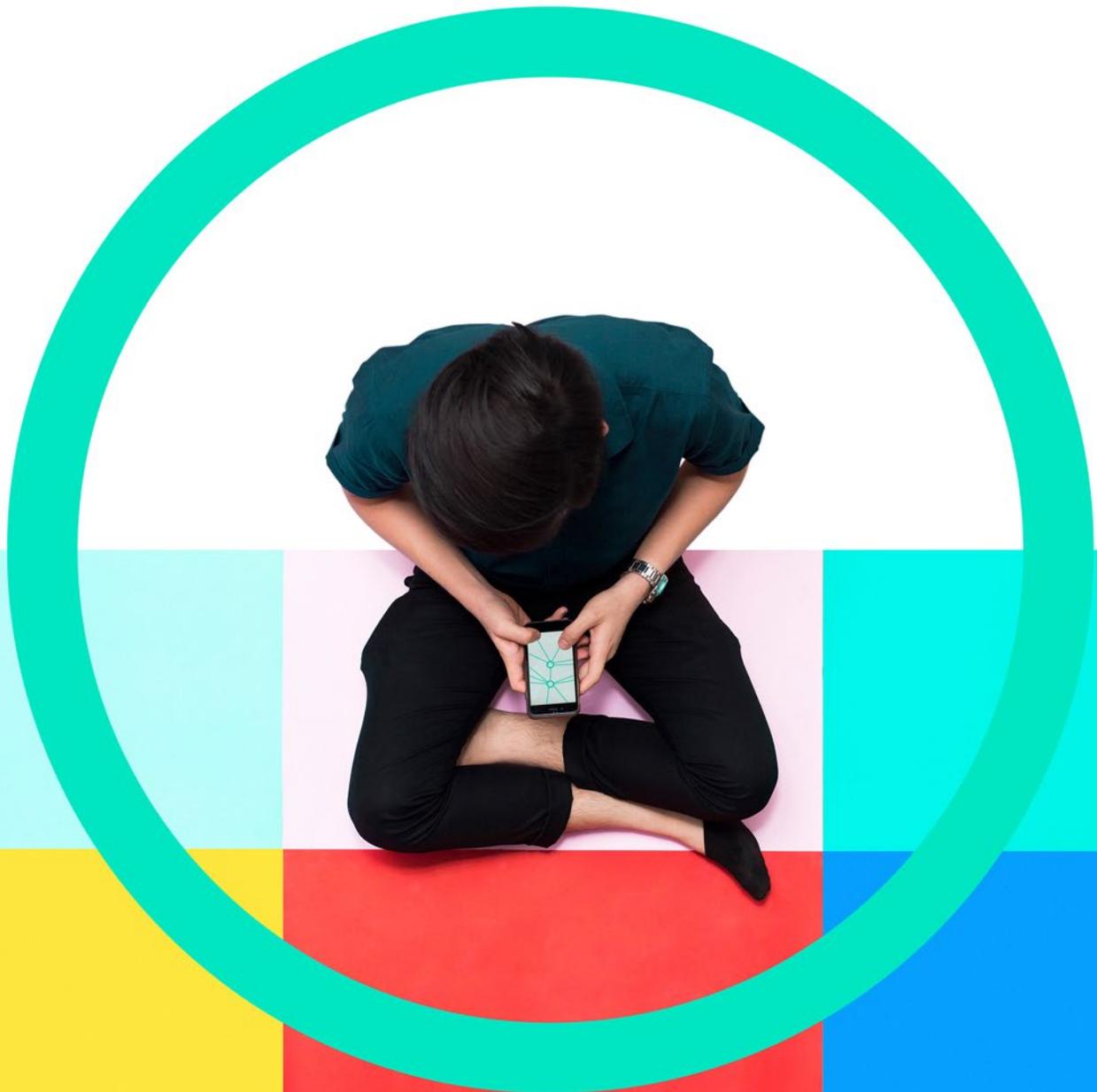


publicis
sapien

3 Proven Architecture Patterns for Integrating Digital Experience Platforms

AMIT XERXES & MURTHY PERI





Digital experience platforms

“Software to manage, deliver, and optimize digital experiences consistently across every phase of the customer life cycle.”

— FORRESTER RESEARCH¹

For information technology (IT) architects and technologists who seek to design, build, and evolve these digital experience platforms, a key challenge is to establish a platform architecture that integrates a variety of different, best-of-breed technologies and capabilities together to deliver an integrated solution. That same platform, however, must also enable a high degree of agility, scalability, and adaptability.

Based on years of experience architecting complex, omni-channel, digital experience platforms for global clients – working together with strategic partners such as Adobe in deploying their Adobe Experience Manager (AEM) solution for different customers and enterprises – we’ve observed different business drivers and architectural patterns.

Here, we collate our thinking and point of view on the most common business drivers that we see, as well as our recommendations on which architecture patterns are most appropriate for different customer scenarios.

