

# OSSユーザーのための勉強会 <OSS X Users Meeting> #7 CloudConductor と PrimeCloud Controller

## CloudConductorの特長と最新動向

---

2014年7月17日

TIS株式会社



# 松井 暢之（まつい のぶゆき）



TIS株式会社  
コーポレート本部 戦略技術センター

 @n\_matsui  nbyk.matsui  nmatsui



- ～2003 現場PJでアーキテクト兼モデラー兼プログラマ兼…を歴任
- 2003～2008 基盤技術センター（現 戦略技術センター）にて不芳PJの火消しに奔走
- 2009 全社生産性向上の企画策定に従事
- 2010～2012 オープンでエッジな技術を活用した事業企画に従事

2013～



**CloudConductor** の企画開発を開始  
design-oriented cloud orchestrator

# Agenda

---

1. 加速するクラウド化
2. クラウド時代のシステムインテグレーション
3. デザイン指向クラウドオーケストレータ CloudConductor
4. CloudConductorのロードマップ
5. CloudConductorの活用例 ～災害対策のデモ～



# Agenda

---

1. 加速するクラウド化
2. クラウド時代のシステムインテグレーション
3. デザイン指向クラウドオーケストレータ CloudConductor
4. CloudConductorのロードマップ
5. CloudConductorの活用例 ～災害対策のデモ～



## 情報システム担当者の悩み

新しいこともやりたいし、  
ビジネスのアジリティや  
コスト削減を考えると

そろそろウチのシステムも  
**クラウド化**すべきだよな…

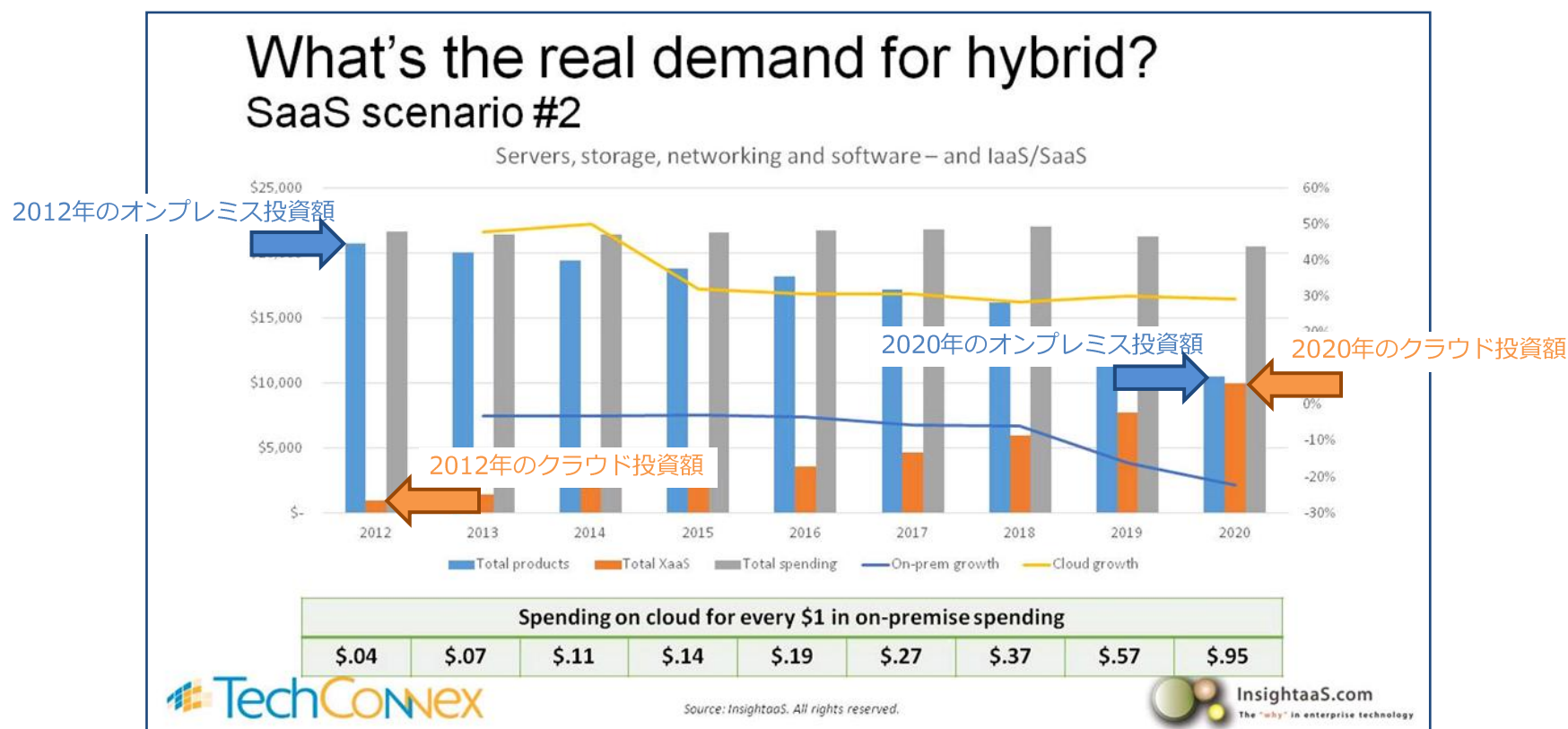
でもクラウドの専門家は  
なかなか居ないし

今の資産を全部クラウドへ  
持っていけないだろうし

そもそもクラウド化して  
データは大丈夫なのか…？

## IT投資額のトレンド予測

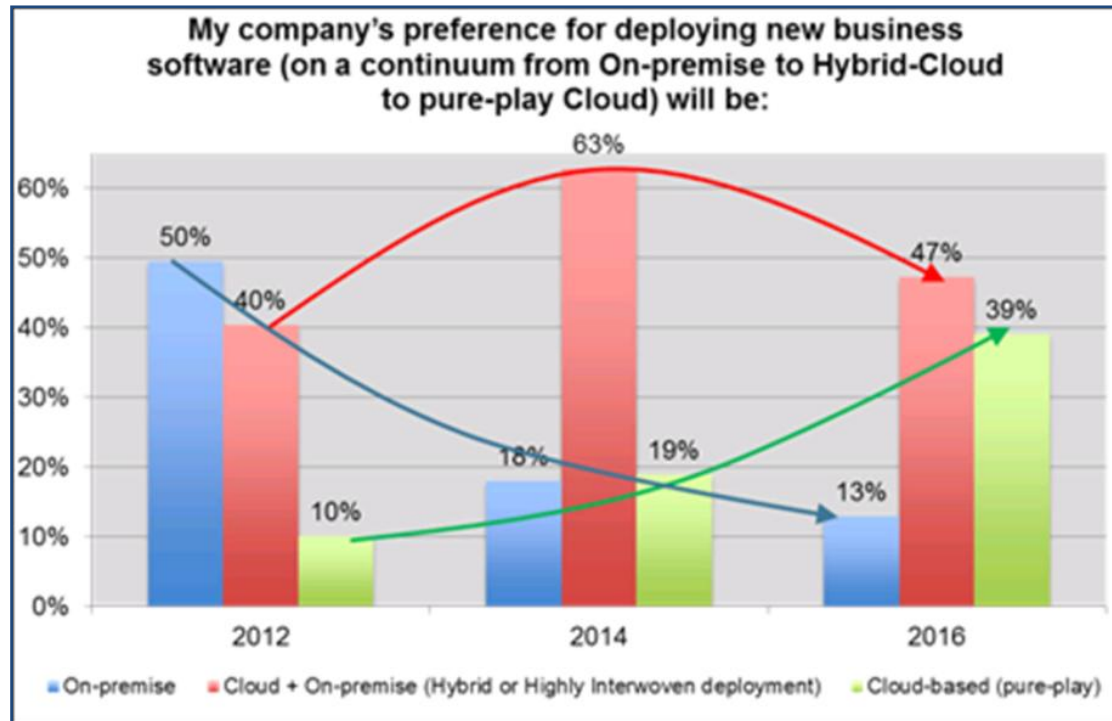
- クラウドとオンプレミスのIT投資額予想 (Mary Allen (2014)<sup>+1</sup>より転載)
  - 2012年にはオンプレミス1ドルに対してクラウドへの投資額はわずか0.04ドルであったが、オンプレミスへの投資性向は年々低下し、2020年には両社がほぼ同じ投資額となる



