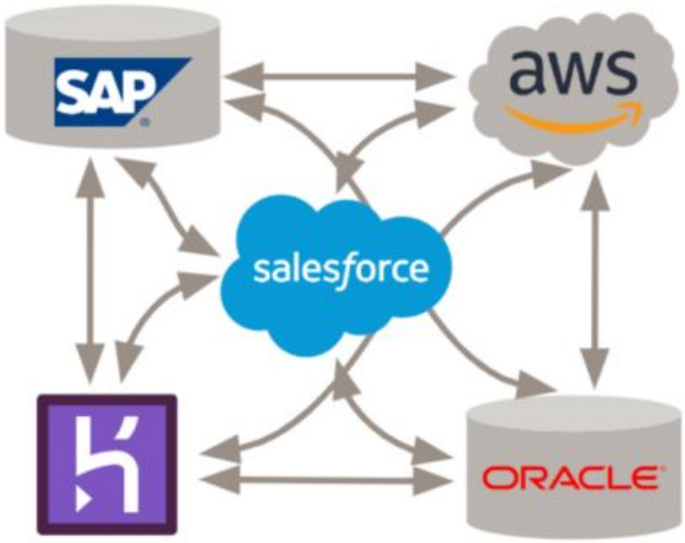
Platform Events

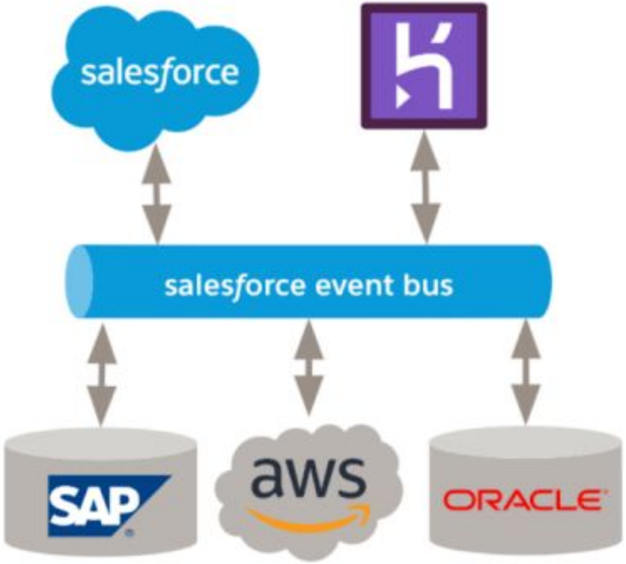
Svet Voloshin

What scales better?

Point to Point Architectures



Event Driven Architectures



Event-Driven Architecture

- Asynchronous by nature: *Remote Process Invocation "Fire and Forget"*
- Moving closer to ***real-time*** because of innovation and **faster bi-directional** communication
- *Unlike request-response communication models, software architecture built on an event-driven model decouples event producers from event consumers, thereby simplifying the communication model in connected systems.*
- Use Cases:
  - Order Fulfillment in a Vendor App
  - Process Merchandise Returns in Salesforce
  - Reassign Lead Records

Platform Events on Platform?

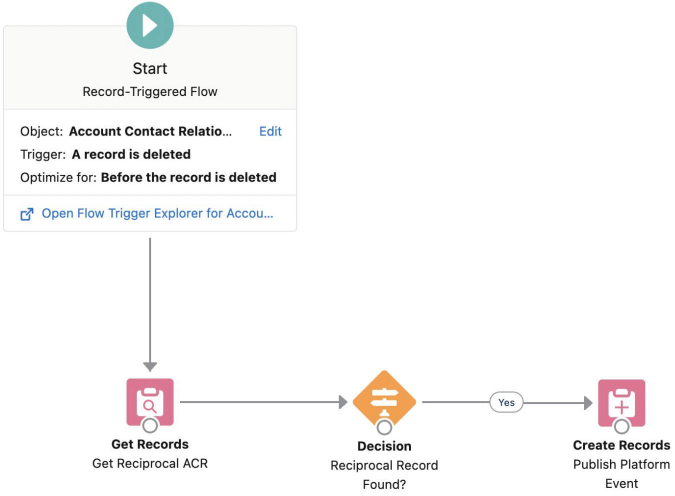
Platform Event—Triggered Flow

Launches when a platform event message is received. This autolaunched flow runs in the background.

