

salesforce

インテグレーションのパターンと実践

バージョン 43.0, Summer '18



 @salesforcedocs

最終更新日: 2018/04/20

本書の英語版と翻訳版で相違がある場合は英語版を優先するものとします。

© Copyright 2000–2018 salesforce.com, inc. All rights reserved. Salesforce およびその他の名称や商標は、salesforce.com, inc. の登録商標です。本ドキュメントに記載されたその他の商標は、各社に所有権があります。

目次

はじめに	1
第 1 章: インテグレーションパターンの概要	1
本書の目的と範囲	1
パターンテンプレート	1
パターンの概要	3
パターン手法	3
パターン選択ガイド	4
ミドルウェアの用語と定義	5
設計パターンのカタログ	10
第 2 章: リモートプロセスの呼び出し — 要求と返信	10
第 3 章: リモートプロセスの呼び出し — ファイアアンドフォーゲット	18
第 4 章: バッチデータの同期	30
第 5 章: リモートコールイン	37
第 6 章: データの変更に基づく UI の更新	48
付録	51
付録 A: リソース — 外部	51
付録 B: リソース — Salesforce	52
付録 C: セキュリティに関する考慮事項	53

はじめに

第1章 インテグレーションパターンの概要

Salesforce を実装すると、他のアプリケーションとのインテグレーションが頻繁に必要になります。インテグレーションシナリオがそれぞれ違っていても、開発者が解決しなければならない要件や問題は共通している場合があります。

このドキュメントでは、こうした数々のインテグレーションシナリオに共通する要件と問題に対する戦略を(パターン形式で)説明します。各パターンでは、特定の実装ではなく、特定のシナリオについて設計と方法を説明します。このドキュメントでは、次の内容を取り上げます。

- 主要な「典型的」インテグレーションシナリオに対処する複数のパターン
- ユーザのシナリオに最適なパターンを判断するのに役立つ選択マトリックス
- インテグレーションのヒントとベストプラクティス

本書の目的と範囲

このドキュメントは、Lightning Platform を社内の他のアプリケーションと統合する必要があるデザイナーとアーキテクトを対象としています。このドキュメントでは、Salesforce アーキテクトおよびパートナーによる数多くの実装の成功例から、本質的なものを抜き出して紹介しています。

Salesforce ベースのアプリケーション(または Lightning Platform や Database.com ツールセット)の大規模な採用を検討している場合は、パターンの概要と選択マトリックスを参照すると、使用可能なインテグレーション機能とオプションを理解できます。Salesforce インテグレーションプロジェクトの設計および実装フェーズでは、アーキテクトと開発者は、こうしたパターンの詳細とベストプラクティスを検討する必要があります。

こうしたパターンを適切に実装すれば、可能な限り早く本番を開始し、安定して拡張性が高く、ほぼメンテナンス不要のアプリケーションを開発できます。Salesforce 独自のコンサルティングアーキテクトは、アーキテクチャレビューでこうしたパターンを参照基準として使用し、積極的にパターンをメンテナンスして改良しています。

これらのパターンは、ほとんどのインテグレーションシナリオに対応していますが、シナリオすべてに対応しているわけではありません。Salesforce ではユーザインターフェース(UI)インテグレーション(マッシュアップなど)を許可していますが、こうしたインテグレーションにはこのドキュメントでは触れていません。自分の要件が、こうしたパターンが規定する範囲外であると思われる場合は、Salesforce 担当者にお問い合わせください。

パターンテンプレート

各インテグレーションパターンの構造は一貫しています。これにより、各パターンで提供される情報の一貫性を確保でき、パターンを比較するのが容易になります。

名前

パターンに含まれるインテグレーションのタイプを示すパターン識別子。

コンテキスト

パターンを使用して対応できるインテグレーションシナリオの全容。コンテキストには、ユーザが実現しようとしている内容や、要件を満たすためのアプリケーションの動作方法に関する情報が含まれます。

問題

このパターンで解決するシナリオまたは問題(質問形式で提示)。パターンを確認する場合、このセクションを読めば、パターンがインテグレーションシナリオに適しているかどうかをすばやく判断できます。

検討項目

規定のシナリオによる解決が難しくなる制約や状況。

ソリューション

インテグレーションシナリオを解決するための推奨方法。

概要図

ソリューションでどのようにシナリオに対応するのかを示す UML シーケンスダイアグラム。

結果

ソリューションをインテグレーションシナリオに適用する方法やそのシナリオに関連付けられている検討項目の解決方法について詳細に説明します。このセクションには、パターンの適用後に生じる可能性のある新たな課題も含まれます。

補足

パターンに関するその他のセクション(重要な技術的な問題、パターンのバリエーション、パターン固有の懸念など)。

例

実際の Salesforce シナリオで設計パターンがどのように使用されるのかを示すエンドツーエンドシナリオ。この例では、インテグレーションの目標や、目標達成のためにどのようにパターンを実装するのかについて説明します。

パターンの概要

次の表に、このドキュメントに含まれるインテグレーションパターンの一覧を示します。

パターンのリスト

パターン	シナリオ
リモートプロセスの呼び出しー要求と返信	Salesforce は、リモートシステムでプロセスを呼び出して、そのプロセスが完了するまで待機し、リモートシステムからの応答に基づいて状態を追跡します。
リモートプロセスの呼び出しーファイアアンドフォワード	Salesforce は、リモートシステムでプロセスを呼び出しますが、プロセスが完了するまで待機しません。代わりに、リモートプロセスが要求を受信して確認し、制御を Salesforce に戻します。
バッチデータの同期	外部システムの更新を反映させるには、Lightning Platform に保存されるデータを作成または更新する必要があります。また、逆に Lightning Platform で変更が発生した場合は、それを外部システムに送信する必要があります。いずれの場合も更新はバッチ形式で行われます。
リモートコールイン	Lightning Platform に保存されるデータは、リモートシステムによって作成、取得、更新、削除されます。
データの変更に基づく UI の更新	Salesforce ユーザーインターフェースは、Salesforce データが変更されたときに、自動的に更新される必要があります。

パターン手法

このドキュメントのインテグレーションパターンは、次の2つのカテゴリに分類されます。


- データインテグレーション**—これらのパターンは、複数のシステムに存在するデータを同期して両方のシステムに常に適時性と意味のあるデータが含まれるようにするための要件に対応します。データインテグレーションの多くは、最も実装が簡単な種類のインテグレーションですが、ソリューションを持続可能でコスト効果の高いものにするために適切な情報管理技法が必要です。こうした技法には、多くの場合、マスタデータ管理 (MDM)、データガバナンス、マスタリング、重複排除、データフロー設計などの要素が含まれます。
- プロセスインテグレーション**—このカテゴリのパターンは、複数のアプリケーションを利用してタスクを完了するためのビジネスプロセスのニーズに対応します。この種類のインテグレーションのソリューションを実装する場合、トリガ側のアプリケーションがプロセスの境界を越えて他のアプリケーションをコールする必要があります。通常、これらのパターンには、オーケストレーション (1つのアプリケーションが中央「コントローラ」になる) とコレオグラフィ (アプリケーションが複数同時参加し、中央「コントローラ」がない) も含まれます。これらの種類のインテグレーションでは、複雑な設計、テスト、および例外処理要件が必要になることがよくあります。また、このような複合アプリケーションは通常、長時間実行されるトランザクションと、レポート作成やプロセス状態の管理機能をサポートするため、基礎となるシステムへの要求が厳しくなります。

システムに最適なインテグレーション戦略を選択することは簡単ではありません。考慮すべき要素や、使用できるツールが多数あるからです。また、タスクによって適切なツールも異なります。各パターンは、システム、データ量、障害対応、トランザクション性の各機能を含む、特定の重要な領域に対応します。

パターン選択ガイド

この選択マトリックス表には、パターンと主要な要素の一覧が含まれており、インテグレーション要件に最適なパターンを判断するのに役立ちます。マトリックス内のパターンは、次の要素で分類されます。

要素	説明
要求元/要求先	インテグレーショントランザクションの要求者とその情報を提供する要求先を指定します。要求元システムと要求先システムの技術的な機能と、インテグレーションの種類とタイミングによっては、追加のミドルウェアやインテグレーションソリューションが必要になる場合があります。詳細は、各パターンの説明を参照してください。
種類	<p>インテグレーションのスタイルを指定します(プロセス、データのいずれか)。</p> <ul style="list-style-type: none"> プロセス—プロセスベースのインテグレーションは、2つ以上のアプリケーション間の機能的な処理フローを統合する方法と定義できますこうしたインテグレーションでは通常、抽象化と複雑さのレベルが高くなります。特にトランザクション性とロールバックについてはその傾向があります。 データ—データインテグレーションは、アプリケーションで使用される情報のインテグレーションと定義できます。これらのインテグレーションは、単純なテーブル挿入や更新/挿入から、参照整合性と複雑な変換を必要とする複雑なデータ更新まで多岐にわたります。
タイミング	<p>インテグレーションのブロック(または非ブロック)特性を指定します。</p> <ul style="list-style-type: none"> 同期—ブロック、つまり「ほぼリアルタイム」の要求は、「要求/応答操作によって呼び出され、プロセスの結果はこの操作を経由して即時に呼び出し側に戻される」と定義できます。¹ 非同期—非ブロック、キュー、またはメッセージベースの要求は、「片方向操作で呼び出され、結果およびすべての障害は、別の片方向操作を呼び出して戻されます」。²そのため、コール元は要求を行った後、応答を待たずに処理を続行します。

 **メモ:** インテグレーションには、インテグレーションシナリオに適用される要素に応じて、外部のミドルウェアやインテグレーションソリューション(エンタープライズサービスバスなど)が必要になる場合があります。

¹ 「同期、および非同期プロセス」、IBM Corporation、最終アクセス日: 2012年5月18日、
<http://publib.boulder.ibm.com/infocenter/adiehelp/v5r1m1/index.jsp?topic=%2Fcom.ibm.etools.ctc.flow.doc%2Fconcepts%2Fcsynchf.html>.

² 同書。

パターン選択マトリックス

次の表には、パターンと主要な要素の一覧が含まれており、インテグレーション要件に最適なパターンを判断するのに役立ちます。

要求元/要求先	種類		タイミング		主要な検討対象パターン
	プロセスインテグレーション	データインテグレーション	同期	非同期	
Salesforce -> システム	X		X		リモートプロセスの呼び出し — 要求と返信
				X	リモートプロセスの呼び出し — ファイアアンドフォーゲット
		X	X		リモートプロセスの呼び出し — 要求と返信
				X	データの変更に基づく UI の更新
システム -> Salesforce	X		X		リモートコールイン
				X	リモートコールイン
		X	X		リモートコールイン
				X	バッチデータの同期

ミドルウェアの用語と定義

次の表は、ミドルウェアに関連する主要な用語と、各パターンで使用される場合の定義の一覧です。

用語	定義
イベント処理	<p>イベント処理とは、識別可能な発生イベントが指定された受信者(「ハンドラ」)で受信されることです。イベント処理に含まれる主要なプロセスは次のとおりです。</p> <ul style="list-style-type: none"> • イベントをどこに転送すべきか識別する。 • その転送アクションを実行する。 • 転送されたイベントを受信する。 • ログへの書き込み、エラーまたは回復プロセスの送信、追加メッセージの送信など、受信したイベントに応じて適切なアクションを実行する。 <p>イベントハンドラは、最終的にイベントをイベントコンシューマに転送します。この機能のミドルウェアでの一般的な使用方法を拡張し、よく使われる「公開/登録」(pub/sub)機能を追加することもできます。公開/登録シナリオでは、ミド</p>

用語	定義
	<p>ルウェアが要求またはメッセージを有効なデータイベント公開者から有効なデータイベント登録者にルーティングします。それを受けて、有効なリスナーを持つこれらのコンシューマが、公開されたイベントを取得できます。</p> <p>ミドルウェアを使用した Salesforce インテグレーションでは、イベント処理の制御をミドルウェアレイヤが担い、すべての関連イベント (同期および非同期) を収集して、Salesforce を含むすべてのエンドポイントへの配布を管理します。</p>
<p>プロトコル変換</p>	<p>プロトコル変換とは、「通常、ソフトウェアアプリケーションによって、あるデバイスの標準または独自プロトコルを別のデバイスに適したプロトコルに変換し、相互運用性を実現することです。</p> <p>ミドルウェアのコンテキストでは、特定のターゲットシステムへの接続はプロトコルによって制約されます。その場合、メッセージ形式を、ターゲットシステムの形式に変換またはカプセル化して、ターゲットシステムでペイロードを抽出できるようにする必要があります。これはトンネリングとも呼ばれます。」³</p> <p>Salesforce は、ネイティブプロトコル変換をサポートしていないため、こうした要件をミドルウェアレイヤかエンドポイントで処理することが前提となります。</p> <p>http://searchsoa.techtarget.com/definition/event-handlerを参照してください。</p>
<p>変換/加工</p>	<p>加工は、あるデータ形式を別の形式に対応付けて、インテグレーション対象のさまざまなシステム間で相互運用性を確保する機能です。通常、これには、送信者または受信者の要件に合うように配信途中でメッセージ形式を再設定することが含まれます。より複雑なケースとして、あるアプリケーションが独自のネイティブ形式でメッセージを送信し、他の複数のアプリケーションがそれぞれ独自のネイティブ形式でそのメッセージのコピーを受信する場合があります。</p> <p>ミドルウェア変換/加工ツールの多くには、レガシーのエンドポイントやその他の非標準エンドポイント用にサービスファサードを作成する機能が含まれています。この機能により、それらのエンドポイントがサービス対応可能であることがわかるようにすることができます。</p> <p>Salesforce インテグレーションを使用する場合、そうした要件がミドルウェアレイヤかエンドポイントで処理されることが前提となります。データの加工は、Apex でコーディングできますが、メンテナンスやパフォーマンスを考慮するとお勧めしません。</p> <p>http://en.wikipedia.org/wiki/Message-oriented_middlewareを参照してください。</p>
<p>キューイングとバッファリング</p>	<p>キューイングとバッファリングは、通常、要求-応答アーキテクチャとは対照的な非同期のメッセージの受け渡しに依存します。非同期システムでは、送信先プログラムがビジーか、接続に障害がある場合、メッセージキューが一時的な</p>

³ Gregor Hohpe、および Bobby Woolf: 『Enterprise Integration Patterns』 (Boston: Addison-Wesley Professional、2003 年)。

