

vFORUM 2019

NS189

VMware で実現する、
クラウドネイティブ時代の
アプリケーションとネットワークの融合

NSX コンテナネットワーキングと NSX Service Mesh のご紹介

ヴァイエムウェア株式会社

ソリューションビジネス本部

ネットワーク & セキュリティ技術部

シニアスペシャリストエンジニア 中奥 洋志彦

Make Your Mark

Agenda

クラウドネイティブ、マイクロサービス、Kubernetes
Kubernetes ネットワーキングを実現する NSX-T Data Center

サービスメッシュは何を解決するか

NSX Service Mesh

デモ

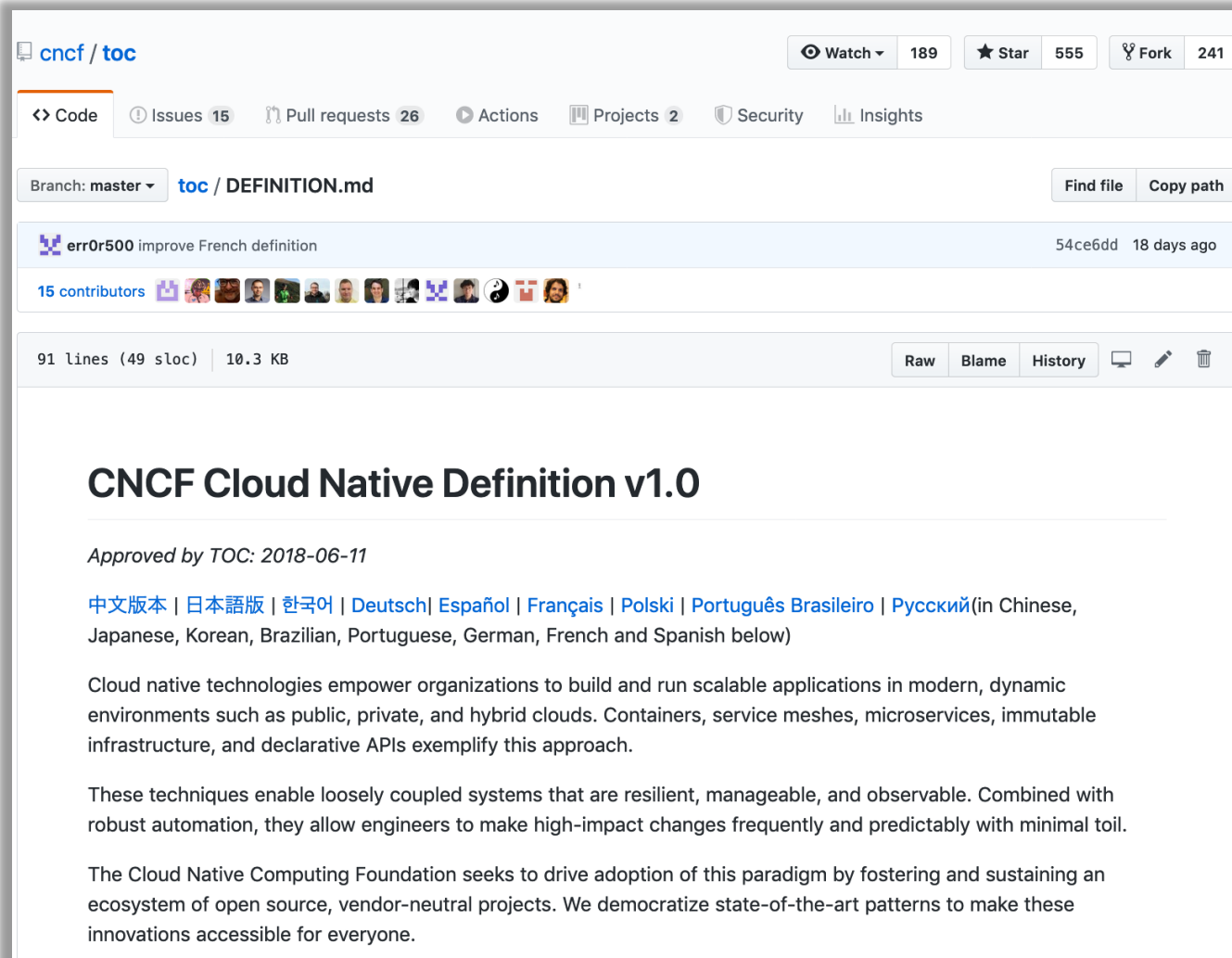
まとめ

クラウドネイティブと マイクロサービス、コンテナ

Kubernetes と NSX-T コンテナプラグイン

クラウドネイティブとは？

<https://github.com/cncf/toc/blob/master/DEFINITION.md>



cncf / toc

Watch 189 Star 555 Fork 241

Code Issues 15 Pull requests 26 Actions Projects 2 Security Insights

Branch: master toc / DEFINITION.md

Find file Copy path

err0r500 improve French definition 54ce6dd 18 days ago

15 contributors

91 lines (49 sloc) 10.3 KB

Raw Blame History

CNCf Cloud Native Definition v1.0

Approved by TOC: 2018-06-11

中文版 | 日本語版 | 한국어 | Deutsch | Español | Français | Polski | Português Brasileiro | Русский (in Chinese, Japanese, Korean, Brazilian, Portuguese, German, French and Spanish below)

Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.

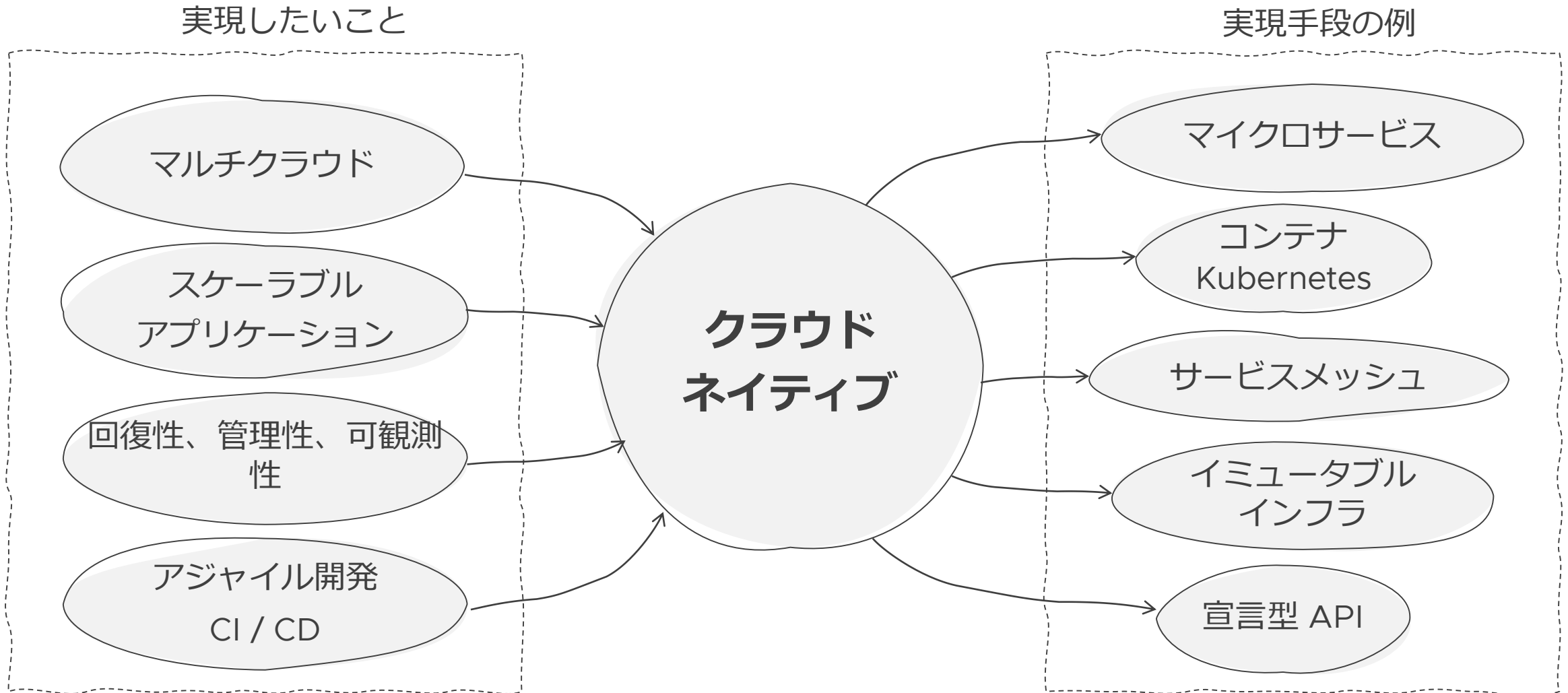
These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.

The Cloud Native Computing Foundation seeks to drive adoption of this paradigm by fostering and sustaining an ecosystem of open source, vendor-neutral projects. We democratize state-of-the-art patterns to make these innovations accessible for everyone.

クラウドネイティブ技術は、パブリッククラウド、プライベートクラウド、ハイブリッドクラウドなどの近代的でダイナミックな環境において、スケーラブルなアプリケーションを構築および実行するための能力を組織にもたらしめます。このアプローチの代表例に、コンテナ、サービスメッシュ、マイクロサービス、イミュータブルインフラストラクチャ、および宣言型 API があります。これらの手法により、回復性、管理力、および可観測性のある疎結合システムが実現します。これらを堅牢な自動化と組み合わせることで、エンジニアはインパクトのある変更を最小限の労力で頻繁かつ予測どおりに行うことができます。