Yes, it's possible to use Platform Events <u>within</u> Salesforce

- Use Case: create and delete reciprocal Account-Contact Relationships for Person Accounts
- Why? Improved User Experience and cleaner data.
  - It's not possible to connect one Person Account to another directly, but it can be done *indirectly* via Account-Contact Relationships
  - When one person is related to another in Salesforce, the other person's record should reflect the same connection - great, do it via Flow!
  - What about record deletion? Again, Flow! ...not so *fast*. You will run into circular triggering error situation.
  - Solution: decouple the delete operations via Platform Event - Triggered Flow

Decoupling the Flows



Start

Record-Triggered Flow

Object: **Account Contact Relatio...** Edit

Trigger: **A record is deleted**

Optimize for: **Before the record is deleted**

Open Flow Trigger Explorer for Accou...

Get Records

Get Reciprocal ACR

Decision

Reciprocal Record Found?

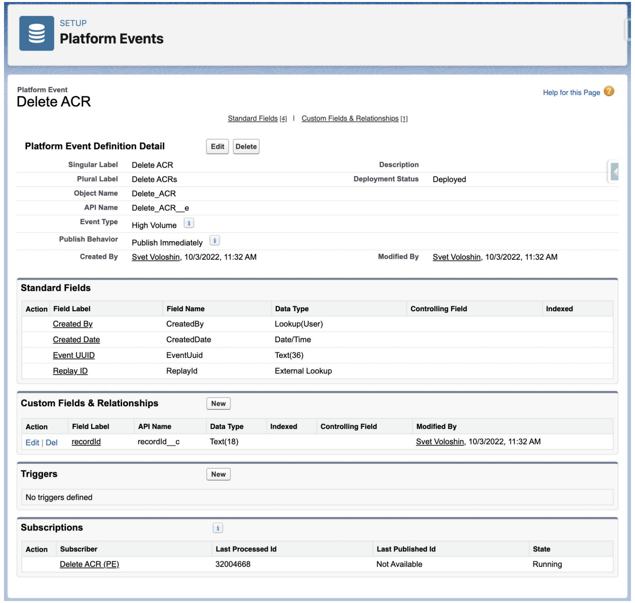
Yes

Create Records

Publish Platform Event

Publishing Flow

(Record-Triggered)



SETUP

Platform Events

Platform Event

Delete ACR

Help for this Page

Standard Fields [4] | Custom Fields & Relationships [1]

Platform Event Definition Detail Edit Delete

Singular Label	Delete ACR	Description	
Plural Label	Delete ACRs	Deployment Status	Deployed
Object Name	Delete_ACR		
API Name	Delete_ACR__e		
Event Type	High Volume		
Publish Behavior	Publish Immediately		
Created By	Svet Voloshin, 10/3/2022, 11:32 AM	Modified By	Svet Voloshin, 10/3/2022, 11:32 AM

Standard Fields

Action	Field Label	Field Name	Data Type	Controlling Field	Indexed
	Created By	CreatedBy	Lookup(User)		
	Created Date	CreatedDate	Date/Time		
	Event UUID	EventUuid	Text(36)		
	Replay ID	ReplayId	External Lookup		

Custom Fields & Relationships New

Action	Field Label	API Name	Data Type	Indexed	Controlling Field	Modified By
Edit | Del	recordId	recordId__c	Text(18)			Svet Voloshin, 10/3/2022, 11:32 AM

