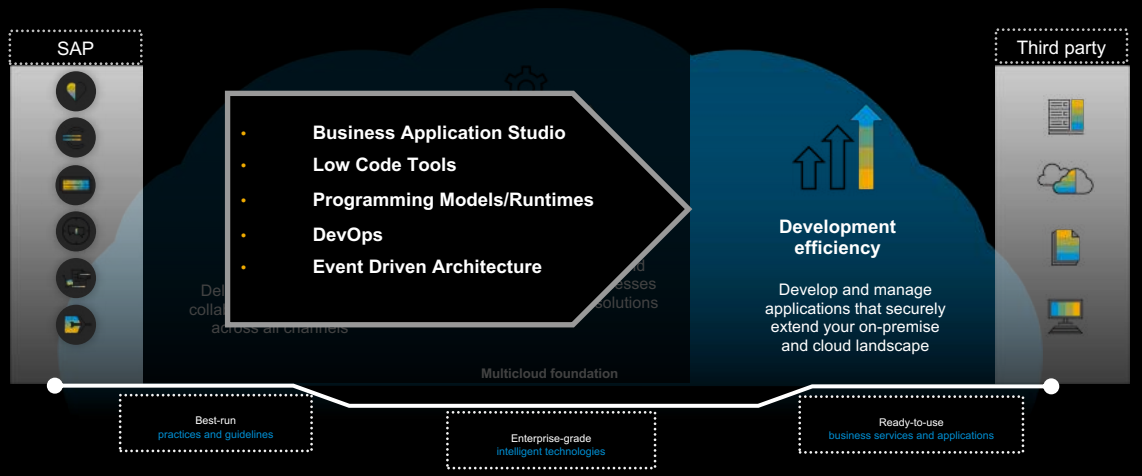
SAP BTP Extension Suite
Cloud Application Programming

SAP BTP Solution Architects
SAP Global Partner Organization

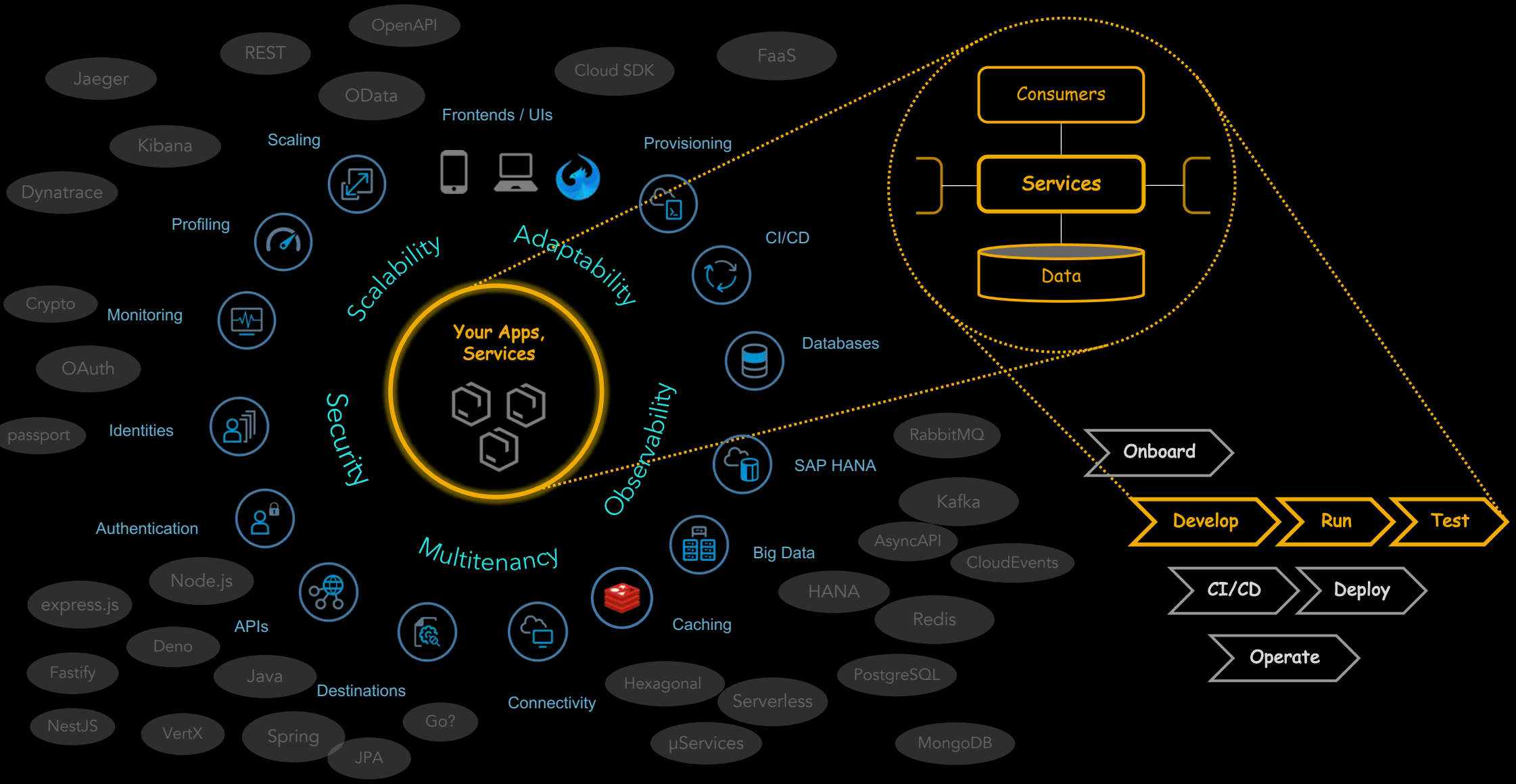
July, 2021

Agenda

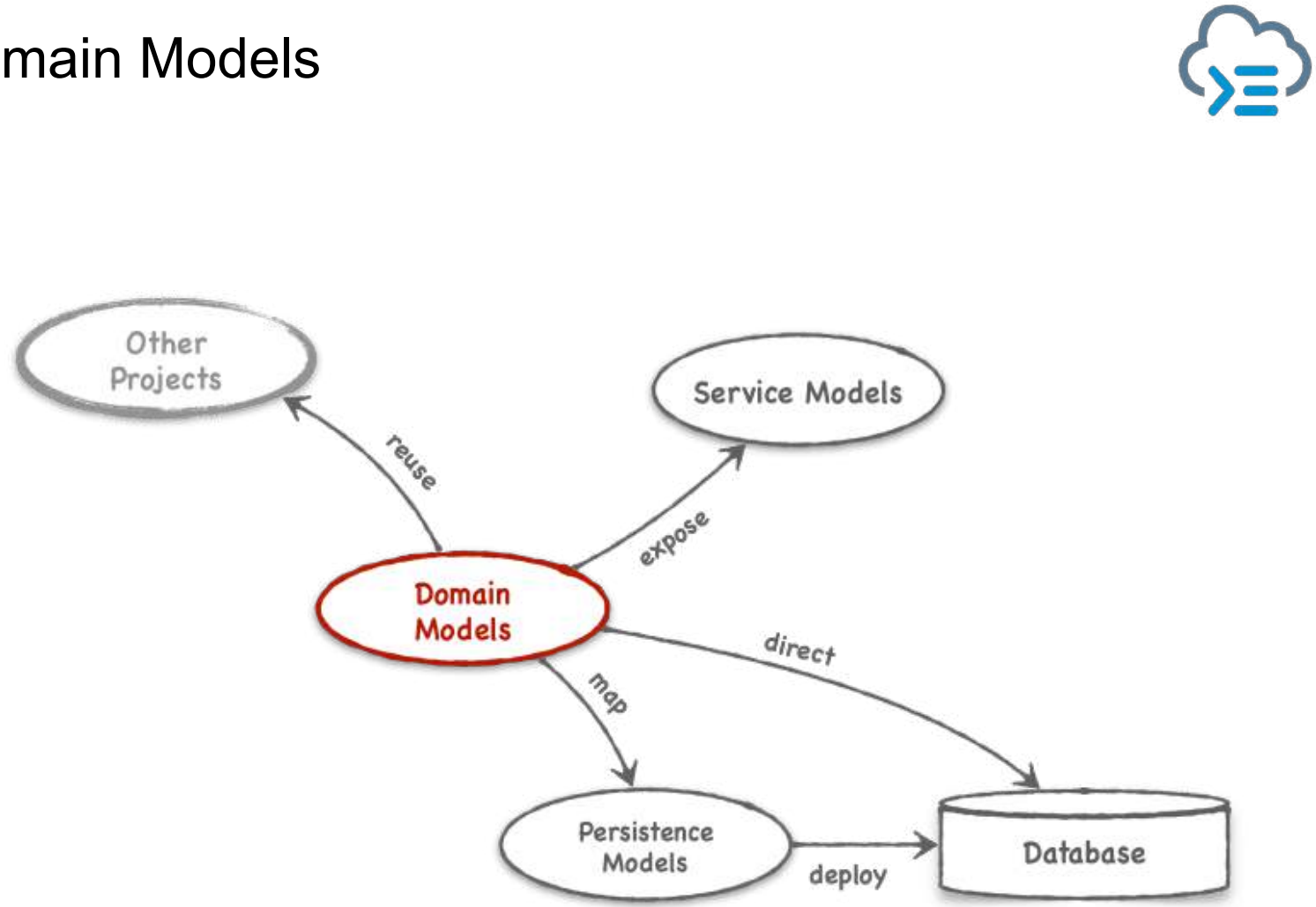
- Motivation (why)
- Core Concepts (what)
- How to Get Started, Demo & Golden Path (how)
- Key Takeaways & Summary



Why CAP? → Guidance & Focus on Domain → *"What, not How"*



Core Concept – Managing Domain Models



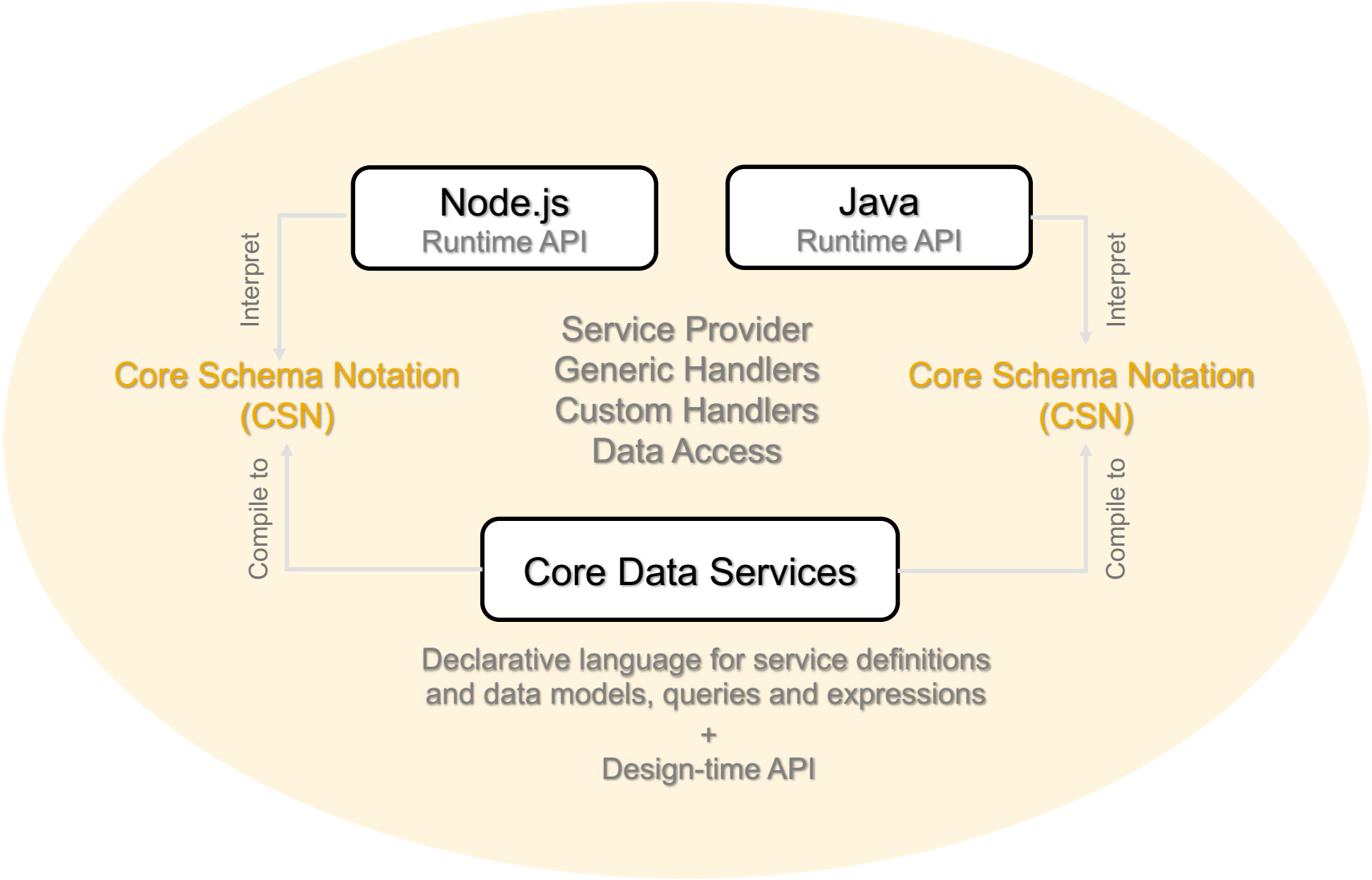
- **Goal:** Foster close collaboration of technical and domain experts (→ Domain-driven Design)
- Therefore: Keep them **clean**, **concise** and **comprehensible** using…

