Available online www.jsaer.com

Journal of Scientific and Engineering Research, 2021, 8(1):230-240

Research Article

ISSN: 2394-2630
CODEN(USA): JSERBR

Integrating IT Service Management and Customer Relationship Management Systems

Chirag Amrutlal Pethad

PetSmart.com, LLC, Stores and Services,
Phoenix, Arizona, USA
ChiragPethad@live.com, ChiragPethad@gmail.com, Cpethad@petsmart.com

Abstract: Integrating ServiceNow with Salesforce enhances operational efficiency and data consistency through various methods, including Integration Hub, MuleSoft, and custom API solutions. Key steps involve enabling Integration Hub, setting up connections, creating integration actions, and testing the integration. Security measures, such as OAuth authentication and data encryption, are crucial. Comprehensive testing, monitoring, and maintenance ensure the integration's robustness. Best practices include clear objectives, documentation, modular design, and stakeholder collaboration. This integration facilitates seamless data flow and automation, improving both internal operations and customer interactions.

Keywords: Integration, Case, Authentication, ServiceNow, Salesforce, IT Service Management (ITSM), Customer Relationship Management (CRM), Integration Hub,MuleSoft, API, Workflow automation, Data synchronization, Security and compliance, Testing, Monitoring and maintenance.

1. Introduction

In the rapidly evolving landscape of modern business operations, organizations strive to enhance efficiency, streamline workflows, and deliver exceptional customer experiences**.** **ServiceNow** and **Salesforce** are two leading cloud-based platforms that play pivotal roles in achieving these objectives.

• **ServiceNow** is renowned for its robust capabilities in IT Service Management (ITSM), workflow automation, and managing internal business processes across various departments such as IT, HR, and security operations. It provides a centralized platform to handle incidents, service requests, asset management, and more, fostering seamless internal operations.

• **Salesforce,** on the other hand, stands as the premier Customer Relationship Management (CRM) platform, empowering businesses to manage their sales, marketing, customer service, and engagement strategies effectively. Its comprehensive suite of tools facilitates lead management, customer support, marketing automation, and insightful analytics, driving superior customer interactions and growth.

This white paper outlines a comprehensive step by step Implementation guide for leveraging the strengths of both ServiceNow and Salesforce through integration. Organizations create a cohesive ecosystem that enhances both internal efficiencies and external customer satisfaction. This synergy not only drives operational excellence but also supports scalable growth and a competitive edge in the marketplace.

2. Benefits of Integration

Integrating ServiceNow with Salesforce bridges the gap between internal operational workflows and customer-facing activities. This integration enables the seamless flow of data and coordinated processes between IT services and CRM functions, resulting in numerous benefits:

A. Enhanced Operation Efficiency

Automates data exchange and workflows, reducing manual data entry and minimizing errors.

B. Unified Customer Experience

Ensures that customer-facing teams have real-time access to internal service data, enabling more informed and responsive customer interactions.

C. Improved Data Consistency

Maintains synchronized data across both platforms, ensuring accuracy and reliability.

D. Streamlined Incident and Case Management

Allows for automatic creation and updating of cases/incidents between systems, enhancing support and service delivery.

E. Better Visibility and Reporting

Combines data from both platforms for comprehensive analytics and reporting, aiding in strategic decision-making.

3. Use Cases

A. Incident to Case Management

- A technical issue is reported in ServiceNow (ITSM). Automatically create a corresponding Case in Salesforce for the customer service team to address.

B. Customer Onboarding

- When a new customer is added in Salesforce, automatically provision necessary IT services and assets in ServiceNow.

C. Asset Management

- Sync IT asset information from ServiceNow with customer account details in Salesforce to provide a comprehensive view of customer assets.

D. Change Management Notifications

- When a change request is approved in ServiceNow, notify the sales or customer service teams in Salesforce to adjust customer interactions accordingly.

E. Knowledge Base Sharing

- Share knowledge articles from ServiceNow with Salesforce to empower customer service agents with up-to-date information.

F. Customer Support

- ServiceNow manages internal IT support, while Salesforce handles external customer support, ensuring comprehensive service coverage.

4. Integration Overview

There are three primary methods to integrate ServiceNow with Salesforce:

A. Native Integration

Utilize built-in connectors and tools provided by ServiceNow and Salesforce for straightforward integrations without extensive coding.

