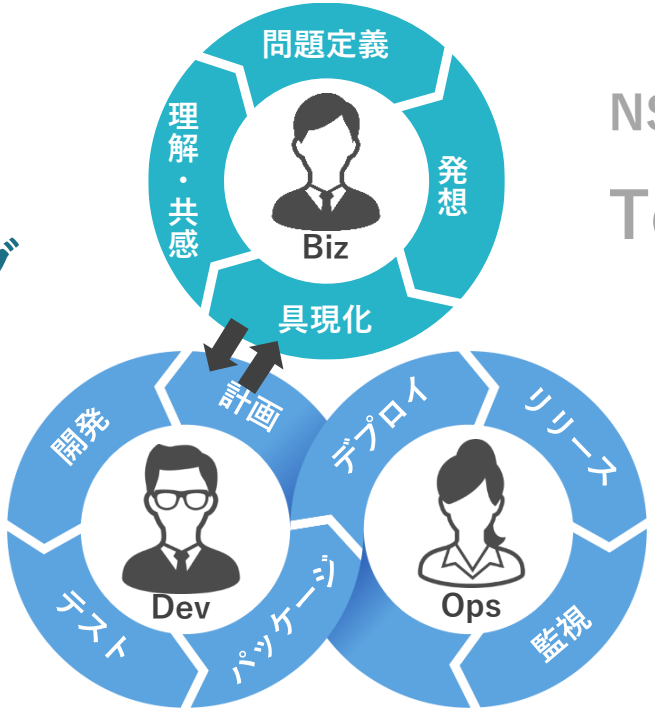
新技術を活用した
システム開発・運用手法の研究

技術本部 システム研究開発センター
若林 耕平
DX&イノベーションセンター
伊藤 宏樹

本セッションの位置付け

システムのアジリティと
信頼性を両立する

マイクロサービス・
クラウドネイティブ



NSSOL開発DXの実現

Tetralink

アジェンダ

1 アジリティを高める新技術の実用化

2 デザインパターンを活用したDXの推進

3 DXのさらなる推進に向けて

1

アジリティを高める新技術の実用化

システム研究開発センターのこれまでの取り組み

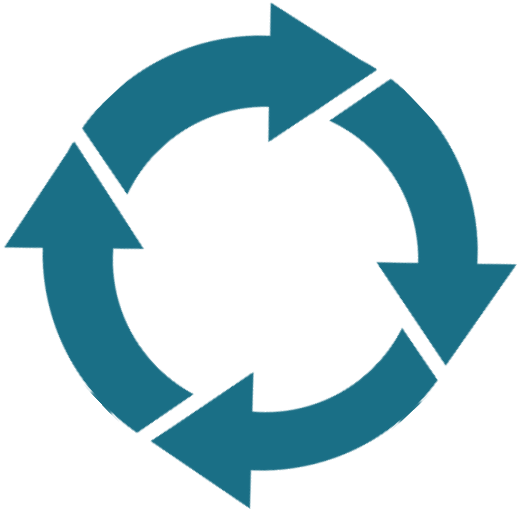
ビジネスが求めるシステムのアジリティ

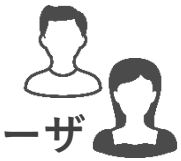
ユーザ評価に基づくサービス改善やプラットフォーマが求める早期のシェア拡大など

アイデアの創出

システム開発

サービス提供

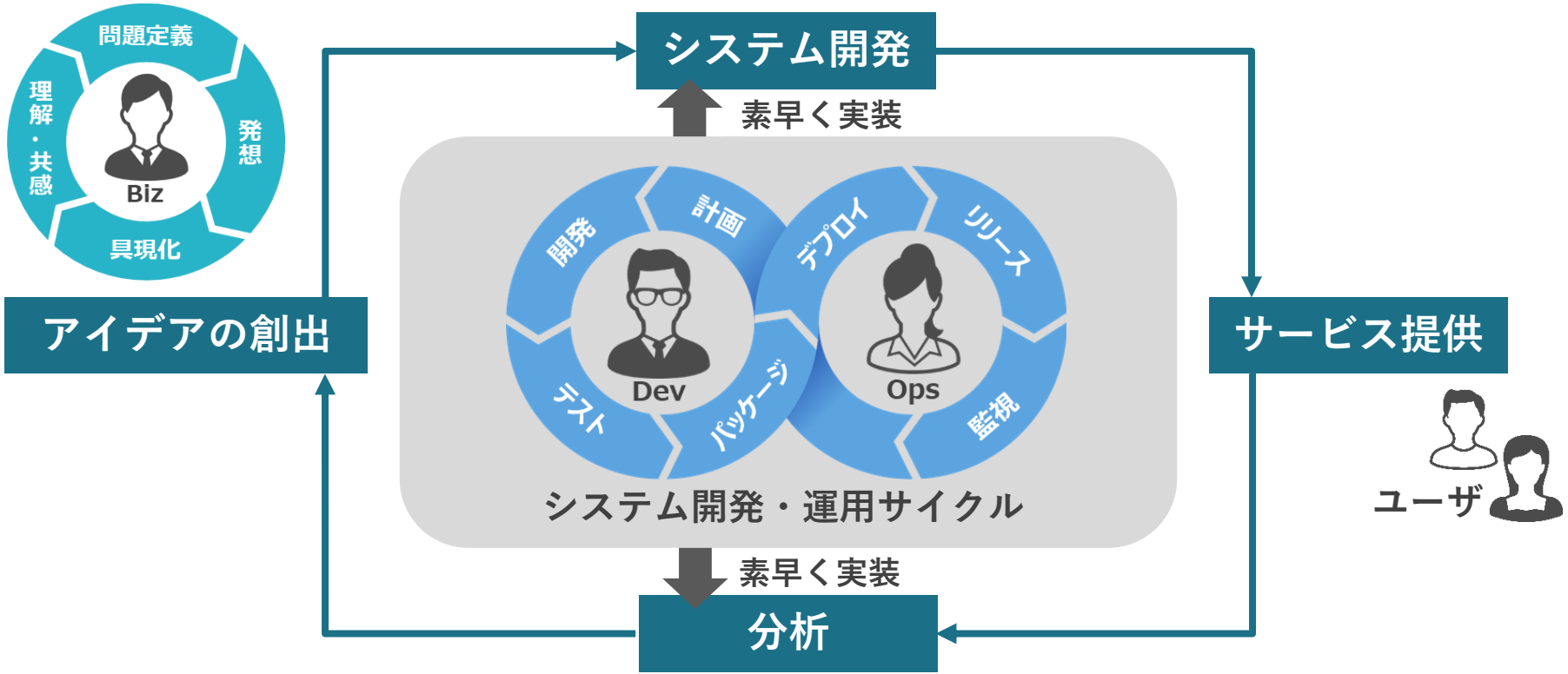




分析

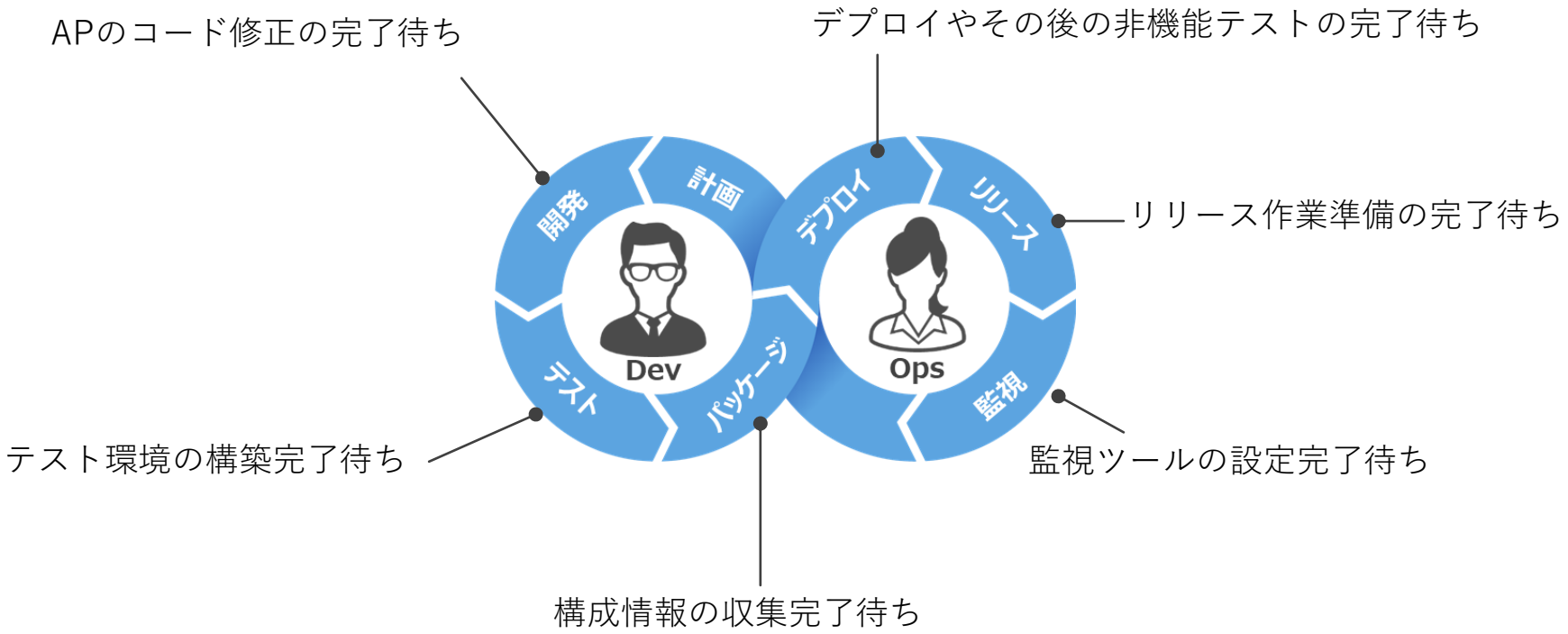
ビジネスが求めるシステムのアジリティ

ユーザ評価に基づくサービス改善やプラットフォーマが求める早期のシェア拡大など



従来の手法の限界

手法を変えずにアジリティを高めようとすると、信頼性が犠牲になる

