IIJ Technical WEEK 2017
予兆監視 適用ステップと解析プラットフォーム

クラウド本部クラウドサービス2部
ビッグデータ技術課
高舘 洋介

Ongoing Innovation

自己紹介

■ クラウド本部クラウドサービス2部ビッグデータ技術課
高舘 洋介 <takadate@iij.ad.jp>

社歴

- 2010～2011年 システムインテグレーション
- 2012～2016年 ビッグデータソリューション/SI
- 2017年～ 「IIJ IoTサービス」開発

セッション概要

1. **異常検知(Anomaly Detection)の仕組みを活用した監視システムに関する概説**

2. **統計解析/機械学習の理論ではなく、予兆監視システムの基本的な仕組みやシステムアーキテクチャ、課題に対する取り組みなどの情報**

※ 本セッションの内容は、IIJサービス設備の検証・取り組み、案件事例 等の情報を統合して記載しています

目次

1. 予兆監視 適用ステップ

- 異常検知の概説、検知方法に関する基本的な仕組み、課題に対する取り組み、など

2. 解析プラットフォーム

- 予兆監視機能を提供するためのシステムアーキテクチャ

3. IoT×予兆監視システム

- IoTセンサーデータを対象にした予兆監視システムの構成例

1. 予兆監視 適用ステップ

異常検知（Anomaly Detection）

■ 正常ではない状態を見つけ出す

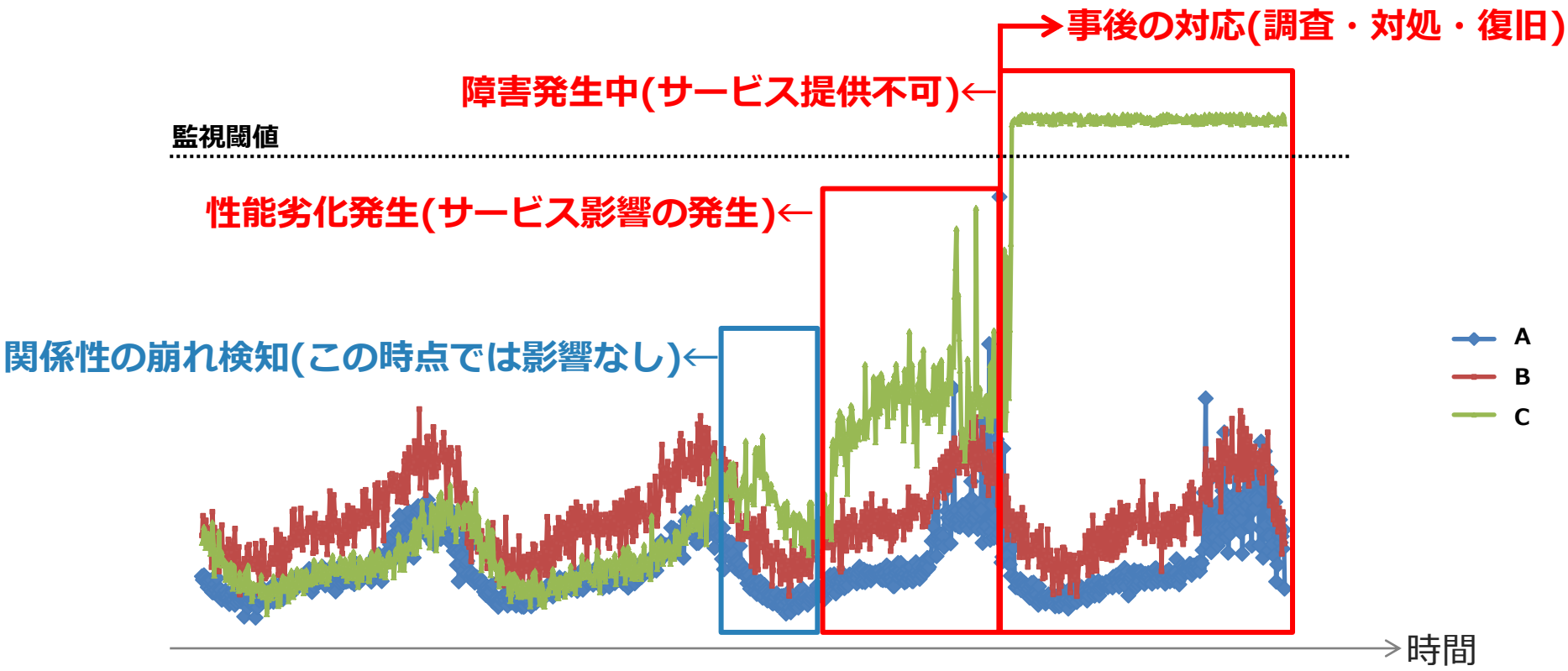
- 何が異常かが予め分かっているのであればそれを検出する
  - これまでの一般的な監視システム（生死監視・リソース閾値監視）

