socotra

WHITE PAPER

Performance That Scales: Socotra's Latency Testing on High-Complexity Commercial Configurations

Synopsis

In today's hyper-competitive insurance marketplace, speed and reliability are table stakes for carriers. We understand that our customers' products vary widely in complexity—from high-volume personal lines to highly intricate commercial offerings with thousands of scheduled items, multiple exposures, and complex rating logic.

To validate Socotra's ability to support complex products at scale, we executed comprehensive latency tests on a sample Commercial Auto product configuration, modeling both simpler scenarios (20-50 scheduled vehicles) and large-scale, complex scenarios (500-5,000 scheduled vehicles). We ran tests on two AWS infrastructure configurations, replicating real-world scenarios including account creation, validation, pricing, underwriting, and issuance under sustained, high-volume load.

This white paper details the methodology, infrastructure, and results of these tests, providing transparency into Socotra's performance under varied conditions and demonstrating how scaling the platform improves latency for even the most complex commercial products.

The results demonstrate that Socotra can maintain industry-leading latency performance for even the most complex products, with 95% of requests completing within sub-second thresholds under heavy load. As infrastructure scaled, latency improved dramatically—proving that Socotra's cloud-native platform can elastically adapt to performance requirements. No matter how complex or unique your products are, Socotra ensures they will run fast, scale smoothly, and deliver the experience your customers expect

Background

The insurance industry is undergoing profound digital transformation, with carriers competing not just on product offerings but also on the speed, consistency, and quality of their digital experiences. Consumers, brokers, and embedded insurance partners expect near-instant responses that mirror the performance of top-tier consumer applications. Even slight latency can undermine Net Promoter Scores (NPS), reduce conversion rates, and prompt intermediaries or partners to choose faster competitors.

Commercial lines amplify these challenges: complex policy structures, high volumes of scheduled items, and intricate rating logic strain traditional policy administration systems. Many insurers still operate legacy or "lift-and-shift" systems that, while migrated to the cloud, are not optimized for horizontal scalability or modern workloads. These systems struggle to keep pace with growing data volumes, concurrent user demands, and dynamic pricing requirements.

Socotra's cloud-native architecture was purpose-built to address these pain points. By leveraging services like Amazon RDS and intelligent auto-scaling, Socotra delivers high throughput and low latency under demanding conditions, even as product complexity increases.

Socotra: A Scalable Solution for Complex Commercial Products

Addressing Scale Without Compromise

Commercial insurance products vary dramatically in structure and scale. A single line such as Commercial Auto may involve thousands of scheduled items, complex rating rules, and specialized underwriting requirements. Managing these workloads efficiently is not just about raw performance—it's about consistency, adaptability, and predictable scaling.

Socotra's core platform is intentionally designed for elasticity. This allows carriers to configure and deploy even the most intricate products without latency penalties or costly infrastructure over-provisioning.

Key Differentiators

- **Elastic Infrastructure in Practice**: Our tests with an average of 3,000 scheduled vehicles demonstrate how scaling infrastructure reduces latency dramatically, ensuring smoot performance even at high complexity.
- **Consistent Latency Across Workflows:** From account creation to pricing, underwriting, and issuance, latency remains stable even as workloads grow, preserving user experience for carriers, brokers, and partners.
- **Cost-Aware Scaling:** Socotra's auto-scaling capabilities respond dynamically to demand, minimizing cost during off-peak periods while meeting peak performance targets.
- **Integration Without Bottlenecks:** An open API framework enables carriers to connect dvanced analytics, embedded insurance flows, or third-party systems without introducing latency overhead.
- **Future-Proof Flexibility:** Carriers can evolve product configurations or expand portfolios confidently, knowing Socotra's platform will adapt without expensive rewrites or migrations.

Test Approach and Results

Test Configurations

To validate Socotra's ability to handle complex commercial products under realistic operating conditions, we conducted a series of performance tests that varied both policy structure and infrastructure provisioning. This mixed approach allowed us to isolate the impact of product complexity from the effects of hardware scaling.

Testing was based on a **Commercial Auto** product configuration from Socotra's product library, modeled closely after those used by production customers. The configuration was designed to reflect the complexity of real-world deployments, incorporating multi-vehicle exposures, dynamic underwriting logic, and document generation. Each exposure and policy record included:

27	70	6	1-3
Fields per exposure	Fields per policy outside of the exposure	Documents per policy	Drivers per policy

Overview of Test Matrix

To evaluate performance under varying levels of complexity and resource allocation, we combined t**wo policy structures** with **two infrastructure configurations**, resulting in three distinct test environments. This approach was designed to isolate the independent effects of **product complexity** (horizontal scaling) and **infrastructure provisioning** (vertical scaling) on overall system performance.

Policy Structures

1. *Small schedule configuration* — 20–50 scheduled vehicles per policy (avg = 35).
2. *Large schedule configuration* — 500–5,000 scheduled vehicles per policy (avg = 2,750).

Infrastructure Configurations

1. *Medium-tier hardware* — representative of typical production environments.
2. *Large-tier hardware* — scaled to emulate higher performance provisioning.

