

オープンな産業オートメーション向けのシステム・オーケストレーション

ARC ホワイトペーパー 2024 年 8 月

Open Process Automation Forum は OASIS TOSCA 標準に基づくシステム・オーケストレーション O-PAS 仕様の開発を計画していますが、市場では標準化されていない商用ソフトウェアツールが好まれる傾向が見られます。このレポートでは、産業オートメーションのサプライヤーとエンドユーザーの選択肢について検討し、両者への推奨事項を提案します。

Harry Forbes
オートメーション、リサーチディレクター
ARC Advisory Group

目次

概要	3
Open Process Automation システムのシステム・オーケストレーション	5
産業オートメーションにおけるオーケストレーション	8
オーケストレーション向けの商用ソフトウェアツール	9
商用オーケストレーション市場の状況	12
オーケストレーションとプラットフォーム・エンジニアリング	15
推奨事項.....	18

システム・オーケストレーションの定義

分散システムやサービスの自動化された構成、デプロイ、調整、インテグレーション、管理。業務上の特定の目標を達成するため、多様なハードウェアやソフトウェアのコンポーネントおよびサービスを連携させます。システム・オーケストレーションはシステムのライフサイクル全体にわたって実行され、時間の経過とともに生じるソフトウェア、構成、利用可能なリソースの変化に容易に対応できる必要があります。

概要

Open Process Automation Forum (OPAF) は先頃、システム・オーケストレーションの O-PAS 仕様を OASIS TOSCA 標準に準拠して構築する意向を発表しました。オープンかつ異種混合の産業オートメーション・システムの状況では、「システム・オーケストレーション」には非常に幅広い意味があります。分散システムやサービスの自動化された構成、デプロイ、調整、インテグレーション、管理などが含まれます。先進的なクラウドネイティブ・アプリケーション・アーキテクチャで複雑なソフトウェア環境を適切に管理するにはシステム・オーケストレーションが欠かせません。こうしたアーキテクチャが産業オートメーションに導入されるにつれ、シス

システム・オーケストレーションは現在のクラウドネイティブなエンタープライズ向けアプリケーションに不可欠です。今後のオープンな産業オートメーション・システムも現在と同様の懸念事項の多くに対処しなければならないため、将来的にもきわめて重要なものとなるでしょう。

テム・オーケストレーションは今後の産業オートメーション・システムにとってもきわめて重要になるでしょう。

産業オートメーションの分野では、先進的なエンドユーザーとサプライヤーが、一部の製品やオープン・オートメーションのテストベッドおよびトライアル・アプリケーションに含まれる商用オーケストレーション・ソフトウェアツールを導入し始めています。2024 年の ARC

Forum で、エンドユーザーとサプライヤーの双方が、そうしたツールを産業オートメーション環境で効果的に使用することで得られる膨大な潜在価値と改善効果を報告しています。

IT とクラウド・コンピューティングの分野では過去 20 年にわたり、多くのソフトウェアツールが開発、市販され、オーケストレーション機能を提供してきました。このようなツールはオープンソースから生まれ、そのいくつかは現在、商用サプライヤーによってサポートされています。大企業で大きなインストールベースを持つツールもあります。そうしたツールには大規模で活発なエンドユーザーのコミュニティもあります。それらは単一の標準ではなく、ドメイン固有言語 (DSL) に対応しています。これと同時期に、ベンダーに依存しない TOSCA の仕様が学術研究機関で採用され、通信業界の一部のプロプライエタリー・ソフトウェアに導入されたと言われていましたが、商用市場にはほとんど影響を及ぼしませんでした。

ここ 10 年では、コンテナ・ソフトウェア・オーケストレーターである Kubernetes のリリースと、Cloud Native Computing Foundation (CNCF) の驚異的な成長を受けて、何百もの企業やソフトウェア製品や、数十万に及ぶコントリビューターを含む広大なソフトウェア・エコシステムの誕生が促進されました。CNCF は、データセンターをはじめエッジコンピューティングや AI などの先進的分野にまでその影響力を拡大しようとしています。

クラウドネイティブなアプリケーションが複雑さを増したことで、プラットフォーム・エンジニアリングの構想が勢いを増しています。この考え方では、インフラストラクチャとサービスの管理はプラットフォーム・エンジニアに委ねられる一方、アプリケーション開発はセルフサービスの内部開発者プラットフォーム (IDP) で行われます。その目的はソフトウェア開発者にかかる負荷を削減し、ポリシーやセキュリティのコンプライアンスを維持しつつ、開発スピードを上げることです。数多くの IT ソフトウェア製品が、製品規模の大小を問わずプラットフォーム・エンジニアリングの方向に進んでいますが、これはまだ先進的な分野であり完全に定着しているとは言えません。

ARC の見解によると、プラットフォーム・エンジニアリングの市場環境は現在、あまりにも広大で変化が激しく、その環境 (Kubernetes) に既存のコンプライアンス・プログラムはあるものの、ベンダーに依存しない国際的な基準から価値を引き出すには至っていません。現在は IT での取り組みが主ですが、プラットフォーム・エンジニアリングの概念は OT と産業オートメーションの分野に関係が深いと言えます。OT のサプライヤーとエンドユーザーはシステムとソフトウェアの構成に影響を与える多くの一般的な業務プロセスを考慮し、この運用環境について計画を立てる必要があります。

