

5Gコアへのテナ活用

～5GCとクラウドネイティブ思想が修羅場すぎる～

NTTドコモ
小原 啓希 谷川 尚也

2025/08/01

はじめに
モバイルテレコム 歴史の振り返り
モバイル通信のトラフィック変化と コアNWの変遷

本日のテーマ

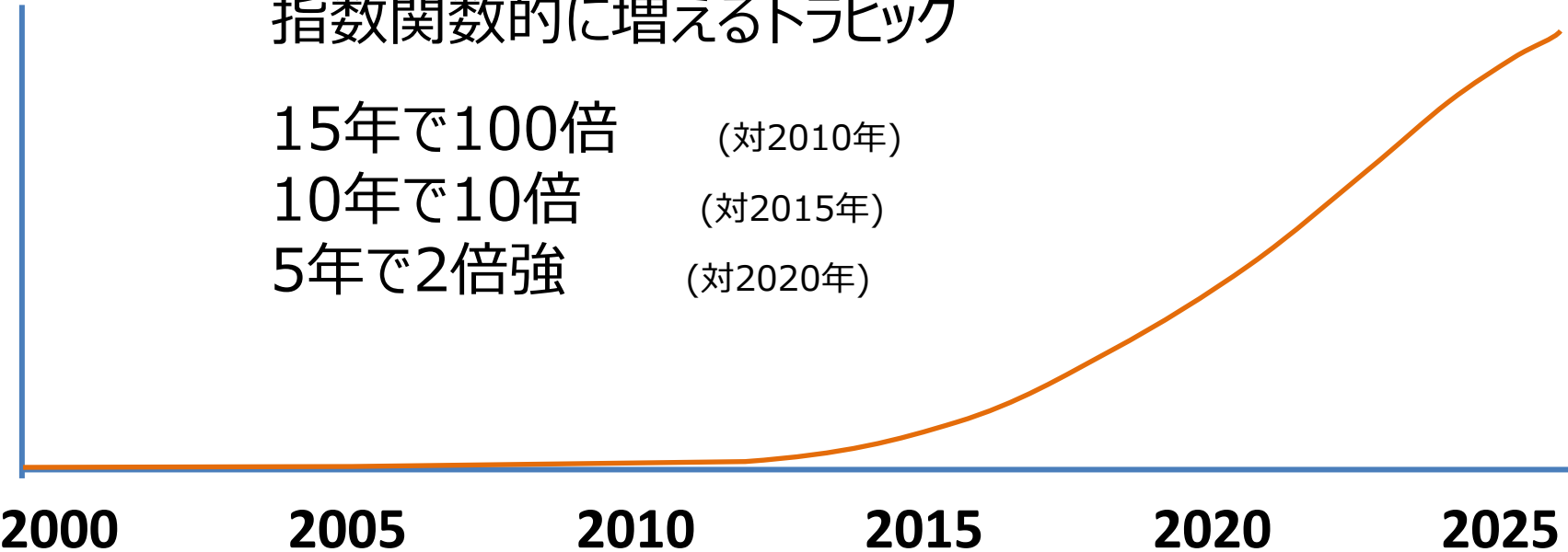
はじめに

モバイル通信量の変化 と通信世代

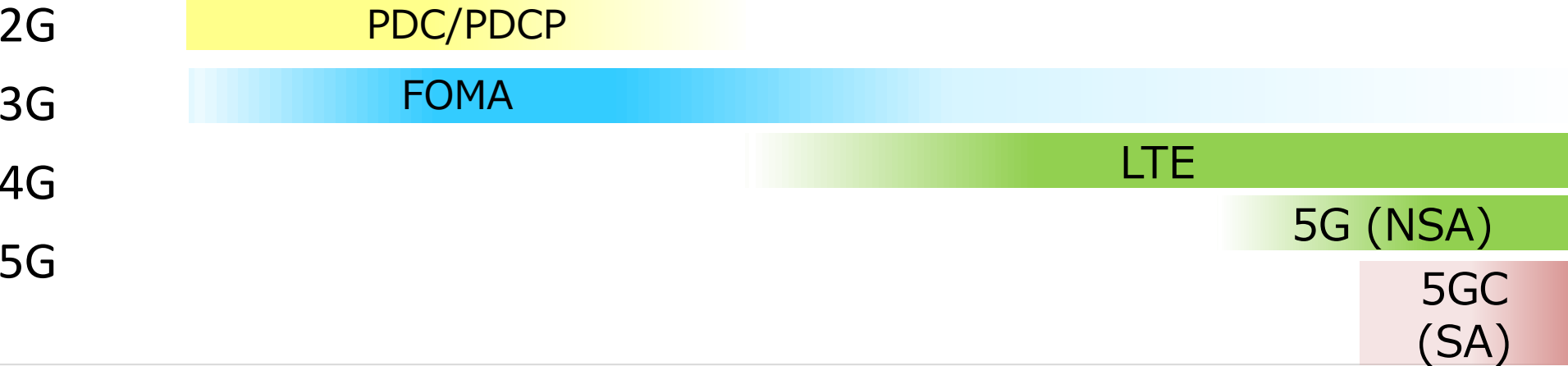
データ量

指数関数的に増えるトラヒック

15年で100倍 (対2010年)
10年で10倍 (対2015年)
5年で2倍強 (対2020年)



通信世代



世代ごとの コアNW装置 の変遷サマリ①

		初期	中期	後期
3G FOMA	装置名	MMS	xGSN	NxGSN
	OS	専用ソフト	CG-linux	
	ハード	専用ハード	Advanced TCA	
	初版	2000年	2003年	2011年
	主な端末	フィーチャーホン		スマートホン
	トラフィック増加に対する 主なアプローチ	装置内プロセッサの 物理的な交換による スケールアップ 2009年ごろから 装置数スケールアウトも併用		
4G LTE	装置名	ESPGW/MME/PCRF	ESPGW2/MME2/PCRF	vSPGW/vMME/vPCRF
	OS	CG-linux		
	ハード	Advanced TCA		仮想化基盤
	初版	2010年	2013年	2016年
	主な端末	スマートホン		
	トラフィック増加に対する 主なアプローチ	装置数 スケールアウト 装置更改による スケールアップ		装置数 スケールアウト 装置内VM※スケールアウト

世代ごとの コアNW装置 の変遷サマリ②

		現行世代
5 G NSA	装置名	vSPGW/vMME/vPCRF/UPF
	OS	CG-linux
	ハード	仮想化基盤
	初版	2016年 / 2021年
	主な端末	スマートフォン
	トラフィック増加に対する 主なアプローチ	装置内VM スケールアウト 装置数 スケールアウト

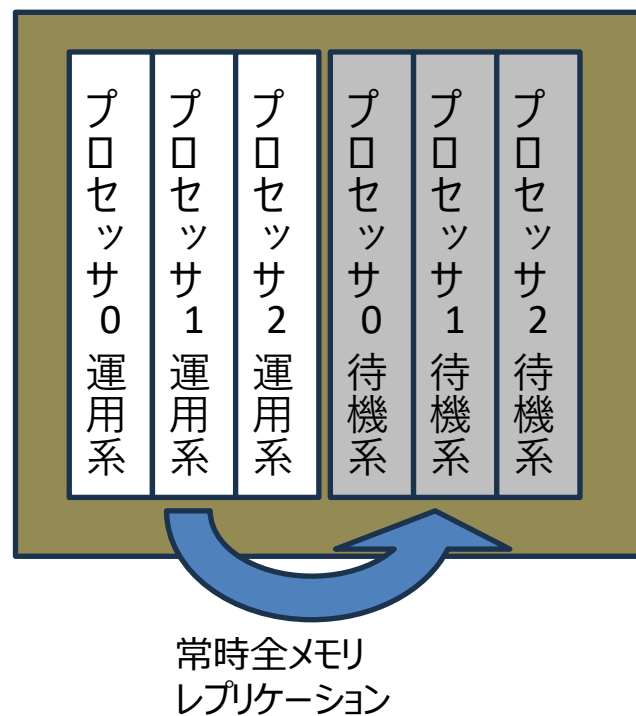
5GC SA	装置名	AMF/SMSF/NSSF/SMF/PCF/CHF/NRF/BSF/UPF
	OS	トピック④ Core OS (Openshift) (CaaS)
	ハード	仮想化基盤
	初版	2021年
	主な端末	スマートフォン
	トラフィック増加に対する 主なアプローチ	装置内pod スケールアウト 装置数 スケールアウト (結局 コンテナになっても 装置単位のスケールアウトも実施)

トピック① 変化するハード構成と 変化が少なかった運用オペレーション

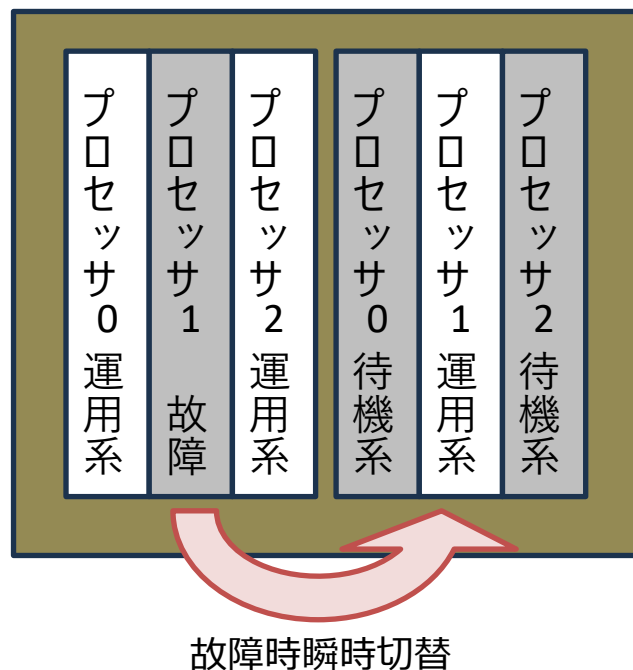
コアNW装置は、5G NSA 装置までの間の過去20年間、ハードウェアは進歩したものの、基本的なアーキテクチャに変化はなく、運用オペレーションに変化は少なかった。

ATCA ⇒ 仮想化時も 枯れた技術を流用し、ハードウェアをVirtual Machine に置き換え、思想変化は無し

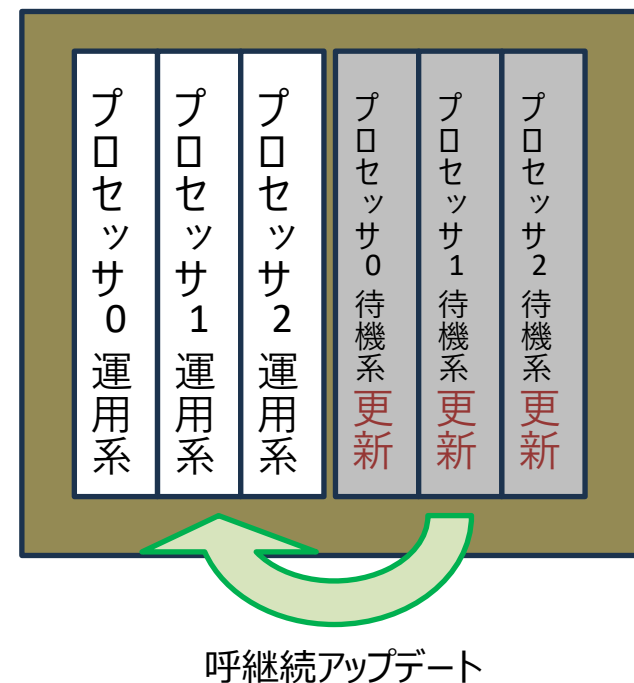
基本構成



装置故障時 (系切替)



ファイルアップデート時

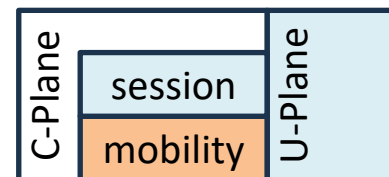


故障等の異常時や、ファイルアップデート時においても、できる限り呼処理を引き継ぐ **ステートフル**な思想

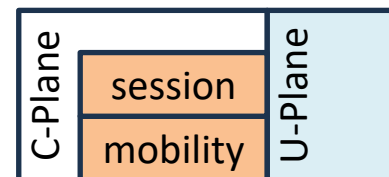
トピック② スマートホンの台頭によるNWへのインパクト

フィーチャーホンとスマートホンの NWから見た違い (詳細 次頁)