- これから起きる異常が何かをわかっていない状態から、正常時との「いつもとの違い」を検出し、事前に対処する

- この手法によって、大規模で複雑なシステムから、未知の異常を捉えられる可能性がある

予兆監視イメージ

■ 相関関係のある3つのデータ



障害が発生する前段階で、データの関係性の変化を「障害の予兆」として捉え、対処することで、サービス影響と障害を回避することができる

予兆監視の効果

■ 異常検知

- 実障害に至る前の異常の兆候を検知し、事後の対応→未然の防止へ
- 閾値や目視の監視では発見できない、サービス影響を及ぼしている状態（サイレント障害）を捉える

⇒システムの安定稼働、サービス品質向上、機会損失防止

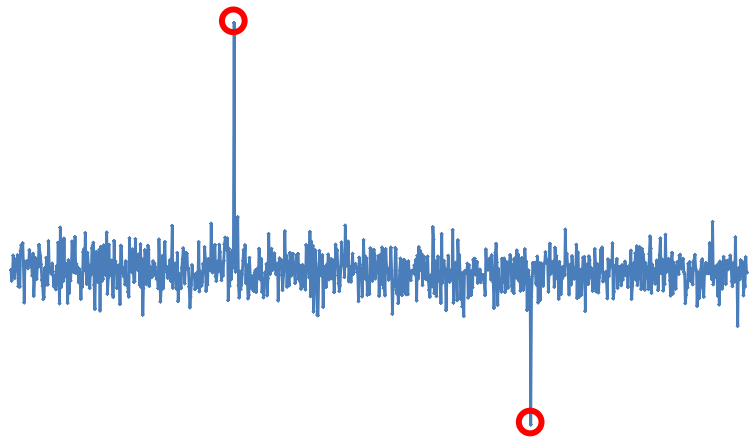
■ 運用リソース効率化

- 大量機器に対する手動での監視設定を縮小
- システムを熟知したエンジニアでなければ対応できない運用の縮小

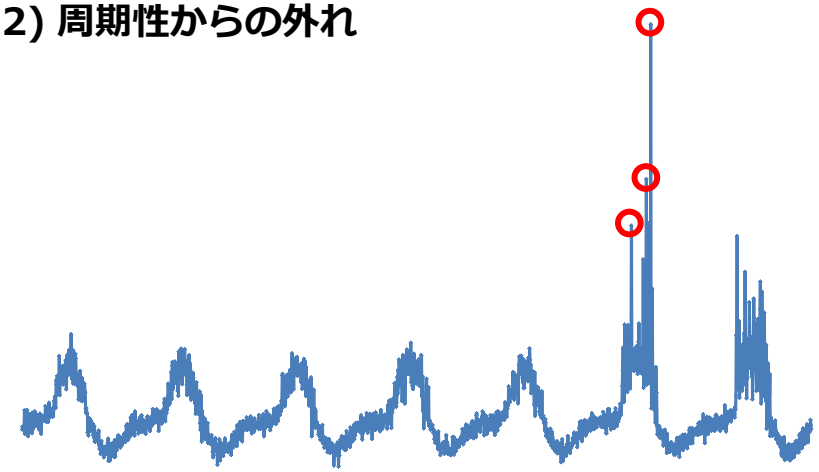
時系列データの異常検知

■ 検知パターン

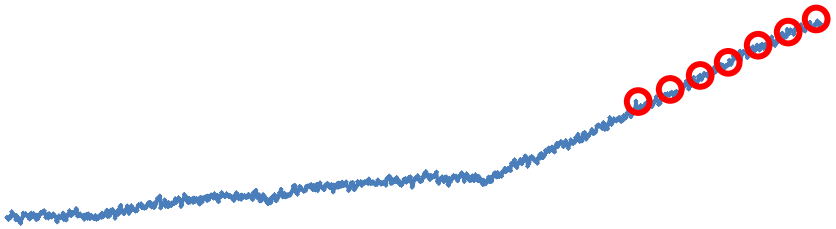
(1) 出現頻度から検出



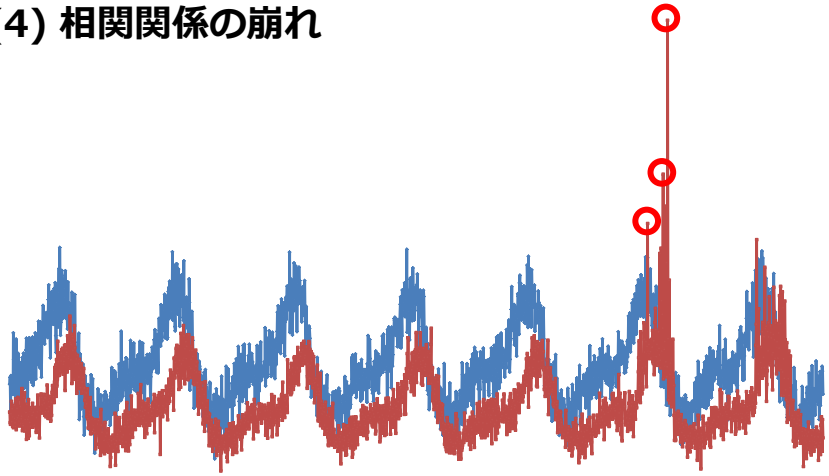
(2) 周期性からの外れ



(3) 上昇・下降傾向の変化



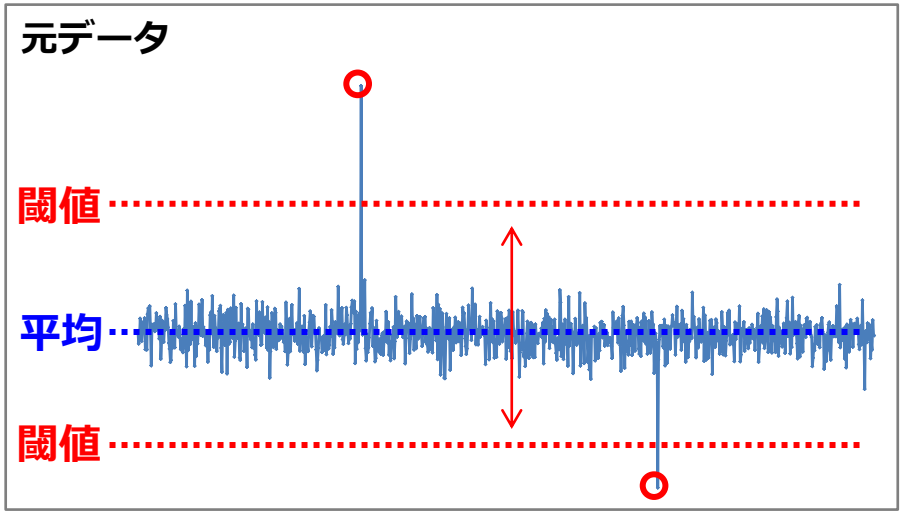
(4) 相関関係の崩れ

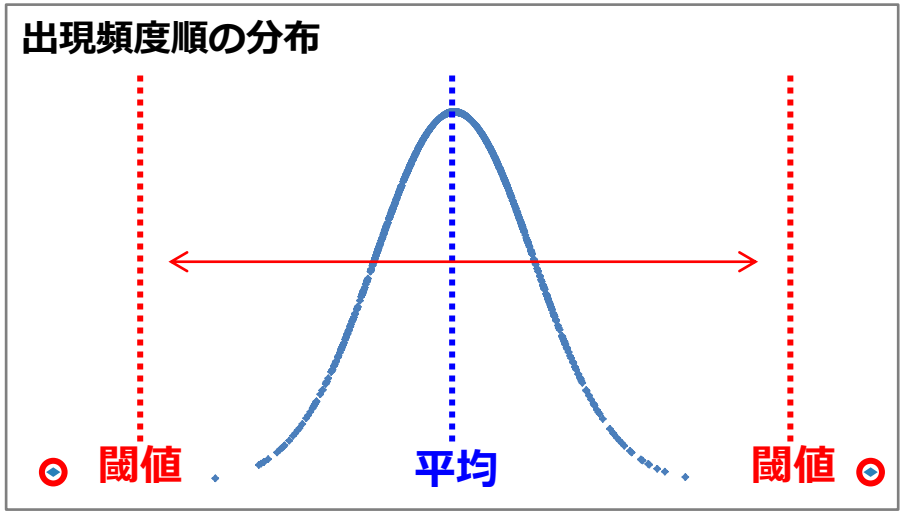


(1)出現頻度から検出

■ 分布上、出現確率の低いものを異常として検出

- 平均から一定以上離れているデータを、出現頻度の低い異常として検出する
- ユースケース
  - 閾値設定の自動化（出現頻度0.3%を異常とする、など）
  - Webレスポンス、バッチ処理時間の遅延
  - 工業製品検査工程での不良品検出

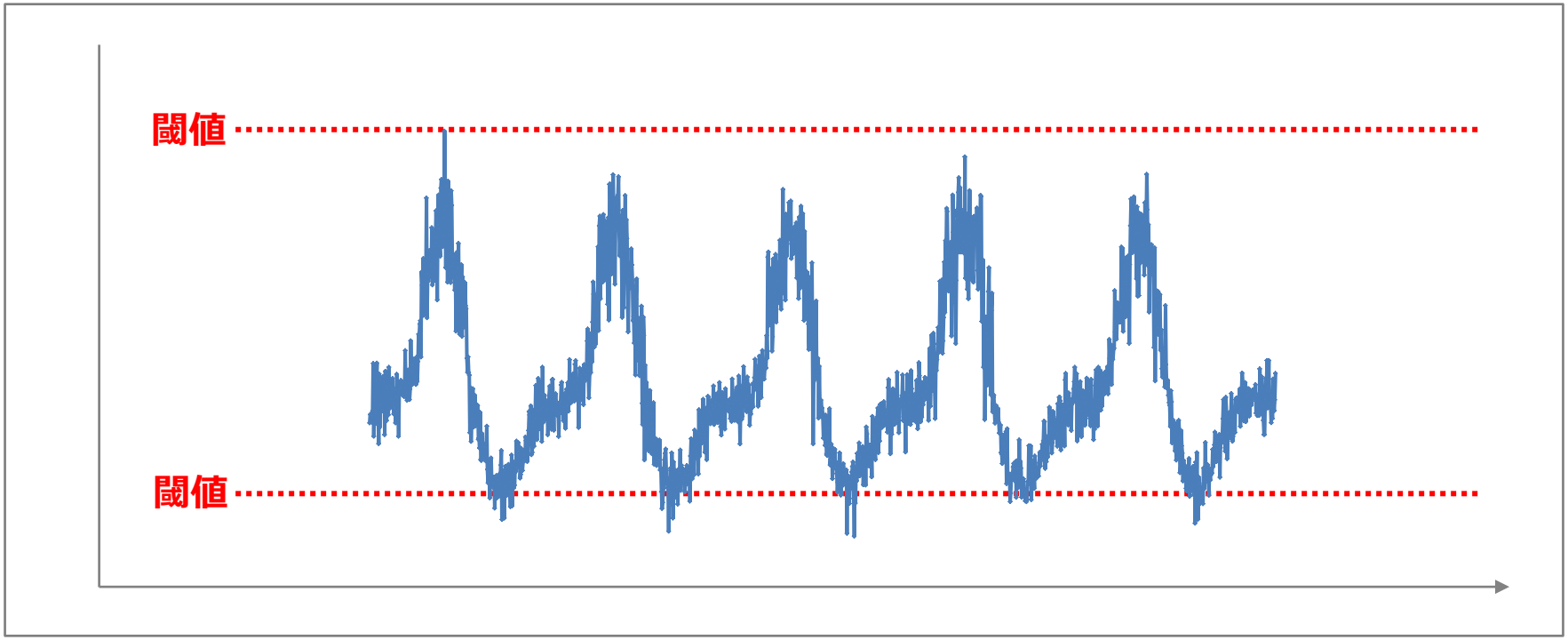




(1)出現頻度から検出

■ 課題

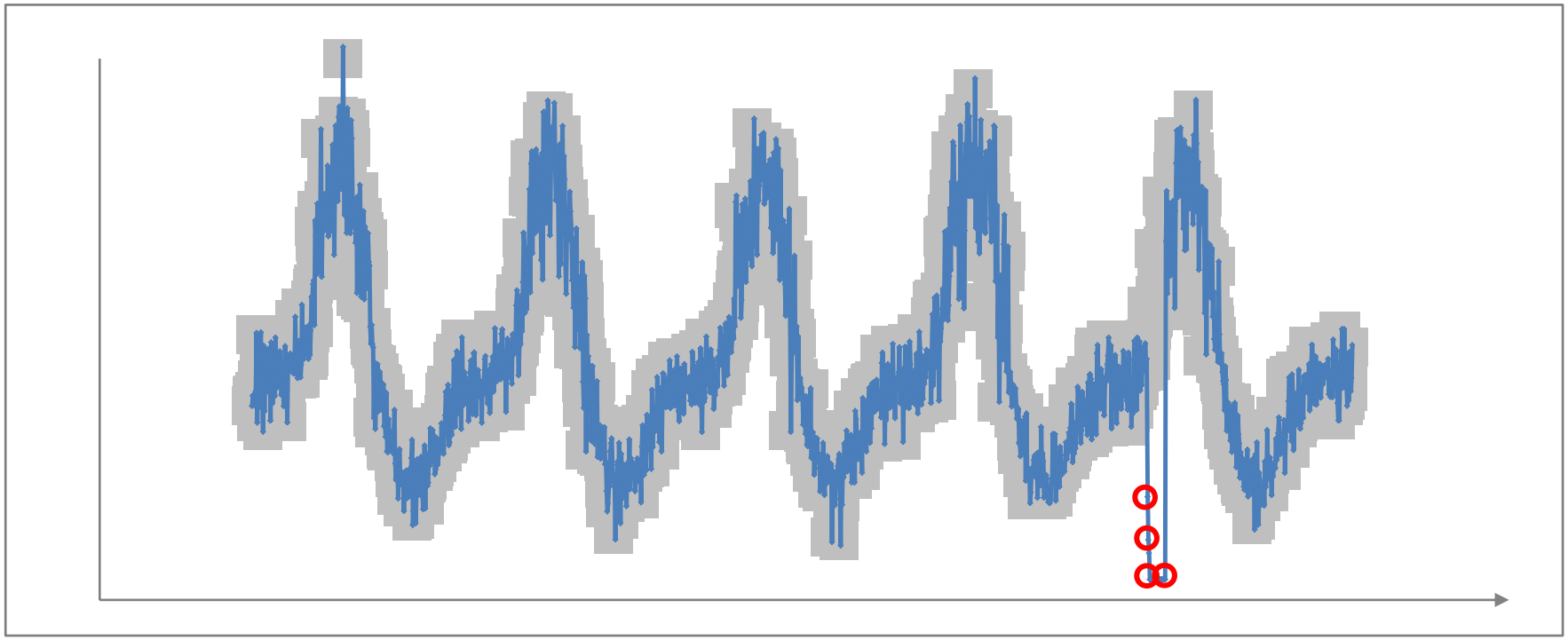
- 固定の閾値では対応できない場合
  - 数値の変動 上下幅が大きく、システムとしては問題なくても頻繁に閾値を超える



