Panel Discussion
コンテナ／マイクロサービス／サーバーレス 環境とクラウド責任共有モデル

Cloud Security Alliance
Application Containers and Microservices WG

モデレータ

- 笹原英司、アプリケーションコンテナ／マイクロサービス WG リーダー

パネリスト

- 野原峰彦氏
  アプリケーションコンテナ／マイクロサービスワーキンググループ
- 井出寛子氏
  健康医療情報管理ワーキンググループ
- 三浦貢造氏
  コンテナセキュリティ、マイクロサービスセキュリティ翻訳メンバー

AGENDA

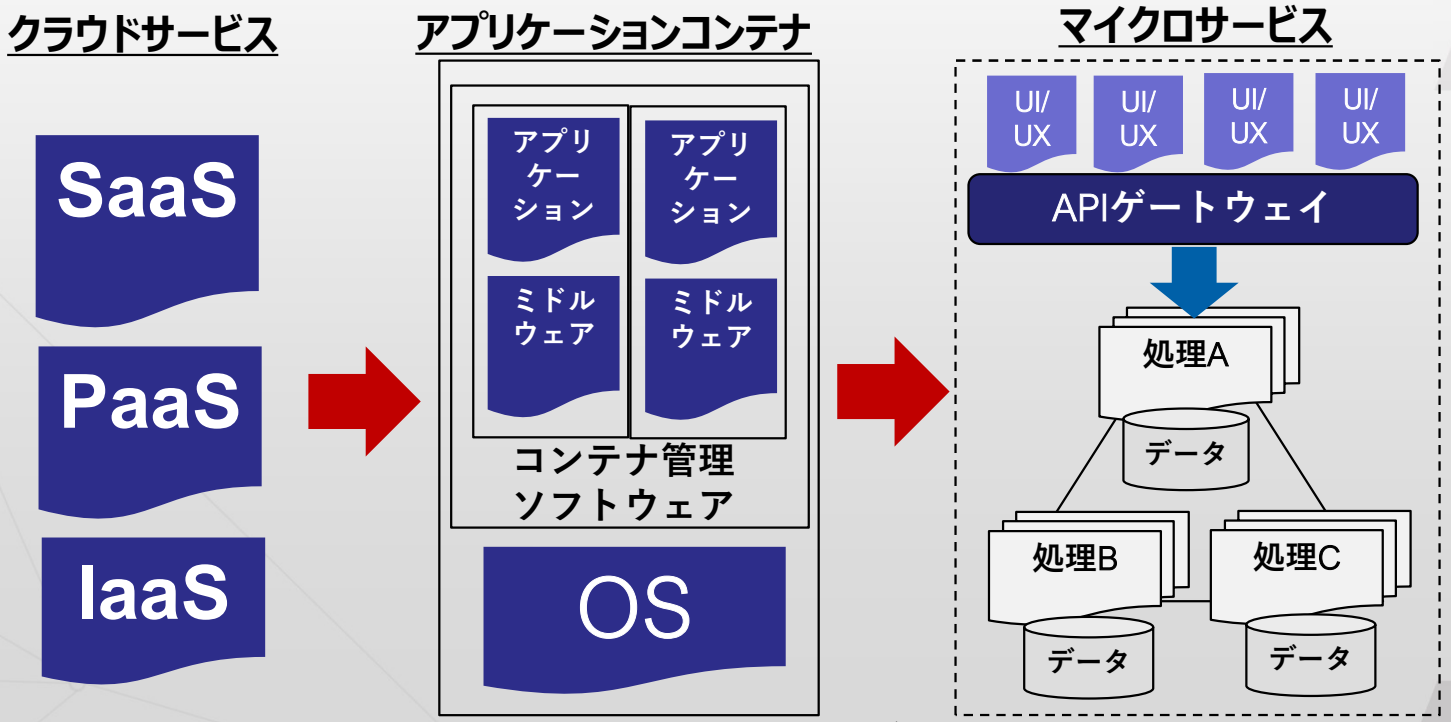
- 1. クラウドセキュリティを取り巻く新用語
  - アプリケーションコンテナ／マイクロサービスとは？
  - ゼロトラストとは？
  - サーバーレスとは？
  - DevOpsとは？