商用オーケストレーション・ソフトウェアは、国際標準に準拠していないものの、依然として現在の IT 市場を支配しています。TOSCA はベンダーに依存しない公開された標準ですが、商用製品に比べてコミュニティからのサポートが大幅に不

オープンオートメーションの支持者は、単一の標準を導入できないとしても、既存の大きな IT コミュニティ、人材プール、主要な商用製品で提供される文書化されたベストプラクティスを活用できます。

足しています。より広範な IT 市場で TOSCA が導入されていない現状では、OT のイニシアチブでこの標準の導入が進むと考えることは現実的ではありません。オープンオートメーションの支持者は既存の大きな IT コミュニティ、人材プール、主要な商用製品で提供される文書化さ

れたベストプラクティスを活用する必要があります。これにより単一の標準は導入できないものの、OPAF が O-PAS システムのオーケストレーションのユースケースを慎重に定義することに取り組み、OT サプライヤーと産業オートメーションのエンドユーザーの両方が好む商用ソフトウェアとともに導入できるようになる可能性があります。

Open Process Automation システムのシステム・オーケストレーション

2024 年 2 月、Open Group による Open Process Automation Forum (OPAF) は O-PAS 標準パート 9 の概要を公開しました。この概要では「システム・オーケストレーション」を取り上げています。O-PAS 標準に正式に組み入れられてはいないものの、この予備文書の目的は「システム・オーケストレーション開発における方向性を示し、その道筋について検討すること」とされています。

この文書では、オープンな産業オートメーション・システムのシステム・オーケストレーションの価値について以下のように述べています。

「この文書では、自動化システムのさまざまなコンポーネントを記述、デプロイ、管理するための標準に基づく統一されたアプローチを定義します。このアプローチにより、一貫したインテグレーション、相互運用性、スケーラビリティを実現します。多様な自動化コンポーネントの管理と調整は、特に種類の異なるシステムやテクノロジーが混在する状況では、困難な課題となることがあります。この複雑さに対処するため、Open Process Automation Forum (OPAF) は Topology and Orchestration Specification for Cloud Applications (TOSCA) 標準を使用して、産業プロセス・オートメーション・システムのオーケストレーションのフレームワークを定義することを選択しました」

```
---
- hosts: all
  become: true
  vars:
    created_username: sammy

  tasks:
    - name: Install aptitude
      apt:
        name: aptitude
        state: latest
        update_cache: true

    - name: Setup passwordless sudo
      lineinfile:
        path: /etc/sudoers
        state: present
        regexp: '^%sudo'
        line: '%sudo ALL=(ALL) NOPASSWD: ALL'
        validate: '/usr/sbin/visudo -cf %s'

    - name: Update apt and install required system package:
      apt:
        pkg:
          - curl
          - vim
          - git
          - ufw
        state: latest
        update_cache: true

    - name: UFW - Allow SSH connections
      community.general.ufw:
        rule: allow
        name: OpenSSH

    - name: UFW - Enable and deny by default
      community.general.ufw:
        state: enabled
        default: deny
```

YAML は人間が判読できるデータシリアライズ言語

TOSCA は、オープンスタンダードおよびオープンソースソフトウェアのための非営利コンソーシアムである OASIS (構造化情報標準促進協会) によって開発された標準です。TOSCA 1.0 は 2014 年初頭に公開されました。改訂により、サポート対象が XML だけでなく、より広く使用されている YAML (人間が判読できるデータシリアライズ言語で、一般に設定ファイルに使用される。図を参照) も対象となりました。最新版 (1.3) はさらにバージョン 2.0 に改訂され、2024 年 6 月にコメント募集用に公開されました。

システム・オーケストレーションとは

オープンな産業オートメーション・システムは、広義の分散システムの一例です。オーケストレーションの手法は、クラウド・コンピューティング、マイクロサービス、DevSecOps といった IT 分散システムの分野で最も発達しています。分散システムのコンテキストでは、システム・オーケストレーションは以下のように定義されます。

分散システムやサービスの自動化された構成、デプロイ、調整、インテグレーション、管理。業務上の特定の目標を達成するため、多様なハードウェアやソフトウェアのコンポーネントおよびサービスを連携させます。システム・オーケストレーションはシステムのライフサイクル全体にわたって実行し、時間の経過とともに生じるソフトウェア、構成、利用可能なリソースの変化に容易に対応できる必要があります。

この定義にはハードウェアも含まれるものの、分散システムの動的な性質は主にソフトウェアに由来します。クラウド・コンピューティングでは、ハイパースケーラー (AWS、Microsoft Azure、Google Cloud) は必要に応じてほぼ任意の量のハードウェアリソース (コンピューティング、ストレージ、ネットワーク) を追加できます。これはオンプレミス・コンピューティングやエッジコンピューティングでは不可能です。オンプレミスおよび (特に) エッジコンピューティングでは、アプリケーションや対応するソフトウェアを改訂すると、分散システムの大半を変更する必要があります。

以下はシステム・オーケストレーションのいくつかの主な側面です。

自動化: ソフトウェア・コンポーネントをデプロイ、管理、スケーリングするために必要な反復的タスクを自動化することで、ヒューマンエラーを防ぎ、デプロイメント全体で確実に一貫性を保つ必要があります。これらのタスクには以下のようなものがあります。