Triggers New

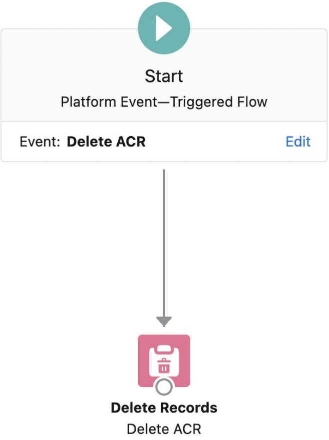
No triggers defined

Subscriptions

Action	Subscriber	Last Processed Id	Last Published Id	State
	Delete ACR (PE)	32004668	Not Available	Running

Platform Event

(recordId)



Start

Platform Event—Triggered Flow

Event: **Delete ACR** Edit

Delete Records

Delete ACR

Subscribing Flow

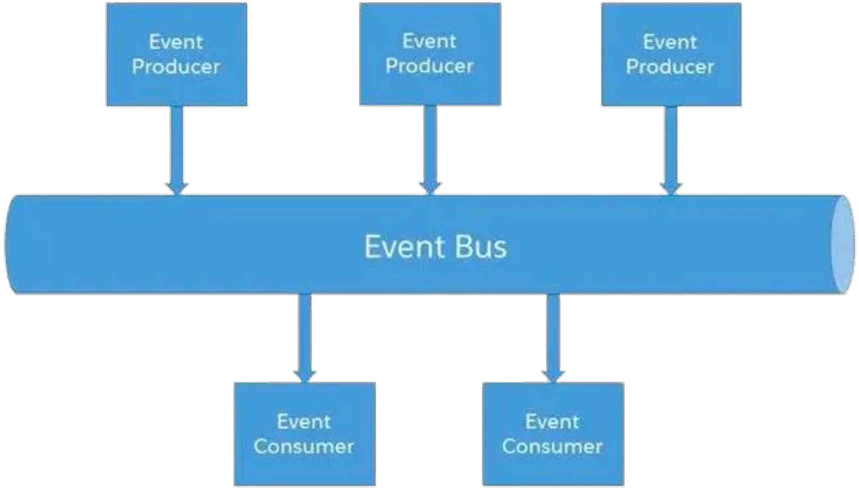
(PE-Triggered)

Pub/Sub Methods

- Publishing Events:
  - Flow
  - Process Builder (legacy)
  - APEX Trigger
- Subscribing to Events
  - CometD client (legacy): Subscribe to Platform Event Notifications with CometD (Documentation)
  - EMP Connector - free, open-source, community-supported tool. Salesforce provides this tool as an example of how to subscribe to events using CometD.
  - Pub/Sub API - latest and the greatest

Components of Event-Driven Systems

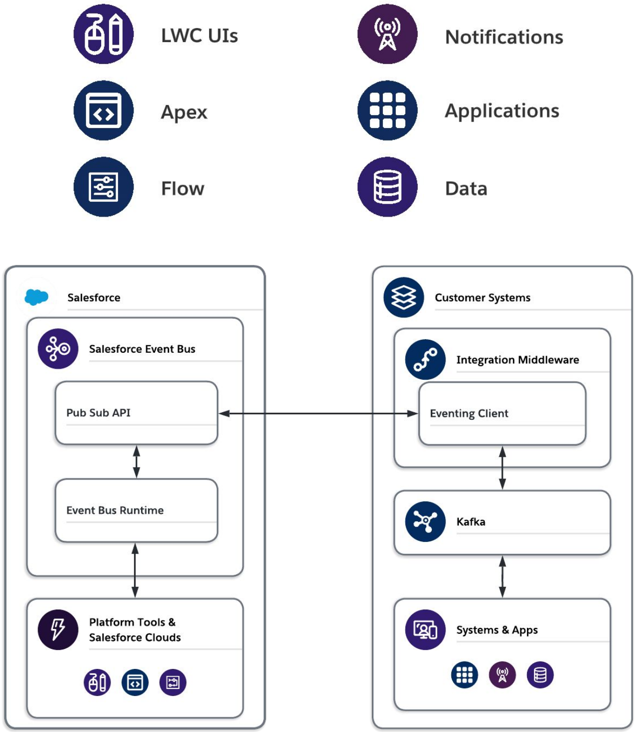
- Event
- Event Message
- Event Producer
- Event Channel
- Event Consumer
- Event Bus



