

GeneXus アプリケーションのアーキテクチャ

GeneXus™

バックオフィス重視の Web アプリケーション

観光名所

観光名所番号

0

観光名所名

コロッセオ

国番号

5

📌

国名

イタリア

都市番号

0

都市名

カテゴリ番号

0

カテゴリ名









観光名所写真

観光名所s

追加

🔍 観光名所名

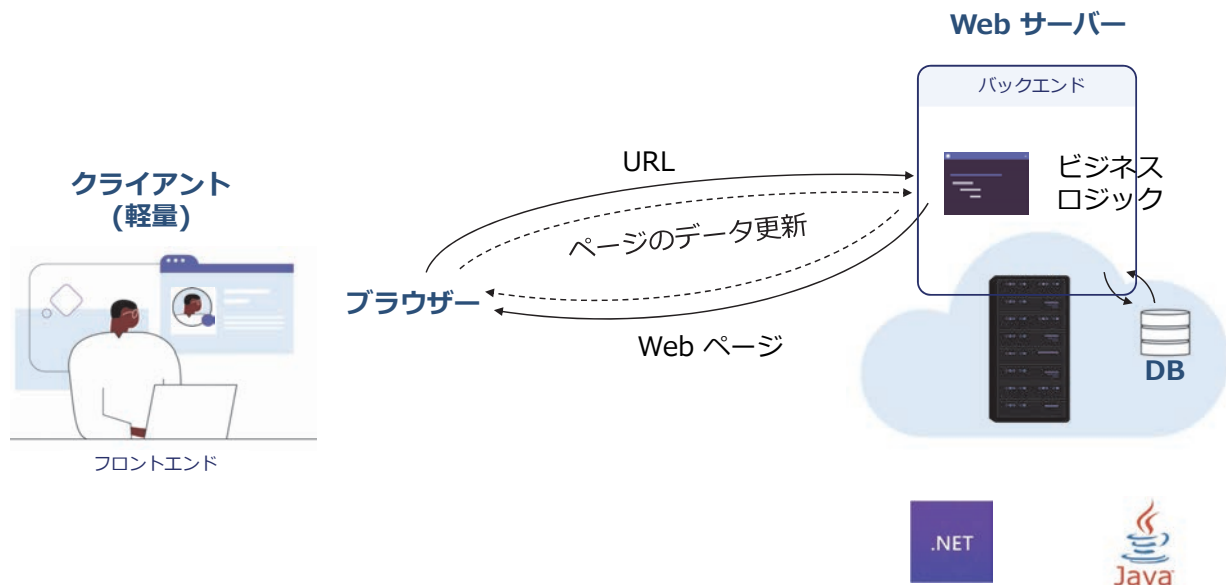
▼

観光名所番号	観光名所名	国番号	国名	都市番号	都市名	カテゴリ番号	カテゴリ名	観光名所写真		
16	エッフェル塔	2	フランス	1	パリ	8	モニュメント		更新	削除
17	サグラダ・ファミリア	9	スペイン	1	バルセロナ	8	モニュメント		更新	削除
18	スミソニアン博物館	10	アメリカ	2	ワシントンDC	6	ミュージアム		更新	削除
20	ノートルダム寺塔	2	フランス	1	パリ	9	有名なランドマーク		更新	削除
19	マティス美術館	2	フランス	2	ニース	6	ミュージアム		更新	削除
15	ルーブル美術館	2	フランス	1	パリ	6	ミュージアム		更新	削除
22	紫禁城	3	中国	1	北京	7	遺跡		更新	削除
21	万里の長城	3	中国	1	北京	7	遺跡		更新	削除

ここまで、トランザクションオブジェクトを中心に機能の説明を行ってきました。
トランザクションオブジェクトに基づき、生成された画面では、各フィールドからフォーカスが移動するときにデータが検証されました。
その結果、メッセージが表示される場合や、外部キーのフィールドであれば、従属項目の値が取得され、自動で表示されました。
このような挙動については、UX を向上させるための「クライアント側検証」と呼ばれていました。

また、トランザクションオブジェクトには、パターンと呼ばれる機能を適用し、いくつかの機能が簡単に実装出来ました。
この機能のうちの 1 つに一覧表示の画面があり、ページングやフィルタが自動で含まれていました。
これらの機能を利用した場合、動的にデータの読み込みが行われ、対象のデータのみが表示されました。