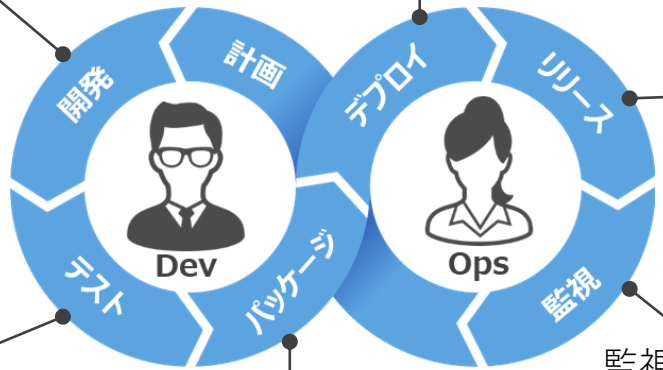


従来の手法の限界

手法を変えずにアジリティを高めようとすると、信頼性が犠牲になる

APのコード修正の完了待ち
→ **開発時間の短縮によるバグの混入**

デプロイやその後の非機能テストの完了待ち
→ **作業ミスや非機能テストの削減による品質低下**



リリース作業準備の完了待ち
→ **作業ミスによる品質低下**

テスト環境の構築完了待ち
→ **テスト項目の削減による品質低下**

監視ツールの設定完了待ち
→ **監視設定の不備による可用性の低下**

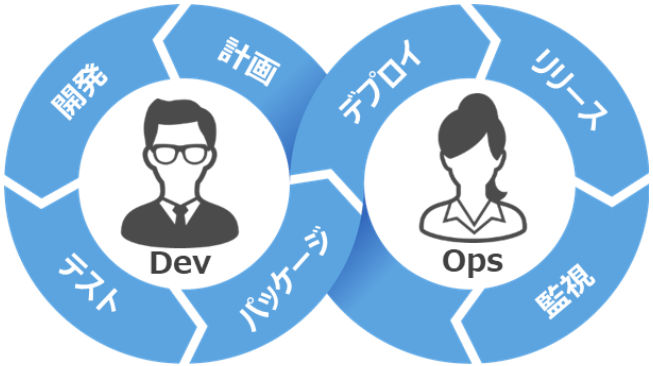
構成情報の収集完了待ち
→ **構成管理の不備による運用負荷の増大**

アジリティを高める新技術

2つの技術で信頼性を損なわずにアジリティを高める

マイクロサービス・アーキテクチャ

APを複数の小さなサービスに分割し、サービス単位での独立した開発を可能にする



クラウドネイティブ

コンテナやkubernetesを用いてシステム開発・運用業務を高度化・自動化する

アジリティを高める新技術

マイクロサービスとクラウドネイティブで従来の手法の限界を超える

複数のマイクロサービスに分割し、

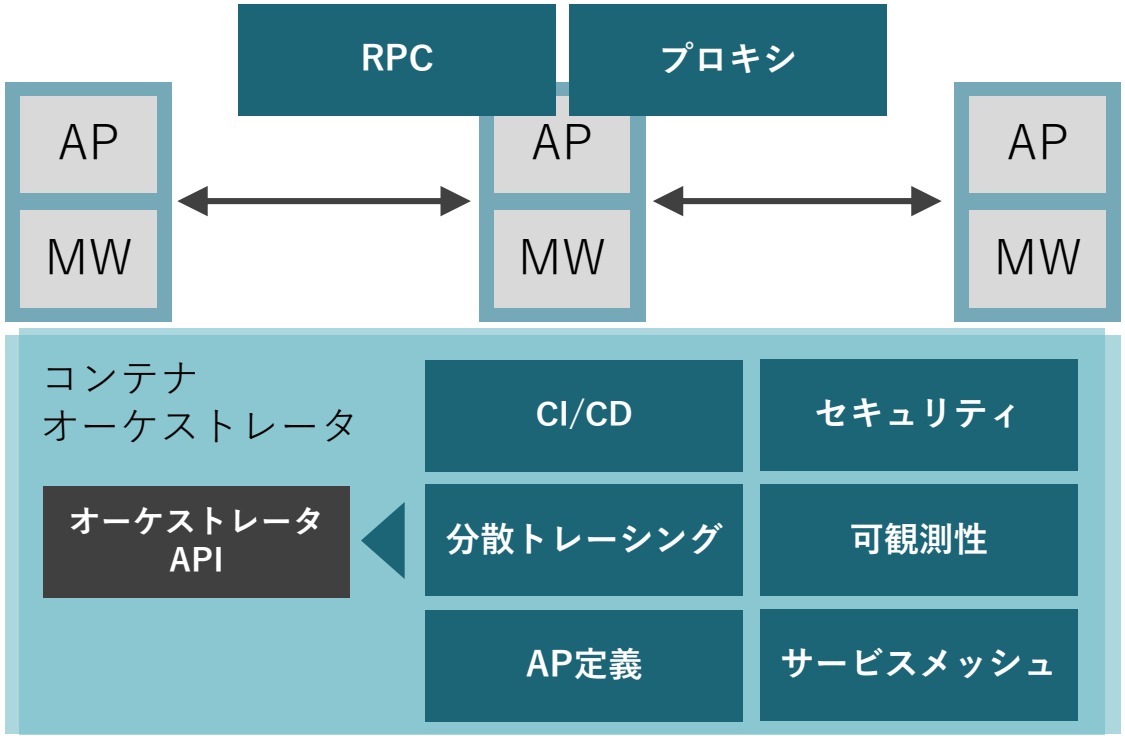
アジリティを高める新技術

マイクロサービスとクラウドネイティブで従来の手法の限界を超える

オーケストレータで管理する

コンテナ技術がオープンなコミュニティで急速に進化。

