MICRO FOCUS

Zensar

Monolith to Microservices

A practical guide

Playbook

Table of Contents

Putting modernization into perspective

Modernization has come to mean a wide range of things over the years, from rewriting applications in newer programming languages, to moving mainframe applications unchanged to the cloud. All these definitions fall short of the real purpose of modernization, which is to ensure that an organization's IT systems can support the modern demands of its business.

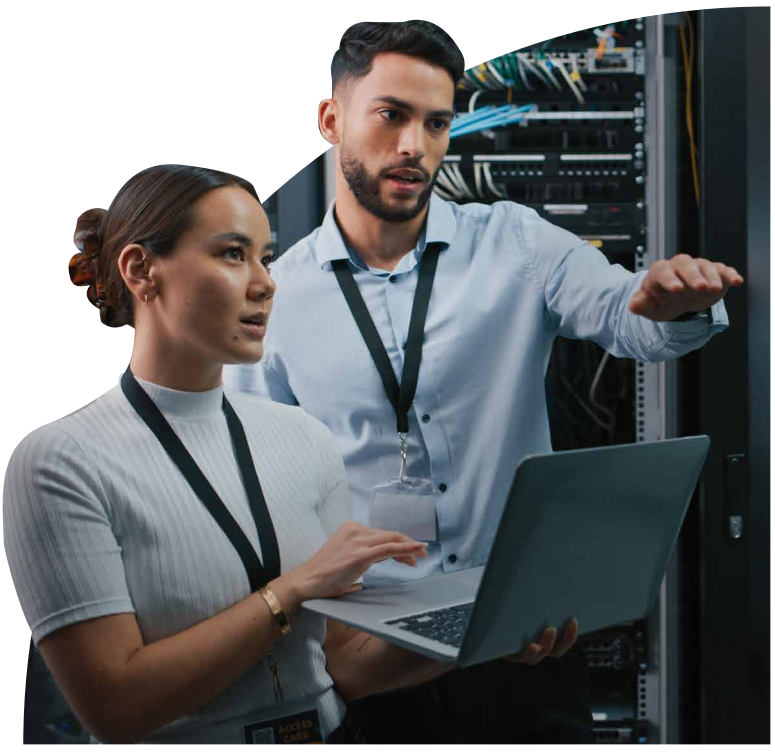
There are three key aspects that must be considered in any effort to ensure that IT systems are able to support and adapt to the needs of a business:

Application modernization for innovation— Executing important application changes and supporting Application Programming Interface (API) initiatives and cloud-native technologies to expose and integrate trusted functionality as new services

Process modernization for velocity— Establishing at scale the steps needed to understand, identify, implement, and test crucial business system changes and new functions

Infrastructure modernization for flexibility— Maximizing the opportunities offered by cloud computing and containerization to deploy new business services with speed, flexibility, and in a cost-effective manner

Quickly achieving flexibility promised by modernization is risky and challenging. It doesn't necessarily realize the 'quick wins' that the business intends to achieve. Velocity is a key aspect of modernization, and plays a vital role in ensuring that the direction that a business takes towards its ultimate vision is in tandem with speed. Multiple studies underline the alarmingly high failure rates of 'rip and replace' IT projects. In that context, it is safer and more economical to reuse tried and tested, unique, business-critical core applications and repeatedly enhanced data.



It is critical to determine the best modernization technique for specific situations based on the existing environment, business needs, and individual applications. An individual organization is more likely to need various modernization techniques to use throughout its enterprise. The Micro Focus Modernization Maturity Model is a framework designed to help with planning, and Zensar's proprietary application transformation framework helps with the execution of modernization programs. By reviewing modernization plans through the lenses of **infrastructure, application, IT process, management, and culture,** this model enables organizations to build a holistic view of their modernization plans. Zensar's end-to-end transformation strategy which includes **Discover, Assess, Plan, Pilot, Migrate, and Realize** helps organizations execute their modernization plans.

This playbook focuses on the most advanced phase for applications (rearchitecting), and microservices architecture (MSA), and assessing when this level of application modernization is necessary, and how to go about it.

Before we dive into microservices, let's discuss monolith applications, and the key drivers that help to integrate it with microservices.

Is monolith bad?

Virtually every company, whether industry giants or small start-ups, are talking about their resilient and scalable MSAs. As the hype of microservices gets stronger, monolith becomes synonymous with archaic, and even faulty architecture that is unable to scale up to current business needs.

The fast-paced nature of the industry causes people to forget that monolith application design was once the preferred architecture, and at some point, perhaps the only one.

We also tend to forget that core monolith systems have been serving the world's largest corporations for decades.

Nostalgia aside, some of the advantages of monolith-built applications are as follows :

Sharing and reusing code is very simple as all the codes are available and can be made accessible with different patterns

Deployment of monolith is straightforward and can be actioned immediately

End-to-end testing is simple due to the single deployment factor

High rate of efficiency as in-process and in-memory communication are the quickest and safest methods of communication

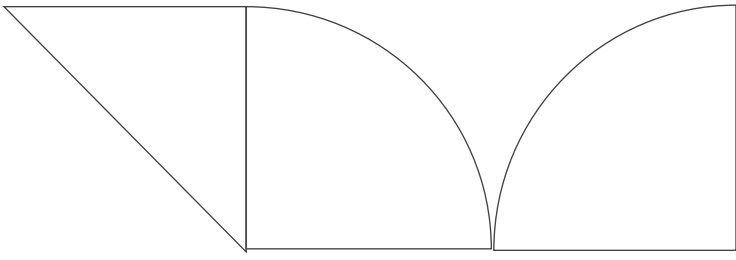
It is worth mentioning here, that even with unchanged monolith applications, businesses can still derive enormous value and save costs by merely modernizing the infrastructure. Cloud-based deployments are available using the Micro Focus Enterprise Server, which allow companies to leverage on elastic computing and agile cost models.

That said, businesses need to change to keep pace with innovations in software and infrastructure. Monolith application owners struggle to keep up due to badly designed application and deteriorating quality of service.

The skills required to maintain these complex systems are scarce due to the generational shift in development teams. Ownership of the different parts of the application also prove challenging.

Perhaps the most significant challenge is the failure to react faster to business demands. Monolith applications by nature are deployed in one shot or in large chunks. The ability to plug in a new feature or a quick fix is paramount nowadays. This proves difficult for monolith applications, and it has a major impact on business.

These shortcomings in monolith led to the emergence of MSA.



