

SAP BTP 解説

SAP Business Technology Platform の全体像・技術仕様・制約・導入設計

2026年6月

はじめに： BTP が生まれた理由

SAP は 2000 年代以降、S/4HANA・Ariba・SuccessFactors・Concur 等のクラウドアプリケーションをそれぞれ独立した SaaS として展開してきました。しかしこの「製品ごとに独立したクラウド」というアーキテクチャは、企業にとって深刻な問題をもたらしました。各アプリケーション間のデータがサイロ化し、統合にコストがかかり、AI・データ分析の基盤もばらばらでした。また、S/4HANA をはじめとするコアシステムに手を加える ABAP カスタマイズは「アップグレードの障壁」となり、SaaS の恩恵（自動アップグレード・クラウドスケール）を享受できませんでした。

SAP BTP（Business Technology Platform）はこの課題を解決するために生まれた「テクノロジー統合基盤」です。アプリケーション開発・統合・データ分析・AI という四つの能力を一つのプラットフォームに集約し、SAP 製品群と企業の独自システムをつなぐ「デジタルコア」として位置付けられています。

本稿では、BTP を深く理解するために必要な「アーキテクチャ・各サービスの技術仕様・制約・コスト構造・実装パターン」まで踏み込んで解説します。「BTP で何ができ何ができないのか」「どのサービスをいつ使うべきか」という実践的な判断基準を提供することを目的とします。

1. SAP BTP の構造: アカウント階層と課金モデル

グローバルアカウント・ディレクトリ・サブアカウントの三層構造

BTP の管理単位は「グローバルアカウント (Global Account) →ディレクトリ (Directory) →サブアカウント (Subaccount)」という三層階層で構成されます。この階層の理解は BTP ガバナンス設計の出発点です。

- グローバルアカウント: SAP との契約単位。エンタイトルメント (利用権・ライセンス) がここに割り当てられます。グローバルアカウント管理者は全サブアカウントの管理権限を持ちます。一企業で複数のグローバルアカウントを持つケースは稀ですが、子会社・M&A 先の統合時に問題となることがあります。
- ディレクトリ: サブアカウントをグループ化する中間管理層 (オプション)。「事業部別」「環境別 (開発・検証・本番)」「地域別」等の管理単位に応じてディレクトリを設計します。ディレクトリへのエンタイトルメント割り当て (子サブアカウントへの均等配分) も可能です。
- サブアカウント: BTP サービスの実際の利用・デプロイ単位。各サブアカウントは独立したリージョン (データセンター地域)・環境 (Cloud Foundry / Kyma / ABAP)・セキュリティ設定を持ちます。開発・検証・本番ごとに別サブアカウントを作成する設計が標準です。

エンタイトルメントとサービスプランの仕組み

BTP サービスを利用するには、グローバルアカウントに「エンタイトルメント (Entitlement)」が付与され、それをサブアカウントに割り当てる必要があります。エンタイトルメントは SAP との契約で決まる「利用可能なサービスとその量・プラン」の定義です。

各 BTP サービスには複数の「サービスプラン (Service Plan)」があり、プランによって機能範囲・リソース上限・課金方式が異なります。

- Free プラン: 開発・PoC 目的の無料枠。本番利用は不可・性能制限あり・SLA なし。フリープランで PoC 後に有償プランへ移行するパターンが一般的。
- Standard プラン: 本番利用向け有償プラン。機能フルセット・SLA 保証 (可用性 99.9%以上等)。

- Premium プラン：高性能・大容量・追加機能（一部サービスのみ）。

消費ベース課金モデル（CPEA・BTPEA）

BTP の課金モデルは大きく「消費ベース（Consumption-Based）」と「サブスクリプション（Subscription-Based）」の二種類があります。

- CPEA（Cloud Platform Enterprise Agreement）：消費ベース契約の旧名称。グローバルアカウントにクレジットプールを購入し、実際に使用した BTP サービスの従量分がクレジットから消費される方式。使用しないサービスのコストが発生しない柔軟性がある一方、クレジット消費速度の予測が難しく、計画外のコスト増が発生するリスクがあります。
- BTPEA（BTP Enterprise Agreement）：2023 年以降の CPEA 後継モデル。基本構造は CPEA と同様の消費ベースですが、対象サービス範囲が拡大され、購入クレジットの有効期間設定が改善されています。
- サブスクリプション：特定サービスを固定料金・固定量で年間契約する方式。Integration Suite のメッセージ数・Build Process Automation のワークフロー件数等、量が予測できる場合はサブスクリプションの方がコスト管理しやすい。

コスト管理の重要ポイントとして、BTP Cockpit の「Cost & Usage」画面でサービス別消費状況をリアルタイムモニタリングする運用が不可欠です。特に開発者が自由にサービスを有効化できる設定になっていると、想定外のクレジット消費が発生します。サブアカウント単位でのエンタイトルメント量の上限設定（Quota）によるコスト制御が推奨されます。

2. 実行環境: Cloud Foundry・ABAP・Kyma の技術比較

三つの実行環境の役割と選択基準

BTPには三つの実行環境（Runtime）が存在し、開発する拡張アプリケーションの性質・開発チームのスキル・将来の保守性によって使い分けます。三つのうち複数と同時に使用するアーキテクチャも一般的です。

Cloud Foundry 環境（CF）

Cloud Foundry（CF）はPaaS（Platform as a Service）として、アプリケーションのデプロイ・スケーリング・ライフサイクル管理を提供します。BTPにおいて最も長い歴史を持ち、最多のBTPサービス（Integration Suite・SAP Work Zoneの特定コンポーネント等）がCF上で稼働しています。

- サポート言語・フレームワーク: Java（Spring Boot・TomEE）・Node.js・Python・Go・Ruby。「Buildpack」というミドルウェア設定テンプレートで各言語環境を自動構成します。
- メモリ・CPU管理: アプリケーションごとにメモリ割り当て（例: 512MB・1GB）を設定します。「Application Instance」の数でスケールアウトします（水平スケーリング）。垂直スケーリング（インスタンスサイズの変更）はアプリ再起動が必要です。
- CF制約: コンテナの実行時間制限（ディスクI/O・長時間バッチ処理には向かない）・ファイルシステムはエフェメラル（再起動でリセット）・コンテナ間通信はHTTP/HTTPSのみ（ソケット通信は原則不可）。
- サービスバインディング: データベース・メッセージキュー・認証等のBTPサービスを「サービスバインディング（Service Binding）」でアプリケーションに接続します。接続情報（接続文字列・認証情報）はCF環境変数「VCAP_SERVICES」に自動注入されます。

ABAP Environment (ABAP クラウド環境)

ABAP Environment は BTP 上で ABAP コードを実行する環境です。オンプレミス S/4HANA の ABAP とは異なり「ABAP クラウド (Steampunk)」と呼ばれる制限付き ABAP サブセットのみが使用可能です。

