International Journal of Science and Research Archive

eISSN: 2582-8185

Cross Ref DOI: 10.30574/ijsra

Journal homepage: https://ijsra.net/

(REVIEW ARTICLE)

Check for updates

Comprehensive Review of Multi-cloud Architecture for Salesforce in Enterprise Environments

Ravichandra Mulpuri *

Lead Salesforce Consultant, W.L. GORE, United States America.

International Journal of Science and Research Archive, 2025, 15(03), 1825-1838

Publication history: Received on 20 May 2025; revised on 25 June 2025; accepted on 28 June 2025

Article DOI: https://doi.org/10.30574/ijsra.2025.15.3.1967

Abstract

The adoption of multicloud architectures is rapidly increasing as enterprises seek flexibility, scalability, and optimization across diverse workloads. This paper examines the integration of Salesforce, a leading cloud-based CRM platform, within multicloud environments, particularly in large enterprise settings. It explores how Salesforce can operate seamlessly alongside platforms such as AWS, Microsoft Azure, and Google Cloud to support real-time data synchronization, resource allocation, and system interoperability. Key integration techniques discussed include API-based communication (REST, OData, GraphQL), data virtualization, and ETL processes to ensure data consistency across cloud platforms. The paper also addresses performance optimization strategies, including the distribution of workloads across specialized cloud services and the use of DevOps practices like CI/CD pipelines and monitoring tools. Security and compliance considerations are explored in the context of regulations such as GDPR, HIPAA, and CCPA, emphasizing the importance of unified governance, encryption, and access controls. Containerization technologies like Docker and Kubernetes are highlighted for their role in managing consistent deployments across multicloud ecosystems. Overall, this review provides a comprehensive analysis of the opportunities and challenges in integrating Salesforce within a multicloud strategy, offering insights into achieving operational efficiency, regulatory compliance, and scalable CRM performance.

Keywords: Multicloud Architecture; Salesforce Integration; Enterprise Cloud Environments; Data Synchronization; Salesforce CRM; Devops And CI/CD; Big Data Analytics; Containerization (Docker; Kubernetes)

1. Introduction

The evolution of Salesforce from a CRM platform to a comprehensive, enterprise-level solution has fundamentally transformed how businesses interact with technology. Initially created as a tool to manage customer relationships, Salesforce now serves as a backbone for diverse business processes, spanning sales, service, marketing, and commerce. This expansion into broader business functions underscores the need for robust, scalable, and flexible architectures to support its increasing role in enterprise operations (Sharma, 2012).

At the heart of this transformation lies the growing reliance on multicloud strategies. The rise of multicloud environments—where businesses use services from multiple cloud providers to fulfill different needs—has become a defining feature of modern enterprise architecture. Salesforce, as a key player in this ecosystem, must operate seamlessly within multicloud architectures to provide the performance, security, and interoperability enterprises demand (Manchar & Chouhan, 2017). This review delves into how Salesforce can be optimized for multicloud environments, focusing on the architectural patterns that support scalability, security, and integration across diverse cloud platforms like AWS, Azure, and Google Cloud.

* Corresponding author: Ravichandra Mulpuri

Targeted at enterprise architects, Salesforce developers, and IT leaders, this review provides in-depth insights into how Salesforce can be effectively integrated into multicloud setups. It explores the technical considerations, best practices, and strategies necessary to make Salesforce a powerful and flexible part of any enterprise's cloud ecosystem (Manchar & Chouhan, 2017).

Through detailed analysis and practical examples, the review aims to equip readers with the knowledge to leverage Salesforce in complex, distributed cloud environments, enhancing business agility and driving innovation (Lenzerini, 2002).

2. Defining Multicloud in the Salesforce Context

Multicloud strategies involve the use of multiple cloud platforms from different providers within the same enterprise architecture (Seth et al., 2024). This can mean utilizing AWS for compute-heavy tasks, Azure for enterprise applications, Google Cloud for machine learning, and Salesforce for CRM, all integrated into a cohesive environment. The multicloud approach allows organizations to optimize their IT operations by choosing the best platform for each workload, thereby achieving a combination of cost-efficiency, performance, and resilience (Islam et al., 2023).

The adoption of multicloud strategies provides enterprises with several benefits. For instance, by not being dependent on a single cloud provider, companies can reduce risks associated with vendor lock-in (Seth et al., 2024). This approach also enhances business autonomy by allowing different departments or business units to independently select the cloud provider that best meets their specific needs. Furthermore, performance optimization is achieved by selecting the most suitable cloud service for different tasks, ensuring that each service performs at its peak (Seth et al., 2024).

