uniform

From stagnation to innovation

Empowering digital teams with composable

The future is composable

Developers are the architects of applications that bring customer journeys to life, yet outdated legacy systems have thwarted the full potential of technical professionals, resulting in burdensome tasks, stifled innovation, and, ultimately, lackluster digital experiences.

Heavier workloads

Even though developers are responsible for coordinating, building, and integrating software components, custom integrations are labor-intensive and time-consuming, requiring APIs to resolve dependencies and intricate interactions. Buried in a plethora of IT tickets, the manual process involved drastically hinders workflows.

Loss of freedom and control

With a legacy platform, developers must rely on the vendor's roadmap or third-party services to integrate new functionality or build new features, lagging behind rapidly-changing trends and business needs.

More manual work, fewer value-add projects

Developers spend a load of time and effort adapting custom code, handling dependencies, and ensuring smooth interactions—so much so they are hard pressed to debug issues or optimize performance. As a result, productivity and innovation slow down.

What is a monolith?

During the early days of software development, single, massive content management systems (CMSes) called monoliths were the norm, handling everything from user interface to business logic and databases. As websites evolved from static brochures to more interactive experiences, CMSes followed suit with all-in-one solutions (now known as legacy systems) that featured robust functionality ranging from analytics to visual editors.

Despite their hefty price tag, monoliths offer the convenience of prepackaged capabilities from a single vendor—an appealing option in a crowded market. You deal with one contract and one vendor only for questions and technical support. Though their features might not be cutting-edge, traditional platforms offer a self-contained, predictable, and stable ecosystem, albeit only for the short term.

"[Monolithic] vendors have large, loyal agency partners and systems integrators that regularly convert their technologies into viable vendor offerings for deployment and adoption because those partners have abundant resources that are trained on the technologies required. That's a common problem for less mature buyers, however: As amazing as the vendor demos might look, the architecture has in reality long been outdated. Obviously, smart brands should avoid those aging platforms at all costs."

Lars Birkholm Petersen,
CEO and cofounder, Uniform

The failed promise of monoliths

Much like old engines that lack the horsepower to traverse modern roadways, monoliths are difficult to maintain and scale as they age. Unsurprisingly, brands that migrate from a monolithic architecture to a faster, more adaptable stack do so because of the following reasons:

- **Inflexibility.** Essential upgrades to resolve issues, such as security vulnerabilities, not only require comprehensive testing that can take months to complete, they're also costly. Furthermore, capabilities not offered by the vendor need third-party add-ons or new applications, which developers must build themselves.
- **Vendor lock-in.** As closed systems, monoliths contain features that are glued together and, thus, inseparable. Modernizing your infrastructure requires a "big bang" replatforming every few years, making adaptability a struggle.
- **Derailed innovation.** Because developers are limited by custom code, technical teams have few resources for experimenting with or implementing solutions through the latest innovations. Marketers might gain authoring tools that simplify editing and publishing tasks, but developers are held back by the vendor's proprietary technology and roadmap.

The cons of monoliths far outweigh the pros

What's more, monoliths are expensive. According to an Ecommerce Platforms Report, online merchants expect to spend up to $500,000 in replatforming costs during 2023. To capitalize on that, large vendors with vast resources and favorable reputations make aging monoliths appear viable to less savvy buyers. As impressive as their presentations and demos might be, the underlying architecture is often outdated and less than ideal for creating data-driven digital experiences. Forward-thinking companies want modern architectures that offer flexibility, longevity, and faster implementation to help achieve digital transformation in an environment where consumer demands are constantly changing. To accomplish that, switching from monolith to composable is the right approach.

Navigating the digital-experience architecture landscape

Driven by the demand for agility, scalability, and fast development cycles, omnichannel has ushered in a new era of digital-experience architectures. The COVID pandemic and market uncertainty have only accelerated those trends, spurring brands to increase spending on their digital properties.

Nevertheless, choosing the right approach for future-proofing a technology stack can be daunting. With the explosion of digital platforms, companies must navigate myriad vendor options and setups.

In addition to the legacy monolith, brands have the choice between one or a combination of the following approaches for building digital experiences:

Software suite

Not to be confused with legacy monoliths, suite vendors build or purchase products that are integrated into their core platform. Despite their modern capabilities, suite technologies are based on a monolithic architecture and tend to shy away from integrating outside their ecosystem. Examples are Adobe Experience Manager and recent versions of Sitecore offerings.

Headless

Headless tools, which apply to decoupled architectures, separate presentation (the head) from the underlying (headless) technologies.

MACH

MACH stands for **m**icroservices, **A**PI-first, **c**loud-based, and **h**eadless. Notably, products built the MACH way meet brand requirements for performance, scalability, and flexibility. However, not all headless vendors are MACH-certified, and not all architectures built with MACH technologies are composable.

Composable

Composable means that your digital-experience stack comprises packaged business capabilities (PBCs), each of which features one or more technologies that you can add or replace without changing the architecture. In a composable architecture, all systems are decoupled, offering maximum flexibility for switching functionality or updating designs.

Headless versus composable: What’s the difference?

Though relying on each other, headless and composable are not interchangeable. Here’s how they differ:

Headless

The agility and flexibility of composable architecture are enabled by decoupled or headless, API-first services: CMSes, commerce, digital asset management (DAM), personalization. Headless products store, manage, and deliver content with the front end separated from the content or presentation layer, i.e., the “head.” One caveat is that since headless products cannot display content in channels, you can create webpages but cannot build them with a headless CMS.

Composable

Composable architecture is a design pattern of reusable components, enabling you to easily assemble, modify, and reassemble the headless tools in your stack. As business needs change, you can plug in and swap out headless elements without replatforming. However, unlike the headless products purchased from vendors, you must build composable architecture yourself.