This pairing produced three total test configurations:

- **Medium 35** — Medium-tier hardware × Small schedule product.
  - Serves as a **baseline scenario** representing a typical production environment.
- **Medium 2750** — Medium-tier hardware × Large schedule product.
  - Represents an **unrealistic but informative stress test,** designed to isolate the impact of increasing product complexity without changing hardware capacity.
- **Large 2750** — Large-tier hardware × Large schedule product.
  - Evaluates the **effect of scaling infrastructure** on latency and throughput while keeping product complexity constant.

By testing both dimensions—product size and infrastructure scale—this matrix provided a complete picture of how Socotra's platform performs across a realistic-to-extreme operational spectrum.

Configuration Details

Configuration 1 - Medium 35:

A baseline scenario representing a moderately complex Commercial Auto product.

AWS provisioning:

RDS:
- MariaDB(1): db.m8g.2xlarge
- MariaDB(2): db.t4g.medium
- Postgres(1): db.t4g.large
- Postgres(2): db.m8g.2xlarge

EC2: m8g

- **Approximate AWS cost:** Under $400/day, presuming constant load.
- **Policy structure:** 20–50 scheduled vehicles per policy (avg = 35).
- **Purpose:** Establish baseline latency and throughput for a typical mid-size commercial product under steady-state load.

Configuration 2 - Medium 2750:

A high-complexity configuration designed to test scalability and performance elasticity under extreme data volume.

AWS provisioning:

RDS:
- MariaDB(1): db.m8g.2xlarge
- MariaDB(2): db.t4g.medium
- Postgres(1): db.t4g.large
- Postgres(2): db.m8g.2xlarge

EC2: m8g

- **Approximate AWS cost:** Under $400/day, presuming constant load.
- **Policy structure:** 500–5,000 scheduled vehicles per policy (avg = 2,750).
- **Purpose:** Evaluate platform behavior when running a large-scale product on mid-tier infrastructure. This configuration intentionally represents an **unrealistic product-to infrastructure pairing**, used to demonstrate how the platform performs under constrained but extreme workloads.

Configuration 3 - Large 2750:

A scaled infrastructure configuration using the same large-schedule product as *Medium 2750*, but with expanded hardware resources to assess the performance impact of infrastructure scaling.

AWS provisioning:

RDS:
- MariaDB(1): db.m8g.2xlarge
- MariaDB(2): db.t4g.medium
- Postgres(1): db.t4g.large
- Postgres(2): db.m8g.2xlarge

EC2: m8g

- **Approximate AWS cost:** Under $600/day, presuming constant load.
- **Policy structure:** 500–5,000 scheduled vehicles per policy (average ≈ 2,750).
- **Purpose:** Quantify latency and throughput improvements achieved through infrastructure scaling while holding product complexity constant.

Testing Conditions

To ensure results accurately reflected real-world system behavior rather than transient optimizations, several measures were taken during test execution:

- The number of scheduled items was randomized within the defined range for each test to prevent caching or repetitive query optimization.
- Each environment was preloaded with at least 50,000 existing policies, simulating the data volumes and operational context of a production-scale system.
- Tests used 200 concurrent virtual users, significantly exceeding typical commercial-auto workloads, to validate scalability under sustained, high-concurrency conditions.
- During each test run, policy creation throughput averaged 980 policies per minute on the Medium 35 configuration, 790 policies per minute for the Medium 2750 configuration and 910 policies per minute on the large configuration.
- To reduce the likelihood of false optimizations by the underlying technology stack, each virtual user introduced a randomized delay of 3 to 15 seconds between each policy creation.

Test Infrastructure

The following diagram provides a high-level overview of Socotra's core platform test infrastructure. Socotra uses key services such as Amazon EKS and Amazon RDS. These services are configured with autoscaling to meet increasing load requirements without impacting services and to optimize cost of the infrastructure.

