merapar

Whitepaper

Serverless First

More than just a technology strategy

Lead. Don't follow

Whitepaper by
Bart van Raalte
Chris Birkinshaw
Klaus Seiler

Serverless-first: More than just a technology strategy

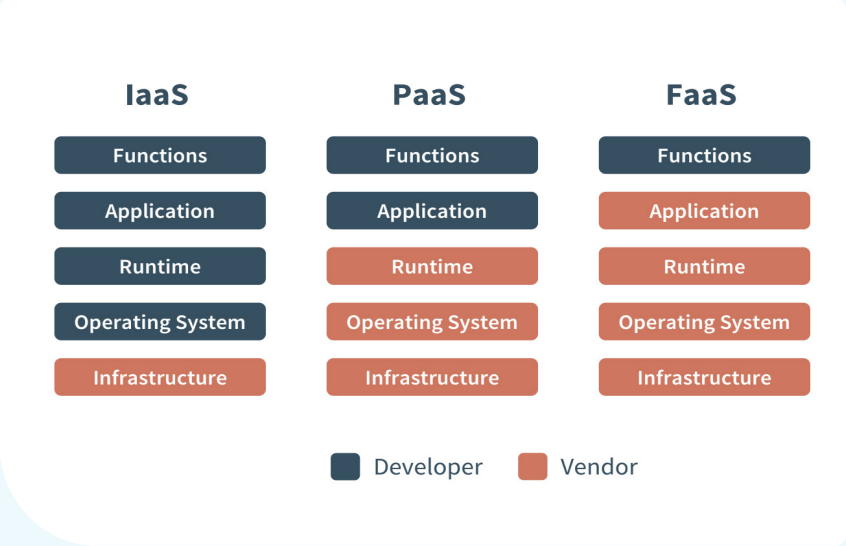
In competitive markets, companies must not only move to serverless systems, but also prepare for organizational transformation - or risk being left behind.

In today's rapidly changing world, companies in many industries need to innovate, scale, launch products, and adapt to new circumstances fast. This is a particularly pressing requirement for industries with short product release cycles, where companies must regularly deliver new features while maintaining their existing systems to stay competitive.

Serverless-first strategies provide an exciting opportunity to address these modern market demands. While not all companies are ready to adopt serverless technologies, doing so can provide significant benefits to a wide range of businesses - as long as the organization can adapt accordingly. Companies that combine serverless-first strategies with radical agile transformation are most likely to achieve success.

This whitepaper will explore the advantages of serverless systems, how and why to organize for success, the processes and skill sets needed, and real-world examples of how going serverless-first can unlock more agile ways of working with faster results.

What can serverless systems do for businesses?



Different cloud vendors define serverless computing in various ways. Generally, though, serverless architecture means that a cloud provider supplies the user with a backend as a service where they can run discrete functions of code, removing the need for developers to manage the underlying infrastructure directly. The most common serverless hosting model is 'Function as a Service' (FaaS). Serverless systems have several key benefits - including:

- **Quicker idea-to-development.** Because businesses using serverless systems are only responsible for application code, developers can add and modify individual pieces of code as needed without having to manage a server's supporting infrastructure. This allows companies to deploy updates fast, without the labor requirements of non-serverless approaches - enabling quick adaptation to market changes.

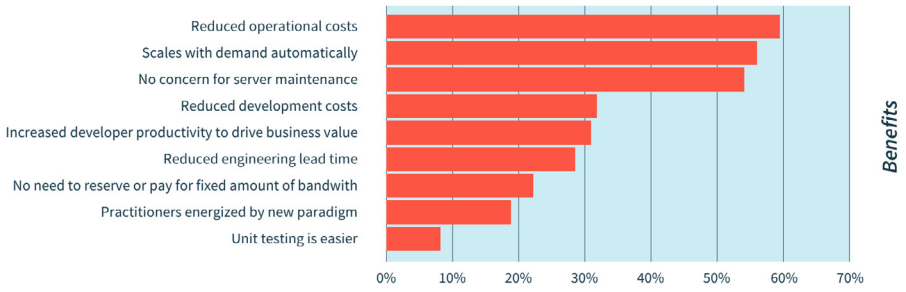
- **Lower operational and infrastructure costs.** Serverless systems typically use a pay-as-you-go pricing structure: the cloud provider charges for only the resources needed to execute a particular function of code. In this way, serverless models allow operational and infrastructure costs to track the business' growth and usage.