Pub Sub Patterns

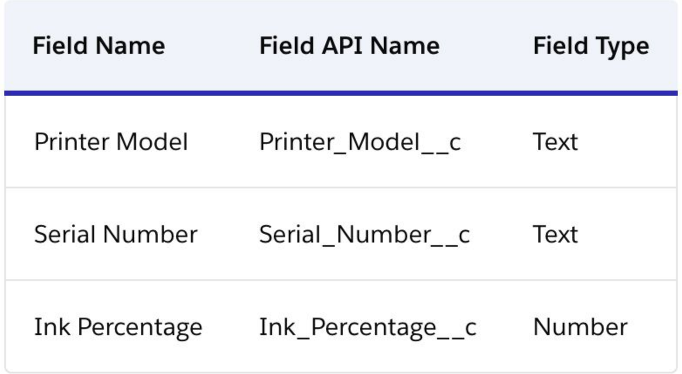
- Use Salesforce APIs to connect the Salesforce Event Bus to an external messaging system, like Apache Kafka, RabbitMQ, or Amazon EventBridge.

- Those messaging systems handle event orchestration and connect external microservices and apps that need to respond to events.



Custom Platform Events

- Publish and process custom notifications.
    - Send order information to an order fulfillment service
    - Send printer ink information that is processed by a service app.
- You define a custom platform event in Salesforce in the same way that you define a custom object.



Field Name	Field API Name	Field Type
Printer Model	Printer_Model__c	Text
Serial Number	Serial_Number__c	Text
Ink Percentage	Ink_Percentage__c	Number

- Create a platform event definition by giving it a name and adding custom fields.
- Platform events support a subset of field types in Salesforce.

Platform Event Fields

Standard Fields

- Label
- Plural Label
- Starts with a vowel sound
- Object Name
- Description
- Deployment Status

Custom Fields

- Checkbox
- Date
- Date/Time
- Number
- Text
- Text Area (Long)

ReplayId System Field - (opaque ID): refers to the position of the event in the event stream. Replay ID values are not guaranteed to be contiguous for consecutive events. *Example: the event following the event with ID 999 can have an ID of 1,025.*

EventUuid System Field - universally unique identifier (**UUID**) that identifies a platform event message.

API Name Suffix - “__e”, instead of the custom “__c”. *Example: New_Order__e*

Publish Behavior

Publish After Commit - published only after a transaction commits successfully.

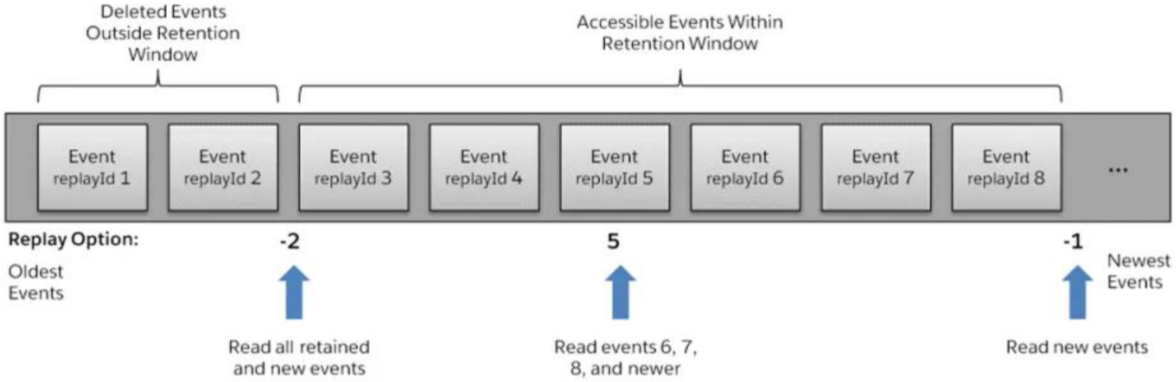
For example, a process publishes an event message and creates a *task record*. A second process that is subscribed to the event is fired and *expects to find the task record*.