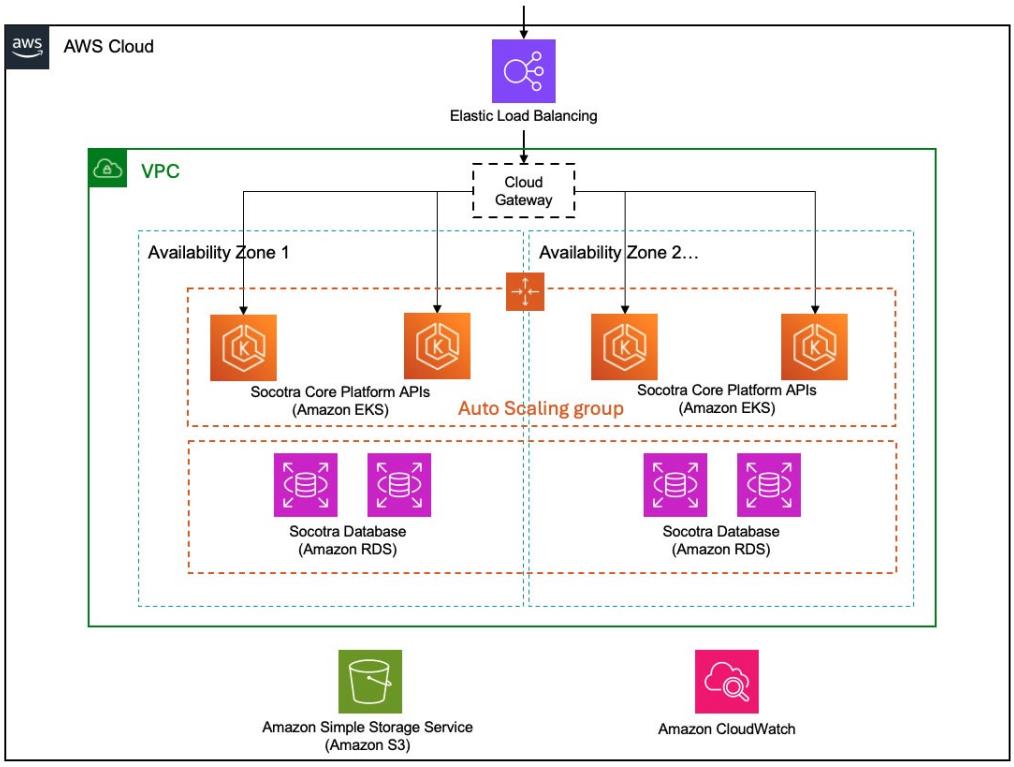


Figure 1: Test Infrastructure Overview

Note: There are additional supporting services in pods in the EKS cluster. The diagram depicts core policy administration services. Socotra also makes extensive use of CloudWatch and related services for security and introspection, not shown here.

Autoscaling

All provisioning occurred using Socotra's automatic autoscalling, which requires no manual provisioning. Our previous AWS-audited performance test featured a load ramp-up, which could be seen graphically in performance results. For more discussion on autoscaling and to view those results, see our 2024 personal auto performance results.

Target APIs

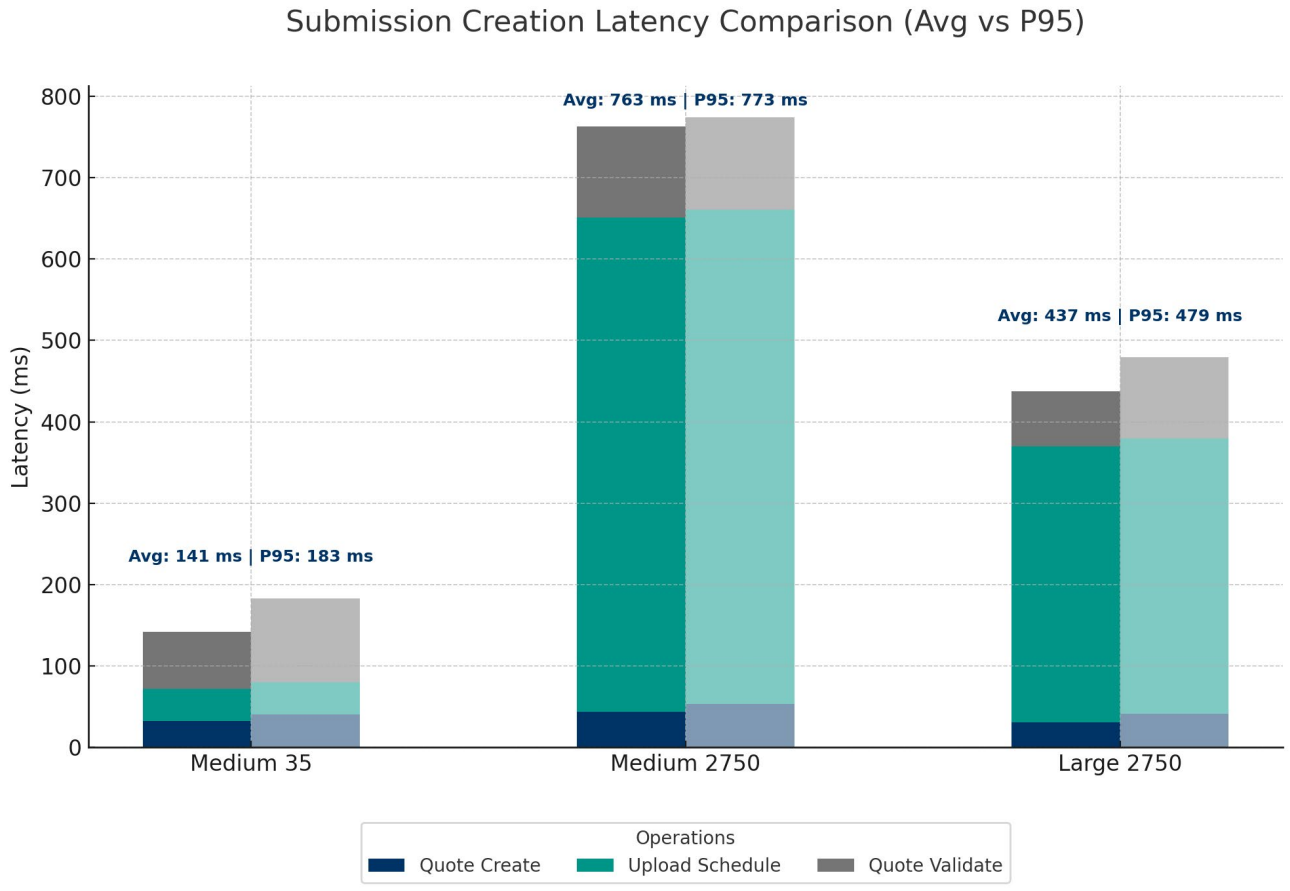
- **Account Create:** Creates an account (i.e., policyholder) in draft state. This entity represents the insured and is a container for any number of policies.
- **Account Validate:** Verifies that all required data is set for the account, and that it is ready for quoting.
- **Quote Create**: Creates a quote in draft state.
- **Schedule Upload:** Uploads the list of scheduled items with their corresponding fields.
- **Quote Validate:** Validates quote data fields, including required data and other rules.
- **Quote Price:** Prices the quote based on rating logic and data contained within the quote and its exposures.
- **Quote Underwrite:** Verifies that the quote meets underwriting requirements based on structure and carrier risk appetite.
- **Quote Accept:** Advances the quote to "Accepted" state. For some carriers, this equates to the quote becoming "bound."
- **Quote Issue:** Creates and issues an on-risk policy using the quote data.

