



SAP S/4HANA Cloud, Public edition ABAP拡張ブートキャンプ[®]

SAP ABAP RESTful Application Programming Modelおよび 開発者拡張でのキーとなる技術要素

後藤健司

SAP BTP App. Dev & Integration, Customer Advisory
SAP Japan 株式会社

INTERNAL – SAP and Partners Only

THE BEST RUN **SAP**

*本資料に関する使用条件、免責条項、開示、または制限事項については、法律上の注意点 (www.sap.com/legal-notice) を参照してください。

本ワークショップのスケジュール

1日目(7月4日)		2日目(7月5日)	
開催場所: SAPジャパン本社 11F 山吹 (オンサイト)			
13:00 – 17:00		10:00 – 17:00	
(1日目は午後から開始します。ご注意ください。)		10:00 – 12:30	演習2 S/4HANA Cloud, public edition 拡張 + RAPを用いた簡易オンラインショップアプリの作成
13:00 – 14:45	プレゼンテーション “SAP ABAP RESTful Application Programming Modelおよび 開発者拡張でのキーとなる技術要素”	12:30 – 14:00	昼食
14:45 – 15:00	休憩	14:00 – 16:30	演習3 Side-by-Side拡張で利用するためのカスタムAPIの作成
15:00 – 17:00	演習1 S/4HANA Cloud, public edition 拡張のシンプルなABAP クラスへの実装	16:30 – 17:00	理解度テスト

Agenda

- “クラウドとオンプレにおけるSAP S/4HANAのためのクラウド拡張オプションの”超”概要”
(2023年6月8日開催のパートナー様向けWebiner セッション)の振り返り
- SAP RESTful Application Programming Model (RAP)について
- RAP ファーサード / RAP BO Interface を用いるための EML (Entity Manipulation Language)
- RAP での Behavior Definition および Behavior Implementation
- 追加トピック: S/4HANA PCE / オンプレミス環境でのATC in Eclipseについて
- ご参考情報

Agenda

- “クラウドとオンプレにおけるSAP S/4HANAのためのクラウド拡張オプションの“超”概要”
(2023年6月8日開催のパートナー様向けWebiner セッション)の振り返り
- SAP RESTful Application Programming Model (RAP)について
- RAP ファーサード / RAP BO Interface を用いるための EML (Entity Manipulation Language)
- RAP での Behavior Definition および Behavior Implementation
- 追加トピック: S/4HANA PCE / オンプレミス環境でのATC in Eclipseについて
- ご参考情報

新しいクラウド拡張オプションの基本的な考え方

SAP Cloud Solution における前提

パブリッククラウドでは、SAP ソフトウェアの更新が自動化され、すべてのテナントにおいて一斉に実行されます。

SAPがご提供する手順に基づいたアップグレードプロセスに準拠。

クラウド拡張オプション共通ルール

その1

拡張では、リリース済 SAP API のみを使用可能。SAP では、これらの API の安定性を維持します。

その2

SAP オブジェクトは、事前定義された拡張ポイントを介してのみ拡張可能。フリースタイル変更は許可されません。

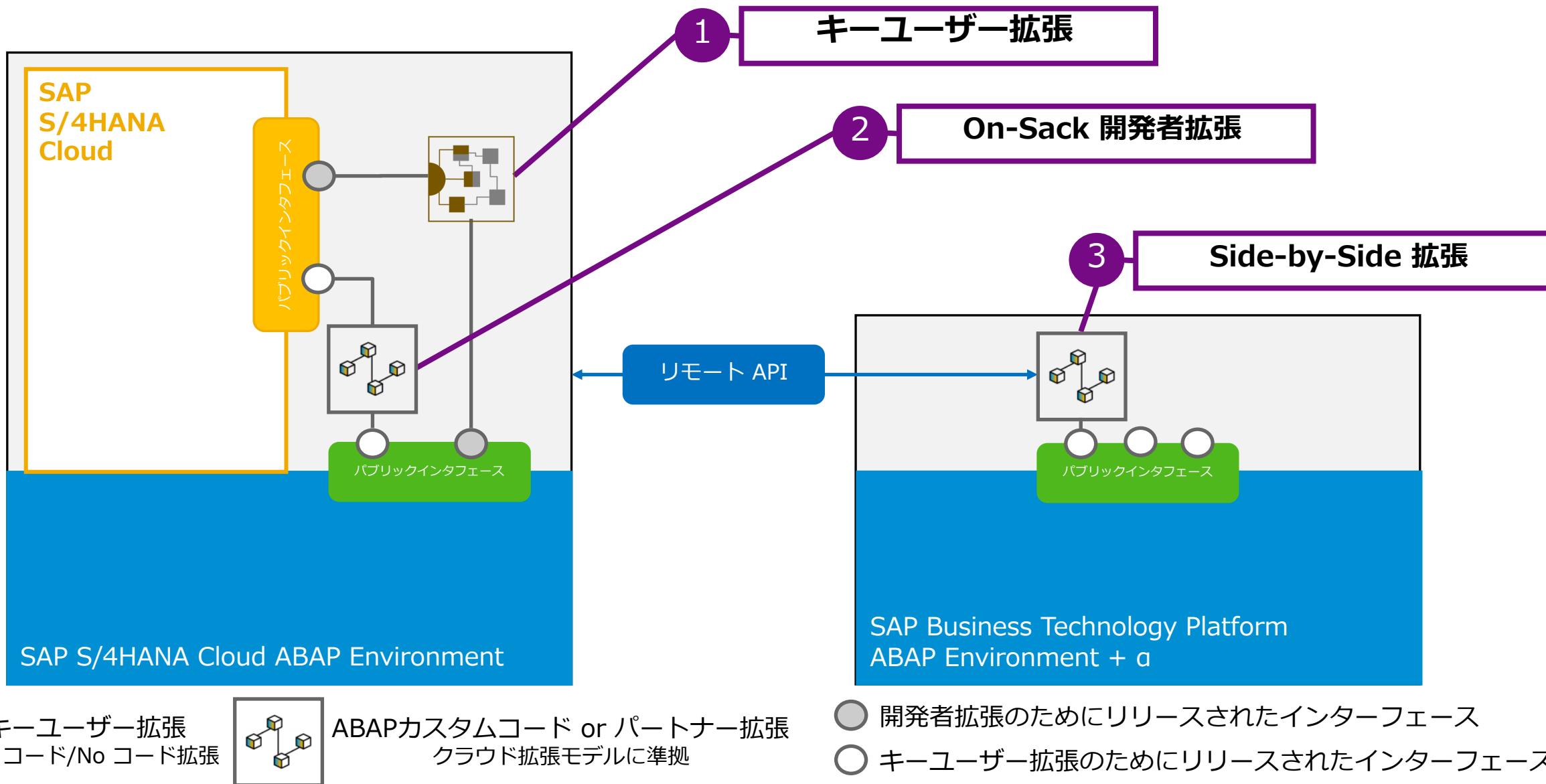
その3

キーユーザおよび拡張開発者は、クラウド対応およびリリース済のテクノロジーのみを使用。開発者およびキーユーザのアクティビティは、アクセス権限によって規制されます。

S/4HANA の各エディション毎の新しい拡張オプションの適応アプローチ (SAP推奨)

	新しいクラウド拡張オプションへの対応方法
SAP S/4HANA Cloud public Edition	必須 <ul style="list-style-type: none">□ 旧来型のクラシックなABAP拡張はサポートなし
SAP S/4HANA Cloud private Editionおよびオンプレミス グリーンフィールド導入	優れたクラウド拡張オプションをベースとして導入開始 <ul style="list-style-type: none">□ スムーズなSAPソフトウェアのアップグレード□ 来るべきクラウド移行のための将来的に安全な拡張
SAP S/4HANA Cloud private Editionおよびオンプレミス クラシック拡張を変更する導入	新しいクラウド拡張オプションを取り入れつつ、適応が難しい部分のみ従来の拡張方法を利用した導入 <ul style="list-style-type: none">□ 開発者は新しいクラウド拡張オプション習得し、少しづつクラシック拡張をクラウドに対応した拡張に移行する

新しいSAP S/4HANA Cloud 拡張オプションの全体像



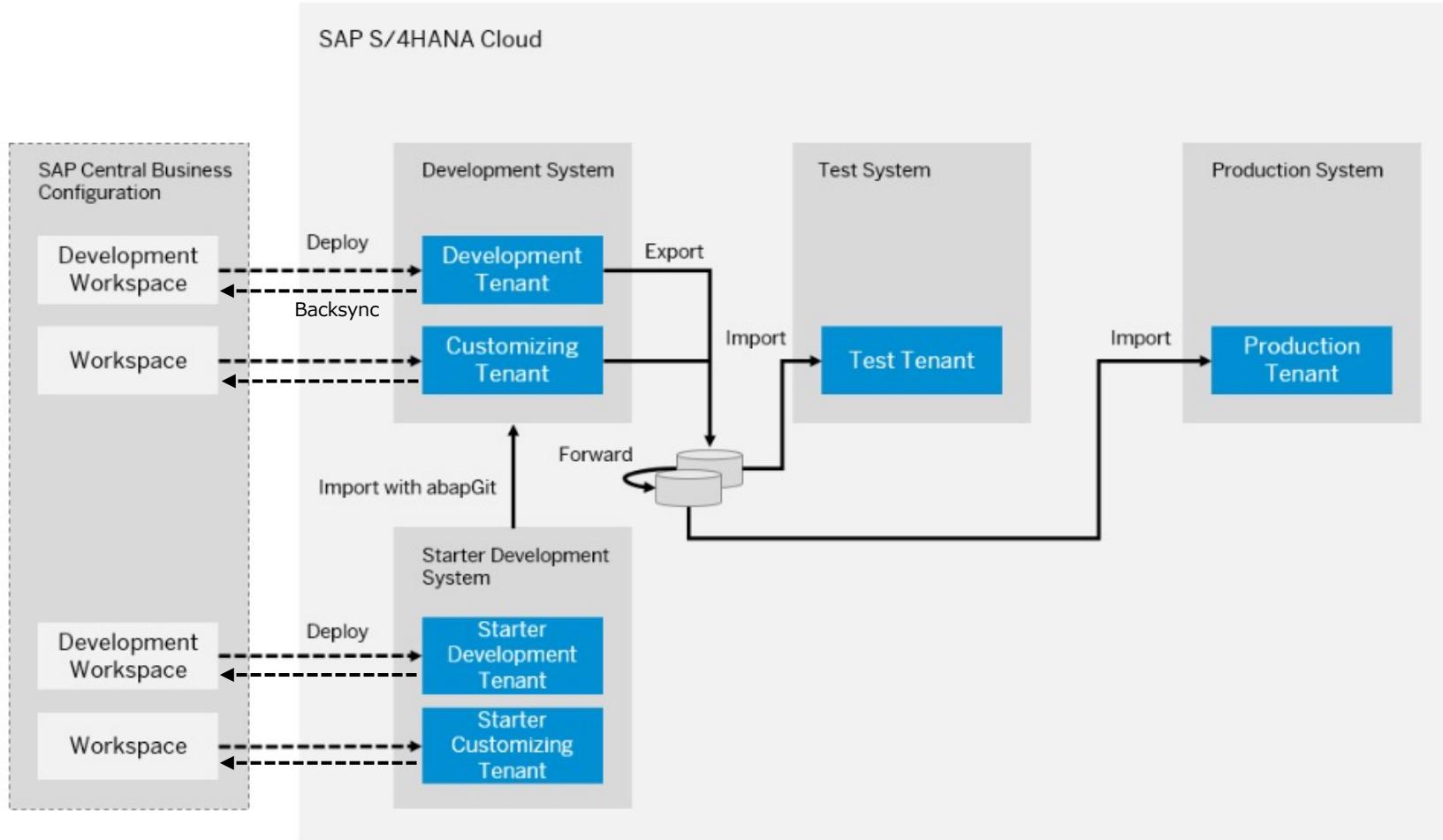
SAP S/4HANA Public Cloud における 拡張性オプションの概要

	キーユーザ拡張	(On-Stack)開発者拡張	Side-by-side拡張性
シナリオ	ビジネスエキスパート、導入コンサルタント、シチズンデベロッパー、キーユーザー	ABAP 開発者	ABAP 開発者 + a
ユースケース	UI の調整、ユーザ定義項目の追加、カスタムビジネスオブジェクトの追加など	SAP S/4HANA データへの頻繁または複雑な SQL アクセスを伴うカスタムアプリケーション SAP アプリケーションと同じ作業論理単位 (LUW) で実行されるカスタム拡張 サイドバイサイドの SAP BTP アプリを提供する、カスタマイズされたカスタムリモート API またはサービス	個別のターゲットグループ向けのカスタムアプリケーション (ERP ユーザなし) ERP から分離して実行されるカスタムアプリケーションワークフロー 機械学習や AI などのインテリジェントな SAP BTP サービスに近接する必要があるカスタムアプリケーション 複数の ERP システムおよびクラウドサービスと統合されるソリューション パートナーが提供する SaaS アプリケーション
利点	SAP S/4HANA Cloud で完全に管理および統合 開発スキルがないか、非常に基本的なスキルのみ必要	SAP S/4HANA Cloud システム内での拡張の開発 リモートアクセスまたはデータレプリケーションなし リリース済の SAP S/4HANA Cloud オブジェクトの使用および拡張	SAP S/4HANA Cloud の運用およびライフサイクル管理に依存しない分離された拡張

