クラウドセキュリティと
責任共有モデル

一般社団法人　日本クラウドセキュリティアライアンス

理事　諸角昌宏

CSAリサーチフェロー、 CCSP、CCSK

2021年10月26日

アジェンダ

1. Cloud Security Allianceについて
2. クラウドの責任共有モデル
3. 様々な観点からの責任共有モデル
4. まとめ

1. Cloud Security Allianceについて

Cloud Security Allianceについて

- CSA本部：　グローバルな非営利活動法人
  - 創立：2009年
  - 会員数
    - 個人会員　9万人以上
    - 地域支部　90以上(日本を含む)
    - 企業会員　400社以上
  - 33のワーキンググループと調査研究プロジェクト
  - 政府、研究機関、専門家団体、企業との戦略的パートナーシップ

- 次世代ITのための実践規範の構築
- 調査研究と普及啓発

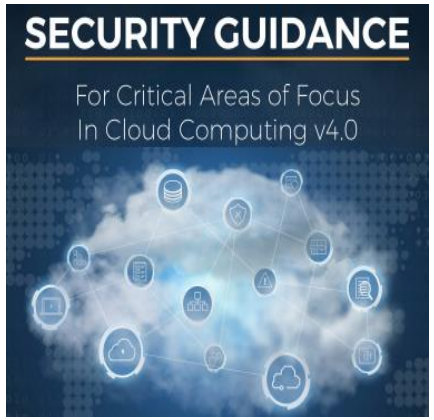
クラウドコンピューティングにおけるセキュリティ保証に向けた実践規範活用の促進と、クラウド利用のための教育を通じてあらゆるコンピュータ利用のセキュリティを高めるための活動への取組み

一般社団法人日本クラウドセキュリティアライアンス設立の趣旨・背景

- 目的
  - Cloud Security Alliance(CSA)の開発するガイドラインやツールを日本で展開・活用するための取組みを中心に活動
  - CSAの活動の活発化、世界的プレゼンスの向上
  - 各国での支部の設立、法人化の進行
  - ますます浸透するクラウドの活用とそのセキュリティ課題の重要性に対応

- 経緯
  - 2010年6月に任意団体として発足
  - 2013年12月にCSA日本支部を法人化（一般社団法人へ）
    - 日本におけるクラウドセキュリティへの取組みの中心を担うべく、法人化し、活動基盤の強化充実を図る

CSAにおけるWG活動とその相互連携



CSA STAR
Security, Trust & Assurance Registry

CCM
Cloud Controls Matrix

他の

ISO27001/ISO27010/
HIPAA/ HITECH Ac/
AICPA/COBIT/ENISA/
FedRAMP/ PCI DSS

CCSK
Certificate of
Cloud Security Knowledge

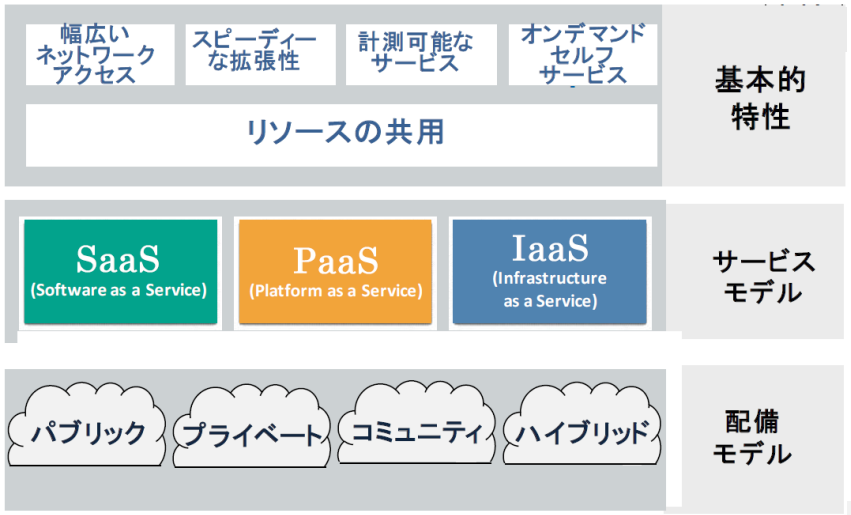
WGや研究活動

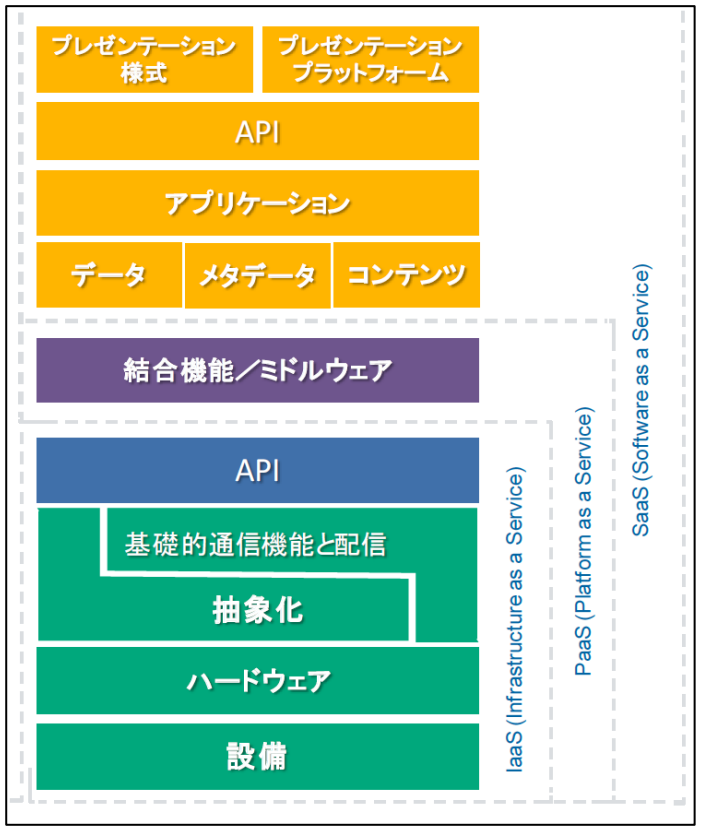
INTERNET of THINGS
Big Data
HIM
SecaaS

2.　クラウドの責任共有モデル

クラウドサービスモデル

- クラウドのセキュリティを考えるときに重要な概念
- NIST定義の採用　（SP800-145）
- SPIモデル
  - Software as a Service
  - Platform as a Service
  - Infrastructure as a Service





