


iPaaSによるイベントドリブン インテグレーション：リアルタイムコネク ティビティの実力



Jesse Menning
CTOグループアーキテクト

solace●



iPaaSでの一般的な実装では、ポーリングによって変更を検知するため、重要な変更が数秒、数分、あるいは数時間も通知されない状況が発生していました。企業に対する消費者の要求が厳しくなるにつれ、このような数秒間の空白でさえユーザエクスペリエンスに悪影響をおよぼすようになります。”

Jesse Menning

Solace CTO オフィス アーキテクト

© Solace

本書の一部または全部を、Solaceの許可なく電子的、機械的、複写、記録、その他のいかなる方式による形式をもって、複製、検索システムへの保存または送信することを禁止します。各種許可に関しましては、legal@solace.comまでお問い合わせください。

目次

- 今日のビジネスにおけるリアルタイムデータの重要性.....4
- iPaaSソリューションの導入効果.....7
- リアルタイムデータが抱える課題.....8
 - グローバルでのコネクティビティ：
制限された範囲..... 9
 - トラフィックの急増：圧倒的なデータ量.....10
 - エラー処理とディザスタリカバリ：極度の複雑さ.....11
 - 新機能の追加：密結合による弊害.....12
- イベントドリブンになる意味とは？ 14
 - Tイベントについての基礎.....16
- イベントドリブンインテグレーションによるリアルタイム関連課題の解決 18
 - グローバルコネクティビティ：
グローバル規模での同時配信.....18
 - トラフィックの急増：バッファリングとキューイング..18
 - エラー処理とディザスタリカバリ：情報の保存.....20
 - 新機能の追加：疎結合とアジリティ.....22
- まとめ 24

今日のビジネスにおけるリアルタイムデータの重要性

今日のビジネス環境においては、単にアプリケーションを連携させるだけの対応では、適切な意思決定を行うことができません。

競合他社の先を行くためには、より優れたカスタマーエクスペリエンスを提供し、市場の変化に応じた迅速かつ効果的な意思決定を下す必要があります。これらの目標を達成するためには、単にA地点からB地点へ情報を届けるだけでなく、さまざまなシステムが必要とする情報を、必要なときに、環境を超え、世界中にインテリジェントに配信する領域へと進まなければなりません。

以下のようなシナリオについて考えてみてください：

- ある顧客がセールを利用したところ、トピーカにある物流センターの在庫が突然1,000から0になった。このような状況を、中国の製造工場、ダラスにいる地域の営業担当者、ニューヨークにいるエグゼクティブ、シカゴ近郊にあるデータレイクにどうやって伝達しますか？ もし、社内の他部門がその情報から利益を得られるとしたら、それを彼らに届けるのにどれくらいのかかりませんか？
- サンパウロでこっそり外出しようとしている10代の若者が、両親のホンダ・シビックのエンジンを掛けました。すぐに両親へそれを伝え、車の発進を止めるにはどうしたらよいでしょう？
- ブルックリンに住む高齢者が、既に服用中の薬と、副作用を起こす可能性のある処方箋による薬を、地元の薬局で調剤してもらおうとしています。薬局は副作用を即座にチェックするサービスを利用しているが、システムに何かあった場合に、そのような重要な情報を失うことがないようにするにはどうすればよいのでしょうか？

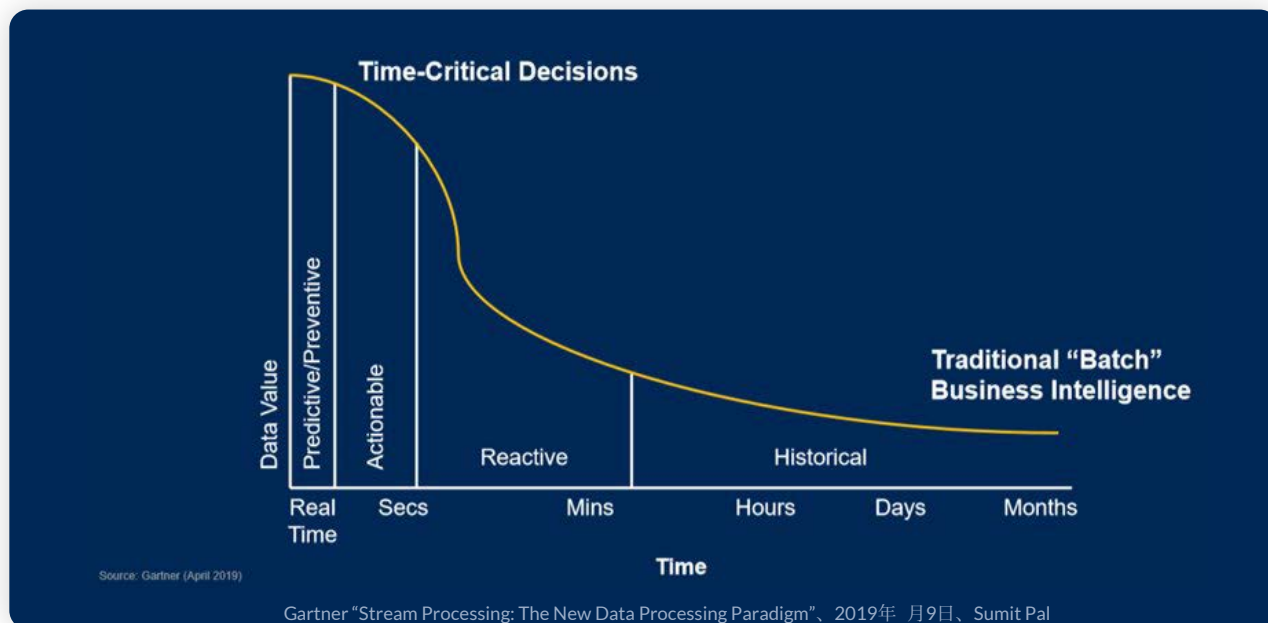
- シンガポールで不正取引が相次いだ後、クレジットカード処理会社は、不正検出システムに人工知能を追加する必要に迫られました。承認プロセスにステップが追加されることで、既存の機能が中断され、処理時間が長くなることはないのでしょうか？立ち上げから稼動までどのくらいかかるのでしょうか？

