software AG

# The pessimist's guide to microservices

## Intro

Microservices make businesses more agile! More secure! More connected! That, at least, is the rallying cry from the endlessly optimistic business leads who are looking for a silver bullet to solve all of their challenges. And you surely don't disagree—the potential is there! But with any conversation about microservices, there should be a big asterisk that says, "If all goes well."

This is your moment to shine. As a pessimist—or perhaps we should say a realist?—you know that with every problem microservices solve, there is the risk of a new problem created in terms of operational issues and governance. The kinds of problems that can bring an app to a messy, convoluted, grinding halt. And when this happens, identifying and fixing the issue is no small task.

The good news in all of this? You—the so-called pessimist—have the potential to be the hero of this story. Some may call you a stick in the mud. The truth is you're the glue holding the entire microservices operation together.

This guide is designed to help you approach conversations with your endlessly optimistic but perhaps underinformed colleagues. What are those pie-in-the-sky promises they make about microservices that make you roll your eyes? And how do you respond to them without coming across as the naysayer?

If you find yourself in these types of conversations daily, feel free to send the guide along to your optimistic colleagues directly so they can correct the errors of their ways.

**Table of contents**

# Stuff optimists say

# Stuff optimists say #1

Microservices will change our entire business model!

## What you might think:

Well yeah, in theory. But which part of our business model? Because some things are working just fine.

## How you should respond:

Awesome idea! Let's sit down and map out the microservices roadmap to the specific business goals to make sure they align.

## Why this is the right response:

You're totally right. When it comes to microservices, you should start with a simple vision and develop your strategy based on business initiatives. Once you understand how quickly a service or product will need to change as well as its growth potential, you'll be able to justify the investment in time and infrastructure required for a microservices project. The sage advice from Gartner? "Take a pragmatic approach to adoption by applying [microservices] selectively and incrementally, where [they] can deliver business value."*

*"Innovation Insight for Microservices," Gartner, Anne Thomas, Aashish Gupta March 4, 2019

# Stuff optimists say #2

With microservices, we can react to market conditions in a flash!

## What you might think:

They *can* make us agile, but it's more likely they will throw us into total chaos if we just start cobbling them together.

## How you should respond:

I'm all for agility! Let's think about ways we can be agile but also stay in control of microservices. They can get out of hand and keep us from reacting quickly if we have to sort through the chaos.

## Why this is the right response:

This is probably one of the most important points when it comes to microservices—how to manage the balance between agility and chaos. Microservices can indeed make your app development more agile out of the gate—but you run the risk of creating solutions that resemble the spaghetti code of the early 2000s. When the app breaks, it's hard to know where to start to find the issue, risking extended downtime. What's more, as business needs evolve and you need to make adjustments, you may not know where to start.

What you need is a platform that acts as a nerve center for all of your microservices—giving you visibility into what each service does, how they're being used, where issues arise and how they are performing. Something that will allow you to manage them with exacting control and complete transparency—just like you do with your APIs. This puts you in the driver's seat.

Once you are there—in the driver's seat, with total control over your microservices—you are in a position to react to market conditions quickly.

# Stuff optimists say #3

Microservices allow for distributed development!

## What you might think:

Yeah, so we can make our apps even more of a mess. When I hear "distributed," all I hear is "silos."

## How you should respond:

I can see how that could help! Let's think about ways to make sure we are communicating and collaborating so that every department can learn from each other.

## Why this is the right response:

It's true. A microservices approach to app development helps distribute the development process across an organization and can reduce the dependencies that one team has on another. This can be a good thing—but it can reduce collaboration and create an imbalance in the innovation and growth across the company.

It can also lead to a lot of duplicated work. What if you have two teams of developers who are working on similar capabilities in different departments? That leads to a lot of wasted hours and services that are similar, but built using different technologies or providers. Autonomy can be a powerful strategy, but only if you can also ensure efficiency.

So when you dive into a microservices approach, do it in a way that everyone has visibility into the work other teams are doing. Choose a platform that automatically exposes your microservices as APIs so they can be reused across teams and governed. Implement ways to create an open conversation across the development teams and to give visibility into each microservice stream that is or has been deployed.

# Stuff optimists say #4

Microservices will make our apps infinitely scalable!

## What you might think:

Well, we can scale with our on-prem servers as well. It's not something that's exclusive to microservices, so let's first start by optimizing what we have before wasting time trying to learn new technologies like Kubernetes® and Docker®.

## How you should respond:

Let's figure out where and why we need to scale, and then think about the hosting environment and development approach that will best accomplish that goal.

## Why this is the right response:

Using microservices in a cloud environment can allow you to scale services up and down with exactness, so you don't pay for server space you don't need; but it can also increase costs when you have a spike in app usage.

On top of that, there will be a learning curve in training the dev team to useall the new tools needed to deploy and operate services in the cloud. The time spent to learn these technologies can be significant—time that otherwise would be spent improving your app.

So as part of the equation, scalability can be a benefit. But—it is just part of the equation.

There are things that can help smooth this learning curve. For example, using a platform that's pre-integrated with standard open source tools like Kubernetes, OpenShift® and Docker as well as cloud hardware providers like AWS and Microsoft® Azure®, and being able to automate scalability *without* maintaining a huge library of prebuilt containers.

# Stuff optimists say #5

You want security? Microservices will practically make our app Fort Knox!

## What you might think:

Well, anything in the cloud is hackable, so I'm not sure why it would be more secure than on-prem alternatives.

