



Event Brokering Using SAP Cloud Platform Enterprise Messaging

Karsten Strothmann, SAP SE
December 5th, 2019

PUBLIC

Disclaimer

- The information in this presentation is confidential and proprietary to SAP and may not be disclosed without the permission of SAP. Except for your obligation to protect confidential information, this presentation is not subject to your license agreement or any other service or subscription agreement with SAP. SAP has no obligation to pursue any course of business outlined in this presentation or any related document, or to develop or release any functionality mentioned therein.
- This presentation, or any related document and SAP's strategy and possible future developments, products and or platforms directions and functionality are all subject to change and may be changed by SAP at any time for any reason without notice. The information in this presentation is not a commitment, promise or legal obligation to deliver any material, code or functionality. This presentation is provided without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. This presentation is for informational purposes and may not be incorporated into a contract. SAP assumes no responsibility for errors or omissions in this presentation, except if such damages were caused by SAP's intentional or gross negligence.
- All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.

Agenda

SAP Cloud Platform Enterprise Messaging

Event-Driven architectures with SAP

How to event enable SAP Backends

Selected Event-Driven Use Cases

Short Demo

Key takeaways

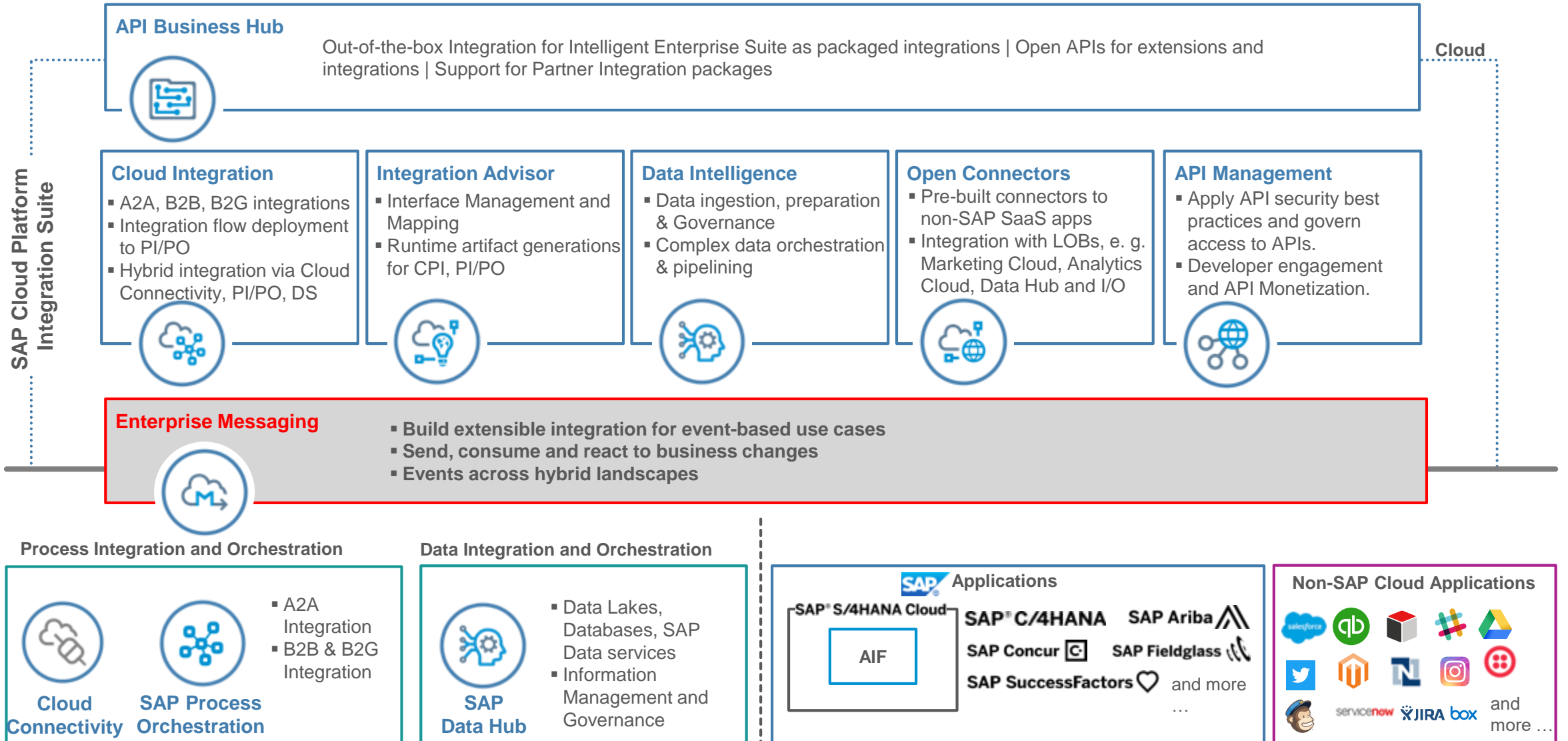


SAP Cloud Platform Enterprise Messaging



Hybrid Integration Platform

Enterprise Integration Suite for hybrid and heterogenous environments



SAP Cloud Platform Enterprise Messaging



Fully-managed cloud service for enabling applications to **communicate asynchronously** through **messages and events**.



