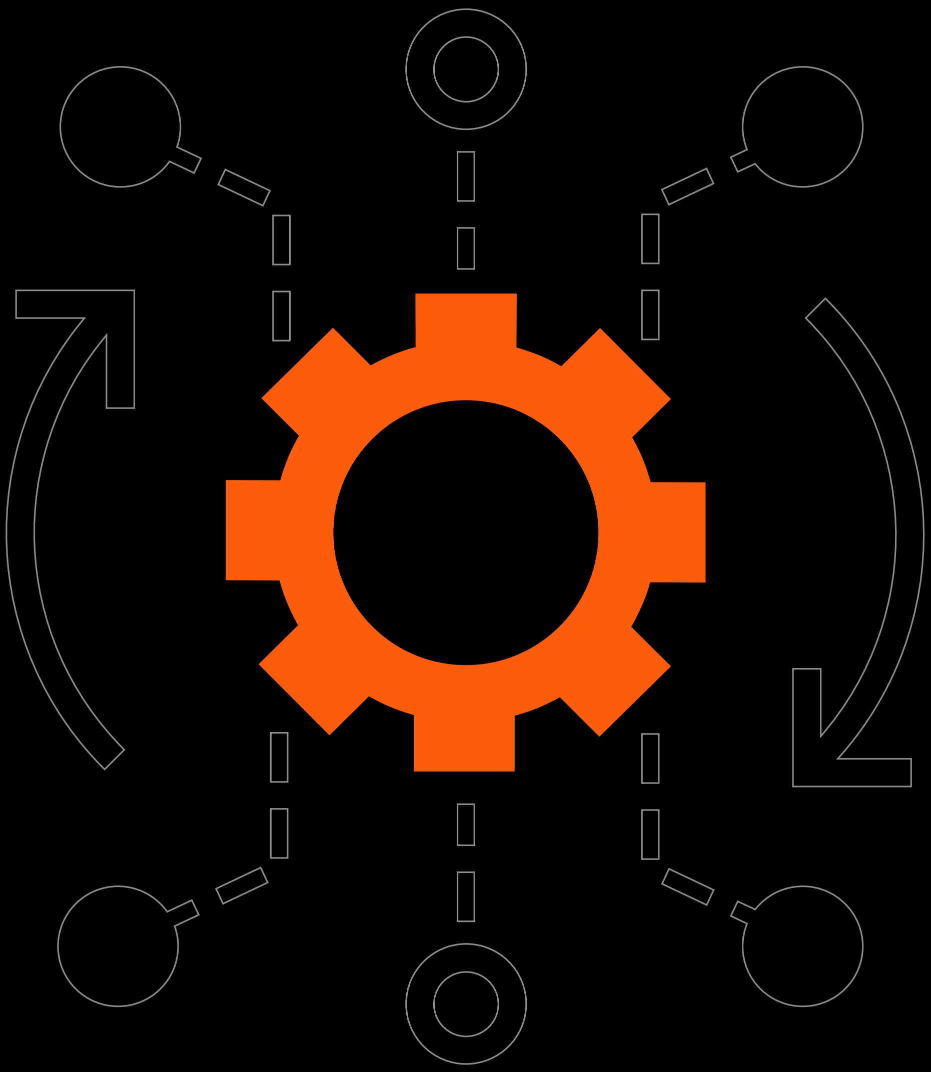
CAMUNDA

Camunda Compared to Alternatives



August 2024

camunda.com

Contents

Introduction: The Need for Process Orchestration

Processes are the algorithms that determine how an organization runs. They define how teams work together, how the organization works with partners and suppliers, and how it delivers value to its customers. More and more, organizations are recognizing that in order to deliver better customer experiences, keep up with competitors, streamline operations, and improve their bottom line, they must automate the processes that are core to their business. This is the digital transformation imperative: embrace automation today or go out of business tomorrow.