(2)周期性からの外れ

■ 周期性からの外れを異常として検出

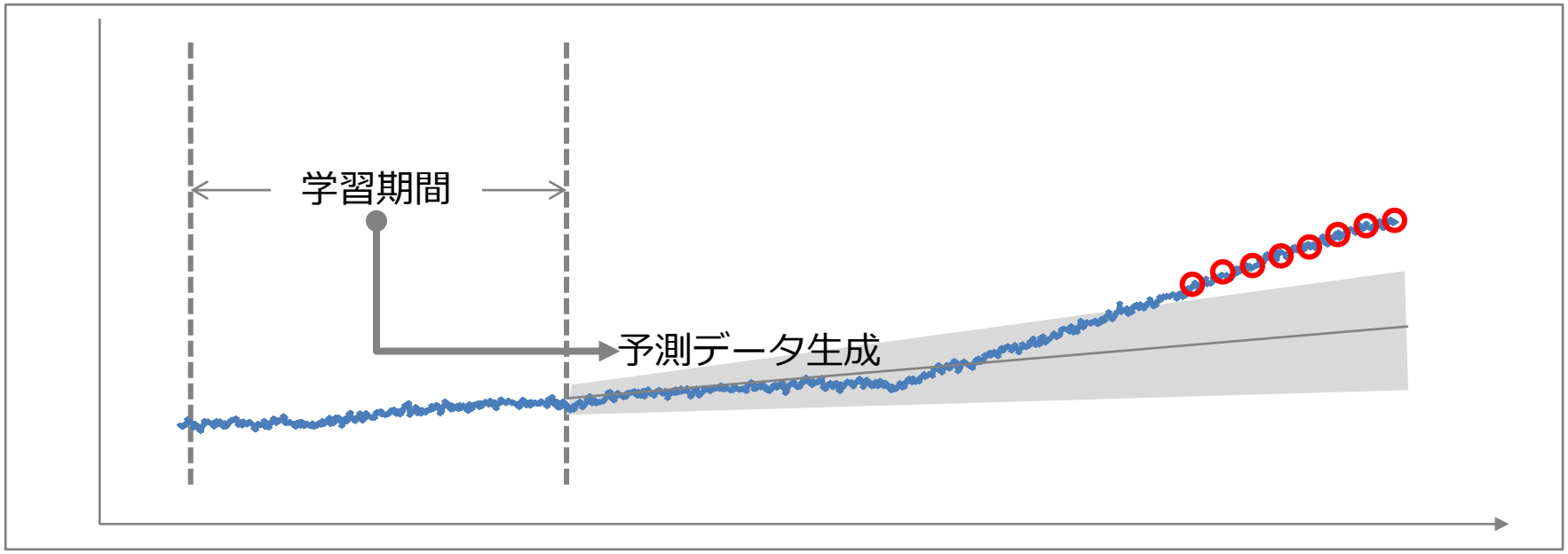
- 過去のデータを学習し、動的な閾値(正常値の範囲)を生成する
- データに周期性があることが前提になる
- ユースケース
  - ネットワークトラフィック異常検出(ポート故障)



(3)上昇・下降傾向の変化

■ 上昇・下降傾向の変化を異常として検知

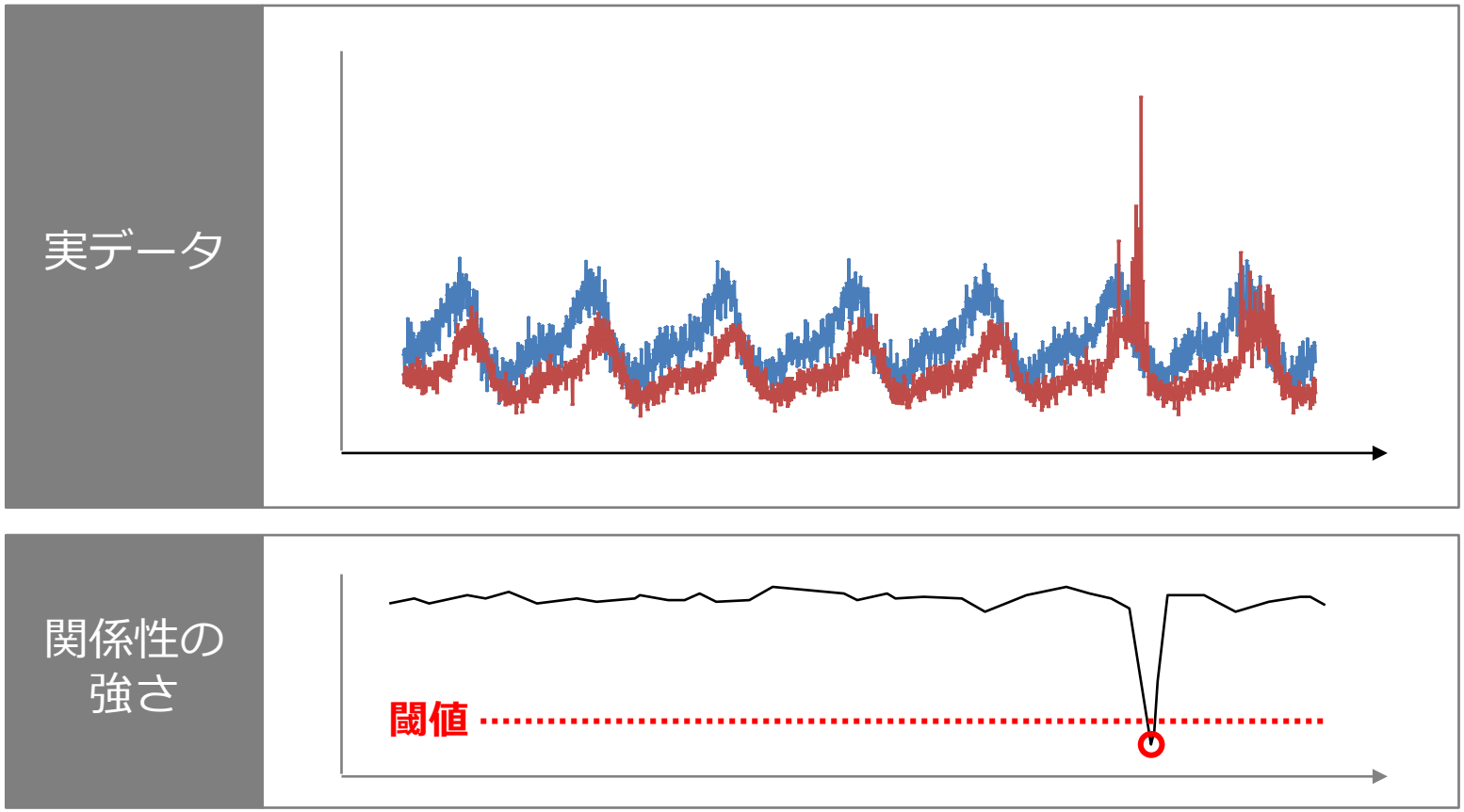
- 過去のデータを学習し、予測値と正常値の範囲を生成する
- 即時性はないが、長期間に渡って増加・減少傾向のあるデータの傾向変化
- ユースケース
  - サーバ上のプログラム不具合によるメモリリークの検出
  - 工場設備の長期的トレンドを見た経年劣化・故障予兆



(4)相関関係の崩れ

■ 複数データの関係性の崩れを異常として検知

- 一方の値が変化すると、もう一方の値も変化する関係性
- 関係性の強さを示す値を数値化し、その数値の変動をいつもと違う挙動として異常検知する



(4)相関関係の崩れ

■ 性能劣化に関する問題の検知

- ユーザ視点でのパフォーマンス劣化は、これまでの閾値監視では気付きづらく、原因特定までの時間も要する
- 実績と関係性のあるデータからパフォーマンス劣化の兆候の気付きを得る

#	実績	関連データ	異常検知
1	DBクエリ数	DBサーバシステムリソース	クエリ実行回数はいつもと変わらないのにサーバCPU/メモリ使用量が高い（DB上の異常プロセス、異常行動）
2	日照量	太陽光パネル発電量	日照量はいつもと変わらないのに発電量が低下してきている（故障予兆、経年劣化）