"The future of cloud is more conversational programming and industrialisation further up the stack. Today is all about serverless and FinOps with concepts like capital flow. Yesterday was all about IaaS and DevOps." - Simon Wardley, Inventor of Wardley Mapping and Researcher for the Leading Edge Forum.[1]

- **Easier scaling.** In a serverless system, the cloud provider optimizes resource utilization, continuously adapting the allocated compute capacity to changing demand. Compared with traditional cloud computing, where users have to manage more of the stack themselves, which requires more people, serverless models make scaling easier and more cost-effective.

- **The ability to define a clearer focus.** Serverless systems allow users to focus on one task and keep iterating to perfect it. Specifically, because the cloud provider leverages its scale, security, and functionality to manage and perfect the infrastructure logic, developers are free to fine-tune the application logic.
- **Reduced environmental impact.** Because serverless systems run on-demand for only the time they are used - in contrast to provisioned-capacity virtual machines (VMs) that are often under-utilized - they can save significant amounts of energy. In turn, this enables reduced CO_2 emissions. The higher hosting density on serverless systems, with usage shared among a wider set of users and servers repurposed for different uses at different times, also drives energy efficiency.

What are the biggest benefits your organisaztion has gained after adopting serverless?



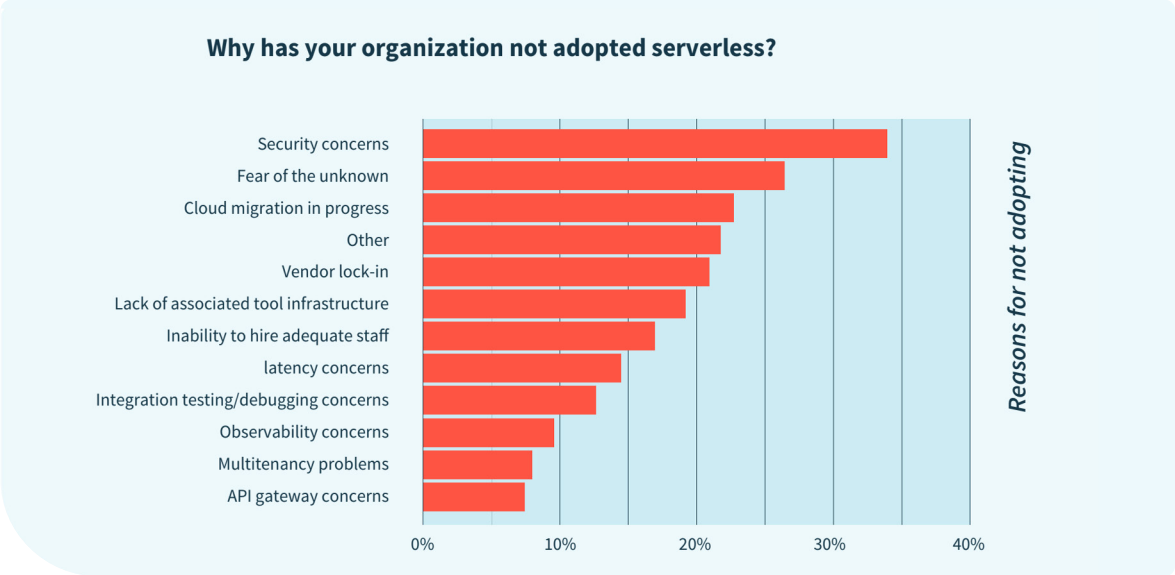
"O'Reilly serverless survey 2019: Concerns, what works, and what to expect"

merapar

[1] https://twitter.com/swardley/status/1436280981087047682

What is a serverless-first strategy - and is it right for everyone?

Despite these benefits, not all businesses are ready to adopt serverless technologies. Many still see the transition as a 'leap of faith', with common concerns including cloud provider security, vendor lock-in, existing sunk costs in other strategies, and difficulty with processes such as debugging and development environments. Risk-averse industries, or those with longer product lifecycles, are particularly likely to favor other models.



"O'Reilly serverless survey 2019: Concerns, what works, and what to expect"

Even in companies that are hesitant to adopt a serverless-first strategy, serverless systems are already bringing benefits to select parts of the business. For instance, while many banks maintain traditional network infrastructures in their underlying databases, they are increasingly implementing more agile approaches for their consumer-facing apps and technologies, in response to growing demand.