On-Stack拡張

Side-by-side拡張

S/4HANA Cloud 3SLの場合の移送プロセスと機能 システムおよびテナント



スタータシステム

- (On-Stack開発のための) 開発テナント
- (設定およびキーユーザ拡張のための) カスタマイジングテナント

開発システム

- (On-Stack開発のための) 開発テナント
- (設定およびキーユーザ拡張のための) カスタマイジングテナント

テストシステム

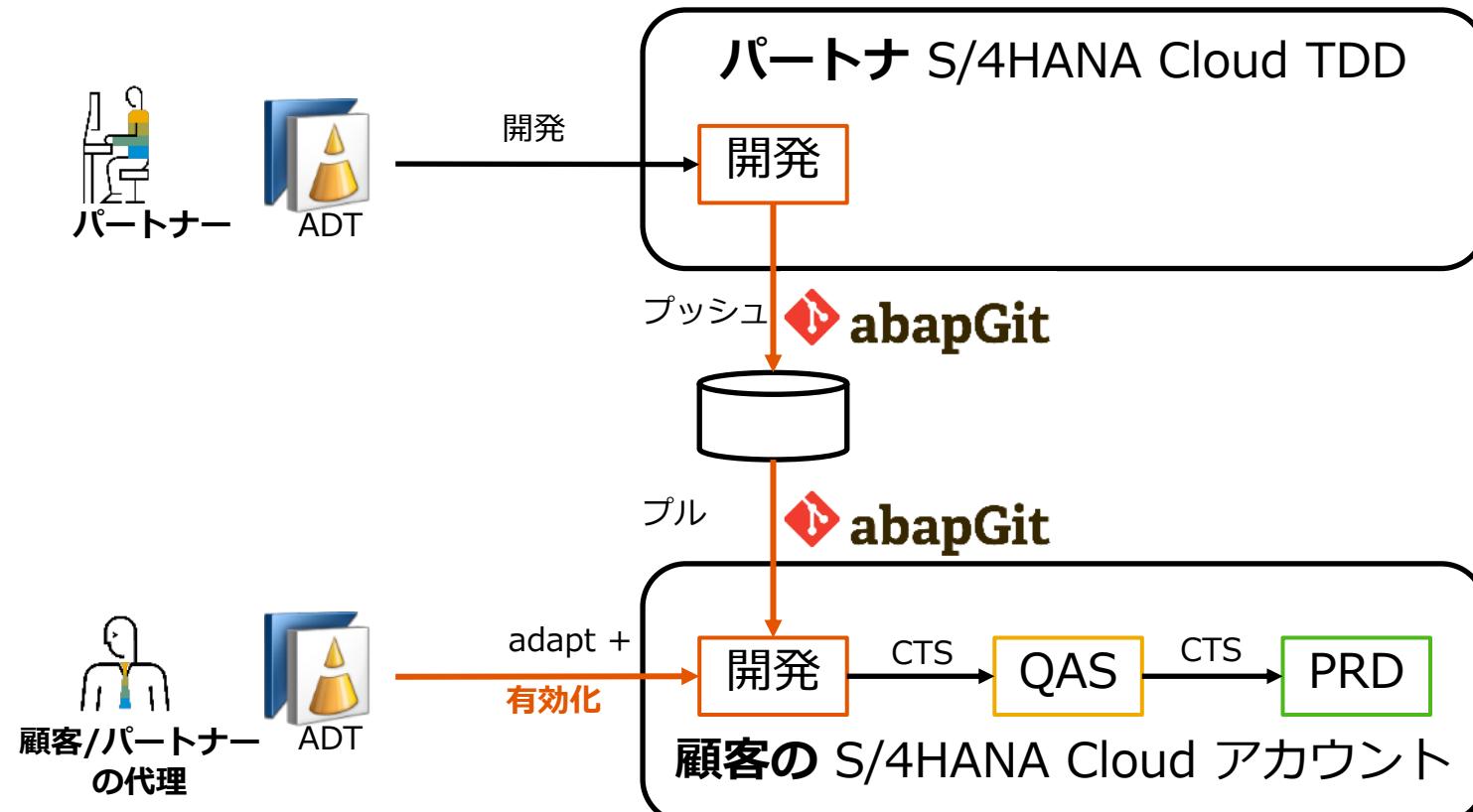
- テストテナント

本稼動システム

- 本稼動テナント

RISE with SAP S/4HANA Cloud

現在サポートされているライフサイクルシナリオ | abapGit によるエクスポート/インポート

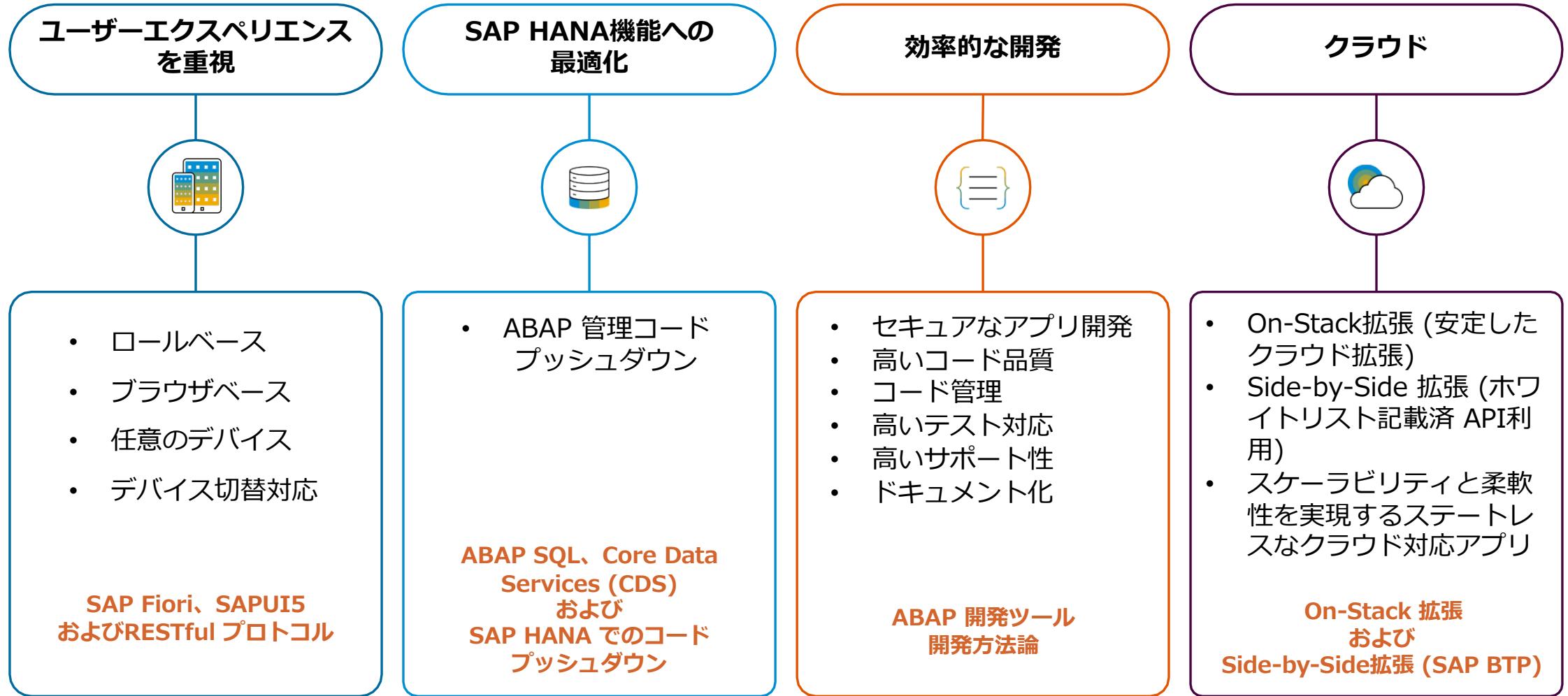


- 現在サポートされているユースケースは、"カスタマの代理での開発"です。
- カスタマ名称領域のみが許可されます**
- 開発は、Eclipse 向け ABAP 開発ツールで使用可能な **abapGit** インタフェースを使用して、(パートナ TDD などから) **エクスポート**し、カスタマランドスケープにインポートすることができます。
- ロードマップでは、追加のライフサイクルツールとシナリオが計画されています (2023 年以降)。

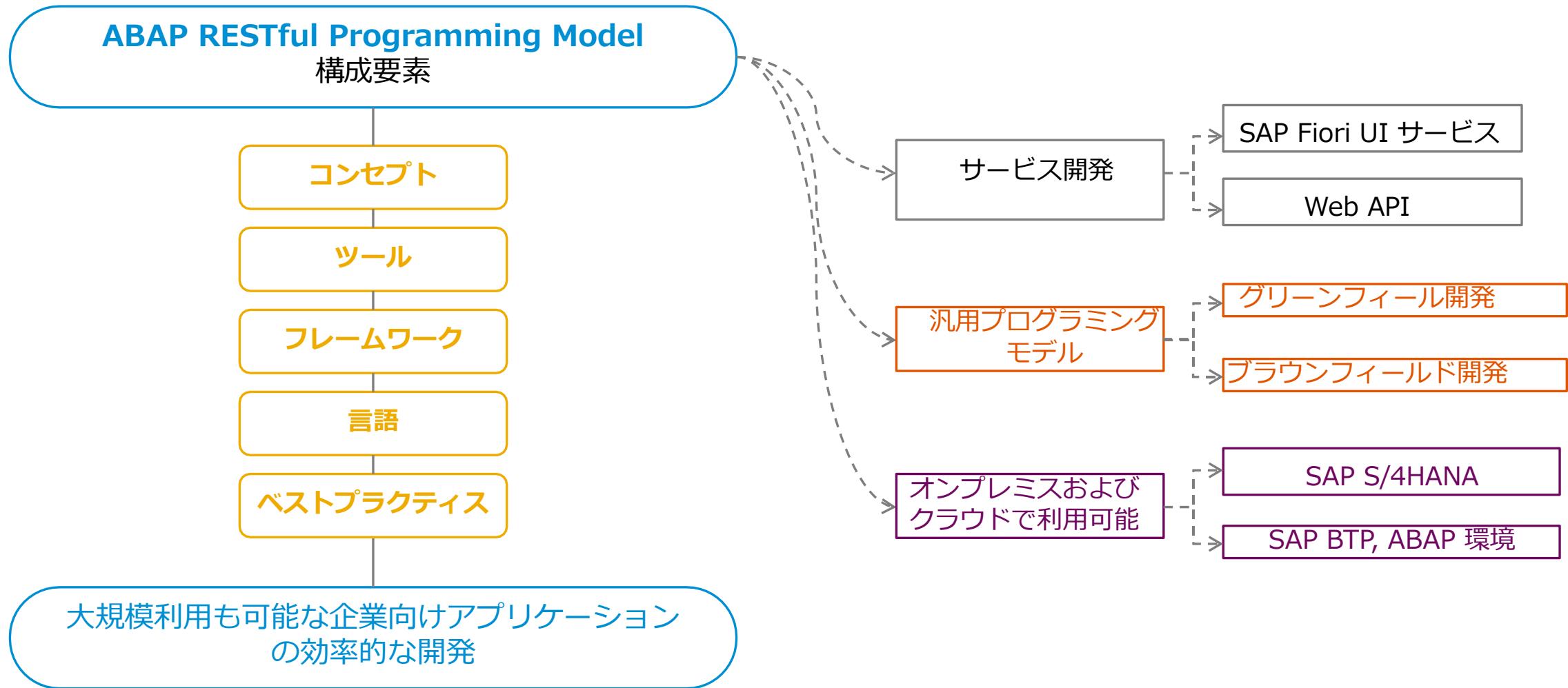
Agenda

- “クラウドとオンプレにおけるSAP S/4HANAのためのクラウド拡張オプションの”超”概要”
(2023年6月8日開催のパートナー様向けWebiner セッション)の振り返り
- **SAP RESTful Application Programming Model (RAP)について**
- RAP ファーサード / RAP BO Interface を用いるための EML (Entity Manipulation Language)
- RAP での Behavior Definition および Behavior Implementation
- 追加トピック: S/4HANA PCE / オンプレミス環境でのATC in Eclipseについて
- ご参考情報

最新の ABAP アプリケーション開発



ABAP RESTful Application Programming Model (RAP) とは?



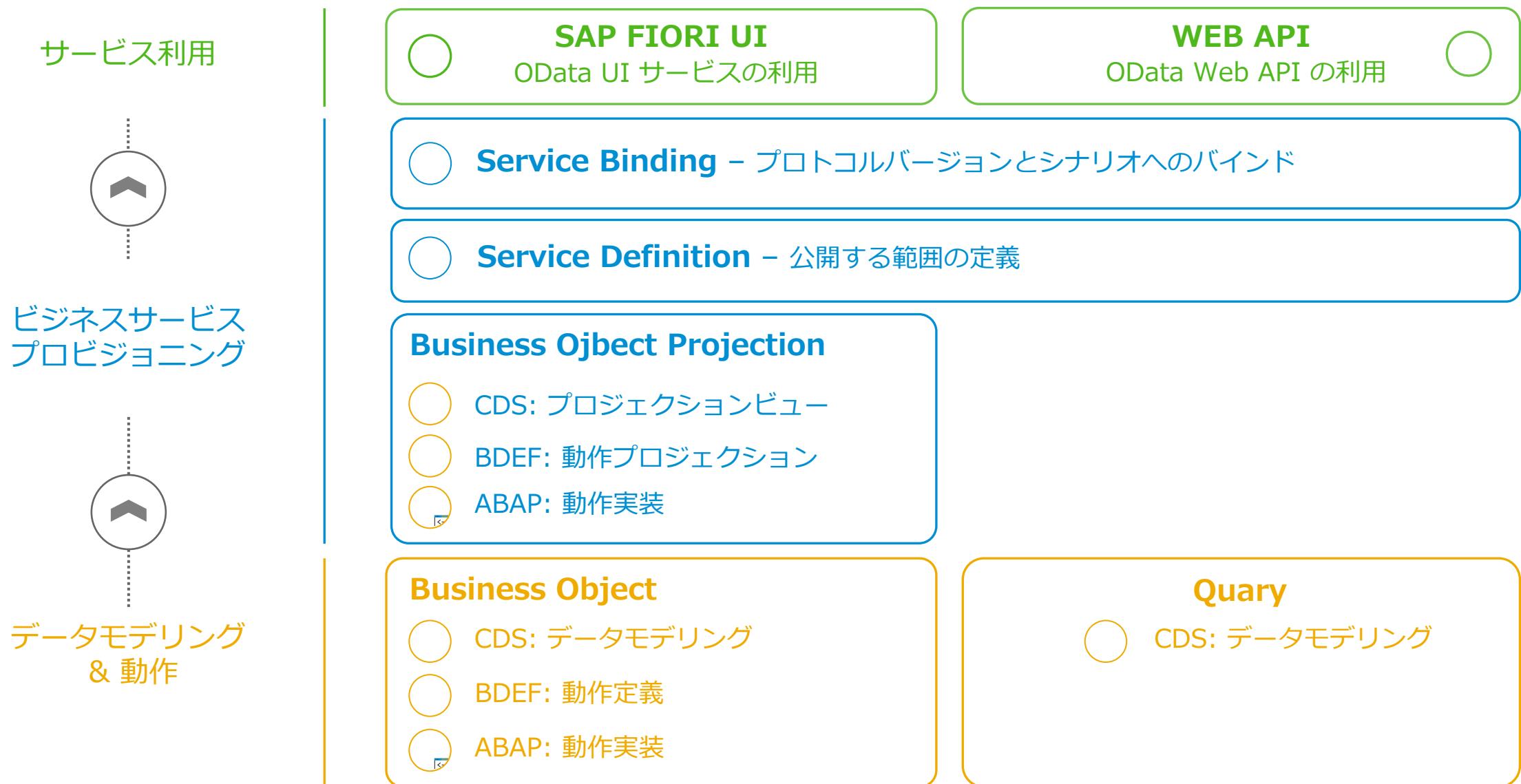
RAP のサポート環境

SAP S/4HANA Cloud, ABAP Environment
(code name: Embedded Steampunk)

