Rocket® software

Innovation Without Disruption

Driving Incremental Modernization of Mainframe Data Pipelines

TM

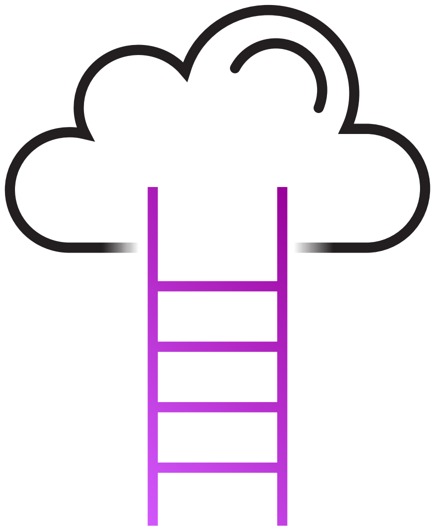
Introduction

We live in a golden age of innovation where change has become the foundation of human existence. Businesses are constantly adapting to customer-driven expectations for change by relentlessly pursuing innovation. Yet, the term "innovation" often conjures up images of disruptive technologies and radical shifts capable of unsettling the status quo. Although disruptive innovation has the potential to be a formidable force for generating new opportunities and fueling growth, it may not always be the most suitable approach.

In certain instances, disruption can result in alienated customers, destabilized business models, and even the collapse of well-established companies. So, what is the alternative? How can businesses maintain their innovative edge without causing excessive disruption? This brochure delves into the concept of "incremental innovation,", specifically focusing on managing complex data pipelines from mainframe systems. It demonstrates how incremental improvements can significantly enhance existing products and services, rather than necessitating the creation of entirely new ones.

Mainframe data access challenges abound

In most data-centric organizations, complex data pipelines ingest data from various sources for processing, analysis, and visualization. Due to the high costs associated with the extensive computing and storage requirements for analytics, as well as and the availability of open-source tools, these complex data pipelines typically reside outside the mainframe environment. However, as mainframes continue to serve as an organization's central platform, access to real-time or near real-time data for comprehensive analysis remains crucial.



One method for transferring data between disparate systems that has garnered significant attention from the technical community is streaming, which offers real-time or near real-time access to data for services operating in data centers or cloud environments. Mainframe environments, however, lack native implementation of a service model that enables multiple clients to securely access mainframe data sources without overloading mainframe infrastructure with request-response traffic processing.

For instance, a mainframe-native application streaming data to an external client typically requires middleware and a connector (often a Java-based implementation in Unix System Services) to connect to an external host for data stream transmission. The latency incurred by such a complex architecture can be undesirable for real-time requirements. Moreover, in a heterogeneous environment involving data transfer from both mainframe and distributed applications, there is no native pathway to a standardized set of services to enable data streaming and processing—this further emphasizes the need for configuration management.

A Rocket Software solution

Anomaly detection framework

Rocket Software has developed a multifaceted strategy to address this issue. The initial aspect of this approach entails creating a z/OS® native implementation optimized for streaming data within mainframe applications. These z/OS native applications can asynchronously utilize streaming services to transmit data to subscriber systems, regardless of their environment, enabling clients to function in distributed, cloud, or mainframe settings.

The second component of the approach involves separating the data aggregation module responsible for collecting data from various sources to enhance scalability.

A framework of services

Rocket® Software has successfully developed a framework of services, adapted from the open-source space, to operate as z/OS-native within an anomaly detection framework. This framework supports a broader range of operating systems, including z/OS, and offers a streamlined interface for consuming, installing, and monitoring essential framework components. In contrast to solutions from other Independent Software Vendors (ISVs), the services designed for detecting anomalies in instrumentation data enable developers to equip legacy mainframe components to stream data without requiring a foreign function interface (such as Java Native Interface [JNI]) in their software implementation. Interfaces like JNI add complexity to the design and increase communication latency between code running inside the Java Virtual Machine (JVM) and z/OS native code in Assembler or C/C++.

