

---

**ABSTRACT**

Cloud computing provides lot of new ways to build and run applications that are accessed over the Internet as utilities, rather than as pieces of software running on client or server machine. Business applications, like customer relationship management (CRM) are promoted to be delivered via a Web browser in this paper, rather than delivering it as “traditional software”. As with the Internet, applications that run in the cloud have grown ubiquitous that almost every business user interacts with collaboration CRM business tools such a Web conferencing application, sales system. In this paper, we will implement optimized Sales CRM architecture on cloud.

**KEYWORDS:** Multitenant, Virtualization, Lead, Account Opportunity, Cloud Indexing Schema, Cloud Object, Cloud Data Model.

---

**INTRODUCTION**

The cloud based platforms [1] is making the delivery of application functionalities more efficient. Applications that run in the cloud looks more like websites and less like platforms. Such platforms grow efficient usage of Application Programming Interfaces (APIs), code libraries, and even programming models. Collectively, these new kinds of development technologies can be thought of as platforms to run apps in the cloud. We are implementing the Sales cloud version of the CRM on the cloud. We would first be discussing the CRM functional modules in of the solutions, then Design pattern of the solution followed by architecture illustration. Various features of the Security model will be elaborated. We will then walkthrough over the implementation guide of the solution. We will discuss the Benefits and conclude the paper at the last. We will realize that how important it is to migrate the CRM solution from On-premise model to Cloud infrastructure. CRM software should be highly customizable and scalable, letting businesses to increase actionable information of customer through back-end analytical engine, creating and nurturing business opportunities with predictive analytics, thus simplifying customer service based operations based out of customer's known history and prior interactions with business. This is mostly achievable with this cloud solution.

**CRM MODULES**

We are designing the Sales Cloud version of Cloud CRM platform in salesforce.com in this paper. By Sales Cloud, we mean, designing the services for the Sale people in the CRM business. It is designed around Campaigns, Leads, Contacts, Accounts, Opportunities, Agreement, Products, and Price Engines. It facilitates the set-up to perform start-to-end of complete sales processes. This design satisfies the general Cloud SaaS features such as Mountable Metadata Platform, Fast Application Development [2] & Customization, Robust Multitenancy and flexible API support.

*Lead Management:* This module helps the business people to derive decisions about where should company invest for the marketing and promotion of their products and show the impact of each marketing activities on company's target line. It facilitates the effective conversion of leads into actual business – Account, Contact and importantly opportunities.

**Contact Management:** This module facilitates the 360-degree comprehensive view of the customers, which includes tracing and maintaining activity history, all business communications.

**Opportunity Management:** This module manages the actual sales deals. It facilitates the intuition of business, creation of quotes and management of information related to the deal.

**Agreement Management:** This module is architected for enterprise use across all divisions providing with tailored views and system configuration based on their unique contract types.

**Supporting Sub-Modules:** The critical modules such as: Reports and Dashboards, Sales Forecasting, Sales Collaboration, Email Integration, Enterprise Territory Management are supporting the effective functioning of modules.

## DESIGN

The Core design principle is the metadata-driven software architecture [3] that enables multitenant applications to meet the high demands of business. It means to share IT infrastructure and resources securely and cost-efficiently. Cloud prominently uses multitenancy technology to share IT resources securely among tenants which actually means businesses, organizations, etc. and multiple applications that are based on cloud technology. Some clouds prefer virtualization-based[4] architectures to distribute tenants. The following sections provide you with an overview of key aspects of the platform's design.

**Platform's Kernel and Virtualization:** All the important operations in the traditional application's DBMS such as caching mechanisms, query optimizer, Memory Catalogs, and application development features are designed to support single-tenant applications and be executed on top of a client or host operating systems. Multitenant clouds database services combine several different persistence technologies that include custom-designed relational database schema, which are distinctively planned for clouds and multitenancy.

**Metadata-Driven Kernel:** Metadata-driven architecture is designed to achieve the distinguished layers of tenant data, the metadata that describes each application and compiled runtime database engine (kernel) [5]. It supports continuous customization and auto-upgradation of all and different application without breaking services are in use. Objects referred as tables and consist of stored procedures, database triggers, abstract constructs and set of fields as per the as metadata.