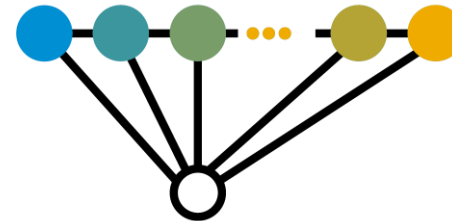
Seamless connectivity

- Connect your applications and services seamlessly using standard protocols (MQTT, AMQP 1.0, REST)
- APIs (Java JMS, Node.js)
- Exchange patterns (e.g. pub/sub, point-to-point, request/reply)



Reliable communication

- Messaging middleware decouples communication between senders and receivers for greater scalability.
- Choose from different Quality of Service levels to ensure messages are delivered reliably.



Non-blocking communication

- Asynchronous message patterns provide elasticity meaning unexpected peaks in data traffic can be handled efficiently.



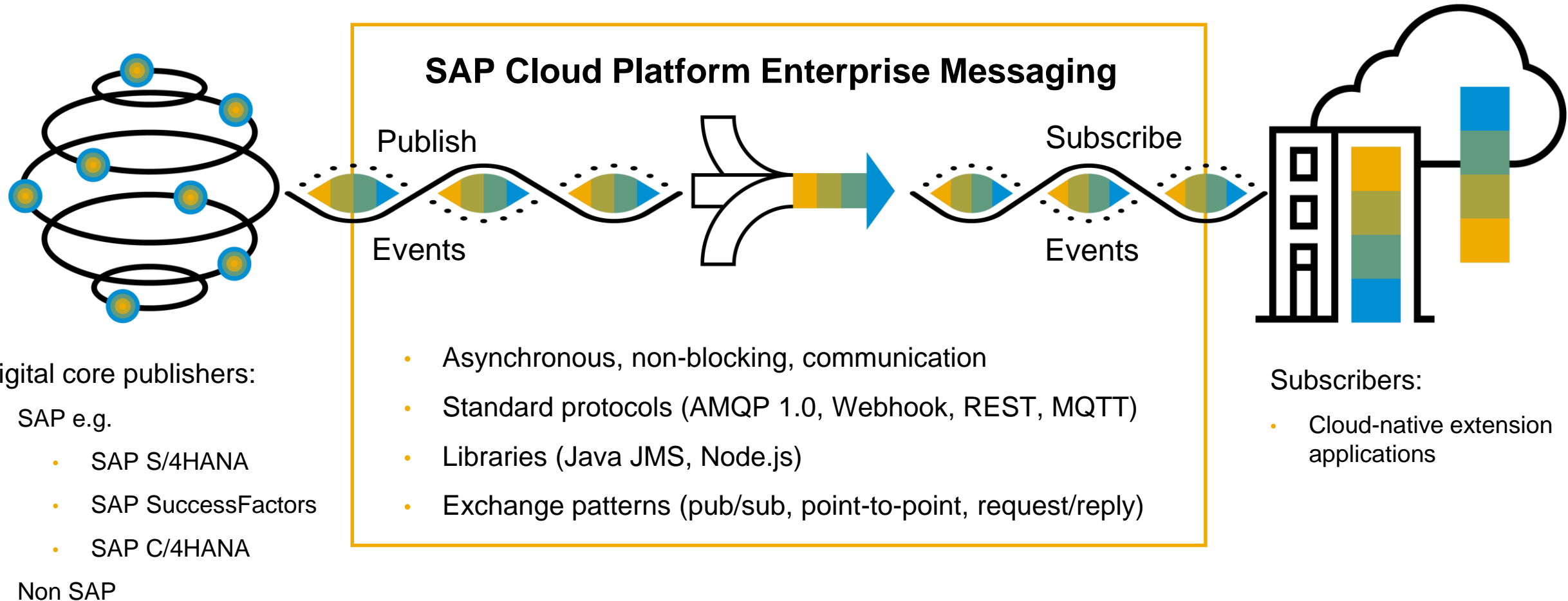
Event enablement

- Transports events (type of message) from event source to the Cloud for seamless integration of hybrid landscapes.
- Build responsive applications and extensions that can react to events without disrupting core business systems.

Benefits

- ✓ **Heterogeneous integration:** connect and integrate all kinds of different clients.
- ✓ **Fire and forget:** message broker delivers messages to consumers so applications can concentrate on their core competencies.
- ✓ **Scalability:** decouples communication between consumers and producers to handle peak loads (e.g. rush hour).
- ✓ **Buffering:** message broker buffers data during times of high load and when consumers are offline, which means no loss of data.
- ✓ **Flexibility/agility:** loose coupling enables you to switch seamlessly between architectures if environment changes.