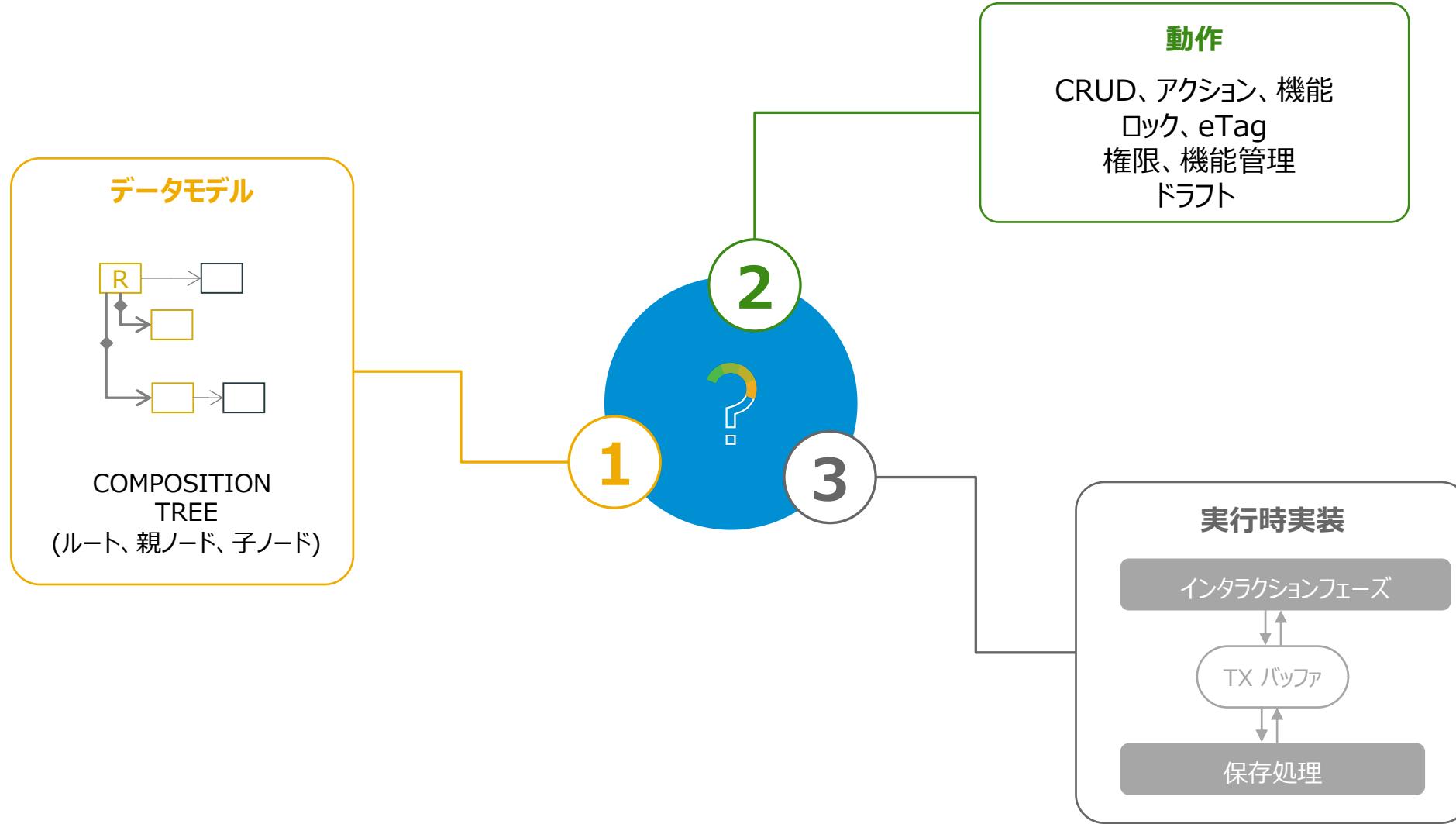
SAP S/4HANA, ABAP Environment (1909~*)

SAP BTP, ABAP Environment
(code name: Steampunk)

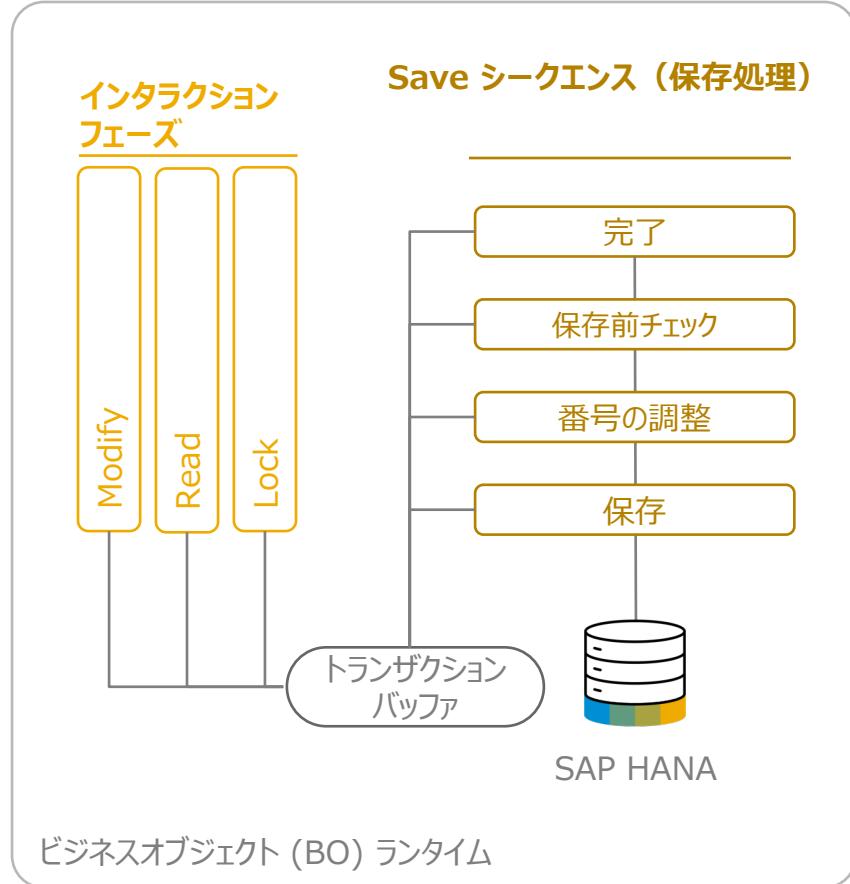
RAP – 全体像



ビジネスオブジェクトとは何ですか。



ビジネスオブジェクト - 実行時実装タイプ



UNMANAGED

アプリケーションコーディングを完全に使用できるブラウンフィールド開発：
インタラクションフェーズ + トランザクションバッファ + 保存順序

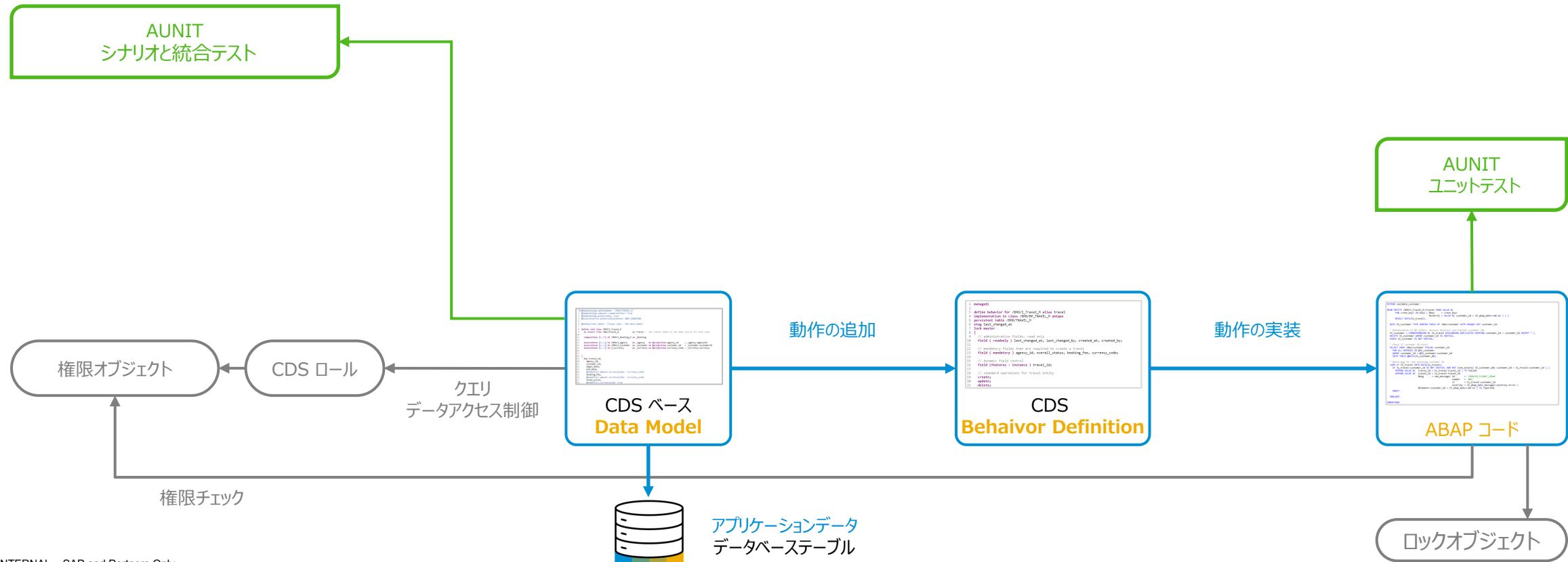
- BO ランタイムを担当する開発者: CRUD 操作
- 既存のコードの統合に必要なアダプタ

MANAGED

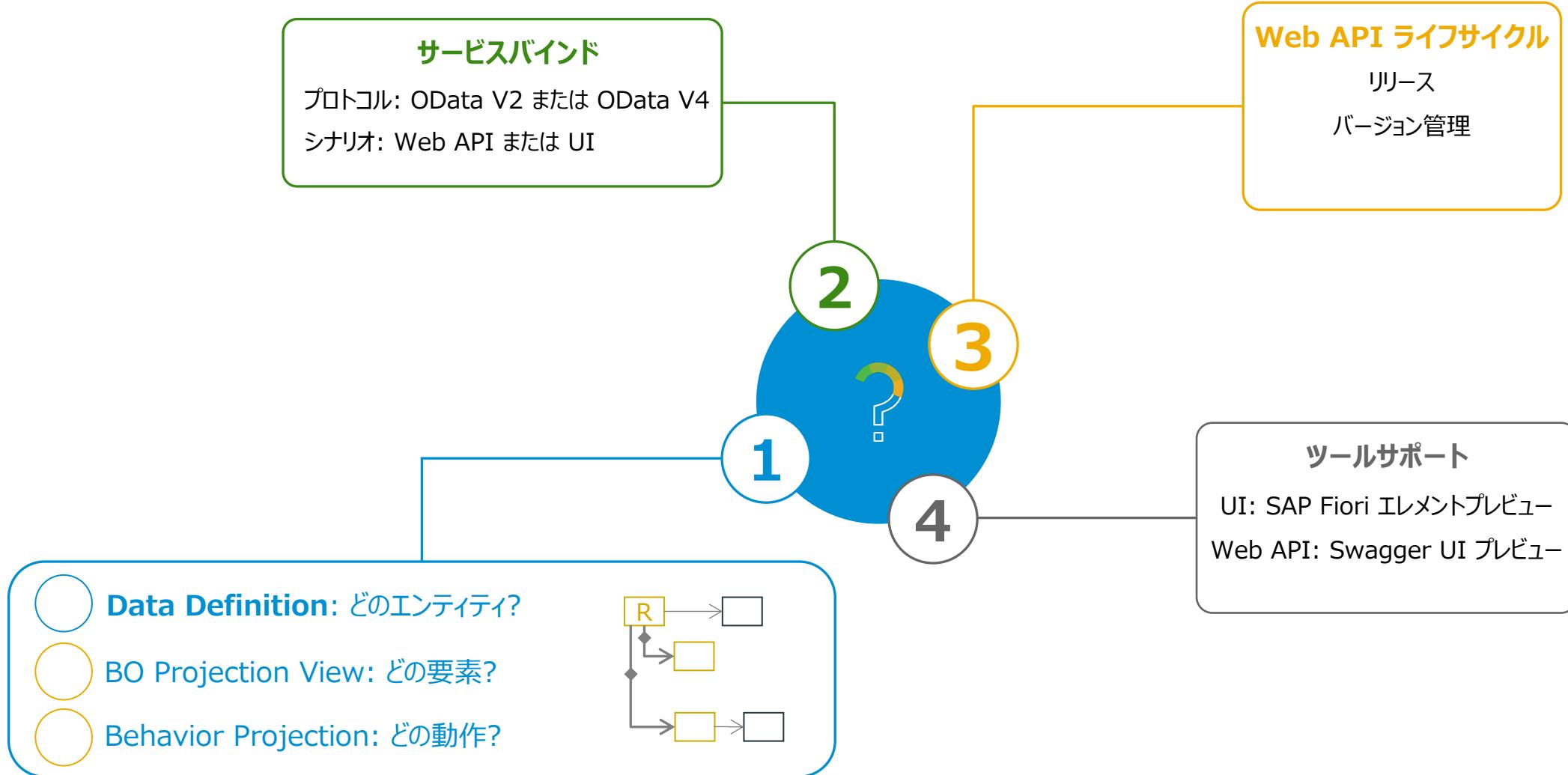
標準実装によるグリーンフィールド開発 (任意管理されていないアプリケーションコンポーネント: DB テーブル、ロック/PFCG オブジェクト、更新タスク FM)

