

A Hitchhiker's Guide to SAP Fiori UX and its Technologies (Part 2)

Oliver Graeff, Director – Product Management, SAP
October 2025

PUBLIC



Disclaimer

The information in this presentation is confidential and proprietary to SAP and may not be disclosed without the permission of SAP. Except for your obligation to protect confidential information, this presentation is not subject to your license agreement or any other service or subscription agreement with SAP. SAP has no obligation to pursue any course of business outlined in this presentation or any related document, or to develop or release any functionality mentioned therein.

This presentation, or any related document and SAP's strategy and possible future developments, products and or platforms directions and functionality are all subject to change and may be changed by SAP at any time for any reason without notice. The information in this presentation is not a commitment, promise or legal obligation to deliver any material, code or functionality. This presentation is provided without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. This presentation is for informational purposes and may not be incorporated into a contract. SAP assumes no responsibility for errors or omissions in this presentation, except if such damages were caused by SAP's intentional or gross negligence.

All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.

Agenda

Part 1 (Tuesday)

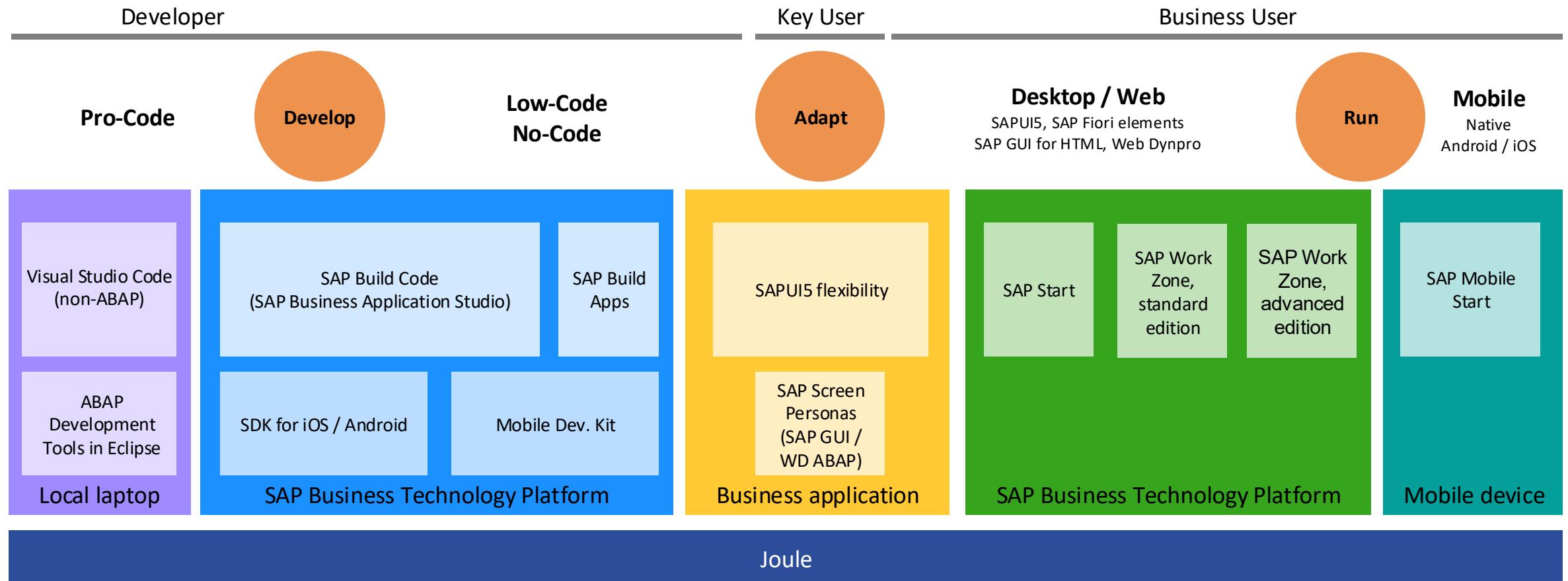
1. SAP Fiori User experience
2. The digital experience for business users
3. Q&A

Part 2 (Today)

1. Building apps with SAP Build
2. Extending standard SAP apps
3. Q&A



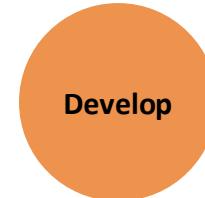
SAP Fiori user experience enabled by technologies & tools



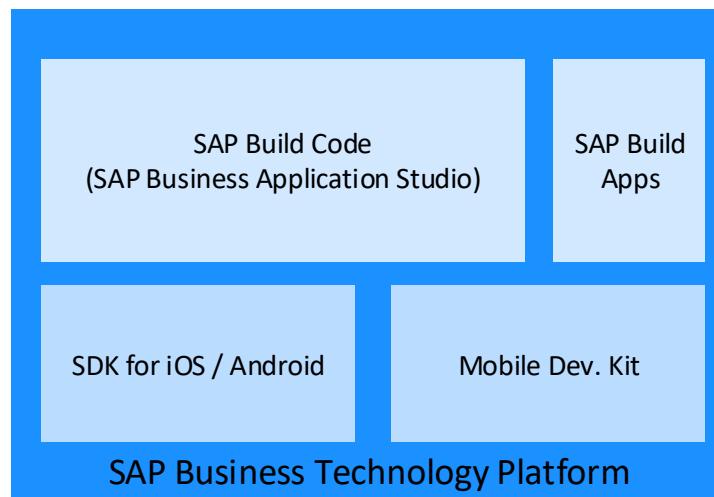
SAP Fiori user experience enabled by technologies & tools

Developer

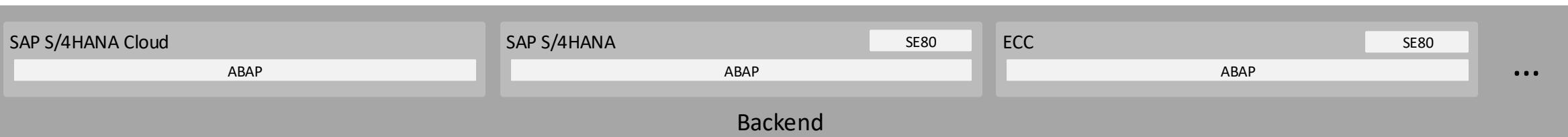
Pro-Code



Low-Code
No-Code



Joule



SAP's Development Environments: pro-code, low-code and no-code



Why SAP Build is the backbone of extensibility

Your strategy wins with extensibility; extensibility requires SAP Build.

Scale with confidence

- Create upgrade-safe and clean core extensions
- Build on governance that you already trust with SAP BTP
- Develop extensions across lines of business and extend SAP S/4HANA, SAP Ariba and SAP SuccessFactors solutions, and more

Reduce complexity

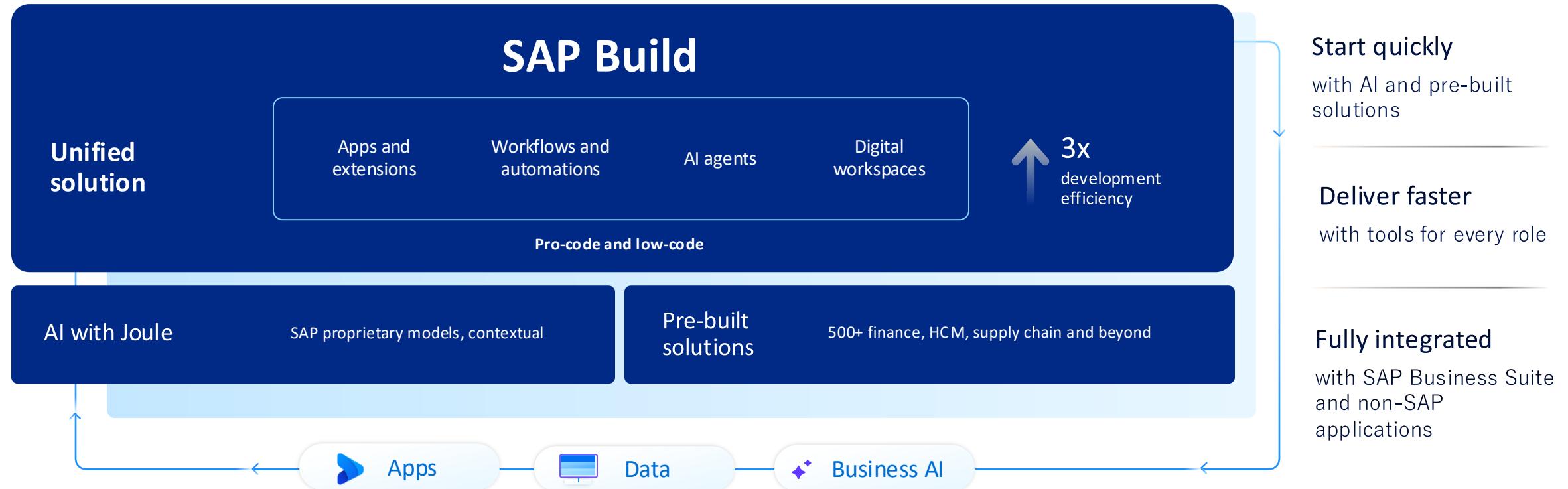
- Use unified SAP Build licensing and see everything included with SAP Business Suite
- Simplify how you build across SAP with a unified platform
- Reclaim developer capacity and reduce development effort by 59%

Deliver faster

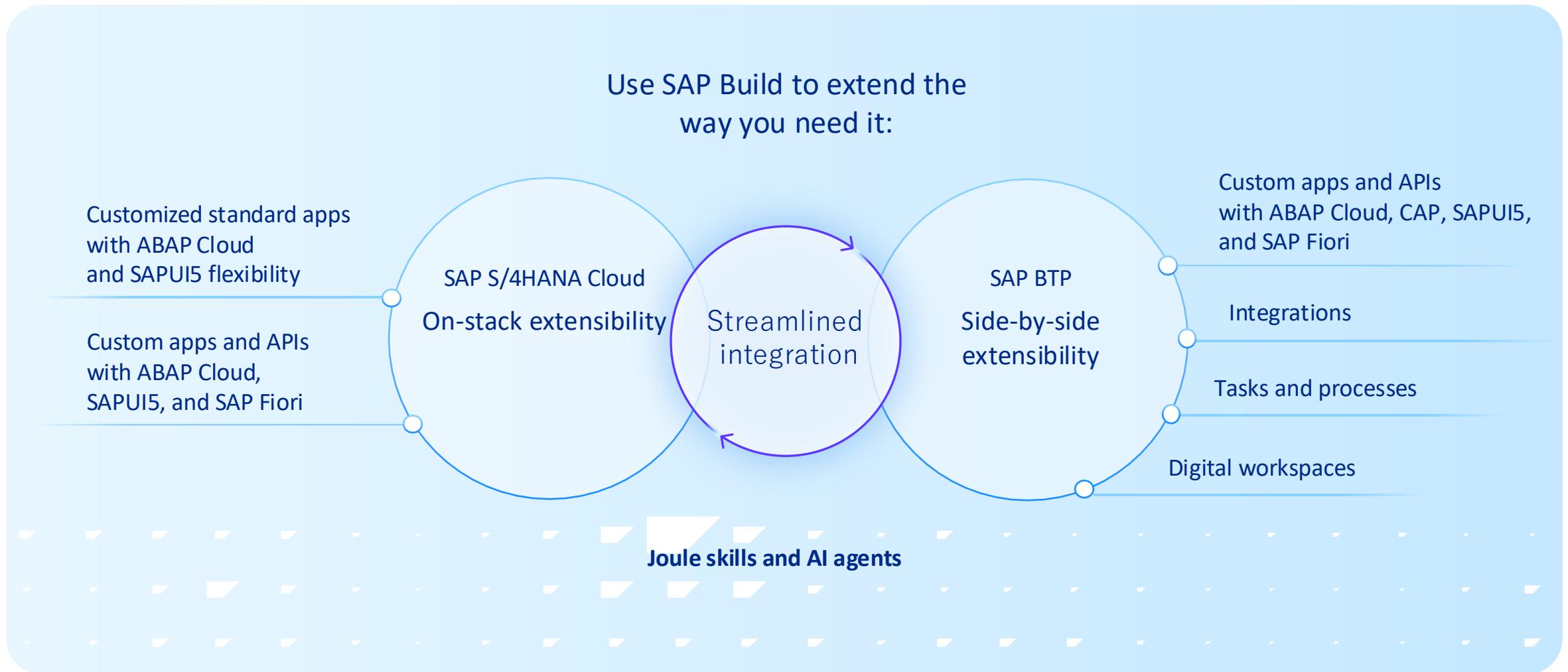
- Accelerate every phase of development with AI
- Use best-practice frameworks (CAP, ABAP Cloud development, SAPUI5)
- Jump-start work with prebuilt solutions and a full library of templates

SAP Build helps you deliver results 3x faster

Build where your business lives, in the context of your applications, data and AI.



