Programming Models in SAP BTP

In this presentation, we'll examine the foundation of these models, why they matter, and dive into the specifics of both the SAP Cloud Application Programming Model and the ABAP RESTful Application Programming Model.

We'll compare these models to help you understand which one might be best suited for your specific development needs and project requirements, highlighting their unique attributes and use cases in cloud development.

por Mayko Silva

Core Data Services (CDS) - The Foundation

CDS serves as the fundamental infrastructure that allows developers to transform databases into meaningful data models. Through these programming models, you can easily expose projected elements for consumption via different protocols.

While we won't explore CDS in great depth today, remember that it functions as the essential building block for both programming models we'll discuss. Think of CDS as your house's foundation - you may not think about it often, but without it, everything would collapse.

Data Modeling

CDS enables meaningful transformation of database structures into logical models

Foundation Element

Serves as the essential building block for SAP programming models

Projection Capability

Allows exposure of elements for consumption through various protocols

Why Programming Models Matter

In today's rapidly evolving technological landscape, particularly in UI design, the focus remains on delivering optimal user experiences. With numerous UI technologies available, SAP provides agnostic programming models to support application development for several compelling reasons.

These models ensure future-proofing, allowing new SAP innovations to integrate seamlessly with existing applications. They protect your investment - crucial for businesses investing millions in these systems. For developers, they accelerate development through specialized libraries and tools.

1 Future-Proofing

Ensures new SAP innovations can easily integrate with your existing applications

2 Investment Protection

Safeguards business investments in SAP systems worth millions

3 Development Acceleration

Provides libraries and tools that significantly speed up the development process

4 Focus on Business Logic

Handles technical aspects behind the scenes, allowing developers to concentrate on data modeling and business logic

SAP Cloud Application Programming Model

The SAP Cloud Application Programming Model, commonly referred to as CAP, is a comprehensive framework that provides languages, libraries, and tools for building enterprise-grade services and applications in SAP BTP.

This model offers a structured approach to application development, ensuring consistency and efficiency throughout the development lifecycle. CAP enables developers to create robust, scalable solutions that leverage the full potential of cloud technologies.

Comprehensive Framework

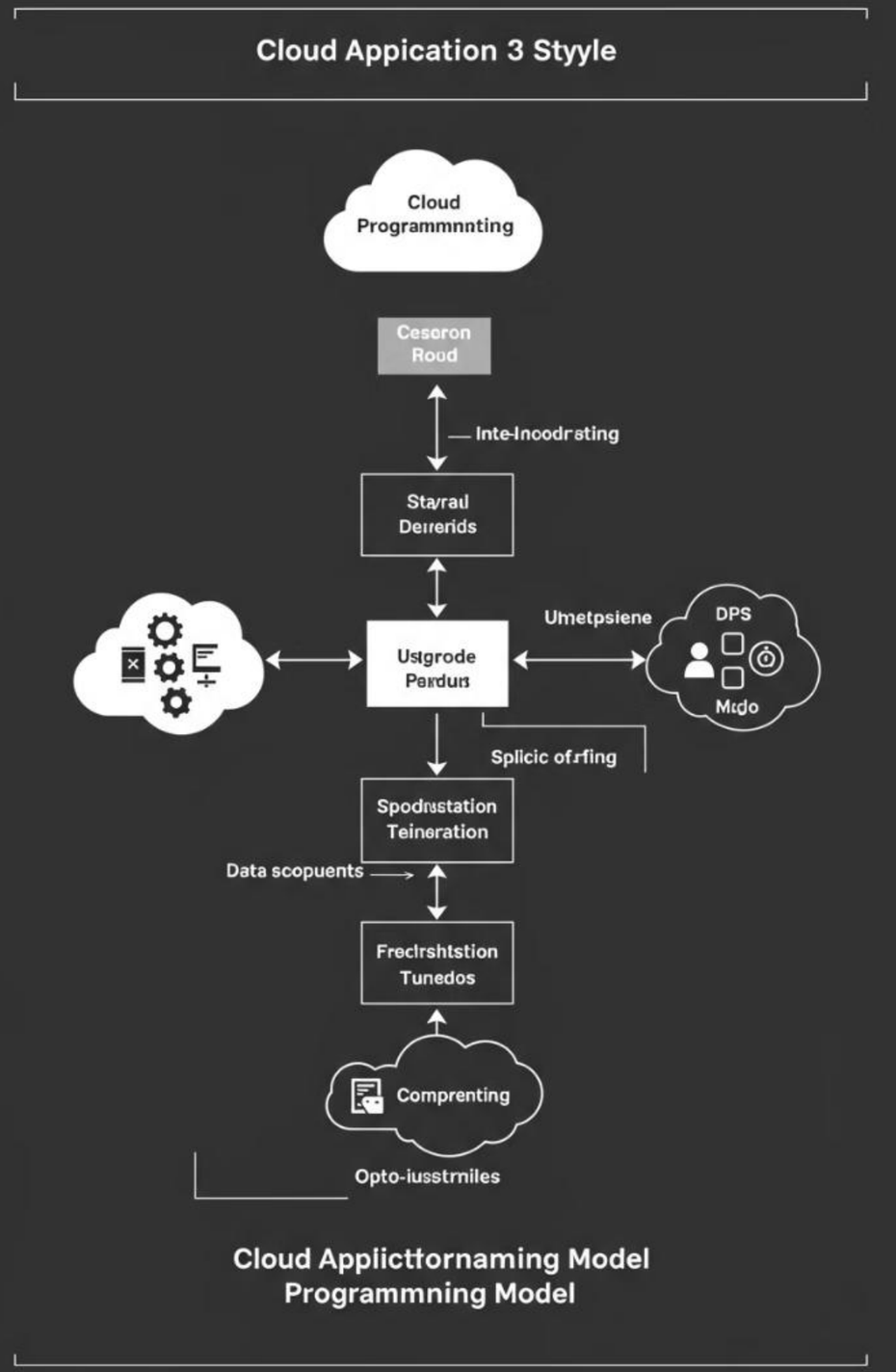
Provides a complete set of tools for enterprise application development

