

サービス ライフサイクルオートメーション を実現するアーキテクチャ Tail-fベースのCisco Network Services Orchestrator



目次

エグゼクティブ サマリー	3
自動化の機会	3
エンドツーエンドのサービス オークストレーションの全容	4
Cisco Network Services Orchestrator. ネットワーク	
サービスにおけるライフサイクル全体の自動化をサポート	4
1 つめの柱: データ モデルとデータ モデルのマッピング	4
2 つめの柱: ステート コンバージェンス	4
3 つめの柱: 複数のドメインにおけるオーケストレーション	5
4 つめの柱: オーケストレートされたアシュアランス	5
オンデマンドのサービス ライフサイクルのオーケストレーション向けプラットフォーム	5
Network Services Orchestrator アーキテクチャの概要	6
データ モデルとマッピング	8
正確なデータ モデルを利用したオーケストレーション	8
ステート コンバージェンス	9
サービス マッピングを利用したステート コンバージェンス	9
ステート コンバージェンスの動作	10
複数のドメインにおけるオーケストレーション	11
マルチドメインのオーケストレーションの動作: NFV MANO	11
マルチドメインのオーケストレーション動作: SDN	13
オーケストレートされたアシュアランス	14
Network Services Orchestrator でオーケストレートされたアシュアランス	14
オーケストレートされたアシュアランスの実装	15
設計時のアシュアランス	16
実行時のアシュアランス	16
Cisco Network Services Orchestrator の動作:	
サービス フルフィルメントとアシュアランスのオーケストレーション	17
アクティベーション テスト モデルの設計	18
継続的な SLA モニタリングのモデル化	19
アクティベーション テストおよびアシュアランスのインスタンス化	20
サービス アシュアランス機能のオーケストレーション	21
Network Services Orchestrator のユーザー インターフェイスで見た、アシュアランスを備えたサービス オークストレーション	22
まとめ	23

エグゼクティブ サマリー

ネットワーク サービス分野は、かつてないほど競争が激化し、ネットワーク機能仮想化 (NFV) を中心とする新テクノロジーのイノベーションにより、ネットワーク サービス プロバイダーへの顧客の期待が一際高まっています。新しいサービスをいち早く開発し、各顧客のニーズに応じてカスタマイズし、ソフトウェア並みの速度で提供できる敏捷な者が勝者となります。こうした目標を達成する決め手となるのがサービスオーケストレーションです。しかし、オペレータ環境は複雑かつ動的であることから、サービスのある一部分でオーケストレーションを実行したり、単一のネットワーク ドメイン内で機能を自動化するだけでは十分とは言えません。真の俊敏性を実現するには、サービス ライフサイクル全体でオーケストレーションを実行する必要があります。

そのニーズに応え、エンドツーエンドのサービス オーケストレーションを実現するためのプラットフォームを提供するのがTail-fベースのCisco® Network Services Orchestratorです。本書では以下について説明します。

- 現代のネットワーク オペレータにおいて、サービス オーケストレータがサービス ライフサイクル全体を自動化するために欠かせない4つの重大な特性
- Cisco Network Services Orchestratorのリファレンス アーキテクチャについて (これらの原則を実現させる方法に関する詳細を含む)
- 現実世界のサービス業務において、ライフサイクル全体のオーケストレーションを示した使用事例

自動化の機会

進化を続けるネットワーク サービス市場では、新しいビジネス機会が見込まれるとともに、大きな課題も待ち構えています。NFV、ソフトウェア定義型ネットワーク (SDN)、そしてクラウドからのセルフサービス受発注システムを組み合わせることで、これまで以上に幅広いサービスをカスタマイズし、より迅速に、より多くの市場に提供するという、非常に大きな事業展開の可能性がオペレータにもたらされます。しかし、こうしたイノベーションは顧客の期待も著しくかき立てるものです。

今日の顧客は、オンラインでさまざまな技術製品を購入することに慣れ親しんでいます。こうした顧客は、製品やサービスを必要とときに自らが思うがままに設定できることを期待し、ネットワーク サービスには何故同じような機能が備わっていないのかと疑問に思います。

高い競争力を持ちたいと願うネットワーク サービス プロバイダーにとって、オンデマンドでカスタマイズした製品を構成し、その場で顧客に提供することが求められる日は、すぐそこまで近づいています。

実際のところ、先頃Heavy Reading社が世界中のオペレータ160人を対象に行った調査では、ネットワーク サービス プロバイダーに求められる最優先事項として、設計の合理化と新サービスの提供を挙げた事業者が群を抜いて多かったことが明らかになっています。

しかし同様の技術動向が特にネットワークの仮想化に関する顧客の期待を駆り立てていることから、サービスの設計やデリバリの複雑性ははるかに増えています。サービス プロバイダーが、顧客が求める迅速性と柔軟性に対応するようになるには、複雑な相互依存サービスとハードコードされたインテグレーションを基本とするモデルから移行する必要があります。それに代わり、使用可能な全てのネットワーク サービスを個々の独立した機能として扱い、要望に応じて無数の方法で組み合わせたり分解できるようなフレームワークが必要になります。つまりサービス プロバイダーには、自動化された、再利用可能で、信頼性の高いエンドツーエンドのサービス オーケストレーションが必要です。

エンドツーエンドのサービス オーケストレーションの全容

このようなオーケストレーションを行うには、サービスのアクティブ化におけるライフサイクル全体の自動化およびネットワークの変更を伴います。これは、1回のトランザクションでの1つのサービス（仮想化された機能だけに限らない）に関係する全エレメントを計画に沿って設定することを意味します。

この自動化された設定は、サービスに関連するすべてのネットワーク デバイスの自動化構成に基づいた、サービス レベルのパラメータから拡張される必要があります。実世界のネットワーク トポロジはハイブリッド型であり、単一のドメインに限定することはできません。また、当面はこの状況が続くと想定されます。オーケストレーションは、NFVオーケストレータに加え、物理および仮想両方のネットワーク構成要素、SDNと従来型のネットワーク管理インターフェイス、通常のOSSシステムを包含する必要があります。最後に、オーケストレーションは、フルフィルメント、アクティベーション テスト、継続的なモニタリングおよびアシュアランスを含め、サービスのアプリケーション ライフサイクル全体に対応する必要があります。

この最後の点は、仮想化が加速する世界では特に重要です。顧客によるオンデマンドでのサービスの発注およびアクティブ化が可能になれば、アシュアランスも並行して考える必要があります。サービスのアシュアランスが別のシステムによって新しい管理モデルに取り込まれマップされている間に、遅滞が生じないようにしなければなりません。アシュアランスは、サービス フルフィルメント プロセスの一環として、包括的なオーケストレーションシステムによって駆動し、また同じサービス モデルを使用している必要があります。

Cisco Network Services Orchestrator、ネットワーク サービスにおけるライフサイクル全体の自動化をサポート

Tail-fベースのCisco Network Services Orchestratorは、エンドツーエンドの自動化を実現することにより、サービスの設計、デリバリー、アシュアランスを高速化します。共通のモデリング言語およびデータストアを使用した同製品は、物理と仮想両方において単一のネットワーク全体のインターフェイスをすべてのネットワーク デバイスやサービスに対して提供します。これにより、オペレータは標準化されたYANGモデリング言語を使用してサービスを定義し、そのサービスをYANGデバイス モデル（YANG以外のデバイスも含む）にマップし、高レベルのインテントからきめの細かいデバイス構成まですべての自動化を実現できます。その結果、オペレータは標準化されたモデルを使用してサービスの作成や変更を、より迅速かつ容易に行えるため、長いカスタム コードの作成やサービスの中断はありません。

またCisco Network Services Orchestratorを使用することにより、物理環境および仮想環境の両方において幅広いマルチベンダーのデバイスを自動化することができ、フルフィルメント、アクティベーション テスト、継続的なサービス レベルのアシュアランスを含め、1回のトランザクションかつ単一のデータ モデルを介してサービス ライフサイクル全体のオーケストレーションを実行できます。また、ソフトウェアと同等の速度でネットワークサービスを継続的に改善し、再パッケージできます。（Network Services Orchestratorの機能および利点に関する概要は、<http://www.cisco.com/web/JP/product/hs/netmgmt/network-services-orchestrator/index.html>をご参照ください。）

Network Services Orchestratorでは、Heavy Reading社がサービス オーケストレーションの成功をもたらす「4つの柱」と呼ぶ手法を取り入れ、上述の機能を提供しています。これらの柱では、ますます複雑化が進み、絶えず変化する今日の仮想化環境下において、サービスのライフサイクル全体を自動化する上で、サービス オーケストレータが提供する必要がある主要特性について説明します。具体的にはデータ モデルとデータ モデルのマッピング、ステート コンバージェンス、複数のドメインにおけるオーケストレーション、ならびにオーケストレートされたアシュアランスがそれに該当します。

1つめの柱：データ モデルとデータ モデルのマッピング

すべてのサービスとデバイスを記述するために共通のデータ モデルとデータ モデルのマッピングを使用する能力は、ネットワーク構成作業自動化の基盤です。Network Services Orchestratorでは、標準化されたYANGモデリング言語を使用して、すべてのサービスを正確かつセマンティック情報をふんだんに活用してモデル化できます。

サービスレベルのインテントとデバイス レベルの構成の両方を共通言語で正確に記述することで、Network Services Orchestratorではデータ モデルをプログラマ的に操作することが可能です。これは、サービス モデルをデバイス モデルに宣言型としてマップでき、手動でコードを作成する必要がないことを意味します。Network Services Orchestratorではサービスの設計、フルフィルメントおよびアシュアランスを高速化し、またそれら全てを自動化します。また、ハードコードされたサービスロジックや増え続けるワークフローに依存する従来の手法と比較すると、圧倒的に柔軟性が高くなっています。

2つめの柱：ステート コンバージェンス

真のエンドツーエンドの自動化を実現するには、オーケストレータはサービスの「インテント」（または求めるサービスや変更のためにブラック ボックス化された入力パラメータ）を受け取れるようにしておき、ネットワーク上で実際の変更に変換する必要があります。この作業を達成するために、現時点ではワークフローの定義に依存しているオペレータが大半で、絶え間なく増え続ける個別のワークフローへの対応に莫大な労力を費やしています。

Network Services Orchestratorでは、サービス ライフサイクル管理に対して、ステート コンバージェンスのコンセプトを用いた異なる方法を採用しています。

同じ共通データ モデルとモデリング言語を使用してサービスとデバイスを記述することから、Network Services Orchestratorではエンドツーエンドでネットワークの作成、修正、削除を完全に自動化します。単一で柔軟性の高いデータモデルを使うことで、設計時のサービス定義から実際のネットワークオペレーションに、プログラマティックにマップできるようになります。また、このデータ モデル マッピングは繰り返し使用可能なプロセスとなります。その仕組みは以下の通りです。

Network Services Orchestratorでは、ワークフロー定義を1つだけ必要とする実行時のステート コンバージェンス アルゴリズムを採用しています。それにより、実行時に利用可能なネットワーク リソースとステータスが検出され、既存のステートと新たなインプットの間における差異を算出し、求めるステートを実現できる最短パスを見つけます。

ノースバウンド システムからの指示によるものか、または管理しているデバイスの変更によるものかにかかわらず、ネットワーク内でなんらかの変更が生じた場合には、同様のプロセスが発生します。Network Services Orchestratorでは、サービスの役割について記述されたモデルを確認した上で、ネットワークを動的に再構成して求められているサービス インテントを実現します。こうした作業は、実行時に数分もしくは数秒で完了します。

あらゆる変更に対応できるように事前に定義しようとすると、ワークフローでは無限に続くこととなりますが、ステート コンバージェンス方式を使用することで、Network Services Orchestratorでは作業が大幅に軽減されます。それと同時に、サービスをアクティブ化した時点および展開後の変更時の信頼性が増し、いずれの場合もサービス提供までの時間が短縮され、費用が軽減されます。

3つめの柱: 複数のドメインにおけるオーケストレーション

大半のオーケストレーション ソリューションは、SDNまたはNFV向けとして専用で設計されたものです。しかし現実世界の多くのオペレータ環境では、仮想化ネットワーク機能 (VNF) は、顧客に対応するサービス全体のほんの一部に過ぎません。真の俊敏性を実現するには、エンドツーエンドでネットワーク サービスを自動化する必要があります。

Network Services Orchestratorは、NFVの管理とオーケストレーション (MANO) スタックで定義されている、NFVサービスオーケストレーションのすべての要件を満たしています。単一のサービス モデルにより、Network Services Orchestratorは起動、設定、継続的なモニタリングおよびVNFのライセンス管理を個別および複雑なサービス チェーンの両方に対して行うことができます。

またオペレータ環境の大部分を構成する、従来のネットワーク管理ドメインと物理的デバイスで設定とモニタリングを自動化することも可能です。オペレータは1回のトランザクションで、エンドツーエンドのサービス全体を自動化できることから、サービス提供までの時間の短縮、運用コストの軽減、ならびにオンデマンドのセルフサービスによる発注を可能にします。

4つめの柱: オーケストレーションされたアシュアランス

サービスのライフサイクル全体を自動化することは、プロビジョニングをオーケストレーションするだけではなく、継続的なサービスのモニタリングとアシュアランスも行う必要があります。Network Services Orchestratorを利用することで、サービスのアクティベーション テスト、継続的なモニタリング、フルフィルメントと同期するサービスレベル契約 (SLA) の保証機能を構成できます。これらも、同じ共通データ モデルが使用されています。