- **プロビジョニング:** コンピューティング、ストレージ、ネットワークに必要なリソースをセットアップする。
- **構成管理:** すべてのソフトウェア・コンポーネントが定義された基準やポリシーに従って適切に構成されるようにする。ソフトウェアリポジトリの管理、バージョン管理、セキュリティなどが含まれる。
- **デプロイメント:** アプリケーションを開発環境またはリポジトリから本番環境に移動する。
- **スケーリング:** 変化するワークロードに対応してリソースを調整することで、可用性とパフォーマンスを維持する。

調整: オーケストレーションによって、さまざまなソフトウェア・コンポーネントとサービス間のインタラクションをライフサイクル全体にわたって調整し、適切に連携するようにします。以下に例を示します。

- **依存関係:** コンポーネントやサービス間の依存関係を管理して、要求された順序で構築、デプロイ、起動、停止、スケーリングされるようにする。
- **ワークフロー管理:** ソフトウェアプロセスやワークフローでタスクの実行が要求される順番を定義して管理する。
- **状態管理:** サービスとワークフローの状態を追跡して管理する。

インテグレーション: 個別のソフトウェア・コンポーネントを、開発パスやデプロイメントにかかわらず連携させます。これには以下が含まれます。

- **API:** 公開されたアプリケーション・プログラミング・インタフェース (API) を使用して異なるサービス間で通信し、データをやりとりする。
- **イベント対応:** イベントがアクションをトリガーしてシステムに影響を及ぼすイベントへの対応でサービスを調整できるようにする。

監視と管理: 最善のパフォーマンスと信頼性を確実に維持します。これには以下が含まれます。

- **健全性の維持:** サービス、アプリケーション、コンポーネントの健全性とパフォーマンスを確認および維持する。
- **ログとトレース:** ログとトレースをキャプチャして分析する。
- **アラート:** システムに問題や異常が発生したときにアラートを作成する。

効果的なオーケストレーションには、次のような大きなメリットがあります。

- **エラーの削減:** 手動の管理タスクを自動化することで、反復タスクでのヒューマンエラーの発生率を低減できる。
- **効率性:** 手動での介入を削減することで、デプロイメントプロセスを迅速化する。
- **一貫性:** さまざまな実行環境にわたるアプリケーションとサービスを、事前に定義された標準とポリシーに従い、一貫性を保ってデプロイおよび管理できる。
- **スケーラビリティ:** 手動で介入することなくサービスをスケールアップまたはスケールダウンできる。
- **堅牢性:** リカバリーとスケーリングを自動化することで、システムの信頼性を向上させる。

先進的なクラウドネイティブ・アプリケーション・アーキテクチャで複雑なソフトウェア環境を適切に管理するには、システム・オーケストレーションが欠かせません。オーケストレーションにより、デプロイされたすべてのコンポーネントとサービスが効率的かつ効果的に連動され、運用の合理化、運用タスクの単純化、システムパフォーマンスの向上を実現できます。アプリケーションのポートフォリオにハイブリッドクラウド・モデルを導入している IT 組織は、ハイパースケーラーとオンプレミスの IT インフラストラクチャのいずれかに、柔軟かつ迅速にアプリケーションを再デプロイする必要があります。このような機能を提供する上でオーケストレーションは欠かせません。

産業オートメーションにおけるオーケストレーション

従来のプロプライエタリーでアプライアンスベースの産業自動化システム (PLC、DCS) では、オーケストレーションの導入は最小限に留まり、通常はほとんどオーケストレーションが行われず、独自のソフトウェアツール経由で行われています。し

ある大手エンドユーザーは「インフラストラクチャの自動化への投資によるライフサイクルの変更やコスト削減、そしてそのライフサイクルを自動化できるということが本当に素晴らしい部分です。多くの人が大きな衝撃を受けるでしょう」と語っています。

かし、将来のオープンな産業オートメーション・システムには、新しいソフトウェアの迅速なデプロイ、プラットフォーム独立性、仮想化やコンテナ化などのより先進的なソフトウェア・テクノロジーの使用が想定されます。この構想を達成するには包括的で効果的なオーケストレーションが必要です。現在、O-PAS などの先進的でオープンなオートメーション・システムの導入実績は限

られています。にもかかわらず、2024 年の ARC Industry Leadership Forum のレポートによれば、エンドユーザーとサプライヤーはいずれも、O-PAS システムのオーケストレーションに高い期待を寄せています。

オープンオートメーションのオーケストレーションからエンドユーザーが得られる価値

2024 年の ARC Forum ではサプライヤーとエンドユーザーの両方が、オーケストレーションは新しいオープンな産業オートメーション・アーキテクチャに大きな価値を提供できると報告しました。以下はその一部の例です。

迅速なオンボーディング: デプロイの自動化によって、これまで数時間かかっていた新しいコントローラーやサーバーのデプロイを、さまざまなレベルのフォールトトレランスで数分間でできるようになります。これにより新しいアプリケーションを迅速にデプロイでき、価値実現までの時間を大幅に短縮できます。