クラウドネイティブを実現するための手段

マイクロサービス、コンテナ、サービスメッシュ



アプリケーションの変革

Application Transformation

モノリシックな アプリケーション



複雑性 – 成長にともない複雑さが増大し、全体の理解が困難になる

俊敏性を損なう – 長いリリースサイクルと長いチェンジウィンドウ

可用性の低下 – 一つのバグがシステム全体に影響を及ぼす可能性

技術革新の遅れ – 実行環境のスタックに対する長期間のコミットメントが必要になる

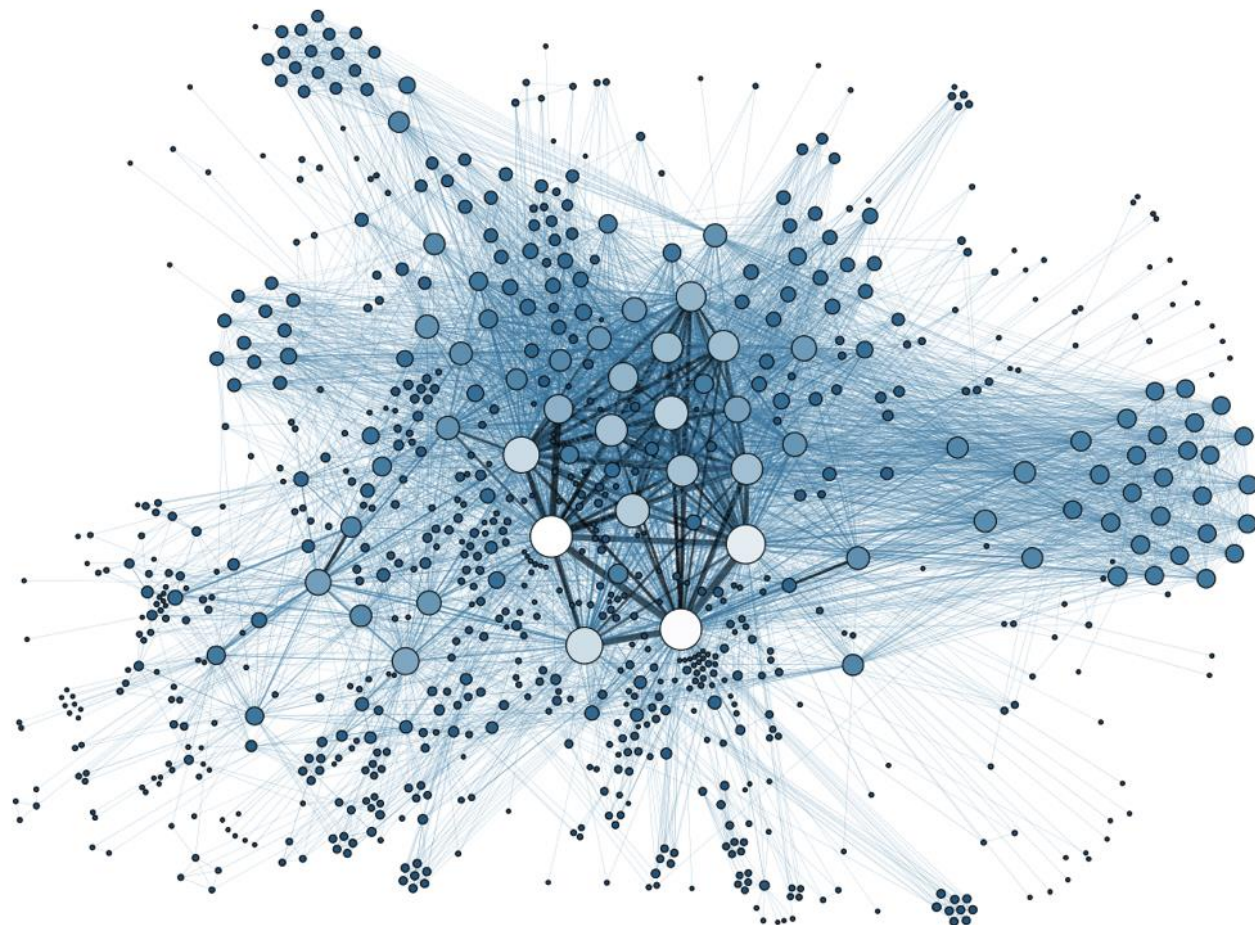
マイクロサービスとは？

マイクロサービス != コンテナ

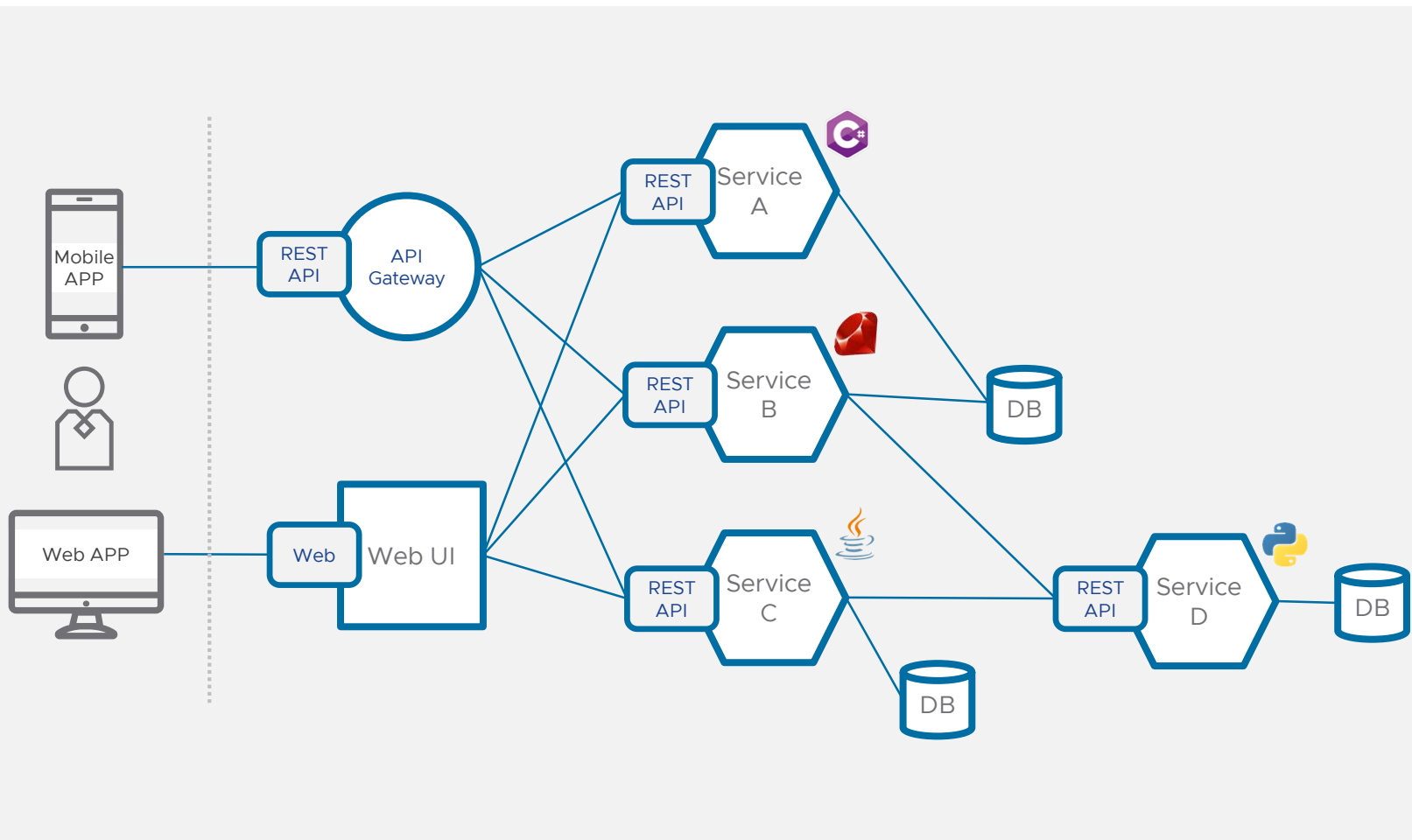
アプリケーションの機能を、異なるチーム、異なる言語で開発された小さなソフトウェアの単位に分割するという考え方

マイクロサービスの間では、言語非依存のAPIを用いて通信を行う

マイクロサービスを提供するホストはVMでも構わないが、より小さなフットプリントで実現できるコンテナの方が適している



マイクロサービス アーキテクチャの利点



シンプル – 各サービスが独立しており、個別のアップグレードも可能

柔軟性 – 水平展開が容易で、多様なプラットフォームへの展開も可能

回復性 – 故障影響範囲を最小化し易い

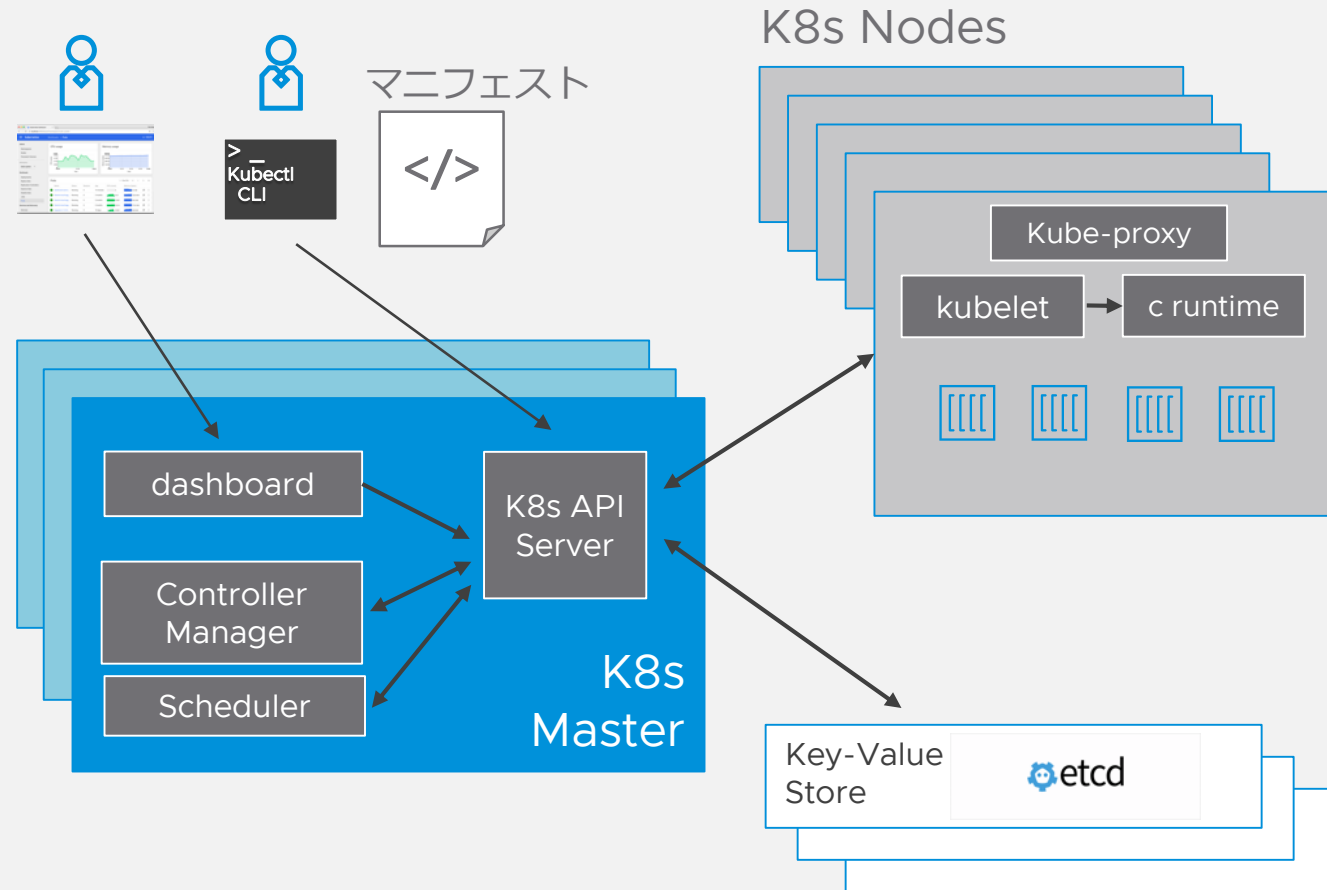
革新性 – 新しい技術を迅速に展開できる、新しいフレームワークや開発言語への適合