オペレータは、サービス モデルの一環として実行時の主要業績評価指標 (KPI) を定義し、それを適切なロケーションから測定するためにサービス モニタリング デバイスを設定できます。これにより、オーケストレーション システムによるネットワークのステートフル表示をリアルタイムで利用でき、ネットワークで仮想プローブのインスタンス化を自動化できます。つまり、間接的なデバイスレベルの評価基準からではなく、アクティベーション時またはその後継続的に顧客体験を直に測ることができます。また、Network Services Orchestratorのステート コンバージェンス機能を利用して検出された問題に対して自動的に応答し、SLAを履行する上で必要に応じてサービスを再配置することができます。これらはすべて、オペレータの運用オーバーヘッドの削減、エラーや出張サービスの減少、顧客体験の向上に結び付きます。

オンデマンドのサービス ライフサイクルのオーケストレーション向けプラットフォーム

NFVの世界において、オンデマンドでのサービス プロビジョニングの高速化という顧客の要求を満たすには、オーケストレーションシステムで将来的に前述の4つの原則が対処される必要があります。Network Services Orchestratorでは製品ロードマップとしてではなく、オペレータ ネットワークで実証可能な実行コードでこれらを今すぐ提供できます。

以降のセクションでは、Network Services Orchestratorのリファレンス アーキテクチャの概要について説明します。この中では、オーケストレーション実現の4つの柱についても取り上げ、その実行方法を紹介しています。最後に、本書では、Network Services Orchestratorの使用事例として、エンドツーエンドのサービスをプロビジョニングする、オーケストレーションされたアシュアランスを含むNetwork Services Orchestratorアーキテクチャを記載しています。

Network Services Orchestrator アーキテクチャの概要

Network Services Orchestratorは現在、世界各地の主要なオペレータ ネットワークにおいて導入されており、オーケストレーションのフルフィルメント ソリューションとしての地位を確立しています。しかし、オーケストレーションとは、単にサービスをプロビジョニングすることだけではありません。

最新リリースのNetwork Services Orchestrator 4.1を使用すれば、シスコの包括的なリファレンス アーキテクチャを利用でき、エンドツーエンド、マルチドメイン サービスの自動化を実現できます(図1)。この中には、サービスのライフサイクル全体でのオーケストレーションも含まれ、フルフィルメント、アシュアランス、NFVオーケストレーション、MANOフレームワークに沿ったサービスのチェーン化もその対象となります。また、シスコがサードパーティ プロバイダーのものかにかかわらず、物理および仮想両方のネットワーク デバイスを網羅しています。

図1 Cisco Network Services Orchestratorによる高レベル アーキテクチャ

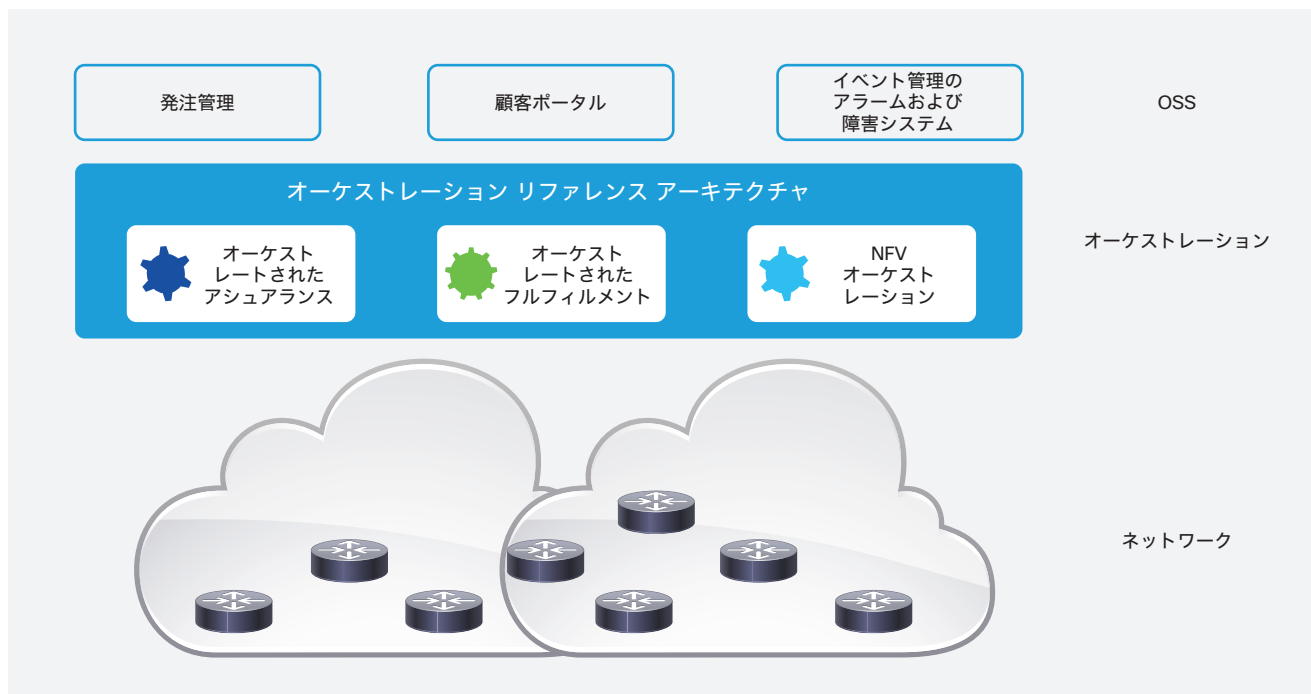
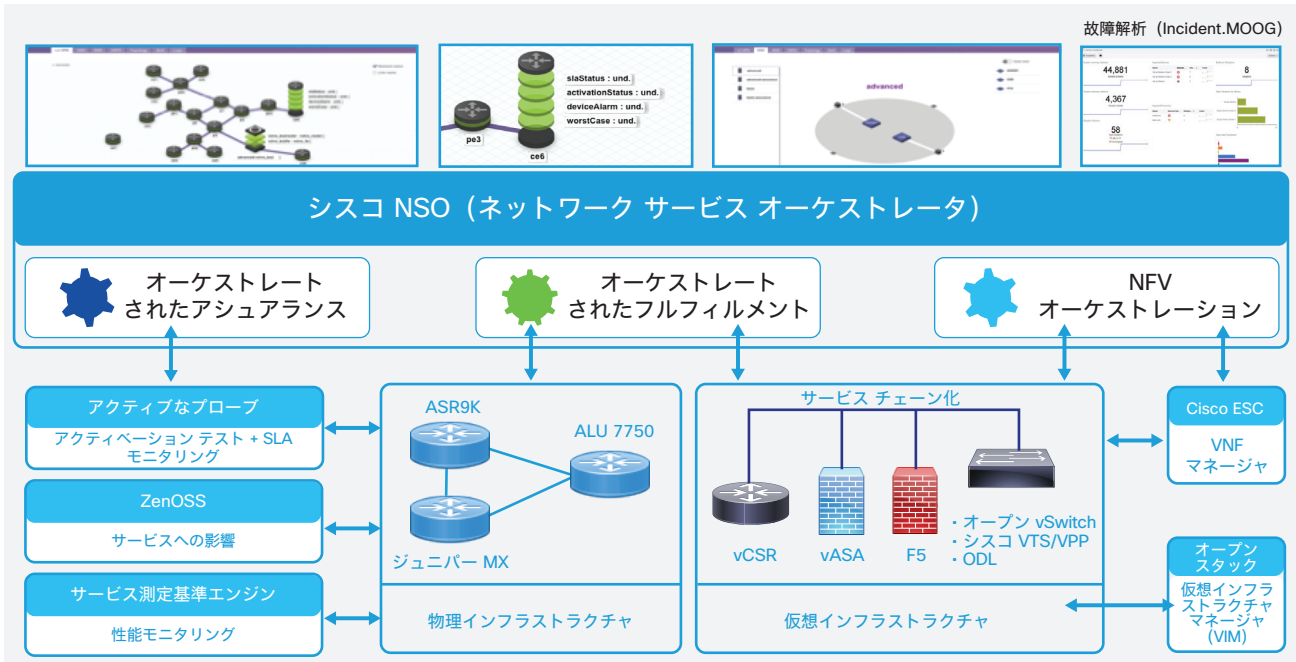


図1のNetwork Services Orchestratorのノースバウンドでは、顧客対応サービスの一部である従来型のOSSコンポーネントを示しています。この中には、発注管理システム、顧客ポータル、ならびにイベント管理のアラームおよび障害システムが含まれます。これらは、技術的にはリファレンス アーキテクチャの一部ではありませんが、Network Services Orchestratorでは、エンドツーエンドのサービス オーケストレーションの一部として統合します。

図2では、実際の運用管理ネットワークにおいて見られるような、アシュアランス、フルフィルメントの詳細とNetwork Services

OrchestratorにおけるNFVオーケストレーション要素を図示しています。ここでは、Network Services Orchestratorがオープンなモジュラとなっている点にご留意ください。このリファレンス アーキテクチャは、VNF管理、オーケストレーション、物理デバイス要素に関するアセットだけでなく、サードパーティ製のオープン デバイス、オープンソース コンポーネントを組み合わせています。これはつまり、オペレータ自身の環境に最適なように要素単位でコンポーネントを追加、変更でき、その後もプラットフォームを継続的に進化させることが可能であることを意味します。

図2 Cisco Network Services Orchestrator リファレンス アーキテクチャ



図のコンポーネントは以下ようになります。

- NFVオーケストレーション:** Network Services OrchestratorのNFVオーケストレーション(NFV-O)コンポーネントは、VNFマネージャ(この場合は、Cisco Elastic Services Controller)を統合してVNFをオンボードします。ここでは、OpenStackの仮想インフラストラクチャマネージャ(VIM)やOpen vSwitchといった幅広く使用されているオープンソースのNFVコンポーネントを利用でき、ここで図示しているように、VNFや物理ネットワーク要素における必要なサービスチェーンを作成します。
- オーケストレートされたフルフィルメント:** 図2の中央では、Network Services Orchestratorにおいて中核となる機能を提供しているサービスオーケストレータを図示しています。これは、ネットワークキングサービスの高レベルのインテントをデバイスの設定へと変換し、こうした設定をサービスで網羅される物理および仮想デバイスに実装します。Network Services Orchestratorはオープンなプラットフォームであることから、さまざまなベンダーから出ているデバイスのオーケストレーションを実行することができます。このイラストにある通り(レイヤ3VPNの一種に対応する使用事例の詳細は後述)、Network Services Orchestratorはシスコ製の仮想ルーティング、スイッチング、ファイアウォールデバイスのほか、サードパーティ製の仮想ファイアウォールも使用してサービスチェーンを作成することができます。

それと同時に、シスコ、ジュニパー、アルカテル・ルーセントのルータなどの物理デバイスの設定も可能です。

- オーケストレートされたアシュアランス:** Network Services Orchestratorそのものは、アシュアランスシステムではありません。ただし、ネットワーク上でのサービスの設定やインスタンス化の一環として、アシュアランスシステムも設定されるため、サービスを認知してアクティブ化されます。ここでも、アシュアランスのオーケストレーションするためにサードパーティ製のシステムをNetwork Services Orchestratorに組み込むことができます。例えば、サービスがアクティブ化されると、Network Services Orchestratorは同時に性能監視機能を有効にして以下を実行できるようにします。
 - 仮想テストプローブをインスタンス化してアクティベーションテストを実行し、サービスに対して定義されているSLAに対するサービスKPIを継続的に測定
 - アシュアランスまたはインベントリシステムのサービストポロジを維持し、これにより、サービスに及ぶ影響と根本原因分析を正確に行えるようにする(例えば、「ZenOSS Dynamic Service View」の使用)
 - 音声や動画セッションに備えて、メトリックエンジンやCisco Unified Communications Session Management Editionを使用するIP SLA監視および管理アプリケーションをインスタンス化

VNFのオンボード化やインスタンス化から、ネットワーク サービスの設定やアクティブ化、SLAの定義化、アシュアランス機能の駆動まで、こうした要素はすべて、Network Services Orchestratorのユーザー インターフェイスを通じて管理することが可能です。また、ツールによるネットワーク サービスディスクリプタの設計や構築も、NFVオーケストレーション機能に含まれます。さらにこのアーキテクチャでは、故障解析のためにincident.MOOG製品を使うなど、通常のOSSコンポーネントであるノースパウンド アシュアランス システムと統合することもできます。

それでは、オーケストレーションを成功させる4つの柱をこのアーキテクチャがどのようにして具現化するのかについて詳しく見ていきましょう。

データ モデルとマッピング

Network Services Orchestratorは、形式言語を使用するという独自の手法でサービスのデータ モデルをすべて定義します。Network Services Orchestratorがオーケストレーションするものは、フルフィルメントまたはアシュアランス、仮想または物理機能であれ、すべてYANGデータ モデルで最初にモデル化されたものです。Network Services Orchestratorは、サービスに関するドメイン固有の情報ではなく、これら抽象化されたデータ モデルに基づいてオーケストレーションを実行します。何故、このことがそれほど重要なのでしょうか？

従来のサービス アクティベーションでは、高レベルでサービスの説明を行う際に使用していた非形式のモデルと、実際のデバイスを設定する際に使用していたきめ細かなコマンドライン インターフェイス (CLI) およびAPIの間に大きな隔りがありました。この隔りが、ネットワーク サービスのライフサイクル自動化における高速化、効率化における最大の障壁となっていました。

