

Integration with Salesforce Platform

Key data integration patterns

Data by value

- Entire datasets are copied between systems
- Facilitated by ETL tools
- Local access is an advantage
- **Stale data is a problem**

Data by reference

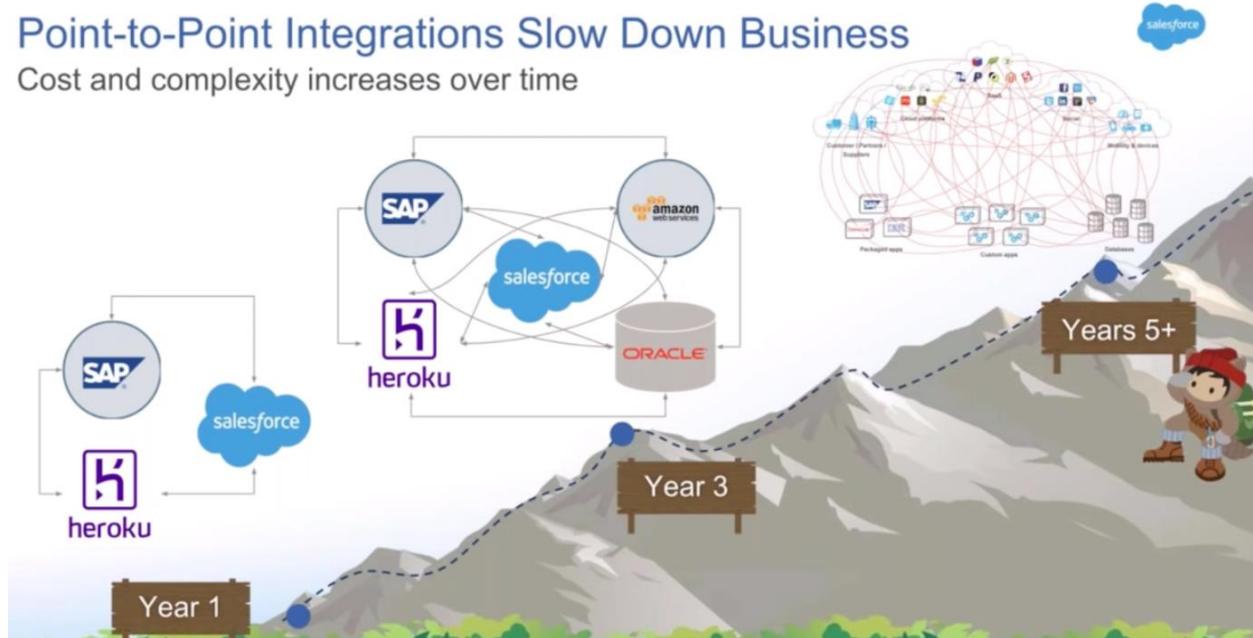
- Virtualized datasets
- Queries served in real time
- Always up to date
- **Latency and aggregation is a problem**

Event-driven architecture

- Business process
- Asynchronous interaction
- Large-scale
- Decoupled
- **Asynchronous model is difficult to get right**

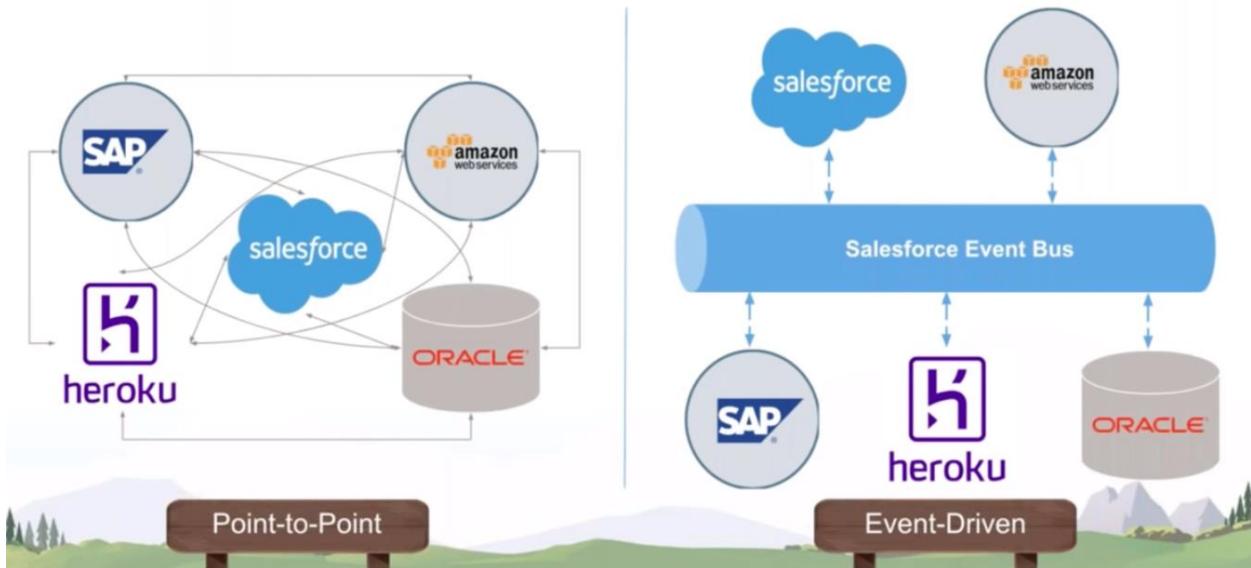
Point-to-Point Integrations Slow Down Business

Cost and complexity increases over time



Scale Integrations with an Event-Driven Architecture

Move from Point-to-Point to a robust, decoupled model



Pros/Cons of Salesforce Canvas



Advantages

- Developers can create applications using the technology stack most familiar to them.
- Canvas SDK includes CSS classes that allow an embedded app to have a seamless UI experience.
- Configuration within Salesforce couldn't be easier - just a few simple steps.
- No additional licenses costs.



Disadvantages

- The web application that must be exposed via Canvas must exist.
- In order to fully leverage Canvas, you will have to use the Canvas SDK, which effectively means it can't be reused in a stand-alone fashion.
- As with nearly every Salesforce solution - developer limits are in place. For example, you are limited to 5,000 Canvas SDK calls per 24 hour period. 9

Getting Data Out of Salesforce

In near real-time

- On platform options
 - 1. Outbound Messaging
 - 2. Apex Callouts
 - 3. Salesforce Connect (External Data Source)
- Off platform options
 - 1. Heroku Connect
 - 2. Streaming API
 - 3. Change Data Capture
 - 4. Polling (REST/SOAP)

Outbound Messaging

Pros

- Automatic Retry
- No Code Required

Cons

- No Guaranteed Order
- No Guaranteed Timeframe
- Not Great with Large Data Volumes
- No Related Data
- Lack of Strong Security

Outbound Messaging

Introduction



Complexity

Simple

Purpose

Data Integration

PROS

1. Send Salesforce data (SOAP messages) to external web service endpoint
2. Trigger integration via workflow
3. Automated retry and batching

CONS

1. Mandates Message format
2. Doesn't support complex structure
3. Doesn't support various authentication mechanisms

Pros/Cons of Outbound Messaging

Clip slide



Advantages

- Simple to setup within Salesforce.
- Can be invoked via workflow rules and Apex.
- Up to 100 notifications can take place within a single SOAP message. This makes it very efficient.
- Guaranteed message delivery for up to 24 hours (retries for that duration).
- Asynchronous so no client latency.
- Session id automatically included for



Disadvantages

- Receiving service must implement Salesforces' outbound messaging WSDL.
- REST is not an option for outbound messaging.
- Governor/developer limits exist.

3

Outbound Messaging

How does it work?