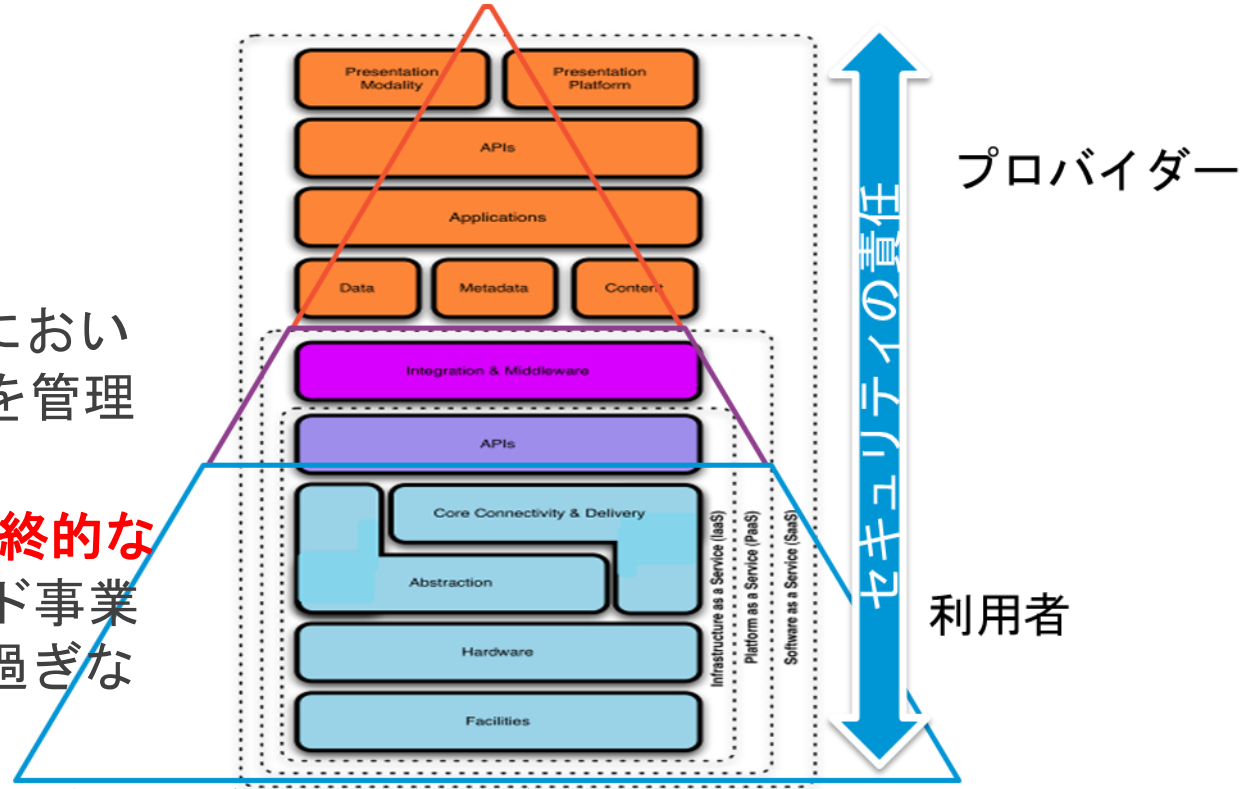
（クラウドコンピューティングのためのセキュリティガイダンス　V4.0から引用）

クラウドセキュリティの基本

責任共有モデル

- クラウド事業者は、一定のリスクに対する責任を負い、クラウド利用者はその先のすべてに責任を持つ
  - Security In the Cloud
  - Security Of the Cloud

- SaaSにおいてはクラウド事業者が、IaaSにおいてはクラウド利用者がより多くのリスクを管理する
- **クラウド利用者は、リスクを所管する最終的な責任（説明責任）を負っており**、クラウド事業者にリスク管理の一部を転嫁しているに過ぎない



（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　V3.0から引用）

責任共有の基本的な考え方

- クラウド利用によるリスクアセスメントとリスク対応方針の策定
  - ベースは組織としてのリスク管理。クラウドはその延長
- 利用者/プロバイダの責任範囲の明確化
  - 責任境界の認識。どちらも責任範囲がゼロになることはない
- 責任範囲を超えた部分の管理
  - 利用者：プロバイダに確認、SLA/契約化
デューデリジェンスが重要
注意点：　説明責任は必ず利用者側
  - プロバイダ：　セキュリティ情報の公開、透明性
  - 有効なツール
    - CCM（Cloud Control Matrix）、CAIQ（Consensus Assessment Initiative Questionnaire）
    - ISO/IEC　27017
    - 第三者監査の結果

3.　様々な観点からの責任共有モデル

責任共有の観点

以下の主な観点において責任共有とその対策について考える

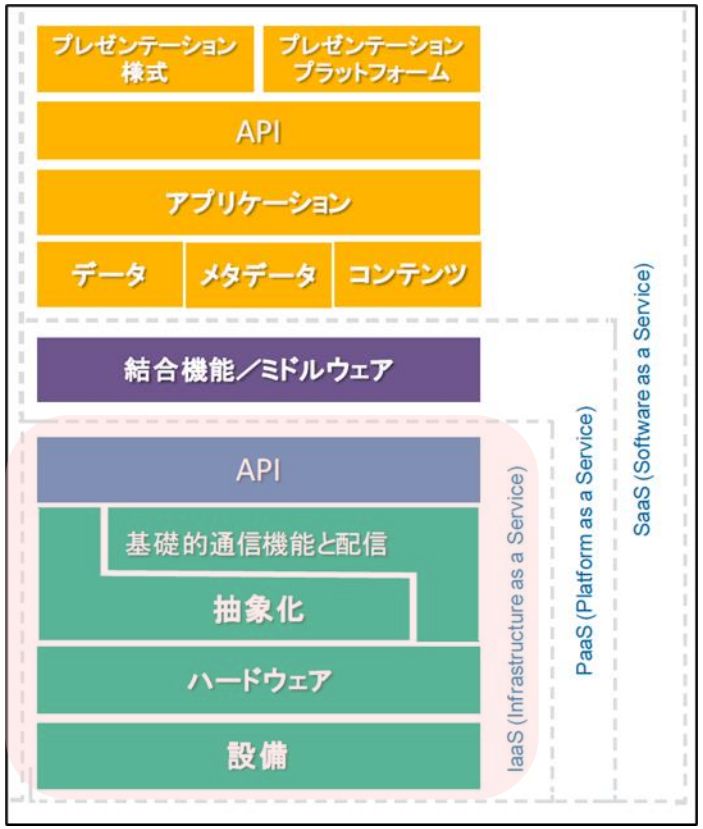
1. リスク管理、セキュリティ
2. インシデントレスポンス
3. ガバナンス、コンプライアンス、監査
4. プライバシー
5. 新しいモデル