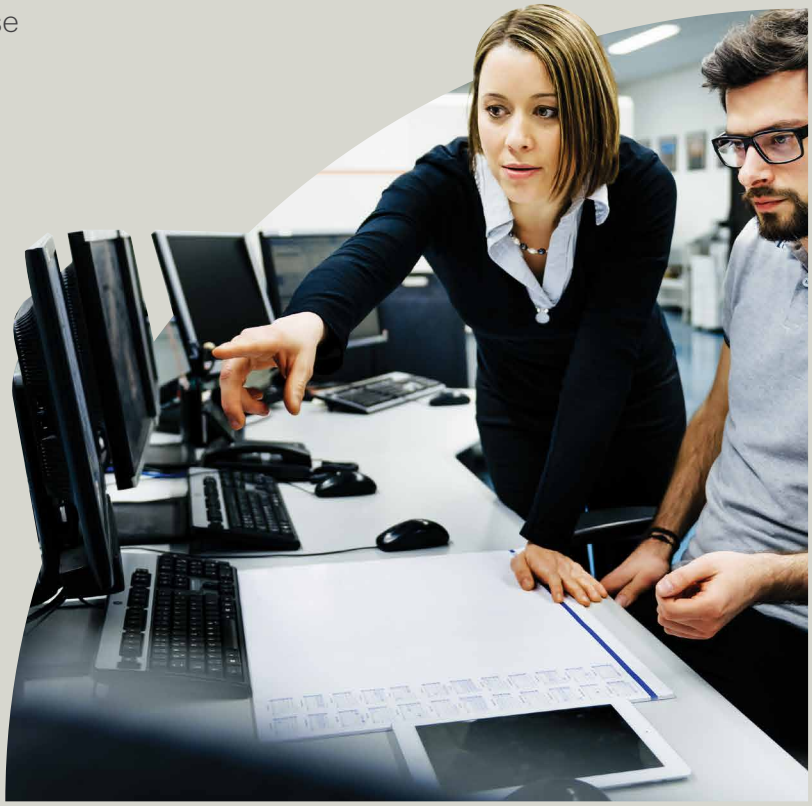
Zensar's POV on microservices architecture

Microservices are an architectural style that structures an application as a collection of loosely coupled services to implement business capabilities.

The microservice architectural style is the approach of developing an application as a suite of small services, each running independently, and communicating with lightweight mechanisms. These services are built around business capabilities and are independently deployable by fully automated deployment solutions.

A microservice can be deployed and scaled independently, enabling greater agility and operational efficiency for the enterprise. The MSA pattern has significant benefits, especially when it comes to enabling the agile development and delivery of complex enterprise applications. The service unit is much smaller and more contained in microservice architecture. It concerns itself with minimal responsibilities, thereby outlining its overall complexity to a single business function that it implements. If a single service becomes overly complex, rewriting that service is easier than having to rewrite the code threaded throughout a monolith.

Microservices architecture is flexible. Individual services can be built using a variety of platforms, languages, and tools, as these choices affect only a small team at a time. Rapid changes can be made, and quick updates can be released by developers without breaking business continuity. These core benefits enable enterprises to build composable applications quickly, allowing them to scale their businesses with increased productivity and fault tolerance.



The strategy of moving to MSA

Modernization from monolith to microservices is often a radical shift for organizations, and requires the right strategy and execution from a business and technology perspective. Organizations need to acquire new tools, technology, and resources to enable the transformation, and usually partner with technology and service integrators to successfully transform their businesses.

Zensar and Micro Focus bring in the right capabilities with proven experience of implementing 'Transform to Microservices Architecture' at large enterprises. Zensar leverages its proprietary IP - Legacy Analyzer Suite, iBREM, and Transform2Micro combined with Micro Focus Enterprise Solution suite, to implement the transformation from monolith to MSA.

As moving to MSA is a high impact project, several crucial steps must be considered for the shift:

Analysis and planning

Careful analysis of the application is needed in order to avoid creating a distributed monolith. Mapping all relationships, and documenting the business logic, data flow and usage is key. Using commercial application intelligence tools such as Micro Focus Enterprise Analyzer will help shorten the analysis process and ensure that all information is collected and shared between team members. The purpose of the analysis is to scope the work, and provide the necessary information to plan the target microservices.

The collated information is used for high level definition of the microservices creation strategy and the target architecture during the planning stage. The entire project will pivot on the mapping between the existing application and the target architecture with the understanding that it will be adjusted over time.

It is important to maintain a balance between not having enough knowledge to proceed, and spending too much time on analysis at this stage in order to avoid analysis paralysis.

Self-funding the project by quick re-platforming

Breaking a monolith into microservices is a long term and expensive project. Companies that maintain their mainframe workloads while migrating microservices to the new platform will have to secure a huge budget, with Return on Investment (ROI) taking many years.
This strategy also means that the learning curve for creating the microservices remains the same as learning the new infrastructure, whether on cloud or on premises.

Two key benefits of re-platforming the application as-is to the target platform instead, are:

a. The costs saved on mainframes goes towards funding the modernization project

b. The new infrastructure learning curve is completed before the MSA project begins

As-is, re-platforming can be achieved in a few months, thereby quickly showing the value proposition and saving business costs.
The mainframe budget can then be allocated to the modernization project.

Gradual, flow-like modernization

Recent research by the Standish Group shows that continuous modernization has the highest success rate. After analyzing thousands of projects using a variety of modernization methodologies, the likelihood of success was determined as below:

https://www.microfocus.com/en-us/assets/application-modernization-and-connectivity/continuous-modernization-boundless-business-value

PROJECT TYPE	SUCCESSFUL	CHALLENGED	FAILED
Developed from scratch	26%	54%	20%
Developed using components	37%	46%	17%
Purchased application (COTS)	44%	36%	20%
Flow Like Modernization	71%	28%	1%

Figure 1: Success rate for modernization methodologies based on The Standish Group's CHAOS 2020 Database of 50,000 project profiles.

It was concluded that an iterative process of identifying candidates for modernization works much better than a sudden system overhaul.

Macroservices vs microservices

When companies create brand new systems using MSA, they try to adhere to the core concept of logic separation, data separation, and other best practices. When re-factoring monoliths into MSA, it is reasonable to balance between best practices and reality using a phased approach. Macroservices are bigger parts of the application that are exposed using standard APIs. It seems transparent to the consumers of these macroservices, but they lack the resilience and scalability that microservices have.

After the initial re-platforming, the application is broken down to macroservices that expose its functionality using API enablement techniques. The next stage is to extract or re-write the microservices if the situation demands.

