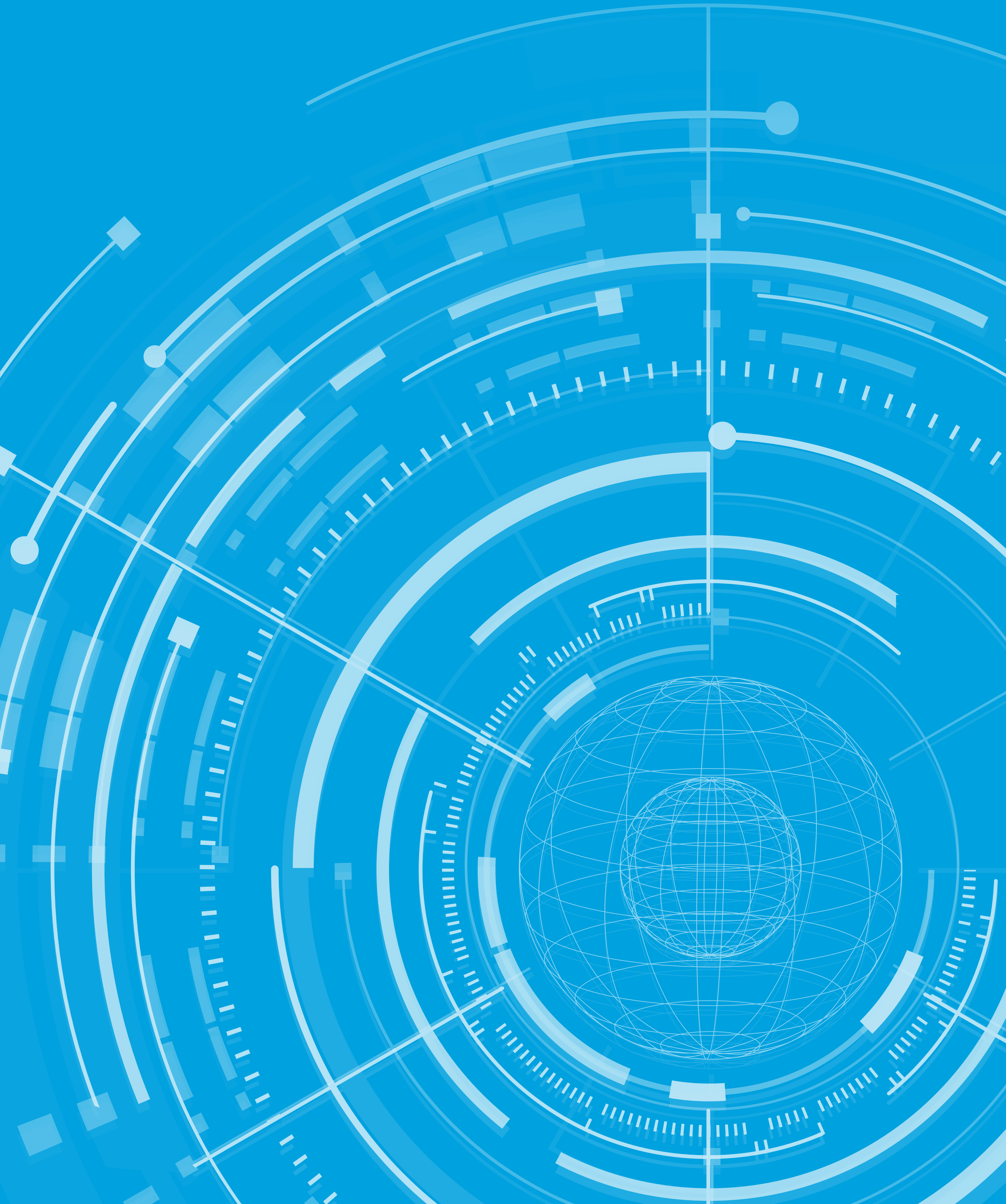
MuleSoft®

WHITEPAPER

APIs and DevOps

Great alone, better together



Contents

Executive summary

DevOps has become not only trendy, but an important rethink of the IT culture at many organizations. And with good reason because the benefits to adopting DevOps are well-documented. A recent study, conducted by Utah State University and sponsored by Puppet Labs, revealed that DevOps practices can boost IT performance and ultimately improve a business's bottom line. High IT performance correlates with DevOps practices like proactive monitoring and continuous delivery and integration.

Interestingly, the report also showed that the longer an organization uses and improves upon DevOps practices, the better it performs, which can lead to improved performance for the entire company. "This is where DevOps is different. It isn't just IT, it's the practice of IT," says Nicole Forsgren Velasquez, an assistant professor at Utah State University and one of the researchers on the study.

But it isn't just DevOps practices that improve the bottom line for companies. Similarly, the ability to create and implement a successful API strategy has become a critical competency for modern IT teams. According to MuleSoft's 2017 Connectivity Benchmark Report, 94% of IT leaders confirm they can deliver products and services to market faster with APIs. And an API strategy is critical to extracting maximum business value out of what IT teams are creating.

DevOps makes it easy to create and release software deployments. But when it's easy to build software, silos within organizations will replicate work without thinking about maximizing efficiency or about how to expose those assets to others for key business functions. What's needed is not only an emphasis on production but an emphasis on consumption that focuses on reusability and the exposing of existing assets to create new capabilities.

In this whitepaper, we'll talk about how organizations can shift from artifact production (the focus of DevOps), to consider a how these produced artifacts are consumed. One way this shift happens is through an API strategy called API-led connectivity. We'll take a look at how this change can be accommodated by the capabilities of MuleSoft's Anypoint Platform.

Introduction

Organizations today tend to take one of two paths when it comes to creating innovative experiences for their customers, employees, and partners. One path prioritizes rapid iteration over ‘getting it right the first time’. It is taken by companies like Amazon, who are seen as leaders in innovation not only in their own industry but across all organizations. As a rule, Amazon does 136,000 software deployments per day.