- ABAP クラウドの制限: 従来の ABAP で使用できた「Function Module・BAPI・内部テーブルの直接 DDIC アクセス・トランスポート層への直接アクセス」等の多くが ABAP クラウドでは利用不可または非推奨です。代わりに RESTful ABAP Programming Model (RAP)・CDS (Core Data Services)・Business Services (OData V4) による開発が標準です。
- RAP モデル: Restful ABAP Programming Model は BTP での ABAP アプリ開発の中核フレームワークです。CDS View でデータモデルを定義し、Behavior Definition でビジネスロジック (CRUD・アクション・バリデーション) を実装し、Service Definition で自動的に OData V4 API を公開します。Fiori Elements UI との親和性が高く、「データモデル定義だけでリスト・詳細・編集画面が自動生成される」という生産性が特徴です。
- S/4HANA オンプレミスとのコード共用: オンプレミス S/4HANA の ABAP コードを ABAP クラウドに移植するには、「ABAP Test Cockpit (ATC)」で ABAP クラウド互換性チェックを行い、非互換 API の置き換えが必要です。大規模な既存 ABAP コードベースの移植は相当な工数を要することが多い点が制約です。

Kyma 環境 (Kubernetes)

Kyma は「Kubernetes (k8s) ベースのコンテナ実行環境」です。BTP のランタイムの中で最もインフラ的な柔軟性が高く、コンテナ化されたマイクロサービスを任意の言語・フレームワークでデプロイできます。

- アーキテクチャ: SAP 管理の Kyma Kubernetes Cluster が提供されます。ユーザーは Helm Chart・Kubernetes Manifest (YAML) でアプリケーションをデプロイします。Istio サービスメッシュ (トラフィック管理・mTLS 通信暗号化・オペレーバビリティ) が標準搭載されています。
- Eventing (イベント駆動): SAP Event Mesh (後述) または Kyma のネイティブ Eventing (NATS/CloudEvents 標準) で S/4HANA や BTP サービスからのイベントを Kyma アプリが受信できます。「S/4HANA で BP (ビジネス

パートナー) が作成されたら Kyma アプリがリアルタイムにトリガーされる」というイベント駆動アーキテクチャが実現します。

- スケーリング: Kubernetes HPA (Horizontal Pod Autoscaler) によるオートスケーリング・ゼロスケール (アイドル時にポッドを 0 にしてコスト削減) が利用できます。ただしゼロスケール状態からのコールドスタートには数秒~数十秒の遅延が発生します。
- Kyma の難易度: Kyma は三つの実行環境の中でインフラ知識 (Kubernetes・Docker・Helm) が最も要求されます。SAP BTP 開発者に Kubernetes スキルがない場合は、Cloud Foundry を優先選択するか、Kyma の学習コストを導入計画に組み込む必要があります。

実行環境の選択ガイドライン

三環境の選択基準を整理します。

- Cloud Foundry: Java または Node.js での標準的な Web API・UI アプリ開発。BTP サービス (Integration Suite 等) のサポートが必要な場面。開発チームに Kubernetes の知識がない場合。
- ABAP Environment: SAP 標準の ABAP で開発された既存ロジックのクラウド移植。S/4HANA のカスタム OData API を公開する場合。ABAP スキルを持つ開発チームが主体の場合。
- Kyma: マイクロサービスアーキテクチャ・コンテナファースト設計・イベント駆動型のリアルタイム処理・任意の言語 (Python/Go/Rust 等) での開発が必要な場合。長時間バッチ・GPU コンピューティング等の CF 非適合ワークロード。

3. SAP Integration Suite: 統合ミドルウェアの詳細

Integration Suite のコンポーネント全体像

SAP Integration Suite (IS) は BTP の統合機能の中核を担うサービスセットです。企業と SAP・非 SAP・オンプレミス・クラウドを問わないシステム間の統合を提供します。サービスとして提供される統合プラットフォーム (iPaaS) の位置付けです。

Cloud Integration (iFlow) : 統合フローの詳細

Cloud Integration (旧称: HCI / SAP Cloud Platform Integration) は「iFlow (Integration Flow)」と呼ばれる統合処理フローをグラフィカルに設計・実行するエンジンです。iFlow は BPMN2.0 ライクな設計ツールで「受信アダプター→処理ステップ→送信アダプター」という構造を持ちます。

アダプタの種類と制約

iFlow が接続できるアダプタ (コネクタ) は 70 種類以上あります。主要アダプタと技術特性を示します。

- HTTPS/REST: 最も汎用的。REST API・Webhook・ポーリングによるデータ取得に対応。OAuth 2.0・API キー・Basic 認証をサポート。
- SFTP/FTP: レガシーシステムとのファイル連携。「ポーリング間隔・ファイルパターン (*.csv) ・処理後の削除/アーカイブ」を設定します。SFTP 秘密鍵認証もサポート。
- IDoc (SAP) : S/4HANA の IDoc 送受信に特化したアダプタ。IDoc タイプを指定し、S/4HANA の RFC 宛先と接続します。IDoc のパース・変換・エラーハンドリングが iFlow で制御できます。
- OData: OData V2/V4 での CRUD 操作。SAP・非 SAP の OData API への接続に使用。Batch Request・\$expand 等の OData 機能も利用可能。
- AS2 (EDI) : B2B の電子商取引 (EDI) で使用する暗号化・署名付き通信プロトコル。取引先との電子商取引ドキュメント (受発注・請求書) の自動交換。

- AMQP/Kafka: メッセージキューブローカー (Azure Service Bus・AWS SQS・Apache Kafka 等) との非同期連携。高スループット・高可用性のイベント連携に使用。

アダプタの制約として「全アダプタが全サービスプランで利用可能ではない」点に注意が必要です。Enterprise 版では利用可能なアダプタが、Standard プランや Trial ではブロックされるものがあります。プロジェクト開始前に使用予定のアダプタがエンタイトルメントに含まれているかを確認する必要があります。

iFlow の処理ステップ: 変換・ルーティング・エラーハンドリング

iFlow 内で利用できる主要処理ステップの種類を示します。

- Message Transformation: XML⇄JSON 変換・メッセージマッピング (Graphical Mapping・XSLT・Groovy Script) ・Content Modifier (ヘッダー・プロパティの追加/修正) ・Filter (条件に合致する要素のみ抽出) 。
- Routing: 「Content-Based Router (メッセージ内容によって送信先を切り替え)」 「Splitter (大きなメッセージを複数の小さなメッセージに分割)」 「Aggregator (複数メッセージを一つに集約)」 「Multicast (同一メッセージを複数宛先に並列送信)」 。
- Persistence: 「Persist (メッセージの中間保存)」 「Data Store (Key-Value での一時データ保存)」 「Local Variable (iFlow 実行中の変数管理)」 。
- エラーハンドリング: iFlow はエラー時の挙動を「Exception Subprocess」で定義します。リトライ設定 (最大リトライ回数・リトライ間隔) ・デッドレターキュー (リトライ上限後の隔離) ・アラート通知 (電子メール・SAP Alert Notification) が構成できます。

