


クラウドテクノロジーの最新企業活用

～マルチクラウドの統合と品質メトリクスの管理～

July 2019

Katsushi Yamashita, IBM Distinguished Engineer

技術理事、IBMアカデミー会員

- 
- A dark blue horizontal bar is located in the top left corner of the slide.
1. 企業システムにおけるマルチクラウドの統合
 2. サービス品質のメトリクス管理

デジタルイノベーション事業部 インターネットのパワーを事業に

- 迅速な開発
- 既存のクラウド活用
- LEAN Start up

<使いたい>

オフィスアプリケーション改革 働き方改革~いつでもどこでも

- Microsoft Office 365 Drive
- Google Drive/Dropbox/Box
- Slack/Facebook B
- IBM Verse
- VCube/WebEX/Zoom



インフラの最適化 企業はインターネットに対応できるか

- クラウドを前提にしたアクセスメソッド
- エンタープライズID管理
- セキュリティポリシーの対応

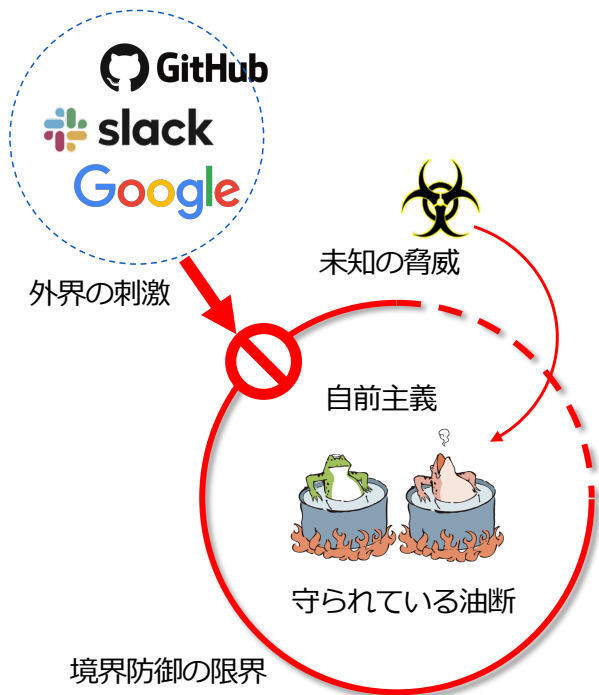
<管理できない>

運用管理の課題 IT部門のデジタルトランスフォーメーション

- 事業部のだれが使っているか？
- インシデントが管理できない！
- バックアップなし運用!?
- コスト管理（性能管理）
- IT投資効率（効果測定）
- セキュリティリスク
- コンプライアンスと法令対応

インターネットで共有されるサービスとのインテグレーション
(オープンイノベーション) H.チェスブロウ

デジタルデマンド

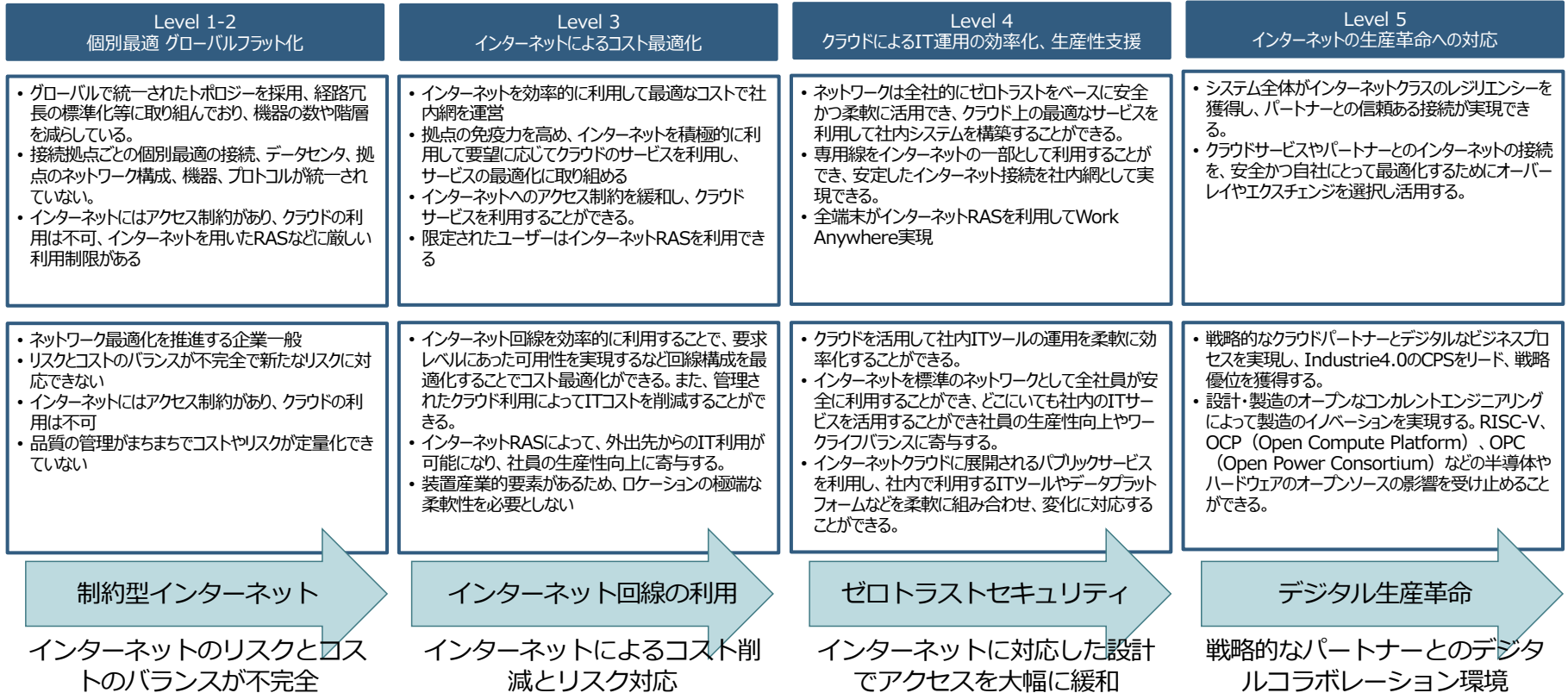


オープンイノベーションへのロードマップ

1. インターネットに無制限にアクセスできる。
 - モバイルアクセス
 - PCコンプライアンス
 - 常時監視と記録の検査
 - 登録と管理
2. 最新のITツールの活用
 - インターネットサービスの利用
 - ID統合とフェデレーション (登録と管理)
 - ITツールの最適化
3. インターネットの生産革命
 - グローバルコラボレーション
 - デジタルデマンドのループ
 - オープン・コンカレントエンジニアリング
 - Industrie4.0やIndustry Internet



