ベアメタルクラウドのネットワーク設計と課題

楽天株式会社
クラウドプラットフォーム部
江川 友寿

Rakuten

本セッションの概要

新社内クラウド基盤プロジェクトで得られた
プラットフォームとネットワークについての知見を報告します

話す内容 (**23** minutes)

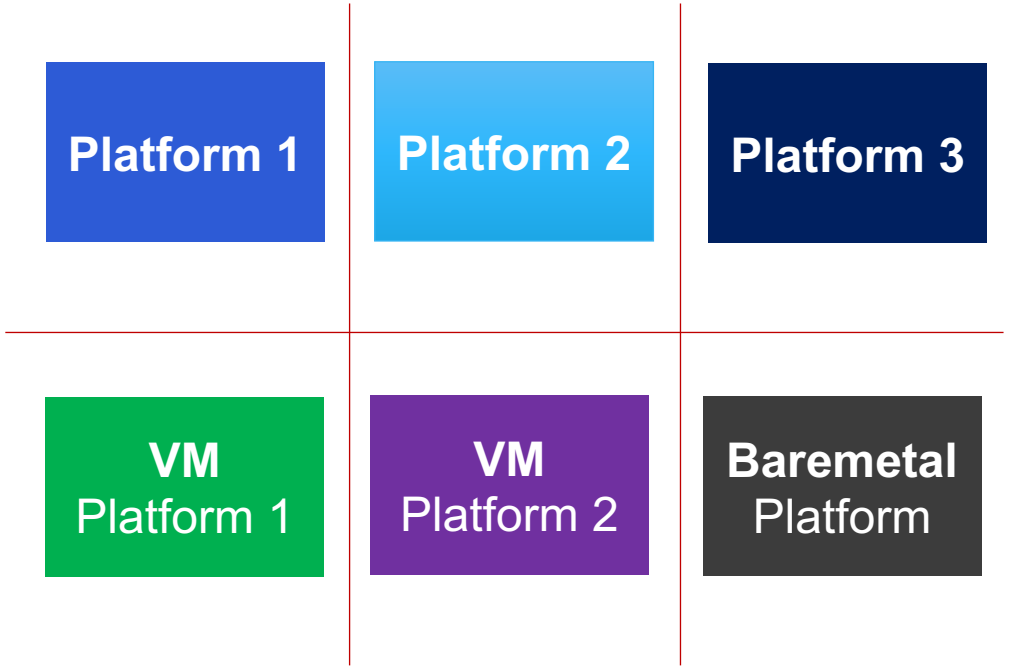
- 弊社インフラ基盤の経緯
- 新社内ベアメタルクラウドの概要
- プラットフォームとネットワーク設計について

質問 & 議論 (**7** minutes)

これまでのインフラについて

2018年: 自由と混沌、そして限界が見え始める

プラットフォーム乱立によるサイロ化



3-5年後先を見据えた場合に懸念あり

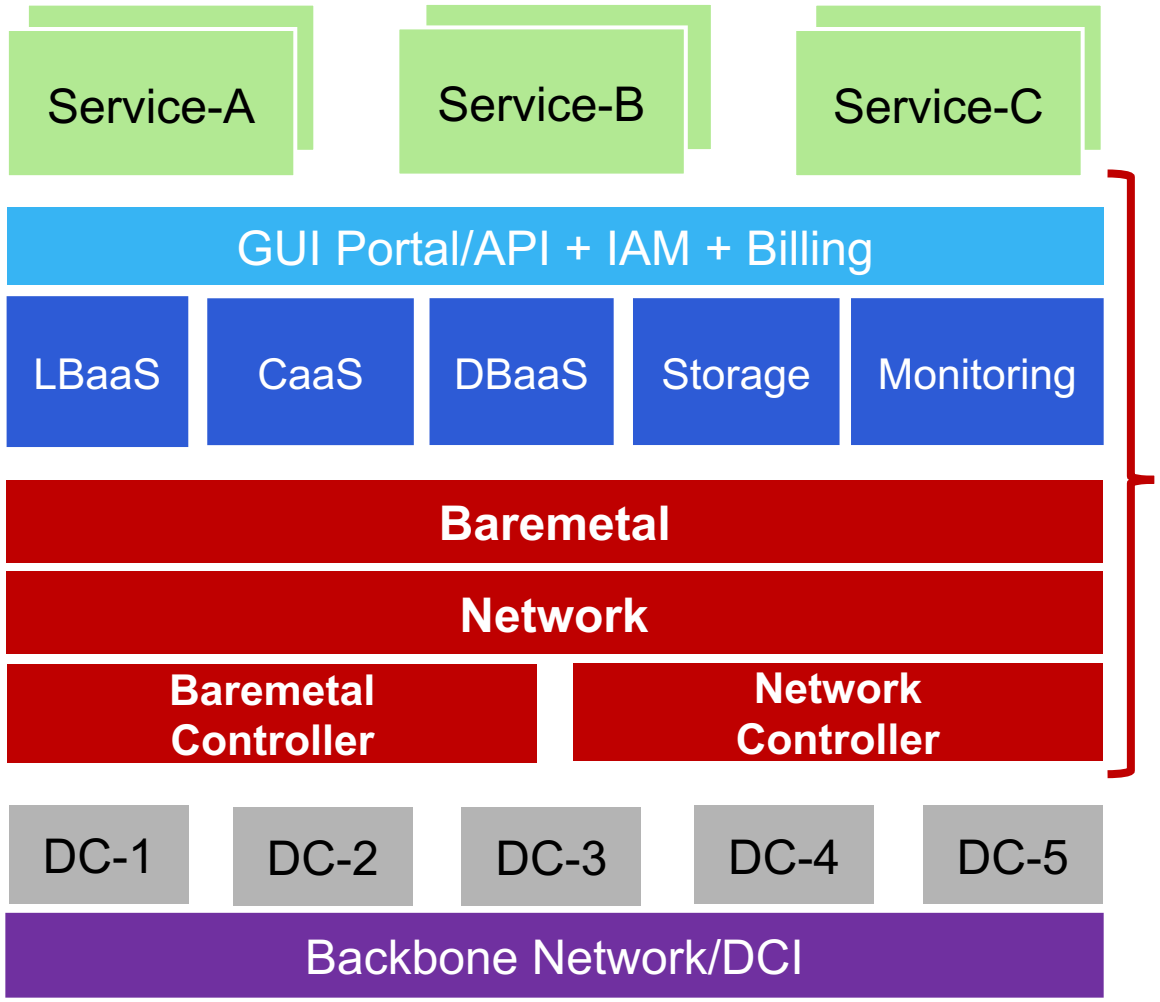
問題点

- 社内エンジニアの開発エクスペリエンスの低下
  - ツールやポータルがプラットフォームごとに複数存在
  - ユーザから見てどれを使うのがベストかわかりにくい
  - 課金や認証がプラットフォームごとに異なる
- プラットフォーマー視点
  - ハイパーバイザのアップグレードがライフサイクルに追いつかない
  - プラットフォームソフトウェア自体の管理や検証コストが重い
  - VM多すぎて自動化してでもOSの管理/セキュリティ更新が大変
- 会社視点
  - 全社レベルでのリソース最適化に課題
  - 統一的なセキュリティコントロール
  - ライフサイクル

2018年後半: インフラ戦略変更+大規模な組織再編

新社内ベアメタルクラウド概要

ベアメタルクラウド 概要 (1)



マネージドサービス化

- 組織構成とマネージドサービスを一致させる
- 開発者はマネージドサービスの組み合わせでサービスを作れる
- 組み合わせてさらに抽象度の高いマネージドサービスを生み出す