✓ **Aspects** to separate and factor out technical concerns, verticalization and extensibility

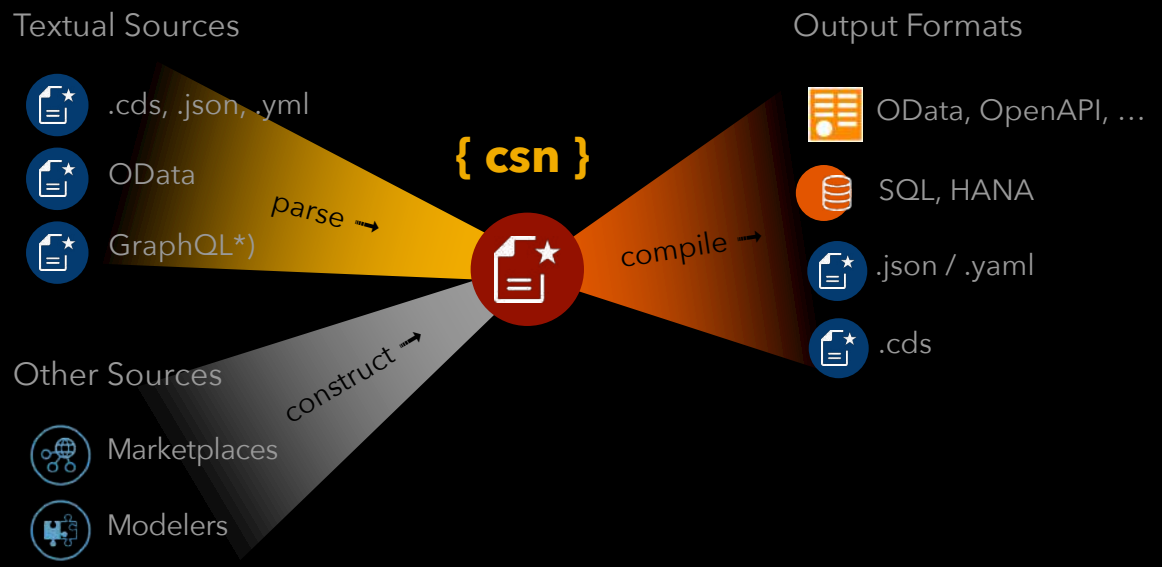
✓ **Common Reuse Types**
e.g. like Code lists, etc.

✓ **Enterprise Features**
e.g. localized data, temporal data, authorization, etc.

Backbone of CAP – Core Data Services (CDS)



Domain Modeling → Focus on Domain, powered by CDS



⇒ **Capture Intent** – that is *What, not How* – in concise and comprehensible models, promoting close collaboration of developers and domain experts to iteratively and gradually refine domain knowledge across layered reuse and extension packages

⇒ **Declarative Approach** fueling **generic runtimes** → out-of-the-box **enterprise services**

CDS Powerful Querying in Comparison

Feature	GraphQL *)	OData	CDS / CQL
Schemas	yes	yes	yes
Mutations / Custom Ops	yes	yes	yes
Queries	yes	yes	yes
- select	yes	yes	yes
- expand	yes	yes	yes
- filtering	**)	yes	yes
- sorting	**)	yes	yes
- pagination	**)	yes	yes
- analytics	**)	yes	yes
References → Associations / Compositions	***)	yes	yes
Select * (→ default in REST)		yes	yes
CRUD defaults		yes	yes
Schema Annotations		yes	yes
Translating b/w QLs → Delegate / Push-down			yes
Views → Generic Providers			yes
Denormalized Projections / Results			yes
Imports → Reuse & Compose			yes
Aspects → Extensibility, Verticalization			yes
Database support ootb, schema evolution			yes
Full SQL query options			yes

*) *with Apollo framework*

**) *requires hard-coded functions*
→ no extensibility

***) *using hard-coded field resolvers*
→ no extensibility
→ bad performance

CDS Frequently Used Commands for the CDS CLI

Description	Command
Jump-start cds-based projects	cds init <project>
Add a feature to an existing project	cds add <feature>
Add models from external sources	cds import <api>
Compile cds models to different outputs	cds compile <models>
Run your services in local server	cds serve <services>
Run and restart on file changes	cds watch
Read-eval-event loop	cds repl
Inspect effective configuration	cds env
Prepare for deployment	cds build
Deploy to databases or cloud	cds deploy
Create an extension project	cds extend <app-url>
Activate an extension project	cds activate
Login to extendable SaaS application	cds login <app-url>
Logout from extendable SaaS application	cds logout

SAP Business Application Studio

Privacy

Dev Spaces

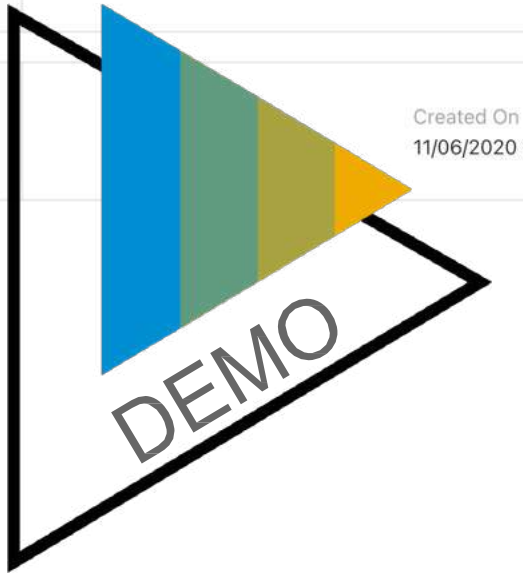
Create and manage your development environment according to the type of applications you want to develop.
You can add extensions and tools to further enhance your development options.

Create Dev Space

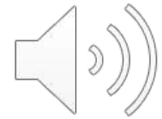
You're using a trial version. Any dev space that hasn't been running for 30 days will be deleted. See the full list of restrictions.

You are allowed a total of 2 dev spaces, with only 1 running at a time.

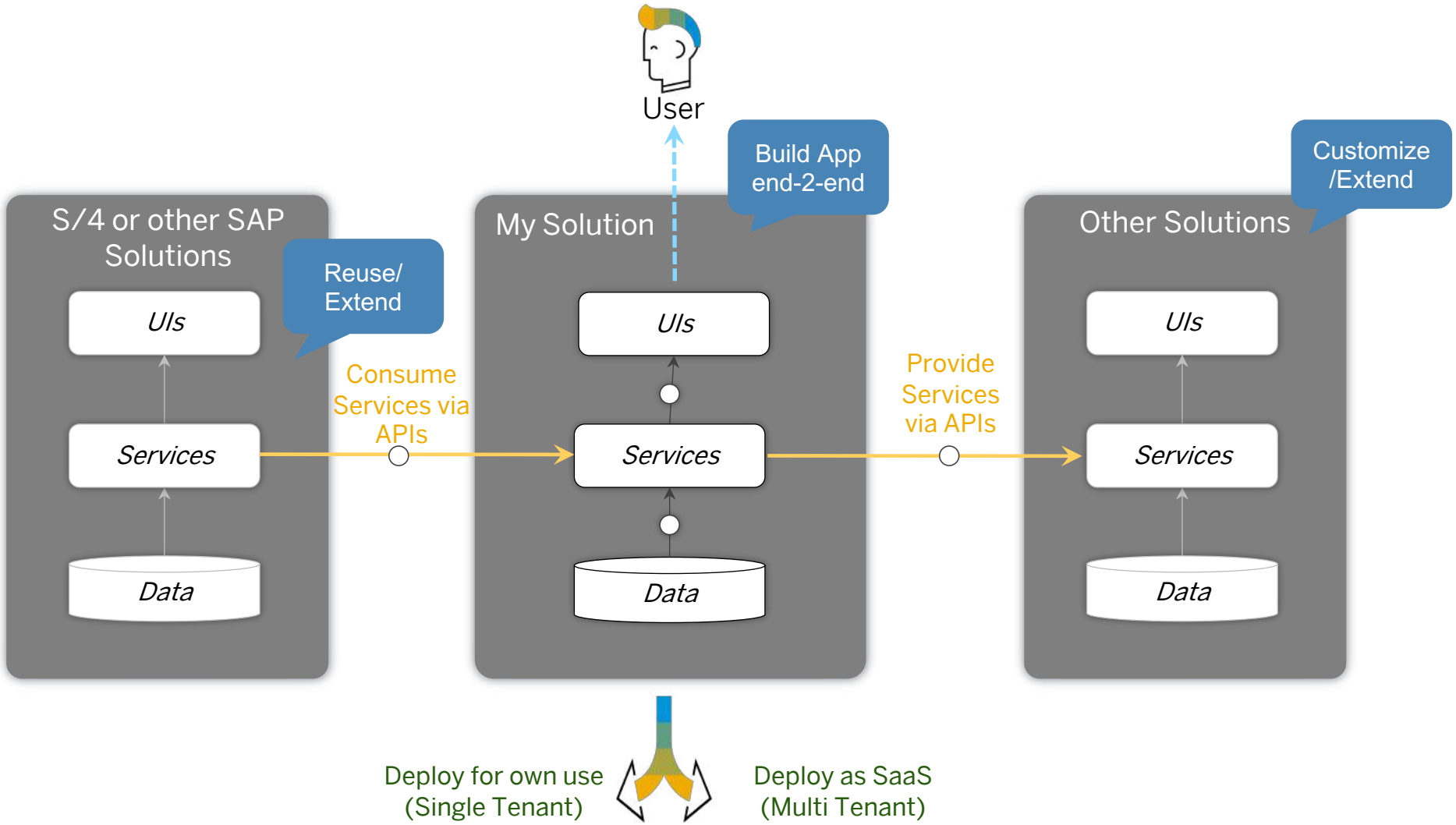
fullstack Full Stack Cloud Application	STARTING	Created On 05/07/2021 4:08 PM	ID ws-6fvx2	
jacob SAP Fiori		Created On 11/06/2020 12:29 PM	ID ws-ps447	



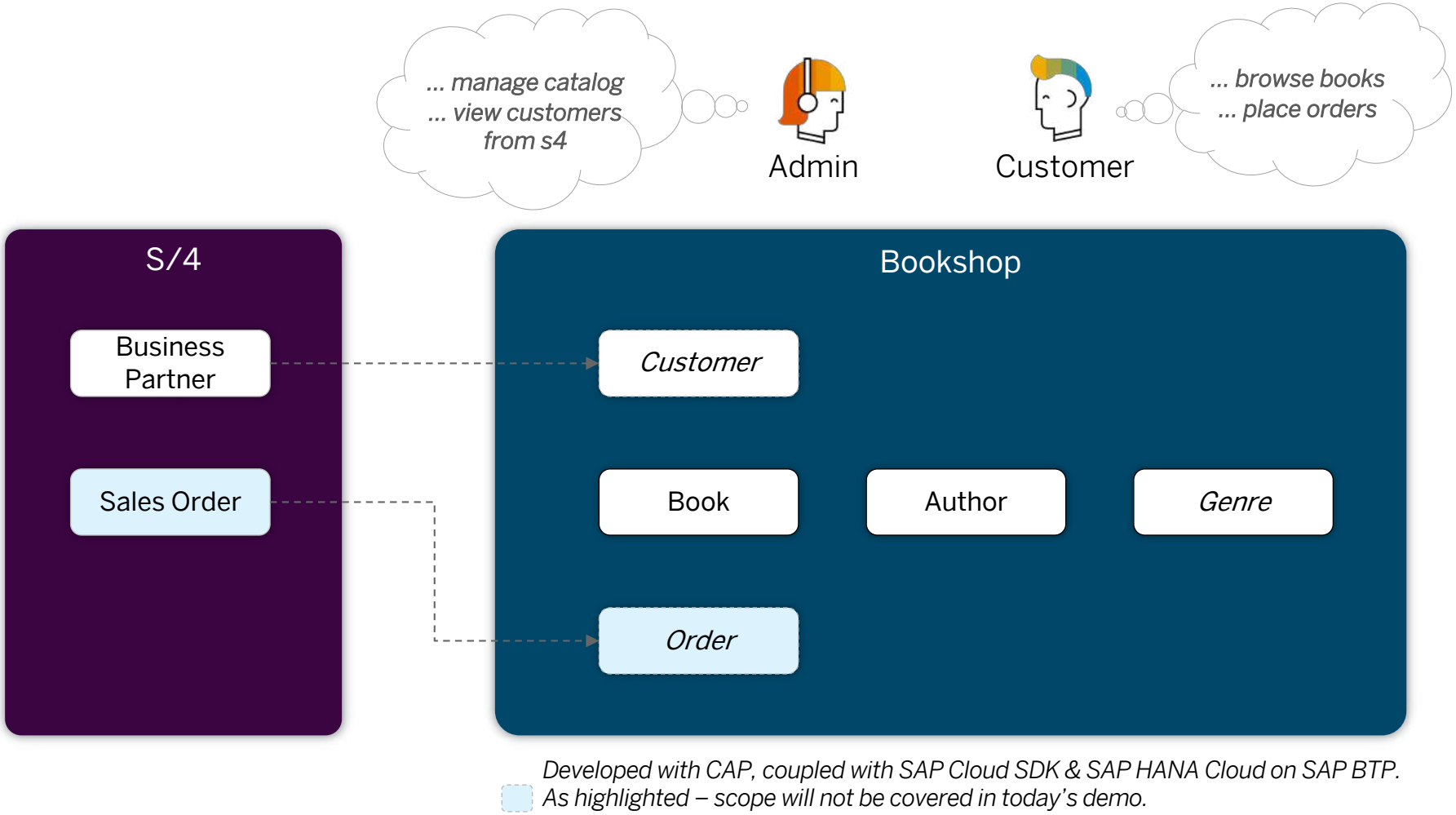
2 minutes



What are some of the typical Use Case Scenarios?