Cloud-Native

Designed specifically for cloud environments and SAP BTP

Developer-Friendly

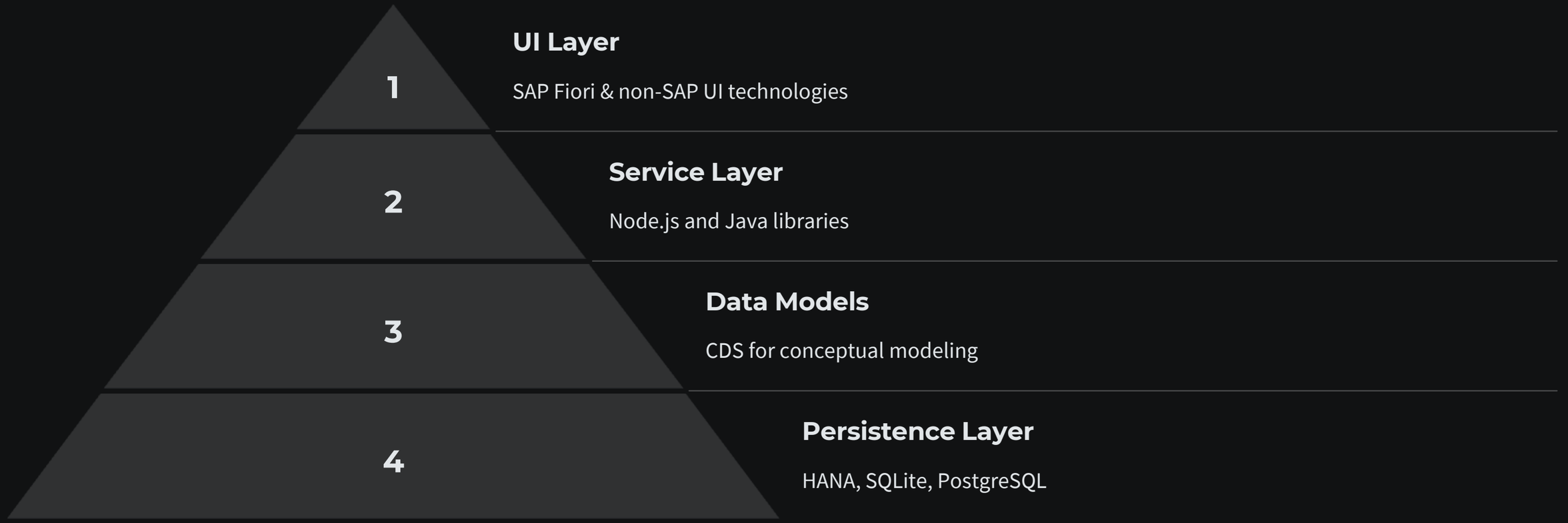
Offers languages, libraries, and tools that streamline the development process



Key Components of CAP

At the core of the SAP Cloud Application Programming Model is CDS, functioning as the backbone for building data models and service definitions at a conceptual level. For the UI Layer, it primarily utilizes SAP Fiori to maintain consistency in user experience, while also supporting non-SAP UI technologies when needed.

