



Enterprise-scale orchestration: A practitioner's guide

Achieving seamless integration,
efficiency at scale, and quick time
to value

Content

Introduction	3	What you need your orchestrator to do	11
You've been automating for a while. What's next?	4	The rise of low-code/no-code automation	13
Orchestration—and why you need more of it	5	Where does RPA fit in?	14
Orchestration use cases	6	When is it time to adopt enterprise orchestration?	15
Delivery of cloud services	6	OpenText Operations Orchestration—the ultimate orchestrator	17
Fulfillment of infrastructure requests	7	Best practice tips	19
Recovery of online services	8	Take your orchestration to new places	21
Renewal of web server certificates	9		
Remediation of incidents	10		

Introduction

Let's say you've been automating for a while and you want to do more—in fact, your business demands it. Enterprise-scale orchestration is a logical next step. But how do you get started? Where does robotic process automation (RPA) fit in? If you're using an open source automation tool, what will you gain by adopting an enterprise-scale orchestrator? What are the best practices?

Here's your guide to achieving seamless integration, efficiency at scale, and quick time to value with enterprise-scale orchestration.



You've been automating for a while. What's next?

Automation everywhere. That's what's required to future-proof your business. You've been designing, deploying, and maintaining automated processes to deliver services faster, with more power, to more users. You've also been working to ensure that your services are cloud-ready, secure, and available 24x7.

But you're under pressure to decrease costs. Not all your automations flow smoothly. Maintaining scripts, tending to brittle integrations, and herding multiple tools are still part of your daily work. Shouldn't some things be easier by now? Things like designing workflows and handling errors, integrating applications across your heterogeneous data center, and automating applications with difficult APIs.

You need a new way to bring automated processes to life. To design end-to-end processes and seamlessly integrate and run thousands of them across domains, with the ability to rerun them on demand or on a schedule.

**Enter enterprise-scale
orchestration.**



Orchestration—and why you need more of it

Orchestration is the heart of enterprise efficiency.

It resides as a layer above automation. You can't have orchestration without automation—you need both. Automation enables a single task to run on its own. Orchestration connects, coordinates, and runs automated tasks to integrate entire processes.

Through orchestrating automation, you can deliver high quality, meaningful services to customers and employees.

An orchestrated system makes decisions, handles errors, and manipulates data. It provides resources and responds to events. It knows how to run and coordinate parallel processes. It cuts out the manual work between each automation step and cuts across enterprise silos. It also connects entire IT ecosystems—error-free—without human intervention.

You can automate almost anything, reduce service delivery from months to minutes, and realize previously unattainable revenue growth.



Orchestration use cases

Any well-understood process with a standardized approach can be orchestrated. A standardized approach assumes that each step of the process is or can be automated.

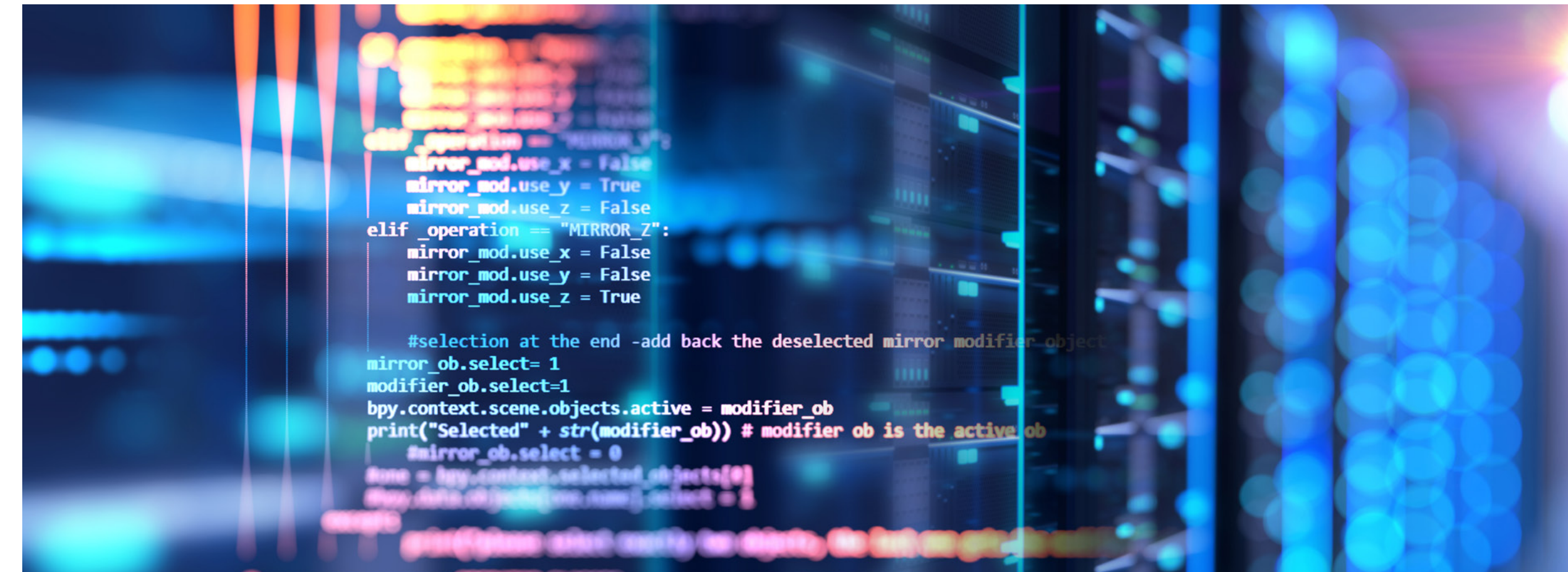
Orchestration use cases range from application deployment to disaster recovery, incident remediation, infrastructure automation and RPA, to service fulfillment.