Extensibility portfolio through SAP Build



SAP Build Apps: full-stack enterprise-apps in minutes – zero coding required

Drag-and-drop the user interface

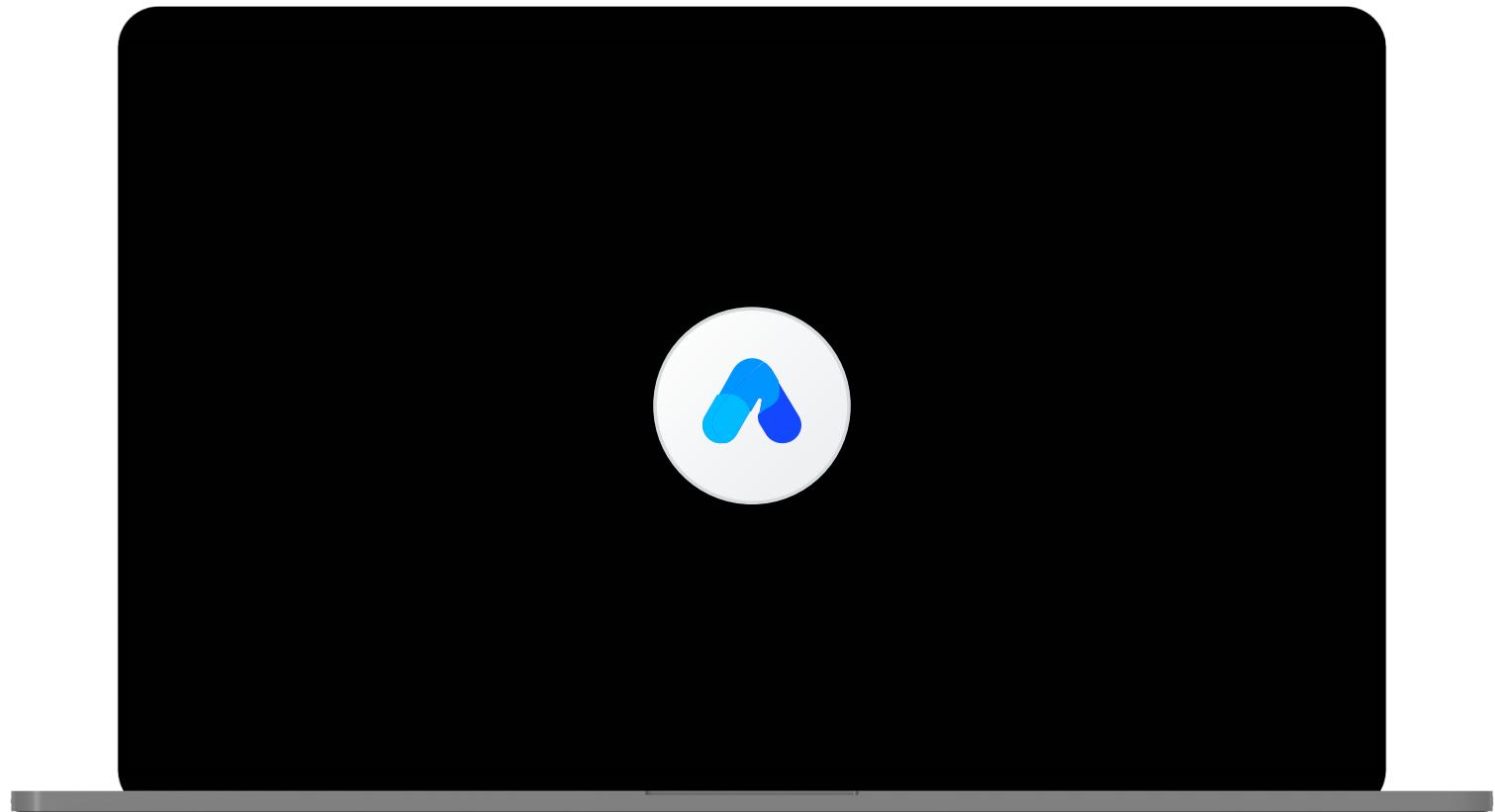
Access a pallet of user interface controls, drag them into place, and preview the final result in real time.

Create any logic without code

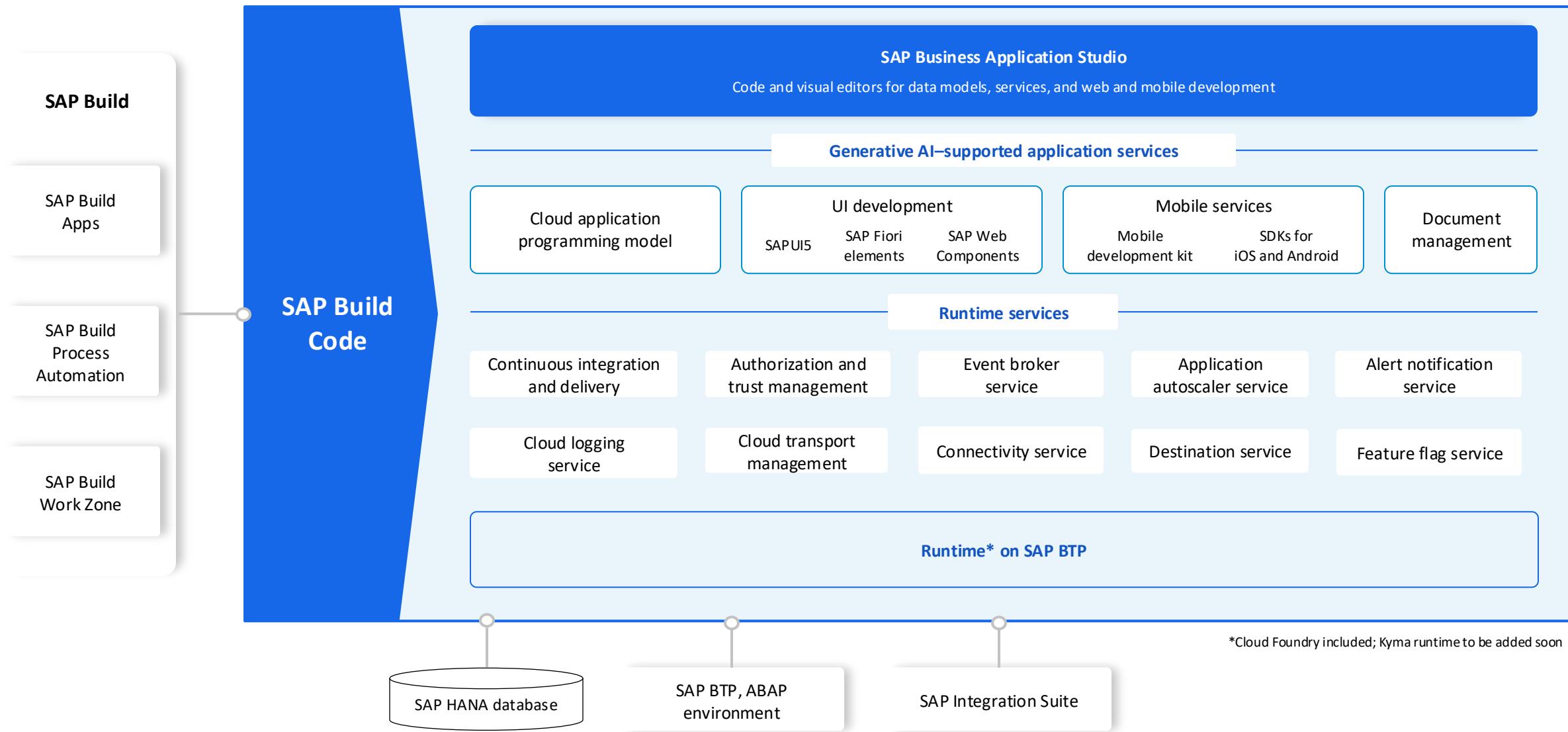
Drag-and-drop any logic function – sensors, file operations, data functions, animations, and more – onto a canvas.

Easily integrate data

Easily add your own data integrations or get started with some of ours.



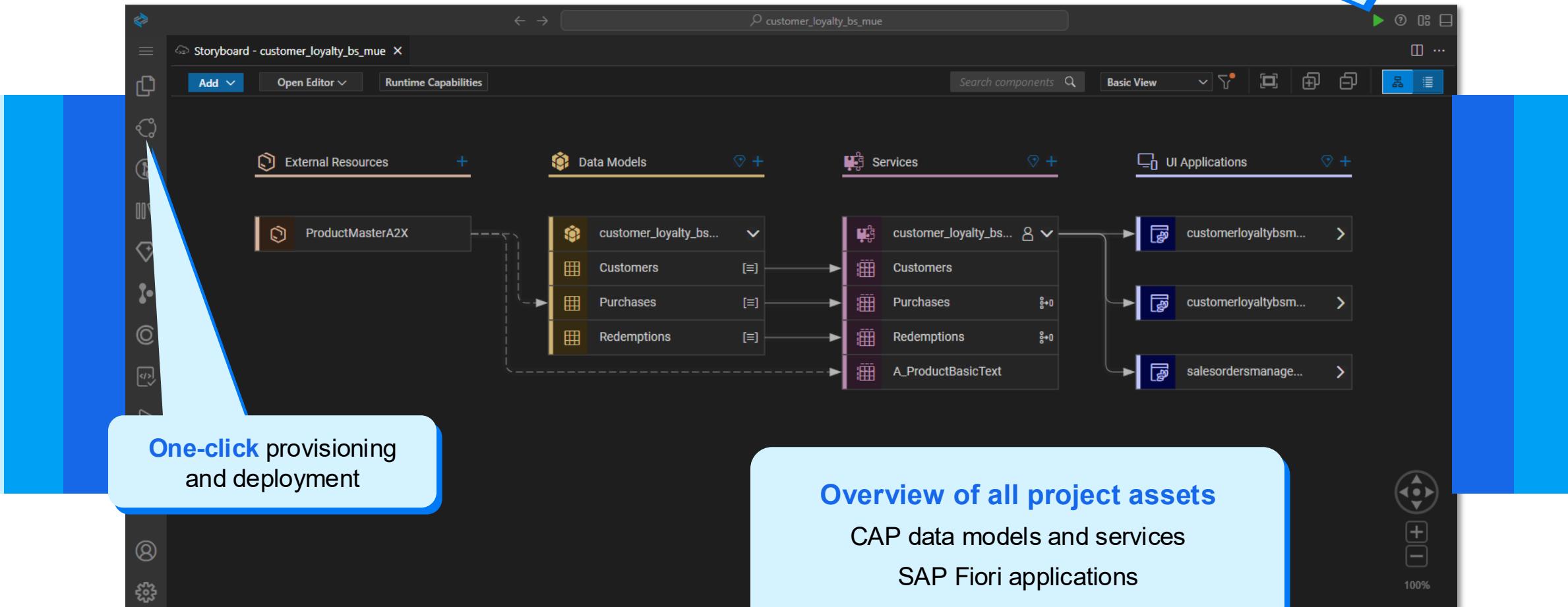
SAP Build Code: What you need to follow best practices of SAP



