SWIMLANE

Technical Whitepaper

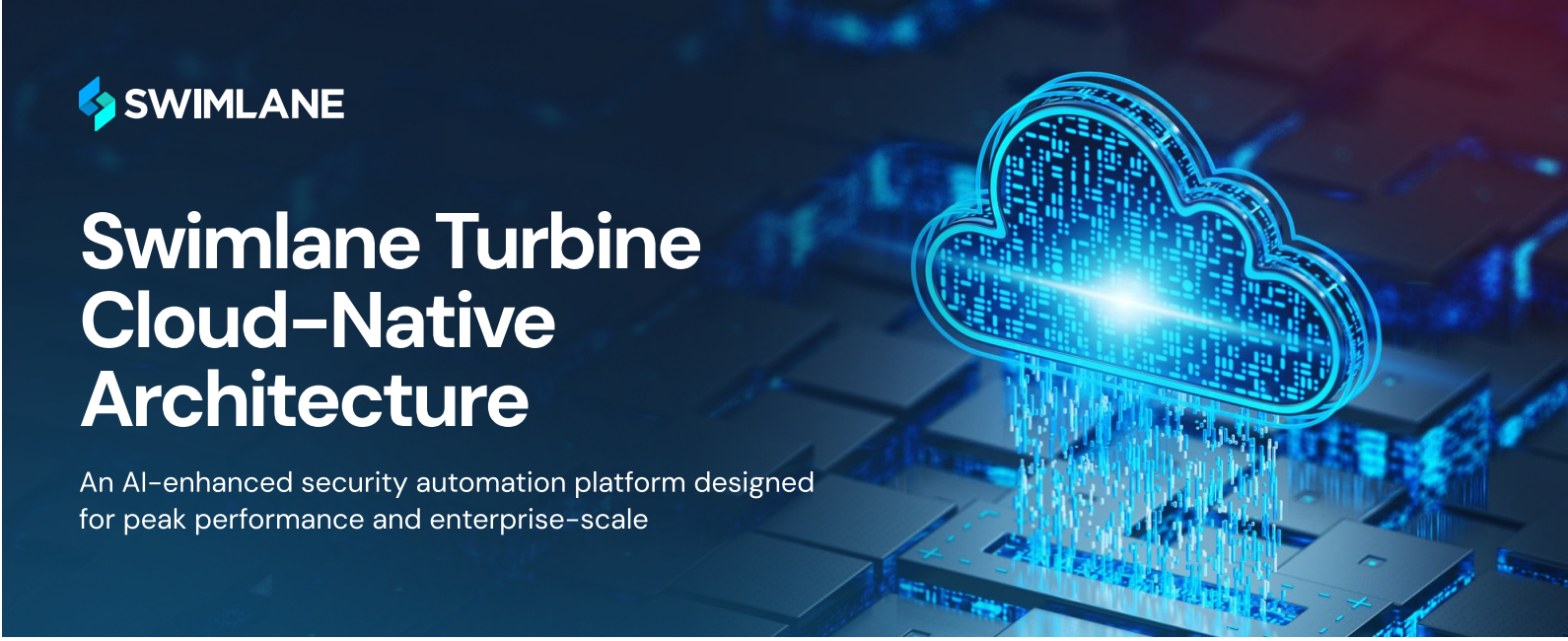
Swimlane Turbine Cloud-Native Architecture

An AI-enhanced security automation platform designed for peak performance and enterprise-scale

SWIMLANE

Swimlane Turbine Cloud-Native Architecture

An AI-enhanced security automation platform designed for peak performance and enterprise-scale



Introducing **Swimlane Turbine**

Swimlane Turbine is an AI-enhanced security automation platform. It's built to be low-code, highly performant, flexible, resilient, and scalable. Turbine's cloud-native architecture enables it to auto-scale its components depending on the workload to ensure high-quality service and performance.

This technical whitepaper delves into the Turbine security automation platform architecture when deployed in the cloud, highlighting its benefits for automation runtime, performance, and scalability. Turbine may also be installed on-premise, including air-gapped environments.

An Overview

Turbine Cloud-Native Infrastructure

Multi-Region Support for Compliance, Data Privacy & Reduced Latency

The Swimlane Turbine cloud-native architecture uses a mature multi-region, multi-tenant SaaS method for deploying the AI-enhanced security automation platform. This architecture is core to Turbine's ability to process and automate tens-of-millions of tasks each day. Turbine's cloud-native infrastructure is deployed in six regions around the globe to ensure compliance with data residency and latency requirements. These regions include the United States, the United Kingdom, the European Union, Singapore, Japan, and Australia. Each region operates independent infrastructure and application deployments. Mutli-national customers can seamlessly integrate their regional accounts using a common GIT repository, identity provider (SSO), and various data sync techniques.

