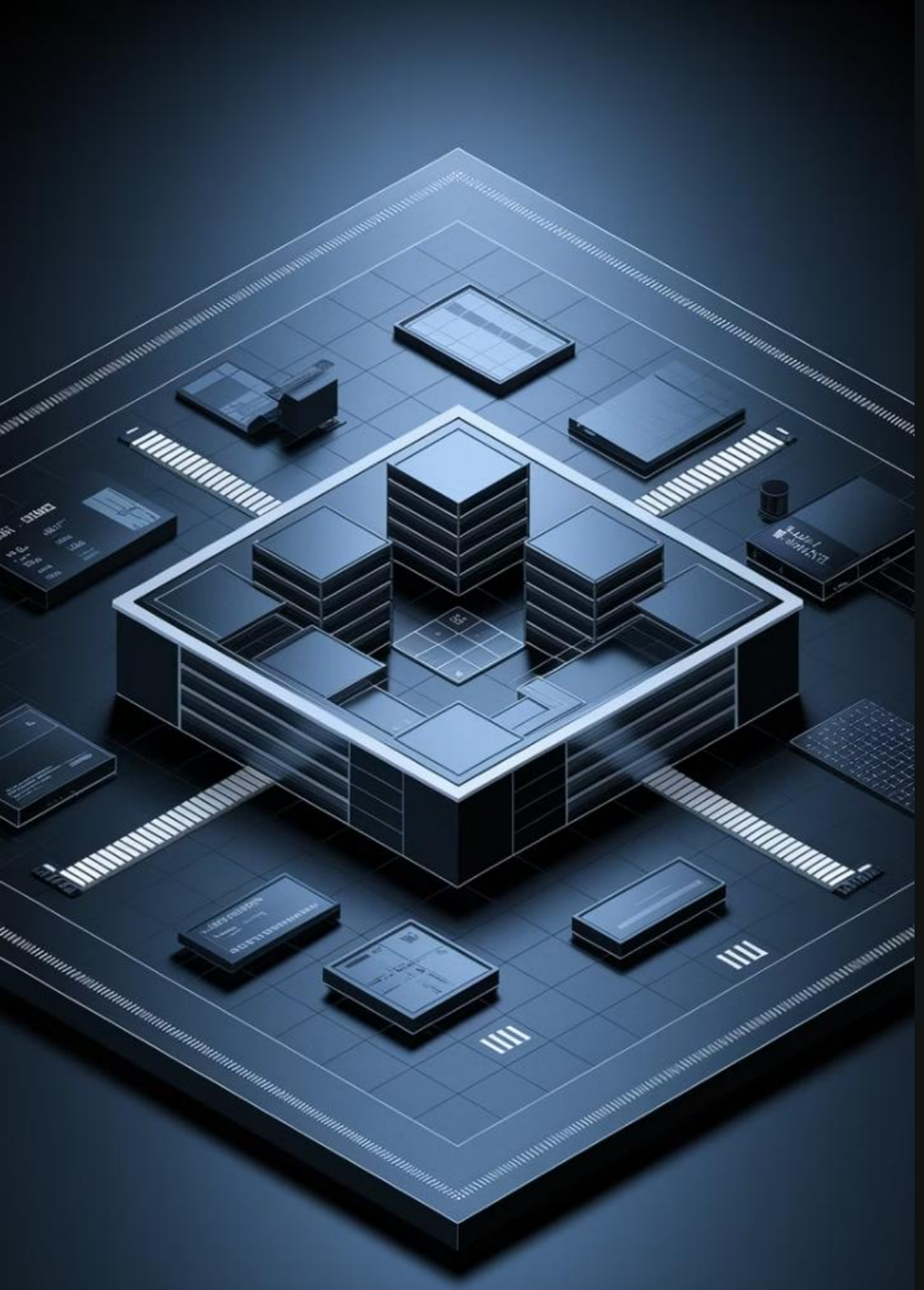
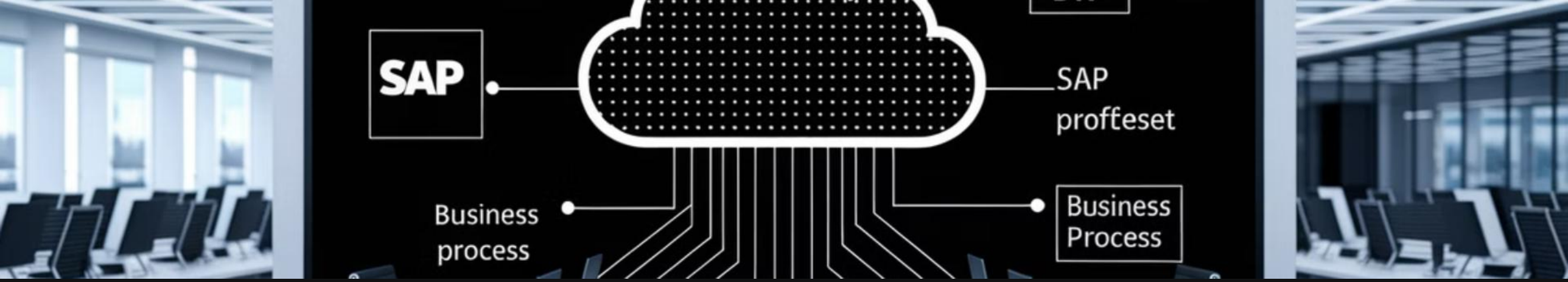