## How you should respond:

You're right, security is so important! Let's look at the ways we can make our applications secure while also making them reliable and agile.

## Why this is the right response:

There is some truth in the theory that microservices can increase security. Since they are compartmentalized, they reduce vulnerabilities to the full app. So instead of using a monolithic API approach where all information comes in and out through the same gateway, microservices can do a better job of keeping data—and access—separate.

However, running any service in the public cloud introduces security risks. Fortunately, all modern cloud hosting environments and tools—like service mesh—have measures built in to keep networks safe, if properly deployed. And microgateways can add much more specific user-based security and data protection. Before deciding that a microservices architecture is or is not the answer to your problems, talk to your IT security folks and think about it in the bigger picture.

# Stuff optimists say #6

Microservices are so much more resilient!

## What you might think:

Well, yes, they *can* be, if everything is properly implemented. But for now, our on-prem solution is going to be more stable.

## How you should respond:

We would love to have the most resilient solution we can. Let's explore the options and map out a path, pulling in the benefits that a microservices architecture can offer.

## Why this is the right response:

There are some big issues that a microservices-based application must consider. For example, what happens if the IP address changes? In a traditional API-driven environment, this simple change would break the app. The good news is that if you have the right architecture in place, you can automatically handle issues like this in your microservices infrastructure to avoid performance impact.

Think of adding a layer of control that catches IP address changes, unexpected downtime and slow service responses—all handled through the reactive architecture. That's something that makes an app infinitely more resilient, but that would be impossible to do at scale without the right tools.

What should be particularly enticing is the idea of managing this with policy and not with code. Which means that getting all of these microservices more resilient is a matter of changing a few settings instead of a laborious coding task.

# Stuff optimists say #7

Microservices will reduce our time to delivery!

## What you might think:

Not these first ones. These ones are going to take forever because we are starting from scratch and there is a lot to learn and figure out.

## How you should respond:

Over time, you are right! They can get us to deployment faster. How soon do you need the next app deployed?

## Why this is the right response:

The premise is correct. A microservice application architecture will get apps out the door faster, with two caveats. First, if you have the right team to build them. And second, if you need the next app built faster, you'll want to investigate using tools to automate testing, deployment, and operations.

First, getting the right team. The skillset for developing on-premise apps is very different than using microservices. Some of this is a matter of technology—understanding how containers are used and how they all work together—and some of this is a matter of agile process. It may be a case of trying to let the tail wag the dog—using microservices to become an agile enterprise is the wrong way to think about it. You first need to think as an agile enterprise and understand the process, and from there can more quickly deploy microservices.

Second, the question of speed to delivery. If you're already using DevOps tools to build and deliver software, adding Docker and Kubernetes to the mix (along with the appropriate cloud vendor tools) is an extension of an existing process. And again, the principle to follow is start small and build as you go—adding new capabilities and operational recommendations with each microservices project you roll out.

# Stuff optimists say #8

Microservices will make us a truly digital business!

## What you might think:

It's going to take more than microservices to make us a digital business. In fact, we could be a digital business *without* microservices.

## How you should respond:

Absolutely! Many digital businesses have adopted microservices. We should consider it as part of our digital business strategy.

## Why this is the right response:

Becoming a digital business is a complicated goal, and in many ways a moving target as the digital world continues to evolve around us. There are scores of digitalization tasks that might precede the deployment of a microservices architecture. And until some of those are done, your microservices plan may run into roadblocks at every turn. (Many of these roadblocks are removed by having a proper data integration strategy.)

Depending on the nature of your business, microservices may be a step too far for your digital strategy. Gartner recommends that microservices architectures be deployed for businesses that have a need to "build dynamic systems." If your business doesn't have that need, then you may consider sticking with your current API-based approach without the additional complexity (and agility) that comes along with a microservices approach. But when the business need for microservices shows up, you'll be ready with pragmatic and sensible guidance to ensure it's successful.

# Conclusion

We love thinking about microservices and what they can and can't do. The good news when it comes to the optimists and the pessimists is that reality sits somewhere between their world views.

Microservices offer huge potential for agility at scale and for making real-time improvements to the apps digital businesses build and manage. However, they also present many challenges—including a risk that they spin out of control and create a confusing chaos of services without knowing where to start to untie a complicated knot.

This dichotomy between the optimists and pessimists—the opportunities and the risks—is what inspired us to build a platform as a bridge between these two worlds. The goal was to enable the best of the optimistic view of microservices and allay the concerns of the pessimist.

The platform is called webMethods AppMesh, and it is the nerve center that helps keep microservices under control. It gives visibility into what is happening with each part of the architecture. It flattens the learning curve. It brings a context layer so that each service can better serve customer needs in real time.

So as you are out there fighting the good fight to ensure that big, airy ideas are executable, you can turn to AppMesh to help make a microservices approach within reach and chaos-free.

For more information about our microservices offerings including AppMesh, please visit
softwareaggov.com/products/webmethods/webmethods-api-management/

**ABOUT SOFTWARE AG**

Software AG began its journey in 1969, the year that technology helped put a man on the moon and the software industry was born. Today our infrastructure software makes a world of living connections possible. Every day, millions of lives around the world are connected by our technologies. A fluid flow of data fuels hybrid integration and the Industrial Internet of Things. By connecting applications on the ground and in cloud, businesses, governments and humanity can instantly see opportunities, make decisions and act immediately. Software AG connects the world to keep it living and thriving. For more information, visit www.SoftwareAG.com.



ebook_microservices-pessimists_en

software AG