The business justification

Today’s consumer doesn’t distinguish between channels and moments of engagement, transaction, and service. A modernized IT platform that supports digital experiences has to allow a seamless journey from awareness to engagement and discovery, purchase and transactions and even customer service and support. Unfortunately, that expectation is not easy to live up to for most enterprises today.

Many enterprises and their IT platforms have traditionally been organized by “functions” (e.g., Marketing, eBusiness, and Customer Support), thereby leading to unwanted silos. To deliver a unified, integrated consumer experience, many enterprises have started to bridge this gap by adopting and embracing the concept of a unified digital experience platform (DEP) that cuts across organizational silos (see Figure 1).

¹Forrester. Vendor Landscape: Digital Experience Platforms. <https://www.forrester.com/report/Vendor+Landscape+Digital+Experience+Platforms/-/E-RES104302?objectid=RES104302>.

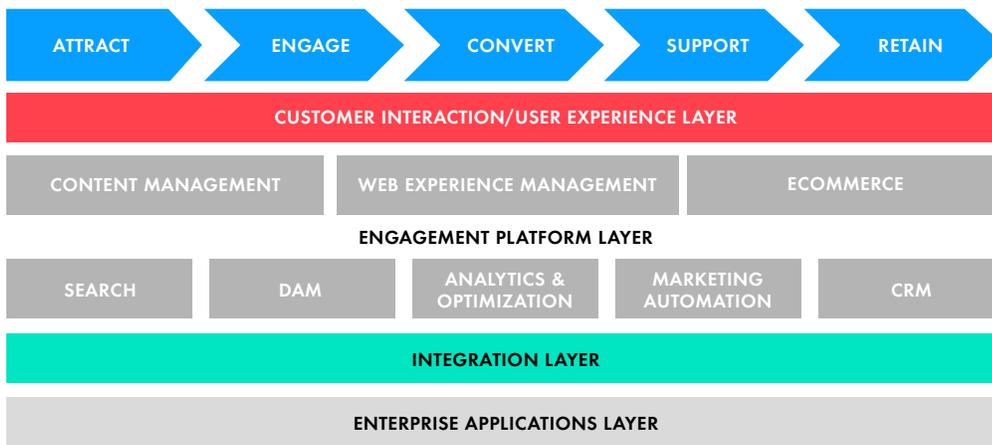


FIGURE 1
Components of a digital experience platform

A digital experience platform delivers a unified, seamless customer experience by bridging the gaps between various technological layers (and organizational silos).

A FEW KEY CONSIDERATIONS FOR DIGITAL EXPERIENCE PLATFORMS

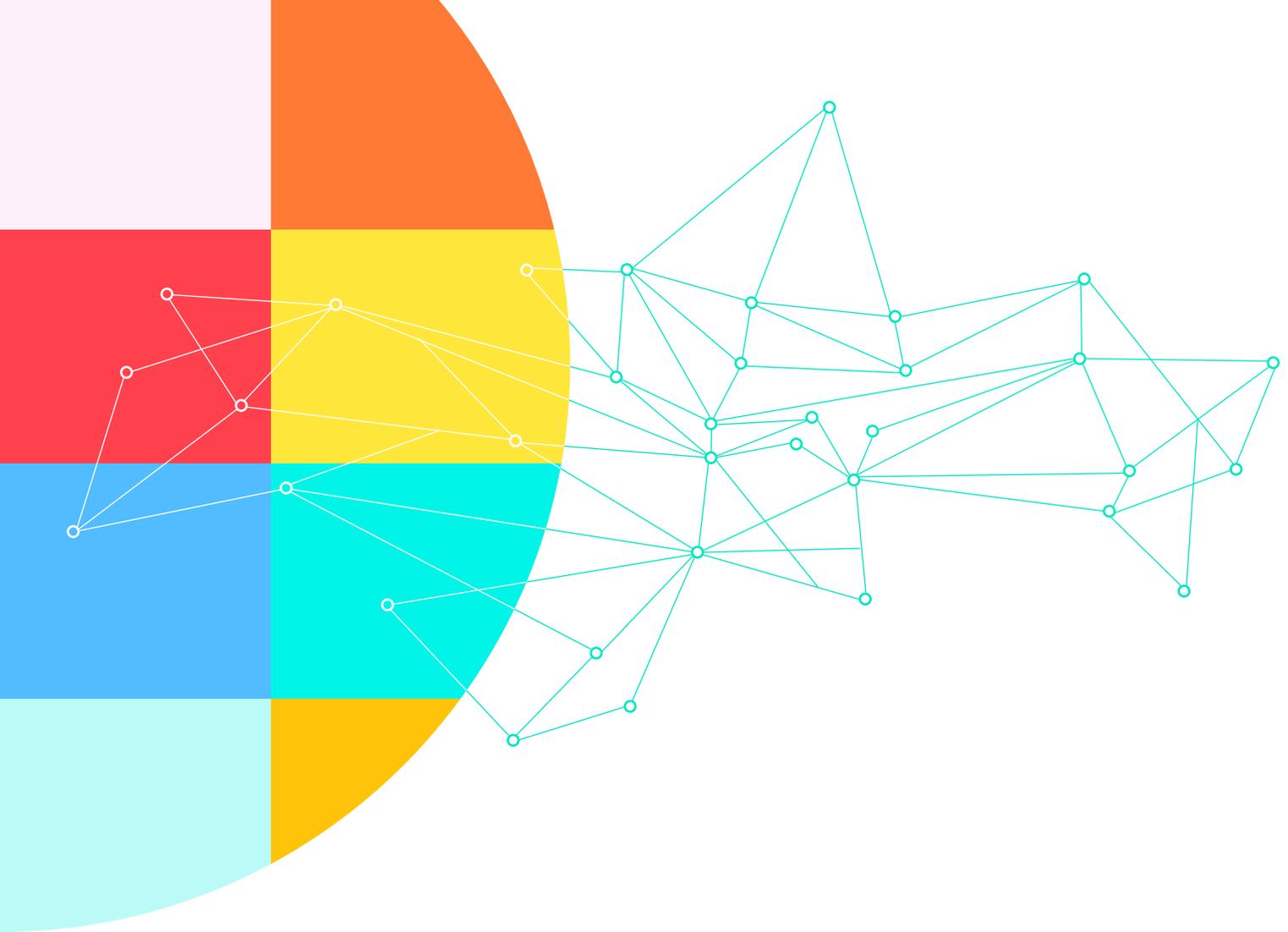
- The consumer journey is typically the key construct against which a digital experience platform is built.
- Everything within the DEP must come together to support an integrated, seamless, cross-channel consumer journey.
- Each layer of the platform is designed to solve for a particular need – e.g., customer interaction vs. back-end orchestration.
- Each layer has a variety of capabilities and solution components deployed.
- For the DEPs in certain organizations, the technology landscape within and across each layer may be heterogeneous (often evolving organically over time).

