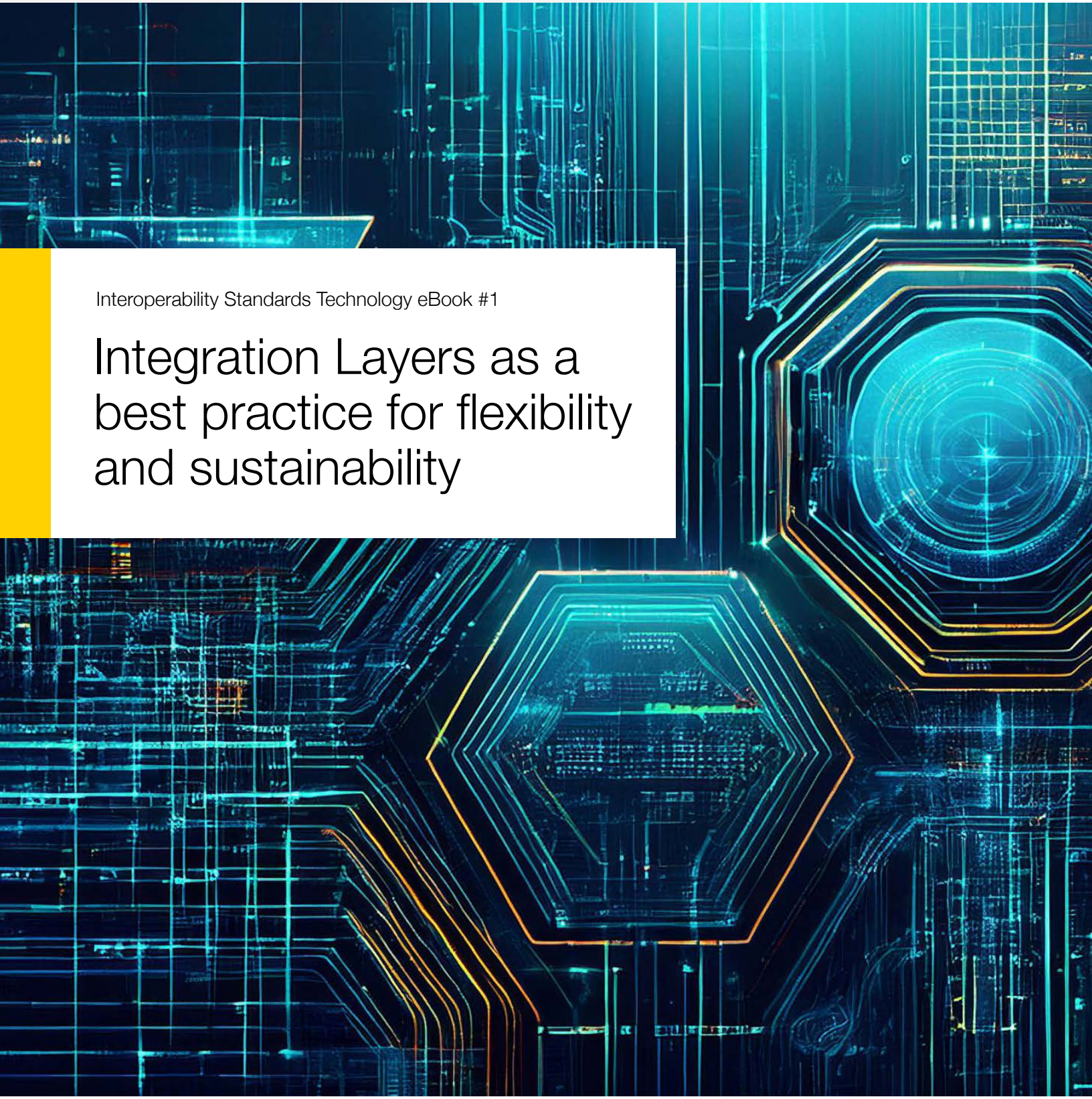
MACH ALLIANCE



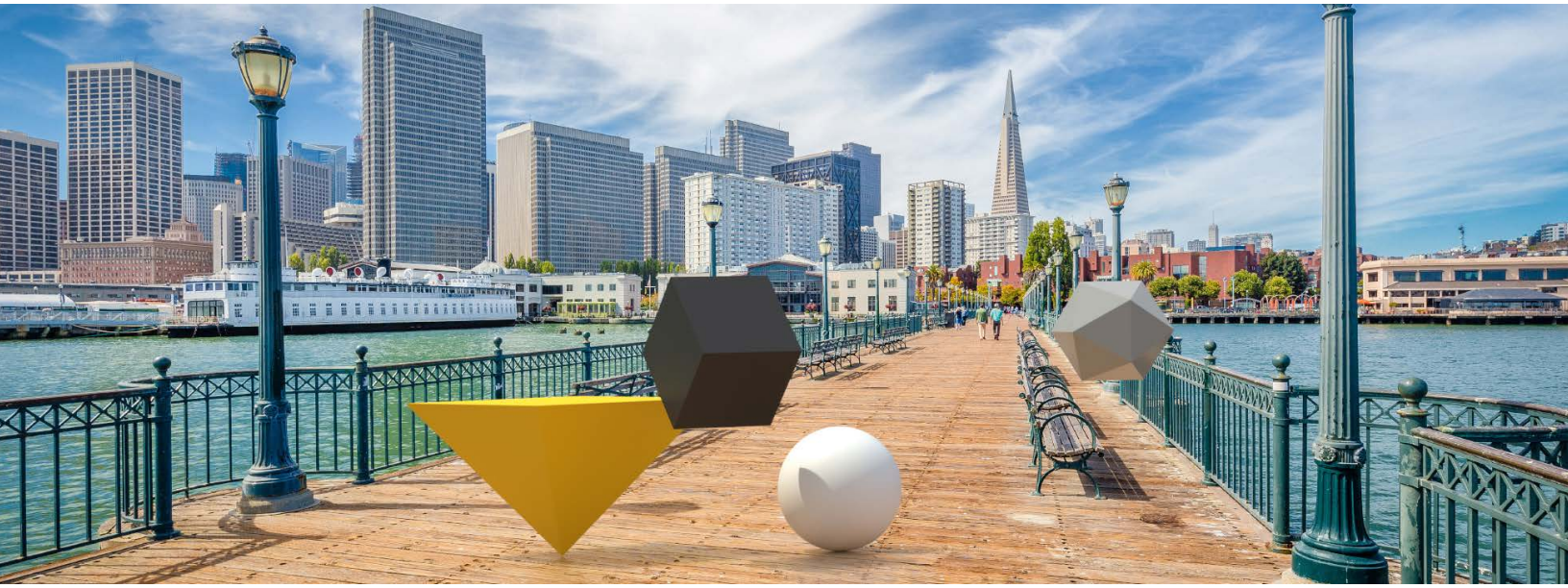
Interoperability Standards Technology eBook #1

Integration Layers as a best practice for flexibility and sustainability

2025

Introduction

The MACH Alliance Interoperability Task Force has produced a number of resources and whitepapers in order to help business leaders and architects within organizations understand the common patterns (and anti-patterns!) within composable approaches.



One key resource is the whitepaper "Understanding composable architectures: key concepts and patterns". These principles serve as the bedrock upon which a composable architecture is constructed, and they guide in making consistent and effective decisions throughout the development and operational lifecycle.

These principles are crafted to ensure seamless interaction, scalability, and resilience in an ever-evolving business landscape. One of the key learnings is to avoid direct integrations between systems to keep your eCommerce ecosystem flexible and scalable – and this whitepaper will go into further detail.