責任共有の観点

1. リスク管理、セキュリティ
2. インシデントレスポンス
3. ガバナンス、コンプライアンス、監査
4. プライバシー
5. 新しいモデル

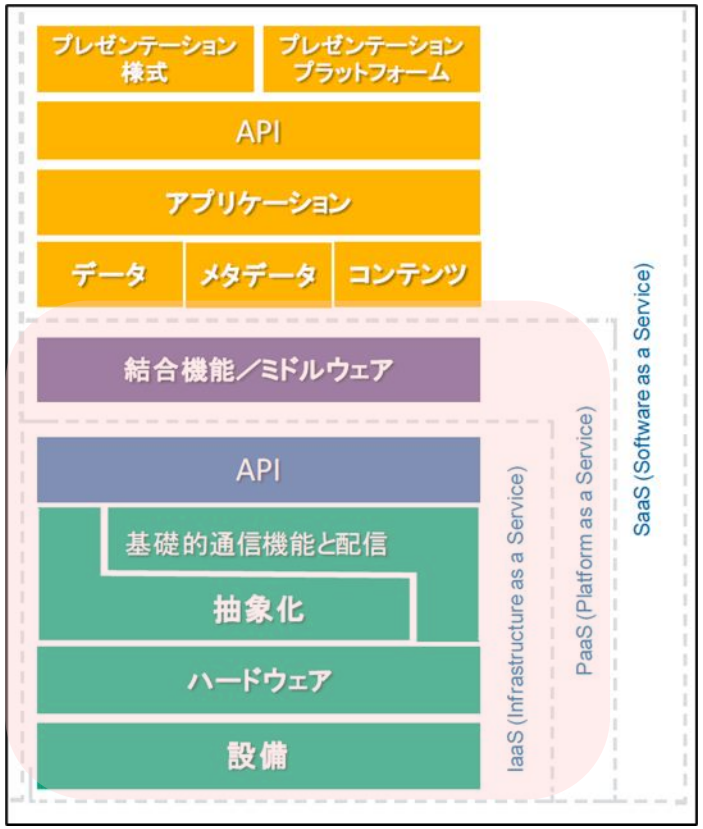
リスク管理、セキュリティにおける責任共有

- IaaS
  - プロバイダ管理責任
    - データセンター、ハードウエア、仮想化のインフラ管理
    - クラウドイベントログの管理、利用者への提供機能
    - クラウドインシデント管理、および、利用者への情報提供
  - 利用者管理責任
    - OSレベル（コンピュート、ストレージ、ネットワーク）
    - 利用者アプリケーション
    - ID/アクセス管理
    - データセキュリティ
    - 統合イベント管理機能



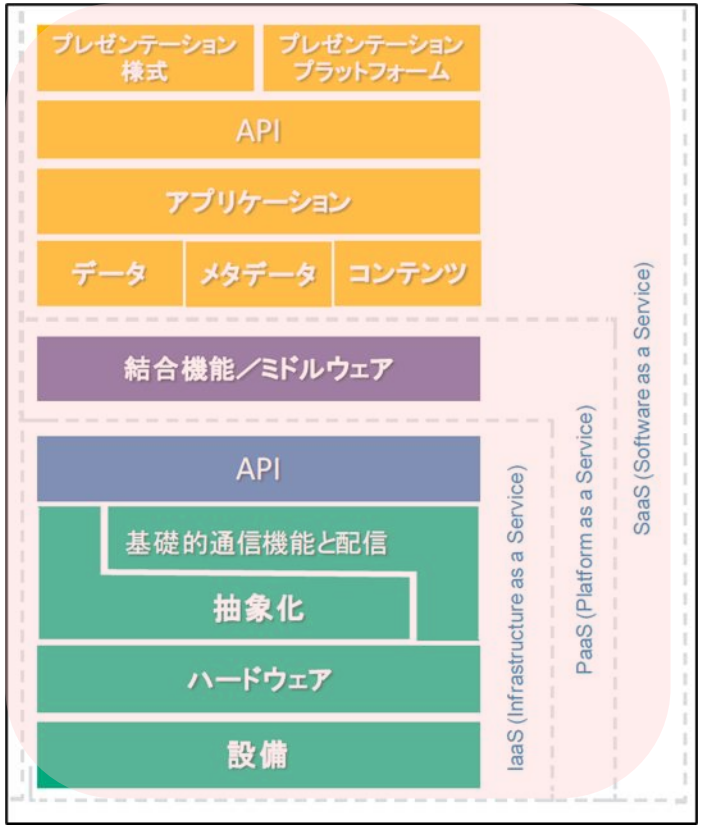
リスク管理、セキュリティにおける責任共有

- PaaS
  - プロバイダ管理責任
    - インフラ全体
    - ライブラリ、ミドルウエア
    - APIのセキュリティ
      - 入力妥当性、出力妥当性など
    - クラウドイベントログの管理、利用者への提供機能
    - クラウドインシデント管理、および、利用者への情報提供
  - 利用者管理責任
    - 利用者作成アプリケーションへのセキュリティ機能の組み込み
    - ID/アクセス管理
    - データセキュリティ
    - 統合イベント管理機能



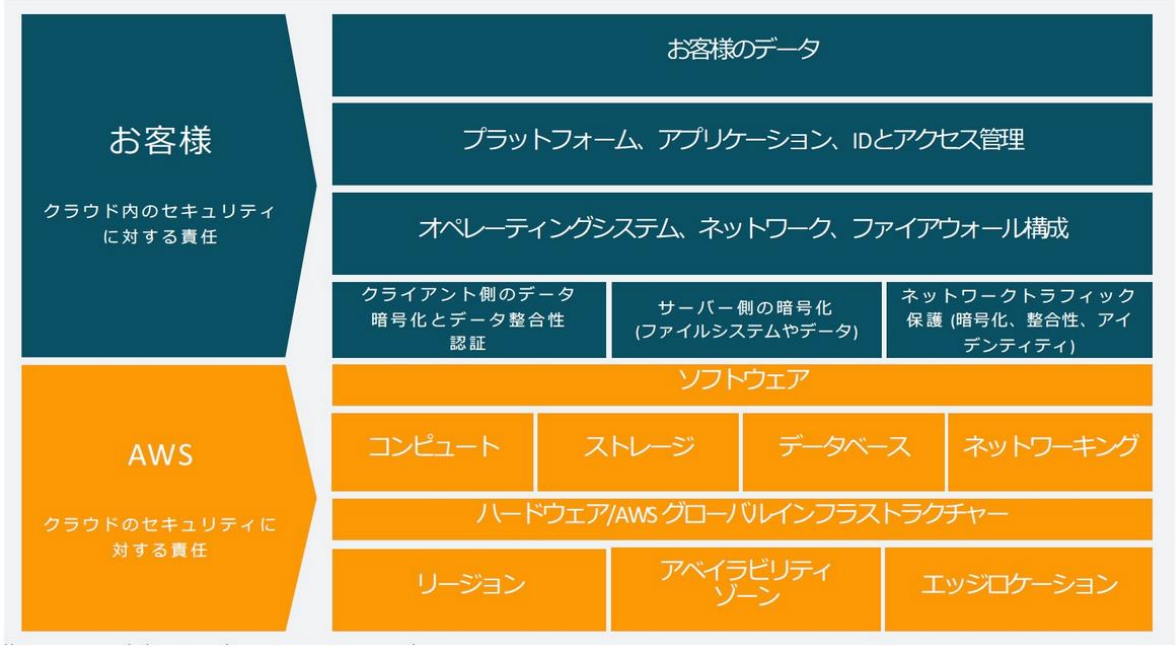
リスク管理、セキュリティにおける責任共有

- SaaS
  - プロバイダ管理責任
    - クラウド全体
    - クラウドイベントログの管理、利用者への提供機能
    - クラウドインシデント管理、および、利用者への情報提供
  - 利用者管理責任
    - ID/アクセス管理
    - データセキュリティ
    - 統合イベント管理機能



例：IaaS責任共有モデル（AWS EC2）

- AWS
  - ホストオペレーティングシステムと仮想化レイヤーから、サービスが運用されている施設の物理的なセキュリティに至るまでの要素を運用、管理、制御
- お客様
  - ゲストオペレーティングシステム (更新とセキュリティパッチを含む)、その他の関連アプリケーションソフトウェア、およびAWSが提供するセキュリティグループファイアウォールの設定に対する責任と管理
  - 使用するサービス、それらのサービスの IT 環境への統合、および適用される法律と規制によって責任が異なるため、お客様は選択したサービスを慎重に検討する必要あり