Callouts

Pros

- Flexible Data Types
- More Security Options

Cons

- No Guaranteed Order
- No Guaranteed Timeframe
- No Retry
- Difficult to Log Failures
- Governor Limits
- Must be in an @future call

Pros/Cons of Salesforce REST API

 Clip slide



Advantages

- Simple to use with nearly all platform languages & technologies.
- A Javascript API ("Javascript REST Toolkit") enables client web applications to place direct calls to Salesforce (i.e., no same origin issue).
- Can support XML or JSON protocols.
- Very comprehensive - a wide range of Salesforce capabilities are exposed via the protocol.



Disadvantages

- As always, Salesforce developer limits apply. For example, depending upon your license, you may not be able to place more than 15k calls a day.
- OAuth is required for accessing standard REST API calls (but those you define via Apex can be exposed without OAuth).
- REST not available for outbound messaging (requires SOAP).

Pros/Cons of Salesforce SOAP API



Advantages

- Platforms such as .NET and Java can leverage code-generation tools that make it fairly straightforward for generating SOAP requests. SDK's exist to further streamline the process for Java, .NET, PHP and Ruby.
- Testing the SOAP web services can be done using free tools such as SoapUI with IDEs such as Eclipse.



Disadvantages

- As always, Salesforce developer limits apply. For example, depending upon your license, you may not be able to place more than 15k calls a day.
- SOAP isn't an option for accessing the Bulk or Chatter APIs - they are only exposed as REST calls.

External Services

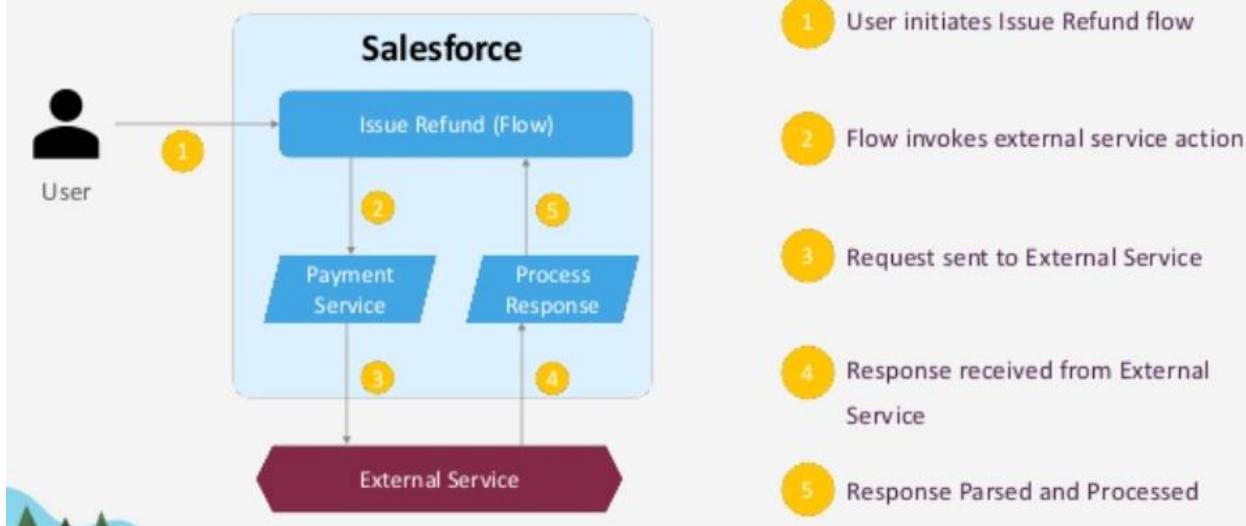
Introduction

Complexity	Moderate	Purpose	Application
PROS		CONS	
<ol style="list-style-type: none">1. Invoke custom Web APIs (Interagent, Swagger Open API 2.0)2. Trigger integration via Flow		<ol style="list-style-type: none">1. Schema limited to 100,000 characters2. Less options to debug	

External Services

Clip Side

How does it work?



Salesforce Connect

Pros

- Real-time
- Low Development Cost

Cons

- Cost
- Reliant On Endpoint Type
- Limited Write Capabilities

Pros/Cons of Salesforce Connect



Advantages

- If your external data already supports the OData protocol, setup within Salesforce couldn't be simpler.
- External objects are treated nearly identically to custom objects.
- Configuration can largely be done in a point-and-click fashion - no custom code required.
- External objects can be included in



Disadvantages

- External data integration limits are applicable. For example, there's a maximum of 50,000 records retrieved or created an hour.
- Debugging tools are weak.
- Performance is a wild card - caching and optimization strategies are largely unknown.
- Salesforce charges an additional monthly fee for Salesforce Connect.

External Objects

Introduction



Complexity

Simple

Purpose

Data Integration

PROS

1. Access remote data like native data/objects without consuming salesforce data storage
2. Ability to **search, view** and report on data like native data and basic **CRUD** operations
3. Uses widely used common data standards (OData, FederatedSearch etc.)

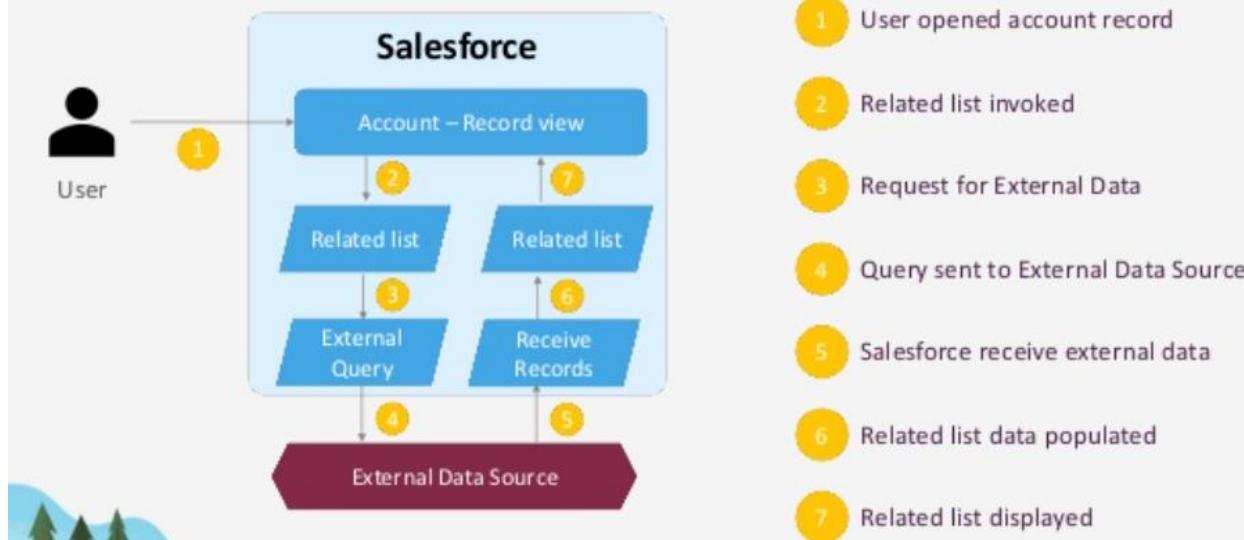
CONS

1. Limited OData support across legacy applications
2. Higher network usage
3. Salesforce connect licenses costs

External Objects

L+ Clip slide

How does it work?



Heroku Connect

Pros

- Bi-directional Data Sync
- Fairly Turn-key

Cons

- Cost
- Might Be Overkill

Streaming API

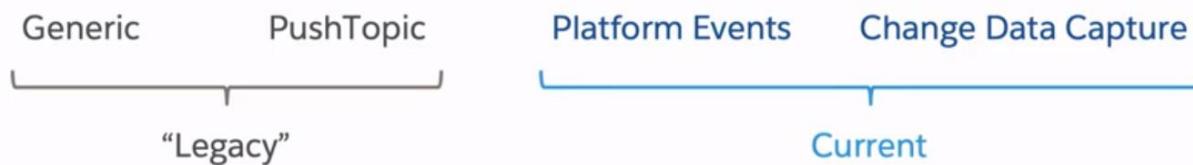
Pros

- True Realtime
- Message Reliability
- Message Durability

Cons

- Lots of Work
- Governor Limits Can Be Tricky

The Four Streaming APIs



What's different between those Streaming APIs?