<sup>+1</sup> Allen (2014). InsightaaS and TechConnex talk cloud tactics and tools.  
<http://www.insightaas.com/insightaas-and-techconnex-talk-cloud-tactics-and-tools/>

## デプロイ先インフラのトレンド予測

- Saugatuck Technologyによる米国、欧州、アジアの大企業CIOに対する調査 (Louis Naugès (2012)<sup>+2</sup>より転載)
  - Business softwareを今後どのようなインフラにデプロイするか？
  - 2012年には50%だったOn-premiseが2016年にはわずか13%へ激減
  - Cloudも伸びるが中期的には半数がhybrid cloud

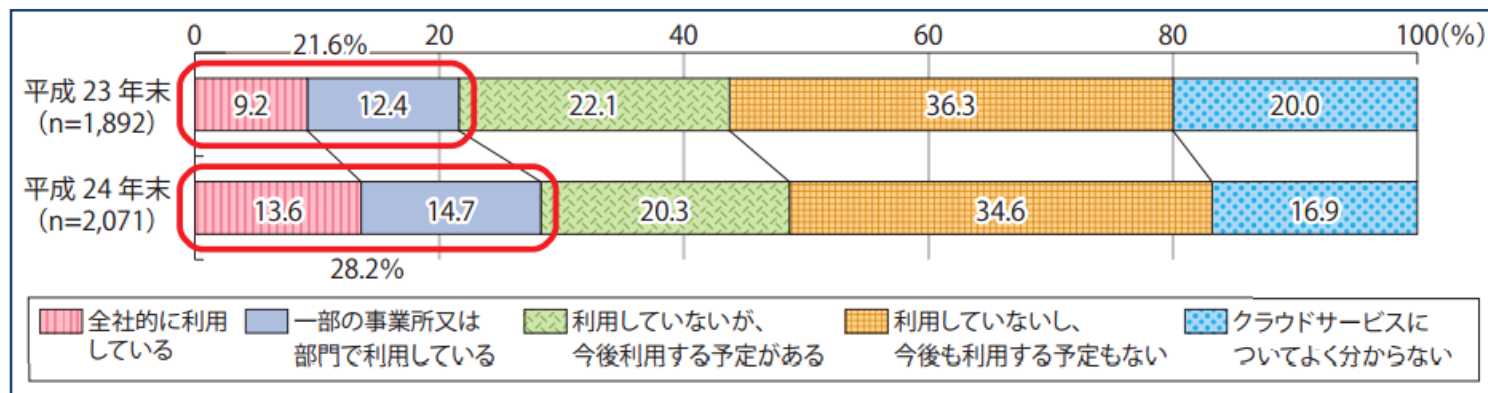


<sup>+2</sup> Naugès (2012). Cloud Computing or the IT Industrial Revolution (ITIR): Economic impacts on supply and demand in IT. <http://www.cloud-migration.eu/en/webedition/business-webedition/louis-nauges.html>

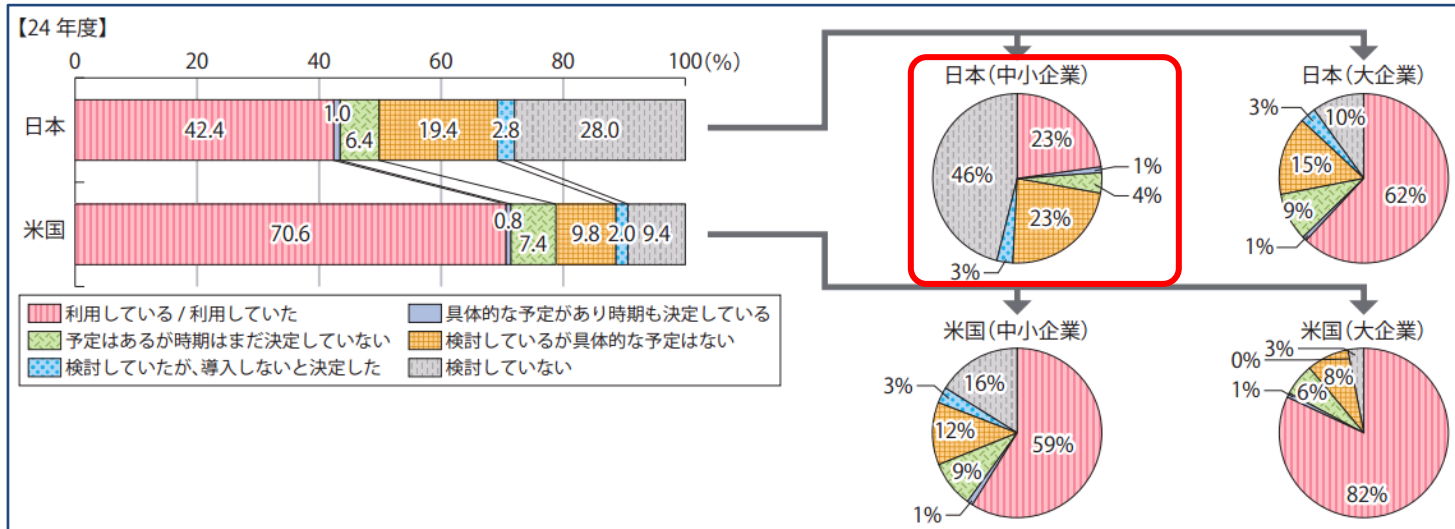


# クラウドサービス利用状況

- クラウドサービス利用状況（総務省 情報通信白書（2013）<sup>+3</sup>より転載）
  - 国内企業のクラウド活用は進んでおらず、その傾向は中小企業でより顕著



## クラウドサービスの利用実績の日米比較



<sup>+3</sup> 総務省 (2013).平成25年版情報通信白書 <http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h25/pdf/>