High productivity environment for developers

Unified development experience for SAP applications and extensions

Preview for
fast turn-around



Benefit using SAP's programming models in SAP Build Code

Increased developer productivity and future-proof IT investments

- Focus on application domain
- Future proof app development projects
- Reduce boilerplate code
- Minimize risks associated with technology changes
- Leverage enterprise best practices
- No lock-in effect through abstraction layer

Frontend SAPUI5 & SAP Fiori elements

Consistent and enterprise-ready UX based on SAP Fiori Design System

- Responsive to all platforms
- Hundreds of enterprise UI elements
- Accessibility
- Translation support
- Easy to integrate and extend

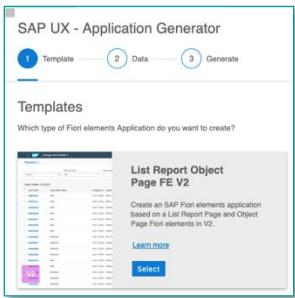
Backend SAP Cloud Application Programming Model

Intrinsic Cloud Qualities

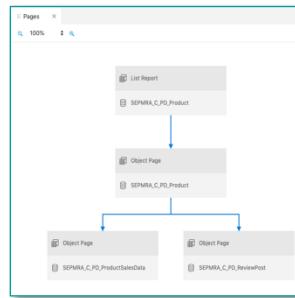
- Multitenancy
- Extensibility
- Cloud Security
- Scalability
- Messaging
- Observability
- Resilience

SAP Fiori tools guides developers through the full cycle

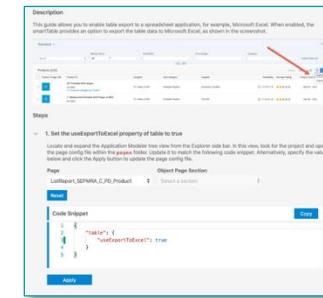
Generates the project environment for development



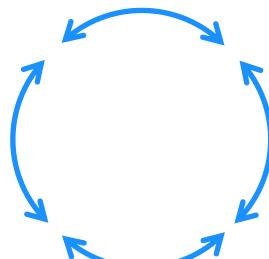
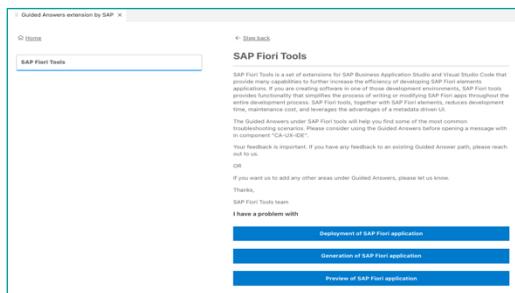
Allows editing of the app structure: pages, content, properties, and navigation



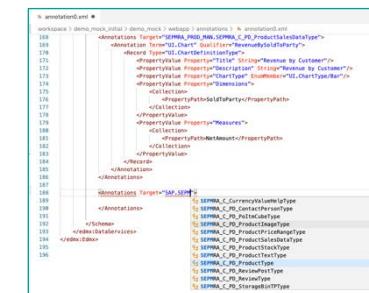
Provides step-by-step development instructions and generates code snippets in-line



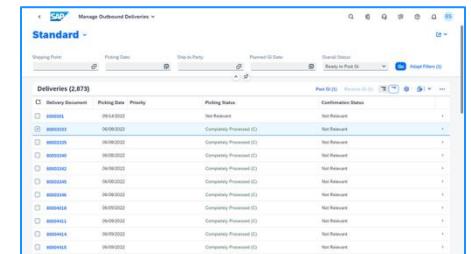
Offers troubleshooting tips (open source)



Delivers code completion using a language server protocol for annotations

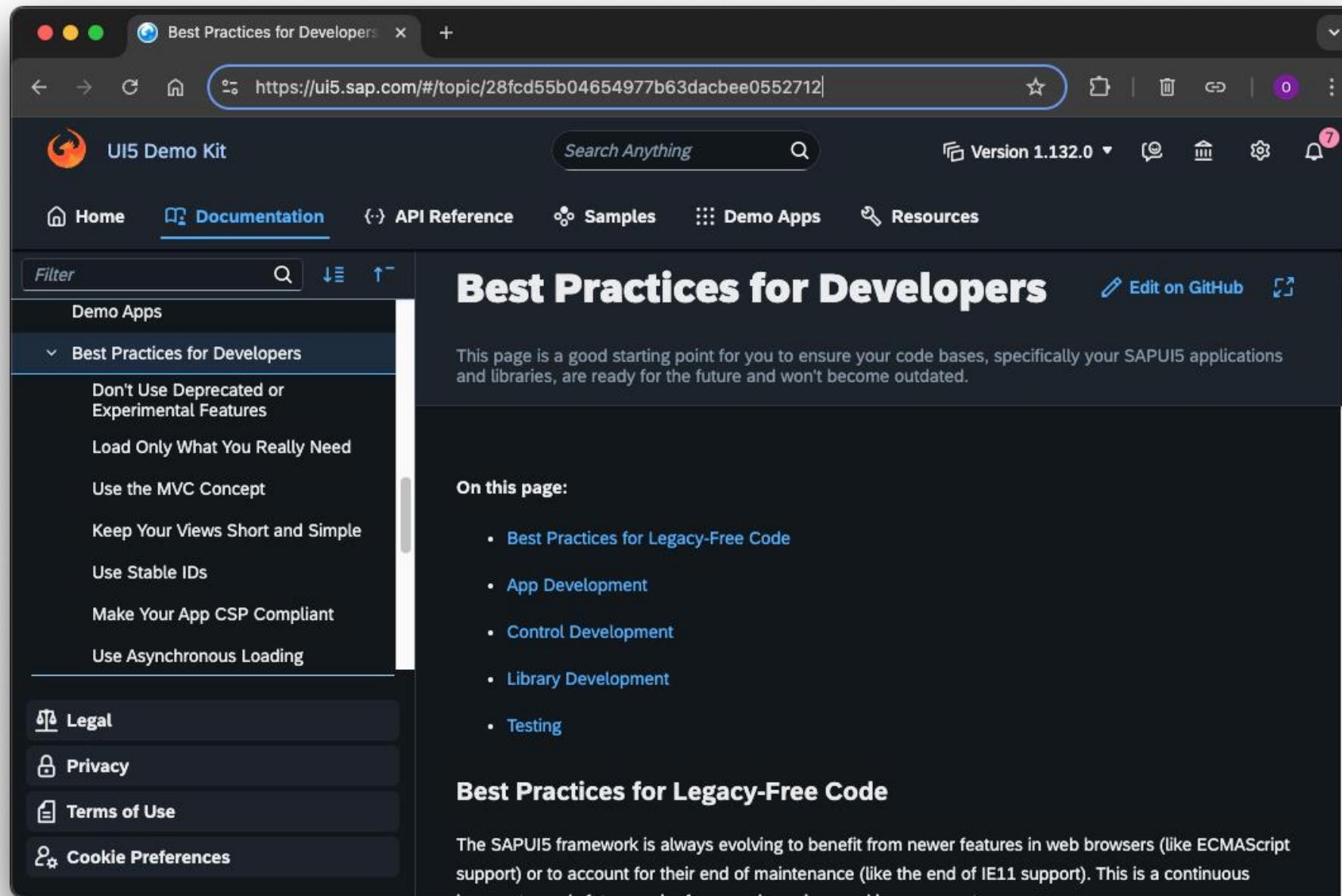


Provides preview of web app in the browser either with mock data or live data



Available in
SAP Business Application Studio
and
Visual Studio Code

SAPUI5 best practices for application development



The screenshot shows a web browser window with the title "Best Practices for Developers" and the URL <https://ui5.sap.com/#/topic/28fcd55b04654977b63dacbee0552712>. The page is part of the "UI5 Demo Kit" and is titled "Best Practices for Developers". The main content area contains a sub-section titled "Best Practices for Legacy-Free Code". The browser interface includes a search bar, a navigation bar with links to Home, Documentation, API Reference, Samples, Demo Apps, and Resources, and a sidebar with a "Demo Apps" section containing a list of best practices: "Don't Use Deprecated or Experimental Features", "Load Only What You Really Need", "Use the MVC Concept", "Keep Your Views Short and Simple", "Use Stable IDs", "Make Your App CSP Compliant", and "Use Asynchronous Loading". The bottom of the page features links for Legal, Privacy, Terms of Use, and Cookie Preferences.

Best Practices for Developers

This page is a good starting point for you to ensure your code bases, specifically your SAPUI5 applications and libraries, are ready for the future and won't become outdated.

On this page:

- [Best Practices for Legacy-Free Code](#)
- [App Development](#)
- [Control Development](#)
- [Library Development](#)
- [Testing](#)

Best Practices for Legacy-Free Code

The SAPUI5 framework is always evolving to benefit from newer features in web browsers (like ECMAScript support) or to account for their end of maintenance (like the end of IE11 support). This is a continuous

Technical debt is slowing down innovations

STATUS QUO



Existing code base

- Unclear responsibility
- Legacy from early Fiori projects
- Historically grown code
- Usage of deprecated APIs
- Neglected best practices

Accumulated
technical debt

How?

AI tool support

Development effort

GOAL



Modern code base using latest standards

- Adopted innovations
- Implemented AI capabilities
- Ready for future SAPUI5 versions
- Easy, constant modernization
- Reasonable investments / effort
- Better quality and maintainability

Future proof
SAPUI5 Apps

Static code analysis with UI5 linter

-  UI5 linter is a static **code analysis tool** for UI5 application code
-  **Autofix** option can automatically correct certain findings
-  UI5 linter validates your SAPUI5 code against **UI5 best practices**
-  Ensure your code is ready for **upcoming SAPUI5 versions**

UI5 linter in Action

UI5 linter scans your UI5 code (JavaScript, TypeScript, XML, JSON, and other files) for:

- Usage of deprecated libraries
- Usage of deprecated framework APIs
- Usage of global variables
- Possible CSP violations
- Deprecated component and manifest configurations



<https://github.com/sap/ui5-linter>



TERMINAL

bash + v □ 🗑 ... > X

user: ui5-application \$

JS App.controller.js X

↻ ← → ⌂ ⌂ ⌂ ⌂ ⌂ ⌂ ⌂ ⌂ ⌂

webapp > controller > JS App.controller.js > ...