Kubernetes (K8s)

コンテナを管理するためのデファクトソリューション



kubernetes



アプリケーションコンテナを管理するためのオープンソースプラットフォーム

K8s クラスタ上でのコンテナのデプロイやスケーリング、ライフサイクル管理を自動化

コンテナを動作させるためにインフラをプロビジョニング

2014年に Google がリリース

現在は CNCF がメンテナンスを行う

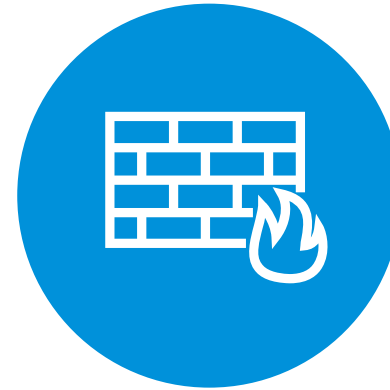
NSX-T Kubernetes インテグレーションのゴール



開発者の
邪魔をしない！



Kubernetes を企業
ネットワークに統合



包括的なファイア
ウォールポリシー



視覚化とトラブル
シューティングツール

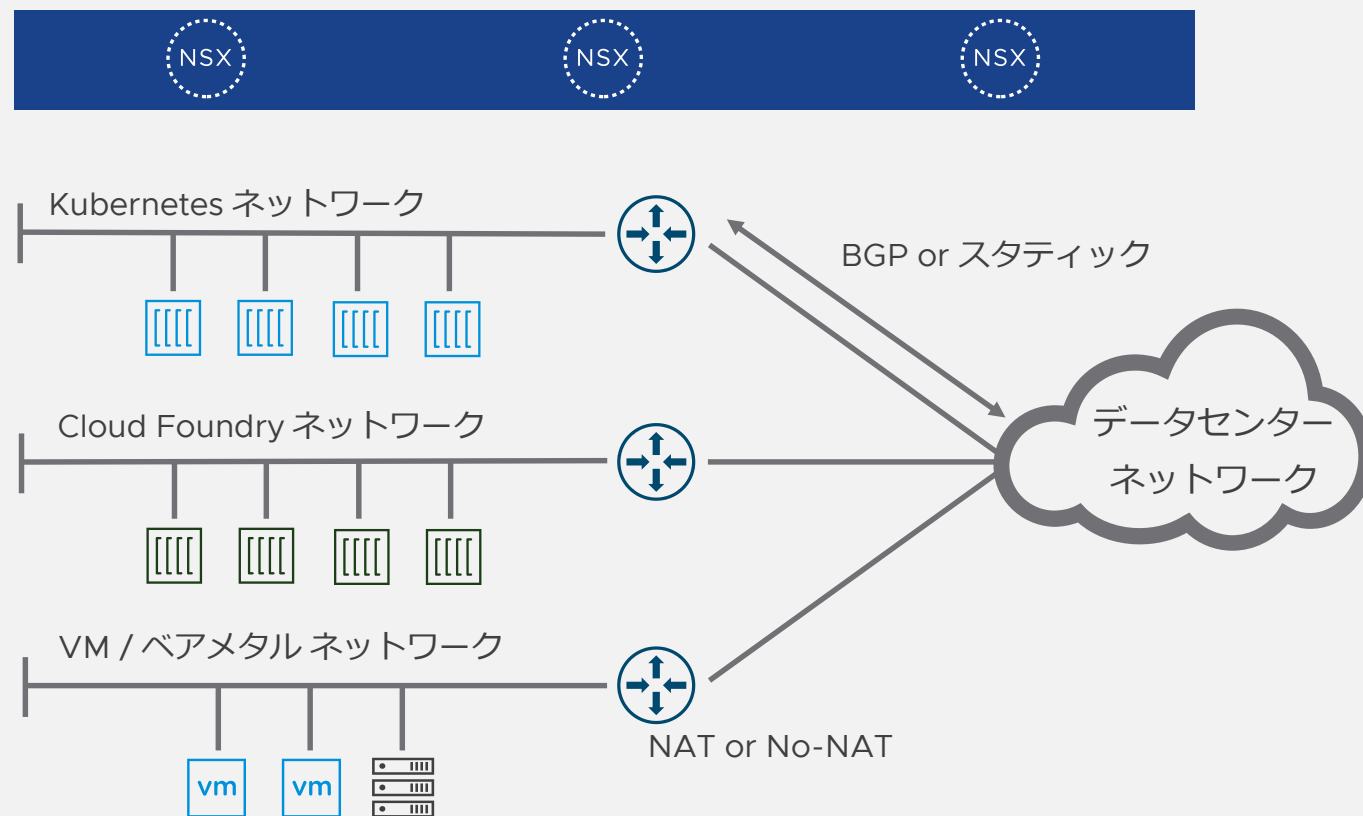
K8s コンストラクトを
マッピング

コンテナ、VM、
およびすべての
エンドポイントの
セキュリティを強化

企業のコンテナ導入を
容易にする

ネイティブなコンテナネットワークングとルーティング

多様なエンドポイントを含むデータセンターネットワーク



機能

データセンターネットワークに
Kubernetes コンテナを接続

柔軟な L2 + L3 ネットワークトポロジ

- ルーテッドトポロジ
- IP 空間の重複を可能にする NAT トポロジ
- NAT と No-NAT のハイブリッド

利点

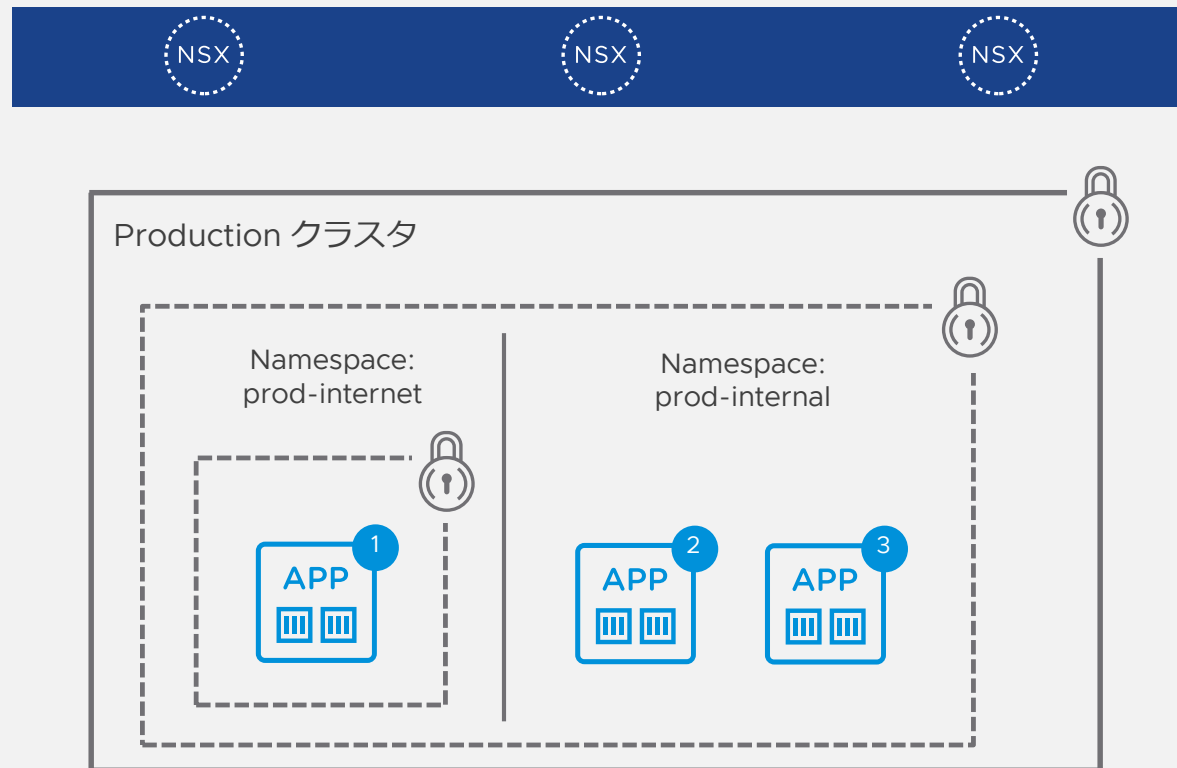
すべてのアプリ基盤のための共通なネット
ワーキングモデル

シンプルな Day 2 運用

コンテナワークロードのスケラビリティ
に対応

コンテナのためのマイクロセグメンテーション

コンテナのコンプライアンスを実現する細やかなセキュリティ



機能

コンテナのインタフェースに適用できるステートフルな分散ファイアウォール

K8s クラスタ内だけでなく、クラスタをまたがるファイアウォールルールも定義可能

ポリシーは K8s ネットワークポリシーを用いて、もしくはセキュリティ管理者自身で定義することが可能

VM とセキュリティポリシーを共用できる

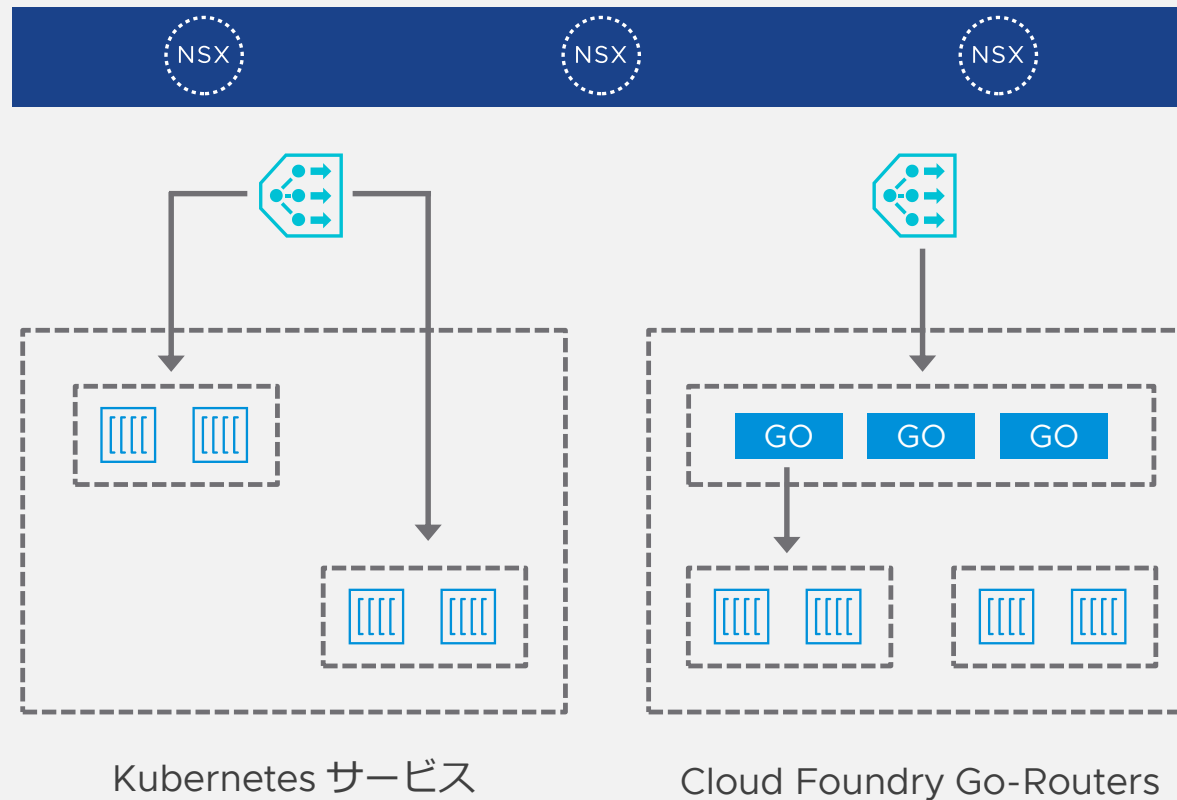
利点

多様なセキュリティ運用モデルに対応できる

セキュリティ運用の負担を削減

コンテナのためのロードバランシング

ソフトウェア定義のロードバランサ



機能

仮想もしくはベアメタルアプライアンスの上で提供

以下のサービスをサポート:

- Service type ClusterIP / NodePort / LoadBalancer
- Ingress

利点

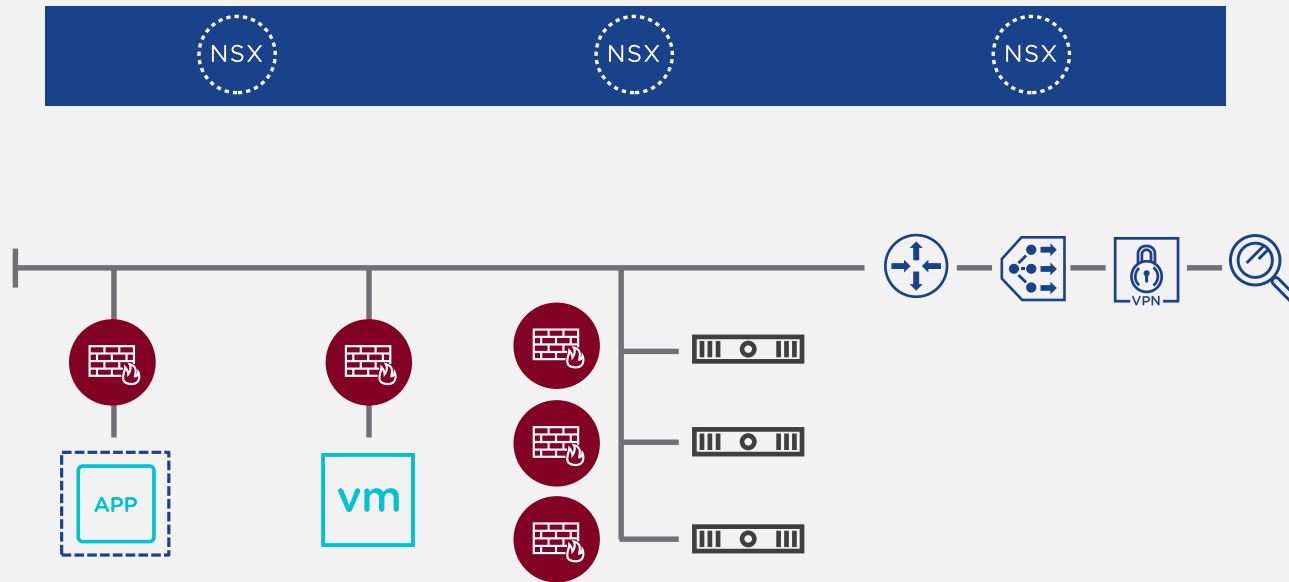
K8s からサービスを提供するための自動化されたワークフロー

サービス提供に関わる時間を削減

スケールアウト可能なソフトウェアロードバランサにより、ロードバランサのコストを抑制