The second path prioritizes stability above all and is best exemplified by a global bank that MuleSoft now works with. Early on in our engagement, this customer shared that it once took them three months and around $1.8 million to deploy one database field name change simply because they didn’t have visibility into the impact of that change. Obviously, that is a crippling state.

Adopting DevOps can seem unnatural to organizations that don’t have a culture of automation and that aren’t set up, either for security and compliance reasons or because of company culture, to deploy software thousands of times per day. Adopting DevOps, therefore, becomes an iterative process; it’s necessary to adjust the practice to fit the needs of the company.

Not every company needs to — or can — deploy code to production every few seconds like Amazon. But there is a way for any company to take the first path, even for the most regulated company with the oldest of legacy systems. It all depends on approaching the problem in the right way and with the right tools.

The danger of following the Amazon path is that it becomes very tempting, particularly with the power that DevOps unleashes, to simply create piles upon piles of software assets without thinking about how they might provide value for the wider company. Not only are these artifacts replicated in business silos, killing business productivity, but they are locked away from other parts of the business, who might be able to make use or and extract further value out of them.

Successful DevOps adoption has to include both an accelerated and efficient production model and an efficient, balanced consumption model. Companies that successfully implement DevOps should be stewards of a new operating model with a virtuous cycle in which teams produce reusable assets in a manner visible to central IT, which then enables consumption of those assets by making them discoverable and available for self-service, all the while monitoring feedback and usage.

In this model, those developing applications and other assets must change their mindset to think about producing assets that will be consumed by others in the business. And, in turn, central IT needs to change its mindset to an enabler of reusable, self-service consumption. Any assets built need to be discoverable and developers need to be enabled to self-serve them in projects. The virtuous part of this production-consumption cycle is to get active feedback from the consumption model along with usage metrics to inform the production model.

DevOps: A profound cultural and technological shift

DevOps has emerged as a practice that focuses on minimal disruption and maximum reliability; faster time to market is also an artifact of automated release processes that DevOps enables. There are thousands of articles and even industry conferences on the topic of DevOps itself, but at MuleSoft, we see one consistent theme that any organization thinking about adopting DevOps needs to consider: DevOps is a profound shift in IT culture, a shift the demands a rethink on IT processes, technology, and the way people work. The consumption model described above is an equally comprehensive shift.

Many start the journey to DevOps by selecting technology that helps development teams remove waste and inefficiency from their software development lifecycle. A common first step is to establish continuous integration processes, taking advantage of tools like Maven and GitHub to automate repeatable aspects of the SDLC where human intervention is typically error- prone. The next step is to produce and deploy software artifacts efficiently, creating a complete pipeline that takes code from concept to production in a repeatable and secure fashion and allows deployments to occur during business hours. Building this pipeline, a blend of technology and quality assurance processes, is the goal of most DevOps practices.

That said, code produced via this pipeline will not necessarily provide maximum business value unless it is given full visibility to the rest of the organization. DevOps can only take the code production machine so far; in order to scale, and more importantly, scale in a sustainable and secure way, IT needs to shift its thinking both to a model predicated on consumption and reuse. This is facilitated through the use of well-designed and built, documented, and visible APIs. This further increases the innovation possibilities DevOps provides and furnishes reusable assets that can grow, and change with ease depending on who wants to use them and what for.

Finally, processes and technology are nothing if your teams do not adopt them and think about them in the right way. If your team is not embracing agile delivery and thinking about test-driven development, then trying to implement DevOps will not be successful. An example of this is with one of our customers who thought about DevOps the same way they thought about delivering large energy projects; for example, in a very waterfall-like fashion. But to deliver technology changes at the speed the market demands, software deployments have to happen faster than a construction project, so the company culture had to shift pretty dramatically.

The benefits of adopting DevOps

There are myriad benefits to adopting DevOps. Many view DevOps as a way to increase speed to market, getting new products out the door quickly without sacrificing the quality of those products, because the testing, QA, security and operation of code is baked into the pipeline itself. By removing some of the inefficiencies and waste in the process, the goal is to reduce the cost of development of new products and services, iterate rapidly on new business priorities and develop innovations that customers actually want more quickly.

Organizations are looking to DevOps to serve as a competitive advantage for their companies, so they can adapt to new changes in the market, and leverage their expertise to address customer needs and deliver innovative solutions to market with great speed.