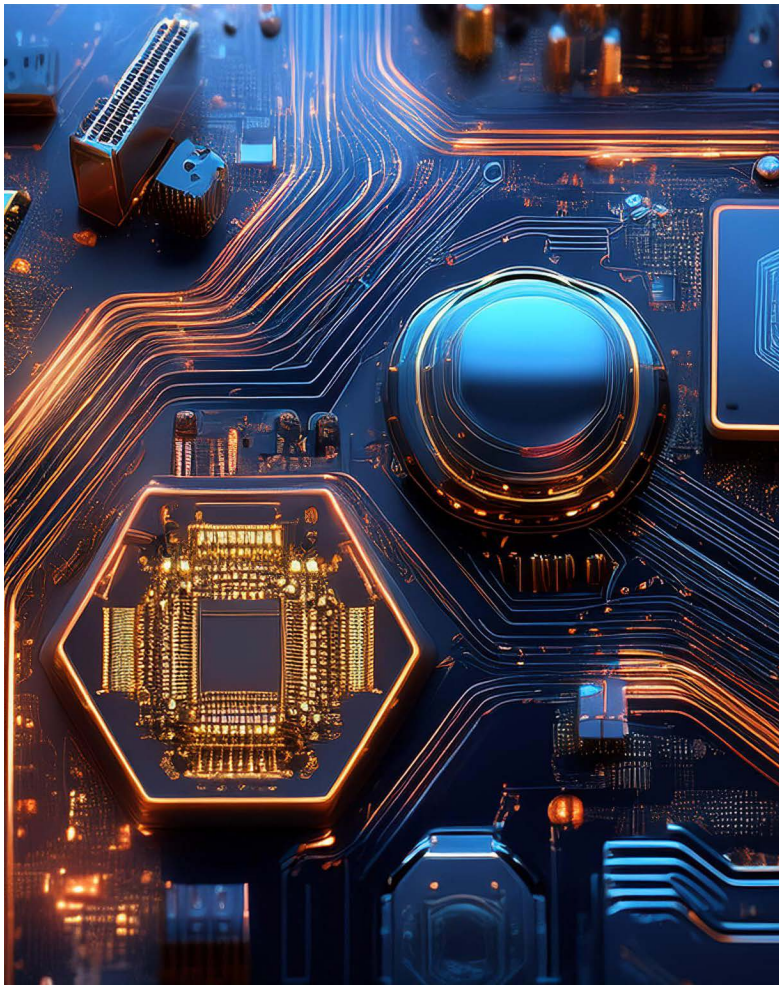
Integration Layers as a best practice for flexibility and sustainability

Horizontal integration layers with integration, federation, and data orchestration capabilities are key elements of composable architecture. These ensure that your application remains flexible and scalable across new touchpoints. We understand that these terms could feel abstract, so let's illustrate their importance from a common story that unfortunately has happened to many businesses.

Setting the scene

A retail goods company was building a new eCommerce landscape. They opted for composable architecture drawn to the promise of flexibility. They planned to start by rebuilding their existing web store and continue by launching mobile applications, and in-store point of sale (PoS) systems.

They selected ten best of breed vendors as many businesses do, including an eCommerce platform, CMS, search tools, and recommendation engines. But, to cut costs, they decided to bypass a horizontal integration and orchestration layer, believing it to be an unnecessary technical expense. Instead, they implemented point-to-point integrations directly between the vendors and the storefront.



What problems did they encounter?

Poor performance of the storefront

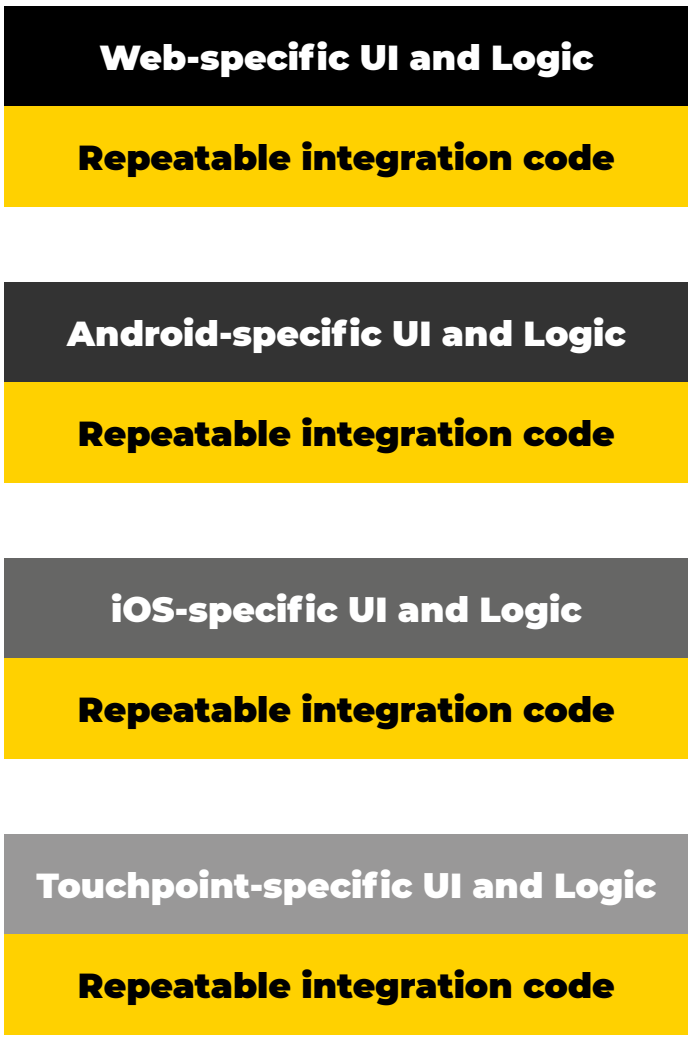
After the eCommerce storefront launch, they noticed the front-end application's performance was below expectations. The application took a few seconds to load and a few more to become fully interactive. This led to an increased bounce rate that negatively affected the company's revenue.