You, a day ago | 5 authors (You and others)

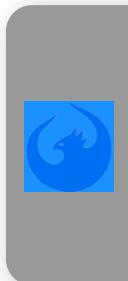
```
1  jQuery.sap.declare("sap.ui.demo.todo.controller.App");
2
3  jQuery.sap.require("sap.ui.Device");
4  jQuery.sap.require("sap.ui.core.mvc.Controller");
5  jQuery.sap.require("sap.ui.model.Filter");
6  jQuery.sap.require("sap.ui.model.FilterOperator");
7  jQuery.sap.require("sap.ui.model.json.JSONModel");
8
9  sap.ui.controller("sap.ui.demo.todo.controller.App", {
10
11    /**
12     * @param {sap.ui.core.mvc.Controller} oController
13     */
14    onInit: function() {
15      this.aSearchFilters = [];
16      this.aTabFilters = [];
17
18      this.getView().setModel(new sap.ui.model.json.JSONModel({
19        /**
20         * @param {sap.ui.core.mvc.Controller} oController
21         */
22        /**
23         * @param {sap.ui.core.mvc.Controller} oController
24         */
25        getModel: function() {
26          return this.getView().getModel();
27        },
28
29        /**
30         * @param {sap.ui.core.mvc.Controller} oController
31         */
32        addTodo: function() {
33          const oModel = this.getModel();
```

Introducing: SAPUI5/OpenUI5 legacy-free version



Challenge

- Legacy or historically grown code from early Fiori projects is not ready for the future.
- Usage of deprecated APIs
- Best practices are neglected.



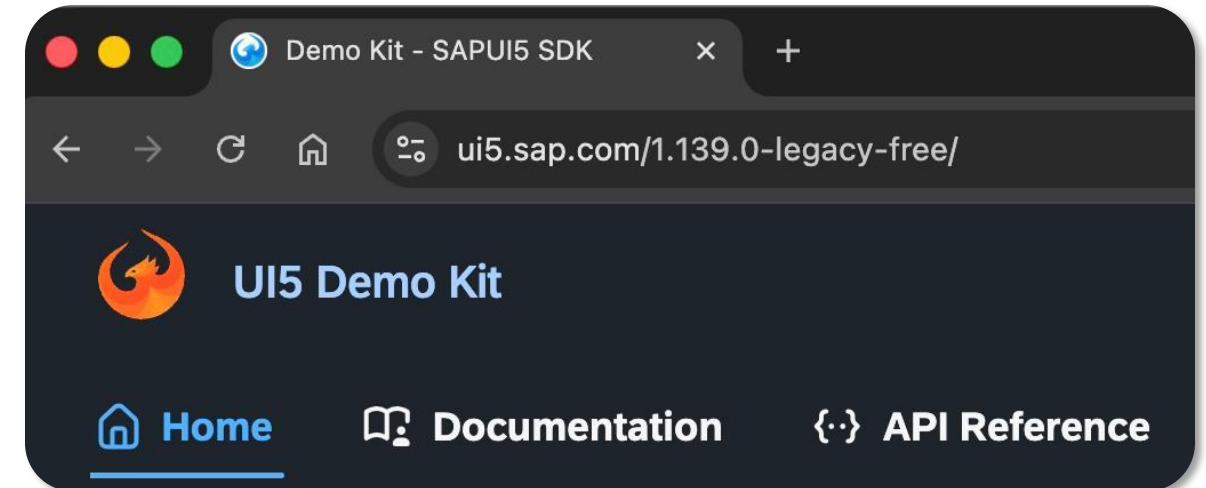
SAPUI5 / OpenUI5 legacy-free

- Availability of SAPUI5 / OpenUI5 1.139-legacy-free with deprecated code removed.

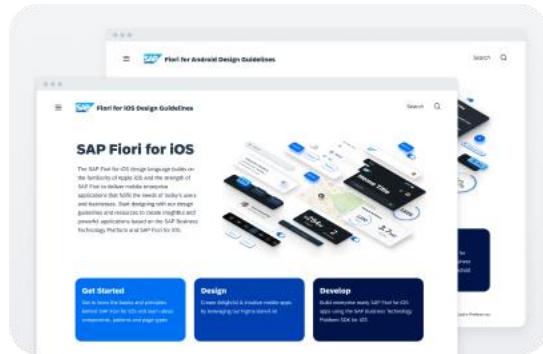


Benefits

- Test how compliant your code is to UI5 best practices.
- Ensure your new code is state-of-the-art and future-proof.
- Provide feedback to the UI5 team.

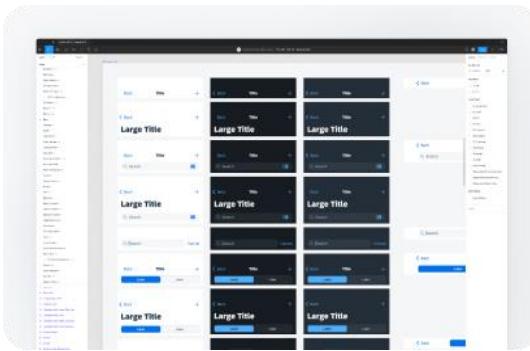


SAP Fiori for Mobile - Native mobile experience for iOS and Android



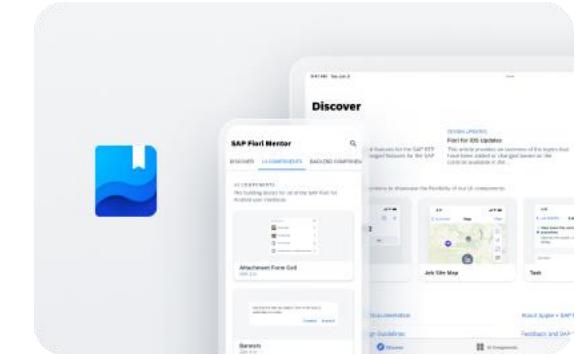
Design Guidelines

[Android](#) [iOS](#)



Figma kits

[Android](#) [iOS](#)



SAP Fiori Mentor App
for iOS and Android

[Android & iOS](#)



SAP Mobile
Development Kit ([MDK](#))



SAP BTP SDK for [iOS](#)



SAP BTP SDK for [Android](#)

SAP Build Code

SAP Build

- [Free trial](#)
- [SAP Build Learning journey](#)
- [SAP Build FAQ: Commercials, Getting Started and More](#)

SAP Build Apps

- [Get started with SAP Build Apps in SAP BTP Trial](#)

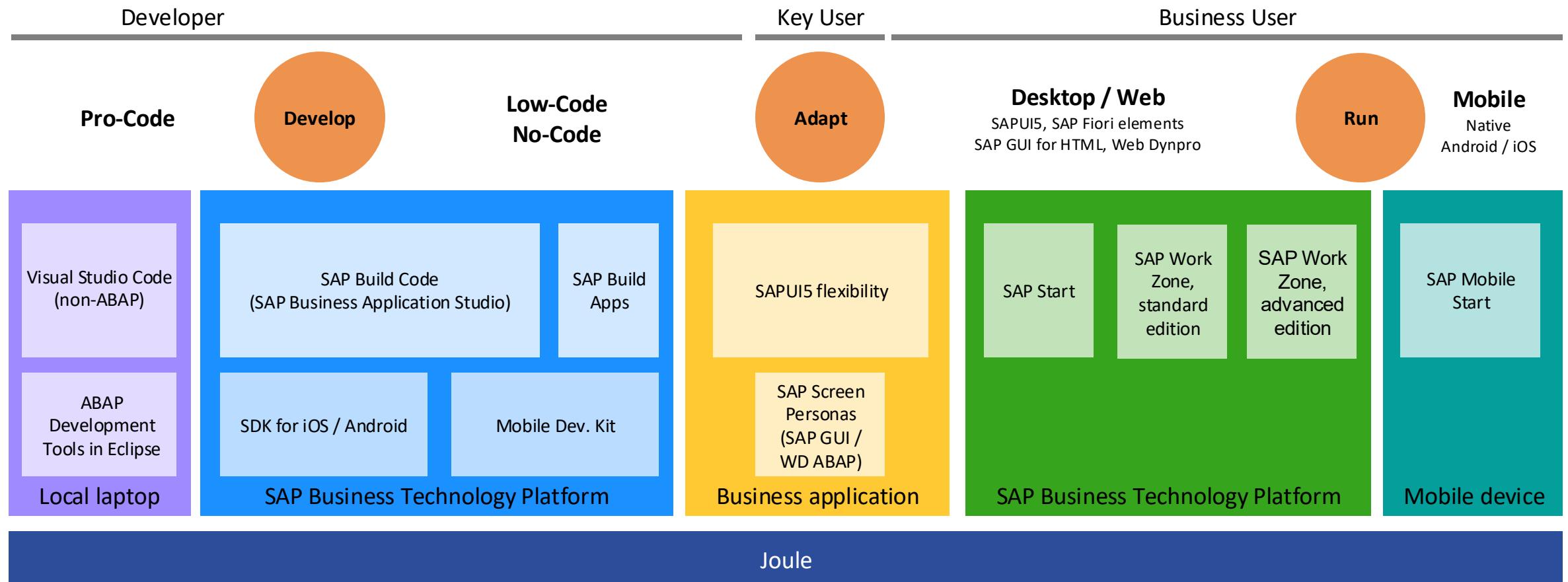
SAP Build Code / SAP Business Application Studio

- [Community page](#)
- [Try SAP Build Code](#)
- [Develop Full-Stack Apps Using Productivity Tools in SAP BAS](#)
- [Video \(Developing Cloud Apps – Big Picture\)](#)

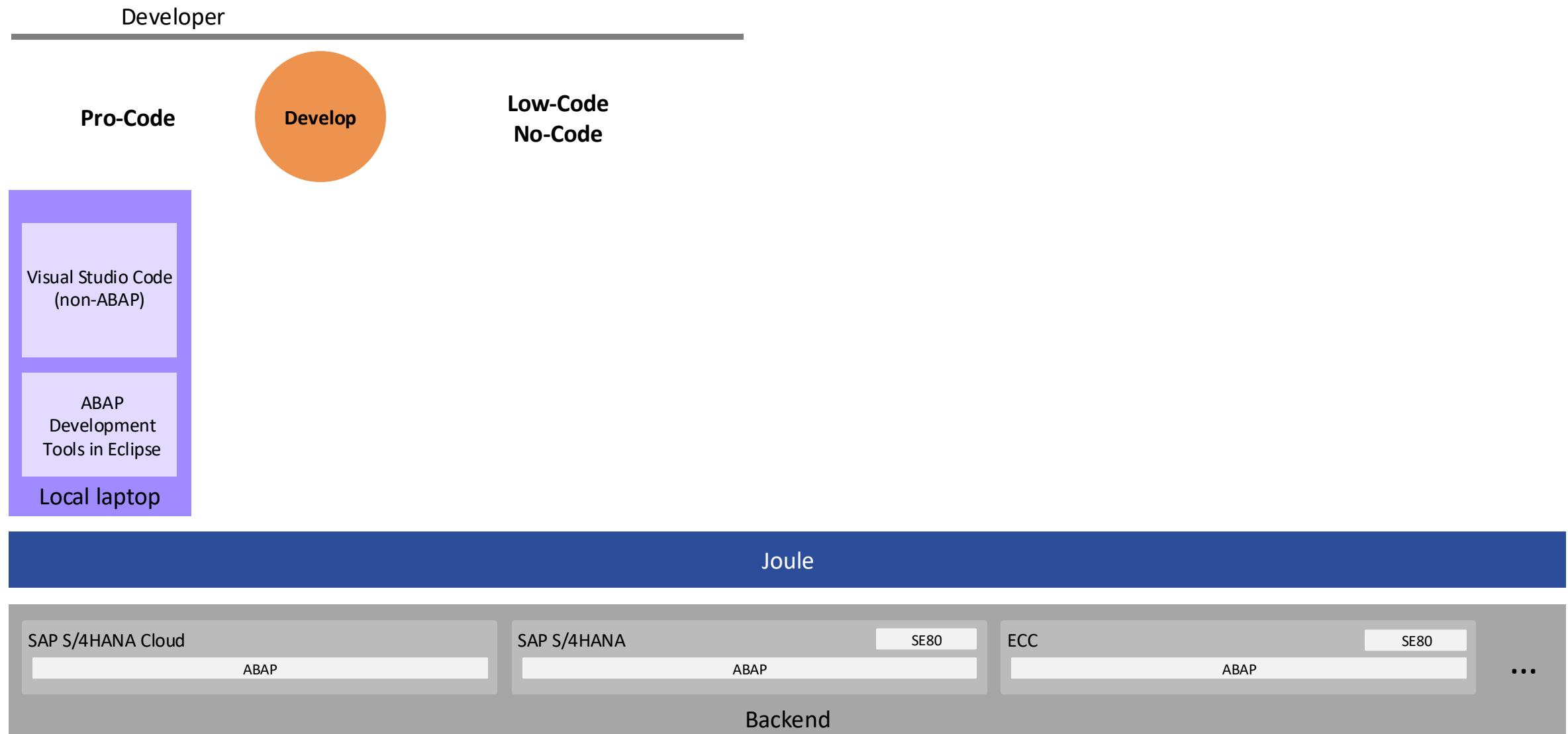
More missions / tutorials: [SAP Discovery Center](#) / [Developer tutorials](#)