Consequently, organizations have adopted a wide variety of tools and technologies to automate every aspect of the business. This includes solutions for customer relationship management (CRM), enterprise resource planning (ERP), robotic process automation (RPA), IT service management (ITSM), and more. Many of these solutions can deliver quick wins for the business because they automate tasks that were previously done manually. However, when they’re pushed beyond their intended capabilities or implemented in a disjointed way, they trap value in business silos. This leads to technical and organizational debt, with a diminishing return on investment. The organization loses sight of processes and leaders don’t have the insights they need to sustain digital transformation.

Process orchestration eliminates silos of trapped value by enabling teams to coordinate the tasks that make up an end-to-end business process in a single process flow. Process orchestration is often compared to the role of a conductor in an orchestra, who makes sure everyone performs at the right time — they tell the musicians when each instrument needs to play to ensure that the song sounds as it should. The process orchestrator is the “conductor” of a process, coordinating and managing the interactions and dependencies of all the tasks in the process, be they manual or automated.

While CRM, ERP, RPA, and ITSM tools have some process automation capabilities, they don’t compare with a dedicated process orchestration solution. This paper analyzes popular categories of automation tools and discusses how well they satisfy the qualities that are required to orchestrate business processes from end to end.

What Matters When Choosing a Process Orchestration Solution

There are many factors to consider when choosing a process orchestration solution. Start by evaluating the complexity and anticipated volume of the business processes that are most critical to automate. If you're looking to automate a single use case, also consider long-term usage and what that might entail. Then, consider potential collaboration points between software developers, low-code developers, and business stakeholders. Many automation tools partially or fully deliver the following capabilities, but a robust process orchestration solution will check all of the boxes.

- **End-to-end process orchestration:** The solution can orchestrate every automated and manual task in an end-to-end business process across endpoints such as APIs, microservices, RPA bots, IoT devices, and AI/ML tools.
- **Support for long-running processes:** It has technical capabilities such as state persistence, which are needed to manage business processes that run for hours, days, weeks, or even longer.
- **Process analytics and optimization:** It provides actionable insights based on real-time and historical process data, with features that help teams identify bottlenecks, analyze performance issues, and continuously improve processes.
- **One model approach:** It uses a consistent process model visualization across design, monitoring, and improvement activities.
- **Standards-based collaboration:** It uses open standards to facilitate communication and alignment between business technologists, low-code developers, professional software developers, and anyone else who is a stakeholder for business processes.
- **Developer friendliness:** It doesn't require software developers to adopt a vendor-specific way of working, but instead meets them in their comfort zone by fitting into both their development toolkit and application architecture.
- **Flexible architecture:** Its components are designed to work together seamlessly but can also be used independently, with easy integration into enterprise tech stacks. It offers on-premise, cloud, and hybrid deployment options.
- **Low total cost of ownership:** Open standards, developer friendliness, and flexible architecture combine to reduce the implementation cost for automation projects and keep ongoing maintenance and improvement costs low.
- **Highly scalable and resilient:** It leverages cloud-native technology to scale process volumes and to provide resilience against unplanned downtime and data loss.

The Automation Landscape

	Process Automation Tools	Monolithic Automation Platforms	Microservices Orchestrators	iPaaS	Task Automation Tools	Camunda
End-to-end process orchestration	🟢	🟡				🟢
Support for long-running processes	🟢	🟢				🟢
Process analytics and optimization	🟢	🟢		🟡		🟢
One model approach	🟡					🟢
Standards-based collaboration	🟢	🟡		🟡	🟡	🟢
Developer friendliness	🟡		🟢			🟢
Flexible architecture	🟡		🟢			🟢
Low total cost of ownership	🟡		🟡			🟢
Highly scalable and resilient			🟢			🟢

🟢 full support   🟡 partial support

Process Automation Tools

When looking to increase the level of automation across important business processes, most organizations adopt a process automation tool. These tools originated in the world of business process management (BPM), which started as a way to model and execute straightforward, repeatable back-office processes. Today, these tools enable teams to model, execute, manage, and analyze processes at all levels of complexity, making them a key enabler of digital transformation.

Process automation tools excel at executing long-running business processes that combine automated tasks with tasks that must be done by people (also known as “human tasks”). They provide operations teams with capabilities to manage processes in production, including features for troubleshooting and resolving problems — for example, editing a process variable that contains corrupted data or migrating a running process to a new version of the process model. Process automation tools are able to provide extensive analytics because they manage all aspects of process execution from start to finish.

