KDDIのサービスを支えるプライベートCaaS基盤「GANTRY」の紹介

KDDI株式会社　ソリューション事業本部
サービス企画開発本部　プラットフォーム技術部
主任　和田雄太郎
主任　野島幸大

Tomorrow, Together
KDDI

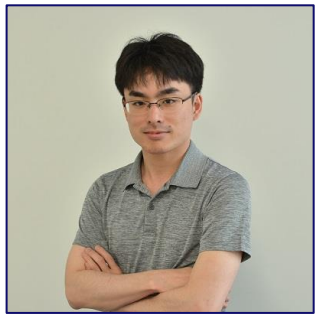
自己紹介

■ 名前：和田 雄太郎（ワダ ユウタロウ）

■ 所属：KDDI プラットフォーム技術部

プロジェクトチームリーダーとしてプライベートクラウドのCaaSプラットフォームの企画開発を実施。

また、コンテナ技術に関する知見を活かし次世代ネットワーク向けプラットフォームの開発にも従事。



■ 名前：野島 幸大（ノジマ コウタ）

■ 所属：KDDI プラットフォーム技術部

開発のメインメンバーとしてプライベートクラウドのCaaSプラットフォームの内製開発を推進。

また、アジャイル開発におけるスクラムマスターを兼任し、プラットフォームの開発チームを率いる。

目次

背景

KDDIにおけるクラウド利用ポリシー

基本的にはクラウドの
利用を志向

アプリケーション
開発部門

最適化ポリシー

取り扱うデータや
サービスの性質などで判断

社内クラウド

**OpenStackベースの
IaaSを提供**

事業用プライベートクラウド

社外クラウド

**各クラウド事業者が
さまざまな機能を
提供**

パブリッククラウド

KDDIにおけるクラウド利用の問題

最適化ポリシー上
社内クラウドを利用したい

もっと手軽にコンテナを稼働
できるプラットフォームはないの？
（社外クラウドならあるのに…）

アプリケーション
開発部門

最適化ポリシー

IaaSだけでは
ニーズに対応しきれない

社内クラウド

enStackベースの
IaaSを提供

事業用プライベートクラウド

社外クラウド

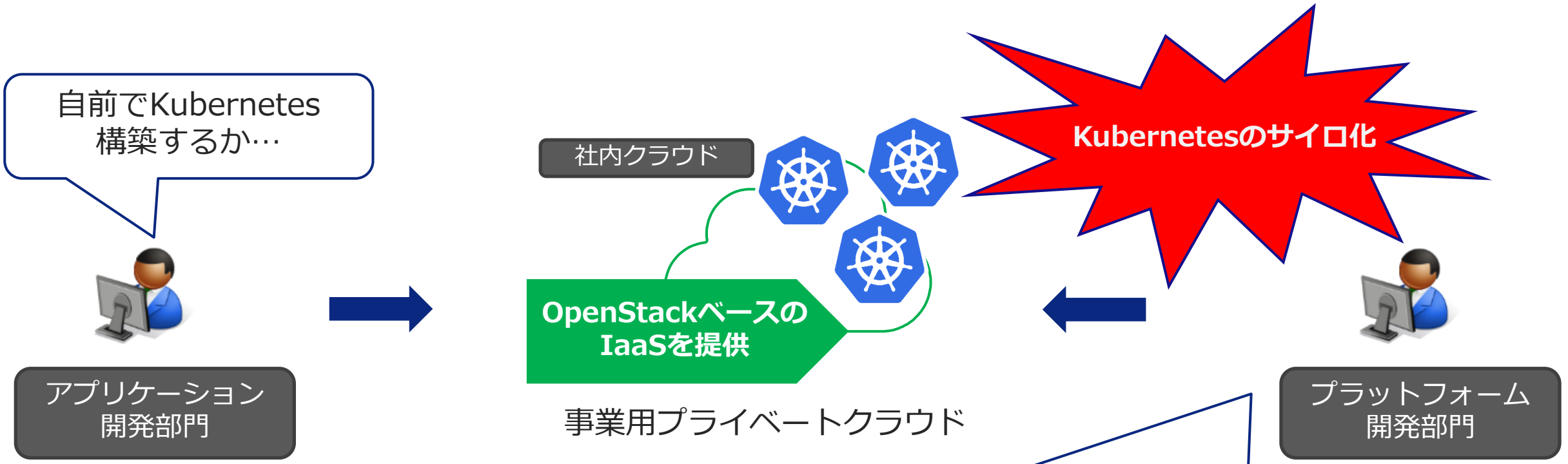
各クラウド事業者が
さまざまな機能を
提供

パブリッククラウド

新規開発したらアプリが
コンテナ化されていた

CI/CDでアプリケーション
のリリースを行いたい

KDDIにおけるクラウド利用の問題



- **Kubernetesに関する知見はプラットフォーム開発部門に集約**したい
- 個別構築された**Kubernetesを集約することで全体最適**を図りたい

KDDIプライベートクラウドにおけるCaaS開発

アプリケーション
開発部門

Kubernetesを
プラットフォーム側で管理して
もらうことで
アプリケーション開発に注力したい

App

App

App

社内クラウド

**OpenStackベースの
IaaSを提供**

事業用プライベートクラウド

従来提供していた**IaaSに加え、
CaaSも提供**したい

プラットフォーム
開発部門

プライベートCaaS基盤 GANTRYの概要

CaaSを実現するGANTRY Project

GANTRY (ガントリー) とは

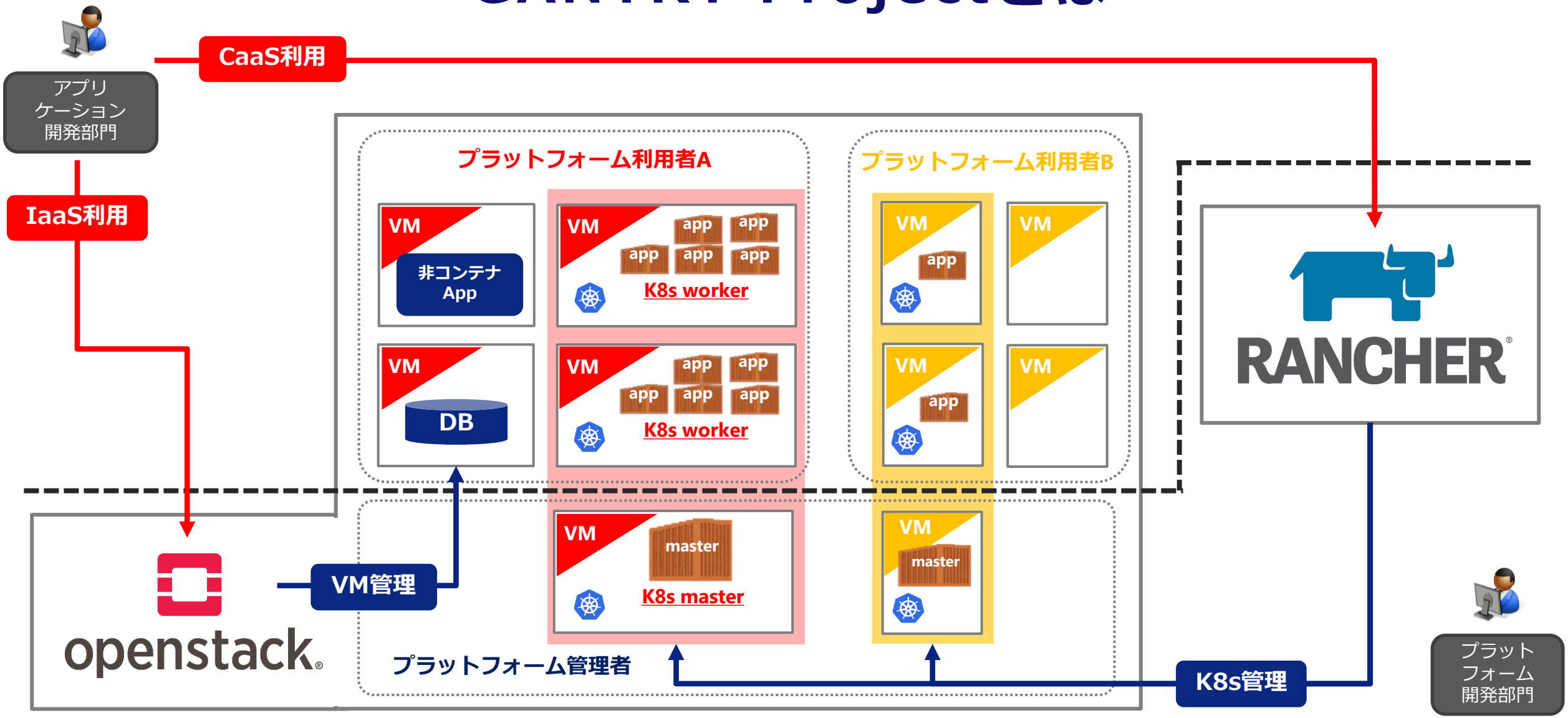
- OpenStackベースの事業用プライベートクラウド上でCaaSを実現するプロジェクト
- 利用者が利用したときに、**「すぐに」「柔軟な」「メンテナンスフリー」**なKubernetesをサービスとして提供