これらの各シナリオで重要なのは、迅速、安全、信頼性が高く、俊敏な方法で全ての情報を漏れなく抽出するだけでなく、ネットワークや個々のシステムを不要な情報で圧迫することなく、必要な場所に効果的に情報を提供することです。そして、このような飛躍を遂げた企業は、市場で大きな優位性を手にすることができます。

本書では、SolaceのCTOオフィスでアーキテクトを務めるJesse Menning が、これらの重要な課題に対応するためにテクノロジーリーダーが採用すべき基本的なテクノロジーとその考え方へのシフトについて解説します。



ガートナーによる情報のデータ価値を示すグラフ：



前述のシナリオで示した課題については、事後対応的に使用するデータから、行動につながるデータ、そして予測可能なデータへと移行し、より優れた、より安全で楽しいカスタマーエクスペリエンスを提供することで解決が図れます（もっとも、サンパウロの若者のケースは該当しないかもしれませんが）。しかし、リアルタイムな対応を加速して目標に到達した後も、容易に継続的なイノベーションを実現できるならば、より多くの企業がその取り組みを実施するでしょう。

リアルタイムな企業となるためには、次の4点を満たす必要があります。

1. 世界中へ、そしてクラウドやオンプレミス環境全体へ、瞬時に情報を配信できること。
2. AI、IoT、モバイルコンピューティングなどのテクノロジーに関わる新たな大量のデータをバックエンドシステムに多大な負荷をかけることなく処理できること。
3. システムが利用できない場合や、不安定あるいは障害が発生した場合でも、重要な情報が失われないこと。
4. お客様の要求を満たし、規制を遵守し、自社を競合優位に導くことができる革新的なソリューションによって、市場の変化に迅速に対応できること。

しかし本書で説明するように、Integration Platform as a Service (iPaaS) だけで、これらすべての要求を満たすことができるわけではありません。

iPaaSソリューションの導入効果

多くの企業が、自社の情報配信を加速させるためにiPaaSを利用するようになっていきます。iPaaSソリューションのアピールポイントは明確で、シンプルで標準的、かつ俊敏な方法で、サイロ、特にSaaS（Software as a Service）アプリケーションを横断して情報を統合できる点です。たとえば、SAPシステムで入力された注文についてSalesforce側でも把握したい、またはSalesforceで作成したアカウントをSAP側でも把握したいと考えるのは自然なことです。

iPaaSソリューションでは、グラフィカルで簡単に利用できるインターフェースによって、インテグレーションに関わる昨今の課題を解決します。開発者チームが世界中に分散配備されている状況下で、昨今のインテグレーションに関わる課題を解決するためには、軽量で標準化され、共有可能でシンプル、強力なツールセットが不可欠です。このような課題には、企業が対処すべき次のような内容が含まれます：

- 特定のアプリケーションからのデータを他でも利用できるフォーマットへ変換すること。
- データベースやディスクドライブに保存されているファイル、あるいはルックアップテーブルの情報を使ってメッセージを充実させること。
- ペイロード内の情報を基にメッセージをルーティングすること。
- 複雑または独自のプロトコルを使用しているアプリケーションに接続すること。
- 1つのメッセージを複数に分割、または複数のメッセージを1つに統合すること。
- 複数の同期型API定義の内容を管理し、内部および外部のお客様へ公開すること。

業界の専門家は、iPaaSソリューションが、ガートナーの考える「ハイブリッドインテグレーションプラットフォーム」(HIP)で重要な部分を占めていることに同意しています。HIPは、古い概念であるエンタープライズサービスバス(ESB)を置き換えるもので、以下を実現します：

- ランタイムを縮小し、俊敏性を向上させ、コンテナやクラウド環境への容易な導入を可能とします。
- クラウド上で提供されるGUIによって、より容易な開発を可能とします。
- ソース管理、デプロイメント、そして管理機能がバンドルされています。
- 専任のインテグレーショングループの専門知識に頼るのではなく、オンデマンドでHIPの機能を利用するプロジェクトチームでの利用を狙っています。

リアルタイムデータが抱える課題

iPaaSは分散アプリケーションのインテグレーションを容易かつ迅速に実現します。それでは、リアルタイムデータでは、何が異なるのでしょうか？特にiPaaSを単独で使用する場合、リアルタイムデータに関する次の4つの要件については注意が必要となります：

1. グローバルなコネクティビティ
2. トラフィックの急激な増加
3. エラー処理とディザスタリカバリ
4. 新機能の追加

以降のセクションでは、これら各項目について若干深掘し、なぜiPaaSだけでは解決が難しい問題なのか、その理由を明らかにすることにします。

グローバルでのコネクティビティ：制限された範囲

今日のビジネス環境は相互に関連し合い、グローバルな性質を持っているため、トピカで起きたことは世界中のシステムに影響を及ぼす波及効果があります。情報がトピカに滞留していたり、迅速に発信されなかったりすれば、ビジネスは短期間の問題や機会に対応する機会を失うかもしれません。たとえば次のようなケースがこれに該当します：

- ジャストインタイム方式を採用する中国の工場で、誤った製品の製造に時間を費やしてしまう。
- 営業担当者がまだ提供可能となっていない部品を販売し、お客様に混乱を与えてしまう。
- 現場の最新状況に、価格設定アプリケーションが追いつかない。
- この地域におけるエグゼクティブのマーケティング戦略変更が遅れが生じる。

