A Pathfinder paper navigates decision-makers through the issues surrounding a specific technology or business case, explores the business value of adoption, and recommends the range of considerations and concrete next steps in the decision-making process.

ABOUT THE AUTHOR

JAY L YMAN
PRINCIPAL ANALYST, CLOUD NATIVE AND DEVOPS

Jay Lyman is a Principal Analyst with 451 Research’s Applied Infrastructure & DevOps Channel. He covers infrastructure software, primarily private cloud platforms, cloud management and enterprise use cases that center on orchestration, the confluence of software development and IT operations known as DevOps, Docker and containers. Jay’s analysis encompasses evolving IT operations and software release models, as well as the technology used to create, deploy and support infrastructure and applications in today’s enterprise and service-provider markets. Key areas of research also include OpenStack, PaaS and enterprise end users.
Introduction

Today’s modern enterprise can drive high-impact digital transformation via technology resources such as cloud computing and emerging approaches to application development and deployment, such as containers and microservices. However, adopting these new technologies brings significant challenges, including operating efficiently at scale, a growing complexity and diversity of technology components to monitor, manage and secure, and cultural challenges that center on efficient organizational collaboration and ‘legacy’ enterprise IT inertia. This paper examines how a service mesh architecture enables organizations to effectively leverage microservices and a cloud-native approach to application development to achieve greater development velocity, greater overall efficiency and set the stage for successful digital transformation. It was written for both enterprise IT decision-makers and technology practitioners to better understand the drivers, challenges and perceptions that are shaping service mesh software and practices.

From Monolithic AppDev to Microservices

The shift from traditional application development to cloud-native application development centers on the transition from monolithic applications to an approach that relies heavily on microservices, a software development approach that breaks large applications into lightweight and modular parts – often in containers – that can scale more simply and horizontally. A microservices approach segments application functionality into composable building blocks held together via RESTful APIs. By breaking down monolithic applications from their massive, complex internal architectures into numerous, smaller parts that can be independently scaled, microservices can make development, as well as updating and deploying, less complex and, therefore, more efficient.

Additional advantages to using microservices include efficient workload portability across hybrid IT infrastructures, increased modularity and enhanced management benefits. Nevertheless, microservices present distinct challenges, such as increased difficulty of monitoring and event logging, as well as performing testing and debugging processes across a decentralized, loosely coupled application environment. In addition, enterprises must contend with application silos spanning multiple platforms and clouds and an explosion of elements to scale, secure and monitor. Other challenges include disjointed security policies and event logs, as well as the diverse, fragmented libraries and code bloat typical of today’s polyglot programming.

Despite the challenges, cloud-native application development is growing. According to 451 Research’s Voice of Enterprise: Cloud, Hosting and Managed Services, Workloads and Key Projects 2018 survey, 80% of organizations are taking a cloud-native approach with at least some of their applications (See Figure 1). As the benefits of using microservices fuel more cloud-native applications in the enterprise, it is becoming clear that this new approach requires new management tools and processes. Additional research, such as our 2017 survey of 200 enterprise IT decision-makers at US organizations using containers, indicates that the top challenges of implementing containers are security, compliance and
regulatory concerns, complexity and production use. Addressing these challenges is where service mesh technology comes in. Service mesh represents a new paradigm by which existing solutions to distributed application programming can be applied consistently and effectively for successful microservices implementations.

**Figure 1: Current adoption of cloud-native or cloud-enabled software for internal development**

*Source: 451 Research’s Voice of the Enterprise: Cloud, Hosting and Managed Services, Workloads and Key Projects 2018*
Service Mesh Definition and Adoption

To understand how microservices can be enhanced by a service mesh, let’s first define it. As we established, a microservice architecture is a software development approach that breaks larger applications into a series of more lightweight components or services for easier horizontal scaling. A service mesh is an inter-service communication mechanism for microservices. Service mesh can be thought of as a software application layer that serves to operate and allow observation of communication patterns among different services. Microservices do not communicate directly with other microservices; instead, they rely on software called a service mesh or sidecar proxy. Service meshes usually support some networking functionality, such as resiliency and service discovery. Some service mesh technologies also support traffic management, policy enforcement, observability, and security and identity management.

