IBM API Connectで学ぶ API管理の勘所

株式会社オージス総研
サービス事業本部 クラウドインテグレーションサービス部
齋藤 伸也（Saito_Shinya@ogis-ri.co.jp）

目次

- API公開の課題
- なぜAPI管理が必要なのか
- API管理のベストプラクティス
- IBM API Connect を使ったAPI管理
- オージス総研のAPI運用サービスの紹介

オージス総研

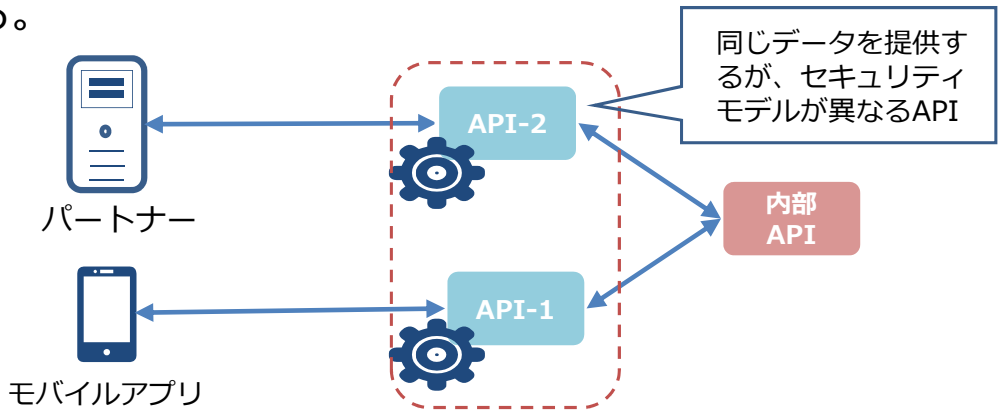
API公開の課題

課題１：増加するAPIの管理

- 公開されているAPIが増えると実際に処理を行う内部APIも増加する

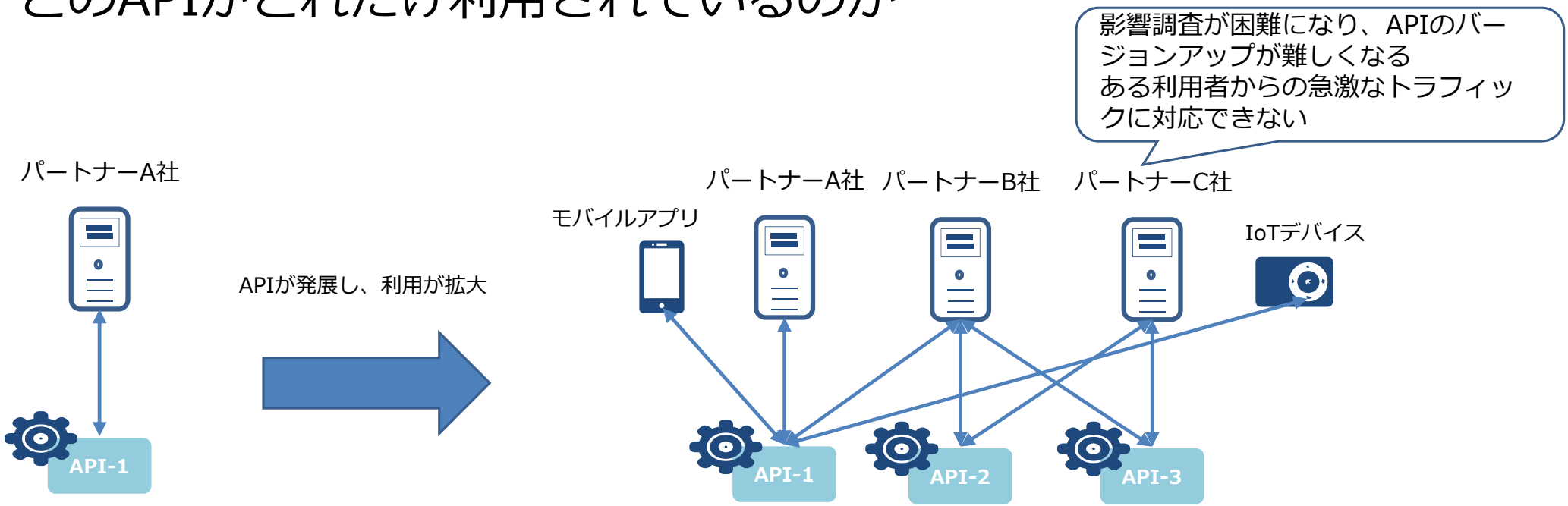
- 初めに参照系のみ場合は５API程度からスタートする場合が多い。更新系が増え、一連のビジネス機能を実行するために必要なAPIを提供すると30～40程度になる。複数のビジネス機能を提供すると数はさらに増える。１組織で100や200のAPIを提供していることも珍しくない。

- 同じ機能を提供するAPIでも、クライアントによってエンドポイント(URI)を変えるケースがある。この場合もAPIが増加する。



課題２：増えるAPIクライアント/利用者の管理

- APIは誰(どのAPIクライアント)から利用されているのか
- どのAPIがどれだけ利用されているのか

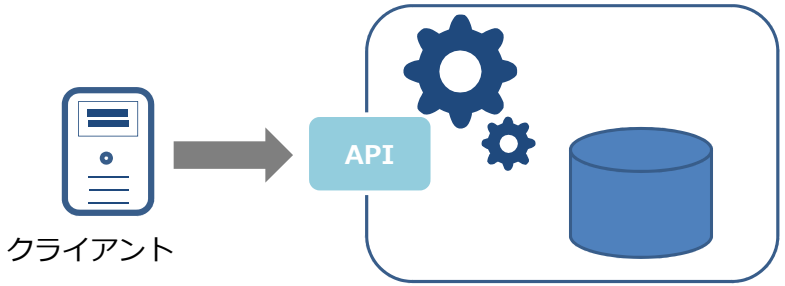


課題３：様々なセキュリティモデルへの対応

- 提供しているAPIは、クライアントに対してアクセス制御したいのか、リソースオーナー(エンドユーザー)に対してアクセス制御したいのかによってセキュリティモデルが異なる

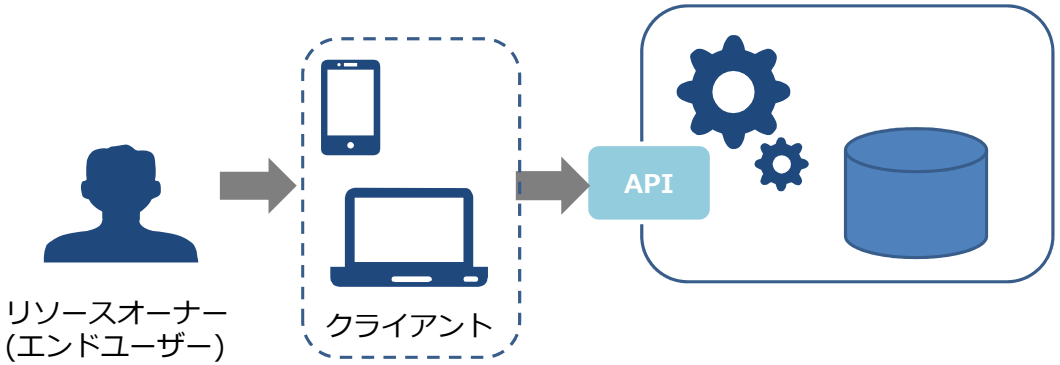
クライアントに対するアクセス制御

ユーザに依存しない特定クライアント向けのデータや機能
例えば、カタログ情報参照API
認証・認可のセキュリティモデル Basic認証、API Key、リクエスト署名



