Deloitte.

Microservices and APIs: Building a Successful Strategy for Application Modernization

Deloitte Consulting LLP

October 2022

Table of contents

Introduction 3

Approach for IT Modernization 4

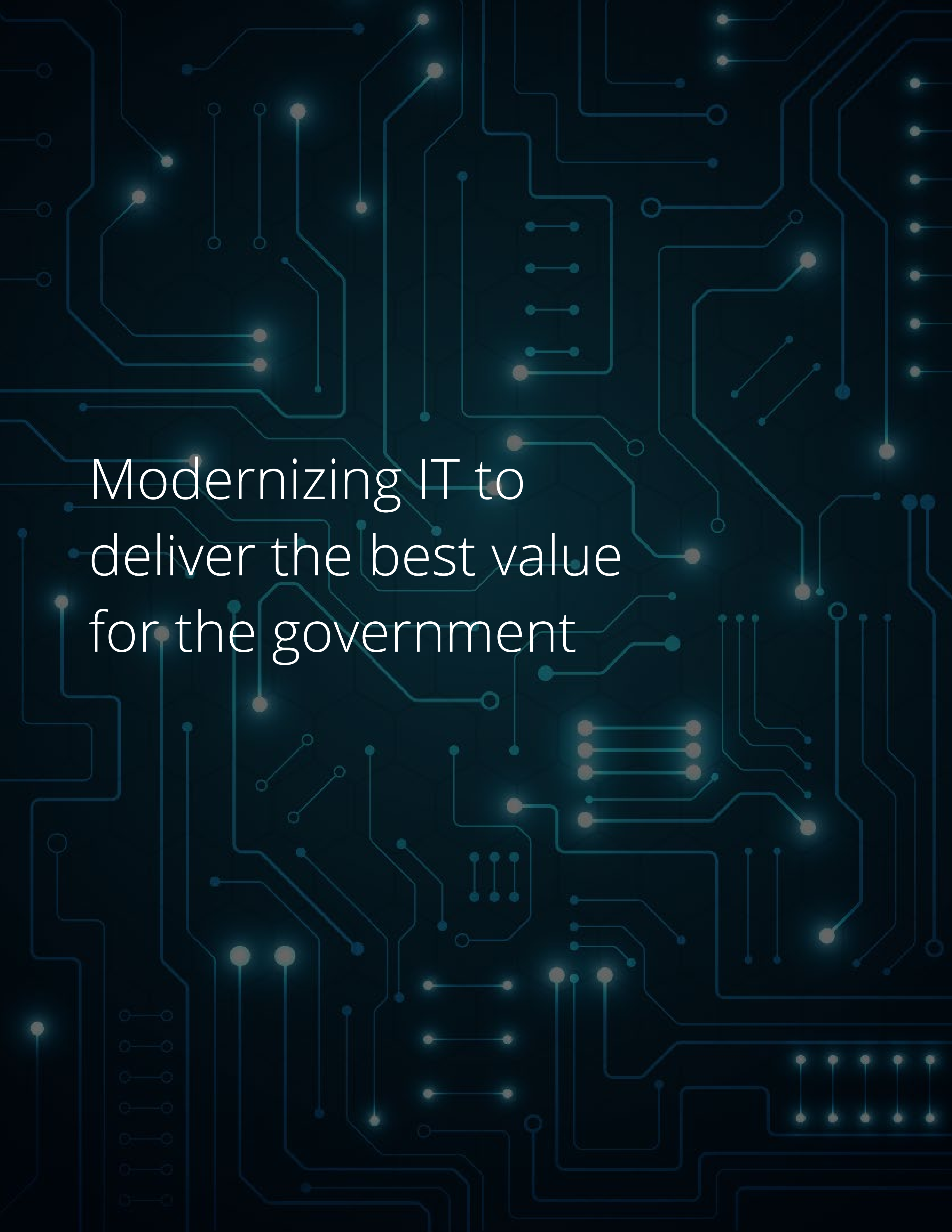
What are Microservices and APIs? 5

Impact and Benefits of Microservices and APIs 8

Getting Started 9

Contributors 10

References 11



Modernizing IT to deliver the best value for the government

Introduction

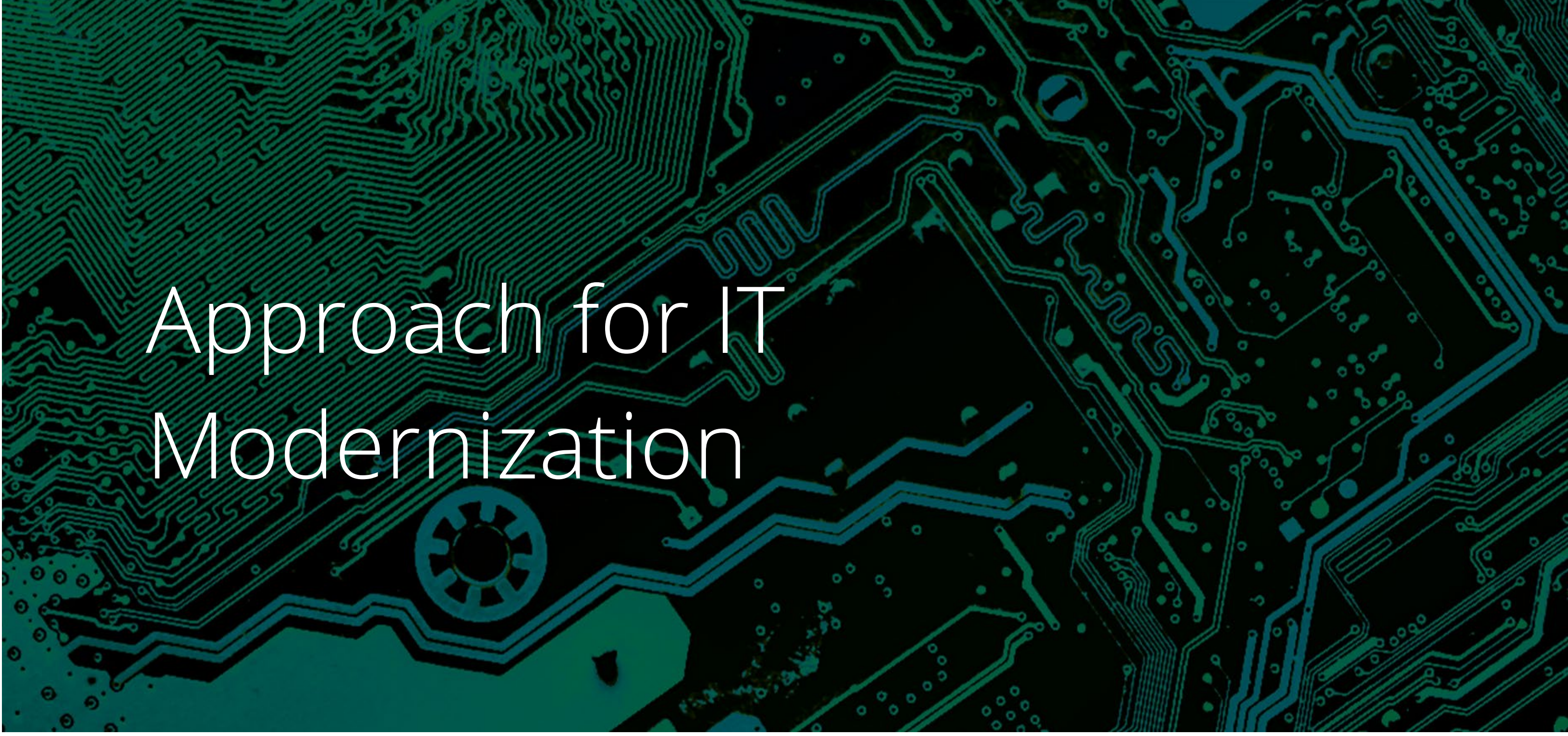
In a world that is advancing rapidly with innovative technologies and solutions disrupting markets, organizations must rethink their IT approach, particularly their application architecture and how it enables application modernization. However, not every government agency can afford the big-bang IT implementations, with its unique constraints and pressures, to modernize their infrastructure without disrupting existing operations. It is imperative for government agencies to adopt an open architecture to promote a flexible IT strategy in a rapidly evolving and changing cloud marketplace. Therefore, agencies must discover ways to deliver incremental value that integrates effectively within their existing ecosystem. As such, digital transformation models based on microservices architecture, and Application Programming Interfaces (APIs) provide the flexibility and speed for government agency leaders who need to achieve meaningful change while wanting to be perceived as responsive.

Challenges Facing the Government

Government agency leaders are facing enormous challenges when it comes to legacy IT. While legislation changes and the addition of mission objectives have significantly increased government agencies' workloads, budgets have constrained by being either slashed or remaining flat over the years. Additionally, rising operational costs to maintain legacy IT systems have increased the pressure for government agencies to find opportunities where they can trim spending, improve efficiency, and increase productivity.

