Why add Solace PubSub+ event broker technology to the integration mix?

Integration Platform as a Service (iPaaS) offerings such as Dell Boomi, MuleSoft and Jitterbit are getting a lot of traction in the integration space, driven by the increasing popularity of cloud-native deployments and desire for citizen integration. iPaaS platforms are being used as an evolution from ESBs to orchestrate integrations between various application platforms, and are even including legacy applications.

This paper explains how the full potential of iPaaS can only be achieved through the implementation of an event-driven architecture, and why Solace PubSub+ is the best advanced event broker for doing so.
# TABLE OF CONTENTS

Executive Summary ................................................................. 1  
  About Solace ................................................................. 2  

Conceptual Overview of iPaaS .................................................. 2  
  A Predominantly REST-oriented iPaaS ........................................ 3  
  Challenges with Point-to-Point Integration ..................................... 4  

Event Enabling an iPaaS ............................................................ 5  
  Loose Coupling ............................................................... 5  
  Open Standards and Event Routing ............................................. 6  
  Event Cataloging and Governance ............................................ 7  
  Guaranteed Delivery and Burst-Handling ...................................... 8  
  Event Sourcing, Message Replay, and Streaming ............................ 9  
  Enabling AI and Machine Learning .......................................... 9  
  Hybrid Cloud ................................................................. 10  
  Enabling Event-Driven IoT .................................................... 11  

Adoption Strategy ................................................................. 11  

Integrating Solace with Legacy Assets and iPaaS .......................... 12  
  Connecting Legacy Systems with PubSub+ On-Prem ...................... 12  
  Connecting Legacy Systems with PubSub+ in the Cloud .................. 12  

Conclusion ................................................................. 13
EXECUTIVE SUMMARY

iPaaS platforms are a cloud-native managed service replacement for ESBs; a new way of integrating applications, information sources and connected devices across public cloud, private cloud and on-premises environments.

The shift from service-oriented architecture (SOA) to event-driven architecture (EDA), as well as the microservices approach to application development, represent the two other major developments in the application infrastructure space.

As these architecture patterns evolve, and distributed systems consist of more cloud-native applications and ecosystems, integration needs to be reconceived.

INTEGRATION COVERS FOUR AREAS OF FUNCTIONALITY:

1. Connecting disparate systems to each other or an intermediary so they can send and receive information;
2. Orchestrating interactions between systems;
3. Transforming messages so systems can send and receive them in a way they can understand; and,
4. Distributing/routing information between systems.

When it comes to the distribution of events and information, some ESB/iPaaS implementations rely on point-to-point synchronous communications. While this can work at a small scale with very few applications and low performance requirements within a datacenter, such an approach does not offer the scalability and flexibility required for serious enterprise IT projects.

An event-driven approach—where the heavy lifting of event routing is delegated to an event broker—allows for a publish/subscribe backbone for iPaaS. This lets you use REST as a protocol for publish/subscribe, guaranteed delivery and smart routing. An event-driven architecture provides an agile and robust backbone for an iPaaS.
ABOUT SOLACE
Solace is a Canadian company based in Kanata, Ontario, near Ottawa. We make an advanced event broker called PubSub+ that offers unparalleled performance, reliability and robustness. Each day we solve the toughest data movement challenges in multiple industries and use cases. Our customers include most of the world’s top investment banks, telcos, the world’s largest manufacturer of commercial vehicles, and government agencies such as the FAA, NOAA, and Singapore’s Land Transport Authority.

CONCEPTUAL OVERVIEW OF IPAAS
An integration platform provides adapters for connectivity, performance payload transformation as needed, orchestrates sequences of interactions, and manages the distribution of data between systems. Solace addresses the data distribution part of the integration problem. Furthermore, with an event-driven integration platform, which supports sophisticated event routing, the need for orchestration is minimized and a self-orchestrated microservices-oriented approach to integration can be adopted. This tends to be leaner, more agile and more responsive. In other words, an event-driven approach to integration leads to more robustness, faster time to market, lower costs and deployment flexibility.
A PREDOMINANTLY REST-ORIENTED IPAAS

Many enterprises adopt a very simplistic approach to integration. They deploy an ESB, iPaaS or API Gateway which has adapters to connect or proxy services from legacy and modern systems. These adapters could be file or database or Oracle or SAP ERP. They could be cloud or SaaS adapters, such as Salesforce or AWS S3. With connectivity established, sometimes payload transformations are required.

REST is a common and simple way to expose integration flow as APIs. This works at small scale, with a few systems and at a small throughput, usually within a datacenter. Such an approach has challenges when more systems are added or the volume of information being shared increases.

