Building E-Commerce Order Management Systems - Technical Article #1 - Architecture

We at TrackIt have recently published an article on medium.com titled 'Rebuilding a Complete Integrated Order Management System' in which we've provided readers with a comprehensive understanding of the multiple considerations to make from a business standpoint when implementing order management systems (OMS). Seven key considerations were discussed, namely reliability, resilience, feature flexibility, reporting, accounting, cost-effectiveness, and region-specific compatibility and adaptability.

This article aims to tackle these 7 key considerations from a technical perspective in order to provide readers with a concrete understanding of the different strategic and architectural choices they can make when building order management systems.

Due to similarities in the strategies used to address them, we have divided these 7 key considerations into three separate categories and we'll provide readers with strategic and architectural suggestions to consider for each.

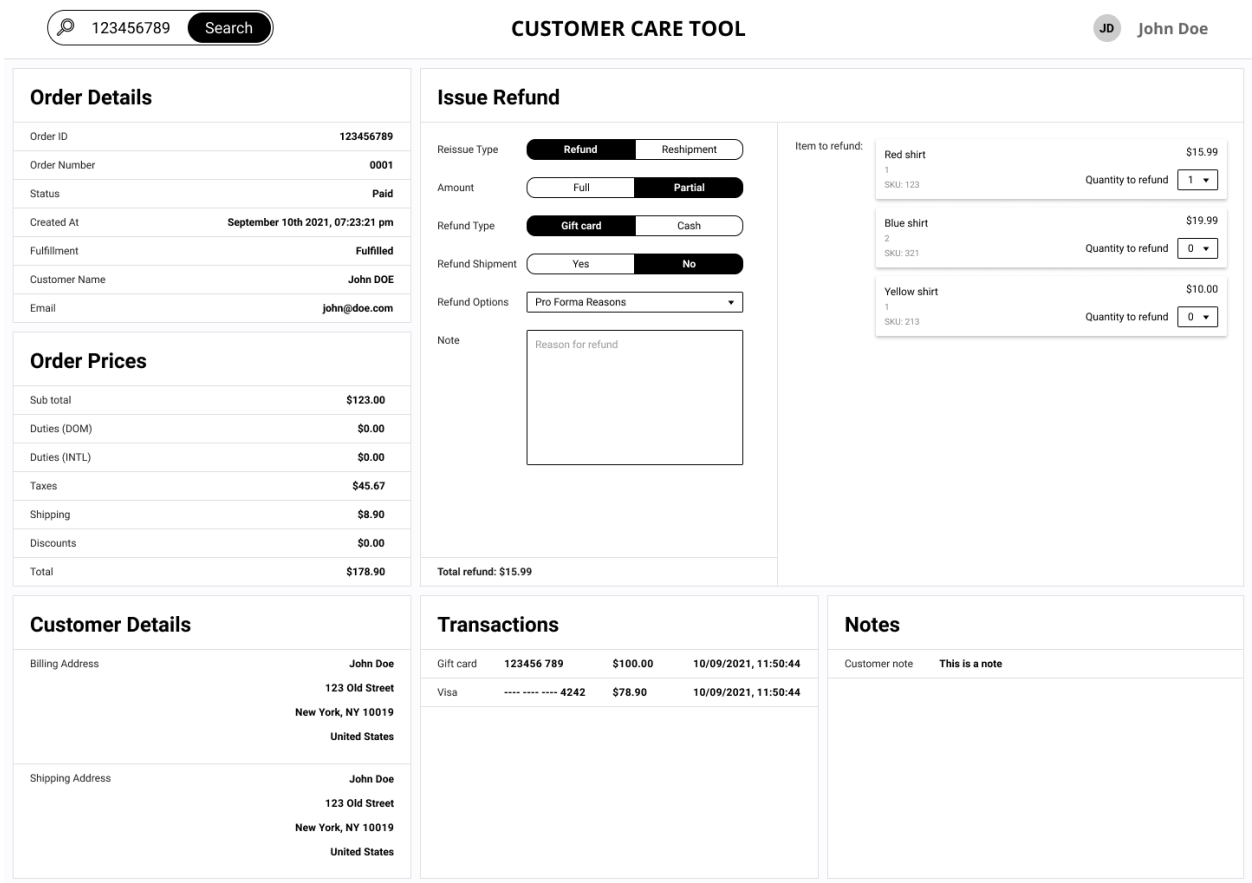
Category #1: Reliability (No Downtime & Scalability) and Cost-Effectiveness

In order to achieve *Reliability* and *Cost-Effectiveness*, companies can consider implementing a solution with Serverless architecture used along with AWS Step Functions that use AWS Lambdas to automate and facilitate the order management process.