[課題例] 誤検知の削減

■ 誤検知

- 正常データを「異常」と判定
  - ロジック/アルゴリズムの不備による本当の誤検知
  - 運用者観点で「これは検知しなくていい」という要望

- 誤検知が多い、または単純に検知数が多いと、予兆監視システムが一気に「使えないもの」になる
  - 大量のアラートは無視されるようになる
  - 運用者にとっては検知後に無駄な調査リソースが割かれる

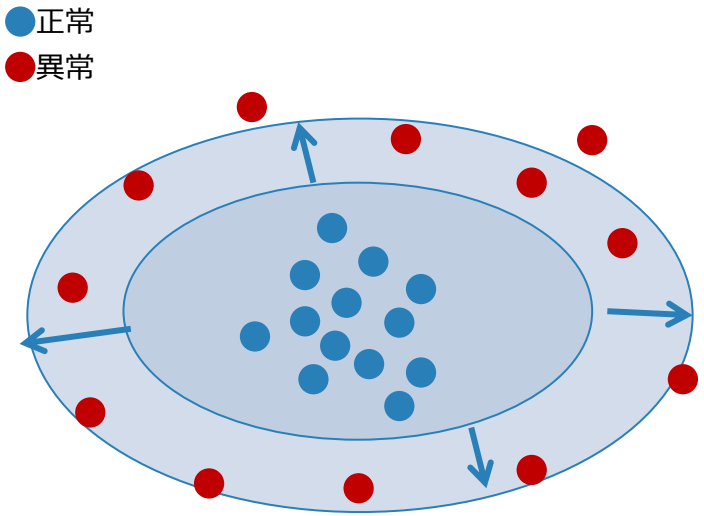
- パラメータ調整や独自ロジックの追加で、誤検知を減らし「本当に検知したい異常」だけを検知するようにチューニングしていく必要がある

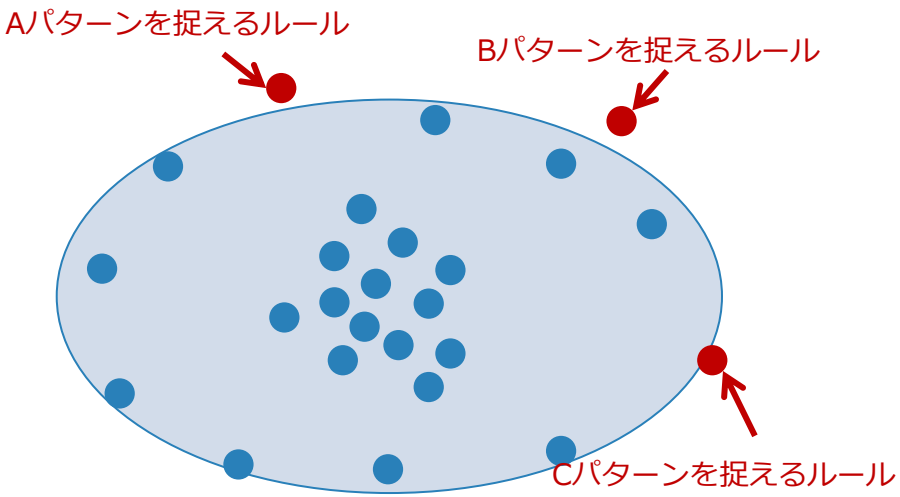
- 誤検知削減の取り組みと同時に、本当に検知したい異常の「見逃し」があってはならない

アプローチ

■ 2つのアプローチで検討

1. 正常時との違いを異常として捉える
2. 予めわかっている異常パターンをルール化しそれを捉える
（熟練者のもつ勘・経験をシステム化する）





1. パラメータと独自ロジックのチューニングで「本当に影響を及ぼす異常」を検知する
⇒未知の異常を検出できる可能性があるが検出数が多くなる
また、原因特定が難しい傾向

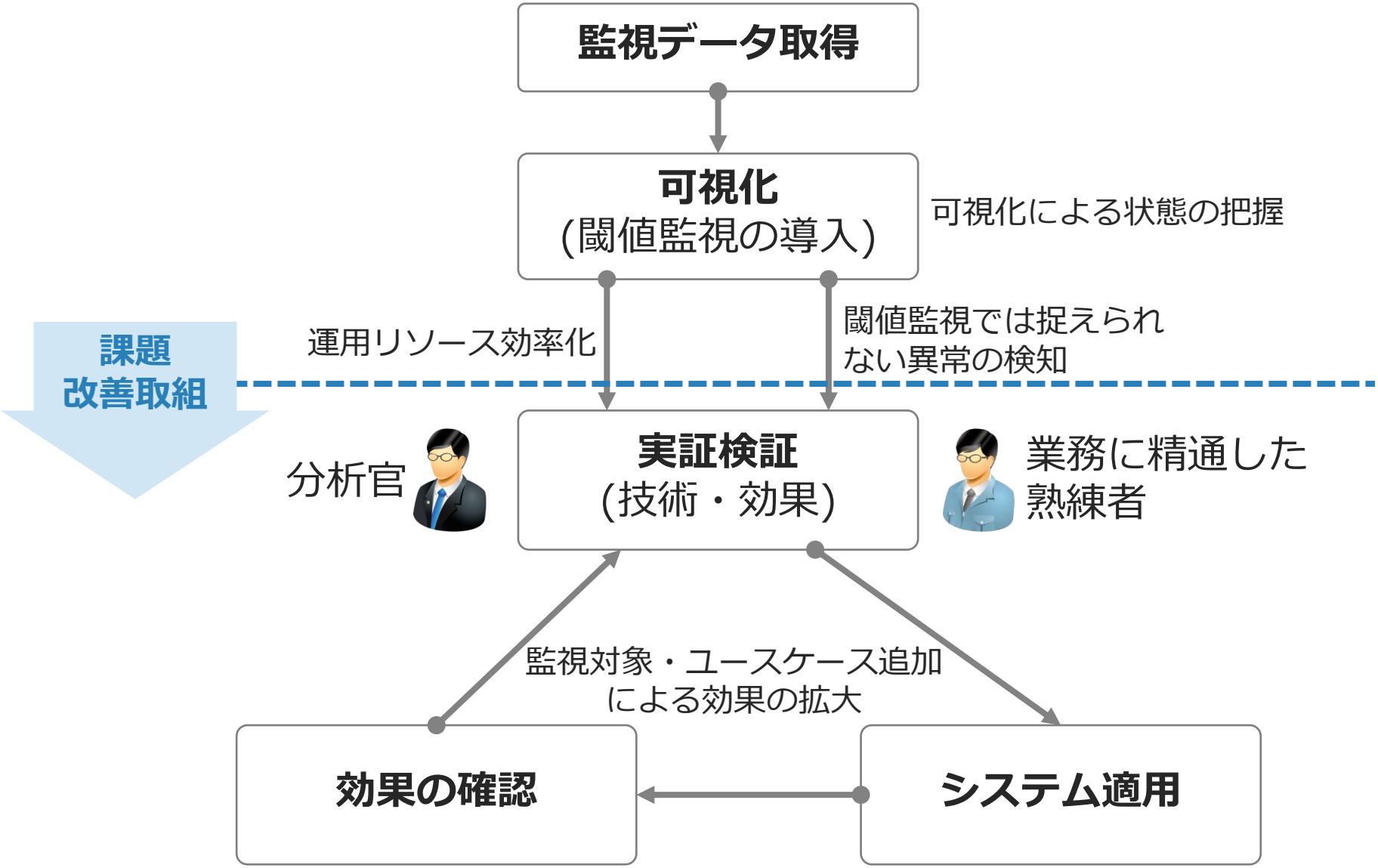
2. 「本当に影響を及ぼす異常」のパターンをルール化し、それを検知する
⇒既知の異常を検出する精度が高くなる可能性が高い
また、原因特定は容易

適用までのステップ

■ 適用効果を確認しながら段階的な適用が必要

- 予兆監視が監視・運用の全ての課題を解決するわけではない
- 特に初期段階では、これまでの閾値監視と併用する
- 実証検証にて、技術的な実現性と共に、課題に対する効果を測り、段階的にシステムへ適用していく

適用までのステップ(例)


監視データ取得
可視化
(閾値監視の導入)
可視化による状態の把握
課題
改善取組
運用リソース効率化
閾値監視では捉えられない異常の検知
分析官
実証検証
(技術・効果)
業務に精通した熟練者
監視対象・ユースケース追加による効果の拡大
効果の確認
システム適用

