Deloitte.

Breaking down Monolithic Applications for the Future of Cloud

Modernizing monolithic applications to cloud-native represents a significant technological and cultural shift for most enterprises. Changing customer expectations, the increasing need for agile enterprises, and the ever-expanding Big Data and cloud-native technologies have led to a rise in application modernization. Around this trend, great progress has been made in modernization approaches and tools that enable large and complex modernizations to be implemented in a low-risk and highly automated manner. Deloitte has invested in products that specialize in modernizing legacy applications through a high level of automation with a strong focus on modernization and refactoring – beyond simple migration or re-platforming.

However, different modernization approaches have various benefits, risks and impacts, which are outlined in this paper. What is the best modernization approach for your enterprise? ❯

Today's data centers are full of monolithic applications that are based on outdated technologies, but are critical for day-to-day operations. After decades of development, fine-tuning and evolution, these applications – and the optimized processes they support – represent enormous business value. Finding the right approach for modernizing these complex monoliths is the most significant challenge that businesses can face, particularly given the additional complexity hiding within. Even the prospect of starting the process seems daunting.

With everything from decades-old mainframe processors to more recent systems running on Java or .NET, what these underlying technologies all have in common is

that they don't fit very well into today's IT landscape: They are structurally complex and monolithic, often lacking documentation and make it extremely difficult to migrate to the cloud or even maintaining reliable operations. At the same time, businesses are facing massive changes in the market from competitive pressures, resource shortages and new technologies. It is vital for businesses to respond to these dramatically changed customer requirements and expectations and to improve time-to-market. But to the applications, modifying or adding new functionality is often difficult, slow, and expensive. There is no one-size-fits-all solution. The right approach for a particular situation will depend on the specific business needs

and business cases. No sense in jumping blindly into a modernization project without first understanding the potential technical debt, complexity, dependencies, cost of ownership and risk.

Three pathways to a future-proof application landscape

There are three primary paths organizations can take to ensure IT systems are ready for the future and possibly the cloud:

1. Replace/Rewrite
2. Evolve
3. Transform + Modernize.

These three approaches differ not only in their potential to change certain technical aspects of the application, but also in the effort required, the risk, and the impact they have on both the business and IT.
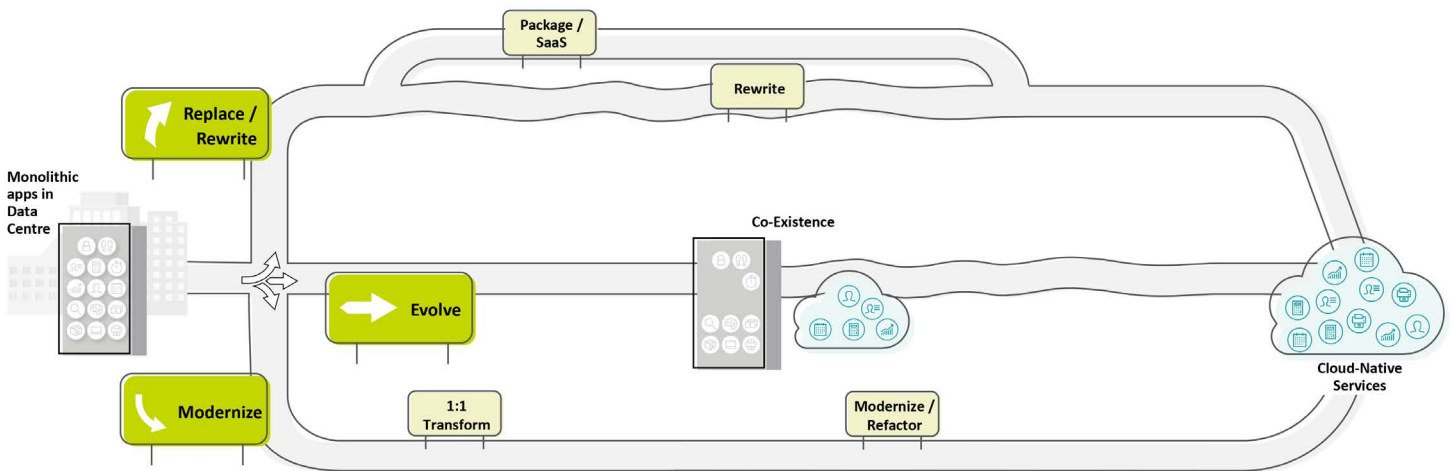
There is no one-size-fits-all solution. The right approach for an enterprise will depend on their particular business needs and business case.

Pathway 1:
Replace/Rewrite

For decades, enterprises have taken the Replace/Rewrite approach to replace critical applications.

Everyone in IT knows precisely how this works. It starts with specifying the requirements for a make-or-buy analysis or for selecting a standard software, followed by a more in-depth needs analysis and design phase, as well as a prolonged implementation phase. The final step is to migrate the data to the new platform.