High-level solution architecture

For integration purposes, the anomaly detection framework provides a unified layer through its API abstractions, simplifying data movement between applications in different environments. The framework handles data caching to address potential back-pressure situations and prevent resource overload. The design's central element is Apache Kafka, an open-source data exchange technology available as a container image for data centers and as a managed service in the cloud environment. Numerous enterprise IT establishments trust Kafka for real-time, big data streaming solutions. This near-real-time data streaming approach enables cluster architecture, granting customers complete control over their environment and scalability.

Rocket Software's anomaly detection framework boasts an extensive set of integration capabilities with other data analytics offerings within the Apache. In the latter part of the design approach, the data aggregation component gathers information from various sources. It features a loosely coupled design to enable granular control over creating and configuring multiple aggregator instances. Each instance operates independently, ensuring parallelism and a high degree of fault tolerance, and is configured separately for optimal resource utilization. As data throughput requirements across source applications may vary, each broker instance is configured to be polled by data consumers independently based on data availability, reducing the consumption of critical system resources.

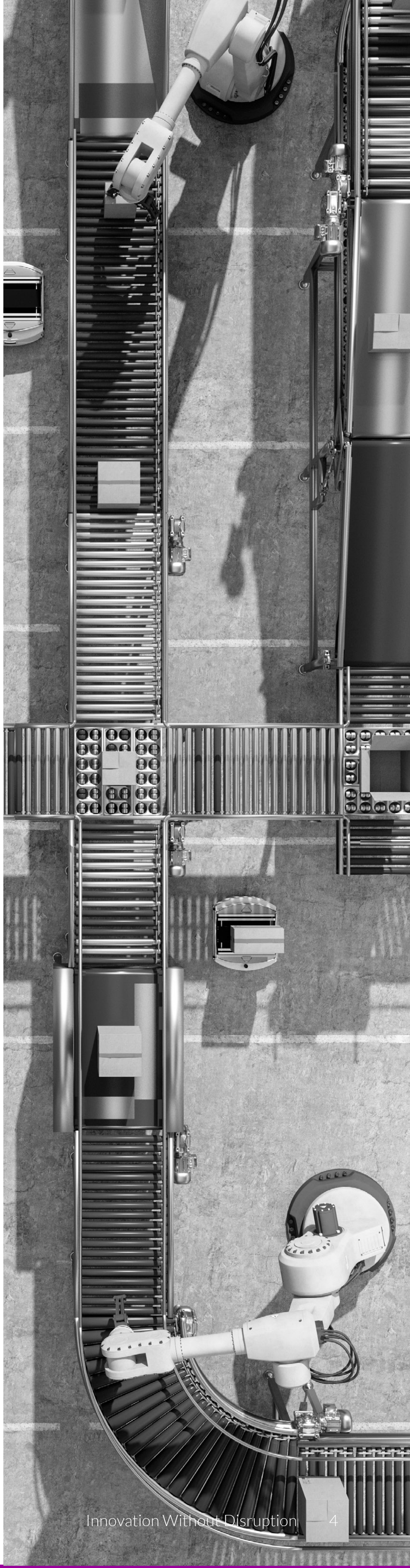


Figure 1 below illustrates the platform-agnostic, highly scalable end-to-end application interaction. On the far left are workload monitoring applications deployed within z/OS, integrated with an anomaly detection framework to publish data to the Kafka event streaming platform in real-time. The data is published to a specific topic asynchronously, eliminating the need for data source applications to wait for server acknowledgment and allowing for data publishing without latency.

The component labeled "Real-Time Event Streams" is a Kafka cluster consisting of one or more servers spanning multiple machines, data centers, or cloud regions. The cluster is configured to form the storage layer, called a Kafka broker. A Kafka cluster can be designed and configured to be highly scalable and fault-tolerant; if any server fails, others ensure continuous operations without data loss.

