ITの、つぎの、幸せへ。

クラウドジャーニーを実現するための第一歩
アマゾン ウェブ サービス活用ノウハウを集約した
“秘伝のタレ”
USiZE クラウドリファレンスキット
for AWS

SCSK株式会社
ITマネジメント事業部門
基盤インテグレーション事業本部
クラウド基盤サービス部
AWS Advisory Consultant
菊田 慶貴

自己紹介

菊田 慶貴（きくた　よしたか）

- クラウドアドバイザリーコンサルタント
- AWS Solution Architect Professional

SCSK株式会社
ITマネジメント事業部門
基盤インテグレーション事業本部
クラウド基盤サービス部

主な仕事： クラウドの導入コンサルタント、構築、社内外への教育

お客様のビジネスの
新しい価値創造を支えるために

人を大切にしながら、
成長し続ける会社であるために

SCSK

夢ある未来を、共に創る。

16:22

月間平均残業時間

18.8日

年間平均有給休暇取得日数

96.8%

育児休業復職率

SCSK

夢ある未来を、共に創る。

NADESHIKO BRAND 2019

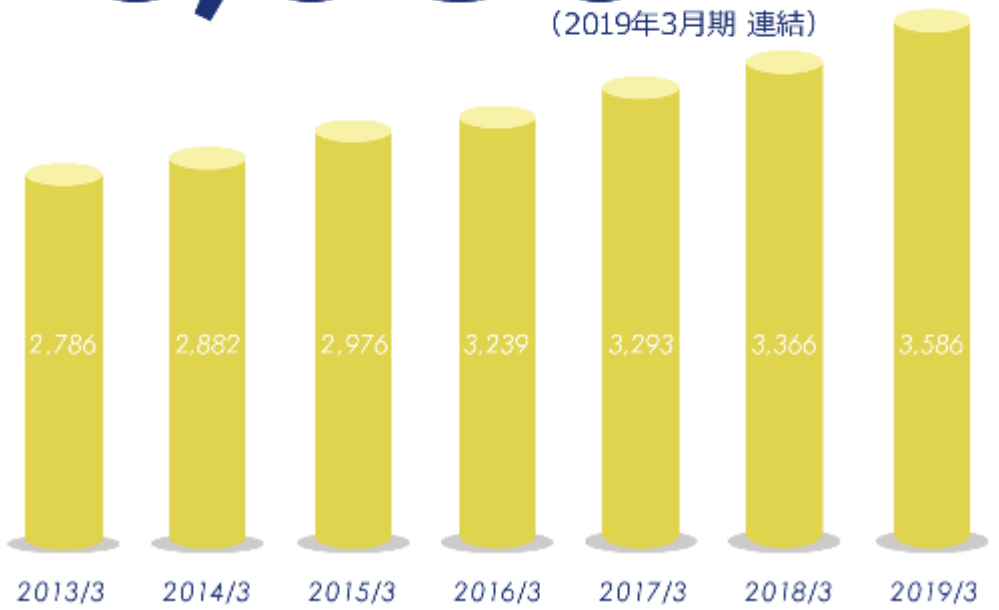
DIVERSITY MANAGEMENT 2.0 PRIME

2019 健康経営銘柄 Health and Productivity

売上高

3,586億円

（2019年3月期 連結）



ITの、つぎの、幸せへ。

SCSK

本日のアジェンダ

- SCSKのクラウドサービスUSiZEのご紹介
- クラウドファースト？なぜクラウドを使うのか
- クラウドジャーニーへようこそ！
  -クラウド利活用に向けたシステム仕訳のツボ-
- 弊社“秘伝のタレ”のご紹介

SCSKのクラウドサービス
USiZEのご紹介

SCSKのクラウドサービス、USiZE

お客様が必要な時に、必要な分だけ
高品質なITリソースを提供

［ Utility You/Your ＋ SiZE ］

powered by aws

アマゾン ウェブ サービスを「USiZEパブリッククラウドモデル(AWS)」としてリセール

SCSKのアマゾン ウェブ サービスへの取り組み

AWS Partner Network (APN)
アドバンスドコンサルティングパートナー

・東京リージョン開設と同時に活動を開始

・顧客数: 506社（大手金融、流通、不動産、製造業が中心）

・エンタープライズの基幹・分散基盤のクラウドマイグレーション
　＝LIFT案件に多くの実績。（DX系も昨今利用拡大）

SCSKのアマゾン ウェブ サービスへの取り組み

- **有資格者数（2019年3月末時点）：延べ914名**
  - Business Profesional　339名
  - Technical Professional　332名
  - Solution Architect Associate　145名
  - Solution Architect Professional　83名
  - DevOps Professional他開発系資格　15名

