zensar

Micro is the new Mega Transformation of **Monolith to Microservices Architecture**

White Paper

Table of Contents

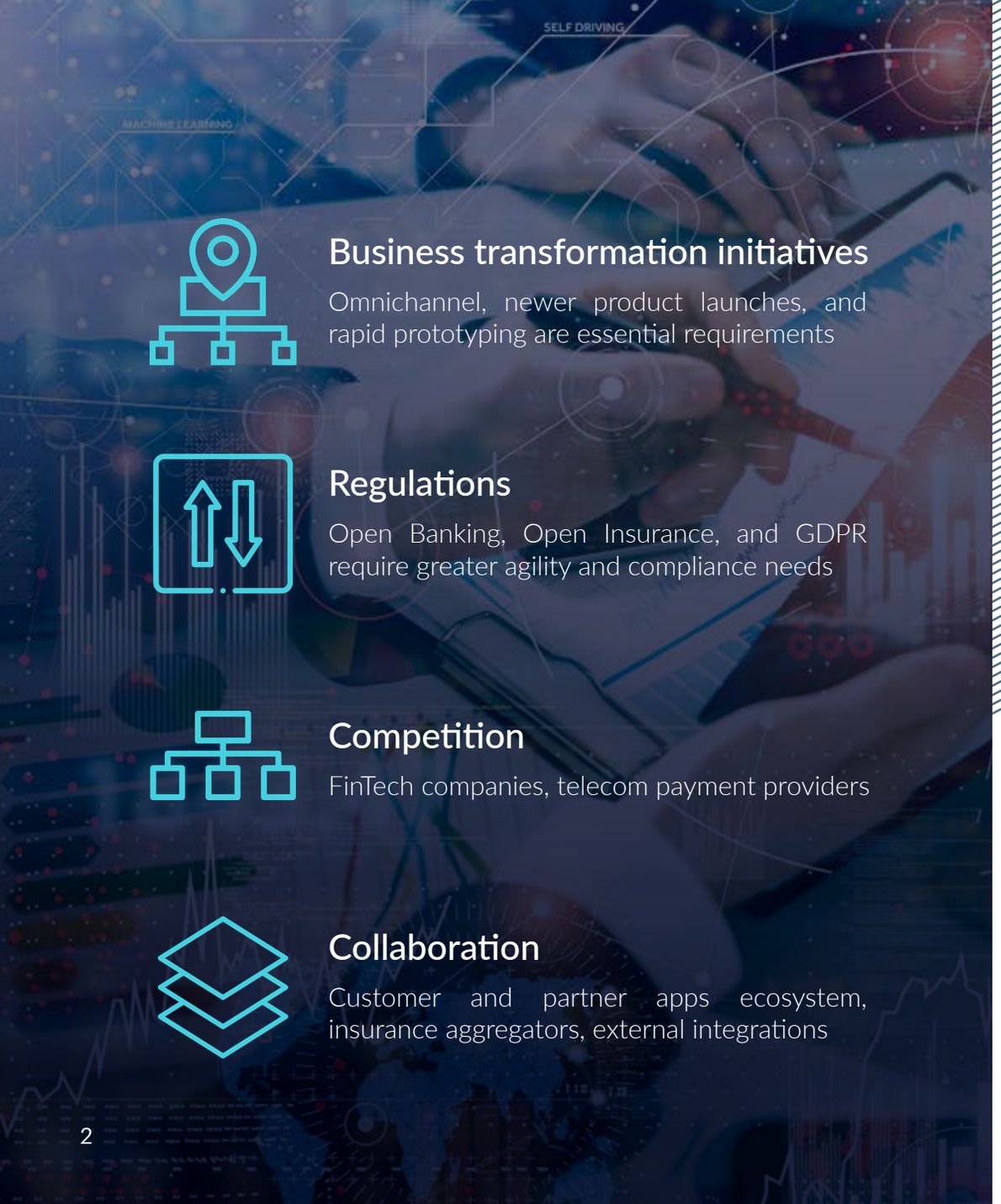
Modernization of legacy systems at financial enterprises

The global pandemic is forcing rapid change, and the ability for organizations to transform into digital-first entities at speed is the key. Many legacy monolith environments like mainframe systems across financial industries are unprepared to handle massive shifts in volumes and are undergoing significant cost pressures.

Gartner predicts that by 2023, 40 percent of professional workers will expect orchestrated business application experiences and capabilities, just like they do with their music streaming experience. “The human desire to have a work environment which is similar to their personal environment continues to rise.”

Many organizations struggle to build a strategy around application modernization and are often unsure of the modernization approach they need to undertake, whether refactoring, rehosting, or otherwise. Simply hosting applications on the cloud, which is the extent of application modernization strategy for many organizations, will not deliver what Gartner is predicting. Instead, organizations need to take an in-depth strategic approach.

Organizations approaching application modernization based on vague promises of the benefits of cloud and productivity improvements will find themselves in a similar position in a few years with a legacy environment that no longer supports the organization’s competitive position.