Adopting a Serverless approach has two main benefits. First, Serverless helps companies eliminate tedious infrastructure management tasks such as capacity provisioning and patching. Second, Serverless is very cost-effective for the majority of modern enterprises. It offers scalability and ensures that companies only pay for what they use.

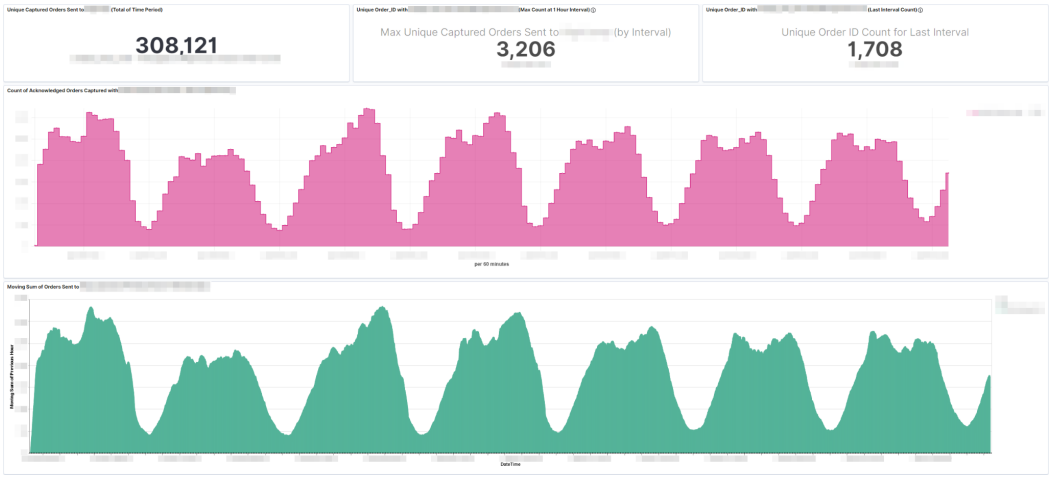
Category #2: Feature Flexibility, Technical Reporting, Resilience (Error tolerance), and Region-Specific Compatibility & Adaptability

To achieve *Feature Flexibility*, companies benefit from implementing an in-house order management system that runs independently versus relying on a 3rd party order management solution which limits them to the vendor's features & infrastructure choices. Choosing to build a solution in-house enables a company to add handpicked custom features such as the customer care tool showcased in the screenshot below:

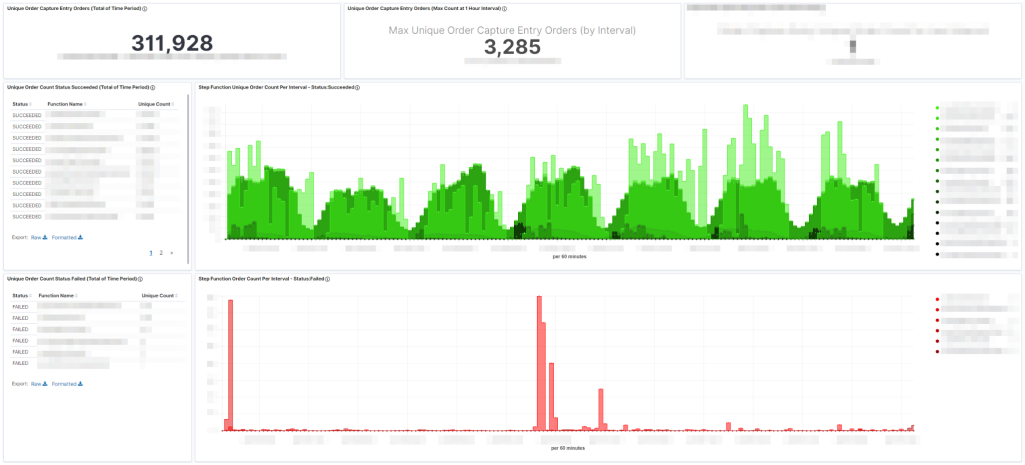


Customer Care Tool

Technical Reporting requires a logging and monitoring service such as OpenSearch Service (previously known as Amazon ElasticSearch) that stores OMS logs. For data visualization, a service like Kibana could be used to visualize and thoroughly monitor the OMS.



