DIFFUSION DATA™

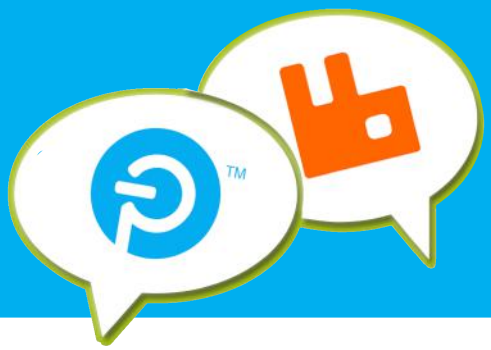
*Rethink Real Time*

# Tech Talk Note: Extend RabbitMQ over the Web

DIFFUSION®

&

RabbitMQ



*The Diffusion Intelligent Event-Data Platform™ consumes, enriches, 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.*

## RabbitMQ

RabbitMQ is an Open Source messaging broker, providing queue-based message delivery with broad support for a variety of protocols and client SDKs. RabbitMQ's design is based on standard MQ architecture, with a focus on reliability and easy integration for back-end data sources.

While this design works well for traditional development models where explicit management of message queues and routing is expected, many applications require functionality that more naturally fits higher-level concerns, such as application-level interactions, flexible data handling, and resilient connection management. In addition, with the majority of new applications being built for web, mobile and IoT devices – where data must be sent over unreliable or congested networks, potentially at large scale – the limitations of traditional MQ products become clear, at the cost of development effort and system complexity.

## Diffusion Intelligent Event-Data Platform

The Diffusion Platform is designed to extend traditional MQ products over the Internet, offering features purpose-built to meet the requirements of modern event-driven application development. As a result, Diffusion reduces application code complexity, simplifying development and speeding time-to-market. By combining RabbitMQ and Diffusion, developers receive the benefits of a well-integrated back-end data pipeline, alongside reliable, high-performance front-end data delivery.

## Pub/Sub Streaming

Both RabbitMQ and Diffusion provide Pub/Sub streaming, but what makes Diffusion unique is that each topic (e.g. the Diffusion term for logical channels through which messages are delivered and logical link between publishers and subscribers) retains its last value. This means that whenever an application subscribes to a topic it will immediately receive the latest state without having to explicitly request and wait for a new update from back-end producers. This allows systems to use Diffusion as both a delivery mechanism as well as a message cache; reducing back-end load and substantially simplifying application design.

**Binary Delta Streaming**

By caching topic values, Diffusion optimizes Pub/Sub data streaming for both delivery speed and network 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, thereby 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 up to 90%.

Diffusion's delta streaming 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.

**Point-to-Point Messaging & Client Control**

For complex systems, some form of dynamic interaction between back-end & front-end applications is often required. While RabbitMQ only provides channels with which to send arbitrary data (with some support for RPC functionality), Diffusion offers a much richer interaction model.

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

- Messages can be sent directly to one or many connections, with delivery acknowledgements and configurable timeouts – providing reliable and predictable messaging behavior.
- Diffusion can automatically load-balance inbound messages across multiple recipients, supporting highly-available back-end systems.
- Diffusion automatically correlates responses to the original message sender, simplifying code complexity and speeding development of complex application interactions.

Clients with the correct authorization can also manager the behavior of other connections, such as subscribing them to topics; throttling other connections; and even closing other connections. This level of control allows developers to safely control the behaviour of front-end applications from a controlled back-end system.



**SILICON VALLEY**
+1 408 780 0720

**LONDON**
+44(0) 2035 880900

**BELFAST**
+44(0) 2035 880900

**Automatic Reliable Reconnect**

Upon connection failure, connections to RabbitMQ will be closed. In order to reconnect, a new connection must be opened and any prior subscriptions explicitly re-established. By comparison, Diffusion automatically handles connection failures for both the server and the client. When a client loses its connection, it retains a buffer of any in-flight messages sent to the server and will automatically attempt to reconnect. Similarly, the server buffers any in-flight messages at the point of disconnection, retains the disconnected client's subscriptions, and queues any subsequent outbound messages. Upon successful recovery both the client and server re-transmit any missed messages before resuming normal operations. Users of the client SDKs are notified of these connection state changes and can respond appropriately (e.g. providing a UI notification), without worrying about the specifics of re-establishing connectivity.

**Time Series**

For systems that collect any form of time-indexed data, the ability to store and query historic events is a key requirement. RabbitMQ has no support for time series data, forcing developers to spend time integrating with additional products. Diffusion provides out-of-the-box support for time series data, allowing topics to be treated as append-only logs. This allows topics to retain the full benefits of real-time pub/sub, while providing a query API in order to select ranges of historic events. To accommodate common use-cases such as Chat applications, multiple clients can concurrently update the same time series topic as well as being able to non-destructively edit historic events.

**Combining Diffusion and RabbitMQ**

These are a few of the many features provided by Diffusion that radically simplify the complexities of building internet-based applications. By integrating Diffusion with RabbitMQ users can:

- Easily and quickly extend existing messaging platforms across the web, without incurring the costs of solving the challenges of delivering at scale.
- Use Diffusion to act as the edge-broker for dispersing data to large numbers of end-user applications & devices, or for efficiently and reliably ingesting data from disparate IoT devices - without any changes required in your pre-existing RabbitMQ architecture.
- Increase reliability and resiliency of web & mobile applications, especially when coping with congested or unreliable networks
- Expand and scale operations on-premise, in-the-cloud, or in hybrid environments, as business demands, with minimal investment in new infrastructure.

DIFFUSION DATA™
Rethink Real Time

SILICON VALLEY
+1 844 449-DATA (3282)

LONDON
+44(0) 2035 880900

BELFAST
+44(0) 2035 880900