Session 2

ZTNA展開図
～Zero Trust Network Access技術解説～

株式会社エーピーコミュニケーションズ
嘉藤 育宏（ya_kato@ap-com.co.jp）

APCommunications

2023/11/21 InternetWeek2023（東京大学 伊藤謝恩ホール）

Agenda

1. はじめに
2. Zero Trust Network Access
3. Zero Trust Network Accessの歴史
4. Reverse PROXY
5. Port knocking
6. Single Packet Authorization
7. VPNとZTNA/SDPの比較
8. Zero Trust Architecture の論理構成
9. SDPの仕組み
10. ZTNAの仕組み
11. おわりに

1. はじめに
2. Zero Trust Network Access
3. Zero Trust Network Accessの歴史
4. Reverse PROXY
5. Port knocking
6. Single Packet Authorization
7. VPNとZTNA/SDPの比較
8. Zero Trust Architecture の論理構成
9. SDPの仕組み
10. ZTNAの仕組み
11. おわりに

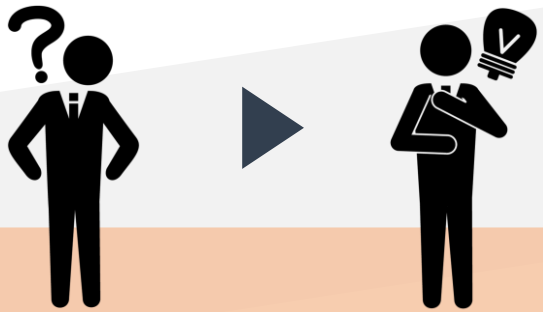
はじめに

対象者
- ゼロトラストを勉強中の方
- 脱VPNについて検討中の方

本セッションの目標 = Level2到達

Level1　興味

「Zero Trust Network Accessとは？」が理解できる



Level2　体感

「ZTNA/SDPの仕組み」が理解できる

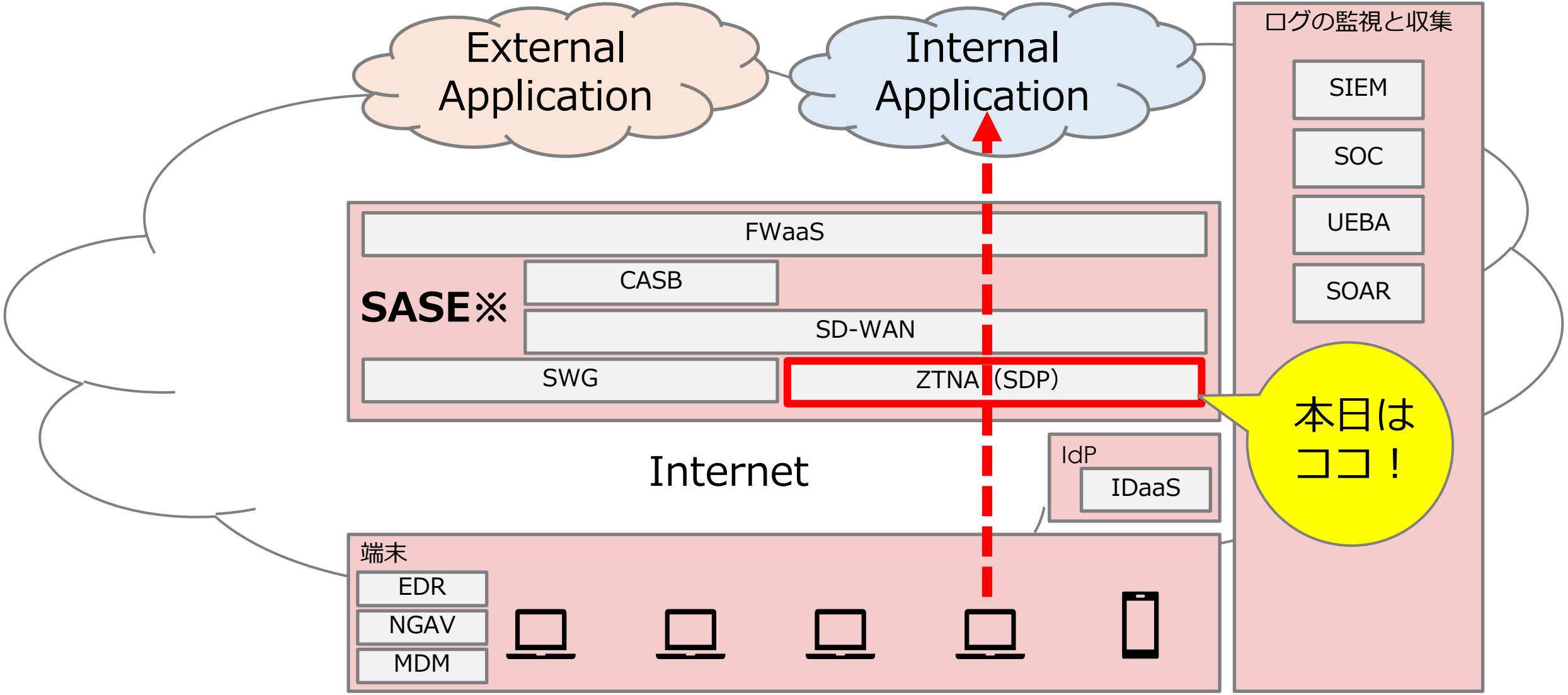
Level3　体験

「脱VPNへの一歩：VPNとZTNA/SDPの違い」を理解・説明できる

「Zero Trust Network Accessって何？」⇒「ZTNA/SDPの仕組みが理解できた」となってもらうことが目標

話の位置づけ

ゼロトラストの主要な技術要素とZTNA（SDP）の位置づけを以下に示します。



※ Secure Access Service Edge, SASE

SASE/SSEとは

セキュリティサービスとネットワークサービスを融合したものを、
Secure Access Service Edge, SASEといいます。

セキュリティサービス	
SWG	FWaaS
ZTNA	認証サービス
CASB	など

＋

ネットワークサービス	
MPLS/VPN	NW最適化
インターネット	QoS
SD-WAN	など

＝

SASE
Secure Access
Service Edge

またSASEのセキュリティ面を担うものを、
Security Service Edge, SSEといいます。

SASE

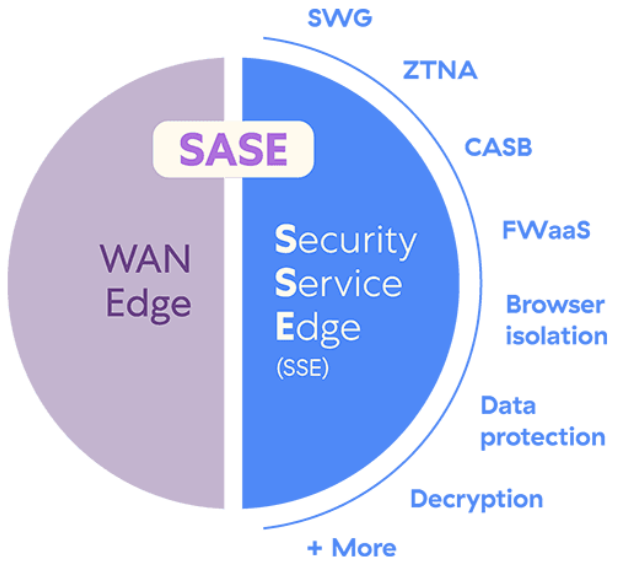
SSE

- SWG
- ZTNA
- CASB

3つのコアコンポーネント

- FWaaS
- DLP
- など

複数のオプションコンポーネント



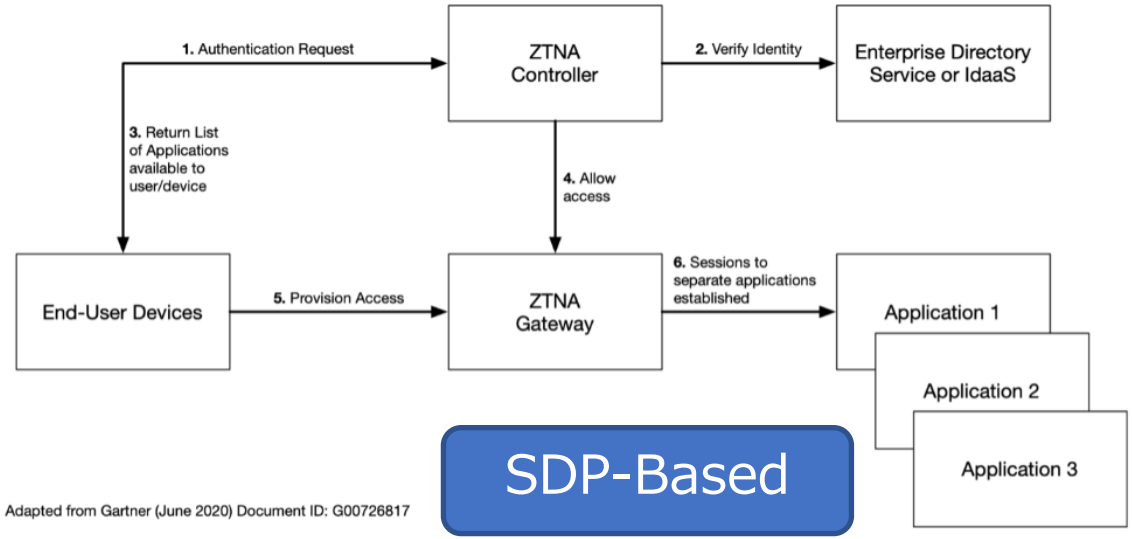
出典：CXO REvolutionaries、「Security Service Edge (SSE) reflects a changing market: what you need to know」

Zero Trust Network Accessとは

Zero Trust Network Access（ZTNA）は2種類あります。

appgate

WAVERLEY LABS

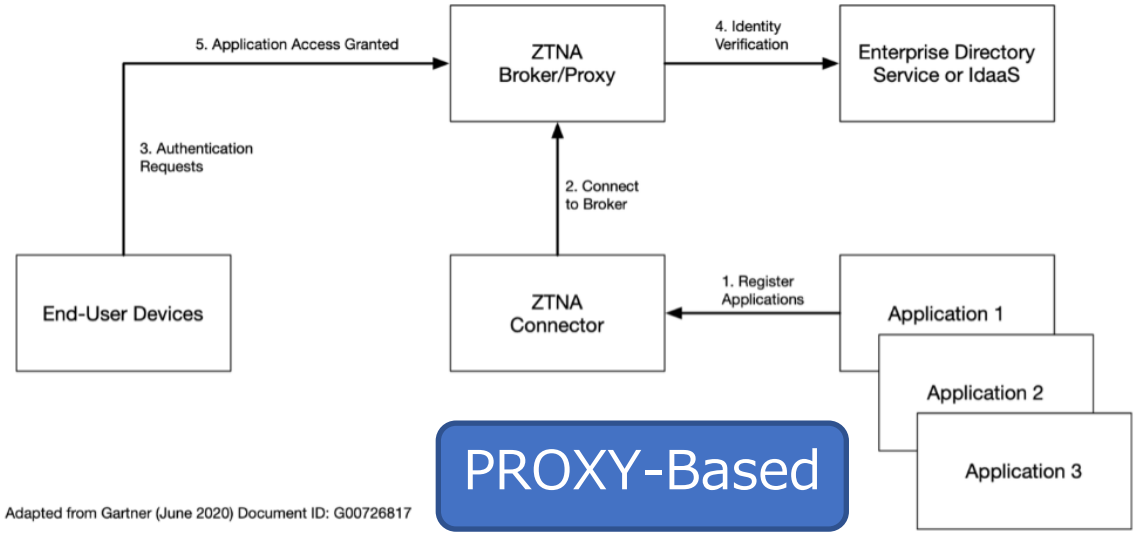


SDP-Based

zscaler　cisco　netskope　CLOUDFLARE　Google

paloalto NETWORKS

他にも色々



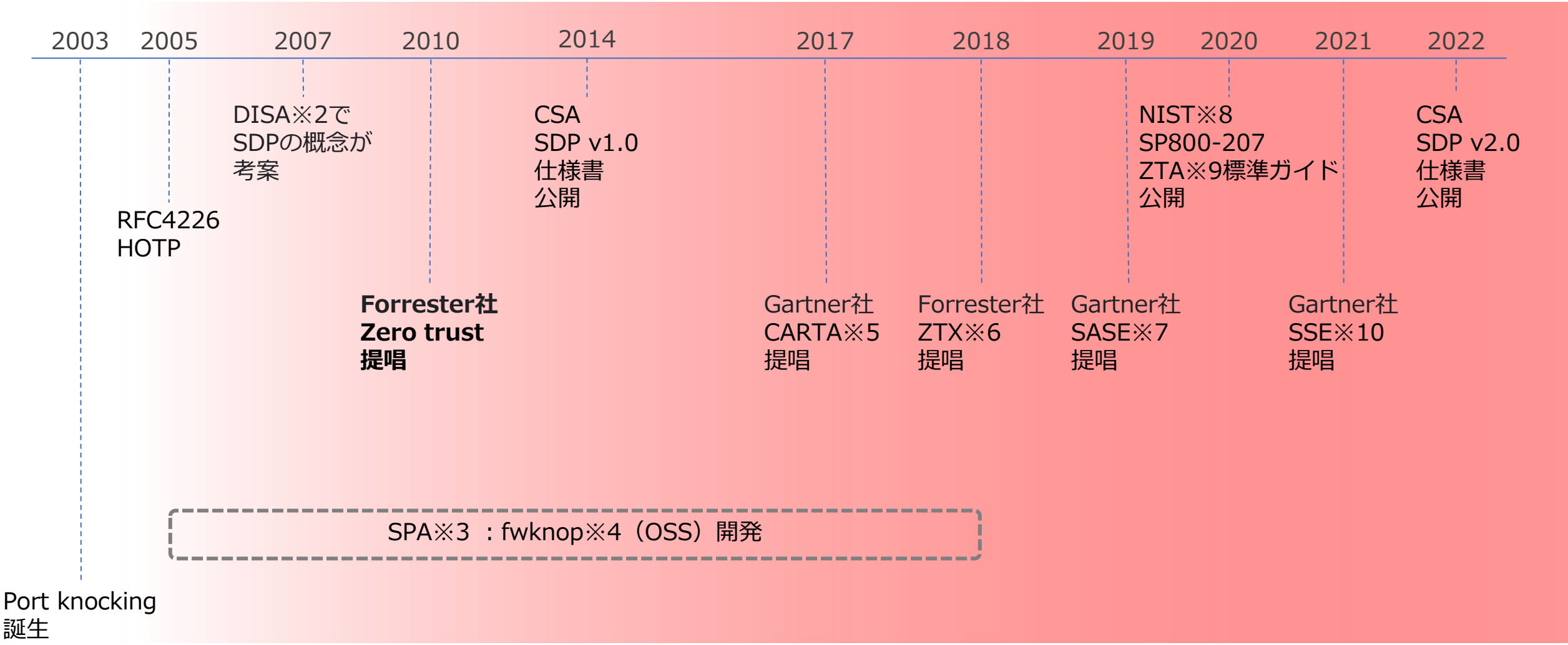
PROXY-Based

Endpoint-Initiated ZTNA
以降、**SDP**と省略します。

Service-Initiated ZTNA
以降、**ZTNA**と省略します。

Zero Trust Network Access（ZTNA）

Zero Trust Network Accessの歴史

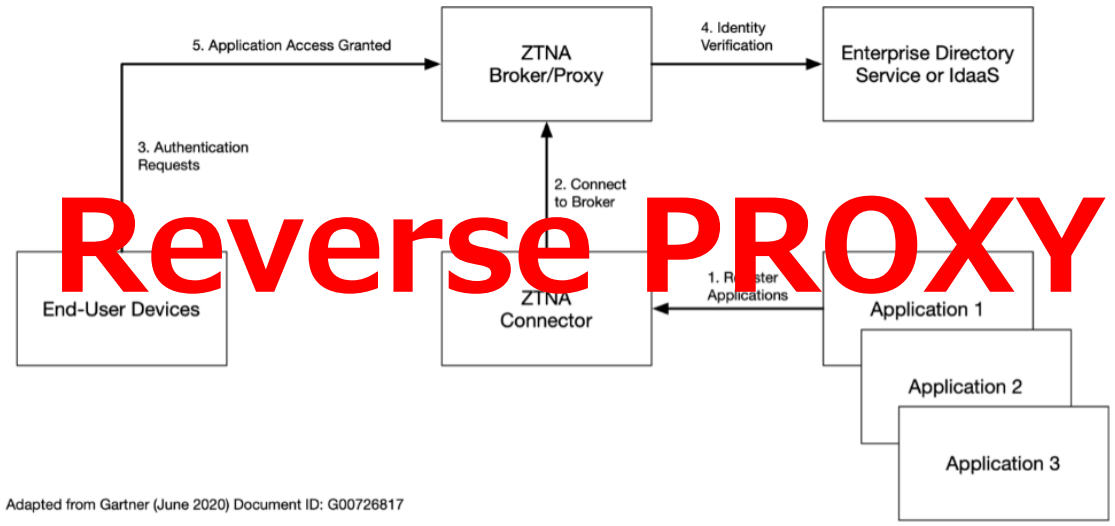


※1 Cloud Security Alliance, CSA
※2 Defense Information System Agency, DISA
※3 Single Packet Authorization, SPA
※4 FireWall KNock OPerator, fwknop https://www.cipherdyne.org/fwknop/
※5 Continuous Adaptive Risk and Trust Assessment, CARTA
※6 Zero Trust eXtended, ZTX
※7 Secure Access Service Edge, SASE
※8 National Institute of Standards and Technology, NIST
※9 Zero Trust Architecture, ZTA
※10 Security Service Edge, SSE

Zero Trust Network Accessの中核

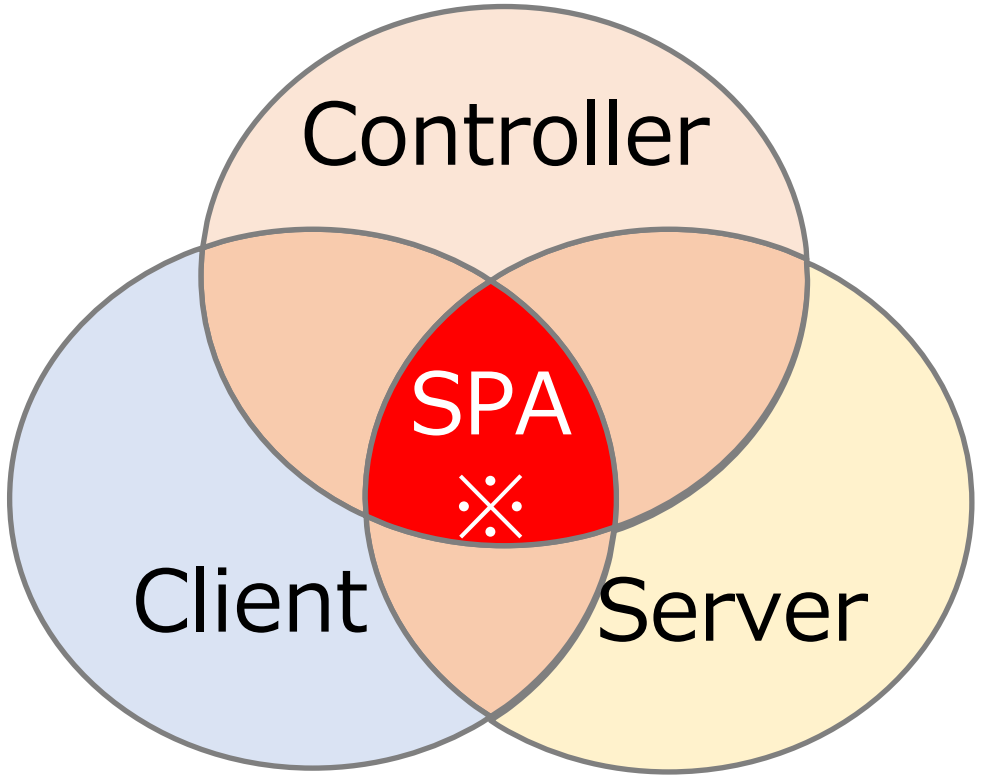
ZTNA / SDPの中核技術とは何でしょうか。

Zero Trust Network Access(ZTNA)



Adapted from Gartner (June 2020) Document ID: G00726817

Software-Defined Perimeter(SDP)



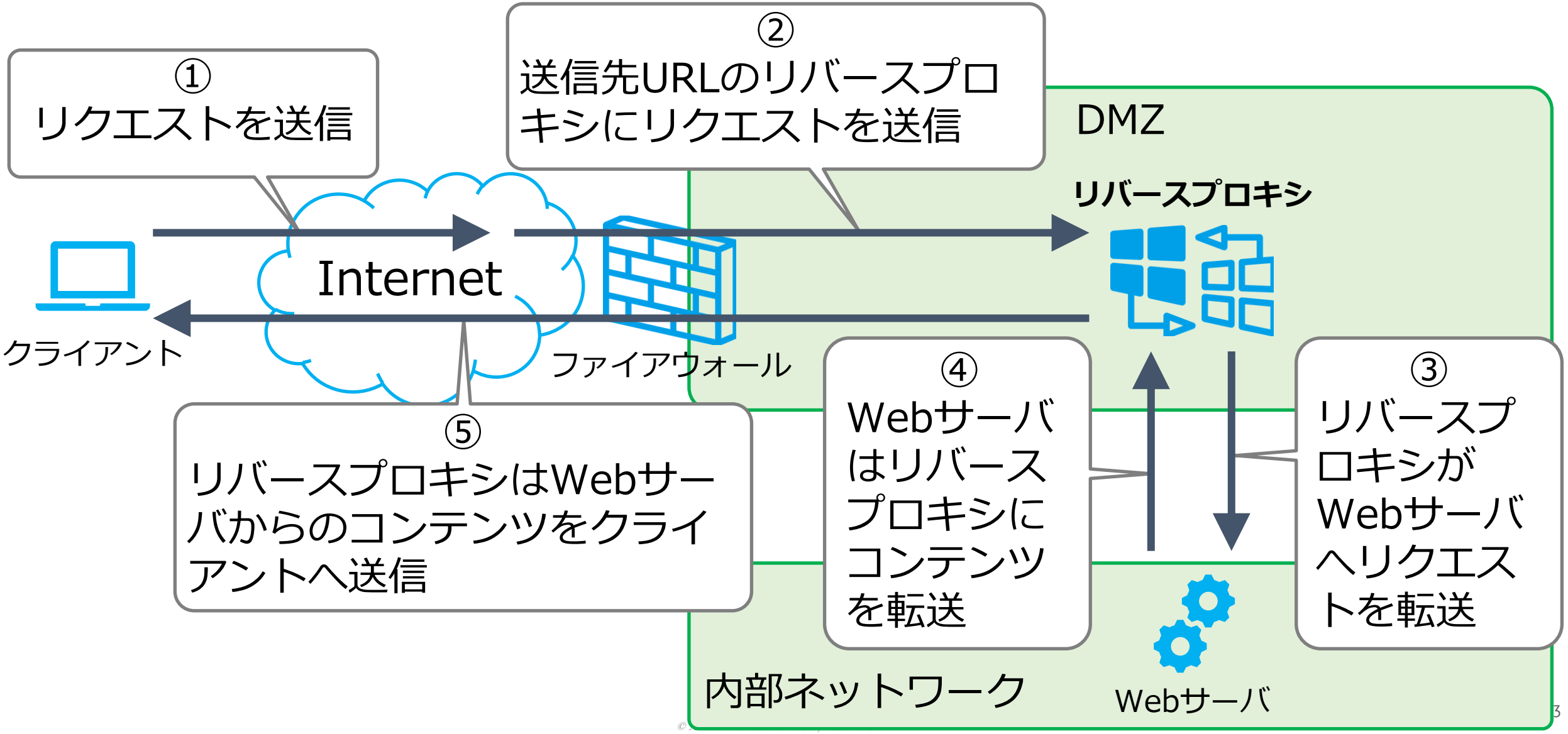
SPAの前身 ＝ Port-Knocking

※ Single Packet Authorization：SPA 詳しくは後述します。

1. はじめに
2. Zero Trust Network Access
3. Zero Trust Network Accessの歴史
4. Reverse PROXY
5. Port knocking
6. Single Packet Authorization：SPA
7. VPNとZTNA/SDPの比較
8. Zero Trust Architecture の論理構成
9. SDPの仕組み
10. ZTNAの仕組み
11. おわりに

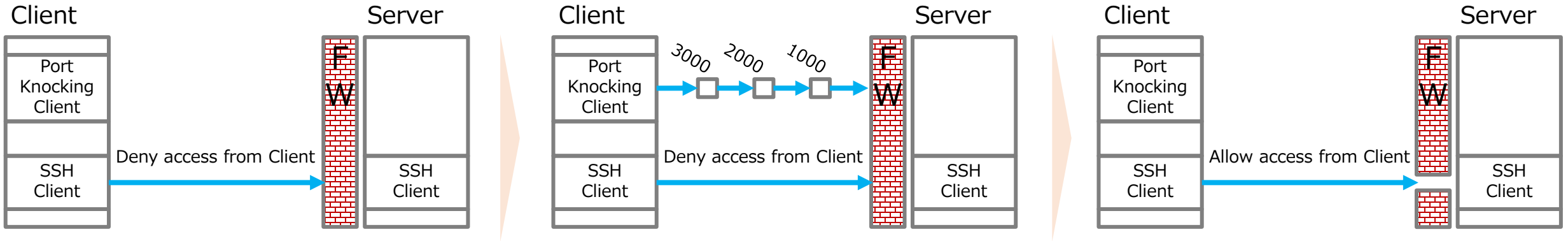
Reverse PROXY

Reverse PROXYとは、内部サーバの代理でクライアントに応答を返す仕組み



1. はじめに
2. Zero Trust Network Access
3. Zero Trust Network Accessの歴史
4. Reverse PROXY
5. Port knocking
6. Single Packet Authorization
7. VPNとZTNA/SDPの比較
8. Zero Trust Architecture の論理構成
9. SDPの仕組み
10. ZTNAの仕組み
11. おわりに

Port knocking概要



最初の状態ではポート22は閉じられています。

ClientからPort knockingに定義されたTCP SYNのシーケンスが送信される。

ServerはTCP/22を開放します。

Deny-Allの状態を維持しつつ、**決められたノックシーケンス**が送られてきた時のみアクセスを許可するということが実現できます。

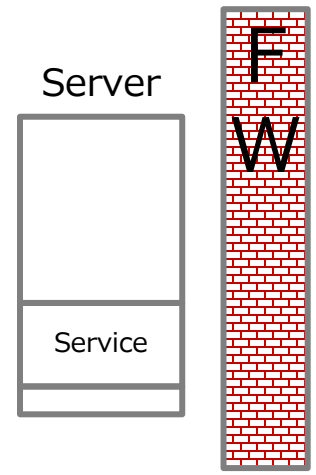
Port knockingのメリット

- パケットはデフォルト・ドロップ
- ポートスキャンに対して有効
- DoS/DDoS攻撃を最小化

Deny access from any（**Deny-all**）

Port knocking用のシーケンス
にならないパケットは**全てドロップ**

基本的にポートは閉めているため、
開きポートはスキャンできない



Port knockingのデメリット

1 リプレイ攻撃に弱い

ノックシーケンスは、**暗号化されずに**そのままネットワーク上を流れます。

2 ノックシーケンスの破壊

第三者がノックシーケンスの一部を送ることで、シーケンスそのものを破壊可能です。

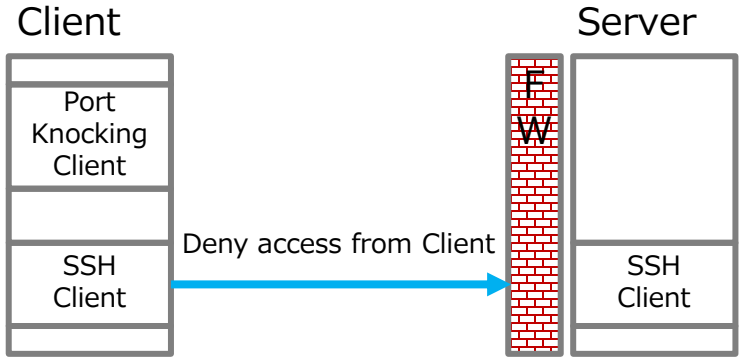
3 IDS、ポートスキャンに弱い

ノックシーケンスは、一連のパケットの流れになるので探索が可能です。

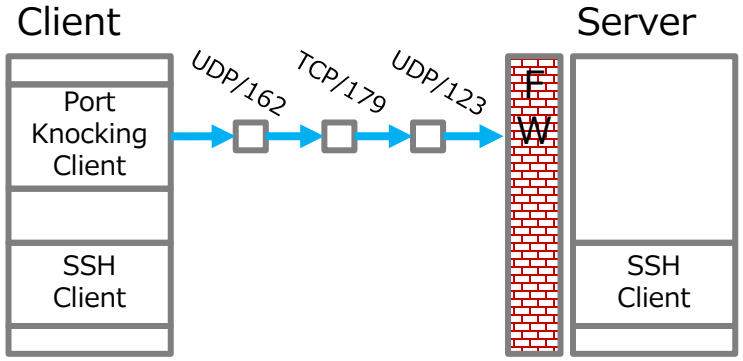
暗号化していないから丸見え

Port knockingデモ

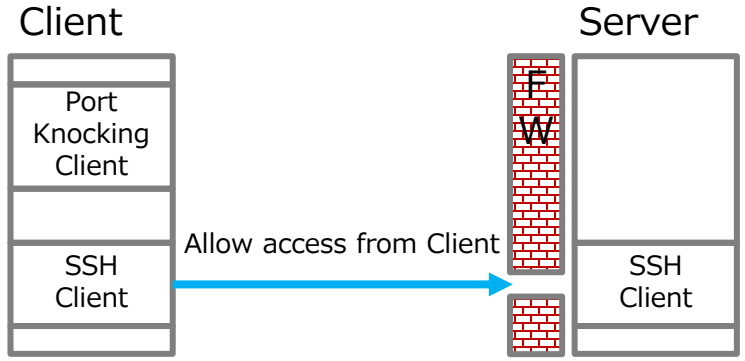
百聞は一見に如かず！



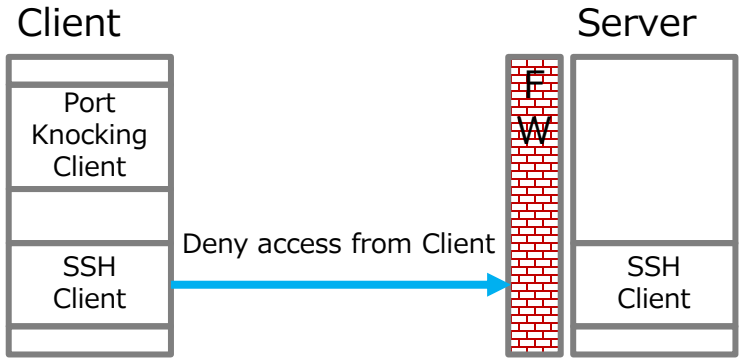
最初の状態ではポート22は閉じられています。



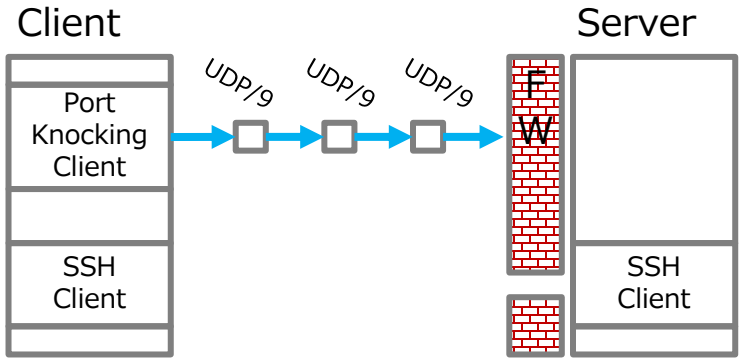
ClientからPort knockingに定義された
TCP SYNのシーケンスが送信します。



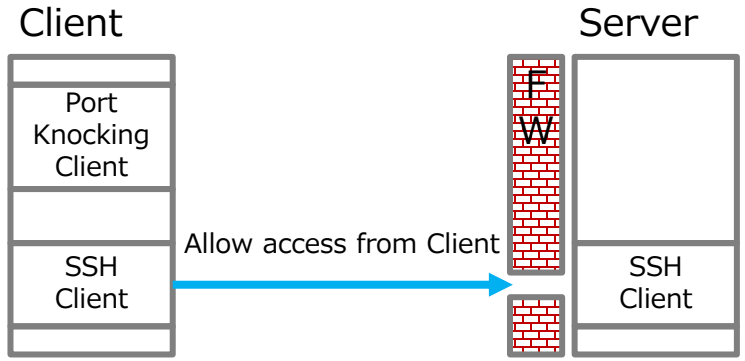
ServerはTCP/22を開放します。
Clientはこの開放している間に
TCP/22に対してアクセスします。



最初の状態に戻ります。



ClientからPort knockingに定義された
TCP SYNのシーケンスが送信します。



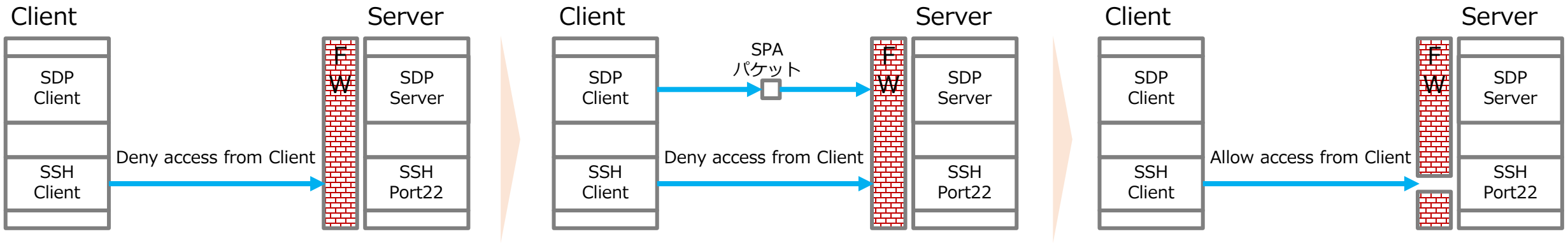
ServerのTCP/22が開放しっぱなしは
セキュリティが甘すぎます。
ClientからノックしてFWを閉めます。

Port knockingデモ

デモ動画は当日公開します

1. はじめに
2. Zero Trust Network Access
3. Zero Trust Network Accessの歴史
4. Reverse PROXY
5. Port knocking
6. Single Packet Authorization
7. VPNとZTNA/SDPの比較
8. Zero Trust Architecture の論理構成
9. SDPの仕組み
10. ZTNAの仕組み
11. おわりに

Single Packet Authorizationの概要



最初の状態ではポート22は閉じられています。

ClientがSPAパケットを送信します。

Serverは一定時間TCP/22を開放します。

Deny-Allの状態を維持しつつ、**正規のSPAパケット**が送られてきた時のみアクセスを許可するということが実現できます。

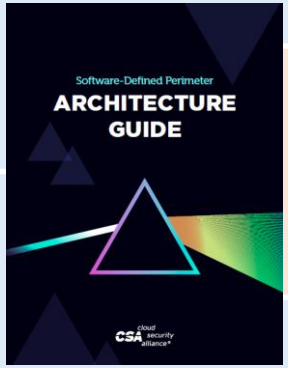
chapter “Trusting the Traffic” in Zero Trust Networks by Evan Gilman and Doug Barth (O’Reilly Media, Inc., 2017).