Typical use cases for process automation tools include:

- Back-office processes such as accounting reconciliation and invoice processing
- Cross-industry use cases such as customer onboarding and data management
- Industry-specific processes such as loan origination and retail returns processing

Challenges

Inflexible architecture and integration

Process automation tools require robust integration capabilities because they need to trigger various task automation tools to execute the tasks that make up a business process. These tools must also accommodate custom software development work, because it’s overwhelmingly common for complex business processes to require some amount of custom code.

However, many process automation tools don’t provide integration or software development capabilities that are flexible enough to accommodate a wide variety of business use cases. Connectors for popular tools are useful until they need to be customized or extended; if the connector is built in a proprietary way or is simply a “black box” that can’t be changed, then teams have to wait on the tool’s vendor to deliver the functionality their processes require. If the tool requires custom code to be written in a specific programming language — or even a less capable scripting language or a proprietary language — then developers might not be able to use the development standards that have been established for their team. This can lead to a skills gap and the need to outsource process automation work, which can further slow down digital transformation initiatives.

Not highly scalable or resilient

The process automation tools that are on the market today vary in their ability to deliver highly scalable, highly resilient automation. Most tools rely on a central database to store information about the status of running processes — a technical challenge known as “state persistence” or “state handling.” Reading from and writing to a single database results in a performance bottleneck that precludes high-volume use cases such as instant global payments and stock trade matching and settlement. Scaling throughput capacity by adding workflow engines and databases costs computing resources, makes infrastructure operations hard to manage, and introduces significant challenges around process management and reporting.

Monolithic Automation Platforms

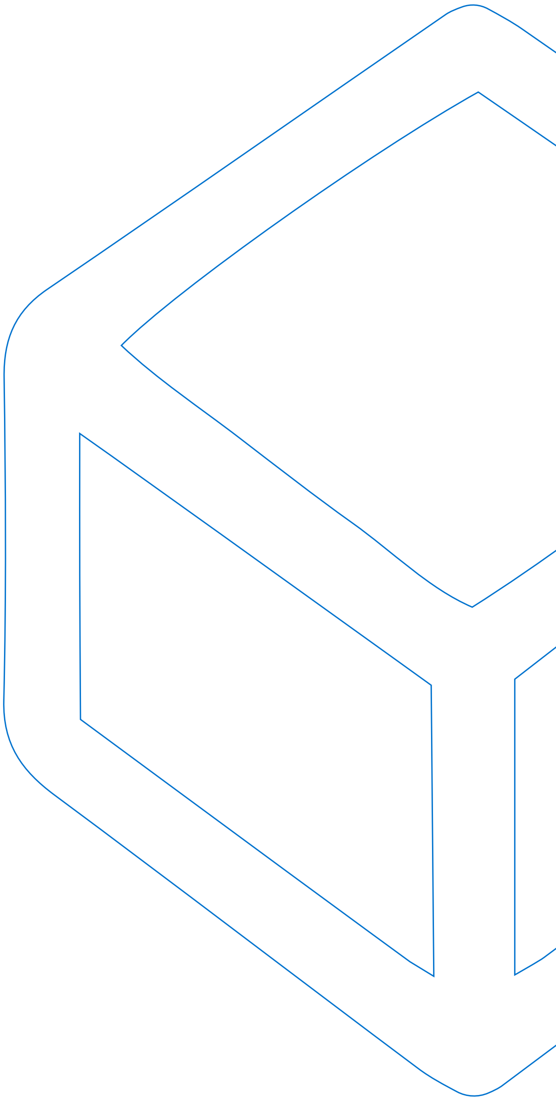
Monolithic automation platforms are sold as all-in-one solutions for designing, executing, managing, and analyzing business processes. They also offer adjacent capabilities such as process discovery, low-code user interface creation, and connectors for popular enterprise tools. The defining quality of these platforms is their architecture; they're built as monoliths, meaning that organizations must implement the whole platform instead of choosing the components that best fit each business and technical use case.

