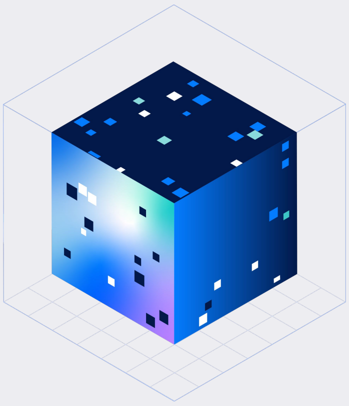
Talon.One

Headless commerce & microservices explained

An introduction to headless commerce and microservices-based software stacks



“Headless commerce & microservices are the future of SaaS”

Christoph Gerber, CEO & Founder, Talon.One

In comparison to traditional ecommerce software solutions, headless, microservices-based software offers a huge range of operational and business benefits.

What’s inside?

→ Here are the top five takeaways from our exploration of headless commerce and microservices

1

What is headless commerce?

Headless commerce decouples front- and backend functions, allowing increased flexibility and agility in how information is displayed across channels.

2

Traditional commerce v headless

Traditional commerce can mean a calcified, hierarchical approach that is slow to change, doesn’t deal well with local requirements, and is difficult to personalize. Headless allows greater flexibility with less upkeep.

3

Benefits of microservices

Backend changes aren’t required to make changes on the frontend. This frees up developer time and gives more flexibility for changes on the frontend, since any changes only need to be made once, in one place.

4

‘Best of Breed’ microservices

‘Best of Breed’ microservices are the specialists in their niche. By providing a service that is more powerful and customizable than an in-house or bundled solution, they allow firms an unprecedented level of customization and add value by being the best at what they do. By bundling a number of ‘best’ microservices, you get broad coverage and deep features.

5

Choosing a headless service

There are four key questions you should bear in mind when deciding which headless microservices to choose; what do I want? What do I need? How much does it cost? And does it work well with other microservices?

If you keep those four questions in mind, you’ll find the best possible fit.

Important definitions

→ As more companies adopt headless software, here's your cheat sheet to understand the developments

Let's start with a useful analogy that should help you understand the difference between headless and legacy ecommerce systems:

The traditional ecommerce approach is like running an international retail business with just one head office that makes all the decisions. Strategy would be defined without local knowledge, and employees in each country would have to use translators to communicate with management, with no say on how the local business should be run.

The headless commerce approach, meanwhile, would use regional head offices in each country of operation. They would coordinate local strategy with local employees in the local language.

Two core components in the wider machine that is headless commerce are microservices and APIs. Microservices are the regional head offices, and the APIs are the translators.

Microservices

Much like the headless approach, the microservices approach favors modularity over a more rigid, monolithic design. Under this approach, software solutions are built from standalone parts (microservices). These microservices are referred to as 'best of breed' or 'best in class' because they each serve their own specific niche or function, e.g. promotions or payments. Because each microservice is specialized, they tend to outperform any bundled or in-house software solution that may include features that serve the same purpose.

APIs

APIs, or Application Programming Interfaces, are the backbone of all headless, microservice-based software systems. They're pieces of code that allow separate microservices, applications, or parts of applications, to communicate with one another. APIs come in a variety of shapes and forms, but they all serve the same purpose.

What is headless commerce?

Businesses turn to headless ecommerce for its enhanced customization and scalability over traditional ecommerce solutions

Driven by the need for targeted and personalized customer experiences, headless commerce has quickly become the go-to approach for innovative businesses selling online.

It's powered by headless applications, which decouple the frontend (webstore layout and design) and backend functions (product information, promotion logic, payments, etc.).

Rather than your commerce stack being one single inflexible entity, as in traditional monolithic systems, this approach connects the frontend and backend via API, allowing them to function fully independently.

This enables agile responses to problem solving and deeper customization.

In this ebook we'll explain how headless commerce works, what makes it so effective and offer tips and best practices to help you prepare your business to go headless.