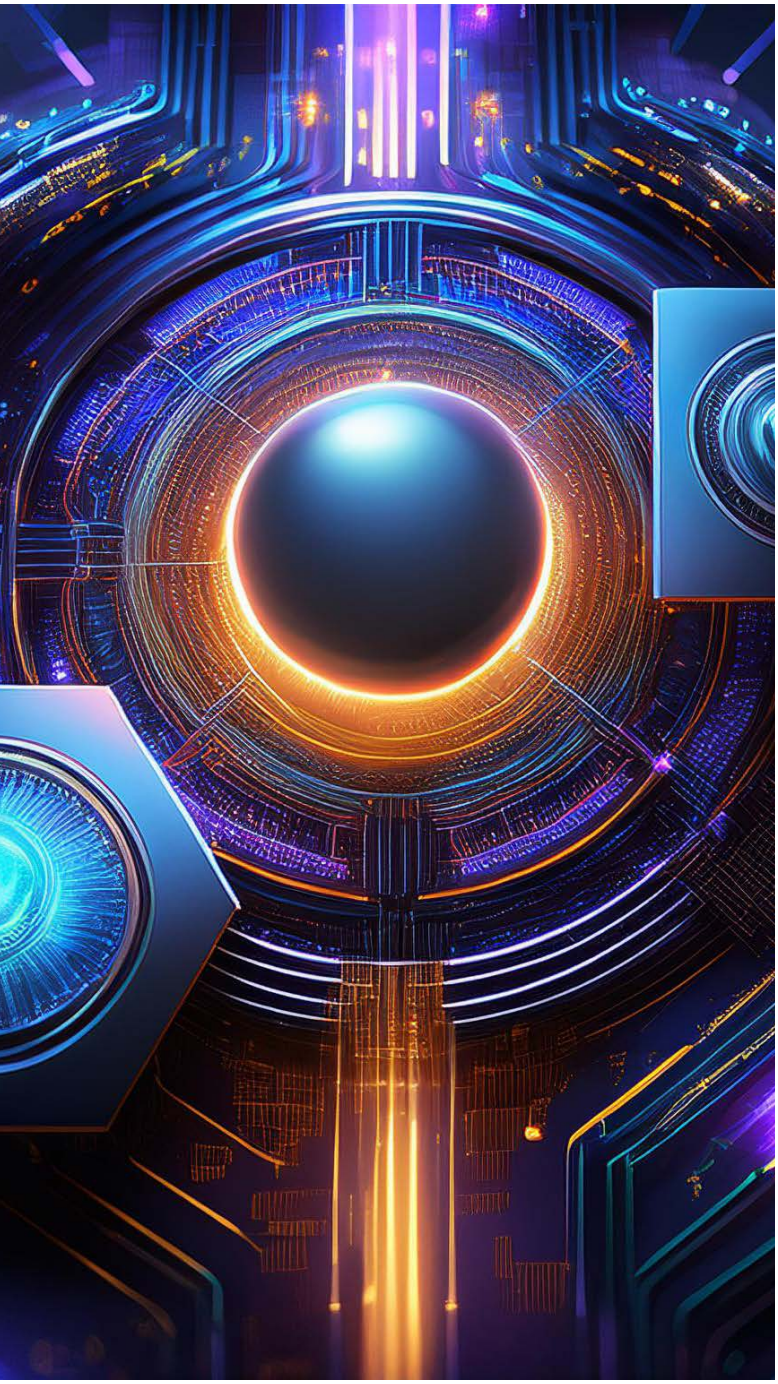
Read more about the impact of web performance optimization on business metrics.