Firstly, many mission-critical systems and IT processes across the government have not been modernized in many years, or in some cases, even decades. Historically, these legacy systems and processes kept pace with the changing mission. Since legacy IT is outdated and not built to meet modern-day needs in a fast-paced environment, they're neither cost justifiable nor sustainable to operate and maintain. Secondly, legacy IT also poses operational risks as many systems are too rigid to adapt to current or expected mission requirements. Additionally, outdated systems pose serious security risks as they are unable to utilize latest security leading practices, leaving critical systems vulnerable to damaging cyberattacks. Thirdly, with over three-quarters of the federal IT budget being spent on operation and maintenance activities, it leaves very little funding for innovation and modernization. As a result, federal CIOs are forced to find creative solutions to optimize their available resources.

Adding to legislative pressures, internal and external customers are increasing their demand for seamless digital experiences that provides system connectivity across platforms. Customers expect their federal IT experience to be accessible, responsive, reliable, and secure and to either match or exceed their digital experiences in the commercial or private sector. With the rising expectations for federal services, the private sector is investing to adopt innovative and newer technologies to meet customer demands.



Approach for IT Modernization

Multi-dimensional problems require multi-dimensional solutions. Digital government is a multi-dimensional problem that cannot be solved by installing one solution. The search for that one comprehensive solution which will meet the expectations of all stakeholders usually ends in futility, but more importantly, that mindset can result in years of stalled programs where incremental progress could have been possible.

Government agencies should seek to evolve into kinetic enterprises with the dexterity and vision to overcome operational inertia and the ability to thrive in an ever-changing environment. As the shifting legislative demands require the federal government to be agile and flexible, the federal government should consider adopting a modular design that enables the development of composable systems through componentization of applications and inter-component integration via APIs. This can help federal government tackle the pressure of doing more with less and enhance inter-agency collaboration.

Interoperability is a unified and uniformed digital experience for customers, and security is integral to this experience. It is one window, one website, one app that answers all application questions. So, when agencies exchange information between each other the data must travel from one system to another through connected interfaces. And whenever data travels across agency boundaries, strong security protocols for the transfer of data must be practiced. As such, API based connectivity models provide an ideal way for enforcing access controls and implementing abstraction (e.g., black-box models) for secured interoperability.

Microservices and APIs—An Emerging Solution

Microservices architecture and API-led connectivity have recently emerged as potential solutions to legacy application modernization. It provides a digital transformation path that allows seamless transition to target architecture from legacy architecture. Simply put, it can enable government agencies to take on the digital transformation journey at their own pace without having to make a full-fledged upfront commitment to the future state architecture. In the journey to embrace digital transformation, the ability to be agile and flexible will be key to success.

What are Microservices and APIs?

Before understanding what microservices architecture are and the how APIs play a role, it is important to look at the evolution of application architecture.

From Monolithic to Modular Architectures

In the past monolithic structures were composed of dependent layers with connectivity through in-process calls. They were developed and deployed as large, single units making them harder to scale. Additionally, they offered very limited reusability of components. The next stage in the evolution was the development of service-oriented architectures to meet the challenge of reusability and integration. The emphasis was on either developing dependent functions or services that provided exposure to the enterprise. The intent was to simplify integration into the backend systems through reusable functions, thereby saving costs.

Although the service-oriented architectures were reusable and well-integrated, they were still composed of dependent systems meaning the systems were rigid and not highly scalable. Consequently, the service-oriented architecture gave way to the modular architecture, where the systems are composed of independently maintained and deployable components/services accessed with the help of APIs.

