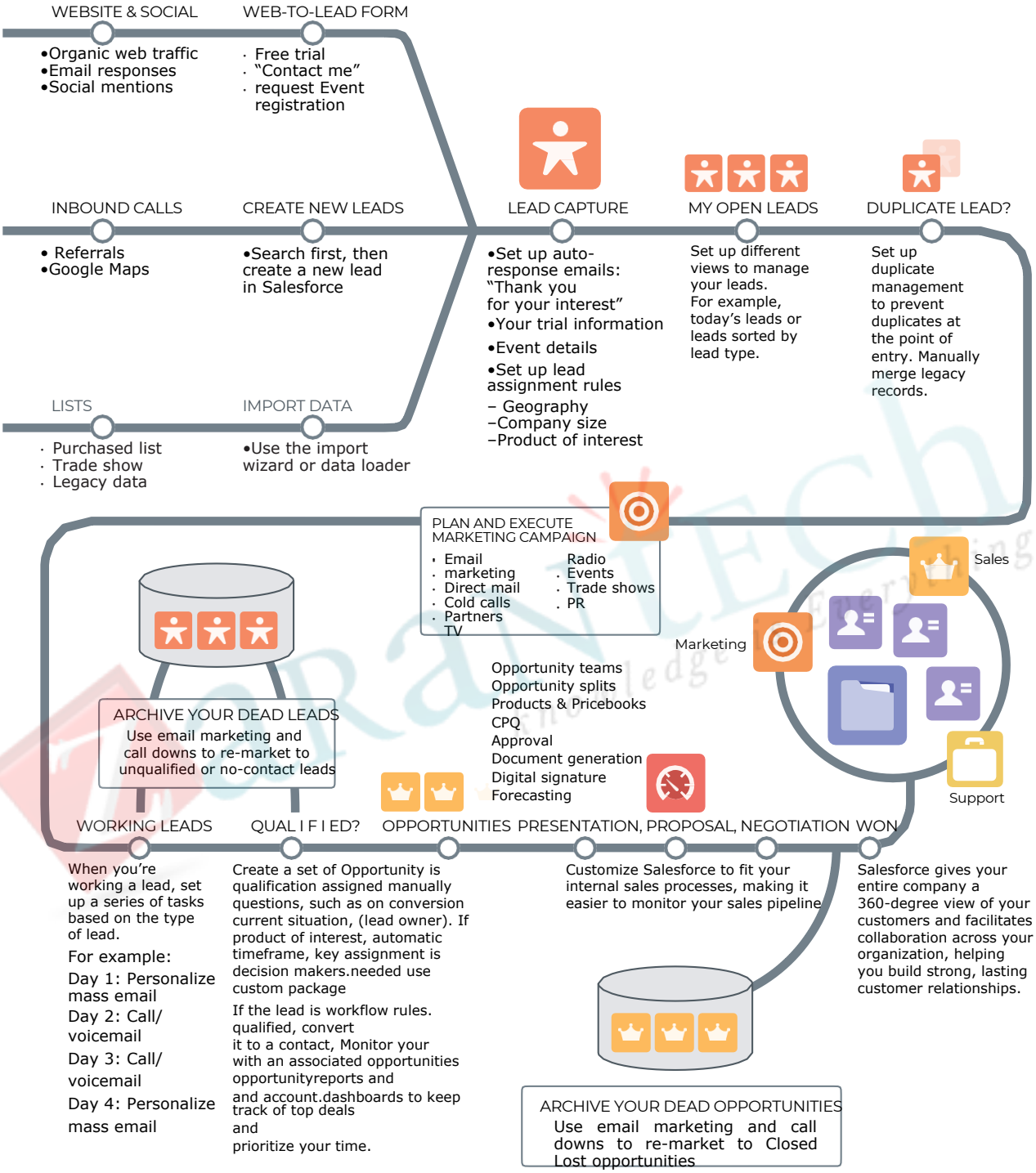
ZaranTech
knowledge is Everything

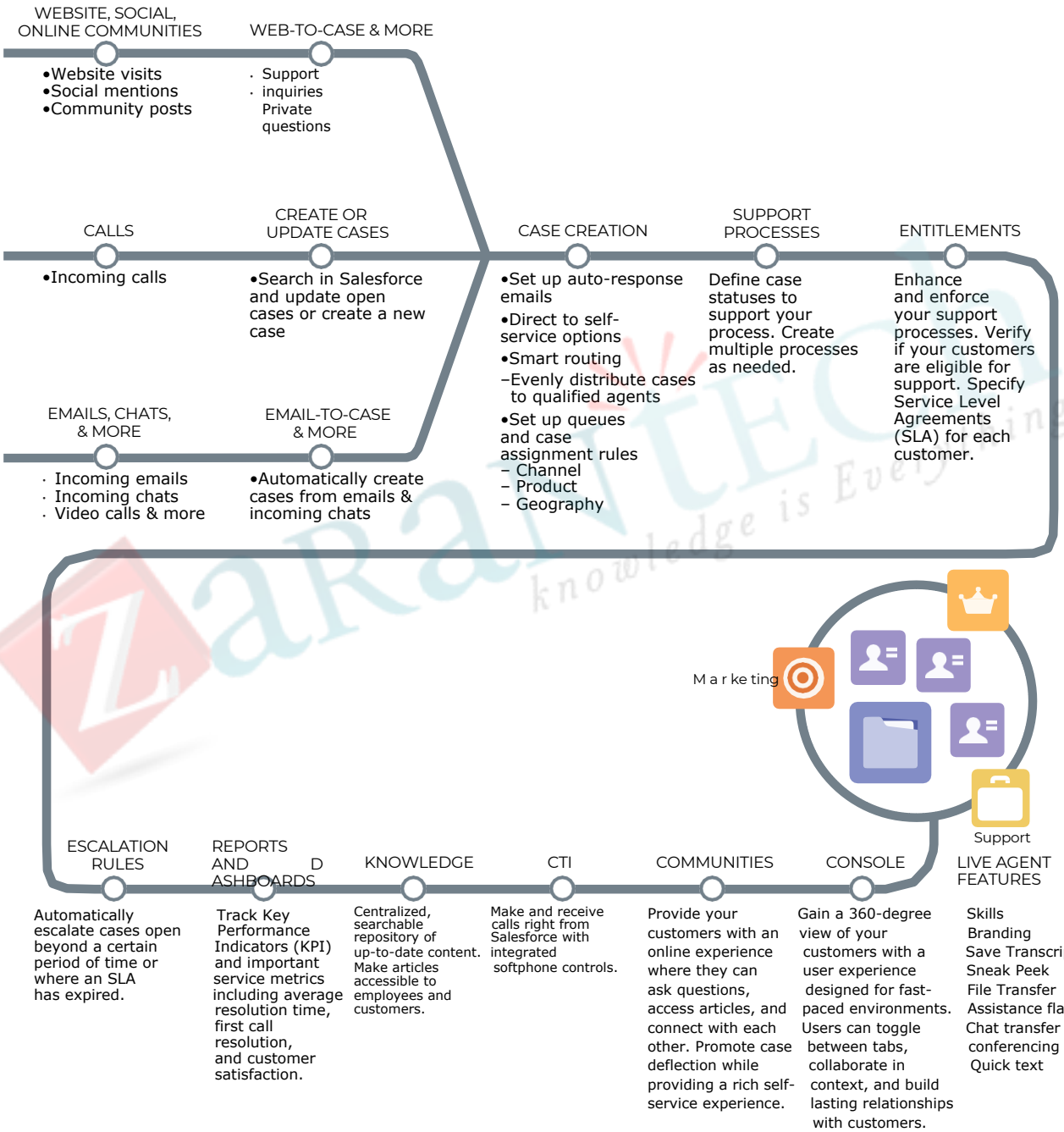
SALESFORCE ULTIMATE CHEATSHEET

Phone/Whatsapp: +1 (515) 309-7846 (USA)
Email: info@zarantech.com
Website: www.zarantech.com

CRM Cheatsheet



Service Cloud Cheatsheet



Forecasting

Forecasts can be used to predict future sales within an organization. Forecast data is aggregated from each user's opportunity records, and the related forecast category of each opportunity's stage.

The advantage of Salesforce.com Forecasting is that it's relatively easy for a sales rep to maintain an accurate representation of an opportunity's status. The software adds up the numbers and produces a forecast based on the close dates of opportunities. Further, Salesforce.com populates pipeline dashboards and allows management to visualize, analyze, optimize and prioritize.

The disadvantage of using pipeline reports as a forecasting tool is that inconsistent and inaccurate opportunity updates by sales reps can severely affect data integrity. Adding to the problem are mutated methodologies and a rep's desire to maintain the 3:1 pipeline ratio requested by management.

Customizable forecasting	Collaborative forecasting
Used for customizing the forecast for the needs of your business. This kind of forecasting is used for custom fiscal year, opportunity lead adjustments, territory management, snapshots and forecast history •Product Family Forecasts •Opportunity Products Schedule Forecasts •Forecast Sharing •Forecast Snapshots & History •Opportunity-level Adjustments •Opportunity Product Level Adjustments •Ability to Override Forecast Category •Ability for Sales Reps to adjust their Forecast	Provides your business with more flexibility and more intuitive user interface. The unique features of this kind of forecasting include ability to rename the forecast categories, expandable forecast tables and forecasting on the opportunity splits •Forecast revenue and/or quantity based on opportunity data •Forecast viewable in monthly or quarterly rollups •Customizable forecast categories •Mutually exclusive forecast categories •Forecast numbers and related Opportunities on the same page •Manager ability to adjust forecasts •Revenue or quantity quotas for sales teams (via API only) •Ability to create Custom Report Types & Custom Reports on Forecast, Opportunity & Quota data •API support •Interactive and easy user experience

Territory Management

Territory management is an account sharing system that grants access to accounts based on the characteristics of the accounts. It enables your company to structure your Salesforce data and users the same way you structure your sales territories.
Particularly if your organization has a private sharing model, you may need to grant users access to accounts based on criteria such as postal code, industry, revenue, or a custom field that is relevant to your business. You may also need to generate forecasts for these diverse categories of accounts. Territory management solves these business needs and provides a powerful solution for structuring your users, accounts, and their associated contacts, opportunities, and cases. Accounts and users can belong to multiple territory, while an opportunity can belong to only one territory.

Account ownership and its effect on record sharing remains valid and unchanged when territory management is in use.

Original Territory Management is available only with Customizable Forecasts and is not supported with Collaborative Forecasts. If original Territory Management is enabled, you can no longer use Collaborative Forecasts. The newer Enterprise Territory Management can't be used with Customizable Forecasts. Enterprise Territory Management and Collaborative Forecasts can both be enabled and used at the same time in your Salesforce organization, but the two features are not currently integrated to work together.

A territory model represents a complete territory management system for your organization. Modeling lets you create and preview multiple territory structures and different account and user assignments before you activate the model that works best.
Both Territory management sharing and role hierarchy sharing can be active at the same time. Users will receive whatever access is most permissive across both hierarchies.

Salesforce Licenses

Identity Licenses

Knowledge Only User

Designed for users who only need access to the Salesforce Knowledge app. This license provides access to custom objects, custom tabs, and the following standard tabs.

- Articles
- Article Management
- Chatter
- Files
- Home
- Profile
- Reports
- Custom objects
- Custom tabs

The Knowledge Only User license includes a Knowledge Only profile that grants access to the Articles tab. To view and use the Article Management tab, a user must have the "Manage Articles" permission.

Identity

Grants users access to Salesforce Identity features.Salesforce Identity connects Salesforce users with external applications and services, while giving administrators control over authentication and authorization for these users.

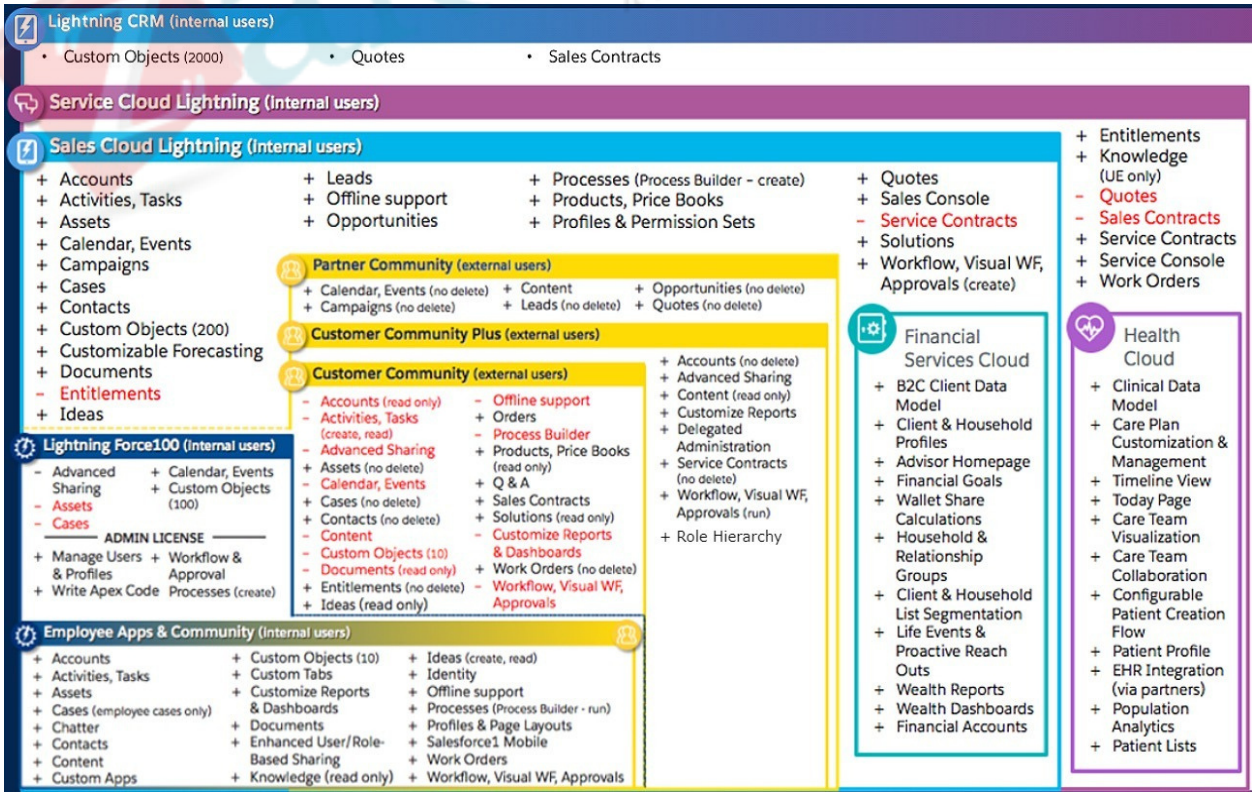
External Identity

Provides Identity features for users outside of your organization's user base (such as non-employees). Store and manage these users, choose how they authenticate (username/password, or Single Sign-On social sign-on through Facebook, Google+, LinkedIn, and others), and allow self-registration.