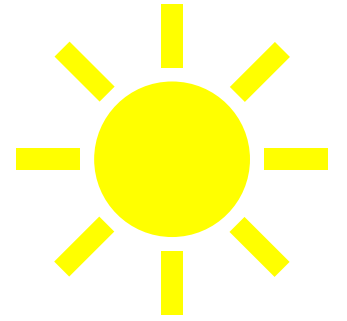
Event bus - decoupled communication between digital core and cloud



Event-driven architectures **with SAP**



Event-Driven Architectures – Your food is ready



What is an event?

- **An event is a significant change in state** (of an object in an enterprise system)
- **Event notifications** can be sent from the system in which the event occurred to inform other systems, microservices, and applications of the change.



cloudevents

- **cloudevents** is a specification for describing event data in a common way
- SAP, Google, Microsoft ... support cloudevents

Last Events



Enterprise Messaging:

```
{"eventType":"BO.BusinessPartner.Changed","cloudEventsVersion":"0.1","source":"https://S4HANAOD.sap.com","eventID":"ABY+SjviHumz99TYDcdQrw==","eventTime":"2019-09-05T09:22:16Z","schemaURL":"https://S4HANAOD.sap.com/sap/opu/odata/IWXBE/BROWSER_SRV/","contentType":"application/json","data":{"KEY":[{"BUSINESSPARTNER":"0000000211"}]}}
```

Topic: S4/BO/BusinessPartner/Changed · 9/5/2019, 11:22:16 AM

Event-Driven Architectures in Nature

Event

A significant change of status in a spider's web

Event

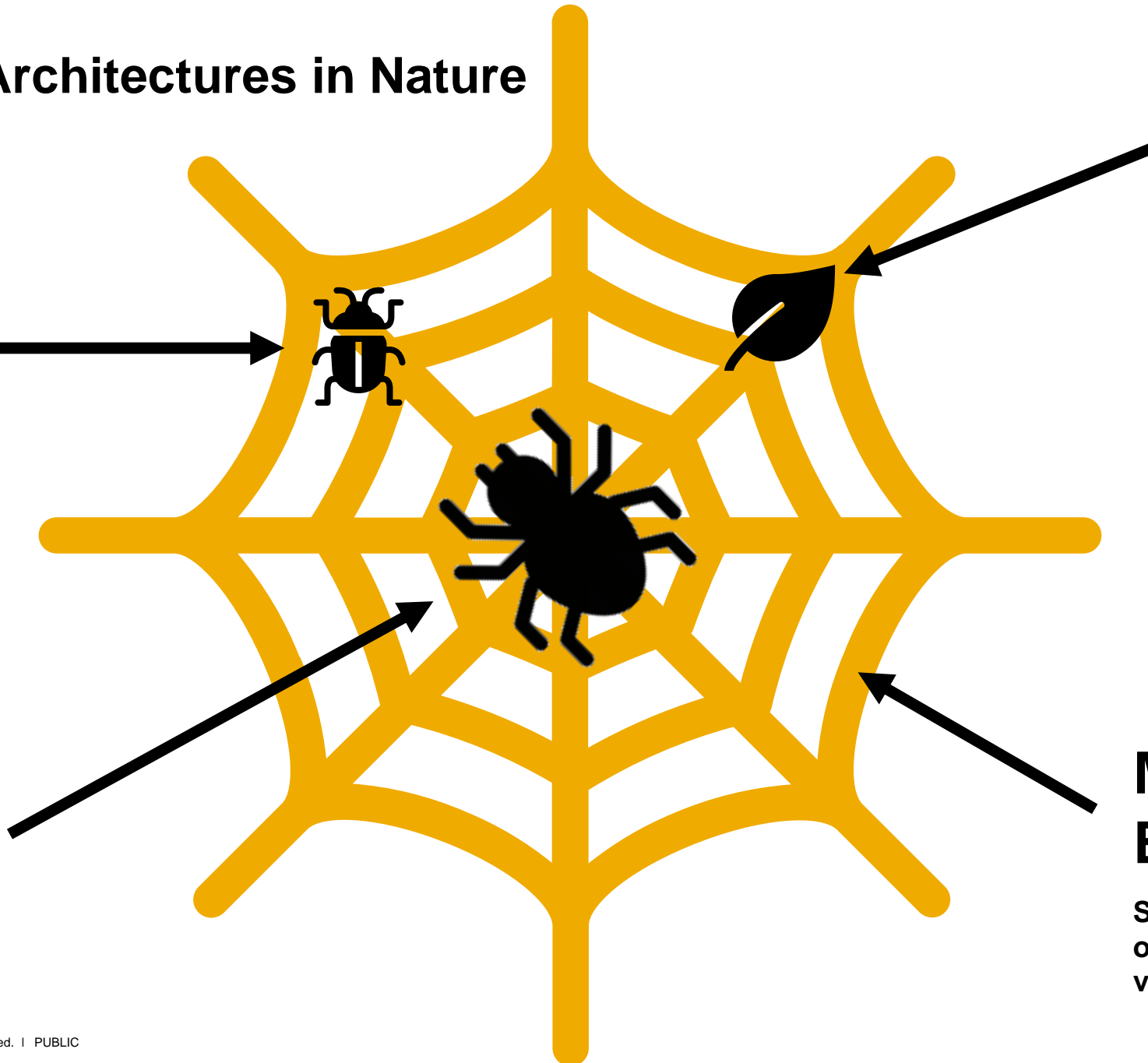
A significant change of status in a spider's web

