SAP NOW AI Tour

Unleash your future with business AI.

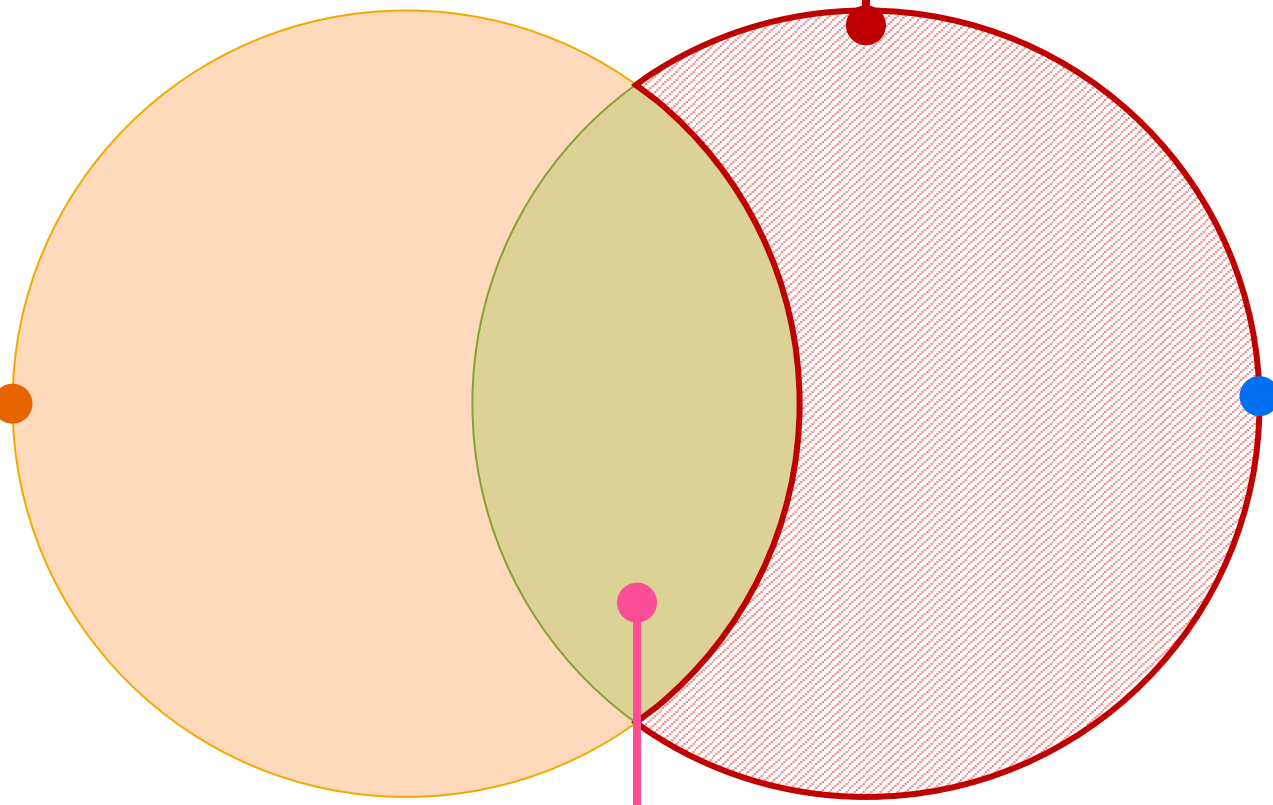
클린코어 실현을 위한 생성형 AI개발 서비스 활용



BTP Solution Advisory

한정우 파트너

ERP의 표준 기능과 확장의 범위



Uncovered scope
Required by customer.
Not yet covered by SAP
standard but might be in future

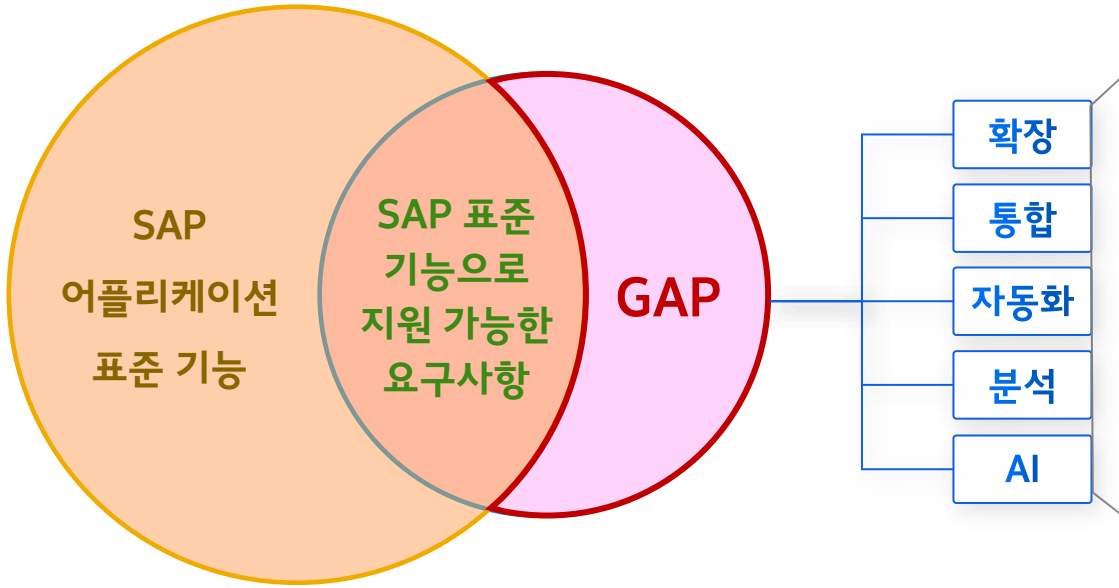
Required scope
Required by customer

Delivered scope
SAP standard
Cloud software

Covered scope
Required by customer,
covered by SAP standard

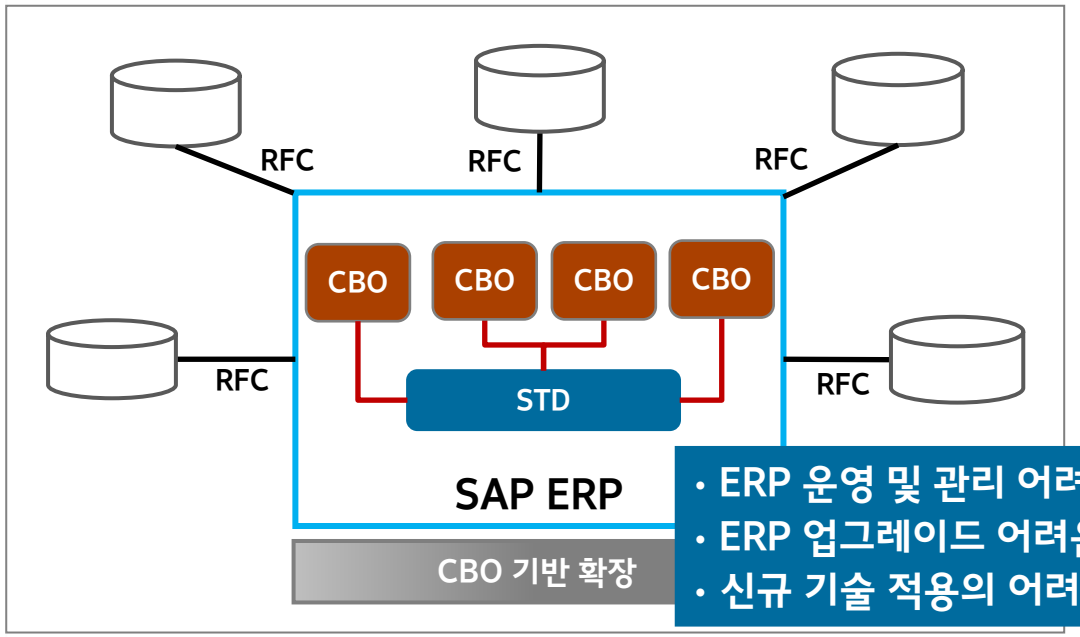
INTERNAL – SAP and Partners Only

전통적인 ERP 아키텍처 및 확장 방법



확장
통합
자동화
분석
AI

전통적인 ERP 아키텍처



- On-Premise 기반 ERP 개발 및 운영
- 기능 중심의 ERP 내부 Custom 개발 중심
- 솔루션 內 표준 기능과 CBO를 혼재해서 개발
- CBO 개발을 위해 표준 테이블, object에 직접 access

INTERNAL – SAP and Partners Only

Custom Code 고객 현황 예시
SAP의 다양한 프로젝트와 서비스 경험에 기반한 고객 상황

Up to 250.000 objects in customer namespaces

>2,000 modification and enhancements

Custom Code Object의 50% 이상은 지난 한달간 사용되지 않음 → 운영 효율화를 위한 Custom Code 주기적 retirement 필요

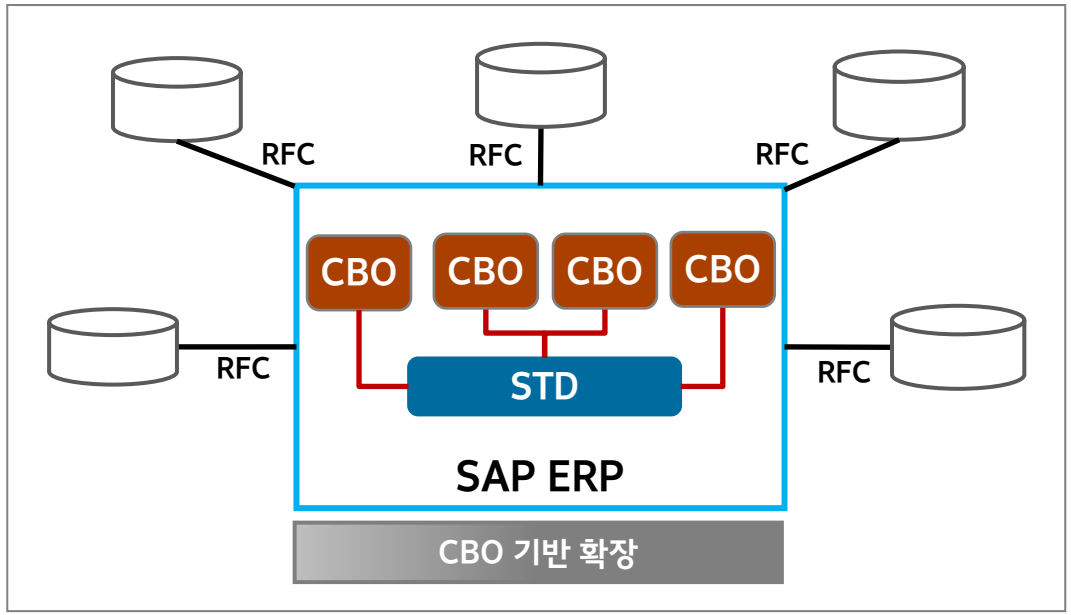
Custom Code Object의 23%가 핵심 프로세스를 지원 → Standard 와 Custom Code의 decoupling 필요

Custom Code Object의 50%는 Upgrade 과정에서 조정 필요 → 솔루션 Upgrade 과정에서 영향도 점검 및 수정 필요

INTERNAL – SAP and Partners Only

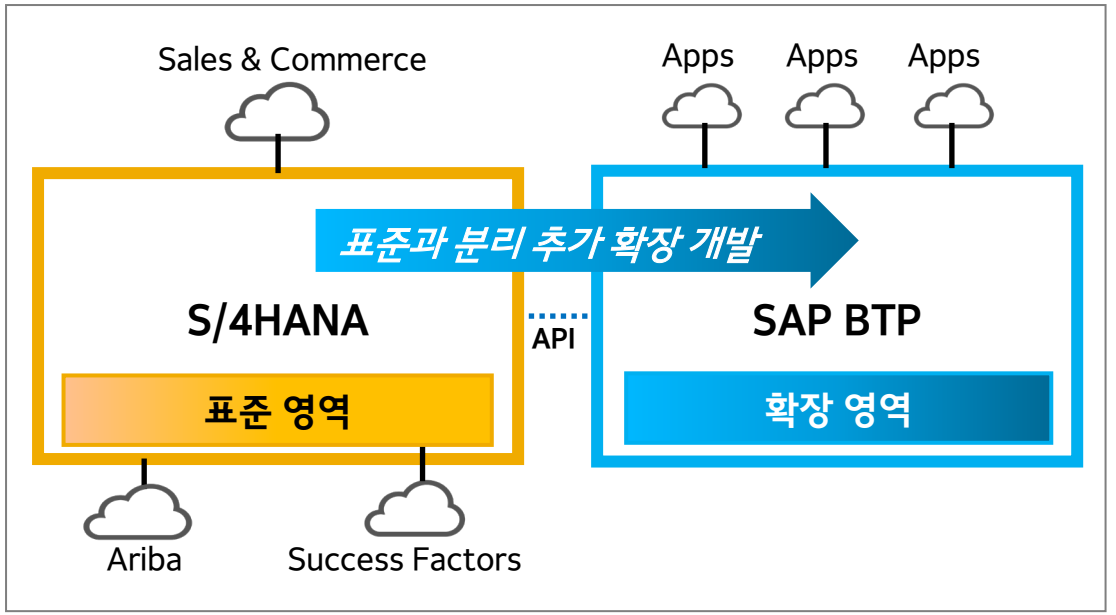
확장과 업그레이드에 유연한 Composable 아키텍처로 변화 (Clean Core)

