# SAP BTP Development Approach

We'll explore the complete development lifecycle: Connect, Build, Access, Deploy, and Operate. Each step plays a critical role in creating successful applications on the SAP BTP platform, and understanding this framework will provide you with a solid foundation for your development projects.

**por Mayko Silva**

# Connect: Linking Systems and Understanding APIs

1

## System Connection

Link SAP BTP with back-end systems like SAP S4 HANA or SuccessFactors through the SAP BTP cockpit under System Landscape Systems. Generate an integration token in BTP and use it in your solution to establish connectivity.

2

## API Discovery

Explore and understand available APIs and events through the SAP Business Accelerator Hub at api.sap.com. This platform serves as your comprehensive resource for all allow-listed APIs and events, with testing capabilities similar to Swagger.

3

## Event Handling

Leverage SAP Event Mesh and SAP Integration Suite, Advanced Event Mesh for event-driven architectures. These tools support the Cloud Event Standard, providing robust capabilities for event processing and distribution.

# Build: Development Tools and Approaches

### SAP Build Apps

A low-code and no-code solution designed for business users to create functional applications without requiring deep technical expertise. Perfect for business analysts and citizen developers who need to rapidly prototype and deploy solutions.

### SAP Business Application Studio

A comprehensive development environment supporting SAP Cloud Application Programming Model, SAP Fiori Elements, SAPUI5, Mobile Development Kit, and SAP HANA Base Development. Features visual editors, command-line functionality, and integration with code repositories like GitHub.

### Alternative Tools

Microsoft's Visual Studio Code for local development preferences, and Apache Eclipse with ABAP Development Tools extension for ABAP development. These provide flexibility for developers with specific tooling preferences.

# Build: Programming Models

## SAP Cloud Application Programming Model

Designed for open source technologies like Java and Node.js, this model provides a framework for developing stateless cloud applications. It offers a structured approach to building enterprise-grade services and applications.

## ABAP RESTful Application Programming Model

Tailored for developers with ABAP expertise, this model enables the creation of modern, RESTful applications while leveraging existing ABAP skills and resources. It bridges traditional ABAP development with modern cloud architecture.

## Alternative Models

Developers can also utilize other open source frameworks like Spring Boot or pursue cloud native development in the SAP BTP Kyma runtime, providing flexibility to work with familiar technologies while benefiting from SAP BTP's infrastructure.


# Access: Providing Application Access

## Web Access

Develop full-stack applications using CDS and REST-based services. Implement user interfaces with SAP Fiori elements for standard patterns or custom SAP UI5 for tailored user experiences, ensuring consistent and responsive web access.

## Mobile Access

Utilize SAP Mobile Services as a proxy between mobile clients and back-end systems. Choose from front-end options including Mobile Development Kit, SAP BTP SDK for iOS or Android, or SAP Build apps to create responsive mobile experiences.

## Robotic Process Automation

Implement automation for repetitive tasks that integrate with core systems through SAP Build process automation. This reduces manual effort and increases efficiency for routine business processes.

# Deploy: Front-End and Back-End Options

## 1 Front-End Deployment

SAP Build WorkZone Standard Edition follows SAP Fiori Launchpad design principles for internal users, providing a familiar interface. For external-facing portals with more intuitive UX and collaboration features, SAP Build WorkZone Advanced Edition offers enhanced capabilities.

## 2 Back-End Deployment: Cloud Foundry

The SAP BTP Cloud Foundry environment supports microservices development using open source technologies, enabling scalable and flexible application architectures with polyglot persistence and service binding.

## 3 Back-End Deployment: ABAP Environment

SAP BTP ABAP environment provides a platform for ABAP-based developments, allowing organizations to leverage existing ABAP expertise while benefiting from cloud capabilities and integration.

## 4 Back-End Deployment: Kyma Runtime

SAP BTP Kyma runtime enables cloud-native microservices development with Kubernetes, offering containerization, serverless functions, and event-based programming for modern application architectures.

# Operate: Managing Your Applications

## SAP BTP Cockpit

Your main administrative interface for comprehensive management of users, roles, deployments, connectivity, security, and more. The cockpit provides a centralized location to monitor and control all aspects of your SAP BTP environment.

## Command Line Interfaces

Both Cloud Foundry and Kyma Runtimes offer robust CLI support for automating administrative tasks. These command-line tools are especially valuable in DevOps pipelines, enabling scripted and automated operations.

## Adapt to Project Needs

While this development approach is comprehensive, it's designed to be flexible. You can adapt the framework to fit your specific project requirements, focusing on the elements most relevant to your use case.

This completes our overview of the SAP BTP development approach. Remember that while the process may seem complex initially, mastering these steps provides a solid foundation for successful development on the platform.