高レベルでは、ドメインのエキスパートはVisio図形、Word文書、統一モデリング言語 (UML) の図形またはその他の非形式な方法を利用して、サービスを定義します。これらのモデルは、計画したサービスに関して、人と人で行われる話し合いでは役立ちます。しかし、これはオーケストレーション システムにプログラマ的に理解されることはありません。これはつまり、これらが自動的に実行不可能であることを意味します。

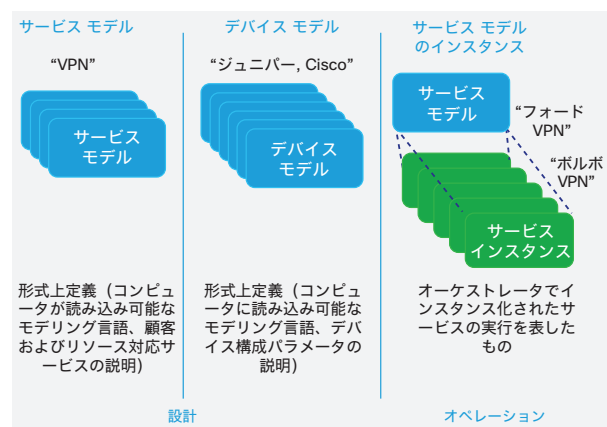
これは、非形式のサービス定義を、サービスをインスタンス化するために特定のデバイス一式のCLIやAPIのワークフローに変換するという、多大な時間とコストを要するプロジェクトに従事する専門サービス チームにとっては吉報と言えます。しかし、新サービスを導入するために、さらに時間とリソースを費やさなければならぬオペレータにとっては嬉しい話ではありません。また、この変換作業は1度だけで済むのではなく、新しいサービス、あるいはデバイスに実装される必要があるオペレーションが登場する度に新たなワークフローが必要になります。

加えて、この手法を採用しているオペレータのドメインエキスパートが、サービス インテントの解釈を、ソフトウェア実行チームに受け渡します。これでは、エキスパートがモデリングしているサービスの最終的なセマンティックスを把握したり、管理することができなくなります。つまり、顧客の要求変更に対する応答は、どうしても遅く、さらに扱いにくく、リスクが高くなる可能性があります。

正確なデータ モデルを利用したオーケストレーション

ここでは、製品チームがサービスに対して具体的で、セマンティック情報をふんだんに活用したデータ モデルを使用すると、どのようなことが実現できるのかについて考えてみます (図3)。ここでは、コンパイラで解析可能な明確に定義された文法を用いたYANGなどの標準化されたモデリング言語を前提としている点にご留意ください。また、オペレータはデバイスの機能を詳細に説明している正確なスキーマが盛り込まれたデバイス (物理と仮想の両方) で、同じ言語の形式データ モデルを使用していることを前提としています。

図3 サービスとデバイスの共通データ モデルを使ったオーケストレーション



オペレータが標準化されたテキストベースのデータ モデルを同一言語のサービスおよびデバイス用に入手した時点で、データ変換プロジェクトを介在させることなく、完全な自動化およびレンダリングが可能になります。プログラマがUnixにおいてコード化していたのと同様の方法で、ここでは自動化されたツールを使用したり、機械的な読み込みでサービス要素を解析したり、次の処理に受け渡すことが可能になります。

これで、オペレータは宣言的データ モデルのマッピングを使用して高レベルのサービス インテントからきめの細かいデバイス構成へと移行できます。これが、サービス提供における高速化、自動化の基盤となります。別の顧客、場所、デバイス(複数のベンダーから、物理および仮想まで)に適用する場合であっても、同じサービスモデルを繰り返し利用することが可能です。

また、オーケストレーションは全面的にデータモデル駆動のため、サービスのセマンティクス全体も、プロフェッショナル チームの代わりにドメインの専門家によって管理されることとなります。大量のデータ変換作業は、もはや不要で、サービスの作成や更新は、これまでよりもずっと簡単になります。また、新しい顧客や機会に対するオペレータの応答は俊敏性を増します。

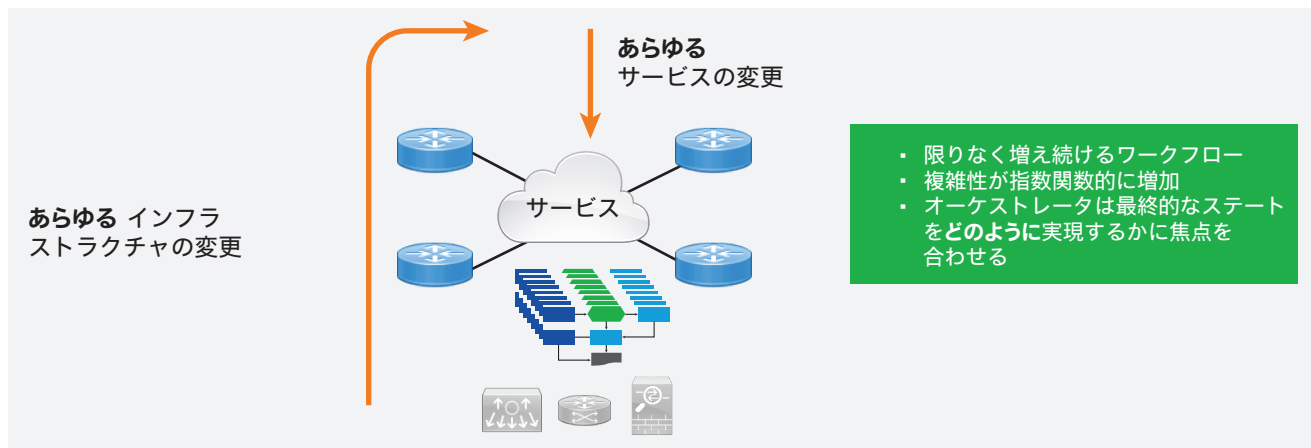
ステート コンバージェンス

近年のオーケストレーション システムではデータ モデルを採用しているものもありますが、設計時のサービス データ モデルを定義すると、実行時にネットワークで実装するときの間には隔たりが生じます。この隔たりを埋めるために、従来はワークフローを

使用してきました。これは、特定のデバイス一式（また可能性がある特定の顧客に）のために、きめ細かな構成詳細を定義する一連のワークフローを通じて、プロビジョニング プロセスのあらゆる段階を表現するためです。

実際にはワークフローベースの手法は、複雑な実世界のオペレータ環境で使用し続けるには正確性に欠け、柔軟性も十分ではありません。オペレータが、必要なワークフローをすべて定義し、再利用可能なワークフローテンプレートを採用したとしても（特定の最終ステートを実現するために各サービス要素の導入、修正、削除方法の詳細を定義）、これは終わりのない作業となってしまいます（図4）。

図4 ステート コンバージェンスを利用しないオーケストレーション



例えば、オペレータが設計時に レイヤ3 VPN用のデータモデルを作成する場合、ネットワークの全ベンダーに対してデバイス データ モデルを組み込みます。一見ただけではワークフローを作成して、モデル化されたサービス インスタンスを正確なデバイス設定に変換するほうが理にかなっていると思われるかもしれませんが、LEAP (Lightweight Extensible Authentication Protocol) のサポートを必要とする顧客はどうしたら良いでしょうか。別のワークフローを作成すれば良いでしょうか。では、VPNに新しい区間を追加する場合は、どうしますか。別のワークフローを用意しますか？では、コストパラメータを変更する場合はどうでしょう。別のワークフローを用意しますか？

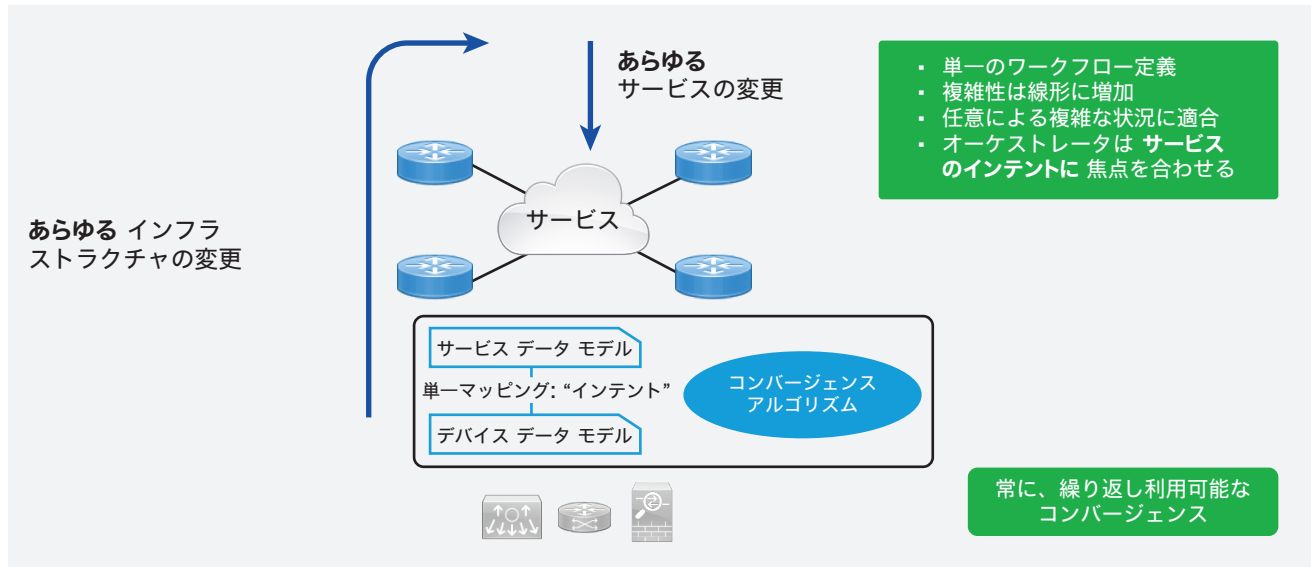
一般的なマルチベンダー オペレータ環境ではワークフローの複雑性が飛躍的に増します。また、それは可能性があるサービスごとに組み合わせて、新しいワークフローを書いているに過ぎません。環境の変化に応じて、不規則に増え続けるこうしたワークフローすべてにおいて一貫性を保つのは、途方もない作業となります。これは言うまでもなく、仮想化に関連する複雑性とはまったく関係がないことで、サウズバンドの仮想デバイスでは自発的にスケールアップやヒーリングする可能性があります。

真の俊敏性を実現するには、オペレータは変更の可能性があるサービスをすべて自動化する必要があります。この中には、ネットワーク デバイスとノースバウンド API (セル サービス ポータルやオーダー マネージャなど) からの指示に対する変更の両方が含まれます。ユーザーや場合によっては顧客も、シンプルな宣言コマンドを利用して、(実行時に継続的にサービスを加速、修正、破棄するために) きめ細かなネットワークの変更を駆動できるべきです。

サービス マッピングを利用したステート コンバージェンス

Network Services Orchestratorでは、真のサービス自動化を実現するために別の方法を採用しています。ステート マシンで最終的に導入されたサービス ステートをモデル化するのではなく、導入をモデル間の継続的なマッピング プロセスとして取り扱います。オペレータや顧客から高レベルのサービス パラメータを効率的に導き出してから、実行時に求める最終ステートを実現するために必要な手順すべてを引き出します。ユーザーは各手順に関する詳細を提出する必要はありません。すると、自動的に求めるステートにネットワークを収束します。（図5）。

図5 ステート コンバージェンスを利用したオーケストレーション



ステート コンバージェンスを、すべての組み合わせにおけるサービス全体のために用意された単一のワークフロー定義とと考えてみてください。ワークフローには段階的に指示を提供するのではなく、実行する上で利用可能な高レベルのサービス インテントとリソースが記述されています。つまり、「私たちの必要とすることはこれ、自由に使用できる物理および仮想デバイスがこれ、今からそれを実現させなさい」といった記述がなされているわけです。

サービス データ モデルの各属性については、Network Services Orchestratorがデバイス データ モデル内の適切な換算値を識別します。例えば、基本的な レイヤ3 VPNでは、Network Services Orchestratorによってサービス データ モデル内で可能性がある3つの異なる属性と、デバイス データ モデル内で可能性があり一致する25の異なる属性を単一の定義内でモデル化できます。

Network Services Orchestrator は続いて、ステート コンバージェンスアルゴリズムを使用して、その高レベルのサービス インテントをネットワーク実行時に遂行します。ノースバウンドシステムから、サービスの作成、削除、修正の指示がNetwork Services Orchestratorにあった場合、アルゴリズムがサービスとネットワークの現在の属性を調査した上で、数秒でデバイス データ モデル用に適切な換算値を導き出します。これで、終わりのない段階的なワークフローの定義は不要となります。その代わりに、オペレータはサービスの単一定義を保持するだけで済みます。また、Network Services Orchestratorは求めるインテントをネットワークに自動的に送ります。

このステート コンバージェンス アルゴリズムは、一度だけ実行されるものではない点にご留意ください。実際には継続的に発生する繰り返しプロセスであり、ネットワークにおけるあらゆる変更(ノースバウンドまたはサウスバウンド)に回答する形で自動的に行われます。

ステート コンバージェンスの動作

このステート コンバージェンス方式は、完全なライフサイクルサービス オーケストレーションにとって必要不可欠なものです。なぜなら、複雑な実世界のオペレータ環境を自動化する上で、これが最速かつ最も効率的な手法であるからです。例えば、仮想ファイアウォールを作成するために、レイヤ3 VPNサービスを1つの要素として、オペレータが外部システム (OpenStack など) をトリガーする必要があると想像してみてください。これには数分とかかりません。