Figure 1: Turbine cloud infrastructure regions.

Multi-Tenant Architecture Drives Cost-Efficiency & Best-in-Class Security Controls

Multi-tenancy is the key element of the Turbine cloud-native architecture. Turbine supports native multi-tenancy at the account level to ensure that microservices can process requests or tasks from different tenants at the same time. Together, Turbine's multi-tenancy and auto-scaling capabilities provide a huge performance capacity pool for each customer.

This whitepaper will use an anonymized real-world case study to illustrate how the Swimlane Turbine architecture easily absorbs load spikes, going more than 100 times higher than typical averages, while remaining cost-efficient to maximize customer return on investment (ROI).

In addition to the performance benefits above, Turbine's multi-tenant capabilities don't compromise on security or data isolation. In Turbine, tenants are isolated on multiple levels:

- Logical isolation, permissions, and role-based access controls (RBAC) at the account level.
- Data isolation at the tenant level ensures that every tenant gets a private database to store sensitive data from records and playbooks.
- Connector workloads run in separate sandboxed containers. Connectors from multiple tenants cannot communicate and will never share data.

Multi-Tenant Architecture Drives Cost-Efficiency & Best-in-Class Security Controls Cont.

In addition to the platform's security and privacy controls, Turbine has mechanisms in place to eliminate “noisy neighbor” performance problems that potentially occur in multi-tenant systems when one tenant affects other tenants during a large load spike. At the playbook engine level, software-managed prioritization logic controls the order in which shared services process workloads from each tenant. This alone ensures a single tenant can never overload an entire cloud region.

Additionally, Turbine Canvas, the platform's low-code playbook-building studio, implements automation design best practices and guardrails during the playbook-building and execution processes. These guardrails ensure that end-users are protected from highly inefficient behavior like infinite looping or running abnormally long operations.

Last but not least, Turbine has built-in observability across the entire stack, covering hundreds of hardware, system, and product metrics, including logs and performance traces. Swimlane teams constantly monitor and analyze data to identify new performance and stability improvements.

High Availability & Resilience with Amazon Web Services

The Turbine cloud-native infrastructure runs on Amazon Web Services (AWS) utilizing three availability zones per-region for extra resilience and high availability. Application services run on an elastic Kubernetes cluster which enables robust node and pod autoscaling. Application design follows microservice ideology with all of the key services being stateless and horizontally scalable. Smaller microservices make the development, deployment, and support of platform services highly efficient.