There are three main categories of monolithic automation platform:

Traditional monolithic automation platforms originated in the world of business process management suites (BPMS). They offer similar functionality to process automation tools; they include process design and management features, they support decision automation or business rules, and they often include tools for building forms and other basic user interfaces.

Modern monolithic automation platforms have emerged as vendors add process automation capabilities to products that were originally designed for purposes such as customer relationship management (CRM), IT service management (ITSM), and personal productivity.

Low-code application development platforms (LCAPs) enable people who aren't professional software developers to build applications. These applications often execute a limited number of tasks as part of a larger, end-to-end business process, although some LCAPs do support simple process automation within the platform itself.



Typical use cases for monolithic automation platforms include:

- IT help desk processes
- Sales and marketing campaign automation
- Accounting and finance processes

Challenges

Inflexible architecture

As the term "monolith" implies, these platforms are architected as a single, black-box application as opposed to a collection of components or microservices that can operate independently. This monolithic design has a ripple effect on many aspects of process automation. It increases development effort, increases the risk that the wrong thing is built, and results in a high total cost of ownership.

High development effort

Complex business processes almost always require some amount of software development work to reach a high level of automation. Software developers are most productive when they can write code in the programming languages they know, using the development and testing tools that they already have set up. Most, if not all, automation monoliths require users to use proprietary tools to design and implement processes. This means developers' time is wasted as they have to learn to use these proprietary tools instead of their own development toolset. In addition, developers have to continuously maintain this knowledge, which can lead to a lack of sufficiently skilled team members in the organization.

When working with automation monoliths, technical teams can't see inside the black box to understand exactly how it executes processes, which means software developers are often stuck reverse-engineering automation code. Connectors for automation monoliths tend to be built in a proprietary way, making it hard to customize or extend the way endpoints are integrated into processes.

No standards-based collaboration

Automation monoliths enable collaboration by providing drag-and-drop tools for building processes and decisions (also known as business rules). However, process visualization is only one piece of the puzzle; for a workflow engine to actually execute a process, it must contain code that can run. Some of this code is generated by the automation monolith itself, but it doesn't take much complexity for a business process to require custom code as well.

When processes and decisions are built using proprietary tools, there's little to no transparency for the software developers who need to make the process executable. It also means that processes can't easily be exported from one tool and imported into another. In contrast, open standards such as Business Process Model and Notation (BPMN) and Decision Model and Notation (DMN) are fully transparent and portable, without sacrificing the type of user-friendly process visualization that enables cross-functional collaboration.

High total cost of ownership

Automation monoliths lock organizations into a proprietary process automation design and execution environment. Organizations that adopt automation monoliths require extensive consulting hours from specialists who know how to use the platform. This doesn't just cost time and money up-front; it also creates maintenance headaches when automated processes need to be updated or expanded.

The black-box nature of automation monoliths means they're often sold as packages that contain many automation features that are adjacent to process automation features. This packaging can be a double-edged sword; while it can reduce overall licensing costs, it locks the organization into using the monolith's implementation instead of facilitating the use of best-of-breed tools.

Microservices Orchestrators

Microservices are fine-grained, autonomous, loosely coupled services that are organized around business capabilities and work together to achieve specific business goals. Although microservices operate independently, there's usually an implied business process that requires each service to be invoked at the right time and under the right conditions.

That's where microservice orchestrators come in; they coordinate work across microservices to ensure that the business goal is achieved, without forcing services to become tightly coupled or dependent on one another. They provide a layer of process visualization and execution on top of microservices, making it easier for teams to understand, troubleshoot, modify, and improve business processes without requiring them to compromise on microservices design principles.

Microservices orchestrators can be used for any use case that invokes multiple services to achieve a business outcome. They're suited to environments where all monolithic business applications have been broken down into discrete services, and the way those services must work together is well-understood. However, they're normally used by software development or IT teams — not by the business.

Challenges

Orchestration for microservices only

Microservice orchestrators are purpose-built to address the challenges of coordinating and managing many autonomous services, especially in high throughput scenarios that require dynamic scaling of resources. However, it's very rare for all of an organization's business processes to be made up of microservices and only microservices. Most organizations have an IT landscape that contains monolithic business applications, legacy systems, multiple user-facing front-end applications, and more.