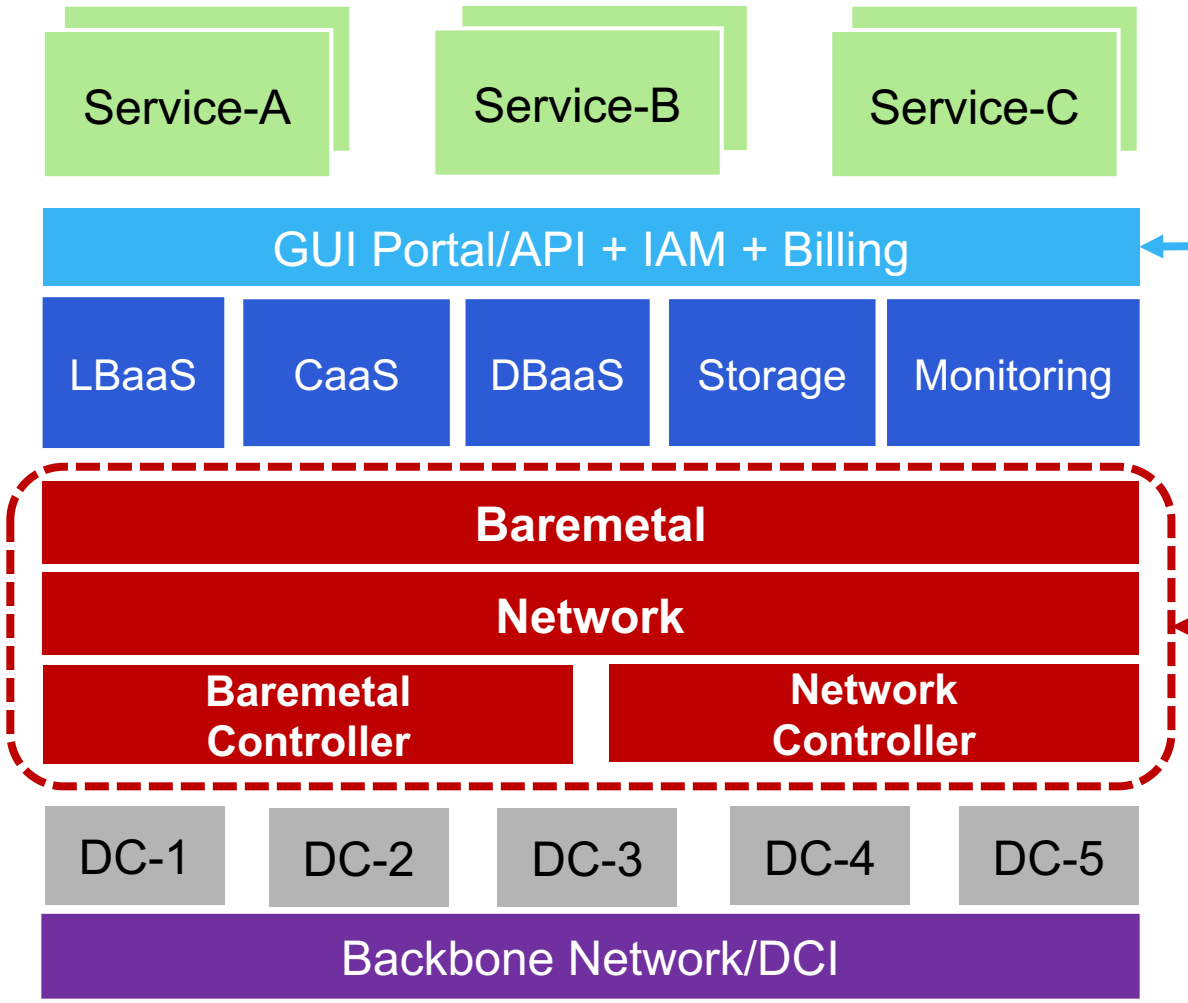
各コンポーネントは疎結合

- 責任分界点の明確化
- 任意のタイミングで upgrade/update が可能
- 時勢に合わせた技術選択が可能

CNCFプロダクト/OSSをメインに切り替え

- デファクトスタンダードなソフトウェアを採用
- 情報が得やすく学習コストを下げられる
- スキルや経験を引き継ぎやすい

ベアメタルクラウド 概要 (2)



認証+認可/ポータルを自社用に開発

- 社内の頻繁な組織再編に対応
- イレギュラーな権限委譲に対応

ベアメタルを基点としたプライベートクラウド

- 全ての社内ユーザにベアメタルサーバを提供可能
- ベアメタルサーバからマネージドサービスを構築

コアインフラ基盤について話します

コアインフラ基盤で目指したもの

コアインフラ基盤で実現したかったこと

1. 持続可能なコアインフラ基盤

- 技術トレンドの変化に強く自社でフルコントロールを持てること
- 常に最新版の維持が可能かつ基盤上のサービスに影響を与えない設計
- マルチテナント対応

2. 安定して+拡張性に優れ+運用がしやすいネットワーク

- 単一障害点をなくして POD 内ネットワーク全断リスクをゼロにすること
- 制限を気にせずに済む充分なスケーラビリティの確保
- 運用が容易 = 次の挑戦に向けた余力が生まれる

3. Network-aaS

- ユーザニーズに応えたネットワーク機能の提供
- ネットワークサービスをプロダクトと位置づけてAPIベースで機能を提供

実現目標 1

持続可能なコアインフラ基盤

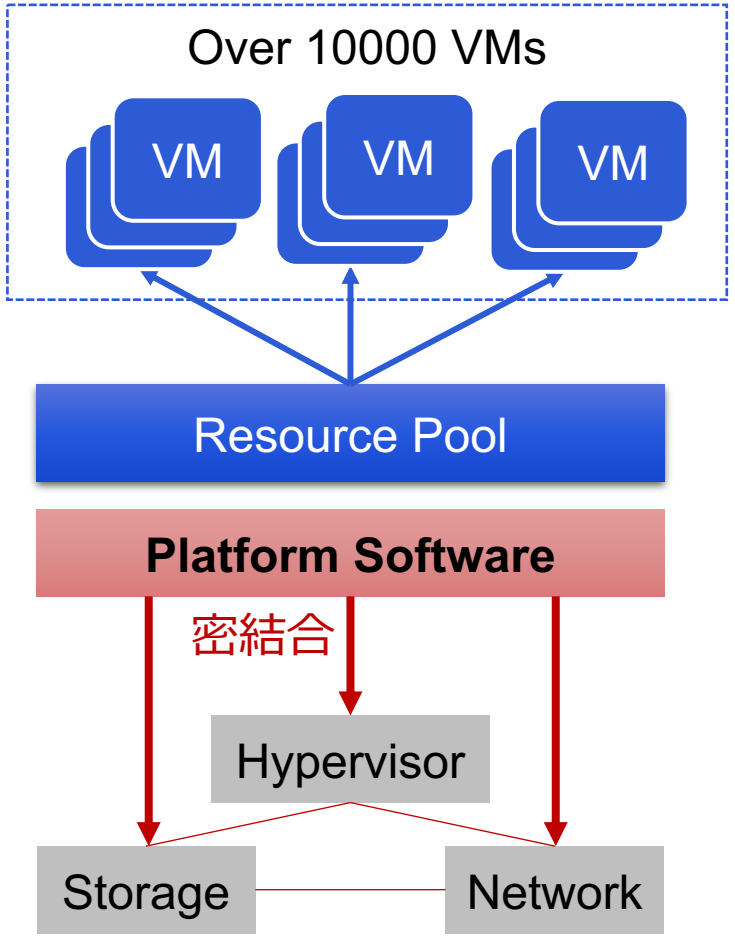
比較: 仮想化インフラ基盤

メリット

- プラットフォームソフトウェアによる統合管理
- リソースの抽象化
  - ユーザは 物理サーバを意識しなくてよい
  - リソースのプール化による使用率最適化

デメリット

- 基盤の運用負荷が高い
  - ライフサイクルやアップグレードに追随するのが大変
  - 変更による全サービスへの影響
- 大量の VM が存在することによる OPEX の増大
  - VM の性質上 構成ドリフトが起きやすい
  - OS レベルの脆弱性対応コスト



VM 数の増加を抑えたい → コンテナ数の比率を上げていく

比較: ベアメタルクラウドのコアインフラ基盤

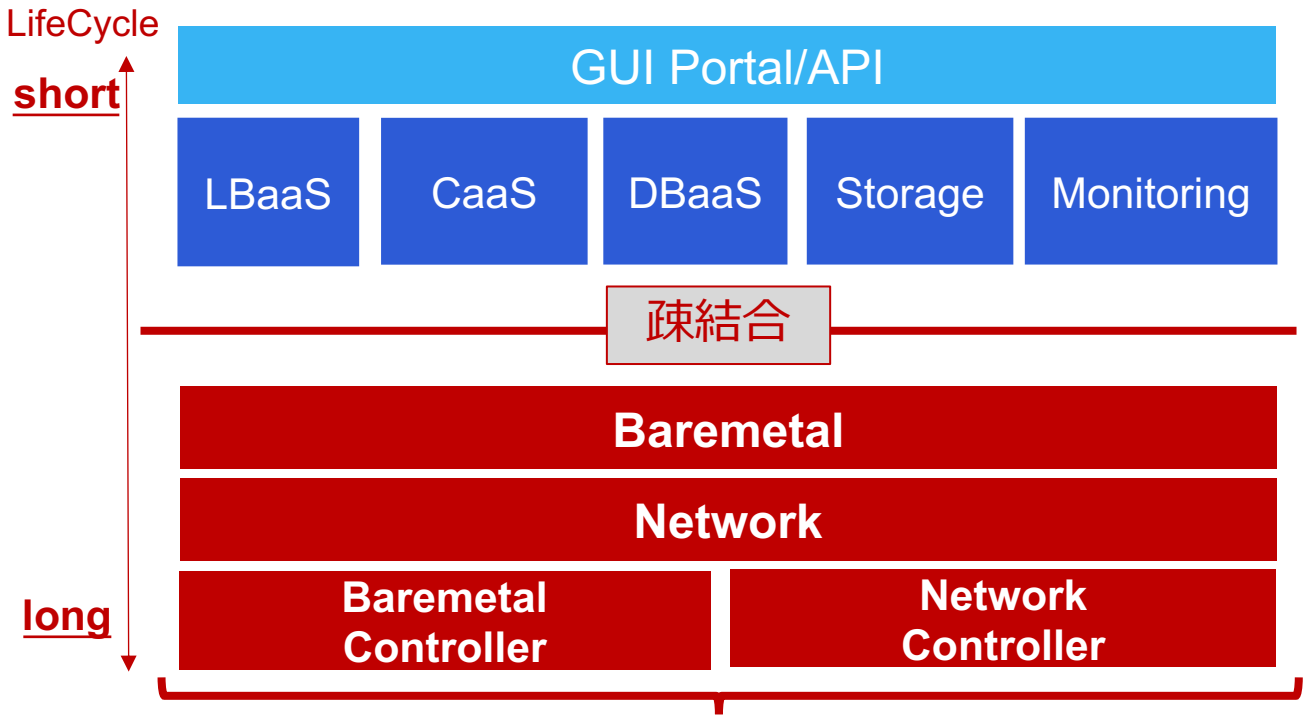
特徴

- 安定かつ高性能な compute リソースの提供
  - リソース競合問題が発生しない
- 技術トレンドの変化に強くシンプル
  - 上位マネージドサービスへ機能をオフロード
- 常に最新の状態に維持可能
  - Controller は永続的なソフトウェア資産

