

The Process Orchestration Handbook

Introduction

Many businesses aspire to achieve true end-to-end automation, where processes run seamlessly across people, systems, and devices. When you're building from scratch as a startup, this level of process automation is easier to achieve than if you are contending with legacy systems.

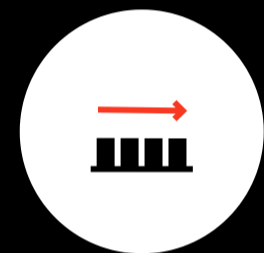
However, for most companies, dealing with legacy systems is the reality. A [survey](#) from Tata Consultancy Services and AWS found that **70% of global companies list legacy modernization as a top strategic business priority**. But, ripping and replacing these systems is disruptive and costly. Beyond legacy systems alone, the quantity or diversity of process endpoints, process complexity, and working in silos can all contribute to challenges with broken or ineffective automation. That's where process orchestration comes in.

Process orchestration coordinates the various moving parts (or endpoints) of a business process, and sometimes even ties multiple processes together. Process orchestration helps you work with the people, systems, and devices you already have – while achieving even the most ambitious goals around **end-to-end process automation**.



Aspiration

Industry-leading levels of automation where processes run seamlessly



Reality

Dealing with a complex mix of systems can cause setbacks and stagnation

In this guide, we will:

- Define process orchestration and why it's important
- Help you understand what you need for successful process orchestration
- Walk through some real-world examples
- Discuss how a universal process orchestrator works

What is process orchestration [and why automation breaks without it]

Today, automation is a business imperative: **92% of IT professionals describe process automation as vital to digital transformation**. Many organizations strive to automate their processes as much as possible. While this goal is ambitious, it's far from reality.

That's because processes are often automated locally, or within a single software system, team, or device. However, most processes – even if they sound simple – follow a more complex logic.

Let's look at an ecommerce process as an example.

Kate orders some tile online from a flooring retailer for her bathroom renovation. While she browses, she confirms that the tile she selects is available at her local store for same-day pickup. So, she selects the "Pick up in store" option at checkout on the website. She receives an email saying that her tile will be ready to pick up within two hours. Before driving to the store a few hours later, she calls to check that her tile is ready, since she never got an email or text confirmation that her order was available. The customer service representative checks the order fulfillment system and tells her that she is actually fifth in line for in-store pickup, and that her order won't be ready until mid-day tomorrow.

As you can see in this example, Kate's customer experience was frustrating, due to delays caused by broken end-to-end process automation. True end-to-end processes span many people (often working in different teams or departments), software systems (e.g. legacy systems, SaaS, homegrown software), and physical devices. Each of these contribute work to fulfill the process and can be thought of as a "process endpoint."

Process orchestration coordinates all of these process endpoints, and sometimes even ties multiple processes together. Without process orchestration, you have a disconnected set of local tasks and automations, which leads to challenges such as:



It's critical to overcome these challenges to increase the degree of automation in your organization. Process orchestration is what you need to make that happen.

Broken end-to-end automation

Since local automations are not integrated with one another, the end-to-end process is not fully automated (like the ecommerce example).

Lack of understanding

The end-to-end process is not fully visible and key metrics are hard to track.

Lack of flexibility

Changing the end-to-end process is difficult since it leads to potential changes in many different systems.

So - when do you need process orchestration?

Simply put, you need process orchestration software when your process has:

Endpoint diversity

Spans across a number of possible process endpoints (people, systems, devices)

Process complexity

Is described by a more complex logic than just a simple sequence of steps.

Endpoint diversity

Every process consists of tasks. In order to execute the process, you need to coordinate the execution of its tasks. Process orchestration software executes the process, but does not execute the tasks. Instead, tasks are executed by endpoints, which can be people, software systems, or physical devices.



People



Systems



Devices

The more diverse those endpoints are, the more you need powerful process orchestration software that lets you integrate those endpoints in a fast, flexible way. For example, people may be working with different front-end devices, or different applications that execute their tasks when invoked via an API.



MORE DIVERSE ENDPOINTS



GREATER NEED FOR PROCESS ORCHESTRATION

Process complexity

Remember how we said that to execute a process, you need to coordinate the execution of its tasks? That point is relevant to process complexity, as well. This coordination is based on a certain logic, which in the real world is rarely just a straightforward sequence of steps.

