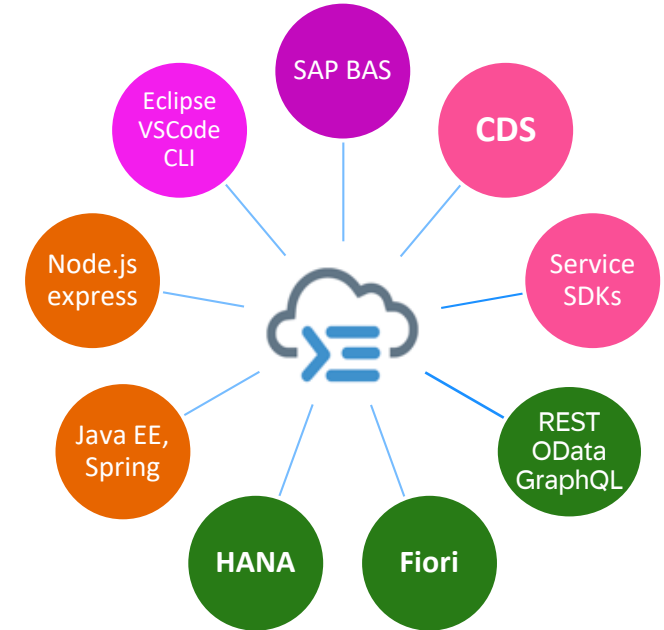**SAP**

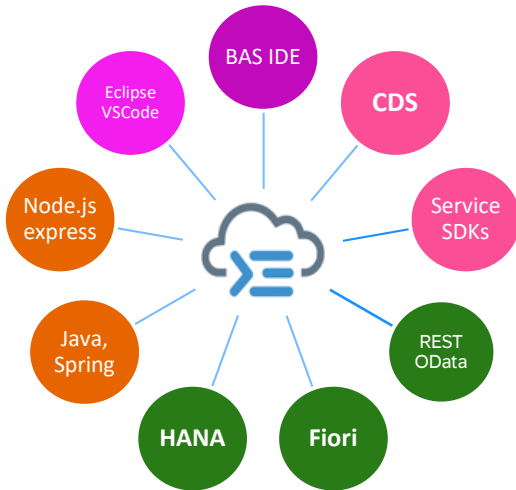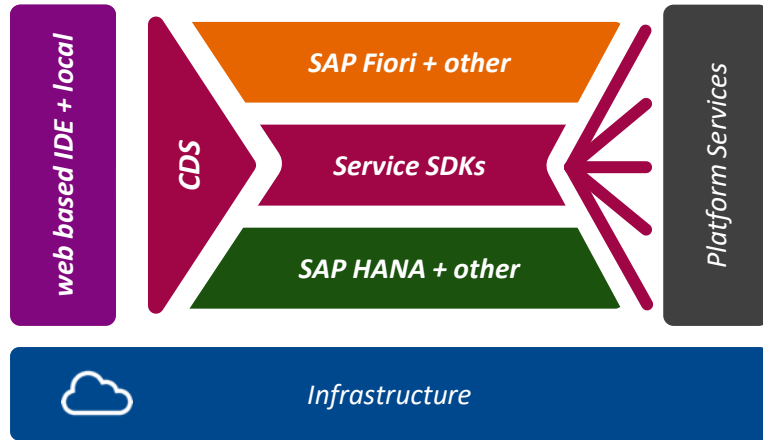# SAP Cloud Application Programming Model Introduction – For CodeJam

Public

Introduction

The SAP Cloud Application Programming Model is an **opinionated**, yet **open** framework of tools, languages and libraries to efficiently build enterprise-grade services and applications.

It guides developers along a 'golden path' of proven **best practices**, while minimizing boilerplate so they can **focus on their domain** problems at hand.

The framework features a mix of broadly adopted **open-source** and **SAP** tools and technologies.