# Agenda

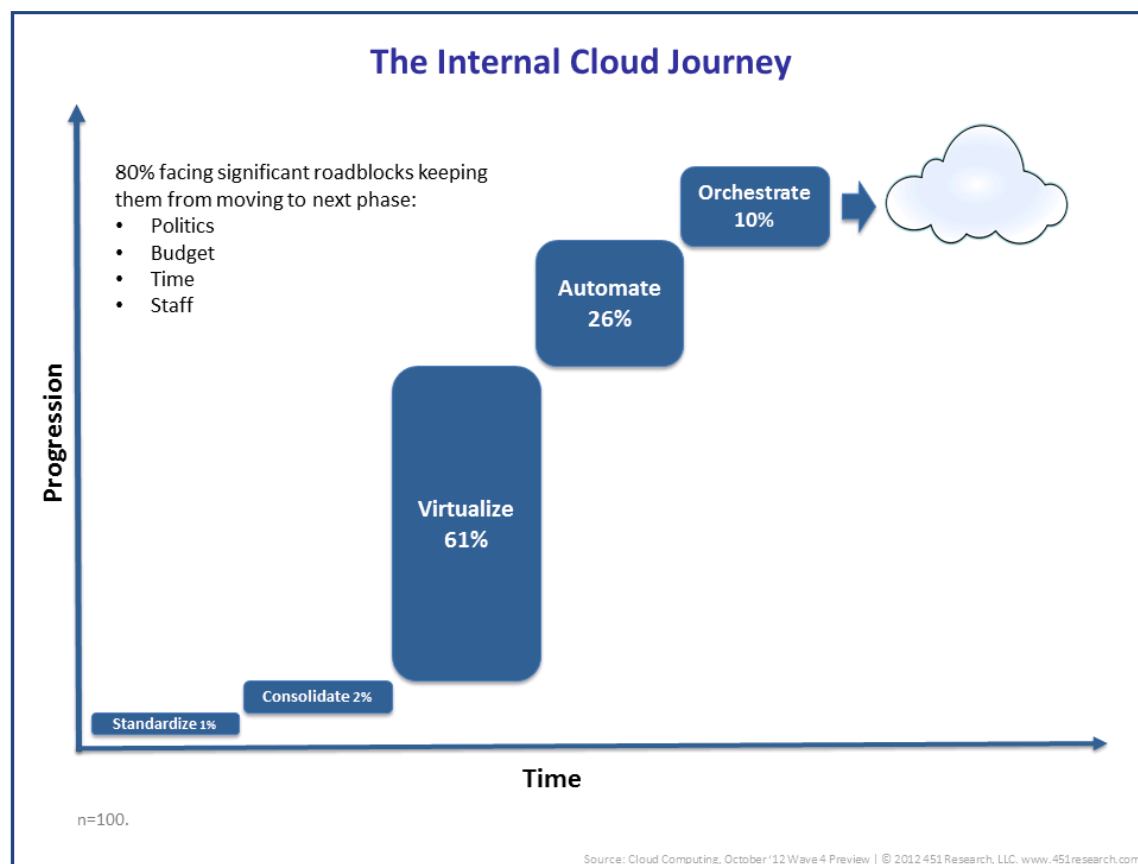
---

1. 加速するクラウド化
2. クラウド時代のシステムインテグレーション
3. デザイン指向クラウドオーケストレータ CloudConductor
4. CloudConductorのロードマップ
5. CloudConductorの活用例 ～災害対策のデモ～



## 仮想化・自動化の次はOrchestration

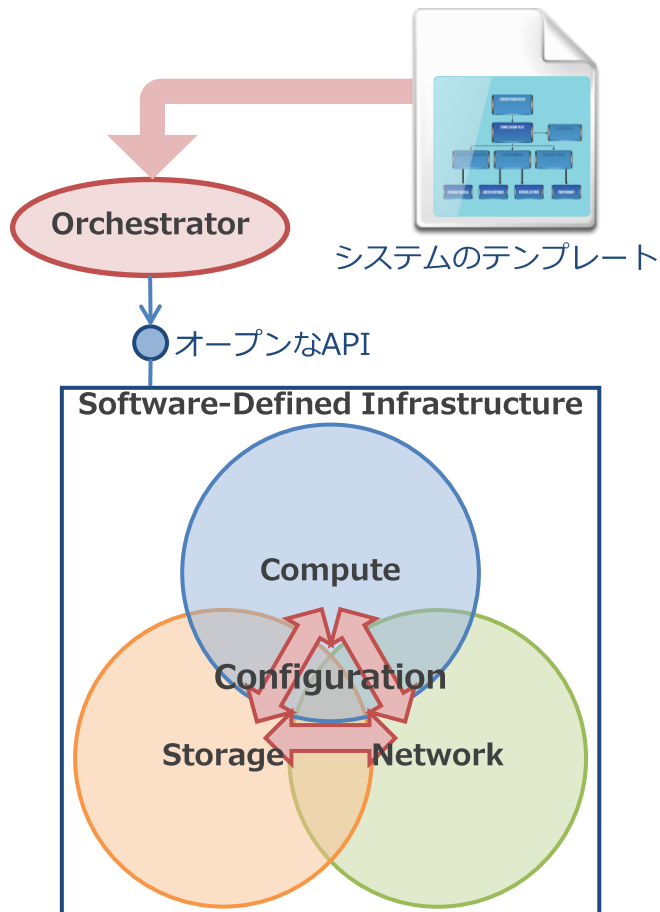
- プライベートクラウドの成熟モデル (Sean Hackett(2012)<sup>+5</sup>より転載)
  - プライベートクラウドはインフラの統合と標準化からはじまり、仮想化・自動化を経てOrchestrationで終わる



<sup>+5</sup> Hackett (2012). Internal Cloud Roadblocks Threaten to Disrupt the Pace of Public Cloud Adoption  
<http://theinfopro.blogs.451research.com/index.php/2012/11/internal-cloud-roadblocks-threaten-to-disrupt-the-pace-of-public-cloud-adoption/>

# Infrastructure as Code と Orchestration

- Infrastructure as Code と Orchestrationの価値
  - システム基盤の設計と構築に関する専門家の暗黙的ノウハウをコードによってパターン化・標準化し、クラウドへ自動構築できる



- ✓ 手作業による手間の削減とミスの回避
- ✓ 設計と実装の乖離の防止
- ✓ インフラ構成のバージョン管理
- ✓ インフラ構成の容易な再現
- ✓ 自動的なインフラヘルスチェック
- ✓ 保守しやすく回復性の高い標準化された運用
- ✓ etc...

# Immutable & Disposable Infrastructure

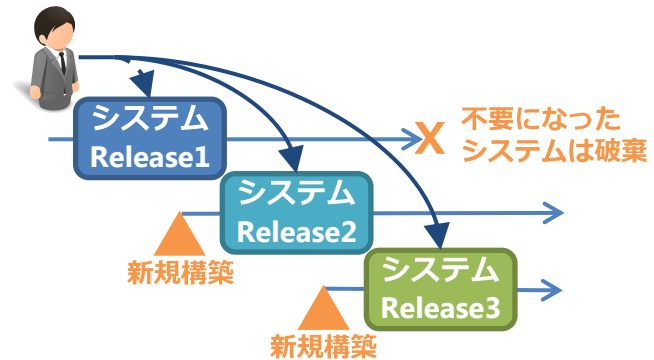
- ImmutableでDisposableなインフラの価値
  - クラウド+Orchestrationにより、一度構築した環境は二度と更新せず、**システムのリリース毎に環境一式を新たに構築**して切り替えれば良くなる

## 従来のリリース



システム更新は停止を伴うか、ローリングリリース等の工夫が必要  
前リリースは失われるため、フォールバックは容易ではない  
更新を重ねるため、同じシステムを再現することは容易ではない

## ImmutableでDisposableなリリース



新システムを構築して切り替えることがリリースにあたる  
フォールバックする場合、前リリースに切り替えなおすだけ  
パターンからのOne-wayのため、システムはいつでもクリーン

- ✓ リリースのダウンタイム削減
- ✓ 容易なフォールバック
- ✓ システムのブラックボックス化抑制
- ✓ etc...