Architecture Design Patterns and Development Approaches in SAP BTP

Good architecture is the bedrock of any successful technology project. Without a solid architectural foundation, even the most promising SAP BTP initiatives can falter. Throughout this presentation, we'll examine key perspectives, proven patterns, and practical approaches to building robust SAP BTP solutions.

por Mayko Silva



Introduction to SAP BTP Architecture

In our previous sessions, we explored SAP BTP's powerful services that drive digital transformation. These capability pillars - application development, automation, integration, data and analytics, and AI - provide the tools organizations need to transform IT systems and reimagine business processes.

The bimodal IT approach we discussed helps organizations maintain agility while ensuring stability in core business operations. However, when embarking on cloud or hybrid transformation journeys, your development approach must follow strong architecture principles.

Digital Transformation

SAP BTP services enable comprehensive digital transformation across the enterprise.

Bimodal IT

Two-speed approach balancing agility with stability in core operations.

Architecture Framework

Solid architecture principles provide the foundation for successful cloud implementations.

Three Key Perspectives in SAP BTP Architecture

Architecture in SAP BTP must be viewed through multiple lenses to ensure comprehensive coverage of all aspects. Each perspective brings unique insights and considerations to the architectural process.

By examining architecture from these three viewpoints, we create a more holistic approach that addresses technical requirements, project governance, and implementation realities.

Architect's Perspective

Focuses on core principles, frameworks, and enterprise architecture alignment.

Project Manager's Perspective

Emphasizes implementation methodology and governance frameworks.

Developer's Perspective

Concentrates on coding standards, patterns, and practical implementation.

The Architect's Perspective

From the architect's viewpoint, understanding core architecture principles and frameworks is essential for successful SAP BTP development. Your BTP architecture must integrate seamlessly with the broader enterprise architecture landscape.

Creating comprehensive documentation, diagrams, and specifications ensures smooth transitions to the cloud. As experienced in a manufacturing company project, investing time in documenting system interactions can prevent significant integration challenges later.