Scenario of a Bookshop Management Solution



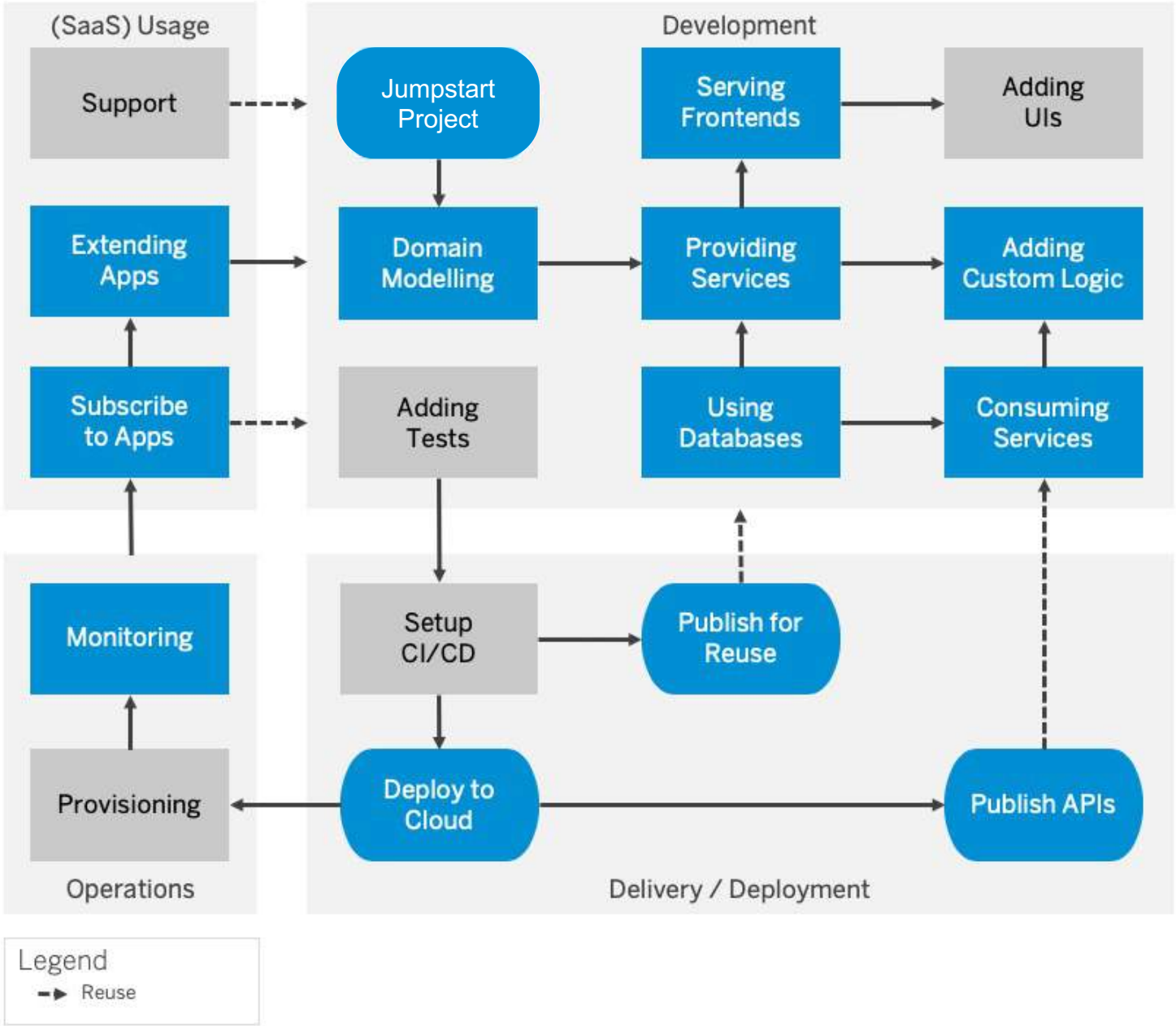
Welcome to CAP

SAP Cloud Application Programming Model

```
cds serve my-services
```

Latest News

- *The **May 2021** release is available → **See What's New***
- *Build a Business Application Using CAP for Node.js → **Try it out***
- *Build a Business Application Using CAP for Java → **Try it out***
- *CAP has its own **Learning Journey** → **Open It***

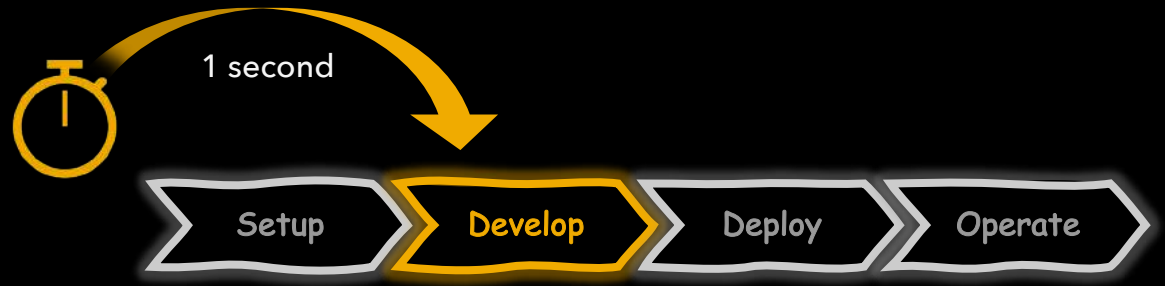


Building Cloud Native App with Enterprise Qualities

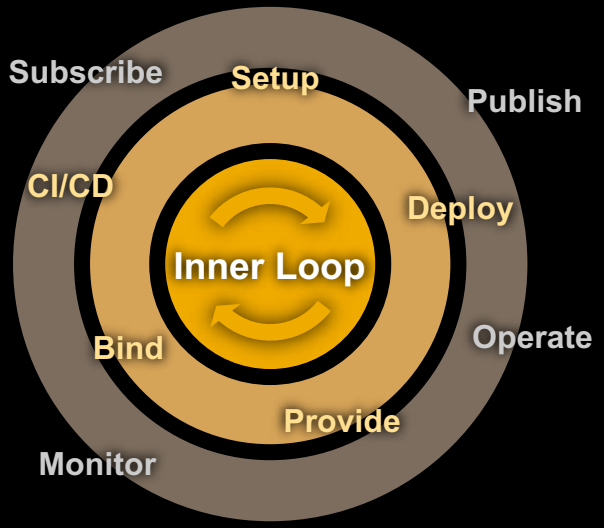
Additional Reading

- Getting Started in a Nutshell
- Project Setup & Layouts
- Jumpstart & Grow-as-you-go
- Features Overview

Jumpstart Project | 2 | 3 | 4 | 5 | 6 | 7 | 8 | …



- *cds init* starts projects w/o upfront setup
- *cds run* + *watch* full-fledged servers from the very beginning



1-digit seconds
turn-around times

Work in Inner Loops as much as possible → min complexity & turn-around times

- *cds add* more things only when required → convention over configuration
- *cds mock* platform services as well as remote business services
- *cds bind* to cloud or backends only when required → min complexity
- *cds deploy* to cloud only when required → saving time & costs

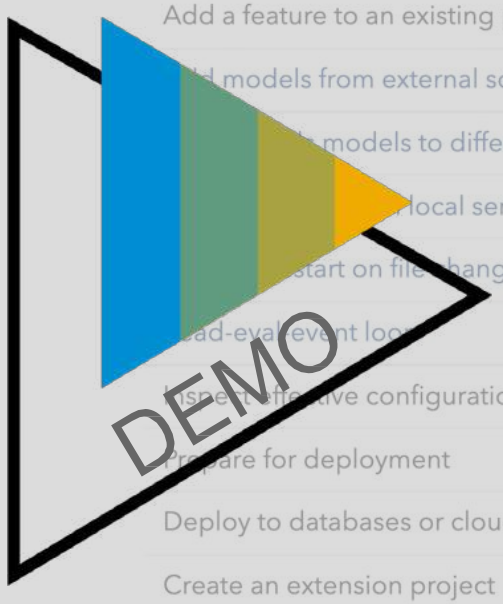
Contracts/APIs First → served out-of-the-box by CAP → parallelize your work

Late-cut μ Services → thru CAP's agnostic services → avoid pre-mature miscuts

Reuse, Compose & Extend → start out from others' work & share with others

Building Cloud Native App with Enterprise Qualities

Jumpstart Project | 2 | 3 | 4 | 5 | 6 | 7 | 8 | …

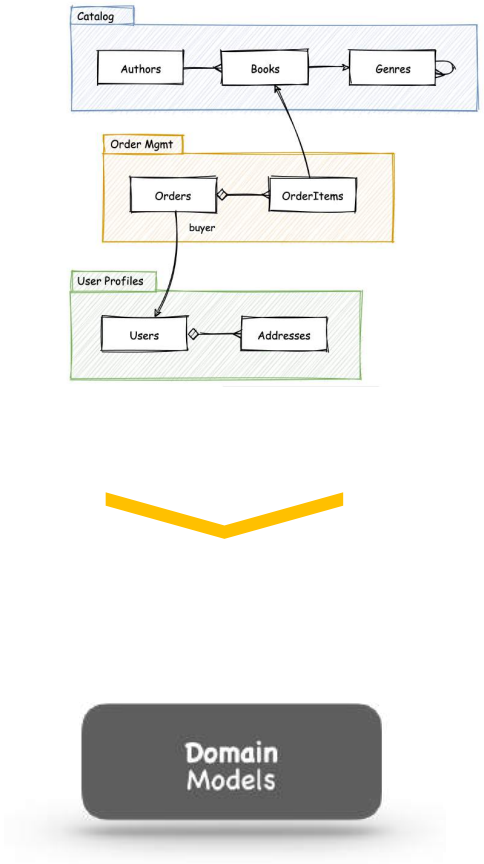


1 minutes

Additional Reading
- Domain modeling with CDS

Building Cloud Native App with Enterprise Qualities

1 | **Domain Modelling** | 3 | 4 | 5 | 6 | 7 | 8 | …




```
namespace capire.bookshop;

entity Books {
  key ID     : Integer;
  title      : localized String;
  descr      : localized String;
  author     : Association to Authors;
  genre      : Association to Genres;
  price      : Decimal;
}
entity Authors {
  key ID     : Integer;
  name       : String;
  books      : Association to many Books;
}
entity Genres : cuid, CodeList {
  children   : Composition of many Genres;
}
entity Orders : cuid {
  Items      : Composition of many {
    book     : Association to Books;
    amount   : Integer;
  }
  status     : enum { delivered; … }
  tags       : array of String;
  buyer      : User;
  virtual total : Decimal;
}
```