# Immutable & Disposable Infrastructure

- ImmutableでDisposableなインフラを実現するためには、システム全体のアーキテクチャと運用プロセスの考慮が必要
  - 特定サーバにしか存在しないデータが無いシステムアーキテクチャ
  - 公開APIのみでサブシステムと結合した疎結合なシステムアーキテクチャ
- 本番障害発生時でも、構築された本番環境に手を入れずパターンを修正してシステムを再リリースする運用プロセス
- もしくは、手を入れざるを得なかった環境をパターンへ確実にリバーシできる運用プロセス

## クラウド時代のシステムインテグレーション

- これからのシステムインテグレーションは、

**高信頼**で**安全**で**柔軟**なシステムを、**ビジネスのアジリティ**を  
満足するスピードで、**安価**に

提供しなければならない

- そのためには、

クラウドを前提としImmutableでDisposableなシステムの  
アーキテクチャを**パターン化**し、システムの構築と運用を  
**自動化・自律化・標準化**する

ことが最も重要となる



CloudConductor

# Agenda

---

1. 加速するクラウド化
2. クラウド時代のシステムインテグレーション
3. デザイン指向クラウドオーケストレータ CloudConductor
4. CloudConductorのロードマップ
5. CloudConductorの活用例 ～災害対策のデモ～



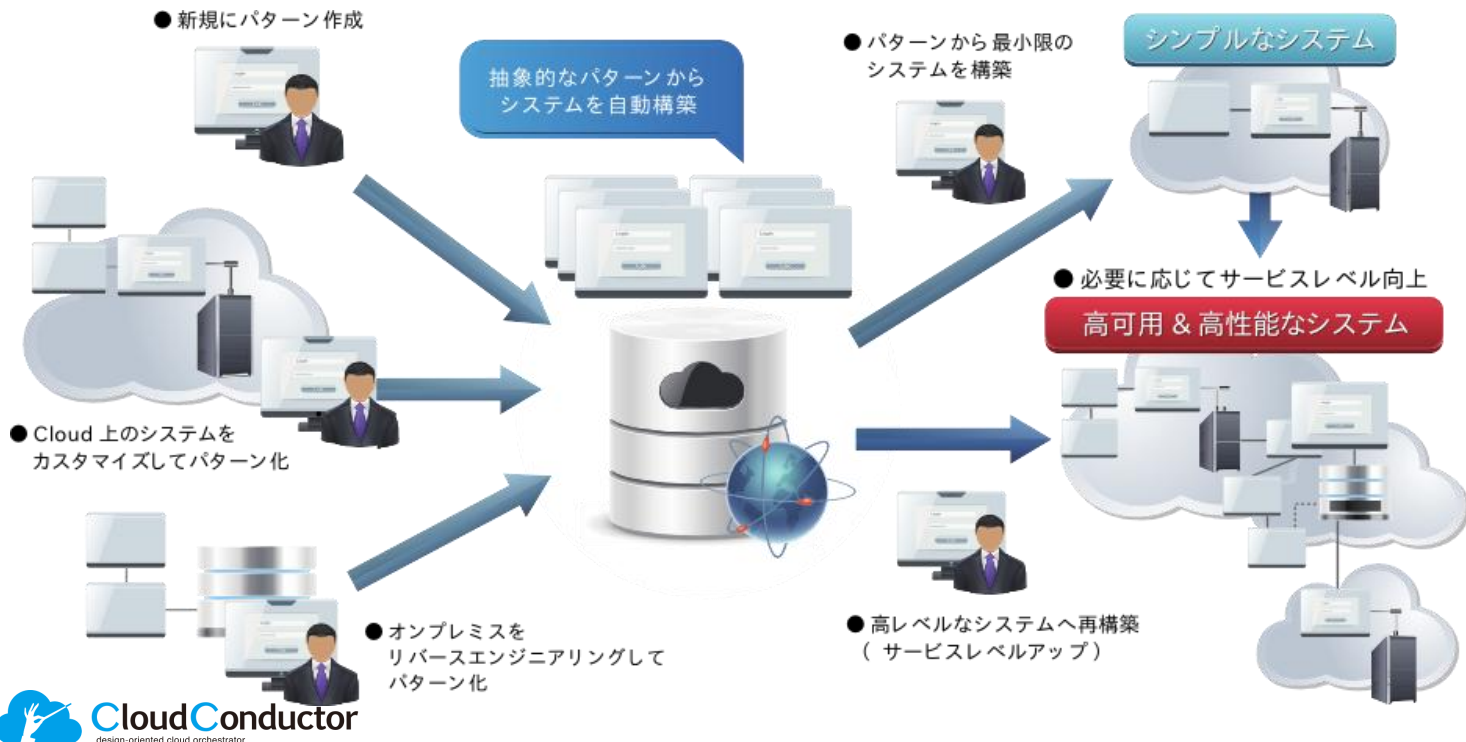


# デザイン指向クラウドオーケストレータ CloudConductor

## ● CloudConductorとは

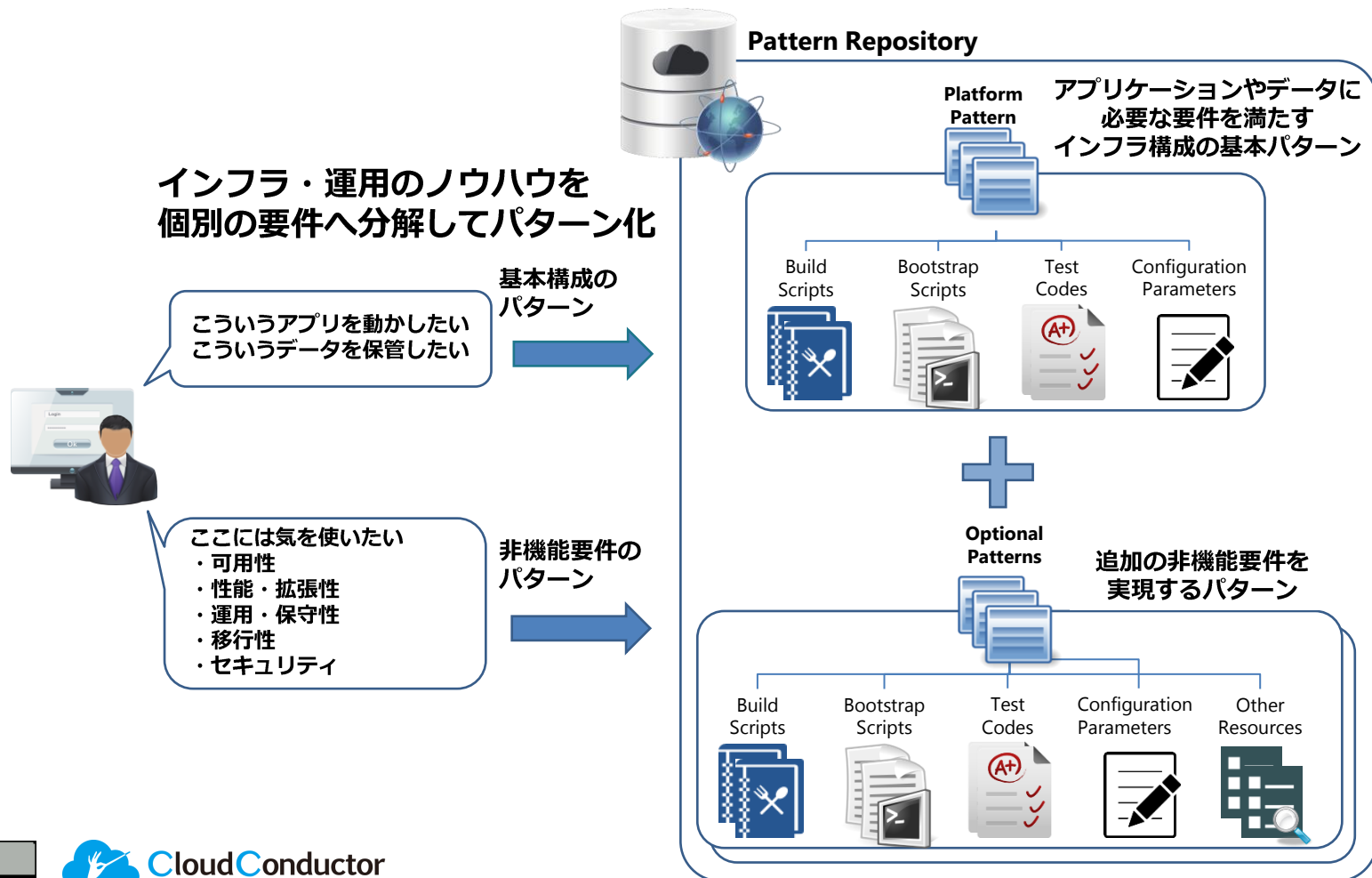
- インフラ・運用のノウハウを込めた**パターン**を中心に
- いつでも誰でもどのクラウドにでも
- その時点で最適な非機能要件を持ったシステムを

