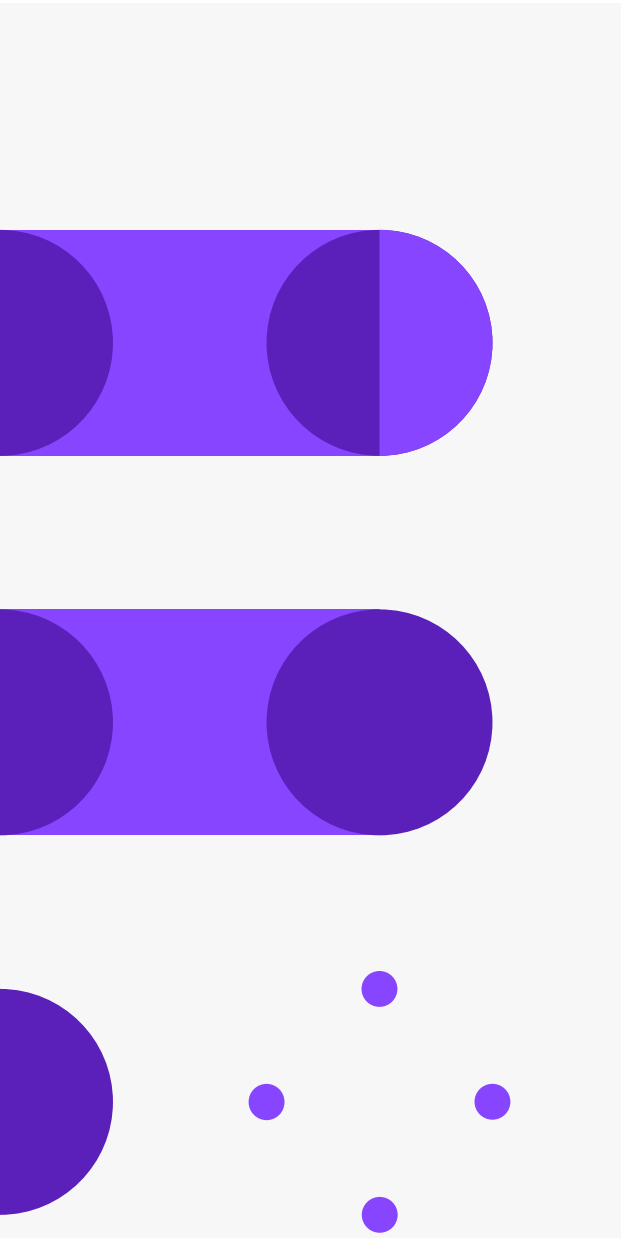
CONTENTSTACK™



Break the Replatform Cycle with MACH Enterprise Architecture

A practical guide to implementing cloud, API, and microservices technologies for a modern enterprise experience.

Table of Contents

The pace of technology adoption is staggering.

Thirteen years after the introduction of the iPhone, over three-quarters of people in advanced economies own a smartphone [1]. Six years ago the Amazon Echo made its debut, now over a quarter of US households have a smart speaker [2]. Customers are becoming digitally fluent at breakneck speed and, to speak to these customers, companies are expected to master new technologies as quickly as they come to market.

It's like the Tour de France switched to the Grand Prix in the middle of the race, and everyone now has to turn their bike into a car without slowing down.

Companies have to embrace new technologies to even stay in the middle of the pack. Pulling ahead requires more than adopting a few one-off tools, but transitioning to a model that supports continued, rapid evolution of digital business.

For many enterprise organizations, this transition is frustrated by a complicated network of outdated legacy tools that make it hard enough to deliver a status quo experience — let alone an innovative one.

"In terms of ability to innovate, everything had a cost associated with it, which put us off doing anything risky. That meant our technology was stagnating — and so was our ability to serve customers," says Tom Morgan, Director of Digital at *The Spectator*, explaining why the British media company decided to move off legacy platforms to a modern architecture.

Modernizing an enterprise architecture is not only a chance to use the current market leading software, but an opportunity to make future evolution a lot less painful.

MACH (microservices, API-first, cloud-native, headless) is a set of design principles behind a rising number of best-of-breed software solutions. Built to integrate easily, these technologies are helping companies like *The Spectato*r untangle and step off their legacy tools.

With a modular design, MACH solutions give businesses the freedom to choose from the best tools on the market today and provide a structure that makes it easy to add, replace, or remove technologies in the future. This composable architecture means that enterprises are essentially facing their last major replatforming effort and, instead, can continuously evolve their ecosystem piece by piece.

Because in the digital experience race, as soon as you finish turning your bike into a racecar it will be time to start evolving it into a rocket.

MACH technologies support a composable enterprise in which every component is pluggable, scalable, replaceable, and can be continuously improved through agile development to meet evolving business requirements.

Microservices

Individual pieces of business functionality that are independently developed, deployed, and managed.

API-First

Built with APIs from the ground up. All functionality is exposed through an API.

Cloud-Native SaaS

Software-as-a-Service that leverages the full capabilities of the cloud, beyond storage and hosting, including elastic scaling of highly available resources. Functionality is updated automatically, no manual effort required.

Headless

Front-end presentation is completely decoupled from back-end logic. Designed to be channel, programming language, and framework agnostic.

The Rise of MACH

> “ It’s not a new microservices world, it’s a new modular world. Companies no longer have to choose one solution, it’s a smorgasbord of solutions that you can combine to exactly meet your needs.”

David Beale
eCommerce Technology Director, EPAM Systems

Expanding Digital Needs

“There are three drivers generally for companies adopting this architecture,” says Anna Smedslett, Technical Manager at Valtech, who has led architectural transformation projects for multiple enterprise companies, “either you have a need for speed, you need a higher quality, or you need better processes.”

