socotra | aws

Boosting Insurance Carriers' Quote-to-Issue and Prospective Pricing Performance with High-Efficiency Core Platforms

Boosting Insurance Carriers' Quote-to-Issue and Prospective Pricing Performance with High-Efficiency Core Platforms

June 2024

Synopsis

In today's highly competitive, rapidly changing insurance landscape, carriers face increasing pressure to deliver efficient, seamless, and reliable services to their customers. To achieve a competitive edge, it is vital for carriers to improve the quote-to-issue process and rapidly price prospective business. High-performing core platforms provide the answer – maximizing efficiency, scalability, and innovation potential in a world of the customized and on-demand.

This white paper examines the performance of Socotra, one such high-performing core platform, revealing how it answers these needs and makes it possible to realize greater business potential *today*.

Background

The quote-to-issue process is critical to insurance carriers' performance, which is currently hampered by various challenges – including data silos, manual processes, regulatory compliance, and rigid legacy systems. High-performing core platforms offer a comprehensive solution that addresses these issues while enabling insurers to capitalize on advanced technologies and streamlined workflows.

Direct insurance is one major trend currently driving the need for high-performance core systems, since end customers are directly exposed to back-end system performance. They expect rapid responses, and are quickly frustrated by experiences that are noticeably slower than other consumer-facing websites. Slow customer interfaces may harm insurers' net promoter score (NPS), and may even cause customers to give up entirely – directly impacting conversion rates and direct written premium (DWP).

Performance is also critical in embedded insurance. Here, policies are quoted and sold through other consumer interactions, such as booking travel or scheduling package shipments. Insurers' retail business partners expect high performance, because it directly impacts their own customers' experiences, and they avoid partnering with providers whose systems are too slow to provide seamless transactions.

Challenges

Insurance carriers face several potential pain points in the quote-to-issue and quote pricing process that can impact efficiency, profitability, and customer satisfaction. Some of their primary challenges include:

Data Silos: Disparate data systems create siloed information across departments, making it difficult for carriers to obtain a comprehensive view of their customers or improve their underwriting decisions.

Manual Processes: Manual processes degrade organizational performance with slow response times, increased costs, and high potential for human error.

Regulatory Compliance: Insurance carriers must grapple with the complexities of ever-changing regulatory requirements, necessitating resource-intensive, urgent processes for data handling, underwriting decisions, and reporting.

Legacy Systems: Outdated legacy systems are ill-equipped to accommodate changing market conditions or respond to emerging market opportunities; many are still running on-premises, demanding copious maintenance and upkeep just to stay operational, let alone meet new challenges. Often designed for now-decades-old mainframe hardware, the code behind them is incapable of meeting the flexibility and scalability of modern, cloud-based solutions. Many of those that have been moved out of insurers' server rooms have simply been "lifted and shifted," even in the best cases still failing to capture the same value potential inherent to modern, cloud-native solutions.

The Solution: High-Performing Core Platforms

A robust, high-performing core platform helps insurance carriers address these challenges and improve their overall organizational performance. Key benefits of such a platform include:

Expanded Capabilities: High-performing core platforms enable advanced technologies such as artificial intelligence, machine learning, and advanced analytics to improve decision-making, underwriting accuracy, and risk assessment during the quote-to-issue process. Moreover, these platforms are built from the ground up to enable seamless integration with external technologies and boost operational efficiency.

Streamlined Workflows: By automating various tasks within the quote-to-issue process, high-performing core platforms reduce or eliminate manual processes, minimize human error, and improve responsiveness. Their optimized workflows allow carriers to capitalize on time-sensitive opportunities and better allocate resources to enhance productivity.

Improved Scalability: Modern core platforms help insurance carriers scale and adapt their operations to meet evolving market conditions and customer demands. By employing these cloud-based solutions, carriers can optimize infrastructure utilization, decrease operating costs, improve overall system performance, and meet challenges in real time.

Enhanced Security and Compliance: High-performing core platforms incorporate cutting-edge security measures to protect sensitive customer data and follow regulatory requirements. By offering compliance tools and automated reporting, these platforms make it easier for carriers to adhere to myriad regulations while reducing the risk of penalties and reputational damage.