Turbine Cloud - Architecture

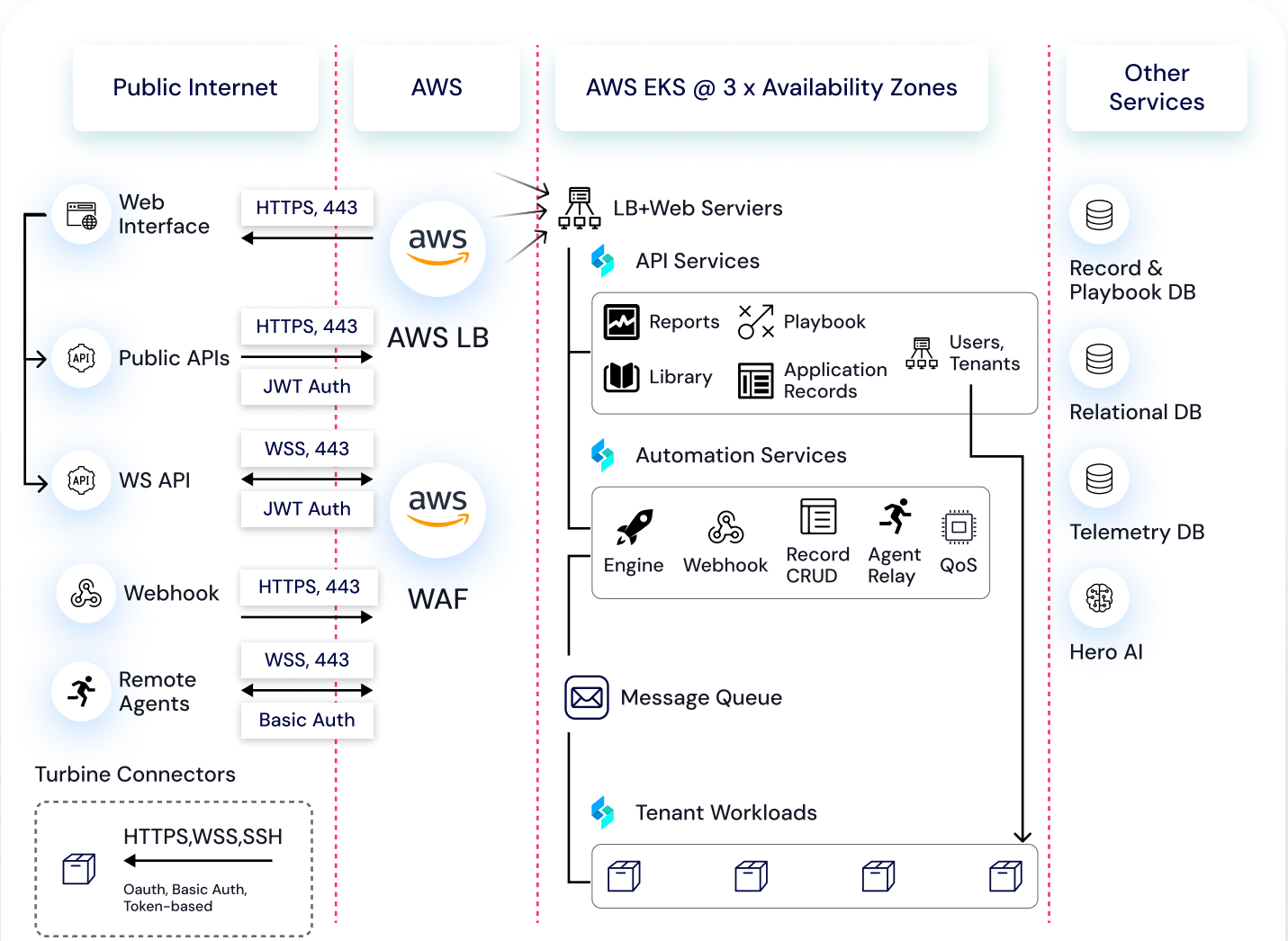


Figure 2: Turbine cloud-native architecture diagram

Multi-Strategy Data Management Facilitates Infinite Customization

Turbine relies on several data management strategies to store and manage large volumes of data including: relational, graph, and document-based. These strategies strictly follow the principle of using the most-appropriate technology for specific operations such as reading, writing, aggregating, and searching. Schema-less document-based design is used for records and playbooks. This enables infinite customization options within the application form builder and Turbine Canvas. Data about users, permissions, settings, tenants, and accounts are stored in a relational format, which better reflects data structure and strict logical connections between these objects. Finally, as an automation development company, Swimlane takes performance, debugging, and telemetry very seriously. Turbine collects, processes, indexes, filters, and searches more than 10 terabytes (TB) of playbook telemetry data monthly, allowing customers to troubleshoot, debug, and optimize their implementations.

Automation Runtime

Trigger and Enrich Automation with Infinite Data Feeds

Data integrations are the foundation of Turbine's low-code automation capabilities. Data ingestion can trigger automated playbooks or be used to correlate or enrich data at any step of a playbook execution. The Turbine data ingestion, enrichment, and export capabilities are part of the art of automation. This gives end-users an open framework to build, expand, and customize virtually any data integration use cases. Turbine can push or pull individual events or batches. It can ingest data based on schedules, listen to webhooks or deploy remote agents on customer's infrastructure. It can work with json, csv files, xml, pdf/binary, and rare or proprietary formats via generic and/or custom connectors.