マイクロサービスの開発・運用を支える多数のツール・サービスを提供。



アジリティを高める新技術

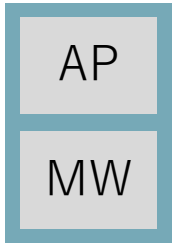
マイクロサービスとクラウドネイティブで従来の手法の限界を超える

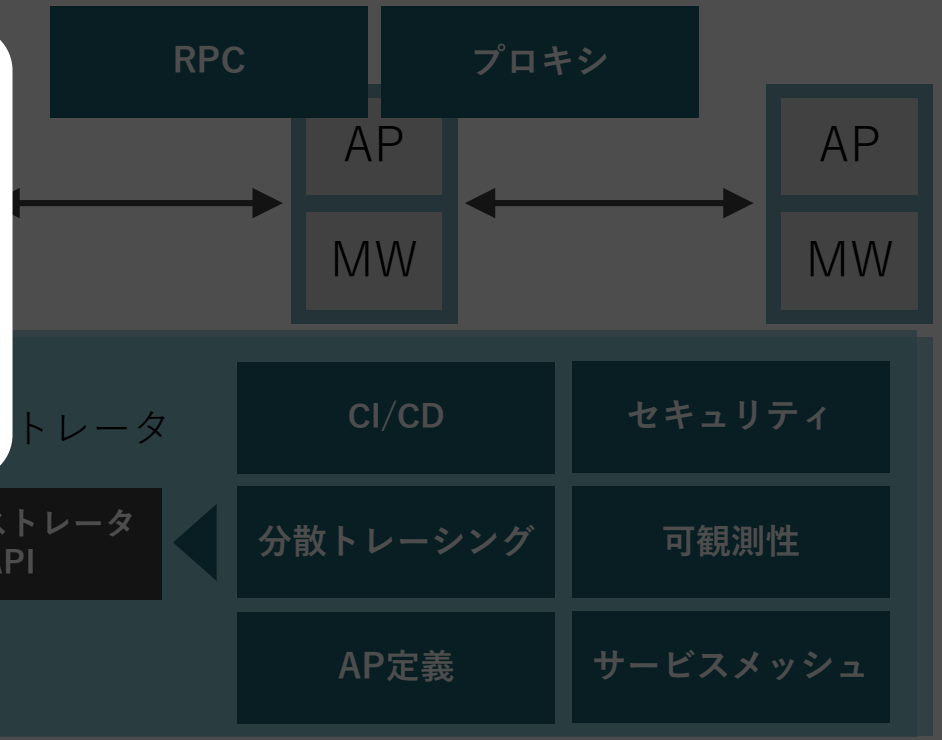
オーケストレータで管理する

・開発時の検討対象を絞り込める
・サービス単位でデプロイできる

従来の複雑さやデプロイの
問題に対処

Dev





アジリティを高める新技術

マイクロサービスとクラウドネイティブで従来の手法の限界を超える

オーケストレータで管理する

コンテナ技術がオープンなコミュニティで急速に進化。

マイクロサービスの開発・運用を支える多数のツール・サービスを提供。

RPC

プロキシ

・業界標準を効率的に活用できる
・各種作業を高度に自動化できる

Ops

従来の「手作業」の問題に対処

CI/CD

セキュリティ

分散トレーシング

可観測性

AP定義

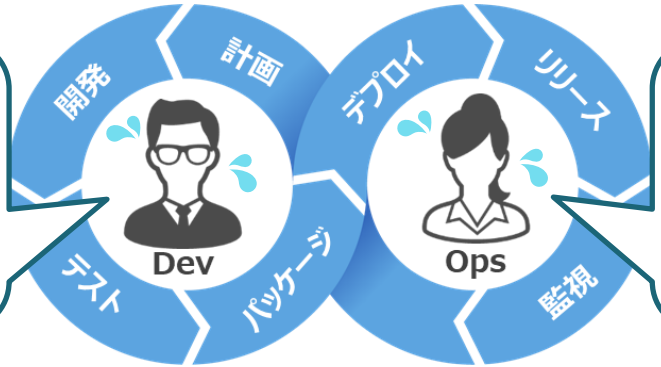
サービスメッシュ

新技術の難しさ

新技術に伴う様々な問題を解決しなければ、むしろ逆効果になる

サービスの分割や連携は？
（新手法の学習の問題）

開発者が信頼性も考えるの？
（業務分担の見直しの問題）



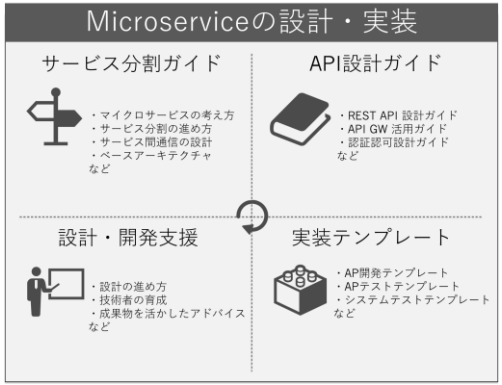
大量かつ進歩の早い技術に迷う…
