How Microservices Help Retailers Deliver on Their Omnichannel Fulfillment Promise

Microservices help retailers and commerce companies create exceptional experiences that build customer loyalty and support business growth. By embracing an agile mindset and using composable architecture to develop services, brands can enable an omnichannel experience that meets their customer needs wherever they are. A seamless customer experience – digital, mobile, online, affiliate, and in-store – starts with the composable architecture of microservices.

Creating an OmniChannel Experience Is No Longer Optional

How do microservices support retailers in delivering on the omnichannel promise?

In the post-COVID world, customers have come to expect – no make that demand – omnichannel experiences from their retailers. The brick-and-mortar companies that didn't fully embrace the unification of their offline and digital experiences half a decade ago are now having to play catch up. For many established retail enterprises, the challenge is to extend commerce beyond their core, legacy systems. Embracing the future starts with meeting the customer's omnichannel needs.

Managing that in a way that is seamless for the customer and efficient for the retailer (not to mention cost-effective to manage) is the secret sauce that every retailer is hoping to find, create, or build.

> "The pace of change for customer expectations has never been faster, yet it will never be this slow again. For brands to meet them they need to adopt agile systems like those based on microservices."
>
> Jason Goldberg
> *Chief Commerce Strategy Officer,*
> Publicis Groupe

Building Omnichannel Nirvana

For a truly seamless and robust omnichannel experience, you must be able to offer your customers an array of interchangeable options for inventory viewing, sales, delivery and return – that not only follows the customer from experience to experience but also from location to location. It must be able to optimize the backend functions so that the most effective, efficient option is offered.

Customers today expect a seamless experience across all channels. And they are less brand loyal than in the past. Today's customers have many options and they will pick the one that offers them the best experience – from selection or availability to simplicity of transactions and delivery, to pricing, to ease of returns.

How can legacy systems manage all that? In most cases, they can't. That's why so many retailers are turning to microservices to meet these ever-changing, ever-upgraded expectations. And it's not just about making the product available or finding a location that has the inventory. Customers expect the entire process to be simple. After all, they think, if Best Buy can do it, why can't you?

> "The bar for exceptional experiences in OMNI channel is very transparent to consumers today. Consumers will leave you pretty quick if you are not at or exceeding that bar."

Narayanan "Nari" Sitaraman
Chief Technology Officer at BARK

Legacy Systems Present Challenges

> "You can't treat in-store and online as two different business units. Today, consumers expect the same experiences across the board."
>
> Sree Sreedhararaj
> *CTO, SVP Sephora*

A typical enterprise commerce architecture encompasses several large, monolithic platforms, each managing multiple functions. Often, these operate in digital silos: ecommerce functions that were built for the web, and store systems function in the brick-and-mortar shops. But to be successful with omnichannel experiences, all of these digital commerce platforms need to integrate with each other and with other systems such as ERP, CRM, OMS, and PIM – not to mention emerging social channels and marketplace channels.

As the number of customer touchpoints expands, the architecture becomes more complex to manage, because the technology team has to update both the front-end interfaces and backend systems. Integrating these solutions is both complicated and expensive, and the technology team is under tremendous pressure to extend and maintain them. The end result doesn't often function seamlessly, and can even hinder business goals around innovation and customer service.

"You can't treat in-store and online as two different business units," said Sree Sreedhararaj, CTO, SVP at Sephora. "Today, consumers expect the same experiences across the board – whether ordering from their sofa or making a purchase in the store. So then you, as an organization, need to deliver that to the consumer demand."

What are Microservices?

Microservices are modular and composable applications that are independently developed, deployed, and managed. Microservices can be combined to create functionality that allows retailers to quickly deliver new features to meet customer demands and create greater efficiencies for better profitability. The key basic feature of microservices is that they use APIs (an application programming interface) to connect various applications both within the organization and outside of it.

Microservices Replace Monolithic Platforms

With complex architecture and specific functions defined based on (sometimes decades old) historic needs, legacy applications are often difficult to customize and expand in ways that allow interaction with new apps or functions. Since these systems are built as large monoliths sharing the same code base and tightly coupled, development changes – to accommodate new use cases or an innovation roadmap – may risk compromising the code.

"The monolithic applications which used to sustain before, whether you talk about ecommerce or any other monolithic platforms in general, has limitations in terms of how many changes you can do at the same time, how much of flexibility it gives you to go to market faster," said Sreedhararaj.

Furthermore, such web-based systems were rarely designed to support today's mobile and multi-touchpoint needs. Incorporating "headless commerce", where the front end is decoupled from backend code, can help tailor design and formatting, but not features and functionality.