GANTRY 名前の由来

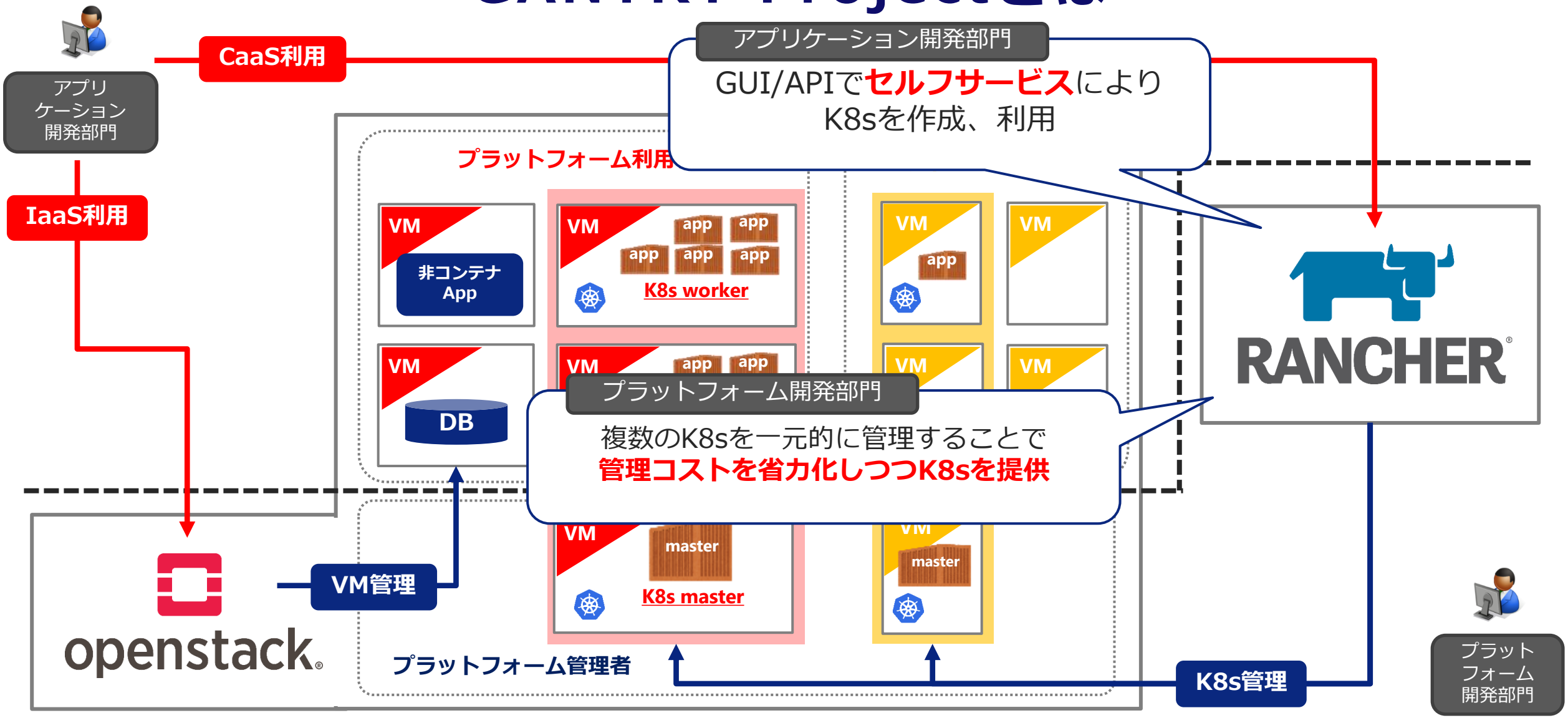
gantry crane【名】コンテナ用貨物船に対してコンテナの積み卸しを行うクレーン。
→ 転じて、**コンテナを自在に操作できるプラットフォーム**実現への思いを込めて命名