SAP Fiori user experience enabled by technologies & tools



SAP Fiori user experience enabled by technologies & tools



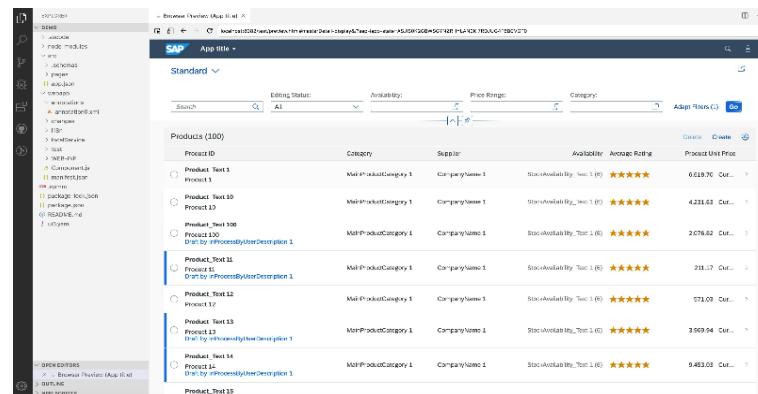
Local Development Environments: Visual Studio Code (VSCode) & ABAP Development Tools (ADT) in Eclipse



Visual Studio Code and SAP Business Application Studio

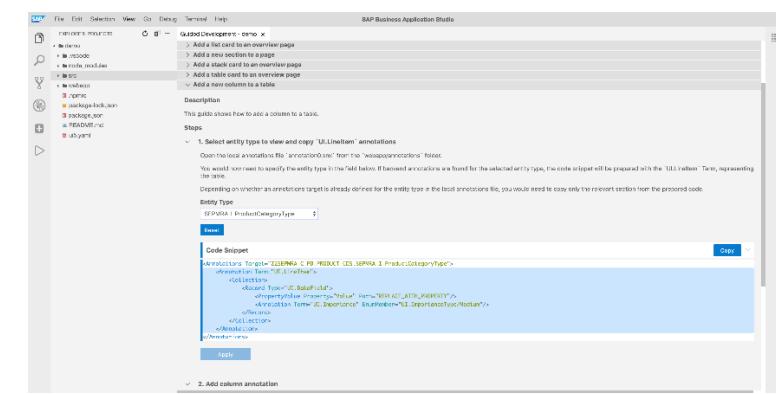
Visual Studio Code

- Provides development flexibility with **offline** capabilities
- Features a **developer-controlled**, highly configurable development environment
- Allows access to thousands of extensions from the Visual Studio Code marketplace
- Is part of a fast-growing open source community



SAP Build / SAP Business Application Studio

- Provides a **managed**, preconfigured, **hosted environment**, optimized for SAP application development (including client and server side extensions).
- Can be **centrally administrated** with tools repositories, systems access and company policies
- Integrates with existing SAP solutions, systems & services
- [Easy access to Visual Studio Code-compatible extensions](#) e.g. [SAP Fiori Tools](#)



Efficient ABAP development in Eclipse

MODERN DEVELOPMENT TOOLSET

Fully eclipse-based

Syntax check, Code completion

Syntax highlighting, Pretty printing

Navigation, Search, Quick Fixes

Available for ABAP releases \geq 7.31

QUALITY ASSURANCE

Static code checks (ATC, CVA) with
remote and local scenarios

Unit testing incl. isolation frameworks

Test seams and injections

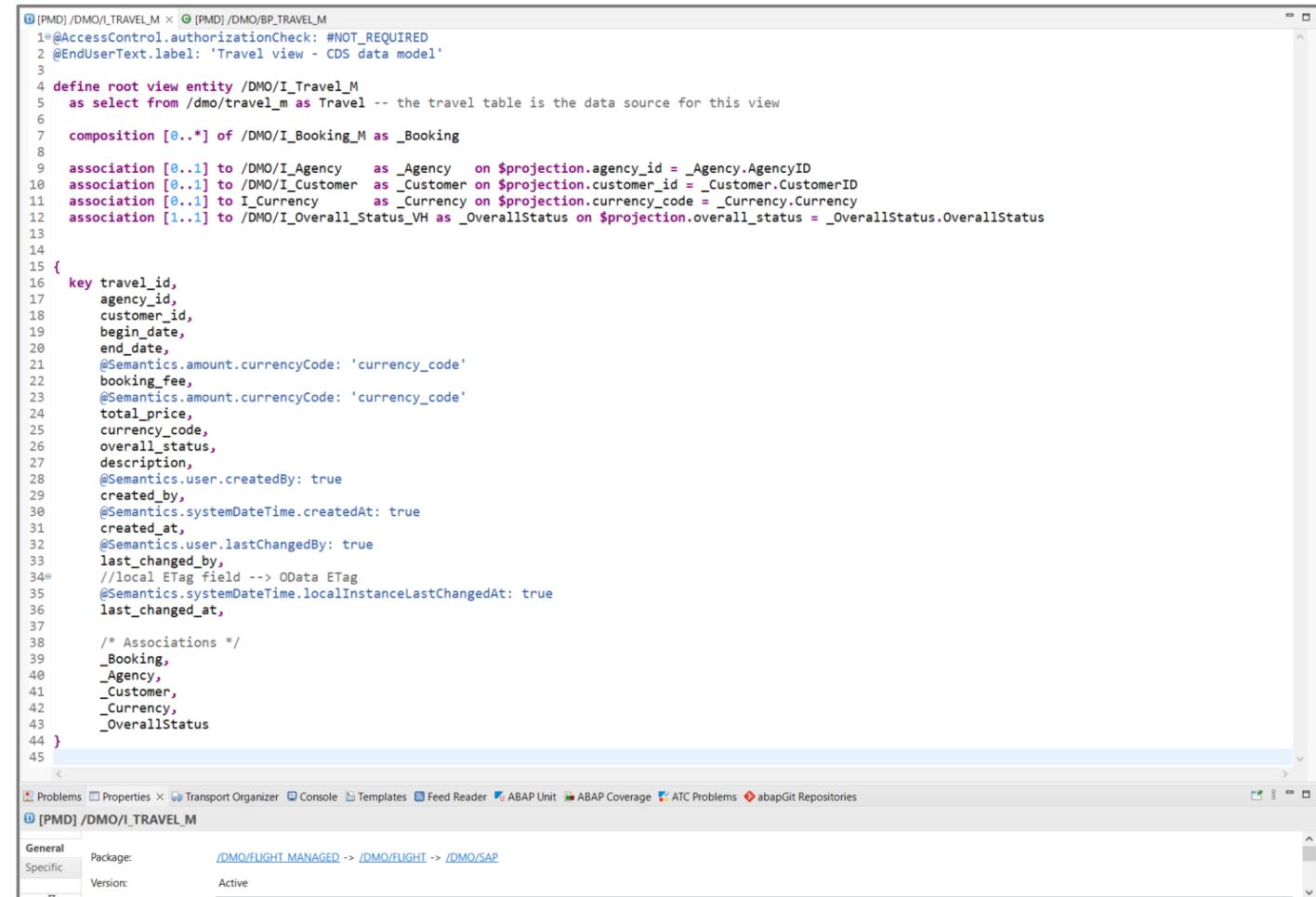
SUPPORTABILITY

Debugging, profiling, tracing

Static and dynamic logging

Runtime monitoring and analysis

HIGH DEVELOPER PRODUCTIVITY WITH THE ABAP DEVELOPMENT TOOLS (ADT)



```
1@AccessControl.authorizationCheck: #NOT_REQUIRED
2 @EndUserText.label: 'Travel view - CDS data model'
3
4 define root view entity /DMO/I_Travel_M
5   as select from /dmo/travel_m as Travel -- the travel table is the data source for this view
6
7   composition [0..*] of /DMO/I_Booking_M as _Booking
8
9   association [0..1] to /DMO/I_Agency as _Agency on $projection.agency_id = _Agency.AgencyID
10  association [0..1] to /DMO/I_Customer as _Customer on $projection.customer_id = _Customer.CustomerID
11  association [0..1] to I_Currency as _Currency on $projection.currency_code = _Currency.Currency
12  association [1..1] to /DMO/I_Overall_Status_VH as _OverallStatus on $projection.overall_status = _OverallStatus.OverallStatus
13
14
15 {
16   key travel_id,
17   agency_id,
18   customer_id,
19   begin_date,
20   end_date,
21   @Semantics.amount.currencyCode: 'currency_code'
22   booking_fee,
23   @Semantics.amount.currencyCode: 'currency_code'
24   total_price,
25   currency_code,
26   overall_status,
27   description,
28   @Semantics.user.createdBy: true
29   created_by,
30   @Semantics.systemDateTime.createdAt: true
31   created_at,
32   @Semantics.user.lastChangedBy: true
33   last_changed_by,
34   //local ETag field --> OData ETag
35   @Semantics.systemDateTime.localInstanceLastChangedAt: true
36   last_changed_at,
37
38   /* Associations */
39   _Booking,
40   _Agency,
41   _Customer,
42   _Currency,
43   _OverallStatus
44 }
45
```

The screenshot shows the Eclipse ADT interface. The main window displays an ABAP class definition for 'I_Travel_M'. The code is color-coded for syntax highlighting. Below the editor, the 'Package Explorer' shows a package structure with 'General' and 'Specific' sections. The 'General' section shows a package named '/DMO/FLIGHT MANAGED' with a sub-package '/DMO/FLIGHT' and a module '/DMO/SAP'. The 'Specific' section shows a version named 'Active'.