Figure 2: Service mesh concept
Source: https://istio.io

Istio Architecture
Service connectivity, security, control and observability

In effect, service meshes act as an application layer for control, allowing developers to focus on the application rather than managing the infrastructure or environments in which it is running. In this way, service mesh can enable diverse elements such as content servers, order management and payment processing to interact. As with other cloud-native software, such as containers and Kubernetes, open source software is typically an important part of a service mesh environment. Modern software development tenets such as modularity and componentization enable integration and flexibility with other technologies to achieve a service mesh. Beyond popular open source service mesh technologies
such as Istio, Calico, Conduit, Envoy and Linkerd, there are additional open source software projects and components that can play a significant role in enterprise deployment of service mesh. This ecosystem around service mesh and related cloud-native software can add to the complexity and confusion about the topic, but it also highlights the broad engagement and innovation among both enterprise vendors and end users.

The current level of service mesh adoption among enterprise organizations is remarkable given its nascent status. According to a survey of 200 enterprise IT decision-makers and practitioners in North America polled in April and May 2018, enterprise verticals such as retail and consumer packaged goods, oil and gas, and manufacturing are experiencing advanced implementation of service mesh, including in production systems (See Figure 3). Overall, organizations across most vertical industries are involved in service mesh discovery and evaluation, trials and test projects, or in testing and development. Only a small minority indicated that they have no current plans to adopt service mesh. The high level of awareness and growing production use of service mesh are evidence of accelerating momentum, despite both technical and cultural challenges. This adoption data is consistent with the rapid pace of growth observed for emerging cloud-native technologies such as containers and Kubernetes.
The rapid uptake of cloud computing infrastructure, containers and microservices among enterprise organizations is further highlighted by this fairly advanced adoption of service mesh technology among some verticals. The same survey of 200 enterprise IT decision-makers and practitioners in North America indicates growing service mesh production use, with respondents in the healthcare, high tech, hospitality, financial services and insurance verticals indicating significant growth of their production implementation of service mesh (See Figure 4). We also highlight that microservices adoption and implementation, particularly production use, will be fueled by evolving service mesh since it presents an effective approach to solving critical microservices challenges.
Service mesh is not only a potential enabler of microservices and net-new cloud-native application development and deployment, but it can also support enterprise migration of legacy applications to the cloud and the modernization of applications for the cloud – often through container technologies. While enterprises have embraced containers and microservices for primarily cloud-native net-new applications, the overall approach is being extended for use with legacy applications in the hope that they can be supported more efficiently and effectively in the cloud. Service mesh can also accompany and complement the use of Kubernetes container orchestration and distributed application frameworks, which is well timed with growing enterprise use of multi- and hybrid-cloud architectures. More than half of those that are developing cloud-native software are designing for multiple clouds (32% on any cloud and 22% on any public cloud), according to our Voice of the Enterprise: Cloud, Hosting and Managed Services, Workloads and Key Projects 2018 survey (See Figure 5).
We also observe a strong connection between cloud-native software development and modern application development approaches including DevOps, Twelve-Factor and Six Sigma, which are quickly growing and will gradually displace traditional methodologies such as waterfall and ITIL. Thanks largely to the pressure to digitally transform, those trends and microservices are also being driven by the top of the organizational chart as business leaders seek to extend the impact of these trends well beyond the grassroots developer level (See Figure 6).