"We have reinvented our operating model, moving to 100% agile for delivery of software, leveraging micro services and building on open-source software. By moving to AWS and re-architecting our entire data set, we have dramatically advanced our ability to manage data at a larger scale, which created the foundation to leverage machine learning." - Chris Nims, SVP for Technology at Capital One.[2]

At Merapar, though, we define serverless-first as using serverless development strategies by default, unless there is a clear technical reason making this impossible. We believe this strategy will normally bring significant value to any organization, regardless of industry - as long as the company is willing and able to adapt its approach.

A strategy that unlocks increased agility

One of the major benefits of a serverless-first strategy is the increased agility - in practices and structures - that it can unlock. Indeed, agile and serverless approaches go hand-in-hand: both enable and encourage quick development, modification, and adaptation.

"Modern applications are built server-less first, a strategy that prioritizes the adoption of serverless services, so you can increase agility throughout your application stack. The operational management required to run and scale applications is handled for you, so you can go to market faster and innovate more." - Nicolas Moutschen, Serverless Specialist Solutions Architect at AWS.

A key feature of agile organizations is more cross-functional and autonomous teams that reduce dependencies and remove barriers to change within the organization. For instance, since the cloud provider is responsible for almost the entire stack, serverless-first companies can merge certain roles, such as developer and infrastructure (DevOps) engineer. This approach is seen in Amazon's "You build it, you run it" philosophy and Netflix's "NoOps" strategy.

"Every developer is responsible for running what they wrote, and the team members take turns to be on call… There is no ops organization involved in running our cloud, no need for the developers to interact with ops people to get things done, and less time spent actually doing ops tasks than developers would spend explaining what needed to be done to someone else." - Adrian Cockroft, former Cloud Architect at Netflix.[3]

[2] https://www.forbes.com/sites/tomgroenfeldt/2021/07/21/capital-one-adding-3000-tech-staff-to-build-on-cloud-banking-leadership/?sh=e017549418a8
[3] http://perfcap.blogspot.com/2012/03/ops-devops-and-noops-at-netflix.html

These changes can improve both speed and quality. They allow for simpler branching and merging strategies, since a single developer can write new code to be deployed immediately once approved. This process can be completed within a matter of hours or days, allowing for several new features to be released within one typical two-week iteration without depending on other teams for deployment. This more agile way of working is ideal for quick time-to-market and continuous or experimental development.

Radical transformation required - "From DevOps to NoOps!"

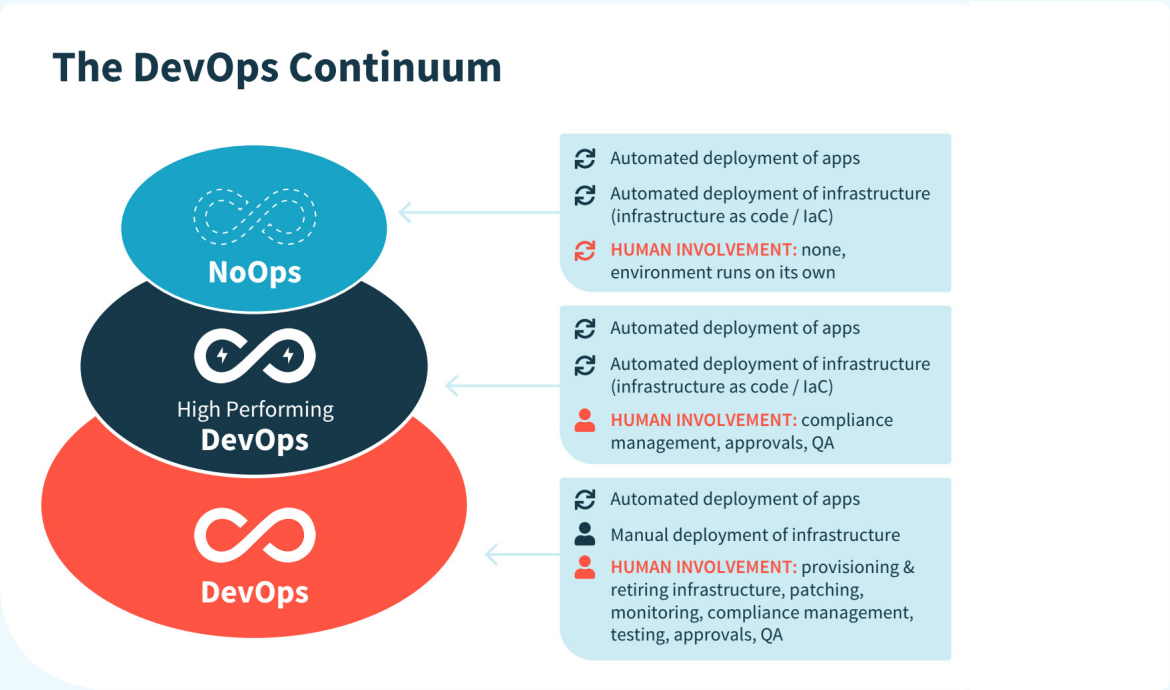
Restructuring for increased productivity often means that, successfully executing a serverless-first strategy requires extensive changes to the organisation and its software architectures. For instance, in a traditional hiring process, developers are expected to be proficient with common languages like Java and Javascript, but not necessarily to have any direct experience of provisioning cloud functions, tuning CPU and memory requirements, setting up monitoring and alerting, and providing operational support. In many cases, candidates expect this to be done by a separate Infrastructure (or 'DevOps') team.