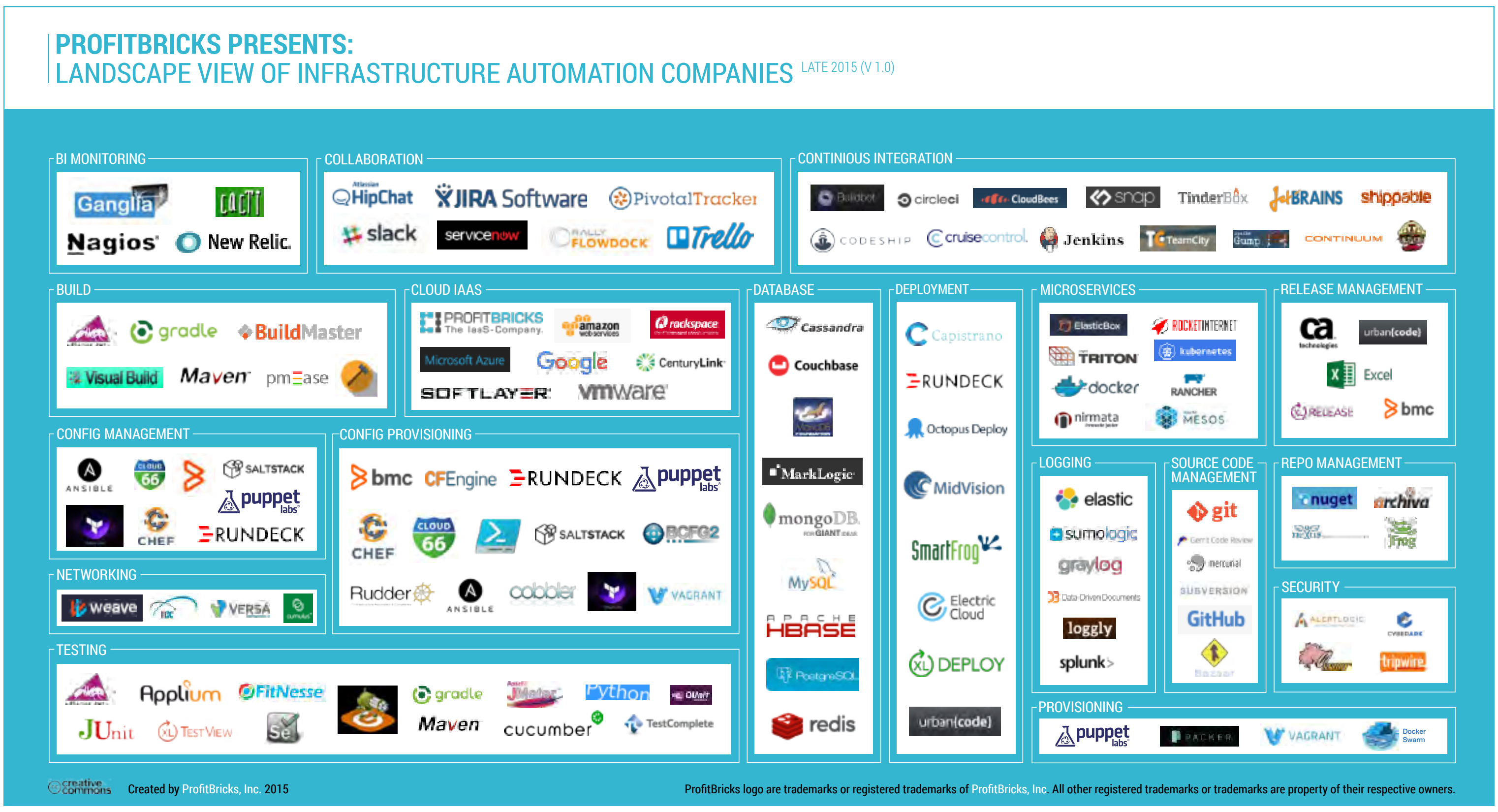

PROFITBRICKS PRESENTS:
LANDSCAPE VIEW OF INFRASTRUCTURE AUTOMATION COMPANIES LATE 2015 (V 1.0)
BI MONITORING
COLLABORATION
CONTINIOUS INTEGRATION
BUILD
CLOUD IAAS
DATABASE
DEPLOYMENT
MICROSERVICES
RELEASE MANAGEMENT
CONFIG MANAGEMENT
CONFIG PROVISIONING
LOGGING
SOURCE CODE MANAGEMENT
REPO MANAGEMENT
NETWORKING
SECURITY
TESTING
PROVISIONING
Created by ProfitBricks, Inc. 2015
ProfitBricks logo are trademarks or registered trademarks of ProfitBricks, Inc. All other registered trademarks or trademarks are property of their respective owners.

Figure 1: DevOps technologies on the market today

Figure 1 illustrates some of the technologies that we at MuleSoft have seen emerge within our customer base for customers pursuing DevOps practices. They fall into numerous buckets, from development and test, to deployment and monitoring, down to security and collaboration. Operations solutions like ServiceNow or Remedy for IT Service Management and JIRA for issue tracking are also common. Many of these technologies are open-source, or at least have open-source versions, so it's very easy for our customers to get started relatively quickly in building out the foundations of their DevOps practice.

How to use Anypoint Platform to set up DevOps in your organization

This whitepaper will help you conceptualize a generic rationale for DevOps adoption and how Anypoint Platform's capabilities can assist across a number of those categories.

First, we will consider the notion of build automation. As you are building your application, how do you create a recipe for the application to be reassembled correctly every time it needs to be taken apart for changes and updates?

Applications built on Anypoint Platform (Mule apps) are often composite applications, containing multiple connectors, multiple file systems, artifacts like .jar files, databases, and even multiple tests. All of those connectors, databases, and tests have specific configurations. If you have to alter those configurations every time you make a change to the Mule application, that is going to slow you down, and that's also very error-prone. MuleSoft customers use built-in and supported build automation tools like Maven to address these concerns.

The next step is test automation. Most developers wait until their code is complete to author tests. How, instead, do you think through the emerging best-practice of test-driven development, where you're actually authoring tests before you write a single line of code?

As MuleSoft started creating solutions for our customers, we learned that as you're building tests out to Salesforce or any other SaaS application, an active connection is required if you want to use real data. This is a pain point if a developer is doing this work on their laptop and can't maintain an active connection. We created MUnit, the MuleSoft integration testing framework, to help our developers and our customers create tests with mock data; and, in fact, both MUnit, and JUnit, a common Java testing framework, are used quite often within our customer base to deliver test automation successfully.

Version control is something that most developers today are very comfortable with. Developers know how to commit code across a disparate set of teams so they can drive parallel development and get instant visibility into changes that are not compatible with what others have been doing.

MuleSoft enables version control best practice via extensibility to GitHub, SVN, Team Foundation, and many others version control systems commonly used within enterprises.

Then, finally, there is deployment automation. As with creating a recipe for how an application is built, a recipe for how applications are deployed should also be created. Which environments should they be deployed to? How do you promote applications across environments: dev, QA, performance, security, stage, and production? How do the configurations change by environment? How do you create rules to block software from being deployed if it fails critical tests? Jenkins is a common solution that our customers use to deliver on this aspect of the pipeline.

Great DevOps practices focus on one additional element: reuse. As an asset is moving through the pipeline, how can you ensure others are aware of it, can access it securely, and know how to use it effectively? The API-led connectivity model is very informative here, as most developers are comfortable using Developer Portals and requesting API client keys and secrets. Industry trends suggest organizations are building an automation pipeline across the entire lifecycle of their APIs; there are a number of technologies across that lifecycle that have emerged as best practice, many of which MuleSoft supports and our customers are already leveraging today.