Test Workflows

We divided the **quote-to-bind workflow** into **three sub-workflows** for clearer analysis:

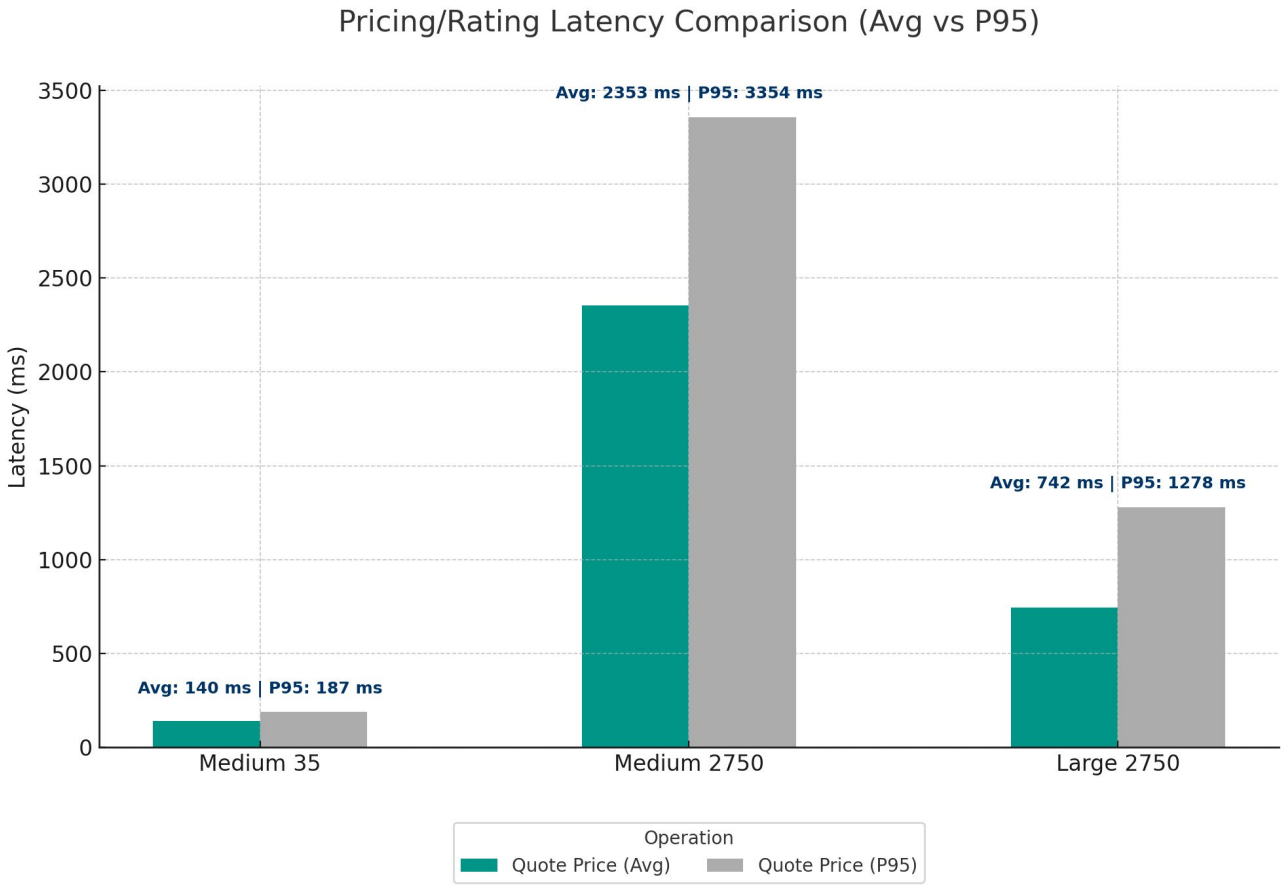
1. Submission Creation (Create → Upload Schedule → Validate): This workflow captures an underwriter's first interaction with the Socotra platform when initiating a new policy. It includes creating a quote, uploading the schedule of items, and validating the submission to ensure data accuracy and adherence to business rules. As expected, latency increases with larger and more complex schedules, with the **Medium 2750** configuration showing higher upload and validation times due to the significantly greater data volume. However, when scaled to the **Large 2750** configuration, total average latency decreases by over 40%, effectively restoring near-baseline responsiveness.