A SCENARIO-DRIVEN APPROACH TO BUILDING DIGITAL EXPERIENCE PLATFORMS

Having worked with hundreds of marquee clients over the last 20 years, we’ve seen a broad array of business drivers and use cases for why a client may want to invest in a digital experience platform like the AEM solution. Correspondingly, we’ve also explored, defined, and implemented a variety of different solution patterns for how a DEP can be constructed and brought to life. Based on our learnings across all of these, it is evident to us that every DEP solution within each organization is somewhat unique – there is really no “single solution” or “silver bullet” for how these platforms can (or should) be architected.

Against this backdrop, our experiences have shown that there are four key business scenarios and use cases for a modernized DEP:

SCENARIO	USE CASES
Brand Marketing Experiences	Brand Marketing Sites, One-off Campaign Sites, Microsites, Corporate Websites
Multi-Brand Multi-Site Digital Marketing Platform	Multi-Channel, Multi-Brand Marketing Multi-Region Brand Marketing Sites Global and Local Brand Marketing Coordination Multi-Agency Coordination Workflows
Experience-Led Business and Transaction Platform (Light Orchestration)	Integrated Browse and Shop Experience Product Catalog Merchandising Product and Category Marketing Content-Driven Commerce
Omni-Channel Consumer Experience at Web Scale (Heavy Orchestration)	Integrated Browse, Shop, Purchase and Customer Service Cross-Channel B2C/B2B Commerce Multi-Brand, Multi-Store Product Catalog with High SKU Count High-Volume Order/Transaction Processing



The architectures

There are various ways in which a digital platform and its underlying solution architecture can be defined. Consequently, there are a number of different design patterns that can be applied to architect the solution as well. These patterns dictate how responsibilities are assigned across the various layers of the architecture, how systems, technologies, and packages are “mapped” to these responsibilities, and how the interactions between the different layers and systems work to deliver the complete solution. Again, as opposed to advocating the “one-size-fits-all” solution pattern that works for

every scenario, we believe in considering a catalog of solution patterns that can then be evaluated and applied against each enterprise’s needs and unique situation.

For the purposes of this paper, we will focus primarily on the Adobe Experience Manager as the web content and experience management solution. We will highlight a few key architecture patterns for building digital experience platforms around the AEM solution and reveal how AEM can be deployed and extended in different ways to support various needs.

We believe in considering a catalog of solution patterns that can then be evaluated and applied against each enterprise’s needs and unique situation when determining the pattern that fits best.