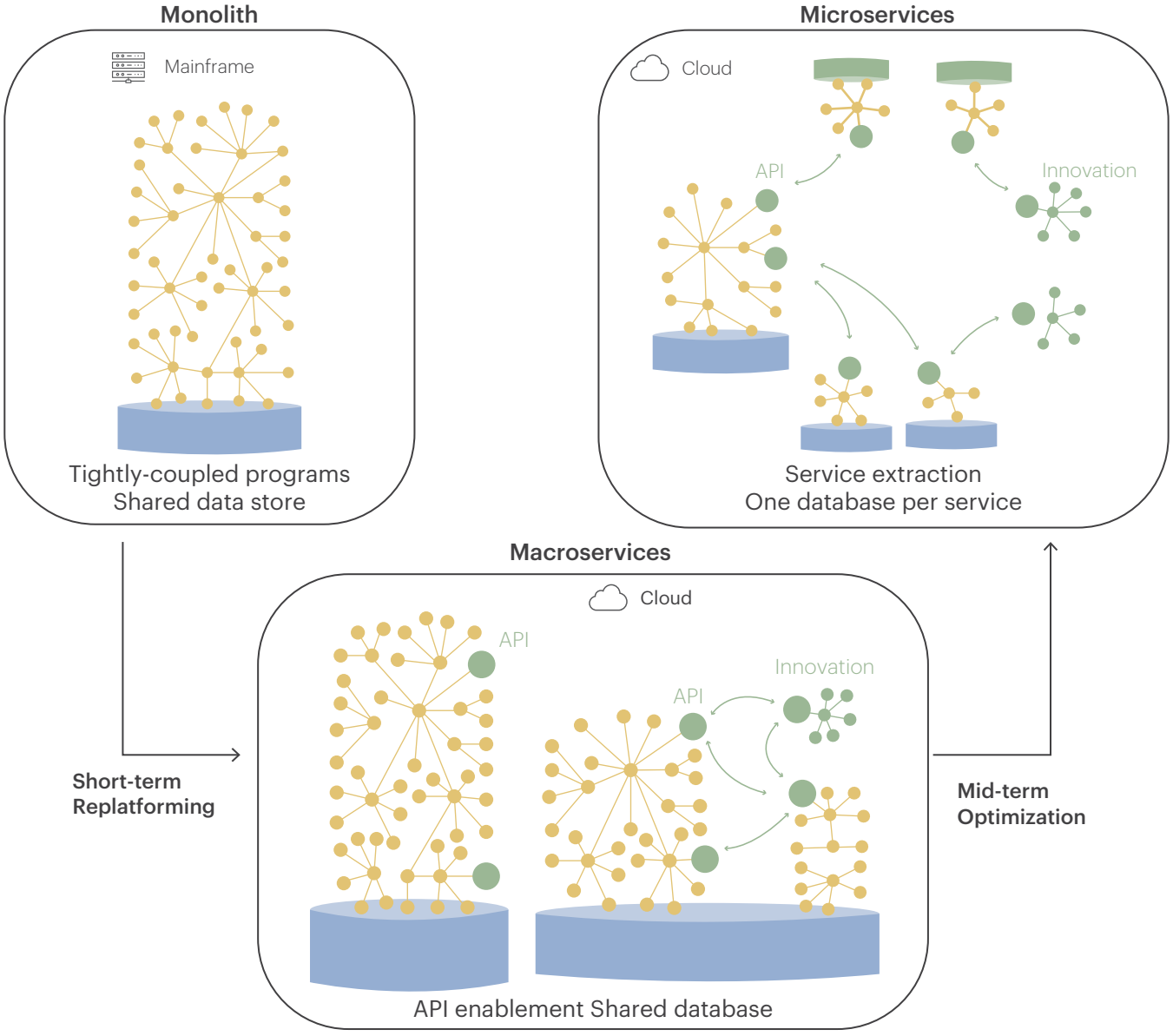


Figure 2: Understanding the monolith – macroservices – microservices differentiation