FY19には延べ1,000名突破の見込み

- **受賞歴**

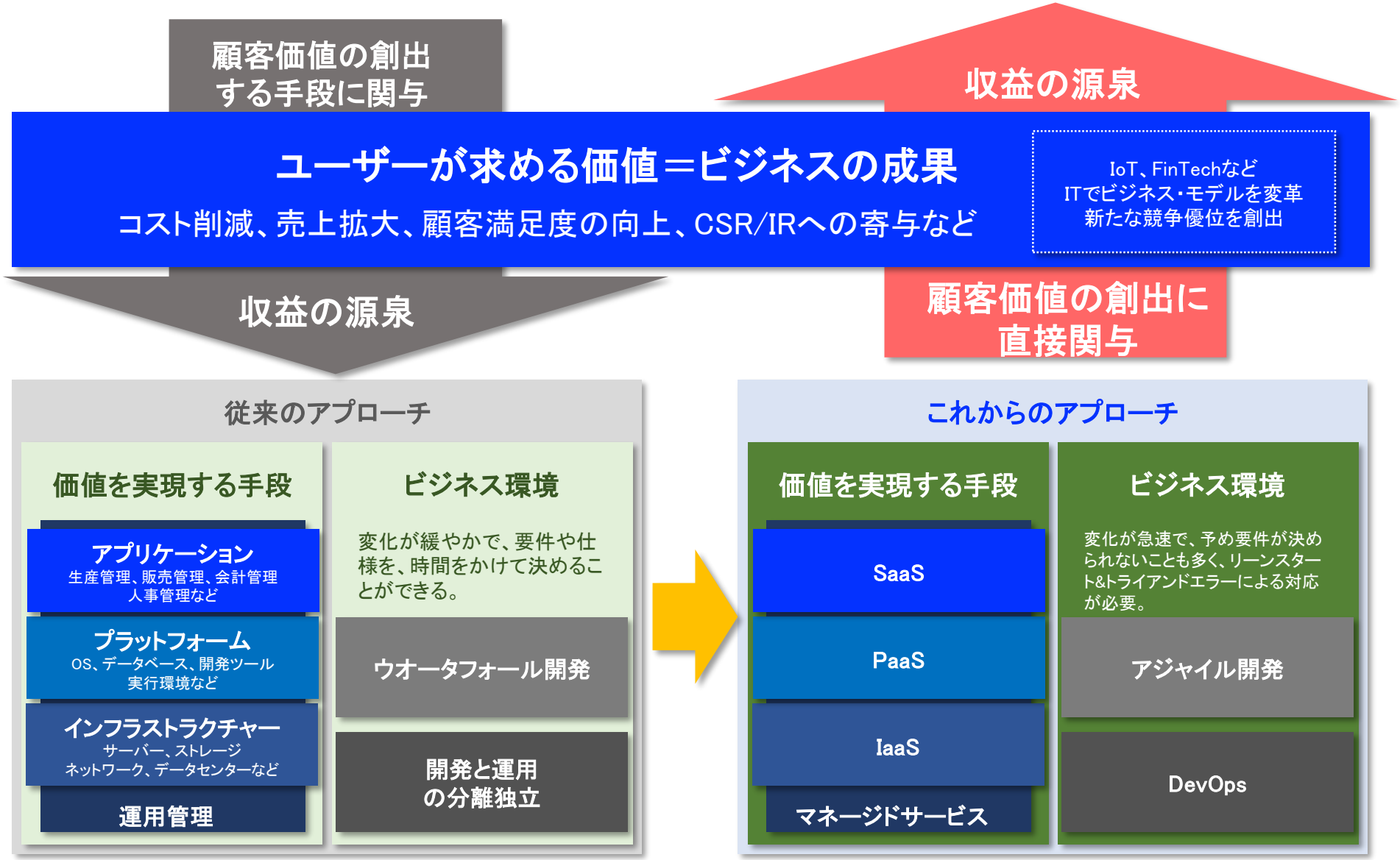
  Partner Award受賞　2回

  - 金融庁 FISC 安対リファレンス

  - ハイパフォーマンスコンピューティング（HPC）の活動

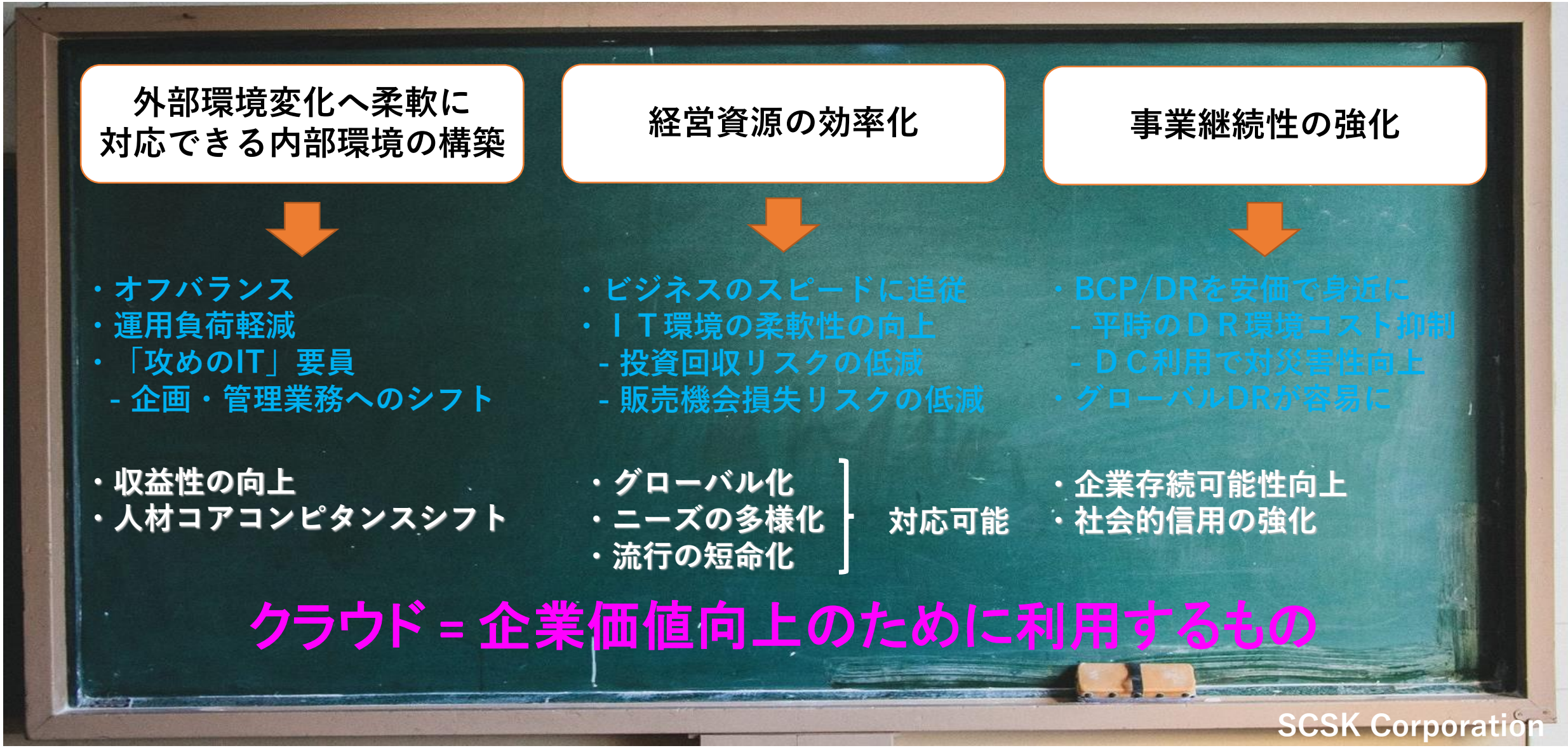
クラウドファースト？
なぜクラウドを使うのか。

手段ではなく、価値の提供が求められる時代に



出展：ネットコマース株式会社
「最新のITトレンドとビジネス戦略」

クラウドをビジネスで活用するメリット



ビジネストラックとテクノロジートラックの2軸で思考

ビジネス
トラック

テクノロジ
トラック

クラウドはビジネスIssueに対する手段(Solution)の一つ

例えば、「なぜ、アマゾンはAWSを始めたのか」を考える

AWSは、Amazon社内の**ビジネス課題**
を解決するために生まれた

Amazonのビジネス課題	発生したITの課題
• 過去の全注文履歴は全て保管	• 増え続けるデータや処理負荷、、
• 全配送センターの物品管理	• 増え続ける調達の手間
• アフィリエイトの支払管理、etc.	• 従来型アーキテクチャの限界

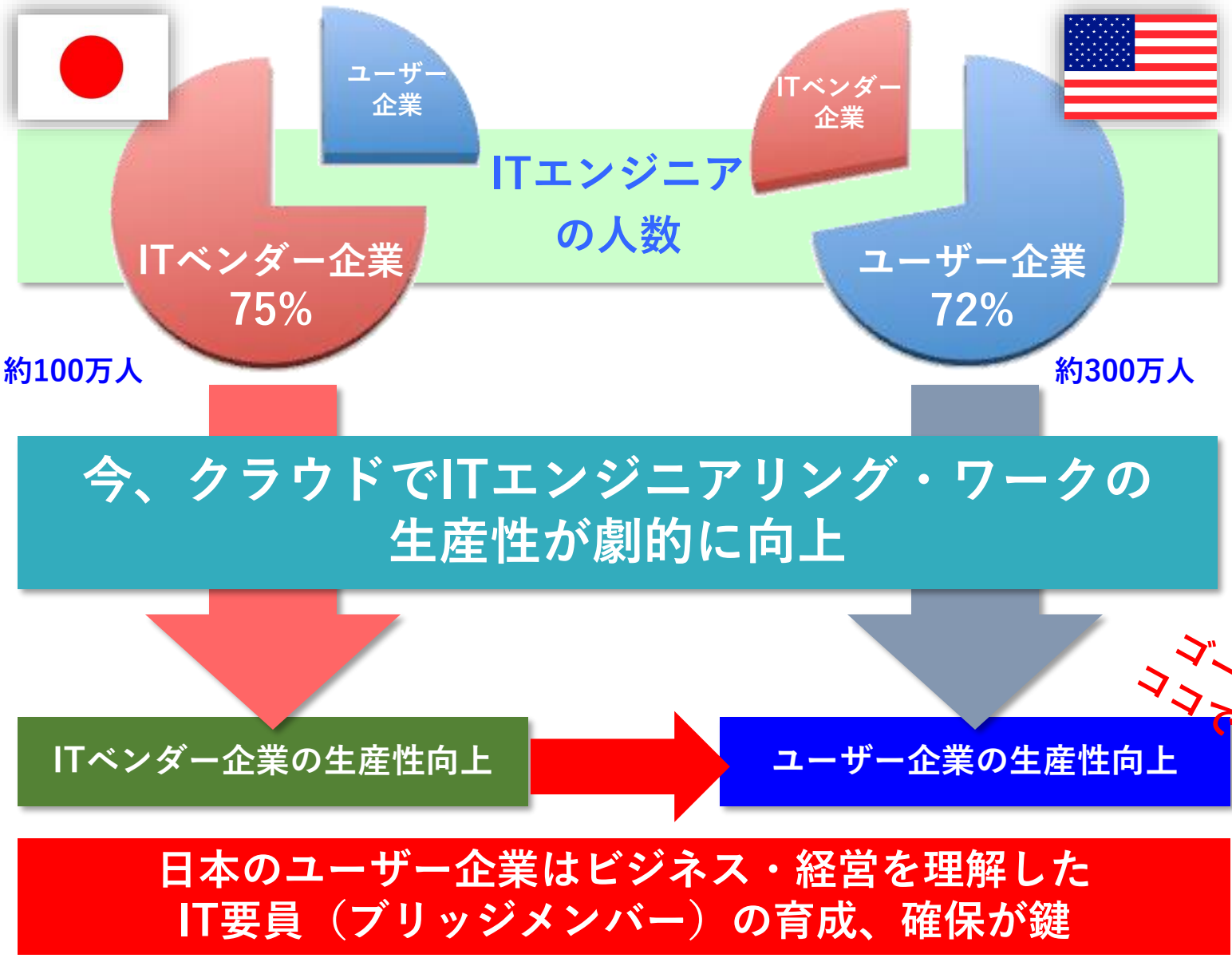
これらを、”クラウド”で解決!!

