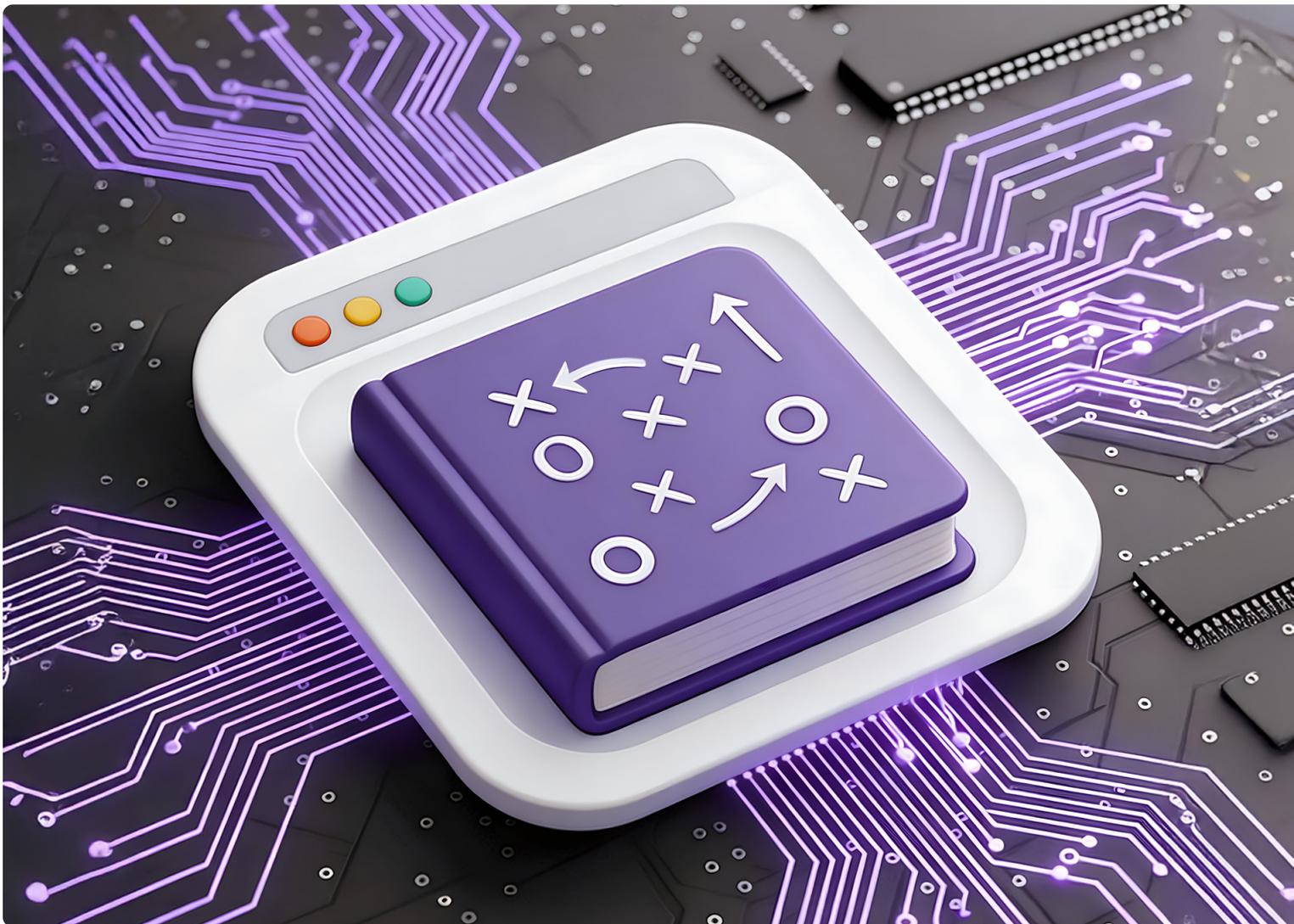


The Application Modernization Playbook

Navigate the 5 Rs with Ensono's proven framework to
unlock agility, scalability, and innovation



Map your modernization path with a tailored strategy

For seasoned IT leaders, the 5 Rs are well understood. The real challenge isn't knowing the options, it's choosing the right combination, in the right order, without introducing unnecessary risk or stalling delivery.

Rehost, Replatform, Rearchitect, Rebuild, and Replace outline standard strategies that can help organizations make smart, cost-effective decisions about how to [modernize their application portfolios](#).

Each "R" represents a different level of effort, risk, and return. You can apply a mix of these strategies across your portfolio, depending on the scope and impact of change required, to drive [measurable impact](#).



