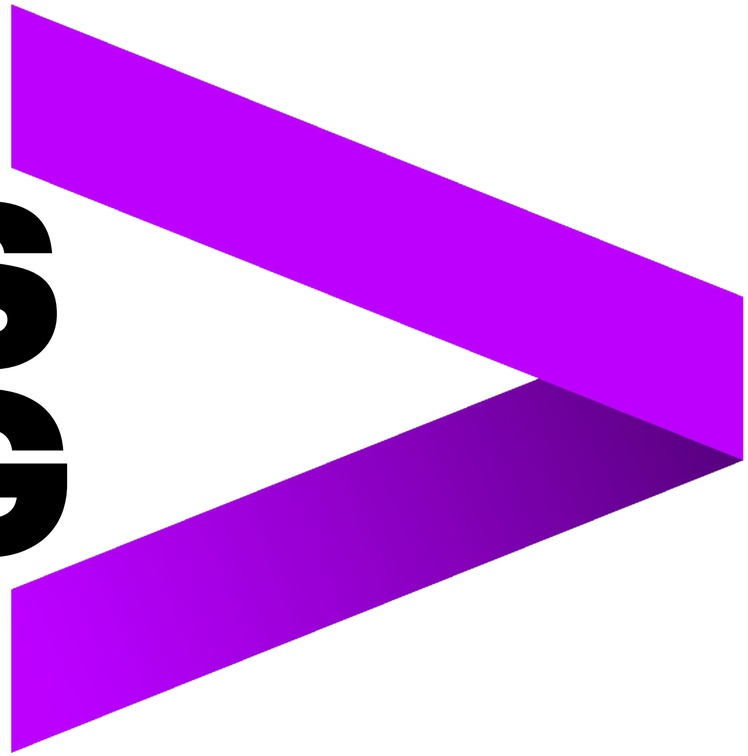
accenture

# SERVERLESS COMPUTING

## The next step in the evolution of cloud



# CLOUD COMPUTING HAS BROUGHT ENORMOUS CHANGE TO THE WORLD OF APPLICATIONS

Cloud makes longstanding constraints on application development and deployment pretty much disappear. It's no exaggeration that much of the innovation in IT over the past decade has been enabled, catalyzed or caused by cloud computing.

## BUT WHAT DOES THE NEXT PHASE OF CLOUD COMPUTING LOOK LIKE?

# ONE PIECE OF THE PICTURE IS SERVERLESS COMPUTING

## What is serverless computing?

Serverless computing is the idea of running an application using a combination of cloud services to build its features, rather than launching virtual servers configured with those features. This includes services like database, queueing, messaging, object storage, and function as a service. Servers are still involved, of course—just not *your* servers. For the developer and operator, how the functions are executed—that is, which infrastructure it uses—is all hidden and, in fact, irrelevant.

This development gives companies a powerful new way to help potentially reduce IT operational costs, deploy services faster, and be more responsive to customer demand. In a serverless environment, a company replaces servers dedicated to specific applications by outsourcing (and abstracting) infrastructure functions to a public cloud vendor such as **Amazon.com, Inc.**, **Microsoft Corporation** or **Google Inc**.

When an application isn't being used, it sits latent, with no computing power running behind it. But when someone makes a request of the application, the infrastructure behind it marshals all the technology components necessary to process the application's business logic and respond to the request.

So serverless computing still has the virtualized, containerized services underneath it, but people don't interact with those servers anymore. All those infrastructure tasks, like provisioning and scaling and cleaning up, are done by machines in a completely automated lifecycle.

# DOES SERVERLESS COMPUTING EQUAL FUNCTIONS-AS-A-SERVICE (FAAS)?
# IN A WORD, NO.

**Serverless is often used interchangeably with FaaS (Functions-as-a-Service). This is only partially correct.**

Serverless provides FaaS as one of its components, but serverless also provides Data-as-a-Service (DaaS) persistence, management of API endpoints, name resolution, data streaming, access and identity subsystem, and monitoring services. Serverless includes FaaS, but FaaS alone is not sufficient to run serverless solutions.

# WHAT'S SPECIAL ABOUT SERVERLESS IS THAT YOU ONLY USE WHAT YOU NEED, WHEN YOU NEED IT

**Computing power on demand creates an efficient solution design that often costs less and performs better.**

At Accenture, for example, we use a scalable, serverless architecture to manage thousands of public cloud accounts for internal consumption and on behalf of our clients. These accounts have millions of resources that need to be tracked for configuration, security, policy, costs and overall governance.