簡単に構築する、クラウドオーケストレータである



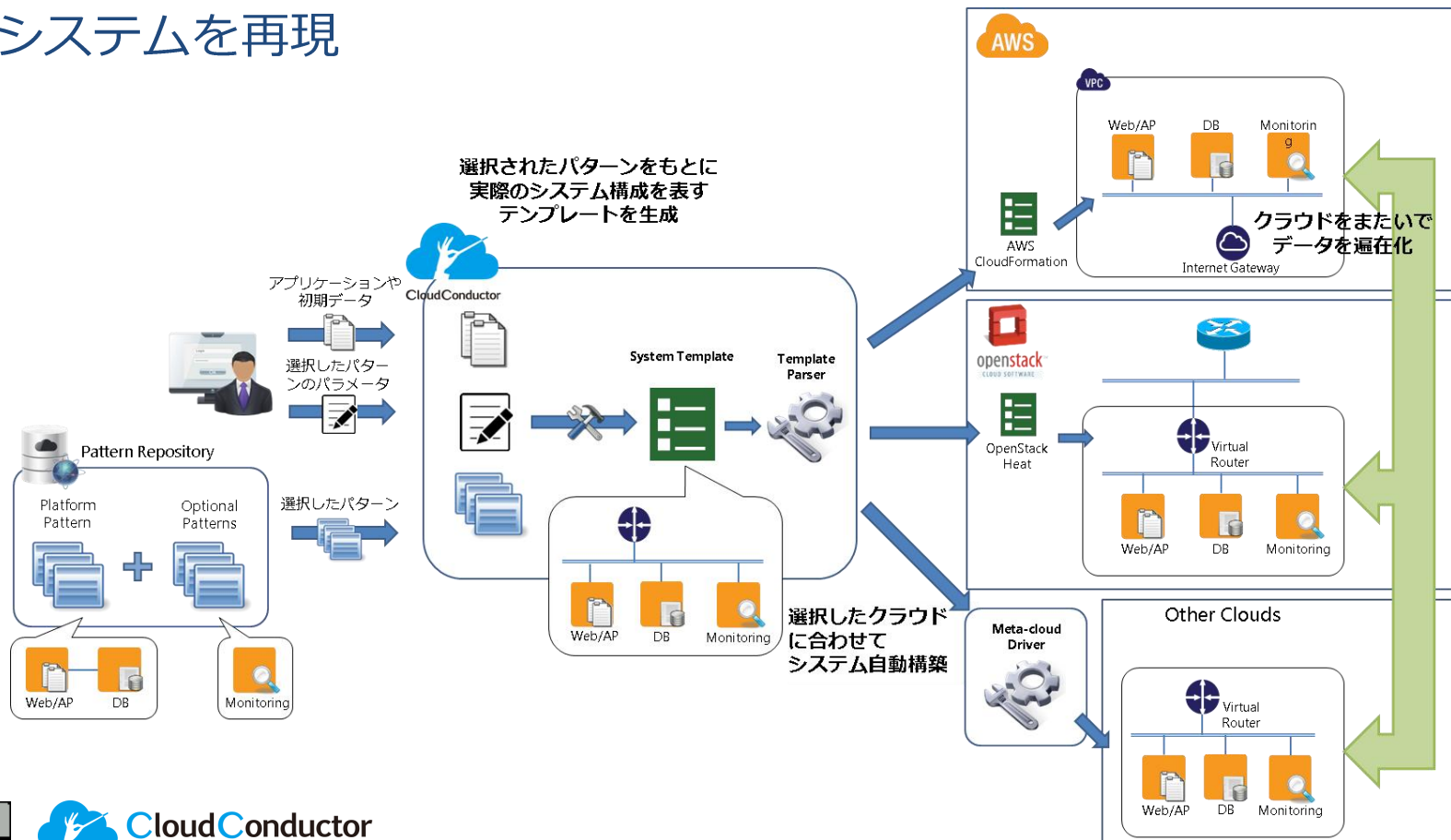
## 特徴①：Infrastructure Design Patterns as Code

- インフラ・運用の設計に関するノウハウを、個別の要件へ分解して依存関係を整理して**パターン化**
- 形式知化されたパターンを集積し集合知を形成



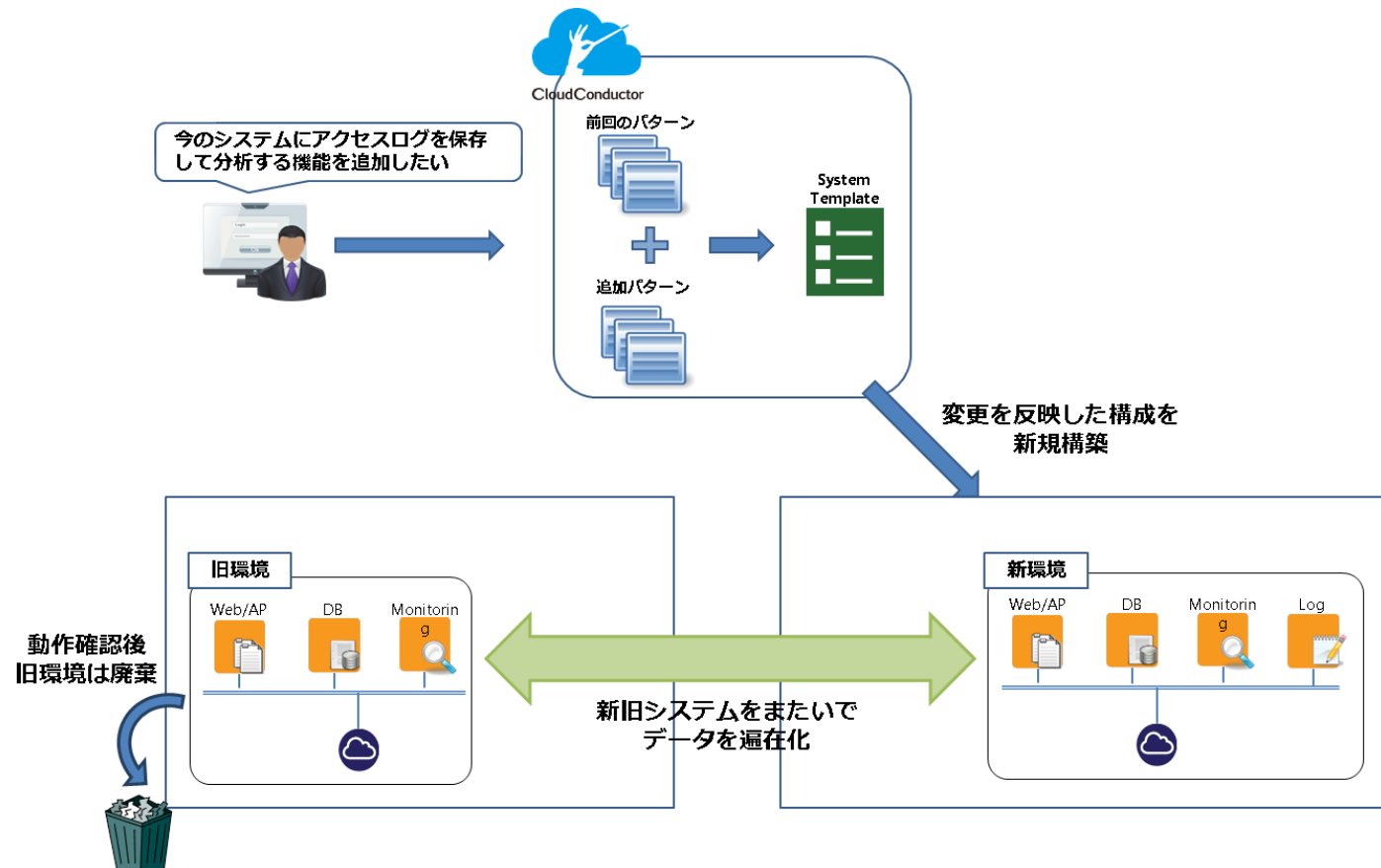
## 特徴② : Everyone, EveryTime & EveryCloud