SPAの原則

SPAを実装するにあたっては、以下の４つの共通の原則を備えることを要求しています。

暗号化と認証

１パケット



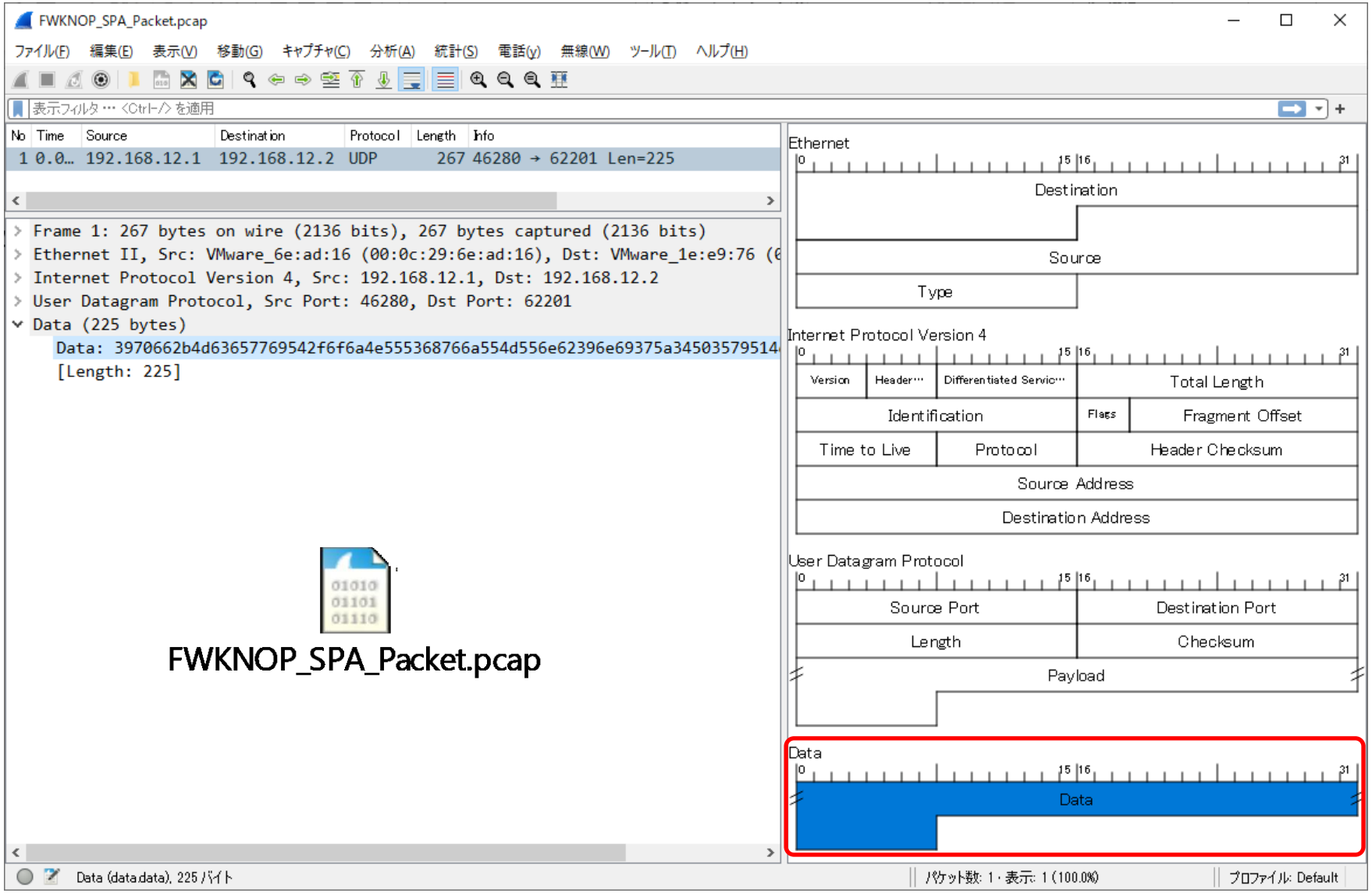
SPAの原則

管理者権限非依存

隠密

SPAパケットの構成

OSS（fwknop）のSPAパケットはUDP 1パケットです。



SPAパケットの構成

SPAパケットの構成は以下のとおりです。暗号化されて1パケットに様々な情報が含まれています。

```
SPA Field Values:
==================
   Random Value: 1471177112137289
       Username: root
      Timestamp: 1682842565
    FKO Version: 2.0.2
   Message Type: 1 (Access msg)
 Message String: 192.168.12.1,tcp/22
     Nat Access: <NULL>
    Server Auth: <NULL>
 Client Timeout: 0
    Digest Type: 3 (SHA256)
      HMAC Type: 3 (SHA256)
Encryption Type: 1 (Rijndael)
Encryption Mode: 2 (CBC)
   Encoded Data: 1471177112137289:cm9vdA:1682842565:2.0.2:1:MTkyLjE2OC4xMi4xLHRjcC8yMg
SPA Data Digest: qgklgxzGr7QdRrqmlvwir11KAnDBk2tEyzRli6bqN9I
          HMAC: FdvJhWhHJI7CYuOZEguBIfMdyA1aLFfYcJ+fuUZDMsk
 Final SPA Data:
+o639oNwQUJl92UrPr02q9Qdlu9/PWtYtzD3tB8Jo+Ey0Bftbf8r0qC6qYNg5d5z+4pquI+oaV1+hIFy9Ich/1qWHZvfGtrTOqBn2osg+Xo9CpdpmFLMllfIczh
O/8QrLaDskF52YiAS0LFBH5Rinkid33Sl+wCF+DhWRHEtIXw67UlL7orFDSFdvJhWhHJI7CYuOZEguBIfMdyA1aLFfYcJ+fuUZDMsk

Generating SPA packet:
          protocol: udp
       source port: <OS assigned>
   destination port: 62201
          IP/host: 192.168.12.2
send_spa_packet: bytes sent: 225
```

①
どんな通信か？

②
暗号化され、ユニークなパケット

SPAの特徴

SPAの特徴は以下のとおりです。

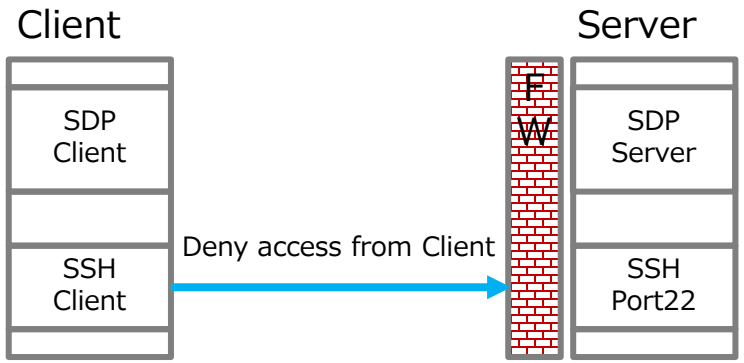
特徴

Deny-All	SPAパケットでないものは全てドロップ
対リプレイ攻撃	過去に来たパケットが来た場合にリプレイ攻撃であると判断
対DDoS攻撃	シーケンスではないのでDDoS攻撃の可能性を低減 詳しくは※を参照ください。
対スニッフィング	1パケットであることから、スニッフィングすることが困難
認証の強化	クライアントのユーザ名など、ユーザに対応した追加の認証等の処理が可能

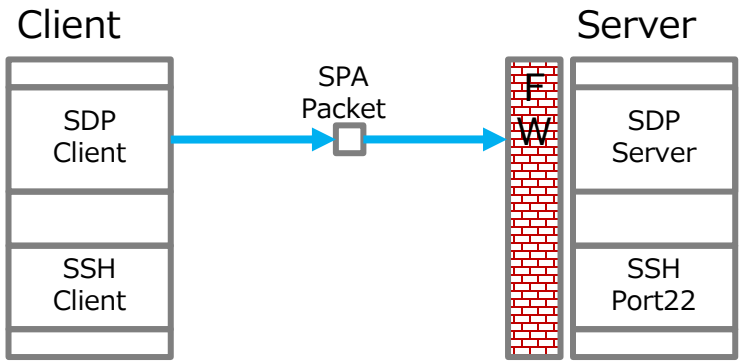
※ Software-Defined Perimeter as a DDoS Prevention Mechanism
https://cloudsecurityalliance.org/artifacts/software-defined-perimeter-as-a-ddos-prevention-mechanism/

SPAのデモ

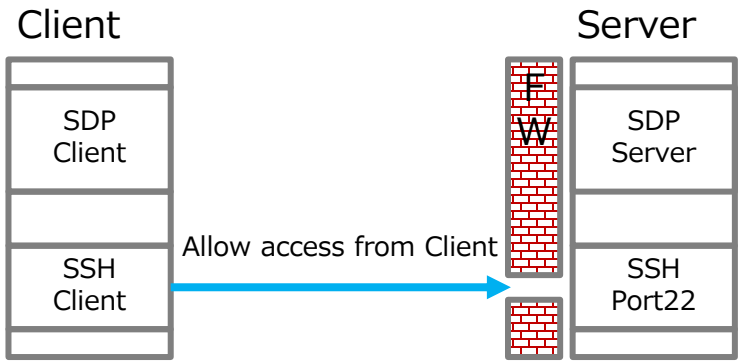
百聞は一見に如かず！



最初の状態ではポート22は閉じています。



ClientがSPAパケットを送信します。



有効なSPAパケットを受信したServerは一定時間TCP/22を開放します。
Clientはこの開放している時間内にTCP/22に対してアクセスします。

Port knockingと異なり制限時間を経過するとポートが自動的に閉められます。

SPAのデモ

デモ動画は当日公開します

サマリー(ここまでのまとめ)

1

Zero Trust Network Accessは2種類ある

PROXY-Based の**Service-Initiated 型 ZTNA**と、SDP-Based の **Endpoint-Initiated 型 ZTNA** の2種類があります。

2

ZTNAの中核技術はReverse PROXY

内部サーバの代理でクライアントに応答を返す仕組みのことです。

3

Port-Knocking

Deny-Allの状態を維持しつつ、決められたパケットシーケンスによりファイアウォールを開閉する仕組みです。**暗号化されていないのが最大のデメリット**です。

4

Single Packet Authorization

Port-Knockingの利点を活かしつつ、Port-Knockingの課題に対処したものです。
SDPの最も重要な要素の１つである「**接続前認証**」を実現するために使われます。

1. はじめに
2. Zero Trust Network Access
3. Zero Trust Network Accessの歴史
4. Reverse PROXY
5. Port knocking
6. Single Packet Authorization
7. VPNとZTNA/SDPの比較
8. Zero Trust Architecture の論理構成
9. SDPの仕組み
10. ZTNAの仕組み
11. おわりに

VPNの攻撃例

認証に関する攻撃を多く受けます。以下に代表的な攻撃例を示します。

1

ブルートフォース攻撃

ログインIDを固定して、パスワードを総当たり的に試行する攻撃

2

リバースブルートフォース攻撃

パスワードを固定して、ログインIDを総当たり的に試行する攻撃

3

パスワードスプレイ攻撃

同じパスワードを複数のアカウントに対して同時に試行する攻撃

VPNの脆弱性

VPNは脆弱性が大変多いです。

2022年最も悪用された脆弱性TOP12（CISA※）

1位

2019年5月
修正済み

CVE-2018-13379
Fortinet社のVPN機器FortiG

2020年、2021年
もランクイン！！

https://www.bleepingcomputer.com/news/security/fbi-cisa-and-nsa-reveal-top-exploited-vulnerabilities-of-2022/
※ Cybersecurity and Infrastructure Security Agency, CISA

2023年の新しい脆弱性

Citrix NetScaler の脆弱性
CVE-2023-3519
CVE-2023-4966

SonicWall の脆弱性
CVE-2023-41712
CVE-2023-41715

Fortinet の脆弱性
CVE-2023-27997

SoftEther VPN
PacketiX VPNの脆弱性
CVE-2023-27395
CVE-2023-22325
CVE-2023-32275
CVE-2023-27516
CVE-2023-32634
CVE-2023-31192

VPNの脆弱性と攻撃の可能性 **TunnelCrack 攻撃**
CVE-2023-36672 CVE-2023-35838
CVE-2023-36673 CVE-2023-36671

Cisco AnyConnect の脆弱性
CVE-2023-36672 CVE-2023-36673

Wireguard の脆弱性
CVE-2023-35838

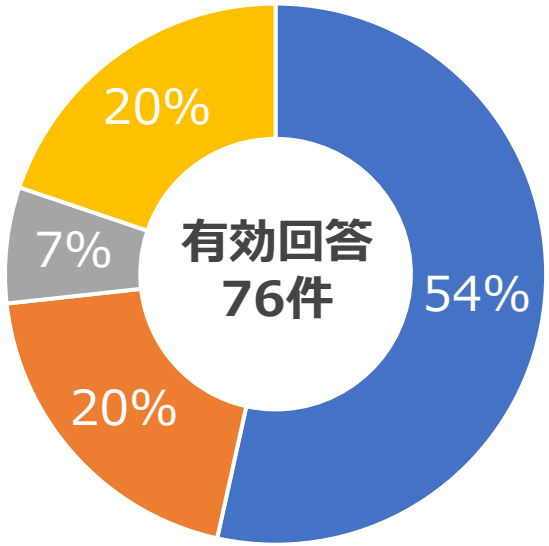
BIG-IP の脆弱性
CVE-2023-43125

Clario VPN の脆弱性
CVE-2023-36671 CVE-2023-36672

VPNの脆弱性

ランサムウェア被害の多くはVPNからの侵入が多い傾向にあります。

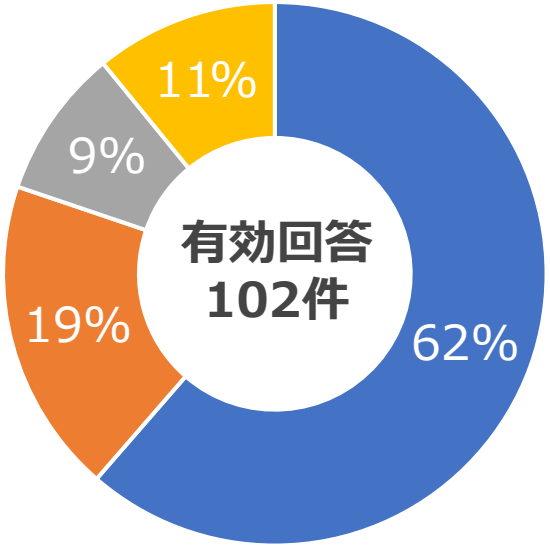
ランサムウェアの感染経路
（2021年）



- VPN機器からの侵入
- リモートデスクトップからの侵入
- 不審メールやその添付ファイル
- その他

（出典）警察庁「令和３年におけるサイバー空間をめぐる脅威の情勢等について」

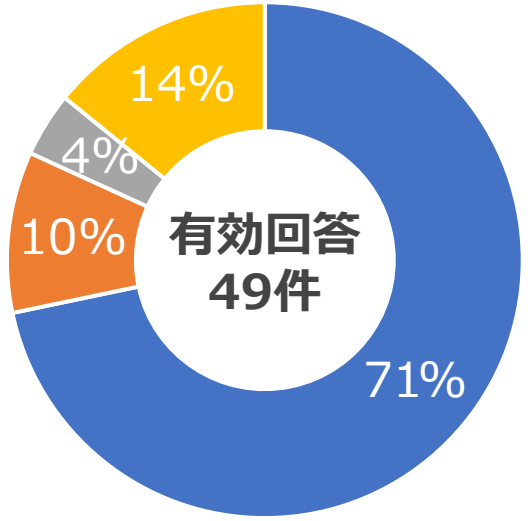
ランサムウェアの感染経路
（2022年）



- VPN機器からの侵入
- リモートデスクトップからの侵入
- 不審メールやその添付ファイル
- その他

（出典）警察庁「令和４年におけるサイバー空間をめぐる脅威の情勢等について」

ランサムウェアの感染経路
（2023年上半期）



- VPN機器からの侵入
- リモートデスクトップからの侵入
- 不審メールやその添付ファイル
- その他

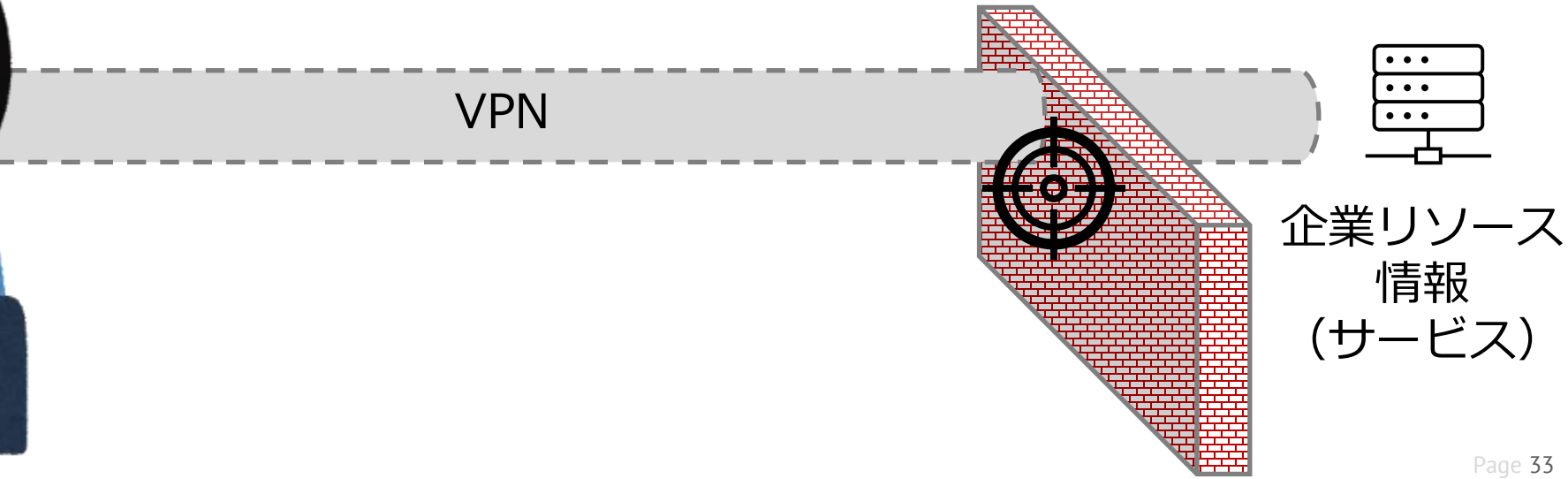
（出典）警察庁「令和５年上半期におけるサイバー空間をめぐる脅威の情勢等について」

VPNが狙われる理由

何故、VPNが狙われてしまうのでしょうか。

認証は最初だけ

PPTP ＝ 1723/TCP
L2TP ＝ 1701/UDP
IPSec ＝ 500/UDP、4500/UDP
OpenVPN ＝ 1194/UDP、443/TCP
各種メーカー固定のポート番号
（定義変更によりDefaultから変更可能）



ZTNAの認証は複雑

ZTNAの認証はアカウント情報とパスワードだけの認証ではありません。

アカウント

行動属性(時間)

位置情報

環境属性

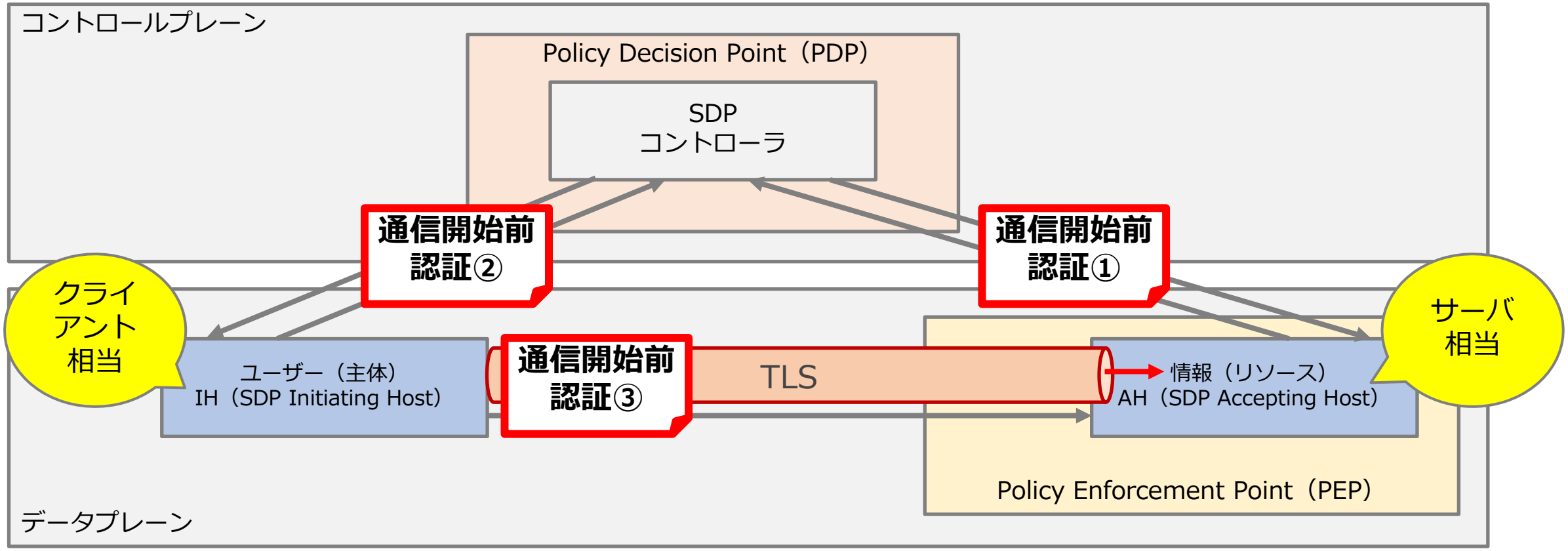
多要素認証

デバイス

パスワード

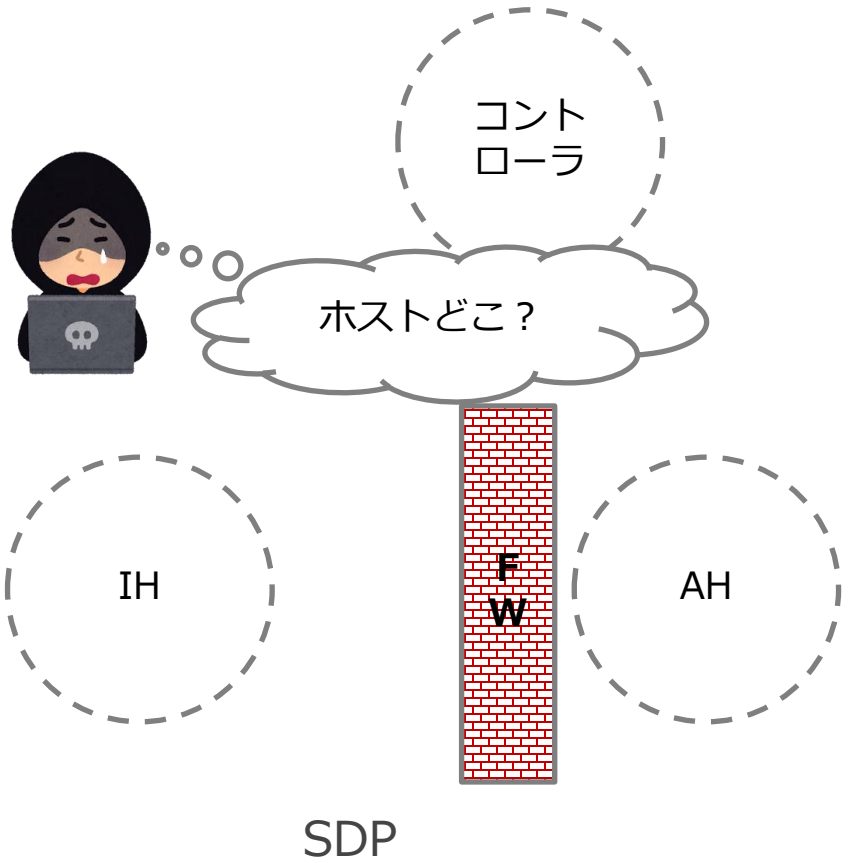
VPNとSDPの違い

何度も異なる認証プロセスを行います。

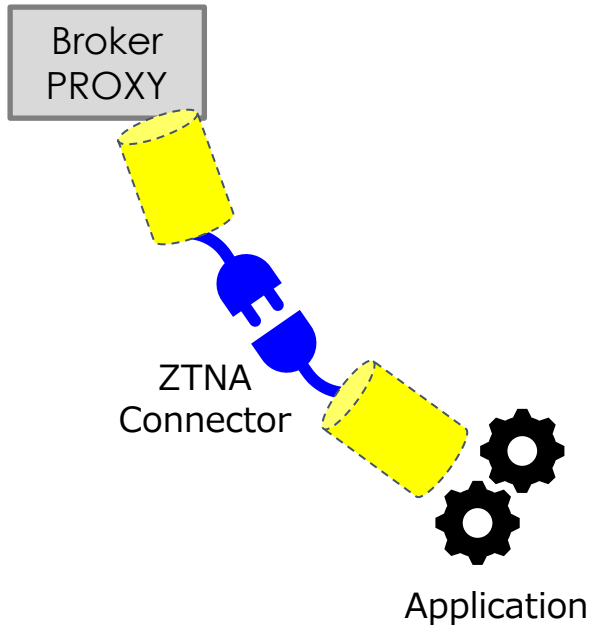


詳しくは後ほど説明します。

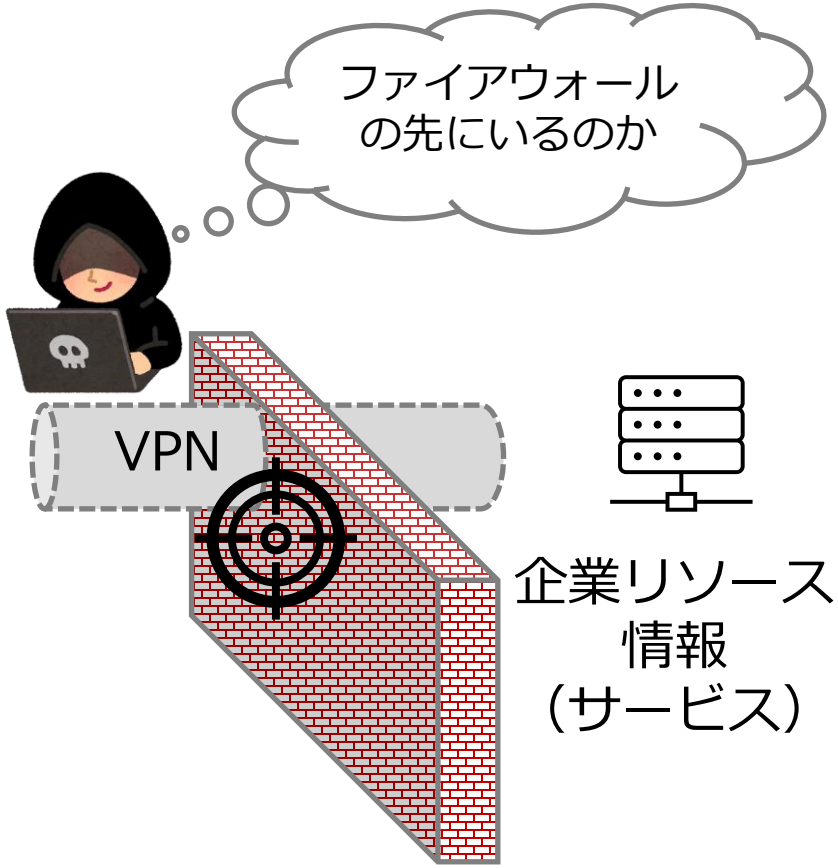
ZTNA / SDPはシステムを隠蔽します。



有効なSPAパケットを受信するまで応答しません。



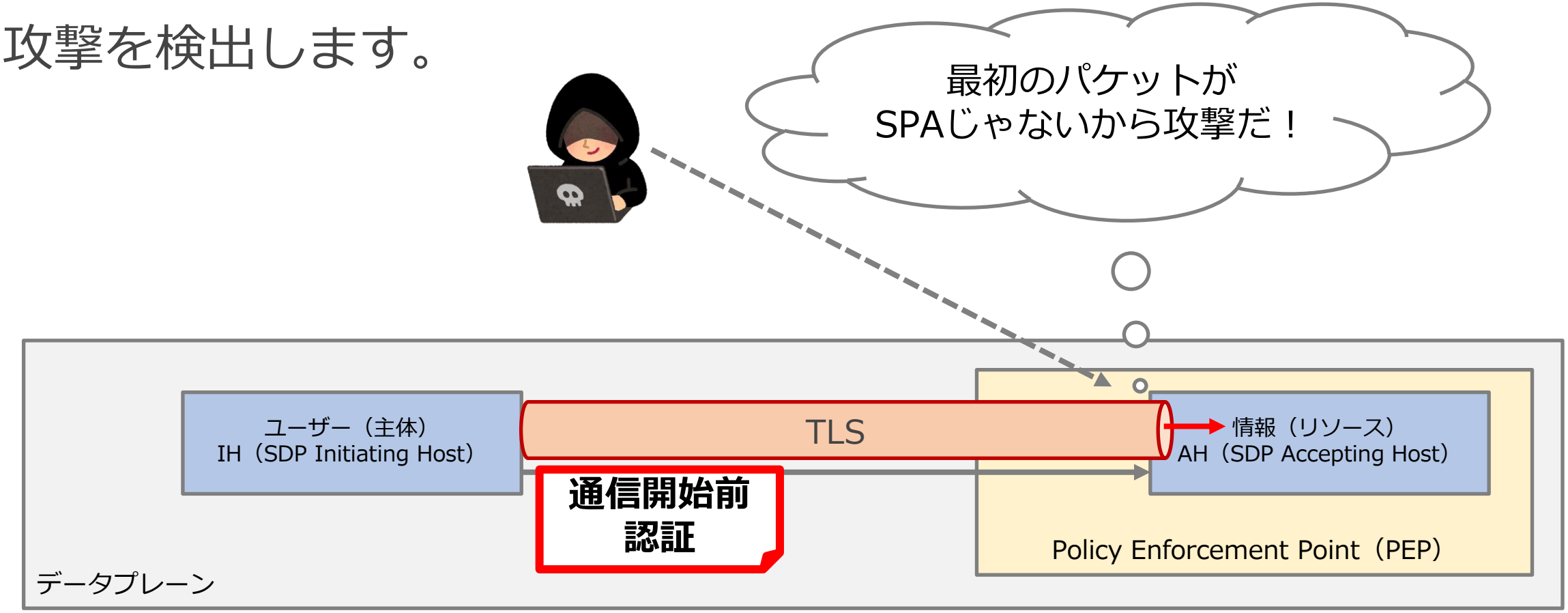
Reverse PROXYによってアプリケーションを隠蔽します。



ファイアウォールの先にサービスが存在することが分かる。

SDPは攻撃を検出します

攻撃を検出します。



最初のパケットはSPAパケットである必要があります。
最初のパケットでSPAパケットを受信しなかった場合は攻撃とみなされます。
1パケットで攻撃か否かを判断できます。

