

クラウドネイティブな5Gコアをつくってみた話

ENOG 84 俺の話を聞け枠

† 渡邊 大記 (WATANABE Hiroki)

† SoftBank Corp.

おことわり：本題に入る前に



- 今日の話の類似コンテンツ

- 「パブリッククラウド上で作るサーバレスな5Gコアネットワーク」
- <https://www.softbank.jp/corp/technology/researchn/story-event/037/>
- 会社組織を代表するものではない

モバイルコアの概要とつらみや学びを共有したいです

モバイルシステムは **“”すごい””** ということを共有したい

背景：社会インフラとしての移動体通信システム

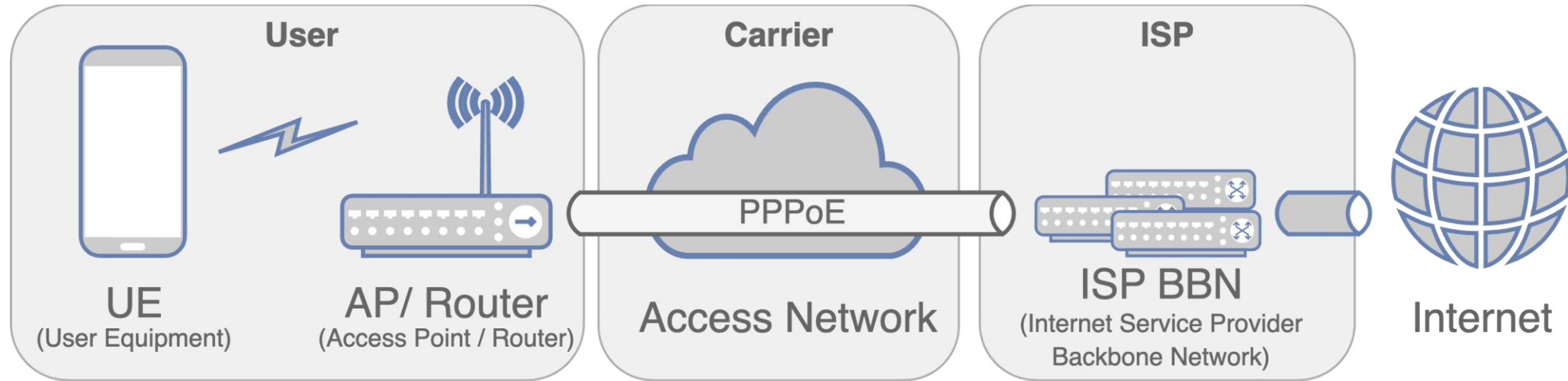
- **1000万オーダーの加入者が接続する巨大通信システム**
 - 5G からは**機能のソフトウェア化(仮想化)**が加速
 - 柔軟な構成がとれることに期待がある
- **ときどき大規模障害を引き起こす**
 - 主な原因：突発的な大量の制御メッセージの流入による輻輳
 - ・ メンテナンス明けからの切り戻し、運用ミスによるメッセージストーム

なぜ障害が大規模化する傾向に？

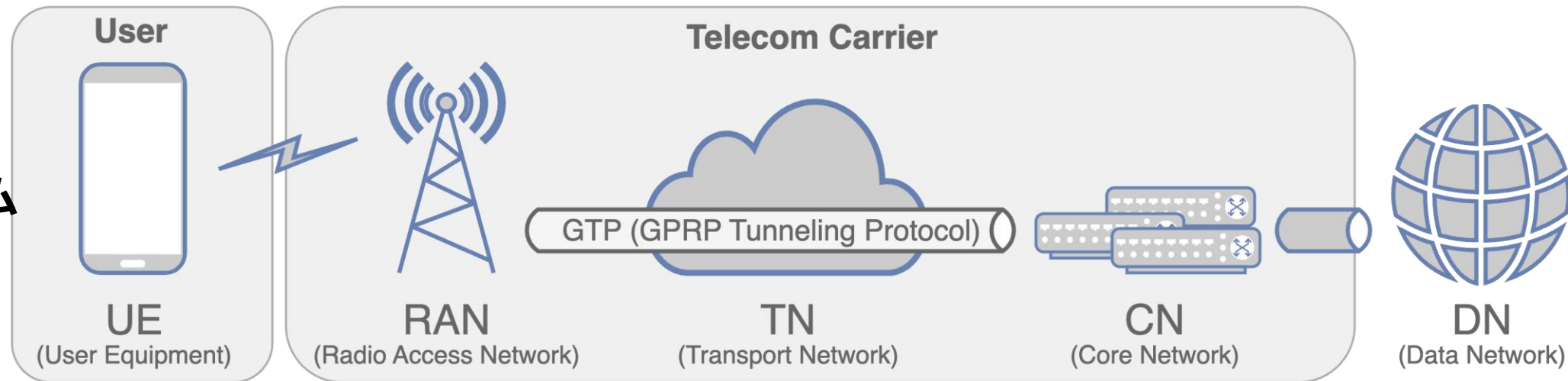
コンピューティング技術の観点から移動体通信システムを整理してみる

接続性におけるアナロジー

インターネット

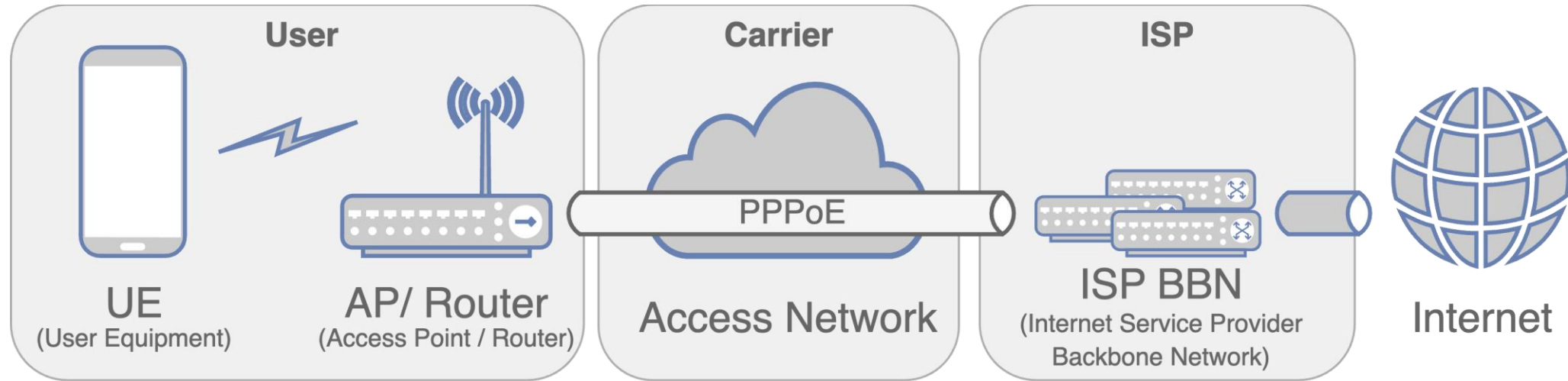


移動体通信システム
(モバイルシステム)

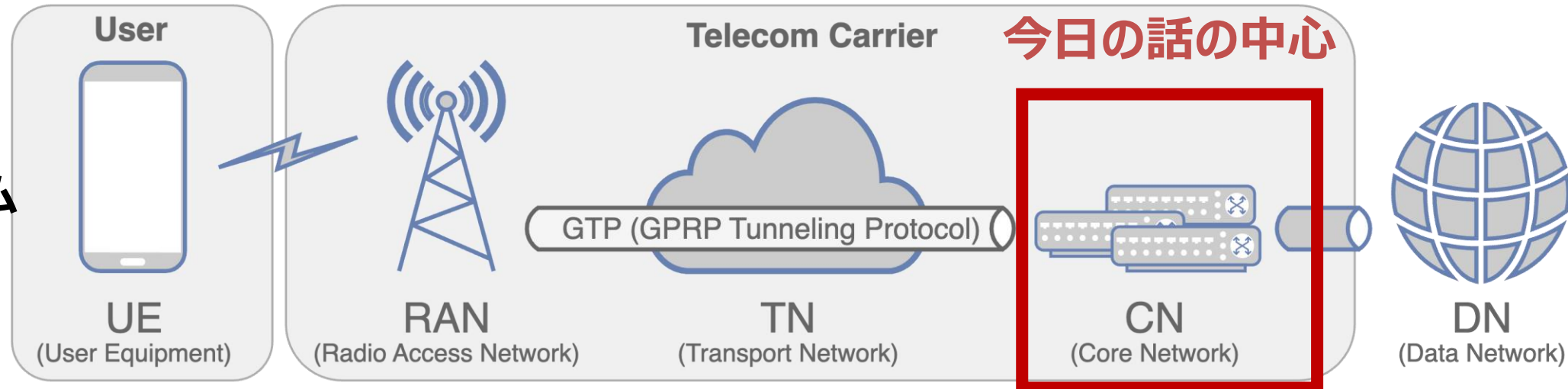


接続性におけるアナロジー

インターネット

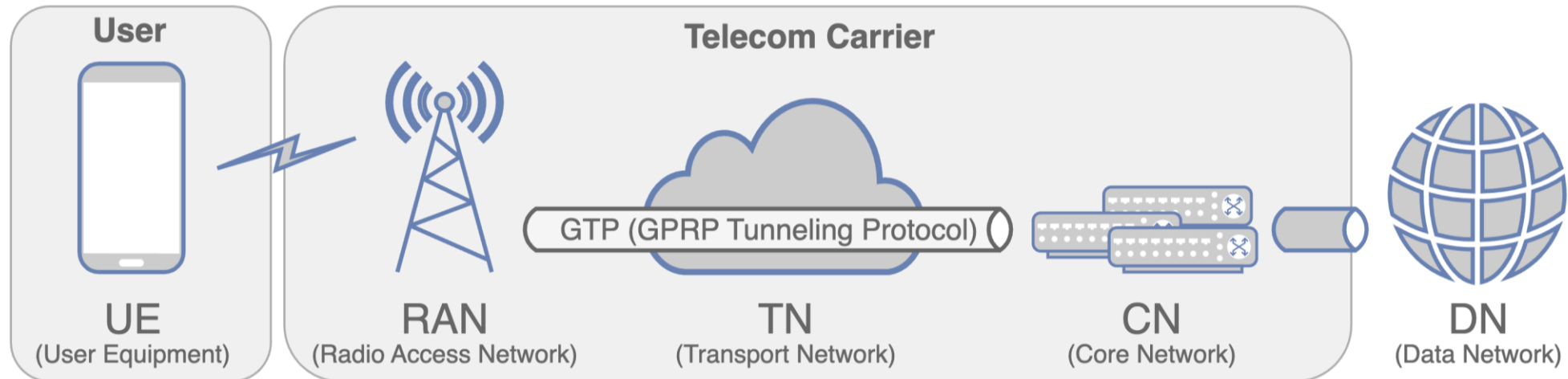


移動体通信システム
(モバイルシステム)