Let's explore a few possibilities.

Delivery of cloud services

Challenges

Customers wait weeks for basic services as a cloud services provider struggles to meet their needs. Multiple handoffs and escalations between finance, IT, and sales teams are needed to resolve customer issues. The company must find a faster, simpler way to deliver cloud offerings—which means integrating systems and processes across a splintered, multivendor cloud ecosystem.



Solution

Orchestration workflows string together disparate tasks to deliver seamless customer experiences—from new cloud provisioning to cloud upgrades and from capacity and alert reporting to pay-as-you-go billing. The company uses out-of-the-box cloud content to easily integrate with Amazon Web Services, Microsoft Azure, and Google Cloud. A series of workflow operations compile and send usage data to the internal accounting system. The workflow handles advanced logic for usage comparison, which is displayed in the customer's cloud dashboard.

Fulfillment of infrastructure requests

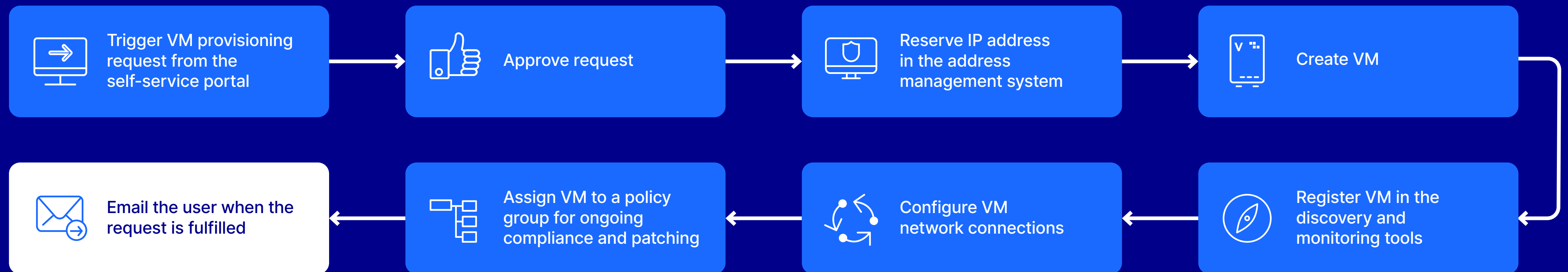
Challenges

Because the IT team is unable to meet intense demand for automated processes, infrastructure requests bounce around departments and wait for weeks to get fulfilled. Non-standard custom workflows increase errors and development time.

Solution

A catalog-based portal offers one-click services for all types of infrastructure requests—from provisioning virtual machines (VM) to right-sizing, increasing storage, configuring firewalls, and creating new user profiles. Services are powered by orchestration workflows that use drag-and-drop automation building blocks to accelerate development times.