スケーラブルなリソース、疎結合＆APIの仕様をオープンに利活用

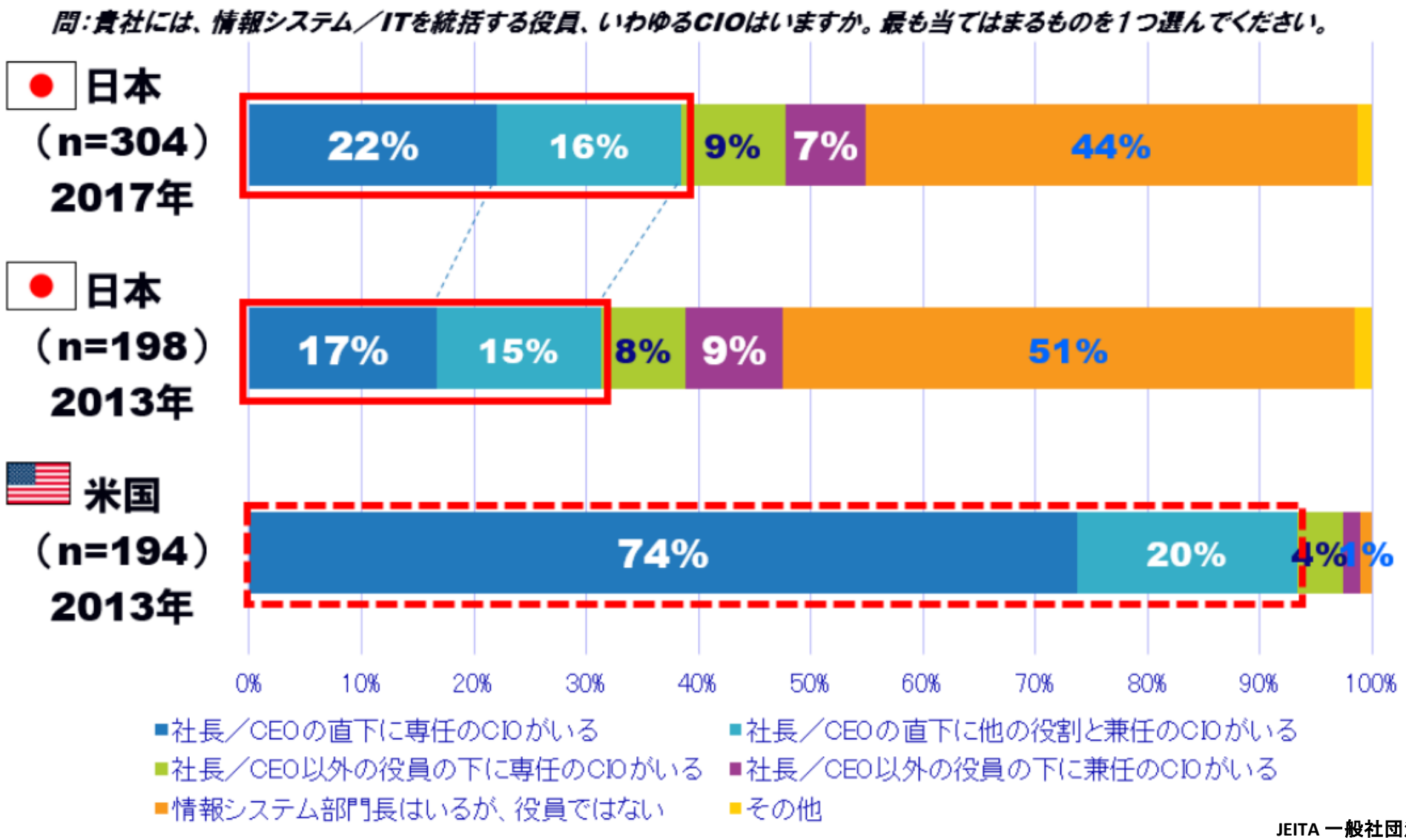
ビジネス上のIssueを解決するため　→　企業価値向上のため

ただし
ヒト・モノ・カネ、経営リソースは有限
（特にヒト）

日米におけるITエンジニア数の差



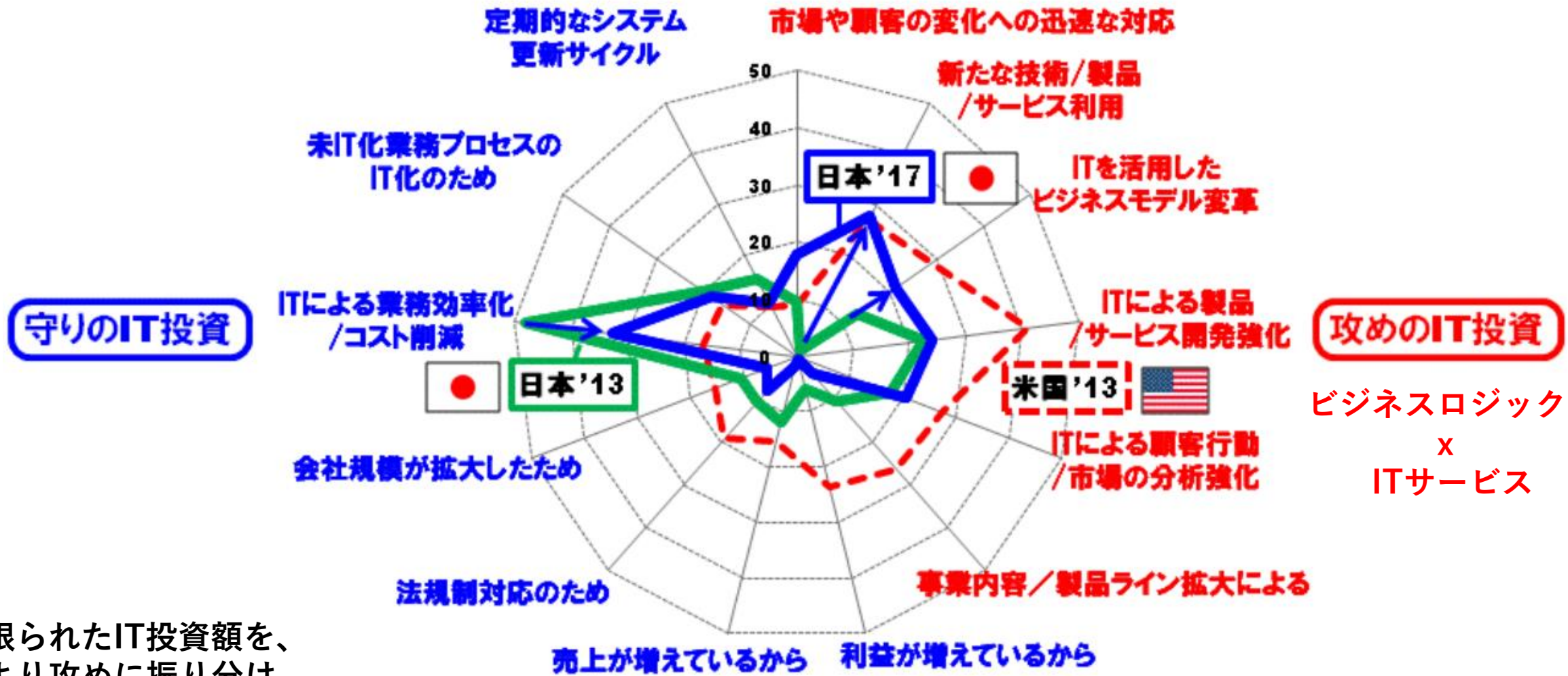
*2 CIO の設置状況



JEITA 一般社団法人電子情報技術産業協会
『国内の民間企業におけるITに対する意識調査』,2017年9月

SCSK Corporation

*4 IT 予算が増える理由/用途



限られたIT投資額を、
より攻めに振り分け、
ユーザ企業の限られたIT要員を
攻めの要員にシフトさせる。

クラウドの利活用による最強の守りは、攻めの源泉でもある。

JEITA 一般社団法人電子情報技術産業協会
『国内の民間企業におけるITに対する意識調査』,2017年9月

攻めのIT
事例3本勝負！

事例①　(製造業)

要件

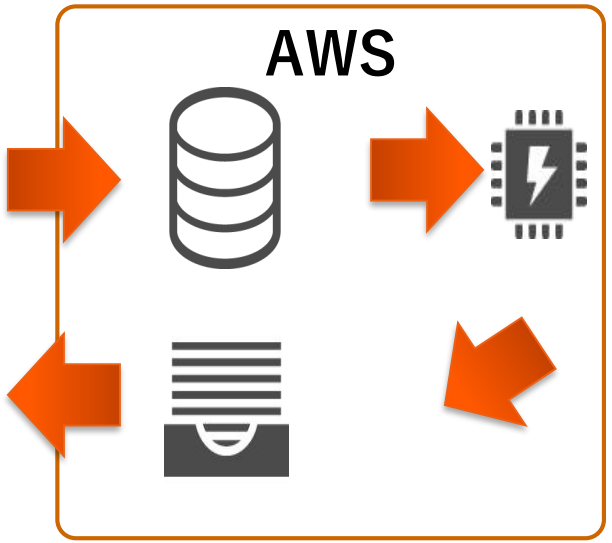
突発的な大量データ解析ニーズに対応する(※)

・自前でリソースが持てないTier2以下のサプライヤへの対応
・リソースを保有する大手の業務的割込みや突発ニーズへ対応

※computer aided engineering
コンピュータ技術を活用して製品の設計、製造、検証の支援を行うこと、またはそれを行うツール

製造業の設計現場の悩み
短納期化、大規模化、突発事態(リコール等)に対応した十分なコンピュータリソースがなく効率が上がらない

クラウドで使いたい時だけ使う



ビジネス: サプライチェーン全体の生産性向上、歩留まり解消
テクノロジ:「CAI & HPC」「Bigdata」

事例②　(製造業)