用語	定義
	<p>保管場所となります。さらに、ほとんどの非同期ミドルウェアシステムは、メッセージキューのバックアップ用に永続的なストレージを提供しています。</p> <p>非同期メッセージプロセスの主要な利点は、受信者アプリケーションに何らかの理由で障害が発生した場合、送信者が影響を受けずに処理を続行できることです。送信されたメッセージはそのままメッセージキューに蓄積され、後で受信者が処理を再開したら処理されます。</p> <p>Salesforce では、ワークフローベースのアウトバウンドメッセージ方式で明示的なキューイング機能のみを提供します。他のインテグレーションシナリオ(オーケストレーション、プロセスコレオグラフィ、サービスのシナリオなど)用に本格的なメッセージキューイングを提供するには、ミドルウェアソリューションが必要です。</p> <p>http://en.wikipedia.org/wiki/Message-oriented_middlewareを参照してください。</p>
同期伝送プロトコル	<p>同期伝送プロトコルとは、次のようなアクティビティをサポートするプロトコルのことです。「コール側の1つのスレッドが要求メッセージを送信し、遮断状態になって返信メッセージを待機し、受信した返信を処理します。(中略)。応答待ちの要求スレッドは、未処理の要求が1つのみであるか、このスレッドではこの要求の返信チャンネルが非公開であることを暗黙的に示します」。⁴</p>
非同期伝送プロトコル	<p>非同期伝送プロトコルとは、次のようなアクティビティをサポートするプロトコルのことです。「コール側の1つのスレッドが要求メッセージを送信し、応答用のコールバックを設定します。応答メッセージは別のスレッドによってリスンされます。応答メッセージを受け取ると、応答スレッドが適切なコールバックを起動し、そのコールバックによってコール側のコンテキストが再確立され、応答が処理されます。この方法により、複数の処理待ち要求が1つの応答スレッドを共有することが可能になります」⁵</p>
仲介ルーティング	<p>仲介ルーティングとは、コンポーネント間の複雑なメッセージフローを指定することです。たとえば、多くのミドルウェアベースのソリューションは、メッセージキューシステムに依存します。実装によって、メッセージングレイヤ自体でルーティングロジックを提供することを許可する場合と、クライアントアプリケーションにルーティング情報の提供を依存したり、両方の仕組みを併用したりする場合があります。こうした複雑なケースでは、(ミドルウェア上の)仲介機能によって、開発、インテグレーション、および検証が簡素化されます。</p> <p>「特に、仲介者はオブジェクトのグループを調整して、互いにどう調整されたかをオブジェクトが認識せずにするようにします。(中略)。その結果、各コンシューマが特定の種類のメッセージに集中でき、調整者[仲介者]は適切なメッセージを適切なコンシューマに確実に配信できます」⁶</p>

⁴ Gregor Hohpe、および Bobby Woolf: 『Enterprise Integration Patterns』 (Boston: Addison-Wesley Professional、2003年)。

⁵ (同書)。

⁶ (同書)。

用語	定義
<p>プロセスコレオグラフィとサービスオーケストレーション</p>	<p>プロセスコレオグラフィとサービスオーケストレーションはそれぞれ「サービスコンポジション」の形式であり、複数のエンドポイントと機能が調整されます。</p> <p>コレオグラフィとサービスオーケストレーションには次の違いがあります。</p> <ul style="list-style-type: none"> コレオグラフィとは、「中央制御なしで相互作用する個々のエンティティのグループから生じる動作」である。⁷ オーケストレーションとは、「互いに独立してタスクを実行する個々のエンティティの動作を調整する中央の指揮者から生じる動作」である。⁸ <p>「オーケストレーションは各サービスの完全な動作を示すのに対し、コレオグラフィでは各サービスのインターフェース動作記述が結合されます」。⁹</p> <p>ビジネスプロセスコレオグラフィの各部分は、Salesforce ワークフロー内で、または Apex を使用して構築できます。Salesforce にはタイムアウト値とガバナ制限があるため、複雑なオーケストレーションはすべてミドルウェアレイヤに実装することをお勧めします(特にトランザクション処理が必要なソリューションの場合)。</p>
<p>トランザクション性(暗号化、署名、信頼できる配信、トランザクション管理)</p>	<p>トランザクション性とは、必要な各リソースに対するすべての必要な操作を網羅するグローバルなトランザクションをサポートする機能と定義できます。トランザクション性は、4つのACID(原子性(atomicity)、一貫性(consistency)、独立性(isolation)、永続性(durability))プロパティすべてをサポートすることを暗黙的に示します。このうち原子性は、作業単位(トランザクション)について「オールオアナッシング」の結果を保証します。</p> <p>Salesforce は、内部的にトランザクション性を備えていますが、分散トランザクションや Salesforce の外部で開始されたトランザクションに参加することはできません。したがって、複雑な複数システムトランザクションが必要なソリューションについては、トランザクション性(および関連するロールバック/補正メカニズム)がミドルウェアレイヤに実装されることが前提となります。</p> <p>http://en.wikipedia.org/wiki/Distributed_transactionを参照してください。</p>
<p>ルーティング</p>	<p>ルーティングは、コンポーネント間の複雑なメッセージフローを指定することと定義できます。最新のサービスベースのソリューションでは、こうしたメッセージフローは、ヘッダー、コンテンツタイプ、ルール、優先度などの複数の条件に基づいている可能性があります。</p> <p>Salesforce インテグレーションを使用する場合、そうした要件がミドルウェアレイヤかエンドポイントで処理されることが前提となります。メッセージルーティ</p>

⁷ 「Choreography and Orchestration: A Software Perspective」(e-Zest、最終アクセス日: 2013年4月11日、<http://www.e-zest.net/blog/choreography-and-orchestration-a-software-perspective/>)。

⁸ 同書。

⁹ 「Orchestration vs. Choreography」(Stack Overflow、最終アクセス日: 2013年4月11日、<http://stackoverflow.com/questions/4127241/orchestration-vs-choreography>)。

用語	定義
	<p>ングは、Apex でコーディングできますが、メンテナンスおよびパフォーマンスを考慮するとお勧めしません。</p>
抽出、加工、読み込み (ETL)	<p>抽出、加工、読み込み (ETL) とは、次の操作を含むプロセスを指します。</p> <ul style="list-style-type: none"> • ソースシステムからデータを抽出する。これには通常、リレーショナル構造および非リレーショナル構造の複数のソースシステムからのデータが含まれます。 • 運用ニーズに合わせてデータを加工する。ニーズには、データ品質レベルが含まれる場合もあります。加工フェーズでは通常、一連のルールまたは関数をソースから抽出されたデータに適用して、最終ターゲットに読み込むデータを引き出します。 • データをターゲットシステムに読み込む。ターゲットシステムは、データベース、オペレーショナルデータストア、データマート、データウェアハウス、またはその他の運用システムなど、多種多様です。 <p>厳密に必要というわけではありませんが、多くの成熟した ETL ツールには変更データキャプチャ機能が搭載されています。この機能では、ツールがソースシステム内で前回の抽出以降に変更されたレコードを識別することで、レコード処理量を削減します。</p> <p>http://en.wikipedia.org/wiki/Extract,_transform,_load および http://en.wikipedia.org/wiki/Change_data_capture を参照してください。</p>

設計パターンのカタログ

第 2 章 リモートプロセスの呼び出し — 要求と返信

コンテキスト

Salesforce を使用して、リードの追跡、パイプラインの管理、商談の作成、リードを顧客に変換する注文詳細の取得を行っています。ただし、Salesforce システムには注文が含まれず、注文の処理も行われません。Salesforce で注文の詳細を取得したら、リモートシステムで注文が作成され、最後まで管理されます。

このパターンを実装する場合、Salesforce は注文を作成するリモートシステムをコールして、作成が正常に完了するまで待機します。正常に完了した場合、リモートシステムは注文状況と注文番号により同期して返信します。同じトランザクションの一部として、Salesforce は注文番号と状況を内部で更新します。注文番号は、リモートシステムへの後続の更新で外部キーとして使用されます。

問題

Salesforce でイベントが発生した場合に、リモートシステムでプロセスを開始して必要な情報をそのプロセスに渡し、リモートシステムから応答を受信して、その応答データを使用して Salesforce 内で更新を行う方法は？

検討項目

このパターンに基づいてソリューションを適用する場合、次のような項目を検討する必要があります。

- リモートシステムへのコールでは、Salesforce は処理を続行する前に、応答があるまで待機する必要があるか？リモートシステムへのコールは同期要求-返信と非同期要求のどちらなのか？
- リモートシステムへのコールが同期である場合、Salesforce が最初のコールと同じトランザクションの一部として応答を処理する必要はあるのか？
- メッセージサイズは小さいか、大きいのか？
- インテグレーションは、特定のイベント (Salesforce ユーザーインターフェースのボタンのクリックや DML ベースのイベントなど) の発生に基づいたものなのか？

解決方法

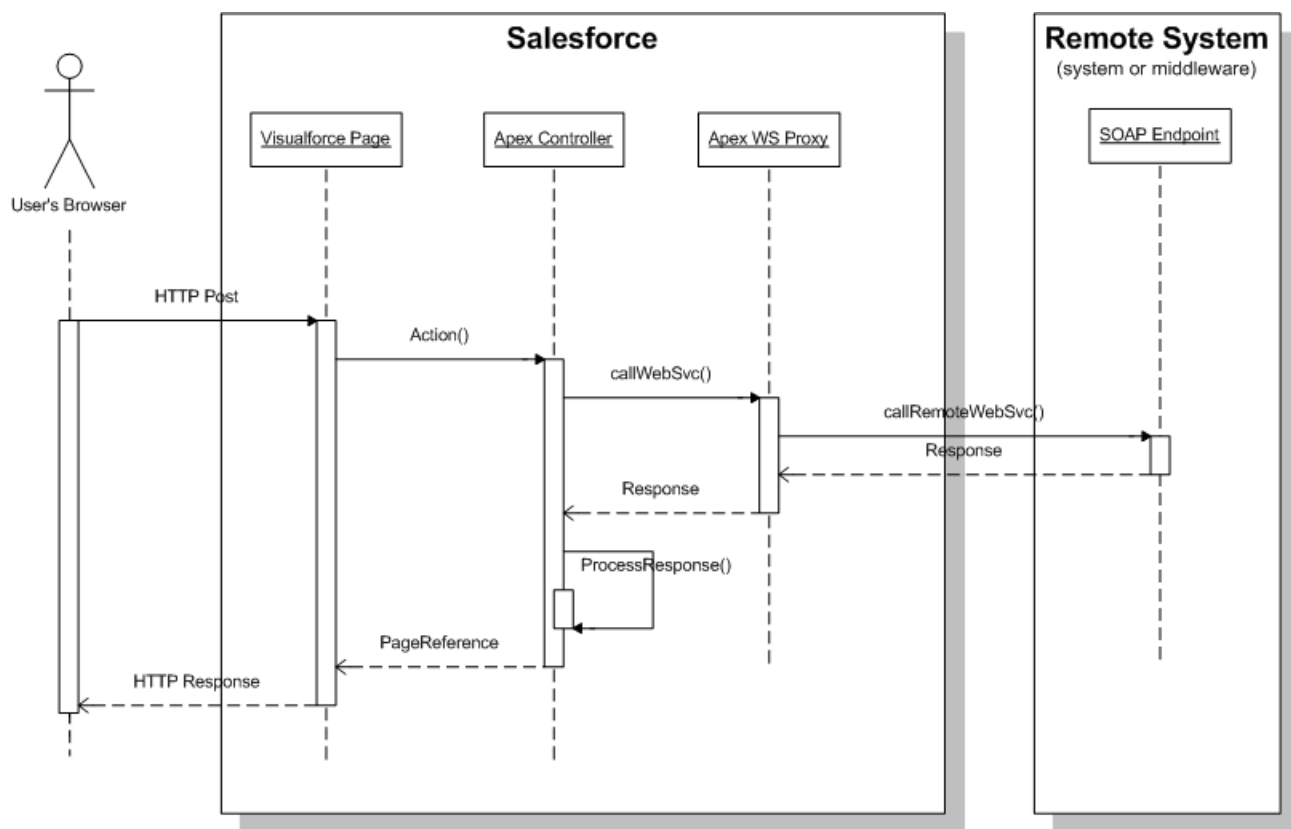
次の表は、このインテグレーションに関する問題へのソリューションの一覧です。

解決方法	適合度	コメント
Apex SOAP コールアウトを同期して開始するカスタム Visualforce ページまたはボタン	最適	<p>Salesforce では、WSDL を使用してプロキシ Apex クラスを生成できます。このクラスには、リモートサービスをコールするために必要なロジックがあります。</p> <p>Visualforce ページでユーザがアクションを開始すると、Apex コントローラアクションがコールされ、このプロキシ Apex クラスを実行してリモートコールが行われます。Visualforce ページでは、Salesforce アプリケーションをカスタマイズする必要があります。</p>
Apex HTTP コールアウトを同期して開始するカスタム Visualforce ページまたはボタン	最適	<p>Salesforce では、標準の GET、POST、PUT、DELETE メソッドを使用する HTTP サービスを呼び出すことができます。いくつかの HTTP クラスを使用して、RESTful サービスと統合できます。手動で SOAP メッセージを作成して SOAP ベースのサービスに統合することもできます。Salesforce では WSDL を使用してプロキシクラスを生成できるため、後者はお勧めできません。</p> <p>Visualforce ページでユーザがアクションを開始すると、Apex コントローラアクションがコールされ、このプロキシ Apex クラスを実行してリモートコールが行われます。Visualforce ページでは、Salesforce アプリケーションをカスタマイズする必要があります。</p>
Salesforce データの変更に基づいて呼び出され、Apex SOAP または HTTP コールアウトを同期して実行するトリガ	準最適	<p>Apex トリガを使用して、レコードデータの変更に基づく自動化を実行できます。</p> <p>Apex トリガを使用すれば、DML 操作の結果として Apex プロキシクラスを実行することができます。ただし、トリガコンテキスト内から作成されたすべてのコールは、開始イベントから非同期に実行する必要があります。そのため、これは、このインテグレーションの問題のソリューションとしてお勧めできません。これは、「リモートプロセスの呼び出し — ファイアアンドフォゲット」パターンに適したソリューションです。</p>
Apex SOAP または HTTP コールアウトを同期して実行する Apex 一括処理ジョブ	準最適	<p>一括処理ジョブでリモートシステムにコールできます。このソリューションにより、リモートバッチプロセスを実行したり、Salesforce でリモートシステムからの応答を処理したりできるようになり</p>

解決方法	適合度	コメント
		<p>ます。ただし、このバッチのコール数は制限されています。詳細は、「ガバナ制限」を参照してください。</p> <p>特定のバッチの実行で複数のトランザクションコンテキストを実行できます (通常、200 レコード間隔)。ガバナ制限は、トランザクションコンテキストごとにリセットされます。</p>

概要図

次の図は、Apex コールを使用したリモートプロセスの同期呼び出しを示します。



このシナリオの内容は次のとおりです。

1. Visualforce ページでアクションを開始します (ボタンのクリックなど)。
2. ブラウザは HTTP POST を実行し、対応する Apex コントローラに対してアクションを実行します。
3. コントローラは、以前に生成された Apex Web サービスプロキシクラスをコールします。
4. プロキシクラスは、リモート Web サービスへの実際のコールを実行します。

5. リモートシステムから Apex コントローラに応答が返されます。コントローラは応答を処理して、必要に応じて Salesforce のデータを更新し、ページを再表示します。

後続の状態を追跡する必要がある場合、リモートシステムは、Salesforce レコードに保存されている一意の識別子を返します。

結果

このパターンに関連するソリューションの適用では、イベントからリモートプロセスを呼び出すことができます。Salesforce がこのプロセスを処理します。

コールメカニズム

コールメカニズムは、このパターンを実装するために選択されるソリューションに応じて異なります。

コールメカニズム	説明
Visualforce および Apex コントローラ	<p>ユーザインターフェースを伴うエンドツーエンドプロセスの一部としてリモートプロセスをトリガし、その結果を Salesforce レコードで表示または更新する必要がある場合に使用します。たとえば、クレジットカードの支払いを外部の支払ゲートウェイに送信する場合はこれに該当します。この場合、支払い結果がすぐに返されてユーザに表示されます。</p> <p>ユーザインターフェースイベントからトリガされるインテグレーションの場合、一般的にカスタム Visualforce ページを作成する必要があります。</p>
Apex トリガ	<p>DML によって開始されたイベントからの Apex コールアウトを使用してリモートプロセスを呼び出す場合に、主に使用します。このコールメカニズムについての詳細は、「リモートプロセスの呼び出し—ファイアアンドフォーゲット」パターンを参照してください。</p>
Apex 一括処理クラス	<p>バッチでリモートプロセスを呼び出す場合に使用します。このコールメカニズムについての詳細は、「リモートプロセスの呼び出し—ファイアアンドフォーゲット」パターンを参照してください。</p>