Event Consumer

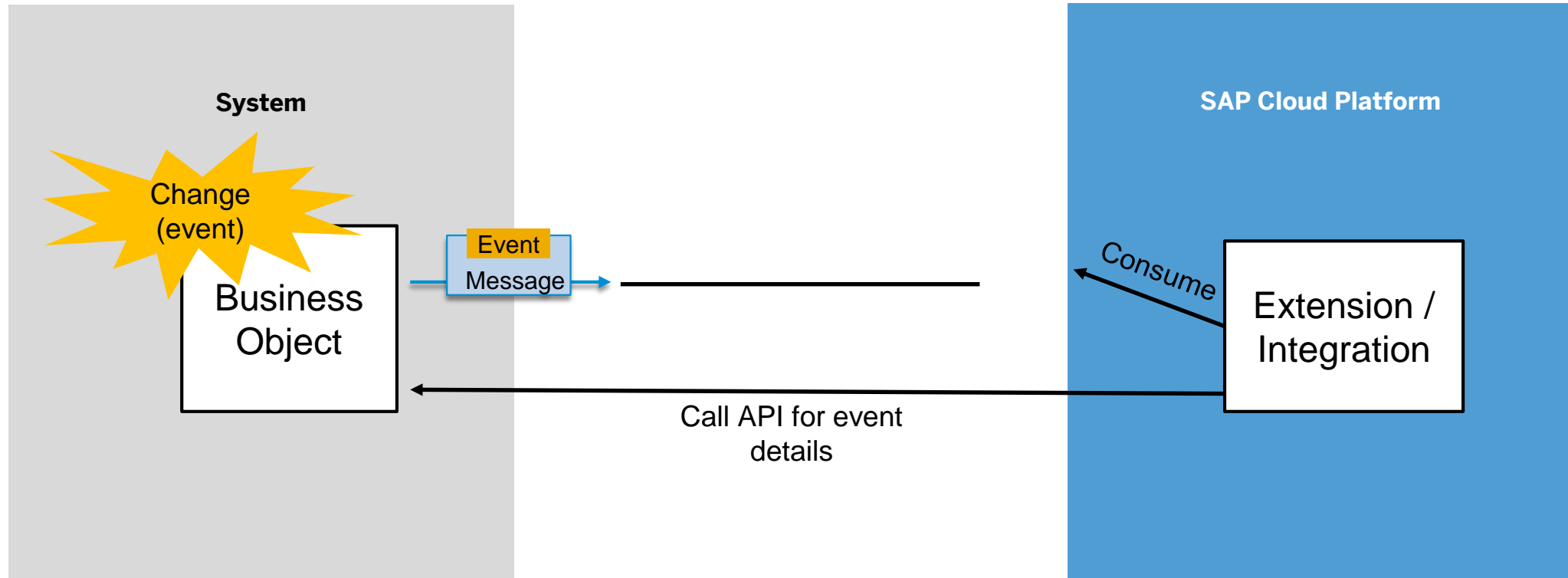
Listening for and reacting to events

Messaging Broker

Spider's web passes on events via vibrations



Event-driven extensions / integrations with SAP



Highly **decoupled**, single-purpose event processing components that asynchronously receive and process events.

Senders of events are **not aware** of event **receivers**.

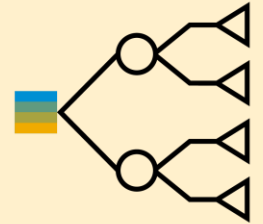
Event-Driven Architectures

Defining elements





- **Distributed asynchronous architecture pattern** used to produce highly scalable applications.
- Made up of **highly decoupled, single-purpose event processing components** that asynchronously receive and process events.
- A key element of **event notification** is that the source system doesn't really care much about the response.