	PushTopic	Generic	Platform Events	Change Data Capture
Data structure	Defined by target object via SOQL	Unstructured (string)	Custom typed fields	Defined by target object
Publication	Automated	REST API	Declarative or Code	Automated
Subscription	Code		Declarative or Code	

Streaming Events



Features	PushTopic Event	Change Data Capture Event	Platform Event	Generic Event
Event Type	Data	Data	Custom	Custom
Publish & Receive	Fields in a SOQL query (1300 characters only)	All new and modified fields	Custom payloads with a predefined schema	Custom payloads without a defined schema
Publish Events	Auto-Publish	Auto-Publish	Apex methods, Process Builder, Flows Builder, APIs	APIs
Replay Events	1 day	3 days (up to 72 hours)	*3 days (High Volume ONLY)	1 day
Supports Shield Platform Encryption	X	Yes	X	X
Supported Objects	Custom and subset of Standard objects <i>(*User, Person Accounts, Events, Product2 included)</i>	Custom and subset of Standard objects <i>(*User, Person Accounts, Events, Product2 included)</i>	N/A	N/A
Fields included in Events	Limited fields (<i>in SOQL query</i>)	All Fields	Limited fields	N/A
Supports DML operations	Create, Update, Delete, or Undelete	Create, Update, Delete, or Undelete	N/A	N/A
Granular Level Details	X	<ul style="list-style-type: none"> Change Origin (UI or APIs) Who made the change? When was the change made? Transaction Keys 	X	X

Feature	PushTopic Event	CDC Event	Platform Event	Generic Event
Define a custom schema as strongly typed fields	N/A	N/A	✓	✗
Include user-defined payloads	N/A	N/A	✓	✓
Publish custom events via one or more APIs	N/A	N/A	✓	✓
Publish events via Apex	N/A	N/A	✓	✗
Publish declaratively using Process Builder and flows	N/A	N/A	✓	✗
Publish to specific users	N/A	N/A	✗	✓
Subscribe via CometD using JavaScript, Java, and other languages	✓	✓	✓	✓
Subscribe via Apex triggers	✗	✗	✓	✗
Filter subscriptions	✓	✗	✗	✗
Receive auto-published event notifications for Salesforce record changes	✓	✓	N/A	N/A
Choose the fields to include in event notifications for Salesforce record changes	✓	✗	N/A	N/A
Receive a versioned event schema	✗	✓	✓	✗
Get field-level security	✓	✓	✗	✗
Get record-sharing support	✓	✗	N/A	N/A
Encrypt field data with Shield Platform Encryption	✗	✓	✗	✗
Replay retained event notifications	✓	✓	✓	✓
Event retention period	1 day	3 days	1 day or 3 days*	1 day

* High-volume platform events are retained for up to 3 days.

Business Use Cases

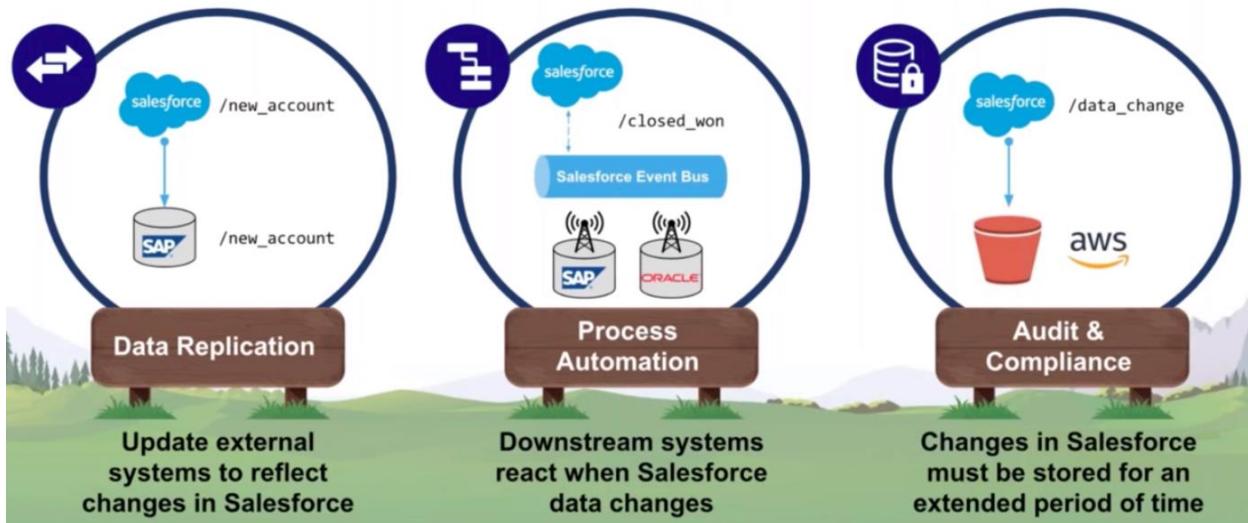
Integration Challenges

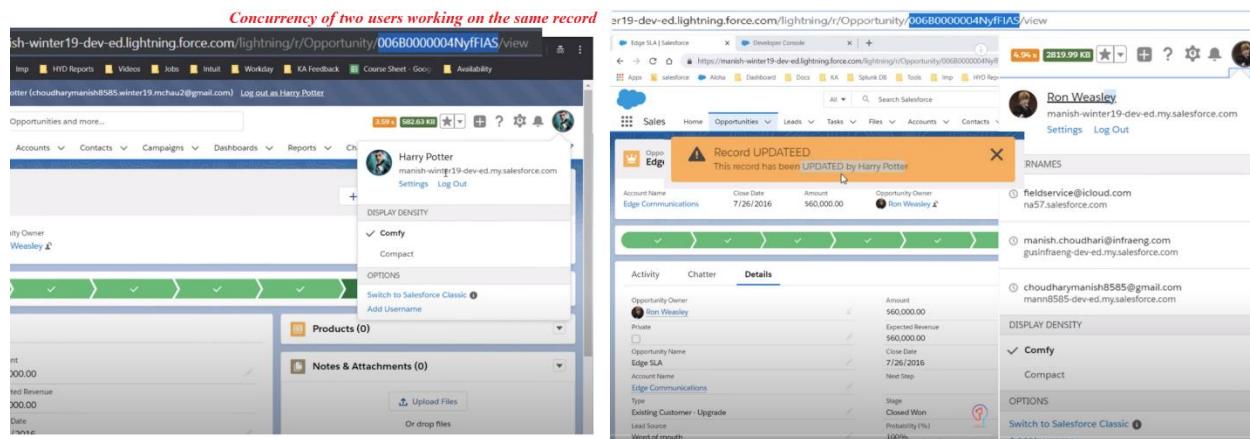
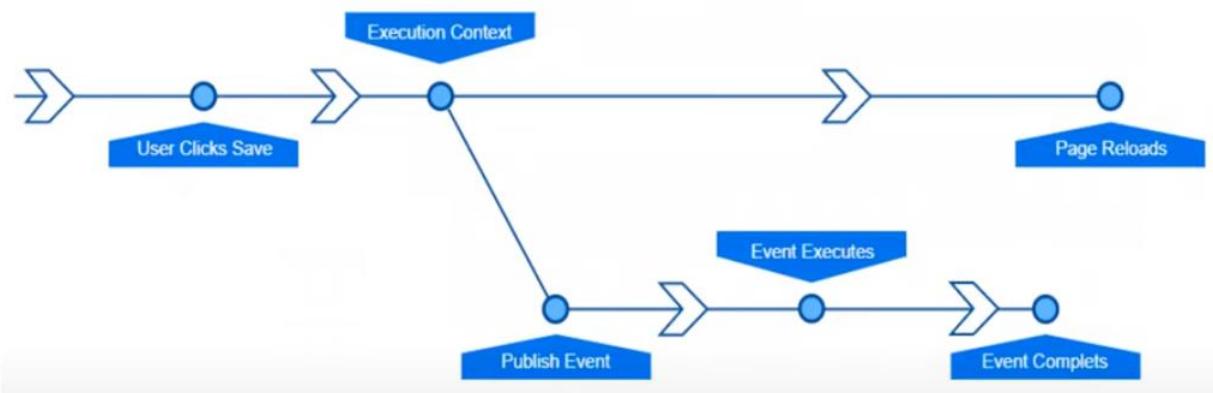
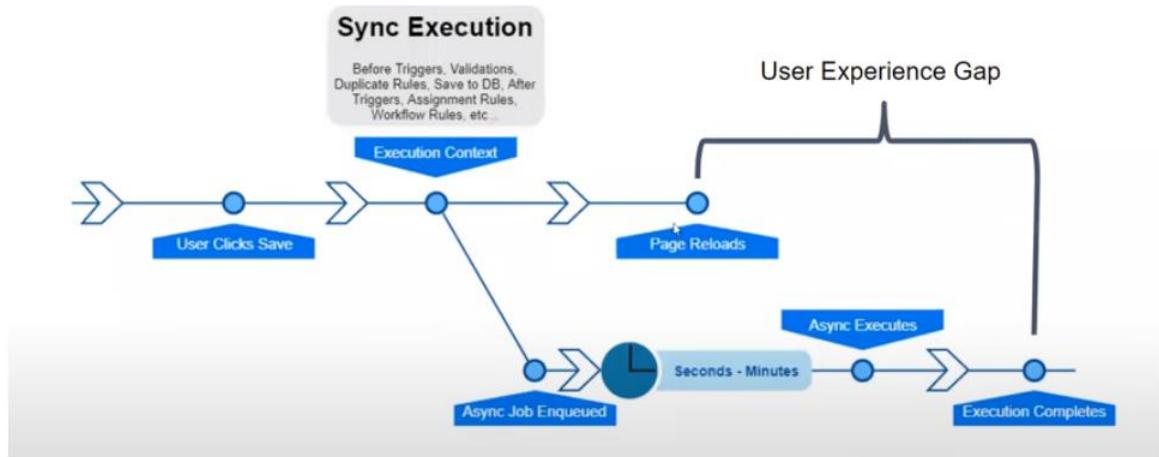