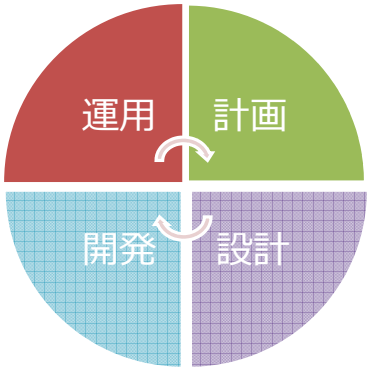
エンドユーザーに対するアクセス制御

ユーザーが承認したデータや機能
例えば、サードパーティアプリに提供する、口座情報参照API
認証・認可のセキュリティモデル OAuth2.0



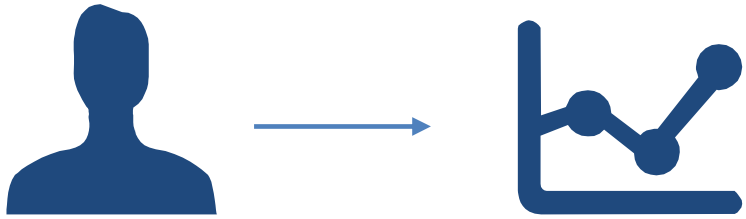
課題４：迅速なAPI開発

- APIは新しいビジネスや新しいパートナーに向けて公開されるため、常に変化が求められる
- APIは提供サービス・データの更新や接続先の増加に合わせて、変更や追加が発生する
  - 平均すると２ヶ月に１回ぐらいの頻度でAPIの変更や追加が発生(当社実施した案件の実績値より)
  - 短い場合は、１週間でAPIの改修、本番デプロイを実施
- いかに早くサイクルを回していくか、が大事になる



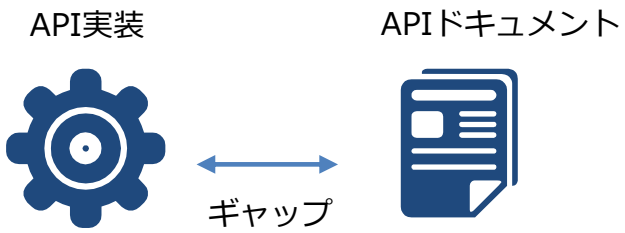
課題５：安定したAPIの提供

- 他のシステムと同様にAPIもダウンして利用できなくなったり、ターンアラウンドタイムが増加し、タイムアウトが発生するような状況は避けなければならない
- 公開されたAPIは様々なクライアントから利用され、ビジネスに直結する機能をもつものも少なくない
- APIの死活監視や、負荷の測定によってパフォーマンスの低下を監視し、APIのダウンを未然に防がなければならない



課題６：実装とドキュメントの乖離

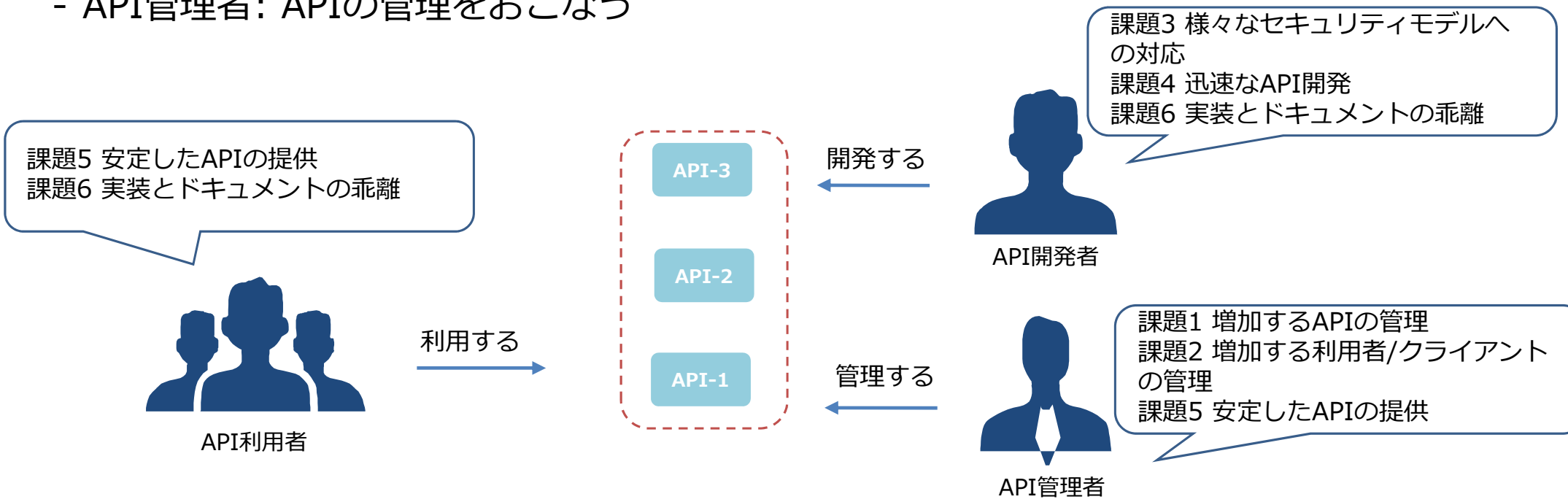
- APIを利用する開発者向けにAPIの仕様をドキュメントとして提供しなければならない。
- WordやHTMLなどの静的なドキュメントでは、変化の多いAPI開発に追従させなければならないが、APIの実装と仕様の乖離が発生してしまうことがある。



関係者ごとの課題

◘ API公開の関係者

- API利用者(アプリ開発者): 公開されているAPIを利用してアプリをつくる
- API開発者: 公開するAPIを開発する
- API管理者: APIの管理をおこなう