Process logic is typically a lot more complex and needs to be described by **advanced workflow patterns**, which are explained in the next section.

Dealing with process complexity comes with two major challenges:

- **Alignment:** Designing, operating, and improving complex end-to-end processes requires collaboration between technical and non-technical stakeholders. That's difficult to achieve.
- **Scalability and Resilience:** Executing processes across endpoints needs to happen in a reliable, scalable way. For example, an ecommerce vendor may experience a spike in orders online that requires processing across systems/people. An orchestrator needs to handle transaction spikes. Or, it must offer high availability; for example, by letting the orchestration logic run in a redundant deployment across multiple data centers.

Advanced workflow patterns

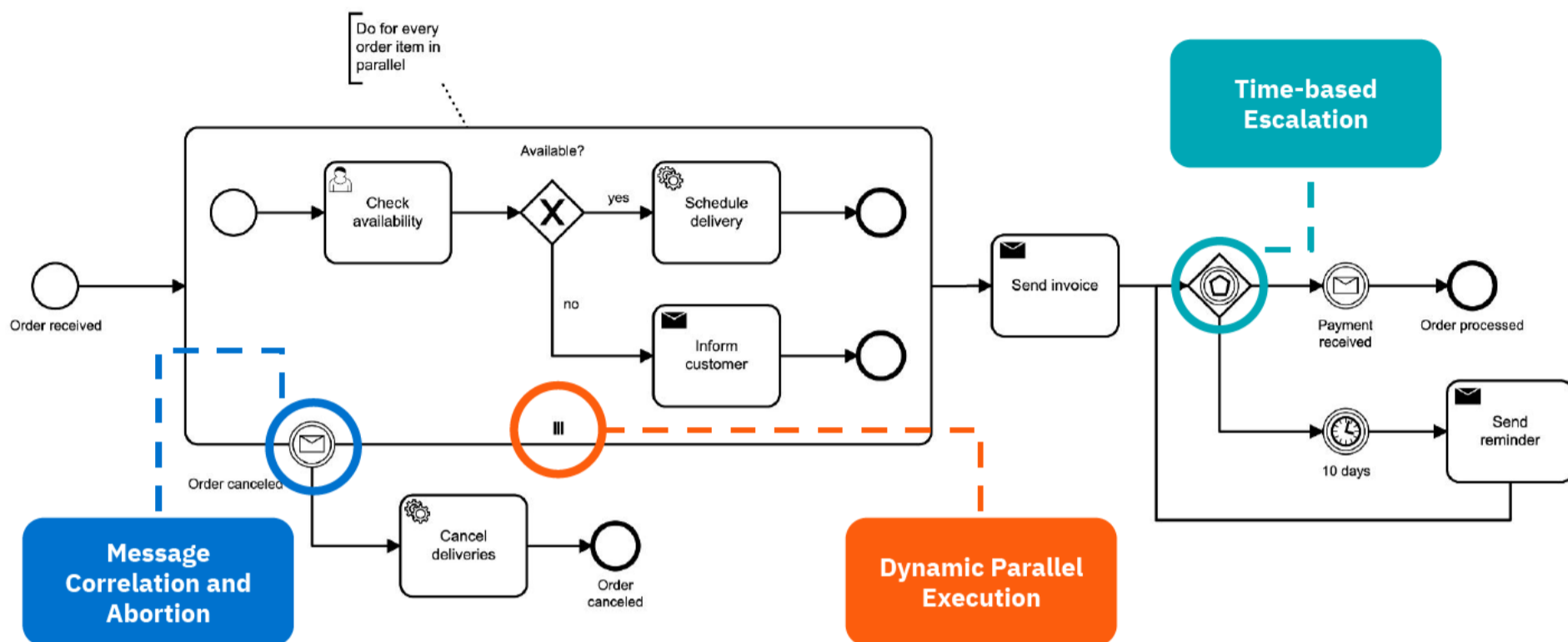
A look inside real-life business processes

Many software products claim that they support end-to-end process automation, whether they focus on RPA, iPaaS, or microservices orchestration. However, the process logic they

implement is typically reduced to basic workflow patterns: Steps in a sequence, branches for if/then-based processes, or parallel execution of steps.