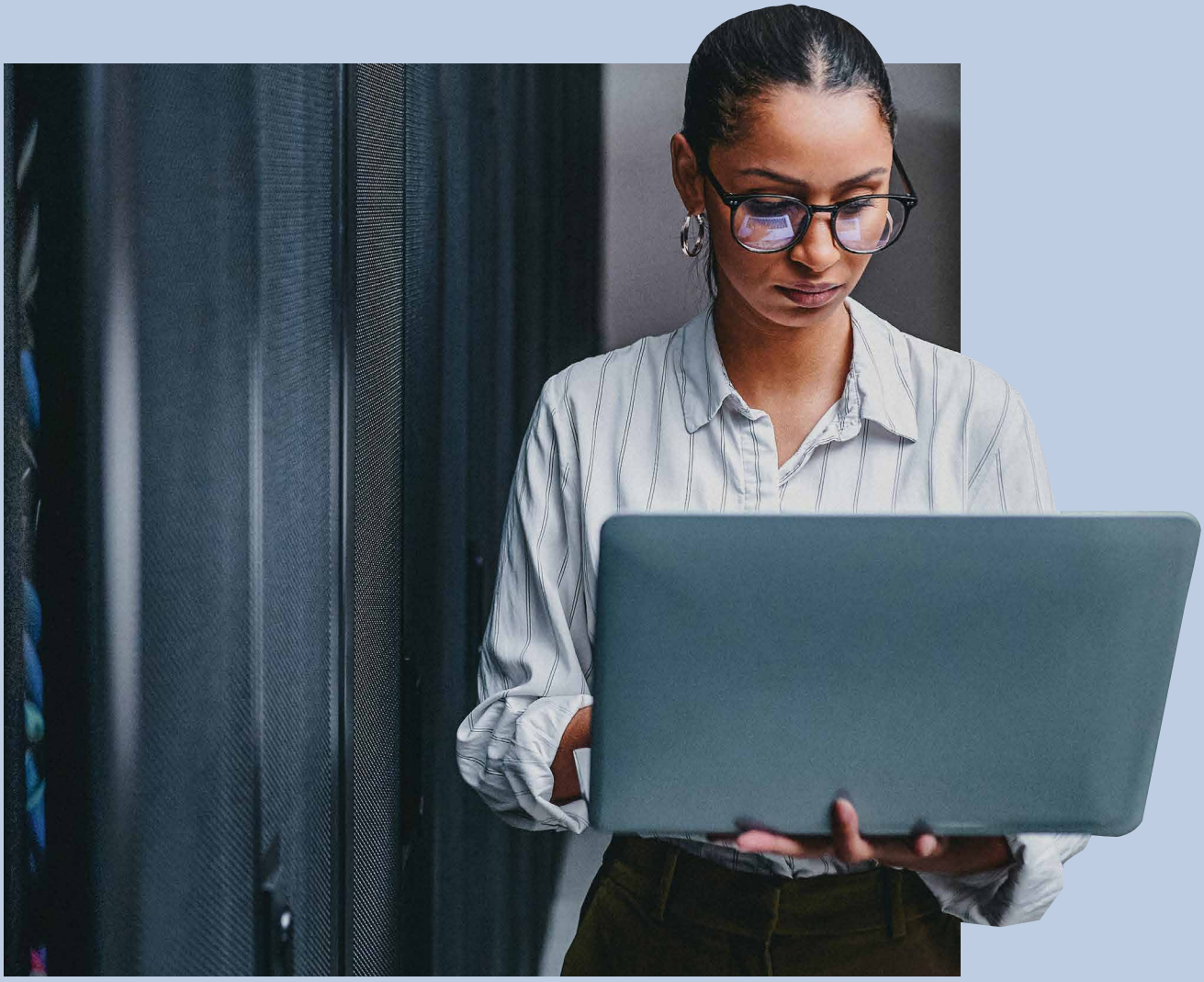
How to start the journey

Businesses have been moving their critical and non-critical applications to the cloud for decades. But the advancement in technologies and increased security requirements for cloud infrastructure has quickened the pace of transformation. It is becoming increasingly urgent in order for modernizing applications to meet the business needs more efficiently.

Cloud strategy and assessment play a vital role to embark on the application modernization journey.

Zensar's end-to-end transformation strategy completes the assessment by performing six different phases i.e. **Discover, Assess, Plan, Pilot, Migrate, and Realize.**

These phases cover additional assessment with respect to application disposition, data governance, and security strategy.



Phase	Activities
Discover Capture the existing in scope landscape	• Understand strategic business • Capture current state, tech stack • Conduct stakeholder interviews
Assess Identify the cloud provider and application migration roadmap	• Assess application maturity • Review application disposition • Identify application redundancies
Plan Create plan for moving the workloads and applications	• Define the overall roadmap • Classify components as 6R • Setup governance model
Pilot Identify the pilot application and there by workloads	• Define the architecture, landing zone • Automate using the DevSecOps platform • Setup security policies
Migrate Perform the cutover and move to cloud	• Perform cutover for pilot • Establish operating procedures • Evaluate performance metrics
Realize Optimize the processes and control the current cloud spend	• Enforce policy automation, monitoring • Optimize operation process • Re-asses the cloud spend

Figure 3: The six phases of Zensar's end-to-end transformation strategy

Various modernization swim-lanes (Paths – DevOps, Cloud, Migration to Cloud)

Modernization can be achieved by breaking down the monolith and architecting the future state solution as microservice-based architecture, especially in the case of the legacy applications built into COBOL and mainframes.

Figure 4 depicts the key application modernization steps that should be followed to transform the legacy application into microservice-based architecture.

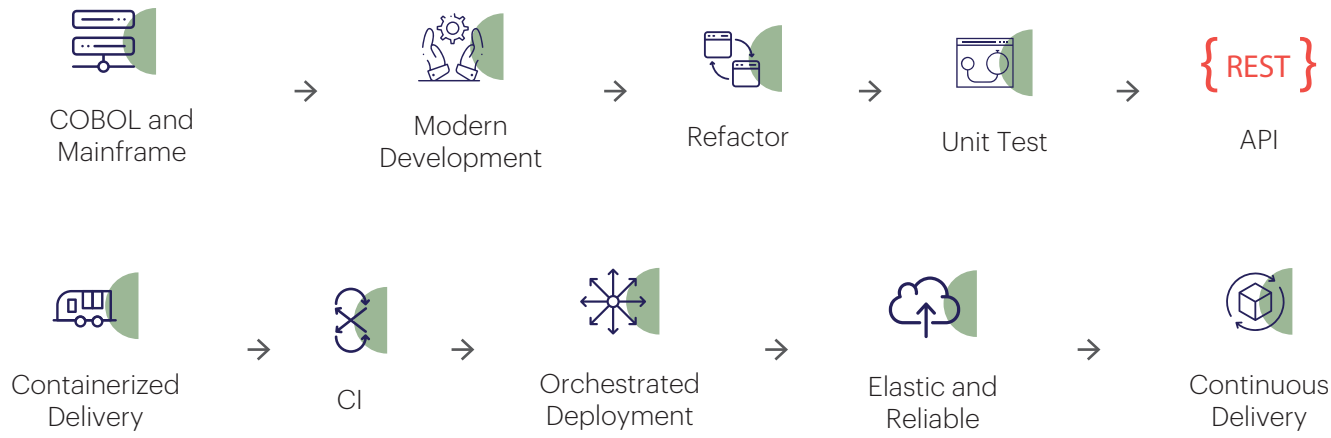


Figure 4: Key steps to transform the legacy application into microservice based architecture

Micro Focus Enterprise Suite allows the above activities to be performed using various Suite products such as -

- Application Analysis
- Application Testing
- Application Development
- Application Deployment

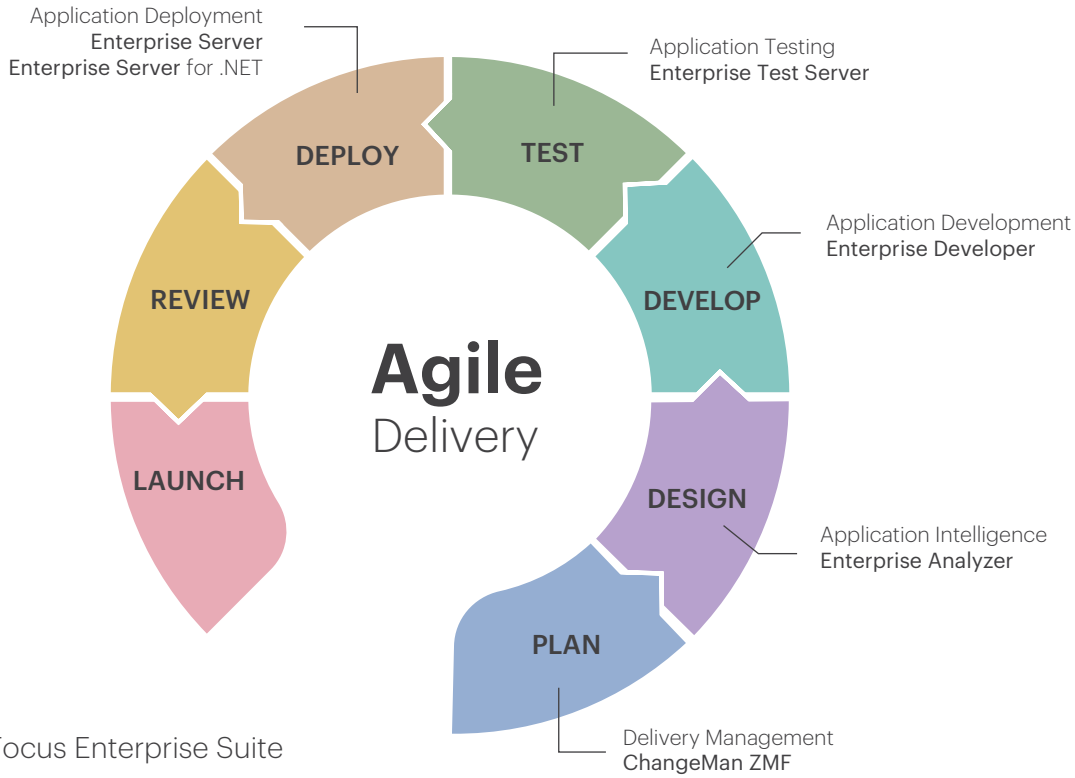


Figure 5 : Micro Focus Enterprise Suite

Micro Focus track record and credentials in enterprise modernization creates a unique and comprehensive capability

