



マイクロサービスという 新標準が導く世界

- マイクロサービスで開発の俊敏性と変更の柔軟性を
- 日本 IBM の 2017 年クラウド戦略
- マイクロサービス導入による破滅を避けるための 8 つのルール



マイクロサービスという 新標準が導く世界

マイクロサービスで開発の俊敏性と変更の柔軟性を -- ガートナー飯島氏

3月16日、「ガートナー エンタプライズ・アプリケーション戦略&アプリケーション・アーキテクチャサミット2017」のセッションの1つとして、ガートナーのリサーチ部門でリサーチディレクターを務める飯島公彦氏が登壇。「マイクロサービス：適用すべきか否か、それが問題だ」と題して講演した。

飯島氏はマイクロサービスについて「単独で展開できる疎結合型のアプリケーションコンポーネント。SOA（サービス指向アーキテクチャ）やドメイン駆動設計（DDD）の組み合わせを基礎とする」と端的に説明。マイクロサービスが望まれる背景について「デジタルビジネスの実現にはアジリティ（俊敏性）が必要」とした。

マイクロサービスとは、「開発の俊敏性、デプロイの柔軟性、精緻な拡張性の3つを実現することを目的に、大きくてモノリシックなアプリケーションを、独立した小さい塊へと分割する考え方」（飯島氏）のこと。依存性を減らして取り払った結果として、小さい単機能のコンポーネントになる。

例えば、ホテルの部屋を予約するという処理の場合、予約を受け付ける機能や、ホテルの空き状況を調べる機能、ポイントを調べる機能、などがスコープとして独立している。これらをマイクロサービスとして動作させることができる。

マイクロサービスのメリットの1つは、他のモジュールに対して影響を受けたり与えたりすることなく、変更やデリバリーが可能になること。目的に合った開発言



ガートナーリサーチ部門リサーチディレクター 飯島公彦氏

語と実行環境をデプロイ単位に選択できることや、必要とするシステムリソースが少なく、負荷の増減に合わせてスケールイン/アウトが迅速にできることもメリットだ。

多くのマイクロサービスは「Web API」の形態で公開される。マイクロサービスが駆動するタイミングと方法については、イベントドリブンで自律的に駆動するやり方もあれば、外部のオーケストレーションソフトからバッチ処理の一部として呼び出されるやり方もある。

マイクロサービスという 新標準が導く世界

飯島氏は、ロジックとデータの持ち方の違いに応じて、マイクロサービスの実装のパターンをいくつか挙げた。基本パターンは、マイクロサービス内にデータを持たず、外部からデータを入力して結果を返すというものだ。一方、ポイント計算サービスなどのように、マイクロサービス内にデータを持ったほうがいいものもある。

需要に合わせてマイクロサービスを開発し、継続的に改善せよ

講演の中盤では、マイクロサービスを適用するに当たって留意すべき事柄について解説した。基本は、依存性を減らすことを考える必要がある。

(1)として、まずはデマンド（需要）を考慮することが大切という。「APIは商品であり、市場が求めるものを提供すべき」（飯島氏）。アウトサイドインの発想が大切だとした。

「APIを利用するのは誰か」を先に考えることによって、マイクロサービスに求められる拡張性やスピードが決まる。「今持っているものを公開する、というプロダクトアウトの発想ではだめ」（飯島氏）

(2)として、プロセスを考慮することが大切になる。「継続的なデリバリーを目指すべきだ。市場からのフィードバックやログ分析を、アジャイルなやり方でマイクロサービスに反映する」（飯島氏）

APIのライフサイクルとサービス実装サイクルを分離することも大切なことという。「APIのライフサイクルは長く、あまり頻繁には変更しない。使う側に影響があるようなAPIの変更は極力減らす。作り手側の都合でAPIをコロコロ変えてはいけない」（飯島氏）

(3)として、マイクロサービスの稼働環境は、クラウドベンダーのメニューを利用するのが現実的とした。マイクロサービスを実現するためのソフトウェアのアーキテクチャが複雑だからだ。

例えば、マイクロサービス自身の実行環境だけでなく、サービス間の通信やサービスの発見、非同期メッセージング、外部とのやりとり、自動化、性能や可用性の監視、などの仕組みで構成する。

(4)として、開発チームは、フルスタックのスキルを備えた小さなチームが望ましい、とした。「設計からデプロイまでを1人で担当する」（飯島氏）。チームは機能ごとに分かれており、機能同士はAPIで連携する。このため、チーム間のコミュニケーションを密にする必要がある。