OMS Data Visualization using Kibana

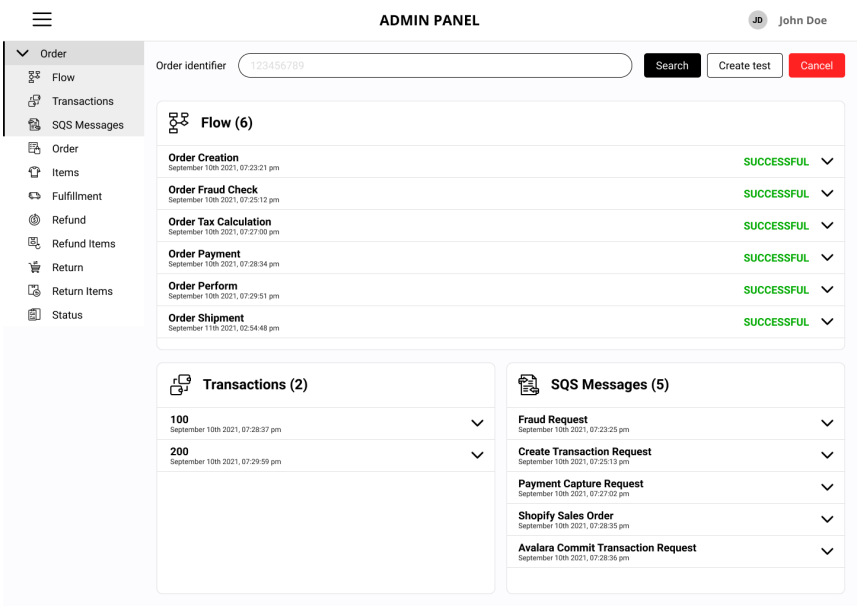


OMS Data Visualization using Kibana

For *Resilience,* i.e. error tolerance, companies need to establish robust error resolution mechanisms that ensure that errors are resolved with maximum efficiency. One possible strategy that companies could use is to establish sequential error resolution systems such as the one described below in which resources are dedicated to error resolution only when necessary:

Stage 1 - Automated error resolution: The solution is designed to automatically resolve typical minor errors and errors on external 3rd party solutions without human intervention. For example, if the payment gateway is unable to capture the payment, we can automate the process of retrying the payment after 5 mins.

Stage 2 - Resolution by non-technical staff: When the OMS isn't able to resolve the issue on its own, the error is redirected to an administrative panel where non-technical staff can quickly resolve it. The following screenshot is of an OMS administrative panel that staff can use to quickly resolve minor issues.



OMS Admin Panel For Rapid Error Resolution by Non-Technical Staff

Stage 3 - Engineer intervention required: If the error cannot be resolved by non-technical staff, it is sent to the engineers. The previous error-resolution stages ensure that engineers only spend their time focusing on issues that cannot be resolved without their intervention.

For *Region-Specific Compatibility & Adaptability*, the OMS needs to be able to recognize foreign language characters and emojis (which are sometimes used by customers to differentiate between different addresses, home and office, etc.) and it needs to ensure that these characters are carried forward till the end of the order management process (emojis are filtered out).

Similarly, the OMS also needs to be programmed to effectively switch between currencies and execute the orders/transactions in the customers' currencies. This could be used to ensure that customers only see prices in their currency on the website, on their invoices & receipts, and also on shipping and customs labels.

Batch #3: Accounting

To address *Accounting* needs, the OMS needs to save as much data as possible for as long as possible. A robust database like Redshift is a good candidate to store business data. Internal data that is used to run the OMS can be stored on a separate database such as Amazon Aurora. The database used for reporting could be connected to a data exploration and business intelligence service such as Looker to perform queries and visualize data.

In some cases, companies may choose to use a separate database to store accounting data. This 'accounting database' could then be connected to an accounting solution such as NetSuite to handle accounting operations.

Conclusion

Architectural choices can have a considerable impact on the performance, flexibility, scalability, efficacy, and cost-effectiveness of an OMS. The various strategies discussed in this article help shed some light on the different choices companies can consider to design and build robust order management systems.

About TrackIt

TrackIt is an Amazon Web Services Advanced Consulting Partner specializing in cloud management, consulting, and software development solutions based in Venice, CA.

TrackIt specializes in Modern Software Development, DevOps, Infrastructure-As-Code, Serverless, CI/CD, and Containerization with specialized expertise in Media & Entertainment workflows, High-Performance Computing environments, and data storage.

TrackIt's forté is cutting-edge software design with deep expertise in containerization, serverless architectures, and innovative pipeline development. The TrackIt team can help you architect, design, build, and deploy customized solutions tailored to your exact requirements.

In addition to providing cloud management, consulting, and modern software development services, TrackIt also provides an open-source AWS cost management tool that allows users to optimize their costs and resources on AWS.