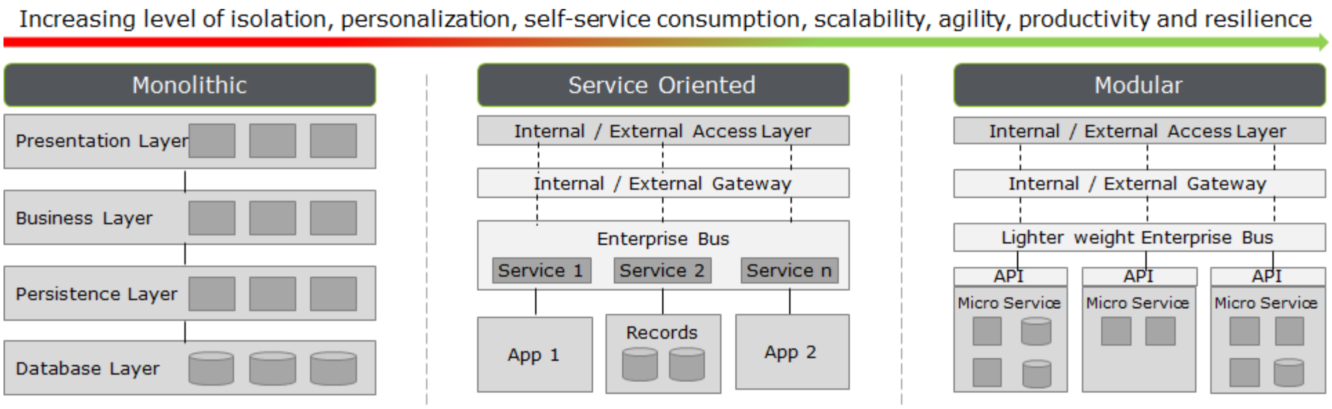

Increasing level of isolation, personalization, self-service consumption, scalability, agility, productivity and resilience
Monolithic
Presentation Layer
Business Layer
Persistence Layer
Database Layer
Service Oriented
Internal / External Access Layer
Internal / External Gateway
Enterprise Bus
Service 1
Service 2
Service n
App 1
Records
App 2
Modular
Internal / External Access Layer
Internal / External Gateway
Lighter weight Enterprise Bus
API
Micro Service
API
Micro Service
API
Micro Service

Figure 1: Evolution of Application Architecture

Comparing Monoliths to Microservices

Microservices architectural style can be best understood by comparing it to the monolithic approach. A monolithic application is built around the concept of functional layering, such as presentation, business, and data layers. Alternatively, a microservices-based application is composed of several small, loosely coupled granular services where functionality is divided into services.

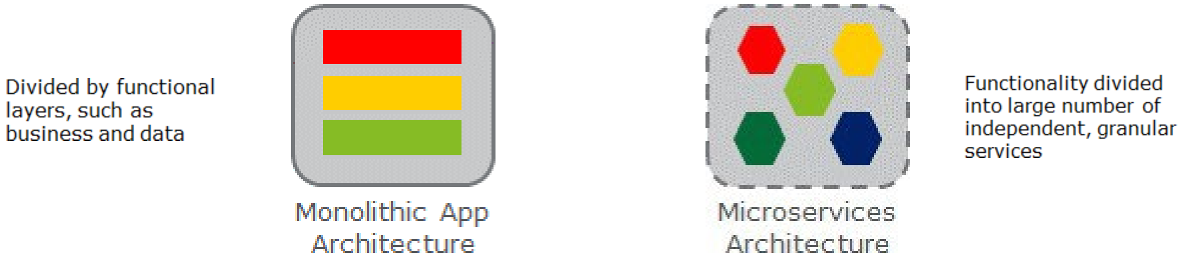

Divided by functional layers, such as business and data
Monolithic App Architecture
Microservices Architecture
Functionality divided into large number of independent, granular services

Figure 2: Monolith vs Microservices—Functionality Decomposition

Microservices offer efficient scaling of systems by allowing components/services that cause performance bottlenecks to be scaled instead of a wholesale scaling of the entire application. For example, if a microservice provided operational supply data from one component of a transactional enterprise resource planning (ERP) system, then effort can be focused on scaling the service rather than the entire ERP. More importantly, microservices architecture allows for development of fault-tolerant designs which prevents an entire application from going down because of a single component failure. This is especially relevant for mission critical functions such as maritime maintenance that must perform consistently with high availability.