## Three characteristics are at the core of delivering benefits from serverless computing:

**1. No servers to manage.**
No operating system to install or patch.

**2. Flexible scaling.**
Scale is managed for you or is done in a way that's defined in terms of the actual capacity of the application as opposed to having to consider things like CPUs and memory and other kinds of server-based concepts.

**3. Automated, high availability.**
All the serverless components of the overall platform have built-in high availability. That's not something you need to design or really think about in your applications. Companies get that high availability out of the box.

# SERVERLESS AND EVENT-BASED COMPUTING

**One of the other key benefits of an event-driven approach, in the context of a serverless architecture, is the idea of helping eliminate waste from an actual infrastructure cost perspective.**

The Event-Driven Architecture (EDA) of serverless computing means that every component is independent and decoupled. So, speed to deployment in serverless is much faster than more tightly coupled systems, and costs to deploy are also reduced. The ability to “listen” to events and react to them once they happen in an elastic, scalable manner is a key advantage of serverless.

One of the other key benefits of an event-driven approach in the context of a serverless architecture is the idea of helping eliminate waste from an actual infrastructure cost perspective. If you think about most of the servers that are out there in the world and in any data center, most of them are spending a lot of their time sitting idle, listening on a port waiting for a request to come in or waiting for an event to happen. Whether your server is active or it’s just sitting there idle, you’re still having to pay for it in one way or another, whether you own the physical hardware yourself or if you’re paying for a virtualized instance in some sort of cloud platform.

With serverless, that whole model is turned on its head to where now you are only paying for the compute time that you’re using. So if your application goes through periods of low utilization and periods of high utilization, as a developer you only need to worry about how to handle each individual event. Then your infrastructure, your actual costs to run that application, only accrue when the code itself is running in response to those events.

# WHY IS THERE SO MUCH HYPE ABOUT SERVERLESS COMPUTING?

## Here are some of the most significant benefits:

### Reduce costs

When you eliminate always-on systems that don't have to be always on, you have an application sitting on a hard drive in the cloud until someone uses it. Because you're only paying for what you use, the drop-in infrastructure costs can be significant—ranging from 50 percent to 90 percent, with 70 percent to 80 percent being about average.

### Auto-scale on demand

Building scale with traditional architectures is very expensive. It takes a lot of people and resources to build an infrastructure that even approaches auto-scaling to meet demand. In addition, you often still need considerable manual intervention—which can spell trouble during a surge. With serverless, applications automatically have access to the right amount of infrastructure support to respond to all the requests they receive.

### Decrease time to market

It generally takes far less time to develop serverless applications. For instance, when Accenture puts together tailored serverless applications, we can generally do so in a matter of weeks, not months. Why the difference? With serverless, we're working with a set of well-known, already-established infrastructure and runtime components. Developers can focus on their main competency, which is designing the core business logic. The underlying software and infrastructure are completely hidden from the developer, thereby reducing the overall time to develop and deploy the software. We don't have to worry about setting up servers, configuring environments, clustering or other facets of building a traditional system.

# WHY IS THERE SO MUCH HYPE ABOUT SERVERLESS COMPUTING?

### Create more transparent, business-event-driven execution

The application is code that is executed only in reaction to a well-defined business event, thereby increasing overall transparency and reliability of the application to handle very specific business events or anomalies.

### Reduce security risks

Prior to the deployment of serverless-computing-based applications, most security organizations focused on perimeter security and access control. But in a serverless environment, security shifts to the orchestration of application code, enabling a complete monitoring of activities, connections and user behaviors. With availability of robust application monitoring and alerting technology, security risks can be potentially reduced. Simple monitoring and alerting can be used as a baseline to identify and stop an anomalous behavior, thereby potentially helping reduce the overall risk posture of an enterprise.

### Minimize service governance

Adoption of cloud generally requires changes to processes such as provisioning of individual cloud services, billing and metering of the services, and usage monitoring. A serverless architecture lessens the burden due to the automation of the provisioning and governance of the infrastructure resources required to execute a business function.

# ACCENTURE'S OWN USE OF SERVERLESS ARCHITECTURE: ACCENTURE CLOUD PLATFORM

**Accenture Cloud Platform (ACP) plays an important role in helping our clients get the most from their cloud journey.**

ACP provides visibility and control enabled by using its serverless architecture to discover resources in customers' accounts. The goal is to make it easy to manage our clients' cloud estate with governance tools, advanced cost analytics capabilities and dashboards.