The items labeled with "Consumer" are Kafka consumers, which are custom implementations containing logic to read, write, and process streams of events in parallel, at scale, and in a fault-tolerant manner, even during network issues or machine failures. The framework has a custom-built Kafka consumer to read event streams, translating data from native z/OS EBCDIC encoding to JSON or AVRO and storing it in a PostgreSQL instance, which is independently scalable to handle many requests. Binary output is available, but JSON should suffice, as it is encoded using UTF-8 format. Like the producer, the Consumer is subscribed to a particular topic and registers a callback function to handle events it receives notifications for from the Kafka broker.

For publishing and subscribing, applications integrating into the anomaly detection framework use abstracted APIs to invoke native interfaces. This abstraction prevents service callers from dealing with implementation complexities and configuration details for managing producer. For example, a data-producing application does not need to know anything about a producer.

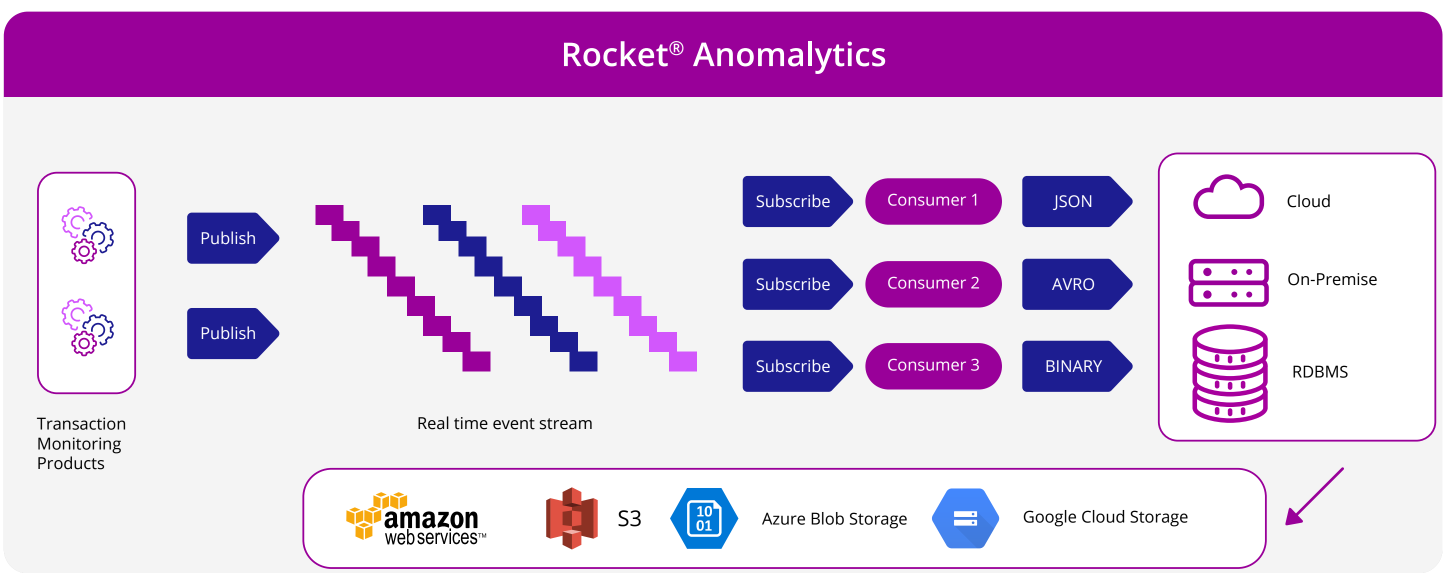


Figure 1

01

Online, offline, and near real-time operation

The platform encompasses a multifaceted approach, including:

- An HTTP/REST-based or services-based system that enables web-based applications to poll for data by subscribing to specific topics and using a consumer interface to receive JSON output
- Batch processing or offline systems that ingest large volumes of data for processing, supporting the development of machine learning models
- Stream processing or near real-time systems that consume input and produce output based on events, such as a new message at a specific topic triggering a notification for the handler to consume the message

02

Cloud and on-premises compatibility

As a platform-agnostic system closely integrated with Kafka, the anomaly detection framework operates in all environments, including cloud and data centers running any operating system—Linux, Unix, Windows, and z/OS.