2.解析プラットフォーム

解析プラットフォーム

■ システム構成パターンの例

A) 簡易バッチ
- 小規模なデータに対して統計的な計算を行うバッチ環境

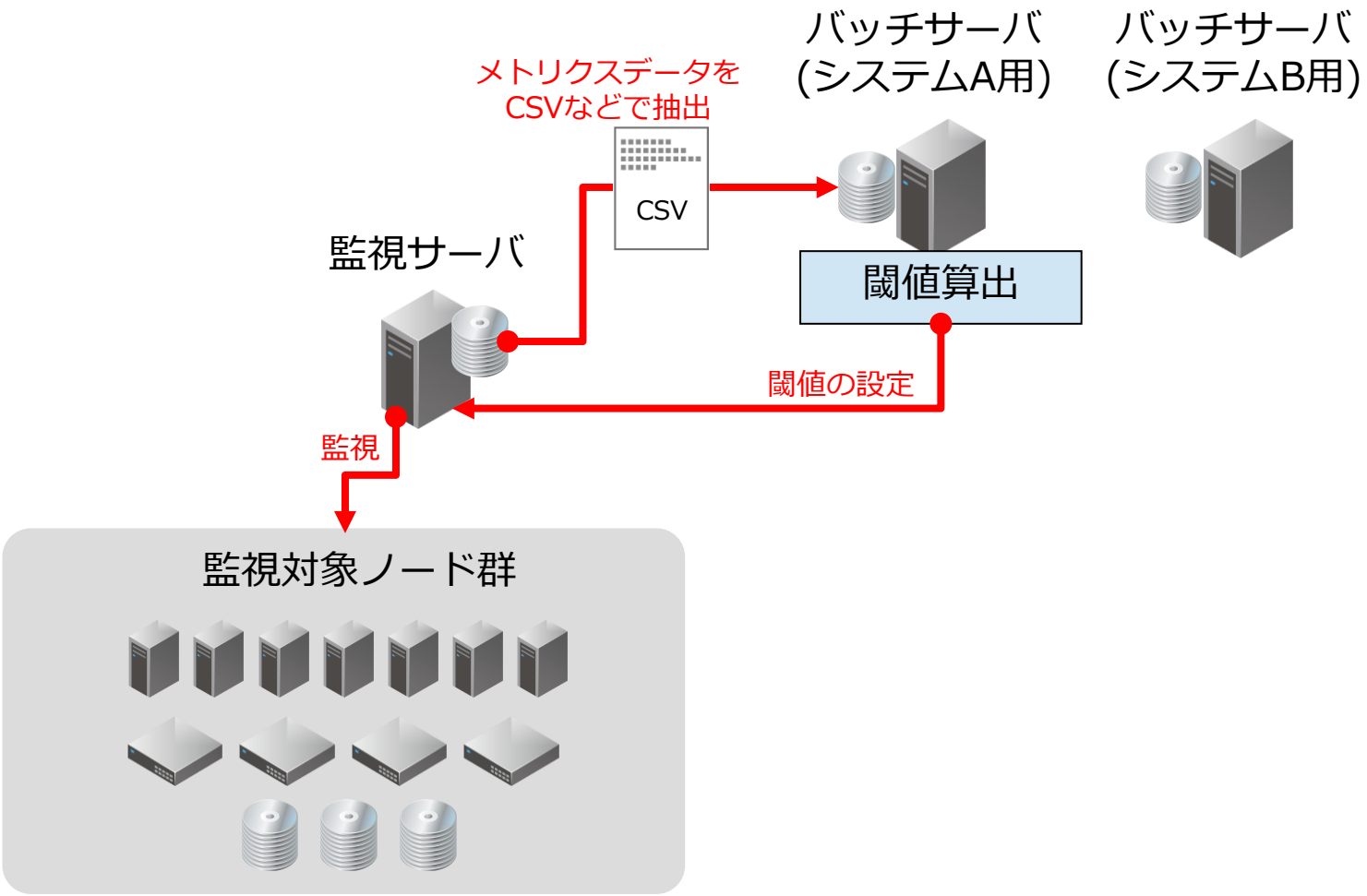
B) ビッグデータバッチ処理
- 大規模なデータに対して機械学習処理を行うバッチ環境

C) リアルタイム処理
- 機械学習を用いたリアルタイムに近い解析処理環境

A) 簡易バッチ

■ システム構成例

- 簡易な計算バッチで定期的に閾値を更新する



A) 簡易バッチ

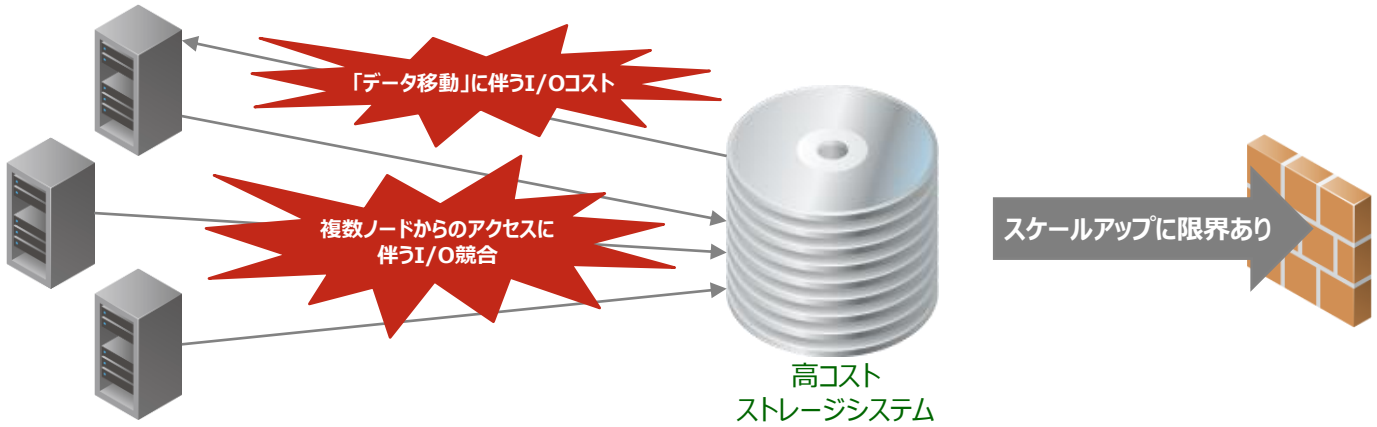
課題

- 監視対象追加によって、大容量のデータ、大量のメトリクスを扱う必要がある場合、1台のサーバではI/Oボトルネックに耐えきれない
  - 蓄積容量、処理性能の拡張性が必要になる