操作イメージ: iFlow の設計と運用

Integration Suite 上で iFlow を設計・デプロイする基本的な流れを示します。

- Package の作成: 統合シナリオをグループ化する「Integration Package (統合パッケージ)」を作成します。パッケージ内に iFlow・Message Mapping・Value Mapping 等のアーティファクトを格納します。
- iFlow の設計: 「Add」→「Integration Flow」で iFlow を追加し、Web ブラウザベースの UI エディタで設計します。パレットからアダプタ・処理ステ

ップをドラッグ&ドロップで配置し、プロパティパネルで各ステップの詳細設定（エンドポイント URL・認証方式・変換ロジック等）を入力します。

- シミュレーション：デプロイ前にテストメッセージで iFlow の処理をシミュレーションできます。各ステップの入出力メッセージを可視化することでデバッグが効率化されます。
- デプロイ：「Deploy」ボタンで iFlow を BTP にデプロイします。デプロイは CF 上の Container (Osgi Bundle) として実行されます。デプロイ完了後、エンドポイント URL が発行され、外部システムからの呼び出し・スケジューラによる定期起動が可能になります。
- モニタリング：「Monitor > Integrations」で実行中・完了・エラーのメッセージ処理状況をリアルタイム確認できます。エラーメッセージの詳細ログ（受信データ・変換後データ・エラースタックトレース）を確認してデバッグします。

API Management: API ゲートウェイの詳細

API Management (APIM) は企業の API を「管理・保護・公開・収益化」する API ゲートウェイです。S/4Hana や BTP サービスの OData API を APIManagement 経由で外部公開することで、認証・レートリミット・分析・バージョン管理を一元管理します。

- API プロキシ (API Proxy)：バックエンド API の前段にプロキシを配置し、ポリシー (Policy) を適用します。ポリシーは「Security (OAuth 認証・API キー検証・IP 制限)」「Traffic Management (レートリミット・クォータ・スパイクアレスト)」「Mediation (リクエスト/レスポンスの変換・ヘッダー操作)」「Extension (Groovy/JavaScript 実行)」の四カテゴリに分類されます。
- Developer Portal (API カタログ)：API 利用者（社内開発者・外部パートナー）向けのセルフサービスポータル。API 仕様 (OpenAPI/Swagger) の参照・テスト実行・API キー発行申請が可能です。API エコシステムの構築 (PartnerAPI 公開・内部 API 活用促進) に活用されます。
- API Analytics：API コール数・レスポンスタイム・エラー率・利用者別トラフィック等の API 利用分析ダッシュボード。SLA モニタリング・異常検知アラート設定が可能です。
- 制約：API Management のレートリミット・クォータ設定は API Proxy レベルでの設定です。複数の APIProxy をまたいだグローバルレートリミット

(ユーザー全体の API コール総量制限) は設定できません。また、GraphQL API の管理は限定的なサポートとなっています。

Event Mesh: イベント駆動統合の技術詳細

SAP Event Mesh は BTP のメッセージブローカー (Message Broker) です。CloudEvents 標準に準拠したイベントの発行 (Publish) ・配信 (Subscribe) ・永続化 (Persistence) を提供します。

- メッセージキューとトピック: Event Mesh は「Queue (キュー、P2P 配信)」と「Topic (トピック、Pub/Sub 配信)」の二つのメッセージングモデルをサポートします。Queue は送信したメッセージが一つの受信者に配信されます。Topic は複数のサブスクライバーが同じメッセージを受信できます (Fan-out)。
- S/4HANA との統合: S/4HANA 2020 以降では「Business Events (ビジネスイベント)」機能により BP の作成・変更・注文の確認等のビジネスイベントが Event Mesh Topic に自動発行されます。Kyma や Cloud Foundry アプリがそのイベントをリアルタイムに受信してリアクティブな処理を実行できます。
- メッセージ永続化: Event Mesh はメッセージを一定期間 (設定可能) 永続化します。サブスクライバーが一時的にダウンしていた間に発行されたメッセージを後から受信できます。この「少なくとも一度の配信 (At-Least-Once)」保証が非同期処理の信頼性を高めます。
- 制約: Event Mesh のメッセージサイズ上限 (デフォルト 1MB) ・キュー深度 (最大メッセージ保持件数) ・スループット上限はサービスプランにより異なります。大容量ファイルの転送には Event Mesh は適さず、SFTP/Object Storage との組み合わせが必要です。

Open Connectors: 非 SAP システムとの接続

SAP Open Connectors は Salesforce ・ ServiceNow ・ Slack ・ Box ・ Google Drive ・ HubSpot 等の 170 種類以上の SaaS 製品に対する標準コネクタを提供します。

Open Connectors の特徴は「各 SaaS の API を正規化 (Normalize) して統一 API として提供する」点です。例えば Salesforce と HubSpot はそれぞれ異なる API 設計を持っていますが、Open Connectors を経由することで同一の API 設計でどちらの

CRMにもアクセスできます。iFlowとの統合により「外部 SaaS のデータを S/4HANA に同期する」統合フローが標準コネクタで構築できます。

制約として、Open Connectors が提供するコネクタは「読み取り・基本 CRUD」が中心であり、接続先 SaaS の高度機能（バッチ API・Webhook 等）については個別サポート状況の確認が必要です。また、コネクタの仕様変更に従うアップデートタイミングが SAP 側の対応速度に依存する点が制約となります。

4. SAP Build: ローコード開発プラットフォームの詳細

SAP Build Apps の技術詳細

SAP Build Apps (旧称: AppGyver) はコード不要 (No-Code) ~ローコード (Low-Code) で Web アプリ・モバイルアプリを開発するビジュアル開発ツールです。

- UI コンポーネント: ドラッグ&ドロップで配置可能な UI コンポーネント (ボタン・テーブル・フォーム・チャート・マップ等) が 200 種類以上提供されます。カスタム CSS・カスタム JavaScript (Formula 関数) の注入も可能で、ノーコードの制限を超えたカスタマイズができます。
- データ接続: OData V2/V4・REST API・GraphQL・BTP Destination (接続設定) を通じて S/4HANA や外部システムのデータをリアルタイム取得・更新できます。BTP 上の CAP アプリ (SAP Cloud Application Programming Model) との連携が最も統合性が高く推奨されます。
- ビジネスロジック: 「Flow Function (処理フロー)」を視覚的に定義します。IF/Else 分岐・ループ・変数操作・API 呼び出し・エラーハンドリングを視覚的なフローチャートで記述します。複雑なロジックには JavaScript Formula 関数を利用します。
- モバイル出力: Web ブラウザに加え、iOS アプリ・Android アプリとしてビルド・配布できます。OTA (Over-The-Air) アップデートにより、ストア再申請なしにアプリコンテンツを更新できます。
- 制約: Build Apps はオフライン対応 (オフラインデータキャッシュ・オフライン操作のキュー保存) をサポートしますが、複雑なオフラインシナリオ (大量レコードの差分同期・コンフリクト解決) の設計は課題が多くなります。また、非常に高度な 3D ビジュアライゼーションやリアルタイム AR 機能は対応していません。