Simple tasks like adding a buy button to digital lookbooks can't be done without APIs, and APIs can only work with a monolith's back-end code as is.

There's where microservices come into play. Microservices make use of APIs and allow ecommerce retailers to establish a **"composable digital experience platform (DXP)"** to deliver modular content and user experiences. In 2021, the Gartner Group talked about this and suggested that retailers must be able to support all levels of composable digital experiences to succeed in the future.

Gartner analysts predict that composable DXP will replace "heavyweight monolithic applications," which vendors will have to abandon due to changes in buyer behavior and market demands. To compete effectively, Gartner analysts suggest that vendors will need to improve the modularity of their products and prices based on consumption-based models.

Microservices vs Headless Commerce

You might have heard the term "headless commerce" used to describe this type of composable architecture, as well. Dirk Hoerig, founder of CommerceTools coined the term in 2013, but Gartner and some other industry experts prefer to call it "composable commerce."

The concept of headless commerce refers to an ecommerce architecture in which the front end (such as a website, social media channel, and so on) is decoupled from the backend functionality. By having an independent architecture, you can update or edit the underlying technology without interfering with the front-end user interface.

That means brands can create a variety of user interfaces based on their needs, sales goals, and target audiences. It also means updates and changes can be created more quickly, to address evolving needs or even unexpected interruptions such as the COVID-19 pandemic.

Microservices, or headless architecture, also make it possible to support updates in real-time. So if a third-party application, provider or seller makes changes, the changes are reflected right away instead of needing interface updates.

Unlike monolithic applications, which contain all the code in one system, microservices are small, autonomous units that address individual functions and work in collaboration with others to make an application work. Because of this, they are a good choice for developers who need to deliver large, complex applications quickly, frequently, and reliably.

Microservices architecture is a modern approach that solves the pain of legacy architectures and siloed systems. Small applications are built around a single business purpose. With their own data stores and well-defined APIs, they can extend to any new touchpoint, regardless of user interface or context. In contrast to a monolithic application that can only serve commerce in a rigid way, microservices can be deployed independently, enabling businesses to use only the services they need.

Because microservices are small, decoupled applications, they can be updated quickly without compromising the code of other services or the extensive testing required by monolithic platforms. Organizations with aggressive digital transformation roadmaps can deliver new code much faster and with less risk. It is easy to launch and remove new experiences without compromising critical sections of the system as data moves freely through the organization.

Four Key Benefits of Microservices

- Multiple teams can work simultaneously on different elements of a distributed system, which improves productivity and speed
- Provides developers with the ability to selectively deploy components of their architecture without causing major disruptions
- The infrastructure can be used with containers (e.g., Docker) easily
- The best tool for the job can be chosen for each application or need

Why Do Businesses Use Microservices?

Agility
One of the key reasons businesses use microservices is for speed of implementation. Microservices provide retailers with agility in meeting customer needs (delivering on the omnichannel promise) without having to "rip out" their existing systems or replace them.

Scale
The ability to rapidly scale on both the front and back end is a key reason to use microservices. This type of architecture allows companies to scale across networks with minimal store-level training needed. Plus, multiple teams can work simultaneously on different elements of a distributed system.

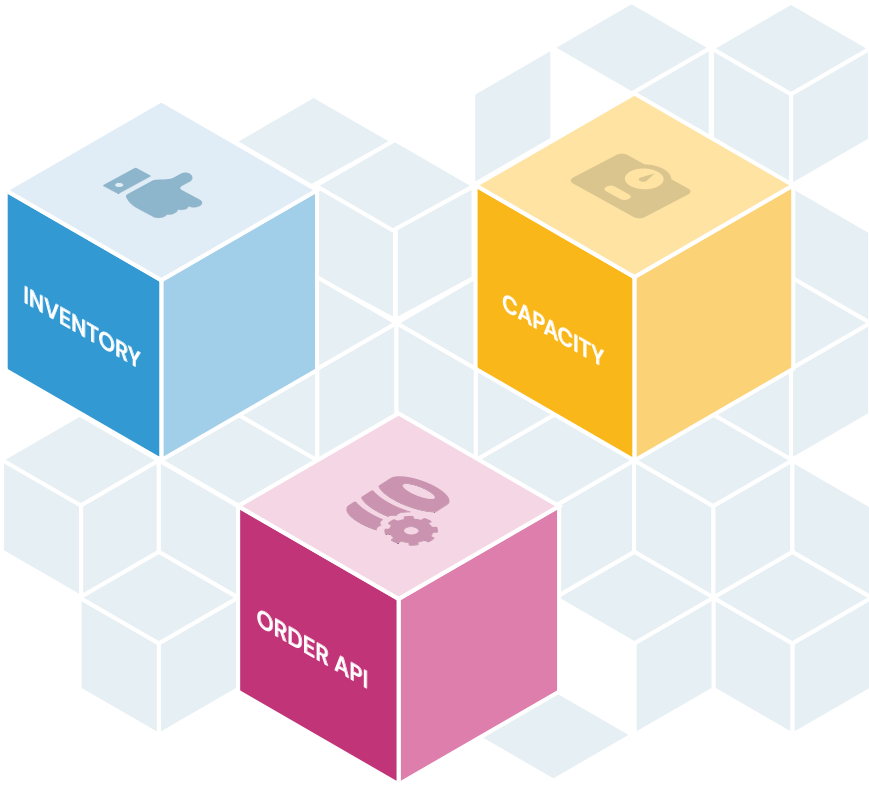
Flexibility
Microservices allow developer teams to selectively deploy specific components of their architecture — without causing major disruptions, and to introduce or update new services or features as the market demands. It also allows you to build once and reuse the same elements.