Once a request is approved, the workflow coordinates and connects all pre- and post-processing automation steps. Here's an example workflow:



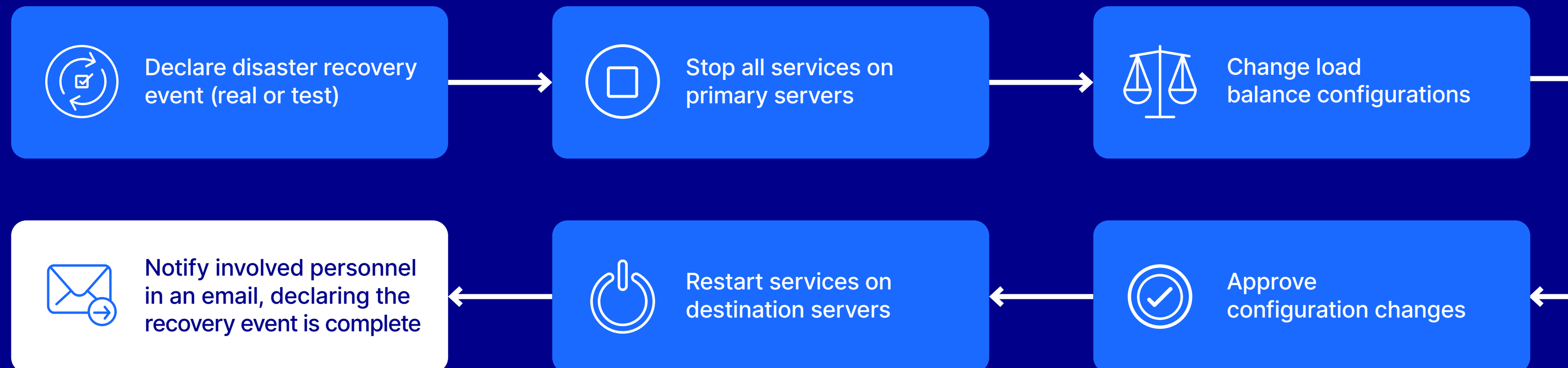
Recovery of online services

Challenges

When IT systems go down, it's difficult to rebuild confidence with employees, partners, and customers. Recovery plans for a company's online services—built around time-consuming, error-prone manual tasks—place heavy demands on IT. Disruptive events must be responded to quickly and with confidence.

Solution

An orchestration workflow starts the recovery process when a change ticket—declaring a disaster recovery event for the online service—is approved. It then moves through these steps:



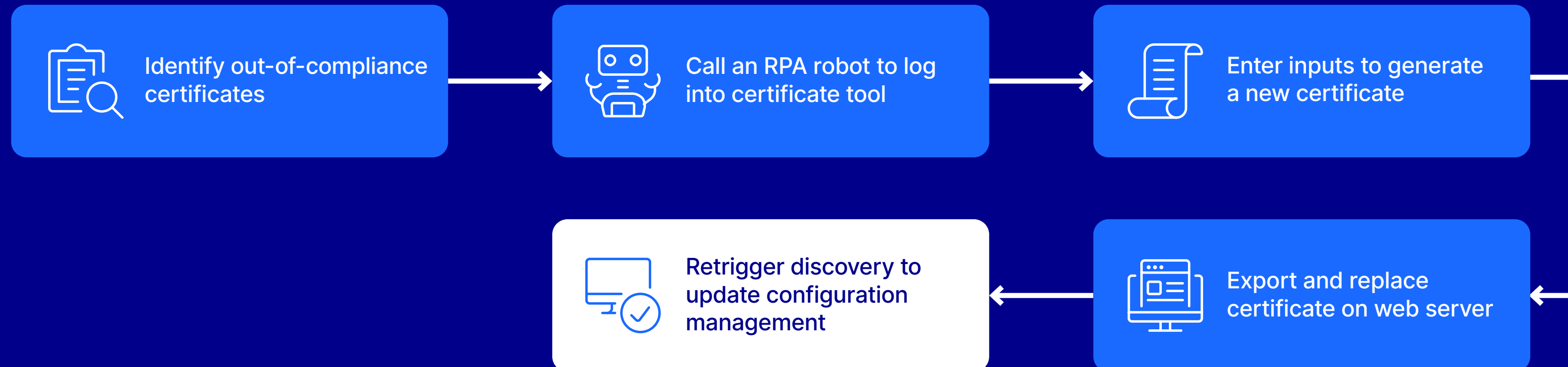
Renewal of web server certificates

Challenges

Expired web server certificates can cause application outages, pose vulnerability risks, and undermine customer experiences with security warnings that block web visitors. Keeping up with expiring web certificates is difficult because third-party certificate tools only offer web-based interfaces that make manual data entry necessary.

