# 「ゼロトラスト環境実現に必要なIGAとPBACについて」

2023/08/23

デジタルアイデンティティWG　リーダー　宮川　晃一
日本電気株式会社

# デジタルアイデンティティWGの紹介

# WGの活動の目的

## 「デジタルアイデンティティＷＧの目的」

本WGは“デジタルアイデンティティ”全般を広く議論する場として、
今年度で設立１８年目を迎えました。

本WGでは、デジタルアイデンティティの課題等について議論し、
導入指針や各種報告書の提示、執筆活動・セミナー・勉強会等での
啓蒙活動および普及促進、関連他団体との連携による市場活性化等を
目的として活動を行っています。

メンバー紹介：
https://www.jnsa.org/active/std_idm.html

# WGのこれまでの活動

| 年 | 成果物等 | WG名称 |
|---|---|---|
| 2005 | WG設立 | 内部統制における<br>アイデンティティ<br>管理WG |
| 2006 | 合宿実施（三浦マホロバツインズ） | |
| 2007 | 内部統制における<br>アイデンティティ管理解説書（第１版） | |
| 2008 | 内部統制における<br>アイデンティティ管理解説書（第２版） | |
| 2009 | | |
| 2010 | クラウド環境における<br>アイデンティティ管理ガイドライン<br>（企業向け調査レポート） | セキュリティにおける<br>アイデンティティ管理<br>WG |
| 2011 | | |
| 2012 | 改定新版　クラウド環境におけるアイデンティティ管理ガイドライン（書籍）<br>エンタープライズ<br>ロール管理解説書（第１版） | |
| 2013 | OpenID ConnectとSCIMの<br>エンタープライズ利用ガイドライン | アイデンティティ管理<br>WG |
| 2014 | エンタープライズ<br>ロール管理解説書（第2版） | |
| 2015 | １０周年記念セミナー！<br>エンタープライズ<br>ロール管理解説書（第3版） | |
| 2016 | エンタープライズにおける特権ID管理<br>解説書（第１版） | |
| 2017 | ID管理システム導入における現状把握チェックリスト（第１版）<br>クロスボーダー時代のアイデンティティ管理セミナー! | |
| 2018 | 内部統制における<br>アイデンティティ管理解説書（第２版） | |
| 2019 | クレデンシャルの歴史（読み物） | |
| 2020 | Software Design<br>11月号特集（雑誌） | デジタルアイデンティティWG |
| 2021 | Enterprise Identity Day！<br>標準化部会セミナー！ | |
| 2023 | 今さら聞けない暗号技術＆認証・認可（書籍）<br>改定新版　エンタープライズにおける特権ID管理ガイドライン（解説編）<br>ミニウェビナー＆Youtube<br>「？？？とアイデンティティ」 | |

# WGのこれまでの成果物

https://www.jnsa.org/result/digitalidentity/index.html

▶・2023/5/8

報告書 ニュージーランド政府による"Identification Management Standards"に関する考察
==NIST SP800-63 "Digital Identity Guidelines"との比較結果等==

▶・2023/3/31

報告書 【改定新版】特権ID管理ガイドライン　解説編

▶・2023/3/6

関連書籍発売「Software Design　今さら聞けない認証・認可」が再編集されて別冊シリーズで発売されました。
技術評論社さんのページにリンクします。

▶・セミナー　2023/5/25開催　参加登録受付中

セミナー｜デジタルアイデンティティWGミニウェビナー「？？？とアイデンティティ」

2021/11/26

セミナー資料　2021年11月26日（金）開催
「Enterprise Identity Day　再考！！エンタープライズ・アイデンティティ～ゼロトラストセキュリティの礎を確立する～」

▶・執筆「Software Design」2020年11月号

特集1「今さら聞けない認証・認可ーセキュアなIAMを実現するために覚えておきたいこと」
技術評論社さんのページにリンクします。

▶・読み物「クレデンシャルの歴史」

▶・報告書「ID管理システム導入における現状把握チェックリスト（第1版）」

▶・出版書籍「＜改訂新版＞クラウド環境におけるアイデンティティ管理ガイドライン」
Amazonにリンクします

▶・報告書「OpenID ConnectとSCIMのエンタープライズ利用ガイドライン」
（JNSAとOpenID Foundation Japanとの共同執筆）

▶・報告書「エンタープライズにおける特権ID管理解説書（第1版）」

▶・報告書「エンタープライズロール管理解説書（第3版）」

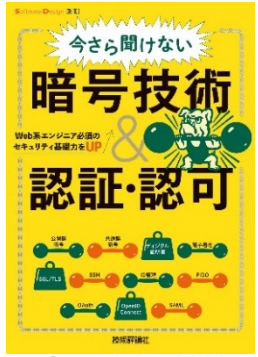
1 デジタルアイデンティティWGミニウェビナー第1回「ネットワークとアイデンティティ」
JNSA Channel・1173 回視聴・6 か月前