- **ServiceNow Integration Hub:** ServiceNow’s low-code integration platform with pre-built connectors (spokes) for Salesforce.
- **Salesforce Connect:** Salesforce’s tool to connect with external data sources, including ServiceNow.

B. Middleware Platforms

Use third-party integration platforms to handle complex workflows, data transformations, and orchestrate communication between ServiceNow and Salesforce.

- **MuleSoft Any point Platform:** A robust integration platform supporting advanced workflows.
- **Zapier:** Suitable for simpler, event-driven integrations.
- **Dell Boomi:** Another powerful integration platform for enterprise solutions.

C. Custom API Integrations

Develop bespoke integrations tailored to specific business needs using the APIs provided by both ServiceNow and Salesforce.

- **ServiceNow REST APIs:** For interacting with ServiceNow data and functionalities.
- **Salesforce REST/SOAP APIs:** For interacting with Salesforce data and services.

5. Prerequisites for Integration Implementation

Before diving into the integration process, we need to make sure we have the following things setup / configured.

- **ServiceNow Instance:** Access with administrative privileges.
- **Salesforce Instance:** Access with administrative privileges.
- **Integration Platform Account:** Depending on the method (e.g., MuleSoft, Zapier). We will explore multiple options in this document.
- **API Credentials:** Necessary API keys, tokens, or OAuth credentials for both ServiceNow and Salesforce.
- **Development Tools:** IDE or text editor (e.g., Visual Studio Code) for custom integrations.
- **Network Access:** Ensure that firewalls and network configurations allow communication between ServiceNow, Salesforce, and any middleware.
- **Documentation Access:** Familiarity with ServiceNow and Salesforce APIs and documentation.

6. Step by Step Implementation

Here's a step-by-step guide for integrating Service Now and Salesforce using different Integration patterns or platforms.

A. Service Now and Saleforce Security Configuration

Step 1: Configuration for Salesforce Connected App [6] [7]

- Log into Salesforce:
    - O Navigate to Setup.
- Create a Connected App:
    - O Go to App Manager.
    - O Click New Connected App.
- Configure Basic Information:
    - O Connected App Name: e.g., "ServiceNow Integration".
    - O API Name: Auto-filled based on the app name.
    - O Contact Email: Your email address.
- Enable OAuth Settings:
    - O Check Enable OAuth Settings.
    - O Callback URL: Enter a valid URL (e.g., https://login.salesforce.com/services/oauth2/callback).
    - O Selected OAuth Scopes: Add required scopes, such as Full Access (full), Access and manage your data (api).
- Save and Continue:
    - O After saving, Salesforce will provide a Consumer Key and Consumer Secret. Store these securely.
- Adjust Policies if Necessary:
    - O Configure IP Relaxation and Permitted Users based on security requirements

Step 2: Configuration for Service Now OAuth 2.0

- Navigate to OAuth Applications:
    - O Go to System OAuth > Application Registry.
- Create a New OAuth Application:
    - O Click New.
    - O Select Create an OAuth API endpoint for external clients.
- Configure Application Details:
    - O Name: e.g., "Salesforce Integration".
    - O Client ID: Auto-generated or custom.
    - O Client Secret: Auto-generated or custom.
    - O Redirect URL: As required by Salesforce or your integration setup.
    - O Token Lifetime: Set appropriate token expiry duration.

- Save the Application:
  - O Store the Client ID and Client Secret securely.
- Assign Roles and Permissions:
  - O Ensure the integration user has the necessary roles to access required ServiceNow resources

B. Using Service Now Integration Hub with Salesforce

Integration Hub [5] is ServiceNow’s powerful tool for creating integrations with minimal coding. Here's how to set it up with Salesforce.

Step 1: Enable Integration Hub in ServiceNow

- Navigate to Integration Hub
  - O Log into your ServiceNow instance.
  - O Use the Application Navigator to go to All > Integration Hub.
- Activate Required Plugins
  - O Go to System Applications > All Available Applications > All.
  - O Search for Integration Hub and activate it.
  - O Activate the Salesforce Spoke for pre-built Salesforce connectors.