Case Study: Socotra

AWS and Socotra conducted the performance test of Socotra's enterprise-grade policy administration system to show how it handles a variety of scenarios. Performance tests examined concurrent load on Socotra's PAS for the "quote-to-issue" flow and "quote-to-price" in particular,[1] using an existing tenant that had accumulated large amounts of data: over 20 million quotes and policies with proportionately high quantities of associated elements and pricing items. Results show that:

1. Existing data had negligible impact on transaction throughput and latency.
2. Socotra tenants on mid-range database and compute instances achieved throughput rates of over 11,500 of policies per minute (quote-to-issue), easily supporting 5,000 concurrent users[2].

3. Quotes can be brought through pricing – supporting comparative rating use cases – at even faster rates, with the platform providing over 12,500 price requests per minute from 5,000 concurrent users.

Two distinct test scenarios were executed against a typical personal automobile product configuration, modeled on a recent Progressive Auto filing in the state of Indiana:

1. Quote-to-issue
2. Quote-to-price

---

[1] Often, in industry vernacular, "Quote-to-Issue" is referred to as Quote-to-Bind, but these tests included the actual issuance of the policy as part of the process.
[2] Each "concurrent user" makes a transaction request to the API every 5-25 seconds, modeling typical behavior for users working quickly through a UI.

Figure 1 Scenario Overview

Both scenarios featured initial account creation and validation[3], followed by quote creation and subsequent progression to pricing (quote-to-price) or all the way to issuance (quote-to-issue). Policies had randomized coverage details to simulate realistic quote variance; for example, all policies had between one and three vehicle exposures, one to three drivers, and annual mileage ranging from 10,000 to 120,000.

Test results strongly indicate that additional performance gains can be realized at higher levels of provisioning on critical services such as RDS. Infrastructure costs in test scenarios held to $300 a day, for constant load; given the pricing model for metered infrastructure and auto-scaling capabilities, we expect that costs could be substantially reduced under operating conditions exhibiting periods of lighter use.[4] Infrastructure can be scaled on-demand to meet performance targets as needed, offering cost savings for less-demanding latency and throughput requirements.

Test Details

Objectives

The tests described in this document were executed to validate the quote-to-issue and quote-to-price workflows. The tests establish a baseline and determine a degradation point for the current Socotra development environment. This baseline will be used to detect future performance improvements or degradation.

---

[3] Accounts are synonymous with "policyholders" in these scenarios.
[4] For example, significant savings could be realized if assuming ~40 hours of active work from each user per week, as opposed to the 24/7 activity reflected in the stated constant load cost.

Test Overview

Both scenarios were executed on a typical production environment.[5] Multiple runs were executed to validate the results and detect any deviations and outlier results. Each run was executed for 15 minutes of constant load, plus additional time for warming/cooldown ramps for concurrent users.[6]

The following plugins were enabled for both scenarios:

1. Validation
2. Pricing

The quote-to-issue flow also included an Underwriting plugin.[7]

Each plugin was implemented with logic representative of real-world expected usage. Requested throughput was:

- Scenario 1: 11,500 quote-to-issue per minute.
- Scenario 2: 12,500 quote-to-price per minute.

Test Infrastructure

The following diagram provides 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.

---

[5] See the appendix for an architectural overview of the platform, along with provisioning details.

[6] Warming and cooldown periods were one minute each. Ramp-up exhibited steady linear growth from zero active users to the target quantity over the span of one minute; likewise, cool-down decreased the peak user quantity back to zero in a linear fashion mirroring ramp-up. Together, ramp-up and cool-down represented one minute of additional constant load, for a total of 16 minutes of constant load.