バックオフィス重視の Web アプリケーションのアーキテクチャ



こうした処理の一部は Web アプリケーションのクライアント側（ブラウザ）で実行され、データベース検索などその他の処理はサーバー側で実行されます。

アプリケーションのロジックはサーバー側にあるため、アプリケーション内でオブジェクトを実行すると、ブラウザからサーバーに、対応するページがリクエストされます。

サーバーは情報を準備し、必要に応じてデータベースからデータを取得し、画面を組み立てて、それを表示のためにブラウザに送信します。

ブラウザにページが表示されたら、何らかの操作を開始することがあります。

たとえばグリッド内でクエリにフィルタを適用します。

この場合、クライアント側のコードがサーバー側のコードと通信します。

フィルタの条件を満たすデータがサーバーから返されたら、画面上で処理に関係のある部分のみが更新されます（この例ではグリッドのみ）。

グリッド内での列の並べ替えや、トランザクションでフィールドを離れたときのメッセージ表示などは、クライアント単独で解決され、サーバーへのアクセスは行われません。

このアーキテクチャでは、クライアントも一定のインテリジェンスを持ちますが、決定はサーバーで行われます。アプリケーションのロジックは完全にサーバー側にあるからです。

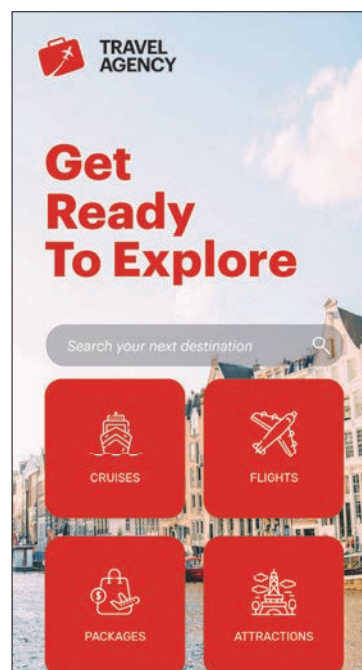
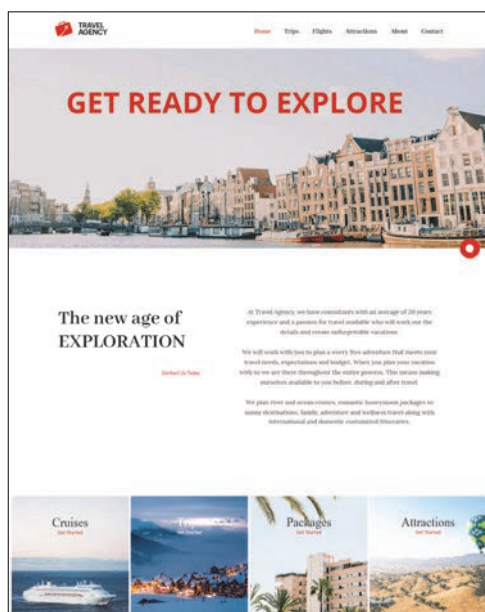
これまで開発してきたのは、基本的にアプリケーションのバックオフィスです。

データベースを操作する方法（トランザクション）、情報を階層的に照会し、追加、削除、変更を行う方法（Work With パターンで構築するものすべて）を見てきました。しかし、この同じアーキテクチャで顧客向けアプリケーションをビルドすることもできます。

