

システムライフサイクルをデジタル化するExastro活用術

- 攻めと守りの自動化でニューノーマルを勝ち抜く -

(2020年9月11日)



 セッション 01

攻めと守りの両面からシステム構築・運用をデジタル化する
オープンソースのソフトウェアスイート「Exastro」とは

Exastro developer

吉田 功一 Koichi Yoshida contact@exastro.jp.nec.com

世の中はクラウドネイティブなシステムを目指して切磋琢磨しています



課題はクラウドネイティブなシステムを提供できる技術者の不足です



現行システムの構築・運用を自動化して技術者を確保する必要があります

新技術を取り込みにくい

密結合で保守しづらいアプリ

従来のスタティックな環境

自動構築・自律運用

Mobile

Social

IoT

5G

Tech

スケーラブルなアプリ

コンテナ
サービスメッシュ
イミュータブルインフラストラクチャ
宣言型API

近代的でダイナミックな環境

パブリッククラウド
プライベートクラウド
ハイブリッドクラウド

自動構築・自律運用

(1)構築・運用を自動化・省力化して

AP

PF

OP

現行システムの構築・運用を自動化して技術者を確保する必要があります

新技術を取り込みにくい

密結合で保守しづらいアプリ

従来のスタティックな環境



(2)張り付いているITエンジニアを解放して

自動構築・自律運用

(1)構築・運用を自動化・省力化して

Mobile

Social

IoT

5G

Tech

スケーラブルなアプリ

コンテナ
サービスメッシュ
イミュータブルインフラストラクチャ
宣言型API

近代的でダイナミックな環境

パブリッククラウド
プライベートクラウド
ハイブリッドクラウド

自動構築・自律運用

AP

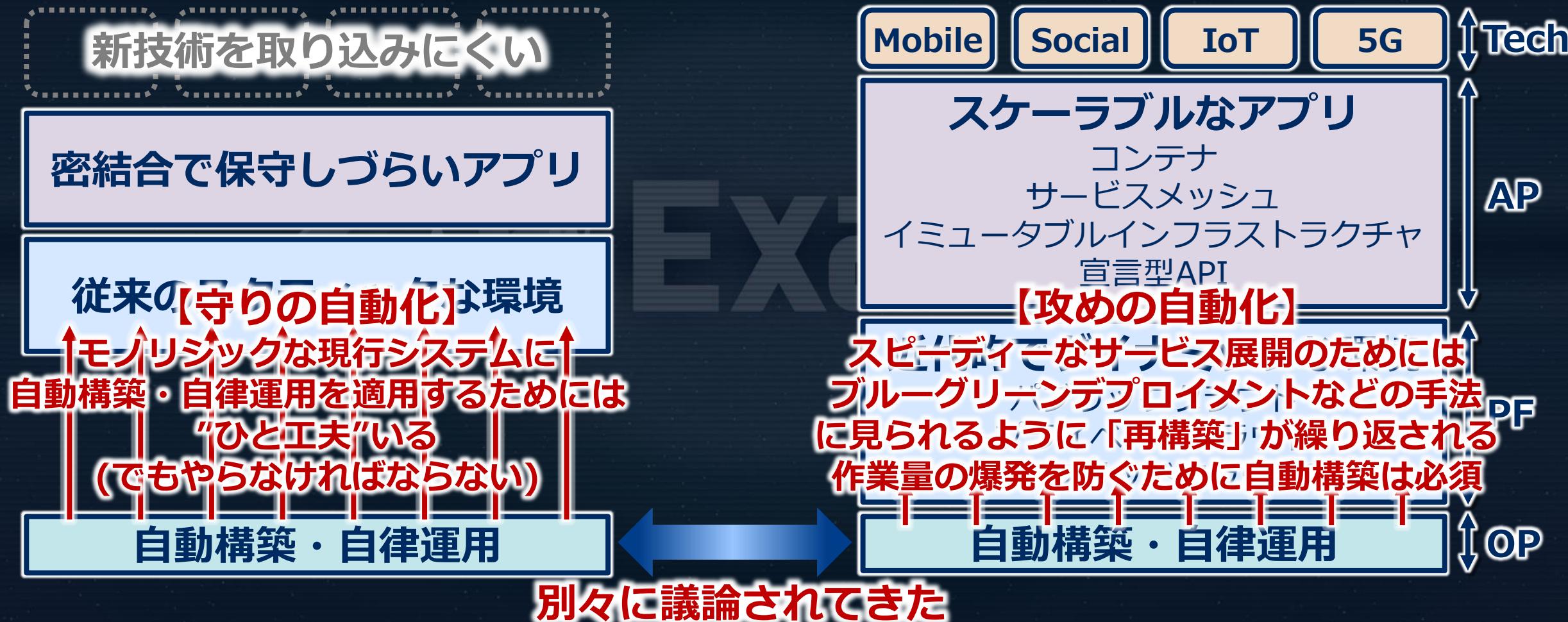
PF

OP

現行システムの構築・運用を自動化して技術者を確保する必要があります



これまで2つの領域のアプローチが別々に議論されてきました



2つの領域の自動化は財務的な視点でも目的を異にします



守りの自動化はOPEX(維持費)の効率化を目的とすることが多いです



攻めの自動化はROI(投資対効果)の向上を目的とすることが多いです



しかし2つの領域は実際にはアプローチを融合して考える必要があります



そしていま両方の領域にて不確実への迅速な対応が求められています

新技術を取り込みにくい

密結合で保守しづらいアプリ

従来のスタティックな環境

モノリシックな
システムの自動化
(守りの自動化)

Mobile

Social

IoT

5G

Tech

スケーラブルなアプリ

コンテナ
サービスメッシュ
イミュータブルインフラストラクチャ
宣言型API

近代的でダイナミックな環境

クラウドネイティブな
システムの自動化
(攻めの自動化)

そしていま両方の領域にて不確実への迅速な対応が求められています

新技術を取り込みにくい

密結合で保守しづらいアプリ

従来のスタティックな環境

モノリシックな
システムの自動化
(守りの自動化)

Mobile

Social

スケーラブル

コン
サービス
イミュータブルインフラスト
宣言型API

近代的でダイナミックな環境

クラウドネイティブな
システムの自動化
(攻めの自動化)

変動性 Volatility
不確実性 Uncertainty
複雑性 Complexity
曖昧性 Ambiguity

そしていま両方の領域にて不確実への迅速な対応が求められています

COVID-19発生

システム構築・運用の現場が直面した危機

従来の静态な環境

モノリシックな
システムの自動化
(守りの自動化)