Duplicated work and maintenance costs when scaling to new touchpoints

When they started working on their mobile application and kiosk touchpoints, they quickly realized that a lot of integration work and business logic was duplication of already implemented logic in the web application. The integration had to be repeated for iOS application, Android application, and in-store Kiosk application, instead of relying on a data orchestration layer to do those pieces of work for them.

As a result, time and cost of innovation – as well as maintenance costs – became much higher than they should have been. Also, scaling to another touchpoint would have linearly increased the effort to ship new features and maintenance work.





High costs of replacing technology

After some time, they realized that their choice of an eCommerce provider was wrong. These situations can happen when you navigate a new landscape of tech vendors and make multiple, strategic decisions at once. This is why it's essential to keep your technology vendors as independent from each other as possible.

Unfortunately, due to the direct coupling between different systems and the eCommerce platform, the company had to rewrite large portions of their code for each touchpoint. This resulted in not only significantly higher costs for their digital transformation, but also forced them into yet another time-consuming re-platforming effort, diverting focus from building true business value.

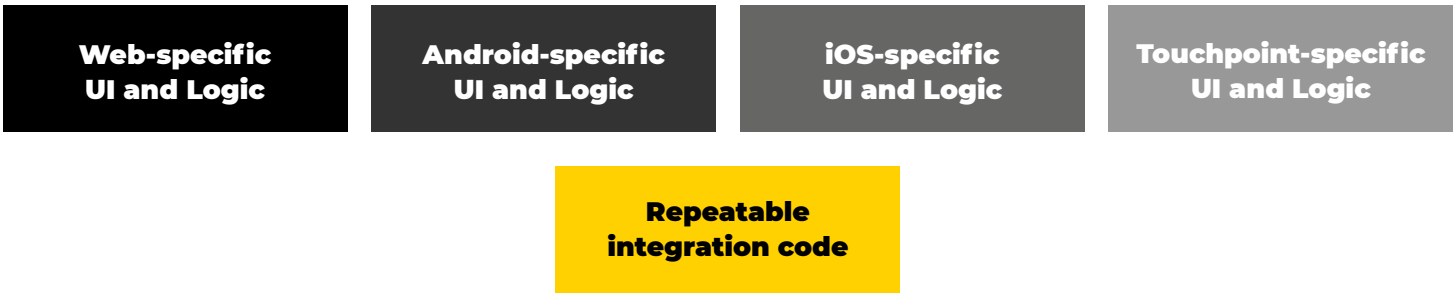
How could these problems be solved?

One of the primary goals of integration layers is to manage integrations between various systems in one place, with a structured unified integration approach. This not only eases the process of integrating new systems, it significantly reduces the amount of work needed to replace systems in the landscape, as changes generally need to be done in one place.

This capability becomes even more important for older systems, as technical debt and data flows between systems grow over time, making it harder to replace them in the future – especially if the systems have multiple integrations.

The separation of concerns that integration layers promote simplifies scaling to new touchpoints, markets, or use cases, and reduces the maintenance costs. The business logic that is not specifically dependent on a specific touchpoint can be shared between them.

MACH Alliance interoperability guidelines and reference architecture promotes decoupled systems using data integration and data orchestration layers. It simply avoids creating monoliths.



Last, but certainly not least, following the separation of concerns allows for building lightweight customer-facing applications faster, and they generally perform much better than their heavier equivalents.

