

The Impact of Microservices on the Future of the Media & Entertainment Industry

Exploring the building blocks of the modern and futureproof media organization

Abstract

Microservices are the building blocks upon which all modern and successful digital businesses are constructed. This relatively new software design architecture is instrumental to the spectacular and seemingly overnight success of Netflix, Uber, Airbnb and essentially every other digital business to gain commercial prominence in the past decade. It is the technology that forms the connective tissue of the Internet, and the Internet of Things (IoT), the linkage holding together a nearly infinitely scalable digital fabric that makes possible driverless cars, big data and the ability to adjust the thermostat in your living room from virtually anywhere in the world. The public cloud, the stock exchange, the music industry — to name just a few trillion-dollar enterprises — are all fueled by microservices. If you want your media operations to be capable of fully exploiting the IP-based digital realm to which they are moving. If you want your company to be a competitive force in a rapidly evolving television industry — without limits on creativity, agility, scalability or speed to market. Then your business needs to be powered by microservices, too.

This document is designed to provide media industry professionals with an understanding of microservices-based application design and, more importantly, the reasons why the adoption of applications based on this architecture is critical to the long-term success of your business and your company's ability to compete with not only your traditional peers, but with a new breed of media creators and distributors that have fully embraced the digital age. It details the characteristics and benefits of microservices-based applications, including their improvements over applications built on monolithic-design principles. It explains the relationship between microservices and Cloud-Native, as well as the advantages that microservices design can bring to businesses that are not ready to move operations to the cloud. Above all, this document draws a straight and bold line between the underlying software-based building blocks of the applications that run your business and the ability of your business to scale, adapt and evolve to seize the opportunities of the software-defined era of the media and entertainment industry.

Introduction: Welcome to the Application Age

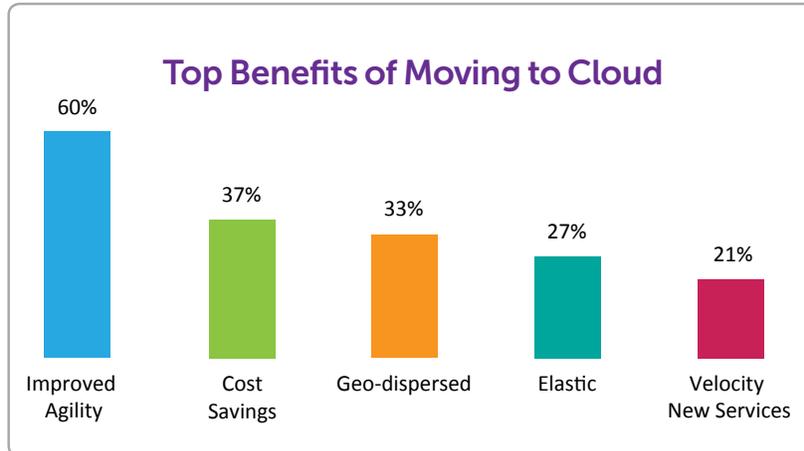
So why should you care about microservices-design principles and other seemingly arcane software development issues? What do the DevOps movement, software design innovation or even the emergence of the so-called "app economy" have to do with broadcasting or streaming video content to billions of consumers? As it turns out, almost everything.

At the core of nearly every successful and disruptive business of the last decade is software that has been natively designed and optimized to run over IT-based networks, including the cloud and the Internet. Software is the engine that drives the digital economy. And for that engine to run at peak performance it needs to be finely tuned to fully exploit the benefits of modern IT-based networks, including the capability to be easily distributed, instantly scale and rapidly incorporate the latest updates and innovations.

It only makes sense that as media companies begin to migrate their operations from purpose-built hardware to the IP-based common computing platforms that power both enterprises and the Internet, the software underlying their

media operations, such as playout, encoding and monitoring, must be designed to fully tap into the digital economy. Building a business on top of a technology foundation that does not support the full exploitation of the digital realm, including rapid innovation and isolated troubleshooting, is tantamount to entering a boxing ring with one hand tied behind your back.

Increasing agility, in fact, is the top benefit of moving to an IP-based, cloud environment, according to more than 60% of media industry professionals who participated in an industry survey on technology and business trends at the end of 2016.