ZTNAは攻撃を検出します

攻撃を検出し、防御します。

予測力

回復力

抵抗力

適応力

Trustworthy
より信頼できる
システム

セキュリティーベンダー（サービス提供者）は、攻撃対象領域（Attack Surface）になり得ます。

Broker
PROXY

Outbound接続のみ許可であるため、外部からの攻撃リスクを削減します。

ZTNA
Connector

End-User

Device

Application

Attack Surface の違い

ゼロトラストは「Need to Knowの原則」でセキュリティを担保

VPN接続イメージ図

許可レベル：IPレベル
情報提供　：かなり多くの情報

ZTNA / SDP 接続イメージ図

許可レベル：アプリケーションレベル
情報提供　：必要最低限の情報

VPNのAttack Surface（攻撃対象領域）

ここまでの話を踏まえてVPNの危険性を想像してください。

Attack Surface 攻撃対象領域

By Type（タイプ別）	By Exposure（露出）	By Attack Vectors（攻撃ベクトル）
デジタル攻撃面	外部攻撃面	認証情報の漏洩または盗難
ソーシャルエンジニアリング攻撃面		弱い認証情報
物理的攻撃面	内部攻撃面	ソフトウェアの脆弱性（CVE）
		暗号化が欠落、若しくは不十分
		認証が欠落、若しくは不十分
		設定ミス
		フィッシング
		悪意のあるインサイダー
		信頼関係
		DoS

外部からの直接攻撃も！

ラテラルムーブメントの容易さ

VPNとZTNA / SDPの違い

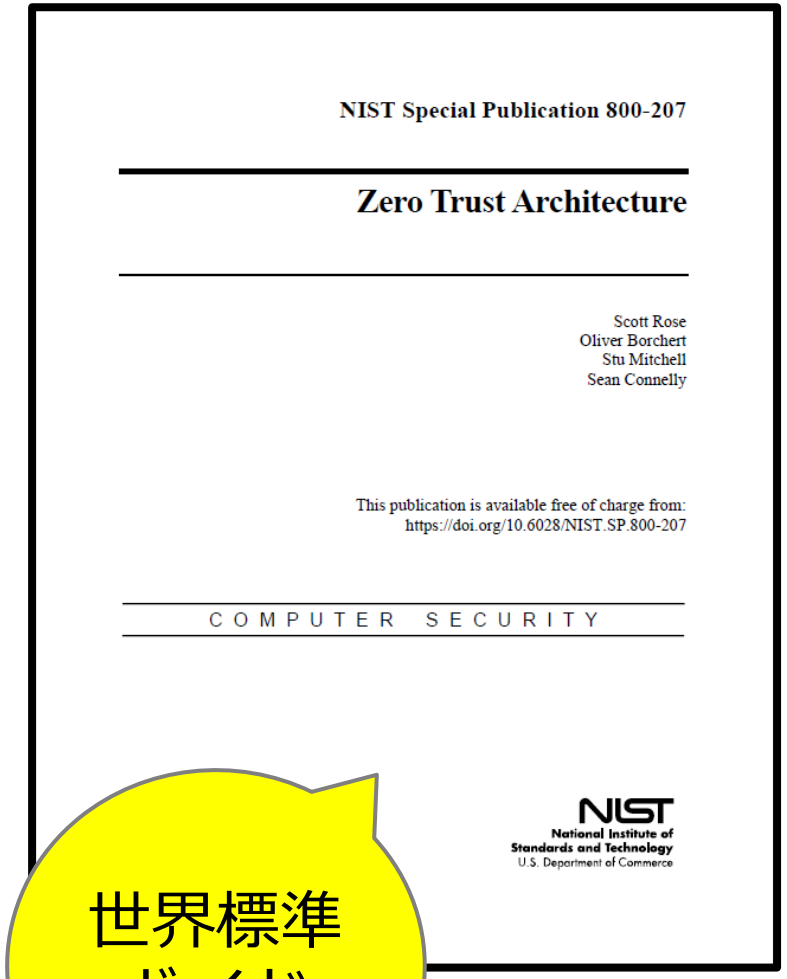
ここまでの比較をまとめます。

	ＶＰＮ	ZTNA	ＳＤＰ
認証の複雑性	一般的に1度かつ単純なログイン検証 ×	様々な条件を組合せてログインユーザを多角的に検証 ○	複数回の異なる認証プロセス ○
システム隠蔽性	ファイアウォールのポート固定 ×	PROXYによりシステム隠蔽 ○	Deny-allでシステムを隠蔽 ○
攻撃対象領域(Attack Surface)	情報過多 ×	必要最低限の情報提供 ○	必要最低限の情報提供 ○
攻撃の検出	なし ×	プロバイダーとして攻撃を検出する機構が備わっている ○	1stパケットはSPAのみで、攻撃を検出します ○
総合評価	×	○	○

1. はじめに
2. Zero Trust Network Access
3. Zero Trust Network Accessの歴史
4. Reverse PROXY
5. Port knocking
6. Single Packet Authorization
7. VPNとZTNA/SDPの比較
8. Zero Trust Architecture の論理構成
9. SDPの仕組み
10. ZTNAの仕組み
11. おわりに

Zero Trust Architectureとは？

Zero Trust Architecture（NIST SP800-207）とは一体何か。


NIST Special Publication 800-207

Zero Trust Architecture

Scott Rose
Oliver Borchert
Stu Mitchell
Sean Connelly

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-207

COMPUTER SECURITY

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

世界標準ガイド

- 序章
- ゼロトラストの考え方
- ゼロトラスト・アーキテクチャの論理的構成要素
- 導入シナリオ／ユースケース
- ゼロトラスト・アーキテクチャに関連する脅威
- 米国政府のガイダンスとの整合性や関連性
- ゼロトラスト・アーキテクチャへの移行

ゼロトラストが生まれた背景

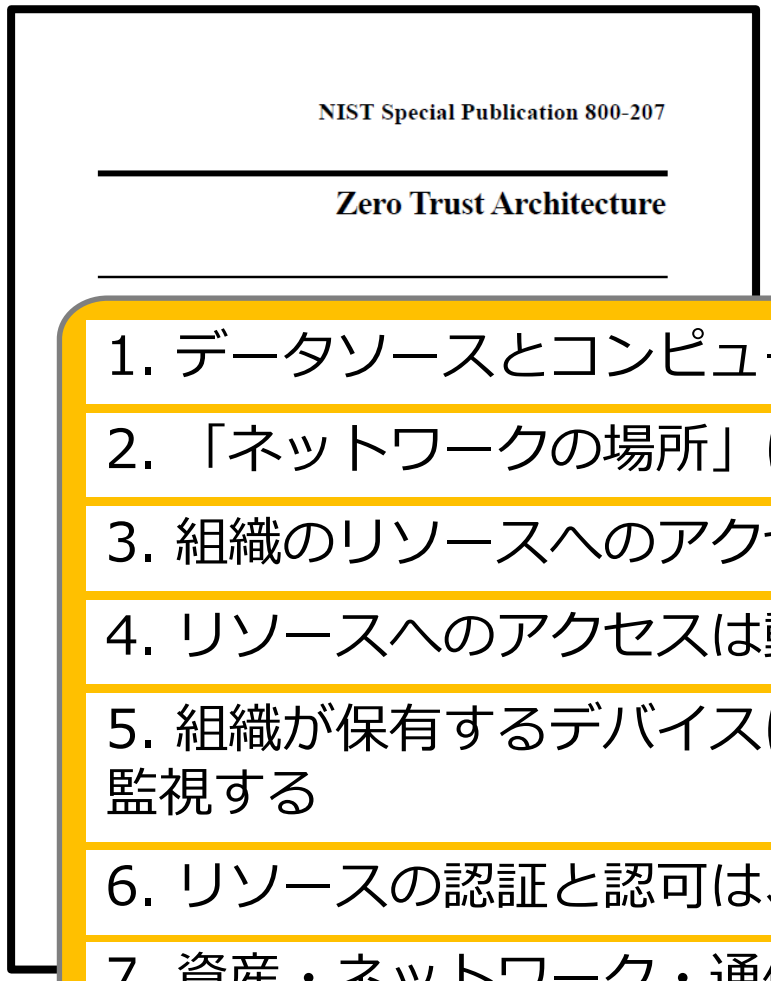
ゼロトラストの基本的な考え方

ゼロトラストの実践方法

ゼロトラストを実践・学習する最良のドキュメント

Zero Trust Architectureの勘所①

大変重要なゼロトラストにおける7つの基本原則！！


NIST Special Publication 800-207

Zero Trust Architecture

序章

ゼロトラストの考え方

ゼロトラストが生まれた背景

トラス
基本的
え方

トラス
実践方
法

1. データソースとコンピュータサービスは、全てリソースと見なす
2. 「ネットワークの場所」に関係なく、通信は全て保護される
3. 組織のリソースへのアクセスは、全て個別のセッションごとに許可される
4. リソースへのアクセスは動的なポリシーによって決定される
5. 組織が保有するデバイスは、全て正しくセキュリティが保たれているように継続的に監視する
6. リソースの認証と認可は、全てアクセスが許可される前に動的かつ厳密に実施される
7. 資産・ネットワーク・通信の状態について可能な限り多くの情報を収集し、セキュリティを高めるために利用する

ゼロトラストを実践・学習する最良のドキュメント

Zero Trust Architectureの勘所②

ゼロトラストアーキテクチャの大変重要な論理構成図！！

NIST Special Publication 800-207

Zero Trust Architecture

ゼロトラストの考え方

ゼロトラスト・アーキテクチャの論理的構成要素

ゼロトラストが生まれた背景

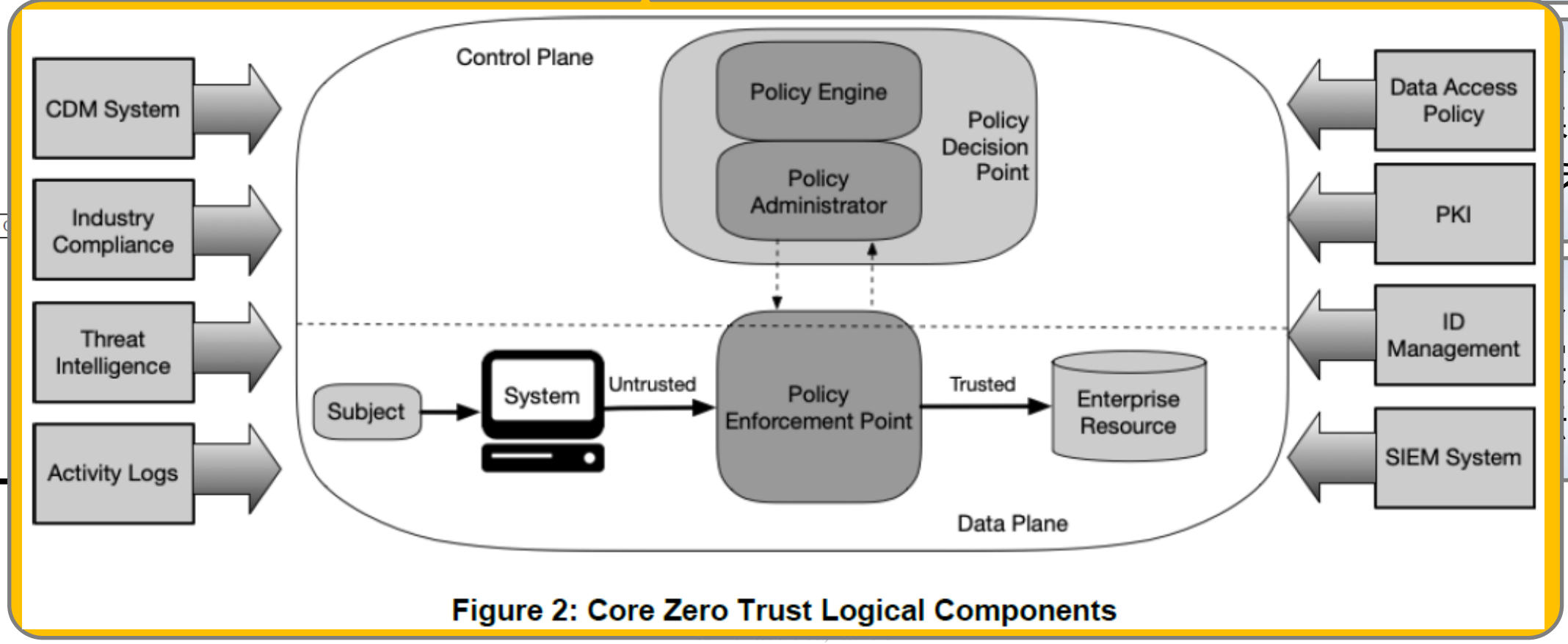
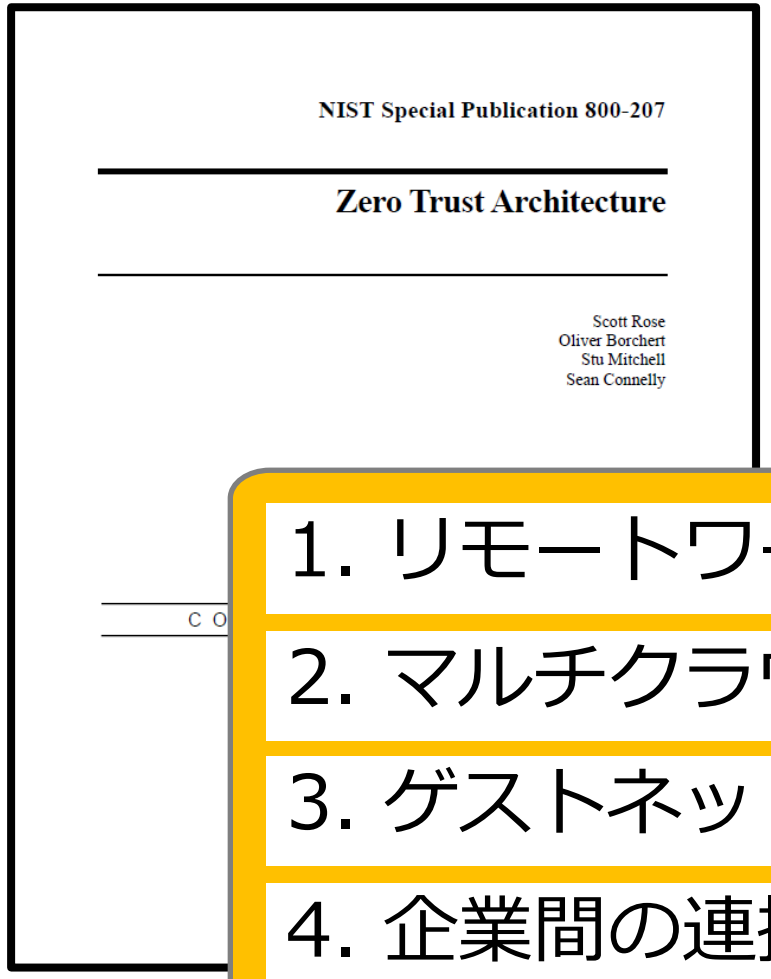


Figure 2: Core Zero Trust Logical Components

Zero Trust Architectureの勘所③

様々な展開シナリオや事例を紹介！


NIST Special Publication 800-207

Zero Trust Architecture

Scott Rose
Oliver Borchert
Stu Mitchell
Sean Connelly

C O

ゼロトラストの考え方

ゼロトラスト・アーキテクチャの論理的構成要素

導入シナリオ／ユースケース

1. リモートワークにおける一般的なユースケース
2. マルチクラウドを利用するユースケース
3. ゲストネットワークに関するユースケース
4. 企業間の連携に関するユースケース

ゼロトラストが生まれた背景

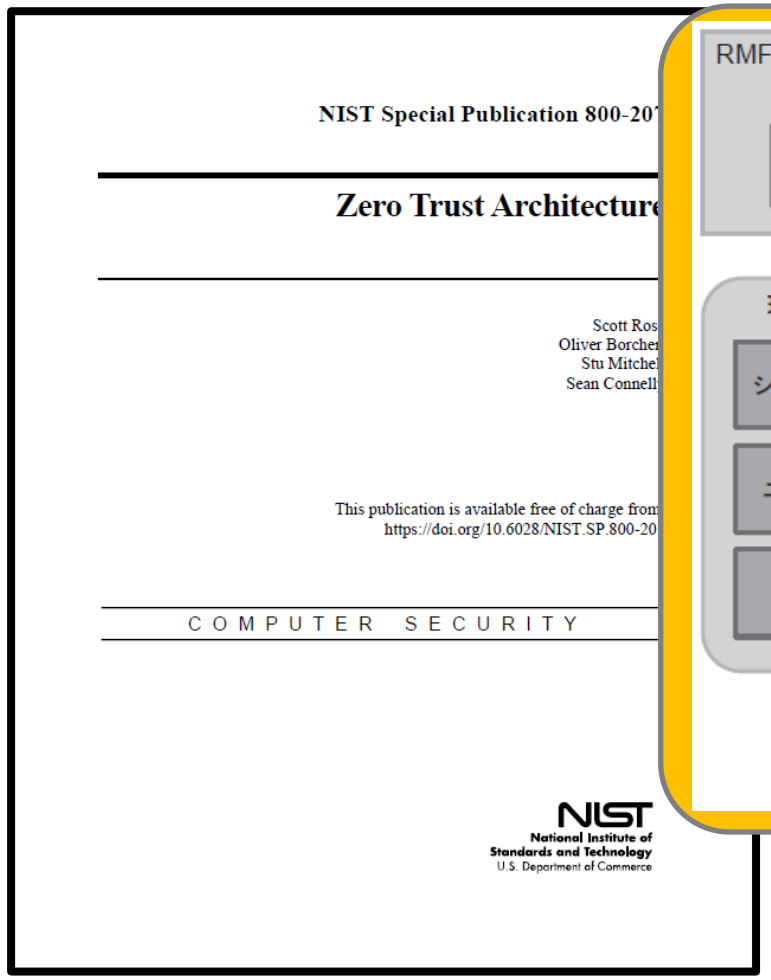
ゼロトラストの基本的な考え方

ゼロトラストの実践方法

ゼロトラストを実践・学習する最良のドキュメント

Zero Trust Architectureの勘所④

既存ネットワークからのゼロトラスト移行方法の紹介！！


NIST Special Publication 800-207

Zero Trust Architecture

Scott Ros
Oliver Borcher
Stu Mitchel
Sean Connelly

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-207

COMPUTER SECURITY

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

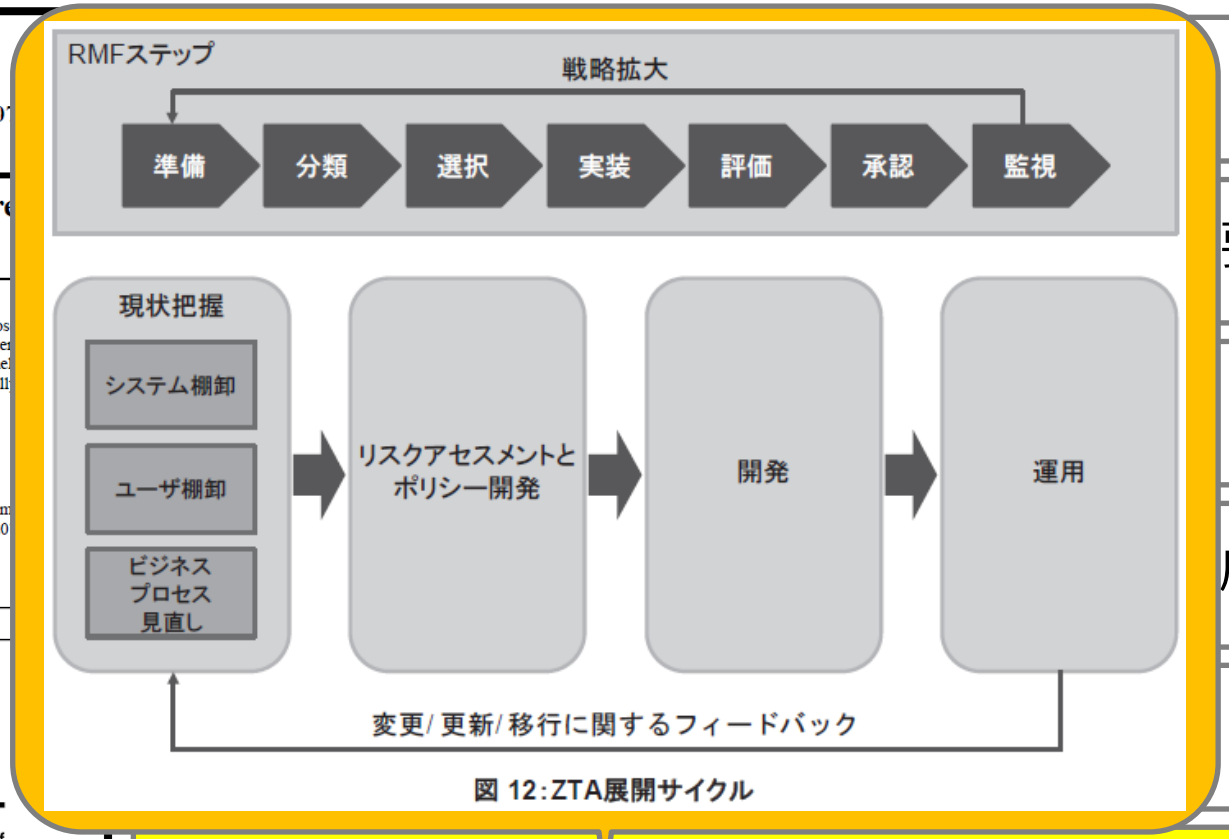


図 12:ZTA展開サイクル

要素

威

ゼロトラスト・アーキテクチャへの移行

ゼロトラストが生まれた背景

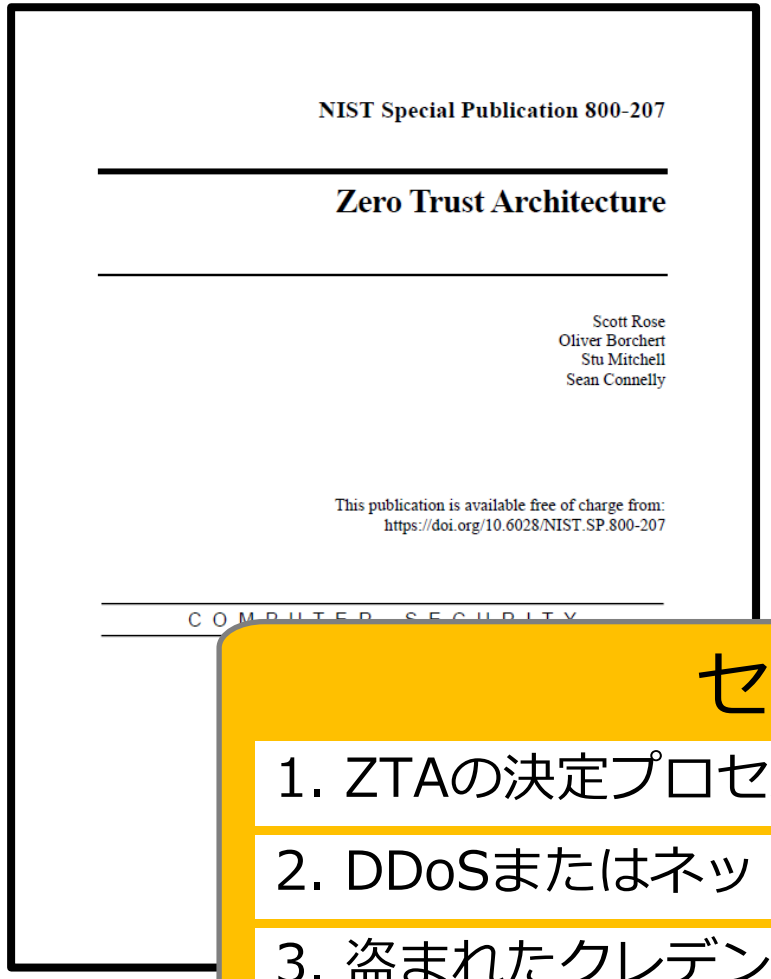
ゼロトラストの基本的な考え方

ゼロトラストの実践方法

ゼロトラストを実践・学習する最良のドキュメント

Zero Trust Architectureの勘所⑤

ゼロトラスト・アーキテクチャに関連する脅威！！


NIST Special Publication 800-207

Zero Trust Architecture

Scott Rose
Oliver Borchert
Stu Mitchell
Sean Connelly

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-207

COMPUTER SECURITY

- ゼロトラストの考え方
- ゼロトラスト・アーキテクチャの論理的構成要素
- 導入シナリオ／ユースケース
- ゼロトラスト・アーキテクチャに関連する脅威

ゼロトラストが生まれた背景

ゼロトラストの基本的な考え方

トラス
実践方
法

セキュリティに関する７つの脅威

1. ZTAの決定プロセスの転覆	5．システムとネットワーク情報の保存
2. DDoSまたはネットワーク障害	6．独自データフォーマットやソリューションへの依存
3. 盗まれたクレデンシャル/内部の脅威	
4. ネットワーク上の可視性	7．ZTA管理におけるNPEの利用

ゼロトラストを実践・学習する最良のドキュメント

Zero Trust Architecture の論理構成

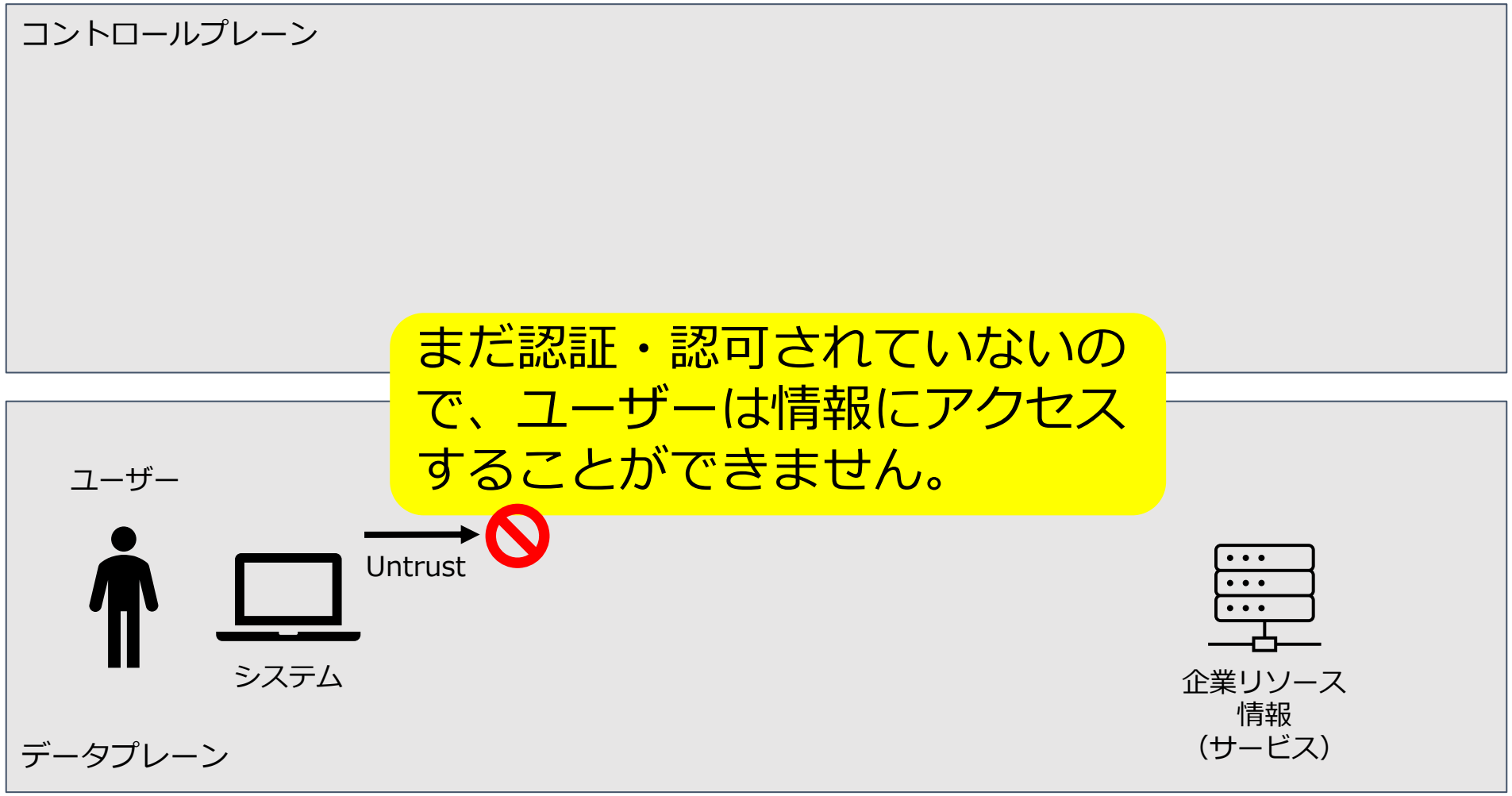
Zero Trust Architecture 論理構成はコントロールプレーンとデータプレーンに分かれています。