Step 2: Set Up a Connection to Salesforce

- Navigate to Connections & Credentials:
  - O Go to Integration Hub > Connections & Credentials > Connections.
- Create a New Connection:
  - O Click New.
  - O Select Salesforce from the Spoke dropdown.
- Configure Connection Details:
  - O Name: Enter a descriptive name (e.g., "Salesforce Connection").
  - O OAuth Credentials: Choose OAuth 2.0 as the authentication type.
  - O Client ID & Client Secret: Obtain these from your Salesforce Connected App.
  - O Token URL & Authorization URL: Typically, these are provided by Salesforce.
  - O Username & Password: Your Salesforce credentials or use a designated integration user.
  - O Scopes: Define required scopes (e.g., api, refresh_token).
  - O Test the Connection:
  - O Click Test Connection to ensure successful authentication.

Step 3: Create Integration Actions [3]

- Access Integration Hub Spokes:
  - O Go to Integration Hub > Spokes > Salesforce.
- Explore Available Actions:
  - O Actions include Create Record, Update Record, Query Records, etc.
- Create a New Flow:
  - O Navigate to Flow Designer via All > Flow Designer.
- Design the Flow:
  - O Trigger: Define the event that starts the flow (e.g., when a new incident is created in ServiceNow).
  - O Action: Use the Salesforce spoke to perform desired operations (e.g., create a Salesforce Case).
- Configure Action Parameters:
  - O Map ServiceNow incident fields to Salesforce Case fields.
- Save and Activate the Flow:
  - O Name your flow and activate it.

Step 4: Test the Integration

- Create a Test Incident in ServiceNow:
  - O Ensure it triggers the flow.
- Verify in Salesforce:
  - O Check that a corresponding Case is created with correct data.
- Monitor Flow Execution:
  - O Use Flow Designer’s execution details to troubleshoot any issues.

C. Using MuleSoft Any Point Platform

MuleSoft [8] offers a robust environment for integrating complex workflows between ServiceNow and Salesforce.

Step 1: Set Up MuleSoft Any Point Platform

- Create a MuleSoft Account:
  - O Sign up at MuleSoft Any point Platform.
- Access Any point Studio:
  - O Download and install Any point Studio, MuleSoft’s IDE for building integrations.

Step 2: Design the Integration Flow

- Create a New Mule Project:
  - O Open Any point Studio.
  - O Go to File > New > Mule Project.
  - O Name your project (e.g., "ServiceNow_Salesforce_Integration").
- Add Connectors for ServiceNow and Salesforce:
  - O In the Palette, search for ServiceNow Connector and Salesforce Connector.
  - O Drag and drop them onto the canvas.
- Configure ServiceNow Connector [9]:
  - O Authentication: Use OAuth 2.0 or basic authentication.
  - O Instance URL: Your ServiceNow instance URL.
  - O Credentials: Integration user credentials.
- Configure Salesforce Connector [10]:
  - O Authentication Type: OAuth 2.0.
  - O Client ID & Client Secret: From Salesforce Connected App.
  - O Username & Password: Salesforce integration user.

Step 3: Implement Data Mapping and Transformation [4]

- Define Triggers:
  - O For example, a Scheduler to poll ServiceNow for new incidents.
- Retrieve Data from ServiceNow:
  - O Use GET Incident operation to fetch new incidents.
- Transform Data:
  - O Use Data Weave to map ServiceNow incident fields to Salesforce Case fields.
- Create or Update Salesforce Records:
  - O Use Create or Upsert operations in Salesforce Connector to manage Cases.

Step 4: Error Handling and Logging

- Add Error Handlers:
  - O Implement Try-Catch scopes to manage exceptions.
- Configure Logging:
  - O Use Logger components to record successful operations and errors.

Step 5: Deploy the Integration

- Test Locally:
  - O Run the Mule application in Any point Studio to ensure functionality.
- Deploy to Any point Platform:
  - O Connect Any point Studio to your MuleSoft account.
  - O Deploy the application to Cloud Hub or your preferred environment.
- Monitor Integration:
  - O Use MuleSoft’s monitoring tools to oversee performance and troubleshoot issues.

D. Building Custom API Integration

For highly customized requirements, developing a bespoke integration using ServiceNow and Salesforce APIs is ideal.

Step 1: Plan the Integration

- Define Use Cases:

O Example: Sync ServiceNow incidents to Salesforce Cases and vice versa.