1. Data Replication - Record every action
2. Real-time Data Synchronization – Keep all distributed systems Up-To-Date
3. Data Auditing & Compliance – Track every User activity



Change Data Capture Unlocks Real-Time Data Sync





Never ever mix the insertion/updating the logging records with real-time data (callout response) in a single transaction.

Ex: Let callout success (salesforce received the response from external system successfully), parse the response into database columns and for audit purpose we logged the response in the audit tables.

If response(payload) size is greater than column size in the audit table, error occurred, transaction fails.

Success callout response also rollback.

Best practice: Split into two transactions (parse and save the response in the database into one transaction and logging into the audit tables into another transaction)

Exception Handling & Logging

<Apex HOURS/>



Exception
Unexpected event occurred during your transaction execution



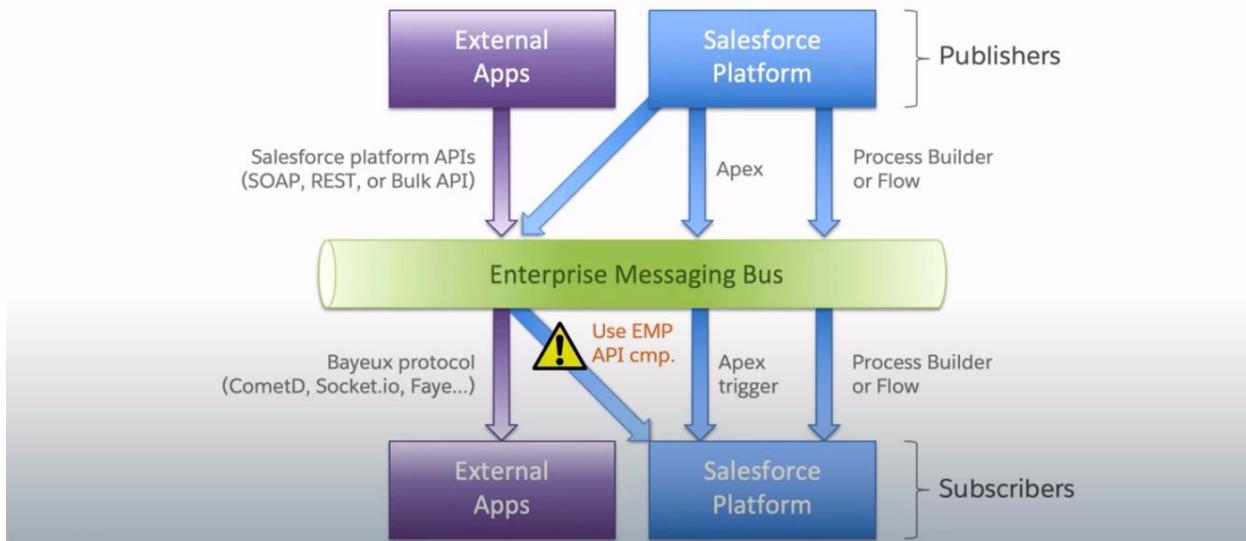
Exception Handling
Deal with unexpected Exception



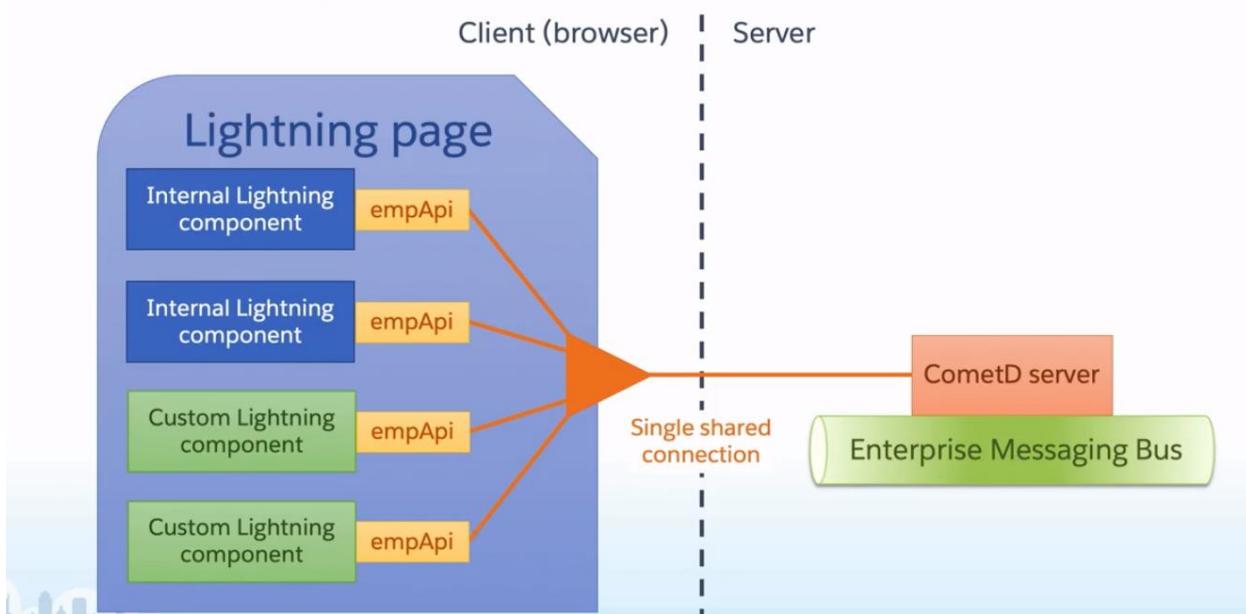
Exception Logging
Store exception details for later analysis and trouble shooting



Platform Events and Change Data Capture



The EMP API Lightning Component



Platform Events

Introduction

Complexity	Moderate - Complex	Purpose	Application
PROS		CONS	
<ol style="list-style-type: none">1. Persistent and immutable messages2. Enables Loosely coupled functionalities3. Enables pub-sub pattern (publisher – subscriber)4. Replay past events		<ol style="list-style-type: none">1. Can result in complicated technical design2. DML limits apply3. Limits on subscribers and events to publish	

Platform Event

How does it work?