Core Principles

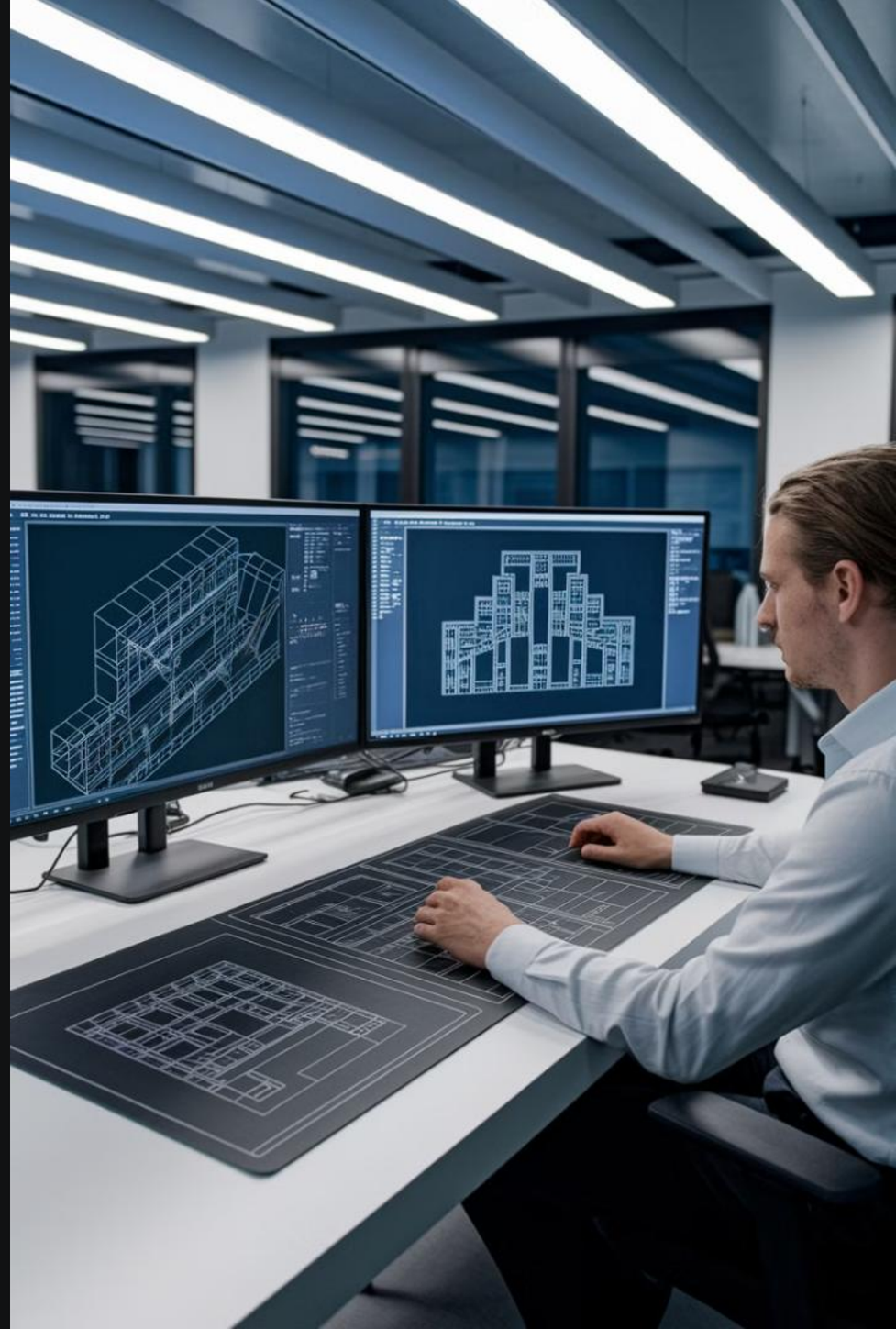
Establish fundamental architectural guidelines that align with enterprise standards.

Enterprise Alignment

Ensure BTP architecture fits within the broader enterprise architecture landscape.

Key Artifacts

Create comprehensive documentation to facilitate smooth cloud transitions.



Common Architecture Design Patterns

Architecture design patterns provide proven solutions to recurring design challenges in cloud and hybrid landscapes. These patterns have gained popularity across industries because they effectively address common architectural problems.

Understanding these patterns helps architects select the most appropriate approach for specific business requirements and technical constraints. The right pattern can significantly impact scalability, maintainability, and overall system performance.

Identify Business Requirements

Understand the specific needs and constraints of your organization.