アプリケーションコンテナ／マイクロサービスとは？(1)

- 米国立標準技術研究所（NIST）「*SP 800-180(Draft): NIST* マイクロサービス、アプリケーションコンテナ、システム仮想マシンの定義」（2016年2月）
  - アプリケーションコンテナ:
    共有OS上で稼働するアプリケーションまたはそのコンポーネントとしてパッケージ化し、稼働するように設計された構成物
  - マイクロサービス：
    アプリケーション・コンポーネントのアーキテクチャを、疎結合のパターンに分解した結果生まれる基本的要素であり、標準的な通信プロトコルや明確に定義された API を利用して相互に通信する自己充足型サービスを構成し、いかなるベンダー、製品、技術からも独立している

アプリケーションコンテナ／マイクロサービスとは？(2)

- クラウドからアプリケーションコンテナ、マイクロサービスへの移行
  - ●APIを軸とする自律サービスの疎結合型アーキテクチャへ



出典：ヘルスケアクラウド研究会（2016年1月)

アプリケーションコンテナ／マイクロサービスとは？(3)

- CSA「クラウドコンピューティングのためのセキュリティガイダンス*v4.0*」（2017年7月）
  - Domain 8：仮想化とコンテナ技術
    - 8.1.4 コンテナ
      ＝移植性の高いコード実行環境

      [主要コンポーネント]
      - 実行環境
      - 統合管理とスケジューリングのコントローラ
      - コンテナイメージまたは実行するコードのリポジトリ

アプリケーションコンテナ／マイクロサービスとは？(4)

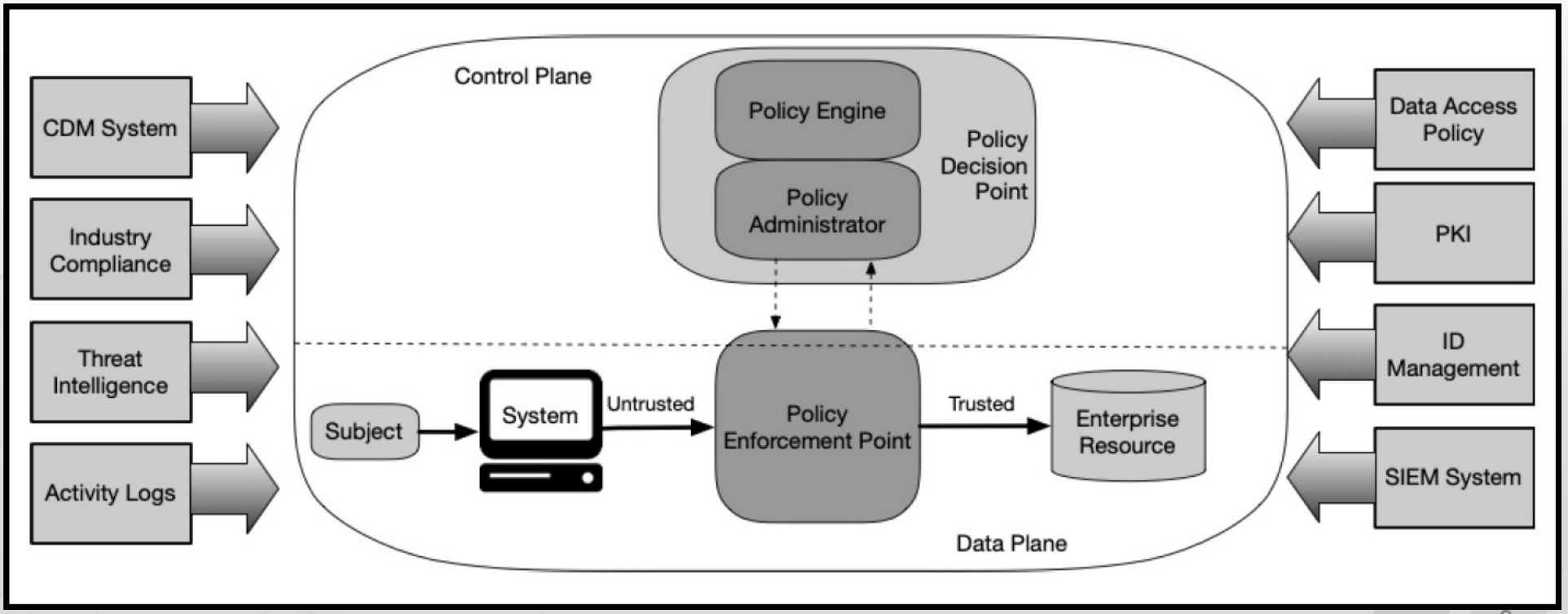
- コンテナのセキュリティ要件
  - 利用するコンテナプラットフォームとその下のOSのセキュリティのための隔離機能を把握し、適切な設定を選択すること
  - コンテナ間の隔離の実施には物理マシンまたは仮想マシンを用い、同一の物理／仮想ホスト上の同一のセキュリティ要件のコンテナはグループ化すること
  - 配備対象となるのは、確実に、承認済みで認知済みでセキュアなコンテナのイメージかコードだけとなるようにすること
  - コンテナの統合化・管理およびスケジューラのソフトウェアのセキュリティを適切に設定すること
  - 全てのコンテナとリポジトリ管理に対して、適切なロールベースのアクセス管理と、強度の高い認証を実装すること。