Source: 2017 Focus Forward Media & Entertainment Report

Its impact on the overall health of your business is the reason why microservices design, the software development architecture at the core of the modern datacenter and the Internet, is not just essential knowledge for software coders and engineers. Awareness of microservices design, and its potential impact on long-term growth and competitiveness, needs to fully infiltrate the C-Suite, as well as the board of directors, of any media company that hopes to keep pace with the rapid evolution of the media & entertainment industry. Failing to recognize the importance of this basic building block of the next-generation of media operations could put your business at risk of falling behind your competition — and the staggering pace of change in the industry. That’s a knowledge and capability deficit from which your business may never recover.

The remainder of this document is designed to help all media industry professionals better understand microservices-design architecture, as well as related software development issues and terminology. It contrasts this relatively new approach to application development with monolithic application design architecture, pointing out how the latter’s shortcomings and limitations can negatively impact your business’ ability to rapidly refresh operations with the latest technology innovation or obtain the agility and flexibility needed to compete in the future. Most importantly, it is designed to convey the essential value of microservices design, including a thorough description of its many benefits, and the urgency behind its adoption — even if you have no plans to move your business to a datacenter or cloud environment in the near future.

Microservices-design principles are essential for building cloud-native applications that take full advantage of the scalability and distributed nature of a cloud environment. This document compares and contrasts cloud-native applications to the merely cloud-enabled, as well as explains why cloud-native applications still offer substantial performance and agility benefits to media companies using a premises-based or centralized deployment model.

Microservices-design principles have been in use for more than a decade. Though best embodied by the flexibility, resiliency and continuous innovation characteristics of the cloud, microservices are also connected to the agile software development movement that has revolutionized the enterprise software industry, enabling the world’s leading businesses, including insurance companies and financial institutions, to convert massively large and closed software systems into open and extensible collections of discrete business services. The media & entertainment business is among the last major industries to fully transition to a software-based model. But that doesn’t have to be a disadvantage. By adopting a microservices-based architecture from the start, media companies can quickly achieve parity with other businesses that have transitioned to an Internet-based model and place themselves firmly on a path to success.

Defining Microservices

The simple definition of microservices-based software design is a process that results in an application, or component, that is constructed of discrete services, each of which performs a specific, often simple, business capability or task. These discrete microservices are often designed to be as autonomous as possible, enabling them to be easily scaled, fixed, upgraded, repurposed or even replaced without impacting other microservices that are part of the same application fabric. A microservice typically runs in its own process and is coupled with other processes through a lightweight mechanism. A microservices-based application is also characterized by minimal centralized management, further ensuring the autonomy of the individual service.

Microservices can be deployed independently, or as part of a group or suite of actions. A microservices design enables the construction of applications that are formed through the coupling, or chaining, together of multiple discrete services. In the broadcast or media domain, for example, a simple operation, such as an AVC encode or a graphic overlay, may constitute a microservice. A workflow or a portion of a workflow may be constructed of multiple microservices, chained together to accomplish a desired business objective. See the section [Microservices at Work](#) for a more detailed explanation of microservices in action.

In many respects, a microservices-based approach to software development is an attempt to increase the utility, efficiency and flexibility of an application by breaking it down into piece parts, which can be replaced or updated as needed, or even reused in a separate application, without replacing the entire unit. The concept has been practically applied to the physical world for several centuries, first manifesting in the construction of simple machines and other devices made with interchangeable parts. Automobiles, personal computers and nearly any other complex and expensive device, are now designed in a modular fashion that enables repair or upgrading to be accomplished on a component-by-component basis, rather than replacing the entire device.

The idea of common components is not completely new to the software world. The use of components and common libraries of functionality are design characteristics of the Unix operating system, for example. Microservices design, though, takes the concept to the next level, breaking up applications into building blocks that can be linked and bonded using only the barest of management overhead.

A microservices approach to application development is essentially a replacement or deconstruction of monolithic application design. Monolithic-design principles, as the name suggests, require an application to be treated as a single entity. That means that every time a bug is found or a capability of the application is updated or replaced, the entire application will need to be tested and updated, regardless of the degree of change. This is the reason most monolithic applications are only updated every few months or, perhaps, twice a year, and why they can only be scaled horizontally, rather than on a service-by-service basis. See the chart [Monolithic vs Microservices](#) for a comprehensive list of the limitations and shortcomings of monolithic design in comparison to microservices-based design.

Why Microservices Matter

How important is it for media industry professionals to understand the latest breakthroughs in software development? Why should you care about the innards of the applications that are driving your operations, the lowest-level building blocks of your business?