オープン・インターネットへのロードマップ例



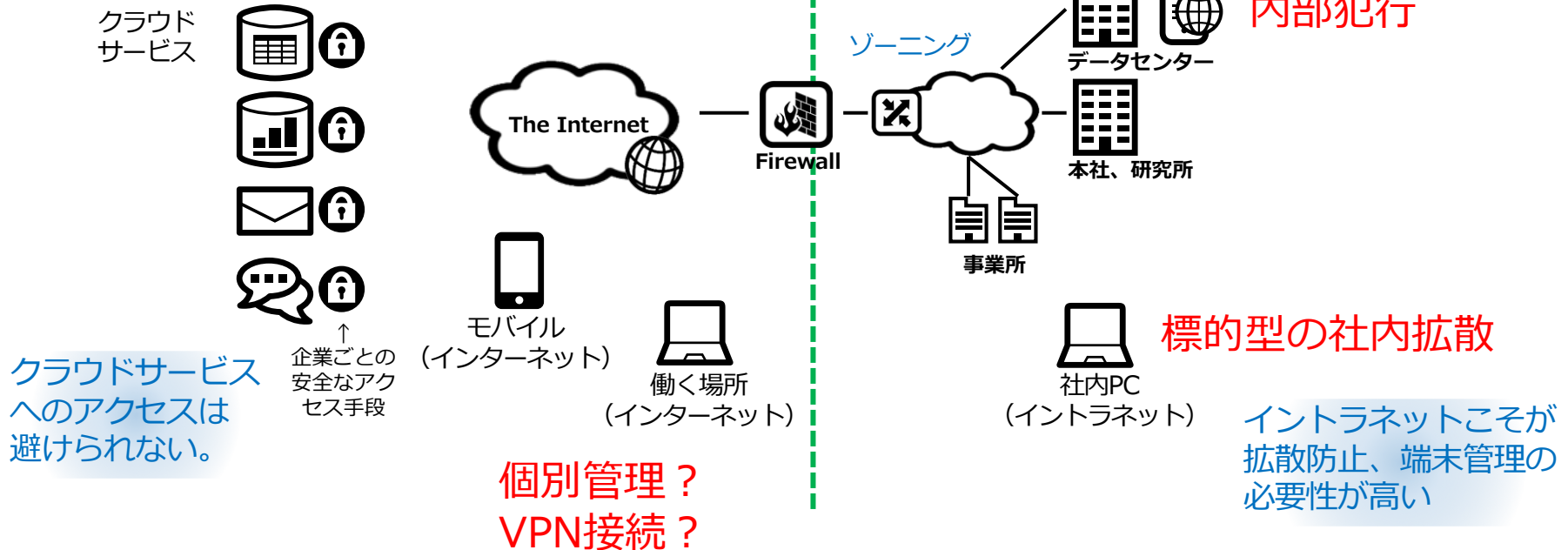
クラウド環境を使う時に起こる諸問題

アクセス環境、データの移動と保管、ユーザーID、コスト管理…



エンタープライズITがインターネットを前提としているか？
(オープン、イノベーション、セキュリティ)

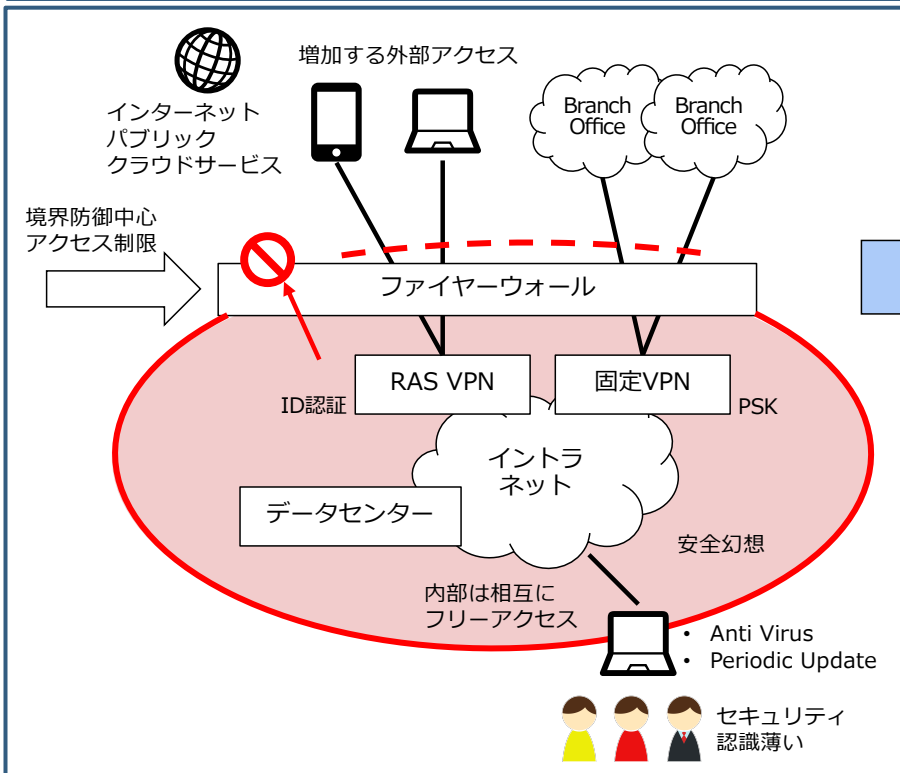
- これまでのネットワーク
 - ゾーニングを中心としたネットワークセキュリティの限界