This might be sufficient for local task automation (such as the work that an application user does in the user interface, which can be automated by RPA) or simple integration flows (to move data from

one application to another using an iPaaS tool.) But, as we said, most processes are not that simple, and require advanced workflow patterns.



Example of advanced workflow patterns modeled in the BPMN standard

Most of these patterns involve reacting to events or handling complex business process logic across multiple endpoints. Process modeling standards like **BPMN** were created to design and execute these advanced workflows. Modeling tools that support BPMN enable a much more effective collaboration between the different stakeholders of a process, such as business users and developers. In addition, execution tools that support BPMN ensure a reliable, scalable orchestration of the end-to-end process, despite its complexity.

Let's take a look at some real-world examples of advanced workflow patterns.



Dynamic parallel execution

Process orchestration solves:

Dynamically coordinating many tasks at the same time

Example: Similar to Kate's situation above, a flooring retailer has 10 customers place "pick up in store" orders through its ecommerce system. The retailer then needs to process 10 order positions to be picked up in stock and removed from the online inventory.



Message correlation + abortion

Process orchestration solves: Lack of message correlation

Example: A customer attempts troubleshooting online but calls into the contact center. The customer becomes frustrated if the agent doesn't know what they've already attempted online (e.g. canceling an order.) This company needs a customer number or unique identifier that can correlate events



Time-based escalation

Process orchestration solves:

Escalating processes that aren't completed within a specific window of time

Example: A professional services firm issues an invoice to a customer, and the invoice is not paid on time. An accounting software system prompts the customer to pay the invoice with an automated email coming from the

These processes must happen in parallel, which requires being able to dynamically coordinate 1-5000 tasks at runtime (when the order comes in).

online with the customer's phone call. While this sounds trivial, this correlation is technically challenging to do properly – especially at scale.

service provider's business email address. In this case, the process is coordinated across both the firm's business email and accounting system.

The bigger picture

Process orchestration and the automation market

Your need for process orchestration technology depends on how much process complexity and endpoint diversity you have.

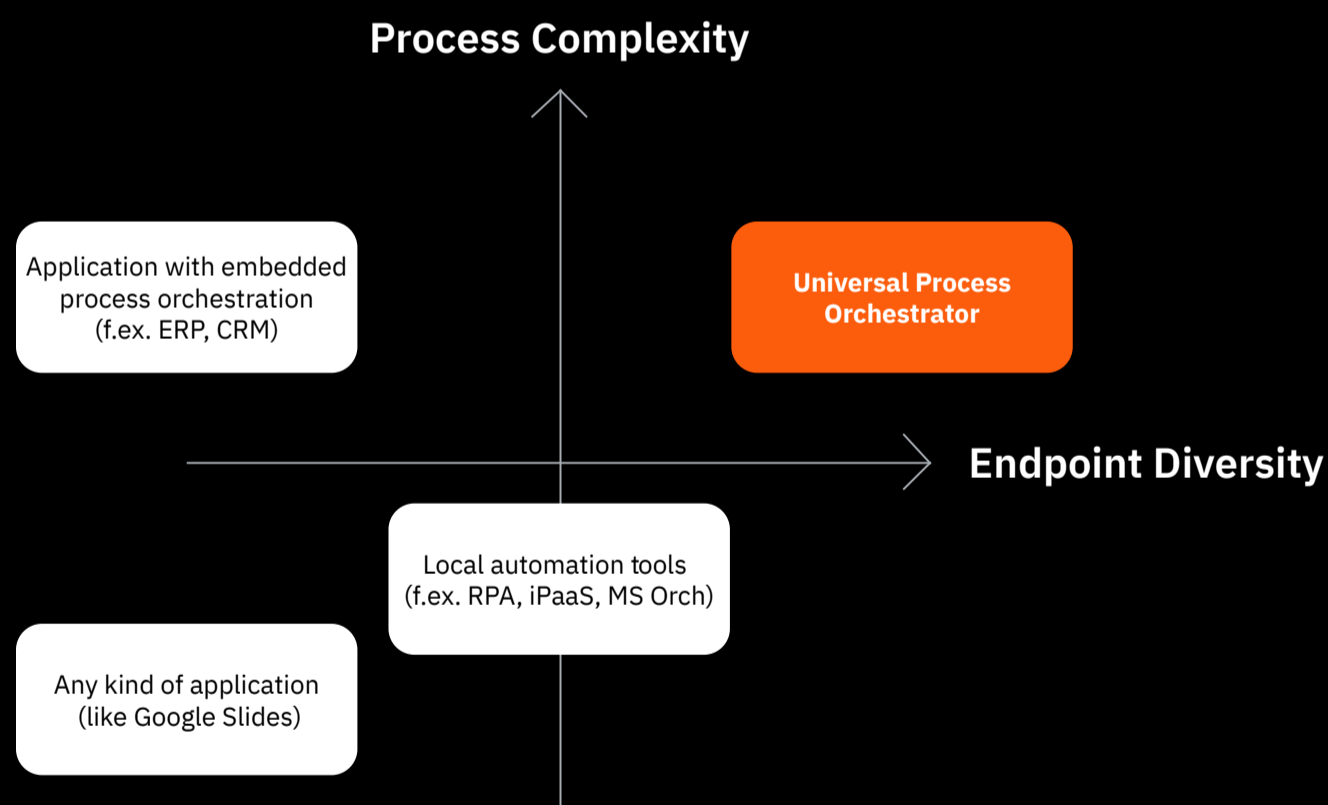
Often, the decision to use process orchestration technology isn't binary. Process orchestration is usually combined with other technologies to get complete coverage of your organization's automation needs. For example, some of the local automation tools (e.g. RPA products) can actually be used in combination with a true process orchestrator.