Platform Events - Change Data Capture

Platform event:

- Like a **Salesforce Object**: describable, packageable, API, Apex support, Created user/date, custom fields
- **Immutable**: An event can be created, but not updated or deleted. An event can't be undone.
- **Replay Id**: Sequence of published events

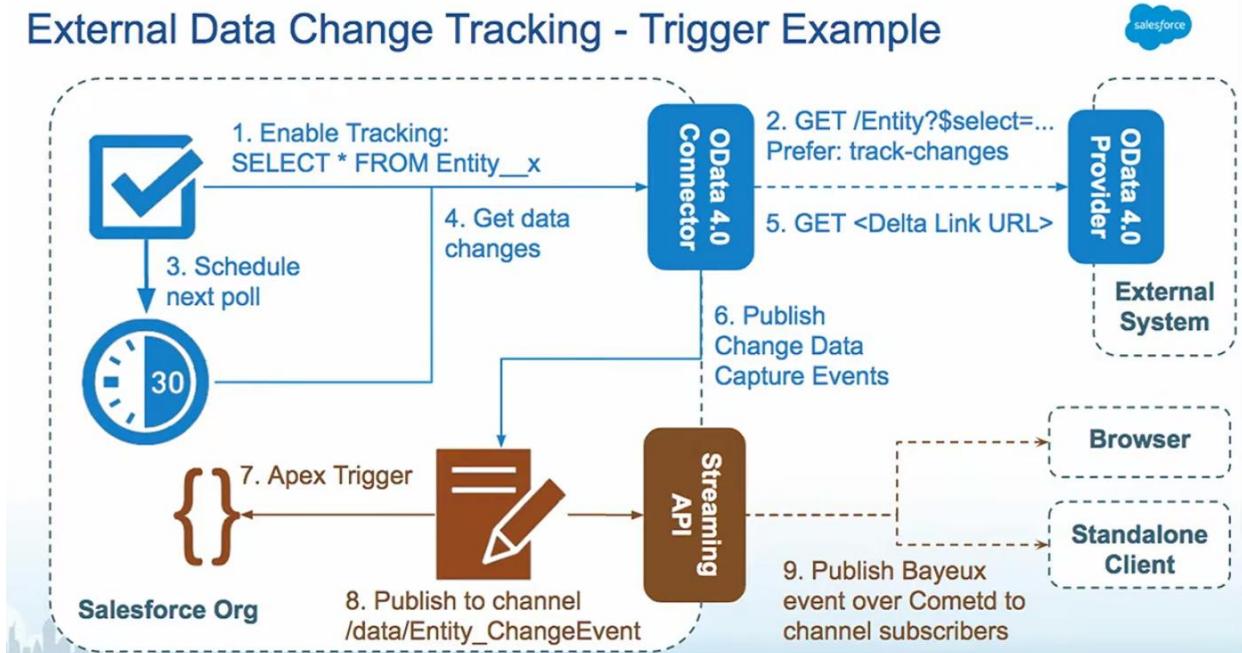
Change data capture event:

- **Is a Platform Event**
- **Associated** to object causing change
- **Schema**: Relates to entity causing change
- **Creatable by Event Source** only
- **Change Event Header**: Common data change information

External Object Products_x	
Standard Fields	
Id	x03...a
ExternalId	21507
CreatedOn	28 May 2019
CreatedBy	005...n
Custom Fields	
Name_c	Fendant
UPrice_c	8.50
Stock_c	1204

Associate Entity Change Data Capture Products_ChangeEvent	
Standard Platform Event Fields	
ReplayId	15
CreatedOn	28 May 2019
CreatedBy	005...n
Change Event Header Fields	
EntityName	Products_x
RecordIds	[x03...a]
...	...
Changed External Object Fields	
UnitPrice_c	8.50
...	...

External Data Change Tracking - Trigger Example



Change Data Capture

Pros

- True Realtime
- Message Reliability
- Message Durability

Cons

- No Related Data
- Lots of Work

Polling (REST / SOAP)

Pros

- Get Exactly What You Want
- Bundle Multiple Queries
- Large Data Ready

Cons

- API Call Limits
- Number of Records
- Datetimes Stored to Second
- Frequency

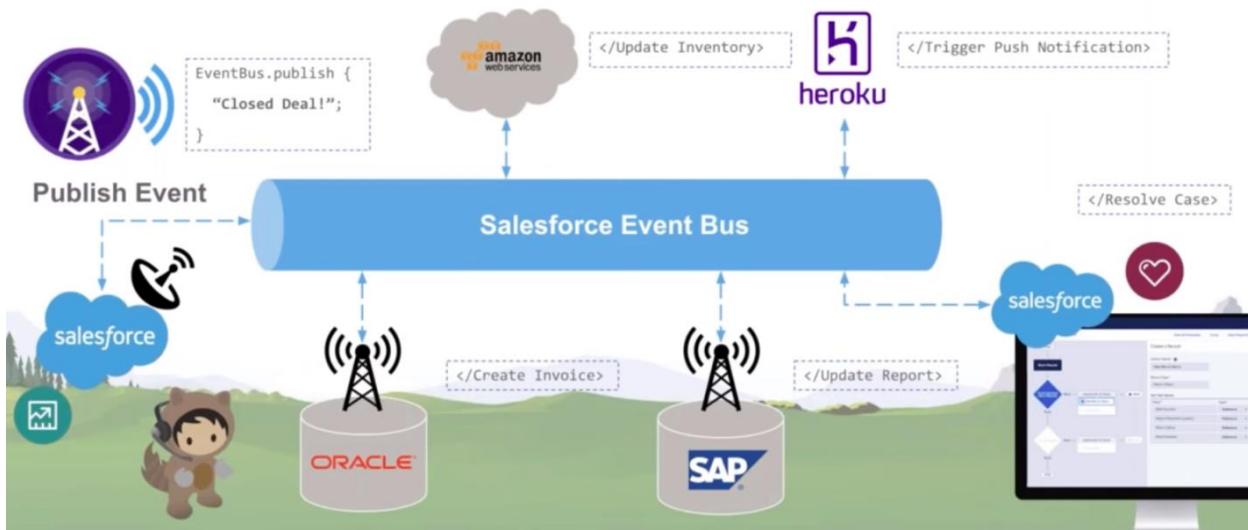
Roadmap



Winter '19 (#DF18)	Spring '19	Summer '19 +
CDC (Dev Preview)	Sharing Support	Filtered Subscriptions
Custom Object Default Enablement	Virtual Channel Definitions	Internal Multi-org Subscriptions
Platform Encryption Support	Apex Triggers on CDC Events	Cross-org Subscription
Monitoring APIs	Internal Enablement Toolkit	Additional Object Support
	Additional Object Support	