トレードオフ

ベアメタルユーザに高いインフラスキルを要求

- ベアメタル障害時の冗長性担保
- リソース抽象化
- 使用率の最適化



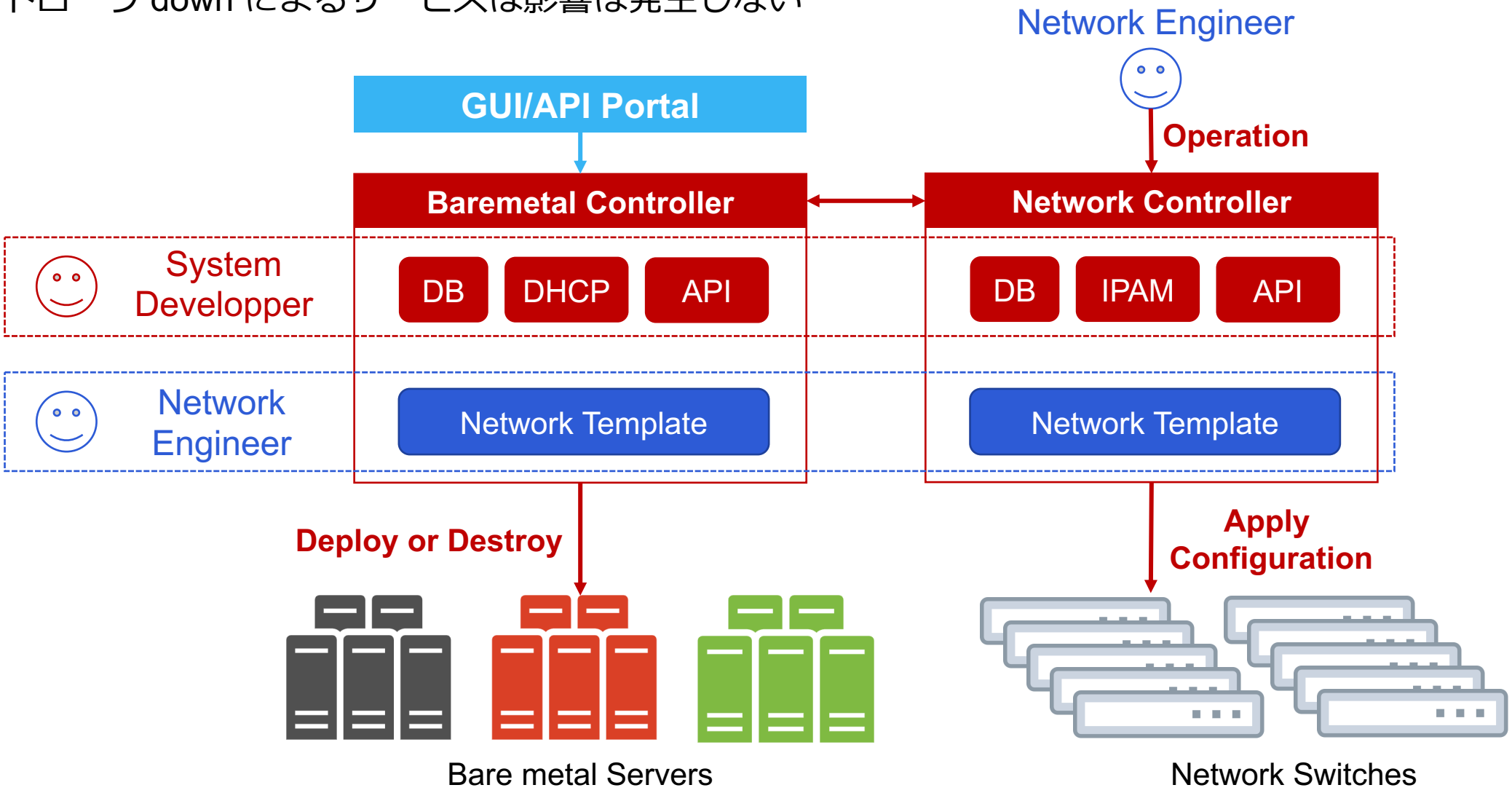
持続可能なコアインフラ基盤

- プロダクト寿命は自社でコントロール
- 継続的に改善を適用
- リスクのある upgrade は実施不要

上位マネージドサービスの組み合わせでサービスを開発をする

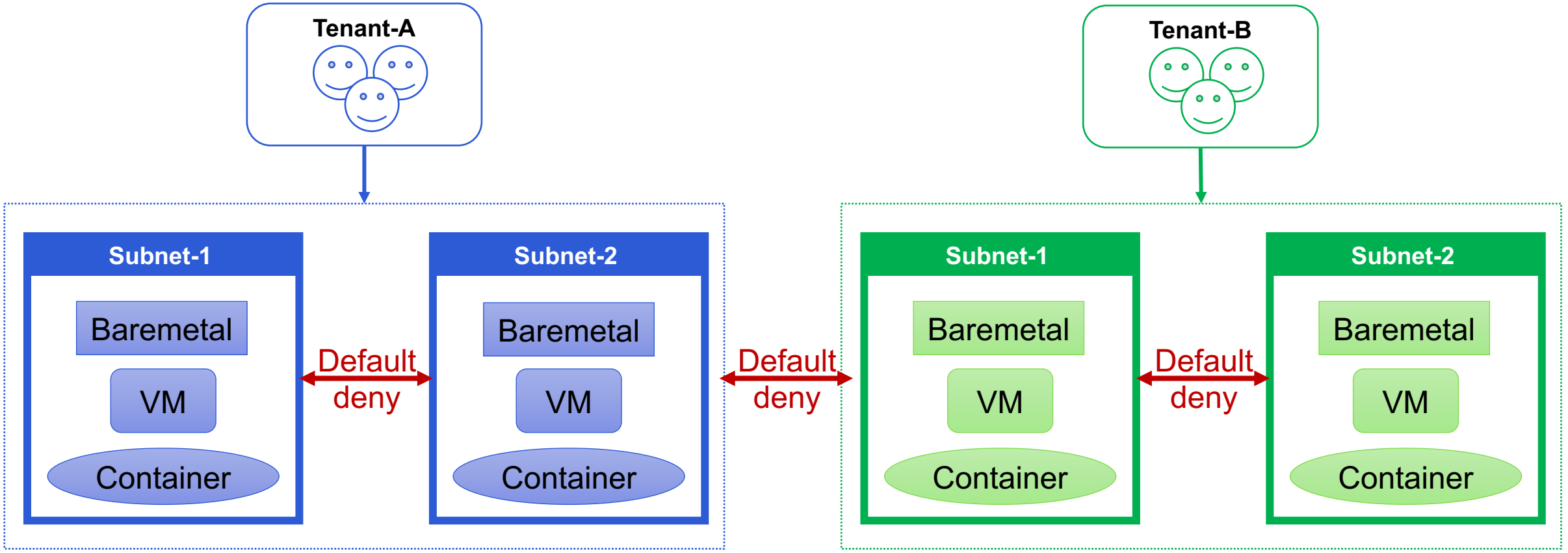
Baremetal / Network Controller

- 各データセンターや POD ごとに独立
- コントローラ down によるサービスは影響は発生しない



マルチテナントネットワークデザイン (理想) 非採用

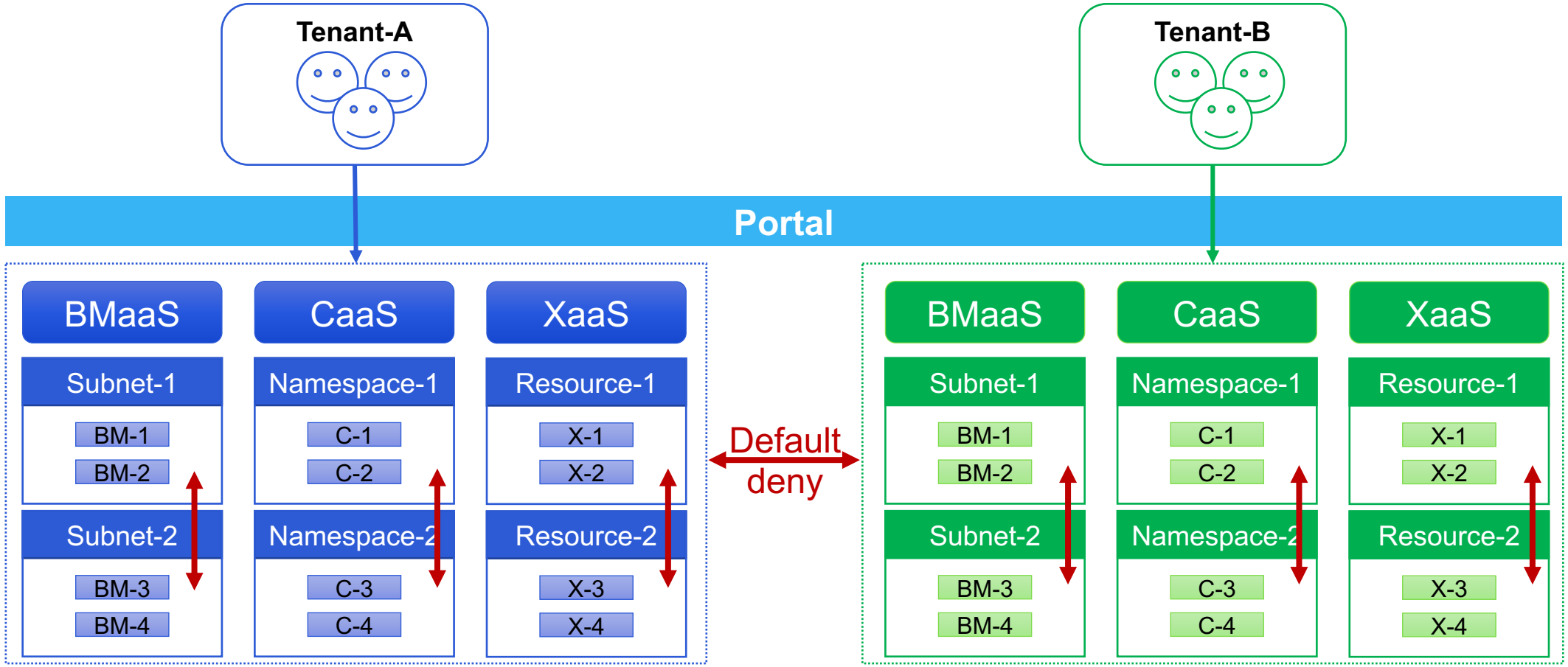
- 異なるワークロードを同じサブネットに配置できるようにすると管理しやすいはず…?
  - ベアメタル、VM、コンテナを同一サブネットにして論理ネットワークをシンプルに??
- コアインフラ基盤とプラットフォームソフトウェアを密結合にする必要があったので破棄



マルチテナントネットワークデザイン (現在) 採用

マネージドサービス側の各仕組みでテナントリソースを分離して提供