Tomorrow, Together
KDDI

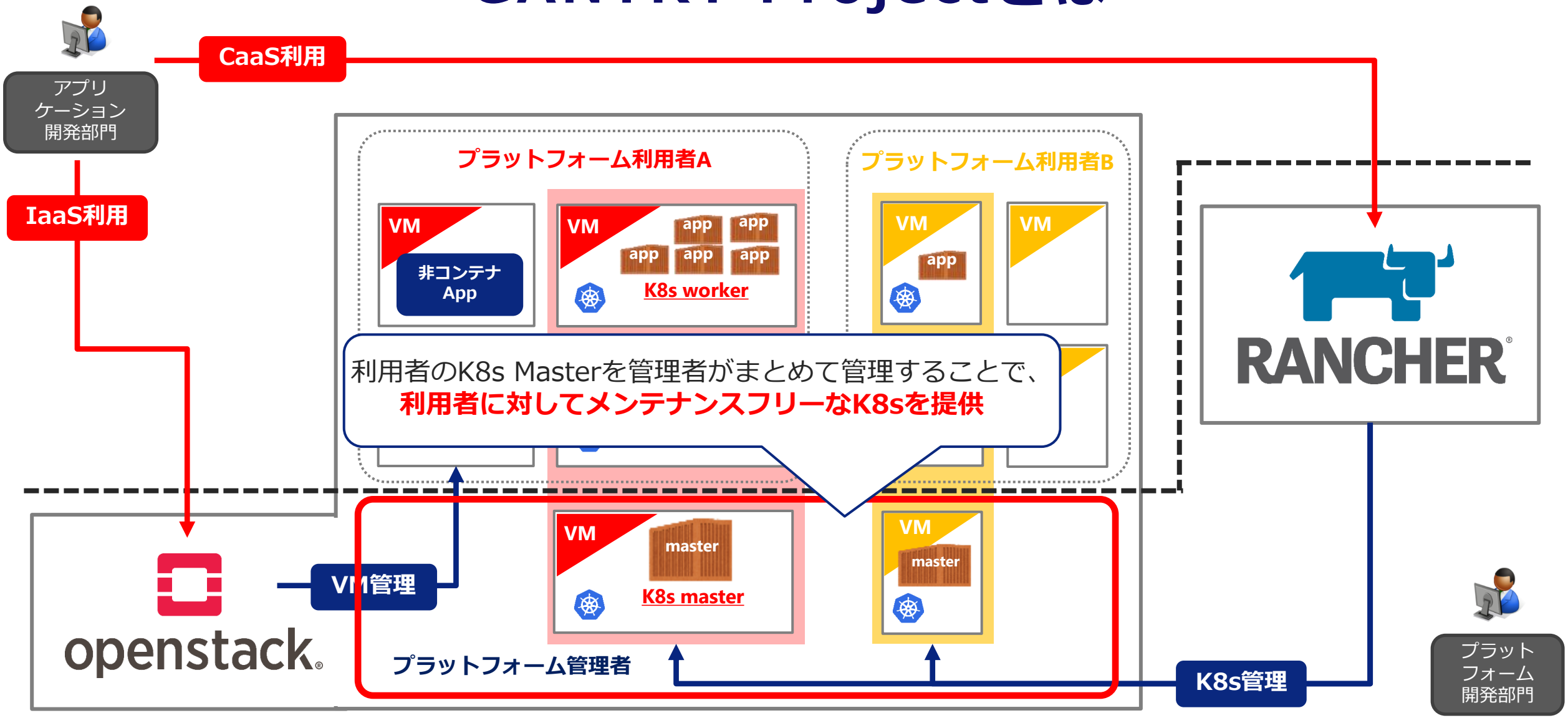
GANTRY Projectとは



GANTRY Projectとは



GANTRY Projectとは



GANTRY Projectとは

GANTRYプロダクト選定
なぜRancherを選んだのか

GANTRYにおけるプロダクト選定

GANTRYにおけるプロダクト選定として、重要視した4つのポイント

① すでにある**IaaS(OpenStack)と連動した Container as a Serviceを提供**できること

② 利用者側で**Kubernetes Masterの運用が不要なサービス**の実現ができること

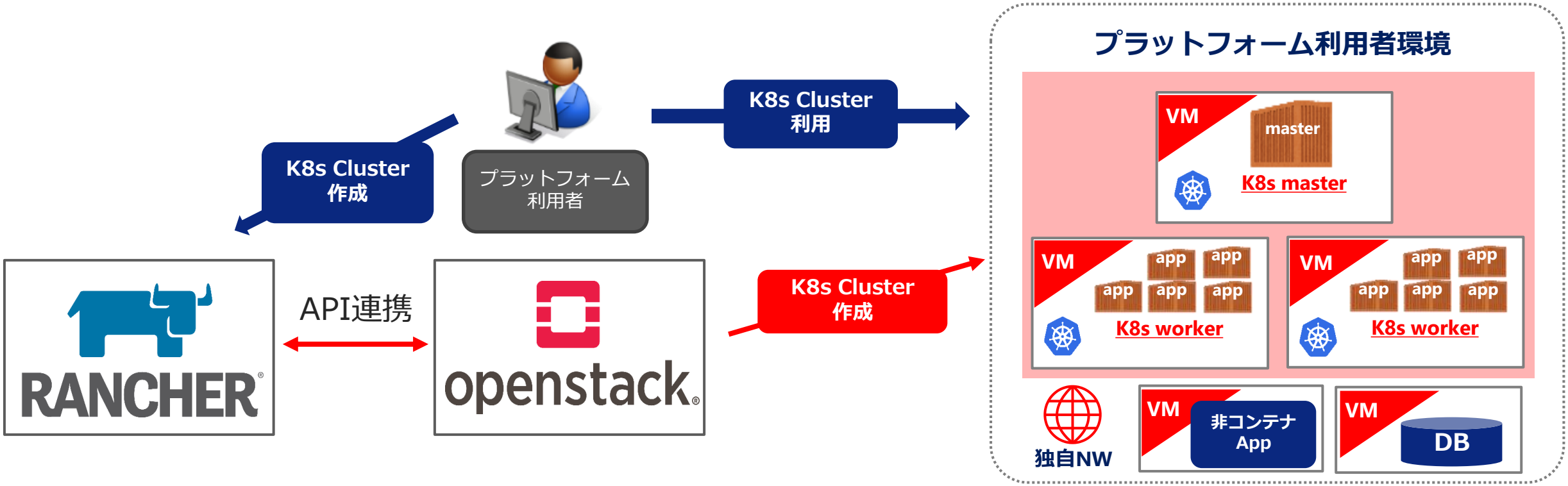
③ KubernetesでKubernetesを管理し、**複数の利用者のKubernetes Clusterを一元的に管理**できること

④ **管理されたKubernetes Clusterの情報を利用した独自の機能拡張**ができること

利用者が必要となった時にすぐにKubernetesを利用でき、多数のClusterを効率的に運用できる基盤を目指す

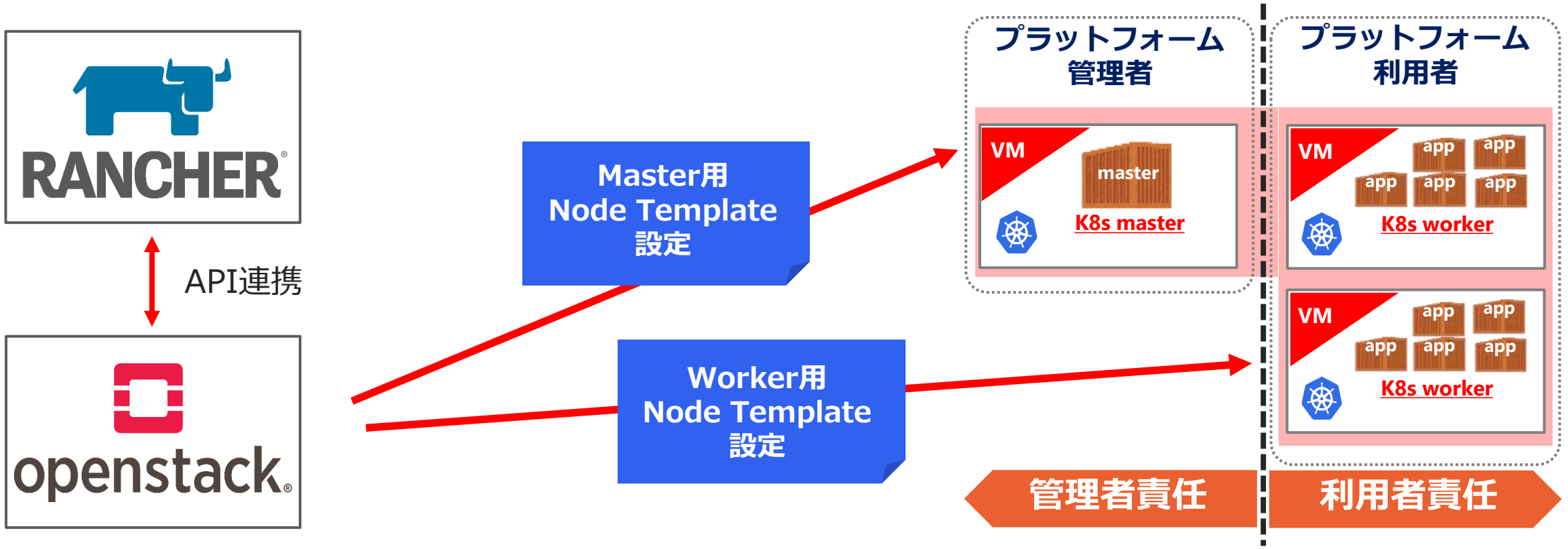
①IaaSと連動したContainer as a Serviceの実現

- 利用者は**Rancherが提供するKubernetesの管理コンソール**(GUI, API)で、Clusterの作成、アクセスが可能
- 対応したIaaSであれば、**Clusterの作成、削除、変更操作は自動**で実施可能
- IaaS上にClusterを生成できるので、**すでにIaaS上で活用しているVM/システムと連動した構成も容易**に構築可能