PATTERN Standard AEM solution deployment

The deployment of a standard AEM solution is ideal for a wide spectrum of brand marketing experiences, including corporate websites and microsites, global and local brand coordination, and multi-brand/channel/regional marketing. With this pattern, IT teams separate concerns and responsibilities by decoupling front-end activities from those of the back end (see Figure 2). This enables both to evolve independently – not to mention allowing front and back-end teams to do what they do best.

Even with a decoupled front end, the application of this pattern retains all the power of the authoring and marketer capabilities available out-of-the-box with the AEM solution. The benefit is given by the seamless integration of the front-end experience and modules with back-end AEM platform components, which (by leveraging a “schema” or contract-driven component development model) minimizes churn and rework in the integration process.

By adopting a front-end view library that supports a “write once, run anywhere” approach, brands are able to rapidly evolve their experiences from largely static ones to interactive, personalized, and dynamic interactions. This, combined with the use of the out-of-the-box (server-side) templating construct available in AEM, enables a multi-tenant architecture that allows multiple brands to derive reuse from a shared platform – yet grants each of them and their corresponding agencies full creative independence in how they design and develop the customer experience.

The standard AEM solution also takes advantage of a “progressive rendering” model that enables a good portion (or even the entirety) of the experience or page to be rendered by the AEM, all while reusing the same front-end view library for rendering across both the server and browser containers. The AEM solution (in this case, AEM Publish) is then used as the primary deployment container and “presentation assembly” layer to host, deliver, and serve the consumer experience. Subsequent integration needs associated with the organization’s marketing ecosystem (e.g., analytics, targeting, digital asset management, search, and consumer data) can be largely handled within the AEM platform itself or done client-side (i.e., browser-side).

FIGURE 2
Standard AEM solution deployment

By visualizing the various components of a standard AEM solution, we can see how the roles and responsibilities are separated by the decoupling of the front- and back-end systems.