Network Services Orchestratorはコンバージェンス アルゴリズムを実行し、求める最終ステートを得ようと試みます。ここで、仮想ファイアウォールがこの時点ではまだ稼働していないことが認識されます。この時点で、プロビジョニングは停止しそのイベントが発生するまで待機します。導入の一環作業で、既に変更されたネットワークは、引き続き有効な状態に保たれます。しかし、ファイアウォールVMが起動した時点で、このイベントはすぐにステート コンバージェンスをトリガーして最初から再実行され、これまでのネットワーク変更がすべて引き続き有効であるかどうかを検証します。

すべてが有効でない場合 (例えば、以前の段階で作成された仮想ルータが故障しているなど)、Network Services Orchestratorはいったん引き返し、問題を修復してから、最終的に希望する状態になるまでコンバージェンス アルゴリズムを繰り返します。または、エラーが解消されない場合、すべての構成をトランザクションの最初にロール バックし、サービスの各手順を「未デプロイ」の状態にして、稼働中のサービスに影響が及ぶことのないように正しい順番にします。

こうしたコンバージェンス機能の中核を成す、Network Services Orchestratorでは「最小差分」のエンジンを利用して、現在のネットワークのステートから求める最終ステートへの変更する際に、必要とされる変更の最小変更数を算出します。これは、すべてのサービス ライフサイクル運用（作成、修正、削除）に適用されます。Network Services Orchestratorは、最終的に希望する状態に収束する上で必要に応じて何度でもデプロイを戻すことができます。先に正常に完了したネットワーク変更手順に影響が及ぶことはなく、システム オーバーヘッドが大幅に軽減されます。

Network Services Orchestratorは、OpenStack VIMやIPアドレス指定システムなど、他のシステムに対するステートの変更もトリガーできます。これは、「パブリッシュ / サブスクライブ」(pub/sub) 型のメッセージング モデルを使用して、アーキテクチャの別のすべての要素を把握し、オーケストレーション リクエストに対する応答を検出し、仮想インフラストラクチャや外部システムに対する導入指示を再度実行します。

ここでも、こうした動きはすべて、ネットワークのノースバウンドやサウスバウンド イベント単位で継続的に発生します。サービスおよびデバイス、デバイス モデル マッピング、繰り返し実行されるステート コンバージェンスに対して、単一データ モデルを使用する、このプロセスにより、ワークフローなしで全自動化することが可能になります。

複数のドメインにおけるオーケストレーション

「エンドツーエンド」という専門用語はテクノロジー ベンダーの間では、非常に大きな意味を持っていますが、実際にはどういうことを指すのでしょうか。ネットワーク サービス プロバイダーの観点から見ると、この用語はサービスのあらゆる側面を実現してくれる単一のオーケストレーション システムを有することを意味し、そこではネットワーク要素が物理または仮想で、SDNコントローラ、NFVオーケストレータ (NFV-O)、または従来のCLIを介してインターフェイスでつながっています。

顧客に対応する完全なサービスでは、SDNやある程度の仮想機能、および従来のネットワーク構成が大きく関連するケースが大半で、最終的にはアプリケーションとアシュアランスも関係してきます (図6)。このため、莫大な数のもののなかから一部だけを管理するオーケストレーション システムでは、真の意味においてサービスをエンドツーエンドで自動化することはできません。

これは、NFVオーケストレーションにおける現在の業界マーケティングでは、心に留めておくべき特に重要なポイントです。

VNFが、アクティベーション化やサービスのチェーン化においてサービスやNFV-Oの重要な要素を提供し、サービス オーケストレーションで重要な役割を果たす場合もあります。しかし、収益は仮想マシンの作成によってもたらされるのではなく、エンドツーエンドのサービスによってもたらされます。これは、オペレータのOSS/BSSおよび要素管理システム (EMS) と密接に結びついていきます。VNFがある程度または大半のサービスのインフラストラクチャを構成していたとしても、オペレータの事業の核となる顧客やサービス固有の設定には、NFVドメインを超えて広がるオーケストレーションが必要になります。

図6 複数のドメインをつなぐオーケストレーション



Network Services Orchestratorは、NFV中心の要素の上にあるオーケストレーション システムを包括的に提供します。そこから、サービスのエンドツーエンド アプリケーションのライフサイクル全体を自動化でき、NFV MANOシステム、SDNコントローラ、従来のOSS、BSS、ならびにEMS要素を統括します。それでは、その仕組みを詳しく見ていきましょう。

マルチドメインのオーケストレーションの動作：NFV MANO

先ほども述べましたが、NFV MANOスタックは、VNFのアクティベーションとアシュアランスだけを個別またはネットワークアプリケーションのサービス チェーン単位で管理します (図7右参照)。これでは、オペレータ環境の大部分を構成している物理デバイスに依存する、顧客対応サービスのこうした側面に取り組むことはできません。

またVNFがアクティブ化されると、ハイブリッドのオペレータ環境の大半がこれを物理ネットワーク デバイスと同じ手法で取り扱うという、さらにやっかいな問題が生じます。つまりこれは、仮想デバイスはマルチテナントのサービスに対応していなければならないことを意味しています。(例えば、単一の仮想アグリゲーション ルータでは、物理アグリゲーション ルータが行うのと同様に、複数の顧客に対応できる機能が求められます)。

顧客対応サービスのエンドツーエンドの自動化には、サービスがVNFで構成されていても、NFV MANOの範囲を超えて追加でOSS構成が必要になることは明白です (図7)。

ネットワーク サービスと仮想ネットワーク アプリケーションの問題解決

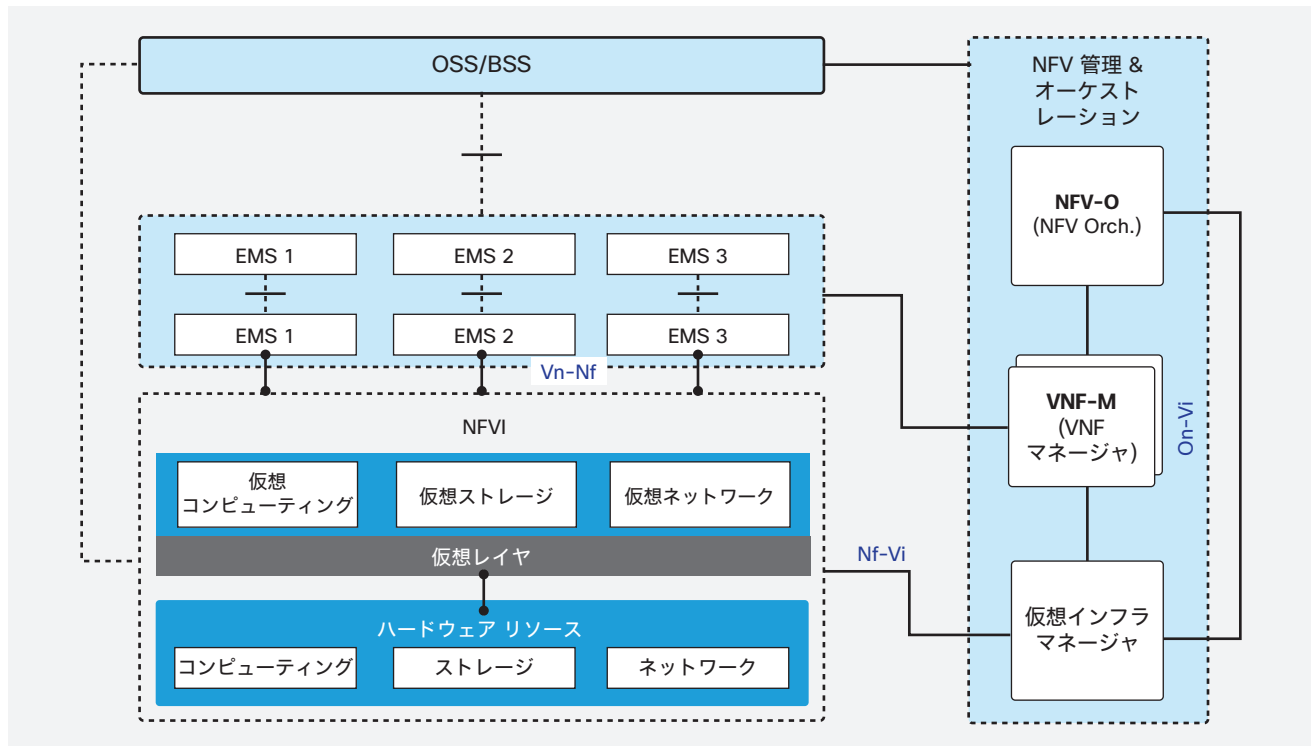
「ネットワーク サービス」という言葉の意味を巡り、NFV MANOに対するものとネットワーク サービス プロバイダーに対するものとの間で、業界では一部混乱が生じています。

欧州電気通信標準化機構 (ETSI) では、NFV MANO スタックにおけるネットワーク サービスとしています。これは、VNF と VNF サービス チェーンにおけるゼロ コンフィギュレーションを指しています。(ネットワーク サービスが起動中の一連のアプリケーションを意味する、クラウド アプリケーションのトポロジとオーケストレーション仕様 [TOSCA] についてもこれが当てはまります)。これは、仮想 IP マルチメディア サブシステム (vIMS) や仮想化モバイル コア (vEPC) など、所定のアプリケーションを提供するために個別 VNF または一緒にチェーン化された VNF の 1 セットで構成可能です。

一方、オペレータはネットワーク サービスをまったく別物として定義します。サービス プロバイダーにとっては、ネットワーク サービスは、複数のネットワーク要素とアプリケーションをつなぐ、高レベルのサービス提供を意味します (E-Line、VPN、4G モバイル データなど)。また、単一のネットワーク要素が複数のサービス インスタンスをサポートする場合があります。

オペレータの観点から見ると、オプションで用意されている vIMS や vEPC インスタンスも、実際のサービスの一部に過ぎません。明確にするために、本書ではネットワーク サービスという用語は、オペレータが理解できる標準的な意味で、ネットワークのアプリケーションとして VNF ならびに NFV MANO スタックのサービス チェーンを指しています。

図7 NFVオーケストレーション



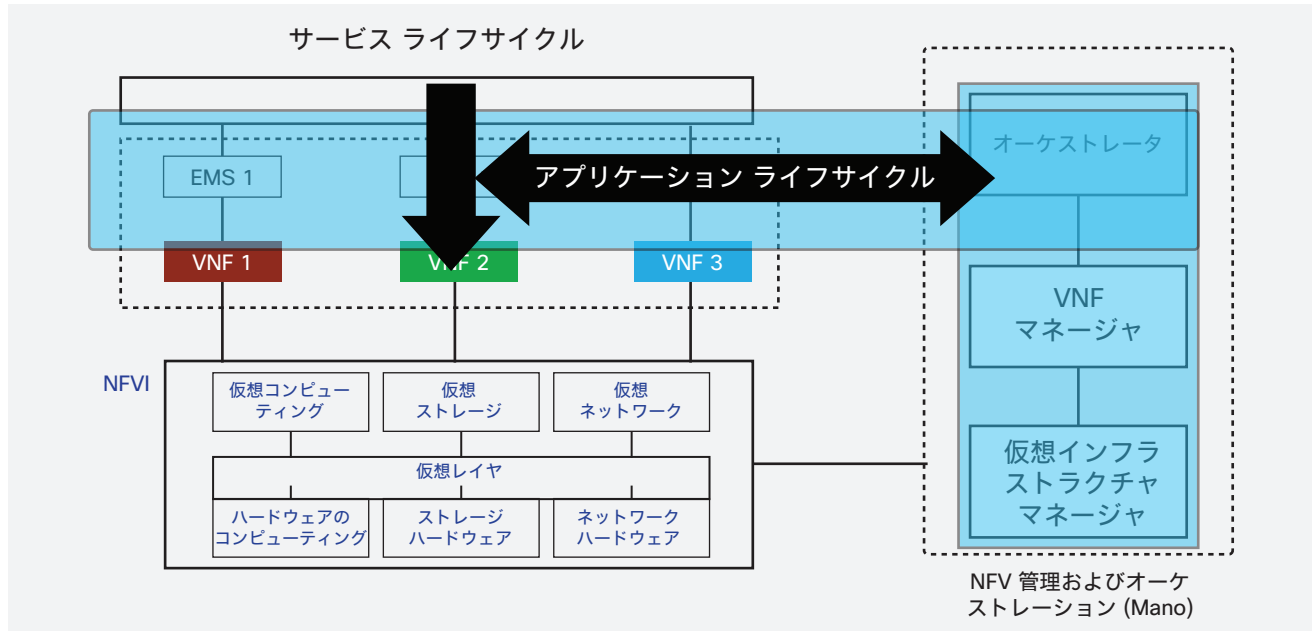
Cisco Network Services Orchestrator は、ライフサイクル自動化フレームワークを包括的に提供します。Network Services Orchestrator は、NFV MANO フレームワーク内における NFV-O の役割を全面的に引き継ぎつつ、OSS 機能も組み込み、最終的なサービス構成に対応できます。サービスの VNF と直接やり取りして、実行時のコンフィギュレーションを設定し、VNF のインスタンス化をトリガーしてサービス チェーンを設定します (図 8)。

NFV MANO の仕様で定義されているように、Network Services Orchestrator は VNF のクラウド面やネットワーク アプリケーション ライフサイクルを管理するために、MANO デ