2 Capture a **Domain Model**, using CDS entities, Associations, and other higher-level constructs…

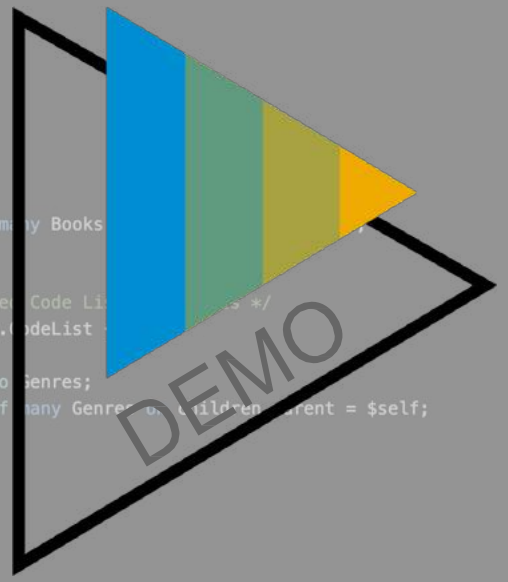
⇨ Conceptual Modelling by CDS: **concise** and **comprehensible** models, allowing to stay close to your conceptual thinking

⇨ Promoting Domain-driven Design: close collaboration of **developers** and **domain experts** to iteratively refine domain knowledge

⇨ Document-oriented Modelling: fueling advanced generic providers, and mapping to relational as well as NoSQL databases

Building Cloud Native App with Enterprise Qualities

1 | Domain Modelling | 3 | 4 | 5 | 6 | 7 | 8 | …

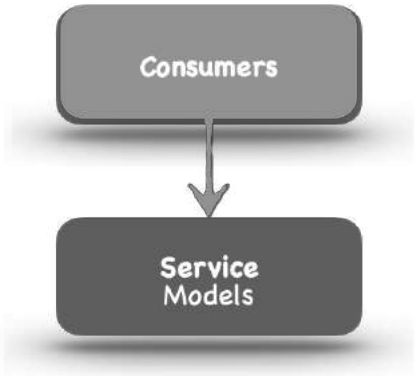


3 minutes

Additional Reading
- Service Modeling with CDS

Building Cloud Native App with Enterprise Qualities

1 | 2 | **Providing Services** | 4 | 5 | 6 | 7 | 8 | …




```
entity Books {
  key ID     : Integer;
  title      : localized String;
  descr      : localized String;
  author     : Association to Authors;
  genre      : Association to Genres;
  price      : Decimal;
}
entity Authors {
  key ID     : Integer;
  name       : String;
  books      : Association to many Books;
}
entity Genres : cuid, CodeList {
  children   : Composition of many Genres;
}
```

```
using capire.bookshop as my from './schema';

// for browsing by end users
service CatalogService {
  entity Books as projection on my.Books {
    *, author.name as author
  } excluding {
    createdBy, modifiedBy
  }
  action submitOrder ( book, amount );
  event news : array of { book, info };
}

// for maintenance by administrators
service AdminService {
  entity Books as projection on my.Books;
  entity Authors as projection on my.Authors;
  entity Genres as projection on my.Genres;
}
```

3 Add **Service Models** as APIs and **facades** to consumers, capturing **behavioral aspects** of your domain

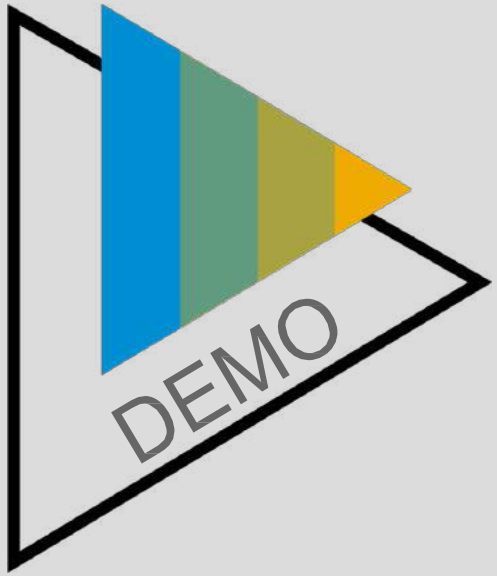
⇨ Service-centric Paradigm:
every active thing is a (**stateless**) Service, processing (passive) data

⇨ Powerful View Building:
to expose **denormalized** views on underlying domain models

⇨ Single-purposed 'Nano' Services:
rule of thumb: 1 service per use case to modularize and optimize

Building Cloud Native App with Enterprise Qualities

1 | 2 | Providing Services | 4 | 5 | 6 | 7 | 8 | …

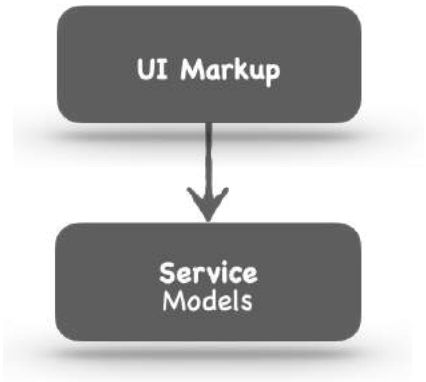


5 minutes

Additional Reading
- Adding SAP Fiori Annotations

Building Cloud Native App with Enterprise Qualities

1 2 3 **Serving Frontends** 5 6 7 8 …




```
// Fiori Markup
annotate Books with @(
  UI.SelectionFields: [
    ID, title, price
  ],
  UI.LineItem: [
    { Value: title, Label: 'Book Title' },
    { Value: author, … },
    { Value: price, … },
  ]
);
```

4 Using **Annotations** for semantic enrichment for various contexts and consumers, e.g…

⇨ UI Markup
e.g. Fiori UIs served completely out of CDS Models

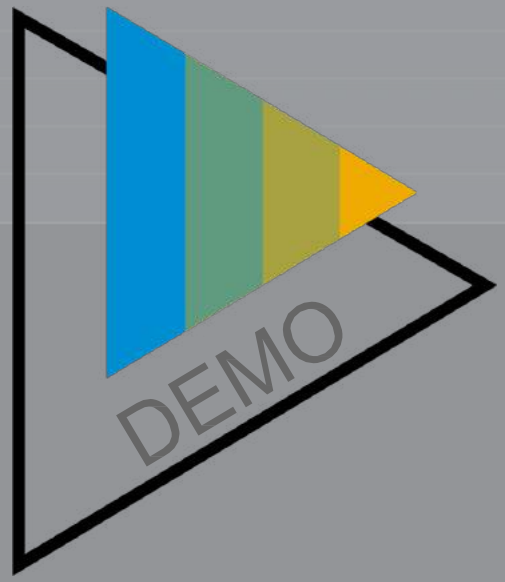
⇨ Analytics Markup
for Embedded Analytics as well as SAP Analytics Cloud

⇨ Custom Vocabularies
everybody can introduce and add new annotation vocabularies

Combined with Aspects:
in separate models, possibly in separate projects

Building Cloud Native App with Enterprise Qualities

1 2 3 **Serving Frontends** 5 6 7 8 …

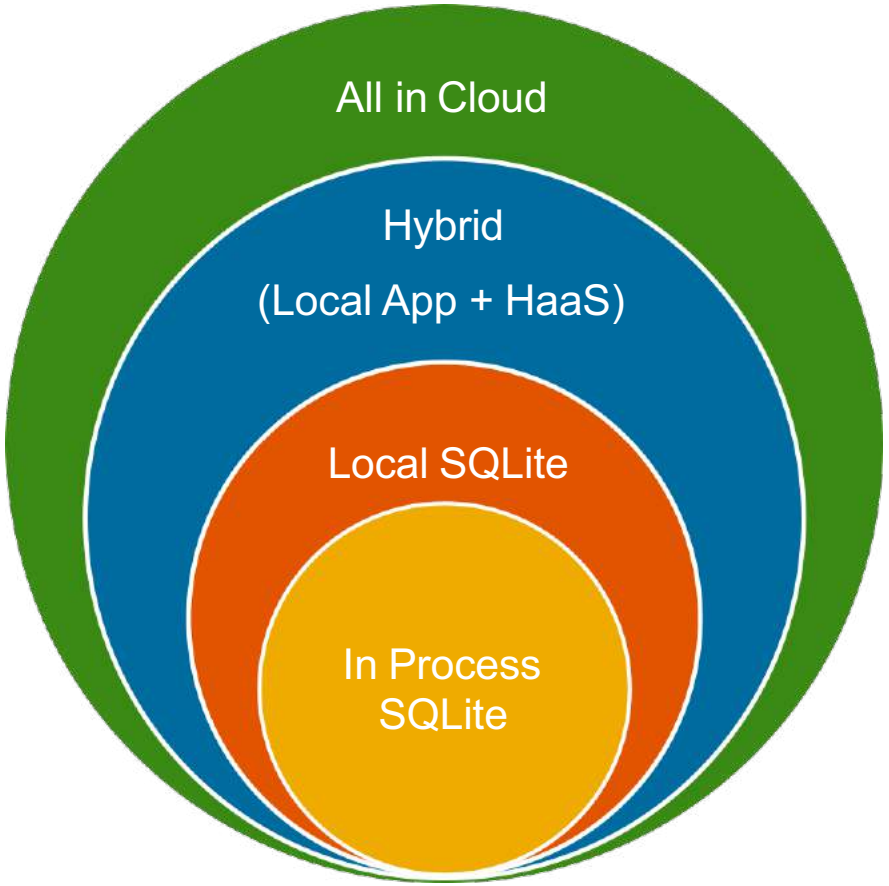


6 minutes

Additional Reading
- Using Databases

Building Cloud Native App with Enterprise Qualities

1 2 3 4 **Using Databases** 6 7 8 …



Deploying for Production [1]
Deploy MTA to SAP Cloud Platform
SAP HANA

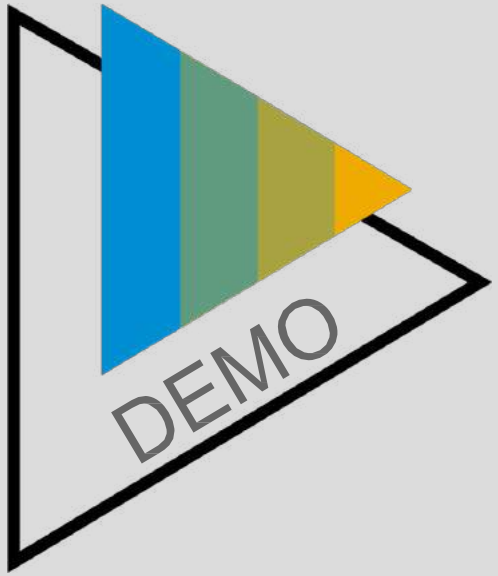
cds deploy --to hana [2]
Deploy data to SAP HANA
App stays local
SAP HANA

cds deploy --to sqlite
Deploy to local SQLite database
Local SQLite

cds run [app] --in-memory
Run app with in-memory SQLite
In-Memory SQLite

Building Cloud Native App with Enterprise Qualities

1 2 3 4 **Using Databases** 6 7 8 …

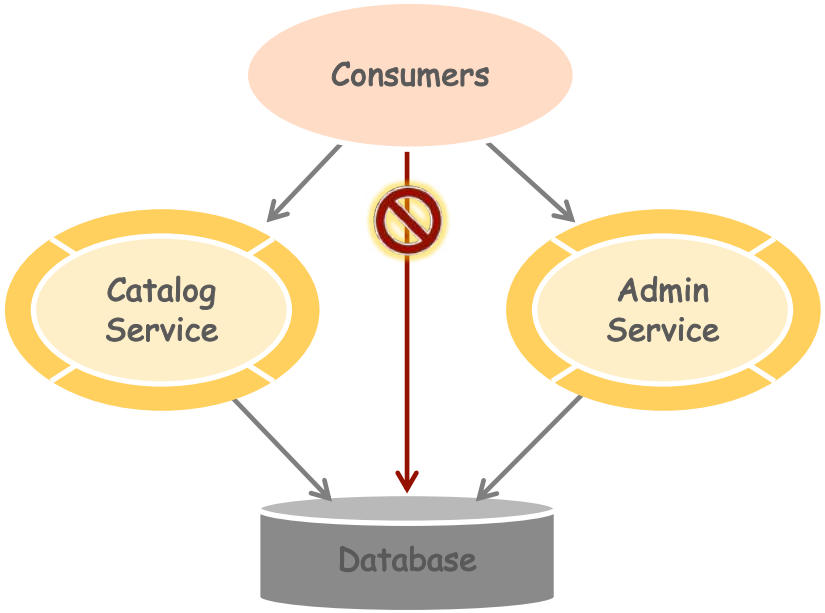


8 minutes

Additional Reading
- Consuming Services

Building Cloud Native App with Enterprise Qualities

1 2 3 4 5 **Consuming Services & Custom Logic** 7 8 …



⇒ Exposing restricted APIs with **denormalized projections** on underlying entities

⇒ All services support **common APIs** to **consume** and **implement** services

Every active thing in CAP is a **Service**…

- always **stateless**, acting on **passive data** → scalability
- reflecting individual **use cases** → 'nano services'
- acting as **facades** for consumers to access data
- including **technical**/framework services in CAP
- exchanging messages via **uniform** APIs