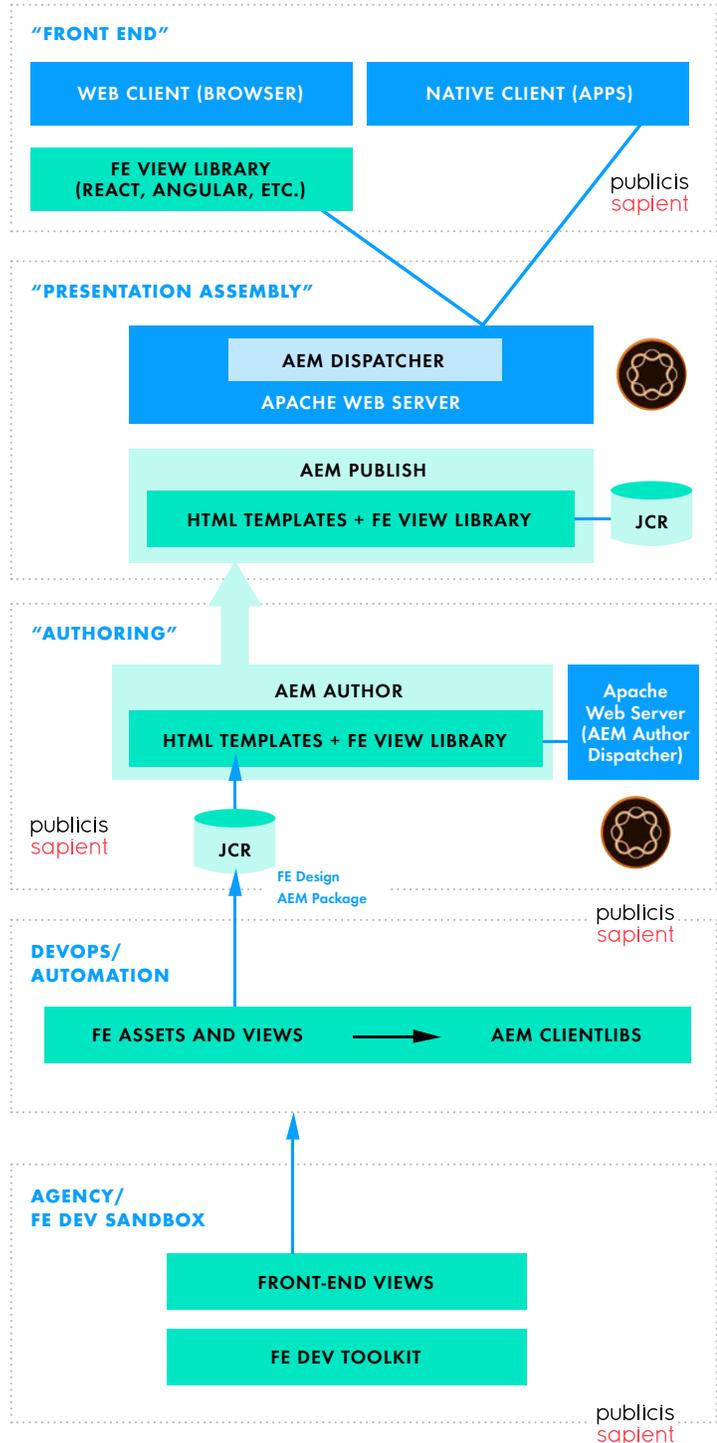
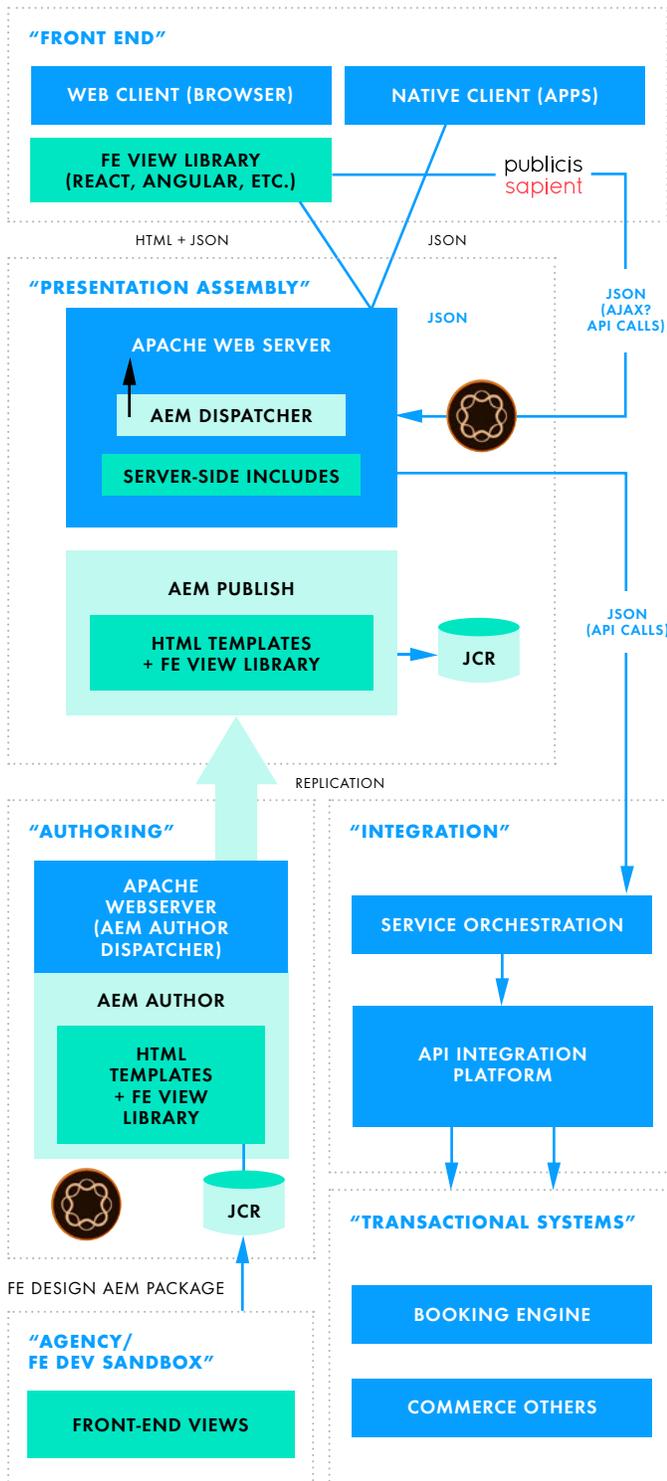


FIGURE 3

Experience-led business solution

In this pattern, we can see how an API-first mindset enables the connection between content capabilities and those of transactional systems.



PATTERN Experience-led business solution

When integrating browsing and shopping experiences or driving content-driven commerce, adopting an experience-driven pattern is preferred. In this deployment, there are three factors at play: the front-end team defining templates and designing user-facing elements, the AEM platform team creating editable content components, and the integration developers building scalable integration services and application programming interfaces (APIs). Together, they combine marketing content and editorial capabilities with the data and business applications of underlying transactional systems to enable a unified consumer experience characterized by integrated experience management (the authoring) and delivery (the rendering).

In this solution, as in the previous one, the consumer experience and back-end systems are again decoupled and evolve independently. The AEM solution retains the power of its out-of-the-box authoring and marketer capabilities as well – this time to enable integrated in-context authoring, editorial tasks, and experience previews within the AEM environment itself. The front-end view library is also carried forward to enable front-end developers to build and manage user-facing elements separately – thereby facilitating greater experience agility.

Lastly, in most scenarios, the solution continues to leverage the AEM Publish Server as the primary “presentation assembly and composition” layer to render and deliver the customer experience. The overall experience, page layout, and marketing components can all be rendered by AEM Publish, however, dynamic components that require integration and orchestration (e.g., product details, booking widgets, etc.) can be resolved and rendered by using either server-side include technologies.