エラー処理と回復

エラー処理および回復戦略を全体的なソリューションの一部として含めることが重要です。

- エラー処理 — エラーが発生した (例外やエラーコードがコール元に返された) 場合、エラーはコール元が処理します。たとえば、エラーメッセージがエンドユーザのページに表示またはテーブルに記録され、追加のアクションが求められます。
- エラー回復 — コール元が正常な応答を受信するまで、変更は Salesforce にコミットされません。たとえば、成功を示す応答を受信されるまでデータベースの注文状況は更新されません。必要に応じて、コール元は操作を再試行できます。

幕等設計に関する考慮事項

幂等機能によって、呼び出しを繰り返しても安全なことが保証されます。幂等性が実装されていない場合、同じメッセージを繰り返し呼び出すと異なる結果になり、レコードの重複作成、トランザクションの重複処理など、データ整合性に問題が生じる可能性があります。

コールされるリモートプロシージャの幂等性を確保することが重要です。Salesforce が 1 回のみコールすることを保証するのはほぼ不可能です (特に、ユーザインターフェースイベントに基づいてコールがトリガされる場合)。Salesforce によるコールが 1 回だとしても、他のプロセス (ミドルウェアなど) でも同様であるという保証はありません。

幂等なレシーバを作成する最も一般的な方法は、コンシューマから送信される一意のメッセージ識別子に基づいて重複を追跡することです。一意のメッセージ ID を送信するように Apex Web サービスまたは REST コールをカスタマイズする必要があります。

また、リモートシステムのレコードを作成する操作で、挿入前に重複をチェックする必要があります。Salesforce から一意のレコード ID を渡してチェックします。リモートシステムにレコードが存在する場合、レコードを更新します。大部分のシステムで、この操作は更新/挿入操作と呼ばれます。

セキュリティに関する考慮事項

リモートシステムへのコールでは、要求の機密性、整合性、および可用性を維持する必要があります。次に、このパターンで Apex SOAP および HTTP コールを使用する場合に固有のセキュリティに関する考慮事項を示します。

- デフォルトでは一方向 SSL が有効になっていますが、自己署名証明書と CA 署名証明書で双方向 SSL もサポートされており、クライアントとサーバの両方の信頼性を確保できるようになっています。
- Salesforce では、現在 WS-Security はサポートされていません。
- 必要に応じて、Apex Crypto クラスメソッドを使用する一方向ハッシュまたはデジタル署名によって、要求の整合性を確保することを検討してください。
- 適切なファイアウォールメカニズムを実装してリモートシステムを保護する必要があります。

「[セキュリティに関する考慮事項](#)」を参照してください。

補足

適時性

このパターンでは適時性は非常に重要です。通常は次の点に注意してください。

- 要求はユーザインターフェースから呼び出されるため、ユーザがプロセスを待機しないようにする必要があります。
- Salesforce では、Apex からのコールのタイムアウト (最大 60 秒) を設定できます。
- リモートプロセスの実行は、Salesforce のタイムアウト制限内やユーザの想定時間内に終わるように適時に完了する必要があります。

データ量

このパターンは、主に少量のリアルタイムアクティビティで使用されます。これは、Apex コールソリューションの要求または応答のタイムアウト値や最大サイズが比較的小さいためです。データペイロードがメッセージに含まれるバッチアクティビティではこのパターンを使用しないでください。

エンドポイント機能と標準のサポート

エンドポイントの機能と標準のサポートは、選択するソリューションによって異なります。

ソリューション	エンドポイントに関する考慮事項
Apex SOAP コールアウト	<p>エンドポイントで HTTP を介して Web サービスコールを受信できる必要があります。Salesforce は、公開インターネットを介してエンドポイントにアクセスできる必要があります。</p> <p>このソリューションでは、リモートシステムに Salesforce でサポートされている規格との互換性があることが求められます。現時点で Salesforce でサポートされている Apex SOAP コールアウトの Web サービス規格は次のとおりです。</p> <ul style="list-style-type: none"> • WSDL 1.1 • SOAP 1.1 • WSI-Basic Profile 1.1 • HTTP
Apex HTTP コールアウト	<p>エンドポイントで HTTP コールを受信できる必要があります。Salesforce は、公開インターネットを介してエンドポイントにアクセスできる必要があります。</p> <p>Apex HTTP コールアウトを使用して、標準の GET、POST、PUT、DELETE メソッドを使用する RESTful サービスをコールできます。</p>

状態管理

システムを統合する場合、継続的に状態を追跡するためにキーが重要になります。2つのオプションがあります。

- Salesforce が、リモートシステムのリモートレコードの主キーまたは一意のサロゲートキーを保存する。
- リモートシステムが、Salesforce の一意のレコード ID または他の一意のサロゲートキーを保存する。

次の表に示すように、マスタレコードを格納するシステムに応じて、インテグレーションキーの処理に関する固有の考慮事項があります。

マスタシステム	説明
Salesforce	リモートシステムは SalesforceRecordId またはその他の一意のサロゲートキーをレコードから保存します。
リモートシステム	リモートプロセスへのコールでアプリケーションから一意のキーが返され、Salesforce はそのキー値を一意のレコード項目に保存します。

複雑なインテグレーションシナリオ

場合によっては、このパターンで規定されるソリューションで複数の複雑なインテグレーションシナリオを実装する必要が生じることがあります。ミドルウェアを使用するか、Salesforce から複合サービスをコールするのが最適な方法になります。これらのシナリオには、次の作業が含まれます。

- ビジネスプロセスと複雑なフローロジックを伴うルールオーケストレーション

リモートプロセスの呼び出し — 要求と返信

- 複数のシステムへのコールにおけるコールとその結果の集約
- インバウンドメッセージとアウトバウンドメッセージの両方の加工
- 複数のシステムへのコールにおけるトランザクションの整合性の確保

ガバナ制限

Salesforce プラットフォームのマルチテナント性により、Apex コールアウトには制限があります。

説明	同期制限	非同期制限
発行される SOQL クエリの合計数 ¹	100	200
SOQL クエリによって取得されるレコードの合計数		50,000
Database.getQueryLocator によって取得されるレコードの合計数		10,000
発行される SOSL クエリの合計数		20
1つの SOSL クエリによって取得されるレコードの合計数		2,000
発行される DML ステートメントの合計数 ²		150
DML ステートメントの結果として処理されるレコードの合計数、Approval.process、または database.emptyRecycleBin		10,000
insert、update、または delete ステートメントによって繰り返しトリガする Apex 呼び出しのスタックの深さの合計 ³		16
トランザクション内のコールアウト (HTTP 要求または Web サービスコール) の合計数		100
トランザクション内のすべてのコールアウト (HTTP 要求または Web サービスコール) のタイムアウトの最大累積値		120 秒
Apex 呼び出し 1 回につき許可される future アノテーションを持つメソッドの最大数		50
System.enqueueJob によってキューに追加される Apex ジョブの最大数		50
許可される sendEmail メソッドの合計数		10
ヒープの合計サイズ ⁴	6 MB	12 MB
Salesforce サーバの最大 CPU 時間 ⁵	10,000 ミリ秒	60,000 ミリ秒
Apex トランザクションごとの最大実行時間		10 分
Apex トランザクションごとに許容される転送通知メソッドコールの最大数		10
各転送通知メソッドコールで送信できる転送通知の最大数		2,000

ミドルウェア機能

次の表は、このパターンに使用されるミドルウェアシステムの望ましいプロパティの一覧です。

プロパティ	必須	望ましい	不要
イベント処理		X	
プロトコル変換		X	
変換/加工		X	
キューイングとバッファリング		X	
同期伝送プロトコル	X		
非同期伝送プロトコル			X
仲介ルーティング		X	
プロセスコレオグラフィとサービスオーケストレーション		X	
トランザクション性(暗号化、署名、信頼できる配信、トランザクション管理)		X	
ルーティング			X
抽出、加工、読み込み (ETL)			X

例

ある公益事業会社は Salesforce を使用しており、顧客請求情報を格納する個別のシステムがあるとします。Salesforce に保存することなく顧客取引先の請求履歴データを表示したいと考えています。既存の Web サービスでは、特定の取引先の請求リストとその詳細を返すことはできますが、このデータをブラウザで表示することはできません。

この要件を満たすには、次のアプローチを使用します。

1. Salesforce で、Apex プロキシクラスから請求履歴サービスの WSDL を使用します。
2. Visualforce ページおよびカスタムコントローラを作成し、取引先番号を一意的識別子として Apex プロキシクラスを実行します。
3. その後、カスタムコントローラは、Apex コールアウトおよび Visualforce ページから返される値を解析し、請求をユーザに表示します。

この例は、次の内容を説明しています。

- Salesforce の取引先オブジェクトに保存された取引先番号を使用した、顧客の状態の追跡
- その後のコール元からの返信メッセージの処理

第3章 リモートプロセスの呼び出し—ファイアアンドフォーゲット

コンテキスト

Salesforceを使用して、リードの追跡、パイプラインの管理、商談の作成、リードを顧客に変換する注文詳細の取得を行っています。ただし、Salesforceは、注文を保持または処理するシステムではありません。Salesforceで注文の詳細を取得したら、リモートシステムで注文を作成して最後まで管理する必要があります。

このパターンを実装する場合、Salesforceは、注文を作成するためにリモートシステムをコールしますが、そのコールが正常に完了するまで待機しません。リモートシステムは、別のトランザクションで新しい注文番号および状況を使用して、必要に応じてSalesforceを更新できます。

問題

Salesforceでイベントが発生した場合に、リモートシステムからの応答を待機せずに、リモートシステムでプロセスを開始して必要な情報をそのプロセスに渡す方法は？

検討項目

このパターンに基づいてソリューションを適用する場合、次のような項目を検討する必要があります。

- リモートシステムへのコールでは、Salesforceは処理を続行する前に、応答があるまで待機する必要があるか？リモートシステムへのコールは同期要求-返信と非同期のどちらなのか？
- リモートシステムへのコールが同期である場合、Salesforceがコールと同じトランザクションの一部として応答を処理する必要があるか？
- メッセージサイズは小さいか？
- インテグレーションは、特定のイベント (Salesforce ユーザーインターフェースのボタンのクリックやDMLベースのイベントなど) の発生に基づいたものなのか？
- Salesforceからリモートシステムへのメッセージの配信保証は必要あるのか？
- リモートシステムは、Salesforceが契約を規定するコントラクトファーストインテグレーションに参加できるのか？ソリューションの一部のバリエーション (アウトバウンドメッセージなど) では、リモートシステムエンドポイントで実装する契約をSalesforceが規定します。
- カスタム Apex 開発よりも宣言的な設定手法の方が適していますか？この場合、Apex コールアウトよりもアウトバウンドメッセージなどのソリューションの方が適しています。

ソリューション

次の表は、このインテグレーションに関する問題へのソリューションの一覧です。

解決方法	適合度	コメント
ワークフロー駆動型アウトバウンドメッセージ	最適	<p>Salesforce では、カスタマイズすることなくアウトバウンドメッセージを実装できるため、挿入または更新イベントに基づいてリモートプロセスが呼び出されるこの種のインテグレーションには、このソリューションをお勧めします。Salesforce には、Salesforce での挿入または更新操作に基づいて SOAP メッセージをリモートシステムに送信できるワークフロー駆動型アウトバウンドメッセージ機能が用意されています。これらのメッセージは非同期に送信され、Salesforce ユーザインターフェースから独立しています。</p> <p>アウトバウンドメッセージは、特定のリモートエンドポイントに送信されます。リモートサービスは、Salesforce が契約を規定するコントラクトファーストインテグレーションに参加できる必要があります。</p> <p>リモートサービスがメッセージの受信時に肯定応答を返さない場合、Salesforce はメッセージの送信を再試行します。これにより、一種の配信保証が実現します。ミドルウェアを使用している場合、このソリューションは「ファーストマイル」の配信保証になります。</p>
アウトバウンドメッセージとコールバック	最適	<p>コールバックは、正しくない順序で送信されるメッセージの影響を緩和できます。また、次のようなシナリオに対応します。</p> <ul style="list-style-type: none">• 冪等性 — アウトバウンドメッセージは、肯定応答を適時に受信できない場合に再試行を行うため、ターゲットシステムに複数のメッセージが送信される場合があります。コールバックを使用すると、メッセージの送信時ではなく特定の時点でデータの受信が確認されます。• その他のデータの取得 — 1 通のアウトバウンドメッセージで送信できるのは 1 つのオブジェクトのデータのみです。コールバックを使用すると、他の関連レコード (親オブジェクトに関連付けられている関連リストなど) からデータを取得できます。