- Streamline development and delivery activities by 40%, using contemporary technology, DevOps agility, and unrivaled flexibility
- Customers have achieved a 50%–90% reduction in IT operations costs and a performance improvement of up to 50% for batch and online transactions
- Micro Focus have delivered 1,000+ modernization projects, supported by a major global partner network and our Modernization Maturity Model— a framework for the planning and implementing a modernization journey

Best practices – deploying COBOL microservices on hyperscale providers

Serverless computing enabled by Kubernetes is now helping the deployment of COBOL (and PL/I) application on modern platforms, from virtual Linux, Unix and Windows (LUW) machines, on premise and on cloud through docker containers.

We will cover some of the options in this section:

Basic – running mainframe code on LUW machines

As mentioned, running the mainframes application on Linux, Unix and Windows (LUW) is the core capability of the Micro Focus Enterprise Server product. It has a strong track record, with over a thousand production mainframe workloads running on it across the globe.
With this functionality, the customer can deploy the virtually unchanged code to alternative platforms, whether physical machines on premises, virtual ones, or cloud environments.
COBOL microservices are no different, and can be run on every platform that the enterprise server supports.

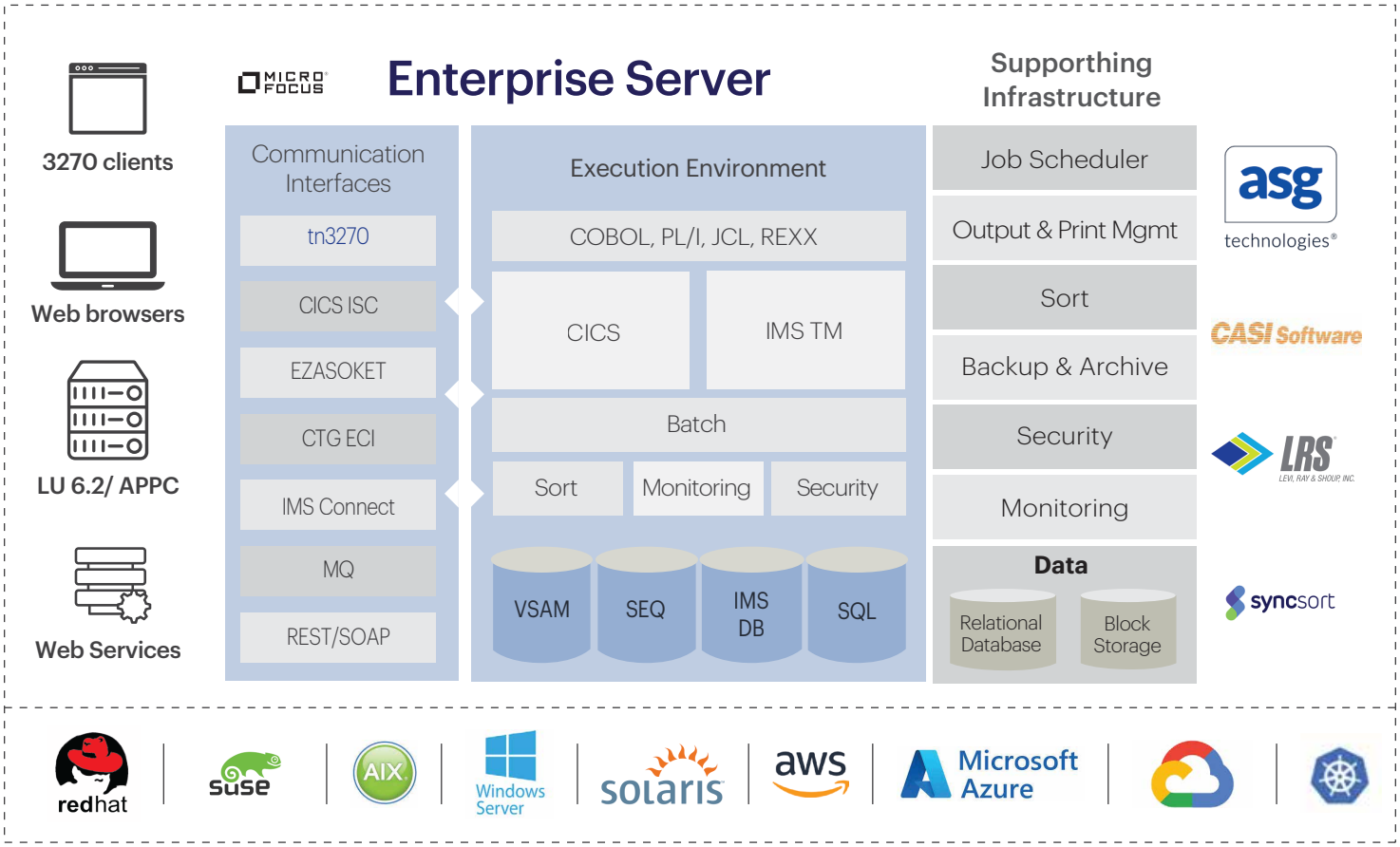


Figure 6: Enterprise server

Additional information on cloud deployment can be found here:

https://aws.amazon.com/blogs/apn/empowering-enterprise-grade-mainframe-workloads-on-aws-with-micro-focus/

High availability and resilience is achieved using the inbuilt scale-out repositories and relational database hosted Virtual Storage Access Method (VSAM).