（新技術への追従の問題）

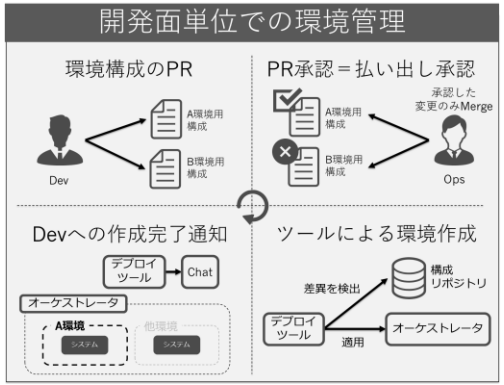
複雑なシステムの障害対応が困難
（システム構成の複雑化の問題）

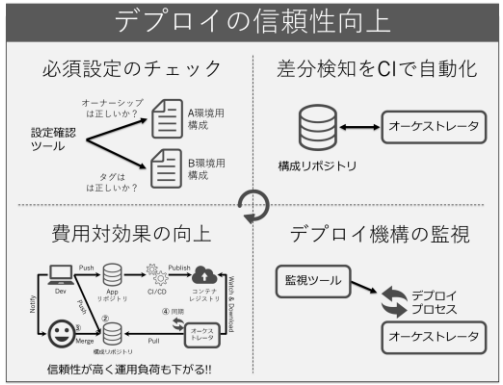
アジリティと信頼性の低下

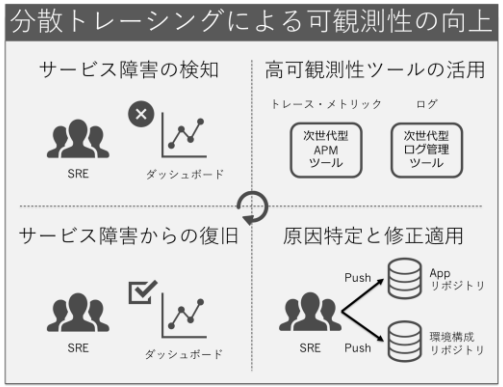
その難しさを解消する方法

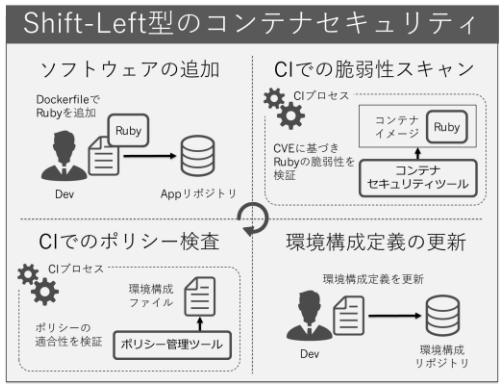
新技術を活用したシステム開発・運用手法をデザインパターンとして整理

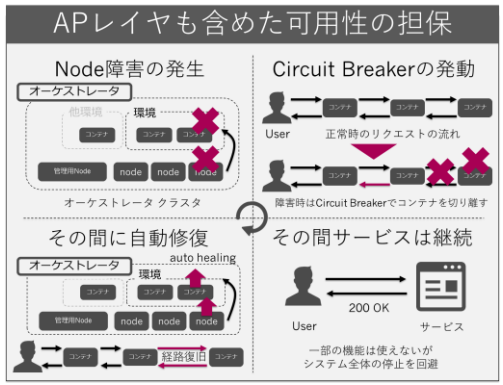






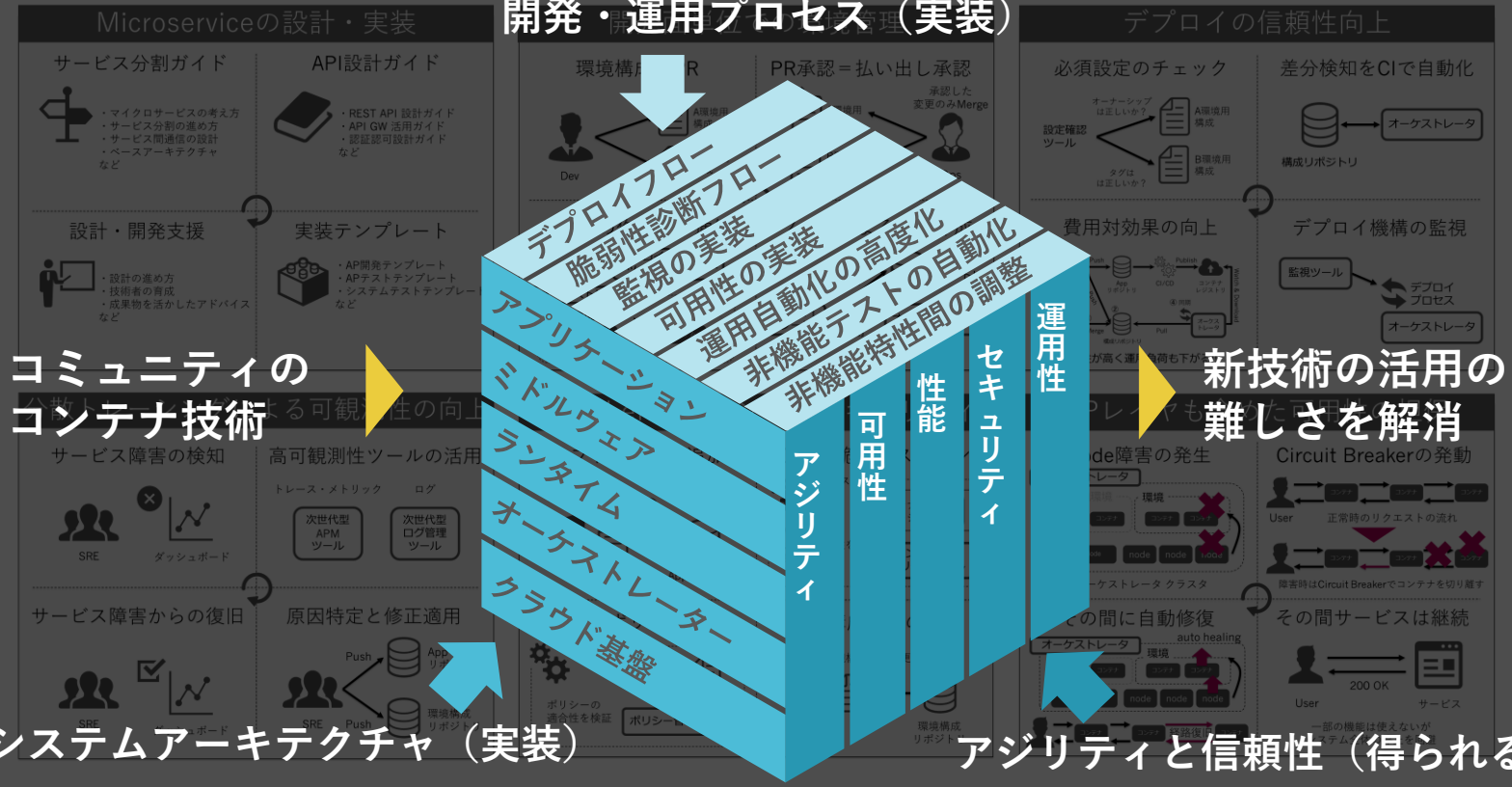






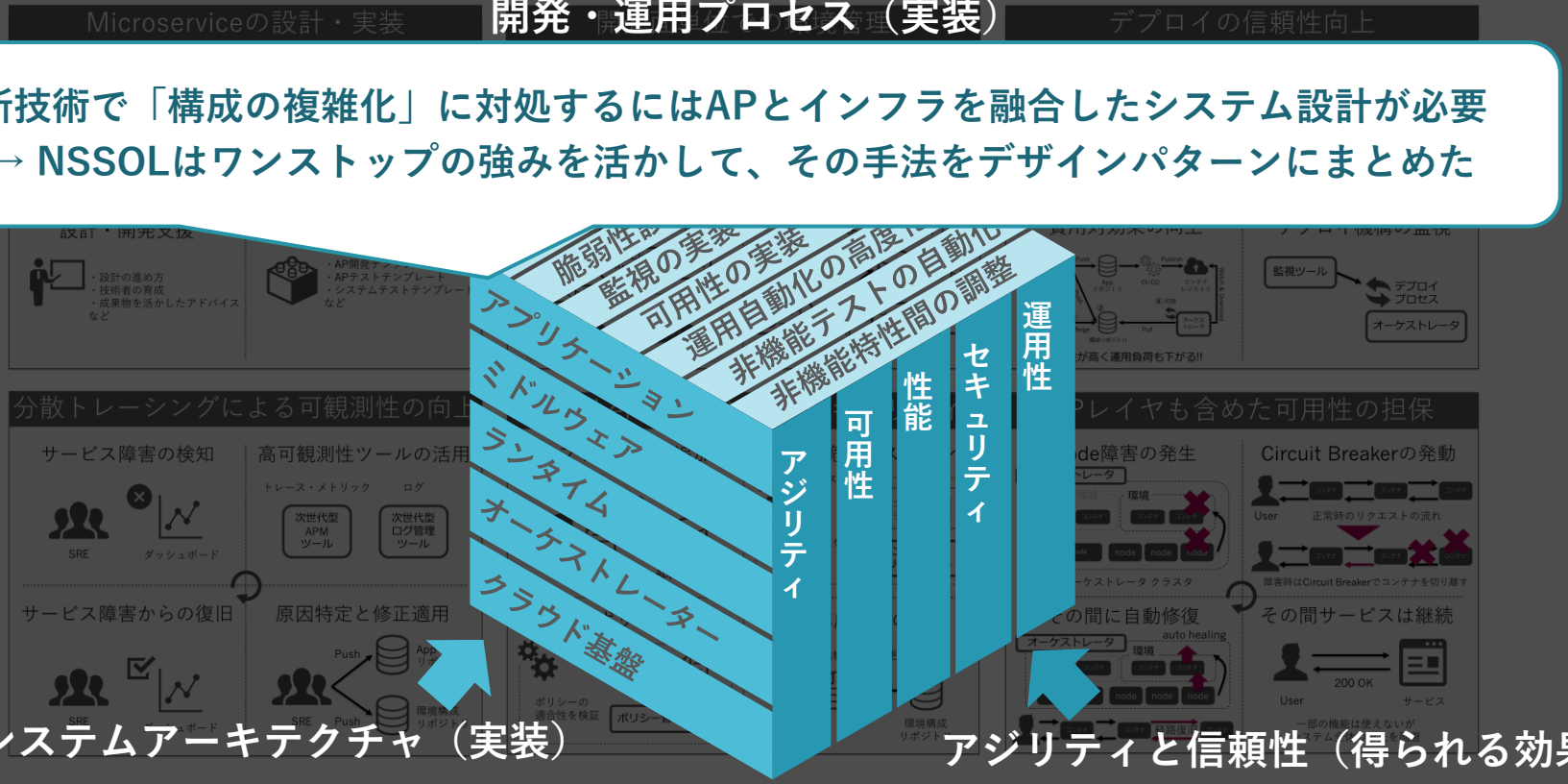
その難しさを解消する方法

新技術を活用したシステム開発・運用手法をデザインパターンとして整理



その難しさを解消する方法

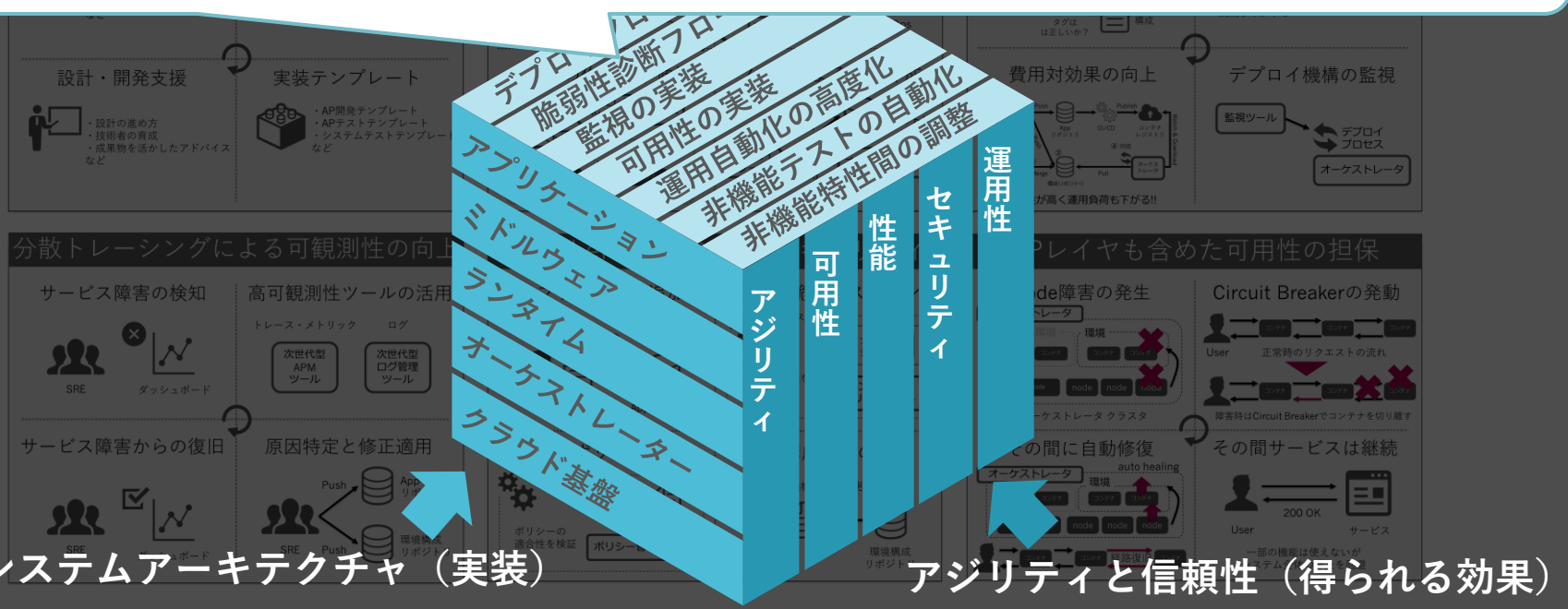
新技術を活用したシステム開発・運用手法をデザインパターンとして整理



その難しさを解消する方法

新技術を活用したシステム開発・運用手法をデザインパターンとして整理

新技術で「業務分担」の問題に対処するには開発と運用を融合した新しい業務プロセスが必要
→ NSSOLはワンストップの強みを活かして、その手法をデザインパターンにまとめた

