How much does it cost?

Transparent Pricing for High-Volume API-Driven Workflow Orchestration

Abstract

Most commercial orchestration engines do not publish prices. Available pricing reports from customers and third parties show that costs typically scale with the number of process executions. With that, the more successful you are at applying process orchestration, the more you pay.

This paper presents a case study for Kiwi, a European travel technology company running a booking related orchestration process with more than 10,000 process instances per day. It describes the migration from a leading BPMN execution engine to SpiffWorks. We use a single exemplary business process to model costs under three leading competitors, and contrast it to actual costs under the SpiffWorks model. We also cover the production challenges the engagement uncovered and how these challenges were overcome.

The Problem: Opaque Pricing

This paper examines three leading BPMN process engines. Two of them follow a cost per execution model. This method of pricing naturally leads to greater complexity. A two step process and a 200 step process can't cost the same amount, so an equation evolves to count things within the process such as human tasks or business rule executions. This equation-based pricing undermines the solution being provided. The need to consider the price for every change in a BPMN diagram adds an additional layer of complexity to building effective business processes.

In a cost-per-execution model, there are innumerable billable events. Start the instance, pay a fee. Once started, many tasks and events trigger additional fees. Further, some of these leading engines require any non-development use to be fully licensed. This is not a "pay as you go" model. It is all or nothing. No major BPMN process engine publishes an algorithm for their cost calculations.

Pricing is opaque across the industry. The lack of published pricing obscures the reality of complex equations and base fees. These are not friendly terms. The lack of transparency–and the potential for complex variability–make process orchestration unpredictable.

Reference Use Case: Kiwi.com Booking Workflow at Scale

In early 2026, Kiwi, a European travel technology company migrated a central orchestration service from Camunda Zeebe (pyZeebe) to SpiffWorks. The service handles a core management workflow with the following characteristics:

Parameter	Value
Daily process instances	10,000 to 15,000
Annual Process instances	3.65M to 5.5M
Incoming messages / correlations	2 (API initiated)
Outgoing Service Task Calls	6

Key facts from the migration:

- During the deployment we implemented additional features that are outlined in the “Beyond Price” section below.
- During our testing we demonstrated the ability to migrate the environment between cloud providers while processing 8,000 live requests with zero failures and zero downtime.
- In 3 months of production level operation, the service experienced one outage. This originated in an external cloud infrastructure incident. Service was restored in 34 minutes. SpiffWorks delivered a written incident report proactively, detailing the timeline, root cause analysis, actions taken, and follow up improvements.

Cost Comparison

As noted above, details on vendor prices are difficult to acquire and calculate. The table below is our best guess at costs across three leading vendors of BPMN Process Orchestration tools. In each case, we’ve noted the source of these calculations, which are historic publications from the organization, or online discussions from existing customers.

Vendor	Billing Model	Billable events	Annual Cost

SpiffWorks	Fixed per environment	1	$60,000 (license) $73,000 (actual)[1]
Camunda 8 Enterprise	Per Process Instance	3,650,000	$330,000[2]
Appian	Per User	Unclear / Not Provided	$72,000 Minimum $400,000 Maximum[3]
Pega Standard	Per Case	3,650,000	$2,920,000

The SpiffWorks Model

SpiffWorks licenses its hosted service and self-hosted licenses on a fixed annual fee per environment, published on their website at https://spiff.works/pricing

---

[1] Includes $13,000 in additional services to tailor the open source software to meet specific requirements of Kiwi. These are outlined in the section “Beyond Price: What production actually requires”.

[2] The $330,000/year data point reflects a reported enterprise self-hosted deployment
https://forum.camunda.io/t/how-do-you-handle-the-additional-costs-of-the-new-8-6-licence/59765
Amazon Marketplace currently shows $175,000.00
https://aws.amazon.com/marketplace/pp/prodview-6y664fcnydiqg

[3] Reddit users quoted “All in top tier service is realistically like a 450k commit for your org”.
https://www.reddit.com/r/Appian/comments/n98afj/appian_pricing/
The last published prices date from 2021. 100 user minimum at $75/user/month. A minimal yearly cost of $72,000
https://www.reddit.com/r/Appian/comments/n98afj/appian_pricing/
“All in top tier service is realistically like a 450k commit for your org”
https://www.reddit.com/r/Appian/comments/1rz1ix2/pricing/

The customer in this case study is on the professional tier at $60,000 per year. Under the executed license agreement, that price is locked for the first three years of the subscription, with any subsequent increases capped at 10% per year on 90 days' written notice.