従来型のiPaaS実装では、ポーリングによって変更を検知するため、重要な変更が数秒、数分、あるいは数時間も気付かれないような状況が発生していました。企業に対する消費者の要求が厳しくなるにつれ、このような数秒間の空白でさえユーザエクスペリエンスに悪影響をおよぼすようになります。



REST APIを使ったiPaaSソリューションの場合、イベントや情報を複数のデータセンター（オンプレミスや異なるクラウド間）、サプライヤーと契約業者間で移動することは非常に困難です。iPaaSプロバイダーがセキュリティ上の懸念からVPN接続を許可しないか、iPaaSがVPN接続を許可したとしても、ネットワークが頭痛の種になり、アーキテクチャが脆いものとなります。

さらに、iPaaSソリューションは、通常、トランザクションをポイントツーポイントで逐次処理します。すべてのアプリケーションが必要な情報を同時に取得するのではなく、最初に中国の工場が、次に営業担当者が情報を取得し、その後その他のアプリケーションが取得するといった状況が発生します。iPaaS自体もプロセスの次のステップを追跡する必要があるため、静的で脆いアーキテクチャになってしまいます。

トラフィックの急増：圧倒的なデータ量

サンパウロの00代の若者が車を出そうとしたのは、両親にとっては重大な問題ですが、現実的には世界中の人々が毎分何百万台もの車のイグニッションキーを回し、車を走らせようとしています。そしてインフラストラクチャーに対する需要は、1日の中でも大きく変化します。午前3時にエンジンをかける車の数ならば完全に制御可能かもしれませんが、ラッシュアワーには膨大な量のデータが一気に押し寄せます。

iPaaSにおいても、突然データが急増した場合に、受信したイベントをリアルタイムで処理するのが困難になる場合があります。iPaaSの処理は、同期APIからのリクエスト、あるいは一定のポーリング間隔で実施されます。同期APIのリクエストによって処理が行われる場合、iPaaS内部でのトラフィック急増に対処するためには、通常、受信情報のフロー処理に使用できるスレッドの数を増やすという対応が必要となります。これがメモリや処理能力の過剰な割り当てにつながることも少なくありません。

仮にiPaaSがこのような負荷に対応できたとしても、ダウンストリームに存在するシステムが不均一で不安定なIoTのデータストリームを処理できない可能性もあります。その場合、iPaaSはダウンストリームのシステムのためにトラフィックをバッファリングする必要があります。従来のiPaaSのアーキテクチャーであれば、通常はスレッドをオープンにしたまま長期間維持しておくこととなりますが、これはどのようなインテグレーションソリューションの場合でも、決して推奨できる方法ではありません。

エラー処理とディザスタリカバリ：極度の複雑さ

ブルックリンに住む高齢者は、処方箋の情報を受け取り、リアルタイムに副作用を確認できたことで、危険な状況を未然に防ぐことができましたが、その情報を失わないこともまた同様に重要でした。

インテグレーションアーキテクトにとってもっとも複雑で時間を要する作業の1つが、何か問題が発生した場合、特に対象が重要なデータの場合に備え、ソリューションに耐障害性を持たせておくことです。そしてiPaaSソリューションもその例外ではありません。ある主要なiPaaSソリューションでは、開発者は堅牢なエラー処理の開発にプロジェクトの 分の1を費やすと予測しています。リトライ、リプレイ、人的介入が必要なエラー、iPaaSのランタイム自体の障害など、関係する要素が多いことを考えると、これは良い見積もりだと思われれます。

おそらく、JavaScriptでカスタムコードスニペットのライブラリを作成することによって、iPaaSプラットフォーム内でこれらの問題を解決しようとすることができます。失敗したトランザクションを手作業で補正し、アダプターを使ってトランザクションがディスクに保存されるようにすると、iPaaS本来のシンプルさが失われてしまいます。

それでも、時には本当の災害が発生する場合があります。ビジネスクリティカルなデータでは、ディザスタリカバリプランは、バックアップのロケーションが稼動したときに、すべての処理中の情報が存在し、新しいロケーションでフローする準備ができていることを保証する必要があります。iPaaSソリューションの場合にも、災害発生時にデータを保存しておくためには、やはり手作業による回避策が必要となります。

新機能の追加：密結合による弊害

現在のマイクロサービスアーキテクチャーは、数十の個別コンポーネントで構成されています。例えば、シンガポールのクレジットカード処理会社は、レジが「ピッ」と承認するまでに複雑な機能を要求することがあります。AI機能を追加することで、不正行為による損失を防ぐことができる可能性はありますが、テストプロセスが複雑になり、パフォーマンスが低下し、依存関係が増えることで、その実装がさらに複雑になります。