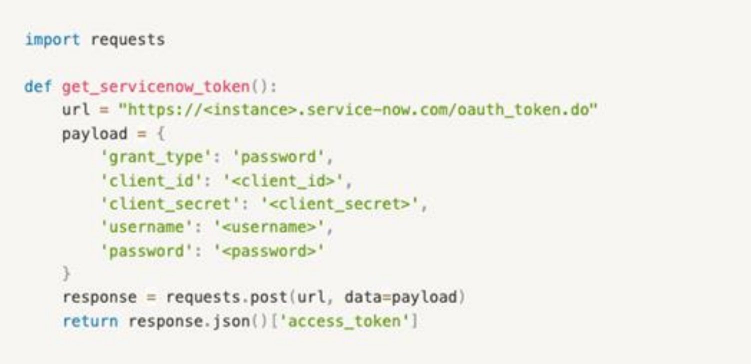
- Map Data Fields:
  O Identify corresponding fields between ServiceNow and Salesforce.
- Choose Technology Stack:
  O Select programming languages (e.g., Python, Node.js, Java).

Step 2: Set Up Development Environment

- Install Necessary Tools:
  O Install your chosen language’s runtime (e.g., Python 3.x).
  O Install IDE or text editor (e.g., Visual Studio Code).
- Install Required Libraries:
  O For Python: requests, simple-salesforce [11], etc.
  O For Node.js: axios, jsforce, etc.

Step 3: Authenticate with ServiceNow and Salesforce APIs

- ServiceNow Authentication:
  O Use OAuth 2.0 or Basic Authentication.
  O Example (Python with OAuth):

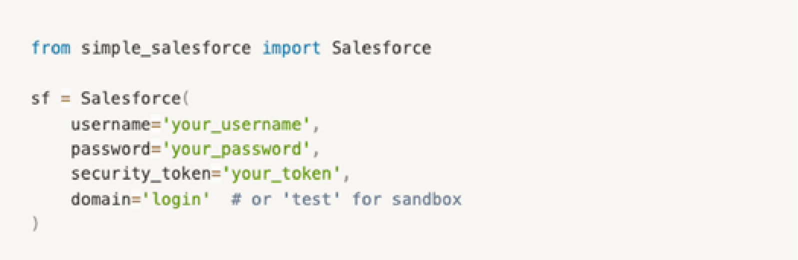


```python
import requests

def get_servicenow_token():
    url = "https://<instance>.service-now.com/oauth_token.do"
    payload = {
        'grant_type': 'password',
        'client_id': '<client_id>',
        'client_secret': '<client_secret>',
        'username': '<username>',
        'password': '<password>'
    }
    response = requests.post(url, data=payload)
    return response.json()['access_token']
```

Figure 1: *OAuth Implementation in Python for Service Now*

- Salesforce Authentication:
  O Use OAuth 2.0.
  O Example (Python with simple-salesforce) [11]:



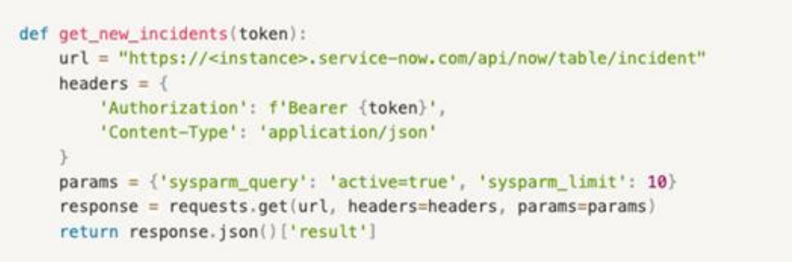
```python
from simple_salesforce import Salesforce

sf = Salesforce(
    username='your_username',
    password='your_password',
    security_token='your_token',
    domain='login'  # or 'test' for sandbox
)
```

Figure 2: *OAuth Implementation in Python for Salesforce*

Step 4: Develop Data Retrieval and Synchronization Logic

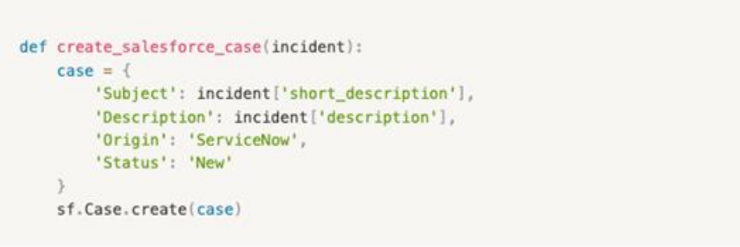
- Fetch Data from ServiceNow:



```python
def get_new_incidents(token):
    url = "https://<instance>.service-now.com/api/now/table/incident"
    headers = {
        'Authorization': f'Bearer {token}',
        'Content-Type': 'application/json'
    }
    params = {'sysparm_query': 'active=true', 'sysparm_limit': 10}
    response = requests.get(url, headers=headers, params=params)
    return response.json()['result']
```