This last point is especially critical for large companies with numerous accounts comprising many resources, and with infrastructure and services spread around the globe. ACP effectively allows massive enterprises to have visibility across all their environments. But importantly, it's not a "single-pane-of-glass" visibility but a co-existence model.

Accenture does not try to provide a portal that wraps every option a user may need (a goal that cannot keep up with the rate of change of public clouds). Instead, ACP does real-time discovery of cloud resources, allowing control and governance no matter how they are created. This helps us in areas like compliance, where violations can be found and remediated quickly.

# THE BUSINESS FACTORS THAT DROVE ACCENTURE CLOUD PLATFORM TO SERVERLESS

**In Accenture's journey to serverless, we wanted to solve a set of business challenges that are common to any product group in a large enterprise faced with a fast-growing market disruption like cloud.**

These challenges and their solutions are instructive to other companies contemplating the move to serverless.

**1. Speed**

Given how quickly cloud services are evolving, we needed to add support and management to these services at speed. With companies like Amazon Web Services, Inc. (AWS), adding as many as three major features a day, it's essential to keep up.

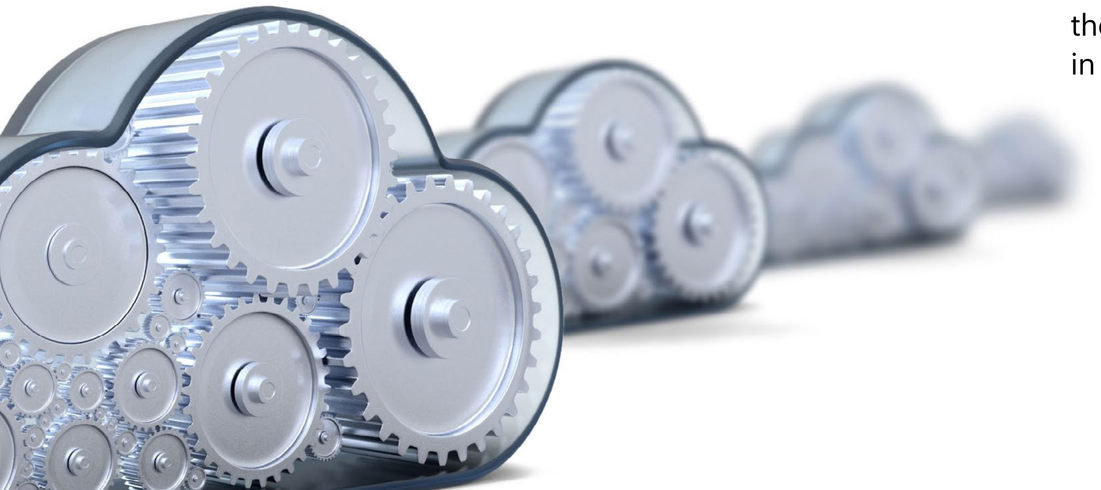
**2. Budget**

Although cloud services grow at very high rates, engineering budgets do not. Companies need a way to disconnect cloud growth from cost growth. From our perspective, the best way to do that was to create a true platform that allowed other members of our ecosystem to serve themselves and others. Think of it as Tom Sawyer getting other kids to paint the fence, but in a win-win manner.

**3. Commercial alignment to market needs**

It's no secret that more and more customers want everything as a service and to pay on a unit-of-consumption basis. The challenge is that the number of "units" of measure is growing rapidly. Serverless allows Accenture to dial the unit of consumption all the way down to a specific customer function, like "update configuration management database" or "put a server to sleep." And it's done in a more cost-effective way.

# ALREADY DECIDED TO GO SERVERLESS?



**If you're ready to start the serverless journey, here are six especially important things to consider:**

1. **Don't try to reuse existing application code in a serverless environment.**
   You may have a specialized application you spent a lot of money building that you'd like to repurpose. But if you do that, you'll end up having way more code and heft than what the serverless environment was designed for.

2. **Use existing platform components from providers for non-business-logic application functionality.**
   Serverless architecture already has all the execution components you need, and they're designed to work within the serverless paradigm. For anything that's not related to business logic, the platform component should always be your first choice.