The key to applying this pattern is an “API-first” mindset (see Figure 3). Based on this, a scalable service orchestration layer within the architecture is established to integrate with back-end systems, including commerce, customer relationship management (CRM), booking engines, etc. This is particularly relevant as the number of integrations increase and greater orchestration is required across services (back-end systems) to deliver and render the customer experience.



PATTERN Microservices architecture

Our last pattern is a game changer when compared to your standard AEM solution, CMS platform, or any other content and experience platform deployment. This is due to three critical, differentiating factors:

ENABLE EXTREME SCALE FOR BUSINESS-CRITICAL TRANSACTIONS

The scalability and agility of this pattern stretches far beyond the simple delivery of content and marketing experiences to enable business transactions, digital commerce, and online customer conversion at incomparable scale.

BUILD THE ENTIRE SYSTEM FOR CHANGE

Our third pattern can rapidly – and independently – evolve both your cross-channel consumer experience and its underlying integrations/transactional systems without needing to “rip and replace” with every major change.

UNITE THE CMO AND CIO IN COLLABORATION

The ultimate goal of this pattern is to empower marketers with the toolsets they need to create and deliver great experiences, while simultaneously gifting IT with an architecture pattern that allows them to not only rapidly scale, manage, and maintain the solution easily but also to become the CMO’s best friend by supporting their (and the architecture’s) need to evolve.

Based on microservices, which are typically the core of the digital experience platform, this pattern is ideal for the heavy orchestration involved in architecting omni-channel consumer experiences at web scale. These applications include crosschannel business-to-consumer or business-

to-business commerce, multi-brand or multi-store product catalogs with high counts, and high-volume order or transaction processing. This third pattern is also preferred for integrating across browsing, shopping, purchasing, and customer service interactions.

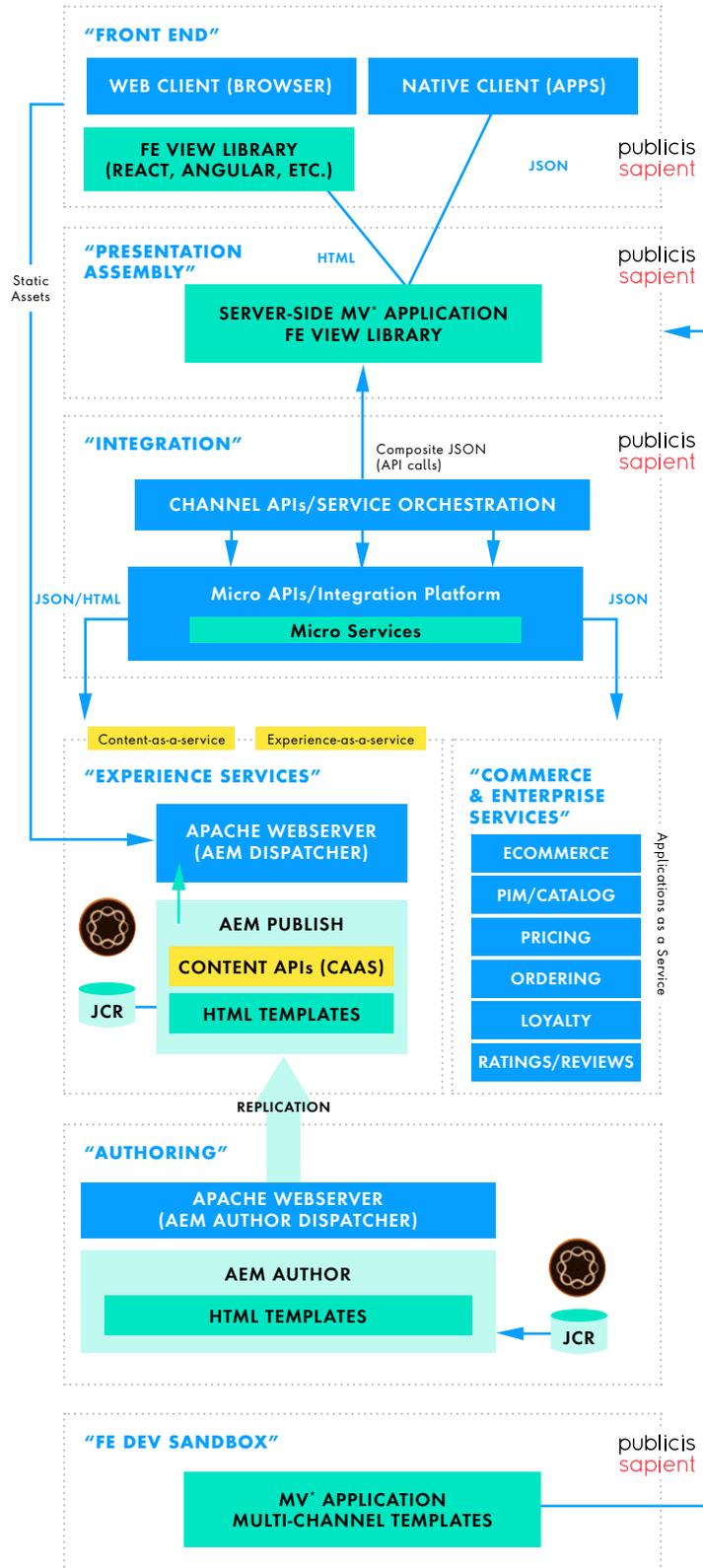
All of the aforementioned examples necessitate a high degree of modularity and agility – along with the ability to manage, deploy, and upscale each layer independently. A cloud-native, pure microservices architecture pattern solves for this by adopting an everything-as-a-service model that identifies multiple interactions (e.g., content consumption, commerce experience management, ordering, product catalogs, promotions, loyalty) as “headless” services (see Figure 4). The headless approach is what allows for a high degree of decoupling and flexibility across the consumer experience. It supports a variety of evolving channels, as well as offers a platform architecture future-proofed for the evolving technology landscape – one in which individual services and their underlying products are likely to change over time.

