CNF (Cloud-native Network Function)の設計自動化の検討について

2021/2/26

NTTネットワーク基盤技術研究所

平井志久(shiku.hirai.uk@hco.ntt.co.jp)

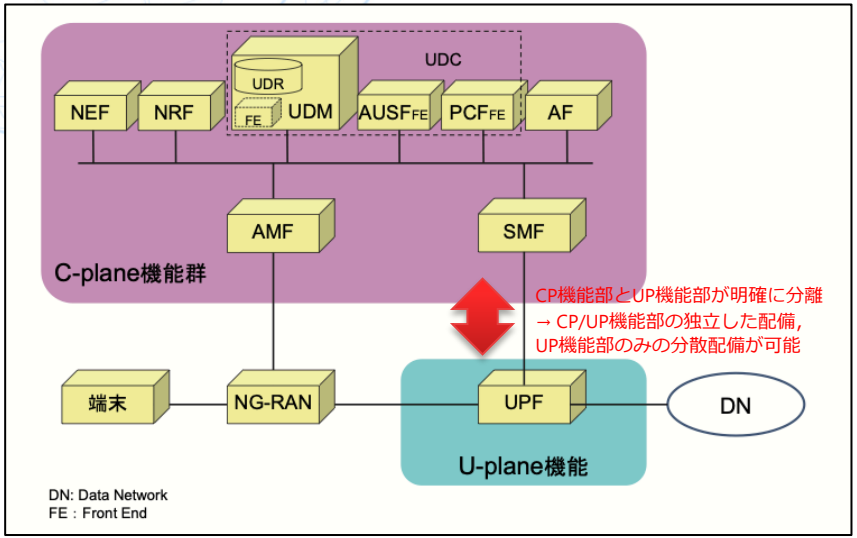
本日お話する内容

- 5Gモバイルコア(5GC)・スライス解説
- NW機能のクラウドネイティブ化とは
- マイクロサービスアーキテクチャって難しい
- CNF設計制御最適化・自動化技術のご紹介
  - CNF自動設計エンジン (ACPE)
  - Kubernetes Operatorによる自律運用制御
  - 5GC制御デモビデオ
- 今後の検討について

5Gモバイルコア(5GC)解説

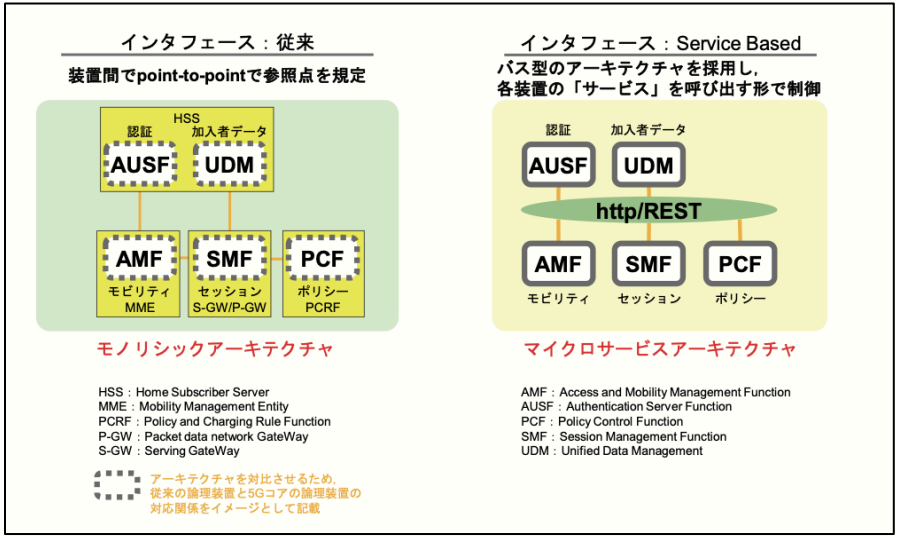
- モバイルNWは主に，RAN(無線アクセス)・TN(トランスポート)・CN(コア)で構成
- CNドメインでは，**モバイルコア**と呼ばれる装置・機能群によって，端末の移動・課金認証・セッション管理機能が提供
- 5Gは既に商用サービスが始まっているが，今後**5Gモバイルコア(5GC)**の導入によって，高速大容量(eMBB)の要件に加え，**高信頼超低遅延(URLLC)・多数同時接続(mMTC)**の実現が期待される

5GC特徴①【C/Uプレーン分離(CUPS)】



(出典) NTT DOCOMOテクニカル・ジャーナル Vol. 25 No. 3(Oct. 2017)
https://www.nttdocomo.co.jp/binary/pdf/corporate/technology/rd/technical_journal/bn/vol25_3/vol25_3_007jp.pdf

5GC特徴②【Service Based Architecture (SBA)】



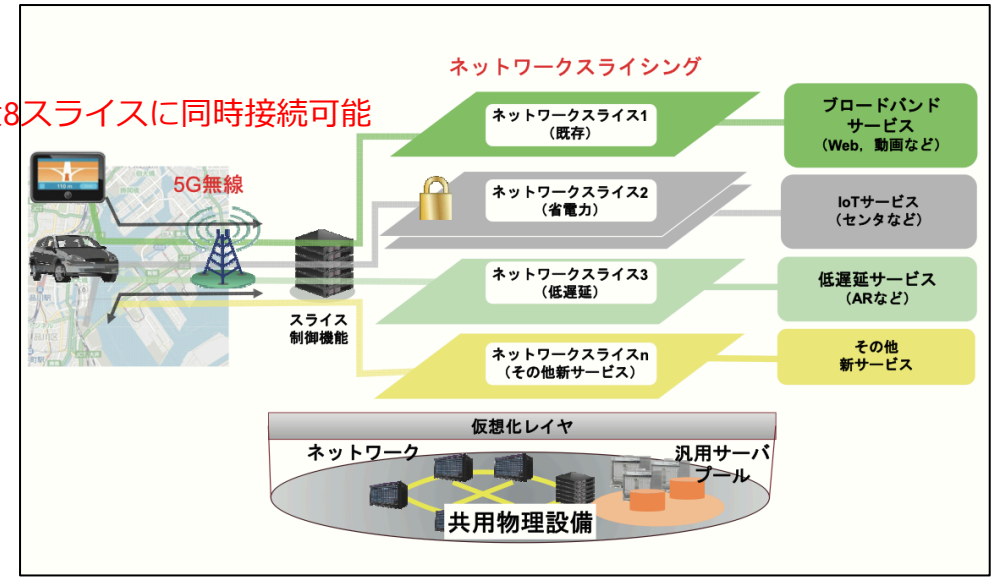
(出典) NTT DOCOMOテクニカル・ジャーナル 25周年記念号(2018)
https://www.nttdocomo.co.jp/binary/pdf/corporate/technology/rd/technical_journal/bn/vol26_e/vol26_e_004jp.pdf