コントロールプレーン（脳）

データプレーン（手・脚）

Zero Trust Architecture の論理構成

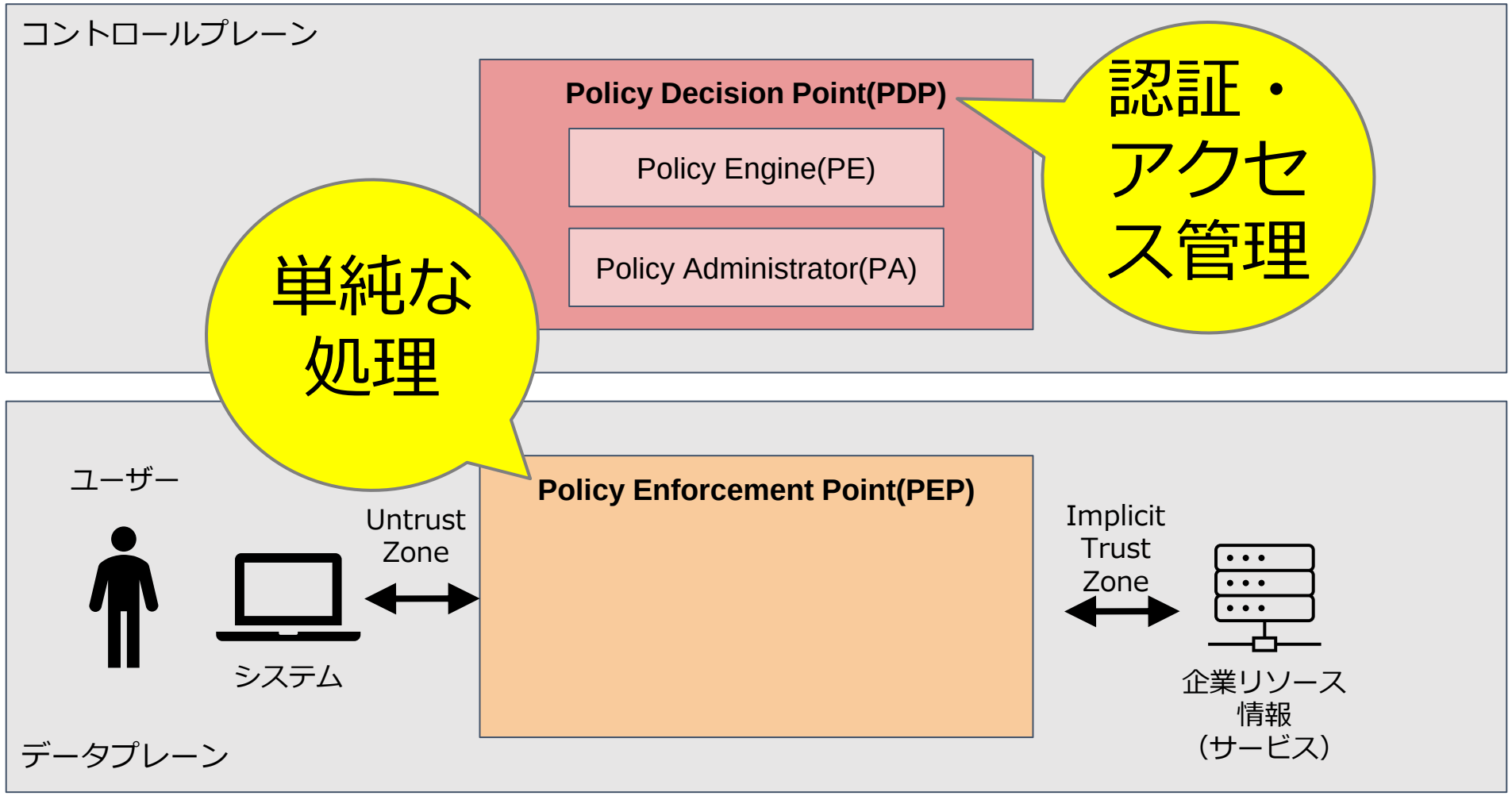
実際にデータをやり取りするデータプレーンと、認証・認可を行うコントロールプレーンを分けて考えると以下のようになります。