These needs arise as companies expand their digital business. New channels and services need to get to market quickly, previously siloed data needs to be connected to power the experiences on these channels which, in turn, need to be delivered at scale.

These channels are also no longer limited to customer acquisition. Digital is becoming increasingly important across the customer lifecycle, and post-purchase interactions are moving beyond the web and towards apps and devices such as voice.

For many companies, each of these new initiatives require a custom integration with a legacy platform. Creating a network of dependencies duct-taped together one at a time. Taking care of that network is no simple task, 66% of developers say maintenance of legacy systems and technical debt hinder productivity [3].

Additionally, when each new initiative requires a complicated work-around it limits opportunity. In the Gartner 2019 CMO Brand Strategy and Innovation Survey, 38% of respondents named “limited technology to support innovation” as a barrier that their marketing organization must overcome to drive innovation. [4]

> Part of the reason we went for this architectural pattern is that we don’t want to ever go through a major replatforming again. If a few years down the line, something emerges that is far and above what we have, we want to be able to adopt it without the barriers of the past.
>
> For the last ten years, unstitching technologies has been too big an effort and cost. The new architectural approach represented by headless provides Burberry more flexibility in evolving in the future.
>
> **Giles Smith**
> Director of Digital Technology, Burberry

No All-In-One Platform

When digital was a narrower part of business it made sense for many companies to have a core platform, like a commerce or content management platform, that did the majority of the heavy lifting. As digital business expands, the spectrum of functionality needed becomes wider than any single platform could master.

Instead of trying to be a jack of all trades, modern software vendors are becoming masters of a specific area and are partnering with other leading tools. According to the Gartner Apply the Principles Behind the Future of Applications to Digital Commerce report,

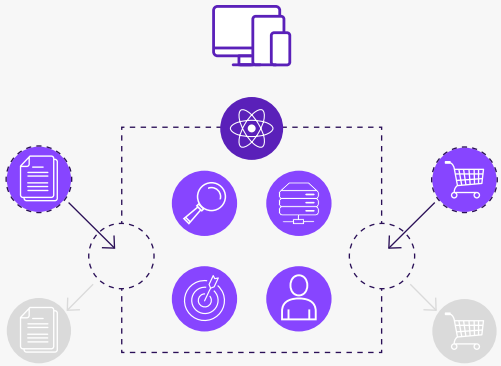
“Where Is the Composable Commerce Architecture of PBCs [Packaged Business Capabilities] Emerging? Loose alliances of best-of-breed vendors appearing such as JAMstack (JavaScript, APIs, and Markup) and, in digital commerce specifically, MACH (Microservices, APIs, Cloud and Headless). The key difference between this new approach to building a commerce portfolio and previous iterations of digital architecture is that there is no single “core” platform — all vendors play together in the composable application, and the end-to-end customer journey would fail without them. At the same time, they are relatively pluggable and replaceable, preventing vendor lock-in for a given capability. [5]

Giles Smith, Burberry’s Director of Digital Technology, says that the freedom to choose the best platforms for each functionality was a key driver behind the company’s move to MACH.

“If we had to choose a monolith that was a solid nine out of ten on commerce, but a seven out of ten on search and content, we would have been limiting ourselves. We wanted to be able to pick the best platform for each of those capabilities. We didn’t have to compromise by going for a single platform that locked us into some strengths and some weaknesses.”

This loss of a core platform is not just felt in the commerce space, but in any industry where digital ambitions are surpassing the capabilities of a single platform. At Contentstack, we see this in enterprises across media, gaming, B2B, hospitality, services and more.

Evolve with MACH



Your company can keep itself market-leading at all times by switching outdated tools and channels with updated new ones that are easy to integrate with the API ecosystem.

Built to Evolve

Designed to play nice with others, MACH technologies create a framework of future flexibility that lets companies evolve at their own pace. Neha Sampat, CEO of Contentstack, explains why companies like hers are focusing on designing products that are extensible.

“Companies are modernizing their approach to digital in stages. They need to be able to access new technology and tools now, while transitioning their stack over time. Modern software needs to integrate not only with other new technologies, but also with legacy tools to make the wider digital transformation as smooth as possible.”

Going MACH doesn’t mean that a company has to completely re-engineer their architecture all at once, and most enterprises step off legacy bit by bit.

Benefits of MACH Technologies

Why enterprise companies are adopting microservices, API-first, cloud-native, and headless technologies.

	BENEFITS:	FITS COMPANIES THAT:
Modular Design Designed to easily integrate, MACH tools are easy to add in and take out of your stack	• No vendor lock in • Select best fit solutions for each domain, instead of one large platform that works "ok" for multiple domains • Evolve stack over time, without large replatforming effort	• Have an existing architecture that contains multiple systems • Have a cross-departmental digital strategy
Freedom to Innovate Natively built with modern development tools and frameworks, MACH solutions are highly flexible and adaptable to unique needs	• Work with the architecture, language, and framework you prefer • Leverage out of the box features of platforms without being limited to them • Design unique customer experiences	• Have access to proficient development talent, in-house or via a partner • Need digital to be a competitive differentiator
Modern Architecture MACH technologies fully leverage the latest advancements in cloud and API design	• Create highly performant, fast, and available experiences at scale • Access cloud capabilities such as auto-scaling and geo-redundancy • Continuous rolling upgrades of SaaS solutions, no manual effort required	• Experience peaks that need scalable infrastructure, such as holidays or live events • Have a substantial percentage of business success dependent on digital • Have a global customer market
Channel Agnostic Decoupling back-end logic from front-end presentation allows MACH solutions to power any touchpoint	• Data, content, and logic can be shared across channels for a consistent experience • Faster time to market for new customer touchpoints • Consumer applications do not change when functionality vendors are replaced	• Manage multiple websites, locals, devices, or touchpoints • Are bridging online and offline customer journeys with kiosks, digital displays, and smart stores • Want to experiment with new channels such as auto, AR and VR • Have digitally native customers
Cultivate Top Talent Attract, grow, and retail top development and marketing talent by creating an environment of autonomy and continuous improvement	• Access to market leading capabilities and features • Less time spent on troubleshooting legacy systems, more time spent on creative and strategic work • Empower teams to make decisions based on business value, not requirement specifications	• Are familiar with agile development and DevOps models • Have a culture that encourages collaboration and feedback • Prioritize employee motivation

Is Your Organization Ready for MACH?