As more upstream and downstream systems get added, a point-to-point integration mechanism becomes exponentially more complex. This is especially true if the same messages/APIs must be sent to multiple systems.
# Challenges with Point-to-Point Integration

## Lack of Agility
For every integration point, the producer or the consumer applications need to be connected one more time. There is tight coupling between various systems, so if there is any bug, change, or outage, the iPaaS, the producers and other consumers are impacted.

## Lossy with Retries
When fully-guaranteed message delivery is required, a simple RESTful iPaaS approach puts the burden of message redelivery on the producer application, which must make sure all downstream consumers receive the information or store it until delivery is made. This compounds the coupling problem and often leads to data loss.

## Tight Coupling
Every time a new integration point is added, the iPaaS flows must be rewired to include it. All performance characteristics of the new integration point are inherited by the existing flow, often making it slower, less robust and more complex. This high impact integration slows time to market, reduces agility and increases fragility.

## Burst-Handling
Burst-handling is an equally big challenge because if a downstream application is slow, the iPaaS will have a load build up that results in a slow down or crash. This introduces the need for rate limiting and load balancing.

## Distributed Deployment and Coupling
Most companies migrate systems to the cloud gradually, versus moving everything at once. For gradual migration, a REST/API-based approach is problematic because it burdens producers with the responsibility of handling message delivery—and asking them to do so with WAN/Internet links in the mix is too much to ask.

## Loss of AI/Streaming Readiness
A point-to-point approach makes it difficult to include AI and machine learning services into business processes. A streaming approach lets you “wire tap” a feed to facilitate the efficient integration of AI, ML and other insights systems.

Simply put, a point-to-point REST or API-based approach to integration does not meet the needs of enterprises with sophisticated, high-volume or highly-distributed use cases.
EVENT-ENABLING AN IPAAS

An event-driven architecture is much better as a foundational layer for an iPaaS and, fortunately, most iPaaS platforms support integration with advanced event brokers like Solace PubSub+.

At the heart of an event-driven architecture is an event broker, ideally an advanced event broker that supports:

- Exchange patterns including publish/subscribe, request/reply, queuing, streaming, fan-out, fan-in;
- Guaranteed and best-effort delivery;
- Open standards such as AMQP, MQTT, REST, JMS and WebSocket;
- Asynchronous API and communication support;
- Sophisticated event routing including wildcards for messaging topics and REST URLs;
- Isolation and graceful recovery of slow and temporarily disconnected consumers; and,
- Hybrid and multi-cloud deployment with guaranteed, WAN-optimized delivery between clouds.

LOOSE COUPLING

An event broker is a core tool in an event-driven architecture. It sits in the heart of an iPaaS and decouples consumers. ESB/iPaaS platforms are only needed when systems don’t follow open standards or a canonical data model, or where transformation is required. For systems that can speak open standards (such as REST, AMQP, JMS, and MQTT), the event broker has the ability to natively talk to them.

With queuing capabilities, PubSub+ acts as a shock absorber and decouples the producers from the consumers such that
the performance of the producer is not impacted by that of the consumers. It even allows consumers who have lost data to replay it back.

With publish/subscribe and streaming capabilities, PubSub+ decouples producers and consumers from each other and the iPaaS itself to enable the easy addition of multiple consumers in the future.

New consumers can simply “subscribe” to the published topics or queues without requiring any change to the producer or the iPaaS flow.

**OPEN STANDARDS AND EVENT ROUTING**

Support for open APIs and protocols like AMQP, JMS, MQTT and REST means producers and consumers are decoupled not only from each other but also from the event broker and its specific APIs. PubSub+ does wireline translation between these protocols, so you could publish an MQTT message from a device and consume it as an AMQP or JMS message.

The same works for a request/reply pattern. Because PubSub+ does protocol translation at a wireline level, no bridges or intermediaries are required. This results in a simpler, better performing system. Topics are translated to URLs (and vice versa) dynamically. Wildcard routing is also allowed on URLs, similar to topics.

Topics and URLs don’t have to be predefined and administratively configured in the routers. Subscribers can simply connect and start subscribing to any topic.
string, such as a/b/c or a/b/ (anything that has a and b as the first 2 segments), or */b/c (will match a/b/c, b/b/c, c/b/c, etc.). Subscriptions are dynamically ingested into the Solace messaging routers. For MQTT, the wildcards will automatically translate to # and + as per the standard.

Similarly, publishers can publish on any topic or any URL, such as a/b/x, a/b/c or http://host:port/a/b/c/d. This dynamic nature of topic routing allows for great operational simplicity.