Zero Trust Architecture の論理構成

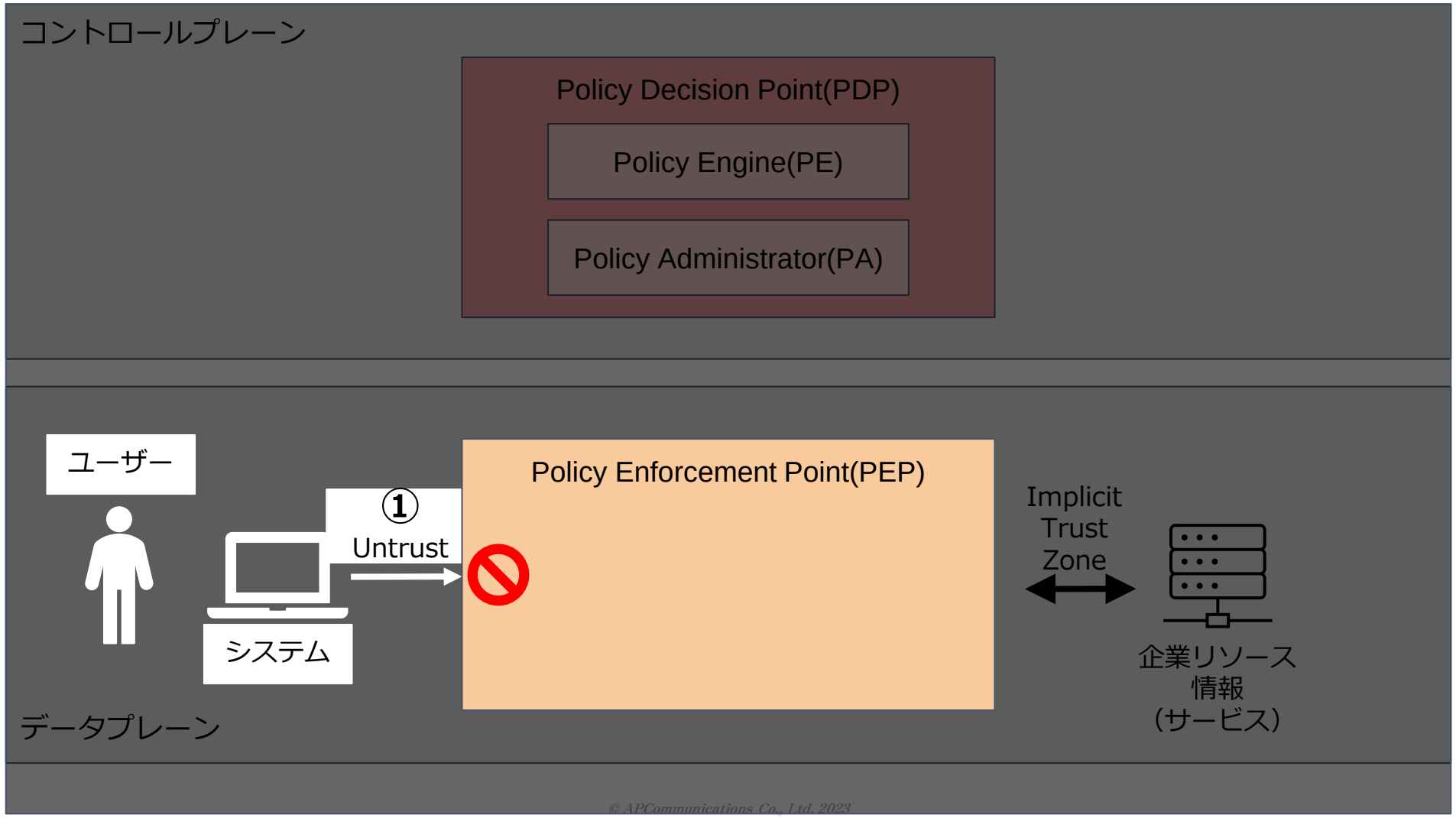
コントロールプレーン、データプレーンの名称を以下に示します。

※ Zero Trust Architecture（NIST SP800-207）で示されている名称に合わせます。



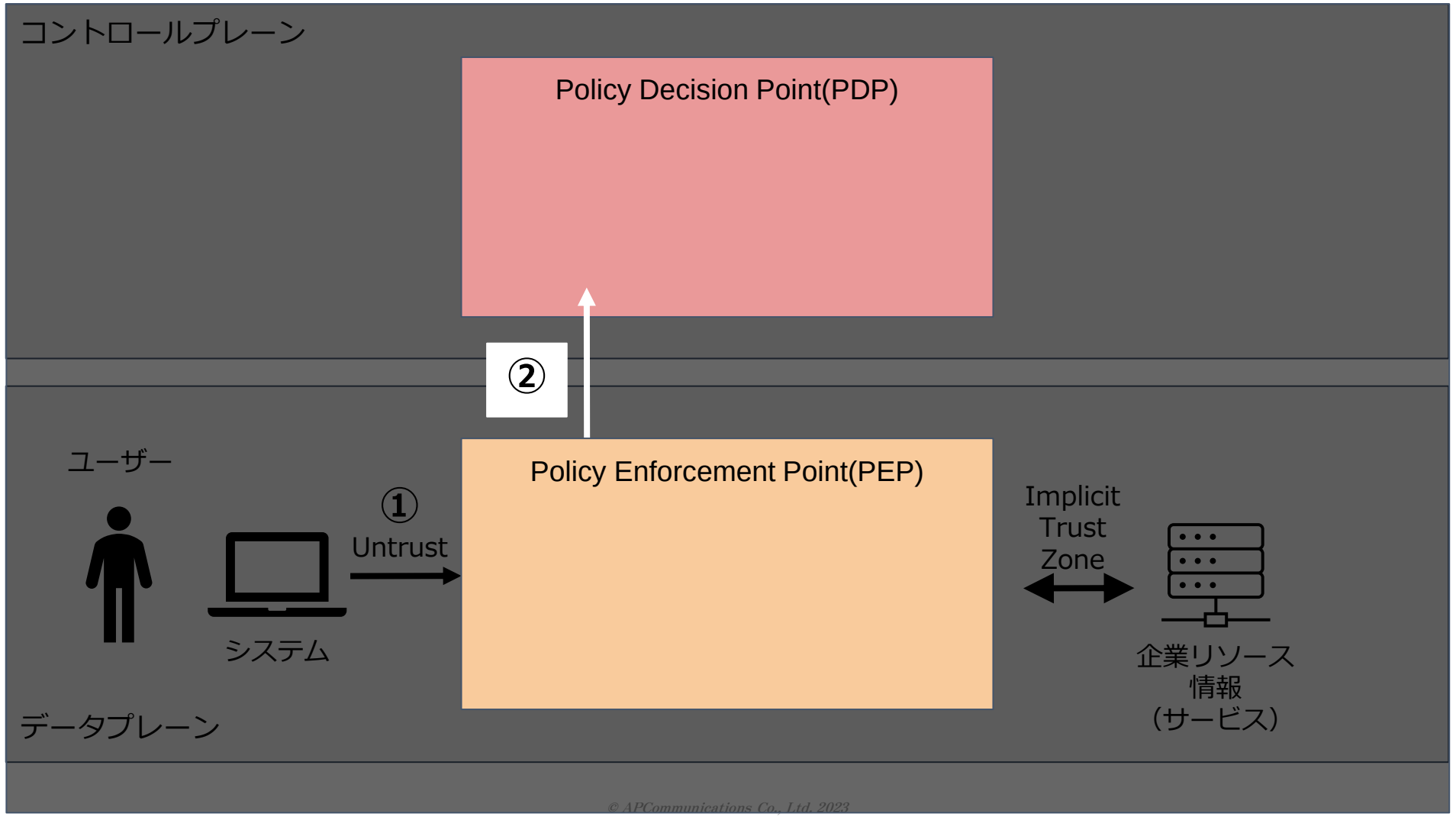
Zero Trust Architecture の論理構成

ユーザーがゼロトラストネットワークで企業リソースにアクセスする流れを以下に示します。



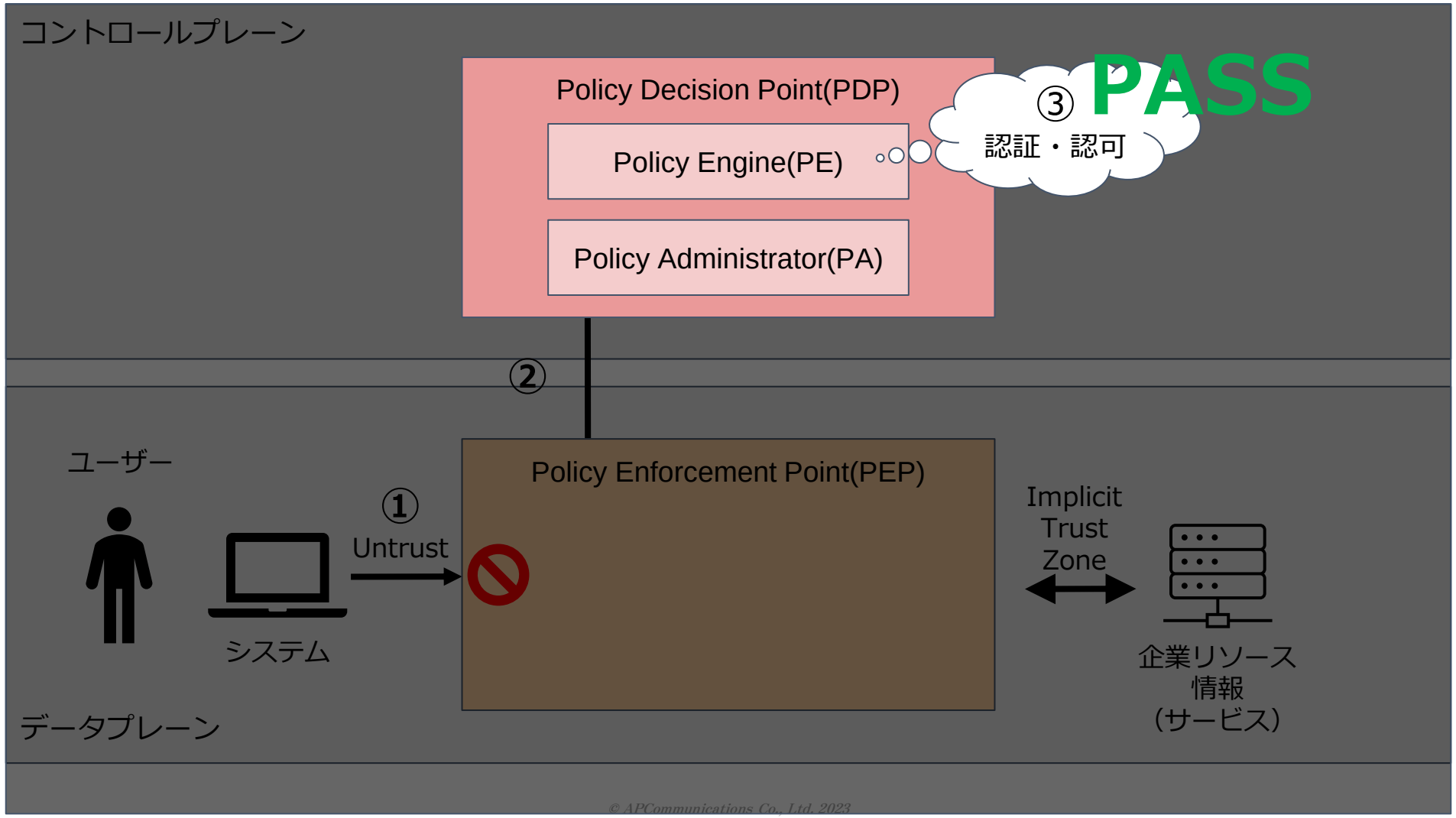
Zero Trust Architecture の論理構成

ユーザーがゼロトラストネットワークで企業リソースにアクセスする流れを以下に示します。



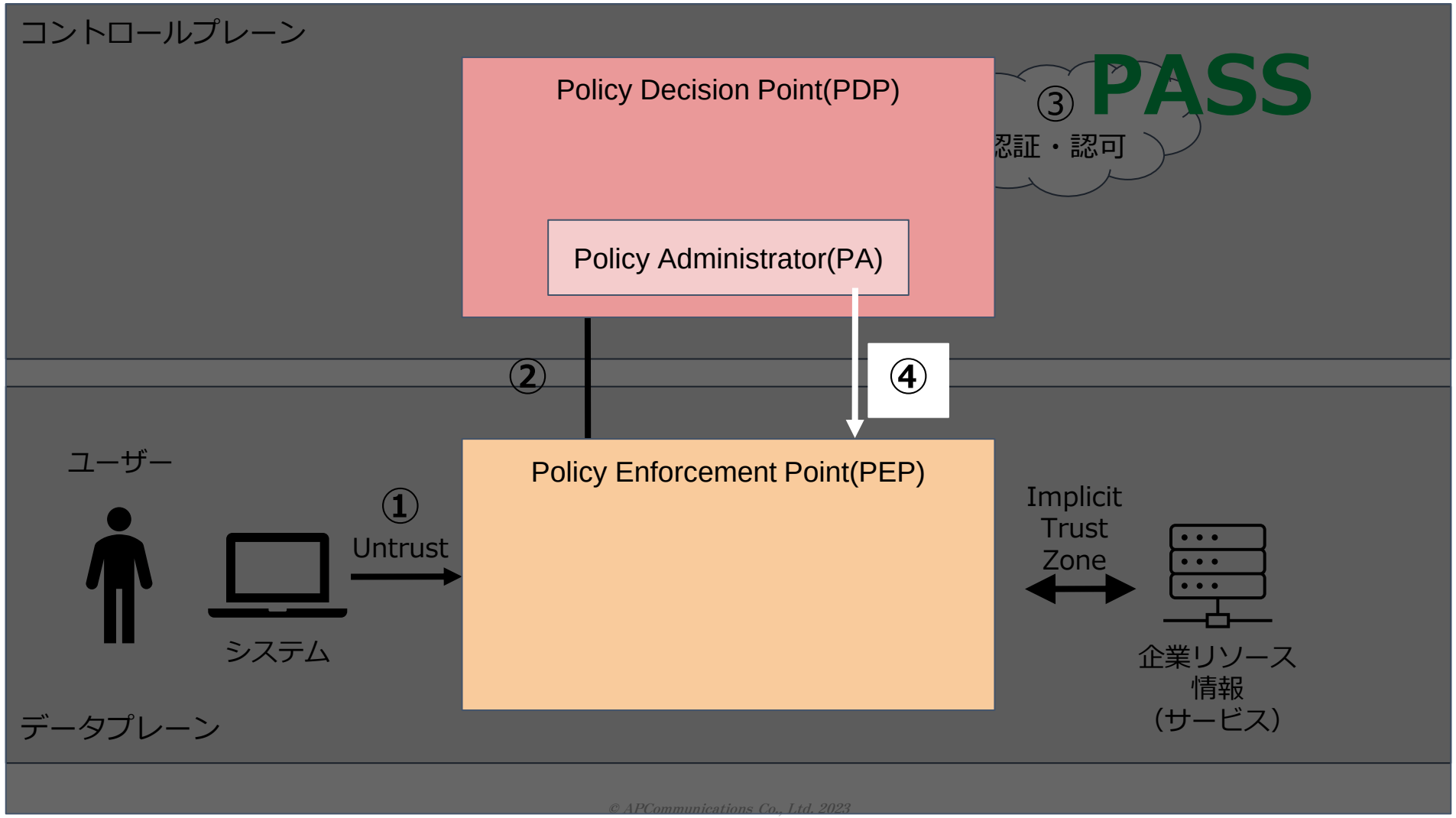
Zero Trust Architecture の論理構成

ユーザーがゼロトラストネットワークで企業リソースにアクセスする流れを以下に示します。



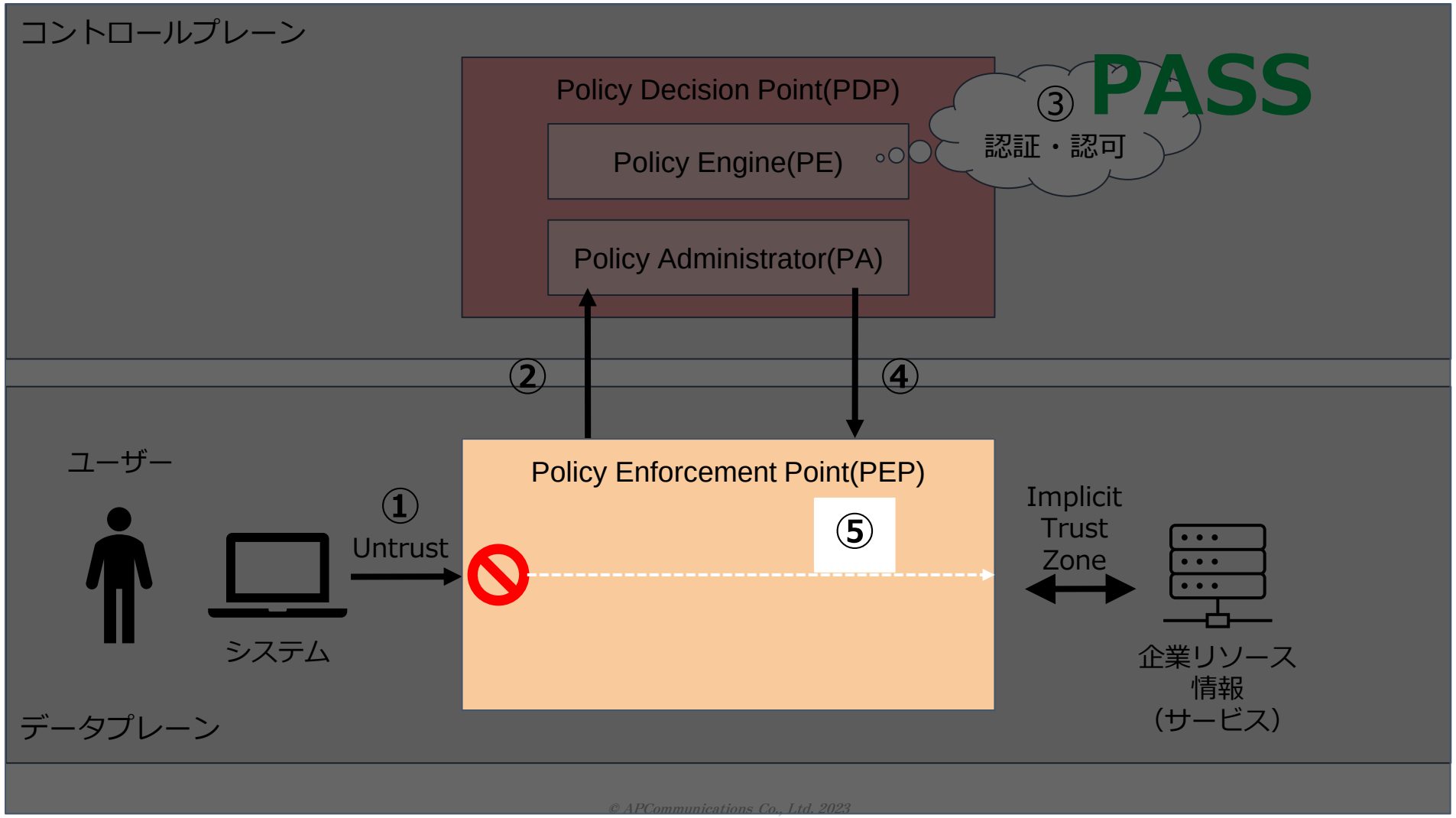
Zero Trust Architecture の論理構成

ユーザーがゼロトラストネットワークで企業リソースにアクセスする流れを以下に示します。



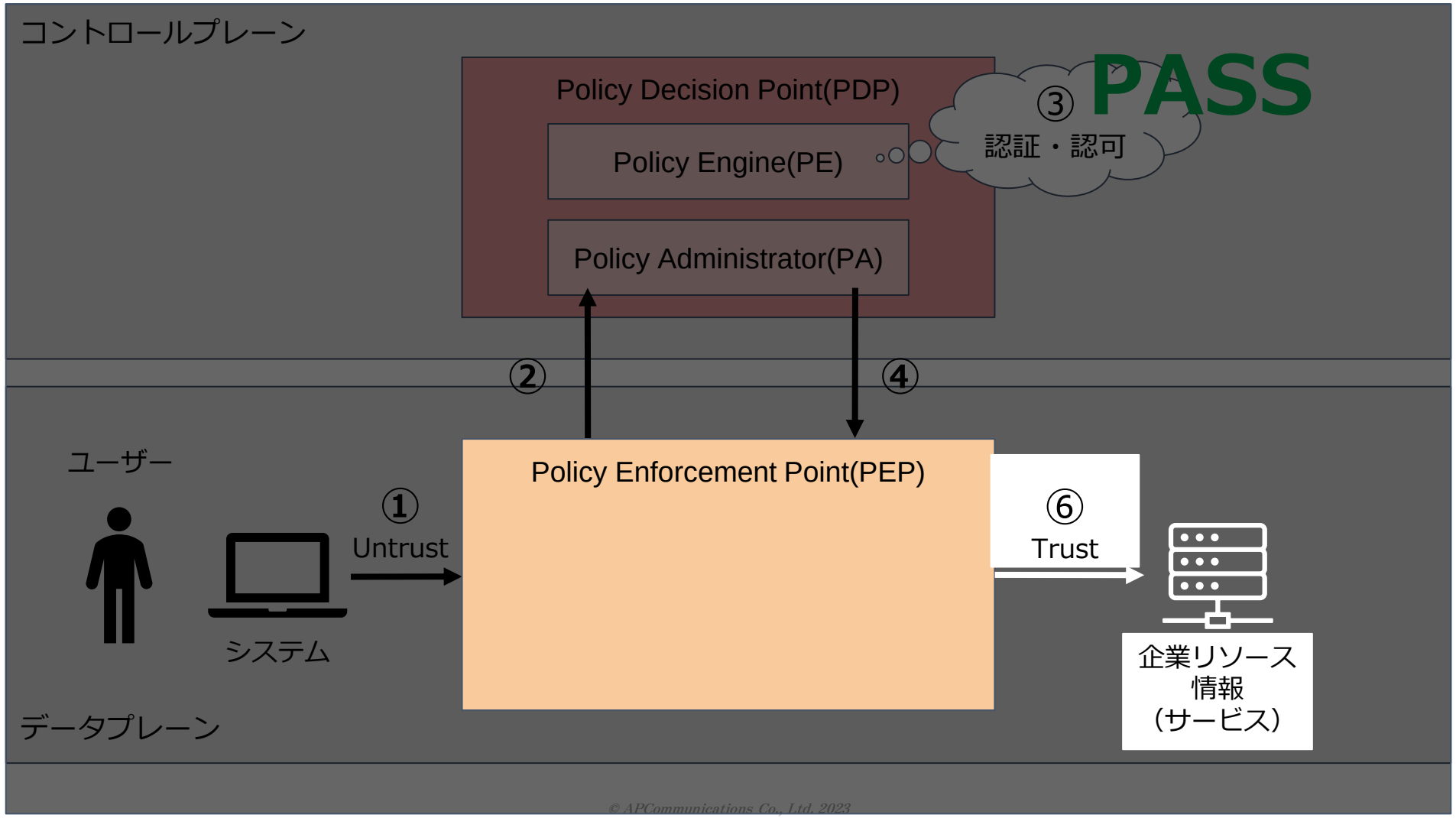
Zero Trust Architecture の論理構成

ユーザーがゼロトラストネットワークで企業リソースにアクセスする流れを以下に示します。



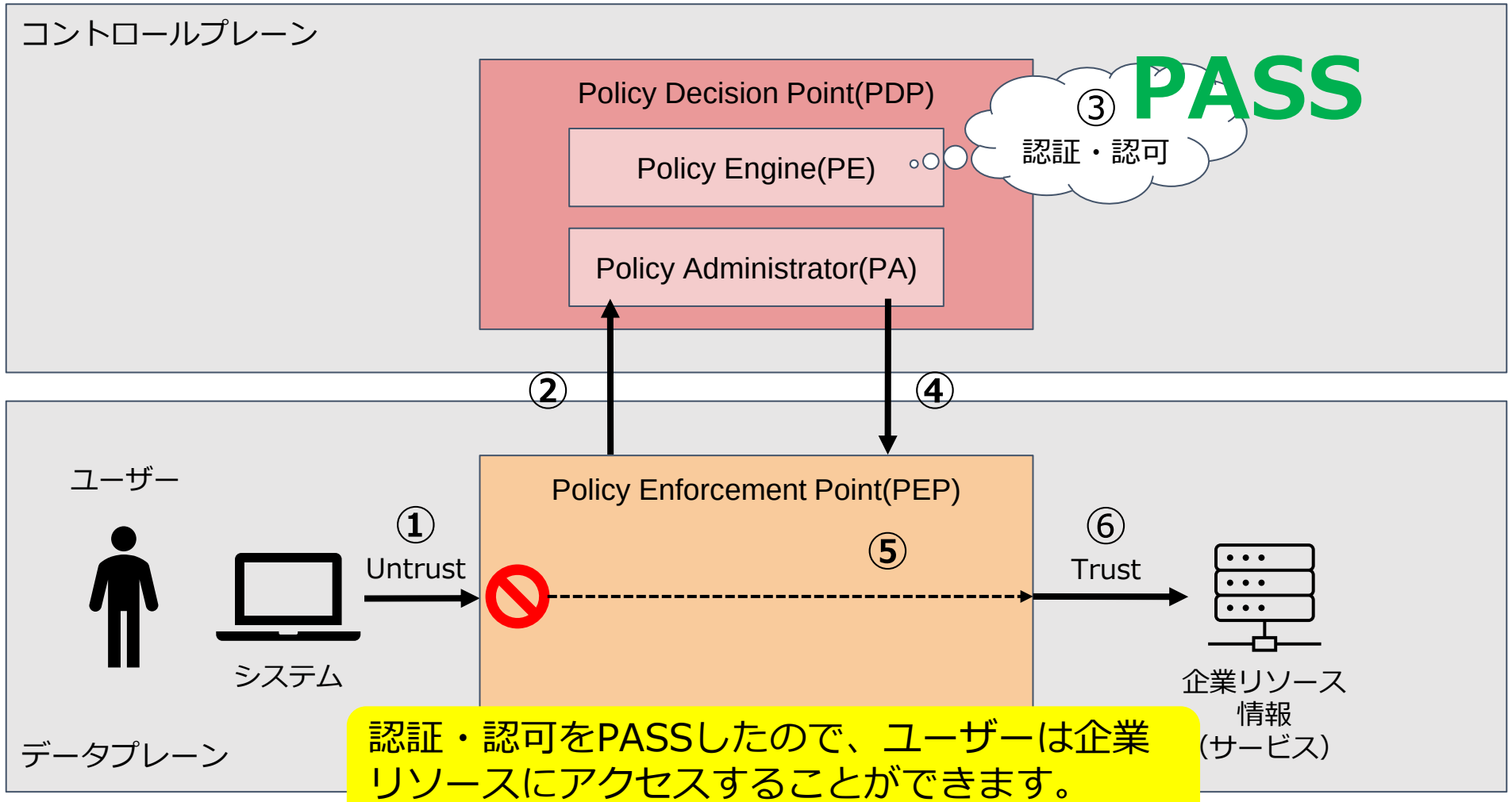
Zero Trust Architecture の論理構成

ユーザーがゼロトラストネットワークで企業リソースにアクセスする流れを以下に示します。



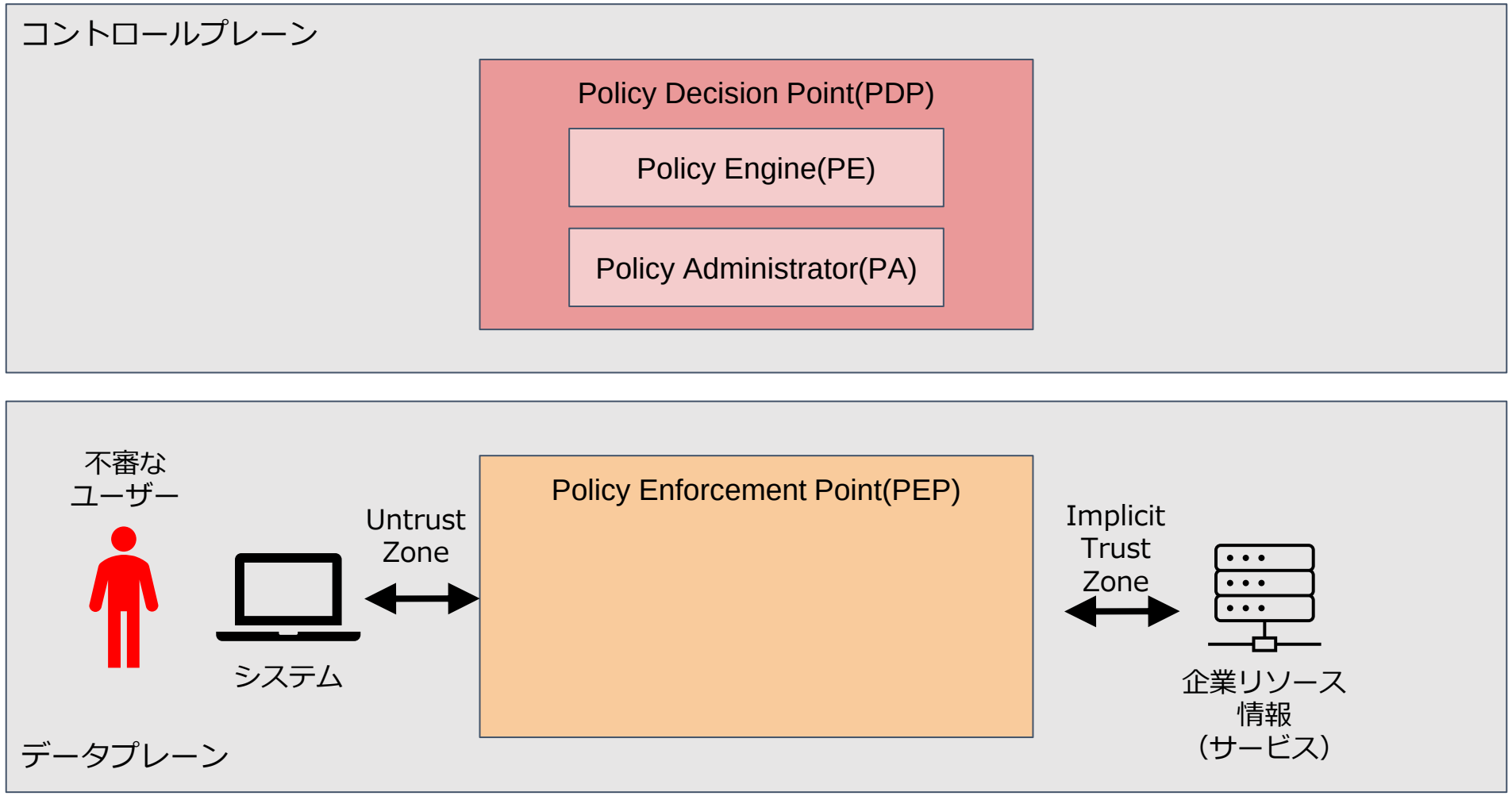
Zero Trust Architecture の論理構成

ユーザーがゼロトラストネットワークで企業リソースにアクセスする流れを以下に示します。



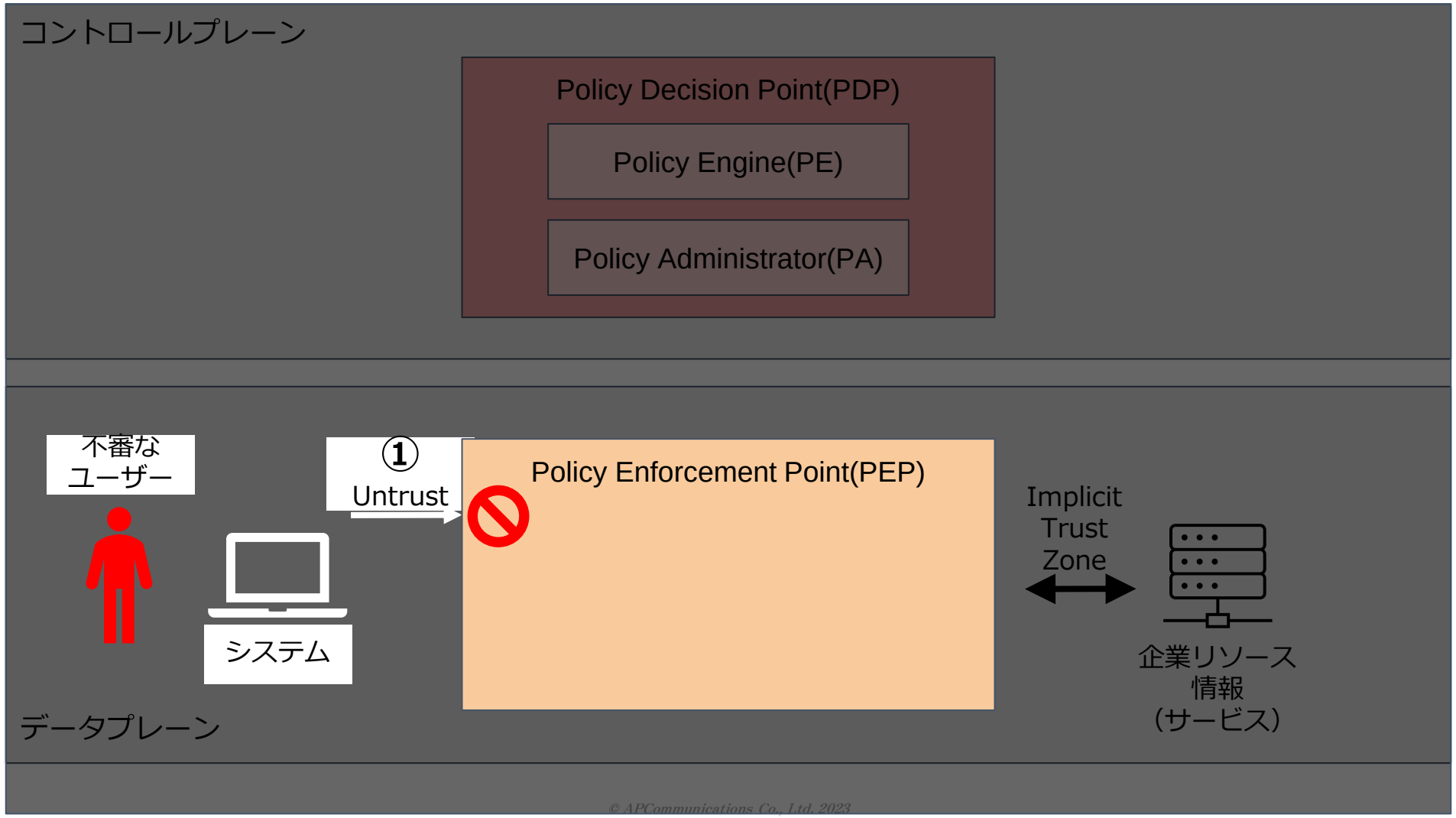
Zero Trust Architecture の論理構成

不審なユーザーが企業リソースにアクセスする場合はどうなるでしょうか。



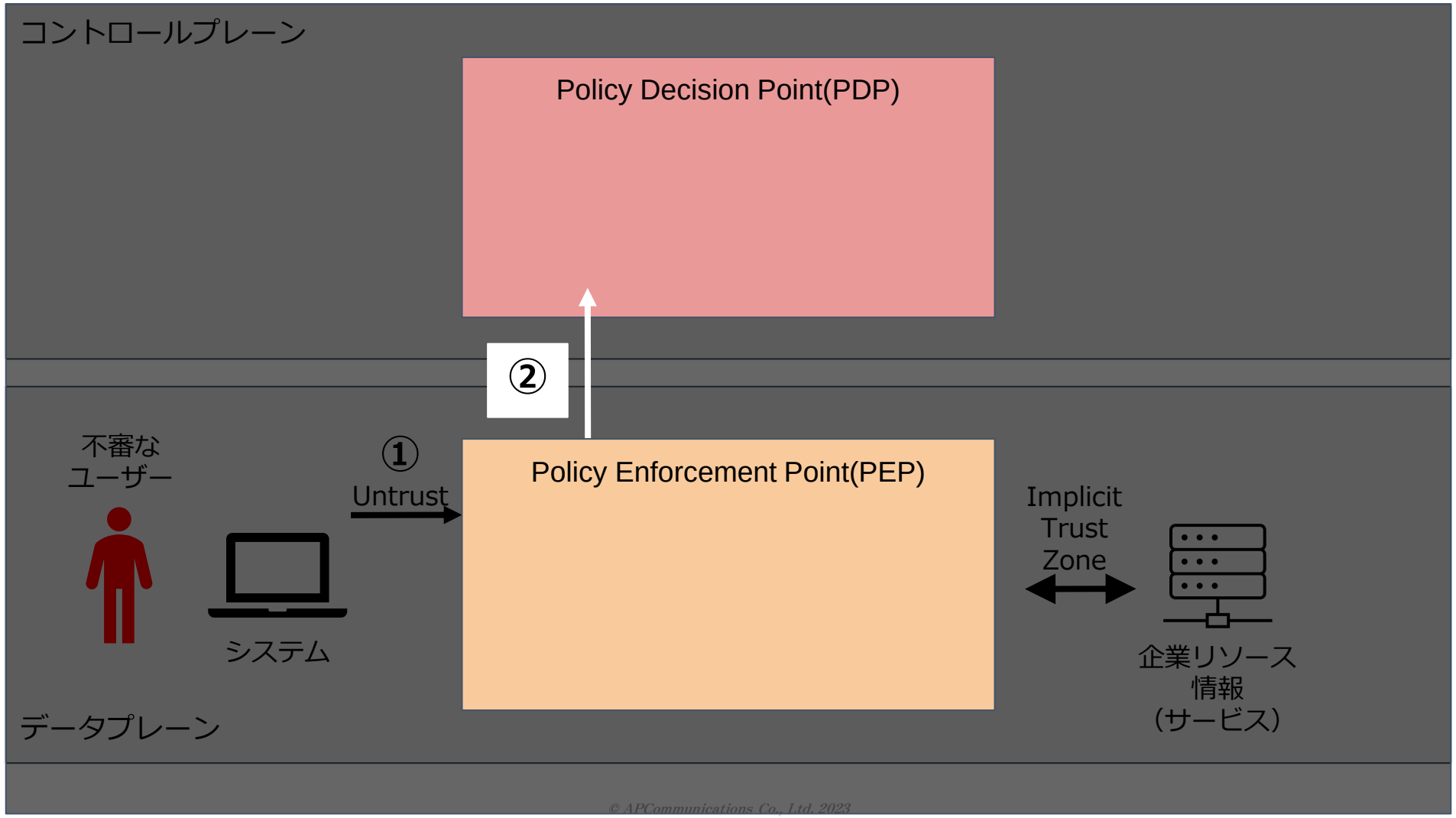
Zero Trust Architecture の論理構成

不審なユーザーが企業リソースにアクセスする場合はどうなるでしょうか。



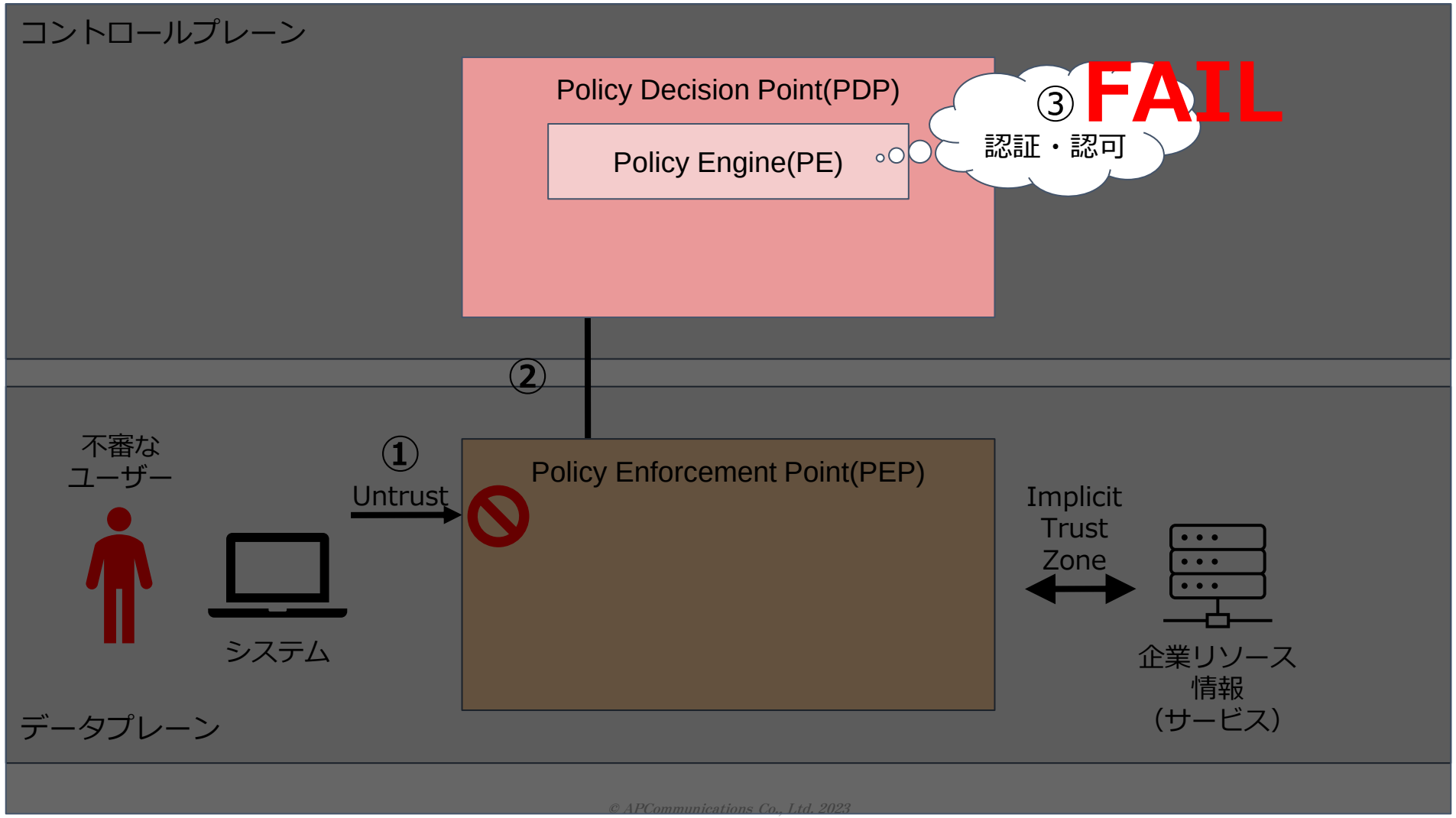
Zero Trust Architecture の論理構成

不審なユーザーが企業リソースにアクセスする場合はどうなるでしょうか。



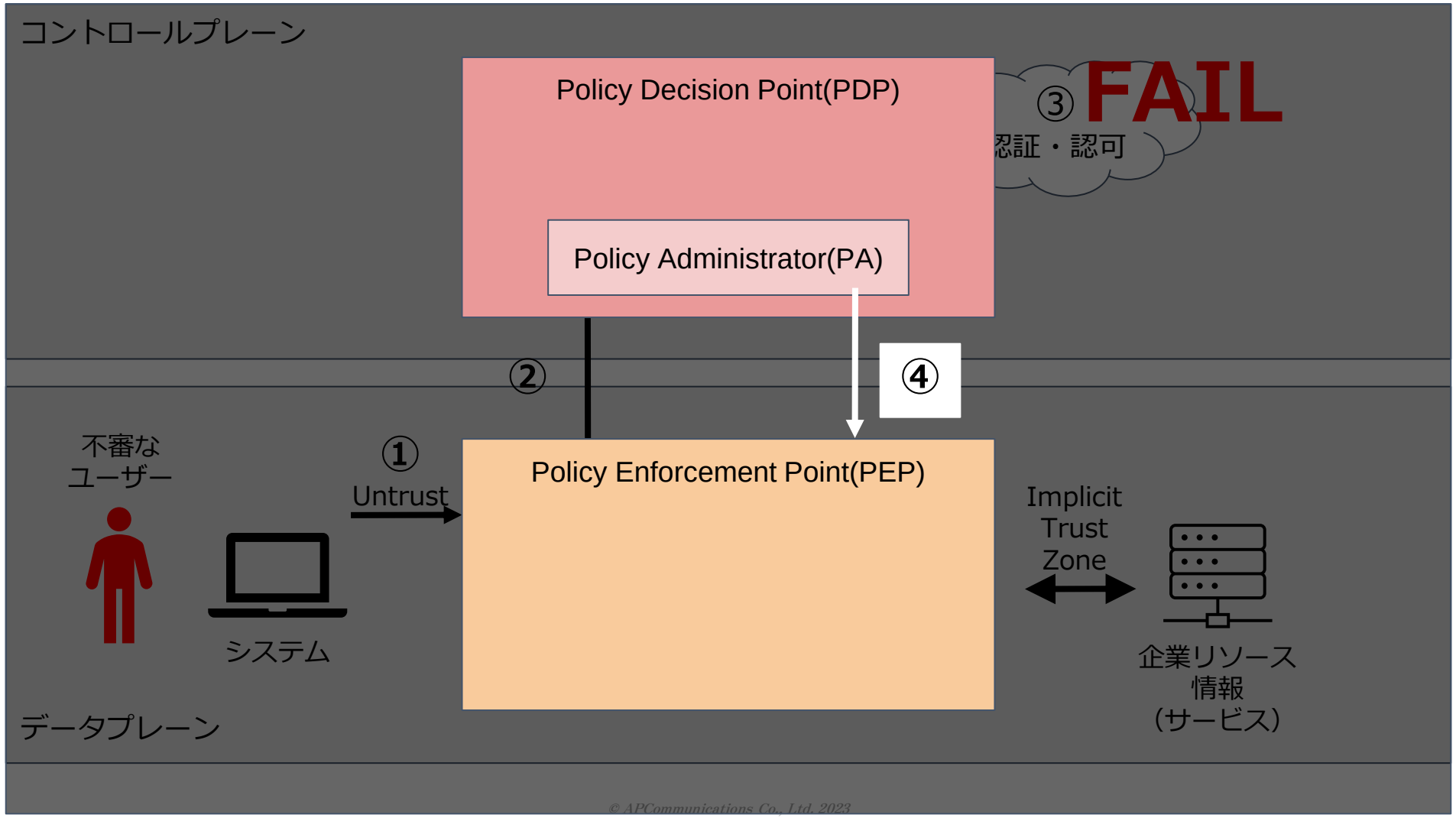
Zero Trust Architecture の論理構成

不審なユーザーが企業リソースにアクセスする場合はどうなるでしょうか。



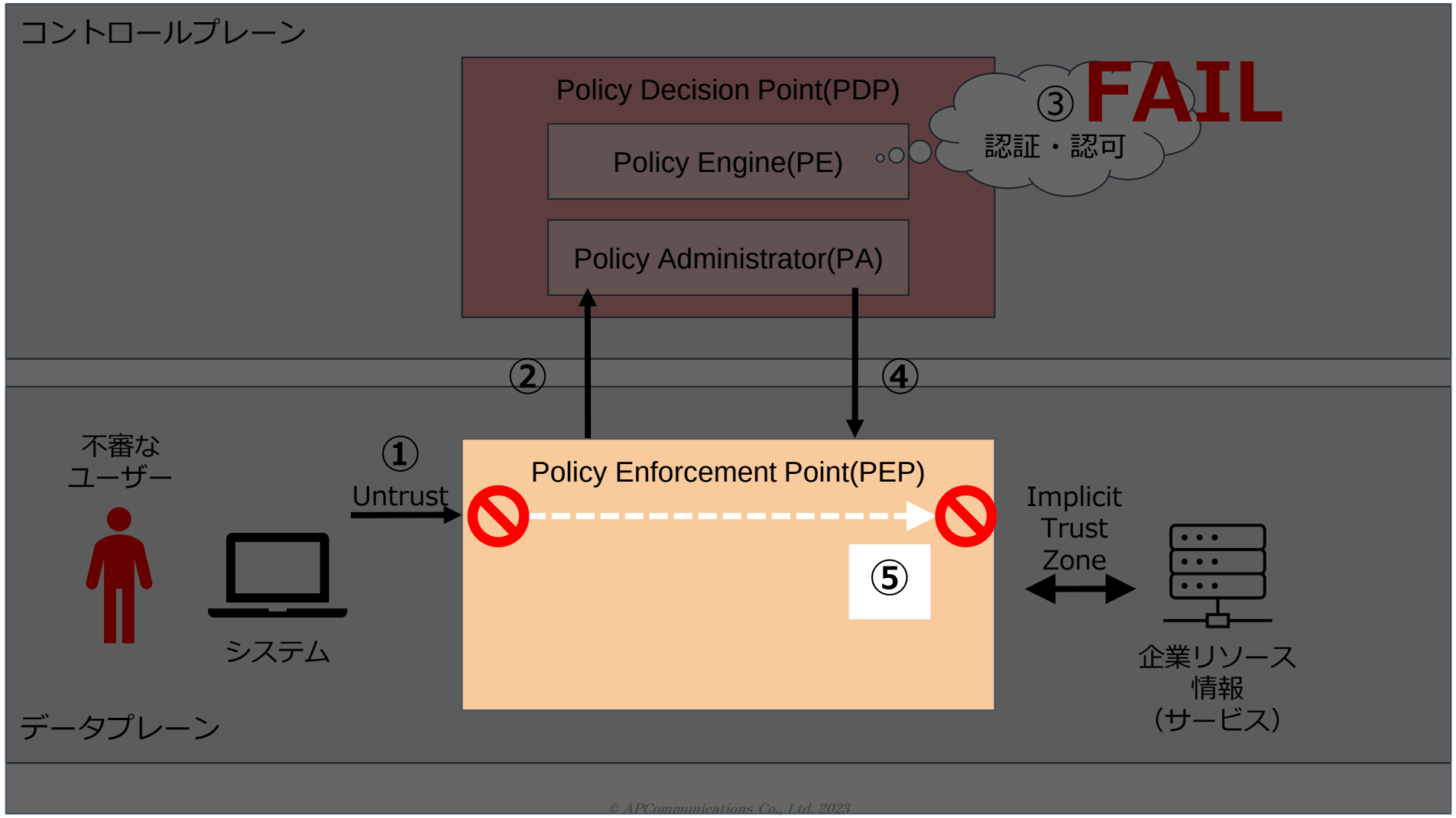
Zero Trust Architecture の論理構成

不審なユーザーが企業リソースにアクセスする場合はどうなるでしょうか。



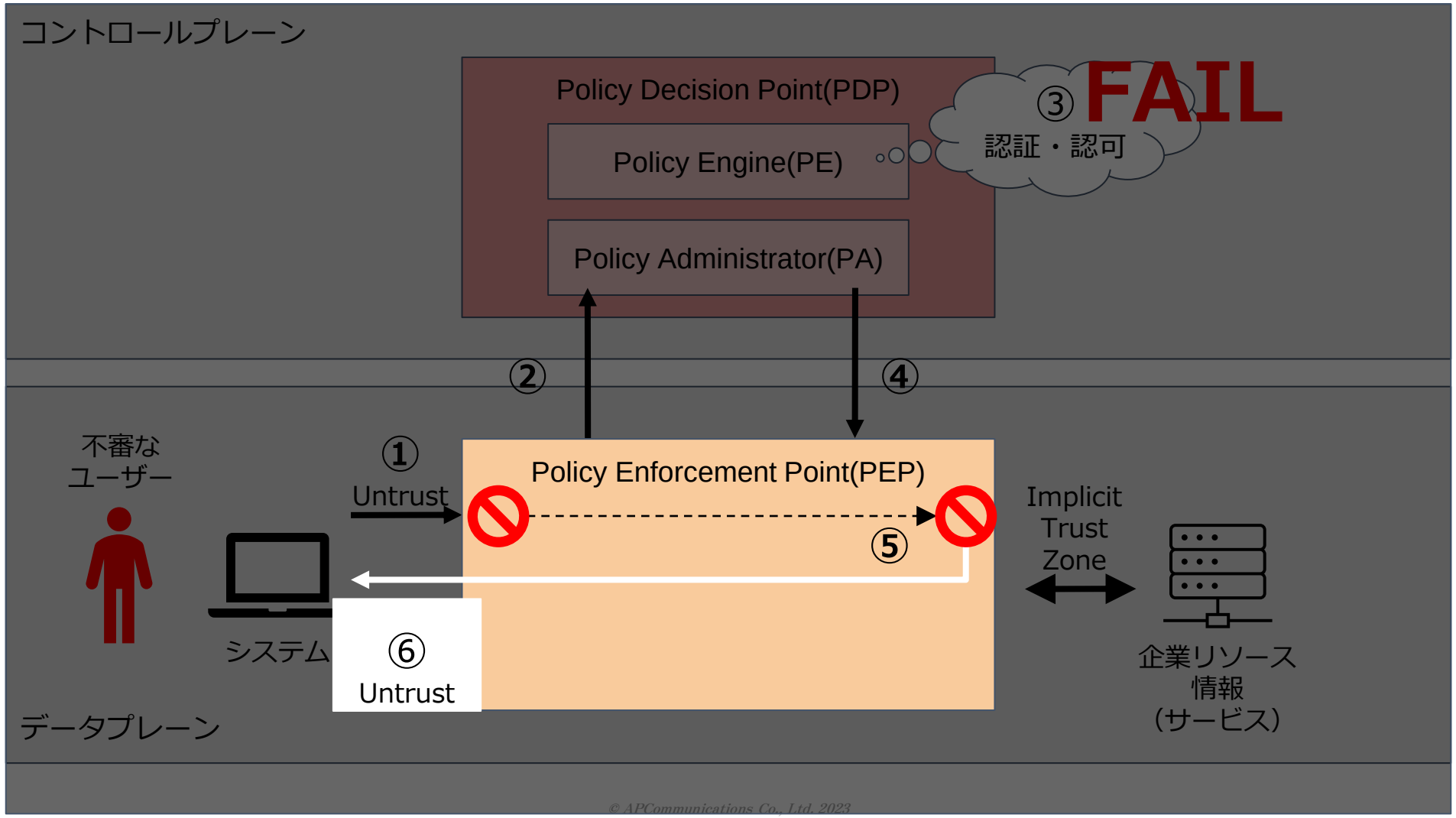
Zero Trust Architecture の論理構成

不審なユーザーが企業リソースにアクセスする場合はどうなるでしょうか。



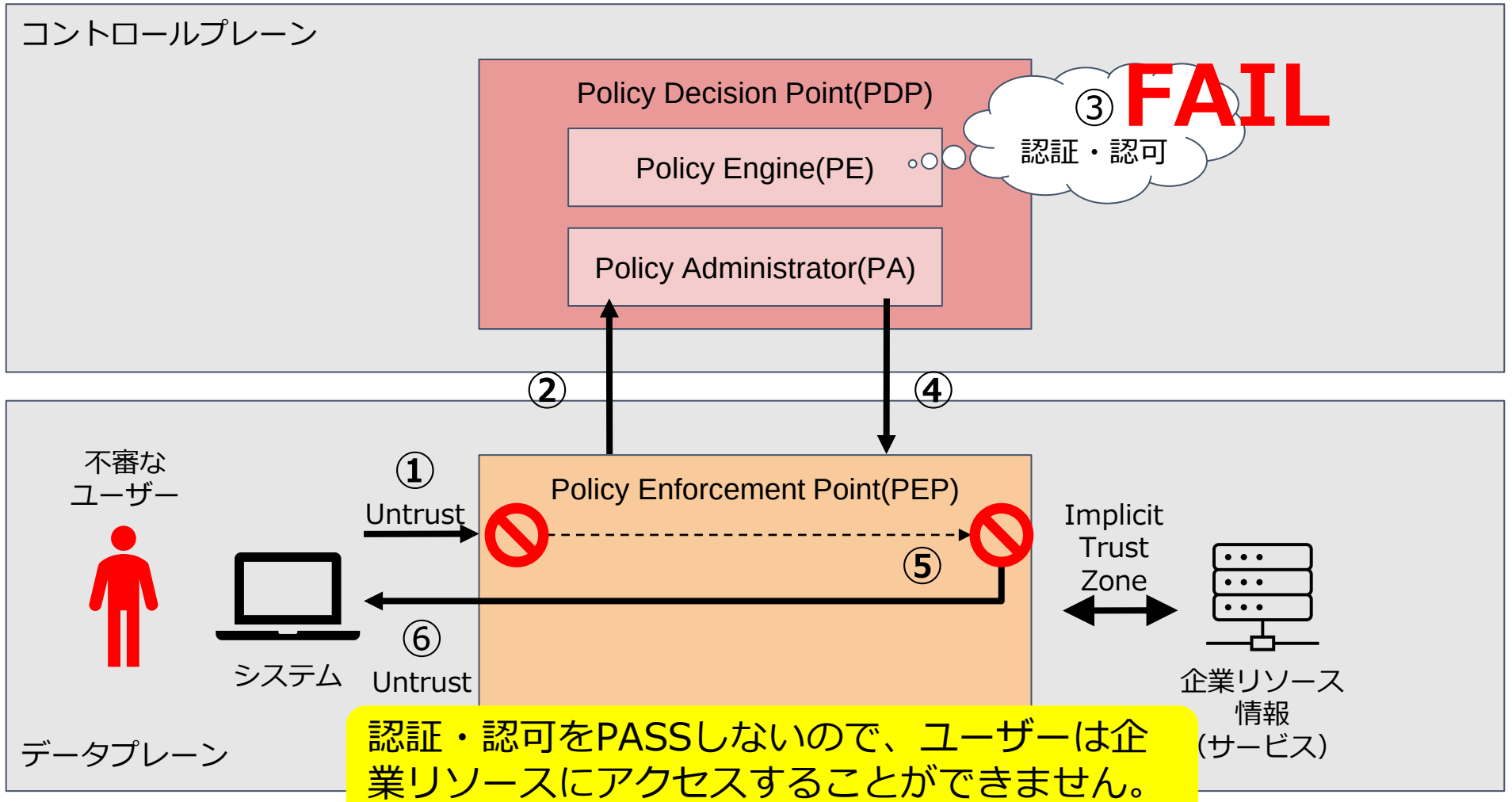
Zero Trust Architecture の論理構成

不審なユーザーが企業リソースにアクセスする場合はどうなるでしょうか。



Zero Trust Architecture の論理構成

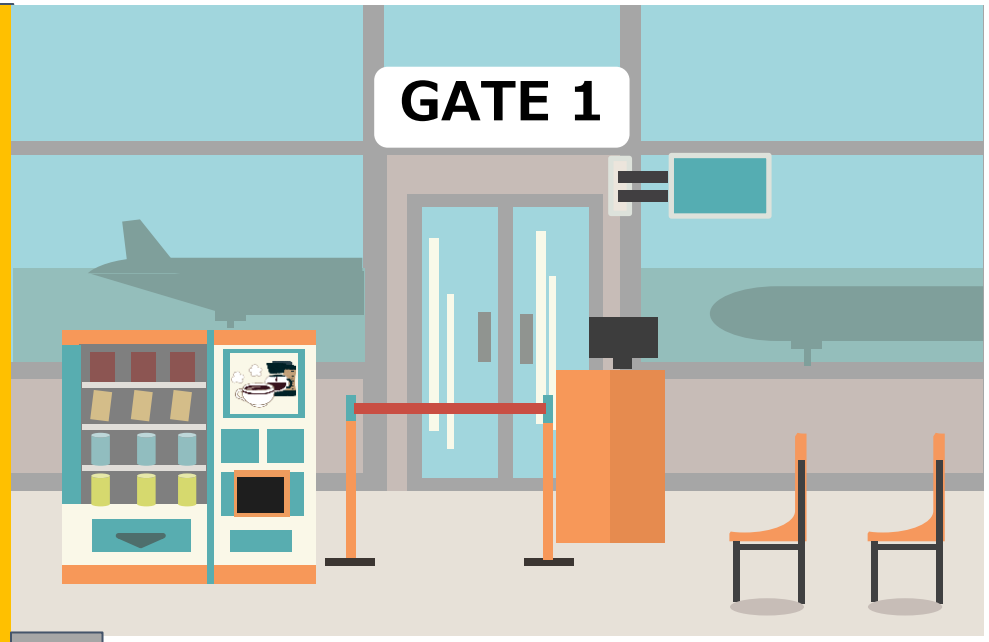
不審なユーザーが企業リソースにアクセスする場合はどうなるでしょうか。



暗黙のトラスト領域を最小化せよ！

空港における乗客の検閲モデルについて考えてみましょう。

PDP/PEP



ユーザー

Untrust Zone

GATE

Implicit Trust Zone
（搭乗エリア）

企業リソース
情報
（サービス）

ここの領域は可能な限り
最小化すべきです。

1. はじめに
2. Zero Trust Network Access
3. Zero Trust Network Accessの歴史
4. Reverse PROXY
5. Port knocking
6. Single Packet Authorization
7. VPNとZTNA/SDPの比較
8. Zero Trust Architecture の論理構成
9. SDPの仕組み
10. ZTNAの仕組み
11. おわりに