スケーラビリティとソフトウェア・ライフサイクル管理: 自動化アプリケーションのソフトウェア・マニフェストを記載した **Playbook** を繰り返し使用してシステムをスケールアップすることで、完全に一貫したプラットフォームを提供できます。オーケストレーションにより、**Playbook** を変更するとその変更がインストール全体に伝播し、システム全体で企業のセキュリティとサービスの標準を満たした状態を維持できます。ある大手エンドユーザーは「インフラストラクチャの自動化への投資によるライフサイクルの変更やコスト削減、そしてそのライフサイクルを自動化できるということが本当にすばらしい部分です。多くの人が大きな衝撃を受けるでしょう」と語っています。

デプロイメントの標準化: 構成のベストプラクティスを製造現場全体および製造組織全体で簡単に共有できます。先述のエンドユーザーは「私たちが重点的に取り組んでいるのはソフトウェアの機能と、このオープンなシステム・アーキテクチャを備えたソフトウェアによる問題解決、さらにソフトウェアのデプロイを自動化してビジネス目標の達成のためにシステムを変更する機能です」と語っています。

産業オートメーションのサプライヤーは同種の価値を実現し、既存の製品にこうした機能を付加しています。特に、ABB、Rockwell Automation、Schneider Electric は Red Hat と提携し、プロプライエタリーなテクノロジーの代わりにオーケストレーションと IT 自動化ソフトウェアを使用して高可用性を実現し、手間が多くエラーが発生しがちなデプロイメントとサービスの作業を自動化しています。

オーケストレーション向けの商用ソフトウェアツール

分散したアプリケーションのデプロイを完全に自動化することは、分散システムやクラウド・コンピューティングの可能性を引き出す上で不可欠です。いくつかの商用ソフトウェア・デプロイメント製品 (Ansible、Chef、Puppet、Terraform、Kubernetes の商用ディストリビューションなど) がリリースされ、デプロイメントの自動化に対するエンドユーザーのさまざまなニーズに応えています。こうした製品のほとんどは宣言型の構成を採用しています (下のサイドバーを参照)。また、通常は YAML で表現されるドメイン固有言語 (DSL) を備えています。DSL は、アプリケーション・コンポーネントとその関係や構成を構造化して記述することによって、システムの望ましい状態を定義するのに使用されます。デプロイメントの自動化を達成するには、こうしたアプリケーション仕様を処理して、要求される操作を自動的に導きだして要求された順序で実行し、(宣言された) 望ましいシステム状態に到達して維持します。

IT のプラクティスではしばしば、複雑な分散アプリケーションのデプロイでさまざまな商用デプロイメント製品を組み合わせる必要があります。多数の商用デプロイメント自動化製品が、さまざまなインフラストラクチャ・コンポーネントまたはクラウドサービスへのアプリケーションのデプロイに特化した機能を提供しています。Terraform はクラウド・インフラストラクチャと大手ハイパースケーラー・プラットフォームのプロビジョニングに注力しており、CloudFormation は AWS のリソースのみを管理しています。Ansible はソフトウェア・コンポーネントの構成管理で広く評価されています。このため、IT 部門の担当者は、エンタープライズ IT のデプロイメントに、複数のデプロイメント自動化ソフトウェアツールを導入することがよくあります。たとえば、IT のエンドユーザーが特定のハイパースケーラーのプラットフォームで VM をプロビジョニングするのに Terraform を使用し、VM で実行される多くのソフトウェア・コンポーネントの特定と管理には Ansible を使用するといったことが起きます。

その結果、ほとんどの IT 部門がアプリケーションのデプロイに複数のデプロイメントツールやモデルを使用し、各モデルではデプロイ全体の一部がさまざまな DSL を使用して記述されることになります。これには複数の商用製品の専門知識が必要となるため、欠点となりかねません。

しかし商用製品のサプライヤーは広範な事前設定済みのソリューション・テンプレートも提供しており (たとえば Ansible Galaxy の Web サイトでは Ansible のエンドユーザーが Ansible の大規模なエンドユーザー・コミュニティで貢献内容を検索したり共有したりできます)、デプロイメントモデルの作成と再利用が大幅に単純化されます。

現在のところ、1 つですべてのニーズに応える万能のデプロイメント自動化製品はありません。任意のアプリケーション・コンポーネントを任意のコンピューティング環境やプラットフォームにデプロイするのを支援する商用製品はありません。これは TOSCA 標準が対処しようとしている問題ですが、公開以来 10 年の間に、商用市場は違う方向へと向かっています。

宣言型の構成とは

宣言型と命令型の構成は、特にクラウドネイティブのソフトウェアと **Infrastructure-as-code (IaC)** のコンテキストで、システムの状態の管理と定義に対する異なる 2 つのアプローチを表します。実際は、現在 IT とインフラストラクチャ・ソフトウェア・ツールには宣言型の構成が想定されています。

宣言型の構成: 宣言型の構成では、システムの望ましい状態を指定しますが、その状態を達成するために必要な手順は指定しません。結果に至る方法ではなく、結果の「内容」を重視します。たとえば宣言型の構成で、特定の仮想マシンの 3 つのインスタンスが、特定のクラウド・プラットフォームに特定の属性 (CPU、メモリー、ネットワーク設定など) を持って存在すると指定しても、それらを作成するための一連のコマンドを詳細に指定することはありません。この望ましい状態を実現し、維持するために必要なアクションを実行するのは、オーケストレーターまたはプロビジョニング・ソフトウェア・ツールの責任です。