引用： https://aws.amazon.com/jp/blogs/news/rethinksharedresponsibility/

リスク管理、セキュリティにおける責任共有モデル ～　対応方法

- サービスモデルごとの対応方法
  - IaaS/PaaS:　主要なプレイヤーは限られている。利用者として、クラウドに関する標準化した要求事項を作成・明確化する
  - SaaS:　クラウドサービスごとに様々。固有の要件の明確化が必要

- デューデリジェンスは非常に重要
  - 内部デューデリジェンス
    - 組織内のセキュリティ要求事項の明確化
    - 組織のコンプライアンス要件の明確化
  - 外部デューデリジェンス
    - クラウドサービス、プロバイダが要求事項を満たしていることの評価
  - デューデリジェンスを契約前に確実に実施するとともに、定期的に実施する

責任共有の観点

1. リスク管理、セキュリティ
2. **インシデントレスポンス**
3. ガバナンス、コンプライアンス、監査
4. プライバシー
5. 新しいモデル

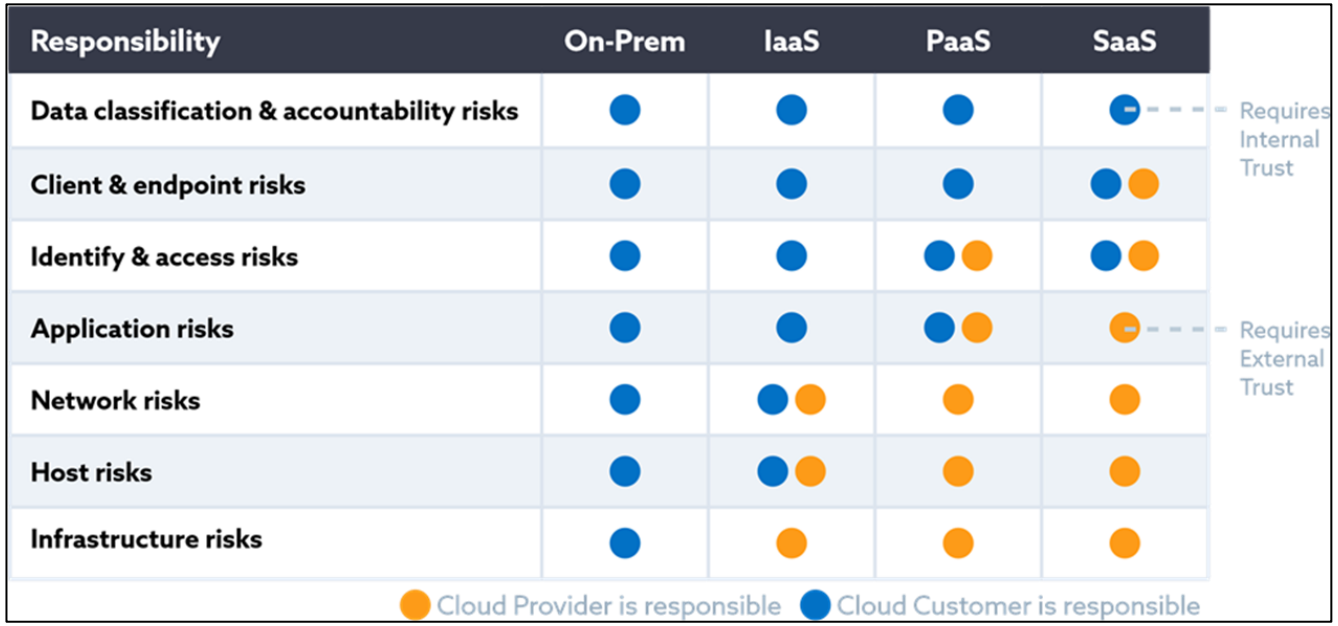
クラウド環境でインシデント対応の何が変わるか？

責任共有モデル

- インシデント対応も責任共有の影響を受ける
- ただし、説明責任は利用者側に残る

クラウドインシデント対応の基本

1. プロバイダが生成するログの取得・管理
2. プロバイダのインシデント対応能力の確認
3. インシデントの通知・連絡方法の合意



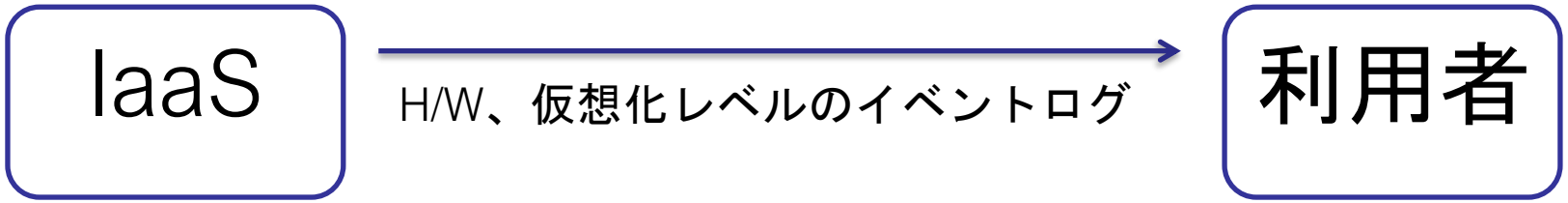
引用：　クラウドインシデントレスポンス（CIR）

責任共有モデルにおけるインシデント対応の考え方

- 利用者
  - 自組織のインシデント対応計画にどのようにクラウドを組み込むか？
    - 技術的
      - ログ管理；　利用者側の監視の集中化。プロバイダ提供のログの取り込み、統一管理
      - ログ機能：　プロバイダから提供を受けるログ機能を利用
    - 組織的
      - 組織の要求事項に対してプロバイダ側のインシデント対応が十分かどうかの評価。
        - プロバイダの役割、責任等、SLAとしてプロバイダと契約
        - プロバイダのインシデント対応能力の証拠の確認（SOC2, STAR, ISMSなど）
- プロバイダ
  - 大多数の利用者の要求事項をカバーできるインシデント対応
    - 利用者ごとに個別に対応するのは難しい。体系化されたスキームが必要
    - SLAとして利用者と契約
  - クラウド環境でのインシデントの通知、報告

インシデント対応における責任共有モデル

- IaaS
  - プロバイダ管理
    - ハードウエアから仮想環境までの監視、ログの提供
  - 利用者管理
    - OS以上（サーバ、ネットワーク、ストレージ）のログの監視
    - アプリケーションの監視、セキュリティの作りこみ（ログの作成など）

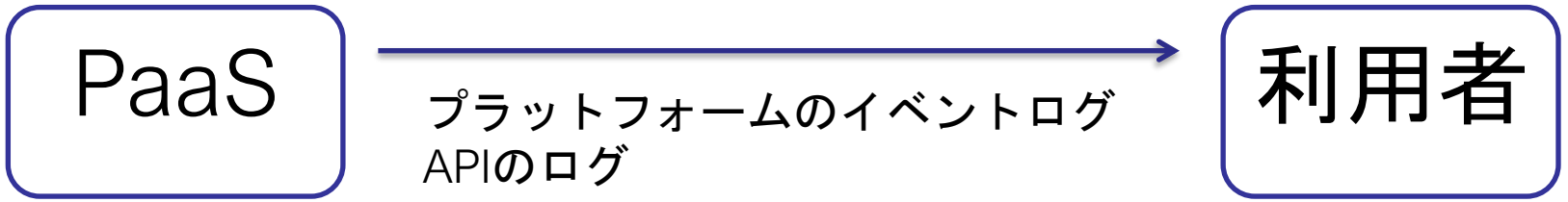


IaaSインシデント対応
- H/W、仮想化環境の監視
- 利用者に監視機能を提供
- 利用者との窓口の設定
- IaaS内インシデントの通知、報告

I利用者インシデント対応
- 利用者作り込み機能（アプリ等）の監視、イベントログ
- OS以上（syslog等）の監視、イベントログ
- ID、データアクセスの監視、イベントログ
- 統一したイベントログの監視、管理
- プロバイダとの窓口の設定