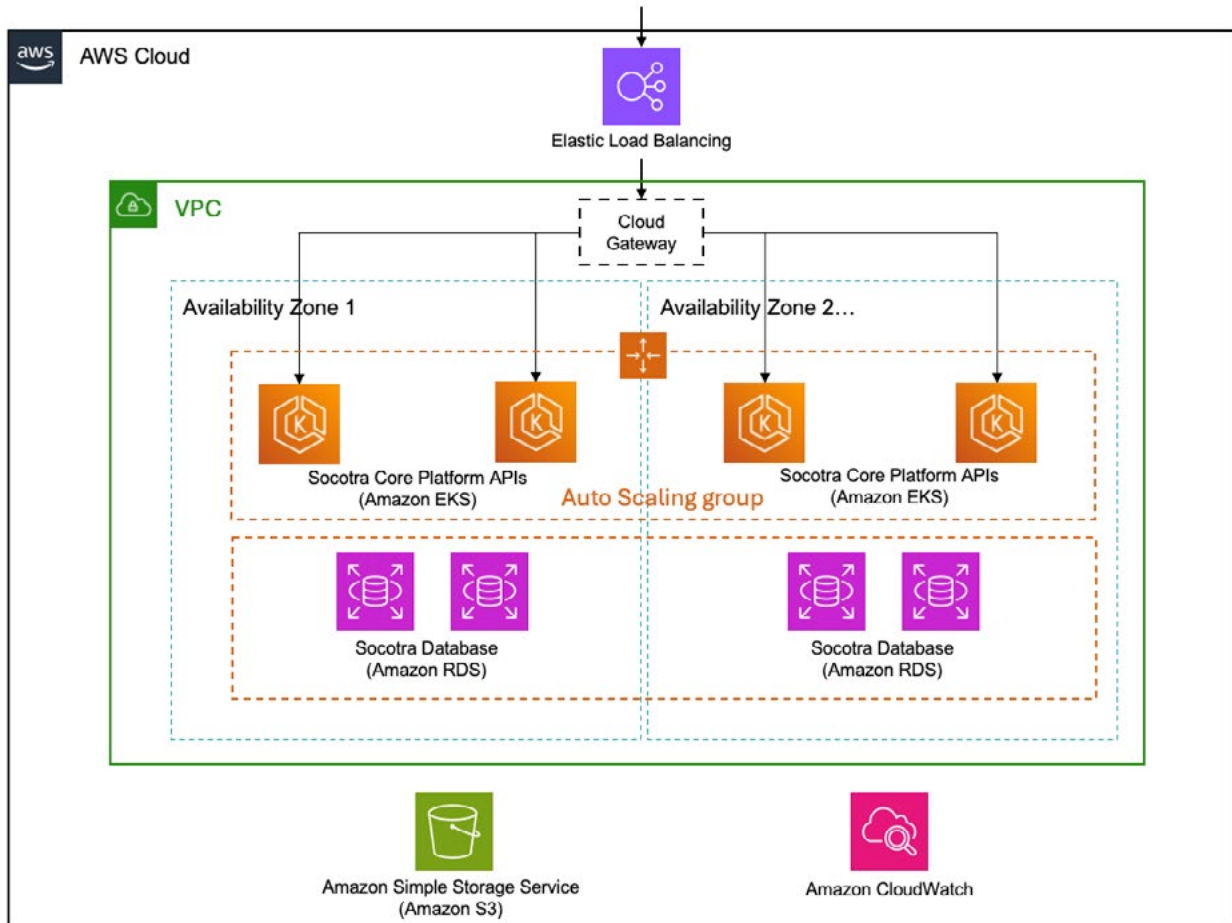[7] These are the typical plugins implemented in production.

Figure 2 Test Infrastructure Overview

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

AWS Service Instance Types

- RDS:
    - MariaDB: db.m6g.8xlarge
    - Postgres (1): db.t3.small
    - Postgres (2): db.m7g.2xlarge
- EC2: m6a.xlarge

Autoscaling

Services in the EKS cluster started with 2 pods each, with the exception of the billing service and policy service, starting with 3 and 5 pods, respectively. By end of testing, each service had scaled by one additional pod, with UI and provisioning services increasing by 2.

- **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 quotation.

- **Quote Create**: Creates a quote in draft state.