命令型の構成: 命令型の構成では、特定の状態を達成するために必要なコマンドと手順を指定します。この手法ではタスクを達成する「方法」を重視するため、必要なアクションと順序についての詳細な指示を提供します。たとえば、サーバーをセットアップするための命令型のスクリプトには、ソフトウェア・パッケージのインストール、ネットワーク設定の構成、サービスの起動の具体的なコマンドが含まれます。

宣言型の構成は最終的な状態に重点を置く一方、命令型の構成はその状態を実現する方法に重点を置きます。宣言型メソッドはエンドユーザーからの明示的な手順の指示を抽象化しますが、特定の手順が必要な場合、関連するソフトウェアツールがそれを実行できる必要があります。

宣言型の構成は望ましい結果のみを記述するため、よりシンプルに作成でき、理解もより簡単です。命令型の構成はプロセスの各手順を非常に詳細に制御できますが、結果として複雑になり、維持するのが困難です。

実際、プロビジョニング、デプロイメント、構成管理、オーケストレーションの先進的なソフトウェアツール (Ansible、Kubernetes、Terraform など) はすべて、ツールの複雑化を避け、より使いやすい宣言型構成をサポートしています。

商用オーケストレーション市場の状況

Kubernetes の台頭と CNCF のエコシステム

過去 10 年で最もインパクトのあったオープンソースソフトウェア・プロジェクトは、間違いなく Kubernetes でしょう (略称は K8s)。Kubernetes はコンテナ化されたソフトウェア・アプリケーションのオーケストレーターで、当初 Google の社内で開発されていました。オープンソースとして 2014 年に最初にリリースされてから 1 年後、12 社が Google と共同で Cloud Native Computing Foundation (CNCF) を結成しました。そのミッションステートメントは「クラウド・コンピューティングをユビキタスに」でした。

2014 年に導入されてから 2 年のうちに、Kubernetes は競合プロジェクトを抑えて主要なコンテナ・オーケストレーターとなりました。以来、CNCF のメンバーは 450 社以上に増え、Kubernetes のほかにおよそ 200 件のオープンソース・プロジェクトのキュレーションを行っています。これらのプロジェクトには 20 万人以上のコントリビューターが参加し、現在では CNCF のプロジェクトと組織は IT 業界で大きな影響力を持っています。広範で今も拡大している CNCF のエコシステムの大半は、Kubernetes をサポートまたは強化するプロジェクトから成り立っています。その一例が、Kubernetes で仮想マシンとコンテナを定義および管理できるようにする Kubevirt プロジェクトです。

190 の CNCF プロジェクトは、主に次のようなカテゴリに分類されます。

- プロビジョニング
- ランタイム
- オーケストレーションと管理
- アプリケーションの定義と開発
- 可観測性と分析
- Kubernetes プラットフォーム

CNCF には Kubernetes 向けのコンプライアンス認証プログラムがあります。現在、このコンプライアンステストに合格した商用製品は 170 以上あります。これらの製品のほとんどは Kubernetes と追加機能を組み合わせたものです。たとえば CI/CD ツール、可観測性、リポジトリ管理、バージョン管理、コスト最適化、エッジデプロイメントなどです。

商用オーケストレーション・ツールの比較

巨大なエコシステム、開発者のマインドシェア、多数の商用ソフトウェア製品への採用などにより、Kubernetes はオーケストレーション分野で無視できない存在となっています。しかし、Kubernetes は依然としてサーバーでホストされるコンテナ化されたワークロードを対象としています。CNCF のメンバー企業はオーケストレーション、VM の管理、(特に) エッジコンピューティングへの Kubernetes のさらなる拡大を優先しました。いくつかの商用製品はエッジコンピューティング環境とワークロードのサポートに特化して設計されています。CNCF が最近最も注力しているのが AI ワークロードへの対応を向上させることです。

オーケストレーション・ソフトウェア・ツール	対応するサプライヤー	ソースコードのライセンス	コントリビューターの数 (GitHub)
Ansible	Red Hat	GPL-3.0	5,558
Chef	Progress	Apache-2.1	679
Puppet	Perforce	Apache-2.0	591
Terraform	Hashicorp	BSL 1.1	1,796
xOpera	なし	Apache-2.0	11

商用オーケストレーション・ソフトウェア・ツールの特性

Kubernetes のほかにも、デプロイメント自動化とオーケストレーション向けにいくつかのオープンソース・ツールが市販され、現在 IT 業務に広く使用されています。

