はじめに

みなさんはソフトウェアエンジニアとしてアプリケーション開発の現場で働いていますか。今、開発をしているアプリケーションに何か課題を感じていますか。それとも「技術的に新しい何か」を学ぶことに目を輝かせていますか。

本書を読まれているということは、本書のタイトルに惹かれたのではないでしょうか。「モダン」という言葉に興味を持たれたのかもしれませんし、逆に「モダン」という言葉に違和感を感じられたのかもしれません。

本書では「モダンアプリケーション」を取り扱います。モダンアプリケーションとは、アプリケーションの設計、構築、管理を継続的に見直し、常に変化を受け入れ続ける開発戦略のことです。「この技術を使えばモダンである」といった話ではなく、フィードバックループを回し、すばやくユーザーに価値を届けるために、実現したいことから逆算をしてテクノロジーやアーキテクチャを選択することが重要です。そんな「モダンアプリケーション」という言葉に込められた意味をお伝えしたいと思っています。

さて、近年はデジタルトランスフォーメーション化が急務であると言われるなど、ICT（Information and Communication Technology）を活用したサービス提供が増えています。また、従来のように長期間の開発期間を設けてすべての機能を実装してからリリースをするのではなく、優先順位の高い機能から積極的にリリースし、ユーザーの声に耳を傾けることによりフィードバックループを回す開発戦略を採用する企業も増えています。それに伴って、アプリケーション開発では今まで以上に柔軟性や俊敏性が重視されることから、アプリケーションの基盤としてクラウドを活用する機会も増えているのではないでしょうか。

クラウドを活用する、と言っても活用方法はさまざまです。従来からよく知られた仮想マシンベースのアーキテクチャもあれば、近年、耳にすることが増えたであろうサーバーレステクノロジーやコンテナテクノロジーを活用したアーキテクチャもあります。そして、アプリケーションの全体最適化を目指すアーキテクチャとして、「モノリシック」と言われるような一枚岩の大きなアプリケーションとして運用する場合もあれば、「マイクロサービス」と言われるような細かなサービスに分割したアプリケーションとして運用する場合もあります。どれを選択すれば良いのでしょうか。どんな観点で選択すれば良いのでしょうか。なんと

なく流行っているから、という理由で選択すれば良いのでしょうか。ソフトウェアエンジニアとして、流行っている技術に興味を持つことは重要ですし、選択したくなる気持ちもわかります（そういうモチベーションで選択することもときには必要かもしれませんね）。

しかし、明確な理由をもってテクノロジーやアーキテクチャを選択するためには、それぞれの特徴や使い所、メリットやデメリットを理解している必要があります。そのためには、多くのテクノロジーを試し、多くのアーキテクチャを吟味し、技術の進歩へ追従するために学び続ける必要があります。また、モダンアプリケーションを実現するためには、常に変化を受け入れ続けることが重要です。それらのテクノロジーやアーキテクチャによって、どのようにビジネスやアプリケーションの要件を満たすことができるのだろうか、という観点も必要です。前述のサーバーレステクノロジーやコンテナテクノロジー、マイクロサービスアーキテクチャといった要素をただ選択するのではなく、要件にあった適材適所のテクノロジーとアーキテクチャを選択する必要があるからです。

さて、筆者（落水）は、ソリューションアーキテクトとして、お客様の技術的な課題の解決を支援しています。近年のビジネス状況の変化から、モダンアプリケーションに関する相談は増えてきていますが、前述のようにモダンアプリケーションには多くのトピックが関連しているため、「難しい」という声をよく聞きます。そういったお客様を支援する中で、モダンアプリケーションそのものであったり、マイクロサービスやThe Twelve-Factor Appといった関連するアーキテクチャ・プラクティスについて、情報収集や学習のコストが高いと感じていました。本書を執筆しようと思ったキッカケは、モダンアプリケーションに対する「難しい」という気持ちを取り除き、多くの方にモダンアプリケーションへチャレンジしてもらいたいと考えたからです。