Our recommendation for MACH projects

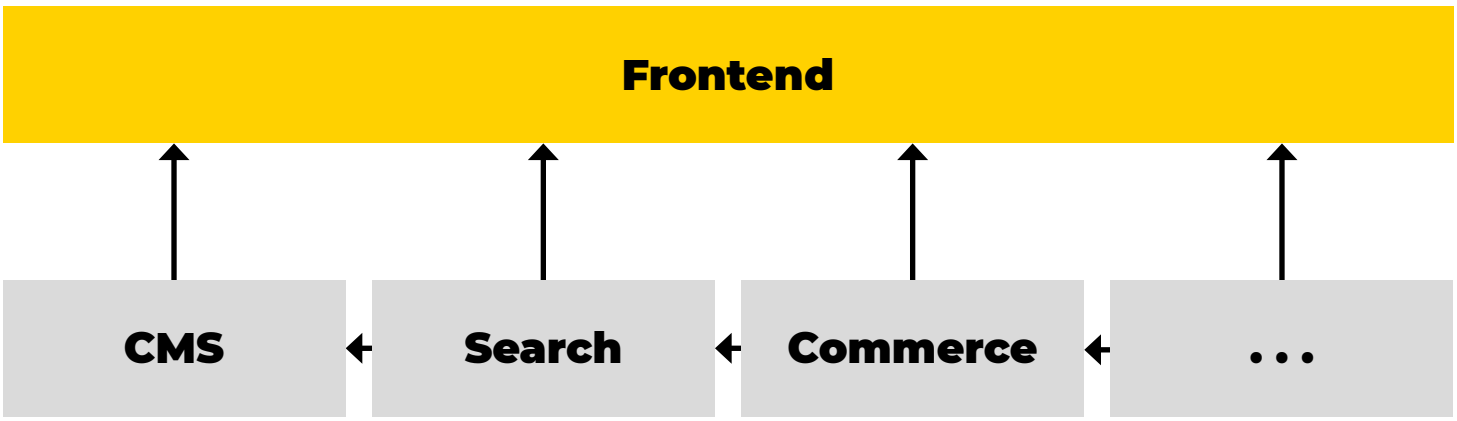
Don't

Integrate multiple systems directly to each other in multiple places. It might be appealing at first but leads to further issues with maintainability and flexibility of the ecosystem sooner rather than later.

Do

Ensure that your composable stack is flexible enough to accommodate future changes and can easily scale to new touchpoints, using a dedicated integration layer to manage connections with the services. Avoid pin-to-point integrations between different elements of your stack including the front-end (web, app etc.).

Use integration layers to abstract the underlying dependencies between systems and connect them in a structured way that ensures ease of future changes.

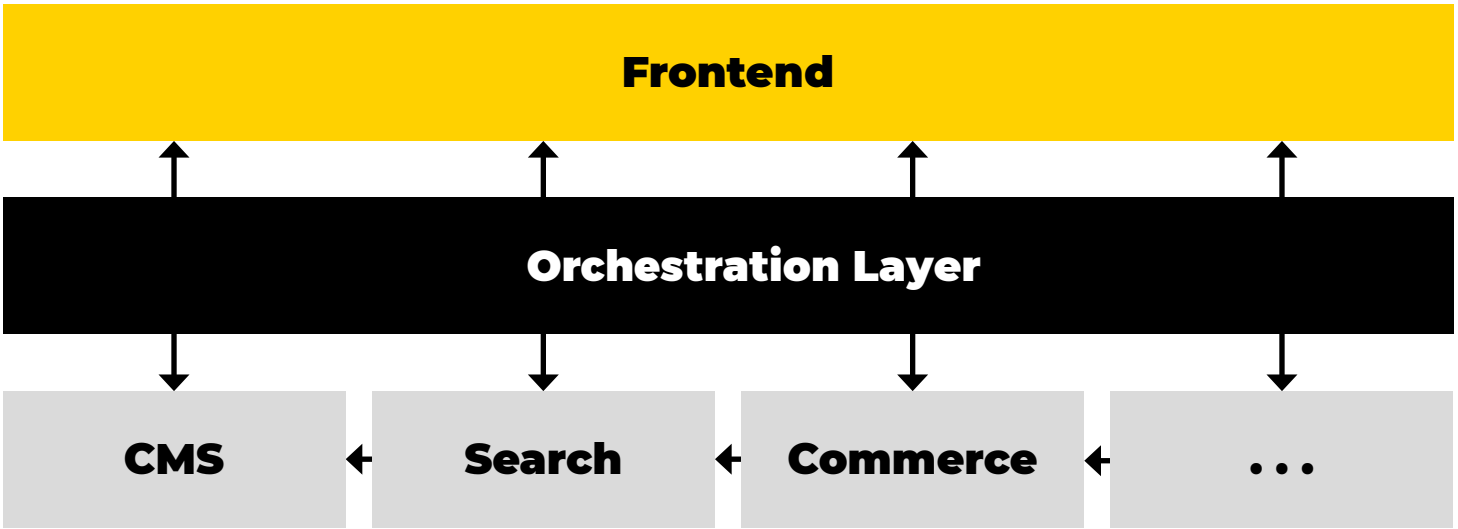


Such integration layers could be:

Data orchestration layer for API federation that simplifies access to data from multiple sources needed for a single context. For example, a Product Page that needs to resend the data from the eCommerce Platform, Content Management System, and Recommendations Engine.