전통적인 ERP 아키텍처



- On-Premise 기반 ERP 개발 및 운영
- 기능 중심의 ERP 내부 Custom 개발 중심
- 솔루션 內 표준 기능과 CBO를 혼재해서 개발
- CBO 개발을 위해 표준 테이블, object에 직접 access

차세대 ERP 추진 아키텍처

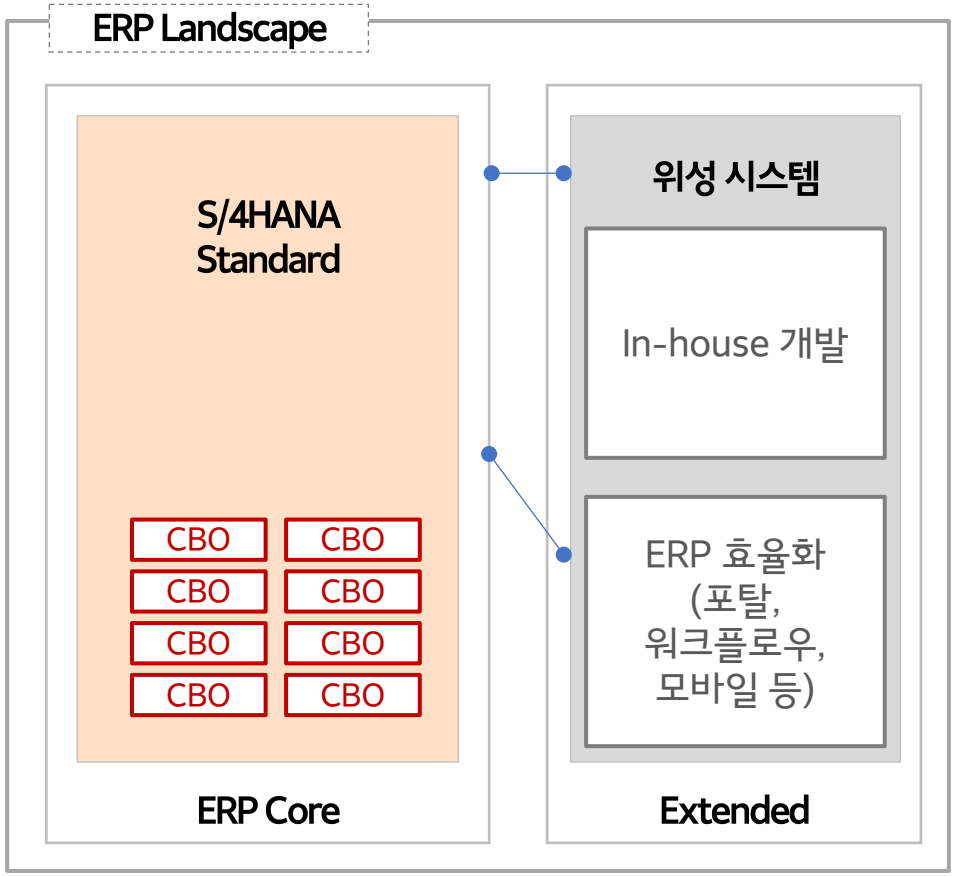


- Cloud 기반 ERP 개발 및 운영
- ERP 표준기능(Standard) 중심의 Clean Core 유지
- Core의 표준 기능과 확장(CBO) 개발의 분리 구성
- 개방형 Open API 기반 Core와 확장 기능의 연결

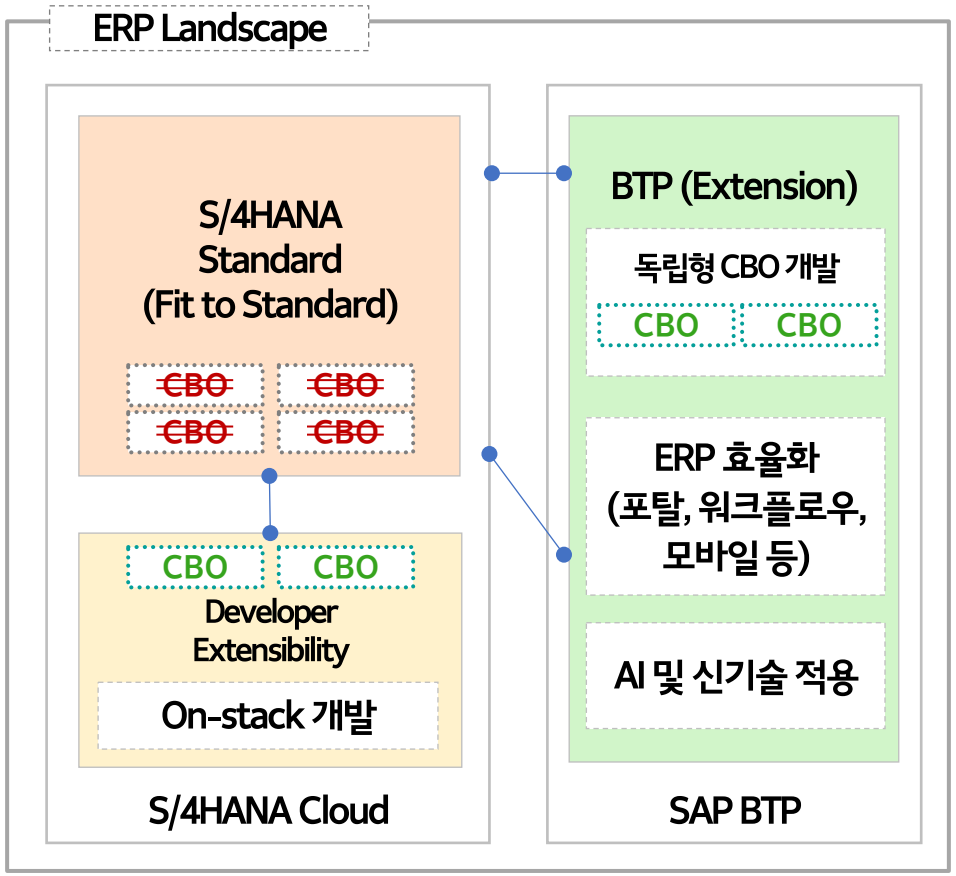
INTERNAL – SAP and Partners Only

Clean Core 적용 방안

전통적인 ERP 구성 모델



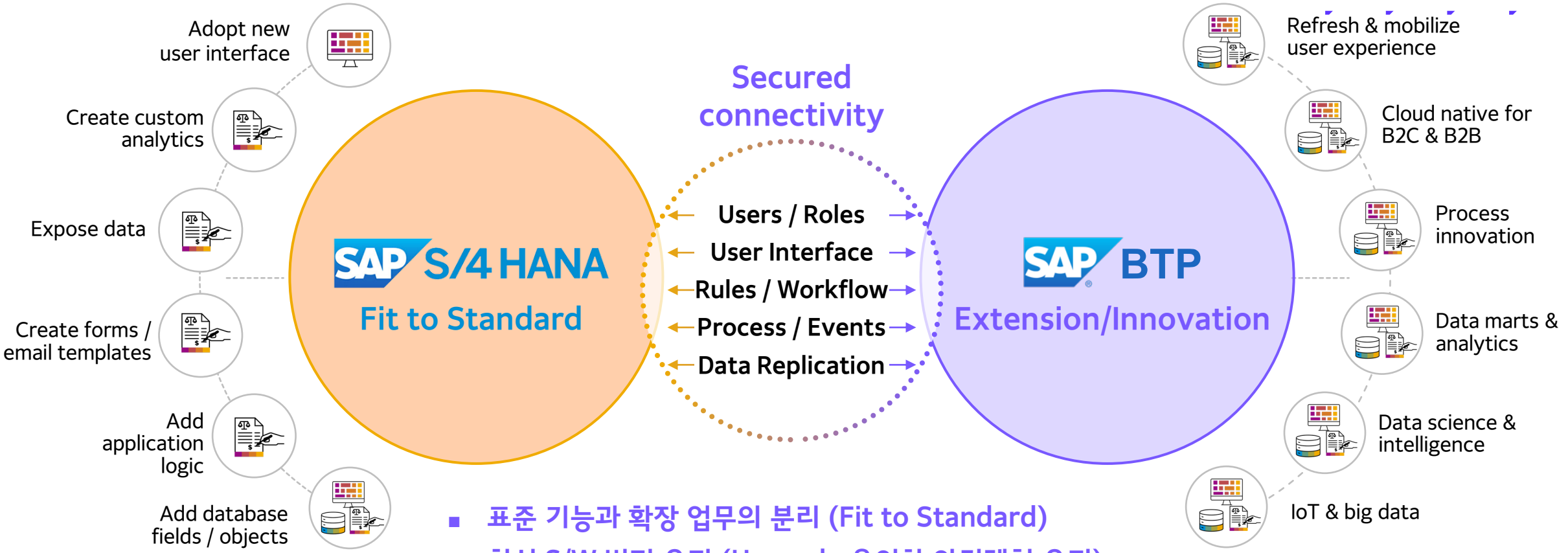
Clean Core ERP 구성 모델



CBO Clean Core 방식 개발

INTERNAL – SAP and Partners Only

어플리케이션 아키텍처의 변화
Composable & Modular Architect



- 표준 기능과 확장 업무의 분리 (Fit to Standard)
- 최신 S/W 버전 유지 (Upgrade 용이한 아키텍처 유지)
- 비지니스 변화의 빠른 대응을 위한 유연한 아키텍처
- 빠른 업무 확장과 AI/ML 등 신기술 적용이 유연한 플랫폼

INTERNAL – SAP and Partners Only

Clean Core : 공통된 고민

- Clean Core 전략과 필요성에 대해서는 대부분 동의
- 그런데 어디서부터 어떻게 시작해야 하나요? (S/4HANA Cloud 도입 전 / 후?)
- (CCA 분석 기반) 클린 코어를 위한 Custom Program 개발은 어떻게 해야 할까요?
- Key Takeaways

INTERNAL – SAP and Partners Only

Clean Core Assessment (CCA) 프로그램

Clean Core 전략을 완성을 위해 사전 ERP 현황 분석 및 향후 방향성 제시

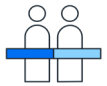
운영 중인 ERP에서 개발 오브젝트 및 CBO 프로그램들을 분석하여, SAP S/4HANA 로 전이 시 Clean Core 전략을 달성하고, 향후 Clean Core 아키텍처 및 전략 수립 방안 등을 제안해 드리는 무상 프로그램

[Day 1]
Clean Core Assessment 소개



[1 ERP 시스템 Pre-Requisite 적용]
** SUSG 활성화 in PRD*
[5 ~7 Days]

[1차 분석 전]
2 SAP BTP 주요 서비스 및 기술 세션



[3 1차 CCMA 분석 세션]
Top 30~50 프로그램 도출
[3 ~ 5 Weeks]

[6~ N Weeks]
4 최종 Top 30 프로그램 상세 분석
Based on AI Tool by SAP



[5 Clean Core 최종 결과 공유 세션]
[1 Day]

권장 아키텍처 및 가이드 지원 세션



목적

- S/4HANA Cloud 도입 전/후, 현 ERP 인스턴스 전반 이해와 향후 Clean Core 전략 수립을 위한 방향성 제시

주요 내용

- S/4HANA에서 제공하는 기능 및 API 가이드
- Clean Core 에 위배 및 권장사항 제시
- Side-by-Side 아키텍처 및 필요한 BTP 서비스 제언
- SAP Business AI 활용 추천

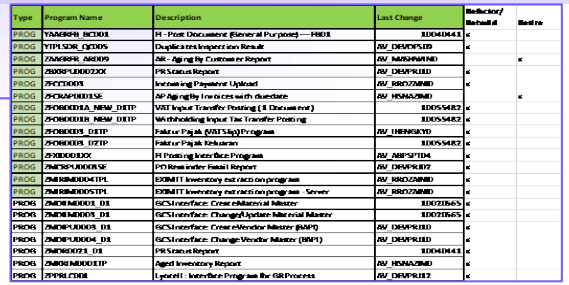
기대 효과

- 향후 S/4HANA의 Clean Core 전략 수립 및 방향성 이해 증대
- 내부 의사 결정에 필요 자료로 활용

INTERNAL – SAP and Partners Only

Clean Core Assessment (CCA) 프로그램

프로그램 단위(혹은 Package 단위) Clean Core extensibility framework 결정 !!!



Object in use?
No
Retire custom program
Yes
Yes
Standard functions exist?
Yes
In S/4?
No
No
Non-ERP users?
No
ERP indepe ndent?
No
Majority of Data resides in ERP?
Yes
Yes
Simple last mile case?
Yes
No
Yes
No
Yes

Side-by-Side Extensibility
Developer tools
Java, Node.js, ABAP
SAP Build Code
Low-Code/No-Code

Key-user Extensibility
SAP Fiori-based
Key User Tools

On-Stack Extensibility
ABAP Development Tools in Eclipse (ADT)

Side-by-Side Extensibility
Developer tools
Java, Node.js, ABAP
SAP Build Code
Low-Code/No-Code

INTERNAL – SAP and Partners Only

Clean Core : 공통된 고민

- Clean Core 전략과 필요성에 대해서는 대부분 동의
- 그런데 어디서부터 어떻게 시작해야 하나요? (S/4HANA Cloud 도입 전 / 후?)
- **(CCA 분석 기반) 클린 코어를 위한 Custom Program 개발은 어떻게 해야 할까요?**
- Key Takeaways

INTERNAL – SAP and Partners Only

Joule for developers(J4D)

개발 전문가들에게 생산성을 !!! 초보자들에겐 자신감을 !!!

