

クラウドコンピューティング連携のための パブリッシュ/サブスクライブアーキテクチャの提案

M2010MM015 影石 英紀

指導教員 青山 幹雄

1. はじめに

現在、ネットワークを介してサービスを提供するクラウドコンピューティングに注目が集まっている。さらに、Salesforce.com の CRM や GoogleApps などインターネット経由でアプリケーションを利用する SaaS (Software as a Service) が実務で使われている。このようなクラウドコンピューティングの普及に伴い、SaaS間、SaaSとオンプレミスアプリケーション間の連携の必要性が増加している。しかし、SaaS やオンプレミスアプリケーションは、ベンダごとに開発されていることから、連携が困難である。本稿ではクラウドコンピューティングにおける SaaS の連携方法を提案する。

2. 研究課題

現在、SaaS やオンプレミスアプリケーションは企業ごとに開発されていることからアーキテクチャが異なる。そのため各アプリケーションはインタフェースとメッセージ交換の方法が異なり、連携を行うためにはデータや振舞いを整合する必要がある。

3. 関連研究

SaaS の連携方法としてリクエストレスポンスを用いた連携方法が提案されている[1]。また、OASIS により Web サービスにおけるパブリッシュ/サブスクライブの処理方法が定義されている[3]。

3.1. SaaS/クラウドサービス間のメッセージ連携方法

SOA(Service-Oriented Architecture)に基づくクラウドコンピューティング環境における SaaS の連携方法が提案されている[1]。

サービスコンシューマはブローカにリクエストを送り、ブローカはデータの変換を行う。そして、サービスコンシューマにレスポンスを送ることで、異なるクラウドコンピューティング環境の SaaS の連携を実現している。この研究ではサービスコンシューマからブローカにリクエストレスポンスによる 1 対 1 のタイミングの連携に対応している。しかし送信側と受信側のメッセージの関係が多対多のタイミングによるメッセージ交換に対応していないという問題がある。

3.2. WS-Notification

WS-Notification は SOA の技術基盤である Web サービス

スを基礎としてパブリッシュ/サブスクライブの処理方法を定義している[3]。

(1)WS-BaseNotification

通知を送信する NotificationProducer とメッセージを受信する NotificationConsumer の 2 つの主要なコンポーネント間のメッセージの方法を示す。NotificationProducer は、発行する通知に関するトピックを公開し、NotificationConsumer はサブスクリプションを NotificationProducer に送信する。

(2)WS-BrokeredNotification

NotificationConsumer と NotificationProducer の間に NotificationBroker を用いることで 2 者間の関係を分離する。NotificationBroker では NotificationProducer が公開するトピックの情報や NotificationConsumer からのサブスクリプションの管理、通知の分散に関する処理などを行う。

(3)WS-Topics

メッセージのトピックの構造を木構造で規定している。

4. アプローチ

クラウドコンピューティング環境における SaaS 間のメッセージ交換のタイミングに着目し、タイミング整合の問題を解決するための方法を提案する。本稿では送信側と受信側のタイミングの分類を行い、それぞれのタイミングをモデル化し分析を行う。

Salesforce.com の CRM と Google の SaaS 連携を前提とし、Salesforce.com のサービスを送信側、Google のサービスを受信側とする。送信側のメッセージはアウトバウンドメッセージとする。

4.1. 送信側のタイミング

Salesforce.com からのアウトバウンドメッセージは Salesforce.com のサービス上でユーザの操作によりトランザクションが実行される。また、アウトバウンドメッセージは One-Way のメッセージでリクエストを要求していないことから非同期メッセージとなる。

4.2. 受信側のタイミング

受信側の SaaS アプリケーションではユーザの要求に対して応答を返す同期型のメッセージ交換となる。

4.3. タイミング整合

送信側の非同期メッセージ送信と受信側の同期メッセージ交換の間で連携を行う場合、送受信のアプリケーション間にブローカを設置し、タイミング整合する必

要がある。送信側と受信側のトポロジの関係を1対1とし、送信側と受信側のタイミングの関係を3つ分類する。

(1)送信側と受信側のタイミングが1対1

送信側からのメッセージ通知毎に、受信側の要求によりブローカから1回で送信する。

(2) 送信側と受信側のタイミングがN対1の場合

送信側から複数のメッセージ通知を受信側に1回で送信する。

(3) 送信側と受信側のタイミングがN対Mの場合

送信側から複数のメッセージ通知を複数回に分けて送信する。

4.4. クラウドコンピューティングへのパブリッシュ/サブスクライブアーキテクチャの適用

SaaS 間の振舞いの整合を行うにはメッセージ交換のタイミングの問題を解決する必要がある。本稿では、送信側と受信側のメッセージの関係が多対多の連携を実現するためにパブリッシュ/サブスクライブアーキテクチャを用いて、メッセージのタイミングの問題を解決し、多対多のタイミングによる SaaS 連携を実現する。