フィーチャーホン： 未使用時はmobilityのみ制御し、
使用時にsession/Uplaneを制御



スマートフォン： 未使用時にもmobility/sessionを制御し、
使用時にU-Plane を起動



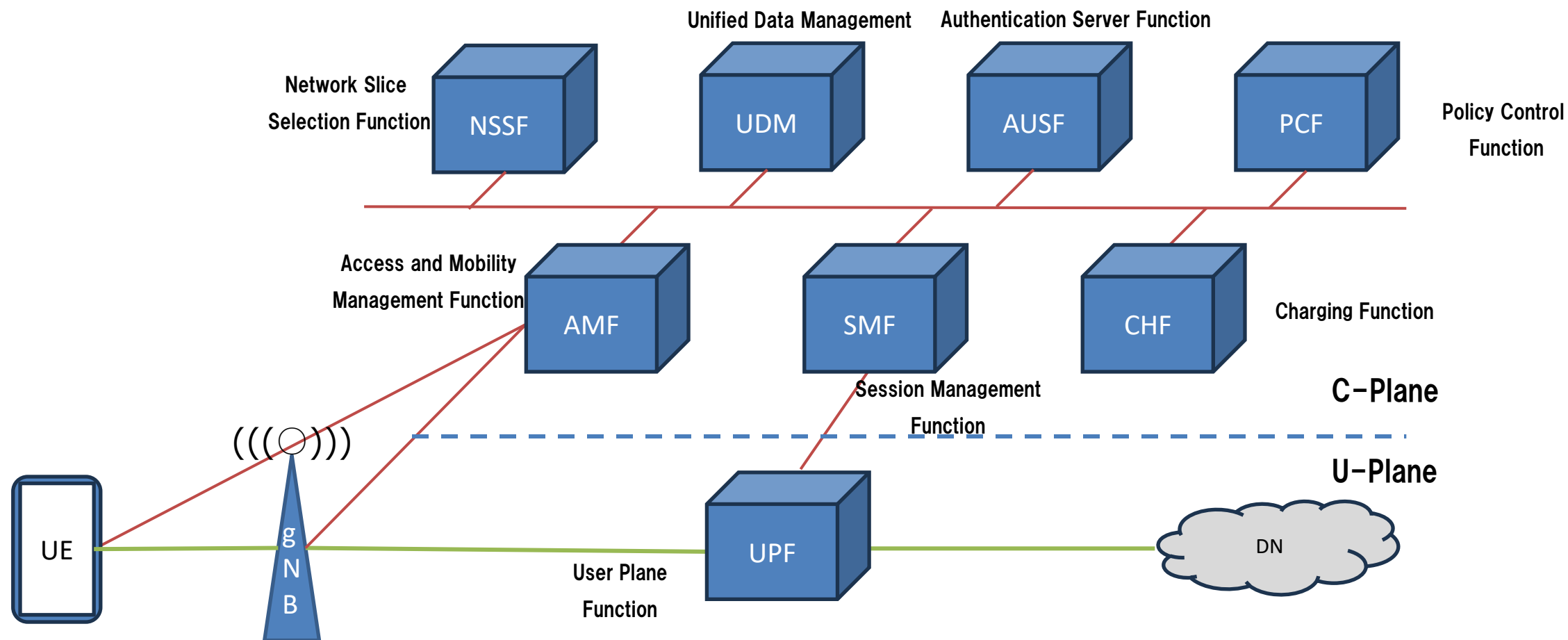
ステート管理 が 倍以上大変になった。

また、 端末は常に接続を維持し続けるため、不具合等の発生によりNWから切断された場合、切断された端末 (数十万台から数百万台)が 一斉(数分の間) に再接続を要求する。

最悪時 1万TPSを超過する 接続要求

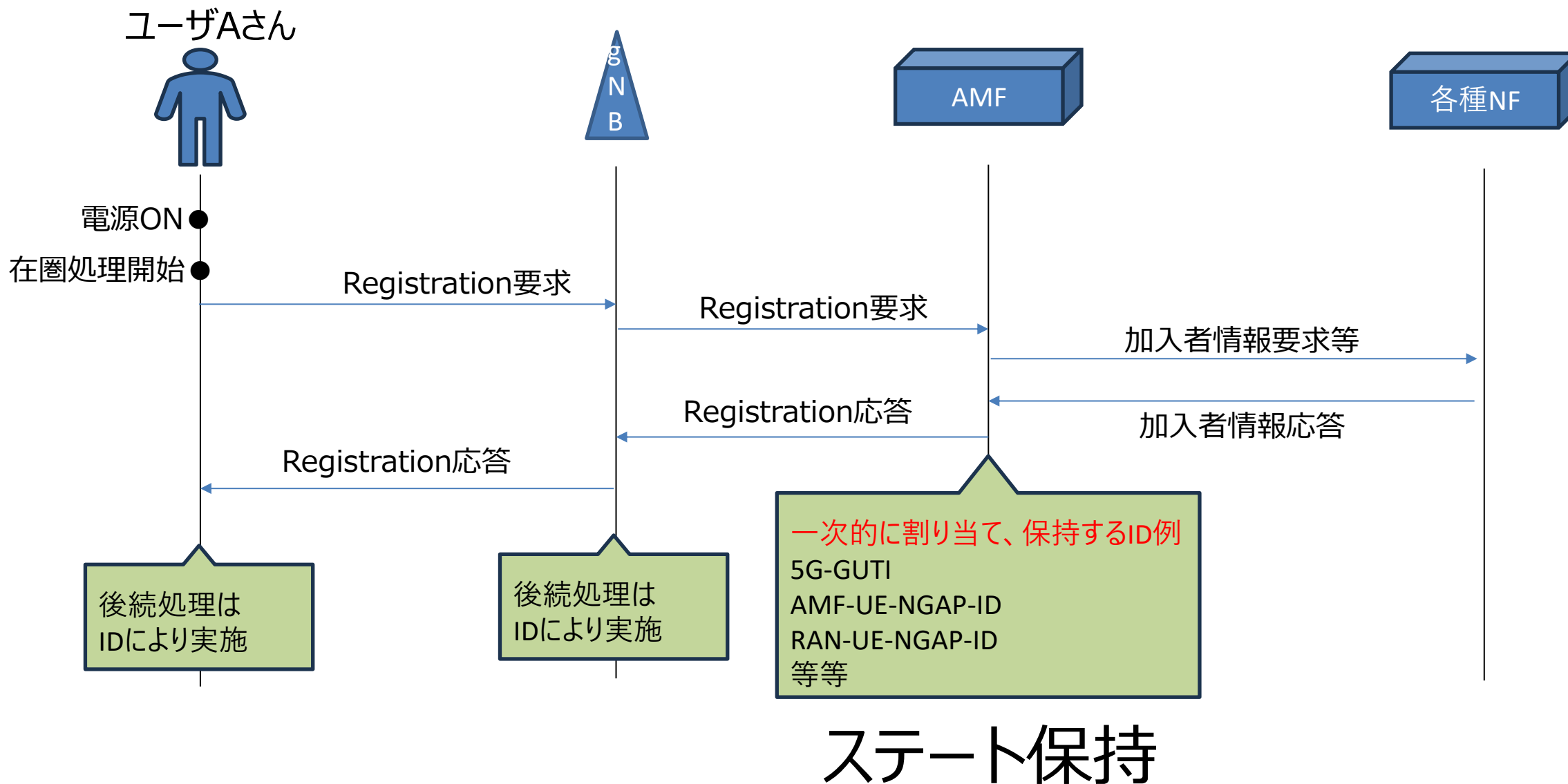
スマートホンの台頭以降から モバイル コアNWは 輻輳 との 戦い

5G-SAを構成するNFは以下の通り

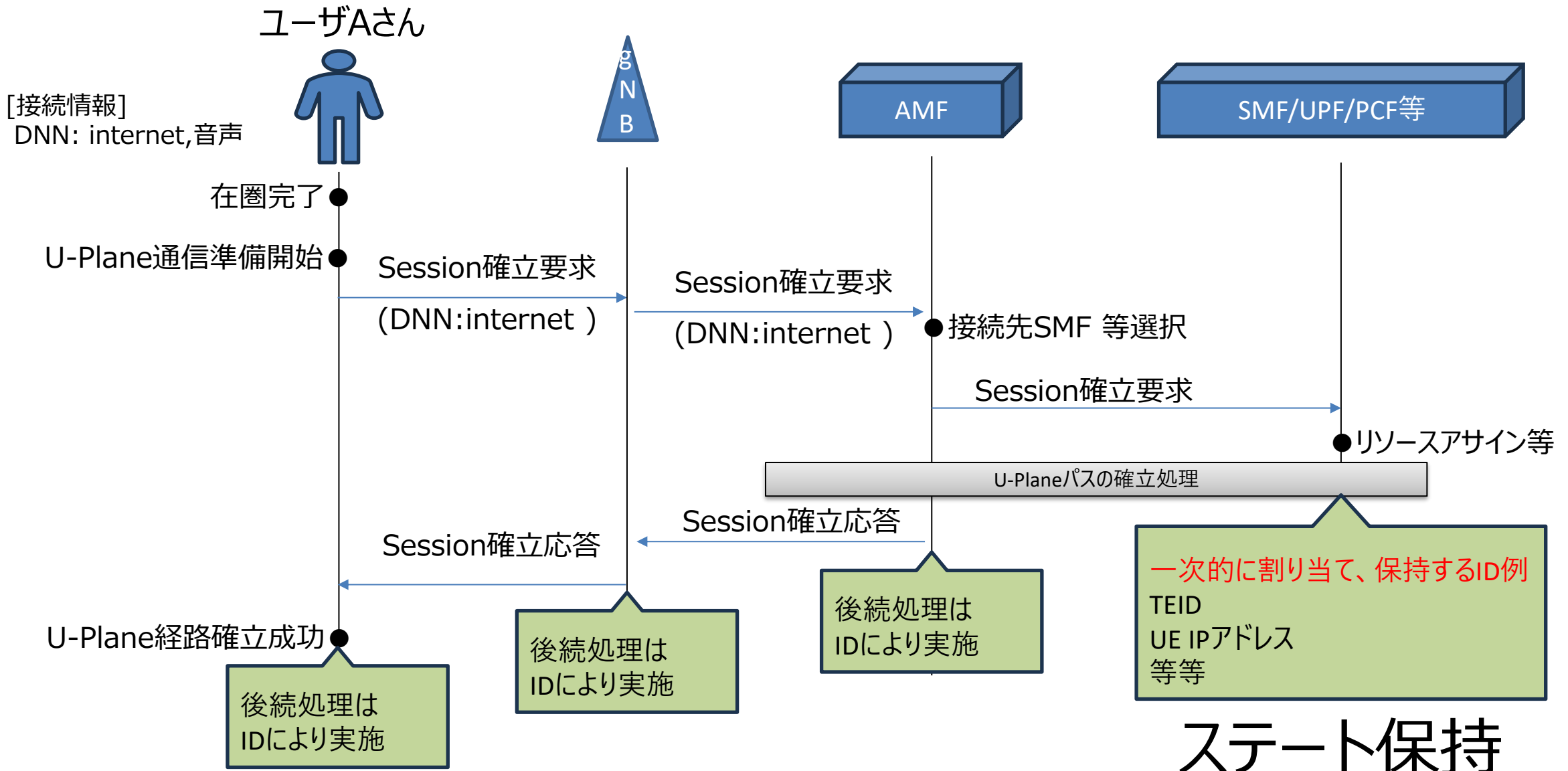


※図以外にも様々なNFが標準で規定されています。

トピック② 参考 接続処理のお話 (mobility)