Name	Alternative names	When to use	Decision signal
Rehost	Lift and shift, redeploy	When speed and low risk are priorities; ideal for moving workloads quickly without changing the application.	Infrastructure cost or data center exit is the primary driver, and the application is stable but not strategic.
Replatform	Lift and reshape	When you need cloud benefits without major code changes; good for improving scalability and security and reducing operational overhead.	The application is still valuable, but operational overhead and scalability are becoming limiting.
Rearchitect	Reengineer, refactor	When technical debt or architecture limits agility; best for improving performance and enabling future modernization and adoption of modern technologies or architectures (e.g., microservices, cloud-native).	Release cycles are slow, changes are risky, or the architecture prevents integration with modern platforms.
Rebuild	Rewrite, redesign	When legacy systems can't meet evolving business needs; opportunity to deliver new functionality and experiences.	The business model has changed, and the current system can no longer support it without significant compromise.
Replace	Repurchasing, drop and shop	When maintaining custom apps is cost-prohibitive; replace with SaaS or COTS for faster ROI.	The application provides commodity capability that no longer differentiates the business.

While the 5 Rs describe what can be done to an application, Ensono’s **Evolve**, **Integrate**, and **Build** offerings reflect how modernization actually happens in practice—incrementally, asymmetrically, and driven by business value rather than technology purity.



[Fig. 1] Ensono’s application modernization framework.

The adoption of microservices architecture enables modular modernization of large, monolithic applications—a common target state for organizations seeking agility and scalability. Cloud computing accelerates this shift, allowing applications to be modernized incrementally rather than through a disruptive rebuild.

Ensono’s three offerings—**Evolve**, **Integrate**, and **Build**—help clients move toward this modern state. Each offering addresses different challenges, giving organizations the flexibility to prioritize the most valuable transformations first while Ensono supports them throughout the journey, including managing legacy systems.

Evolve: Stabilize, Secure, and Enable Change

Objective:

Modernize existing applications to reduce risk, improve resilience, and increase the speed and safety of change.

Key results:

- Reduce critical security vulnerabilities and audit findings across the application estate.
- Increase application availability and elasticity to support business demand.
- Improve release frequency and reduce change failure rates through automation.
- Lower total cost of ownership by reducing legacy platform and operational overhead.

Evolve—which encompasses both Replatform and Rearchitect—is often the first step in [modernizing](#) existing applications and in some cases, the only step required. It involves migrating to cloud-native services, upgrading platform runtimes, language transformation (such as converting COBOL to .NET or Java), containerization, migrating to managed databases, enhancing CI/CD, implementing automated testing, improving observability, using Infrastructure as Code, and enabling DevSecOps.

Evolve is focused on laying the foundation to increase the speed of change by mitigating risk, enhancing availability and scale, improving business agility, and delivering ongoing operational cost savings.



Evolve focuses on three outcomes

- 1 Reducing operational risk:** Resolving security vulnerabilities, runtime upgrades, observability, DevSecOps, and Infrastructure as Code.
- 2 Increasing speed and reliability of change:** Automating build, test, and deployment processes with CI/CD, implementing automated testing, and using containerization to enable faster, more predictable releases.
- 3 Lowering cost and complexity:** Reducing licensing, infrastructure, and management overhead by shifting to managed databases and cloud-native services.

Evolve key elements

Modernization step	Description
 Enhances security and compliance	Evaluate the codebase using AI-driven analysis and tooling to identify and remediate security vulnerabilities, applying cloud-native security controls and removing issues such as hardcoded secrets or exposed sensitive data.
 Migrate to cloud-native services	Replatform existing services to cloud-native equivalents to improve security, resiliency, and scalability.
 Platform runtime upgrades	Upgrade frameworks and technologies (e.g., .NET Framework 3.5 to .NET 8, SQL Server 2008 to newer versions).
 Language transformation	Convert legacy languages like COBOL to modern ones such as .NET or Java.
 Containerization	Replatform workloads to run in containers (e.g., Docker, Kubernetes) to improve portability, scalability, and deployment speed.
 Migrate to managed databases	Shift to managed database offerings to remove costly licenses, enable horizontal scaling, and reduce management overhead.
 CI/CD	Automate validation and deployment to improve code quality and reduce time to market.
 Automated testing	Use test automation frameworks across layers to improve reliability, quality, and speed.
 Observability	Enhance monitoring, logging, and alerting to better understand and diagnose system behavior.
 Infrastructure as Code	Manage infrastructure with tools like Terraform or Ansible for consistency, repeatability, and version control to support active-active and resilient infrastructures.
 DevSecOps	Embed automated security testing into build pipelines using cloud-native security features to support vulnerability management.