Social

変動性 Volatility

不確実性 Uncertainty

複雑性 Complexity

曖昧性 Ambiguity

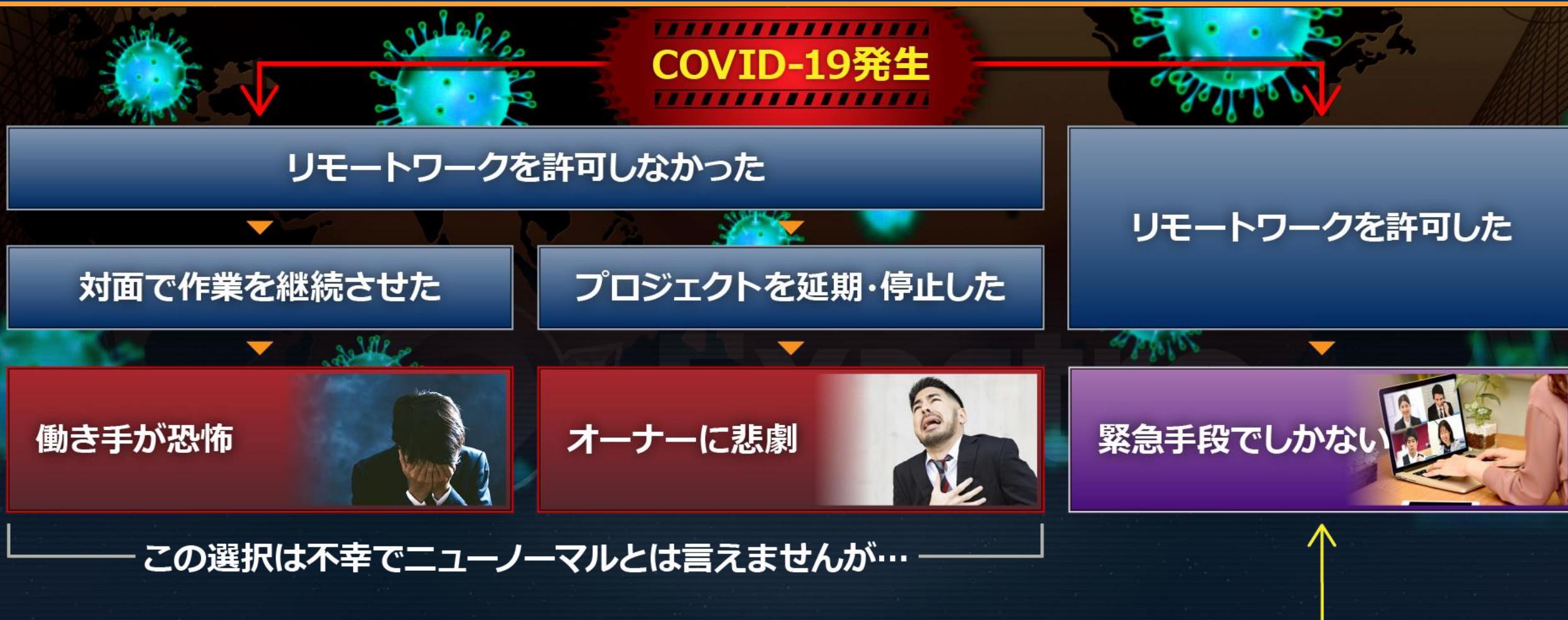
近時でダイナミックな環境

クラウドネイティブな
システムの自動化
(攻めの自動化)

COVID-19発生！ システム構築・運用の現場が直面した危機



COVID-19発生！システム構築・運用の現場が取った**3つの行動**



最も危険なのはこの選択です。緊急事態をうまく乗り越えられた気になりますが、実は今まで理由あって守ってきたセキュリティ対策を破棄している可能性があります。

システム構築・運用のリモートワークを実現するためには「自動化」が必要



どうすればシステム構築・運用業務で
恒久的なリモートワークを実現できるのか？

どうすればシステム構築・運用の現場を
新しい時代に合った形に改善することができるのか？

その手段の一つとなるのが
「自動化」です

とはいえる「狭義の自動化」だけでは不足

例えば「対面で作業を継続させた」という選択はなぜ起きたのでしょうか?
それはシステム構築・運用の現場の課題に対して
「定型作業を自動化するだけ」では解決策として不足していたからです

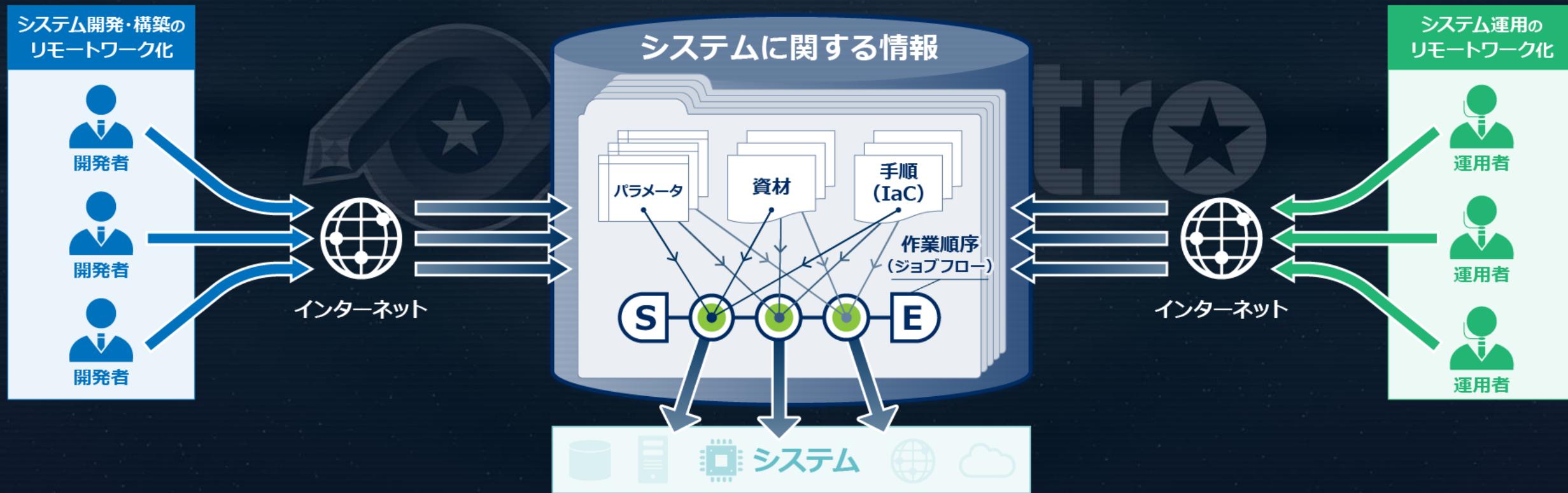
システム構築・ 運用の現場

- Excelベースのマニュアルや手順書で構築・運用している
- システムのパラメータの現在値や過去の変更履歴を管理できていない
- 何か起こると大量のマニュアルを読み替えながら、複数人体制で慎重に実行するという光景
- 必要な情報を知っているのは特定の担当者だけ
- 作業経験者が限られたりしている

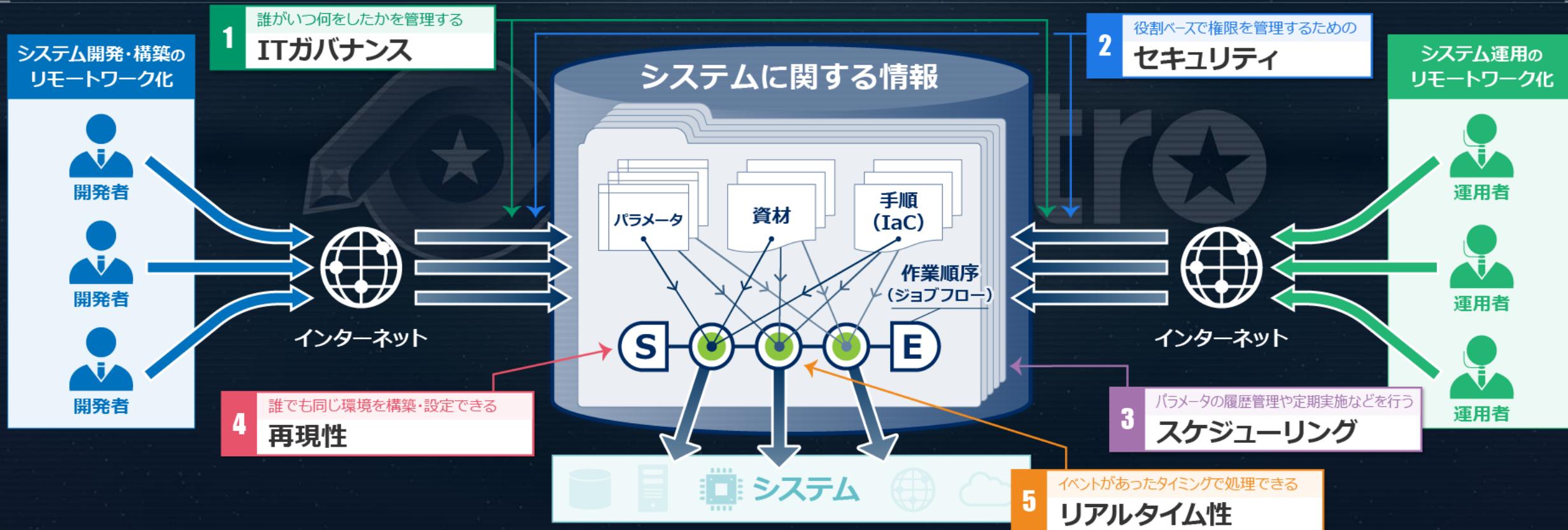
つまり人手でカバーしてきた「管理不足」や「属人化」といった潜在的な課題が
パンデミック情勢を受けて大きく顕在化したのです