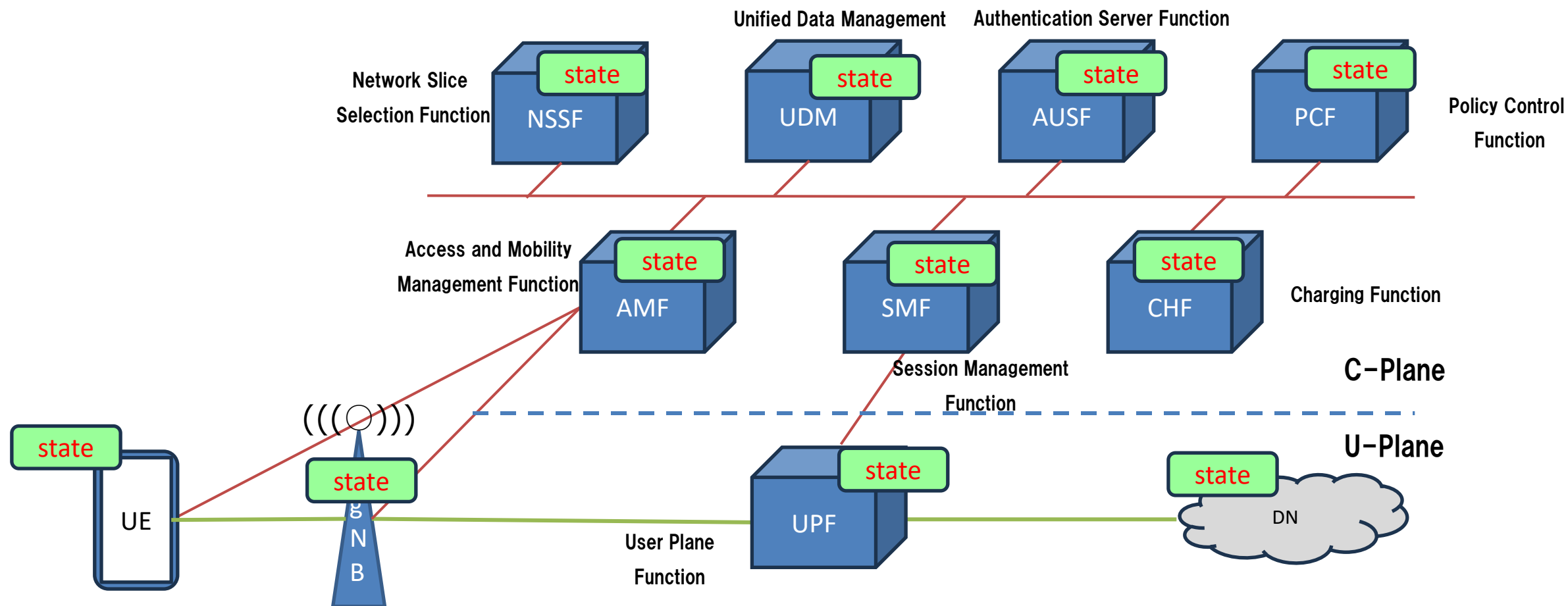


トピック② 参考 接続処理のお話 (session)



トピック② 参考 UE～ DNまで全体でstate状態を管理

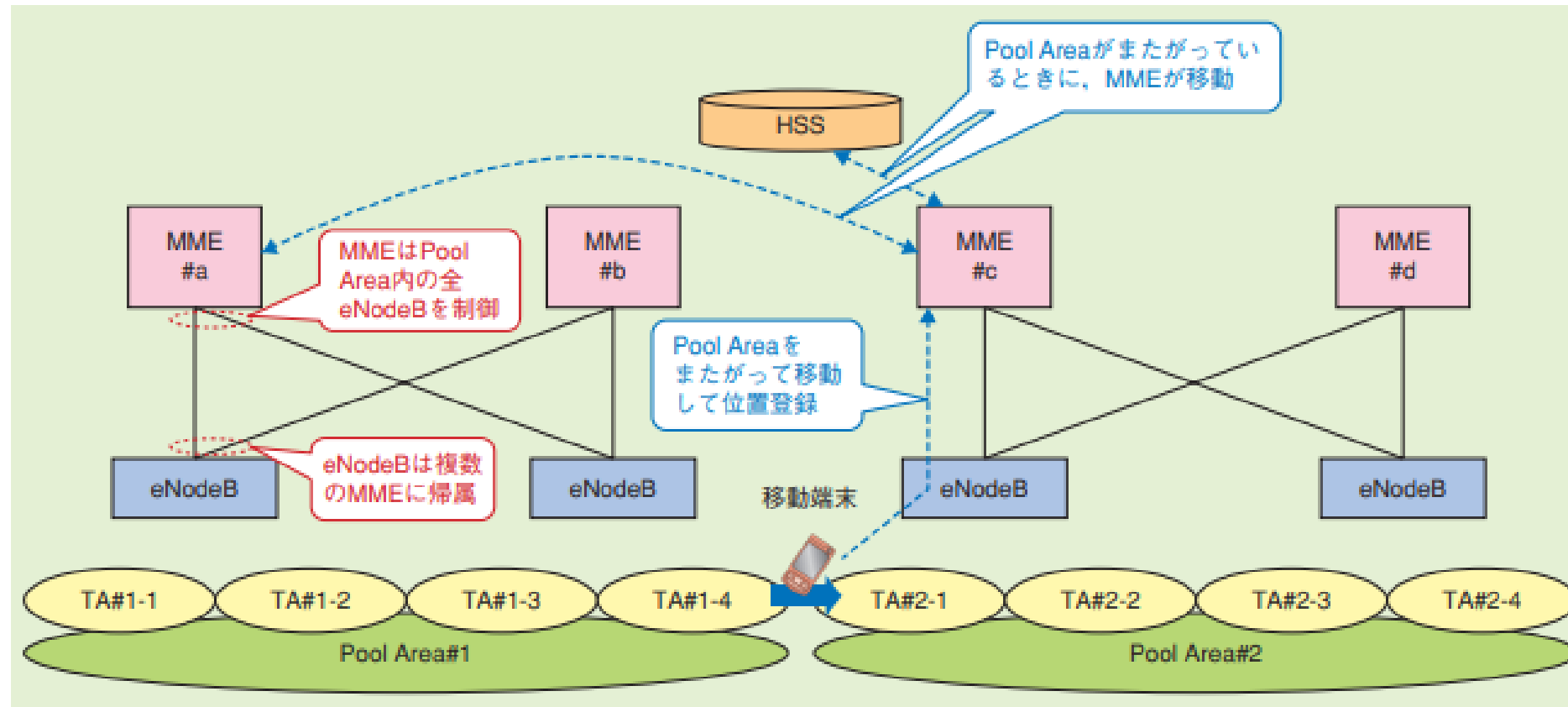
5G-SAを構成するNFは以下の通り



※図以外にも様々なNFが標準で規定されています。

トピック③ コア NW構成概要 pool構成 (LTEの場合)

モバイル網では、基地局 (eNode B) とコア網が flex (メッシュ構成)で接続されており、そのエリアを pool と呼称している。



Pool化によるメリット

交換機をまたがる 位置登録単位を大きくとることができる

ユニットの冗長化を取ることができる

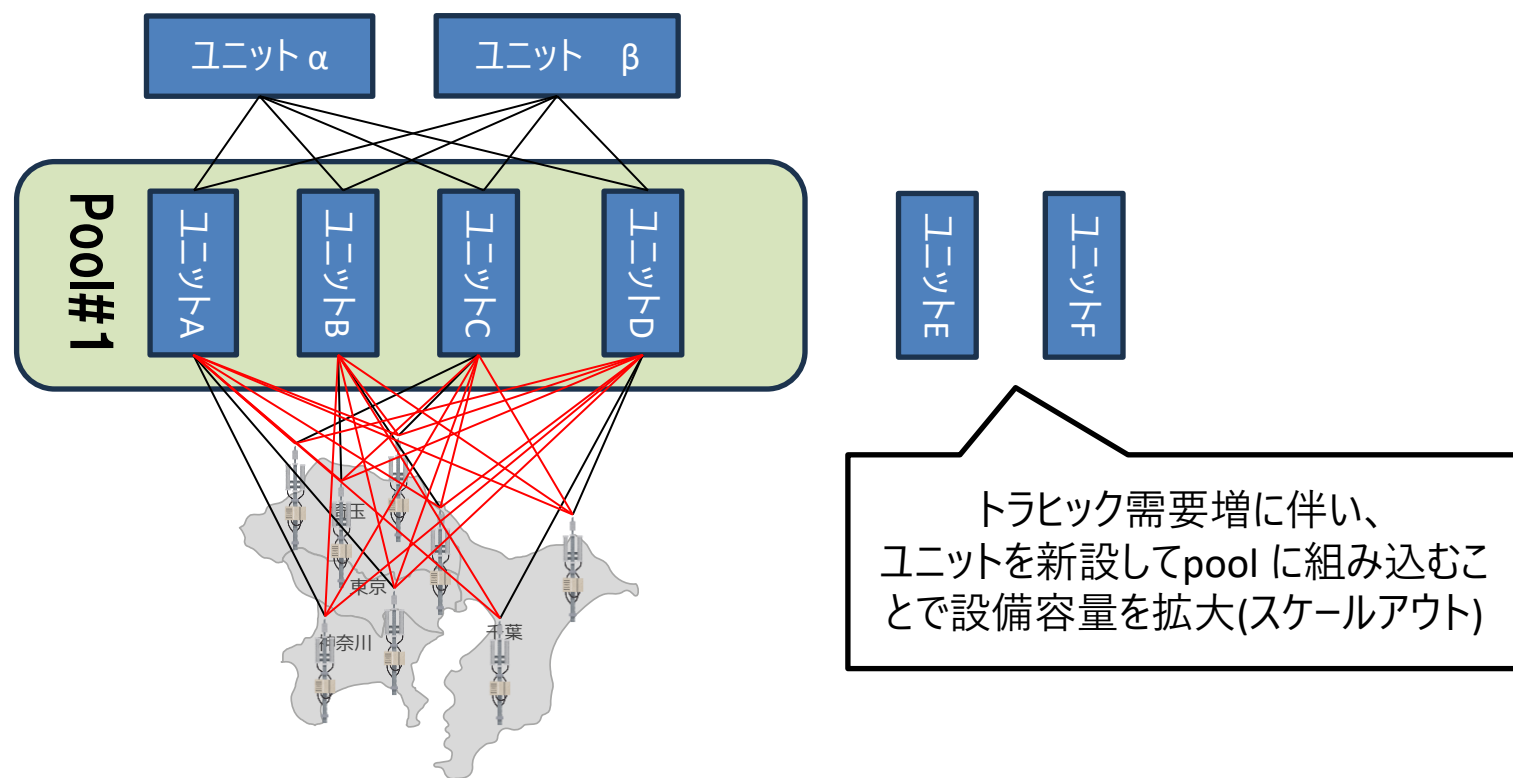
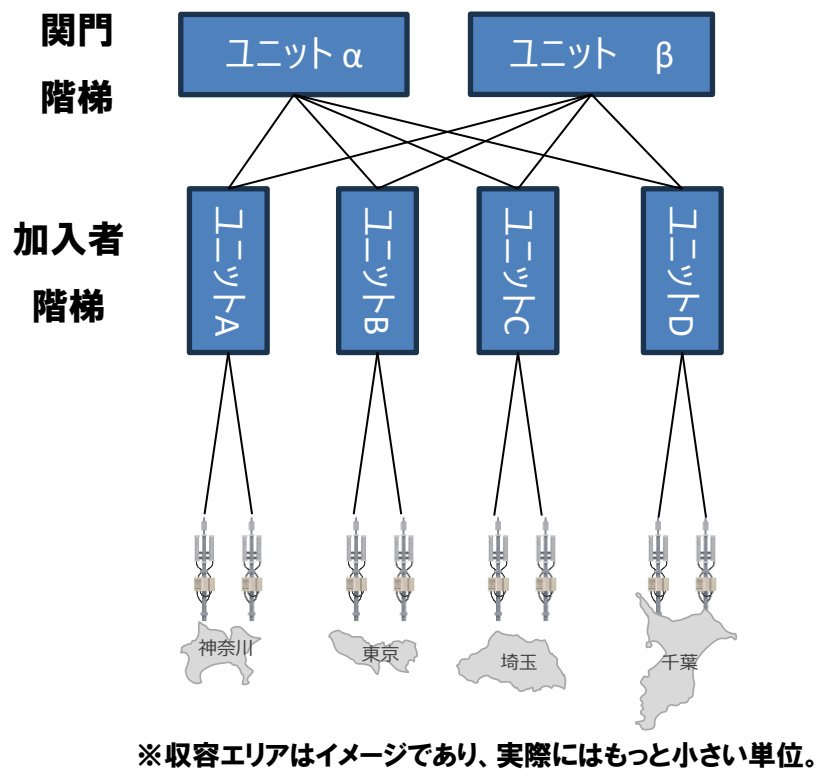
トピック③ pool構成の導入 と 装置数(ユニット単位)スケールアウト

