CTO向け eコマース アーキテクチャガイド

人材、プロセス、そしてテクノロジー

July 16, 2021

aws retail

注意事項

お客様は、本ドキュメントに記載された情報を独自に評価する責任を負います。本文書は、(a)情報提供のみを目的としたものであり、(b)現在のAWSの製品提供および実務を示すものであり、予告なく変更される場合があります。AWSの製品またはサービスは、明示または黙示を問わず、いかなる種類の保証、表明、または条件なしに「現状のまま」提供されます。AWSの顧客に対する責任および義務は、AWSの契約によって管理されるものであり、本文書はAWSと顧客との間のいかなる契約の一部でもなく、またそれを変更するものでもありません。

目次

概要

適切な e コマース・アーキテクチャを確立することは、企業の全体的なビジネス戦略をデジタル販売チャネルが確実にサポートするためには不可欠なものです。自社の e コマース・アーキテクチャを継続的に再評価し、ビジネス戦略の変更、新たな消費者動向、テクノロジーの進歩を反映していることを確認することが重要です。本書では、経営幹部が徹底的な評価を実施できるよう、トレードオフや教訓を交えながら、e コマース近代化の意思決定プロセスを説明します。また、組織の連携や、選択したアプローチを実現する方法についても触れています。また、本稿では Aman WebĀĀ Services（AWS）がこれらの目標の達成を支援する方法を紹介します。

イントロダクション

Salesforce Industry Insights によると、世界のeコマース収益は2020年第2四半期から2021年第1四半期にかけて平均 62.5%（前年同期比）増加しました。Covid-19 の大流行は eコマースの普及を加速させ、多くの潜在的なビジネスチャンスを生み出しました。世界の小売売上高に占める e コマースの割合はわずか 18% であり、大きな成長の可能性を秘めています。

ほぼすべての業界の CEO が CTO や CIO に eコマース機能の提供や改善を求めています。また、増え続ける販売チャネルをサポートし、接続し利用するためでもあります。この業界は、初歩的なウェブサイトから始まり、スマートフォン向けに最適化されたモバイル機能を追加し、携帯電話やタブレット向けのネイティブアプリケーションやハイブリッドアプリケーションへと進歩してきました。現在では、デジタル販売チャネルは、物理的な販売チャネルやマーケットプレイスとシームレスに統合する必要があります。新たなトレンドとしては、ソーシャルコマースや、特にライブストリーミング eコマースが挙げられます。ユーザーは、デジタル市場をリードする企業が提供する、使いやすさ、信頼性の高いサービス、サイトの応答性、競争力のある価格設定など、一流のデジタル体験に慣れ親しんでいます。

CTO は、このような期待と新たなトレンドを認識し、以下のコア要件を満たす e コマースアーキテクチャと組織を提供する必要があります：

- **ビジネスバリュー**を産み出すこと
- インダストリのトレンドに**素早く追随**すること
- あらゆるマーケットプレイスやプラットフォームと**簡単に統合できる**こと
- パフォーマンス、可用性、拡張性、保守性など、eコマース・システムの**中核となる非機能要件を満たす**こと

先進企業と後発企業を分ける主な差別化要因は、規模に応じた最適化、実験、革新、そしてオンラインとオフラインのショッピング体験のシームレスな統合です。

"収益成長率が上位10%に入る企業の50%以上は、アイデアをテストし、結果を測定し、製品、サービス、働き方の変更を実行することにおいて、同業他社よりも効果的です。"

[3] mckinsey.com

eコマースアーキテクチャにおける4つのファンダメンタル

最高技術責任者（CTO）から話を伺った際、最も言及されているトピックは eコマース・アーキテクチャの中核となる以下の課題です：

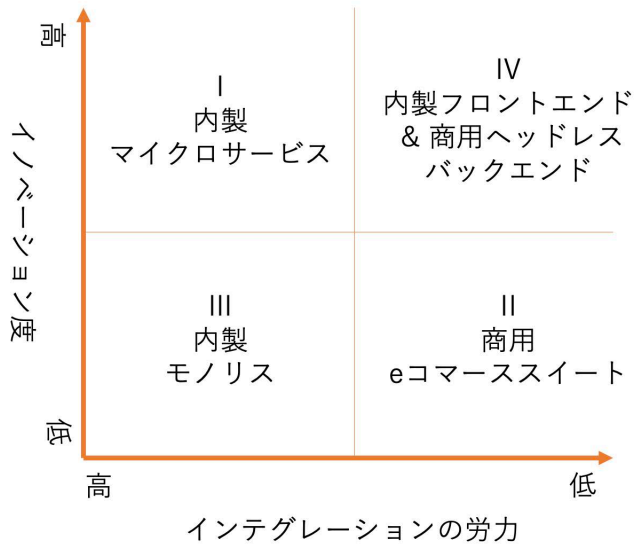
- 販売チャネルをサポートするデジタルおよび物理的なフロントエンドにおける**イノベーションのスピード**
- 増え続ける他のITシステムを**統合**する労力

イノベーションを重視するCTOは通常、社内に開発チームを抱えています。また、収益やその他の指標を重視し、スピードに最適化し、迅速に実験を行います。彼らはデータと柔軟性に対する要求が高く、eコマースはビジネスに不可欠な要素です。ソフトウェア開発は組織の一部であり、たとえ追加能力として外部の労働力を利用するとしても。ソフトウェアを構築する方法を知っているため、「作るか、買うか」の意思決定が容易で、「作る」傾向が強いです。

一方、異なるシステムの統合を重視し、外部ベンダーから最善のソリューションを調達することを好む CTO もいます。その場合、コスト削減と取り組みの最適化に重点を置いています。サプライチェーン・マネジメントやエンタープライズ・リソース・プランニングのようなバックエンド・システムに容易に統合できる e コマース機能を購入します。彼らは通常、実験を行いません。eコマースはコアビジネスではありません。単なるウェブショップであり、アプリがないこともあります。実質的なソフトウェア開発能力はなく、必要に応じて代理店やコンサルティングパートナーを利用します。ビルダーとしてのスキルがないため、買うか作ってもらうかという傾向が強いのです。

イノベーションと統合に関して、次ページに示ようなマトリックスにすると、以下の4種類の eコマース・アーキテクチャを簡単にマッピングすることができます：

- 内製マイクロサービス・アーキテクチャ
- 商用の(COTS)eコマース・スイート
- 内製モノリシック・アーキテクチャ
- 内製フロントエンドと商用ヘッドレスバックエンド



内製マイクロサービス アーキテクチャ

マイクロサービス、自己完結型システム、サーバーレスアプローチに従ったカスタムビルドの eコマースアプリケーションは、通常、ソフトウェアエンジニアリングチームによって作成され、強力なビルダーメンタリティを持つ企業に多く見られます。このアーキテクチャは、**スピード、アジリティ、イノベーションのために最適化されています**。多くの場合、多様なフロントエンドを持ち、容易に利用可能なコネクタを持つCOTSと比較して、バックエンドシステム（CRM、ERP、財務、請求、レポート）との統合に多くの労力を要します。顧客と接するウェブページやアプリの画面は、それぞれ独立して構築、保守、デプロイできる小さなマイクロサービスと相互作用します。

これらのシステムは通常、顧客のコンバージョンを最大化するために、直感的なユーザーインターフェイスで高度に最適化されています。柔軟性の高いマイクロサービス・アーキテクチャと、アジャイルな構築-測定-学習の考え方により、既存の機能を改善し、新機能を展開するための頻繁な実験が奨励されています。