Real-Time Integration Across Systems with Platform Events

Drive robust process automation with an event-driven architecture

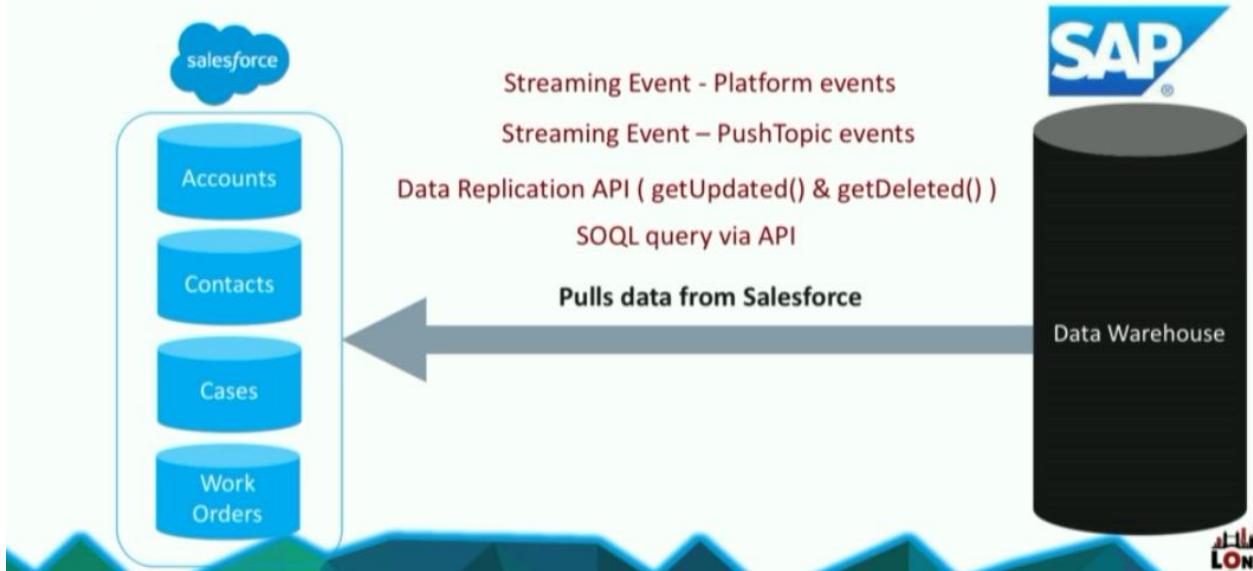


The screenshot shows the integration between a Heroku application and Salesforce. Three browser windows are displayed:

- Top Left:** A browser window showing the URL <https://cdc-inventory.herokuapp.com>. The page title is "Ikemoto Inventory for CDC". It displays a table of product inventory with one row (Iris) highlighted in yellow.
- Top Right:** A browser window showing the URL <https://gs0.lightning.force.com/one/one.app#/sObject/801B00000008acb>. The page title is "00000121 | Salesforce". It shows the Order detail for Order #00000121, which includes the highlighted "Iris" product row.
- Bottom Left:** A browser window showing the URL <https://api-inventory.herokuapp.com>. The page title is "Ikemoto Inventory for getUpdated". It displays a table of product inventory with the same highlighted "Iris" row.

Current Options for Change Data Capture

Customers Use Today



What does this mean?

- CDC allows you to collect information about the state of your data once a change has been applied
 - In Salesforce this means:
 - Creating a new record
 - Updating a record
 - Deleting a record
 - Undeleting a record
- With the information of what changed and when you can keep an external system up to date

Salesforce SOQL via API

- SQL-like query language to retrieve data from your org
 - Pros
 - Common query language on all orgs
 - Data is available in multiple formats
 - Data retrieved on-demand
 - Cons
 - Takes a long time to query large tables
 - Hard to capture granular changes
 - Client retains state for queries

Salesforce Replication API

- API calls to get updated or deleted records
 - Pros
 - Very simple to use
 - 600K records returned per call
 - Data retrieved on-demand
 - Cons
 - Multiple calls are needed to retrieve data
 - Hard to capture granular changes
 - Client must keep state of last time API called
 - Long running transactions can cause difficulties

Salesforce Streaming API

- Streaming based API which sends data to subscribers when data matches SOQL query
 - Pros
 - Customers define the capture SOQL
 - PushTopic Events are sent near real-time
 - Granular changes are captured
 - Cons
 - SOQL query has low limits (1300 characters)
 - Not all SObjects are supported
 - Multiple Apex updates can cause out of order events

Salesforce Platform Events API

- Streaming based API which sends data to subscribers when an event is published
 - Pros
 - Customers manage the capture logic
 - Platform Events are sent near real-time
 - Granular changes can be captured
 - Cons
 - Customers MUST manage the capture logic
 - Not all operations in the system can have a trigger associated
 - Limits may be low depending on change rates

Change Data Capture

- Capture every change in Salesforce and send the changes to subscribers
 - Event streams will be automatically set up for customers
 - Captured Events will be delivered in order
 - All objects will be (eventually) supported
 - Additional information will be provided in the event
 - Where did the call initiate
 - What was the transaction ID, SCN, execution order, etc...
 - What was the operation exacted on the record

Change Data Capture

Data meets events

Event-driven approach to data synchronization combines the benefits:

- Datasets are updated in near real-time
 - Polling load on the master copy is eliminated
 - Strong ordering of changes preserves replica's integrity

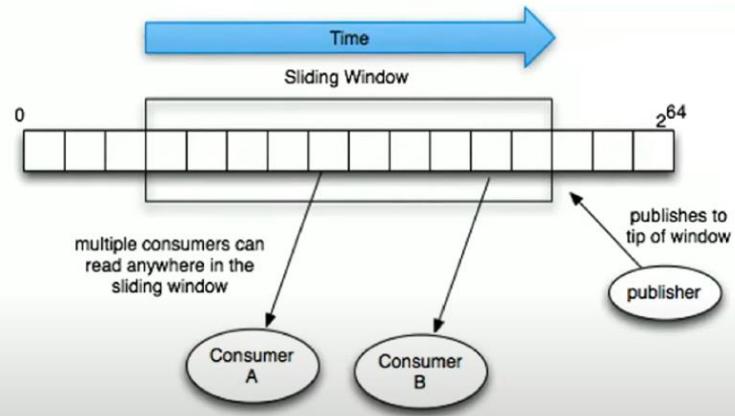
Uses Salesforce Event Bus for change distribution



Change stream durability

Fault-tolerant replication

- Time-ordered stream
- Resumed by replayId
- ~3 day replay window
- Consume at your pace



Delivery semantics

Avoid data loss or duplication



CDC

1. A streaming product on lightning platform .
2. Integrate between salesforce data and external system.
3. CDC is an outbound process from salesforce on real time basis.
4. CDC supports CRED and undelete operations.

CDC is nothing but Streaming Push:-

Description
Change Data Capture sends notifications to subscribers whenever a data change in Salesforce occurs.
Notification messages are sent to the event bus to which clients can subscribe through a channel.
Event-driven systems streamline the communication between distributed enterprise systems, increase scalability, and deliver real-time data.

When CDC:-

Type
Initial copy of entire set of data into the external system.
Continuous synchronization of new and updated data to the external system
Reconciliation of duplicate data between the two systems
it publishes the deltas of Salesforce data, whether for new records or changed records.
Change Data Capture requires an integration app for receiving events and performing updates in the external system.
Supports all CRED operations. Use a versioned event schema. Subscribe to mass changes in a scalable way. Get access to retained events for up to 3 days.

Security in CDC:-

Security Considerations

Learn about the user permissions required for subscription, field-level security, and Shield Platform Encryption.

Required Permissions for Subscription

Change Data Capture ignores sharing settings and sends change events for all records of a Salesforce object. To receive change events, the subscribed user must have one or more permissions depending on the channel that is subscribed to. The following table outlines the required permissions per channel.

Field-Level Security

Change Data Capture respects your org's field-level security settings. Delivered events contain only the fields that a subscribed user is allowed to view. Before delivering a change event for an object, the subscribed user's field permissions are checked. If a subscribed user has no access to a field, the field isn't included in the change event message that the subscriber receives.

Change Events for Encrypted Salesforce Data

If Salesforce record fields are encrypted with Shield Platform Encryption, changes in encrypted field values generate change events.