SDPの仕様書

Software-Defined Perimeter(SDP)仕様書v2.0とは何か？

- はじめに
- SDPのデザイン
- SDPプロトコル
- まとめ
- 参考文献
- 付録

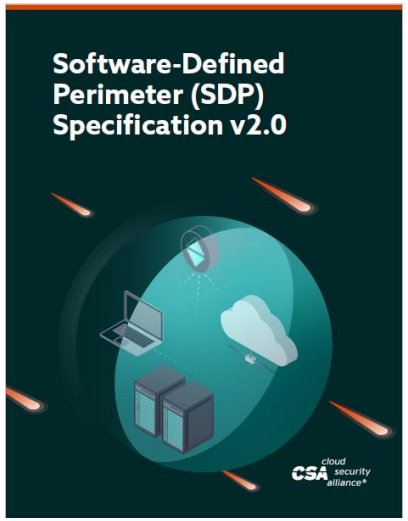
約８年、ゼロトラストの原則に対する熱意は凄いです！

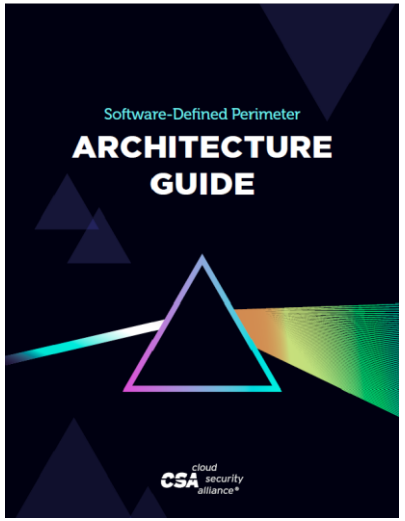
- SDPが生まれた背景
- SDPの基本的な考え方
- SDPプロトコル詳細

Software-Defined Perimeter(SDP) とは何かを把握する最良のドキュメント

SDPの仕様書と重要ドキュメント

Software-Defined Perimeter(SDP)仕様書v2.0とアーキテクチャガイド





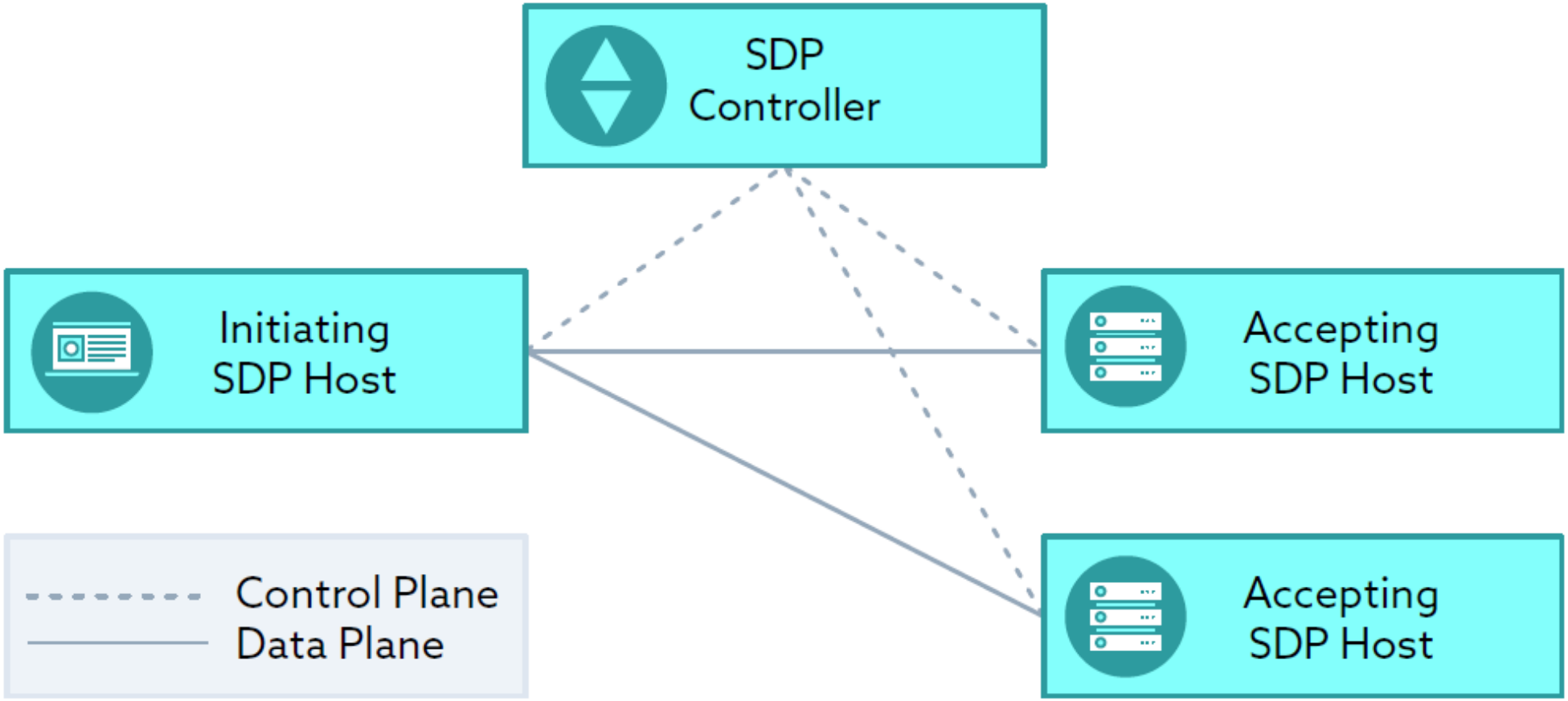
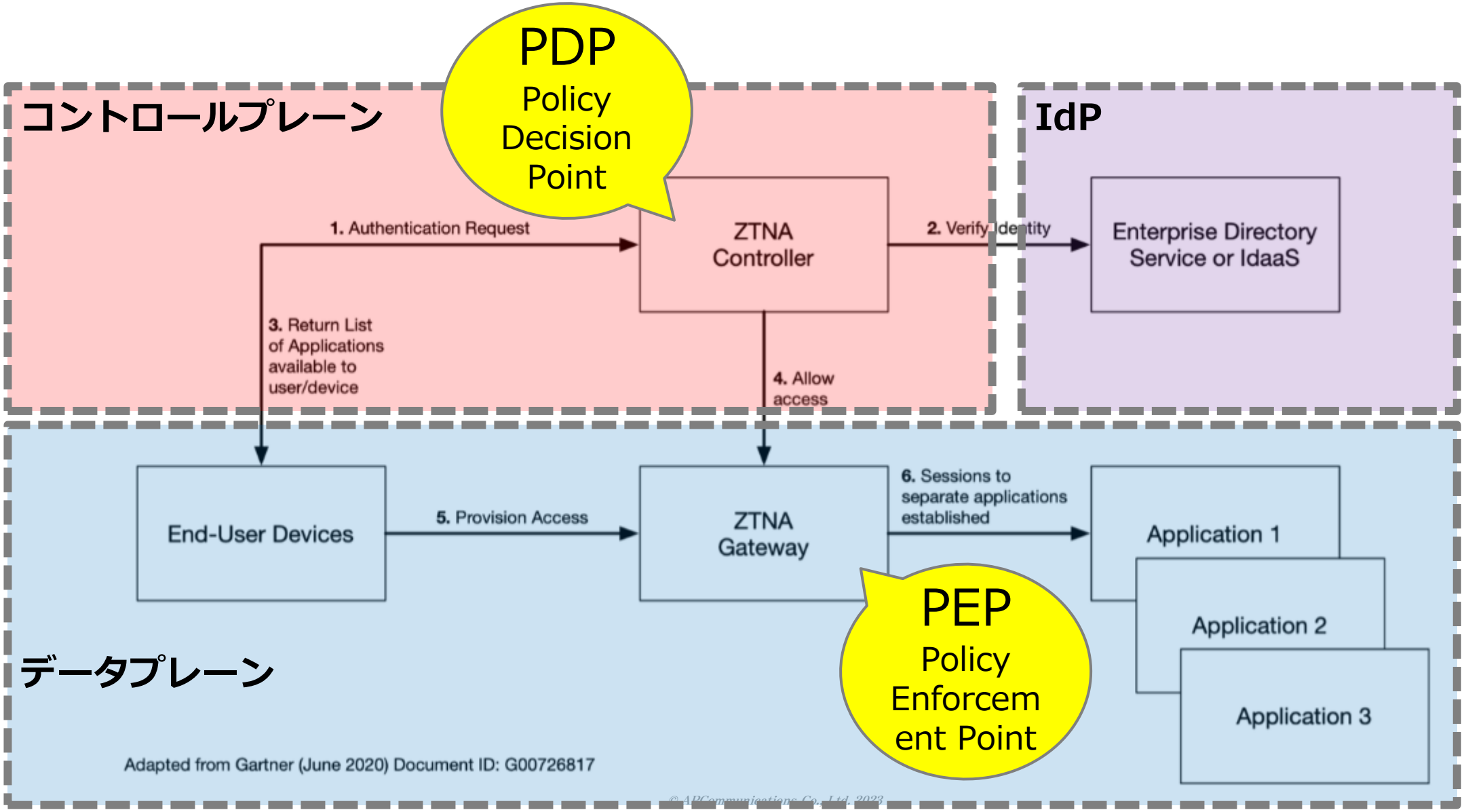


Figure 1: SDP Architecture (previously published by CSA in Software Defined Perimeter and Zero Trust)[5]