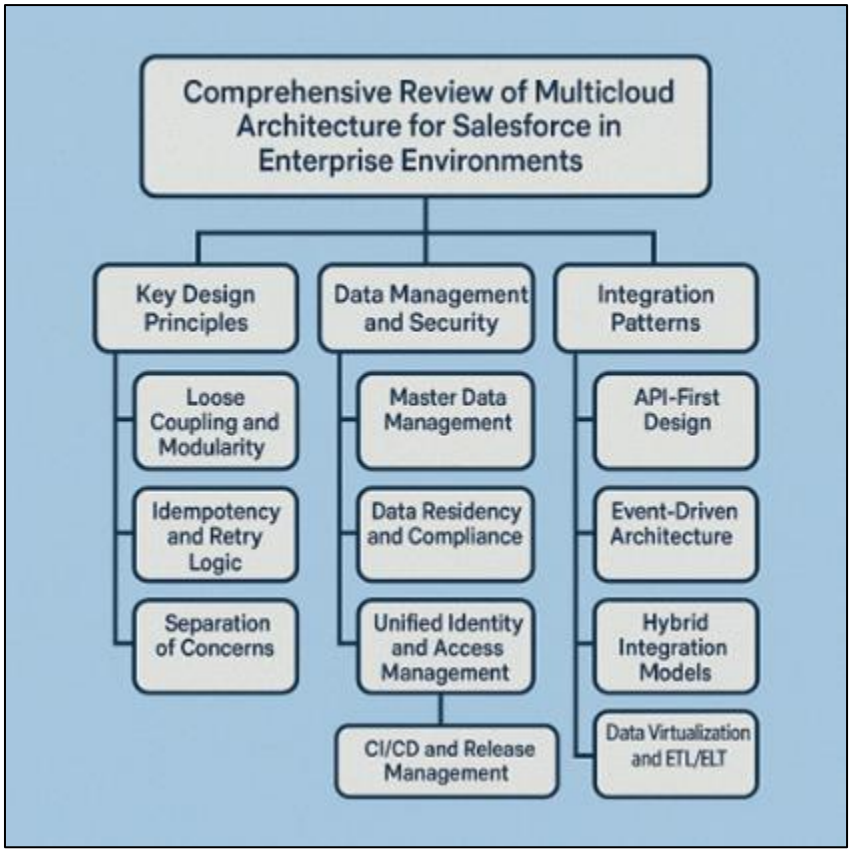

Figure 1 Comprehensive Review of Multicloud Architecture for Salesforce

Legend:A high-level flowchart summarizing the architectural pillars of multicloud deployment for Salesforce in enterprise environments. It organizes principles into three major domains: Key Design Principles (e.g., loose coupling, idempotency), Data Management and Security (e.g., MDM, compliance), and Integration Patterns (e.g., API-first, event-driven). It also includes a unified CI/CD and release management strategy, highlighting the layered, modular nature of scalable Salesforce deployments.

2.1. Salesforce as Part of the Multicloud Stack

Salesforce's role within the multicloud stack is significant, serving as a critical piece of the puzzle for enterprise organizations. As a core platform, Salesforce supports several essential business functions—sales, service, marketing, and commerce. These core clouds, integral to Salesforce's offering, are often complemented by external cloud platforms such as AWS, Azure, and Google Cloud (Buxmann et al., 2008).

Salesforce's flexibility and cloud-native architecture make it an ideal candidate for integration within multicloud environments. By leveraging tools such as Salesforce Connect and the ability to integrate through APIs, businesses can easily synchronize Salesforce data with other cloud platforms, ensuring a seamless user experience and consistent data flow across the enterprise (Choosing AWS and Salesforce Data Integrations, 2025; Heroku Pattern: Cross-Org Data Synchronization, 2020; Manchar & Chouhan, 2017).

2.2. Multicloud Drivers

Several key drivers make multicloud adoption an attractive strategy for large organizations. Regulatory compliance is one of the foremost considerations. Different regions have varying data residency laws and privacy regulations, such as GDPR in Europe or CCPA in California, that require businesses to manage their data accordingly. Multicloud strategies enable organizations to distribute their data across regions, ensuring compliance with local regulations (Samira et al., 2024).

Vendor diversity also plays a significant role in multicloud adoption. Relying on multiple providers allows enterprises to mitigate the risk of outages or performance issues that might arise if all services are hosted with a single vendor. Furthermore, multicloud environments promote business unit autonomy, allowing different departments or teams to choose the services that best suit their needs. Lastly, multicloud setups allow organizations to optimize the performance of their services by selecting the best-in-class providers for specific workloads, ensuring that the enterprise architecture is as efficient as possible (Yao & Lu, 2014).

3. Architectural Principles for Scalability

The principles of scalability are foundational to designing a successful multicloud architecture, particularly in enterprise environments. To ensure that Salesforce can scale effectively in a multicloud setup, certain principles must be adhered to (Balalaie et al., 2015). These principles are critical for maintaining performance, flexibility, and reliability in the face of growing demands .

3.1. Loose Coupling and Modularity

The concept of loose coupling is central to designing scalable and adaptable systems. By decoupling different components of the system, enterprises ensure that they are not locked into a specific technology stack or vendor (Carreras et al., 2022). This allows each component to evolve independently, thereby minimizing risks related to updates or failures in one part of the system.

In Salesforce, this principle is realized through the use of APIs, event-driven architectures, and microservices. By leveraging Salesforce APIs (e.g., REST, SOAP, and Bulk API), external systems can interact with Salesforce without being tightly integrated, allowing for greater flexibility (Designing for High-Volume Writes in Salesforce, 2021). For example, an enterprise might use Salesforce Connect to link Salesforce data with external databases, enabling the integration of data without physically storing it in Salesforce, making the system more flexible and less prone to performance degradation.

Another powerful example is the use of microservices. A company might have a sales team using Salesforce for CRM purposes while another business unit uses Salesforce for marketing automation (Manchar & Chouhan, 2017). With microservices, these units can operate independently, scaling their services without interfering with each other. This modular approach allows both the marketing and sales components to evolve separately without being dependent on a single system (Kaloudis, 2024).