Business transformation initiatives

Omnichannel, newer product launches, and rapid prototyping are essential requirements

Regulations

Open Banking, Open Insurance, and GDPR require greater agility and compliance needs

Competition

FinTech companies, telecom payment providers

Collaboration

Customer and partner apps ecosystem, insurance aggregators, external integrations

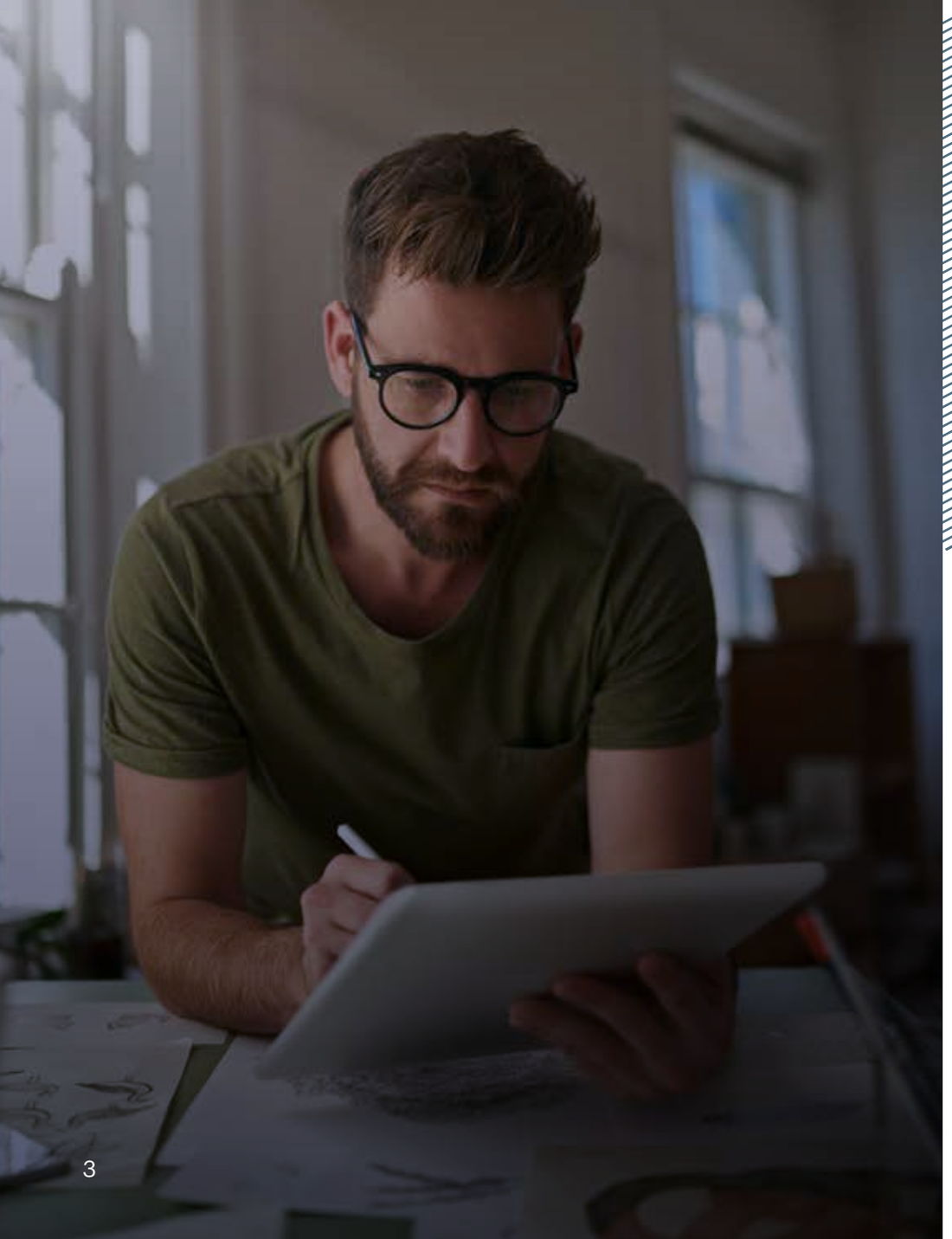
Nowhere is this approach more pronounced than the **Financial Services** industry. Even though there may hardly be any change in the core system functionality, customer interfaces churn faster and faster, like a whirlpool.

Even newer **technologies and platforms** usher in the need for modernization.

The core functionality in this industry, as we all know, is notoriously encapsulated in decades-old but highly stable monolith systems. However, the imperatives listed above accentuate the need to make these core capabilities available in a form that allows us to compose modern applications.

Breaking these **Monoliths into Microservices Architecture (MSA)** based modern applications seems to be the only way out. There is hardly any conversation that happens without bringing MSA into the modernization strategy.

But is MSA the panacea, and, if so, how do we go about modernizing a monolith?



Are you making the most of the Monolith?

Most of us believe that monolith is an outdated word in today's IT world, and we must get rid of it — but the reality is different. A monolith is a challenge when it continues to grow without much thought process or creates too many cross-dependencies for a single business functionality.

If the core logic resides inside a well-designed monolith, doesn't change frequently, and causes no performance issues, then a better strategy would be to let it provide the functionality uninterrupted. However, one must consider building additional features outside the boundaries of this application.

For example, an insurance company may want to offer a better self-servicing option by allowing customers to change their beneficiary allocations through multiple channels like mobile, web, etc. However, the closed nature of the legacy policy administration system may not allow this feature. Hence, we can consider creating an independent set of microservices that accept these channels' request, validate and authorize the same as necessary. Only then the request is sent to the core system where a bot can make the entry, thus providing the desired end-to-end automation.

This allows companies to make the most of their monolith and hence is a much more economical option. It provides complete flexibility to effect any change in the way the organization wants to interact with its customers, allows more channels to be added at will, and, most importantly, it is relatively risk-free.

Should companies migrate their Monolith to Microservices Architecture (MSA)?

MSA is an architectural style that structures an application as a collection of loosely-coupled services to implement business functionalities that can communicate language agnostically. Within MSA, different best practices exist for data separation, avoiding macroservices, RESTful and HATEOAS, serverless computing (like AWS Lambda and Azure functions) and so on.

It is imperative to be conscious of the significant challenges that MSA tries to solve:

Business Agility

New version release (with minor updates or feature additions) of the monolith take forever and can be fraught with a high risk of rollback.