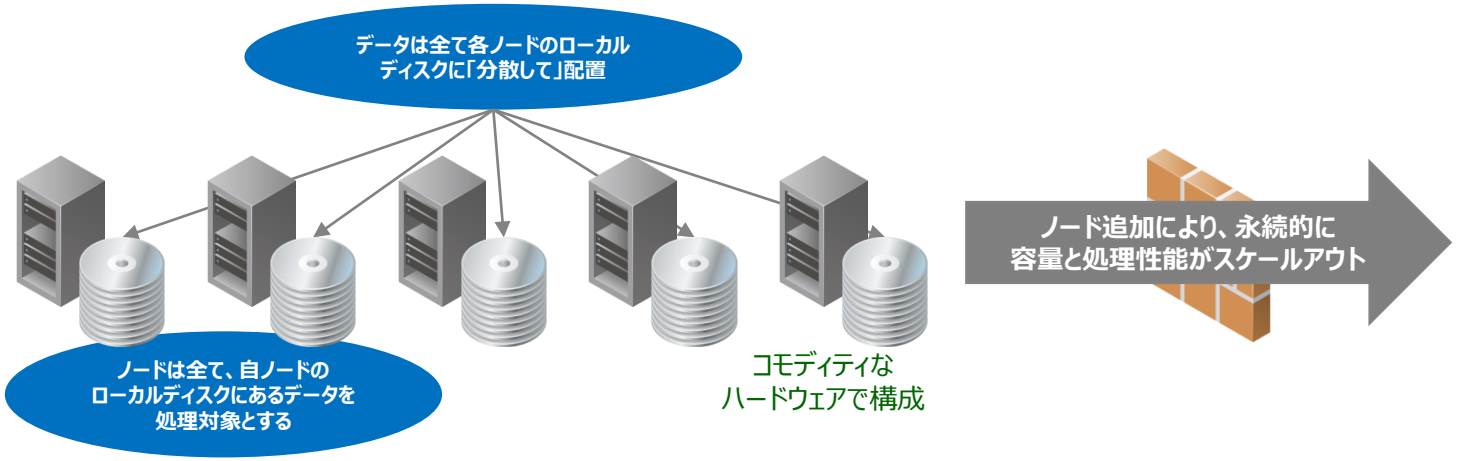
B) ビッグデータバッチ処理

■ I/O分散による拡張性

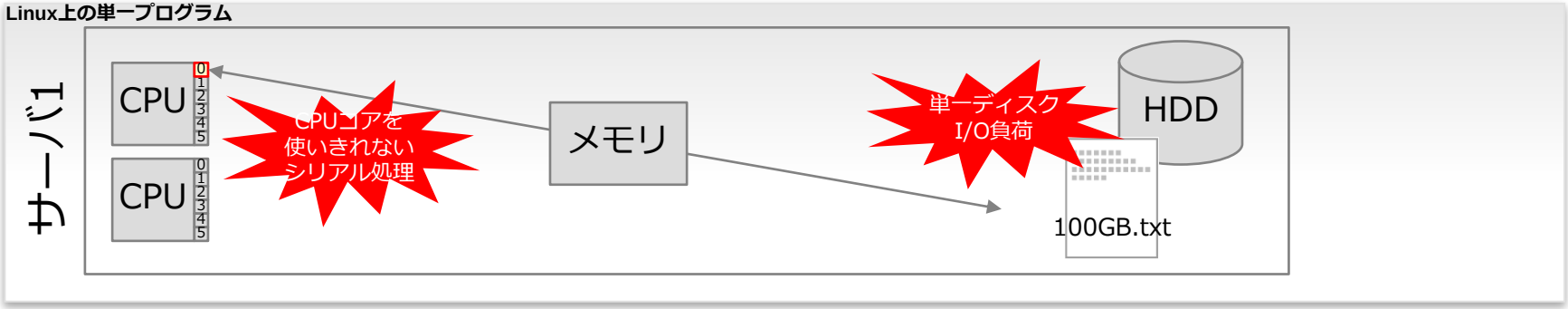
一般的なストレージの構成

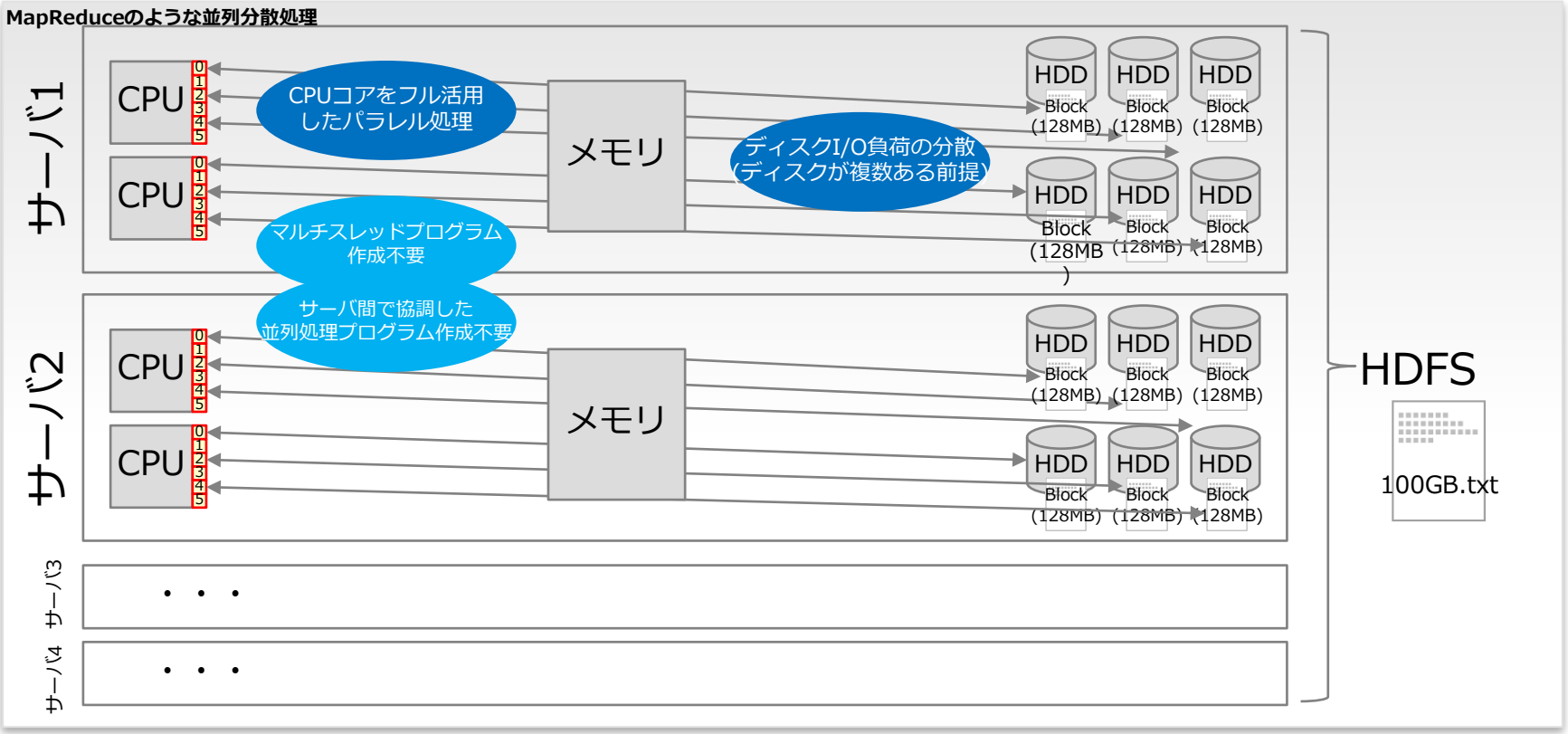


ビッグデータ基盤の構成（シェアード・ナッシング）



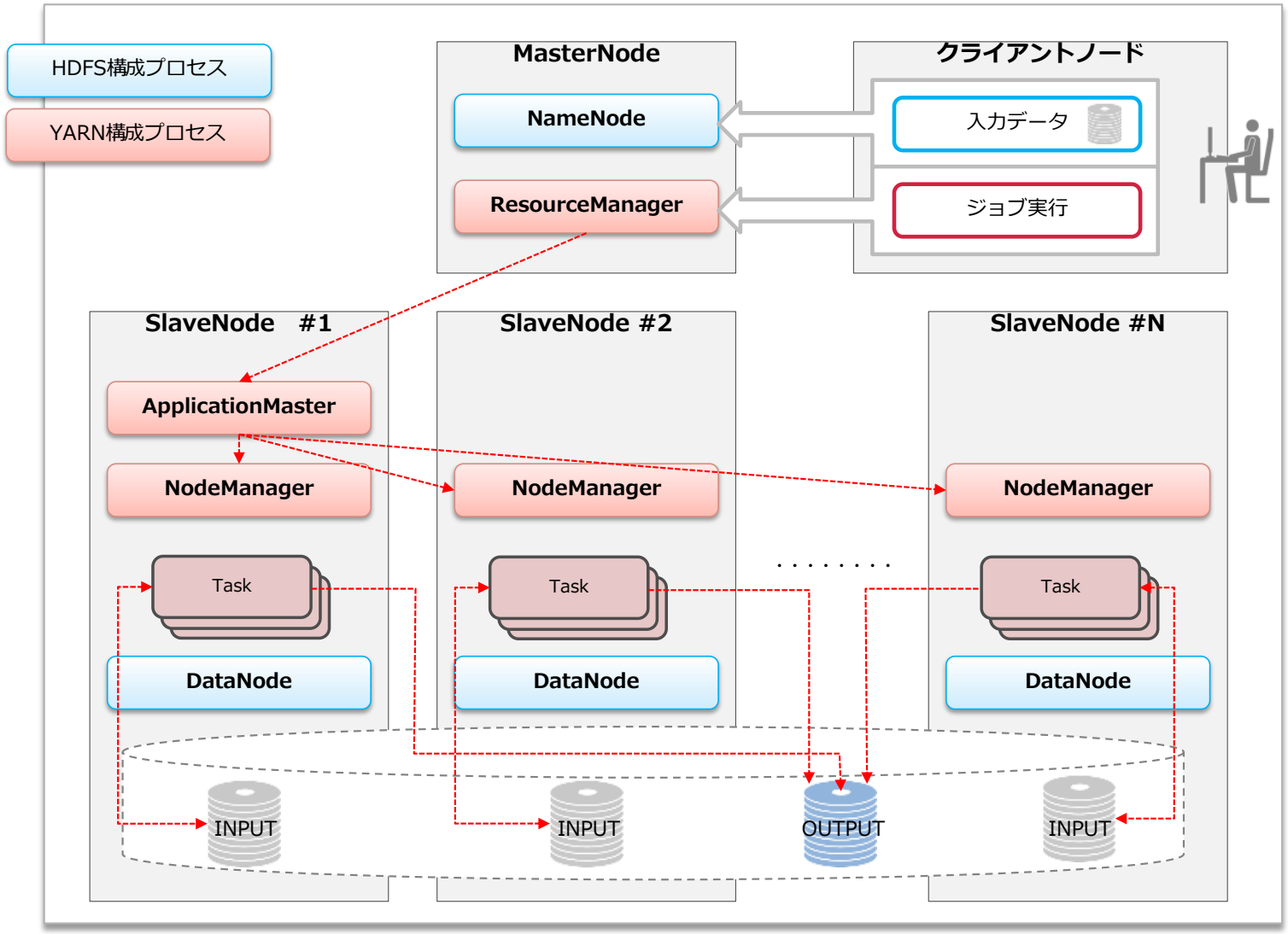
B) ビッグデータバッチ処理





B) ビッグデータバッチ処理

■ Hadoopアーキテクチャ



C) リアルタイム処理

■ 監視システムでは即時性が重要になる

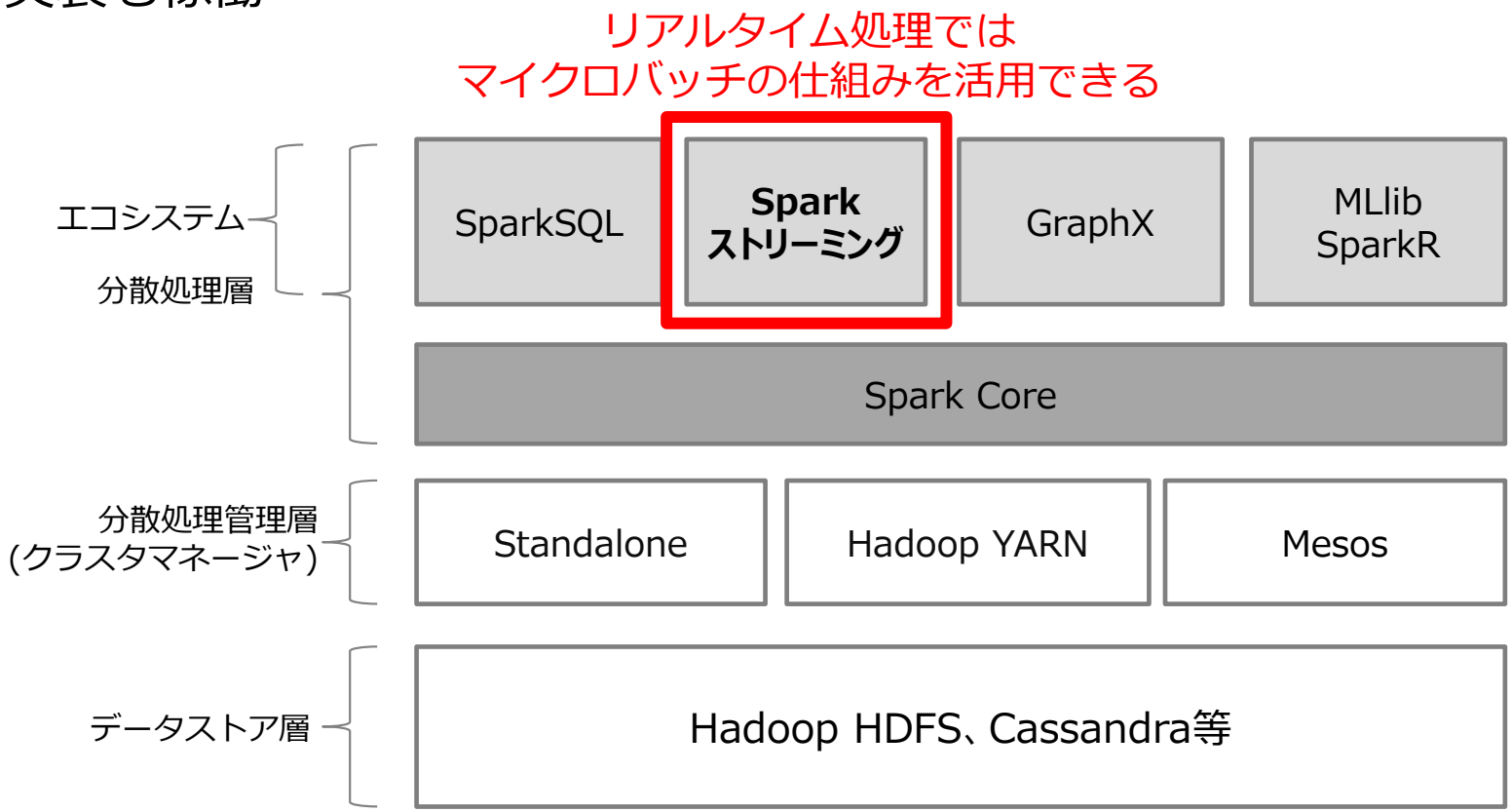
- Hadoopなどのバッチ処理
  - 即時性が求められない、または、重要ではない
  - 大容量のログファイルの集計・解析
  - 日次での解析

- リアルタイムに近いストリーム処理
  - 即時性が何より重要になる