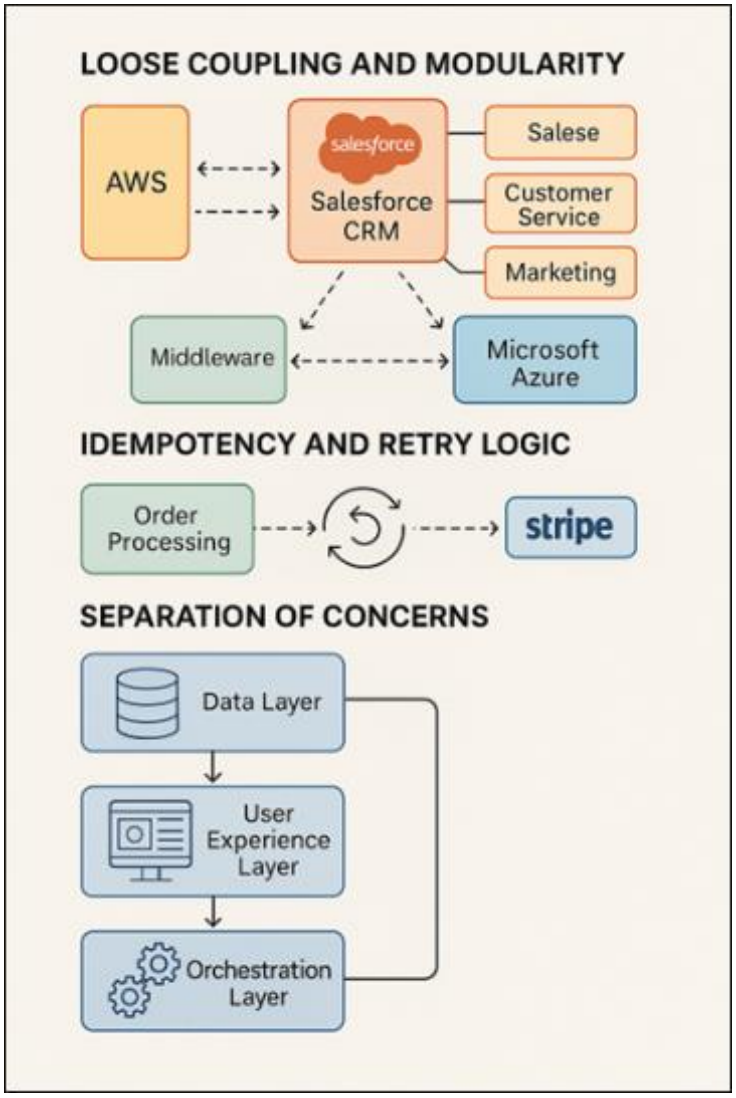

Figure 2 Data Virtualization and ETL/ELT in Multicloud Architecture

3.2. Legend

An architectural diagram demonstrating how customer data is extracted from Salesforce via ETL/ELT processes and routed into AWS for storage and analysis. Using tools like OData and GraphQL, this system enables real-time virtual access to unified data views across multicloud platforms (e.g., AWS Data Lake, analytics platforms), reducing data redundancy while ensuring regulatory compliance and real-time reporting.

In a global retail enterprise, a company might use Salesforce for customer service while integrating it with AWS for inventory management via microservices. The service dealing with inventory management can scale independently from the customer service system, allowing both systems to be optimized according to their individual demands (Kaloudis, 2024).

3.3. Idempotency and Retry Logic

Idempotency and retry logic are critical for ensuring data integrity, especially in asynchronous systems where messages or transactions are processed over time. In a multicloud environment, where data might be processed across multiple platforms, ensuring that duplicate transactions do not occur or cause inconsistent outcomes is crucial (Pourmajidi et al., 2018).

In Salesforce, idempotency is implemented through various mechanisms, ensuring that if the same event is processed multiple times due to network issues or system retries, the result remains consistent. For instance, when an order is processed through an external API integrated with Salesforce, if the connection fails and the request is retried, the same

order should not be duplicated. Using Platform Events or Change Data Capture (CDC) in Salesforce ensures that data flows in a controlled manner without duplication, preserving the accuracy of critical data (Finkelstein et al., 2009).

Consider a financial services company that integrates Salesforce with an external payments platform. When a payment is processed, the transaction data is sent to Salesforce via an API. If there is a failure in the network and the request is retried, idempotency ensures that the payment is recorded only once in Salesforce, avoiding duplicate financial records and ensuring that accounting systems remain accurate (Abu-Raqabeh, 2018).

3.4. Separation of Concerns

Separation of concerns is another core principle in designing scalable systems. It refers to organizing a system into distinct layers, each responsible for a specific aspect of the application. By separating concerns, developers can ensure that different parts of the system evolve independently without impacting the entire architecture (Nordli et al., 2023).

In a Salesforce multicloud environment, separation of concerns can be implemented by dividing the system into the data layer, user experience layer, and orchestration layer. The data layer handles interactions with external systems, ensuring that data flows seamlessly between platforms like AWS, Azure, or Google Cloud and Salesforce (Hammad et al., 2023). The user experience layer focuses on the front-end aspects, delivering a clean and intuitive interface for users. Lastly, the orchestration layer ensures that processes across these different components are well-coordinated (Hammad et al., 2023).

For instance, in a retail environment, Salesforce might serve as the central hub for customer data and order processing (data layer), while a custom application built on Azure might handle real-time inventory tracking (user experience layer). The orchestration layer ensures that inventory updates are reflected in Salesforce immediately, creating a seamless customer experience (McRoberts, 2014).

A healthcare provider could separate their architecture into layers to handle different concerns. The data layer might pull patient records from an on-prem database while using Salesforce Health Cloud for patient relationship management. The user experience layer would include a web application built using Salesforce Lightning Components for patient portals (Anshari & Almunawar, 2012). The orchestration layer ensures that when a patient updates their medical history, this change is propagated across both the Salesforce system and the external database, keeping all systems synchronized in real-time.

4. Key Salesforce Architectural Patterns

4.1. API-First Design

An API-first design emphasizes the creation of APIs as the foundation for building any integration, ensuring that external systems can communicate seamlessly with Salesforce. This approach simplifies integration efforts by focusing on well-documented and standardized interfaces, such as REST, SOAP, and Apex APIs (Manchar & Chouhan, 2017). By implementing API-first, businesses are ensuring that their systems are modular and can easily evolve over time without breaking integrations.