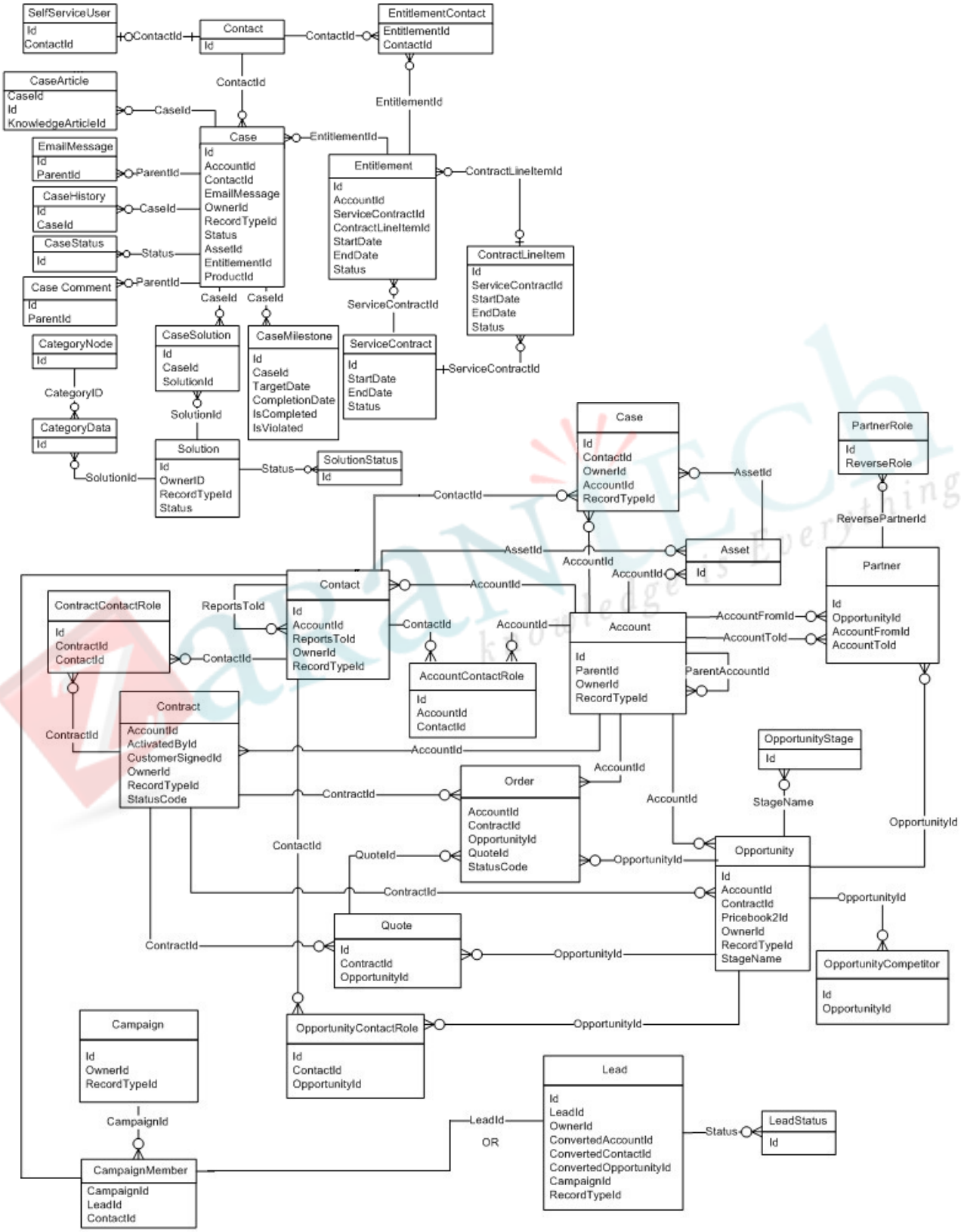
Chatter Licenses

Feature	Chatter External (Access limited to items and people in the groups customers are invited to)	Chatter Free	Chatter Only (a.k.a. Chatter Plus)	Employee Apps Starter
Chatter Desktop client	✓	✓	✓	✓
Use the Salesforce app (Downloadable apps require the "API Enabled" profile permission)	✓ Downloadable app users can't access Groups or People list views.	✓	✓	✓
File sharing	✓	✓	✓	✓
Files Connect			✓	✓
Invitations to join groups	✓ Only customers who are also group managers can invite Chatter users from groups they have access to or people outside Chatter.	✓	✓	✓
Topics and hash tags		✓	✓	✓
Global search	✓ Search results include only those items that customers have access to via groups.	✓	✓ Chatter only users have access to reports and dashboards but cannot use global search to find them.	✓
Custom objects			✓ Up to 10 custom objects	✓
Accounts and contacts			✓ Read only	✓
Calendar and events			✓	✓
Content library			✓	✓
Ideas and answers			✓	✓
Reports and dashboards			✓	✓ (access to dashboards is read-only)
Tasks and activities			✓	✓
Using and approving workflows			✓	✓

Other Licenses



Salesforce Data Model



Encryption methods

CLASSIC ENCRYPTED FIELDS

•Only Text (Encrypted) field type is supported
•Only users with the permission "View Encrypted Data" can see dataand clone th evalue
•Cannot be unique, have an external ID, or have default values.
•For leads are not available for mapping to other objects.
•Are limited to 175 characters because of the encryption algorithm.
•Are not available for: Salesforce Mobile Classic, Connect Offline, SalesforceforOutlook, lead conversion, workflow rule criteria or formulas, formula fields, outbound messages, default values, and Web-to-Lead and Web-to-Case forms.
•Existing custom fields cannot be converted into encrypted fields nor can encrypted fields be converted into another data type.

SHIELD PLATFORM ENCRYPTION

•Gives your data a whole new layer of security while preserving critical platform functionality.
•Data stored in many standard and custom fields and in files and attachments is encrypted using an advanced HSM-based key derivation system
•You can reference encrypted fields in most places in your flows and processes.
•You can encrypt certain fields on the Account, Contact, Case, and Case Comment objects.
•You can encrypt custom fields of type email. Phone, url , datetime and all text type fields
•Only users with the "View Encrypted Data" permission can see thecontents of encrypted fields
•Not supported by many SF Apps (Heroku, SF CPQ, Data.com, Marketing Cloud)

Feature	Classic Encryption	Platform Encryption
Pricing	Included in base user license	Additional fee applies
Encryption at Rest	✔	✔
Native Solution (No Hardware or Software Required)	✔	✔
Encryption Algorithm	128-bit Advanced Encryption Standard (AES)	256-bit Advanced Encryption Standard (AES)
HSM-based Key Derivation		✔
Manage Encryption Keys Permission		✔
Generate, Export, Import, and Destroy Keys	✔	✔
PCI-DSS L1 Compliance	✔	✔
Masking	✔	
Mask Types and Characters	✔	
View Encrypted Data Permission Required to Read Encrypted Field Values	✔	
Encrypted Standard Fields		✔
Encrypted Attachments, Files, and Content		✔
Encrypted Custom Fields	Dedicated custom field type, limited to 175 characters	✔
Encrypt Existing Fields for Supported Custom Field Types		✔
Search (UI, Partial Search, Lookups, Certain SOSL Queries)		✔
API Access	✔	✔
Available In Workflow Rules and Workflow Field Updates		✔
Available In Approval Process Entry Criteria and Approval Step Criteria		✔

Object relationship types

Master-detail
•Closely links objects together such that the master record controls certain behaviors of the detail and subdetail record.
•Detail and subdetail records inherit security settings and permissions from the master record. You can't set permissions on the detail record independently.
•The Owner field on the detail and subdetail records is not available and is automatically set to the owner of the master record.Custom objects on the "detail" side of a master-detail relationship can't have sharing rules, manual sharing, or queues, as these require the Owner field.
•The master-detail relationship field (which is the field linking the objects) is required on the page layout of the detail and subdetail records.
•Deleting a detail record moves it to the Recycle Bin and leaves the master record intact; deleting a master record also deletes related detail and subdetail records. Undeleting a detail record restores it, and undeleting a master record also undeletes related detail and subdetail records. However, if you delete a detail record and later, separately, delete its master record, you cannot undelete the detail record, as it no longer has a master record to relate to.

Lookup
•Links two objects together. Lookup relationships are similar to master-detail relationships, except they do not support sharing or roll-up summary fields.
•Can be required, prevent deletion and cascade delete if setup

External lookup
•An external lookup relationship links a child standard, custom, or external object to a parent external object.
•The standard External ID field on the parent external object is matched against the values of the child's external lookup relationship field. External object field values come from an external data source.

Indirect lookup
•Links a child external object to a parent standard or custom object.
•When you create an indirect lookup relationship field on an external object, you specify the parent object field and the child object field to match and associate records in the relationship. Specifically, you select a custom unique, external ID field on the parent object to match against the child's indirect lookup relationship field, whose values come from an external data source.

Hierarchical
•A special lookup relationship available for only the user object. It lets users use a lookup field to associate one user with another that does not directly or indirectly refer to itself. For example, you can create a custom hierarchical relationship field to store each user's direct manager.

Many to many
•Created by creating a junction object with 2 master-detail or lookup relationships

Different ways to manage files & content

	Files Home	Salesforce CRM Content	Salesforce Knowledge	Documents Tab	Attachments
Purpose	Upload, store, find, follow, share, sync, and collaborate on Salesforce files in the cloud.	Publish and share official corporate files with coworkers and deliver them to customers.	Create and manage content, known as articles, in a knowledge base. Internal users and customers (on your Customer Portal, partner portal, Service Cloud Portal, or Lightning Platform Sites) can quickly find and view articles they need.	Store Web resources, such as, logos, DOT files, and other Visualforce materials in folders without attaching them to records.	Attach files to records from the Attachments related list on selected detail pages.
Common Uses	Upload a file and store it privately until you're ready to share it. Share the file with coworkers and groups to collaborate and get feedback. Attach files to posts in a Chatter feed on the Home tab, Chatter tab, a profile, a record, or a group.	Create, clone, or modify a sales presentation and save it so only you can see it and work on it. When you're ready, publish it so other users in your company have access to it. Create a content pack and send it to customers.	Write, edit, publish, and archive articles using the Articles Management tab or find and view published articles using the Articles tab. Customers and partners can access articles if Salesforce Knowledge is enabled in your Customer Portal, partner portal, Service Cloud Portal, or Lightning Platform Sites.	Add a custom logo to meeting requests by uploading your logo to the Documents tab.	Add a file to a specific record, like an event, marketing campaign, contact, or case by attaching it on the Attachments related list.
Supported File Types	All	All	All	All	All
Maximum File Sizes	2 GB	▪ 2 GB ▪ 2 GB (including headers) when uploaded via Chatter REST API ▪ 2 GB (including headers) when uploaded via REST API ▪ 38 MB when uploaded via SOAP API ▪ 10 MB when uploaded via BULK API ▪ 10 MB for Google Docs ▪ 10 MB when uploaded via Visualforce	5 MB for attachments	▪ 5 MB ▪ 20 KB for a custom-app logo	▪ 25 MB for file attachments ▪ 2 GB for feed attachments

Apex transaction Limits

Description	Synchronous Limit	Asynchronous Limit
Total number of SOQL queries issued[1]	100	200
Total number of records retrieved by SOQL queries	50,000	
Total number of records retrieved by `Database.getQueryLocator`	10,000	
Total number of SOSL queries issued	20	
Total number of records retrieved by a single SOSL query	2,000	
Total number of DML statements issued[2]	150	
Total number of records processed as a result of DML statements, `Approval.process`, or `database.emptyRecycleBin`	10,000	
Total stack depth for any Apex invocation that recursively fires triggers due to `insert`, `update`, or `delete` statements[3]	16	
Total number of callouts (HTTP requests or Web services calls) in a transaction	100	
Maximum cumulative timeout for all callouts (HTTP requests or Web services calls) in a transaction	120 seconds	
Maximum number of methods with the `future` annotation allowed per Apex invocation	50	
Maximum number of Apex jobs added to the queue with `System.enqueueJob`	50	
Total number of `sendEmail` methods allowed	10	
Total heap size[4]	6 MB	12 MB
Maximum CPU time on the Salesforce servers[5]	10,000 milliseconds	60,000 milliseconds
Maximum execution time for each Apex transaction	10 minutes	
Maximum number of push notification method calls allowed per Apex transaction	10	
Maximum number of push notifications that can be sent in each push notification method call	2,000	

Organization strategy

An org strategy is a plan of how to best use Salesforce with your business. It outlines the underlying org architecture that will be used in your Salesforce solution. Two possible approaches:

- Single-Org
- Multi-Org

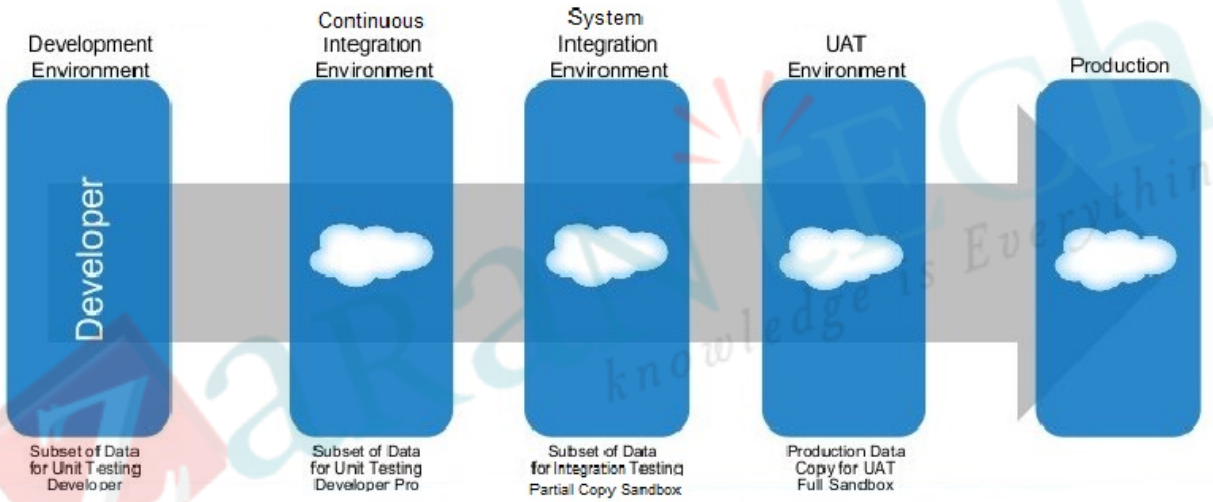
Approach	Pros	Cons
Single-Org	Cross business unit collaboration Salesforce Chatter shared in the organization Aligned processes, reports, dashboards, security – consolidated customization Ability to share data Unified reporting Single login to access multiple business functions 360 view from a central point of view – overall reports possible Interfaces are easier to maintain	Org complexity could become a barrier to progress Potential to hit specific Org limits, such as number of custom tabs, objects and code lines Org-wide settings could become difficult to govern and manage Time to market and innovate could be impacted by number of teams rolling out new functionality More teams updating shared configuration and code means more regression testing is needed as complexity increases over time Fewer sandbox environments reduces testing capabilities Local administration is difficult
Multi-Org	Logical Separation of data Reduced risk of exceeding Org limits Org-wide settings are easier to be governed and managed. Lower data volumes within a single Org – potentially improves performance Improved time to market and freedom to innovate Fewer teams impacted by shared updates Reduced complexity within a single Org More sandbox environments means more testing capabilities Local administration and customization possible	Harder to get a clear global definition of processes and data Less reuse of configuration and code Solutions for shared common business requirements need to be deployed into multiple Orgs Inferior collaboration across business units (no shared Chatter) Duplicated administration functions required Increased complexity for single sign on Merging/Splitting Orgs and changing integration endpoints is very difficult. The administration is extensive for configurations which cannot be deployed by automated processes. (deployment strategy needed)

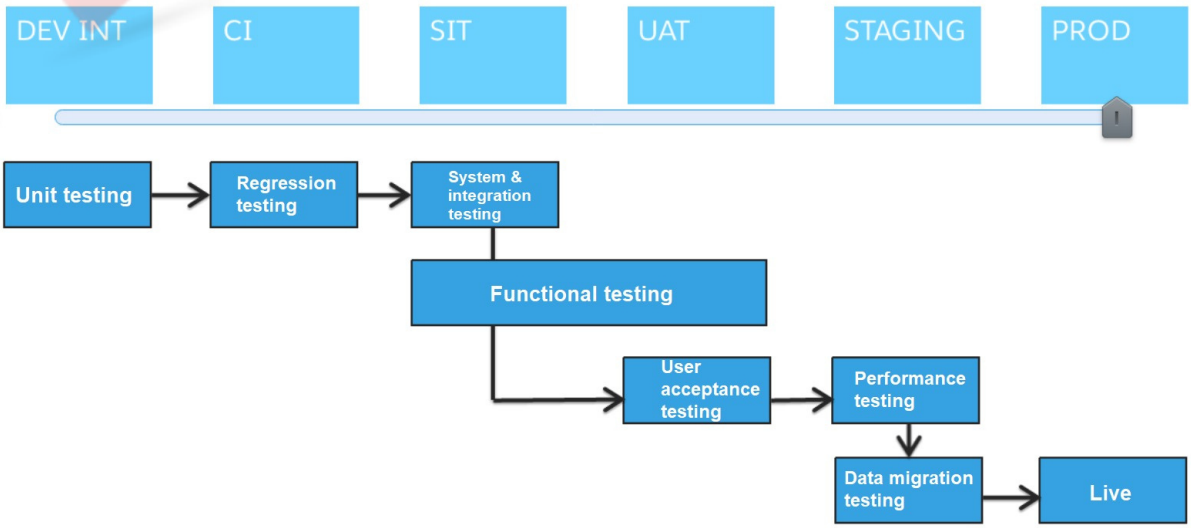
The multi-org strategy can be declined in different approaches:

1. *Complete Autonomy*
   - each organization is not directly linked to each other environment
2. *Master Child*
   - a master org pushes a subset of data to all child linking organizations
3. *No centralized org*
   - each organization is directly linked to other organizations using Salesforce2Salesforce integration for exchanging and updating data

Sandbox strategy

Sandbox Type	Refresh Interval	Storage Limit	What's Copied	Sandbox Templates
Developer Sandbox	1 day	Data storage: 200 MB File storage: 200 MB	Metadata only	Not available
Developer Pro Sandbox	1 day	Data storage: 1 GB File storage: 1 GB	Metadata only	Not available
Partial Copy Sandbox	5 days	Data storage: 5 GB File storage: 5 GB	Metadata and sample data	Required
Full Sandbox	29 days	Same as your production org	Metadata and all data	Available





DATABASE

Query & Search Optimization Cheat Sheet

Overview

When building queries, list views, and reports, it's best to create filter conditions that are selective so Force.com scans only the rows necessary in the objects your queries target—the Force.com query optimizer doesn't use an index to drive queries containing unselective filter conditions, even if the fields those filter conditions reference already have indexes on them. (This cheat sheet's "Index Selectivity Exceptions" section points to several things that automatically make filter conditions unselective.) Because filter conditions are also unselective if they exceed the Force.com query optimizer's thresholds, selectivity is especially important when your queries target objects containing more than one million records. Read on to learn how to write selective filter conditions, minimize your query response times, and optimize the database's overall performance.