GeneXus でこれらのアプリケーションをビルドするには、.NET、または Java を使用します。

GeneXus では、クライアント（フロントエンド）とサーバー（バックエンド）の両方のコードにこれらの言語を使用します。

よりインタラクティブなコンテンツを備えた高性能アプリケーション



「顧客向けアプリケーション」について、従来のアーキテクチャに基づき、生成することが可能であると説明しました。

しかし、このタイプのアプリケーションは、利用するユーザーに重点が置かれ、画面上の情報を充実させるべきものです。

そのため、極めて高いパフォーマンスを発揮し、デザインと双方向性に重点を置く必要があります。

また、優れたユーザーエクスペリエンスを目指し、ページは必要な時に必要な部分だけロードを行います。

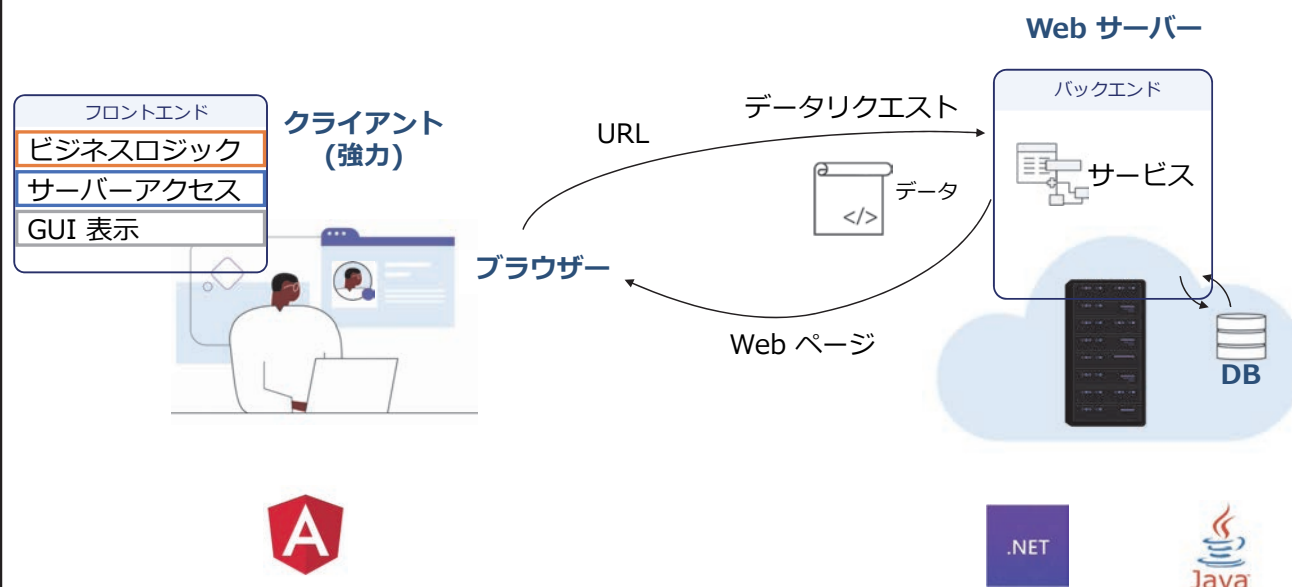
このような実装のためには、異なるアーキテクチャが必要となります。

「顧客向けアプリケーション」は、Web アプリケーションとして生成される場合と、スマートデバイスにインストールされるネイティブモバイルアプリケーションの場合があります。

GeneXus では、異なるアーキテクチャによる「顧客向け Web アプリケーション」も「ネイティブモバイルアプリケーション」も生成可能です。

ただし、本コースでは、これらのアプリケーションの生成について詳細は取り扱いません。

UX 重視の Web アプリケーションのアーキテクチャ



このアーキテクチャは、クライアントで実行される部分とサーバーで実行される部分がある点は同じですが、アプリケーションのほとんどのロジックがクライアント側にある点が異なります。

クライアントには 3 つの異なるレイヤーがあります。ビジネスロジック、サーバーとの通信、そしてブラウザ内の画面表示の部分です。

この構成では、ほとんどの機能がクライアント上で実行されます。サーバーにアクセスするのは、データベース内のデータや、サービスが提供するその他のリソースをリクエストまたは変更するときのみです。