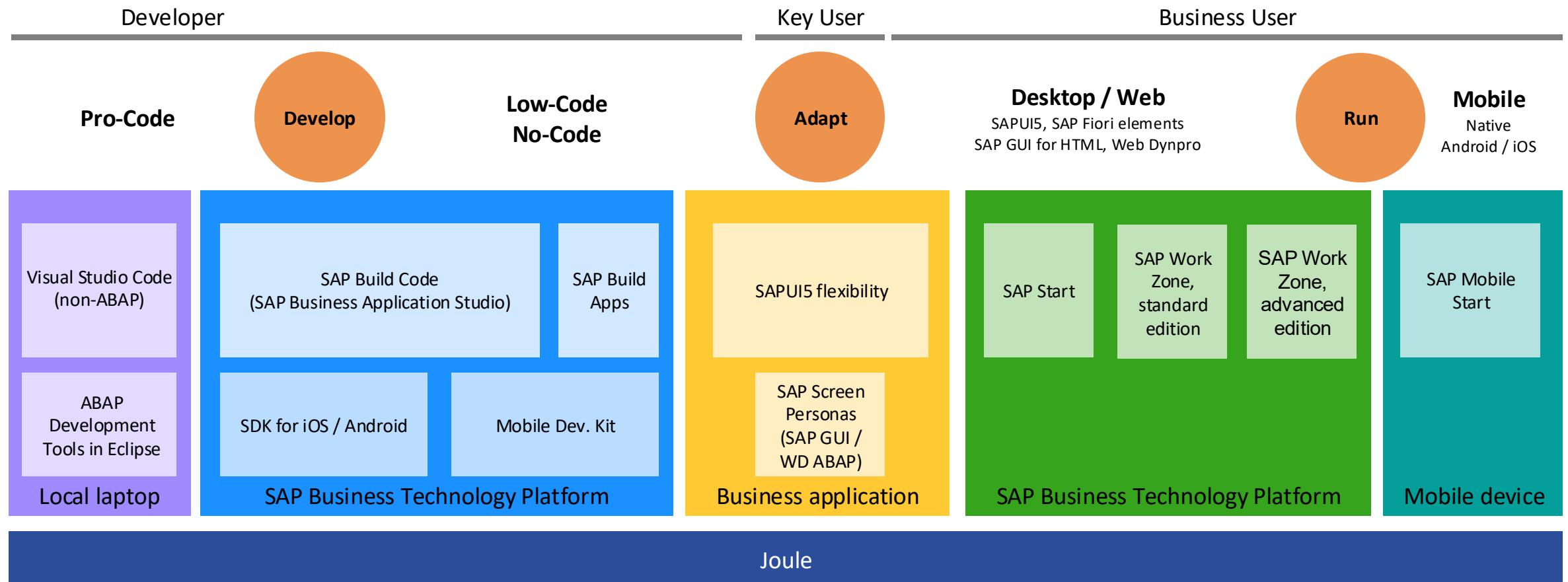
Local Development Environments

[Set Up SAP Fiori Tools in Your Visual Studio Code](#)

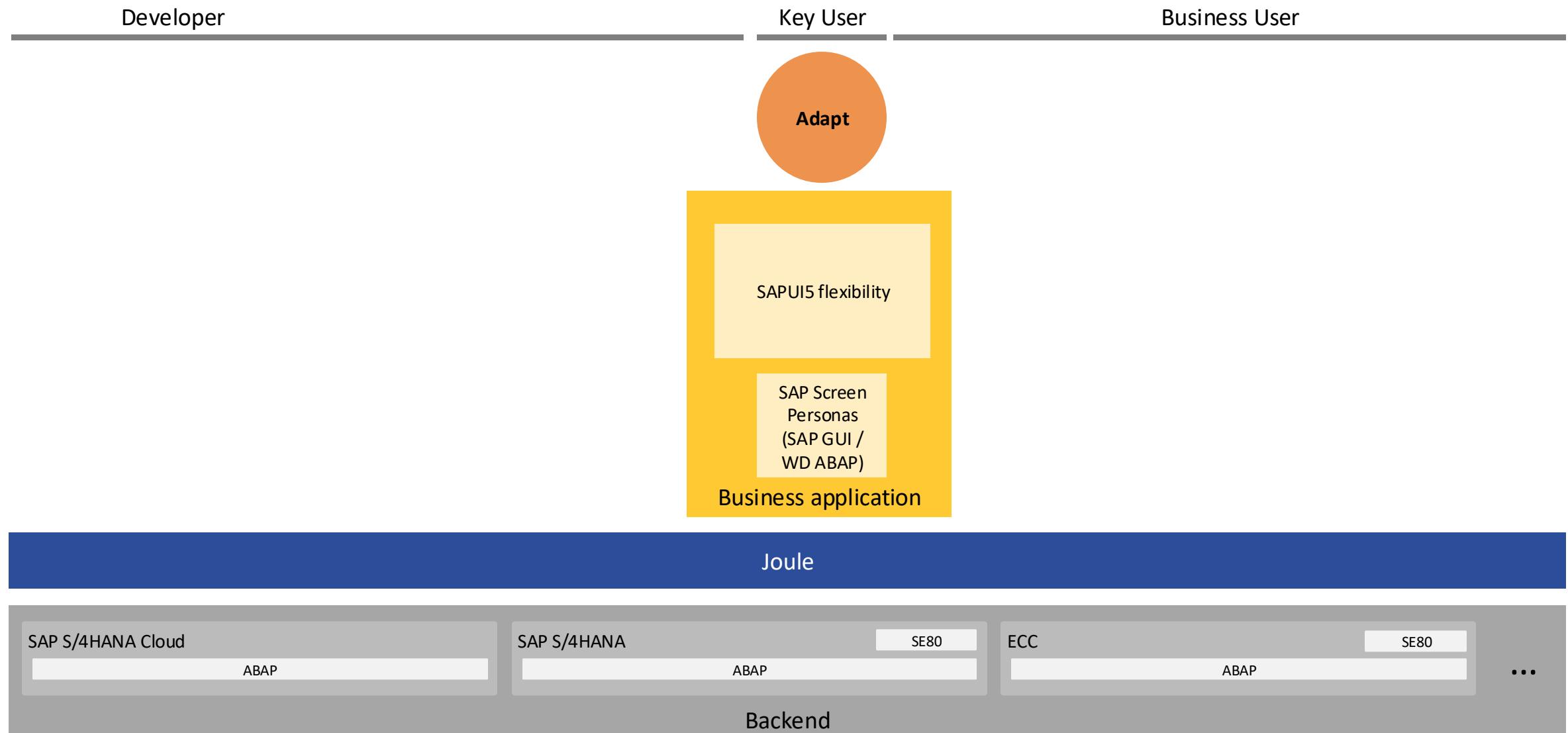
[Install ABAP Development Tools \(ADT\) in Eclipse](#)



SAP Fiori user experience enabled by technologies & tools



SAP Fiori user experience enabled by technologies & tools



Extending / Adapting SAP Standard Apps: SAP Screen Personas & SAPUI5 Flexibility



SAP Fiori Apps Reference Library

Library of all SAP content for the SAP Fiori launchpad. Explore, plan, and implement SAP Fiori apps as well as classic applications based on SAP GUI and WebDynpro.

- Overview of all SAP Fiori apps
- Information for each app, including technical data for installation and configuration
- Configuration information required to integrate classic SAP GUI and WebDynpro applications into the SAP Fiori launchpad
- Data for previous app versions
- Direct navigation to related resources, such as app documentation, Product Availability Matrix, and Maintenance Planner
- Recommendations for SAP Fiori apps for your needs

[SAP Fiori Apps Reference Library](#)

The screenshot shows the SAP Fiori Apps Reference Library interface. On the left, a list of SAP Fiori lighthouse apps is displayed, with 'Cash Flow Analyzer' selected. On the right, detailed information for the 'Cash Flow Analyzer' app is shown, including its title, required back-end product (SAP S/4HANA), line of business (Finance), app launcher title, application type (Analytical (SAP Fiori (SAPUI5))), database (HANA DB exclusive), device type (Desktop, Tablet), and app ID (F2332). The 'IMPLEMENTATION INFORMATION' tab is selected, showing a description of the app's functionality: 'With this app, you can overview the aggregated amounts and line item details of cash positions for all bank accounts and liquidity items. The data presented in the app is aggregated by quarters, or years for all bank accounts and liquidity items.' A red arrow points to this tab with the label 'Extensibility information'. At the bottom, a table of bank accounts is shown, with a red arrow pointing to the 'Filter' icon and another red arrow pointing to the table with the label 'Filter'.

Cash Flow Analyzer
for Cash Manager

SAP S/4HANA

Required Back-End Product SAP S/4HANA

Line of Business Finance

App Launcher Title(s) Cash Flow Analyzer

Application Type Analytical (SAP Fiori (SAPUI5))

Database HANA DB exclusive

Device Type(s) Desktop, Tablet

App ID F2332

PRODUCT FEATURES **IMPLEMENTATION INFORMATION**

With this app, you can overview the aggregated amounts and line item details of cash positions for all bank accounts and liquidity items. The data presented in the app is aggregated by quarters, or years for all bank accounts and liquidity items.

Read more in [App Documentation](#)
App demo: [D1: Cash Flow Analyzer](#)
Learn more about solution capability: [Cash and Liquidity Management \(S/4\)](#)

Filter

Bank Name / Bank Account Number	Description
11111111	11111111
ABN AMRO BANK N.V. - SA	ABN AMRO BANK...
ABN AMRO BANK N.V. - SA	ABN AMRO BANK...
BOC Global	BOC Global
Bank 2 - SAMPLE BANK	Bank 2 - SAMPLE...
25800000	Dummy text
Development Bank of S.	Development Bank...
500	Bank 1 Giro
404142483	Dummy text
Payroll	Payroll
399999911	Dummy text
399999912	Dummy text
Bank 1 - SAMPLE BANK	Bank 1 - SAMPLE...
Bank 1 - SAMPLE BANK	Bank 1 - SAMPLE...
Bank 1 - SAMPLE BANK	Bank 1 - SAMPLE...
Bank Heerhout Sample 1	Bank Heerhout Sa...

