



End of the Road for App Integration Agony

More and more organizations looking to hop on the cloud wagon are facing application and data integration roadblocks. Getting around them is difficult but possible—and a step-by-step approach can help.

[EDITOR'S NOTE](#)

[NINE QUESTIONS FOR INTEGRATING IN THE CLOUD](#)

[APPLICATIONS WITHOUT BORDERS](#)

[USE DEVOPS TO HEAD OFF CLOUD COMPLICATIONS](#)

Rx for Integration Pain

IN A 2014 survey, readers of SearchCloud-Applications and its sister sites identified application and data integration as the biggest challenges in software-as-a-service adoption. Industry researchers agree. Gartner analyst Eric Knipp calls app integration the No. 1 barrier to cloud deployment success, and 451 Research ranks it as the second-most-acute cloud pain point, beaten only by security.

Integration pain isn't a cloud adoption deal breaker for most companies, though. Otherwise, the cloud applications market wouldn't be so huge—\$30 billion in cloud application subscription reviews in 2014. And it wouldn't be growing so quickly, to the tune of \$67 billion in projected subscriptions in 2018, according to IT researcher Apps Run the World.

Competitive pressure spurs that growth, and integration issues have to be handled. A starting point could be answering the [nine questions](#) on cloud application integration in this

guide's first article, written by Valerie Silverthorne. Each query comes with an expert how-to or why-to discussion.

An overall cloud app integration strategy should include a unifying plan for adoption projects, writes consultant Tom Nolle in his tip on [integrating applications](#) across

Industry researchers agree that application and data integration are the biggest challenges in software-as-a-service adoption.

cloud boundaries. Without a step-by-step integration approach, each cloud app adoption project has to start from scratch. Nolle advises organizations to codify best practices for each application integration component or layer.

Making applications play nicely together is only the means to cloud application integra-

HOME

EDITOR'S NOTE

NINE QUESTIONS
FOR INTEGRATING
IN THE CLOUD

APPLICATIONS
WITHOUT BORDERS

USE DEVOPS TO
HEAD OFF
COMPLICATIONS

tion's goal, [enabling workflow](#). "The purpose of cloud application integration is to connect workflows through components," writes Nolle in this guide's third article. Beware of tool vendors that want to tie integration projects to their cloud services and not the project's targeted workflow goals, he advised.

Planning provides the best relief from

chronic integration pain. If your organization is feeling an ache we haven't addressed in these pages, perhaps we can help. Write to me at jstafford@techtarget.com. ■

JAN STAFFORD

Executive Editor, SearchCloudApplications

[@jlstafford](#)

Nine Questions for Integrating in the Cloud

HOME

EDITOR'S NOTE

NINE QUESTIONS
FOR INTEGRATING
IN THE CLOUD

APPLICATIONS
WITHOUT BORDERS

USE DEVOPS TO
HEAD OFF
COMPLICATIONS

THE GOOD NEWS is your organization has data and applications in several clouds. The bad news is very little of it is integrated.

But you're far from alone. While market research firm Infonetics said 74% of the companies they surveyed had a [hybrid cloud strategy](#) by the end of 2014, the integration part of that strategy remains murky at best for many companies.

We asked integration vendors, analysts and industry experts what nine questions companies should be asked when putting a hybrid cloud application integration strategy in place. Here is their collective wisdom:

1. What are you not going to move to the cloud? It seems obvious, but there are legacy applications that are doing just fine in the data center, said SnapLogic's vice president of marketing Darren Cunningham. "You need to make sure you have an

integration plan for those systems," he said. "There are a plethora of applications behind the firewall that are going to continue to stay there. So you have to be sure you have an integration strategy that includes those applications."

- 2. Where will most of the future development take place?** To integrate and ultimately monitor an application, start by determining where it's going to be built, advised Forrester Research analyst Dave Bartoletti. "Apps you build in the cloud look different from other applications, so when it comes time to integrate them, or eventually monitor them, companies need to take that into consideration. A lot of companies don't have the tools in place for this."
- 3. What is the use case?** "That's the first question to ask," said Ajay Gandhi,

Informatica's vice president of cloud marketing. "Do I want to connect to [on-premises data](#), and if I do, are there complicated APIs and databases? A lot of the reason you want an integrated hybrid cloud is to take care of the back-end integration. You shouldn't be in the business of monitoring the code. That's what the integration solution is for."

Tom Nolle, president of consulting company CIMI Corp., believes that customers and vendors underappreciate the difficulties involved in true integration. Businesses must have a laser-sharp focus on what the goals really are, he said. "Companies [jumping on the cloud bandwagon] are just taken in by too much publicity and not enough benefit case."