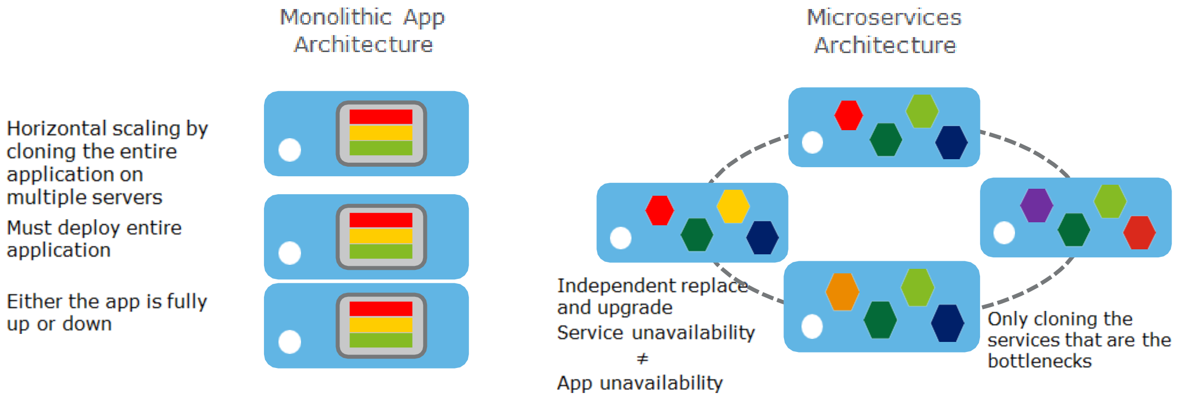

Monolithic App Architecture
Microservices Architecture
Horizontal scaling by cloning the entire application on multiple servers
Must deploy entire application
Either the app is fully up or down
Independent replace and upgrade
Service unavailability
≠
App unavailability
Only cloning the services that are the bottlenecks

Figure 3: Monolith vs Microservices—Deployment, Availability and Scaling Differences

Because microservices are independently developed and deployed, teams no longer need to wait for long and tedious test cycles for application upgrades, thereby reducing system development lifecycle. Moreover, microservices take advantage of the infrastructure automation techniques that have enormously evolved over the last few years. Teams that build microservices based applications make extensive use of automation techniques to streamline processes and reduce deployment and configuration overhead which allows for continuous delivery. Simply put, development teams can use microservices in conjunction with infrastructure automation techniques to break free of lengthy, cumbersome development lifecycles and introduce innovations at a rapid pace.

Microservices are built on the principle of loose coupling which helps reduce dependencies between the services and their consumers. A microservice provides a set business function expressed as a contract and delivers business-oriented APIs. Since consumers of the service are receiving a contractually defined output, they are no longer impacted by upstream changes to the microservice. This result enables service owners to change upstream data sources, such as either modifying or replacing underlying systems of record or service compositions, without any downstream impact.

Lastly, cross-functional teams are necessary for microservices development as the services are designed to be autonomous. This is a stark contrast to the traditional siloed approach where UI developers work on the application front-end, backend developers work on the core business functionality, and database administrators work on managing systems of record. The cross-functional approach helps facilitate a strong mission-oriented "You Build it, You Run it" culture—which leads to an unbounded IT working in concert to deliver the mission objectives.