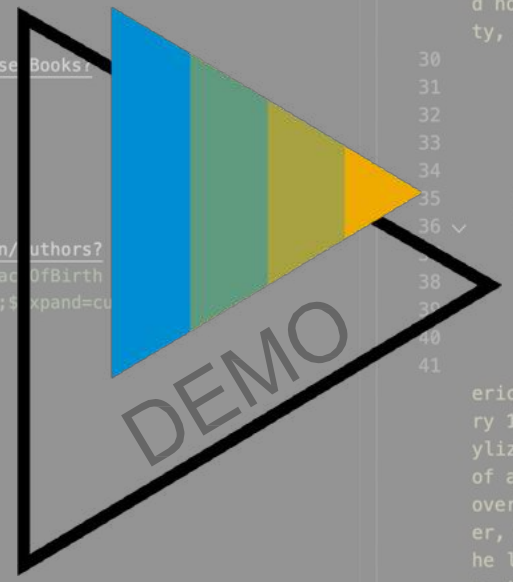
```
// some consumer code…
const cats = cds.connect.to ('CatalogService')
let books = await cats.read ('Books')
for (let book of books)
   if ('→ like it? ') cats .addToBasket (book)

// implementing provided services…
CatService .on ('READ', 'Books', (req)=>{ … })

// intercepting consumed services…
db .on ('READ', '*', (req)=>{ … })
```

Building Cloud Native App with Enterprise Qualities

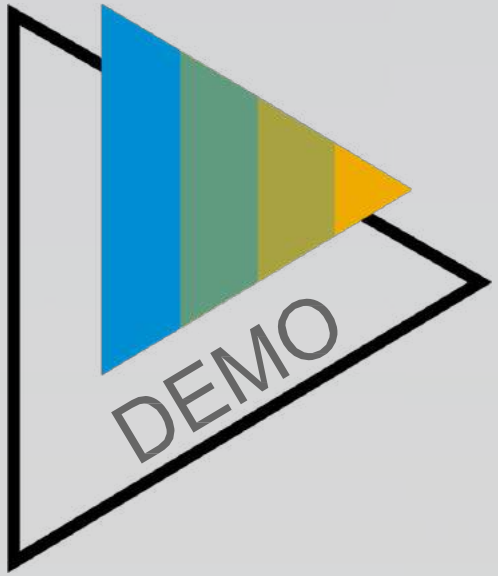
1 2 3 4 5 **Consuming Services & Custom Logic** 7 8 …



4 minutes

Building Cloud Native App with Enterprise Qualities

1 2 3 4 5 6 **Deploy to Cloud** 8 …



10 minutes

https://bookcatalog-bootcamp.cfapps.eu10.hana.ondemand.com/api-docs/admin/

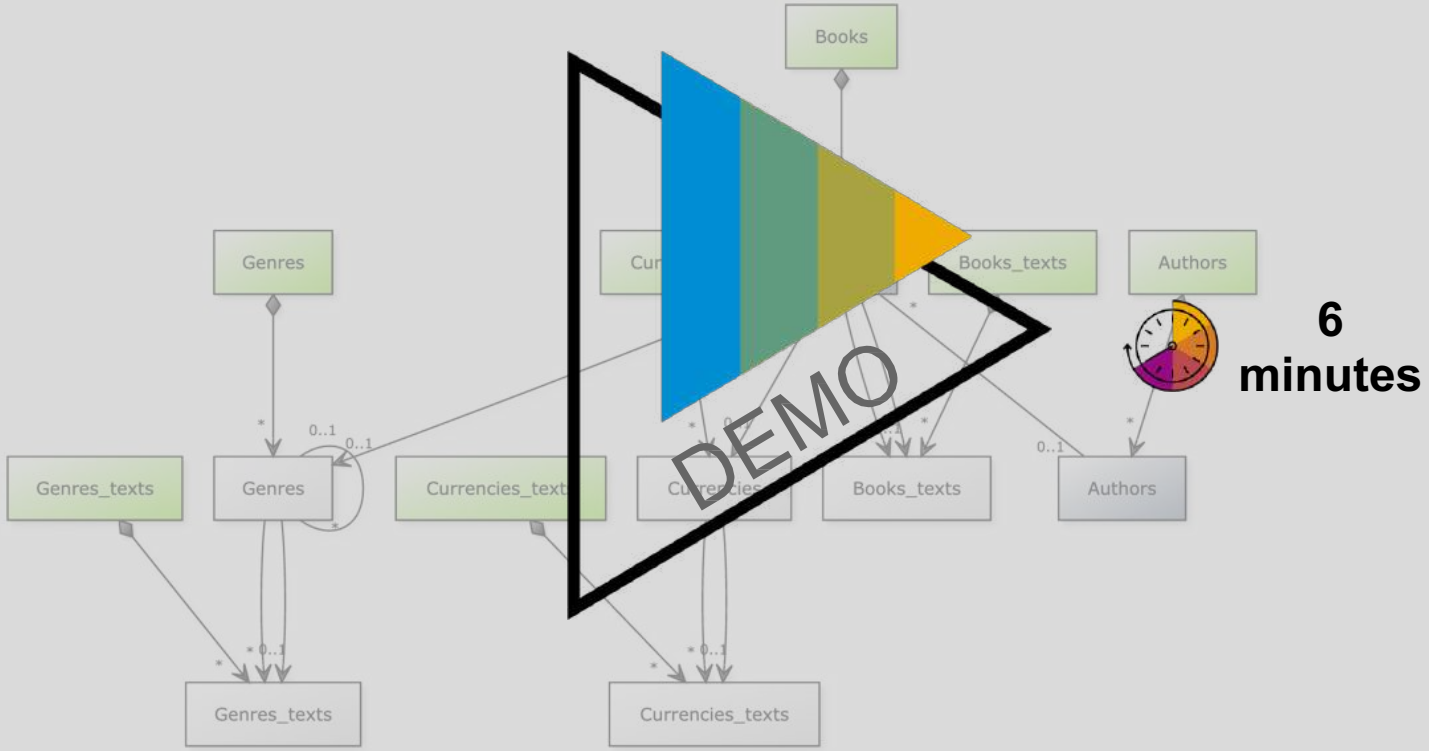
Swagger

Building Cloud Native App with Enterprise Qualities

1 2 3 4 5 6 7 **Advanced Features** …

Service for namespace AdminService

Entity Data Model



6 minutes

Legend

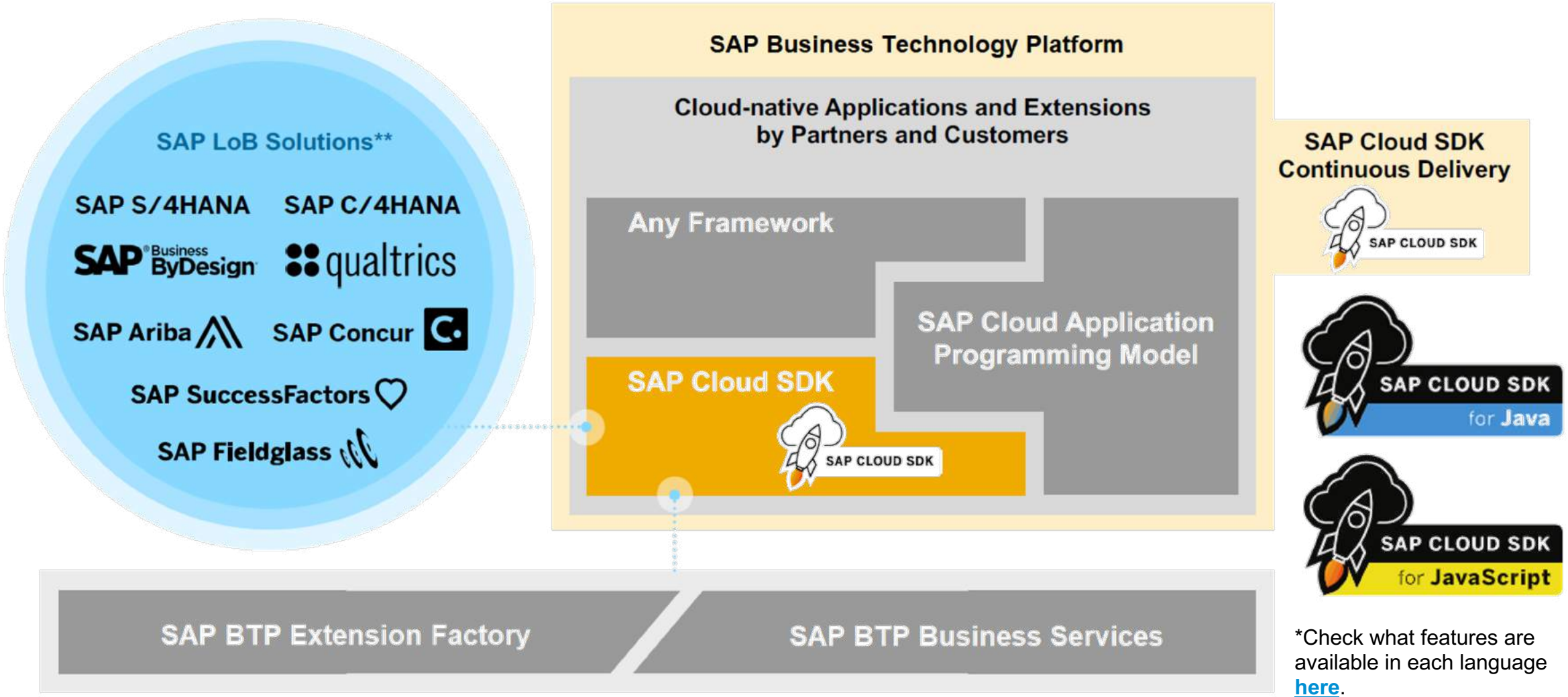
EntitySet/Singleton/Operation | EntityType | ComplexType | External.Type

Building Cloud Native App with Enterprise Qualities

Additional Reading

- SAP Cloud SDK official documentation
- Blog on a working example
- Getting started with Cloud SDK NodeJS
- S4 Virtual Data Model

1 2 3 4 5 6 7 8 **SAP Cloud SDK** …



*Check what features are available in each language **here**.

Benefits of SAP Cloud SDK

Example: Connecting to SAP S/4HANA

Without SDK
(already simplified)

Instead of menial, error-prone, low-level code…

```
// Implement tenant-aware logic to retrieve S4HC destination from SAP Cloud Platform
String destinationUrl = retrieveDestinationFromCloudPlatform();
```
Retrieve config

```
// Implement potentially complex authentication flow (OAuth 2) depending on customer configuration
String authHeader = createAuthorizationHeader();
```
Authenticate

```
// Manually build up request URL (and, possibly request body)
StringBuilder url = new StringBuilder(destinationUrl);
url.append("/sap/opu/odata/sap/API_BUSINESS_PARTNER/A_BusinessPartner");
url.append("&$select=BusinessPartner,LastName");
url.append("&$filter=BusinessPartnerCategory eq '1'");

URL urlObj = new URL(url.toString());
```
Build request

```
HttpURLConnection connection = (HttpURLConnection)
urlObj.openConnection();
connection.setRequestMethod("GET");
connection.setRequestProperty("Content-Type", "application/json");
connection.setRequestProperty("Accept", "application/json");
connection.setRequestProperty("Authorization",authHeader);
if(onPremise) {
    // Determine and add connectivity header required by SAP Cloud Connector
    // ...
}
connection.setDoInput(true);
```
Connect

```
try {
    int responseCode = connection.getResponseCode();
} catch (IOException e) {
    // Exception handling (non-resilient)
}
final InputStreamReader in = new InputStreamReader(connection.getInputStream());
String response = CharStreams.toString(in);

// Implement own Java class for result set with 100+ properties and parse response
List<MyBusinessPartner> result = Arrays.asList(
    new Gson().fromJson(response, MyBusinessPartner[].class));
```
Handle response

Convert result

With SDK

… let developers focus on solving business problems.