Upper Left

You may have a lot of processes all happening within one software product (either one you've purchased or built in-house), like a CRM, ERP, or HRIS. That means there may be an embedded process orchestrator moving things along.

Upper Right

The more processes that are happening across different endpoints (e.g. spanning over your CRM and ERP system), and the more they reflect your actual end-to-end process (and come with a certain complexity), the more you require a standalone process orchestrator.



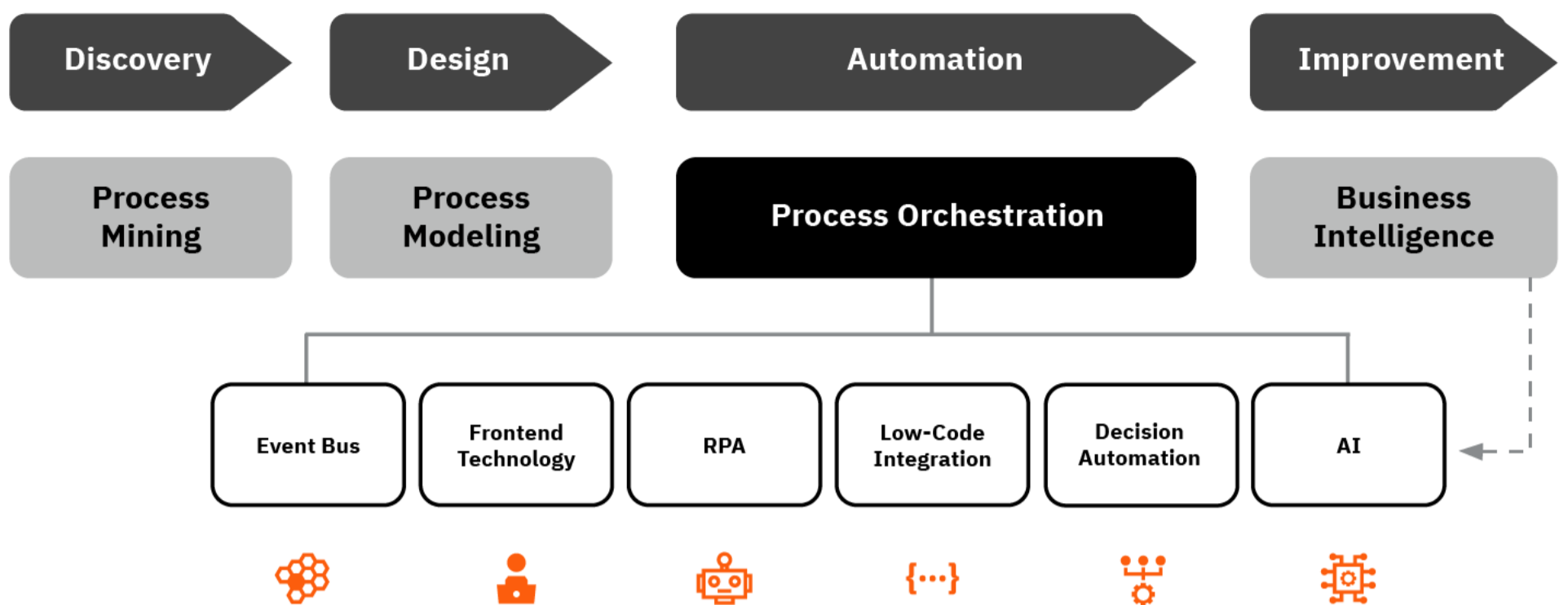
Lower Left

If you don't have a lot of processes or endpoint diversity, a single piece of software might be sufficient.

Lower Right

If your processes span different endpoints of the same type (e.g. only legacy systems, API endpoints, or microservices), and if they are not really end-to-end but more about simple, local procedures – then an RPA, or iPaaS, or microservices orchestrator could be sufficient.

The Hyperautomation Tech Stack

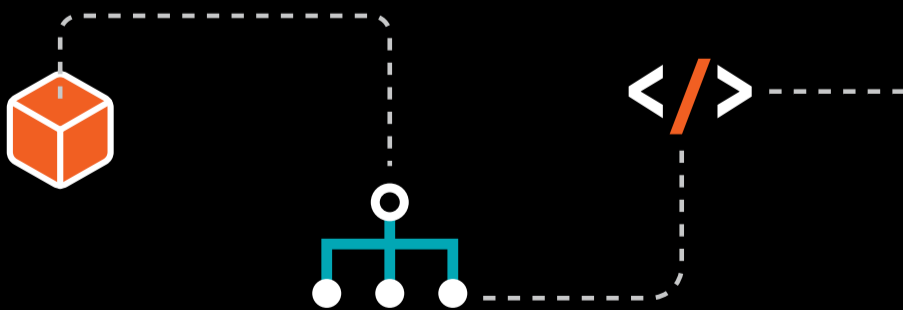


Read more: [What's in your hyperautomation tech stack?](#)

How Camunda Platform works

Camunda has helped organizations all over the globe collaborate on and transform complex business processes that span people, systems, and devices into seamless customer experiences, at scale.

Camunda Platform is designed to enable universal connectivity, allowing for seamless integrations with all of the endpoints that make up end-to-end business processes – from microservices to legacy systems, and everything in between. This is accomplished via a connectivity ecosystem, which includes out-of-the-box connectors that ship with the platform, as well as an integration framework that accelerates the creation of connectors – all of which can then be reused in a low-code modeling environment.



Business + IT collaboration

