

Piece it together

From monolith to microservices: Horror stories and best practices



Bill Doerrfeld

Consultant, Doerrfeld.io

They say software is getting smaller—micro, in fact. Microservices are the latest software architecture trend, and its evangelists preach [enterprise microservices](#) adoption. "Decompose the monolith" and "decouple your infrastructure" are mantras that echo from the boardroom to the developer terminal.

The urge to chisel away the monolith is guided by philosophies such as [Conway's Law](#), which states that a system mimics its communication structure, and by thought leaders such as [Martin Fowler](#), who has spearheaded thought on microservices architecture.

The theory of decoupling seems sound—microservices' benefits extend to [application security](#), [speed](#), and [scalability](#). But what are the repercussions of migrating to a microservices design in the real world? Who has adopted microservices, and, more importantly, who has done so successfully?

Here are three companies' journeys from monolith to microservices. Architects currently or formerly at Best Buy, Cloud Elements, and Wix.com share some of the world's most advanced microservices design escapades—horror stories

and best practices from the microservices trenches.

From big box to online: Taking on the monolith

Best Buy, Cloud Elements, and Wix.com have all been through the bogs of decomposing a monolith. And they must have done something right, because they were willing to talk about it.

[Aviran Mordo](#), vice president of engineering at Wix.com, reported that the company's "original architecture was a huge Java monolith that handled everything." From user registration to login, to editing Wix sites and serving content, it did it all.

Similarly, according to [Ross Garrett](#), vice president of product and marketing at [Cloud Elements](#), version 1 of its architecture was a monolithic application. Internally nicknamed "Soba," it quickly became a "tangled mess," as the noodle-related name would suggest.

Previously the senior product manager for the Best Buy API platform, [Kristen Womack](#) arrived at the company to face a tightly coupled Oracle database. Tied to a single monolithic structure, the company's three architectures became intermingled, which prevented the business from being "as nimble as it needed to be," Womack said. (Womack is currently principal of Night Sky Web Co., which she founded.)

Why they turned to microservices

Wix.com adopted microservices to reverse a massive [technical debt](#) that had created serious stability issues. Since applications were interconnected, Mordo noted, "bugs in one part of the system had the potential to bring down our whole system." In 2010, the company began to break things into smaller services to better handle scalability and quality assurance.

Similarly, Best Buy's interdependent architecture became a bottleneck for

deployments. Womack explained that "downtime was too long to sustain for conducting business online."

At Cloud Elements, the transition also came out of necessity. Garrett described the firm as a "hyper-growth company." For a company experiencing exponential growth, a microservices design helps to continually iterate to match new demands.

"Simply put, what was true in 2014 was no longer true three or four years later, and we needed to re-architect and scale the core platform features in a way that wasn't possible in the original monolithic application."

—[Ross Garrett](#)

Hurdles along the way

Once you enter the distributed systems paradigm, unique issues arise, such as handling communications among microservices, addressing failures, and debugging problems. Wix.com had to invent new integration and end-to-end testing patterns and foster a new internal culture around these guidelines.

"We built a highly specialized microservices framework to handle [cross-cutting concerns] for every microservice that is built on top of this framework."

—[Aviran Mordo](#)

At Cloud Elements, Garrett described a journey of discovery that "requires design and debate." Differing opinions and conflicting requirements may lead to many people quoting (and reinterpreting) Conway's Law.

For Womack, one of the biggest challenges at Best Buy was building trust. Instead of packing features upon features into infrequent releases, teams had to make a dramatic pivot. "Cultural resistance to how software is built and deployed is to be expected," she said, because it's "especially challenging when

it changes how people do their jobs."

Show and tell: What's in your microservices stack?

Hardly any support tools existed when Wix began its microservices journey in 2010, so it implemented its own JSON/RPC protocol, and built a microservices framework on top of SpringMVC.

Garrett is happy working with many tools at Cloud Elements: Minikube and Docker help run services locally, and Now runs them remotely. All microservices are in Node.js; therefore they leverage npm-based packages such as Ava for unit testing, NYC for Istanbul's code coverage module, and XO for linting.

According to Womack, the Best Buy transformation introduced CI and deployment tools such as Chef and Jenkins. However, the biggest change was moving to Riak, a distributed NoSQL key-value data store.