- 必要なパターンを組み合わせ、誰でも最適なインフラ設計を獲得
- 組み合わせたパターンから各クラウドに適合したテンプレートを生成しシステムを自動構築
- クラウドを跨ってデータを遍在化させ、いつでもどのクラウドでもシステムを再現

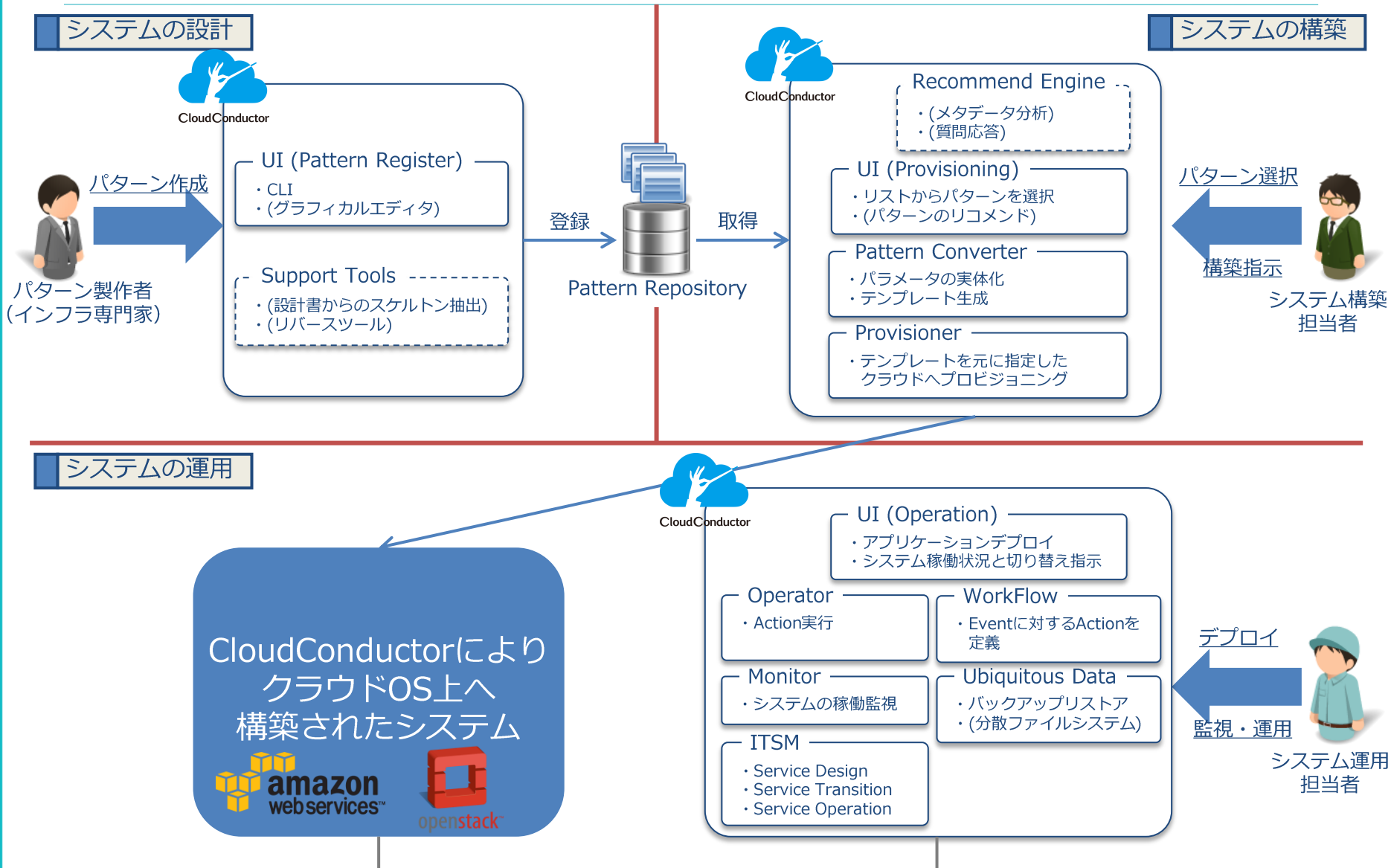


## 特徴③ : OnDemand Service Level

- 必要に応じてRASIS（信頼性、可用性、保守性、完全性、機密性）に関するパターンを個別適用
  - 「負荷分散」「可用性」「業務継続」等、必要なパターンを適用したシステムを新規に作って乗り換える



# CloudConductorの全体像



# 経済産業省の補助金事業

## ● CloudConductorの開発と公開は経済産業省の補助金事業に採択

平成26年度

中小企業等のクラウド利用による革新的省エネ化実証支援事業クラウド基盤ソフトウェア導入実証



平成25年度

産業技術実用化開発事業費補助金（ソフトウェア制御型クラウドシステム技術開発プロジェクト）



# Agenda

---

1. 加速するクラウド化
2. クラウド時代のシステムインテグレーション
3. デザイン指向クラウドオーケストレータ CloudConductor
4. CloudConductorのロードマップ
5. CloudConductorの活用例 ～災害対策のデモ～