92% of 1502 survey respondents reported at least some success with their switch to microservices-based commerce.

O'Reilly

Making the choice

→ An overview of the differences between headless and monolithic software systems

By bundling a number of 'Best of Breed' microservices, you get broad coverage and deep features.

Headless software

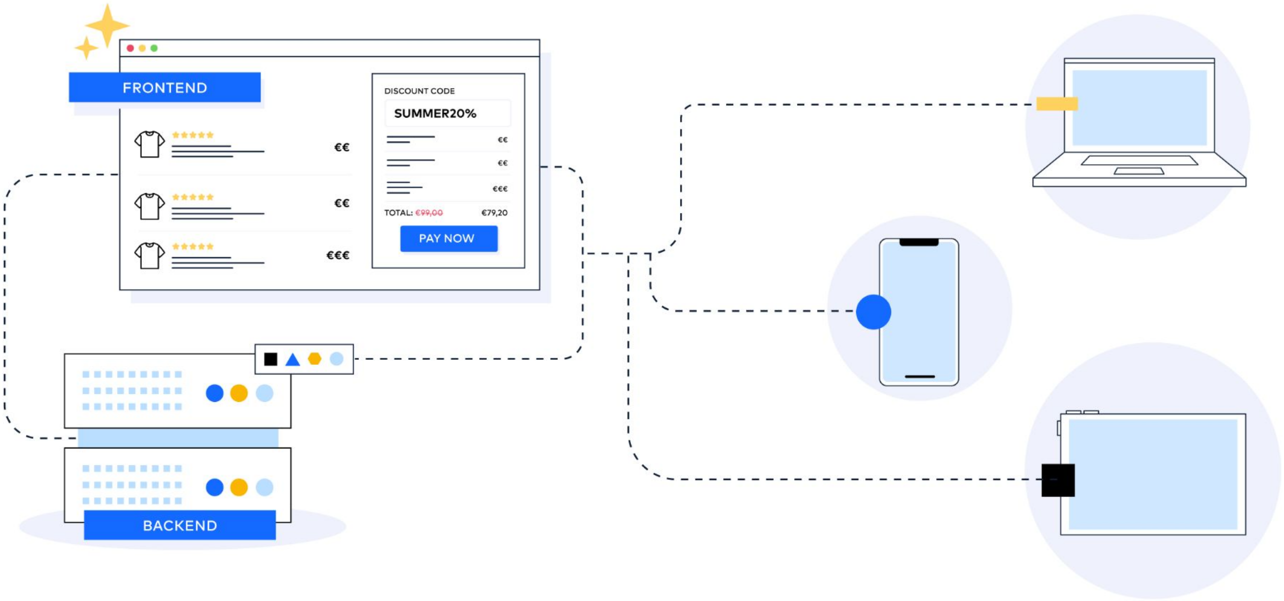
Headless software is backend software (the bit that users can't see) that functions without a frontend (the bit users interact with). Headless software can be coupled to many different frontend interfaces at once and, as a result, there's no need to manually code layout changes between different interfaces/channels.

- ✓ Highly flexible across many different frontend platforms
- ✓ Design can be customized as much as you want
- ✓ Making changes to the system is quick and less labor intensive than monolithic systems

Monolithic software

Rather than separating the backend and frontend as independent parts, the traditional approach to software architecture integrates them more tightly as one unit. In many fields, this 'monolithic' approach is becoming less popular because of the various advantages offered by headless software.

- ✗ Frontend changes are limited by the connection to the backend
- ✗ Much fewer customization options
- ✗ Coding changes is more difficult, and there's a higher chance of causing disruption to the rest of the system



The benefits

A fast food restaurant wants to launch a special standalone website to help promote a new menu item. They want the website to:

- Include a promotional minigame that awards users with points/credit that can be used on the restaurant's app
- Link back to the restaurant's standard website with the option to immediately buy the new menu item
- Offer full functionality on all available channels (desktop, mobile, tablet, etc.)