5. クラウド連携のためのパブリッシュ/サブスクライブアーキテクチャの提案

5.1. アーキテクチャに対する要求

タイミングの整合を実現するために、ブローカは以下の3つの要求を満たす必要がある。

(1)複数のメッセージ通知を受信し保持する

送信側のメッセージはユーザのトランザクションにより、非同期で送信されることから一方の複数のメッセージが送信される。受信側はリクエストレスポンス型の同期メッセージであることから、送信側から複数のメッセージを保持しておく必要がある。

(2)受信側の要求に応じて通知するメッセージの分類

送信側から送られるメッセージを受信側にすべて送信するのはネットワークの負荷がかかり、非効率的である。SaaS APの要求を指定し、ブローカアーキテクチャがフィルタリングを行い SaaS APの要求を満たすメッセージを送信すべきである。

(3)受信側にメッセージの通知

ブローカで保持したメッセージを、ブローカは受信側に通知する必要がある。

以上の3つの要求を満たすためにパブリッシュ/サブスクライブアーキテクチャを用いて送信側、受信側間のタイミングを整合するブローカアーキテクチャを提案する。

5.2. ブローカアーキテクチャの機能

5.1 節の要求を満たすブローカのユースケース図を図1に示す。ブローカの機能はサブスクリプション登録、アウトバウンドメッセージ受信、フィルタリング、メッセージ受信、メ

ッセージストアへ登録の5つである。

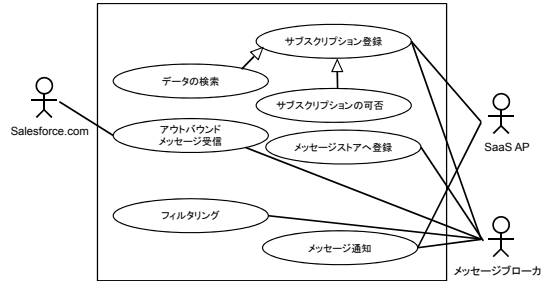


図1 クラウド連携ブローカのユースケース図

5.3. ブローカアーキテクチャの構成

5.1 節の3つの要求を満たすクラウド連携ブローカアーキテクチャの構成を図2に示す。ブローカアーキテクチャはWS-BaseNotificationの仕様を基に作成する。クラウド連携ブローカは2つのマネージャ(PublisherRegistration マネージャ, Subscription マネージャ)と3つのデータベース(パブリッシュストア, サブスクリプションストア, メッセージストア)から構成される。

PublisherRegistration マネージャで Salesforce.com から送信されるアウトバウンドメッセージの情報を管理し、Subscription マネージャで SaaS AP のサブスクリプション情報を管理する。SaaS アプリケーションとクラウド連携ブローカ間のメッセージプロトコルは標準メッセージである SOAP/REST を使用する。

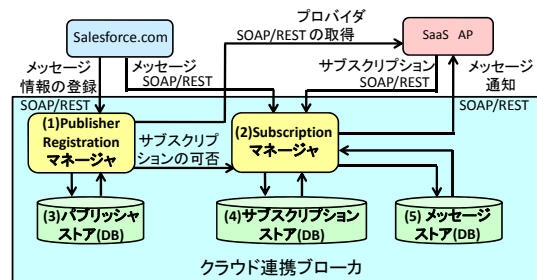


図2 クラウド連携ブローカアーキテクチャ

次にブローカ内の各コンポーネントの機能を定義する。

(1) PublisherRegistration マネージャ

Salesforce.com からの送信されるアウトバウンドメッセージの登録を行う。Subscription マネージャからサブスクリプションの可否を検索し、結果を返信する。

(2) Subscription マネージャ

SaaS AP からのサブスクリプションの登録を行う。サブスクリプションの可否の問い合わせ、メッセージのフィルタリング、データの検索、SaaS AP への配信を行う。