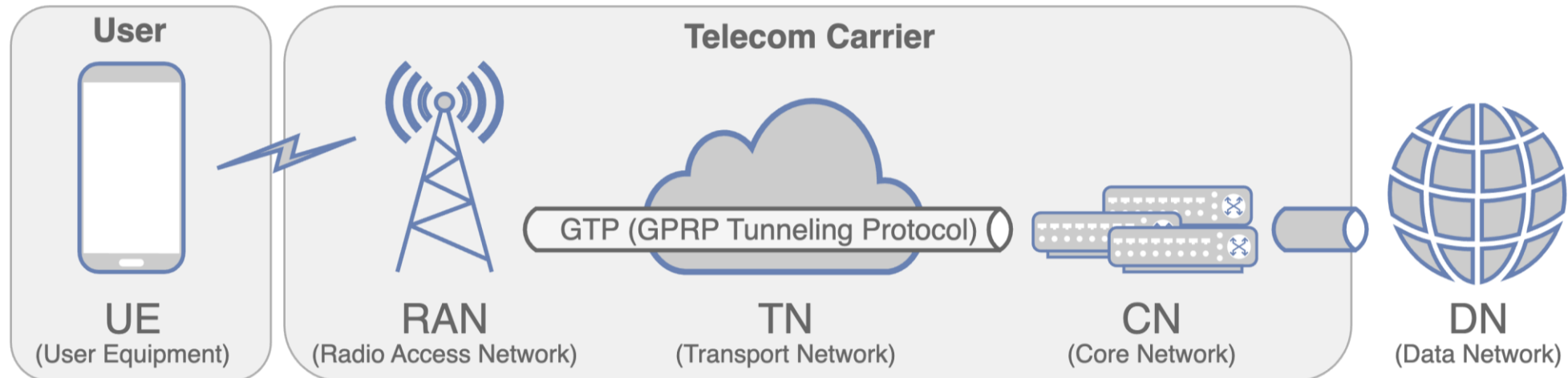
モバイルシステムとコアネットワーク（CN）

- CN：ユーザの認証認可や移動管理などをつかさどる制御部



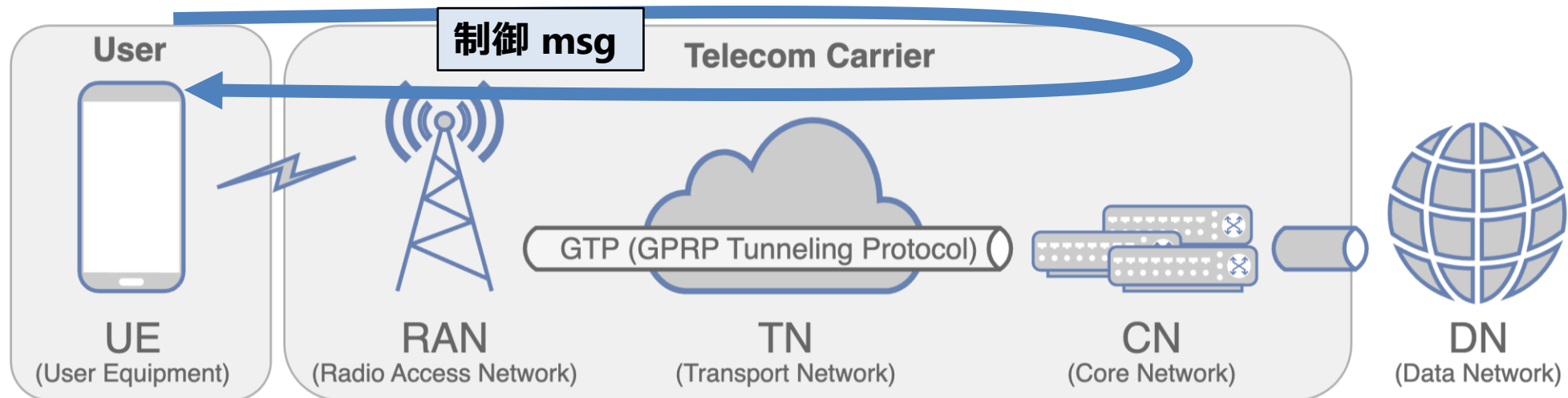
モバイルシステムとコアネットワーク（CN）

- **CN：ユーザの認証認可や移動管理などをつかさどる制御部**
 - **Req-Res 型のサーバサイドアプリケーションと類似**
 - ・ 制御メッセージのプロトコルシーケンスに基づくように見なせる



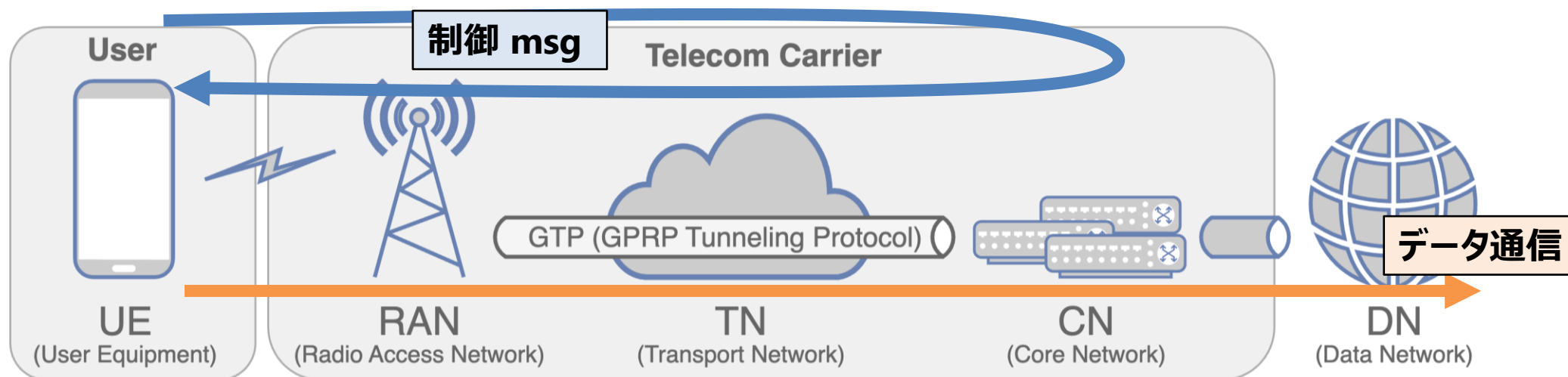
モバイルシステムとコアネットワーク（CN）

- **CN：ユーザの認証認可や移動管理などをつかさどる制御部**
 - **Req-Res 型のサーバサイドアプリケーションと類似**
 - 制御メッセージのプロトコルシーケンスに基づくとそう見なせる



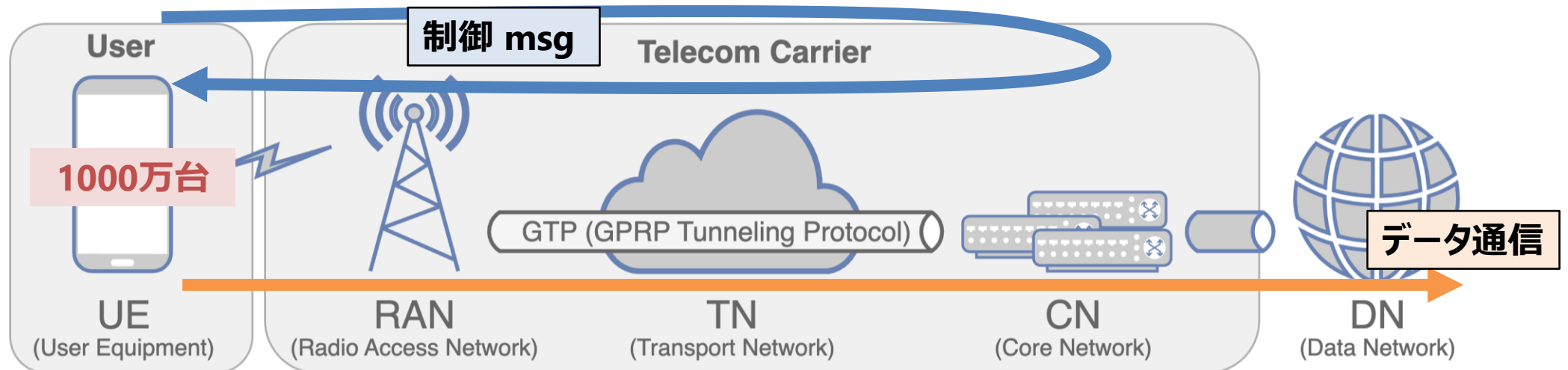
モバイルシステムとコアネットワーク（CN）

- **CN：ユーザの認証認可や移動管理などをつかさどる制御部**
 - **Req-Res 型のサーバサイドアプリケーションと類似**
 - ・ 制御メッセージのプロトコルシーケンスに基づくように見せる



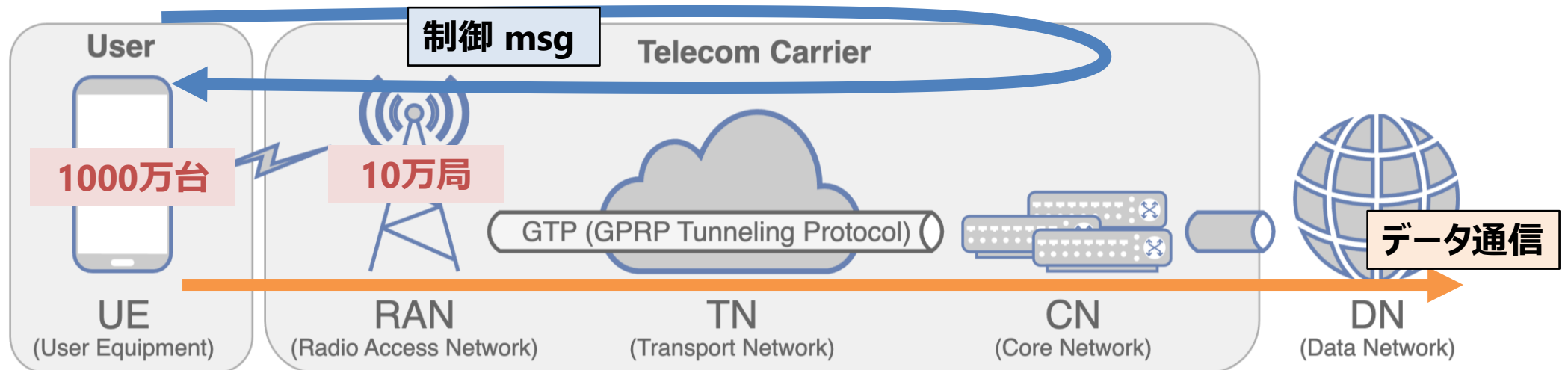
モバイルシステムとコアネットワーク（CN）

- **CN：ユーザの認証認可や移動管理などをつかさどる制御部**
 - **Req-Res 型のサーバサイドアプリケーションと類似**
 - 制御メッセージのプロトコルシーケンスに基づくように見なせる



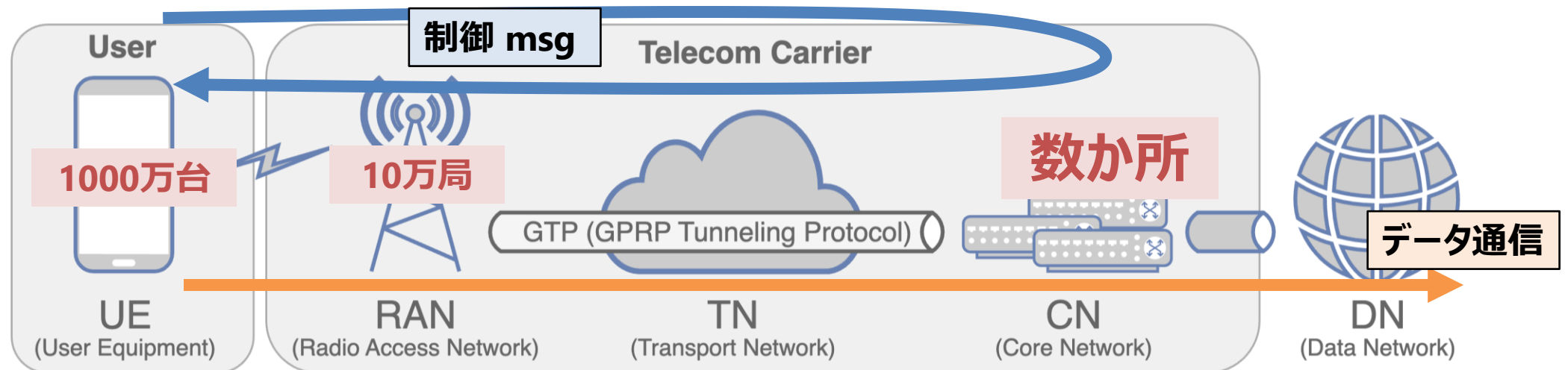
モバイルシステムとコアネットワーク（CN）

- **CN：ユーザの認証認可や移動管理などをつかさどる制御部**
 - **Req-Res 型のサーバサイドアプリケーションと類似**
 - 制御メッセージのプロトコルシーケンスに基づくように見せる



