SAP BTP Runtimes and Architecture Patterns

por Mayko Silva

SAP BTP Runtime Fundamentals

SAP Business Technology Platform (BTP) provides two powerful runtime environments that serve as the foundation for your applications: Cloud Foundry and Kyma.

These environments offer different approaches to application deployment and management, giving developers flexibility in how they build and scale their solutions. Understanding the strengths of each runtime is essential for optimizing your development strategy and leveraging the full potential of the SAP BTP.

Cloud Foundry

An open-source platform-as-a-service that abstracts away infrastructure concerns, allowing developers to focus purely on application code.

Kyma

A Kubernetes-based environment that provides advanced containerization capabilities for complex, cloud-native applications.

Runtime Selection

Choosing between these runtimes depends on your application complexity, integration needs, and development priorities.

Cloud Foundry: Your Application's General Contractor

Cloud Foundry functions like a general contractor for your applications, handling the infrastructure details so you can focus on building your solution. This open-source platform-as-a-service has a robust community of developers continuously improving its capabilities.

One of Cloud Foundry's greatest strengths is its flexibility. You can deploy it across multiple cloud providers including AWS, Azure, and Google Cloud, giving you freedom from vendor lock-in and the ability to leverage your preferred cloud infrastructure.

Open-Source Community

Benefit from continuous improvements and innovations from a global developer community.

Multi-Cloud Flexibility

Deploy your applications across AWS, Azure, Google Cloud, and other providers without modification.

Developer Focus

Concentrate on application logic while Cloud Foundry manages the underlying infrastructure.

Cloud Foundry Components and Deployment

Cloud Foundry excels in deployment automation, efficiently handling the process of setting up your application in its new environment. This automation significantly reduces deployment complexity and potential errors.

The platform consists of several key components working together to support your applications. Each component plays a specific role in ensuring your application runs smoothly and securely in the cloud environment.

1

Router

Directs incoming traffic to the appropriate applications, functioning as the front door to your services.

2

Authentication

Ensures secure access by verifying user identities and permissions before allowing interaction with applications.

3

Application Lifecycle

Manages the staging, starting, and stopping of applications, keeping everything running optimally.

4

App Storage and Execution

Provides the environment where your application lives and operates, handling data persistence and runtime execution.

Cloud Foundry Limitations

While Cloud Foundry offers numerous advantages, understanding its limitations is crucial for effective planning. These constraints help determine whether Cloud Foundry is the right choice for your specific application needs.

Being aware of these limitations upfront allows you to design your architecture appropriately or consider alternative approaches when necessary. In some cases, these limitations might lead you to explore Kyma as an alternative runtime.

1 **No Docker-based Deployments**

Cloud Foundry uses its own containerization technology rather than supporting Docker containers directly, which may limit compatibility with Docker-centric workflows.

2 **Limited TCP Routing**

The platform has restrictions on TCP routing capabilities, which can impact applications requiring specific network protocols beyond standard HTTP/HTTPS.

3 **Resource Constraints**

Applications face certain memory and processing limitations that might affect performance for resource-intensive workloads.

Introducing Kyma: Application Superpowers

Kyma represents the next evolution in cloud application platforms, built on the powerful foundation of Kubernetes. This runtime environment gives your applications enhanced capabilities for complex, distributed architectures.

At its core, Kyma leverages containerization—a technology that packages applications with all their dependencies, making them portable and consistent across different environments. This container-based approach provides greater isolation, scalability, and resource efficiency.

Kubernetes Foundation

Built on the industry-standard container orchestration system

Container Technology

Packages applications with all dependencies for consistent deployment

Microservices Support

Optimized for building distributed application architectures

Enhanced Scalability

Efficiently scales individual components based on demand

Kyma's Advanced Features

Kyma extends the container concept with powerful features that enhance application connectivity, scalability, and management. The platform provides sophisticated tools for building modern, cloud-native applications.

One of Kyma's standout capabilities is its support for serverless architecture, allowing applications to dynamically allocate resources as needed. This approach optimizes resource utilization and can significantly reduce operational costs.

1

Application Connector

Securely connects your applications with external systems and services, enabling seamless integration across your technology landscape.

2

Serverless Computing

Automatically scales resources up or down based on demand, eliminating the need to provision and manage servers manually.

3

Event-Based Architecture

Facilitates communication between components through events, creating loosely coupled systems that are easier to maintain and extend.

4

API Management

Provides tools for creating, publishing, and managing APIs, enabling controlled access to your application services.

Choosing Between Cloud Foundry and Kyma

Selecting the right runtime environment depends on your specific application requirements and development priorities. Each platform offers distinct advantages that make it suitable for different scenarios.

Cloud Foundry excels in simplicity and rapid development, while Kyma provides greater flexibility and control for complex applications. Your choice should align with your technical needs, team expertise, and long-term application strategy.