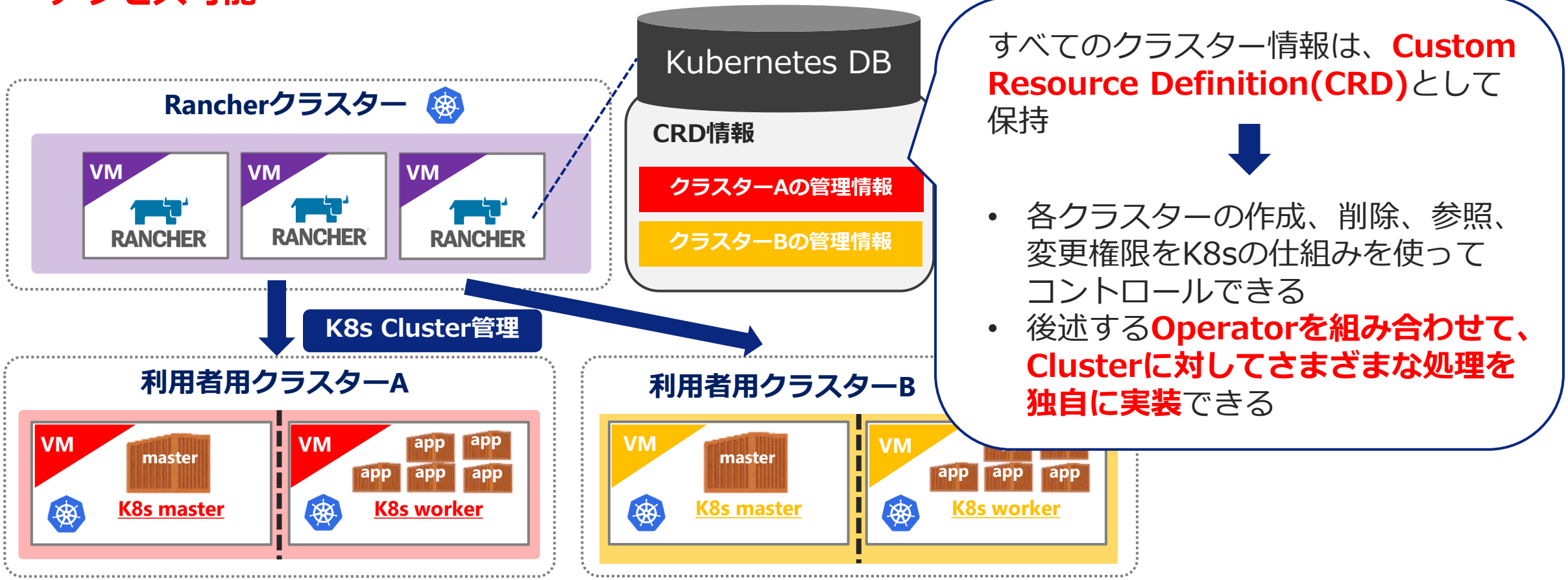
②利用者側でKubernetes運用が不要なサービスの実現

- Rancherの「Node Template」を利用して、**Master/Workerを異なるテナントに作成**
  - Masterは管理者側のテナント：管理者が管理
  - Workerは利用者側のテナント：利用者が管理
- **テナントで責任範囲を分離**しつつ、**Worker側は利用者の要件に合わせた構成**が可能



③複数Kubernetes Cluster一元管理の実現

- RancherはKubernetesのDB（etcd）に管理情報をCRDとして保持
- 特別なインターフェースを実装せずとも、Kubernetesの仕組みを使って管理情報へアクセス可能

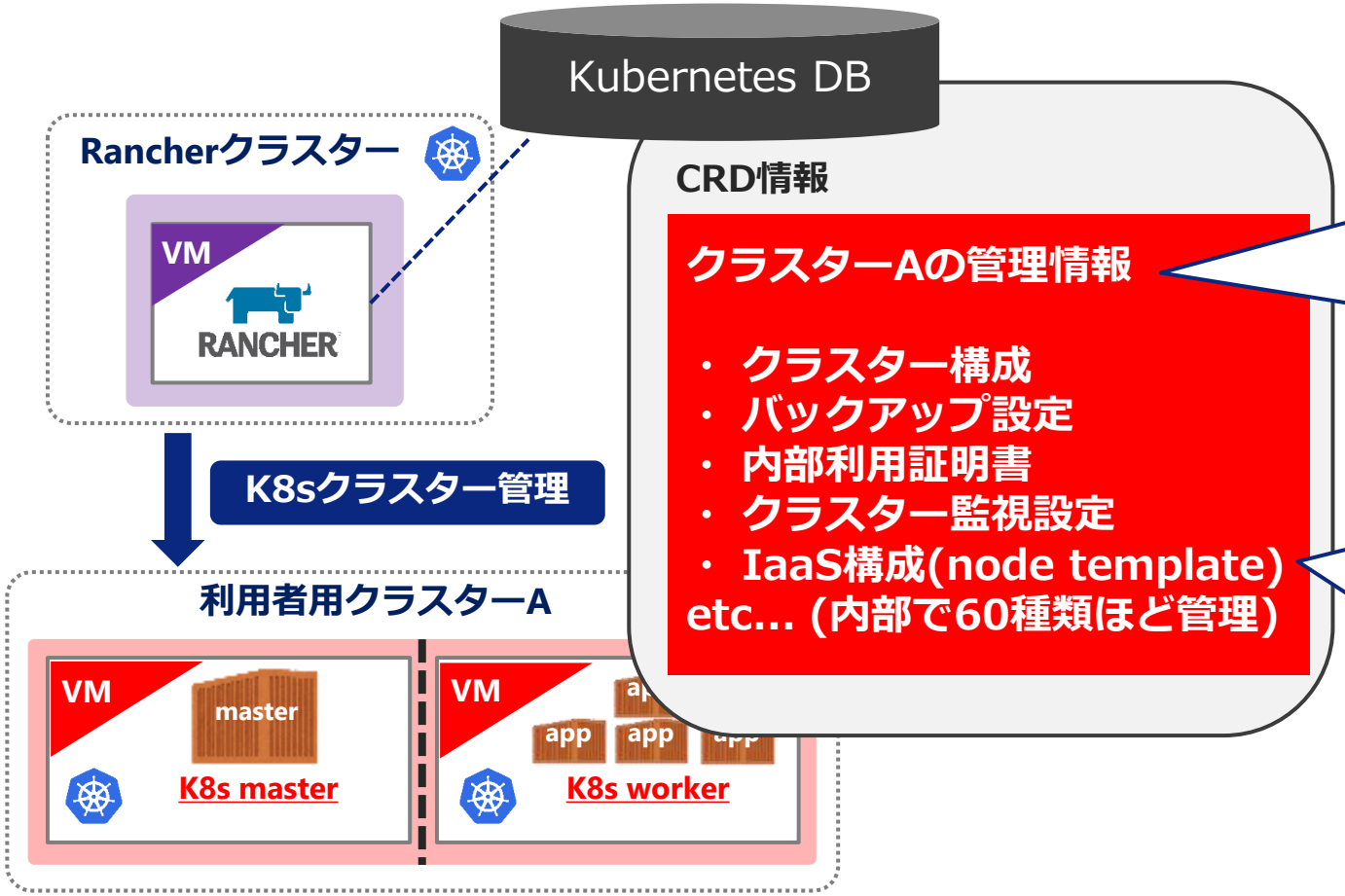


すべてのクラスター情報は、Custom Resource Definition(CRD)として保持

- 各クラスターの作成、削除、参照、変更権限をK8sの仕組みを使ってコントロールできる
- 後述するOperatorを組み合わせて、Clusterに対してさまざまな処理を独自に実装できる

④Clusterの情報を利用した独自の機能拡張の実現

- CRDで管理された情報を参照して機能拡張のために活用可能



Rancher内部で実装されたOperator

- K8s Operator
  K8sのクラスターの作成、削除、変更を管理する
- etcd Operator
  K8sのクラスター内部で利用されているDBのバックアップと復元を管理する
- cert Operator
  K8sのクラスター内部で利用されている証明書を管理する（自動更新など）

GANTRY独自で実装した機能拡張

- 構成管理
  IaaS/CaaSの構成情報を紐づけて管理できる仕組みの実現
- 監視自動登録
  利用者のクラスターが作成されたことを自動検知して、**K8sのコンポーネントの監視を自動で行う機能**の実装

GANTRYにおけるプロダクト選定

①	すでにある**IaaS(OpenStack)と連動した Container as a Serviceを提供**できること	○
②	利用者側で**Kubernetes Masterの運用が不要なサービス**の実現ができること	○
③	KubernetesでKubernetesを管理し、**複数の利用者のKubernetes Clusterを一元的に管理**できること	○
④	**管理されたKubernetes Clusterの情報を利用した独自の機能拡張**ができること	○