ゼロトラストとは?(1)

- 米国立標準技術研究所（NIST）「*SP 800-207*：ゼロトラスト・アーキテクチャ」（2020年4月19日）
  - ゼロトラスト（ZT)＝危険に晒されたと見なされるネットワークに直面した情報システムおよびサービスにおいて、正確な、特権の少ない、あらかじめ要求されたアクセスに関する意思決定を強制する際に、不確実性を最小化するように設計された概念やアイデアの集合
  - ゼロトラスト・アーキテクチャ（ZTA)＝ゼロトラストの概念を活用して、コンポーネントの関係、ワークフロー計画、アクセスポリシーを包含した、エンタープライズのサイバーセキュリティ計画
  - ゼロトラストエンタープライズ：ゼロトラストアーキテクチャ計画のプロダクトとして、エンタープライズに配置されるネットワークインフラストラクチャ（物理的および仮想的）および業務ポリシー

ゼロトラストとは?(2)

- ゼロトラストアーキテクチャの論理的コンポーネント：
  - コントロールプレーン
  - データプレーン



出典：NIST「SP 800-207 Zero Trust Architecture」（2020年4月11日）

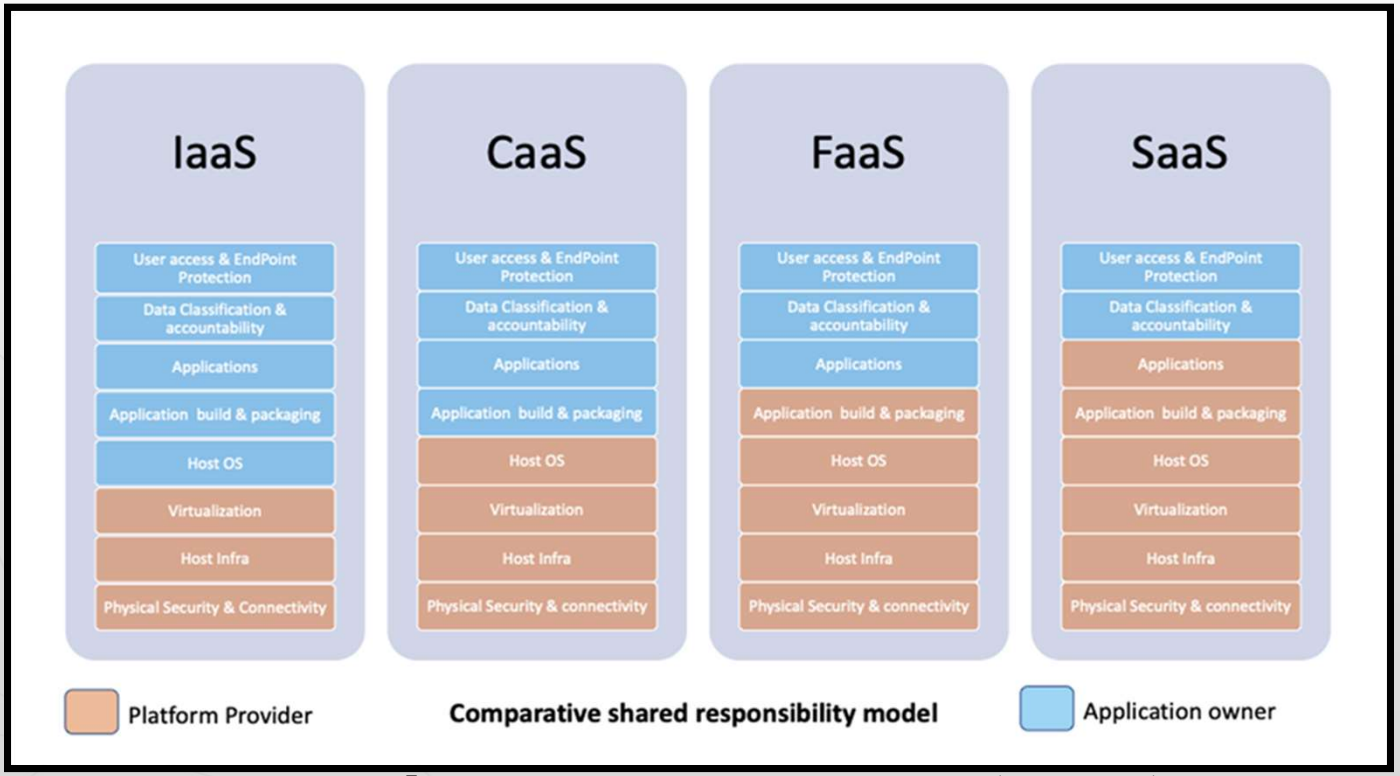
サーバーレスとは？(1)

- CSA「2020年サーバーレスコンピューティング・セキュリティ」（2020年12月公開予定）
  - サーバーレスコンピューティング:
    クラウドプロバイダーが計算処理のランタイム管理面の負荷を軽減し、計算処理、ストレージ、ネットワークに関するすべての面を含む、物理的または仮想的なマシンリソースの割当設定を動的に管理する、クラウドコンピューティング展開モデル。サーバーレスアプリケーションのオーナーは、このようなタスクを管理する必要はない。
    - Amazon: Lambda、Fargate、AWS Batch
    - Google: Cloud Functions、Knative、Cloud Run
    - Microsoft: Azure Functions、Azure Container Instances
    - Nimbella: OpenWhisk
    - IBM: OpenWhis　など

サーバーレスとは？(2)