Lifecycle stage	Stage activities	Anypoint Platform components
Design	• Create an API specification • Define data types • Add security patterns	• API designer • Anypoint Exchange
Simulate	• Describe examples Author error messages	• Mocking Service NOTE: With a mock service, frontend developers can begin crafting the user experience before the API is implemented, allowing parallel development and reducing time to market
Collect feedback	• Create on-ramp • Publish portal	• Developer Portal • Anypoint Exchange
Validate	• Create runnable tests • Orchestrate API calls	• Anypoint Studio • Anypoint Exchange
Build	• Import API spec • Manage dependencies • Compose with connectors and templates • Static config analysis • Link to version control system	• Anypoint Studio • Anypoint Exchange
Test	Author unit and functional tests Define pass/fail criteria	• MUnit • Anypoint Exchange
• **Deploy** • **Secure** • **Promote**	• Deploy artifacts • Promote across environments • Apply policies • Control access	• Anypoint CLI • Anypoint Platform • Runtime manager • API manager
• **Operate** • **Secure** • **Monitor** • **Manage** • **Analyze**	• Measure SLAs, monitor utilization, adjust resources • Track performance against KPIs	• Runtime manager • Management agent • Anypoint Analytics

DevOps and API lifecycle capabilities on Anypoint Platform

Anypoint Platform provides a unified suite of design, management and runtime capabilities to make it easy for organizations to address the full API lifecycle. The components of Anypoint Platform guide developers from design to implementation to production and beyond—enabling discovery and self-service.

Each of the steps described above has certain aspects that can be automated. Most developers are familiar with the build stage; work done in Anypoint Studio can be easily plugged in to your continuous integration pipeline, leveraging some of the technologies that were mentioned earlier: GitHub for version control, Maven for dependencies management and build automation, JUnit and MUnit for test automation, and JIRA or ServiceNow for issue and service management. Customers with Microsoft investments building Mule apps often leverage Team Foundation and Visual Studio.

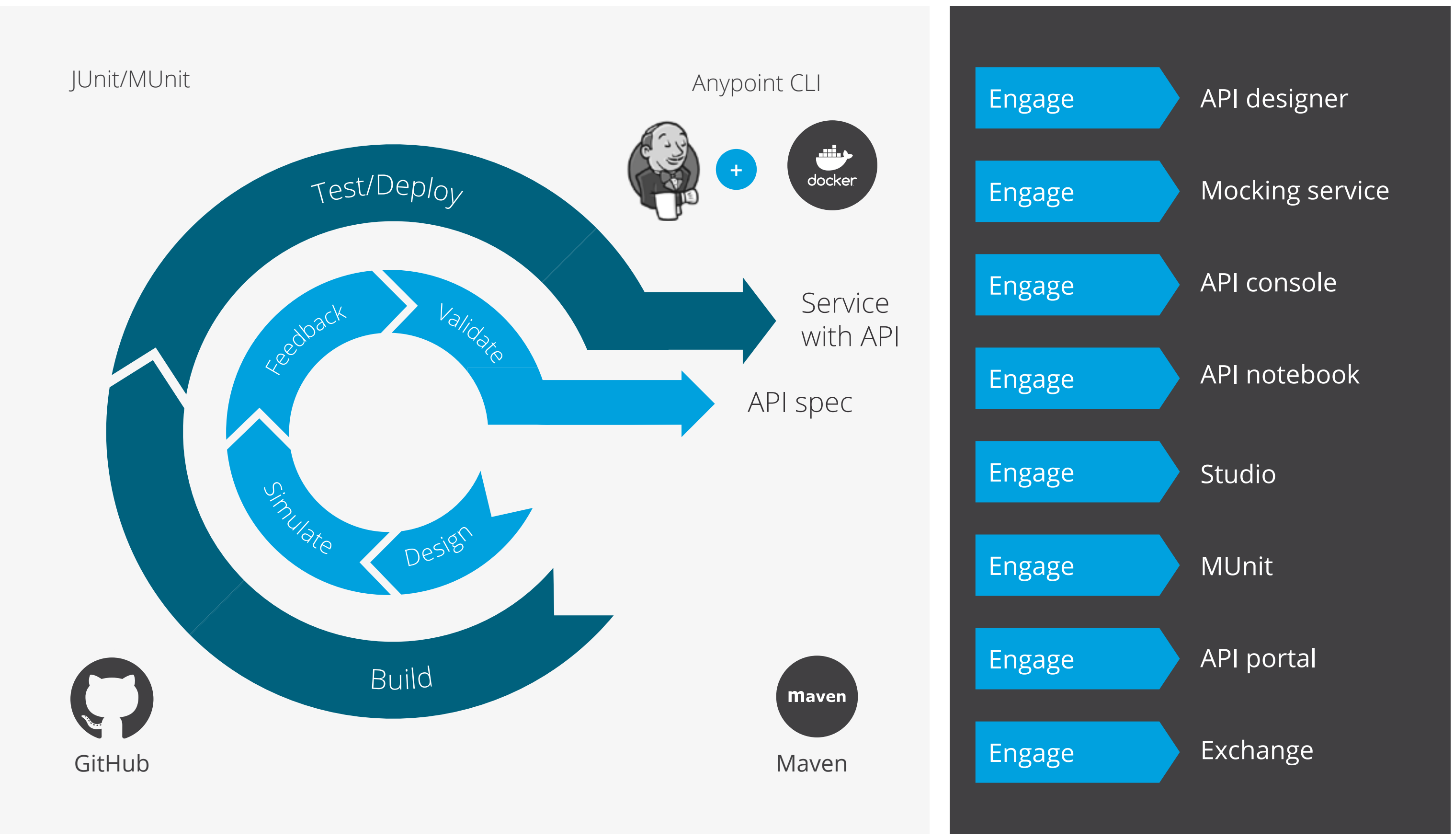


Figure 2: Design and implementation lifecycle

When the code is ready to be shipped, MuleSoft customers often leverage Jenkins, Puppet, Chef, or HP ALM for both deployment automation and the orchestration of release processes across a number of different environments: dev, test, stage, and production. Puppet and Chef can also be used to automate the installation and configuration of Mule runtimes. Docker is rapidly gaining popularity to reduce the complexities of continuous deployment, and the Mule runtime can be shipped in a Docker container. And through the management agent, Anypoint Platform for APIs and Anypoint Platform CLI, users can further automate processes using configuration automation and management tools such as Puppet, Chef, Ansible, and Salt, instead of the Anypoint Platform user interface.