スライス解説

- 5Gモバイルコア(5GC)導入以降，**サービス毎に必要なNWリソースをスライスとして共通インフラから論理的に切り出し提供**
- スライスの基本タイプとしては，**高速大容量 (eMBB)**，**超高信頼低遅延 (URLLC)**，**多数同時接続 (mMTC)**があるが，Beyond 5G時代にはさらに要件の多様化・エクストリーム化

5GC特徴③【ネットワークスライシング】

同一端末が最大8スライスに同時接続可能

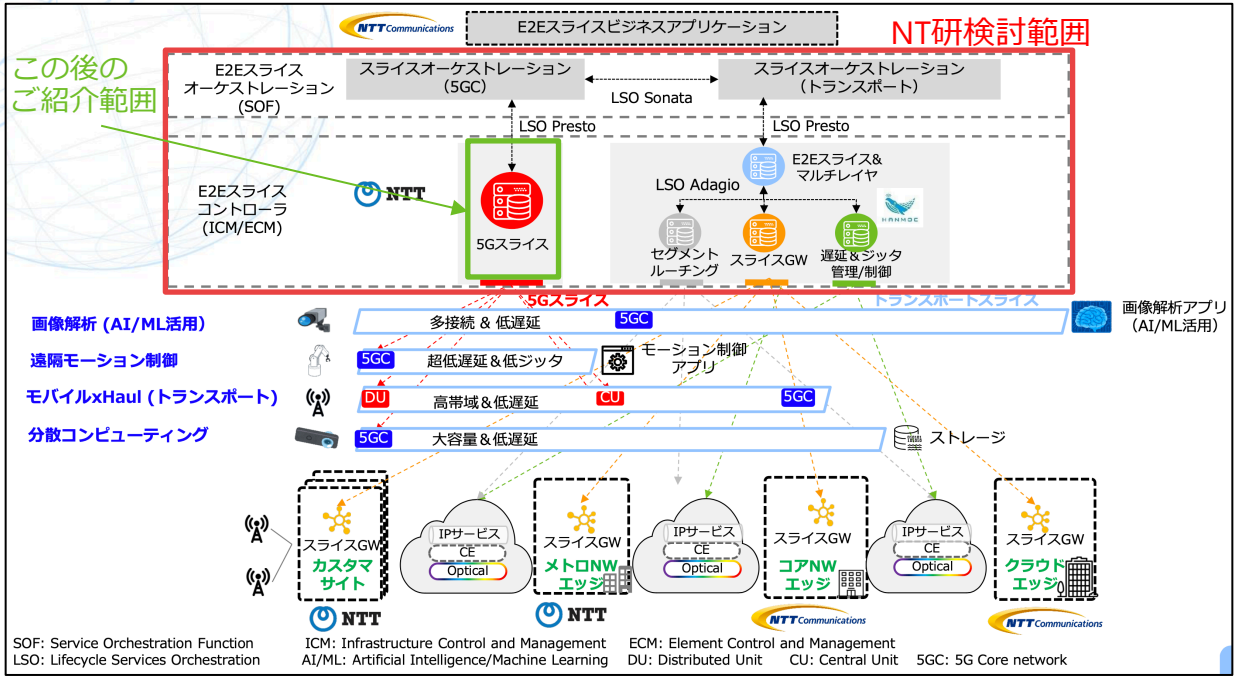


(出典) NTT DOCOMOテクニカル・ジャーナル 25周年記念号(2018)
https://www.nttdocomo.co.jp/binary/pdf/corporate/technology/rd/technical_journal/bn/vol26_e/vol26_e_004jp.pdf

(参考) NT研のE2Eスライス管理制御技術

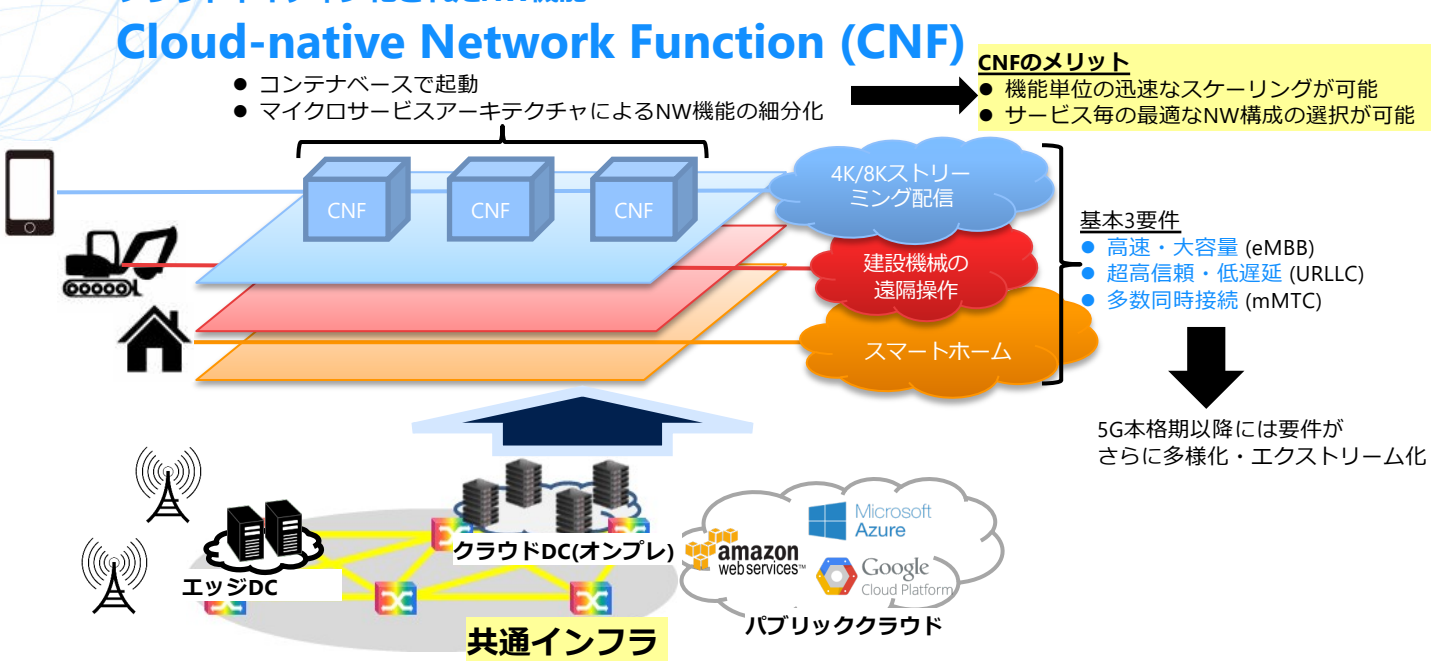
- RAN・TN・CNの複数ドメインに跨り**End-to-End (E2E)でスライスSLAを満たすことが重要**
- NT研では，E2Eスライス管理制御技術として，**各ドメインにおける管理制御コントローラの開発と，上位のオーケストレータとの連携制御**を検討