Human work is another factor. For example, imagine a microservices-powered loan origination process that requires a lending officer to manually review the documentation for applications who have multiple financial risk factors. To achieve complete visibility for that process, this manual work and all automated work must be represented in a single process flow.

Limited collaboration features for different types of users

The nature of microservices — including the fact that they operate in a loosely coupled way — means that microservices orchestrators are heavily geared toward the needs of technical users such as software developers and IT operations staff. While many microservices orchestrators use open standards and modern, widely adopted protocols, they don't offer many features that enable collaboration between the teams that build and deploy microservices and the subject matter experts on the business side of the organization.

For example, microservices orchestrators usually lack a graphical representation of the process model that business users can view and modify on their own. If the orchestrator does have a graphical editor for process models, it often doesn't have the same graphical representation for process monitoring or reporting. If there is a visualization, it reflects code, and can't be modified without editing code. This makes it hard for business users to be involved beyond process design into process management and improvement. It also means that IT resources are required to troubleshoot issues with running processes.

Integration Platforms as a Service (iPaaS)

Integration Platforms as a Service (iPaaS) address the need to move, copy, synchronize, and manipulate data as it flows between many different systems. They act as "glue" between systems across the enterprise, enabling teams to integrate technologies on a point-to-point basis. iPaaS tools offer pre-built connectors for a wide variety of software products, along with capabilities for transforming and mapping data as it passes between systems.

Although iPaaS tools focus on point-to-point integration, many of them support some level of process modeling and execution to enable teams to visualize the flow of data between systems. Rarely, iPaaS tools support the Business Process Modeling and Notation (BPMN) standard for process design.

Typical use cases for iPaaS include:

- Synchronizing data between enterprise resource planning (ERP) and customer relationship management (CRM) systems
- Connecting ecommerce software to accounting platforms
- Processing large amounts of data from Internet of Things (IoT) devices

Challenges

Lack of end-to-end process orchestration

iPaaS tools are primarily focused on cross-system integration and data management rather than process automation, which prevents them from orchestrating business processes from end to end. For example, they offer little to no support for processes that combine automated tasks with work that has to be done by people (often called "human tasks"). They also lack support for sophisticated workflow and process execution patterns, such as an external task pattern where workers retrieve ("pull") automated tasks from the workflow engine instead of waiting for the engine to push tasks to them.

High total cost of ownership

iPaaS vendors often charge based on resource consumption — for example, number of connectors used, number of integrations deployed, or amount of data that the tool processes. Consumption-based pricing can be advantageous for small teams or organizations that have a limited and predictable need for iPaaS functionality. But consumption-based pricing doesn't scale well for large organizations or for new projects with unpredictable throughput. The need to strictly control pricing can prevent organizations from rolling out iPaaS capabilities to all teams because of the fear that they'll quickly lose control of iPaaS spend. In cases where pricing is negotiated beforehand, iPaaS can still result in a high total cost of ownership; for example, in situations where extensive training and consulting is needed to get automation projects into production.

Task Automation Tools

Task automation, also known as local automation, is the use of software to automatically perform certain tasks without human intervention. Task automation is necessary to automate the many individual tasks that make up an end-to-end business process. Process orchestration coordinates the execution of tasks throughout the process flow. When an organization combines task automation with process orchestration, they can achieve process automation.

Point solutions automate a specific type of task. For example, intelligent document processing (IDP) tools can read data from PDFs and other digital document formats. Workload automation tools, also known as service orchestration and automation platforms (SOAPs), can provision and manage infrastructure resources (such as Kubernetes clusters).

Robotic process automation (RPA) tools implement a specific type of task automation: they use screen-scraping to simulate mouse and keyboard actions in order to automate tasks within legacy systems that don't have an API. RPA tools are a popular choice for low-code developers and business technologists who own processes that require legacy systems.