## CloudConductorの公開

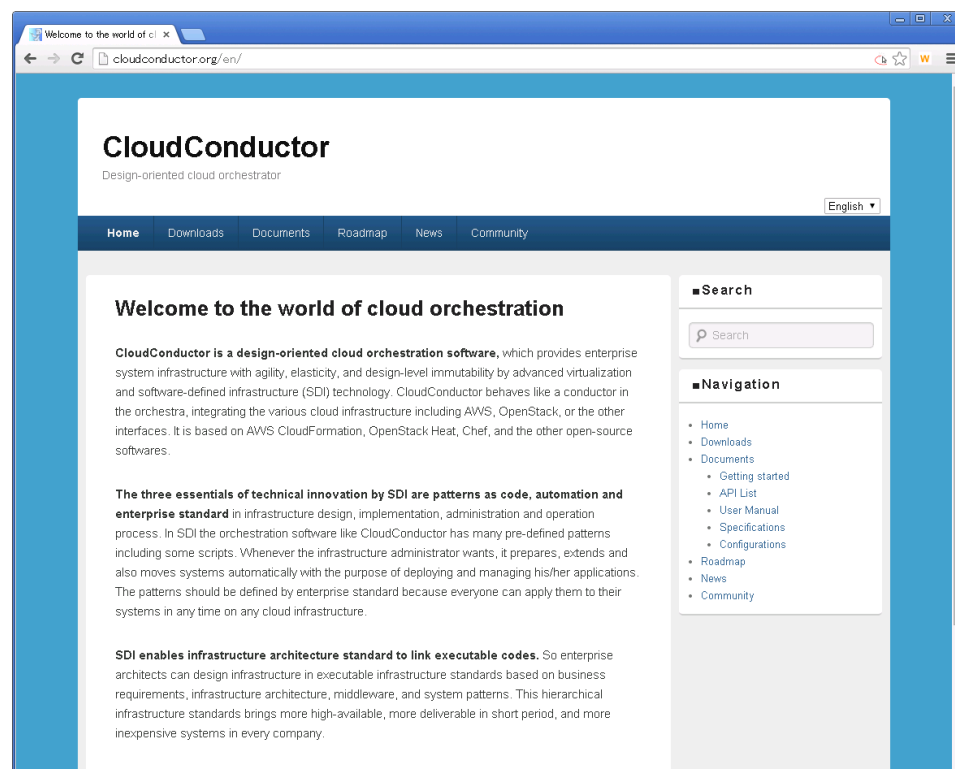
- CloudConductorはApache License 2.0で公開中



# CloudConductor

design-oriented cloud orchestrator

- 公式サイト  
<http://cloudconductor.org>
- Github  
cloudconductor
- Twitter  
@ccndctr
- Facebook  
CloudConductor





# ロードマップ



## Architecture

- システム構築のアーキテクチャを再設計
  - AWS CloudFormation, OpenStack Heatを活用したプロビジョニングへ移行
- パターン仕様の再設計
  - Platform PatternとOptional Pattern

## System Management

- 外部DNSサービスとの連携
  - システムの再構築に伴うIPアドレスの変動に追従 (Amazon Route53, Dynamic DNS)
- 外部監視システムとの連携
  - 構築したシステムの基本的な監視項目を登録 (Zabbixに対応)
- SSHコンソール
  - CloudConductorを通して構築した各インスタンスのSSHコンソールを管理可能に

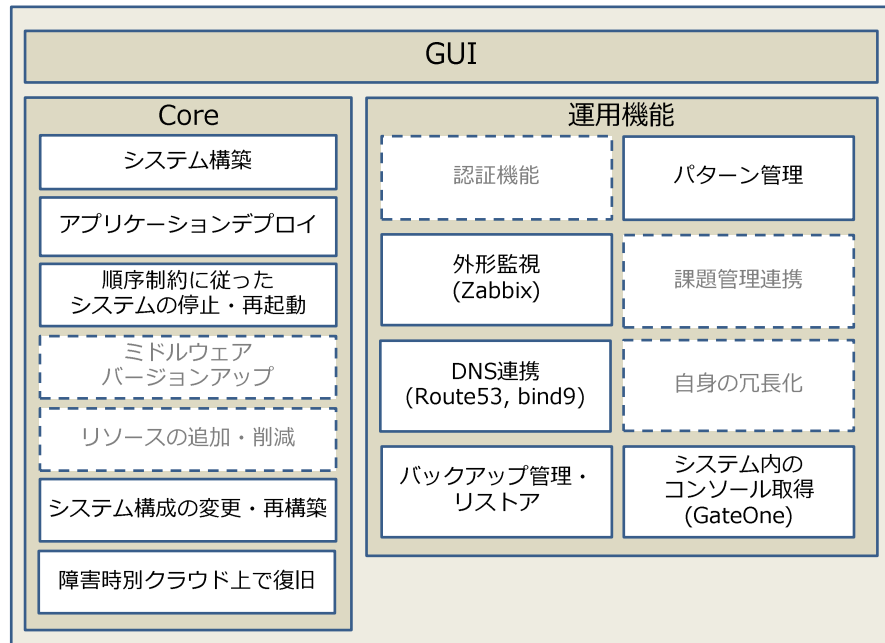
## Patterns

- 以下のPlatform Patternを提供
  - Java Application with PostgreSQL (Apache, Tomcat, PostgreSQL)
  - Rails Application with MySQL (Nginx, Unicorn/Rails, MySQL)
- Platform Patternと組み合わせて利用可能な以下のOptional Patternを提供
  - Monitoring (Zabbix)

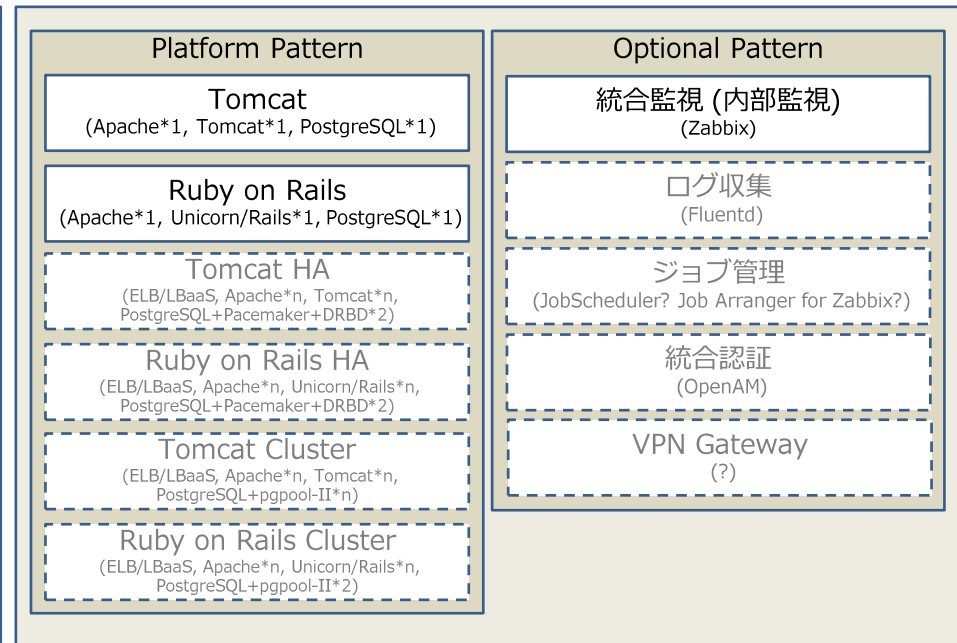
コアとなるシステム構築機能の  
整備を中心に活動予定

# Version0.3でのCloudConductorの機能とサンプルパターン（予定）

## CloudConductor



## Patterns



# Agenda

---

1. 加速するクラウド化
2. クラウド時代のシステムインテグレーション
3. デザイン指向クラウドオーケストレータ CloudConductor
4. CloudConductorのロードマップ
5. CloudConductorの活用例 ～災害対策のデモ～