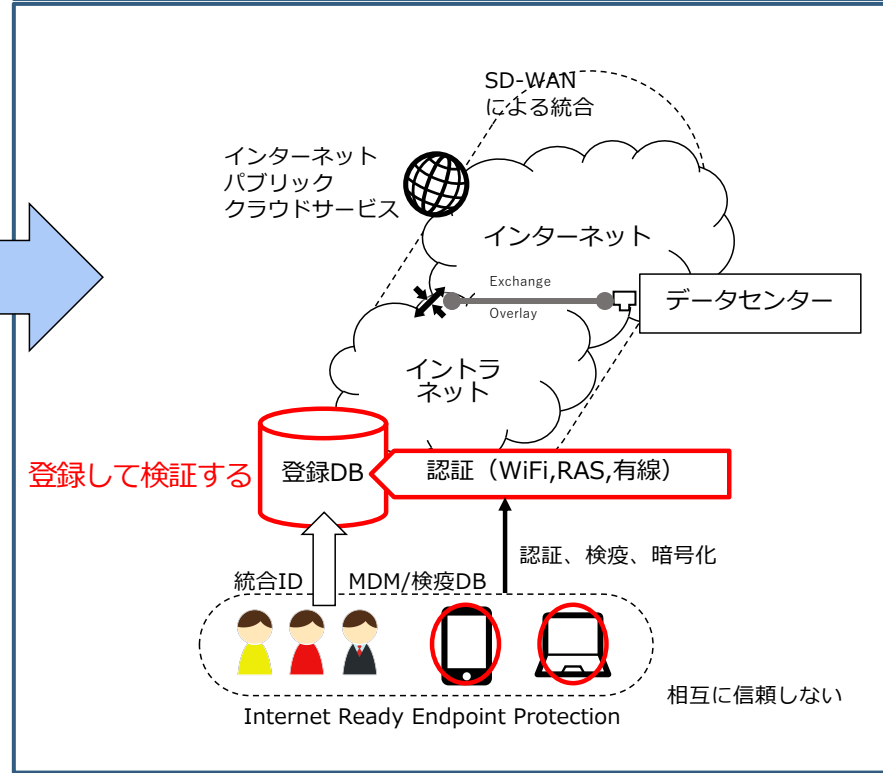
管理視点の変化が求められている



境界を防衛、内部を信用してしまう



信頼せず登録して検証する



- パロアルト「**ゼロトラスト・ネットワーク・セキュリティ**」
<http://www.atmarkit.co.jp/ait/articles/1405/23/news081.html>
 - 社内・社外ネットワークを区別せずに、「必要な情報だけを知ること」を実現
 - 検証して、信頼しない→検査してログを取る
- Google「**ビヨンド・コープ**」 <https://cloud.google.com/beyondcorp/>
 - 端末の識別→端末インベントリと識別
 - ユーザーの厳密な識別→ユーザーとロールのDBとシングルサインオン
 - 無制限型のネットワークとネットワーク認証
 - アプリケーションとワークフローの外部への公開（アクセスプロキシとDNS）
 - 推定信頼度に基づくインベントリへのアクセスコントロール
- IBM「**サイバー・レジリエンシー（回復性）**」 <https://www.ibm.com/blogs/think/2017/05/resilience/>
 - ネットワークのシンプル化～ゼロトラスト・イントラネット
 - セキュリティ・インテリジェンス（時系列ログの相関）
 - コグニティブ・セキュリティ（推論ベースの分析、サイバーレピュテーションの信頼度推定）



