ISSN: 2583-9888
DOI: doi.org/10.51219/JAIMLD/Naveen-koka/180

URF PUBLISHERS
connect with research world

Journal of Artificial Intelligence, Machine Learning and Data Science

https://urfpublishers.com/journal/artificial-intelligence

Vol: 2 & Iss: 1

Research Article

Developing a Comprehensive Salesforce-Like Platform

Naveen Koka*

Citation: Koka N. Developing a Comprehensive Salesforce-Like Platform. *J Artif Intell Mach Learn & Data Sci 2023*, 1(1), 720-723. DOI: doi.org/10.51219/JAIMLD/Naveen-koka/180

Received: 02 January, 2023; **Accepted:** 18 January, 2023; **Published:** 30 January, 2023

***Corresponding author:** Naveen Koka, USA, E-mail: na.koka@outlook.com

A B S T R A C T

Developing a comprehensive Salesforce-like platform aims to revolutionize application development by integrating key components like authentication, entities, apps, reports and dashboards, and events. This platform streamlines business processes, providing a cohesive ecosystem for efficient data management and dynamic user interactions. By employing modern UI frameworks such as React or Vue and leveraging microservices architecture with languages like Java or Python, the platform ensures scalability and flexibility. Cloud infrastructures from providers like AWS, Google Cloud, or Azure host these components, enhancing reliability and performance. Open-source reporting tools facilitate advanced data visualization and insightful decision-making. The platform's configuration and runtime capabilities enable seamless customization and real-time data processing, while defining expressions and criteria improves data listing, record viewing, and action execution. Continuous feedback loops foster agility and rapid development, minimizing maintenance costs by having a single system to learn and manage. This approach not only reduces operational expenses but also empowers organizations to create tailored solutions for sectors like manufacturing and banking, driving efficiency, innovation, and competitive advantage.

Keywords: Salesforce, Low code app development, UI, Reports & Dashboards

1. Introduction

Developing a comprehensive Salesforce-like platform is a strategic initiative aimed at streamlining application development and enhancing user experience across various sectors. This platform integrates essential components such as authentication, entities, apps, reports and dashboards, and events, creating a cohesive ecosystem for managing business processes. By leveraging modern UI frameworks like React or Vue, microservices in Java or Python, and cloud infrastructure such as AWS, Google Cloud, or Azure, the platform ensures scalability and flexibility. Open-source reporting tools further enhance data visualization and decision-making. This unified approach simplifies system learning, management, and maintenance, reducing operational costs and fostering agility. Ultimately, the platform empowers organizations to build custom solutions tailored to their unique needs, driving efficiency and innovation.

2. Problem Statement

Salesforce began as a low-code app development platform designed to enhance sales operations. Its primary appeal lies in its elasticity and user-friendly configuration, which allows customers to build various types of applications with relative ease. This flexibility has led to a diverse range of apps developed for different use cases within the Salesforce ecosystem.

Given its subscription-based model, Salesforce's costs can accumulate significantly over time, especially for enterprise customers who use the platform for multiple use cases. This financial burden becomes more pronounced as businesses expand their reliance on the platform to support a growing number of applications and functions. Consequently, what starts as an efficient solution for sales operations can evolve into a substantial expense as usage broadens.

As enterprises continue to explore the potential of Salesforce for a variety of applications, the subscription fees can become a significant budgetary concern. The convenience and capabilities of the platform come at a cost, which can escalate as companies scale their operations and integrate more complex and varied applications. This financial impact necessitates careful consideration and strategic planning to manage and mitigate the expenses associated with long-term use of the Salesforce platform.

3. Solution

For enterprises with minimal use cases, developing their own platform can be a cost-effective alternative to relying on subscription-based services like Salesforce. By focusing on five fundamental principles-Authentication and Authorization, Entities, Apps, Reports & Dashboards, and Events-companies can create a customized solution tailored to their specific needs. This approach allows for greater control over functionality and costs, avoiding the financial burden of escalating subscription fees.

Authentication and Authorization ensure secure access control, while Entities represent the core data structures. Apps are developed to handle various business processes, and Reports & Dashboards provide insights and analytics. Lastly, Events facilitate real-time updates and interactions within the system. By building their platform around these principles, enterprises can achieve the necessary functionality and scalability without the ongoing costs associated with third-party platforms.