ここで「広義の自動化」が実現の鍵になる

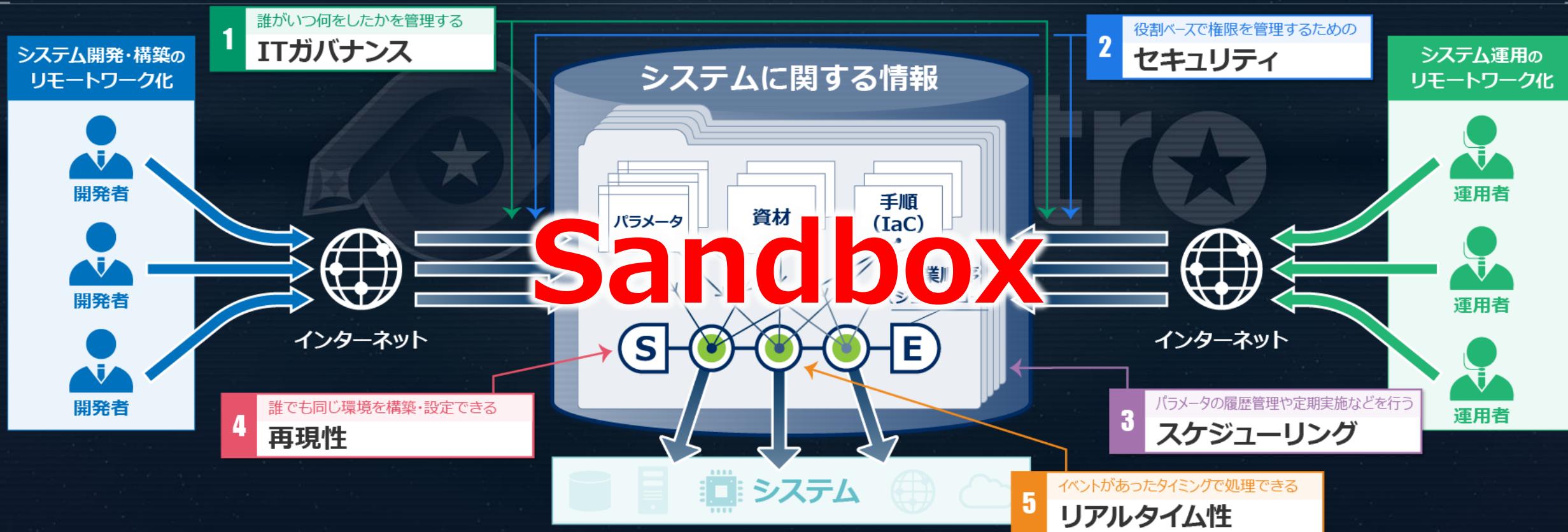
人手でカバーしてきた「**管理不足**」や「**属人化**」を解決するために
取り組むべき最大のテーマは、アナログで取り扱ってきたシステム情報を
デジタル化して一元管理する仕組みを作ることです。

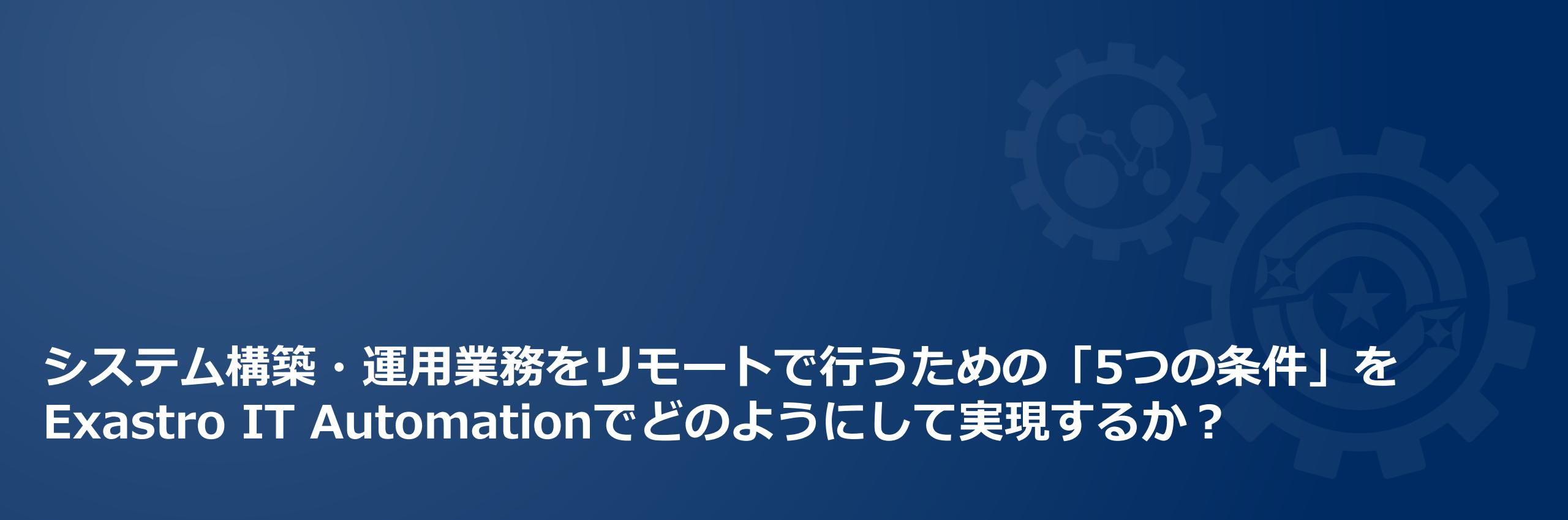


このテーマは「リモート作業を正しく行うためには何をどう仕組み化すればよいのか?」と言い換えることもできます
そしてこの視点から大きく「**5つの条件**」が必要になるのです



このテーマは「リモート作業を正しく行うためには何をどう仕組み化すればよいのか?」と言い換えることもできます
そしてこの視点から大きく「**5つの条件**」が必要になるのです