In Salesforce, API-first design enables the integration of Salesforce with various external platforms—whether internal systems or third-party services. It allows developers to create APIs that can easily expose Salesforce data or business logic to external applications while maintaining security and access control through features like Named Credentials and External Services (De, 2017).

A global retail organization might integrate Salesforce with an eCommerce platform, ERP system, and a payment gateway using an API-first approach. By defining clear REST APIs for managing orders, customer records, and inventory, the organization can ensure that every update in the Salesforce CRM, such as customer contact details or order status, automatically triggers changes in the external eCommerce platform and ERP system (Yang & Han, 2020). For added security, Named Credentials in Salesforce would be used to securely store API keys and authentication tokens, ensuring that only authorized applications can access Salesforce data.

4.2. Event-Driven Architecture

An event-driven architecture allows systems to react to changes or events as they occur, ensuring real-time communication between systems. In Salesforce, this is achieved using Platform Events and Change Data Capture (CDC)

(Manchar & Chouhan, 2017). These features capture changes made in Salesforce (such as updates to customer records, opportunities, or product data) and instantly communicate those changes to other systems or applications.

This architecture is especially beneficial for businesses that need to keep multiple systems in sync and require near-instantaneous updates to maintain operational efficiency (Pathak, 2024).

Consider a financial services firm using Salesforce as its CRM system while also maintaining an external transaction processing system. As soon as a user updates the customer's profile in Salesforce, the change can trigger a Platform Event, which notifies the external system to initiate a series of actions like credit checks or updating account balances. Similarly, CDC can be employed to monitor changes in critical data records (like opportunities or cases), ensuring that the external systems, such as ERP or order management systems, are instantly notified and stay up-to-date with the latest data (Sutton et al., 2020).

For example, if a new lead is created in Salesforce, CDC can capture this change and immediately update the marketing automation system to initiate a follow-up email. This ensures real-time customer engagement and marketing operations (Adlin et al., 2019).

4.3. Hybrid Integration Models

A hybrid integration model leverages both direct integrations (using Salesforce APIs) and intermediary middleware systems (like MuleSoft, Dell Boomi, or Informatica). This approach is particularly useful when integrating Salesforce with a variety of cloud services, on-premise systems, and third-party applications. Hybrid integration combines the flexibility of direct API calls with the centralized control of an orchestration layer (Treiber & Bernhardt, 2021).

Using Salesforce Connect, businesses can access and display data stored outside of Salesforce (in external systems such as SAP or Oracle databases) as if it were native Salesforce data, without the need to replicate it. This allows for real-time integration without the overhead of data duplication (Leow, 2010).

Figure 3 API-First Integration with Salesforce

4.4. Legend

This diagram illustrates the API-first integration pattern where Salesforce CRM connects to external systems such as an eCommerce platform, ERP system, and payment gateway. It emphasizes modular integration via well-defined APIs, ensuring that updates in Salesforce can trigger workflows and synchronize data across the enterprise ecosystem without breaking external dependencies.

In the case of a large manufacturing company, Salesforce Connect could be used to display real-time inventory data from an on-premise ERP system directly within Salesforce (Kenge & Khan, 2020). At the same time, MuleSoft could orchestrate more complex workflows, such as syncing data between Salesforce and a marketing automation platform

or managing customer service interactions with external helpdesk systems. This hybrid integration model ensures that all systems are well-connected and that the data flows seamlessly, with minimal duplication and real-time updates.

5. Multicloud Integration Strategies

5.1. Orchestration Platforms

Orchestration platforms help manage and automate complex workflows across multiple systems, ensuring that tasks are performed in the right order and that all necessary systems are involved in the process. Platforms like MuleSoft, Apache Camel, and AWS Step Functions are essential for managing workflows in a multicloud setup, where different services (e.g., Salesforce, AWS, Azure) are involved in the same business process (Tiwari et al., 2018).

These orchestration tools facilitate the integration of Salesforce with various cloud services, enabling the seamless movement of data and tasks between disparate systems. They provide centralized control over workflows, enabling businesses to monitor and manage the entire process, ensuring efficiency and reducing errors (Manchar & Chouhan, 2017).

Consider a global logistics company that uses Salesforce for managing customer relationships but also relies on AWS Lambda for real-time processing and Azure Logic Apps for workflows. With MuleSoft as the orchestration platform, the company can build an automated workflow that starts when a customer's order is placed in Salesforce. MuleSoft ensures that the order details are sent to the AWS Lambda function for processing, while the shipping details are automatically transferred to Azure Logic Apps for shipment tracking, providing a unified solution for managing the full customer journey across multiple clouds (Erturk & He, 2018).

5.2. Data Sync and Virtualization

In multicloud environments, data synchronization across different platforms is critical to ensuring that all systems reflect the same, up-to-date information. Data virtualization allows organizations to access and query data stored across different cloud platforms without needing to replicate or move that data physically. This reduces latency, improves efficiency, and lowers storage costs.

The use of ETL/ELT processes (Extract, Transform, Load/Extract, Load, Transform) and modern API-based methods like OData and GraphQL enable businesses to synchronize and query data from multiple sources in real-time (Muddasir et al., 2021).

A global eCommerce company might use AWS for its data lake, Google BigQuery for analytics, and Salesforce for CRM. Using ETL processes, the company could periodically extract customer data from Salesforce, transform it into a suitable format, and load it into AWS Redshift for further processing. OData or GraphQL APIs can then be used to virtually access the combined data from multiple platforms, ensuring that reports or dashboards reflect real-time data from all sources without needing to replicate or store redundant copies (Knox et al., 2019).