小さく始めて継続的な適用を心がけよ

講演の後半では、マイクロサービスを使いこなす上での、ガートナーとしての提言を示した。

マイクロサービスの適用は、機会と準備状況を勘案し、慎重に判断することが大切になるという。また、マイクロサービスの適用の是非に関わらず、外部にサービスを公開する場合は特に、API化を目指すべきとした。

PoC（試験導入）やパイロットプロジェクトからスタートすることも大切なことだという。「最初は小さく、具体的に始め、成果と知見を獲得し続ける。継続的改善、継続的構築、継続的適用を心掛ける」（飯島氏）

講演の最後に、CIOとITリーダーに向けた行動計画として、直ちに実行すべき事項、90日以内に実行すべき事項、180日以内に実行すべき事項を提示した。

マイクロサービスという 新標準が導く世界

直ちに実行すべき事項はこうだ。「マイクロサービスの適用を検討している企業は、適用の機会や準備状況について確認する。デジタル・ビジネスの取り組みの中で、アジリティとスケーラビリティが必要な領域を探し、マイクロサービス適用の機会を探る」。

90日以内に実行すべき事項はこうだ。「マイクロサービス案件の担当開発者は、RESTful Web サービス、イベント駆動、アジャイル開発手法などのスキル

を持つ人材を配置する。マイクロサービス案件の推進に当たっては、素早く結果を出すと同時に、知見を蓄える。既存のアプリケーションに対しては、サービスAPI化の箇所と機会を探り、API化を開始する。外部向けに公開するAPIに関しては、APIを商品として取り扱う「プロダクト管理者」を設ける」。

180日以内に実行すべき事項はこうだ。「試行の結果を勘案し、ビジョンを適宜調整する」。

マイクロサービスという 新標準が導く世界

日本 IBM の 2017 年クラウド戦略 -- マイクロサービス開拓へ

日本 IBM は 4 月 6 日、2017 年のクラウド戦略について記者説明会を開催した。日本 IBM の取締役専務執行役員で、IBM クラウド事業本部長を務める三澤智光氏は、先日米ラスベガスで開催したイベント「IBM InterConnect」で、IBM の最高経営責任者（CEO）、Ginni Rometty 氏が打ち出した全体メッセージが「IBM のあらゆるサービスのコアにコグニティブを組み込む」だったことを紹介し、それを背景にしたクラウド戦略を説明した。

三澤氏が強調したクラウド戦略のキーワードは「Lift&Shift」。新たな収益を得るなどの目的で「新たに構築するシステム」、以前からあるものの処理能力やセキュリティ面を考慮して「変えた方がいいシステム」、問題がなく「変えなくていいシステム」の 3 つに分けられるとする。「安定的に稼働しているシステムをわざわざ変えなくてもいいという判断をするケースは十分にある」と三澤氏は指摘した。

その上で、新しく構築するシステム、変えていくシステムを考えると先に前提となる技術としてマイクロサービスアーキテクチャを挙げた。マイクロサービスは、さまざまな機能を API 経由で組み合わせ、すばやく新たなアプリケーションを構築するための仕組みだ。Uber などが、新たな枠組みでタクシー業に必要な機能をすばやく組み合わせ、短期間で「破壊的なサービス」をつくった例などが引き合いに出されることが多い。

日本 IBM は、そのマイクロサービスを活用する企業アプリケーション構築において、クラウドネイティブ環境での高可用性の確保、ビジネス単位での実装、継



日本 IBM の取締役専務執行役員で、IBM クラウド事業本部長を務める三澤智光氏

続的な改善を実現する開発ライフサイクル全体の管理を主な要件として説明する。

説明会では、マイクロサービスの開発プラットフォームとして、Bluemix を中心に IBM が考案したアプリケーション開発手法「IBM Bluemix Garage Method」を発表。クラウドネイティブのアプリケーション開発で求められる開発、テスト、導入、運用というプロセスを、DevOps の手法を用いて実践できるようにする。

さらに、Bluemix Garage Method での開発にあたって利用するツールとして、IBM が提供するものと、顧客が使い慣れたものを組み合わせ利用できる「Open Toolchain」も併せて発表した。自由度の高い開発体制の整備や開発チームごとのプロジェクト管理、複数チームによる大規模なクラウドネイティブアプリ