複数プロダクトを比較検討した結果、①～④を満たすKubernetes管理製品としてRancherを採用

KDDIにおけるプラットフォーム開発

Rancherを導入するにあたっての壁

Rancherの標準機能で我々の要件を満たせなかった点

① OpenStackによるKubernetesノードの自動作成

② 管理者による利用者のKubernetes Master監視

③ 物理ホスト上のコンテナ稼働状況の把握

要件に合うように機能のカスタマイズ、追加実装を実施

壁の破り方

足りない機能については新規開発し問題を解決

パートナーとの内製開発

3つの問題について機能開発を実施

① OpenStackによるKubernetesノードの自動作成

② 管理者による利用者のKubernetes Master監視

③ 物理ホスト上のコンテナ稼働状況の把握

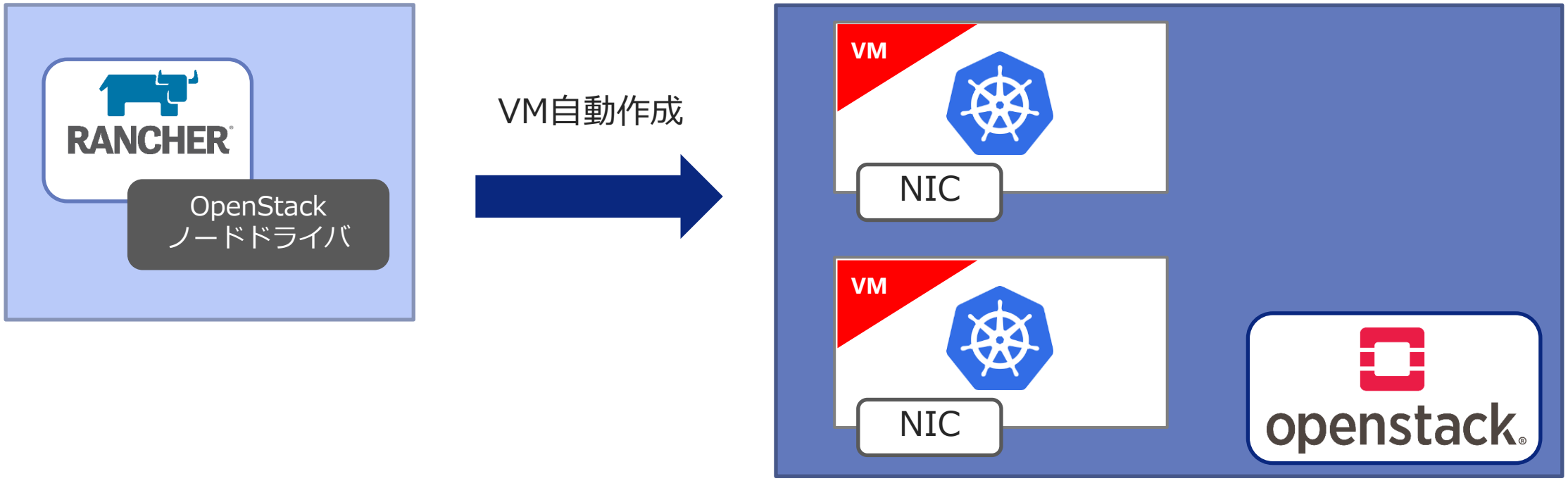
① Multi NIC対応OpenStackノードドライバの開発

② 利用者のKubernetes Master監視自動登録機能の開発

③ コンテナ情報取得APIの開発

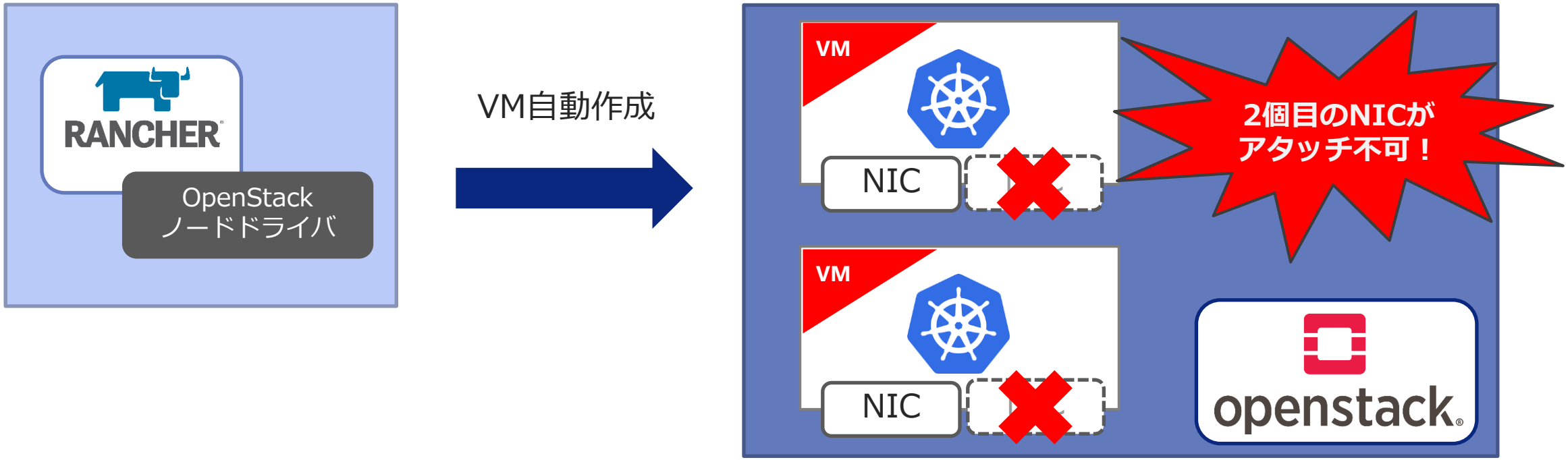
①OpenStackによるKubernetesノードの自動作成

- **K8s構築、増設時にノードを自動で作成したい**
  - ▶ **OpenStackノードドライバを利用**
    - → **Rancher標準のOpenStackノードドライバでは複数NICを利用不可**
      それでも複数NICはつけたい・・・ (監視用、K8s内部通信用、サービス用)



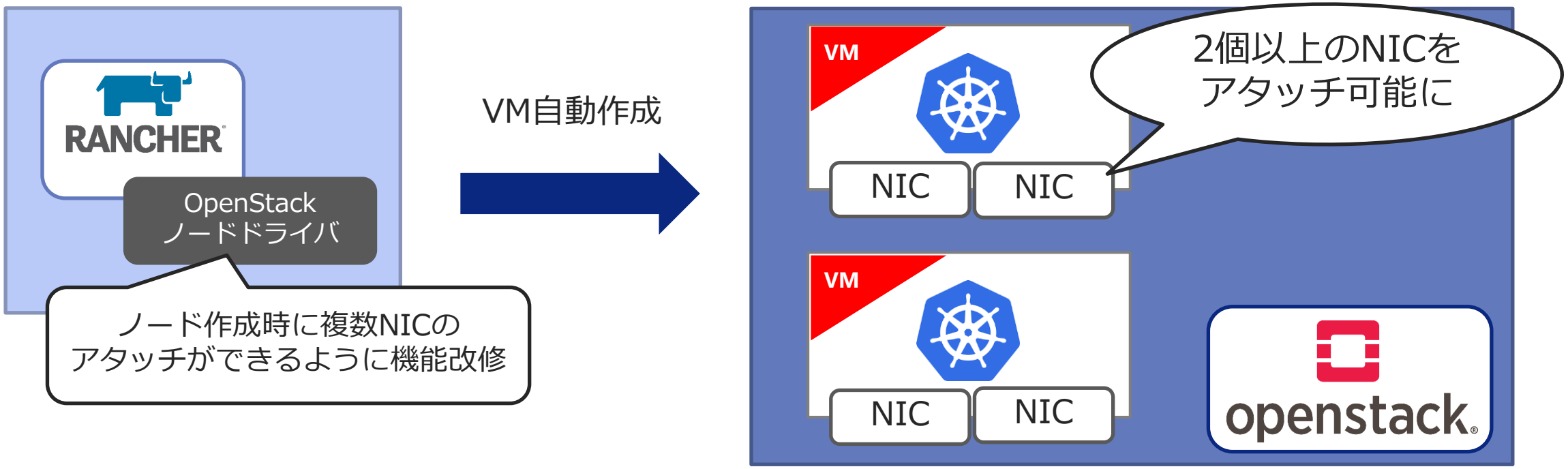
①OpenStackによるKubernetesノードの自動作成

- **K8s構築、増設時にノードを自動で作成したい**
  - ▶ **OpenStackノードドライバを利用**
    - → **Rancher標準のOpenStackノードドライバでは複数NICを利用不可**
      それでも複数NICはつけたい・・・ (監視用、K8s内部通信用、サービス用)