Scalability

Only specific modules of the monolith require scalability; however, these modules are not isolated enough to be scaled independently.

Autonomous Lifecycle

There is a strong business case to leverage the business functionality, residing inside the monolith, in new applications or the partner ecosystem. A particular service or module in the monolith has become a critical micro-application and has established the need for its autonomous lifecycle.

Resilience and Robustness

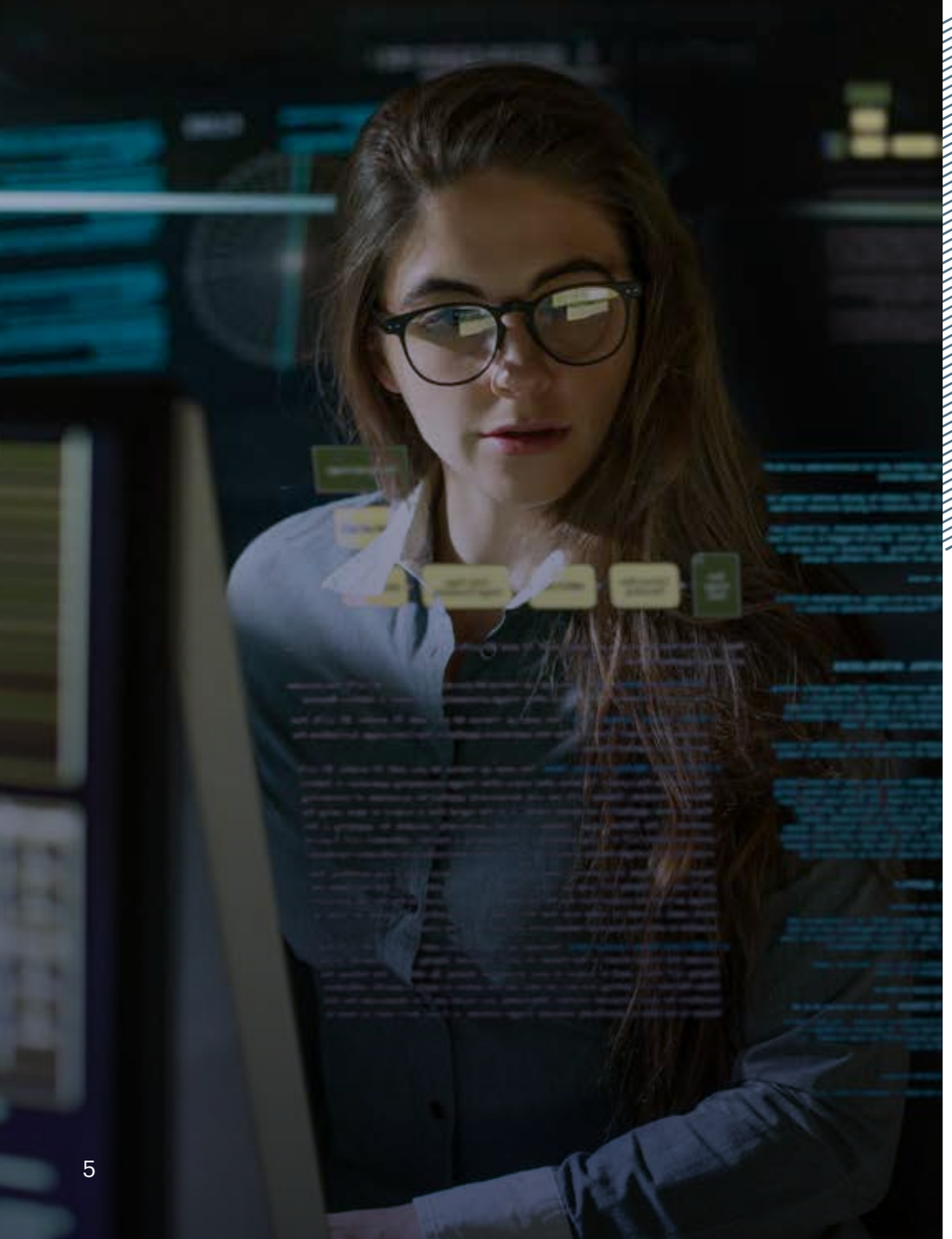
Improve the reliability of the application by allowing isolated failures and quick recovery.

Independent Testing

Regression testing cost for a small feature change is more than the development cost of the change or enhancement.

Ease of Operations

Understanding a big or an ever growing monolith becomes a challenge for any operations team. MSA distributes the knowledge among smaller teams.



While these make an excellent filter to shortlist the monolith for migration to MSA, it still requires analysis to move forward with the change. Let us start with an obvious example, such as a particular module that requires frequent change, probably due to new partners being added or regulations. However, bear in mind that MSA brings complexity unimagined. It suffices to say that even a well-defined monolith module can get decomposed into dozens of microservices. Then companies must deal with choreography and orchestration between microservices, eventual consistency, multiple data stores, and so on. The first step in this complex journey is to define the boundaries of the modules.