Digital Experience Orchestration (DXO) that orchestrates interactions between systems as event-driven workflows. For example, when we need to keep the data synchronized between multiple systems or perform a sequence of real-time API calls where the next one is taking the previous one's result as it's input.

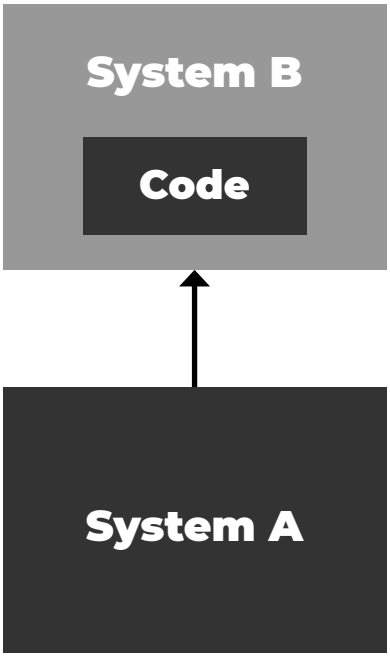
Integration Platform as a Service (IPaaS) typically for integrating legacy systems like Enterprise Resource Planning (ERP) tools and other slower systems of record to a modern composable architecture, or using pre-designed connectors to easily integrate standard SaaS services like CRM, or marketing software.



Our recommendation for MACH vendors

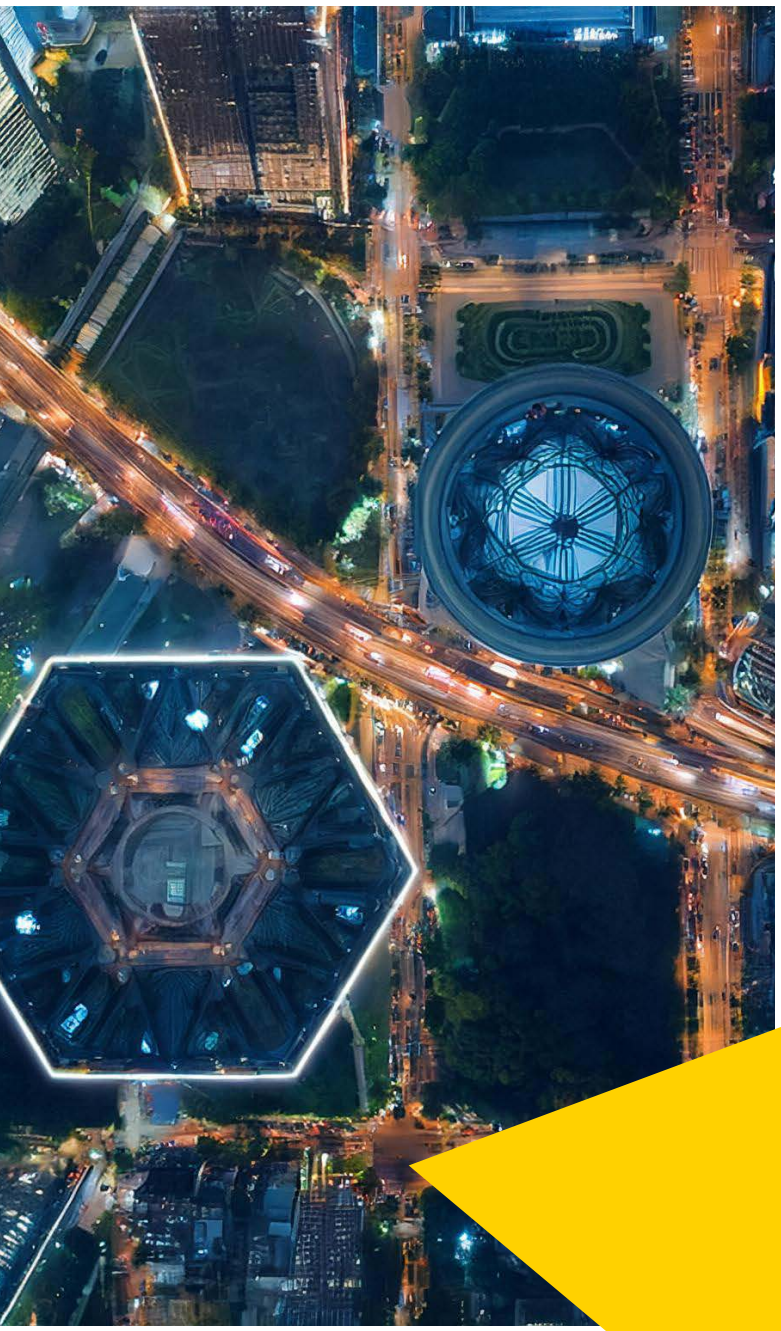
Ensuring that your solutions are flexible, scalable, and future-proof is critical for the long-term success of your customers. MACH and composable architecture promises, and delivers, the kind of agility businesses need. Only when the right approach to integration is selected can you deliver on the promise of significantly lower total cost of ownership (TCO) in years 2, 3 and beyond.