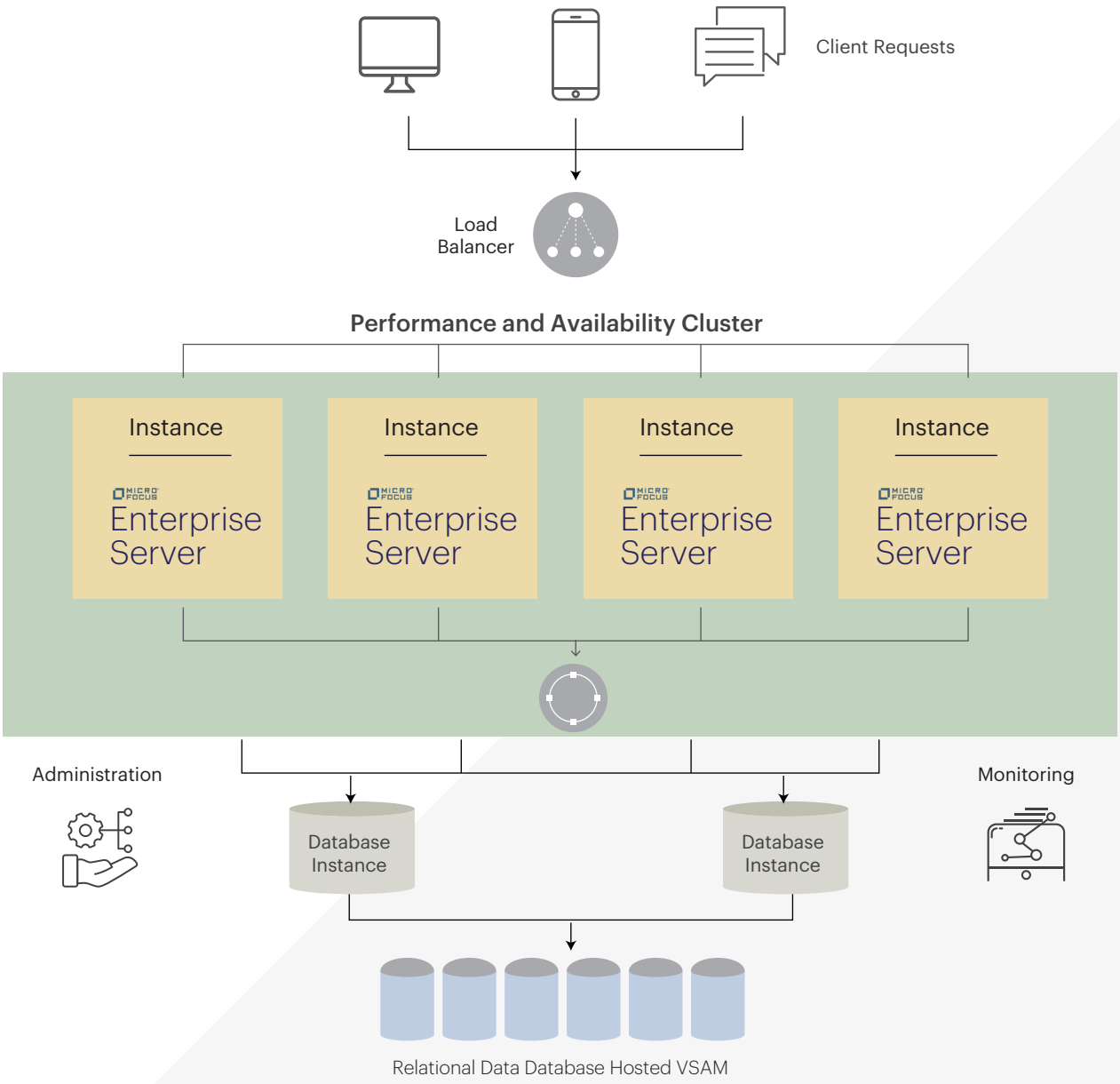


Figure 7: Use of the inbuilt scale-out repositories and relational databases hosted on VSAM

Containerized deployment

As containers are the de-facto modern deployment technology, COBOL microservices can also run inside containers orchestrated by Kubernetes or by the cloud provider technologies. Each Kubernetes pod can host one or more microservice, and the clusters may contain COBOL services alongside the services that are implemented in different technologies. This also achieves language independency which is an important aspect of MSA:

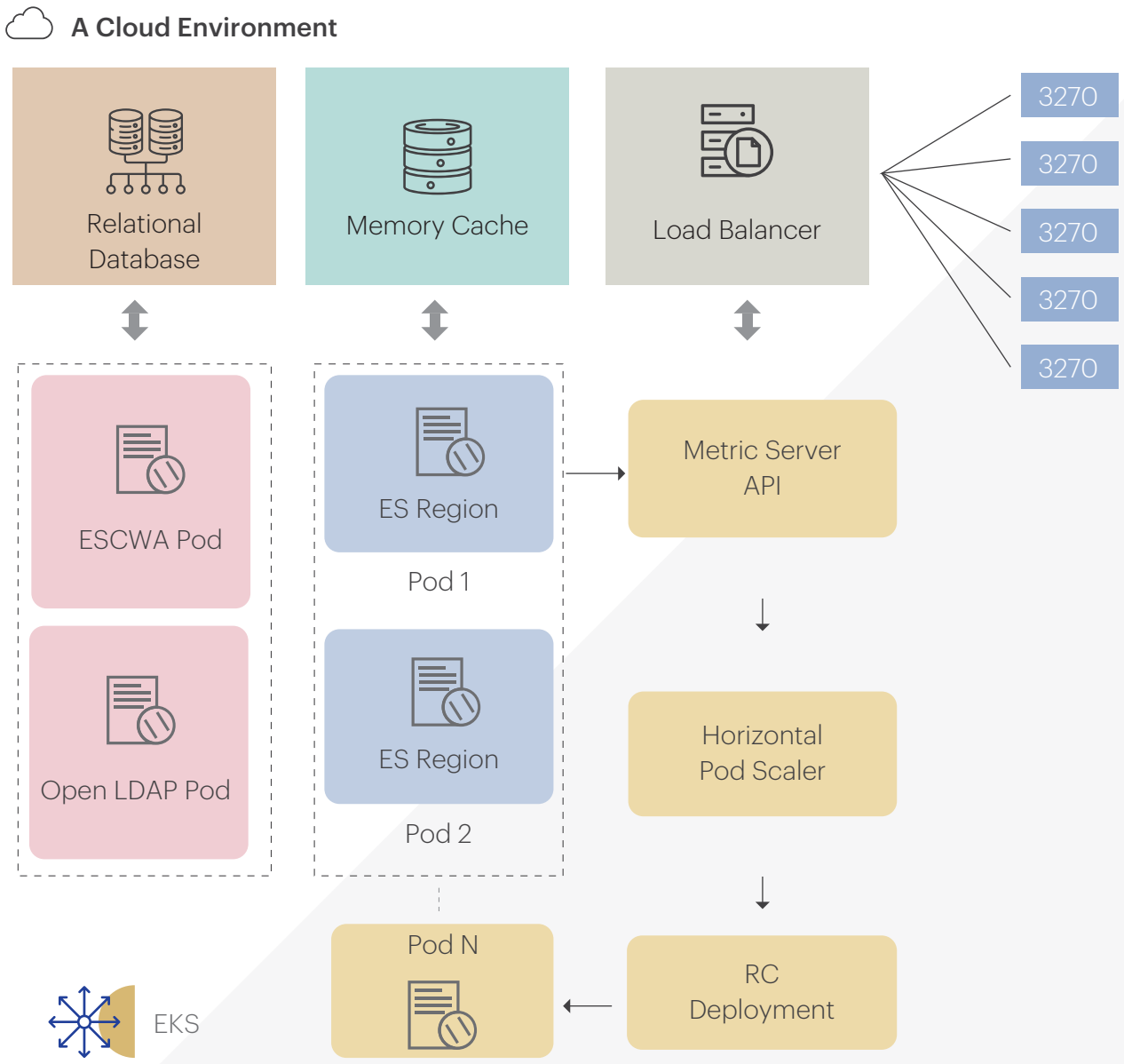


Figure 8: A cloud environment

Integration with .NET, JVM and serverless computing

COBOL logic that does not need mainframe resources and mainframe system emulation is easier to modernize. Microservices that are stripped of mainframe constructs and are made purely of COBOL can leverage a Micro Focus compiler option that generates .NET or JVM binaries. These binaries can be deployed inside .NET or JVM applications, and can easily invoke or be invoked by other parts of the application written in C# or Java, for example. Once the modules are compiled to those modern frameworks, they can also leverage the rich libraries capabilities seamlessly inside the COBOL code.

Being .NET or JVM binaries, they can run inside modern application servers and seamlessly leverage the modern serverless computing options such as AWS Lambda and Azure functions. This is an additional benefit.

```
16      procedure division using ws-Store-Information
17                               , ls-localtime
18                               .
19          display "MFOCALDIR" upon environment-name
20          display "../gb" upon environment-value
21          move ws-name-of-store to cmd-line
22          call "getonestore" using by reference cmd-line
23                                   by reference ws-Store-Information
24          end-call
25          set ls-localtime to type LocalTimeHelper::GetLocalTime(ws-latitude, ws-longitude)
```

LocalTimeHelper.cbl

```
8       *>> <returns>The local time in a given latitude/longitude according to Google</returns>
9       method-id GetLocalTime static.
10      
11      procedure division using by value latitude as float-long, by value longitude as float-long,
12                               returning res as string.
13      
14          declare client = new type RestSharp.RestClient("http://api.geonames.org")
15          declare request = new RestSharp.RestRequest("timezoneJSON", type RestSharp.Method::GET)
```

Ln: 9 Ch: 18 SPC CRLF

```
26
```

Figure 9: Standard procedural COBOL using .NET REST API invocation library

Services creation and API enablement

One of the core principals of MSA is that the different services communicate between each other in a standard manner, mostly REST using JSON.

The Micro Focus Enterprise Developer provides multiple ways to expose programs as interfaces. They are:

Enterprise Developer Interface Mapping Toolkit (IMTK) – This toolkit detects the linkage section content and maps it into the newly created service interface. HTTP methods and paths can be created and customized to create an easy-to-use RESTful service. Once defined, it can be deployed, tested, and debugged inside Visual Studio or Eclipse.

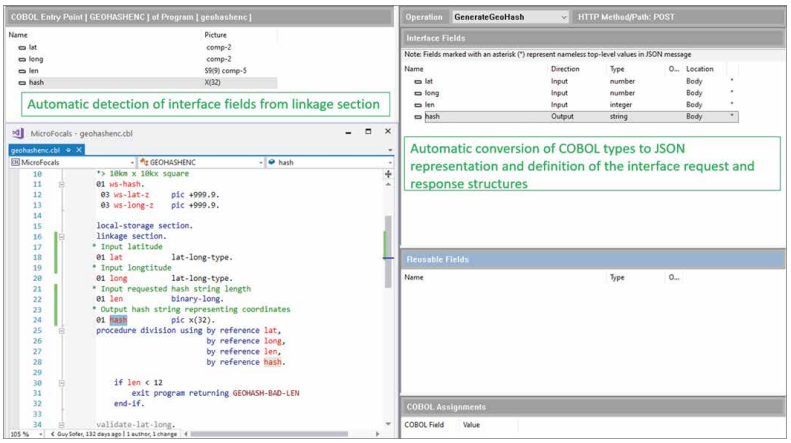


Figure 10: Programs as interfaces

Enterprise Developer CICS Web Service Wizard – If the code is using CICS resources, the CICS Web Service wizard can be used. This will allow creating SOAP or REST services and preserve the CICS environment as is, so that all the data access and permissions continue to work just like a standard CICS transaction.

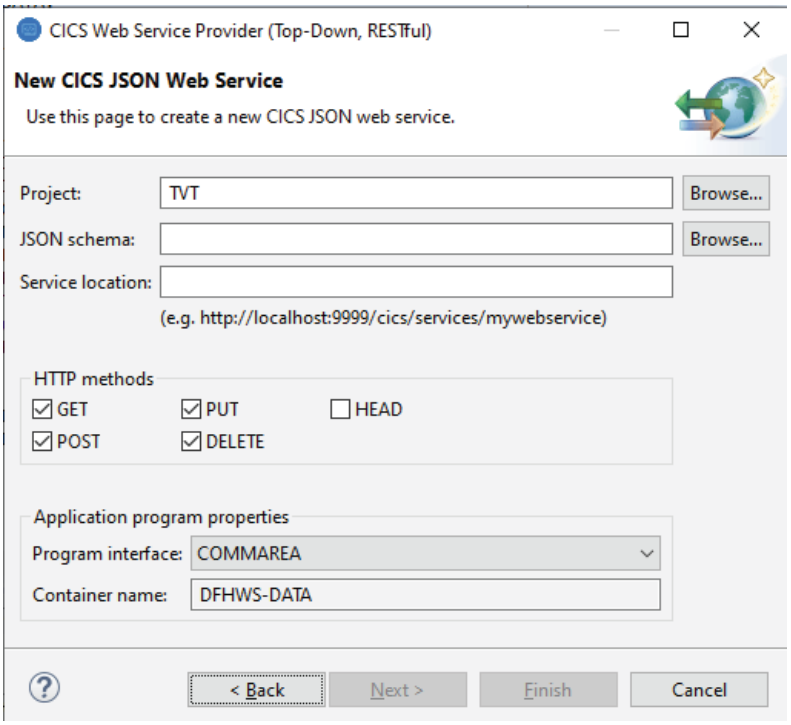


Figure 11: Enterprise Developer CICS Web Service Wizard when using CICS resources

Needless to say, once it runs on modern platforms using modern development tools, other methods of exposing code-as-a-service applies to COBOL and PL/I code like any other language.