NTTドコモオープンハウス2021 (2021.2.1～2.7)にも技術出展



NW機能のクラウドネイティブ化

- E2Eスライスを構成するRAN(CU/DU)・CN(5GC)のNW機能は**クラウドネイティブ化(コンテナ起動・マイクロサービスアーキテクチャ導入)**されエッジ～クラウドまで様々なプラットフォームを跨り展開
- コンテナやマイクロサービスアーキテクチャのメリットを活かした，迅速かつ柔軟なスライス提供が期待



マイクロサービスアーキテクチャって難しい

- 3GPP規定の5GCのSBA (Service Based Architecture)では，マイクロサービスアーキテクチャを導入，各NFの機能はNFサービスとして細分化され，NFサービス間はREST APIを介して互いに呼び出し通信を実現
- これらNFサービスは実態として，**複数拠点に跨り起動されることが想定されるため，HWとSWの組合せ数は急激に増大し，特にNW性能を保証・担保するためのリソース設計・制御が困難化**

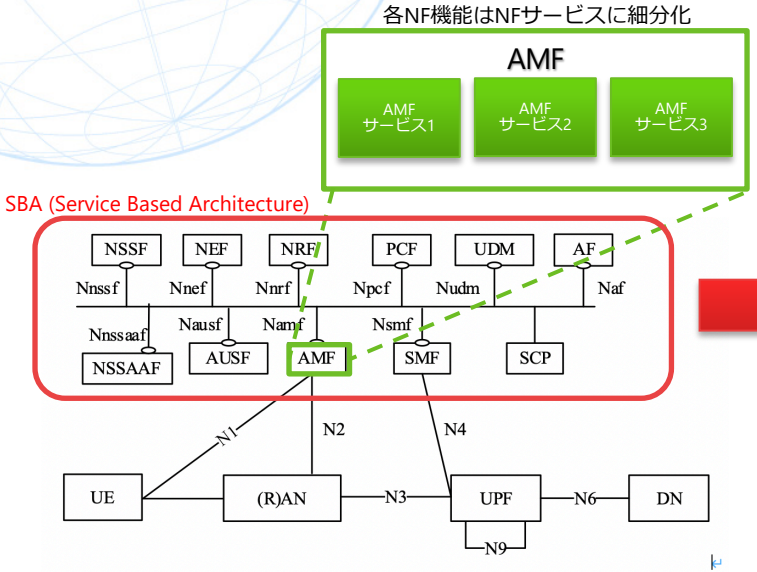
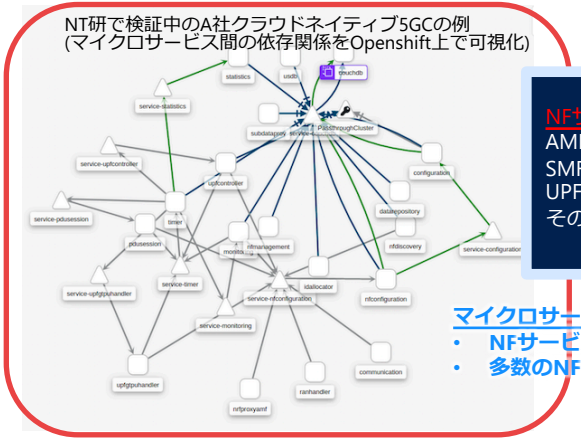


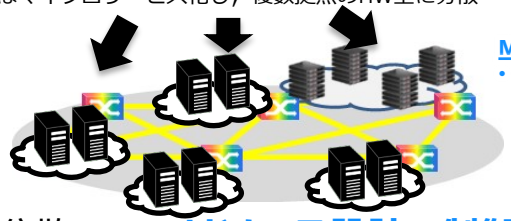
Figure 4.2.3-1: 5G System architecture
(出典) 3GPP TS 23.501 System architecture for the 5G System (5GS)