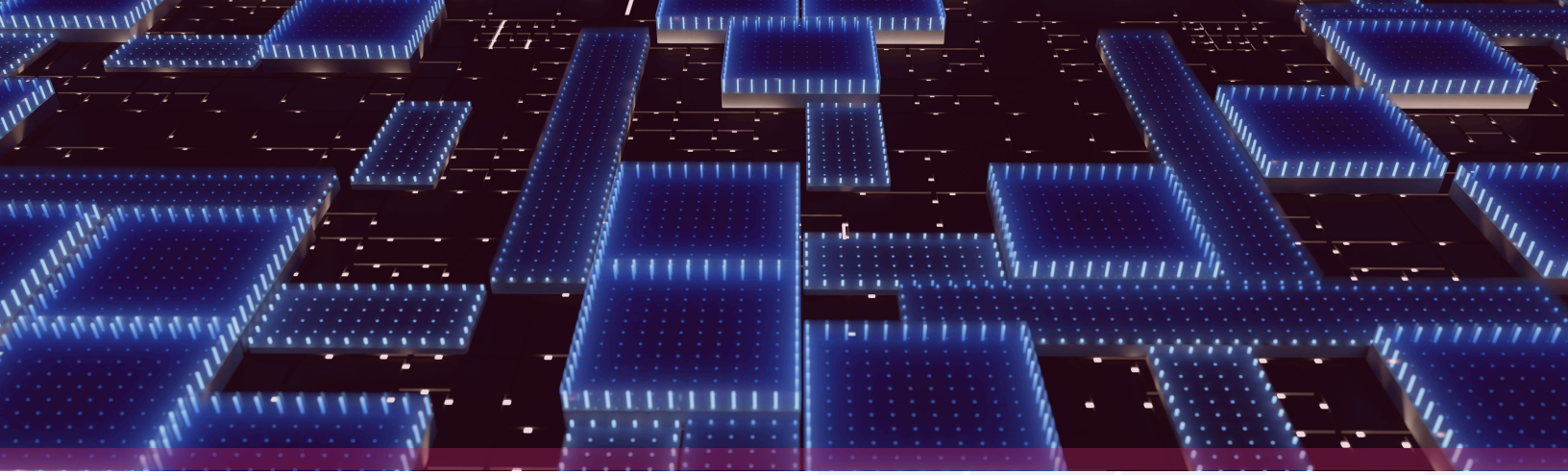
Fit to Need
Microservices make it possible to choose the best tool for each problem. This means you can select the option to upgrade only the parts you need to, whether that means inventory management, BOPIS, shipping support, or other areas.

Better Customer Experience
Microservices allow you to create better customer experiences by offering the types of services and features that customers want, which builds customer loyalty.

Reduce Costs
Using microservices reduces the cost of ownership (including updates and changes) and cost of integration with third-party services.

Maintainability
When you break down the code into more modular segments, it makes it easier to log and monitor. In this way, the ability to detect problems and manage them grows considerably.





Where Do You Start?

You need to start somewhere, and that should be based on your business needs and goals. Determine what the most important issue to solve is, or where you'll experience the best ROI or improvement in customer service.

Although at a basic level building microservices is a coding exercise, it's important to keep in mind that you're solving a business problem. So you need to make sure you approach it from that angle: what are your primary business needs? Or what primary problems do you need to solve?

1. The most important thing is to identify your business needs, what are the challenges you need to solve? Where are the bottlenecks? Where is the demand? In what areas are you lagging behind your competitors?