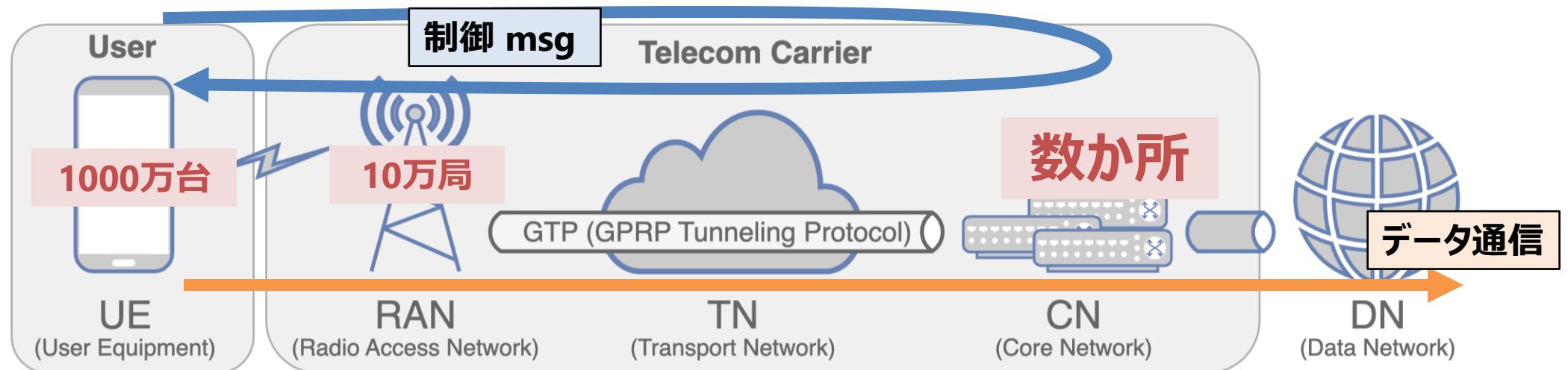
モバイルシステムとコアネットワーク（CN）

- **CN：ユーザの認証認可や移動管理などをつかさどる制御部**
 - **Req-Res 型のサーバサイドアプリケーションと類似**
 - ・ 制御メッセージのプロトコルシーケンスに基づくとそう見なせる



モバイルシステムとコアネットワーク（CN）

- **CN：ユーザの認証認可や移動管理などをつかさどる制御部**
 - **Req-Res 型のサーバサイドアプリケーションと類似**
 - ・ 制御メッセージのプロトコルシーケンスに基づくように見なせる
 - **巨大な箱に全端末を収容するコンピューティングモデル**

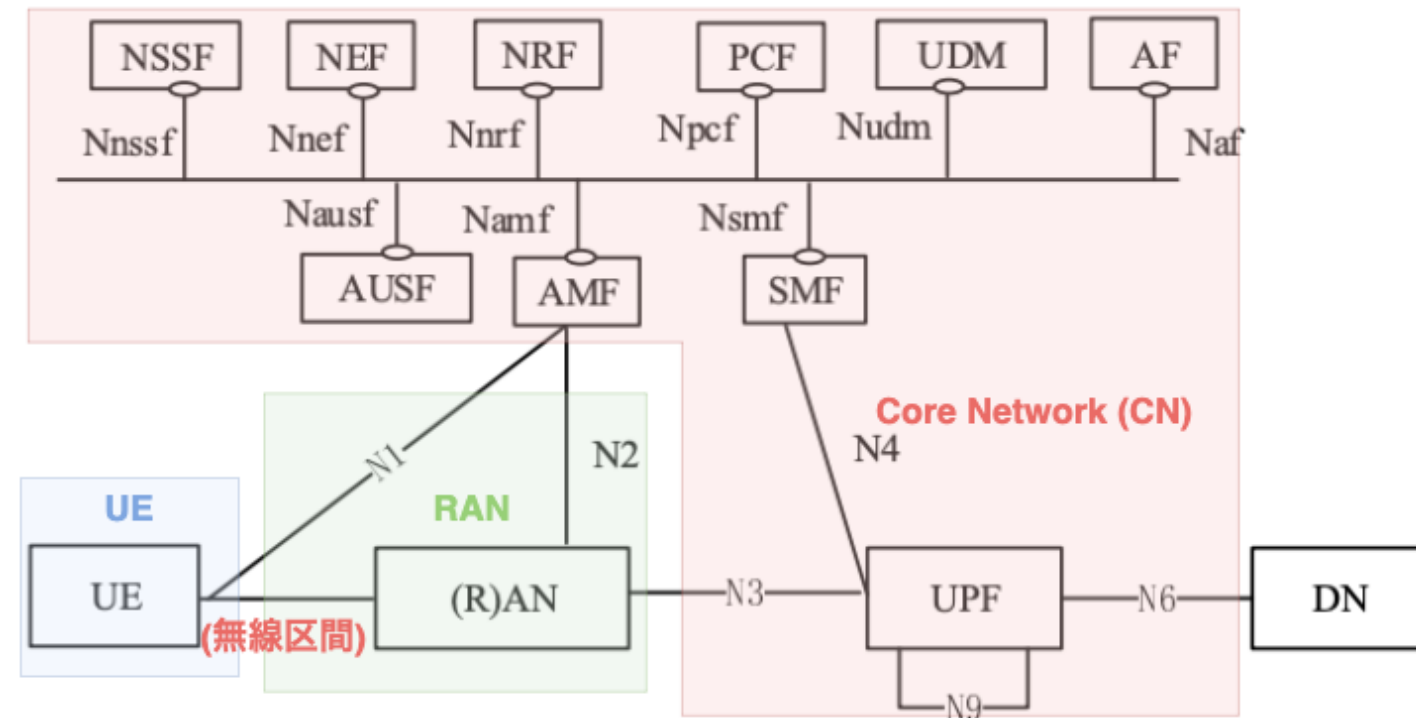


5G System (5GS) リファレンスアーキテクチャ^[1]

- **UE/RAN/CN から構成**

- CN は NF による
マイクロサービスアーキテクチャ
- 制御プレーンは各 NF の
相互呼び出し (REST API)

UE: User Equipment
RAN: Radio Access Network
NF: Network Function
CN: Core Network
AMF: Access and Mobility management Function
SMF: Session Management Function
UPF: User Plane Function



[1]. 3GPP, **System architecture for the 5G System (5GS). Technical Specification (TS) 23.501**

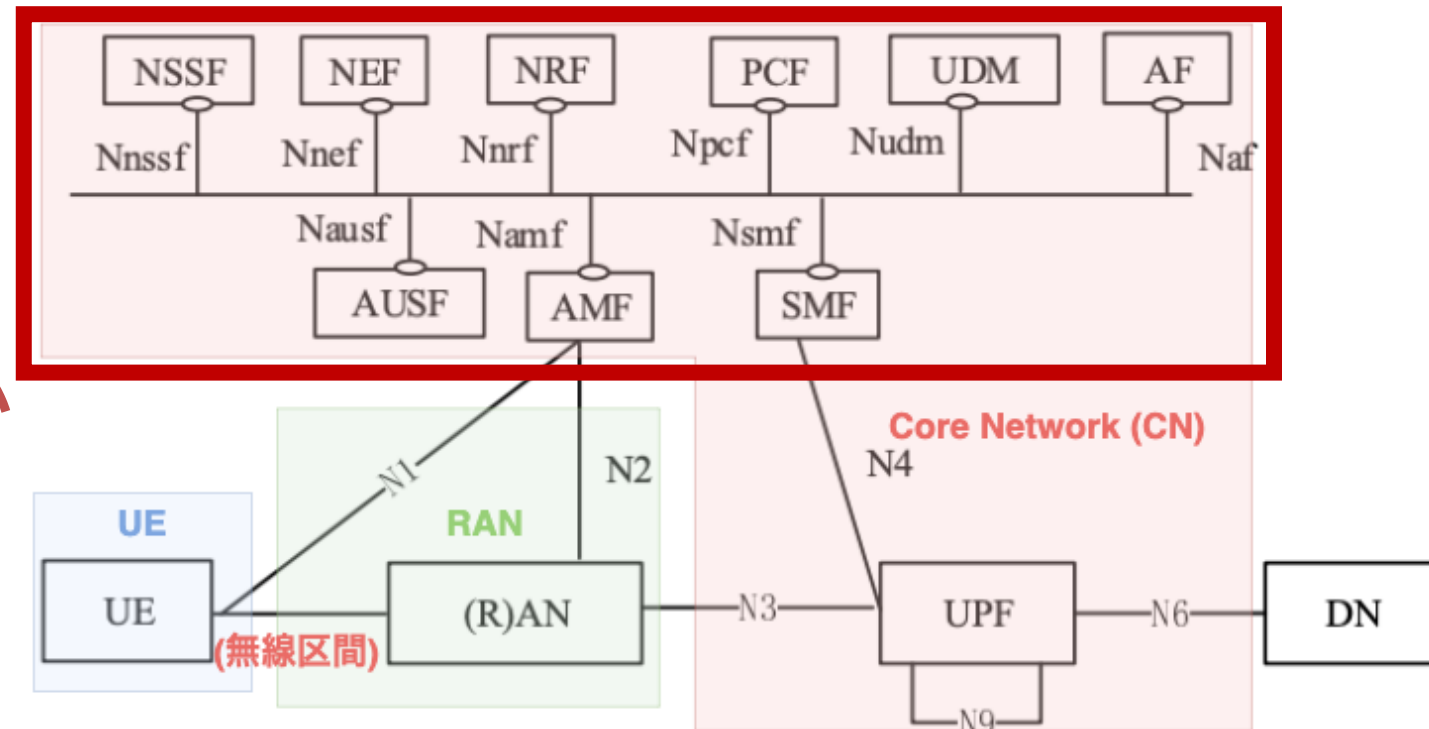
5G System (5GS) リファレンスアーキテクチャ^[1]

- **UE/RAN/CN から構成**

- CN は NF による
マイクロサービスアーキテクチャ
- 制御プレーンは各 NF の
相互呼び出し (REST API)

UE: User Equipment
RAN: Radio Access Network
NF: Network Function
CN: Core Network
AMF: Access and Mobility management Function
SMF: Session Management Function
UPF: User Plane Function

今日の話の中心



[1]. 3GPP, **System architecture for the 5G System (5GS). Technical Specification (TS) 23.501**