4. What kinds of applications are involved?

In application development, things are not straightforward anymore, warned Chris Purpura, vice president of business development for MuleSoft. "Applications are becoming composite," he said. "That makes things trickier. What happens when an

application has to ping 10 other external systems to get information back? That's another thing to think about."

5. Who will manage the integration? Forrester's Bartoletti said companies need to clearly identify the people or groups that will be charged with managing the process because, like the hybrid cloud, it might be a hybrid group of traditional IT employees, DevOps people and perhaps even folks from the business side. SnapLogic's Cunningham stressed the potential for do-it-yourself integration, where there wouldn't be a single group in charge of integration. "We're seeing customers letting users drive the integration. The users fill out the use case, submit the forms and then are provisioned to do it themselves. It's the reality of self-service."

6. How quickly can this happen? Customers don't need to reinvent the wheel when thinking about integration. Many vendors—including Informatica and SnapLogic—have developed preconfigured templates for common use cases that drop easily in place.

“Customers should ask what work has already been done for them,” Gandhi suggested. Informatica has prebuilt content for customers to choose from, which streamlines the integration process. And Cunningham said SnapLogic’s templates and best practices have turned a job that took several weeks into an integration task that can be completed in a day or two.

7. What’s the next application? “You really have to take the long view, not just integrating one application but the next—and the next,” warned Cunningham. “That one application is going to have a whole other set of apps to eventually integrate with. This isn’t about point-to-point integration but about multipoint integration.”

8. Can I make do with what I have? This is a critical question to ask, Forrester’s Bartoletti said. “Should I buy the [best of breed](#) and deal with integration later or make what

I have work?” Whichever way a business goes, time is of the essence and integration is key. Bartoletti suggested always asking current vendors whether existing functionality can be extended some to aid in integration.

9. Should you trust the vendor? Nope. Cunningham didn’t hold back: “Don’t believe what the cloud application vendor is telling you if he says to push the integration down the road or he makes it seem trivial because his company has great service and/or an open platform. You can’t wait on integration. You just can’t.”

And customers should be skeptical when an on-premises vendor starts bragging about how his product will easily extend into a hybrid cloud environment, Gandhi said. “Those products have a ‘vener of cloudiness’ as we call them,” he said. “You want to get to the reality of it all.”

—Valerie Silverthorne

Applications Without Borders

HOME

EDITOR'S NOTE

NINE QUESTIONS
FOR INTEGRATING
IN THE CLOUD

APPLICATIONS
WITHOUT BORDERS

USE DEVOPS TO
HEAD OFF
COMPLICATIONS

INTEGRATING APPLICATIONS DISTRIBUTED in public and private clouds in a case-by-case manner is an invitation to reliability problems and lost savings. Architects and planners need to understand the layers of cloud integration, plan an approach at each layer and ensure their integration strategies meld with one another and with application lifecycle management.

Integrating applications in the cloud starts with interfaces and addressing. In order for an application to cross a cloud boundary, it has to be made up of discrete components, representing separate interfaces. Each of these components has to be addressable for work to flow within the application. The application is only a collection of services at this highest level of integration, so it doesn't matter whether some or all of it is hosted in the cloud.

Component addresses are registered in a directory such as universal description,

discovery and integration or domain name system. At the highest level of integration, it's essential that application directories be accessible by the processes steering the workflow. A service bus, when used, must be able to access the directory to find the next component in the workflow.

That's normally easy since the components don't address one another. If components directly reference other components, then architects may need to take special steps to establish directory access for each component and ensure the address space used by one cloud is compatible with the others. This can be an issue if users employ private IP addresses, because these IP addresses can't be referenced, except on [virtual private networks](#) (VPNs). Architects who use such addresses will have to add all of their cloud resources onto a common VPN to open connections among components.

LAYERED CONSIDERATIONS

The second level of integration is deployment and management practices. Launching an application and its components in a cloud generally requires creating a load or machine image and using a management API to call for it to be loaded. The details of the image and the API will likely vary among public cloud providers, between public and private clouds, or between the cloud and the data center. Variation will have to be accommodated in the tools used or through manual processes.

There are two promising paths to addressing these two layers at once: integration platform as a service (iPaaS) and orchestration. The evolving idea of iPaaS is when a cloud service takes responsibility for deploying application components and integrating them as needed. Dell Boomi or Mulesoft's CloudHub are examples of iPaaS tools. The other path is orchestration, a higher-layer development and systems operations (DevOps)-like function that can be used to collect the different DevOps scripts or models for multiple clouds and organize them to permit deployment across cloud boundaries.

OpenStack Heat is an evolving open source orchestration engine and CloudFormation template language. OpenStack orchestration products include Ubuntu's Juju, a service orchestration tool for cloud deployments. In commercial products, consider those based on the leading orchestration specification,

There are two promising paths to addressing the two layers of application integration at once: integration platform as a service, or iPaaS, and orchestration.

the OASIS TOSCA Standard for Managing Applications across Clouds. Orchestration tools from Fujitsu, Hewlett-Packard, Huawei, IBM, SAP and others use TOSCA's templates to move applications from one cloud to another and orchestrate them after the move.