Figure 3: *Python Implementation to Fetch Service Now Data*

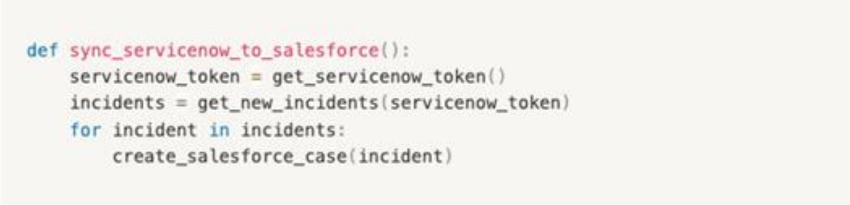
- Create Salesforce Cases:



```python
def create_salesforce_case(incident):
    case = {
        'Subject': incident['short_description'],
        'Description': incident['description'],
        'Origin': 'ServiceNow',
        'Status': 'New'
    }
    sf.Case.create(case)
```

Figure 4: *Python Implementation to Create Case in Salesforce*

- Main Synchronization Function:



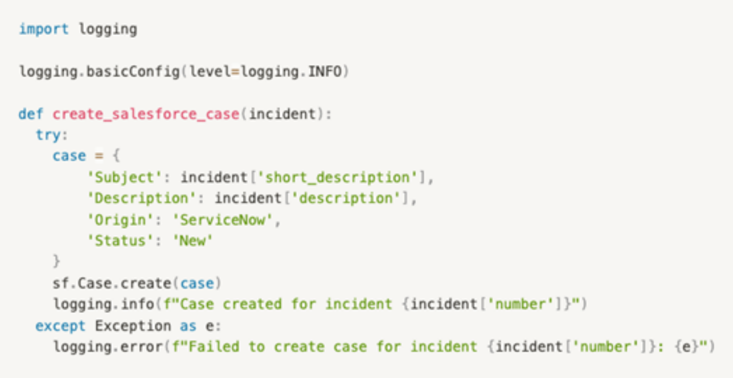
```python
def sync_servicenow_to_salesforce():
    servicenow_token = get_servicenow_token()
    incidents = get_new_incidents(servicenow_token)
    for incident in incidents:
        create_salesforce_case(incident)
```

Figure 5: *Main Function to orchestrate the functionality*

- Schedule the Synchronization:
    O Use CRON jobs, task schedulers, or serverless functions to run the synchronization at desired intervals.

Step 5: Implement Error Handling and Logging

- Error Handling:
    O Use try-except blocks to catch and handle exceptions.



```python
import logging

logging.basicConfig(level=logging.INFO)

def create_salesforce_case(incident):
  try:
    case = {
        'Subject': incident['short_description'],
        'Description': incident['description'],
        'Origin': 'ServiceNow',
        'Status': 'New'
    }
    sf.Case.create(case)
    logging.info(f"Case created for incident {incident['number']}")
  except Exception as e:
    logging.error(f"Failed to create case for incident {incident['number']}: {e}")
```

Figure 5: *Exception Handling in Python*

- Logging:
    O Log successful operations and errors to a file or monitoring system.

Step 6: Deploy and Test

- Deploy the Script:
    O Host on a server, cloud function, or your preferred environment.
- Run Tests:
    O Execute the synchronization manually first to verify functionality.
    O Check Salesforce for created Cases.
- Monitor Logs:
    O Ensure logs are capturing relevant information and errors are handled appropriately.

E. Configuring Workflows and Automation

Once the integration is established, setting up specific workflows and automations ensures that processes run smoothly and efficiently.

Example Workflow: Automatic Case Creation in Salesforce from ServiceNow Incidents

Step 1: Define Trigger in Service Now

• Event: New Incident Created.

• Conditions: Incident severity is high.

Step 2: Configure the Integration to Listen for the Trigger

• Using Integration Hub:

O Set the Flow Designer to trigger on new incident creation.

• Using MuleSoft:

O Configure the Mule flow to poll ServiceNow for new incidents periodically or use webhooks.

• Custom Integration:

O Implement event listeners or use ServiceNow's Webhooks to notify your integration script.