Solution

A fully automated workflow is scheduled to run every month to check for certificates expiring in the next 30 days, declaring an event if it finds any. Then the system runs through these steps:



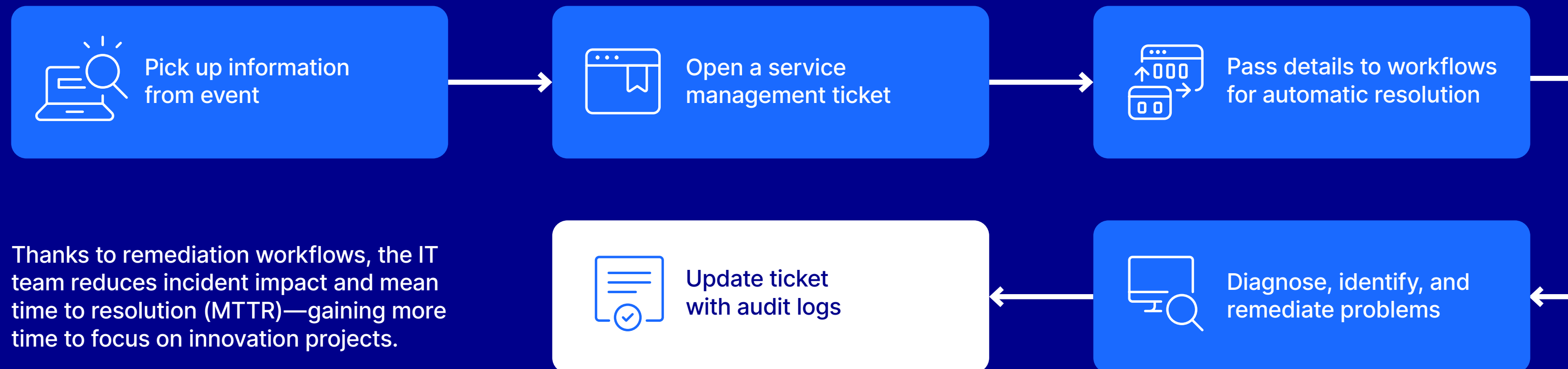
Remediation of incidents

Challenges

IT Tier 1/2 support teams use a combination of runbook templates and documented manual processes to react to alerts and incidents. They manually run health checks; update incident records in the service management tool; coordinate corrective actions to repair incidents and track, update, and close them. They diagnose issues by calling on senior engineers and application owners and escalate incidents to domain experts when problems are not resolved.

Solution

When events arise, the orchestrator directs the entire remediation process working through these steps:



What you need your orchestrator to do

Let's turn our attention to the tool or platform that helps you achieve orchestration: the orchestrator.

According to Gartner®, “Service orchestration and automation platforms (SOAPs) enable infrastructure and operations (I&O) leaders to design and implement business services through a combination of workflow orchestration, workload automation, and resource provisioning across an organization’s hybrid digital infrastructure. SOAPs provide a unified administration console and an orchestration engine to manage data pipelines and event-driven application workflows.”¹

SOAPs have six key differentiating capabilities:²

- **Workflow orchestration**

Provide a unified view to design and orchestrate workflows across multiple applications—on premises and in the cloud. These tools typically include a graphical flow designer to design workflows, visualize interdependencies, and connect disparate tasks and data sources. Automation engineers can also use YAML, Python, or shell scripting to edit and debug the code behind the workflows.

- **Event-driven automation**

Sense inputs (triggers), validate them against configured rules and policies, and respond by taking appropriate actions. New messages in message queues or new files created in a file system are some examples of event-based triggers.

- **Self-service automation**

Offer users a self-service administration console to orchestrate their own workflows with role-based access controls. User satisfaction is improved because users don't have to submit and wait on IT tickets.

- **Scheduling, monitoring, visibility, and alerting**

Provide visibility into IT processes and help meet SLAs. SOAPs provide options for real-time service monitoring, time-based scheduling, and business-driven scheduling (for example, identifying and addressing processes to ensure SLAs are met).



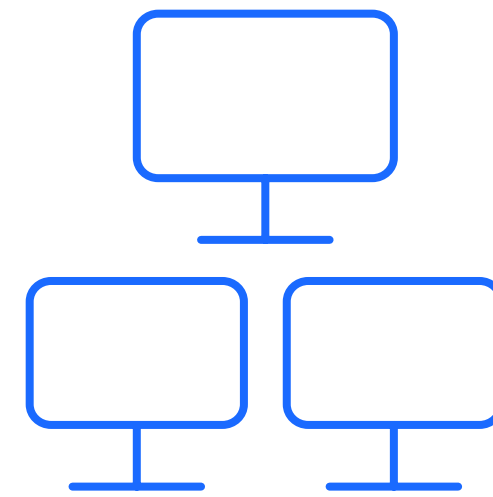
- **Resource provisioning**

Provision compute, network, and storage resources on premises and in the cloud. Resource provisioning tends to be platform-specific and capabilities vary between SOAP providers. SOAPs can also integrate with infrastructure automation tools.