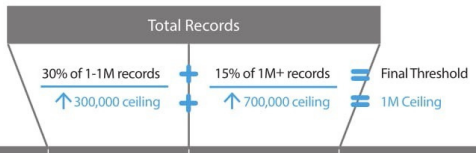
Fields with Database Indexes

Indexed Standard Fields, All Objects
`Id`
`Name`
`OwnerId`
`CreatedDate`
`SystemModstamp`
`RecordType` (indexed for all standard objects that feature it)
Master-detail fields
Lookup fields
Other Indexed Fields
Unique fields
External ID fields

Index Selectivity Conditions and Thresholds

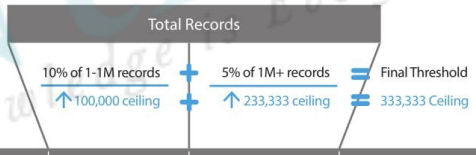
Condition	Thresholds	Index Used
Unary: standard index	Filter targets less than: • 30% of the first million records • 15% of records after the first million records • 1 million total records	Standard index
Unary: custom index	Filter targets less than: • 10% of the first million records • 5% of records after the first million records • 333,333 total records	Custom index
`AND`	Filter targets less than: • Twice the index selectivity thresholds for each filter • The index selectivity thresholds for the intersection of the fields The Force.com query optimizer can detect date and number ranges, and treats their filters on the same field as a single, combined filter.	Composite index
`OR`	Filter targets less than: • The index selectivity thresholds for each filter • The index selectivity thresholds for the sum of those fields	Index union
`LIKE`	For conditions that don't start with a leading wildcard, Force.com tests the first 100,000 rows for selectivity.	Standard index or custom index

Force.com Query Optimizer - Standard Index



# of records	First Threshold	Second Threshold	Final Threshold
≤ 1 million	30% of total	n/a	30% of total
≤ 2 million	300,000	150,000	450,000
≤ 3 million	300,000	300,000	600,000
≤ 4 million	300,000	450,000	750,000
≤ 5 million	300,000	600,000	900,000
≥ 5.6 million	300,000	700,000	1 million

Force.com Query Optimizer - Custom Index



# of records	First Threshold	Second Threshold	Final Threshold
≤ 1 million	10% of total	n/a	10% of total
≤ 2 million	100,000	50,000	150,000
≤ 3 million	100,000	100,000	200,000
≤ 4 million	100,000	150,000	250,000
≤ 5 million	100,000	200,000	300,000
≥ 5.6 million	100,000	333,333	333,333

Index Selectivity Exceptions

Filter Conditions With	In SOQL	In Reports and List Views
Negative filter operators	• `!=` • `NOT LIKE` • `EXCLUDES`	• not equal to • does not contain • excludes
Comparison operators paired with text fields	• *`text_field <`* • *`text_field >`* • *`text_field <=`* • *`text_field >=`*	• *text field* less than • *text field* greater than • *text field* less or equal • *text field* greater or equal
Leading "%" wildcards	`LIKE '%string%'`	contains
References to non-deterministic formula fields	Cross-object formula fields	

salesforce force.com

http://developer.salesforce.com

Asynchronous Apex

Apex offers multiple ways for running your Apex code asynchronously. Choose the asynchronous Apex feature that best suits your needs.

This table lists the asynchronous Apex features and when to use each.

Asynchronous Apex Feature	When to Use
Future Methods	• When you have a long-running method and need to prevent delaying an Apex transaction • When you make callouts to external Web services • To segregate DML operations and bypass the mixed save DML error
Queueable Apex	• To start a long-running operation and get an ID for it • To pass complex types to a job • To chain jobs
Batch Apex	• For long-running jobs with large data volumes that need to be performed in batches, such as database maintenance jobs • For jobs that need larger query results than regular transactions allow
Scheduled Apex	• To schedule an Apex class to run on a specific schedule

Apex Unit tests

• At least 75% test coverage is required
• Every trigger must have some test coverage.
• All classes and triggers must compile successfully.
• Classes and methods defined as@isTestcan be eitherprivateorpublic.
• Classes defined as @isTest must be top-level classes and can't be interfaces or enums.
• Methods of a test class can only be called from a running test, that is, a test method or code invoked by a test method, and can't be called by a non-test request.
• Test methods can’t be used to test Web service callouts. Instead, use mock callouts. (Test.setMock + HttpCalloutMock, WebServiceMock interface)
• You can use @testVisible to enable access to private methods
• Annotate your test class or test method with IsTest(SeeAllData=true) to open up data access to records in your organization.
• Use test setup methods (methods that are annotated with @testSetup) to create test records once and then access them in every test method in the test class.
• The runAsmethod enables you to write test methods that change the user context to an existing user or a new user so that the user’s record sharing is enforced. You can also use the runAs method to perform mixed DML operations in your test by enclosing the DML operations within the runAs block.
•You can use Test.startTest and Test.stopTest to manage governor limits consumption

Deployment methods

Tool	Best for	Limitations
Change Sets	• Straight sandbox to production migrations • Change management without using a local file system • Auditing previously deployed changes • Enforcing code migration paths • Deploying the same components to multiple orgs	
Ant Migration Tool	• Development projects for which you need to populate a test environment with a lot of setup changes—Making these changes using a web interface can take a long time. • Multistage release processes—A typical development process requires iterative building, testing, and staging before releasing to a production environment. Scripted retrieval and deployment of components can make this process much more efficient. • Repetitive deployment using the same parameters—You can retrieve all the metadata in your organization, make changes, and deploy a subset of components. If you need to repeat this process, it's as simple as calling the same deployment target again. • When migrating from stage to production is done by IT—Anyone that prefers deploying in a scripting environment will find the Ant Migration Tool a familiar process. • Scheduling batch deployments—You can schedule a deployment for midnight to not disrupt users. Or you can pull down changes to your Developer Edition org every day.	
Force.com IDE	• Project-based development • Deployment to any org • Synchronizing changes • Selecting only the components you need	• Some setup required • Not always upgraded at the same time as other Salesforce products • Repeatable deployments require re selecting components, which can be time consuming and introduce errors
Force.com Workbench	• Ad hoc queries • Deploy or retrieve components with a package.xml • file Metadata describes • Lightweight data loads	• Not an officially supported • product No project management features
Force.com CLI	• Scripted commands and automated tasks • When your security policies dictate that passwords must not be stored on disk; forces interactive login	• Logging in can be difficult behind a firewall
Unmanaged Packages	• One-time setup of a development environment • A starting point configuration that can be customized	• You can't make further changes to packaged components using subsequent pac kag es • Requires a Developer Edition org
Managed Packages	• Commercial applications • Functionality you want to add in multiple, possibly non- related orgs	• Access to code is limited or hidden • Unique namespace can be bothersome or a blocker • Difficult to modify or delete components • Requires a Developer Edition org

Support multiple languages

- An administrator can enable multiple languages in setup. Every language has to be enabled separately
- Standard objects and fields are translated automatically and translations can be edited in „Rename tabs & labels" interface
- Use Translation Workbench to maintain your translated labels in your org. You can manage translated values for any Salesforce supported language.
- Almost everything can be translated including object names, field labels, picklist values, validation messages, etc.
- If a customized component doesn't have a translated value, the component uses the org's default language. When you deactivate a language, all translations for that language are still available in the Translation Workbench, but users with that language selected see the org's default language values.
- Translations can be exported and imported from a file

Multi-Currency

- By default, Salesforce organizations use a single currency. Once you set the required currency locale in your company settings, all currency values on records display in that currency.
- Once you activate multicurrency for your org, you can specify which currencies are supported by activating or deactivating them
- Every record has a Currency field that specifies the currency type for amounts in that record.All currency amounts display in therecord's currency and are also converted to the personal currency of the record owner, based on the conversion rates entered by youradministrator. Amounts in the user's personal currency are displayed in parentheses
- Amounts in reports are shown in their original currencies, but can be displayed in any active currency
- An administrator can modify conversion rates in Setup. By default, previous conversion rates are not stored and all conversions within opportunities, forecasts, and other amounts use the current conversion rate.
- **Advanced currency management** allows you to manage dated exchange rates within opportunities using Salesforce
- Dated exchange rates are used for opportunities, opportunity products, opportunity product schedules, campaign opportunity fields, opportunity splits, and reports related to these objects and fields. Dated exchange rates are not used in forecasting, currency fields in other objects, or currency fields in other types of reports.
- Cross-object formulas always use the static conversion rate for currency conversion.

Protecting Custom Sharing Code

- Sharing records with „Manual" row cause sharing reason are deleted on record owner
- change Sharing records via custom code can be protected by:
  - •Using Apex Sharing Reasons(supported on custom objects only)
  - •Using Outbound Messagingto external system which will restore sharing
  - •Using a Trigger which calls an Assignment engine built on Salesforce (CoreSite SM)
  - •Using a Shadow Sharing Table with a trigger

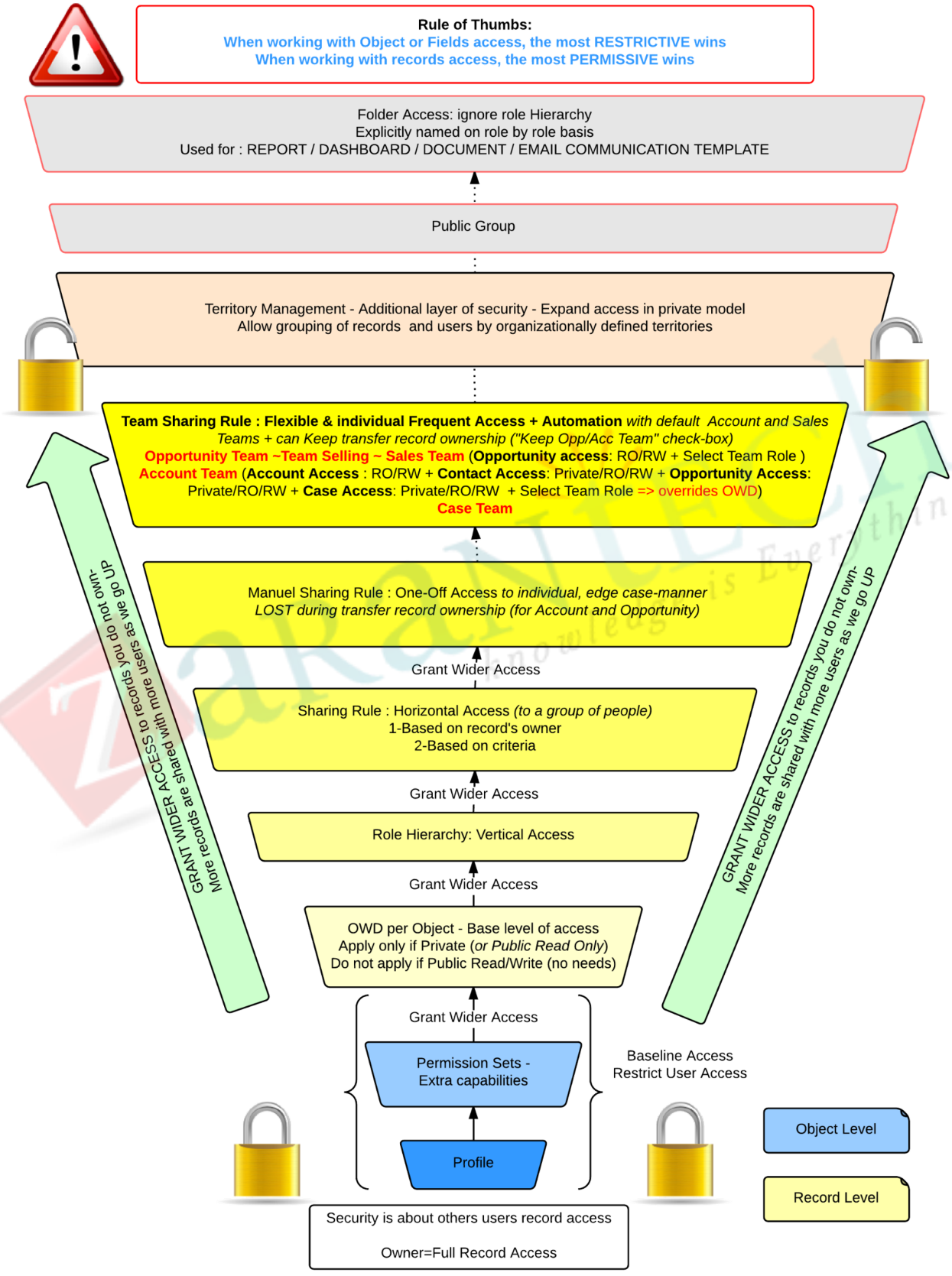
Testing Types

Testing type	Description	When & where
Unit testing	The process of testing each unit of code in a single component. This testing is carried out by the developer as the component is being developed.	During development on a developer environment
Code review	Systematic examinationof computer source code. It is intended to find mistakes overlooked in software development, improving the overall quality of software. Reviews are done in various forms such as pair programming, informal walkthroughs, and formal inspections.	At the moment of merging, commiting code on a developer environment
Functional Testing	A type of black-box testing that bases its test cases on the specifications of the software component under test. Functional testing usually describes what the system does.	After development of a certain piece of functionality, on the QA environment
Integration testing	Testing the interface between the modules; it can be top down, bottom up, big bang.	After merging multiple functionalities or features on the CIT environment
System Integration testing	A high-level software testing process in which testers verify that all related systems maintain data integrity and can operate in coordination with other systems in the same environment. The testing process ensures that all subcomponents are integrated successfully to provide expected results.	SIT phase on the SIT environment
User Acceptance testing	Last phase of the software testing process. During UAT, actual software users test the software to make sure it can handle required tasks in real-world scenarios, according to specifications.	UAT phase on the UAT environment
Performance testing	Includes Stress and Load testing. Coordinated with Salesforce,. Done using LoadRunner, SilkPerformer, Red View, ...	Scheduled during development, SIT, UAT
Smoke testing	Preliminary testing to reveal simple failures severe enough to reject a prospective software release	After deployment to any environment
Regression testing	A type of software testing that ensures that previously developed and tested software still performs the same way after it is changed or interfaced with other software	After deployment to any environment
Data migration testing	Testing that data is correctly migrated and that data integrity is maintained between systems.	UAT or Staging phase on the UAT or Staging environment

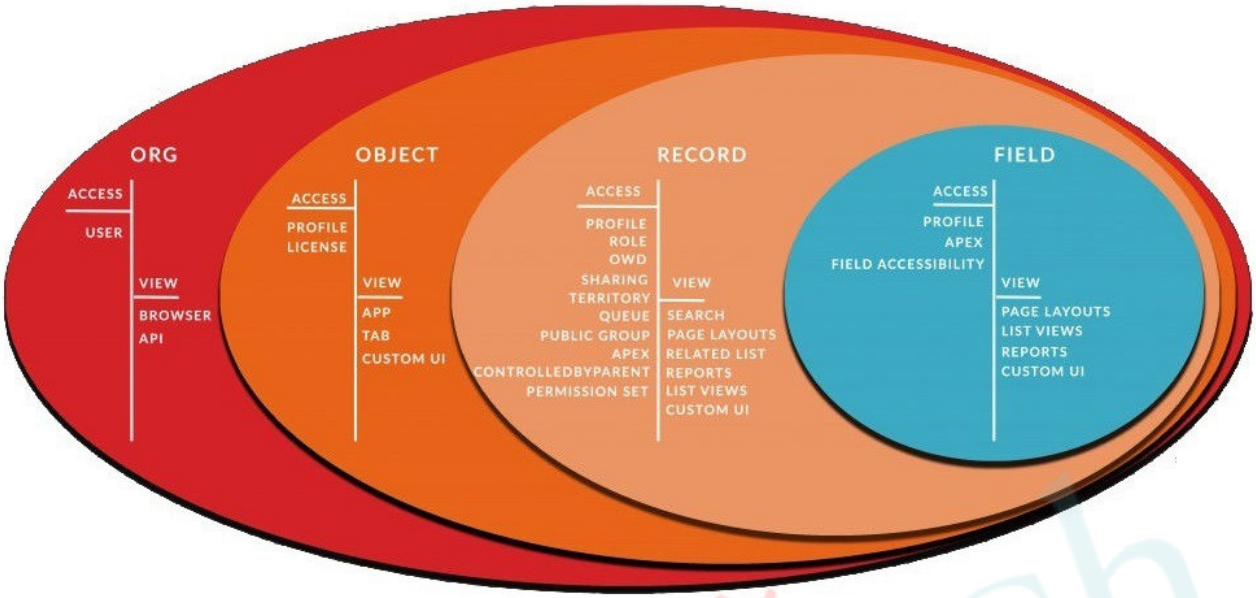
Salesforce Sharing Methods

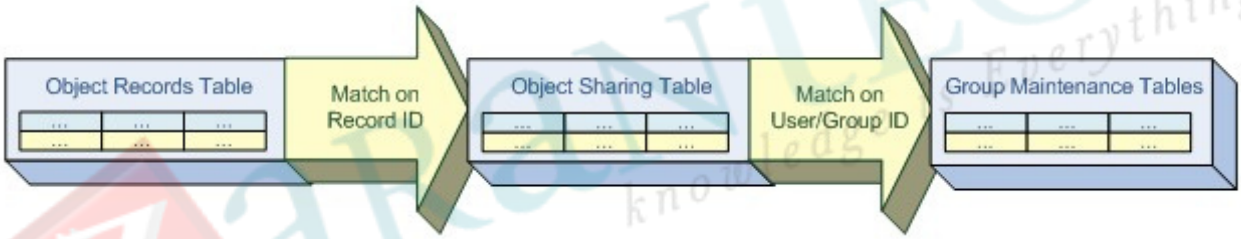
Method	Description
Profiles	A profile is a group/collection of settings and permissions that define what a user can do in salesforce. A profile controls “Object permissions, Field permissions, User permissions, Tab settings, App settings, Apex class access, Visualforce page access, Page layouts, Record Types, Login hours & Login IP ranges.
Permission Sets	Very similar to profile. The main difference between these two is that user can have only one profile and can have multiple permission sets at time. In such case, most permissive setting applies.
Organization Wide Defaults	Organization-wide sharing defaults set the baseline access for your records. Options include Private, Public Read Only, Public Read/Write and Controlled By Parent
Role Hierarchy	Represents a level of data access that a user or group of users needs. Users assigned to roles near the top of the hierarchy get to access the data of all the users who fall directly below them in the hierarchy.
Sharing rules	Automatic exceptions to your organization-wide sharing settings for defined sets of users. Standard way to open up record access.
Sharing Sets	Grants high-volume users access to any record associated with an account or contact that matches the user’s account or contact. You can also grant access to records via access mapping in a sharing set, which supports indirect lookups from the user and target record to the account or contact. For example, grant users access to all cases related to an account that’s identified on the users’ contact records.
Sharing Groups	Allow you to share records owned by high-volume community users with internal and external users in your communities.
Partner Super Users	Partner Super Users have access to data owned by their peers (same role). Partner super user access applies only to cases, leads, custom objects, and opportunities.
Public Groups	A group consists of a set of users. A group can contain individual users, other groups, or the users in a particular role or territory. It can also contain the users in a particular role or territory plus all the users below that role or territory in the hierarchy.
Queues	Queues help you prioritize, distribute, and assign records to teams who share workloads. Queue members and users higher in a role hierarchy can access queues from list views and take ownership of records in a queue. Use queues to route lead, order, case, and custom object records to a group.
T eams	For accounts, opportunities, and cases, record owners can use teams to allow other users access to their records. A team is a group of users that work together on an account, sales opportunity, or case. Record owners can build a team for each record that they own. The record owner adds team members and specifies the level of access each team member has to the record, so that some team members can have read-only access and others can have read/write access. The record owner can also specify a role for each team member, such as “Executive Sponsor.” In account teams, team members also have access to any contacts, opportunities, and cases associated with an account.
Terrirory management	An account sharing system that grants access to accounts based on the characteristics of the accounts
Implicit sharing	Sharing not configured by administrators; it is defined and maintained by the system to support collaboration among members of sales teams, customer service representatives, and clients or customers.Includes Parent, Child, Portal and High Volume
Manual sharing	Sharing done directly by record owners by clicking the Share button
Apex managed sharing	Sharing generated through Apex by creating records in the Sharing objects