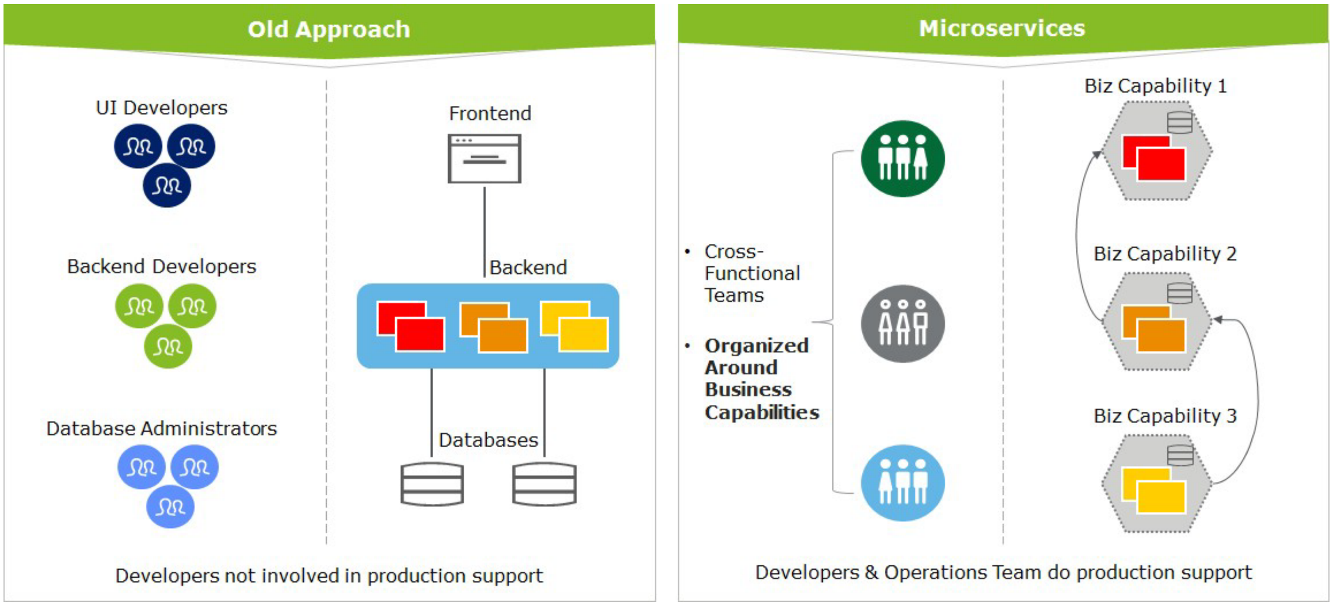


Figure 4: Software Development Teams—Traditional Siloed vs Cross-Functional

Impact and Benefits of Microservices and APIs

Risk Reduction through Agile and Flexible Architecture

Microservices can help lower the operational risks to the organization by providing an agile and flexible architecture to meet the shifting mission needs. The automation culture, introduced as part of the microservices architecture, is designed to overhaul the outdated processes and bring efficiency to the system development lifecycle. Furthermore, the new architecture style will increase system availability and reliability through fault-tolerant designs and reduce security risks by enabling the systems to deploy current security leading practices. This style can also enable more rapid and flexible responses to either evolving cybersecurity or legislative mandates, thus increasing security and decreasing the risk of oversight noncompliance

Composable Enterprise

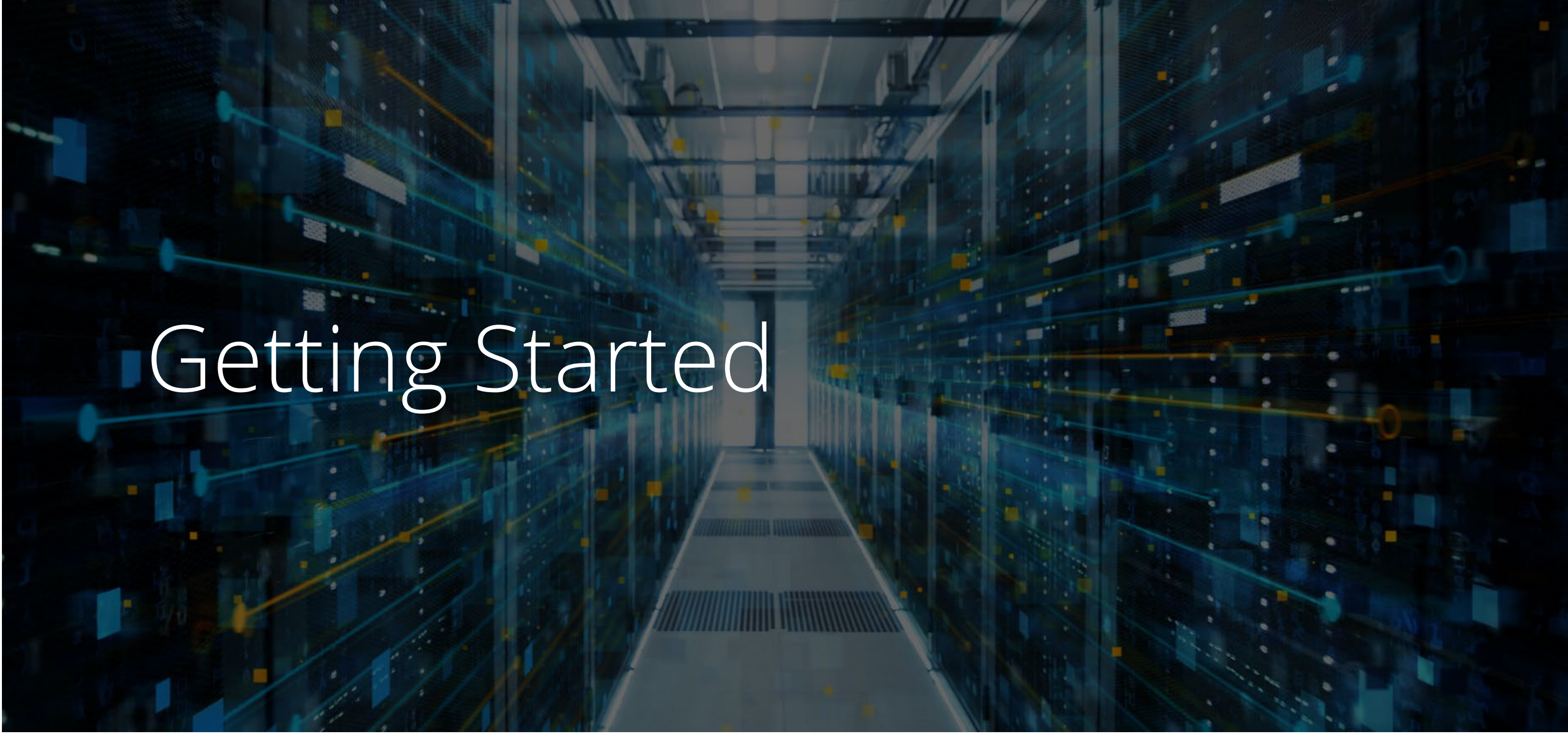
Microservices offer the benefit of component/service reusability, which will not only reduce redundant capabilities thereby reducing development and maintenance costs, but also provide composability to the enterprise. Currently, federal agencies face the challenge of non-standard business processes and closed systems which limit interoperability and connectivity. By developing business capabilities in the form of services, agencies can possess modular digital assets that can be used in different business contexts providing unprecedented value to the clients. At the same time, microservices unlock the power of data analytics as IT becomes increasingly more decentralized.