NFサービスの分割数の例
AMF→ 3種類のNFサービス
SMF→ 2種類のNFサービス
UPF→ 3種類のNFサービス
その他合せて計32NFサービスに分割

マイクロサービス化による影響
- NFサービス間の依存関係は動的に変化し複雑化
- 多数のNFサービスの組合せは膨大

SWはマイクロサービス化し，複数拠点のHW上に分散



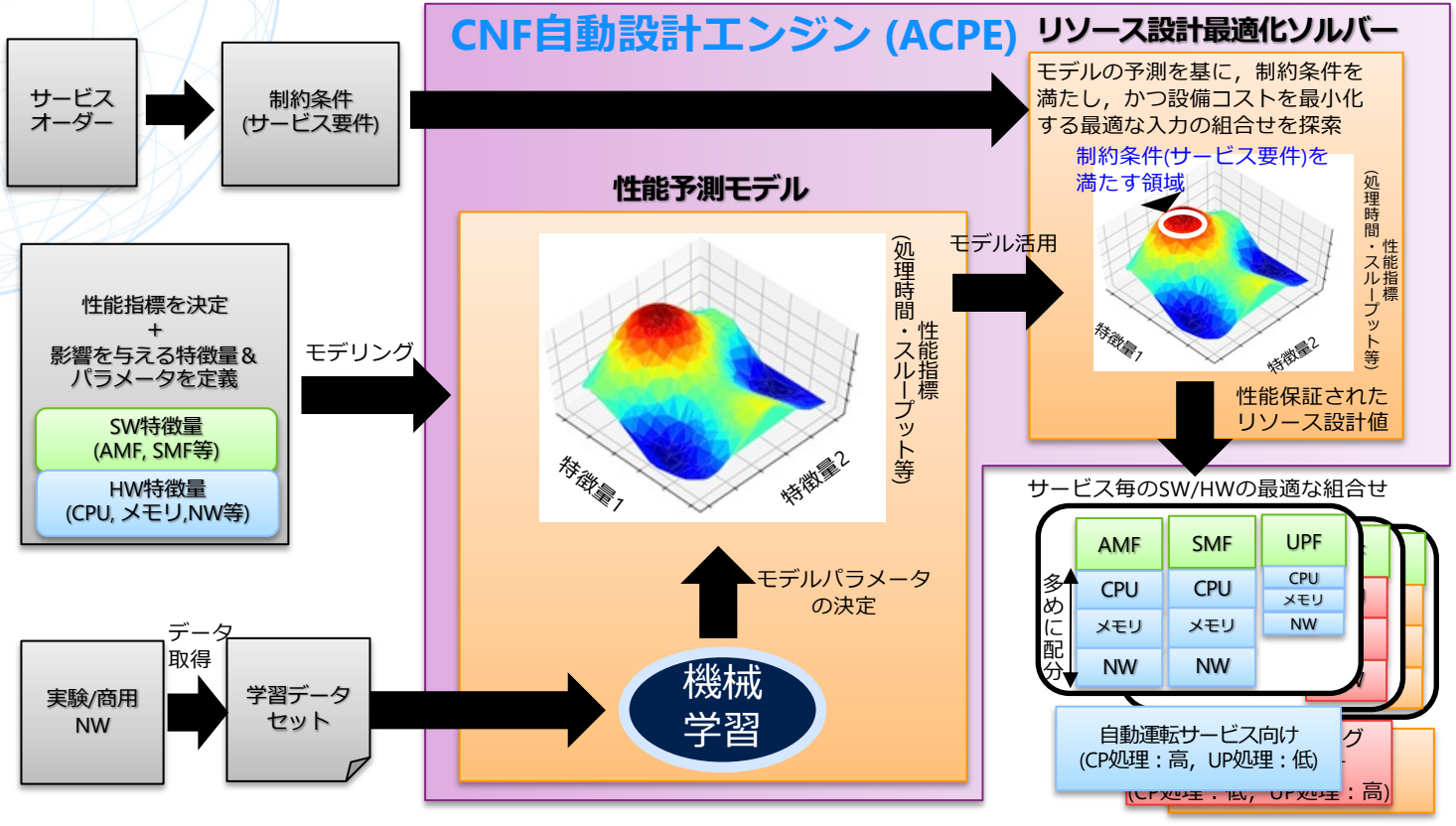
MEC/クラウド等分散配備の課題
- HW差分/共存する
  SW影響によるNW性能の変化

SWのマイクロサービス化×複数拠点のHWへの分散→ **HWリソース設計・制御が困難化**

CNF設計制御最適化・自動化技術のご紹介

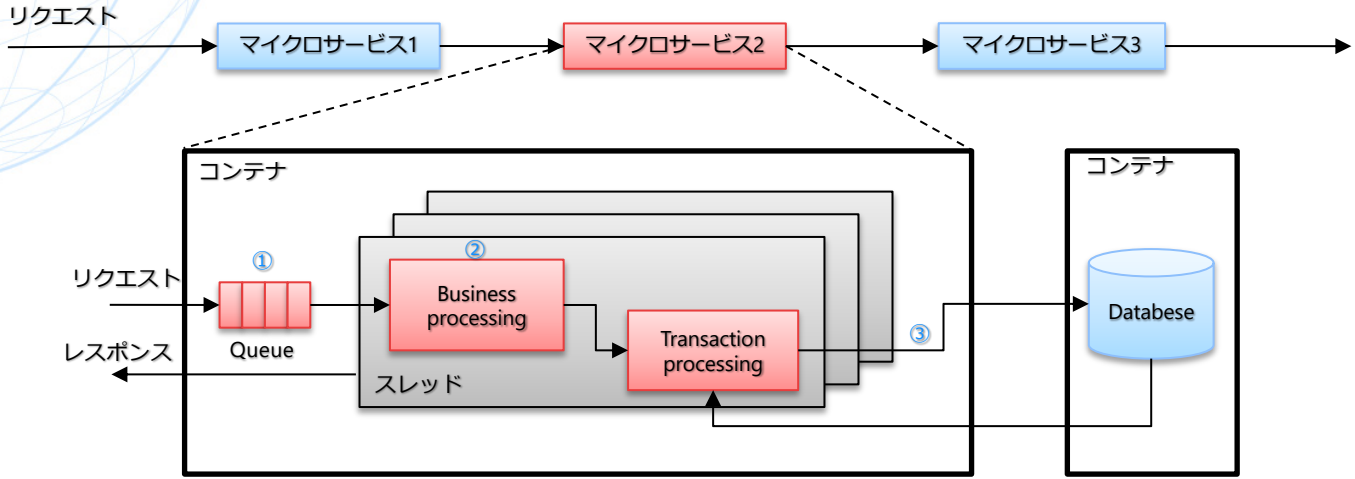
性能推定に基づくCNF設計制御最適化

- NT研ではスライスのサービス要件に基づき，各CNFに割当てるHWリソースを最適化する**CNF自動設計エンジン(ACPE)**を提案
- ACPE内部では，**性能予測モデル**と**リソース設計最適化ソルバー**が連携



(参考) マイクロサービスアプリの性能予測モデル

- Baoら[1]が提案するマイクロサービス・アプリケーションのモデリング手法
  - マイクロサービス・アプリケーションに対して汎用性の高いモデルを構築
  - 性能指標として**リクエスト処理時間**を選択し，それを3要素に分割
  - HWとSWの影響を分離し，機械学習によりモデルパラメータを推定



リクエスト処理時間　$PT(t) = T_r(t) + T_b(t) + T_d(t) + \gamma$

①リクエスト待ち時間　②ビジネスロジック処理時間　③トランザクション処理時間

[1] L. Bao, C. Wu, X. Bu, N. Ren, and M. Shen, "Performance Modeling and Workflow Scheduling of Microservice-Based Applications in Clouds," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 9, pp. 2114–2129, 2019.