Introduction

We complement *cloud-native technologies* with…

- a **CDS-based Services** framework…
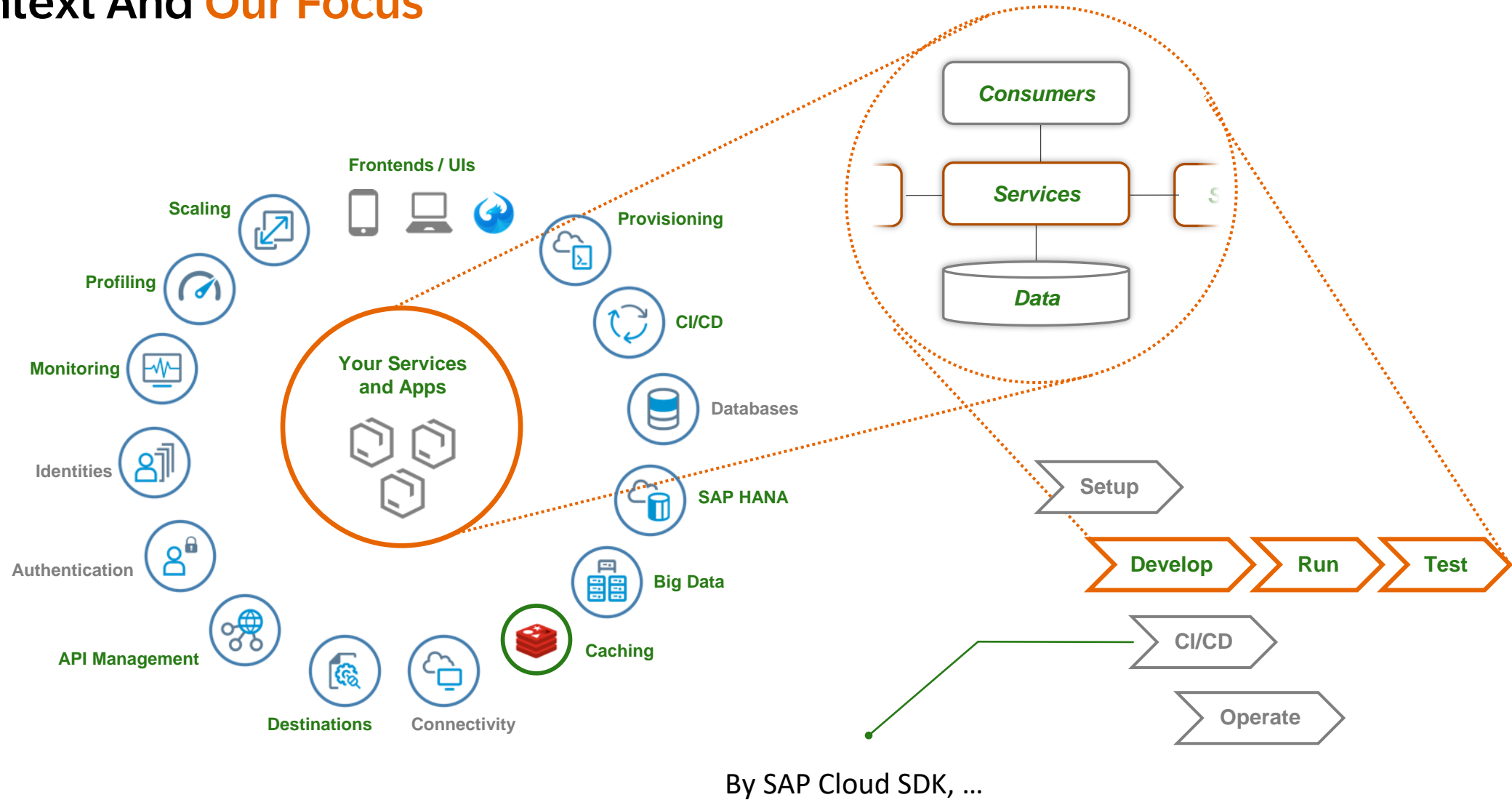- w/ intrinsic **enterprise** & **cloud** qualities

w/ *1st-class support* by and for…

- **SAP Business Application Studio**
- **SAP Fiori**
- **SAP HANA**
- **S/4 Extension** scenarios, …

without lock-in!

→ **Guidance** and **best practices** leveraging proven SAP technologies **+** Open Source software

→ A *consistent* end-to-end **programming model** for *enterprise* services & applications