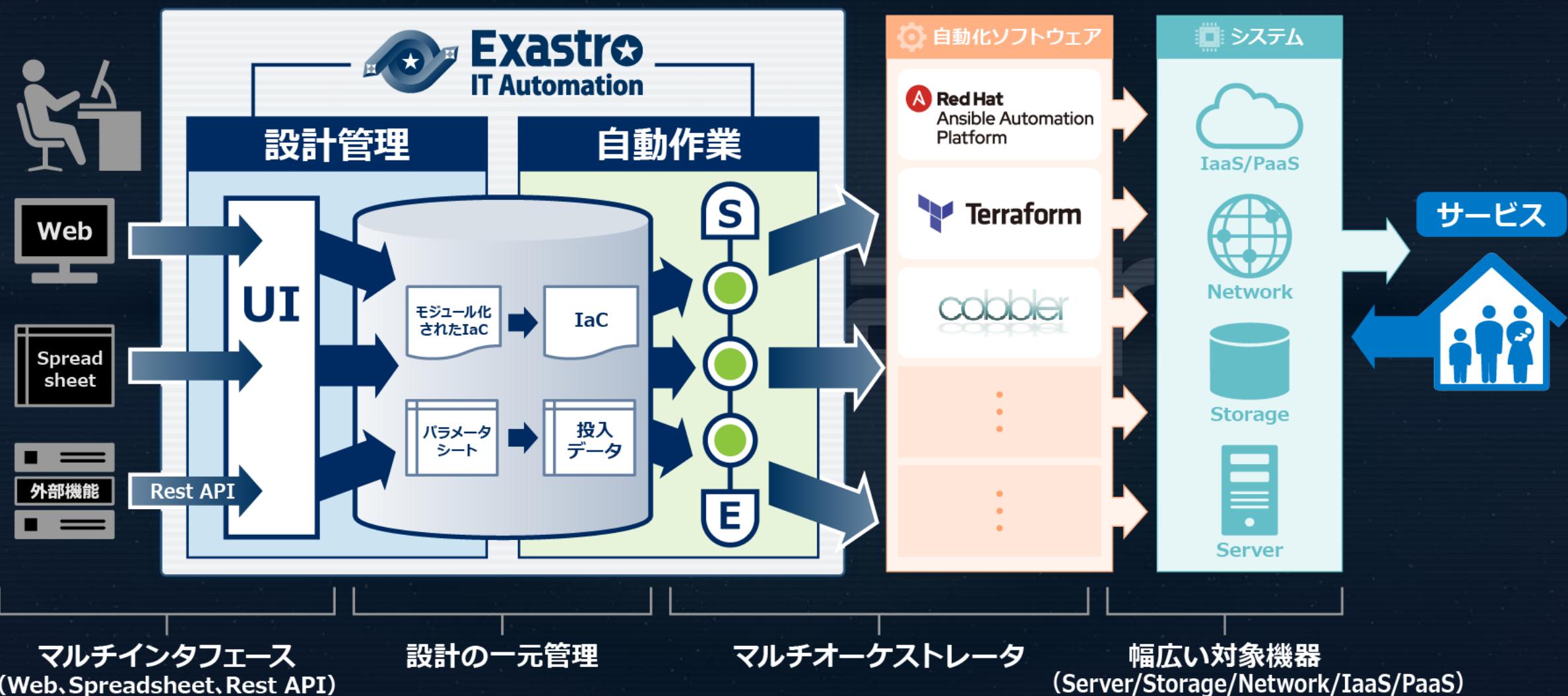


システム構築・運用業務をリモートで行うための「5つの条件」を
Exastro IT Automationでどのようにして実現するか？

Exastro IT Automation : システム情報をデジタル管理するためのフレームワーク

設計フェーズ

作業フェーズ



Exastro IT Automationの「7つの特徴」とリモートワークのための「5つの条件」



Exastro IT Automation 「7つの特徴」

I マルチインターフェースとRBAC

II パラメータをグルーピング／履歴管理する

III IaCを解析して変数を刈り取る

IV IaCをモジュール管理して再利用性を高める

V 複数の自動化ソフトウェアを繋げて実行する

VI 自動化を止める最後の切り札Pioneerモード

VII 実行状況をリアルタイムで監視する

システム構築・運用業務をリモートで行うための「5つの条件」

1 誰がいつ何をしたかを管理する
ITガバナンス

2 役割ベースで権限を管理するための
セキュリティ

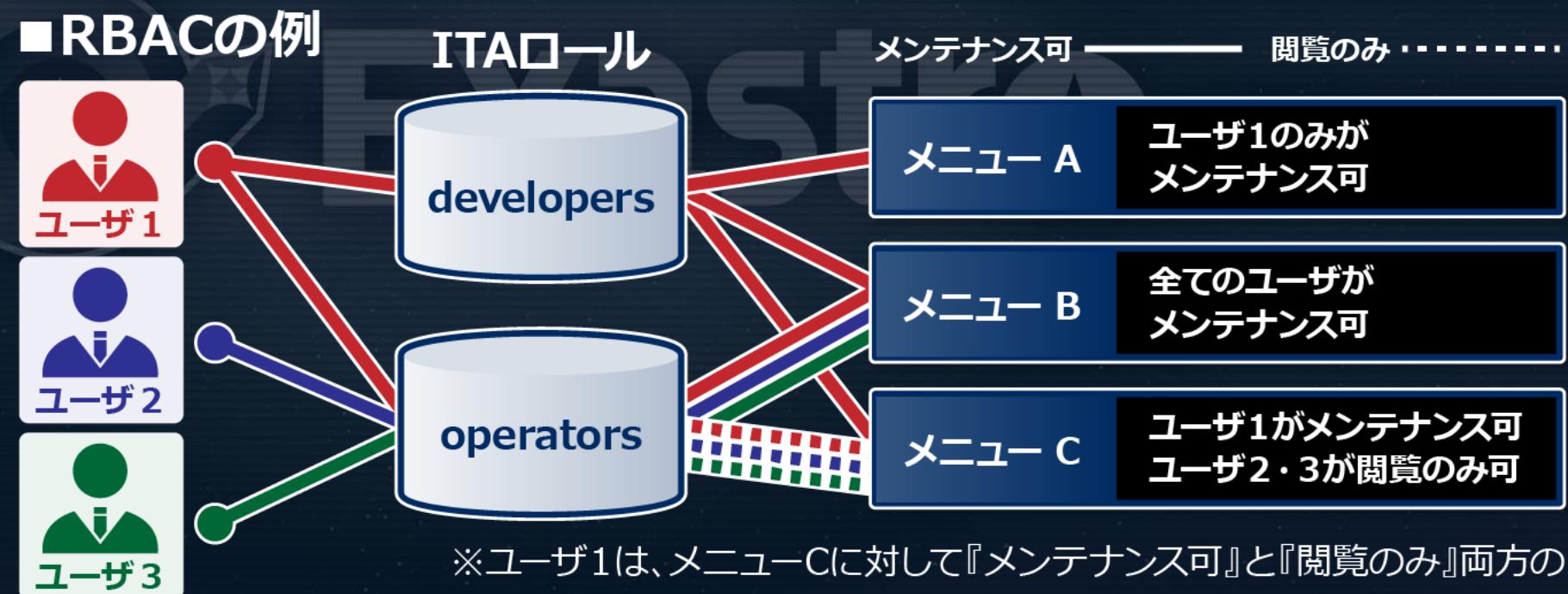
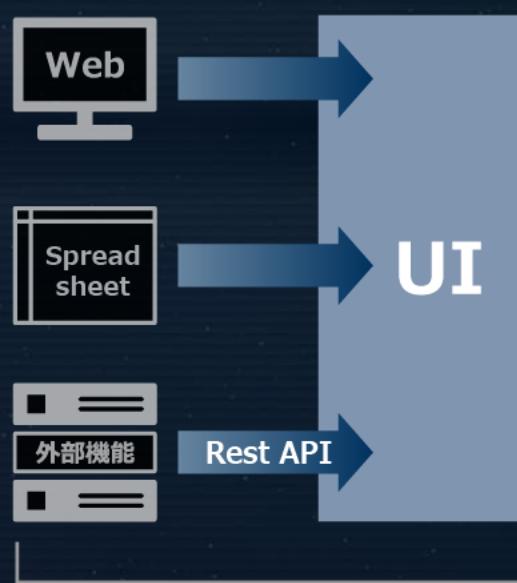
3 パラメータの履歴管理や定期実施などを行う
スケジューリング

4 誰でも同じ環境を構築・設定できる
再現性

5 イベントがあったタイミングで処理できる
リアルタイム性

1つめの特徴 - マルチインターフェースとRBAC

ユーザ操作を3種類のI/F(Web, Excel, RestAPI)から実行可能
どのI/Fからの操作でも「誰が・いつ・何をしたか?」を記録する
RBACを備えており、開発者、作業者、運用者といった役割りを定義でき
その役割りごとに出ること(参照のみ、更新、実行)を制御できる



Exastro IT Automationの「7つの特徴」とリモートワークのための「5つの条件」



「7つの特徴」

I マルチインターフェースとRBAC

II パラメータをグルーピング／履歴管理する

III JACを解析して変数を取り取る

ITガバナンス

VPNなどを使ってリモートからアクセスする場合のコンプライアンスや内部統制を確保するもの。

V 複数の自動化ソフトウェアを繋げて実行する
▼Exastro IT Automationでの対応

VI どのインターフェースからのユーザー操作でも、漏れなく「誰が、いつ、何をしたか」を記録する。
自動化を止めない最後の切り札Pioneerモード

VII 実行状況をリアルタイムで監視する

システム構築・運用業務をリモートで行うための「5つの条件」

1 誰がいつ何をしたかを管理する
ITガバナンス

2 役割ベースで権限を管理するための
セキュリティ

セキュリティ

リモートから作業できるといつても、開発者や運用者が同じ権限を持つてしまえば、システムの安全性は担保できない。

再現性

▼Exastro IT Automationでの対応

RBACで役割ごとに操作(参照のみ、更新、実行)を制御する。

リアルタイム性

2つめの特徴 - パラメータをグルーピング／履歴管理する

システムのパラメータ情報をグルーピング／履歴管理する



2つめの特徴 - パラメータをグルーピング／履歴管理する

パラメータシートは履歴管理機能を標準装備
設計履歴から抽出した情報を使ってシステム更改する仕組み

ITA の履歴管理機能つきパラメータシート

ホスト	オペレーション		パラメータ				設計日
	日時	作業名	P1	P2	P3	…	
hostA	12/20	クリスマス対応	1024	512	2048	…	10/1
hostA	10/9	hostB 増設	512	256	1024	…	8/3
hostA	9/3	システムリリース	256	128	512	…	7/7
hostB	12/20	クリスマス対応	16	32	64	…	10/1
hostB	10/9	hostB 増設	32	64	128	…	8/3