Data Feeds

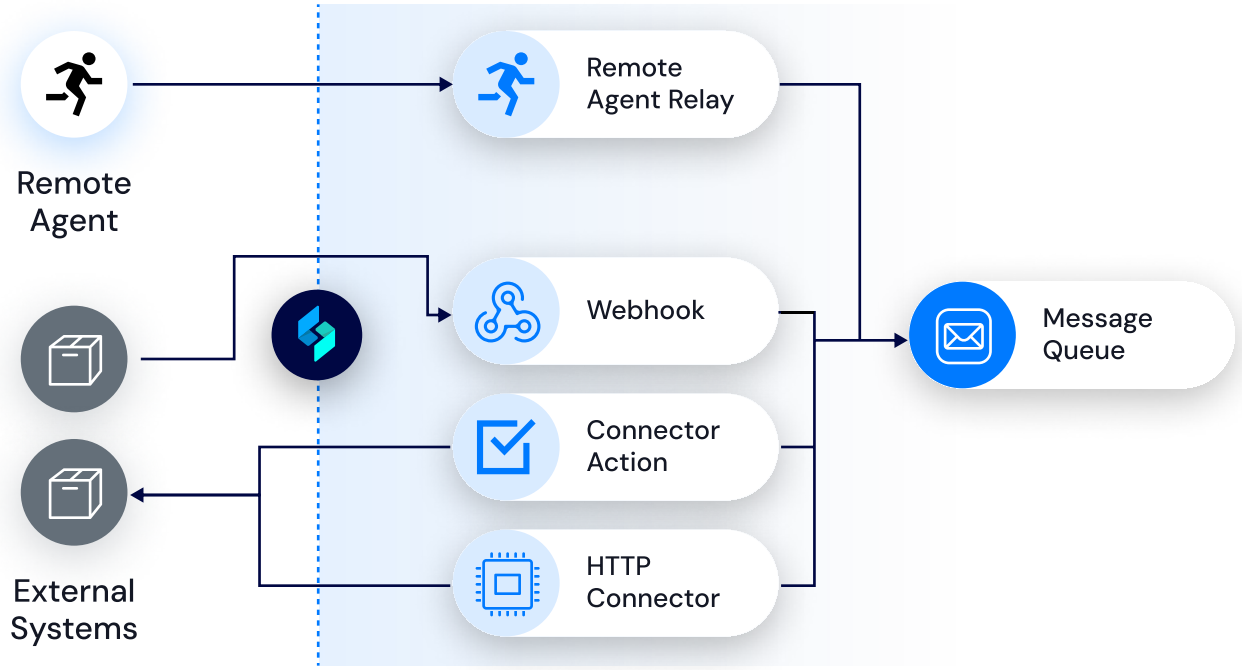


Figure 3: Turbine message queue and data feeds

Once Turbine ingests an event, it's posted to the message queue, which is the centerpiece of the automation architecture. Turbine's highly available and scalable message queue handles more than three billion messages on peak days. This architecture ensures that Turbine never loses alerts, even during massive ingestion spikes. The asynchronous, message-based design allows ingested events to be processed in parallel. A single event, like a Microsoft Defender alert, might have thousands of detections that need to be normalized and enriched in parallel. Using a high degree of parallelism makes Turbine the most performant and scalable security automation platform currently available, far exceeding what's possible with micro-automation built into SIEM, EDR, or XDR platforms.

Parallel Processing

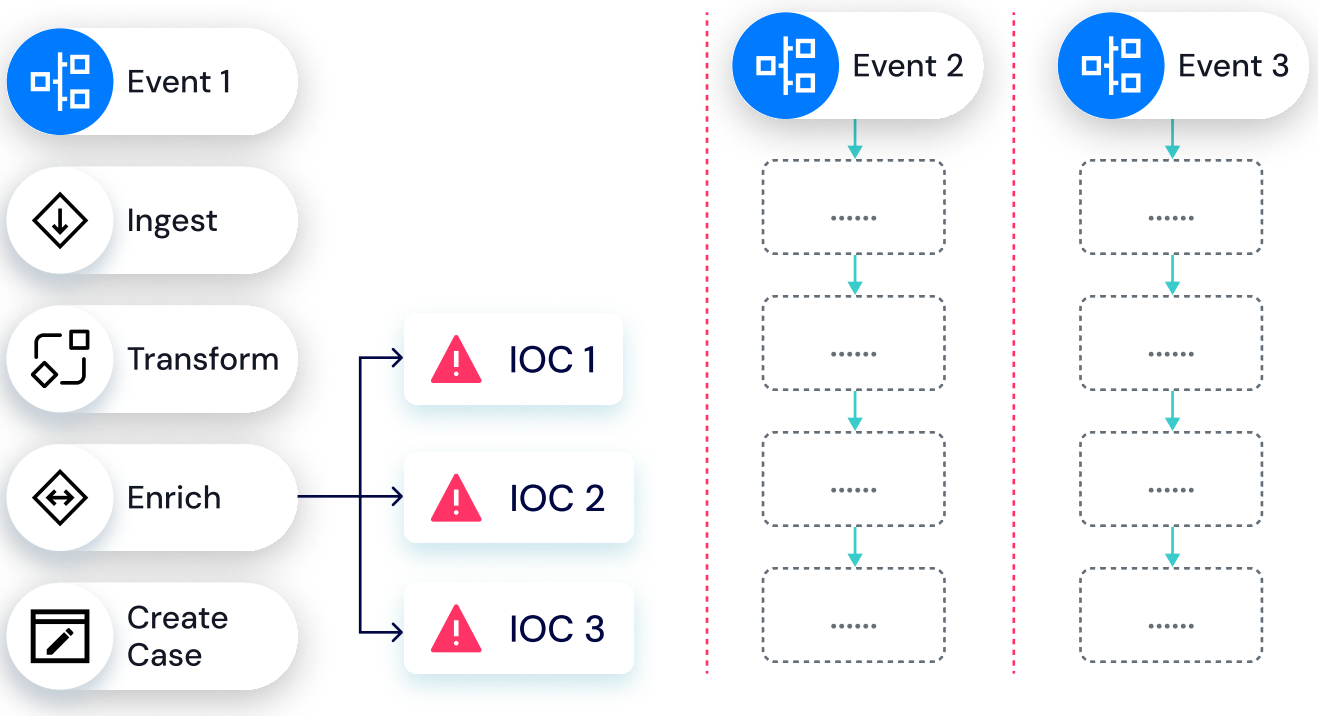


Figure 4: Parallel processing in Turbine

The Power Behind Turbine: An Engine and Agent

Automation runtime in Turbine is powered by two key components: The engine and the agent. The ability to independently scale and optimize each automation action allows Turbine to meet the most sophisticated customer demands.