- **Data pipeline management**

Automate file transfer and orchestrate data pipelines. Data pipelines can be used to ingest and process data for batch processing or interactive processing. SOAPs must allow these pipelines to be defined as code for easy maintenance, versioning, testing, and integration.

OpenText believes an orchestrator should make it easy to design workflows across applications—even in a matter of minutes. It should put you in charge of your orchestration—from making decisions to handling errors, manipulating data, and calling subflows. An open, extensible, vendor-agnostic platform should make it possible for you to integrate almost anything and invoke your orchestration from almost anywhere. With a platform that's built for enterprise scalability, you can run thousands of processes across multiple sites.



80%

of organizations currently delivering workload automation will be using service orchestration and automation platforms to orchestrate cross-domain workloads, by year-end 2024.³

5 Orchestrator must-haves

1. Graphical interface for low-code/no-code authoring.
2. Advanced logic for superior orchestration control.
3. Vendor-agnostic platform for seamless cross-domain orchestration.
4. Extensible API-rich platform for invoking your orchestration from anywhere.
5. Scalable architecture for running large-scale processes across multiple sites.

The rise of low-code/no-code automation

In a Forrester predictions report, they noted that during the COVID-19 crisis, enterprises with low-code platforms, digital process automation, and collaborative work management were able to respond faster than those relying on traditional development. As companies adjust to a collaborative and remote work environment, Forrester predicts that the adoption of low-code platforms will grow—even for projects like legacy application modernization.⁴

While developers typically handle large-scale automations, many companies also have smaller workflows they want to automate—and low-code/no-code platforms can play a big role. In small-scale automation scenarios, basic tasks usually performed by humans can be easily attended to by low-code/no-code platforms. Think of the time and cost savings potential of automating tasks that require email, database, and transactional system access.⁵

When companies reduce or remove coding requirements, two things happen:

- Freed-up IT teams start focusing on large-scale enterprise processes they didn't previously have time for.
- Empowered non-developers start designing their own automations—using drag-and-drop, wire-and-click, and plug-and-play techniques.

In terms of productivity, everyone wins.

⁴ Jeffrey Hammond, Randy Heffner, Rob Koplowitz, Diego Lo Giudice, John Rymer, Margo Visitacion, John Bratincevic, Christopher Condo, Chris Gardner, and Abigail Livingston, "Predictions 2021: Software Development", Forrester, October 30, 2020.

⁵ Chris Johannessen and Tom Davenport, "When Low-Code/No-Code Development Works—And When It Doesn't", Harvard Business Review, June 22, 2021.



Where does RPA fit in?

Robotic process automation (RPA) extends an orchestrator's reach to automate processes that couldn't be automated before—namely, screen-based human actions. Think about RPA as one more capability that helps the orchestrator cut manual work between automation steps to make touchless automation possible.

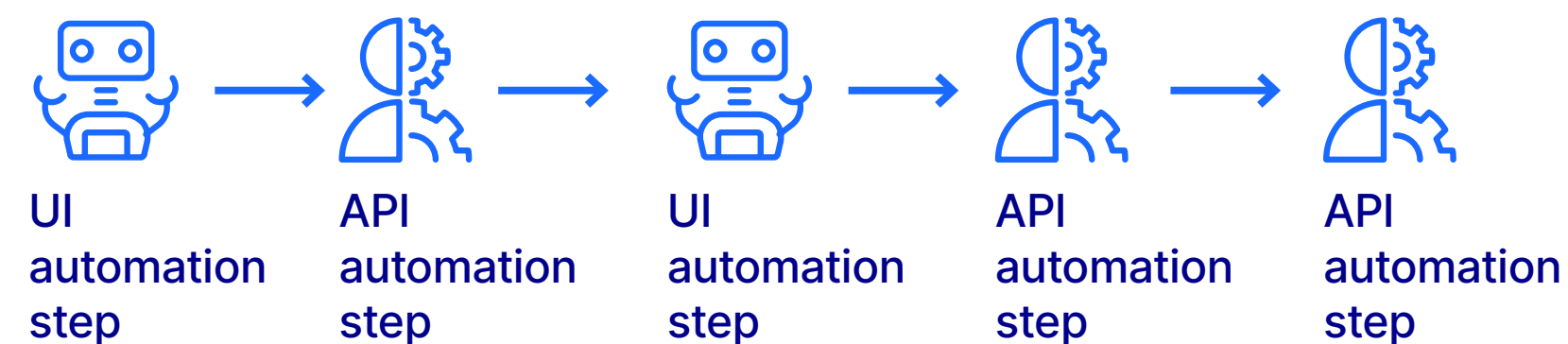
With most basic automations covered, companies are looking to automate deeper, human-driven processes. Employees spend enormous amounts of time on repetitive tasks—logging in and out of legacy and web apps; copying and pasting data; and preparing, manipulating, or merging data from multiple sources. RPA robots mimic screen-based human actions to perform repetitive tasks and expand automation to interfaces with difficult or no application programming interfaces (APIs).