要件

IoTを活用し重機の稼動・メンテナンス情報を収集しユーザをサポート

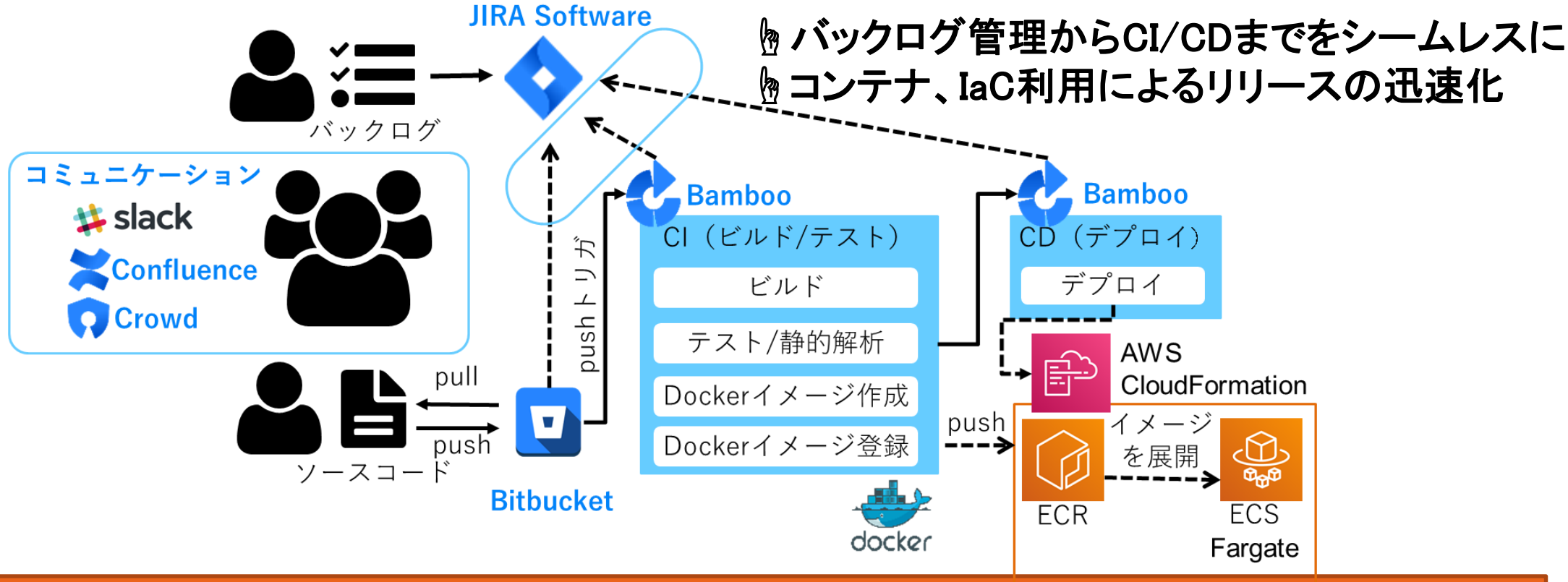
・エラーの早期検知と早期保守
・稼働状況の見える化によるユーザのサポート

ビジネス: 高信頼性と使い易さで機器のシェア拡大、買替喚起
テクノロジ:「テレマティクス」「Bigdata & AI」

事例③ (金融業)

要件

迅速なサービスリリースと、チャレンジの容易化
- ビジネス要件、顧客要望に対する素早い対応
- インフラ投資は極力スモールに（Agility重視）

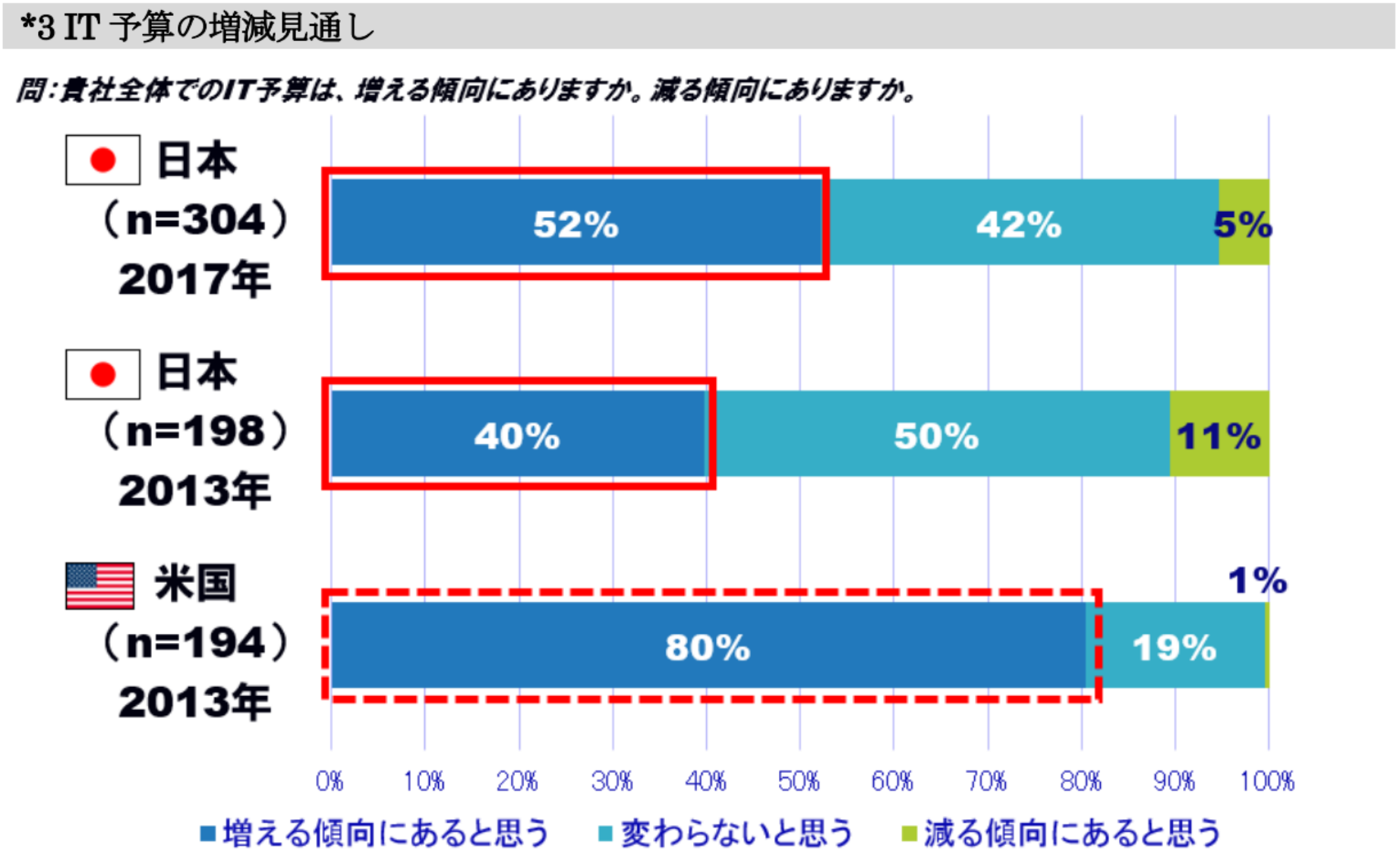


バックログ管理からCI/CDまでをシームレスに

コンテナ、IaC利用によるリリースの迅速化

ビジネス: 顧客満足度の向上、ビジネス局面変化への対応
テクノロジ: 継続的インテグレーション/継続的デリバリー(CI/CD)