・サーバーレスコンピューティングの責任共有モデル

- CaaS（Container as a Service）
- FaaS（Function as a Service）

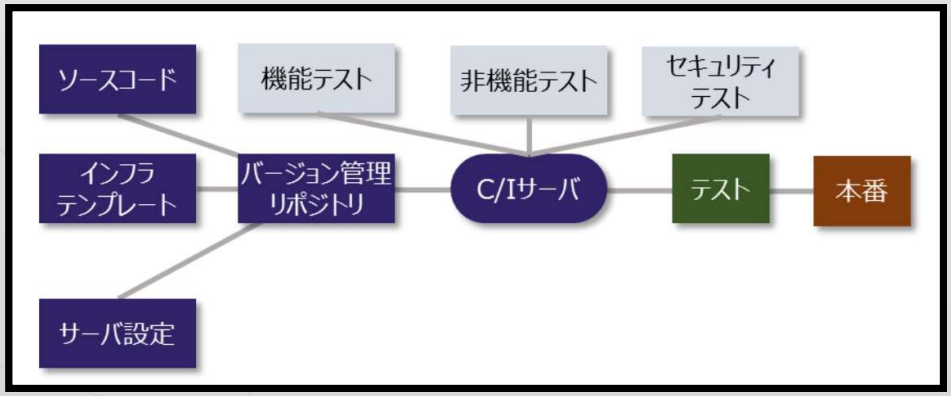


出典：CSA「Serverless Computing Security in 2020」（in progress）

DevOpsとは？(1)

- アプリケーションの開発と配備を自動化することにフォーカスした、アプリケーション開発の新しい方法論であり考え方である
- 開発チームと運用チームの間の協力とコミュニケーションを改善してより深く結びつけることを意味し、特にアプリケーション配備とインフラストラクチャ運用の自動化に焦点を当てている
- コード堅牢化、変更管理、本番アプリケーションのセキュリティを改善するだけでなくセキュリティ運用全般をも強化してくれる

継続的インテグレーション／継続的デプロイ（CI/CD）パイプライン



出典：日本クラウドセキュリティアライアンス「クラウドコンピューティングのためのセキュリティガイダンス v4.0」日本語版1.1（2017年7月）

DevOpsとは？(2)

- DevOpsのセキュリティへの波及効果

項目	波及効果と長所
標準化	DevOps では、本番に組み込まれるものはすべて、承認済みのコードと設定用テンプレートに基づき、継続的インテグレーション／継続的デプロイ（CI/CD）パイプラインによって生み出される。開発、テスト、本番（のコード）はすべて完全に 同一のソースファイルから派生しており、周知となっている優れた標準からの逸脱を防いでいる。
自動化されたテスト	広範な種類のセキュリティテストは、必要に応じて補助的に手動 テストを加えることで、CI/CD パイプラインに組み込むことが可能である。
不可変性(immutable)	CI/CD パイプラインは、素早く確実に、仮想マシンやコンテナ、インフラストラクチャスタックのマスターイメージを生成する。これにより配備の自動化と不可変(immutable)なインフラストラクチャを実現する。
監査と変更管理の改善	CI/CD パイプラインはソースファイルにある 1 文字の変更に至るまでの全て を追跡調査できる。バージョン管理リポジトリに格納され たアプリケーションスタック（インフラストラクチャを含む）の全履歴と共に、その変更は変更を行った人物と紐づけられる。
SecDevOps/DevSecOps と Rugged DevOps	SecDevOps/DevSecOps は セキュリティ運用を改善するために DevOps の自動化技術を使う。Rugged DevOps はアプリケーション開発過程にセキュリティテスティングを組み入れることを意味し、よ り強固で、よりセキュアで、より障害耐性の高いアプリケーションを生み出す。

出典：日本クラウドセキュリティアライアンス「クラウドコンピューティングのための
セキュリティガイダンス v4.0」日本語版1.1（2017年7月）

AGENDA

- 2. アプリケーションコンテナ／マイクロサービス／サーバーレスのユースケース
- 新型コロナウイルス感染症対策ソリューションのユースケース

新型コロナウイルス感染症対策ソリューションのユースケース（1）

・厚生労働省「*COCOA (COVID-19 Contact-Confirming Application)*」（2020年6月19日）

- Bluetoothを介した近接検知メカニズムと、Google／Appleが提供する暴露通知APIを利用した新型コロナウイルス感染症向け接触確認アプリケーション
- Microsoft Azureを採用

Source: The Ministry of Health, Labor, and Welfare, "(COCOA) COVID-19 Contact-Confirming Application", June 22, 2020 (https://www.mhlw.go.jp/stf/seisakunitsuite/bunya/cocoa_00138.html)