SAP Build Process Automation (SBPA) の技術詳細

SAP Build Process Automation (SBPA) はワークフロー自動化と RPA (Robotic Process Automation) を統合したサービスです。ビジネスプロセスの承認フロー・タスク管理・ボット自動化を一つのプラットフォームで設計・運用できます。

ワークフロー (Workflow Management) の仕組み

- プロセス設計: BPMN ライクなビジュアルエディタでワークフローを設計します。「ユーザータスク (承認・入力)」「自動ステップ (API コール・条件分岐)」「並列分岐 (複数承認者への同時割り当て)」「タイマー (期日アラート・エスカレーション)」を組み合わせます。
- フォームデザイナー: ワークフロー内のユーザータスクに表示する「入力フォーム」をノーコードで設計します。フィールドタイプ (テキスト・数値・日付・選択肢・添付ファイル)・バリデーション・条件表示 (特定条件の場合のみフィールドを表示) が設定できます。
- My Inbox (タスクポータル): 承認担当者は SAP Build Work Zone の My Inbox またはモバイルアプリから承認待ちタスクを処理します。タスク一覧・タスク詳細・承認/却下/転送の操作がシンプルな UI で提供されます。
- プロセス可視性 (Process Visibility): 現在実行中のワークフローインスタンスのステータス (どのステップで止まっているか)・過去実行の履歴・平均処理時間・ボトルネック分析をダッシュボードで確認できます。

RPA (Desktop Automation) の技術詳細

- SAP の RPA は「Cloud Studio (ボット設計)」と「Desktop Agent (ボット実行)」の二つのコンポーネントで構成されます。Desktop Agent はボットを実行する PC にインストールするエージェントソフトウェアです。
- 自動化対象: 画面操作の記録 (Record & Play) で自動化シナリオを作成します。Web ブラウザ (Chrome/Edge)・Windows デスクトップアプリ・SAP GUI・Excel・PDF 操作を自動化できます。
- SAP GUI 自動化: SAP GUI (BEx・SE11・ME21N 等のトランザクション画面) の自動操作は「SAP GUI Automation」コンポーネントで専用サポートされています。画面上の要素 (フィールド・ボタン・テーブル行) を Technical Name (ID) で特定するため、画面レイアウト変更に対する堅牢性が高いです。

- 制約: RPA ボットは Desktop Agent が起動している PC でのみ実行されます。PC がシャットダウン・ロックされているとボットは停止します。サーバーサイドでの常時稼働には VM (Virtual Machine) 上の Agent 実行が必要です。また、キャプチャ (CAPTCHA) ・多要素認証 (MFA) が必要な画面の自動化は困難です。

SAP Build Work Zone: デジタルワークプレイスの詳細

SAP Build Work Zone (旧称: SAP Launchpad Service) は SAP Fiori アプリ・BTP アプリ・外部 Web アプリ・コンテンツ (ニュース・ナレッジ記事) を一つのポータル画面に統合する「デジタルワークプレイス」です。

- Site Designer: Work Zone のポータル UI (Site) をノーコードで設計します。ページ・セクション・カード・ウィジェットをドラッグ&ドロップで配置し、役割 (Role) に応じたページの出し分け (Personalization) を設定します。
- ビジネスコンテンツ統合: S/4Hana の Fiori アプリ・BTP のカスタムアプリ・SAP Analytics Cloud の Analytics ダッシュボード・Ariba・SuccessFactors の各ポータルへのリンクを一元化します。ユーザーは Work Zone にログインすることで、複数システムへのシングルアクセスポイントを得ます。
- My Inbox 統合: SBPA のワークフロー承認タスク・S/4HANA のワークフロー (SWDD/SWI5 相当) を Work Zone の My Inbox に統合表示できます。承認作業の One-Stop 化が実現します。
- 制約: Work Zone の「Standard 版」と「Advanced 版」で機能が大きく異なります。Advanced 版のみ利用できる機能 (カスタムコンテンツ (Page Builder) ・HR 統合・Microsoft 365 統合・AI 推薦コンテンツ) が多数あります。ライセンスプランの選定に注意が必要です。

5. SAP HANA Cloud: クラウドデータプラットフォームの詳細

HANA Cloud のアーキテクチャと特性

SAP HANA Cloud (HC) はインメモリ列指向データベースを中核とするクラウドネイティブ DBaaS です。オンプレミス SAP HANA (S/4HANA の基盤となる DB) のクラウド版として位置付けられますが、アーキテクチャは大きく異なります。

- **コンピューティング・ストレージ分離:** HANA Cloud はコンピューティング (CPU・インメモリ) とストレージ (ディスク) が分離されたアーキテクチャを採用しています。データは「Native Storage Extension (NSE)」によってホットデータ (インメモリ) ・ウォームデータ (ディスク) ・コールドデータ (オブジェクトストレージ) の三層に自動的にティアリングされます。これにより、全データをメモリに載せるコストを抑えながら高速クエリを維持します。
- **マルチモデル:** リレーショナル (SQL) ・グラフ (Cypher/SPARQL) ・空間 (Spatial/GIS) ・ドキュメント (JSON) の複数データモデルを単一 DB エンジンで扱います。グラフ分析と空間分析を SQL と組み合わせて単一クエリで実行できます。
- **HANA Cloud Data Lake (QRC) :** コールドデータを SAP HANA Data Lake (HDL) に格納し、HANA Cloud 経由で SQL 連携するアーキテクチャ。大容量の過去データを低コストで保持しながら、必要に応じて HANA Cloud 上で JOIN クエリが実行できます。
- **拡張モデル:** BTP アプリ (CAP アプリ等) から HANA Cloud への接続は HDB (HANA Database) ドライバを紹介します。Node.js (@sap/hana-client) ・Java (NGDBC) ・Python (hdbcli) ドライバが提供されています。

BTP 上の CAP アプリと HANA Cloud の統合

「SAP Cloud Application Programming Model (CAP)」は BTP でのアプリケーション開発フレームワークです。CAP は Node.js (cds npm) または Java (Spring Boot 拡張) で記述し、BTP の CF/Kyma にデプロイします。

CAPはHANA CloudをネイティブDBとして利用するように設計されています。CDS (Core Data Services) でデータモデルを定義すると、HANA CloudのテーブルDDLが自動生成され、OData V4 APIが自動的に公開されます。「データモデル定義→APIが自動生成される」というRAD (Rapid Application Development) 的な開発効率が特徴です。