- ポータル上で一元管理できるので異なるワークロードを扱いやすくする
- 例) コアインフラ基盤はベアメタルサーバ群をサブネットで分離



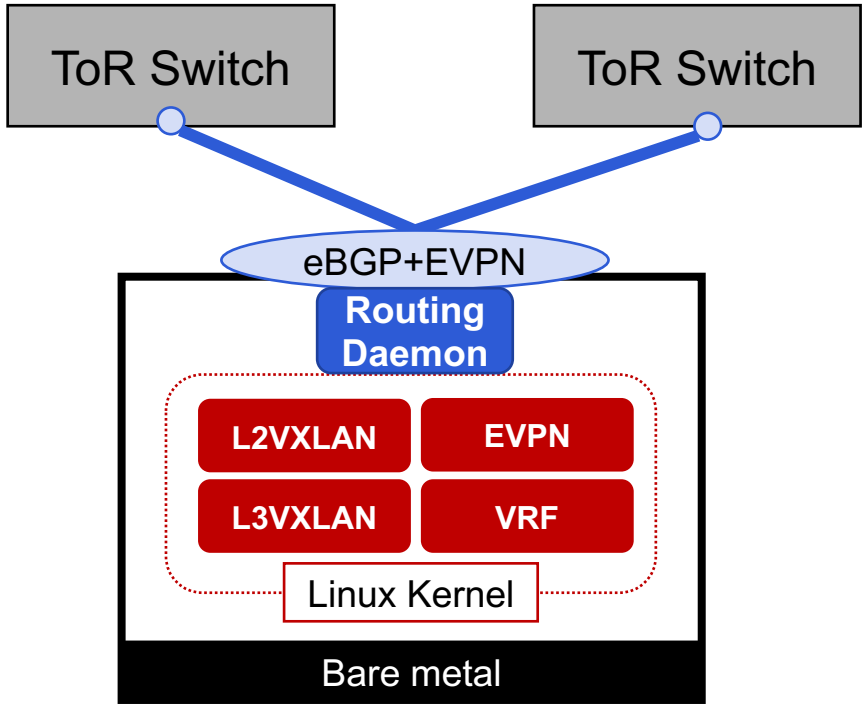
実現目標 2

安定して+拡張性に優れ+運用がしやすいネットワーク

L3 ベアメタルサーバの検討

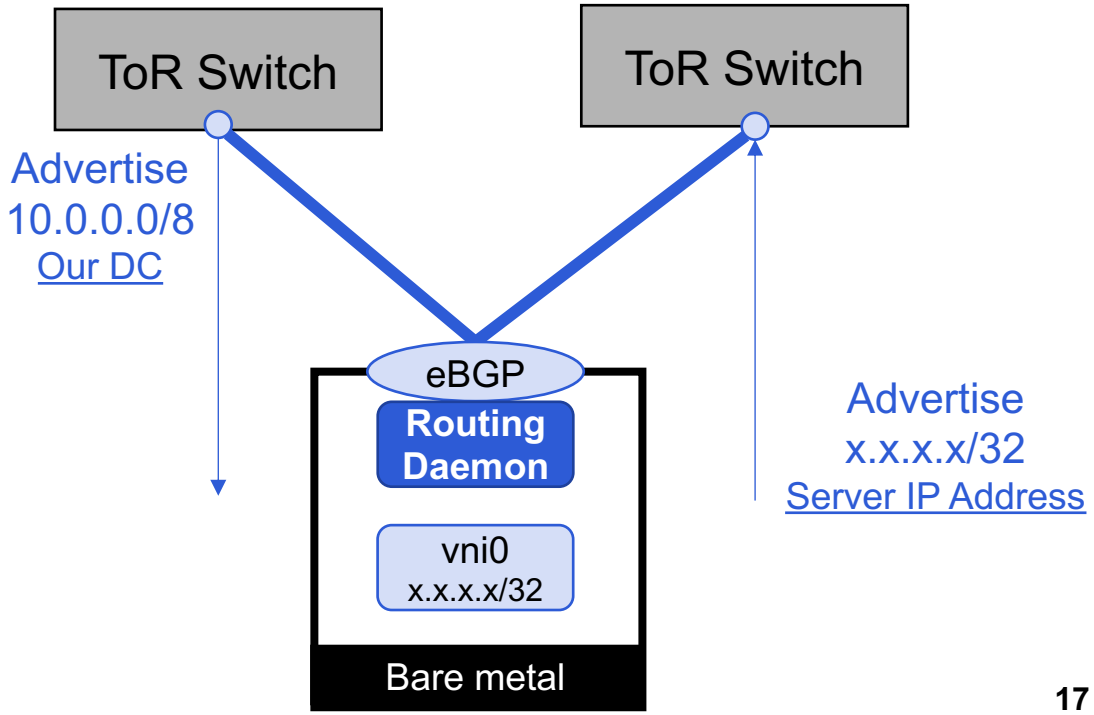
Approach 1: **VXLAN/EVPN on the Host** 非採用

- (Good) MLAG ナシでラック間 L2 ネットワークを展開可能
- (Bad) 運用負荷が高い
  - ホスト上の複雑なネットワーク設定 → 安定性に不安
  - トラブルシューティングが難しい → 運用が大変
  - Linux Kernel に機能を多く依存 → シンプルじゃない



Approach 2: **Routing on the Host** 採用

- スイッチは BGP のみサポートしていればよい
  - スイッチの bug を踏む確率が少なくなり安定する
  - 学習コストが低くトラブルシュートが容易
  - 乗り換え先 NOS の選択肢が増える → スイッチHWの選択肢も増える
- ECMP によるシンプルな冗長
  - MLAG からの解放