Using a traditional system

Making these changes with a traditional ecommerce system would require a lot of work from the restaurant's developers and it would be much harder to amend later.

It would also have to be built to the existing frontend specs. This comes with tradeoffs that would limit the scope of the project.

1. Because the frontend is tightly coupled to the backend, backend changes would also have to be made to accommodate the new website/display. This could result in disruptions to the rest of the restaurant's online services.

2. To avoid making backend changes, the restaurant would have to build the new site using the same ecommerce template as the existing one. This would restrict the layout and design options available to them, and make the new page less responsive.

‘Best of breed’ microservices

→ Perhaps the biggest benefit of headless commerce is the option to integrate diffe microservices

Although it may sound much more complicated to combine a number of separate software applications to get the functionality of a single platform, this approach actually tends to provide much greater functionality than monolithic systems.

“Modularity is a big draw for businesses who want to stay agile in a competitive business environment”

Some of the key benefits of a microservices-based architecture are:

- Shorter development cycles
- Easier to scale and maintain
- Easier to build around your own business requirements

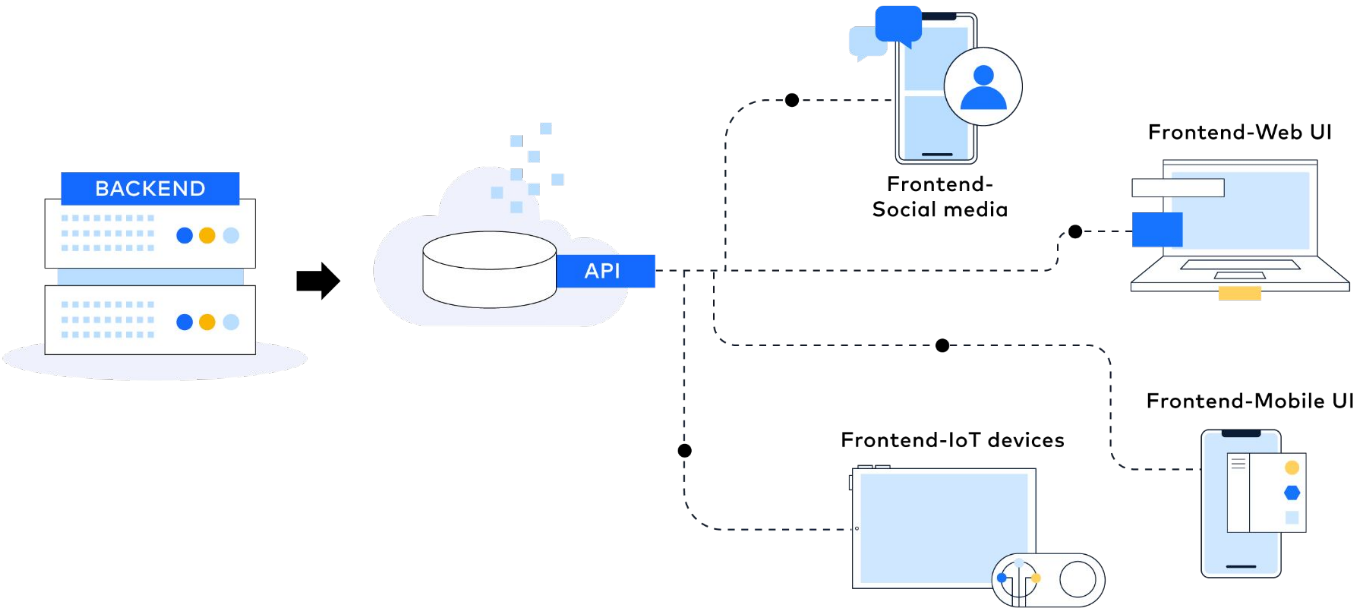
Omnichannel is now the go-to strategy to provide the personalized and tailored experience that the modern consumer expects — and microservices are the most effective way to deliver it. But not all microservices are built the same way, and some are more effective than others.