Avoid direct integrations between systems to keep your customer's system flexible and scalable. Relying on point-to-point connections as default creates a brittle system that is difficult to adapt, scale, and maintain.



Instead, encourage the adoption of integration layers with capabilities for integration, federation, and orchestration.

This helps to abstract the complexity of connecting multiple systems, reducing technical debt, and improving the overall agility of the ecosystem. Ultimately, keeping your clients happy and extending their subscription.



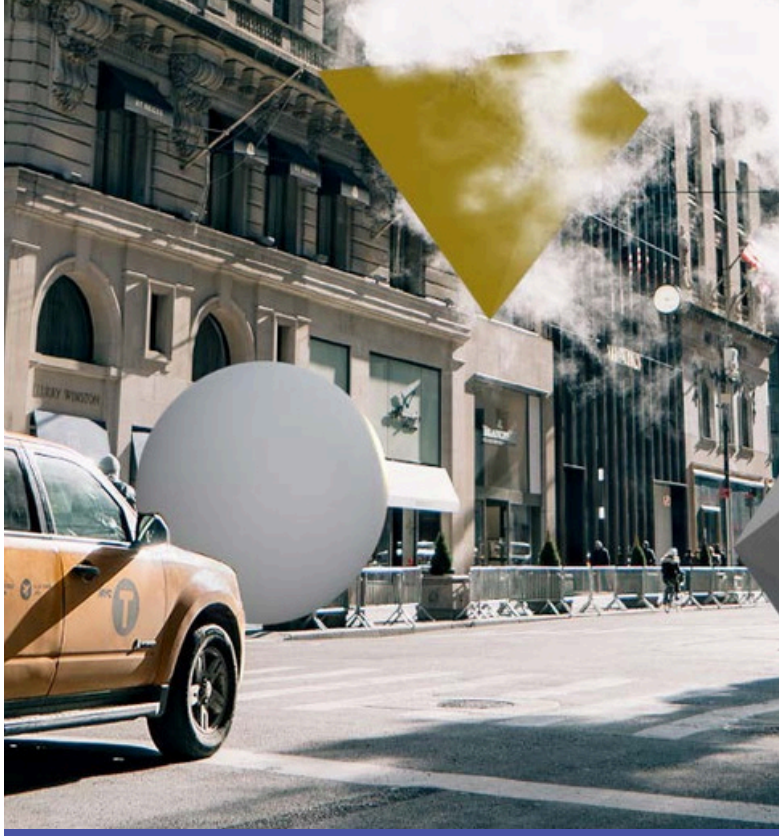
Conclusion

Our recommendation for businesses

- **Ask about** integration methods and be cautious about point-to-point integrations.
- **Request data** orchestration and data integration layers.
- **Pro tip:** Use the MACH Alliance reference architecture diagrams to validate the suggested solution designs.

MACH ALLIANCE

This eBook was brought to you by the MACH Alliance Interoperability Taskforce.



To learn more, visit the MACH Alliance Interoperability Hub

About The MACH Alliance

The MACH Alliance is an independent, not-for-profit 501(c)(6) industry body dedicated to advancing composable enterprise architecture. As AI transforms how organizations architect and scale digital experiences, architecture based on MACH principles has become foundational to delivering the agility, innovation, and future adaptability enterprises need to thrive. Guided by the five principles—Composable, Connected, Incremental, Open, and Autonomous—the MACH Alliance defines the framework for the next era of enterprise technology. Through member certification, expert community, and real-world guidance, the MACH Alliance supports organizations in confidently adopting open and transformative technologies to future-proof their business. The MACH Alliance represents over 100 member companies worldwide, including global enterprises, technology vendors, and system integrators. It welcomes enterprise practitioners to the community who share the same vision for the future.

Learn more at machalliance.org or LinkedIn.