2 デジタルアイデンティティWGミニウェビナー第2回「内部統制/IT全般統制とアイデンティティ」
JNSA Channel・459 回視聴・4 か月前

3 デジタルアイデンティティWGミニウェビナー第3回「デバイスとアイデンティティ」
JNSA Channel・228 回視聴・3 か月前

4 デジタルアイデンティティWGミニウェビナー第4回「IaaSとアイデンティティ」
JNSA Channel・162 回視聴・1 か月前

YouTube JP

# アジェンダ

1．ゼロトラストとIGA/PBACの関係
2．IGA（Identity Governance and Administration）とは
3．PBAC（Policy Based Access Control）とは
4．まとめ

（概要）

ゼロトラスト環境を実現するには、大きく3つの要件が必要です。１）情報収集、２）収集情報とポリシーを組み合わせたアクセスレベルの適切な決定、３）アクセスレベルに応じた動的なアクセス制御の実施です。情報収集はゼロトラストを始めるための最初の一歩と言えますが、その中でも**最も重要な情報が、アクセスしてきたユーザー（主体）のアイデンティティ情報です。アクセスしてきたのが「誰」なのかがわからなければ、適切なアクセスレベルの判断を行うことはできませんし、適切なアクセス制御も不可能です。アイデンティティ情報はゼロトラストの根幹であり、これをなくしては実現不可能です。**本セッションではアイデンティティ情報を高度に管理するための**IGA**（アイデンティティガバナンス管理）と**動的に適切なアクセス制御を実現するための方法**として、**PBAC**（ポリシーベースアクセス制御）をご紹介します。