インシデント対応における責任共有モデル

- PaaS
  - プロバイダ管理
    - インフラからライブラリまでの監視、ログの提供
    - APIの監視、ログの提供（入力妥当性、出力妥当性などを含む）
  - 利用者管理
    - アプリケーションの監視、セキュリティの作りこみ（ログの作成など）

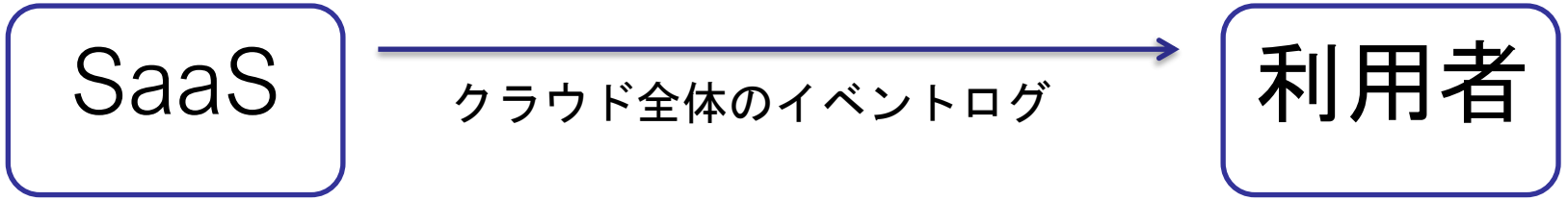


PaaSインシデント対応
- プラットフォームの監視
- 利用者にプラットフォーム、APIの監視機能を提供
- 利用者との窓口の設定
- PaaS内インシデントの通知、報告

l利用者インシデント対応
- 利用者が作り込むアプリの監視、イベントログ
- PaaSイベントログを含めた統一した監視、管理
- プロバイダとの窓口の設定

インシデント対応における責任共有モデル

- SaaS
  - プロバイダ管理
    - インフラからアプリケーションまでの監視、ログの提供
  - 利用者管理
    - インシデント対応の観点からは特になし



SaaSインシデント対応
- クラウド全体の監視
- 利用者にクラウド全体の監視機能を提供
- 利用者との窓口の設定
- SaaS内インシデントの通知、報告

利用者インシデント対応
- SaaSログを含めた統一したイベントログ監視、管理
- プロバイダとの窓口の設定

クラウドインシデントレスポンス　フレームワーク

- クラウドインシデントレスポンス（CIR）フレームワークの考え方
  - **CSCのインシデントレスポンス計画にCSPを含める**
  - NIST SP800-61

  - 従来のIRフレームワークとCIRフレームワークの違い
    - CSCとCSPの間に「**責任共有モデル**」が存在する
      - 情報、ログなど、システムの所有者がCSPとなる
      - CIR計画のプロセスにCSPを含める必要
      - CSCはCSPのIRの理解が必要　->　SLA/契約との整合性

インシデント情報共有の重要性

- インシデント情報を他組織やパートナーと共有する能力を持つことは重要

- 適切な情報共有
  - ビジネスインパクトに関する情報
    - 影響を受ける7企業のミッションを保全することに関心のある組織にのみ報告
    - ビジネスインパクト情報を外部組織と共有することは避けるべき
  - 技術情報
    - 可能な限り多くの技術情報を共有すべき
      攻撃や新たな脅威の技術的な詳細を共有することで、防御を強化

- CSPダッシュボード
  - プロバイダはインシデントに関するダッシュボードを提供
  - 利用者にダッシュボードとして提供そろことで、多数の利用者にインシデント情報を提供可能

責任共有の観点

1. リスク管理、セキュリティ
2. インシデントレスポンス
3. **ガバナンス、コンプライアンス、監査**
4. プライバシー
5. 新しいモデル

クラウドにおけるガバナンスの考慮点

- 直接管理できなくなる
- 複数の法律、裁判管轄
- 透明性
- 利用者要件に基づいたカスタマイズができない
- クラウドサービスごとに異なるセキュリティの成熟度
- サプライチェーン全体のセキュリティの維持
- 責任共有モデル、プロバイダ/利用者の責任境界
- テストではなく評価が中心になる

- 以上の状況においても利用者側に以下の責任が存在する
  - 説明責任
  - データのオーナーシップ

サービスモデルごとのガバナンスの考え方

- IaaS
  - プロバイダの数が限られているので、**すべてのプロバイダに適用できる統一したポリシー**を作成し単純化が可能
- PaaS
  - 基本的にはIaaSと同様。PaaSとIaaSは一緒に利用されるケースが多い。
  - 契約内容の精査が必要
- SaaS
  - 統一したガバナンスモデルの作成は困難。様々なプロバイダの様々な成熟度がある。
  - クラウド全体にわたっての透明性が低い
  - それぞれのプロバイダの**評価、契約、SLA、監視**が重要

ガバナンスにおける責任共有

- RACIマトリックス
  - Responsible(管理責任、実行責任)
  - Accountable(説明責任)
  - Consulted(相談先)
  - Informed(報告先)

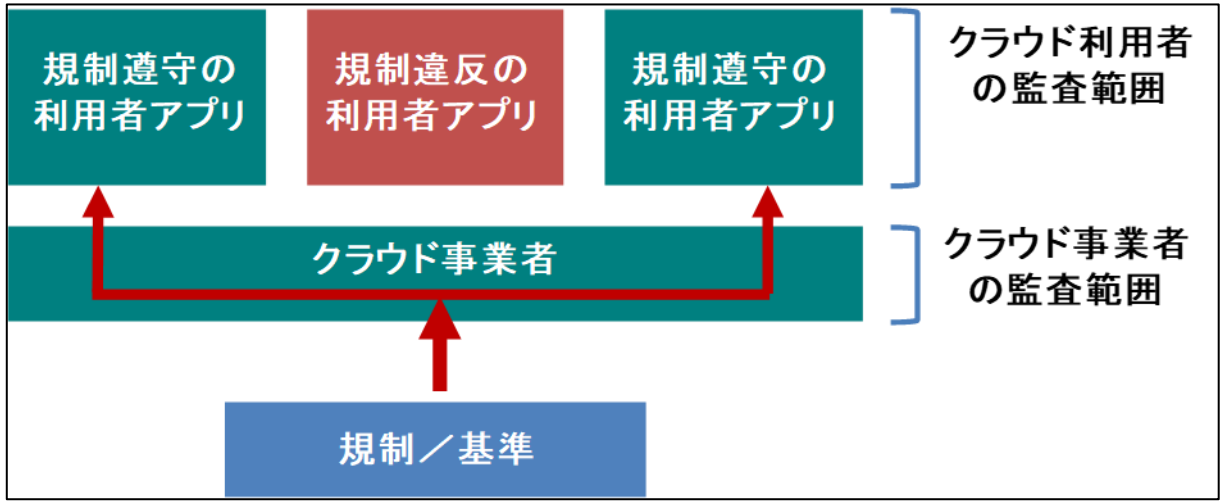
- クラウド利用において、RACIに基づく役割／責任の明確化は重要（責任共有）
- 説明責任と文書化
- 変化の激しいクラウドへの対応