Benefits

- **Decouples** communication
- Removes idle downtimes
- Enables **responsive applications** that react to events in other systems/applications
- Permits **events to trigger functions** (as a service) in response to execute defined business logic
- **Facilitates pay-per-use consumption** costs by removing “always on”



Event-Driven Architectures – scenarios and use cases

| | Real-Time Event Information | Event-Driven Architectures | | |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use Case | Inform all interested apps/services of changes, updates, and additions to data in underlying processes | Distributed and Flexible Architectures | Isolation | Resilience |
| Description | An event , a change of state , is communicated to all interested parties. | Event processing engines and downstream event-driven activities can flexibly change over time allowing for extremely loose coupling and a highly distributed landscape. | Isolate event publisher from other components by limiting access channels to events for information and clearly defined APIs for access. | Loose coupling leads to higher resilience since components, specifically event publisher, do not block and can operate independently from each other in case of a failure of a component |
| Advantages | <ul style="list-style-type: none"> Events are communicated as they happen (real-time) Components still remain autonomous Interested parties can listen to events in different networks Is non-blocking for event publisher (fire and forget) Event publisher and event processing engine can work at their own speed Enables greater responsiveness | <ul style="list-style-type: none"> Supports highly distributed architectures Event publisher does not need to know about architecture or who is listening to events Provides great flexibility Allows for a completely new development paradigm | <ul style="list-style-type: none"> API calls only when needed (after relevant event received) Clearly separates / isolates components Is highly scalable | <ul style="list-style-type: none"> Systems remain reactive Higher resilience |
| Example Project | Car Production Updates – TechEd 2018 Keynote  | Cache data in the Cloud – Chemical Company  | Restrict backend access – CP Mission  | Protect backend – Airline  |

How to Event Enable SAP Backends



Selected SAP Event Sources

SAP S/4HANA Cloud +
On Premise¹ 

| | |
|----------------------------------------|-----------------------------------------------------------------------|
| Billing Document Request | Created, cancelled |
| Business Partner | Created, changed |
| Business Situation | Created, changed, completed |
| Contract Accounting Billing Document | Created |
| Contract Accounting Invoicing Document | Created |
| Credit Memo Request | Created |
| Customer Returns Delivery | Created, changed, deleted |
| Debit Memo Request | Created, changed, deleted |
| Inbound Delivery | Created, changed, deleted |
| Kanban Control Cycle | Created, changed, deleted, Container created, changed, deleted |
| Legal Document | Created, accepted, Version created |
| Legal Transaction | Created, cancelled, expired, terminated |

A number of business objects have been event-enabled with more to come. Can be used out-of-the-box.

¹ Planned innovation

SAP SuccessFactors



| Recruiting | Onboarding and Offboarding | Hiring | Job Changes | Talent Management |
|---------------------------|----------------------------|-----------------|--------------------|---------------------------|
| Update of Job Requisition | Onboarding Step Complete | Employee Hire | First Time Manager | Add Development Objective |
| Update of Job Application | Offboarding Step Complete | Employee Rehire | Change in Manager | Spot Award |
| Approval of Offer Detail | | | Job Transfer | New Successor |

SuccessFactors Intelligent Service events can be enabled for Enterprise Messaging following this blog [How to use SAP Enterprise Messaging as an event bus for SuccessFactors](#)

Setup and Configuration for a S/4HANA Event Driven Architecture

Make EM Instance ready for Eventing - Service Key with clientID, clientsecret and URI

Create S/4 Communication System, User and Arrangement

1

Maintain Event Topics for a Channel

2

Queue Subscription to the Event Topic Pattern

3

SAP S/4HANA

Change
Business Object

Event Channel

2

Communication Arrangement

1

Communication System & User



SAP Cloud Platform

Topic

Queue

SAP Cloud Platform Enterprise Messaging

Consumer A

Consumer B

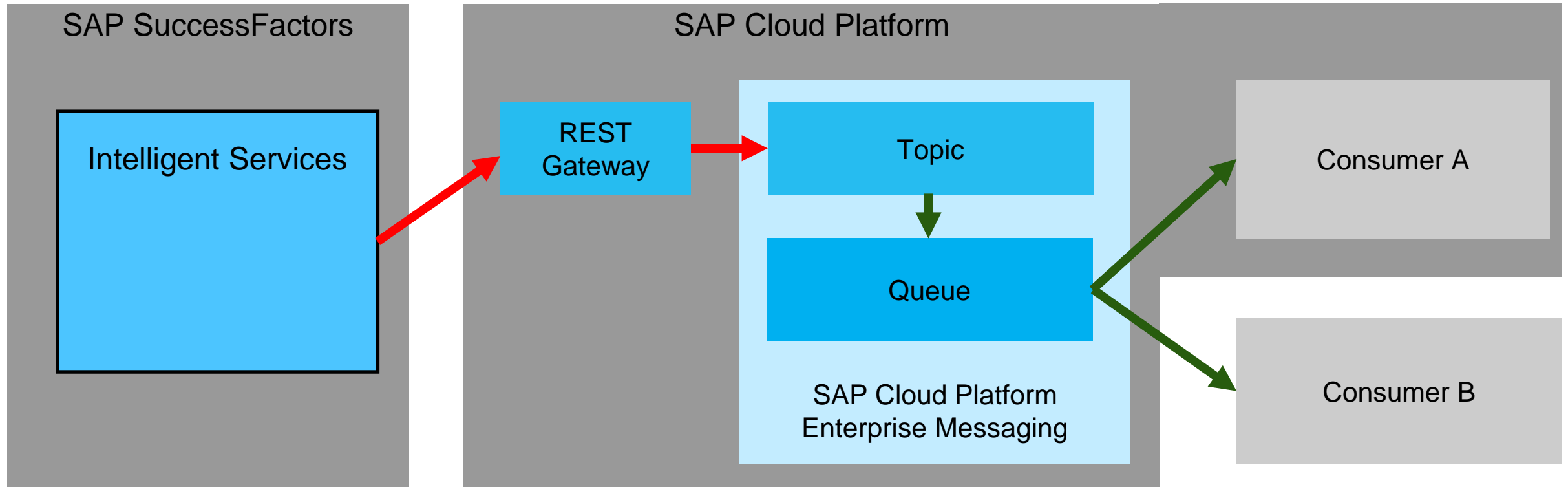
SAP SuccessFactors / SAP Cloud Platform Enterprise Messaging EDA