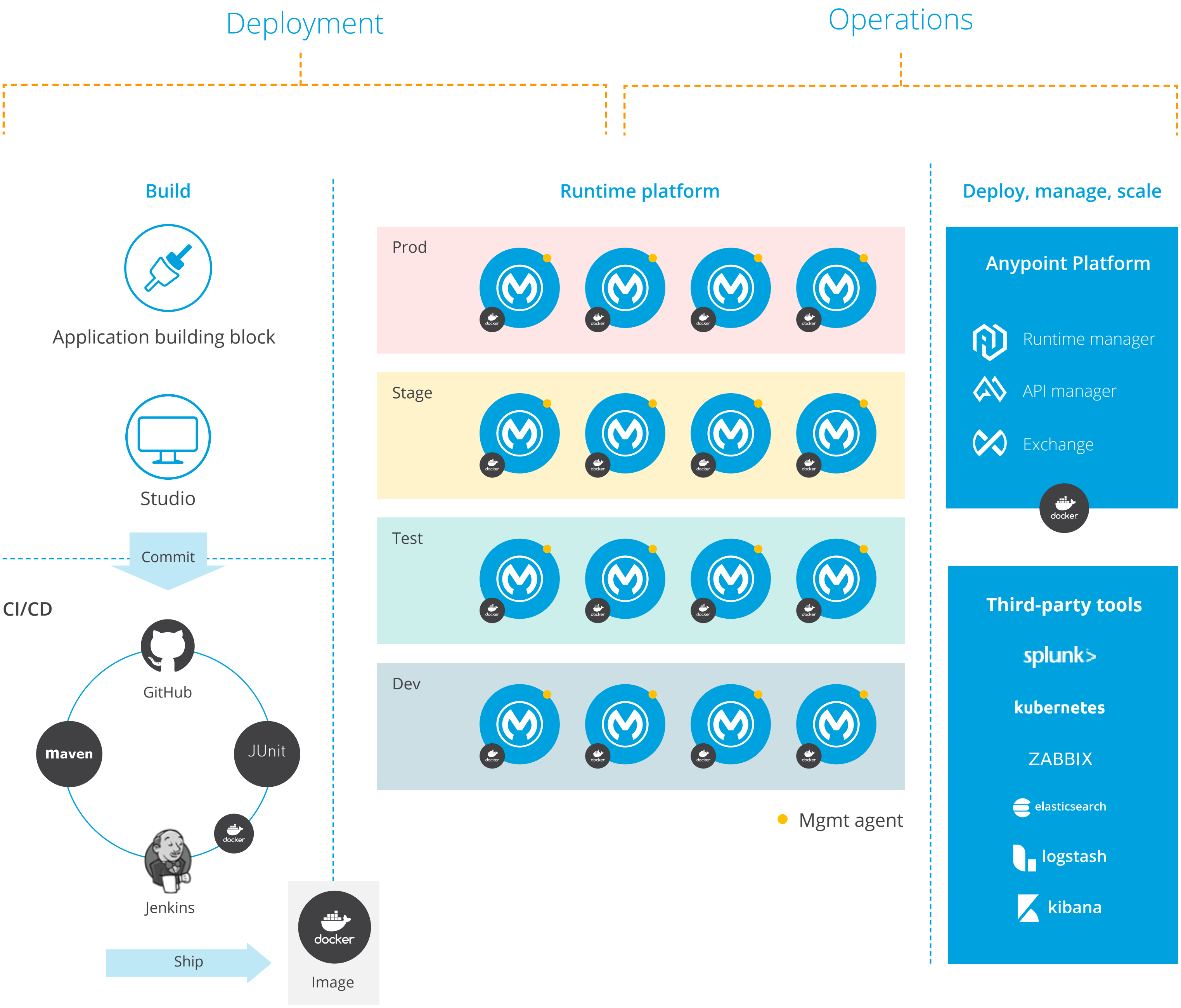


Figure 3: Operating Anypoint Platform

Finally, for monitoring and analytics, developers and Platform operators have the option to either use the out of the box capabilities provided by Anypoint Platform or plug into any existing third-party tools: monitoring and analytics tools such as Splunk, Nagios, AppDynamics, New Relic, or the open-source Elastic Stack: Elasticsearch, Logstash, and Kibana. All this is made possible through the extensibility of the Mule runtime engine.

Shifting from production to production + consumption

The more discoverable you make the assets that you create through your DevOps pipeline, the faster you will become at making them ready for reuse, enabling line of business IT, partner developers, and internal developers to securely self-serve on top of these assets. This will make your organization more innovative in creating new products and services because you're exposing capabilities to audiences that may or may not have otherwise known about them, or have had access to these pieces of data or connectivity artifacts. Reusability enables speed, innovation, and agility.