Context And Our Focus

Frontends / UIs

Scaling

Profiling

Monitoring

Identities

Authentication

API Management

Destinations

Connectivity

Your Services and Apps

Provisioning

CI/CD

Databases

SAP HANA

Big Data

Caching

Consumers

Services

Data

Setup

Develop · Run · Test

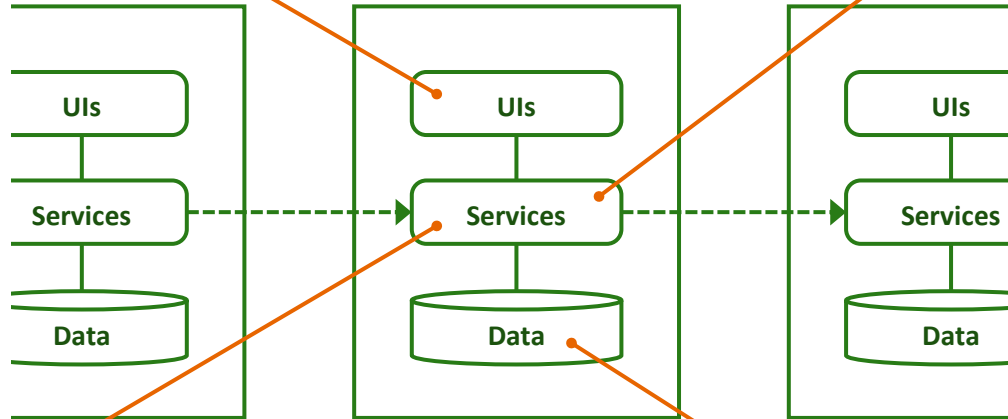CI/CD

Operate

By SAP Cloud SDK, ...

Focus Topics

Serving Fiori UIs

- OData v2+v4
- Draft support
- $batch support

Providing Services

- Service Definitions in CDS
- Service Providers
- Event Handlers → generic + custom
- **Emitting Events**

UIs

UIs

UIs

Services

Services

Services

Data

Data

Data

Consuming Services

- Import Service Definitions from OData, …
- Mashup with local services (using CDS)
- Service Client APIs → typed / cds.ql; pre-built S/4
- Connectivity & Resilience by SDKs
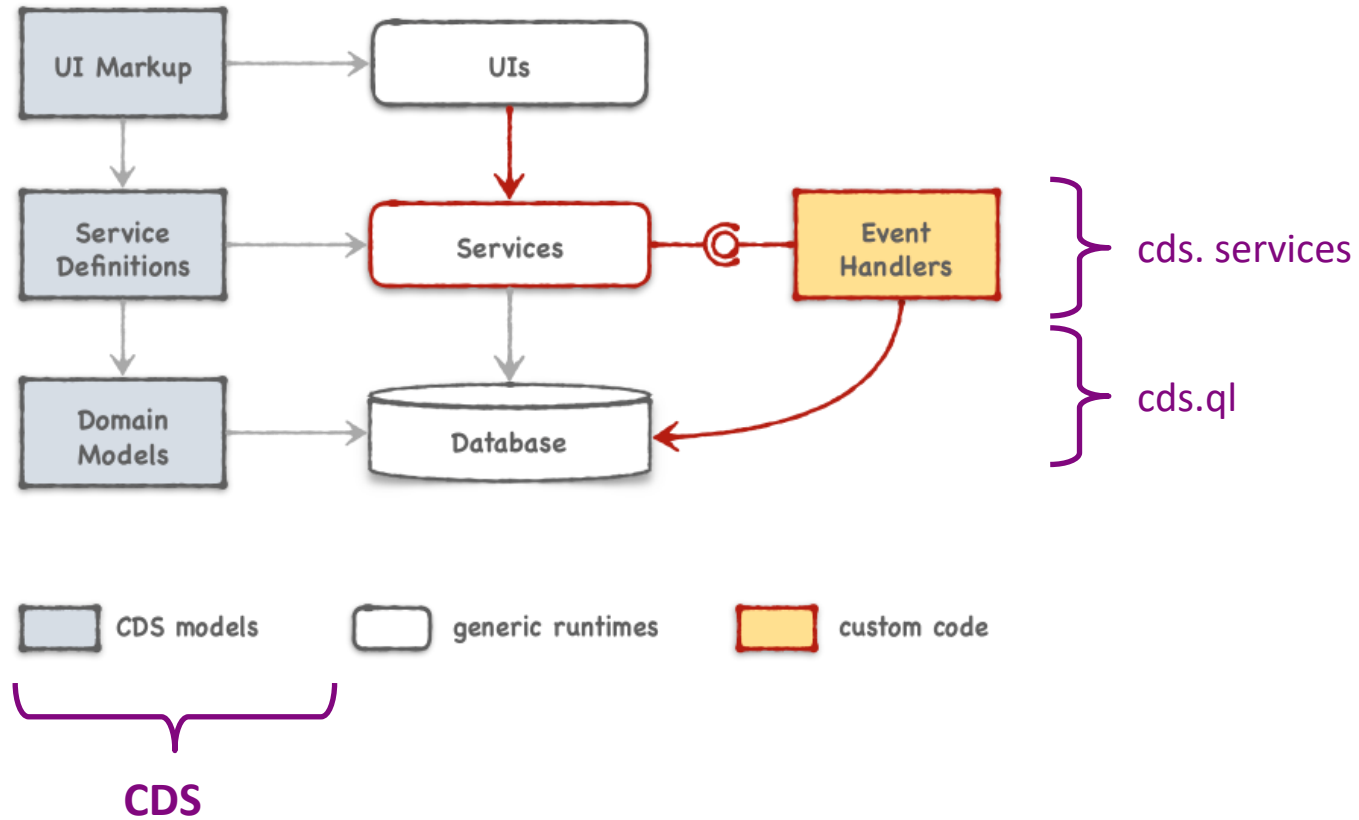- **Subscribing to & Handling Events**