Publish Immediately - published when the publish call executes.

For example, the immediate publishing behavior is suitable for an event used for *logging purposes*. With this option, a subscriber might receive the event message **before** data is committed by the publisher transaction.

Replaying Events

- A subscriber can store a replay ID value and use it on resubscription to retrieve events that are within the retention window.
- For example, a subscriber can retrieve missed events after a connection failure.



Standard Platform Events (legacy)

Events with predefined fields

- **AssetTokenEvent** - OAuth 2.0 authentication activity
- **BatchApexErrorEvent** - reports errors encountered in batch Apex jobs
- Salesforce publishes standard platform events in response to an action that occurred in the app or errors in batch Apex jobs. You can subscribe to a standard platform event stream using the subscription mechanism that the event supports.

High-Volume Platform Events

Use high-volume platform events to publish and process millions of events efficiently and to scale your event-based apps.

- Available with API v45.0 and later - high-volume PEs by default.
- Standard-volume events are still supported but not available for new event definitions.
- High-volume platform events offer better scalability than standard-volume platform events.
    - **Asynchronous Publishing:** After the publishing call returns with a successful result, the publish request is queued in Salesforce. **The event message isn't always published immediately.**
    - Separate Event Allocations: See Platform Event Allocations

High-Volume Platform Event Default Allocations

Description	**Performance and Unlimited Editions**	**Enterprise Edition and Professional Edition (with API Add-On)**	**Developer Edition**
Event Delivery: maximum number of delivered event notifications in the last 24 hours, shared by all clients. (Applies to CometD and Pub/Sub API clients, empApi Lightning components, and event relays only.)	50,000	25,000	10,000
Event Publishing: maximum number of event notifications published per hour. (Applies to all publishing methods, including Apex, Pub/Sub API and other APIs, flows, and Process Builder processes.)	250,000	250,000	50,000

The number of delivered events to clients is counted for each subscribed client, including event relays. If you have multiple client subscribers, your usage is added across all subscribers. For example, you have an **Unlimited Edition org with a default allocation of 50,000 events in a 24-hour period**. Within a few hours, **20,000** event messages are delivered to **two subscribed clients**. So you **consumed 40,00**0 events and are **still entitled to 10,000** events within the **24-hour** period.

HPE Add-On License & Usage-Based Entitlement

Description	**Performance and Unlimited Editions**	**Enterprise Edition and Professional Edition (with API Add-On)**
Event Delivery: entitlement for delivered event notifications, shared by all clients. (Applies to CometD and Pub/Sub API clients, empApi Lightning components, and event relays only.) You can exceed this entitlement by a certain amount before receiving an error. Salesforce uses the monthly entitlement for event overage monitoring. The monthly entitlement is returned in the limits REST API resource.	Last 24 hours: 150,000 (50 K included with org license + 100 K from add-on license) Monthly entitlement: 4.5 million (1.5 million included with org license + 3 million from add-on license)	Last 24 hours: 125,000 (25 K included with org license + 100 K from add-on license) Monthly entitlement: 3.75 million (0.75 million included with org license + 3 million from add-on license)
Event Publishing: maximum number of event notifications published per hour. (Applies to all publishing methods, including Apex, Pub/Sub API and other APIs, flows, and Process Builder processes.)	275,000 (250 K included with org license + 25 K from add-on license)	275,000 (250 K included with org license + 25 K from add-on license)

When you purchase an add-on license, the hourly event publishing allocation increases by **25,000 events per hour**.

The **maximum event message size** that you can publish is **1 MB**. If your event object has **hundreds of custom fields** or many long text area fields, you can hit this **limit**. In this case, the publishing call gets an error.

Streaming Monitor App

Streaming Monitor

By Salesforce Labs

Monitor streaming events (PushTopic, Generic, Platform Events, CDC...)