(参考) リソース設計最適化ソルバー

NTT

最適化問題を以下のように定式化

目的：設備コスト最小化

設計最適化ソルバー：
設備コスト＝利用時間[sec]×Podが起動するNodeの単位時間当たりの設備コスト[/sec]÷そのNodeで起動するPod総数

利用時間[sec]＝全端末アタッチ完了までの各NFサービス内部処理時間の合計(モデルの出力より)
Podが起動するNodeの単位時間当たりの設備コスト＝CPUコア数＋クロック数＋メモリ量(要規格化)

制約条件を考慮：
CPU： CPU上限量 < Nodeで利用可能なCPU量
メモリ：メモリ上限量 <Nodeで利用可能なメモリ量
処理時間：処理時間 < QoS遅延規定

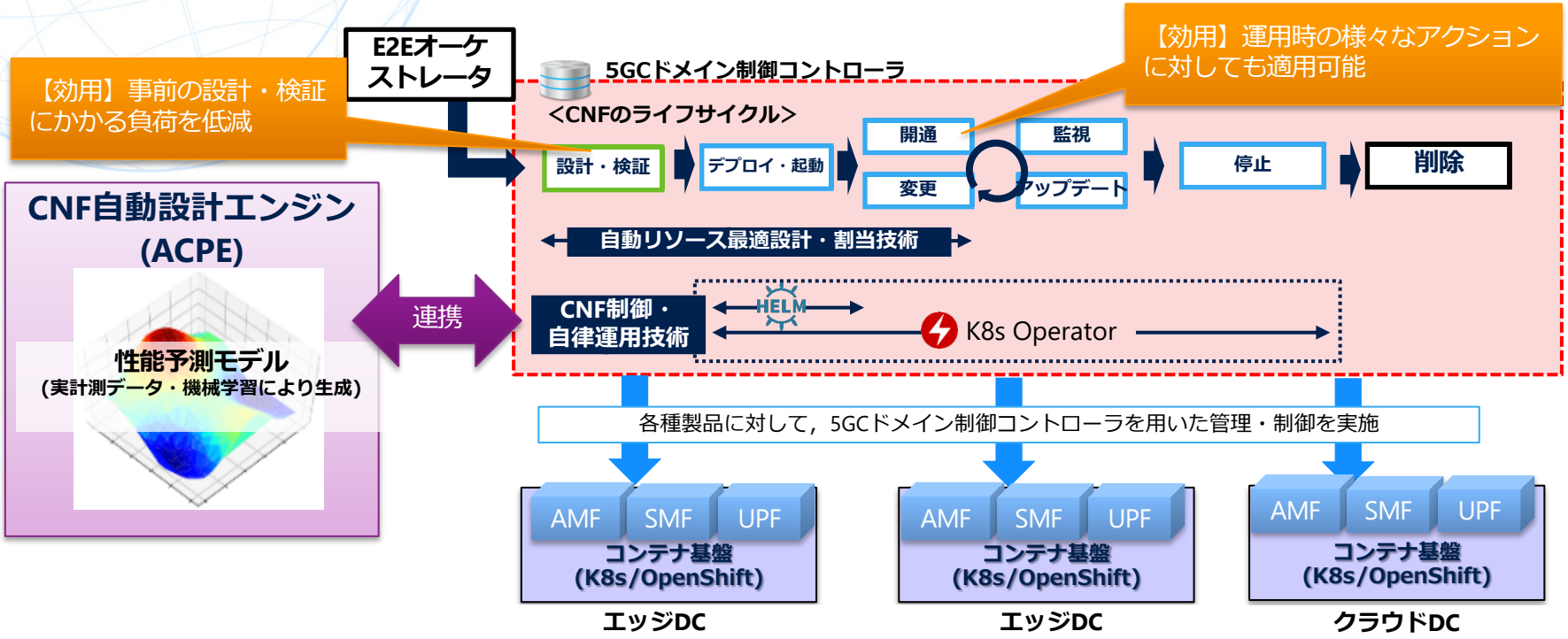
ACPEとK8s Operatorの連携によるCNF自律運用制御

NTT

- CNFのライフサイクル管理では，コンテナ基盤上でCNFを自律制御するコントローラ相当の技術が必要

→ Kubernetes API・制御ロジックを独自に拡張する**K8s Operator**を活用，ACPEとの連携機能を実装

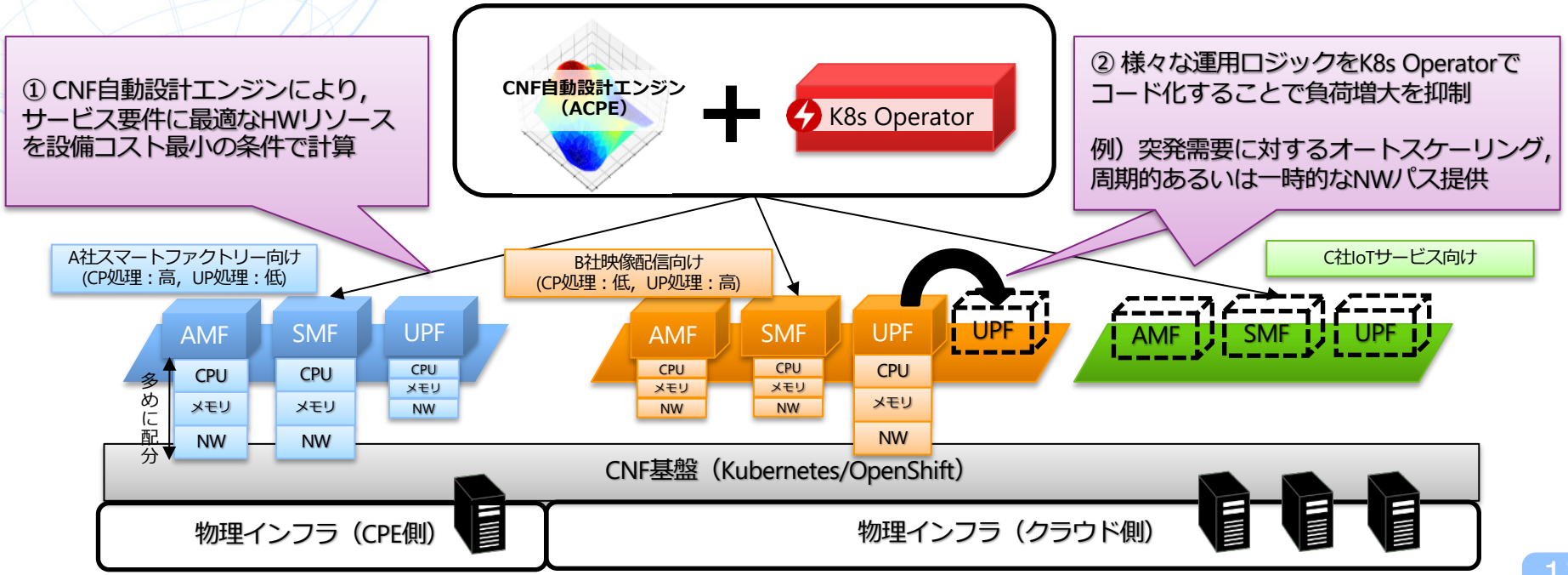
- **CNF自動設計エンジンによる最適設計とK8s Operatorによる自律制御を組合せる**ことにより，NW設計・構築・運用にかかる時間・人手負荷を低減



