

# Understanding the Three-Tier Model in SAP Extensions

Throughout this presentation, we'll explore what the Three-Tier Model is, why it matters, and how to implement it effectively in your organization. We'll also cover best practices and resources for further learning.

Let's begin our journey into understanding how this model can help balance upgrade stability with cloud compatibility while maintaining existing functionality.



por Mayko Silva

# What is the Three-Tier Model?

## Guidance for Cloud Migration

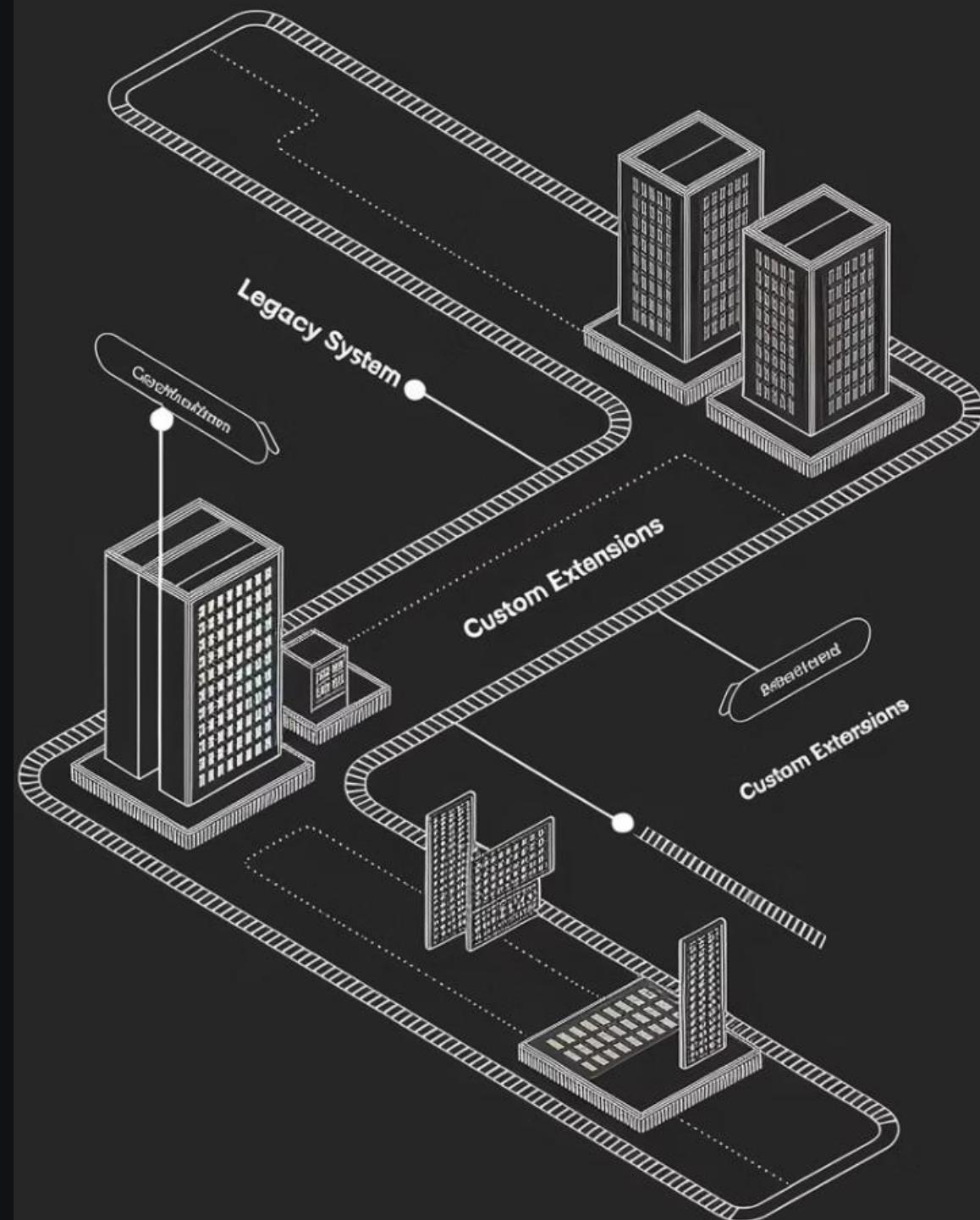
The Three-Tier Model is SAP's guidance to help customers who have been running their SAP ERP systems on-premise for years and now want to move to the cloud.