> "MACH is not a silver bullet. If you can't use the benefits of the solution then you just get the drawbacks of complexity."

Anna Smedslett
Technical Manager, Valtech

Just as you wouldn't wire up an entire smart home if the end goal was to use an app to turn off the lights, you wouldn't create a full MACH architecture if all you needed was a pretty website.

You can gain the benefits from these modern platforms without having to go fully MACH and, for many companies, starting with a single platform is all that is needed.

Going fully MACH is most beneficial for enterprises who have digital ambitions beyond the capability of legacy tools, or already feel slowed down by their current infrastructure. If competitors are rolling out new customer experiences far faster, talent is frustrated (or leaving) because they can't do their job successfully with outdated tools, and new initiatives are shelved because of the effort needed to rewire the system — it might be the right time to go fully MACH.

Making this transition requires just as much organizational change as technology change, and companies need to ensure they are ready for this new modular way of working.

Digital Maturity

Modern technologies let companies move quickly, but leveraging this speed requires moving from a traditional to a modern mindset.

"We know that in a world of A/B testing maybe 80% of our ideas may fail a test, so we needed to be able to have a platform and engineering process, and a culture, where we could deliver things super fast," Smith says about the necessary digital maturity. When starting this project, his team was already moving to a fairly mature Agile and DevOps model. "Without those things in place, you're going to struggle to get the early momentum that a program like this needs."

Project to Product

Alongside a culture of fast feedback and iteration, switching to MACH means looking at each initiative as a full product, rather than a one-off project. Each product has a multidisciplinary team that is responsible for the full life cycle — planning, design, implementation, launch, value measurement, improvement, maintenance, and retirement.

Products can range in granularity, from a mobile site to a single API, and have discrete capabilities and business value attached to them. Teams continuously test, evaluate, and improve the product based on this business value. Whereas projects have a clear end date and focus on delivering to requirements, products have evolving roadmaps and focus on delivering functionality and measurable value add.

This style is a long term strategy that favors testing and gradual improvement over time.

Early Adoption

Working with cutting-edge technologies brings the opportunity to roll out customer experiences far ahead of the competition. While leading the pack down a new digital road lets you pull ahead in the experience race, it also means driving on a course not fully marked yet.

"There isn't a nice Medium article that can tell you all about how it works, you've got to really work through it yourself," says Morgan, who emphasized the need to not underestimate the amount of hands-on learning required for a MACH approach. "Let's be very honest — I think that if you don't invest time in really understanding the architecture, it's a bit of a risk."

New technologies can be a powerful differentiator, but if technology is not a current priority for the organization it will be hard to get the mandates needed to hire the right teams, to iteratively develop, and to be trusted to know when the reward is worth the risk.

"No one is going to fire you for looking at the more monolithic and traditional platforms," says Smedslett. "On the other hand, if you're a challenger, it stops making sense to use the exact same setup that your competition is using."

A New Type of Partnership

When you are behind the wheel of your car you may feel in control but, in reality, you are trusting the people who designed the engine, brakes, airbags, etc to get you safely where you're going. As a system gains complexity, it's less about control and more about reliability. [6]

"When you make the choice to adopt something 'as-a-service' you're trading a degree of ownership and flexibility for convenience, speed, and business results. You move to the cloud and you give up control of your servers, you don't configure them anymore, but you gain the ability to do work faster and cheaper in the cloud." explains Matthew Baier, COO and co-founder at Contentstack.

This level of reliance, combined with the support you will need as an early adopter, makes it important to select highly collaborative, transparent vendors.

One issue that does arrive from this new form of partnership is the increased demand in the sourcing process and managing more partners . "If we had gone with a big platform, we might have ended up with two technologies for five capabilities." explains Smith, "we ended up with five for five. Which meant we needed increased support from our internal partners Procurement; Security and Legal , as well as our own team five times over a short period."

> When we got to a position where we were very comfortable with the technology, it was always about developing a partnership. I see the teams we've chosen, the companies, as partners, never vendors, never service providers. The way I see it is that Contentstack are underpinning Burberry's experience, and they are responsible as much as my content team for amazing content experiences and innovations. I think a culture like that, beyond the contract, beyond the commercial, is truly a partnership. Where you can rely on anyone at any level in that organization to feel responsive to an issue.
>
> **Giles Smith**
> Director of Digital Technology, Burberry

Brand Alignment

When technology touches every part of the organization, making a big shift means getting a spectrum of stakeholders on board. Every department won't speak the language of APIs and cloud, but everyone has an interest in the brand.

"When you work in an organization like ours, that is two hundred years old, the brand is incredibly deep rooted in people's decision making." says Morgan about the importance of mirroring the architectural change with the *Spectator* brand.

"In order to have a chance of changing people's perceptions around technology, I really needed to be able to speak the brand language. That was a case of spending immersive time with our product, but also spending immersive time with our customers. Above all the other change mechanisms we did, all the prototypes and all the bulldozing, that was probably the most powerful one."