リモートプロセスの呼び出し — ファイアアンドフォー
 ゲット

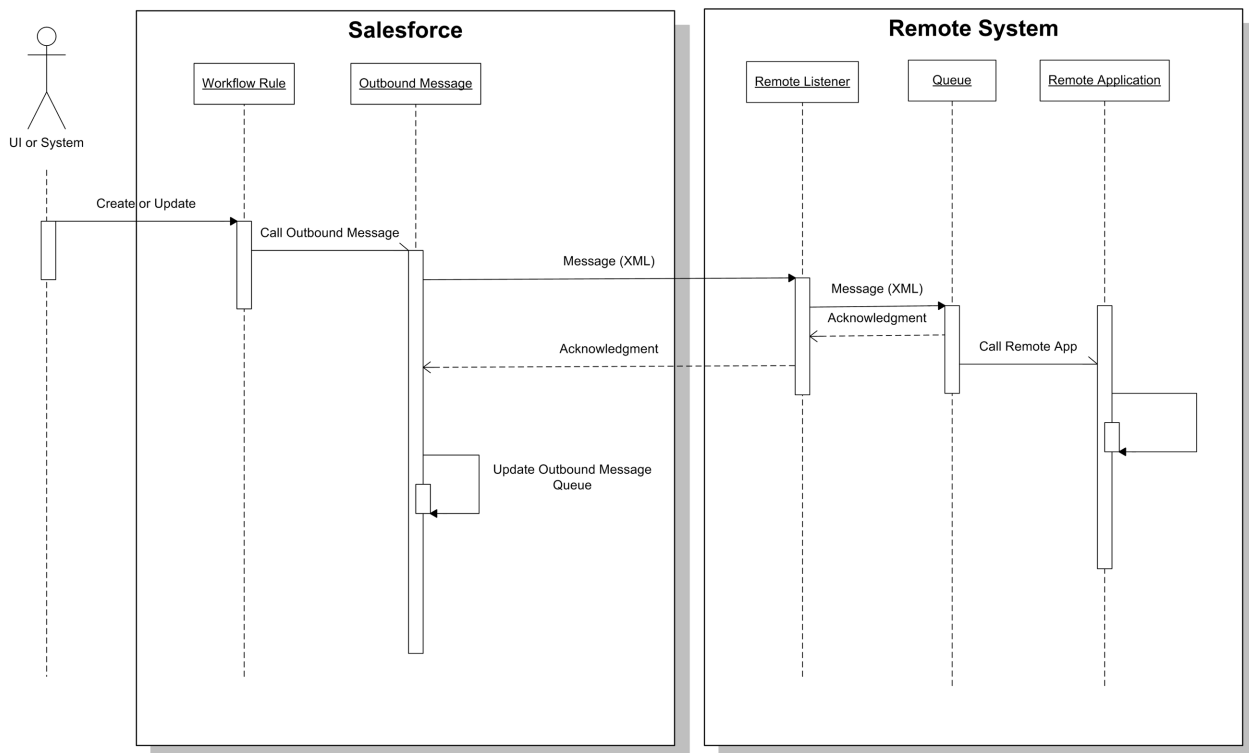
解決方法	適合度	コメント
		<p>アウトバウンドメッセージでは、認証トークンとして使用できる一意の SessionId が送信されます。認証トークンを使用して、SOAP API または REST API のいずれかでコールバックを認証および承認できます。コールバックを実行するシステムは Salesforce への認証を個別に行う必要はありません。その後、いずれかの API の標準メソッドを使用して、目的のビジネス機能を実行できます。</p> <p>通常、このバリエーションは、Salesforce からリモートシステムにアウトバウンドメッセージを送信してレコードを作成する場合に使用されます。コールバックは、リモートシステムで作成されたレコードの一意のキーで元の Salesforce レコードを更新します。</p>
<p>Apex SOAP または HTTP 非同期コールアウトを開始するカスタム Visualforce ページ</p>	<p>適切</p>	<p>通常、このソリューションはユーザインターフェーススペースのシナリオで使用されますが、カスタマイズする必要があります。また、このソリューションではメッセージの配信保証をコードで処理する必要があります。</p> <p>Apex コールアウトと共に Visualforce ページを使用することを規定する「リモートプロセスの呼び出し — 要求と返信」パターンのソリューションと似ています。このパターンでは、ユーザに制御を渡す前に Salesforce が要求が完了するまで待機しないという点が異なります。</p> <p>メッセージを受信すると、リモートシステムは応答してメッセージを受信したことを示し、メッセージを非同期に処理します。リモートシステムはメッセージを処理する前に Salesforce に制御を戻すため、Salesforce は処理が完了するまで待機する必要はありません。</p>
<p>Salesforce データの変更に基づいて呼び出され、Apex SOAP または HTTP 非同期コールアウトを実行するトリガ</p>	<p>準最適</p>	<p>Apex トリガを使用して、レコードデータの変更に基づく自動化を実行できます。</p> <p>Apex トリガを使用すれば、DML 操作の結果として Apex プロキシクラスを実行することができます。ただし、トリガコンテキスト内から作成されたすべてのコールは、非同期に実行される必要があります。</p>

リモートプロセスの呼び出し — ファイアアンドフォーゲット

解決方法	適合度	コメント
Apex SOAP または HTTP 非同期コールアウトを実行する Apex 一括処理ジョブ	準最適	リモートシステムへのコールは一括処理ジョブから実行できます。このソリューションにより、リモートバッチプロセスを実行したり、Salesforce でリモートシステムからの応答を処理したりできるようになります。ただし、特定のバッチコンテキストでは、コール数に制限があります。詳細は、『Salesforce の制限クイックリファレンスガイド』を参照してください。

概要図

次の図は、Salesforce からリモートシステムへのコールを示します。このコールは、レコードの作成操作または更新操作によってトリガされます。



このシナリオの内容は次のとおりです。

1. Salesforce の特定のレコードセットで DML による更新または挿入が行われます。
2. 特定の条件セットに基づいて、Salesforce ワークフロールールがトリガされます。
3. このワークフロールールは、SOAP ベースのメッセージをリモートリスナーに送信する事前設定済みのアウトバウンドメッセージを呼び出します。
4. リモートリスナーは SOAP メッセージを受信して、メッセージをローカルキューに配置し、同じトランザクションコンテキストの一部として Salesforce に肯定応答を返します。

リモートプロセスの呼び出し — ファイアアンドフォー ゲット

5. キューアプリケーションは、処理するためにリモートアプリケーションにメッセージを転送します。
6. Salesforce は、肯定応答を受信して要求を完了しますが、リモートアプリケーションがメッセージを処理するまで待機しません。
7. Salesforce は、リモートシステムからの肯定応答を最大で10秒待機します。10秒が経過すると、Salesforce は、最大で24時間アウトバウンドメッセージ要求の送信を再試行します。

リモートシステムがSalesforceに対して操作を実行する必要がある場合、追加で利用できるコールバック操作を実装できます。アウトバウンドメッセージでは、Salesforce への API または Web サービスコールを認証および承認するためにコールバックで使用できる SessionId が送信されます。

結果

このパターンに関連するソリューションの適用では、次の操作が可能です。

- ユーザーインターフェースから開始されたリモートプロセスの呼び出し (トランザクションの結果はエンドユーザに表示される場合がある)
- DML イベントから開始されたリモートプロセスの呼び出し (トランザクションの結果はコールプロセスで処理される場合がある)

コールメカニズム

コールメカニズムは、このパターンを実装するために選択されるソリューションに応じて異なります。

コールメカニズム	説明
Visualforce および Apex コントローラ	Apex コールアウトを使用してリモートプロセスを非同期に呼び出す場合に使用します。
ワークフロールール	アウトバウンドメッセージソリューションの場合にのみ使用します。DML イベント (作成および更新) によって Salesforce ワークフロールールがトリガされます。その後、そのルールにより、リモートシステムにメッセージを送信できます。
Apex トリガ	DML によって開始されたイベントからの Apex コールアウトを使用してリモートプロセスを呼び出す場合に、主に使用します。
Apex 一括処理クラス	バッチモードでリモートプロセスを呼び出す場合に使用します。

エラー処理と回復

エラー処理および回復戦略は、全体的なソリューションの一部として検討する必要があります。最適な方法は、選択するソリューションによって異なります。

ソリューション	エラー処理および回復戦略
Apex コールアウト	<ul style="list-style-type: none">• エラー処理 — リモートシステムがエンドプロセスの呼び出しを渡すため、コールアウトはリモートサービスの最初の呼び出しの例外のみを処理します。たとえば、リモートコールアウトから肯定応答を受信しない場合にタイ

リモートプロセスの呼び出し — ファイアアンドフォー ゲット

ソリューション	エラー処理および回復戦略
	<p>ムアウトイベントがトリガされます。非同期処理のために最初の呼び出しが渡されたら、リモートシステムが後続のエラーを処理する必要があります。</p> <ul style="list-style-type: none"> エラー回復 — このシナリオでは回復はより複雑です。サービス品質要件で定められている場合、カスタム再試行メカニズムを作成する必要があります。
アウトバウンドメッセージ	<ul style="list-style-type: none"> エラー処理 — このパターンは非同期であるため、リモートシステムがエラーを処理します。アウトバウンドメッセージの場合、タイムアウト期間内に肯定応答を受信できないと、Salesforce は最大で 24 時間再試行操作を行います。事実上、メッセージは「ファイアアンドフォーゲット」形式でリモートシステムに渡されるため、リモートサービスでエラー処理を実行する必要があります。 エラー回復 — このパターンは非同期であるため、サービスのサービス品質要件に基づいてシステムが再試行を開始する必要があります。アウトバウンドメッセージの場合、タイムアウト期間内にアウトバウンドリスナーから肯定応答を受信できないと、Salesforce は最大で 24 時間、再試行を行います。再試行間隔は時間と共に大幅に増えます。最初の間隔は 15 秒ですが、最後の間隔は 60 分になります。システム管理者は、24 時間の配信期間を超えているメッセージがないかどうかキューを監視し、必要に応じて手動で再試行する必要があります。 <p>カスタム Apex コールアウトの場合、サービス品質要件で必要とされていれば、カスタム再試行メカニズムを作成する必要があります。</p>

幂等設計に関する考慮事項

「リモートプロセスの呼び出し — 要求と返信」パターンの[幂等設計に関する考慮事項](#)は、このパターンにも適用されます。

アウトバウンドメッセージは非同期であり、肯定応答を受信できないと再試行が開始されるため、幂等性が重要になります。そのため、リモートサービスは幂等性が確保されるような形式で Salesforce からのメッセージを処理する必要があります。

アウトバウンドメッセージでは、メッセージごとに一意の ID が送信されます。この ID はどの再試行でも同じです。リモートシステムは、この一意の ID に基づいて重複するメッセージを追跡できます。更新されるレコードごとに一意のレコード ID も送信されます。この ID を使用すれば、重複するレコードの作成を回避できます。

セキュリティに関する考慮事項

リモートシステムへのコールでは、要求の機密性、整合性、および可用性を維持する必要があります。適用されるセキュリティに関する考慮事項は、選択するソリューションに応じて異なります。

リモートプロセスの呼び出し — ファイアアンドフォー ゲット

ソリューション	セキュリティに関する考慮事項
Apex コールアウト	<p>リモートシステムへのコールでは、要求の機密性、整合性、および可用性を維持する必要があります。次に、このパターンで Apex SOAP および HTTP コールを使用する場合に固有のセキュリティに関する考慮事項を示します。</p> <ul style="list-style-type: none">デフォルトでは一方向 SSL が有効になっていますが、自己署名証明書と CA 署名証明書で双方向 SSL もサポートされており、クライアントとサーバの両方の信頼性を確保できるようになっています。Salesforce では、Apex プロキシクラス生成時の WS-Security の使用はサポートされていません。必要に応じて、Apex Crypto クラスメソッドを使用する一方向ハッシュまたはデジタル署名で要求の整合性を確保することを検討してください。適切なファイアウォールメカニズムを実装してリモートシステムを保護する必要があります。
アウトバウンドメッセージ	<p>アウトバウンドメッセージでは、一方向 SSL がデフォルトで有効になっています。ただし、Salesforce アウトバウンドメッセージ証明書と共に双方向 SSL を使用することもできます。</p> <p>次に、その他のセキュリティに関する考慮事項を示します。</p> <ul style="list-style-type: none">リモートインテグレーションサーバのホワイトリストに Salesforce サーバの IP 範囲を登録します。適切なファイアウォールメカニズムを実装してリモートシステムを保護します。

「[セキュリティに関する考慮事項](#)」を参照してください。

補足

適時性

ファイアアンドフォーゲットパターンでは、適時性はそれほど重要ではありません。制御はすぐに (または、リモートシステムに正常に渡されたことを示す肯定応答の後に) クライアントに戻されます。Salesforce アウトバウンドメッセージでは、24 時間以内に肯定応答が返されないと、メッセージの有効期限が切れません。

データ量

データ量の考慮事項は、選択したソリューションによって異なります。各ソリューションの制限については、『[Salesforce の制限クイックリファレンスガイド](#)』を参照してください。

エンドポイント機能と標準のサポート

エンドポイントの機能と標準のサポートは、選択するソリューションによって異なります。

リモートプロセスの呼び出し — ファイアアンドフォー ゲット

ソリューション	エンドポイントに関する考慮事項
Apex SOAP コールアウト	<p>エンドポイントで HTTP を介して Web サービスコールを処理できる必要があります。Salesforce は、公開インターネットを介してエンドポイントにアクセスできる必要があります。</p> <p>このソリューションでは、リモートシステムに Salesforce でサポートされている規格との互換性があることが求められます。現時点で Salesforce でサポートされている Apex SOAP コールアウトの Web サービス規格は次のとおりです。</p> <ul style="list-style-type: none"> • WSDL 1.1 • SOAP 1.1 • WSI-Basic Profile 1.1 • HTTP
Apex HTTP コールアウト	<p>エンドポイントは、HTTP コールを受信でき、Salesforce が公開インターネットを介してアクセスできる必要があります。</p> <p>Apex HTTP コールアウトでは、標準の GET、POST、PUT、DELETE メソッドを使用する RESTful サービスをコールできます。</p>
アウトバウンドメッセージ	<ul style="list-style-type: none"> • エンドポイントは、Salesforce から送信される定義済み形式の SOAP メッセージを受信できるリスナーを実装できる必要があります。 • リモートリスナーは、契約が Salesforce によって提供されるコントラクトファースト実装に参加する必要があります。 • 各アウトバウンドメッセージには、定義済みの独自の WSDL があります。

状態管理

システムを統合する場合、継続的に状態を追跡するために一意のレコード識別子が重要になります。たとえば、レコードがリモートシステムで作成された場合、次の 2 つの方法があります。

- Salesforce が、リモートシステムのリモートレコードの主キーまたは一意のサロゲートキーを保存する。
- リモートシステムが、Salesforce の一意のレコード ID または他の一意のサロゲートキーを保存する。

次の表に、このパターンで状態を管理する場合の考慮事項を示します。

マスタシステム	説明
Salesforce	リモートシステムが Salesforce RecordId または Salesforce レコードのその他の一意のサロゲートキーを保存する必要があります。
リモートシステム	<p>Salesforce がリモートシステムの一意の識別子への参照を保存する必要があります。プロセスは非同期であるため、元のトランザクションの一部としてこの一意の識別子を保存することはできません。</p> <p>Salesforce は、リモートプロセスへのコールで一意の ID を提供する必要があります。その後、リモートシステムは Salesforce にコールバックし、Salesforce の一意</p>

リモートプロセスの呼び出し — ファイアアンドフォー ゲット

マスタシステム	説明
	<p>の ID を使用して、リモートシステムの一意的識別子で Salesforce のレコードを更新する必要があります。</p> <p>このコールバックは、処理の完了時にコールバックに使用される、該当のトランザクションの Salesforce の一意的識別子を保存する、あるいは、Salesforce の一意的識別子をリモートシステムのレコードに保存するためにリモートアプリケーションでいずれかの特定の状態を処理することを意味します。</p>

複雑なインテグレーションシナリオ

このパターンの場合、複雑なインテグレーションシナリオ(加工や処理のオーケストレーションなど)の考慮事項はソリューションごとに異なります。

ソリューション	考慮事項
Apex コールアウト	<p>場合によっては、このパターンで規定されるソリューションで複数の複雑なインテグレーションシナリオを実装する必要があります。ミドルウェアを使用するか、Salesforce から複合サービスをコールするのが最適な方法になります。これらのシナリオには、次の作業が含まれます。</p> <ul style="list-style-type: none">• ビジネスプロセスと複雑なフローロジックを伴うルールのオーケストレーション• 複数のシステムへのコールにおけるコールとその結果の集約• インバウンドメッセージとアウトバウンドメッセージの両方の加工• 複数のシステムへのコールにおけるトランザクションの整合性の確保
アウトバウンドメッセージ	<p>アウトバウンドメッセージの静的かつ宣言型の特性により、複雑なインテグレーションシナリオ(集約、オーケストレーション、加工など)は Salesforce で実行できません。このような操作は、リモートシステムまたはミドルウェアで処理する必要があります。</p>

ガバナ制限

Salesforce プラットフォームのマルチテナント性により、アウトバウンドコールアウトには制限があります。制限は、アウトバウンドコールの種類やコールのタイミングによって異なります。アウトバウンドメッセージに対しては、ガバナ制限はありません。『Salesforce の制限クイックリファレンスガイド』を参照してください。