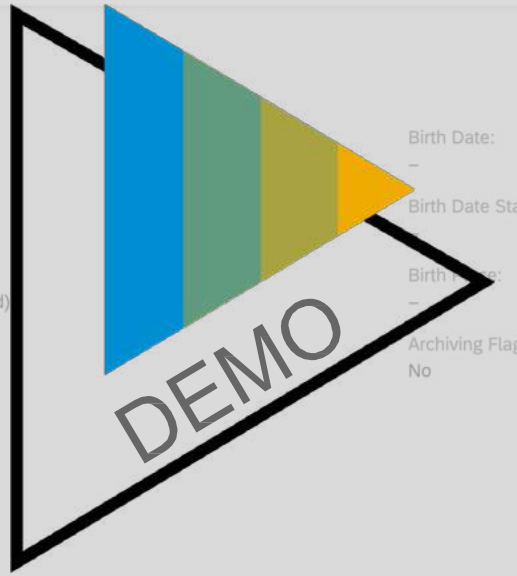
```
final List<BusinessPartner> businessPartners = service.getAllBusinessPartner()
        .select(BusinessPartner.BUSINESS_PARTNER, BusinessPartner.LAST_NAME)
        .filter(BusinessPartner.BUSINESS_PARTNER_CATEGORY.eq(CATEGORY_PERSON))
        .execute();
```

Easy, **type-safe**, and fluent access to **CRUD APIs** + additional **advanced features**:

- Dependency injection & mocking
- Optimistic concurrency control
- Multi-tenancy
- Resilience*, Cache* and Security
- Extensibility
- And more…

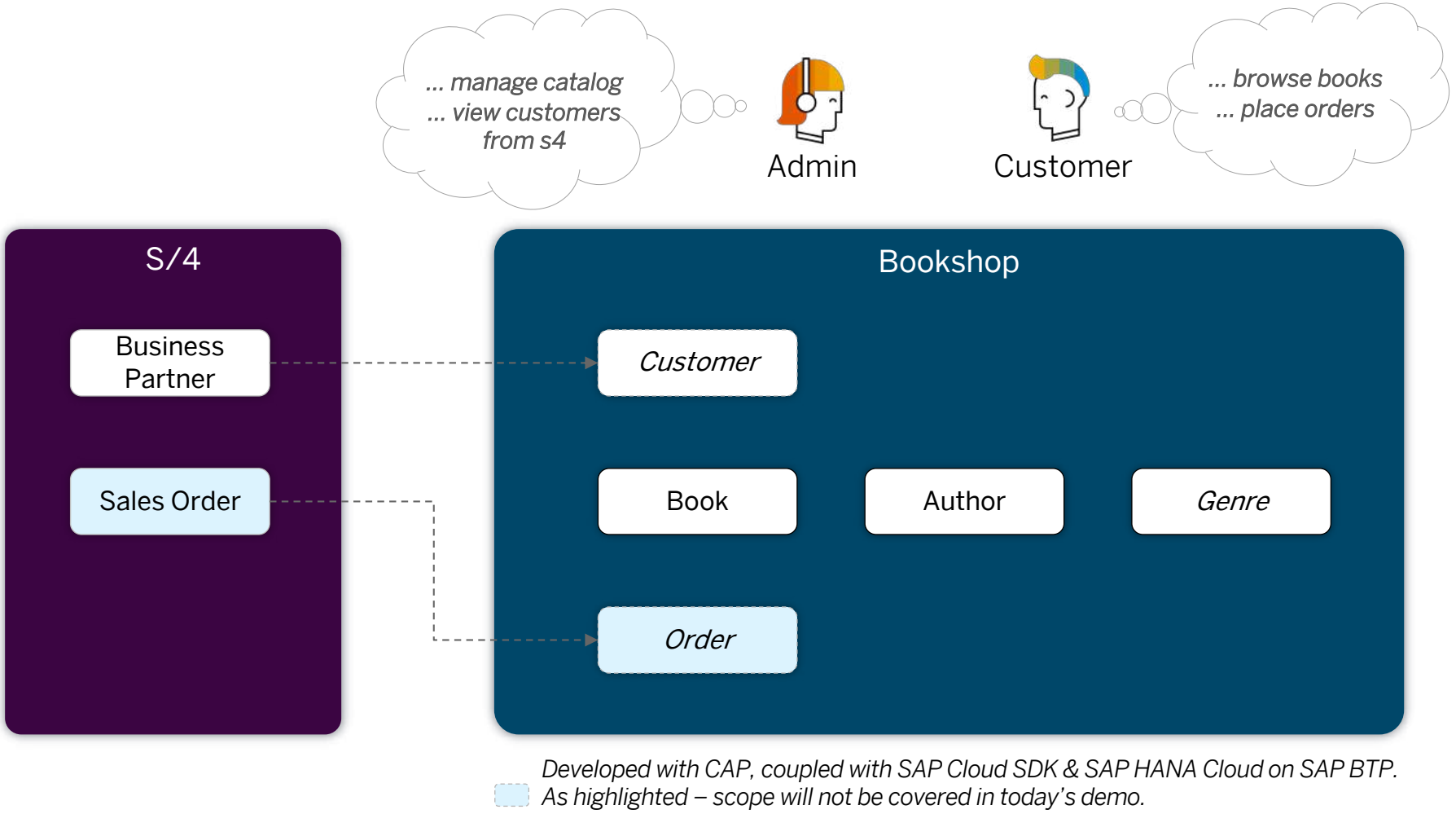
Building Cloud Native App with Enterprise Qualities

1 2 3 4 5 6 7 8 **SAP Cloud SDK** …

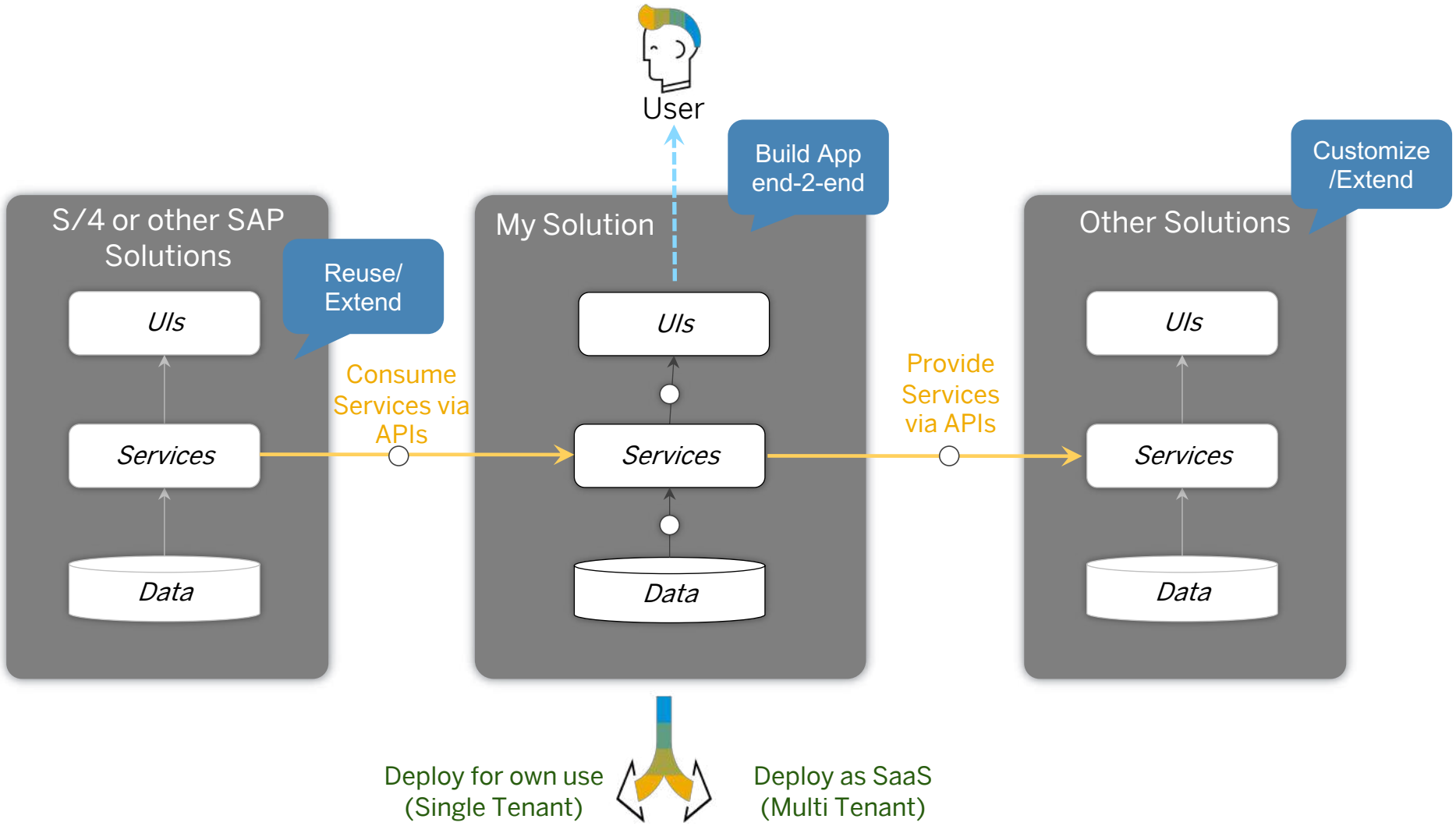


10 minutes

Recap: Scenario of a Bookshop Management Solution



Recap: Use Case Scenarios



Recap & Summary

What a Journey! Highlighting CAP Features & Qualities

Serving Frontends

- *Angular, React, Vue.js, … via ajax*
- *Fiori via Annotations + Draft*
- *Analytical Queries*

Streamlined & Accelerated Development

- *Jumpstart & Grow-as-you-go…*
- *Scalability, Security, Multitenancy*

Platform Integration out-of-the-box

- *Simplified, Platform-Agnostic APIs*
- *Safeguarding Investments*

Providing Services

- *Generic Providers → Low Code*
- *Synchronous & Asynchronous*
- *Via REST, OData, …*

Core Concepts

- *Domain Modeling → focus on domain; powd. by CDS*
- *Mixed-in Aspects → sep. of concerns – verticalization*
- *Powerful Querying → de-normalized views – mashups*
- *Pervasive Services → service-centric paradigm*
- *Ubiquitous Events → intrinsic messaging*

Consuming Services

- *Via Uniform, Protocol-Agnostic APIs*
- *Synchronous & Asynchronous*
- *CDS-based Mashups*

Enterprise Features out-of-the-box

- *Best Practices served out-of-the-box*
- *for data-centric business apps*
- *Minimizing tedious recurring tasks*

Cloud-Native Qualities by Design

- *Adaptability, Extensibility, Observability*
- *Scalability, Security, Multitenancy*

Database Support

- *CDS → DB Schema; Schema Evolution*
- *HANA, H2, SQLite, … → SQL + NoSQL*

Key Takeaways

The **Application Programming Model** is SAP's programming model for building enterprise business applications by guiding developers with SAP best practices for full stack development.

Seamlessly integrates Open Source Software and SAP technologies into one consistent e2e programming model and development experience.

Light-weight & modular implementation of CDS allowing rapid development of persistency, business logic and UI, tailored for SAP Business Technology Platform.

Guides developers with by providing a set of enterprise-grade tools, languages and libraries, allowing them to focus only on their domain logic.

Call to Action

Now, it's your turn to take action to put your learnings into prototypes.

Complete 3 Learnings Resources

1. Build an Application End-to-End Using CAP, Node.js and VS Code (SAP Dev Tutorial) ~240 minutes
2. In-depth Tutorials of CAP (YouTube Playlist) ~150 minutes
3. Any one of the learning resources shared later based on your specific exposure.

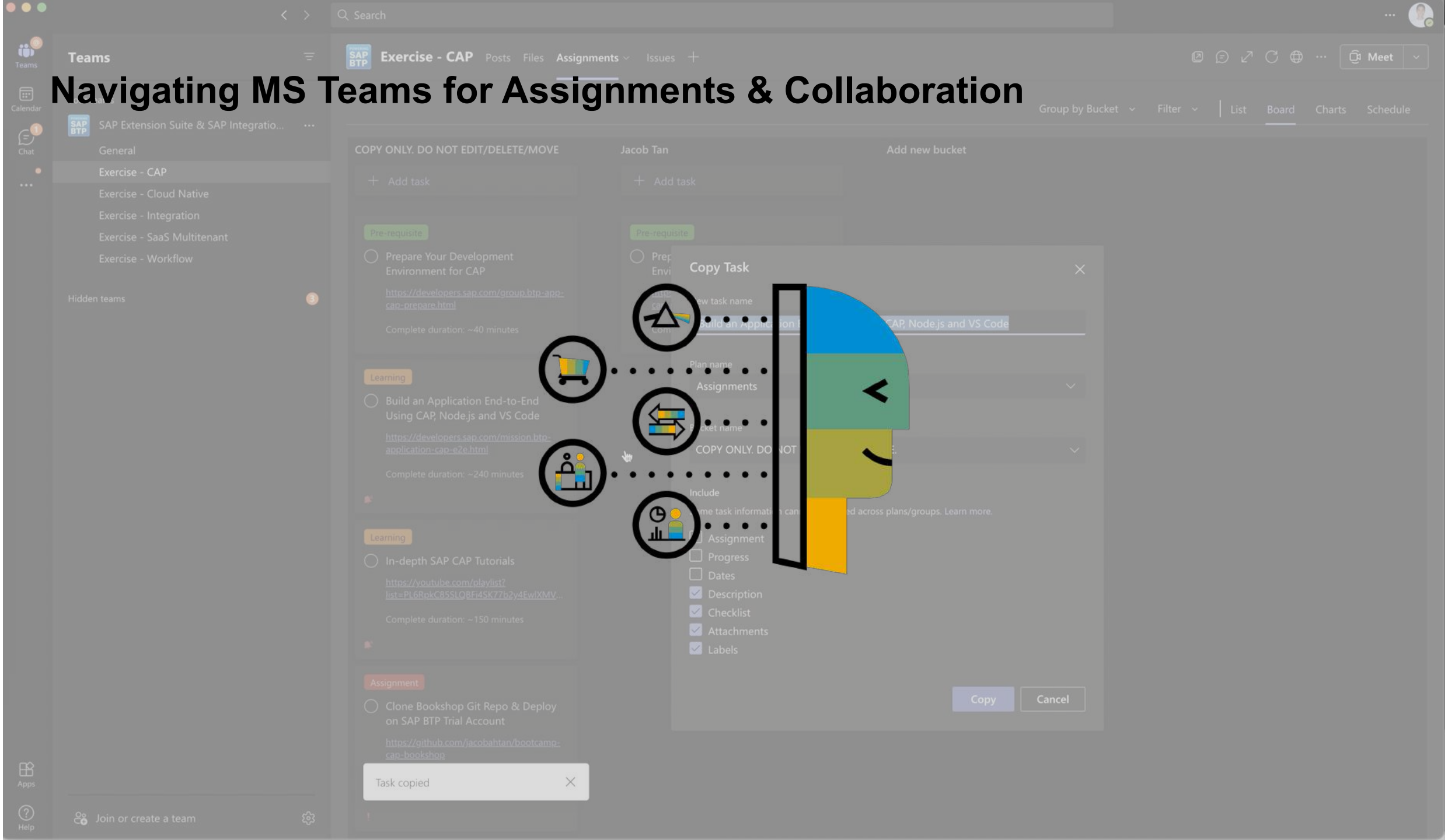
Objectives

For new & seasoned developers to be familarise with SAP Cloud Application Programming model & NodeJS.