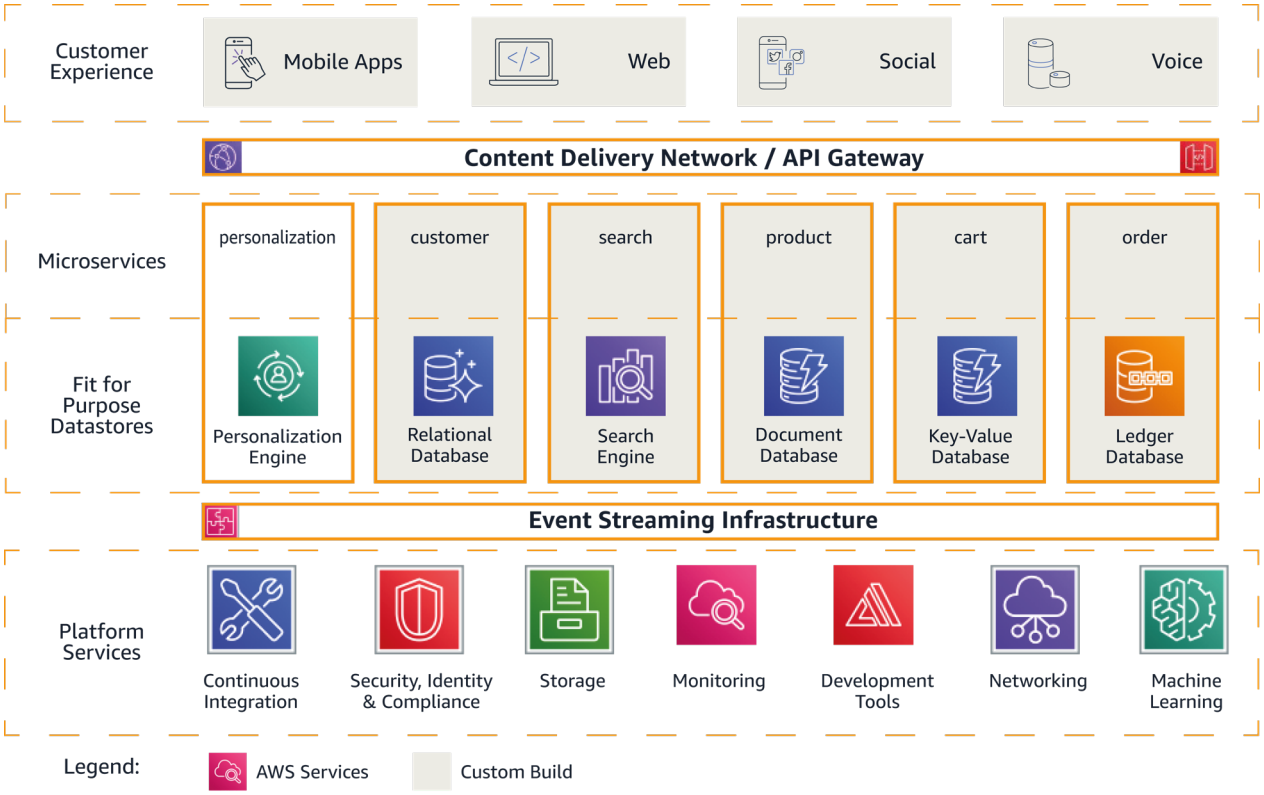
商用ソフトウェアは、一部の特殊なビジネス機能（商品検索、決済、レコメンデーションなど）に使用される場合があります。購入するかどうかの意思決定は、より詳細なレベルで行われます。**マイクロサービス・アーキテクチャは、クラウド・サービスを容易に統合して**、柔軟性、安全性、コスト効率、弾力性、グローバルな可用性を高めることができます。これらのアーキテクチャは、コンピューティング、ストレージ、モニタリング、バックアップなどのクラウドサービスだけでなく、アナリティクス、人工知能（AI）、機械学習、ユーザー管理、レコメンデーションなどのより複雑なサービスも統合します。柔軟なアーキテクチャにより、あらゆるレベルでコストの最適化が可能になります。

ビルダー・イノベーションを重視する組織には、ビジネス・センスだけでなく、強力なソフトウェア・エンジニアリングとオペレーションのスキルを持つ人材が必要です。

ビルド＆ランモデルを採用し、エンドツーエンドのオーナーシップのメリットを活用したいと考えている組織は、このアーキテクチャの恩恵を受けることができます。チームはビジネス能力に基づいて編成され、自らの運命を握り、自らのペースで動くことができます。迅速なデリバリーを促進するために、機能別チームは、セキュリティ、モニタリング、継続的インテグレーションなどの基本的な横断的関心事に対するプラットフォームのサポートを必要とします。

しかし、カスタムメイドのマイクロサービス・アーキテクチャ・アプローチのリスクは、**複雑性の増大**です。組織は、エンジニアリングとオペレーショナル・エクセレンスに高いハードルを設定することで、このリスクを軽減することができます。迅速な変化と継続的な学習を受け入れる文化が必要です。 Otto, Zalando, Adidas, Nike, Dunelm, Fast Retailing はいずれもカスタムメイドのマイクロサービスアーキテクチャの実装に成功しています。

技術的な視点



このアーキテクチャの主な特徴は、**コンポーネント間の高度なデカップリング**です。フロントエンドはバックエンドのサービスから独立しています。これらはシングルページ、モバイルアプリケーション、あるいはチャットや音声インターフェースです。

消費者に対してフレンドリーなAPIを綺麗に定義することから始めるのが、デカップリングを実現する最善の方法です。バックエンドでは、信頼性、可用性、効率を向上させるために、可能な限りイベント駆動型のアプローチを使って、サービスを非同期で通信させる戦略を採用します。

このアーキテクチャでは、バインドされた異なるコンテキスト間で明確な分離を定義します。それぞれのコンテキストが、モジュール化されたマイクロサービスアーキテクチャを使用して実装されています。すべての概念がAPI 定義によって明快に表現されることにより、バックエンドは、モバイルアプリケーション、店舗内のデジタルタッチポイント、ソーシャルチャンネルなど複数のフロントエンドに対してサービスを提供できます。

各バックエンドサービスは、独立してスケール、デプロイ、開発することができます。API定義が安定している限り、チームは、1つまたは複数のサービスを所有し、他のチームと密に連携することなく独立してライフサイクルを管理できます。

さらに下流では、各サービスの機能は目的に合った専用のマネージド・サービスを使用します。例えば、Amazon OpenSearch Service は検索に、Amazon Personalize は商品のレコメンデーションに使用できます。

このアプローチでは、API 定義に変更がなければ、要件や消費者の透明性のニーズの変化に応じて、新しいエクスペリエンスを作成およびデプロイし、ロギングシステムやデータストアなどを進化させることが容易です。

クラウド・サービスのプロビジョニングとカスタムメイドのマイクロサービスのデプロイは完全に自動化されています。このようにして、あらゆる種類のアップデートが迅速かつ個別に実行され、エラーや障害、リリースによる影響範囲が最小化されます。

このアプローチの課題は、マイクロサービス・アーキテクチャと分散システムから受け継いだものです。システム全体の複雑さは、独立したサービスの数とともに増大します。したがって、**複雑さを管理するには、自動化、ロギング、モニタリング、および分散トレースへの一貫したアプローチが必要**です。AWSのマネージドサービスはシームレスに統合され、迅速な開発ライフサイクルをサポートするため、これらの懸念にうまく対処できます。

商用の (COTS) eコマース・スイート

e コマースアプリケーションは、SAP、Adobe、Oracle、Salesforce などの COTS (Commercial Off-the-Shelf) ベンダーからライセンスを取得します。このアーキテクチャは、CRM、ERP、財務などの**バックエンドシステムとの統合が容易**で、すぐに最大限の機能を提供できるように設計されています。アプリケーション・スイートは多くのモジュールで構成され、初期導入は容易ですが、カスタマイズが必要になる場合もあります。このアーキテクチャを採用する利点は、ロールアウト前の段階でのカスタマイズ作業が最小限に抑えられるため、**市場投入までのスピードが速い**ことです。多くの機能がすぐに利用できるため、**開発スタッフの数も少なくて済みます**。eコマース・アプリケーションを組織の要件に適合させるための慎重な製品選択プロセスにより、IT部門とビジネス・ユーザーの両方に対する全体的な変更管理の影響は最小限に抑えられます。最後に、十分にサポートされたエンタープライズ ITエコシステムがあれば、バックエンドやサードパーティシステムとの統合は、展開と保守が容易になります。

しかし、このようなスイート上でイノベーションを起こすのは困難です。顧客は、ベンダーが機能を実装するのを待つか、顧客自身で構築しなければなりませんが、社内に適切なスキルがない場合は困難です。商用ソフトウェアを使用する組織は、ソフトウェア・ベンダー、ベンダー社内のプロフェッショナル・サービス組織、またはサードパーティのコンサルタントに依存することが多くなります。コア・システムにカスタム機能が追加されると、アップグレードが難しくなります。アップグレードを適用しない企業もあり、高メンテナンスのシナリオに陥ります。具体的なメリットがないのにメジャーアップデートを余儀なくされることもあります。