SAP Screen Personas simplifies complex classic ERP screens

Benefits

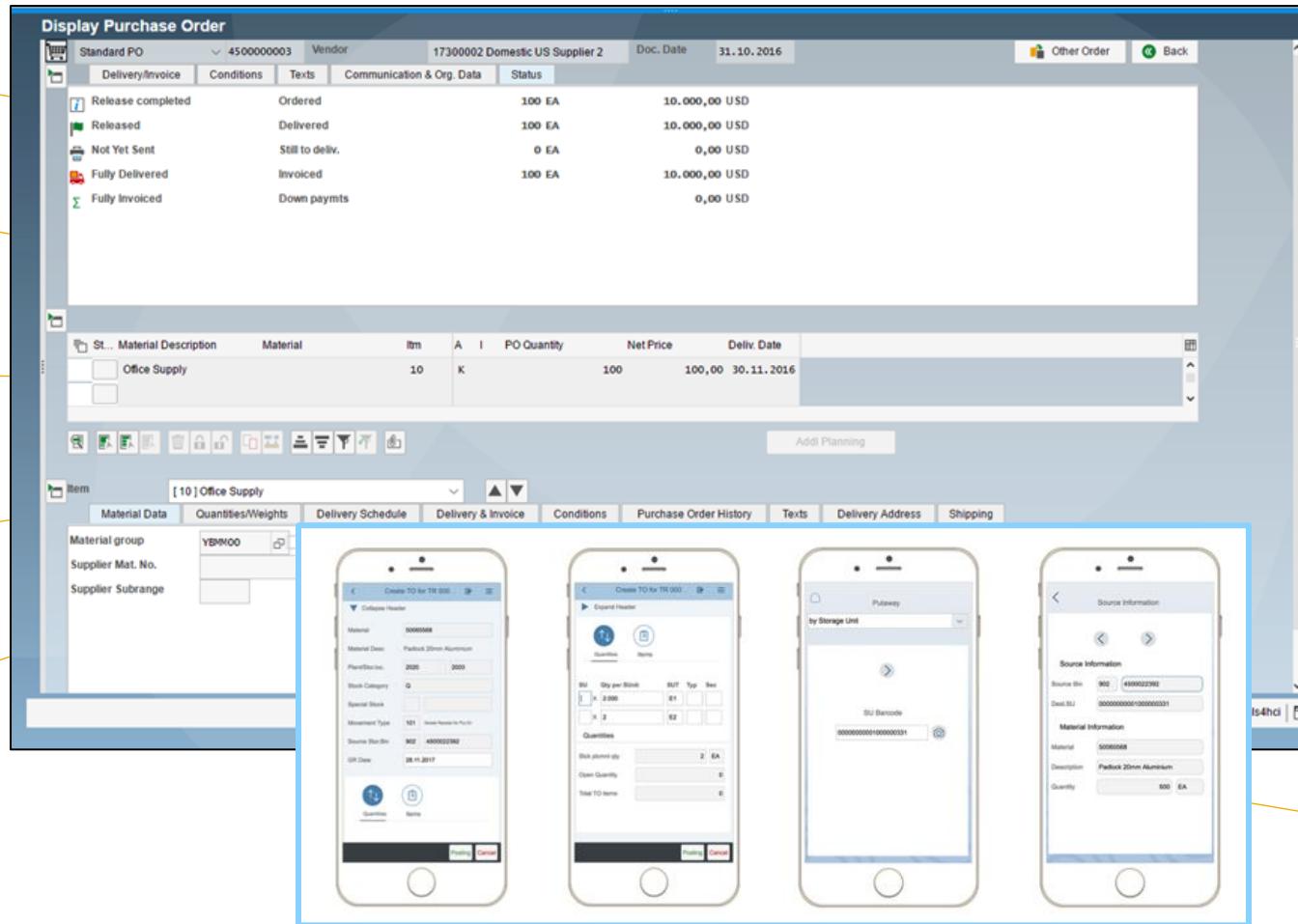
Improve business productivity

Increase SAP adoption by making screens easier to use

Decrease training time for SAP users with intuitive screens

Enhance user satisfaction

Reduce the cost of personalization compared to custom coding



www.sapscreenpersonas.com

Capabilities

Hide / rearrange screen elements, reduce scrolling

SAP Fiori design creation for custom or standard transactions, including SAP S/4HANA; embed into SAP Fiori launchpad

Adapt Dynpro or Web Dynpro ABAP screens

Pre-filled fields & pull-down menus as well as automation of repetitive tasks

Slipstream Engine renders flavors on desktop, tablet, or mobile phone

Extending SAP standard apps with SAPUI5 flexibility

SAP customers/partners experience challenges when extending standard apps, e.g. by adding custom fields:

- Pre-defined extension points are often not sufficient.
- Managing the lifecycle of extensions, which are coupled with the SAP standard.
- Safe-guarding investments in custom extensions.

A new standard for extensibility: [Clean Core](#) and [SAPUI5 flexibility](#), supporting different user roles:

End User

“I personalize apps for myself.”

Key User

“As a business expert, I adapt apps for my teams.”

Developer

“I extend SAP apps with custom code.”

Upgrade stability with layered changes in SAPUI5 flexibility



Extending SAP standard apps with SAPUI5 flexibility

Customers / partners need to future-proof their business and safeguard their investments

Benefits and Values of adapting standard apps with SAPUI5 flexibility



Cost-efficient
UI change process
for extending apps



Intuitive
low-code/no-code tooling
for different personas



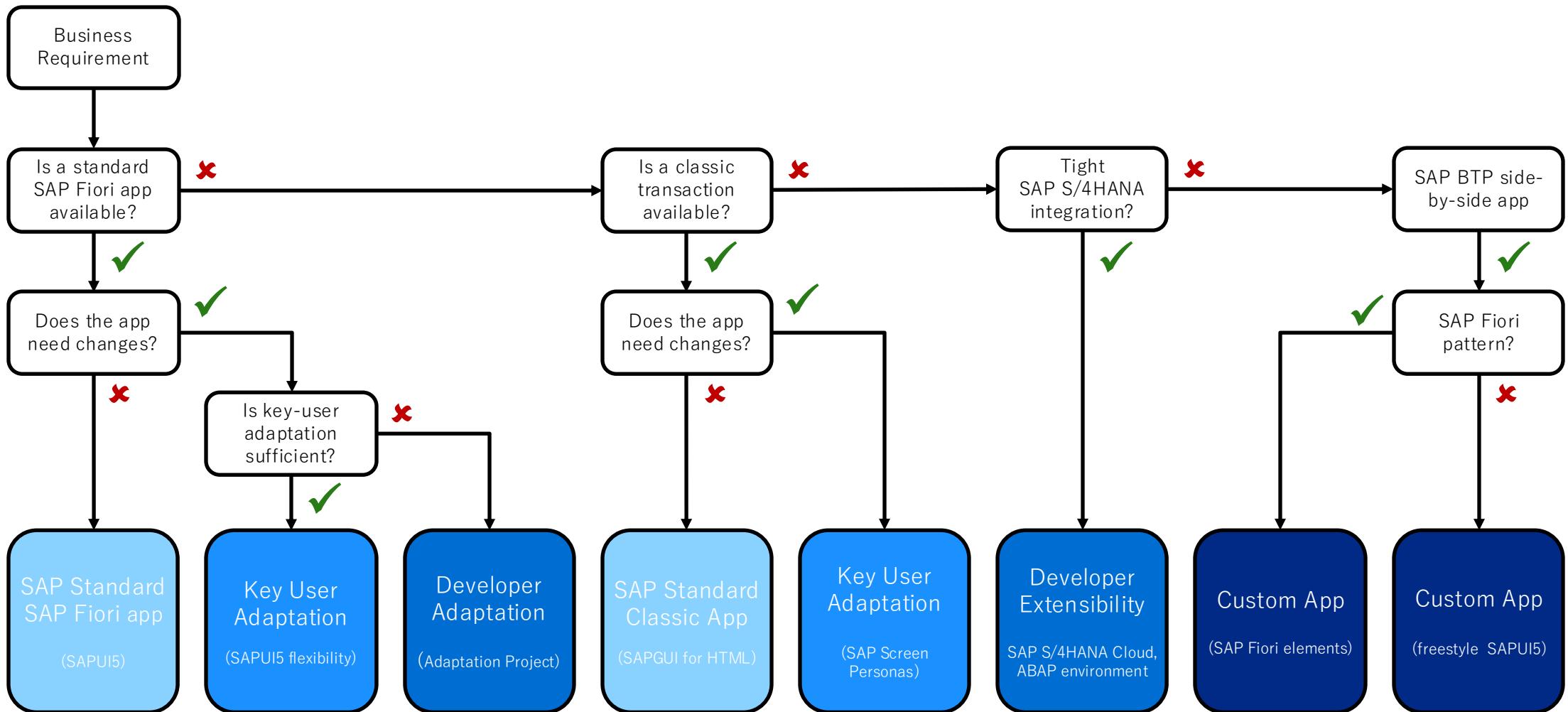
Lifecycle-stable,
modification-free UI
changes based on deltas



SAPUI5 framework
feature, no predefined
extension points needed

Extending standard SAP apps with SAP Build and SAPUI5 flexibility
realizes upgrade-safe and clean core extensions.

Choose the path to the SAP Fiori user experience based on your needs



Extending SAP standard apps: SAP Screen Personas & SAPUI5 Flexibility

[SAP Screen Personas](#) - SAP Community topic page

[SAPUI5 Flexibility](#) - All You Need to Know

<https://ui5.sap.com> > Demo apps > SAPUI5 Flexibility Demo Apps

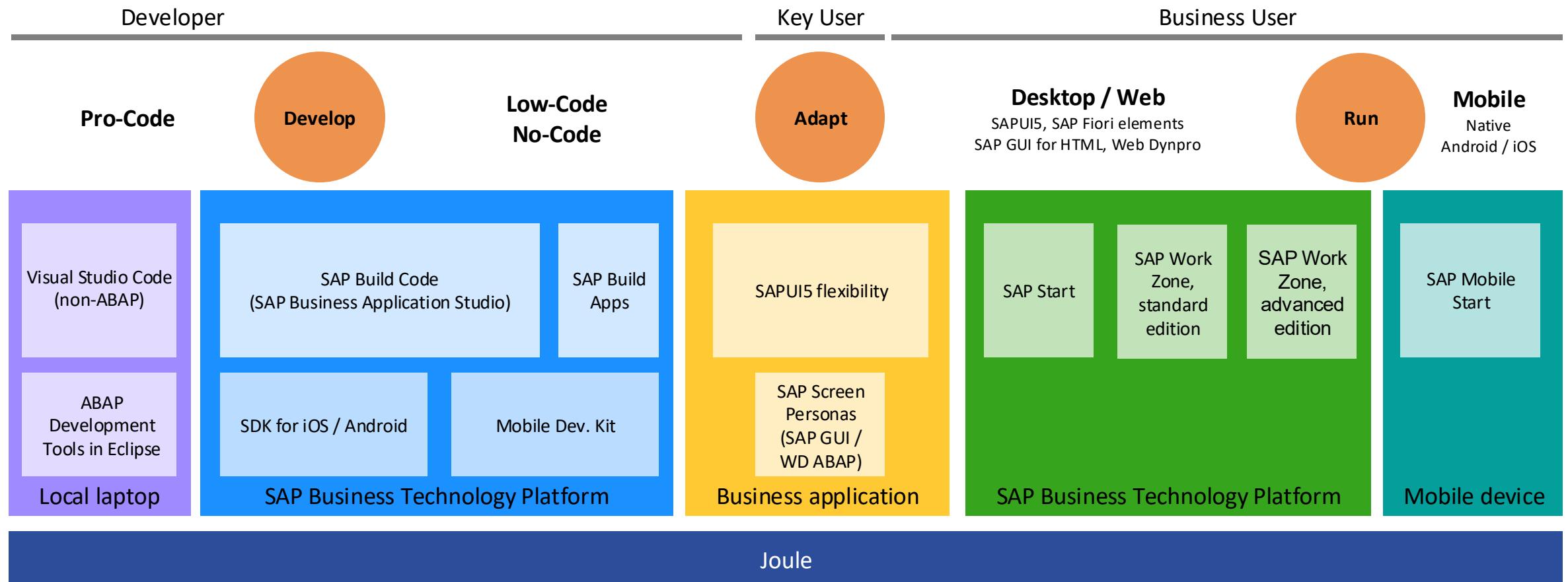
[Key User Adaptation Tutorial \(1\)](#)

[Key User Adaptation Tutorial \(2\)](#)