Evaluate Design Patterns

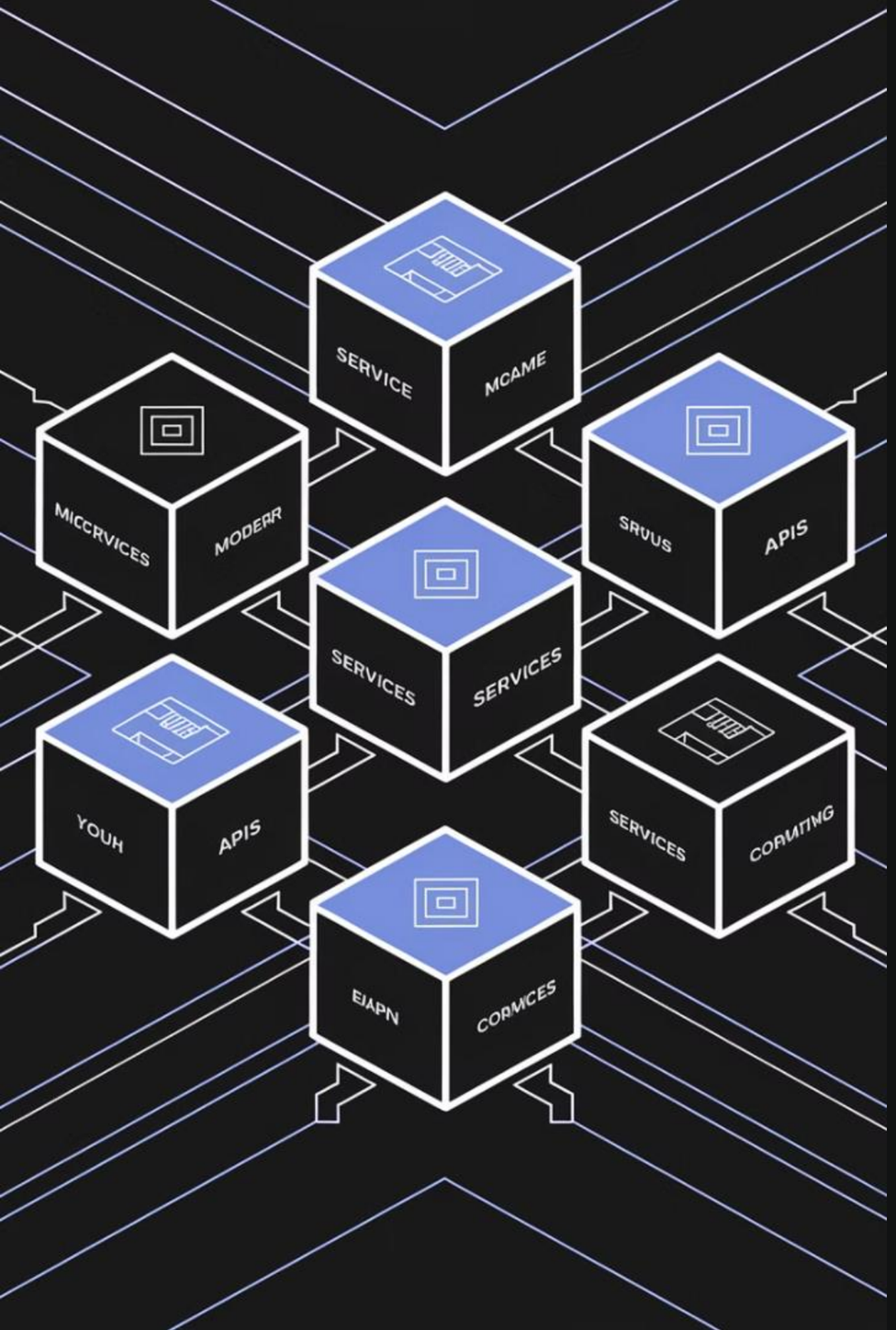
Assess which patterns best address your particular challenges.

Apply Selected Patterns

Implement the chosen patterns within your SAP BTP architecture.

Review and Refine

Continuously evaluate and improve your architectural approach.



Microservices Architecture

Microservices architecture breaks down applications into small, independent services, each performing a specific business function. This approach enables teams to deploy and scale components independently, significantly increasing development agility.

A retail client demonstrated the power of this approach when they transitioned from a monolithic e-commerce platform to microservices on SAP BTP. This transformation accelerated their release cycle from quarterly updates to weekly feature releases, as teams could work on different services without interference.

1

Decomposition

Break application into small, focused services with clear boundaries.

2

Independence

Enable autonomous development, deployment, and scaling of services.

3

Specialization

Allow teams to focus on specific business functions.

4

Resilience

Improve system stability through isolation of failures.

API-First Design

API-first design prioritizes creating well-defined interfaces between systems before implementing the underlying functionality. This approach facilitates seamless integration across different platforms and supports building robust partner ecosystems.

A logistics company exemplified this approach by exposing their shipping services as APIs through SAP BTP. This strategy enabled hundreds of small businesses to integrate shipping directly into their websites without complex EDI implementations, creating new business opportunities and improving customer experience.

1

Interface Definition

Create clear, consistent API contracts before implementation begins.

2

Platform Independence

Enable communication between diverse systems regardless of underlying technology.