# ゼロトラストとIGA/PBACの関係

# ゼロトラストの３大要件

## １．情報収集

アクセスレベルや最終的なアクセス制御を行うために必要な情報を収集する。

ID管理システムやSIME、脅威インテリジェンスなど様々な情報を収集する。

## ２．アクセスレベルの決定

情報収集から得られた情報をもとにアクセスのレベルを決定する。

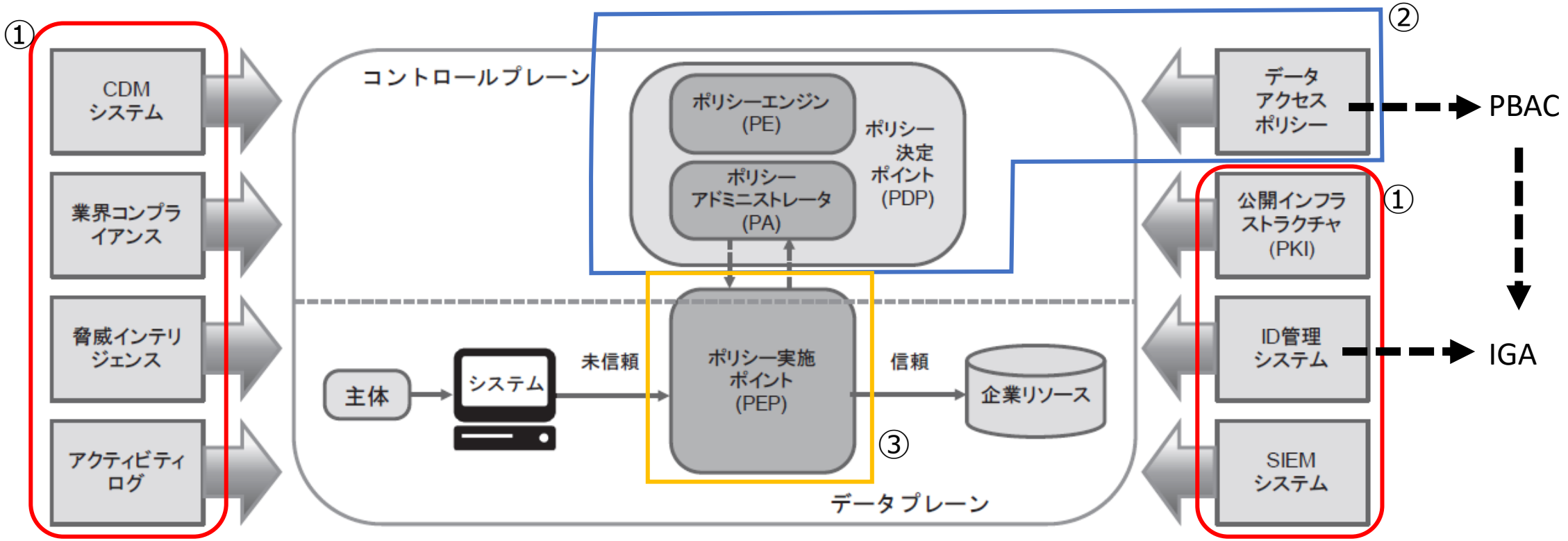
アクセスレベルの決定はリアルタイムに更新される。

## ３．動的なアクセス制御

決定されたアクセスレベルに従って、主体がアクセスできるリソースが動的に制御する。

# ゼロトラストとIGA/PBACの関係

①情報収集 → ②アクセスレベルの決定 → ②動的なアクセス制御




**※CDM: Continuous Diagnostics and Mitigation**
サイバーセキュリティのリスクを検知し、
優先順位に応じて緩和する仕組み。

**図 2:ゼロトラストの中核となる論理コンポーネント**

出典：NIST SP800-207 日本語翻訳版に筆者が加筆

# IGA (Identity Governance and Administration)

# デジタルアイデンティティから見た Never Trust , Always Verify の前提条件

**主体（ユーザ等）が客体（データ・アプリ等）に対して**
**どのようなアクセスする権限を持っているかを**
**正しく定義・管理されている状態であること！**

**最小権限の原則**

業務に必要とされる最低限のデータ・アプリケーションに対するアクセス権のみを与えるという情報セキュリティの基本となる考え方

＋

**Need to Knowの原則**

ビジネス上の役割に応じて、権限の付与・見直し・更新・削除を実現

＝

**アイデンティティガバナンス**

# IGAとは

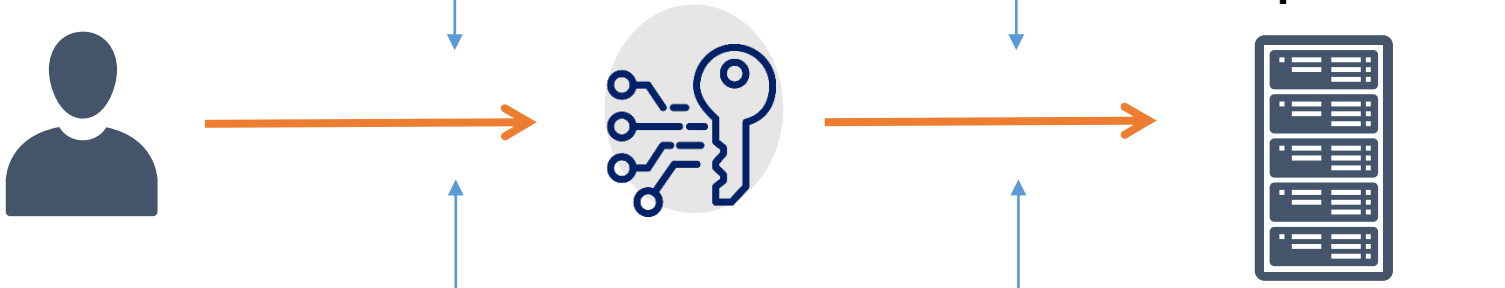
**IGAとは旧来からのプロビジョニングを主体としたID管理に加えて、**
**それがガバナンス上「正しく」実行されているかをチェックし管理すること。**