コンプライアンスにおける責任共有

- コンプライアンスにおける責任共有の考慮事項
  - プロバイダ：　たくさんの利用者に対応できるコンプライアンス対応が必要
  - 利用者：　最終的な監査・コンプライアンスの責任を負う

- **コンプライアンス継承**
  - プロバイダのインフラやサービスはプロバイダのコンプライアンス範囲
  - 利用者が作り込むものはすべて利用者の範囲
  - コンプライアンス継承により、プロバイダのインフラやサービスは、利用者のコンプライアンスの対象外にできる

コンプライアンス継承



引用：　クラウドセキュリティガイダンス V4.0

監査における責任共有

- 監査マネージメント
  - 適用範囲、適用宣言書
    - 内部・外部の要求事項の文書化
    - どのようなシステムにどのようなコンプライアンス要件が必要になるかを明確化
  - 監査証明
    - 第三者からの法的なステートメント
    - クラウド利用者にとって、プロバイダを評価し利用する際の重要なツールとなる
  - クラウドで監査マネージメントがどのように変わるか？
    - 利用者は、プロバイダの第三者監査証明に頼る
      - 監査証明の開示要求（NDAが必要となる場合もある）
    - プロバイダは、プロバイダが義務を果たしていることを示すために、厳格な第三者の監査証明を提供すべき
    - 監査証明が最新の結果であることを保つ（プロバイダ、利用者）

責任共有の観点

1. リスク管理、セキュリティ
2. インシデントレスポンス
3. ガバナンス、コンプライアンス、監査
4. **プライバシー**
5. 新しいモデル

クラウドにおけるプライバシー　(1)

責任共有モデルとは言いにくいが、責任分担として考えてみる。

- 役割及び責任
  - データコントローラ
    - 個人データの管理者。個人データの取り扱いの目的および手段を決定する。
  - データプロセッサ
    - データコントローラに代わって、個人データを処理する

- クラウドでの状況
  - クラウド利用者がデータコントローラ、CSPがデータプロセッサであるケース

  OR
  - クラウド利用者とCSPの両者がデータコントローラであるケース

クラウドにおけるプライバシー(2)

- データコントローラの責任範囲
  - データサブジェクト（データ主体）に対する説明責任

- データプロセッサの責任範囲
  - データコントローラの指示に従った処理の実施
  - 処理行為の記録の保存
  - 監督機関との協力
  - 処理の安全性
  - 個人データ侵害のデータコントローラへの通知
  - データ保護担当者の指名
  - 第三国へのデータ移転に関する義務

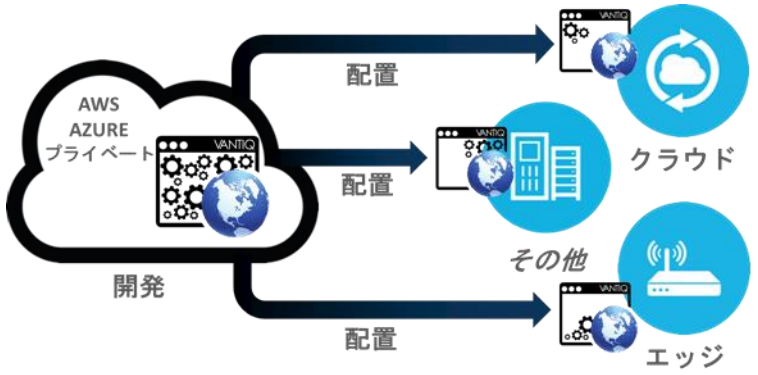
責任共有の観点

1. リスク管理、セキュリティ
2. インシデントレスポンス
3. ガバナンス、コンプライアンス、監査
4. プライバシー
5. **新しいモデル**

コンテナ、マイクロサービス、サーバレスの影響

- マルチクラウド
  - 複数のクラウド（オンプレを含む）を組み合わせた最適な利用
- エッジコンピューティング
  - 5G
  - IoTなど、よりデバイスに近いところで処理
  - リアルタイム性の追求
- サービスのプラットフォーム化
  - 適材適所のコンポーネントの配備
  - 開発から配備まで一貫した管理

クラウド集中から<u>クラウド分散</u>への移行

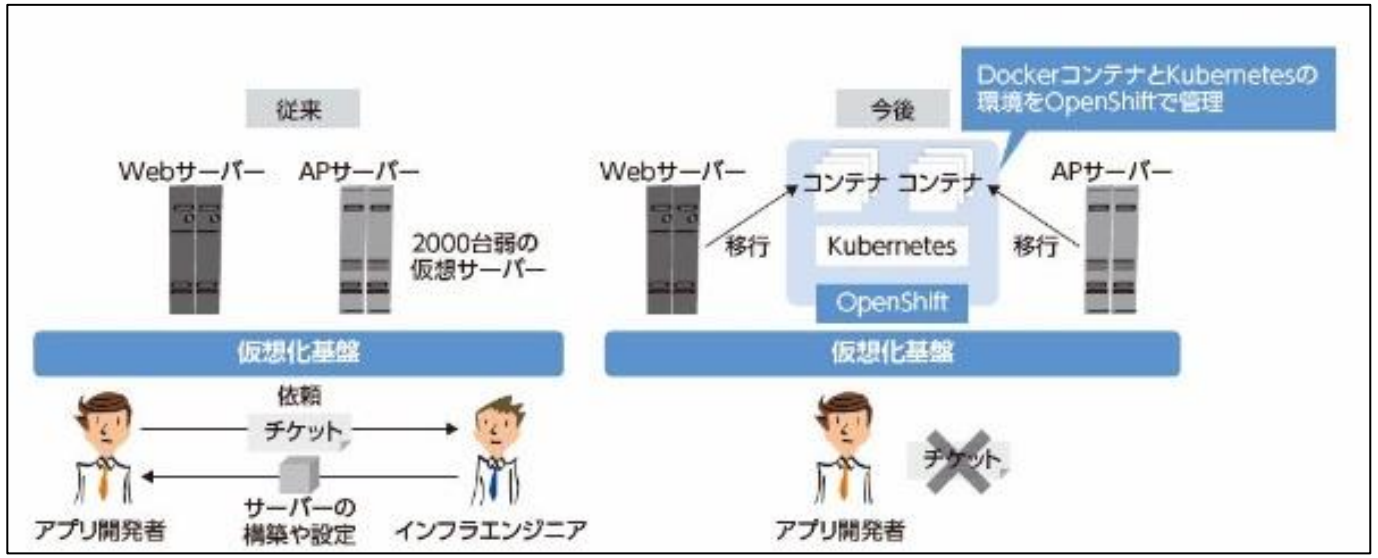


CSA勉強会　「IoT環境を支えるイベント・ドリブン・アーキテクチャ（EDA)とそのセキュリティ」　資料より引用

クラウド分散を支える技術(1)