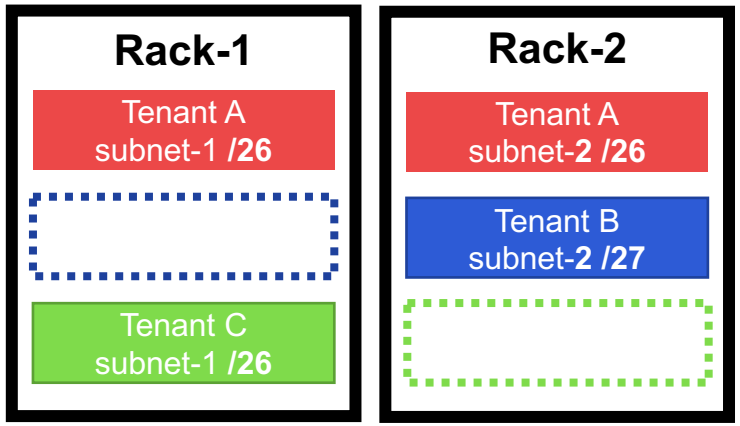


L3 ベアメタルサーバの使用例

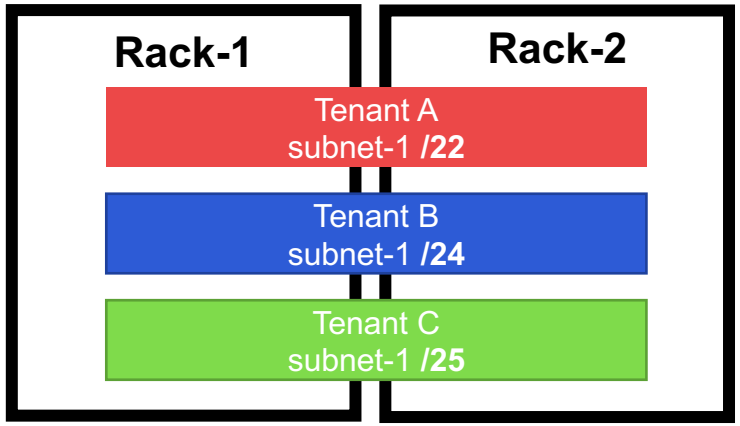
Rack wide Mobility

- ラックを横断してサブネットを展開可能
  - ラック位置を問わずサーバを選択してよい
  - サブネット構成のシンプル化

No VXLAN
on IP-CLOS

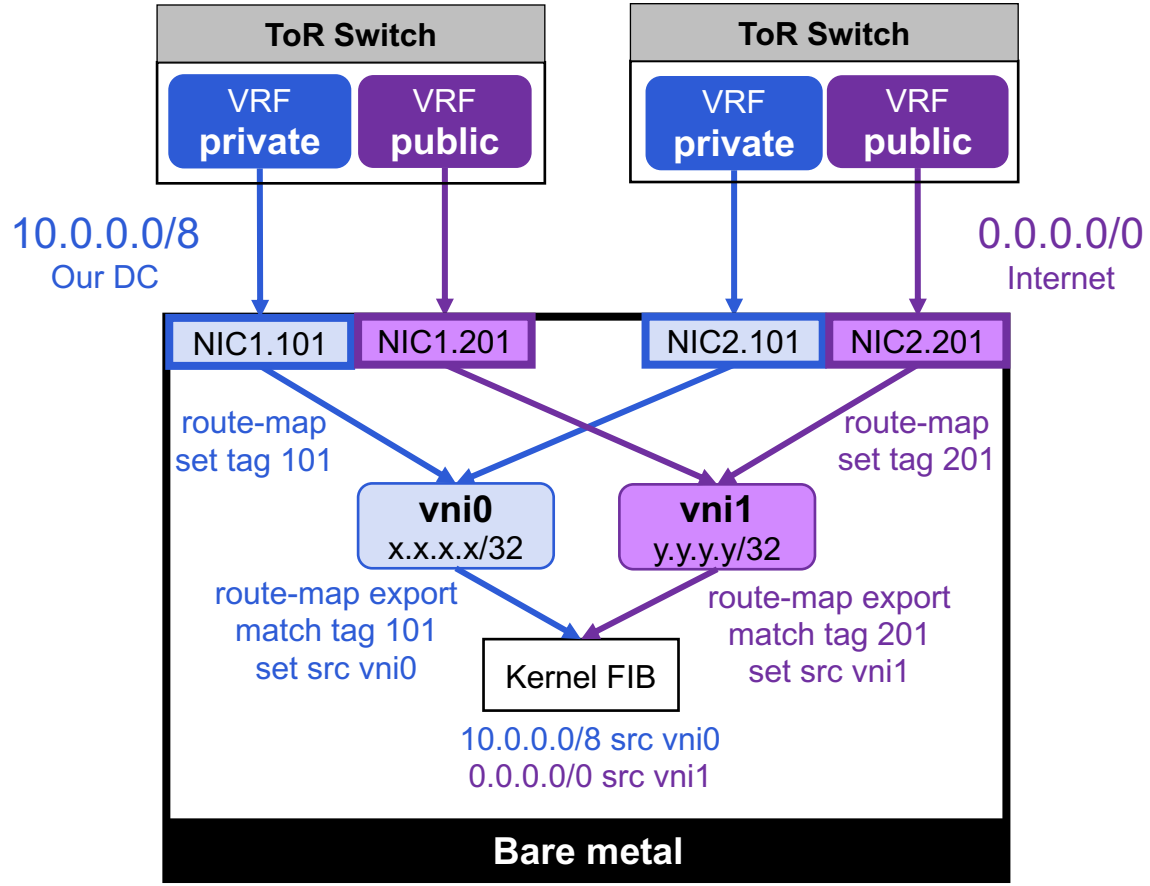


Routing-on-Host
on IP-CLOS



Multi Network

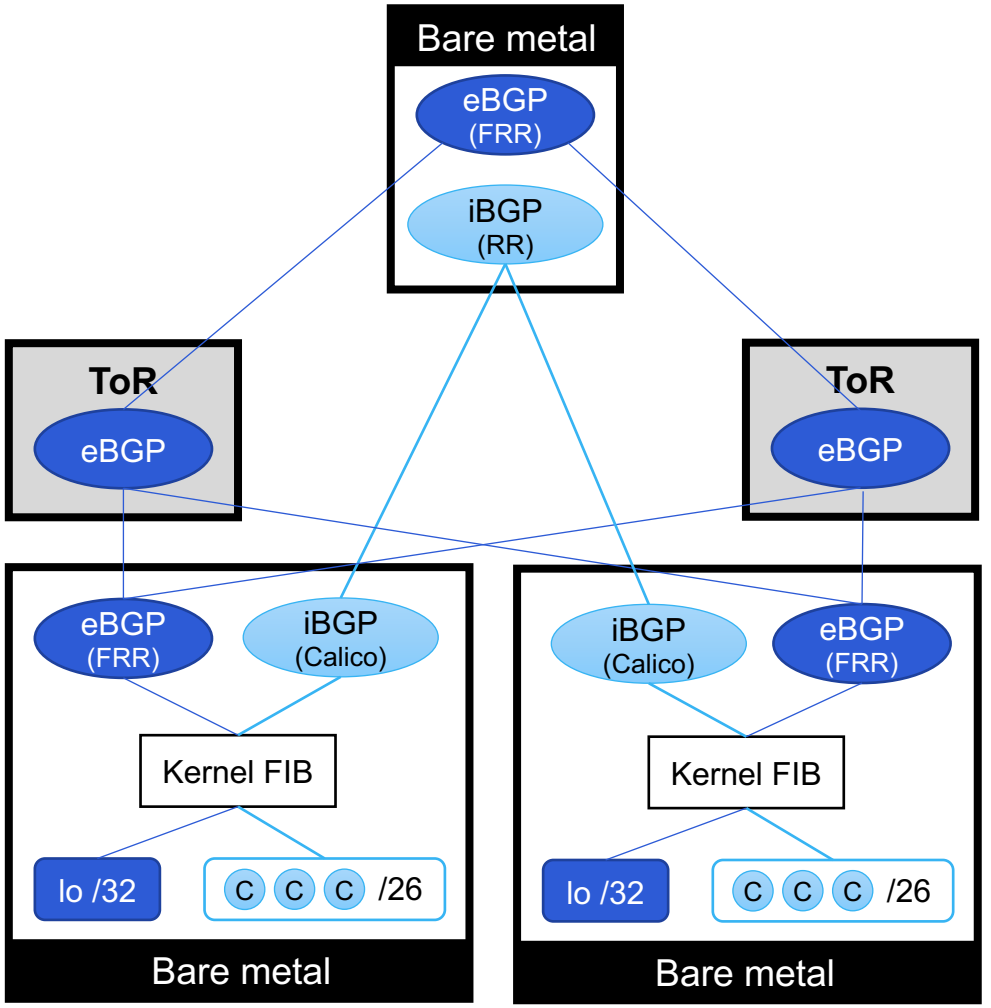
- 用途に合わせた VRF をベアメタルに直接提供
- 例) LBaaS ノードに Global IPアドレスを付与
- FRR に設定追加済みで提供 (route-map etc..)