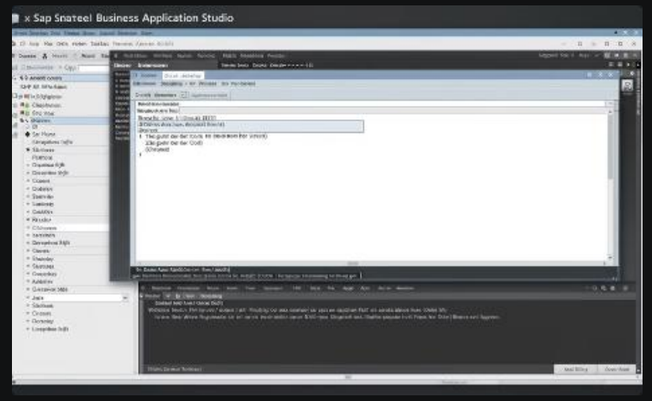
The Service Layer employs both Node.js and Java libraries to expose and consume services. For the Persistence Layer, CAP supports multiple database options including SAP HANA, SQLite, and PostgreSQL, providing flexibility in data storage solutions.



Development Environment for CAP

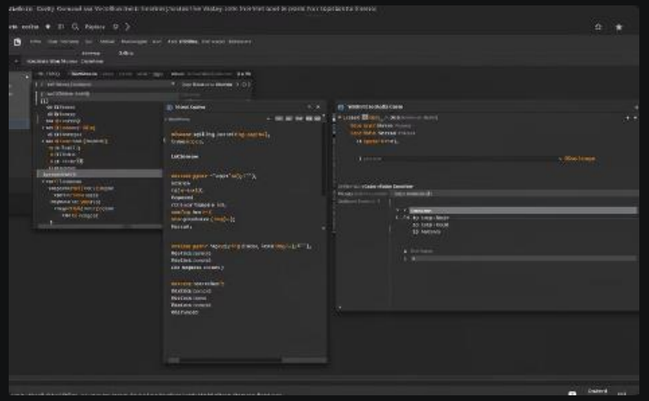
While SAP recommends using SAP Business Application Studio for cloud application development, developers also have the flexibility to use alternative environments such as Visual Studio Code for local development work.

This flexibility allows development teams to leverage familiar tools while still adhering to SAP's best practices and guidelines. Many developers appreciate the ability to work in their preferred environment, especially during the initial stages of application development.



SAP Business Application Studio

Recommended by SAP for cloud application development, offering integrated tools and services specifically designed for SAP technologies



Visual Studio Code

Popular alternative for local development, providing a familiar environment with extensions to support SAP development workflows

Flexible Development

Developers can choose their preferred environment while still adhering to SAP best practices and guidelines

Core Principles of CAP

SAP designed the Cloud Application Programming Model with three fundamental principles in mind. It's open - supporting both open-source and SAP technologies, allowing system architecture design based on specific needs. It's opinionated - providing guidelines, best practices, and out-of-the-box services to simplify development.

The model is also platform-agnostic - making applications independent from the underlying infrastructure, which prevents vendor lock-in. This is particularly valuable for companies concerned about being restricted to a single vendor. For detailed documentation on building with CAP, visit cap.cloud.sap/docs.

Open

Supports both open-source and SAP technologies, allowing flexible system architecture design

Opinionated

Provides guidelines, best practices, and out-of-the-box services to simplify and standardize development

