Legacy Application Modernization

Bringing Business-Critical Applications to the Cloud

Legacy Application Modernization: Bringing Business-Critical Applications to the Cloud

Overview

Most enterprises today have built their business on legacy applications. In many cases these applications are still critical to the business, but are not nearly as performant or cost-effective as they once were. Legacy applications have long release cycles and are costly and resource-intensive to maintain. Many of these applications are dependent on specific infrastructure and services, essentially locking the enterprises in to more outdated and inefficient technology. This hurts time to market and competitiveness.

To become faster and more agile, enterprises want to modernize their legacy applications. Modernization enables innovation, controls costs, and provides new ways to consume information. There are several approaches to modernization, ranging from a simple "lift and shift"—where an existing application is moved to a more conducive environment—to a complete decomposition and re-architecture of a legacy application.

The Apcera Platform is designed to support both cloud-native and legacy applications. Because Apcera is secure by default, leverages true hybrid cloud infrastructure, and is a complete, enterprise-grade solution, the platform is ideal for enterprises that are in the midst of their modernization journey. In particular, Apcera is for enterprises that want to containerize and migrate legacy applications to the cloud.

Key Takeaways

Running and maintaining legacy applications involves considerable challenges.

People tend to think of legacy applications as applications that are old. But David Williams, Principal at Williams & Garcia (an independent technology solution provider), offered a slightly different definition. Williams sees "legacy" as applications and architectures that are not modern or current, regardless of their age. More precisely, a legacy application is anything that is not a 12-factor application.

WHAT IS A 12-FACTOR APPLICATION?		
I.	Codebase	One codebase tracking in revision control, many deploys
II.	Dependencies	Explicitly declare and isolate dependencies
III.	Config	Store config in the environment
IV.	Backing services	Backing services
V.	Build, release, run	Strictly separate build and run stages
VI.	Processes	Execute the app as one or more stateless processes
VII.	Port binding	Export services via port binding
VIII.	Concurrency	Scale out via the process model
IX.	Disposability	Maximize robustness with fast startup and graceful shutdown
X.	Dev/prod parity	Keep development, staging, and production as similar as possible
XI.	Logs	Treat logs as event streams
XII.	Admin processes	Run admin/management tasks as one-off processes

Legacy applications present enterprises with multiple challenges, which include:

- **Lock-in.** Enterprises can get locked into their technology stack as legacy applications are often dependent on infrastructure.
- **Resource intensive.** Legacy applications are resource intensive to update and maintain. It is also becoming increasingly difficult to find people proficient at maintaining older technology.
- **Long release cycles.** An enterprise might be stuck with an 18-month release cycle on a core application. In today's fast-paced, competitive landscape, this can hinder the pace of innovation and prevent companies from getting to market quickly with new products.
- **Difficult to thoroughly test.** Many legacy applications were built without adequate functional tests, and any testing with legacy apps is manual and ad hoc.

- **Deployment difficulties.** Deployment processes are manual and slow, and if mistakes are made in deployment they might not be discovered for weeks.

> "Legacy applications can be time consuming, costly, and risk prone. The net result is that your enterprise is stuck spending most of its time in an effort dealing with legacy applications, and is unable to move forward."
>
> – Henry Stapp, Director of Product Management, Apcera

Modernization of legacy applications has tremendous benefits, but comes with real challenges.

Modernization of an organization's technology is critical to provide the innovation, speed, and agility that companies need to compete in the marketplace. Modernization also enables enterprises to reduce costs around infrastructure and licensing. For example, when companies manage their own dedicated hosts to run an application, they may end up with VM or infrastructure sprawl, which increases costs. In contrast, when modernizing applications, enterprises that move to a modern container-based deployment model can dramatically reduce infrastructure costs and get a better handle on licensing costs, deployment, and provisioning.

Benefits of Modernization

- Enables fast-paced innovation
- Controls IT costs
- Makes IT more proactive
- Provides new ways to consume information

While legacy apps require IT organizations to be reactive in fighting fires and working just to keep the lights on, modernization enables IT to be proactive and strategic. Modernization also enables application information to be consumed in new ways. The traditional way of consuming an application is through a web browser, which is still the leading use case. But modernization allows for application information to be consumed through mobile, social, and even virtual reality. Modernization makes it possible to open up application architecture to provide access to internal data flows, such as data from the Internet of Things.