- 即座に利用可能な標準 CRUD 操作
- 開発者による BO 固有ビジネスロジックの追加

開発フロー



ビジネスサービスとは

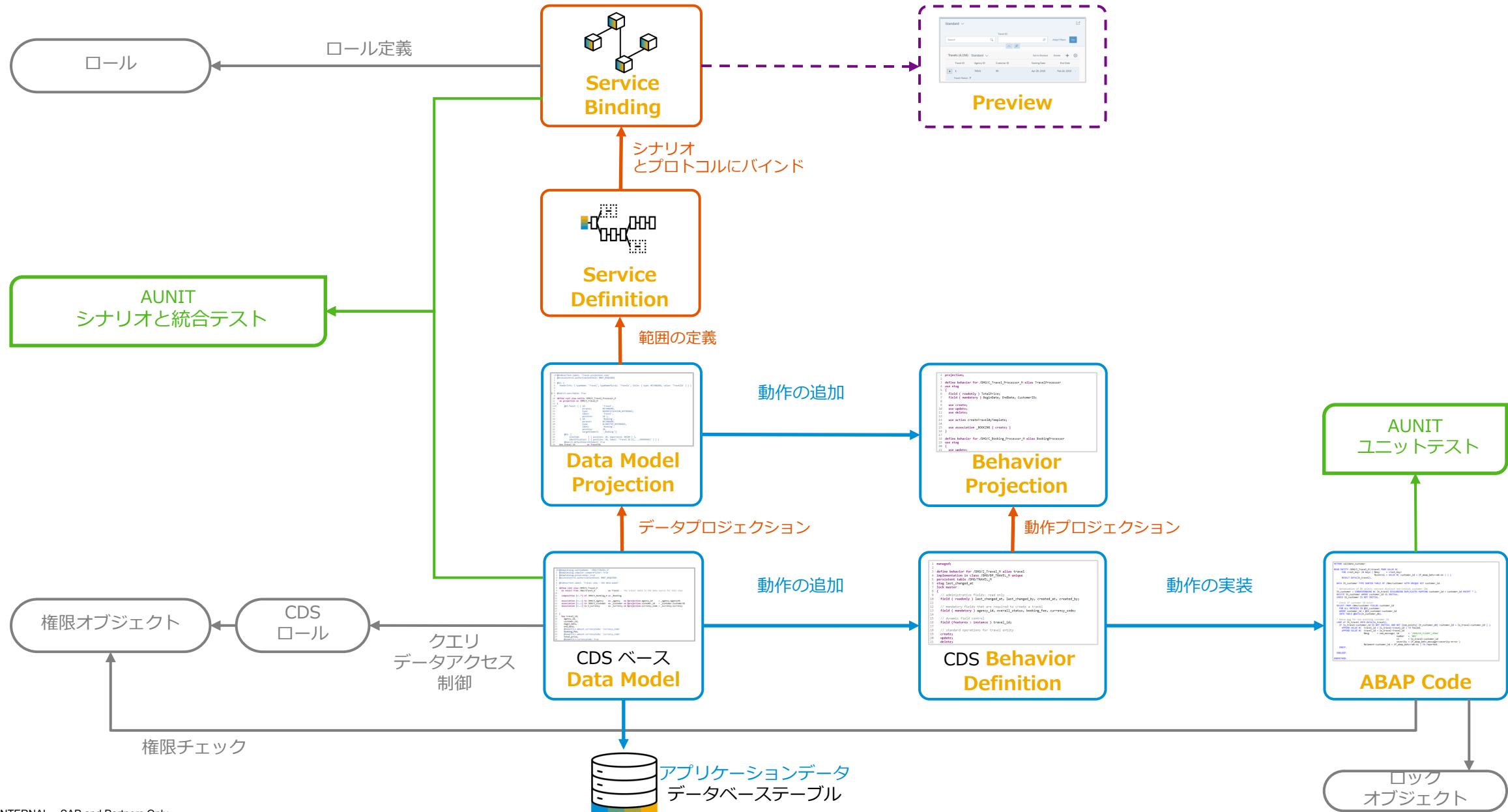


SAP Fiori UI と Web API のサービスで必要な要素

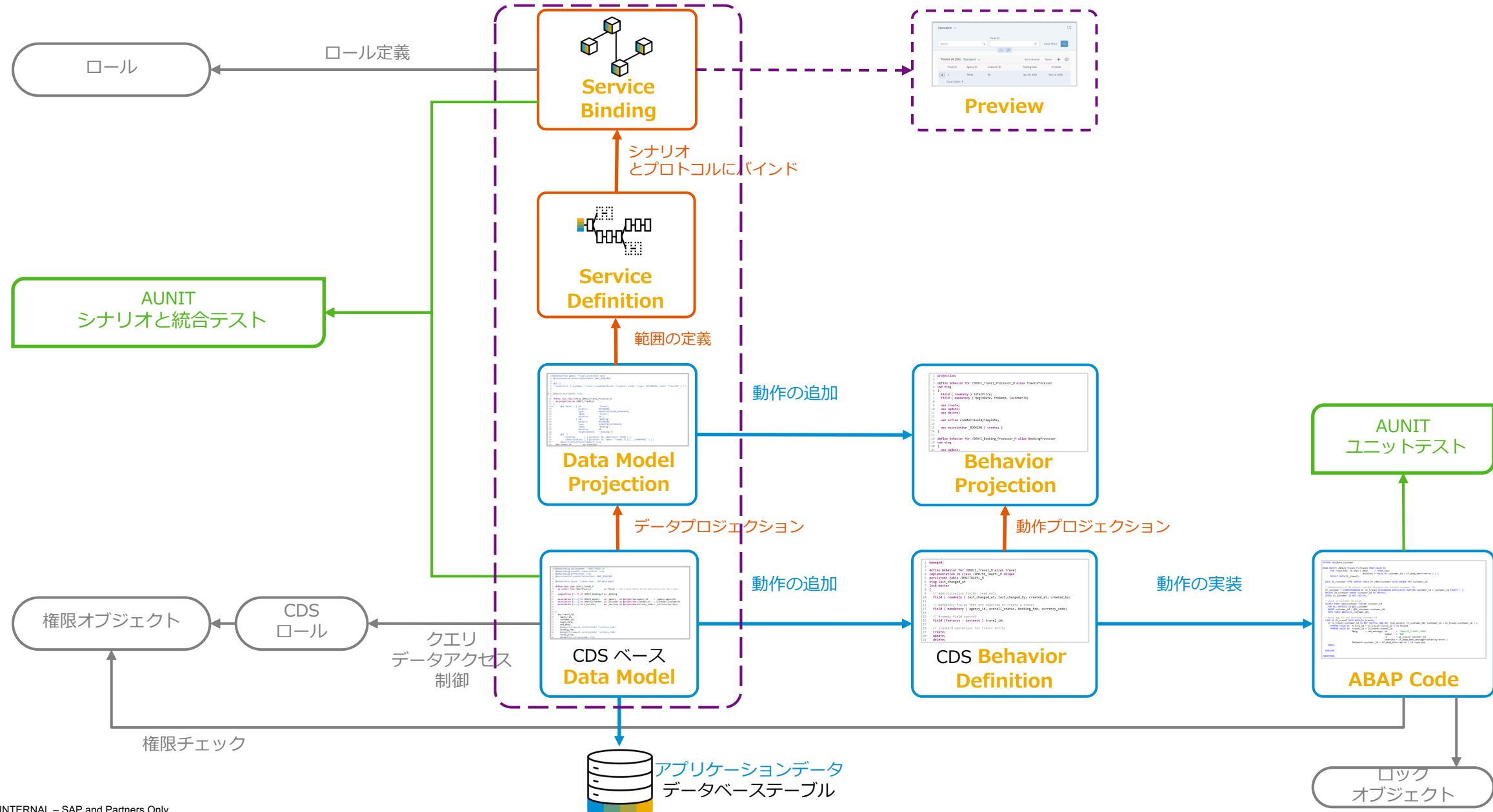


	SAP Fiori UI	Web API
UI アノテーション	必須	不要
入力ヘルプ	必須	不要
動的なfeature 制御	必須	不要
ドラフト	必須	不要
複合的な\$batch リクエストのためのコンテンツ ID	不要	必須
リリース、バージョン管理、および非推奨の定義	不要	必須

開発フロー



コード解説: 最単純なCDSアプリケーション(READ オペレーションのみの実装)



Data Model –サンプル

```
1 @EndUserText.label: 'Travel data XXX'  
2 @AccessControl.authorizationCheck: #CHECK  
3 define root view entity ZI_TRAVEL_M_510  
4   as select from ztravel_510 as Travel  
5  
6 /* Associations */  
7 association [0..1] to /DMO/I_Agency as _Agency on $projection.agency_id = _Agency.AgencyID  
8 association [0..1] to /DMO/I_Customer as _Customer on $projection.customer_id = _Customer.CustomerID  
9 association [0..1] to I_Currency as _Currency on $projection.currency_code = _Currency.Currency  
10  
11 {  
12  
13 key mykey,  
14   travel_id,  
15   agency_id,  
16   customer_id,  
17   begin_date,  
18   end_date,  
19   @Semantics.amount.currencyCode: 'currency_code'  
20   booking_tee,  
21   @Semantics.amount.currencyCode: 'currency_code'  
22   total_price,  
23   currency_code,  
24   overall_status,  
25   description,  
26  
27  
28-- Admin data --*  
29 @Semantics.user.createdBy: true  
30   created_by,  
31   @Semantics.systemDateTime.createdAt: true  
32   created_at,  
33   @Semantics.user.lastChangedBy: true  
34   last_changed_by,  
35   @Semantics.systemDateTime.lastChangedAt: true  
36   last_changed_at,  
37  
38 /* Public associations */  
39   _Agency,  
40   _Customer,  
41   _Currency  
42 }  
43
```

Rootビジネスオブジェクト"Travel"の定義
エンティティ"ztravel_510"からデータ選択

エンティティ"ztravel_510"と他のエンティティの連携を定義
[0..1] は、それぞれの連携にゼロまたは 1 つのエンティティを関連付けることができることを示します。

標準テーブルの"currency_code"を参照するアノテーション

データ作成したユーザーを保存するアノテーション

データ作成した日付を保存するアノテーション

Data Model Projection – サンプル

```
① @Search.searchable: true → エンティティ検索を可能にするアノテーション
② 9 define root view entity ZC_TRAVEL_M_510
10   as projection on ZI_TRAVEL_M_510
11 {
12   @UI.facet: [ { id: 'Travel',
13     purpose: '#STANDARD,
14     type: '#IDENTIFICATION_REFERENCE,
15     label: 'Travel',
16     position: 10 } ]
17
18   @UI.hidden: true
19   key mykey      as TravelUUID,
20
21
22   @UI: {
23     lineItem: [ { position: 10, importance: '#HIGH' },
24       identification: [ { position: 10, label: 'Travel ID [1,...,99999999]' } ] ]
25   @Search.defaultSearchElement: true
26   travel_id      as TravelID,
27
28   @UI: {
29     lineItem: [ { position: 20, importance: '#HIGH' },
30       identification: [ { position: 20 } ],
31       selectionField: [ { position: 20 } ] ]
32   @Consumption.valueHelpDefinition: [{ entity : {name: '/DMO/I_Agency', element: 'AgencyID' } }]
33
34   @ObjectModel.text.element: ['AgencyName'] ---meaning?
35   @Search.defaultSearchElement: true
36   agency_id      as AgencyID,
37   _Agency.Name    as AgencyName,
38
39   @UI: {
40     lineItem: [ { position: 30, importance: '#HIGH' },
41       identification: [ { position: 30 } ],
42       selectionField: [ { position: 30 } ] ]
43   @Consumption.valueHelpDefinition: [{ entity : {name: '/DMO/I_Customer', element: 'CustomerID' } }]
44   @ObjectModel.text.element: ['CustomerName']
45   @Search.defaultSearchElement: true
46   customer_id    as CustomerID,
47
48   @UI.hidden: true
49   _Customer.LastName as CustomerName,
50
51
52   @UI: {
53     lineItem: [ { position: 40, importance: '#MEDIUM' },
54       identification: [ { position: 40 } ] ]
55     begin_date    as BeginDate,
56
57   @UI: {
58     lineItem: [ { position: 41, importance: '#MEDIUM' },
59       identification: [ { position: 41 } ] ]
60     end_date     as EndDate,
61
62   @UI: {
63     lineItem: [ { position: 50, importance: '#MEDIUM' },
64       identification: [ { position: 50, label: 'Total Price' } ] ]
65   @Semantics.amount.currencyCode: 'CurrencyCode'
66   total_price    as TotalPrice,
67
68   @Consumption.valueHelpDefinition: [{entity: {name: 'I_Currency', element: 'Currency' }}]
69   currency_code  as CurrencyCode,
70
71   @UI: {
72     lineItem: [ { position: 60, importance: '#HIGH' },
73       { type: '#FOR_ACTION, dataAction: 'acceptTravel', label: 'Accept Travel' } ],
74     identification: [ { position: 60, label: 'Status [0(Open)|A(Accepted)|X(Canceled)]' } ] ]
75   overall_status  as TravelStatus,
```

Service definition / Service binding

1. Service Definition

```
1 @EndUserText.label: 'Service definition for travel '
2 define service ZUI_C_TRAVEL_M_510 {
3   expose ZC_TRAVEL_M_510 as TravelProcessor;
4   expose /DM0/I_Customer as Passenger;
5   expose /DM0/I_Agency as TravelAgency;
6   expose /DM0/I_Airport as Airport;
7   expose I_Currency as Currency;
8   expose I_Country as Country;
9 }
```

Data Projection "ZC_TRAVEL_M_510"
を"TravelProcessor"として公開

2. Service Binding

The screenshot shows the "Service Binding: ZUI_C_TRAVEL_M_510" configuration page. It has two main sections: "General Information" and "Service Versions".

General Information: Describes the service binding. A red box highlights the "Binding Type: OData V2 - UI" field.

Service Versions: Allows defining service versions. A red box highlights the "Version" column for the entry "1.0.0".