筆者（吉田）は、シニアテクニカルトレーナーとして、おもに有償トレーニングを通じてお客様の人材育成を担当しています。クラウドを活用したテクノロジーやアーキテクチャを学びたいと思っても、どうやって学べば良いかわからなかったり、独学で学んでみたけどわからないことが多く諦めたりしたという声をお客様からよく聞きます。そういったお客様の「学びたいという気持ち」を支える仕事に携わっています。その中でも、モダンアプリケーションに関連するトピックは、サーバーレステクノロジーやコンテナテクノロジー、マイクロサービスアーキテクチャなど、学ぶべき要素が多いと感じられることから独学では学びにくいのかもしれません。そこで、これらのトピックを効率的に学ぶために有償

トレーニングを受講されるお客様も増えています。本書を執筆しようと思ったキッカケは、有償トレーニング以外でも、モダンアプリケーションに興味を持っているお客様の「学びたいという気持ち」を支えたいと感じたからです。

ぜひ、本書をきっかけにモダンアプリケーション化の第一歩を踏み出しましょう。

本書で取り扱う内容

前述のとおり、本書ではモダンアプリケーションに関連する代表的なトピックを取り扱います。そして、以下のような読者を対象としています。なんとなく流行っているから、という理由でテクノロジーやアーキテクチャを選択するのではなく、実現したいことから逆算をして、自信を持ってテクノロジーやアーキテクチャを選択できるようになることを目的としています。

- **モダンアプリケーションに興味がある**
- **モダンアプリケーション化を検討しているが、どこから始めれば良いのかわからない**
- **歴史のあるサービスに敬意を払いつつ、課題も多いので改善したい**
- **とくに理由もなく、なんとなく技術選定が行われている現場に違和感を感じる**

本書で取り扱わない内容

一方で、以下のような方は本書の対象読者として想定していません。本書は、モダンアプリケーション化を検討するための「考え方」を伝えることに比重を置いているからです。説明のためのサンプルコードや設定例、またそれぞれのAWSサービスの仕様なども一部載せていますが、詳細には説明していません。ドキュメントを読んだり、公開されているハンズオンを実施したりすることで、より学習効果を高められるでしょう。

- **具体的なプログラミング実装を学びたい**
- **AWS サービスの操作方法を学びたい**
- **ハンズオン形式で手を動かして学びたい**

本書で大切にしていること

本書を執筆するにあたり、筆者陣でとても大切にしていることがあります。

■初学者でも読める「わかりやすさ」を追求すること

まずは、初学者でも読める「わかりやすさ」を追求することです。近年、何か新しいテクノロジーやアーキテクチャを学びたいと思ったときに、書籍やe-Learning、動画、有償トレーニングもありますし、ドキュメントやブログ記事などのインターネットコンテンツ、そしてホワイトペーパーを読んで学ぶこともできます。学ぶためのコンテンツは多くあるため、限られた時間のなかで、コンテンツの取捨選択をする時代であるとも言えます。しかし、コンテンツによっては、読者にある程度の技術的な前提知識を求めることがあり、実際にコンテンツを手にすると「理解できなかった（自分にはまだ早かった）」と感じたこともあるのではないでしょうか。

読者のみなさんが学びながら感じるであろう「なぜ」という疑問を大切にすることで、暗黙的な知識にできる限り依存しないように意識しました。極端な例を挙げます。突然「テクノロジーXを紹介します。次はテクノロジーYを紹介します。」と聞くと唐突に感じますが「現在はAという課題があり、Bという特徴によって課題を解決できるため、テクノロジーXを紹介します。」と聞くと、スッキリと理解できるのではないでしょうか。ほかにも、比喩表現を活用したり、図表を多く載せる工夫もしています。もし本書を読んで「わかりにくい」と感じられた場合はきっと筆者陣の問題です。ぜひ改善のためにフィードバックをお待ちしています。

■具体的なシナリオに沿って読み進められるようにすること

モダンアプリケーションの中には、多くのプラクティスやパターンが存在します。プラクティスやパターンは、抽象度を高め、適用の幅を広げながら一般化したものであるため、読者のみなさんによってはわかりにくく感じられることがあるはずです。そういった抽象度の高い内容を説明するときに有効なのは、具体的な例を挙げることです。

本書では、架空の企業とその企業が開発・運用をするアプリケーションという具体的なシナリオを使って解説をすることで、読者のみなさんがイメージをしながら読み進められるようにしています。当然ながら、読者のみなさんが仕事など