Sounds easy, right? But is it?
We have all heard stories of long, drawn-out implementations, cost overruns and never-ending customization efforts. For large systems complexity in this approach becomes hard to handle. Managing require-

ments and avoiding over-customization are key to steer clear of failure. However, under certain circumstances, it may be the right path for some. Replace/Rewrite could be a good fit for enterprises if:

- Organizations require generic functionality that is readily available as a COTS or SaaS solution without extensive customization.

- The system has limited complexity and is quick to implement.

| The benefits	The risks	Our impact and how to minimize the risks
Total flexibility: IT developers and system integrators love this approach because it gives them complete freedom to design a new solution tailored to the company's specifications. If implemented correctly and with strong requirements management, moving to a standard solution or a less complex custom solution can provide incredible value and, in some cases, improve current system functionality. In addition, a well-managed COTS/SaaS implementation can be an opportunity to drive standardization and simplify existing processes and future maintenance.	Without strong guidance and strict management, these projects tend to become more and more complex, take longer than planned and drive up costs. The majority of businesses underestimate the complexity of the process. We could cite countless examples of companies forced to keep both the new and old systems in operation when migration proves too complex, or the new system never becomes able to fully substitute the old one.	A number of recent advances in supporting tools that analyze existing code structures and dependencies have made these projects are much more feasible. Using our mining tool, we can perform a code-level analysis to gain a deep understanding of business functions executed by the code. In fact, code-mining allows the extraction and documentation of business rules and requirements directly from the old system.

Pathway 2:
Evolve – also known as "The Strangler Pattern"

This has become a rather popular approach in recent years. Instead of addressing the entire monolith at once, IT teams build a target landscape next to the legacy system to allow partial and continuous offloads.

In many cases, bi-directional-synchronized data synchronization is required, which may be implemented using methods such as change data capture, event streaming and/or database native mechanisms.

The benefits

As a more gradual approach, companies use Evolve to avoid the risks associated with a rapid transition to a new platform. The migration teams can gather first experiences on the future platform, learn from them and shift functionality over time whilst maintaining a continuing production environment. It allows them to focus their efforts where modernization makes the most impact and explore the future architectures in laboratory conditions instead of instantly shifting complex networks of applications. Applying domain-driven design principles to the monolith enables a business to decouple legacy applications into independently scalable services managed by a DevOps team.

The risks

Companies following this strategic, more gradual approach – run the risk of building an entirely new IT landscape that may never be able to fully replace the legacy. The approach also needs an architecture of coexistence which allows legacy code, hybrid interactions and modern native functionality to be run in parallel and throughout the iterative modernization. Overall, this modernization approach takes more time, and is usually more costly and requires consistently defined business, data, and application patterns to mitigate distractions and manage changing requirements.

Our impact and how to minimize the risks

Once the secondary landscape is in place, we use a "strangler pattern" approach to re-implement, re-architect, or migrate individual features of the monolith to the future platform. This step-by-step approach limits implementation complexity and avoids the associated risk. Gradually shifting functionality from the old to the new platform, hiding the change behind a common interface, effectively "strangles" the monolith and bit by bit reduces its functional footprint. Using our Large Scale Code Refactoring approach, we can recycle existing code to create new modules based on existing business logic to speed up the process.
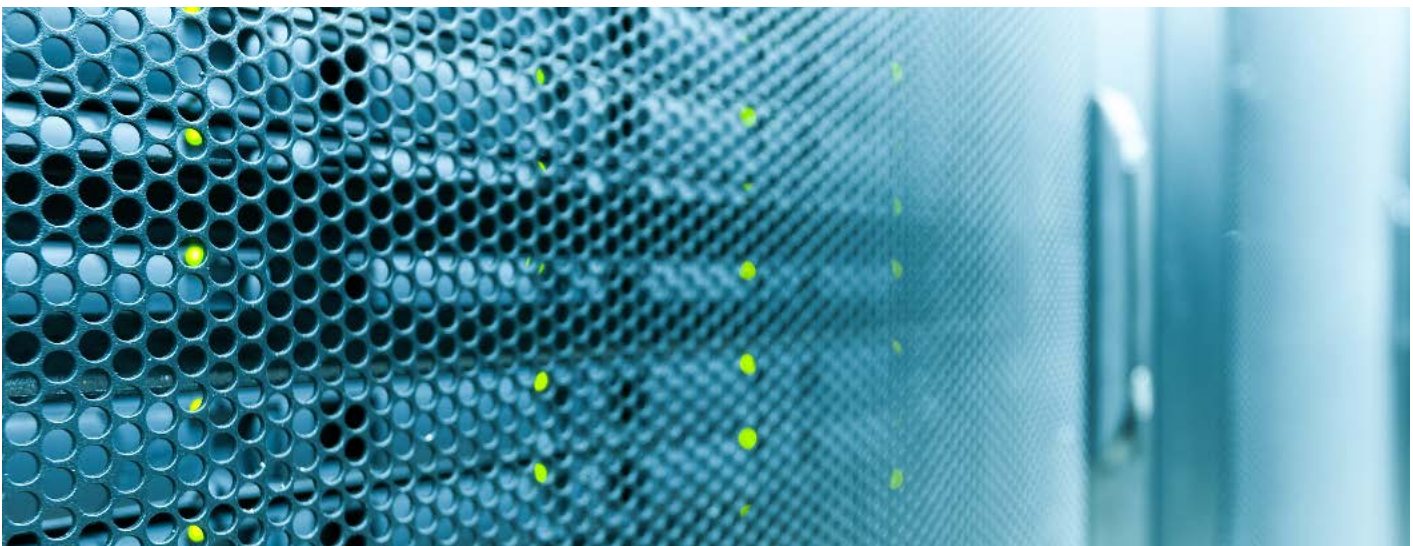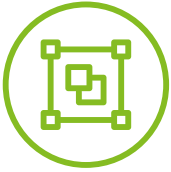
Pathway 3:
Transform/Modernize