When to Choose Cloud Foundry

- Straightforward applications with standard requirements
- Projects requiring rapid development and deployment
- Teams new to cloud-native development
- Applications with minimal container orchestration needs

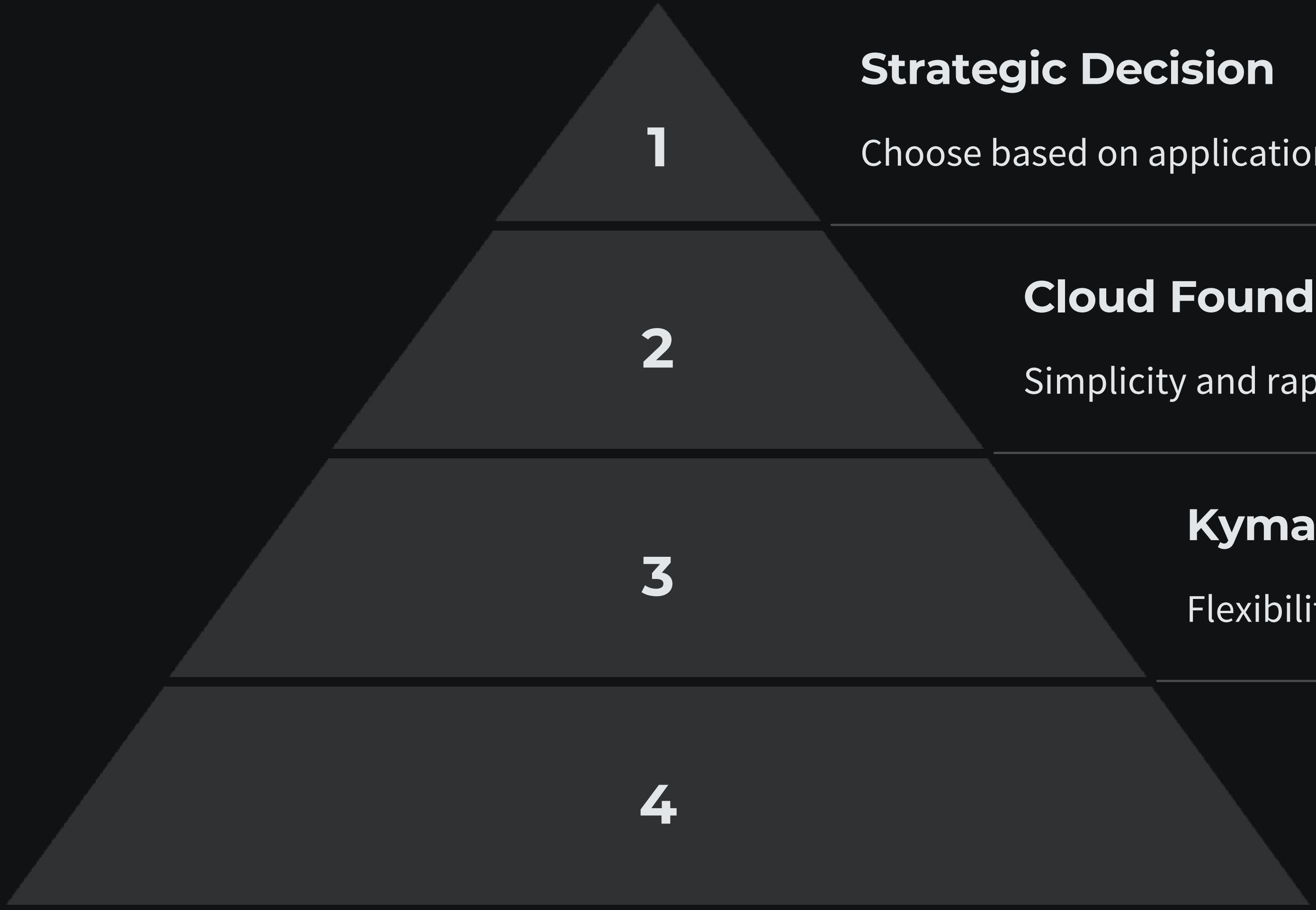
When to Choose Kyma

- Complex applications with multiple components
- Systems requiring independent scaling of components
- Projects leveraging microservices architecture
- Applications needing advanced container orchestration

SAP BTP Runtime Summary

SAP Business Technology Platform provides two powerful runtime environments that cater to different application development approaches. Understanding the strengths of each runtime is essential for maximizing the value of your SAP BTP investment.

Think of Cloud Foundry as your reliable, easy-to-use toolkit for building standard applications, while Kyma serves as your advanced workshop for creating complex, cloud-native solutions. Both have their place in a comprehensive development strategy.



Strategic Decision

Choose based on application needs

Cloud Foundry

Simplicity and rapid development

Kyma

Flexibility and cloud-native power

SAP BTP Foundation

Unified platform for both runtimes

Key Takeaways and Next Steps

Understanding SAP BTP runtimes is fundamental to developing effective cloud solutions. Cloud Foundry and Kyma each offer distinct advantages that can be leveraged based on your specific application requirements.

As you move forward with your SAP BTP projects, consider evaluating your application needs against the capabilities of each runtime. This thoughtful approach will help ensure you select the most appropriate environment for your development goals.

Assess Your Application Needs

Evaluate complexity, integration requirements, and scalability needs to determine which runtime best suits your project.

Build Expertise in Both Runtimes

Develop skills across both Cloud Foundry and Kyma to maintain flexibility in your development approach.

Start with Proof of Concepts

Test your applications in both environments to gain practical experience with their strengths and limitations.

Implement Best Practices

Apply architecture patterns appropriate to your chosen runtime to maximize performance and maintainability.