信頼できるメッセージング

信頼できるメッセージングは、個々のコンポーネントが信頼できないリモートシステムへのメッセージ配信保証の問題を解決しようとしています。リモートシステムでメッセージの受信を保証する方法は、選択するソリューションによって異なります。

リモートプロセスの呼び出し — ファイアアンドフォー ゲット

ソリューション	信頼できるメッセージングに関する考慮事項
Apex コールアウト	<p>Salesforce では、信頼できるメッセージングプロトコル (WS-ReliableMessaging など) の明示的なサポートは提供していません。Salesforce メッセージを受信するリモートエンドポイントで信頼できるメッセージングシステム (JMS や MQ など) を実装することをお勧めします。このシステムにより、最終的にメッセージを処理するリモートシステムへの完全なエンドツーエンドの配信保証が実現します。ただし、このシステムにより、Salesforce からコール先のリモートエンドポイントへの配信が保証されるわけではありません。</p> <p>配信保証は、Salesforce をカスタマイズして処理する必要があります。カスタム再試行ロジックに加えてリモートエンドポイントからの肯定応答の処理などの特定の手法を実装する必要があります。</p>
アウトバウンドメッセージ	<p>アウトバウンドメッセージでは、信頼できるメッセージング形式が実現します。リモートシステムから肯定応答を受信できないと、プロセスが最大で 24 時間再試行を行います。このプロセスでは、リモートリスナーまでの配信しか保証されません。</p> <p>大部分の実装では、リモートリスナーが別のリモートサービスをコールします。信頼できるメッセージングシステムでこのリモートサービスの呼び出しを実行し、完全なエンドツーエンドの配信保証を実現することをお勧めします。Salesforce アウトバウンドメッセージへの肯定応答は、リモートリスナーが各自のメッセージをローカルキューに正常に配置した後に行われます。</p>

ミドルウェア機能

次の表は、このパターンに使用されるミドルウェアシステムの望ましいプロパティの一覧です。

プロパティ	必須	望ましい	不要
イベント処理		X	
プロトコル変換		X	
変換/加工		X	
キューイングとバッファリング	X		
同期伝送プロトコル			X
非同期伝送プロトコル	X		
仲介ルーティング		X	
プロセスコレオグラフィとサービスオーケストレーション		X	
トランザクション性(暗号化、署名、信頼できる配信、トランザクション管理)	X		

リモートプロセスの呼び出し — ファイアアンドフォー ゲット

プロパティ	必須	望ましい	不要
ルーティング			X
抽出、加工、読み込み (ETL)			X

ソリューションのバリエーション — アウトバウンドメッセージおよびメッセージの順序

1 通のメッセージが 24 時間にわたって再試行される可能性があるため、Salesforce アウトバウンドメッセージでは、メッセージの配信順序を保証することはできません。リモートシステムでメッセージの順序を処理する方法は複数あります。

- Salesforce はアウトバウンドメッセージのインスタンスごとに一意のメッセージ ID を送信します。リモートシステムは重複するメッセージ ID を持つメッセージを破棄します。
- Salesforce は RecordId のみを送信します。リモートシステムは Salesforce にコールバックし、要求を処理するために必要なデータを取得します。

ソリューションのバリエーション — アウトバウンドメッセージおよび削除

Salesforce ワークフロールールでは、レコードの削除は追跡できません。ルールでは、レコードの挿入または更新のみを追跡できます。そのため、レコードの削除に基づいてアウトバウンドメッセージを直接開始することはできません。次のプロセスを使用してメッセージを間接的に開始できます。

1. 削除されたレコードのキー情報を保存するカスタムオブジェクトを作成します。
2. ベースレコードが削除されたときに起動する Apex トリガを作成して、一意の識別子などの情報をカスタムオブジェクトに保存します。
3. カスタムオブジェクトレコードの作成に基づいてアウトバウンドメッセージを開始するワークフロールールを実装します。

状態の追跡が有効になっていることが重要です。状態の追跡を有効にするには、リモートシステムの一意の識別子を Salesforce に保存するか、Salesforce の一意の識別子をリモートシステムに保存します。

例

ある電気通信会社が、リードから商談へのプロセスで取引先を作成するためのフロントエンドとして Salesforce を使用したいと考えていたとします。商談が成立したら Salesforce で注文の作成が開始されますが、バックエンド ERP システムがデータマスタです。注文は Salesforce 商談レコードに保存され、注文が作成されたことを示すように商談状況が変更されます。

次の制約が適用されます。

- ERP システムは、サービスで Salesforce WSDL インターフェースを実装するコントラクトファーストインテグレーションに参加できる必要があります。
- Salesforce でのカスタム開発は行いません。
- 商談が注文に変換された直後に注文番号を通知する必要はありません。

これは、Salesforce アウトバウンドメッセージを使用した最適な実装例ですが、リモートシステムでプロキシサービスを実装する必要があります。

Salesforce で、次の処理を実行します。

リモートプロセスの呼び出し — ファイアアンドフォー ゲット

- (商談状況が「商談成立」に変更されたときなどに)アウトバウンドメッセージを開始するワークフロールールを作成します。
- 後続のコールバックで商談の RecordId および SessionId のみを送信するアウトバウンドメッセージを作成します。

リモートシステムでは、次の処理を実行します。

- Salesforce アウトバウンドメッセージの WSDL インターフェースを実装できるプロキシサービスを作成します。
- 商談が注文に変換されたことを示す 1 つ以上の通知をこのサービスで受信します。
- このサービスは、メッセージを変換してローカルメッセージキューに配置し、受信の通知で Salesforce アウトバウンドメッセージに肯定応答を返します。
- ローカルメッセージキューは、注文を作成できるようにメッセージをバックエンド ERP システムに転送します。
- 注文が作成されたら、個別のスレッドで SessionId を認証トークンとして使用して Salesforce にコールバックします。コールバックによって、注文番号と状況で商談が更新されます。このコールバックは、ドキュメントで説明されているパターンのソリューション (Salesforce SOAP API、REST API、Apex Web サービスなど) を使用して実行できます。

この例は、次の内容を説明しています。

- 非同期に呼び出されるリモートプロセスの実装
- エンドツーエンドの配信保証
- レコードの状態を更新するための Salesforce への後続のコールバック

第4章 バッチデータの同期

コンテキスト

CRM 実装を Salesforce に移動して、次の作業を実行しようとしています。

- 現在の CRM システムから取引先、取引先責任者、および商談を抽出して加工し、データを Salesforce に読み込む (初期データインポート)。
- リモートシステムから週次で (継続的に) 顧客請求データを抽出して加工し、Salesforce に読み込む。
- 週次で (継続的に) Salesforce から顧客活動情報を抽出し、オンプレミス型のデータウェアハウスにインポートする。

問題

これらのインポートとエクスポートにより営業時間中のエンドユーザ操作に支障が出る可能性があり、また大量のデータを扱うことを考慮した場合、Salesforce へのデータのインポートと Salesforce からのデータのエクスポートをどう行えばよいか?

検討項目

このパターンに基づいてソリューションを適用する場合、次のようなさまざまな項目を検討する必要があります。

- データを Salesforce に保存する必要があるか? 必要がない場合、アーキテクトが検討可能で検討すべきインテグレーションオプションが他にあります (たとえば、マッシュアップなど)。
- データを Salesforce に保存する必要がある場合、データはリモートシステムのイベントに応じて更新する必要がありますか?
- データは定期的に更新する必要がありますか?
- データは主なビジネスプロセスをサポートしているか?
- Salesforce でこのデータが使用できるかどうかによって影響を受ける分析 (レポート) 要件があるか?

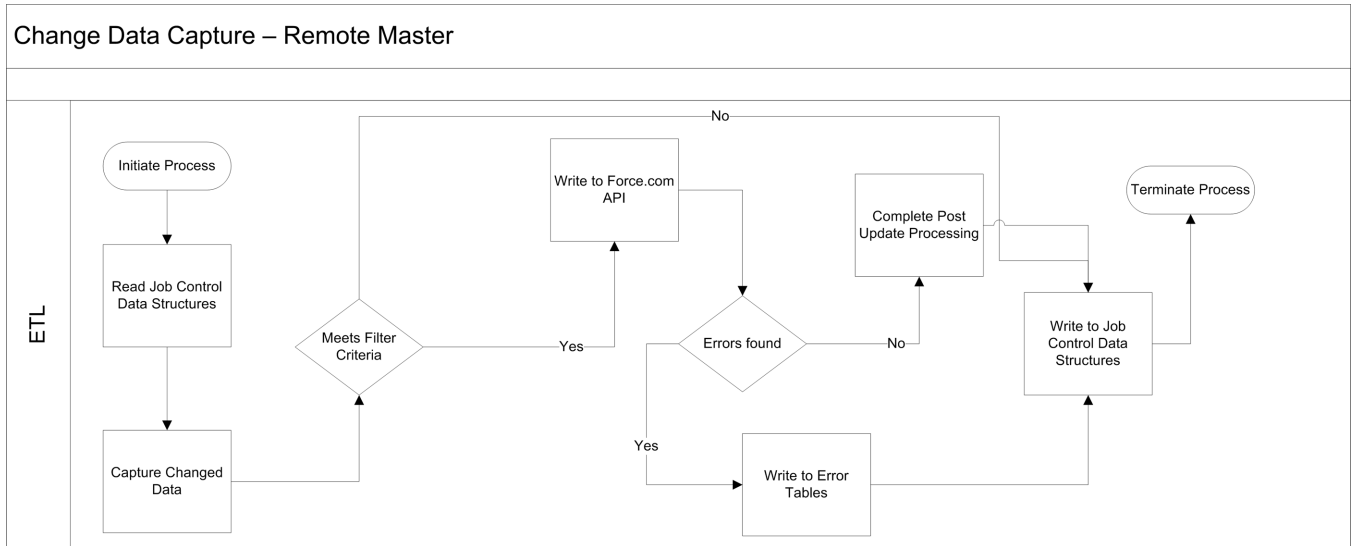
ソリューション

次の表は、このインテグレーションに関する問題へのさまざまなソリューションの一覧です。

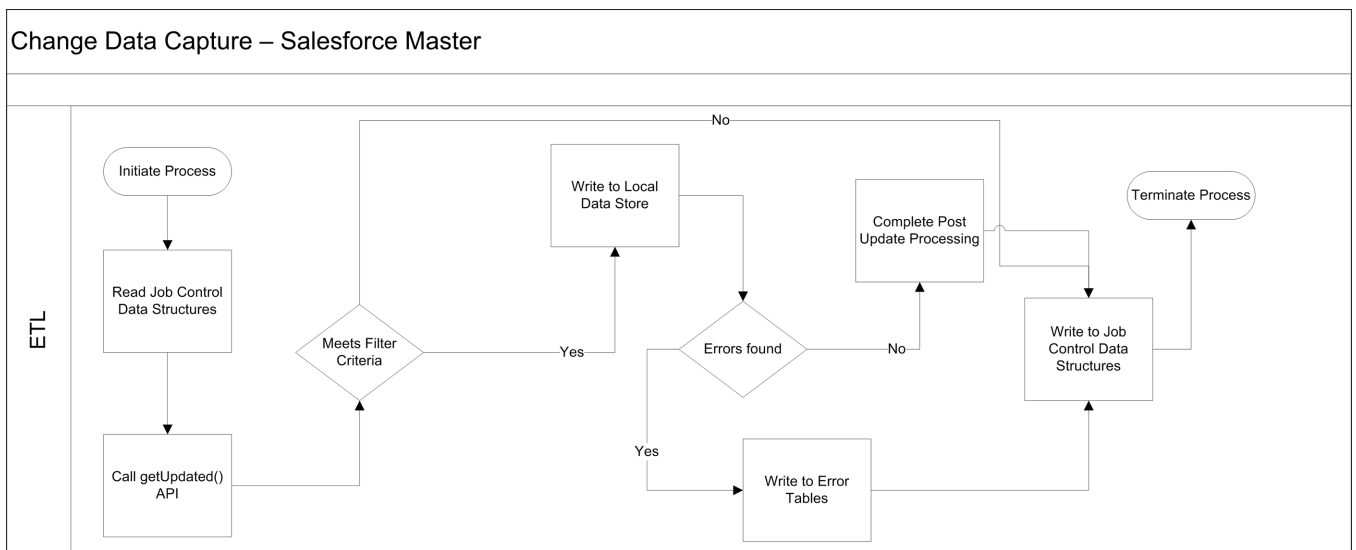
ソリューション	適合度	データマスタ	コメント
変更データキャプチャ	最適	リモートシステム	<p>サードパーティの ETL ツールを利用して、ソースデータに対する変更データキャプチャを実行できます。</p> <p>ツールはソースデータセットの変更に反応し、データを加工してから、Salesforce Bulk API をコールして DML ステートメントを発行します。これは、Salesforce SOAP API を使用して実装することもできます。</p>
変更データキャプチャ	最適	Salesforce	<p>サードパーティの ETL ツールを利用して、ERP および Salesforce データセットに対する変更データキャプチャを実行できます。</p> <p>このソリューションでは Salesforce がデータソースであるため、各行の時間/状況情報を使用してデータをクエリし、ターゲット結果セットを絞り込むことができます。これは、SOQL と共に SOAP API と <code>query()</code> メソッドを使用するか、SOAP API と <code>getUpdated()</code> メソッドを使用することで実装できます。</p>
リモートコールイン	準最適	リモートシステム	<p>データ更新が発生したら、リモートシステムはいずれかの API を使用して Salesforce にコールインし、データ更新を実行できます。ただし、これによって2つのシステム間に大量のトラフィックが継続的に発生します。</p> <p>このパターンでは更新が絶え間なく発生する可能性があり、エンドユーザのパフォーマンスに影響を与えることが予想されるため、エラー処理とロックにも十分に注意する必要があります。</p>
リモートプロセス呼び出し	準最適	Salesforce	<p>データ更新が発生したら、Salesforce はリモートシステムにコールインしてデータ更新を実行できます。ただし、これによって2つのシステム間に大量のトラフィックが継続的に発生します。</p> <p>このパターンでは更新が絶え間なく発生する可能性があり、エンドユーザのパフォーマンスに影響を与えることが予想されるため、エラー処理とロックにも十分に注意する必要があります。</p>

概要図

次の図は、このパターンでリモートシステムがデータマスタとなる場合のイベントの順序を示します。



次の図は、このパターンで Salesforce がデータマスタとなる場合のイベントの順序を示します。



結果

次のシナリオでは、外部から取得したデータを Salesforce と統合できます。


- 外部システムがデータマスタ — Salesforce が1つのソースシステムまたは複数のシステムによって提供されるデータのコンシューマになります。このシナリオでは、通常、Salesforceにデータをインポートする前に、データウェアハウスまたはデータマートでデータを集約します。
- Salesforce がデータマスタ — Salesforce が特定のエンティティのレコードシステムになります。

典型的な Salesforce インテグレーションシナリオでは、実装チームは次のいずれかの作業を行います。

- ソースデータセットに対する変更データキャプチャを実装します。
- 制御テーブルと呼ばれる、サポートデータベース構造のセットを、中間のオンプレミス型データベースに実装します。

