

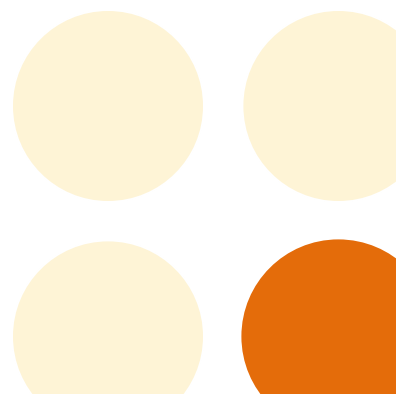


Building Scalable Business Automation with Microservices

December 2019

Kemsley Design Ltd.

kemsleydesign.com
column2.com



Overview

Many organizations are hampered with legacy infrastructure, monolithic application design, proprietary platforms and “untouchable” code, which has resulted from decades of outdated application development practices and over-customization of packaged software solutions. This legacy technical debt is swinging the pendulum from “buy” back to “build” for the automation of core business operations, and driving the trend towards business automation platforms that support rapid software development in-house. Although there are single-vendor business automation platforms available, many enterprises are building their own platforms that support internal application development by distributed teams rather than starting with a monolithic, proprietary product.

This paper compares monolithic and microservices-based business process automation platforms, with use cases for each platform model, plus best practices for migrating from a monolithic architecture to a best-of-breed microservices business automation platform.

About the Author

Sandy Kemsley is an independent analyst, consultant and process architect specializing in digital process automation, business process management and the social enterprise. She writes the popular column2.com blog and is a featured conference speaker on BPM and digital automation. She is a contributing author to books on social BPM and adaptive case management, the winner of the 2016 Marvin L. Manheim award for significant contributions in the field of workflow, and recipient of the 2019 Workflow Management Coalition award for Outstanding Business Transformation Consultant.

Sponsor

This white paper was sponsored by Camunda.

Using technology to survive and thrive

Businesses need two fundamental capabilities from their IT platforms: agility and scalability.

Agility is the ability to respond quickly to changes in the business environment to offer new products or services. This requires the flexibility to swap in new capabilities and business models, and a fast time-to-market to outpace the competition. Technological agility enables disruptive business innovation even within established organizations, with new business models such as personalized products or services, usage-based pricing, and algorithmic services.

Scalability is the ability to increase the throughput of a specific business function in response to increased demand, then decrease it again when demand slows. This requires near-instantaneous (usually automatic) elastic scaling of the IT platforms, and an architecture that avoids the traditional bottlenecks such as shared data access layers that can cause failure when scaling up.

In short, businesses need agility to innovate, and scalability to survive.

Business automation platforms

Businesses that need agile and scalable applications require a business automation platform that they can use to build these applications. Sometimes called a digital transformation or digital process automation

platform, a business automation platform is an application development environment that may be used by both technical developers and citizen developers to develop and deploy applications.

Businesses run on processes and decisions, and those capabilities need to be at the core of the platform, but it's much more than just a business process management system (BPMS). The platform should include:

- Process and decision engines compliant with Business Process Model & Notation (BPMN) and Decision Model & Notation (DMN) standards. This allows for the creation of graphical models that are both understandable by business analysts, and executable to automate processes and decisions.
- Unstructured content management, for content ranging from business documents to multimedia files.
- Event management to handle asynchronous messages between systems.
- Analytics and recommendations to guide the application user through their tasks.
- Machine learning and AI to inform the analytics and automate tasks.
- Security integrated with corporate security and identity access models.
- Application development workbenches for technical and non-technical developers to build applications with traditional coding or low-code tooling.
- Devops capabilities for deploying applications.

Monolith: the enemy of agility and scalability

Given that businesses need their technology to be agile and scalable in order to thrive and survive, it's amazing how many are stuck on monolithic architectures, where all functionality is built into a single large system.

Many organizations still run their core transaction processing on 30-year-old custom mainframe applications that are considered "untouchable" because no one really understands how all the pieces fit together, and they're afraid that any change may have unintended consequences in other parts of the system. Instead, a whole ecosystem of other applications has sprung up around the mainframe applications: to perform functions the core applications can't, to export the data for other uses, or to integrate with modern applications. The same is true for monolithic off-the-shelf software such as enterprise resource planning (ERP) or human resources (HR) systems, which turn out to be anything but off-the-shelf: they often end up highly customized and too brittle to even upgrade the underlying commercial system.

But monolithic architecture isn't limited to older transactional applications: BPMS platforms are expanding into the application development market and attempting to become business automation platforms. These monolithic "iBPMS" platforms include a BPMS at the core, plus many other inseparably coupled capabilities (sometimes hastily assembled via partnerships or acquisitions) including decision

management, data object modeling, analytics, event management, machine learning, user interface and collaboration. Like a legacy monolithic mainframe application, these BPMS-based platforms usually can't replace capabilities – such as swapping out the vendor-provided machine learning services for a best-of-breed competitor – which reduces agility. They typically scale monolithically, and have performance bottlenecks based on the shared BPMS engine and data layer at the core. They often use proprietary low-code development environments that work well for business developers, but hamper the flexibility of technical developers.

Microservices-based platforms

Although there are use cases for monolithic business automation platforms that are purchased from a single vendor, many organizations are moving towards creating their own business automation platforms in-house by assembling best-of-breed components in a loosely-coupled microservices architecture.

A microservice is a discrete unit of business functionality, including business logic, data, configuration and run-time server, all deployed as a single unit. You can develop, test, deploy, upgrade and retire a microservice without redeploying your entire stack or application. You can scale it up and out independently of other components, so that you don't end up with services being scaled just because another service in the same deployment package has to be scaled.