**Figure 5: Approaches to cloud-native software development**
*Source: 451 Research’s Voice of the Enterprise: Cloud, Hosting and Managed Services, Workloads and Key Projects 2018*

- Design it to run effectively on any cloud environment: 32%
- Design it to run on a specific public cloud environment: 30%
- Design it to run effectively on any public cloud environment: 22%
- Design it to run on our own private cloud: 17%

**Figure 6: Significance of microservices to digital transformation, by company size**
*Source: 451 Research*

<table>
<thead>
<tr>
<th>Employee Count</th>
<th>A primary, central role</th>
<th>A significant role</th>
<th>A minimal role</th>
<th>No role</th>
<th>Not sure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,000-4,999 employees</td>
<td>21%</td>
<td>63%</td>
<td>13%</td>
<td>2%</td>
<td>1%</td>
</tr>
<tr>
<td>5,000-9,999 employees</td>
<td>14%</td>
<td>63%</td>
<td>21%</td>
<td>2%</td>
<td>2%</td>
</tr>
<tr>
<td>10,000+ employees</td>
<td>36%</td>
<td>41%</td>
<td>23%</td>
<td>0%</td>
<td>1%</td>
</tr>
</tbody>
</table>

PATHFINDER | EMBRACING A CLOUD-NATIVE APPROACH
While a service mesh typically provides some network benefits (e.g., resiliency and service discovery), it does not necessarily solve the wide spectrum of networking challenges at all layers of the software stack, such as automating security. We believe that best practices dictate a layered approach to supporting dynamic application environments. Such an approach would leverage service mesh along with other technologies – such as software-defined networking and software-defined storage – to address problems at different layers, which also vary based on the needs of stakeholders including developers, IT operations, combined DevOps teams, network administrators, storage administrators and security administrators.

**Critical Elements, Features and Capabilities of Service Mesh**

Service mesh software (and methodology) consists of key elements such as control plane components for the management of microservices, data plane components that transport data along with applications, APIs, and integration points. As for service mesh features and capabilities, customers view simplified workflows, granular troubleshooting and distributed tracing tools as the most critical. Additional service mesh capabilities center on discovery and ‘just knowing what’s there’; federation for establishing consistency amid the complexity; observability/visibility through logging, monitoring and alerting; connectivity and traffic management to avoid bottlenecks and outages; control and unified policy across services for easier management; and improved security capabilities to reduce overall risk exposure.

The foundational elements of service mesh are important for addressing microservices challenges that center on security and compliance, complexity, consistency, and visibility. The benefits and challenges of microservices and thus the utility of service mesh can also vary greatly depending on the perception of the stakeholder. While IT leadership, directors and C-level individuals place a high priority on security and compliance, developers and IT administrators are more focused on resolving issues around complexity and inconsistency. Service mesh is typically top of mind across all stakeholders in matters of visibility and observability, highlighting how service mesh serves as a central control and viewing plane for microservices.

Workload portability is also driving a shift to microservices, particularly among larger enterprises adopting and leveraging hybrid cloud strategies that involve multiple public clouds, private clouds and on-premises IT environments. Additional microservices drivers include improving scalability, simplicity, faster deployment time, enabling modularity, and overall management benefits – all of which can be enabled by and supported via service mesh.
Is Service Mesh Ready for You?

The most widely known and adopted service mesh is Istio, an open source service mesh technology started by Google, IBM and Lyft with growing involvement from IT vendors and end users. Istio is designed to allow its users to connect, control, secure and observe microservices. Core features of Istio focus on traffic management, platform support across hybrid infrastructures, integration and customization, observability, and security. Istio 1.0 was released earlier this year and is the latest and most stable version of the software intended to connect, manage and secure microservices. In addition to its focus on scaling high-volume and complex microservices, Istio 1.0 is designed to let developers and operators control microservices management, data and security. Additional enhancements in the works for Istio include unified support and experience across public clouds and on-premises environments.

While Istio is the most well-known service mesh technology in the industry, there are several other service mesh technologies of note, including some that are used within Istio. Envoy, for example, is an open source, high-performance proxy used to manage inbound and outbound traffic for the services in a service mesh. Envoy is a primary component of Istio and other service mesh software, such as F5 Networks’ Aspen Mesh. Another service mesh technology is Calico, an open source network policy control plane that can be used with service mesh to secure applications. Finally, we mention Conduit, which is a lightweight open source service mesh for Kubernetes and has its roots in the Linkerd service mesh technology.