Application Modernization with Minimum Disruption

Microservices can help accomplish this by exposing connectivity to the currently siloed systems and introducing newer technologies to the presentation layer at the outset without changing the systems of record. The backend systems and associated services can be changed during the later phases of the modernization. This can reduce risk and user disruption associated with large-scale changes to systems of record.

Unbounded IT

Microservices architecture breaks the functional silos which exists within the traditional software development teams. As microservices require organizations to form cross-functional teams for software development, this prompts a change in teams' mindset as they see themselves supporting products instead of projects. This shift in focus can lead to increased collaboration among teams as well as adoption of innovative technologies and processes. Additionally, the DevSecOps and Automation culture, a pre-requisite for microservices architecture, helps drive efficiencies and reduce system development lifecycle.

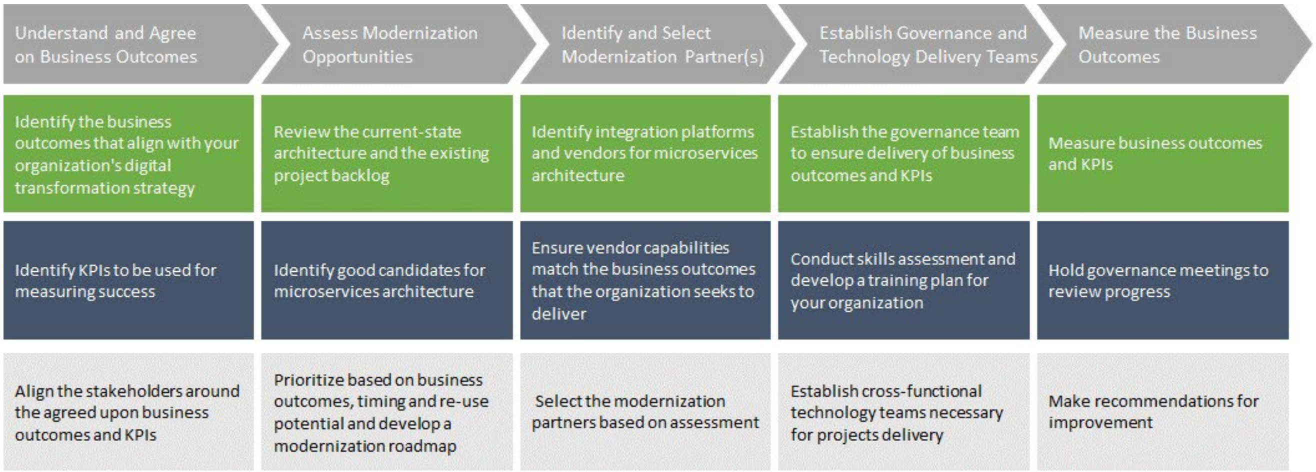


Getting Started

To successfully take on the microservices and APIs journey, government agencies should look to adopt an outcome-based delivery approach, where the organization begins with an understanding of what digital transformation means to them and what it aims to accomplish by embarking on this journey.

IT transformation can be done in small increments if there is a robust integration strategy in place. Firstly, identifying outcomes and KPIs at the outset can help agencies build a value-driven, agile culture within their organizations. This approach helps engage all the stakeholders from the very beginning and align them around the same goals, which makes it easier for the organization to track progress and measure overall success. Identifying the business outcomes upfront also provides understanding of the organization's direction, establishes key stakeholders, and clearly communicates KPIs for measuring success.