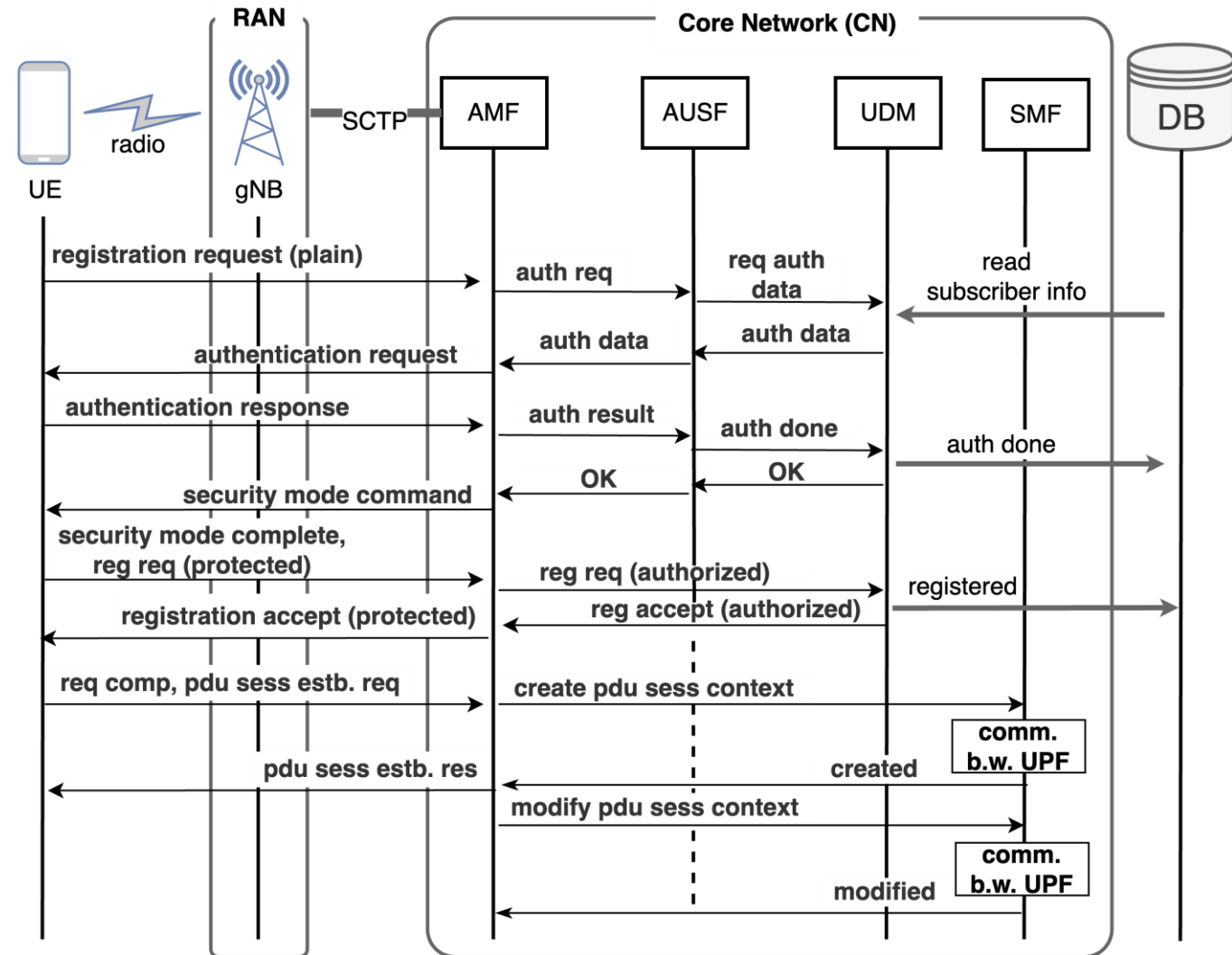
コアネットワーク (CN; 制御部/交換機)

- UE の状態を管理する

- 各 NF の役割ごとにデータ管理
- NF 間にまたがる整合性が重要

UE の状態を示す揮発性データ

暗号鍵、seq 番号、**認証情報**、**UE IP Addr.**、**無線区間 ID** 等

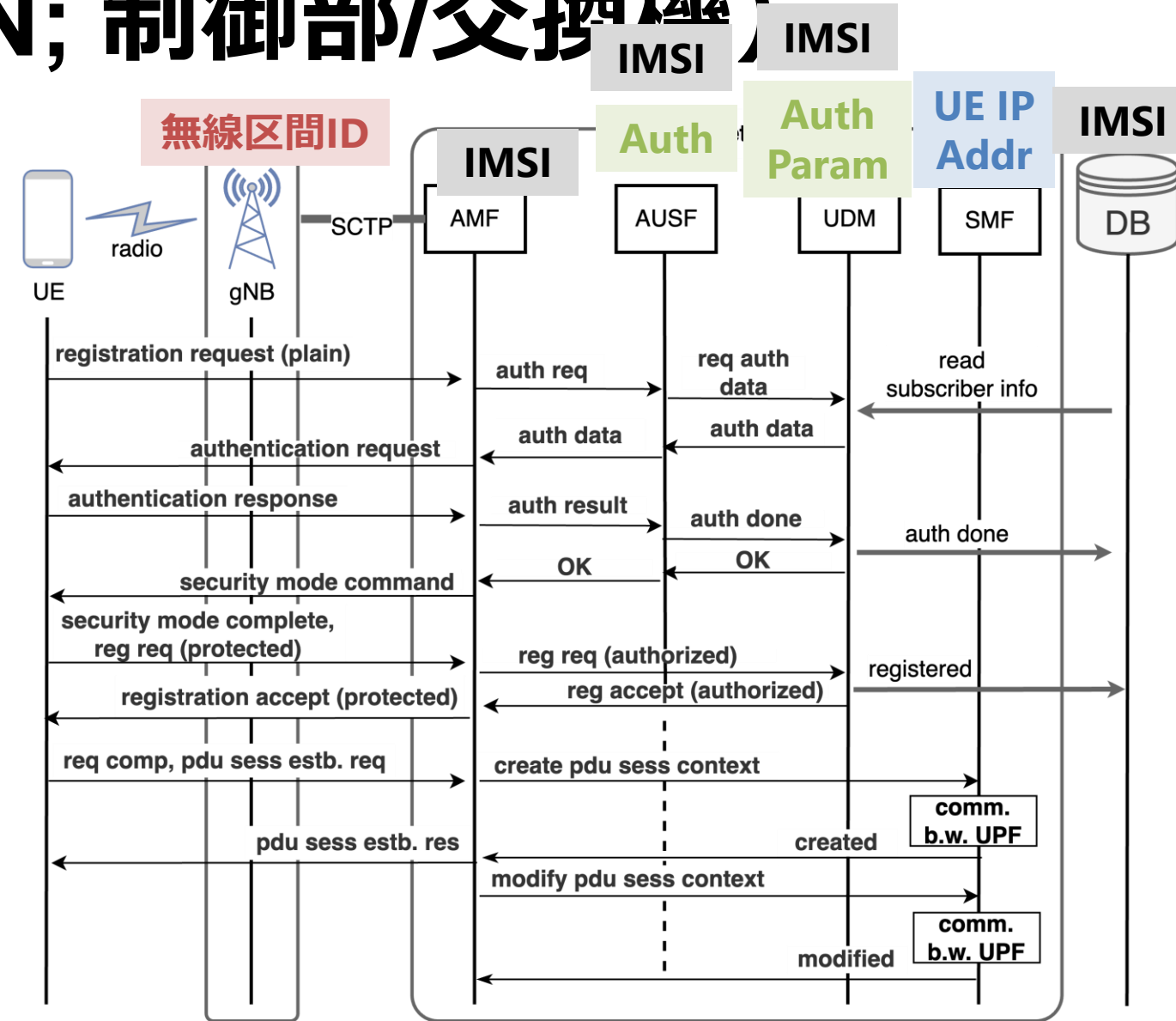


コアネットワーク (CN; 制御部/交換機)

● UE の状態を管理する

- 各 NF の役割ごとにデータ管理
- NF 間にまたがる整合性が重要

UE の状態を示す揮発性データ
 暗号鍵、seq 番号、**認証情報**、**UE IP Addr.**、**無線区間 ID** 等



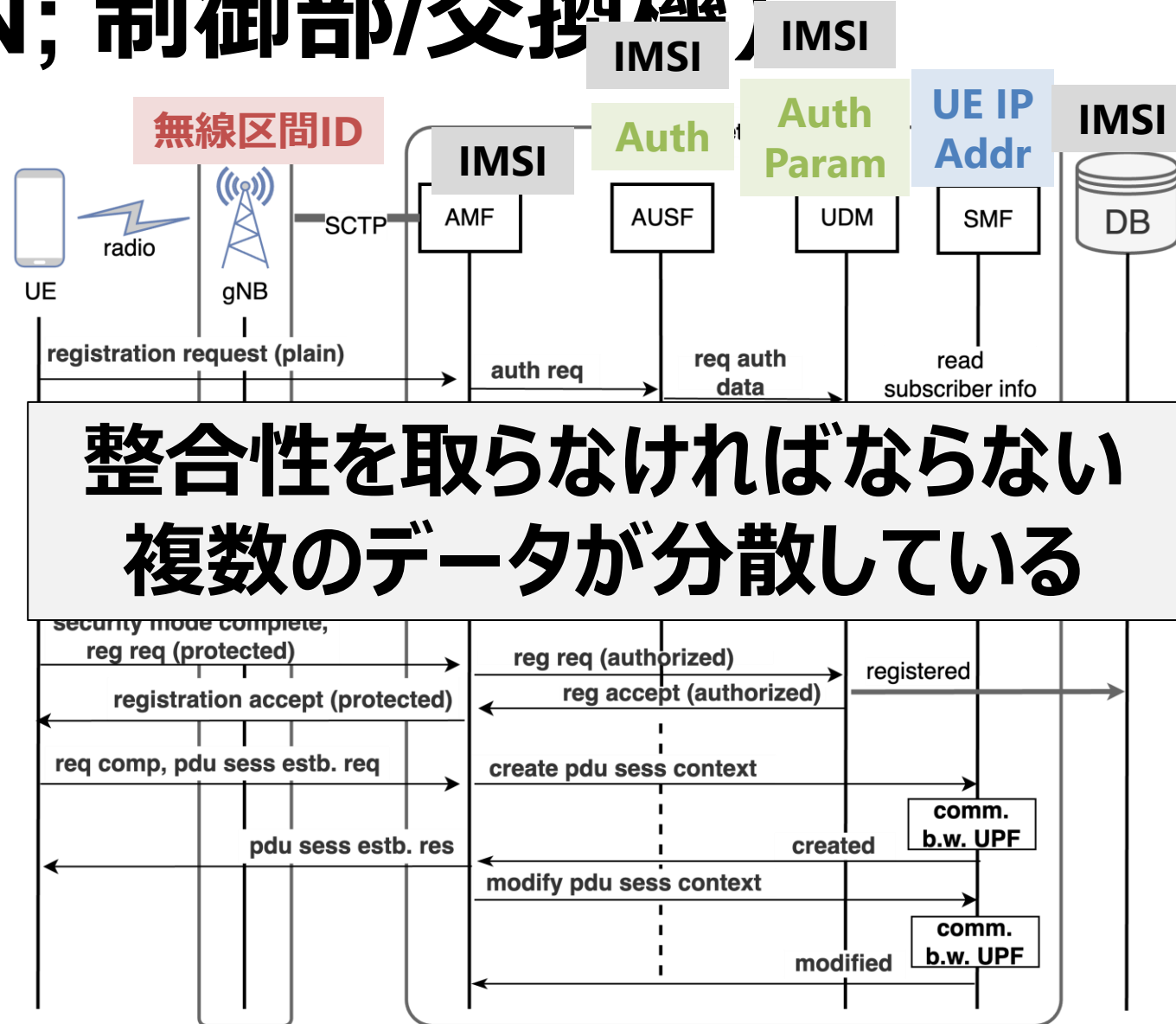
コアネットワーク (CN; 制御部/交換機)