VMware® PKS、OpenShift をサポート

プラットフォームをまたいだネットワーク管理機能



機能

コアプラットフォーム

Syslog、ダッシュボード機能、検索機能、Traceflow、パケットキャプチャ、Central CLI

AAA/RBAC

AD/Radius 連携、2段階認証、事前定義の権限設定

ロギング統合

vRealize LogInsight, Splunk

視覚化

vRealize Network insight, NSX Intelligence

利点

MTTR の短縮

監査とコンプライアンス

NSX-T Data Center が提供する Kubernetes ネットワーク機能

Kubernetes 環境におけるネットワーク機能を統合

	VMware	オープンソーススタックの例
L2 機能	VMware NSX-T™ Data Center 論理スイッチング	Flannel (East / West Pod トラフィック)
L3 機能	論理ルーティング	Calico L3 機能 / L4 機能 (IP Tables)
セキュリティ機能	分散ファイアウォール	
ロードバランシング	論理ロードバランサ	NGINX/HA Proxy ロードバランシング
VM との相互接続性	同じ論理ネットワーク上で同じポリシーを用いて VM との相互接続が可能	一般的な手法による VM との相互接続
エンドポイント管理	Port Connection, Traceflow	エンドポイント間の視覚化や トラブルシューティングが複雑
既存運用ツールの利用	VMware vRealize® 製品との相互運用性	新しい運用ツールが必要

サービスメッシュは 何を解決するか

NSX Service Mesh

マイクロサービス: 数多くの利点と、新しい課題

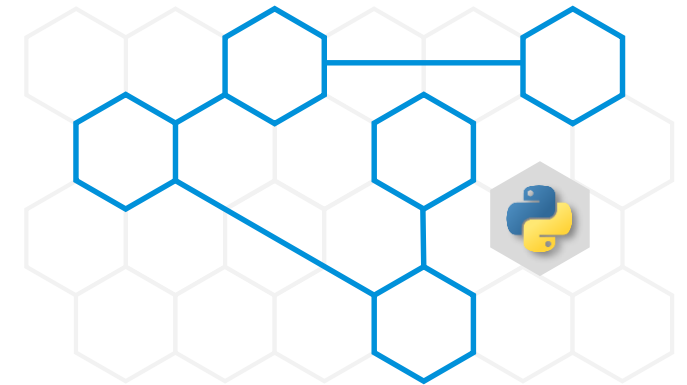
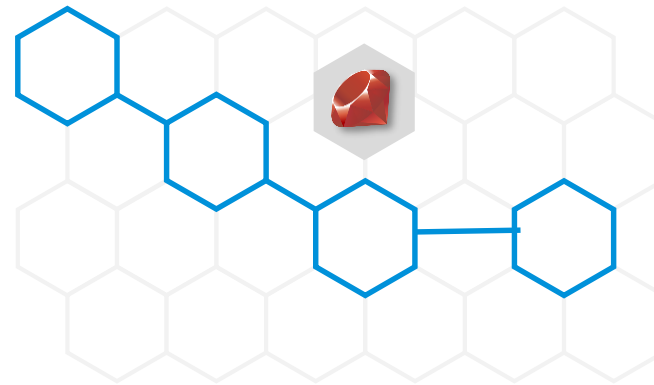
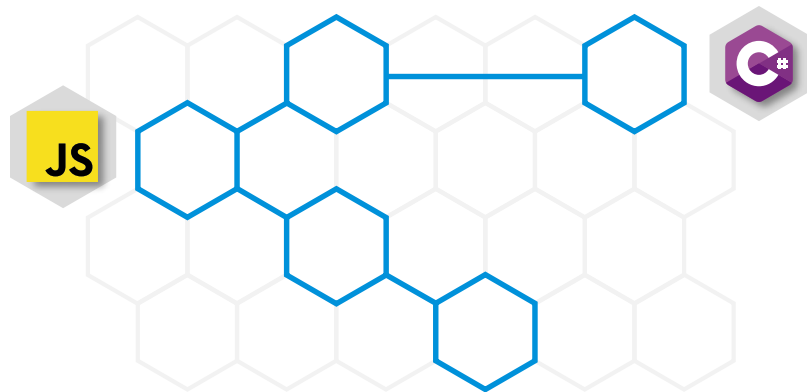
いかにして安定的にマイクロサービスを監視、制御し、セキュリティを担保するか

多言語・多フレームワーク
への習熟

セキュアな接続性と
トラフィック制御

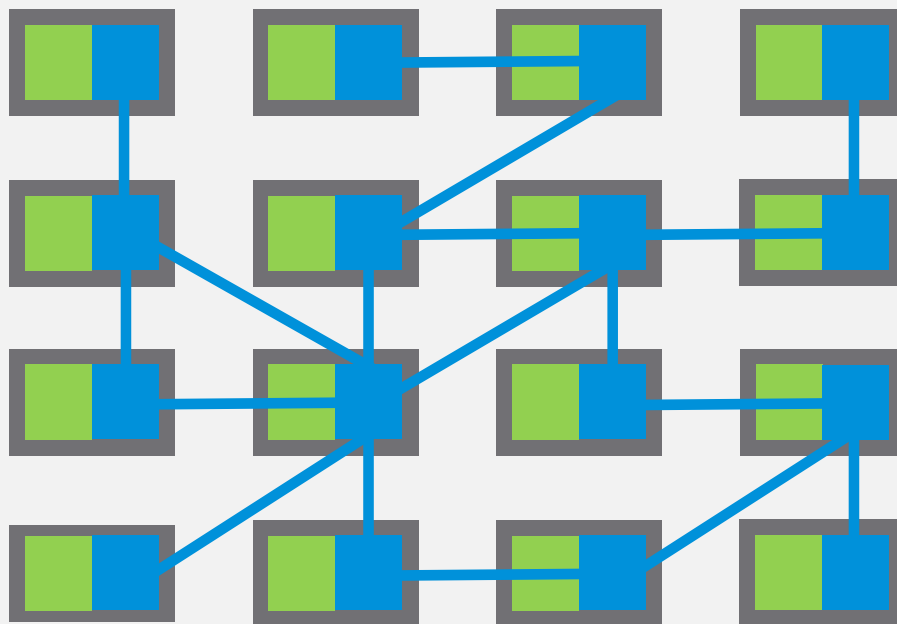
可観測性 - マイクロ
サービスの正常性の監視

プラットフォームをまた
がる視覚化と監査



サービスメッシュ

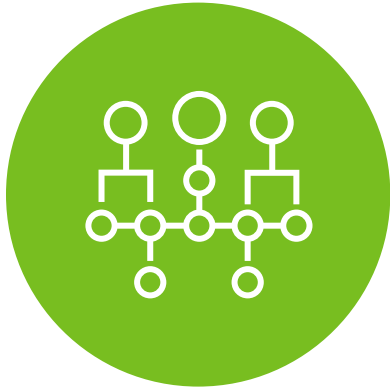
インフラ・ネットワークに依存せずにマイクロサービスの課題を解決



サイドカープロキシ

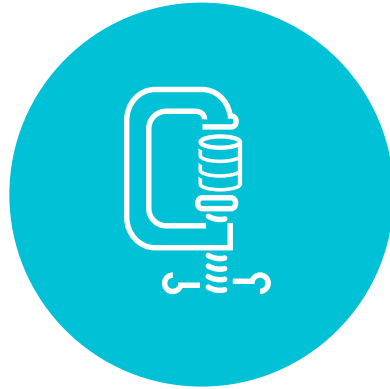
- 共用インフラから切り離された専用の通信レイヤー
- サービス間通信を扱う
- 複雑なサービス間トポロジを管理
- 軽量なネットワークプロキシの配列
- アプリケーションプロセスと一緒に展開される
- アプリケーション側で認識する必要がない