To deliver an omnichannel strategy effectively, you need quick, agile software systems that can deliver a consistent experience no matter how customers decide to shop.

But how should you go about picking best of breed microservices for your software stack?

What should you prioritize and keep in mind when choosing between competing products?

There are four key questions you should bear in mind when making this decision.



Using headless commerce

Using a headless ecommerce system, these changes could be tested and implemented more efficiently.

The minigame could be build separately, and then linked with the necessary interactive elements from the ecommerce system (stock, payments, product data, etc.). This would allow customers to buy the new menu item via the game. The restaurant could then easily add more complex logic to the game/site and not have to worry about poor performance. Using the headless commerce approach would benefit the restaurant in a number of ways.

1. Not only would it require less work from the developers, the game and the other elements it uses would be more responsive and less likely to cause disruption to the rest of the business's sales channels if something went wrong.

2. The headless commerce approach would give the developers much more freedom to create a unique new channel that will contribute to overall objectives of the promotional campaign.

3. It would be much easier to update the new website/game in the future.

Factors to consider	Additional considerations
Which functions do you want?	
Think about the software functions and features you would pick if you could build the best possible ecommerce system (within reason) for your business right now.	● Once you've picked your most desired functions and features, split them into groups based upon the type of microservice that would best be equipped to provide each. ● This will help you select the most efficient combination of microservices to cover all your most desired features. Remember, it's unlikely you would get anywhere near this level of functionality with a monolithic system.
Which functions do you need?	
Now, rather than listing the functions you want, list the core functions you *need most*.	● Once you've picked out the core must-have functions and features, use them as a template. By comparing and contrasting the features you want with the features you need, you'll be able to assess the real value each microservice can provide to your business. ● It's not always possible to build the perfect ecommerce solution for your business. But being able to determine which functions are essential, and which are a luxury will help you choose between competing microservices in each space.
What are the costs?	
Each microservice comes with its own costs (subscription, maintenance, upgrades etc.). To pick the most cost-effective solution from a number of different competitors, you need to weigh up features against price.	● Ultimately, the cheapest solution isn't always the best. One microservice may be cheaper than its competitors in the same space. But if it doesn't offer the same functionality, choosing this platform simply because of price is a false economy. ● Instead, you should prioritize features first and foremost. It's software's features and capabilities that add value to your business, and having to upgrade to a different platform at a later date will only cost you more money in the long run.
How easily can your selected microservices be integrated with one another?	
It's theoretically possible to integrate any headless microservice via API. But integrating some platforms requires more work than others.	● Check which microservices already offer integrations with others. If there's an established partnership in place, the integration process will be much easier for your own dev team. ● You should also check whether the microservies you've selected have been built following the core headless software principles (see next page).

The future of headless commerce & microservices

→ Headless commerce and microservices have a bright future ahead

Given the shift towards omnichannel, personalizable and agile systems we're seeing in ecommerce and other industries, headless commerce and microservices are only set to increase in popularity. More and more businesses are seeking flexible, scalable systems that will support their online sales operations.

Currently, enterprise-level businesses are the main market for headless commerce and microservices-based systems. But smaller businesses are increasingly seeking out flexible, scalable systems too. **The barrier to entry is lower than it ever has been, and fully headless ecommerce systems are affordable for non-enterprise businesses too.**

For these products to keep providing the same benefits to organizations, no matter their size, uniformity and interconnectability are essential.