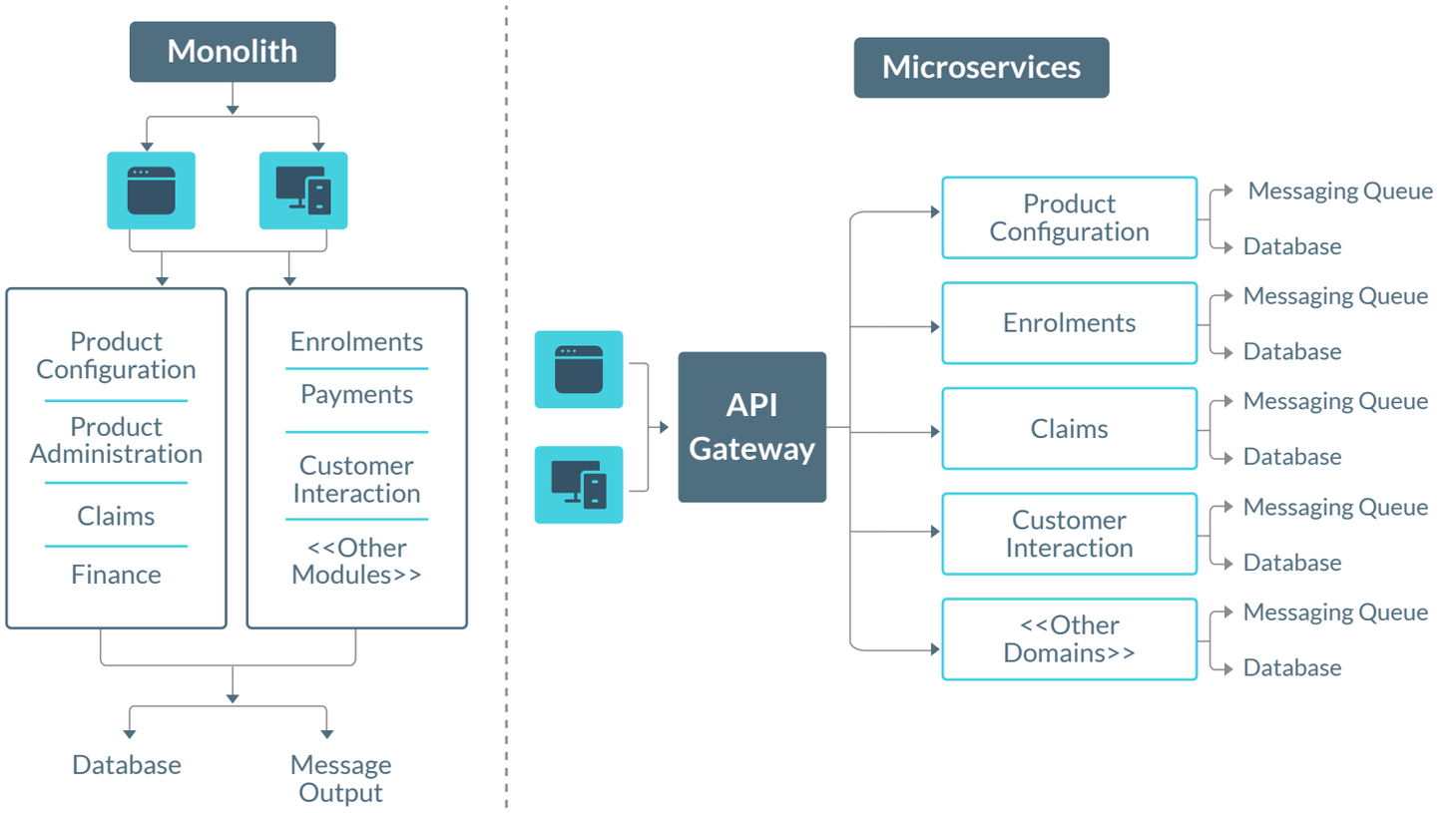
This step, in Zensar's view, is the most critical aspect to consider when moving to MSA. The ability to isolate modules into a suite of microservices by leveraging bounded context and removing cross-dependencies between modules makes a strong case for moving to MSA. This step is an absolute prerequisite to harness the benefits of MSA like faster time-to-market, faster prototyping, an autonomous lifecycle of the module, and even independent scalability.

What could be the strategy to move to MSA?

The MSA foundation includes developing an application composed of multiple feature services, each following the "Single Responsibility" principle. However, MSA is not just about splitting the monolith into a suite of small services. The monolith provides business capabilities, and the focus must remain the same when transforming the same into an MSA.

The whole of the monolith must be reimagined, and various business capabilities must be grouped in such a manner that they cater to one set of functionalities with clear boundaries. For example, in the policy administration system, all functionality related to enrollments can be thought of as one block or an independent application. Identifying these blocks is the starting point to break the monolith into an MSA-driven application and is usually referred to as Domain Driven Design (DDD) in the microservices space.

The diagram below captures the same at a high level:

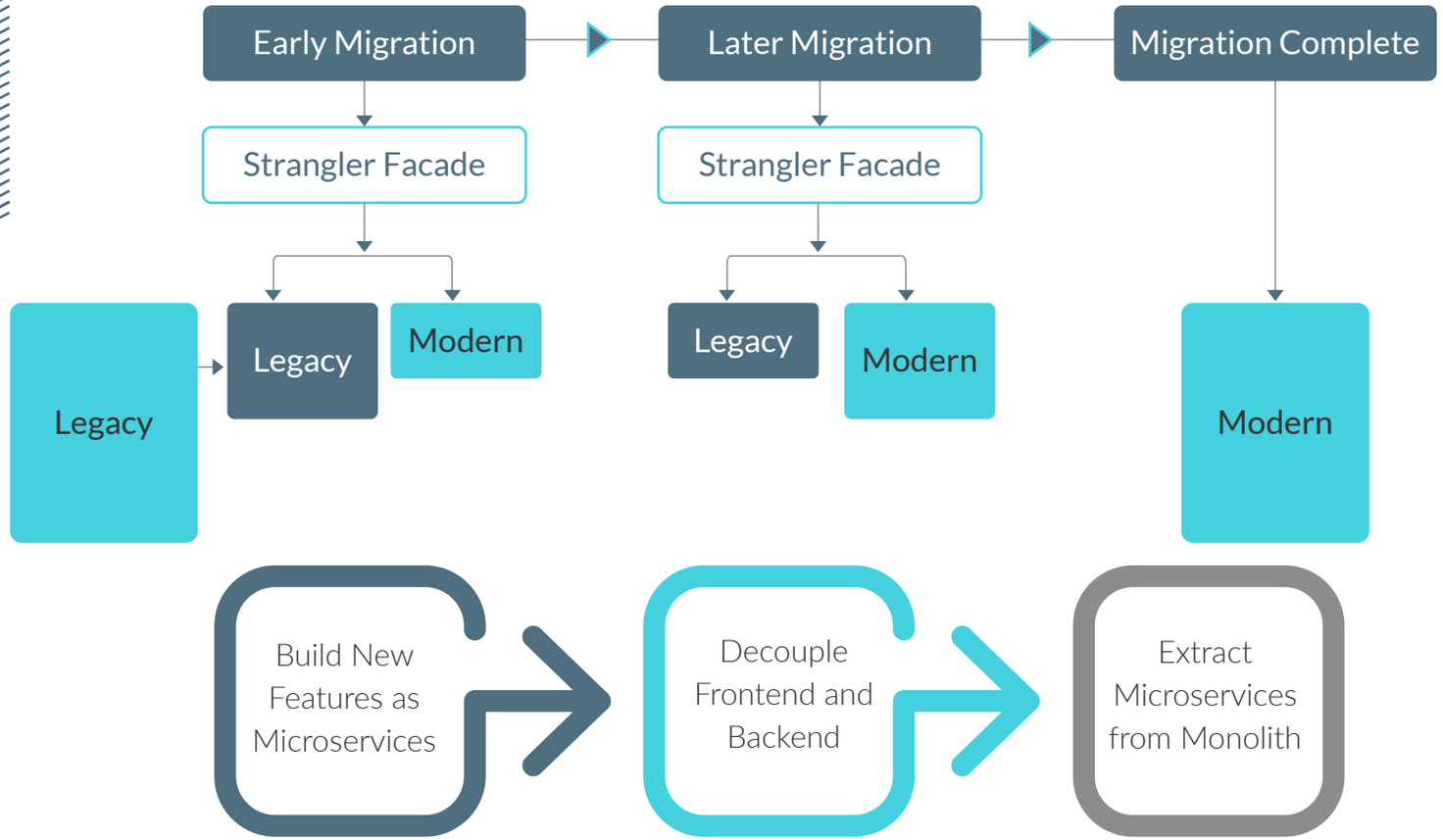


Not getting the boundaries right is one of the primary reasons for failure when organizations embark on the MSA journey.

The microservices design must consider the development and deployment challenges of a monolith. As mentioned above, services must be independently developed and deployed, preferably in an agile methodology, and leverage the best of DevOps practices.

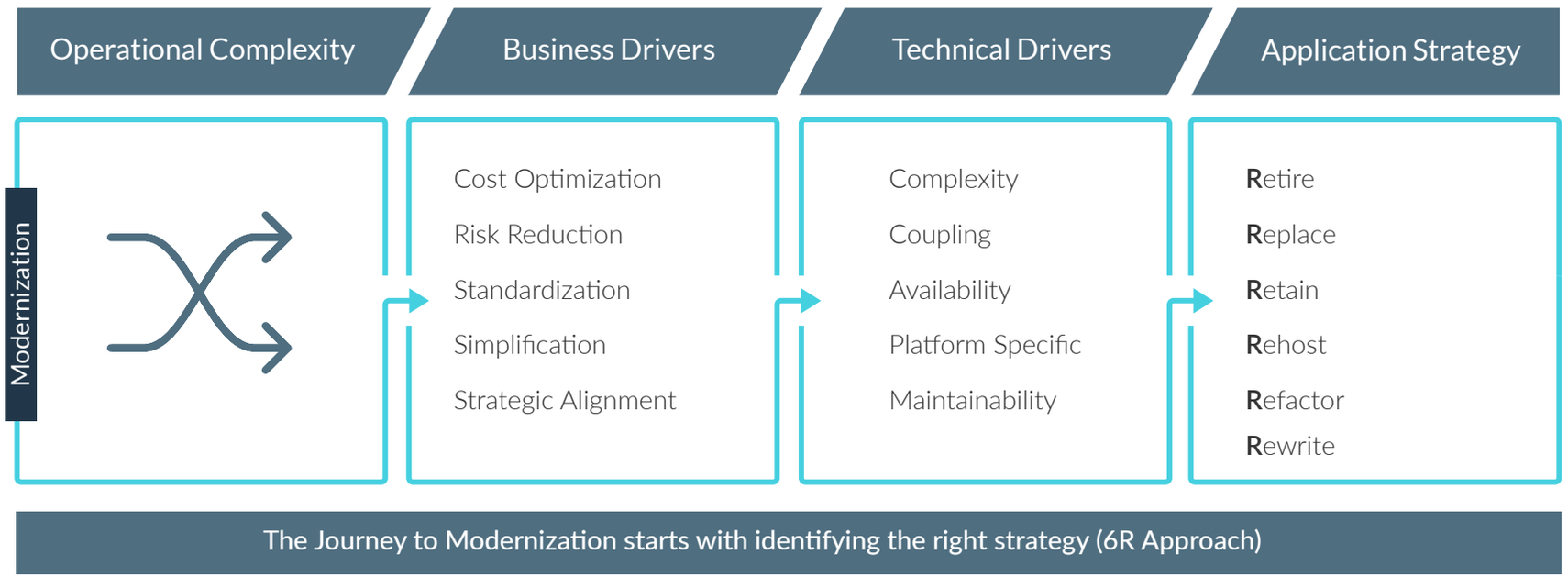
Therefore, migration requires carefully unraveling the monolith and converting it into MSA piece by piece. At the heart of this migration is the now clichéd “Strangler Pattern” approach.