サービスメッシュの利点



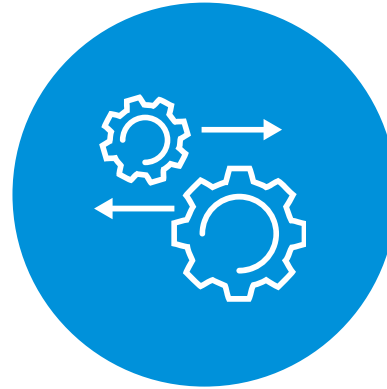
サービスの検出

サービスは互いを見つけることができる



回復性

ビルトインの堅牢なフレームワーク、ロードバランシングとテスト機能



設定の柔軟さ

サービスの実行時に動的に設定できる



可観測性

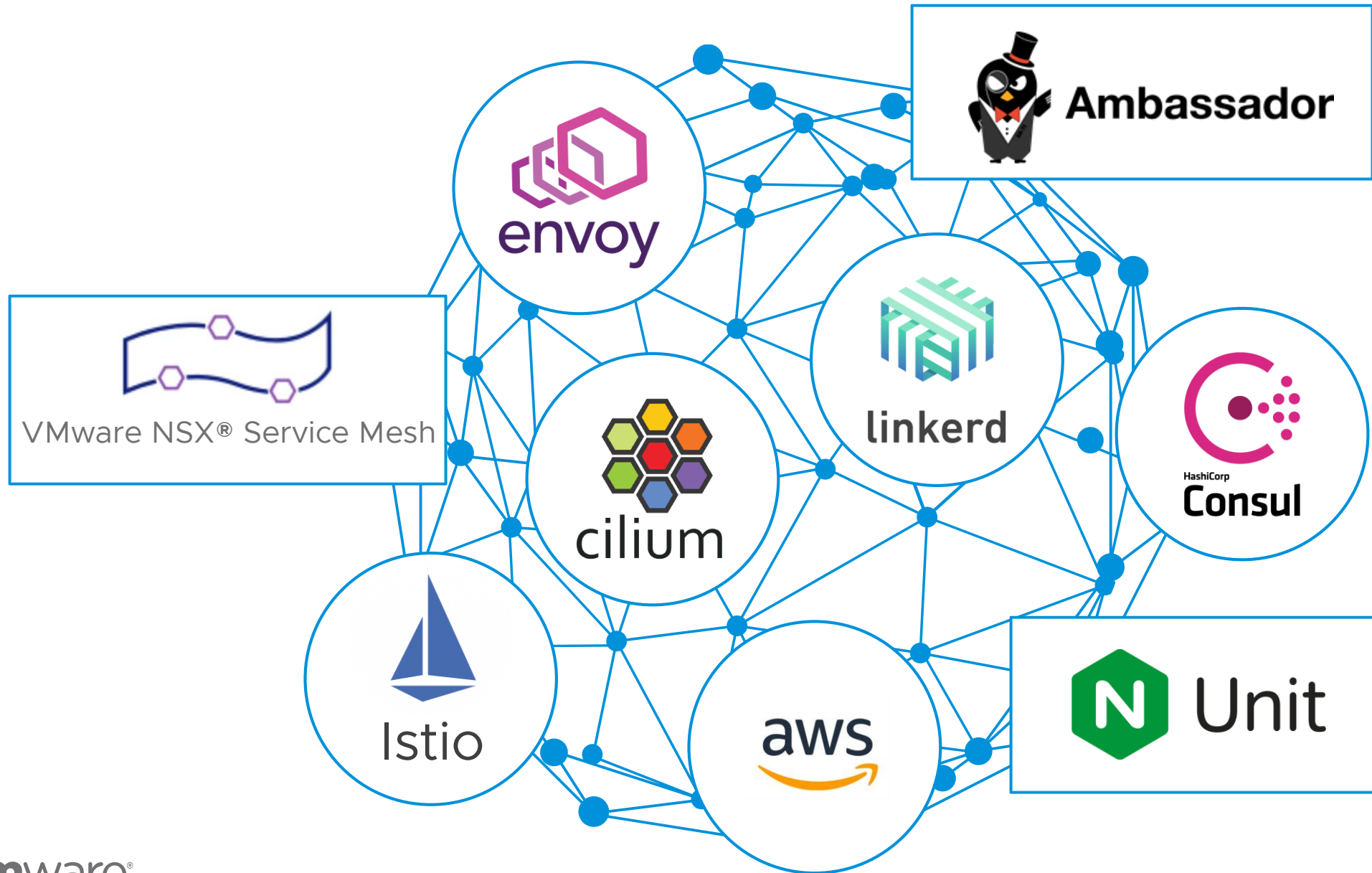
標準的なメトリック、ロギング、監視と分散トレーシング



セキュリティ

サービス間通信を暗号化して保護

サービスメッシュの世界



Istio とは

istio / istio

Watch

845

★ Star

15,816

Fork

2,512

<> Code

! Issues

834

Pull requests

166

Wiki

Insights

Connect, secure, control, and observe services. <https://istio.io>

microservices

service-mesh

lyft-envoy

kubernetes

api-management

circuit-breaker

polyglot-microservices

enforce-policies

proxies

microservice

envoy

consul

nomad

request-routing

resiliency

fault-injection

7,069 commits

26 branches

45 releases

333 contributors

Apache-2.0

コントロールプレーン

Google

IBM

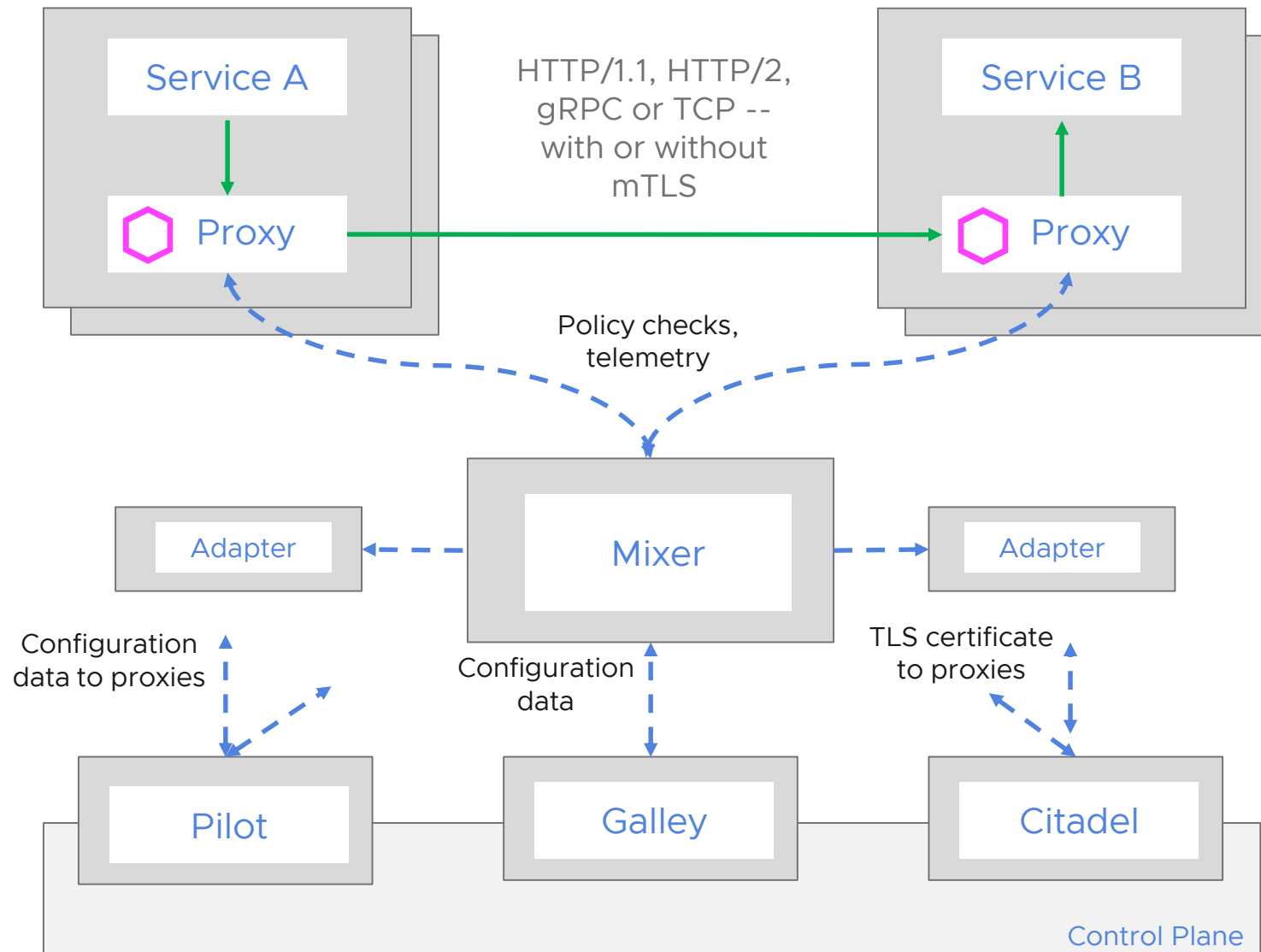
データプレーン

lyft

2017年5月にプロジェクト開始

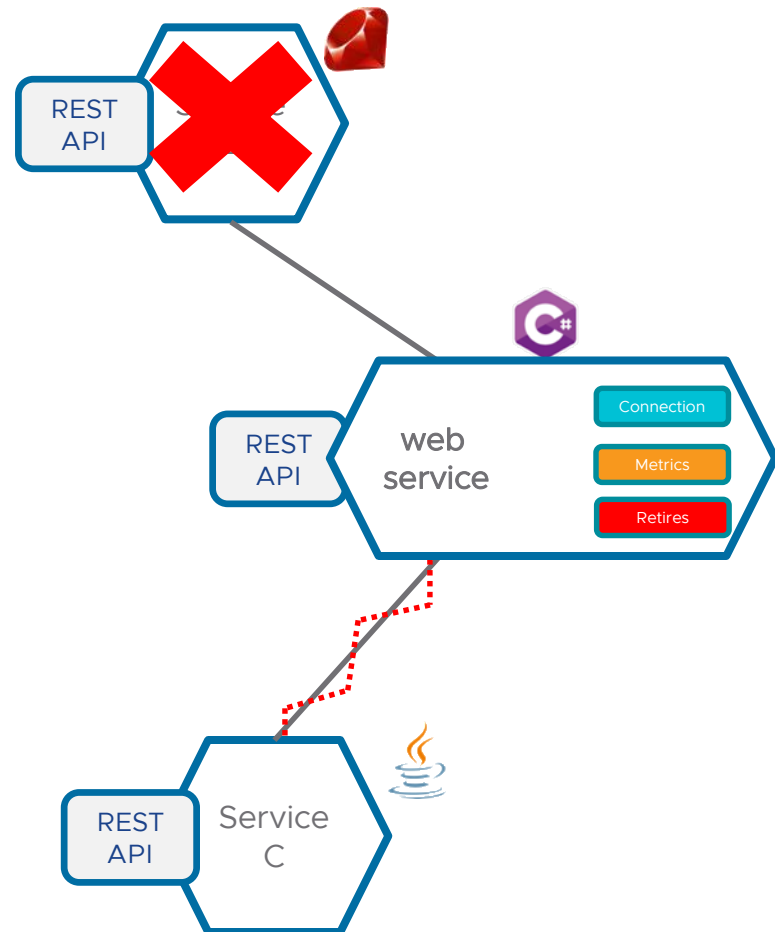
2018年7月に 1.0 に到達

Istio アーキテクチャ



サービス間コネクション管理のオフロード

サービス検出、暗号化、エラー検出とモニタリング