Unified offering for AI capabilities across SAP Build solutions

SAP Build Code

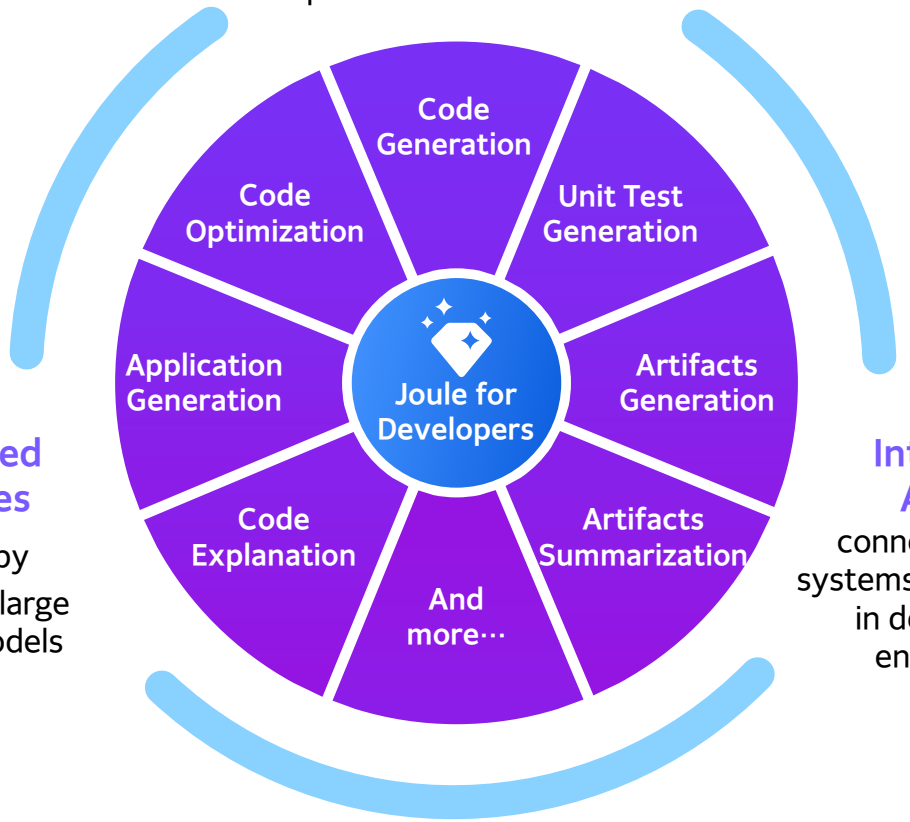
Side-by-side extension

ABAP Environment

On-stack & Side-by-side extension

Comprehensive AI capabilities

for new application development, existing code optimization and automation



Specialized outcomes

Powered by SAP-centric large language models

Integrated AI tools

connected to SAP systems and integrated in development environment

SAP Build Apps

Low-code AI-assisted development

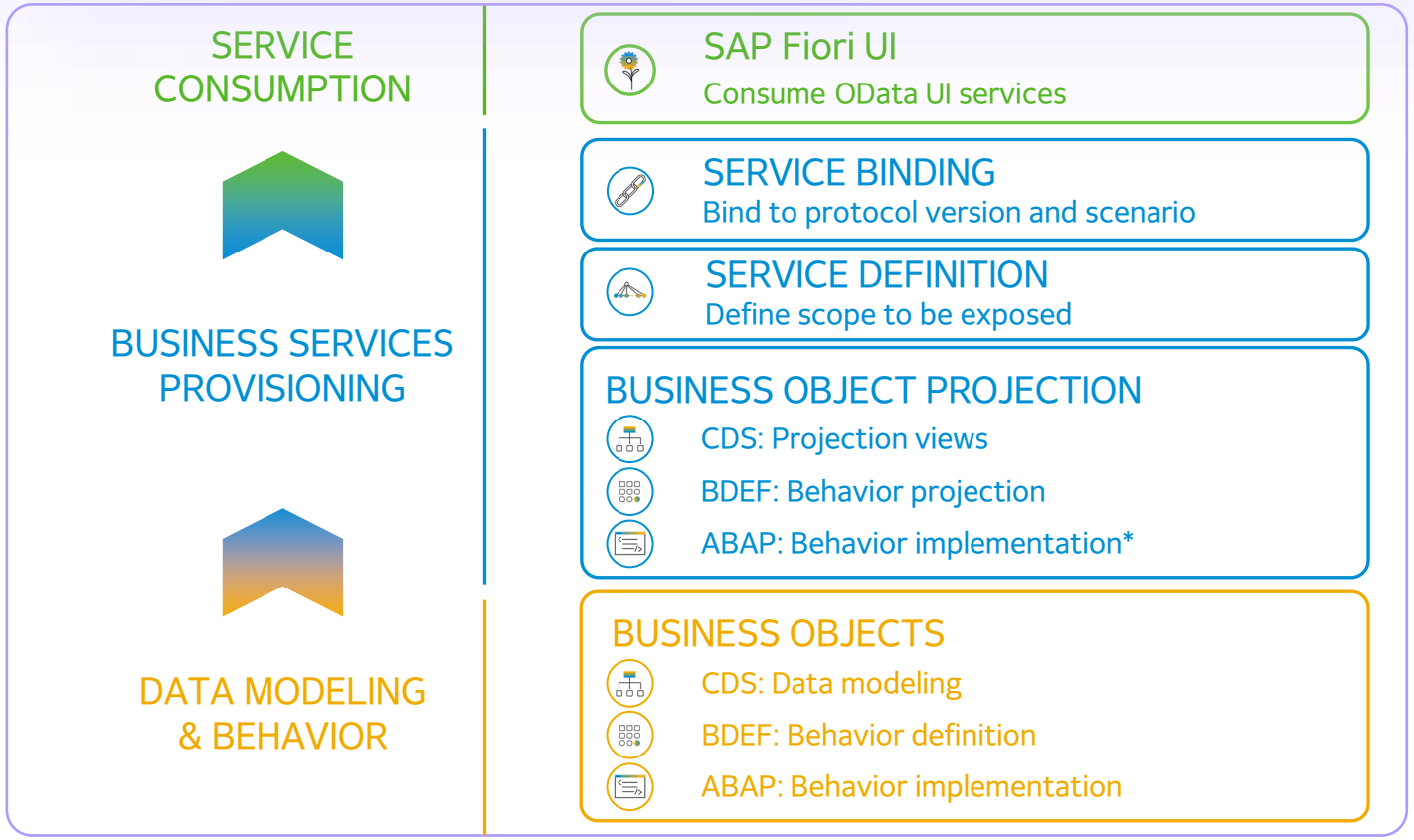
SAP Build Process Automation

Designed to accelerate automation workflows

INTERNAL – SAP and Partners Only

J4D기반 ABAP Cloud 프로그램 개발

생성형 AI 기반 Business Object & 개발 단계



J4D 활용 범위

- Data 모델링
- Behavior Definition & Implementation 활용
- Troubleshooting
- 코드 테스트
- 프롬프트 -한글과 영문 기반

Draft table 생성 (for Custom table) → Data modeling (Interface & CDS) → Behavior Definition & Implementation → Validation & Determination → Custom Action

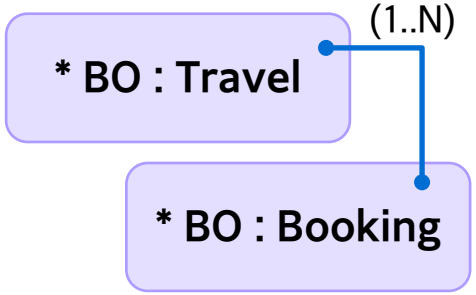
INTERNAL – SAP and Partners Only

Lesson & Learned : Learning ABAP Cloud & Adapting J4D

Joule for Developers 기반 Cloud ABAP 개발의 예상 Learning Curve



INTERNAL – SAP and Partners Only



* BO: Business Object

INTERNAL – SAP and Partners Only

Step-1 : Draft table 생성

[QJ6] ZRAP120_TRAVEL | [QJ6] ZRAP120_TRAVEL_D | [QJ6] ZRAP120_BOOK

```
@EndUserText.label : 'RAP Travel table'
@AbapCatalog.enhancement.category : #NOT_EXTENSIBLE
@AbapCatalog.tableCategory : #TRANSPARENT
@AbapCatalog.deliveryClass : #A
@AbapCatalog.dataMaintenance : #RESTRICTED
define table zrap120_travel {

  key mandt             : abap.clnt not null;
  key travel_id         : /dmo/travel_id not null;
  agency_id             : /dmo/agency_id;
  customer_id           : /dmo/customer_id;
  begin_date            : /dmo/begin_date;
  end_date              : /dmo/end_date;
  @Semantics.amount.currencyCode : 'zrap120_travel.currency_c
  booking_fee           : /dmo/booking_fee;
  @Semantics.amount.currencyCode : 'zrap120_travel.currency_c
  total_price           : /dmo/total_price;
  currency_code         : /dmo/currency_code;
  description           : /dmo/description;
  overall_status        : /dmo/overall_status;
  attachment            : /dmo/attachment;
  mime_type             : /dmo/mime_type;
  file_name             : /dmo/filename;
  last_changed_at       : abp_lastchange_tstmpl;
  created_by            : abp_creation_user;
  created_at            : abp_creation_tstmpl;
  local_last_changed_by : abp_locinst_lastchange_user;
  local_last_changed_at : abp_locinst_lastchange_tstmpl;

}
```

Travel
Draft : Travel_D

(1..N)

Booking
Draft : Booking_D

Joule

[QJ6]

You

Please create draft table for the zrap120_travel. Consider consistency and reusability.

Joule

> Used objects (1)

To create a draft table for the `ZRAP120_TRAVEL` table with a focus on consistency and reusability, you'll want to design it in a way that adheres to RAP principles and includes consistent draft-handling functionalities. Here's how you can define this draft table:

Draft Table Definition for `ZRAP120_TRAVEL`

ABAP — Insert | Copy

```
@EndUserText.label: 'RAP Travel Draft Table'
@AbapCatalog.enhancement.category: #NOT_EXTENSIBLE
@AbapCatalog.tableCategory: #TRANSPARENT
@AbapCatalog.deliveryClass: #A
@AbapCatalog.dataMaintenance: #RESTRICTED
define table zrap120_travel_draft {

  key mandt                 : abap.clnt not null;
  key travel_id             : /dmo/travel_id not null;
  key draftuuid             : sdraft_uuid not null;

  agency_id                 : /dmo/agency_id;
  customer_id               : /dmo/customer_id;
  begin_date                : /dmo/begin_date;
  end_date                  : /dmo/end_date;
  @Semantics.amount.currencyCode : 'zrap120_travel_draft.currency_code'
  booking_fee               : /dmo/booking_fee;
  @Semantics.amount.currencyCode : 'zrap120_travel_draft.currency_code'
  total_price               : /dmo/total_price;
  currency_code             : /dmo/currency_code;
  description               : /dmo/description;
  overall_status            : /dmo/overall_status;
  attachment                : /dmo/attachment;
  mime_type                 : /dmo/mime_type;
  filename                  : /dmo/filename;
  lastchanged_at            : abp_lastchange_tstmpl;
  created_by                : abp_creation_user;
  created_at                : abp_creation_tstmpl;
  local_last_changed_by     : abp_locinst_lastchange_user;
  local_last_changed_at     : abp_locinst_lastchange_tstmpl;

  "%admin" : include sych_bdl_draft_admin_inc;
}
```

Prompt : Draft 테이블 생성, Consistency & Reusability 고려해서 생성

key draftuuid : sdraft_uuid **not null;**

"%admin" **: include** sych_bdl_draft_admin_inc;

