

Order Management: A fresh perspective

From monolith to microservices and beyond





Digital Agility is Essential

Consumer demands are evolving fast. The only constant is change and the rate of change is accelerating quickly. Customers today expect to be able to shop where and when they want and they expect convenience. Digital agility is essential.

The seamless Amazon shopping experience is the new standard for what customers expect. The tech required to keep up with customer demands needs to be quick, flexible, and reliable. Amazon does <u>1.6 deployments per second</u>. So, how do you keep up? And keep customers coming back for more?

Meanwhile, the number of sales channels and fulfillment locations has exploded. It's not just your core ecommerce site, but marketplaces, regional websites, and social channels. At the same time online orders are being fulfilled from more stock locations than ever before. In addition to DCs and stores, there has been a big uptick in the use of 3PLs and Drop Ship Vendors, and other retail partnerships. But when you add new sales channels and stock locations it increases the complexity of an already complex business problem: order fulfillment operations.

The Monolith



Rigid legacy systems are hard to change and customizations block your upgrade path to access new features

Inflexible

Dictates where you can add custom code. Other customizations require core code updates.

Slow to create value

Difficult to develop and deploy.

High Total Cost of Ownership (TCO) Expensive to maintain.

Ecommerce fulfillment has become complex

Did you know that if you ship from 1000 locations, for every order that's split into 2 shipments there are potentially 499,500 ways to fulfill an order? That's a lot!

Making sure you can orchestrate orders across channels and locations in a way that maximizes the margin on each order is more important than ever. And while Commerce platforms and ERPs are great, they weren't designed to make these types of decisions, at scale, in near-real time. Nor were traditional monolithic systems.

Monoliths block innovation

Rigid monolithic architecture, by definition, is 'composed all in one piece'. So, how does a business digitally transform with a monolith? They have to re-write different pieces of the product, a few at a time, and one team can't deploy their new code unless all teams' initiatives are ready. This big bang has cross-functional dependencies that create delays and risks. Plus, massive portions of the IT budget are allocated for maintenance. Not to mention the time lost. Upgrades to access new features take months or even years to implement. Overall, more time and budget go towards maintaining a system than actually delivering value.

What's more, you are reliant on a vendor's rate of innovation. And if a vendor is acquired, their

FLUENT ORDER MANAGEMENT

Change efficiently and scale dynamically

Best-in-class solution with scalable architecture for full ownership and control of your order fulfillment operations

Implement faster

Lower development costs

Adapt quickly to ongoing change



rate of innovation will slow. In addition, vendor roadmaps may not align with your strategic objectives, and at the end of the day, no one vendor could do everything well, so 'suite solutions' always require a compromise. There has to be a better way.

MACH architecture unlocks innovation

The MACH (Microservices-based, API-first, Cloud-native SaaS, Headless) architecture is the modern approach to building software that's more agile, scalable, and flexible and unlocks innovation in several ways:

- Microservices-based: MACH architecture is built on a microservices-based approach that allows for the creation of smaller, independent services that can be developed, tested, and deployed separately. This enables faster development cycles, reduces the risk of system failures, and allows teams to work on different services in parallel.
- API-first: With an API-first approach, MACH solutions can be integrated more quickly and easily than monolithic systems to provide a seamless customer experience. What's more, mature solutions offer several API patterns to support the many different use cases that enterprises require, for example REST, GraphQL, Synchronous,



Asynchronous, web hooks, standard APIs (for single record updates), batch APIs (that can process multiple records).

- **Cloud-native SaaS:** Cloud-native solutions are optimized to take full advantage of the services provided by the cloud vendor. This results in solutions that are more scalable, resilient, and cost-effective than traditional applications.
- Headless: In the context of order management, when the front-end and back-end of an application are decoupled it means you can use other systems to control and update how orders are processed. This might include Point of Sale, call center, or any custom built applications, to reduce training and change management, and increase efficiency.

By combining these four elements, MACH architecture allows organizations to develop and deliver applications faster, improve the customer experience, take an incremental approach to rollout, and adapt quickly to changing market demands. The result? A faster speed to market and return on investment.



What makes Fluent Order Management's approach to **Microservices different?**

No one vendor can provide a 'perfect' solution, so rather than trying to build for every use, Fluent Commerce is focused on providing the tools and components you need to:

- Adapt quickly to changes in customer preferences, and
- Use fulfillment as a competitive differentiator.

Fluent Order Management breaks down microservices into even smaller components. Both in frontend, and the business logic. And each component can be independently configured and deployed. Let's take a look at how it works.

Micro Frontend Architecture

Fluent Order Management uses a 'micro frontend' architecture so businesses can tailor the UI to meet their requirements. While it provides a library of pre-built UI components that you can configure, we understand that every business is unique. So we also provide an SDK that lets you extend the solution by creating new UI components to fit your business needs. And because you can independently modify and deploy UI components, you can innovate and deliver business value faster.

What kind of components? Anything you want. This could include tables, views, new dashboard tiles, new dashboards, new input screens, etc. You can even create components that display third party data. And a similar approach is applied to your business workflows.



Flexible User Experience

Configure and extend the user experience to fit your business. With Fluent Order Management you can easily create new components, even ones that display third party data.



Micro Workflow Architecture

Just like the front end, Fluent Order Management's workflows (that contain all your business logic) use a micro architecture as well. Each workflow is made up of reusable rules. You can choose to use rules from the library provided as part of Fluent Order Management, or create new rules. These rules are combined into rulesets that perform different functions. This flexibility empowers you to optimize fulfillment.... *your way*.

Flexible Workflows



Configure and extend the business process workflows to fit your business. With Fluent Order Management you can easily create new rules for to support unique sourcing logic, or to trigger or respond to events in other systems.



The possibilities are endless. For example you could:

• Prioritize fulfillment based on location type, such as store, dark store, or warehouse



- Create your own attributes against a location that you can use in your fulfillment logic, such as
 - Current number of open orders at a store (so you can route orders to the store with the most capacity)
 - Maximum number of open orders (so you don't overburden a store even if it's the closest)
 - Last pickup time for carriers (so you don't route expedited orders to a location that won't be able to pick it in time)
- Create your own rules around the maximum number of split shipments (for example by order value or delivery distance) and prioritize how the order will be split, for example:
 - \circ $\,$ $\,$ First try to ship from two DCs $\,$
 - \circ $\,$ $\,$ Second try DC and Store $\,$
 - Third try Store and Store
- Bulk reallocate orders in the case of a backlog
- Automatically use transfer strategies during fulfillment to reduce delivery costs (e.g., DC to DC, DC to Store, Store to Store)
- Accept pre-orders, allocate against future inventory, and automatically fulfill orders once stock is available

Your fulfillment optimization strategy shouldn't be limited by any vendor.

Fluent Order Management

Get the digital agility you need to grow your business





Future-proof your customer experience

In the world of composable commerce the power is in your hands. How you build will define you. Imagine the digital agility you could achieve using MACH architecture. How much faster could you innovate and deliver value to your business? What if you had more control over your fulfillment logic so you could deliver on time while minimizing delivery costs? And craft seamless, best in class customer experiences that will set you apart from the competition?

With Fluent Order Management you can do all this and more. Handle complexity, deliver value faster, and future-proof your business. Sound interesting?

The need for speed makes traditional systems obsolete

Legacy systems used to work... but they don't anymore.

Ready to learn more?

Get a demo



fluentcommerce.com