advanced
People. Partnership. Solutions.

# Reimagining Technology Convergence to Embrace Digital Transformation

## Tackling legacy technology convergence at the programming layer is the last step towards converging around modern standards

Digital transformation is here; it is already a disruptive driving force in all facets of business. Market leaders understand that the Cloud is delivering agility and offering differentiation – and they are making it the foundation of their business plans. The right technology decision can create a platform that embraces an uncertain world and offers choices and new ways of working. Organizations are increasingly looking towards innovative technologies in order to take control and become a reinvigorated and connected business. They want to accelerate the adoption of digital to ensure they can respond quickly to changes in the business landscape and take advantage of all the benefits it offers. What is clear is that it is no longer just an option to incorporate digital technology into an organization - it's becoming a necessity.

### Technology Convergence

Technology convergence within an organization's technology stack has become essential for digital transformation and the main driver for the majority of today's legacy Application Modernization initiatives.

In most organizations with legacy applications, the majority of the layers within their technology stack have already been converged, including the infrastructure and database layers. What has yet to be converged is their application layer.

### The Operating System Layer

Over the last 15 years, most organizations have moved towards standard operating systems and hardware architectures in their data centers. Today, almost all organizations have standardized on Linux and Windows, and nearly all hardware specific operating systems have been eliminated. This convergence to commodity operating systems has enabled organizations to easily transition to hosted and Cloud operating environments.

Vendor-specific operating systems such as z/OS, OS/400, OpenVMS and Nonstop are still in use, but their numbers have declined dramatically. By converging operating systems, IT departments have been able to reduce the total cost ownership, to increase flexibility to rapidly scale with the business, and to decrease reliance on limited and expensive operational resources

# Reimagining Technology Convergence to Embrace Digital Transformation

with the specialized skills to run these legacy O/S platforms.

This drive to operating system convergence has meant that organizations with legacy applications running on non-commodity computing platforms have had to look at ways of moving these applications to Linux or Windows. The answer in many cases has been to undertake re-platforming, re-hosting or migration projects. The focus of which has typically been to leave the applications as unchanged as possible, but move them to a standard production platform.

There are several re-platforming solutions that can be utilized for a migration project, and they can all be separated into two main types. First, there are software conversion tools that can automate any changes required to the legacy code and data. This automation reduces project time and cost, avoiding the possibility of introducing errors that might occur if the work was completed manually. The second type of tool used is a re-platforming framework or container that sits natively in the Linux or Windows environment but provides support for the key system functions that the legacy application has been designed to use. A good example of this is CICS. Most IBM mainframe online applications have been developed to use the CICS transaction processor. When these applications are migrated to Linux or Windows in order to keep the application largely unchanged, it is important that there is a framework that provides the functionality of CICS and supports the CICS API already used in the legacy application This then avoids the need for vast re-engineering of the application. All good re-platforming vendors provide these types of migration frameworks for the various platforms they provide solutions for.

### The Database Layer

Another area where there has been a steady move to technical convergence is the database layer. Most organizations have now standardized their legacy transactional data on relational database solutions. Oracle, DB2 and SQL Server are the dominant data repositories for most transactional applications. This provides the significant benefits of making data more accessible and useable throughout the organization. This database technology convergence has enabled the revolution in Business Intelligence and Business Analytics and has become the foundation of the new generation of Big Data and Data Science initiatives.

### The Application Layer

The first wave of data modernization, historically, began with legacy extension. This took the form of a middleware modernization approach, employing ODBC and JDBC support to extend legacy, non-relational databases. In more recent years the approach has been to move onto a Service Oriented Architecture leveraging legacy application code for services to provide not only access to data but also business logic to manipulate that data before making it available to the client application or data warehouse.

Add to the mix that many older legacy applications were built on proprietary data models, such as ISAM (VSAM), or hierarchical databases (IMS, IDMS, etc). You can still find these data management products in use on the mainframe today, but often as organizations have re-platformed their legacy applications to Linux and Windows, they have taken the opportunity to migrate the application data to a standard relational database as part of the project.