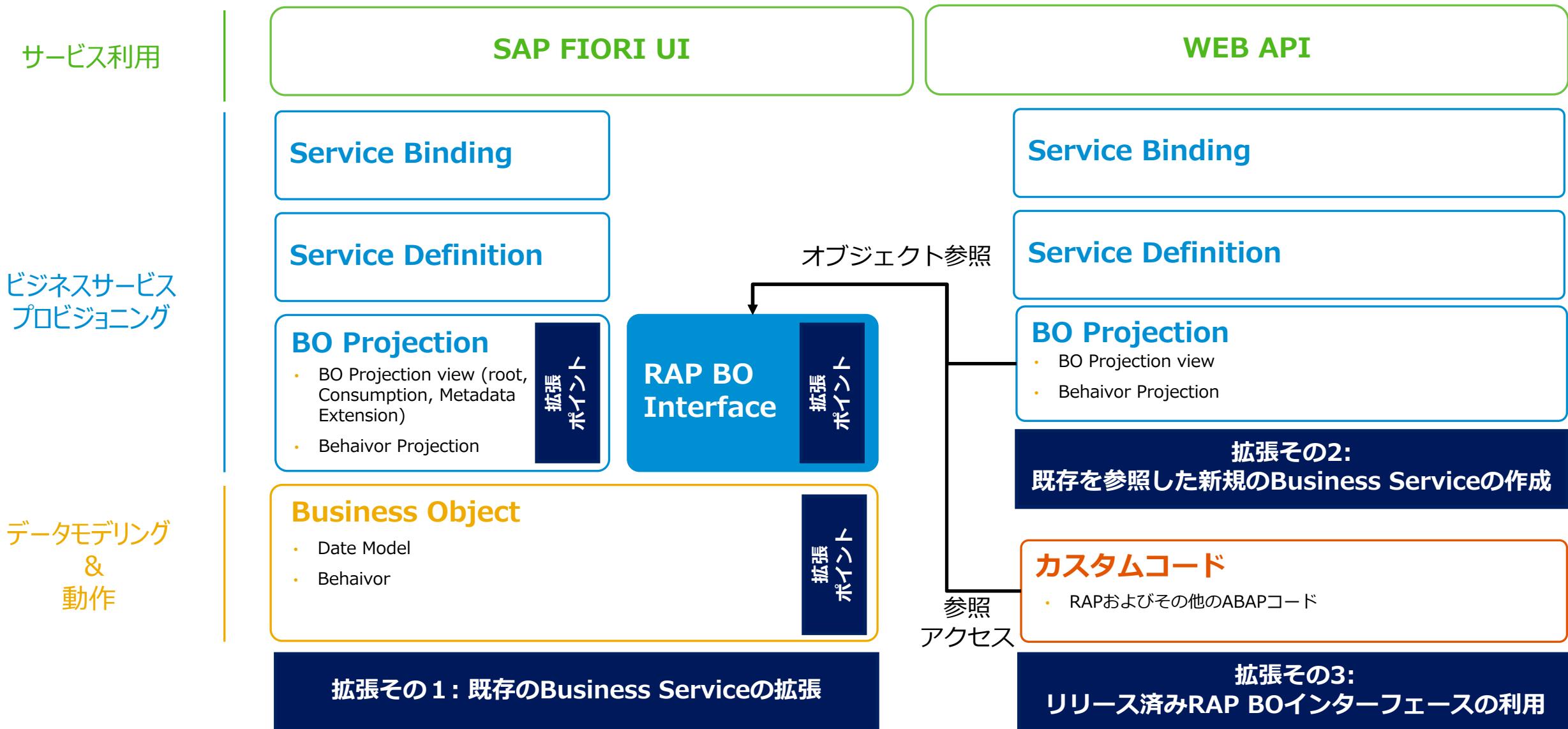
Version	API State	Service Definition	Add...	Remove
1.0.0	Not Released	ZUI_C_TRAVEL_M_510		

"TravelProcessor"をOdata V2として
設定

Agenda

- “クラウドとオンプレにおけるSAP S/4HANAのためのクラウド拡張オプションの”超”概要”
(2023年6月8日開催のパートナー様向けWebiner セッション)の振り返り
- SAP RESTful Application Programming Model (RAP)について
- **RAP ファサード / RAP BO Interface を用いるための EML (Entity Manipulation Language)**
- RAP での Behavior Definition および Behavior Implementation
- 追加トピック: S/4HANA PCE / オンプレミス環境でのATC in Eclipseについて
- ご参考情報

RAP における拡張



SAP S/4HANA Cloud のBusiness API, 拡張ポイントおよびイベント

リモートアクセス	(主に)ローカルアクセス	ローカル拡張ポイント
Side-by-Side 拡張 でのS/4HANA Cloud のビジネスオブジェクトのデータにアクセス	On-Stack 拡張 でのS/4HANA Cloud のビジネスオブジェクトのデータにアクセス	On-Stack 拡張 でのS/4HANA Cloud のビジネスオブジェクトのロジックを拡張
<ul style="list-style-type: none">• Odata サービス• SOAP サービス• イベント	<ul style="list-style-type: none">• CDS View***• RAP BO インターフェース (RAP ファサード)• クラス• イベント(計画中****)	<ul style="list-style-type: none">• Business Add-Ins (BAdIs)• RAP BO エクステンション

* リリース済みのS/4HANA Cloud のAPIについては製品ドキュメントおよび[SAP Business Accelerator Hub \(旧: SAP API Business Hub\)](#)をご確認ください。

** もしも上記でAPIが未リリースであることがわかった場合は、Customer Influence Channel for SAP S/4HANA Cloud (<https://influence.sap.com/sap/ino/#/campaign/2759>)からリクエストすることが可能です。

*** ODBC接続による外部ツールからにアクセスが可能 ([SAP Help](#))

**** 2023年Q2現在

*オリジナル文章に加筆
28

EML の概要

エンティティ 管理言語

ABAP 言語の拡張
SQL のような構文

RAP コンテキストでのトランザクション BO 動作の管理

RAP BO への API を用いた直接アクセス

標準 EML API

RAP BO へのタイプセーフなREADおよびModifyメソッドを用いたアクセス

一般 EML API

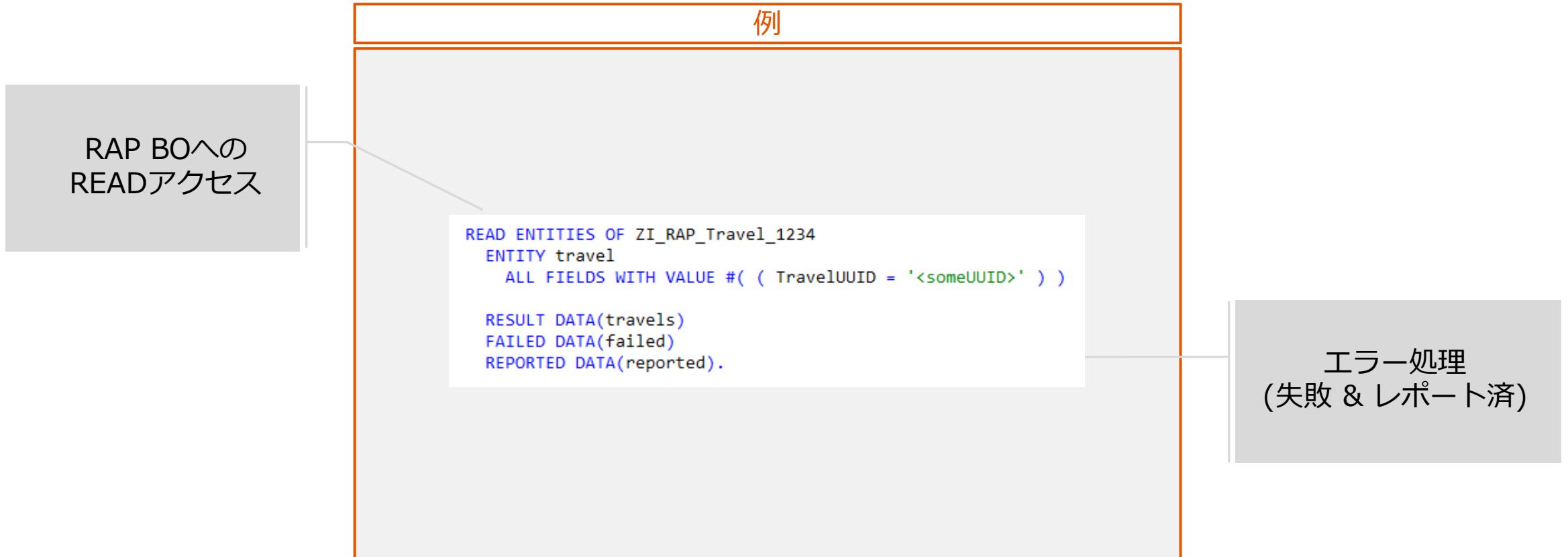
RAP BO の他のフレームワークへの統合

データベース LUW によってサポートされるデータ整合性

変更を保持するには COMMIT 操作が必要

エンティティ操作言語 (EML) の理解

READ オペレーション



エンティティ操作言語 (EML) の理解

MODIFY CREATE オペレーション

例

```
MODIFY ENTITIES OF ZI_RAP_Travel_1234
  ENTITY travel
    CREATE
      SET FIELDS WITH VALUE
        #( ( %cid      = 'MyContentID_1'
          AgencyID   = '70012'
          CustomerID = '14'
          BeginDate  = cl_abap_context_info=>get_system_date( )
          EndDate    = cl_abap_context_info=>get_system_date( ) + 10
          Description = 'I like RAP@openSAP' ) )

      MAPPED DATA(mapped)
      FAILED DATA(failed)
      REPORTED DATA(reported).

    COMMIT ENTITIES
    RESPONSE OF ZI_RAP_Travel_1234
    FAILED  DATA(failed_commit)
    REPORTED DATA(reported_commit).
```

RAP BO への
MODIFY CREATE
(新規インスタンス作
成)アクセス

エラー処理
(失敗 & レポート済)

適切な LUW 処理
のための
COMMIT 文

エンティティ操作言語 (EML) の理解

MODIFY UPDATE オペレーション

RAP BOへの
MODIFY UPDATE
(更新)アクセス

例

```
MODIFY ENTITIES OF ZI_RAP_Travel_1234
ENTITY travel
UPDATE
SET FIELDS WITH VALUE
#( ( TravelUUID = '<someUUID>'
    Description = 'I like RAP@openSAP' ) )

FAILED DATA(failed)
REPORTED DATA(reported).

COMMIT ENTITIES
RESPONSE OF ZI_RAP_Travel_1234
FAILED DATA(failed_commit)
REPORTED DATA(reported_commit).
```

エラー処理
(失敗 & レポート済)

適切な LUW 処理
のための
COMMIT 文

エンティティ操作言語 (EML) の理解

MODIFY DELETE オペレーション

RAP BOへの
UPDATE DELETE
アクセス

例

```
MODIFY ENTITIES OF ZI_RAP_Travel_1234
  ENTITY travel
    DELETE FROM
      VALUE
        #( ( TravelUUID = '<someUUID>' ) )

    FAILED DATA(failed)
    REPORTED DATA(reported).

  COMMIT ENTITIES
    RESPONSE OF ZI_RAP_Travel_1234
    FAILED   DATA(failed_commit)
    REPORTED DATA(reported_commit).
```

エラー処理
(失敗 & レポート済)

適切な LUW 処理
のための
COMMIT 文

エンティティ操作言語 (EML) の理解

MODIFY EXECUTE ACTION オペレーション

RAP BO に対する
特定のAction (標準オ
ペレーション以外の処
理)の実行

例

```
MODIFY ENTITIES OF ZI_RAP_Travel_1234
  ENTITY travel
    EXECUTE acceptTravel
    FROM VALUE
      #( ( TravelUUID = '<someUUID>' ) )

    RESULT DATA(result)
    MAPPED DATA(mapped)
    FAILED DATA(failed)
    REPORTED DATA(reported).

  COMMIT ENTITIES
    RESPONSE OF ZI_RAP_Travel_1234
    FAILED DATA(failed_commit)
    REPORTED DATA(reported_commit).
```

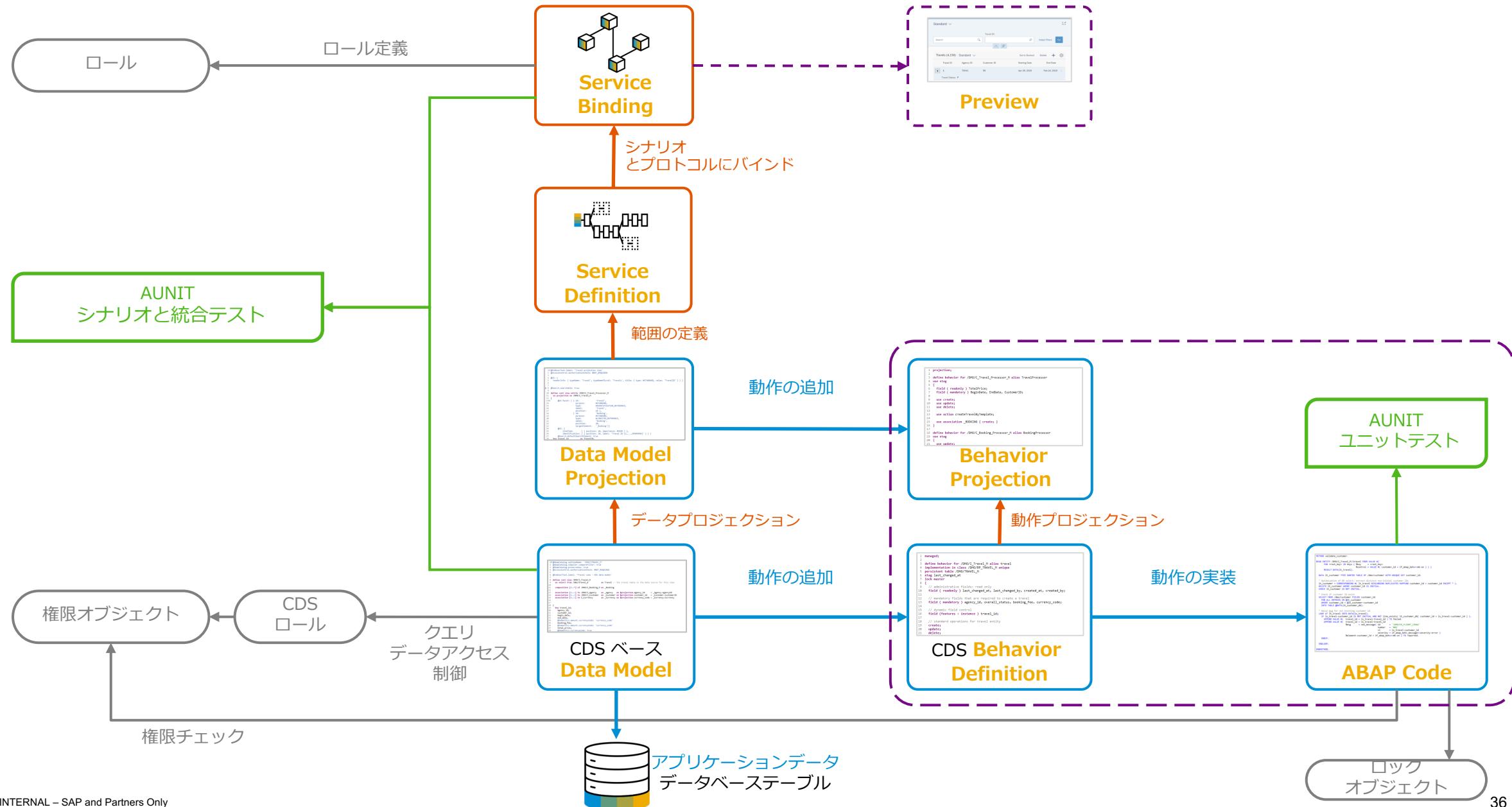
エラー処理
(失敗 & レポート済)