By taking the time up front to conduct interviews, workshops, journey mapping, and being very honest about the customer pain points in their current digital experience, Morgan was better able to guide the program overall.

"It means that now, when I sit with other stakeholders and talk about decisions from a technology point of view, I can talk with a position of authority about what that means to our readers and what that means to our customers, but also what it means to our legacy and our institution."

> We're going to be the first UK publisher to be doing headless at this scale. You can't do that without an understanding from your solution partner that it's more than just a software-as-service relationship. It's a deeper part of the journey.
>
> **Tom Morgan,**
> Director of Digital, *The Spectator*

Evaluating MACH Solutions

> "Architecture is very much a social thing. It is that fuzzy, embedded understanding that works."
>
> **Martin Fowler**
> Making Architecture Matter [7]

Identify Your Core Criteria

Moving to MACH gives enterprises the freedom to design an architecture around their specific business needs. The core tenets of that design, where the focus makes the most business sense, will differ for every enterprise company.

This can make evaluating new technologies complicated. When your architecture is custom to your business, there is no standard industry RFP. Not to mention that once you open the possibility of using any best-of-breed solution, you also open the door to the ever-growing Pandora's box of enterprise software.

Speaking from his experience helping multiple organizations go through this process, David Beale, eCommerce Technology Director at EPAM Systems, recommends first creating a core, prioritized criteria list that is very specific to your needs. "It's not just about basic functionality, there are a lot of peripheral considerations about operation, deployment, changes, and support that need to be taken into account."

The emphasis here is "core".

When *The Spectator* started this journey, Morgan was given a feature list as long as his arm and a backlog of ideas the business had been sitting on for years. "The first job I had was to try and whittle that down until we got to really core premise basics. Before we even made any architectural decisions, we agreed on three core tenets: make it really fast, make it really pretty, and make it really usable."

At Burberry, Smith's team went through a similar process. "We summarized our assessment criteria to capability, commercial value and partnership. Capability encompasses the business tools and their capabilities for our teams, but also the technology, extensibility, and framework that it's built on. But if you had to weigh these things, for me personally, partnership was slightly ahead of all of these."

Aligning on core criteria early, and making them a mantra throughout the process, makes it easier to make unified decisions along the way. As well as enable teams to make independent decisions that are cohesive with the overall architecture.

Core Tenets of Digital Transformation Decisions

THE SPECTATOR	BURBERRY
• Fast	• Partnership
• Usable	• Capability
• Pretty	• Commercial Value

Cross Departmental Teams

Aligning on these core criteria requires input from every department that will be impacted by the new solutions.

For Burberry, this meant that platform decisions were made by a cross section of engineers, content managers, merchandisers, and architects. Smith says that this early collaboration helped the overall program run smoothly at every stage. "There will always be mini compromises along the way, but you can have a lot of unity from the very beginning in the choices you make. That sets you up well for the phase where you deliver."

Setting up this collaborative culture, where everyone trusts they are working towards the shared goals, also leads to quality feedback down the road.

"We encourage all our developers to sit with the editorial teams when they do any demos and take feedback on the chin." says Morgan. "It's important to not be too thin skinned around that. It's part of the lifecycle of building a good product.

Evaluate Hands On

"I always recommend to any other company going on this journey to hack your way past the sales deck." says Smith. To do this, his teams put the shortlisted platforms to the test with mini-hackathons that tackled real business cases.

"There were three key selection decisions that were thrown completely on their head by doing a small, one day hackathon. We were able to evaluate bottom up, with facts and evidence, on business cases signed off on by the CIOs. Effectively, decisions were changed because we spent that little bit of time proving things worked."

Using genuine business cases for these evaluations, beyond a standard demo of a home page or product page, is important.

First, you don't waste time and effort on work that will be thrown away. For example, Burberry's task for commerce platforms was to streamline a payment journey from auto-capture to refund, and they evaluated DAM solutions on the ability to integrate with their current photo studio.

Second, you can ensure the product works how its marketing demo and brochure says it does. Using the APIs yourself will help you determine the quality, extensibility, and supporting documentation.

Third, you get to see if the company is a good match to partner with. Says Smith, "If an organization is up for this it suggests that they are very confident in the authenticity of what they've got as a platform."

Build vs. Buy

> An overarching question you need to consider on the topic of build versus buy is what are the consequences of buying? Will you be able to change or add functionality later on? What does the licensing model look like? Does it lock you in or do you pay for only what you use and have the freedom to replace the solution if you want?
>
> If you can say yes to that, then you should probably buy it, and focus on building the capabilities that differentiate your offer.

Anna Smedslett
Technology Manager, Valtech

> In general we see that back-end capabilities, the SaaS platforms, as ones to buy. It's not exclusive, but that's the way we lean. Our front end is custom built by us. The view is that we will differentiate ourself through our custom website, personalization, and customer data so we own those capabilities ourself.

Giles Smith
Director of Digital Technology, Burberry

> Don't reinvent the wheel. There is no need to create custom functionality if it is already available from a third party. Build what is unique to you..

David Beale
eCommerce Technology Director, EPAM Systems

Common MACH Criteria

Decoupled

Services have less dependencies on one another and are independently deployable and scalable. Changes can safely be made on one service without affecting the others, making it easier to test, develop and implement custom features.

WHAT TO LOOK FOR:

- Is functionality available in discrete, individual pieces or do you have to use the entire application?

Cloud-Native

Software is hosted and managed by the vendor, as a SaaS or PaaS. Able to leverage cloud beyond storage and hosting.