設計者は設計に集中できる

例えば
“10/9”で
パラメータを
抽出すると

“10/9” のシステムの期待値

ホスト	パラメータ				設計日
	P1	P2	P3	…	
hostA	512	256	1024	…	8/3
hostB	32	64	128	…	8/3

運用者は 運用に
集中できる

システム更改

妥当性確認



システム

Exastro IT Automationの「7つの特徴」とリモートワークのための「5つの条件」



「7つの特徴」

I マルチインターフェースとRBAC

II パラメータをグルーピング／履歴管理する

III IaCを解析して変数を刈り取る

スケジューリング

履歴管理が大きなポイント。アを繋げて実行する
エンジニアが1箇所に集まるような環境では相互に
コミュニケーションを取ることでギャップを埋めて
きたが、リモートでそれぞれのエンジニアが更新を
掛けるとシステムの整合性が失われる。

システム構築・運用業務をリモートで行うための「5つの条件」

1 誰がいつ何をしたかを管理する
ITガバナンス

2 役割ベースで権限を管理するための
セキュリティ

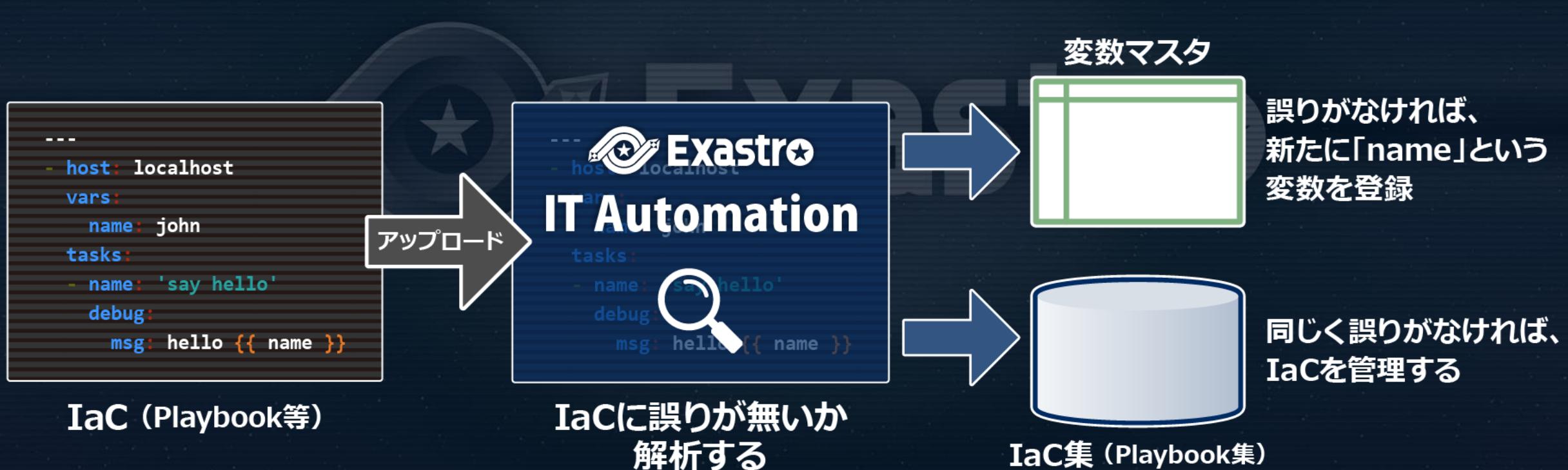
3 パラメータの履歴管理や定期実施などを行う
スケジューリング

誰でも同じ環境を構築・設定できる
▼Exastro IT Automationでの対応

履歴管理を活用し、設計履歴の中に現在のシステム構成と同じ状態を保持しておく、これを使って、システム更改や設定値のスコアリングを可能とする。

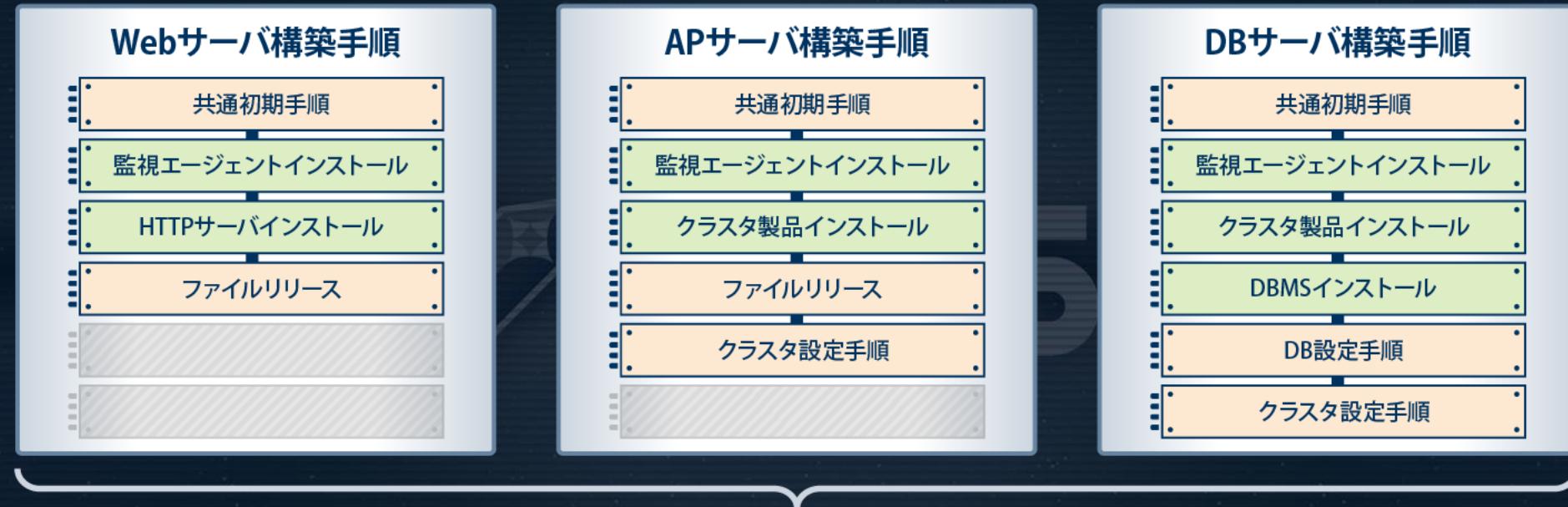
3つめの特徴 - IaCを解析して変数を刈り取る

IaCがアップロードされるとまずIaCに誤りが無いか解析する
誤りがなければ、IaCの記述から変数名を刈り取って管理する
変数名を選択式で利用し誤植等のヒューマンエラーを防止する