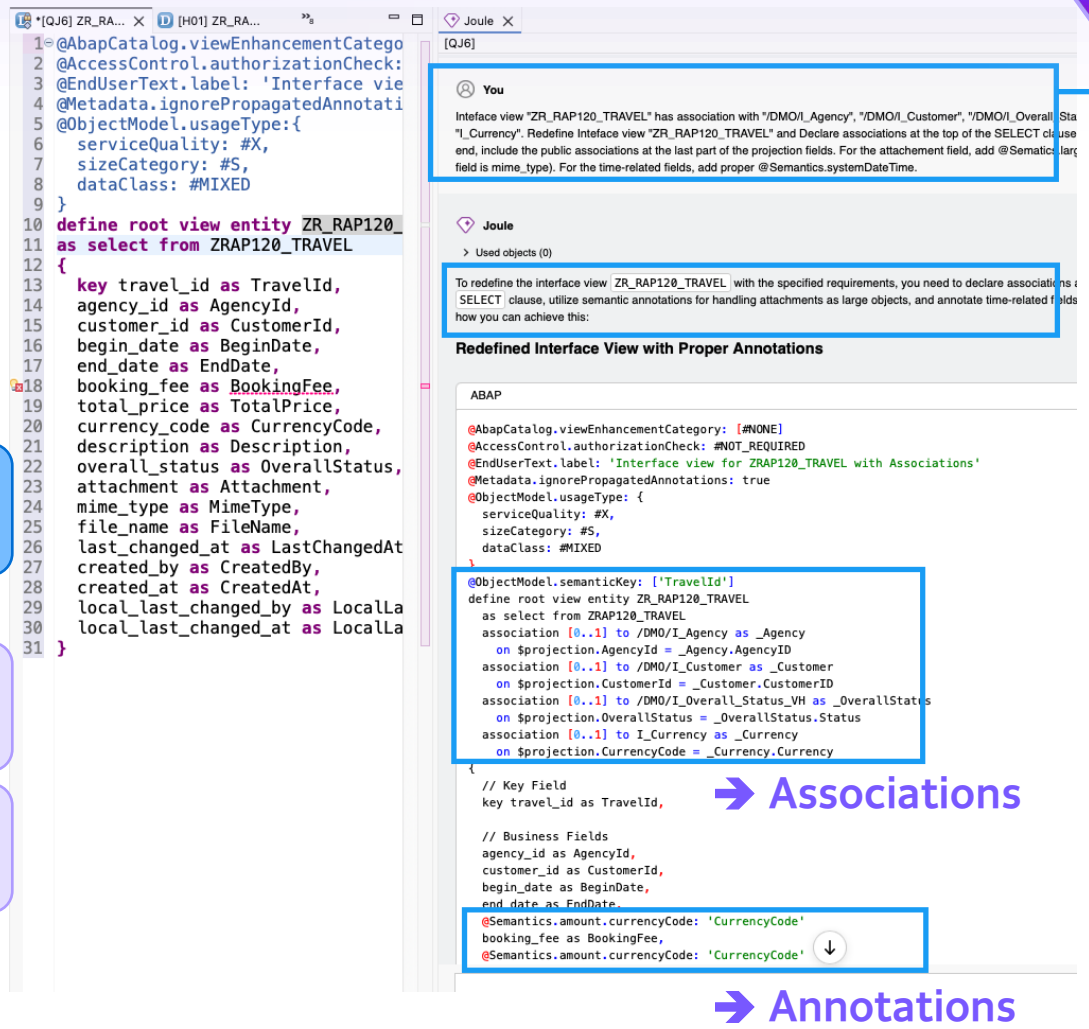
Step-2 : Data modeling (Travel)

Tx Interface CDS 생성 (영문 프롬프트)

Data modeling
Tx Interface CDS

Travel, Travel_D

Booking Booking_D




```
@AbapCatalog.viewEnhancementCatego
@AccessControl.authorizationCheck:
@EndUserText.label: 'Interface vie
@Metadata.ignorePropagatedAnnotati
@ObjectModel.usageType:{
  serviceQuality: #X,
  sizeCategory: #S,
  dataClass: #MIXED
}
define root view entity ZR_RAP120_
as select from ZRAP120_TRAVEL
{
  key travel_id as TravelId,
  agency_id as AgencyId,
  customer_id as CustomerId,
  begin_date as BeginDate,
  end_date as EndDate,
  booking_fee as BookingFee,
  total_price as TotalPrice,
  currency_code as CurrencyCode,
  description as Description,
  overall_status as OverallStatus,
  attachment as Attachment,
  mime_type as MimeType,
  file_name as FileName,
  last_changed_at as LastChangedAt
  created_by as CreatedBy,
  created_at as CreatedAt,
  local_last_changed_by as LocalLa
  local_last_changed_at as LocalLa
}
```

영문 Prompt : (다음장 ➔ 한글)

- Interface view **"ZR_RAP120_TRAVEL"** has association with **"/DMO/I_Agency", "/DMO/I_Customer", "/DMO/I_Overall_Status_VH" and "I_Currency".**
- Redefine Interface view "ZR_RAP120_TRAVEL" and Declare **associations** at the top of the SELECT clause in a CDS view.
- At the end, include the public associations at the last part of the projection fields.
- For the attachment field, add @Sematics.largeObject(mime_type field is mime_type).
- For the time-related fields, add proper @Semantics.systemDateTime.

INTERNAL – SAP and Partners Only

Step-2 : Data modeling (Travel)

Tx Interface CDS 생성 (한글 프롬프트)

Data modeling
Tx Interface CDS

Travel, Travel_D

Booking Booking_D



한글 Prompt :

- **인터페이스 뷰 "ZR_RAP120_TRAVEL"은 "/DMO/I_Agency", "/DMO/I_Customer", "/DMO/I_Overall_Status_VH", "I_Currency"와의 Association을 가지고 있습니다.**
  CDS 뷰에서 SELECT 절의 상단에 Association을 선언하도록 인터페이스 뷰 "ZR_RAP120_TRAVEL"을 재정의하세요.
- 마지막에는 public association들을 projection 필드의 마지막 부분에 포함시키세요.
- 첨부(attachment) 필드에는 @Semantics.largeObject (mime_type 필드는 mime_type) 어노테이션을 추가하세요.
- 시간 관련 필드에는 적절한 @Semantics.systemDateTime 어노테이션을 추가하세요.

INTERNAL – SAP and Partners Only

Step-2 : Data modeling (Travel)

Projection CDS 생성

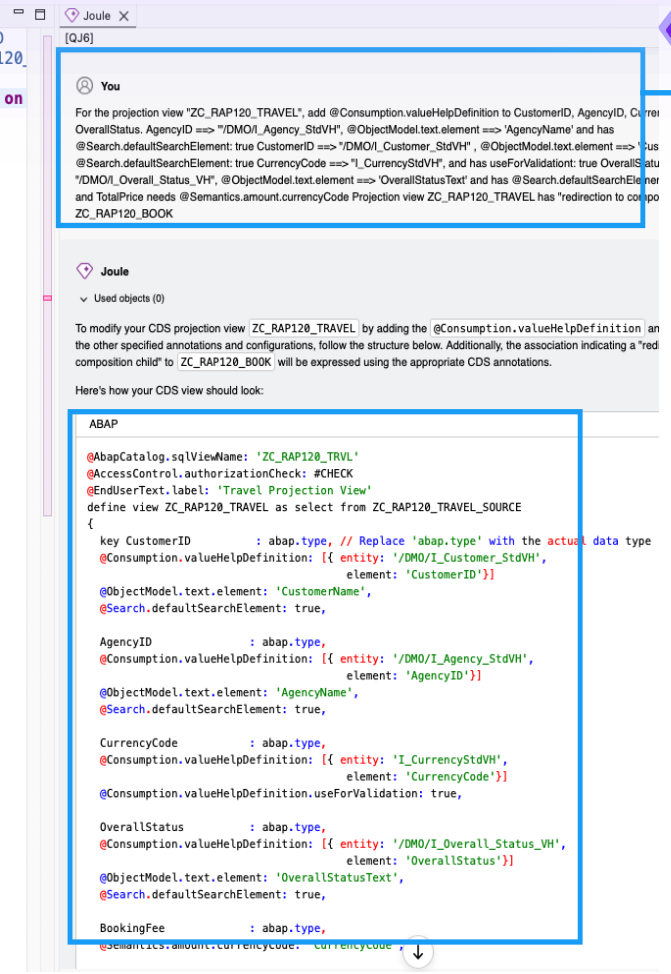
```
@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'Projection view for ZR_RAP120_
@Metadata.ignorePropagatedAnnotations: true
define view entity ZC_RAP120_BOOK as projection on
{
  key TravelID,
  key BookingID,
  BookingDate,
  CustomerID,
  CarrierID,
  ConnectionID,
  FlightDate,
  BookingStatus,
  FlightPrice,
  CurrencyCode,
  LocalLastChangedAt,
  /* Associations */
  _BookingStatus,
  _Carrier,
  _Connection,
  _Currency,
  _Customer,
```

Data modeling Projection CDS

Data modeling Tx Interface CDS

Travel, Travel_D

Booking Booking_D



Prompt :

- For the projection view "ZC_RAP120_TRAVEL", add @Consumption.valueHelpDefinition to CustomerID, AgencyID, CurrencyCode and OverallStatus.
- AgencyID ==> '"/DMO/I_Agency_StdVH", @ObjectModel.text.element ==> 'AgencyName' and has @Search.defaultSearchElement: true
- CustomerID ==> "/DMO/I_Customer_StdVH" , @ObjectModel.text.element ==> 'CustomerName' and has @Search.defaultSearchElement: true
- CurrencyCode ==> "I_CurrencyStdVH", and has useForValidationt: true
- OverallStatus ==> "/DMO/I_Overall_Status_VH", @ObjectModel.text.element ==> 'OverallStatusText' and has @Search.defaultSearchElement: true
- BookingFee and TotalPrice needs
- @Semantics.amount.currencyCode
- Projection view ZC_RAP120_TRAVEL has "redirection to composition child" to ZC_RAP120_BOOK

INTERNAL – SAP and Partners Only

Step-3 : Behavior Definition

Travel Behavior 생성 (영문 프롬프트)

A / V / D / S*

Behavior Def.
Behavior Imp.

Data modeling
Projection CDS

Data modeling
Tx Interface CDS

Travel, Travel_D

Booking Booking_D

```
1  managed implementation in class zbp_r_rap120_tr
2  strict ( 2 );
3
4  define behavior for ZR_RAP120_TRAVEL //alias <a
5  persistent table zrap120_travel
6  lock master
7  authorization master ( instance )
8  //etag master <field_name>
9  {
10   create ( authorization : global );
11   update;
12   delete;
13     // Read-only fields
14   field ( readonly )
15     LocalLastChangedAt,
16     LastChangedAt,
17     TravelID,
18     TotalPrice;
19
20   // Mandatory user input values
21   field ( mandatory )
22     BeginDate,
23     EndDate;
24
25
26
27   field ( readonly ) TravelID;
28   association _Booking { create; }
29 }
30
31 define behavior for ZR_RAP120_BOOK //alias <ali
32 persistent table zrap120_book
33 lock dependent by _Travel
34 authorization dependent by _Travel
35 //etag master <field_name>
36 {
37   update;
38   delete;
39   field ( readonly ) TravelID, BookingID;
40   association _Travel;
41 }
```

[QJ6] ZC_RA... | [QJ6] ZR_RA... | [H01] ZRAP11...

Joule

[QJ6]

You

For the behavior definition "ZR_RAP120_TRAVEL", I would like to indicate contents as followings. (you make sur you're the RAP development expert) readonly fields ==> LocalLastChangedAt, LastChangedAt, TravelID, TotalPrice mandatory user input value ==> BeginDate, EndDate Side effects ==> BookingFee attects TotalPrice. CurrencyCode affects TotalPrice□Side effects example □□side effects { field A affects field B; }

Joule

› Used objects (0)

To incorporate your requirements into the behavior definition for `ZR_RAP120_TRAVEL`, you should specify read-only fields, mandatory user input fields, and side effects. Below is a definition demonstrating these elements:

ABAP — Insert Copy

```
managed implementation in class zbp_r_rap120_travel unique;
strict ( 2 );

define behavior for ZR_RAP120_TRAVEL
persistent table zrap120_travel
lock master
authorization master ( instance )
//etag master <field_name>
{
  create ( authorization : global );
  update;
  delete;

  // Define read-only fields
  field ( readonly )
    LocalLastChangedAt,
    LastChangedAt,
    TravelID,
    TotalPrice;

  // Specify mandatory user input values
  field ( mandatory )
    BeginDate,
    EndDate;

  // Define side effects
  sideeffects
  {
    field BookingFee affects TotalPrice;
    field CurrencyCode affects TotalPrice;
  }

  // Association to _Booking with creation capability
  association _Booking { create; }
}
```

Prompt : (다음장 ➔ 한글)

- For the behavior definition "ZR_RAP120_TRAVEL", I would like to indicate contents as followings. (you make sure you're the RAP development expert)
- readonly fields ==> LocalLastChangedAt, LastChangedAt, TravelID, TotalPrice
- mandatory user input value ==> BeginDate, EndDate
- Side effects ==> BookingFee attects TotalPrice. CurrencyCode affects TotalPrice
  Side effects example ==>
  side effects
  {
  field A affects field B;
  }

* A : Actions, V: Validation, D: Determination, S: additional Save logic

Step-3 : Behavior Definition

Travel Behavior 생성 (한글)

A / V / D / S*

Behavior Def.
Behavior Imp.