WHAT TO LOOK FOR:

- Platform updates should happen automatically, if there is manual work, or license fees, required for versions and upgrades it's likely a managed service model.
- Functionality should be available over a URL or other form of API. You should not have to download the software and run it yourself.

> There's a huge difference between developing a system and adding APIs compared to building a true, API-first platform when the APIs are created alongside – not after – development. Contentstack takes a MACH approach from the ground up, so there is nothing in our management console, nothing in the product UI, nothing in the back end that you can't address equally through the API.
>
> **Matthew Baier**
> COO and Cofounder, Contentstack

Natively Headless and API Enabled

Back-end logic is completely separated from presentation, enabling use of any touchpoint. APIs give access to every function within the platform and work with all front-end frameworks.

WHAT TO LOOK FOR:

- If there is a presentation layer that has been "decoupled" with APIs, this is a sign that it is API-added, not API-first.
- There should be software development kits (SDKs) for a variety of front-end providers, mobile apps, and traditional application programming languages.

Modular Implementation

Ability to easily integrate with other technologies. Supports incremental implementation and evolution, as opposed to a Big Bang launch.

WHAT TO LOOK FOR:

- Are integrations with your critical solutions built into the product, available as a how-to-guide, or in the same class of technologies the vendor has experience integrating with?
- What are examples of early prototypes clients have gotten to market quickly? How did these prototypes mature over time?

Agility and Extensibility

Seamless, transparent and regular platform updates. Supports a CI/CD approach.

WHAT TO LOOK FOR:

- How frequent are new features, how are they released, and how do you receive them?
- How is backwards compatibility handled? Is there a clear versioning method in place for APIs?

Scalability

Scale resources automatically for peaks and flash events. Fully elastic consumption maintains optimal performance at the lowest possible cost.

WHAT TO LOOK FOR:

- Are there performance statistics available from a recent event, such as the holiday season or a large live event?
- Is there documentation on caching, webhooks, and containerization?

Stability

Reliable service and uptime regardless of updates, traffic, and special events.

WHAT TO LOOK FOR:

- Is the vendor transparent about availability and performance? For instance, providing a publicly available status and availability monitoring.
- How do you access customer support, when is that support available, and does it come with an extra cost?

Strategic Alignment

Vendor is a partner who understands and supports your business needs and goals. There is a highly collaborative and transparent relationship.

WHAT TO LOOK FOR:

- Read customer reviews and ask to speak with reference customers.
- What is a recent feature that was added based on a customer suggestion?
- Does their feature roadmap support your strategy?

Documentation

Easy-to-use, searchable, self-help knowledge database. Clear and up to date API documentation, SDKs, and how-to-guides.

WHAT TO LOOK FOR:

- Test the quality of vendor's documentation with a hands-on evaluation, or hackathon, focusing on a real business need.

Ease-of-Use

The solution is equally valuable for all users. Features, functionality, and UI are quickly adopted by both technical and business users.

WHAT TO LOOK FOR:

- Include a cross section of users in the evaluation process.
- Is there training available, live or on-demand, for all users of the product?

> One of the biggest challenges was learning about the wider architecture, as it's fairly new. You find yourself relying a lot on platform documentation.
>
> **Tom Morgan**
> Director of Digital, *The Spectator*

> "The best question you can ever ask in an evaluation is if they can give you a specific example of where they deployed something recently that came from a customer idea. For the platforms we chose, within seconds a person in the room could tell us one or two very good recent features they built because a customer suggested it. That is really authentic insight showing that they are listening to and partnering with their customers."
>
> **Giles Smith**
> Director of Digital Technology, Burberry

Moving to MACH

> "A lot of times it's not a direct mapping from an old school approach to this next generation approach, it's how you think. If you're adopting this new style it's a different paradigm."
>
> **Kelly Goetsch**
> Microservices for Modern Commerce [8]

Sourcing Strategy

Moving to a MACH architecture requires finding the talent and expertise to support the new approach, and it should be understood early on where both of these will come from.

For instance, certain roles will only be needed for the initial shift, while others will be needed for the long term evolution of the platform. It may be helpful early on to bring on an external implementation partner with MACH experience. They can source the talent needed to get the ball rolling, help design and implement a solid foundation, and be an information resource for your long-term team.

Vendor partners are also good to leverage early in the processes. Offering a product designed for modern digital business, these companies usually have experience with finding solutions to unique requests. Having vendors work directly with your team also means that real business cases can be used for training and education.

Choose the Granularity

"It's easy to get carried away with creating many different services." says Beale, "You've got to follow a good design methodology about understanding what constitutes a service. One key to that is Domain Driven Design, which is understanding the boundaries of each service. So you're not adding unnecessary complexity to the application."

A recent Gartner report discusses the method of designing around business capabilities:

"The future of applications includes the creation and integration of packaged business capabilities (PBCs), which are software components that represent a well-defined business capability, recognizable as such by a business user. A PBC consists of a bounded collection comprising a data schema and a set of services, APIs and event channels. Well-implemented PBCs are functionally complete to ensure autonomy with no critical external dependencies and no need for direct external access to their data. The creation, adoption and integration of PBCs enables assembly of custom application experiences and is a foundation of a "composable enterprise" [5]

Decide up front just how "micro" you want your microservices to be to keep complexity in check.

Map Service Interaction

Having multiple platforms working in tandem means extra consideration for how the services interact. Document what service owns which bit of data, as well as processes around considerations like caching, image delivery, static and dynamic delivery, plugins, libraries, and workflow.

Organizing this information early, such as in an internal wiki, will help solve issues quickly.

During *The Spectator's* six month program to switch to MACH architecture, Morgan said that almost every week there was a new consideration to take into account. While these did take time, he said that "the good news is that the answer is always to call an API and make it happen, which is easy."