**EVENT CATALOGING AND GOVERNANCE**

Governance is critical for integration at scale. PubSub+ has been built for performance, especially fan-out. This allows visibility via event cataloging. Custom dashboards can “eavesdrop” on events and create both statistical and data flow views of how messages are flowing (and how those flows are related). This visualization leads to better governance and more effective analysis.

For design-time governance, each microservice has topics and URLs that it publishes and subscribes to, and the schema of the payload. The event catalog can be created and compared with the generated event catalog as explained above.

Runtime governance is exercised by ensuring that the right events are published by and subscribed to by only authorized microservices or integration flows. You can even throttle and tune the delivery of events on a per queue/stream level.
You can also restrict a system’s ability to send or receive information through the use of sophisticated access control lists (ACLs) that support wildcards, whitelists and blacklists. For example, a device could publish on a topic a/b/c if it is allowed to publish on a/> or a/*/c.

Schema validation is also plug-and-play, and not limited to any payload protocol, such as Avro, Swagger or Google Protobufs. This is done via schema validation microservices, which can consume raw high-level topics like a/b/c/raw and, if the schema is valid, forward the message to a validated topic, such as a/b/c/val.

That means governance is handled both at design-time and runtime. Topic hierarchy conventions and ACLs should be agreed upon at design-time, covering things such as version number and metadata. This becomes the event-driven microservices catalog of endpoint and conventions, which applications use to publish and subscribe to events of interest. Runtime governance is carried out by throttling, security enforcement and monitoring, all of which PubSub+ provides out of the box.

GUARANTEED DELIVERY AND BURST-HANDLING

An iPaaS without a message bus/event broker tightly couples producers with consumers. Consider, for example, a publisher that needs to send a message to multiple back-end systems, such as an order management app, data warehouse, and inventory management system. If one or more of these systems are down, should the order be accepted? Can the order still be processed if only the order management app and inventory management system received the message—since the data warehouse can be updated a few minutes later?

There are two ways data can flow as part of such interactions. The first is serial, or synchronous, where a requestor waits for a reply. The other is parallel, or asynchronous, referring to the delivery of an update without expectation of a response.

There is immense benefit in having PubSub+ as an event broker providing guaranteed delivery, decoupling producers from consumers. PubSub+ takes the messages from the producer and stores them, acknowledging the publisher. PubSub+ then delivers the events to all consuming applications in a “store and forward” manner. Once all the consumers acknowledge that they have received the event, PubSub+ deletes the event from its spool, decoupling producers from consumers.

If the producer suddenly publishes a large number of events (more than a consumer can receive and process), PubSub+ acts as a shock absorber, buffering the burst by queuing all messages so they can be delivered at whatever rate each subscribing application can accept.
**EVENT SOURCING, MESSAGE REPLAY, AND STREAMING**

PubSub+ supports message replay so you can play back messages days, weeks or even months after they were initially sent and received. This lets you easily recover from misconfiguration issues, bugs and corrupt databases. This also lets you more smoothly add new applications to your system by testing them with real-world data in non-production environments.

Message replay can be initiated either by the subscribing application using Solace client libraries or by the administrator. It can be performed for any consumer, no matter what protocol or API they are using (including JMS1.1 and 2.0, AMQP, MQTT, REST/WebHook), and the messages are replayed in their original order, across all topics, with support for wildcard subscriptions on queues and topic endpoints. Messages can be replayed from a specific time, or from the beginning of the replay log.

**ENABLING AI AND MACHINE LEARNING**

Without an event-driven architecture, integrating data lakes, machine learning algorithms and AI systems means you need to add another integration point either for the data producers or for the databases (systems of record) via polling. PubSub+ can add such systems as recipients of information without impacting data producers or systems of record.

Many modern insights systems support streaming ingestion via technologies like Spark, Flink, Storm and Beam. PubSub+ works with all of these technologies via open standards and guarantees data delivery even if they have outages, and it can do so without the rest of the ecosystem even taking notice.
The last mile of an event-driven architecture is cloud-native integration. PubSub+ can run natively in all major cloud environments or as a fully managed service. Solace provides AMIs for AWS, and PubSub+ is easily deployed on Azure and Google within a Docker container or with Kubernetes. PubSub+ also natively gets deployed in PaaS platforms such as Pivotal Cloud Foundry and OpenShift.

WAN-optimized, encrypted and lossless bridges enable intelligent and dynamic routing among different instances of PubSub+, creating an “event mesh” so data can move between on-premises and cloud deployments. Producers and consumers do not need to worry about where the rest of the application is deployed; PubSub+ delivers data to all appropriate applications.