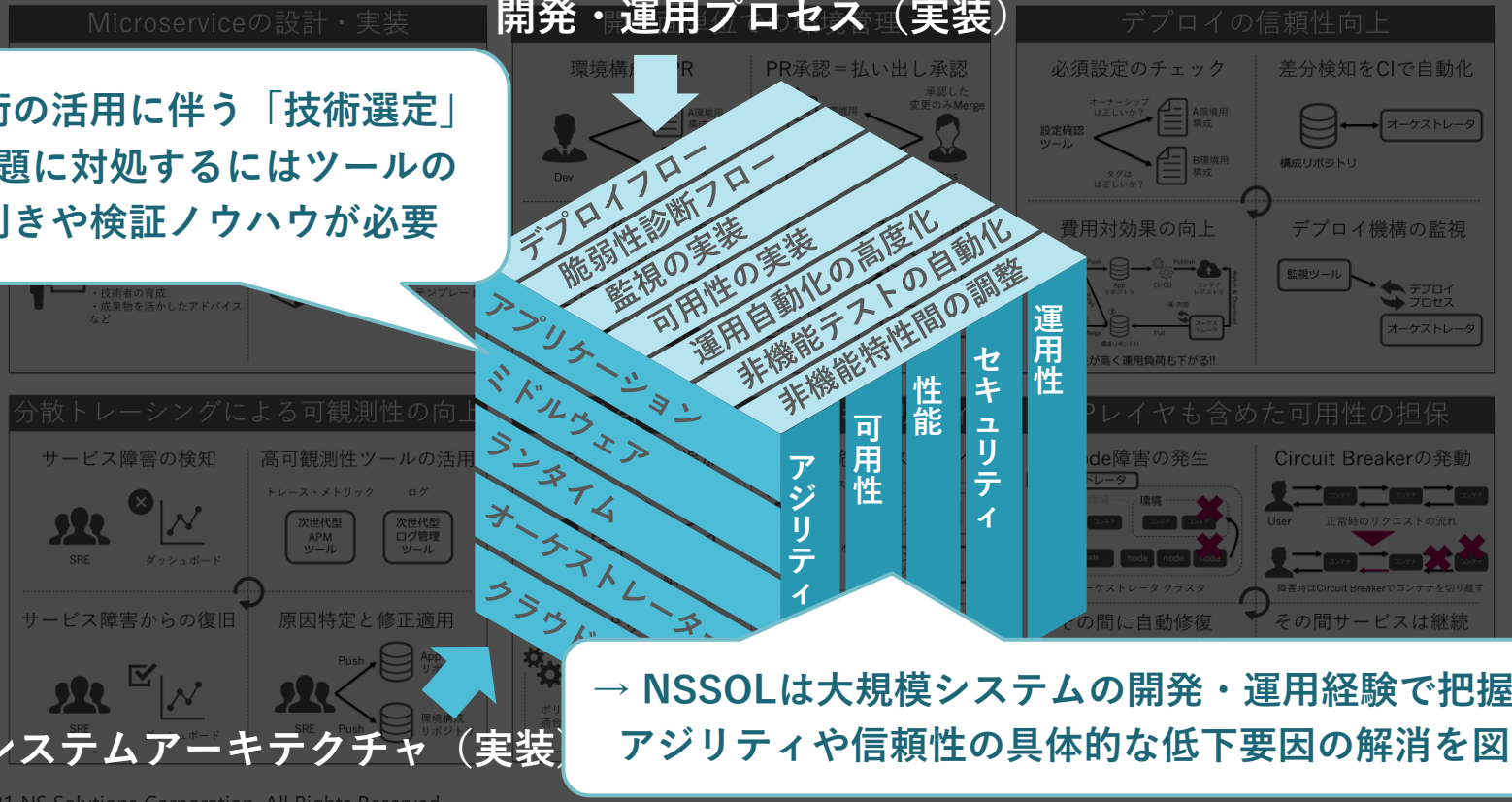


システムアーキテクチャ（実装）

アジリティと信頼性（得られる効果）

その難しさを解消する方法

新技術を活用したシステム開発・運用手法をデザインパターンとして整理



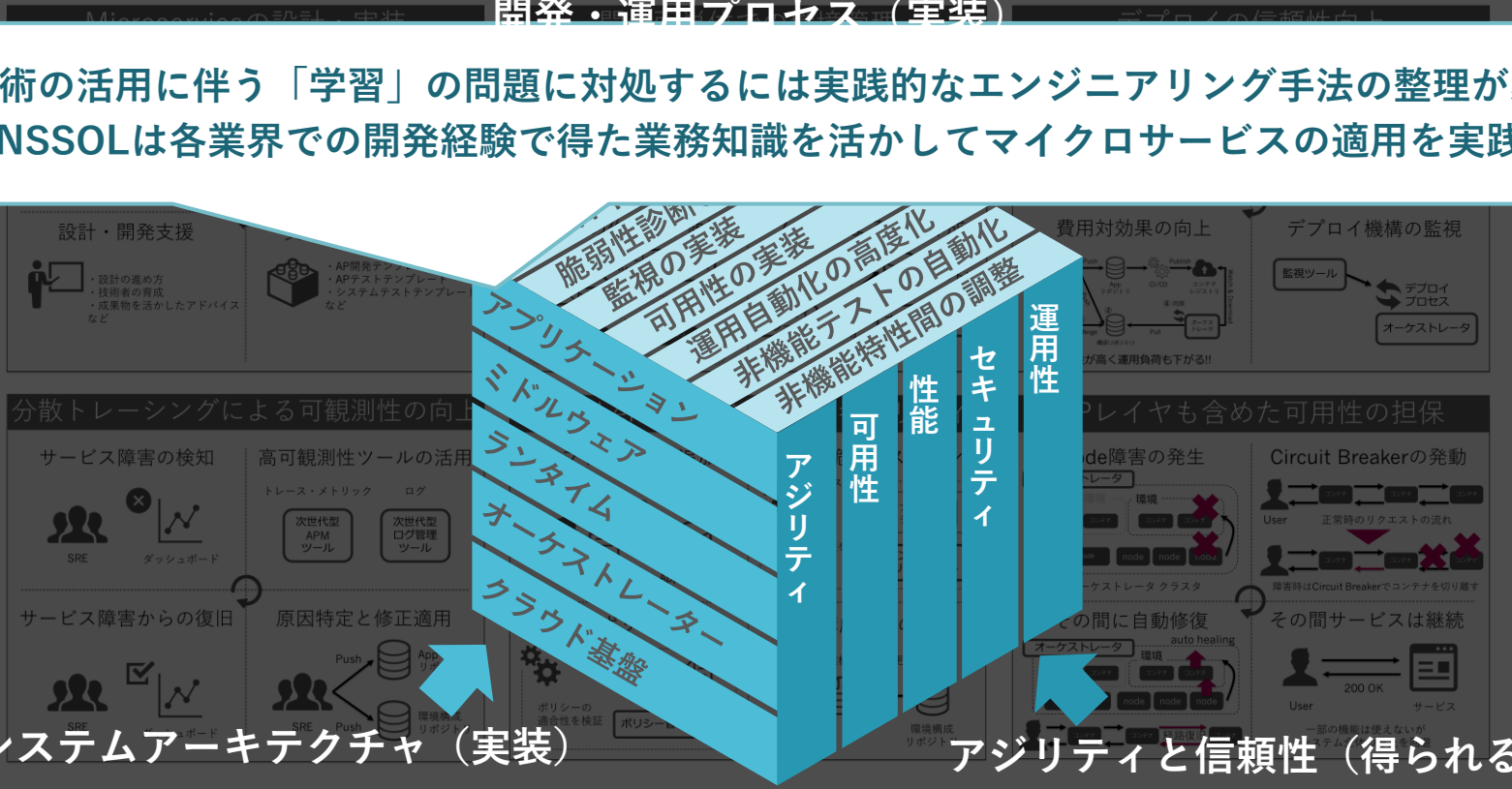
新技術の活用に伴う「技術選定」の問題に対処するにはツールの目利きや検証ノウハウが必要

→ NSSOLは大規模システムの開発・運用経験で把握したアジリティや信頼性の具体的な低下要因の解消を図れる

その難しさを解消する方法

新技術を活用したシステム開発・運用手法をデザインパターンとして整理

新技術の活用に伴う「学習」の問題に対処するには実践的なエンジニアリング手法の整理が必要
→ NSSOLは各業界での開発経験で得た業務知識を活かしてマイクロサービスの適用を実践中