俊敏性を支援するために、iPaaSの競合他社は通常、統合されたソース・コントロールとデプロイメント機能を備えています。

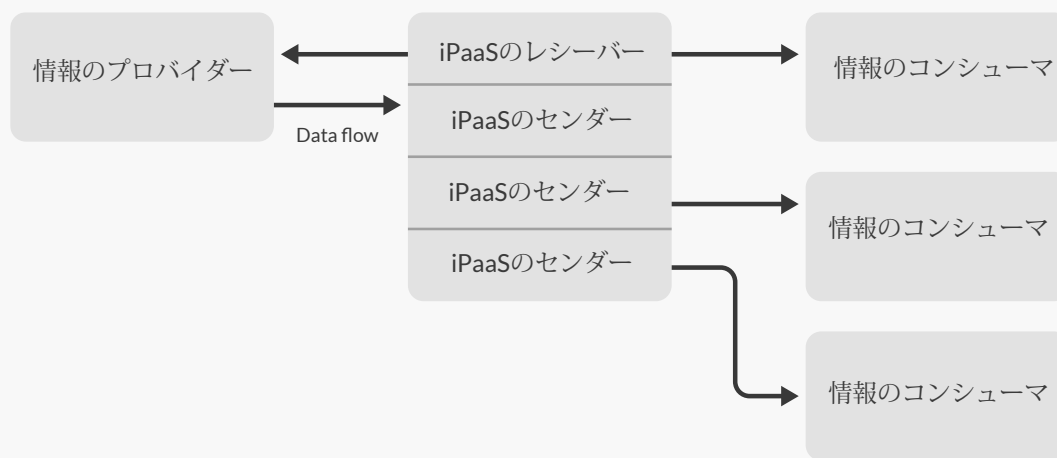
すぐに使えるソリューションとしてこれらのコンポーネントを利用できるため、iPaaSの構築を開始するために必要なインフラストラクチャーを削減することができます。GitリポジトリのセットアップやJenkinsの設定は必要はありません。また、ほとんどのiPaaSには、単一のマスタープロセスから複数の送信プロセスを呼び出す機能が含まれており、クレジットカード承認のようなビジネスプロセスをよりモジュール方式でオーケストレーションすることができます。

しかし、コードプロモーションインフラストラクチャーを導入し、サブプロセスを使用しても、従来のiPaaSオーケストレーションは単一のデプロイメントアーティファクトを作成します。

このコードブロックは、情報を受け取り、処理ロジックを適用し、ダウンストリームのコンシューマに送信するといった役割を担う複数のコンポーネントで構成されています。このアーキテクチャーによって生じる依存関係（密結合 (tightly coupled) と呼ばれる場合もあります）は、市場に革新的な機能を提供する上での妨げとなり、以下のような課題を抱えています：

- 情報のコンシューマを追加または更新すると、マスターフローの再テストと再デプロイメントが必要になります。
- 新しく情報のコンシューマを追加する際、特にその処理が遅い場合、または信頼性が低い場合、プロセス全体に悪影響を与えます。
- マスターフローはサブフローをオーケストレーションするために（また、複数のエラーシナリオも予測されるため）複雑になります。

複数のチームが扱う異なる情報のコンシューマが存在する場合、責任分担が曖昧になる可能性があります。これは特に、HIPスタイルの一般開発者が独自のインテグレーションを作成する際に発生します。



複数の密結合なコンシューマを持つiPaaSソリューション

それでは、どうすればこれらの課題を解決できるのでしょうか？求められるのは、瞬時でのグローバルな情報ルーティング、急増するトラフィックへの対応、情報の保存、革新的なテクノロジーの採用を可能にすることで、iPaaSソリューションの強みを十分に活かせるパートナーシップを構築することなのです。

イベントドリブンになる意味とは？

イベントとは、ビジネス上での状態変化の瞬間であり、あらゆる企業活動の源泉となるものです。前述の例でいえば、トピカでの注文、サンパウロでのシビックの始動、ブルックリンでの老人の処方箋受け取り、そしてシンガポールでのクレジットカードのSwipeがこれに該当します。イベントとは、企業における広大な情報の海のどこかで発生する状態の変化なのです。

現代の企業は、世界のあらゆる場所にサプライヤー、ベンダー、顧客、従業員、契約社員、設備、そして製品を抱えることができます。そして、これらの様々なイベントソースのすべては、オンプレミスのデータセンター、または複数のパブリッククラウドやプライベートクラウド環境に存在するかも知れません。ここでイベントドリブンアーキテクチャーの登場となります。

イベントドリブンアーキテクチャーでは、特定のシステムやアプリケーションで発生したイベントが、他のシステムやアプリケーションに変更を加えるきっかけとなります。トピカにおける在庫の低下は、遠く離れた中国の工場へ即座に通知されます。これと同時に、営業担当者は携帯のプッシュ通知で在庫の欠品を知り、またAIベースの価格設定アプリケーションは価格設定の前提条件を変更し、さらにエグゼクティブのダッシュボードには変更可能な戦略案が提示されます。

イベントが発生する度にイベントドリブンアーキテクチャーが作動し、イベントがどこから来たかに関わらず、企業全体へ確実に伝達します。

イベントドリブンアーキテクチャーの心臓部とも言えるのが”イベントブローカー”で、どのアプリケーションがどのイベントに関心があるかを判断し、対象となるイベントを配信します。

イベントブローカーは交通整理を行う警官のような存在です。警官はすべての人が正しい方向に進むよう指示します。一方、イベントブローカーは、イベントの配信に特化したアーキテクチャー層です。

イベントブローカーは、システムそれぞれが関心を持つイベントを登録し、イベントデータの配信についての詳細を処理します。その結果、企業はどのアプリケーションがイベントを消費するかを直接意識することなく、イベントを生成するアプリケーションを追加することができます。これは、デカップリング(decoupling)と呼ばれる分離化の概念であり、前述の密結合(tight coupling)とは正反対のものです。

単独のイベントブローカーだけでも強力ですが、複数の場所に存在するイベントブローカーをつなぎ合わせた、いわゆるイベントメッシュを作成することで、その威力はさらに大きくなります。イベントメッシュを導入することで、世界中のアプリケーションは、ブローカーが存在する場所であればどこにでもキューを配置し、リアルタイムで、デカップリングされた形で情報を送受信することが可能となります。

結果として、イベントメッシュによって企業の内部、または企業を横断した形で、情報移動の効率化と加速化を図ることができます。

イベントの基礎

架空のタクシー会社が、SaaSプロバーダーであるBambooを使ってドライバーを新規に雇用するというシナリオを使って、イベントの基礎を説明することになります。なお、新規雇用に関わるダウンストリームのアプリケーションとして、Slack、Salesforce、ServiceNowの3つが存在するものと想定します。