3.1. Pre-Setup

To develop a platform efficiently, several key technical details need to be addressed. First, Authentication can be streamlined by leveraging the existing company's system, eliminating the need for additional investment in this area. This approach ensures secure access control and simplifies the integration process with the company's current infrastructure.

The user interface (UI) is a crucial component, and technologies like React, Vue, or similar frameworks are ideal for rapid development of UI components. These languages and frameworks offer robust, scalable solutions that facilitate the creation of dynamic, responsive interfaces, enhancing the user experience.

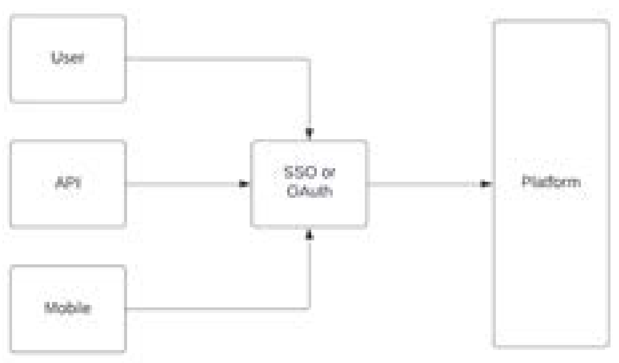
For the backend, implementing Microservices using Java or Python is recommended. These languages are well-suited for building scalable, maintainable microservices architectures. They provide flexibility and reliability, ensuring that the platform can handle various business processes efficiently.

Infrastructure should be hosted on one of the major cloud providers such as AWS, Google Cloud, or Azure. Choosing a cloud provider ensures that all infrastructure components are scalable, secure, and well-supported. These platforms offer a wide range of services that can be tailored to meet the specific needs of the enterprise.

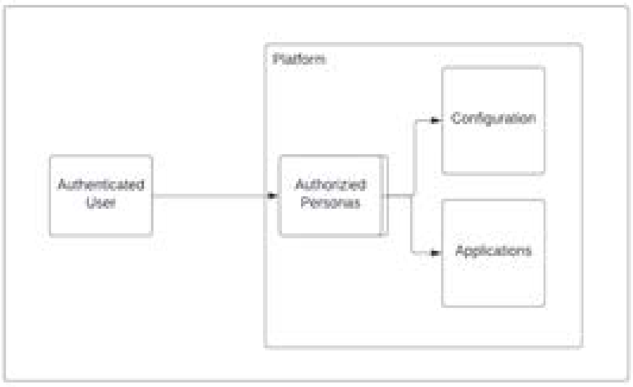
Finally, for reporting, leveraging open-source libraries can provide powerful data visualization capabilities. These libraries can generate comprehensive reports and dashboards based on the provided data, offering valuable insights and analytics without the need for expensive proprietary solutions. This approach allows for customization and flexibility in reporting, ensuring that the platform can adapt to evolving business requirements.

3.2. Authentication & Authorization

To develop an efficient platform, authentication can be streamlined by leveraging existing cloud-based systems like Google or Microsoft, which support standard mechanisms such as OAuth and SAML for secure access control.



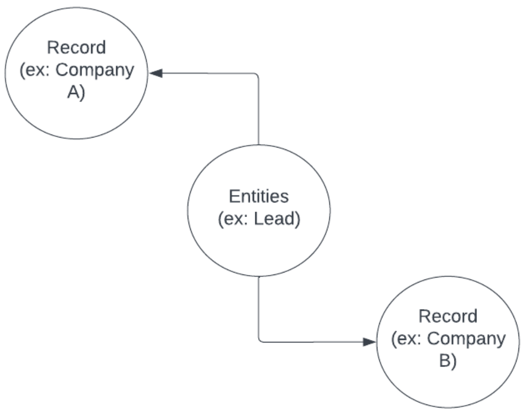
When focusing on platform implementation, a critical area of concern is authorization. This involves defining various user roles or personas and associating specific access rights to both users and applications. By structuring roles effectively, the platform can enforce controlled access across its functionalities. This configuration ensures that only authorized users and applications can perform designated actions, safeguarding data and maintaining operational integrity.



3.3. Entities

Entities serve as the foundational data storage units within the platform. Each entity supports four primary visual representations: listing, creating/editing, viewing, and deleting records.