Database Support

- Out-of-the-box: **HANA, sqlite**, PostgreSQL (community)
- Extensible Data Access layers (cds.ql)
- **Schema evolution**

Key Value Propositions

✓ Guidance and Best Practices → opinionated *and* staying open and flexible

✓ Intrinsic Enterprise & Cloud Qualities → taken care of out-of-the-box

✓ Integration and Interoperability → easily reuse models and services

✓ Accelerated Development → grow as you go & time to market

✓ Focus on your Domain → what, not how; minimized boilerplate

✓ Minimized Footprint → enabling serverless deployments

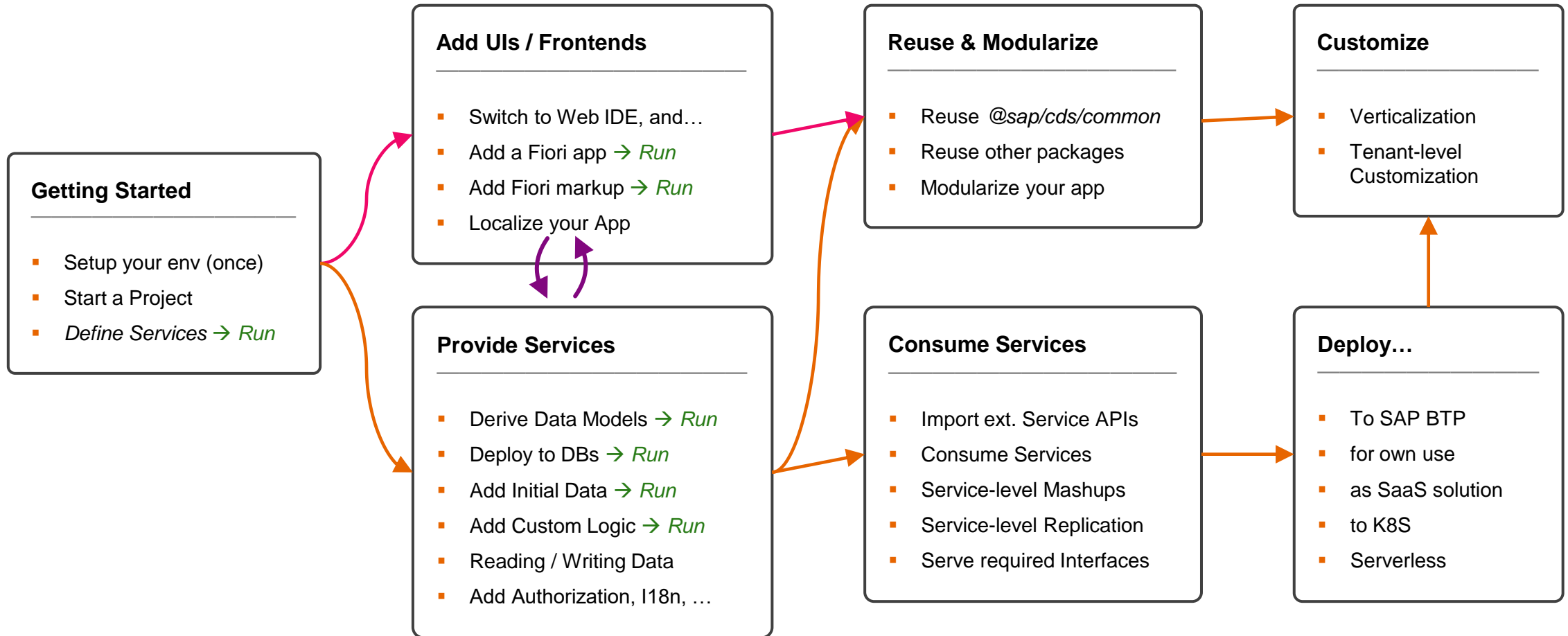✓ Safeguarded Investments → e.g. on journeys to new platform picks