★★★★★ 4.71 Average Rating (21 Reviews)

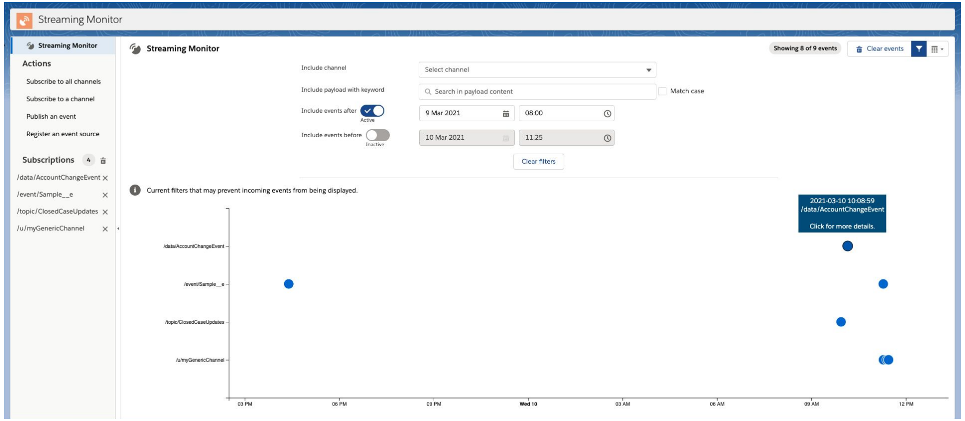
Philippe Ozil

pozil

Developer/Admin tool that lets you monitor streaming events (PushTopic, generic, standard/custom platform events, CDC and monitoring events)

Install from the AppExchange or build from GitHub sources:

https://github.com/pozil/streaming-monitor



/data/AccountChangeEvent ✕
/event/Sample__e ✕
/topic/ClosedCaseUpdates ✕
/u/myGenericChannel ✕

Include channel: Select channel

Include payload with keyword: Search in payload content □ Match case

Include events after ✓ Active: 9 Mar 2021 08:00

Include events before Inactive: 10 Mar 2021 11:25

Current filters that may prevent incoming events from being displayed.

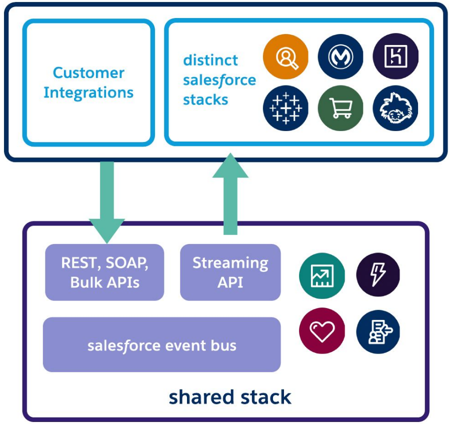
2021-03-10 10:08:59
/data/AccountChangeEvent

Click for more details.

Salesforce Event Bus - Legacy Version

- Multi-tenant eventing service built on Apache Kafka
- Very popular and very scalable technology used by customers for...
    - offline sync for mobile apps
    - integrating external chat bots with Service Cloud, etc.
- Within Salesforce:
    - Quip, Einstein Prediction Builder, and Real-Time Event Monitoring

Shared the same infrastructure as products like Sales Cloud and Service Cloud while powering pub/sub for all Salesforce clouds.

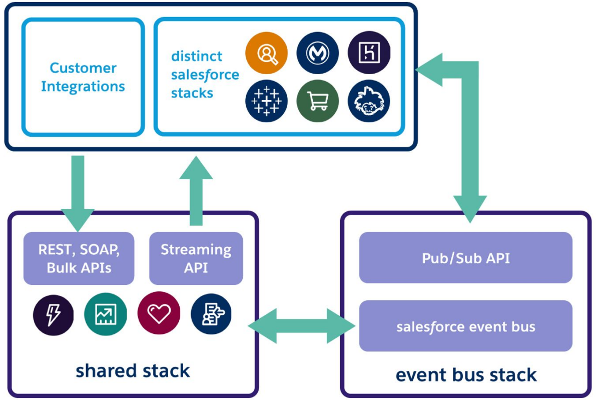


Salesforce Event Bus - Independent Run Time

Scalability: This new Event Bus Service is *internal* to Salesforce and **sits outside of the servers** powering Sales Cloud and Service Cloud.