The diagram below shows the process at a high level:



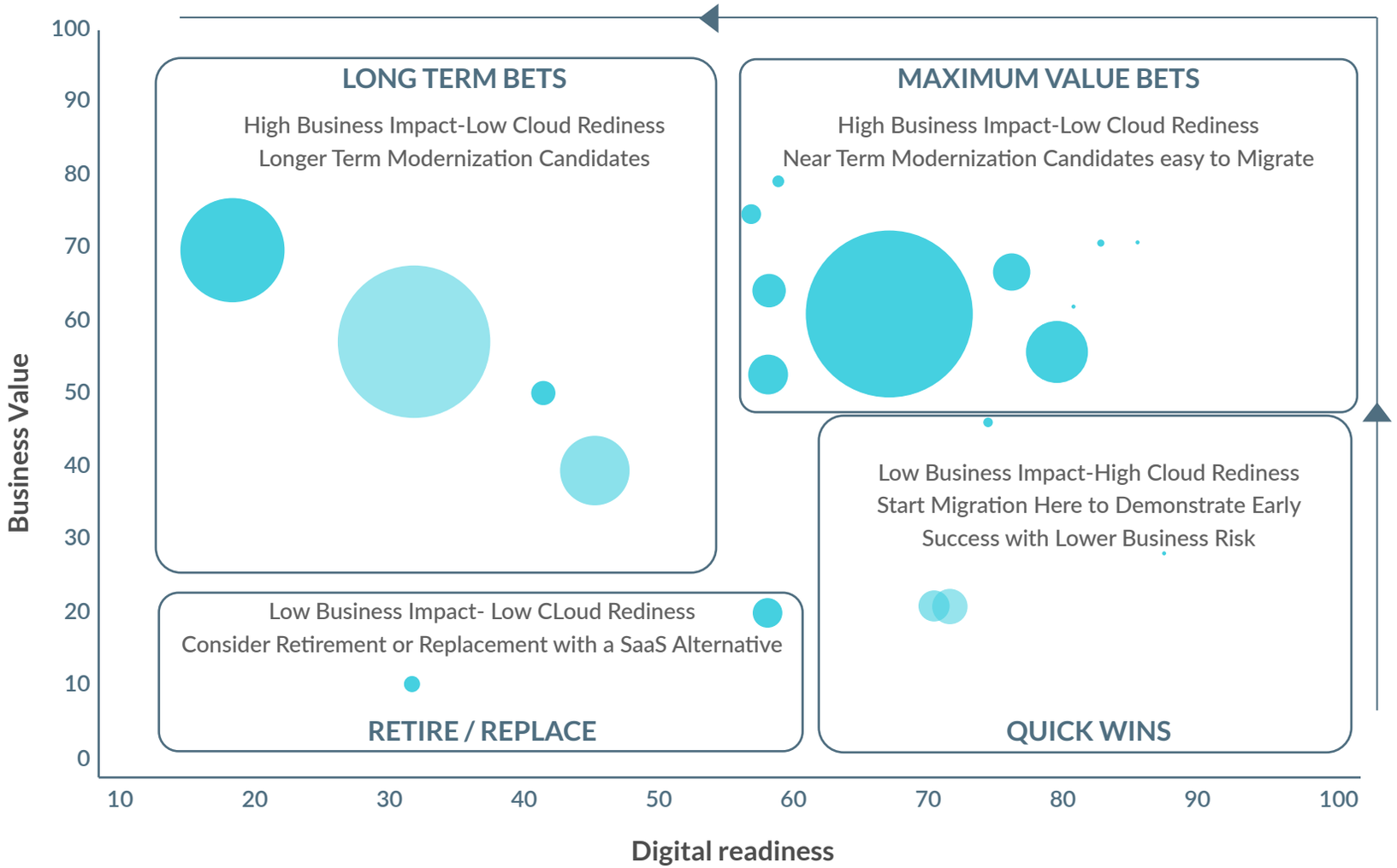
How to prioritize the applications to be modernized?

Any transformation of the legacy systems, including “refactoring to MSA,” should be carried out by identifying the right target applications for this journey.



Determining the right approach to application modernization can be a complex project in its own right — and it’s essential to do it correctly. If managed poorly, the application will become a legacy software again, inhibiting the business from working competitively.

Enterprises should take a holistic approach in terms of modernizing the application portfolio. Microservices might not make sense for all applications, after all. A data-driven and objective-based approach will help identify the right applications that generate significant business value from the transformation. Automated framework helps identify suitable refactor candidates and build a roadmap for the modernization journey.



Note: The bubble size indicates the size of the application.

Quick Wins

These are high digital-ready applications that have low business impact. These applications are ideal candidates to start the modernization journey to demonstrate early success, establish patterns, and iron out any issues upfront.

Maximum Value Bets

These are high digital-ready and high business impact applications. These provide maximum value and are ideally the focus of the modernization using MSA.