適切な LUW 処理
のための
COMMIT 文

Agenda

- “クラウドとオンプレにおけるSAP S/4HANAのためのクラウド拡張オプションの”超”概要”
(2023年6月8日開催のパートナー様向けWebiner セッション)の振り返り
- SAP RESTful Application Programming Model (RAP)について
- RAP ファーサード / RAP BO Interface を用いるための EML (Entity Manipulation Language)
- **RAP での Behavior Definition および Behavior Implementation**
- 追加トピック: S/4HANA PCE / オンプレミス環境でのATC in Eclipseについて
- ご参考情報

コード解説: データの更新を含むBO機能の実装



Behavior definition -サンプル

```
1 managed implementation in class ZBP_I_TRAVEL_M_510 unique;
2
3 define behavior for ZI_TRAVEL_M_510 alias Travel
4 persistent table ztravel_510
5 etag master last_changed_at
6 lock master
7
8 // key that will be automatically generated by the framework
9 field ( readonly, numbering : managed ) mykey;
10
11 // semantic key calculated in a determination
12 field ( readonly ) travel_id;
13
14 // administrative fields (read only)
15 field ( readonly ) last_changed_at, last_changed_by, created_at, created_by;
16
17 // mandatory fields that are required to create a travel
18 field ( mandatory ) agency_id, overall_status, booking_fee, currency_code;
19
20 // mandatory fields that are required to create a travel
21 field ( mandatory ) Begin_Date, End_Date, Customer_ID;
22
23 // standard operations for travel entity
24 create;
25 update;
26 delete;
27
28 // instance action and dynamic action control
29 action ( features : instance ) acceptTravel result [1] $self;
30
31 // validations
32 validation validateCustomer on save { field customer_id; }
33 validation validateDates on save { field begin_date, end_date; }
34 validation validateAgency on save
35 { field agency_id; }
36
37 // determination
38 determination CalculateTravelKey on modify
39 { create; }
40
41 }
```

myKey, travel_idなどのビジネスオブジェクトの管理項目を Readonly に設定

myKey項目はフレームワークが自動的にセット

その他の入力項目を必須項目に設定

Create, update, deleteといったスタンダードなオペレーションを利用を宣言

標準ではないオペレーションであるActionを定義。
Feature: instanceは特定のインスタンスを選択した場合に動作。\$selfはパラメータとしてインスタンスをActionに受け渡す。

Validation はビジネスオブジェクトがOn Saveの場合に動作する機能。文字通り値のチェックで利用

Determinationは値がOn Modify のオペレーションの中で動作

Behavior Projection –サンプル

```
1  projection;
2  //strict ( 1 ); //Uncomment this line in order to enable strict validation
3
4  define behavior for ZC_TRAVEL_M_510 alias TravelProcessor
5  use etag
6  {
7      use create;
8      use update;
9      use delete;
10
11     use action acceptTravel;
12 }
```

Behavior Implementation –サンプル

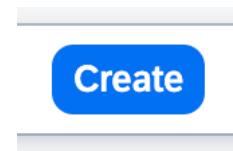
→ 2.Save シークエンス(On Save)で動作

```
1  /*** use this source file for the definition and implementation of
2  ** local helper classes, interface definitions and type
3  ** declarations
4
5 CLASS lhc_travel DEFINITION INHERITING FROM cl_abap_behavior_handler.
6
7 PRIVATE SECTION.
8
9   TYPES tt_travel_update TYPE TABLE FOR UPDATE zi_travel_m_510.
10
11 METHODS validate_customer          FOR VALIDATE ON SAVE IMPORTING keys FOR travel~validateCustomer.
12 METHODS validate_dates             FOR VALIDATE ON SAVE IMPORTING keys FOR travel~validateDates.
13 METHODS validate_agency            FOR VALIDATE ON SAVE IMPORTING keys FOR travel~validateAgency.
14
15 METHODS set_status_completed      FOR MODIFY IMPORTING keys FOR ACTION travel~acceptTravel      RESULT result.
16 METHODS get_features               FOR FEATURES IMPORTING keys REQUEST requested_features FOR travel      RESULT result.
17
18 METHODS CalculateTravelKey        FOR DETERMINE ON MODIFY IMPORTING keys FOR Travel~CalculateTravelKey.
19
20
21 ENDCLASS.
```

→ 1.Create オペレーション (On Modify)で動作

3.Actionで動作 ←

Behavior Implementation – Modify Create オペレーション



1. Modify Create
オペレーション起動

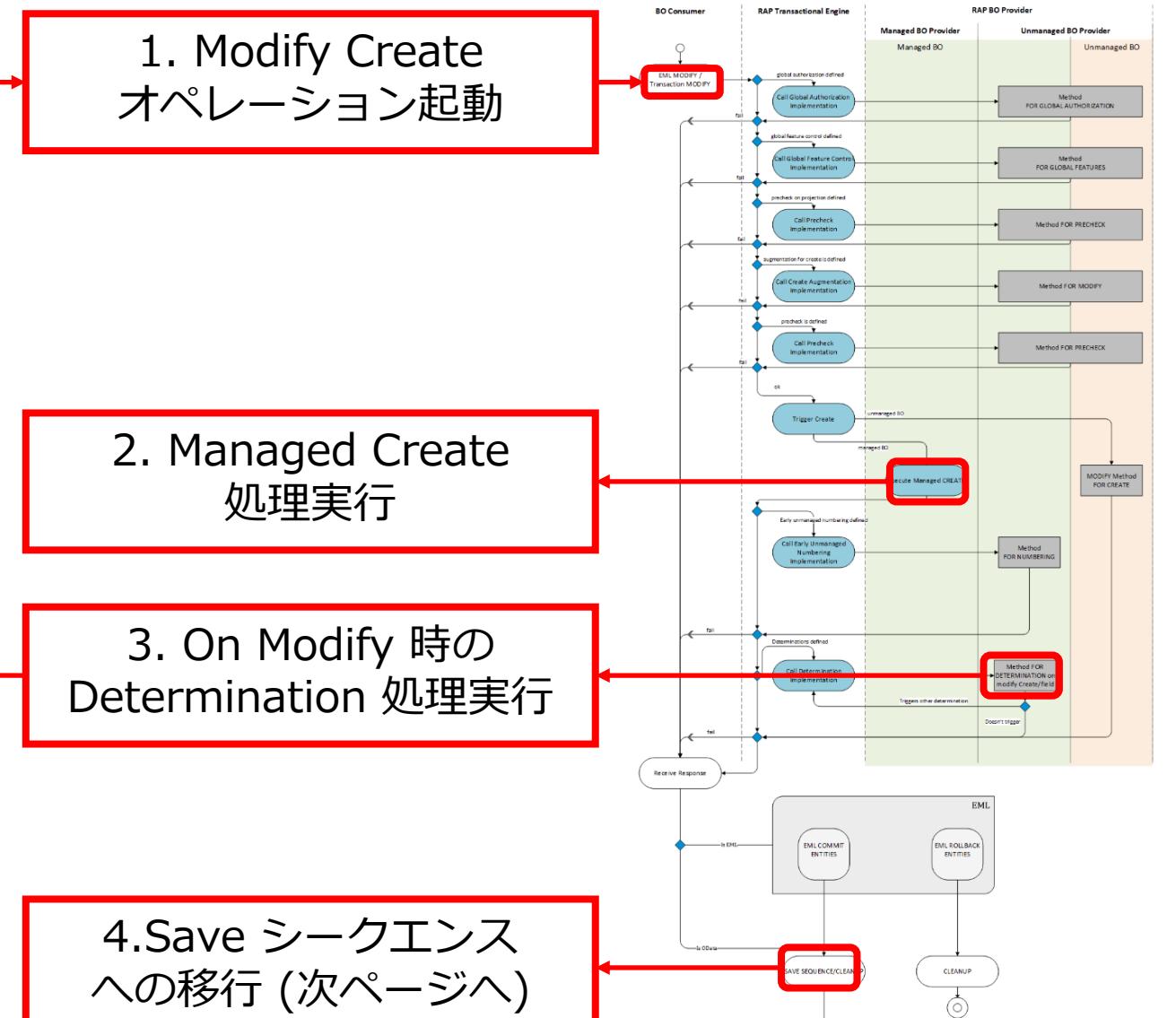
2. Managed Create
処理実行

3. On Modify 時の
Determination 処理実行

METHODS CalculateTravelKey

4. Save シーケンス
への移行 (次ページへ)

Create 時のランタイムダイアグラム



Behavior Implementation – Save シーケンス

[Save シーケンスのランタイムダイアグラム](#)

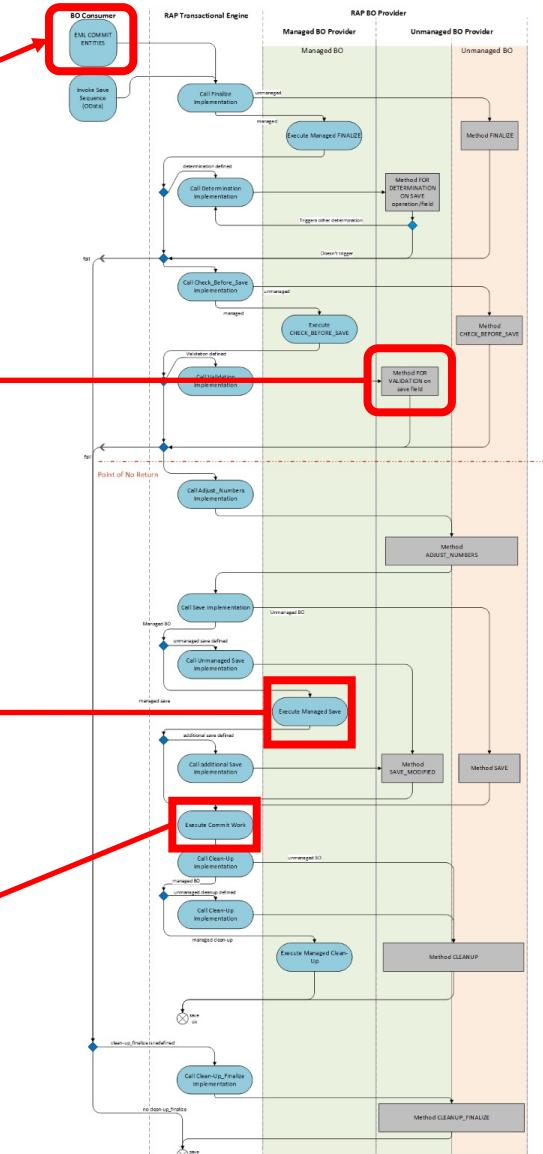
```
METHODS validate_customer  
METHODS validate_dates  
METHODS validate_agency
```

1. Save シーケンス
(前ページより)

2. On Save 時の
Validation 处理実行

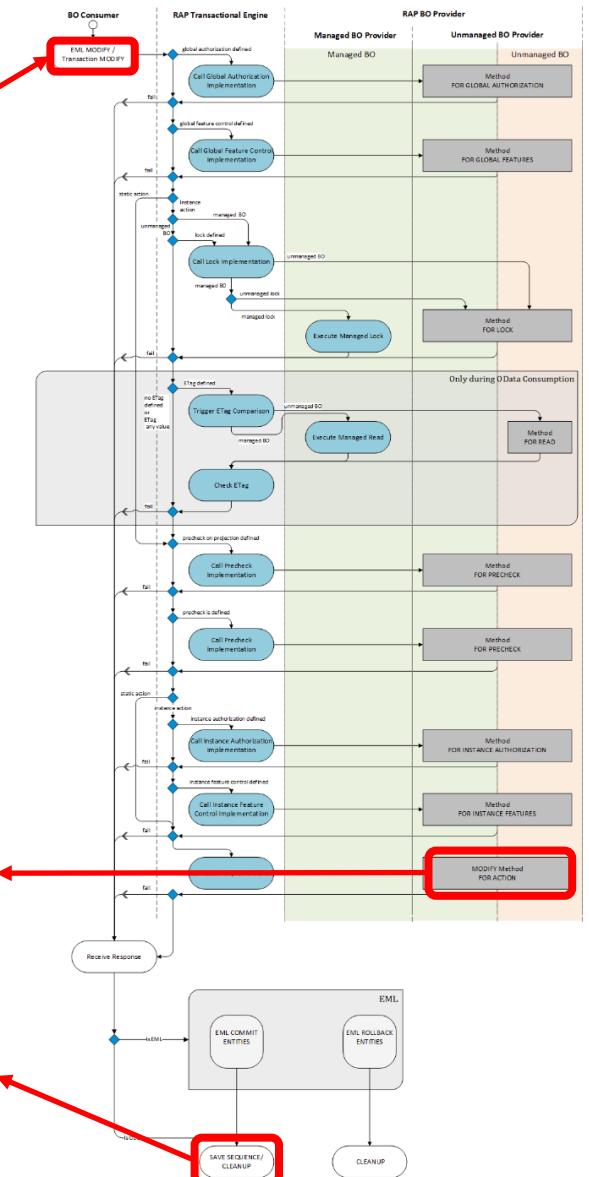
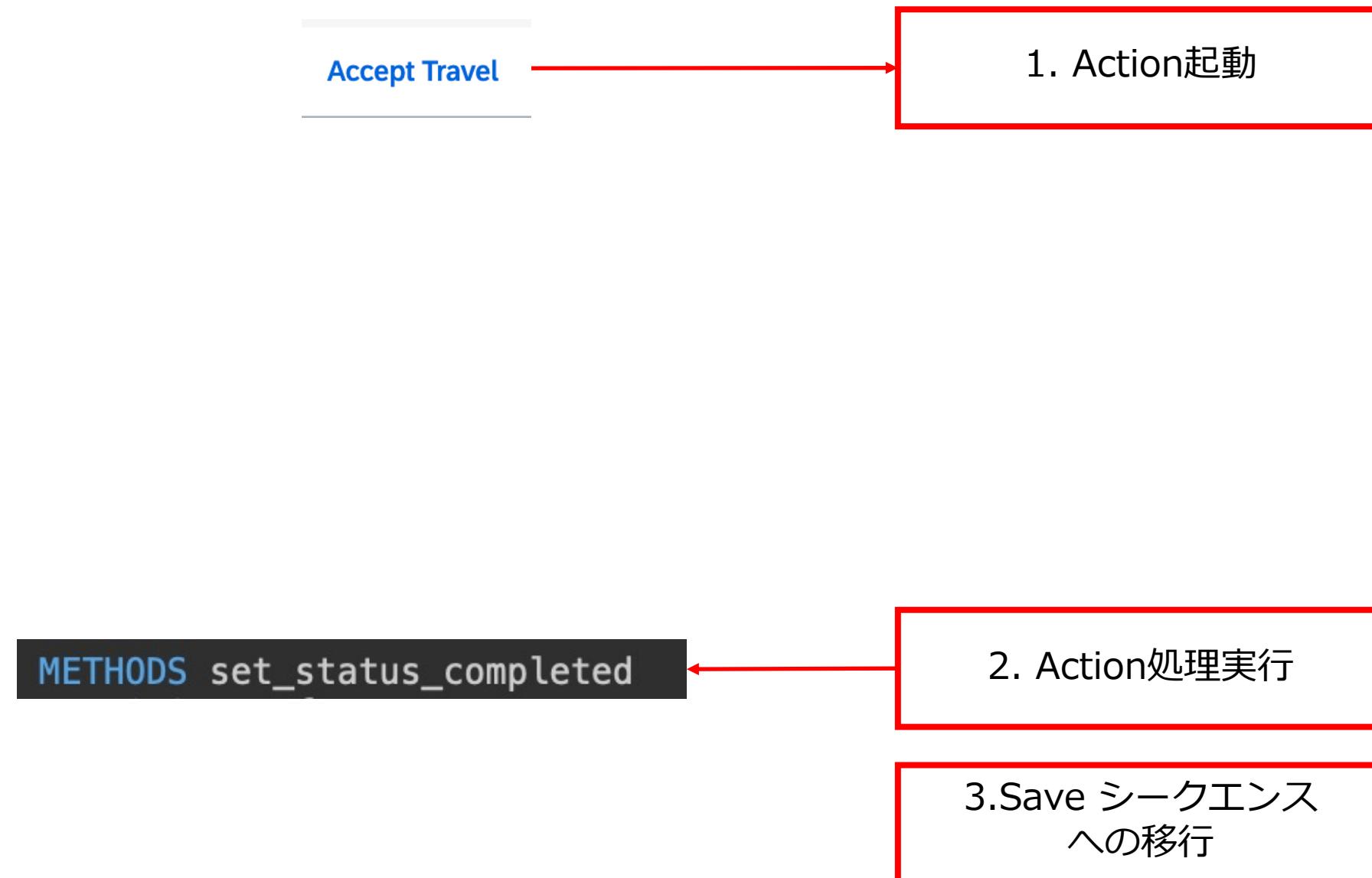
3. Managed Save
処理実行