Combining headless and composable yields flexibility, incremental adoption, and a superior digital experience. Still, incorporating the right tools is just the first step. **You must also invest in content operations and system processes, and develop an organizational mindset.**

Composed doesn't equal composable

All too often, brands that move from legacy systems to a vendor-agnostic stack expect a composable architecture but end up with a composed solution instead. Some composed solutions come in the form of prepackaged capabilities—effectively all-in-one suites that have co-opted "composable" and "headless" terminology and practices.

However, composed can also comprise composable systems that work together through low-value glue code only. In that case, instead of offering flexibility and agility as composable does, composed silo teams, increase bottlenecks, and complicate workflows.

Conversely, with composable, organizations can select and combine functional modules from different platforms with minimal coding, facilitating the workflows of marketers and business users. Composable differs from composed in that businesses must balance the needs of business and technical teams to adopt new technologies so that marketers can create experiences without developer assistance. In return, developers are freed from the chore of integrating composable elements and can work with the front-end frameworks they prefer.

Beware of the composed MACH monolith

The composed MACH monolith is another hurdle businesses face when transitioning from monolith to composable. Though a composed MACH monolith uses MACH-certified technologies for CMSes, DAM, and other best-of-need solutions in a headless, API-first manner, those technologies are hardwired together with glue code. What started as one or two connected systems quickly becomes complex and unsustainable as integrations multiply. Developer-centric with little focus on the business-user experience, companies become stuck with an in-between version of the legacy-suite approach and the new composable way of designing architectures.

Why composable is the way forward

In the quest to balance team and customer expectations, more brands are leaving behind outdated systems for composable solutions that prioritize modularity, flexibility, and user empowerment. A global survey revealed that 85 percent of respondents from companies with 5,000 or more employees use microservices, a clear shift from single-vendor platforms to modular architectures.

And for good reason: Through composability, brands can break free of rigid solutions that waste money, time, and effort, switching instead to features that promote maximum agility and speed to market.

Key advantages

Flexibility	You choose the tools that best serve your needs.
Best-of-need approach	Instead of settling for a monolithic system's prepackaged capabilities, you adopt the solutions of your choice.
Experimentation	You test new tools, vendors, and frameworks in an environment in which it's easy and intuitive to integrate or update technologies.
Scalability	Your digital stack can grow as the business expands.
Multivendor environment	Free from the constraints of a single system, you—not the vendor—are in control of the tools that power your customer experience.

Common pitfalls to avoid

Despite their many benefits, composable solutions have inherent drawbacks that impact various teams.

- **Slow user adoption**. Though modularity is central to a composable approach, multiservice stacks can impede and complicate onboarding for new users.
- **Delayed sprints.** As marketing tickets mount, developers are inundated with tasks that deprive them of time to focus on enhancing site performance.
- **Clunky glue code.** Developers must write complex yet low-value code to connect composable systems to make them work together.
- **Stifled innovation.** Due to the challenges in updating the stack, developers are stuck with software that's been overtaken in the market and that doesn't support key features.

Digital experience composition delivers next-generation consumer experiences

To reap the full rewards of composable, brands need a digital experience composition platform (DXCP) featuring a visual workspace where teams can release high-performance experiences out of the box. With DXCP's low-code or no-code environment, developers are liberated from writing repetitive code and can then focus on value-add features—all while working with the frameworks they know best.

Running alongside headless systems, a DXCP's workspace combines composable, disparate technologies into a state-of-the-art and highly adaptable stack with three components:

Experience builder: Digital teams build and manage high-performance digital experiences with no-code tools without developer involvement.

API integration: Prebuilt connectors for most major composable services lend more flexibility and agility to technology stacks. Developers can focus on innovating instead of coding and maintaining integration logic for those services.

Front-end orchestration: Developers are free to work with the front-end components that best meet their needs. Digital teams can update and publish experiences faster without involving technical teams.

Benefits of a DXCP

Unlike monolithic systems, which can hold back the teams that use them, DXCPs offer developers notable benefits:

Agility

With composable multivendor tools, technical teams need not tackle publication tasks, and business teams can escape IT backlogs.

Compatibility

Business teams can quickly build digital experiences, and developers work faster in a composable environment with the tools of their choice.

Sustainability

Maintaining a long-term tech stack is easier because of the elimination of integration needs

Affordability

Adding, changing, and removing solutions in the team workflow is simple and straightforward without incurring the time, effort, and costs associated with replatforming.

Availability

Thanks to the elimination of integration needs and the availability of a low-code or no-code environment for business teams, technologies become more feasible for small and mid-size organizations that lack extensive technical resources.

With Uniform, you can—

- Orchestrate composable tools.
- Personalize digital experiences.
- Enable advanced API connections.
- Create omnichannel experiences.
- Run A/B tests in a low-code or no-code environment.

Learn more in a demo

About Uniform

Uniform DXCP is a versatile platform on which businesses can efficiently build high-performing and compelling digital experiences. By offering a consistent visual layer for content orchestration in any channel, Uniform DXCP gives business users, such as marketing and e-commerce teams, the ability to build experiences quickly and test ideas with no need for developer support. Also, prebuilt integrations eliminate the need for custom code to connect content sources, such as legacy DXPs, customer data platforms, and headless services. Businesses can then deliver projects faster and at lower cost while retaining long-term flexibility for innovation.

In addition, Uniform DXCP delivers content, including personalized content, directly to the edge of modern CDNs.

Customers that have adopted Uniform include Cobham Satcom, Sunweb, and Triumph.

Learn more at uniform.dev and follow us on LinkedIn and X.