Security



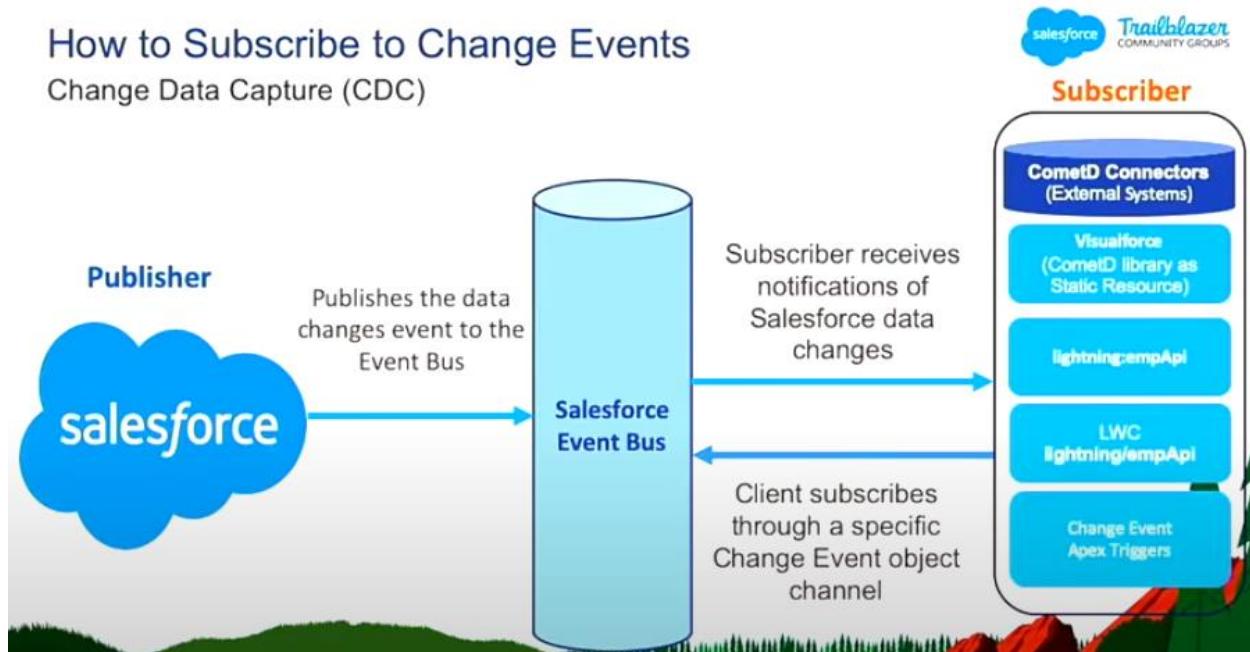
Field Level Security	<ul style="list-style-type: none"> Change Data Capture respects org's field-level security settings. Delivered events contain only the fields that a subscribed user is allowed to view. 	
Channel	Required Subscribed User Permission(s)	
/data/ChangeEvents	View All Data AND View All Users	
/data/User ChangeEvent	View All Users	
/data/<Standard_Object_Name> ChangeEvent Or /data/<Custom_Object_Name>_ ChangeEvent	View All for the object OR View All Data	
/data/ SAPEvents_chn /data/ OracleEvents_chn	View All for the object OR View All Data View All Users for User change events	

CDC Allocation:-

Default Allocations			
Description	Performance and Unlimited Editions	Enterprise Edition	Developer Edition
Maximum number of entities, including standard and custom objects, that can be selected for Change Data Capture	5	5	5
Event Delivery: maximum number of delivered event notifications within a 24-hour period, shared by all CometD clients	50,000	25,000	10,000
Event Publishing: maximum number of event notifications published per hour	250,000	250,000	50,000

How to Subscribe to Change Events

Change Data Capture (CDC)

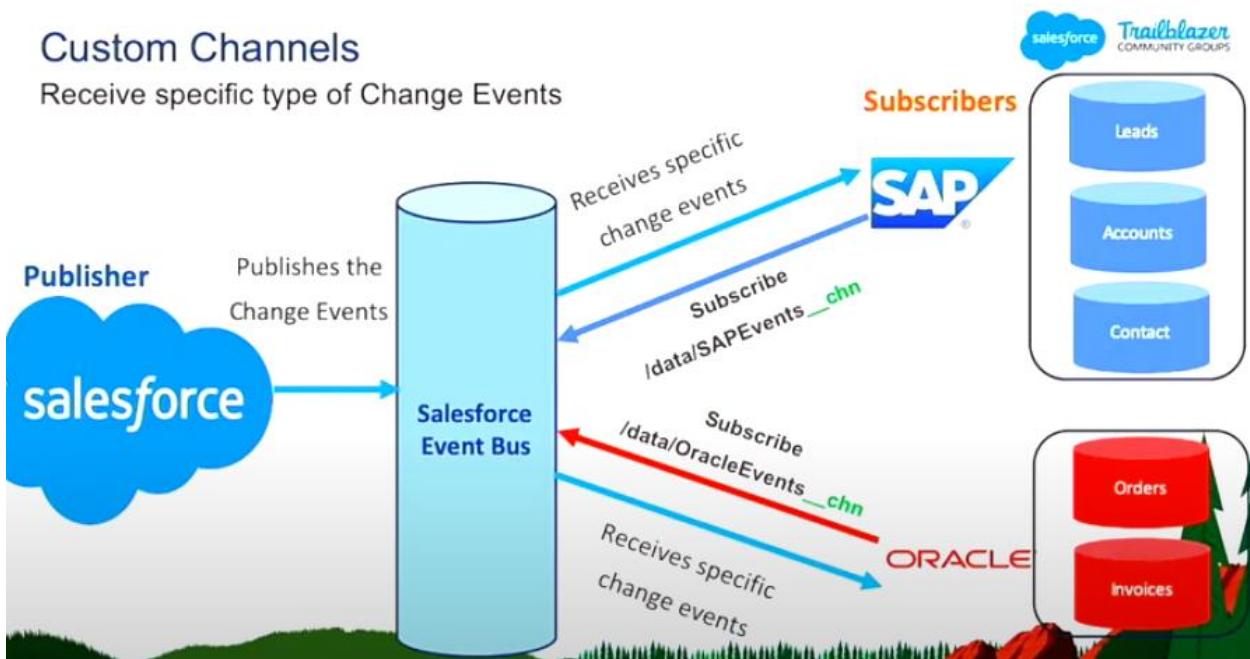


Change Event Subscription Channels

Change Events for	Channel	Example
All Objects	/data/ChangeEvents	N/A
A Standard Object	/data/<Standard_Object_Name>ChangeEvent	For accounts, the channel is: /data/AccountChangeEvent
A Custom Object	/data/<Custom_Object_Name>__ChangeEvent	For Employee__c records, the channel is: /data/Employee__ChangeEvent
Custom Channel	/data/YourChannelName__chn	/data/SAPEvents__chn /data/OracleEvents__chn

Custom Channels

Receive specific type of Change Events



Subscribe the CDC into a different system:-

1. We can assume the other system will behave as like EMP Connector..
2. Get the details of EMP Connector from below link.
<https://github.com/forcedotcom/EMP-Connector>.
3. Connect the EMP Connector along with your Salesforce Org by using the below inside MAVEN Box.
`java -jar target/emp-connector-0.0.1-SNAPSHOT-phat.jar <username> <password> /data/Employee__ChangeEvent.`
4. CDC will take care of Create,Update,Delete and Undelete operations.
5. For a std object the channel name will be `/data/<Standard_Object_Name>ChangeEvent`.
6. For a customer object the channel name will be `/data/<Custom_Object_Name>__ChangeEvent`.
7. Change Data Capture respects your org's field-level security settings. Delivered events contain only the fields that a subscribed user is allowed to view.
8. Change Data Capture ignores sharing settings and sends change events for all records of a Salesforce object.
9. Create a custom channel if you have multiple subscribers and each subscriber receives change events from a different set of entities.
10. For example, if your channel name is SalesEvents, the subscription channel is: `SalesEvents_chn`.