● UE の状態を管理する

- 各 NF の役割ごとにデータ管理
- NF 間にまたがる整合性が重要

UE の状態を示す揮発性データ

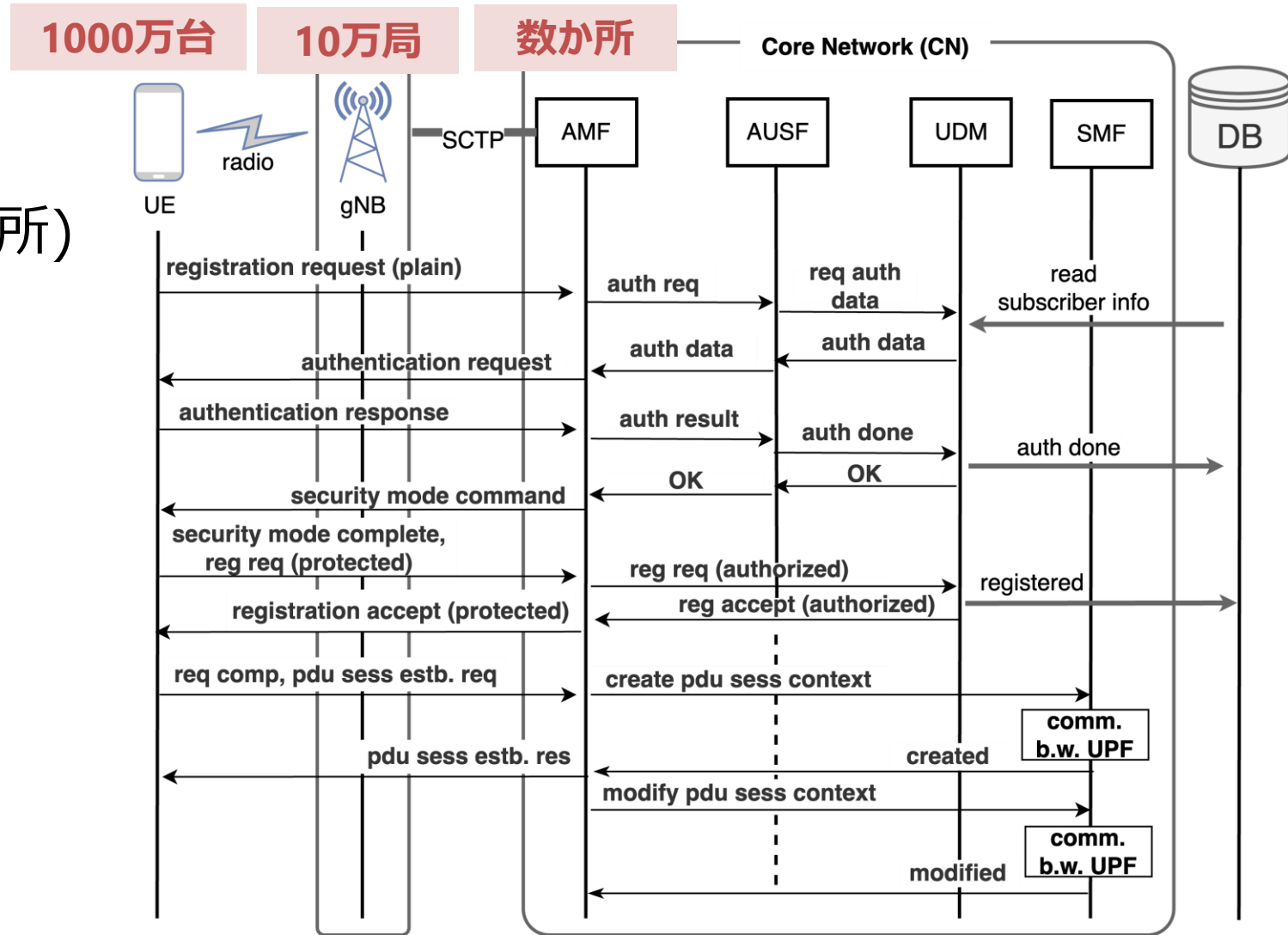
暗号鍵、seq 番号、**認証情報**、**UE IP Addr.**、**無線区間 ID** 等



端末数 x 複雑バケツリレー = 障害の大規模化

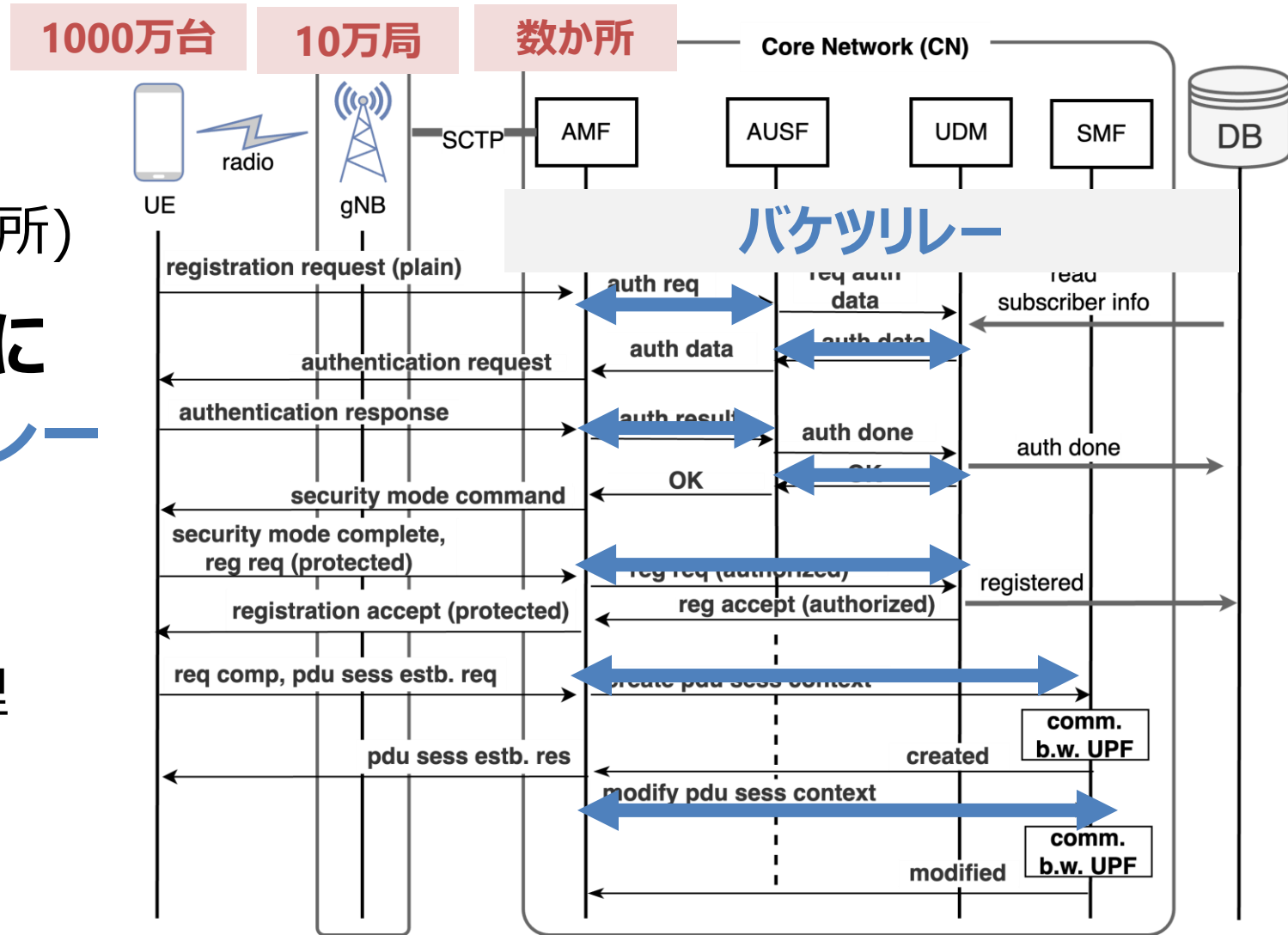
- AMF への過集中

- UE(10⁷台) : AMF(<数ヶ所)

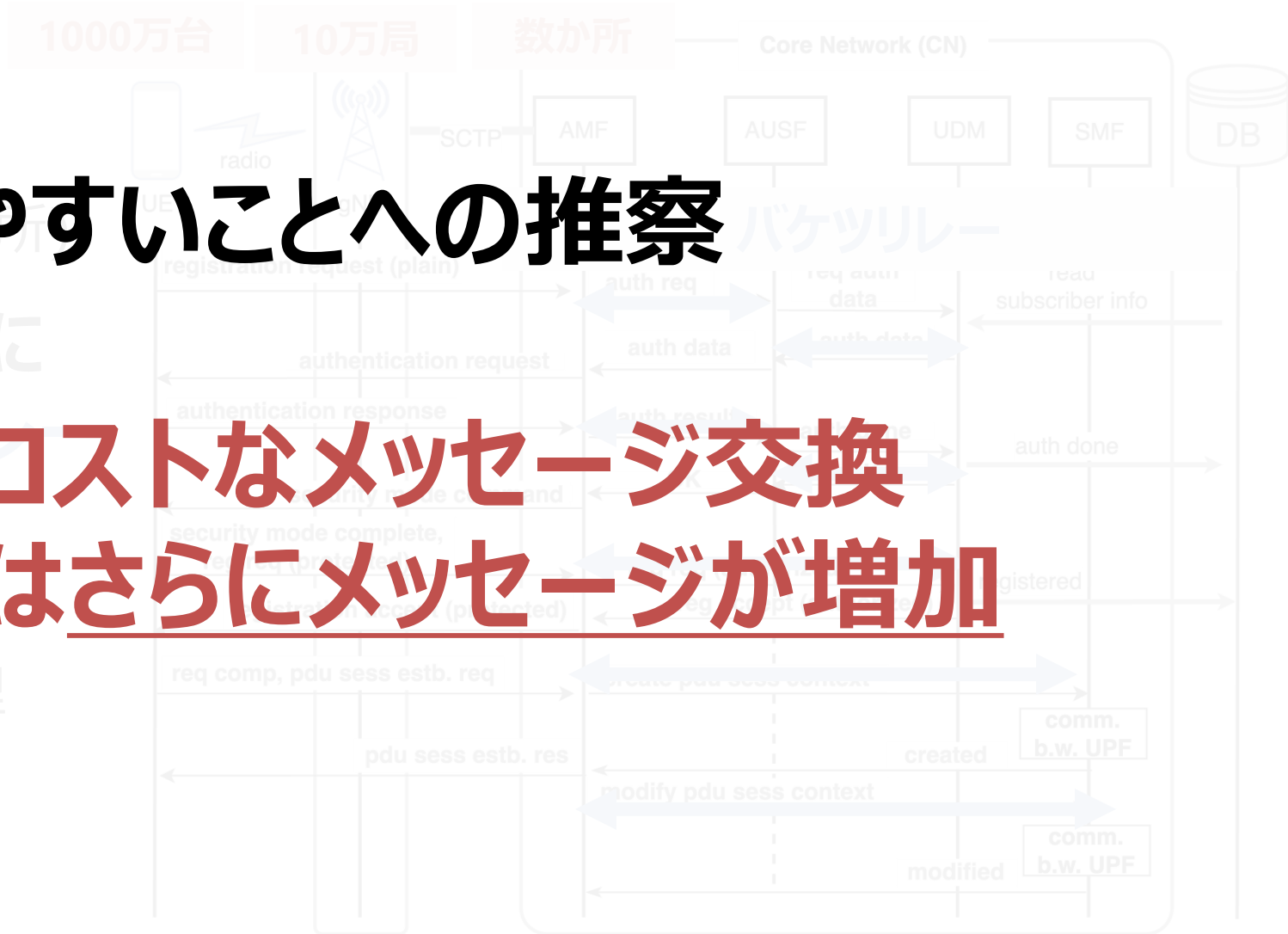


端末数 x 複雑バケツリレー = 障害の大規模化

- **AMF への過集中**
 - UE(10⁷台) : AMF(<数ヶ所)
- **1 msg 処理するために複数 NF 間でバケツリレー**
 - 各 NF は msg に対して **トランザクション処理**
 - 失敗時にはロールバック処理