Kubernetes Networking on L3ベアメタル

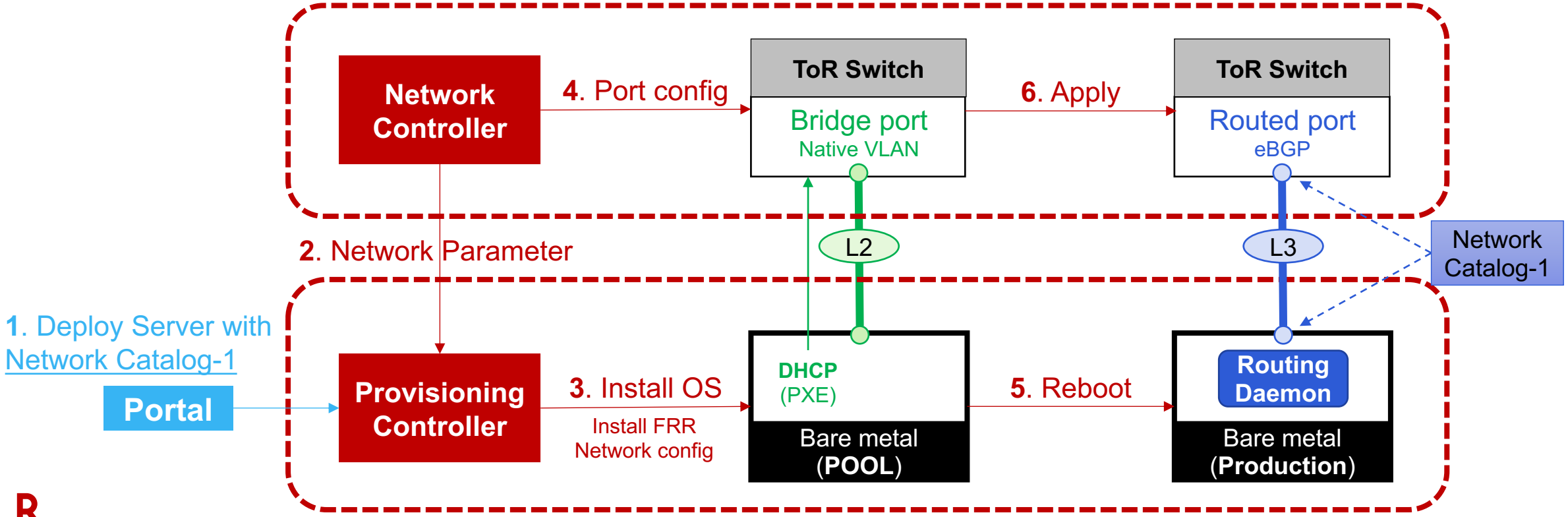
CaaS 用にベアメタルクラウドに最適なネットワーク設計を考える

- 責任分界点を作る
  - FRR の問題はコアインフラ基盤が対応
  - Calico (BIRD) の問題は CaaS が対応
- コンテナ増加による経路数爆発の懸念がない
  - コンテナ経路はベアメタルネットワークに広報されない
  - Container-aware Load balancing by LBaaS が検証中
- 注意点
  - Container OS用に FRRコンテナを作成 (systemd-nspawn)
  - FRR の port 番号を変更 (BIRD:179, FRR:20179)
  - CaaS ベアメタル上で ECMP を OFF
    - ベアメタルからの送信のみ Active-Standby
    - FRR:IPv6 Unnumbered 上で BIRD:iBGPでコンテナ経路を受信すると BIRDが不正な経路とみなしてしまうため



L3 ベアメタルサーバの構築

- セルフサービスなベアメタルサーバの提供
  - Linux ディストリビューション, サブネット, ネットワークカタログを選択
- ToR スイッチのポート設定が常に動的に変更
  - サーバ を構築: スイッチポートを L2 → L3 に変更 (サブインターフェース)
  - サーバ を POOL に戻す: スイッチポートを L3 → L2 に戻す



L3 ベアメタルサーバの注意点

- 経路ハイジャック対策
  - ユーザがベアメタルの root 権限もってるので FRR の設定を変更できてしまう
  - 不正な経路をサーバから流されないようにスイッチ側で経路フィルタリング (ベアメタルのホストルートのみ許可)

- ベアメタルユーザへの Knowledge 共有
  - 例) ユーザ側で IPv6 を disable にされると BGP 接続が切れてしまう
  - 例) Kubernetes を kubeadm で展開しようとすると小技で回避が必要など…

- Linux distribution ごとに調整が必要
  - 新しい distribution が出るごとにコアインフラ基盤側で微調整

- L2 接続要件に対応していない
  - L2 前提のミドルウェアが動かない (Redis+Sentinel, keepalived etc..)
    - コアインフラ基盤から代替のネットワーク機能を提供予定 (Anycast IP)
  - L2 接続が必須な場合はサーバのネットワークカタログを変更してもらう
    - 従来の IP-CLOS ネットワークで L2接続が使用可能だがサブネット管理が複雑化

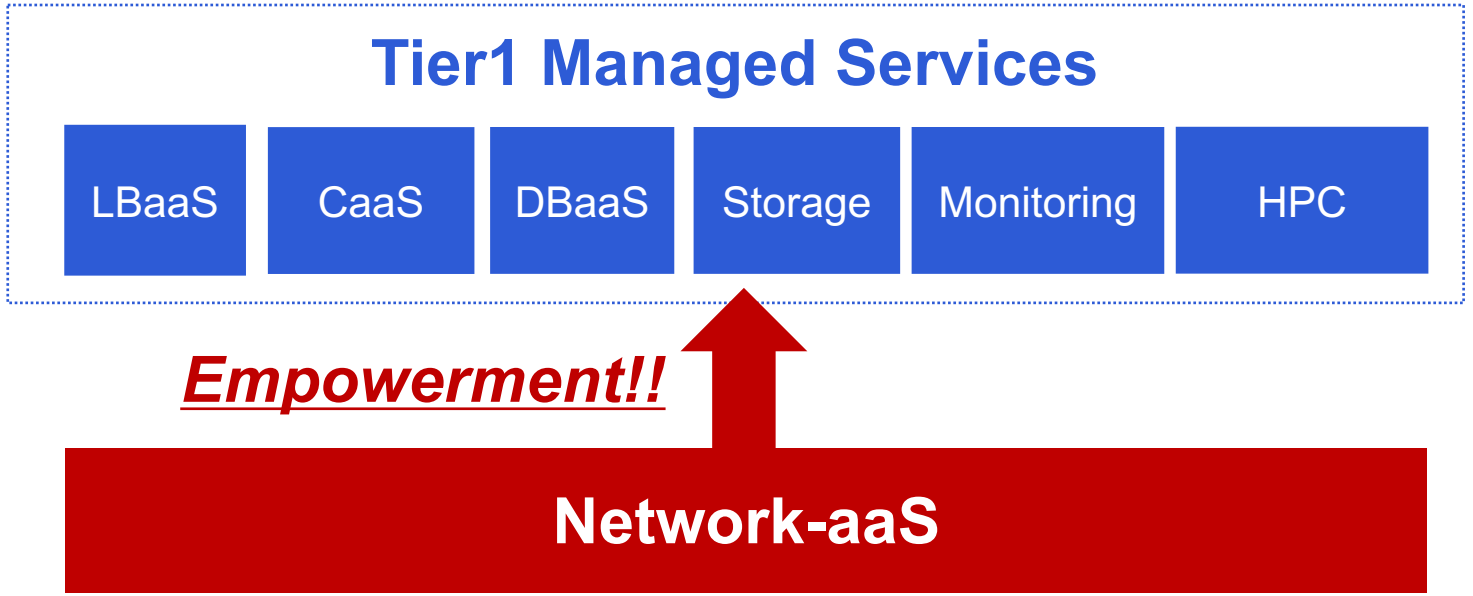
実現目標 3

Network as a Service

Network as a Service

ネットワークサービスをセキュリティを考慮済みの機能としてユーザに提供

- **Distributed Firewall**
  - Multi tenant network 分離をどう実現するか?
- **Internet Gateway**
  - Proxy を経由せずに安全なインターネット接続を提供