Data modernization

One of the most critical parts of the application is its data. This could be VSAM or QSAM files, DB2, Oracle, IMS, or any other types and technologies.

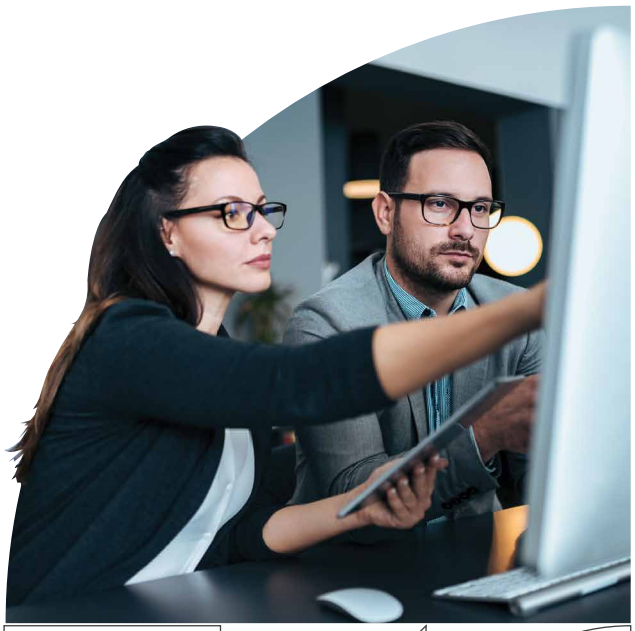
Data modernization is a vast topic and has many different strategies. Some of them are covered in the highly recommended book, Monolith to Microservices by Sam Newman.

Some of the capabilities that our customers are using to enable data modernization are listed below:

Continuous use of data files on the new platform - The Enterprise Server and its File Handler allows customers to repeatedly use the same files from the mainframe on the new target platform. While this is not a modernization activity, it does support the gradual modernization strategy discussed above. Another benefit is that re-use is the quickest and safest way to move the application data.

LUW versions of the database - in some cases, data resides in databases that have distributed versions, such as DB2 LUW. In this case, migrating the data is usually simpler, and it only requires minor changes in the application to access it.

Using Micro Focus Database File Handler (MFDBFH) to host the files data in a relational database - this approach allows hosting the data that resides in data files on a relational database without making any changes to the application code. The application code continues to access the data as if it were reading and writing files, while the actual data is updated on the database. In this case, resilience and redundancy are achieved without the need for a complete data re-architecture and migration project.



Micro Focus customers leverage savings and performance improvements to invest in onward modernization

Over one thousand customers have modernized their mainframe workloads using Micro Focus technology. Listed below are some examples of how these successful projects led to onward modernization.

1. **German Pension Fund (Deutsche Rentenversicherung)** needed to leverage proven COBOL business logic across traditional platforms and within new cutting-edge architectures to improve the performance of critical business functions.

They chose to create a microservices architecture, leveraging the ability to compile COBOL to Java Byte Code and run it inside their Java Virtual Machine. In addition to proving the logic of their COBOL code, the new solution has achieved a 70% reduction in batch job run time.

"With the Micro Focus solution, we gained a robust web service that we can operate with standard tools. We now use COBOL programs flexibly as microservices integrated in a service-oriented architecture to accelerate business processes even further, helping the organization deliver excellent client services."

- PETER PALMREUTHER, Senior Java Developer, Java Competence Center, German Pension Fund

More info here:
https://www.microfocus.com/es-es/case-study/german-pension-fund

2. **FIS** is a leading global provider of technology solutions for merchants, banks, and capital market firms. They wanted to improve the speed, efficiency, and quality of software development, support, and delivery processes, while simplifying operations and improving resilience and recovery.

FIS has leveraged the support for COBOL containerization and compilation to JVM COBOL, and now have a cloud deployed application that performs 20% faster. It is horizontally scalable, which is connected to a continuous deployment pipeline.

"When we rolled out the replatformed application with Visual COBOL we actually saw an overall performance boost of more than 20 percent. Not only are our processes running faster, but we can scale out and run more concurrent processes. In line with our streamlined and agile development this is a key differentiator for our business."

- CHUCK WAINSCOTT, Director of Architecture, FIS asset management group

More info here:
https://www.microfocus.com/en-us/case-study/fis

3. Spanish bank **LABORAL Kutxa's** implementation of new technologies, languages, and databases, together with modern development methodologies enabled them to look at further modernizing their applications.

Taking advantage of these newer technologies has improved the bank's application performance by 80% and cut operation costs by an additional 80%.

"In the Micro Focus-powered environment, one batch process was reduced from 2 hours to just 12 minutes. Others that used to take over 100 minutes are now completed in only a minute. The efficiency of our systems improved by over 80 percent overall, while select processes improved by more than 90 percent."

- JOSEBA MARURI, Director of Technology and Systems, LABORAL Kutxa

More info: https://www.microfocus.com/media/case-study/laboral-kutxa-cs.pdf

4. A **global insurance applications** leader leveraged Kubernetes and Micro Focus for their digital transformation journey to a high-availability cloud solution. By modernizing and scaling up to a full cloud-agnostic insurance application running Micro Focus COBOL powered by Kubernetes, the company's formerly mainframe-based solution now delivers a <40% reduction of operating costs along with reduced RPO/RTO, rapid scalability, and new digital capabilities through open APIs – all successfully deployed across their customers' sites.

Conclusion

Leveraging existing investments in COBOL application and taking advantage of new technology and paradigms by migrating from monoliths to microservices is possible. The partnership of Zensar and Micro Focus can help you achieve this as we have helped businesses plan and execute modernization in keeping with their individual readiness and business requirements to deliver the right value.

Please reach out to us if you would like to achieve application modernization and transformation of your legacy mainframe application.

Contact Us

Ross Botha

Director of Alliances
South Africa – Zensar

ross.botha@zensar.com

Allyson Towle

Country Marketing Manager,
South Africa – Micro Focus

allyson.towle@microfocus.com

zensar
An RPG Company

We conceptualize, build, and manage digital products through experience design, data engineering, and advanced analytics for over 130 leading companies. Our solutions leverage industry-leading platforms to help our clients be competitive, agile, and disruptive while moving with velocity through change and opportunity.

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: velocity@zensar.com | www.zensar.com