他のサービスとの連携

サービスの検出

コネクション情報 (通信の秘匿性、暗号化)

異なる言語のサポート

エラーハンドリング

ビジネスロジック – データを取得して Web に表示する

遅延の検出と対応

メトリックの収集

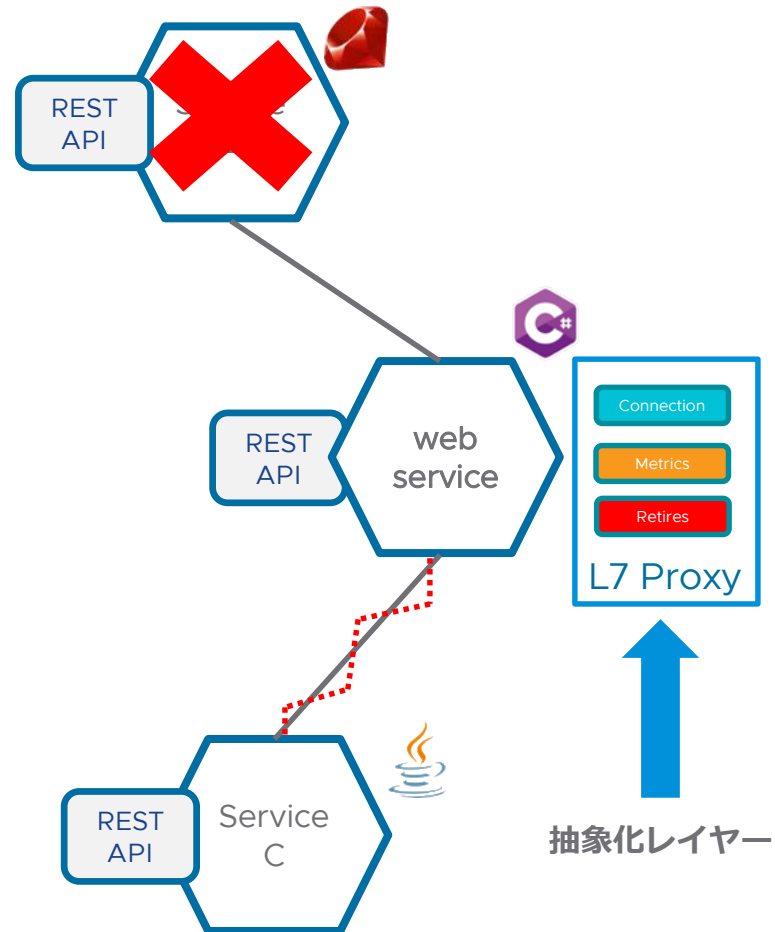
メトリックの送信

セルフヒーリング

エラー検出とハンドリング

サービス間コネクション管理のオフロード

サービス検出、暗号化、エラー検出とモニタリング



他のサービスとの連携

サービスの検出

コネクション情報 (通信の秘匿性、暗号化)

異なる言語のサポート

エラーハンドリング

ビジネスロジック - データを取得して Web に表示する

遅延の検出と対応

メトリックの収集

メトリックの送信

セルフヒーリング

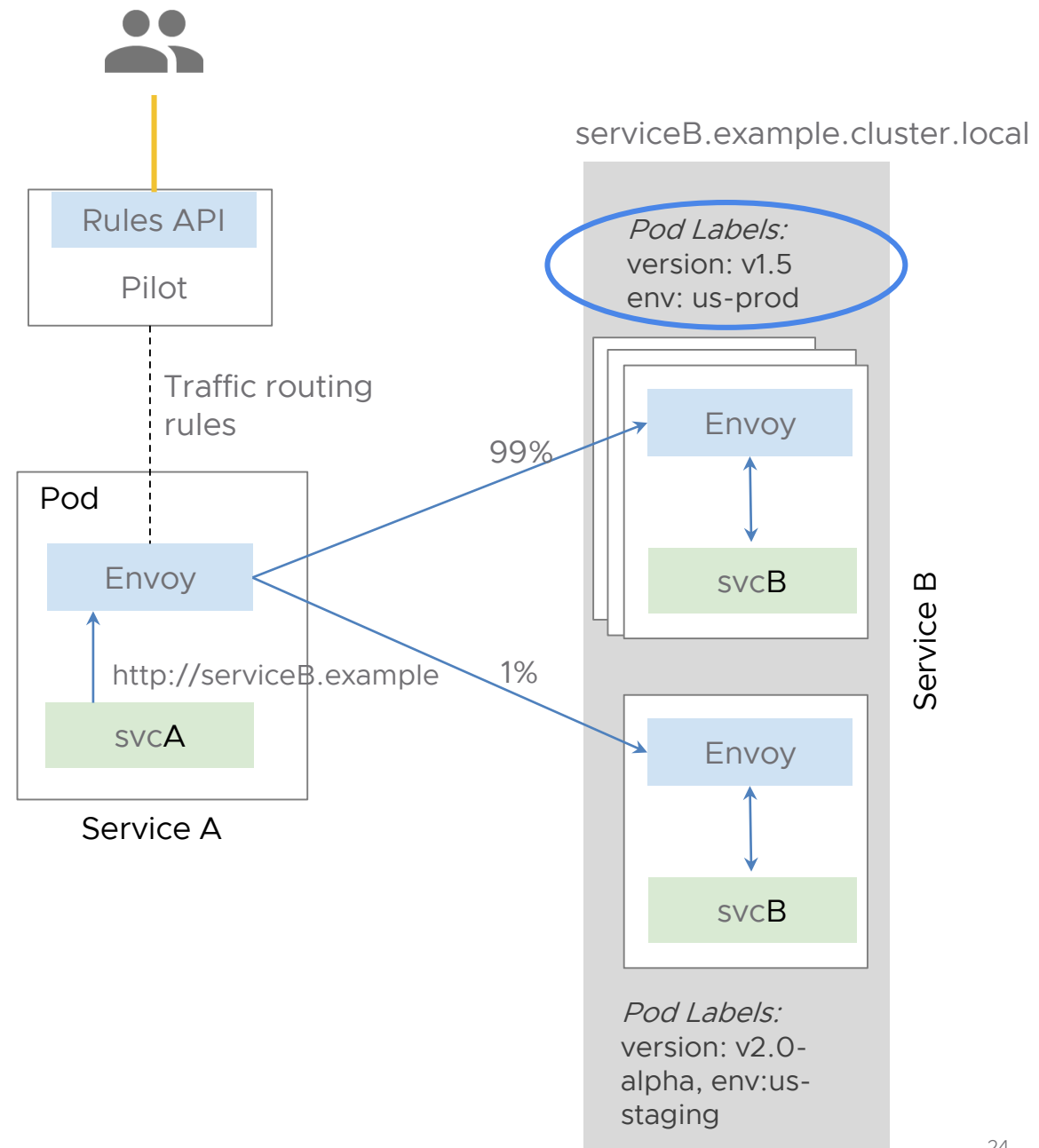
エラー検出とハンドリング

トラフィック スプリッティング

```
// A simple traffic splitting rule

destination: serviceB.example.cluster.local
match:
  source: serviceA.example.cluster.local
route:
- tags:
  version: v1.5
  env: us-prod
  weight: 99
- tags:
  version: v2.0-alpha
  env: us-staging
  weight: 1
```

インフラから切り離された
トラフィック制御



トラフィック ステアリング

```
// Content-based traffic steering rule
```

```
destination: serviceB.example.cluster.local
```

```
match:
```

```
  httpHeaders:
```

```
    user-agent:
```

```
      regex: ^(..*?;)?(iPhone)(;.*)?$
```