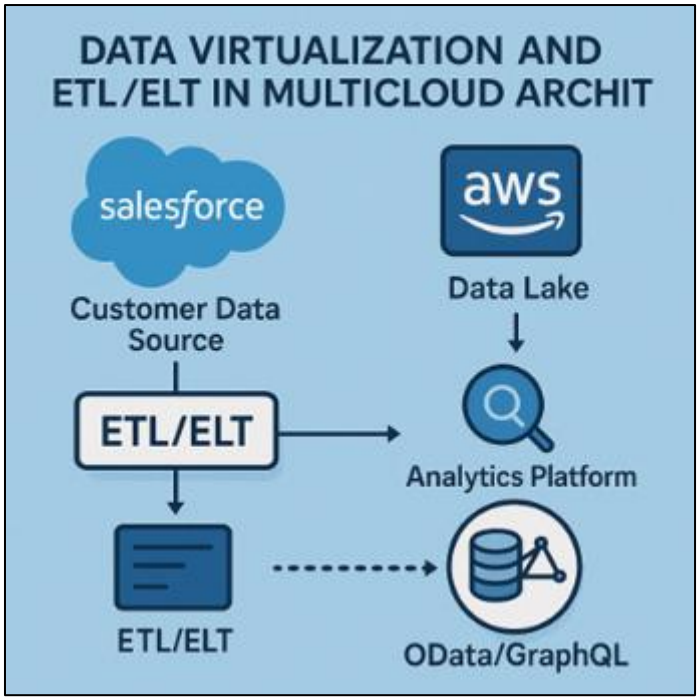

Figure 4 Loose Coupling, Idempotency, and Separation of Concerns in Salesforce Multicloud

5.3. Legend

A detailed conceptual map visualizing three core design tenets:

- Loose Coupling and Modularity: Demonstrates how middleware like MuleSoft bridges Salesforce CRM, AWS, and Azure, allowing independent scalability of business units (sales, service, marketing).
- Idempotency and Retry Logic: Shows a payment workflow where duplicate events are handled safely, preventing multiple charges via Stripe.
- Separation of Concerns: Breaks the architecture into data, user experience, and orchestration layers, allowing individual development and scalability of each system layer.

5.4. Real-Time vs. Batch Processing

When integrating systems across multicloud platforms, businesses often need to decide between real-time processing or batch processing. Real-time processing provides instant updates, ensuring that systems react to changes immediately. However, it can be more complex and costly due to the need for constant connectivity and low-latency data transfer (Bobryakov et al., 2021).

Batch processing, on the other hand, involves collecting and processing data in scheduled intervals, reducing system load and network traffic. While it's more efficient in terms of cost, it introduces some latency, as updates are not immediately reflected in the systems (Brook, 2015).

In a banking environment, real-time processing might be necessary to update account balances and transactions as they happen, ensuring that users see their up-to-date financial information immediately. This can be achieved through Platform Events in Salesforce, integrating with external banking systems (Badmus et al., 2024).

However, in a retail scenario, batch processing might be more appropriate for inventory updates. The retailer could synchronize product stock data across Salesforce and its ERP system in nightly batches, as real-time updates may not be critical for stock levels and would incur unnecessary costs. Batch jobs could be scheduled to run after hours, ensuring that the data is updated overnight (Salesforce Stuff, 2018).

6. Data Architecture and Governance

Master Data Management (MDM) is crucial for ensuring data integrity and consistency across Salesforce and external platforms. As businesses scale, managing core data (such as customer records, inventory, and financial data) across various systems becomes increasingly complex (Adesina et al., 2024). MDM strategies ensure that businesses can maintain a single, authoritative source of truth—the "golden record"—across all systems, thus reducing inconsistencies and ensuring that all platforms operate with the same, accurate data. In a multicloud architecture, MDM becomes even more critical, as data is distributed across multiple cloud services, making it essential to establish strong governance practices to preserve the integrity and accuracy of data as it flows between systems (Sun et al., 2014).

Data residency and compliance are key concerns for enterprises operating on a global scale. With the rise of strict data protection regulations, such as GDPR and CCPA, businesses must ensure that they are managing their data in compliance with regional laws. In a multicloud environment, this challenge is amplified as data may reside in various cloud platforms across different regions (Farhad, 2024). To meet these compliance requirements, businesses must implement strategies that ensure data is stored, processed, and accessed according to regional regulations. Additionally, encryption and data sovereignty must be prioritized to protect sensitive information and ensure compliance with industry standards (AllahRakha, 2023).

Unified Identity and Access Management is essential for securing multicloud environments. With the integration of Salesforce and multiple external platforms, managing user access across these systems becomes increasingly complex. Single Sign-On (SSO), OAuth, and centralized identity providers like Azure Active Directory are fundamental for creating a seamless and secure authentication process across all cloud services (Babel & Sedlmeir, 2023). By centralizing identity management, businesses ensure that users can securely access all integrated platforms while reducing the administrative overhead of managing separate logins and credentials for each system (Lasance, 2011).

Figure 5 Multi-cloud data governance overview

7. Deployment and DevOps in a Multicloud Setup

The adoption of continuous integration and continuous delivery (CI/CD) practices in Salesforce environments ensures that updates to both declarative and programmatic components are integrated and deployed consistently and efficiently. CI/CD pipelines automate the process of testing and deploying code, ensuring that new features, bug fixes, and security patches are seamlessly introduced across development, staging, and production environments. By leveraging tools like Salesforce DX, GitHub Actions, and Azure DevOps, businesses can streamline their development workflows, allowing for faster release cycles, reduced manual errors, and greater control over deployment processes (Wiedemann et al., 2019).

Effective release management across platforms ensures that businesses can maintain consistency and compatibility between Salesforce and external systems. Version control is a fundamental practice in multicloud setups, ensuring that all declarative configurations and code are properly tracked and managed. Release management practices help teams coordinate updates, avoid deployment conflicts, and ensure that any changes made in one platform do not negatively impact others. A well-defined release management strategy enhances the overall stability and scalability of Salesforce implementations in multicloud environments (Raikar et al., 2021).