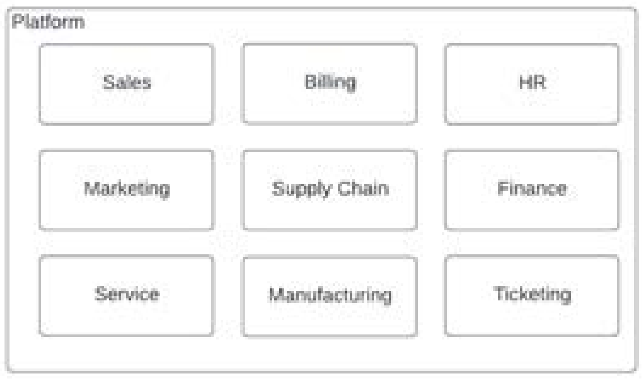
1. **Configuration:** Configuration involves defining each entity by assembling a collection of attributes that delineate its structure and behavior. This phase sets the blueprint for how data will be organized and managed within the platform's ecosystem.
2. **Runtime:** At runtime, the platform dynamically renders entities based on their predefined configurations. This entails developing diverse components that can visually present the data according to its attribute types. For instance, text attributes might utilize text boxes for input, while time attributes might employ date-time components to ensure accurate data representation and interaction. This runtime flexibility ensures that the platform adapts seamlessly to user needs and data handling requirements.



3.4. Apps

Apps are aggregations of entities within the platform, facilitating cohesive workflows and functionality.

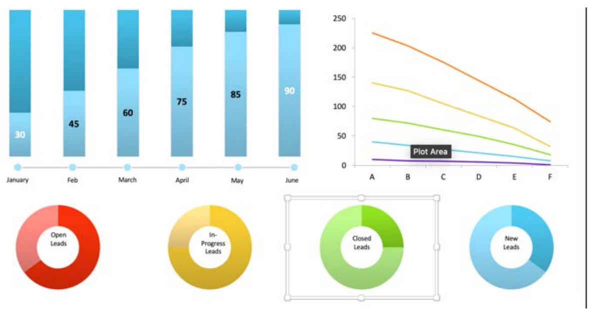
1. **Configuration:** In the configuration phase, apps are defined by assembling and orchestrating various entities throughout their lifecycles. This involves structuring multiple entities for actions like creation and viewing within a unified app framework. Configuration also includes designing the layout of entities and specifying actions related to each entity. For example, defining an action such as "close lead" dictates the behavior when users interact with that specific entity record, ensuring consistent and predictable outcomes.
2. Creating a rich configuration UI enhances the platform elasticity.
3. **Runtime:** During runtime, the platform's engine dynamically renders app interfaces, orchestrates interactions, and interfaces with backend services to store and retrieve data. This operational phase ensures that users experience a seamless and responsive environment where they can efficiently manage and manipulate data within the defined app structures.



3.5. Reports & Dashboards

Reports and dashboards provide visual representations of data, enabling users to gain insights and make informed decisions.

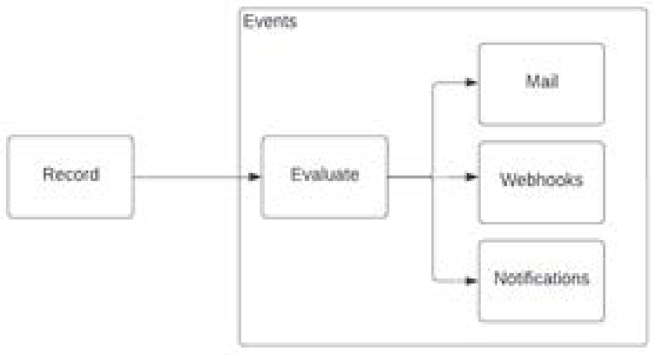
1. **Configuration:** The configuration phase involves setting up rich charts that can effectively display various types of data. Users should be able to configure both the actions and charts within a dashboard, customizing the layout and content to meet specific needs. Additionally, providing the ability to preview charts allows users to see how their configurations will look in real time. Enabling the slicing and dicing of data lets users drill down into details and analyze data from different perspectives.
2. **Runtime:** At runtime, the platform leverages existing services to display the configured reports and dashboards. This ensures that the visual representations are up-to-date and accurately reflect the underlying data. The runtime engine dynamically renders charts and dashboards, providing users with an interactive and responsive experience.