Make EM Instance ready for Eventing - Service Key with clientID, clientsecret and URI

Enable Intelligent Services

Expose Intelligent Services Events in Admin Center

Queue Subscription to the Event Topic Pattern

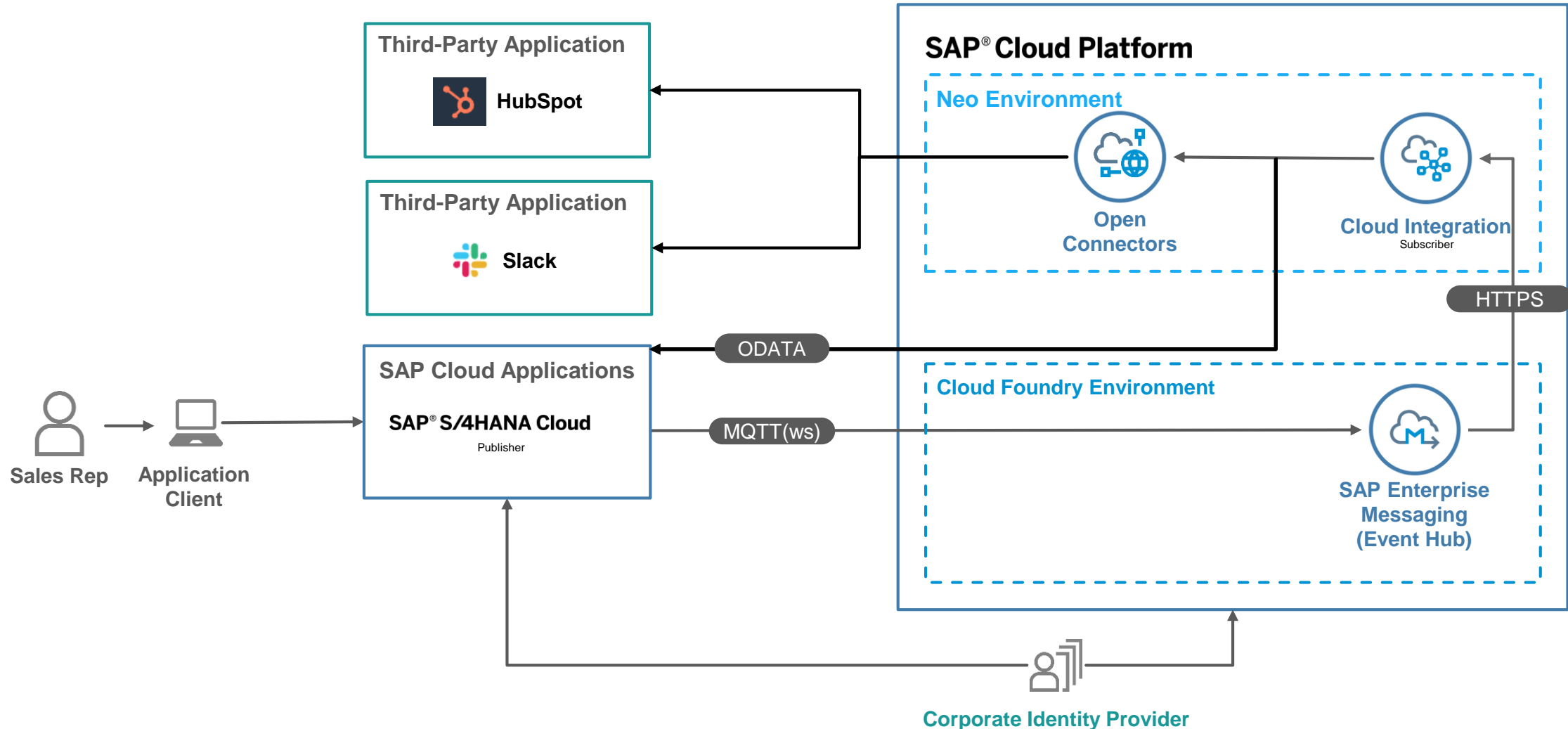


Selected Use Cases

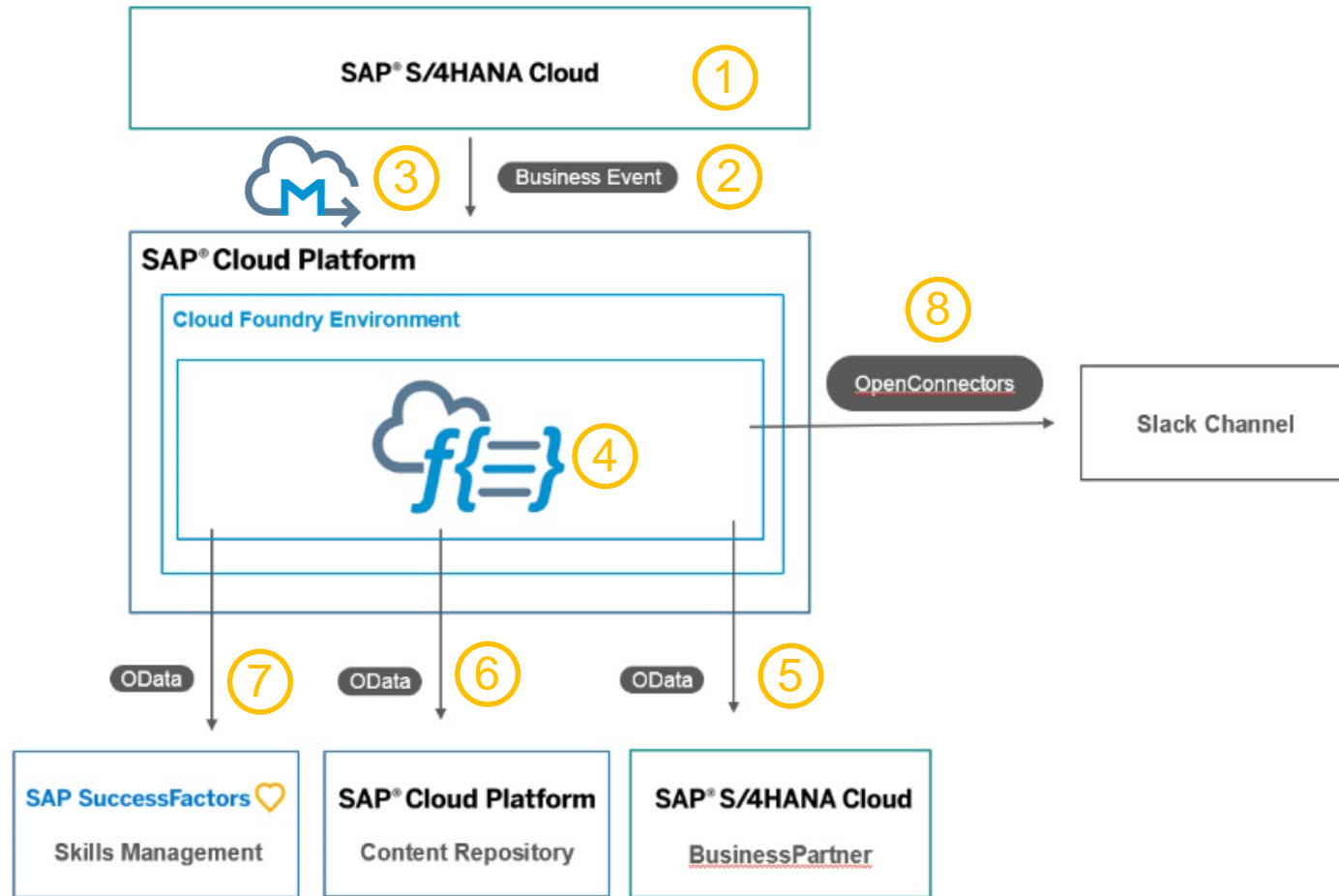


TechEd 2019 Hands-On Exercise: Integration using Event Pattern

Business Partner Replication from S/4Hana Cloud to Slack and HubSpot



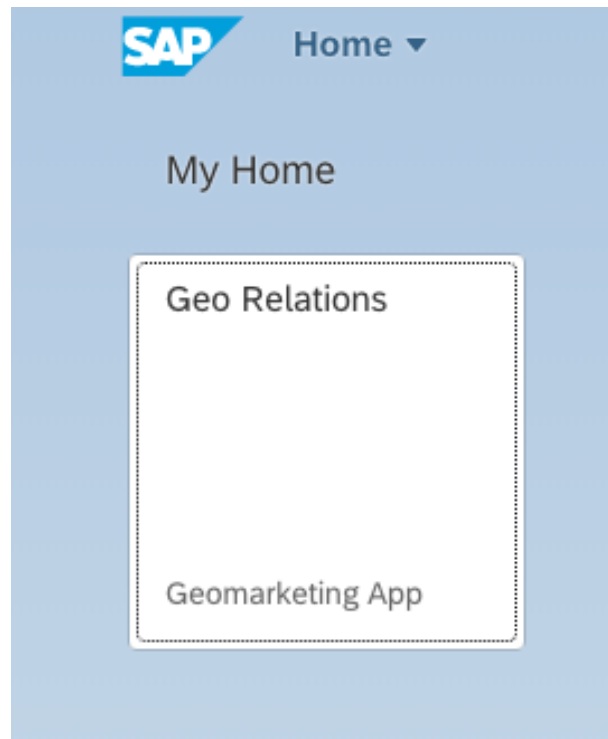
TechEd Keynote Scenario - Barcelona



1. Business partner created in SAP S/4HANA.
2. Generates event BusinessPartner.Created
3. Event sent using Enterprise Messaging to extension application where it triggers short-lasting function.
4. Function extracts BusinessPartnerID and EmployeeID from event.
5. Uses API call to retrieve employee's name from SAP S/4HANA.
6. Calls Consent Repository using OData service to ascertain permission to share data.
7. Calls Skills Management to retrieve skills.
8. Data retrieved is published to Slack Channel using Open Connectors.

SAP S/4HANA Cloud & Cloud Platform Mission Example App