Next, organizations assess their current state architecture and project backlog to identify good candidates for modernization. Projects should be prioritized and selected based on business outcomes, timing, and re-use potential to ensure maximum value is delivered quickly to the organization to accelerate digital transformation. Vendors should be selected based on the alignment of their capabilities with the business outcomes and the organization's overall digital transformation strategy. It is also vital to establish governance teams, such as steering committees, to that ensure business outcomes are delivered in a timely fashion. Cross-functional delivery teams, a hallmark of modular application development, should be established to enable agile development and delivery. Lastly, and most importantly, KPIs and business outcomes should be measured, reviewed, and refined to ensure sustainable digital transformation through a culture of agility and innovation.



Understand and Agree on Business Outcomes	Assess Modernization Opportunities	Identify and Select Modernization Partner(s)	Establish Governance and Technology Delivery Teams	Measure the Business Outcomes
Identify the business outcomes that align with your organization's digital transformation strategy	Review the current-state architecture and the existing project backlog	Identify integration platforms and vendors for microservices architecture	Establish the governance team to ensure delivery of business outcomes and KPIs	Measure business outcomes and KPIs
Identify KPIs to be used for measuring success	Identify good candidates for microservices architecture	Ensure vendor capabilities match the business outcomes that the organization seeks to deliver	Conduct skills assessment and develop a training plan for your organization	Hold governance meetings to review progress
Align the stakeholders around the agreed upon business outcomes and KPIs	Prioritize based on business outcomes, timing and re-use potential and develop a modernization roadmap	Select the modernization partners based on assessment	Establish cross-functional technology teams necessary for projects delivery	Make recommendations for improvement

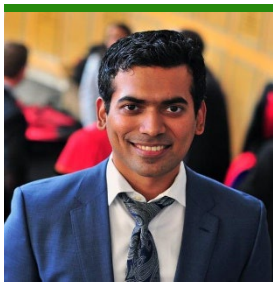
Figure 5: Getting Started: An approach to adopting Microservices and API architecture

Contributors

Oniel Cross
Principal
Public Sector, Cloud Engineering
Deloitte Consulting LLP
Tel: +1.571.814.7265
Email: ocross@deloitte.com

Ed DelBeccaro
Senior Manager
Public Sector, Systems Modernization
Deloitte Consulting LLP
Tel: +1.571.858.1112
Email: edelbeccaro@deloitte.com

Samiran Ghosh
Senior Manager
Public Sector, Cloud Engineering
Deloitte Consulting LLP
Tel: +1.215.446.4332
Email: samighosh@deloitte.com



Akhter Khan
Manager
Public Sector, Cloud Engineering
Deloitte Consulting LLP
Tel: +571.249.7376
Email: akhkhan@deloitte.com

Muhammad Khan
Manager
Public Sector, Cloud Engineering
Deloitte Consulting LLP
Tel: +1.571.882.7212
Email: muhakhan@deloitte.com

Adina Barnett
Analyst
Public Sector, Cloud Engineering
Deloitte Consulting LLP
Tel: +1.703.340.3848
Email: adbarnett@deloitte.com

References

1. Bill Briggs and Craig Hodgetts, *Tech Trends 2017—Introduction: The kinetic enterprise*, Deloitte University Press, February 7, 2017, https://dupress.deloitte.com/dup-us-en/focus/tech- trends/2017/tech-trends-introduction.html

2. Atilla Terzioglu, Martin Kamen, Tim Boehm, and Anthony Stephan, *IT unbounded: The business potential of IT transformation*, Deloitte University Press, February 7, 2017, https://dupress.deloitte.com/dup-us-en/focus/tech-trends/2017/it-transformation-business- potential.html

3. U.S. GAO, “Federal Agencies Need to Address Aging Legacy Systems”, May 25, 2016, https://www.gao.gov/products/GAO-16-696T

4. Microsoft, “Why a microservices approach to building applications?”, July 2, 2017, https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-overview-microservices

5. James Lewis and Martin Fowler, “Microservices”, March 25, 2014, https://martinfowler.com/articles/microservices.html

6. MuleSoft, “Best Practices for Microservices” https://www.mulesoft.com/ty/wp/best-practices-microservices

7. Sam Newman, “Building Microservices: Designing Fine-Grained Systems”, O'Reilly Media, February 20, 2015

Deloitte.