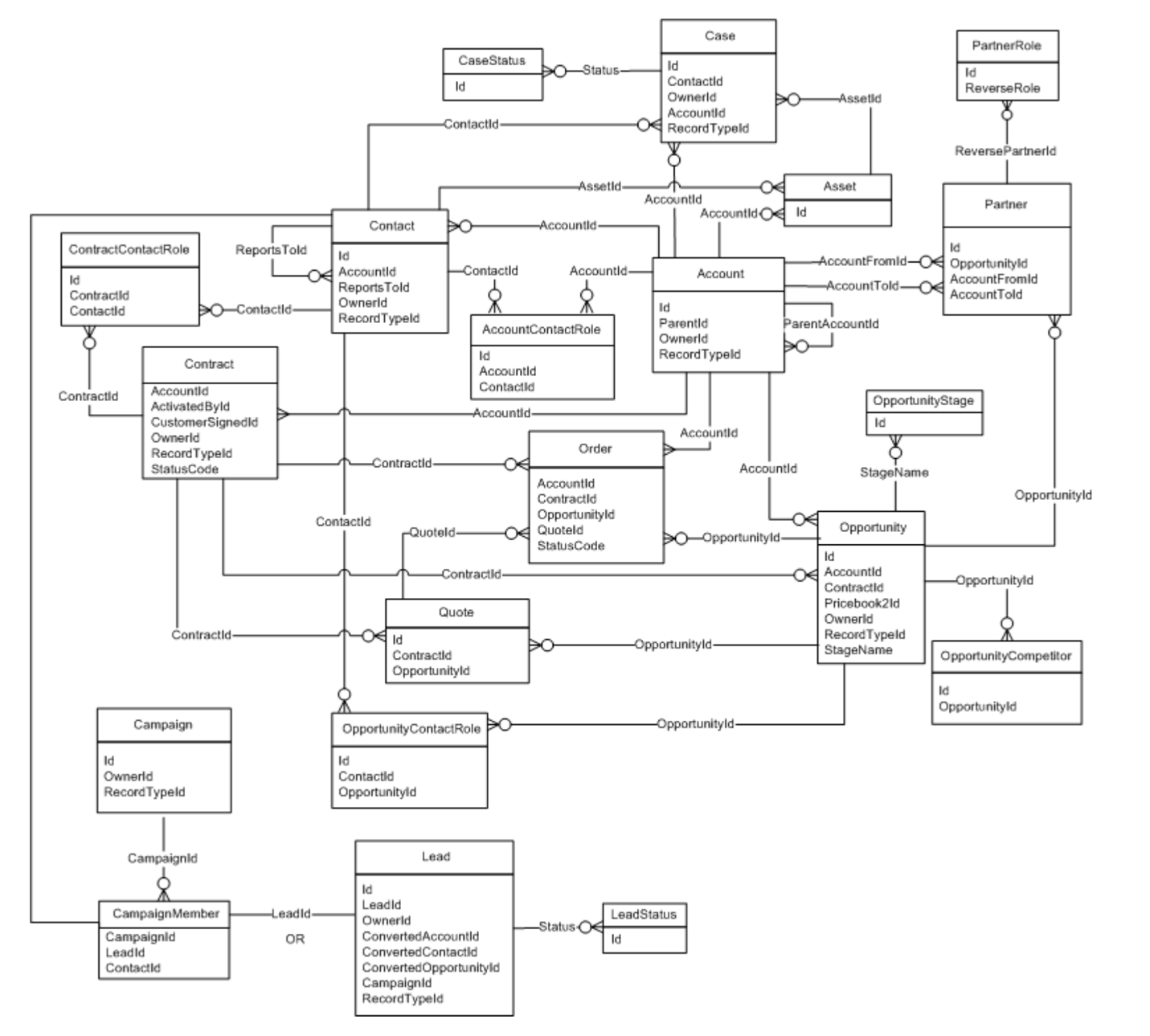
**Salesforce.com Overview**

**What’s Salesforce.com?**

Salesforce.com (SFDC or SF) is a CRM software and we usually use it to implement customer relationship and how to stay the relationship. There are 3 important clouds in Salesforce.com, Sales Cloud, Service Cloud and Marketing Cloud. They represent the process of pre-sales, sales and after sales.