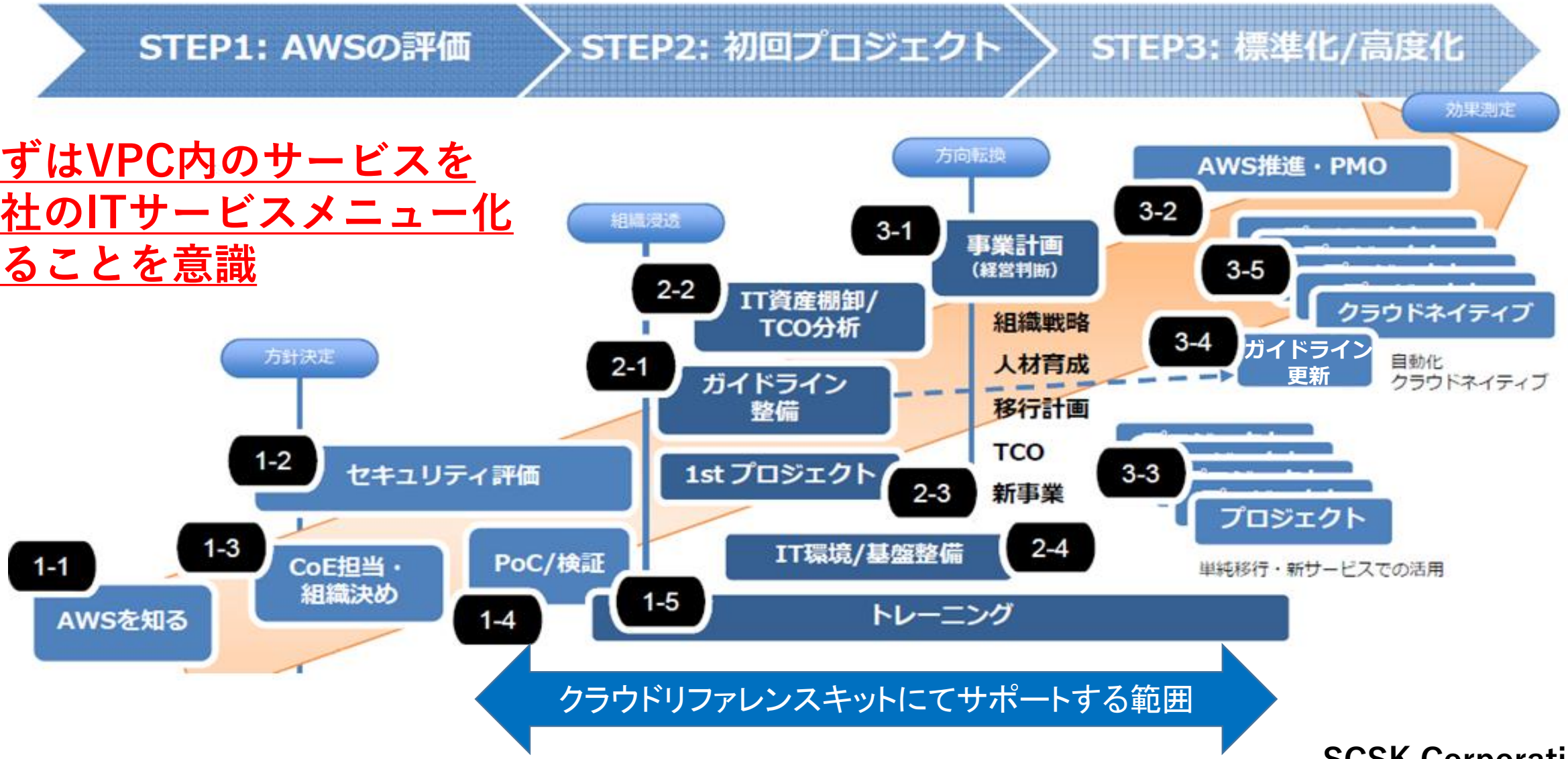
ITはコスト削減手段のみならず。投資である。



JEITA 一般社団法人電子情報技術産業協会
『国内の民間企業におけるITに対する意識調査』,2017年9月

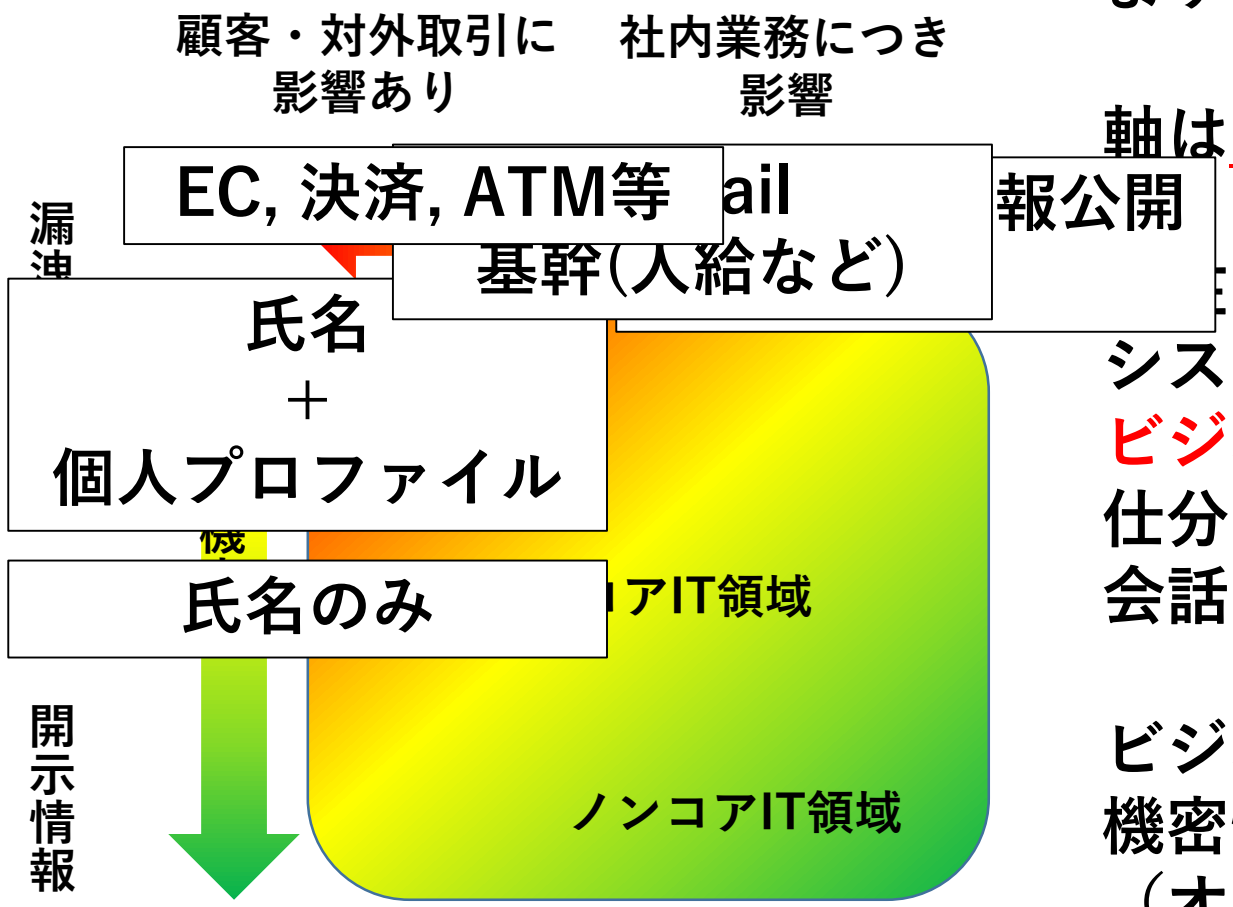
クラウドジャーニーへようこそ！
-クラウド利活用に向けたシステム仕訳のツボ-

クラウドジャーニーのロードマップ



まずはVPC内のサービスを自社のITサービスメニュー化することを意識

業務を知る、リスクベースドアプローチ



クラウドサービスにはSLAが存在するため、
まずは自社の業務が求めるSLAを把握する。

軸は可用性と機密性の2軸

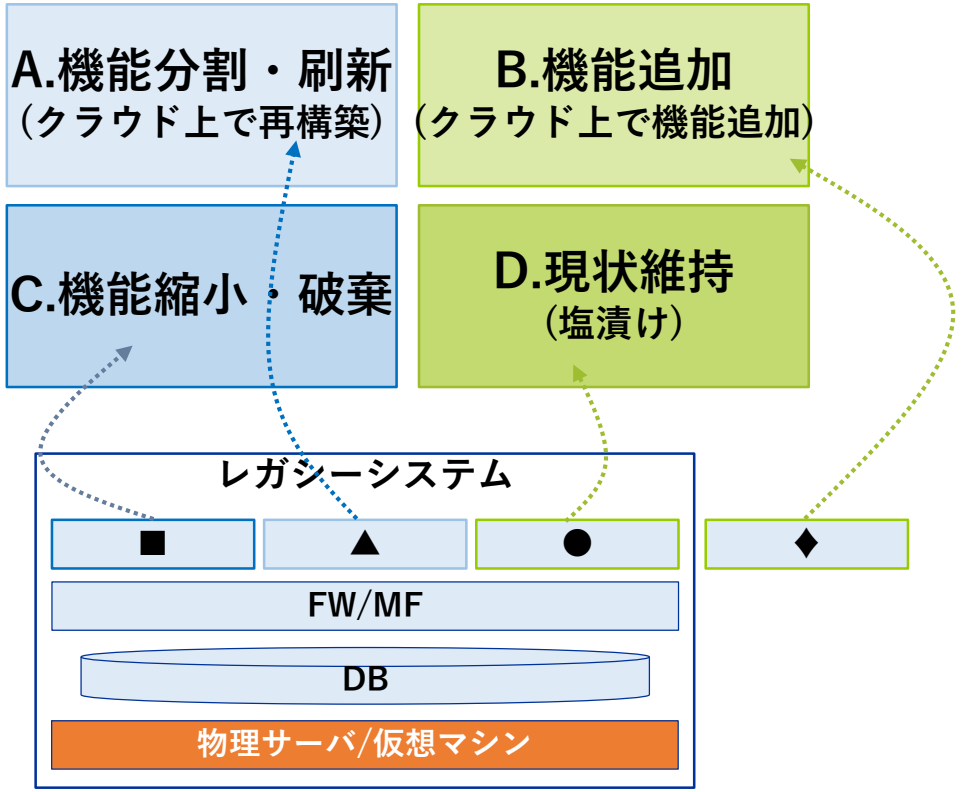
のインフラ構成ではなく、
システムダウン＆情報漏洩した際の
ビジネスインパクト(実損)が高いか、低いかで
仕分けするとビジネスユニットおよび経営層との
会話が成立しやすい。

ビジネスインパクトに見合わない可用性、
機密性の実装は過剰投資、要・見直し。
（オンプレミスのクラスタ構成は特に注意）

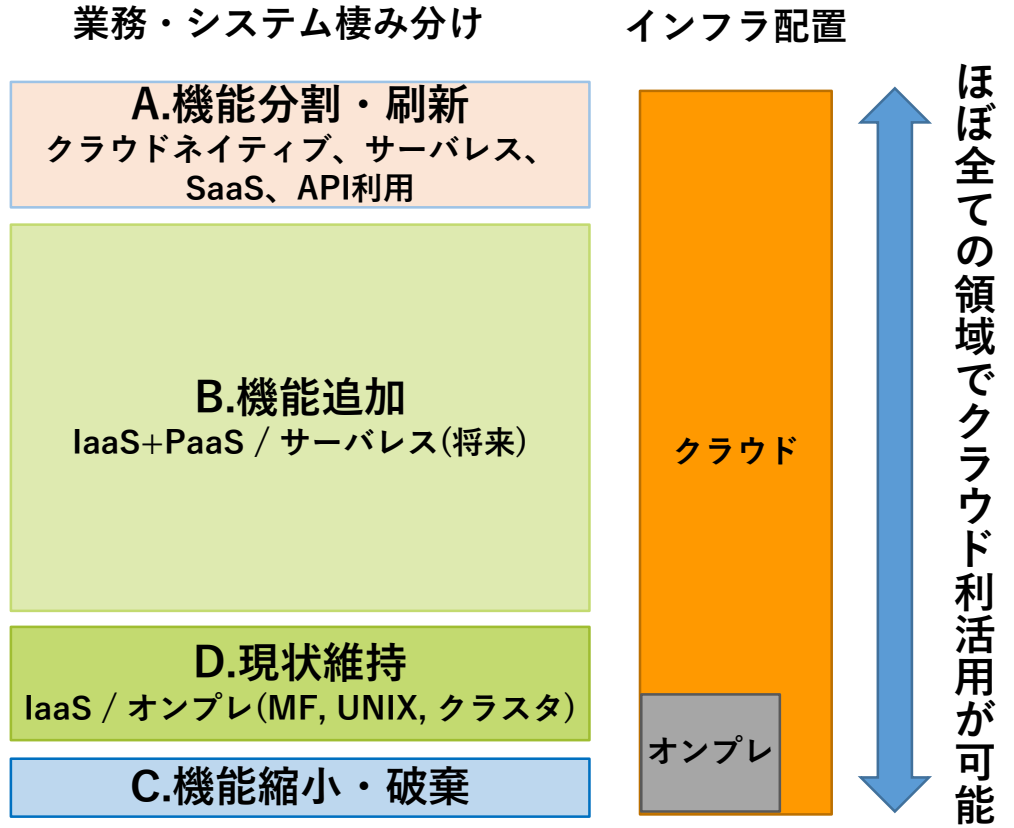
クラウドファースト、情報資産の現状評価と仕訳

※平成30年9月7日経済産業省　デジタルトランスフォーメーションに向けた研究会
「DXレポート　～ITシステム「2025年の崖」の克服とDXの本格的な展開～」
　DX推進システムガイドラインの構成案より