- **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 Results

Results for both scenarios are depicted in two graphs, the first for the 95th percentile of all requests, and the second for the average of all requests. Each graph shows results for full-duration tests in increments of 1,000 users, up to 5,000 concurrent users. For example, a 95th-percentile graph at the 2,000-user value should be interpreted as "a complete test with load of 2,000 virtual users saw 95% of all requests completing within the time indicated on the y-axis."

Each graph line corresponds to one of the API calls for the quote flow illustrated in Figure 1 and described in "Target APIs", above. Both quote-to-issue and quote-to-price include API calls for account creation and validation, followed by quote creation, validation, and pricing. Quote-to-issue also includes subsequent "underwrite", "accept", and "issue" API calls. These are all part of Socotra's open API, the details of which can be found on docs.socotra.com.

Scenario One: Quote-to-Issue

In this scenario, we observed quote-to-issue handling of 11,500 policies per minute with 5,000 concurrent users. 95% of each of the API requests in the quote-to-issue flow completed in less than 300 milliseconds (ms), indicating consistently fast performance for each transaction type:

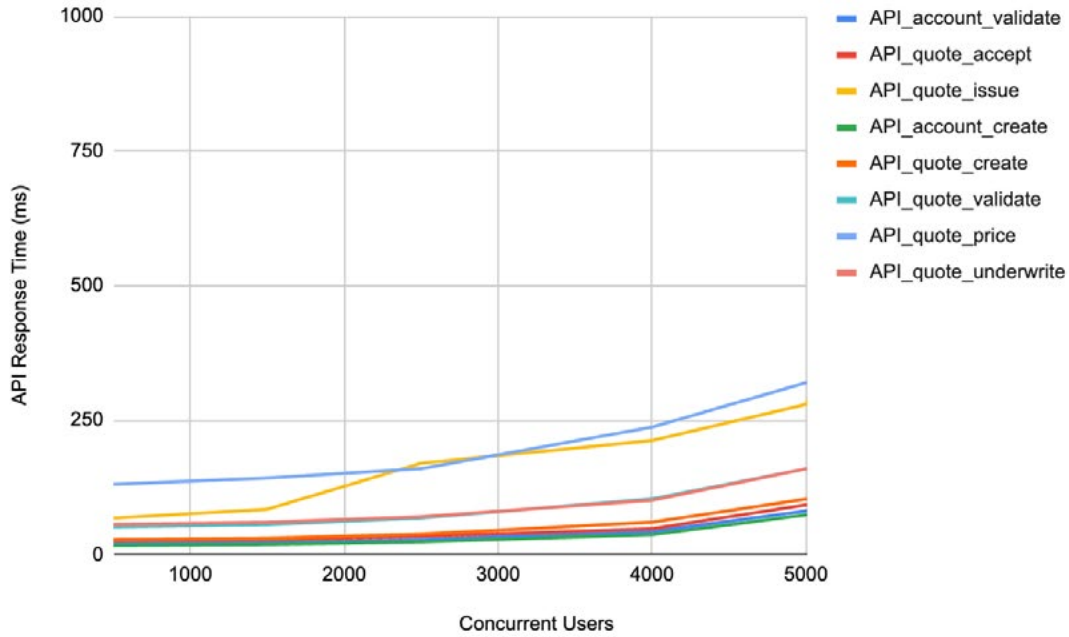## Load Test
Quote to Bind 95th Percentile