The first half of this approach, "Transform", has been an established practice for many years: With the help of advanced code transformation tools, legacy applications can be converted to run on new underlying technology. This is most commonly used to transform mainframe or mid-range systems' applications written in COBOL (or any of the other legacy languages) into Java, .NET/C# or other modern technologies (e.g. low-code).

This phase addresses some of the problems with legacy applications, but not all of them. It makes companies less dependent on hard-to-find talent with mainframe skills, allows teams to scale while also continuing to securely operate, and maintain the platform. Removing the dependency on legacy technologies typically results in significant cost savings that can help fund the modernization. All of this is done with minimal impact on the business and with very low risk since the majority of the transformation process is automated.

What the Transform phase does not address is the structural deficiencies of the applications themselves. This is where the "Modernize" phase comes in: Using our advanced Large Scale Code Refactoring approach, we can rapidly decompose existing monolithic applications into multiple self-contained services. Starting from the modern code produced during the Transform phase, we use our refactoring tools to automatically break down an application based on its internal structure, and either split it into services with well-defined APIs, introduce higher levels of abstraction, object-orientation and inheritance into the code base, or make the most of state-of-the-art architectures and cloud services.

The benefits	The risks	Our impact and how to minimize the risks
By shifting to a modern technology platform, the Transform phase can quickly satisfy a business case and tackle the most urgent problems first and fund the modernization. Our fully automated code refactoring and data migration capability transforms legacy code into modern development languages and enables deployment in the cloud while avoiding business disruption and reducing risks associated with the complexities of a legacy ecosystem.	The Transform phase is inherently low risk as it is automated to a large degree. However, if the code base is incomplete or relies heavily on third-party products, this approach may not be the best fit or could even fail. An initial consultation with a comprehensive code discovery can rule out this risk and develop a suitable mitigation strategy.	This approach differentiates between Technology Change and Modernization. It minimizes the risk of the transformation and accelerates the migration to a new technology platform. With automated code transformation as a first step, enterprises can cut operating costs by as much as 60 to 85% and amortize the transformation in under three years.
It is often the smallest step towards an organization's goal to migrate the existing system with as little change to business processes as possible first and modernization after the successful transformation.		With our modernization capability, we leverage unique tools and accelerators to decompose code and data into independent services and databases. Modernization enables organizations to take advantage of modern cloud-services, dynamic workload and management and elasticity in the cloud, allowing a transition to modern DevOps practices. We focus not only on the technological modernization and introduce state-of-theart DevOps and agile concepts, methods, and tools early in the process, but also ensure that the existing development teams are transitioning to the new way of working.
The Modernize phase involves the structural transformation of a system into a true future-proof services-based architecture that takes full advantage of the cloud.		
This path is the fastest with the best business case and lowest risk. The savings from the transformation can be used to fund the modernization journey and modernization effort is targeted where it produces the greatest business impact.		

Our fully automated code refactoring and data migration capability transforms legacy code into modern development languages and enables deployment on cloud while avoiding business disruption and reducing risks associated with the complexities of a legacy ecosystem.

Organizations should choose the application modernization approach that will have the greatest impact on and create the most value for their business. Before they embark on their migration journey, it is important to evaluate which of these options will achieve the desired effect for which applications.

Deloitte's Center of Excellence for Application Modernization and Migration regularly supports clients with all three approaches and helps them to find the right approach for their business. Our deep application modernization expertise and our own innoWake tool suite help to reduce the complexities and accelerates an end-to-end modernization journey. This includes also a manageable upskilling of existing application development teams to equip them with all the innovative features of the cloud.

Where to start?

Speak to our modernization team. Often the right first step will be an assessment of the code base using our Deloitte innoWake Discovery. With our tooling we explore legacy code and supporting technologies. Understanding the structure of the code and its dependencies is critical for planning a smooth migration and modernization journey. With Discovery we eliminate surprises and guess work and create the factual basis for planning an efficient, low-risk modernization.

Contacts

Dr. Karsten Ballüder
Partner
CoE for Application
Modernization & Migration
Tel: +49 69 9713 7322
kballueder@deloitte.de

Elena Kasemir
Senior Manager
CoE for Application
Modernization & Migration
Tel: +49 211 8772 4508
ekasemir@deloitte.de

Deloitte.