これらの課題の結果として、実験、構築、測定、学習、最適化が広く採用されず、KPIや顧客のニーズに基づいて最適化を行わないことになります。ベンダーのシステムが対応していないため、専門的なビジネス機能の追加や交換も困難です。ソリューションのメンテナンスコストは、通常、カスタマイズのレベルよりも大きな割合で長期的に増大します。商用ソフトウェアのデータは、アクセスしにくく、解釈や関連付けが容易でないことがよくあります。

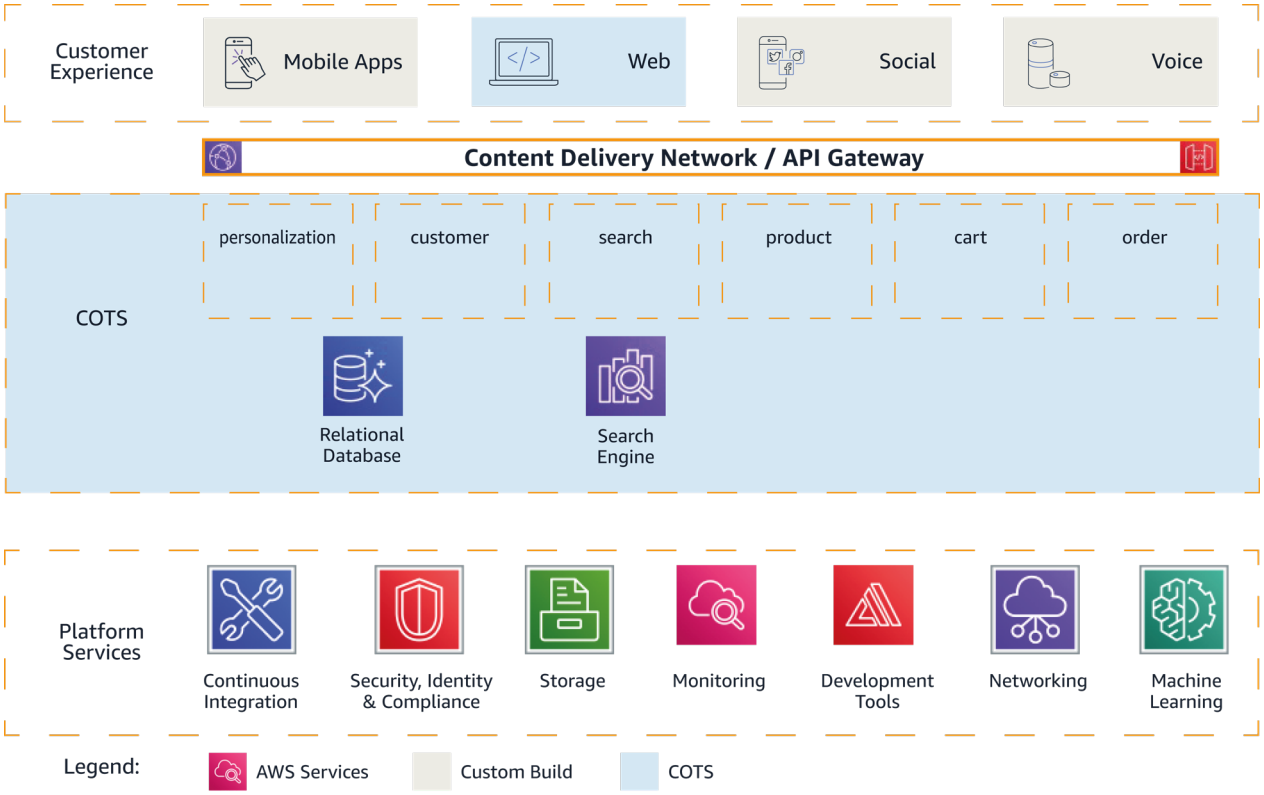
商用ソフトウェアのライセンスコストは簡単に計算できますが、運用レベルで最適化するのは困難です。**商用ソフトウェアのアーキテクチャは、規模が大きくなるにつれ非常に高価になります**。最も一般的なコスト要因は、季節的なピークに対応するためのコンピュート・リソースとストレージの過剰なプロビジョニング、ライセンス・コスト、サード・パーティのコンサルティング・サービスの料金です。しかし、アプリケーションをクラウドに移行すれば、総所有コストの削減、セキュリティの向上、弾力性、俊敏性の向上につながる可能性が高くなり、コア・インフラ・コンポーネントのモダナイゼーションが可能です。ソリューションは、典型的なユースケースと機能を十分にカバーしていますが、どのカテゴリーにおいても**ベスト・オブ・ブリードの内製ソリューションのパフォーマンスには及びません**。

組織には、商用スイートに精通した従業員がいなければなりません。このようなスキルや経験を持つ人材の採用は困難な場合が多いため、組織は適切な人材を見つけるために、高価な外部のコンサルタントや人材紹介会社に頼らざるを得ない場合があります。能力やノウハウが限られているため、**商用ソリューションの潜在能力をフルに発揮できない**ことがよくあります。

しかし、組織がこれらのシステムをクラウドに移行し、長期的な戦略として、eコマース・アーキテクチャをイノベーションに焦点を当てた上位象限の1つに移行すれば、人材を引き付けやすくなります。技術スタックのモダナイゼーションは、イノベーションカルチャーを導入するための重要なステップです。

英国第2位のスーパーマーケットチェーンであるSainsbury'sは、AWSへの移行に成功し、70〜80%のパフォーマンス向上を達成しました。同社は年間5、6回のリリースを1日に複数回行うようになり、インフラ利用率を30%削減しました。

技術的な視点



商用プラットフォームは、セルフホスト型か、SaaS（Software-as-a-Service）モデルでプラットフォーム・ベンダーが管理します。どちらの場合もクラウドを利用します。まず、共通項をいくつか調べてみましょう。

顧客の需要に迅速に対応するためには、より迅速なリリース・サイクルをサポートする継続的インテグレーション（CI）プラットフォームを持つことが有益です。自動化をサポートする基盤を構築するCOTSベンダーが増えています。構成は、APIを呼び出したり、CLI (Command-Line Interface) を使用したり、新しい完全なシステム構成をアップロードすることによって変更されるかもしれません。

内製ソフトウェアから得られるのと同じく、 継続的インテグレーションの利点は、より速くアップデートを提供すること、より早くバグを見つけること、開発者の生産性を向上させることなどであり、内製と関連性があり、ベストプラクティスが適用可能です。AWSは、新しいバージョンのビルド、パッケージ化、自動テストを目的としたサービスの組み合わせを提供します。

商用プラットフォームには、製品カタログ、製品価格、マーケティング・コンテンツ、プロモーション、在庫情報などのデータが必要です。このようなデータのほとんどは、利用可能なインターフェイスを介して手動で設定できますが、ソースシステムからのデータフローを自動化することで、エラーを排除し、効率化を図ることができます。リアルタイムまたはバッチのデータ転送に関わらず、AWS はいくつかのアプリケーション統合サービスを提供し、分離されたコンポーネント間の通信を簡素化します。

e コマース業務から得られるメトリクスとアナリティクスは、構築-測定-学習のサイクルを実現する重要な要素です。データは基本的なビジネス上の意思決定を促し、新機能の展開の優先順位付けに役立ちます。アナリティクス機能は、e コマースイニシアチブのあらゆる側面を一枚のガラスで見ることができ、コンバージョン率、平均注文サイズ、訪問者 1 人当たりの収益、サイトトラフィック、直帰率などの主要なパフォーマンス指標の算出に役立ちます。
商用プラットフォームにデータ分析機能が組み込まれている場合もありますが、顧客情報、在庫レベル、サプライチェーンデータ、ソーシャルメディアの感情分析、サードパーティのマーケットプレイスデータなど、社内外の追加データセグメントを評価することは有益です。