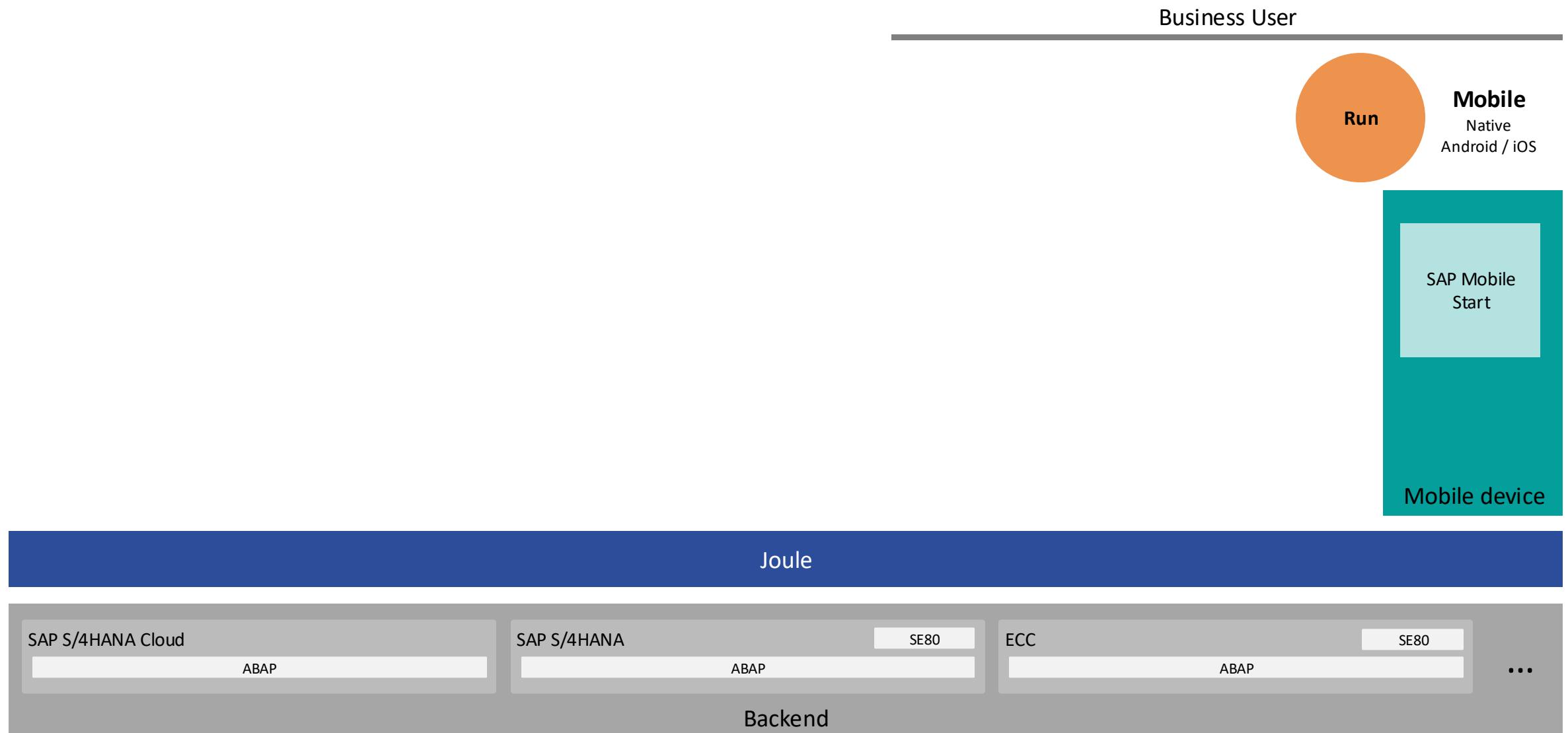
[Developer Adaptation Tutorial](#)



SAP Fiori user experience enabled by technologies & tools



SAP Fiori user experience enabled by technologies & tools

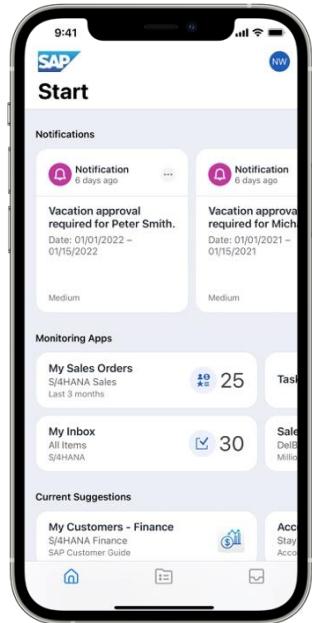


SAP Mobile Start: Native entry point to the Intelligent Enterprise

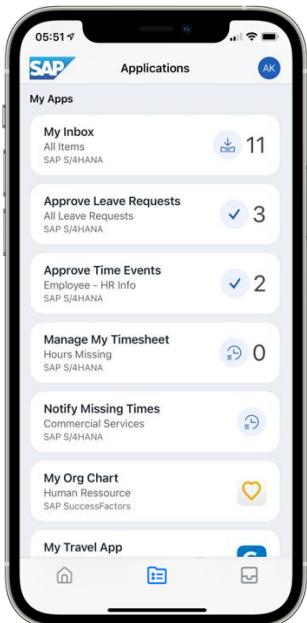


SAP Mobile Start

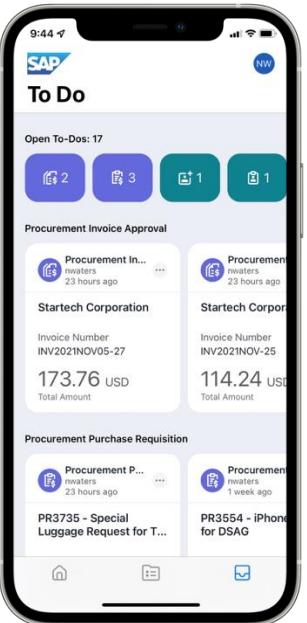
Native entry point to the Intelligent Enterprise Suite



Start screen
Latest notifications,
& app suggestions



Applications
Native apps,
Web responsive apps,
Web links & News feeds



ToDo screen
Tasks and Notifications

SAP S/4HANA

SAP Ariba 

SAP SuccessFactors 

SAP Fieldglass 

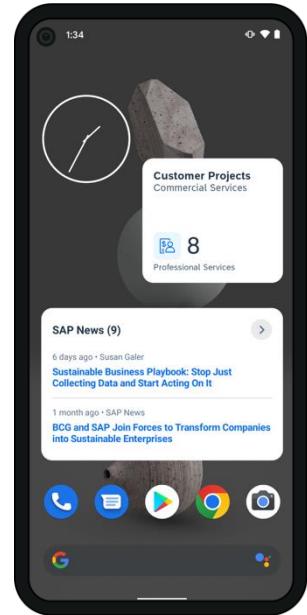
SAP Concur 

SAP Customer Experience

*This is the current state of planning and may be changed by SAP at any time without notice.

PUBLIC

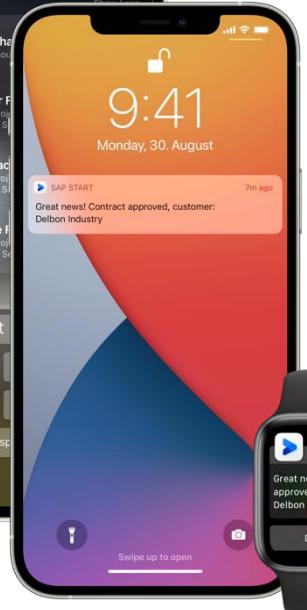
Mobile device integration (Android, iOS)



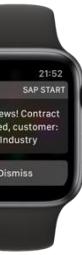
Widgets



iOS Spotlight
Search



Notifications



Watch Companion App for iOS



Complications*
Monitoring information
and number of tasks



To Dos
List of your To Dos, with details and filter
option



Take Action
Take action on suitable
tasks right from your
watch



Monitoring Info
Get your monitoring
info on your watch

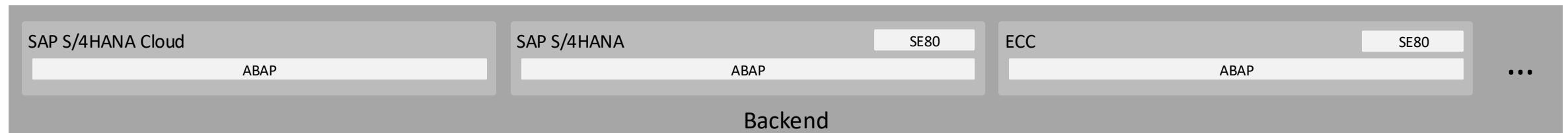
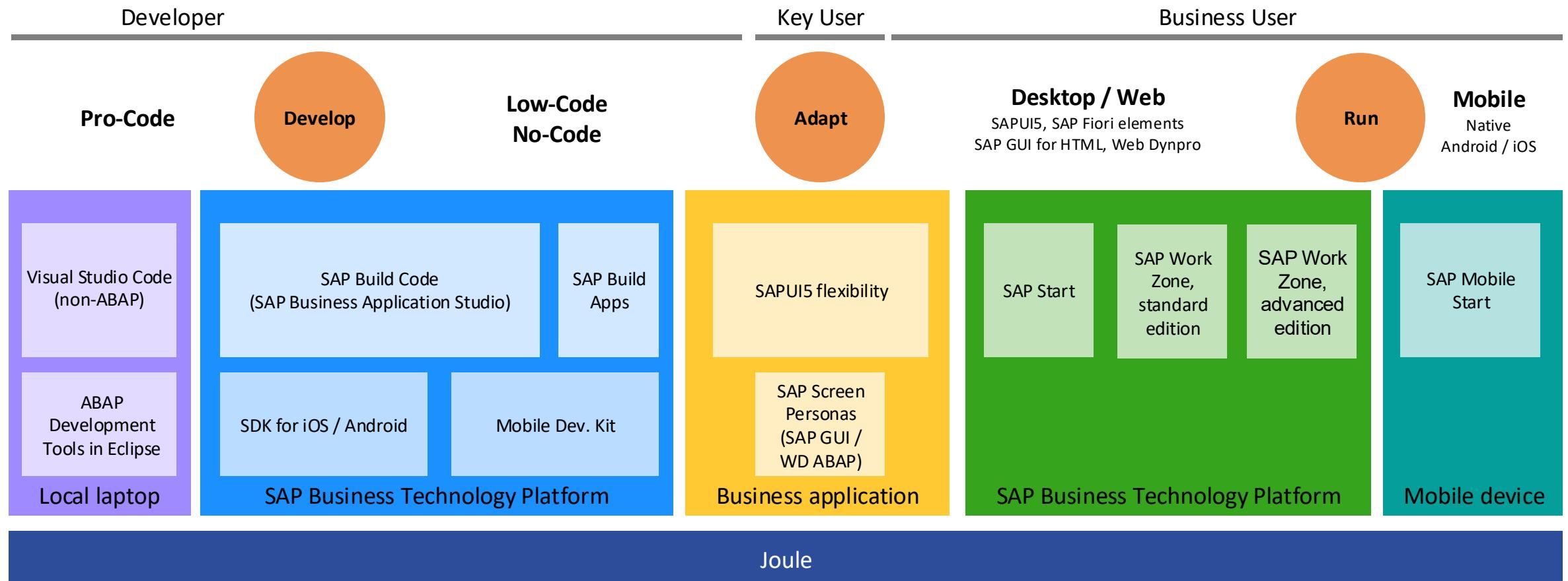
SAP Mobile Start: Native entry point

[SAP Mobile Experience community](#)

[Video: SAP Mobile Start – Your native entry point to the Intelligent Enterprise](#)



SAP Fiori user experience enabled by technologies & tools



Thank you.

... and don't panic!

Contact information:

Oliver Graeff

Director – Product Management, SAP



E-mail: oliver.graeff@sap.com

LinkedIn: [olivergraeff](https://www.linkedin.com/in/olivergraeff)

Questions?