スクリプタや TOSCA テンプレートなど、事前に定義されたテンプレートもサポートしています。この中には、アプリケーション インスタンスのゼロ設定、ネットワーク アプリケーション インスタンスのインストール、アンインストールや再インストール、ならびにアプリケーション インスタンスの起動または再起動が含まれます (例えば、VNF 構成変更の実行など)。

このように、Network Services Orchestrator は、仮想化されたアプリケーションに対して NFV-O 機能を全面的に提供します。また、顧客対応サービスのアプリケーション ライフサイクルを完璧に自動化するために、NFV MANO の枠を超えて包括的なオーケストレーション フレームワークを実行します。

図8 アプリケーション ライフサイクルにおけるNFVのオーケストレーション



マルチドメインのオーケストレーション動作:SDN

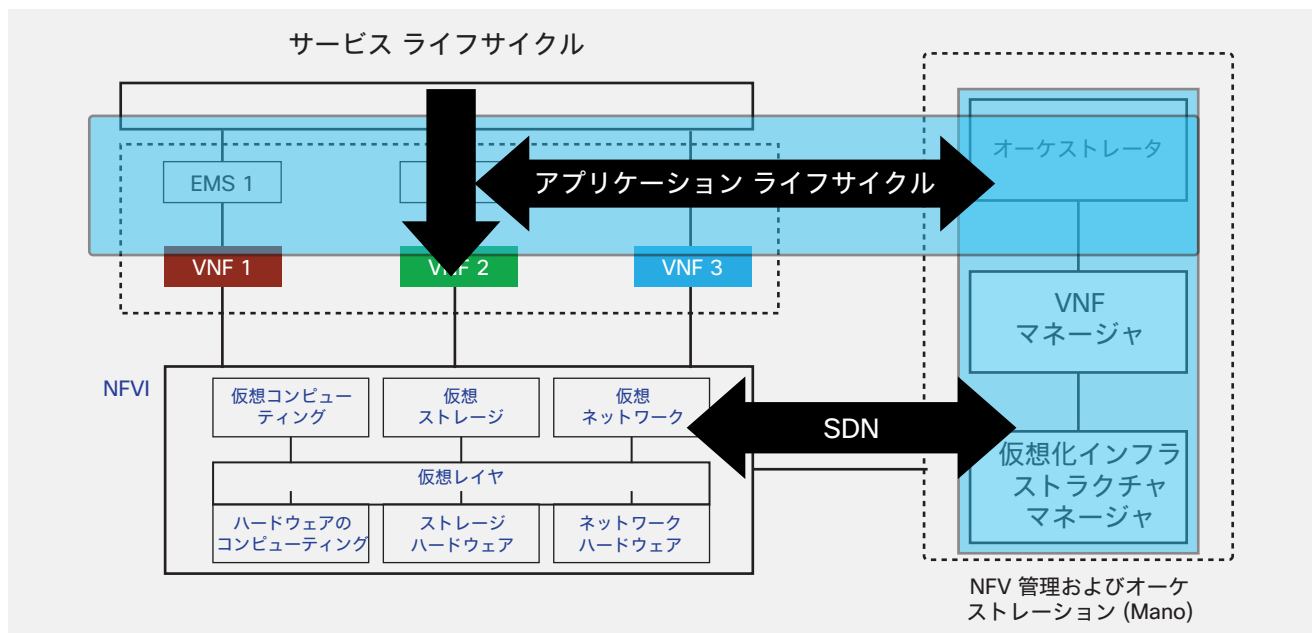
Network Services Orchestratorは、SDNコントローラによって駆動される顧客対応サービスのネットワーク要素を管理することもできます。この代表的な使用事例としては、データセンターでVNFをアクティブ化する際、オペレータがSDNを使用する場合があります。SDNコントローラは、VNFをステッチングして求められるネットワークアプリケーションにまとめ、データセンターをネットワークに接続させる役目を負います。

このような場合、Network Services Orchestratorは引き続き、エンドツーエンド サービス モデルのソースとなります。

しかし、NFV-Oをトリガーしたり、ネットワーク内の物理デバイスと直接インターフェイスを取る代わりに、SDNコントローラをトリガーして必要とされる物理および仮想ネットワーク機能をアクティブ化します(図9)。

ここで重要なポイントは、SDN機能が極めて重要ではありますが、顧客に対応する完全なサービスの一部であるということで、Network Services Orchestratorはサービス ライフサイクル全体を網羅する、エンドツーエンドのオーケストレーションフレームワークの一部としてその機能を自動化します。

図9 SDNを利用したオーケストレーション



オーケストレーションされたアシュアランス

「アシュアランス」という言葉を聞くと頭に思い浮かぶのは、複雑な各種システムや計器のテストです。しかし突き詰めれば、これはすべて「お客様は、支払いに見合う体感品質 (QoE) を得られているのか」という1つの質問に集約されると言えます。残念ながら、大半のオペレータ環境において、この質問に答えることはたやすい作業ではありません。

その原因の一部として、サービス フルフィルメントとサービス アシュアランスが、従来、2つの独立したプロセスとして処理され、別々のシステムを通じて、異なる時期に、しばしばまったく異なる運用チームによって管理されてきたことが挙げられます。大半のオペレータ環境では、回路のアクティベーションが完了し、稼働していることを確認できた時点で、フルフィルメント チームの役割は終わります。そこからネットワーク データを集め、サービスが約束どおりに実行されていることを確認するという、容易ではなく高いコストがかかりがちなプロセス(特に新しいサービスの場合)を完了させるのは、アシュアランス チームの仕事になります。

オンデマンドのセルフサービス式システムのネットワーク サービスに移行するにつれて、オペレータには、サービスが約束どおりに実行されていることを保証し、サービス アシュアランス プロセスを加速化させることという両方に長けることが求められるようになります。しかし、そこには主な障壁が3つ立ちまわっています。

- 一貫したアクティベーション テストの欠如: オペレータは、あらゆるサービス導入が、少なくともアクティベーションの時点で想定どおりに動作していることを検証できるのが理想的です。しかし残念ながら、これが実行できる状況はめったにありません。サービスレベルのテストと測定は従来、高額な機器と相当の運用リソースを必要とするため、大規模で最も重要な顧客のためのみに限られています。残りの顧客では、大半の導入においてテストは一切行われず、基本的なpingテストに頼っています。このようなテストでは、接続性は検証できますが、顧客の実体験については何も明らかになりません。サービス到達可能性、遅延、適切なトラフィック プライオリティの設定、およびその他の主要パラメータについては何もわからないままです。このため、相当数の初回導入が正しく行われずという結果が生じます。これは、現場の作業コストが上がり、収益を生み出すまでの時間に遅延が生じ、顧客の不満につながります。
- フルフィルメントとアシュアランスのサイロ化: これまでフルフィルメント業務から切り離されてきたアシュアランス システムは、プロビジョニング ツールとはまったく異なるものへと進化してきました。フルフィルメント システムが通常、ネットワークやサービスに関する詳しいリアルタイムの情報を持ち合わせているのに対し、アシュアランス システムは一般に、直近の変更や更新が反映されていないかもしれない限定的な情報を頼りにしています。

情報が比較的新しくても、従来型のアシュアランス ツールは、比較的静的なネットワークを前提としています。これはSDNやNFVを利用してオペレータが移行を進めている動的環境とは正反対の環境です。従来型のアシュアランス システムは、プロビジョニング プロセスとの同期も取れていません。使用するサービス モデルも異なり、フルフィルメントから受け渡された後で行われるため、どうしても遅延がもたせられてしまう本質を備えています。このため、NFV環境のように、オペレータがサービスから短時間でプロビジョニング、アシュアランス、収益化の開始を目指すオンデマンドのサービス提供には不向きということになってしまいます。

- サービスのKPIが直接測定されず、推測されている: 正確なサービスレベルのKPIを把握することは、何十年の間、オペレータにとって難題であり続け、この結果、大多数のオペレータが間接的なボトムアップ型アプローチに依存する状況が起きています。オペレータは膨大な量の情報(デバイス中心のカウンタ、リアルタイムのネットワーク トラップとアラーム、インベントリシステムのオフライン データ)を収集して相関性を見出し、間接的に顧客体験を推測しようとします。そこで大量のデータを捕捉しますが、その大半はノイズに過ぎません。オペレータは動的にスケールアップ/スケールダウン可能な仮想機能の対処に苦慮しており、サービス パフォーマンスの寄せ集めとなった全体像は、実世界の顧客体験との相関性が低いものです。

質の高いサービスを、高速かつオンデマンドで実現するには、サービス フルフィルメント業務とアシュアランス業務を1つの同期トランザクションの一環として密接に結合させる必要があります。新しいサービスを導入するたびに、自動的に一貫したアクティベーション テストを行う必要があります。そうすることにより、オペレータは、随時継続的にサービスに関する実世界の顧客体験を確認して修復することができるようになります。

Network Services Orchestratorでオーケストレーションされたアシュアランス

Cisco Network Services Orchestratorは、自動化されたアクティベーション テストやサービスレベルKPIの継続的モニタリングの両方を網羅する完全サービス アシュアランスと、サービスフルフィルメント トランザクションを統合します。同じサービス データ モデルとYANGモデリング言語およびツールを使用するため、オペレータは設計時に各サービスに合わせて適切なKPIを定義することができます。さらに、プロビジョニングと並行してサービスのテストを行い、継続的に顧客、およびサービス固有のSLAと照らし合わせてサービスのモニタリングと修復を図ることができます。

Network Services Orchestratorを使用することで、オペレータはアシュアランス機能をオーケストレーションし、以下を行うことができます。

- **回線上で直接KPIを測定**: Network Services Orchestratorを使用することで、オペレータはシスコとサードパーティ製の各種ツールを活用して、オーケストレーションが実行されたフルフィルメント トランザクションの一環として、サービスエンドポイントで顧客の視点からサービス体験を測定することができます。この中には、顧客およびサービスごとにカスタマイズされた仮想プローブを作成し、ネットワーク内の最適な場所に自動配置することが含まれます。
- **詳細アクティベーション テストの自動化**: プロビジョニング プロセスの最終段階として、Network Services Orchestratorは導入サービスに対し（基本のpingテストだけでなく）サービスレベルのストレステストを実施して、顧客の期待どおりにサービスが実行されていることを確認します。それによりメトロ イーサネット フォーラム (MEF) が推奨しているベスト プラクティスである「出生証明書」と言えるものが顧客対応サービスに対して作成されます。
- **インスタンス化された瞬間からサービスのアシュアランスが可能に**: Network Services Orchestratorを使用すると、従来のフルフィルメント プロセスとアシュアランス プロセス間の受け渡しに伴う本質的な遅延が排除されます。サービスが導入されると、根本原因分析と改善を処理する外部のアシュアランス システムに瞬時に通知が届き、定義されたSLAに対する継続的なサービス モニタリングがトリガーされます。
- **SLA違反を自動修復**: サービスが想定どおりの機能を果たしていないとき、大半の場合、サービスのある時点で構成が

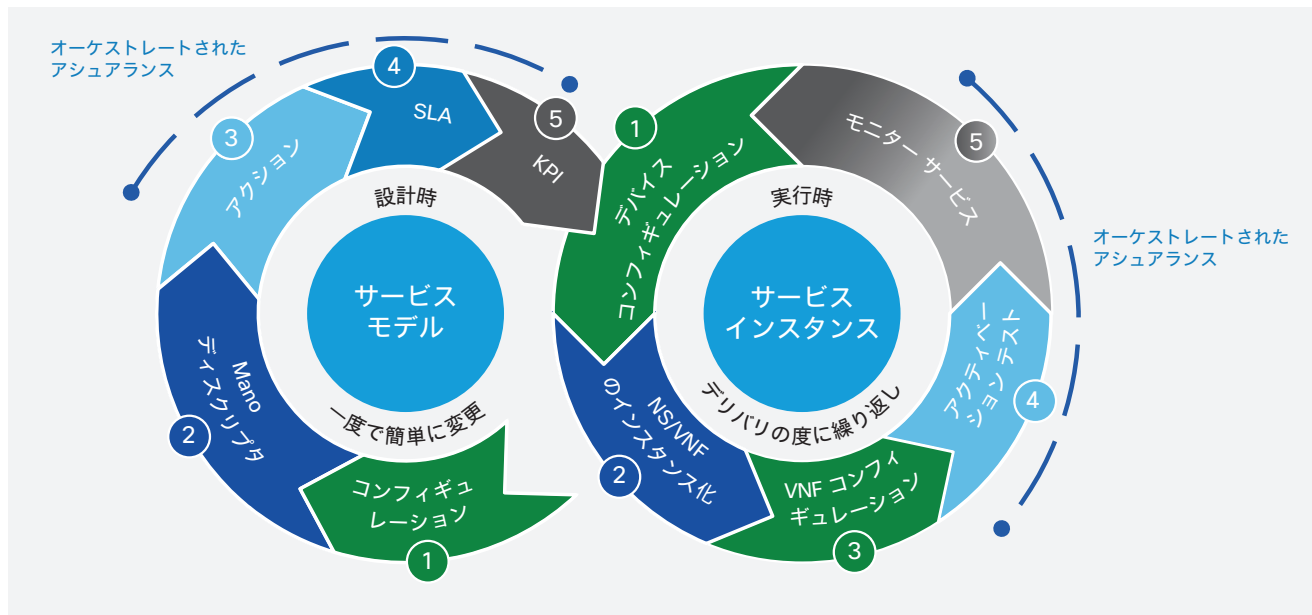
正しく行われなかったことが原因になっています。Network Services Orchestratorでは、デバイス構成に関する包括的なリアルタイムの情報がサービス モデルに備わっているフルフィルメントをアシュアランス プロセスにリンクさせることによって、この問題を直ちに認識し、修復することができます。