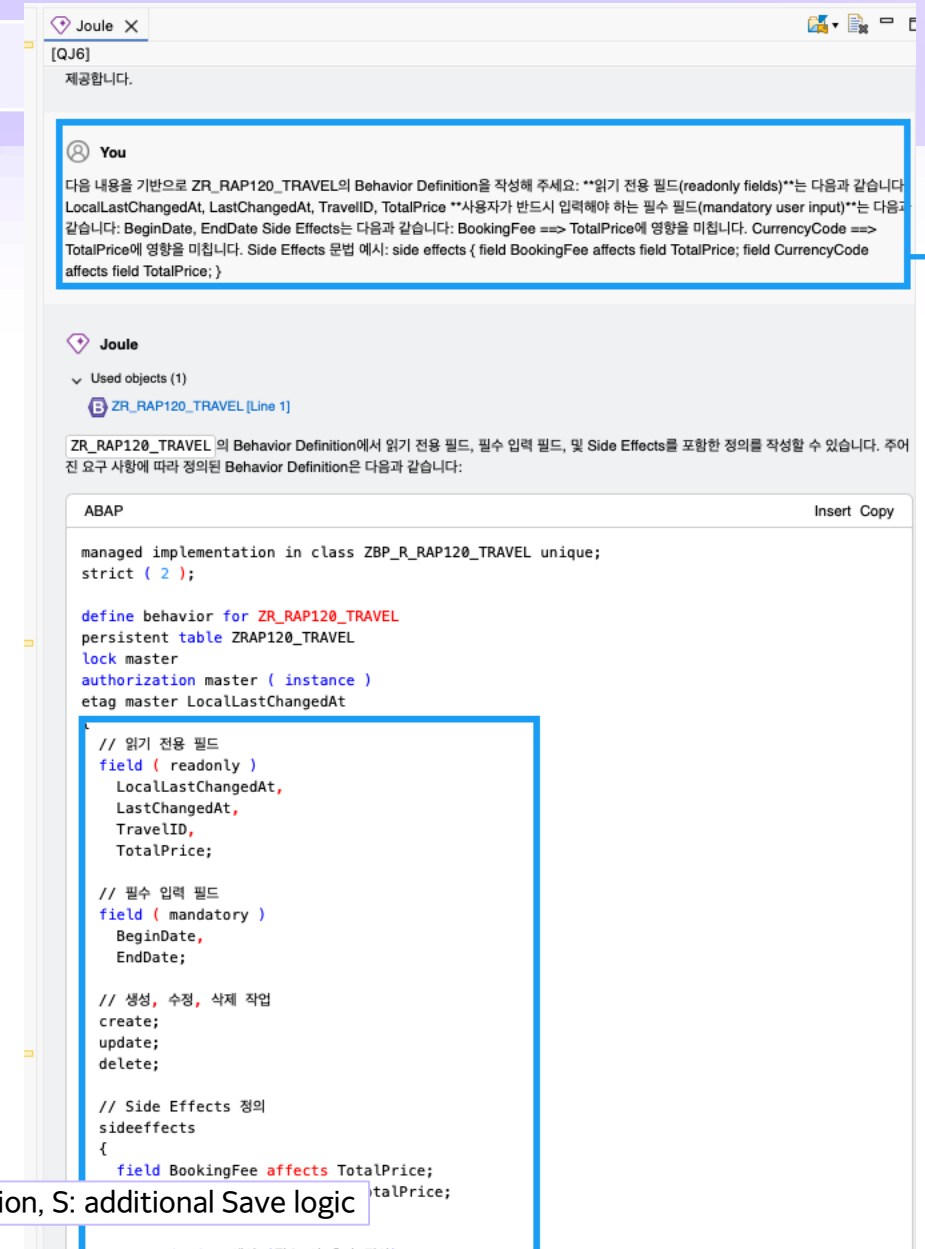
Data modeling
Projection CDS

Data modeling
Tx Interface CDS

Travel, Travel_D

Booking Booking_D

* A : Actions, V: Validation, D: Determination, S: additional Save logic



Prompt : (한글 프롬프트)

- 다음 내용을 기반으로 ZR_RAP120_TRAVEL의 Behavior Definition을 작성해 주세요
- 읽기 전용 필드(readonly fields)는 다음과 같습니다 ==>
- LocalLastChangedAt, LastChangedAt, TravelID, TotalPrice
- 사용자가 반드시 입력해야 하는 필수 필드(mandatory user input)는 다음과 같습니다 ==> BeginDate, EndDate
- Side Effects는 다음과 같습니다:
- BookingFee ==> TotalPrice에 영향을 미칩니다. CurrencyCode ==> TotalPrice에 영향을 미칩니다.
- Side Effects 문법 예시 ==> side effects { field BookingFee affects field TotalPrice; field CurrencyCode affects field TotalPrice; }

Step-3 : Behavior Definition

Travel Behavior 생성 (영문 프롬프트)

A / V / D / S*

Behavior Def.
Behavior Imp.

Data modeling
Projection CDS

Data modeling
Tx Interface CDS

Travel, Travel_D

Booking Booking_D

```
1  managed implementation in class zbp_r_rap120_tr
2  strict ( 2 );
3
4  define behavior for ZR_RAP120_TRAVEL //alias <a
5  persistent table zrap120_travel
6  lock master
7  authorization master ( instance )
8  //etag master <field_name>
9  {
10   create ( authorization : global );
11   update;
12   delete;
13     // Read-only fields
14   field ( readonly )
15     LocalLastChangedAt,
16     LastChangedAt,
17     TravelID,
18     TotalPrice;
19
20   // Mandatory user input values
21   field ( mandatory )
22     BeginDate,
23     EndDate;
24
25
26
27   field ( readonly ) TravelID;
28   association _Booking { create; }
29 }
30
31 define behavior for ZR_RAP120_BOOK //alias <ali
32 persistent table zrap120_book
33 lock dependent by _Travel
34 authorization dependent by _Travel
35 //etag master <field_name>
36 {
37   update;
38   delete;
39   field ( readonly ) TravelID, BookingID;
40   association _Travel;
41 }
```

Joule

[QJ6]

You

For the behavior definition "ZR_RAP120_TRAVEL", I would like to indicate contents as followings. (you make sure you're the RAP development expert) Validation "validateDate" ==> during creation, BeginDate and EndDate would be validated. Determination "calculateTotalPrice" ==> during modification of field BookingFee, CurrencyCode, logics will be added.

Joule

› Used objects (1)

To include validation and determination logic in your behavior definition for `ZR_RAP120_TRAVEL`, you need to specify these elements under the `define behavior` statement. Here's how you can incorporate the `validateDate` validation and the `calculateTotalPrice` determination:

ABAP — Insert Copy

```
managed implementation in class zbp_r_rap120_travel unique;
strict ( 2 );

define behavior for ZR_RAP120_TRAVEL
persistent table zrap120_travel
lock master
authorization master ( instance )
//etag master <field_name>
{
  create ( authorization : global, validation : validateDate );
  update;
  delete;

  // Read-only fields
  field ( readonly )
    LocalLastChangedAt,
    LastChangedAt,
    TravelID,
    TotalPrice;

  // Mandatory user input fields
  field ( mandatory )
    BeginDate,
    EndDate;

  // Side effects
  sideeffects
  {
    field BookingFee affects TotalPrice;
    field CurrencyCode affects TotalPrice;
  }

  determination calculateTotalPrice
    on modify { field BookingFee; field CurrencyCode; }

  validation validateDate
    on create { field BeginDate; field EndDate; }

  // Association to _Booking with creation capability
  association _Booking { create; }
}

define behavior for ZR_RAP120_BOOK
persistent table zrap120_book
```

Prompt : (다음장 ➔ 한글)

- For the behavior definition "ZR_RAP120_TRAVEL", I would like to indicate contents as followings. (you make sure you're the RAP development expert)
- Validation "validateDate" ==> during creation, BeginDate and EndDate would be validated.
- Determination "calculateTotalPrice" ==> during modification of field BookingFee, CurrencyCode, logics will be added.

* A : Actions, V: Validation, D: Determination, S: additional Save logic

Step-3 : Behavior Definition

Travel Behavior 생성 (한글)

A / V / D / S*

Behavior Def.
Behavior Imp.

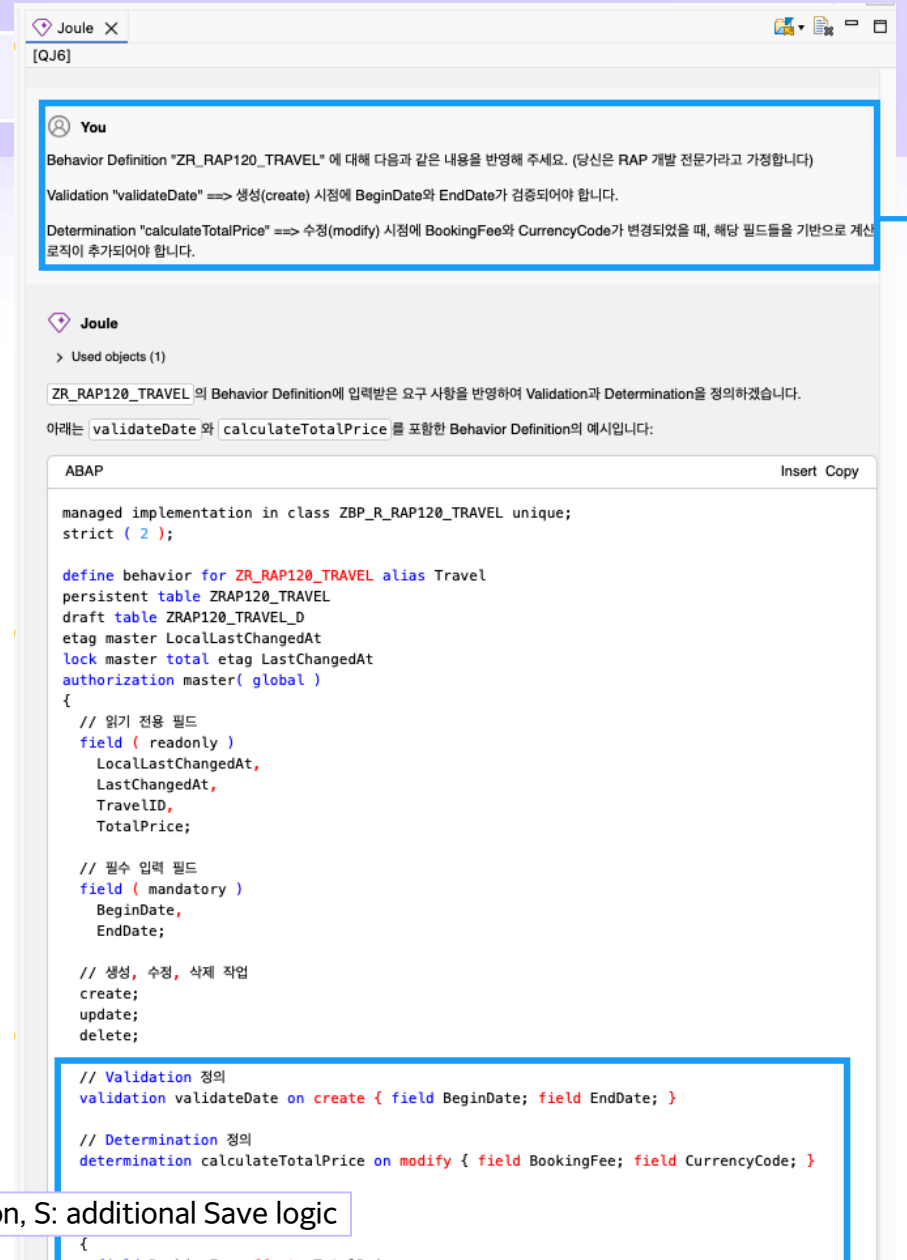
Data modeling
Projection CDS

Data modeling
Tx Interface CDS

Travel, Travel_D

Booking Booking_D

* A : Actions, V: Validation, D: Determination, S: additional Save logic

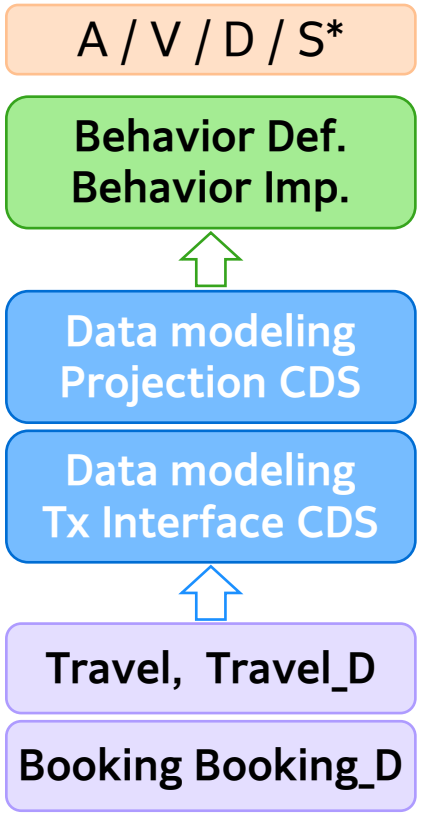


Prompt : (한글 프롬프트)

- Behavior Definition "ZR_RAP120_TRAVEL" 에 대해 다음과 같은 내용을 반영해 주세요. (당신은 RAP 개발 전문가라고 가정합니다)
- Validation "validateDate" ==> 생성(create) 시점에 BeginDate와 EndDate가 검증되어야 합니다.
- Determination "calculateTotalPrice" ==> 수정(modify) 시점에 BookingFee와 CurrencyCode가 변경되었을 때, 해당 필드들을 기반으로 계산 로직이 추가되어야 합니다.