2000年台 (3G網初期)

2010年以降 (3G網後期/4G網)

基地局 : 交換機 = N:1

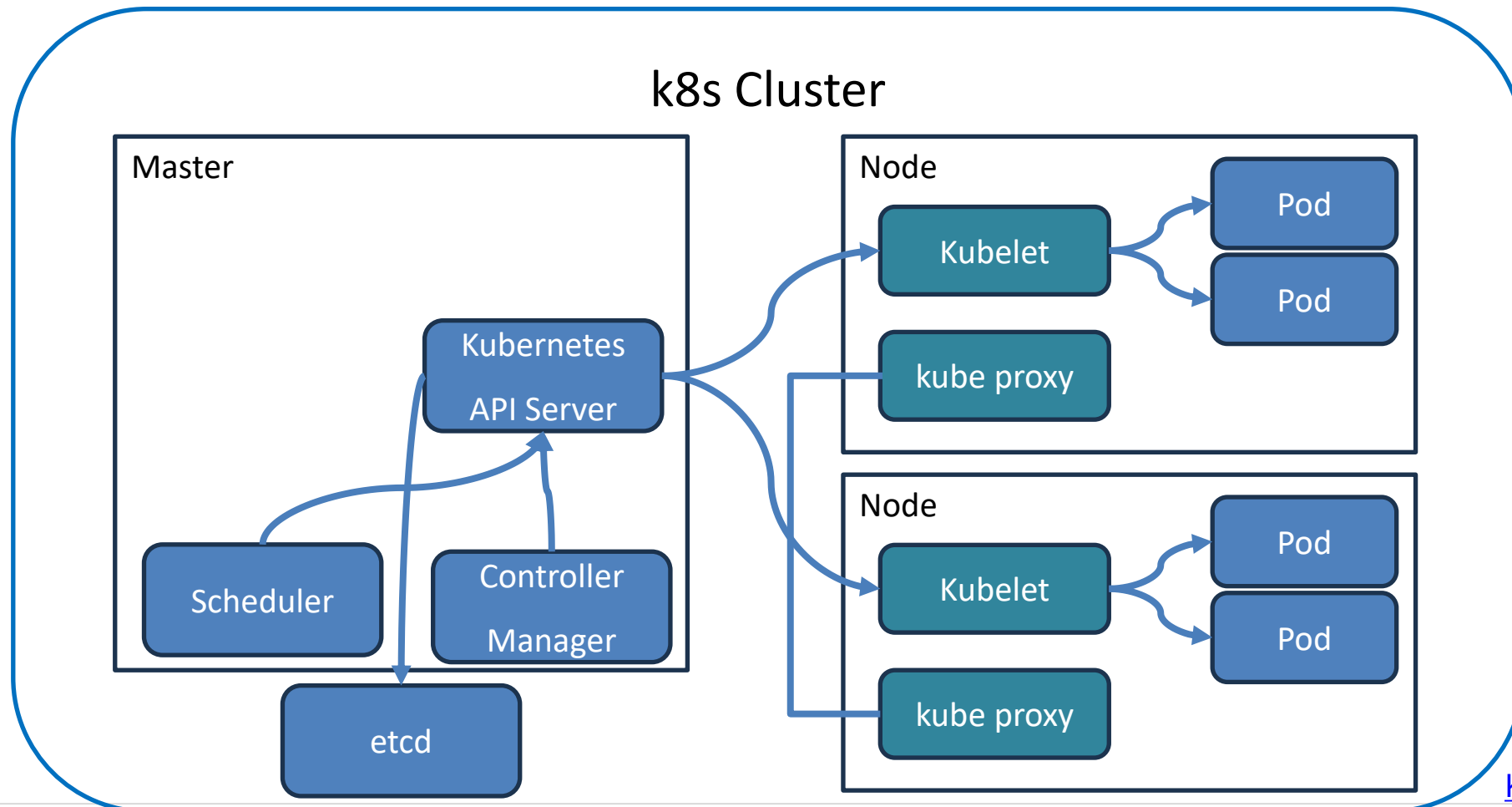
基地局 : 交換機 = N:M



トピック④ OpenShift 構成_そもそもk8s

コンテナ化されたサービスを管理するプラットフォーム

コンテナを束ねて管理する。Podと呼ばれるリソースにコンテナが含まれる



[Kubernetes.io](https://kubernetes.io) を参考

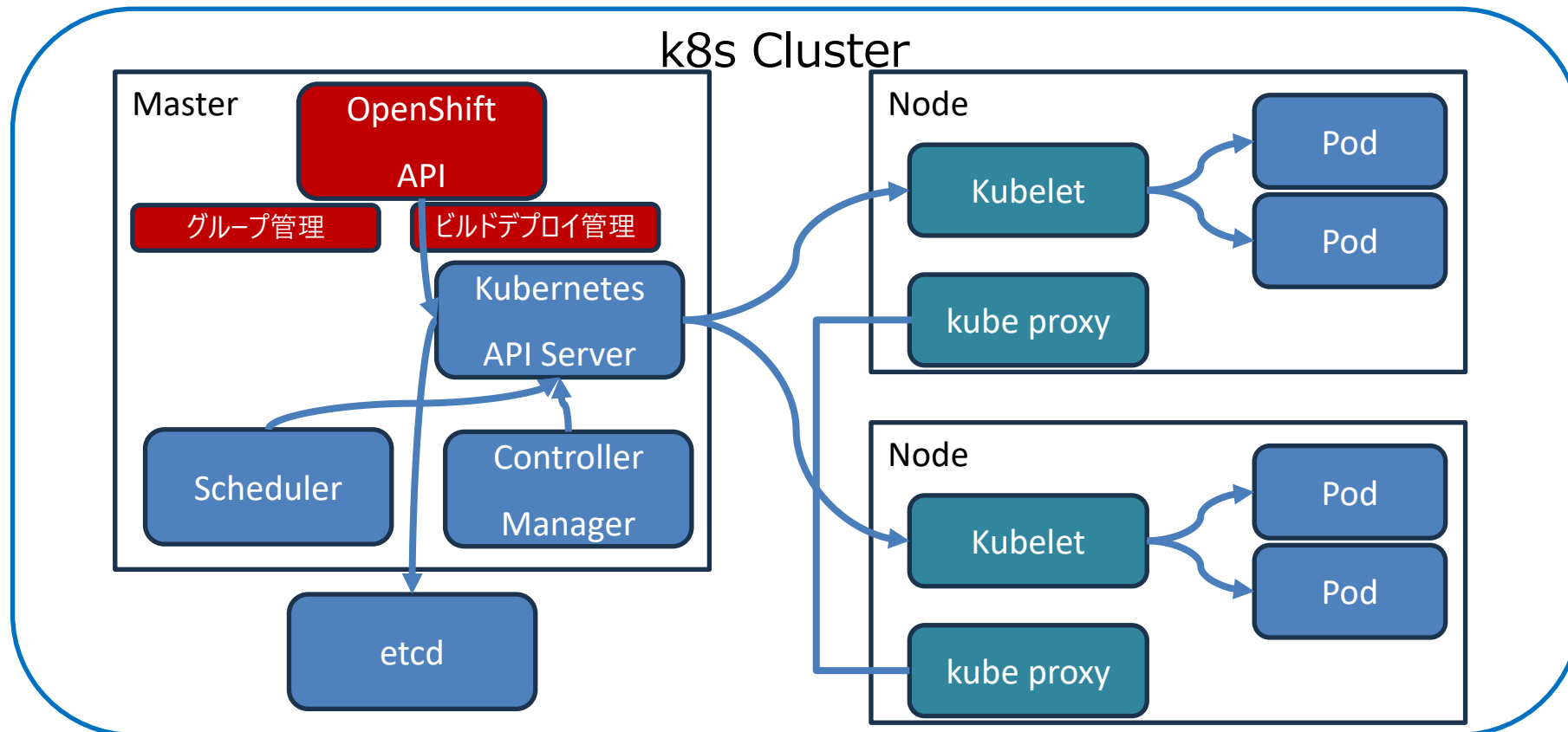
トピック④ Open Shift 構成_その上でOpen Shift

Kubernetesの最もメジャーな商用ディストリビューションの1つ

Kubernetes単体にはない機能が多数追加されている

CI/CDなどDevOps関連,ウェブコンソール,コンテナビルド機能,Ansibleによるインストーラ etc.

エンタープライズ版k8sとして利用するケースが多数



結論:ステートレス思想から始まったコンテナとステートフル思想なモバイルコア

コンテナでの運用だと

ワークロードの再起動時に元の値がすべて失われてしまう

コンテナは比較的もともと **ステートレス**

宣言型、ハイアベイラビリティ、自動スケーリング、クローン作成など

モバイルコア

U-Planeのパスの管理やQoSや課金などの管理も必要

基地局+UE規模を踏まえてNWがスケールする必要

VM内でDBと呼処理が一緒に担っていたもの(密結合)を各コンテナ(疎結合)にしたことで問題

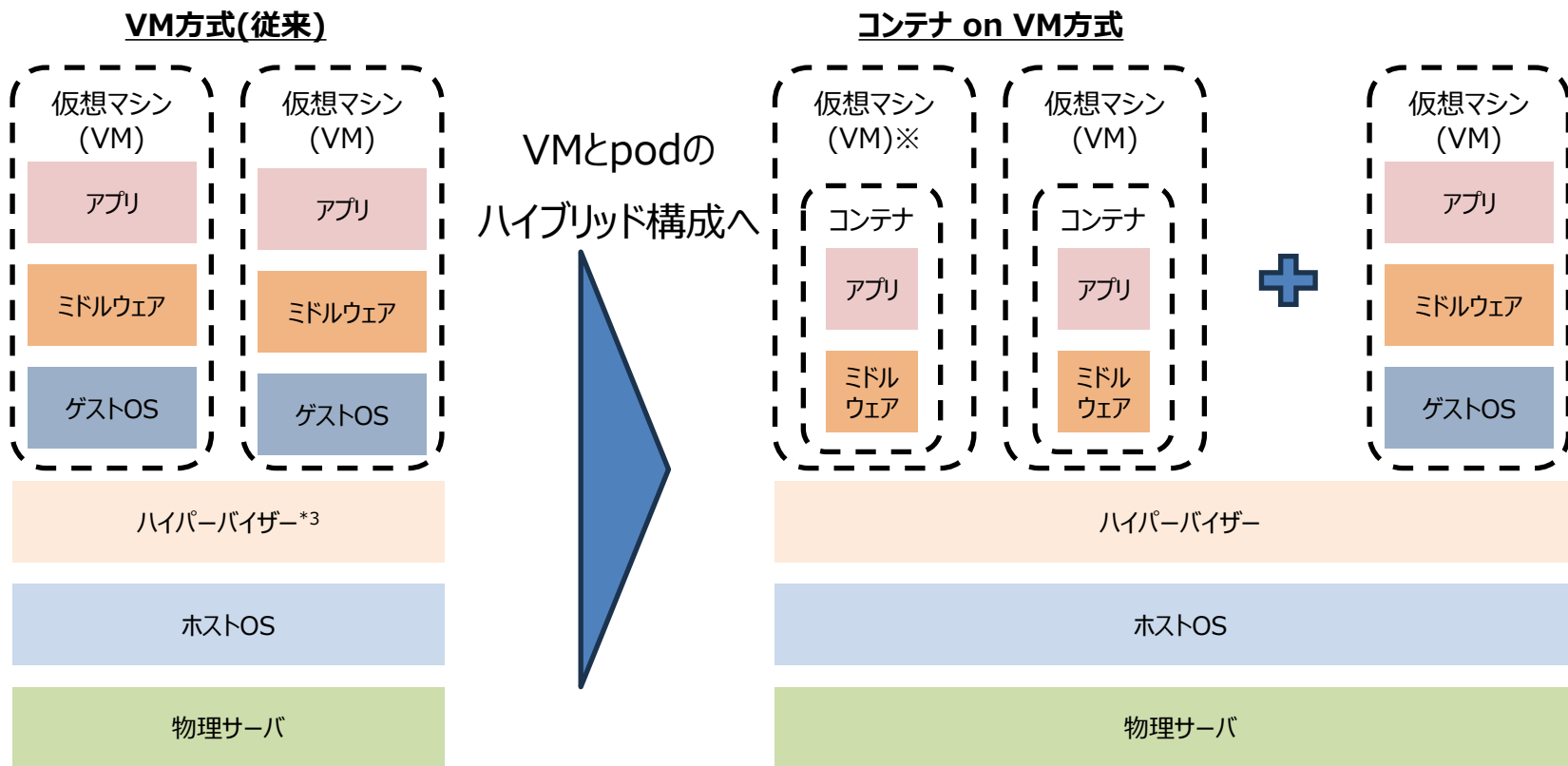
呼処理を引き継ぐ思想で中央集権的な**ステートレス**は難しい…

親和性が低い? !