アシュアランス システムによって、仮想デバイスまたはNFV インフラストラクチャにサービスの低下が検出されると、Network Services Orchestratorは自動的にステート コンバージェンス アルゴリズムを起動させてVNFを再展開し、問題に対処します。その後、データ モデルで定義されたサービスの完全なライフサイクル（プロビジョニング、アクティベーション テスト、継続的モニタリングの構成）は、閉ループ フィードバック（図10）の一環として、新規サービスの場合と同様に継続されます。Network Services Orchestratorはまた、物理ネットワークに検出された不具合に対して、外部のアシュアランス システムに警告することもできます。これによりオペレータは、デバイスレベルのエラーがサービスに及ぼす影響を従来型のアシュアランス システムが認識するまで待つことなく、顧客およびサービス固有のパフォーマンスに関する問題を直ちに確認し、これに対処することができますようになります。

オーケストレーションされたアシュアランスの実装

図10では、Network Services Orchestratorが、サービスの完全なライフサイクル（プロビジョニング、アクティベーションのモニタリング、継続的アシュアランス）のオーケストレーションをどのように実行しているかを示しています。この図からわかるとおり、オペレータは単一のサービス モデルを使用して、連続した閉ループ プロセスで設計時と実行時の自動化を図ることができます。

図10 サービス ライフサイクル全体を通してオーケストレーションされたアシュアランス



設計時のアシュアランス

左側の円は、設計時にサービス モデルが遭遇するすべての要素を表しています。最初の2つの要素（仮想機能のコンフィギュレーションとサービスディスクリプタ）は、どんなオーケストレーション システムにも備わっています。前述したとおり、Network Services Orchestratorでは、サービスとネットワーク デバイス ディスクリプタの両方が、同じYANGモデリング言語を用いて、同一のデータ モデル内で定義されます。ただしNetwork Services Orchestratorサービス モデルは、(3) ~ (5)の要素（テスト アクションの構成、およびサービスのSLAとKPIの定義）を通じて、サービス ライフサイクルのアシュアランス段階まで拡張されています。

この概念は、サービス オーケストレーション業界では新しいものかもしれませんが、従来型のソフトウェア開発業界では標準の手順です。開発者がコードを記述するたびに、プロセスの一環としてそのコードのテスト ケースも開発することが、ベストプラクティスとされています。Cisco Network Services Orchestratorでサービスとデバイスのモデリングに使用されている、セマンティック情報をふんだんに活用するYANGモデリング言語は、幸いなことにアシュアランスのモデリングにも適しています。

サービス オーケストレーションのコンテキストでは、オペレータはハイレベルな視点から開始することができます。提供されるサービスの種類は？レイヤ3 VPNか？ビデオ会議か？VoIPか？そこからサービスに適したテスト アクション（アクティベーションテスト、ユーザー データグラム プロトコル[UDP]モニタリング、ビデオ モニタリング）のモデルを作成し、これらが実行されるエンドポイントを定義することができます。

この設計時プロセスの一環として、オペレータはこのサービスの名前付きSLA一覧（ゴールド、シルバー、ブロンズなど）のモデルを作成し、そのエンドポイントのSLAしきい値を設定することもできます。最後にオペレータは、顧客のエンドポイントで直接測定される、最も関連性の高いKPIを定義して、顧客が受けたサービスがSLAを満たしていることを確認します。

Network Services Orchestratorは、NFV MANOからエンドポイントまでのテスト エージェントのインスタンス化を駆動します。このような仮想プロブは、アシュアランス システムによるテストおよびモニタリング アクションを引き起こします。さらにデータ モデルで、これらのシステムの実出力（アクティベーション テストの結果、現在のKPI値、SLAのステータス）を読み取り専用値としてキャプチャし、これをNetwork Services Orchestratorで使用して、実行時のアシュアランスのオーケストレーションに使用することができます。

実行時のアシュアランス

実行時（図10の右側の円）に進んだNetwork Services Orchestratorは、データ モデルに従ってネットワークを設定します。最初に、従来のオーケストレーション システムと同様に、そのサービスの物理デバイスと仮想機能が設定されます。同じ全体的な流れの一環として、アクティベーション テストも開始されます。また、SLAに対するサービスKPIの継続的モニタリングも開始されますが、これもすべてこのサービスに対する同一のYANGデータ モデルに基づいています。

先ほども述べましたが、アシュアランスにおいて中心となる質問（お客様の要件を満たしているか？）に答える一番の方法は、エンドポイントで直接KPIを測定することです。従来は、この測定にはオンサイトの機器が必要になり、高額な出張サービスを覚悟しなければなりません。しかし、Network Services Orchestratorでは、ネットワーク トポロジ全体とエンドポイントの情報を使ってオーケストレーションが可能のため、仮想テストプロブのインスタンス化や、Cisco IP SLAなど使用可能な測定機能の構成を行い、すべて適切な場所でこのようなKPIを測定することができます。

最後に、図10の右側の輪の(5)で、Network Services Orchestratorは継続的モニタリングおよびアシュアランス機能のオーケストレーションを行います。この中には、ネットワークやサービスに加えられた変更に応じて、アシュアランス システムに自動変更を促す処理も含まれます。オーケストレータが新しい仮想コンポーネントの追加、既存のVPNへの区分の追加と削除、サービス品質(QoS)構成の変更、あるいはその他の変更をサービスに加えた場合、Network Services Orchestratorは改めてステート コンバージェンスのアルゴリズムのアクティベーションを行います。これにより、こうした変更がアクティベーション時に完全にテストされるよう徹底できると同時に、継続的なアシュアランス システムが行われます。

また、このプロセスの一環として密接に結合しているNetwork Services Orchestratorは、サービスに影響を及ぼす従来のシステムも更新します。これで、旧来のアシュアランス ツール（アラーム、インベントリ、デバイス モニタ）に反映されていた、サービスまたはトポロジに対する動的な変更もトリガーとなってNetwork Services Orchestratorがステート コンバージェンスのアルゴリズムを実行し直すため、閉ループのフィードバックができあがります。

最終結果として、オペレータはNetwork Services Orchestratorを使うことで、フルフィルメントとアシュアランス間のサイロを打破でき、これで配信までの時間が短縮され、オンデマンド サービスに対応します。さらに、顧客体験の正確な全体像が、ネットワークのデバイスレベルのモニタリング データの中からぼんやりと浮かび上がることを願うのではなく、最も関連性の高いKPIを直接測定することで、SLAに対するサービスのアシュアランスを自動的に保証することができます。

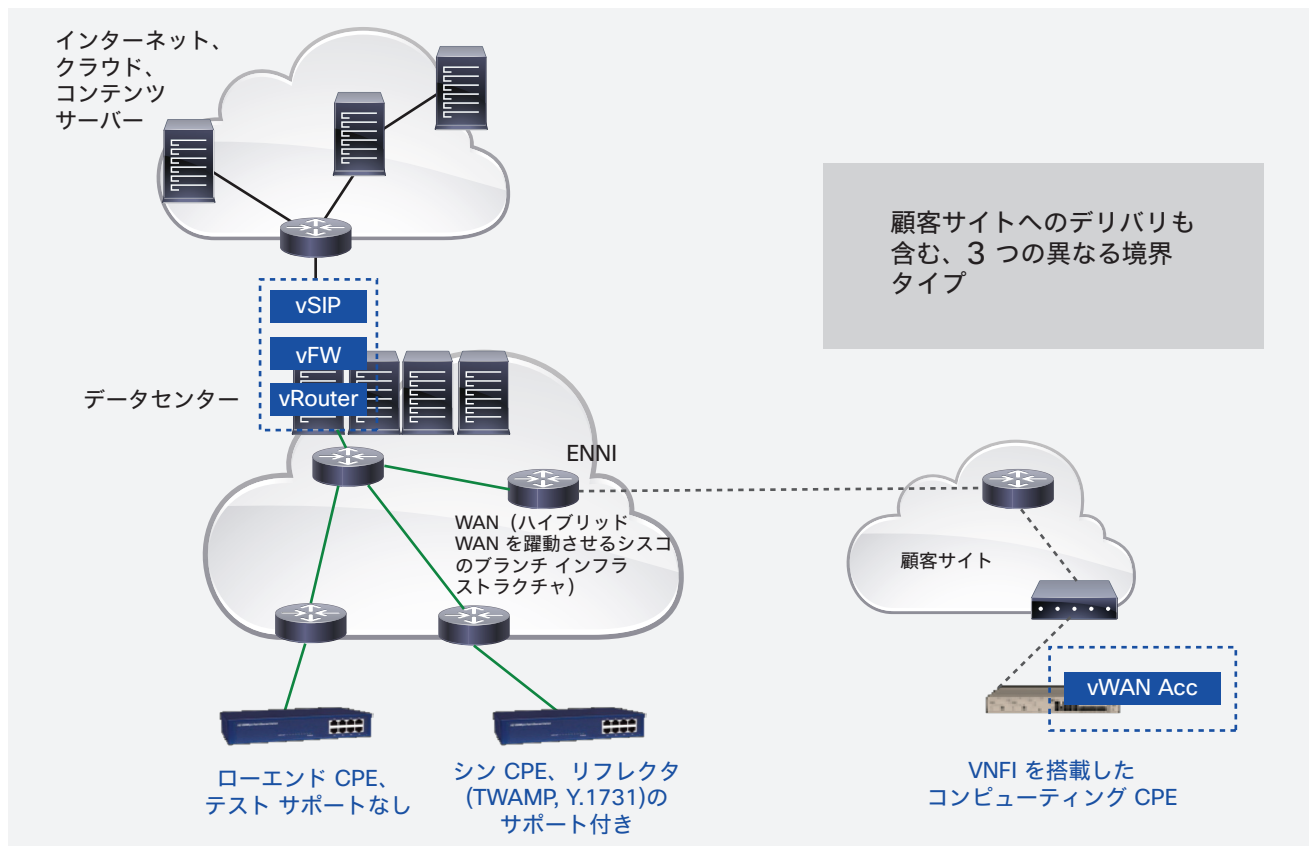
Cisco Network Services Orchestratorの動作 : サービス フルフィルメントとアシュアランスのオーケストレーション

次の使用事例は、Network Services Orchestratorがネットワーク サービスのライフサイクル全体をどのようにして自動化できるかを示すものです。ここでは、レイヤ3 VPNの設計とインスタンス化を詳しく取り上げ、オペレータのネットワークでオーケストレートできる、設計時と実行時のアシュアランスに注目します。ここで紹介するプロセスには比較的ハイレベルな視点が用いられていますが、実世界の概念実証 (PoC) のデモンストレーションを基盤とし、実行コードと実際のNetwork Services Orchestratorユーザー インターフェイスが使用されています。

図 11 は、次の3つの異なる境界デバイスを網羅する、レイヤ3 VPNのシナリオの概要を示しています。

- テスト、運用、管理、保守 (OAM) サポートのない、ローエンドの加入者宅内機器 (CPE)
- リフレクタ (双方向アクティブ測定プロトコル[TWAMP]やY.1731など) のハードウェア サポートが付いた、シンCPE
- VNFインフラストラクチャと共に導入されたハイパワーコンピューティングCPE

図 11 レイヤ3 VPNのシナリオの概要



この図では、仮想化されたWANアクセラレータが顧客サイトのCPE上に存在し、複数の仮想化機能がデータセンター (仮想ファイアウォール、ルータ、Session Initiation Protocol [SIP]ゲートウェイ) 内に存在することを前提としています。この図には、顧客サイトへのデリバリー (特にQoSがこのパス経路で提供されている場合にエラーを起こしやすい) シナリオが含まれていることにご留意ください。

これら3つのすべてのエンドポイントの場合、サービスのアシュアランスのためにオペレータは直接測定方法を使用する点にもご留意ください。これにより、以下の3つのことを実現できます。

1. 最初に展開される時点で、自動化されたアクティベーション テストを有効にします。この中には、メトロ イーサネット フォーラム (MEF) で定義されているキャリア グレード、ネットワークレベルの測定も含まれます。さらに、データ センター内のCPEから VNF までのサービスチェーンにおけるサービス検証も網羅しています。これにより、エンドポイントにおけるパフォーマンスの実際の測定に基づいて、サービス ライフサイクルの継続的なアシュアランス段階に向かうか、エラーになったものを再構成または再配置する必要があるかを、Network Services Orchestrator で認識できるようになります。
2. SLA モニタリングも有効にします。サービス データ モデルで定義されているように、Network Services Orchestrator は顧客のサービス体験に対するリアルタイムのインサイトを得るために、関連するサービスおよび顧客に固有のKPIのモニタリングのオーケストレーションをエンドポイントで実行します。Network Services Orchestrator は、顧客対応ポータルのほか、社内アラームおよびレポート作成システムにも使用できます。最も重要なことは、実行時に認識が可能であり、SLA 違反があった場合は、対処できるということです。