イベント：企業のどこかで何かが変わる、ビジネス上の瞬間をイベントと言います。ここでは、タクシー会社でのドライバーの雇用がイベントに相当します。雇用イベントに関する情報は、イベントのペイロード（コンテンツ）に格納されます。たとえば、ドライバーの名前、社会保障番号 (SSN)、住所などがJSONまたはXMLフォーマットでペイロードに格納されます。

トピック：イベントのルーティングをより効率的に行うために、イベントの内容を簡単に説明する階層構造を持つ文字列です。トピックはメタ情報であり、ペイロードとは分離されています。

トピック文字列のフォーマットは、企業によって異なります。このメッセージに対するトピック文字列は、次のようになります：

```
/taxico/hr/bamboo/Driver/hired/v1/349034
```

この文字列は、以下のように分解できます：

taxico: 会社名

hr: 部門名

Bamboo: イベントを生成したアプリケーション

Driver: 何に関するイベントか（名詞）

Hired: 発生内容（動詞）

v1: イベントのバージョン

349034: ドライバーの社員ID

サブスクリプション： ダウンストリームのアプリケーションは、サブスクリプションを使って配信を受けたいイベントを示します。これらのサブスクリプションは、トピック文字列と同じ形式ですが、真に威力を発揮するのはワイルドカードを使用できる点です。"Slack側が雇用されたドライバー全員について知りたい場合には、次のような文字列でサブスクリプションを指定します： /taxico/hr/bamboo/Driver/hired/v1/*

* はすべてを意味するワイルドカードです。

もしSalesforce側が雇用、解雇または昇進があったドライバーについて知る必要があれば、次のサブスクリプションを使用することができます： /

```
taxico/hr/bamboo/Driver/*/v1/*
```

キュー： イベントブローカーは、受信イベントのトピック文字列とマッチするサブスクリプションを見つけると、その受信イベントのコピーを作成し、サブスクリプションを持つキューに配置します。ダウンストリームのアプリケーションが処理できるようになるまで、キューはメッセージを保持します。キューにはいくつかの特筆すべき点があります。キューは順序を保証し、アプリケーションに問題が発生した場合にはメッセージを保存し、さらに一定期間に大量のメッセージが発生してアプリケーションが処理できない場合にはバッファとして機能します。

一致するサブスクリプションが複数存在する場合、関連するすべてのキューがメッセージのコピーを取得するため、メッセージをトピックにパブリッシュしキューから取得するという方法は、1つのソースを元に関心のある複数のコンシューマへ情報を配信するための優れた方式となります。



イベントドリブンのインテグレーションによってリアルタイムの課題を解決

グローバルコネクティビティ：グローバルな同時配信

イベントメッシュによって、トピカから世界に向けてデータを瞬時に配信できるようになります。イベントブローカーは、アプリケーションがイベントを生成するとすぐに処理を行い、関連するクライアントの場所を特定して、接続しているイベントブローカーへイベントのコピーを配信します。

この通信は、アプリケーションやiPaaSがイベントの詳しい到達経路を知る必要なく実行されます。また、イベントに関心を持つアプリケーションがメッセージのコピーを取得し、それぞれ同時にメッセージを処理できるため、プロセス全体が高速化します。

データの移動をイベントブローカーに委任することで、次のような複数のメリットが得られます：

- 社内の各業務部門、外部のパートナー、そしてお客様との間でデータを民主化することができるようになります。
- オンプレミスのレガシーシステムと、クラウド環境の最新のアプリケーションを組み合わせ活用できるようになります。
- DevOpsチームが言語、プロトコル、そしてアプリケーションのロケーションを自由に選択できるようになります。

トラフィックの急増：バッファリングとキューイング

イベントブローカーは、IoTデバイスからiPaaSの機能にアクセスするための理想的な手段と言えます。

何百万台ものコネクテッドカー（サンパウロのシビックも含め）やその他のリアルタイムの情報ソースは、企業がお客様に関する新たなインサイトに対応する能力を向上させる、膨大な機会を提供します。

ただし、情報量は”諸刃の剣”となる可能性も秘めています。需要が高い時期にはトラフィックが急増する可能性があるからです。

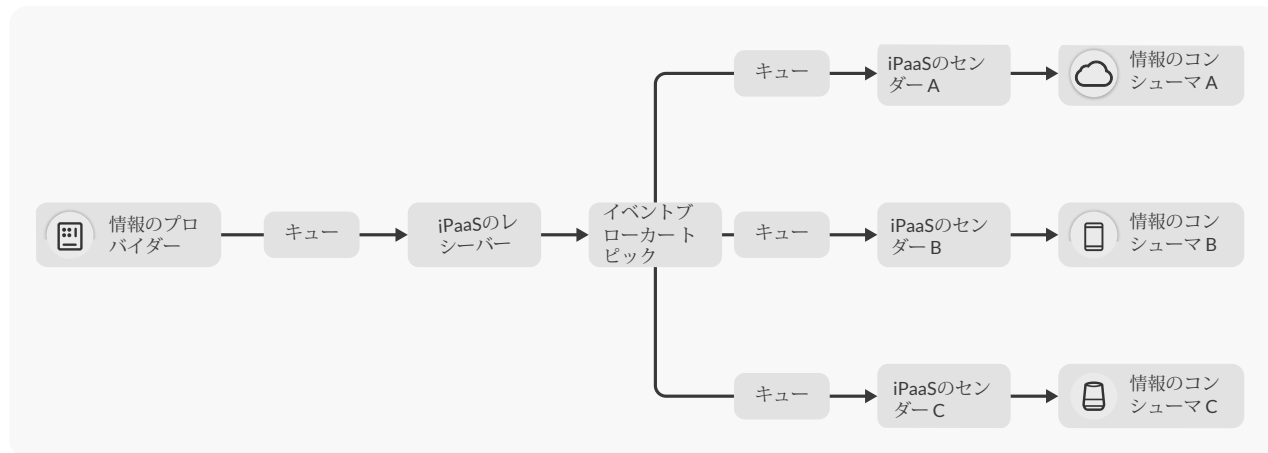
ただし、情報量は”諸刃の剣”となる可能性も秘めています。需要が高い時期にはトラフィックが急増する可能性があるからです。