現時点までの流れと障壁

5GC NF構成 導入初版 (20年度)

従来の技術である Virtual Machine と 新技術の pod のhybrid な構成によるいいところどりを目指そうとした



- 基盤やVM運用方法等の既存資産の活用
- マイクロサービスの設計思想を元に各機能をpodのアプリに分ける
- 実際は一部負荷の高くなるpodのアプリケーションをVMで運用してた…

Hybrid 構成による困りごと

Hybrid 構成による弊害

- VMとpod の機能ブロックごと 分割損の発生により 処理能力が伸びない。
- 運用上 更新対象が、 pod とVMで別れてしまい、 作業回数が倍増。
 - あわせてオペレーション装置の流用はできなかったため、作業手法が てんでばらばらにとっ散らかった。

⇒既存の思想 機能から脱却し 23年度に 全てpod 構成（フルコンテナ化）を実施を決定
主に

VMのファイル更新やMANO連携の作業をなくしてファイル更新作業の簡素化.

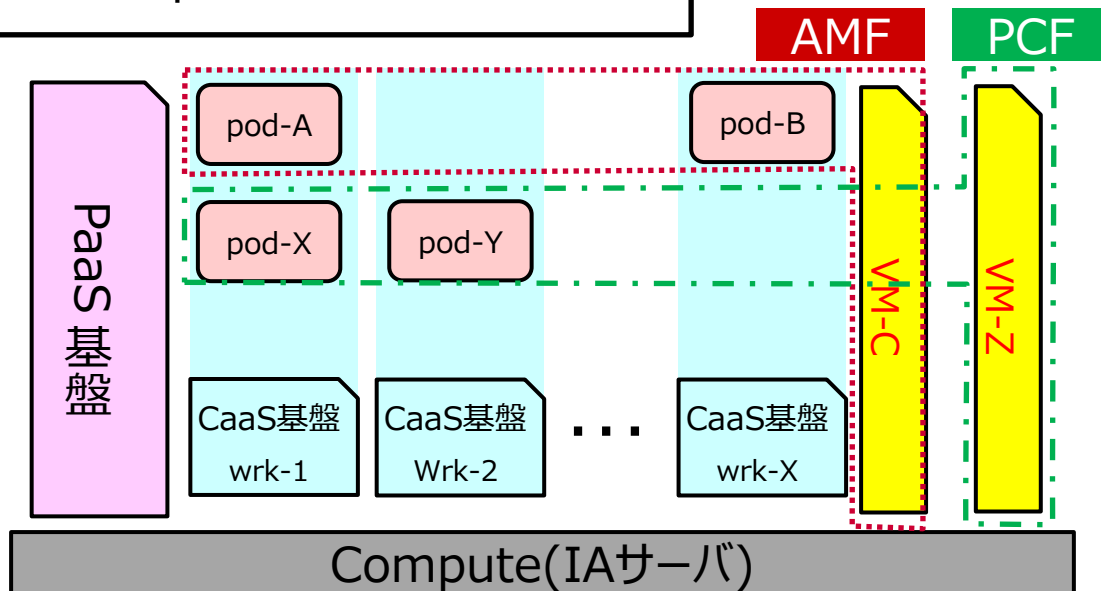
VM配備時の分割損解消.APL配備をフレキシブルにすることで空きリソースの最小化.収容効率改善

フルコンテナ化概要

NF のVM 部分をすべて pod化することで、収容効率の改善並びに、運用稼働削減を目指した。

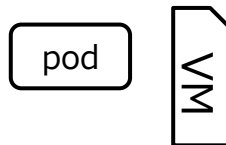
Before(22年度まで)

VM + pod のハイブリッド構成

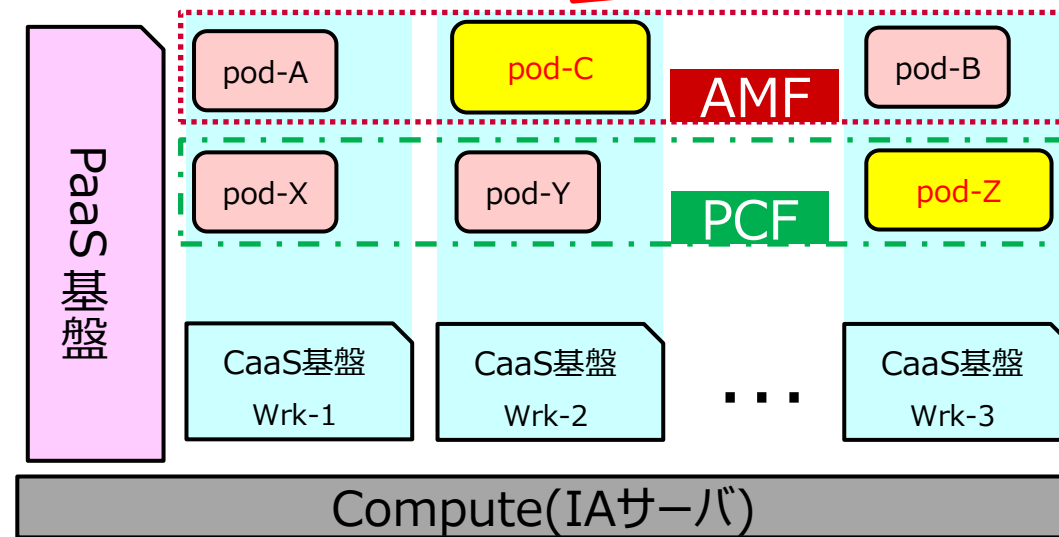


After(23年度以降)

凡例 :



VM → podに変更 (フルコンテナ化)