Assignment

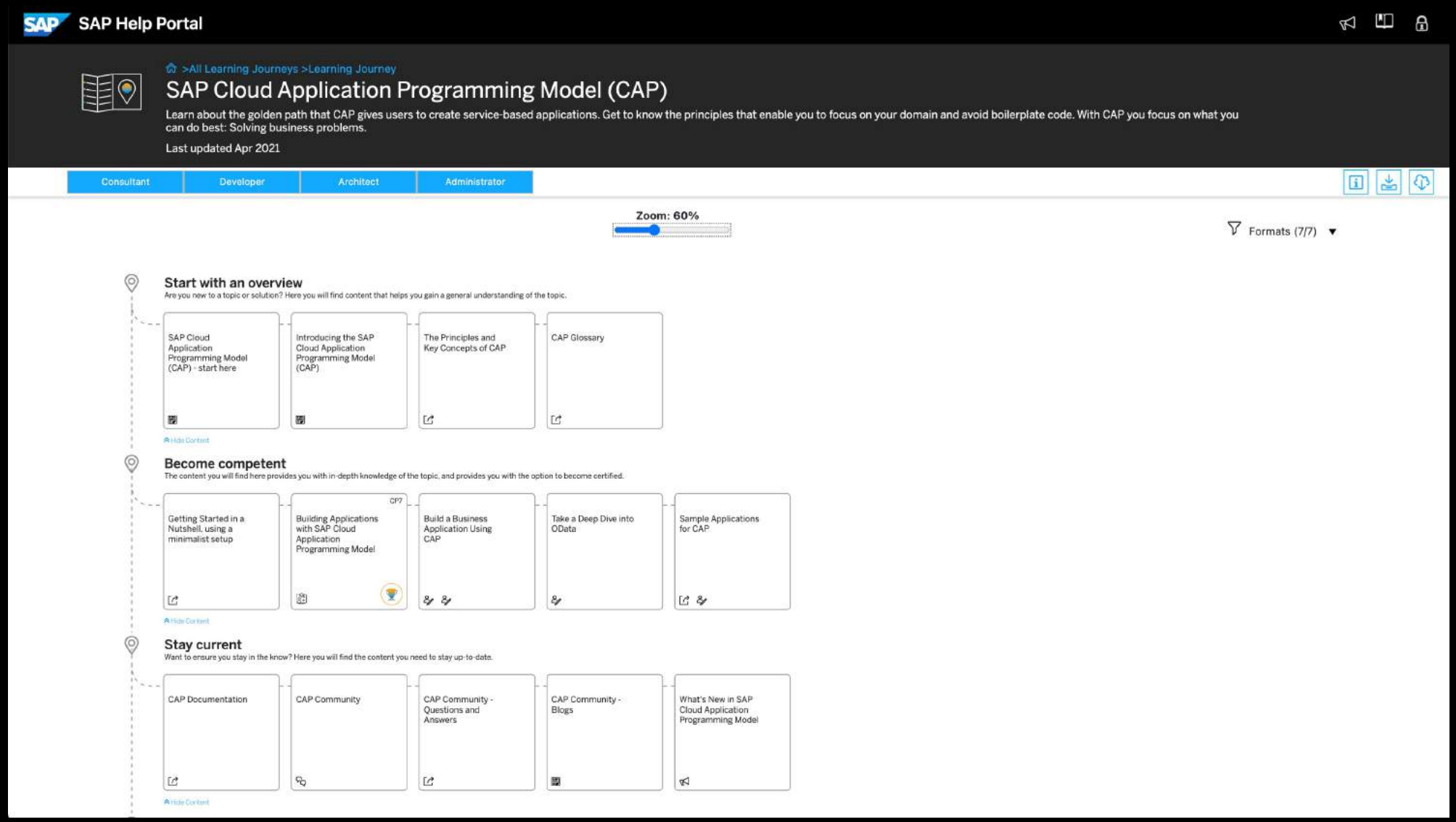
1. Clone Bookshop Git Repository.
2. Deploy to your own SAP BTP Trial Account.
3. Complete & Extend Use Case of your selection from the following choices:
   a. Extend S/4HANA's Sales Order module into existing Bookshop use case as illustrated in the demo.
   b. Extend ANY system with a valid use case of your choice with CAP.
   c. Enhance CAP project of the existing Bookshop solution WITHOUT any extended system.

Navigating MS Teams for Assignments & Collaboration



Resources

SAP Learning Journey on SAP CAP (Link)



Building Applications with SAP Cloud Application Programming Model

openSAP – Self-paced Course

Duration: 4 weeks

Effort required: 4-6 hours per week

Week 1: Introducing SAP Cloud Application Programming Model

Week 2: Development Tasks

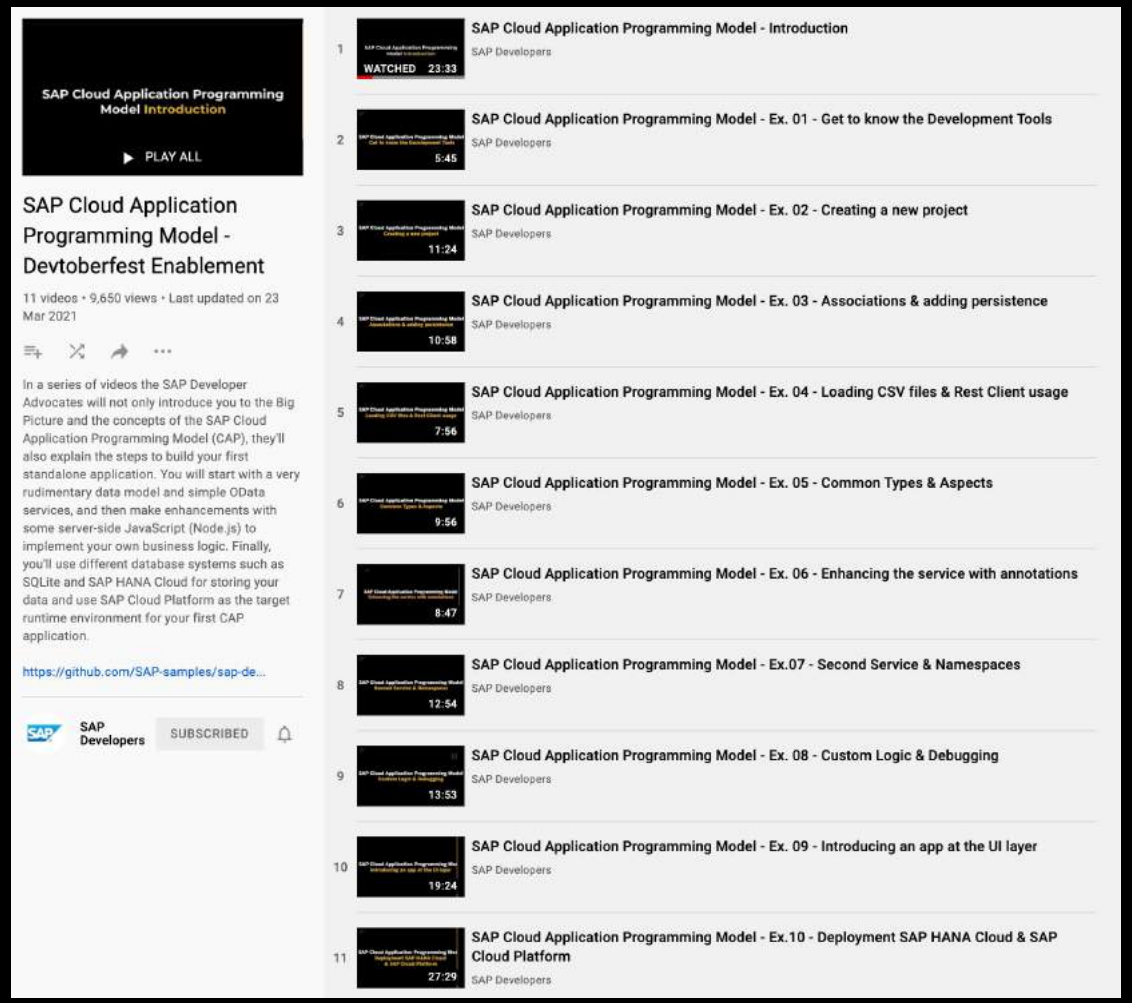
Week 3: Add Enterprise Qualities

Week 4: Extend the Digital Core and Other Topics

Sign up for free: https://open.sap.com/courses/cp7

openSAP
open.sap.com

In-depth CAP focused video tutorials & other resources (Link)



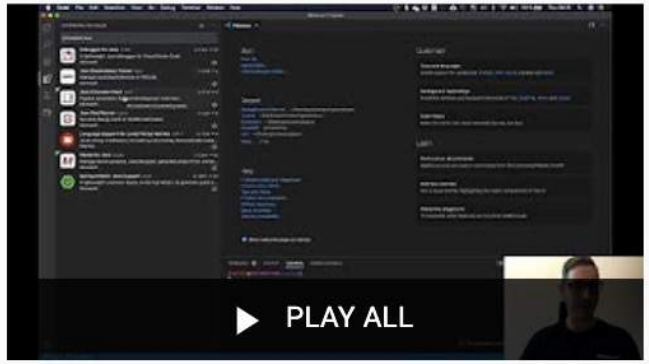
Key Take Away

- Getting Started & Learning about the Dev Tools
- First project based on CAP model
- Adding persistence layer to your app
- Building microservices with annotations
- Adding custom logic and debugging
- Custom UI layer
- Deployment

EXPECTED COMPLETION DURATION

Duration may vary based on technical scenario and Partner expertise level, below we present an expectation only.

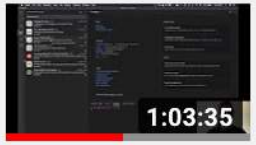
Video Tutorials on CAP Model focusing with Java (Link)



Cloud Application Programming Model - Java

5 videos • 539 views • Last updated on 27 May 2020

Max Streifeneder SUBSCRIBE

1. 

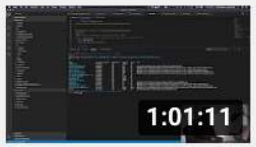
Ep0. - CAPch (catch) up with Java
Max Streifeneder

2. 

Ep1. - CAP Java - Custom Logic
Max Streifeneder

3. 