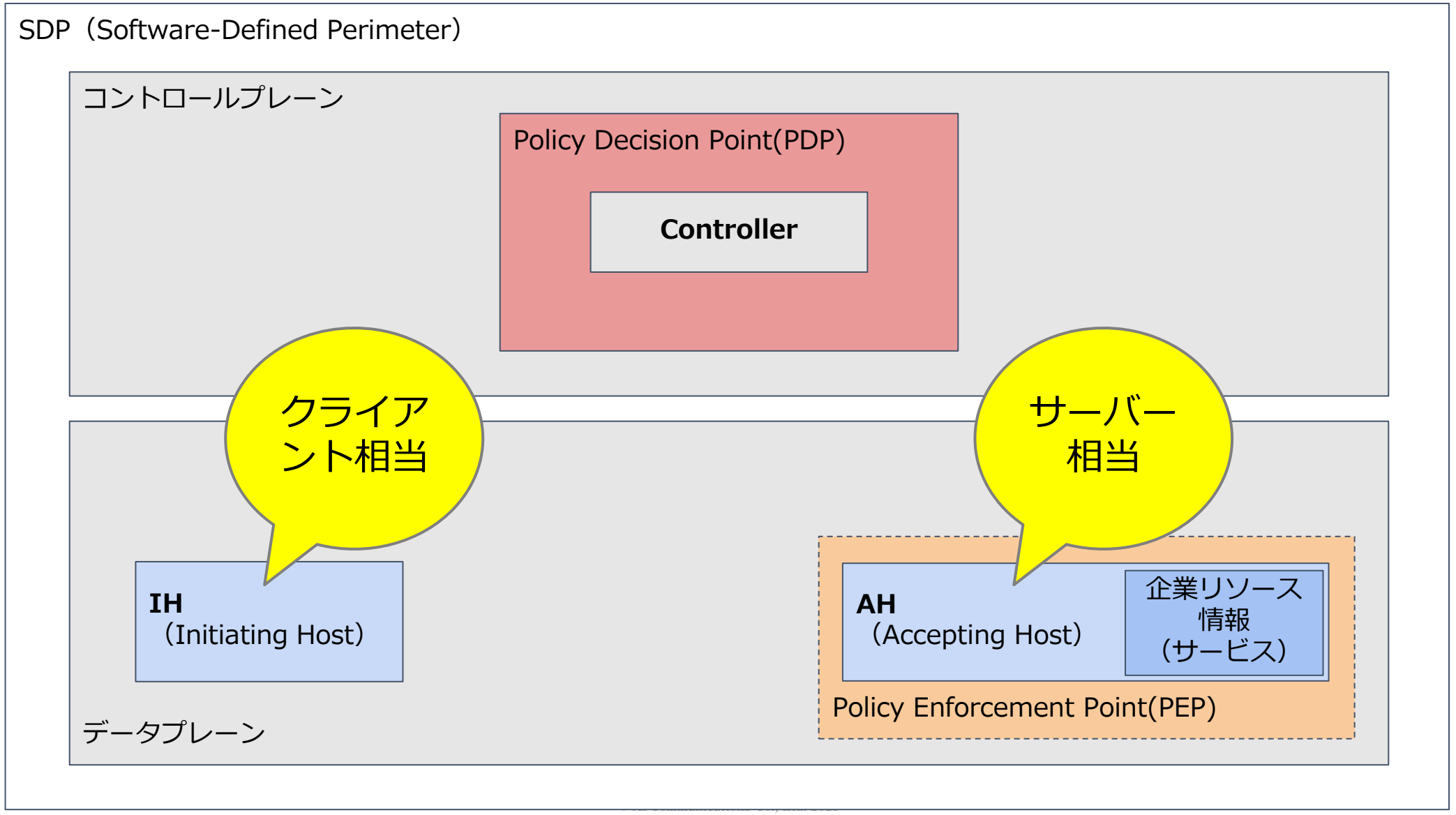
SDPの論理構成

SDPもコントロールプレーンとデータプレーンに分かれています。



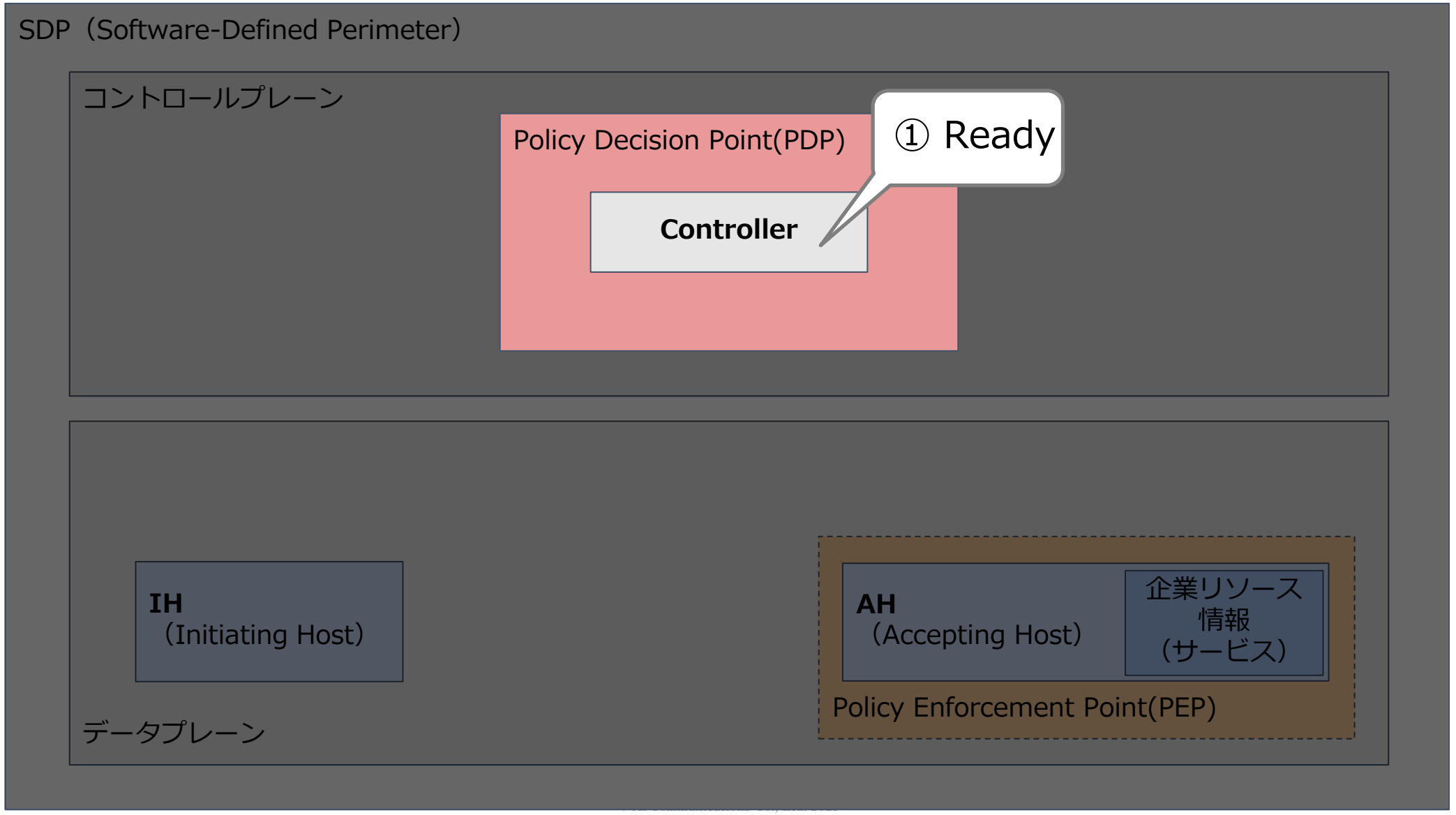
SDPの仕組み

Software-Defined Perimeter(SDP)仕様書v2.0を基にZTAの簡易モデルを書き換えます。



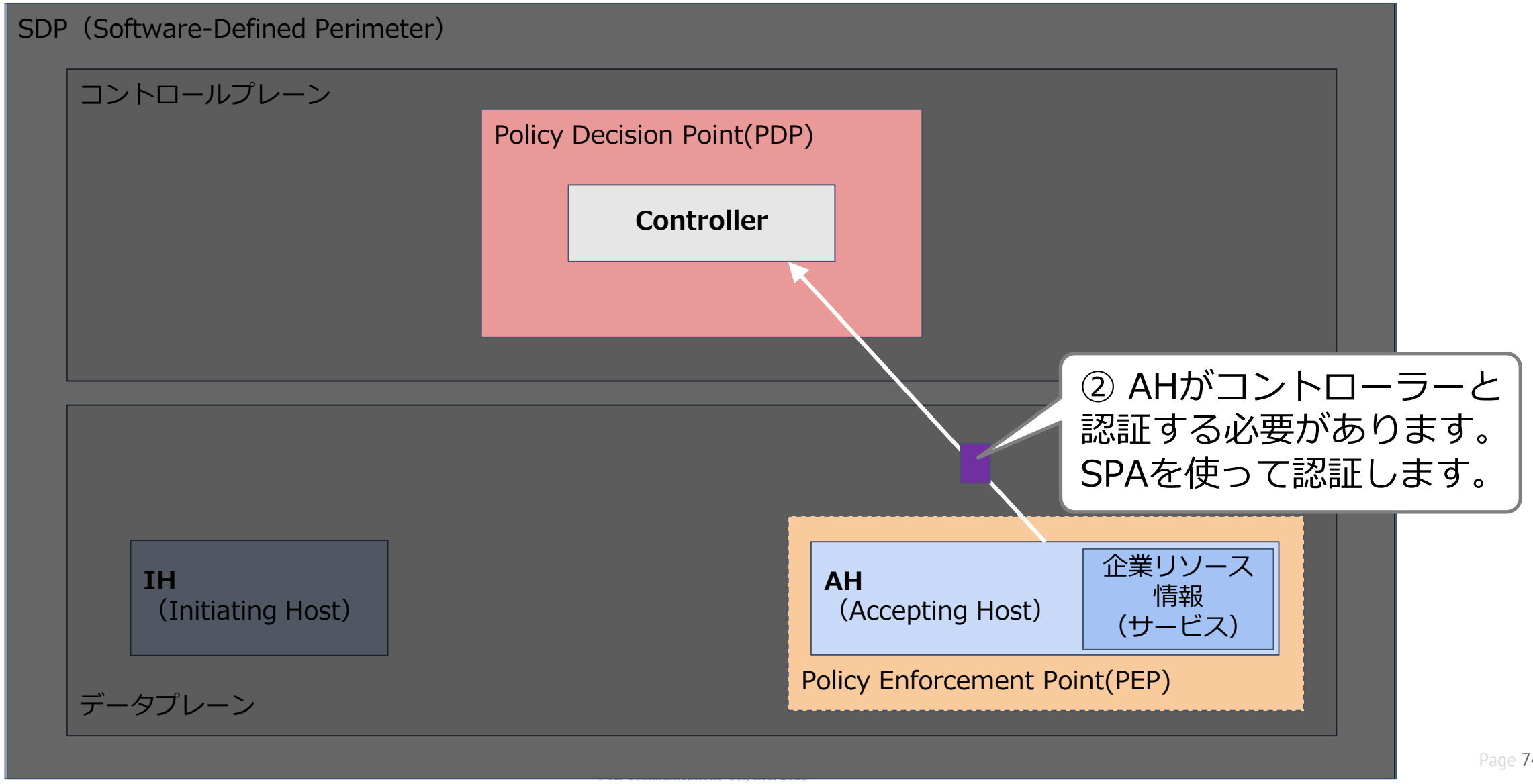
SDPの仕組み

① 大前提として、SDPコントローラーがオンライン（使用可能状態）である必要があります。



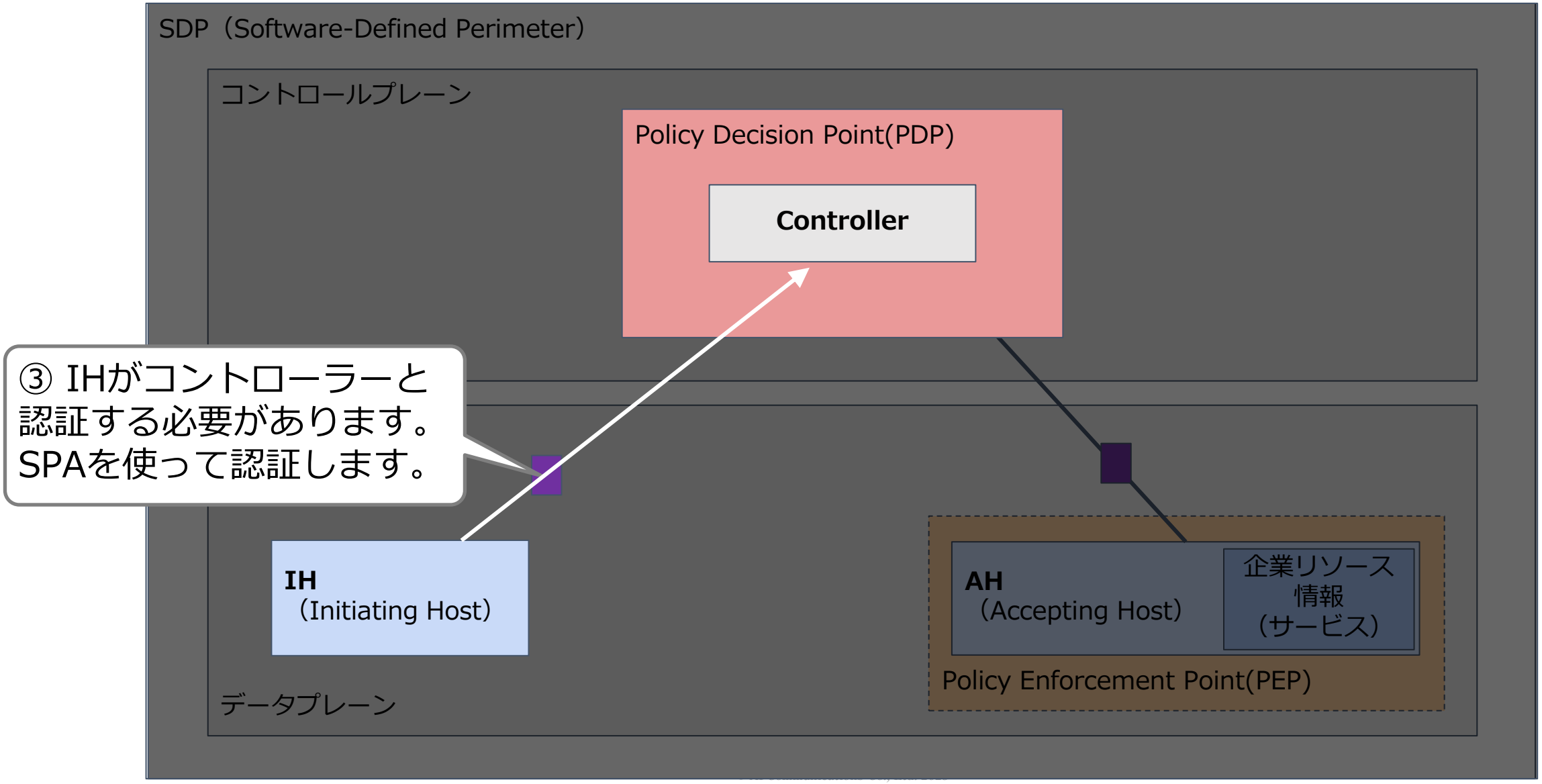
SDPの仕組み

② AHがコントローラーと認証する必要があります。



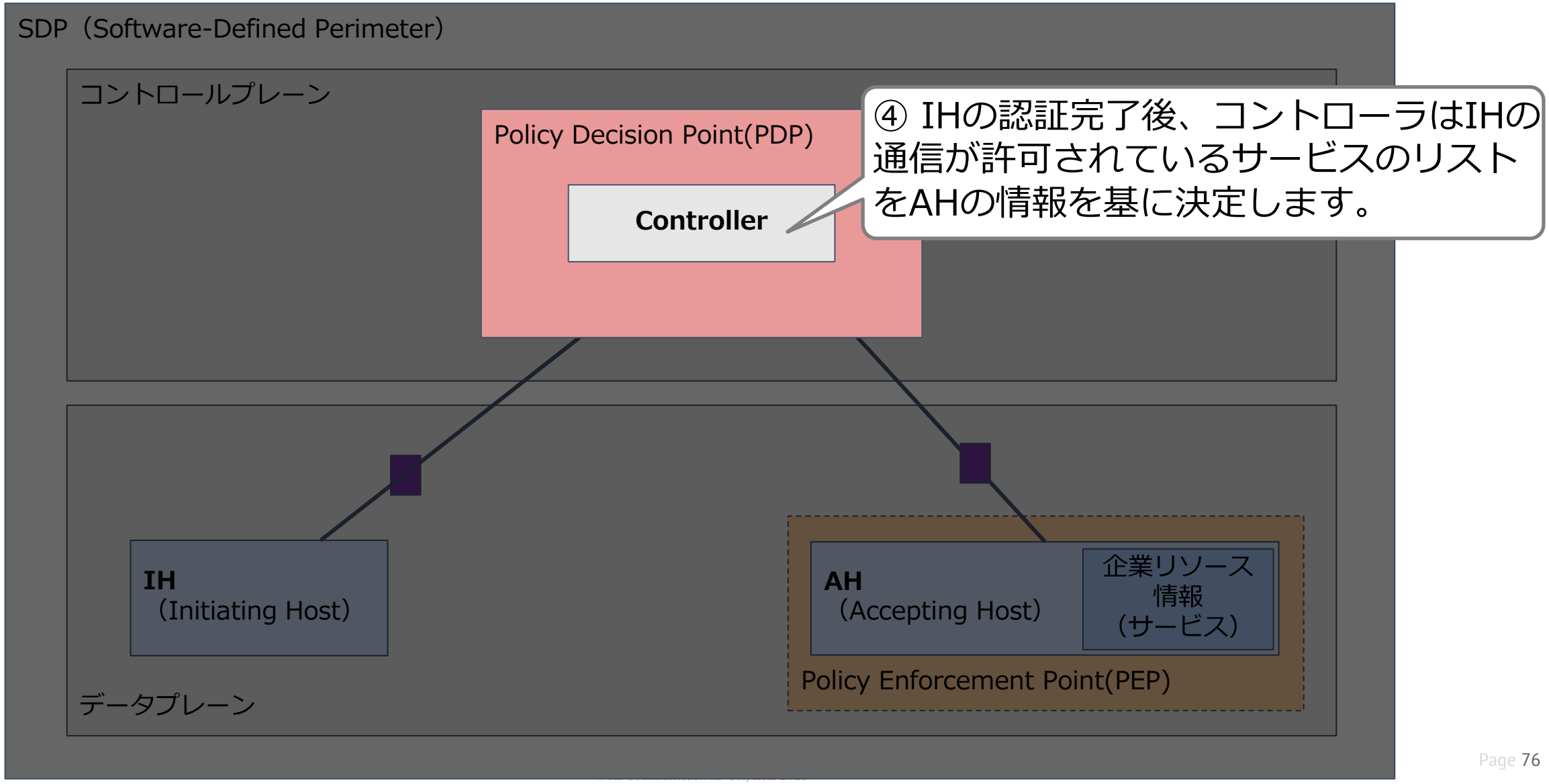
SDPの仕組み

③ IHがコントローラーと接続して認証を行います。



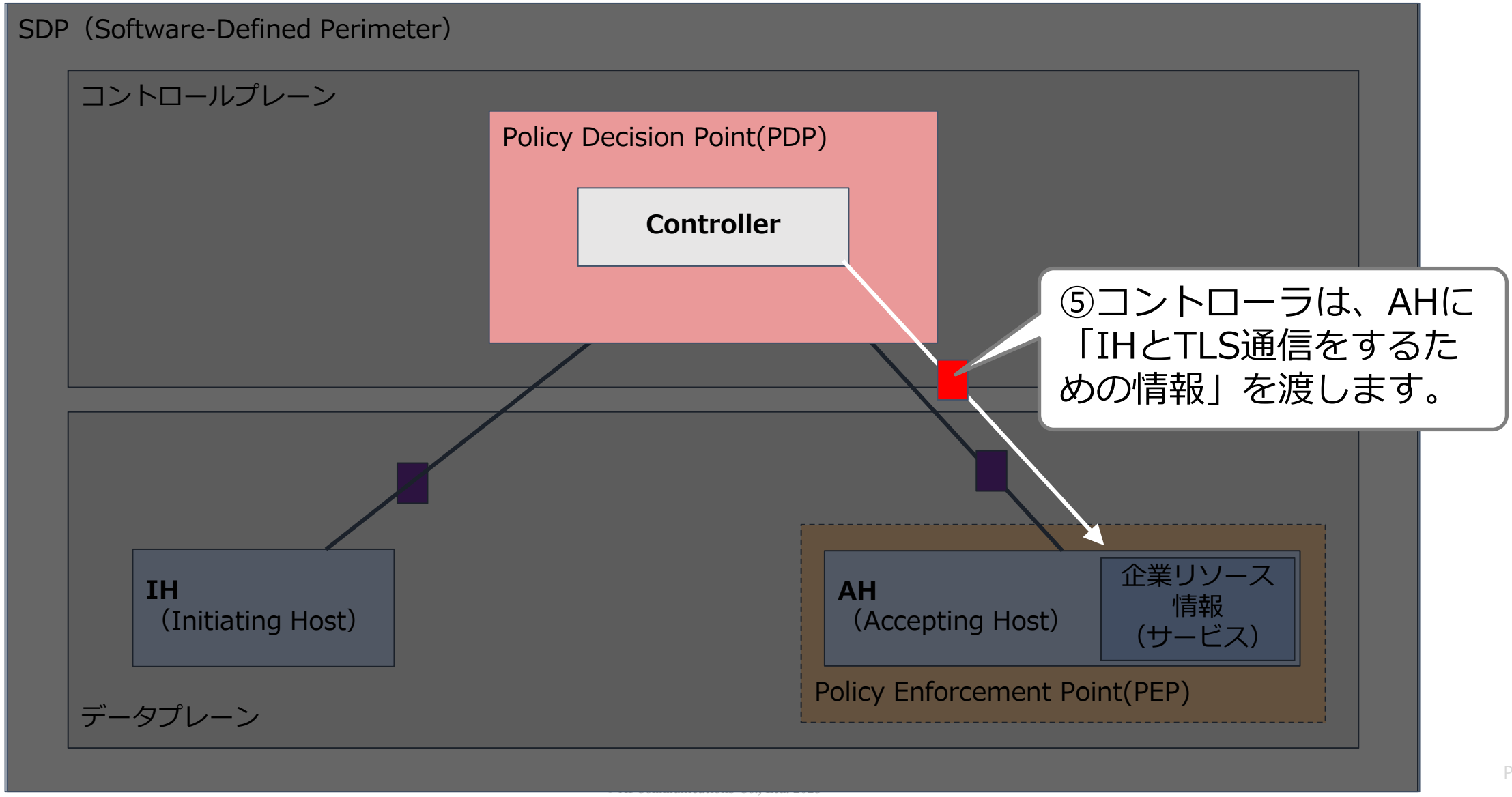
SDPの仕組み

④ IHの認証完了後、コントローラはIHの通信が許可されているサービスのリストをAHからの情報を基に決定します。



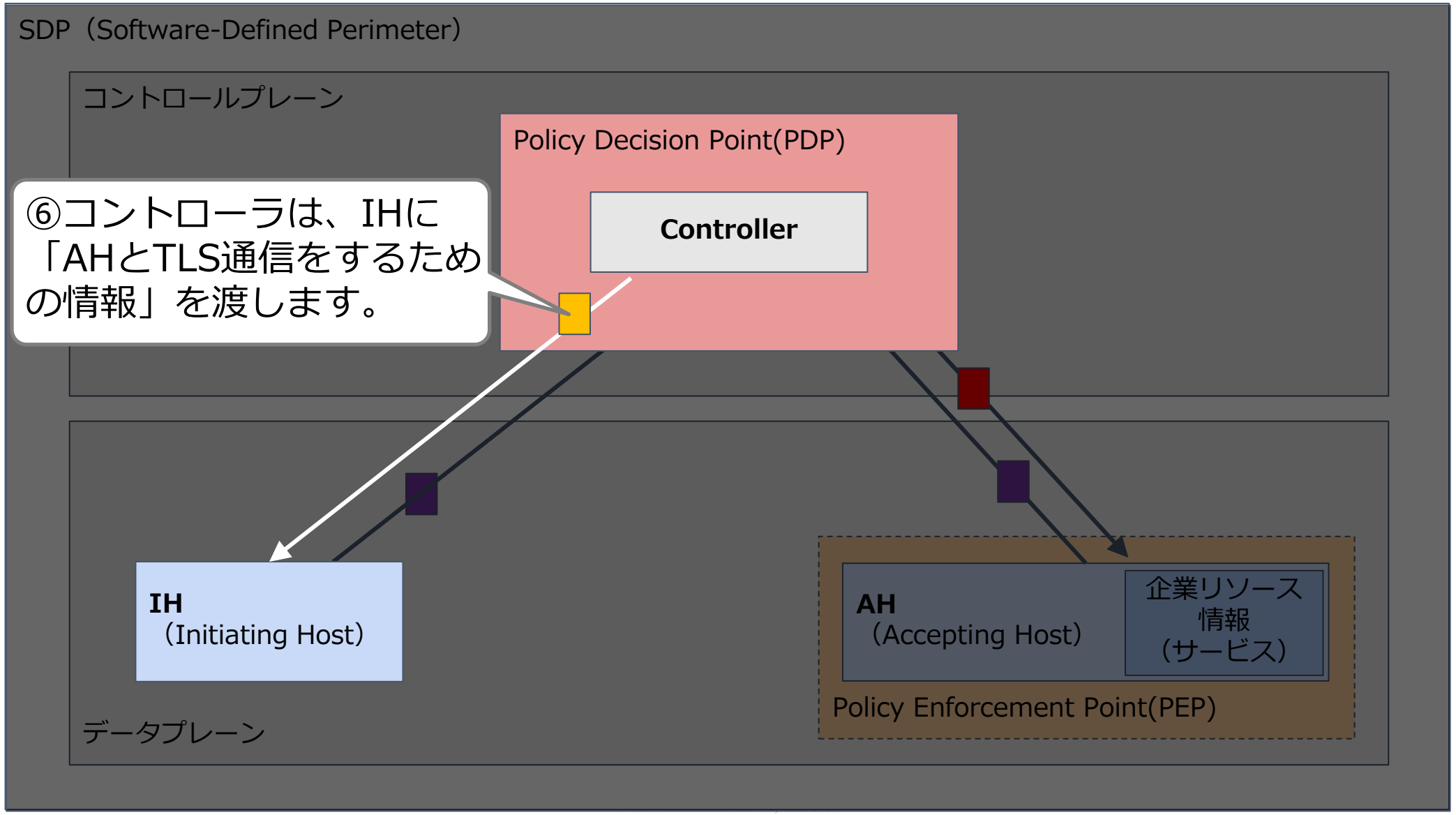
SDPの仕組み

⑤ コントローラは、AHに「IHとTLS通信をするための情報」を渡します。
AHは、IHからのSPAパケットの受信待ち状態となります。



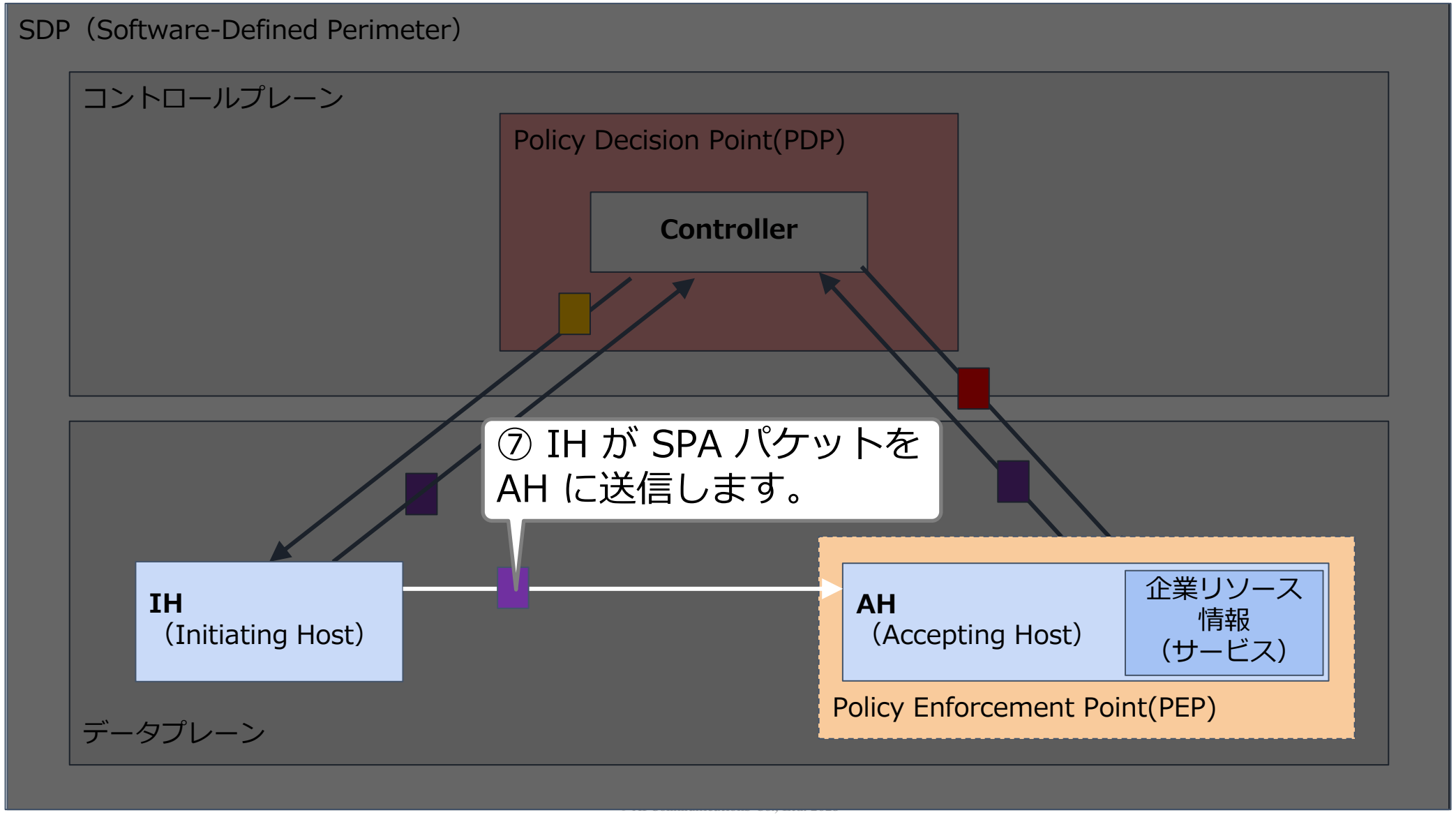
SDPの仕組み

⑥ コントローラは、IHに「AHとTLS通信をするための情報」を渡します。



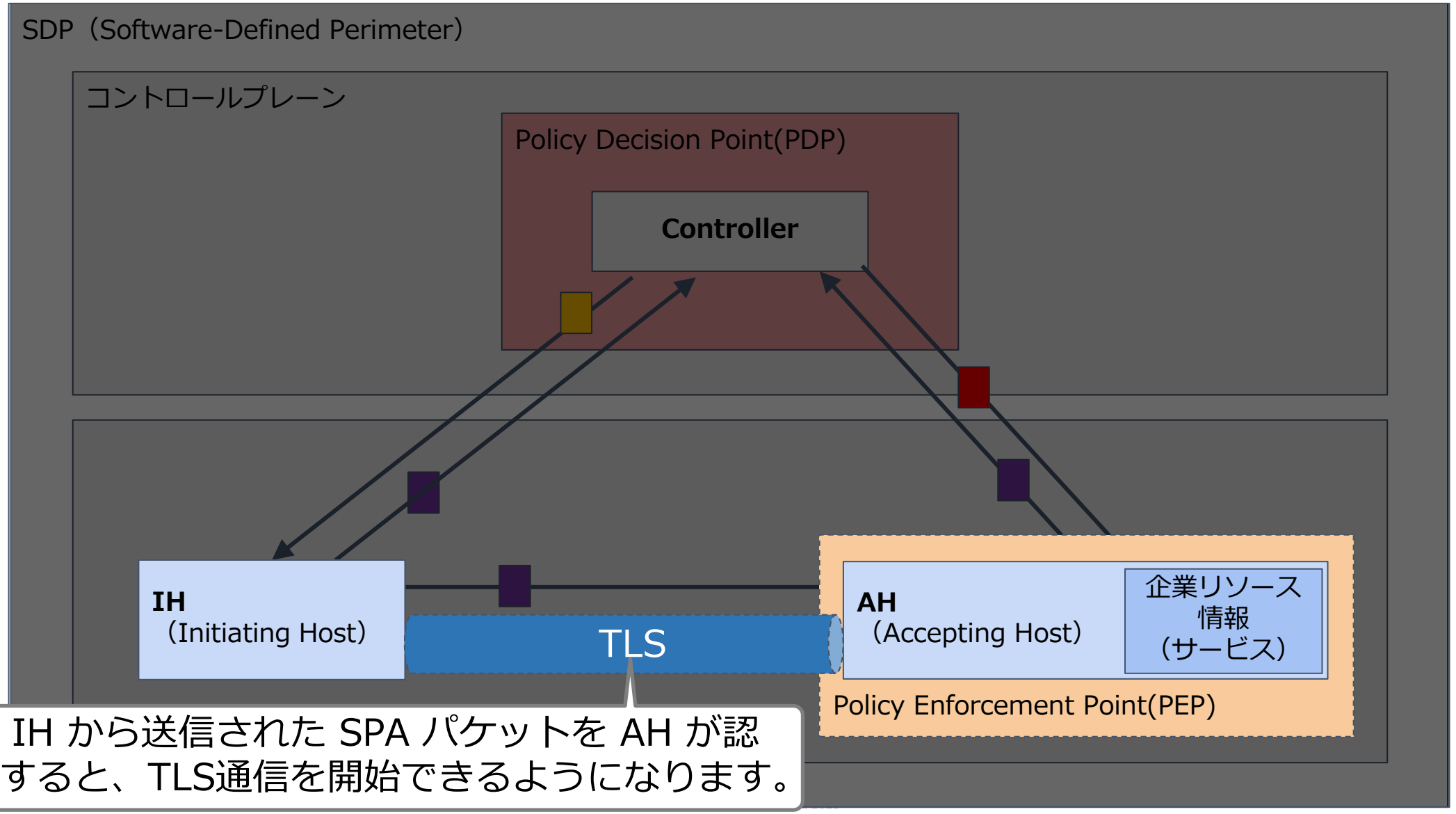
SDPの仕組み

⑦ IH が SPA パケットを AH に送信します。



SDPの仕組み

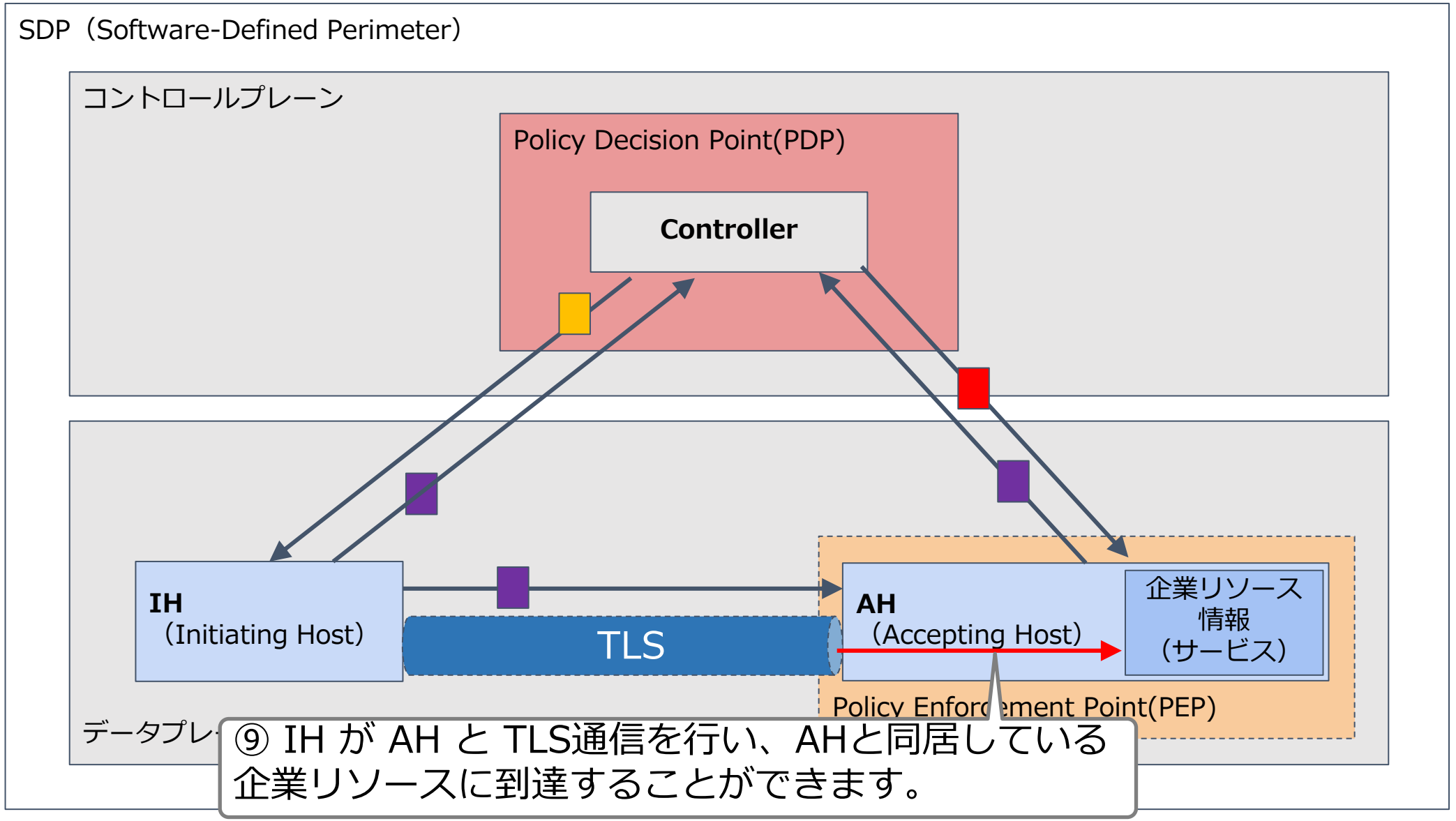
⑧ IH から送信された SPA パケットを AH が認証すると、TLS通信を開始できるようになります。



⑧ IH から送信された SPA パケットを AH が認証すると、TLS通信を開始できるようになります。

SDPの仕組み

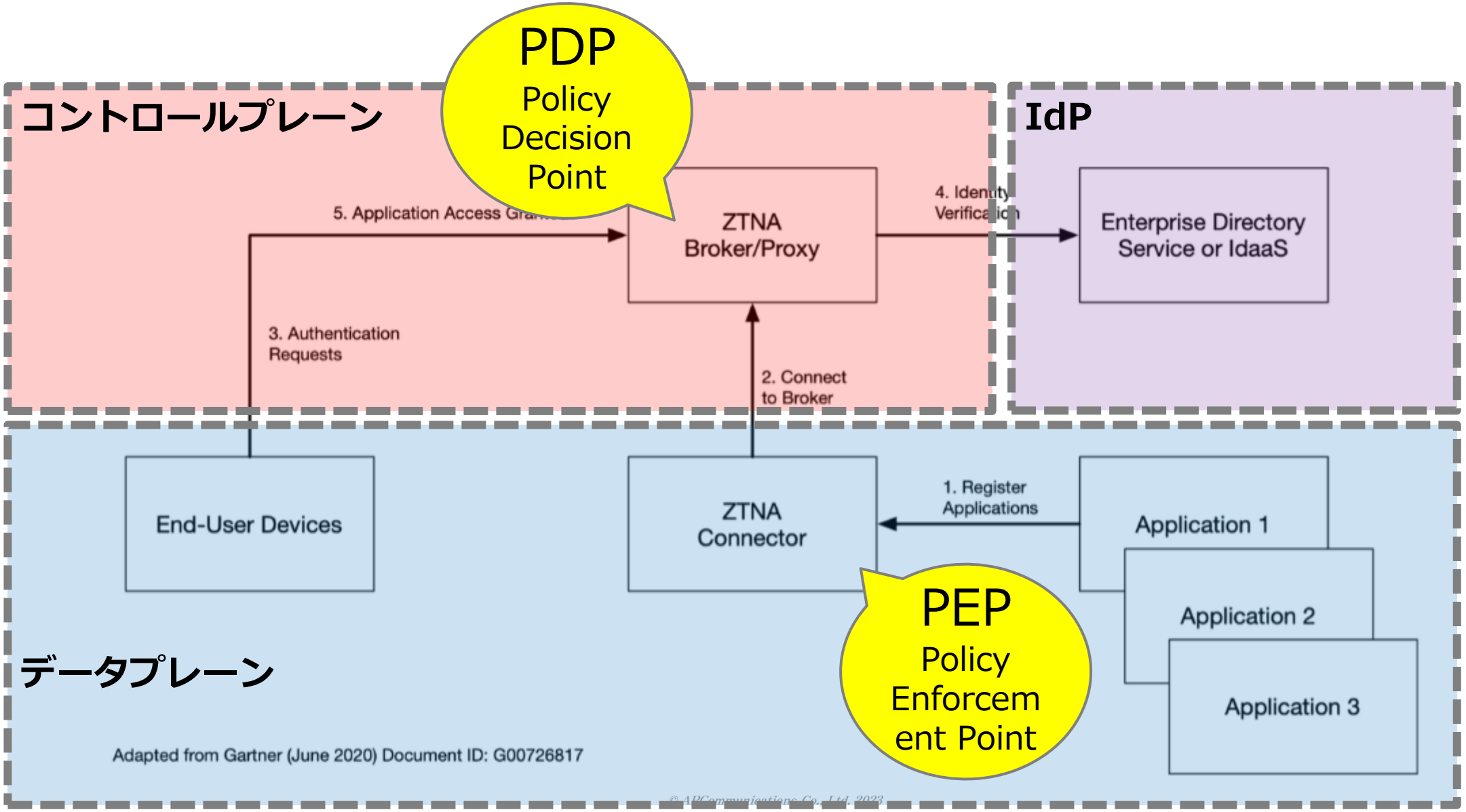
⑨ IH が AH と TLS通信を行い、AHと同居している企業リソースに到達することができます。



1. はじめに
2. Zero Trust Network Access
3. Zero Trust Network Accessの歴史
4. Reverse PROXY
5. Port knocking
6. Single Packet Authorization
7. VPNとZTNA/SDPの比較
8. Zero Trust Architecture の論理構成
9. SDPの仕組み
10. ZTNAの仕組み
11. おわりに