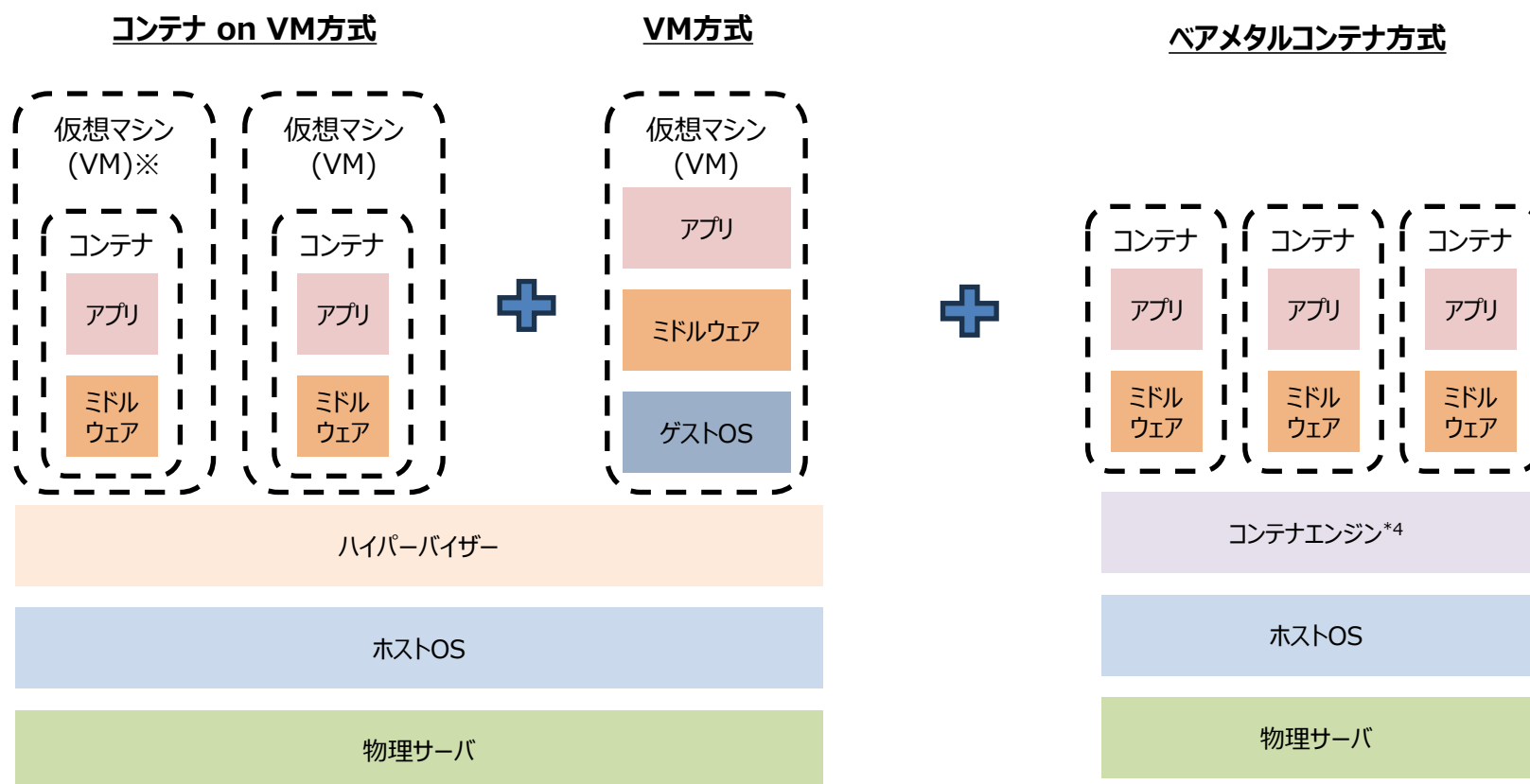
フルコンテナ化以降の状況

一方で 結局 コンテナに親和性の高いNFとVM型仮想化が適したNFが混在し続けている状況

C-Plane→コンテナ、U-Plane→VM等

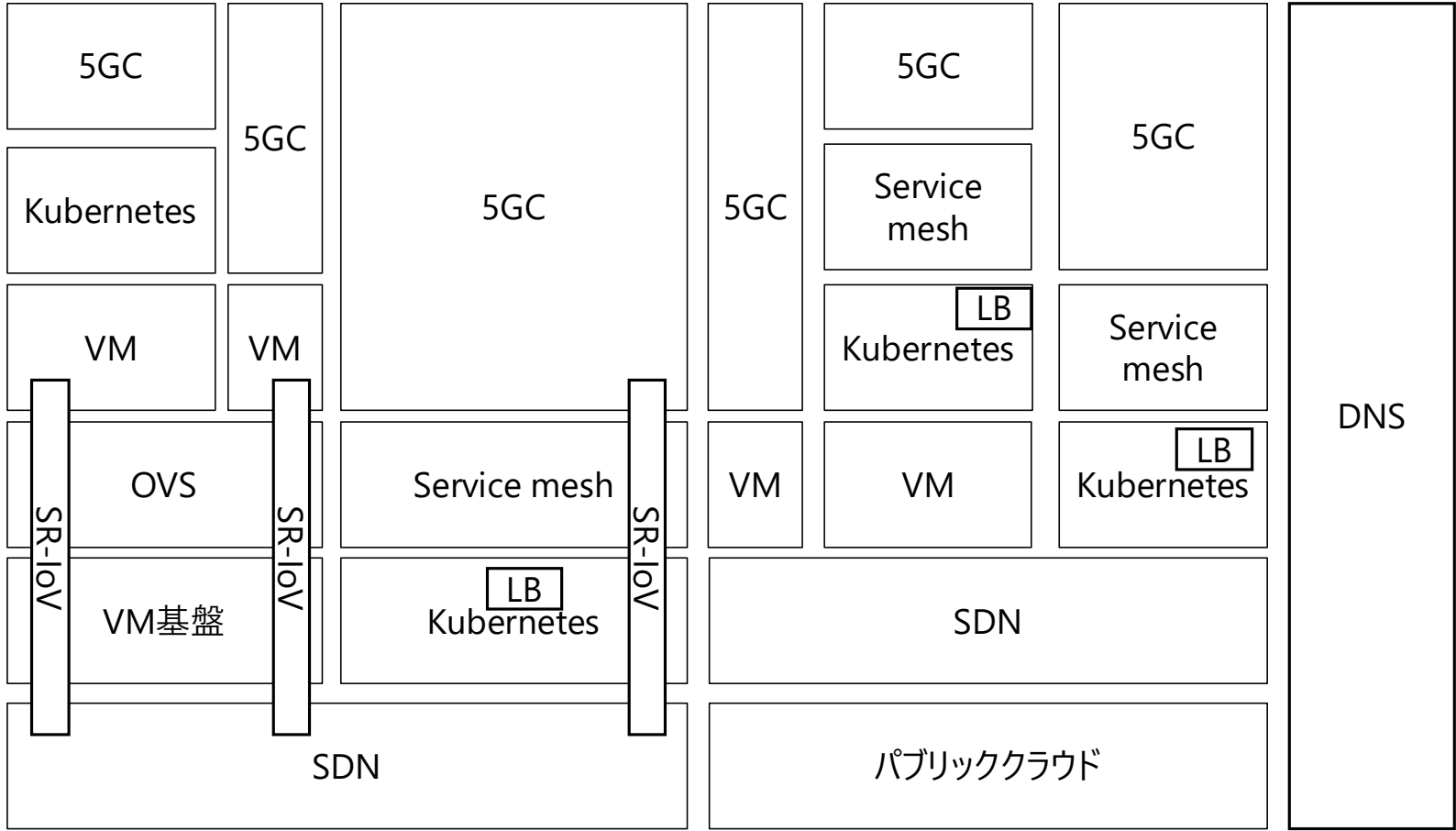
パブリッククラウド上の5GCを検討する中でベアメタルコンテナの構成も出てきた…

※JANOG54 “通信事業者特有のNW要件に合わせたクラウドのNW設計 ～出たばかりのマルチVPC ENIを5GConAWSで活用してPoCしてみた～”等を参照



その結果

フルコンテナ化と言いつつもVM構成も残り、ベアメタルコンテナ構成も出てきてより複雑に…
一概に5GCをコンテナ化したと言っても、要件によって構成が複雑になり 運用 も大変になる



ここまでのまとめ

従来の技術で 動かし続けてきた モバイルテレコム

新たな思想としてのクラウドネイティブ の導入

23年度のフルコンテナ化によりいったんは単純化させたが、また複雑化する構成
残り続ける課題...

フルコンテナ化後に残る今後の懸念・課題

Openshift 基盤 の 困りごと

2年に一度 式年遷宮してます

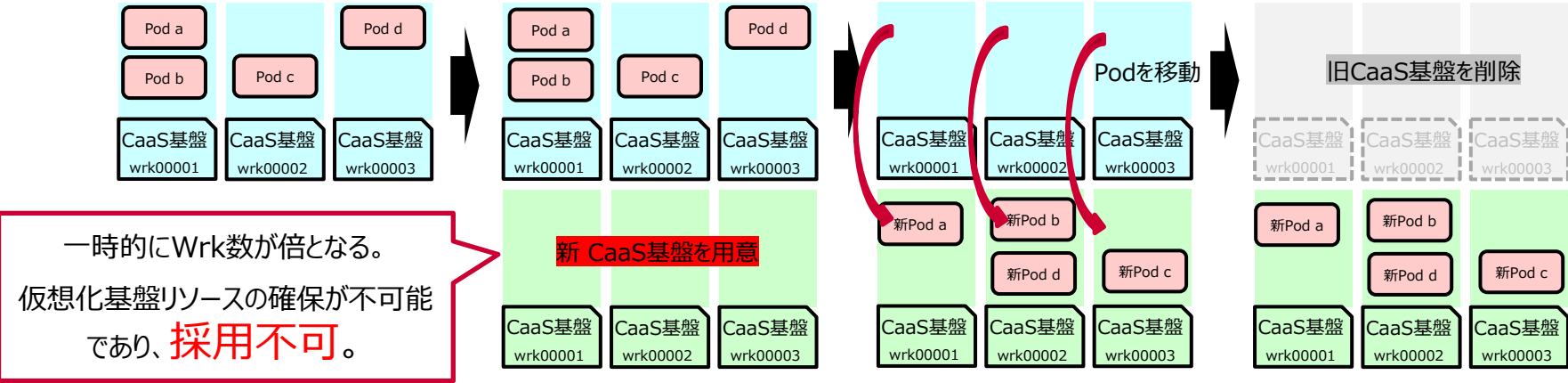
2021年の Sin 後、 実は2回 ターミネーション・再インスタンスーションにより すべて 作り変えています。

年度	ファイルアップデート方法	理由
21年度	ターミネーション・再インスタンスーション	SAスマホ導入のためのアーキテクチャ変更
22年度	オンラインアップデート	
23年度	ターミネーション・再インスタンスーション	フルコンテナ化のためのアーキテクチャ変更
24年度	オンラインアップデート	

ブルーグリーンデプロイメントができるような基盤容量を、同一NW上に 確保できず、 やむなしでした。

ブルーグリーンデプロイメント方式 （仮想化基盤リソース確保ができず、見送った案）

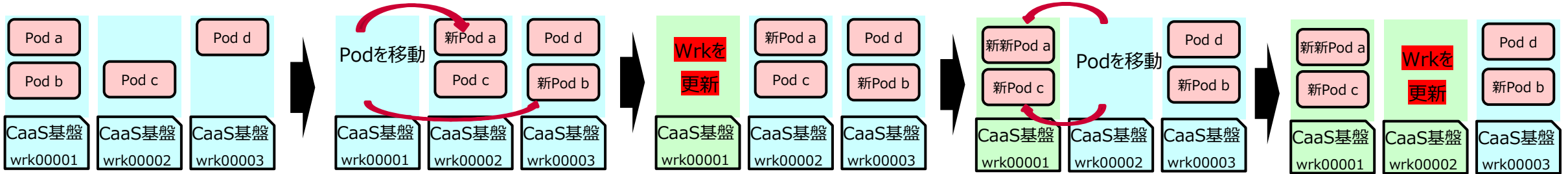
新旧 IPアドレスを同一流用により重複ができないため同一NW上に構築せざるを得ず、ターミネーションした後に再インスタンスーションで新ファイルを立ち上げなおし。



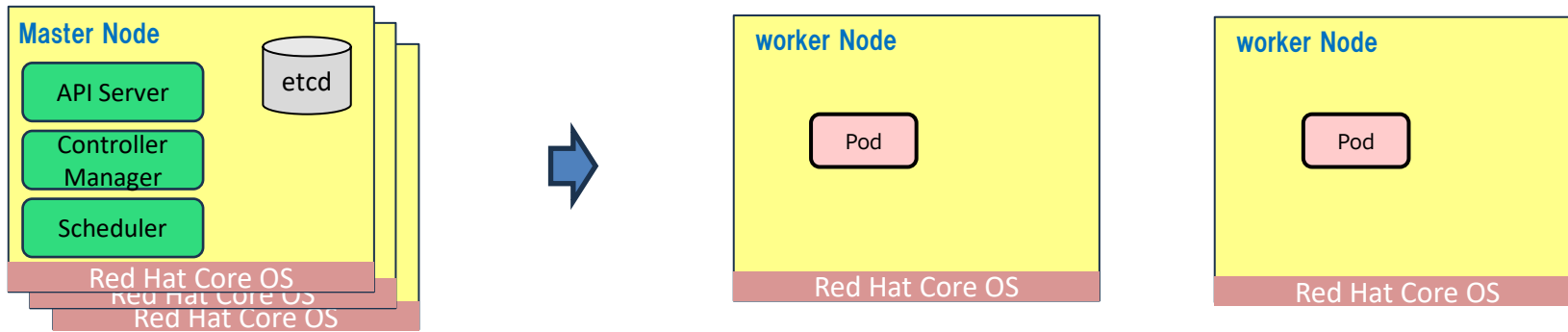
Openshift 更新時に 収容装置の迂回が必要だったりしました

オンラインアップデートを実施した年度もopenshift 更新は 上位アプリpod をローリングしながらアップデートするため、基本影響はないことを前提にしてましたが、検証結果としてmst更新時に 5 分くらいパケロス時間があったりしました。

ローリングアップデート



Mst更新時に workpod側でパケットロスが発生

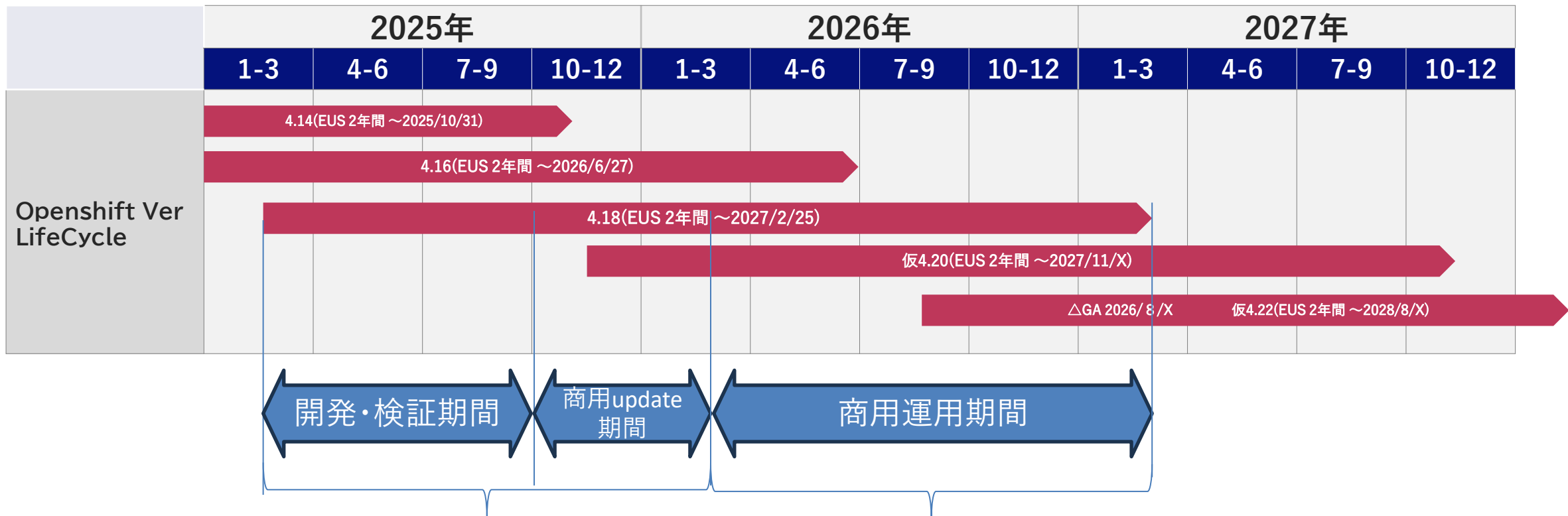


5 分のパケットロス ⇒ 全セッションが切断されるレベルなので、そのままだと、集中した再接続要求による輻輳懸念があったため、まず収容されている装置から別装置に緩やかにトラフィックを迂回させる必要がありますごく手間でした。

⇒ こんなことって起こったことがありますか？ アプリの作りが悪いのでしょうか？

Openshift ライフサイクル

Openshift ライフサイクル平均、8～9か月 毎リリースの EUSは 2年。
開発期間・検証期間・更改期間を踏まえると、商用運用可能な 期間は1年足らず。
毎年 更新が必要なのは 結構大変です。