Config	Quote Create (ms)		Upload Schedule (ms)		Validate (ms)	
	Avg	P95	Avg	P95	Avg	P95
Medium 35	32.61	40.49	39.51	39.51	69.60	103.07
Medium 2750	43.86	53.02	607.38	607.38	111.88	113.54
Large 2750	30.83	40.93	338.88	338.88	67.79	99.34



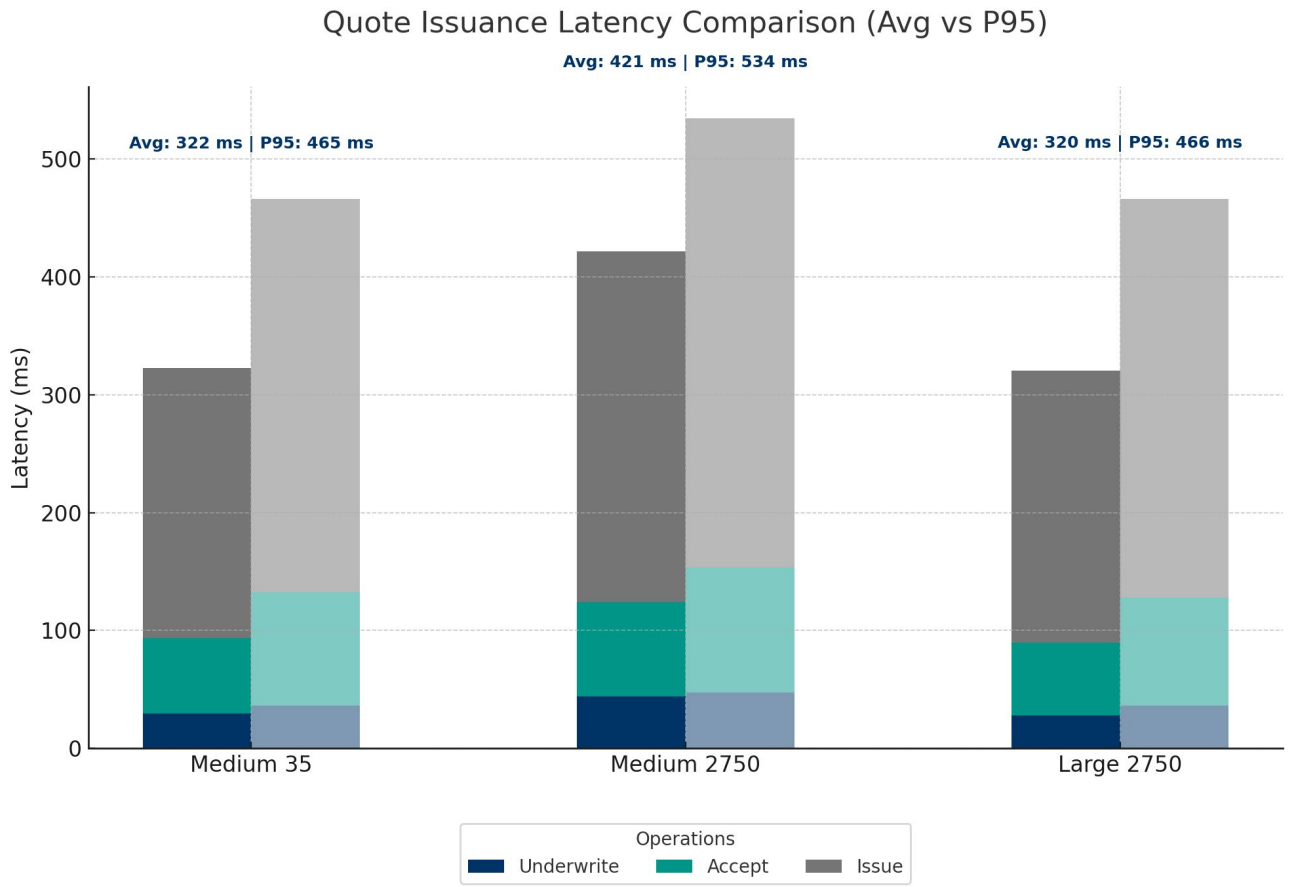
2. Pricing/Rating (Pricing only): This workflow represents the most compute-intensive step in the quote-to-bind process: applying the rating algorithm and business rules to each scheduled item in a commercial policy. As expected, latency increases with the number of scheduled items, with the **Medium 2750** configuration exhibiting significantly higher processing times due to the size and complexity of the dataset. When scaled to the **Large 2750** configuration, total average latency improved by nearly 70%, and P95 latency improved by more than 60%.