Sales Cloud is focus in how to convert the Lead to Opportunity, Contract. And it provides the report and dashboard to view the data or process directly.

Sales Model:



Service Cloud is focus in how to follow and stay the relationship, and also provide some toolkit, such as Call Center, Live Agent to improve the quality.

We have to know Sales Cloud and Service Cloud are built on Force.com. Force.com is a platform. It provides powerful API to process task or integrates with what you what.

**How to do development in Force.com?**

**Apex Language**:

Common use:

* **Batch**: Related interface: Batchable, Stateful, AllowsCallouts, BatchableContext

Batch usually to process big data, it contains 3 methods, start, execute and finish. You have to know the 3 methods is not in 1 scope, they have their own scope, so the limitation is calculated separately.

* **Schedule**: It defines schedule apex class to run so that can stay data correct in org. It usually uses with Batch tighter. We can schedule it how long to run in page (Days level). In special situation, we can use it in apex code to achieve shorter time calls (Hour level is the best way, do not try the minute or second level).
* **Queuable**: A way to calls in async. And it will not delay while it’s called. And we can monitor it just like jobs. There is higher limitation in it but it will limit with apex call during transaction.
* **Future**: It’s an annotation. And it will be most useful during development. But we need to know it does not provide the promise of the result. And also, you can increase the limitation if you need.
* **Database**: It’s a namespace in sf, just provide several methods to use which can more customize. Usually, we can use key words insert, update, delete directly, but in some special situation, we need more powerful and customize methods, so that we need to use Database namespace.

**VF page**: Visualforce pages are the top-level container for custom apps built with Visualforce. Create Visualforce pages by adding Visualforce components (standard or custom), static HTML markup, and CSS styles and JavaScript to the page.

VF page is a standard MVC mode, it easy to develop.

Static Resource: it usually used for vf, such as js/css/images file.

**Lightning page**: A Lightning page is a custom layout that lets you design pages for use in the Salesforce mobile app or Lightning Experience.

It’s deferent with VF page, not only mode, but also the experience. Lightning is more modern, and it route more quickly than VF page.

**Workflow**: Workflow lets you automate standard internal procedures and processes to save time across your org. A workflow rule is the main container for a set of workflow instructions. These instructions can always be summed up in an if/then statement.

**Process Builder**: Many of the tasks you assign, the emails you send, and other record updates are vital parts of your standard processes. Instead of doing this repetitive work manually, you can configure processes to do it automatically. Process Builder helps you automate your business processes and gives you a graphical representation as you build it.