Salesforce Sharing Illustrated



Sharing Architecture

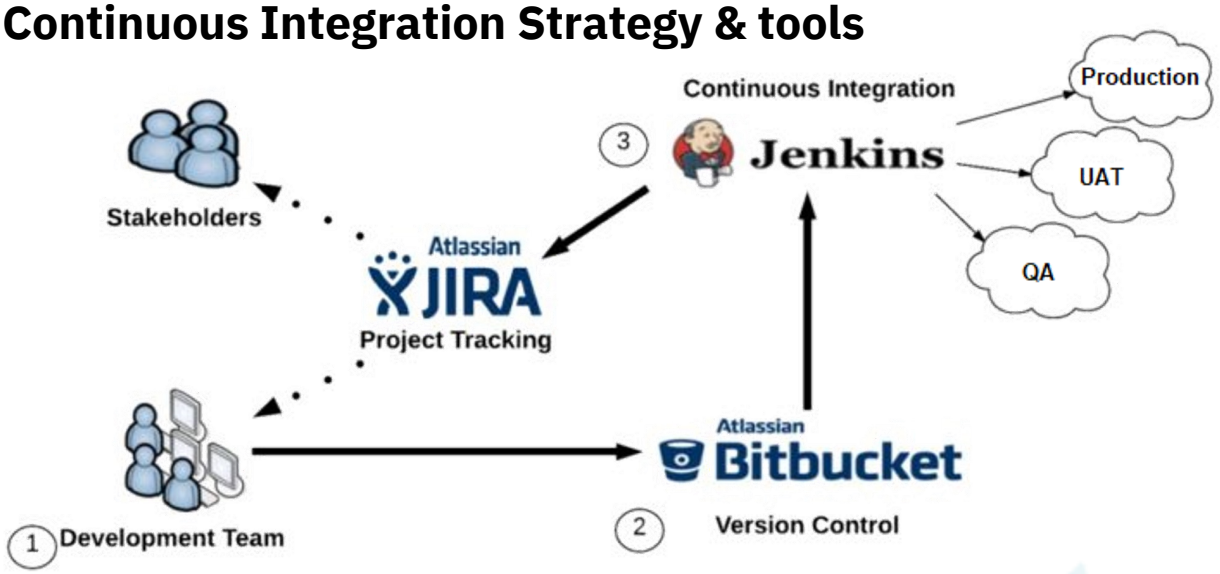


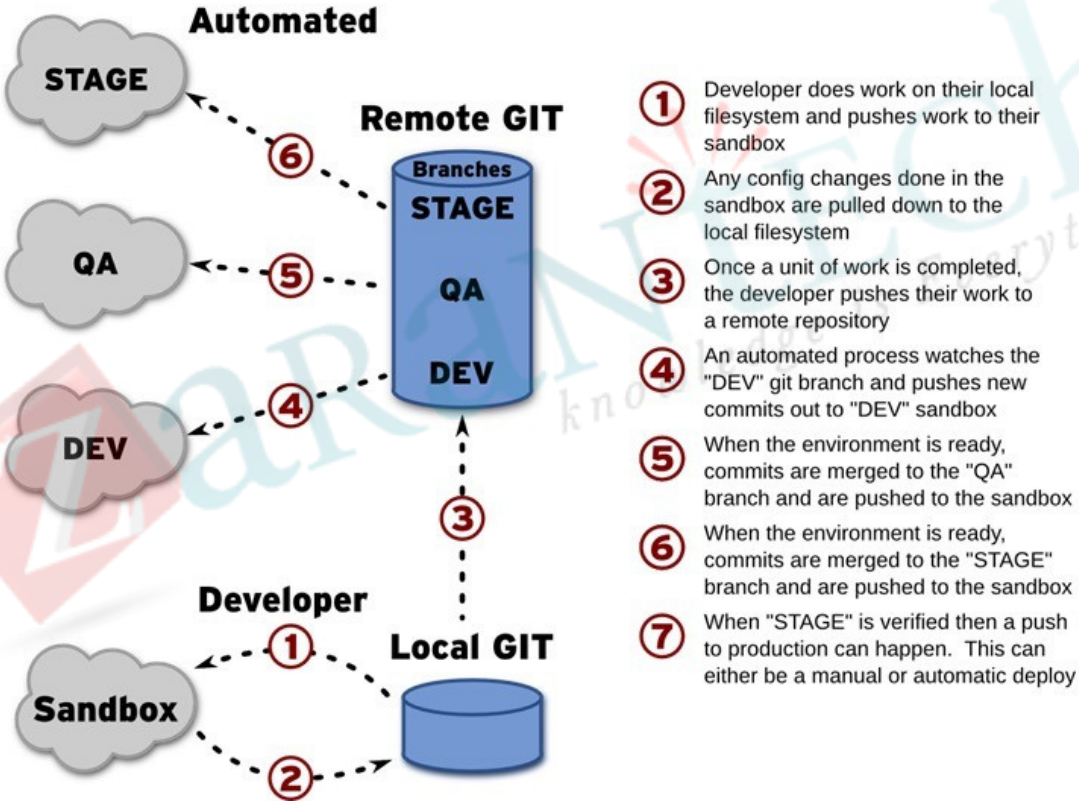


Restricting Org Access

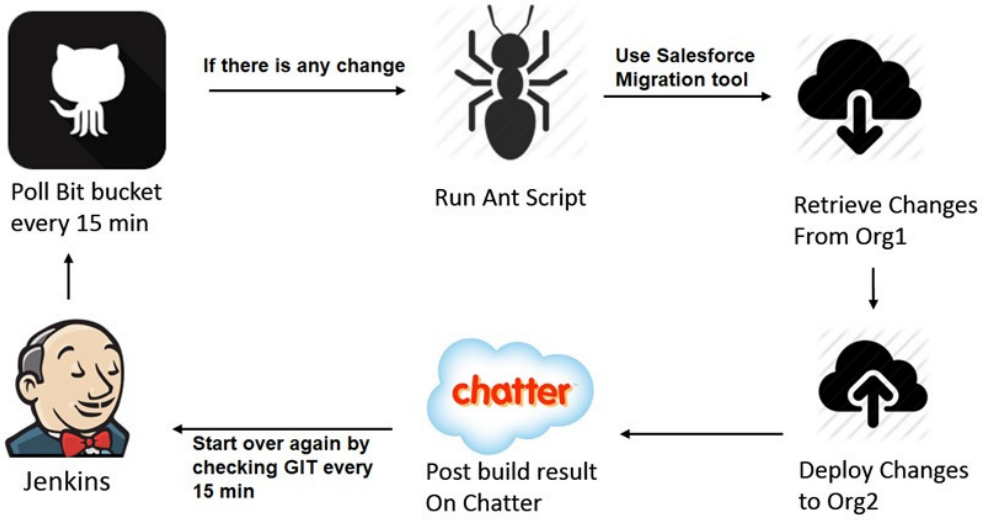
Method	Description
My domain	Select a login policy to prevent users from logging in with the generic https://<instance>.salesforce.com/ login page and then being redirected to your subdomain URLs after login.Supports branding and SSO.
Login IP ranges	Control login access at the user level by specifying a range of allowed IP addresses on a user's profile. When you define IP address restrictions for a profile, a login from any other IP address is denied.
Login IP hours	Specify the hours when users can log in based on the user profile.
Two-Factor Authentication	For each profile, you can require a verification code (also called a time-based one-time password, or TOTP) instead of the standard security token. Users connect an authenticator app that generates verification codes to their account.
Trusted IP Ranges	Define a list of IP addresses from which users can log in without receiving a login challenge for verification of their identity, such as a code sent to their mobile phoneor token.

Continuous Integration Strategy & tools





1. Developer does work on their local filesystem and pushes work to their sandbox
2. Any config changes done in the sandbox are pulled down to the local filesystem
3. Once a unit of work is completed, the developer pushes their work to a remote repository
4. An automated process watches the "DEV" git branch and pushes new commits out to "DEV" sandbox
5. When the environment is ready, commits are merged to the "QA" branch and are pushed to the sandbox
6. When the environment is ready, commits are merged to the "STAGE" branch and are pushed to the sandbox
7. When "STAGE" is verified then a push to production can happen. This can either be a manual or automatic deploy



ETL & ESB integration

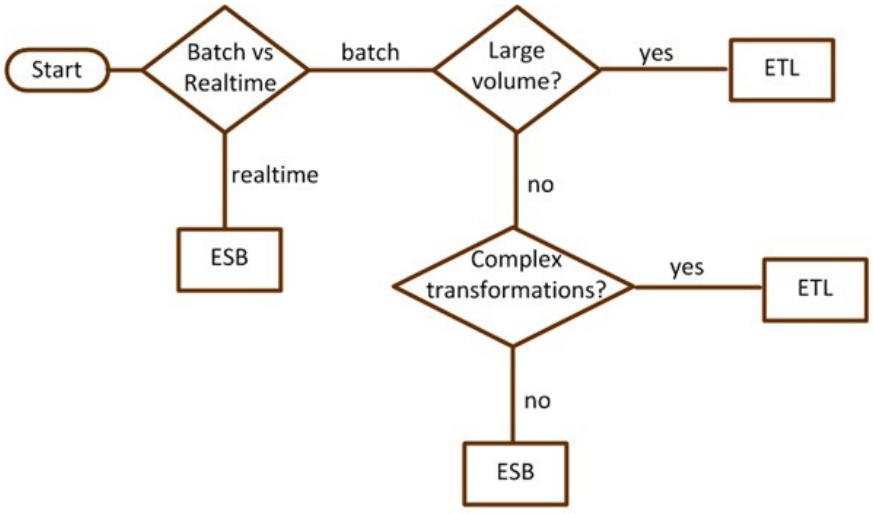
ETL tools support the process of Extracting, Transforming and Loading large volumes of data from multiple data sources. Used for batcch, scheduled or ad-hoc data operations. They can be used to perform data migration or data integration. Some examples include Talend, Informatica, Jitterbit, Data migrator, etc.

An enterprise service bus (ESB) is a middleware tool used to distribute work among connected components of an application. ESBs are designed to provide a uniform means of moving work, offering applications the ability to connect to the bus and subscribe to messages based on simple structural and business policy rules.The most often used ESB is MuleSoft.

When to use ESB:
•When system integration points grow beyond two, with additional integration requirements.
•When using multiple protocols such as FTP, HTTP, Web Service, and JMS etc.
•When there is a requirement for message routing based on message content and similar parameters.

How an ESB architecture maps to our five core integration principles:

•**Orchestration**: Composing several existing fine-grained components into a single higher order composite service. This can be done to achieve appropriate "granularity" of services and promote reuse and manageability of the underlying components.
•**Transformation**: Data transformation between canonical data formats and specific data formats required by each ESB connector. An example of this would be transforming between CSV, Cobol copybook or EDI formats to either SOAP/XML or JSON. Canoncial data formats can greatly simplify the transformation requirements associated with a large ESB implementation where there are many consumers and providers, each with their own data formats and definitions.
•**Transportation**: Transport protocol negotiation between multiple formats (such as HTTP, JMS, JDBC). Note: Mule treats databases like another "service" by making JDBC just another transport (or endpoint) where data can be accessed.
•**Mediation**: Providing multiple interfaces for the purpose of a) supporting multiple versions of a service for backwards compatibility or alternatively, b) to allow for multiple channels to the same underlying component implementation. This second requirement may involve providing multiple interfaces to the same component, one legacy interface (flat file) and one standards compliant (SOAP/XML) interface.
•**Non-functional consistency**: For a typical ESB initiative, this can include consistency around the way security and monitoring policies are applied and implemented. Additionally, the goals of scalability and availability can be achieved by using multiple instances of an ESB to provide increased throughput (scalability) and eliminate single-points-of-failure (SPOFs), which is the key objective for highly available systems.



Account models

B2C account models

Model	Description
Person Accounts	Person accounts store information about individual people by combining certain account and contact fields into a single record.Has some limitations and can not be reverted.
Private Contact	Allows Contacts to be created without accounts. Not recommended as an B2C model
Household model	The standard Salesforce Account object acts as the household, with numerous contacts and opportunities associated with it. Default and recommended for NPSP.
1-to-1 Account Model	Works by creating an Account object for each Contact you create. The Account is automatically created when you create a Contact, and the name of the new Account is identical to the name of the new Contact. The Account object itself, however, acts "invisibly" behind the scenes, and exists for the sole purpose of conforming to the standard Salesforce Account model (which requires all Contacts be associated with an Account).
The Bucket model	All Contacts are associated with a single Account, as if all of your contacts have been dumped in one "bucket".

Account Hierarchy models

Model	Description
Global Enterprise Account	Establish one global account and link all contacts, opportunities, cases, and so on to that single overarching account. Using one global account makes it easy to find that account's records and to report on that account at the enterprise level. But it's harder to manage a large mass of information, and not being able to easily view the big picture might make it hard to see what each location needs from you for your relationship to be successful.
Location-Specific Accounts	Establish accounts for each location and create contacts, opportunities, cases and so on separately for each location. With this option, you maintain more accounts and need to set up a few more complex reports to get the big picture. But using multiple accounts means you can take advantage of account ownership, hierarchies, specific sharing settings, and more granular reporting. You can also more easily track and report on opportunities, cases, and other interactions for each account.

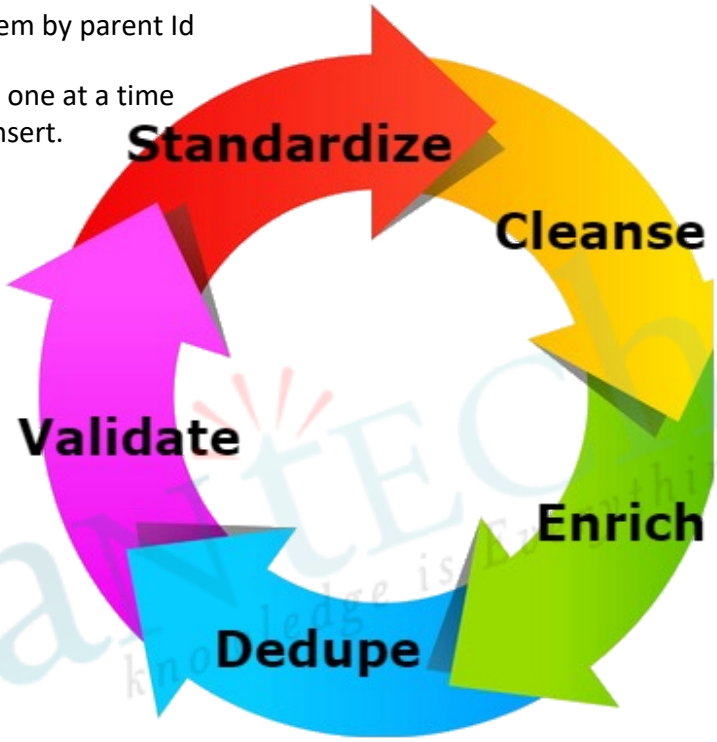
Contacts to multiple Accounts models

Model	Description
Account Contact Roles	Standard salesforce solution. Does not support custom fields, layouts, validation, workflow, triggers, record types, field history, etc. Sharing is derived from Account.
Contacts to Multiple Accounts	Standard salesforce solution. Supports custom fields, layouts, validation, workflow, triggers, etc. Supports record types and field history. Sharing is derived from Account and Contact
Custom Junction Object	Create a custom object to connect a Contact with multiple Accounts. Supports all features, but requires custom development.