新型コロナウイルス感染症対策ソリューションのユースケース（2）

・アイルランド保健サービス委員会（HSE）の接触追跡アプリケーション「*COVID Tracker*」（2020年7月7日）

- アプリケーション・コードをオープンソース化（COVID Green）してLinuxファウンデーションに寄贈 ⇒ AWS Lambdaを採用

出典：Health Service Executive (HSE)「COVID Tracker App」（2020年7月7日）（https://covidtracker.gov.ie/）

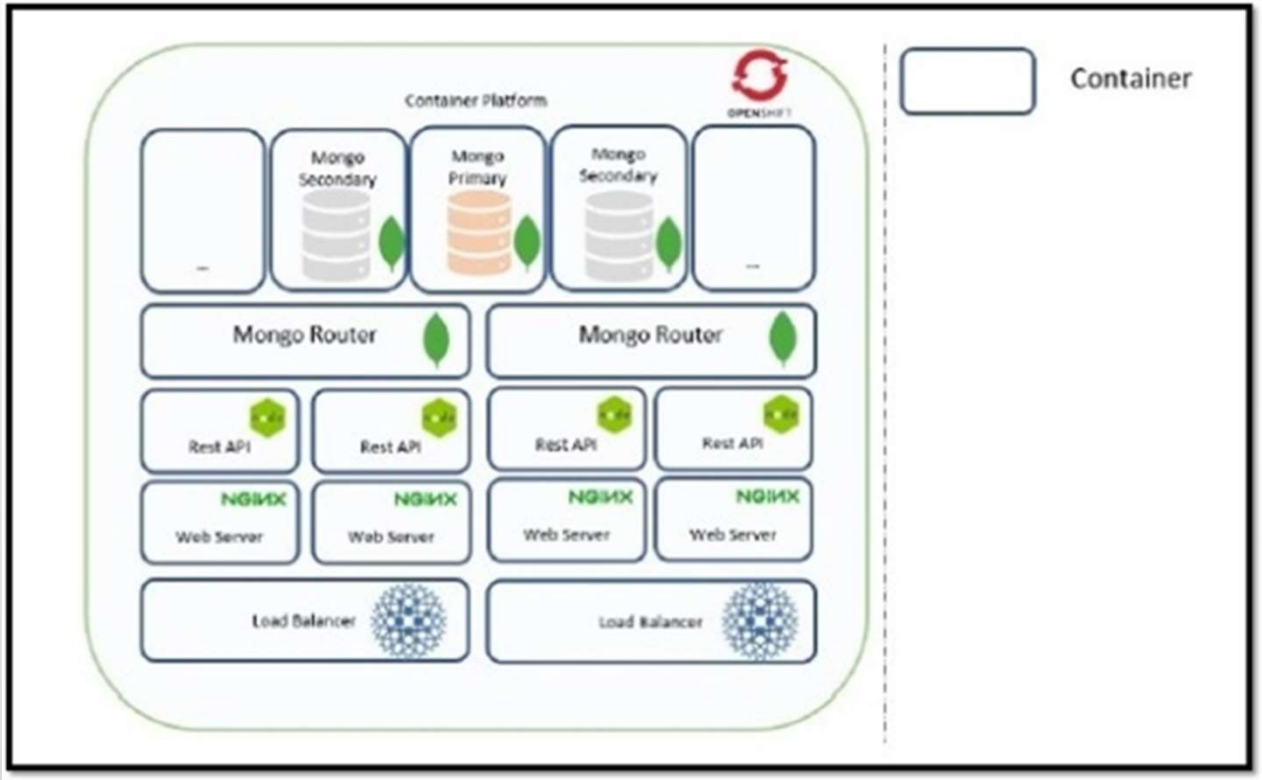
出典：github.com: COVID GREEN（https://github.com/covidgreen/covid-green-lambdas）

新型コロナウイルス感染症対策ソリューションのユースケース（3）

・欧州委員会「接触追跡アプリケーションの相互運用性に関する技術仕様*V1.0*」（2020年6月16日）　＊2020年10月より本稼働

[越境連携エコシステム導入例]