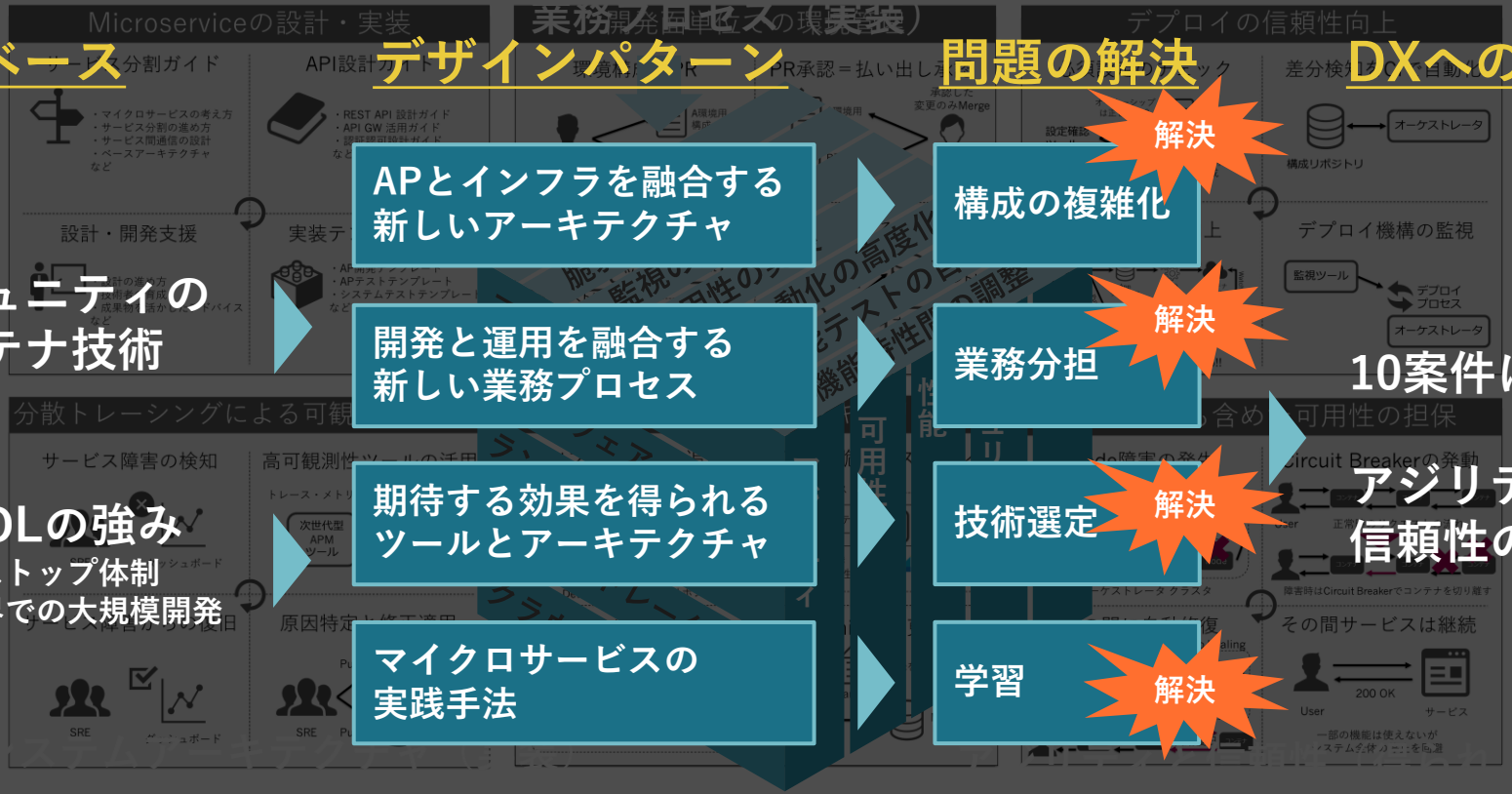


システムアーキテクチャ（実装）

アジリティと信頼性（得られる効果）

その難しさを解消する方法

デザインパターンでアジリティと信頼性に優れたシステムを効率的に実現する



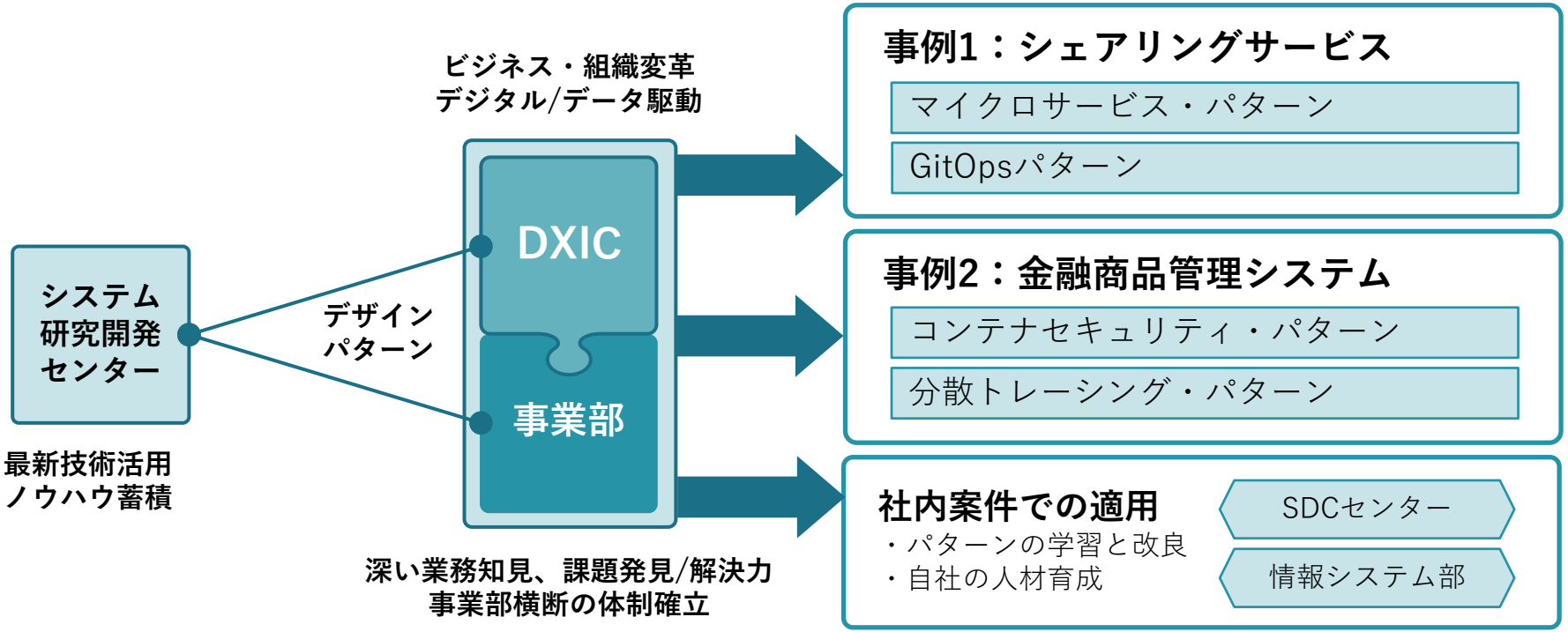
2

デザインパターンを活用したDXの推進

DXICのDX案件事例

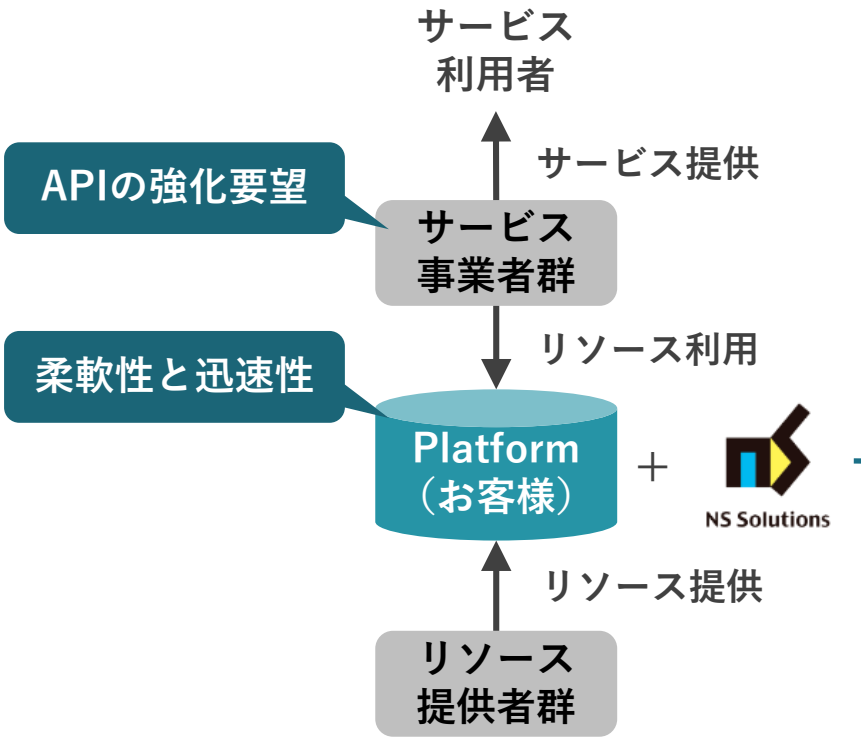
新技術を活用したDXの推進

デザインパターンを習得したDXICが事業部横断の体制を技術でリード



事例1：シェアリングサービス

お客様のビジネスが求めるシステムの柔軟性と迅速性を新技術で実現



事業の方針をお客様から共有して頂く過程で
「今まで以上の柔軟性や迅速性」が必要であり
「社会インフラとしての信頼性」も担保すべき
ことをNSSOLは強く認識。

新技術の可能性とリスクを評価した上で
以下のパターンをお客様に提案。