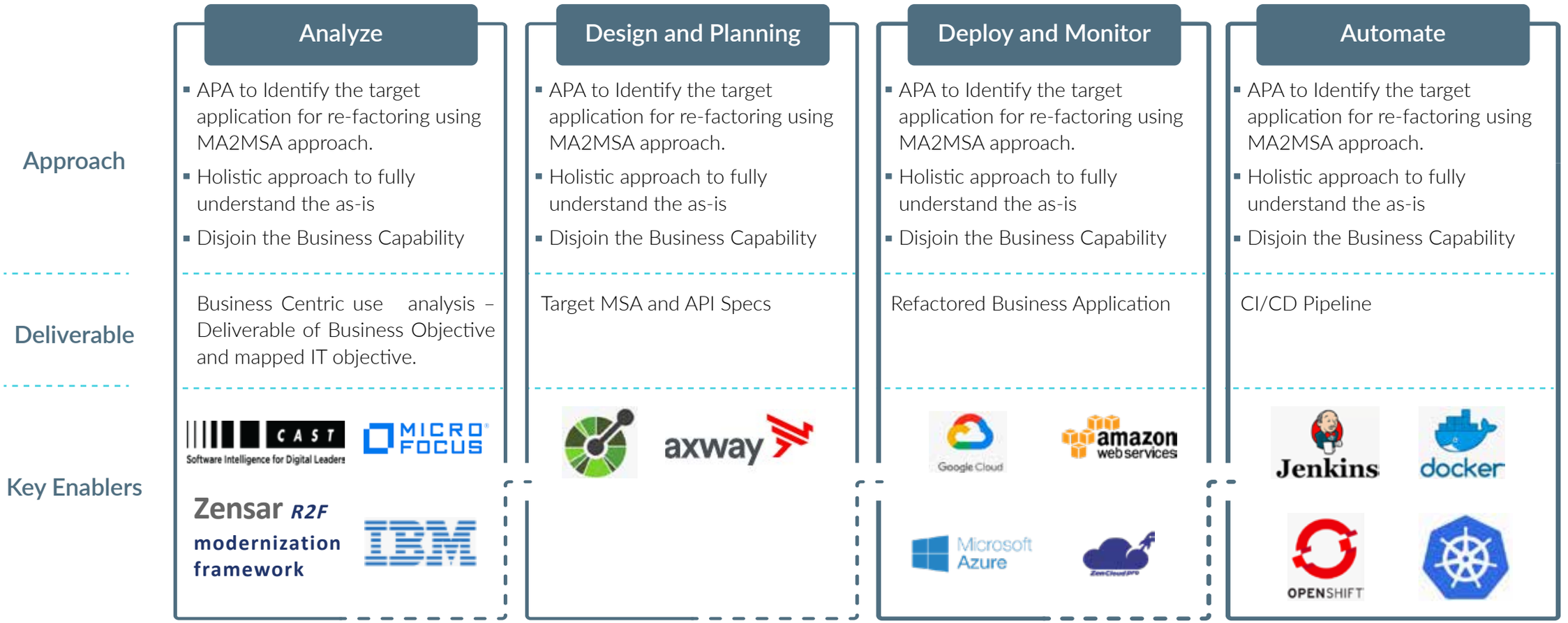
Long term Bets

These applications have a high business impact but a low digital readiness score and may present significant risk to modernize upfront. These applications should be considered long-term modernization candidates.

Retire/Replace

These applications have a low digital readiness score and offer low business impact. These applications are ideal candidates for decommissioning.

How can you start your Monolith to Microservices journey?



Monolith to Microservices Migration Framework – Enabling strategic Application Modernization

Best practices for enterprises while migrating to Microservices

To reap the full benefits of Microservices, businesses should modernize with clear goals — and measure progress, outcomes and benefits.

Design MSA with API-first approach in mind

With Open Banking, Open Insurance, collaboration with partner ecosystem being the future, think API-first approach and more specifically, Open APIs as the goal of the modernization to achieve maximum benefits. Any microservice must be consumer agnostic and an API-first design approach ensures just the same.

Transaction Monitoring

Being able to uniquely track the transactions, for instance in banking processes, must be given the focus as that of functional requirements. Otherwise the operations team may be swamped with such service requests taking away most of their bandwidth.

Test-driven approach

When you are trying to break the monolith without losing any functionality, it is necessary to first develop all possible test cases before embarking on re-architecture. This is particularly true for financial services industry where the combination of code and business rules produce different results for same types of transactions. If there is no such test pack available, then extracting modules out of monolith is not likely to be a success.

CI/CD/DevOps and Cultural changes are the keys to move from monolith to MSA

To truly experience the benefits of MSA, changes are needed in how companies build, test, deploy, and release the application or parts of the application. A fundamental shift in thinking is required and should take into consideration small, well-defined business functions, distributed teams leveraging multiple technology suites, and continuous delivery.

Modernization means much more than a "cloud-washed" application

Not every issue gets resolved by hosting applications in the cloud. In fact, this may prove costlier in some scenarios. The most straightforward use case is to expose the services in a monolith as APIs to the outside world. It is ineffective to move to the cloud without taking an API-first design approach.

Microservices can be appropriately managed on-premise

Modernization resulting from MSA is about splitting the services in a monolith and has little to do with the underlying infrastructure. If companies are ready to host containers on-prem, the same can be done for microservices.

Design cloud platform-agnostic

By going serverless, companies risk being tied to a particular cloud provider, losing the freedom of remaining cloud platform-agnostic. A better strategy is to deploy on-premise, in the cloud, or hybrid, leaving opportunity to change as the need arises.

Packaged software doesn’t require Microservices

Businesses selling packaged software may not be a good candidate for MSA, as their customers may not be able to handle the complexities that are inherent to MSA.

Zensar’s experience

Re-imagined enrollment for high performance and on-demand scalability

A fortune 500 US-based specialty insurance company needed to optimize its time-consuming, complex policy enrollment process for existing partner carriers as well as onboarding new partner carriers. The policy enrollment process was cumbersome and unable to scale. Also, co-existing validation rules at the database layer led to instability, such as outages and timeouts.