(3) パブリッシュストア

Salesforce.com のメッセージを送信するパブリッシュを登

録する。

(4) サブスクリプションストア

受信側のアプリケーションのサブスクリプション情報を登録するデータベース。

(5) メッセージストア

Salesforce.com から送信されたアウトバウンドメッセージの要素を保存するデータベース。

5.4. ブローカの振舞い

ブローカの振舞いをサブスクリプション登録、メッセージ通知の2つに分けて説明する。

(1) サブスクリプションの振舞い

1) サブスクリプションのアクティビティ

SaaS AP のサブスクリプションの実行からシステムのアクティビティを図3に示す。

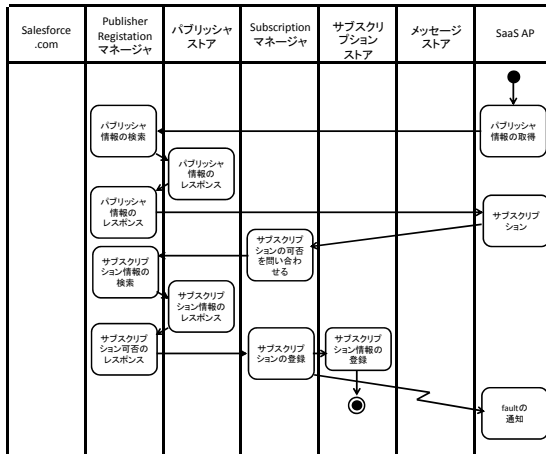


図3 サブスクリプション登録のアクティビティ図

2) サブスクリプションの処理の流れ

SaaS AP のサブスクリプションの実行からシステムの処理の流れを図4に示す。

- (a) クラウド連携アーキテクチャが公開するパブリッシャ情報を基に、SaaS AP は PublisherRegistration マネージャからパブリッシャ情報を要求する。
- (b) PublisherRegistration マネージャはパブリッシャストアにパブリッシャ情報の検索を行い、結果を SaaS AP に返信する。
- (c) SaaS AP は要求する Subscription 情報を Subscription マネージャに送信する。Subscription マネージャは SaaS AP のサブスクリプションの可否を PublisherRegistration マネージャに問い合わせる。
- (d) PublisherRegistration マネージャはパブリッシャにパブリッシャ情報の検索を行い、Subscription マネージャに返信する。
- (e) Subscription マネージャはサブスクリプション可能であれば、サブスクリプションストアにサブスクリプションの登録を行い、サブスクリプション不可能であれば fault を SaaS AP に返信する。

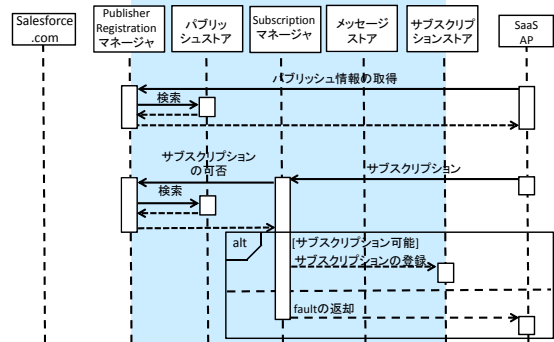


図4 サブスクリプション時のブローカの振舞い

(2) メッセージ通知

1) メッセージ通知のアクティビティ

Salesforce.com のアウトバウンドメッセージから SaaS AP へメッセージ通知を行うアクティビティを図5に示す。

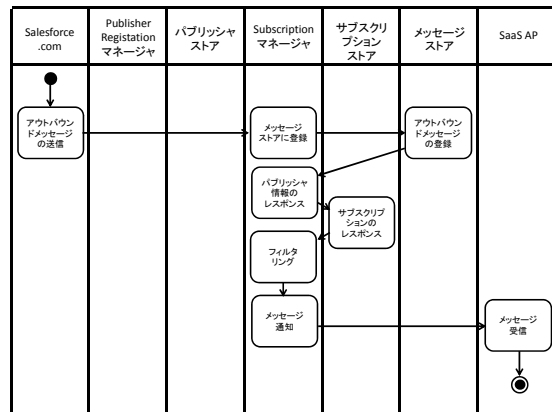


図5 パブリッシャからメッセージ通知のアクティビティ図

2) メッセージ通知の処理の流れ

Salesforce.com からのアウトバウンドメッセージから SaaS AP へ通知までのブローカの振舞いを以下に示す(図6)。

パブリッシャ、フィルタリング、メッセージ通知の順で実行する。

- (a) **パブリッシャ**: Salesforce.com がアウトバウンドメッセージを Subscription マネージャへ送信する。Subscription マネージャは受信メッセージを受け取り、メッセージストアにメッセージの内容を登録する。
- (b) **フィルタリング**: Subscription マネージャはサブスクリプションストアに事前に SaaS AP に指定されたサブスクリプションの取得を行う。さらに、保持している Salesforce.com からのメッセージの内容を取得する。Subscription マネージャは WS-Topics で指定された条件に基づいてメッセージのフィルタリングを行う。
- (c) **メッセージ通知**: Subscription マネージャはフィルタリングしたメッセージを、受信側の要求に応じたタイミングで SaaS AP

