SFDC Architect Topics

Mainly asked by Cognizant people in May 2021.   
Updated in January 2024 with additional content.

1. **Custom Metadata Types Usage**

<https://help.salesforce.com/articleView?id=sf.custommetadatatypes_about.htm>

<https://developer.salesforce.com/docs/atlas.en-us.apexref.meta/apexref/apex_methods_system_custom_metadata_types.htm>

1. **Platform Events**

<https://developer.salesforce.com/docs/atlas.en-us.platform_events.meta/platform_events/platform_events_intro.htm>

High-volume platform event messages are stored for 72 hours (3 days). Standard-volume platform event messages are stored for 24 hours (1 day). You can retrieve past event messages when using CometD clients to subscribe to a channel.

1. **Capture Data Events / Asynch Apex Triggers**  
   Change data events and asynch apex triggers work mainly with platform events.  
     
   Use Change Data Capture to update data in an external system instead of doing periodic exports and imports of data or repeated API calls.  
   <https://developer.salesforce.com/blogs/2018/08/what-is-change-data-capture.html>  
     
   [Asynchronous Apex Triggers in Salesforce (sfdcblogs.com)](https://www.sfdcblogs.com/post/asynchronous-apex-triggers-in-apex)

Asynchronous Apex triggers Aka Change Event Triggers.

With these change event trigger you can now work on change event messages. These triggers run asynchronously after the database transaction is completed.

1. **Customer 360 Data Manager Basics**  
   Integrate data among multiple Service Cloud orgs and Commerce Cloud. To help you connect case history, order history, and customer data across your enterprise, Customer 360 Data Manager assigns a global profile to each unique customer. Having a global profile lets service agents, for example, view a customer’s order history in Service Console without having to swivel to Commerce Cloud.
2. **Authorization Flows**

<https://help.salesforce.com/articleView?id=sf.remoteaccess_oauth_flows.htm&type=5>

1. **OAuth Basics**

Types of different flows.

How SAML works.

Describe Single Sign On.

How OAuth works.

Difference between SAML and OAuth.

1. **Bulk API limits for humongous data loads.**

<https://developer.salesforce.com/docs/atlas.en-us.salesforce_app_limits_cheatsheet.meta/salesforce_app_limits_cheatsheet/salesforce_app_limits_platform_bulkapi.htm>

How to import 150 million records.

Data Loader supports upto 5 million.

Explore App Exchange to find an appropriate tool.

- Pentaho

- JitterBit

- Dataloader.io

Bulk API Limit

Maximum number of records uploaded per 24-hour rolling period 150,000,000

(15,000 batches x 10,000 records per batch maximum)

Difference between Bulk and Bulk 2.0  
 In Bulk API 2.0, batches are created for you automatically. In Bulk API, you must create the batches yourself.

<https://developer.salesforce.com/docs/atlas.en-us.api_asynch.meta/api_asynch/bulk_common_diff_two_versions.htm>

1. **Custom Permissions [Different than permission sets]**

<https://help.salesforce.com/articleView?id=sf.custom_perms_overview.htm&type=5>  
  
To determine which users have access to a specific custom permission, use Apex and do something like the following.

Boolean hasCustomPermission = FeatureManagement.checkPermission('your\_custom\_permission\_api\_name');

1. **Sharing Rules and Sharing Architecture**

Limits of Sharing Rules

<https://salesforceben.com/tips-for-planning-and-creating-salesforce-sharing-rules/#:~:text=You%20can%20define%20up%20to,rules%20to%2050%20per%20object>

1. **Attachment object**

<https://support.formtitan.com/topics/Salesforce/Attachment_Object>

ContentDocument

ContentDocumentLink - JUNCTION table between Object(Case etc.) and ContentDocument

1. **Async SOQL Versus SOQL**

<https://developer.salesforce.com/docs/atlas.en-us.bigobjects.meta/bigobjects/async_query_overview.htm>

1. **Real-Time Event Monitoring, gain greater insights into:**

Who viewed what data and when

Where data was accessed

When a user changes a record using the UI

Who is logging in and from where

Who in your org is performing actions related to Platform Encryption administration

Which admins logged in as another user and the actions the admin took as that user

How long it takes a Lightning page to load

Threats detected in your org, such as anomalies in how users view or export reports, session hijacking attacks, or credential stuffing attacks

1. **Big Objects**

<https://developer.salesforce.com/docs/atlas.en-us.bigobjects.meta/bigobjects/big_object.htm>

Two types of big objects.