セルフホスト型の商用プラットフォームの場合、クラウドへの移行には多くのメリットがあります。モノリシックな eコマース・アプリケーションであっても、1つ以上のデータベース・エンジンと複数のコンピューティング・リソースが必要になります。オンプレミスのホスティング・プラットフォームでは、日次や季節の負荷急増に対応できるキャパシティが必要ですが、クラウドでは、自動的に弾力的にオンデマンドで拡張できるオプションがあるため、スケーラビリティが根本的に変わります。マネージドなクラウド・ネイティブのデータベースは、ディザスタリカバリのための自動バックアップとポイント・インタイム・リストアがすぐに利用できる、メンテナンスフリーのデータベース・オプションです。これらのうちいくつかの商用データベースにおいては、データベースが完全にサポートされるためにライセンスを別途取得する必要がある場合があります。
複数の国や大陸にまたがって事業を展開する企業のために、AWS はグローバルに分散した e コマース・アプリケーションをサポートするサービスを提供しています。テクノロジーの選択次第では、いくつかのサービスは、さらに統合とメンテナンスを簡素化する、組み込みのクロスリージョン機能を備えています。

リージョン内およびリージョン間で高可用性を実現できます。AWSは、個々のサーバー、ストレージデバイス、またはデータセンターが停止した場合に備えて、ビジネスの継続性を確保するためのさまざまなサービスを提供しています。

通常、商用 eコマース・アプリケーションの**サーバー側のパフォーマンスを最適化するのは困難**です。バックエンドのスケーリングを決定するには、コンピュート、ストレージ、データベースのいずれのレベルであっても、インフラストラクチャの利用状況に関するデータインサイトが必要です。一方、クライアント側のパフォーマンスは、Amazon Cloudfront のようなコンテンツ・デリバリー・ネットワーク（CDN）で改善できます。

eコマースの商用プラットフォームをクラウドに移行した企業は、自動スケーリングを実装したり、クラウド・マネージド・サービスを使用して機能を拡張したりすることで、コスト効率とパフォーマンスを最適に組み合わせながら、スタックの各レイヤーを適切にサイズ調整するアーキテクチャの進化が容易であることに気づきました。消費者の体験をより迅速に向上させたい企業は、同じ商用 e コマース・バックエンドを使用しながら、カスタム・フロントエンドを構築しています。商用プラットフォームの API の成熟度によっては、基盤となるプラットフォームの複雑さを隠す、無駄のないマイクロサービス・レイヤーを実装する企業もあります。また、モダナイゼーションの最初のステップとして、バックエンドの一部を再実装または置き換えて、エンドツーエンドでの機能群を移行する企業もあります。

内製モノリシックアーキテクチャ

このアーキテクチャは、**もはや企業の第一選択肢としては主流ではありません**。しかし、多くの大企業はまだこのようなeコマースシステムを運用しています。

このアーキテクチャでは、ソフトウェアエンジニアリングチームは、伝統的なレイヤードアーキテクチャパターンを使用してカスタムeコマースアプリケーションを作成します。このアーキテクチャを使用している組織は、通常、初期のｅコマース導入企業です。彼らはもともとイノベーションに重点を置いていたり、商用ｅコマース・スイートを使用するよりも、独自のアプリケーションを構築することを好んでいました。

モノリシックアーキテクチャには多くの欠点があります。複雑になり、メンテナンスが大変です。変更は下流の副作用を引き起こし、開発とテストの労力を増加させます。新しいバックエンドシステムを統合したり、新しいフロントエンドチャンネルを適応させるのは、通常、時間がかかり、エラーが発生しがちです。アプリケーション全体がカスタムであるため、企業は実験と革新を行うことができますが、モノリシックなアーキテクチャでは速いペースで反復することが難しいため、その程度は非常に限定的です。

モノリシック・アプリケーション特有の複雑さにもかかわらず、コストを最適化することは可能です。しかし、それには多くの努力が必要です。主なコスト要因は、高価な商用データベースとオンプレミスのデータセンター・インフラになりがちです。

モノリシック・アーキテクチャは、他のアーキテクチャに比べてメンテナンスと管理が最も必要なため、企業にはソフトウェア・エンジニアリングと運用のスキルが必要です。その結果、エンジニアリング・チームにはイノベーションを起こす時間がほとんどなく、ビジネス・ステークホルダーとの対立がしばしば生じます。レガシーなモノリシックｅコマース・アーキテクチャを持つ企業は通常、ビジネスおよび技術系の人材を採用するのが困難です。

短期的にはクラウドでeコマース・アーキテクチャをモダナイズし、その後、イノベーションを重視する上位の象限のいずれかに移行することをお勧めします。

技術的な視点

セルフホスト型の商用 e コマースプラットフォームの技術的な原則と懸念事項は、カスタムビルドのモノリスにも当てはまります。大きな違いは、既存のエンジニアリングチームが、カスタムモノリシックアプリケーションの機能をより迅速に最新のアーキテクチャに進化させることができる点です。

モノリシックアーキテクチャの近代化が目標である場合、AWS は最優先事項に対処するサービスを提供します。たとえば、Amazon ElastiCache のようなキャッシュサービスは、バックエンドの重いデータベース操作をオフロードしてサイトパフォーマンスを向上させます。商品のレコメンデーションやその他のパーソナライズ技術などの新機能は、Amazon Personalize を使用することで、機械学習の専門知識がなくても導入できます。

AWS Marketplace から利用可能なAWSアプリケーションパフォーマンスモニタリングソリューションは、アプリケーションの使用状況に関する深いレベルの洞察と理解を提供します。ビジネス運用の洞察と全体的な戦略とともに、これらのアドオン・ソリューションは、新しいターゲット・アーキテクチャを定義するための貴重な洞察を提供します。また、ペインポイントや新機能の優先順位付けにも役立ちます。

最初のステップとして、デプロイとテストを自動化することで、企業はモノリシック・アーキテクチャを進化させる道筋をつけることができます。安定したリグレッション・テストを実施することで、企業は特定の機能を外部化して置き換えることができ、よりスムーズな移行が可能になります。一方、フロントエンドにAPIコントラクトを導入することで、機能の分解が簡素化され、新たな境界コンテキストでの成長が容易になります。

内製フロントエンドと商用ヘッドレスバックエンド

このアーキテクチャは、すでに説明した両方のアーキテクチャの長所を約束するものです。しかし、いくつかの欠点もあります。このアーキテクチャでは、フロントエンド（ウェブ、モバイル、ソーシャル、ビデオ、実店舗）をソフトウェア・エンジニアリング・チームが作成し、eコマースのバックエンドはSAP, Spryker, Elastic Path, Salesforceなどのベンダーが提供する商用ソリューションです。これらのベンダーは、eコマースのビジネスロジックがバックエンドのマイクロサービス層にカプセル化されるヘッドレスアーキテクチャアプローチを使用しています。APIを使用して、このバックエンドマイクロサービスレイヤーは、同じビジネスロジック（商品カタログと検索、ユーザー管理、在庫）で異なるフロントエンドをサポートします。ヘッドレスバックエンドは多くのモジュールと機能で構成されており、実装が容易で、カスタマイズが必要な場合もあります。

このソリューションは、**さまざまな販売チャネルをサポートするさまざまなフロントエンドで、あらかじめ定義された標準機能セットを使って、迅速なイノベーションができるように最適化**されています。また、このソリューションは、CRM、ERP、財務、請求、レポートなどのバックエンドシステムと簡単に統合できます。

フロントエンドでの技術革新は容易ですが、バックエンドでの技術革新は困難です。しかし、商用ソリューションの中には、顧客自身で新しいバックエンド機能を構築できるものもあります。

企業は Amazon AppFlowを使用して、商用システムの周囲に最新のイベント駆動型アーキテクチャを構築できます。Amazon AppFlowは、これらのサードパーティシステムからイベントを転送し、カスタムビルドサービスのバックボーンであるクラウドベースのイベントバスにプッシュします。