For starters, you need to care about microservices because you are now competing with digital-first companies, like Netflix and Amazon, who built their businesses from scratch on the most agile and innovative technology foundation available. Born and raised in a cloud-based environment, these companies are unburdened by legacy technology or practices. Rapid innovation, continuous delivery and velocity to market are attributes that have been imprinted into their DNA. Content producers and distributors need to make sure that even the lowest layers of their operations are optimized for the digital realm to properly level the playing field with new media entrants and other competitors.

You should also care about microservices design because your business, to be as responsive and nimble as possible, needs to mimic the physical world, where nearly everything is broken down into parts and components that can be replaced or enhanced on an individual basis — without updating the entire system. Imagine the futility of trying to

compete in the Formula One racing circuit with a car that had to be completely overhauled or replaced with every tire change, tune-up or modification. The resulting expense and inability to make modifications in a timely fashion would make it impossible to be competitive — no matter how fast your car or skilled your driver.

In short, you should care about microservices-design architecture because of its profound impact on the future of your business — and your ability to keep pace with competitors in a digital-first world. How thoroughly and how quickly you move to a microservices model will play a significant role in determining your business' ability to adapt and react to future market challenges and opportunities, including:

- Seamlessly integrating new technologies and capabilities into your workflows
- Quickly and cost-effectively introducing the latest video- and audio-quality enhancements
- Sufficiently satisfying or exceeding the rapidly evolving consumption preferences of content consumers
- Generating new revenue through the easy and risk-free introduction of new business and distribution models

Never in its history has the television industry been challenged by such a high frequency of change. Largely a passive audience for the past 75 years, television viewers are now calling the shots. They expect to have the biggest say in when, how and where they interact with content. The genie has escaped the bottle and continued consumer fickleness will require constant adjustments and modification by media companies. Only a technology foundation based on a microservices-design approach can provide your business with the flexibility and agility to keep the next generation of consumers happy.

What follows is a summary of the most important characteristics and capabilities of a microservices-based infrastructure, and how they might benefit the future fortunes of your business:

Scale: The distributed nature of microservices and their suitability for virtual environments, such as virtual machines and containers, naturally increase their efficiency in utilizing elastic resources. Even more beneficial, though, is that a microservices design enables finely tuned control of compute and processing resources. Processes can be scaled vertically, per microservice, rather than horizontally, which requires launching additional and entire instances of the monolithic application. This means offering a new service, such as launching a channel, can be accomplished much more efficiently and quickly than in the past, since required functionality can be scaled selectively.

Geo-dispersion: What makes the Internet or the cloud such a powerful and hyper-efficient environment is that it comes with specialized orchestration and automation built into it. Microservices-designed applications are built to seamlessly tap into all of the advanced capabilities of a distributed and geo-diverse environment, which include scaling resources up and down as needed, robust reliability and easily setting up redundancy and business continuity scenarios. A microservices approach is the only way to complete in the cloud. The relationship between cloud-native and microservices is explained later in this report.

Continuous Delivery: Microservices design introduces new efficiencies into the software development process by supporting the ability to make frequent modifications, fixes and enhancement, essentially on the fly and in opposition to monolithic application development and maintenance. Version control becomes a far easier and fluid exercise in a microservices-based environment. This benefits media companies by ensuring that the applications powering your operations are always up to date and always at peak performance.

Speed to Market: A related benefit of a design architecture that segments software into separate business capabilities or processes is the ability to quickly incorporate the latest technology or introduce add-on capabilities. Rather than waiting until the next version is released in six months, media companies are able to roll out new features or capabilities as each new microservice is introduced, better protecting your business from falling behind the competitive curve.

Resiliency: Though a cliché, the dangers of putting all of one's eggs into a single basket provides a powerful metaphor for illustrating a key advantage of microservices design over a monolithic approach. With the latter, a single failure can spell catastrophe for your entire operation. A microservices approach, where each egg is isolated in its own basket, better protects the organization from losing service, as well as simplifying and hastening the repair process. It's much easier to quickly spin up a lightweight and flexible process than a brittle, overweight application.

Simplicity and Extensibility: The ability for developers to work on individual processes, rather than requiring familiarity with the entire application improves and speeds development by enabling different teams to work on different portions of the application. Microservices-design principles, which isolate individual processes, also increase the openness of your applications, reducing reliance on the technology vendor for new innovation and easing integration with third-party or other external sources for additional innovation.