For example, when *The Spectator* publishes a piece of content, a webhook passes details to a Lambda which creates a static version of the page, stores it in Amazon S3, and serves it through CloudFront. The next time the page is updated, it takes roughly a minute for CloudFront to invalidate the previous cache and serve the updated page.

This meant that content editors were not instantly seeing their changes when they published, and they thought the CMS was broken. As a solution, Morgan and his team created an editorial environment that doesn't cache.

To help uncover these types of considerations early, Morgan suggests assigning someone the role of content architect. A person who can think about content structure from the perspective of both editors and developers and help guide topics such as reference objects and workflow.

Decouple Your Front End

"A common first step is being able to decouple your front end." Beale advises his enterprise clients, "One way to do this is, in effect, leaving the front end to be identically as it is currently and API-enable the back end. It's a relatively simple first step and, while it brings no immediate improvement to the customer experience, you've laid the foundation to be able to start your journey."

Decoupling early gives businesses the agility to make changes to the front end quickly while updating the bigger back-end changes at a more moderate pace.

"Being able to have a decoupled front end that is built in the technology we want means that the front end has come together ten times quicker than if we had done

anything else." Morgan says that the speed of the front end counter balances the additional experimentation time that comes with being an early adopter.

Once decoupled, you now have the facility to replace what that front end is connected to, service-by-service.

Prototype for an Early Win

"You've got to build a prototype as soon as possible. Not just to show people but to improve your own understanding." says Morgan, who also recommends starting with a prototype that is commercially well positioned and can start driving revenue.

Morgan started with *The Spectator's* checkout page. Using Angular for the front end and Piano for the back, the team was able to re-engineer the page into a modern static site in five weeks. Not only did conversion rates significantly improve, but the solution still runs today as the main checkout for subscriptions.

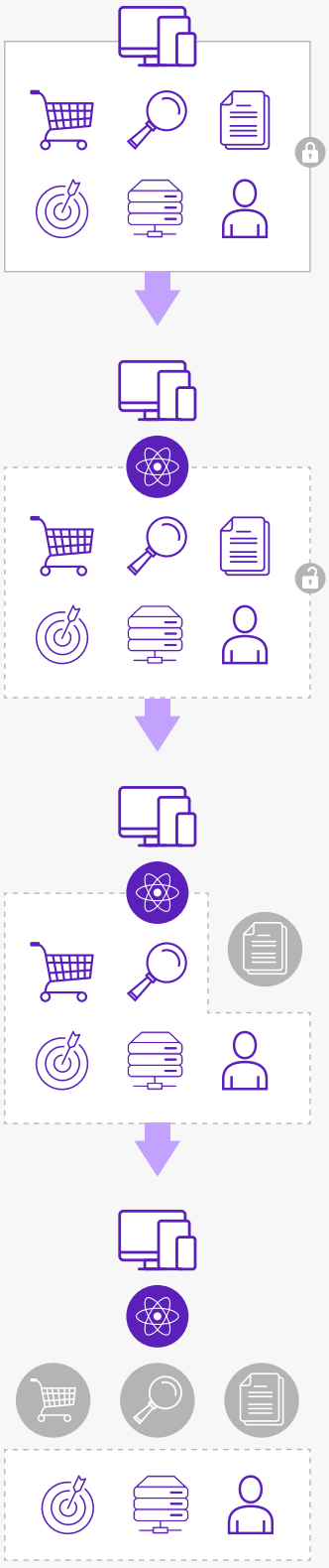
"That was the most powerful prototype you can deliver because every single day when we get a conversion report and subscriptions report we can see the numbers are materially different. The proof in the pudding was when the business was no longer asking about what technology was being used, but what the focus was. I think when you win that argument you can essentially have carte blanche to pick the technology that aligns with your core criteria."

To choose which capability to start with, Smith and his team at Burberry looked at the options through the lens of customer impact, productivity and team impact, and commercial cost impact. He explains, "Each of these capabilities, whether it be search, identity, digital asset management, or content management, they had a very different weighting of these strengths."

Considering these criteria, they focused on search first because it offered the most customer impact for the least complexity. Working with Attraqt (who acquired Fredhopper site search in 2017), the team was able to go from kick off to live in 12 weeks.

"It's about constantly looking for the most lean journey to give you momentum." says Smith. "That gives confidence to people that the rest of your choices, and the rest of the project, are worth doing."

Decouple at Your Own Pace



Step Off the Monolith

Smith says that one of the strengths of Burberry's approach was to chop the program into small bits and not attempt to remove the legacy platform all at once. "There is a delicate balance, which was one of the biggest challenges, of how to get to the world of new technology and all the benefits it brings without having to switch off our existing website or, even worse, to stop innovating on it. We have our own trading needs and our own passion and reputation in digital, so there is always this balance of going off the old to a sensible degree while not slowing down your ability to get into the new world."

While the exact steps will differ for every company, a key principle is to stop feeding the monolith. The monolith should, with each new functionality, become increasingly dependent on the services.