Turbine Runtime

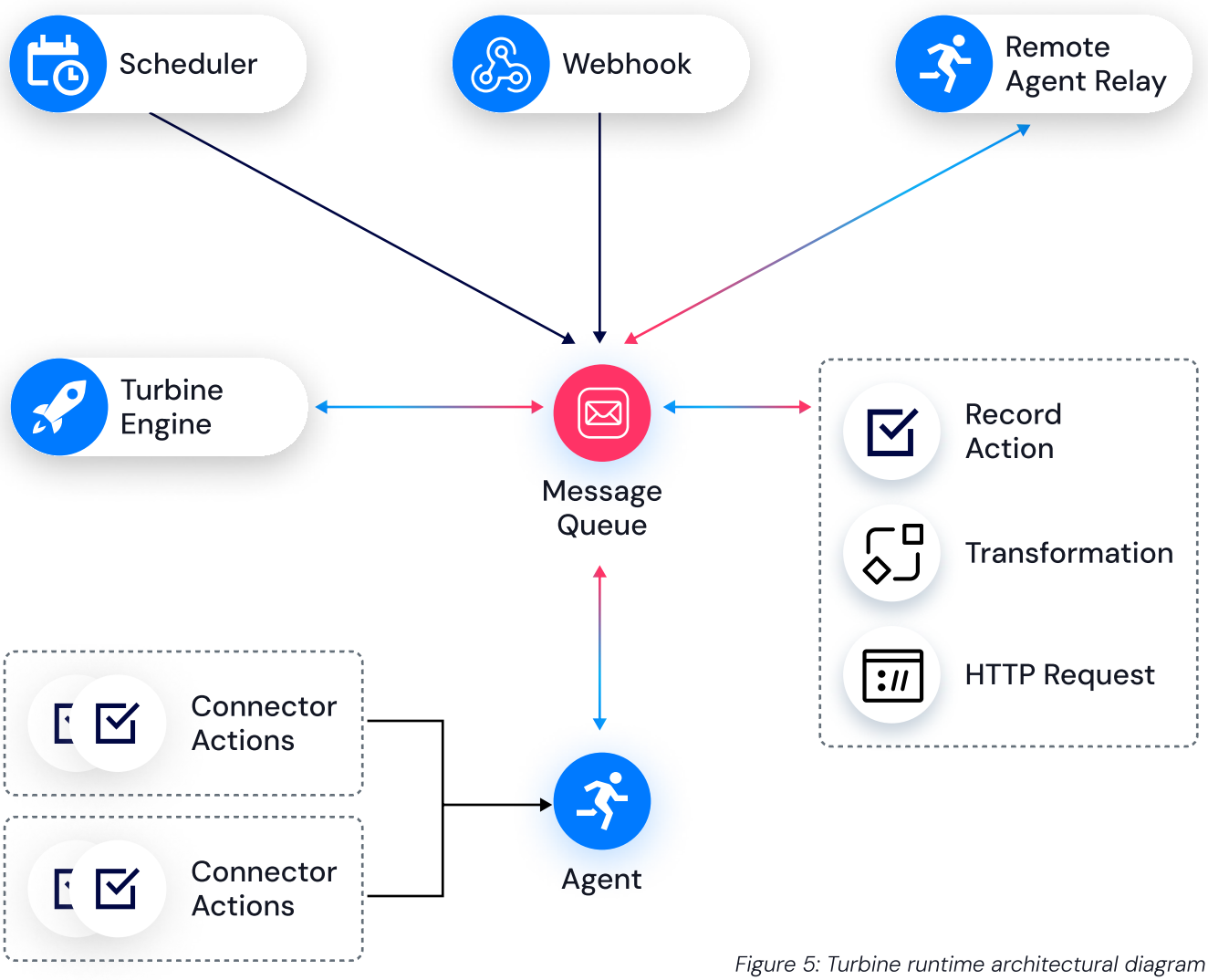


Figure 5: Turbine runtime architectural diagram

The Turbine Engine

The Turbine automation engine is the brain of the platform. It handles everything from general playbook data flow, conditions, loops, parallel vs sequential actions, various validations, and guardrails (e.g. timeout and infinite loop protection). The engine dynamically scales up and down based on the workload. Single engine pods can orchestrate between 1,000 and 2,000 playbook runs per-minute. They are stateless by design and can upscale more than 10 times in just seconds, ensuring a seamless user experience even during massive load spikes.