Typical use cases for point solutions and RPA tools include:

- Processes that require interaction with mainframe systems
- Document-driven processes such as invoice processing
- Email, chat, and SMS automation
- Cloud infrastructure automation

Process Orchestration and AI

Artificial intelligence (AI) and machine learning (ML) offer a wide variety of possibilities for enhancing process automation and orchestration. Here are just a few.

AI-powered copilots help boost productivity for both business and IT users by intelligently suggesting actions they can take when building processes. A copilot provides a fast-track to orchestrated processes by speeding up the design phase and providing immediate feedback to users, preventing process design errors that might otherwise only be detected in production.

Intelligent processes help teams get the most out of AI/ML tools and services by providing the toolset to orchestrate them in the same way as other process endpoints (including human work). In particular, intelligent processes enable teams to orchestrate generative and predictive AI functionality in a combined process. Operationalizing AI in this way helps organizations build in critical governance for the use of AI and data.

Autonomous process intelligence uses AI/ML to uncover hidden value in an organization's most critical end-to-end business processes. It identifies opportunities for increased automation and drives continuous improvement of automated processes.

Learn More

Challenges

Lack of end-to-end process orchestration

Task automation solutions are designed to automate one piece of work at a time, so the concept of "process" isn't built into them. That is, the solution isn't aware of tasks that have already been executed and it can't make a decision about what task should be executed next.

Some task automation tools can trigger a next action — a capability that teams can leverage to chain together a series of automated tasks. However, chaining automated tasks is incredibly limited when compared to true end-to-end process orchestration. It doesn't make the process visible, it can't be used to implement even minimally complex processes, and it doesn't lead to the process-level insights that help teams evaluate and improve process performance.

No support for long-running processes

When looking at a process from the business' point of view — that is, where business owners consider the process to start and end — it's common to have business-critical processes that run for hours, days, weeks, or even longer. Imagine a mortgage application; it takes some time to complete, involves processing large amounts of sensitive data, and is strictly regulated by government entities. Keeping track of the status of a mortgage application, along with all relevant data and an audit trail, is a non-negotiable business requirement for a financial institution.

Keeping track of status is also a significant technical challenge known as "state persistence" or "state management." State information has to be stored somewhere, whether in a central database (as traditional process automation tools do) or in a modern, event-source way (as Camunda's workflow engine, Zeebe, does). Point solutions and RPA tools don't have the capability to store this information or to access it at the right time — for example, when the mortgage applicant signs the closing paperwork for their new house. Without the ability to persist state, these tools cannot implement long-running business processes.

Not highly scalable or resilient

Point solutions and RPA tools are designed to execute one task at a time, which poses issues as process volume increases. To increase task execution capacity, teams have to add more instances of a tool or, in the case of RPA, more bots. This can lead to rapidly escalating operations costs and an implementation that's brittle and prone to failure. RPA bots in particular can easily break when the systems they interface with change.

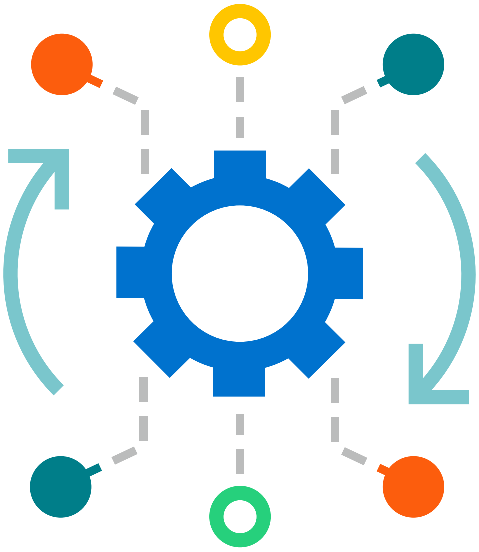
While scalability and resilience are challenges for task automation, organizations can use process orchestration to mitigate the impact of task slowdowns and failures. A robust process orchestration solution supports processes that have built-in timers, timeouts, and fallback flows, which ensure the process isn't stuck waiting for a response from a point tool. This can ensure that the customer still has a good experience, even if every task wasn't executed as intended.