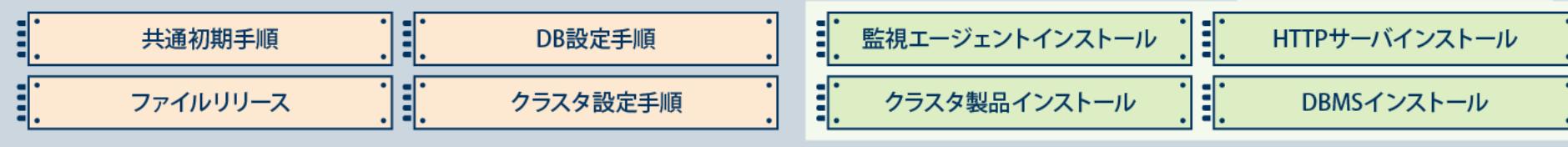


4つめの特徴 - IaCをモジュール管理して再利用性を高める

IaC(Playbook等)を一発モノで終わらせず再利用して利用し続けられる
ように、モジュール化して作業時に組み立てる仕組み



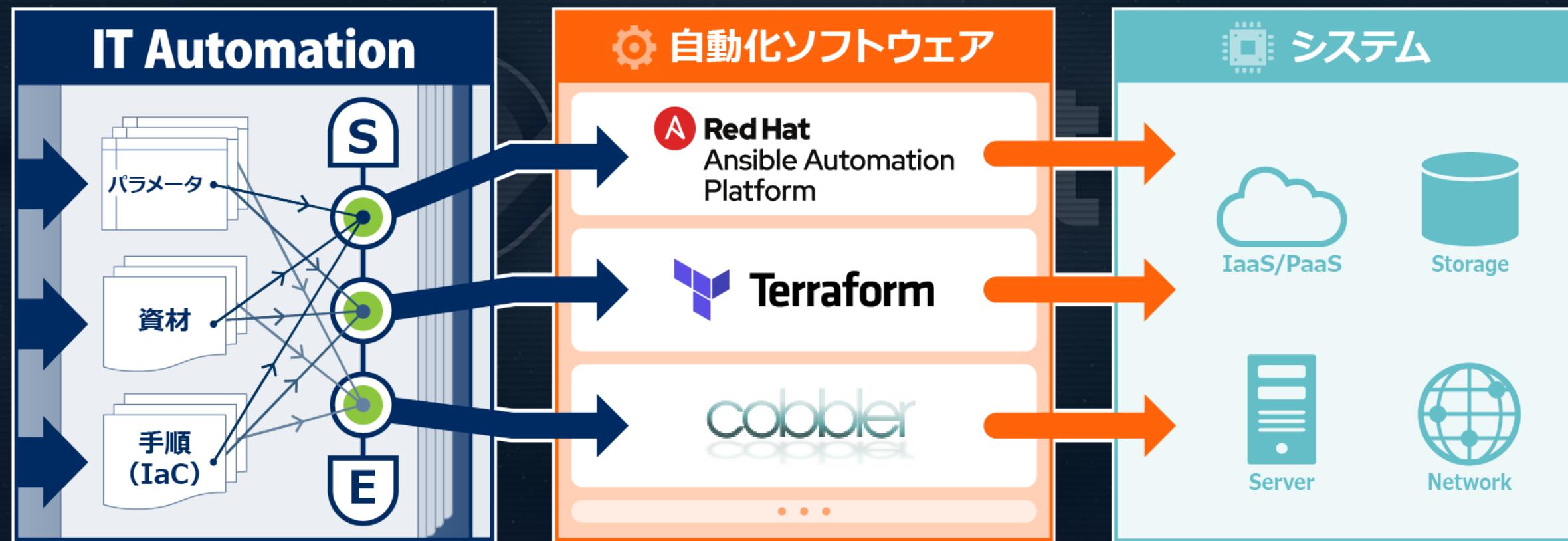
共通の手順はモジュール化し再利用できるように管理する



5つめの特徴 - 複数の自動化ソフトウェアを繋げて実行する

複数の自動化ソフトウェアを繋げて一本の作業フローを定義できる
また自動化ソフトウェアの動作に必要な投入データを自動生成する

例) (Ansibleの場合) 必要なPlaybookを集めて繋げ、ノード毎にパラメータからhost_varsを作る



6つめの特徴 - 自動化を止めない最後の切り札Pioneerモード

Ansibleのどのモジュールを使っても自動化できない場合に、手動作業を挟んでしまうと自動化のメリットが半減する。そこで、**自動化を止めない最後の切り札として、ITAではPioneerモードを提供。**

▼ Pioneer専用「対話ファイル」

expectコマンド
+
変数埋め込み
+
条件分岐/繰り返し
を可能とした
独自のIaC



A Red Hat
Ansible Automation
Platform



どれを使っても
自動化できない場合



ssh / telnet

ITA専用のAnsibleモジュール
ssh / telnet のいずれかOKであれば対話可能

Exastro IT Automationの「7つの特徴」とリモートワークのための「5つの条件」



「7つの特徴」

I マルチインターフェースとRBAC

II パラメータをグルーピング／履歴管理する

III IaCを解析して変数を刈り取る

IV IaCをモジュール管理して再利用性を高める

V 複数の自動化ソフトウェアを繋げて実行する

VI 自動化を止めない最後の切り札Pioneerモード

VII 実行状況をリアルタイムで監視する

再現性

履歴管理とも関連するが、リモート作業で別のエンジニアが実施した作業を完全に再現するためには、マシンの正確性を活用することが避けられない。（「自動化」が求められる最大のポイント）

再現性

▼Exastro IT Automationでの対応

左記の4つの特徴を駆使することで管理性を向上し、作業負荷の大幅な低減を可能とする。

● 4 誰でも同じ環境を構築・設定できる
再現性

5 イベントがあったタイミングで処理できる
リアルタイム性

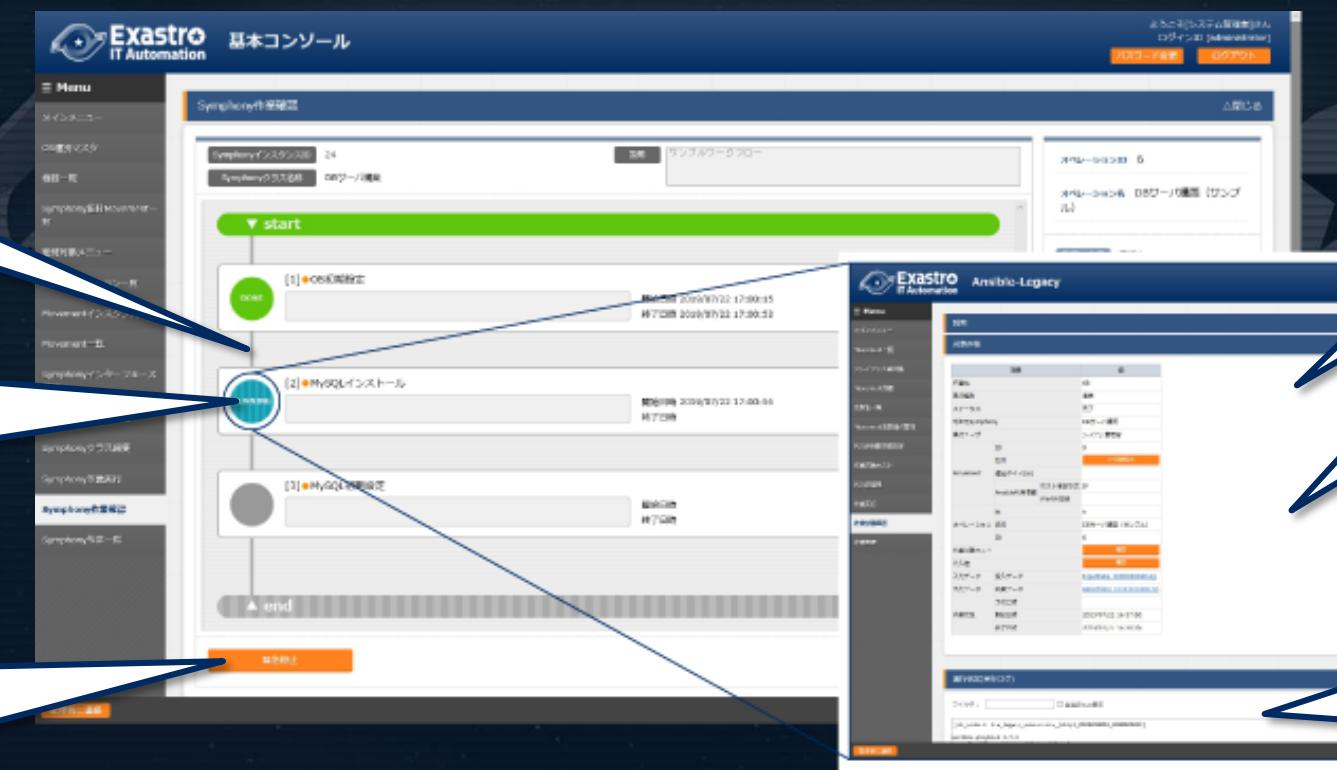
7つの特徴：⑦実行状況をリアルタイムで監視する

手動作業と比較して遜色なく実行状況をリアルタイム把握することを重視
また実行記録(作業エビデンス)を管理し欲しい時にダウンロード可能

作業フローの途中に
「保留ポイント」
を設定可能

実行状況をクリック
すればドリルダウン
が可能

非常時には「緊急停
止」で作業をストッ
プすることが可能



投入データ(自動生成)
がダウンロード可能
(zip)