An orchestrator with RPA capabilities often includes a design studio—ideally, a single visual interface—where you can record screen actions, parametrize

input variables, and design end-to-end workflows by combining screen recordings (the UI automation) with IT operation steps.

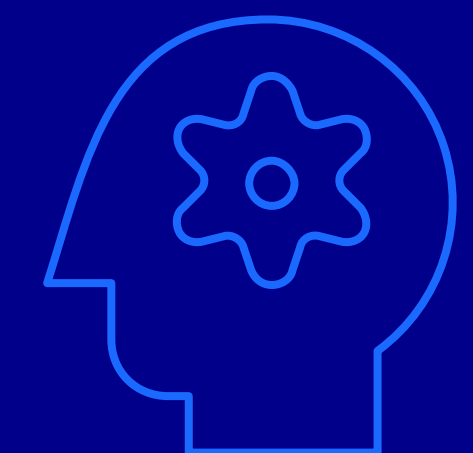
When does it make sense to add RPA to your orchestration workflows?

- When you don't have an easy way to integrate with applications that lack or have difficult APIs. Even if an application is API-enabled, the API may take a long time to understand or may require domain expertise. However, with RPA, you can record screen actions and then easily insert those recordings into workflows.
- When you want to automate processes without changing existing systems and applications.
- When you want to automate quickly.



RPA is perfect for automating processes when you want to automate quickly. Writing code for complex interfaces takes time. RPA takes a simpler path to get things done.

Connecting RPA to other technologies is a common integration use case. That's because companies want to automate entire processes end to end. Orchestration makes it possible by combining UI automation with IT operations steps in a single workflow.



When is it time to adopt enterprise orchestration?

Your needs evolve as your automation becomes bigger and more complex. If you're using a free open source automation tool, growing pains are to be expected. With easier authoring, greater flexibility, and built-in security and scalability, an enterprise-level orchestrator can move you forward.

Gauge your readiness for enterprise orchestration. Ask yourself if any of the following statements ring true for you:

1. You spend too much time creating and maintaining content.

All your automation and integration must be coded. You don't have a central place to design and debug your automation. You maintain a complex structure of scripts, configuration items, and dependencies that offer no easy versioning or maintenance.

2. You want to make automation more accessible by offering self-service options.

Reliance on scripting means non-developers have a high learning curve. A graphical low-code/no-code platform would make work easier and help overcome developer shortages.

3. You need better control of your orchestration logic.

Creating extensive logic is not straightforward—sometimes it's hard to program exactly what you want. Error-handling is a big challenge. Generic error messages and difficult-to-understand error logs make it hard to identify the root cause of issues.

4. You don't have support for all your systems.

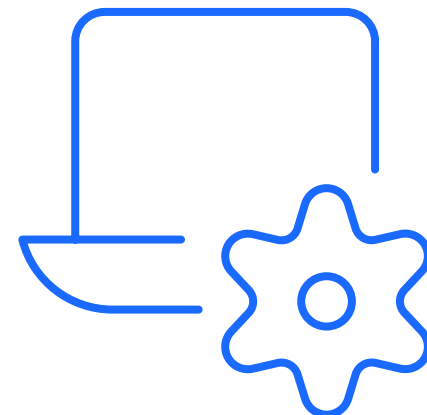
Your code is platform-dependent, so code written for one OS doesn't work on another. Support for Microsoft® Windows® is minimal, which means you must keep checking compatibility for new devices.

5. Your automation lacks standardization.

Because scripts reflect the author's style, enforcing code uniformity is difficult. You want to make code easier to understand and maintain for all—less tribal, more standardized.

6. You're looking for out-of-the-box security, scalability, and high availability.

It takes time to build and manage your own credentials and security protocols. You also must develop and configure components for scaling and high availability. A tool with built-in RBAC and cluster architecture (to easily run and distribute workloads across sites) would simplify your job.