どんなに予防し防御しても、
侵入は防げない
入ってきてから、やっつける



ZERO⁰トラスト・セキュリティ

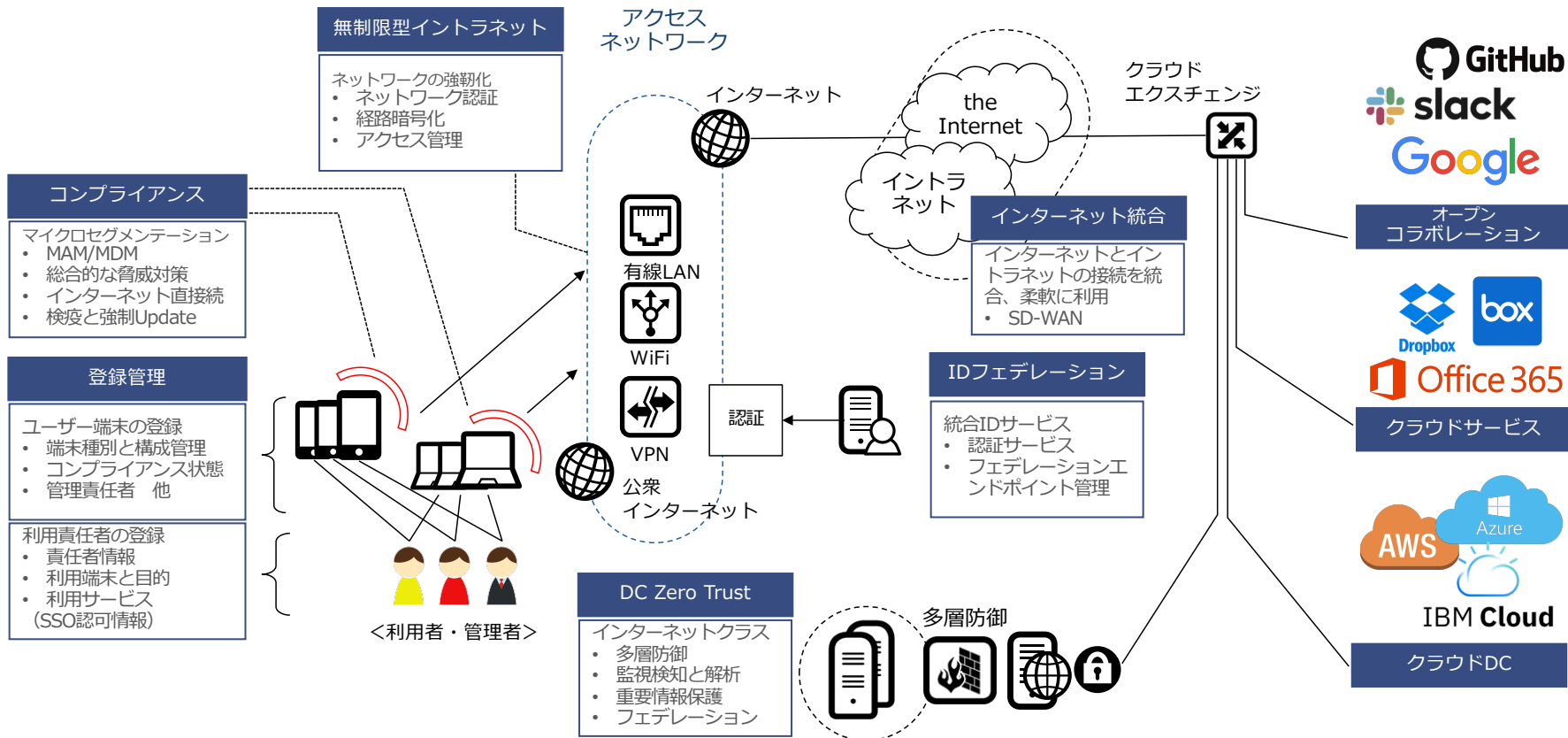
3 + 1

多層防御・監視検知・セキュリティインテリジェンス + 重要情報の保護

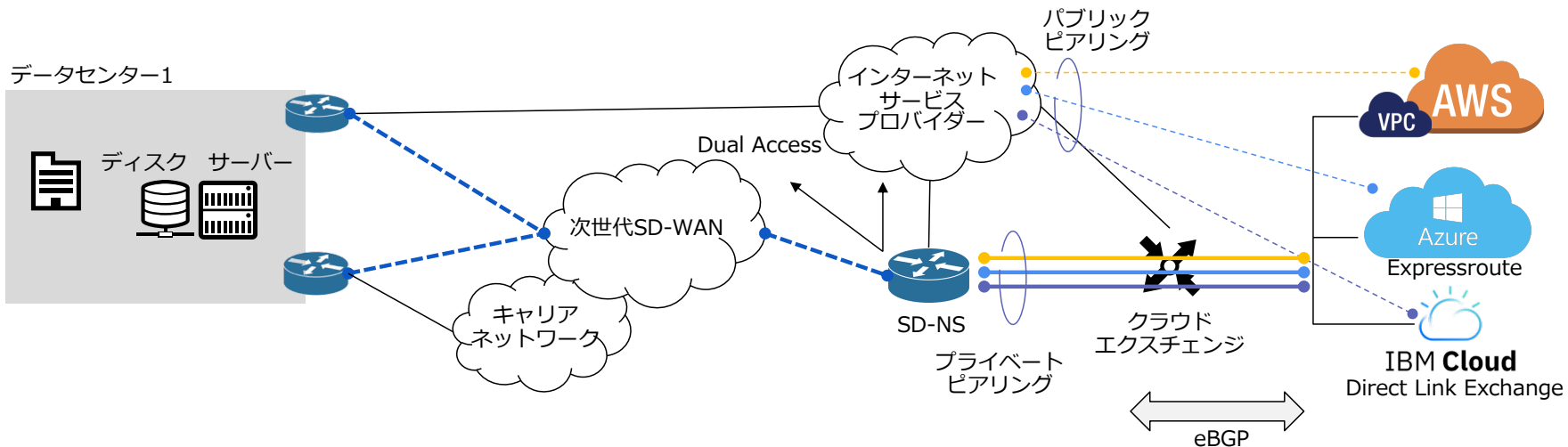
検証して、**信頼しないことを前提**とする

パロアルトが性悪説の「ゼロトラスト・ネットワーク・セキュリティ」を解説
<http://www.atmarket.co.jp/ait/articles/1405/23/news081.html>

登録して管理すること、信頼せずに検証するための情報基盤



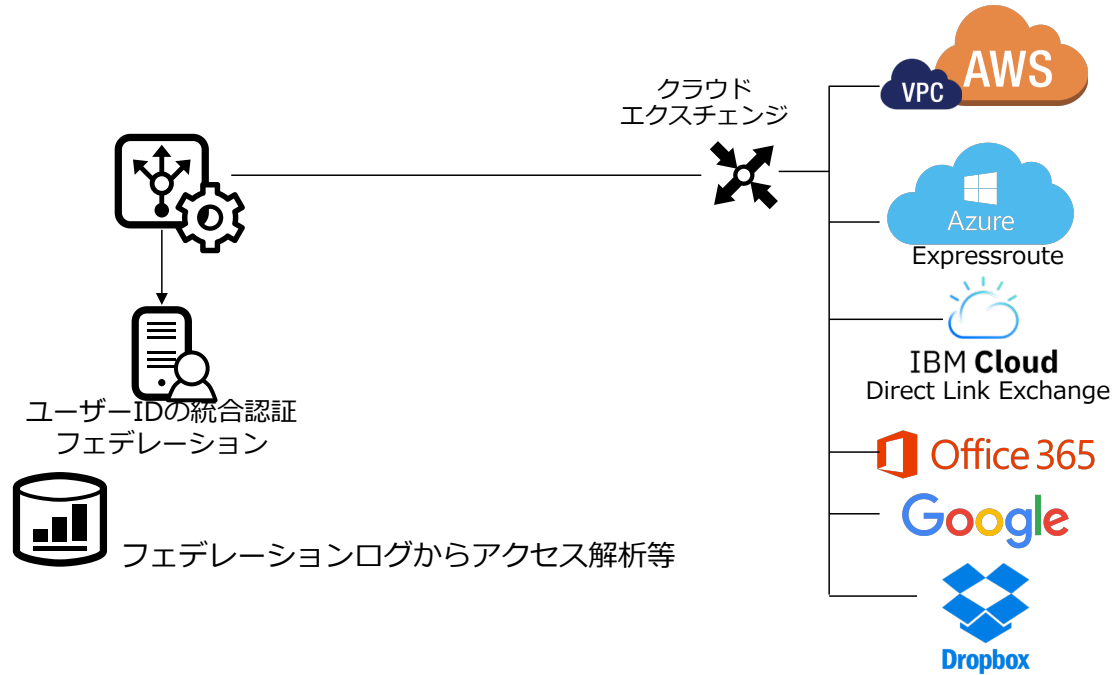
クラウドエクステンジによるインターネット統合



- Express Route等を終端するためのeBGPルーターはインターネットCPEでも良い。
- クラウドエクステンジでの接続も可
- インターネット接続管理、広帯域接続の活用など