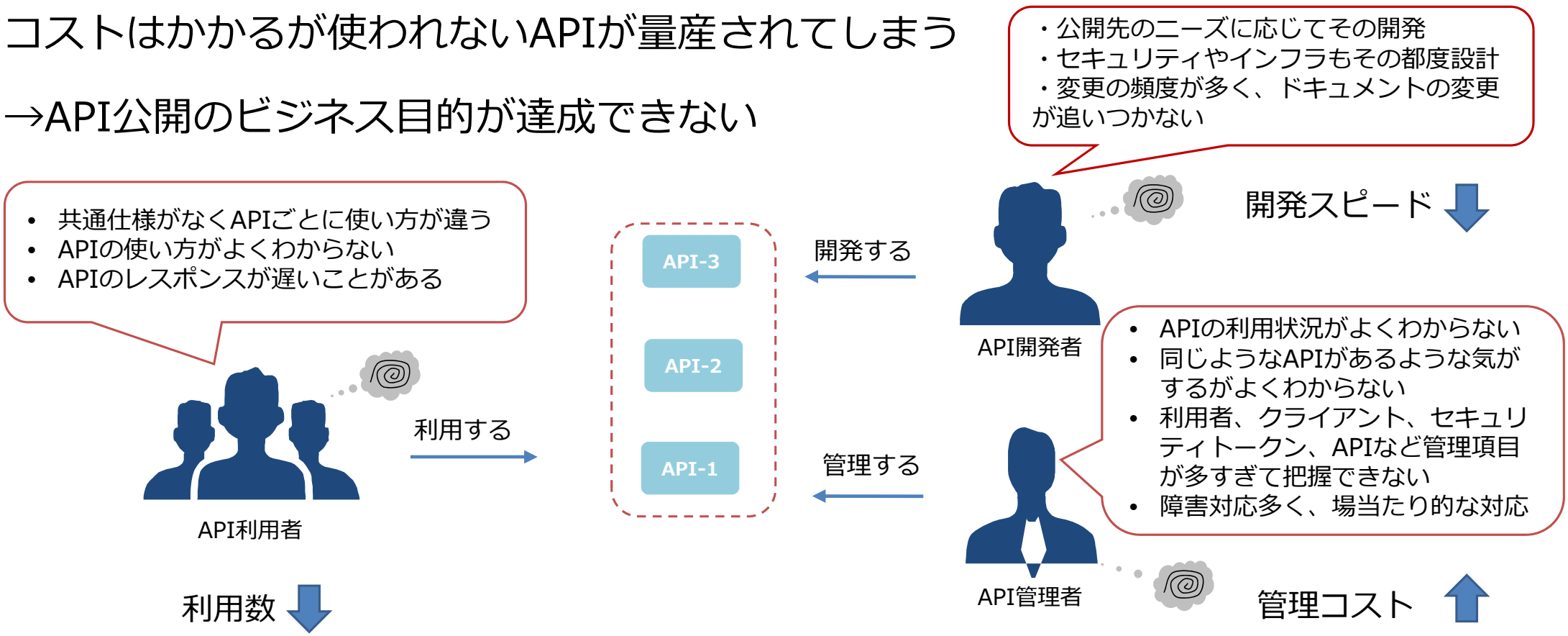
オージス総研

なぜAPI管理が必要なのか

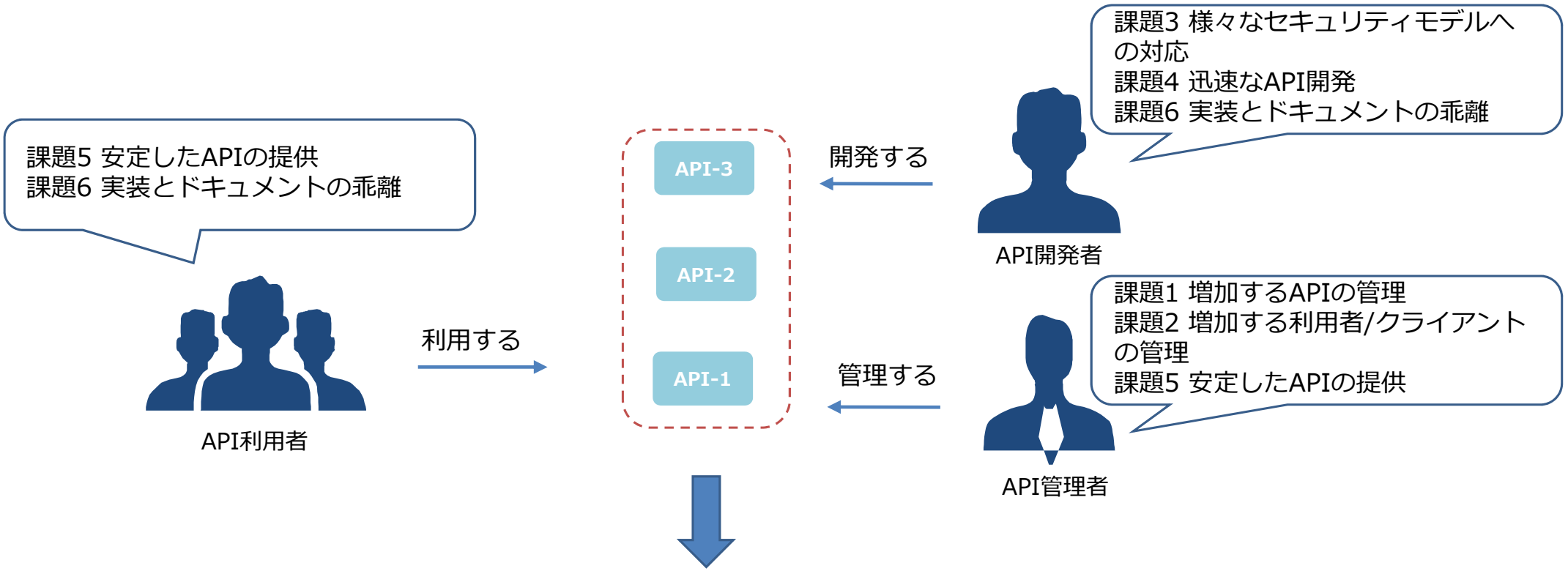
API管理をちゃんとしないと、、、

コストはかかるが使われないAPIが量産されてしまう

→API公開のビジネス目的が達成できない



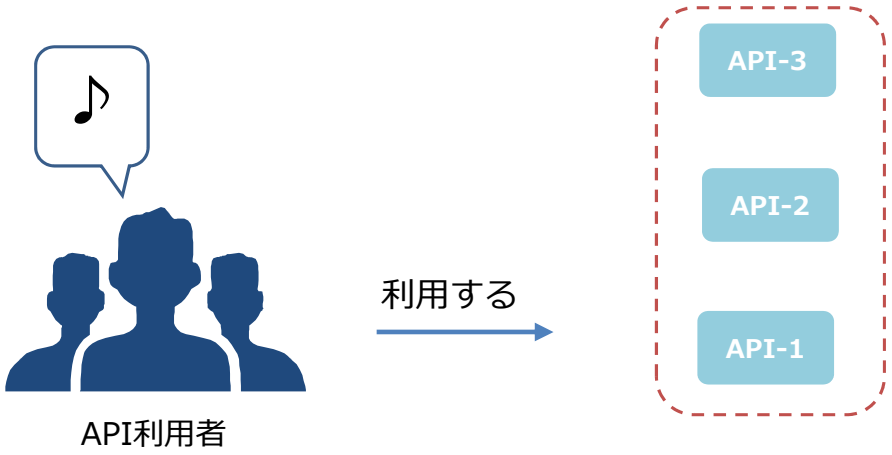
API管理の目的



課題を解決し、API公開のビジネス目的の達成を目指す

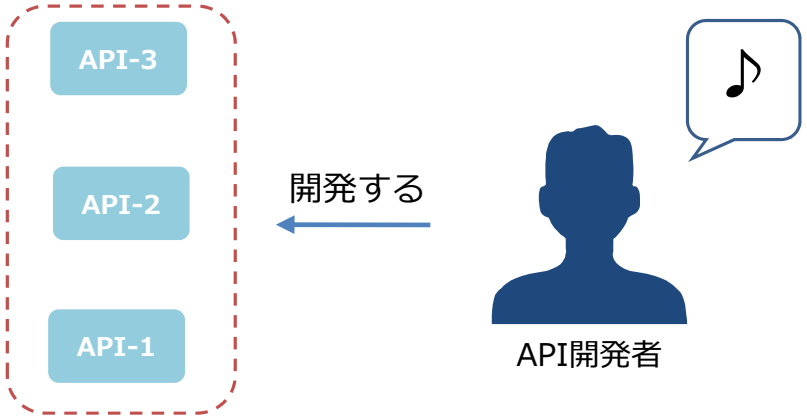
アプリ開発者(API利用者)のメリット

- APIの使い方がよくわかる
- 標準化されているため呼び出すのも簡単
- 安定している
- 変更や障害の情報発信が行われており、安心して使える



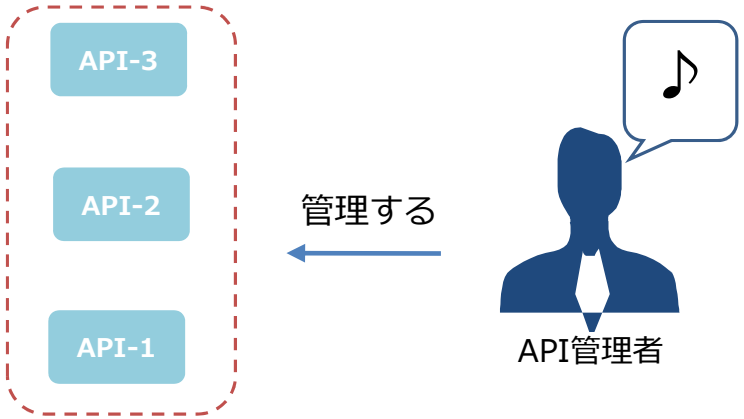
API開発者のメリット

- 標準されたセキュリティモデルを適用できる
- API公開に共通的に必要な部分を共有できる
- APIの追加、変更を迅速に行うことができるようになる
- 実装とドキュメントの乖離がなくバージョン管理が実施できるようになる



API管理者のメリット

- APIの開発-リリースのプロセスが明確になる