Extension application on SAP Cloud Platform to react to changes in S/4 BusinessPartner BO



The screenshot shows the 'Manage Geo Relations' application interface. At the top, there is a header bar with the SAP logo and a 'Manage Geo Relations' dropdown menu. Below the header, there is a search bar and several filter fields: 'Editing Status:' (set to 'All'), 'Customer ID:', 'Customer Status:', and 'Processing Status:'. Below these filters, there is a table titled 'Customer Processes (8)' with a 'Create' button in the top right corner. The table has four columns: 'Customer ID', 'Customer Name', 'Customer Status', and 'Processing Status'. The table contains 8 rows of data.

| <input type="checkbox"/> Customer ID | Customer Name | Customer Status | Processing Status |
|--------------------------------------|----------------------|-----------------|-------------------|
| <input type="checkbox"/> 111102 | Lutz Schneider | Win-Back (3) | Critical |
| <input type="checkbox"/> 1000064 | Rainer Tomski | | Kick-Off |
| <input type="checkbox"/> 233 | Helena Moserr-Müller | Inactive (4) | Closed |
| <input type="checkbox"/> 1000007 | Helena Vega | Active (2) | OK |
| <input type="checkbox"/> 1000066 | Bradley Morrison | | Kick-Off |
| <input type="checkbox"/> 1000932 | John Smith | Active (2) | Closed |
| <input type="checkbox"/> 1000055 | Jackie Sharp | | Kick-Off |
| <input type="checkbox"/> 211 | Florentina Abt | | Kick-Off |



Short Demo

Key takeaways

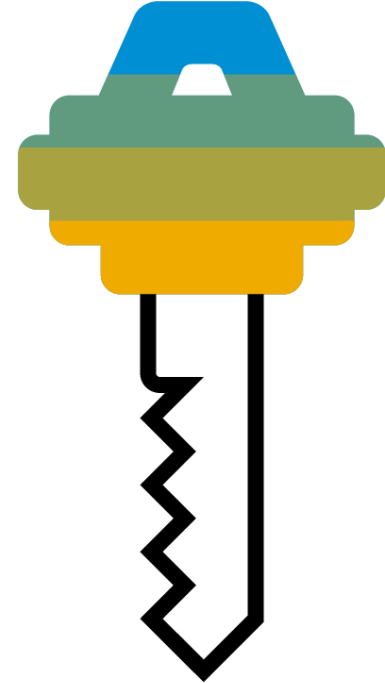


Event Brokering Using SAP Cloud Platform Enterprise Messaging

Key takeaways

SAP Enterprise Messaging

- Is **SAP's Messaging as a Service** and highly scalable
- Supports **extensible event-driven integrations**
- Is available as Message broker standalone or as part of SAP offerings for the Intelligent Enterprise like the **Integration Suite**.



More information

Event Bus:

- SAP Community: <https://community.sap.com/topics/enterprise-messaging>
- Blog: <https://blogs.sap.com/2019/01/17/sap-cloud-platform-enterprise-messaging-as-an-eventbus-for-successfactors/>

Extensibility:

- SAP Help Portal:
 - [Extending SAP Solutions](#)
 - [Extending SAP SuccessFactors](#)

Questions and Answers



Thank you.

Karsten Strothmann
Product Management
SAP Cloud Platform Core