Figure 3 Quote to Bind 95th Percentile

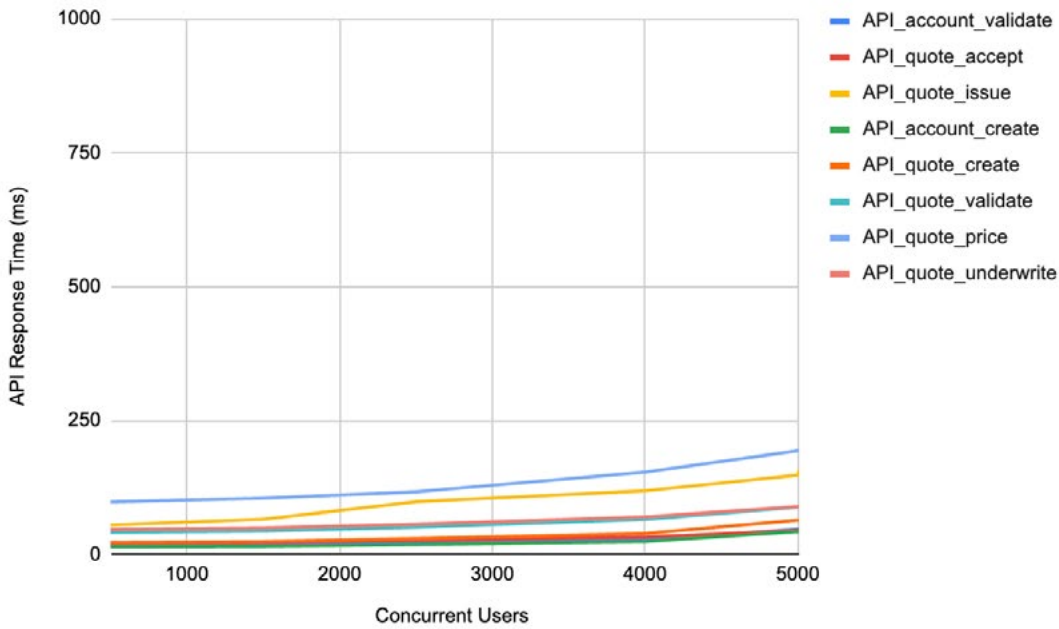## Load Test
Quote to Bind Average

Figure 4 Quote to Bind Average

Scenario Two: Quote-to-Price ("Comparative Rater")

The following graphs show the response time met by 95% of requests and the average of all requests, respectively. In this scenario, Socotra achieved rates of 12,500 quote pricings per minute with 5,000 concurrent users. 95% of each of the API requests made to bring a quote from creation through rating completed in less than 250 milliseconds (ms), again exhibiting consistently fast performance, with negligible growth in response times under increasing loads.

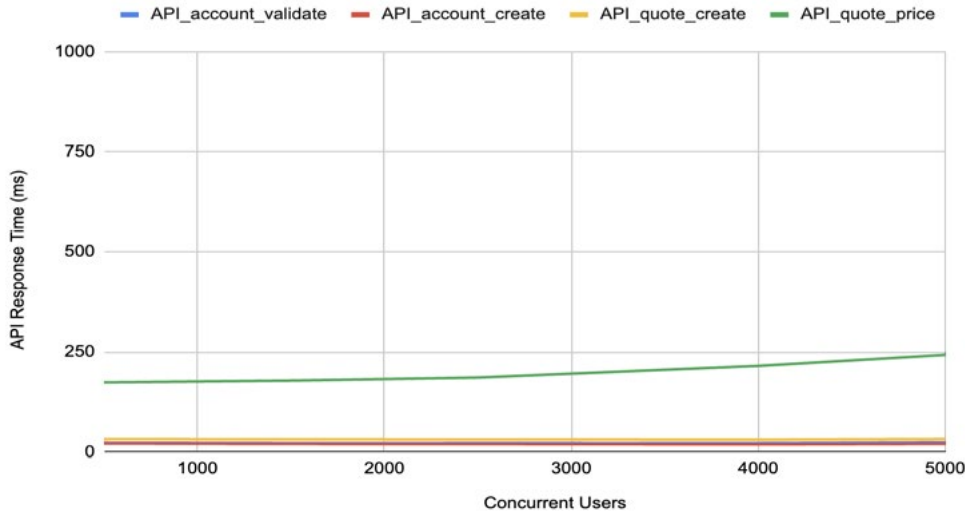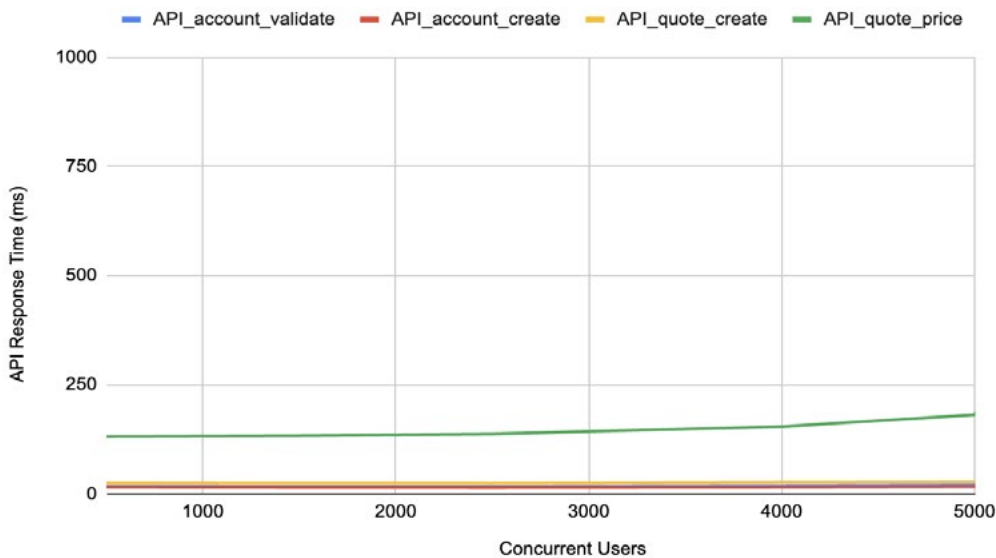*Figure 5 Comparative Rater 95th Percentile*