Evolve benefits



Enhance security and compliance

Reduce enterprise security and regulatory risk by embedding automated controls and auditability across the application estate.



Increase availability and enable rapid scaling

Protect revenue and customer experience by assuring critical applications are resilient, highly available, and able to scale on demand.



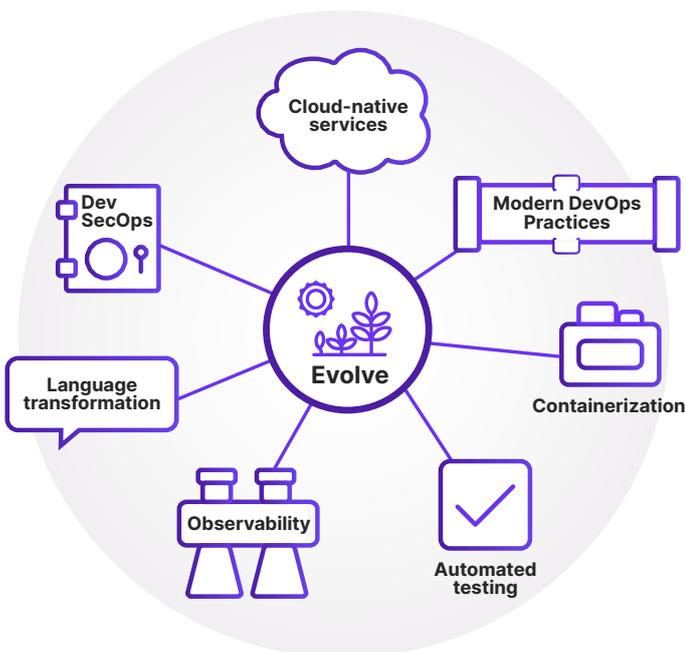
Improve business agility and innovation

Accelerate time-to-market by enabling faster, safer delivery of new products, features, and regulatory changes.



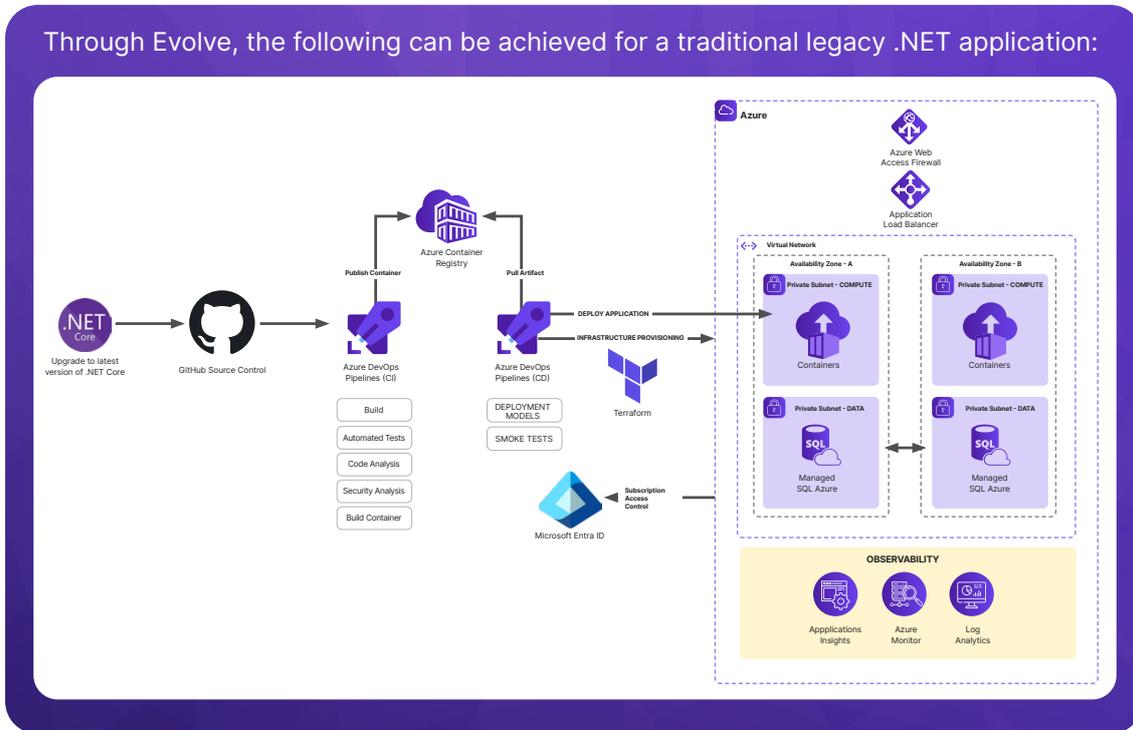
Deliver cost savings and efficiencies

Lower total cost of ownership by reducing legacy platform dependency, automating operations, and shifting spend toward growth initiatives.