Subscribe to Change Events Using an Apex Trigger:-

- Write an Apex trigger to subscribe to change events.
- Generate change event notifications by making updates in Salesforce.
- Verify debug log messages from the trigger in the debug log.
- Important to remember:
 - Employee__ChangeEvent event
 - EventBus.ChangeEventHeader
 - for (String field : header.changedFields)
 - if (event.First_Name__c != null)
- Test Class Important things to remember:
 - @isTest static void **testChangeEventTrigger()** { // Enable all Change Data Capture entities for notifications. Test.enableChangeDataCapture(); // Insert one or more test records // ... // Deliver test change events Test.getEventBus().deliver(); // Verify the change event trigger's execution // ... }

GAP and OverFlow:-

Gap Events

Change Data Capture tracks record operations that are part of transactions in Salesforce application servers by generating change events. However, some operations are applied directly in the database, outside of a transaction in the Salesforce application servers. For example, archiving of activities or a data cleanup job in the database. To not miss these operations, gap events are generated to notify you about those changes.

A gap event contains information about the change in the header, such as the change type and record ID, but doesn't include details about the change, such as record fields. Valid change type values are:

GAP_CREATE
GAP_UPDATE
GAP_DELETE
GAP_UNDELETE

OverFlow: -This is for more than 100000 records.

Advantages of CDC



- No Custom change-capture coding required.
- Capture granular data changes.
- Real-time data driven integration pattern.

No Polling
No Batch ETL Processes
No point-to-point integration



Apex Triggers

Customizable business logic written natively on Platform



Transactional Execution

Trigger logic is executed and is guaranteed to be complete before the data is committed

Seamless Interaction Experience

End-users get instant feedback for input errors, logic decisions, and computation results

Tightly Coupled with DML Operations

All processing, triggers, updates are considered one unit of work that completely succeeds or rolls back

```
trigger CaseTrigger on Case (after insert) {  
    // Setup  
    ...  
    // Perform logic  
    ...  
    // Cleanup / Finish  
    ...  
}
```

Async Apex Triggers

Complex computations and logic executed outside of the transaction



True Post-Commit Execution

Triggers execute asynchronously after a transaction

Faster End-user Experience

Less logic and computation inside the transaction leads to quicker end-user experiences and fewer limit concerns

Event-driven and Decoupled

Errors in processing won't rollback or impact the transaction

```
trigger CaseChangeEventTrigger on CaseChangeEvent (after insert) {  
    // Setup  
    ...  
    // Perform heavy (slow) computation  
    ...  
    // Cleanup / Finish  
    ...  
}
```

When should I use Async Triggers?



Any calculation or piece of logic that can happen after the data is changed

Keep critical logic in the transaction

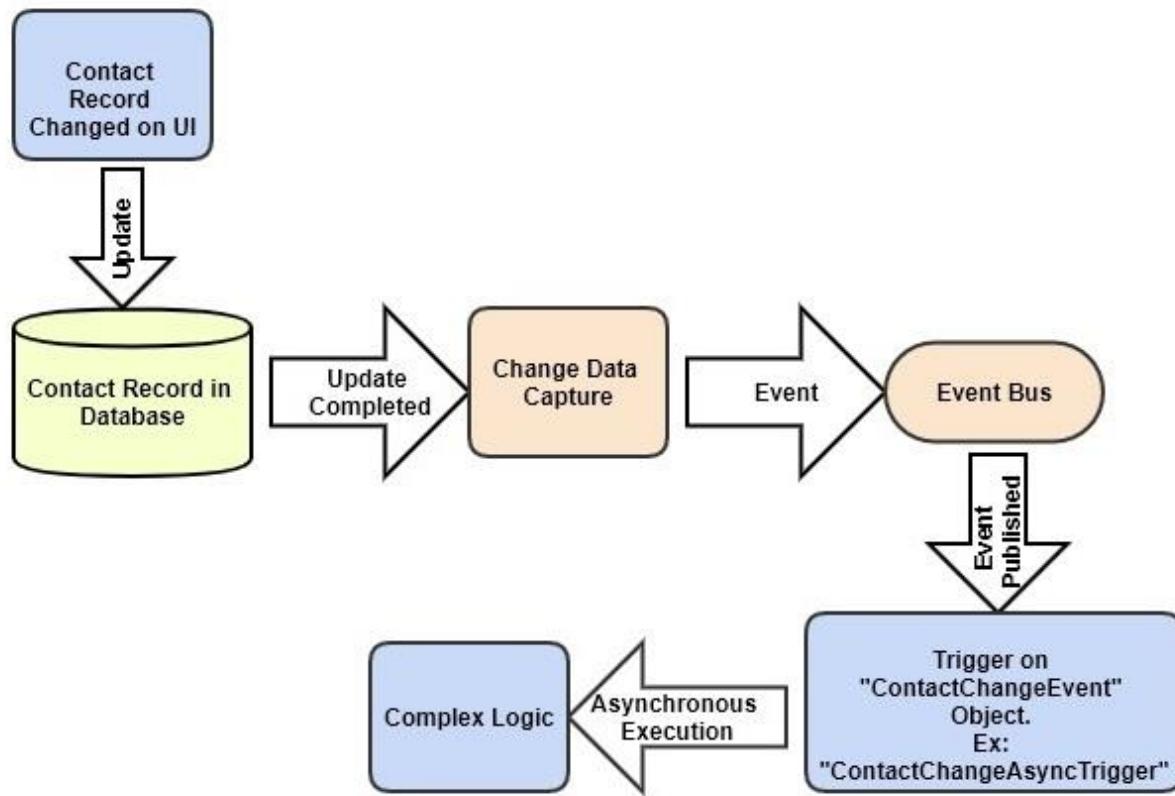
Data validation, data consistency, and required prerequisite operations before data is changed

Processing that can happen “later”

- Heavyweight computations
- Einstein (or other ML/AI) operations
- Notifications to other systems
- Joining of data from different sources
- Summary calculations

```
trigger CaseChangeEventTrigger on CaseChangeEvent (after insert) {  
    List<CaseChangeEvent> changes = Trigger.new;  
    Set<String> caseIds = new Set<String>();  
  
    // Get all Record Ids for this change  
    for (CaseChangeEvent change : changes) {  
        for (String recordId : change.ChangeEventHeader.getRecordIds()) {  
            caseIds.addAll(recordIds);  
        }  
    }  
  
    // Perform heavy (slow) computation determining Red Account  
    RedAccountPredictor predictor = new RedAccountPredictor();  
    Map<String, boolean> accountsToRedAccountStatus =  
        predictor.predictForCases(new List<String>(caseIds));  
  
    // Publish platform events for predicted red accounts  
    List<RedAccount> redAccountEvents = new List<RedAccount__e>();  
    for (String accId : accountsToRedAccountStatus.keySet()) {  
        if (rating=='Red') {  
            redAccountEvents.add(new RedAccount__e(Account__Id__c=accId, Rating__c=rating));  
        }  
    }  
  
    if (redAccountEvents.size() > 0) {  
        EventBus.publish(redAccountEvents);  
    }  
}
```

Example:



Difference between using Asynchronous Triggers and using Queueable or Future apex in triggers.

Async Triggers	Future/Queueable
Executes only when a database transaction is committed in the database.	May Execute in continuation even without the record commit if the queue is empty. In case the record fails in the parent trigger transaction , it will throw an exception here.
Consistent approach of handling complex transactions	Just another way of handling complex transaction. Before the launch of Async triggers it was the only way.
Use context variables such as Trigger.new	Context variables can not be used.
To use this we need to enable Change Data Capture for objects in the setup.	Doesn't require object specific enabling.
Doesn't need trigger new map / Old map to find out the value change in fields.	Need to worry about checking trigger context variables, comparing values,creating lists, passing them etc.
Doesn't have higher set of governor limits.	Has higher set of governor limits
Now the native and streamlined way of decoupling complex logic.	Old way of decoupling complex business logic.