# IGA（Identity Governance and Administration）

**User Administration & Provisioning**

ライフサイクル
入社
異動
退職

グループ、ロール、
プロファイル、個別設定



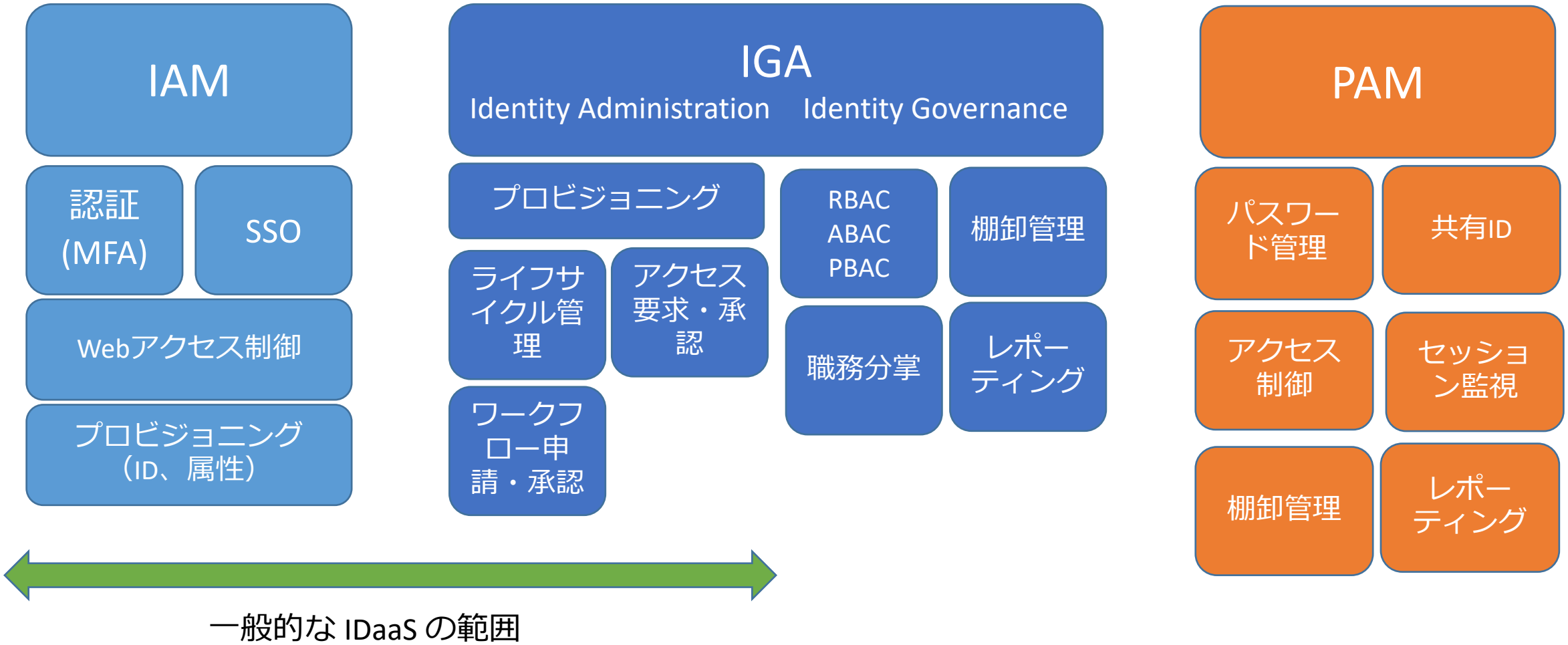

**Identity & Access Governance**

誰が申請、承認したか
いつ付与されたか
現在も必要なのか
最後に利用したたのはいつ
利益相反関係はないか

誰がデータオーナーか
特権的なアクセスか
最後に確認したのは誰か

正しくの意味
PKI/電子署名
も重要要素！

# IAM/IGA/PAMの違い



# IGAまとめ

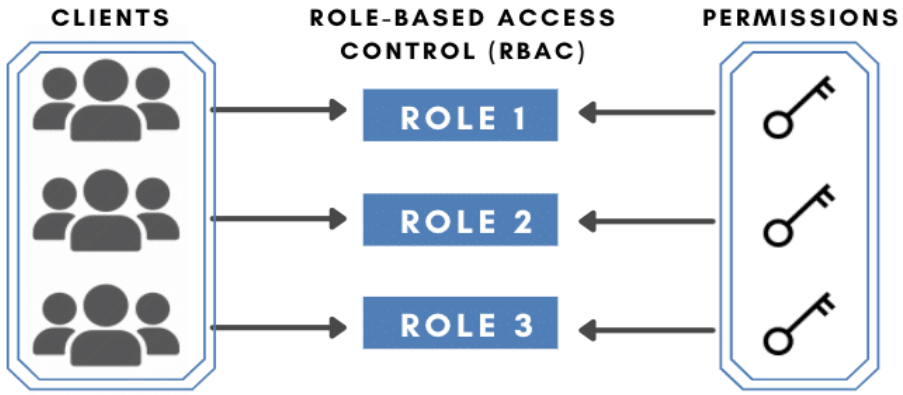
ある時点のアカウント情報・アクセス権限が正当・確かであったとしても将来的に、あるいは「今」が正当で確かであるとは限らない。
よって、継続的に、アイデンティティ情報を確認・レビューし、
**「最小権限の維持」**を続けることが必要である。