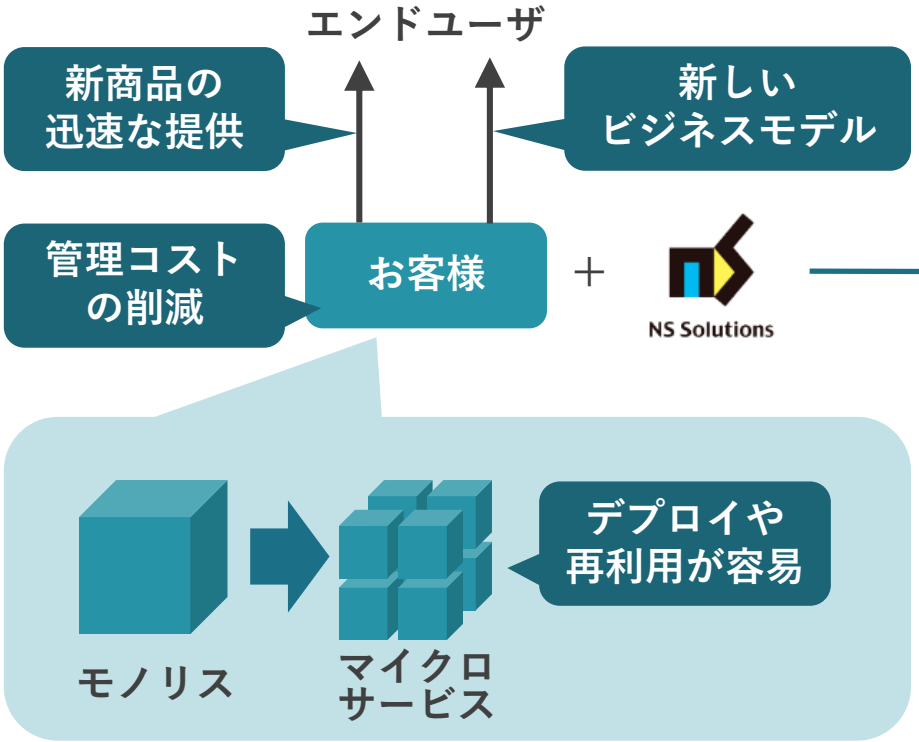
＜採用したデザインパターンの例＞

マイクロサービス・パターン
指針やテンプレートによる設計の効率化と品質向上

GitOpsパターン
デプロイのセルフサービス化によるアジリティ向上

事例2：金融商品管理システム

アジリティに加えて、システムの可用性やセキュリティも追求



お客様の業務の理解を進める過程で、
アジリティだけでなく、業務の継続性を
高めるシステムの信頼性も重要と認識。

マイクロサービスの信頼性を高める
以下のパターンを提案。

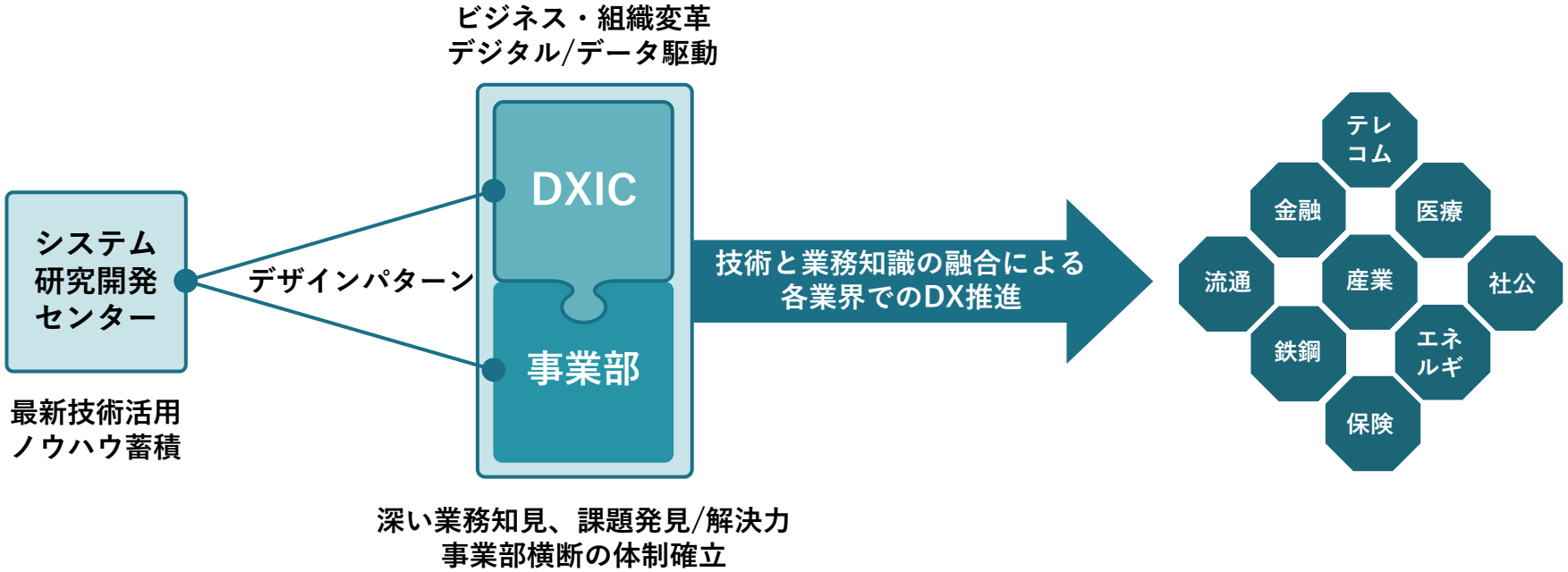
＜採用したデザインパターンの例＞

コンテナセキュリティ・パターン
脆弱性診断のDev向けセルフサービス化

分散トレーシング・パターン
システム状態の可視化による障害対応力の向上

新技術を活用したDXの推進

デザインパターンを習得したDXICが事業部横断の体制を技術でリード