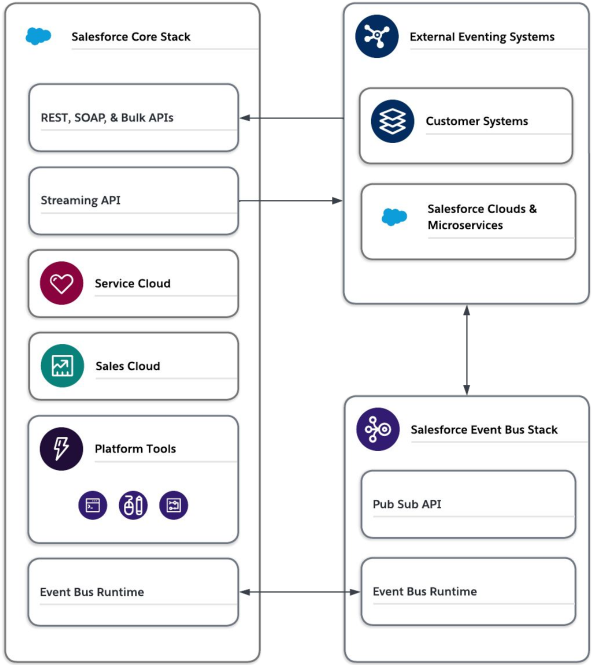
- Expansion into other clouds
- Adoption of microservice architecture for eventing
- Foundation for the rollout of Pub/Sub API

Why new? *Salesforce has grown and acquired new companies like MuleSoft and Tableau, the time has come for a change.*



Pub-Sub API

- Generally Available (GA) - June 30th, 2022
- Remote Procedure Call (RPC) framework
- Based on gRPC - API format that was originally created by **Google** for microservice architectures.
- Works with:
  - High-volume platform Events (newest version of PE)
  - Change Data Capture (technically Batch Data Sync)
  - Real-Time Event Monitoring (Shield)
- Runs on a **dedicated infrastructure** that's distinct from the REST, SOAP, Bulk, and Streaming APIs that have been used to publish and subscribe to events.