実行結果(作業エビデ
ンス)がダウンロード
可能(zip)

自動化ソフトウェア
の実行状況をリアル
タイムで表示

Exastro IT Automationの「7つの特徴」とリモートワークのための「5つの条件」



「7つの特長」

I マルチインターフェースとRBAC

リアルタイム性

通常の運用管理ツールはジョブやバッチ処理などの結果を待つものだが、システム運用業務はバッチ処理とは違い、よりシビアなリアルタイム性が要求される。その要求はリモートワークでも変わることはない。

▼Exastro IT Automationでの対応性を高める

現場での手動作業と比較して遙かに実行状況をリアルタイムに把握することができる。同時に作業記録を漏れなく管理することで作業の再現性に繋げる。

VI 自動化を止めない最後の切り札Pioneerモード

VII 実行状況をリアルタイムで監視する

「5つの条件」

1 誰がいつ何をしたかを管理する

2 パラメータの履歴管理や定期実施などを行う

3 スケジューリング

4 誰でも同じ環境を構築・設定できる

再現性

5 イベントがあったタイミングで処理できる

リアルタイム性

Exastro IT Automationの「7つの特徴」とリモートワークのための「5つの条件」



Exastro IT Automation 「7つの特徴」

I マルチインターフェースとRBAC

II パラメータをグルーピング／履歴管理する

III IaCを解析して変数を刈り取る

IV IaCをモジュール管理して再利用性を高める

V 複数の自動化ソフトウェアを繋げて実行する

VI 自動化を止める最後の切り札Pioneerモード

VII 実行状況をリアルタイムで監視する

システム構築・運用業務をリモートで行うための「5つの条件」

1 誰がいつ何をしたかを管理する
ITガバナンス

2 役割ベースで権限を管理するための
セキュリティ

3 パラメータの履歴管理や定期実施などを行う
スケジューリング

4 誰でも同じ環境を構築・設定できる
再現性

5 イベントがあったタイミングで処理できる
リアルタイム性

活用事例



【事例①】大規模システムの様々な運用監視設定を効率化しました

課題



システムのパラメータが時系列で構成管理できておりらず、またPlaybookの部品化ができておらず、構築および運用における作業の自動化/効率化が実現できていなかった。



Exastro IT Automationの機能で

- システムのパラメータを構成管理
 - Ansible Playbookを部品管理
- することで設計～構築、および運用のフェーズを繋げて、作業の自動化を実現した。

解決策



効果

例1: 運用監視設定作業

構成管理
自動化前

年間4億円の作業工数で実施していた

4億円/年

構成管理
自動化後

年間1億円の作業工数で
実施できるようになった(75%削減)