In her guide to breaking up a monolith, Zhamak Dehghani gives the follow example:

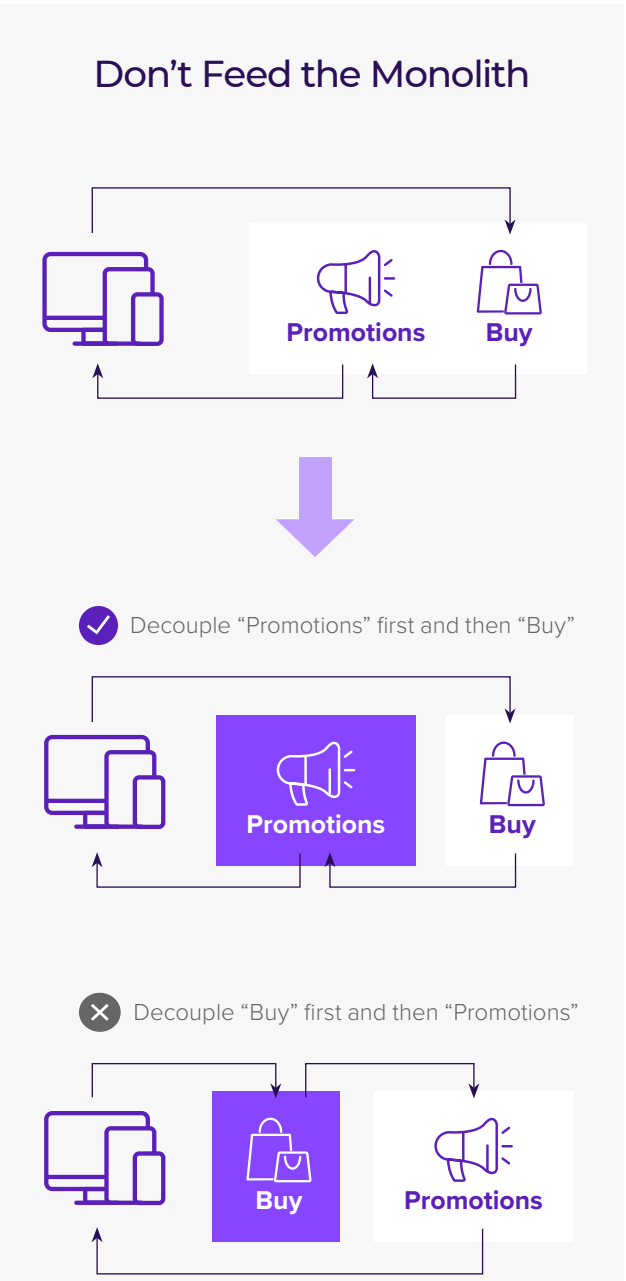
"Consider in a retail online system, where 'buy' and 'promotions' are core capabilities. 'Buy' uses 'promotions' during the checkout process to offer the customers the best promotions that they qualify for, given the items they are buying. If we need to decide which of these two capabilities to decouple next, I suggest to start with decoupling 'promotions' first and then 'buy'. Because in this order we reduce the dependencies back to the monolith. In this order 'buy' first remains locked in the monolith with a dependency out to the new 'promotions' microservice." [9]

This also means that new features and functionalities should be implemented as services. Performing a one-off "quick fix" by adding something new to the monolith is just moving the goal post further away.

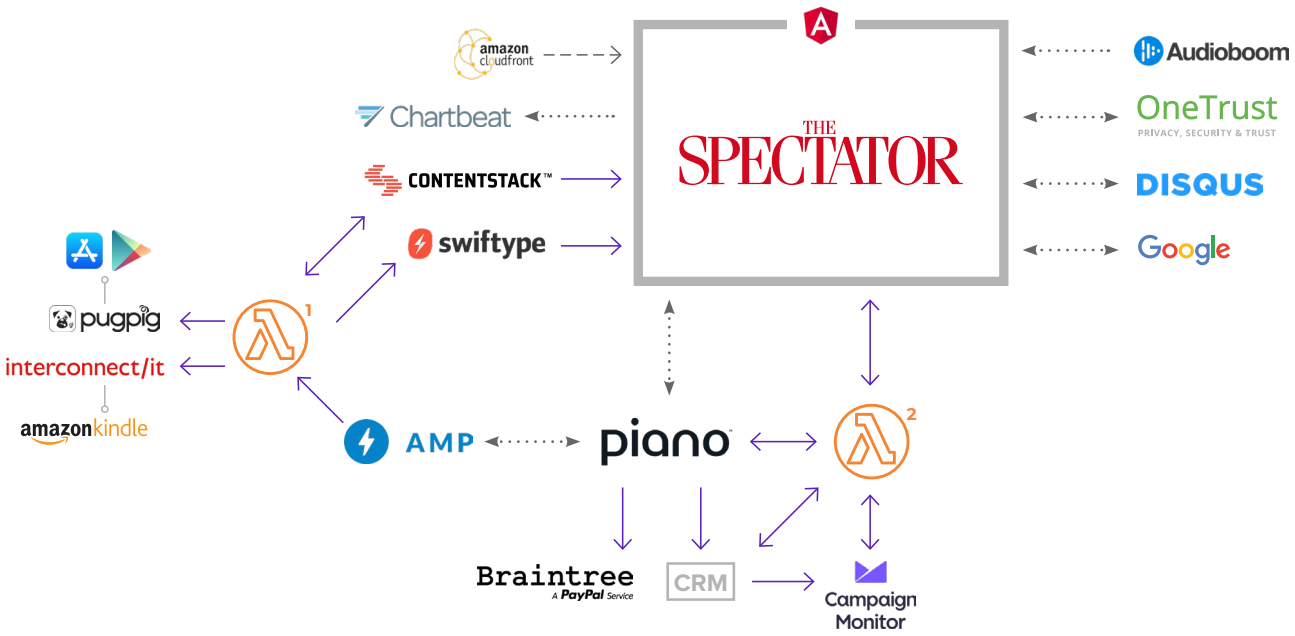
The following readings offer a more in-depth look at technical strategies to step off legacy tools.

BLUEPRINT ARCHITECTURE FOR MODERN COMMERCE
commercetools
Diagrams and details of two potential headless commerce architectures.

HOW TO BREAK A MONOLITH INTO MICROSERVICES
Zhamak Dehghani
A step by step look at how to choose what, and how much, to decouple.