Similar to the experience-driven pattern, the microservices architecture also leverages an API-first model when interfacing with all underlying systems and services. In fact, this is what establishes the highly scalable, non-blocking service orchestration and integration layer within the architecture. This layer is separated from the actual microservices. The service orchestration layer is highly optimized for a channel (typically delivering a single, composite, and optimized response for each one), but that channel can spread across several underlying microservices. It is the architecture that combines the responses from several underlying API calls into one unified response: the customer experience.

FIGURE 4

Microservices architecture

This pattern is supported by a “headless” services mindset that caters to strong decoupling and flexibility across various underlying technology components of the customer experience.



That experience (the overall application) is supported by a set of loosely-coupled, fairly independent modules that can follow their own development lifecycles – also referred to as “non-monolithic” deployment. It has a preference for a “decoupled presentation assembly” layer (a.k.a. a “decoupled glass”) in which a server-side MV* (Model-View-Wildcard) application enables:²

- Routing
- Server-side presentation logic (controllers)
- Orchestration across APIs and back-end services
- Dynamic experience composition
- Asynchronous processing and parallel execution
- Event-driven and reactive programming models

These activities are typically done outside of the underlying systems (e.g., content, commerce, etc.) via a “Universal JavaScript” application pattern for the assembly, rendering, and presentation layer –wherein the same MV* application can assemble and render the dynamic experience across server and client. The application is typically deployed as the “decoupled glass” that simultaneously enables high-speed, high-performance rendering on the server, front-end rendering in the browser, and rapid updates to the customer experience layer.

With this, the consumer engagement platform is able to rapidly update with faster release cycles and more frequent deployments, all while isolated from the underlying mission-critical transaction system (which may follow its own release and update schedule). It is this notion of “bi-modal” or “two-speed IT” that enables the right balance between experimentation and agility vs. stability and scalability.

Lastly, experience management needs are relatively simpler than what is typically seen with the experience-driven solution. Some degree of layout, page, and template management is still desirable and required, but this is either limited to specific sections of the customer experience (e.g., product discovery) or is focused on specific needs such as component/slot placement (primarily for the purposes of merchandising).

That being said, IT leaders have the opportunity to share, leverage, and adapt the underlying AEM solution within the enterprise across multiple solution patterns. For example, the experience-driven and headless patterns (second and third, respectively) can be combined using the same AEM infrastructure – a decision that makes sense if the enterprise has already invested in one Adobe solution and now wishes to leverage the unique benefits of another.

²MV* refers to any serverside technology framework that references one of many model-view design patterns, such as MVVM (model-view-view model), MVP (model-view-presenter), or MVC (model-view-controller).

The benefits

Organizations and enterprises worldwide are betting on digital experience platforms as the new innovation frontier and primary customer engagement vehicles. Many are taking a look at their business needs and scenarios and considering how the implementation of an optimal DEP solution could benefit them in both the short- and long-term.³ Given that each organization has their own unique business goals and drivers, these inputs are vital when determining how best to architect that solution (see Figure 5).

The business scenarios and solution patterns outlined in this paper are meant to guide the

process, helping IT architects and application development teams determine which architecture makes the most sense for any given organizational situation. This catalog of patterns should enable architects, consultants, and client IT teams to collaborate jointly and have conversations on which approach is most applicable and beneficial to them.

Of course, these patterns can be further extended, modified, and tweaked as necessary. In fact, leading companies are taking steps to plan for these types of flexible digital experience platforms and then rapidly implement them with partners to achieve business goals.

FIGURE 5

Comparative analysis of three solution architecture patterns

Based on both business needs and technological wants, our three patterns can be analyzed against the following points to determine which DEP architecture is the right one for the organization at hand.