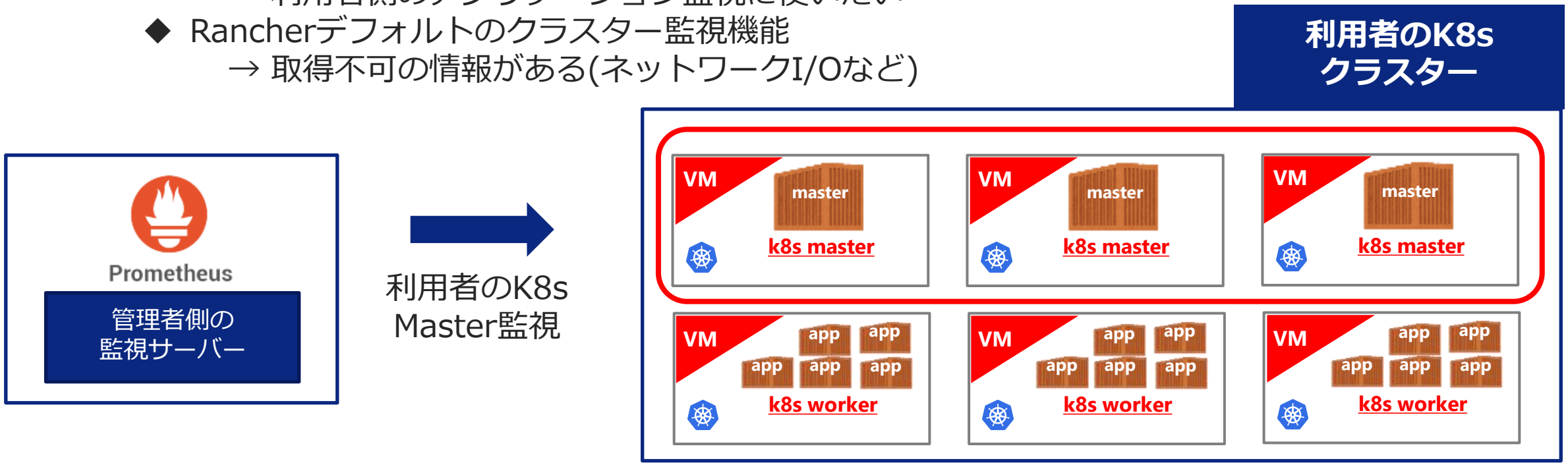
①Multi NIC対応OpenStackノードドライバの開発

■ OpenStack用のノードドライバを開発
- Rancher提供のノードドライバを修正
- 複数NWのアタッチが可能



②管理者による利用者のKubernetes Master監視

■ 利用者に払い出したKubernetesの**Masterを外部から監視したい**
- ▶ Kubernetes Masterはプラットフォーム管理者側で監視、運用をしたい
- ▶ Rancher**標準の監視機能では難しい**
  - ◆ RancherのMonitoring機能
    → 利用者側のアプリケーション監視に使いたい
  - ◆ Rancherデフォルトのクラスター監視機能
    → 取得不可の情報がある(ネットワークI/Oなど)