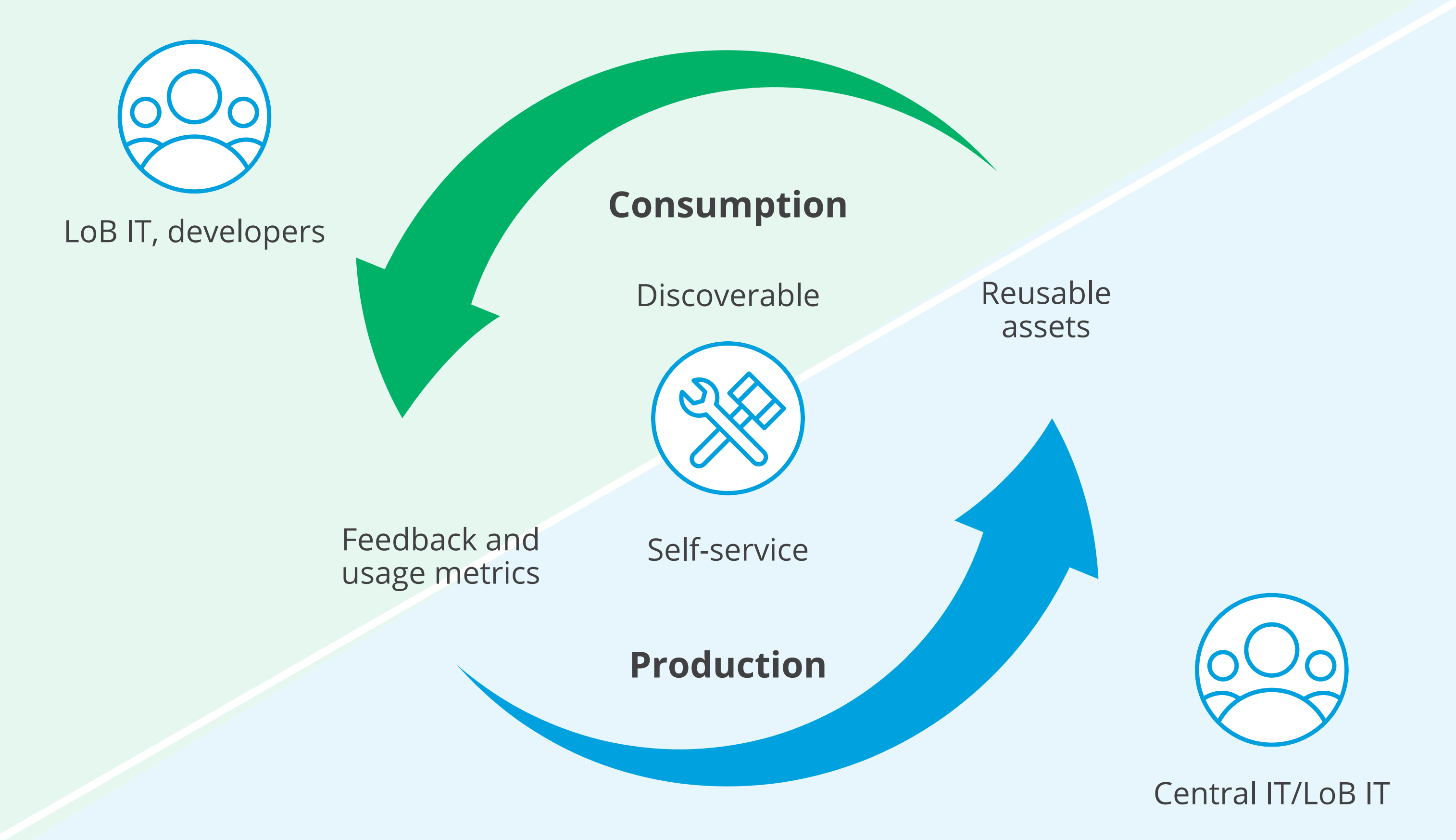


Figure 4: A new operating model: Focusing on production + consumption

DevOps increases reliability and minimizes disruption, with the pleasant side effect of increasing speed; what it doesn't do is provide control. An emerging construct for making sure that organizations aren't needlessly replicating effort, aren't creating applications and assets in business silos, and are effectively exposing the assets they create for maximum business value, is an approach to integration called "API-led connectivity." In our experience, this approach provides the answer to producing reusable assets and enabling their consumption to drive useful speed and innovation.

API-led connectivity provides an approach for connecting and exposing assets. With API-led connectivity, every asset becomes a managed API, a modern API, which makes it discoverable through self-service without losing control.