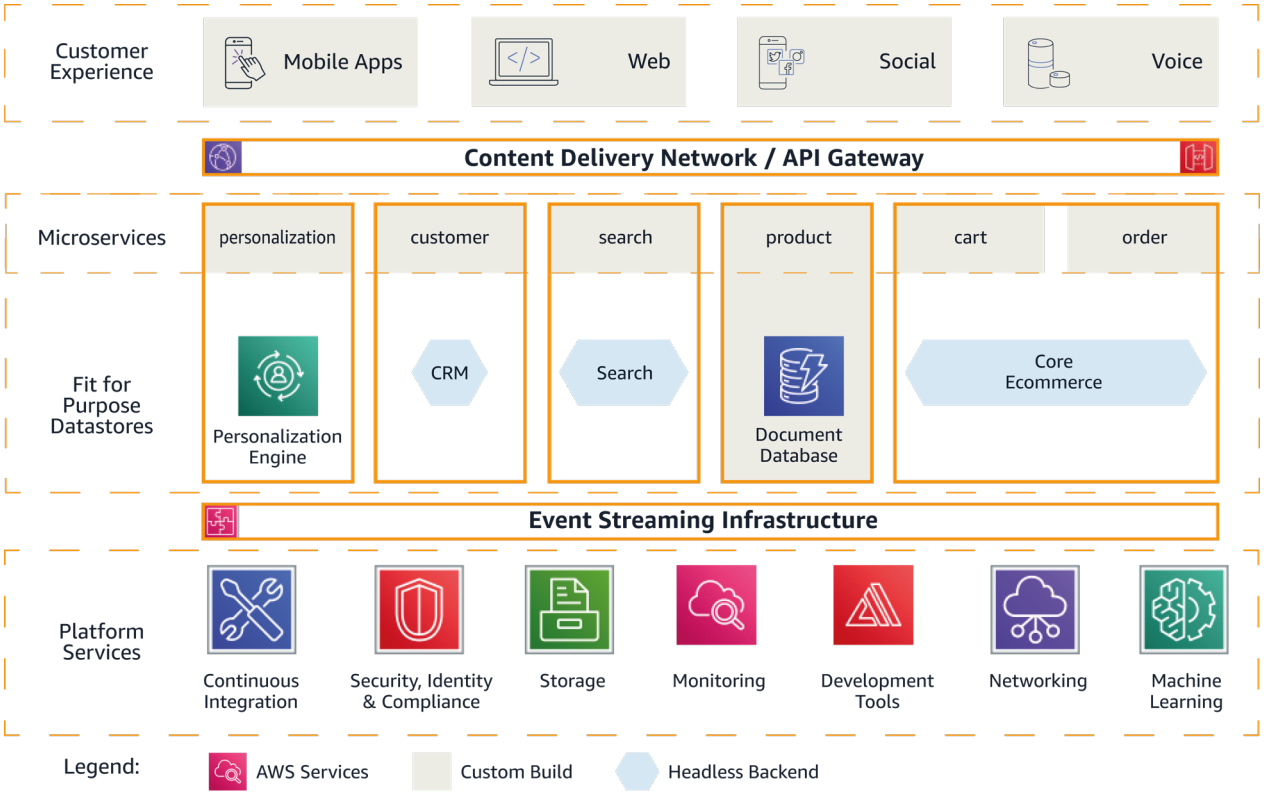
企業は部品レベルで作るか買うかを判断できます。商用ベンダーの制約にもよりますが、クラウドサービスを利用することも可能です。バックエンドが複数のシステムで構成されている場合、ある程度の統合とオーケストレーションの労力が必要です。

これらのシステムは、柔軟性の高いフロントエンド・アーキテクチャを採用しているため、ユーザー・インターフェースやユーザビリティの観点から高度に最適化されていることがよくあります。バックエンドのサービスに触れる実験や最適化はより難しく、時間がかかるため、全体的なコンバージョン率の最適化には限界があります。社内にリソースがない場合、商用ソリューションの導入や最新のフロントエンドの開発・導入を支援する専門のデジタルエージェンシーがあります。

コストの最適化はフロントエンドでは簡単ですが、商用バックエンドでは困難です。商用eコマース・スイートと同様に、このアーキテクチャは**規模が大きくなると高価になる**可能性があります。コストの最適化は通常、商用ベンダーとの契約交渉やクラウドインフラの利用によってのみ可能です。商用システムのデータはアクセスしにくく、解釈や関連付けが容易でないことがよくあります。実際のユースケースに具体的なメリットがないメジャーアップデートを強いられることもあります。**企業は、フロントエンドではソフトウェア・エンジニアリングと運用のスキルを、バックエンドでは商用システムの管理者を必要とします**。

この種のアーキテクチャを採用する組織では、フロントエンドとシステム全体を管理するための強力なソフトウェア開発スキルが必要です。また、商用スイートに精通した従業員も必要です。このようなスキルと経験を持つ人材の採用は困難な場合が多いため、企業は高価なサードパーティのコンサルタントに依存することになります。**能力やノウハウが限られているため、商用ソリューションの潜在能力をフルに発揮できないことが多い**のです。

技術的な視点



バックエンドからフロントエンドを開発することで、企業はさまざまなテクノロジーを選択できる最大限の柔軟性と俊敏性を手に入れることができます。さらに、マイクロフロントエンドアプローチは、分離された最新のウェブ機能により、企業の自主性とスピードをさらに高めます。

開発者は商用ヘッドレスバックエンドで利用可能なAPIレイヤを通してフロントエンドをバックエンドに接続することができます。しかし、商用ベンダーのAPIは冗長すぎたり、非常に複雑であったり、あなたの組織とは異なるAPIアプローチに従っているかもしれません。したがって、商用バックエンドのプロキシまたはファサードとして、無駄のないマイクロサービス・レイヤーを導入することは有益です。

APIプラットフォームの利用者は、学習曲線と採用をスピードアップするための一貫した経験を持つことになります。各マイクロサービスは、商品、顧客、検索などのビジネスエンティティや機能を中心に構成されるべきです。このアプローチはまた、不要なバックエンドの複雑さを隠すので、企業はAPI契約を簡素化するために複数のバックエンドエンドエンドポイントを集約してオーケストレーションすることができ、キャッシング、フォールバック、レートスロットリングなどの追加機能を備えたモニタリングのための中央レイヤーを提供することができます。

最も一般的な機能を備えたターンキーパッケージから、特定のeコマース機能のためのポイントソリューションまで、幅広い商用ヘッドレスeコマース製品があります。ポイントeコマースソリューションを選択した企業は、エンドツーエンドの一貫したエクスペリエンスを提供するための追加製品が必要になります。この場合、企業は一貫したAPI定義を保証するためにカスタムビルドのAPIレイヤーを使用する必要があります。

このアーキテクチャでは、横断的な懸念事項が非常に重要になります。多くのシステムが関係するため、企業は高度なセキュリティ、可用性、パフォーマンスを確保するためのさらなる課題を抱えることになります。異なるシステム間の自動テストを含む継続的インテグレーションの実践も、おそらく課題を提起するでしょう。これらは、包括的なeコマース・ソリューションを構築するために複数の製品を選択し、採用する際の重要な考慮事項です。

データの一貫性は、同じビジネスエンティティ（例えば、記事や顧客）を共有するプラットフォームを組み合わせる場合のもう一つの課題です。例えば、検索エンジンと別個の製品カタログは同じ製品データを共有します。レコードシステムでデータが変更された場合、企業はパブ/サブシステムを使用して、カタログと検索アプリケーションの両方にメッセージを送信することができます。

もしこれらのシステムが2つの異なるシステムであり、一方がリアルタイムに近く、もう一方がそうでない場合、データに一貫性がなくなり、矛盾を補うためにスタック内でいくつかの回避策が必要になり、カスタマーエクスペリエンスに悪影響を及ぼす可能性があります。

理想的には、異なるアプリケーションは、セキュリティを管理するために会社の集中型アイデンティティ・アクセス管理（IAM）ソリューションを使用する必要があります。同時に、各プラットフォームは、中央の IAM 管理のために ID フェデレーションを使用する必要があります。

運用の観点からは、関係するシステムが多ければ多いほど、e コマース・ソリューション全体を管理するのが難しくなります。インシデントや問題に対処するには、システム間のインターフェイスを明確に定義し、綿密な監視を行う必要があります。中央ログ・監視システムは、運用プロセスを簡素化します。中央リポジトリへのログ送信は、ダッシュボードとアラートによる単一で透明性のある環境を提供します。