Custom Big Object Use Cases

Differences Between Big Objects and Other Objects

1. **Asynchronous**

[Asynchronous Apex | Apex Developer Guide | Salesforce Developers](https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_async_overview.htm)

[Transaction Finalizers | Apex Developer Guide | Salesforce Developers](https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_transaction_finalizers.htm)

**Integration patterns**

| **Pattern** | **Scenario** |
| --- | --- |
| [Remote Process Invocation —Request and Reply](https://developer.salesforce.com/docs/atlas.en-us.integration_patterns_and_practices.meta/integration_patterns_and_practices/integ_pat_remote_process_invocation_state.htm) | Salesforce invokes a process on a remote system, waits for completion of that process, and then tracks state based on the response from the remote system. |
| [Remote Process Invocation —Fire and Forget](https://developer.salesforce.com/docs/atlas.en-us.integration_patterns_and_practices.meta/integration_patterns_and_practices/integ_pat_remote_process_invocation_fire_forget.htm) | Salesforce invokes a process in a remote system but doesn’t wait for completion of the process. Instead, the remote process receives and acknowledges the request and then hands off control back to Salesforce. |
| [Batch Data Synchronization](https://developer.salesforce.com/docs/atlas.en-us.integration_patterns_and_practices.meta/integration_patterns_and_practices/integ_pat_batch_data_sync.htm) | Data stored in Lightning Platform is created or refreshed to reflect updates from an external system, and when changes from Lightning Platform are sent to an external system. Updates in either direction are done in a batch manner. |
| [Remote Call-In](https://developer.salesforce.com/docs/atlas.en-us.integration_patterns_and_practices.meta/integration_patterns_and_practices/integ_pat_remote_call_in.htm) | Data stored in Lightning Platform is created, retrieved, updated, or deleted by a remote system. |
| [UI Update Based on Data Changes](https://developer.salesforce.com/docs/atlas.en-us.integration_patterns_and_practices.meta/integration_patterns_and_practices/integ_pat_ui_updates_from_data_changes.htm) | The Salesforce user interface must be automatically updated as a result of changes to Salesforce data. |
| [Data Virtualization](https://developer.salesforce.com/docs/atlas.en-us.integration_patterns_and_practices.meta/integration_patterns_and_practices/integ_pat_data_virtualization.htm) | Salesforce accesses external data in real time. This removes the need to persist data in Salesforce and then reconcile the data between Salesforce and the external system. |

**Architectural Patterns**

* [The API gateway pattern versus the direct client-to-microservice communication - .NET | Microsoft Learn](https://learn.microsoft.com/en-us/dotnet/architecture/microservices/architect-microservice-container-applications/direct-client-to-microservice-communication-versus-the-api-gateway-pattern)

This pattern is a service that provides a single-entry point for certain groups of microservices. It's similar to the [Facade pattern](https://en.wikipedia.org/wiki/Facade_pattern) from object-oriented design, but in this case, it's part of a distributed system. The API Gateway pattern is also sometimes known as the "backend for frontend" ([BFF](https://samnewman.io/patterns/architectural/bff/)) because you build it while thinking about the needs of the client app.

Therefore, the API gateway sits between the client apps and the microservices. It acts as a reverse proxy, routing requests from clients to services

* [14 software architecture design patterns to know](https://www.redhat.com/architect/14-software-architecture-patterns)
* [Microsoft MSDN Patterns](https://learn.microsoft.com/en-us/azure/architecture/patterns/)

**Architectural Questions**

* <https://dineshyadav.com/salesforce-architect-interview-questions/>

**Developer Questions**

<https://www.edureka.co/blog/interview-questions/salesforce-interview-questions/>

* Skinny Table  
  In Salesforce, skinny tables are used to access frequently used fields and to avoid joins. This largely improves performance.
* Apex Best Practices

<https://www.salesforceben.com/12-salesforce-apex-best-practices/>