An iPaaS can provide adapters and transformation as an overlay to PubSub+, which works as the core, lossless data movement backbone. Any network blips or any downstream system issue (whether on-prem or in the cloud) will not impact producers, consumers or even the iPaaS if PubSub+ is doing the heavy lifting of data movement.
ENABLING EVENT-DRIVEN IOT

A Solace-based event-driven architecture can incorporate IoT as well as applications, allowing them to leverage the same backbone rather than needing to bring in new tools, new complexities, and new training.

PubSub+ natively supports MQTT at very high scale, and translates between MQTT and back-end messaging protocols such as AMQP, JMS and REST at the wireline level.

ADOPTION STRATEGY

PubSub+ works with all standards-based iPaaS and ESB platforms because it supports standard APIs and protocols like AMQP, JMS, MQTT, REST and WebSocket. So as you go down the road of integrating new applications using iPaaS and an event-driven architecture you can implement PubSub+ as an event-distribution layer underneath your legacy app servers, EAI software and ESBs. Once this is done, PubSub+ underpins iPaaS stacks both on-premises and in the cloud so new applications are easily integrated with legacy systems. This lets new applications interoperate seamlessly with legacy systems so you can migrate workloads in a step-by-step, flow-by-flow manner instead of trying to execute an impractical “lift and shift.”
As data moves through the system, AI, ML, and big data systems become passive listeners and simply consume events. PubSub+ also has native REST capabilities, so it can act as a gateway for API/microservice access.

Lastly, PubSub+ powers IoT connectivity at scale with support for millions of concurrently connected, streaming devices. Adoption of Solace, therefore, is on the back of a current in-flight project, where existing integration requirements can leverage event brokering. Further to that, applications move on project-by-project, opportunistically, so that testing efforts can be optimized. Over time, Solace grows into the core event broker, integrating microservices on-prem and in the cloud, while filling AI and ML lakes on the side.

INTEGRATING SOLACE WITH LEGACY ASSETS

PubSub+ works with leading iPaaS platforms through partnerships with Dell Boomi, MuleSoft and Adaptris, and it offers plug-and-play interoperability with many other iPaaS, API and ESB offerings.

CONNECTING LEGACY SYSTEMS WITH PUBSUB+ ON-PREM

You may want to deploy Solace PubSub+ appliances or software within your datacenter, and use them to meet the messaging needs of select applications while leaving others untouched. Some data entering the system via PubSub+ may be needed by applications using different messaging or data movement technologies. For example, certain records might need to be stored in a database. In such scenarios, you could set up your preferred iPaaS to consume messages from the PubSub+ message broker, then use the iPaaS JDBC or SQL component to insert the messages into a database via SQL query.

CONNECTING LEGACY SYSTEMS WITH PUBSUB+ IN THE CLOUD

If you’re deploying PubSub+ message brokers in the cloud or setting up instances of PubSub+ Cloud to meet the messaging needs of a new cloud-based service, you’ll frequently still need to exchange information with legacy applications running on-premises. In this scenario, you can configure the iPaaS to consume messages from the legacy system and publish them to your Solace PubSub+ Cloud message broker, which would then publish messages to all intended recipients via the new queue/topic.
CONCLUSION

There are many advantages to iPaaS, but limiting yourself to point-to-point REST and API-based connectivity limits the potential value to your organization in terms of agility, performance and robustness. Pairing your iPaaS with an advanced event broker to implement an event-driven architecture eliminates these limitations and unlocks high-value digital transformation by linking applications, IoT devices and microservices across hybrid and multi-cloud environments. Solace PubSub+ is the best event broker for achieving that goal, and it works with leading iPaaS vendors to create a best-of-breed, scalable, robust, agile and secure architecture. We have the know-how and experience to guarantee success, and we look forward to working with your team.
ABOUT SOLACE

We are the creators of PubSub+, an advanced event broker that can be used to create an event mesh. As the only unified event broker that supports publish/subscribe, queuing, request-reply and streaming using open protocols and APIs across hybrid cloud and IoT environments, PubSub+ rapidly and reliably routes information between applications, devices and people across clouds. Established enterprises such as SAP, Barclays, and the Royal Bank of Canada as well as high-growth companies such as VoiceBase and Jio use our smart data movement technologies to modernize legacy applications and successfully pursue analytics, hybrid cloud and IoT strategies. Learn more at solace.com.

A FEW OF OUR CUSTOMERS

![Airtel](image1.png) ![Standard Chartered](image2.png) ![Barclays](image3.png) ![Bet365](image4.png)

![NASA](image5.png) ![SAP](image6.png)

OUR FEATURED PARTNERS

![Dell](image7.png) ![Boomi](image8.png) ![Pivotal](image9.png) ![SAP](image10.png) ![F5](image11.png)