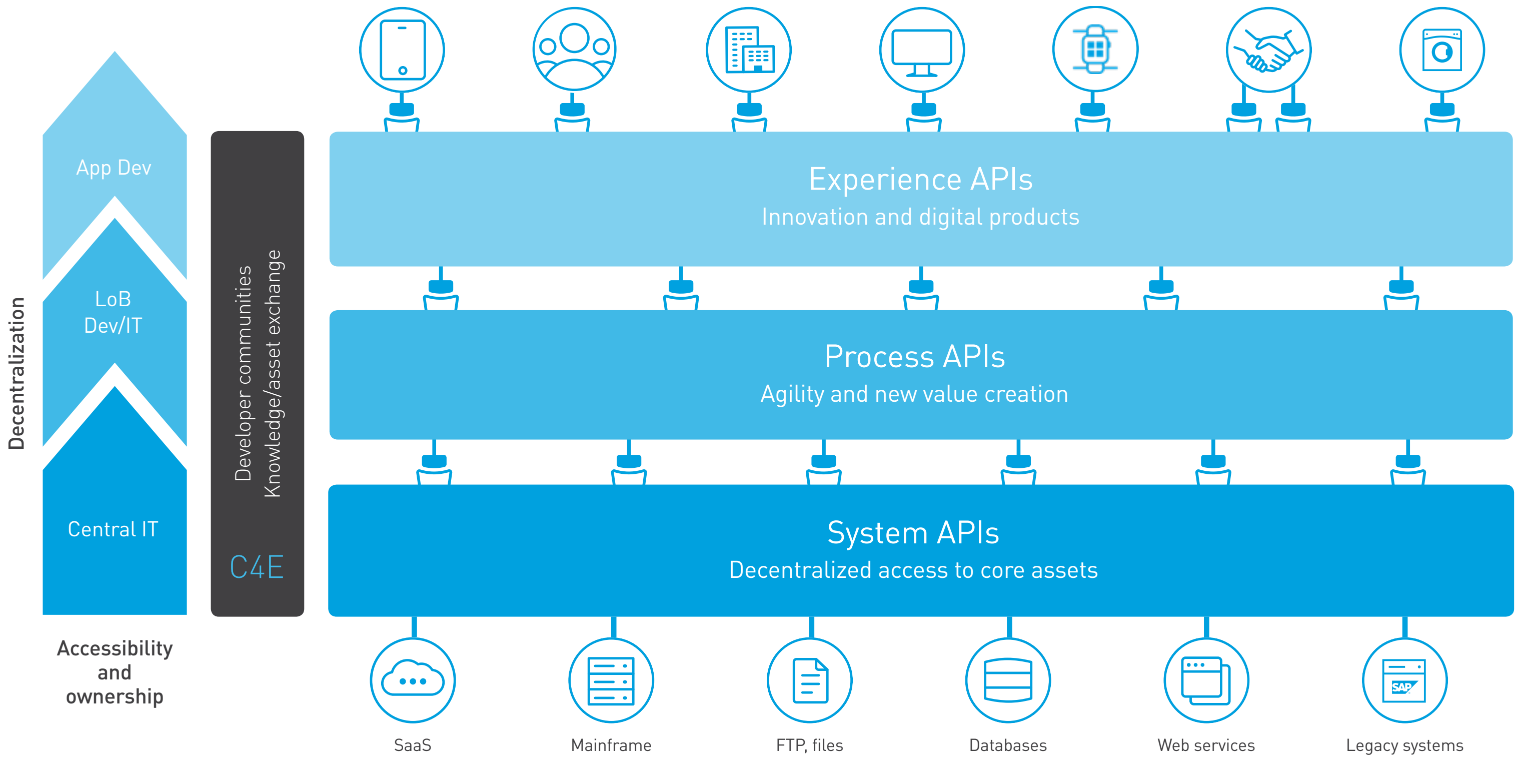


Figure 5: A new approach to API-led connectivity

Systems APIs: These unlock data from your core systems by putting APIs in front of them. Then a "System API" layer is formed, which provides consistent, managed, and secure access to backend systems.

Process APIs: It is possible to build on the System APIs by combining and streamlining applications with a certain amount of business logic, and then mapping them to create new capabilities. These "Process APIs" take core assets and combine them with some business logic to create a higher level of value. Importantly, these higher-level objects are now useful assets that can be further reused, so they should also be APIs themselves.

Experience APIs: Finally, an API is built that provides an end-user experience combining the capabilities exposed by the other two API layers. These are "Experience APIs" that are designed specifically for consumption by a specific end-user app or device. These APIs allow app developers to quickly innovate on projects by consuming the underlying assets without having to know how the data got there. In fact, if anything changes to any of the systems or processes underneath, it shouldn't require any changes to the app itself.

Center for Enablement (C4E): A cross functional team across Central IT, LoB IT charged with being the stewards of assets created through the DevOps pipeline. This team is goaled to ensure assets are consumable, consumed broadly, and fully leveraged. Once established, this team ensures teams follow DevOps best practices, productizes and publishes assets, encourages collaboration and drives self-reliance.

Now, with API-led connectivity and a C4E, when you are tasked with building a new app or asset, there are reusable assets to build from, eliminating a great deal of the work needed to build them. Even better, it is now much easier to innovate, there is now greater visibility into what has been already created, and it easy to expose capabilities to partners, third parties, and other consumers; as we shall see below, this function is critical to even the most expert DevOps practitioners.

The benefits of the MuleSoft approach to DevOps

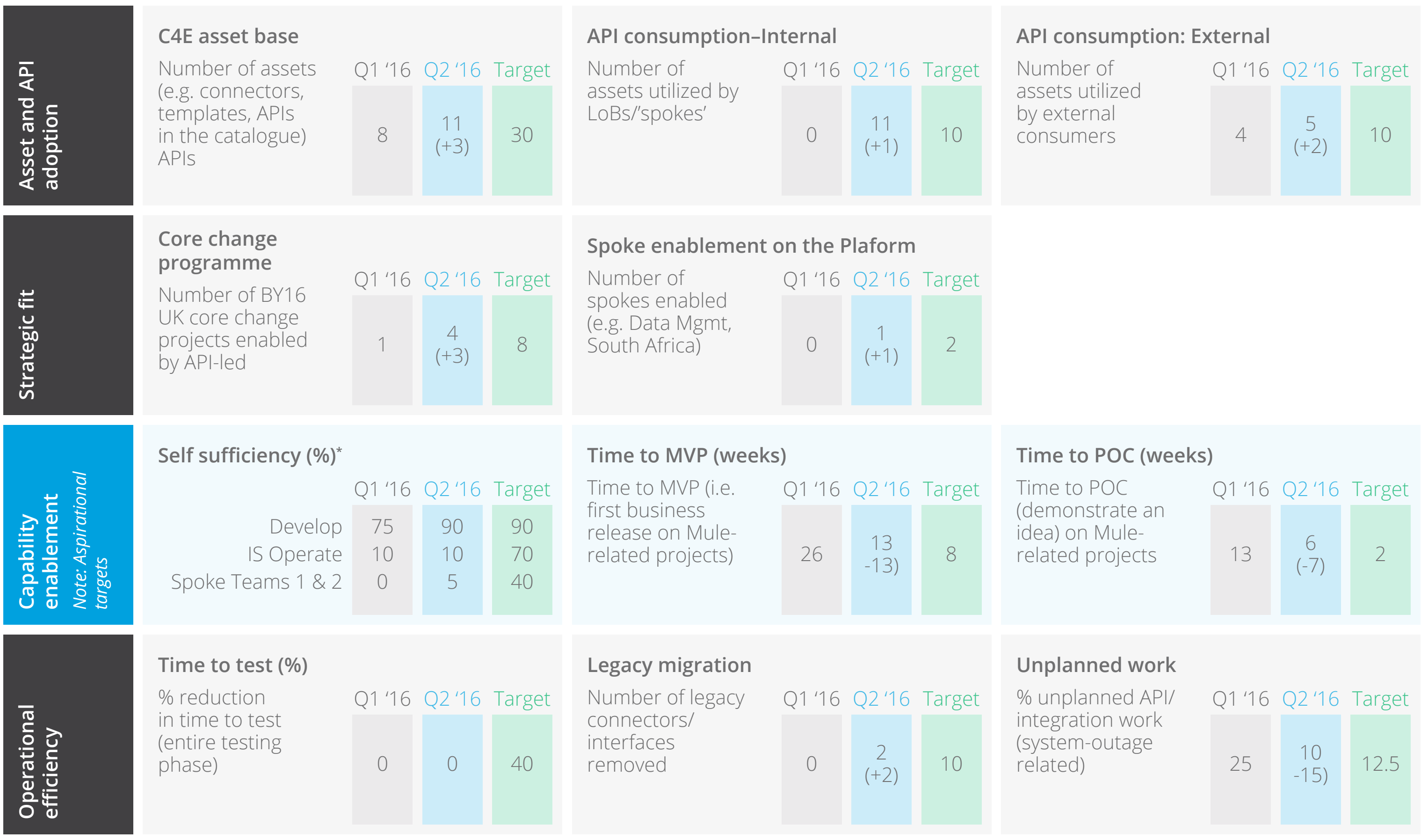
A customer that's really taken advantage of this API-led approach to connectivity is Spotify. Spotify is already a leader in DevOps; production and deployment was not a problem. It was so easy to deploy and produce assets, that there would be duplicates of applications created by multiple departments, and those applications were quite difficult to take advantage of outside their particular business silo.