1億円/年

75% 削減

例2: 運用コマンドリストメンテナンス作業

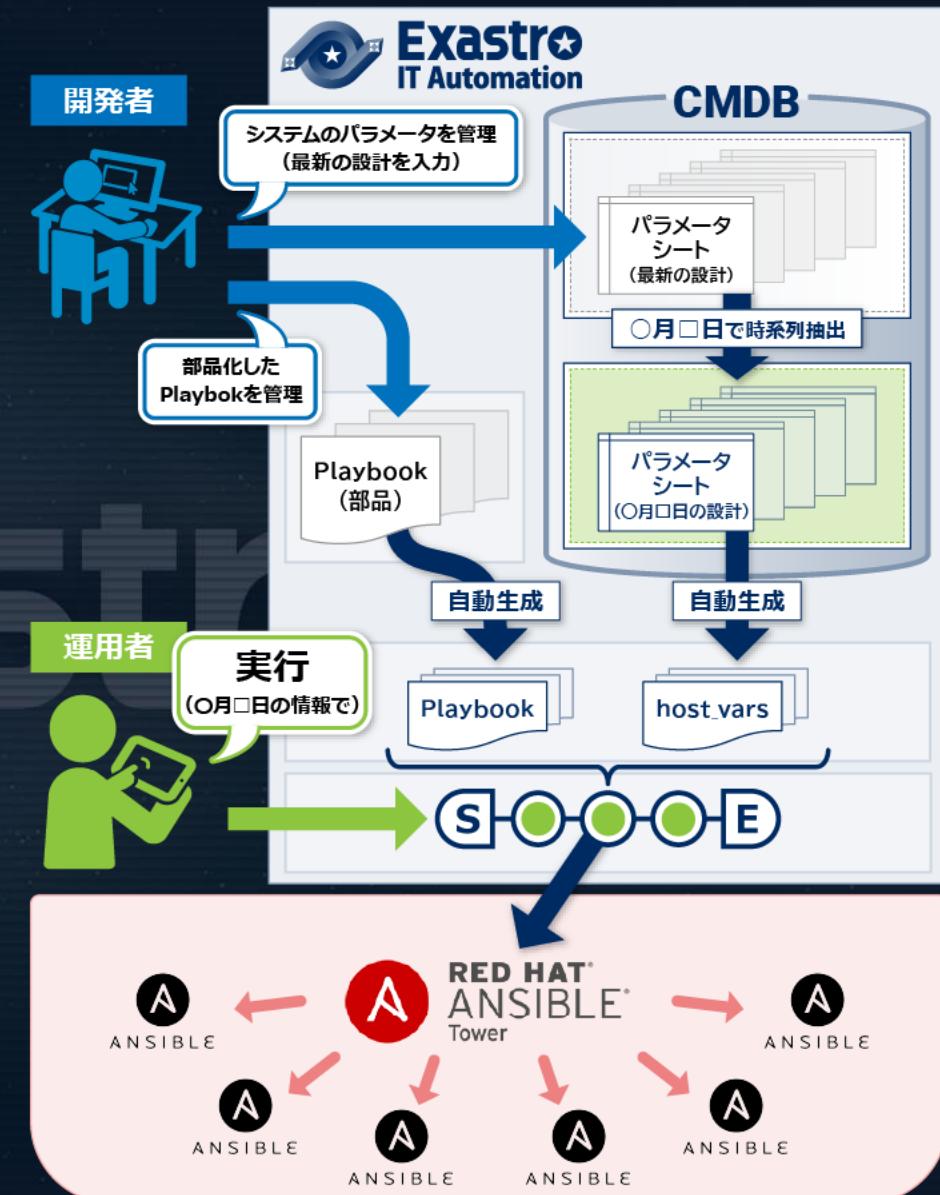
年間4,000万円の作業工数で実施していた

4,000万円/年

年間800万円の作業工数で
実施できるようになった(80%削減)
メンテナンスまでのリードタイムが不要となり
即時反映がされることで利便性も向上した。

800
万円/年

80% 削減



【事例②】システムの構成管理を活用して機器停止のサービス影響予測しました

課題



大規模キャリアシステムにおいて、機器の計画停止、および突発的な停止によるサービス影響の調査に多くの工数がかかっていた。

解決策

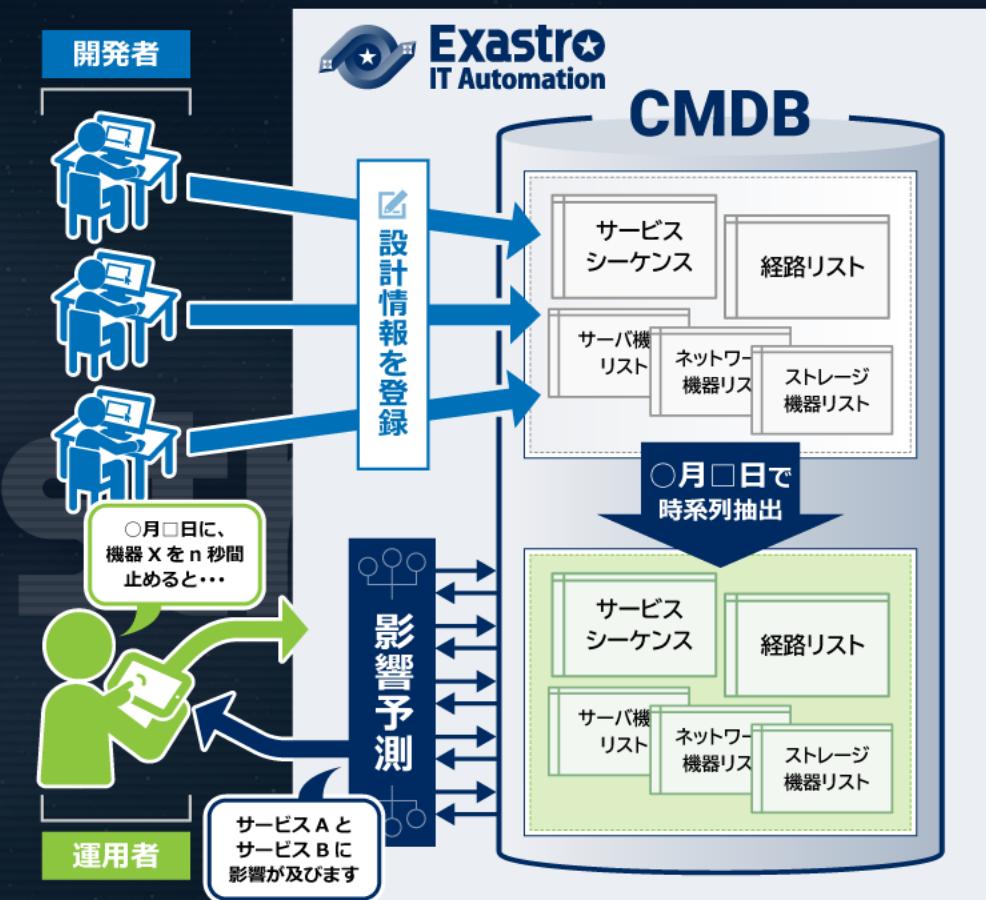


Exastro IT Automationの機能で**システムを構成管理**することにより、機器停止による**サービス影響を自動予測**することを可能とした。

効果



1回の調査で約80万円の工数/費用を要していたところ、本施策により工数不要となった。年間約120回の調査があったので**約9,400万円/年の削減効果**が確認された。



※影響予測はITAが提供する画面フレームワークを使ってカスタム開発。

【事例③】大規模イベント向けに3万台の機器を自動キッティングしました

課題



大量のネットワーク機器の初期設定を行う必要があった。しかし、手作業では多くの工数が必要であり、また大量の設定値を表計算ソフトで管理するのにも限界があった。

解決策

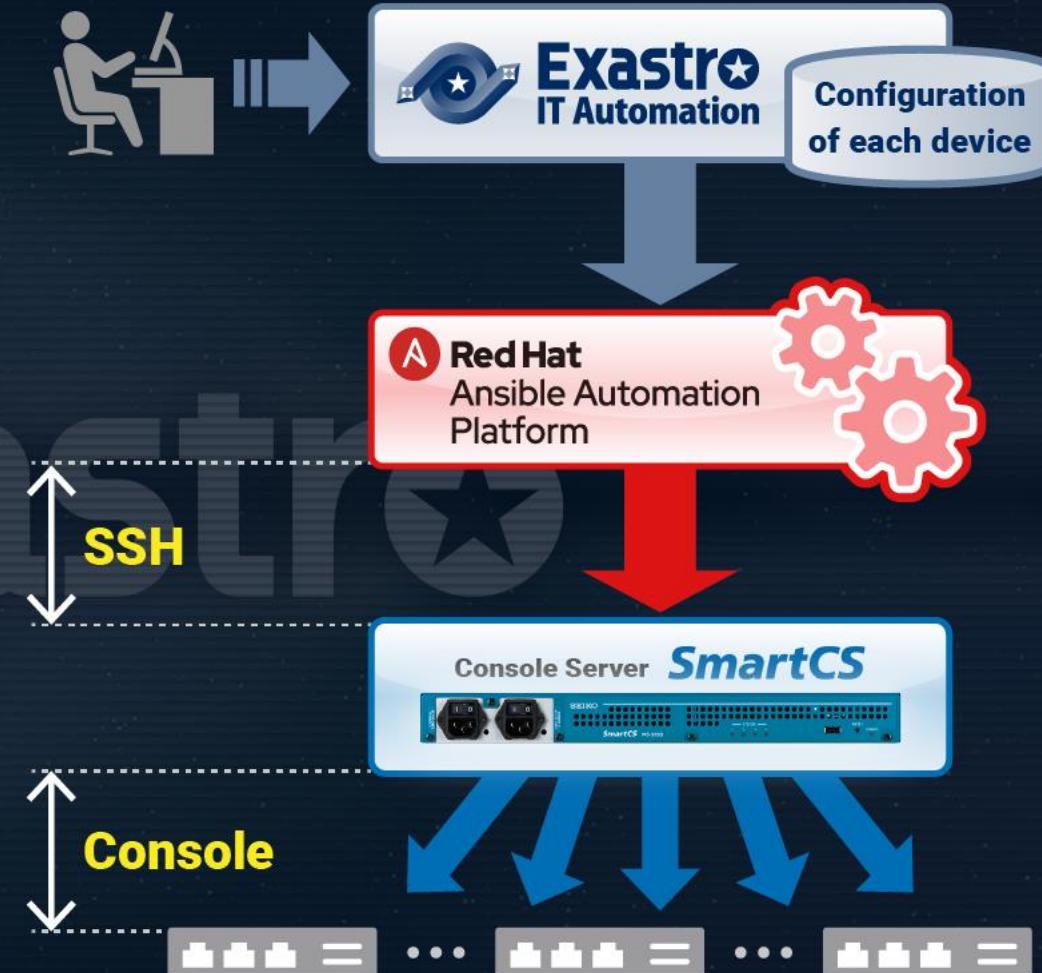


SmartCSを活用して複数のネットワーク機器を同時に設定可能にした。
また、設定値は**CMDBで一元管理**し、**Ansibleで設定を自動投入**することで課題を解決できた。

効果



スイッチなど6種類のネットワーク機器の初期設定を自動化することができ、
2,000台/月のリリースを達成した。



本日のタイムスケジュール



本日のタイムスケジュール



► トークライブ01 13:45~14:15

30
min

NTTドコモと共に考える！

運用の進むべき道～システム運用からサービス運用へ～



パネラー

山上 拓也

NTTドコモ
サービスデザイン部
運用企画担当



パネラー

遠藤 裕己

NEC
OMCS事業部

モデレーター 吉田 功一 NEC サービス＆プラットフォームSI事業部

► セッション02 14:20~14:55

35
min

まずは守りの自動化を！

日々繰り返すアラート対応はExastroにお任せあれ



講 師

河野 友生子

NEC
サービス&プラットフォーム
SI事業部

「守りのデジタル化・自動化」を適用すべき代表的なケースとして、日々繰り返すアラート対応を如何にして自動化・省力化することができるのかを、実機デモを交えながら分かりやすく解説します。

➡ 実機デモあり！ ⬅

► トークライブ02 14:55~15:20

25
min

レッドハットと共に考える！

クラウドネイティブに対応したあるべきインフラの姿とその運用



パネラー

安楽 慎吾

レッドハット株式会社
テクニカルセールス本部
パートナーソリューション
アーキテクト部



パネラー

久富 孝司

NEC
先端SI技術開発本部
OSS推進センター

モデレーター 吉田 功一 NEC サービス＆プラットフォームSI事業部

► セッション03 15:25~16:00

45
min

そして攻めの自動化へ！

Exastroによるクラウドシフトの実現方法



講 師

脇谷 徹

NEC
サービス&プラットフォーム
SI事業部

ガバナンスの効いたAWSシステムを構築し、さらに稼働中の仮想マシン(EC2)にパッチ適用するといったシナリオを題材に、クラウドネイティブにおいて2種類の自動化メソッドの両方が必要になることを解説します。

➡ 実機デモあり！ ⬅





Exastro 