- コンテナ管理プラットフォーム
  （例：Red Hat OpenShift）
- REST API
  （例：Node.jsとExpress）
- 分散NoSQLデータベース
  （例：MongoDB）
- ロードバランサー
  （例：Docker上のHAProxy）
- Webサーバー
  （例：Docker上のNginx）



出典：European Commission「Technical specifications for interoperability of contact tracing apps - eHealth Network Guidelines to the EU Member States and the European Commission on Interoperability specifications for cross-border transmission chains between approved apps」（2020年6月16日）
(https://ec.europa.eu/health/sites/health/files/ehealth/docs/mobileapps_interoperabilitydetailedelements_en.pdf)

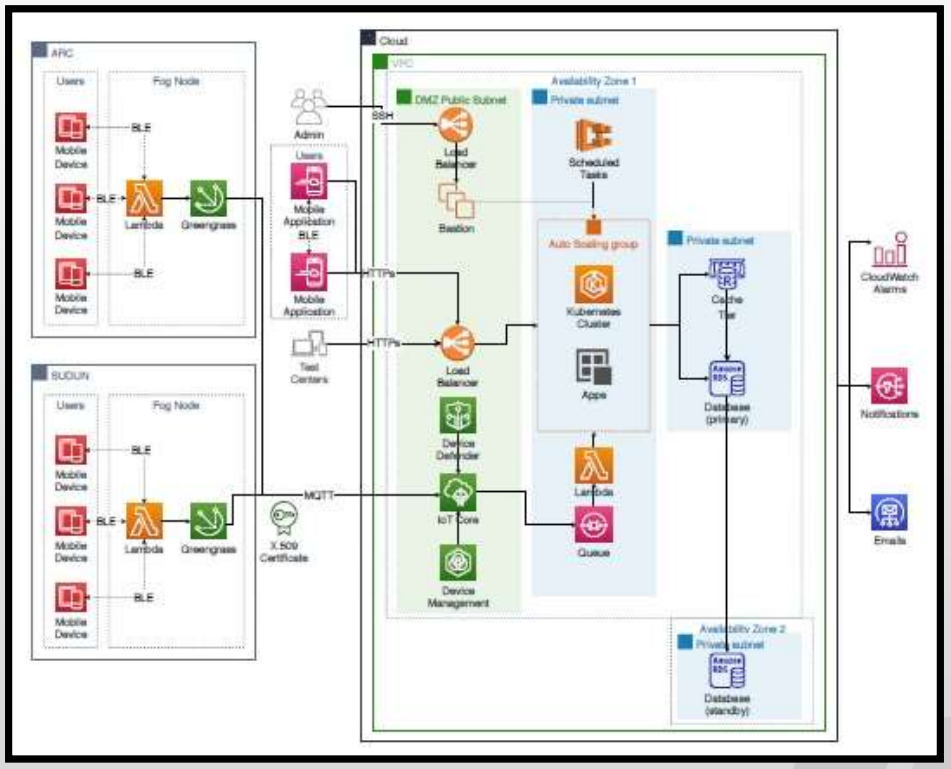
新型コロナウイルス感染症対策ソリューションのユースケース（4）

・Md Whaiduzzaman et al.「*COVID-19*地域感染追跡・予防向けプライバシー保護モバイル／フォグコンピューティングフレームワーク」（*2020*年6月）

[地域感染追跡・予防向けモバイル／フォグコンピューティングフレームワーク]

- AWS Lambda
  (サーバーレスアーキテクチャ)
- AWS IoT Greengrass
- AWS IoT Core
- AWS IoT Device Defender
- AWS IoT Device Management
- Amazon Simple Queue Service(SQS)
- Amazon Relational Database Service (RDS)

など



出典：Md Whaiduzzaman et al.「A Privacy-preserving Mobile and Fog Computing Framework to Trace and Prevent COVID-19 Community Transmission」（2020年6月）