Figure 6: An example of a single cloud region engine auto-scaling on a typical day

Turbine Agents

Agents are worker pods that execute individual playbook actions. There are two types of actions that agents run:

- General-purpose containerized actions
- Native actions

Containerized agents offer a secure and performant way to execute generic workloads, including inline Python scripts and custom connectors which can be built and uploaded to the cloud by Swimlane customers. Each containerized agent executes in a separate container, isolated by both network rules and container sandboxing settings. Each agent maintains a pool of 20 pre-warmed containers to eliminate any form of cold-start delays.

Native Actions in Turbine

Native actions are pre-built into Turbine to provide an intuitive experience to customize, configure, and optimize performance. The concept of a “native action” was designed to boost performance and simplicity for frequently used automation steps. Each native action type is implemented with a highly specialized micro-service that is optimized for the corresponding workload type and can independently scale horizontally. Types of native actions include:

- Data transformation and normalization
- Generic HTTP connector
- Record CRUD and search operations
- Hero AI

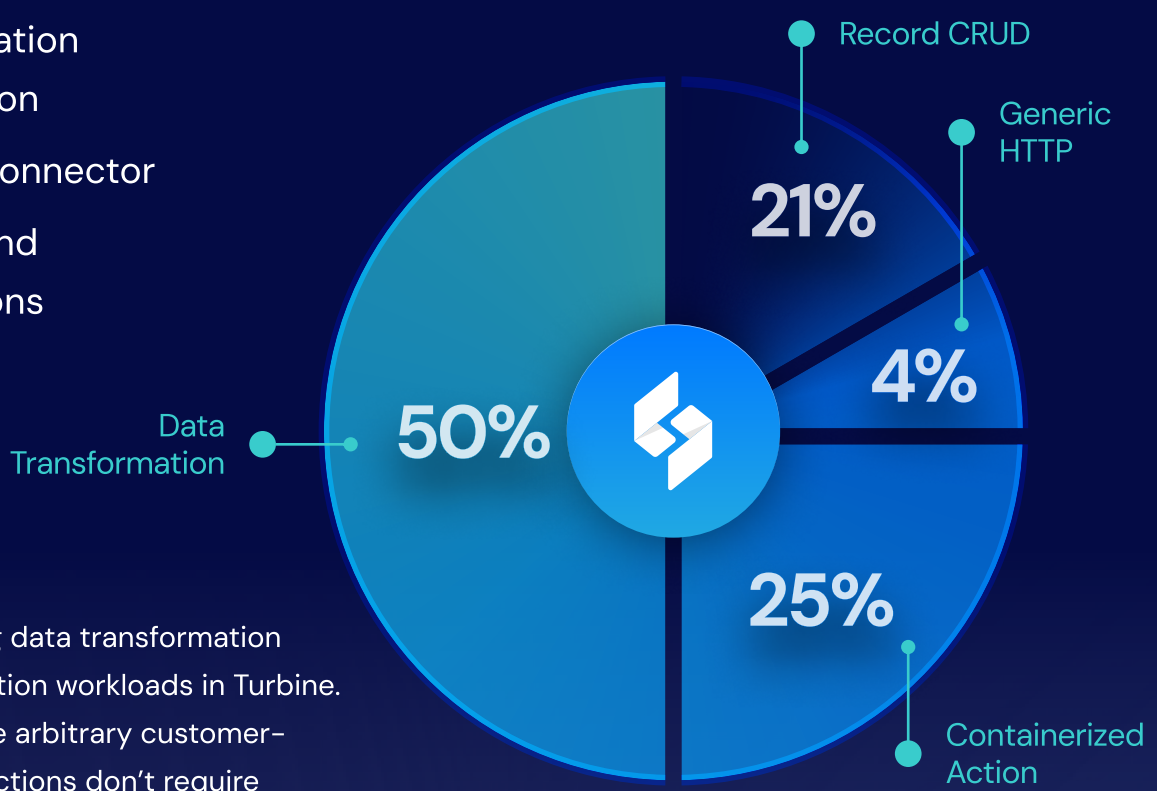


Figure 7: Share of action runs by type

Native actions, including data transformation cover 75% of all automation workloads in Turbine. Since they don't execute arbitrary customer-authored code, native actions don’t require sandboxed containers to run, and can be optimized for specific use-cases. Turbine native actions execute at unprecedented speeds.

For example, data transformations account for half of all automation workloads and take under 50 milliseconds (ms) to run. Record operations represent more than 20% of all workloads, averaging approximately 900ms for record updates and 90ms for searches. Turbine can run a large number of both native and containerized actions in parallel, creating the foundation necessary to handle millions of events and automation playbooks per day, per customer.