4. Commit Work
処理実行



Behavior Implementation – Action

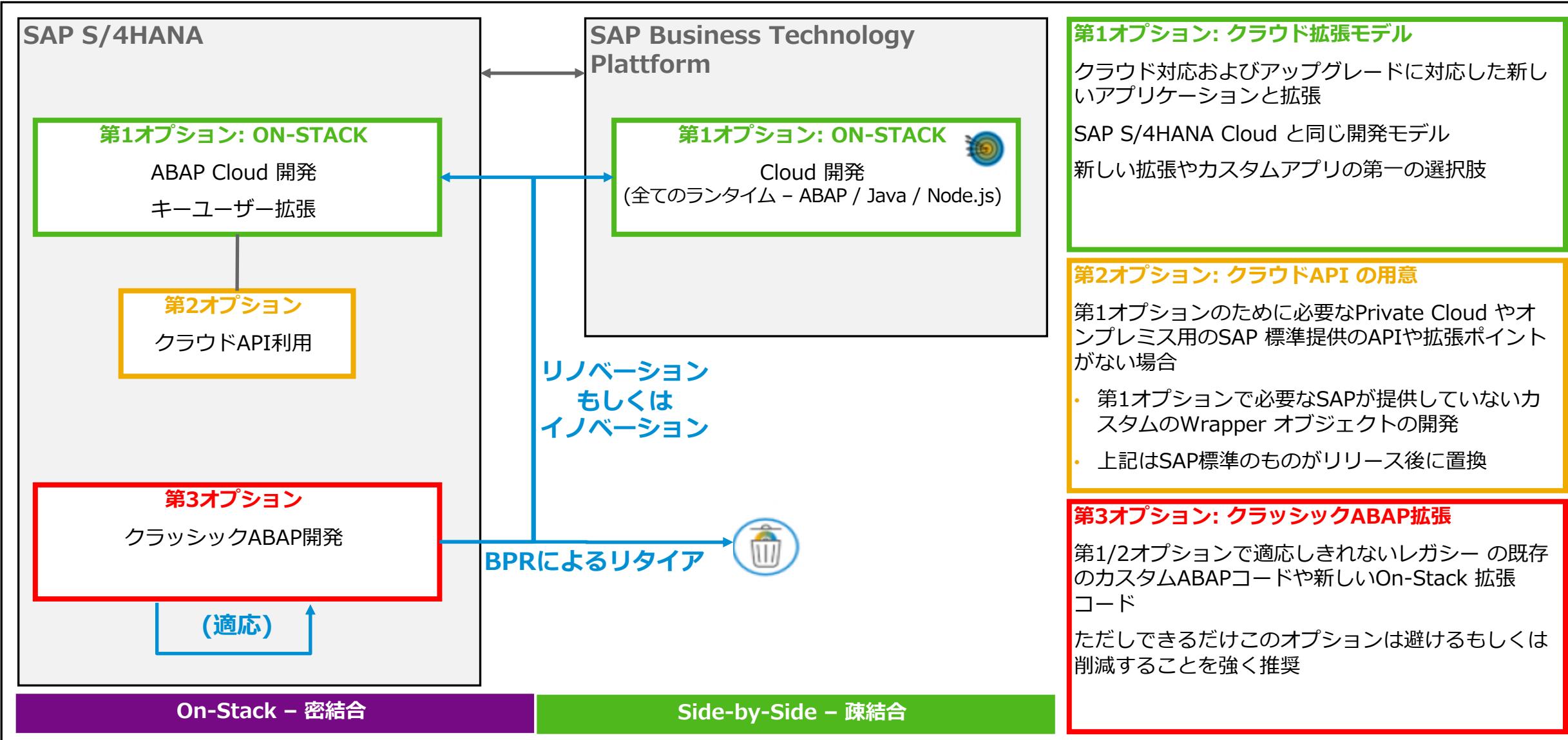
Actionのランタイムダイアグラム



Agenda

- “クラウドとオンプレにおけるSAP S/4HANAのためのクラウド拡張オプションの”超”概要”
(2023年6月8日開催のパートナー様向けWebiner セッション)の振り返り
- SAP RESTful Application Programming Model (RAP)について
- RAP ファーサード / RAP BO Interface を用いるための EML (Entity Manipulation Language)
- RAP での Behavior Definition および Behavior Implementation
- 追加トピック: **S/4HANA PCE / オンプレミス環境でのATC in Eclipseについて**
- ご参考情報

S/4HANA Cloud private Edition およびオンプレミスにおける拡張の考え方



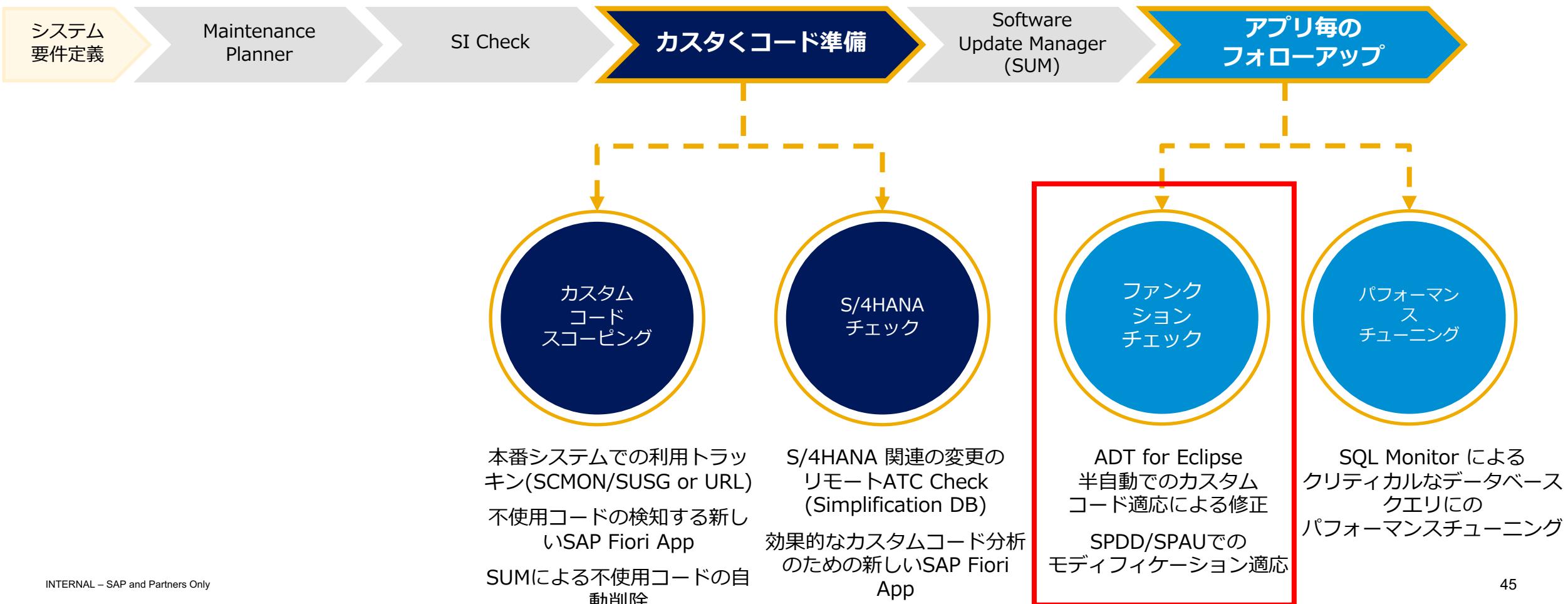
SAP S/4HANAへのシステムコンバージョンのための カスタムコードマイグレーションプロセスの全体像*

*詳細は

[Custom Code Migration Guide for SAP S/4HANA](#)
[2022 Feature Package Stack 01](#)

(2023年Q3時点で最新)をご確認ください。

準備フェーズ



Eclipse ADT でのクラウドの半自動コード調整

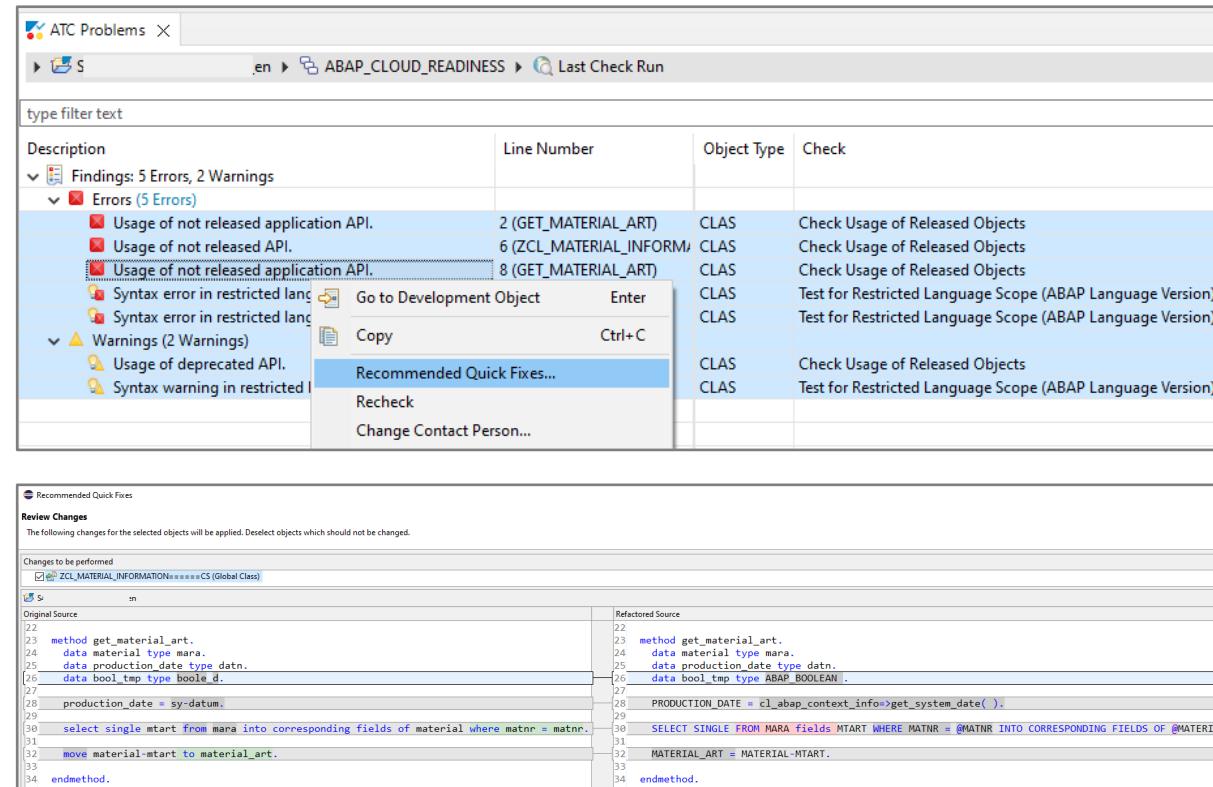
適応作業を最小化するための ADT の有効クリックフィックス

- ▶ ABAP 言語バージョンクラウドへの適応
- ▶ リリース済オブジェクトの用途への適応
- ▶ SY 項目への直接アクセスではなく API 呼出

例：クリックフィックスの使用ケース

- ▶ データエレメント置換
- ▶ リリース済データエレメント
(->ABAP_BOOLEAN)
- ▶ 構文エレメントを新しい構文エレメント (例:MOVE)

Eclipse 版 ABAP 開発ツールの ATC Problems ビューのクリックフィックス



カスタムコード分析の推奨事項

- リリース済の API の使用に関する分析に最初に焦点を当てます。
- 公開リリース済 API が SAP S/4HANA Public Cloud で利用可能かどうかの確認
- SAP S/4HANA 2022 では、後継のリリース済 API が Eclipse の ADT の ATC 結果に表示されます。
- 以前の SAP S/4HANA リリースでは、SAP ノート [3088062 - Identifying local APIs for S/4HANA Cloud \(S/4HANA Cloud のローカル API の特定\)](#) を参照してください。
- 公開リリースされていない API は、SAP Influence キャンペーン ([SAP S/4HANA Cloud および SAP S/4HANA for On-Stack API](#)) で依頼できます。