**Polyglot Persistence:** This model has transaction database engine and relational database engine runs in parallel to make unique data model for efficient multitenancy. It also uses other persistence technologies [6] to deliver fast, scalable performance for various operations. This separation of operations between the search service and transaction engine allows applications efficiently process records without the overhead of indexing and searching.

**Multitenant Indexes:** This platform uses a single flex column to store the data of many fields with varying structured datatypes whereas Traditional database systems rely on native database indexes to quickly locate specific rows in a database table that have fields matching a specific condition. Platform's custom query optimizer uses internally maintained table structure to aid improved related data access processes, when an inner system query contains a search parameter that references a structured field in an object. At runtime, the query optimizer automatically builds data access operations so that the optimized SQL statement filters on the corresponding case-folded data values, which in turn corresponds to the literal provided in the search request. This platform uses the pivot table dedicated to indexing to support uniqueness for custom fields. The purpose of this table to perform a comparatively simple query that retrieves the Name of each referenced record for display.

## ARCHITECTURE

The previous section has illustrated the design of solution to store metadata and data. This section explains the architecture which based metadata based. Application Development in the Multitenant Architecture can happen as:

**Browser-Based Development:** Developers can shape server-side application components [7] using this platform's Web browser-based development environment. It is a declaration oriented point-and-click UI that supports all aspects

of the application processes such as creation of an application's front –end interfaces, data model ,Sharing and Security models, logics in stored procedures and triggers .

**APIs:** This platform supports standard and open APIs that developers can use to build various applications. The important feature of the APIs are : Manipulate metadata that describes an application schema, CRUD operations - Create, read, update, and delete business data, Expose a stream of data in a scalable and secure way, Asynchronously Bulk-load a large number of records, Embedding application with social networking functionality

**Query Languages:** Applications can custom the powerful database queries using the Salesforce Object Query Language (SOQL). Like Structured Query Language (SQL), SOQL[8] permits to declare the source object, conditions for selecting rows in the source object, and a list of fields to retrieve. Applications can influence this search engine using the Salesforce Object Search Language (SOSL)[9] to perform text searches. SOSL enables developers to search text for multiple objects in parallel.

**Query Processing:** This platform maintains a complete set of optimizer statistics to deliver sufficient statistics for defining optimal query execution[10] plans in a multitenant system. An optimizer also relies on internal security-related tables that maintain information about the security domains of system users.

**Search Processing:** This system is designed with search engine that is separate from its transaction engine. The search engine receives data from the transactional engine, with which it creates search indexes. Search engine maintains the indexing process by copying synchronously modified data pieces into an internal table. The search engine separates indexes for each organization automatically. Force.com maintains an MRU (most recently used) cache[11] of recently updated rows to avoid unexpected search results originating from old indexes.

**Bulk Operations:** Transaction-intensive applications generate less overhead and perform much better when they combine and execute repetitive operations in bulk. If the engine detects errors during any step, the engine rolls back offending operations and all side effects, removes the rows that are responsible for the faults, and continues, attempting to bulk process the remaining subset of rows.

## SECURITY

The security model [12] should not only define ways to shield data but also protect it from unauthorized access internally and externally.

**Platform Security:** The first aspect is users, watching how users are authenticated, monitoring the network-based security that controls the IP ranges from which a user can access the application. The second aspect is programmatic security[13]. Any authorized user that needs to log in to the platform, does so through a Web services SOAP interface. The third aspect is the Salesforce.com platform security framework, which developers can develop to define the access permissions to authenticated users within the cloud CRM application. Salesforce.com has the extensive security standard which is SysTrust SAS 70 Type II.

**User Authentication:** Users are authenticated on the login page. After logging in, the user will have access to CRM application on Cloud system. There is one user authentication mechanism designed to implement single sign-on[14] within an integrated set of applications.

**Network-based Security:** In this cloud model, network-based security means to limit where users can login from and when. Salesforce.com provides two options for limiting access based on the user's network location. It facilitates allowing from users from trusted locations [15], but challenge users when they try to login from new and untrusted locations.