Config	Pricing (ms)	
	Avg	P95
Medium 35	140.8	187.2
Medium 2750	2,353.0	3,354.8
Large 2750	742.7	1,278.6



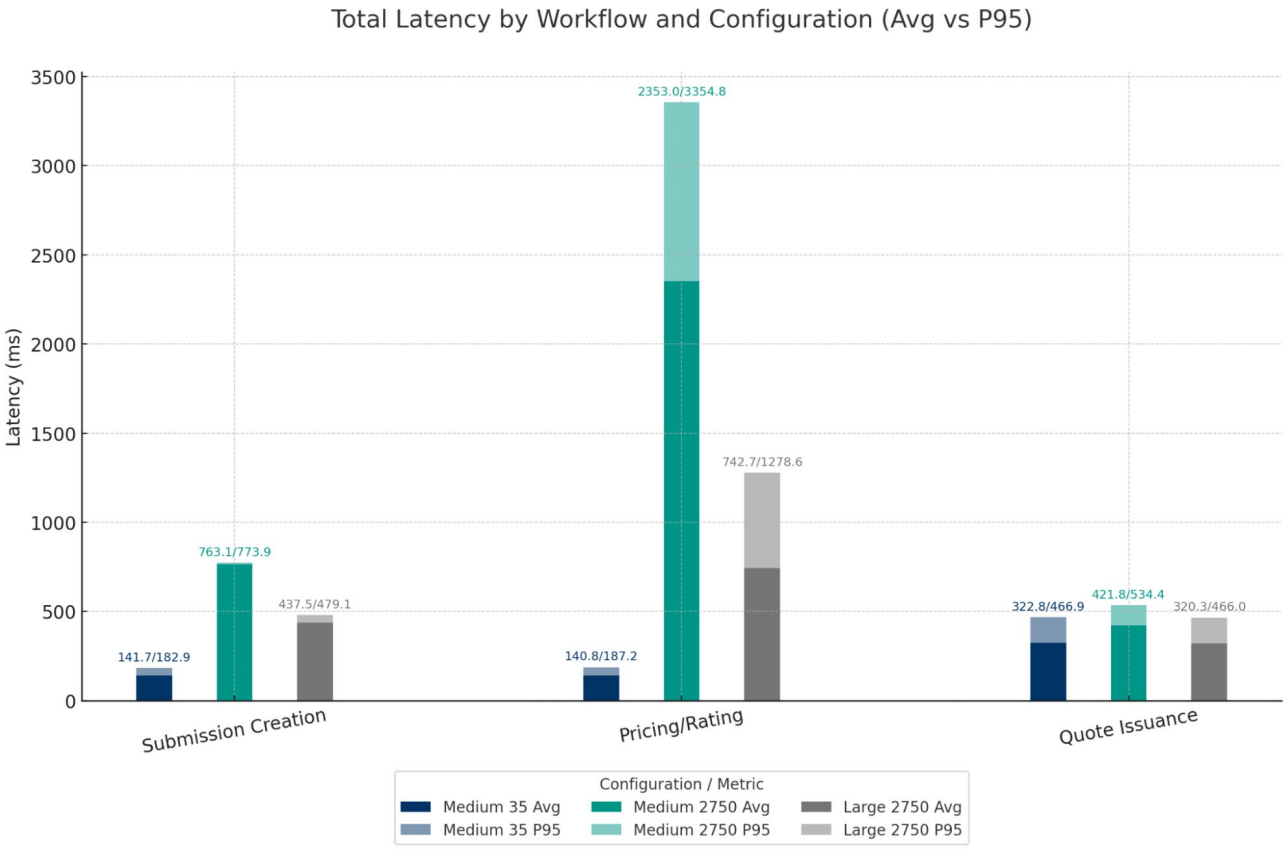
3. Quote Issuance (Underwrite → Accept → Issue): This workflow captures the final stage of the quote-to-bind process, where a quote advances through underwriting, acceptance, and issuance to bind coverage. Latency in this phase remains consistently low across all configurations, demonstrating Socotra's ability to maintain responsiveness even as complexity scales. When the workload increased to the Medium 2750 configuration, total average latency rose to 421.8 ms and P95 to 534.4 ms, representing increases of 30.7% and 14.5%, respectively, when compared to the **Medium 35** configuration. After scaling to the **Large 2750** configuration, total average latency dropped back to 320.3 ms and P95 to 466.0 ms.

Config	Quote Create (ms)		Upload Schedule (ms)		Validate (ms)	
	Avg	P95	Avg	P95	Avg	P95
Medium 35	29.77	36.13	63.64	96.48	229.42	333.30
Medium 2750	43.88	47.08	80.34	106.32	297.56	381.01
Large 2750	28.02	36.29	61.61	91.96	230.69	337.81