"This changed the entire data model and how people and systems accessed the product catalog."

—*Kristen Womack*

How microservices affected business

Since Wix improved developer speed and the quality of its Wix services, Mordo said, the migration is "one of the best decisions we have ever made." Similarly, at Cloud Elements, the ability to design and deploy independent services has been a godsend, bringing performance gains and operational predictability to the development pipeline, Garrett said.

On the other hand, moving to a microservices design doesn't guarantee benefits for every business. A monolithic codebase may be more efficient, with less overhead, for smaller-scale applications.

"A microservices architecture is definitely not a one-size-fits-all solution. Many teams start with a monolith and then move to microservices as their product or business scales. Seeing how your organization and product mature over time will inform how your architectures evolves."

—Kristen Womack

Another concern is your talent pool. Where software companies such as Cloud Elements have the skills necessary to handle such a transition, Garrett said, the same is not always true for businesses with roots outside of the software domain. Still, these non-software businesses could gain a competitive edge by going the microservices route.

"Microservices—implemented thoughtfully—can provide a backbone upon which innovation can thrive. They can provide the reusability of services that developers and product teams require to move quickly."

—Ross Garrett

Garrett added that microservices have also aided [DevSecOps](#) at Cloud Elements. By creating reusable micro-components for authentication and authorization, security traits can be inherited by any product.

Advice for others

These seasoned experts all recommended embarking on a careful consideration process before diving in. As writer and TED speaker [Simon Sinek](#) put it, "[Start with why](#)." This means careful attention to business needs, and ensuring that the technical architecture addresses them.

"Don't confuse architectural elegance with business value. In other words, microservices are discussed loudly and often amongst architects and developers, with little regard for the business."

—Ross Garrett

Microservices are not a silver bullet, Mordo said. The benefits from microservices must outweigh the time spent in migration and maintenance for new issues. And staying with a monolith is a much easier task than handling a distributed system.

"If you do decide to do it, take it slow and start with only two services. Actually start with two monoliths; you will learn a lot by just having a couple. After you have experience with a couple of services and you have figured out all the operational costs, then go into smaller and smaller microservices."

—Aviran Mordo

In other words, don't fall victim to baseless hype, and ensure that the new solution improves upon what you already have. Before throwing it all away, Womack said, new entrants should appreciate their legacy code.

"As you think about legacy code, it's helpful to remember two things. First, it's been running your business this long and got you to where you are today. That is something to celebrate. Second, today's modern technology is just tomorrow's legacy."

—Kristen Womack

Case studies to model

No one is an island. We all pull inspiration from the work of others, and most of these teams modeled other case studies or took advice from luminaries along the way.

To reach microservices excellence, Garrett recommended learning from Martin Fowler and modeling [Netflix](#). In his words, "The leadership demonstrated by Netflix can't be overstated." He has also proposed the concept of "[miniservices](#)," which take the microservices pattern just far enough, without

over-engineering.

In addition to Netflix, the Best Buy team took inspiration from [Etsy's move to an API-first architecture](#). Womack cited the work of [Jez Humble](#) in building out CI/CD efforts. For insights on team organization in microservices, she recommended Susan Fowler's book [Production-Ready Microservices](#).

You can also read more about Womack's role in the [Best Buy transformation](#). She remains a big believer in autonomy as an organizing principal for teams.

Team autonomy "will drive better outcomes and collaboration, but will also help create the decoupled architecture of microservices."

—Kristen Womack

The ROI of microservices

We've explored monolith-to-microservices migration, seeing firsthand how companies embrace [microservices](#) and the numerous benefits they bring. Whether paying off technical debt at Wix.com, streamlining deployment at Best Buy, or allowing a platform to enable growth at Cloud Elements, microservices are an approach that keeps on giving.

"Microservices allow an organization to reduce dependencies, develop faster, and scale."

—Aviran Mordo

To take this route, companies must wrangle service-to-service intercommunication and advance their testing implementations. They must also transcend organizational debates and foster trust in disrupting storied behaviors. In short, for microservices to work, the investment must pay off.

Keep learning

- **Take a deep dive into the state of quality**