

Diffusion® Real-Time API Management Platform™



The Diffusion Real-Time API Management Platform provides a single, secure, unified means of distributing and synchronizing a shared data model across all connected applications and systems. Using Diffusion as an edge-tier layer, developers no longer need to invest time and effort into connecting varied and disparate back-end data sources to front-end applications, and the operational overhead of exposing and scaling multiple back-end services across the Internet is minimized.

Service Integration

A common concern with system architectures is the need to extend back-end data sources across the Internet. While some MQ products provide SDKs for front-end environments, these are typically less feature-rich and often rely on inefficient transports such as polling which reduces overall system performance. Moreover, other products – such as Kafka – are simply not designed to be used outside of internal data centers and have few provisions for such operation. Even when back-end sources can be connected to front-end applications, developers must still contend with integrating and managing multiple SDKs while ensuring consistent quality of service and security rules.

Diffusion radically simplifies the process of extending back-end data sources across the Internet by providing adapters for a variety of common data sources such as REST APIs, Kafka, JMS & MQTT. These adapters allow multiple back-end services to publish data into Diffusion, with front-end applications consuming this data through a single Diffusion SDK.

By consolidating the means of data distribution and access, the complexity of integrating necessary data sources in front-end applications is greatly reduced while facilitating easier creation of new features and capabilities.

"For operational teams, Diffusion provides a central point of control for configuring quality-of-service and data access policies, and reducing the overhead of managing and scaling application data across the Internet."

Synchronized Data Model

All data published to Diffusion is retained in the form of a hierarchical topic tree. Consumers of data (e.g. front-end applications) can use Diffusion's Client SDKs to subscribe to one or more topics in order to receive each topic's data via a publish/subscribe access pattern. Updates to topics are automatically sent to any subscribers in real-time, providing a live distributed data model. Because each consumer can subscribe to specific topics, developers have granular control over precisely what data they receive at an application level.

"Diffusion decouples data production from consumption which radically simplifies the management and evolution of front-end applications, by allowing front-end developers to implement new features that require specific data without being concerned about the location of the data source."

Centralized Quality-of-Service

Diffusion introduces key quality-of-service optimizations at the transport layer. By acting as a central distribution mechanism, Diffusion can apply per-connection flow-control to ensure that applications receive data at a rate that they can handle. This prevents back-end systems from being bottlenecked by slower consumers of data. Even unoptimized systems can be effectively utilized with no additional development work required to cope with higher volumes of traffic. Diffusion's network-first design uses minimal overhead for each connection, supporting far higher numbers of concurrent connections on a single instance (approx. 70k) than traditional back-end MQ products are designed to handle. Diffusion only requires a single connection for back-end data sources to publish data; therefore, developers use Diffusion as a general-purpose scaling layer, eliminating the work to scale multiple back-end data sources. Further, Diffusion provides a common target for tuning and monitoring data delivery across multiple systems and applications, removing the need to configure each individual back-end service separately in order to achieve maximum performance. The use of Diffusion as middleware between back-end and front-end simplifies modification or replacement of individual components and eliminates affects on other components or service availability.

Security

When clients connect to Diffusion, a back-end system can provide a set of Roles which will be applied to the new connection.

Each Role provides a granular set of Permissions that control what a connection is able to do and what data it is able to receive. Because these permissions are stored server-side, connections need only authenticate once; there is no need to keep requesting authorization tokens for privileged operations, which reduces network traffic and simplifies application logic. Using Diffusion to handle front-end security policies reduces the need to duplicate policies across multiple back-end services that would otherwise be connected directly to these front-end applications, which simplifies the implementation of security policies as well as making it easier to track and audit any possible violations. Because Diffusion applies these same security policies to any connections to back-end data sources, dev-ops teams can still control precisely what data is exposed via Diffusion to front-end applications, reducing the risk of data leakages or attack vectors for malicious actors.

Reduced Infrastructure

Because of Diffusion's ability to handle high volumes of traffic and data, the number of servers required for production deployments is vastly reduced compared to those required for less performant approaches. Instead of having to deploy multiple instances of various REST or MQ servers, with separate load-balancer pools for each back-end service, Diffusion can be used to consolidate edge-tier infrastructure - driving down cost and simplifying the management of architecture deployment.

"Using Diffusion as a Single Point-of-Access for your system's distributed data model radically simplifies: architectural complexity, operational management, & front-end development."

The ability to extend traditionally bottle-necked, back-end, data sources across the Internet in a simple and scalable manner provides new business opportunities, while reducing the time and resources required to maintain existing infrastructure."