Both approaches have current limitations on the scope of their processes, and both can require considerable work on the part of the cloud architect. The best strategy is to review all of the available options, starting with a list

[HOME](#)[EDITOR'S NOTE](#)[NINE QUESTIONS
FOR INTEGRATING
IN THE CLOUD](#)[APPLICATIONS
WITHOUT BORDERS](#)[USE DEVOPS TO
HEAD OFF
COMPLICATIONS](#)

of public cloud provider candidates and private cloud [software stacks](#). IT teams can return to their lists should there be changes in service levels or pricing down the line. Users report that iPaaS tools are further along than orchestration tools, so the planned evolution of orchestration tools and features should be explored carefully if that option is considered.

The key point for both iPaaS and orchestration is that there are two layers. The first considers the applications deployed as a collection of software-as-a-service components linked with APIs. The second focuses on deploying and managing the application components on the optimum platforms. This two-layer model should be explicitly acknowledged in an IT team's tool selection. If not, there's a risk that a new cloud option, or the need to integrate a data center-hosted component, will require considerable manual tuning.

INTEGRATION COMPLEXITIES

At the bottom of the deployment and management layer is the management of cloud

application quality of experience. Both iPaaS and orchestration tools vary in their ability to support FCAPS (fault management, configuration, accounting, performance and security) on a per-cloud basis. If it's necessary to manage user quality of experience closely for an application or group of applications, look for a tool that offers strong support for the collection and integration of management data. In particular, look at how failures of a host or network connection will be reported. It's impossible to manage a response to a problem architects can't detect, and cloud services vary significantly with respect to their reporting. In some cases, it may be best to integrate management components with functional components to be deployed so full management can be exercised directly at the component level.

This is also where the tie between cloud integration and application lifecycle management (ALM) is likely to be a factor in planning. It's not only important that ALM addresses components in realistic data-driven scenarios but also that it address the variations in workflow that could occur as a result of differences in how components are spread across cloud

[HOME](#)[EDITOR'S NOTE](#)[NINE QUESTIONS
FOR INTEGRATING
IN THE CLOUD](#)[APPLICATIONS
WITHOUT BORDERS](#)[USE DEVOPS TO
HEAD OFF
COMPLICATIONS](#)

providers. This can add a new and complex dimension to ALM testing at all levels if application components can be spread across multiple clouds. Users will often elect to tie specific groups of components to specific implementation options in order to reduce the variability and manage the integration more efficiently.

The complexity of integrating applications across cloud boundaries makes all-purpose frameworks appealing. These middleware suites have the tools for integrating heterogeneous applications, automating processes, scaling applications and more. There are many

to evaluate, such as JBoss Enterprise Middleware, BEA WebLogic and Oracle Fusion Middleware suite. I've worked mostly with Oracle Fusion and found it, particularly the business process execution language process manager and service bus, to be a good framework for building cloud applications that can be integrated.

There are often benefits to crossing cloud boundaries with applications, but there are also often costs and risks. Architects should review both paths for addressing integration carefully and recalibrate their processes whenever cloud pricing or service levels change. —*Tom Nolle*

Use DevOps to Head Off Cloud Complications

THE FLEXIBILITY THAT cloud promises can have a high cost. Integration with applications that are moved into the cloud is complicated. Cloud architects should think of integration in workflow terms, align the scope of tools and practices with their application goals, focus extra care on the special transition points and look ahead to how business changes could dictate integration changes.

The purpose of cloud application integration is to connect workflows through components. This connection process should be viewed in a technology-neutral way because most businesses will use the elasticity of the cloud, moving things in and out on demand. Thus, cloud architects want cloud-compatible tools for workflow integration and not cloud-specific tools.

Most cloud integration tools will support a design, deploy and manage progression that relates to the progression familiar to

cloud architects; it's known as DevOps, development and operations. The design phase is aimed at mapping workflows to components, the deployment phase at deploying the components and connecting them to the workflow, and the management phase at supporting life-cycle management and problem management while the application is in service. A [cloud integration strategy](#) has to accommodate all of these phases. Manual integration will be necessary at some point if a given tool doesn't do that.

TARGETED SUPPORT

While all integration tasks include those three points of progression, what is supported in each phase will vary depending on the type of cloud application being targeted by the tool provider. For example, if the goal of cloud application integration is to move pieces of

current applications to the cloud, then the design process is really a matter of mapping components, connecting workflows and managing policies to handle any necessary dynamic positioning of components. If the goal of the integration is to create new cloud-friendly applications that will mean a restructuring of business processes and even new development, then the design process will have to be very sophisticated.