2年間のうち 商用運用 しているのはおおむね 1 年程度。 OS更新頻度としては、短く感じる。

⇒ 検証期間と商用運用期間のバランスってどれくらいですか？

アプリ の 困りごと

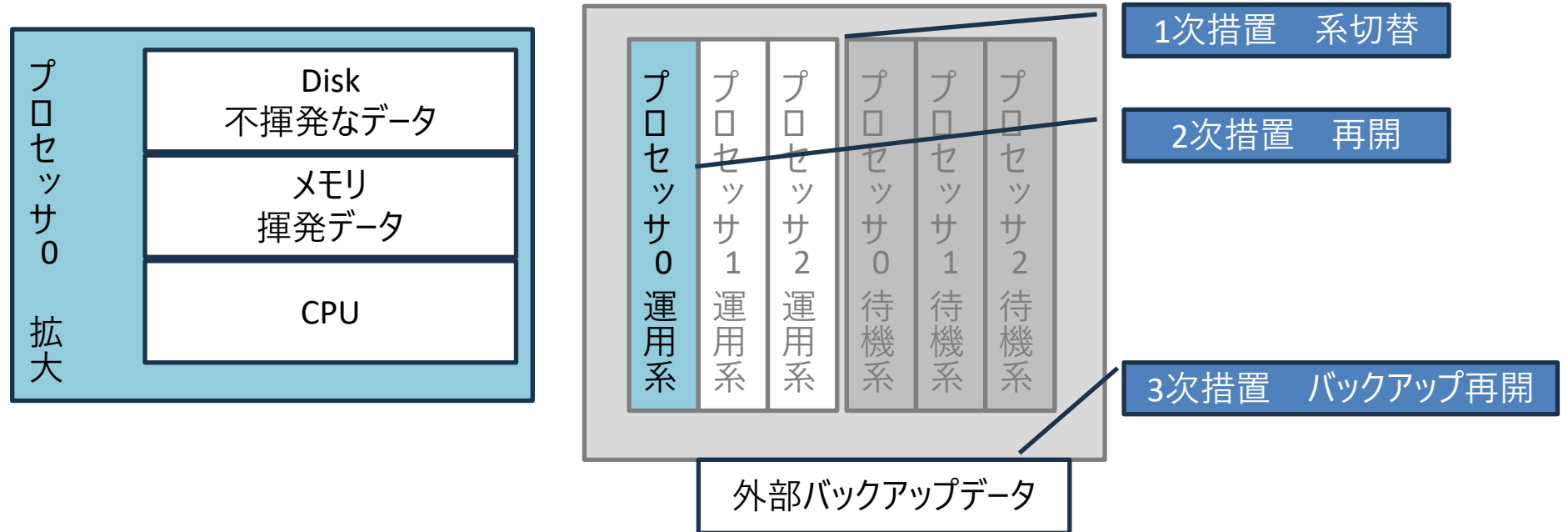
- ① 変化した 構成に対する 措置オペレーションの追従方法
- ② ステートを保持DBpodと処理系pod

①変化した 構成に対する 措置オペレーションの追従方法

過去の装置の構成 と 段階ごとの 措置オペレーション

プロセッサ内にCPU、メモリ、diskが存在。

設定データはdisk内に不揮発なデータとして設定、 運用情報/呼処理 は揮発データとしてメモリに格納。



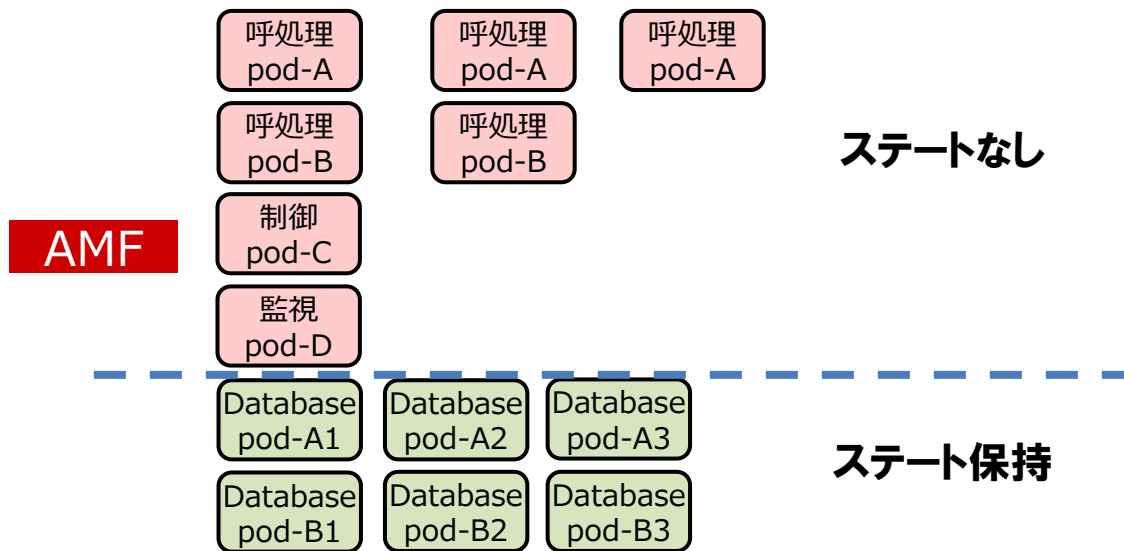
1次措置として 系切替で呼処理を継続させながらの措置を試み、 復旧しない故障などの発生時には、
2次措置として プロセッサ再開（リブート）し、不揮発なデータから再立ち上げすることで、10分程度で復旧可能。
3次措置として 外部バックアップデータからの復旧（数年に1度くらいのレアケース。）

実際のオペレーションは、2次措置または3次措置の前段階として、ユニットを網から切り離すことで、他ユニットへの迂回を行うことが多い。

①変化した 構成に対する 措置オペレーションの追従方法

5GC構成 と 段階ごとの 措置オペレーション

5GC NFは複数のpod群により構成。



テレコム過去のシステムとの差分

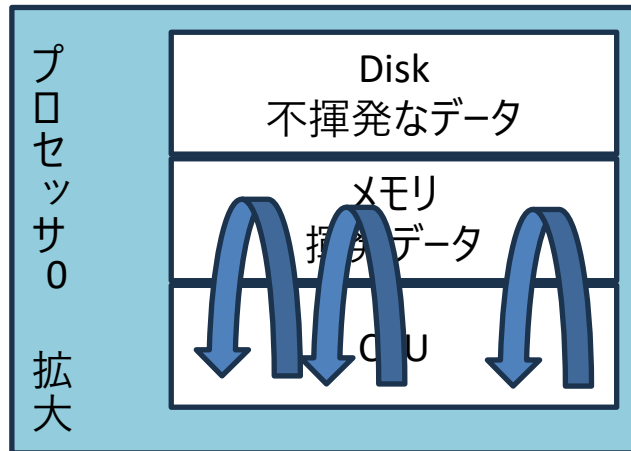
- Pod数がプロセッサと比較して多い。
- ステート保持するDBとステートレスな呼処理podが分離
- pod単体としては正常動作しているように見えても、システムとして異常な状態となっていたことがあった。
- 被疑箇所特定が困難な場合が多い。
- 被疑プロセッサだけをまるっと再開してきれいに立ち上げなおすという措置ができない。
(全部DB クリアして、立ち上げなおし=再インスタに等しい)

1次措置として 被疑箇所が即時特定できれば、系切替やpodヒーリングで呼処理を継続させながらの措置を試みるが、
2次措置として ユニットの切り離しを行いサービス復旧までは行うが、そののちの復旧方法は不具合の状態により
復旧方法がまちまちでパターン化ができておらず、体制構築の上復旧作業を実施。