**Process Builder vs. Workflow**

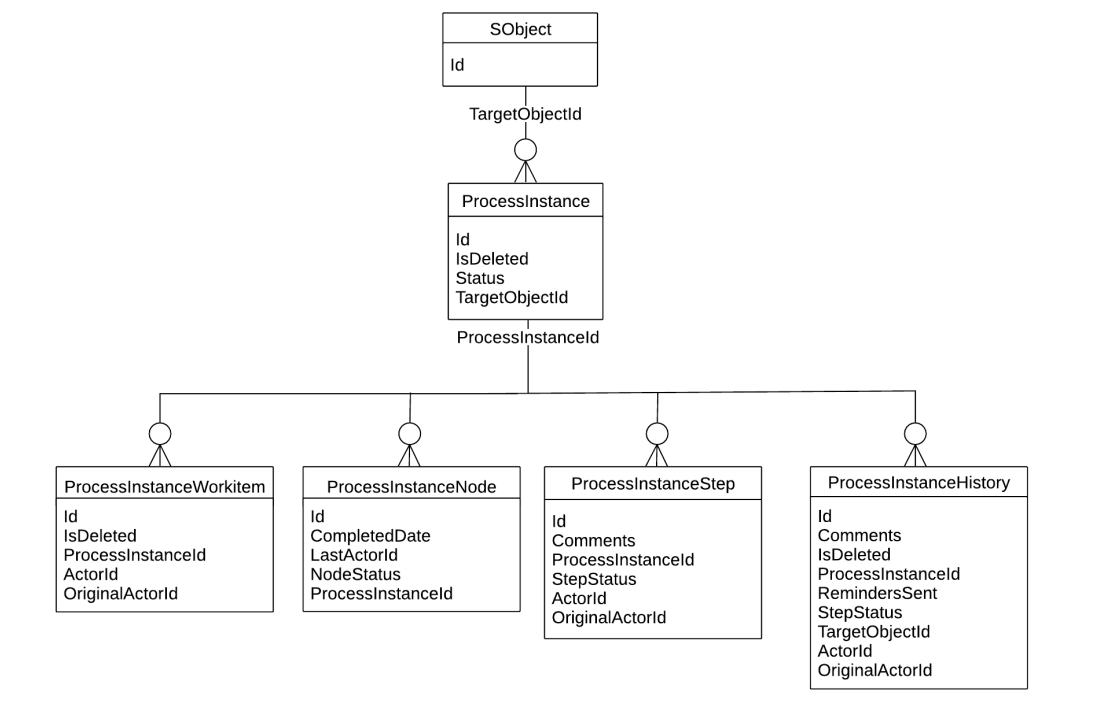
Use the more powerful and flexible Process Builder to perform the same actions as Workflow. With Process Builder, you can:

* Create a record of any object type
* Update any related record—not just the record or its parent
* Use a quick action to create a record, update a record, or log a call
* Invoke a process from another process
* Launch a flow—you can’t schedule this action with workflow
* Send an email
* Post to Chatter
* Submit a record for approval

**Trigger**: Apex triggers enable you to perform custom actions before or after changes to Salesforce records, such as insertions, updates, or deletions. There are 7 ways to handle the sobject: before delete, before insert, before update, after delete, after insert, after update and after undelete.

Trigger is allowed to more in single sobject, but it’s not a well way. Usually we need to Trigger Management to manage the trigger and handler to finish the business flow. And we can use it with custom settings to implement which step/flow need to run or in-active. It’s very useful in development work.

**Approval**: Approvals take automation one step further, letting you specify a sequence of steps that are required to approve a record. Also, we can use below data model to customize.



**Useful unit**

**Custom Settings**: Custom settings are similar to custom objects. Application developers can create custom sets of data and associate custom data for an organization, profile, or specific user. All custom settings data is exposed in the application cache, which enables efficient access without the cost of repeated queries to the database. Formula fields, validation rules, flows, Apex, and the SOAP API can then use this data.

Usually, we use it to store mappings and some configuration. And there are 2 types: List and Hierarchy.

**Custom Labels**: As you know, sf supports multiple language. SF will translate the standard components to fit the current language, but not include the vf/lightning page. Face the situation, we need to use custom labels to complete the challenge. For the sobject and fields and so on or the tab and application, you have to use ‘Translate Workbench’ and ‘Rename Tabs and Labels’.

**Oauth2.0**

Sf implements the Oauth2.0 protocol, so we can use it to integrate with other system easily.

**Connected App**: It defined what’s scope with sfdc for third party to call, and it provides related consumerKey(Client Id) and consumerSecret(Client Secret) .