  - IoTセンサーデータから送信されるデータの状況把握と分析
  - ユーザの直近行動に基づくアクション（リアルタイムのレコメンド）

C) リアルタイム処理

Apache Spark

- MapReduceのバッチ処理モデルを補完するインメモリ処理基盤
- バッチもストリーミング処理も同じ実装で実現
- MLlib/SparkRに加え、統計解析/機械学習ライブラリの豊富なpython実装も稼働

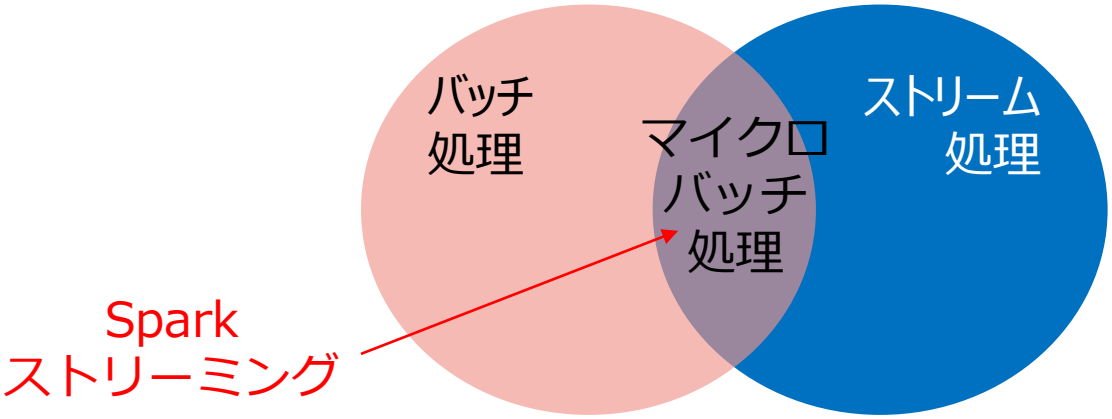


C) リアルタイム処理

Sparkストリーミング

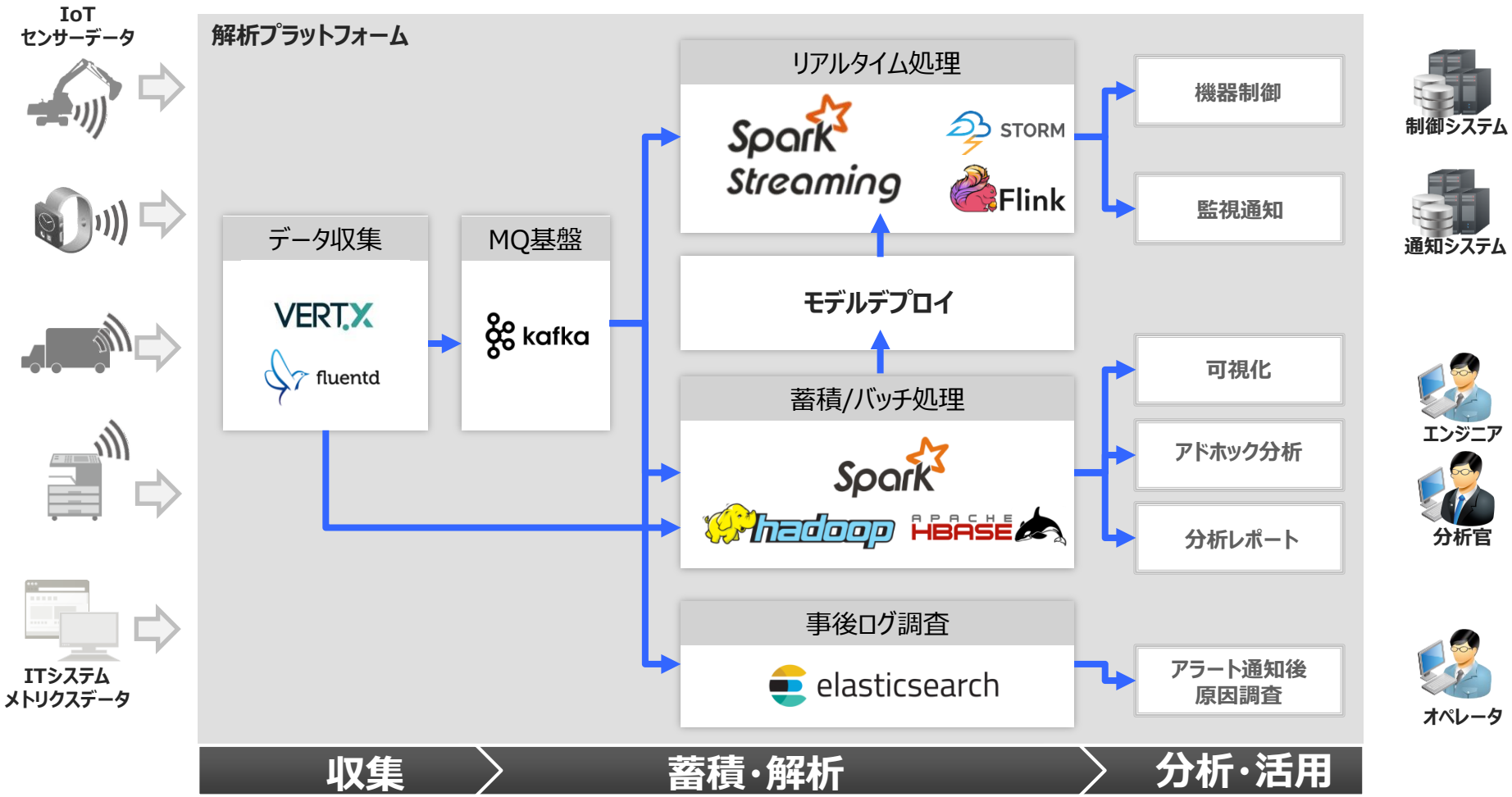
- 全てのデータを保管する必要がない場合、リアルタイムの検知処理を行い、最低限のデータのみを保持する
  - 異常検知用途では、1年前のデータを学習するシーンは少ない

- データをディスクに書き出すことなく、メモリ上で処理をおこなって結果を返すストリーミング処理
- 厳密には「マイクロバッチ方式」のため、ミリ秒レベルのレイテンシであれば別の方式が必要になる



