ＪＡＮＯＧ４４　ライトニングトーク（１）

k8sを使ってレガシーなネットワークの運用自動化を始めました*事前ドラフト版*

KDDI 次世代運用推進本部　コアネットワーク部　辻　広志

２０１９年7月２５日

枚数が多いので当日版は差し替えます…！

Tomorrow, Together

KDDI

本日の発表内容

- **運用**と**開発**が同じ組織になり
- 『**脱・ベンダ任せ**』をスローガンに
- **レガシーなネットワーク**で
- **k8sをプラットフォーム**に
- **モダン**な運用を目指してます

自己紹介

辻　広志

兵庫県　三田（さんだ）市出身

自然と歴史豊かないい街です！

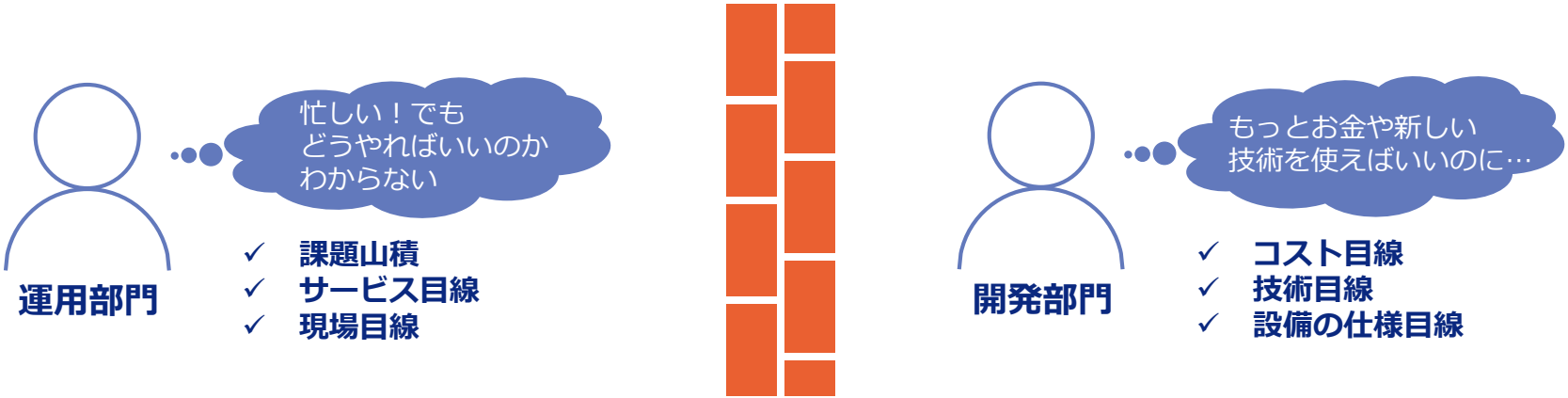
KDDIでの主な業務は
固定電話サービスの「開発」と「運用（のお手伝い）」。

メインはシステムのNFV化
※OpenStackやMANOと呼ばれるNFV領域、DPDKなどが得意

好きなRFCはRFC3261

画像出典：https://www.city.sanda.lg.jp/kouhou/gaiyou.html

きっかけは運用と開発の組織統合



組織統合

- 若手中心で運用環境の改善活動を開始

脱・ベンダ任せ／「ハック」で自ら近代化を加速

自分たちの欲しい仕組みは自分たちで作ろう

Hack accelerates modernization

HAM Project

今回自動化で対処した課題

いまさら・・・固定電話？
それでも運用していれば障害・トラブルは起こる

大規模災害

イベント輻輳

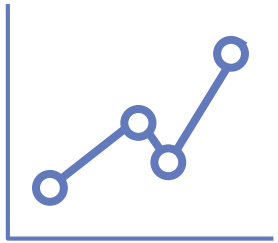
サイレント障害

障害時のデータ集計業務が大変

- なにか起こっていることを**瞬時に把握できない**
- **影響範囲の特定に時間**がかかる