⇒ 皆様 どこまでの措置を、汎用手順化されていますか？

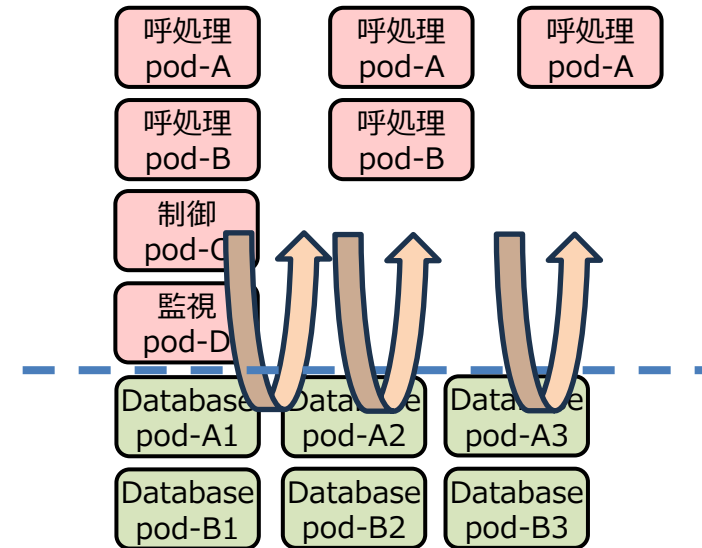
②ステートを保持するDBpodと処理系pod 間通信

過去システム



ステート情報取得はプロセッサ^内で完結

5GC

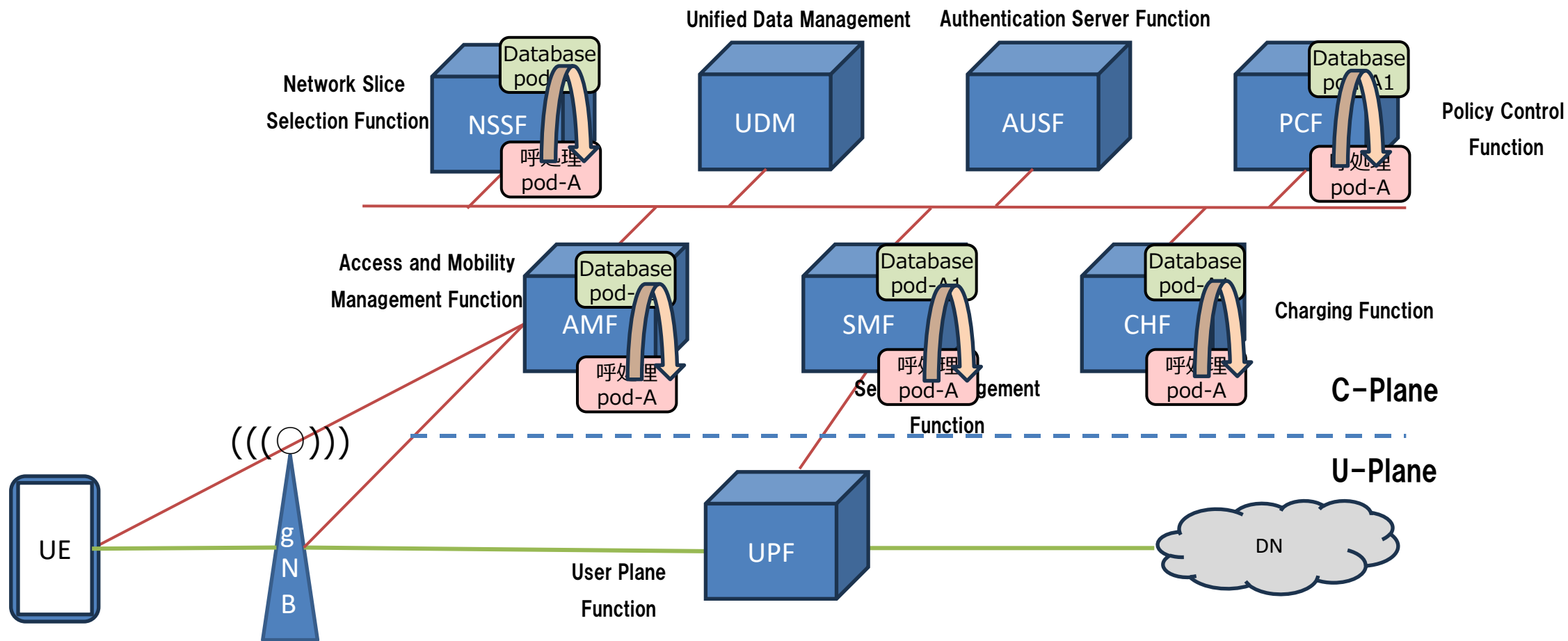


ステート情報取得がpod^間通信

内部処理が 外部通信に変化

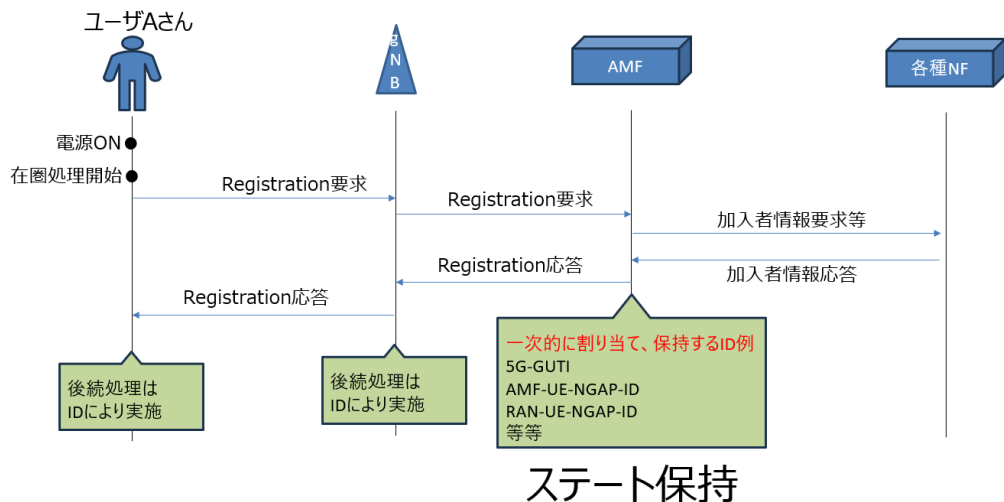
② NFへのアクセス

1UEからの要求だけでも複数のNFへのアクセス（＝DBアクセス）

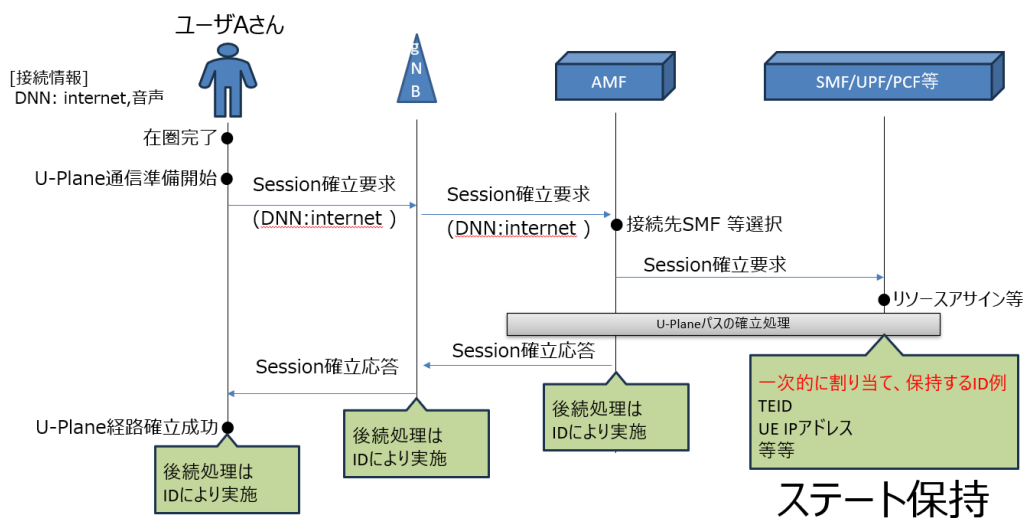


② テレコム通信によるDBアクセス頻度

トピック② 参考 接続処理のお話 (mobility)



トピック② 参考 接続処理のお話 (session)



左記 二つは一番基本の接続処理

実際には、数十m移動して接続するgNBが変わるだけのpath switch処理レベルでも

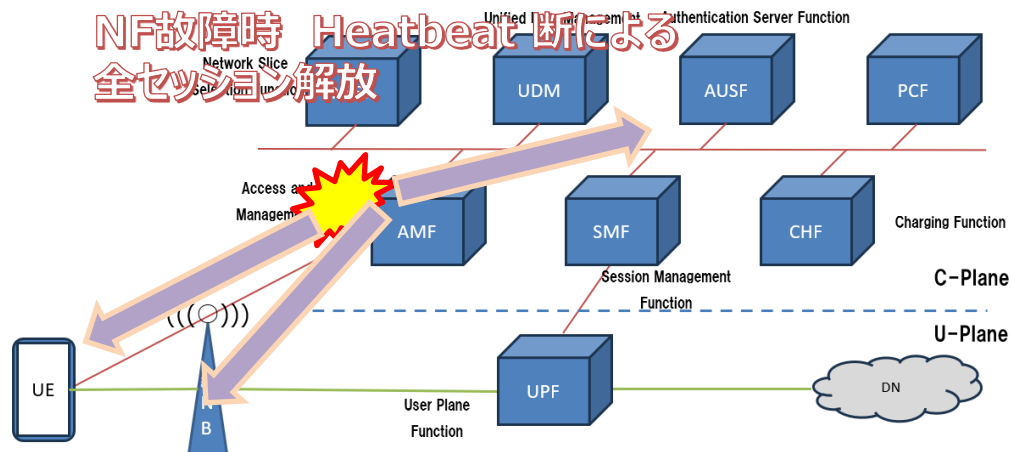
- ・pod毎に加入者情報取得でDBアクセス(N2-IF, SBI毎とか)
- ・パス更新処理でDB書き込み
- ・上記各タイミングで呼処理の進行情報保存

等複数回のDBアクセス

現時点だと SAエリア展開状況から eps/5gs間の RAT間 N26 H.O.も非常に多く よりハード

テレコムの通信頻度 多すぎ。。

② 一斉再接続/信号再送による DBアクセス 輻輳 懸念

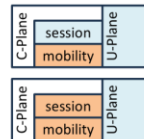


トピック② スマートホンの台頭によるNWへのインパクト

フィーチャーホンとスマートホンの NWから見た違い (詳細 次頁)

フィーチャーホン：未使用時はmobilityのみ制御、
使用時にsession/Uplaneを制御

スマートフォン：未使用時にもmobility/sessionを制御し、
使用時にU-Plane を起動



状態管理 が 倍以上大変になった。

また、端末は常に接続を維持し続けるため、不具合等の発生によりNWから切断された場合、
切断された端末 (数十万台から数百万台) が 一斉(数分の間) に再接続を要求する。

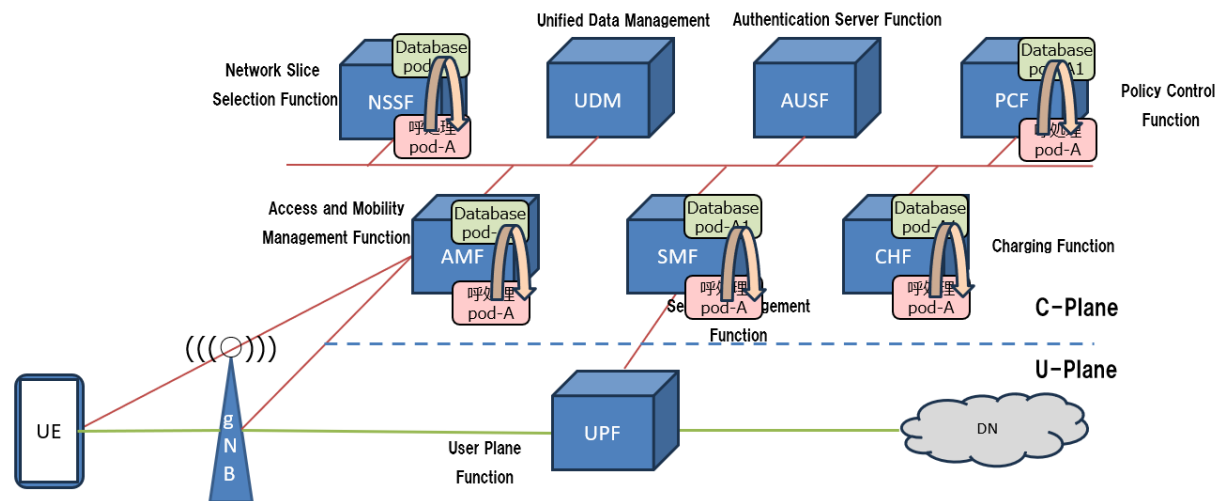
最悪時 1万TPSを超過する 接続要求

スマートホンの台頭以降から モバイル コアNWは 輻輳 との 戦い

©2025 NTT DOCOMO, INC. All Rights Reserved.

10

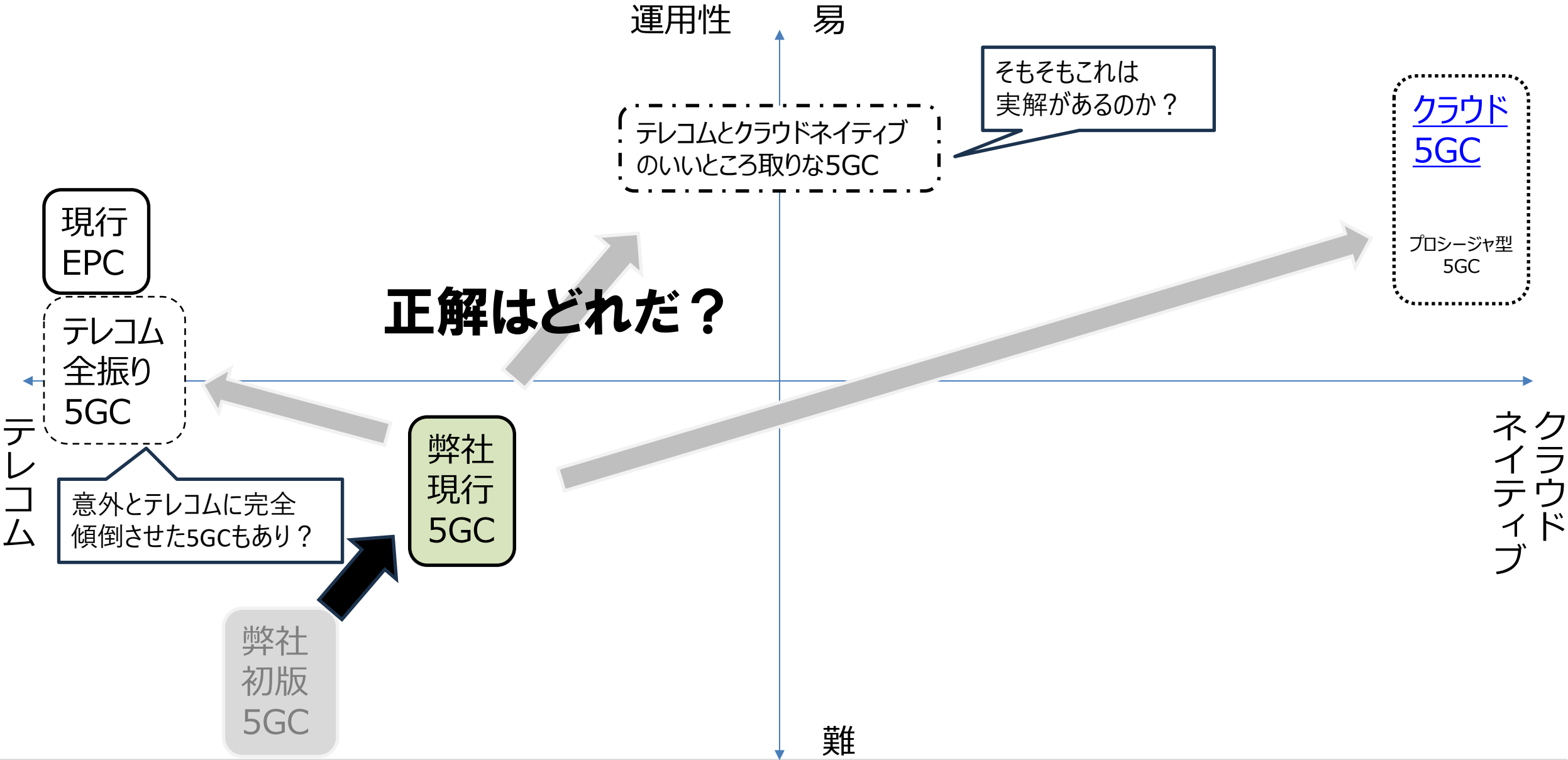
数十万端末からの一斉再発信時におけるDBアクセス



信号再送等を考慮し、
DBへのアクセスをうまくコントロールしないと
Pod/DB間アクセスが輻輳する
懸念有り

2年前に実際発生 実績あり。。。

修羅場から脱却したが、向かうべきゴールはどこ？



DISCUSSION