Step-4 : Behavior Implementatio

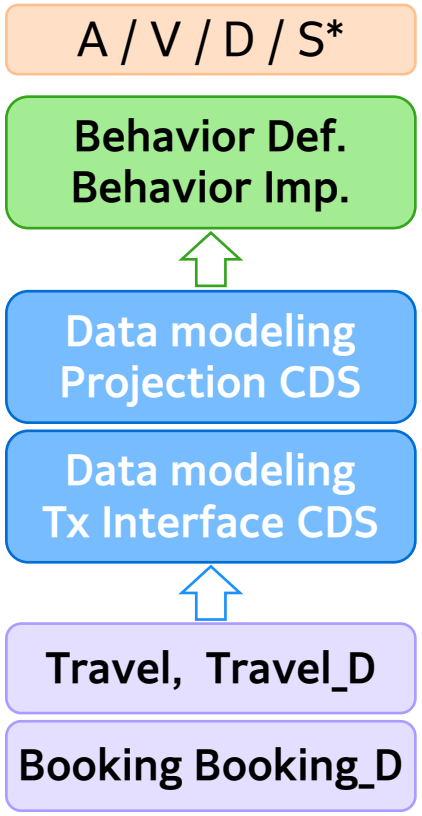
Validation : 영문 주석



* A : Actions, V: Validation, D: Determination, S: additional Save logic

Step-4 : Behavior Implementatio

Validation : 한글 주석
(+ 사전 정의된 Message Class활용)



* A : Actions, V: Validation, D: Determination, S: additional Save logic

Step-4 : Behavior Implementatio

Determination :
한글 주석. Std BO활용

A / V / D / S*

Behavior Def.
Behavior Imp.

Data modeling
Projection CDS

Data modeling
Tx Interface CDS

Travel, Travel_D

Booking Booking_D

Prompt : (한글 프롬프트)

- agencyID를 사용하여, I_BusinessPartner에서 거래처 이름(Business Partner Name)을 찾고, 해당 값을 Description에 설정 하세요.

* A : Actions, V: Validation, D: Determination, S: additional Save logic

```
124 *
125 *  " Collect all updated reported data
126 *  reported = CORRESPONDING #( DEEP update_reported ).
127 *
128 *ENDMETHOD.
129
130 METHOD setInitialDescription.
131 "agencyID를 사용하여, I_BusinessPartner에서 거래처 이름(Business Partner Name)을 찾고, 해당 값을 Description에 설정 하십시요.
132
133
134 ENDMETHOD.
135
136
137 METHOD changeStatus.
138   TRY.
139
140
141       " Read current status
142       READ ENTITIES OF ZR_TRAVEL_010 IN LOCAL MODE
143         ENTITY ZrTravel010
144           FIELDS ( Status )
145           WITH CORRESPONDING #( keys )
146         RESULT DATA(lt_travel).
147
148       " Update the status
149       MODIFY ENTITIES OF ZR_TRAVEL_010 IN LOCAL MODE
150         ENTITY ZrTravel010
151           UPDATE
152             FIELDS ( Status )
153             WITH VALUE #( FOR travel IN lt_travel ( %tky   = travel-%tky
154                                                     Status = 'P' ) )
155         REPORTED DATA(update_reported).
156
157       " Ensure the reported data is appropriately handled
158       reported = CORRESPONDING #( DEEP update_reported ).
159
160     CATCH cx_root INTO DATA(error).
161       " Capture exception text for logging or display
162       DATA(lv_error_text) = error->get_text( ).
163
164       " Log the error using transaction SLG1, application log, or custom logging mechanism
```

Lesson & Learned : Learning ABAP Cloud & Adapting J4D

Joule for Developers 기반 Cloud ABAP 개발의 예상 Learning Curve

Learning ABAP Cloud

(4-6 weeks)

⇨

Adapting J4D (Prompt Engineering)

(1-2 weeks)

⇨

Clean Core ABAP Cloud development based on J4D

(1-2 day / App)

- J4D로 부터 90%정도 완성된 소스를 제공 받을 수 있으며, 10%정도는 재질문 통해 수정 해야 함.
  (재질문 요령 : 직접 적인 에러메시지 제공, 대안 방법 제시 요청, 예제 syntax 제공 시 더 좋은 답변)
- 해당 프로젝트 상에서 생성된 소스들을 참조해서 코드 생성을 해 주지만, 다른 프로젝트의 커스텀 소스는 약간의 조정이 요구됨.(이럴 땐, 예제 소스 제공 시 더 좋은 답변)
- Predictive code completion의 경우, 단계별 코멘트(주석) 통해 소스 자동 완성. 단계별 코멘트는 가급적 블럭 단위로 구성.(조건문(If..then..else) , 반복문(looping) 등 논리적 흐름이 명확한 블럭 단위로 주석 작성)

INTERNAL – SAP and Partners Only

Joule for developers(J4D)

개발 전문가들에게 생산성을 !!! 초보자들에겐 자신감을 !!!