Solution:

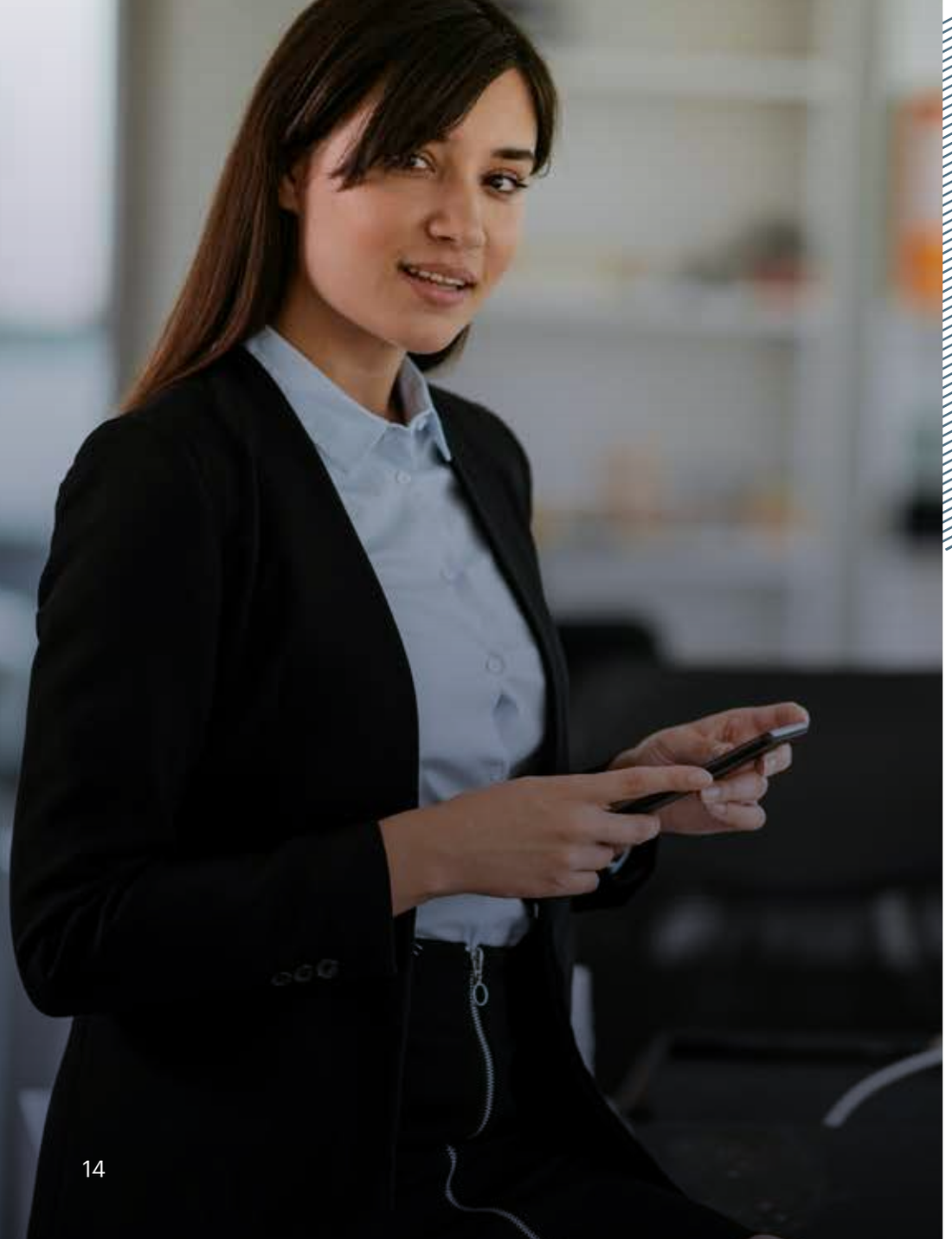
Our team worked with the insurance domain client to develop a **scalable and robust** application using Azure PaaS offerings. Zensar led by creating proof of the proposed architecture and ran multiple scenarios with vivid data sizes to develop an optimized architecture. This helped our client streamline its policy enrollment process for partner carriers. We provided:

- End-to-end solution design
- Performance improvement
- Agile delivery model

Impact:

With Zensar-proposed architecture, our client realized key business benefits, including:

- Up to 70% reduction in processing time for enrollment files from existing partner carriers
- Approximately 50% reduction in new client onboarding time with enhanced configuration options
- Availability and scalability using Azure platform capabilities
- API enablement for on-demand enrollments
- Dashboard for monitoring the state of enrollment files processing



Conclusion and next steps

Transformation of Monolith to Microservices Architecture can be complicated and time-consuming but can also be simplified if performed systematically. Enterprises should not implement a big-bang approach for redesigning applications to MSA or moving the monolith as-is to the cloud. Instead, consider incrementally refactoring applications into a set of microservices that would independently deliver business value. There are three strategies for this: implementing new functionality as microservices, splitting the presentation components from the business and data access components, and converting existing modules in the monolith into services. Business-centric and incremental modernization is critical to ensure that the journey from monolith to MSA derives the expected benefits.

For more information, please contact:

Vipul Jain
Principal Solution Architect,
API Economy and Microservices

Durga Prasad Pulipati
Global Head,
Modernization Practice

zensar
An RPG Company

We conceptualize, build, and manage digital products through experience design, data engineering, and advanced analytics for over 200 leading companies. Our solutions leverage industry-leading platforms, and help clients be competitive, agile, and disruptive as they navigate transformational changes with velocity.

With headquarters in Pune, India, our 10,000+ associates work across 33 locations, including San Jose, Seattle, Princeton, Cape Town, London, Singapore, and Mexico City.

For more information please contact: **marketing@zensar.com | www.zensar.com**