SpiffWorks does not meter process instance count, decision table executions, service task calls, incoming API message correlations, or user headcount. The same $60,000 fee applies whether the service runs 3.65 million instances or 36.5 million. It is bounded by the execution capacity of the environment itself.

The hosted infrastructure is managed by SpiffWorks, with the following benefits:

- Software updates
- Analytics dashboard
- Visual Design Studio
- SSO integration
- Slack and email support with defined severity-based response times (Severity 1 critical issues monitored 24/7 with a 4-hour response target);
- 99.5% monthly uptime commitment with service credits

SpiffWorks is privacy-focused. Because each customer's SpiffWorks deployment is a dedicated, isolated environment, process data, analytics, and workflow state belong exclusively to the customer. In our configuration with Kiwi, all of Kiwi's customer data is maintained in encrypted form within the hosted environment; SpiffWorks cannot decrypt or access it in plain text.

Beyond Price: What Production Actually Requires

Your internal business processes are not a commodity. SpiffWorks offers a professional service, based on open standards and a fair pricing model. Every business is different. Many software companies expect you to shift your business to their expectations. We worked with Kiwi to update our open source software to support their specific needs, strengthening our product, without compromising Kiwi's institutional knowledge.

We worked collaboratively with Kiwi to identify needs, estimate development costs, and deliver solutions in days to meet their emerging requirements. During the adoption, we identified several needs that did not exist in SpiffWorks. We provided additional software development services to implement the following additions to the underlying open source engine. Kiwi paid an additional $13,000 for these enhancements to assure a smooth transition:

Asynchronous service tasks: The customer's connector proxy returns a 202 Accepted on every service task invocation, completing the task asynchronously via a callback URL. This pattern, which matches how many modern external services operate, was not supported in the platform at the time of migration. The feature was designed, reviewed, merged to the open-source repository, documented, and deployed within days of the request.

Per-task retries with exponential backoff. High-throughput services fail transiently. The customer required configurable retry behavior on service tasks. The feature was implemented with configurable backoff (defaulting to 3, 9, and 27-second intervals), reviewed, and promoted to production within days of the formal request.

Message TTL buffering. The customer's external systems call back to the workflow engine approximately 1–2 seconds after a service task completes. Because the engine's Message Catch Event subscription may not yet be registered at that moment, messages arrived before their subscriber — a publish-before-subscribe race condition common in high-throughput async flows. By implementing a feature to allow for the published message to wait for its corresponding subscriber to be registered, SpiffWorks closed a race window, eliminating a class of errors that previously required manual recovery.

Git-based process model deployment. The customer uses GitLab for source control. SpiffWorks configured a webhook-based deployment pipeline: a push to the production branch deploys updated process models to the production environment in under one second. Separate sandbox and production environments track separate branches, giving the team the same staging/production promotion discipline they use for application code.

These capabilities were not delivered as pre-built features waiting in a catalog. They were requested, scoped, priced transparently as addenda to the base subscription ($8,000 for async service task support; $5,000 for retry logic), and built. The base subscription remains $60,000 per year. This model — a modest, predictable base with transparent, bounded expansion — stands in contrast to a consumption-based model where the bill simply grows, and the growth is not bounded by a negotiated scope of work.

This is a pricing model with the right incentives. When Kiwi needed enhancements, they got them. Fast. SpiffWorks is motivated to resolve issues. In the pay-as-you-go model, there is no incentive to add a feature or resolve an issue.

Conclusion: The Cost of Success Should Not Be Paid Twice

Consumption-based pricing encodes a hidden tax on effective automation. Effective automation might look like an organization successfully deploying a workflow engine, growing its usage, and relying on it for increasing business volume. Under the pricing model of every major competitor, success is rewarded with an invoice that grows in proportion to that success. The rational response is to run fewer process instances, avoid decision tables, and reduce service task calls. Each of these responses undermines the product's value.

A fixed pricing model decouples cost from volume. It makes the annual expense a known, budgeted line item rather than a variable that moves with business activity. It creates no incentive to limit automation. And it places the customer's workflow data in a dedicated environment that belongs to the customer, and not in a shared cluster whose telemetry feeds a vendor's billing engine and potentially leaks your business activities.

The customer described in this paper runs 10,000 to 15,000 process instances per day, through a complex integration architecture, on a $60,000 annual subscription that will not increase for three years. The same workload, modeled against other industry leaders, produces costs ranging from the low six figures to nearly three million dollars annually. Variable pricing doesn't make it easy for their customers to budget. The penalty for success is not a law of nature. It is a pricing choice.

Schedule a conversation. To discuss your orchestration requirements and request a proof-of-concept, contact us at dan@spiff.works or visit https://spiff.works.