A: 頻繁に変更が発生する機能は**クラウド上で再構築**
B: 変更されたり、新たに必要な機能は**適宜クラウドへ追加**
C: 肥大化したシステムの中に**不要な機能があれば破棄**
D: あまり更新が発生しない機能は**塩漬け**

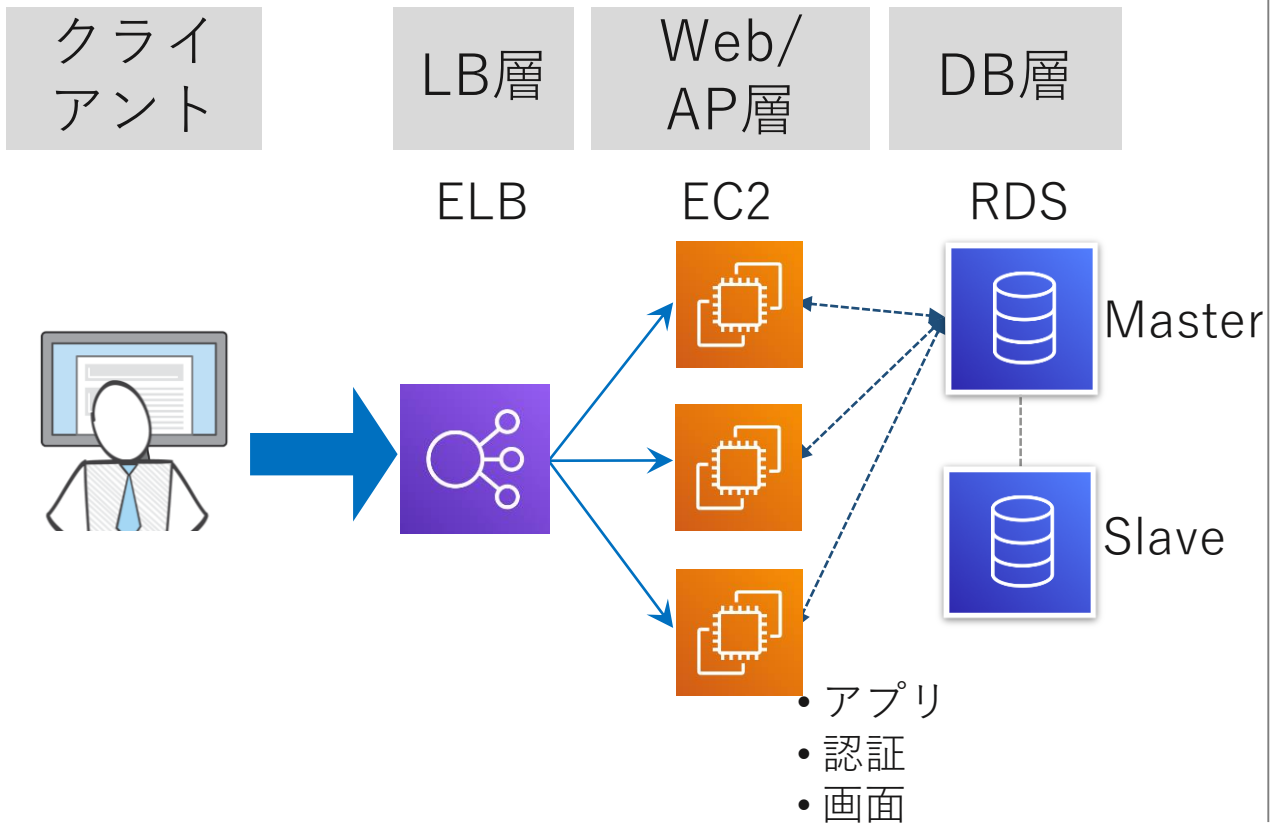


弊社が実際にコンサルテーションし、仕訳した例



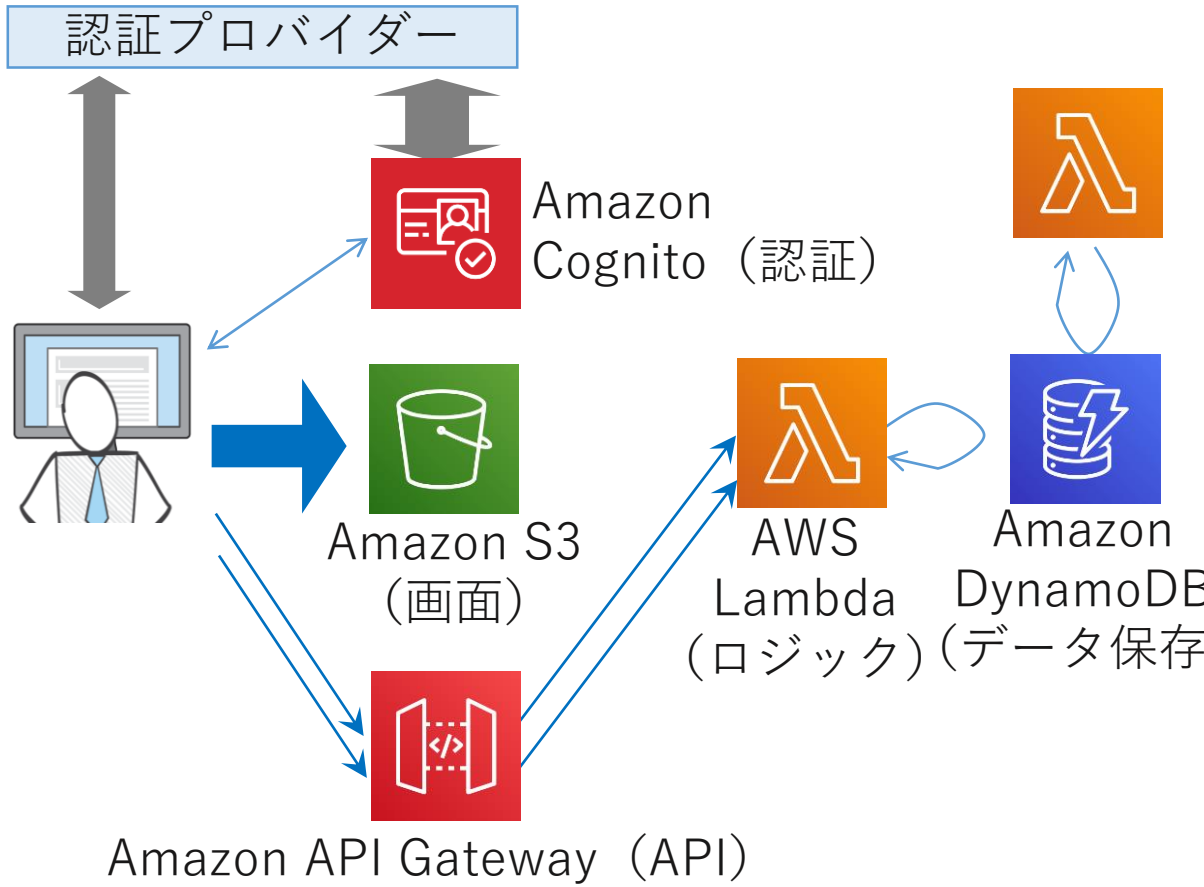
クラウド化、どこまでを目指せばよいのか

EC2ベースのアーキテクチャ



まずは、VPCの中にある基本サービス活用
既存システムをリフトする
（最初はここを目指す）

サーバーレスアーキテクチャ

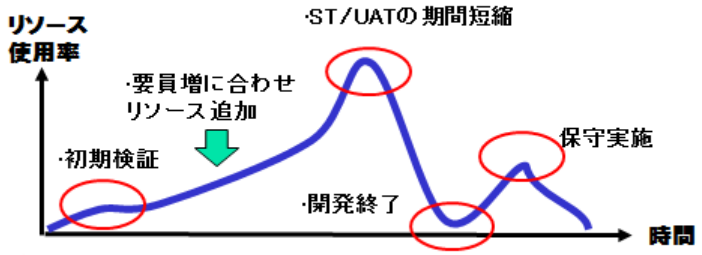


新規webアプリは積極的にサービス利活用
クラウドネイティブにシフト

とはいえ、
コスト削減も利益率向上には
やっぱり効果的です

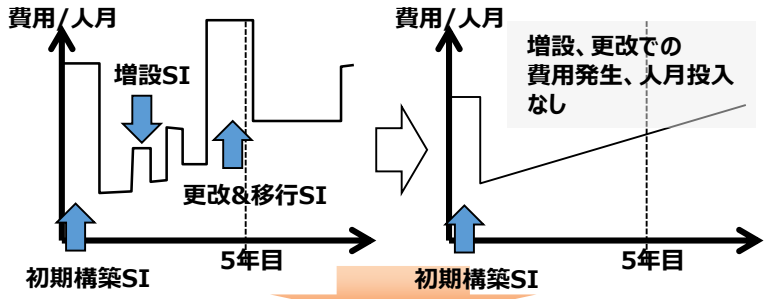
クラウドでコストを削減しやすいシステム

1．開発環境・検証環境での利用



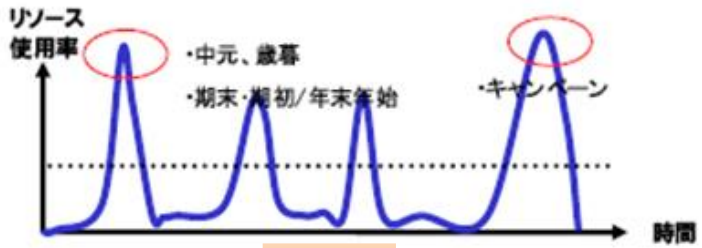
実績(社内)：40%コストダウン
開発環境立上：2w⇒1d

2．ライフサイクル対応からの脱却



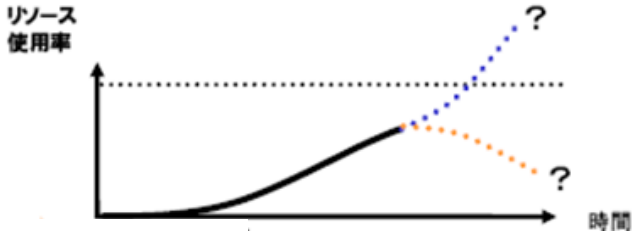
D社実績：5年TCO 15%削減(SI費用－α)
D社見込：10年TCO 40%ダウン

3．定期変動型業務での利用



L社実績：バッチのみ増強：20%削減
N社実績：ｷｬﾝﾍﾟｰﾝのみ増強：40%削減

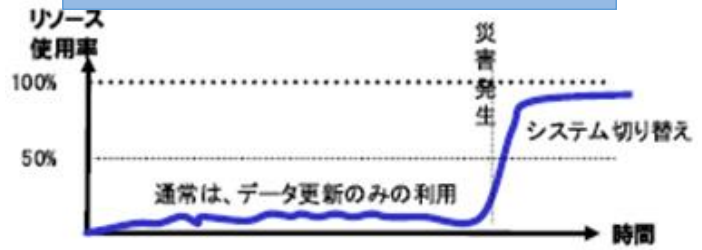
4．新規ビジネスの立上げや業務統合での利用統合延命



N社実績：業務統合計画延期にも追加費用発生せず
C社実績M&A一時費用を70%削減

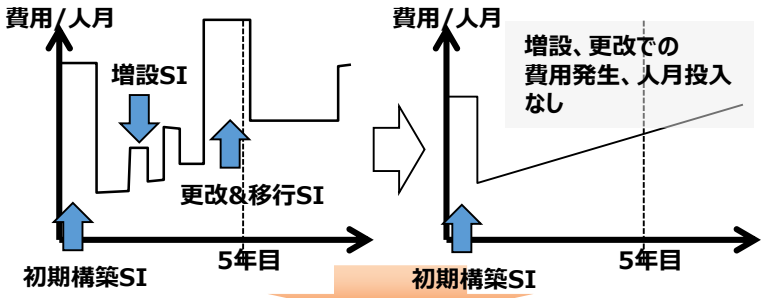
クラウドでコストを削減しやすいシステム

5．災対環境での利用



D社実績：70%削減
M社：30%削減+対応システム増

6．アーカイブ利用



F社(AWS)：50%削減

弊社
“秘伝のタレ”のご紹介

Well-Architected　Ready

SCSK クラウドリファレンスキット for AWS

ノウハウ

多種多様なAWS構築案件を
手がけてきたSCSKのノウハウを
集約しています

自動化

自動化ツールにより、
設計構築作業の時間や
品質のブレを回避できます

最新

SCSKが実案件で得たナレッジや、
AWSのサービス更新を継続的に
反映しています

ITガバナンス
AWSノウハウ

工期短縮
安定品質

運用自動化
設定ミス削減

Well-Architected　Ready

AWS利用時の、社内の課題を解決！

Before

- **属人化**
  AWSは特定の人頼み。もし異動になったら?
  でもイチから社員教育する時間も無い･･･。
- **ノウハウ不足**
  社内に詳しい人がおらず、
  なかなかAWSの機能を活かせない･･･。
- **シャドウIT**
  各部署での利用ルール作りに手が回らず、
  対策できていない･･･。
- **時間が掛かる**
  毎回イチから構築や運用設定をしているので、
  準備に時間が掛かってしまう･･･。
- **品質が一定しない**
  設計時の考慮不足で、予期せぬシステムダウンや
  セキュリティ設定不備が発生してしまった･･･。

After

- **AWSノウハウ共有**