Step 3: Map Incident Data to Salesforce Case Fields

• ServiceNow Fields:

O number

O short_description

O description

O priority

O assigned_to

• Salesforce Fields:

O Case Number (auto-generated)

O Subject (map from short_description)

O Description

O Priority

O Owner (map from assigned_to)

Step 4: Implement Data Transformation

• Convert Data Types: Ensure data types match between platforms (e.g., priority levels).

• Handle Lookups: Map ServiceNow users to Salesforce users, possibly using a lookup table.

Step 5: Automate Status Synchronization

• Bidirectional Updates: If an incident is resolved in ServiceNow, update the corresponding Salesforce Case status to "Closed".

• Implement Triggers:

O ServiceNow to Salesforce: On incident status change.

O Salesforce to ServiceNow: On Case status change.

Step 6: Notify Relevant Stakeholders

• Send Notifications: Use Salesforce’s email alerts or ServiceNow’s notification system to inform users of updates.

• Integration Hub Actions: Use actions to send emails or trigger other notifications.

Step 7: Handle Data Enrichment

• Fetch Additional Data: Retrieve related records or metadata to enrich Salesforce Cases.

• Use ServiceNow Knowledge Base: Attach relevant knowledge articles to Salesforce Cases.

7. Testing

Comprehensive testing ensures the integration works as intended and handles edge cases gracefully.

A. Unit Testing

• **Test Individual Components:** Validate that each part of the integration (e.g., API calls) works independently.

• **Mock External Services:** Use mocking tools to simulate ServiceNow and Salesforce responses.

B. Integration Testing

• End-to-End Scenarios: Test complete workflows from ServiceNow to Salesforce and vice versa.

• Data Integrity: Verify that data is accurately transferred and mapped between systems.

• Error Scenarios: Simulate API failures, timeouts, and incorrect data to ensure the integration handles them gracefully.

C. User Acceptance Testing (UAT)

- Engage End-Users: Involve stakeholders to validate the integration meets business requirements.
- Gather Feedback: Collect and incorporate feedback to refine the integration.

D. Performance Testing

- Load Testing: Assess how the integration performs under high data volumes.
- Scalability Testing: Ensure the integration can scale as data volumes grow.

E. Security Testing

- Vulnerability Scanning: Identify and address security vulnerabilities in the integration.
- Penetration Testing: Conduct penetration tests to assess the security posture.

8. Security and Compliance

Security is paramount when integrating two enterprise systems. Follow these guidelines to protect data and comply with regulations.

A. Authentication and Authorization

- Use Secure Authentication Methods: Prefer OAuth 2.0 over Basic Authentication for enhanced security.
- Least Privilege Principle: Create dedicated integration users with only necessary permissions in both ServiceNow and Salesforce.
- Token Management: Securely store and manage API tokens and credentials. Implement token rotation policies if supported.

B. Data Encryption

- Encrypt Data in Transit: Use HTTPS for all API communications to ensure data encryption.
- Encrypt Sensitive Data at Rest: Ensure that any stored data (e.g., logs) is encrypted as per organizational policies.

C. Network Security

- Firewall Configurations: Allow only necessary traffic between ServiceNow, Salesforce, and middleware platforms.
- IP Whitelisting: Restrict API access to known IP addresses.

D. Compliance

- Data Protection Regulations: Ensure the integration complies with GDPR, HIPAA, CCPA, or other relevant regulations.
- Data Residency: Respect data residency requirements by controlling where data is stored and processed.

E. Audit and Logging

- Maintain Audit Logs: Record all integration activities for auditing purposes.
- Monitor Access: Track who accessed or modified integration configurations.

9. Monitoring and Maintenance

Continuous monitoring and maintenance are essential to ensure the integration remains robust and efficient.

A. Monitoring Tools

- Integration Platform Dashboards: Use dashboards provided by Integration Hub, MuleSoft, or other platforms to monitor integration health.
- Custom Monitoring: Implement custom logging and alerting mechanisms if using custom integrations.

B. Alerts and Notifications

- Set Up Alerts: Configure alerts for integration failures, high latency, or unusual activity.
- Notification Channels: Use email, SMS, or collaboration tools (e.g., Slack) to receive alerts.

C. Regular Maintenance

- Update Connectors and APIs: Keep integration connectors and APIs up to date with the latest versions.
- Review Logs Regularly: Periodically check logs for any anomalies or recurring issues.
- Optimize Performance: Identify and address performance bottlenecks.