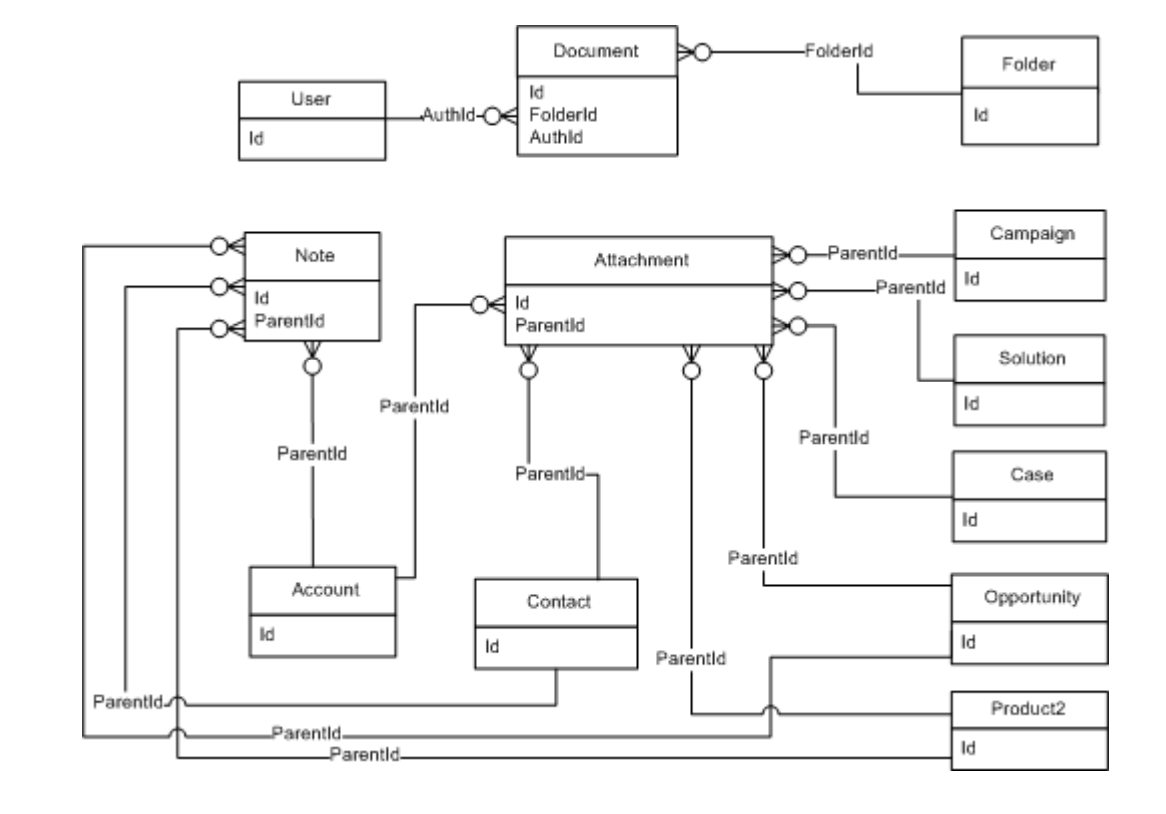
It contains 3 ways:

* Web Server:
* User Agent: The user-agent authentication flow is used by client apps (consumers) that reside on the user’s device or computer. It’s also used by client apps running in a browser using a scripting language such as JavaScript. These apps can protect per-user secrets. But, because the apps are widely distributed, the client secret can’t be confidential.
* Username Password:

**File**

* **Attachment**: It’s the default file mode with sobject store.
* **Document**: It’s not used for sobject. Just for personal file and use for the application Logo Image.
* **Note**: It’s the note in salesforce, and it used with attachment together.
* **ContentDocument**: It also calls Salesforce File. It just like a enhance attachment. It has all of functions with attachment and It also allow owner to share it inner or outer, outer mode it will give you a link to send others to quick view or download.

Above all, attachment&note and ContentDocument is depending on how your org’s preferences are set.



**Test Class**

Any Apex/Trigger codes have to run test before it’s deployed to production environment. The code coverage is not lower than 75 percent. Be attention, the 75 percent means whole org. Usually, we need the code coverage reaches 90~95 percent. And the test class is easy to write, it’s same with generic class just special in annotation ‘@isTest’, and you have to know, the test class stays in different space with generic code. It means that special space is freed when the test is complete and does not affect any data during the test.

**Deployment**

* **Change Sets**: the most common useful way to deploy
* **Third-party tools that use Metadata API or Tooling API**: such as sublime, Force.com IDE and so on. It’s very useful when we need to init sandbox or test some components quickly.
* **VS Code with Salesforce DX plug-ins**: It also use metadata API to complete. And DX is built in CLI, a new way to develop with Force.com and it easy to process version control, such as Github.
* **SOAP API**: It also use metadata API to complete.