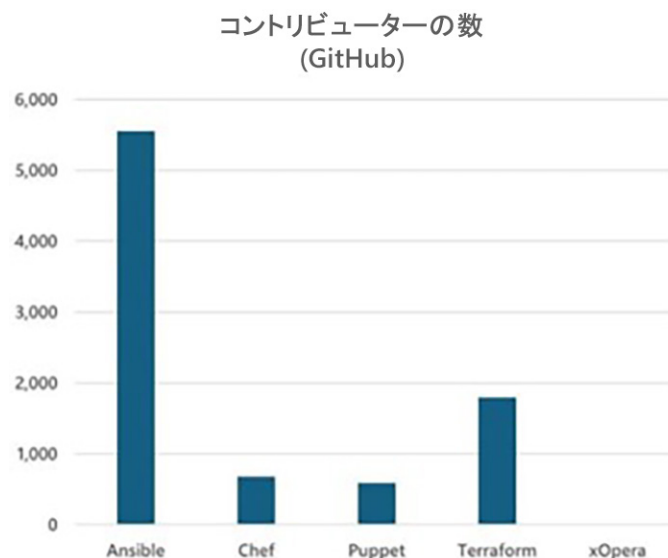
Ansible: 現在、Red Hat Ansible Automation Platform として Red Hat により市販されているインフラストラクチャ自動化ツールで、ストレージ、サーバー、ネットワーク、アプリケーション、ワークフローのプロビジョニングと構成を管理します。2015 年に Red Hat が買収した Ansible の大規模なエンドユーザー・コミュニティは、54,000 を超えるプロジェクトのコミット、Ansible プロジェクトへの 5,000 人のコントリビューター、31,000 の依存するリポジトリを抱え、活発に活動しています。

Chef: 2020 年に Progress Software によって買収され、Progress Chef として商用化された構成管理ツールです。Progress Chef には 1,000 を超える顧客のインストールベースがあり、そのうち 50 社はフォーチュン 500 企業であるとしています。

Puppet: Puppet はシステム構成を記述する独自の DSL を含むソフトウェア構成管理ツールです。設立は 2005 年で、長年ベンチャー企業として資金を調達してきましたが、2022 年に Perforce Software に売却され、現在は Perforce Puppet として商用化されています。Puppet はオープンコアのビジネスモデルを使用しています。無料製品は Apache Software License のもとでリリースされ、商用サポート対象のバージョンである Puppet Enterprise はプロプライエタリー・ライセンスで販売されています。

Terraform : HashiCorp が作成したソフトウェアツールです。ユーザーは HashiCorp Configuration Language (HCL) という宣言型 DSL を使用して、データセンター・インフラストラクチャを定義およびプロビジョニングします。Terraform は、Terraform を拡張して主要なクラウドサービス (AWS、Azure、GCP など) の API を利用する「プロバイダー」経由で、エンドユーザーのパブリッククラウド・インフラストラクチャを管理します。2024 年に IBM が HashiCorp の買収に合意し、2024 年末には手続きが完了する予定であることを発表しました。

TOSCA: 上記のテクノロジーでは、ツール固有のデプロイメントモデル言語を採用しています。同じ時期に、OASIS TOSCA 標準が開発され、クラウド・アプリケーションをベンダーに依存しない方法で記述する手段として公開されました。TOSCA が最初に公開されたのは Kubernetes の発表とリリースの直前でした。今日、TOSCA は学術研究に使用されているものの、IT 業務の担当者が使用する商用ソフトウェア製品ではサポートされていません。このため、学術研究と IT のプラクティスの間にギャップが生まれています。この標準を使用するオープンソースのソフトウェアツールはいくつかあるものの (xOpera が本命と言われる)、商用ソフトウェア市場ではこうしたツールの採用はごくわずかです。TOSCA の支持者は、大手通信企業が 5G ネットワークの自動化と管理に TOSCA を使用していると報告していますが、この業務では商用ソフトウェア製品でなく社内で開発されたビジネス戦略ソフトウェアが使用されています。



さまざまなオーケストレーション・ソフトウェア・ツールのコントリビューターの数
(出典: GitHub)

オーケストレーション・ツールをめぐる状況

上記の図表は、商用オーケストレーション製品と TOSCA オーケストレーターを比較したメトリックの一部を示しています。「コントリビューター」という語には、コア開発者から改善提案を 1 件提出したエンドユーザーまで、幅広いロールが含まれます。それでも、商用製品を使用するエンドユーザーのコミュニティのほうがずっと大きく、市販製品を導入して使用するユーザーにとっての大きなメリットとなっています。商用ソフトウェアツールのほうが、はるかに大きな体験ベースやコミュニティを提供でき、ベストプラクティスや実装コードを共有できます。こうしたタイプのエンドユーザー・コミュニティは、すでにツールを生産的に使いこなしている、経験豊かな人材の供給源でもあります。

オーケストレーションとプラットフォーム・エンジニアリング

IT 分野の重要な先進的コンセプトに、プラットフォーム・エンジニアリングがあります。プラットフォーム・エンジニアリングの定義は、完全でセルフサービス、自動化および統合されたソフトウェアの開発、デプロイメント、管理の機能を提供することです。



プラットフォーム・エンジニアリングの機能
(出典: Red Hat)

プラットフォーム・エンジニアリングのコミュニティの主要なコントリビューターである Luca Galante 氏は、プラットフォーム・エンジニアリングを次のように定義しています。

「クラウドネイティブの時代に、ソフトウェア・エンジニアリング組織にセルフサービス機能を実現するツールチェーンとワークフローを設計および構築する分野。プラットフォーム・エンジニアは、アプリケーションのライフサイクル全体にわたる運用上の必要性に対応する統合製品を提供します。これはよく『内部開発者プラットフォーム』と呼ばれます」