イベントブローカーは、多くのIoT情報ソースがやり取りできるイベントのエントリーポイントを提供することで、これらの課題を克服しています。

また、メッセージングに特化したアプリケーションでは、エンタープライズグレードのイベントブローカーは大量の同時接続を処理することができます。

トラフィックによってiPaaSやダウンストリームのシステムが圧迫されないように、イベントブローカーに到達したイベントはバッファリングされます。イベントを直接iPaaSに引き渡すのではなく、キューに格納していきます。iPaaSが処理待ちの状態であれば、すぐに処理に入ることができます。トラフィックの急増によりiPaaSのリソースが不足している場合、イベントはiPaaSが処理できるまでその順番を待ちます。

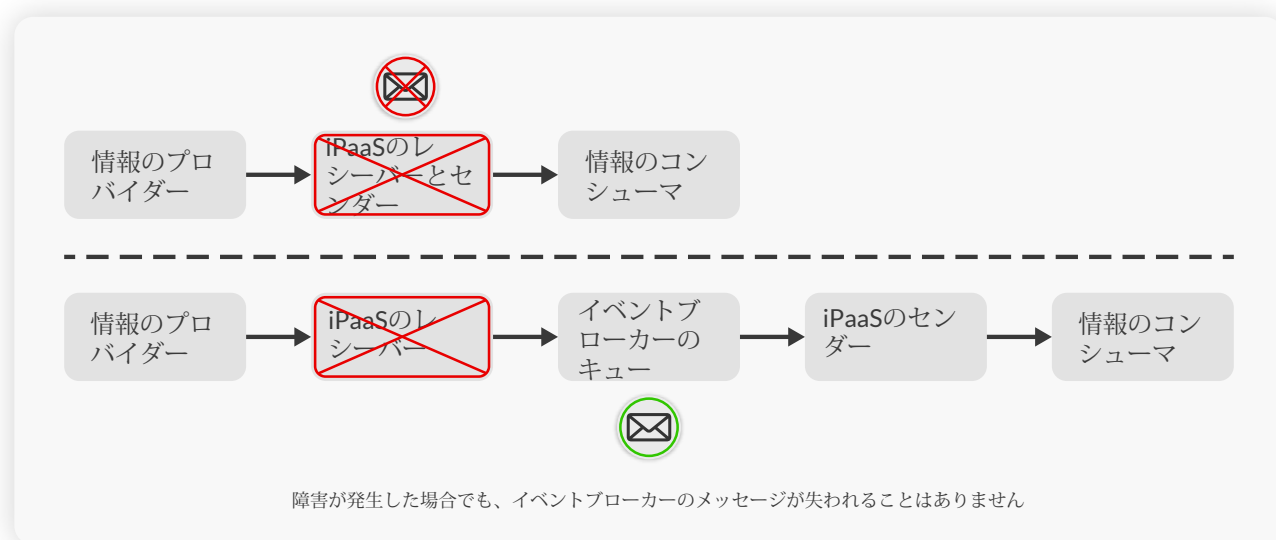
イベントブローカーは、アウトバウンド側にとって重要な存在となります。IoTデバイスの主要な特性の1つは、デバイスが常に利用できる状態にあるとは限らない点です。このため、デバイスがオンラインになるまでイベントを保持しておく必要があります。



エラー処理とディザスタリカバリ：情報の保存

企業内を流れる情報が飛躍的に増加する中で、たった1つのデータの喪失が致命的な（あるいはお金に直結した）問題につながる可能性があります。誰かの処方箋のデータを失ったり、潜在的な副作用を通知できなければ、危機的な結果を招いてしまう可能性があります。

分離（Decoupled）アーキテクチャーの一部としてイベントブローカーを導入することで、iPaaSのロジックをシンプルに保ちながら、重要な情報が失われないよう保持することができます。イベントブローカーは、iPaaSに到着したメッセージをキューまたはトピックに永続的に保存することで、iPaaS全体またはプロセスの一部が停止するような障害が発生した場合でも、カスタムロジックをコーディングする必要なく、メッセージが失われないことを保証できます。