効果的なeコマース技術組織の立ち上げ方

適切な組織がなければ、どのようなeコマース・アーキテクチャを定義しても、その目的を達成することはできません。CTO として、あなたは、ビジネス価値を提供する文化、人材、プロセス、およびテクノロジーから作業システムを構築しなければなりません。効果的なeコマース組織には、マーケティング、プロダクト、テクノロジーの各分野の人々が、高度に一致した共通の目的と原則を持って協力し合います。

顧客中心、リーン、アジャイル、そしてDevOps

この組織の重要なコンセプトは、顧客中心主義、リーン・アジャイル、DevOpsです。顧客中心主義とは、構築するすべての機能が顧客に価値を提供することを意味します。手近な方法論はデザイン思考とリーンスタートアップで、特にリーンスタートアップのコンセプトである最小実行可能製品と構築-測定-学習サイクルです。Amazon では、"working backwards"と呼ばれるアプローチを使っています。プレスリリースとFAQを書き、ビジュアルを追加して、顧客、そのニーズとベネフィット、問題とソリューションのその他の重要な側面を説明します。AWSはこの手法を顧客と共有しています。

エクストリーム・プログラミング、スクラム、カンバンなどのリーンでアジャイルな方法論は、組織が**予測可能で信頼性の高いプロセスで迅速かつ安全に機能を構築**できるようにするのに役立ちます。

DevOpsには、予測可能で、信頼性が高く、反復可能で、安全で、高速なプロセスで顧客に機能を提供するためのすべてのプラクティスが含まれます。すべてのチームがいつでも機能を本番環境にリリースでき、ソース管理から本番環境へのリリースの期間が日単位ではなく分単位で測定されるようになれば万々歳です。

部門横断的な製品開発チーム

部門横断的な製品開発チームが、この組織の中核を築きます。「部門横断的」とは、これらのチームが、各担当者の機能的なレポートラインに関係なく、製品の定義、構築、出荷、および運用のために協力する、異なる機能領域の人々で構成されていることを意味します。これらのチームの典型的な役割は、プロダクトマネージャー、ソフトウェアエンジニア、ユーザビリティエンジニア、デザイナー、テスターです。

健全なチームの人数は6人から10人程度です。Amazon では、アメリカンサイズのピザ2枚で賄える人数以上のチームは作れないというピザ2枚ルールに従っています。このようなチームのソフトウェア・エンジニアは、専門的な技術的専門知識を持ち、アプリケーション・スタックのあらゆる部分に取り組むことができる**ジェネラリスト**であることを強く推奨します。このようないわゆるT字型またはフルスタックエンジニアは、通常、eコマースソリューションの最も重要な要素を管理することができます。商用ソリューションを使用する組織は、次のような部門横断チームも活用する必要があります。

必要なソフトウェア開発が少ない場合でも、機能横断的なチームを活用する必要があります。ソフトウェア開発者は、それぞれの商用ソリューションの構成の専門家です。

短期的なプロジェクトではなく、長期的なプロダクトを中心に

高いパフォーマンスを発揮するクロスファンクショナルチームは、チェックアウト、商品検索、レコメンデーション、決済などのビジネスプロセスを、すべての技術レイヤー（フロントエンド、バックエンド、データベース）にわたって担当します。これは、チームが顧客のニーズを完全に理解し、長期的な保守性を考慮したソフトウェア品質を設計するための最良の方法です。

これは、プロジェクト・ベースのアプローチとは対照的す。プロジェクト・ベースの場合、チームが何度も結成され、プロジェクトの個々の部分を提供します。結果に対するチームの責任は、納品をもって終了します。メンテナンスや保守性は優先されません。プロジェクト・ベースのチームは、単に一緒に働く個人のグループです。

機能ではなく、価値を届ける

eコマース組織の全メンバーは、機能が必ずしも価値を提供するとは限らないという基本的な信念を共有しなければなりません。一日に何度も変更をリリースし、提供することができるにもかかわらず、組織のKPIが変化しない組織をよく見かけます。その組織は、アウトプットは提供しますが、成果は提供しません。組織は最高の意図を持って機能を構築しますが、その機能は必ずしも顧客のニーズを満たしていません。

このような機能主導の考え方を克服するために、企業は、エリック・リースが著書「リーン・スタートアップ」で初めて詳細に説明した「構築-測定-学習」の考え方を採用する必要があります。仮説駆動型の思考では、アイデアをテストが必要な実験として扱います。組織は、新しいアイデアが現在の実装や他の可能性のあるソリューションに対してどのように機能するかを判断するために、いくつかの**A/Bテストを並行して実行**する必要があります。

これらの実験の成功は、データに基づいて判断されなければなりません。そのためには、強力なビジネスインテリジェンスとデータ分析のスキルが必要です。**アナリティクス、AI、MLは、成功するeコマース組織に不可欠な能力**です。

シンプルで簡単、ストレスのないユーザーエクスペリエンスは、あらゆる eコマースサイトの重要なパフォーマンスドライバーです。

製品管理、エンジニアリング、データスキルに加え、**ユーザー・エクスペリエンス・デザインは、組織が習得すべき4番目の分野です**。ユーザーエクスペリエンスデザインには、フロントエンドの魅力的なルック＆フィールや、合理化されたプロセスフローも含まれます。特にユーザーエクスペリエンスに関しては、データとユーザーからのフィードバックに基づいて意思決定を行う必要があります。ユーザーエクスペリエンスに対するすべての変更は、A/Bテストされなければなりません。

緩やかな結合と高度な連携

ビジネスでは、意思決定、実験、新機能のデプロイ、アップデートのリリースなど、スピードが重要です。どのアーキテクチャを選択するにしても、組織のスピードを最大化するために、疎結合で高度に連携した組織を設定します。

疎結合とは、チームが互いに、また経営リーダーからできるだけ独立して機能できることを意味します。そのためには、**組織内のコミュニケーションやシステム・アーキテクチャの依存関係を減らす**必要があります。

企業は、無政府状態に陥ることなく、自律性を高めるべきです。管理職は、戦術的な日々の意思決定への関与を減らし、その代わりに組織内の戦略的アライメントに焦点を当てるべきです。アライメントを生み出すには、原則や信条、戦略文書、テクノロジー・レーダーなど、いくつかの手段を使います。一方、OKRやカンバンフライトレベルは、チームの戦術的オーケストレーションに適したツールです。