Key Concepts

- **Declarative Approach by CDS**
- Services and Events
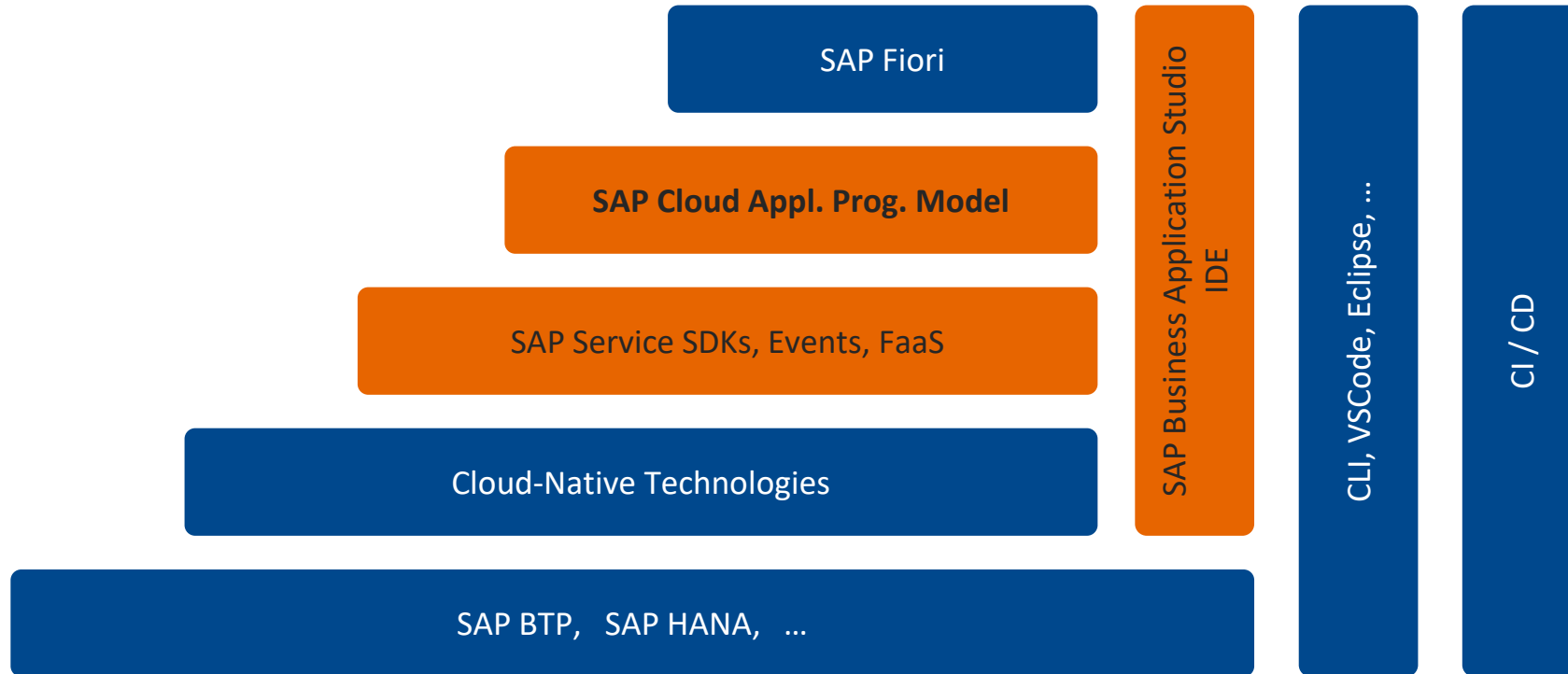- Queries and Views
- Aspects and Mixins

Cookbook — Walking Through Your Tasks…

Getting Started
- Setup your env (once)
- Start a Project
- *Define Services → Run*

Add UIs / Frontends
- Switch to Web IDE, and…
- Add a Fiori app → *Run*
- Add Fiori markup → *Run*
- Localize your App

Provide Services
- Derive Data Models → *Run*
- Deploy to DBs → *Run*
- Add Initial Data → *Run*
- Add Custom Logic → *Run*
- Reading / Writing Data
- Add Authorization, I18n, …

Reuse & Modularize
- Reuse @*sap/cds/common*
- Reuse other packages
- Modularize your app

Consume Services
- Import ext. Service APIs
- Consume Services
- Service-level Mashups
- Service-level Replication
- Serve required Interfaces

Customize
- Verticalization
- Tenant-level Customization

Deploy…
- To SAP BTP
- for own use
- as SaaS solution
- to K8S
- Serverless

Layered Model

Thank you.

Contact information:

Thomas Jung
thomas.jung@sap.com

THE BEST RUN **SAP**