- 誰がどのAPIを利用しているのか明確になる
- 利用状況から、安定運用のための施策をうてるようになる



オージス総研

API管理ベストプラクティス

製品やサービスだけでは十分ではない

オージス総研

- API管理の製品やサービスは様々な機能が提供されているが、それらを全て使わなければならないわけではない。
- 製品やサービスは『**あくまでツール**』：管理の体制やプロセスを効率的に実践する道具立て
- 製品やサービスの導入だけでは解決できない問題
  - 適切なセキュリティモデルは何を選択するべきなのか
  - APIの設計-開発-テスト-リリースはどのように進めるべきなのか
  - APIの管理はどのようにおこなうべきか
  - 利用者、クライアントの管理はどのようにおこなうべきか
- 最適なプロセス、体制、製品機能の組合せを考慮する必要がある

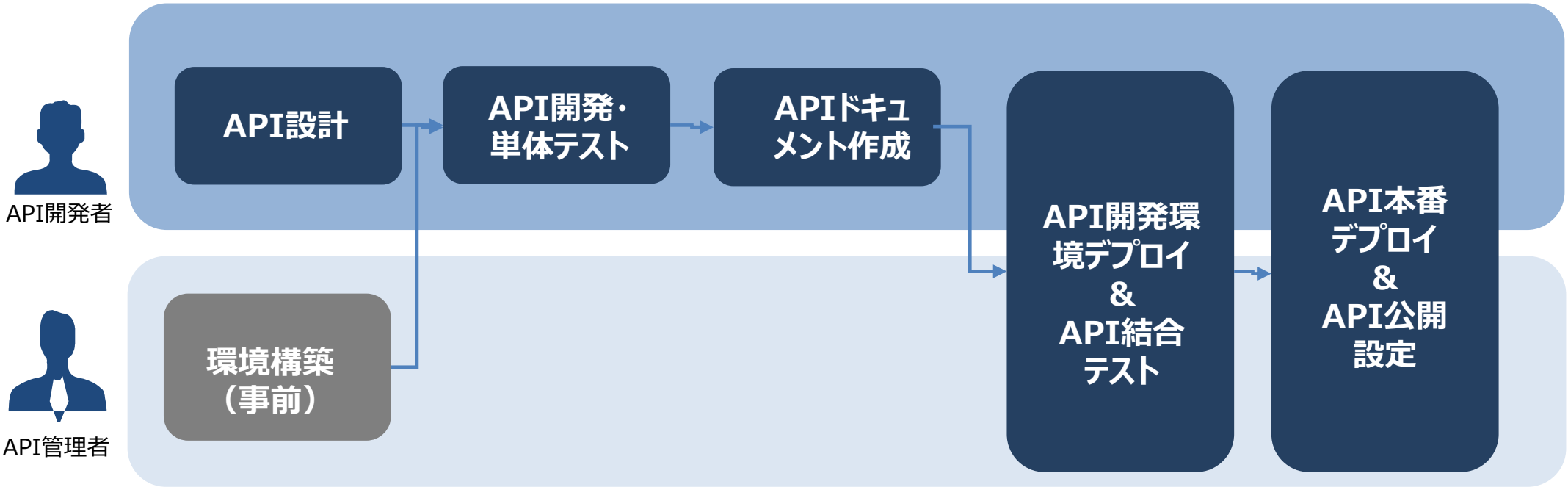
運用・管理の体制・プロセスが重要

オージス総研

- 運用・管理の体制
  - API開発者: APIの開発を担当する。
  - API管理者: API公開の設定やAPI自体、利用者の管理を行う。
- 代表的な３つのプロセス
  - API開発プロセス
  - API利用プロセス
  - 障害・アラートの対応プロセス