Salesforce Core Stack

REST, SOAP, & Bulk APIs

Streaming API

Service Cloud

Sales Cloud

Platform Tools

Event Bus Runtime

External Eventing Systems

Customer Systems

Salesforce Clouds & Microservices

Salesforce Event Bus Stack

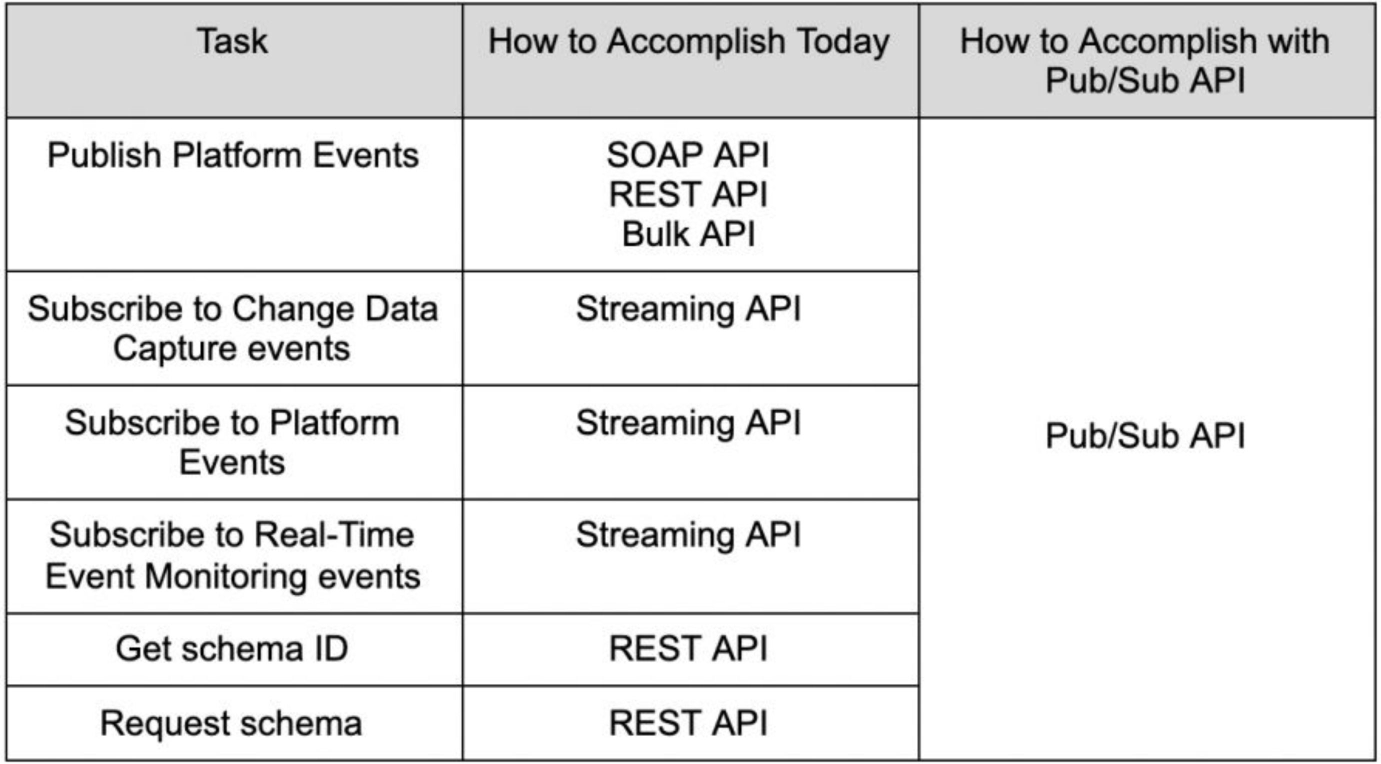
Pub Sub API

Event Bus Runtime

Pub-Sub API Innovation

- **Subscription Flow Control**: Specify how many events you want a subscriber to receive in a subscribe call.
- **Real-Time Publish Acknowledgements:** receive a **real-time response** that the event was successfully published. This improves the visibility of publish operations and enables publish retry mechanisms if needed.
- **Publish, Subscribe, and Get Schema via the Same API.** Do more with one API.

The API Difference…



Task	How to Accomplish Today	How to Accomplish with Pub/Sub API
Publish Platform Events	SOAP API REST API Bulk API	Pub/Sub API
Subscribe to Change Data Capture events	Streaming API	
Subscribe to Platform Events	Streaming API	
Subscribe to Real-Time Event Monitoring events	Streaming API	
Get schema ID	REST API	
Request schema	REST API	

Main Pub/Sub API Benefits

- **Better performance:** gRPC is about seven to ten times faster than REST due to the fact that its payload format is compressed, lightweight byte buffers instead of JSON.
- **Bidirectional streaming:** This enables both a client and a server to operate independently, meaning that a client can consume and publish events in any order it wishes. In addition, gRPC preserves the order of messages within the stream.
- **Built on HTTP 2:** This new standard guarantees better performance, increased security, and lower overhead compared to APIs built on HTTP 1.

Working with Pub/Sub API

Pub Sub API MuleSoft connector

Create an Event Fabric: Introducing the MuleSoft Connector for Salesforce’s Pub/Sub AP

Build your own Pub Sub API client

- Developer Doc
- GitHub Repo

Additional Resources