  ドキュメントによりイチから覚えなくても済み、
  引継ぎも楽々に!
  詳しい人がいなくても、AWSの機能活用や
  サービス更新に追従できる!
- **ガバナンス整備**
  クラウド利用の社内ITガバナンスが整えられ、
  利用が活性化できる!
- **工期短縮**
  インフラ構築をイチから実施せずに、
  構築や運用設定の時間を短縮できる!
- **構築、運用設定品質の一定化、設定ミス削減**
  設計考慮不足を防ぎ、システムダウンや
  セキュリティリスクを軽減!

中身を少しだけ： ガイドラインサンプル

3. システム構成

AWS 上に構築するシステムは、3 つシステム構成(デザインパターン)から選択して構成することとする。

3.1 デザインパターン

システムの負荷や可用性に対する想定サービスレベルに応じ、3 通りのクラス（デザインパターン）を設ける。

表 2 デザインパターン

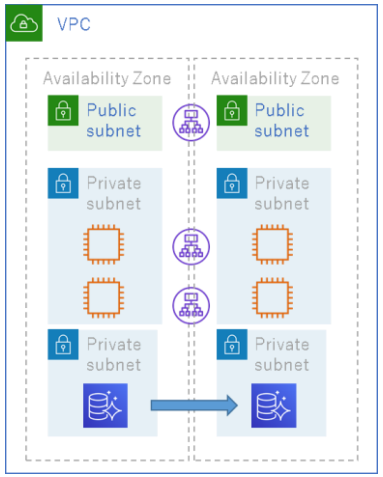
項目		Platinum クラス	Gold クラス	Silver クラス
対象システム想定		コンシューマ向けシステム トラフィック（負荷）の予測が困難であり、高い可用性が求められるシステム等	社内向けシステム トラフィック（負荷）の予測が可能であり、高い可用性が求められるシステム等	開発・検証システム トラフィック（負荷）が少なく、メンテナンスや障害等による多少の停止が許容されるシステム等
稼働率目標		99.95%＋α 月間停止時間: 約 20 分	99.95% 月間停止時間: 約 20 分	99% 月間停止時間: 約 7 時間
サービス提供時間	サービス提供時間	24 時間 365 日	24 時間 365 日	24 時間 365 日
	定期メンテナンス	システムごとに調整	システムごとに調整	システムごとに選択
障害時影響	単一障害時停止許容時間	15 分	15 分	60 分
	単一障害時性能	100%	100%	50%以上
バックアップリストア	リストア時間	1-2 時間 サービス再開: 6 時間	1-2 時間 サービス再開: 6 時間	1 営業日
	リストア時リカバリポイント	EC2:1440 分 DB(RDS):直前 (5 分前)	EC2:1440 分 DB(RDS):直前 (5 分前)	EC2:1440 分 DB(RDS):直前 (5 分前)
	バックアップ保管世代	7 世代	7 世代	7 世代

ガイドライン クラウドツールキット

3.1.2 Gold クラス

Gold クラスは、トラフィック（負荷）の予測ができ高い可用性が求められる、社内基幹システムなどを想定したデザインパターンである。
システムを構成する各コンポーネントは完全冗長化し、可用性を確保する。

表 4 Gold クラス

構成イメージ	
構成説明	● Web/AP サーバ側は ELB を利用し Availability Zone を分け、Active-Active の冗長構成とする。 ● アプリケーション、ミドルウェアは Active-Active の冗長構成に対応している必要がある。 ● Web/AP サーバはセッションステートレスな構成にし、単一障害時に自動的に別のサーバでセッションを引き継げるようにする。 ● サーバはハードウェア障害時を考慮し EC2 Auto Recovery による自動復旧設定の併用を推奨する。 ● DB は耐障害性を考慮し、RDS の Multi-AZ 構成とする。

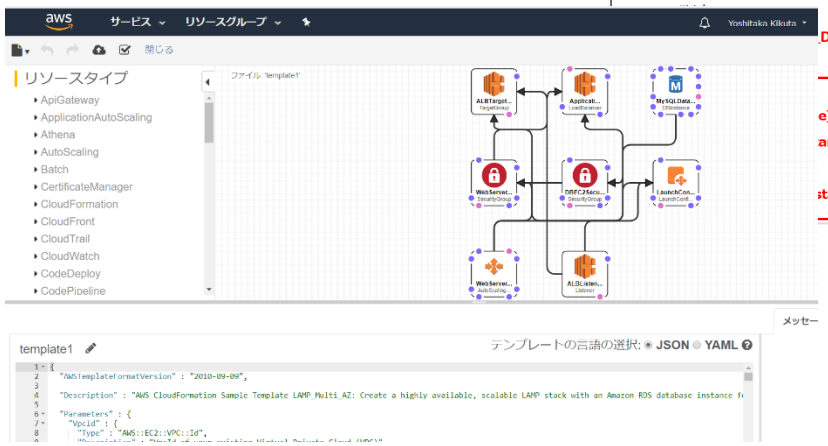
非機能部分とAWS 独自部分を網羅的に記載。
AWSを社内に展開する際のルール作りに
掛かる時間を短縮し、検討に抜け、漏れなく。