## Managing Legacy Extensions

Organizations that have used SAP systems for decades with numerous custom extensions need a clear path for handling those legacy extensions when moving to the cloud.

## Bridging API Gaps

This model becomes especially important when publicly released SAP APIs don't cover all the functionality you need, helping to bridge that gap effectively.



# The Three Tiers Explained: Tier 1

## 1 Cloud Development

This is where you want to be whenever possible. All new applications and extensions must follow cloud readiness standards, with the ABAP Cloud development model being mandatory.

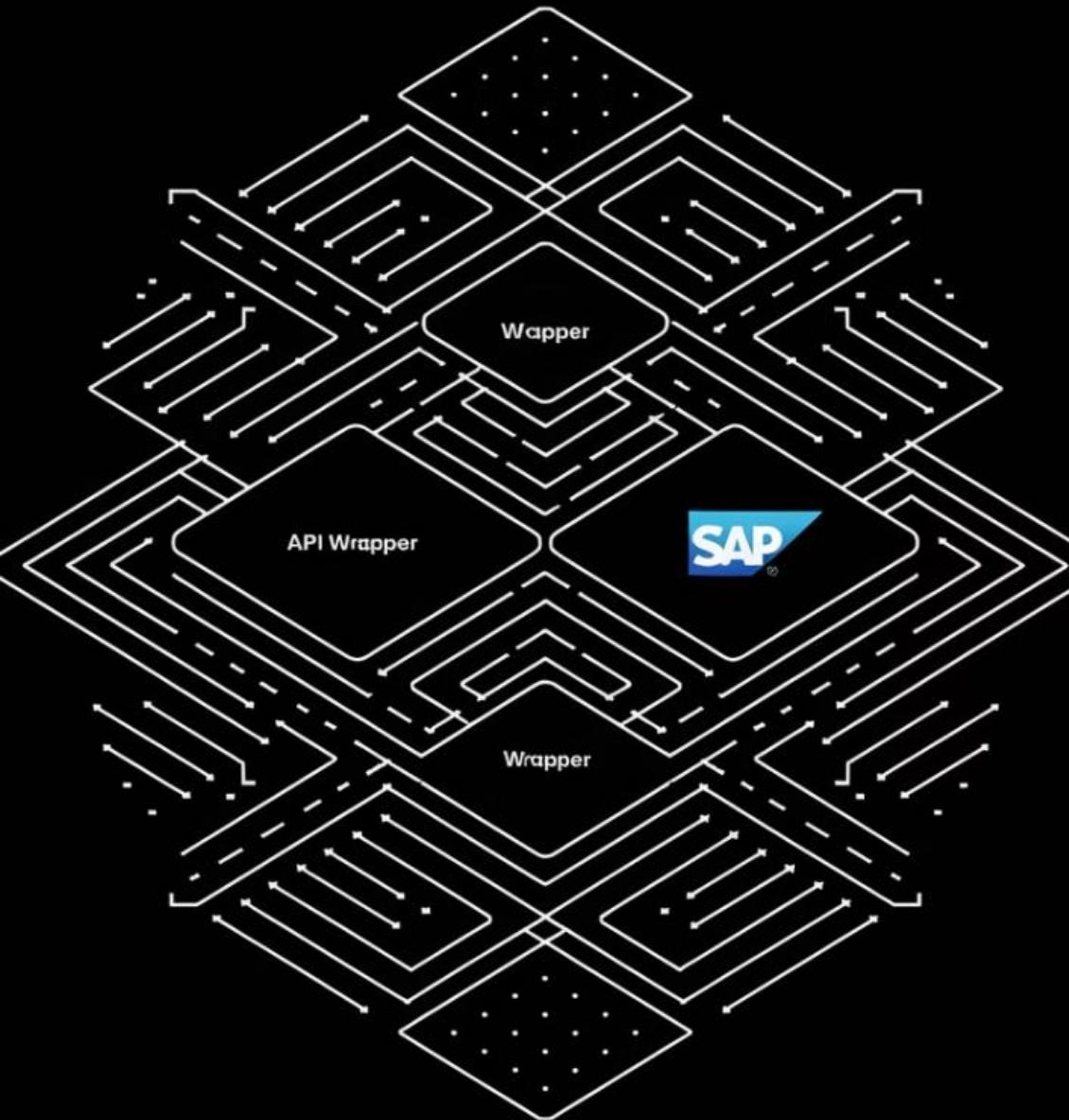
## 2 Enforced Compliance

Compliance is enforced through syntax and runtime checks that cannot be bypassed - they're there to protect your system's integrity.