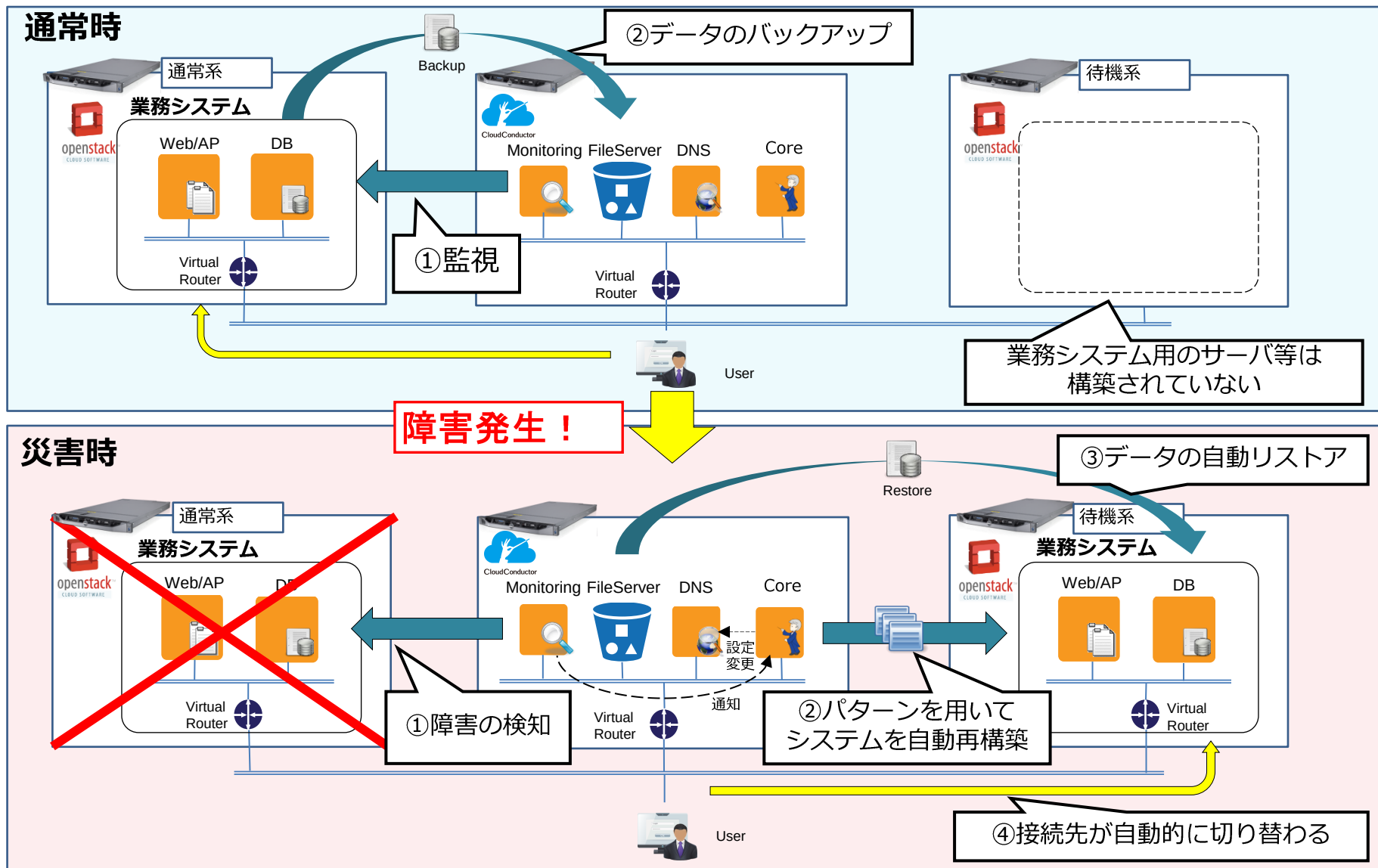


## 災害対策デモの想定要件

- 想定要件

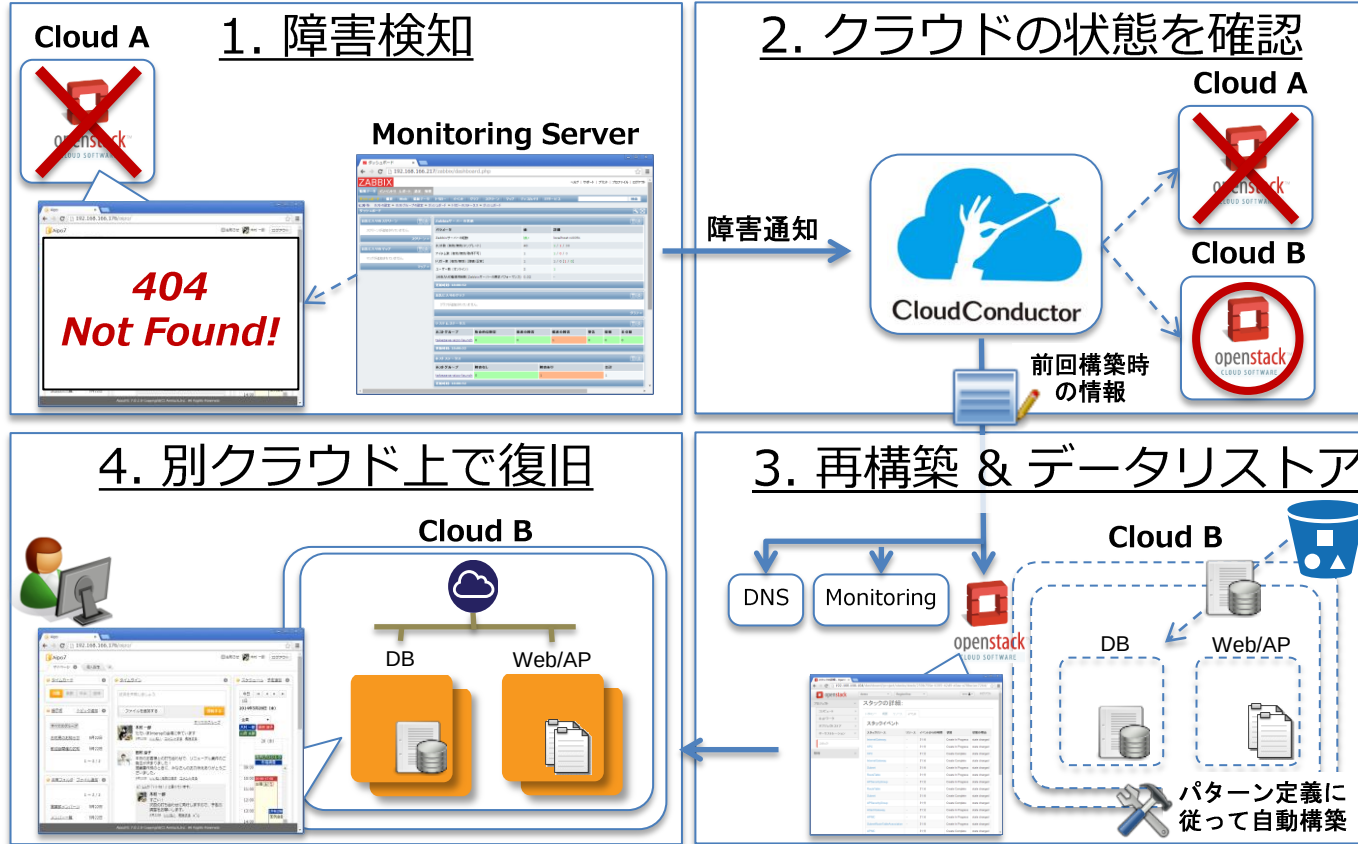
- 災害時にシステムがダウンしても、**現実的な目標復旧時点（RPO） / 目標復旧時間（RTO）でサービスを復旧（バックアップ・リストア）**する。
  1. サービス監視やバックアップ・リストア手順も含め、システム全体をパターン化
  2. パターンより通常系システムを構築
  3. 災害発生時には、被災していないクラウドへパターンを用いて自動的に災対システムを立ち上げ、データも自動的にリストア
  4. DNSを自動的に切り替え、通常系と同じURLでサービスを継続
- サービス継続に要する**金額的・電力的・人的コストを最小限**にする。
  - ✓ 災害対策用のシステムをホットスタンバイさせず、災害発生時に自動構築することで、無駄なリソース消費・電力消費・費用を抑制
  - ✓ 通常系も災対系も同じパターンから自動構築することで、災対系の維持運用に手間が取られることが無い
  - ✓ 災対系のURLを覚えておくなど、利用者に負担を書けることが無い

# 災害対策デモの全体像



# デモ実施

## 障害発生時の動作



- デモ動画はYouTubeにも公開しています
- <https://www.youtube.com/user/ccndctr>
- ✓ <https://www.youtube.com/watch?v=Im81YXeUhC8>