IDフェデレーション

クラウドサービスのユーザーIDをコントロールする



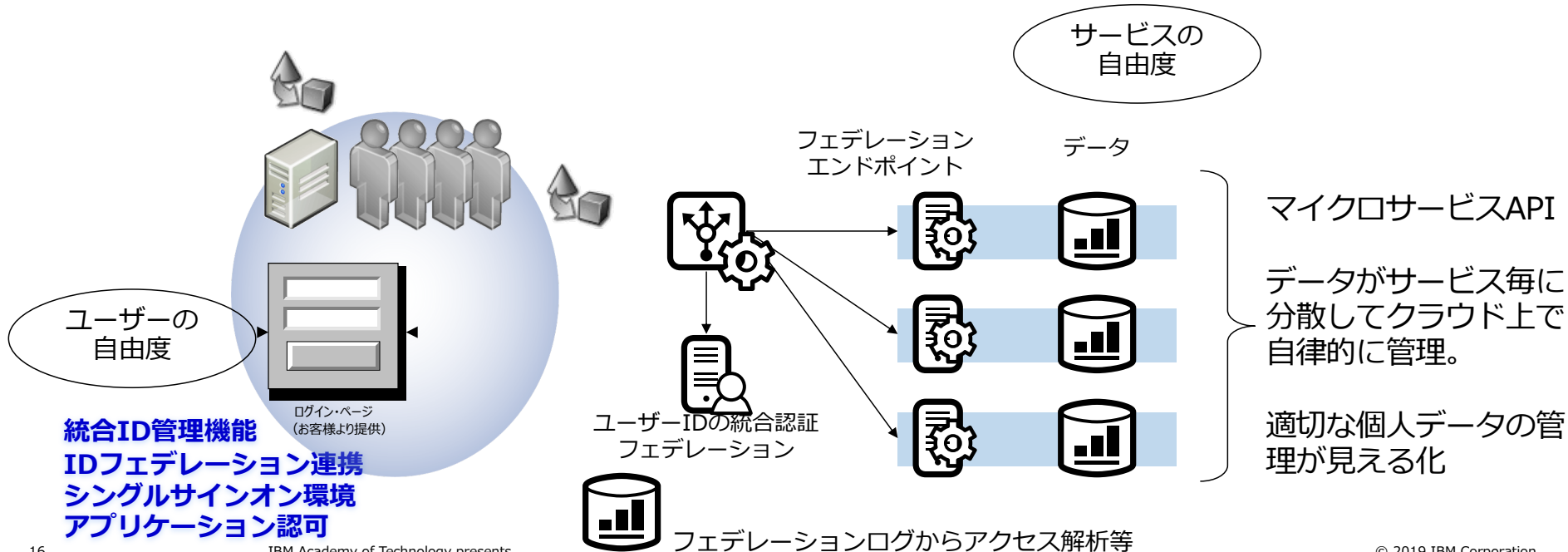
IDフェデレーションを中心とした個人データの管理



マルチクラウドへのアクセスポータル

- 外部連携IDの要求、アクセス許可
- 汎用サービスの提供（データバックアップ等）
- リスクアクセスポータル（Continuous Compliance等）

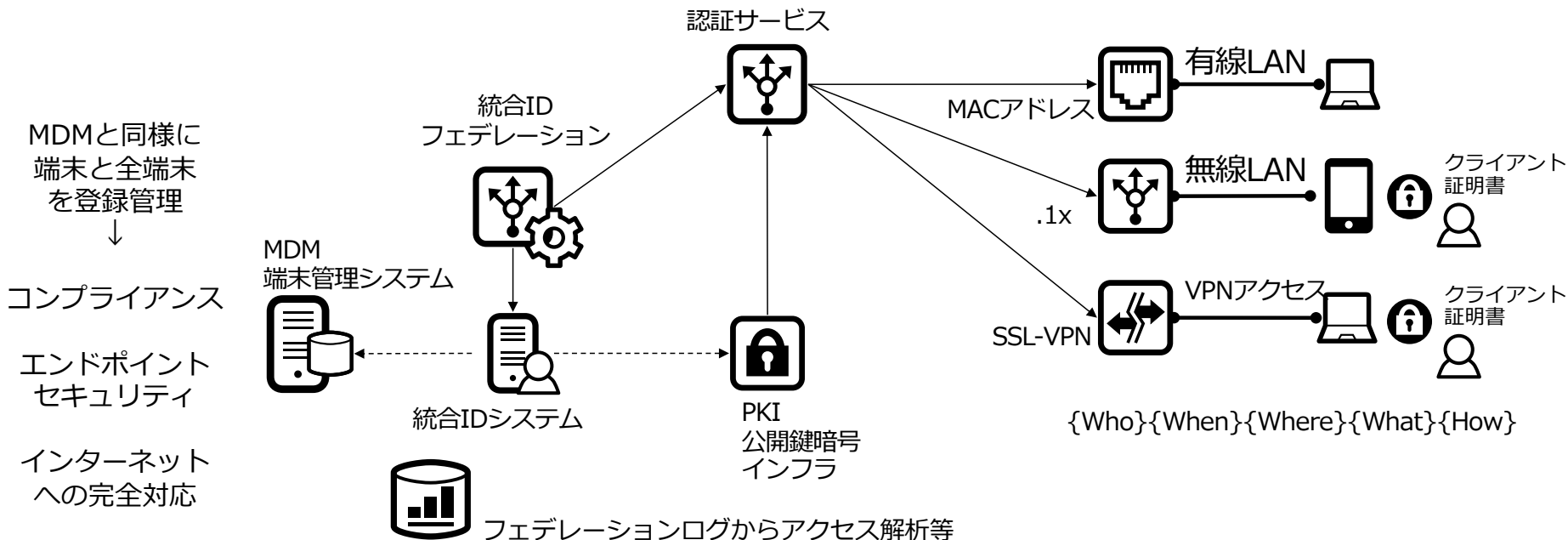
エストニア
国民ID



無制限型イントラネット



インターネットであるかどうかには関わらない認証ログの取得



有線LANがもっとも危険


- 挿せる
- 暗号化されていない
- 信頼してしまう
- 数が多く管理できない

無線LAN インターネットVPN

- 認証
 - 暗号化
 - しない
 - 管理プラットフォーム
- 認証
 - 暗号化
 - しない

ZERO トラストセキュリティ ネットワーク・ソリューション



- 
- A dark blue horizontal bar is located in the top left corner of the slide.
1. 企業システムにおけるマルチクラウドの統合
 2. サービス品質のメトリクス管理

これまでのWEBアプリは機能設計中心
(商品検索、受注決済処理、ポイントカード..)