Unified offering for AI capabilities across SAP Build solutions

SAP Build Code

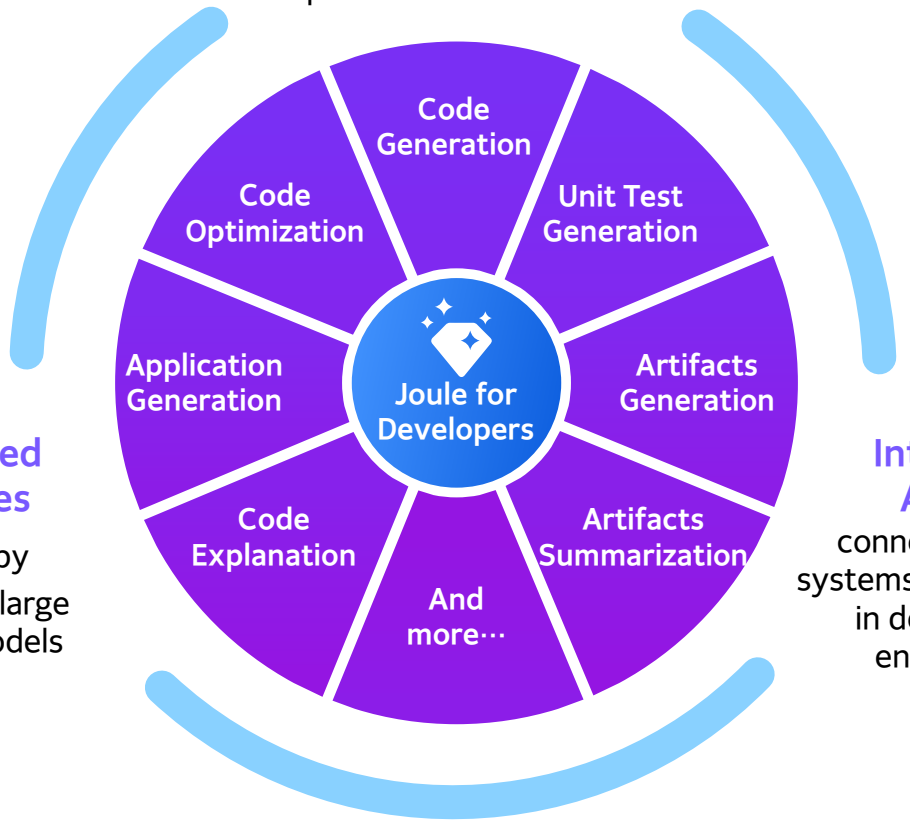
Side-by-side extension

ABAP Environment

On-stack & Side-by-side extension

Comprehensive AI capabilities

for new application development, existing code optimization and automation



Specialized outcomes

Powered by SAP-centric large language models

Integrated AI tools

connected to SAP systems and integrated in development environment

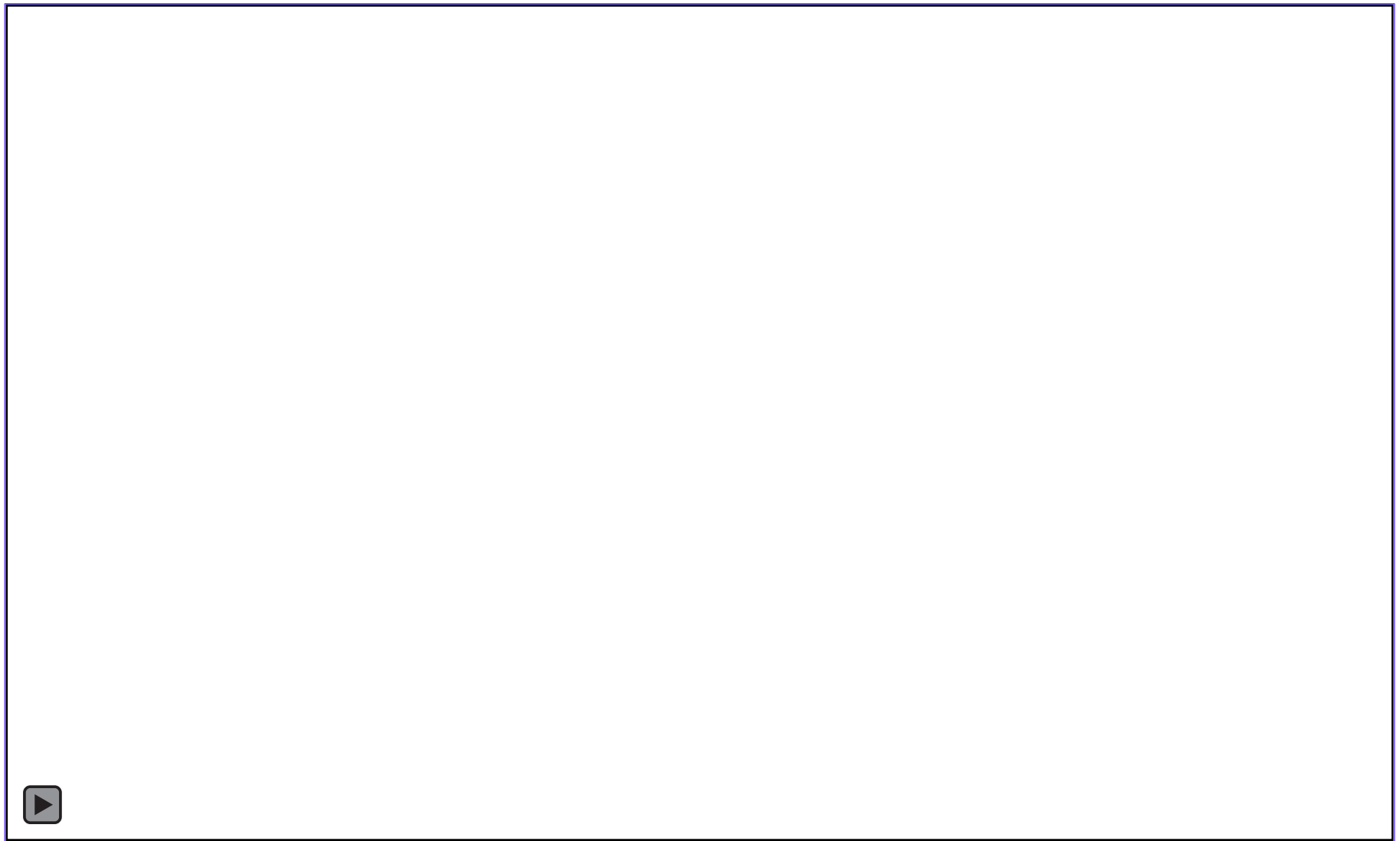
SAP Build Apps

Low-code AI-assisted development

SAP Build Process Automation

Designed to accelerate automation workflows

SAP Build Code로 개발된 어플리케이션 예시



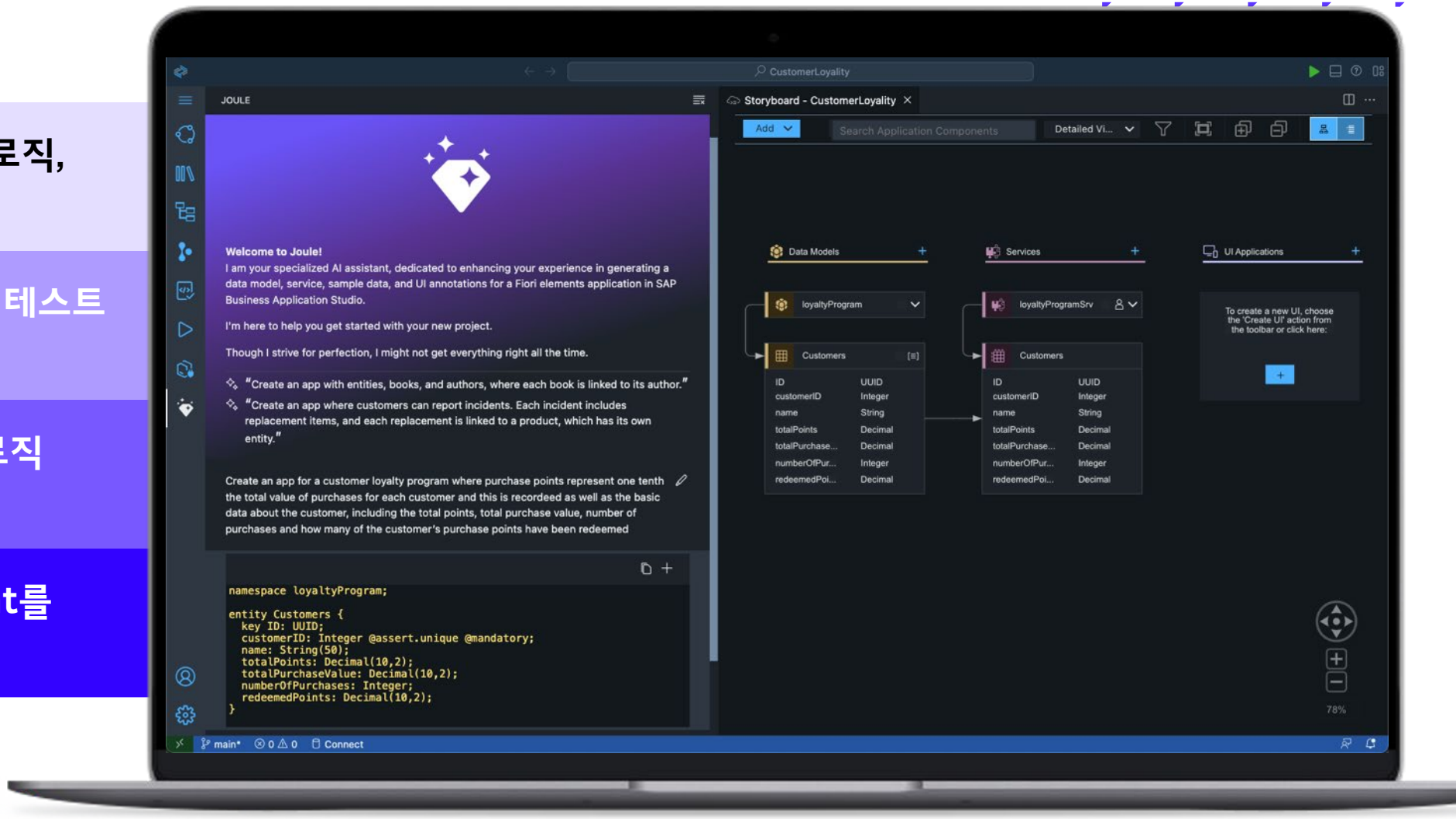
오픈소스 개발 - SAP Build Code를 활용한 어플리케이션 개발

AI co-pilot Joule을 통해 앱 로직, 데이터 모델 자동 생성

샘플 데이터 자동 생성 및 유닛 테스트 제공

프롬프트 기반 어플리케이션 로직 코드의 생성 및 변경

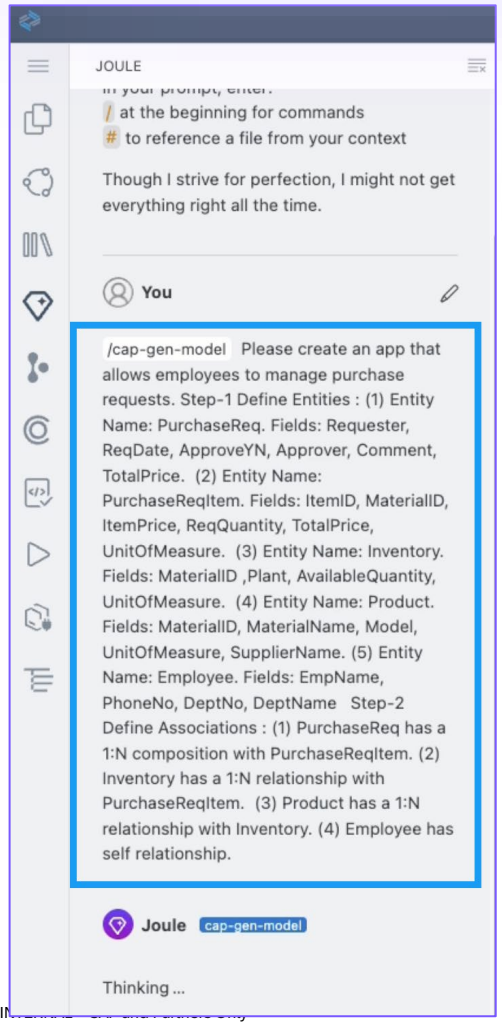
SAPUI5와 SAP Fiori element를 활용한 Fiori web app 개발



INTERNAL – SAP and Partners Only

오픈소스 개발 - SAP Build Code를 활용한 어플리케이션 개발

비지니스 (데이타) 모델 생성



1) BTP상에 Backend logic 개발(CAP 프로그래밍)
- Data modeling□ □ □ prompt

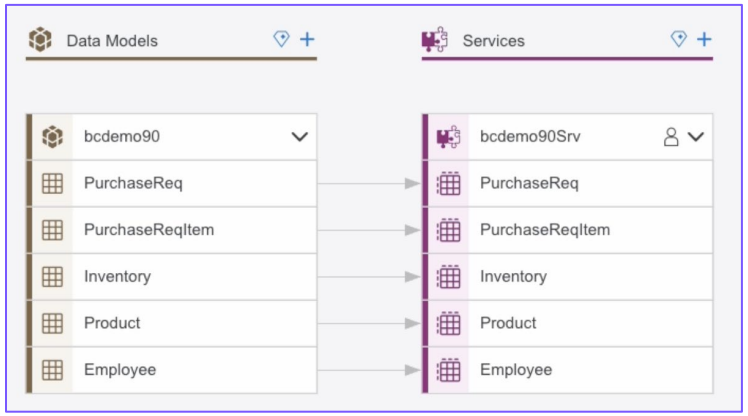
Prompt : (다음장 ➔ 한글)

- Please create an app that allows employees to manage purchase requests.
- Step-1 Define Entities :
- (1) Entity Name: PurchaseReq. Fields: Requester, ReqDate, ApproveYN, Approver, Comment, TotalPrice.
- (2) Entity Name: PurchaseReqItem. Fields: ItemID, MaterialID, ItemPrice, ReqQuantity, TotalPrice, UnitOfMeasure.
- (3) Entity Name: Inventory. Fields: MaterialID ,Plant, AvailableQuantity, UnitOfMeasure.
- (4) Entity Name: Product. Fields: MaterialID, MaterialName, Model, UnitOfMeasure, SupplierName.
- (5) Entity Name: Employee. Fields: EmpName, PhoneNo, DeptNo, DeptName

- Step-2 Define Associations :
- (1) PurchaseReq has a 1:N composition with PurchaseReqItem.
- (2) Inventory has a 1:N relationship with PurchaseReqItem.
- (3) Product has a 1:N relationship with Inventory.
- (4) Employee has self relationship.

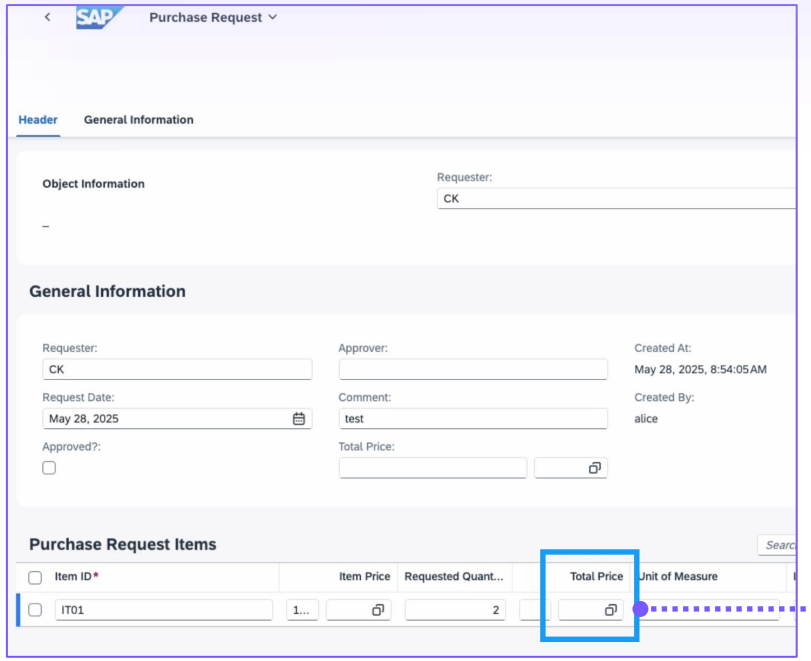
- Finally, please create 20 sample data.

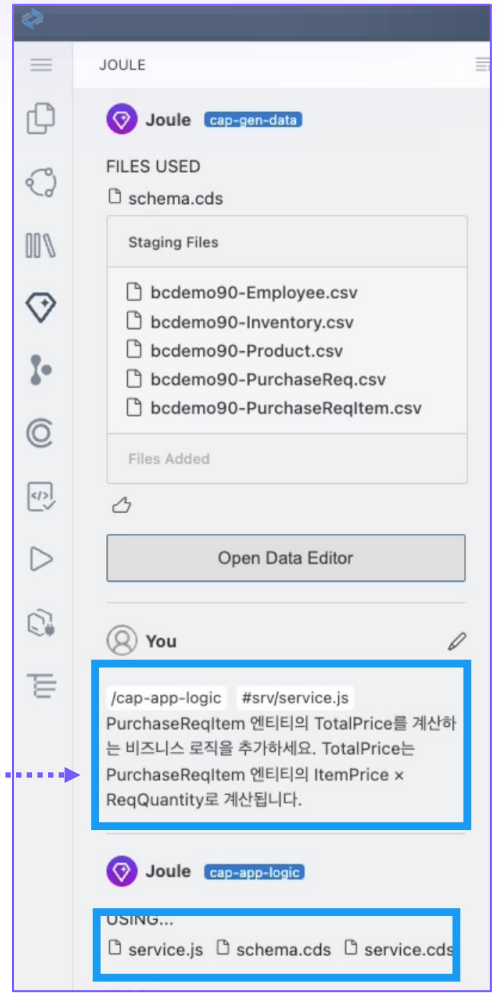
2) Design time artifacts 확인.

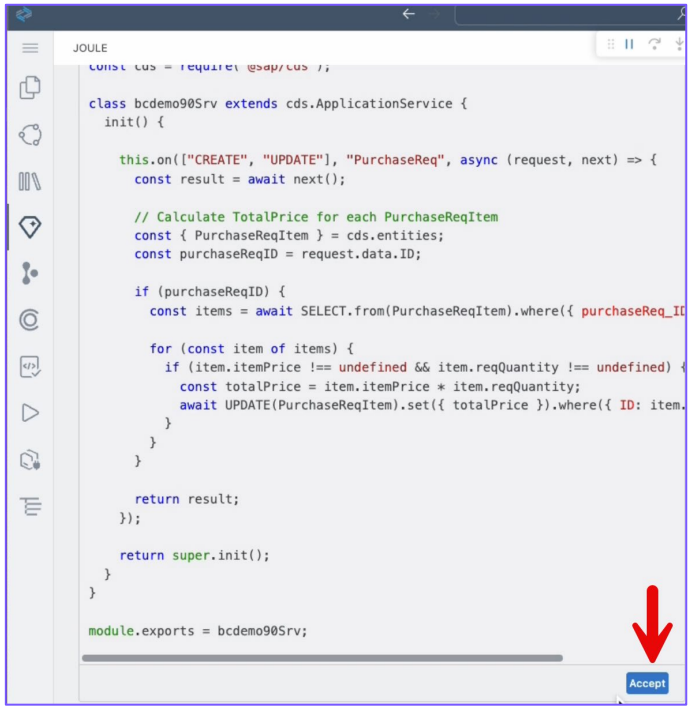


오픈소스 개발 - SAP Build Code를 활용한 어플리케이션 개발

비지니스 Logic 추가

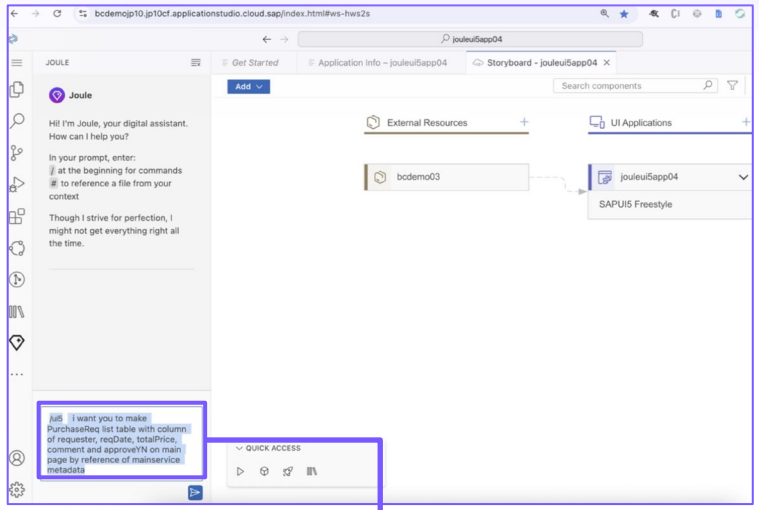






오픈소스 개발 - SAP Build Code를 활용한 어플리케이션 개발

UI 개발



Prompt 1 :

- i want you to make PurchaseReq list table with column of requester, reqDate, totalPrice, comment and approveYN on main page by reference of mainservice metadata