②利用者のKubernetes Master監視自動登録機能の開発

■監視用PrometheusにK8sの監視設定を自動投入

- KubernetesのOperatorとして動作

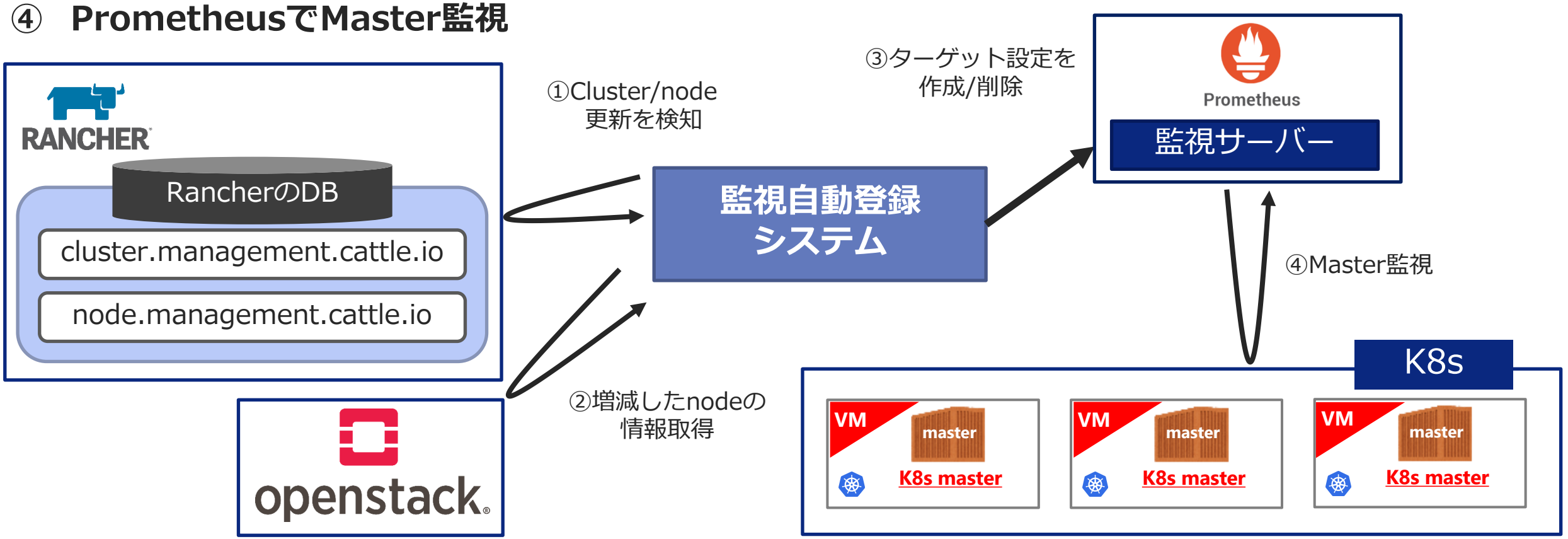
Step1
- RancherのCluster/node CRDを監視

Step2
- ノード作成を検知したらOpenStackからノード情報を取得

Step3
- 取得したノード情報からPrometheusのターゲット設定を作成し登録

処理のイメージ

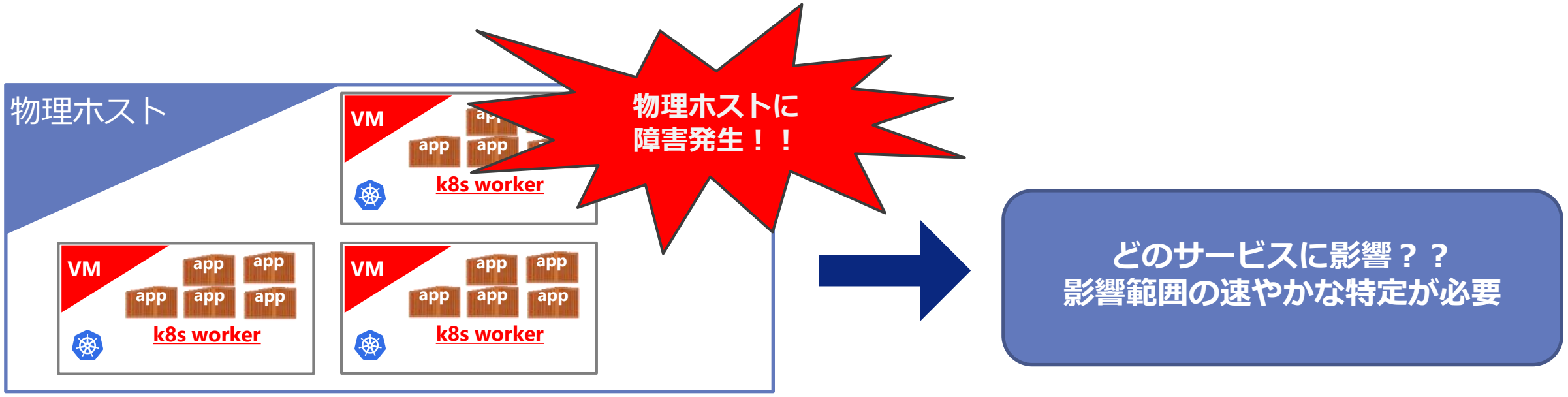
① Cluster/node更新を検知
② 増減したnodeの情報取得
③ ターゲット設定を作成/削除
④ PrometheusでMaster監視



③物理ホスト上のコンテナ稼働状況の把握

■ どの**物理ホストでどのコンテナが動いているか**知りたい

- ▶ **障害が起こった時の通知に利用**
  - ◆ 物理ホストダウン → ホスト上のVMもダウン → そのVM上のコンテナもダウン
  - ◆ 障害時に影響を受けた利用者に連絡する必要がある → 品質基準を満たすため
  - ◆ 物理ホスト-VM-コンテナの構成の一元管理が必要
- ▶ **構成管理用のDBにコンテナ情報を保存したい**
  - ◆ 情報は自動更新が必要



③物理ホスト上のコンテナ稼働状況の把握

■ どの**物理ホストでどのコンテナが動いているか**知りたい

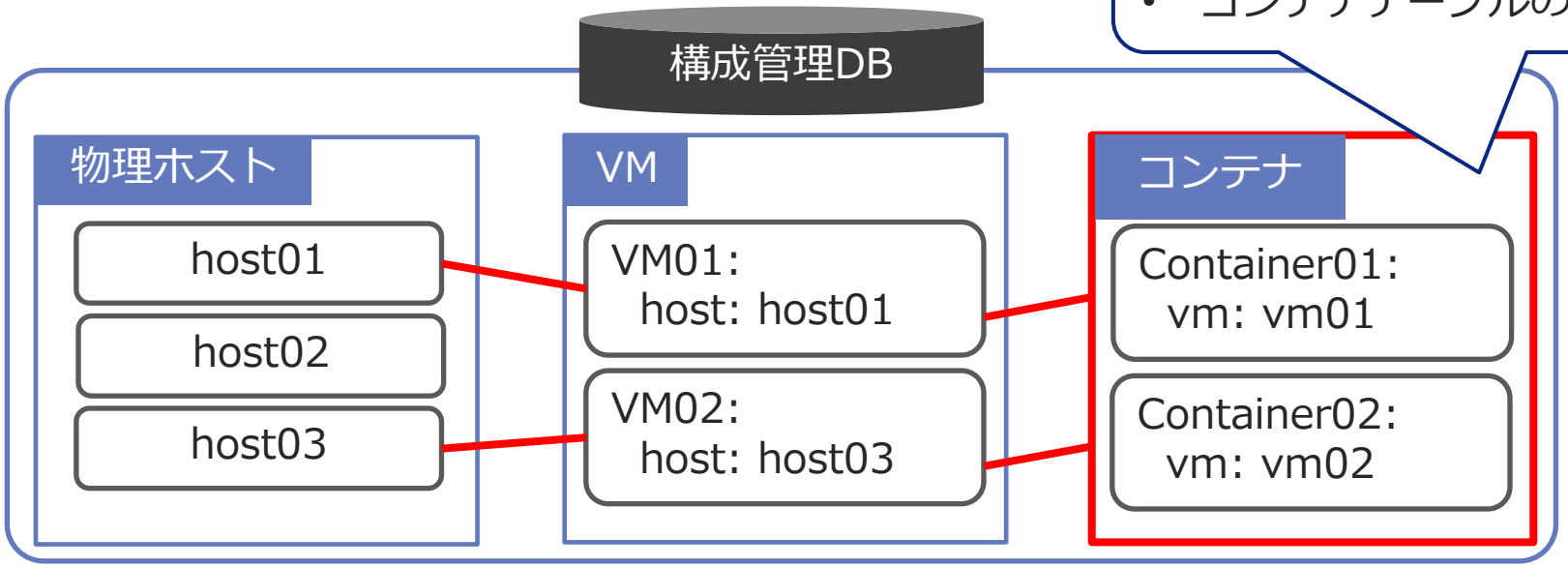
▶ **障害が起こった時の通知に利用**

- ◆ 物理ホストダウン → ホスト上のVMもダウン → そのVM上のコンテナもダウン
- ◆ 障害時に影響を受けた利用者に連絡する必要がある → 品質基準を満たすため
- ◆ 物理ホスト-VM-コンテナの構成の一元管理が必要

▶ **構成管理用のDBにコンテナ情報を保存したい**

- ◆ 情報は自動更新が必要

- 物理ホスト – VMの構成に紐づくコンテナのテーブルを自動追加
- コンテナテーブルの情報は自動更新



③物理ホスト上のコンテナ稼働状況の把握

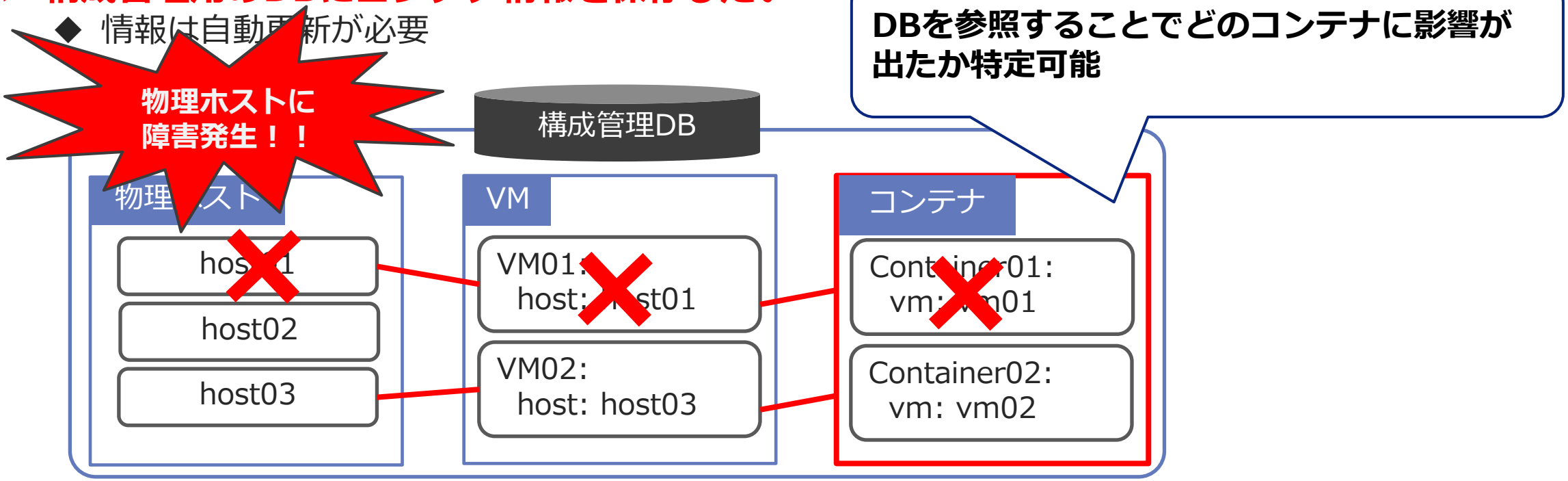
■ どの**物理ホストでどのコンテナが動いているか**知りたい

▶ **障害が起こった時の通知に利用**

- ◆ 物理ホストダウン → ホスト上のVMもダウン → そのVM上のコンテナもダウン
- ◆ 障害時に影響を受けた利用者に連絡する必要がある → 品質基準を満たすため
- ◆ 物理ホスト-VM-コンテナの構成の一元管理が必要