3.6. Events

Events are actions triggered when records are inserted or updated, allowing the system to respond dynamically to changes.

1. **Configuration:** During configuration, users can define the various attributes needed for executing events. This includes specifying the conditions under which events are triggered and the actions to be taken, such as sending emails, triggering webhooks, or generating notifications. This setup allows for flexible and tailored responses to different record changes.
2. **Runtime:** At runtime, when a record is saved, the system evaluates the record against the configured events. If the conditions for an event are met, the system executes the defined actions, ensuring that the appropriate responses are carried out efficiently and accurately. This dynamic execution ensures that the platform remains responsive and adaptable to real-time data changes.



4. Expressions

Expressions are crucial for defining criteria that determine eligibility for data listing, record viewing, action execution, and data display in reports and dashboards.

1. **Configuration:** In the configuration phase, a common component is built to handle expressions across the entire platform. This component allows users to define criteria flexibly, specifying conditions for when data should be listed, records should be viewed, actions should be executed, and data should be shown in reports and dashboards.

This unified approach ensures consistency and ease of management across different areas of the platform.

2. **Runtime:** At runtime, the platform uses the defined expressions to evaluate data and actions dynamically. When data operations occur, the system checks the criteria specified in the expressions component to determine the appropriate actions. This ensures that the platform consistently applies the defined rules, providing accurate and relevant data and actions in real-time scenarios.

4.1. Cost Savings

The internally managed platform simplifies application development and user reviews, promoting agility through continuous feedback that results in swift outcomes. By having a single platform to learn, manage, and maintain, organizations can significantly reduce maintenance costs. This streamlined approach ensures that development processes are efficient, user interactions are seamless, and overall operational expenses are minimized, enhancing both productivity and cost-effectiveness.

5. Uses

Manufacturing: In the manufacturing sector, the platform can be utilized to create a variety of applications essential for operations. These include employee portals for managing worker information and communication, resource procurement apps for ordering and tracking materials, and consumption tracking systems to monitor resource usage. Additionally, customer support apps can be developed to improve service and engagement, ensuring a comprehensive and efficient manufacturing process.

Banking: In the banking sector, the platform can be utilized to develop a wide range of applications to enhance financial services. These include customer portals for managing accounts, transactions, and communication, loan processing apps for streamlining approvals and disbursements, and risk management tools to monitor and mitigate financial risks. Additionally, compliance management systems can be developed to ensure adherence to regulatory requirements, and customer relationship management (CRM) apps can improve service delivery and customer engagement. Overall, the platform supports efficient, secure, and customer-centric banking operations.

6. Conclusion

In conclusion, developing a comprehensive Salesforce-like platform involves integrating crucial components such as authentication, entities, apps, reports and dashboards, and events, all underpinned by a robust configuration and runtime system. Leveraging modern UI frameworks, microservices, cloud infrastructure, and open-source reporting tools ensures a scalable and flexible solution. The platform's ability to define expressions and criteria enhances data management and action execution. By fostering continuous feedback and maintaining a unified system, organizations can achieve greater agility and reduced maintenance costs. Ultimately, this approach enables streamlined operations across various sectors, including manufacturing, and supports efficient, cost-effective business processes.

7. References

1. Kumar R, Sharma Y, Agarwal S, Pragya, Parashar B. Extremely effective CRM Solution Using Salesforce. JETIR 2014;1.
2. Juhas G, Molnar L, Juhasova A, Ondrisova M, Mladoniczky M, Kovacik T. Low-code platforms and languages: the future of software development. 2022 20th International conference on emerging elearning technologies and applications (ICETA) 2022; 286-293.
3. Sanchis R, García-Perales O, Fraile F, Poler R. Low-Code as Enabler of digital transformation in manufacturing industry. Appl Sci 2020;10: 12.
4. Boniface M, Nasser B, Papay J, et al. Platform-as-a-Service architecture for real-time quality of service management in clouds. 2010 Fifth International Conference on Internet and Web Applications and Services 2010; 155-160.
5. Ram N, Tirupati S, Srinivas PVS. Deploying an Application on the Cloud. Int J Advanced Computer Science and Applications 2011;2.
6. Shahzadi S, Iqbal M, Qayyum Z, Dagiuklas T. Infrastructure as a Service (IaaS): A comparative performance analysis of open-source cloud platforms. 2017 IEEE 22nd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD) 2017.