



Selecting The Right Microservices Platform

Key Challenges and Considerations to accelerate Cloud Native Journey

Overview

As enterprises embrace Cloud Computing and DevOps as a part of their modernization journey, they are faced with a decision on whether to forklift existing application to cloud or re-architect the applications to meet the distributed nature of the underlying cloud. A forklift of monolithic application is often seen as a short term transitional effort as organization re-architect their applications to a Microservices based architecture. In this position paper, we will highlight the role of Microservices in a modern enterprise and list out important criteria for choosing a platform for developing and deploying Microservices. This position paper is designed to serve as a framework for the enterprises to evaluate Microservices platform in their enterprise modernization journey.

Introduction

The emergence of cloud in the last decade has changed how enterprise IT is being done and there is considerable pressure on organizations to transform themselves to effectively compete in a fast-moving global economy. Software is driving innovation across all verticals pushing organizations to embrace software as the pathway to innovation. This brings focus on IT, as an enabler of innovation by making it easy for developers to take an idea to code and deploy in production many times a day than months long cycles in the past. In a modern enterprise, IT is part of the core innovation than just a cost center.

As cloud computing gained traction and helped enable organizations deliver business value much more rapidly than anytime in the past, software fast became the dominant platform for innovation. Software helped organizations to not just innovate rapidly on their core competency, it also helped them innovate in adjacent areas or even in entirely new line of business. Software driven innovation became the competitive advantage rather than a cost center aiding their core competency. CIOs started focussing on transforming their organizations to be a modern enterprise, driven by a software platform leveraging cloud.

With cloud computing gaining widespread acceptance as the defacto infrastructure for modern applications, enterprises embraced DevOps as the next step in their modernization journey to remove the bottlenecks in the application deployment lifecycle. With Cloud and DevOps, enterprise IT helped enable accelerated application delivery, resulting in application deployed in production in matter of days/weeks than months. Even though most IT organizations gained significant increase in the speed at which they delivered business value, the ROI is still sub-optimal even with IT modernization. The sub-optimal ROI can be traced back to deploying monolithic applications on cloud infrastructure, which is distributed in nature and scales differently from these applications. Microservices, a suite of modular set of services architected around a business capability or function, is the next frontier in enterprise modernization journey. With Containers gaining tremendous momentum with advantages from developers to IT operations, they can be considered as the right encapsulating mechanism for Microservices.

Microservices architecture goes beyond the needs of enterprise agility and serves as a foundation for IoT, Big Data and other modern application use cases. Embracing Microservices for newer apps is the most convenient starting point for the transformation journey but it is also important to have a strategy in place so that other applications are re-architected in due course. In this position paper, we will focus on the Microservices architecture in modern enterprise, it advantages, challenges and

considerations for selecting the right Microservices platform. This paper provides some prescription on the key factors to consider as organizations evaluate various Microservices platform suitable for their needs.

Slaying the Monoliths, Microservices as the foundation

Even though some legacy apps might have a monolithic architecture in how business logic is implemented, the monolith problem is also present in applications with modular architecture in the implementation of business logic. Even these applications are deployed as a monolith, inconsistent with the more distributed architecture of the underlying cloud infrastructure. For example, many traditional enterprise Java applications are deployed as a single WAR file or a self contained executable. The reasons for the dominance of Monoliths in enterprises include ease of development, testing and deployment, But any potential benefit with Monoliths are negated with challenges like slow deployment speed, complexity in scaling, inflexibility in application evolution with modern tools, etc.. Microservices are considered the next important frontier in the modern enterprise journey.

Even though there are many variations and confusion in the definition of Microservices, Rishidot Research subscribes to the definition [put forward by Martin Fowler](#), one of the leading advocates of Microservices architectural patterns.

The microservice architectural style is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API. These services are built around business capabilities and independently deployable by fully automated deployment machinery.

Our extensive research and based on our conversations with software architects from both startups and enterprises, we have concluded that the above definition of Microservices is a perfect fit into our [own definition of a Modern Enterprise](#).

The key components of a Microservices architecture are:

- Composability
- Lightweight
- Independent deployability
- Functional representation of an unit of business logic
- Lightweight communication protocols

Leading edge enterprises like Amazon, Netflix and others have embraced Microservices architecture, gaining competitive edge over their traditional counterparts. As other enterprises embrace the modernization journey, they are faced with the dilemma of moving to a more modern architecture like Microservices architecture without disrupting their mission critical applications running as monoliths. A brute force rip and replace approach will not work for

these organizations. A more methodical modernization journey that also meets their security, governance and compliance needs is needed. Off late, organizations beyond the leading edge enterprises like Amazon, netflix, Uber, etc. are considering Microservices as a part of their IT modernization journey. The current phase in the Modern Enterprise is focussed on providing standardized platforms that enables key building blocks and patterns needed to help mainstream developers build their applications using Microservices architecture.