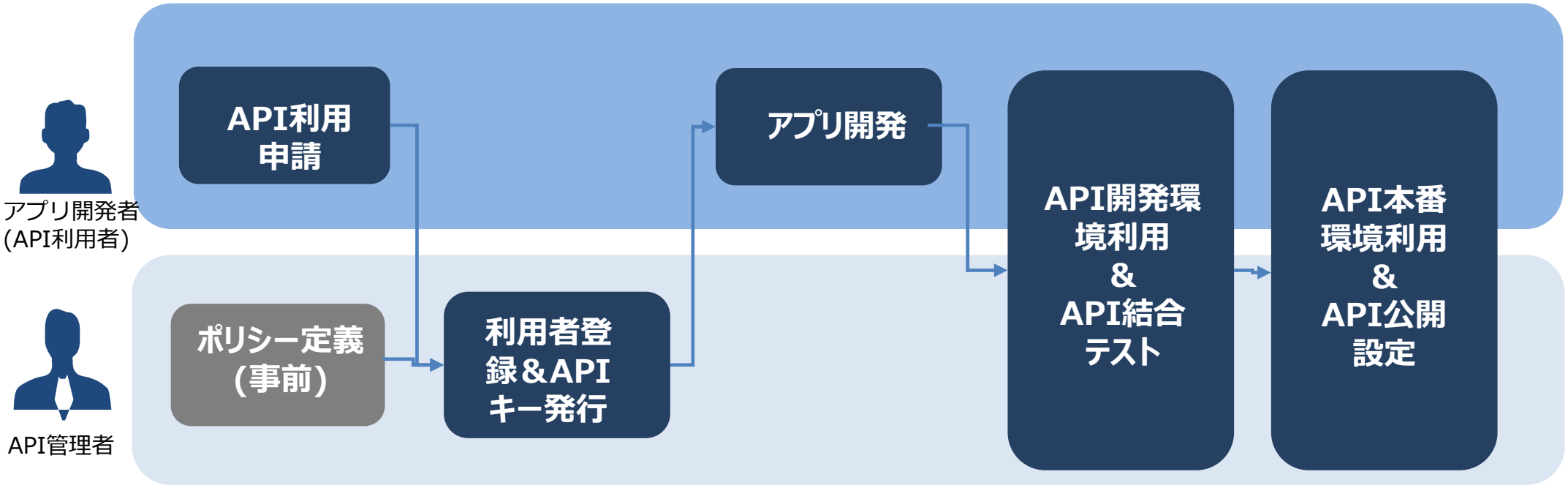
APIの開発プロセス

- APIの要求から本番リリースまでをカバーするプロセス
- CI/CD のツールを活用し、出来る限り自動化を行う



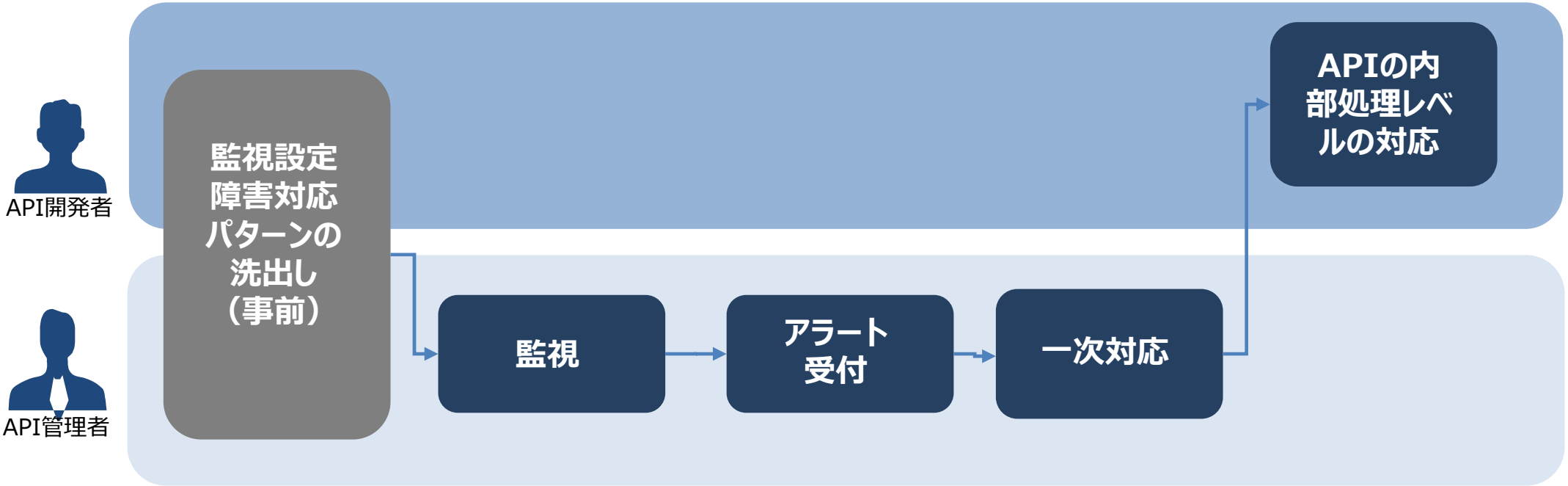
API利用プロセス

- アプリ開発者(API利用者)が利用申請からアプリの本番リリースまでをカバーするプロセス



障害・アラート対応プロセス

- 事前に起こりうる障害パターンを洗出し、それを検知可能な監視および対応方法を明確にする。
  - レイヤー構造：インフラ、ミドルウェア、API実装
  - 分散構造：API Gateway、マイクロサービス：リクエスト/レスポンスのトレーサビリティ



開発チームと運用チーム

- 開発チーム(API開発者)と運用チーム(API管理者)の役割を明確にし、必要に応じてオペレーションマニュアルを作成する
- できるだけシームレスに連携できるように体制・プロセスを構築する

例：ユースケースと開発・運用切り分け

No	ユースケース	開発・運用切り分け		オペレーションマニュアル作成対象
		開発	運用	
1	公開APIの開発環境を作成する		○	○
2	公開APIの本番環境を作成する		○	○
3	内部APIをリリースする（バックエンドAPIのプロキシー）	○		
4	公開APIを新規作成する（パラメータ変換処理等を含む）	○		
5	公開APIを更新する（パラメータ変換処理等を含む）	○		
6	公開APIをリリースする		○	○
7	公開APIを利用会社に通知する		○	○
8	公開APIの利用状況を確認する		○	○
9	公開エンドポイントを死活監視する		○	
10	公開APIのパフォーマンスを監視する		○	
11	ユーザ(APIキーも含む)・ロールを追加・変更する		○	○
13	障害対応を行う（ログ取得、サポート問合せ）		○	○

オージス総研

IBM API Connect を使ったAPI管理

APIのライフサイクルを完全にカバーする API管理プラットフォーム

オージス総研

作成

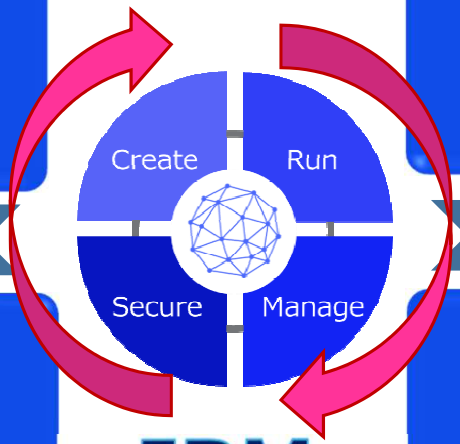
APIを楽々開発

- 既存のデータストアやサービスから容易にAPI開発を行えるツール
- ビルド、テスト、デプロイ
- StrongLoop An IBM Company のNode.jsを提供