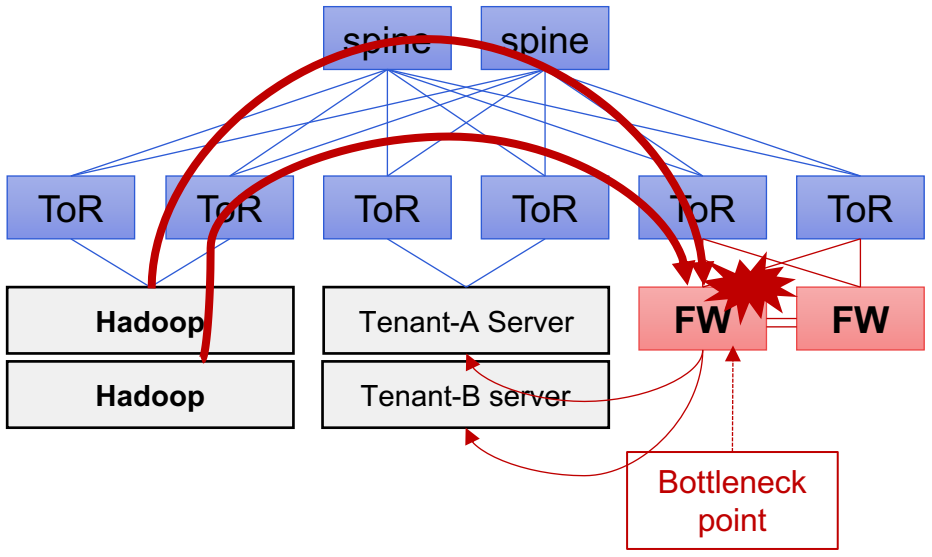


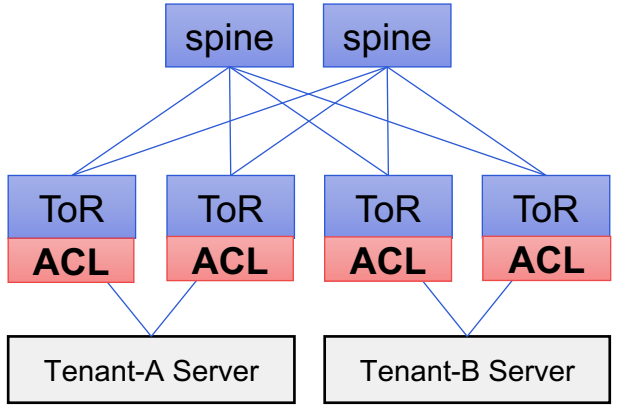
マルチテナントサブネットの分離

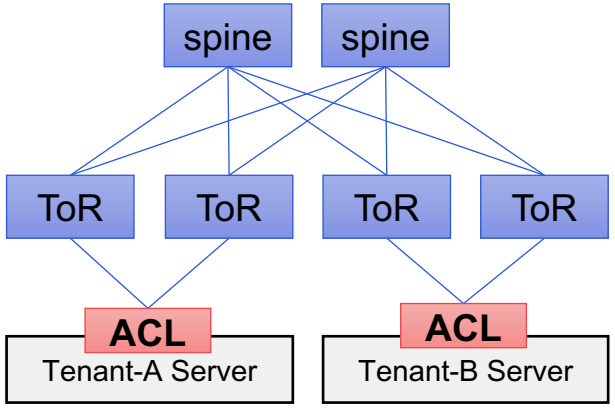
要件: サブネット間を隔離する、必要に応じて柔軟に通信を許可させたい

- OpenStack なら SecurityGroup があるがベアメタルクラウドにそのような機能は存在しない
- 5-tuple を用いて Host/Subnet 両単位で通信制御を実施したい

Approach	1.Firewall Appliance	2.Switch based ACL	3.Distributed Firewall 採用
追加設置コスト	高価なシャーシが必要	なし	独自実装が必要
ルール数の上限	ほぼ問題ない	上限あり (ASIC処理のため)	問題ない (ホストのメモリ依存)
テナントのルール管理	やり方による	難しい	可能





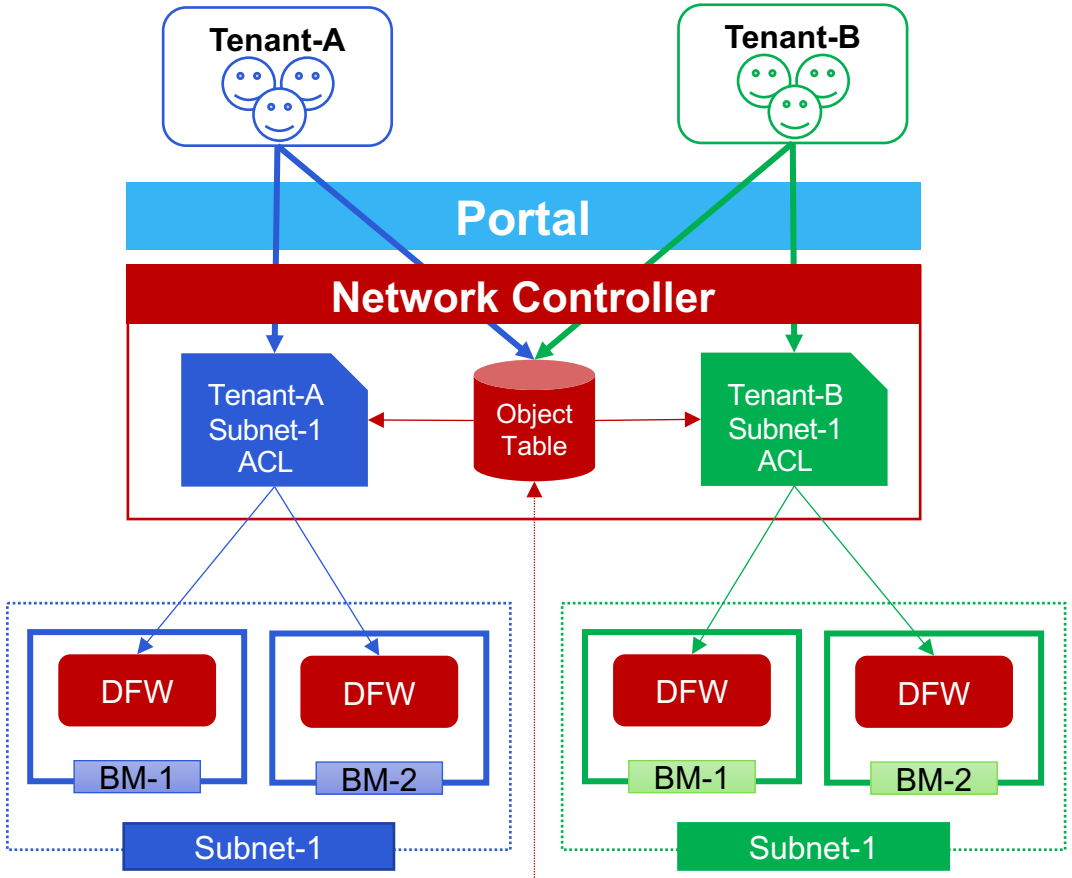


Distributed Firewall

Pre-release

自サブネット内のベアメタルを外部から保護するための機能

- ユーザのベアメタルに Firewall コンテナを配布
  - K8s の sidecar と似たデザイン (systemd-nspawn)
  - Default-Deny を適用
  - 自サブネットの ACL 情報を問い合わせてルールを適用
- ユーザ支援機能: Object Table
  - Source/Destination の IPAddress を Object として定義
  - Firewalll ルール設定時に Object を選択
- 課題: コアインフラ基盤としてルールを強制できない
  - ユーザは root権限を持ってるので DFW をパージ可能
  - 動作状況確認のため Health-Check 機能の追加



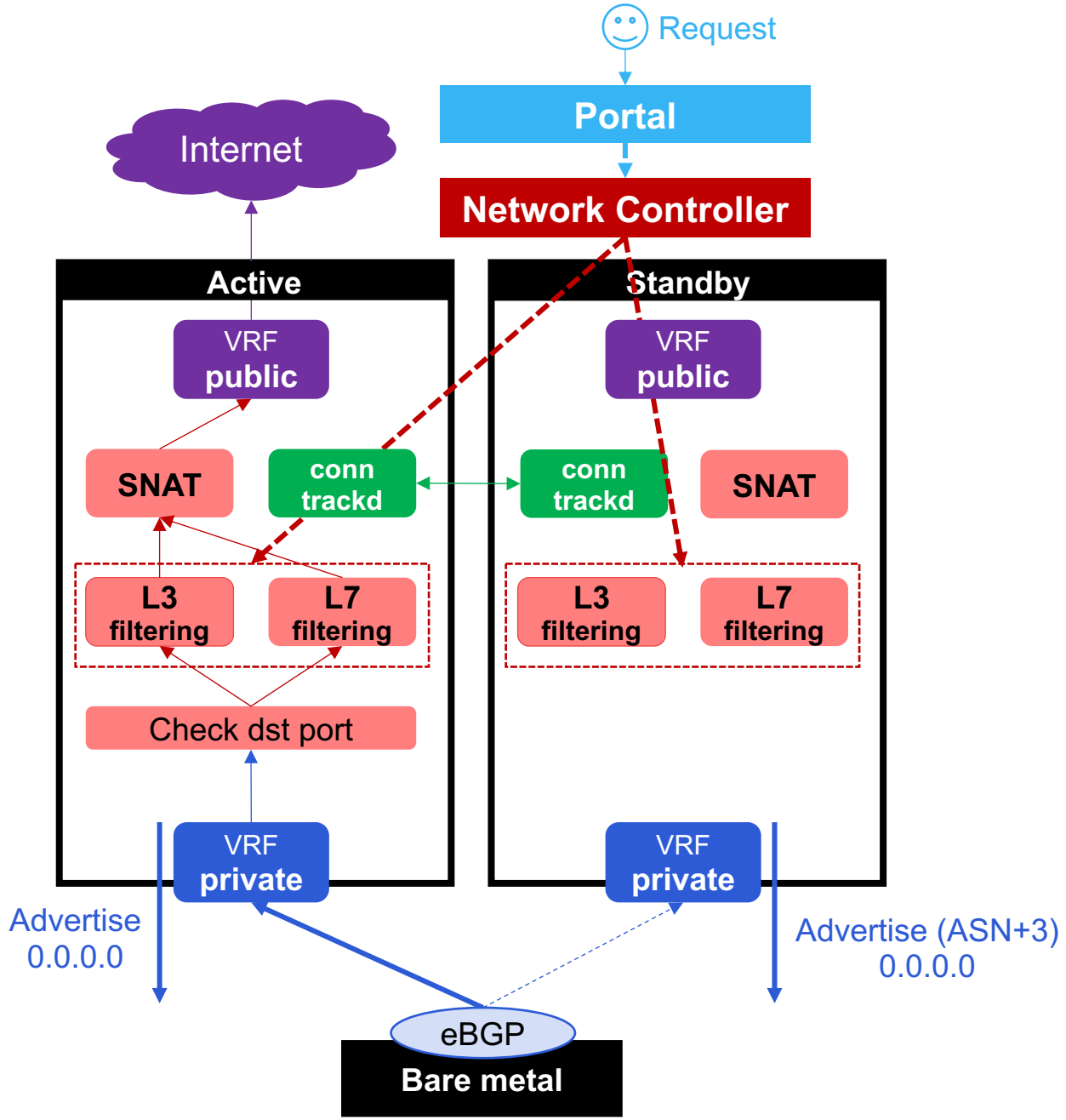
Object	IPAddress	Owner
Service-A	x.x.x.x./24	Tenant-A
Service-B	- y.y.y.y/24 - z.z.z.z/25	Tenant-B