- **古い設備**なので、人手で**頑張って耐える**

- 各種データを**事が起こる前から**集約し、
- 有事の際には**すぐに取り出せる**ようにしよう

データの集約ツールの選定と課題



Metrics

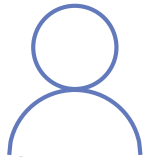
- ✓ 統計情報
- ✓ ログの集計結果
- ✓ 数値データ

Logs

- ✓ アクセスログ
- ✓ 信号ログ
- ✓ 文字列データ

elasticsearch

kibana

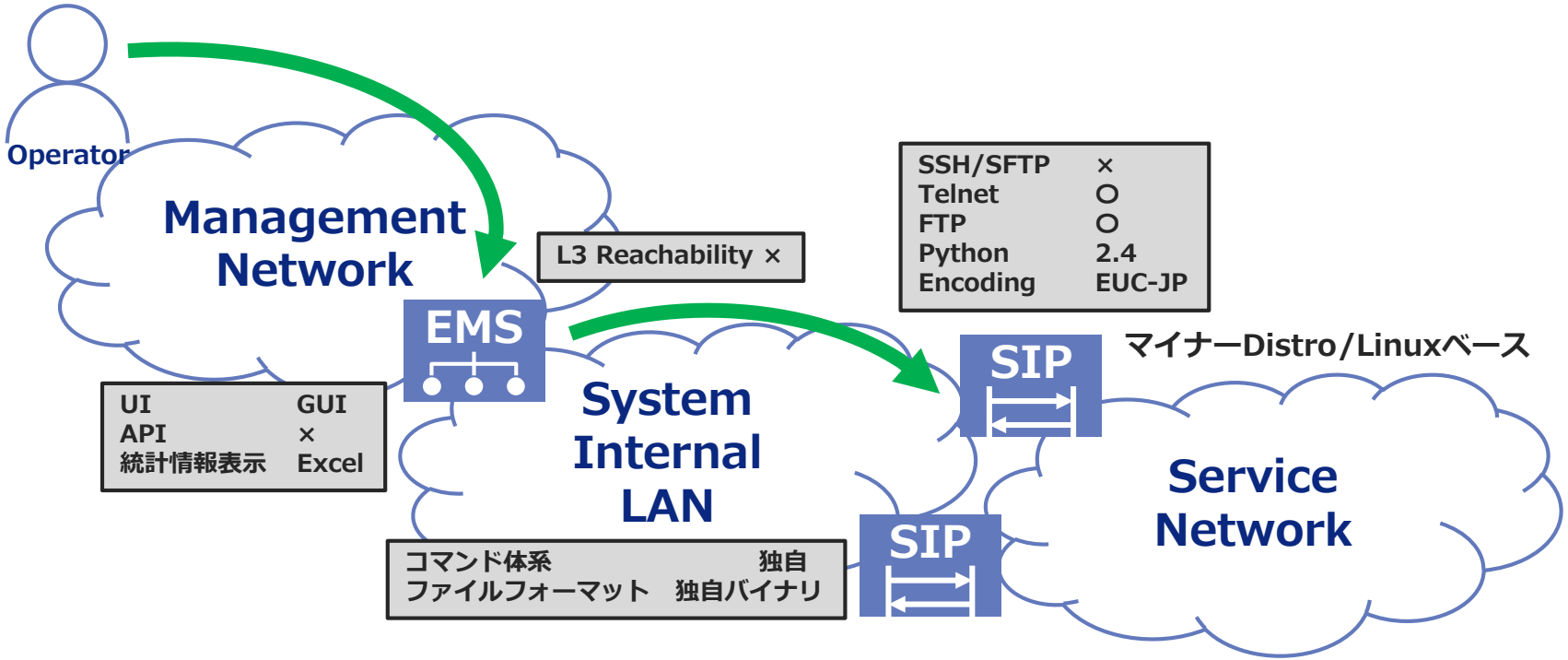


Operator

複数種類のデータソースをElasticsearchに投入し可視化したい

どれぐらいリソースがいりそう？柔軟にスケールするか？

古い設備・レガシーなネットワークとは



レガシーネットワークに対する自動化ツールと課題

LogstashやFilebeatなどは使えない
ネットワークの構成は変えられない・・・
Ansibleなどもそのままでは使えない・・・

行きついた先は自分たちでスクリプトを書くこと
Pythonぐらいならがんばって書いたほうが早そう

管理方法や信頼性に課題

- どこでどうやって実行する？
- 確実性や信頼性は？
- 障害時は動く？

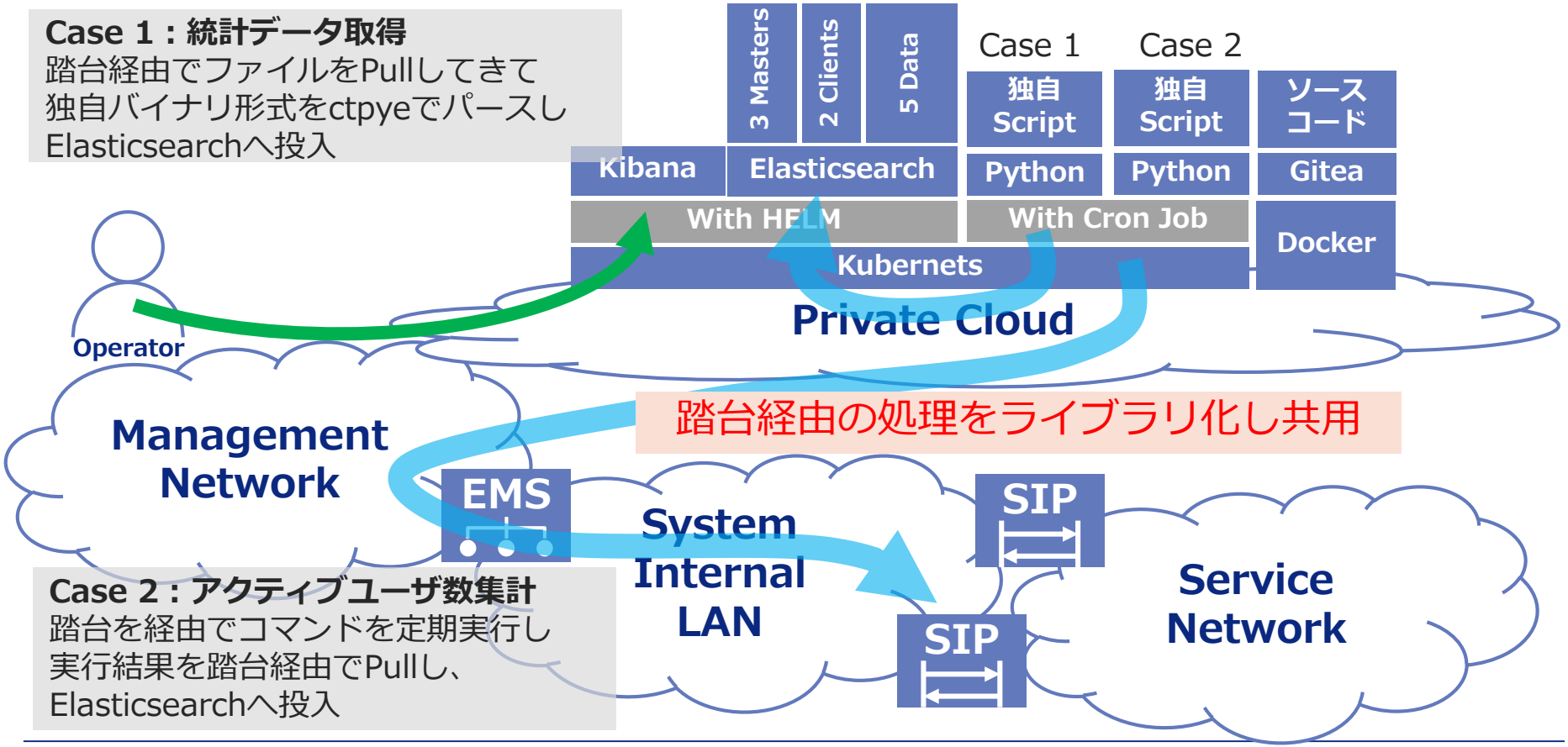
プラットフォームの検討

課題
- 自作スクリプト等の高信頼化
- スケーラビリティ

	On-premises	Virtual machines	Containers
Costs	✕ 物理サーバ 場所・電力	○ VM払い出し	○ VM払い出し
Availability	✕ 自前保守	△ VMのHAあり	○ Replica セルフヒール
Flexibility	✕ 物理増設	△ VM増設 VM単位分割	○ VM増設 CPU時間単位