End-to-End Workflow Comparison

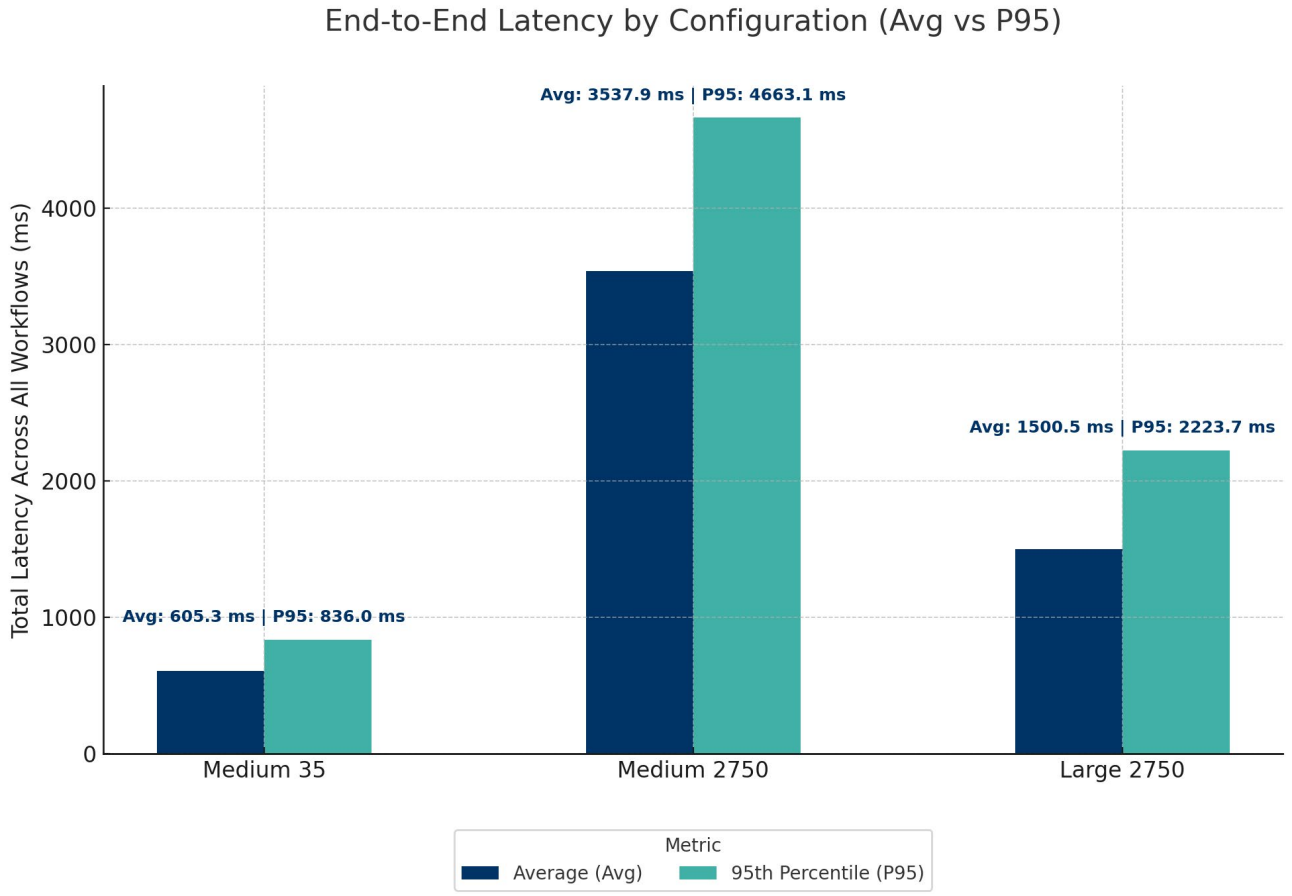
The preceding sections explored each stage of the quote-to-bind process—Submission Creation, Pricing/Rating, and Quote Issuance—demonstrating Socotra's ability to sustain low latency and predictable performance across varying levels of complexity. To provide a complete picture, the following analyses consolidate these results, comparing performance across configurations and summarizing total latency throughout the entire workflow



This chart compares total latency across the three primary subworkflows for each configuration—**Medium 35, Medium 2750, and Large 2750**—providing a detailed view of how performance scales with increasing data volume and infrastructure size.

As expected, latency rises with larger datasets. The **Medium 2750** configuration shows the highest latency across all workflows, driven by the processing demands of thousands of scheduled items. When scaled to the **Large 2750** configuration, performance improves dramatically: average latency decreases by 58% for Submission Creation, 68% for Pricing/Rating, and 24% for Quote Issuance, restoring responsiveness close to baseline levels.

The Medium 35 configuration serves as a practical baseline for mid-sized commercial workloads, while the **Large 2750** configuration highlights the efficiency and elasticity of Socotra's cloud-native architecture. Across all workflows, the system scales predictably—ensuring consistent responsiveness for both user-driven and compute-intensive operations.



Aggregating the total latency across all three workflows reveals the platform's overall throughput and responsiveness. Moving from Medium 35 to Medium 2750, total average latency increases from 605 ms to 3,538 ms, while P95 latency rises from 836 ms to 4,663 ms, reflecting the expected computational cost of scaling to large, data-rich products.

However, when infrastructure is scaled to the Large 2750 configuration, end-to-end latency improves substantially—average latency drops to 1,501 ms and P95 to 2,224 ms, representing reductions of 57.6% and 52.3%, respectively, compared to the Medium 2750 configuration.

Across all configurations, the relationship between average latency and P95 latency remains stable, with P95 values averaging roughly 1.4× to 1.6× higher than the mean. This consistency demonstrates the platform's ability to deliver predictable performance even at the 95th percentile, a critical factor in maintaining reliability for production-scale systems.

Taken together, these results confirm Socotra's ability to scale gracefully with increasing system load and data complexity. Whether serving smaller regional carriers or large enterprises processing thousands of exposures per transaction, the platform maintains both speed and stability across all operations.