端末数 x 複雑バケツリレー = 障害の大規模化



- AMF への過集中

輻輳しやすいことへの推察

- UE(10⁷台) 数ヶ所
- 1 msg 処理するために

平常時も高コストなメッセージ交換

問題発生時にはさらにメッセージが増加

- 各 NE は msg に対して
- 失敗時にはロールバック処理

障害事例：AT&T (2024/02/22)

- **きっかけ：新機能展開時のオペミス**

- 展開の3分後に不通（午前2時45分）
- 明け方までにロールバックが完了したが **Registration congestion 発生**
- Timer 制限を緩和して午後12時30分に完全回復とした（約11時間の障害）
 - ・ 午後も一部の緊急呼ができないなど
 - ・ ソース：<https://www.lightreading.com/wireless/fcc-pins-all-blame-on-at-t-for-february-s-massive-mobile-outage>

- **米国経済に 推定 5億ドル の影響を与えた**

- ・ ソース：<https://www.forrester.com/blogs/att-outage-shines-a-spotlight-on-network-dependability/>

ここまでのまとめ

- **CN** : 巨大な箱で全端末からの処理を請け負うモデル
- **5G CN は NF によるマイクロサービスアーキテクチャを採用**
 - システムを構成する機能 : NF ごとに独立 (疎結合)
 - システムを成立させるメッセージ交換 : **NF 間のバケツリレー (密結合)**
 - **1000万 端末に対してバケツリレーを実施**

ここまでのまとめ

- **CN** : 巨大な箱で全端末からの処理を請け負うモデル
- **5G CN は NF によるマイクロサービスアーキテクチャを採用**
 - システムを構成する機能 : NF ごとに独立 (疎結合)
 - システムを成立させるメッセージ交換 : **NF 間のバケツリレー (密結合)**
 - **1000万 端末に対してバケツリレーを実施**

ある NF の機能を開発・動作のために別の NF が必要
→ **プロセスがたくさんあるモノリス (分散モノリス)**

ここまでのまとめ

モバイルコアというアプリのユースケース

- CN：巨大な相対全端末からの処理を請け負うモデル
 - 5G CN は **を支えるシステム設計として**
 - システムを構成する機能：NF ごとに独立（疎結合）
 - システムの単位はNFではなく、NF間の連携（疎結合）
 - 1000万 端末に対してバケツリレーを実施
- あまり親和しなかったのでは？**
(マイクロサービスアーキテクチャが悪いわけではない)

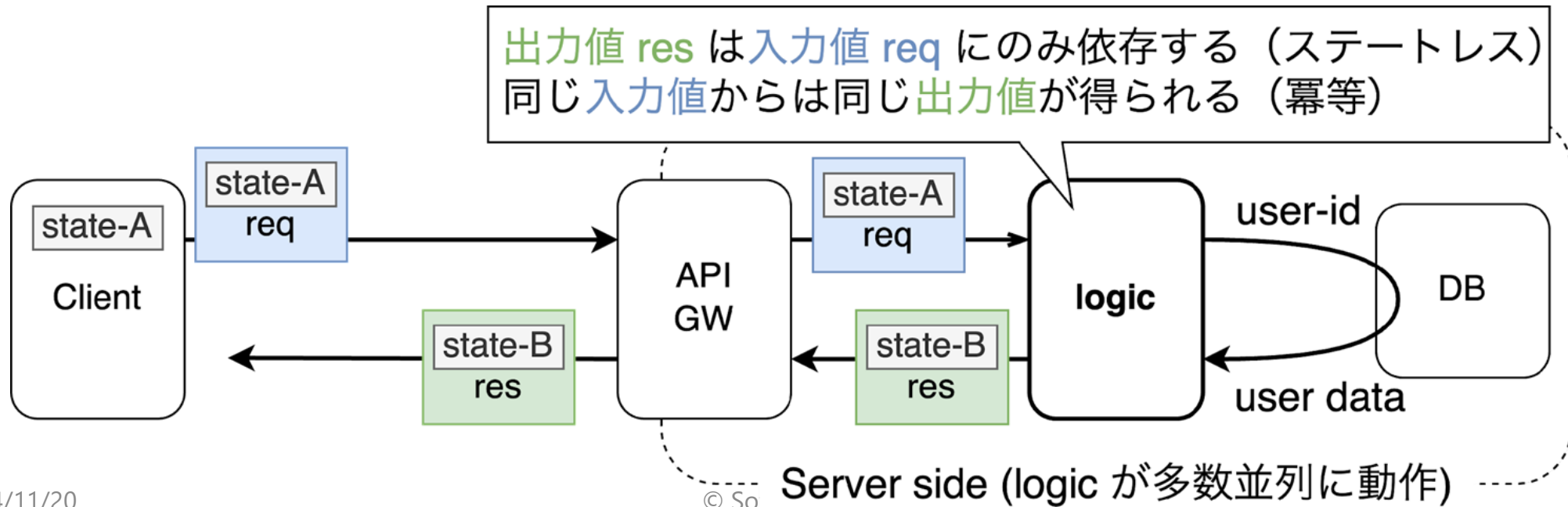
比較：EC サイト（大規模 WEB アプリ）

- いわゆる “クラウドネイティブ” なアプリケーションの一例
 - 利用者数…アリババ EC サイト: 7億7900万人（2020年） [2]
 - 「独身の日」のイベントセールでは驚異のダウンタイムゼロ（2020年） [3]
 - ピーク時には 1億4,000万クエリ/秒
- クライアント・サーバ型のアプリケーション
 - Req-Res 型のメッセージングが中心（モバイルコアと同じ！）

大規模 WEB アプリと モバイルコア とでは
何が違うのか？

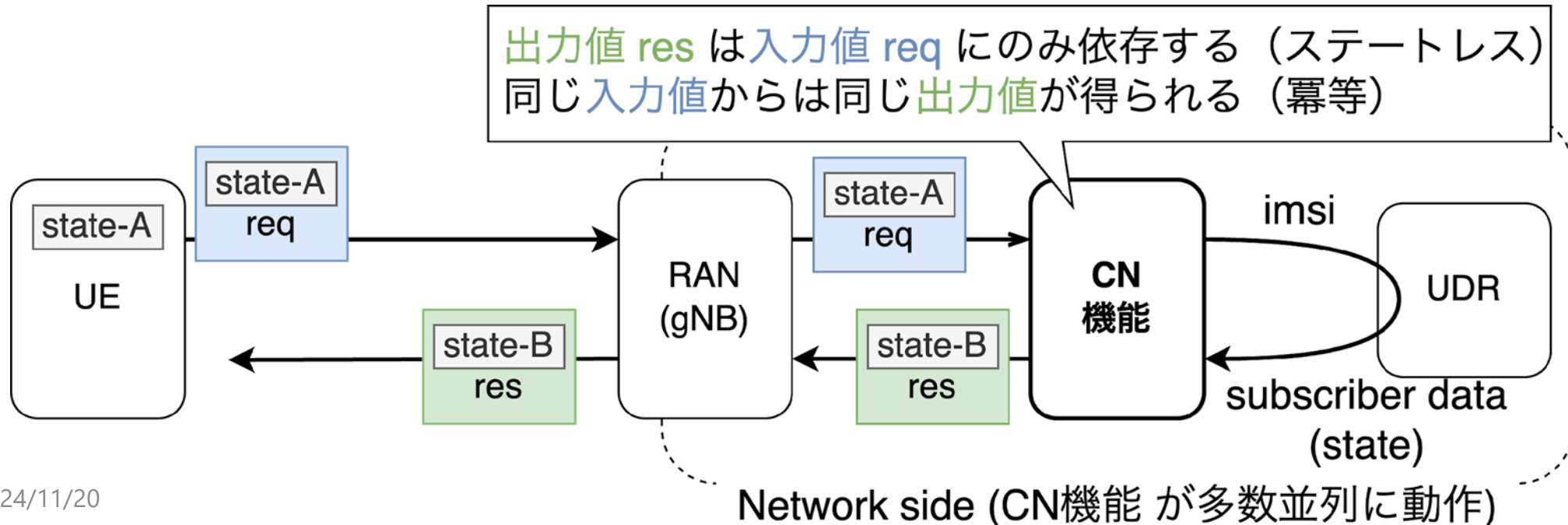
大規模 WEB アプリのシステム設計（一例）

- **メッセージハンドラがメッセージ到着ごとに並列に動作**
 - クライアントのステートはメッセージに格納
 - 入力メッセージと DB の情報にのみ依存して処理・出力メッセージの返却



モバイルシステムに適用すると...？

- **CN 機能がメッセージ到着ごとに並列に呼び出されるのみ**
 - 極端に言えば 1 msg ごとに 1メッセージハンドラの実装
 - 現行はそうならない (現行：CN 機能間のバケツリレーの応酬)



モバイルシステムに適用すると...？

- CN 機能がメッセージ到着ごとに並列に呼び出されるのみ

WEB アプリのように動作するモバイルコアをつくってみた

- 現行はそうならない (現行: CN 機能間のバケツリレーの応酬)