Complexity	○ シンプル	△ まぁまぁ シンプル	✕ マルチレイヤ 仮想化
Functionality	✕ DIY	✕ DIY	○ Helm Cron Job

全体像

Case 1：統計データ取得
踏台経由でファイルをPullしてきて
独自バイナリ形式をctpyeでパースし
Elasticsearchへ投入



Case 2：アクティブユーザ数集計
踏台を経由でコマンドを定期実行し
実行結果を踏台経由でPullし、
Elasticsearchへ投入

実際に運用してみた結果

課題だったこと	効果	副作用
データの集計業務	当初の狙い通り工数減	データがまだまだ足りない
Elasticsearchのスケール	Helmで楽々インストール	Cloudのリソース不足からスケールできない・・・
Pythonによる自動化ツール作成	標準ライブラリも豊富で柔軟な処理が可能 ライブラリの共通化を実現	ノウハウ不足
自作スクリプトの高信頼化	k8sの機能の恩恵を授かる コンテナでバージョンコントロールも楽々	コンテナを理解できる技術者不足

まとめと　今後の展望

Why Kubernetes ?

- レプリカ生成やセルフヒーリングなどの冗長化
- Cron Jobによる自作ジョブの高信頼での実行
- HELMによるOSSの簡単なデプロイ
- ソフトウェア開発のための高機能で使いやすいプラットフォームは
  当然運用自動化にも役に立った

効果のほどは？

- k8sを含めて、思ってたよりちゃんと動いている
- 「欲しいものは自ら作る」ことは、スキルは必要だが自動化への圧倒的近道

今後に向けては？

- 「Kubernetesの運用」はやっぱりつらい（それ本業じゃない）
- 基盤の拡張、課題の吸い上げと実装
- スキル向上、作れる、触れる人をどんどん増やす


Tomorrow, Together
KDDI