Microservices have only loose coupling, or no coupling at all except through events. If you find a new service that you like better, you can remove the old one and replace it with little or no disruption. If you want to add a new capability, you drop in a new service and have it listen for and emit events to communicate with other services, or wire it into a BPMS orchestration.

A multi-vendor, best-of-breed microservices architecture for a business automation platform makes sense when your core processes are a competitive differentiator. Think back to the earlier points on agility and scalability: do you need to be able to swap out one service for another, or add new services, in order to provide innovation? Do you need functions within the platform to scale independently and elastically when your business volume unexpectedly increases and decreases, to avoid outages and control costs? If you answered “yes” to these questions, you should be considering a microservices architecture. The only downside is that it requires a robust development team to assemble and maintain the platform, and is usually only taken on by large organizations or by tech startups where software development is a core competency.

Although a BPMS is a core component, a fully-capable business automation platform needs to be much more than just an augmented BPMS. To provide best-of-breed capabilities, a business automation platform integrates components from multiple suppliers, with new capabilities being added as the business

requirements and technology evolve. Deploying these capabilities in a microservices architecture provides robust yet efficient scalability that’s just not possible in a monolithic platform.

For technical developers, the platform provides API access to all capabilities when building applications. This may include more standard process orchestration applications, but if the core engines are embeddable, a developer can use the platform to develop a new microservice that has a process or decision engine embedded within it. Technical developers use their familiar development and source code control environments rather than being forced into proprietary tooling.

The platform may also provide a low-code application prototyping and development environment for non-technical developers, as well as access to the modelers provided by the process and decision engines.

Comparing monolithic and microservices architectures

There are several factors to consider when selecting a monolithic BPMS or a microservices architecture for your business automation platform.

Application architecture. This is a somewhat subtle point but quite important: most monolithic BPMS-based platforms assume that end-to-end process orchestration (or a forms-driven milestone-based workflow) is the backbone for every application. In other words,

process is king. There may be no tools for creating an event-driven microservice with an encapsulated process engine, or indeed for creating microservices at all.

Development tooling and software engineering functionality. Robust technical development teams require interfaces with standard tools for versioning and source code control, and with automated testing tools. Many monolithic BPMS platforms are a proprietary walled garden that enforce a low-code development environment on everyone: great for citizen developers, but “death by properties panel” for technical developers. A business automation platform built on a microservices architecture is effectively an API surface for technical developers to use with their existing application development tooling, plus model-driven development tools for specific capabilities such as process or decision modeling.

Extensibility and flexibility. Although either architecture will allow you to add a third-party service to provide a new capability, a monolithic BPMS usually can't un-deploy a service that you don't want: you pay for it (in licensing and infrastructure overhead) even if you're not using it.

Scalability. Monolithic BPMS platforms tend to scale monolithically, requiring that the entire platform be scaled even though only one capability may be under stress. They often have a common process engine with a relational database store for the entire platform, creating a potential critical bottleneck. Microservices architecture, by their

nature as loosely-coupled systems, scale each component independently; the scaling is usually automatic and may also include automatic scaling down when load decreases. Although the applications built on the microservices-based platform could be designed to use a common process engine, they may also take advantage of embeddable engines within applications or services that can scale independently.

Although a microservices architecture has some obvious benefits, it requires greater technical effort and isn't suited to all situations. In particular, small to mid-sized companies (or departments within a larger organization) may use a monolithic BPMS as a business automation platform for non-core processes, or to augment a commercial system such as an ERP or sales automation system. This provides a semi-technical team with a low-code environment to quickly build and deploy applications, and a single vendor to deal with.

Migrating your monolith to microservices

Monoliths are everywhere in the world of business technology: legacy mainframe applications, commercial packaged software and proprietary (including many iBPMS-based) business automation platforms. When your monolithic technology environment starts impacting your organization's ability to survive and thrive due to insufficient agility and scalability, consider migrating some or all of it to a best-of-breed microservices-based business automation platform.

Avoid a “big bang” approach to this migration, using the following best practices:

- Start by isolating and containerizing the monolith(s), and wrapping the service points in an abstraction layer. This is especially useful if the monolith will persist in the medium or long term as a data store or transaction engine.
- Add the new microservices-based platform alongside, and gradually refactor the monolith's functionality by creating microservices to replace individual capabilities, and note that the new microservice endpoints may not match the monolith's APIs directly. Each of these new microservices will include process, decision and data within it, and may invoke the monolith's services while data or transactions are still being processed there.
- Re-think the old method of “single source of truth” centralized data store, and re-architect the data model to allow state data to be distributed across microservices, so that each microservice has the data that it requires, and the services communicate to make the overall system state eventually consistent.

New applications and refactored old applications will use the new service endpoints, whether in the monolith abstraction layer or the new microservices, increasing agility. And as the single shared database is replaced by distributed data, the primary scalability bottleneck is removed.

At some point, even if the monolith is maintained in the short term for its persistent enterprise data repository, a modernized data architecture should be created and the data store migrated. In the case of replacing a monolithic BPMS-based business automation platform, the platform typically does not contain long-running state or persistent enterprise data, although it may contain historical process information for analytical purposes that can be exported to a data warehouse.

Summary

If your organization lacks agility and scalability due to legacy infrastructure, monolithic application design, proprietary platforms and “untouchable” code, implementing a microservices-based business automation platform may be a solution. Building your own business automation platform isn't for everyone, but is especially valuable for large organizations with robust IT teams and core processes that are a competitive differentiator, or for smaller software startups to allow for unexpected expansion.

Migrating from your existing monolith doesn't have to happen all at once: risk can be reduced by implementing the microservices-based platform in parallel with the monolith, then gradually refactoring and replacing the monolith's capabilities.