By contrast, in a serverless world, the design, deployment, and operation of infrastructure are the responsibility and remit of the serverless team. This means that traditional DevOps structures will need to be changed. Specifically, siloed DevOps roles are no longer required. These functions and tasks are absorbed by the serverless development team, resulting in a shift from the traditional 'DevOps' to 'NoOps' - or maybe to the controversial 'IntegratedOps'.

In turn, this means that, when hiring for a serverless strategy, at least the lead or senior developers must have hands-on experience developing, deploying, and supporting serverless architectures. And, within a company's existing teams, some colleagues may also need to be retrained for new roles, or existing job descriptions may need updating.

> *"Engineers now need to understand everything, but interface through different abstraction layers. Serverless architectures allow engineers to work effectively in this fashion. It's possible for a small team to do everything (honestly, it is). For this to happen, the engineers need to be comfortable with some attributes that the software engineers of yesteryear may not have been." - David Anderson, Director of Technology, Liberty Information Technology.*[4]

[4] https://thenewstack.io/the-evolution-of-the-serverless-first-engineer/



Organizations will also usually need to re-architect solutions when transitioning to a serverless approach to optimize performance and cost. For example, switching to asynchronous event-based architectures pairs well with the constraints of serverless functions (startup latency and execution time limits), while maximizing the benefits of on-demand billing. Organizations may also need to transition to more operational-expenditure-driven cost structures: since they are responsible for less code, they cannot capitalize on capital expenditure as much as when using a non-serverless approach.

To go serverless, you need to let go of cloud-agnostic

For organizations with concerns about vendor lock-in, a cloud-agnostic strategy can look like the best option. But the advantages of these strategies are often outweighed by their drawbacks, asthey require a high investment that often fails to pay off. Writing software for multiple clouds removes the ability to use any features specific to one cloud, eliminating any competitive advantages of using a specific vendor.

Unless a cloud-agnostic user actually moves providers before the software is refreshed, they are unlikely to get their initial investment back; rebuilding the software every few years (if necessary) is likely to be cheaper in the long term. By contrast, serverless users can get more value by using more services from their cloud provider - services that will already be compatible, integrated, tried and tested in different configurations. It is very difficult and restrictive to make a serverless architecture cloud-agnostic. But, if serverless users do need to switch cloud providers, the smaller amount of code that the user is responsible for, means there will be less to rewrite.

How Merapar supports customers to go serverless-first

At Merapar, we support customers to adopt serverless-first strategies - and to take maximum advantage of them - using our expertise in the associated technologies. To demonstrate the possible added value of serverless-first strategies, we often start with a smaller piece of innovation that becomes the catalyst for a wider organizational transformation.

In addition, we support, coach, and stimulate companies to organize themselves in a way that maximizes the benefits of serverless-first. To this end, we help our customers to adopt agile principles and carry out the necessary changes to make the transition to serverless-first a success. To deliver our serverless support, we work with cloud providers such as Amazon Web Services (AWS) and Google Cloud Platform (GCP). In this way, we’ve been able to add value for several companies.

True agility in a global Media company

Merapar have advised on serverless architecture, developer recruitment, and Scrum methodology to a rapidly growing Innovation group in a global media company since its inception in 2019. For this department, teams needed the ability to quickly form around new product ideas and begin delivering and iterating fast, without relying on centralized resources that would inhibit scalability. Equally important was that hosting costs were low for early prototypes, but solutions were still ready to scale up quickly when required. With very short cycle times and each team deploying to production multiple times a day, teams had to be self-organizing and not dependent on other teams to ship releases.