Data migration & Backup

Data Loading best practices:

- Use BULK API for more that a few thound records
- Use the fastest operation Insert > Update > Upsert
- Use Public Read/Write security to avoid sharing calculation
- Disable triggers, workflow rules and validation
- Some fields might be transformed during migration (id, autonumber, audit fields)
- When changing child records, group them by parent Id
- Use defer-sharing
- Activate sharing rules after loading and one at a time
- Audit fields can be populated only on insert.



Backup types:

- Full
- Incremental
- Partial

Data archiving options:

- BigObject
- Outside of Salesforce
- Weekly export
- Data loader
- Reporting snapshot

Data governance & stewardship

Data governance is a process to ensure usability, quality, and policy compliancce of the data asset. It includes business definitions, data quality and security rules, supports UI and integration design.
It defines what is collected, how is it kept secured, who can CRUD, what quality rules are there, how available and usable is the data, ...

Data stewardship is a cross-functional tactical role and activities to ensure adherence to data governance rules and spirit. It includes data quality monitoring, work flow, and maintenance.

Mastar data management is the effort made by an organization to create one single master reference source for all critical business data, leading to fewer errors and less redundancy in business processes. It consists of 3 pillars: Mastering data, Mastering data relationships and Mastering events.

Understand the **linear flow of data**:
How are records generated?
Why are they created?
What are they used for?
What is reported on?
Where is the hand-off?

Methods to provision users

Method	Description
Manual provisioning	A user is created manually by an administrator.
API provisioning	Provision users by using the SOAP or REST API on theUser object.
Programmatic provisioning	Povision users in Apex code.
JIT provisioning with SAML	Use a SAML assertion to create regular and portal users on the fly the first time they try to log in. This eliminates the need to create user accounts in advance.A user is created when he logs in via SSO.
Mass user provisioning	Create a large number of users by using Bulk API, Data Loader or an ETL tool.
Identity connect with AD	Integrates Microsoft Active Directory (AD) with Salesforce. User information entered in AD is shared with Salesforce seamlessly and instantaneously. Companies that use AD for user management can use Identity Connect to manage Salesforce accounts.Changes in AD are reflected in Salesforce in near real time.
Self-registration	Users can self-register when first visiting the site. Works with community users only.
Social sign-on provisioning	Users can sign in using a social site credentials. Supported sites include LinkedId, Facebook, Twitter, Google, Janrain, Salesforce, and any srevice who implements the OpenID Connect protocol or Oauth. Works with community users only.

Single Sign On Methods

Method	Description
SSO with multiple Orgs	SSO between multiple Salesforce orgs. Can be enabled in Setup for both orgs. Works only with internal users.
SSO with AD	Salesforce is integrated with AD using Identity Connect or ADFS
Social Sign On	Sign on via a Social site credentials. Works with community users only.
Federated Authentication	The platform receives a SAML assertion in an HTTP POST request. The SAML assertion has a limited validity period, contains a unique identifier, and is digitally signed. If the assertion is still within its validity period, has an identifier that has not been used before, and has a valid signature from a trusted identity provider, the user is granted access to the application. If the assertion fails validation for any reason, the user is informed that their credentials are invalid.
Delegated Authentication	An internal WS authenticates users. It receives an username, password and sourceIP and returns true or false.

*Never enable SSO for Admin users

Data quality and duplicate management

Data quality attributes:

- Age
- Completeness
- Accuracy
- Consistency
- Duplication
- Usage

Best data quality and archiving policy is defined at design time.

Use data quality scoring and reporting.

How to safely delete data:

1.Isolate suspects
2.Flag for elimination and color code
3.Hide with security
4.Wait (around 3 months)
5. Backup
6. Delete

Deduplication step order:

1.Accounts vs Accounts
2.Contacts within Account
3.Contacts between Accounts
4.Accounts vs Accounts
5. Leads
6.Leads to Contacts

Data quality tools:

- Data.com (works on Accounts, Contacts, and Leads)
- Duplicate rules (works on Account, Lead, Contact, and Custom object)
- Merging records (works on Accounts, Contacts, and Leads)

Archival strategy options:

1.In place archiving –Storage objects, record archived indicator
2.External archiving –on premise DWH, external cloud solution ,flat file archiving
3.Hybrid solution –data tiering

Two factor Authentication

As a Salesforce admin, you can enhance your org's security by requiring a second level of authentication for every user login. You can also require two-factor authentication when a user meets certain criteria, such as attempting to view reports or access a connected app.

When a user logs in from outside a trusted IP range and uses a browser or app we don't recognize, the user is challenged to verify identity. We use the highest-priority verification method available for each user. In order of priority, the methods are:

1.Verification via push notification or location-based automated verification with the Salesforce Authenticator mobile app (version 2 or later) connected to the user's account.
2.Verification via a U2F security key registered with the user's account.
3.Verification code generated by a mobile authenticator app connected to the user's account.
4.Verification code sent via SMS to the user's verified mobile phone.
5.Verification code sent via email to the user's email address.
6.Login Flows After identity verification is successful, the user doesn't have to verify identity again from that browser or app, unless the user:

•Manually clears browser cookies, sets the browser to delete cookies, or browses in private or incognito mode
•Deselects **Don't ask again** on the identity verification page

Managed packages to propose

Use Case	Package/solution
Document generation	Conga Composer WebMerge Drawloop
eSignature	Docusign EchoSign
ESB	Mulesoft
CPQ	Salesforce CPQ CloudSense CPQ
Marketing	Marketing Cloud Marketo
Accounting	FinancialForce Intaact
Analytics	Wave analytics
Cloud Data storage and long operations	Heroku AWS
Document Storage	Box Google Drive SharePoint DropBox CRM Content

Mobile Strategy Decision

	Native	HTML5	Hybrid	Salesforce
Graphics	Native APIs	HTML, Canvas, SVG	HTML, Canvas, SVG	App Standard Salesforce + Visualforce
Performance	Fast	Slow	Slow	Slow
Native look and feel	Native	Emulated	Emulated	Emulated(VF)
Distribution	Appstore	Web	Appstore	Appstore
Camera	Yes	No	Yes	No
Notifications	Yes	No	Yes	No
Contacts, Calendar	Yes	No	Yes	Yes
Offline storage	Secure file storage	Shared SQL	Secure file system, shared SQL	Yes
Geolocation	Yes	Yes	Yes	Yes
Swipe	Yes	Yes	Yes	Yes
Pinch, spread	Yes	No	No	No
Connectivity	Online and offline	Mostly online	Online and offline	Mostly online
Development skills	ObjectiveC, Java	HTML5, CSS, Javascript	HTML5, CSS, Javascript	Visualforce
Cross-Platform	No	Yes	Yes	Yes
Easy to deploy	No	Yes	No	Yes
Fully customisable UI	Yes	Yes	Yes	No
Speed of Development	Slow	Medium	Medium	Fas t

Salesforce Communities

Community templates:

- Visualforce + Tabs
- Customer Service (Napili)
- Kokua & Koa
- Lightning Communities
- Partner Central
- Customer Account Portal

Rollout Strategy phases:

- Establish
- Manage
- Engage
- Measure

Branding options:

- Select color scheme
- Customize emails
- Logo
- Login page
- Custom domain (my domain)

Project methodology

Agile vs Waterfall

Factor	Status	Agile	Waterfall
Project Size and Complexity	Small, less complex	Yes	
	Large, more complex		Yes
Customer Availability	Frequent	Yes	
	Not too frequent		Yes
Level of Integration	Simple	Yes	
	Complex		Yes
Customer Tolerance for Scope	Flexible in budget and schedule	Yes	
	Fixed budget		Yes
	Regulatory/compliance needs		Yes
Time to market	Rapid deployment	Yes	
	Full-featured application must be delivered		Yes

Methodology Artifacts

•Centralized Communication/Collaboration/Documentation repository
•Requirements management from Business Requirements traced all the way through test cases and deployed technical components
•Governance levels (Steering committee, Center of Excellence, Architectural Review board with a detailed understanding of interdependency between groups)
•Quality Control (Development of Technical Design Standards, Common Usability Requirements, Peer Review, Code Review, Deployment checklist, etc.)
•Thorough testing strategy with as much automation that can reasonably be built
•Dependency management through PMO, scrum teams, scrum of scrums, status reports, risk & issue management
•Key Tools:
- o Project management software (MS Project, Agile Accellerator)
- o Requirements Repository (Rally, Jira, Excel)
- o Traceability Matrix
- o Test Suite Management (allows to track tests against requirements)

Center of Exellence

•A team of knowledgeable and experienced business process management, customer relationship management, and business domain experts equipped with an arsenal of best practices and tools. At its most mature state, it is a highly formalized and self-directing entity that is responsible for supporting business users and shepherding even the most complex projects to successful completion. The COE promotes the use of business process management as means of linking an organization's strategy to their day to day operations.
•A cross-functional team looking both inside and outside the organisation to capture new knowledge and practices
•Scope: Charter, Business backlog, technology release management, communications
•Structure: Consolidated, federated, hybrid
•Roles and responsibilities: Release, business, scrum teams, architectural, training, support
•Key roles: Business Analyst, Enterprise Architect, Project Management Office
•Best Practice Centers – These are the "sharers," focusing on creating environments where business units can collaborate with each other on like-minded processes and technologies, designed to more rapidly enable lines of business.
•DevOps Centers – These are the "doers," focusing on providing a shared service to scope, design, develop, and deliver across with optimal IT governance, designed to foster standardization and reuse.
•Competency Centers – These are the "guiders", focusing on establishing best practices and standards to enable, build competency, and embed expertise within individual lines of business through the creation of targeted improvement agendas.
•Innovation Centers – These are the "creators," focusing on the incubation and experimentation required to develop the capabilities with (emerging) technologies, designed to accelerate maturity and time to value.

Chatter features

Feature	Description
Connect to Business Processes	Create support cases, update sales opportunities, and approve project funds — all from within the community. Connect data and records to internal systems to maximize efficiency and eliminate errors.
Actions	Approve expense reports, create support cases, update orders, and more, all from the feed. Customize actions, integrating third-party or your own custom apps with full social and mobile capability.
ile	Post to a group, download a file, or update a service case or sales opportunity using the Salesforce mobile app from any device. Use custom mobile actions to drive progress forward, wherever you are.
Engagement	Engage your organization to participate. Contribution scores give employees incentives to join the conversation, while badging and endorsements highlight the key contributors.
ips	Structure a discussion and activity. Create groups for your team, an event, a new account, or a campaign. Share files and records, integrate video, and even invite customers. Private groups protect discussions.
	Assess employee opinions on any topic, at any time. Anyone can post a poll to a group for feedback on a specific subject, or connect with the broader organization to gauge popular sentiment.
Rich Feeds	Do it all in the feed. Keep up with critical projects, topics, and teams. Post files, videos, images, and other assets. Even collaborate on sales opportunities, service cases, and marketing campaigns.
cs	Present the most up-to-date content on any subject discussed in the community. Topics automatically collect relevant posts and answers, suggest groups, experts, files, and other related resources.
Recommendations	No need to search for information, people, or files. Chatter uses your interests and activity to update content on subjects, projects, or products and sends it to your personalized feed or profile page.
Salesforce Files	Secure, social, and mobile file sharing puts the files you need, and new resources, right in your feed. Unlock and securely sync files to any mobile device from third-party repositories with Files Connect.
Answers	Collect answers to common questions and structure them so employees quickly find the information they need. Highlight questions about internal processes, projects, products, or any subject you need.

Presentation outline

Slide	Description
Title	
Agenda	
About me	Short bio
About the company	Who is the client
Requirements	A detailed list of all requirements
Assumptions	
Actors & licenses	
System landscape	Diagram
Integrations	Explanation of integartion methods
Data model	Diagram
Role hierarchy	Diagram
Mobile application	Native vs Hybrid vs Html5 vs SF1
Communities	Customer or Partner
Authentication methods	Standard, SSO, Social Sign-On
LDV mitigation	Which objects, volume, mitigation strategy
Reporting	Object, reporting method
Other Salesforce features used	Workflow rules, Profiles, Permission Sets, OWD, Outbound messaging, Assignment rules, Escalation rules, Lead conversion, Process builder, territory management, sharing rules, CRM Content, Translations, ...
Data migration	Objects, Systems, process
Project management & governance	Agile vs Waterfall
Methodology artifacts	Teams, documents, process
Development strategy	CI, Solution design, tools
Sandbox structure	Which sandboxes will be used
Testing strategy	Which tests will we do &
Risks & mitigation strategies	when Other risks
Thank you, Q&A	

Other Security features

Certificates and Keys

Salesforce certificates and key pairs are used for signatures that verify a request is coming from your organization. They are used for authenticated SSL communications with an external web site, or when using your organization as an Identity Provider. You only need to generate a Salesforce certificate and key pair if you're working with an external website that wants verification that a request is coming from a Salesforce organization.
Certificates can be self-signed or signed by an authorization authority.

Named credentials

A named credential specifies the URL of a callout endpoint and its required authentication parameters in one definition. To simplify the setup of authenticated callouts, specify a named credential as the callout endpoint. If you instead specify a URL as the callout endpoint, you must register that URL in your org's remote site settings and handle the authentication yourself. For example, for an Apex callout, your code would need to handle authentication, which can be less secure and especially complicated for OAuth implementations.

WSDL Types

Enterprise WSDL:
a) The Enterprise WSDL is strongly typedwhich makes it easier to use
b) The Enterprise WSDL is tied (bound) to a specific configuration of Salesforce (ie. a specific organization's Salesforce configuration).
c) The Enterprise WSDL changes if modifications (e.g custom fields or custom objects) are made to an organization's Salesforce configuration.

For the reasons outlined above, the Enterprise WSDL is intended primarily for

Customers. **Partner WSDL:**

a) The Partner WSDL is loosely typedwhich makes it harder to use
b) The Partner WSDL can be used to reflect against/interrogate any configuration of Salesforce (ie. any organization's Salesforce configuration).
c) The Partner WSDL is static, and hence does not change if modifications are made to an organization's Salesforce configuration.

For the reasons outlined above, the Partner WSDL is intended primarily for Partners.

Reports and Dashboards

Report formats

Format	Description
Tabular	Tabular reports are the simplest and fastest way to look at data. Similar to a spreadsheet, they consist simply of an ordered set of fields in columns, with each matching record listed in a row. Tabular reports are best for creating lists of records or a list with a single grand total. They can't be used to create groups of data or charts, and can't be used in dashboards unless rows are limited.
Summary	Summary reports are similar to tabular reports, but also allow users to group rows of data, view subtotals, and create charts. They can be used as the source report for dashboard components. Use this type for a report to show subtotals based on the value of a particular field or when you want to create a hierarchical list, such as all opportunities for your team, subtotaled by Stage and Owner. Summary reports with no groupings show as tabular reports on the report run page.
Matrix	Matrix reports are similar to summary reports but allow you to group and summarize data by both rows and columns. They can be used as the source report for dashboard components. Use this type for comparing related totals, especially if you have large amounts of data to summarize and you need to compare values in several different fields, or you want to look at data by date and by product, person, or geography. Matrix reports without at least one row and one column grouping show as summary reports on the report run page.
Joined	Joined reports let you create multiple report blocks that provide different views of your data. Each block acts like a "sub-report," with its own fields, columns, sorting, and filtering. A joined report can even contain data from different report types.

*Reports can only go 3 levels deep

Dashboard component types

Component Type	Image	Description
Chart		Use a chart when you want to show data graphically. You can choose from a variety of chart types.
Gauge		Use a gauge when you have a single value that you want to show within a range of custom values.
Metric		Use a metric when you have one key value to display.For example, if you have a report showing the total amount for all opportunities in theClosed,Commit, andBase Case stages in the current month, you can name that value and use it as a revenue target for the month displayed on the dashboard.
Table		Use a table to show a set of report data in column form.Supports sort order, and conditional highlighting.
Visualforce Page	N/A	Use a Visualforce page when you want to create a custom component or show information not available in another component type.Visualforce pages must have a StandardSetController (recordsetvar) or a CustomController to appear in the Dashboards.