「IDP の導入が示しているのは、ツールやプロセスの変化だけではありません。組織のソフトウェア開発へのアプローチ方法におけるパラダイムシフトを表しています...ソフトウェア開発に携わる組織や開発者にとっての将来は、分散システムや新しい技術の複雑さの一部を単純化し、開発者の能力を強化するプラットフォームにあります...本当のジャーニーは、これらのプラットフォームとそれに伴うエコシステムの継続的な進化の中にあります。こうしたプラットフォームの成功は、他のあらゆるプラットフォームと同様に、開発者、プラットフォーム・エンジニア、組織のリーダーらが変化を受け入れ、変化を促進するオープンソースのコミュニティに貢献し、自社組織の開発の未来を形成できるかにかかっています」

- Red Hat「開発者ポータル」、2024 年

プラットフォーム・エンジニアリングは、大手オンラインテック企業 (Google、Meta、Lyft、Netflix、Spotify、Uber など) から生まれました。そのコンセプトは現在、あらゆる種類の企業に広がっています。その主眼は「内部開発プラットフォーム」(IDP) を提供することに置かれています。このプラットフォームの目的は、開発者が扱う複雑さと認知的負荷を削減してアプリケーションへの集中を高められるようにし、同時に (プラットフォームを介して) ソフトウェアソーシング、デプロイメント、ID、認可、セキュリティ、ネットワーク、可観測性などの定義されたポリシーと標準に準拠できるようにすることです。このプラットフォームはアプリケーション開発者が担うソフトウェアのデリバリーとデプロイメントの複雑さを削減し、その責任を、標準、リポジトリ、ポリシー、IDP を支えるソフトウェアツールの複雑なスイートを管理するプラットフォーム・エンジニアに移転します。また、このプラットフォームは企業が要求するあらゆる方法 (クラウド、ハイブリッドクラウド、オンプレミス、エッジ) でアプリケーションをデプロイする機能を備えている必要があります。

プラットフォーム・エンジニアリングの商用化の状況

プラットフォーム・エンジニアリング向けソリューションとして位置づけられている商用ソフトウェア製品は数多くありますが、このコンセプトは現在、パッケージとして市販できるものでないことをほとんどのベンダーが認めています。コンセプトは進化し続けており、既存のプラクティスと手順のコンテキストで実装する必要があるためです。こうした製品のサプライヤーは、大手ハイパースケーラーから大手または小規模の独立系ソフトウェアベンダー (ISV)、さらにベンチャー企業までさまざまです。

プラットフォーム・エンジニアリングのビジョンは現在、IT のコンセプトに基づいています。しかし、OT および産業オートメーションのサプライヤーは、これとオープンな産業オートメーション・システムとの関連性が高まっていくことに注意する必要があります。産業オートメーションのコンテキストでは、オープン・オートメーションのシステムインテグレーターや設計者は、プロセスエンジニア、さらにはプロセスユニット・オペレーターをある種の「開発者」と捉える必要があります。こうした人々の作業プロセスがシステムのソフトウェア構成に影響を与える場合があるためです。

デバイスの調整やアプリケーションの制御、生産計画への日常的に起きるありふれた変更は、適切に維持するためにアップストリームのソフトウェア構成管理に取り込む必要があります。「内部開発者プラットフォーム」のコンセプトを製造オートメーションに拡大し、システム構成に影響を与える可能性があるプロセス・オートメーション・ソフトウェアのユーザーを対象に含めてください。

「自動化」の正確な意味を理解する

IT と OT の融合が進む今、どちらの関係者も、今後は「自動化」や「オートメーション」という用語を形容詞とともに使用するべきだということを知っておく必要があります。

OT の専門用語の「オートメーション」は、製造プロセスを管理する産業オートメーション装置のアプリケーションと産業オートメーション・システム (PLC、DCS、ドライブなど) を指します。

IT の分野で「自動化」はデプロイメントの自動化、構成管理、バージョンとビルドの管理、Infrastructure as Code、CI/CD パイプライン、ビジネスプロセス自動化などを指します。

混同に気をつけましょう。IT/OT の会話をするときは、形容詞と正確な用語を使用しましょう。

おすすめのリソース

オーケストレーション・ソフトウェア・ツールの分野は今日、急速に進化しています。この進化を促進しているのは主に、クラウドネイティブのソフトウェア・アーキテクチャとプラクティスを導入しつつ、従来のアプリケーションや新しいビジネス・アプリケーションの既存のポートフォリオを継続的に進化させるという IT 分野のニーズです。IT 組織は圧倒的にサポート付き商用オーケストレーターを使用し、その多くは複数のツールをさまざまなロールで使用しています。それぞれが独自の DSL を使用する、複数のツールの習熟と管理は困難です。しかし商用ツールでは貴重なベンダーサポートやコミュニティサポートが提供されます。市場での導入が示しているのは (そして ARC が信じるところでは)、現在のオーケストレーション技術があまりに動的であり、市場があまりに未成熟であるために、完全にベンダー中立な標準を成立させることは (望ましいにしても) 不可能ということです。OT のサプライヤーとエンドユーザーはこのことを認識しておくべきでしょう。