## 3 Extension Approaches

For Tier 1 extensions, you have two main approaches: key user extensibility and developer extensibility. These ensure your extensions remain stable during upgrades and compatible with cloud environments.





# The Three Tiers Explained:

## Tier 2



### Cloud API Enablement

Tier 2 is a middle ground that enables Tier 1 extensions when you face certain limitations in available APIs.

### Custom Wrapper Development

You create custom wrapper development for unreleased SAP objects, which can then be consumed in your Tier 1 extensions.

### Transitional Solution

It's a bridge solution until SAP releases official APIs. You should plan to eventually retire these custom wrappers once SAP releases official APIs for the functionality you need.

# The Three Tiers Explained: Tier 3

## Legacy Development

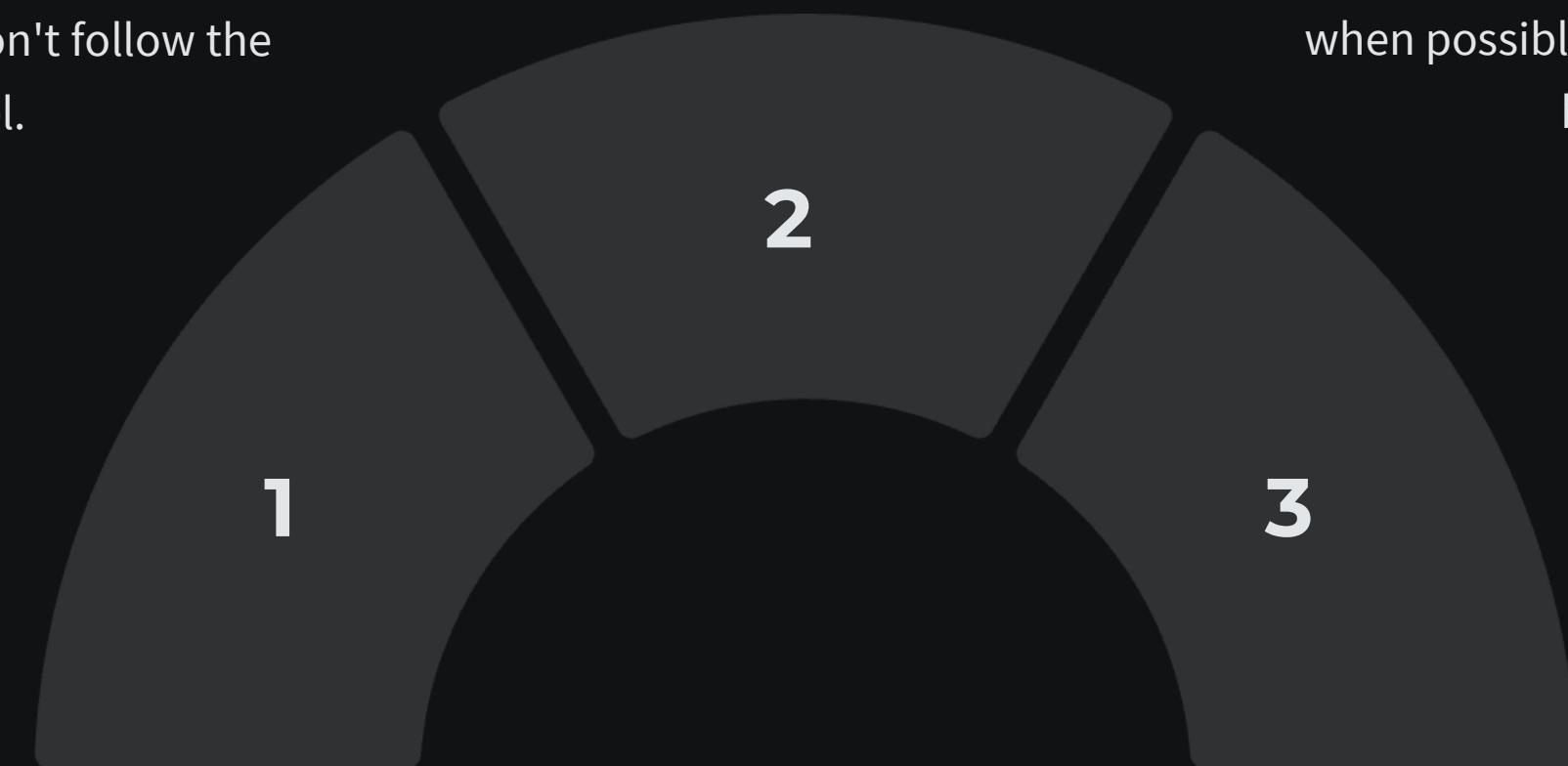
Tier 3 is the last resort, including legacy ABAP extensions that don't follow the cloud model.