Integration

Integration Types

- UI Integration
- Data integration
- Security integration
- Business process integration

Integration Design pattern

- API Wrapper class
- Delegator class
  - Optional components (logging, mapping, session handling, exception handling)

Integration Mechanisms

Mechanism	Description
External objects	Data is fetched from an external system using Salesforce Connect Adapters. Addapter types supported are: Cross-org, OData 2.0, OData 4.0, Custom adapter created via Apex.
Canvas	Canvas enables you to easily integrate a third-party application in Salesforce. Canvas is a set of tools and JavaScript APIs that you can use to expose an application as a canvas app. This means you can take your new or existing applications and make them available to your users as part of their Salesforce experience.
Push notifications	Mobile push notifications allow Lightning Platform mobile application developers to easily push notifications to their users' mobile devices when business events occur in the customers' organizations.
REST API	REST API provides a powerful, convenient, and simple REST-based web services interface for interacting with Salesforce. Its advantages include ease of integration and development, and it's an excellent choice of technology for use with mobile applications and web projects.It provides the same methods as SOAP API. It integrates via simple HTTP calls(GET, POST, PUT, PATCH, DELETE) and both XML and JSON.
SOAP API	Use SOAP API to create, retrieve, update or delete records, such as accounts, leads, and custom objects. With more than 20 different calls, SOAP API also allows you to maintain passwords, perform searches, and much more. Use SOAP API in any language that supports Web services.
Chatter REST API	Use Chatter REST API to display Chatter feeds, users, groups, and followers, especially in mobile applications. Chatter REST API also provides programmatic access to files, recommendations, topics, notifications, Data.com purchasing, and more.
Bulk API	Used to extract and load large volumes of data into and from Salesforce
Metadata API	Used to retrieve, deploy, create, update or delete customization information, such as custom object definitions and page layouts, for your organization. This API is intended for managing customizations and for building tools that can manage the metadata model, not the data itself.
Streaming API	Use Streaming API to receive notifications for changes to Salesforce data that match a SOQL query you define, in a secure and scalable way.Supported via PushTopics.
WebService API	Allows developers to create Web Services on Salesforce that can be called by external third parties.
Tooling API	Use Tooling API to build custom development tools or apps for Lightning Platform applications. Tooling API's SOQL capabilities for many metadata types allow you to retrieve smaller pieces of metadata. Smaller retrieves improve performance, which makes Tooling API a better fit for developing interactive applications.Tooling API provides SOAP and REST interfaces.
Apex callouts	Call external SOAP, Rest or other web services from Apex.
Outbound Messages	Standard Salesforce outbound integration. Send an SOAP message with data. The message can include Session Id and supports 24 hour retry mechanism.
Email	Integrate by sending and receiving emails. Must implement InboundEmailHandler
Middleware	Integration via ESB or ETL

OAuth & SSO Flows

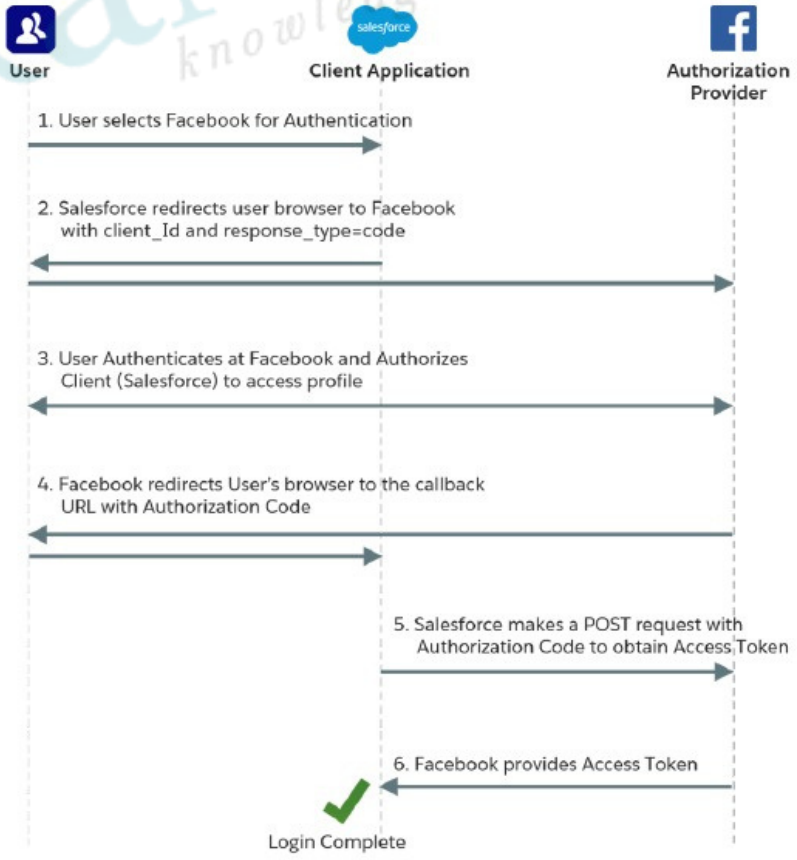
Flow	Description
Web Server	Apps hosted on a secure server use the web server authentication flow. A critical aspect of the web server flow is that the server must be able to protect the client secret. This flow uses an OAuth 2.0 authorization code grant type.
User-Agent	Users can authorize a desktop or mobile application to access data using an external or embedded browser (or user agent) for authentication. These apps often use a scripting language, such as JavaScript, running within the browser. This flow uses the OAuth 2.0 implicit grant type.
JWT Bearer Token Flow	The main use case of the JWT Bearer Token Flow is server-to-server API integration. This flow uses a certificate to sign the JWT request and doesn't require explicit user interaction.
Device Authentication Flow	Command-line apps or applications that run on devices with limited input and display capabilities, such as TVs, appliances, and other IoT devices, can use this flow. Users can connect these applications to Salesforce by accessing a browser on a device with more advanced input capabilities, such as a desktop or a smartphone. Not supported in Salesforce Communities.
Asset Token Flow	Client applications use this flow to request an asset token from Salesforce for connected devices. An OAuth access token and an actor token are exchanged for an asset token. This flow combines issuing and registering asset tokens for efficient token exchange and automatic linking of devices to service cloud asset data.
SAML Bearer Assertion Flow	An app can also reuse an existing authorization by supplying a signed SAML 2.0 assertion, as specified in theSAML 2.0 Profile for OAuth 2.0 Client Authentication and Authorization Grants. A digital signature applied to the SAML assertion authenticates the authorized app.
SAML Assertion Flow	This flow is an alternative for orgs that are using SAML to access Salesforce and want to access the web services API in the same way.
Username and Password	Because the username and password flow passes credentials back and forth, avoid using this flow. Use it only for testing, when a user is not present at app startup, or with highly privileged apps. In these cases, set user permissions to minimize access and protect stored credentials from unauthorized access.
OAuth 2.0 Refresh Token Flow	The web server and user-agent flows also use aOAuth 2.0 Refresh Token Flowto renew access tokens. After a client is authorized for access, it uses a refresh token to get a new access token (session ID).
Canvas App User Flow-Signed Request	This is the default authorization method for canvas apps. The signed request authorization flow varies depending on whether the canvas app's Permitted Users field is set to "Admin approved users are pre-authorized" or "All users may self-authorize."
Canvas App User Flow-Oauth	Canvas supports OAuth 2.0 for authorization. When using OAuth, you have two options: •Web Server OAuth Authentication Flow—When users run your canvas app, they can authorize the app to access their data. •User-Agent OAuth Authentication Flow—When users run your canvas app, they can authorize the app to access their data by using just the browser for authentication. Recommended to use only for development.
SAML Single Sign-On for Canvas Apps	Whether you use signed request or OAuth authorization, you can use SAML-based single sign-on (SSO) to provide your users with a seamless authentication flow. You can leverage Salesforce as an identity provider or as a service provider. SAML SSO enables you to give your users automatic authentication into your canvas app via SAML and authentication into Salesforce via the signed request.

LDV Mitigation strategies

Strategy	Description
Indexes	Salesforce supports custom indexes to speed up queries, and you can create custom indexes by contacting Salesforce Customer Support.
Skinny tables	Salesforce can create skinny tables to contain frequently used fields and to avoid joins. Doing so keeps the skinny tables in sync with their source tables when the source tables are modified. If you want to use skinny tables, contact Salesforce Customer Support. When enabled, skinny tables are created and used automatically where appropriate. You can't create, access, or modify skinny tables. For each object table, Salesforce maintains other, separate tables at the database level for standard and custom fields. This separation ordinarily requires a join when a query contains both kinds of fields. A skinny table contains both kinds of fields and does not include soft-deleted records.
Data archiving	Remove old data from Salesforce objects that are actively used to another internal or external archiving object or database
Managing Data Skew	Avoid records with a lot of child records (Account Skew, Ownership Skew, Lookup Skew). Use triggers instead of workflows, picklists instead of lookup and schedule automated updates at low load times to avoid record locking if needed.

OAuth & SSO Flow diagrams

Social Sign-On (SP-Initiated SSO + Oauth)

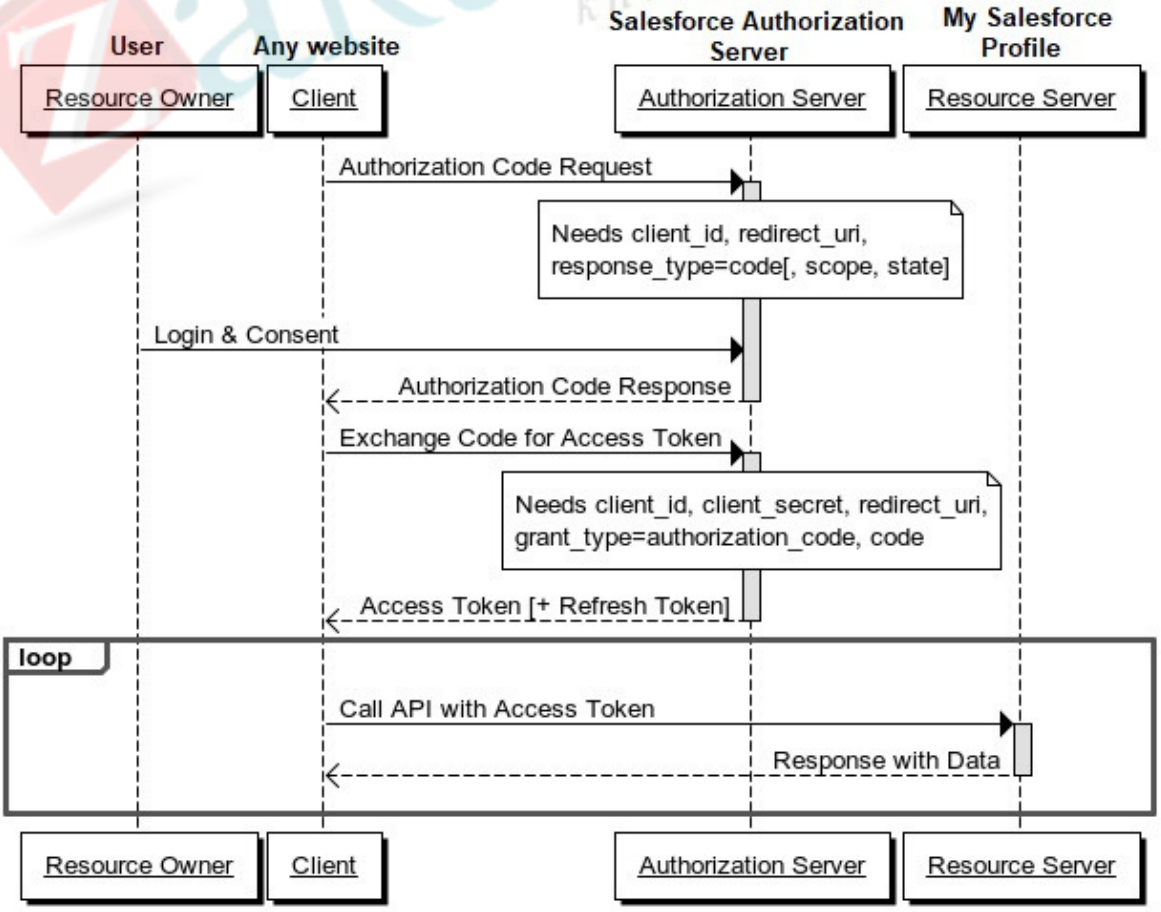


OAuth & SSO Flow diagrams

Web Server Flow

It should be used as soon as the client is a web server. It allows you to obtain a long-lived access token since it can be renewed with a refresh token (if the authorization server enables it).

Scenario:
1.A website wants to obtain information about your Google profile.
2.You are redirected by the client (the website) to the authorization server (Google).
3.If you authorize access, the authorization server sends an authorization code to the client (the website) in the callback response.
4.Then, this code is exchanged against an access token between the client and the authorization server.
5.The website is now able to use this access token to query the resource server (Google again) and retrieve your profile data.
You never see the access token, it will be stored by the website (in session for example). Google also sends other information with the access token, such as the token lifetime and eventually a refresh token.
This is the ideal scenario and the safer one because the access token is not passed on the client side (web browser in our example).

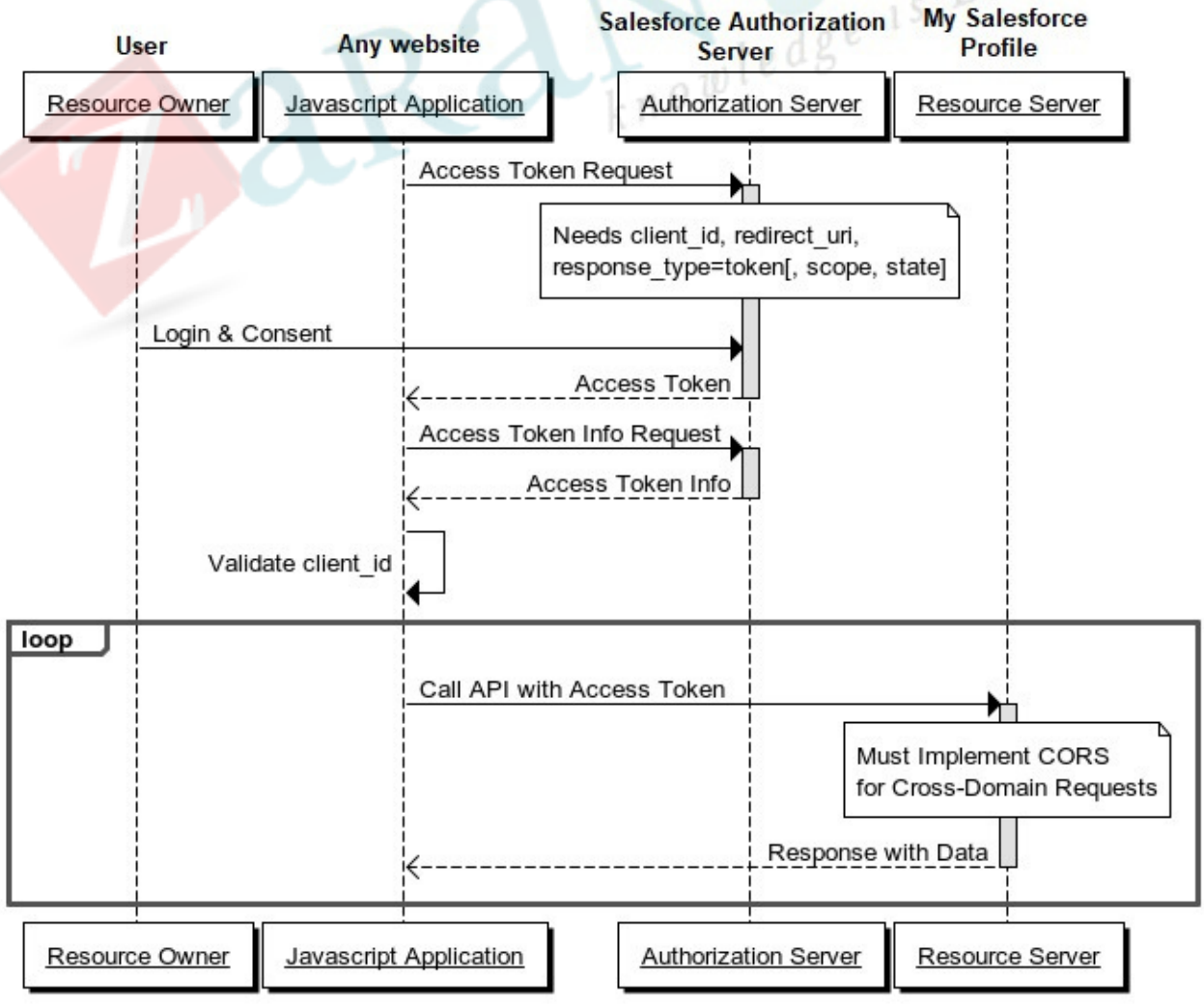


OAuth & SSO Flow diagrams

User Agent Flow

It is typically used when the client is running in a browser using a scripting language such as Javascript. This grant type does not allow the issuance of a refresh token.

Scenario:
1.The client (AngularJS) wants to obtain information about your Facebook profile.
2.You are redirected by the browser to the authorization server (Facebook).
3.If you authorize access, the authorization server redirects you to the website with the access token in the URI fragment (not sent to the web server). Example of callback: http://example.com/oauthcallback#access_token=MzJmNDc3M2VjMmQzN.
4.This access token can now be retrieved and used by the client (AngularJS) to query the resource server (Facebook). Example of query: https://graph.facebook.com/me?ccess_token=MzJmNDc3M2VjMmQzN.
This type of authorization should only be used if no other type of authorization is available. Indeed, it is the least secure because the access token is exposed (and therefore vulnerable) on the client side.