Testing automation is a critical component in maintaining the stability and quality of Salesforce systems as they evolve. Automated performance, regression, and integration tests ensure that code changes do not introduce new issues or regressions across environments. These tests help identify potential issues early in the development cycle, reducing the risk of costly production errors. By automating testing, businesses can ensure that their Salesforce implementations remain robust and reliable, even as they integrate with multiple external cloud platforms (Schwartz, 2025).

8. Security and Compliance Design

Zero Trust Architecture is a security model that assumes that no entity, inside or outside the organization, should be trusted by default. In multicloud environments, where data and applications are spread across different platforms, applying a Zero Trust approach is vital for ensuring data security and preventing unauthorized access. Continuous verification and authentication of all access requests—whether they originate internally or externally—ensure that only authorized users can access sensitive data. Tokenization, FIPS (Federal Information Processing Standards), and Bring Your Own Key (BYOK) further strengthen data protection measures by ensuring that sensitive data remains encrypted and secure across cloud platforms (Kaushik & Gandhi, 2019).

Auditing and monitoring are critical components of a security and compliance strategy. Continuous monitoring allows businesses to track all activities across Salesforce and external systems, ensuring that they can detect potential security incidents or compliance violations in real-time. Tools like Splunk and AWS CloudWatch provide centralized logging and alerting capabilities, making it easier for security teams to monitor activities across all cloud platforms. Regular audits ensure that the systems remain compliant with relevant regulations, such as GDPR or HIPAA, and that any deviations from established security protocols are promptly addressed (Iqbal et al., 2022).

9. Case Studies and Architectural Blueprints

Case studies provide valuable insights into how Salesforce multicloud architectures function in real-world environments. These case studies highlight how businesses have successfully integrated Salesforce with various cloud platforms to optimize their operations. From retail enterprises using Salesforce alongside AWS to healthcare providers ensuring HIPAA-compliant multicloud integration, these examples showcase the power of Salesforce in a multicloud context. Each case study emphasizes the challenges faced, the architectural patterns used, and the results achieved, providing businesses with valuable lessons for their own Salesforce integrations (Lenzerini, 2002).

10. Challenges and Mitigation Strategies

Data silos and latency issues are common challenges in multicloud architectures. As data is stored across different platforms, ensuring that it remains synchronized in real-time is crucial for maintaining operational efficiency. Strategies like caching and edge computing are employed to reduce latency and provide near-instantaneous access to data. By placing data closer to where it is needed—either through local caching or edge computing solutions—businesses can minimize the delays associated with retrieving data from central cloud platforms, improving both performance and user experience (Kallimani et al., 2023).

Complexity in governance is another challenge faced by enterprises with multicloud environments. As businesses integrate multiple cloud platforms, they must establish clear governance frameworks to manage resources, monitor access, and enforce security policies. Solutions such as enterprise architecture boards and shared service models provide centralized governance that ensures consistency across systems while allowing each platform to operate autonomously. This governance model helps businesses maintain control over their multicloud setup while empowering individual teams to manage their respective systems (Qian et al., 2009).

Skills and cultural barriers are also significant obstacles in successfully implementing multicloud architectures. Cross-skilling between Salesforce teams and cloud infrastructure teams ensures smoother transitions and better collaboration. By fostering a culture of continuous learning and cross-platform collaboration, businesses can break

down silos and ensure that teams are equipped with the knowledge needed to manage multicloud environments effectively (Lin et al., 2014).

11. Future Trends

AI-driven integration is expected to become increasingly important as businesses seek to leverage artificial intelligence for predictive analytics and decision-making. By integrating AI services such as Salesforce Einstein with external platforms like Google Vertex AI and OpenAI, organizations can gain deeper insights into customer behavior, optimize operations, and automate decision-making processes.

Composable enterprise architecture is another emerging trend, enabling businesses to build more agile and adaptable systems. With the rise of Packaged Business Capabilities (PBCs), businesses can modularize their services, making it easier to compose and decompose business processes based on evolving needs. This trend is supported by platforms like Salesforce, which enable organizations to quickly adapt to changing business requirements.

Event mesh and global messaging systems like Kafka, Solace, and EventBridge are becoming essential tools for managing scalable, event-driven architectures. These tools facilitate reliable, low-latency message delivery across distributed systems, ensuring that data can be processed in real-time, even as businesses scale globally. These technologies will play a pivotal role in the future of multicloud integration, providing businesses with the scalability and reliability they need to thrive in a rapidly changing digital landscape.

12. Conclusion

In conclusion, embracing multicloud architectures with Salesforce requires a strategic approach to ensure agility, security, and scalability. By following best practices in data architecture, deployment, security, and compliance, businesses can unlock the full potential of Salesforce in multicloud environments. The integration of AI, composable enterprise architecture, and advanced messaging systems will drive future innovation, allowing businesses to stay competitive in the ever-evolving digital landscape. Embracing these principles will empower businesses to adapt quickly to changes, protect sensitive data, and deliver seamless experiences across a diverse range of cloud platforms.