3

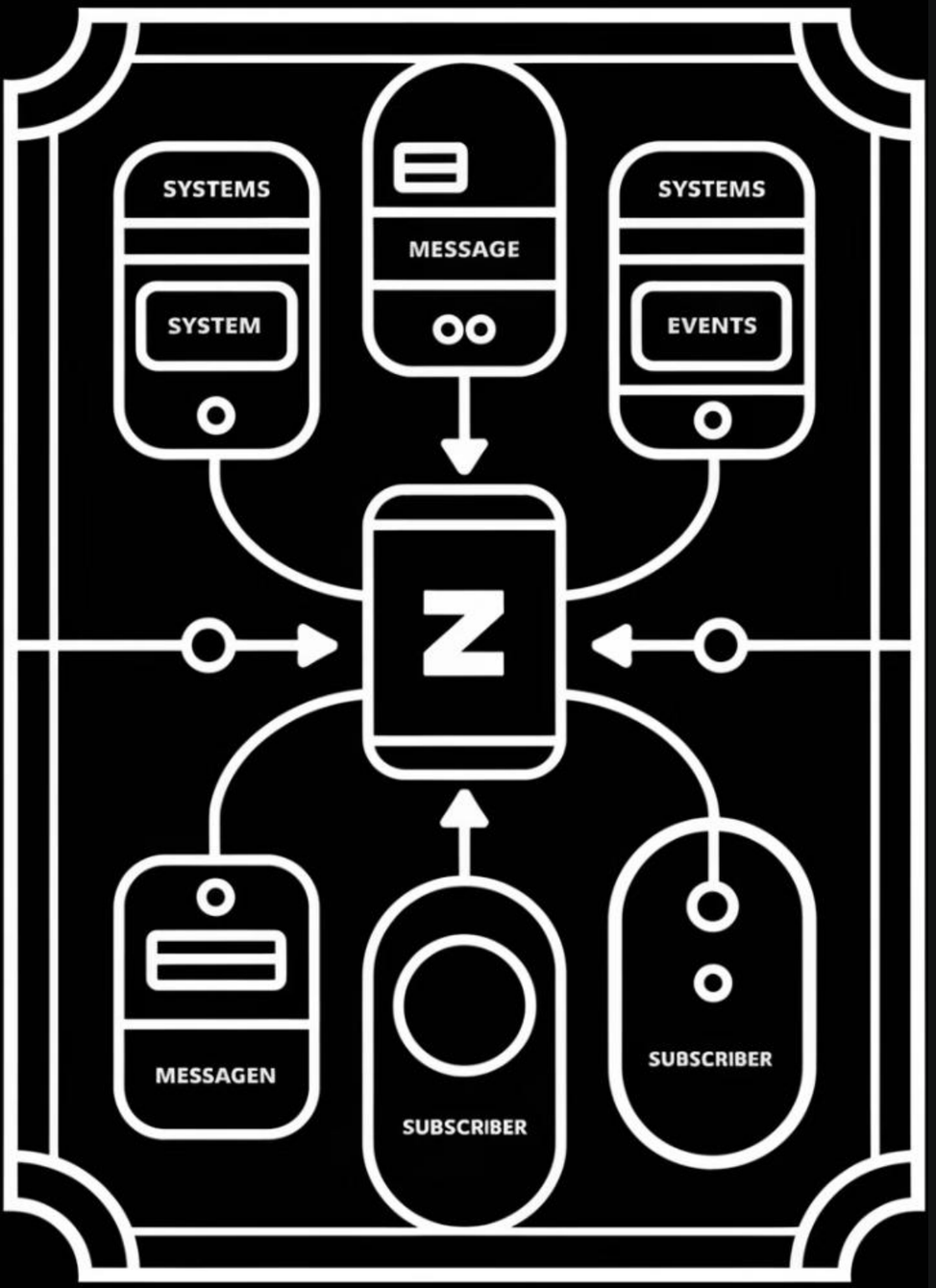
Developer Experience

Focus on creating intuitive, well-documented APIs that are easy to consume.

4

Ecosystem Enablement

Facilitate partner integration and extend business capabilities through standardized interfaces.



Event-Driven Architecture

Event-driven architecture enables systems to communicate through events rather than direct requests. This approach reduces coupling between components and significantly improves scalability by allowing asynchronous processing of system events.

A large bank implemented this architecture on SAP BTP with remarkable results. During a major promotion that increased transaction volume by 400%, their event-driven system handled the load without performance issues, demonstrating the pattern's effectiveness for high-scale scenarios.