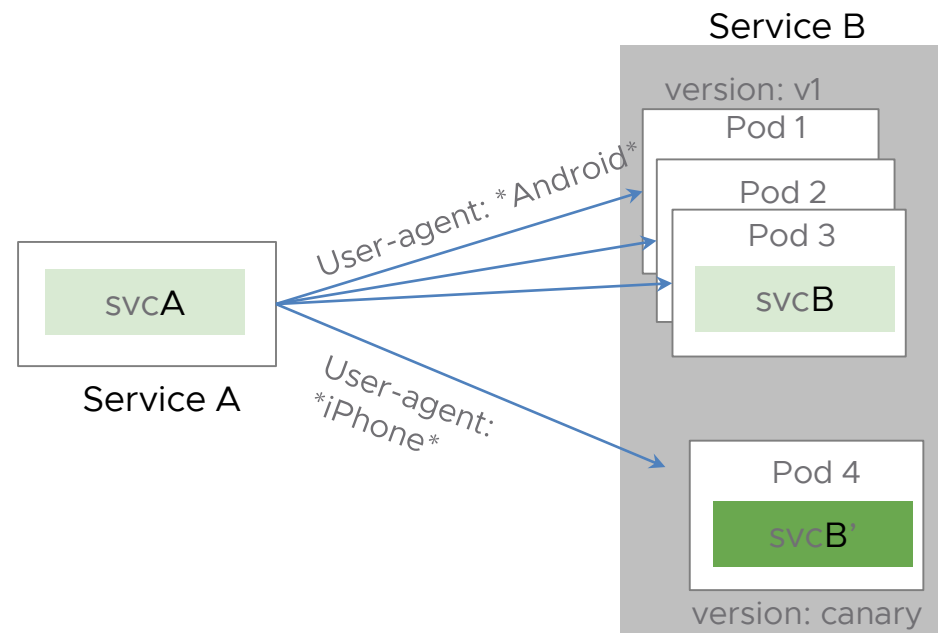
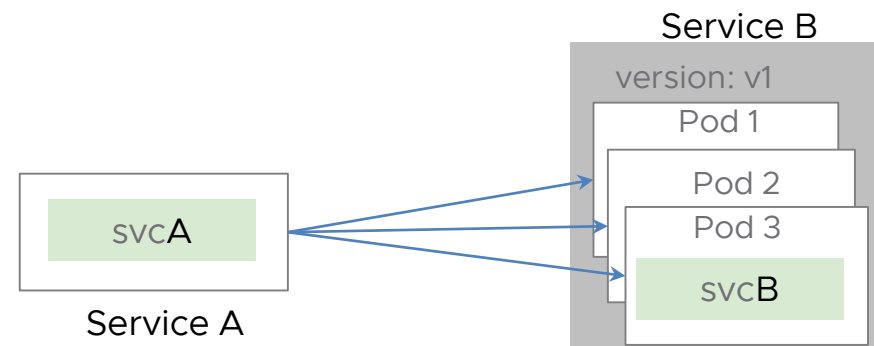
```
precedence: 2
```

```
route:
```

```
- tags:
```

```
  version: canary
```

コンテンツに基づく
トラフィックステアリング



NSX Service Mesh ~ 理想のサービスメッシュに向けて

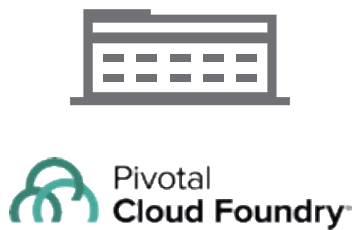
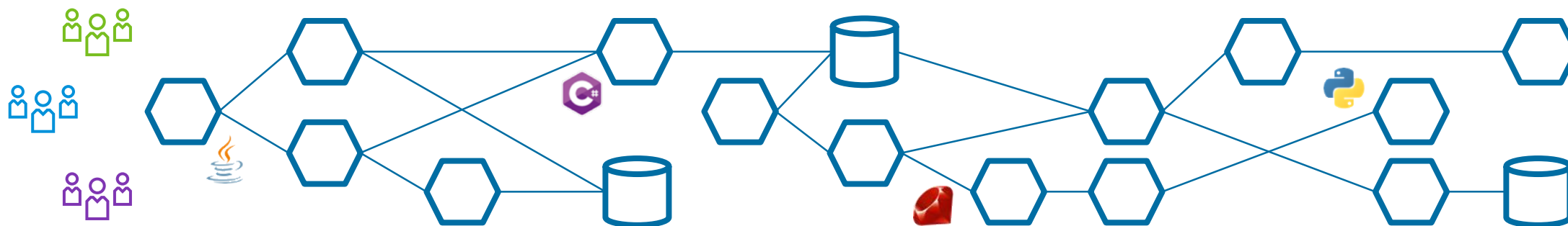
安定したサービス検出、視覚化、制御とセキュリティ