## Non-Compliant Extensions

This includes new extensions that for some reason cannot adhere to the ABAP Cloud development model.

## Tightly Coupled Code

Extensions that require classical ABAP due to tight coupling with core systems fall into this tier. This should be avoided when possible, but sometimes you might have no choice.



# Why This Model Matters



## Balance

The Three-Tier Model helps you balance upgrade stability with cloud compatibility, ensuring your systems remain functional while moving forward.



## Roadmap

It creates a clear roadmap for modernizing your extensions, giving you direction on how to evolve your custom development over time.



## Decision Framework

This model provides a framework for making informed decisions about where to place new development, helping prioritize cloud-ready approaches.



# Visual Structure

## Tier 1: Cloud-Ready Extensions

Loosely coupled, upgrade-stable

---

## Tier 2: Bridge Solutions

API wrappers, transitional

---

## Tier 3: Legacy Extensions

Tightly coupled, classical ABAP

The Three-Tier Model organizes extensions in a visual hierarchy where you want as much as possible at the top (Tier 1), and as little as possible at the bottom (Tier 3). This pyramid structure helps visualize your extension portfolio and set clear goals for modernization.

# Implementation Strategy

## Classify Existing Extensions

Start by categorizing all your current custom developments according to the three tiers to understand your starting point.

## Prioritize Tier 1 for New Development

For any new extensions, always attempt to implement at Tier 1 first, only dropping to lower tiers when absolutely necessary.

## Document Tier 2 Cases

When you must use Tier 2 wrappers, document these cases thoroughly for future migration to official APIs.

## Minimize Tier 3

Keep Tier 3 extensions to an absolute minimum and create a plan to gradually move extensions up the tiers over time.





# Best Practices

1

## Document Wrappers

Create thorough documentation for all Tier 2 wrappers so they can be easily identified and retired when official APIs become available.

2

## Establish Governance

Develop clear governance rules around which tier new extensions should use, ensuring consistent application of the model.

3

## Regular Portfolio Review

Schedule periodic reviews of your extension portfolio to identify opportunities to move developments up the tiers as new APIs become available.

4

## Developer Training

Invest in training your developers on the ABAP Cloud development model to make Tier 1 the default approach for all new extensions.

# Resources for Further Learning



SAP provides detailed documentation and guidance on extending SAP S/4HANA in various environments. You can find resources about extending SAP S/4HANA with ABAP-based extensions, development guidelines for Tier 2 extensions, and technical use cases with recommended technologies for ABAP.

In our next session, we'll look more closely at the ABAP Cloud Development Model and how it supports clean core extensions. Thank you for your attention!