# PBAC（Policy Based Access Control）

# RBAC/ABAC　（従来からのパーミッションモデル）

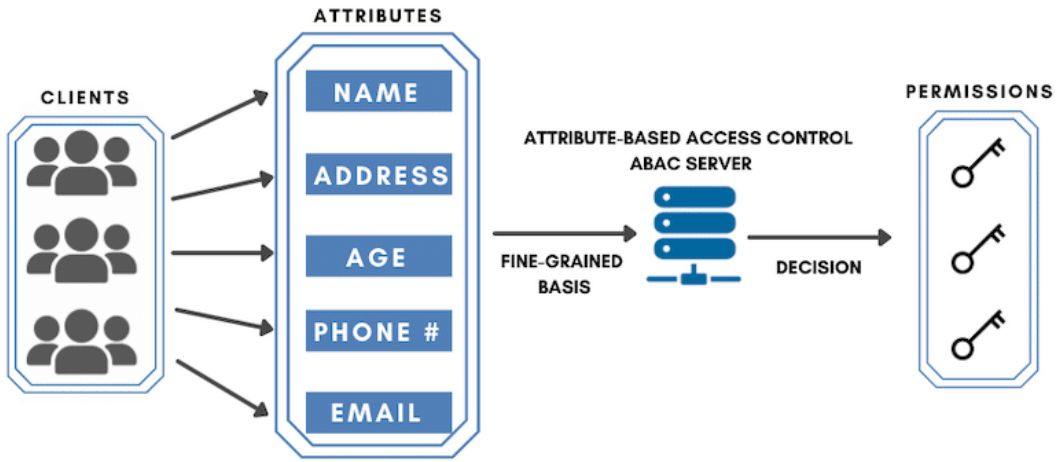
## Role Based access control (RBAC)

・直観的で分析しやすい
・きめ細かい権限管理をしようとすると
　意図せずにロールが増加
・スタティックな評価




## Attribute Based access control (ABAC)

・個人の属性値（役職、組織、決済レベル等）
　　をベースに権限を判断する
・柔軟性が高い
　　（リソースが増えても対応しやすい）
・トップダウンでの把握が難しい
・一元管理が困難で監査に不向き
・リアルタイムな評価



# PBAC

## Policy Based access control (PBAC)

・リソースとユーザの組み合わせによるきめ細かい制御
・プリンシパル（主体・利用者）やリソースの現在の属性値に基づいてアクセスを判断
・理解しやすく、容易にメンテナンスが可能なルールを生成（理解しやすい言語の開発）
・アプリケーションからの独立性が高い
・ABACをわかりやすく進化させたもの

例）

head

```
permit(
 principal in User::"alice",
 action == Action::"view",
 resource in Album::"Vacation")
```

condition

```
when {
 resource.owner == "alice"
};
```

# Policy as Code とは

**ポリシーを理解しやすい言語で表現する＝”Policy as Code”**

**【HashiCorp】**

Policy as Code とは、高級言語を用いて書かれたコードによって、ポリシーを管理および自動化するという考え方である。テキストファイルにコードとしてポリシーを表現することによって、バージョン管理や自動化テスト、自動デプロイのような実績のあるソフトウェア開発のベストプラクティスを適用することができる。

**【Palo Alto Networks】**

Policy as Code とは、コードを使ってポリシーを定義、更新、共有、適用するようなポリシー管理の試みである。手動プロセスの代わりにコードベースの自動化を活用してポリシーを管理することで、チームは迅速に動くことができ。またヒューマンエラーによるミスの可能性を減らすことができる。