	PATTERN 1 Standard AEM solution deployment	PATTERN 2 Experience-led business solution	PATTERN 3 Microservices architecture
Separation of layers/responsibilities	Low	Medium	High
Decoupled presentation/front-end	Loosely coupled	Moderately decoupled	Fully decoupled
Isomorphic rendering support	Low	Medium	High
Performance viewpoint: Scalability and performance of content-heavy, marketing applications	High	Medium	Medium
Performance viewpoint: Scalability for dynamic, transactional applications	Low	Medium	High
Performance viewpoint: Scalability for dynamic rendering throughout of non-cached pages and apps	Low	Medium	High
Overall degree of decoupling across layers and systems	Low	Medium	High
Modularity, flexibility, and adaptability of the solution architecture	Low	Medium	High
Leverage of authoring toolset and experience management capabilities	High	High	Low
Deployment viewpoint: Flexibility, speed of deployment, and ability to make rapid changes in Production	Low	Medium	High
Deployment viewpoint: Complexity of overall deployment architecture, number of tiers to be managed, etc.	Low	Medium	High
Operations complexity: Maintenance, management, and upgrades	Low	Medium	Medium
Primary benefits	Simple deployment model; scales well for content heavy app	Separation of integration tier behind API gateway; power to marketers for experience management	Highly scalable and extensible architecture for omni-channel delivery; high degree of agility and flexibility; "cloud-native" architecture
Primary limitations	Does not scale well for transactional apps, highly dynamic interactions and heavy integrations	Overall system performance and scalability is still limited to a certain degree by rendering throughout of Experience Manager	Inherent complexity in managing a distributed architecture of this nature, necessitating strong use of mature DevOps, Automation and Cloud-Native design patterns

³Publicis Sapient, The Rise of Digital Experience Platforms. <http://www.publicissapient.com>.

Authors



AMIT XERXES

Senior Director Engineering
Publicis Sapient India

Amit Xerxes is a Senior Director Engineering at Publicis Sapient, and is based out of our offices in Noida, India. He has 18+ years of experience in Marketing Platforms and various capabilities within Marketing Platforms spanning Web Experience Management, Content Management, Digital Asset Management, Portals, Digital Experience Platforms, and Global Digital Marketing Platforms. During this tenure, Amit has played multiple roles across different teams and initiatives, ranging from architecting marketing platform solutions for large-scale complex digital platforms for our global clients, supporting package evaluation and selection exercises for platforms, and leading Engineering teams focused on creating reference architectures, frameworks, blueprints, and solution accelerators to enable innovation, quality and efficiency in solution delivery. Amit has experience with a variety of different commercial and open-source products and packages over the years. Amit was the technology lead for a major engineering initiative at Publicis Sapient responsible for evolving our capabilities around Architectures and Decoupled and Headless Experience Platform solutions. In his current role, he is leading our Cognitive Marketing engineering program focused on the application of Data Sciences and Machine Learning to Content and Marketing Platforms, enabling the delivery of hyper-personalized experiences by leveraging consumer intelligence.



MURTHY PERI

Group Vice President Engineering
Head of Marketing Platforms

As a head of marketing platforms, Murthy focuses on engagements advising clients on their digital transformation enabled by technology and marketing cloud platforms. Murthy is passionate about realizing the value of capabilities enabled within enterprises and doing so with engineering excellence. Murthy brings 20 years of experience in deploying complex digital solutions for Fortune 500 organizations. He's well versed and a subject matter expert in marketing technology solutions spanning campaign, experience, assets and personalization components. Some of the clients include Corteva, Samsung, Invesco, SSGA, Allianz, Analog Devices, P&G, and GiantEagle. Prior to Publicis Sapient, Murthy served as a Vice President of Technology at Digitas, where he helped launch an enterprise strength mobile platform for Aflac field force. On P&G, Murthy designed a global digital platform to launch major brand websites including Tide, Downy, Pantene, Eukanuba and Head & Shoulders. Before that, Murthy spent 10 years at Agency.com as a technical director.



GET IN TOUCH

To learn more about how we partner with Adobe, contact Brent Cowan brent.cowan@publicissapient.com or Drew Kurth drew.kurth@publicissapient.com



Publicis Sapient is a digital transformation partner helping established organizations get to their future, digitally-enabled state, both in the way they work and the way they serve their customers. We help unlock value through a start-up mindset and modern methods, fusing strategy, consulting and customer experience with agile engineering and problem-solving creativity. As digital pioneers with 16,000 people and 53 offices around the globe, our experience spanning technology, data sciences, consulting and customer obsession – combined with our culture of curiosity and relentlessness – enables us to accelerate our clients' businesses through designing the products and services their customers truly value. Publicis Sapient is the digital business transformation hub of Publicis Groupe. For more information, visit publicissapient.com.