To work effectively as part of a wider headless software system, each microservice or component software should comply with some key principles. These are:

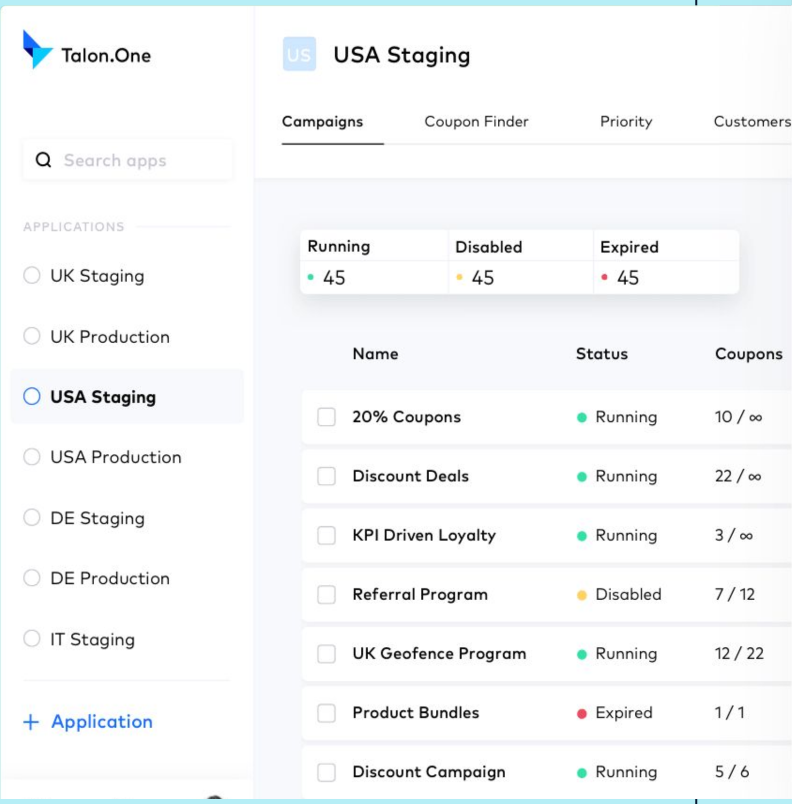
- API-first
- Cloud-based
- Security-focussed

Because consistency among headless, microservices-based software solutions is so important, the MACH Alliance (**M**icroservices, **A**PI-first, **C**loud-native, **H**eadless) was created to help promote standards throughout the industry, and provide businesses with guidance and support.

As a member of the MACH Alliance, we're doing our bit to ensure businesses in any industry can integrate a powerful, flexible promotions software as part of their software stack.

Talon.One's solution

The Promotion Engine allows you to tailor loyalty programs as much, or as little as you need to



Running	Disabled	Expired
● 45	● 45	● 45

	Name	Status	Coupons
□	20% Coupons	● Running	10 / ∞
□	Discount Deals	● Running	22 / ∞
□	KPI Driven Loyalty	● Running	3 / ∞
□	Referral Program	● Disabled	7 / 12
□	UK Geofence Program	● Running	12 / 22
□	Product Bundles	● Expired	1 / 1
□	Discount Campaign	● Running	5 / 6

Setting up flexible and personalized campaigns across channels is quick and simple using Talon.One's Campaign Manager. Our headless, API-first platform can be integrated alongside a huge variety of business software to create a custom solution for any business.

With the Promotion Engine you can adapt the inputs and outcomes of your campaigns to meet your current business goals, and adjust them accordingly using the wide variety of customization options.

Many promotion services are restrictive in their customizability and the number of integration options they offer.

At Talon.One we understand how important these options are when it comes to creating promotional campaigns that provide value to your business.

We don't limit the functionality of our software, because we understand that the ability to personalize is key to building and running successful promotional campaigns, whether you've opted for referrals, coupons, loyalty programs or all of the above.

Book a demo online today and discover how to build your loyalty program with Talon.One.

Talon.One

The World's Most Powerful Promotion Engine

Create and deploy creative and targeted campaigns
using any of your customer and live session data

MAKE YOUR PROMOTIONS POSSIBLE

promotions@talon.one

BOOK YOUR FREE DEMO

talon.one/book-a-demo