**Auditing:** A final aspect of this security is auditing. This platform does support the auditing of metadata and system changes. Administrators can keep track of the last 20 logins to the CRM application; can access the important data such as login data, which includes IP addresses, browser types and so on. Data auditing[16] is achieved by checking the Track Field History checkbox while creating the field.

**Platform Security Framework:** System permissions, which is about permitting users in the application with the ability to perform various actions and access parts of the cloud platform. Component permissions, which implies granting access to different components defined with the CRM. Record-based sharing, which means granting access privilege to individual records in an object. The following sections examine each of these aspects of security.

**System Permissions:** System permissions are the privileges or the permissions granted to all authorized users, providing capabilities to the environment and set up. Every user is assigned to one profile and System permissions are granted to profiles.

**Administrative Permissions:** The broad system access and permissions are typically established by administrators to other system administrators. Such permissions are around Managing users, Viewing and Setting up the system Configuration

**Data:** The important permissions around the data manipulation are Modify All Data, View All Data, Edit Read-Only Fields, View Encrypted Data, Weekly Data Export etc.

**Component Permissions:** The components such as Applications, Tabs, Record types, Apex classes, and Visualforce pages can be allowed or disallowed based on profile. **Record-based Sharing:** The finer level of access control of the user data is maintained through record-based sharing.

**Organization-wide Defaults:** The setting can be private, which only permits an owner to access the record, public read-only, which permits everyone to read the records, or read write, which permits all users to read and write all records.

## IMPLEMENTATION

Any application development requires proper planning, which is generally carried out per the approved SDLC practice [17] followed by the implementation team. Similarly CRM on cloud can be implemented by planning few steps.

**Step 1. Prepare** - Migrating the CRM to cloud may require the initial enterprise investment and its success depends upon the clear implementation plan [18]. The blue print should be planned around the features: 1.) Building team that is proficient with Salesforce and Cloud Technology Skillset. 2) Clear definition of company's objective and their prioritization with respect to implementation. 3) Development of Execution plan importantly with process flow or work flow diagram 4) Develop the CRM functionality in the Cloud as the business requirement.

**Step 2. Set Up Salesforce CRM** - This section discusses the next steps about developing important features in the Salesforce CRM Set up such as profile definition as per the business, Access and permissions for the user. Data Access can be referred from the section above.

**Step 3. Engage** - Focusing on training of business users to derive adoption is important. Different kinds of users will have distinct goals and perform different tasks in the cloud CRM. So their training should be Relevant to their goals.

**Step 4. Extend the Cloud CRM Solution** - The extensions and enhancements can be planned around the additional functionalities, Integration to other different systems for effective business communication, Provision and development of Productivity Tools, Data Quality Management tools.

**Application Logic Development[19]:** The platform enables to implement the CRM business logic in two ways. 1. Point-and-click features – Wizard like use point-and-click features to implement app logic which is easy to implement and no additional backend code and unit tests are required..2. Apex Code - The power of Apex Code for implementing custom business logic should be considered when the standard point-and-click features can't meet the specific business requirements.

**Process Automation Suite[20]:** The platform's automation tool can be chosen per the needs and type of business process that needs automation.

**Process Builder:** This tool helps developer to easily automate business processes using a convenient graphical representation of the business process as it gets designed.

**Visual Workflow:** Visual Workflow - The product that includes the process of designing, managing, and running flows.

**Time-based Workflow:** It can be used to automate standard internal procedures and processes to save time across the CRM.

**Routing Approvals:** An approval process is an automated process that can be used to approve records in Salesforce.

**Apex basic and Database:** Apex is a programming language that uses Java-like syntax and acts like database stored procedures. Apex enables developers to add business logic to system events, such as button clicks, updates of related records, and Visualforce pages. The features of Apex are: a.) Apex is saved, compiled, and executed on the cloud server, which means it is hosted. b.) Apex is object oriented language, supports classes, interfaces, and inheritance. c.) Apex validates references to objects at compile time. d.) Apex provides direct access to records and their fields, and provides statements and query languages to manipulate those records.

e.)Data focused Apex provides transactional access to the database, allowing you to roll back operations.