▶ **構成管理用のDBにコンテナ情報を保存したい**

- ◆ 情報は自動更新が必要



③コンテナ情報取得APIの開発

■ **構成管理DBに必要な情報を一元管理させるための情報を提供**

■ **ノードごとのコンテナー覧を取得**

▶構成管理DBから呼び出される

▶REST API形式

Step1
- Rancherからノード情報、利用者のK8sからPod情報を取得
- 利用者K8sへのアクセスはRancherを経由

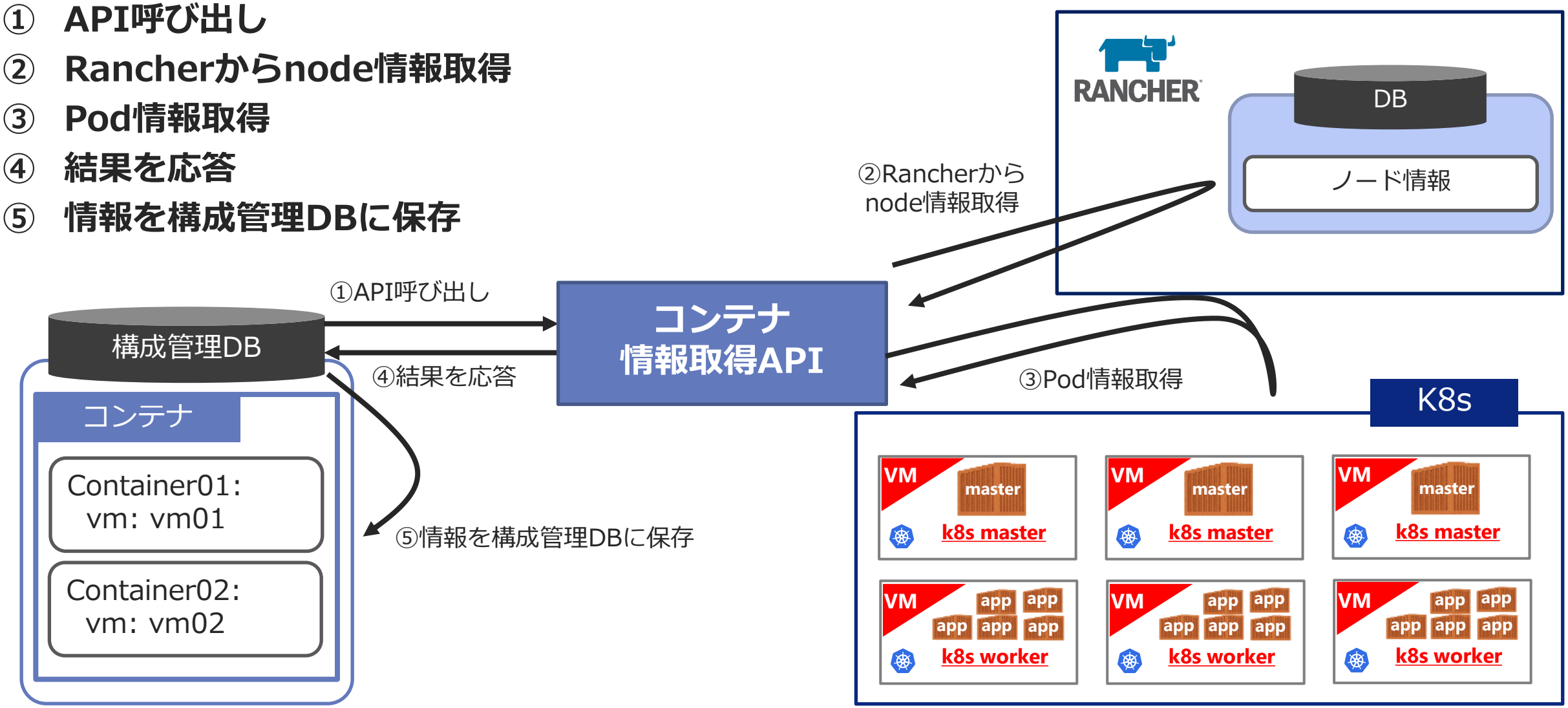
Step2
- ノードごとのコンテナのリストを返す

Step3
- 構成管理DBは返された情報を保存

Tomorrow, Together KDDI

処理のイメージ

① **API呼び出し**
② **Rancherからnode情報取得**
③ **Pod情報取得**
④ **結果を応答**
⑤ **情報を構成管理DBに保存**



得られた価値

得られた価値

■それぞれの機能を開発し得られた効果

▶①Multi NIC対応OpenStackノードドライバの開発

◆設計通りのノード作成を達成

◆利用者が複数NWを利用した柔軟な設計が可能に

▶②利用者のKubernetes Master監視自動登録機能の開発

◆Rancherによるノードの作成/削除に合わせて監視対象登録を達成

◆利用者に監視コンポーネントの存在を意識させずに監視が可能に

◆仕組みを流用しさらなる機能拡張も可能

▶③コンテナ情報取得APIの開発

◆物理ホスト-仮想マシン-コンテナの一元管理を達成

◆利用者が障害時の影響をいち早く知ることが可能に

◆社内の評判が1番よかった

今後の展望

今後の展望

■ Rancherのクラスター管理機能を利用したエンハンス

▶ 監視自動登録の仕組みを利用し機能拡張

- ◆ （例）K8sクラスターの作成を検知しLBなど外部のツールを自動構築

■ 利用者がアプリケーションを使う際の利便性向上

▶ MasterノードのK8sコンポーネントのログ取得

- ◆ MasterノードはGANTRY管理者の責任範囲なので、利用者は直接アクセス不可
- ◆ 責任範囲外でもデバッグなどで利用者がAPIのログを見たいことはあり得る
- ◆ 責任分界を壊さない範囲でログを利用者に提供する仕組みを作る

■ GANTRYの拡張計画

- ▶ 2020年度中に**東西両拠点でのGANTRYサービスを展開**
- ▶ 2020年から2021年にかけて、利用者数を倍増し、サービスを軌道に乗せる

Rancherへの要望

Rancherへの要望

- **Rancherにあったらいいなと思っているもの**
  - **修正したOpenStackノードドライバの公式採用**
    - パートナーと開発した際プルリクエストは送ったが・・・
  - **Rancherの長期サポートバージョン**
    - 1年ごとにアップデートするのはつらい・・・

まとめ

まとめ

■ まとめ

▶ KDDIのプライベートクラウドとしてRancherを用いたGANTRYをリリース

- ◆ OpenStack上にK8sを展開する形で利用者にK8sを提供
  - インフラ構築の工数を半減
- ◆ K8s Masterの運用なしでK8sを利用可能
  - プラットフォーム利用者におけるKubernetes Masterに関わる運用コストがゼロに
  - アプリ開発に注力するすることで、迅速にアプリのリリースが可能に

▶ Rancherを活用した機能拡張

- ◆ Multi-NIC対応OpenStackノードドライバの開発
- ◆ RancherのCRDを利用し利用者K8sクラスターの監視自動登録機能の開発
- ◆ 物理層からコンテナ層まで一貫した構成管理の実現

利用者/管理者双方の**運用コストを低減したK8s管理プラットフォーム**の実現

Tomorrow, Together

KDDI