メッセージ到着ごとにメッセージハンドラが並列実行

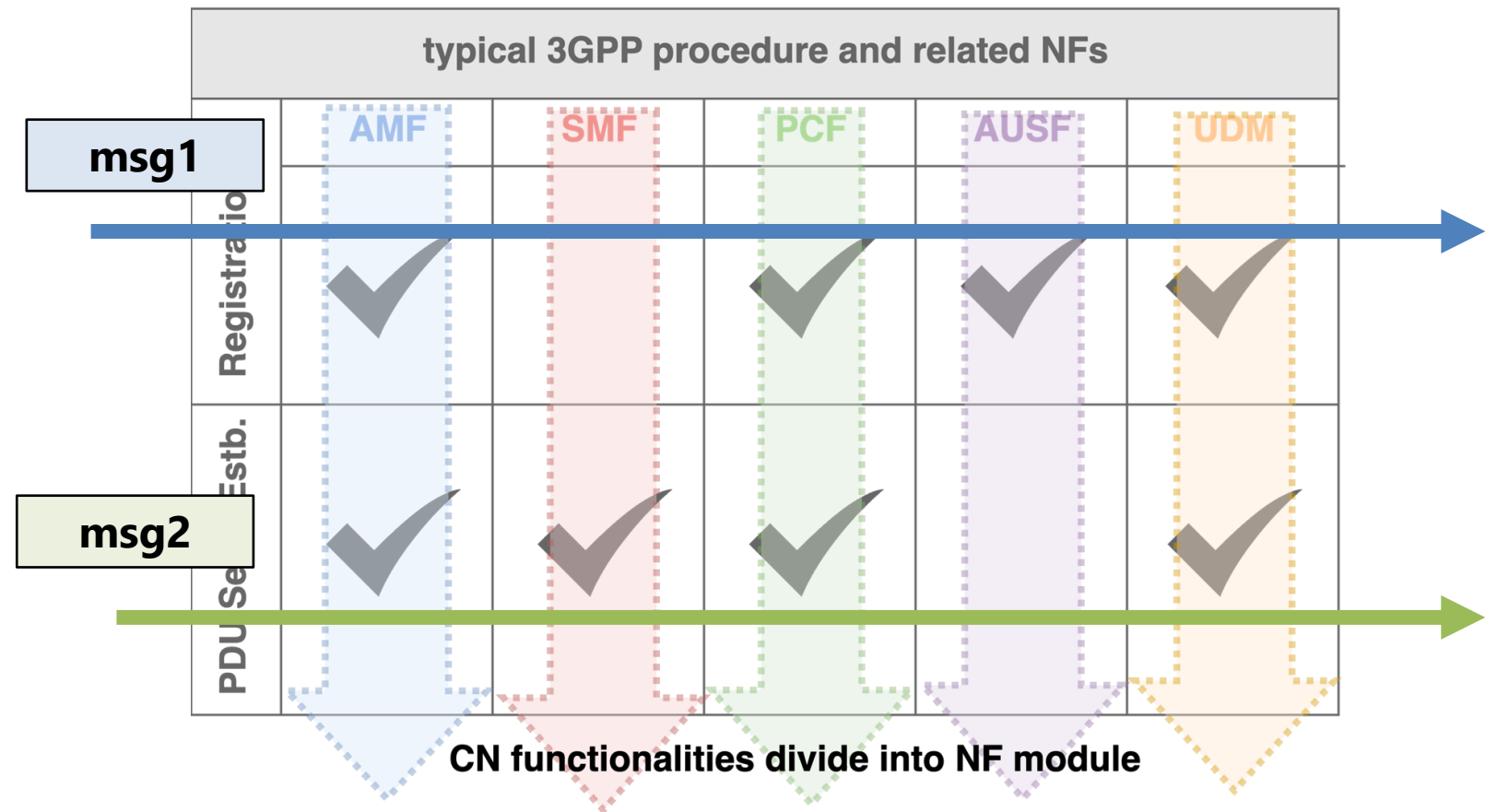
バケツリレーなし



5G CN の機能の分割方法の見直し

- 現在：NF という機能ブロック単位で実装・提供

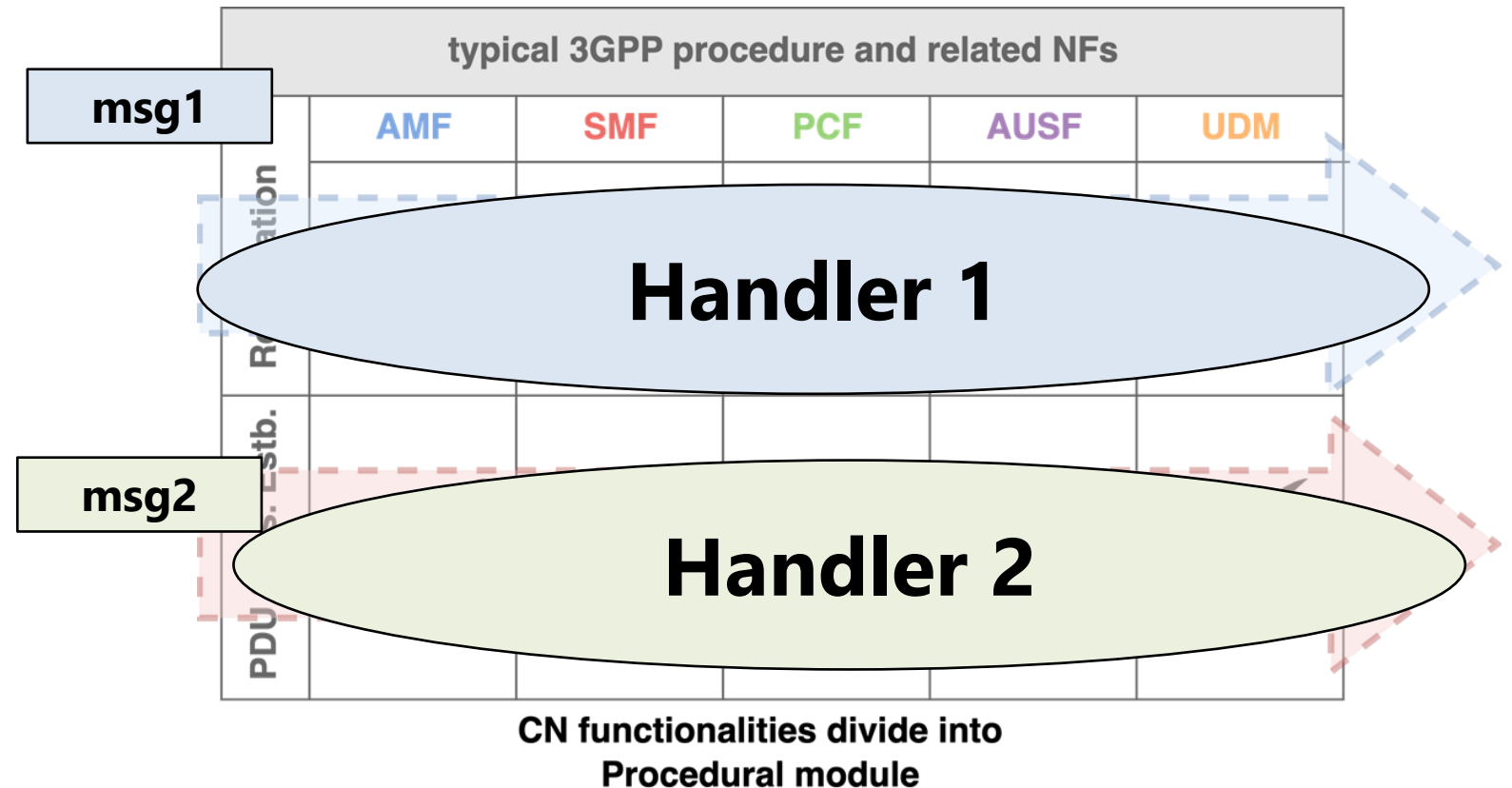
メッセージ種別ごとに
チェックが付いた NF
のバケツリレーが必要



発想の転換

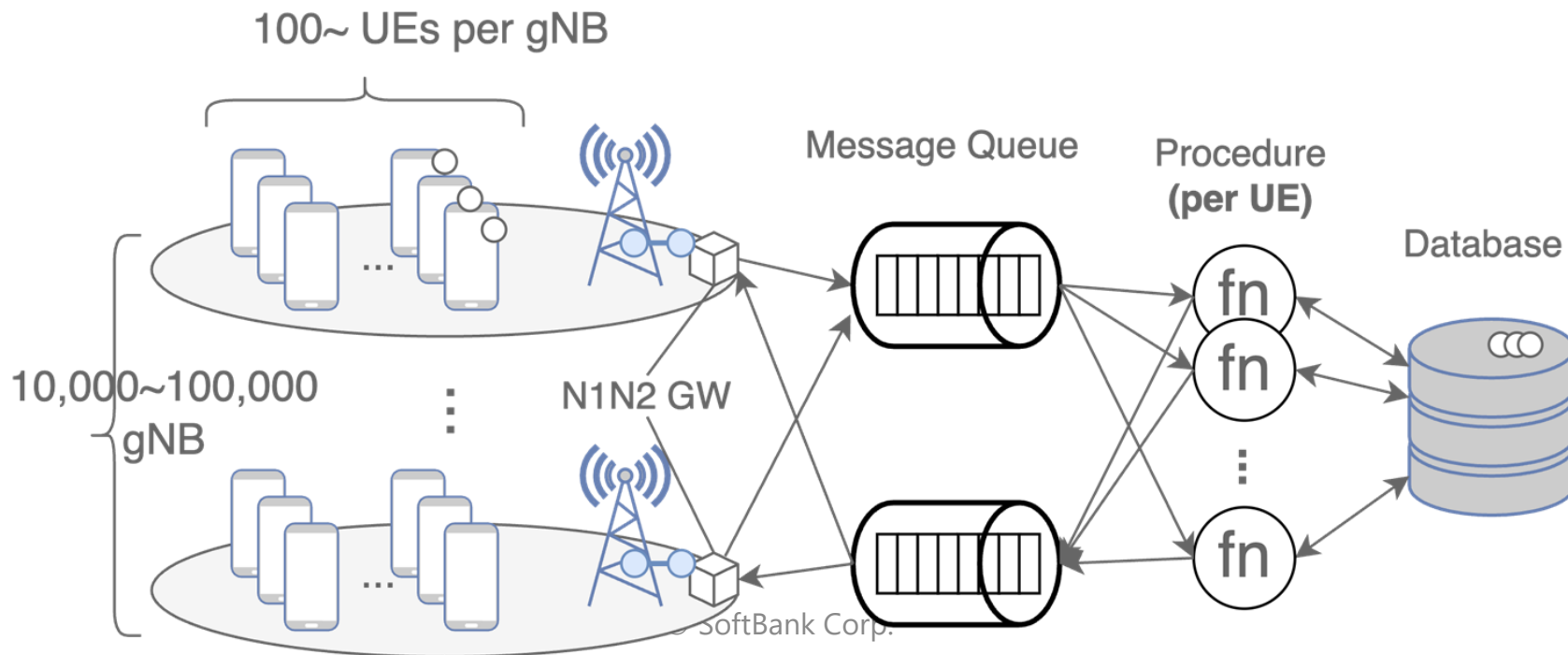
- メッセージ種別ごとにハンドラを実装・提供

メッセージ処理時に
バケツリレー不要



サーバレスな 5G コアネットワーク^{[4][5]}

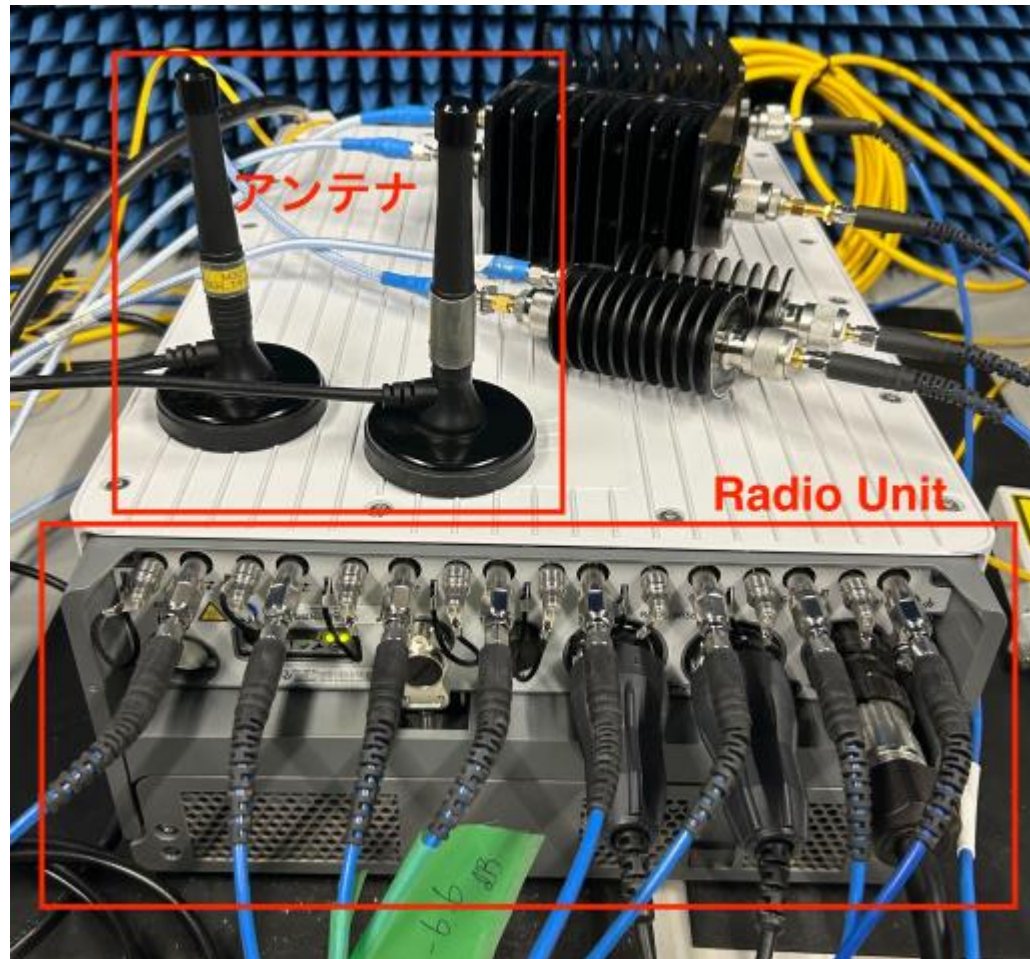
- **メッセージ到着ごとに CN 機能(fn)をインスタンス化**
 - 各 fn はステートレスでリアクティブに実行可能
 - Knative / AWS Lambda のような FaaS 基盤を用いて動作済み^{[4][5]}