実行

APIを安定・高速稼動

- APIの実行環境の管理
- 監視、スケーリング
- 膨大なAPIが稼動可能



万全のセキュリティ

- 実績多数の堅牢なゲートウエイ
- API利用者の認証とアクセス制御
- アクセス数のレート制限

保護

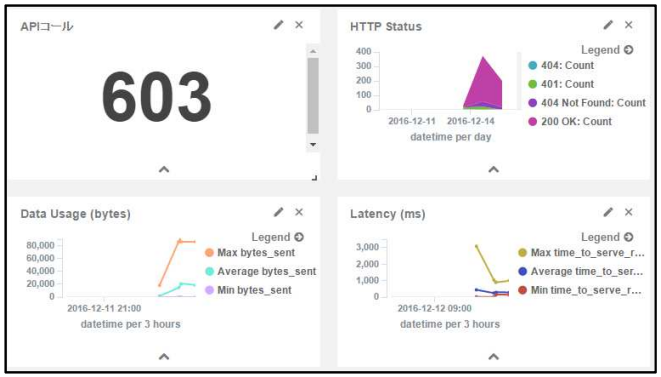
緻密な利用管理

- 公開APIの容易なポリシー定義
- 利用状況の分析、課金
- 開発者ポータルへの自動連携

管理

API Connect のAPIの管理モデル

- APIのグルーピング
  - APIのバージョン
  - 環境(本番、開発)
  - 利用プラン(無料、エンタープライズ)
- APIランタイムの管理項目
  - 死活監視
  - トラフィック量
  - ターンアラウンドタイム
- API利用者の管理項目
  - API開発者
  - 開発者の組織
  - APIクライアント
  - APIキー



API Analytics

APIプランの管理

セキュリティモデル

- クライアントに対してアクセス制御するケースでは、セキュリティレベルや実装コストのバランスを検討し、選択する必要がある。
- エンドユーザーに対してアクセス制御するケースでは、OAuth2 がデファクトスタンダードである。

種類	標準仕様	トークン/クレデンシャル有効期限	リクエストの改ざん防止	スコープによるアクセス制御
API Key（ユニーク文字列)	なし	実装によるがないケースが多い	ない(SSL/HTTPSと組み合わせて実現)	なし
Basic認証	RFC7235	ない	ない(SSL/HTTPSと組み合わせて実現)	なし
OAuth2	RFC6749	有り	ない(SSL/HTTPSと組み合わせて実現)	有り

APIの開発、テスト、デプロイを支援するツール

APIの開発

- HTTPリクエストレスポンスのハンドリング
- データリソースへのアクセス
- バリデーション
- 認証・認可
- ログ
- キャッシュ

APIのテスト

- HTTPリクエストを送信し、実際にAPIを叩くテスト

APIのデプロイ

- APIを実行環境に配置する

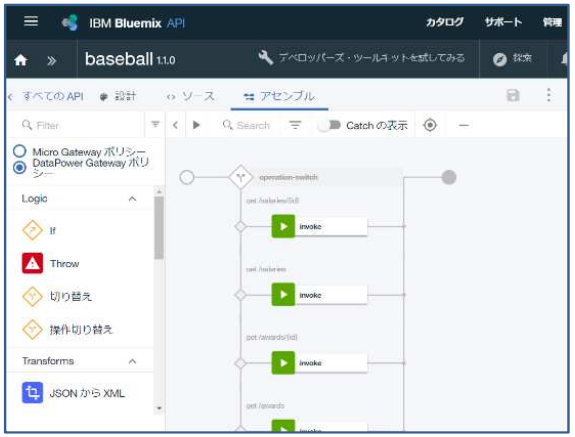
API Tool Kit

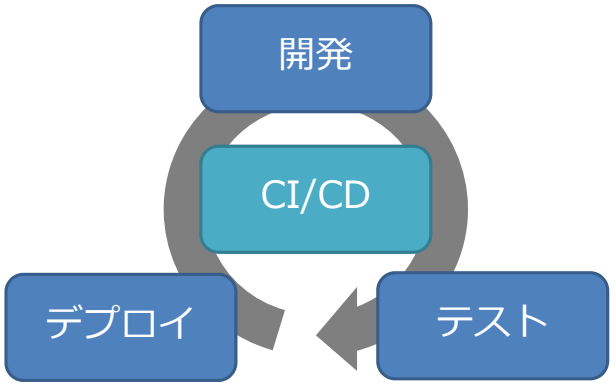
- API Designer

  → API をグラフィカルに開発

- API CLT

  → CI/CDに組み込みやすいツール



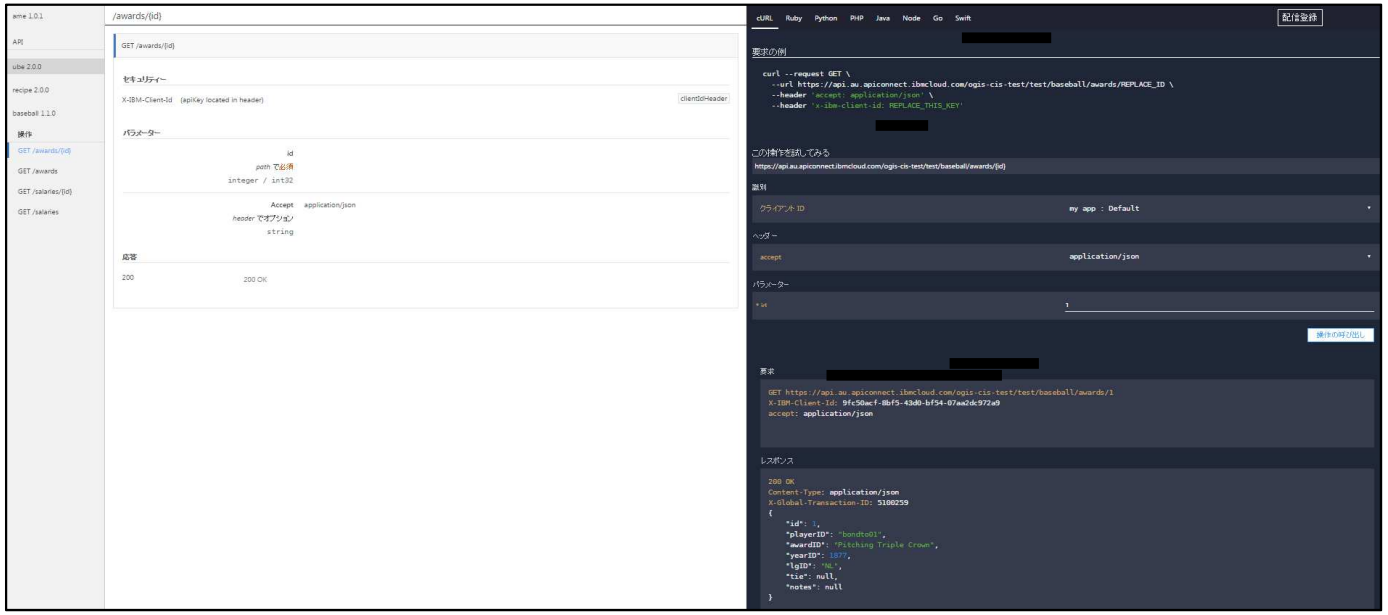


シームレスなAPI定義とドキュメント

- API定義言語 Open API Spec(Swagger)を利用することで、インタフェース実装とドキュメントがリンクする。
- APIのドキュメントを管理者ポータルで提供することで、APIコールも試すことができる

APIの設計ツール(Swagger)