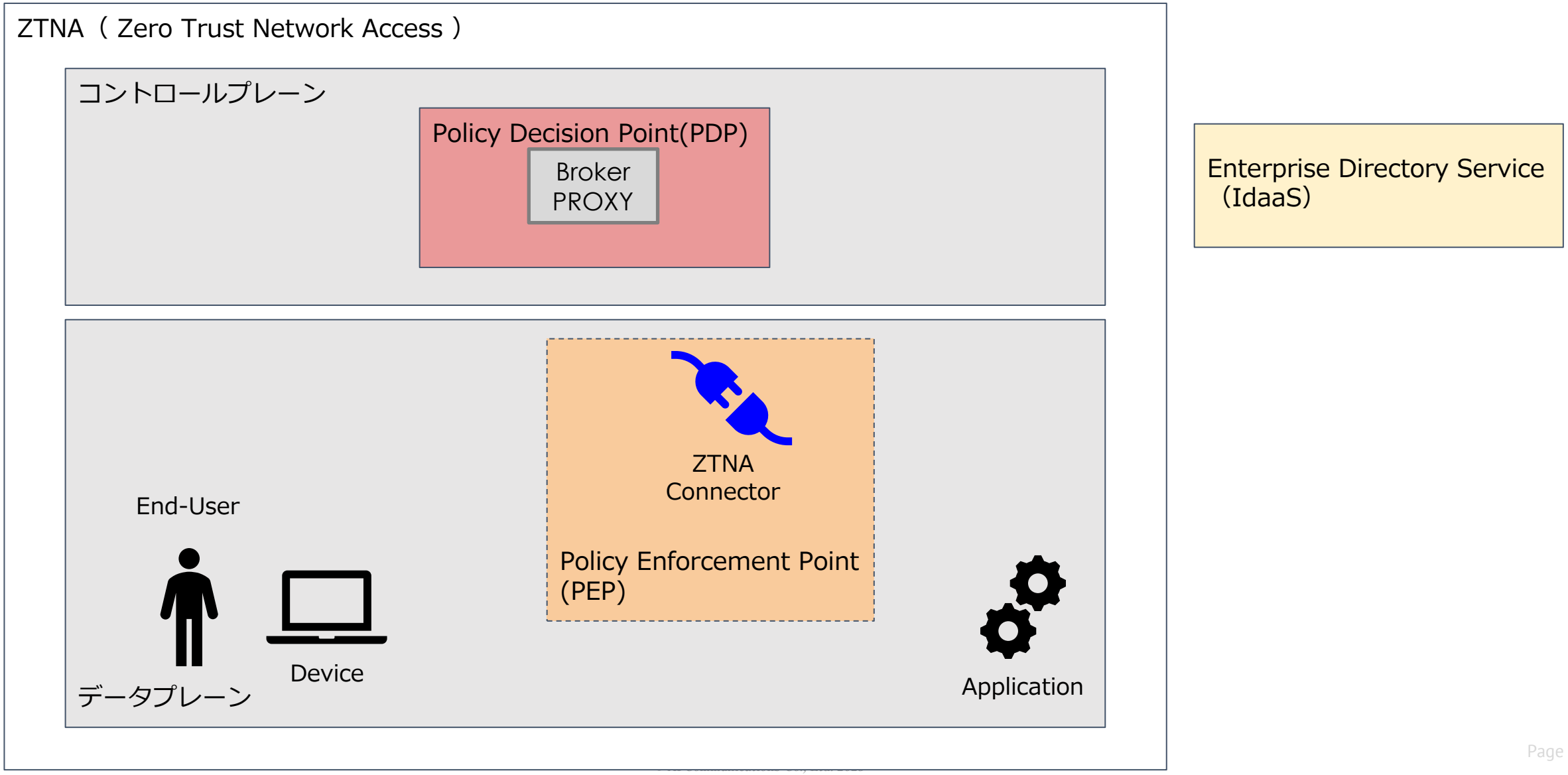
ZTNAの論理構成

ZTNAもコントロールプレーンとデータプレーンに分かれています。



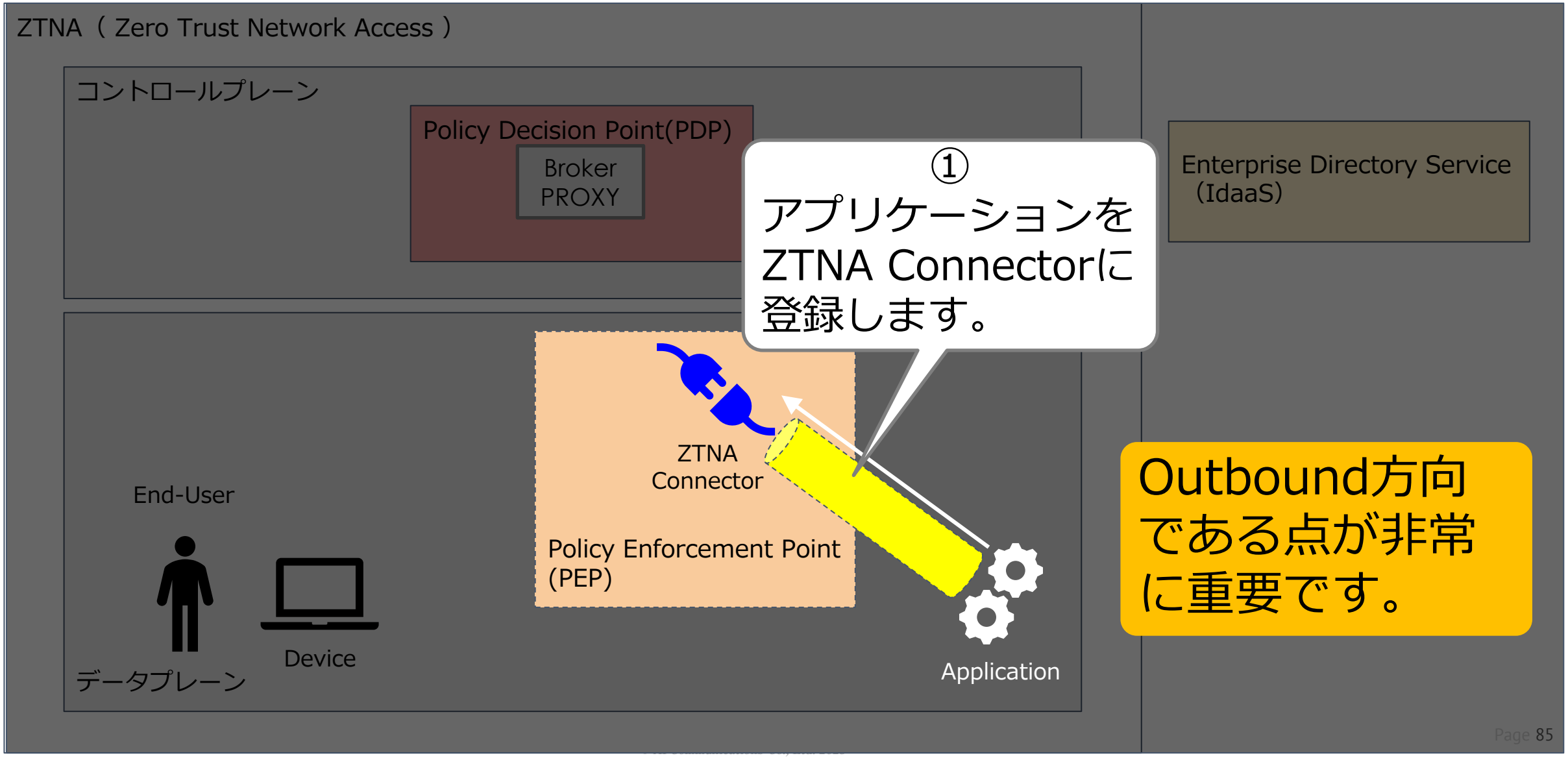
ZTNAの仕組み

Garter社が提示しているZTNAの図を基にZTAの簡易モデルを書き換えます。



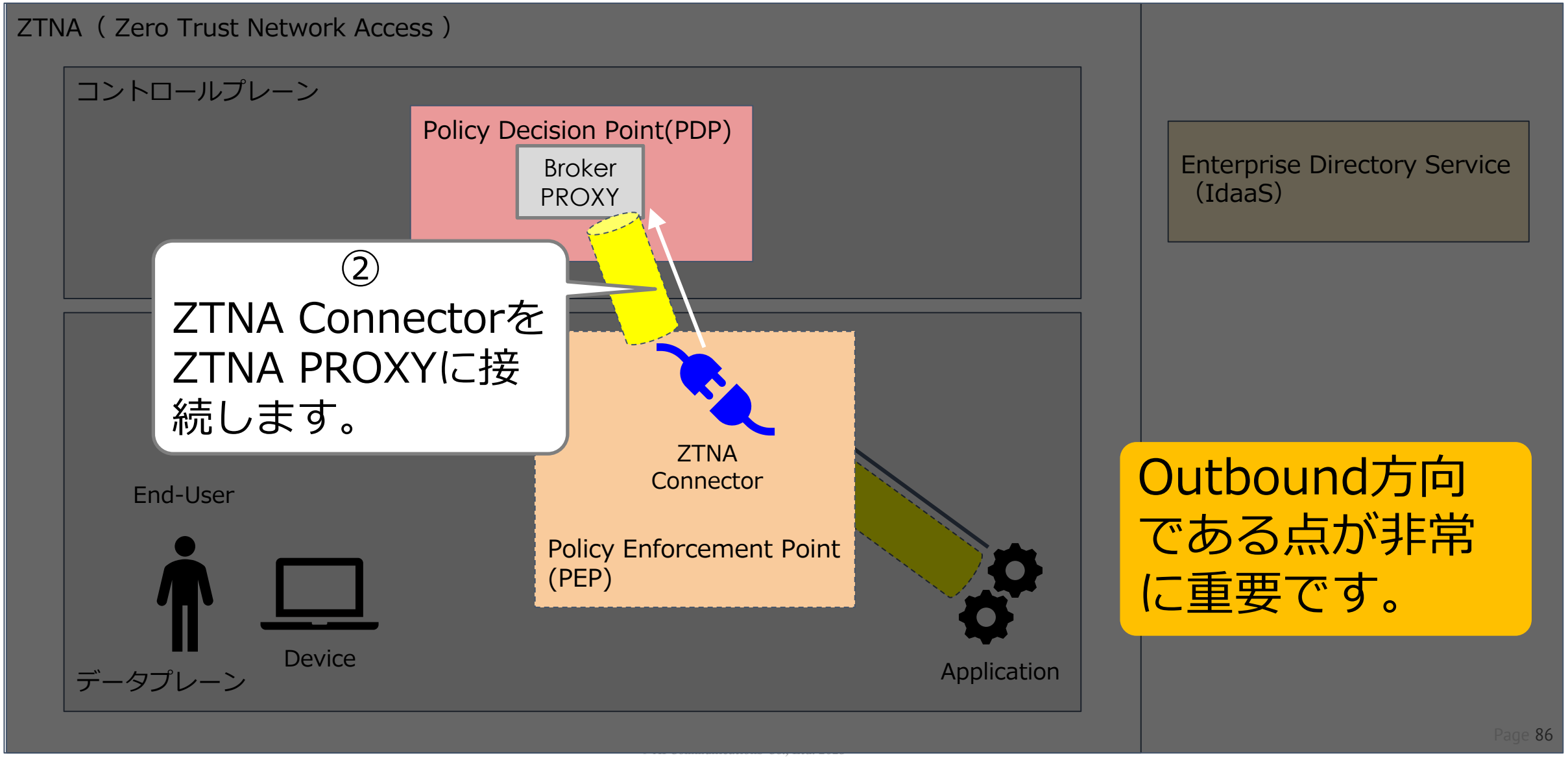
ZTNAの仕組み

① アプリケーションをZTNA Connectorに登録します。



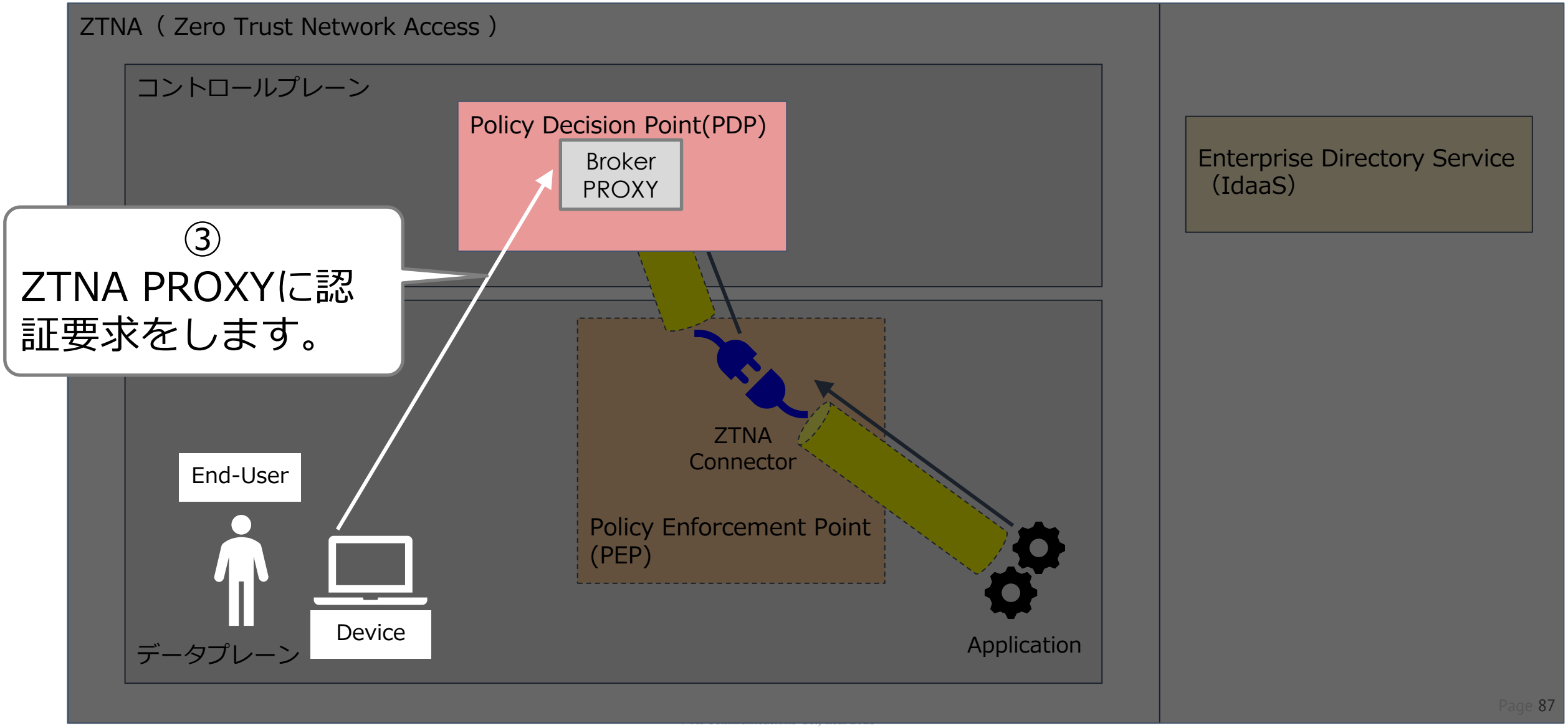
ZTNAの仕組み

② ZTNA ConnectorをZTNA PROXYに接続します。



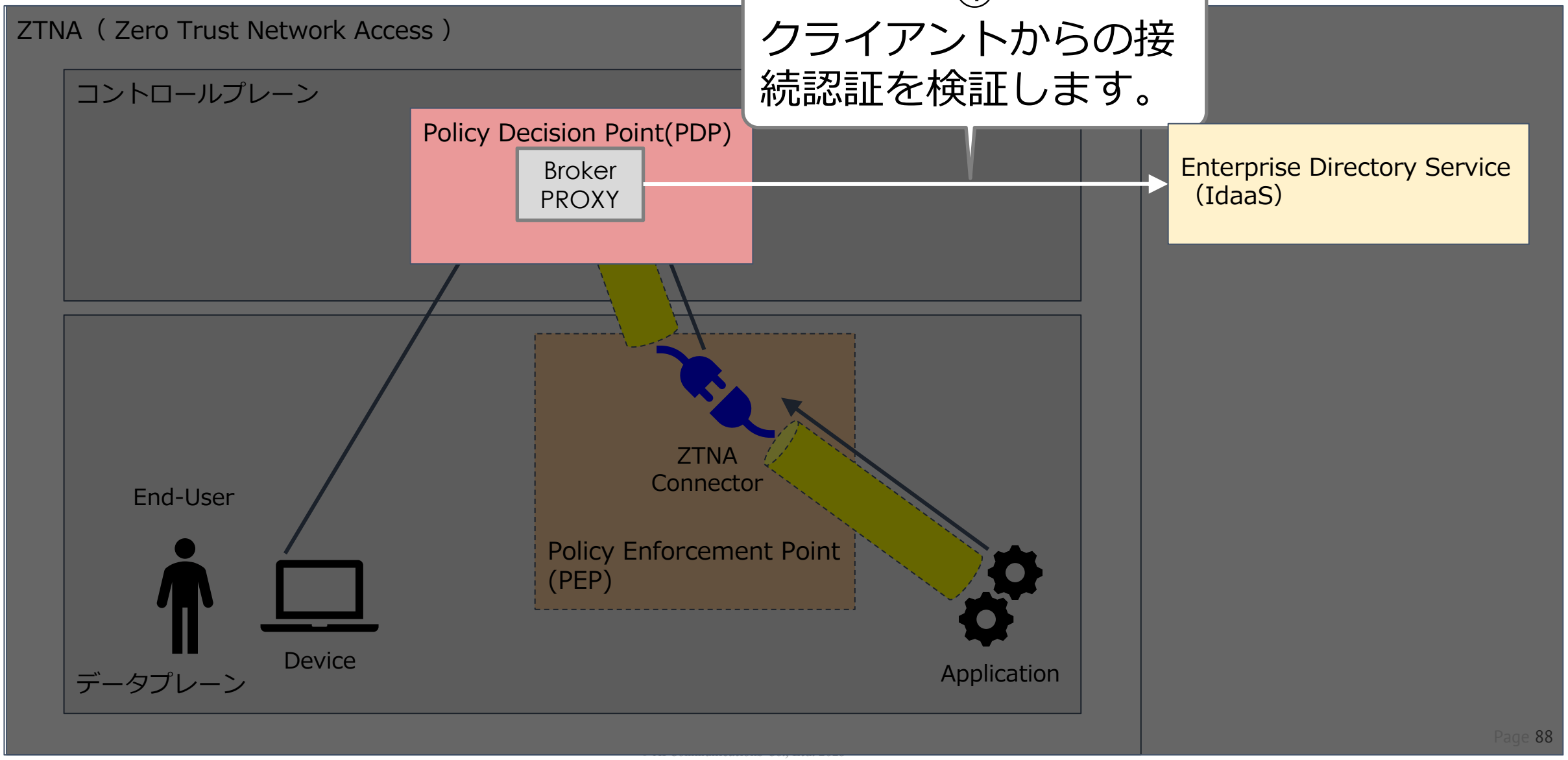
ZTNAの仕組み

③ ZTNA PROXYに認証要求をします。



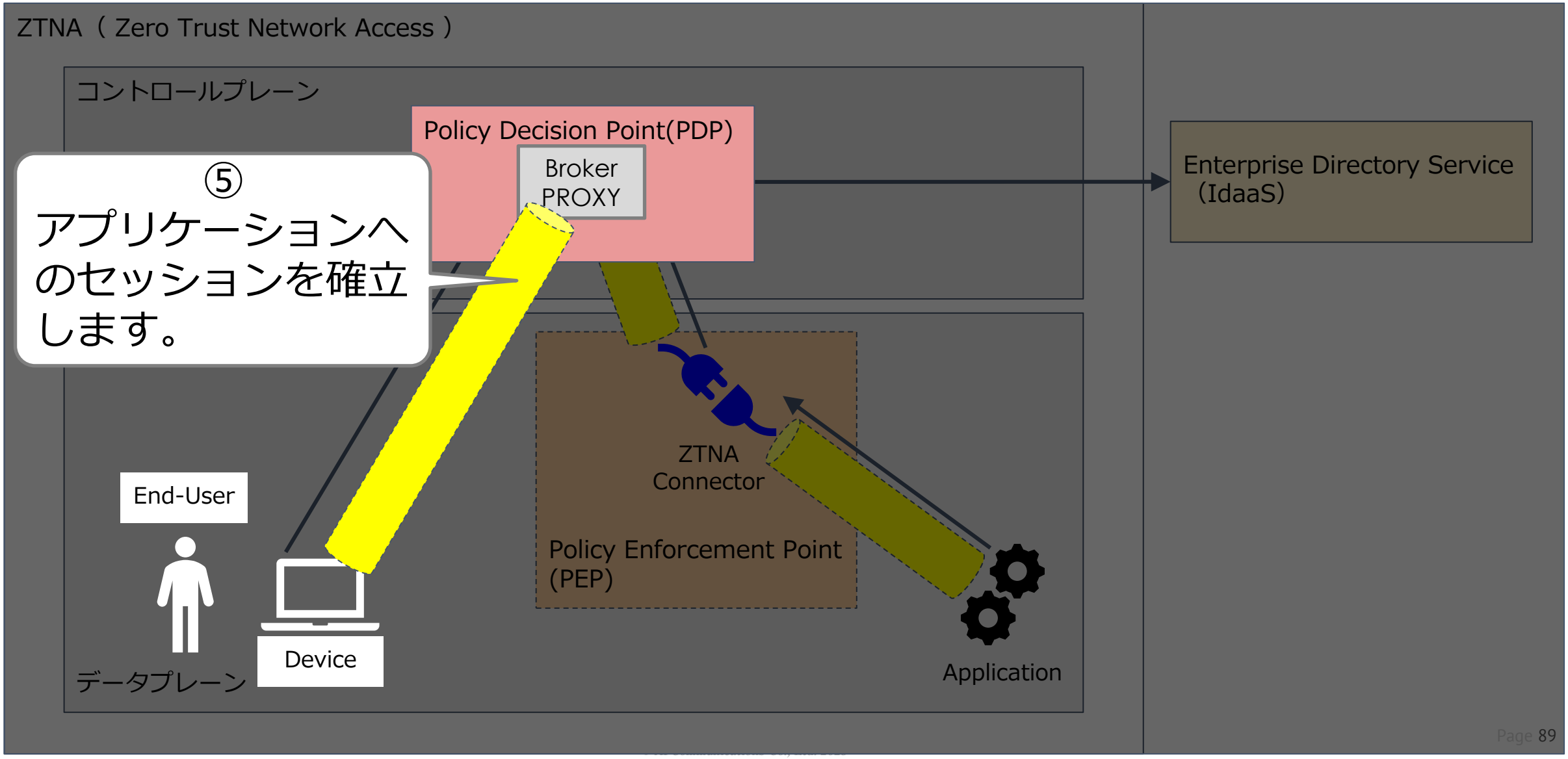
ZTNAの仕組み

④ クライアントからの接続認証を検証します。



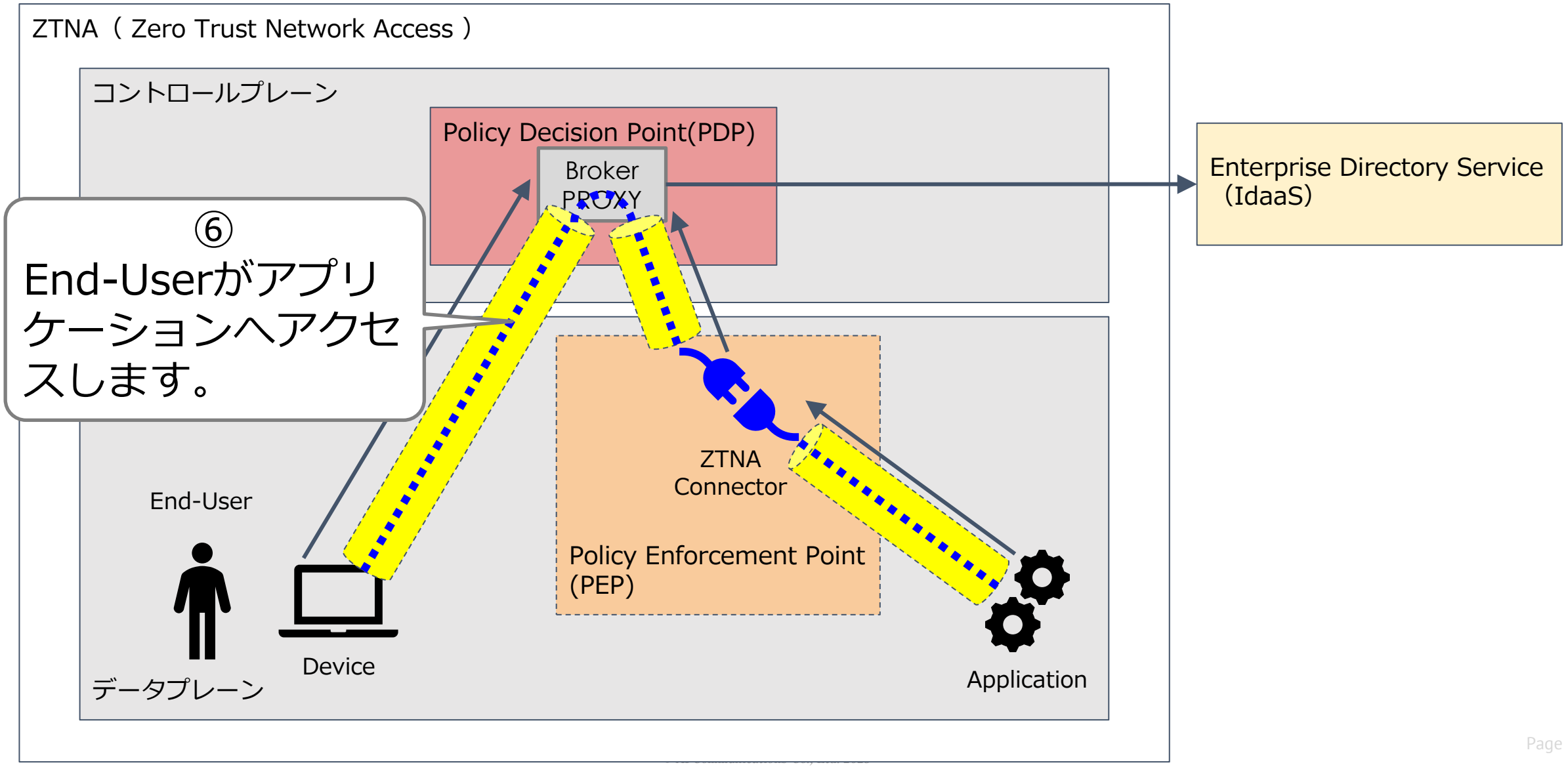
ZTNAの仕組み

⑤ アプリケーションへのセッションを確立します。



ZTNAの仕組み

⑥ End-Userがアプリケーションへアクセスします。



まとめ

1

VPN と ZTNA / SDPの違い

一般的なVPNと比較すると、認証のプロセス ／ システムの隠蔽性 ／ 攻撃対象領域 ／ 攻撃の検出など、ZTNA / SDPの方がセキュリティ強度が強い。

2

Zero Trust Architectureの論理構成

ポリシーの決定をしているコントロールプレーンと、ポリシーを適用・実施するデータプレーンに分かれている。

3

SDPの仕組み

通信を開始する前に接続前認証を複数個所で行い、認証に必要な情報をコントローラーから得て、異なる認証プロセスを経て通信を暗号化してます。

4

ZTNAの仕組み

アプリケーションはOutbound方向でのみconnectionを張るため、攻撃対象領域が削減される。Reverse PROXYとなるセキュリティベンダー（サービス提供者）が通信を保護する。

1. はじめに
2. Zero Trust Network Access
3. Zero Trust Network Accessの歴史
4. Reverse PROXY
5. Port knocking
6. Single Packet Authorization
7. VPNとZTNA/SDPの比較
8. Zero Trust Architecture の論理構成
9. SDPの仕組み
10. ZTNAの仕組み
11. おわりに

本セッションを振り返って

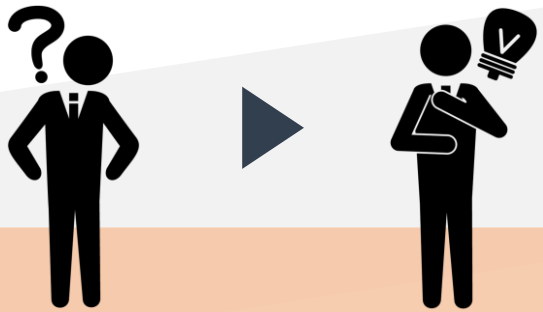
対象者

- ゼロトラストを学習中の方
- VPNとゼロトラストの違いに興味のある方

本セッションの目標 = Level2到達

Level1　興味

「Zero Trust Network Accessとは？」が理解できる



Level2　体感

「ZTNA/SDPの仕組み」が理解できる

Level3　体験

「脱VPNへの一歩：VPNとZTNA/SDPの違い」が理解・説明できる

「Zero Trust Network Accessって何？」⇒「ZTNA/SDPの仕組みが理解できた」となってもらうことが目標

脱VPNへの一歩

1. Attack Surfaceの把握/削減
2. 暗黙のトラスト領域を最小化
3. ZTNA/SDP について仕組みの理解
4. 何を目指す？何を目指したい？

質疑応答

ご質問のある方は挙手をお願いします。

名　前：XXXXX
会社名：XXXXX
一　言：この業界についての知見有無など
質　問：XXXXX

ご清聴ありがとうございました

Appendix

参考URL

■SDP Specification v1.0
https://cloudsecurityalliance.org/artifacts/sdp-specification-v1-0/

■Software-Defined Perimeter (SDP) Specification v2.0
https://cloudsecurityalliance.org/artifacts/software-defined-perimeter-zero-trust-specification-v2/

■ SDP Architecture Guide v2
https://cloudsecurityalliance.org/artifacts/sdp-architecture-guide-v2/

■ Software-Defined Perimeter (SDP) and Zero Trust
https://cloudsecurityalliance.org/artifacts/software-defined-perimeter-and-zero-trust/

■ Software-Defined Perimeter as a DDoS Prevention Mechanism
https://cloudsecurityalliance.org/artifacts/software-defined-perimeter-as-a-ddos-prevention-mechanism/

■Zero Trust Architecture（NIST SP800-207）
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf

■ソフトウェア定義の境界とは
https://www.zscaler.jp/resources/security-terms-glossary/what-is-software-defined-perimeter

■How to choose a Zero Trust architecture: SDP or Reverse-Proxy?
https://cloudsecurityalliance.org/blog/2021/02/15/how-to-choose-a-zero-trust-architecture-sdp-or-reverse-proxy

■SPA （Single Packet Authorization）解説
https://cloudsecurityalliance.jp/newblog/2019/08/27/448/

■Single Packet Authorization A Comprehensive Guide to Strong Service Concealment with fwknop
https://www.cipherdyne.org/fwknop/docs/fwknop-tutorial.html

Port knockingの構築（1/2）

Server（Knock-server）とClient（Knocker）で若干構築手順が異なります。

Server（Knock-server）

① install　knock-server-0.7-1.el7.nux.x86_64.rpm

事前にlibpcap※が必要

【コマンド】
rpm -ivh http://li.nux.ro/download/nux/dextop/el7Server/x86_64/knock-server-0.7-1.el7.nux.x86_64.rpm

② edit　/etc/sysconfig/knockd

③ edit　/etc/knockd.conf

事前にlibpcap※が必要

Client（Knocker）

① install　knock-0.7-1.el7.nux.x86_64.rpm

【コマンド】
rpm -ivh http://li.nux.ro/download/nux/dextop/el7Server/x86_64/knock-0.7-1.el7.nux.x86_64.rpm

※What is libpcap used for?

Libpcap enables administrators to capture and filter packets. Packet sniffing tools like tcpdump use the Libpcap format. For Windows users, there is the WinPcap format. WinPcap is another portable packet capture library designed for Windows devices.

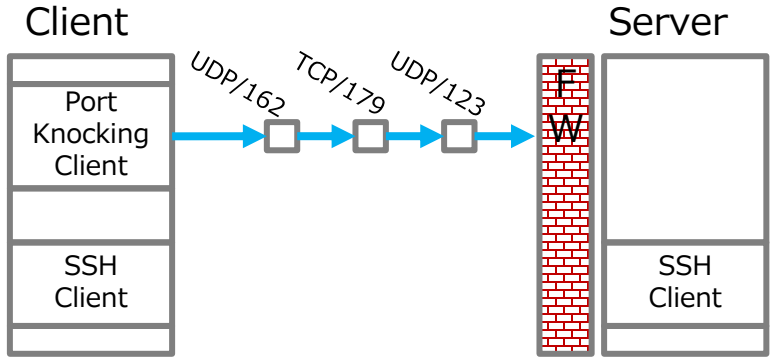
WindowsでいうところのWinPcap
→ パケットキャプチャに使うライブラリのことです。

Port knocking構築（2/2）

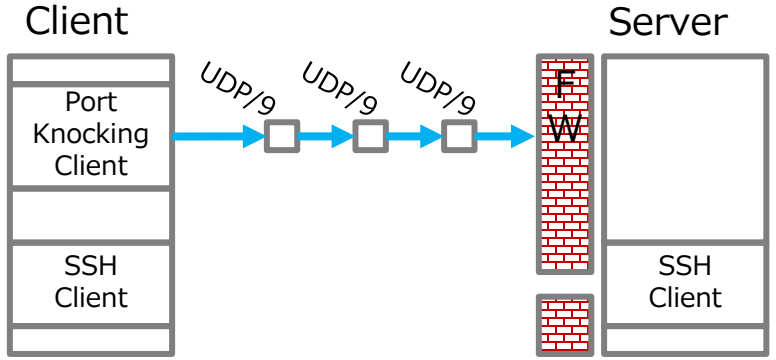
```
/etc/knockd.conf
!
[options]
        UseSyslog
logfile = /var/log/knockd.log

[openSSH]
        sequence      = 123:udp,179:tcp,162:udp
        seq_timeout   = 15
        tcpflags      = syn
        start_command = /sbin/iptables -I INPUT -s 172.16.12.2 -p tcp --dport ssh -j ACCEPT
        cmd_timeout   = 10

[closeSSH]
        sequence      = 9:udp,9:udp,9:udp
        seq_timeout   = 15
        tcpflags      = syn
        start_command = /sbin/iptables -D INPUT -s 172.16.12.2 -p tcp --dport ssh -j ACCEPT
        cmd_timeout   = 10
```

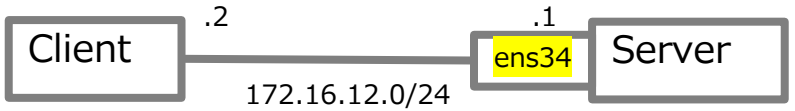
```
/etc/sysconfig/knockd
!
OPTIONS="-i ens34"
```

ServerのFWを開ける時



ServerのFWを閉める時





SPAの構築（1/2）

SPAの構築手順（概要）は以下のとおりです。

Server と Client 共通

① tarball ダウンロード & 解凍

【コマンド】
wget http://www.cipherdyne.org/fwknop/download/fwknop-2.6.10.tar.gz

【コマンド】
tar xfz fwknop-2.6.10.tar.gz

② configure

【コマンド】
cd fwknop-2.6.10

./configure --prefix=/usr --sysconfdir=/etc && make

事前に libpcap が必要

③ make install

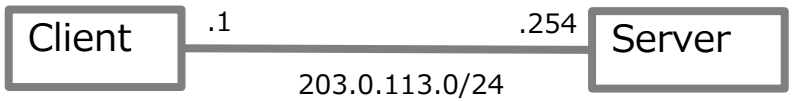
【コマンド】
make install

Client 側での作業

① KEY 生成

以下に生成する $HOME/.fwknoprc

【コマンド】
fwknop -A tcp/22 -a 203.0.113.1 -D 203.0.113.254 --key-gen --use-hmac --save-rc-stanza



Server 側での作業

① edit /etc/fwknop/ fwknop.conf

② edit /etc/fwknop/ access.conf

SPAの構築（2/2）

/etc/fwknop/access.conf

Server side

Clientにて生成したKEYをコピペ

```
!
  ～　途中省略　～
#### fwknopd access.conf stanzas ###

SOURCE              ANY
KEY_BASE64          xO5mM5lEJUVKxMn6PcNUKTn1qdivpLA1AHsMALKdhlU=
HMAC_KEY_BASE64     i0Asqvm0zGB867vcZT15RlL9TWrkbUs+4tNXAemTYF/D4MBWC                    EPMBc/TNlGYwTlLCEVbVQ==

  ～　以下省略　～
```

/etc/fwknop/fwknopd.conf

Server side

```
###############################################################
#
#       [+] fwknopd - Firewall Knock Operator Daemon [+]
#
# This is the configuration file for fwknopd, the Firewall Knock Operator
# daemon.  The primary authentication and authorization mechanism offered
  ～　途中省略　～
# Define the ethernet interface on which we will sniff packets.
# Default if not set is eth0.  The '-i <intf>' command line option overrides
# the PCAP_INTF setting.
#
PCAP_INTF               ens224;
  ～　以下省略　～
```