中身を少しだけ:　構築自動化ツール

ガイドラインと紐づいたテンプレート取り扱い説明書


```
#M4,C4 は有効にする
#EbsOptimized: true
IamInstanceProfile: ${IAM_Role_Name}
ImageId: ${AMI_Id}
InstanceType: ${Instance_Type}
KeyName: ${Key_Pair}
Monitoring: true
#SG_template.yml で Output 定義したセキュリティグループ ID を指定
SecurityGroupIds: [!ImportValue Out${SG_Name}]
#VPC_xx_template.yml で Output 定義したサブネット ID を指定
SubnetId: !ImportValue Out${SUB_Name}
Tags:
  - Key: Name
    Value: ${EC2_NameTag}
  - Key: Environment
    Value: ${EC2_EnvTag}
  - Key: System
    Value: ${EC2_SysTag}
Tenancy: default
#追加ボリュームがある場合記載(例 Device /dev/xvdf, vol-12345678)
DevName}, VolumeId: ${EC2_VolName}]
e}:
anceName}
tanceName}
```

スクリプトの編集が終わったら、AWS CloudFormationにファイルをアップロードして実行するだけで環境構築完了！



スクリプト本体と、ユーザによる変更パラメータの説明をわかりやすく記載

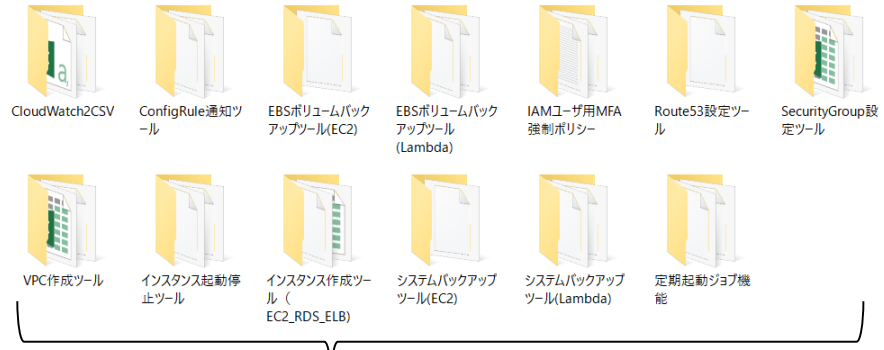
No.	変数	説明	例
1	EC2_InstanceName	EC2 インスタンス名	EC2A1TSTSRV01
2	EC2_AZ	アベイラビリティーゾーン	ap-northest-1a
3	IAM_Role_Name	IAM ロール	別途作成した IAM ロール名
4	AMI_Id	AMI-ID	ami-12345678
5	Instance_Type	インスタンスタイプ	c4.large
6	Key_Pair	キーペア	別途作成したキーペア
7	**SG_Name**	セキュリティグループ ID	OutSGRA3TSTWEB01 ※項番 2.1.5 で設定したセキュリティグループ
8	**SUB_Name**	サブネットグループ ID	OutSGRA3TSTWEB01 ※項番 2.1.5 で設定したセキュリティグループ
9	EC2_NameTag	ネームタグ	EC2-A1TST-SRV01
10	EC2_EnvTag	環境名	A1
11	EC2_SysTag	システム名	TST
12	EC2_DevName	デバイス名	/dev/xvdf
13	EC2_VolName	ボリューム名	vol-12345678

表 2-21 EC2 インスタンス変数表

DescribeStacks		
OutputKey	OutputValue (例)	内容
OutEC2A1TSTSRV01	i-029d325db286axxxx	EC2 インスタンス ID

表 2-22 アウトプット出力例_EC2

中身を少しだけ: 運用自動化ツール



運用自動化ツール
（Pythonスクリプト）

AWS
Lambdaを利用し、
サーバレスで
運用スクリプトを実行

Amazon EC2の
起動/停止

EBSボリュームの
バックアップ

Amazon
CloudWatchの
統計情報取得

等々・・・

実態はPythonスクリプトそのもの

```
# -*- coding: utf-8 -*-
import boto3
import time
import subprocess
import yaml
import urllib2
import sys
from datetime import datetime
import warnings
warnings.filterwarnings("ignore", category=UnicodeWarning)

# Instance情報取得
def get_target_instance_infos(backup_tag, client):
    rt = []
    vols = []
    try:
        instance_id = urllib2.urlopen('http://169.254.169.254/latest/meta-data/instance-id').read()
        res = client.describe_instances(
```

パブリッククラウドモデル(AWS) MSP

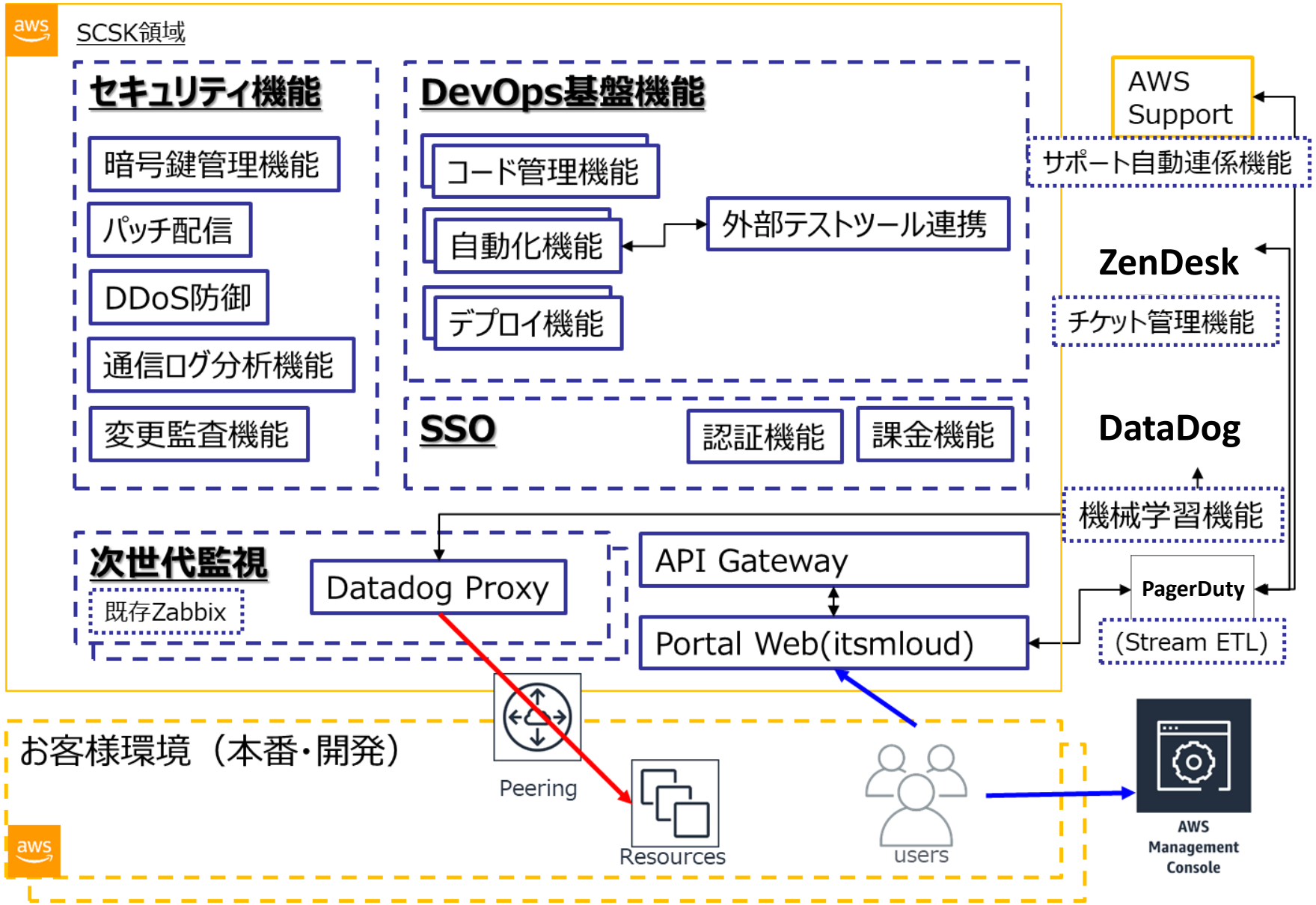
※Managed Service Provider
※2019年3Qサービス開始予定

エンタープライズ基盤に
必要な機能をワンストップで提供

- インスタンスの監視
- SLA & インシデント管理
- コンプライアンス対応
- セキュリティオートメーション
- CI/CD環境の提供
- 課金管理

Well-Architected　Ready

MSPを支えるサービスアーキテクチャ: 自らDevOpsを実践



進化し続けるサービス

全てIaCの考えで実装。
ソフトウェアエンジニアリングの手法で継続的に改善

データ連携は疎結合。
他ツールとの連携も容易に。

基盤運用をより簡単に

障害1次対応の完全自動化し、お客様負担を軽減。
失敗時は自動音声コール実施。

サポート対応迅速化のため自動AWSエスカレーション実装

セキュリティオートメーション実装。
TrendMicro社製品と連携しリスクを自動修正。

ITの、つぎの、幸せへ。

SCSK

May the Cloud be with you. :)