What they needed was to expose the capabilities they were moving through that pipeline out to third parties, partners, and across internal channels so that they could more effectively drive new initiatives and create new opportunities for the business. Spotify is now leaning on MuleSoft to provide those reusable APIs and microservices that take advantage of their speedy time-to-market by pushing things through the existing DevOps pipeline. They are promoting self-service to enterprise data within teams, creating faster time to market for numerous business functions. With MuleSoft, Spotify can securely expose assets created through their existing DevOps practice out to systems such as data warehouses with company ERP infrastructure layers. This enhances functions like forecasting and reporting through improved information sharing across the enterprise.

But not every organization has the fortune to be in a position like Spotify who, from their founding had a DevOps practice and developer culture with automation at the core. Another example is a MuleSoft customer, Siemens, who, not that long ago had a software delivery model that will be familiar to many organizations. They had third party resources responsible for developing and testing code; that code was then handed off to the operations team to make sure it ran appropriately in various servers. On the side, a data management silo that slowly feed information to these development teams as it was requested.

Their first iteration of change was bringing in an ESB partner in MuleSoft and a BPM solution to help streamline some of the processes by which information was accessed. They started putting services on a bus and encouraging development teams to reuse those services and processes. Outside of Central IT, few of these services were utilized and the data management team often became a bottleneck to new services. As they started to establish their DevOps practice and followed an API-led approach, they quickly began seeing the value in having teams serve as champions for their own services and processes. Treating DevOps as a practice across people, processes, and technology, Siemens created a center for enablement.

This DevOps Center for Enablement plays a critical role in helping teams adopt best practices in developing and deploying code in highly reusable ways. With this approach, Siemens has removed data bottlenecks by securely exposing artifacts managed by the data management team via System APIs. Partners and internal teams can now self-serve and deliver innovative solutions.



	Metric	Description	Q1 '16	Q2 '16	Target
Asset and API adoption	C4E asset base	Number of assets (e.g. connectors, templates, APIs in the catalogue) APIs	8	11 (+3)	30
	API consumption–Internal	Number of assets utilized by LoBs/'spokes'	0	11 (+1)	10
	API consumption: External	Number of assets utilized by external consumers	4	5 (+2)	10
Strategic fit	Core change programme	Number of BY16 UK core change projects enabled by API-led	1	4 (+3)	8
	Spoke enablement on the Plaform	Number of spokes enabled (e.g. Data Mgmt, South Africa)	0	1 (+1)	2
Capability enablement *Note: Aspirational targets*	Self sufficiency (%)*	Develop	75	90	90
		IS Operate	10	10	70
		Spoke Teams 1 & 2	0	5	40
	Time to MVP (weeks)	Time to MVP (i.e. first business release on Mule-related projects)	26	13 -13)	8
	Time to POC (weeks)	Time to POC (demonstrate an idea) on Mule-related projects	13	6 (-7)	2
Operational efficiency	Time to test (%)	% reduction in time to test (entire testing phase)	0	0	40
	Legacy migration	Number of legacy connectors/ interfaces removed	0	2 (+2)	10
	Unplanned work	% unplanned API/ integration work (system-outage related)	25	10 -15)	12.5

* A subjective measure of knowledge/capacity

Figure 6: Figure 6: Siemens DevOps Center for Enablement ROI

The results speak for themselves. Siemens' time to MVP — getting a first business-release of the capability out to the business on related projects — has decreased from 26 weeks to 13 weeks in just a quarter as a result of this capability enablement. They slashed that time in half in just over three months. Similarly, time to POC — demonstrating an idea to a business unit — has been slashed in half, from 13 weeks to six weeks. All the while, unplanned work has dropped below expectations, because teams are able to reuse some of the artifacts that they created with through an API-led connectivity approach.

Siemens is not reinventing the wheel every time a new project comes up, improving productivity and speed. DevOps plays a key role in that, and an API-led connectivity approach and Center for Enablement promote reuse of the artifacts that come through the DevOps pipeline.

Conclusion

DevOps can help make companies increase reliability and minimize disruption, helping them move faster and adjust to change with greater ease. But it's an approach that needs to be undertaken with a holistic understanding of how technology, processes, and teams work together. MuleSoft has helped numerous organizations implement DevOps successfully and unlocked new capabilities with its API-led approach to connectivity.

Contact us to find out the best way to get started.

About MuleSoft

MuleSoft's mission is to help organizations change and innovate faster by making it easy to connect the world's applications, data and devices. With its API-led approach to connectivity, MuleSoft's market-leading Anypoint Platform™ is enabling over 1,400 organizations in approximately 60 countries to build application networks.

For more information, visit **mulesoft.com**