解析プラットフォーム

■ アーキテクチャ例



3.IoT×予兆監視システム

IIJ IoTサービス×予兆監視

■ IoTに必要な技術要素は多岐に渡る

センサー/デバイス	ネットワーク	プラットフォーム(クラウド)	業務開発
データの取得	コネクティビティ	蓄積・可視化・連携	活用

IIJ IoTサービス×予兆監視

■ IoTに必要な技術要素は多岐に渡る

センサー/デバイス	ネットワーク	プラットフォーム(クラウド)	業務開発
データの取得	コネクティビティ	蓄積・可視化・連携	活用
	IIJ IoTサービス		予兆監視

IIJ IoTサービス

IoT専用SIMを中心に付加機能を自由にアドオン

モバイルアクセス
IoT用途向け モバイル通信SIM

＋

- データハブ
- データストレージ
- デバイス モニタリング
- プライベート コネクタ
- プライベート モバイルゲートウェイ
- デバイス コントロール

IIJ IoTサービス×予兆監視

■ IoTにおける予兆監視のメリット

- 多数のセンサー/デバイスから送信されるデータ全てに対し、ひとつひとつ監視設定を加えていく設定作業負荷

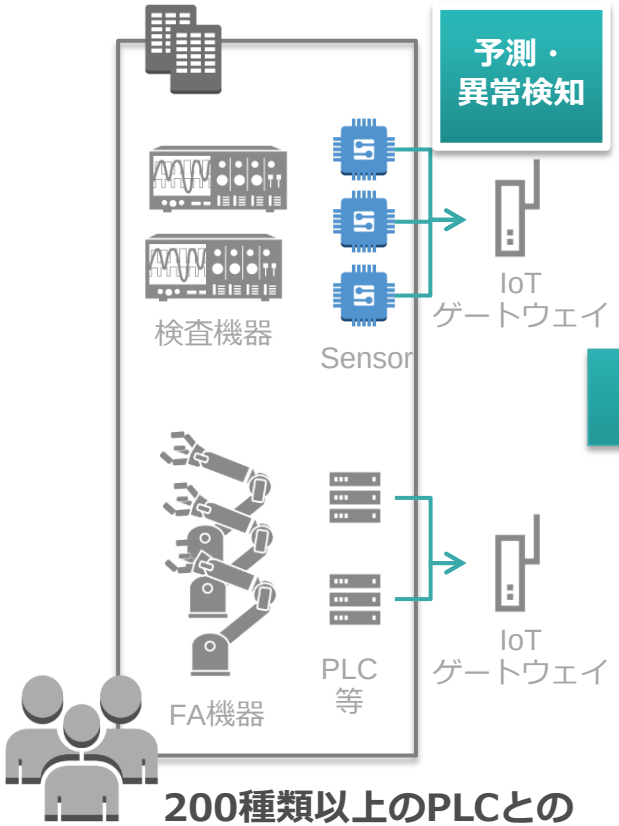
- これまでの閾値監視、個々のセンサーデータの監視では検知出来なかった異常を捉えられる可能性がある
  - 複数のセンサーデータの相関関係
  - 故障予測・経年劣化

- データ分析の面では、人の介在が少ないM2Mデータが多いため、検証した通りの結果が出やすい傾向がある

ブレインズテクノロジー「Impulse」

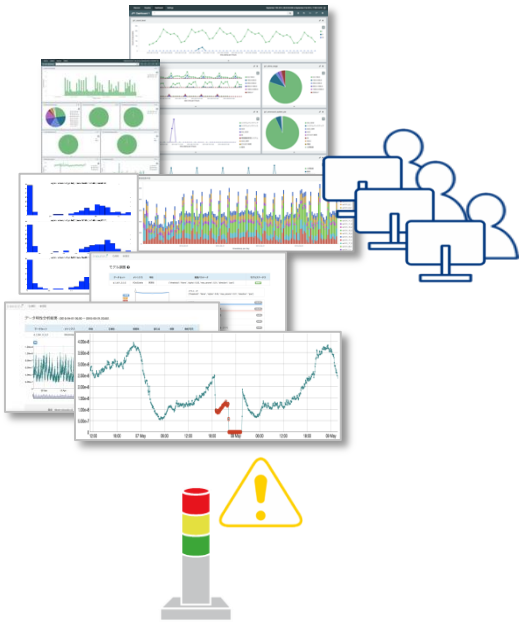
- 様々なデータを収集・構造化し、集計や異常検知等の高度なリアルタイムデータ分析処理を実現するプラットフォーム

先端クラウド技術により、
エッジでの異常検知処理を実現



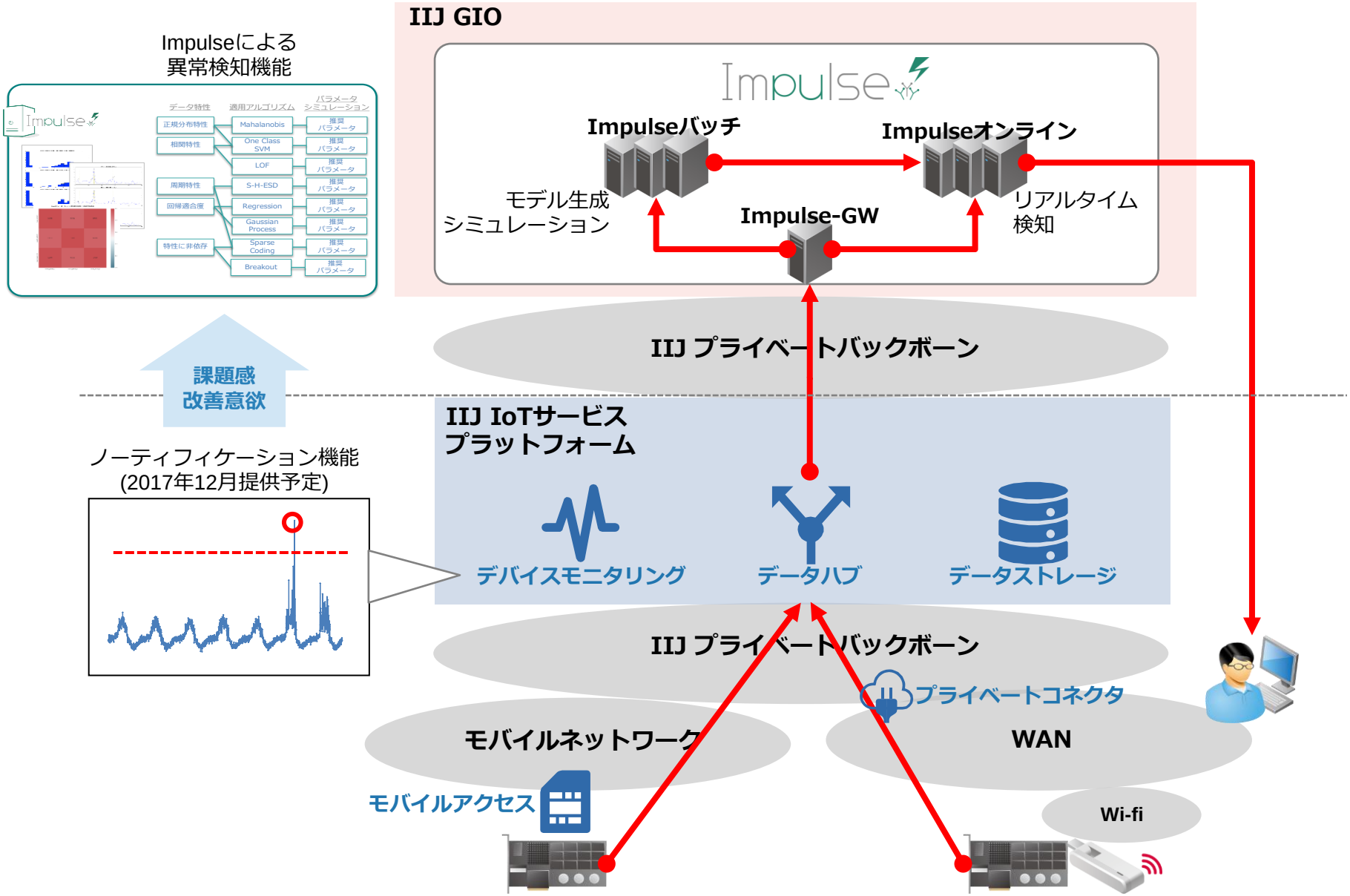
200種類以上のPLCとの
シームレスな接続を実現

アルゴリズム、モデルの
自動選択により誰でも、
簡単に始められる分析基盤



機械学習を利用した
リアルタイムでの
故障予兆検知が可能

IIJ IoTサービス×予兆監視



まとめ

1. 予兆監視 適用ステップ

- 異常検知の特性・課題を把握し、効果につなげるための実証検証を行った上で、段階的なシステム適用を進めて行く必要がある

2. 解析プラットフォーム

- より大規模なデータを扱い、リアルタイムに検知を行う予兆監視システムのアーキテクチャ

3. IoT×予兆監視システム

- IoTの「つなぐ」から開始し、より具体的なデータ活用のひとつとして異常検知の取り組みにつなげることをIIJサービス/ソリューションが支援

IIJ Internet Initiative Japan

Lead Initiative

日本のインターネットは1992年、IIJとともにはじまりました。以来、IIJグループはネットワーク社会の基盤をつくり、技術力でその発展を支えてきました。インターネットの未来を想い、新たなイノベーションに挑戦し続けていく。それは、つねに先駆者としてインターネットの可能性を切り拓いてきたIIJの、これからも変わることのない姿勢です。IIJの真ん中のIはイニシアティブ

IIJはいつもはじまりであり、未来です。

Ongoing Innovation