A good example of this issue is projects that address mobile-worker productivity enhancement through more event-driven business processes. This involves the creation of new workflows and new applications, though most of the components may already be available. Simple mapping tools aren't likely to be helpful here, because workflow design is most often a software architect task. It's possible to address this extra complexity through software development tools or extend the design features of integration tools. Deployment, similarly, may be handled through repurposing DevOps tools like IBM's Service Virtualization or by selecting an integration tool designed to support all three phases, such as Jitterbit.

The three evolutionary phases of cloud integration create two transition points that require special attention. The design-to-deploy transition is similar to the transition intrinsic to DevOps tools—the goal is to faithfully create what the design phase mandates. However, it's easy for the design-to-deploy transition to become brittle, meaning that practices and tools may be dependent on the specific needs of one or more early target projects.

One possible way to avoid brittle transitions between integration phases is to consider as wide a range of applications as possible. Many users find this leads to projects without boundaries and vast overruns on requirements and costs. A better approach is to try to generalize the goals of these transitions based on the early projects. Think of the design-to-deploy transition in terms of applying data mappings and business policies from the design phase, with the deployment phase focusing on specific issues of platforms and scaling or failover.

The transition from deployment to management can be complex because it often necessitates looking back to the design phase. The management phase is a combination

of [application service-level agreement](#) support and application lifecycle management (ALM). The former requires that deployment policies reflect user quality of experience, which means that real-time event-driven interaction is necessary. The ways that this interaction is then fed back to deployment create an interdependency, a two-directional flow. On the ALM side, the workflow models and policies are used to establish goals that ALM processes enforce and validate when changes are made. This means that design-phase activity jumps over deployment to organize management processes, making the true model more complicated than the general linear progression that applies in most cases.

BACK TO THE FUTURE

ALM is a good introduction to the final point, which is future-proofing the integration process. Architects point out that cloud integration may be the topic that leads most often to overspecialization when selecting tools and

systems. In particular, there's a tendency to see the deployment phase as the main issue and fail to consider workflow design on the front end or ALM at the back. There is a value to adopting a complete cloud deployment and integration suite, obviously, because these are most likely to track application evolution in the broadest sense.

The problem users report is that the cloud suites tend to be focused on the vendor's own development and middleware tools. IBM, Microsoft and Oracle often share accounts, so it may be difficult to achieve harmony through a suite unless you go to an independent vendor. There is also the option of supporting the three phases of design, deployment and management with phase-specific tools.

It comes back to those transition points. Weaknesses in cloud integration tools and practices will show up at the phase boundaries, so no matter what approach cloud architects take, it's smart to first look closely at these points and resolve issues before they grow.

—Tom Nolle

HOME

EDITOR'S NOTE

NINE QUESTIONS
FOR INTEGRATING
IN THE CLOUD

APPLICATIONS
WITHOUT BORDERS

USE DEVOPS TO
HEAD OFF
COMPLICATIONS

VALERIE SILVERTHORNE is a writer and editor with nearly 30 years of experience covering business, trade, technology, real estate and lifestyle trends. She works as a freelance writer and frequent contributor for TechTarget sites. Email her at valerierice@gmail.com.

TOM NOLLE is president of CIMI Corp., a consulting company specializing in telecommunications and data communications. He is also the publisher of Netwatcher, a journal addressing advanced telecommunications strategy issues, and writes for many TechTarget sites. Email him at tnolle@cimicorp.com.

STAY CONNECTED!



Follow [@SearchCloudApplications](https://twitter.com/SearchCloudApplications) today



End of the Road for App Integration Agony is an e-publication of SearchCloudApplications.com.

Scot Petersen | Editorial Director

Jason Sparapani | Managing Editor, E-Publications

Joe Hebert | Associate Managing Editor, E-Publications

Jan Stafford | Executive Editor

Brein Matturro | Managing Editor

Linda Koury | Director of Online Design

Neva Maniscalco | Graphic Designer

Doug Olender | Publisher

dolender@techtarget.com

Annie Matthews | Director of Sales

amatthews@techtarget.com

TechTarget

275 Grove Street, Newton, MA 02466

www.techtarget.com

© 2015 TechTarget Inc. No part of this publication may be transmitted or reproduced in any form or by any means without written permission from the publisher. TechTarget reprints are available through [The YGS Group](http://TheYGSGroup.com).

About TechTarget: TechTarget publishes media for information technology professionals. More than 100 focused websites enable quick access to a deep store of news, advice and analysis about the technologies, products and processes crucial to your job. Our live and virtual events give you direct access to independent expert commentary and advice. At IT Knowledge Exchange, our social community, you can get advice and share solutions with peers and experts.

COVER ART: THINKSTOCK