マルチプラットフォーム、
マルチクラウドの
フェデレーション

中央集中型の
可視化と監視、
セキュリティ

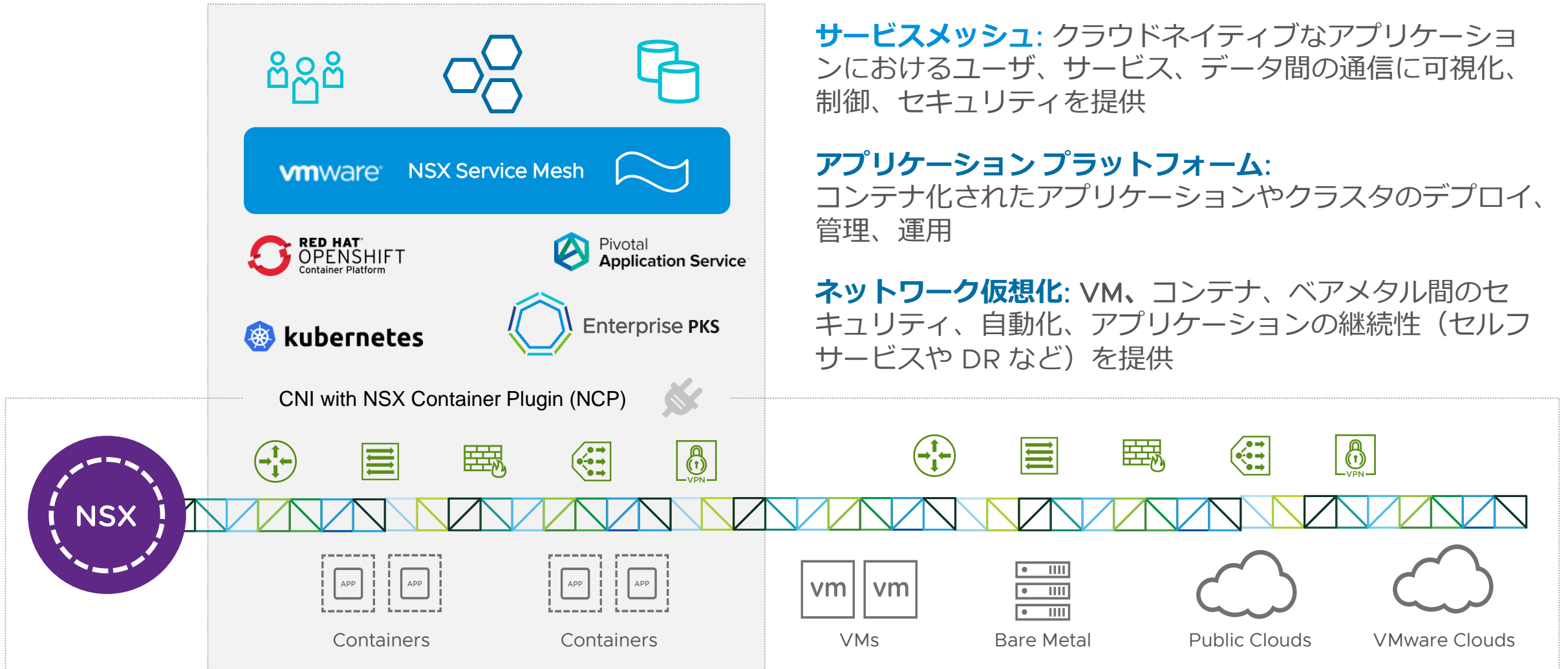
ユーザ、サービス、
データのグローバルな
ポリシー管理

コンテナ以外の
プラットフォーム
への対応



VMware NSX Service Mesh

NSX ポートフォリオにサービスメッシュを拡張

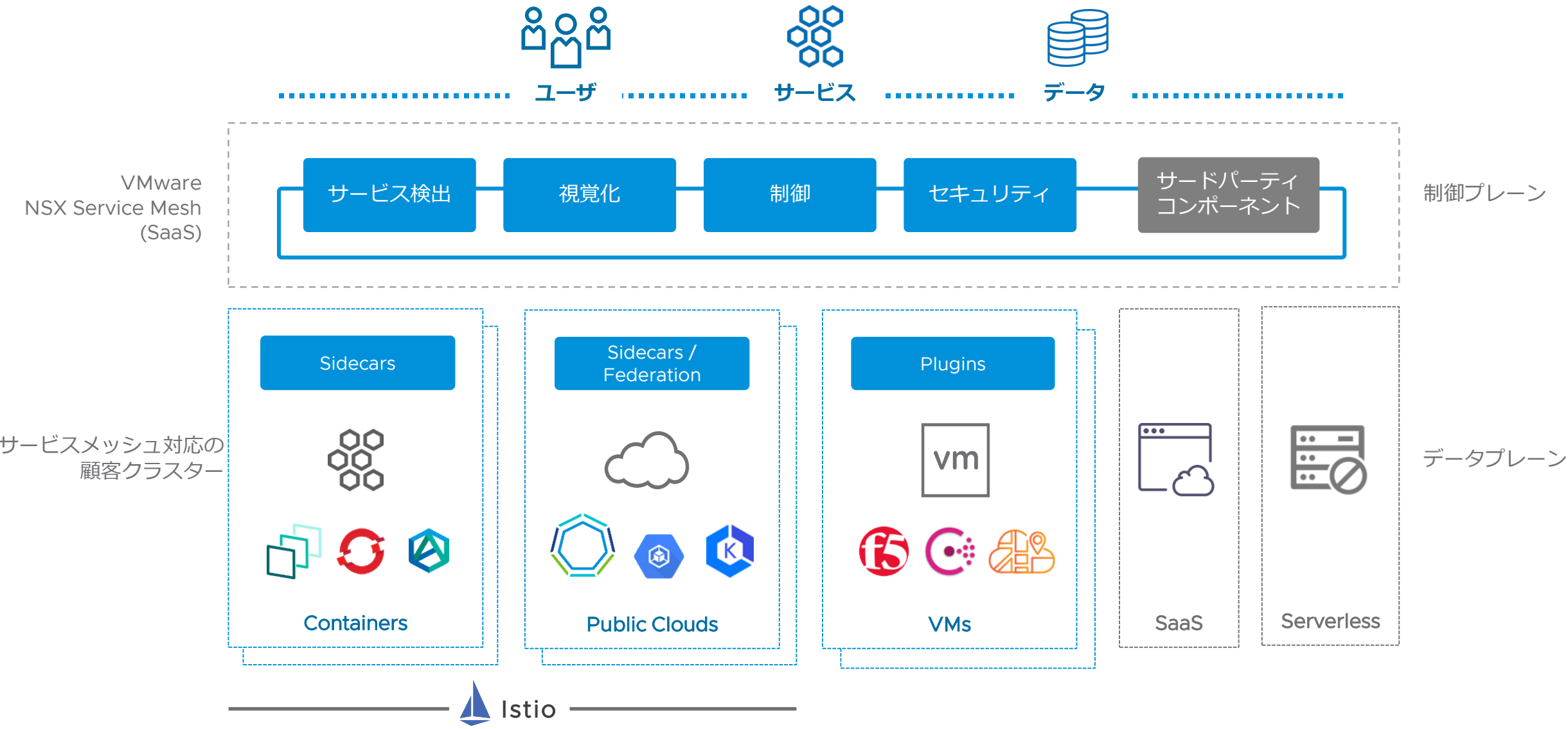


サービスメッシュ: クラウドネイティブなアプリケーションにおけるユーザ、サービス、データ間の通信に可視化、制御、セキュリティを提供

アプリケーションプラットフォーム: コンテナ化されたアプリケーションやクラスタのデプロイ、管理、運用

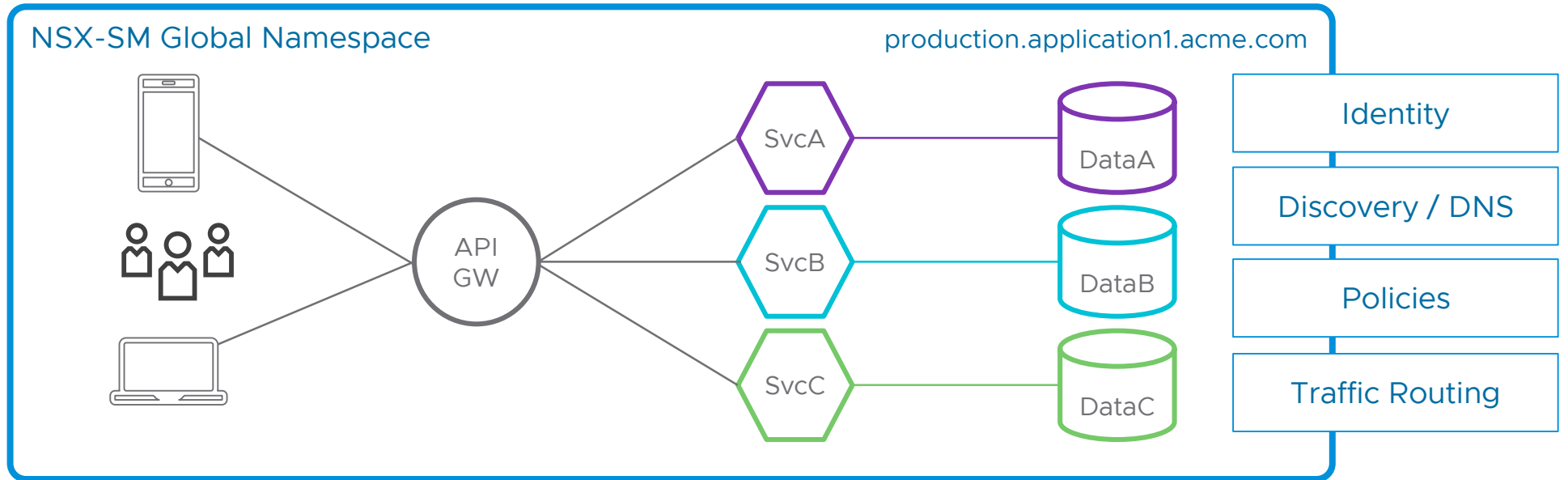
ネットワーク仮想化: VM、コンテナ、ベアメタル間のセキュリティ、自動化、アプリケーションの継続性（セルフサービスや DR など）を提供

VMware NSX Service Mesh のビジョン

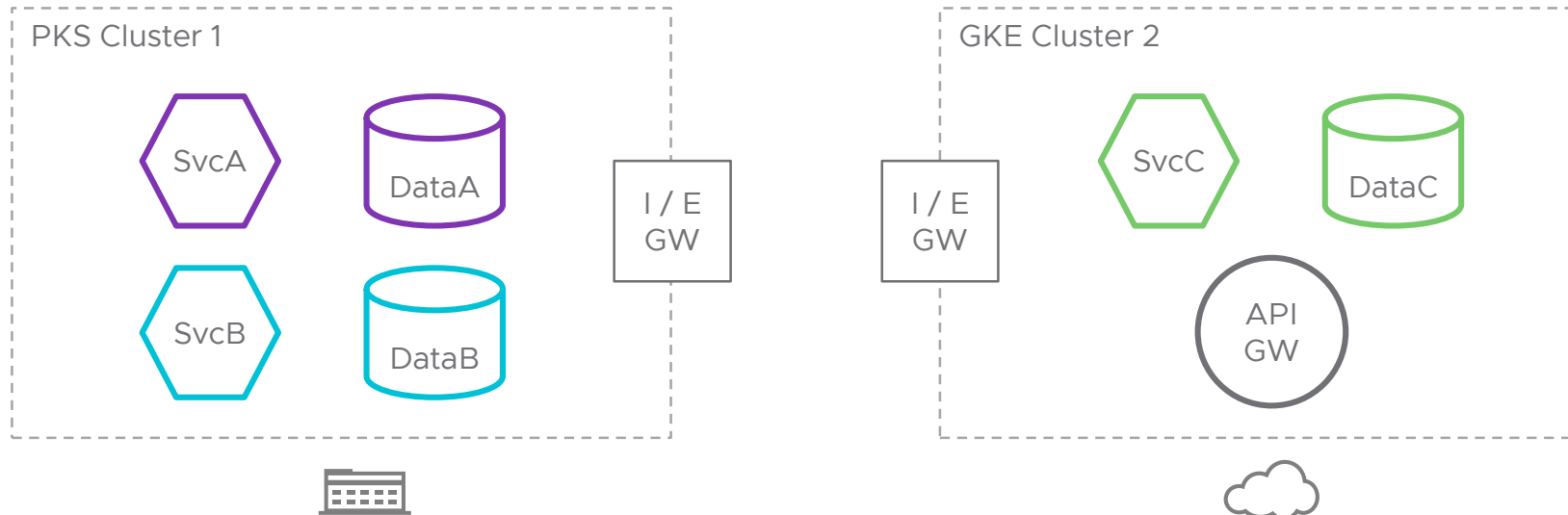


NSX Service Mesh Global Namespaces

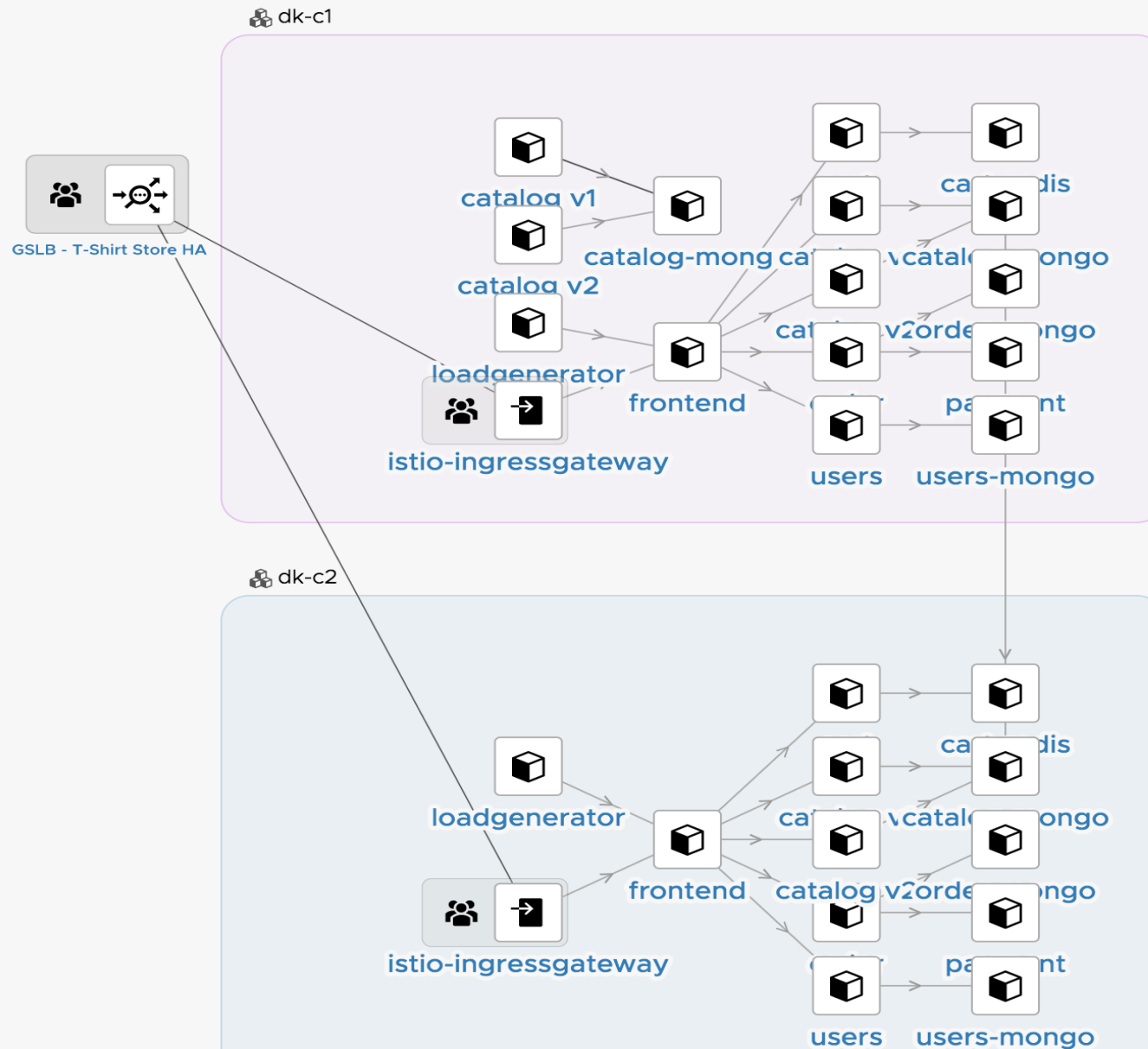
論理ビュー



物理ビュー



NSX Service Mesh Global Namespaces



まとめ

NSX-T Data Center コンテナネットワーキング と NSX Service Mesh

NSX-T Data Center コンテナプラグイン

Kubernetes ネットワークコンストラクトを
NSX-T Data Center 上で構成

- L2 / L3 ネットワーク
- マイクロセグメンテーションのポリシー
- ロードバランサ (Service, Ingress)

Kubernetes の構成変更に応じて自動的に
ネットワークを設定 / オートスケール

K8s on VMware vSphere®, PKS, OpenShift
に対応

NSX Service Mesh

Kubernetes に展開された Istio サービスメッシュを管理

- SaaS ポータルからの展開と視覚化、可観測性

様々な Kubernetes サービスの上で動作可能

- NSX-T Data Center と共存可能
- Global Namespace によるクラウドをまたがったサービスメッシュの管理

2019年11月現在、ベータリリース

ご清聴、ありがとうございました