アナリティクス、AI、機械学習の重要性

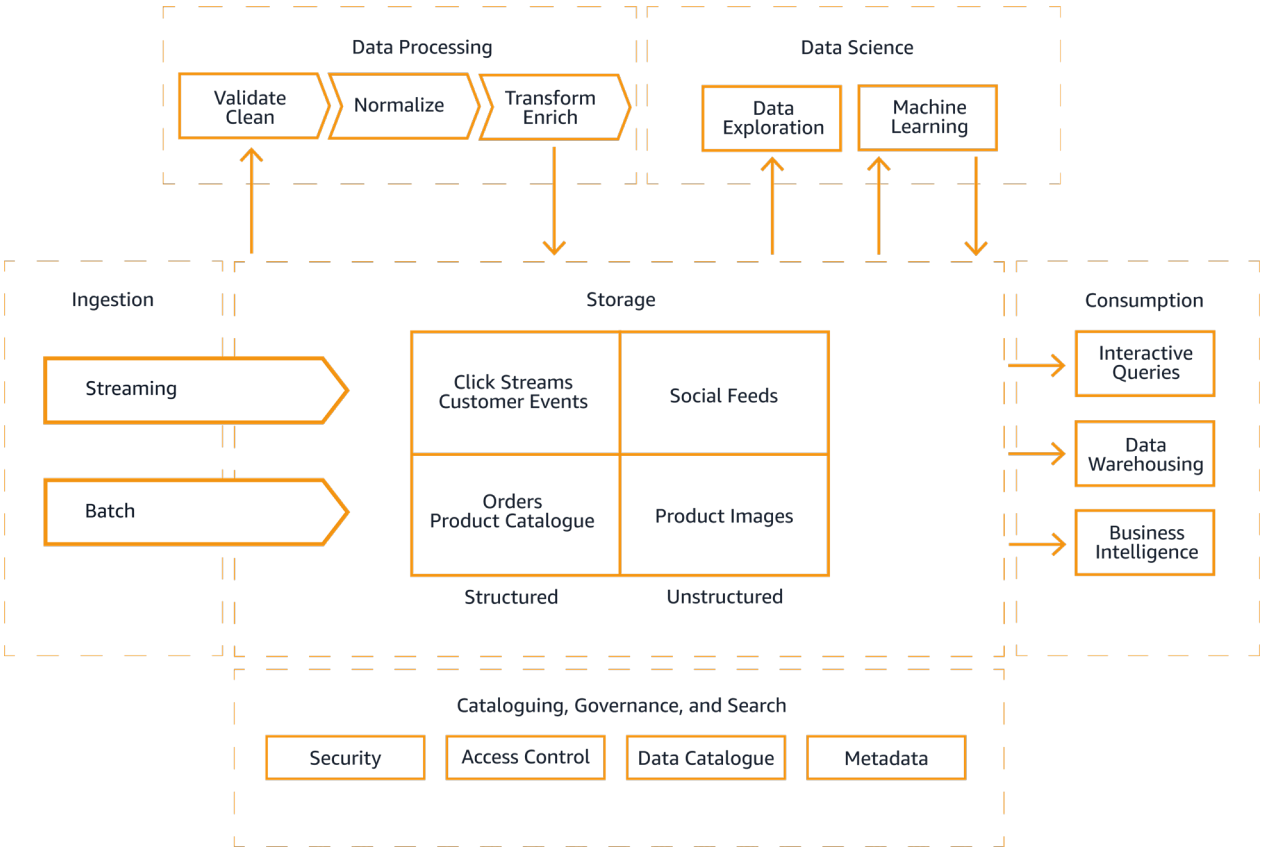
IT部門や管理部門だけでなく、特に製品管理、ユーザーエクスペリエンス、マーケティング、サプライチェーン、ロジスティクスなど、組織全体におけるアナリティクス能力の重要性はいくら強調しても足りません。IT部門は、データを保存し、処理し、可視化するためのインフラを提供します。しかし、**企業はこのデータを洞察、意思決定、行動に変える必要**があります。例えば、パフォーマンスベースのマーケティング費用の管理、SEOとSEM活動の最適化、コンバージョンファネルの改善、商品の品揃えと価格の決定、商品開発ロードマップの優先順位付けのために、企業はデータインサイトが必要です。私たちはこれをデータ駆動型企業と呼んでいます。

今日、AIやMLサービスによって意思決定の一部を自動化し、アプリケーションやワークフローに既製のインテリジェンスを提供することで、ビジネスの成果を向上させることができます。アマゾンは、これらと同じテクノロジーを使って、私たち自身のビジネスを強化しています。Amazon Personalizeを使ったレコメンデーションや、Amazon Connect を使ったコンタクトセンターのインテリジェンスなど、MLの専門知識がなくてもAIを活用したアプリケーションを構築することができます。企業は、Amazon SageMakerを使用して独自のモデルやアルゴリズムを使用し、データサイエンティストや開発者が高品質のMLモデルを迅速に準備、構築、訓練、展開できるようにすることができます。

その技術的基盤がデータレイクです。データレイクとは、構造化・非構造化データをあらゆる規模で保存する一元的なリポジトリのことです。ダッシュボードやビジュアライゼーションからビッグデータ処理、リアルタイムアナリティクス、MLまで、さまざまなタイプのアナリティクスを処理し、より良い意思決定を導くために、データを構造化することなくそのまま保存することができます。データレイクの分散型アプローチは、データ・メッシュ・アーキテクチャです。

組織的な観点からは、アナリティクス、AI、MLのインフラストラクチャは、データプラットフォーム自体を提供し、ユーザーサポートやトレーニングを担当する専任のデータプラットフォームチームによって構築・管理されるべきです。製品チームは、データの生産者と消費者として機能します。彼らは、データとメタデータをデータレイクに公開し、メトリクスの定義と可視化、データ駆動型の製品とインサイトの作成を担当します。

リファレンスアーキテクチャ



提案するソリューションは、コンポーネント指向のアーキテクチャです。コンポーネント指向は、関心事の分離、タスクの切り離し、したがって、プロデューサであれコンシューマであれ、プラットフォームと相互作用する異なるチーム間の自律性を保証します。

アーキテクチャの中心となるストレージレイヤーは、耐久性、拡張性、安全性、コスト効率に優れたコンポーネントを提供し、膨大な量のデータを保存します。Amazon Simple Storage Service(Amazon S3)は、これらすべての要件に適合します。

構造化されているか非構造化されているかにかかわらず、データは事前に定義されたスキーマを必要とせずにAmazon S3オブジェクトとして保存することができます。データの品質や様々なユースケースに基づいて、ストレージレイヤーはさらに以下のようなゾーンに整理されるかもしれません：

- **Raw** オリジナル形式で格納されているデータ。
- **Cleaned** 検証と標準化が完了したデータ。
- **Curated** 組織標準とデータモデルを尊重し、消費可能な状態をホスティングするためのデータ。キュレーションゾーン内のデータは通常、パーティショニングされ、カタログ化されています。

インジェストレイヤーは、AWS Data Migration Service、クリックストリームやモニタリングメトリクスのような内部または外部ソースからのストリーミングデータのためのAmazon Kinesis Data Firehoseのようなサービスを使用して、リレーショナルまたはNoSQLのような運用データソースのような様々なソースからデータをインポートするために、専用のAWSサービスのセットを使用します。Google Analytics 、Salesforce、MarketoのようなSaaSアプリケーションを使用している組織は、完全に管理された統合サービスであるAmazon AppFlowを活用することができます。

処理レイヤーの目的は、マルチステップデータ処理パイプラインを作成またはオーケストレーションすることです。AWS GlueとAWS Step Functionsは、大量のデータを処理するために簡単に拡張できるパイプラインを構築、オーケストレーション、実行するためのサーバーレスコンポーネントを提供します。企業は、ストレージレイヤーのさまざまなゾーンを通過するデータセットのカタログ化、検証、クリーニング、変換を行うことができます。

カタログ化とガバナンスのレイヤーは、ストレージレイヤーのデータと、他のすべてのレイヤーの処理リソースを保護します。アクセス制御、暗号化、使用状況の監視、カタログ化、監査のメカニズムを提供します。AWS Lake Formationはこれらの機能を提供します。

データサイエンスのために、AWSは幅広いMLサービスを提供しています。

eコマースシステムでは、機械学習はユーザーインターフェイスのパーソナライズ、商品レコメンデーション、需要予測、セグメンテーション、またコンタクトセンターのチャットボットや通話後の分析に利用されています。センチメント分析や顧客セグメントのような予測分析は、分析とコンシューマのために最適化されたストレージゾーンに永続化することができます。

企業は、インタラクティブSQL、ビジネスインテリジェンスダッシュボード、およびバッチ処理のために、完全に管理された専用アナリティクスサービスを使用することにより、アナリティクス消費レイヤーを作成することができます。Amazon Athenaは、標準SQLを使用してAmazon S3 内のデータを分析する対話型クエリサービスです。AWS Lake Formationを使用してデータに対してクエリを実行する場合、適切なパーミッションが検証されます。Amazon Redshiftは、ペタバイトのデータをホストして処理し、数千もの高性能なクエリを並行して実行できるフルマネージドデータウェアハウスサービスです。

最後に、Amazon QuickSightは、データを可視化するためのリッチでインタラクティブなダッシュボードを作成するためのサーバーレスビジネスインテリジェンスサービスです。Amazon QuickSightは、予測、異常検知、ナラティブハイライトなどのユースケース向けに自動生成されたMLインサイトをサポートしています。アナリティクスとデータサイエンス機能の拡張を計画している大企業は、データ管理に対するデータメッシュアーキテクチャのアプローチを検討する必要があります。

構築、移行、展開の方法

新しいeコマースシステムを構築、移行、展開する方法についての一般的なアドバイスは、データ駆動型の段階的、反復的アローチを使用することです。

インクリメンタル開発では、製品は完全に機能する垂直方向のスライスに分割されます。各インクリメントは顧客に価値を提供しなければなりません。特定の小さな機能の一部から始め、それを構築し、さらに機能を追加していきます。インクリメントは、地域、顧客タイプ、製品セグメント、データ属性、および非機能要件に基づいて区別することができます。例えば、商品詳細ページの最初のインクリメントは、商品画像、商品名、商品説明など、いくつかの属性を持つシンプルなウェブページです。次の段階では、画像、商品属性、チェックアウトボタン、ウェブトラッキング、モニタリングなどを追加します。