> "There are a lot of companies experimenting and innovating to try to discover new ways to engage their customers. . . . You can't really do that easily with traditional or legacy application architectures because they don't have the means of facilitating those kinds of consumption models."
>
> – David Williams, Principal, Williams & Garcia

Challenges arise because modernization affects more than just the application. It impacts infrastructure, the application architecture, processes, and even the culture within an organization.

Legacy applications are often unique and idiosyncratic in how they are built, configured, and deployed, which makes them difficult to standardize. Legacy applications often have specific infrastructure dependencies, where one app runs on one server, limiting infrastructure flexibility. Modernization of applications results in a more agnostic infrastructure, where assets are used more efficiently.

Also, legacy apps rarely use automation, while modernized apps are heavily based on automation, and legacy apps tend to have siloed lifecycle management compared to modernized apps, which have more fluid, agile lifecycle management. This allows enterprises to release features and functionalities for customers right away and iterate quickly.

The benefits of legacy application modernization are important to the modern enterprise, but it is equally important that the people and processes are amenable to change to support the modernized application.

FIGURE 1 | Challenges During Modernization Process

Legacy Application	Modernized Application
Unique build, deployment and configuration	Standardized build, deployment and configuration
Infrastructure dependencies	Infrastructure agnostic
Rarely uses automation Based	Based heavily on automation
Tend to have siloed, waterfall lifecycle management	More fluid, agile DevOps lifecycle management

Enterprises face critical decisions about what and how to modernize.

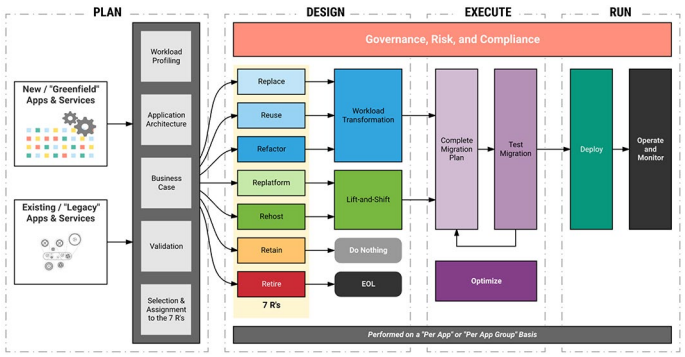
Most enterprises have dozens or even hundreds of legacy applications. In deciding to modernize, it is not possible to modernize everything at once, requiring prioritization. Deciding where to start requires:

- Treating every legacy application as unique and understanding each application's differences.
- Determining the targeted future state for each legacy application, which includes thinking about the desired consumption models for an application.
- Prioritizing applications with the greatest business impact.
- Prioritizing applications that are easier to modernize.

There are seven general approaches to modernizing.

These seven approaches, shown below, are "the 7 R's."

FIGURE 2 | Modernization Approaches



The 7 R's are:

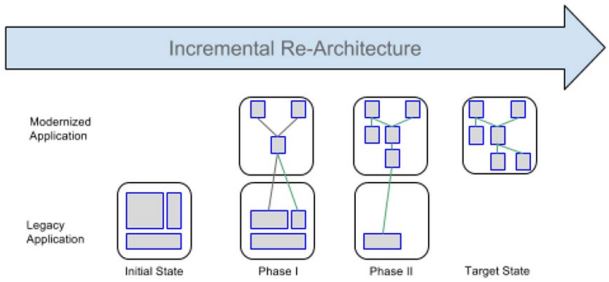
- **Retire.** An application is no longer needed and a plan is developed to sunset it. The application has reached the end of its useful life.
- **Retain.** An application is kept and left as it is. It isn't moved to a different platform or modernized.
- **Rehost.** With minimal changes, an application is moved ("lifted and shifted") into a different type of runtime environment. This is fast and affordable, and can be effective at reducing some data center costs,
- **Replatform.** The application is lifted and shifted with minimal changes, but more significant infrastructure changes are made to address environmental dependencies the application needs. For example, instead of reusing an Oracle or SQL Server database, an enterprise could take advantage of Amazon RDS to realize the operational advantages of a cloud provider. This is where real modernization comes in, which can transform an enterprise.
- **Refactor.** This involves making very specific changes and revisions to an application to leverage the cloud.
- **Reuse.** This is consolidating services and application components, and productizing services that an application (or multiple applications) provides. This is being seen to some degree with microservices.
- **Replace.** This is a rare and extreme move. This might be done by completely replacing an existing application with a SaaS service or building cloud-native applications. It is a complete rewrite from the ground up, rather than trying to refactor or reuse components from the original application.