메인 테이블 화면 구성

port8080-workspaces-ws-hws2s.jp10.applicationstudio.cloud.sap/index.html

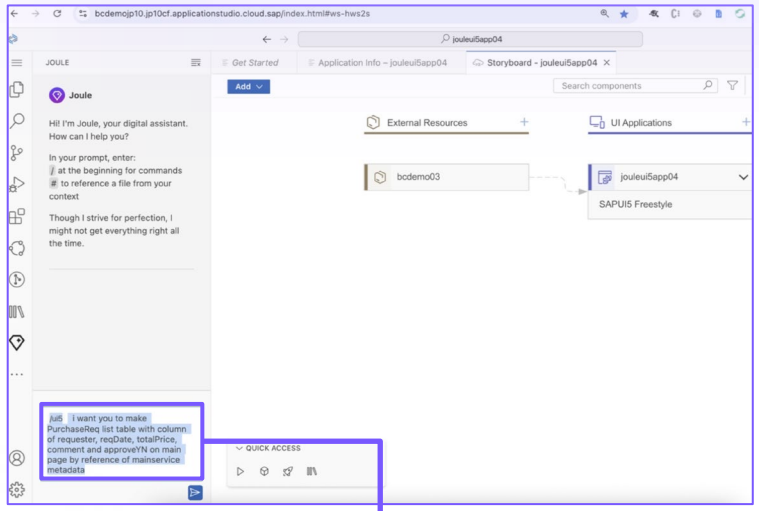
Joule based UI App04 Title

Requester	Request Date	Total Price	Comment	Approved
test	May 27, 2025	1,000.00	test11	Yes
test	May 27, 2025		admin	No
bbbbbbbbbbbb	May 27, 2025	0.00	test	Yes
test_002	May 28, 2025		test	Yes
SONG	Dec 28, 2025		16:08	Yes
nnnnn	May 27, 2025	1,000.00		Yes
임단비	May 12, 2026		111	Yes
bh001	May 27, 2025	1,200.00		No
ababab	May 29, 2025			No

INTERNAL – SAP and Partners Only

오픈소스 개발 - SAP Build Code를 활용한 어플리케이션 개발

UI 개발



Prompt 2 :

- can you change column name to Korean and bind it in i18n?

Prompt 3 :

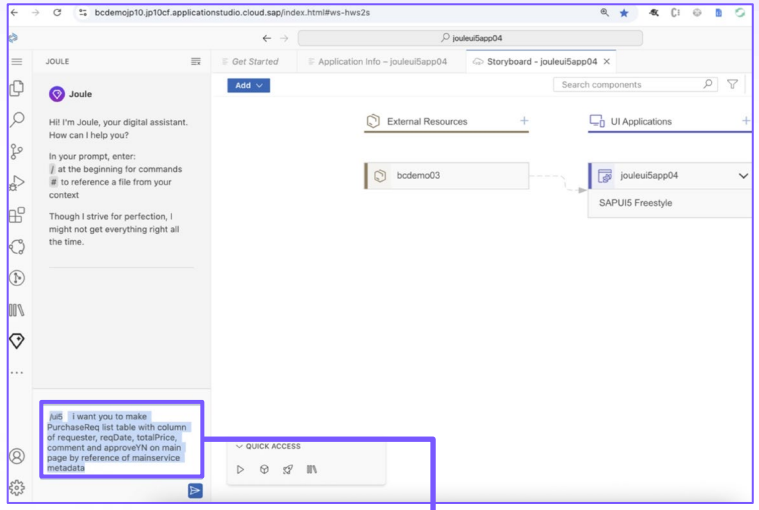
- i want filterbar to seach main table using as following xmlns.
- xmlns:comp="sap.ui.comp.filterbar" xmlns:core="sap.ui.core"

필터 바 & 테이블 헤더를 한글로



오픈소스 개발 - SAP Build Code를 활용한 어플리케이션 개발

UI 개발

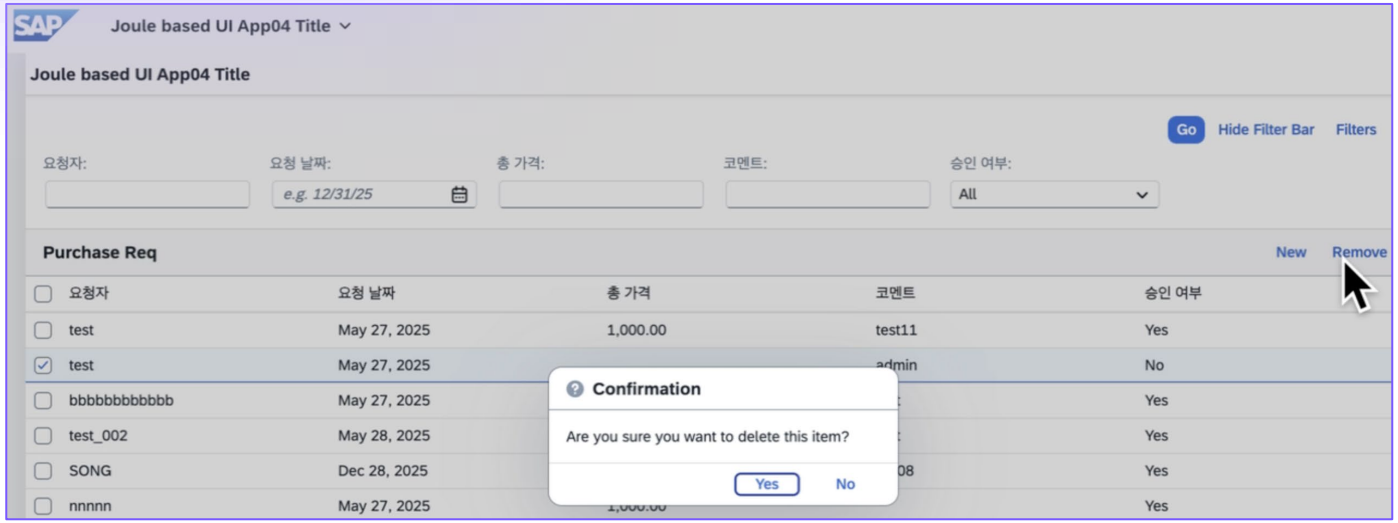


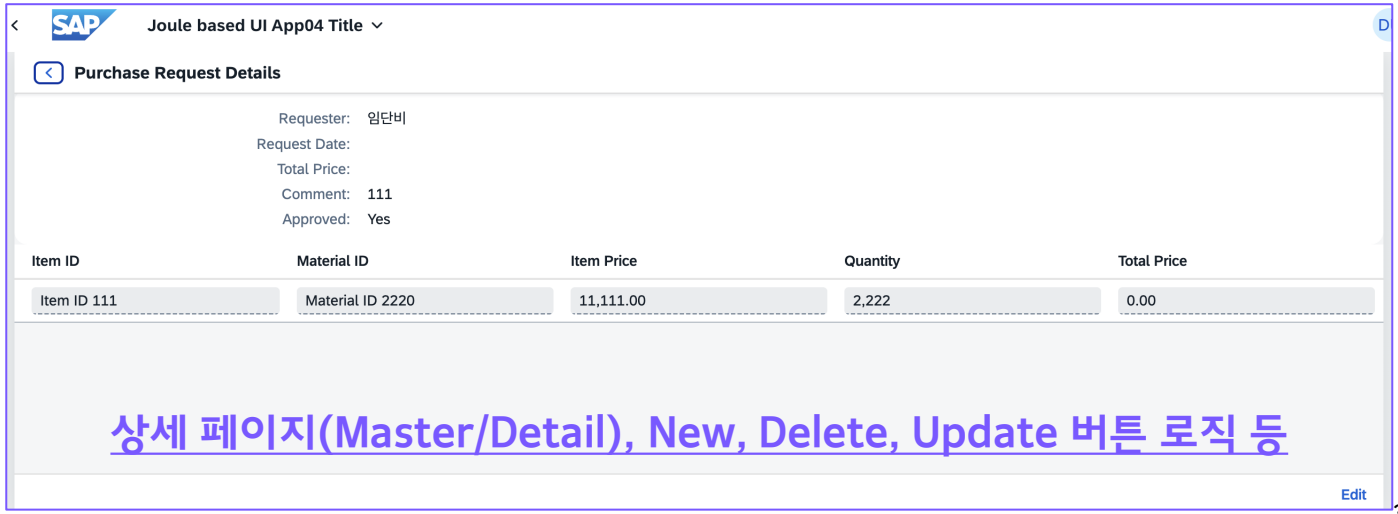
Prompt N :

- i want to make "Detail" page about PurchaseReq details and its PurchaseReqItem. when you make this, please use metadata as a reference.

...

- i think it should be nicer in form maybe you forget to put layout of Form in detail view
- can you change form to simple form?

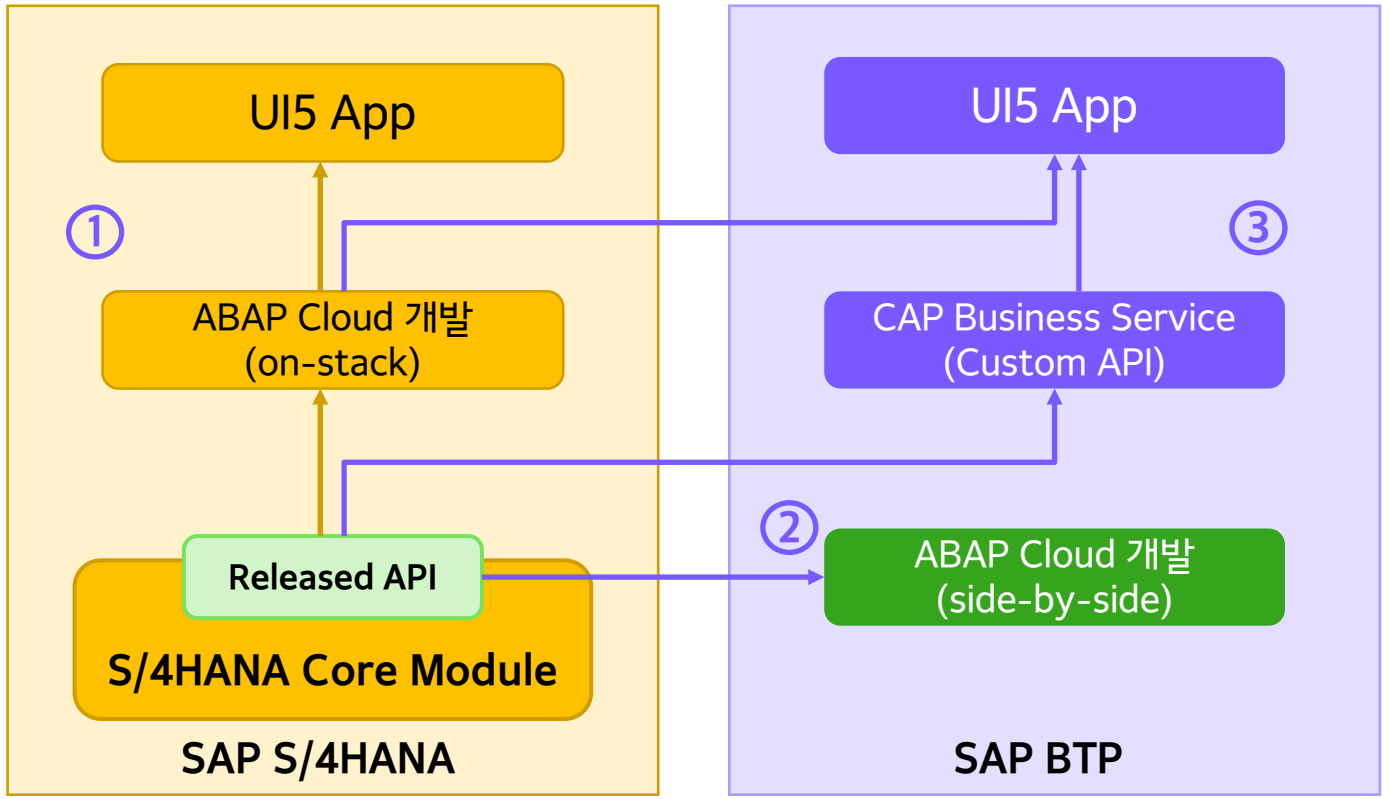




상세 페이지(Master/Detail), New, Delete, Update 버튼 로직 등

Key Takeaways

생성형 AI 개발 서비스 활용 범위



Clean Core Assessment 제공(CCA)

- 프로그램 진단 및 extensibility framework 적용

① J4D 기반 Clean Core on-stack ABAP Cloud 프로그램 개발

- On-stack ABAP Cloud application 개발
- 로드맵 상 예정 : Classic ABAP to ABAP Cloud 전환

② J4D 를 기반으로 한 Clean core side-by-side ABAP Cloud 개발

- ABAP Cloud 기반 ERP 확장 application 개발

③ J4D 를 기반으로 한 Clean core(side-by-side) – 오픈 소스(Javascript) 개발

- CAP(Cloud Application Programming) framework – Backend API 개발.
- UI5 freestyle frontend app개발

INTERNAL – SAP and Partners Only

SAP NOW AI Tour

감사합니다.

BTP Solution Advisory

한정우 파트너

Bring out your best.

© 2024 SAP SE or an SAP affiliate company. All rights reserved. See Legal Notice on www.sap.com/legal-notice for use terms, disclaimers, disclosures, or restrictions related to this material.