現在のプラットフォーム・エンジニアリング は市販の製品ソリューションではありません (前述の Red Hat のドキュメントからの引用を参照)。ただし、多くの重要な商用ソフトウェア製品はその方向に向かっています。オープンオートメーション製品を設計している OT のサプライヤーは、プラットフォーム・エンジニアリングをより長期的な目標として捉え、プロセスオペレーターからサプライヤーのソフトウェア開発チームに至るまでの広範なロールを「開発者」としてサポートしてください。

オープンな産業オートメーションのエンドユーザーは、プラットフォーム・エンジニアリング・ソリューションに参入しているベンダーは非常に幅広く、そのコンセプトは重要ではあるものの比較的新しく、進化中であることを認識しておく必要があります。システム・オーケストレーションとプラットフォーム・エンジニアリングのベンダー選定基準には、ベンダーの成熟度、標準フレームワークの有無、ユーザーコミュニティのサポート、既存のツールやプラクティスとの統合の容易さなどを含めるとよいでしょう。

商用オーケストレーション・ソフトウェアは、国際標準に準拠していないものの、依然として現在の IT 市場を支配しています。TOSCA はベンダーに依存しない公開された標準ですが、商用製品に比べてサポートするコミュニティが大幅に不足しています。この市場には、CNCF の Kubernetes プラットフォーム・プログラムを除き、既存のコンプライアンス認定プロセスがありません。しかし Kubernetes は広範なオーケストレーション分野のごく一部に過ぎません。

オーケストレーションの基礎としてベンダー中立の標準を使用するという OPAF の希望は理解できます。しかし、ベンダー中立の唯一の標準である TOSCA は市場

で大きな支持を得られていません。より大きな IT 市場で TOSCA が導入されていない現状では、OT のイニシアチブで標準の導入が進むと考えることは現実的ではありません。唯一提供されているグローバル標準を利用するより、O-PAS のメンバーは既存の大きな IT コミュニティ、人材プール、主要な商用製品をサポートする文書化されたベストプラクティスを活用する必要があります。

OPAF にとってより堅実なプランは、少なくとも（ベンダー中立でなくても）オープンソースに基づき、大規模なユーザーまたは開発者のコミュニティの恩恵を受けることができ、既存の商用製品に裏打ちされたオーケストレーション・テクノロジーを採用することでしょう。

これによってベンダー中立の標準が採用されなくなっても、代わりに O-PAS は O-PAS システムのオーケストレーションのユースケースを定義することに集中できるようになります。これが正しく整理され、詳細に定義されれば、エンドユーザーとサプライヤーは、OT サプライヤーと産業オートメーションのエンドユーザーの両方が好む商用ソフトウェア製品を使用して実装できます。

ARC アナリスト: Harry Forbes

エディター: Larry O'Brien、Sharada Prahladao

頭字語の一覧:

AI	人工知能	IT	情報技術
API	アプリケーション・プログラミング・ インタフェース	K8s	Kubernetes
AWS	Amazon Web Services	OASIS	Organization for the Advancement of Structured Information 標準規格
DSL	ドメイン固有言語 パフォーマンス管理	OPAF	Open Process Automation Forum
CI/CD	継続的インテグレーション/ 継続的デプロイメント	O-PAS	Open Process Automation Standard
CNCF	Cloud Native Computing 基盤	OT	オペレーショナル・テクノロジー
CSP	クラウドサービス・プロバイダー	PLC	プログラマブル・ロジック・ コントローラー
DCS	分散制御システムDSL ドメイン 固有言語	TOSCA	Topology and Orchestration Specification for Cloud Applications
GCP	Google Cloud Platform	VM	仮想マシン
IDP	内部開発者プラットフォーム	XML	Extensible Markup Language
IaC	Infrastructure as code	YAML	Yet Another Markup Language
ISV	独立系ソフトウェアベンダー		

1986 年に創立された ARC Advisory Group は、産業、インフラストラクチャ、都市に関するテクノロジーリサーチとアドバイザリーのリーディングカンパニーです。ARC は情報技術 (IT)、オペレーショナル・テクノロジー (OT)、エンジニアリング・テクノロジー (ET)、および関連するビジネストrendに関して、他を寄せ付けない知識を持っています。当社のアナリストとコンサルタントは業界の知識と実地経験を持ち、お客様の組織が直面する複雑なビジネス課題に最善の答えを見いだせるように支援します。テクノロジー・サプライヤーのクライアントに戦略的市場調査を提供し、エンドユーザーのクライアントには適切な導入戦略を作成し、ニーズに合った最善のテクノロジー・ソリューションを評価および選択できるよう支援します。

このレポートのすべての情報の所有権と著作権は ARC に帰属します。レポートのいかなる部分も ARC による事前の許可なしに複製することはできません。この調査の一部は Red Hat が後援しています。ただし、この文書で ARC が表明した意見は ARC 独自の分析に基づきます。

ARC のアドバイザリーサービスを通じて、ARC の広範かつ継続的な調査とスタッフメンバーの経験をご利用いただけます。ARC のアドバイザリーサービスは、特に組織の戦略開発と方向性決定の責務を負う幹部の皆様向けに設計されています。会員登録に関する情報は電話、メール、Web サイトにてお問い合わせください。

ARC Advisory Group, Three Allied Drive, Dedham, MA 02026 USA • +1 781-471-1000 • www.arcweb.com