As we look at legacy applications nowadays, we can see that there has already been a great deal of technology convergence in most areas of the technology stack. Operating systems have converged down to two main platforms, databases have converged down to a handful of relational database products and another convergence has quietly happened around user interfaces where browser-based UIs are now the standard for applications. The only layer of legacy applications that has been slow to converge around modern standards is the programming layer.

The majority of these legacy applications are not written in Java or .Net. In fact, the majority of legacy applications are written in COBOL. And there are still a number written in RPG and even less standard languages, such as Natural, Fortran, BASIC and Assembler.

# Reimagining Technology Convergence to Embrace Digital Transformation

For years the industry has talked about Application Modernization, but as we have discussed, as far as technology convergence is concerned, most of the effort to date has been toward operating systems and database modernization, and the standard approach to Application Modernization has been merely to provide interoperability between new and old applications. It is only now that organizations are starting to come to grips with true Application Modernization.

This pressure to reduce the dependency on older 3 GL programming architectures is prompting companies to undertake legacy application "deconstruction projects." Projects aimed at analyzing the applications in a legacy portfolio and breaking them down into the component parts to create specific modernization strategies for each part. As legacy portfolios can be so large, it's the old "eating the elephant one bite at a time" approach.

Using Application Discovery and Understanding tools, which have been honed to work with legacy application portfolios, organizations can begin to untangle the spaghetti of legacy application interdependencies that have built up over the decades. Combining this tools-based analysis with an understanding of application functionality and use, enables organizations to formulate plans to replace, re-platform, re-engineer, re-write or retire their applications. Compared to classic portfolio rationalization projects, legacy portfolio deconstruction projects identify how legacy applications are integrated and interact with each other and how these interactions can be supported as part of an Application Modernization strategy.

Inevitably, having analyzed the modernization options for a legacy application portfolio with the goal of achieving technical convergence of legacy systems to enable a digital transformation strategy, some critical applications will be identified where the underlying programming architecture needs to be modernized. In other words, legacy 3GL code will need to be converted to modern Object Oriented (OO) Java or .Net code.

One of the reasons this has not been a common undertaking in the past is that it has traditionally been fraught with problems. Automated 3GL transformation solutions have been around for a while, but they always ranged from highly automated approaches that yielded poor, difficult to maintain code that didn't make the grade as an OO end state, and therefore could be extended and easily integrated with newer applications. Alternatively, to achieve a true OO end state and a code base that was maintainable and extensible, the solutions were much less automated and required large amounts of manual effort, making this approach expensive and similar in time and risk to a rewrite project.

The good news is that demand for viable methods to automate the conversion of legacy 3GL code and produce a maintainable, extensible and OO end state, have meant that there are now code transformation solutions on the market that actually achieve these goals. The secret has been introducing steps that automate the abstraction of the legacy code to an intermediate model, then taking that model to a common OO modeling environment ,such as UML, and then generating the new

# Reimagining Technology Convergence to Embrace Digital Transformation

application from the OO model. In this way, the re-architecting is done at a model level rather than syntax level and this approach produces a viable end state for the converted application.

Clearly, this approach has to be language specific because the initial automated conversion of the 3GL to an intermediate model requires tools that are tuned to the 3GL being transformed. This means that this option is not available to every legacy 3GL programming language. In fact, companies should be cautious of vendors offering solutions that can transform "any" 3GL to Java or .Net. as specialization is key in the language transformation business.

The good news is that this sophisticated approach to code transformation is available for COBOL, which represents by far the largest amount of mission-critical legacy application code in use today. It also represents one of the largest programming skill sets about to leave the workforce in the next few years.

To summarize, technology convergence for legacy applications is not only a logical and now achievable next step for digital transformation initiatives, it is a necessity for organizations faced with losing skilled legacy support resource and tribal knowledge.

Digital transformation dictates that companies have agile and adaptive applications, typically Cloud delivered, with appealing systems of engagement that provide rapid and seamless integration to custom-made systems of record. Technology convergence is mandatory to delivering these essential capabilities within digitally mature organizations. Now is the time for organizations to address this issue.

## More information

**w** oneadvanced.com/us
**t** 770 933 1965
**e** hi@oneadvanced.com

3200 Windy Hill Road, Suite 230 West, Atlanta, GA 30339