システムの水平構築は避けましょう。例えば、属性とリレーションシップでデータベースを作成し、ビジネスロジックを構築し、それからフロントエンドを追加するといった方法です。通常、このアプローチは多くの推測的な設計、誤った仮定、複雑さをもたらします。

反復型ソフトウェア開発では、アイデアからリリースまでの全プロセスを何度も連続して実行します。この繰り返しが、開発プロセスを硬化させ、改善し、多くの場合、品質と予測可能性の向上につながります。反復の終わりには、すべての完成したインクリメントが顧客に提供されます。継続的デリバリーとデプロイメントを備えたマイクロサービスアーキテクチャでは、このイテレーションは1日に数回行われるかもしれません。

データ駆動型ソフトウェア開発では、本番環境にリリースされ、QAによって承認された時点で、その機能が完成したことにはなりません。完了とは、顧客のニーズが満たされたことを意味し、**ニーズが満たされたかどうかは、それを測定して初めてわかるのです**。開発者は、その機能が本当に機能することを示すデータや逸話を集める必要があります。例えば、何人の顧客が新機能を見つけたか、何人がそれを使用したか、何人の顧客がその機能の背後にあるタスクを成功裏に完了させたかを測定します。構築、測定、学習、最適化は、目標が達成されるか、取り組みが改善をもたらさなくなるまで続けます。
機能の成功率が低い場合は、ECシステムの全体的な複雑さを軽減するために、機能を削除することを検討してください。システム準備の観点から、このアプローチは「機能の完成」よりも「ビジネス価値の提供」に焦点を当てます。

これら3つの戦略は、本稿で説明する4つのアーキテクチャすべてに有効です。しかし、具体的な戦術や実装プロセスは異なるかもしれません。

あなたのビジネスにeコマースを導入する場合、デプロイに時間がかかる機能満載のショップを構築するのではなく、最小限の機能セットから始め、ユーザーのニーズを学ぶことをお勧めします。最初の段階から、顧客からのフィードバックで成功を測定し、提供ロードマップを推進する方法に焦点を当てましょう。

旧システムから新システムに移行する場合は、ハイブリッドセットアップの中で、ページごと、機能ごとに段階的なアプローチをお勧めします。旧システムと新システムのKPIをA/Bテストやパフォーマンスモニタリングで常に測定し、比較します。しばらくの間、両システムは並行して運用されるため、短期的には運用が複雑になりますが、このアプローチにより、非常にきめ細かいレベルでリスクを軽減することができます。

インクリメンタル・ハイブリッド・アプローチに従えず、新しいシステムを並行して構築する必要がある場合は、全体のトラフィックのほんのわずかな量を使って、新しいシステムをかなり早い段階で本番稼動させます。その後、旧システムの全体的なビジネス・パフォーマンスを達成するまで、新しいインクリメントを構築し、最適化します。機能が追加され、パフォーマンスが向上すれば、新システムのトラフィックを増やすことができます。旧システムの機能を完全に使いこなす必要はありません。システムの性能に匹敵するか、それを上回る必要があります。

よくある間違い

データドリブンアプローチを用いない

最も一般的な間違いは、eコマースにデードリブンタのアプローチを用いていないことです。例えば、フロントエンドのユーザーインターフェイスに変更を加える際に、その影響を測定せず、KPIをモニタリングしないことです。データインサイトは、すべてのビジネスおよび技術的意思決定のロードマップと優先順位を推進する必要があります。インサイトを取得しても、インサイトを意思決定の推進に適切に使用しない企業は成功しません。常に価値を優先し、頻繁に実験を行い、データで仮説を検証すべきです。

メジャーリリース主義

eコマースの分野では、通常、顧客の流れが絶えません。新しい仮説のテストは、大きな機能の一部だけで可能なこともあります。企業は、本格的なeコマースソリューションの新たなメジャーリリースのために何ヶ月も何年も待つことは避けるべきです。常にモニタリングしながら、少しずつリリースする方がずっと良い選択です。

最初に戦略を定義しない

eコマース・チャネルをあまりにも長い間無視し、時代遅れのテクノロジーで運営してきた企業は、やがて見逃してきた可能性に気づくようになります。そうなると、大胆な目標を設定し、大規模なオーバーホール・プロジェクトに資金を提供したくなるものです。その代わりに、最新のeコマース・プラットフォームには、クラウド技術を採用するのと同様に、根本的に異なる考え方が必要です。最初のステップとして、企業は正しいeコマース戦略を定義するための専門知識を求めるべきです。

細かな最適化への固執

技術的および組織的な観点から、特に非結合の分散アーキテクチャでは、ローカルな最適化に集中する傾向があります。リードアーキテクトは、たとえそれが最高のコンポーネントで作られたとしても、分散した泥の玉で終わることを避けるべきです。その代わりに、顧客に焦点を当て、ジャーニー全体で最適化します。分散アーキテクチャの領域において、もう1つの罠は、組織が管理できる範囲を超えてシステムや技術コンポーネントの数を増やすことです。これは個々のチームにも同様に当てはまります。すべてのレベルで認知的負荷を適切に管理することは、長期的に見返りがあります。

結論

ここまで、以下3つのアーキテクチャについて議論してきました:

- 内製マイクロサービス – ウェブ、モバイル、ソーシャルコマース、小売キオスク/エンドレスアイル、多くのサードパーティパートナーとのロイヤリティプログラムなど、意欲的なオムニチャネルプログラムを展開し、イノベーションと差別化を重視する企業に最適です。
- 商用スイート – ウェブとモバイル以外のオムニチャネルをほとんど持たず、高度に革新的で差別化されていない標準的なeコマース機能を求める企業に最適です。
- 内製フロントエンドと商用バックエンド – 標準的なeコマース・プロセスと、異なる販売チャネル向けに高度に差別化されたユーザー・インターフェースを求める企業に最適です。

しかし、**eコマース・アーキテクチャやツールよりも重要なのは組織**です。これには、あなたの考え方やプロセスだけでなく、あなたが惹きつける人材も含まれます。機能を構築することよりも、顧客のニーズを満たし、ビジネス価値を提供することに重点を置くべきです。その代わり、プロジェクトではなく、製品に責任を持つ独立した部門横断チームを作りましょう。データと顧客からのフィードバックが取り組みの原動力です。

データドリブンのeコマース・ソリューションの基盤は、組織内のすべての適切な人々がすぐに利用できるリアルタイムの洞察、予測、推奨を提供する強力なデータ・プラットフォームです。

著者 / 寄稿者

Matthias Patzak はAWSのプリンシパルアドバイザーです。AWSに入社する前、MatthiasはAutoScout24のIT担当副社長とHome Shopping Europeの最高デジタル責任者 (CDO) でした。両社において、彼はリーン-アジャイル運用モデルを大規模に導入し、納期短縮とビジネス価値の向上をもたらすクラウド変革を成功に導きました。

Paul Vassu はAWSのManager Solutions Architectureで、小売企業を担当しています。Paulはソフトウェアエンジニアリングで20年の経験があり、AWSに入社する前は adidasのソフトウェアエンジニアリングのディレクターでした。そこでは、アジャイルとDevOpsのプラクティスの採用を推進しながら、eコマースのランドスケープの近代化に貢献しました。

以下の AWS 寄稿者に感謝します:

- **Michael Needham**, SA Readiness Specialist
- **Till Hohenberger**, Senior Solutions Architect
- **Jordi Fernandez Moledo**, SA Senior Specialist
- **FrankBlessing** , Solutions Architect
- **DavidDorf** , Principal Retail GTMS
- **Alejandro Mondragon**, Principal BDM
- **Bastien Leblanc**, Senior Solutions Architect
- **Kempei Igarashi (訳者)**, Sr. Manager, Retail, Japan

出典

1 https://public.tableau.com/profile_salesforcEcommercecloud#!/vizhome/SalesforceShoppingIndex/SalesforceShoppingIndex

2 Global Ecommerce Update 2021, Inside Intelligence, eMarketer, Jan 2021, Ethan Cramer-Flood

3 https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/think-fast-how-to-accelerate-e-commerce-growth