03

Managed data storage

Backup pressure concerns, which prevent the broker from being overwhelmed with data during high throughput from data-producing applications, are addressed using a data partitioning approach. This method breaks a single topic into multiple logs, allowing each log to reside on a separate cluster. Consequently, storing and processing messages are manageable and can be distributed across various nodes within the cluster.

04

Enterprise security

The framework fully supports enterprise-level security through role-based access, TLS 1.2 encryption, audit logging, and secret protection.

05

Performance and scalability

By design, the system scales in cloud, data center, or hybrid environments, enabling significant performance enhancements. The data processing components in the anomaly detection framework are containerized, allowing granular control over resources available for consumption as demand for each component fluctuates. This can result in substantial cost savings in the cloud and hybrid environments. Tools like Kubernetes in a data center environment or managed services like Kafka Cloud are excellent for automatically scaling up and down distributed components to handle multiple tasks for simultaneous data processing. Another cost-saving strategy for batch jobs and fault-tolerant workloads is using spot instances instead of on-demand instances, as cloud providers typically offer discounts of up to 90%.

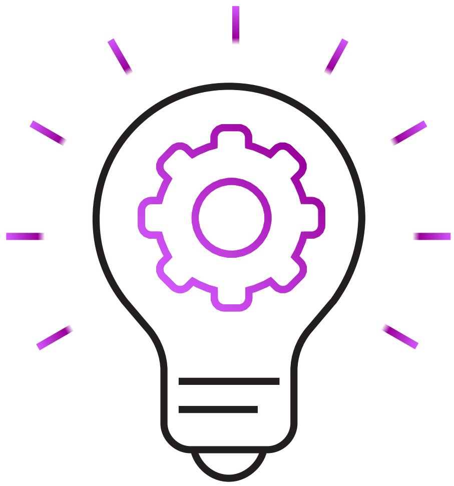
06

Concurrent design

As a platform-agnostic system closely integrated with Kafka, the anomaly detection framework operates in all environments, including cloud and data centers running any operating system—Linux, Unix, Windows, and z/OS.

Innovate, incrementally

Businesses today face many obstacles, such as security issues caused by a widely dispersed remote workforce, currency fluctuations brought on by a recession, rising hardware and software costs, and a need for more high-end skills to stay up with technological improvements. Additionally, customers may feel frustrated after making substantial technological investments to have it swiftly become obsolete.



Having access to and using cutting-edge technology provides a competitive advantage. It is expected that technological investment will increase year over year. Budgets, however, for technology can be moderate because of this. The return on investment can be increased with careful planning. A future-proof strategy will ensure that the technological components are interoperable and scalable for the future, depending on the organization's needs.

Rocket Software's anomaly detection framework provides a uniform interface compatible with existing enterprise systems. The innovation the anomaly detection framework brings is incremental to your existing mainframe investments, making it future-proof for any integrations planned in the times ahead as you embrace change and growth in your business, customers, and infrastructure.

About Rocket Software

Rocket Software partners with the largest Fortune 1000 organizations to solve their most complex IT challenges across Applications, Data and Infrastructure. Rocket Software brings customers from where they are in their modernization journey to where they want to be by architecting innovative solutions that deliver next-generation experiences. Over 10 million global IT and business professionals trust Rocket Software to deliver solutions that improve responsiveness to change and optimize workloads. Rocket Software enables organizations to modernize in place with a hybrid cloud strategy to protect investment, decrease risk and reduce time to value. Rocket Software is a privately held U.S. corporation headquartered in the Boston area with centers of excellence strategically located throughout North America, Europe, Asia and Australia. Rocket Software is a portfolio company of Bain Capital Private Equity. Follow Rocket Software on LinkedIn and Twitter.

The future won't wait—modernize today.

Visit RocketSoftware.com ›

Talk to an expert

MAR-6314_InfraMod_Broch_Anomalytics_V2