Additional Scenario: Extreme Exposure Count

In certain lines of commercial insurance, exposure counts can grow to exceptionally high levels. Examples include large fleet auto policies covering thousands of vehicles, inland marine or equipment schedules listing every insured asset, and property portfolios for enterprises with extensive real estate holdings across multiple regions. In these cases, a single policy may contain hundreds of thousands of scheduled items.

While real-time responsiveness is not typically expected in such scenarios, Socotra's engineering team sought to understand how the platform performs under these extreme conditions and to validate its scalability limits.

To evaluate this, we ran a stress test scenario designed to push the system far beyond typical production usage. The test environment simulated a Commercial Auto product configuration with 500,000 scheduled items, each containing 254 individual data fields, representing a highly complex, data-dense structure. The test measured the time required to create, rate, and issue a complete policy under these conditions.

Socotra successfully uploaded a schedule containing 500,000 items—each with 254 data fields—in approximately two minutes, after which the policy was successfully rated and issued. This outcome is remarkable given the volume and complexity of data processed. Importantly, this result does not represent the platform's upper limit; internal engineering estimates suggest Socotra could support tens of millions of scheduled items before reaching meaningful performance boundaries.

Although these tests extend beyond the practical needs of most insurers, they illustrate the platform's architectural resilience and elastic scalability. Even under extraordinary conditions, Socotra demonstrates the capacity to scale predictably and handle data volumes that would exceed the design limits of most legacy policy administration systems.

Key Takeaways

The results across all test configurations—from typical commercial workloads to extreme-scale scenarios—demonstrate Socotra's ability to deliver reliable performance and predictable scalability. These findings reflect both the platform's technical efficiency and its architectural maturity, confirming its readiness to support the most demanding commercial insurance products.

- **Scalability at every level:** Even in intentionally high-load test conditions, Socotra's core platform showed linear and predictable scaling behavior. The Medium 2750 configuration—representing a workload far larger than typical production scenarios and running on mid-tier infrastructure—achieved stable and consistent performance. In real-world deployments, carriers with similarly large-scale products would provision higher-tier compute and database resources, as demonstrated by the Large 2750 configuration, which achieved faster throughput at a modest increase in infrastructure cost.

- **Performance consistency:** Across all tests, average latency remained approximately **75% of P95 latency,** indicating minimal variance and consistent performance even under sustained concurrent load.

- **Sub-500 ms responsiveness where it matters most**: Both **Submission Creation** and **Quote Issuance** workflows maintained **average latency below 500 ms,** even under high-load conditions, ensuring responsive user experiences for underwriters and agents.

- **Pricing/Rating scales efficiently:** While **Pricing/Rating** contributes the largest share of total latency for large schedules, scaling infrastructure reduced **average latency by 68%** and **P95 by 62%,** confirming that Socotra's architecture scales efficiently under complex rating conditions.

- **Balanced performance-to-cost ratio:** Test environments demonstrated efficient resource use at moderate infrastructure costs—**Medium 35** and **Medium 2750** configurations operated at **under $400/day,** while the **Large 2750 configuration** achieved measurable performance gains at **under $600/day.** This confirms that Socotra's elastic architecture enables carriers to scale performance in line with product complexity and demand without unnecessary cost.

- **Future-ready architecture:** The platform's ability to process **hundreds of thousands of exposures** and support **tens of millions of scheduled items** demonstrates scalability sufficient for large commercial lines and complex product portfolios.

- **Operational reliability:** Performance remained stable across all tested workflows and configurations, indicating that Socotra's architecture is built for consistent, real-world reliability and sustained throughput at scale.

Looking Forward

The 2025 Commercial Auto performance benchmarks reaffirm Socotra's commitment to continuous innovation and scalability. As carriers expand into increasingly complex product offerings—ranging from small fleets to multi-thousand-item commercial schedules—the demand for fast, reliable, and cost-effective core systems will only intensify.

These tests demonstrate that Socotra's cloud-native architecture can not only meet today's performance expectations but also scale far beyond real-world scenarios—handling 254-field schedules, 200 concurrent users, and even half a million scheduled items without degradation. Our engineering team's estimates of tens of millions of scheduled items as a practical upper limit showcase the platform's future-proof design.

Looking ahead, Socotra will continue to:

- Run regular latency tests to proactively identify and correct any bottlenecks that might be introduced as we expand platform capabilities—ensuring consistent, industry-leading performance as new features and product enhancements are delivered.
- Test newer configurations to showcase the scale and complexity Socotra can handle, aligning with the expanding needs of carriers and validating performance for emerging commercial product requirements.
- Collaborate with partners and carriers to ensure that testing scenarios remain representative of evolving market demands and operational environments.

By pushing beyond realistic workloads, Socotra proves it is prepared for the next decade of insurance innovation. Whether carriers are adding new lines of business, scaling embedded insurance partnerships, or modernizing commercial operations, Socotra delivers a core platform built to perform—today and in the future

Contributors

The following Socotra team members authored this document:

- Ivan Velasco, Staff Software Developer in Test
- Konstantin Kalin, Software Architect
- Sonny Patel, CPTO
- Dan Woods, CEO

socotra