In this section, we will talk about the benefits and challenges with Microservices architecture and help enterprise IT leaders embrace a strategy that better fits their organizational needs. A clear understanding of the challenges with the modern architecture pattern is imperative for the evaluation of Microservices platforms that can meet the enterprise needs. As enterprises embrace Microservices, they are faced with a scale involving thousands of services and it is important that the stakeholders understand the underlying dynamics as they plan their platform strategy.

The Microservices Advantage

Microservices offer some key advantages for enterprise IT as it helps them allocate personnel and resources more efficiently, maximizing the ROI and delivering the key competitive asset needed to bring transform IT to be part of the core innovation team.

Some of the advantages of Microservices include:

- **Deployment:** Provides deployment agility and fine grained control over replication, scaling, security, technology choices, etc.
- **Resiliency:** Modern enterprise applications focus on resiliency over reliability. Microservices architecture help provide resiliency for applications because even if a single Microservice or group of Microservices are not available, the availability of the application is not affected. The application can continue to serve users with limited functionality than going down completely, as in the case with monoliths
- **Zero Downtime Upgrades:** Microservices architecture makes it easy to to do blue green deployments or other zero downtime upgrades
- **Better Resource Utilization:** Since Microservices architecture allows components to scale independently, it not only makes scalability much more efficient but it also allows for better resource utilization. Instead of scaling all the resources used by an application, we can just scale the underlying resources for features having heavy load
- **Technology Evolution:** Microservices makes it easy to use newer programming frameworks or technologies because changing the technology underlying any single service will not impact the application as a whole. This allows for using the right technologies for specific needs, allowing a better evolution of the application to meet the rapidly changing business requirements
- **Better Talent Management:** With Microservices architecture, it is possible to use the [Two Pizza Team](#) approach to application delivery. With Microservices and DevOps, it is easy for organizations to effectively manage human resources to maximize innovation without much waste

With Microservices and DevOps, the modern enterprise IT is very much part of the core innovation team focussed on delivering business value rapidly and giving a competitive edge in the fast moving market.

Understanding The Challenges

Microservices architecture is not the magic pill that completely transforms IT overnight to be part of the core innovation team. Even though Microservices offer some key advantages in the IT transformation journey, there are some key challenges that needs to be addressed before realizing the benefits. In this section, we will highlight some of the challenges in using the Microservices architecture. These challenges can be mitigated with the right platform and the first step towards selecting the right platform is in understanding the challenges.

Microservices remove the complexity in the monolithic architecture that impacting application scalability, reliability, evolution, etc.. However, it moves the complexity to the application deployment, bringing in some challenges that needs to be understood and addressed:

- With hundreds or thousands of services that constitute an application, deployment becomes more complex. It involves more teams focussed on deployment than in the traditional IT set up. With traditional cloud infrastructure, deployment becomes overly complex leading to sub-optimal speed and ROI
- With hundreds or thousands of services, management of services and its dependencies becomes more complex than traditional monolithic applications. Resource allocation should be efficient and the elasticity should be managed more coherently and based on the needs
- The performance of Microservices can be impacted if the dynamic discovery and service lookups are not managed properly. Especially with individual services communicating with each other over a network than internal calls, any bottleneck in service discovery will have huge impact on application performance
- With multiple services relying on a contract that defines the interactions between the services, modifying the services and the contract becomes more complicated. The complexity overlays both in the setting up of the contract and in the management of changes as different services evolve
- Setting up and managing automated tests are more difficult than in the case of traditional monoliths and requires a more automated approach to managing these tests across various services and teams
- The inter process communication should be better managed to ensure resiliency during failure of a single service or a group of services. Even though it is imperative for the developer to handle these failures gracefully, it is important for the underlying platform to help streamline the inter process communication needs
- Managing database transactions is more complex than traditional monoliths. Without a seamless solution to handle database, developer productivity will be impacted
- With so many services and a fluid perimeter, managing the security is difficult than in traditional IT. Similarly, managing the compliance requirements are much more complex because the traditional models of perimeter security fails in the case of

Microservices. Managing identity and authentications across these services adds to the complexity of meeting the compliance requirements

- A typical enterprise will have hundreds or thousands of services and managing the governance is much more complex than in the traditional enterprise IT

Even though using Microservices is an advantage, it is important to understand the challenges and find a platform that better mitigates these challenges.