次に ETL ツールを使用して、次の作業を行うプログラムを作成します。

1. 制御テーブルを読み取ってジョブの最終実行時間を判別し、必要な他の制御値を抽出します。
2. 上記の制御値を検索条件として使用してソースデータセットをクエリします。
3. 検証、改良など、事前定義された処理ルールを適用します。
4. ETL ツールで使用可能なコネクタ/加工機能を使用して、ターゲットデータセットを作成します。
5. データセットを Salesforce オブジェクトに書き込みます。
6. 処理が成功したら、制御テーブルの制御値を更新します。
7. 処理が失敗したら、制御テーブルを再起動可能にする値で更新し、終了します。

 **メモ:** 制御テーブルと関連するデータ構造は、ETL ツールがアクセスできる環境に作成することをお勧めします。その環境で Salesforce へのアクセスができない場合でも同様です。これにより、適度な回復力が提供されます。このプロセスでは、Salesforce をスポークとして扱う必要があり、ETL インフラストラクチャがハブになります。

ETL ツールがデータ同期機能から最大の効果を得られるように、次の点を考慮してください。

- ETL ジョブをチェーンにして順に並べ、まとまった1つのプロセスにする。
- 入力データの照合には両方のシステムの主キーを使用する。
- 特定の API メソッドを使用して更新されたデータのみを抽出する。
- 主従関係または参照関係の子レコードをインポートする場合、ソースでの親キーを使用してインポートするデータをグループ化し、ロックを回避する。たとえば、取引先責任者データをインポートする場合、取引先責任者データを親の取引先キーでグループ化し、1回の API コールで1つの取引先に対して最大数の取引先責任者を読み込めるようにします。インポートするデータがグループ化されていないと、通常、最初の取引先責任者レコードが読み込まれ、API コールのコンテキスト内ではその取引先の後続の取引先責任者レコードは読み込みに失敗します。
- トリガなど、インポート後処理は、必ずデータを選択的に処理する。
- シナリオで大量のデータを扱う場合、ホワイトペーパー『[大量のデータを使用するリリースのベストプラクティス](#)』を参照してください。

エラー処理と回復

エラー処理および回復戦略は、全体的なソリューションの一部として検討する必要があります。最適な方法は、選択するソリューションによって異なります。

エラー表示場所	エラー処理および回復戦略
Salesforce からの読み取り	<ul style="list-style-type: none"> • エラー処理 — 読み取り操作中にエラーが発生した場合、インフラストラクチャ関連ではないエラーについては再試行を実装します。失敗が繰り返される場合は、制御テーブル/エラーテーブルを使用した標準処理を ETL 操作のコンテキストに実装して次の処理を行う必要があります。 <ul style="list-style-type: none"> - エラーをログに記録する - 読み取り操作を再試行する - 失敗した場合は終了する - 通知を送信する

エラー表示場所	エラー処理および回復戦略
	<ul style="list-style-type: none"> エラー回復—失敗した読み取り操作から回復するには、ETL プロセスを再起動します。 <p>操作は成功してもレコード単位で失敗する場合は、ジョブの即時再起動か後続処理の実行で問題に対処します。この場合、遅延再起動は、エラーの原因と思われるデータに優先順位を付けて修正する時間を取れるため、適したソリューションとなる可能性があります。</p>
Salesforce への書き込み	<ul style="list-style-type: none"> エラー処理—書き込み操作中のエラーは、アプリケーション内の複数の要因が組み合わされて発生することがあります。API コールは、以下の情報で構成される結果セットを返します。この情報を書き込み操作の再試行に使用する必要があります (必要な場合)。 <ul style="list-style-type: none"> レコード識別情報 成功/失敗の通知 各レコードのエラーのまとめ エラー回復—失敗した読み取り操作から回復するには、ETL プロセスを再起動します。 <p>操作は成功してもレコード単位で失敗する場合は、ジョブの即時再起動か後続処理の実行で問題に対処します。この場合、遅延再起動は、エラーの原因と思われるデータに優先順位を付けて修正する時間を取れるため、適したソリューションとなる可能性があります。</p>
外部マスタシステム	<p>エラーは、マスタシステムのベストプラクティスに従って処理する必要があります。</p>

セキュリティに関する考慮事項

リモートシステムへのコールでは、要求の機密性、整合性、および可用性を維持する必要があります。適用されるセキュリティに関する考慮事項は、選択するソリューションに応じて異なります。

- Salesforce API への認証された API アクセスを可能にするには、Lightning Platform ライセンスが必要です。
- パスワードアクセスのセキュリティを確保するために、標準の暗号化を使用することをお勧めします。
- Salesforce API へのコールを行うときには、HTTPS プロトコルを使用します。必要に応じて、オンプレミス型のセキュリティソリューションで Salesforce API へのトラフィックにプロキシを適用することもできます。

「[セキュリティに関する考慮事項](#)」を参照してください。

補足

適時性

このパターンでは、適時性の重要性はそれほど大きくはありません。ただし、インターフェースの設計では、すべてのバッチプロセスが指定されたバッチ時間帯内に完了するように注意する必要があります。

どのバッチ指向操作にも言えることですが、バッチ処理時間帯の間は、ソースシステムとターゲットシステムを隔離することをお勧めします。営業時間内にバッチを読み込むと、何らかの競合が発生し、ユーザの更新が失敗するか、より深刻な事態としてバッチ読み込み(あるいは部分的なバッチ読み込み)が失敗することがあります。

グローバルな事業展開をしている組織の場合、システムは常時使用中であるため、すべてのバッチプロセスを同時には実行できない場合があります。こうした場合、レコードタイプやその他の検索条件を使用したデータセグメント化手法により、データの競合を回避できることがあります。

状態管理

状態管理を実装するには、2つのシステム間でサロゲートキーを使用します。Salesforce のエンティティ全体で何らかのトランザクション管理が必要な場合、Apexを使用する [リモートコールイン](#) パターンを使用することをお勧めします。

プラットフォームで標準の緩やかなレコードロックが行われる場合、APIを使用した更新が発生すると、レコードを編集中のユーザにはレコードを再取得して処理を最初からやり直すことが要求されます。Salesforce API のコンテキストでは、緩やかなロックとは、次のようなプロセスを指します。

- Salesforce は、特定のユーザが編集中のレコードの状態を維持しない。
- 読み取り時、データが抽出された時間を記録する。
- ユーザがレコードを更新して保存すると、Salesforce は、その間に他のユーザがそのレコードを更新していないかチェックする。
- レコードが更新されていた場合、システムはユーザに、更新が行われたためレコードの最新バージョンを取得してから更新を続行するように通知する。

ミドルウェア機能

このパターンの実装に使用する最も効果的な外部技術は、従来の ETL ツールです。重要なのは、選択するミドルウェアツールが、Salesforce Bulk API をサポートしていることです。

ミドルウェアツールが `getUpdated()` 関数をサポートしていることは、役には立ちますが、不可欠ではありません。この関数は、Salesforce プラットフォームで標準の変更データキャプチャ機能にきわめて近い実装を提供します。

次の表は、このパターンに使用されるミドルウェアシステムの望ましいプロパティの一覧です。

プロパティ	必須	望ましい	不要
イベント処理		X	
プロトコル変換		X	
変換/加工	X	X	
キューイングとバッファリング	X		
同期伝送プロトコル			X
非同期伝送プロトコル	X		
仲介ルーティング		X	

プロパティ	必須	望ましい	不要
プロセスコレオグラフィとサービスオーケストレーション	X		
トランザクション性(暗号化、署名、信頼できる配信、トランザクション管理)		X	
ルーティング		X	
抽出、加工、読み込み (ETL)	X		

例

ある公益企業では、メインフレームベースのバッチプロセスを使用して、見込み客を個々の営業担当者とチームに割り当てています。この情報は、夜間処理で Salesforce にインポートする必要があります。

この顧客は、市販の ETL ツールを使用してソーステーブルの変更データキャプチャを実装することを決定しました。

ソリューションは次のように機能します。

- cron に似たスケジューラが、一括処理ジョブを実行して見込み客をユーザとチームに割り当てます。
- 一括処理ジョブが実行されてデータが更新された後、ETL ツールが変更データキャプチャを使用してこれらの変更を認識します。ETL ツールが、データストアからの変更を順に並べます。
- ETL コネクタが Salesforce SOAP API を使用して変更を Salesforce に読み込みます。

第 5 章 リモートコールイン

コンテキスト

Salesforce を使用して、リードの追跡、パイプラインの管理、商談の作成、リードを顧客に変換する注文詳細の取得を行っています。ただし、Salesforce システムでは注文の保存や処理は行っていません。注文は外部(リモート)システムによって管理されています。リモートシステムは、注文が処理フェーズで渡されると、Salesforce の注文状況を更新する必要があります。

問題

リモートシステムはどのように Salesforce に接続して認証され、既存のレコードを更新するのか?

検討項目

このパターンに基づいてソリューションを適用する場合、次のようなさまざまな項目を検討する必要があります。

- Salesforce へのコールでは、リモートプロセスが応答を待機してから処理を続行する必要があるか? リモートプロセスは、非同期コールのシミュレーションに必要なければ応答を破棄できますが、Salesforce へのリモートコールは常に同期された要求-返信です。
- メッセージ形式はどれか (たとえば、SOAP、REST、SOAP over HTTP、REST over HTTP など)?
- メッセージサイズは比較的小さいか、大きいのか?
- SOAP 対応のリモートシステムの場合、リモートシステムは、Salesforce が契約を指示するコントラクトファースト手法で参加できるか? これは、事前定義された WSDL が提供される、SOAP API を使用する場合は必須です。
- トランザクション処理は必須か?
- Salesforce アプリケーションのカスタマイズ耐性はどの程度か?

ソリューション

次の表は、このインテグレーションに関する問題へのさまざまなソリューションの一覧です。

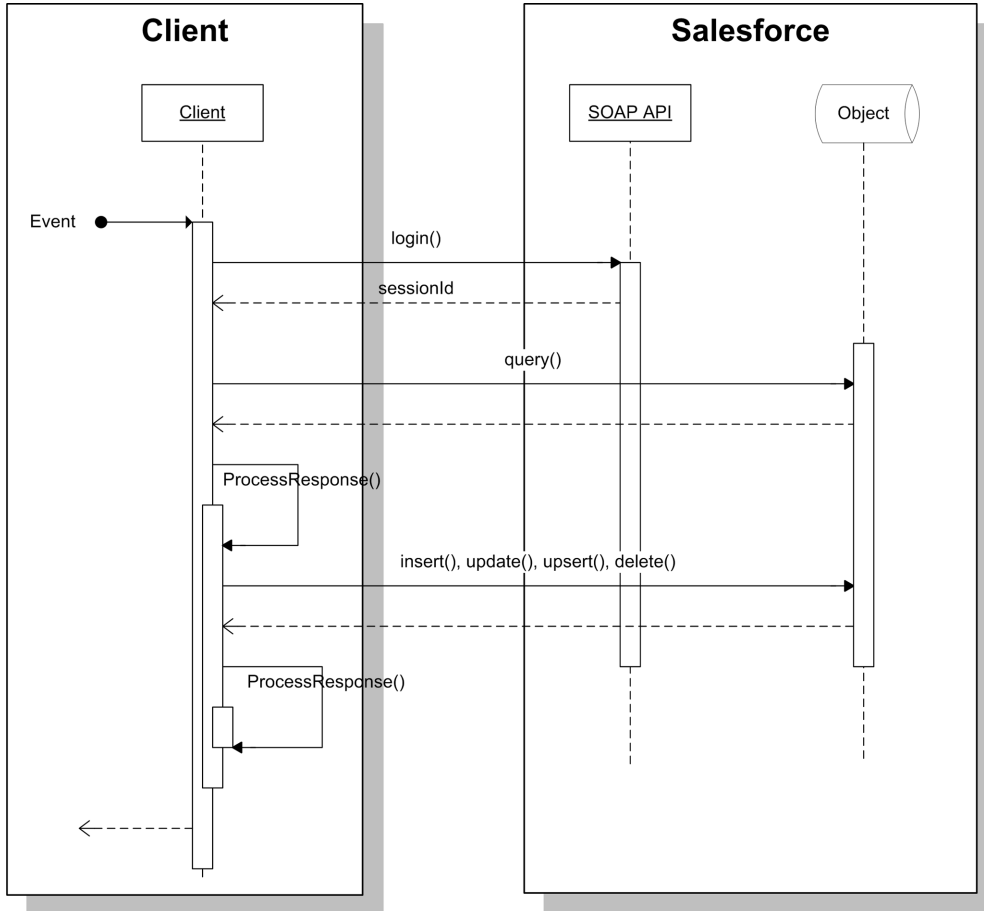
ソリューション	適合度	コメント
SOAP API	最適	<ul style="list-style-type: none"> ● アクセス性— Salesforce が提供する SOAP API を使用してリモートシステムは以下の操作を実行できます。 <ul style="list-style-type: none"> - 組織内のデータのクエリ - データの追加、更新、削除 - 組織のメタデータの取得 - 管理タスク実行のためのユーティリティの実行 ● 同期API— API コールが行われた後、リモートクライアントアプリケーションはサービスから応答を受信するまで待機します。Salesforce への非同期コールはサポートされていません。 ● 生成された WSDL — Salesforce では、リモートシステムに 2 つの WSDL を使用できます。 <ul style="list-style-type: none"> - Enterprise WSDL — Salesforce 組織に固有の強く型付けされた WSDL を使用できます。 - Partner WSDL — Salesforce 組織に固有ではない、緩やかに型付けされた WSDL を使用できます。 ● セキュリティ— SOAP API を実行するクライアントは、有効なログイン情報を持ち、API コールを実行するためのセッションを取得する必要があります。API は、ログインユーザのプロファイルに基づいてアプリケーションに設定されたオブジェクトレベルおよび項目レベルのセキュリティを反映します。 ● トランザクション/コミット動作— デフォルトでは、各 API コールは、一部のレコードがエラーとマークされた場合に部分的な成功を許容します。これを「オールオアナッシング」動作に変更して、エラーが発生したらすべての結果をロールバックすることができます。複数の API コールにまたがってトランザクションを実行することはできません。この制限を打開するために、1 つの API で複数のオブジェクトに影響を与えることができます。 ● 大量データ— 大量データ (500,000 件を超えるレコード) 操作の場合、REST ベースの Bulk API を使用します。

ソリューション	適合度	コメント
REST API	最適	<ul style="list-style-type: none"> • アクセス性— Salesforce が提供する REST API を使用してリモートシステムは以下の操作を実行できます。 <ul style="list-style-type: none"> - 組織内のデータのクエリ - データの追加、更新、削除 - 組織のメタデータの取得 - 管理タスク実行のためのユーティリティの実行 • 同期API— API コールが行われた後、リモートクライアントアプリケーションはサービスから応答を受信するまで待機します。Salesforce への非同期コールはサポートされていません。 • REST API と SOAP API の比較— REST はリソース (エンティティ/オブジェクト) を URI として公開し、HTTP 動詞を使用してこれらのリソースに対する CRUD 操作を定義します。SOAP とは異なり、REST API では、事前定義した契約は不要で、応答には XML と JSON を利用し、緩やかな型付けを使用します。REST API は軽量であり、Salesforce との相互作用のための単純な方法を提供します。インテグレーションおよび開発が容易という利点があり、モバイルアプリケーションおよび Web 2.0 プロジェクトで使用するためには最適な選択です。 • セキュリティ— REST API を実行するクライアントは、有効なログイン情報を持ち、API コールを実行するためのセッションを取得する必要があります。API は、ログインユーザのプロファイルに基づいてアプリケーションに設定されたオブジェクトレベルおよび項目レベルのセキュリティを反映します。 • トランザクション/コミット動作— デフォルトでは、各レコードは、個別のトランザクションとして扱われ、個別にコミットされます。1つのレコード変更が失敗しても、他のレコード変更はロールバックされません。これを「オールオアナッシング」動作に変更できます。 • 大量データ— 大量データ (500,000 件を超えるレコード) 操作の場合、REST ベースの Bulk API を使用します。