[Fig. 2] Key steps in Evolve.

Through Evolve, the following can be achieved for a traditional legacy .NET application:



[Fig. 3] A legacy .NET application to modern cloud-native architecture.

Integrate: Decouple, Derisk, and Create Optionality

Objective:

Reduce release and modernization risk while increasing flexibility by decoupling systems and enabling incremental change.

Key results:

- Reduce cross-system dependencies and coordinated releases.
- Increase the speed of change by enabling independent delivery of integrated capabilities.
- Introduce SaaS and third-party platforms without disrupting core operations.
- Improve system resilience by isolating failures and reducing blast radius.

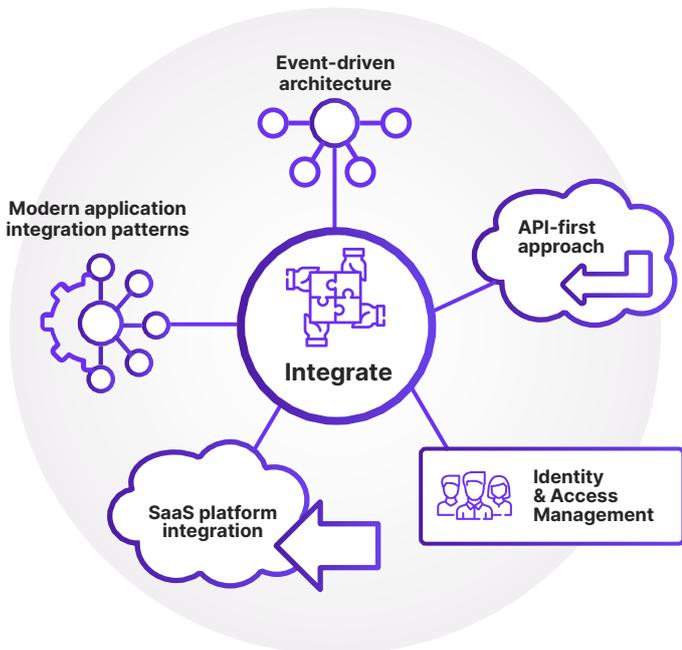
Integration-led modernization allows organizations to change where it's safest first—decoupling customer-facing and high-change areas while stabilizing core systems that still perform reliably. For many organizations, introducing APIs and event streams around legacy systems enables SaaS adoption and new digital experiences without touching core transaction engines—reducing risk while increasing speed.

The goal is to reduce the risk of releases, increase flexibility, and enable the ability to replace certain capabilities with SaaS platforms. **Integrate** delivers safer change, faster response, and lower modernization risk by decoupling systems and enabling incremental evolution.

Ensono believes that businesses should maintain focus on their core mission by building systems high up in the value stream and not worry about building low-level components that the business doesn't specialize in. These areas make good candidates to integrate with third-party platforms, such as Identity and Access Management.

Just because an application is classed as legacy doesn't mean it needs to be rebuilt from the ground up, which can be costly, time-consuming, and risky. Techniques such as the strangler pattern and anti-corruption layers allow gradual replacement of legacy functionality, simplify integration with modern platforms, and reduce dependency on outdated systems—making modernization more achievable.

Integrate key elements	
Modernization step	Description
 Modern application integration patterns	Replace tightly coupled systems with loosely coupled event-driven architectures to improve flexibility, resilience, and speed of change.
 API-first approach	Prevent direct database access by enforcing versioned APIs, reducing risk and enabling modernization.
 API management	Expose and control APIs with features like throttling and transformation (e.g., SOAP to REST), supporting flexibility and abstraction from legacy systems.
 Event-driven architecture	Use pub/sub, CQRS, and event sourcing to enable scalable, decoupled, real-time communication between services.
 Event streaming	Process data in real time to support responsive, scalable applications.
 Integration with SaaS platforms	Reduce complexity by leveraging SaaS for non-core capabilities, avoiding unnecessary custom development.
 Identity and Access Management	Securely manage user identities and access across integrated systems to support compliance and scalability.



[Fig. 4] Integration patterns in action.

Integrate benefits

- 
Reduce release risk
 Minimize production incidents and business disruption by decoupling releases and automating validation and rollback.
- 
Increase flexibility and speed of change
 Enable faster, independent delivery of changes by reducing system coupling and dependency bottlenecks.
- 
Enable an easier transition from legacy systems
 Lower modernization risk by progressively isolating and replacing legacy capabilities without disrupting core operations.
- 
Improve system resilience and fault tolerance
 Strengthen operational resilience by assuring systems continue to operate during failures, spikes in demand, or partial outages.