マイクロサービスという 新標準が導く世界

ケーション開発、各種ツールの連携による開発ワークフローの自動化や可視化、ガバナンスの向上を図れるという。

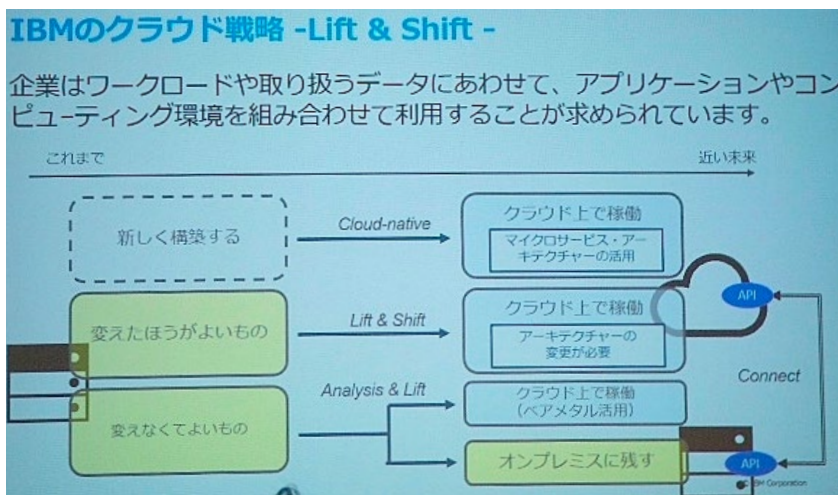
このほか、アプリケーションをさまざまなインフラで利用する際に、一括管理できるようにするサービスとして「IBM Cloud Automation Manager」(CAM)を、Bluemixの無料サービスとオンプレミス向け製品の両面で提供することも明らかにした。CAMは、クラウド基盤でアプリケーション環境を展開するための「IBM PureApplication」と運用管理ツール「Tivoli」をクラウド向けに書き換えたものだ。

このほか、ブロックチェーンの商用利用を進めるため、「Hyperledger Fabric v1.0」など各種機能やツ

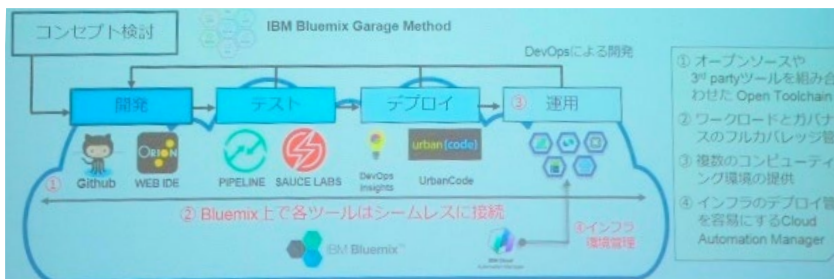
ルを Bluemix 上で提供し始めたことも紹介。オープンソースソフトウェアを最大限に活用するために提供するハイエンド機である「LinuxONE」上で稼働するシステムを、月額 160 万円で提供する。

日本 IBM は 2 月、クラウドとコグニティブシステムを組み合わせたソフトウェアやサービスを提供する専任組織を立ち上げており、クラウドネイティブなアプリケーション提供に注力するとしている。

三澤氏は、American Airlines や Royal Bank of Canada など、IaaS を超えて PaaS 領域まで踏み込んでシステムを構築した例に触れる一方で、「マイクロサービスなどの先進的な技術を全面導入するようなケースは、日本ではまだない」と話している。



Lift&Shiftのイメージ



IBM Bluemix Garage Methodのイメージ

マイクロサービスという 新標準が導く世界

マイクロサービス導入による破滅を避けるための 8 つのルール

多くの組織は、マイクロサービス（粒度の小さい、再利用可能な機能のユニット）を、一枚岩の反応の遅いシステムを分割して、より敏捷性の高いものに変えるための希望の光のように考えている。しかし、ビジネスで用いられるほかのテクノロジーと同じで、マイクロサービスは万能薬ではないし、一つ間違えば事態は簡単に悪化する。要件や変更の要求が積み上がるにつれ、ソフトウェアの提供が滞り、スケジュールが期待されていた範囲から外れる可能性は十分にある。特に、多くのことが手動で進められている開発組織ではそうなりがちだ。

最近投稿された 2 つの記事は、安易にマイクロサービスの開発と導入に手を染めることを戒めている。1 つ目の記事は、Asurion のエンタープライズアーキテクチャおよび戦略担当主席アーキテクトである Vijay Alagarasan 氏のもので、もう 1 つの記事は IT 戦略コンサルタントの Tony Bradley 氏のものだ。

マイクロサービスとは何だろうか。Alagarasan 氏によれば、マイクロサービスとは「分散した、機能ドメインとして整理されている業務上の機能をサポートする、粒度の細かいサービスを構築することにより、サービス指向アーキテクチャ（SOA）を提供するアプローチ」だ。要するに、これはよりスケーラブルな SOA だと言える。Bradley 氏は、マイクロサービスと DevOps の関係性を指摘し、これらを「DevOps とコンテナを包含し、アプリケーションを小さな個別のコンポーネントに分割する、ソフトウェア開発と導入の革命」だと呼んでいる。