The Spectator's Experience Architecture



API ⟷

3rd Party JS ◄·····►

HTML/Images ⟷

Out of Scope/ Unknown ——○

Arrows indicate direction of data flow

Angular
Front-End Framework

Swiftype
Search

Contentstack
Content Management System

AMP
Web Component Framework

AWS Lambda Edge
Serverless computing
1) Post-publishing actions
2) Service broker

interconnect/it
Kindle publishing

Pugpig
App delivery

Piano
Identity management, subscriptions, experience delivery

Braintree
Online payment

CRM
Customer Relationship Manager (undisclosed)

Campaign Monitor
Email marketing

Google Tools
Tag Manager, Analytics, Optimize, Marketing Platform, Ads, AdSense, Ad Manager

CloudFront
Content Delivery Network (CDN)

Disqus
Comments

Chartbeat
Content analytics for publishing

OneTrust
Cookies and consent management

Audioboom
Podcast delivery

Self-Built Services
Recommendation, Most Popular, Read Next

Recreated with permission from: Spectator / The rebuild / Part 1: The tech stack

Ecommerce MACH Architecture

API Orchestration/ Back End for the Front End

An API gateway that coordinates the experience services. This is the logic layer that determines what information will be given to the user. Usually fully owned by the enterprise.

MACH Experience Technologies

Agile, flexible tools that are focused on the external user experience. Loosely coupled, these tools pull in information and adapt their output to changing situations.

Business Data and Processes

Tools that focus on data and processes of the internal business. They prioritize consistency over agility and do not need to be as flexible as the experience technologies. How they integrate with and push information to MACH solutions is a way to differentiate the customer experience.

A full MACH architecture may not be for every company, yet.

Traditional platforms are more familiar for many organizations, and are relatively easy to deploy out of the box. On the outside, these platforms seem simple. However, on the inside they become more complex over time. Every new feature or dependancy is duct-taped on one at a time, and this growing complexity slows down development year over year. Releasing new initiatives to market can take months, even years, or become too complicated to attempt at all.

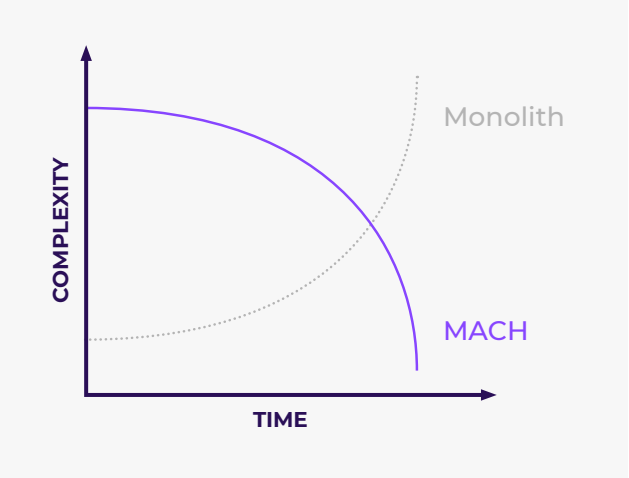
MACH technologies, on the other hand, do take time and planning up front to map out and properly set up. The initial push is a more complex process than deploying a traditional platform but, once the structure is in place, MACH technologies make digital evolution simple. Their building block style of design means that releasing a new feature or creating a new dependency just requires clicking in a new block and can happen in weeks, days, or even hours.

This building block nature also means that companies can start using MACH in any corner of the business and transition over time. For many enterprises, starting with a single MACH technology can be and often is the catalyst for a wider change and digital maturity.

The digital experience race isn't going to slow down. Rapid change — in consumer demands, business needs, the technology marketplace — is inevitable and enterprises tangling themselves up in legacy platforms are at a disadvantage to the nimble organizations that can quickly leverage new tools, features, and channels.

Speed gives the power to differentiate with digital. While every company in the race is tasked with changing their bike into a car, some need a racecar and others an off-road powerhouse. Companies with an experience best adapted to their customers' needs are the ones who will win. MACH technologies let companies design their kit piece by piece and create an experience that sets them apart from the competition.

The speed, scale, and performance that MACH technologies offer, along with the ubiquitousness of cloud and APIs, lean towards a future where this type of modular design will be one of the common enterprise architecture patterns. Early adopters are paving the way for this to happen, exiting the replatforming loop, and securing their spot at the front of the digital experience race.



REFERENCES

1. Silver, L. Smartphone Ownership Is Growing Rapidly Around the World, but Not Always Equally. Pew Research Center. 5 February 2019.

2. Perez, S. Over a Quarter of US Adults Now Own a Smart Speaker, Typically an Amazon Echo. TechCrunch. 8 March 2019.

3. Devada. 2019 State of the Developer Report. 19 September 2019.

4. Gartner Press Release "Gartner Reveals the Top Three Barriers to Innovation in Marketing," 11 November 2019.

5. Gartner, "Apply the Principles Behind the Future of Applications to Digital Commerce," (Gartner subscription required), Jason Daigler, et al, 8 January 2020.

6. Waterhouse, H. Everything is a Little Bit Broken; or, The Illusion of Control. [Presentation]. O'Reilly Conference. 11 December 2019.

7. Fowler, M. Making Architecture Matter. [Presentation]. O'Reilly Conference. 23 July 2015.

8. Goetsch, K. Microservices for Modern Commerce. O'Reilly. January 2017.

9. Dehghani, Z. How to Break a Monolith into Microservices. martinfowler.com. 24 April 2018.

CONTENTSTACK™

49 Geary Street, Suite 238
San Francisco, CA 94108

info@contentstack.com
+1 415 255 5955
www.contentstack.com