Microservices Platform

When it comes to Microservices platform, most of the industry discussions center around the programming frameworks that help developers create Microservices. But an evaluation of Microservices platform goes well beyond the programming frameworks and it should consider the entire lifecycle of developing and deploying Microservices. Without any support for deployment and management of Microservices, the platform offers limited benefits and it may end up increasing the costs and impacting the agility. In this section we will highlight some key capabilities of Microservices platforms and considerations one should take into account during the platform evaluation.

Key Capabilities

Some of the key capabilities of Microservices platform are outlined below:

- The platform should abstract away the Microservices patterns and make it easy for developers to create the code
- The platform should have good meta-data models for describing service capabilities
- Support for synchronous (public facing APIs) and asynchronous calling (Internal calls)
- Necessary infrastructure for publishing the services in a registry and enabling the discovery of service
- Support for messaging including support for protocols like queuing, pub-sub and other patterns. Such messaging components should follow the security model needed for the platform
- Multiple language support and elastic runtimes
- Support for data persistence
- Secure API Gateway
- Security and Compliance management
- Deployment automation with support for containers and container orchestration tools
- Microservices analytics

These are some of the Microservices focussed capabilities any platform should have along with other enterprise grade features needed for application deployment.

Key Considerations

Any evaluation of a Microservices platform should include the following considerations to maximize the benefits. The key capabilities we identified in the previous section along with the following considerations should serve as a framework for evaluating a Modern Enterprise Microservices Platform. Rishidot Research strongly urge you to ask the platform vendor the following questions and get their answers during the evaluation phase.

1. Can the platform handle Microservices at scale? A typical modern enterprise will have thousands of services and the platform should be able to handle the scale. Ask the vendor about any benchmarks they can provide on this
2. Does it support multiple underlying platforms? Does it support IaaS/PaaS from various vendors in the market? Read the documentation to ensure that your infrastructure provider is supported as a first class citizen through tighter integration
3. Is it a polyglot platform offering support for multiple programming languages and frameworks including both proprietary and open source?
4. Is the platform Multi-tenant? Multi-tenancy is critical from resource efficiency in a Microservices environment
5. Does it offer integration with enterprise technologies to meet your needs today and in the future?
6. Does it support the standards based container tools like Docker and Kubernetes, DevOps lifecycle tools, etc?
7. Does it offer seamless control for application release and deployment through the service delivery lifecycle?
8. Does it offer support for lightweight RESTful messaging tools?
9. Does it offer extensive logging support and monitoring?
10. Does it integrate with Active Directory or other single sign on tools available in the market? Does the platform meet my governance needs?
11. Does it provide a management console to handle the security and compliance needs of your organization?
12. When you have hundreds or thousands of Microservices, managing the relationships between them becomes complex, both from the resiliency of the application and also from a governance perspective. Does the platform map the relationships between the services and offer it through the management plane?

These are some of the questions you need to ask as you evaluate various platforms available in the market.

Microservices platform can range from DIY IaaS+ options, PaaS offerings and purpose built Microservices platforms. For an organization planning to embrace Microservices across the organization, a purpose built platform offering end to end capabilities for Microservices development, deployment and management may come handy. As you evaluate these platforms, you need to match your organization's needs with the key capabilities of the platform to pick the right one.

Summing Up

Embracing Microservices is a critical part of Modern Enterprise journey and it goes beyond the architectural considerations of the application. Microservices model brings with it both technical and cultural changes inside the organization, enabling smaller teams with common goals to innovate faster and at scale. Even though a move to Microservices is very valuable for enterprises, it comes with lot of technical challenges that requires careful consideration and a selection of right platform to meet these challenges. A successful Modern Enterprise will blend Microservices architecture with DevOps to streamline the value creation. In order to be successful, organizations should exercise deliberate evaluation from abstracting the architectural patterns to ease of deployment and management to security and governance. With a right platform for Microservices implementation, your modern enterprise journey will produce higher ROI and will help your organization outsmart the competition.



Macaw is an enterprise grade prescriptive Microservices platform. Macaw provides Application Modernization path for enterprises to build modern application architectures leveraging Containers, DevOps, Microservices and Micro Analytics. Macaw accelerates Enterprise journey to embrace Microservices and Containers with many built-in foundational services and turnkey operational capabilities. For more information visit www.macaw.io



Rishidot Research is a next gen research firm helping organizations transform to modern enterprise, ready to compete in today's market. Our services will empower the business and IT leadership in their transformation journey and help teams develop right strategy for their organization. For more information, visit www.rishidot.com