このタイプの顧客向けアプリケーションは、優れたユーザーエクスペリエンスや双方向性を実現するために強力なクライアントを必要とします。

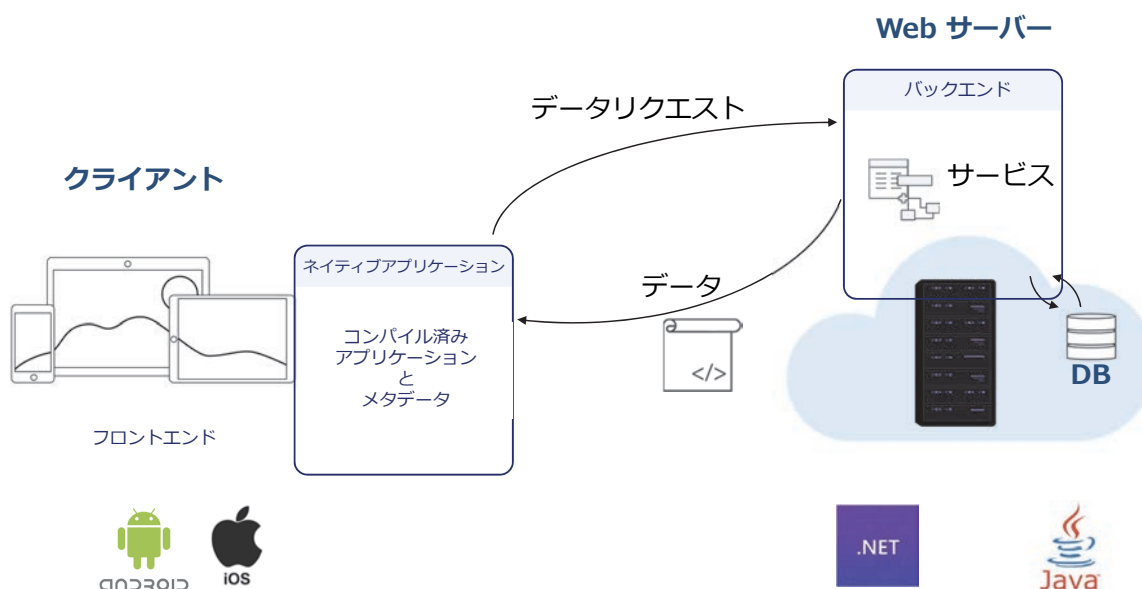
こうした需要に対応するため、このようなアプリケーションに必要なスマートクライアントをビルドできる Angular などのフレームワークが市場に登場しました。

前述のように、GeneXus では、.NET、Java、でも Web アプリケーションを開発できます。

しかし、顧客向けの Web アプリケーションを生成する場合、Angular でプログラムを生成します。

そのメリットは、ユーザーインターフェース用にデザインする同じ Panel オブジェクトをほとんど変更せずに再利用して、Android や iOS などのネイティブモバイルアプリケーションを生成できることです。

ネイティブモバイルアプリケーションのアーキテクチャ (オンライン)



本コースでは、Web アプリケーションを対象としますが、GeneXus では、ネイティブモバイルアプリケーションの生成も行えます。
このアプリケーションの場合、Angular ジェネレーターによる顧客向けアプリケーションと似たアーキテクチャとなります。

ネイティブモバイルアプリケーションにも、クライアント、つまりスマートデバイス上で実行される処理と、サーバーで実行される処理があります。
アプリケーションのプログラムは、スマートデバイス上で実行され、データが必要な場合、サーバーへ情報提供を行うようにリクエストを行います。
また、ここで説明しているアーキテクチャは、スマートデバイスがサーバーに接続可能なネットワークにつながっている場合を対象にしています。
GeneXus では、オフラインアプリケーションを生成することもできますが、その場合は、異なります。

GeneXus では、Android および iOS 向けのネイティブモバイルアプリケーションを生成することができます。
この場合も、サーバーのサービスも生成は必要となり、.NET、Javaから選択可能です。

2 種類の実装方法

軽量
クライアント
+
サーバー上の
ビジネス
ロジック

バックオフィス重視の
Web アプリケーション

スマート
クライアント
+
サーバー上の
サービス

UX 重視の
Web アプリケーション
モバイルデバイス向け
ネイティブアプリケーション

アプリケーションのアーキテクチャが、アプリケーションの多くの側面、特に実装方法に影響します。

軽量クライアントの場合、アプリケーションの要求のほとんどは、サーバーによって解決されます。

このため、実装は、データベースへのアクセスなど、常にサーバー処理に重点を置きます。

これに対して、強力なクライアントの場合、多くの処理を、サーバーに頼らずにクライアント上で実行できます。ただし、主にデータベースアクセスを伴う場合にサーバー処理が必要となります。

このため、特定のコードをクライアント側で実行する場合と、サーバー側で実行する場合とで、異なる構文が必要となります。

これは、モバイルデバイス向けネイティブアプリケーションのプログラミングにも当てはまります。

GeneXus[™]