現場からのニーズ

DXに不可欠な新技術をデザインパターンを通して継続的に担保する

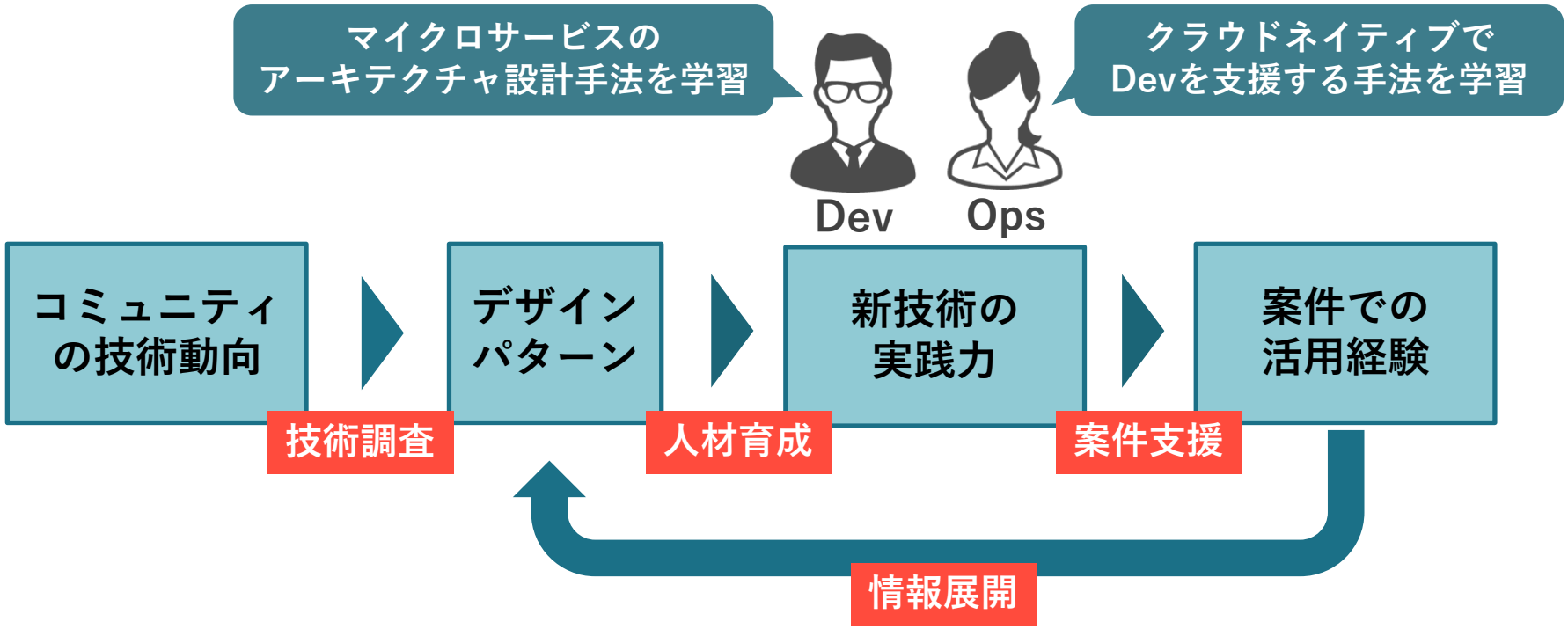
今後の動き	現場のニーズ
コミュニティによる 継続的な技術革新	デザインパターンの 継続的な開発
お客様の DXの取り組み強化	デザインパターンを 活用できる人材の育成

3 DXのさらなる推進に向けて

システム研究開発センターの今後の活動方針

現場のニーズに継続的に対応

デザインパターンを媒体にした技術力の強化を推進する



商標について

X Integrator、 hifive、 Geminant、Geminant（ロゴ）、安全見守りくん、IoX、IoXプラットフォーム、匿丸、匿丸（ロゴ）、OptBrain、OptBrain（ロゴ）、Tetralink、Tetralink（ロゴ）、日鉄ソリューションズ、NSSOL、NS(ロゴ)は日鉄ソリューションズ株式会社の登録商標です。

その他本文記載の会社名及び製品名は、それぞれ各社の商標又は登録商標です。