ATC 結果でリリースされた API

```
10 class zcl_get_material_information > a GET_MATERIAL_ART
11 public
12 	* Line 13 is marked with a red error indicator
13 	create public .
14
15 	public section.
16 	protected section.
17 	private section.
18 	methods get_material_art
19   	importing
20     	matnr type matnr
21   	protecting
22     	mara_start type mara_start.
23   	exception
24     	material_not_found.
25 	endclass.
```

Global Class | Class-relevant Local Types | Local Types | Test Classes | Macros
ATC Problems > C _dolinskaja_en \ ABAP_CLOUD_READINESS > Last Check Run
Type Filter Details
Description
Findings: 3 Errors
Errors (3 Errors)
• Usage of not released application API
• Usage of not released application API
• Usage of not released application API
Appl. Comp. Check / Check Class / Message Code
BC-DV-B-UTL-CLS / CL_CLS_CI_CHECK_ENVIRONMENT /
Details of Analysis
Usage of not released application API. Use of Table MARA is not permitted. Use CDS Entity L_PRODUCT instead.
• Cannot be referenced using a pragma or pseudo-comment
• Component Object (MARA_TABLE)
• Component ID: LO-MD-MM
• Referenced Package: MG
• Referenced Software Component: S4CORE

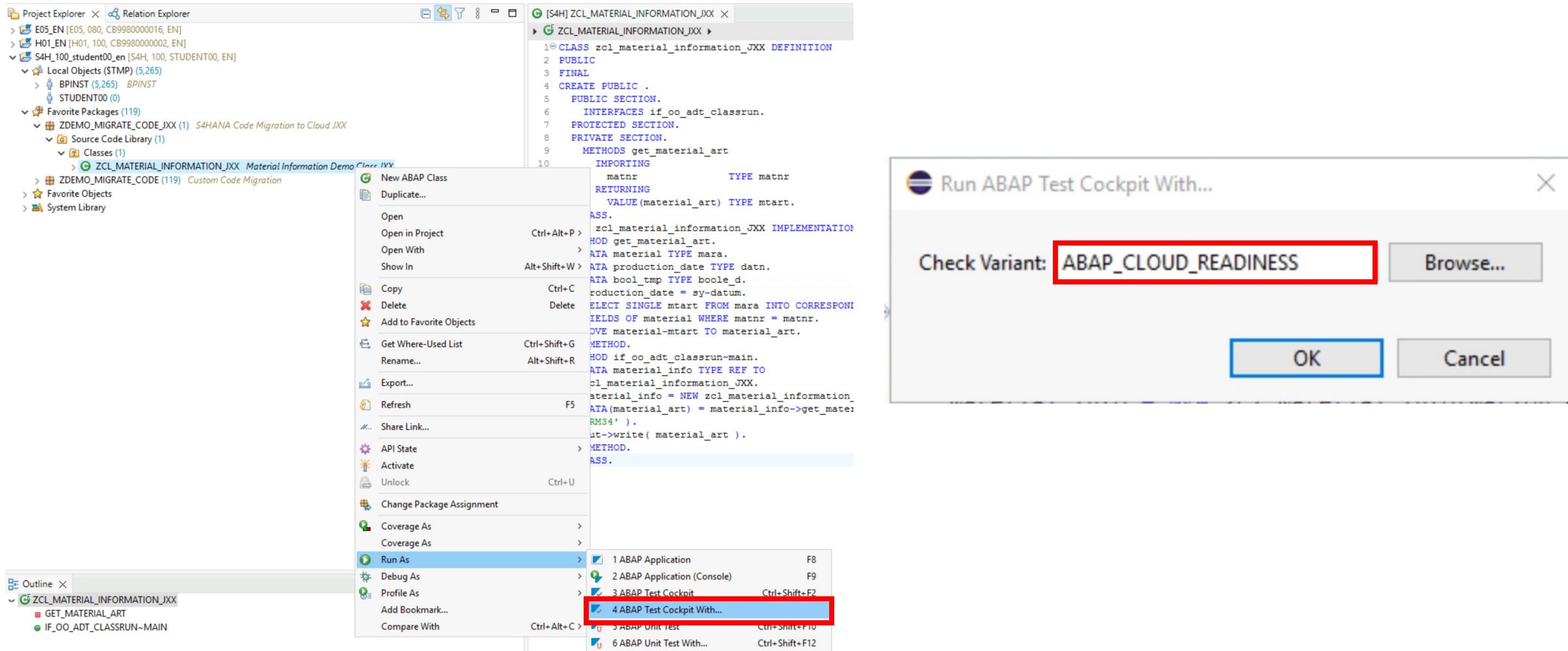
不足しているリリース済 API の依頼

Customer Influence opportunity
Give us your feedback! We will accept requests for new APIs, enhancements on existing APIs for the SAP S/4HANA Cloud and SAP S/4HANA product family.
Influence campaign: [SAP S/4HANA Cloud & SAP S/4HANA for On-stack API](#)



[SAP Blog : How to make your custom code cloud-ready and upgradeable](#)

Eclipse ABAP Test Cockpit (ATC)の起動



Eclipse ABAP Test Cockpitの実行結果

The screenshot shows the Eclipse ABAP Test Cockpit interface. The breadcrumb navigation at the top indicates the path: S4H_100_student00_en > ABAP_CLOUD_READINESS > Last Check Run. A search bar labeled "type filter text" is present. On the left, a tree view displays "Findings: 5 Errors, 2 Warnings". Under "Errors (5 Errors)", three items are listed: "Usage of not released application API.", "Usage of not released ABAP Platform APIs.", and "Usage of not released application API.". A fourth item, "Syntax error in restrict", is highlighted with a yellow warning icon. A context menu is open over this item, with the option "Recommended Quick Fixes..." highlighted by a red rectangle. Other menu options include "Go to Development Object", "Copy", "Change Contact Person...", "Group By", and "Configure Columns". The right side of the interface shows "Details" for the selected finding, including the class and message code.

type filter text

Description

Lin...

Findings: 5 Errors, 2 Warnings

Errors (5 Errors)

- Usage of not released application API. 2 (GET)
- Usage of not released ABAP Platform APIs. 6 (ZCL)
- Usage of not released application API. 6 (GET)
- Syntax error in restrict
- Syntax error in restrict

Warnings (2 Warnings)

- Usage of deprecated
- Syntax warning in res

Appl. Comp. Check / Check Class / Message Code

BC-DWB-UTL-CLS / CL_CLS_CI_CHECK_ENVIRONMEM

Go to Development Object Enter

Copy Ctrl+C

Recommended Quick Fixes...

Change Contact Person...

Group By

Configure Columns

INTERNAL – SAP and Partners Only

Eclipse ABAP Test Cockpitによる半自動コード修正

The screenshot shows the Eclipse ABAP Test Cockpit interface. On the left, the 'Original Source' tab displays the initial ABAP code. On the right, the 'Refactored Source' tab shows the same code with various parts highlighted in red boxes, indicating automated code fixes. The annotations in the refactored code include:

- * Quick Fix Replace 'OLD SQL' with NEW SQL.
- * 06/15/2023 12:49:39 STUDENT00
- * Transport S4HK902422 TR for S4HANA Code Migration to Cloud JXX
- * Replaced Code:
 - SELECT SINGLE mtart FROM mara INTO CORRESPONDING FIELDS OF material WHERE matnr = matnr.
 - SELECT SINGLE FROM MARA fields MTART WHERE MATNR = @MATNR INTO CORRESPONDING FIELDS OF @MATERIAL.
- * End of Quick Fix
- * Quick Fix Replace MOVE with an assignment.
- * 06/15/2023 12:49:39 STUDENT00
- * Transport S4HK902422 TR for S4HANA Code Migration to Cloud JXX
- * Replaced Code:
 - MOVE material-mtart TO material_art.
 - MATERIAL_ART = MATERIAL-MTART.
- * End of Quick Fix
- * End of Quick Fix

At the bottom, there are navigation buttons: '?', '< Back', 'Next >', 'Finish', and 'Cancel'.

コード修正後のテスト実行結果

The screenshot shows the SAP ABAP Development Workbench. On the left, the code editor displays class ZCL_MATERIAL_INFORMATION_JXX. A red box highlights the line 'INTERFACES if oo adt classrun.' in the PUBLIC SECTION. On the right, the ABAP Console window shows the output 'ROH' with a red box around it. A red arrow points from the highlighted code in the editor to the 'ROH' output in the console.

```
[S4H] ZCL_MATERIAL_INFORMATION_JXX X
ZCL_MATERIAL_INFORMATION_JXX > GET_MATERIAL_ART
  FINAL
  CREATE PUBLIC .
  PUBLIC SECTION.
    INTERFACES if oo adt classrun. // Red box here
  PROTECTED SECTION.
  PRIVATE SECTION.
    METHODS get_material_art
      IMPORTING
        matnr          TYPE i_product-product
      RETURNING
        VALUE(material_art) TYPE i_product-producttype.
  ENDCLASS.
  CLASS zcl_material_information_JXX IMPLEMENTATION.
    METHOD get_material_art.
      DATA material TYPE i_product.
      DATA production_date TYPE datn.
      DATA bool_tmp TYPE boole_d.
      production_date = sy-datum.
    ENDMETHOD.
  ENDCLASS.
  * Quick Fix Replace 'OLD SQL' with NEW SQL.
  * 06/15/2023 12:49:39 STUDENT00
  * Transport S4HK902422 TR for S4HANA Code Migration to Cloud JXX
  * Replaced Code:
  *   SELECT SINGLE mtart FROM mara INTO CORRESPONDING
  *     FIELDS OF material WHERE matnr = matnr.
  *
  * SELECT SINGLE      FROM i_product FIELDS producttype
```

ABAP Console
ROH

Interface **if_oo_adt_classrun**を用いて
ABAP Console への出力

Agenda

- “クラウドとオンプレにおけるSAP S/4HANAのためのクラウド拡張オプションの”超”概要”
(2023年6月8日開催のパートナー様向けWebiner セッション)の振り返り
- SAP RESTful Application Programming Model (RAP)について
- RAP ファーサード / RAP BO Interface を用いるための EML (Entity Manipulation Language)
- RAP での Behavior Definition および Behavior Implementation
- 追加トピック: S/4HANA PCE / オンプレミス環境でのATC in Eclipseについて
- ご参考情報

RAP 関連ドキュメント

SAP S/4HANA Cloud, ABAP Environment	SAP S/4HANA, ABAP Environment (1909~*)	SAP BTP, ABAP Environment
<ul style="list-style-type: none"><input type="checkbox"/> リリースノート<input type="checkbox"/> RAP 開発者ガイド<input type="checkbox"/> ABAP キーワードドキュメント<input type="checkbox"/> 外部のODBCツールから ABAP-マネージドデータへのアクセス<input type="checkbox"/> その他のABAP開発ユーザーガイド	<ul style="list-style-type: none"><input type="checkbox"/> リリースノート<input type="checkbox"/> RAP 開発者ガイド<input type="checkbox"/> ABAP キーワードドキュメント<input type="checkbox"/> 外部のODBCツールから ABAP-マネージドデータへのアクセス<input type="checkbox"/> その他のABAP開発ユーザーガイド	<ul style="list-style-type: none"><input type="checkbox"/> リリースノート<input type="checkbox"/> RAP 開発者ガイド<input type="checkbox"/> ABAP キーワードドキュメント<input type="checkbox"/> SAP Analytics Cloud への分析データモデルとLiveデータ接続<input type="checkbox"/> 外部のODBCツールから ABAP-マネージドデータへのアクセス<input type="checkbox"/> その他のABAP開発ユーザーガイド

*RAP自体は1909以降で対応可能ですが、S/4HANA Cloud, Public Edition と同等の機能は2022以降になります。

ご参考情報

- SAP Tutorial: [Develop a Fiori App Using RESTful Application Programming Model \(Managed Scenario\) on SAP BTP, ABAP Environment](#)
- 本資料作成のために引用した上記チュートリアルの解説SAP Blog
 - Get Know about RAP : [Introduction](#)
 - Get Know about RAP : [Define Data Model – Part 1](#)
 - Get Know about RAP : [Define CDS-based data model – Part 2](#)
 - Get Know about RAP : [CDS Behavior Definition](#)
 - Get Know about RAP : [Enhance Behavior with action and validation](#)
- SAP Help – Behavior definition/Metadata Extension 関連のアノテーション
 - [UI Annotation](#)
- SAP Help - Behavior implementation関連
 - [Operations](#) – managed operation やunmanaged operationなど
 - Managed operation に関する処理 ([Determinations](#), [Validations](#), [Save Options](#)など)
 - [Action](#) – Modify 処理を定義する標準ではないオペレーション
 - [Entity Manipulation Language \(EML\)](#)

7/5補足情報

- RAPを用いたExcel Upload の開発(SAP Blog)
<https://blogs.sap.com/2022/01/19/excel-upload-using-rap-part-1/>
- RAPのLOCK機構について (SAP Help)
<https://help.sap.com/docs/btp/sap-abap-restful-application-programming-model/concurrency-control?locale=en-US>
<https://help.sap.com/docs/btp/sap-abap-restful-application-programming-model/update-operation-runtime?locale=en-US>
- RAP BO Interface Product DescriptionにCreate オペレーションがない (Business Accelerator Hub)
https://api.sap.com/bointerface/I_PRODUCTTP_2
- EMLのModify Entity の書式 (SAP Help)
https://help.sap.com/doc/abapdocu_cp_index_htm/CLOUD/en-US/index.htm?file=abapmodify_entity_entities.htm
- If_oo_adt_classrunのSAP Blog (SAP Blog)
<https://blogs.sap.com/2021/02/01/printing-to-abap-console/>
- Draft データの削除 (SAP Help)
<https://help.sap.com/docs/btp/sap-abap-restful-application-programming-model/deleting-instances-of-draft-bo?locale=en-US&version=Cloud>



無料の openSAP コース

ABAP RESTful Application Programming Model (RAP) によるアプリの構築
(英語コース)

セルフペースモード

- 第 1 週目：導入
- 第 2 週目：読み取り専用一覧レポートアプリの開発
- 第 3 週目：アプリの開発
- 第 4 週目：既存のコードトランザクション動作の有効化
- 第 5 週目：サービス利用と Web APIを処理
- 第 6 週目：最終試験

今すぐ受講！

<https://open.sap.com/courses/cp13>



ご清聴ありがとうございます！

コンタクト先:

Speaker's Name
Email