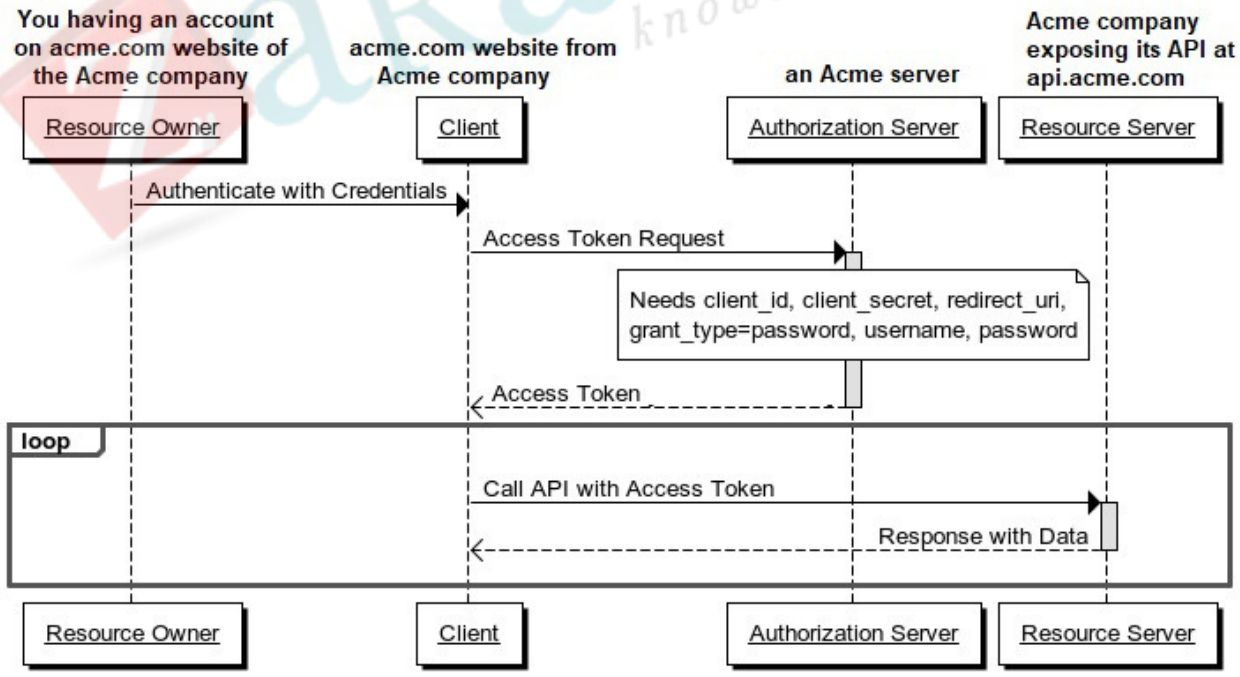
OAuth & SSO Flow diagrams

Username Password flow

With this type of authorization, the credentials (and thus the password) are sent to the client and then to the authorization server. It is therefore imperative that there is absolute trust between these two entities. It is mainly used when the client has been developed by the same authority as the authorization server. For example, we could imagine a website named example.com seeking access to protected resources of its own subdomain api.example.com. The user would not be surprised to type his login/password on the site example.com since his account was created on it.No refresh token is issued.

Scenario:
1.Acme company, doing things well, thought to make available a RESTful API to third-party applications.
2.This company thinks it would be convenient to use its own API to avoid reinventing the wheel.
3.Company needs an access token to call the methods of its own API.
4.For this, company asks you to enter your login credentials via a standard HTML form as you normally would.
5.The server-side application (website acme.com) will exchange your credentials against an access token from the authorization server (if your credentials are valid, of course).
6.This application can now use the access token to query its own resource server (api.acme.com).

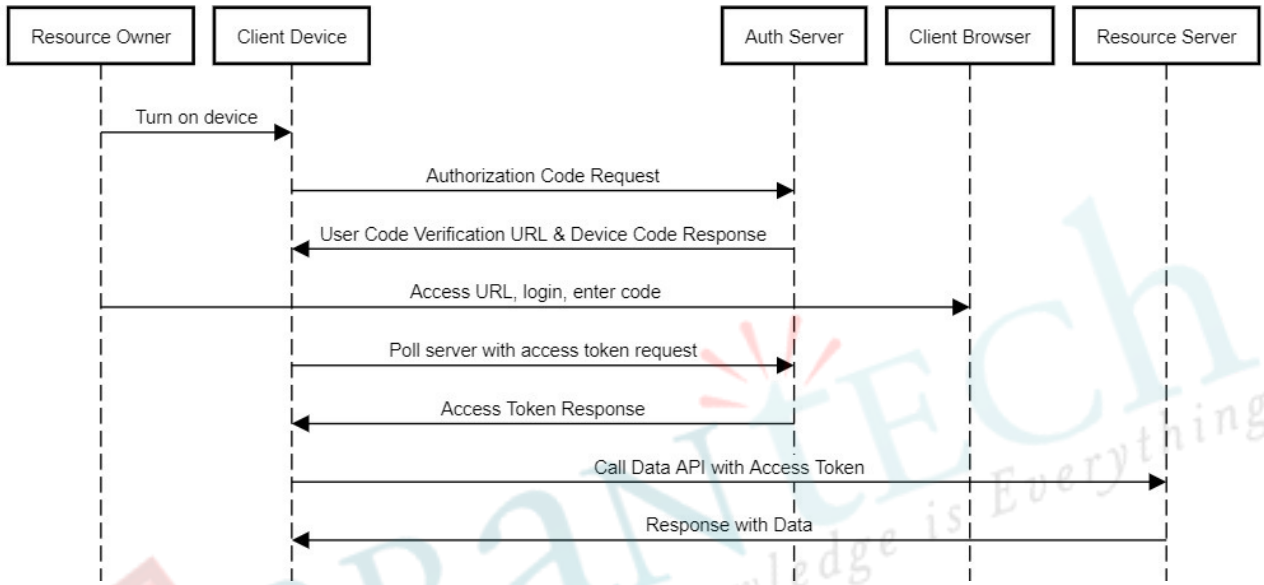
Example: Salesforce App



OAuth & SSO Flow diagrams

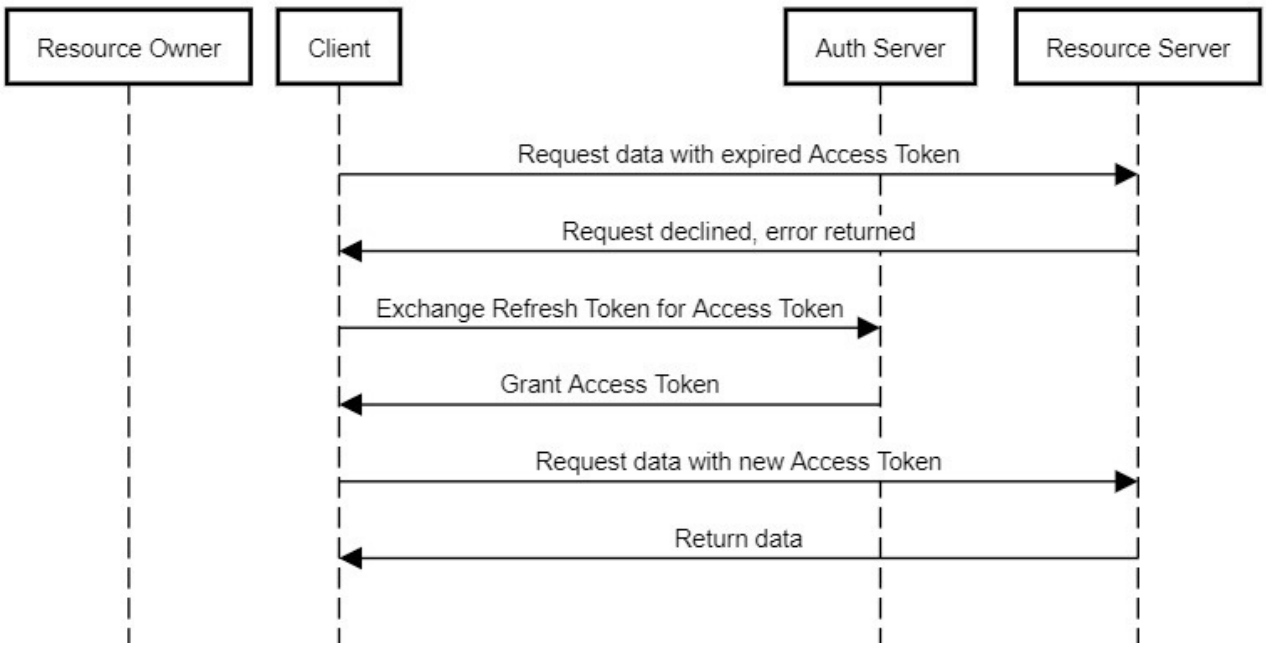
Device authentication flow

The OAuth 2.0 device authentication flow is typically used by applications on devices with limited input or display capabilities, such as TVs, appliances, or command-line applications. Users can connect these client applications to Salesforce by accessing a browser on a separate device that has more developed input capabilities, such as a desktop computer or smartphone.



Refresh token flow

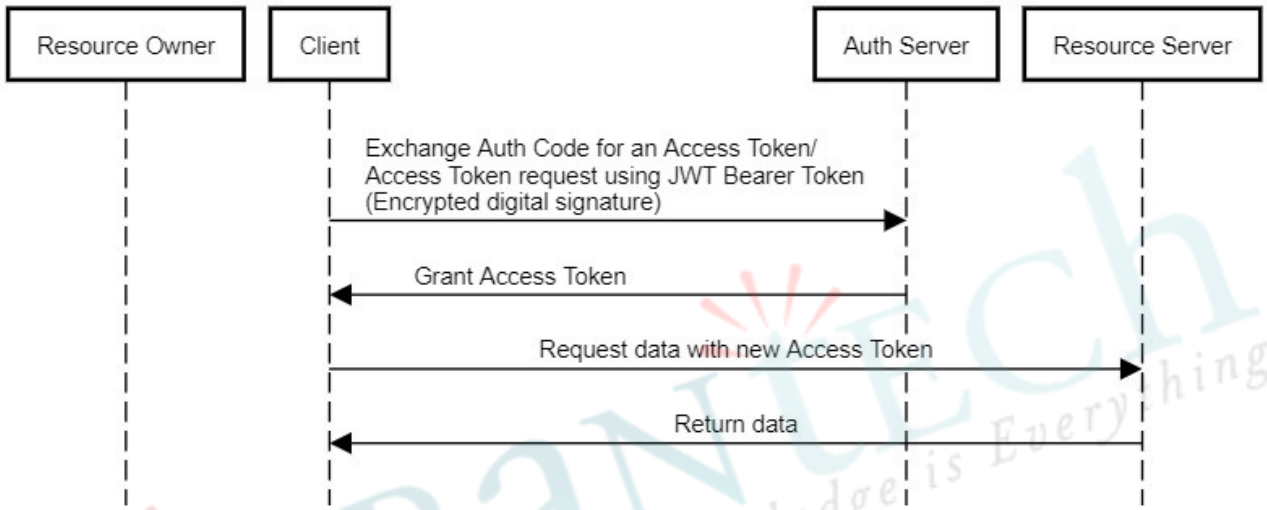
The OAuth 2.0 refresh token flow is for renewing tokens issued by the web server or user-agent flows.



OAuth & SSO Flow diagrams

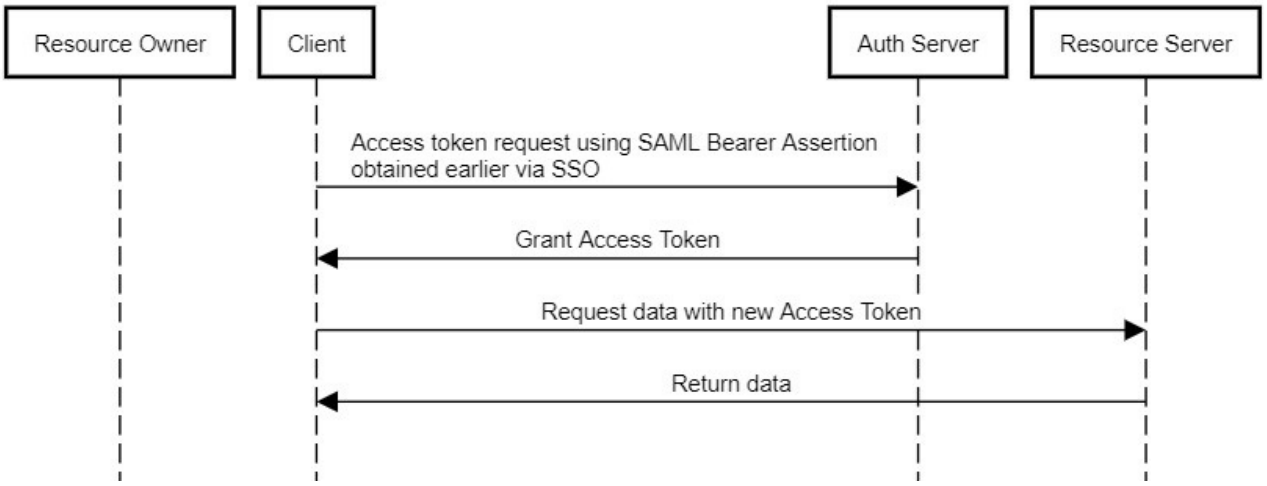
JWT Bearer token flow

JSON Web Token (JWT) is a JSON-based security token encoding that enables identity and security information to be shared across security domains.
The OAuth 2.0 JWT bearer token flow defines how a JWT can be used to request an OAuth access token from Salesforce when a client wants to use a previous authorization. Authentication of the authorized app is provided by a digital signature applied to the JWT.



SAML Bearer Assertion flow

A SAML assertion is an XML security token issued by an identity provider and consumed by a service provider. The service provider relies on its content to identify the assertion's subject for security-related purposes. The OAuth 2.0 SAML bearer assertion flow defines how a SAML assertion can be used to request an OAuth access token when a client wants to use a previous authorization. Authentication of the authorized app is provided by the digital signature applied to the SAML assertion.

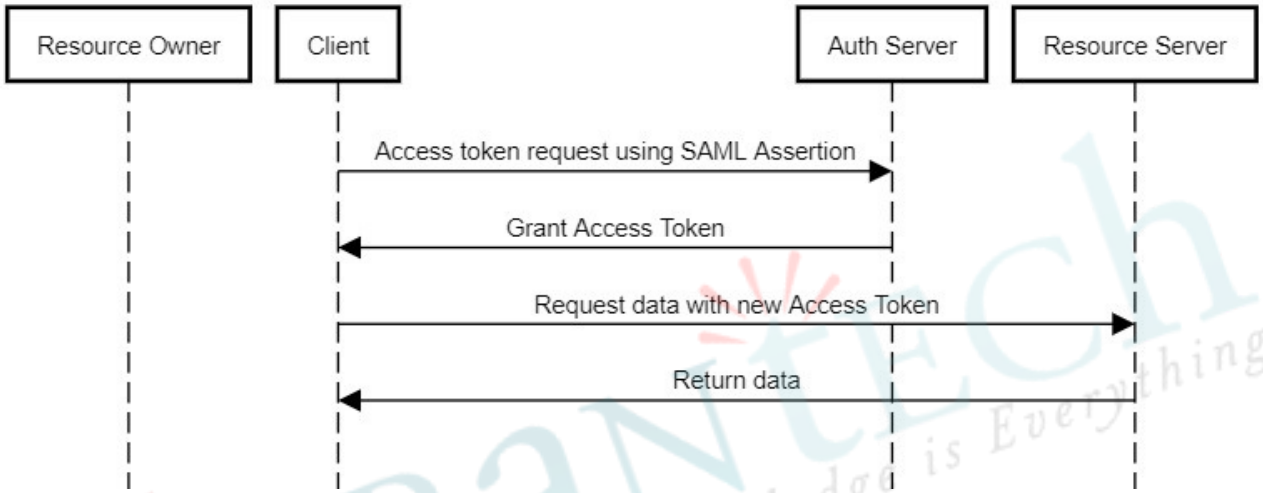


OAuth & SSO Flow diagrams

SAML Assertion flow

The SAML assertion flow is an alternative for orgs that are currently using SAML to access Salesforce and want to access the web services API the same way. You can use the SAML assertion flow only inside a single org. You don't have to create a connected app to use this assertion flow. Clients can use this assertion flow to federate with the API using a SAML assertion, the same way they federate with Salesforce for web single sign-on.

Example: SSO with a company credentials(AD) using Salesforce App which then connects to Salesforce

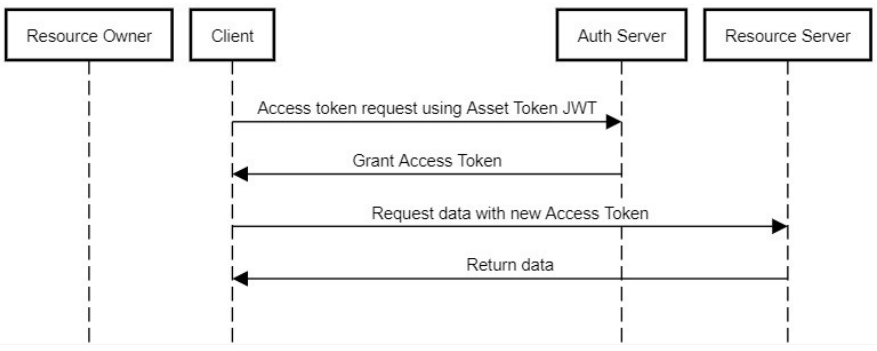


Asset Token Flow

The OAuth 2.0 asset token flow is used by client applications to request an asset token from Salesforce for connected devices. In this flow, an OAuth access token and an actor token are exchanged for an asset token. This flow combines asset token issuance and asset registration, for efficient token exchange and automatic linking of devices to Service Cloud Asset data.

During asset token flow, Salesforce attempts to link the asset token to an existing Asset or to create an Asset.