2. Second, you need to understand your customers. What are they looking for? What expectations or experiences are required? Understanding your customers is the only way to build the journey successfully.

> "Start separating out monolithic applications from your heads – or the site aspects and experience aspects – to deliver the biggest benefit."
>
> Sree Sreedhararaj
> *CTO, SVP Sephora*

"Developing microservices is a journey," said Sreedhararaj. "You can't expect to take a 'big bang' approach. You need to start with baby steps. Learn from what others have done, where they have failed, and where they have succeeded."

"Before you start the microservices journey, you should think about creating a headless approach," he adds. That means considering how you can decouple your functions from their purpose. You need to be able to expose key functionality and data through individual APIs.

Solve for Specific Business Problems

An important aspect of planning for microservices is solving for a specific business problem. You want to use as few steps as possible, so define the problem in the most specific way possible, without going too granular, which would make maintenance and updates more difficult. You want the microservice to be specific but also reusable.

"So the key thing there to look at is how many hoops or levels are you going through to define a business problem? If you're going through more than two or three, then you're going to end up in a situation where you're going to find observability and traceability a big challenge," said Nari.

In practice this may mean, for example, breaking down fulfillment into two or three steps: inventory, picking, and shipping.

> "Microservices just for the sake of modernization, that's the fastest way to see your initiative die."
>
> Nari Sitaraman,
> *Chief Technology Officer, BARK*

An Example: Solving for Multiple Shipping Locations

One way to do this is to determine the specific steps or services that require data input from disparate locations. So in the case of Sephora, for example, the company has built a series of microservices to manage order fulfillment.

“One microservice determines inventory availability. Another sets the shipping. Another was created to manage the packing slips that go in the box,” said Sreedhararaj.

Similarly, Nari described a series of **fulfillment microservices** built for Williams-Sonoma that managed these types of functions. The shipping microservice could determine, for example, whether ground shipping or next day ship from store should be used.

“It identifies, based on where you’re shipping from and where you’re shipping to, what’s the most optimal way to get the product at the cheapest cost yet meeting or exceeding customer expectations,” said Nari.

In addition, another separate microservice sets the packing slip. Typically that sort of function is built into the warehouse management system. But decoupling it and offering it as a microservice allows it to be delivered based on the point of shipping, even when drop shipped from a vendor, and allows more customized marketing messages to be printed for each customer.

“So breaking this down into concepts that make sense is one way to ensure that we can reuse this and maintain this microservices more effectively,” said Nari.

An Example: Solving for a Customer Need

Nari gives another example from Crate and Barrel. The company works with a lot of mom-and-pop manufacturers of furniture. They may not have access to complex inventory systems or have a computer hardwired to the company's internal system. But whether a customer places an order that includes items shipped from a warehouse or an "on demand' shipment from a mom-and-pop artisan, the customer experience needs to be the same.

"As a customer, if you order both furniture as well as a tabletop place setting, you're not going to be interested in trying to figure out for yourself when things are going to arrive and from where," said Nari. "You want the retailer to provide that in a way that it looks seamless to you, even though that information might be coming from disparate sources."

Sometimes the solution to a problem comes from a business angle, such as putting a contract in place that requires partners to update order status or shipping status in a certain way or at specific intervals. Using a microservice can make that sort of third-party update easier to manage.

In this case, you could have a microservice tied to an app that says "the order shipped" and allows the supplier to enter a confirmation code using a mobile phone or digital application, or "shipment is delayed" and the supplier can enter the appropriate information to update the customer.

Why Use APIs to Create Microservices?

APIs enable digital transformation by connecting business applications.

- Share data between business applications
- Enable faster updates, iteration and response
- Open new ways of doing business
- Build value for customers
- Integrate services
- Scale faster

Are You Ready for Microservices?

The question of whether or not your organization is ready to implement a microservices approach is a bigger call than just being able to deliver features to the customer. You also have to think about whether it makes sense from a business perspective, and whether you have the skills and capabilities internally to manage it.