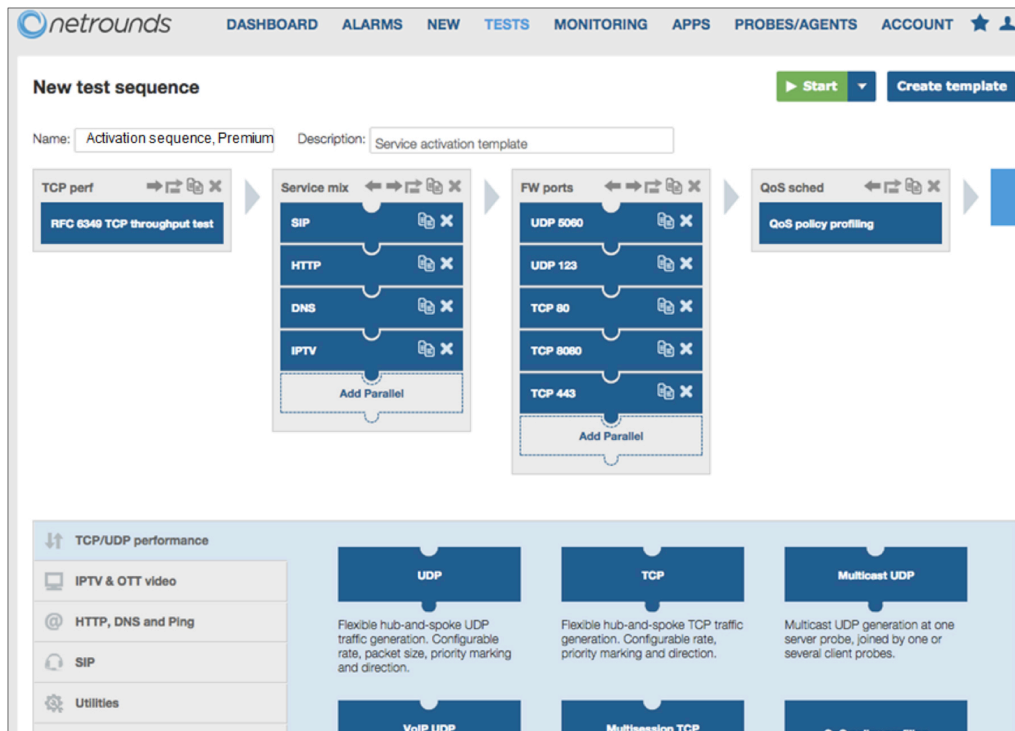
3. Network Services Orchestrator はサービス ライフサイクルにおける顧客の体験を継続的に検証します。稼働中の VNF が更新、再配置、変更されるたびに、Network Services Orchestrator は、サービス検証をトリガーして、顧客の観点から見て変更によってサービスの質が低下していないか確かめます。これはソフトウェア並みの速度で自動発生します。

直接測定を使ったこれら3つの機能はすべて、フルフィルメントと同期してサービス アシュアランスのオーケストレーションを実行する上では必要不可欠で、最終的にサービス提供をエンドツーエンドで自動化します。それでは、オペレータがサービス アシュアランスを構成する方法を詳しく見ていきましょう。

アクティベーション テスト モデルの設計

Network Services Orchestrator はオープン プラットフォームであることから、サービス オーケストレーションの一部としてサードパーティ製のアシュアランス ソリューションを組み込むことも可能です。この場合、オペレータはサードパーティ製の Netrounds ツールを使用して詳細なアクティベーション テストを構成し、サービスを保証するために使用する仮想テスト プロブを設計することができます。図12では、設計された特定のサービスのためにオペレータが構築する、カスタマイズ化されたテストのシーケンスを図示しています。

図12 自動化されたアクティベーション テストの構築



このVPNサービスでは、オペレータは以下を選択します。

- RFC6349 TCPスループット テスト
- SIPメディア ゲートウェイが作動し、音声品質が顧客の期待を
 応えているかを検証するためのサービス関連のテスト
- HTTP DNSのテスト
- IPTVのテスト

オペレータは、サービスで使用される特定のポートを検証するためにもアクティベーション テストを設定します。また、QoSポリシープロファイル テストを設定して、エンドポイントのアプリケーショントラフィックが期待通りの優先順序になっているか検証します。これはすべて、Network Services Orchestratorのサービスデータ モデルから得ている点にご留意ください。これで、サービス用に定義された属性は、テスト用ドメインでモデル化できます。

この設計テスト シーケンス全体は、これでテンプレートとしてキャプチャされ、Network Services Orchestratorによって同じサービス データ モデル内で、オーケストレーションされたフルフィルメント プロセスの一部として使用可能です。

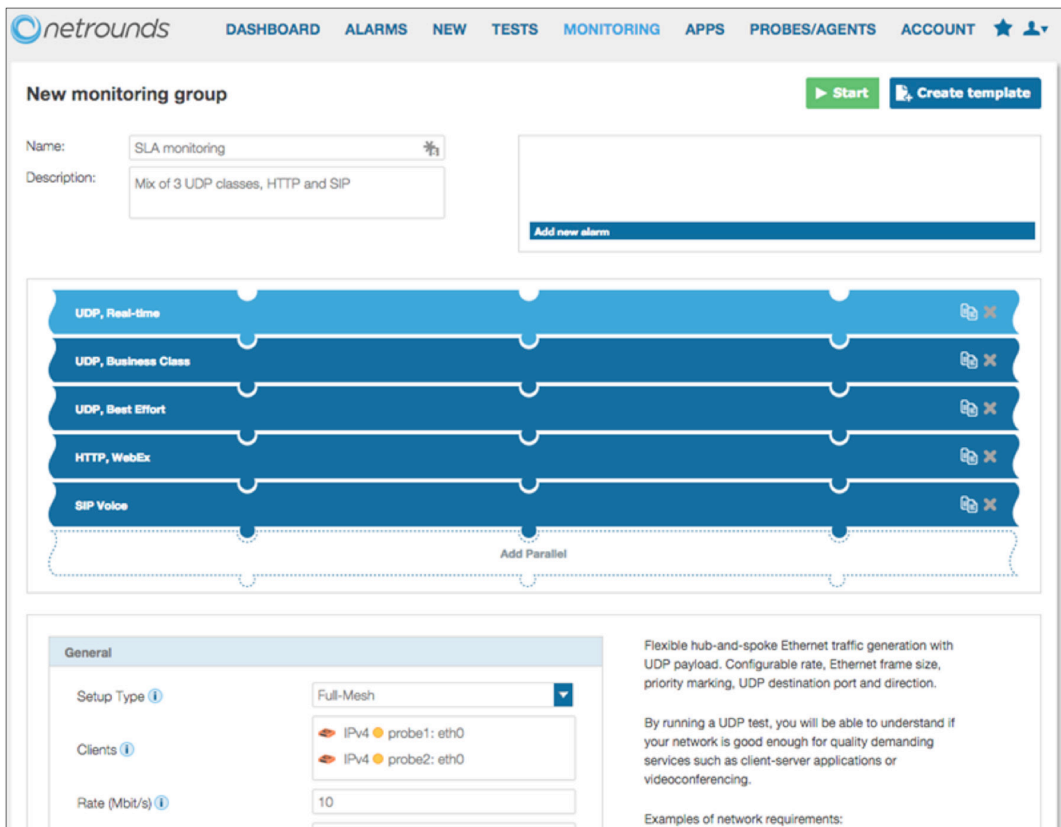
Network Services Orchestratorは、テンプレート内でサービスおよび顧客に固有のパラメータ（定義されたSLAしきい値など）を入力変数として提供し、テスト エージェントがテストやその他の関連するパラメータでそれを使用するようになっています。続いて、サービスがアクティブ化された時点でこのテスト シーケンスを自動的に開始します。またこれは、サービスのライフサイクル全体で、VPNパスの変更が発生するたびに実行されます。

継続的なSLAモニタリングのモデル化

同様な方法で、顧客のSLAに対するVPNを継続的に保証することもできます。ここでも、オペレータはサードパーティ製のNetrounds ツールを使用して継続的なSLAモニタリングを構成します。この場合、このUDPキュー、HTTP、SIPエンドポイントをモニタリングします（図13）。

アクティベーション テスト シーケンス同様に、こうしたモニタリング動作は、Network Services Orchestratorが継続的なSLAアシュアランスを実施する際に使用するテンプレート内でキャプチャされ、ここでも、同じサービス データ モデルの一部になります。

図13 SLAアシュアランス モニタリングの定義



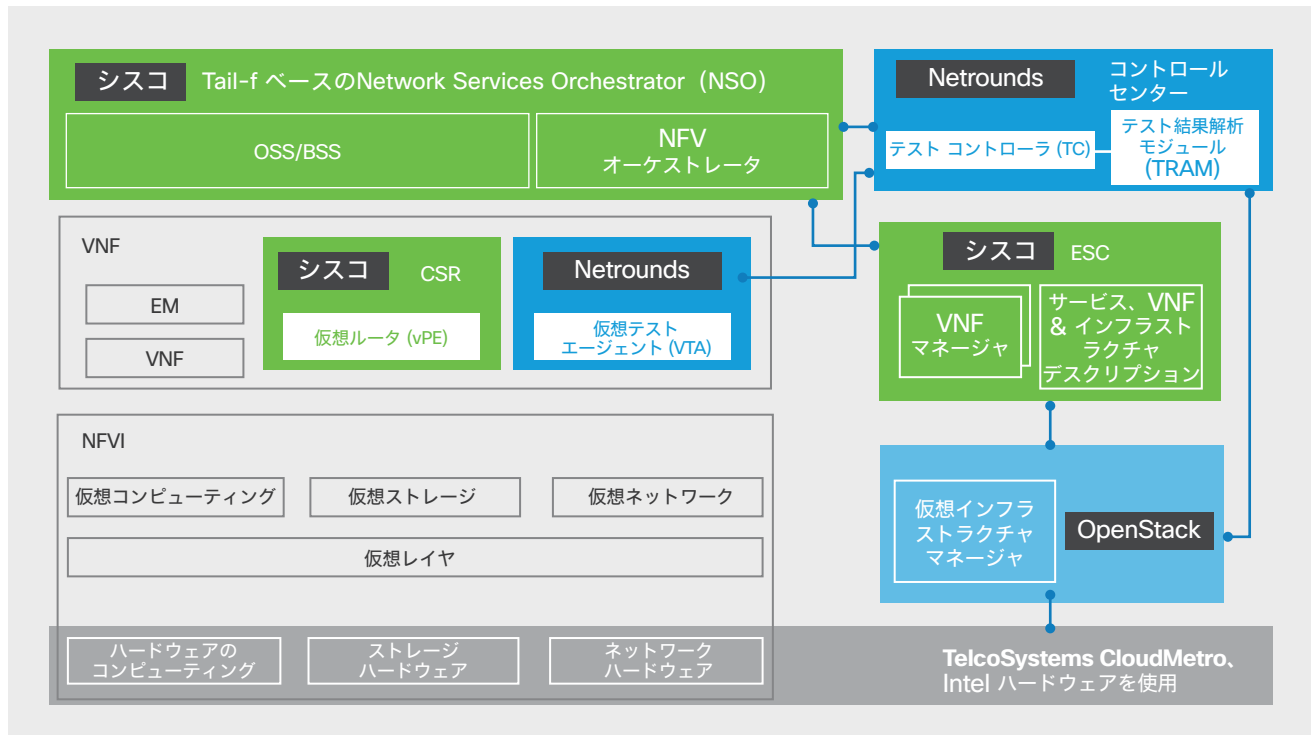
アクティベーション テストおよびアシュアランスのインスタンス化

図 14 では、VPN サービスを提供するために、Network Services Orchestratorがオーケストレーションされたアシュアランス機能を、ETSI NFV MANO フレームワークにどのようにマップしているのかをハイレベルな視点で図示しています。

この場合、Network Services OrchestratorはOSSおよびNFVオーケストレータとして振る舞い、Cisco Elastic Services

ControllerはVNF マネージャ、仮想インフラストラクチャ マネージャ (VIM) はOpenStackとなります。オペレータは、一元型のテストコントローラと仮想テスト エージェントを利用するためにNetroundsを使用しています。VPN サービスでは、仮想宅内機器ルータ、およびIntel ハードウェア上でTelcoSystems CloudMetroによるCPEのコンピューティングが実行されているため、シスコ仮想クラウド サービス ルータ(クラウド サービス ルータ 1000V シリーズ) を使用します。