で開発・運用をするアプリケーションとは規模やフェーズも異なるでしょう。適宜読み替えて読み進めるのも良いでしょう。

■アクティビティを通して考える余白を作ること

本書では、プラクティスやパターンを通じて、こうしたら課題を解決できる、という「考え方」に比重を置いて解説しています。テクノロジーやアーキテクチャの技術選定に限った話ではありませんが、当然ながら決まった答えが「1つ」あるわけではありません。ほかにも選択肢はあるでしょうし、市場やアプリケーションの規模、または企業の技術戦略によっても選択肢は変わります。さらに、その時点では良い選択肢だったものが、数ヵ月後には課題を抱えてしまう可能性すらあります。本書では「アクティビティ」という「読者のみなさんだったらどう考えますか」という考えながら読めるしかけを各章に散りばめています。本書で紹介していない選択肢を考えてみるのも良いでしょうし、チームでアクティビティをテーマにディスカッションをしてみるのも良いでしょう。本書をきっかけに考え続けてほしいと考えています。

本書の構成

本書の章立ては以下のような構成になっています。単独で読むこともできますが、第1章から第8章まで、シナリオとしてつながっています。順番に読むことをおすすめします。

- **第1章：モダンアプリケーションとは何か**
- **第2章：サンプルアプリケーションの紹介**
- **第3章：アプリケーション開発におけるベストプラクティスを適用**
- **第4章：データの取得による状況の可視化**
- **第5章：サーバーレスやコンテナテクノロジーによる運用改善**
- **第6章：CI/CDパイプラインによるデリバリーの自動化**
- **第7章：要件にあったデータベースの選択**
- **第8章：モダンアプリケーションパターンの適用によるアーキテクチャの最適化**

第1章では、イノベーションが求められる現代におけるモダンアプリケーションの必要性を紹介します。さらに、モダンアプリケーション化のメリットやモダ

ンアプリケーション化に必要なベストプラクティスを紹介します。

第2章では、本書で取り扱う具体的なシナリオとして、架空の企業Sample Companyとその企業が開発・運用をするアプリケーションSample Book Storeを紹介します。どのような企業が、どのような課題を抱えてモダンアプリケーション化を目指すのか、できる限り具体的にまとめています。

第3章では、モダンアプリケーション化に着手する前の準備として、アプリケーション開発におけるベストプラクティスであるThe Twelve-Factor AppとBeyond the Twelve-Factor Appを活用し、アプリケーションの見直しを行う方法を紹介します。これまで十分に価値を生んできたアプリケーションを、最初から抜本的に変える必要はなく、歩幅を小さく一歩一歩改善できることを紹介します。

第4章では、データを取得することの必要性を紹介します。ビジネスデータ、運用データ、そしてシステムデータなど企業やアプリケーションに関連するさまざまなデータを取得し可視化することで、モダンアプリケーション化に限らずより良い判断ができるようになります。

第5章では、では、サーバーレステクノロジーやコンテナテクノロジーを活用することで得られるメリットを紹介します。また、サーバーレスワークロードとコンテナワークロードの比較も紹介します。具体的なシナリオに当てはめて、なぜサーバーレステクノロジーやコンテナテクノロジーを選択するのかをまとめています。

第6章では、アプリケーションを継続的にリリースするための考え方として、継続的インテグレーションと継続的デリバリーとは何かを紹介します。そして、それらを自動化するために、CI/CDパイプラインに求める機能や要件を整理しつつ、構成例を紹介します。

第7章では、データベースに焦点を当て、要件にあった適材適所のデータベースを選択する方法を紹介します。そして、データベースに求める機能や要件を整理しつつ、具体的なシナリオに当てはめて、構成例を紹介します。

第8章では、アプリケーションをより開発しやすく、運用しやすく、拡張しやすくするための最適化としてモダンアプリケーションパターンを紹介します。アーキテクチャを設計したり、アプリケーションを実装したりするときには何かしらの課題を抱えたり、満たすべき複数の要件がトレードオフの関係になったりすることもあります。そういった、起きうる課題とその解決策になるパターンを、具体的なシナリオに当てはめて紹介します。

目次