Because Apex is a data-focused language, hence managing records is easy with Apex DML operations. *SOQL Queries*: This platform provides the Salesforce Object Query Language or SOQL that can be used to read saved records. *SOSL Queries*: Salesforce Object Search Language (SOSL) is a Salesforce search language that is used to perform text searches in records. *Apex triggers* allow performing custom actions before or after events such as insertions, updates, or deletions to records in Salesforce. Apex provides trigger support for managing records just like database systems support triggers. The *Apex testing* framework enables developers to write and execute tests for Apex classes and triggers on the Force.com platform.

**Visualforce[22]**: Visualforce is a Web-based framework that lets UI designers develop sophisticated and custom UIs quickly for Platforms supported desktop and mobile applications. It is a blend of native Visualforce markup and standard Web development technologies like HTML5, CSS, JavaScript, and jQuery, which makes easy and rich UIs for any application. To develop new functionality and build completely new applications, Visualforce can be used and extended with Salesforce's built-in features. Developers create Visualforce pages by composing components using markup, and optionally adding HTML and CSS styling elements. Like HTML, Visualforce accepts JavaScript and related JS frameworks allowing you to create animated, responsive and visually rich user interfaces.

**Integration[23]**: Integration helps enterprise to have many and different applications integrated. Integrating separate but related applications helps organizations achieve greater levels of efficiency, quality, and operational consistency. *Security Integration* means Integration of authenticated mechanisms across applications to improve the user experience and minimize user administration. *User Interface Integration* means combining the UIs of two or more applications to create composite and collaborative information exchange. Business Logic Integration means Extending related business logic from one application to another helps to implement complete end to end business processes. *Data Integration*: Integrating applications at the data layer is the most common and applied scenario. Multiple applications written in different programming languages can all use an open API and manage related data in one shared database.

## BENEFITS

According to Forbes, large companies have deployed CRM on cloud and it helps them to achieve Effective identification of new sales leads, Effective proposal generation, Effective configuration tools and price quotations, Effective agreement closures. Cloud CRM helps business to go to mobile seamlessly easy.

- a) *Easy conversion to Mobile CRM*[24], which does not just increase use of the software, but also the productivity of the workforce.
- b) Cloud permits the *access from multiple devices*.
- c) *Digital marketing* achieved via cloud CRM solutions responds to increase Return on investments on investments made by the business.
- d) *Lower up-front CRM cost*[25] - Instead of the capital expense of traditional on-premise CRM systems, a cloud subscription is treated as an operational expense by subscribing to a cloud service.
- e) *Faster returns from CRM* - Such solutions let users quickly begin personalizing the database and start benefiting from CRM to create early successes.
- f) *Reduced IT overhead* – On-premise CRM database administrators need to back-up data, apply patches and configure other CRM updates in addition to the other overheads of running a CRM server including power. IT administration is immediately reduced with a cloud deployment.
- g) *Easy integration* with other Cloud technologies.
- h) *Predictive Cash-Flow* through Auto-upgrades: On-premise CRM users need to update their software periodically to preserve compatibility with other technology. Cloud CRM is easier to budget for as they remove the variable of future upgrade costs.
- i) *Hardware Infrastructure*- The considerable cost differential on hardware infrastructure between on-site and cloud is probably the most compelling argument in favor of the cloud.
- j) *Advanced Data and Analytics*[26]: Many businesses are turning to cloud CRM for business intelligence. On-premise solutions generally require manual uploading, syncing, and backup, whereas cloud solutions sync in real time. Cloud CRM provides an up-to-date picture of customer data, sales pipelines, invoices, and email.



We have implemented an efficient, secure, scalable and flexible Sales cloud CRM architecture on Cloud using salesforce.com. This security architecture helps you to define how users log in, determining for example which IP ranges are acceptable, what hours of the day are allowed, how long sessions stay active for and so on. It also lets you define programmatic control.