Business and IT collaborate via an intuitive visual Modeler that adheres to BPMN and DMN standards, so teams can easily model the complexities of modern business processes. Modeling with Camunda Platform reduces the time it takes to develop and maintain real-world business processes, so your business can react and adapt without costly delays.

Once teams have collaborated to design processes, those processes can be deployed for execution via Zeebe, the cloud-native workflow engine that powers Camunda Platform. Zeebe ensures processes run quickly and reliably by allowing for unlimited horizontal scalability, consistent low latency, and a failover architecture that provides rock-solid availability for mission-critical processes.

Knowing where to start is as simple as choosing a small, pilot project as a proof of concept (PoC) that can demonstrate the value of end-to-end process orchestration to key stakeholders. That starts with a small group of developers who are responsible for driving the PoC. Choose a project that can be completed within the span of a month, maximum, yet can have a positive impact on a high-visibility internal process or customer experience.

Gaining alignment and integrating with your tech stack

Using the BPMN and DMN standards with **Camunda Modeler**, the PoC team can design and model workflows in a flowchart-style diagram, which can be easily understood by both IT and business stakeholders to gain alignment. Cloud modeling allows for easy collaboration between stakeholders. The process model includes the people, systems and devices that are most critical to a specific process. From there, the PoC can be launched, completed and evaluated within Camunda Platform.

It's easy to integrate Camunda Platform with the technology you have. Camunda provides out-of-the-box connectors that make it easy to integrate popular endpoints into process models, plus a developer-friendly integration framework so teams can build connectors for legacy systems and speciality software.