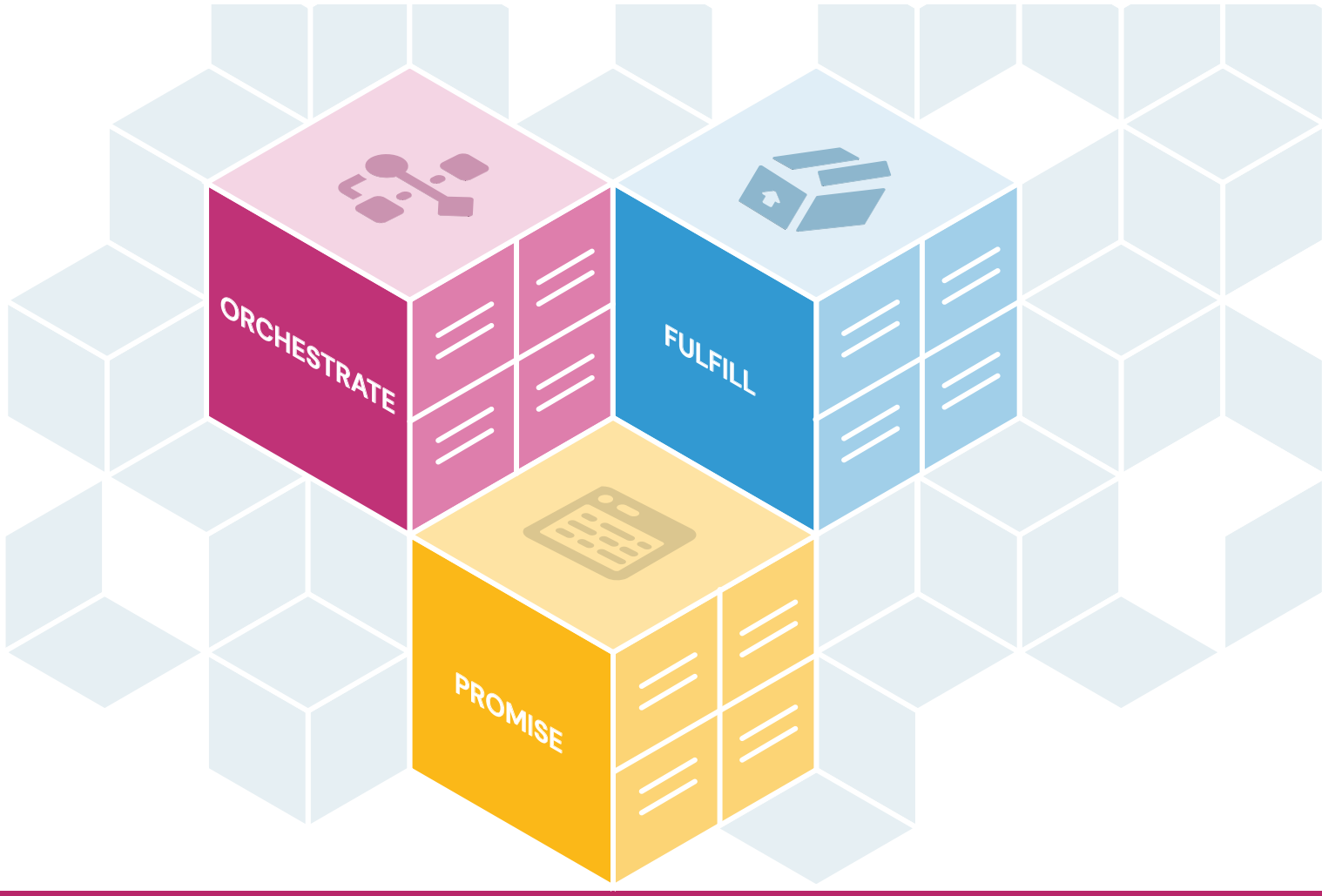
Do you have a strategic plan for developing microservices? Because it's not just a one-year goal that you can reach, deliver a set of microservices, and then be done. It has its own ongoing roadmap management, support and maintenance. Plus, a whole lot of things that you need to think about when you take on the journey.

Microservices might not be for you if the size and scale of your business don't match up in a way that makes microservices efficient. If you're mainly working with large bulk orders, for example, microservices that can adapt to individual orders might not be necessary.

In addition, you will need to think about whether or not you have the capability in-house to manage a shift to microservices, or whether you need to use system integrator partners, or partners who can develop microservices to augment the internal skills available.

Need help with microservices development and management?

Nextuple offers a microservice management platform as well as expertise in creating composable microservice to meet specific needs. Contact us to learn more!



About Nextuple

Retailers need a better way to increase fulfillment capacity without breaking the bank. Nextuple offers a combination of software and logistics services that enable mid-market retailers to fulfill orders faster using existing infrastructure.

The Nextuple Platform gives retailers the speed and flexibility they need, at lower cost, while solving the logistical challenges using the existing fulfillment network.

A new fulfillment strategy that pulls inventory from multiple stores, consolidates shipping into fewer packages and makes any product available for pickup anywhere in that market.

Talk to us today

nextuple

USA – Massachusetts
1 Elm St Unit 2C,
Andover, MA 01810 USA
www.nextuple.com