## CONCLUSION

Cloud-based services are gaining attention and acceptance in the IT industry due to improved time to market, reduce capital expenditures, and improved overall competitiveness. Cloud application development platforms are effective because they let businesses develop quick software solutions on demand, and minimize the costs and complexity associated on-premises hardware and software set up.

Salesforce.com's managed approach for the deployment of production applications ensures top-notch performance, scalability, and reliability for all applications. Integral system features—such as the bulk data-processing API; Apex, an external, full-text search engine; and a unique query optimizer—help make dependent applications highly efficient and scalable with little or no effort from developers. Hence it is evident to conclude that we have implemented, configured and deployed critical CRM Sales model on Cloud.

## REFERENCES

- [1] An optimized architecture to Implement CRM using the Salesforce.com by Rohit Parashar, published in IJCESR , Volume-3, Issue-5, ISSN (Online): 2394-0697, ISSN (Print): 2393-8374
- [2] Implementing CRM application in the Cloud by Rohit Parashar, published in IJCEM, Page No. 212, ISBN: 9789383917457
- [3] Cloud Application Architectures by George Reese, Published by O' Reilly Media, Inc.
- [4] Cloud Computing Concepts Technology and Architecture by Thomas Erl, Published by Prentice Hall
- [5] Cloud Design Patterns by Microsoft.
- [6] [https://developer.salesforce.com/trailhead/en/starting\\_force\\_com/starting\\_understanding\\_arch](https://developer.salesforce.com/trailhead/en/starting_force_com/starting_understanding_arch)
- [7] [https://developer.salesforce.com/page/Multi\\_Tenant\\_Architecture](https://developer.salesforce.com/page/Multi_Tenant_Architecture)
- [8] Handbook of Cloud Computing by Furht and Escalant, Published by Springer
- [9] Migrating to Cloud Native Applications Architectures by Matt Stine, Published by O'Reilly Media.
- [10] Cloud Computing Bible by Barrie Sosinsky, Published by Wiley Publishing, Inc.
- [11] Privacy and Security for Cloud Computing by Siani and George , published by springer
- [12] Cloud Security by Krutz and Vines, Published by Wiley Publishing, Inc
- [13] <https://developer.salesforce.com/page/Security?language=en>
- [14] [https://developer.salesforce.com/page/Secure\\_Private\\_Trustworthy\\_Force.com\\_Whitepaper](https://developer.salesforce.com/page/Secure_Private_Trustworthy_Force.com_Whitepaper)
- [15] Cloud Databases: A Paradigm Shift in Databases by Indu and Anu, Published in IJCSI, Vol. 9, Issue 4, No 3, July 2012
- [16] CLOUD DATABASE DATABASE AS A SERVICE by Waleed Al Shehri, Published in IJDMMS ) Vol.5, No.2, April 2013
- [17] [https://developer.salesforce.com/docs/atlas.en-us.api.meta/api/data\\_model.htm](https://developer.salesforce.com/docs/atlas.en-us.api.meta/api/data_model.htm)
- [18] Database Management System as a Cloud Service by Gelogo and Lee , Published in IJFGCN Vol. 5, No. 2, June, 2012
- [19] Salesforce Handbook Kindle Edition by Wes Nolte and Jeff Douglas
- [20] Salesforce CRM: The Definitive Admin Handbook Second Edition Kindle Edition by Paul Goodey
- [21] Development with the Force.com Platform: Building Business Applications in the Cloud (Developer's Library) Kindle Edition by Jason Ouellette (Author)
- [22] Visualforce Development Cookbook Kindle Edition by Keir Bowden (Author)
- [23] <https://developer.salesforce.com/page/Integration?language=en>
- [24] <http://www.forbes.com/sites/louiscolombus/2015/05/22/gartner-crm-market-share-update-47-of-all-crm-systems-are-saas-based-salesforce-accelerates-lead/#5302673a4e6e>
- [25] <http://www.gartner.com/newsroom/id/3056118>
- [26] <https://www.forrester.com/report/Capitalize+On+SaaS+CRM+Solutions+With+Better+Governance/-/E-RES80161>