図 14 ETSI NFV リファレンス アーキテクチャへのマッピング

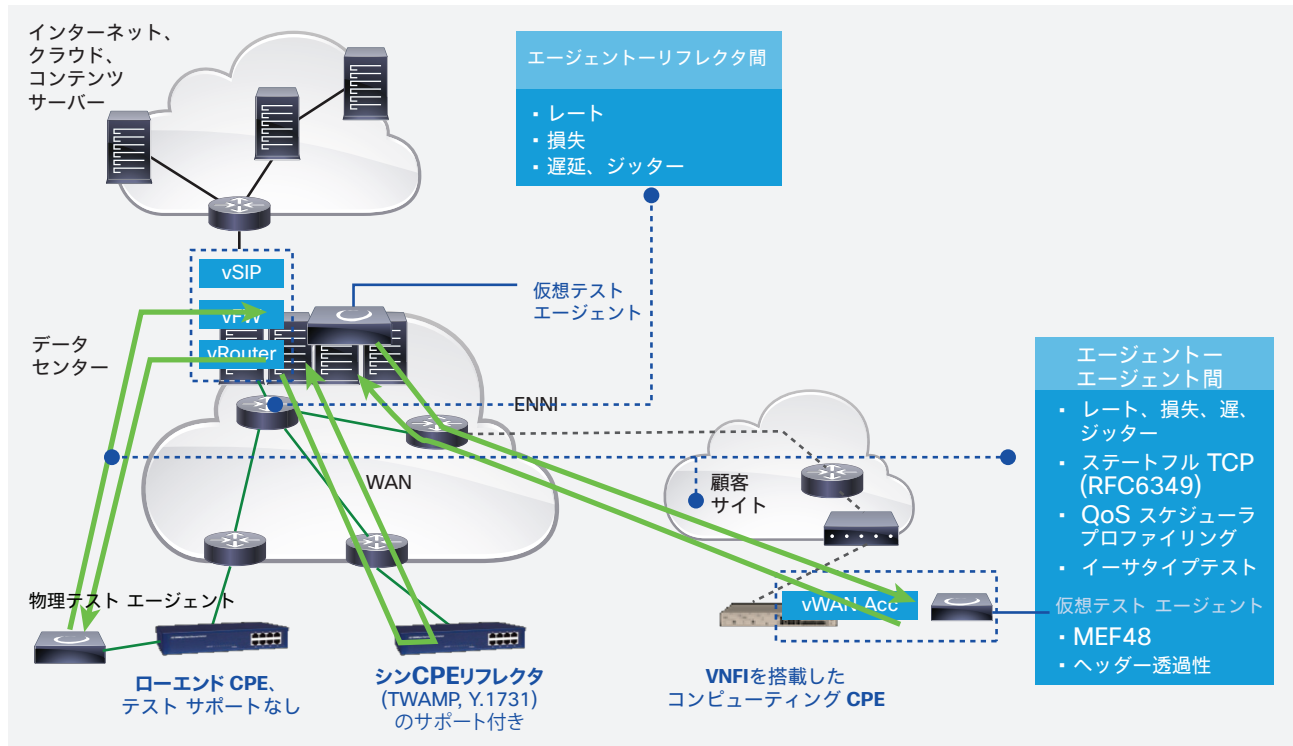


NFV インフラストラクチャを持つ顧客サイトのエンドポイントの場合、オペレータはフルパワーのCloudMetro CPEを使用して、アクティベーション テストとSLA アシュアランスのモニタリングを実行します。テスト能力がないローエンドのCPEを対象にするには、オペレータは同じコードを実行している廉価な物理x86 サーバーを追加します。リフレクタサポート付のシンCPEエンドポイントの場合、オペレータはその他のエンドポイントのように、アクティベーション テストとSLA モニタリングの

全機能を実行することはできません。しかし、標準的な接続性テストを用いて行うよりも、アシュアランス システムは直接かつより一貫したサービス エクスペリエンスを測定することができます。

図 15 では、テストおよびアシュアランス機能の全範囲を図示しています。Network Services Orchestratorによってオペレータはオーケストレーションされたすべてのサービスを利用できるようになります。

図 15 ネットワーク性能の測定



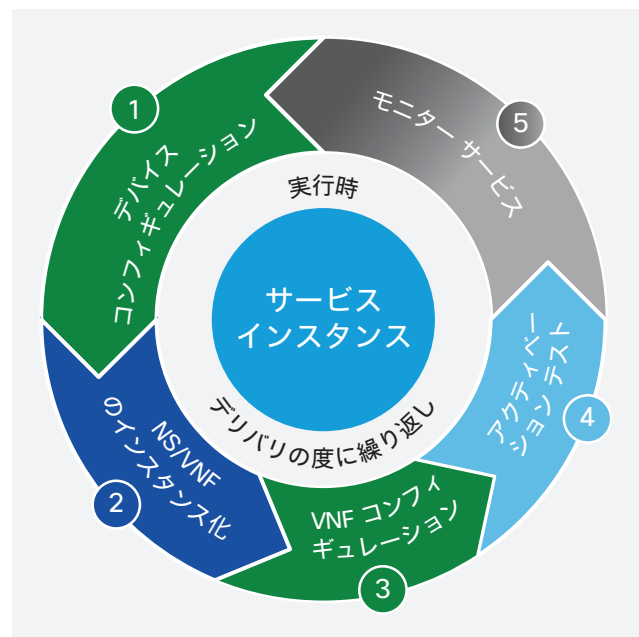
ローエンドのCPEと顧客サイトのエンドポイントの両方において、オペレータはここでVPNサービスの性能に関するきめの細かい測定基準（ステートフルTCPと高度なQoSテストなど）を、顧客がサービスを受けるときに直に取り込むことができます。シンCPEの場合でも、TWAMPとY.1731でスループット、遅延、ジッターといったネットワークの重要な性能測定基準を、リアルタイムで行えます。

アクティベーション テストおよび継続的なサービス アシュアランスの一環として、オペレータはここでサービス チェーン全体でサービスを検証することもできます(例えば、ダウンロード率、応答時間、TCP 接続回数の測定など)。また、こうしたテストとアシュアランス プロセスはすべてここで、同じデータ モデルを使用して Network Services Orchestrator内のサービス フルフィルメントと同期されて自動化されます。

サービス アシュアランス機能のオーケストレーション

ここで紹介するのは、オペレータがVPNサービス用に構成する詳細なアシュアランス運用のうちの一つかとなります。では、VPNサービスの一部として Network Services Orchestratorがどのようにしてオーケストレーションを実行するのか見て行きましょう(図 16)。

図 16 オーケストレーションでの自動化



最初に、サービス データ モデルに基づき、Network Services Orchestratorはデバイスコンフィギュレーションを実行して、顧客固有のVPNを作成し、サービス データ モデルのパラメータをデバイス データ モデルのパラメータに自動的に変換します。Network Services Orchestratorは、VPNサービスで使用されているデバイスを設定し、また仮想PEルータと仮想テストエージェントを含めた仮想機能もインスタンス化して設定します。

このプロセスの一環として、Network Services OrchestratorはMANOネットワーク サービスディスクリプタ (NSD) を使用して、これらの仮想プロブをネットワーク内の適切な場所に自動的に配置します。この点が、顧客のネットワークで変更が生じた際、手動で再配置する必要がある従来のテスト プロブとは大きく異なります。Network Services Orchestrator のステートコンバージェンスでは、顧客が移動したり、VPNサービスにエンドポイントを追加すると、仮想プロブとサービスディスクリプタのすべてがネットワーク内の適切な場所に自動的に同調して移動します。

VPN サービスの仮想デバイスが起動されて実行されると、Network Services Orchestratorはサービス固有の設定を作成します。

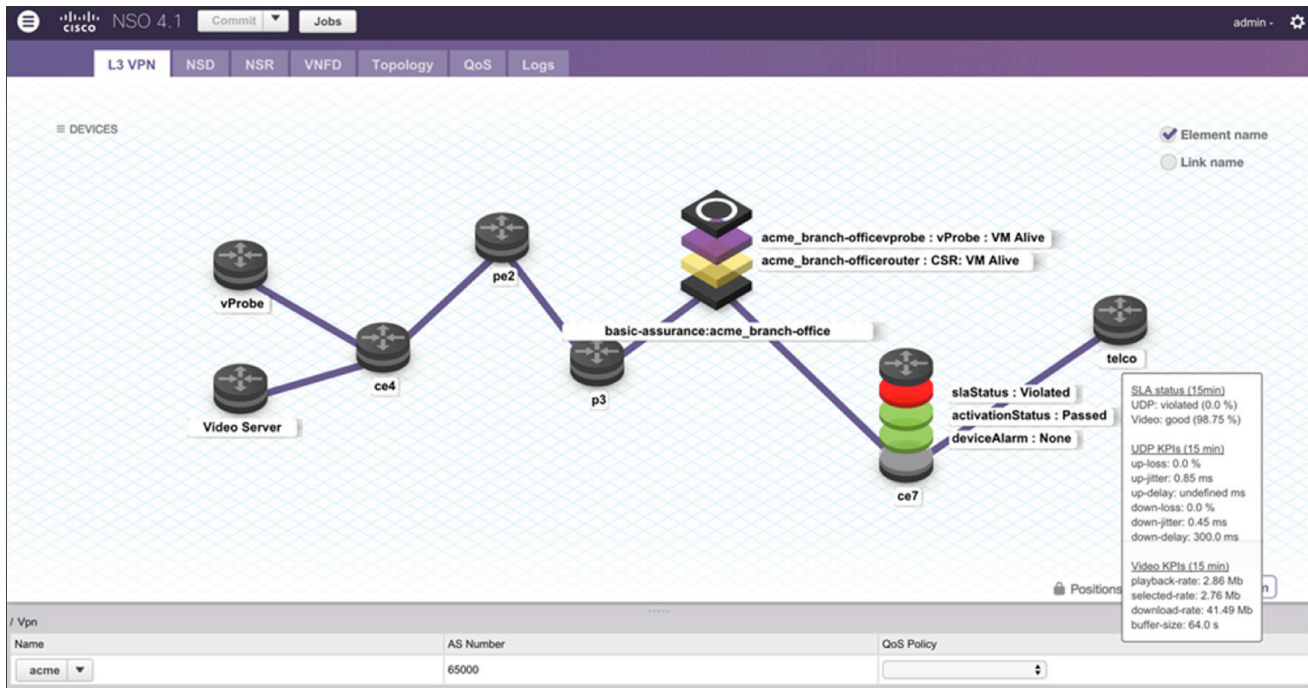
また、オーケストレーションが実行されたトランザクションとデータモデルの一環として、アシュアランス システムをトリガーし、VPNと顧客サイトのサービスの両方が期待されている性能を發揮しているかどうか検証します。

これは、まずオペレータが設定したサービス固有のアクティベーション テスト シーケンスと並行して始まります。いったん完了すると、Network Services Orchestratorは、継続的なSLAモニタリングに備えてアシュアランス システムに受け渡します。Network Services Orchestratorでは、VPNトポロジ、エンドポイント、そして定義済みのSLAパラメータが利用できるため、アシュアランス システムはこれで、顧客のサービスに備えて継続的にSLAと比較してKPIを直接測定することができます。

Network Services Orchestratorのユーザーインターフェイスで見た、アシュアランスを備えたサービス オーケストレーション

図 17では、Network Services Orchestrator UIを使用するオペレータから見た流れを図示しています。ここでは、Network Services OrchestratorがVPNサービスのトポロジをリアルタイムで表示します。この中には、顧客エンドポイント3つ (CEデバイス)、仮想プロバイダー エッジ ルータ、仮想テスト エージェントが含まれています。

図 17 Network Services Orchestratorでのオーケストレーションされたアシュアランス



中央右にある四角は、NFV MANOにおける仮想サービスのインスタンス化を担うNFVのオーケストレータを表しています。クリックすると、ユーザーはシスコの仮想クラウド サービス ルータと仮想テスト エージェントは起動しており、直接測定に備えて稼働中のテスト プローブがネットワーク内の適切な場所に配置されていることを確認できます。

右手には、顧客のCPEにリンクしているVPN内の区間がUIトポロジで表示されています。そのデバイスをクリックすると、ユーザーはアシュアランス情報をリアルタイムで確認することができます。この中には、SLAステータスやエンドポイントで実行されたアクティベーション テストの履歴も含まれます。緑色の円形

は、デバイスでアラームが発生していないことを表しています。また、オーケストレーションの実行時には、アクティベーション テストシーケンスを一時停止します。上記の図にある赤い円形は、VPNサービス内のそのエンドポイントにおいてSLA違反があったことを表しています。

エンドポイントにマウスオーバーすると(さらに右手にあるボックスを参照)、オペレータはVPNにおけるビデオとネットワーク両方について、現在のサービス固有の実行時のKPIを確認できます。Network Services Orchestratorでは、過去のある時期に取得されたデータに基づく推論ではなく、リアルタイムの直接測定が表示される点にご留意ください。

まとめ

急速に近づくネットワーク サービスの未来において、顧客体験がどのようなものになるかはすでに明らかです。オペレータは新しいサービスを簡単に設計できるようになり、物理および仮想インフラストラクチャの複雑性、難解な管理システム、ならびに多様なネットワーク ドメインは消えてなくなります。こうしたサービスは、各顧客のニーズや好みに応じてカスタマイズされます。サービスは、顧客の観点から見て期待通りであることを保証するために、常時モニタされます。また、こうしたサービスは要求に応じて、分単位、秒単位のごくわずかな時間で顧客に提供されます。

ライフサイクルの自動化を成功させる4つの柱を実装した、Network Services Orchestratorは未来のネットワーク サービスを今すぐ提供できます。共通の形式データ モデルとデータ モデル マッピングを使用して、サービスをプログラマティックにアクティベーションし、手動でコードを作成する必要はありません。高レベルのサービス インテントをネットワーク活動に変換するステート コンバージェンスを搭載したNetwork Services Orchestratorは、ネットワーク内になんらかの変更があれば自動的に応答します。

Network Services Orchestratorはまた複数のドメインにおいて、サービスのオーケストレーションをエンドツーエンドで実行します。また、フルフィルメント、アクティベーション テスト、継続的なサービス レベルのアシュアランスを含め、1回のトランザクションでサービスのライフサイクル全体に対応します。

これらはNetwork Services Orchestratorが近い将来実現するかもしれない計画された機能ではありません。オペレータがすぐに導入して迅速性と俊敏性を得られるように、既に実行コードに実装されています。

Network Services Orchestratorの詳細については、シスコのアカウント マネージャにお問い合わせいただくか、<http://www.cisco.com/web/JP/product/hs/netmgt/network-services-orchestrator/index.html> をご覧ください。