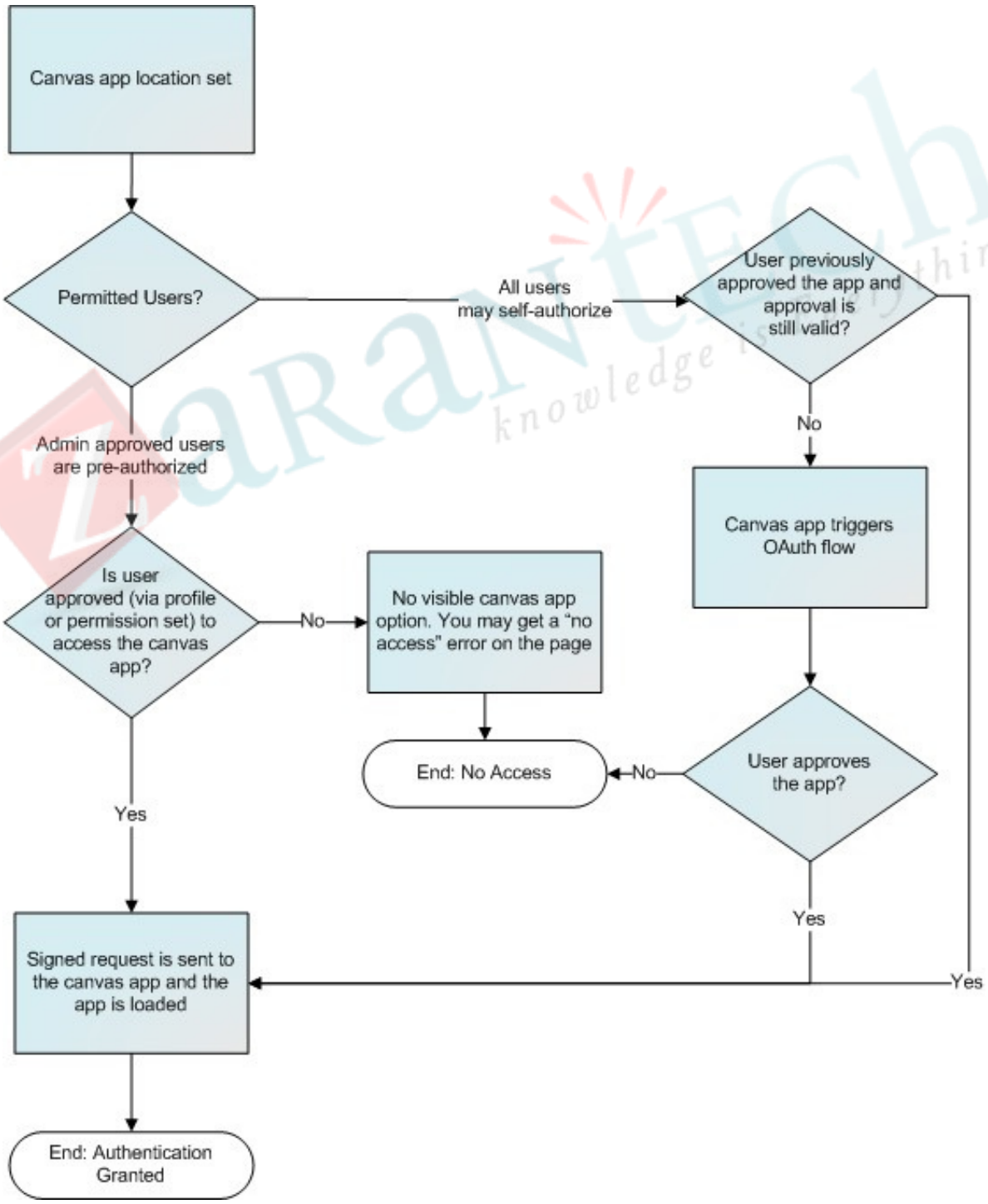
1.If the asset claim contains an ID claim, attempt to link to an existing Asset with a matching ID.
2.Otherwise, if the asset claim contains a SerialNumber claim, attempt to link to an existing Asset with a matching serial number.
3.Otherwise, if the asset claim contains a Name claim, create (register) an Asset.
4.Otherwise, don't link to or create an asset. You can separately link an asset later, via the API.



OAuth & SSO Flow diagrams

Canvas App User Flow - Signed Request

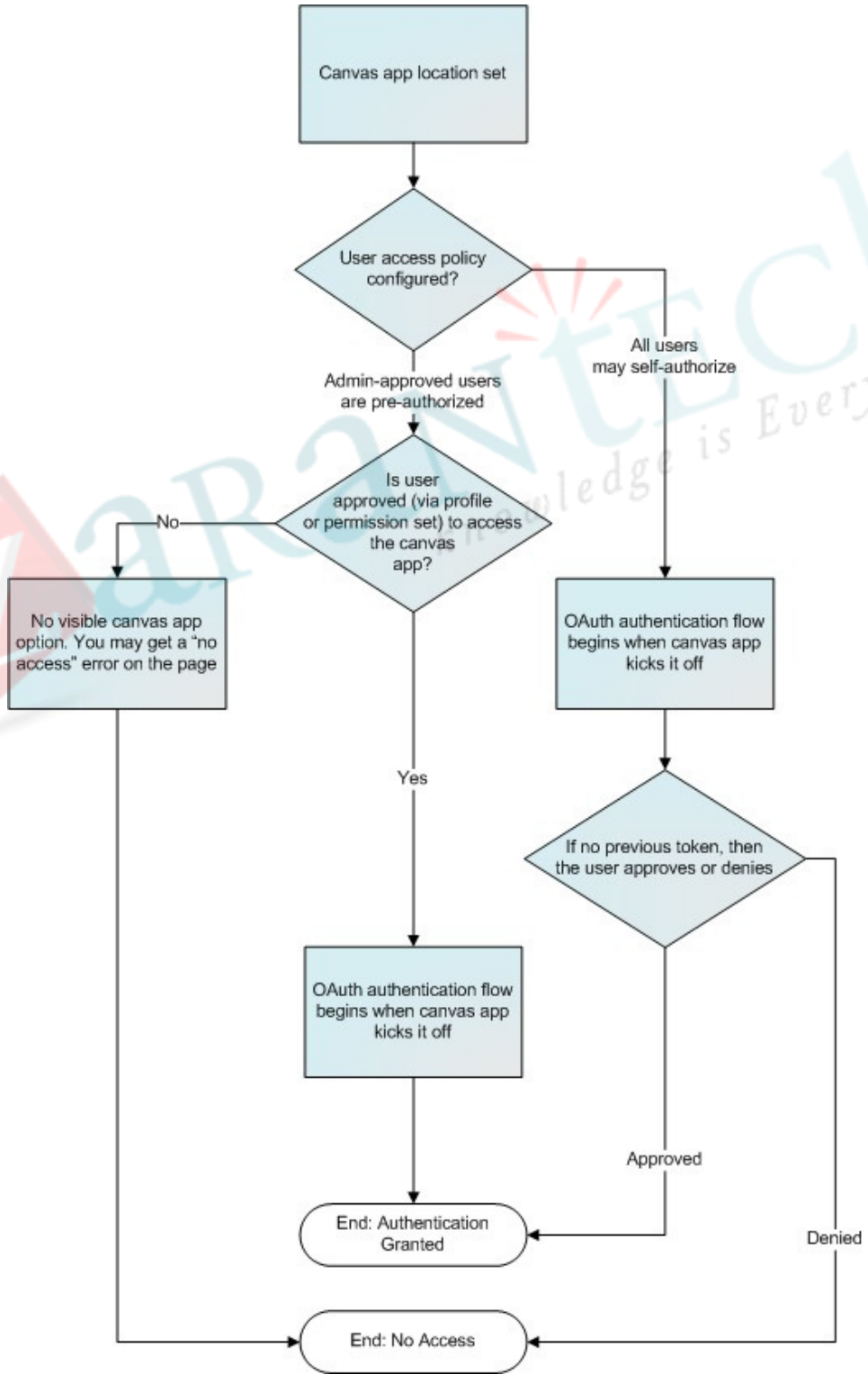
The default method of authentication for canvas apps. The signed request authorization flow varies depending on whether you configure the canvas app so that the administrator gives users access to the canvas app or if users can self-authorize. The signed request containing the consumer key, access token, and other contextual information is provided to the canvas app if the administrator has allowed access to the canvas app for the user or if the user has approved the canvas app via the approve/deny OAuth flow.



OAuth & SSO Flow diagrams

Canvas App User Flow - Oauth

Canvas apps can use the OAuth 2.0 protocol to authenticate and acquire access tokens.
If your canvas app uses OAuth authentication, the user experience varies depending on where the canvas app is located in the user interface and how the user access is set. This diagram shows the user flow for a canvas app that uses OAuth authentication.



OAuth & SSO Flow diagrams

SAML Single Sign-On for Canvas Apps

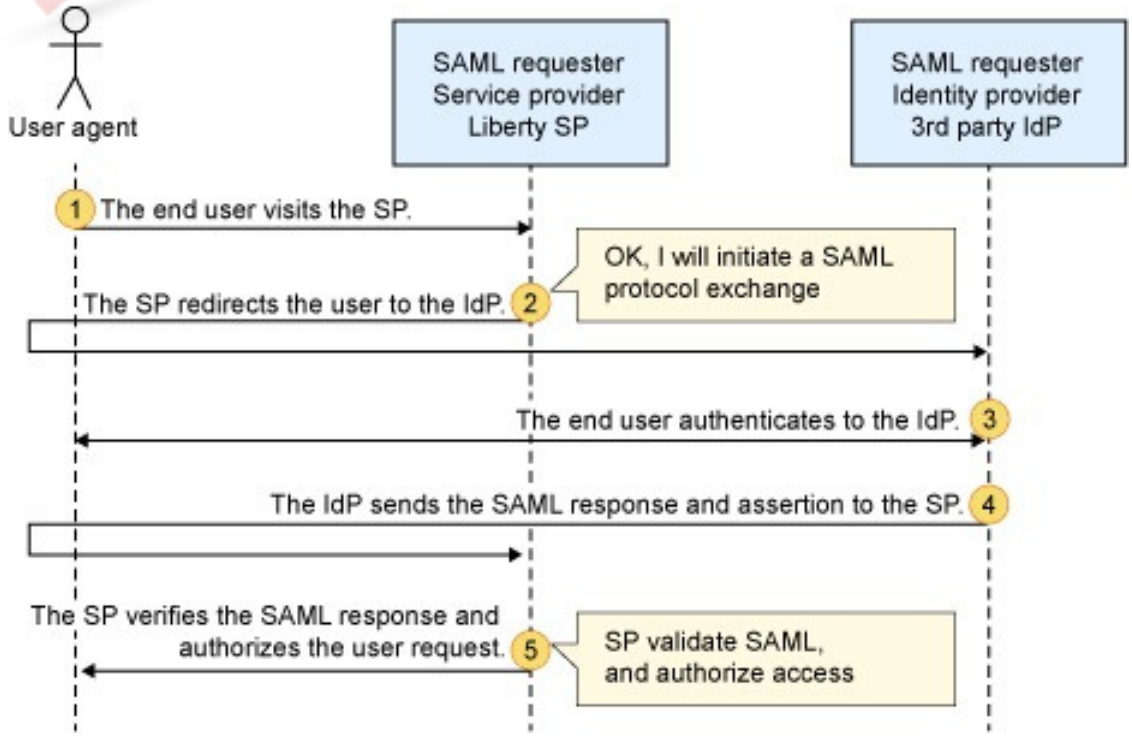
With this feature you can create a canvas app that begins a standard SAML authentication flow when opened by a user. After this process completes, the user is authenticated into your Web application.

For canvas apps that use signed request authentication, two methods that are included in the Canvas SDK enable your canvas app to call into Salesforce to receive a new signed request directly or enable Salesforce to repost the signed request to your Web application endpoint. This results in a complete end-to-end authentication flow.

refreshSignedRequest Method
Returns a new signed request via a callback. After the SAML SSO process is completed, your app can call this method and receive a new signed request. This method is intended for developers who need to retrieve the signed request by using a more client-side JavaScript approach. (The Canvas SDK sends the signed request to your app.)

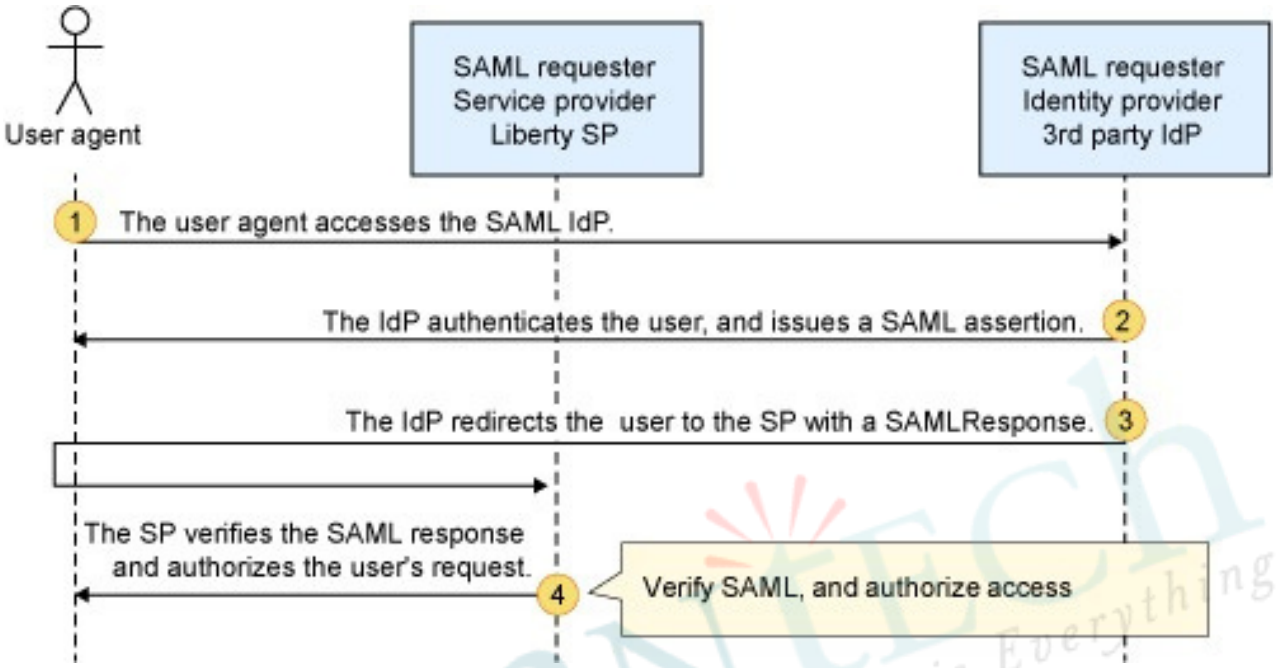
repost Method
Requests the parent window to initiate a POST to your canvas app and reloads the app page with a refreshed signed request. After the SAML SSO process is completed, your app can call this method and a new signed request is sent to your app via a POST. This method is for developers who want to retrieve the signed request using a more server-side approach. (Salesforce POSTs the signed request to your server.)

SP Initiated SAML SSO

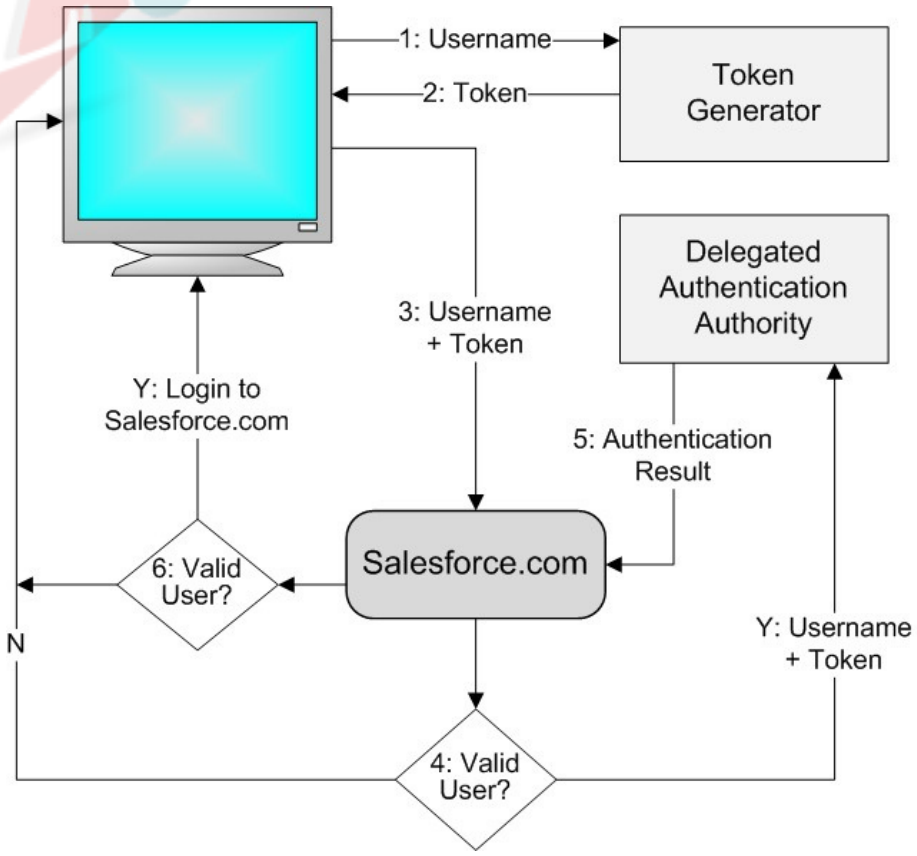


OAuth & SSO Flow diagrams

IdP Initiated SAML SSO



Delegated authentication flow



THANK YOU

Corporate Training Course Catalog
https://bit.ly/salesforce-course-catalog

Salesforce Learner Community
https://www.linkedin.com/showcase/salesforce-learner-community/

Get any Salesforce Video Training
https://zarantech.teachable.com/courses/category/salesforce

Phone/Whatsapp: +1 (515) 309-7846
Email: info@zarantech.com
www.zarantech.com