ローカル5G事業者へのNWソリューション提供への活用例

① CNF自動設計エンジンの期待される効果
- 事前の設計検証なくサービス要件に最適なHWリソースを，設備コスト最小の条件で割当可能

② K8s Operatorとの組合せで期待される効果
- 様々な状況変化によって発生するCNFオペレーション稼働を，K8s Operatorでコード化し自律動作させることで運用負荷増大を抑制



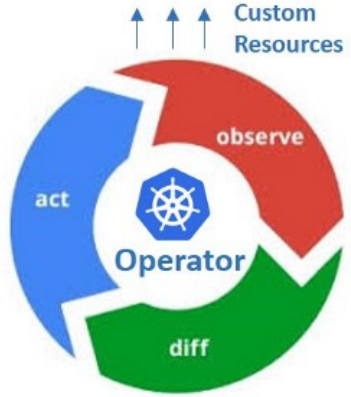
(参考) Kubernetes Operatorとは

NTT

- ステートフル・アプリや管理が複雑なアプリに対して，**そのアプリ特有の運用ノウハウをコード化し，Kubernetes 上でのライフサイクル管理を可能にする技術**
- 具体的には，K8s APIを拡張し(**“Custom Resource Definition”**)，独自のロジックを備えたコントローラ(**“Custom Controller”**)を実装
- K8s APIの拡張には内部の複雑な構造の理解が必要だが，OSSの**Operator Framework (Operator SDK)**を利用することで比較的容易に実装可能

K8s Operator = CRD (Custom Resource Definition) + Controller (Custom Controller)

- **Custom Resource Definition**: Kubernetes APIを拡張
- **Custom Controller**: 独自ロジックを備えたコントローラ



Day2
Management tasks for stateful / complex workloads

K8sマニフェスト(YAML)