Build: Differentiate, Scale, and Enable Growth

Objective:

Deliver new digital capabilities that support strategic growth, differentiation, and long-term scalability.

Key results:

- Accelerate time-to-market for new customer experiences and digital products.
- Align technology delivery with evolving business and regulatory priorities.
- Deliver scalable, resilient platforms capable of supporting growth and peak demand.
- Eliminate legacy constraints and technical debt that limit innovation.

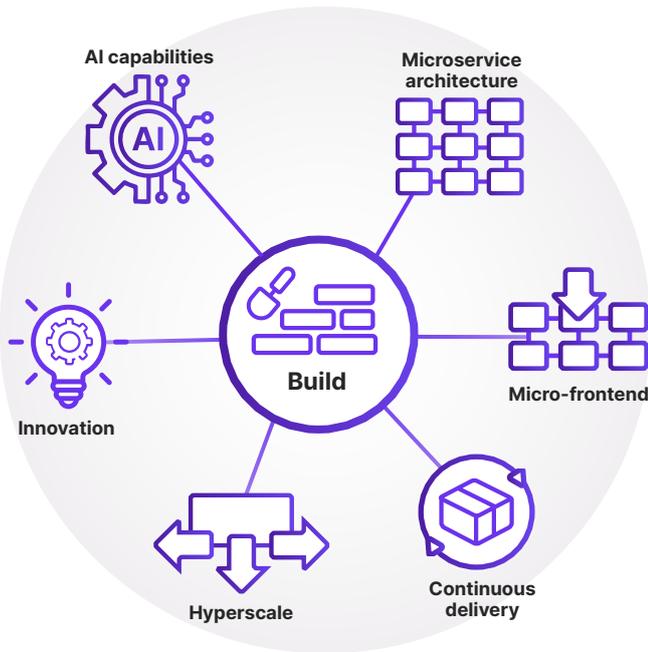
Build involves building new, modern systems. Although it's often seen as the most daunting in terms of time for value, cost to deliver, and risk to transition, a rebuild shouldn't be a mere technology upgrade of the current platform. Successful rebuilds start with business capabilities and data, not technology. Without this focus, organizations risk creating distributed monoliths that are harder—not easier—to operate. It should be an opportunity to deliver value through new features and customer experiences.

A rebuild is appropriate when the current platform can't be invested in or when business goals have changed substantially.

Modern new applications often attempt to adopt a distributed architecture, commonly microservices. It's one thing to read about building microservice architectures, but another to deliver them successfully. Without the right skills and expertise, you can end up with a distributed monolith that's just as hard to build, release, and maintain. Ensono has experience in delivering hyperscale microservices and micro-frontend applications, with a strong focus on data.

Build key elements

Modernization step	Description
 Microservice architecture	Adopt distributed architectures to improve scalability and agility while avoiding pitfalls like distributed monoliths through expert implementation.
 Modern web applications	Build responsive, user-centric applications that deliver new customer experiences and align with evolving business needs.
 Micro-frontends	Break down frontend monoliths into modular components to enable independent deployment, faster iteration, and improved maintainability.
 Decompose data to new data stores	Restructure legacy data architectures to support microservices and modern workloads, enabling better scalability and performance.
 Continuous delivery	Automate deployment pipelines to accelerate release cycles, reduce risk, and improve software quality through rapid iteration.



[Fig. 5] A modernization journey map.

Build benefits



Accelerate the delivery of new customer experiences

Reduce time-to-market for new digital products and customer journeys by enabling independent, rapid delivery of new capabilities.



Ensure strategic alignment with evolving business goals

Align technology delivery with business strategy by building platforms that can adapt quickly to changes in market direction, regulation, and growth priorities.



Deliver scalable and resilient architecture

Support growth and reliability by adopting scalable, fault-tolerant architectures that can evolve without compromising performance or availability.



Eliminate legacy constraints and technical debt

Remove structural and technical limitations by replacing obsolete platforms and architectures that restrict innovation, scalability, and change.

Application modernization isn't one-size-fits-all. Ensono's [flexible approach](#) helps organizations focus on the most valuable transformations first. Whether modernizing legacy systems, decoupling integrations, or building new platforms, Ensono supports clients through every stage of their journey.

Ready to evaluate your application modernization options?

A structured portfolio assessment can help identify which applications to evolve, integrate, build—and in what order—to maximise value while controlling risk.

Contact Ensono to start your modernization journey with clarity and confidence.

Let's Connect