コンテナ

- OSのリソースを共有し活用する、OSの中で稼動するコード実行環境（ワークロード）



引用：https://tech.nikkeibp.co.jp/atcl/nxt/column/18/00001/00761/

Kubernetes

- 自動配備、スケーリング、操作するためのプラットフォーム

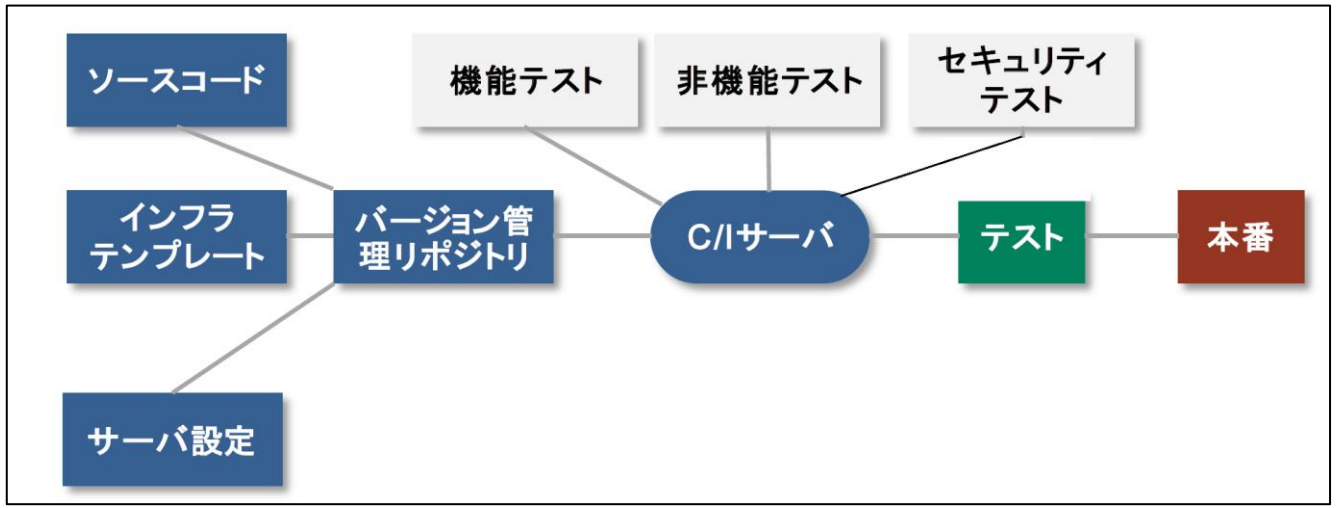
クラウド分散を支える技術(2)

- **サーバレスコンピューティング**
  - クラウド利用者が下層のハードウェアあるいは仮想マシンを何も管理せず、単に外部に提示される機能にアクセス
    - 例：Amazon Lambda
  - インフラの設定、容量の計画、サーバ管理は、フル マネージド サービスとしてクラウド プロバイダーが実施
  - 利用者がインフラについて悩む必要がない
    - 必要に応じてスケーリング可能
    - アプリを自由に作成、管理、配備可能

- **マイクロサービス**
  - 疎結合型のソフトウエア開発

クラウド分散を支える技術(3)

- **DevOps、DevSecOps**
  - 開発から配備までを一貫して実施。
  - ソフトウエアにより開発から配備までの一貫した管理（Infrastructure a Code）
  - イミュータブル（immutable）なワークロード
    - パッチを当てたりあるいは他の変更を加えたりはしない
    - 稼働イメージは新しいイメージでオーバーライトする

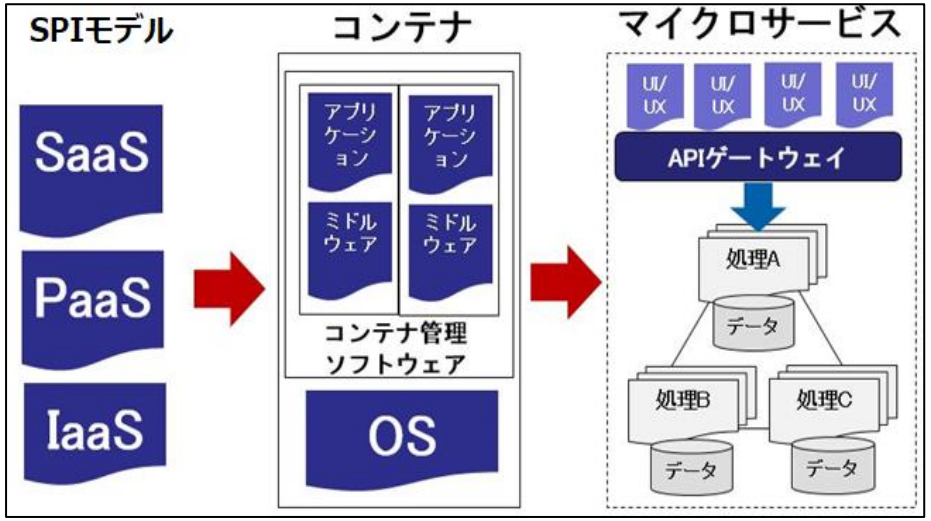


クラウド分散 - 疎結合型クラウド

- コンシューマードリブンなビジネスモデルはXaaSからマイクロサービスのプラットフォームへ（例：Fintechサービス）

- クラウド分散化におけるプラットフォーム・セキュリティの考慮事項
  - コンポーネントのユーザー認証／認可
  - コンポーネント間通信のセキュリティ
  - コンポーネントの完全性保証
  - サードパーティアプリケーションとの通信のセキュリティ
  - デバイス／センサー自体のセキュリティ
  - デバイス／センサーとプラットフォーム間通信のセキュリティ
  - 開発環境（特にマルチテナント環境）のセキュリティ
  - 利用者側の監査／コンプライアンスへの対応