Compliance with ethical standards

Disclosure of conflict of interest

Authors declare no conflict of interest whatsoever

References

[1] Abu-Raqabeh, T. (2018). Accounting Information Systems. Education and Linguistics Research, 4(2), 104. https://doi.org/10.5296/elr.v4i2.14045

[2] Adesina, A. A., Iyelolu, T. V., & Paul, P. O. (2024). Leveraging predictive analytics for strategic decision-making: Enhancing business performance through data-driven insights. World Journal of Advanced Research and Reviews, 22(3), 1927. https://doi.org/10.30574/wjarr.2024.22.3.1961

[3] Adlin, F. N., Ferdiana, R., & Fauziati, S. (2019). Current Trend and Literature on Electronic CRM Adoption Review. Journal of Physics Conference Series, 1201(1), 12058. https://doi.org/10.1088/1742-6596/1201/1/012058

[4] AllahRakha, N. (2023). Legal Challenges for International Fintech Startups. International Journal of Law and Policy, 1(8). https://doi.org/10.59022/ijlp.148

[5] Anshari, M., & Almunawar, M. N. (2012). Evaluating CRM Implementation in Healthcare Organization. arXiv (Cornell University). https://doi.org/10.48550/arXiv.1204.3689

[6] Babel, M., & Sedlmeir, J. (2023). Bringing data minimization to digital wallets at scale with general-purpose zero-knowledge proofs. arXiv (Cornell University). https://doi.org/10.48550/arXiv.2301.00823

[7] Badmus, O., Rajput, S., Arogundade, J. B., & Williams, M. H. (2024). AI-driven business analytics and decision making. World Journal of Advanced Research and Reviews, 24(1), 616. https://doi.org/10.30574/wjarr.2024.24.1.3093

[8] Balalaie, A., Heydarnoori, A., & Jamshidi, P. (2015). Migrating to Cloud-Native Architectures Using Microservices: An Experience Report. arXiv (Cornell University). https://doi.org/10.48550/arXiv.1507.08217

[9] Bobryakov, A., Прокопенко, С. А., Мисник, А. Е., & Krutalevich, S. K. (2021). Modeling of Industrial and Technological Processes in Complex Systems Based on Neuro-Fuzzy Petri Nets. Journal of Physics Conference Series, 2096(1), 12173. https://doi.org/10.1088/1742-6596/2096/1/012173

[10] Brook, A. (2015). Evolution and Practice: Low-latency Distributed Applications in Finance. Queue, 13(4), 40. https://doi.org/10.1145/2756506.2770868

[11] Buxmann, P., Heß, T., & Lehmann, S. (2008). Software as a Service. WIRTSCHAFTSINFORMATIK, 50(6), 500. https://doi.org/10.1007/s11576-008-0095-0

[12] Carreras, A., Navarro, J., Sans, C., & Zaballos, A. (2022). Communication Technologies in Emergency Situations. Electronics, 11(7), 1155. https://doi.org/10.3390/electronics11071155

[13] Choosing AWS and Salesforce Data Integrations. (2025). https://community.aws/content/2jlOxWSMllW2PRMKqm2lsvk7r4Y/aws-and-salesforce-data-integrations?lang=en

[14] De, B. (2017). API Management. In Apress eBooks (p. 15). https://doi.org/10.1007/978-1-4842-1305-6_2

[15] Designing for High-Volume Writes in Salesforce. (2021). https://medium.com/salesforce-architects/designing-for-high-volume-writes-in-salesforce-285689a08100.

[16] Erturk, E., & He, S. (2018). Study on A High-integrated Cloud-Based Customer Relationship Management System. arXiv (Cornell University). https://doi.org/10.48550/arXiv.1812.09005

[17] Farhad, M. A. (2024). Consumer data protection laws and their impact on business models in the tech industry. Telecommunications Policy, 48(9), 102836. https://doi.org/10.1016/j.telpol.2024.102836

[18] Finkelstein, S., Jacobs, D., & Brendle, R. (2009). Principles for Inconsistency. arXiv (Cornell University). https://doi.org/10.48550/arXiv.0909.1782

[19] Hammad, H. M., Sahmoud, T., & Ghazala, A. A. R. A. (2023). Convert Monolithic Application to Microservice Application. arXiv (Cornell University). https://doi.org/10.48550/arXiv.2306.08851

[20] Heroku Pattern: Cross-Org Data Synchronization. (2020). https://medium.com/salesforce-architects/heroku-pattern-cross-org-data-synchronization-addcaa2c970a

[21] Iqbal, Y., Tahir, S., Tahir, H., Khan, F., Saeed, S., Almuhaideb, A. M., & Syed, A. M. (2022). A Novel Homomorphic Approach for Preserving Privacy of Patient Data in Telemedicine. Sensors, 22(12), 4432. https://doi.org/10.3390/s22124432

[22] Islam, R., Patamsetti, V., Gadhi, A., Gondu, R. M., Bandaru, C. M., Kesani, S. C., & Abiona, O. (2023). The Future of Cloud Computing: Benefits and Challenges. International Journal of Communications Network and System Sciences, 16(4), 53. https://doi.org/10.4236/ijcns.2023.164004

[23] Kallimani, R., Pai, K., Raghuwanshi, P., Iyer, S., & López, O. L. A. (2023). TinyML: Tools, Applications, Challenges, and Future Research Directions. arXiv (Cornell University). https://doi.org/10.48550/arXiv.2303.13569

[24] Kaloudis, M. (2024). Evolving Software Architectures from Monolithic Systems to Resilient Microservices: Best Practices, Challenges and Future Trends. International Journal of Advanced Computer Science and Applications, 15(9). https://doi.org/10.14569/ijacsa.2024.0150901

[25] Kaushik, S., & Gandhi, C. (2019). Capability Based Outsourced Data Access Control with Assured File Deletion and Efficient Revocation with Trust Factor in Cloud Computing. International Journal of Cloud Applications and Computing, 10(1), 64. https://doi.org/10.4018/ijcac.2020010105

[26] Kenge, R., & Khan, Z. U. (2020). A Research Study on the ERP System Implementation and Current Trends in ERP. Shanlax International Journal of Management, 8(2), 34. https://doi.org/10.34293/management.v8i2.3395