Swimlane Turbine Performance Summary

Turbine is a mature, high-throughput SaaS security automation platform that scales linearly, easily scaling to 10 times its automation capacity in a matter of seconds. The platform can handle massive load spikes, allowing single tenants to reach more than 20 times their historical averages. Scalability is achieved by the high degree of parallelism in playbook execution and asynchronous message-based architecture. Cross-tenant influence is eliminated by queue prioritization logic.

Performance Fact-Sheet per Tenant

Workload	Certified per-min	Certified per-day	Max peak per-min
Actions	15,000	22,500,000	75,000
Playbook Runs	2,000	3,000,000	10,000
Events	300	450,000	1,500

Figure 8: Fast facts for Swimlane Turbine performance statistics

Case Study

Practical Validation for the Swimlane Turbine Cloud Architecture

The cloud-native architecture and defining technical characteristics of Swimlane Turbine lead to highly-scalable implementations. This anonymized real-world case study illustrates how the Turbine per-tenant performance provides value to Swimlane customers.

As a baseline perspective, the following is a top-level usage snapshot across all Swimlane customers in the region:

- Tenants: Increased from 270 to 288
- Playbooks executed: more than 21 million, triggered by 5.4 million unique events
- Playbook actions executed: more than 116 million

In order to support this volume, services automatically scale up and down every few minutes, operating between 16 and 40 compute nodes. Figure 9 below shows the number of nodes over time to support the top-level regional usage outlined above.

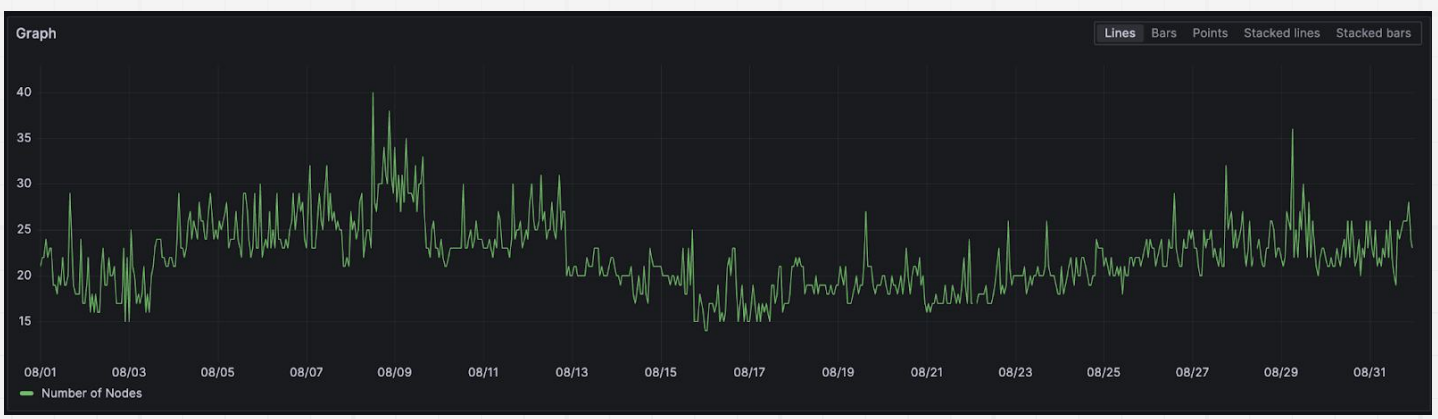


Figure 9: Number of compute nodes by day

The Turbine load, which translates to resource utilization for computing power, data storage and bandwidth, is not distributed uniformly over time or across tenants. In this case study, the top seven out of 288 tenants produced 33% of the total load, and the top 60 out of 288 represented 99% of the load. These big tenants cause pressure on the system, further illustrating the importance of having the correct architectural design. The following sections provide real-world tenant usage data to demonstrate how the Swimlane Turbine platform auto-scales to meet these requirements.

Tenant A

Tenant A handled an average of 272,000 daily automation workloads. Its load was distributed across 72 playbooks, three of which were responsible for 60% of the load. The daily load pattern, illustrated in Figure 10, shows that Tenant A's load started increasing during the second half of the month, with a maximum load on August 23rd.

Figure 10: Automation Actions by Day for Tenant A

Figure 11 provides a more granular view of the August 23rd spike, breaking the day into one-minute intervals. This view shows us that there were three massive spikes, two lasting around one hour, and one short-lived, lasting less than five minutes. During each spike, the load ranged between 3,000-4,000 actions per-minute, nearly 20 times more than the average rate for Tenant A. This single customer volume is comparable to the average rate for the entire cloud region! Turbine was able to handle the load spikes without any failures and without any impact on other tenants in the region.