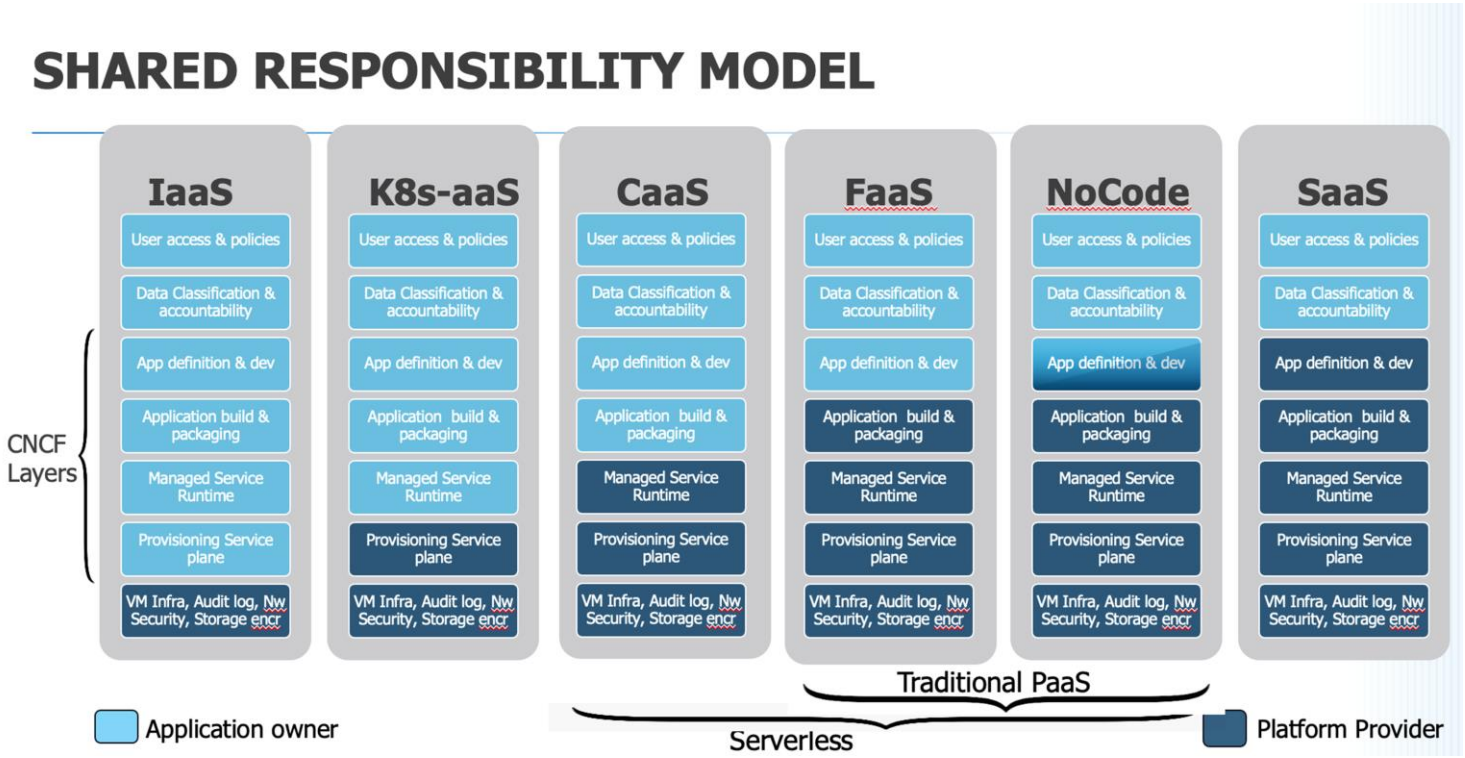
モノリシックな三層型アーキテクチャから　　自律サービスの疎結合型プラットフォームへ



出典：ヘルスケアクラウド研究会（2015年11月）を基に作成

コンテナ、マイクロサービス、サーバレスと責任共有（1）

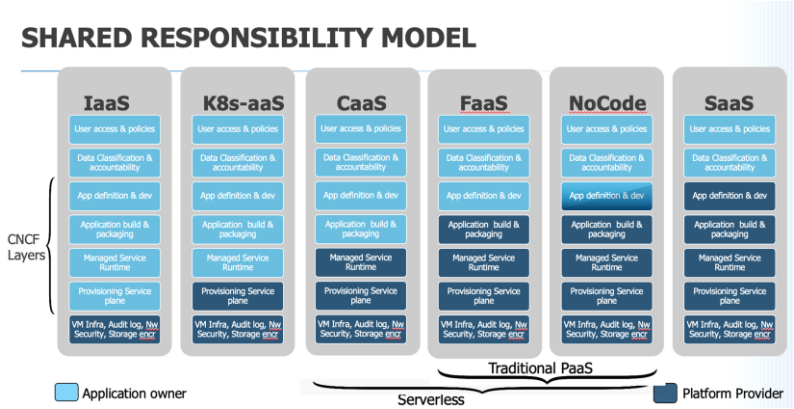
- 3つのサービスモデル、SaaS、PaaS、IaaSはすでに過去のことであり、モデルは新しいプラットフォームを包含するように進化する必要がある、という考え方。
  ただし、まだオフィシャルではない。
- コンテナ、マイクロサービス、サーバレス等を考慮した新しいサービスモデル
- より多くの責任がプラットフォームプロバイダになる



引用：　CSAジャパンブログ（2020年2月9日）より

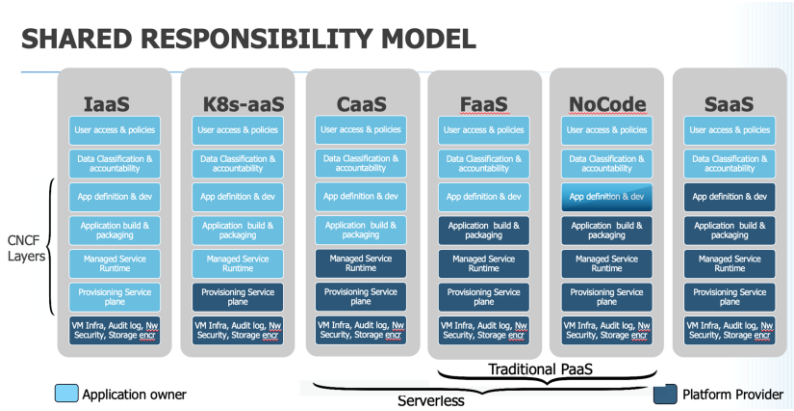
コンテナ、マイクロサービス、サーバレスと責任共有（２）

- K8s-aaS
  - Kubernetesコントロールプレーンは、プラットフォームプロバイダが管理
  - データプレーンのライフサイクルとその管理は、アプリケーションの所有者
  - 例：Amazon Elastic Kubernetes Service（EKS）、Azure Kubernetes Service（AKS）、Google Kubernetes Engine（GKE）
- CaaS
  - アプリケーションの所有者がアプリケーションコンテナを提供し、プラットフォームプロバイダがコントロールプレーンとデータプレーンの両方を管理
  - 例：Amazon Web Services（AWS）Fargate（ECSFargateとEKSFargateの両方）、Azure Container Instances（ACI）、GoogleCloudRun
- FaaS
  - アプリケーションの所有者は、機能を実行するレイヤーとともにビジネスロジックを提供
  - サービスコントロールプレーンとデータプレーンは、サービスプロバイダによってすべて処理
  - 例：AWS Lambda、Azure Functions、Google CloudFunctions



コンテナ、マイクロサービス、サーバレスと責任共有（３）

- NoCode-as-a-Service (NCaaS)
  - コードロジックはアプリケーションの所有者によって提供
  - サービスプロバイダは、仕様と構成からコードを生成し、ソフトウェアをビルド、パッケージ化、および実行
  - **例：** Azure Power Apps、Google AppSheet、AWSHoneycode



- Serverless
  - サーバレスプラットフォームを使用すると、開発者は開発と配備をより迅速に行うことができ、コンテナクラスタや仮想マシンなどのインフラストラクチャを管理しなくてもクラウドネイティブサービスに簡単に移行できる
  - **例：** CaaS / FaaSおよびNCaaSプラットフォーム

4. まとめ

まとめ

1. クラウドセキュリティの基本は責任共有モデル
2. 利用者とプロバイダの間で責任境界を明確にし、それぞれの責任範囲を理解することが大切
3. 利用者は説明責任を果たすことが重要。デューデリジェンスが重要
4. 様々な観点からの責任共有モデルの理解
5. 変化の激しいクラウド環境への追従が必要

CSAの活動　==　「場」の提供！

様々なワーキンググループ活動の「場」

自由な情報発信の「場」

https://cloudsecurityalliance.jp

info@cloudsecurityalliance.jp

ありがとうございました