実機を用いた動作検証（データ通信成功）

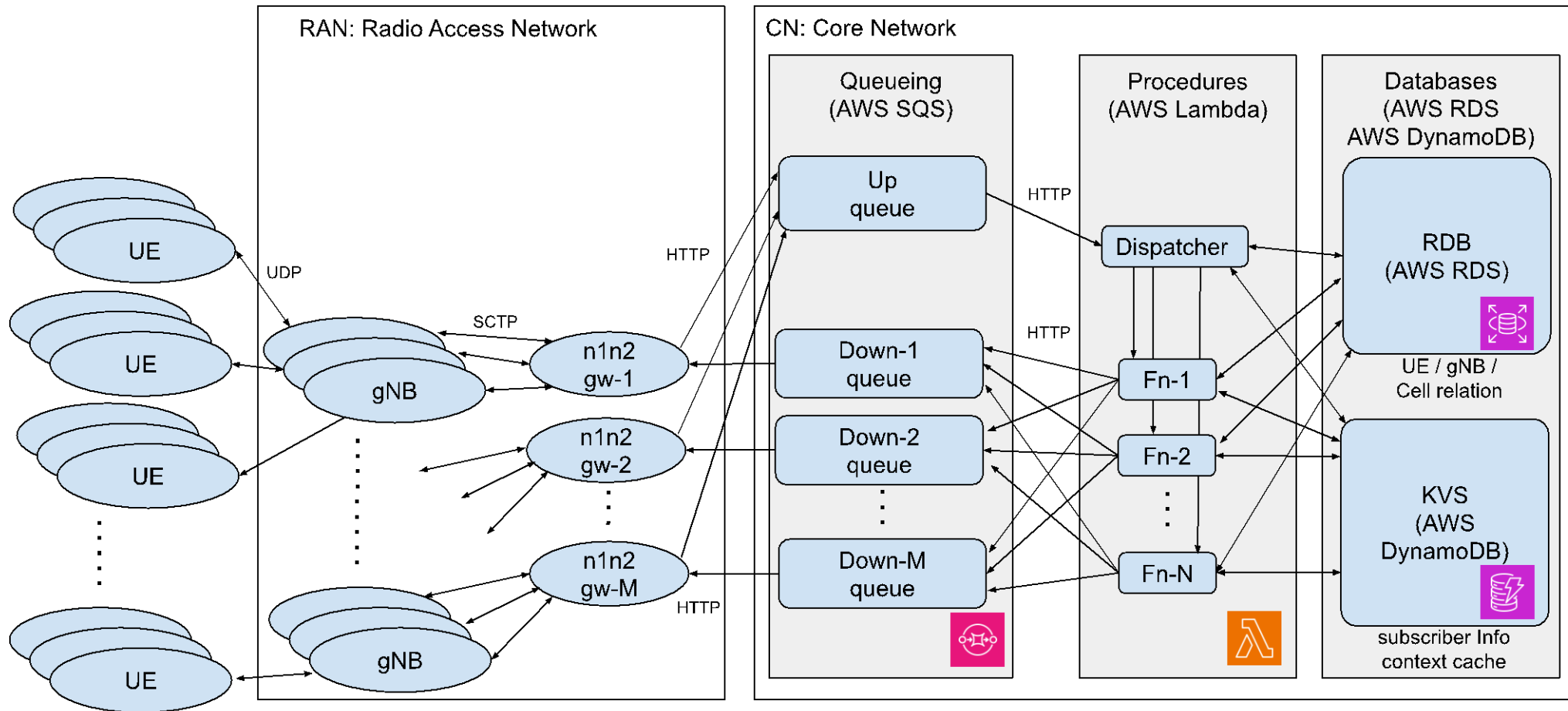


Release-15 対応 UE
京セラ



Release-15 対応 RAN
Ericsson (画像は RU とアンテナのみ、ほかに Baseband Unit)
© SoftBank Corp.

AWS Lambda 上にデプロイ



性能

当日のみ

引用文献

K. Akashi et al., "**Cloud5GC : Design and implementation of scalable and stateless mobile core system on public cloud,**" *2024 International Conference on Information Networking (ICOIN)*, <https://ieeexplore.ieee.org/document/10572149>

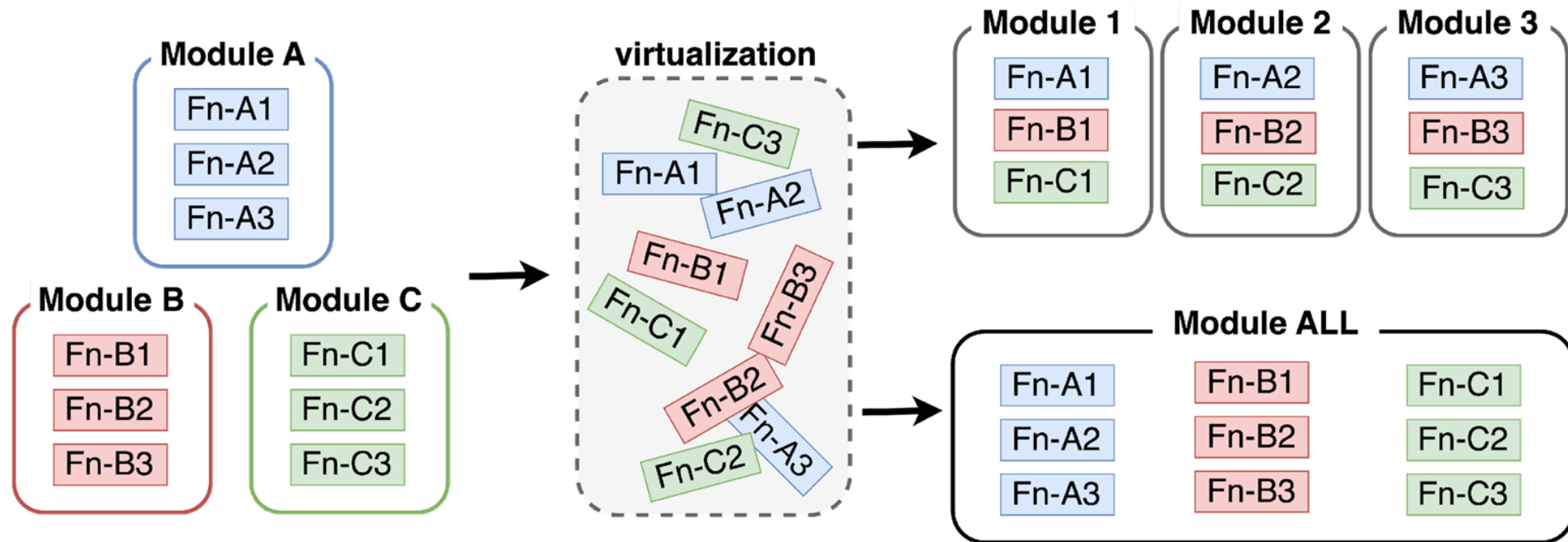
やったこと

FaaS の Function プログラムをつくるのみ

K8s の設定、オートスケール、並列化 などはしていない
そういうコンピューティング的に難しい部分は
クラウド（コンピューティング専門家）に乗る

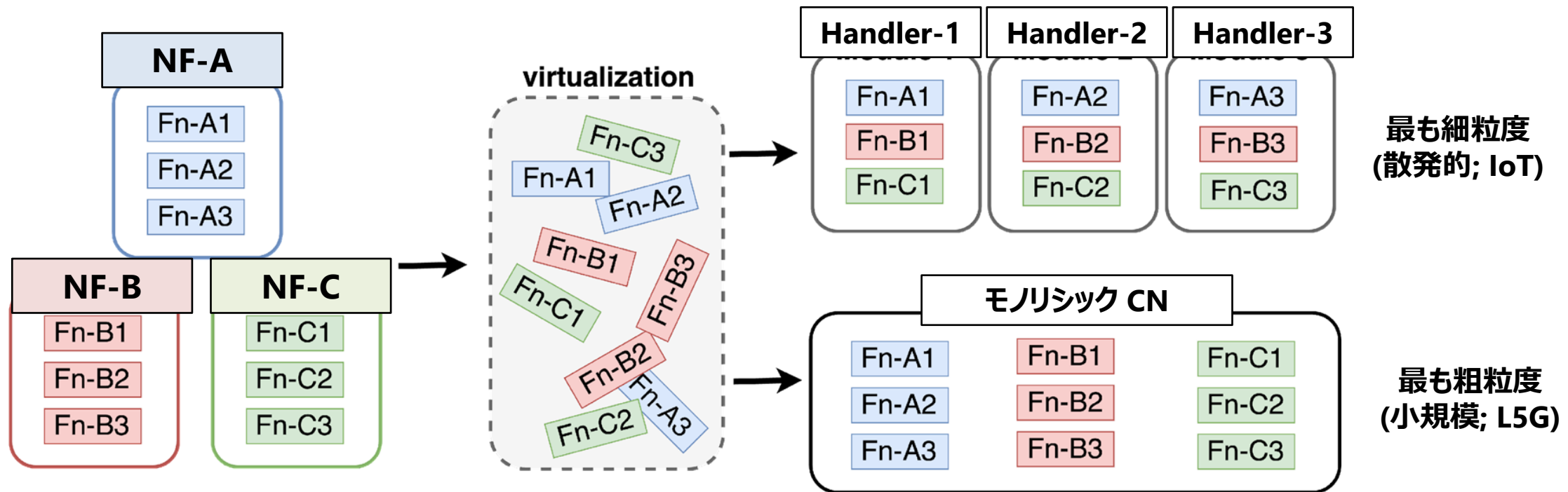
学び：通信機能のソフトウェア化（仮想化）の本質

- システムの構成要素の境界を自由に分解・再構成できること
 - 専用ハードウェア装置群と同じ境界を維持してソフトウェア化しただけでは効果薄



学び：通信機能のソフトウェア化（仮想化）の本質

- システムの構成要素の境界を自由に分解・再構成できること
 - 専用ハードウェア装置群と同じ境界を維持してソフトウェア化しただけでは効果薄



まとめ

- **現代の通信システムはコンピューティングで成立している**
 - 機能群はソフトウェア実装が進む
- **機能群をユースケースに応じて分解・再構成できる**
 - 機能群がソフトウェアで実現される最大の恩恵（仮想化の本質）
- **ユースケースを分析してシステムアーキテクチャを採用したい**
 - コンシューマ向け、IoT 向け、Local 5G 向け
 - それぞれに適したシステムアーキテクチャを選択するのが良い

参考文献

- [1]. 3GPP, System architecture for the 5G System (5GS). Technical Specification (TS) 23.501
- [2] 日本経済新聞、アリババ、中国のEC利用者数で首位陥落、
<https://www.nikkei.com/article/DGXZQOGM167S00W1A310C2000000/>
- [3] Alibaba Japan. アリババクラウド、世界最大のオンライン・ショッピング・フェスティバルを成功に導いた最新技術を公開. <https://www.alibaba.co.jp/news/2020/11/post28.html>
- [4] H. Watanabe and et al. "A Design of Stateless 5G Core Network with Procedural Processing." In *Proc. of IEEE BlackSeaCom 2023*
- [5] K. Akashi and et al. "Cloud5GC : Design and implementation of scalable and stateless mobile core system on public cloud." In *Proc. of IEEE ICOIN 2024*



R&D

SOFTBANK CORP.
RESEARCH INSTITUTE OF
ADVANCED TECHNOLOGY