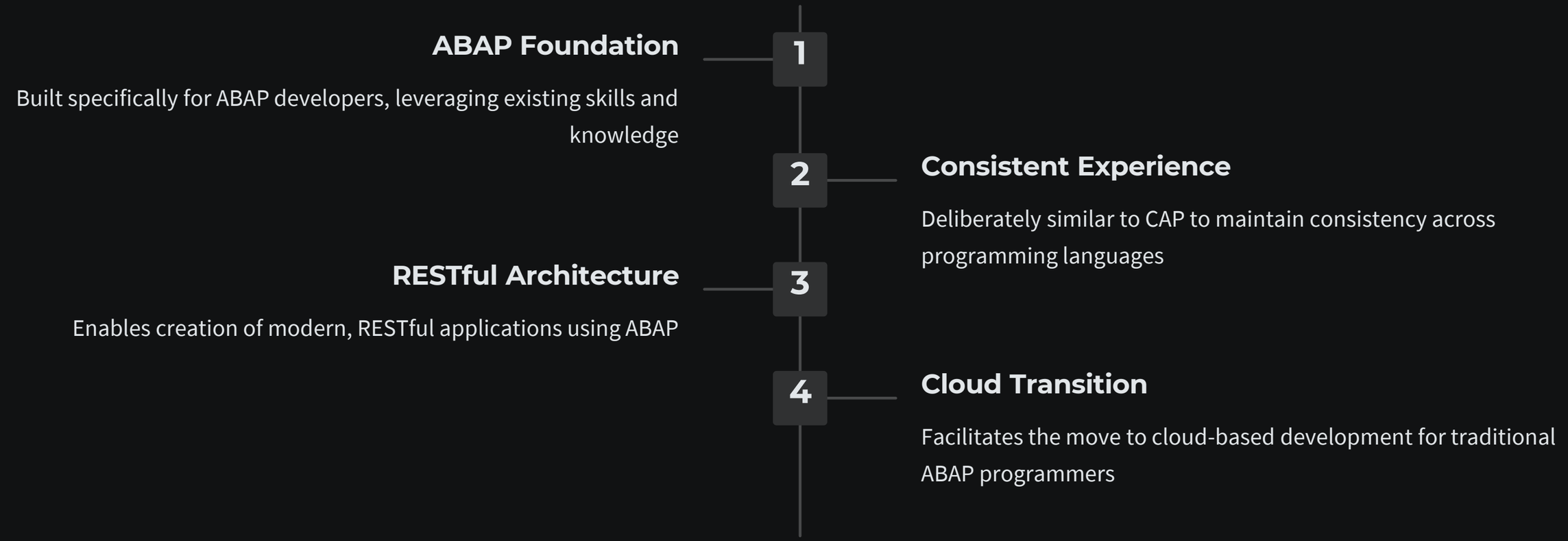
Platform-Agnostic

Makes applications independent from underlying infrastructure, preventing vendor lock-in

ABAP RESTful Application Programming Model

The ABAP RESTful Application Programming Model is specifically designed for ABAP developers. It shares significant similarities with the SAP Cloud Application Programming Model - a deliberate design choice to maintain consistency in the development experience regardless of the programming language being used.

This model enables ABAP developers to create modern, RESTful applications while leveraging their existing skills and knowledge of ABAP. The familiar structure helps streamline the transition to cloud-based development for traditional ABAP programmers.



Evolution of ABAP Development

ABAP development has undergone significant evolution over the years. Before ABAP 7.5, there was only best practice guidance for application development with various UI technologies like Web Dynpro, Floorplan Manager, and BSP - creating a somewhat inconsistent landscape.

ABAP 7.4 introduced CDS as a foundation for data modeling and optimizing SAP HANA capabilities. Early SAP Fiori presented challenges in reconciling frontend (SAPUI5), communication protocols (OData), and ABAP backend development. This led to the ABAP Programming Model for SAP Fiori, based on SAP Gateway, CDS, and BOPF, eventually evolving into today's streamlined ABAP RESTful Application Programming Model for SAP BTP.

Pre-ABAP 7.5

Only best practice guidance for development with various UI technologies (Web Dynpro, Floorplan Manager, BSP)

ABAP 7.4

Introduction of CDS as foundation for data modeling and optimizing SAP HANA capabilities

Early SAP Fiori

Challenges reconciling SAPUI5, OData protocols, and ABAP backend development

ABAP Programming Model for Fiori

Based on SAP Gateway, CDS, and Business Object Processing Framework (BOPF)

ABAP RESTful Application Programming Model

Modern, streamlined approach for SAP BTP ABAP environment

Comparative Review of Programming Models

The choice between SAP Cloud Application Programming Model and ABAP RESTful Application Programming Model depends on your background and project requirements. CAP runs on Node.js or Java servers, while the ABAP model runs on SAP NetWeaver ABAP stack. For databases, CAP supports SAP HANA, SQLite, and PostgreSQL, while the ABAP model is limited to SAP HANA.

Generally, the open-source community with Node.js and Java backgrounds prefers CAP, while businesses running ABAP-based solutions prefer the ABAP model. CAP excels in event-driven architecture with transactional capability, while the ABAP model is ideal for applications with both analytic and transactional capabilities. Both models improve developer experience and provide an agnostic approach to protect investments.

Feature	SAP Cloud Application Programming Model	ABAP RESTful Application Programming Model
Runtime	Node.js or Java server	SAP NetWeaver ABAP stack
Database Support	SAP HANA, SQLite, PostgreSQL	SAP HANA only
BTP Integration	Mature integration with SAP BTP services	Limited but expanding SAP BTP services integration
I/O Model	Nonblocking I/O model	Blocking I/O model (workaround: CL_ABAP_PARALLEL)
Ideal Use Case	Event-driven architecture with transactional capability	Applications with analytic and transactional capabilities