「ドリルを買いに来た人はドリルが欲しいの
ではなく穴が欲しい。」



ユーザーのコンテキストを理解する。

<機能面の課題>

SPA (Single Page Application) Architecture



UI/UX Contents

サブページを含んだ
画面の枠組み
(コンテンツはない)

キャッシュからロード

アドレス
Cookie
地域
...その他



SPAはフロントエンドの
プログラム実行能力により
これまでにない
業務の継続性を提供します



アドテク

他サイトのAPIコール

ログイン

サブページ
ごとのコンテンツ

サービスの並列化

Micro-nano
サービス
インスタンス

データ
レイク



独立したマイクロサービス
→いつでも入れ換える

HTML5/JSの強力なデータ収集の分散処理



Googleが提供する標準の分析プログラム Analytics.js

Applications



Browser

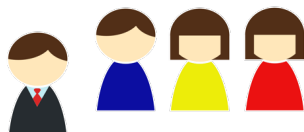


Social Network 2081 108 111 8

System Metrics



Profiling



なにが表示された？

ユーザーのアクション？

SNSへの連携？

ネットワークや表示の速度

ユーザー分析

Loading

HTML Page
Virtual Page View
X-site view
Historical data

User Event

Download
Click on AD
Gadget, Flash, Ajax
Play movie, motion GIF

Social Event

Share, Like on Facebook
Tweet on Twitter
Bookmark on hatena
Share on Google+
X-platform analytics

Speed Metrics

Loading Speed
Ajax request response
Custom time measurement

X-site User activity

Cookie
User ID
Historical Data



ユーザーの姿をデジタルで描き出すフロントエンド
データは戦略的な設計によって作り出す

(計測して改善するLeanスタイル)

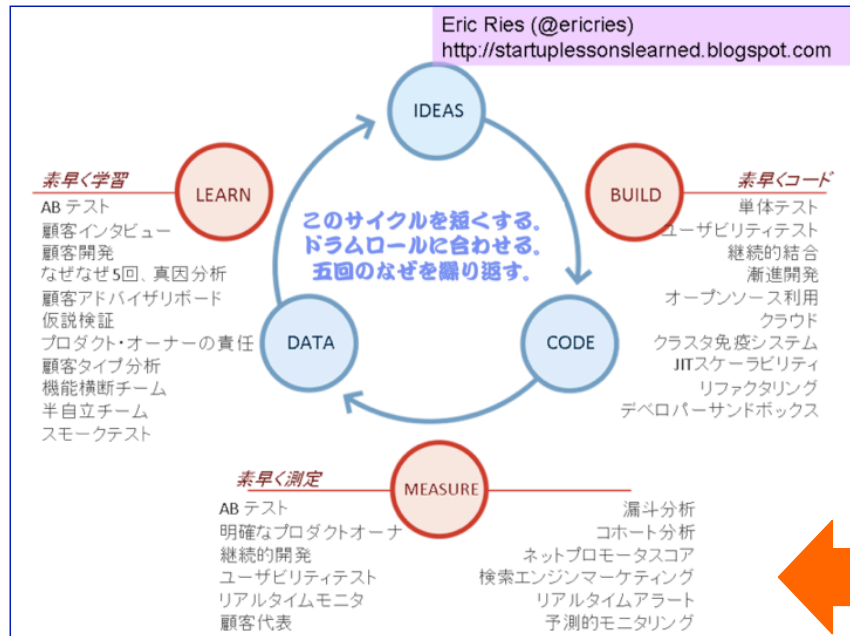


**バックエンドアーキテクチャ
は変化への対応のため**

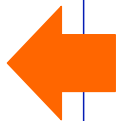
目指しているのは、クラウドの自動化がITにもたらしたものの



- クラウドが前提の新しいシステム構築手法はLEAN(TOYOTA WAY)に基づいている
- 学習と実装のサイクルを高速化するために、ドラムロールと改善活動を取り入れている



- クラウドのシステムは構築してそのまま運用するこれまでのスタイルとは異なり、度重なるリリースによって改善するシステム。
- ドラムロールによって開発と運用システムが同時にシステムリリースを行い、新しい課題に対して真因を追求して解決する手法。
- Facebookは創業当時月に20回のリリースを行っていたが、リリース改善を行うことにより月間660回のリリースに成功。



サービス品質の
計測と学習



SOAはマイクロサービスのアンチ・パターンか



上流設計

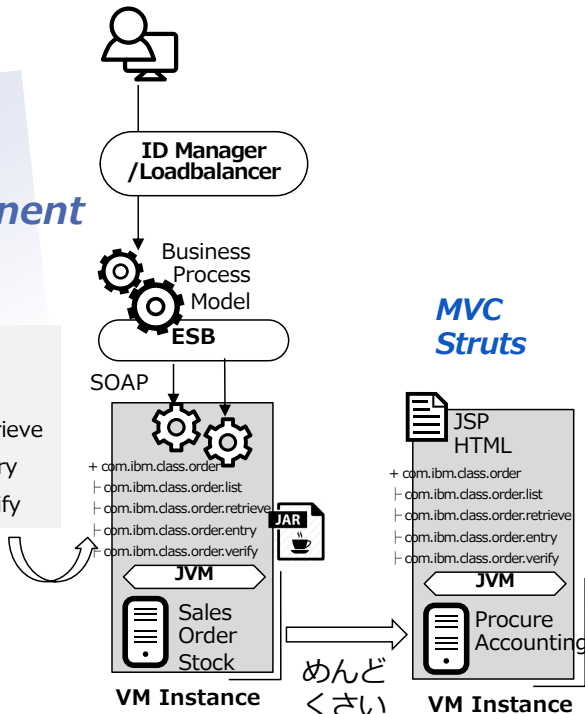
→理想的な構成

→実際の構成

Business OBJECT Modeling

Business Component

- + com.ibm.class.order
- └ com.ibm.class.order.list
- └ com.ibm.class.order.retrieve
- └ com.ibm.class.order.entry
- └ com.ibm.class.order.verify



SPAにしてくれ!



