

# Diffusion<sup>®</sup> & Messaging for Today's Demanding Applications



*The Diffusion Real-Time API Management Platform<sup>TM</sup> synchronizes, manages, and distributes data among applications, devices, and systems – via web, mobile, and satellite networks.*

*Diffusion can be used to handle all of your application data streaming and messaging needs and it can extend the reliability, scalability, and functionality of in-place messaging products.*

## **The Messaging Paradigm**

Messaging as a paradigm for delivering data is well established; countless protocols and products are used for messaging applications. The fundamental principal of maintaining a queue of messages, which may be routed to one or more receivers, maps to many common requirements within back-end systems.

## **Changing Development Requirements**

With the rise of Micro Services, reliable & scalable data distribution is increasingly critical for business applications across industries. Micro Service architecture is a method of developing software systems that structures an application as a collection of fine-grained, loosely coupled services. The inherent scalability of this structure is ideal for supporting a wide range of platforms and devices—spanning web, mobile, Internet of Things, and wearables—or when future device support needs are unclear. The benefit of decomposing an application into discrete smaller services is that it improves modularity and makes the application easier to understand, develop, and test.

The shift in application development focus has led to development of new data handling products to support changing business application development requirements. For example, Apache Kafka, which mixes traditional Message Queue functionality within a Pub/Sub model, is a popular tool for handling Event Processing/Distribution due to its ability to decouple producers and consumers of data.

## **The Application Development Objective**

The overarching objective for modern application development teams is to establish, monitor, and manage data pipelines that extend Enterprise back-end systems through to the constantly proliferating array of end-user devices, connected over a variety of unreliable networks. To satisfy business mandates, the applications must deliver consistent, reliable, high performance and operate in a cost-effective manner.

## Traditional Messaging Assumptions

Products that handle data are designed with a set of underlying assumptions regarding the underlying networks and environments in which they will operate successfully.

Most MQ products are designed with the following assumptions:

- Bandwidth will always be sufficient and inexpensive.
- Latency will be reasonably consistent.
- Network interruptions will be quickly resolved.
- Per-node traffic volume will remain relatively low and consistent.
- Target platforms are primarily back-end systems.

Unfortunately, the huge consumer demand for web and mobile applications, coupled with the explosion of IoT data, invalidates these assumptions, resulting in substandard application performance and substantial added work for development teams.

## Messaging Product Challenges

MQ products typically do not handle extremely high load - such as a popular application or website with hundreds of thousands of concurrent users – without degrading performance or requiring large server cluster infrastructure.

Many MQ products are not designed with a Web/mobile-first approach. This impacts application responsiveness and user engagement when bandwidth is constrained or networks are congested.

In addition, MQ products often lack purpose-built web or mobile-specific SDKs, complicating development projects and lengthening time-to-market.

To overcome the MQ product challenges, developers are faced with the time-consuming task of implementing additional functionality on top of the core MQ product and increasing the complexity of their infrastructure.

The bottom line is increased cost, application development and infrastructure complexity, and delivery time for development projects.

## The Solution

The Diffusion Real-Time API Management Platform is designed to simplify the development of applications that need to transport data across unreliable networks. Today's application development challenges - high load, variable traffic, unreliable connections, and complex implementations - are better served by higher level paradigms such as Pub/Sub and Point-to-Point messaging to deliver reliable, high-quality and consistent application performance.

### *Unique Delta Data Streaming*

Diffusion's real-time Pub/Sub data streaming is optimized for delivery speed and network, bandwidth, and infrastructure efficiency:

- 1) Diffusion only sends data to subscribers when the value of a topic changes. This avoids clients being forced to continually poll for new data, reducing both code complexity and network usage.
- 2) Diffusion only sends the differences (deltas) between topic updates, with automatic value reconstruction at the client side, drastically reducing the amount of data sent over the network.

Diffusion's delta-data steaming eliminates many MQ product challenges and provides substantial efficiency, reliability, and resiliency benefits:

- Fewer and smaller messages sent over the network results in a low and consistent latency profile – the data gets to consumers faster, especially on unreliable networks such as mobile or satellite.
- Less bandwidth usage means that sending data costs you & your customers less.
- With low network overhead per connection, Diffusion can support vast numbers of concurrent clients.
- Topic updates are queued on a per-client basis, adapting to congested networks and allowing full recovery in the event of disconnections
- Newly connected subscribers automatically receive the latest values for subscribed topics, reducing system complexity.

The Pub/Sub model is a natural fit for Micro Service architectures because it explicitly decouples producers & consumers of data, making it ideal for both broadcasting and ingesting data to & from thousands of applications or devices.

#### *Point-to-Point Messaging*

Diffusion's Point-to-Point messaging provides advanced RPC-like functionality with strict delivery guarantees:

- Send messages directly to one or many connections, with delivery acknowledgements and configurable timeouts – providing reliable and predictable messaging behavior.

- Diffusion can automatically load-balance messages across multiple remote recipients, supporting highly-available back-end systems
- Diffusion automatically correlates responses to the original message sender, simplifying code complexity and making it quicker to develop complex application interactions.

#### *Client Connectivity*

In order to account for volatile networks, Diffusion provides a reliable reconnection mechanism. Inbound & outbound messages for each client are queued, even when that client is disconnected. Clients will automatically attempt to reconnect if their connection to the server is lost. Upon reconnection, any missed messages will be re-transmitted. Diffusion also provides session failover, ensuring that even when a server fails client connections are reliably moved to another server without interruption in service.

#### *Flexible Deployment*

In order to increase time-to-market, Diffusion is available as an on-premise installation, a SaaS offering, or a hybrid configuration. Most MQ products are provided as-is, with any SaaS options being provided via 3<sup>rd</sup> Parties. Diffusion delivers consistent product functionality across all environments, with 1<sup>st</sup> class support, allowing customers to choose the options that match their business, infrastructure, and cost requirements.



**Rethink Real-Time™**

**SILICON VALLEY**

+1 408 780 0720

**LONDON**

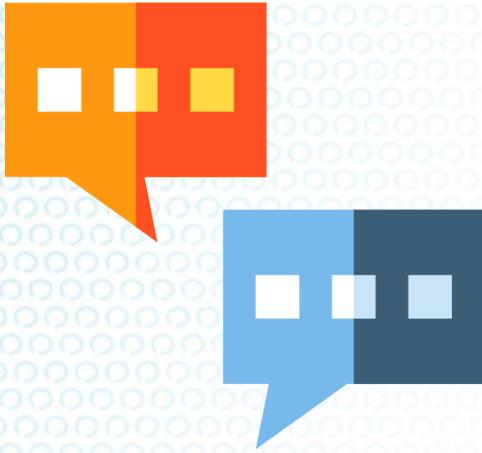
+44(0) 2035 880900

**BELFAST**

+44(0) 2035 880900

### *New Development Projects*

For new development projects, Diffusion can handle all of your application data streaming and messaging requirements



### *Existing Development Projects*

For existing applications, Diffusion extends any in-place messaging products by providing:

- Improved and consistent performance
- Multiple MQ adaptors and integration points
- Increased reliability and resiliency on congested or unreliable networks
- Lower and efficient bandwidth usage,
- Fast, flexible, demand-adaptable scalability without added infrastructure,
- Expanded operations on-premise, in-the-cloud, or in hybrid environments.

*The Diffusion Intelligent Data Platform combines Pub-Sub Streaming and Point-to-Point Messaging to handle your application data requirements; and it extends the reliability, scalability, and functionality of in-place messaging products.*