ソリューション	適合度	コメント
Apex Web サービス	準最適	<p>Apex クラスメソッドは、外部のアプリケーションに Web サービスメソッドとして公開できます。これは SOAP API に代わる選択肢であり、通常は、次の追加要件を満たす必要がある場合に使用されます。</p> <ul style="list-style-type: none"> • 完全なトランザクションサポートが必要である (たとえば、取引先、取引先責任者、および商談をすべて1つのトランザクションで作成する場合など)。 • コミットする前に、カスタムロジックを Salesforce 側で適用する必要がある。 <p>Apex Web サービスを使用する利点と、Salesforce でメンテナンスが必要な追加コードを比較して評価する必要があります。</p>
Apex REST サービス	準最適	<p>Apex クラスは、それに対して定義された HTTP 動詞 (POST や GET など) を使用して特定の URI に対応付けられた REST リソースとして公開できます。</p> <p>SOAP とは異なり、クライアントがサービス定義/契約 (WSDL) をコンシュームしてクライアントスタブを生成する必要はありません。リモートシステムには、HTTP 要求を作成して、返された結果 (XML または JSON) を処理する機能のみが必要です。</p>
Bulk API	バッチに最適	<p>Bulk API は、REST 規則に基づいており、大規模データセットの読み込みまたは削除用に最適化されています。アクセス性とセキュリティ動作は、REST API と同じです。</p> <p>Bulk API では、クライアントアプリケーションは、複数のバッチを送信し、そのバッチが Salesforce によりバックグラウンドで処理されることで、大量のレコードをクエリ、挿入、更新、更新/挿入、削除できます。対照的に、SOAP API は、一度に少数のレコードを更新するリアルタイムのクライアントアプリケーション用に最適化されています。</p> <p>SOAP API を使用しても多数のレコードを処理することはできますが、数百から、数千、数百万のレコードがデータセットに含まれている場合には実用性に欠けます。これは、比較的オーバーヘッドが高く、パフォーマンスが低いという特性のためです。</p>

概要図

次の図は、SOAP API を使用してこのパターンを実装する場合のイベントの順序を示します。イベントの順序は、REST API を使用する場合と同じです。



結果

このパターンに関連するソリューションの適用では、次の操作が可能です。

- リモートシステムがSalesforce APIをコールして、データベースをクエリし、単独のオブジェクト操作(作成、更新、削除など)を実行する。
- リモートシステムが、複数オブジェクトのトランザクション操作とカスタムの処理前/後ロジックをサポートできるように作成されたカスタム Salesforce API (サービス) をコールする。

コールメカニズム

コールメカニズムは、このパターンを実装するために選択されるソリューションに応じて異なります。

リモートコールイン

コールメカニズム	説明
SOAP API	リモートシステムは Salesforce Enterprise または Partner WSDL を使用してクライアントスタブを生成します。このスタブは標準SOAP APIの呼び出しに使用されます。
REST API	リモートシステムは、Apex REST サービスにアクセスする前に認証を行う必要があります。リモートシステムでは、OAuth 2.0 またはユーザ名/パスワード認証を使用できます。いずれの場合も、クライアントは認証 HTTP ヘッダーと適切な値 (SOAP API へのログインコール経由で OAuth アクセストークンまたはセッション ID を取得可能) を設定する必要があります。 次に、リモートシステムは、適切な動詞で REST 呼び出し (HTTP 要求) を生成し、返された結果 (JSON および XML データ形式がサポート対象) を処理します。
Apex Web サービス	リモートシステムはカスタム Apex Web サービス WSDL をコンシュームしてクライアントスタブを生成します。このスタブはカスタム Apex Web サービスの呼び出しに使用されます。
Apex REST サービス	REST API のように、リソース URI と該当する動詞は、@RestResource、@HttpGet、および @HttpPost アノテーションを使用して定義されます。
Bulk API	Bulk API は、REST ベースの API であるため、コールメカニズムは REST API と同じです。

エラー処理と回復

エラー処理および回復戦略は、全体的なソリューションの一部として検討する必要があります。

- **エラー処理**— すべてのリモートコールインメソッド、標準またはカスタムの API では、リモートシステムがタイムアウトや再試行の管理など後続で発生するエラーを処理する必要があります。ミドルウェアを使用して、エラー処理と回復のロジックを提供できます。
- **エラー回復**— サービス品質要件で定められている場合は、カスタムの再試行メカニズムを作成する必要があります。この場合、設計特性を幕等にすることが重要です。

幕等設計に関する考慮事項

幕等機能によって、呼び出しを繰り返しても安全で、悪影響がないことが保証されます。幕等性が実装されていない場合、同じメッセージを繰り返し呼び出すと異なる結果になり、レコードの重複作成、トランザクションの重複処理など、データ整合性に問題が生じる可能性があります。

リモートシステムは複数回の(重複した)コールを管理して、エラーやタイムアウトの場合は、重複挿入や冗長な更新などを回避する必要があります(特に下流のトリガやワークフロールールが起動する場合)。Salesforce 内でこうした状況を管理することは可能ですが(特にカスタム SOAP および REST サービスの場合)、リモートシステム(またはミドルウェア)でエラー処理と幕等設計を管理することをお勧めします。

セキュリティに関する考慮事項

適用されるセキュリティに関する考慮事項は、選択するパターンソリューションに応じて異なります。すべての場合で、プラットフォームはログインユーザのアクセス権(プロファイル設定、共有ルール、権限セットなど)を使用します。さらに、プロファイル IP 制限を使用して、API へのアクセスを特定の IP アドレス範囲に制限できます。

ソリューション	セキュリティに関する考慮事項
SOAP API	<p>Salesforce は、Secure Sockets Layer v3 (SSL) および Transport Layer Security (TLS) プロトコルをサポートします。暗号鍵の長さは、128ビット以上でなければなりません。</p> <p>リモートシステムがセッションIDを取得するには、有効なログイン情報を使用してログインする必要があります。リモートシステムにすでに有効なセッションIDがある場合は、明示的なログインなしでAPIをコールできます。これは、このドキュメントですでに説明したコールバックパターンで使用されます。このパターンでは、先行する Salesforce アウトバウンドメッセージに含まれているセッションIDとレコードIDが後続の作業で使用されます。</p> <p>クライアントは、コールごとに新しいセッションIDを取得するのではなく、SOAP API キャッシュをコールし、セッションIDを再利用してパフォーマンスを最大化することをお勧めします。</p>
REST API	<p>リモートシステムで認証用に OAuth の信頼を確立することをお勧めします。それによって、HTTP 動詞を使用して特定のリソースに対し REST コールを実行できます。また、他の手段で(たとえば、SOAP API をコールして取得、送信メッセージ経由で提供など)取得された有効なセッションIDで REST コールを行うことも可能です。</p> <p>クライアントは、コールごとに新しいセッションIDを取得するのではなく、REST API キャッシュをコールし、セッションIDを再利用してパフォーマンスを最大化することをお勧めします。</p>
Apex Web サービス	セキュリティに関する考慮事項は、SOAP API と同じです。
Apex REST サービス	セキュリティに関する考慮事項は、REST API と同じです。
Bulk API	セキュリティに関する考慮事項は、REST API と同じです。

「[セキュリティに関する考慮事項](#)」を参照してください。

補足

適時性

SOAP API および Apex Web サービス API は同期します。次のタイムアウトが適用されます。

- セッションタイムアウト — Salesforce 組織のセッションタイムアウト設定に基づいて、アクティビティがない場合、セッションはタイムアウトします。
- クエリタイムアウト — 各 SOQL クエリには、120 秒という個別のタイムアウト制限があります。

データ量

データ量に関する考慮事項は、選択するソリューションと通信の種類に応じて異なります。

リモートコールイン

ソリューション	通信の種類	制限
SOAP API または REST API	同期	<ul style="list-style-type: none"> ログイン—ログイン要求のサイズは、10 KB 未満に制限されています。 作成、更新、削除—リモートシステムが一度に作成、更新、または削除できるレコード数は最大 200 件です。複数回のコールを行えば合計 200 件を超えるレコードを処理できますが、要求ごとのサイズは 200 レコードに制限されています。 クエリ結果サイズ—デフォルトでは、<code>query()</code> または <code>queryMore()</code> コールで返される、クエリ結果オブジェクト内に返される行数 (バッチサイズ) は 500 に設定されています。返される行数がバッチサイズを上回る場合、<code>queryMore()</code> API コールを使用して複数のバッチを反復処理します。最大バッチサイズは 2,000 レコードです。ただし、追加のルールが適用される場合もあるため、詳細は、『Salesforce の制限クイックリファレンスガイド』を参照してください。
Bulk API	同期	<p>Bulk API は、大規模なデータセットの非同期のインポートまたはエクスポート用に最適化されています。</p> <p>Bulk API は、バッチ要求と関連データを送信するときは同期します。データの実際の処理は、バックグラウンドで非同期に行われます。API とバッチ処理制限の詳細は、『Bulk API の制限』を参照してください。</p> <ul style="list-style-type: none"> 24 時間あたりに送信可能なバッチ数は最大で 2,000 件に制限されています。 1 つのバッチには、最大で 10,000 件のレコードを含めることができます。

エンドポイント機能と標準のサポート

エンドポイントの機能と標準のサポートは、選択するソリューションによって異なります。

ソリューション	エンドポイントに関する考慮事項
SOAP API	<p>リモートシステムは、Salesforce で事前に定義されたメッセージ形式に基づいて Salesforce SOAP API をコールするクライアントを実装できなければなりません。</p> <p>リモートシステム (クライアント) は、契約が Salesforce によって提供されるコントラクトファースト実装に参加する必要があります (Enterprise または Partner WSDL など)。</p>
REST API	<p>リモートシステムは、Salesforce で定義された REST サービスを呼び出し、XML または JSON の結果を処理する REST クライアントを実装できなければなりません。</p>

リモートコールイン

ソリューション	エンドポイントに関する考慮事項
Apex Web サービス	<p>リモートシステムは、Salesforce で定義された、事前定義形式の SOAP メッセージを呼び出すクライアントを実装できなければなりません。</p> <p>リモートシステムは、Apex Web サービスが実装された後に契約が Salesforce によって提供される、コードファースト実装に参加する必要があります。各 Apex Web サービスには独自の WSDL があります。</p>
Apex REST サービス	エンドポイントに関する考慮事項は、REST API と同じです。

状態管理

システムを統合するとき、継続的な状態追跡にキーは重要です。たとえば、レコードがリモートシステムで作成され、そのレコードへの継続的な更新をサポートする場合などです。2つのオプションがあります。

- Salesforce が、リモートシステムのリモートレコードの主キーまたは一意のサロゲートキーを保存する。
- リモートシステムが、Salesforce の一意のレコード ID または他の一意のサロゲートキーを保存する。

この同期パターンでインテグレーションキーを処理する場合に固有の考慮事項があります。

マスタシステム	説明
Salesforce	このシナリオでは、リモートシステムは SalesforceRecordId またはその他の一意のサロゲートキーをレコードから保存する必要があります。
リモートシステム	このシナリオでは、Salesforce が、リモートシステムの一意の識別子への参照を保存する必要があります。プロセスは同期するため、キーは、外部 ID 項目を使用した同じトランザクションの一部として指定できます。

複雑なインテグレーションシナリオ

このパターンでは、加工やプロセスオーケストレーションなどの複雑なインテグレーションシナリオを処理する場合、ソリューションごとに異なる考慮事項があります。

ソリューション	考慮事項
SOAP API または REST API	SOAP API と REST API は、オブジェクトに対する単純なトランザクションを提供します。集約、オーケストレーション、加工など、複雑なインテグレーションシナリオは Salesforce では実行できません。これらのシナリオは、リモートシステムまたはミドルウェアで処理する必要がありますが、ミドルウェアをお勧めします。
Apex Web サービスまたは Apex REST サービス	カスタム Web サービスでは、クロスオブジェクト機能、カスタムロジック、より複雑なトランザクションサポートを提供することができます。このソリューションは慎重に使用する必要があり、加工、オーケストレーション、エラー処理ロジックなどにミドルウェアが適しているか常に考慮する必要があります。

ガバナ制限

Salesforce プラットフォームは、マルチテナントという性質上、API を使用するときには制限があります。

ソリューション	制限
SOAP API、REST API、およびカスタム Apex API	<ul style="list-style-type: none"> API 要求制限—Salesforce は、24 時間あたりの API コール数に制限を適用します。制限は、Salesforce のエディションの種類とライセンス数に基づきます。たとえば、Unlimited Edition では、24 時間あたり、Salesforce または Lightning Platform ライセンスにつき、5,000 API 要求を使用できます。詳細は、『Salesforce の制限 クイックリファレンスガイド』を参照してください。 API クエリカーソル制限—ユーザは一度に最大 10 個のクエリカーソルを開くことができます。10 個を超えると、10 個のカーソルのうち最も古いものが解放されます。リモートアプリケーションが解放されたクエリカーソルを開こうとすると、エラーになります。たとえば、インテグレーションユーザログイン情報を共有する場合、最大クエリカーソル数を考慮する必要があります。ミドルウェアは、「ラウンドロビン」式に複数のユーザの要求を実行することが必要になる場合があります。 コール制限—作成、更新、およびクエリの制限については、補足の「データ量」を参照してください。
Bulk API	詳細は、補足の「データ量」を参照してください。

信頼できるメッセージング

信頼できるメッセージングは、個々のコンポーネント自体が信頼できない可能性がある場合、リモートシステムへのメッセージ配信保証の問題を解決しようとしています。Salesforce SOAP API および REST API は同期し、信頼できるメッセージングプロトコル (WS-ReliableMessaging など) の明示的なサポートは提供しません。

リモートシステムに信頼できるメッセージングシステムを実装して、エラーとタイムアウトのシナリオを適切に管理することをお勧めします。

ミドルウェア機能

次の表は、このパターンに使用されるミドルウェアシステムの望ましいプロパティの一覧です。

プロパティ	必須	望ましい	不要
イベント処理		X	
プロトコル変換		X	
変換/加工		X	
キューイングとバッファリング	X		
同期伝送プロトコル	X		
非同期伝送プロトコル			X
仲介ルーティング		X	

リモートコールイン

プロパティ	必須	望ましい	不要
プロセスコレオグラフィとサービスオーケストレーション		X	
トランザクション性(暗号化、署名、信頼できる配信、トランザクション管理)	X		
ルーティング			X
抽出、加工、読み込み (ETL)		X (大量/バッチ)	

例

ある印刷関連の消耗品とサービスを提供する企業が、Salesforceをフロントエンドとして使用して取引先と商談を管理しているとします。既存の取引先の商談は、クライアントサイトのプリンタを定期的に監視するオンプレミス型のプリンタ管理システム (PMS) からの印刷使用統計情報で更新されます。商談が作成されると、アウトバウンドメッセージがPMSに送信されて新規商談が登録されます。PMSはSalesforce IDを保存します (Salesforceが商談レコードのマスタ)。