API Management

REST JSON API



また、やっちゃった

悲鳴
オープンソースは
無敵じゃないよ〜

マイクロサービスの魔法で StrutsをAngularでSPAにしてくれて言っても JSPがSQLだらけで

無理ゲー

マイクロサービスは万能薬ではない——マイクロサービスの生々しい話を聞いてきた

白石 俊平 (HTML5 Experts.jp編集長) マイクロサービス2017年9月19日 <https://html5experts.jp/shumpei-shiraishi/24247/>

Starlight & Storm 代表社員の長谷川裕一がHTML5カンファレンス2017にて語る。

ビューの部分がStrutsで書かれた業務システムを、Angularなどを使ってSPA (Single Page Application)化したいというご要望もかなり多い。「業務アプリを刷新するなら、マイクロサービスもSPAもやっておこう」みたいなノリ。ただ、Struts (※)で書かれたものをSPAに…ってのも、実はすごく。ビューがちゃんと層として分かれているシステムなら楽ですが、そうっていないシステムも多い。タープライズ・アプリケーション・アーキテクチャ・パターンに沿っているようなシステムならいいけど、そんなのはごく少数。ひどいものになると、JSP (※)の中に思い切りSQL書いてあったりしますから。そういうシステムをSPA化するとなると、結局全面的に作り直しが必要になっちゃいますよね。合わせてマイクロサービスも…なんてやってたら、途方もないコストが必要になっちゃう。

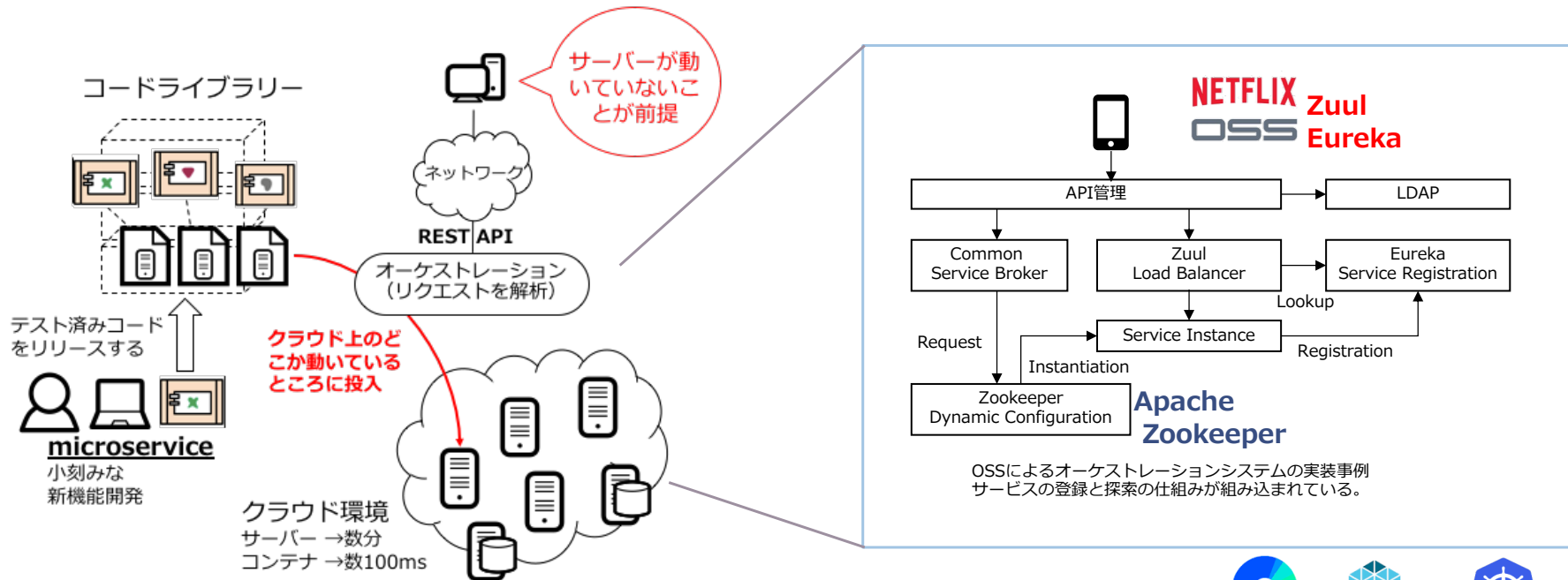
※Struts…2000年代初頭に登場し、一世を風靡したJavaのWebアプリケーションフレームワーク

※JSP…Java Server Pagesの略。サーバサイドJavaで使われる標準的なテンプレート機能

コードベース・インフラストラクチャの実装事例



JVMと違い**アプリケーションは動いてないことを前提**にリクエストに応じて環境をオーケストレーション



今では→



MARATHON



MESOS



kubernetes

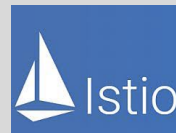
よくみると、ソフトウェアの機能を作っていない。
リリースエンジニアリングをソフトウェア化するために開発されたOSSたち
インターネットでスケールすることが目的のアプリケーション



warden

CLLOUDFOUNDRY
BOSH

Amalgam8



CLLOUDFOUNDRY



kafka



CHEF



openstack.