Figure 11: Automation actions by minute for Tenant A on August 23

Tenant B

Tenant B handled an average of 316,000 daily automation workloads. Its load was distributed across 60 playbooks, three of which were responsible for 60% of the total tenant load. The daily load pattern, demonstrated in Figure 12, shows further non-uniformity. This chart shows that peak days, like August 1st, have 50-270% load utilization spikes compared to the monthly average for Tenant B. As with the Tenant A example, the load is concentrated in spikes lasting anywhere from minutes to hours, with the maximum rate reaching 3,300 actions per minute on August 1st.

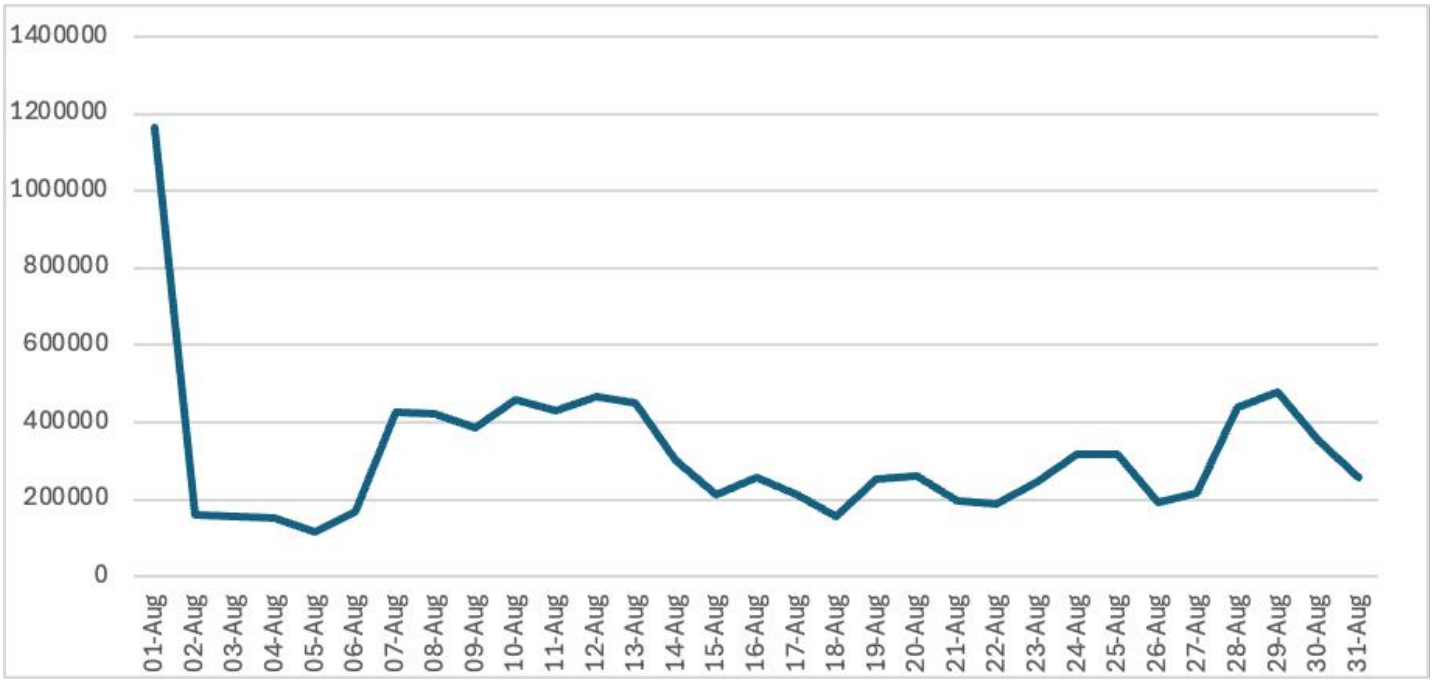


Figure 12: Automation actions by day for Tenant B.

The Moral of the Story

The Turbine cloud-native architecture allows automation services to scale horizontally and vertically to account for high volume spikes and sustained loads over long periods. This ensures Turbine maintains a consistently high level of service quality and delivery for customers, while offering data isolation, process isolation and ensuring a 99.9% uptime.

SWIMLANE

Swimlane Turbine - Built for the Future

Security automation is not a short-term investment, so it's critical to invest in a platform capable of scaling as your organization evolves with performance when it matters most. The Swimlane Turbine cloud-native architecture provides simple and modern user satisfaction, while increasing business agility, and ensuring long-term success with automation. To learn more and see Swimlane Turbine in action, visit swimlane.com/demo.

About Swimlane

Swimlane delivers automation for the entire security organization. Swimlane Turbine is the AI-enhanced, low-code security automation platform that unifies security teams, tools, and telemetry in-and-beyond the SOC into a single system of record to reduce process and data fatigue while quantifying business value and ensuring overall security effectiveness.