While 451 Research’s surveys and industry conversations indicate that service mesh is starting to be implemented for production applications and microservices, there are distinct challenges to full-scale production use. These challenges include lack of visibility, difficulties in troubleshooting outages, fragile service communication and the potential for cascading failures.

Service Mesh Use Cases and Stakeholders

Service mesh is still a nascent and evolving class of technology, but there are some consistent use cases that have emerged among enterprise organizations we have studied. Among the primary service mesh use cases is service discovery (since organizations need to understand things such as what services are running, whether and how services are communicating, and which data stores are being accessed). Service mesh can aid in service discovery by simplifying service registry and enabling the generation of topology maps.

Another common service mesh use case is to address management federation, typically in environments that use multiple platforms (IaaS, PaaS, SaaS, etc.) and multiple clouds and infrastructures (public, private, on-premises). Furthermore, service mesh is deployed for enhanced security, drawing on capabilities such as access control and service encryption. Other service mesh use cases center on observability and visibility for both developers and IT operations, connectivity, and traffic management. Finally, service mesh fits with modern DevOps methodology by supporting tasks and requirements of both developers and IT operations teams, as well as other stakeholders including architects, security professionals and site reliability engineers.
Conclusion and Outlook

Containers, container orchestration, microservices, and other cloud-native technologies and approaches are integral to successful enterprise IT digital transformation. The latent promise of greater application agility, portability and scale for cloud-native applications, as well as for migration of legacy applications, is driving rapid adoption of containers and microservices, but this approach brings distinct challenges in complexity, consistency and security.

Service mesh can be an effective tool to address these challenges and reap the benefits of emergent cloud-native technology and methodology. While it is still early, we see microservices and service mesh evolving and maturing rapidly, helped along by the fact that they are being driven and guided not only by vendors, but also end users and the broader open source software communities and ecosystem around cloud-native technology. As microservice architectures are deployed more broadly by a growing variety of enterprises, we expect service mesh will emerge as an effective approach to mitigating challenges and, thus, a key enabler of digital transformation.

By leveraging network virtualization to extend networking and security policies from traditional workloads and data centers to containers and public clouds, you are building a foundation for connected and protected cloud-native environments at the infrastructure layer. At the application layer, you can complement these benefits by leveraging service mesh technology to gain observability, control, and security across not only services, but also the data they access and the users that interact with them. Creating a seamless federation of policy and communication across multiple cloud-native platforms and unifying operations across development, infrastructure, operations, and security teams gives you a powerful end-to-end solution for mastering the cloud-native world, from infrastructure to application.

To learn more about the core service mesh technology, visit Istio.io. And to find out how VMware is contributing to and building additional functionality on top of the Istio project, visit cloud.vmware.com/service-catalog.
About 451 Research

451 Research is a preeminent information technology research and advisory company. With a core focus on technology innovation and market disruption, we provide essential insight for leaders of the digital economy. More than 100 analysts and consultants deliver that insight via syndicated research, advisory services and live events to over 1,000 client organizations in North America, Europe and around the world. Founded in 2000 and headquartered in New York, 451 Research is a division of The 451 Group.

© 2018 451 Research, LLC and/or its Affiliates. All Rights Reserved. Reproduction and distribution of this publication, in whole or in part, in any form without prior written permission is forbidden. The terms of use regarding distribution, both internally and externally, shall be governed by the terms laid out in your Service Agreement with 451 Research and/or its Affiliates. The information contained herein has been obtained from sources believed to be reliable. 451 Research disclaims all warranties as to the accuracy, completeness or adequacy of such information. Although 451 Research may discuss legal issues related to the information technology business, 451 Research does not provide legal advice or services and their research should not be construed or used as such. 451 Research shall have no liability for errors, omissions or inadequacies in the information contained herein or for interpretations thereof. The reader assumes sole responsibility for the selection of these materials to achieve its intended results. The opinions expressed herein are subject to change without notice.