A serverless-first strategy has proved beneficial for this rapidly growing team in all these areas. One case is the Sports Recap product. Using 100% serverless development, our developers were able to deliver a proof-of-concept (POC) in six weeks, then start production and launch to millions of customers in only two months.

The serverless pricing and scaling model has enabled an extremely productive developer workflow: ephemeral environments are created automatically in the Cloud for each branch in the code repository, enabling the full test suite to be run (unit, integration, contract, system/E2E). Manual exploratory testing is then executed on this environment and, following code review, can be merged to the main branch. At this stage, the developer conducting the code review can see whether the full test suite has passed, as well as reviewing the test code itself before approving the merge. Following the merge, the ephemeral environment is automatically removed, the build is deployed on the test environment, and the tests are all run again. After this, the build is deployed to the staging environment, smoke tests are executed, and it is finally deployed to production.

This process is so streamlined and predictable that senior business stakeholders have been happy for live deployments to be done on critical parts of the application during live Premier League football matches (typically taking about 15 minutes from code commit to production deployment). In this way, the serverless-first strategy has been a great fit for the innovation group- enabling a growing number of small self-organizing teams to rapidly innovate around new products and technologies, while still creating solutions that are secure, scalable, and cost-effective.

"We have moved on way past the issue of adopting cloud, we are deep into how to optimize our organization and teams for using not just cloud hosting but serverless approaches. This extends to team structure, for example we encourage our teams to eschew titles such as Architect, DevOps, or QA and instead these are roles that teams distribute as needed. If you are serverless, you don't need a distinct DevOps architecture to deploy, the infrastructure is tied in with the code, removing an external dependency that is so often a blocker." - Tom Des Jardins (Director of Innovation and AI at Sky)

Once the PoC had been demonstrated and further development approved, Merapar worked in two-week sprints to refactor the code and add new features. The nature of the serverless architecture meant that the system was already secure, resilient, and scalable even at the PoC stage.

Adopting a serverless-first strategy here enabled a small, independent team to move forward very quickly on a new idea to develop a functioning prototype, and then use that architecture as the foundation of what is now a very mature and feature-rich platform.

Bringing ideas to life quickly

A global OTT Operator experienced similar benefits when we provided a small development team to develop a new backend platform for their video streaming service, using a serverless- first strategy combined with Scrum agile methodology. Specifically, our team developed a Content Management System PoC with user segmentation and targeting in three weeks using the latest serverless technologies. All the Operator had to do was provide cloud accounts and access to the APIs that exposed the video content metadata.

"It's all about flexibility and personalization… The new platform gives us the ability to present users with a balance of editorial, segmented and personalized content, enhancing the overall customer experience. Increased viewing was the main aim of the project and we are pleased with the results and how quickly we achieved them." - Andy Clarkson, VP Technology Operations for hayu.

The time to go serverless-first is now

From quicker time-to-market, to lower infrastructure costs and reduced environmental impact, the advantages of a serverless-first strategy are undeniable – and can be enjoyed by a wide range of businesses across all industries. But, with these strategies requiring entirely new solution architectures, team structures, and training approaches, adapting is essential for success.

Indeed, it's these organizational changes that are set to make widespread transition to serverless-first systems possible. When companies start hiring teams with the skills to manage and develop serverless-first systems, they will no longer have the skills to manage traditional cloud architectures and networks. A future where software teams are unfamiliar with networks could be a reality in the next 5-10 years.

"Serverless is what's currently industrializing, becoming more of a utility - that's where you should be 10 years ago. Of course, Infrastructure as a Service, what we call cloud, is where you wanted to be. But now, you need to be thinking further up the stack." - Simon Wardley, Inventor of Wardley Mapping and Researcher for the Leading Edge Forum.[5]

Considering the current landscape, companies that don't adopt these strategies - and soon - are at risk of being left behind. Organisation inertia or teams aversion to change are no longer reasons to avoid adopting a serverless-first strategy. Taking small steps in parts of your organisation with the support of a partner like Merapar can make your transition as smooth and successful as possible.

Want to see how we can support your organization's successful transition to serverless-first?

Contact us today at info@merapar.com.

[5] https://stories.platformdesigntoolkit.com/mapping-doctrine-and-culture-in-the-future-of-organizing-with-simon-wardley-1aa1762a753e