- マルチテナント: CAPはBTPのマルチテナントSaaS拡張アプリの開発フレームワークとしても利用されます。テナント分離はHANA Cloudのスキーマ分離 (テナントごとの独立スキーマ) で実現します。SAPのISVパートナーがBTP上でSaaS製品を構築する際の標準フレームワークです。
- 制約: CAPの生産性は高いですが、HANA Cloud固有の高度な機能 (Advanced Analyticsビュー・Calculation View等) をCAPのCDSから直接活用するには、HDBアーティファクト (.hdbtable・.hdbcalculationview等) との混在設計が必要となり、設計の複雑性が増します。

SAP Datasphere: データファブリックの詳細

SAP Datasphere (旧称: SAP Data Warehouse Cloud) はデータ統合・データモデリング・データ仮想化・データ共有を一元管理するデータファブリックプラットフォームです。

- ビジネスデータファブリック: DatasphereはSAP HANACloud・S/4HANA・Azure Synapse・Snowflake・Databricks等の複数データソースを「仮想テーブル (Remote Table)」として統合します。実際のデータ移動なしに複数DBを横断したSQL/Analytics Viewが作成できます (データ仮想化)。
- SAP BW Bridgeによる移行支援: SAP BWからDatasphereへの移行を支援する「BW Bridge」機能が提供されています。BW InfoObject・BW Query (BEx) をDatasphereのエンティティに変換するツールが用意されており、既存のBW投資を段階的にクラウドに移行できます。
- 知識グラフ (Business Layer): DatasphereのBusiness LayerはSAPアプリケーション (S/4HANA・Ariba・SuccessFactors) のセマンティクス (業務意味付け) を自動取込します。「仕入先・受注・損益」等の業務概念がデータモデルに自動反映され、分析ユーザーが技術的なテーブル名を知らずにデータを理解・活用できます。

- SAC (SAP Analytics Cloud) との統合: Datasphere を SAC のライブ接続先として設定すると、Datasphere のモデルを SAC の Story/Planning からリアルタイム参照できます。Datasphere 側でのデータアクセス制御 (行レベルセキュリティ) が SAC 側にも透過的に適用されます。
- 制約: Datasphere のデータ仮想化 (Remote Table) は、クエリを実行するたびにリモートデータソースにクエリが委譲されます。リモート DB の性能がクエリ性能を制約します。大容量データのアドホック分析にはデータ複製 (Replication Flow) で Datasphere のローカルストレージに取り込む設計が適しています。

6. BTP のセキュリティ・アイデンティティ管理の詳細

SAP Identity Authentication Service (IAS) の詳細

SAP IAS (Identity Authentication Service) は BTP の ID プロバイダー (IdP) です。SAP Cloud 製品 (BTP・S/4HANA Cloud・Ariba・SuccessFactors 等) へのシングルサインオン (SSO) を提供します。

- SAML と OpenID Connect: IAS は SAML 2.0 と OpenID Connect (OIDC) の両プロトコルをサポートします。BTP アプリへの SSO は OIDC (Authorization Code Flow) が主流です。SAP Fiori Launchpad や BTP サービスへの SSO には SAML が使われるケースもあります。
- コーポレート IdP との連携 (Proxy): 社内の Active Directory (Microsoft Entra ID / 旧 Azure AD) や Okta 等の企業 IdP を IAS の PROXY 構成で利用できます。ユーザーは企業の IdP で認証し、IAS を経由して BTP サービスに SSO します。この構成により「ユーザーの ID 管理は既存企業 IdP に委ね、SAP クラウドへの SSO は IAS が担う」という責任分離が実現します。
- MFA (多要素認証): IAS は TOTP (Google Authenticator 等)・SMS OTP・Biometrics (FIDO2/WebAuthn) の MFA をサポートします。特定アプリケーション・特定ユーザーグループ・特定 IP レンジ以外からのアクセスに対して MFA を強制するポリシー設定が可能です。
- Risk-Based Authentication: アクセスのリスクレベル (未知 IP からのアクセス・海外からのアクセス・勤務時間外アクセス等) に応じて認証要求レベルを自動的に上げる「リスクベース認証」がサポートされています。

SAP Authorization and Trust Management (XSUAA)

SAP XSUAA (Extended Services User Account and Authentication) は BTP の OAuth 2.0 認可サーバーです。BTP アプリが発行するアクセストークン (JWT) の検証・スコープ管理・ユーザーと Role Collection のバインディングを管理します。

- Role・Role Collection・Scope: BTP アプリのアクセス制御は「Scope (権限の最小単位) →Role (Scope の集合) →Role Collection (Role の集合) →User/Group (Role Collection の割り当て)」の四層で構成されます。アプリケーション開発者が Scope を定義し、管理者が Role Collection をユーザーに割り当てます。
- アプリ間の認可 (OAuth Client Credentials) : BTP アプリ同士・BTP アプリから S/4HANA API への M2M (Machine-to-Machine) 認証には「OAuth 2.0 Client Credentials Flow」を使用します。XSUAA がクライアント ID・クライアントシークレットに対してアクセストークンを発行し、API ゲートウェイでトークンを検証します。
- Principal Propagation (ユーザー伝播) : BTP から S/4HANA への API 呼び出しにおいて「BTP アプリにログイン中のユーザー ID を S/4HANA 側に伝播する」Principal Propagation 設定が可能です。これにより「BTP アプリから S/4HANA を操作した記録が S/4HANA 側の変更文書に実ユーザー ID で残る」という監査トレーサビリティが確保されます。

Destination Service: 接続先管理の詳細

Destination Service は btp 上のアプリが外部システム (S/4HANA・Ariba・外部 REST API 等) に接続するための「接続設定 (認証情報・エンドポイント URL) を集中管理するサービス」です。

接続設定 (Destination) は Subaccount または Instance (アプリインスタンス) レベルで管理され、アプリケーションコード内に接続先 URL や認証情報をハードコードする必要がありません。環境 (開発・検証・本番) ごとに同じ Destination 名で異なる接続先を定義することで、コードを変更せずに環境切り替えができます。

- 認証タイプ: Basic 認証・OAuth 2.0 (Client Credentials・Authorization Code・SAML Bearer Assertion)・Certificate (クライアント証明書)・Principal Propagation (ユーザー伝播) をサポート。
- On-Premise 接続 (SAP Connectivity Service + Cloud Connector) : Destination が「On-Premise」タイプの場合、BTP からオンプレミスシステム (S/4HANA オンプレミス・社内 REST API 等) への接続は「Cloud Connector (SCC)」を経由します。SCC は社内ネットワークに設置するプロキシソフトウェアで、BTP とオンプレミスシステム間の暗号化トンネルを確立します。インターネットに対してポート開放が不要な「リバーストンネル」方式のため、ネットワークセキュリティ上の懸念が少ない接続方式です。