1. **Event Production**

   System components publish events when significant state changes occur.

2. **Event Distribution**

   Event broker routes notifications to interested subscribers.

3. **Event Consumption**

   Subscribers process events asynchronously based on business logic.

4. **System Reaction**

   Components respond to events without direct coupling to event sources.

The Project Manager's Perspective

From the project manager's viewpoint, implementation methodology provides the essential framework for project governance. Project managers must align technical architecture with business objectives while establishing processes that maintain architectural integrity throughout the project lifecycle.

Weekly architecture review sessions, as implemented in a pharmaceutical company's SAP BTP project, help identify potential issues early before they become expensive problems. This proactive approach to architectural governance is critical for project success.

1

Align Objectives

Ensure technical architecture supports business goals and delivers expected value.

2

Establish Governance

Create processes to maintain architectural integrity throughout implementation.

3

Balance Priorities

Manage the tension between delivery speed and solution quality.

4

Coordinate Teams

Facilitate collaboration across teams working on interconnected components.

The Developer's Perspective

For developers, architecture principles translate into practical coding standards, patterns, and practices. Understanding how to implement selected architecture patterns is essential for creating maintainable, high-quality solutions on SAP BTP.

Experience shows that developers who understand the architectural vision build components that integrate smoothly, while those who ignore architecture guidelines often create modules requiring complete rebuilds later. This highlights the importance of architectural alignment at the development level.

Implementation Knowledge

Understanding how to translate architecture patterns into working code.

Service Selection

Identifying which SAP BTP services best support chosen architecture patterns.

Best Practices

Following established guidelines for creating maintainable, testable code.

Implementing Bimodal IT with SAP BTP

SAP BTP provides the ideal platform for implementing bimodal IT, balancing stability in core systems with agility in innovation areas. This approach allows organizations to maintain reliable business operations while rapidly developing new capabilities to address market opportunities.

A manufacturing client successfully applied this strategy, maintaining stable core production systems while quickly developing customer-facing applications that evolved based on market feedback. This dual-mode approach maximized both operational reliability and innovation speed.

Mode 1: Stability

Use SAP BTP integration services to connect with core ERP systems, ensuring reliable operations and data consistency. This mode focuses on predictability, security, and maintaining critical business processes with minimal disruption.

Mode 2: Agility

Leverage BTP's rapid development tools to quickly build innovative applications that respond to market opportunities. This mode emphasizes speed, experimentation, and continuous delivery to address evolving business needs and customer expectations.

Development Approach for SAP BTP Projects

A comprehensive development approach for SAP BTP projects begins with business objectives rather than technology considerations. This ensures that technical solutions directly address organizational needs and deliver measurable value.

Documenting architecture decisions, creating reusable patterns, establishing governance processes, and conducting regular reviews are essential practices that maintain architectural integrity throughout the development lifecycle.

Business Objectives

Start with clear business goals rather than technology.

Regular Review

Continuously evaluate and adapt as requirements evolve.

Architecture Documentation

Record decisions and rationale for future reference.

Pattern Creation

Develop reusable solutions for common scenarios.

Governance Implementation

Establish processes to maintain architectural integrity.

Iterative Implementation Approach

The most successful SAP BTP projects follow an iterative implementation approach. This methodology begins with establishing a solid architectural foundation, then proceeds through small, manageable increments with frequent reviews and adjustments.

This approach allows teams to deliver value quickly while maintaining architectural integrity. By breaking large initiatives into smaller components, organizations can better manage complexity, reduce risk, and adapt to changing requirements throughout the implementation process.

1

Architectural Foundation

Establish core principles and frameworks.

2

Incremental Development

Build in small, manageable components.

3

Regular Reviews

Continuously evaluate progress and alignment.

4

Adaptive Refinement

Adjust approach based on feedback and results.

Conclusion

Architecture design patterns provide the essential structure for successful SAP BTP projects. When properly understood and applied, these patterns create solutions that are both robust and flexible—meeting current needs while adapting to future challenges.

Remember that architecture isn't merely a technical exercise but a key component of project governance. Effective architecture ensures your technical work aligns with business goals and delivers lasting value. By integrating the perspectives of architects, project managers, and developers, you create a comprehensive approach that addresses all aspects of SAP BTP implementation.