D. Handling Platform Upgrades

- Monitor ServiceNow and Salesforce Releases: Stay informed about updates or changes that might affect the integration.

- Adjust Integration Accordingly: Modify the integration to accommodate any platform changes.

E. Backup and Recovery

- Data Backup: Regularly back up integration configurations and any stored data.
- Disaster Recovery Plan: Develop and test a plan to recover from integration failures or data loss.

10. Best Practices

Adhering to best practices enhances the efficiency, reliability, and maintainability of your integration.

A. Start with Clear Objectives

- Define Goals: Understand what you aim to achieve with the integration.
- Scope Management: Begin with essential use cases and expand gradually.

B. Documentation

- Integration Architecture: Document the overall architecture, including data flows and component interactions.
- Configuration Details: Keep records of all configurations, mappings, and transformations.
- Operational Procedures: Document maintenance routines, monitoring setups, and troubleshooting steps.

C. Use Version Control

- Track Changes: Use version control systems (e.g., Git) to manage integration code and configurations.
- Collaborative Development: Facilitate teamwork and code reviews through version control.

D. Modular Design

- Separation of Concerns: Design the integration in modular components to simplify maintenance and scalability.
- Reusable Components: Create reusable modules for common tasks like authentication, data transformation, etc.

E. Error Handling and Retries

- Graceful Failures: Ensure the integration can handle errors without crashing.
- Retry Mechanisms: Implement retries for transient failures like network issues.

F. Security First

- Protect Credentials: Store API keys and secrets securely, using environment variables or secure vaults.
- Regular Audits: Periodically review security configurations and access controls.

G. Performance Optimization

- Efficient Data Handling: Optimize data queries and minimize unnecessary data transfers.
- Asynchronous Processing: Use asynchronous processing where appropriate to enhance performance.

H. Stakeholder Collaboration

- Cross-Functional Teams: Involve IT, business, and end-users in the integration process.
- Feedback Loops: Establish channels for continuous feedback and improvement.

11. Conclusion

ServiceNow and Salesforce are powerful platforms that address distinct but complementary aspects of business operations. ServiceNow excels in managing internal workflows, IT services, and operational efficiencies, while Salesforce shines in managing customer relationships, sales, marketing, and external service delivery. By understanding the strengths and capabilities of each platform, organizations can effectively leverage them individually and in tandem to drive comprehensive digital transformation, enhance operational efficiency, and deliver superior customer experiences.

Integrating ServiceNow with Salesforce to automate processes can dramatically improve operational efficiency, data consistency, and cross-departmental collaboration. This comprehensive implementation guide has walked you through multiple integration methods, detailed configurations, security considerations, testing strategies, and best practices to ensure a successful integration. By following this guide and adhering to best practices, you can create a robust and scalable integration between ServiceNow and Salesforce that drives business value and enhances your organization's operational capabilities.

Integrating ServiceNow with Salesforce unlocks synergistic benefits, enabling seamless data flow and automation across IT and customer-facing functions. This integration fosters a more unified approach to managing both internal operations and external customer interactions, ultimately contributing to improved business outcomes and competitive advantage.

References

[1]. Apex Developer Guide - https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_dev_guide.htm

[2]. OAuth2.0 Documentation - https://oauth.net/2/

[3]. Service Now Integration Actions - https://docs.servicenow.com/bundle/vancouver-source-to-pay-operations/page/product/source-to-pay-operations/concept/integration-hub-actions.html

[4]. Service Now Developers Guide - https://developer.servicenow.com/dev.do#!/guides/xanadu/now-platform/tpb-guide/scripting_technical_best_practices

[5]. Service Now Integration Hub -https://www.servicenow.com/products/integration-hub.html

[6]. Connected App Creation Guide - https://help.salesforce.com/s/articleView?id=sf.connected_app_create.htm&type=5

[7]. Connected Apps Documentation - https://help.salesforce.com/s/articleView?id=sf.connected_app_overview.htm&type=5

[8]. MuleSoft API Management - https://www.mulesoft.com/platform/anypoint-platform-features

[9]. MuleSoft Service Now Connector - https://www.mulesoft.com/exchange/com.mulesoft.connectors/mule-servicenow-connector/

[10]. MuleSoft Salesforce Connector - https://www.mulesoft.com/exchange/com.mulesoft.connectors/mule-salesforce-connector/

[11]. Python Simple Salesforce - https://pypi.org/project/simple-salesforce/