Transformational Process Orchestration with Camunda

End-to-end process orchestration

Most organizations use hundreds or even thousands of off-the-shelf and homegrown applications to execute core business processes. Therefore, when mapping a complete business process from start to finish, it's likely that the process spans multiple systems or services, fragmenting the process into different parts that are executed in isolation. This fragmentation causes a lack of visibility, integration, and control of the end-to-end process, slowing down or even preventing effective troubleshooting, reporting, and analysis.

Camunda orchestrates both automated and manual tasks across end-to-end business processes, no matter how many applications or systems are involved.



Support for long-running processes

Many organizations have business processes that can run for hours, days, weeks, or even longer. Long-running processes present a variety of technical challenges, such as tracking the state of a process, correlating all activities and data related to the process, and triggering timeouts. Also, long-running processes often lead to additional business requirements; for example, if the payment for an online order fails, the customer might be allowed a certain number of days to retry with a different payment method.

Camunda process orchestration is based on the ISO-standard Business Process Model and Notation (BPMN), which allows teams to design processes that are both graphical and executable. BPMN has built-in support for long-running processes; it automatically handles technical challenges such as state persistence, data correlation, and timeouts. BPMN also facilitates monitoring, alerting, and reporting on the process level, which are key features for managing long-running processes.

Process analytics and optimization

Most automation tools provide some level of reporting about the data they collect, often including dashboards with graphs that visualize data. However, these reports are limited to the tasks that the tool executes itself; they lack the context of the end-to-end business process that invoked those tasks, which means teams have an incomplete picture of process performance. This incomplete picture makes it difficult or impossible to identify bottlenecks and other areas where the organization can take action to improve processes.

Camunda orchestrates end-to-end business processes, and thus can access a 360-degree view of process execution data. This data results in a deep set of analytics, complete with intuitive visualizations and heatmaps that are useful for both technical and business stakeholders.

One model approach

Camunda's "one model" approach means users work with a consistent process model visualization across design, monitoring, and improvement activities. At design time, business and IT users use BPMN to build visual representations of complex business processes that are executed directly by Camunda's workflow engine, Zeebe. When monitoring running processes, users see information about process status and incidents overlaid on the same model, so they don't have to interpret technical performance data or decipher server logs. Reports show historical process execution data with the same visualization, including heatmaps that provide an intuitive way to understand process performance.

The one model approach enables business users to be involved in the entire process lifecycle, instead of being limited to collaboration during design time. It also eases cross-organization information sharing and reporting to regulatory organizations.

Standards-based collaboration

Bridging the gap between IT and the business is a challenge for every organization, especially with process automation playing a key role in digital transformation. Business and IT stakeholders often have different goals, incentives, and priorities, and these differences tend to slow down communication, prevent alignment on project priorities, and cause implementation errors.

Camunda uses the globally recognized standards of Business Process Model and Notation (BPMN) and Decision Model and Notation (DMN) to bridge the gap between IT and the business. Standards are a common language that all stakeholders can speak, so nothing is lost in translation between business requirements and technical implementation. BPMN and DMN allow business users to create visual process diagrams and decision tables, while also allowing technical users to round out the technical implementation of automated processes and decisions by editing the underlying code.

Developer friendliness

Many automation tools, particularly automation monoliths, take a vendor-specific approach to application development with the goal of minimizing the amount of code that needs to be written. However, core business processes are complex and require bespoke solutions, so organizations run into trouble as soon as they need to implement requirements that are outside the realm of what their automation tool supports. To work around technical and functional limitations, software developers have to learn the vendor-specific way of automating tasks, which takes time and can lead to implementations that aren't optimized for performance, maintenance, or stability.

Camunda is designed with developers in mind, so they can quickly start automating processes without learning a vendor-specific development framework or being forced to use a proprietary IDE. Teams can use the type of environment they're familiar with when creating, testing, and operating the applications they develop. For example, they can work in their preferred code editor, program in the language they like, store their code in a version control system, automatically test their code, implement continuous integration, and manage their containerized applications on a platform such as Kubernetes. There's no need to adjust to a Camunda-specific way of working.