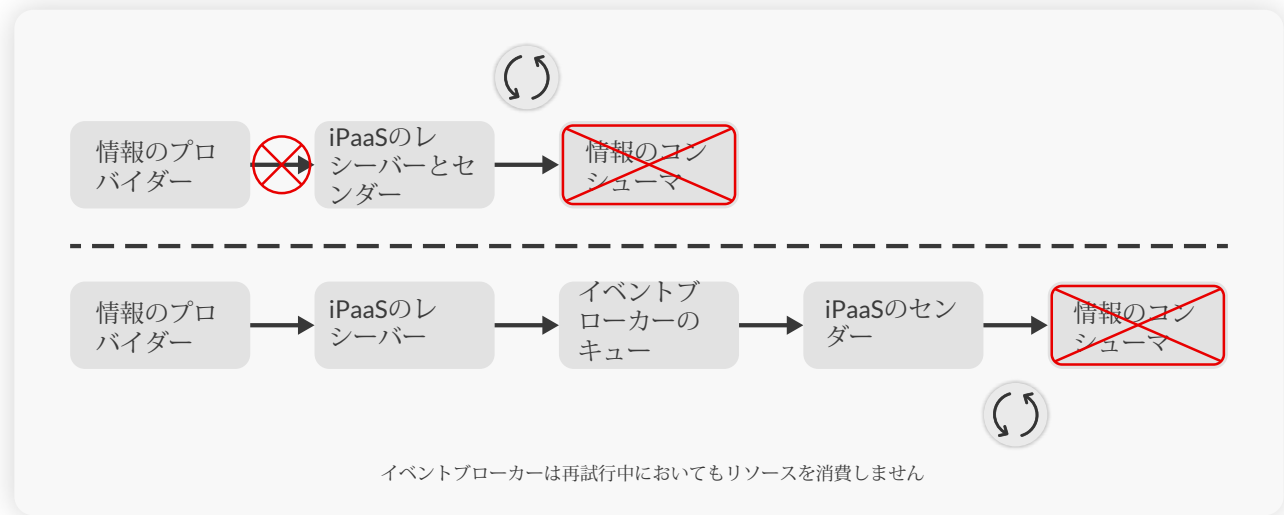


障害が発生した場合でも、イベントブローカーのメッセージが失われることはありません。メッセージがイベントブローカーに保存されると、iPaaSのレシーバーはメッセージが正常に処理されたことを情報のプロバイダーに通知し、次のメッセージを受け取ります。

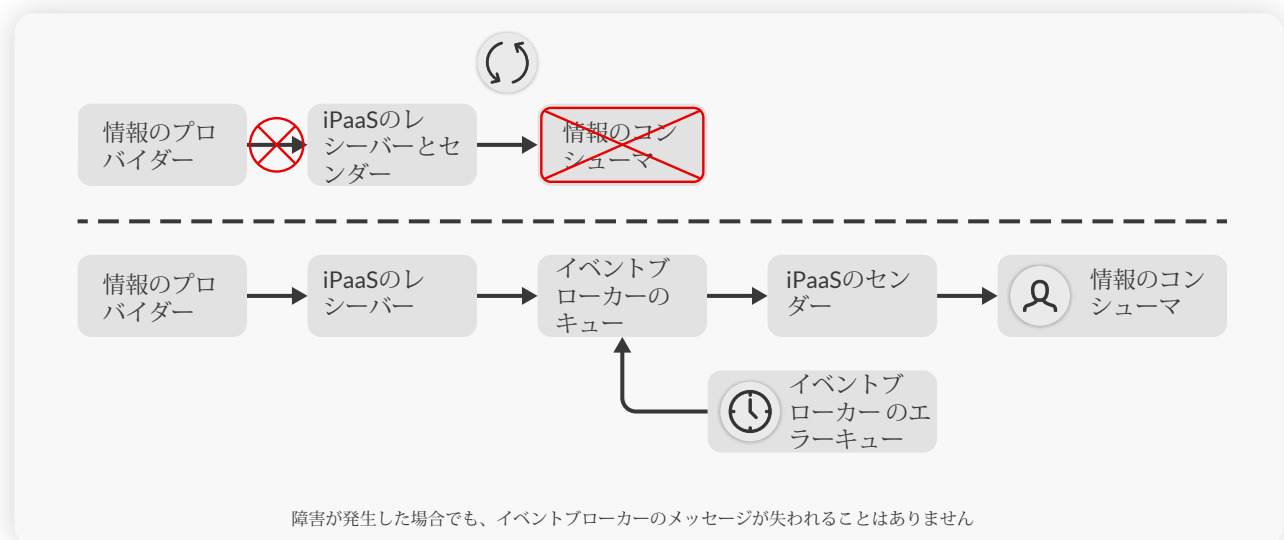
レシーバーがメッセージを迅速かつ他と独立して処理できるこの機能は、ダウンストリームに位置する情報のコンシューマが処理に対応できない場合に役立ちます。

一般的に、情報のコンシューマが利用できない場合、トランザクションが成功するまで何度も再試行が行われます。しかし、レシーバーとセNDERが同じスレッド上にいる場合、iPaaSのリソースがすぐに不足し、受信トランザクションをブロックする連鎖反応を引き起こす可能性があります。

イベントブローカーによるメッセージのバッファリングによって、ソリューションの継続した稼動が可能となります。



エラーの解決に時間を要する場合（アナリストがルックアップテーブルに値を追加する必要があるなど）、イベントブローカーはイベントをエラーキューに配置します。その後アナリストが問題を解決した場合、それらのメッセージは処理用のキューに移すことができます。



これらのシナリオはすべて、オンプレミスであれクラウドであれ、基礎となるイベントブローカーがデータセンター内で安全かつ健全であることを前提としています。しかし、それは必ずしも確かな仮定ではありません。災害は起こり得るものであり、そのような状況下でも業務を継続できなければなりません。

そのような（願わくば）まれな状況においても、明確に定義されたリカバリープランがすべての違いを生みます。最良のディザスタリカバリープランでは、主要コンポーネントのスタンバイ・インスタンスを地理的に異なる場所に設置します。これにより、例えばシカゴ近郊のデータセンターで火災が発生しても、フェニックス近郊のデータセンターで運用を継続することができます。

しかし、アプリケーション・コンポーネントを新しい場所で起動し、新しく作成されたイベントの処理を開始するだけでは十分ではありません。前述の処方箋の例のように、処理中の重要なデータは、新たな場所でも利用できる必要があります。イベントブローカーには、処理中の情報を保持するための高速で信頼性の高いレプリケーション技術が含まれており、アプリケーションは復旧が始まるとすぐにその情報を処理することができます。

新機能の追加：疎結合とアジリティ

ビジネスにおいて唯一断言できるのは、明日遭遇する課題は、今日直面している課題とは異なるということです。クレジットカード詐欺は時代遅れの犯罪行為だと思っけていても、また誰かがこのシステムを打ち破る方法を見つけ出します。