[27] Knox, S., Tomlinson, J., Harou, J., Meier, P., Rosenberg, D. E., Lund, J. R., & Rheinheimer, D. E. (2019). An open-source data manager for network models. Environmental Modelling & Software, 122, 104538. https://doi.org/10.1016/j.envsoft.2019.104538

[28] Lasance, M. (2011). Single Sign-on(SSO) to Cloud based Services and Legacy Applications “Hitting the IAM wall.” In Vieweg+Teubner eBooks (p. 53). https://doi.org/10.1007/978-3-8348-9788-6_5

[29] Lenzerini, M. (2002). Data integration. 233. https://doi.org/10.1145/543613.543644

[30] Leow, P. (2010). What is customer relationship management (CRM). http://repo.uum.edu.my/2056/

[31] Lin, C., Yu, W.-C., & Wang, J. (2014). Cloud Collaboration: Cloud-based Instruction for Business Writing Class. World Journal of Education, 4(6). https://doi.org/10.5430/wje.v4n6p9

[32] Manchar, A., & Chouhan, A. (2017). Salesforce CRM: A new way of managing customer relationship in cloud environment. 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT), 1. https://doi.org/10.1109/icecct.2017.8117887

[33] McRoberts, M. (2014). Software Licensing in the Cloud Age. arXiv (Cornell University). https://doi.org/10.48550/arXiv.1401.5346

[34] Muddasir, N. M., Raghuveer, K., & Dayanand, R. (2021). Towards Comparative Analysis of Resumption Techniques in ETL. Indonesian Journal of Information Systems, 82. https://doi.org/10.24002/ijis.v3i2.3776

[35] Nordli, E. T., Haugeland, S. G., Nguyen, P. H., Song, H., & Chauvel, F. (2023). Migrating monoliths to cloud-native microservices for customizable SaaS. Information and Software Technology, 160, 107230. https://doi.org/10.1016/j.infsof.2023.107230

[36] Pathak, P. G. (2024). Research Paper on Salesforce Technology. International Journal of Advanced Research in Science Communication and Technology, 98. https://doi.org/10.48175/ijarsct-15215

[37] Pourmajidi, W., Steinbacher, J., Erwin, T., & Miranskyy, A. (2018). On Challenges of Cloud Monitoring. arXiv (Cornell University). https://doi.org/10.48550/arXiv.1806.05914

[38] Qian, L., Luo, Z., Du, Y., & Guo, L. (2009). Cloud Computing: An Overview. In Lecture notes in computer science (p. 626). Springer Science+Business Media. https://doi.org/10.1007/978-3-642-10665-1_63

[39] Raikar, K., Parikh, A., Ekbote, A., & Singh, V. (2021). Cloud Based ERP Adoption in Educational Institutions. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.3867951

[40] Salesforce Stuff. (2018). https://www.sfdcstuff.com/2018/11

[41] Samira, Z., Weldegeorgise, Y. W., Osundare, O. S., Ekpobimi, H. O., & Kandekere, R. C. (2024). Comprehensive data security and compliance framework for SMEs. Magna Scientia Advanced Research and Reviews, 12(1), 43. https://doi.org/10.30574/msarr.2024.12.1.0146

[42] Schwartz, C. (2025). Salesforce Test Automation: Complete 2024 Guide. https://www.leapwork.com/blog/salesforce-test-automation

[43] Seth, D., Nerella, H., Najana, M., & Tabbassum, A. (2024). Navigating the Multi-Cloud Maze: Benefits, Challenges, and Future Trends. https://doi.org/10.21428/e90189c8.8c704fe4

[44] Sharma, A. (2012). Chapter 9 Emerging Transformations in the Business-to-Business Global Salesforce. In Advances in business marketing & purchasing/Advances in business marketing and purchasing (p. 219). Emerald Publishing Limited. https://doi.org/10.1108/s1069-0964(2012)0000018014

[45] Sun, Y., Zhang, J., Xiong, Y., & Zhu, G. (2014). Data Security and Privacy in Cloud Computing. International Journal of Distributed Sensor Networks, 10(7), 190903. https://doi.org/10.1155/2014/190903

[46] Sutton, R. T., Pincock, D., Baumgart, D. C., Sadowski, D., Fedorak, R. N., & Kroeker, K. I. (2020). An overview of clinical decision support systems: benefits, risks, and strategies for success [Review of An overview of clinical decision support systems: benefits, risks, and strategies for success]. Npj Digital Medicine, 3(1). Nature Portfolio. https://doi.org/10.1038/s41746-020-0221-y

[47] Tiwari, R., Upadhyay, S., Lal, G., & Tanwar, V. (2018). Project Workflow Management: A Cloud based Solution-Scrum Console. International Journal of Engineering & Technology, 7(4), 2457. https://doi.org/10.14419/ijet.v7i4.15799

[48] Treiber, M., & Bernhardt, H. (2021). NEVONEX—The Importance of Middleware and Interfaces for the Digital Transformation of Agriculture. 3. https://doi.org/10.3390/engproc2021009003

[49] Wiedemann, A., Forsgren, N., Wiesche, M., Gewald, H., & Krcmar, H. (2019). The DevOps Phenomenon. Queue, 17(2), 93. https://doi.org/10.1145/3329781.3338532

[50] Yang, X., & Han, X. (2020). Analysis of the Impact of Customer Relationship Management on Data Outsourcing Enterprises. Proceedings of the 6th International Conference on Humanities and Social Science Research (ICHSSR2020). https://doi.org/10.2991/assehr.k.200428.086

[51] Yao, W., & Lu, L. (2014). A Selection Algorithm of Service Providers for Optimized Data Placement in Multi-Cloud Storage Environment. In Communications in computer and information science (p. 81). Springer Science+Business Media. https://doi.org/10.1007/978-3-662-46248-5_11