docker

kubernetes

operator



MESOS



APACHE
ZooKeeper™



MARATHON



Long-run batch instances



etcd

パフォーマンスの計測

- 成熟度ではなく**ケイパビリティ**
- 出力(Output)ではなく**成果(Outcome)**

従来は

- コード生産性
- ベロシティ
- 稼働率

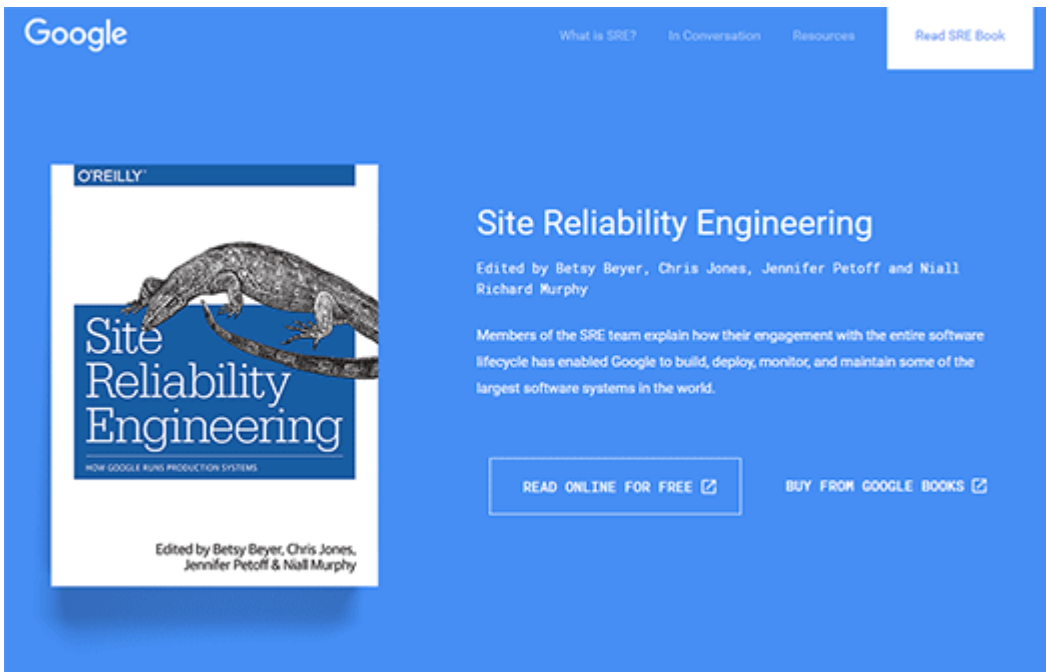


訪れる変化

- リードタイム
- デプロイの頻度
- 平均修復時間(MTTR)
- 変更失敗率

インフラや運用をソフトウェアで改善していく新しいアプローチ

- 「**Site Reliability Engineering**」 (SRE) とは、GoogleのシニアVPであるBen Treynor氏が提唱した、高い信頼性や性能を発揮するシステムインフラを実現し、改善していくアプローチのひとつです。



The image shows a screenshot of the Google Site Reliability Engineering (SRE) book page. The page has a blue header with the Google logo on the left and navigation links: 'What is SRE?', 'In Conversation', 'Resources', and 'Read SRE Book'. Below the header is a large blue area containing the book cover on the left and text on the right. The book cover features the title 'Site Reliability Engineering' in white on a blue background, with a lizard illustration. Below the cover, it says 'Edited by Betsy Beyer, Chris Jones, Jennifer Petoff & Niall Murphy'. To the right of the cover, the title 'Site Reliability Engineering' is repeated in white. Below the title, it says 'Edited by Betsy Beyer, Chris Jones, Jennifer Petoff and Niall Richard Murphy'. Further down, a paragraph describes the book's content: 'Members of the SRE team explain how their engagement with the entire software lifecycle has enabled Google to build, deploy, monitor, and maintain some of the largest software systems in the world.' At the bottom of this section are two buttons: 'READ ONLINE FOR FREE' and 'BUY FROM GOOGLE BOOKS', both with external link icons.

Like traditional operations groups, we keep important, revenue-critical systems up and running despite hurricanes, bandwidth outages, and configuration errors. Unlike traditional operations groups, we view software as the primary tool through which our systems are managed, maintained, and minded;

従来の運用チームと同様に、私たちは重要かつ売り上げに直結するクリティカルなシステムを、たとえハリケーンが襲い、ネットワークが落ち、構成エラーが起きようとも、稼働状態を保つようにします。一方、従来の運用チームとは異なり、私たちはソフトウェアを主なツールとし、私たちのシステムを運用管理し、維持し、監視します

http://www.publickey1.jp/blog/17/googlesite_reliability_engineering.html

これまでのSilo型のインフラ組織では対応できない



■ 従来業務

- ITの実務上の問い合わせを受けける窓口業務
- 定型オペレーションを繰り返す定常業務
- トラブルに対応する障害対応業務
- インフラの管理業務
(構成管理やキャパシティー管理など)

■ クラウドによるインフラ運用の変化

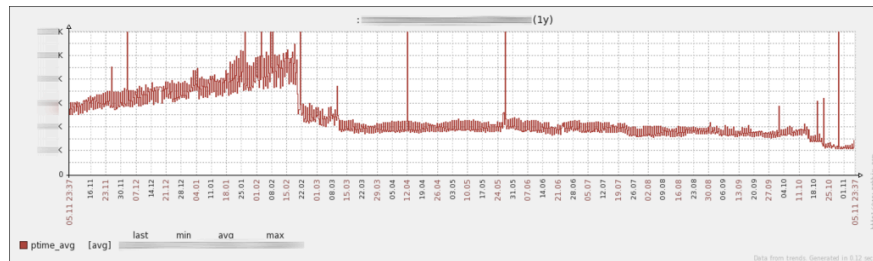
- リソース管理→KPIによるコスト管理
- 自律的モニタリングとスケーリング
- IaC(Infrastructure as Code)によるCI/CD

■ SREの視点

- 変更への即応性や信頼性の高いシステム基盤を設計
- 運用管理の自動化/自律化の仕組みを設計・構築
- 開発者が利用しやすい標準化されたポリシーやツールの整備

■ メルカリのインフラチームの挑戦

- インフラチーム改め Site Reliability Engineering チームになりました。 <Mercari Engineering Blog>
 - APIサーバ、ミドルウェアの可用性の維持・向上
 - APIサーバ、ミドルウェアのパフォーマンスの向上
 - ログ収集・分析基盤の構築、運用
 - サーバプロビジョニング・デプロイの整備
 - セキュリティの担保
 - 開発環境などの整備



- SREチームのKPI : APIサーバーの平均レスポンスタイム

<改善のループ>
リリースエンジニアリング



コントラクト→サービス要求→  Software Defined
サービスインスタンス



Deployment Plan

Service Level
Autonomous Manager



仮想マシン



Resource Scheduling

資源、資産管理→



ハードウェア

コスト管理=性能管理
(Performance Metrics)

IT投資効率=効果測定
(User Story Convergence)



IBM, IBMロゴ、ibm.comは 世界の多くの国で登録されたInternational Business Machines Corp. の商標です。他の製品名およびサービス名等は、それぞれIBMまたは各社の商標である場合があります。現時点でのIBMの商標リストについては、www.ibm.com/legal/copytrade.shtml をご覧ください。

当資料をコピー等で複製することは、日本アイ・ビー・エム株式会社および執筆者の承認なしではできません。