- 制約: Destination Service はサブアカウント単位で Destination リストを管理します。大規模環境で接続先が多い場合、Destination の命名規則・バージョン管理・変更追跡の仕組みを設計しないと管理が煩雑になります。Destination をソースコードリポジトリで IaC (Infrastructure as Code) 管理するための BTP CLI / BTP Terraform Provider の活用が推奨されます。

7. クリーンコア戦略: BTP による拡張の設計原則

クリーンコアとは何か: SAP の技術戦略

「クリーンコア (Clean Core)」は SAP が推進する S/4HANA の標準コードを「汚染」しない拡張アーキテクチャの設計原則です。従来の SAP 導入では、業務要件に合わせて ABAP コードを修正・追加する「改造 (Modification)」が常態化しており、SAP 標準コードの上書き (EXIT・BAdI・Enhancement Spot) が大量に行われていました。

この改造の蓄積が S/4HANA へのマイグレーション・定期アップグレード (SAP S/4HANA は年 1 回のメジャーアップデート) の最大の障壁となっています。クリーンコア戦略は「S/4HANA のコアコードを標準のまま維持し、拡張はすべて BTP の外側で行う」という方針です。

拡張パターンと技術選択の詳細

クリーンコア準拠の拡張には三つのパターンが定義されています。

- In-App Extension (インアプリ拡張) : S/4HANA 内部で ABAP クラウド API (Released API・Business Object Interface) のみを使用した拡張。ABAP クラウドの RAP モデルで S/4HANA のカスタムオブジェクトを追加します。S/4HANA 標準コードを変更せず、Released API の範囲内に留まるため、アップグレード影響を受けません。ただし「Released API として公開されていない業務ロジックへのアクセスは不可」という制約があります。
- Side-by-Side Extension (サイドバイサイド拡張) : BTP 上のアプリ (CAP アプリ・Build Apps アプリ等) が S/4HANA の Public OData/RFC の API を呼び出して拡張する方式。S/4HANA とは完全に独立したコードとして管理され、S/4HANA のアップグレード影響を受けません。複雑な UI・外部システム統合・AI 機能の追加に適しています。最もクリーンコア原則に沿った拡張方式です。
- Developer Extension (ISV/パートナー拡張) : SAP のパートナー・ISV が BTP 上で SaaS 拡張製品を構築して SAP App Center で提供する形式。顧客は標準を維持しながら、パートナーの付加価値機能をサブスクリプションで追加します。

Released API と Unreleased API の管理

クリーンコア準拠の実装において最大の課題は「S/4HANA の Released API (公式サポート API リスト)」の範囲確認です。

SAP API Business Hub (api.sap.com) で S/4HANA の Released OData API・BAPI・CDS View のリストを確認できます。しかし「業務要件に必要な機能が Released API に含まれていない」ケースは実際の導入で頻繁に発生します。この場合の対処選択肢は「① S/4HANA の Extensibility 機能 (Key User Extensibility) で標準機能を設定拡張して API 化する」「② SAP Support Notes に解決策がないか確認する」「③クリーンコア原則を一時的に妥協して In-App 拡張でカバーする (将来リファクタリングを計画)」です。

SAP は Released API の範囲を毎年拡張しており、過去に対応できなかった業務要件が API として公開されるケースも増えています。クリーンコア準拠度は「パーフェクトなゼロ改造」を目指すのではなく、「改造のない S/4HANA コア (改造スコアの定量的な管理)」と「BTP 側での機能代替計画」を組み合わせた現実的な目標設定が重要です。

8. SAP AI Services と GenAI Hub

BTP 上の AI サービス概要

SAP は BTP 上で AI 関連サービスを「SAP AI Core・SAP AI Launchpad・SAP Generative AI Hub・個別 AI サービス（Document Information Extraction・Business Entity Recognition 等）」として提供しています。BTP は SAP アプリケーション全体の AI インフラとして機能します。

SAP AI Core: 機械学習基盤の詳細

SAP AI Core は機械学習モデルのトレーニング・デプロイ・推論を実行する MLOps プラットフォームです。BTP 上の Kyma (Kubernetes) 基盤上で動作します。

- ワークフロー (Workflow) : ML パイプライン (データ前処理→モデルトレーニング→モデル評価→モデル登録) を Kubernetes ネイティブの「Argo Workflows」で定義します。GPU 対応の Kubernetes ノードでのトレーニングが可能です。
- デプロイメント (Deployment) : 学習済みモデルを REST API エンドポイントとして公開します。モデルの A/B テスト・カナリアリリース・スケーリング (リクエスト量に応じた自動スケール) が管理できます。
- モデルレジストリ: 学習済みモデルのバージョン管理・メタデータ (精度指標・学習データセット・ハイパーパラメータ) を記録します。SAP AI Launchpad の UI からモデルの評価比較・プロモーション (dev→prod 昇格) が操作できます。
- 制約: SAP AI Core は Kubernetes 基盤のため、MLOps 経験・Kubernetes 知識が必要です。GPU 付きインスタンスの利用は別途費用とクォータ申請が必要。日本リージョン (jp10) での GPU ノード利用可能性は事前確認が必要です。

SAP Generative AI Hub: マルチ LLM プラットフォーム

SAP Generative AI Hub は複数の LLM（大規模言語モデル）プロバイダーへのアクセスを一元管理するサービスです。プロバイダーとして OpenAI（GPT-4o 等）・Anthropic（Claude 等）・Google（Gemini 等）・Meta（Llama 等）・MistralAI 等をサポートしており、単一の API（GenAI Hub API）でモデルを切り替えて利用できます。

- LLM アクセスのガバナンス: GenAI Hub は単なる LLM プロキシではなく、「モデルごとのアクセス制御・コスト追跡（モデル別トークン消費量）・プロンプトフィルタリング（有害コンテンツ除去）・コンテンツログ（監査）」を一元管理します。SAP のデータプライバシーポリシーに基づき、入力データが LLM プロバイダーのトレーニングに使用されないよう制御されています。
- Orchestration Service: GenAI Hub 内の「Orchestration Service」は RAG（Retrieval-Augmented Generation）・プロンプトテンプレート・Chain-of-Thought 等の LLM アプリケーション構築パターンを Orchestration フレームワークとして提供します。BTP アプリから Orchestration API を呼び出すことで RAG システムを短時間で構築できます。
- SAP Joule との関係: SAP Joule は GenAI Hub の LLM を使用して動作します。Joule が提供する NL（自然言語）インターフェースのバックエンドは GenAI Hub です。企業がカスタム AI アシスタントを構築する場合も GenAI Hub API を基盤として利用します。