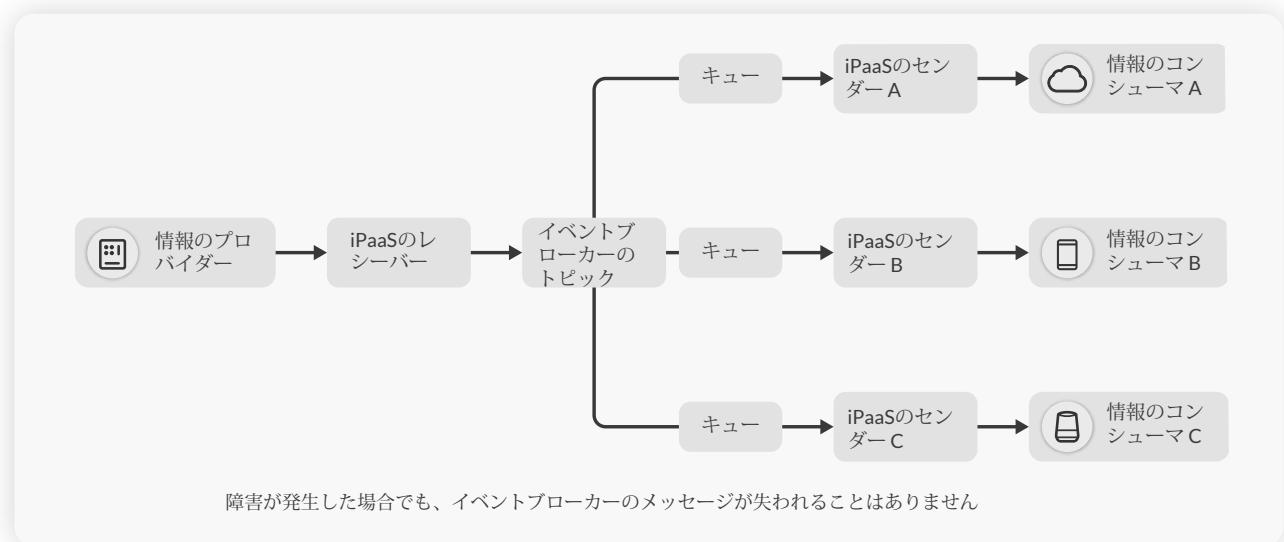
だからこそ、今、イノベーションが求められるのです。

ここまで、センターとレシーバーを相互に分離することで得られる優位性に注目してきましたが、分離すること自体にもメリットがあります。そして、その真価が実際に発揮されるのは、パブリッシュ/サブスクライブ (pub-sub) メカニズムを使用して、複数の多様な情報コンシューマにトランザクションの配信を開始する時です。

イベントブローカーは、セNDERとレシーバー間の仲介役となることで、セNDERを変更してもプロセス全体の再デプロイと再テストを必要とせず、パブリッ
シュ/サブスクライブの世界へ移行することを可能にします。

さらに、新しいコンシューマを既存の導入済みソリューションとシームレスに連
携させることもできます。

イベント・ブローカーでは、iPaaSのツールセットを使用して、iPaaSが提供する
レガシー・コネクティビティとグラフィカル・マッピングを組み込み、新しいセ
ンダーをデザインすることができます。そして新しいサブスクリプションを追加
すると、イベントが新しいプロセスに流れ始め、さらに情報のコンシューマへと
流れていきます。新しいセNDERは、既存のシステムとやり取りすることが可能
ですが、システム全体を再テストする必要がないため、イノベーションの展開を
大幅に加速することができます。



まとめ

イベントドリブンなインテグレーションは、情報の流れをリアルタイムへ加速することで、顧客により良いサービスを提供することを可能にし、IoTセンサー、モバイル・アプリケーション、ハイブリッド・マルチクラウドの採用、マイクロサービス、機械学習などの最先端技術に企業の枠を広げるのに役立ちます。

さらに、iPaaS/HIPとイベントドリブンなインテグレーションによって、インテグレーションの過程で発生する課題を単純化し、革新的なアイデアを容易に市場に投入できるようになります。

iPaaSおよびイベントドリブンインテグレーションが自社に最適だとお考えの場合、また、その先に何が待ち受けているのかを理解したいとお考えの場合には、「The Architect's Guide to Implementing Event-Driven Integration with iPaaS（アーキテクトのためのiPaaSとのイベントドリブンインテグレーション実装ガイド）」をご覧ください。本ガイドでは、企業をイベントドリブンに転換させるための技術的な手順を紹介しています。



Jesse Menning の紹介

Jesse Menningは、Solace CTO オフィスのアーキテクトです。

彼は、iPaaSなどのインテグレーションテクノロジーとSolaceを組み合わせ、イノベーションをより迅速に推進するという取り組みに重点を置いています。Solace入社以前は、主要な医療機関、大規模小売店、政府機関、中規模企業など、様々なクライアントでイベントドリブンインテグレーションソリューションの構築に携わってきました。



Ottawa | Toronto | New York | Chicago | Atlanta | Silicon Valley | London | Paris | Zurich
東京 | Seoul | Hong Kong | Shanghai | Singapore | Mumbai | New Delhi | Melbourne | Sydney



Solaceについて

Solaceは、企業がビジネスオペレーションや顧客とのやり取りをイベントドリブンにするために必要なすべてを提供し、モダンでリアルタイムな企業の実現を支援します。市場初で唯一のイベント管理プラットフォームであるPubSubにより、イベントの作成、文書化、発見、ストリーミングを、イベントが作成された場所から消費される必要がある場所まで、安全、確実、迅速かつ保証された形で配信する包括的な方法を提供します。詳しくは solace.com をご覧ください。

フォローする



お客様企業の一例



パートナー企業