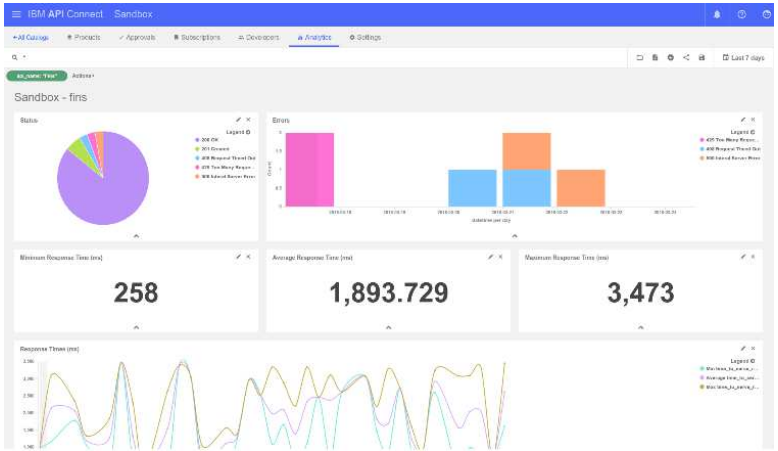
開発者
ポータル
へ連携



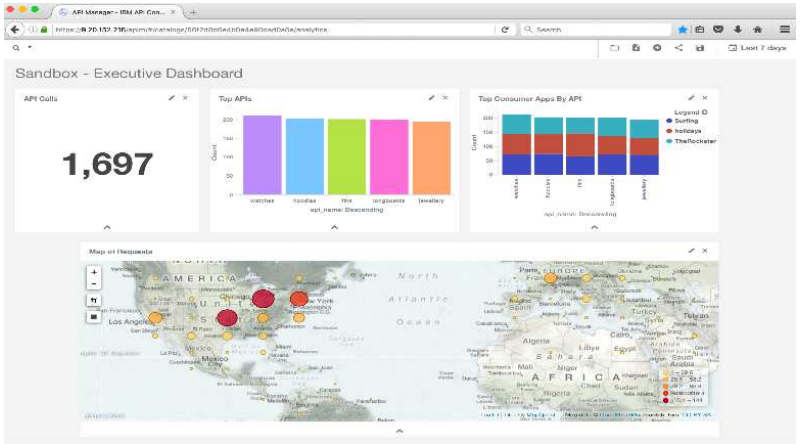
開発者ポータル

API利用監視・分析

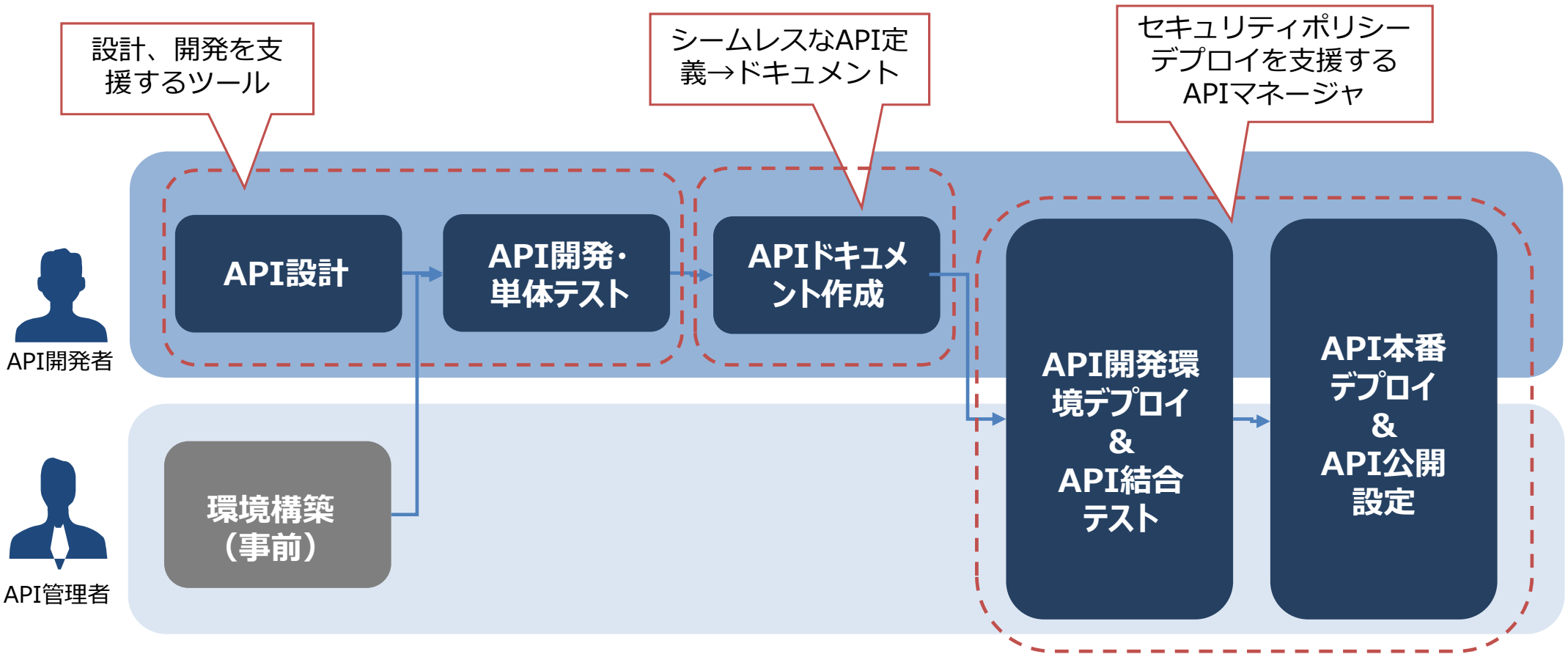
- **APIの利用状況を可視化する分析ダッシュボード**
  - APIのイベントデータを元に様々な軸による解析機能を提供
  - APIやクライアント単位でのアクセス頻度からAPIの利用状況やトレンドを把握
  - APIの応答時間や応答コードからAPIの健全性を確認
  - 時間やメッセージ・コンテンツ、各種メタ情報によるフィルタリングも可能