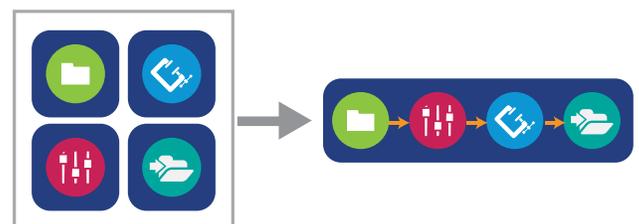
Symmetry: As more and more of your infrastructure moves to a microservices-based architecture, your business will be able to increasingly leverage the benefits of a common code base that spans your entire organization. Repurposing the same software component within different business capabilities, for example, eliminates costs and leverages knowledge and trust of familiar technology. In addition, software commonality facilitates tighter integration among workflows, reducing complexity and improving efficiency by eliminating silos that have traditionally existed within most media companies, particularly between operational and business functions.

Agility: This is essentially a composite benefit, the end result of aggregating all of the above capabilities. Speed to market, rapid innovation, extensibility, geo-dispersion and vertical scaling all add up to an infrastructure that is responsive and reliable enough to enable media companies to keep pace with shifting and evolving consumption patterns, as well as weather the blistering pace of technology innovation.

Monolithic	Microservices
Interdependencies: Processes shared among multiple business capabilities	Autonomy: Business capabilities are isolated into individual processes
Versioning: Developers must update and test entire application with each fix or modification, often reducing release frequency to once or twice yearly	Continuous Innovation: The autonomous nature of microservices enables updates and additions to be made effectively on the fly, promoting rapid absorption of new technologies into workflows
Complex: Applications grow overweight and brittle overtime, making them susceptible to failures and unwieldy to maintain	Simple: By isolating processes, microservices remain flexible and lightweight, increasing resiliency and simplifying future development
Closed: Monolithic nature makes it nearly impossible to open application to third-party programmers or media companies	Open: Business-capability-per-process nature promotes easy integration of third-party technology and allows media companies access to applications through SDK
Cloud-Enabled: At best, non-microservices-based applications can be made accessible from the cloud but lack the capabilities to tap into the full benefits of a distributed environment	Cloud-Native: Microservices provide the building blocks for applications that are optimized to take advantage of the Internet and other IT-oriented cloud environments
Horizontal scaling: Increasing performance requires a new instances of the entire application to be commissioned and deployed	Vertical scaling: Microservices-design principles support vertical scaling of individual business capabilities, reducing the cost, time and resources required to expand an existing or launch a new service

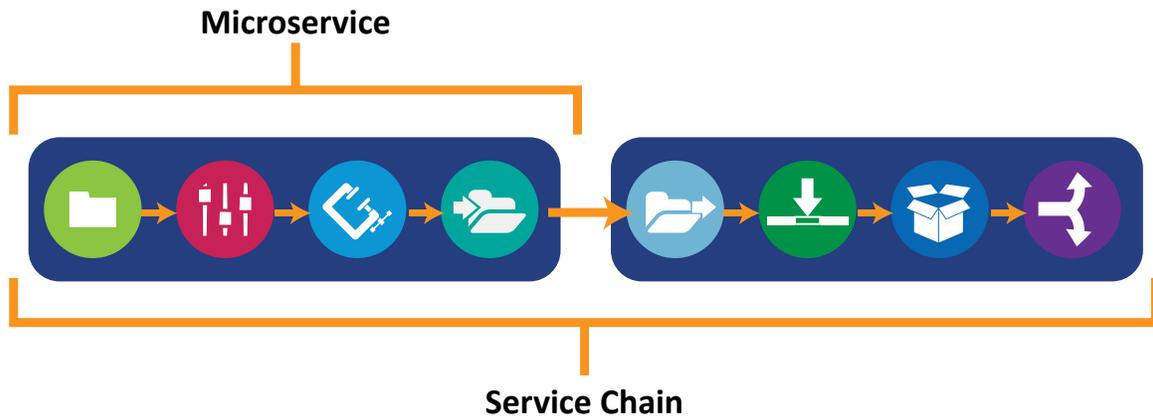
Microservices at Work

The differences between a microservices-based software architecture and a monolithic approach is fairly easy to visualize. In a microservices-based architecture, processes, which often represent business capabilities, are extracted from a common framework and isolated as individual, autonomous processes.