7. Sometimes you need to run agent-based operations.

For highly secure environments—where agentless access and protocols are off-limits—you need to install agents on target servers.

8. You don't have secure, central reporting for your automation.

To see everything that happens in your environment, you need to connect to each system, access each log file, and collect all log files in a central place. What's more, you don't have built-in authentication and encryption for confidential information. That means you must ensure the sensitive data in your logs, such as usernames and passwords, is not exposed.

9. Your licensing model doesn't align with how you're automating.

Node-based licensing is difficult to quantify when your automation doesn't touch a node or does so only occasionally—for example, when you're using your tool as an integrator or when your tool touches the cloud once and not again. You need flexibility for removing node restrictions, improving usage visibility, and paying only for what you use.



OpenText Operations Orchestration—the ultimate orchestrator

OpenText™ Operations Orchestration (OO) is a true orchestrator that automates and orchestrates end-to-end processes with refreshing ease and superior control.

Here are nine ways OO stands out:

1. **Hybrid authoring**

Design workflows in an intuitive graphical interface or author workflows with YAML-based open-source language. Your edits are immediately represented and viewable as text in the workflow designer.

2. **Authoring ease**

Drag and drop out-of-the-box operations from the content library, debug with breakpoints, and generate integration content in minutes using the API content generator. Do all of this in one visual low-code/no-code workflow designer.

3. **Clean content structure**

Don't spend time managing your content and dependencies—OO will automatically manage them for you. Operations, workflows, and configuration

items are organized in content packs for easy versioning and maintenance. Just start a new project, plug and play content, and you're ready to deploy.

4. **Self-service portal**

Make orchestration easy and accessible. Users can trigger pre-defined processes from a secure friendly portal, without the need to access an administrative console.

5. **Advanced orchestration logic**

Run powerful orchestration flows with decision-making, parallel- processing, data-manipulation, and error-handling logic. Define and control all steps in your flows with ultimate precision.

6. **Centralized orchestration**

Because your workflows don't have to reside on hosts, you can orchestrate seamlessly across heterogeneous systems. OO manages your workflows in one central location and assigns orchestration workers to wherever they're needed.

Did you know?

You don't need to rebuild your entire automation when using OO. Reuse what you have. That means you can plug and play configuration scripts—for example: Ansible® playbooks, Puppet scripts, Terraform® modules—in your workflows. With OO, you have options.

Did you know?

OO is open source. CloudSlang is the powerful engine and YAML-based language used in OO. Start developing your own content (for free), use community content, and share your code through GIT.

7. Open, extensible platform

With vendor-agnostic OO, you can achieve integrations across your IT ecosystem. OO exposes REST APIs, so you can invoke orchestration from anywhere—including monitoring tools, service catalogs, cloud portals, command line interfaces, and websites. OO also supports multiple automation protocols—REST, SOAP, SSH, PowerShell, Python, and UI automation with RPA.

8. Automation, powered by RPA robots

Automate what you couldn't automate before with RPA robots that mimic screen-based human actions to automate interfaces with difficult or no APIs. Do

all your work in the workflow designer—record and edit screen actions, parametrize inputs, and combine screen automation with IT operation steps. Entire processes can be automated in a single workflow.

9. Flexible licensing options

Get a license model that aligns to your usage and growth needs. OO offers node-based, concurrent workflow, and hourly licenses. Concurrent workflows can be run in parallel and are ideal for end-to-end integration use cases. When you want to increase capacity with bursts of workflow runs, pick hourly workflows.

Refreshing ease—when orchestration is easy to design and use

- Hybrid authoring (graphical and textual)
- Out-of-the-box operations
- Built-in debugger
- API content generator
- Clean content structure (content packs)
- Self-service portal

Superior control—when orchestration puts you in charge

- Advanced orchestration logic
- Centralized orchestration
- Open, extensible platform
- RPA robots
- Flexible licensing options

OO is your API broker and more. With OO you can design RPA robots too.

Do all your work in one screen—record screen-based human actions and combine them with IT operation steps in a single workflow for end-to-end automation.



Best practice tips

Here are 10 best practice tips—curated from OO users and product experts—for getting started with enterprise-scale orchestration:

1. **Start small**

Start with simpler use cases for everyday operations like resetting passwords, cleaning files, and restarting services.

2. **Identify quick wins**

Generate immediate business value with high-ROI use cases.

3. **Design reusable subflows (instead of slower, bigger, unmanageable flows)**

Insert simpler, faster subflows inside other flows and parallelize when possible.