個別 AI サービスの詳細

BTP が提供する個別 AI サービス（AI Services）は特定の業務タスクに特化した学習済みモデルを API として提供します。

- Document Information Extraction（DIE）: 請求書・注文書・納品書等の非構造化ドキュメント（PDF・画像）からキーフィールド（発行日・金額・仕入先名・品目等）を自動抽出するサービス。OCR+NLP の組み合わせで動作します。SAP Cash Application の入金消込自動化で活用されています。
- Business Entity Recognition（BER）: テキスト（メール・契約書・報告書）からビジネスエンティティ（会社名・製品名・取引先担当者名等）を認識・抽出します。
- Translation Hub: SAP 公式の機械翻訳サービス。SAP 製品 UI の多言語化・ドキュメント翻訳に特化した翻訳品質を提供します。

- Intelligent Situation Automation (ISA) : S/4HANA の業務例外（在庫不足・支払遅延等の「Situation」）を検知して自動対処アクションをトリガーする AI 機能。Situation の検知→影響分析→推奨アクション（担当者通知・自動転記等）を自動実行します。

9. BTP のリージョン・可用性・SLA

グローバルリージョンとデータレジデンシー

SAP BTP はグローバルで 35 以上のリージョン（データセンター地域）で提供されています。リージョンは基盤となるハイパースケラー（Infrastructure Provider）によって以下に分類されます。

- SAP 独自データセンター（CF 専用）： SAP 自社運用のデータセンター（EU10/eu1： Frankfurt 等）。最も長い提供歴史を持ちますが、新機能の提供が他ハイパースケラーリージョンより遅延するケースがあります。
- Microsoft Azure リージョン（ap21 等）： Azure のグローバルインフラ上で稼働する BTP リージョン。Azure Native のサービスと BTP サービスの組み合わせが容易。
- Amazon Web Services（AWS リージョン）： AWS インフラ上で稼働する BTP リージョン（ap10： シドニー・us10： バージニア等）。
- Google Cloud Platform（GCP リージョン）： GCP インフラ上で稼働する BTP リージョン（us30 等）。
- 日本リージョン（jp10）： AWS の東京リージョンをインフラとする BTP リージョン。データ所在地が日本国内に限定されます。日本の個人情報保護法・金融規制・医療規制等のデータ所在要件に対応。

マルチリージョン・ディザスタリカバリ設計

BTP の可用性設計においてマルチリージョン構成とディザスタリカバリ（DR）の考え方を理解する必要があります。

BTP のプロダクション環境は原則として単一リージョン（単一サブアカウント）で稼働します。SAP BTP の SLA（Service Level Agreement）は「月次可用性 99.9%以上（一部サービスは 99.5%）」が標準です。ただし SLA はあくまで「SAP 起因の障害」に対する保証であり、ユーザーのアプリコードに起因する障害・計画メンテナンスは対象外です。

アクティブ-アクティブなマルチリージョン構成（複数リージョン間での負荷分散）は BTP 標準機能では提供されません。高可用性が要求されるシステムでは「アク

ティブ-パッシブ DR 構成（プライマリリージョン障害時に別リージョンで手動フェイルオーバー）」の設計が必要となり、アプリのデプロイ・データレプリケーション（HANA System Replication 等）の追加設計が必要です。

計画メンテナンスとアップグレード管理

SaaS 基盤である BTP は定期的な計画メンテナンス（Platform Upgrade）が実施されます。計画メンテナンスは SAP Cloud Availability Center（SAP for Me）で事前通知されます。

特に Cloud Foundry 環境のランタイムアップデート・HANA Cloud のメンテナンスウィンドウは、本番業務への影響を事前に評価するための管理プロセスが必要です。「変更管理プロセスに SAP のメンテナンス通知をインプットする」運用設計が重要です。

10. BTP 導入プロジェクトの設計・ガバナンス

BTP Center of Excellence (CoE) の設計

BTPはアプリ開発・統合・データ・AI・ローコードが混在するプラットフォームのため、「誰が何をどう使うか」というガバナンス設計が導入の成否を決定します。多くの企業でBTP CoE (Center of Excellence) を組織横断的なチームとして設置する事例が増えています。

- グローバルアカウント設計: グローバルアカウントの管理権限 (Global Account Administrator) は組織の中央IT部門が保持し、各事業部・プロジェクトにサブアカウントを払い出すモデルが標準です。事業部が独自にグローバルアカウントを作成すると、ライセンス・セキュリティ・コストの管理が分散して管理コストが増大します。
- コスト配賦モデル: BTPの消費ベース課金は「使った分だけコストが発生」するため、各サブアカウントのコスト配賦 (誰がどれだけ使ったか) の仕組みが必要です。BTPのCost Reporting APIを使ったコスト可視化・サブアカウント別予算割り当て・アラート設定を組み合わせたFinOps (クラウドコスト最適化) の実践が重要です。
- 開発・検証・本番の分離: BTPプロジェクトでは開発 (Dev) ・品質保証 (QA) ・本番 (Prod) ごとに独立したサブアカウントを作成し、デプロイパイプライン (CI/CD) による自動プロモーションを構築します。手動でのBTP設定変更・手動デプロイは本番環境では禁止し、全変更をCI/CDパイプライン経由でトレースできる体制が推奨されます。

BTP CLI と Infrastructure as Code

BTP Cockpit (WebUI) による手動設定は可視性・再現性・変更管理の面で限界があります。BTPのCI/CD・IaC化に利用できるツールを示します。

- BTP CLI (btpcli) : コマンドラインからグローバルアカウント・サブアカウント・エンタイトルメント・サービスインスタンスを管理するCLIツール。スクリプト化によるサブアカウントの自動作成・サービス有効化の自動化が可能です。

- Terraform Provider for SAP BTP: HashiCorp Terraform で BTP リソース (サブアカウント・エンタイトルメント・サービスインスタンス・Destination サービスの Destination 等) をコードとして管理するプロバイダー。IaC 化により BTP インフラの変更管理・バージョン管理・再現性が向上します。
- SAP Cloud MTA Build Tool (MBT) : Cloud Foundry へのデプロイに使用する MTA (Multi-Target Application) ビルドツール。Java・Node.js アプリ・iFlow 等を一つの MTA アーカイブ (.mtar) にパッケージ化してデプロイします。
- GitHub Actions / Azure DevOps 統合: BTP の CI/CD は SAP 標準の Piper (SAP pipeline framework) または GitHub Actions・Azure DevOps 等の外部 CI/CD との統合で実現します。コード Push→自動ビルド→自動テスト→BTP への自動デプロイというパイプラインが標準的な構成です。

スキル要件: BTP 導入に必要な技術スキル

BTP プロジェクトに必要なスキルセットは多岐にわたります。「一人の SAP エンジニアが全部できる」という前提は危険で、役割別のスキルマトリクスを明確にすることが重要です。