From there, Camunda Platform can orchestrate anything from end-to-end, including legacy systems, microservices, RPA bots, APIs, AI/ML tools, IoT devices, and more. The process flows are based on the model you created in BPMN.

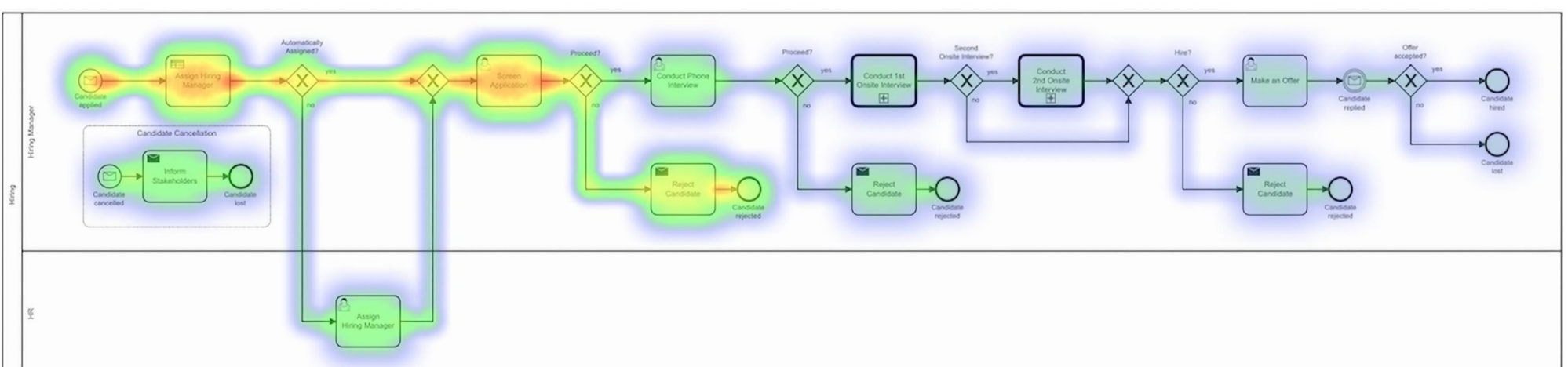
Executing process orchestration at scale

Powered by the **cloud-native workflow engine Zeebe**, you can automate processes at scale. Since Zeebe doesn't depend on an external database, there are no performance bottlenecks when it comes to your processes. It's easy to distribute processing across a cluster of machines to deliver high throughput. That means you can process a high volume of transactions at low latency. Zeebe also insures against downtime and data loss, and can operate at enterprise-grade resilience.

Continuously improving your processes

To learn more about how your processes are working and **optimize process flow**, our platform allows you to view heat maps that visualize process steps.

Camunda has powerful reporting, dashboards and data filtering capabilities that help IT teams and business stakeholders view the data that's most important. In addition, utilize fully customizable alerts that help stakeholders stay on top of what's happening in real time.



Example of a process heat map in Camunda Platform

Final thoughts

To sum up, process orchestration technology makes the most sense when your business processes span a diverse set of process endpoints, and are described by a more complex logic than just a simple sequence of steps. With that said, complex processes don't require complex technology. A universal process orchestrator like Camunda can be both powerful and developer-friendly.

Taking advantage of process orchestration can help your team make the most out of automation – and achieve industry-leading levels of efficiency.

Platform

[Overview](#)

[Modeler](#)

[Pricing](#)

[Try Free](#)

[Trust Center](#)

[Release Policy](#)

Solutions

[Solutions by Industry](#)

[Solutions by Use Case](#)

[Solutions by Initiative](#)

Community

[Developer Community](#)

[Forum](#)

[Camunda Platform 8 Slack](#)

[Events](#)

[On Demand Events](#)

[Camunda meet-ups](#)

[Code of Conduct](#)

Resources

[Developers](#)

[Documentation](#)

[Best Practices](#)

[Case Studies](#)

[Whitepapers](#)

[Webinars](#)

[Why Camunda](#)

[Blog](#)

Camunda Platform 7

[Camunda Platform 7 Docs](#)

[Download Camunda Platform 7](#)

Company

[About](#)

[Careers](#)

[Services](#)

[Customers](#)

[Partners](#)

[Leadership](#)

[Press](#)

[Common Good](#)

[Contact Us](#)