Flexible architecture

Monolithic automation platforms — and even some process automation and iPaaS tools — are provided as a tightly integrated set of tools or components that cannot be unbundled. They promise that organizations can use the tool to do everything needed to automate business processes. In reality, this closed architecture approach locks the organization into the vendor's product, reduces deployment options, and hinders integration with other IT systems.

Organizations need the full flexibility to control each part of their automation technology stack. Camunda offers the best of both worlds: loosely coupled components that fully integrate with one another, yet are designed to integrate seamlessly into an existing technical architecture to create a composable solution. For additional flexibility, Camunda components can be deployed to on-premises infrastructure, to public or private clouds, or in a hybrid configuration. Camunda also offers a hosted SaaS option that provides fast, massive scalability for high-volume, high-performance use cases.

Low total cost of ownership

Many factors contribute to a product's total cost of ownership: licensing, training courses, the initial development period before moving to production, the turnaround time for changes and new developments, hiring consultants for projects that internal teams can't complete, the cost of the infrastructure needed to run the product, and so on. Many automation tools have a reputation for high total cost of ownership due to long ramp-up times (sometimes measured in years), ongoing consulting fees, and high infrastructure costs.

Camunda approaches total cost of ownership from several angles:

- Camunda's developer-friendly approach and use of open standards makes it easy to get started and reduces time-to-value because the IT team doesn't have to spend time learning a vendor-specific development framework or proprietary development tools.
- Camunda's open architecture allows teams to choose which components to use, so they can easily integrate Camunda into the organization's existing tech stack.
- Camunda is a lightweight solution that requires few infrastructure resources, and that can run on premises or in a public, private, or hybrid cloud.

The Process Orchestration Maturity Model

Over-investing in automation tooling without a process orchestration strategy creates compounding technical debt, trapped automation value, and a diminishing return on investment. While the business might initially benefit from quick wins that these tools deliver, it still takes a lot of effort and IT resources to integrate a variety of tools into end-to-end business processes.

Process orchestration is key to avoiding technical debt from disjointed automation tooling. Organizations can use the Process Orchestration Maturity Model to analyze their ability to execute a process orchestration strategy across a number of different success factors. Improving maturity helps teams overcome the technology and people challenges that stand in the way of meeting the business' automation goals.

Explore the Model

Highly scalable and resilient

Camunda's next-generation workflow engine, Zeebe, is designed to accommodate high-throughput use cases out of the box. While traditional workflow engines rely on a central database that causes a performance bottleneck, Zeebe uses event-streaming technology instead, enabling it to deliver unparalleled scalability and performance. It maintains the state of running process instances in a way that scales up with high transaction volumes, that's resilient to failures, and that performs well at scale.

In addition, Zeebe has a distributed architecture that ensures resilience, even when load is high. This distributed architecture is ideal for geo-replication that further protects data. Zeebe distributes data across all brokers in a cluster with storage directly on the server filesystem, so if one broker goes down — or if a whole datacenter goes down — another can replace it with no data loss.

Conclusion

Processes are the algorithms that determine how an organization runs and, now more than ever, automating them is necessary to deliver better customer experiences, keep up with competitors, streamline operations, and improve your bottom line. At Camunda, we take a modern approach to process automation that enables your organization to:

- **Deliver better customer experiences** by ensuring that all aspects of an end-to-end business process are orchestrated across different systems and endpoints
- **Get to value and achieve business goals faster** by delivering automation projects faster while relying on an open architecture as a foundation for future growth
- **Increase business agility** and rapidly respond to changes in your market by continuously improving processes through standards-based business-IT collaboration
- **Drive operational cost savings** by automating manual tasks, minimizing errors, and optimizing processes with deep insights into process performance

Get a Demo

About Camunda

Camunda enables organizations to orchestrate processes across people, systems, and devices to continuously overcome complexity and increase efficiency. A common visual language enables seamless collaboration between business and IT teams to design, automate, and improve end-to-end processes with the required speed, scale, and resilience to remain competitive. Hundreds of enterprises such as Atlassian, ING, and Vodafone orchestrate business-critical processes with Camunda to accelerate digital transformation. To learn more visit **camunda.com**.