Figure 6 Comparative Rater Average

During testing, we observed one short period of increased latency due to auto-scaling, with negligible impact on throughput. This attests to the effectiveness of auto-scaling in production scenarios for platforms like Socotra, which are optimized to take advantage of horizontal scaling.

Figure 5 Latency during Test Period

Looking Forward

Improving technical performance is crucial for insurance carriers looking to transform their quote-to-issue processes and boost overall efficiency. High-performing, seamlessly-integrated core platforms enable carriers to break down data silos, automate manual processes, adhere to regulatory requirements - and retire legacy systems. By capitalizing on advanced technologies and streamlined workflows, insurers stand to gain competitive advantages that can redefine their positions in the sector and drive future success.

Socotra is thrilled with the results of this case study, and believes that the performance of its core platform positions the company as a true market leader in policy administration systems. With so many insurers still relying on outdated technology that cannot meet the demands of today's consumer, Socotra is proud to showcase the results of this testing as proof of its competitive position in the insurance technology marketplace.

"With this paper, Socotra has demonstrated the fastest audited PAS performance metrics to date, and done so on inexpensive, high-reliability cloud infrastructure. It delivers the highlighted benefits of cloud-native high performance, with the flexibility required to power a wide range of insurance product structures, across P&C and life. In a world of increasing IT scale, throughput, and responsiveness expectations, all PAS vendors will need to subject their systems to comparable performance tests to remain relevant. Socotra is proud to bring this level of vendor maturity to policy administration, where it is badly needed." – Sonny Patel, Chief Product and Technology Officer, Socotra

Contributors

The following individuals authored this document:

- John Duffy, Partner Manager, Socotra
- Konstantin Kalin, Software Architect, Socotra
- Sam Shay, Creative Director, Socotra
- Mark Szymanski, Senior Product Manager, Socotra

Special thanks to the following individuals for their contributions:

- Rachel Beck, Director of Marketing, Socotra
- Artemis Berry, Senior Partner Development Manager, Amazon Web Services
- Kevin Bolger, Director of Product, Socotra
- Terry Buechner, Principal Insurance Specialist, Global Core Systems Lead, Amazon Web Services
- Chintan Sanghavi, Sr Partner Solutions Architect, Amazon Web Services
- Ivan Velasco, Sr Software Developer in Test, Socotra
- Dan Woods, CEO, Socotra

socotra

aws

WWW.SOCOTRA.COM

sales@socotra.com