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: example
```

独自のK8sリソースが使用可能に！

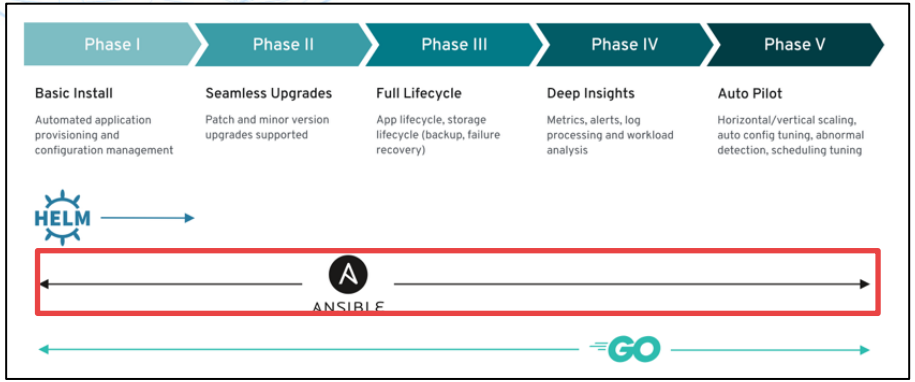
```
apiVersion: deploy.example.com/v1alpha1
kind: MyApp
metadata:
  name: example-myapp
```

(出典) Kubernetes Operators and Helm — It takes Two to Tango
https://cloudark.medium.com/kubernetes-operators-and-helm-it-takes-two-to-tango-3ff6dcf65619

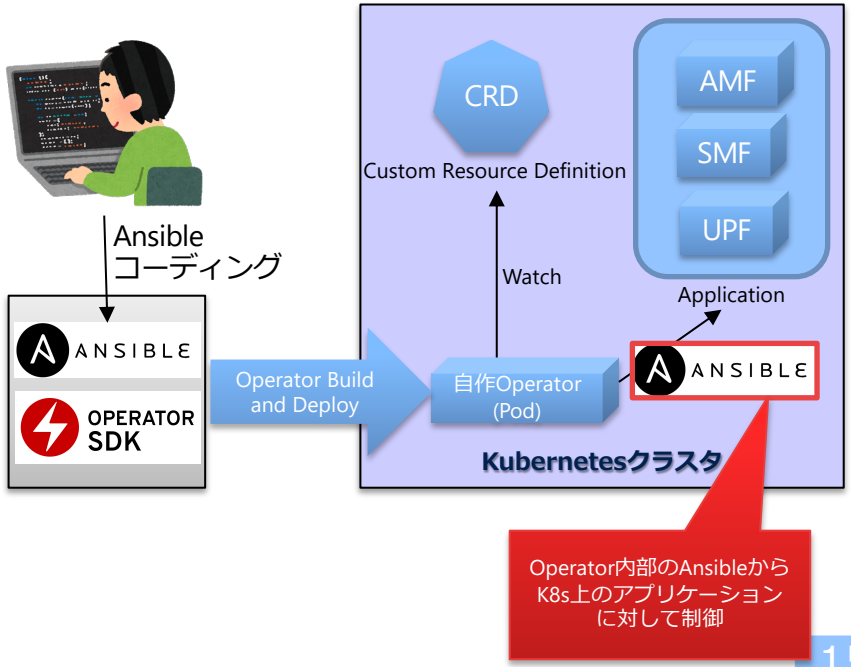
(参考) AnsibleベースのOperator

- Operator SDKによる開発パターンの内，今回は**AnsibleベースのOperator開発**を選択
- Ansibleの様々なモジュールを駆使し，通常のK8s機能を利用するだけでは実現できない制御を実装可能
  例）外部システムのステート取得，REST API実行，実行条件分岐・制御等

Operator SDKの開発パターン

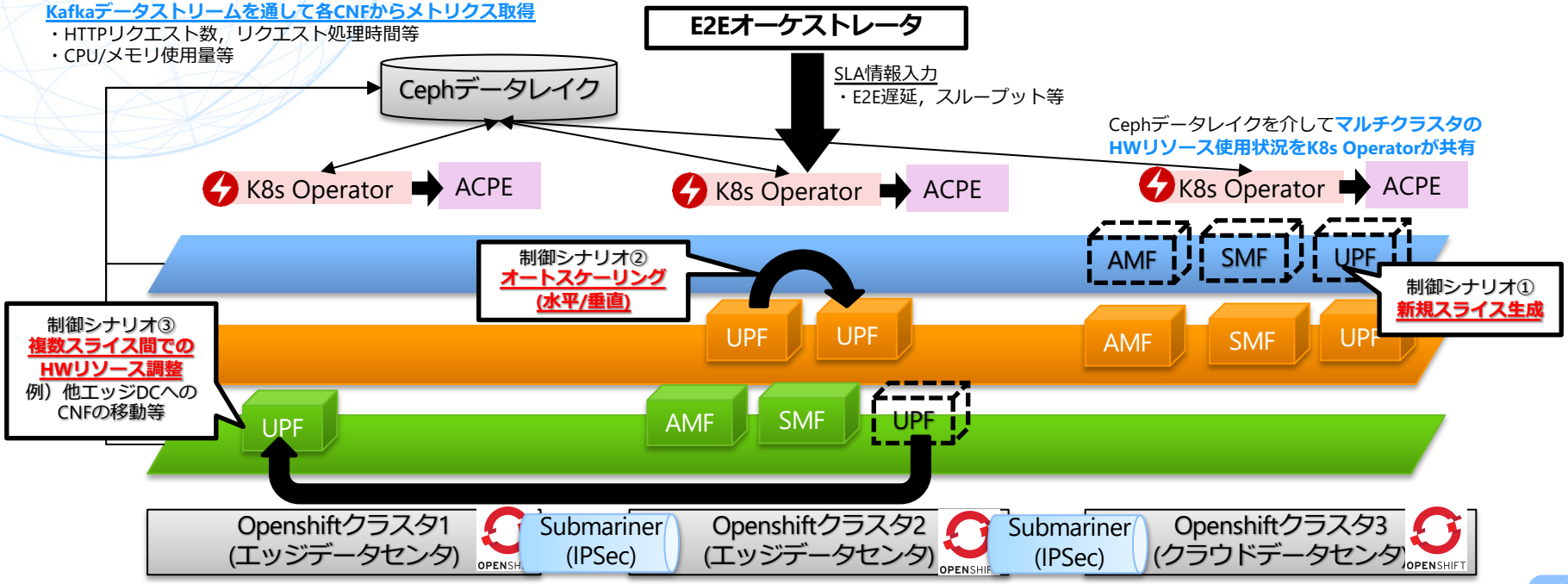


(出典) Operator 4.2 | Red Hat Customer Portal
https://access.redhat.com/documentation/ja-jp/openshift_container_platform/4.2/html-single/operators/index



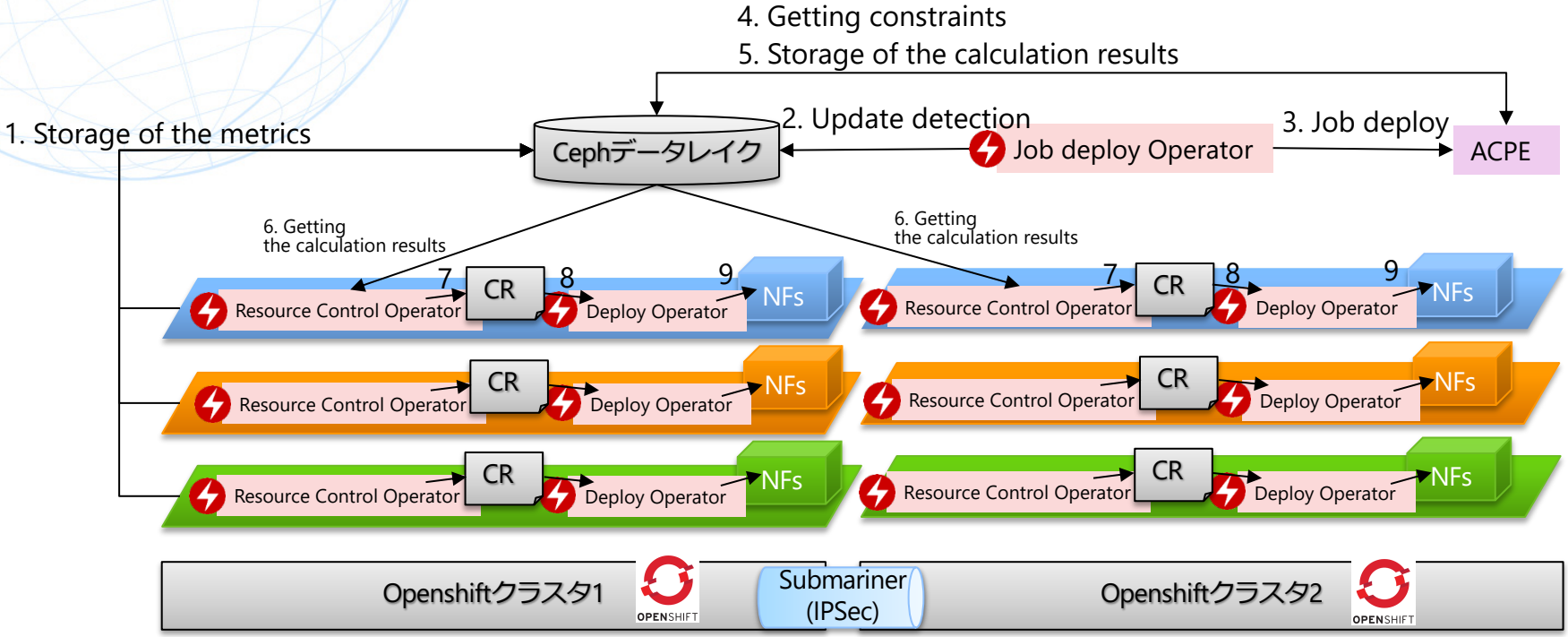
5GC制御全体アーキテクチャと制御シナリオ

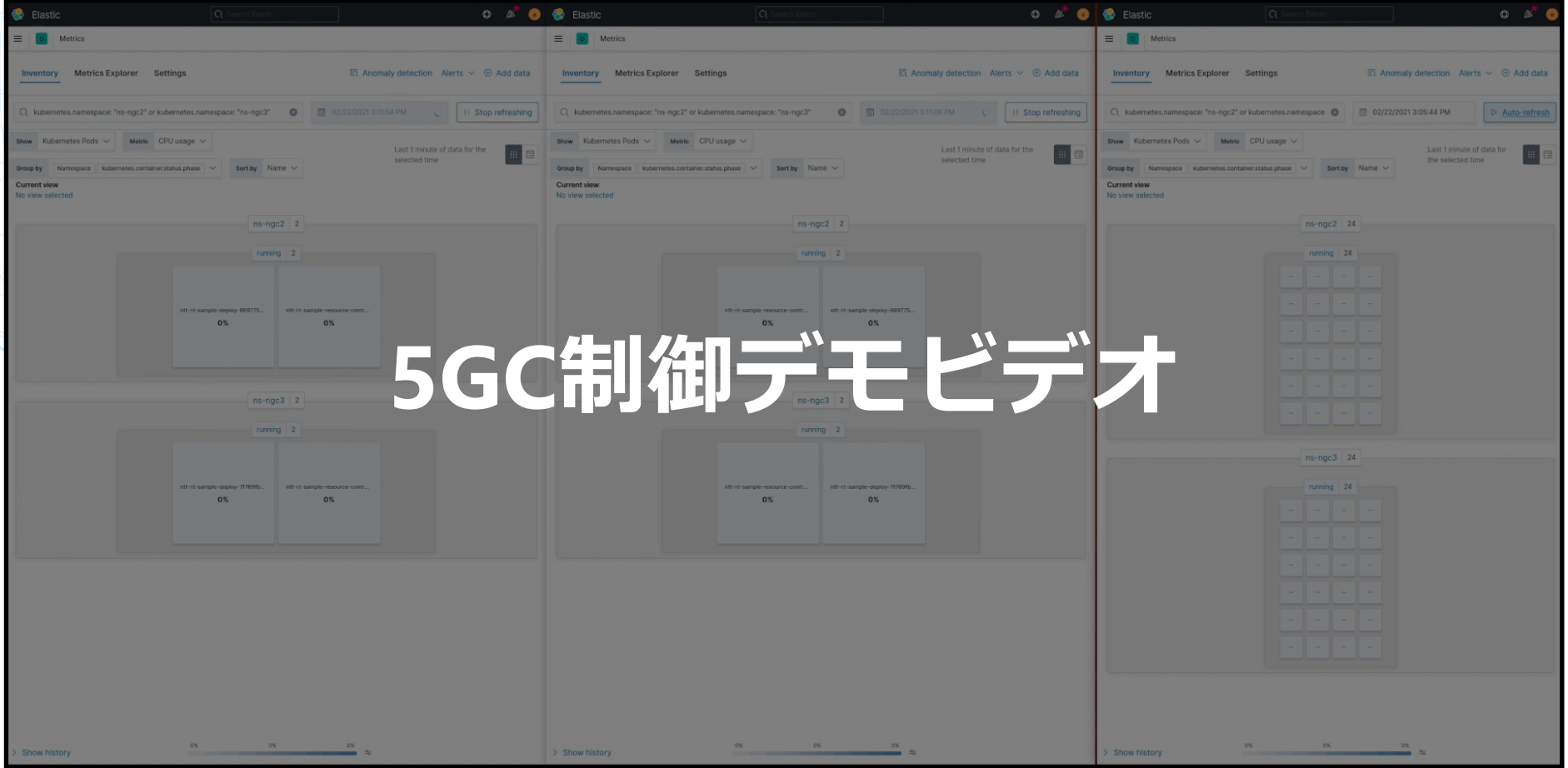
- 5GC制御全体アーキテクチャと制御シナリオを策定，これをベースにK8s Operatorを実装
  - **マルチクラスタでの5GC制御**
  - **ACPE構築時の機械学習環境を考慮したCeph＋Kafkaの利用(CephがOperator制御の“Single truth of source”に)**