This separation, or isolation, provides media companies with significant flexibility and agility improvement. For example, the traditional procedure for launching additional channels requires spinning up a new monolith for each channel. A microservices design, by comparison, enables the broadcaster to selectively devote resources to specific processes, or microservices. A discrete process, such as a graphic overlay or an AVC encode, for example, could be spun up individually, rather than part of an entire monolith.

Individual microservices can also be linked, or chained, together to create a sequence of business capabilities, which can be preserved as a blueprint for future deployment.



Moving to a Microservices Architecture

Unlike Uber, Airbnb or Netflix, most media companies did not originate in a cloud environment. That means that moving their operations to a microservices-based architecture will require a transition from applications based on a monolithic model. This will not happen overnight and media companies should work with technology suppliers that offer seamless, manageable and multiple transition paths.

One approach is to gradually migrate to a more flexible architecture by surrounding a monolithic design with microservices. The application, as it evolves and is updated over time, will eventually morph from monolithic to a microservices composition. This process, however, could be lengthy, as well as complex.

An alternative approach is to work with a technology supplier that was early to recognize the merits of microservices design and is now delivering the latest versions of its software based on a microservices-design approach. This enables media companies to move to microservices, or cloud-native, operations as part of the natural and scheduled upgrade of their technology infrastructure.

A third migration approach is to begin to move some of your operations to cloud-based environments, adopting a hybrid premises-cloud approach that allows you to take advantage of some of the flexibility and efficiency benefits of a microservices infrastructure, while simultaneously migrating your premises-based operations toward a microservices-based infrastructure.

Cloud-Native & Microservices

Applications that have been constructed using microservices-design principles are considered to be cloud-native. That's because microservices design is the underlying software development architecture of cloud environments, including the Internet. A cloud-native application by nature is constructed to take full advantage of the inherent properties of a cloud environment, including elastic scaling, geo-dispersion and advanced automation. Cloud-native applications are also described as cloud-centric.

The good news is that cloud-native applications, though designed to excel in virtualized environments, are not exclusive to cloud settings. Even media companies that are not ready to move operations to the cloud can still benefit from the adoption of a microservices application architecture. In fact, most of the advantages of microservices-designed applications over monolithic applications are still applicable in an on-premises setting. Agility and flexibility benefits, including the ability to rapidly adopt new technologies and standards, instant and efficient scalability and the ability to speed the introduction of new services, are all accessible to media companies that have yet to move operations to the cloud.

The distinction between a microservices-designed application and a cloud-native one is subtle, at best. That's not the case when it comes to comparing cloud-native to cloud-enabled. Cloud-enabled applications were originally designed to run on purpose-built hardware, most often located in the company's central facilities, but have been retrofitted to be accessible from a public or private cloud environment. The process of cloud-enablement often involves encasing the original application in some sort of IP wrapper. Through this process, cloud-enabled applications are made accessible from the cloud, but lack the ability to tap into traditional cloud characteristics, such as elastic scaling, geo-dispersion and advanced automation and orchestration. In addition, they retain their monolithic nature, severely limiting their responsiveness and ability to rapidly incorporate new features or new technologies.

Regardless of where you are in your journey toward a next-gen infrastructure, knowing the differences between cloud-native and cloud-enabled is fundamental to the future of your business.

Conclusion

The television industry's transition to a digital realm based on standard computing platforms is inevitable — and already underway. The broadcast facility of the not-to-distant future will more closely resemble a datacenter than a traditional broadcast facility. It's imperative that media companies construct this new technology foundation using building blocks that will not put limitations on their future agility, versatility or creativity. Only by moving to a microservices-based design will media companies be able to compete with their peers, as well as a new breed of competitors that are native to cloud environments.

A bold and direct line connects your commitment to embracing a microservices-based technology foundation to your business's ability to truly innovate, speed the delivery of new services and meet the rapidly evolving consumption preferences of your customers. The very future of your business rests upon the building blocks you select to undergird the core of your operations.



Powered by Zenium

Zenium™ is a pure microservices platform for building smarter virtual applications and services. Its open architecture fosters a third-party ecosystem of best-in-class features built on a completely modular platform and enables seamless portability between environments — from appliance to datacenter to cloud. A groundbreaking approach to technical adoption, Zenium enables media organizations to easily create custom solutions, adapt more quickly to market dynamics and deploy new services with unparalleled speed, reliability and flexibility.