Many enterprises are getting the greatest bang for their buck through lift and shift. Doing so is fast, reduces costs, and increases agility. An example is moving from physical to virtual or from virtual to containers. Lift and shift is ideal for some legacy applications, such as commercial off-the-shelf applications like an SAP application that can't be rearchitected, or applications that are less resource-intensive.

> "With lift and shift, the idea is you want to modernize the environment of the application without doing workload transformation on the application itself."
>
> – Henry Stapp, Director of Product Management, Apcera

With the reuse approach, an enterprise may conclude that some redesign is required, but doing so incrementally has advantages like faster time to value, risk mitigation, and a path to full modernization. This looks at modernization as a process with multiple steps.

FIGURE 3 | Reuse — Incremental Modernization Pattern



> "There are cases where you say, 'We want to start decomposing this application and moving it towards a more modern architecture,' but you don't want to do that all at once. . . . Over time you modernize your application into the desired target space."
>
> – Henry Stapp, Director of Product Management, Apcera

Apcera—designed for cloud-native and legacy applications.

The Apcera Platform was created to support cloud-native and legacy applications, while simultaneously supporting the modernization of legacy applications. Apcera is the only enterprise-grade platform that provides value out-of-the-box and can scale with the organization.

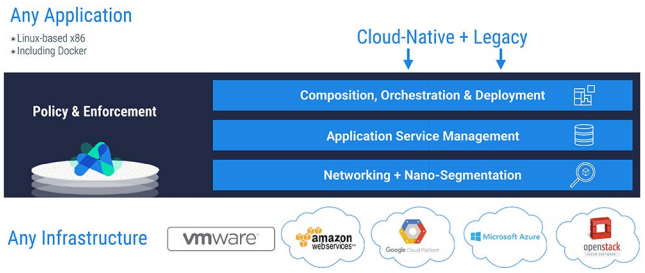
Key elements of this platform are:

- **Cloud-Native and Legacy** – One platform for all your applications.
- **Security by Default** – Ingress and egress are tightly controlled by policy and all doors are closed by default, providing enterprise-grade security.
- **True Hybrid Infrastructure** – Treat all your infrastructure as a single cluster.
- **Advanced Networking** – Providing high levels of isolation and control at the workload level.
- **Full Stack Solution** – Everything you need in a single solution including orchestration, storage, logging, auditing, and governance.
- **Turnkey** – Faster time to value as it is production-ready on day 1.

> "We [at Apcera] focused on designing a platform that could run both cloud-native and legacy applications."
>
> – Henry Stapp, Director of Product Management, Apcera

> "We talk about being an enterprise-grade container management platform, but are really focused on enterprises that want to migrate legacy applications to the cloud, want to containerize their legacy applications."
>
> – Henry Stapp, Director of Product Management, Apcera

FIGURE 4 | Apcera Platform



Examples of results from the Apcera platform are:

SAP Hybris – Packaged E-commerce Application

Example of lift and shift

- Reduced infrastructure footprint by over 50% (24 servers to 7)
- 14 F5 load balancers no longer needed
- Significant cost savings
- Deployment now under 10 minutes
- Increase in the number of deploys from 4 to 8 per day
- Containers helped lower cost, increase agility

> "What we're seeing are tangible results when you apply this platform and you actually go through the modernization process. Whether it's a lift and shift or an incremental modernization, our customers are achieving tangible results from this platform."
>
> – David Williams, Principal, Williams & Garcia

Unpackaged Custom Application

Example of incremental rearchitecture of a legacy application

- Apcera was up and running within 1 week on 8 VMs (competitor took 35 VMs and 4-5 weeks to get installed)
- "We've done more development work in the past six months than the team did over the past three years."

About Apcera

The Apcera trusted cloud platform is a highly secure, policy-driven multi-cloud platform for cloud-native applications, containers, microservices and legacy applications. Apcera enables developers and DevOps teams to use any modern tool or software they want while giving IT and Operations teams the assurance that their infrastructure is safe and secure. With Apcera, companies can innovate at speed with full confidence and trust.

Global 2000 companies use Apcera to securely develop, deploy, orchestrate and govern diverse workloads across multiple cloud providers, resulting in lower cost, faster time to market, and mitigated risk. Apcera is headquartered in San Francisco. For more information, visit http://www.apcera.com, read the company's blog or follow on Twitter: @apcera.