4. **Make your error-handling workflows reusable too**

Error-handling workflows help you manage exceptions consistently and simplify your troubleshooting.

5. **Work with what you have**

Integrate existing scripts—Ansible playbooks, Python scripts, Terraform modules—into your orchestration workflows. You can even connect workflows with existing service management and monitoring tools to unify your automation and create greater efficiencies.

6. **Monitor your orchestration environment**

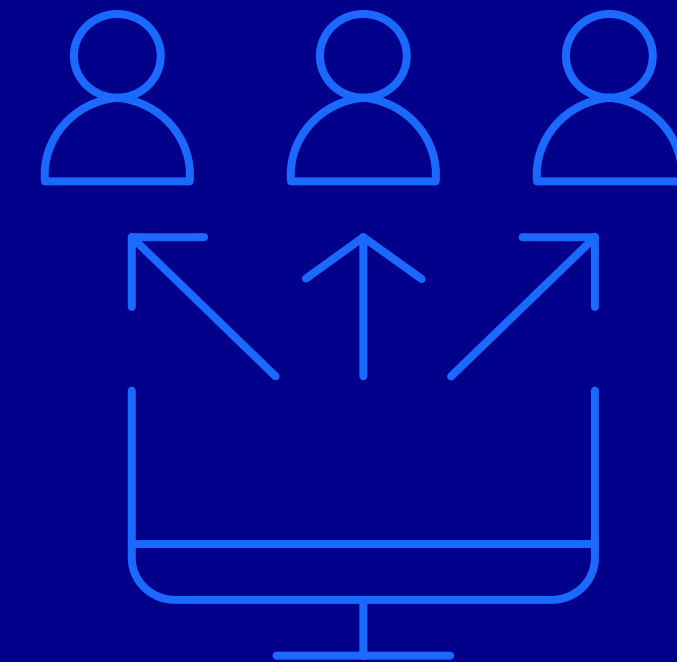
An environment that's monitored is a healthy one. Take a proactive approach to monitoring your orchestration with built-in alerts and automated remediation.

7. **Bring on the RPA robots**

Don't cut your automation short because of interfaces with difficult or no APIs. Use RPA robots to automate screen-based human tasks. You can then automate entire processes by combining UI and API automation.

No more spaghetti flows.

How do you organize a flow that doesn't fit on a big screen? You don't! Split them into microflows.



8. **CI/CD your process orchestration**

Automate the automation! Save time by automating the testing and deployment of your orchestration workflows. By integrating the process orchestration lifecycle into a CI/CD pipeline, you can reduce errors and release faster.

9. **Plan for disaster recovery**

Build a reliable disaster recovery plan with orchestrated workflows. Test your disaster recovery plan every time your IT environment changes—you're not ready if you're not testing. Update your plan continuously.

10. **Offer one-click services**

Empower users with self-service request fulfillment. Offer pre-defined processes that can be triggered from a secure self-service portal. Design for all types of users.



Take your orchestration to new places

We hope this guide has shown you how enterprise-scale orchestration can help you achieve seamless integration, efficiency at scale, and quick time to value (TTV).

With a tool like OO, you can orchestrate anything and everything—with refreshing ease and superior control—across multivendor systems. High ROI will follow.

Visit <https://otex.ly/oo> to learn more about OO or [contact our team for a short demo](#).

“OpenText Operations Orchestration is at the heart of the automation solution. It enabled us to integrate our entire ecosystem. We have a varied set of systems, across different technologies from many vendors, and we needed an orchestrator that could integrate everything. We reduced the time for cloud upgrade requests from two months to an incredible eight seconds.”

Head of Cloud Management, OO Telco Customer

“We wanted a tool that knew how to automate processes end to end. A tool that knew how to orchestrate millions of automated tasks to increase efficiency at scale. A tool that could help us create automated flows in a matter of hours. We decided to give OO the task to orchestrate everything. OO brings us the flexibility and simplicity to run our operations.”

Cloud and Automation Engineer, OO Transportation Company Customer



About OpenText

OpenText, The Information Company, enables organizations to gain insight through market leading information management solutions, on premises or in the cloud. For more information about OpenText (NASDAQ: OTEX, TSX: OTEX) visit opentext.com.

opentext.com

[X \(formerly Twitter\)](#) | [LinkedIn](#) | [CEO Blog](#)

Copyright © 2024 Open Text. All Rights Reserved. Trademarks owned by Open Text. For more information, visit: <https://www.opentext.com/about/copyright-information> 04.2024 | 243-000009-001.EN