- BTP アーキテクト: アカウント設計・セキュリティ設計・統合アーキテクチャ・コスト見積もり。SAP 認定資格「SAP Certified Associate - SAP BTP Solution Architect」が参照基準。
- Cloud Foundry/CAP アプリ開発者: Node.js・Java・CAP フレームワーク・OData API・HANA Cloud。
- ABAP クラウド開発者: ABAP クラウド・RAP モデル・CDS View・Released API。S/4HANA の ABAP 経験があっても、ABAP クラウドへの学習転換が必要 (使用可能な API・フレームワークが大きく異なるため)。
- 統合開発者 (iFlow) : Integration Suite・iFlow 設計・アダプタ設定・Groovy Script・XML マッピング。EDI・メッセージング経験があると親和性が高い。
- インフラ/DevOps エンジニア: Kubernetes (Kyma)・Docker・Terraform・CI/CD パイプライン設計。

11. 他社導入事例

事例 1: グローバル製造業 A 社 — S/4HANA と Ariba ・ SuccessFactors のフルスイート統合

背景と課題

グローバルに展開する製造業 A 社は、SAP S/4HANA ・ Ariba ・ SuccessFactors ・ Concur をそれぞれ独立した SaaS として導入していました。しかし各製品間のデータ連携が「ポイントツーポイント統合」の積み重ねとなっており、統合の管理・障害時のトレーサビリティ・統合の変更コントロールが困難な状況でした。また、S/4HANA への改造（ABAP 修正）が蓄積しており、年次アップグレードのたびに改造コードの修正対応に数百万円規模のコストが発生していました。

BTP 導入と成果

A 社は BTP の Integration Suite（Cloud Integration）を統合ハブとして採用し、すべての SAP 製品間の統合を Cloud Integration 上の iFlow に集約しました。SAP 標準の統合シナリオ（Integration Packages）を最大限活用することで、iFlow 開発工数を従来のポイントツーポイント統合比で大幅に削減しました。また、S/4HANA の改造コードを BTP 上の CAP/Side-by-Side 拡張に段階的に移行するリファクタリングプロジェクトを実施し、S/4HANA のアップグレードを阻害する改造スコアを改善しました。Integration Suite のモニタリングにより統合エラーの早期発見・対処が可能となり、統合障害対応時間が大幅に短縮されたと報告されています。

事例 2: 小売業 B 社 — Build Process Automation による受注処理自動化

背景と課題

消費財を扱う小売業 B 社は、小規模取引先からの受注が FAX ・ メール ・ 電話で届き、受注担当者が SAP GUI に手入力するというアナログプロセスが残っていました。繁忙期には受注入力工数が激増し、入力ミスによる誤発注が頻発していました。受注入力自動化のための EDI 導入を検討しましたが、小規模取引先側の EDI 対応工数 ・ コストが障壁となって推進できませんでした。

BTP (SBPA + DIE) 導入と成果

B社はBTPのDocument Information Extraction (DIE)で受注メールの添付PDF(注文書)からキー情報(取引先名・品目コード・数量・納期)を自動抽出するAI処理フローを構築しました。抽出データはBuild Process Automationのワークフローで「低リスク受注(既存取引先・正常金額範囲内)は自動承認→SAP GUIへの自動入力(Desktop Automation RPA)→受注確認メール自動送信」するフローを実現しました。「高リスク受注(新規取引先・大口受注)」のみ担当者の確認フローを経由します。受注担当者の手入力工数が大幅に削減されると共に、入力ミスによる誤発注が減少したと報告されています。

事例3: 金融サービス業C社 — Kymaによるリアルタイムリスクモニタリング

背景と課題

証券・資産運用を主事業とする金融サービス業C社は、リスクモニタリングに使用しているシステムがS/4HANAと外部マーケットデータプロバイダーの二系統に分かれており、統合されたリアルタイムリスク可視化ができていませんでした。S/4HANAのカスタム改造で統合することも検討しましたが、監査・コンプライアンス上の要件からS/4HANA標準コードの維持が必要でした。

BTP (Kyma + Event Mesh + HANA Cloud) 導入と成果

C社はS/4HANAのビジネスイベント(Business Events)をEvent Mesh経由でKyma上のリスク集計マイクロサービスにリアルタイム配信するアーキテクチャを構築しました。Kymaアプリは外部マーケットデータプロバイダーのストリームAPIからのリアルタイムデータと、S/4HANAのイベントデータを組み合わせてHANA Cloudにリスクサマリーを書き込みます。SAC(SAP Analytics Cloud)のLive Connection経由でHANA Cloudのリスクサマリーをリアルタイム可視化するダッシュボードをデジタルボードルームとして整備しました。S/4HANAの標準コードを一切変更せずに外部データとの統合リアルタイム分析を実現し、監査上の課題が解消されたと報告されています。

12. 制約事項と導入リスクの総括

BTPの「できること」と「できないこと」

BTPは広大な機能領域をカバーしますが、特定の場面では制約や代替手段の設計が必要です。

- ◎できる：SAP製品群のAPIベース統合（Cloud Integration）・カスタムアプリ開発（CAP/Build Apps）・ワークフロー自動化（SBPA）・APIゲートウェイ（API Management）・クラウドDB（HANA Cloud）・LLM活用（GenAI Hub）
- △条件次第：オンプレミスシステムとのリアルタイム同期（Cloud Connectorが必要・レイテンシ課題あり）・複雑なオフラインモバイルアプリ（Build Appsで基本対応可だが高度な差分同期は困難）・ミッションクリティカルなリアルタイムERPトランザクション（BTPアプリは補完的役割、コアトランザクションはS/4HANA上で実行が基本）
- ×できない：オンプレミスS/4HANAのABAPコード直接実行（ABAP EnvironmentはABAPクラウドのみ）・Cloudに非対応のネットワーク経路（閉域網専用DBへの直接接続）・インターネット非接続環境でのBTPサービス利用（BTPはパブリッククラウド前提）

導入前に確認すべき重要事項

BTP導入を成功させるために、プロジェクト開始前に確認・設計すべき重要事項を示します。

- エンタイトルメントの確認：使用予定のサービス・機能がSAPとの契約に含まれているか。消費ベース課金のクレジット残量は十分か。サービスプラン（Free/Standard/Premium）による機能制限はないか。
- リージョン選択の確認：データ所在地要件に対応したリージョンが利用可能か。日本リージョン（jp10）で必要なサービスが提供されているか（一部新サービスはまず欧米リージョン先行でリリースされる）。
- スキル調達計画：プロジェクト開始時点でのチームのBTPスキル評価と、必要なトレーニング・採用・外部支援の計画。BTPは「SAP経験者が学習すれ

ば即戦力」とはならず、Cloud Native 開発・統合・MLOps 等の追加スキルが必要な領域が多い。

- ガバナンス先行設計：アカウント階層・コスト管理・セキュリティポリシー・開発プロセス（CI/CD）をプロジェクトの Day 1 から設計・運用することが、BTP 投資の長期的な健全性を維持する最重要要件です。

以上