次の制約が適用されます。

- PMSはコントラクトファーストインテグレーションに参加できます。この場合、Salesforceが契約を提供し、PMSがSalesforceサービス (Enterprise または Partner WSDL 経由で定義) のクライアント (コンシューマ) として機能します。
- Salesforceでのカスタム開発は行いません。

この例は、Salesforce SOAP API または REST API を使用して実装するのが最適です。

Salesforce で、次の処理を実行します。

- Enterprise または Partner WSDL をダウンロードして、リモートシステムに提供します。

リモートシステムでは、次の処理を実行します。

- Enterprise または Partner WSDL からクライアントスタブを作成します。
- インテグレーションユーザ(または、セッションIDが初期アウトバウンドメッセージで提供されると想定して、レコードを作成した商談所有者)のログイン情報を使用してAPIにログインします。
- アウトバウンドメッセージで提供されたSalesforceレコードIDに対して更新操作をコールし、関連項目の更新(使用統計情報)を渡します。

この例は、次の内容を説明しています。

- Salesforce同期APIクライアント(コンシューマ)の実装。
- レコード更新のためのSalesforceへのコールバック(前述の要求/返信アウトバウンドパターンと連携)。

第 6 章 データの変更に基づく UI の更新

コンテキスト

Salesforce を使用して、顧客のケースを管理します。カスタマーサービス担当者は、ケースについて顧客からの電話に対応します。顧客が支払いを行ったら、カスタマーサービス担当者は、支払い処理アプリケーションによる Salesforce のリアルタイム更新で、顧客による注文残高の支払いが正常に完了したことが示されているかどうかを確認する必要があります。

問題

Salesforce でイベントが発生した場合に、画面を更新することなく、潜在的な作業損失を回避して Salesforce ユーザーインターフェースでユーザに通知する方法は？

検討項目

このパターンに基づいてソリューションを適用する場合、次のようなさまざまな項目を検討する必要があります。

- 影響するデータを Salesforce に保存する必要があるか？
- このデータを表示するためのカスタムユーザーインターフェースレイヤを作成できるのか？
- カスタムユーザーインターフェースを呼び出すためのアクセス権はユーザに付与されるか？

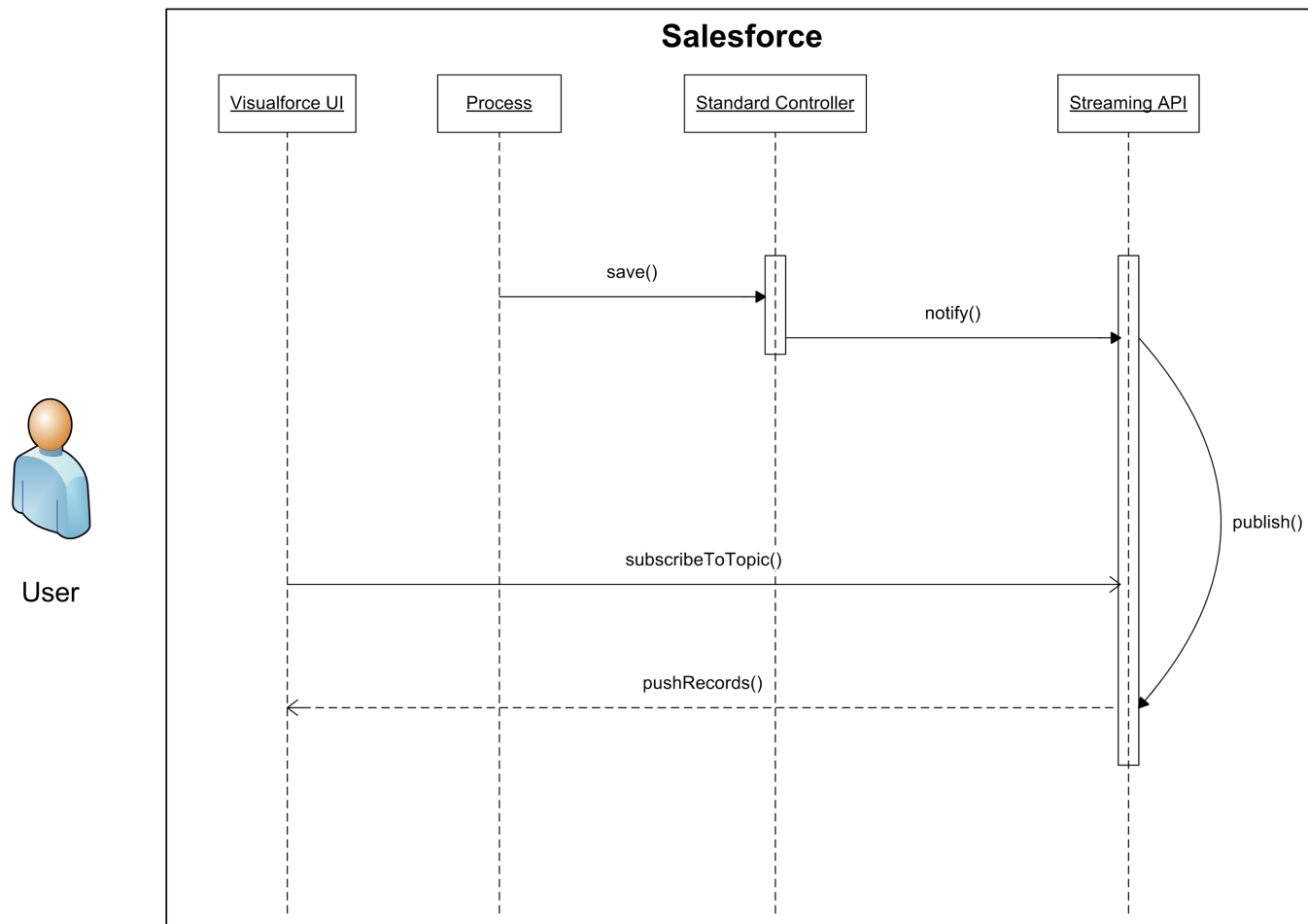
ソリューション

このインテグレーションの問題の推奨ソリューションは、Salesforce ストリーミング API を使用することです。このソリューションは次のコンポーネントで構成されています。

- 次の操作を可能にするクエリ定義を含む PushTopic
 - 更新をトリガするイベントの指定
 - 通知に含めるデータの選択
- ユーザーインターフェースで使用できる [Bayeux](#) プロトコルの JavaScript ベースの実装 (現在は [CometD](#))
- Visualforce ページ
- 静的リソースとして含まれる JavaScript ライブラリ

概要図

次の図は、ストリーミング API を実装して、通知を Salesforce ユーザーインターフェースにストリーミングする方法を示します。これらの通知は、Salesforce のレコードの変更によってトリガされます。



結果

利点

このパターンに関連するソリューションの適用には、次の利点があります。

- カスタムポーリングメカニズムを作成する必要がなくなる
- ユーザがフィードバックループを開始する必要がなくなる

サポートされない要件

このソリューションには次の制限があります。

- 通知の配信は保証されない
- 通知の順序は保証されない
- 通知は、Bulk API によるレコードの変更に基づいて生成されない

セキュリティに関する考慮事項

標準の Salesforce 組織レベルセキュリティに従います。HTTPS プロトコルを使用して、ストリーミング API に接続することをお勧めします。「[セキュリティに関する考慮事項](#)」を参照してください。

補足

最適なソリューションを実現するには、Salesforce でカスタムユーザーインターフェースを作成する必要があります。カスタムユーザーインターフェースを表示するために使用できる、適切なユーザーインターフェースコンテナを考慮することは不可欠です。サポートされるブラウザは、ストリーミング API の[ドキュメント](#)に記載されています。

例

ある電気通信会社が、Salesforce を使用して、顧客のケースを管理しているとします。カスタマーサービスマネージャは、ケースがカスタマーサービス担当者によって正常にクローズされた場合に自動的に自分に通知が送られるようにしたいと考えています。

このパターンで規定されるソリューションを実装する場合、顧客は次の操作を実行する必要があります。

- [状況] が [クローズ] で、解決が「成功」の状態ではケースが保存された場合に通知を送信する PushTopic を作成する。
- カスタマーサービスマネージャが使用できるカスタムユーザーインターフェースを作成します。このユーザーインターフェースは、PushTopic チャンネルに登録されます。
- マネージャのカスタマーサービス担当者によって生成されるアラートを表示するロジックをカスタムユーザーインターフェースに実装します。

付録

付録 A リソース — 外部

1. Hohpe、Gregor、Bobby Woolf 『*Enterprise Integration Patterns*』 (Boston: Addison-Wesley Professional、2003 年)
2. Microsoft Corporation 『*Integration Patterns (Patterns & Practices)*』 (Redmond: Microsoft Press、2004 年)
3. IBM Corporation 『*Application Integration Patterns*』 (IBM Corporation、2004 年)
4. 「同期、および非同期プロセス」 (IBM Corporation、最終アクセス日: 2013 年 3 月 18 日、
<http://publib.boulder.ibm.com/infocenter/adiehelp/v5r1m1/index.jsp?topic=%2Fcom.ibm.etools.ctc.flow.doc%2Fconcepts%2Fcsynchf.html>)
5. 「Hub and Spoke [or] Zen and the Art of Message Broker Maintenance」 (Enterprise Integration Patterns、最終アクセス日:
2013 年 3 月 18 日、http://www.eaipatterns.com/ramblings/03_hubandspoke.html)

付録 B リソース — Salesforce

開発者向けドキュメント

- [SOAP API 開発者ガイド](#)
- [REST API 開発者ガイド](#)
- [Salesforce ストリーミングAPI 開発者ガイド](#)
- [Bulk API 開発者ガイド](#)
- [Apex 開発者ガイド](#)
- 『Force.com SOQL および SOSL リファレンス』の「Salesforce Object Query Language (SOQL)」
- 『Force.com SOQL および SOSL リファレンス』の「Salesforce Object Search Language (SOSL)」
- [Salesforce の制限クイックリファレンスガイド](#)

ホワイトペーパー、プレゼンテーション、Web サイト

- [大量のデータを使用するリリースのベストプラクティス](#)
- [Force.com Connect: インテグレーションを成功させるための5つのパス](#)
- [リソースセンターの参照: 大量のデータ](#)

付録 C セキュリティに関する考慮事項

エンタープライズポートフォリオの有効なメンバーであるためには、関連するセキュリティのメカニズムに基づいてすべてのアプリケーションの作成と統合を行う必要があります。最新のIT戦略では、社内サービスとクラウドベースのサービスを組み合わせて使用します。

通常、クラウドツークラウドサービスのインテグレーションでは Web サービスと関連する認証に焦点を当てていますが、多くの場合、社内サービスとクラウドサービスを接続すると複雑さが増加します。このセクションでは、セキュリティツール、手法、Salesforce 固有の考慮事項について説明します。

リバースプロキシサーバ

「リバースプロキシサーバは、インターネットを介して侵入を試みる攻撃者から内部のサーバを保護するセキュリティ機能を実行することにより、社内イントラネット上の HTTP サーバ (中略) を保護するためにネットワーク DMZ 内に通常デプロイされるセキュリティデバイスです」。¹⁰

これは、「プロキシサーバの一種で、1つ以上のサーバからクライアントの代わりにリソースを取得します。その後、これらのリソースはクライアントに返されますが、クライアントからはプロキシサーバ自体から返されたように見えます。リバースプロキシは、その関連サーバ (通常は近隣のサーバ) の仲介として機能し、これらの関連する1つあるいは複数のサーバから提供されるリソースのみを返します」。¹¹

Salesforce 実装では、このようなサービスは外部ゲートウェイ製品で提供されることが一般的です。たとえば、Apache HTTP、lighttpd、nginx などのオープンソースオプションを使用できます。市販製品には、IBM WebSeal や Computer Associates SiteMinder などがあります。これらの製品は、内部要求者の代わりにすべてのアウトバウンド Salesforce 要求をプロキシおよび管理するように簡単に設定できます。

暗号化

一部の企業では、社内アプリケーションとクラウドベースのアプリケーション間で、選択したトランザクションやデータ項目を暗号化する必要があります。組織で追加のコンプライアンス要件に準拠する必要がある場合は、次のような代替方法を実装できます。

- Salesforce の CipherCloud、IBM DataPower、Computer Associates など、社内の市販の暗号化ゲートウェイサービス。これらの各ソリューションでは、暗号化エンジンまたはゲートウェイは、暗号化されたペイロードを送信および受信することによってトランザクション境界で呼び出されるか、HTTP 要求を実行する前に特定のデータ項目を暗号化または復号化するために呼び出されます。


¹⁰ 「リバースプロキシサーバとは」、IBM Corporation、最終アクセス日: 2012年4月11日、http://publib.boulder.ibm.com/infocenter/sametime/v8r5/index.jsp?topic=%2Fcom.ibm.helpsametime.v851.doc%2Fconfig%2Fst_adm_port_rproxy_overview_ch.html

¹¹ 「リバースプロキシ」、Wikipedia、最終アクセス日: 2012年4月11日、http://en.wikipedia.org/wiki/Reverse_proxy。

- Salesforce Shield Platform Encryption など、クラウドベースのオプション。Shield Platform Encryption では、重要なプラットフォーム機能を保持しながらデータに新しいセキュリティ層が追加されます。選択したデータは、高度な鍵派生システムを使用して暗号化されます。以前より安全にデータを保護することができます。詳細は、Salesforce オンラインヘルプを参照してください。

特殊な WS-* プロトコルのサポート

セキュリティプロトコル (WS-* など) の要件に対応するには、次の代替方法をお勧めします。

- セキュリティ/XML ゲートウェイ — WS-Security ログイン情報 (IBM WebSeal または Datapower、Layer7、TIBCO など) をトランザクションストリーム自体に挿入します。このアプローチでは、アプリケーションレベルの Web サービスまたは Salesforce からの Web サービスの呼び出しを変更する必要はありません。また、Salesforce インストールでこのアプローチを再利用することもできます。ただし、既存のセキュリティゲートウェイアプローチへの適切な WS-Security の導入を管理するには、追加の設計、設定、テスト、メンテナンスが必要になります。
 - 伝送レベルの暗号化 — 双方向 SSL および IP 制限を使用して通信チャネルを暗号化します。このアプローチでは、WS-* プロトコル自体を直接実装することはありませんが、ユーザ名とパスワードを渡すことなく、社内アプリケーションと Salesforce 間の通信チャネルを保護します。また、Salesforce で生成されるクラスを変更する必要もありません。ただし、(アプリケーション自体またはミドルウェア/ESB レイヤで)一部の社内 Web サービスを変更する必要が生じる場合もあります。
 - Salesforce カスタム開発 — WSDL2Apex ユーティリティを使用して WS-Security ヘッダーをアウトバウンド SOAP 要求に追加します。これにより、内部サービス呼び出す WSDL ファイルから Java のような Apex クラスが生成されます。DMZ 内のバックエンド Web サービスまたは追加コンポーネントを変更する必要はありませんが、次のことが必要になります。
 - 構築およびテスト作業の増加
 - WS-Security 属性のコードを手動で作成するための比較的複雑な手動プロセス (Apex コード内の XML 逐次化など)
 - より高度な長期メンテナンス作業
-  **メモ:** 最後のオプションは複雑でリスクが高く、このようなインテグレーションは Salesforce の定期更新に基づいて定期的に確認する必要があるため、お勧めできません。