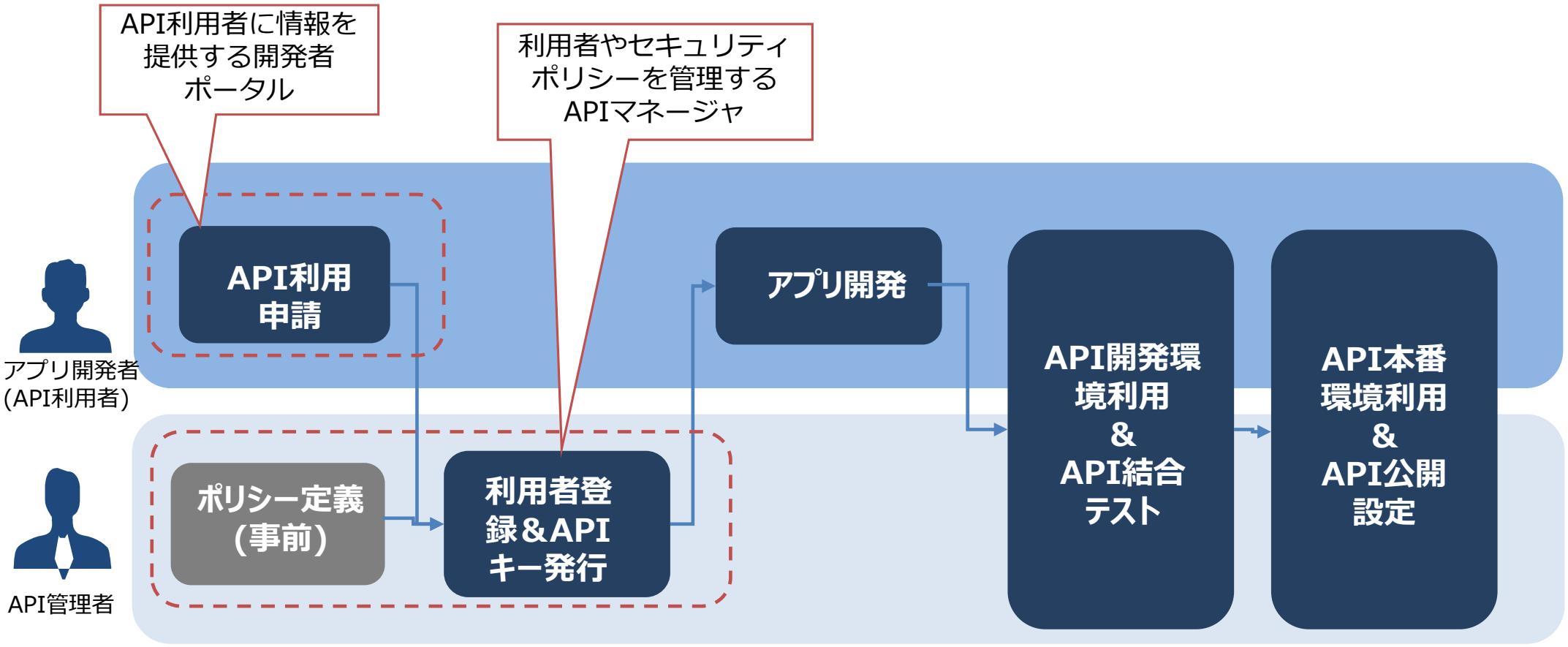
- **多様なレポーティング・ニーズを効果的にサポート**
  - ダッシュボード上の分析グラフやデータはカスタマイズ可能
  - デフォルトで提供されるテンプレートによって基本的な分析をすぐに開始
  - 作成したダッシュボード定義を保管し、他の管理者と共有
  - イベントデータのエクスポートや、外部分析システムへの転送も可能



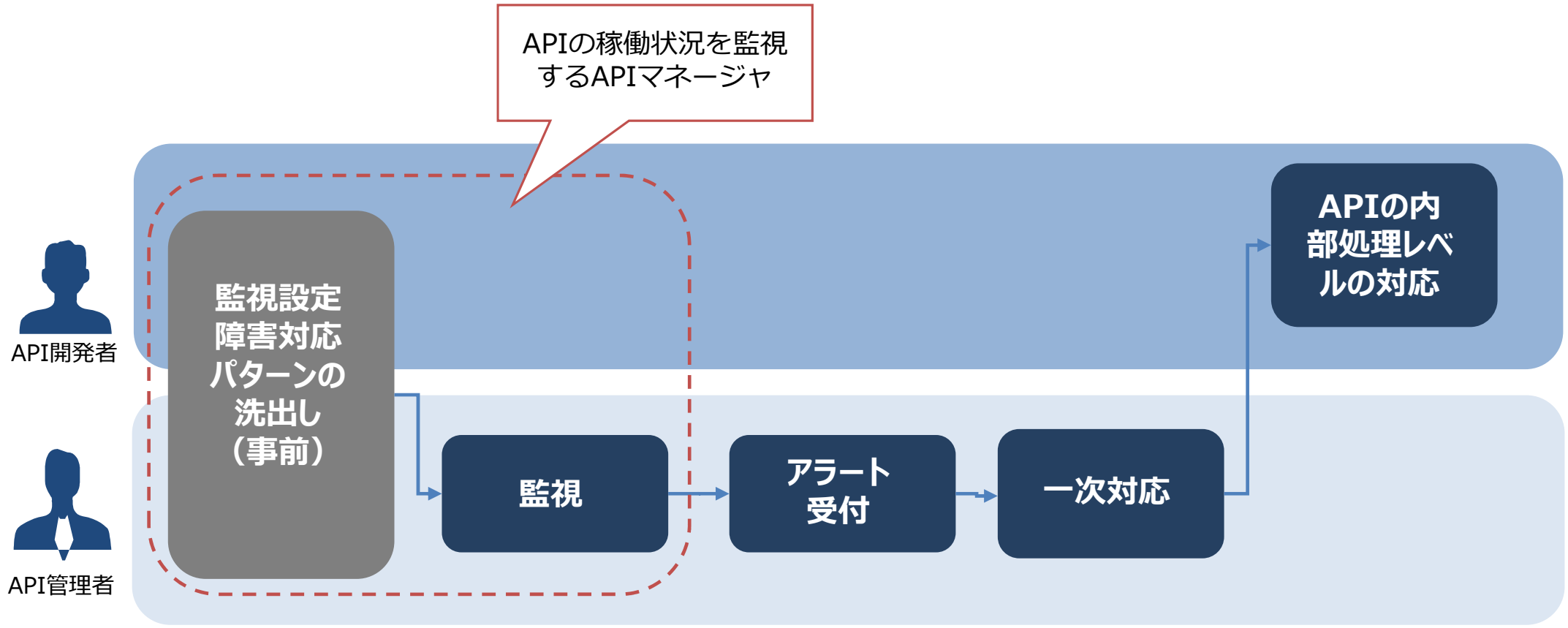
APIの開発プロセスにおけるAPI Connect



API利用プロセス



障害・アラート対応プロセス



オージス総研

オージス総研の API運用サービスのご紹介

サービスメニュー

■ API構築サービス

APIプラットフォーム構築	クラウド、ハイブリッド、オンプレミスに対応 要件にマッチした製品/サービスを利用してご提供
API開発	API開発のベストプラクティスを詰め込んだAPI開発スタックを使い、効率的にAPIを開発

■ API運用サービス

APIアップデート	APIのデータ項目の追加などAPIのアップデート、リリースの実施
APIプラットフォームメンテナンス	定期的なAPIプラットフォームのセキュリティアップデート
APIプラットフォーム監視	APIプラットフォームの障害・異常検知および通知
API障害分析、対応	障害発生時の原因調査、切り分けおよび復旧対応
API利用状況レポート	APIの利用状況のレポーティング
APIユーザーサポート	API利用者への問い合わせ対応、APIクライアント開発支援

APIの安定的に運用

- APIに早くから取り組み、ノウハウを蓄積してきたオージス総研だからできること：**「APIを安心・安全」**に運用
  - APIアップデート
    → データ項目の追加など軽微なAPI更新作業、リリース作業の実施
  - APIプラッフォームメンテナンス
    → セキュリティパッチの適用、設定のバックアップ
  - APIプラットフォーム監視
    → APIプラットフォームの障害・異常検知および通知
  - APIの障害分析・対応
    → 障害発生時の原因調査、切り分けおよび復旧対応

運用にプラスα

- API利用状況レポート
  - APIの利用状況のレポーティング
- APIユーザーサポート
  - API利用者への問い合わせ対応、APIクライアント開発支援