現在のNT研実装状況と5GC制御デモ

- 現在の実装では，**複数種類のOperatorによる連携制御**を実施
- 5GC制御デモは現時点では，以下シナリオで動作可能
  - **シナリオ1：新規スライス生成**
  - **シナリオ2：運用時のスライスのスケーリング（現状はスライス要件の変化のみに対応）**
  - **シナリオ3：複数スライス間のHWリソース調整（今回はクラスタ間のNFの移動をご紹介）**




Elastic
Metrics
Inventory
Metrics Explorer
Settings
Anomaly detection
Alerts
Add data
Stop refreshing
Auto-refresh
Show
Kubernetes Pods
Metric
CPU usage
Group by
Namespace
kubernetes.container.status.phase
Sort by
Name
Current view
No view selected
ns-ngc2
ns-ngc3
running
Show history

5GC制御デモビデオ

今後の検討について

NTT

- CNF自動設計エンジン(ACPE)開発
  - **性能予測モデルのUプレーン機能(UPF)への適用**
  - **UPF・vRAN(DU)等で想定されるHWアクセラレーション(SR_IOV，GPUオフロード等)への対応**

- K8s Operator開発
  - **メトリクスの変化をトリガーとした本来あるべきオートスケーリングへの対応**
    （→K8s/Openshift側のアラート機能をトリガーに監視対象のPodを制御する実装を想定）
  - **5GCアプリケーション・サービスレベルまで考慮した制御**
    例）クラスタ間でPodを移動させる際のサービス断回避方式，またその際のSDNコントローラとの連携，
    移動前後でのセッション情報等のステート管理等

- 情報共有・ディスカッション・勉強したいこと
- マルチK8sクラスタ管理・コネクティビティ関連技術について
（GitOps，Kubefed，Openshift ACM，Submarinner，Istio，Skupper等）
- モバイル関連OSSについて（free5GC，OAI，Open5GS，magma等）
- K8s上での機械学習パイプライン構築について（Kubeflow等）