これらのことを念頭に、この記事では、マイクロサービスを基礎としたプロジェクトの計画と実施に役立つガイドラインを紹介する。経験を積んだ IT 業界人なら、過去のテクノロジーで言われていたこととの共通点を見いだすかも知れない。万有引力の法則のように、ずっと残る続ける真実というものはあるのだ。

1. IT 部門だけでなく、事業部門にも価値を提供する。

前述の通り、これは正しいことが実証されているルールだ。このルールは長年の間に登場した多くのテクノロジーでその正しさが証明されており、マイクロサービスのプロジェクトが正当化できるものかどうかを検討するのにも有効だ。「マイクロサービスに対するアプローチは、組織に恩恵をもたらすものでなければならない。価値をもたらし、業務上の目的とよく噛み合うものでなくてはならず、ただマイクロサービスの流行に乗るためのものであってはならない」と Bradley 氏は述べている。

2. 現在と将来のマネージャーと開発者に指針を示す。

サイズが「マイクロ」であっても、これらのサービスはほかの大型ソフトウェアプロジェクトと同水準のマニュアルとサポートを必要とする。

3. ガバナンスの体制を整備する。

テクノロジーを基盤としたほかの取り組みと同じで、しっかりしたガバナンスがなければ、マイクロサービスの取り組みは簡単に軌道を外れてしまう。Alagarasan 氏は、「これを予防するには、サービスに関連のないビジネス機能が追加されないよう、ガバナンスを行う必

マイクロサービスという 新標準が導く世界

要がある。サービスとビジネス機能は明確に対応して
いなくてはならず、その境界線から外れたものを追加
すべきではない」と述べている。さもなければ、サー
ビスは「緊密に結合したアーキテクチャになってしま
い、提供時のエンтроピーは増大し、結合の強さが混
乱を生み出すことになる」と同氏は警告している。

4. 早めに、多くのものを自動化する。

「マイクロサービスを成功させようとする取り組みに
は、多くの『可動部分』が存在する」とBradley氏は言う。
また Alagarasan 氏は、マイクロサービスに取り組む
企業には、最低でも「テストと導入を自動的に行う手段」
が必要だと警告している。「マイクロサービスは、必要
な変化に対応できるだけの敏捷性の獲得を目指すもの
だ。品質管理のためには、各サービスがユニットテスト、
機能テスト、セキュリティテスト、性能テストを自動
で行える必要がある」（Alagarasan 氏）

5. サービスの分割は、技術的な要因ではなく、業務 上の機能をもとに行う。

これは、技術的な結合性を理由にサービスを融合させ
ることは避け、互いに独立して機能するというマイク
ロサービスの能力を維持するということだ。「複数の、
技術的または物理的なサービスレイヤを作ることは、
提供時の複雑さとランタイム時の非効率性を生むだけ
だ」と Alagarasan 氏は言う。「サービスは、求めら
れている単一の業務上の機能を実現するために必要な
ものすべてを実装した、ビジネスエンティティの最小
単位として捉えるべきだ。自己完結したサービスは、
レイヤ化されたサービスよりも自律性とスケーラビリ
ティが高い」

6. あらゆるものをテストする。

前述の自律性ととも、「導入するすべてのサービスは、
テストのための機能一式を持っていなければならない。
これは、すべてのサービス機能、セキュリティ、性能、
エラー処理のテスト、および現在および未来のすべて
の消費者の視点から実施する消費者主導テストの、す
べてをカバーしている必要がある。これは、自動化さ
れた帰帰的テストのビルドパイプラインの一部として
含まれていなくてはならない」と Alagarasan 氏は述
べている。

7. バージョン管理の掌握。

Alagarasan 氏は、サービスにバージョンが1つしか
ないと仮定すべきではないと述べている。サービスが
導入されると、エンドユーザーは変更を求めるもので
あり、機能を追加した新たなリリースが必要になる。
「一部の企業は、愚かにもバージョンを増やすのを避け
ようとしている」と同氏は指摘する。「サービスは、変
化は避けられないという前提の下に設計されるべきだ。
前方互換性を確保した上でサービスの変更を管理でき、
消費者が適切にアップグレードを行えるような戦略を
立てなければならない。さもなければ、消費者は1つ
のバージョンのサービスに縛り付けられることになり、
変化が起これば破綻してしまう」（Alagarasan 氏）

8. 取り組みを継続する。

Bradley 氏は、「一旦マイクロサービスの利用を選択
したら、元のやり方に戻すのは極めて困難だ。マイク
ロサービスを取り入れることで起こる、文化、ツール、
そしてプロセスの変化は、簡単に元に戻せるものでは
ない」として、一度始めたら止めることはできないと
述べている。