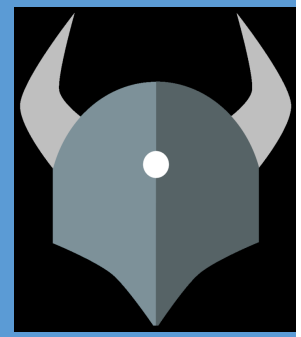
# RBAC/ABAC/PBAC　比較

| | RBAC | ABAC | PBAC |
|---|---|---|---|
| **Permissions depend on**<br>権限の依存元 | 事前に定義された役割のセット | 権限を決定するための特定の属性のセット | 関連する属性と権限を決定する特定のルールのセット |
| **Type of access control**<br>アクセス制御の粒度 | 粗い粒度 | きめの細かい | きめの細かい |
| **Real-time parameters**<br>リアルタイムな対応 | No. | Yes. | Yes. |
| **Context specificity**<br>コンテキストの特異性 | 新しいコンテキストごとに必要な権限を新しいロールとして定義 | 必要な権限の属性のセットを新しいコンテキストごとに定義 | ポリシーを定義し、新しいコンテキストごとに属性を調整する |
| **Cost to implement and maintain**<br>導入と維持にかかるコスト | 小規模な IT 部門で維持できるため、比較的安価 | 保守には大規模な経験豊富な IT チームが必要なため、コストがかかる | 導入にはコストがかかるが、メンテナンスにはコストがかからない |
| **Ease of compliance**<br>コンプライアンス確保の容易さ | 役割を割り当てると、特定の従業員に機密情報への望ましくないアクセスが与えられる可能性がある。コンプライアンスを確保するのは困難 | 非常に具体的な権限を付与することはできるが、企業ポリシーをビジネス側から IT側に変換するときに、エラーが発生し、コンプライアンスが弱まる可能性がある（ビジネスロジックの反映が難しい） | 使いやすい GUI （UMA) とポリシーへの適用により、ビジネスマネージャーは権限を直接設定でき、高いコンプライアンスを確保できる（ビジネスロジックの反映が容易） |
| **Ease of modification**<br>修正の容易さ | 新しいロールを定義したり、特定のロールの特定の権限を取り消したりするのは非常に簡単だが、ロールが爆発的に増加するリスクは高くなる | 新しい属性を定義するのが難しい | 高レベルのポリシー言語を使用して新しいポリシーを定義するのが比較的簡単 |

出典：https://www.styra.com/blog/what-is-rbac-vs-abac-vs-pbac/

# ユースケース１：Open Policy Agent (OPA)

## Open Policy Agent (OPA)



- オープンソースの汎用ポリシーエンジン
- CNCF graduated project (CNCF: Cloud Native Computing Foundation)
- Domain依存
  特定のドメインに依存せずに利用可能
  例：Kubernetes, Envoy, Terraform, AWS, Linux, SQL, GraphQL API　etc.
- 構造化データを入力(input) として受け取り、ポリシーを評価(evaluate) して、判定結果(decision) を返す
- ポリシーは “Rego” という宣言型言語で記述する

# ユースケース２：Amazon Verified Permissions

Cedar言語：
Amazon Verified Permissions
2023/5 オープンソース化

直観的で読みやすい
拡張可能でスケーラブル
高速で機械的な構文解析に向いている

head
```
permit(
 principal in User::"alice",
 action == Action::"view",
 resource in Album::"Vacation")
```
condition
```
when {
 resource.owner == "alice"
};
```

**P**rincipal: アクションを行う実行者
**A**ction: 実行しようとしている操作
**R**esource: 操作対象のリソース
**C**ontext: その他の条件

Can **P**rincipal perform **A**ction on **R**esource with the **C**ontext **?**
この条件において、この人はこのリソースに対するこの操作を実行できるか？

# PBACまとめ

- コードによってポリシーを運用するPolicy as Code という考え方
- ゼロトラスト／アクセスコントロールの文脈におけるPBAC
- コードを用いることによるメリット：
  ＊より柔軟な制御が可能であり開発者にとって扱いやすい
  ＊ドメインに依存せずインテグレーションが容易
- ポリシー評価をアプリケーションと分離することでわかりやすくなる
- スタティック評価からリアルタイム評価へ

# 本セッションのまとめ

- ゼロトラストの３大要件は
  情報収集／アクセスレベルの決定／動的なアクセス制御
- アイデンティティ情報の収集は、ゼロトラストの肝である
  そして、その鮮度と正確性を担保するために“IGA”の考え方が重要
- きめ細やかで柔軟であり、リアルタイムなアクセス制御を行うためには、
  “PBAC”を利用して実現するのが最適