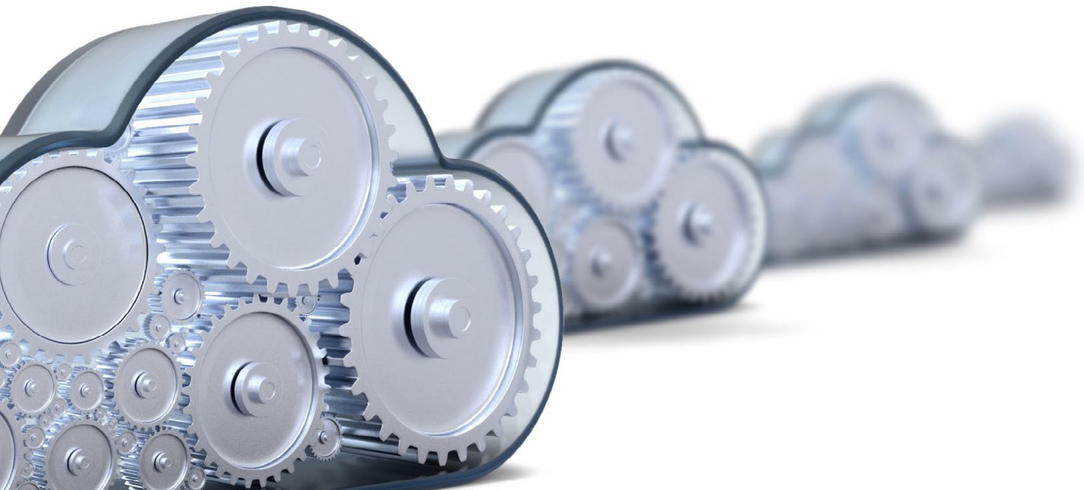
3. **Create a reference architecture to guide all application development.**
   By creating a blueprint for what systems should look like, you ensure consistency across all your applications. You can move people across different projects, and they'll still understand the components; while minimizing support issues inherent in "rogue" code.

# ALREADY DECIDED TO GO SERVERLESS?

**4. Support DevOps.**
Make sure your application supports a DevOps-enabled systems development life cycle process before you begin development.

**5. Recognize that the skills required for success in serverless are different from those of the average developer.**
Serverless technology is still in its early days. So, when adopting it, you'll likely find that you may either need to retrain some of your existing people or bring in new talent who is well-versed in serverless development.

**6. Be prepared for a massive cultural shift.**
In our experience, you may need to be ready to do as much work restructuring organization and culture as you devote to building of serverless applications. The massive horizontal IT bureaucracy that's common in most companies is antithetical to radically simple applications that are the hallmark of serverless.



# SERVERLESS COMPUTING SUCCESS STORIES

**Companies and government agencies are using serverless computing to support a wide variety of opportunities and needs.**

### DECREASED TIME TO MARKET

Consider the marketing technology company Plexure, which uses serverless computing to help customers customize its platform with features unique to the customer's needs. Plexure uses microservices architecture and continuous delivery, but serverless means that customers can tailor their experience without waiting for Plexure to deploy code specific to them.[1]

### AUTO-SCALE ON DEMAND

Another example is Alameda County, CA, which needed to update its election reporting viewer for the 2016 election. Its on-premise system had crashed in 2014, so they needed a different approach. They deployed a serverless solution which performed without error on election night, even with 3,000 concurrent users. And the price? $25. The county feels that, taking a traditional route, it would have spent thousands of dollars for a solution that might not have worked, and then they would have kept paying for that infrastructure even if they didn't need it.[2]

**1** https://www.plexure.com/plexure-blog/2017/6/19/going-serverless-to-empower-customers
**2** https://aws.amazon.com/solutions/case-studies/alameda-county/

# CONCLUSION: THE SERVERLESS FLASHPOINT

Looking across the portfolio of Accenture's Global 2000 clients, the number of them that are already experimenting internally with serverless is striking. They're already generating minimum viable products to try to understand and take advantage of this new computing paradigm. Accenture believes that there will be a flashpoint soon where all of those small pilots turn into full production-scale systems. Will you be ready? It's all going to happen very quickly. Plan now to stay ahead of the curve.

## About Accenture

Accenture is a leading global professional services company, providing a broad range of services and solutions in strategy, consulting, digital, technology and operations. Combining unmatched experience and specialized skills across more than 40 industries and all business functions—underpinned by the world's largest delivery network—Accenture works at the intersection of business and technology to help clients improve their performance and create sustainable value for their stakeholders. With more than 442,000 people serving clients in more than 120 countries, Accenture drives innovation to improve the way the world works and lives. Visit us at www.accenture.com.

## Authors

**Miha Kralj** Managing Director, Cloud Architecture
miha.kralj@accenture.com

**Matthew Lancaster** Associate Director, Tech Architecture Science
matthew.d.lancaster@accenture.com