Internet Gateway

PoC

安全なインターネット接続の提供

- 全てのベアメタル に default-route を広報
- Active - Standby 構成
  - ベアメタル 2台構成
  - Session table の同期 (conntrackd)
  - 調停スクリプトによる制御 (ASN path-prepend)
- Filtering
  - L7: URI ベースでフィルタ (Squid)
  - L3: HTTP/HTTPS 以外のプロトコル (nftables)
- 課題
  - Active - Active 化の検討
  - 各種パラメータの調整



現状と今後の取り組み

現在のステータス

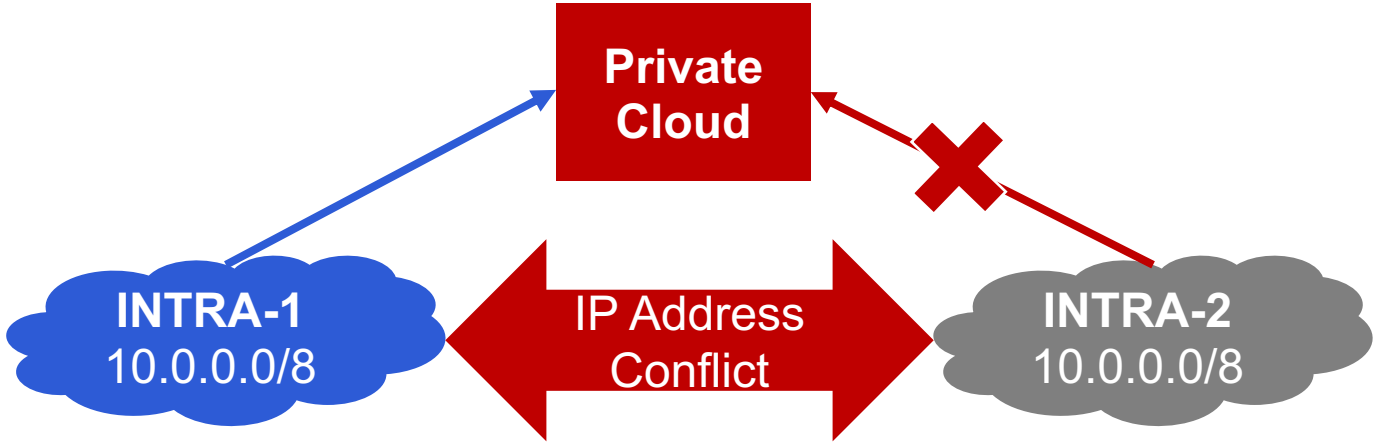
- **グローバルに 3 拠点、4 データセンター を展開**
- **Backbone Network で相互接続** (JANOG43)

- **新規マネージドサービスの拡充**
- **ユーザ向けトレーニングの実施**

https://power-point-design.com/ppt-design/world-map-for-powerpoint/

懸念事項

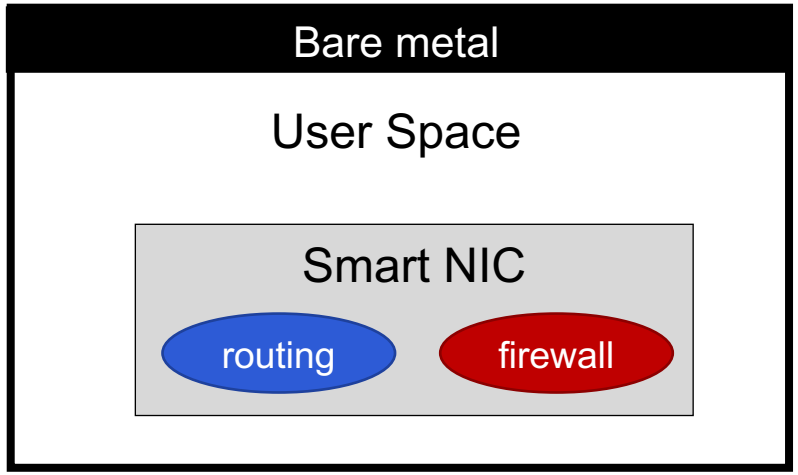
- 社内 Private IPv4アドレスリソースの残りが少ない
  - 新しい POD ができるごとに大量消費
  - オーバーレイを使わずフラットネットワークで設計したため
  - 内部ネットワークも IPv6 化を検討、クラスEの解放
- 既存プラットフォームとの接続する際のセキュリティポリシー
  - 対向がマルチテナントネットワークの設計ではないのでポリシーの定義が大変
- 既存ネットワークとの IPアドレス空間の重複
  - 専用VRF を作って NAT (検証中)



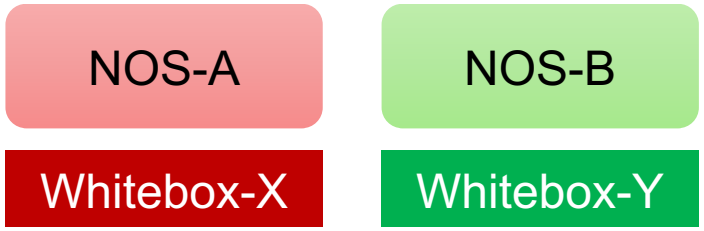
今後の取り組み

ネットワーク機能のオフロード

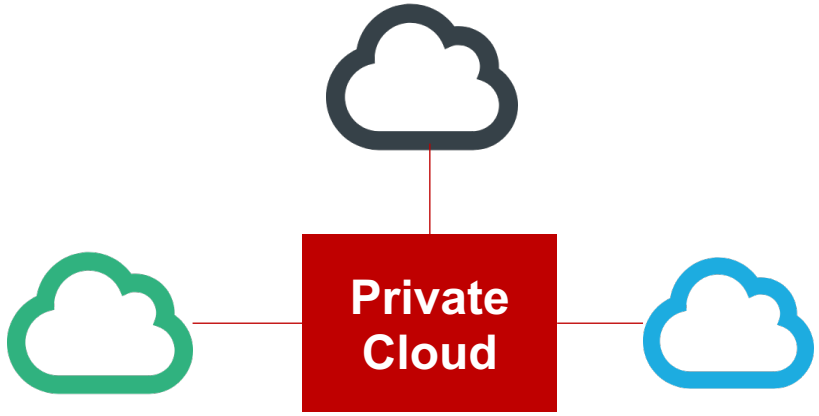
- ユーザに純粋なベアメタルサーバの性能を提供したい
  - Smart NIC へのパケット処理オフロード
- ベアメタル内にユーザ不可侵なコントロールポイント
  - 1VM: 1Hypervisor 化などの検討



次の Whitebox NOS の検討



Public Cloud との接続性/親和性を強化



まとめ

変化の激しい現代に対応可能な継戦能力が高いインフラ基盤を実戦投入した

自社サービスのための本格的なクラウドを構築した

- マネージドサービス指向に舵切り、コアインフラ基盤を簡素化
- マルチテナント対応 + サーバ L3 化による安定性とスケーラビリティ
- Network-aaS としてセキュリティ機構を組み込んだネットワーク機能の提供

質問 & 議論


Rakuten