APへ配信する。

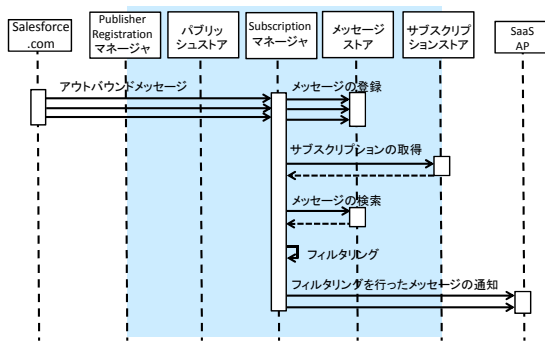


図6 フィルタリングと通知時のブローカの振舞い

5.5. タイミング制御

Subscription マネージャで送信側と受信側のメッセージのタイミング整合を行うために、WS-BaseNotification の仕様に基づき、サブスクリプションメッセージを送信する。サブスクリプションメッセージでは次の3つのプロパティを指定する。

- (1) ProducerProperties: サブスクリプションの可否の条件式を指定する。
- (2) MessageContent: サブスクリプションするメッセージの内容をXPathで指定する。
- (3) SubscriptionPolicy: ユーザが要求する配信方法の指定する。SubscriptionPolicyでメッセージの通知回数を制御することでSaaS APの要求するタイミングでブローカから通知される。

図6より、ブローカがSalesforce.comからのアウトバウンドメッセージを受信し、フィルタリングを行うことでタイミングの異なるSaaSからのメッセージの連携を可能としている。

6. 評価と考察

6.1. 評価

(1) タイミングの異なるメッセージ交換の整合

提案したアーキテクチャはパブリッシュ/サブスクライブを用いた。パブリッシュ/サブスクライブモデルは、非同期の連携を可能としていることから、送信側の非同期メッセージ、受信側の同期メッセージの2つの異なるタイミングのアプリケーションの連携の実現を可能としている。

(2) 受信側アプリケーションの要求に応じた配信

受信側のSaaS APのサブスクリプション方法はWS-BaseNotificationの仕様に基づきサブスクリプションの指定を行うことで要求に応じた配信することが可能である。

6.2. 考察

論文[1]では、SaaS連携を実現しているがサービスコンシューマからのリクエストレスポンスによりデータ連携をしていることから、1対1の連携になる。このため複数のタイミングによるN回のメッセージの連携に対応してなかった。パブリッシュ/サブスクライブアーキテクチャを用いることで、送

信側と受信側のメッセージの関係が多対多の連携を可能とした。

7. 今後の課題

今後の課題として、以下の2つの課題が挙げられる。

(1) クラウド環境での検証

本稿の目的はクラウド環境における送信側と受信側のメッセージの関係が多対多のタイミングによるメッセージ交換である。パブリッシュ/サブスクライブアーキテクチャとWS-BaseNotificationをクラウド連携に適用し、アーキテクチャを提案した。Salesforce.comとGoogleのSaaSを用いてプロトタイプを実装し、クラウド環境におけるブローカアーキテクチャの検証を行う必要がある。

(2) メッセージ変換方法の検討

異なるプロトコルを利用するSaaSアプリケーションとの連携を実現するために、SOAPとREST、JSONとのメッセージの相互変換の詳細を検討する。

8. まとめ

クラウドコンピューティング環境における多対多のタイミングによるメッセージ交換を実現するために、本稿ではメッセージの振舞いに着目し、パブリッシュ/サブスクライブアーキテクチャの適用を検討した。さらに、WS-BaseNotificationを適用し、提案したアーキテクチャのブローカ内でメッセージのタイミングの制御を行い、受信側のSaaSの要求に応じたメッセージの配信を可能とした。

参考文献

- [1] バンダリ スワティ, 他, SaaS/クラウドサービス間のメッセージ連携方法の提案と評価, 情報処理学会 第72回 全国大会 講演論文集 (3), Mar. 2010, pp. 389-390.
- [2] P. Th. Eugster, et al., The Many Faces of Publish/Subscribe, ACM Computing Survey, Jun. 2003. pp. 114-131.
- [3] OASIS, Web Services Base Notification (WS-BaseNotification), Ver. 1.3, 2006, http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-os.pdf.
- [4] OASIS, Web Services Brokered Notification (WS-BrokeredNotification), Ver. 1.3, 2006, http://docs.oasis-open.org/wsn/wsn-ws_brokered_notification-1.3-spec-os.pdf.
- [5] OASIS, Web Services Topics (WS-Topics), Ver. 1.3, 2006, http://docs.oasis-open.org/wsn/wsn-ws_topics-1.3-spec-os.pdf.