Ep2. - CAP Java - Database operations and Deployment options
Max Streifeneder

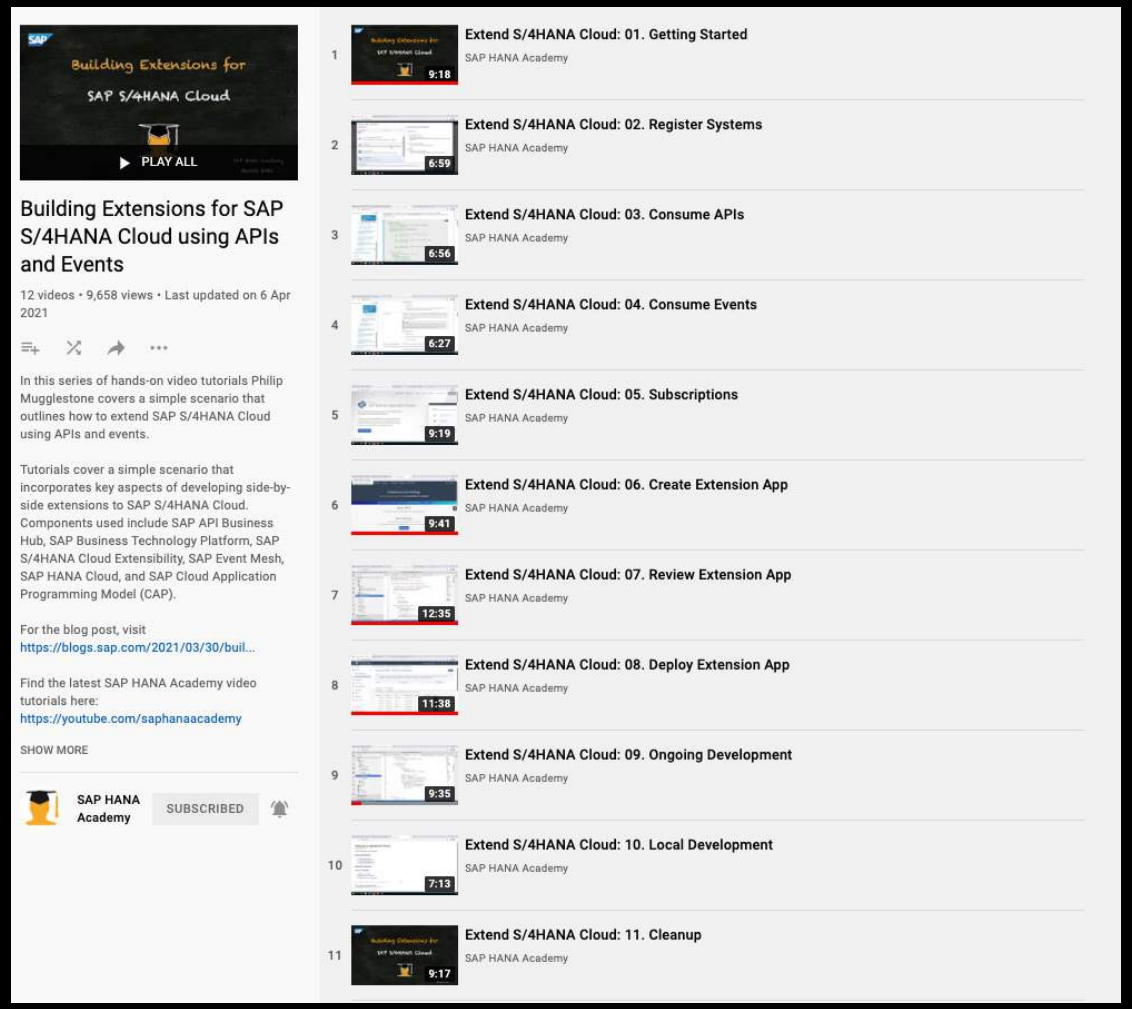
4. 

Ep3. - CAP Java - MTA, CF CLI, Monitoring
Max Streifeneder

5. 

Ep4. - CAP Java - Logging/Monitoring
Max Streifeneder

In-depth S4HANA Cloud with CAP and Cloud SDK focused video tutorials & other resources (Link)



Key Take Away

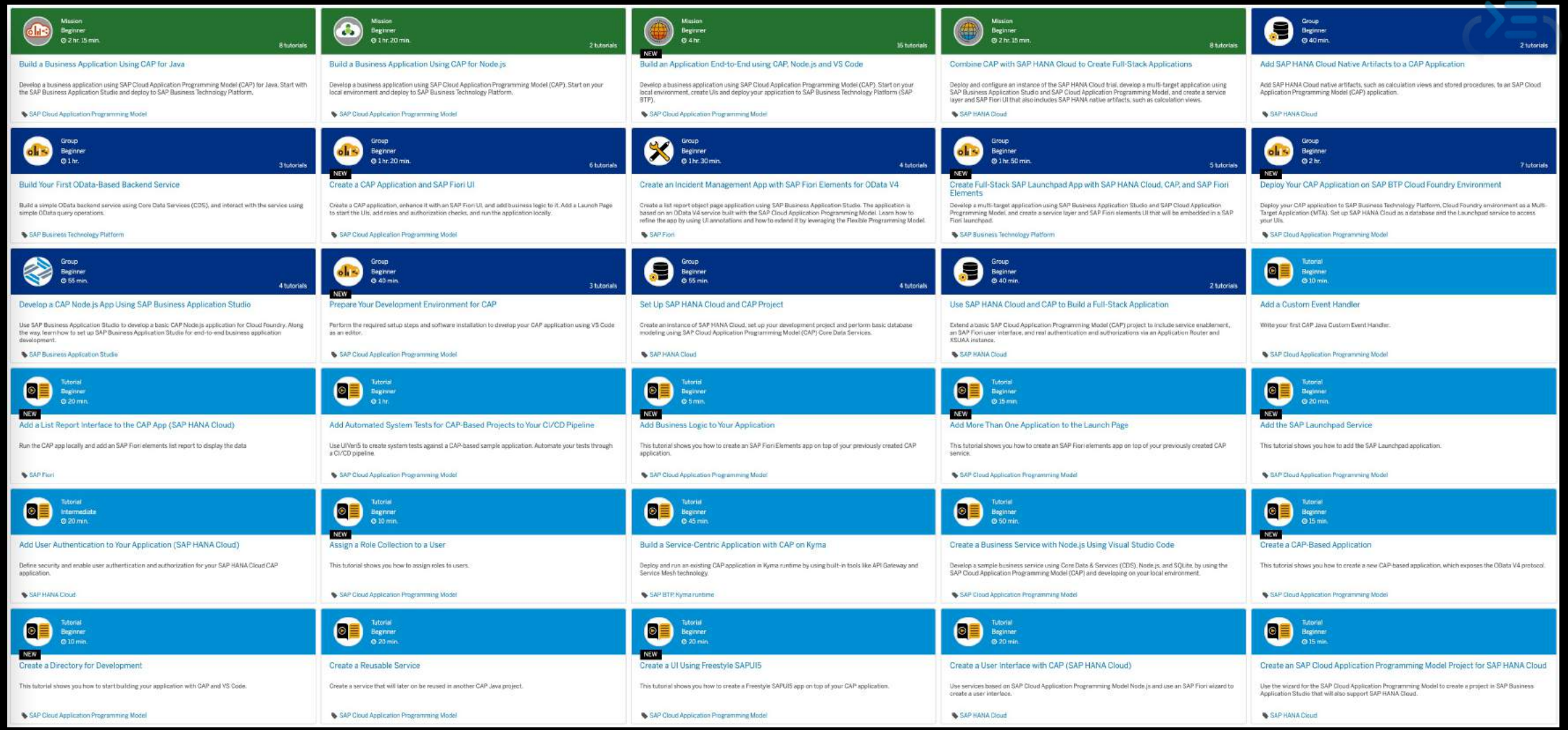
- Side-by-side extensions to S/4 HANA Cloud
- How to consume events from external app
- Creating and deploy an extension app
- Using CAP and Cloud SDK to extend S4 Business Partner

EXPECTED COMPLETION DURATION

Duration may vary based on technical scenario and Partner expertise level, below we present an expectation only.

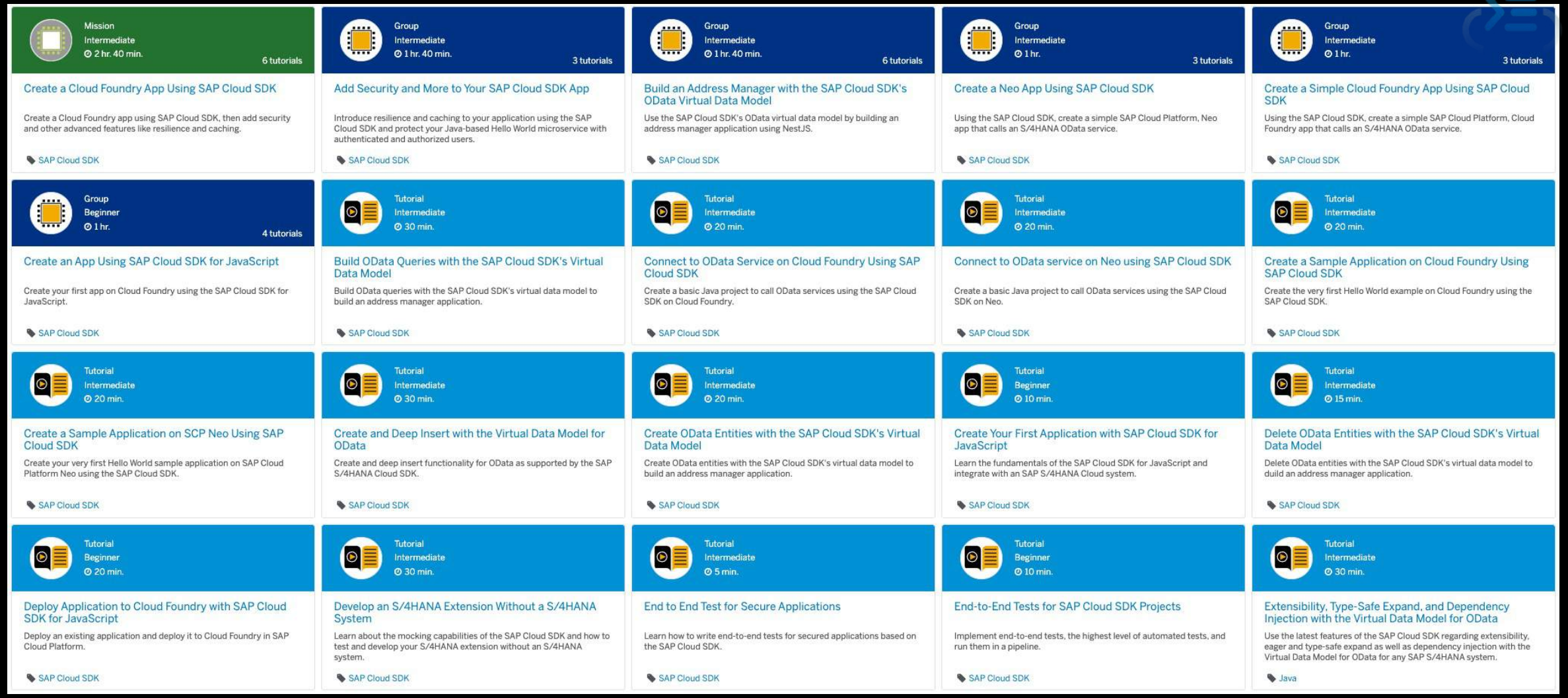
SAP Developer Tutorials focus on SAP CAP (Link)

4 Mission 10 Group 48 Tutorial



SAP Developer Tutorials focus on SAP Cloud SDK (Link)

1 Mission 5 Group 28 Tutorial



Clone the Git Repo from today's demo (Link)

jacobtan89 / bootcamp-cap-bookshop

Bookshop example for Private Partner Bookcamp 2021.

View license

0 stars 0 forks

Star | Unwatch

Code | Issues | Pull requests | Actions | Projects | Wiki | ...

master

jacobtan89 bootcamp ready ... 2 minutes ago 7

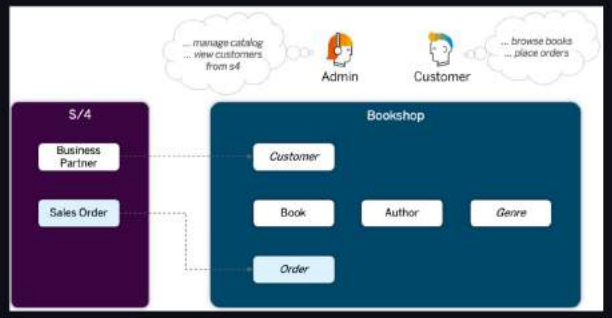
app	1 hour ago
db	1 hour ago
docs	1 hour ago
srv	16 minutes ago
test	1 hour ago
.cdsrc.json	1 hour ago
.gitignore	1 hour ago
LICENSE	1 hour ago
README.md	2 minutes ago
Use Case.png	2 hours ago
mta.yaml	1 hour ago
package-lock.json	1 hour ago
package.json	2 hours ago
xs-security.json	1 hour ago

README.md

Welcome

This project is developed for the purpose of private partner bootcamp. Participants may clone the repository, install/build & run on your local environment or the SAP Business Application Studio. It is a use case to illustrate a simple bookshop management solution extended to S/4HANA Cloud with SAP Cloud SDK. ***To deploy this successful, please make sure you've the prior knowledge of CAP and attended the bootcamp session.***

Use Case Scenario



Pre-requisites Check

Regardless if you're using Visual Studio code or SAP Business Application Studio, please ensure you have these components installed & setup in your local environment.

> Check the following (line-by-line) command if they have already been installed, if not, install them with the respective commands below.

```
node -v
cf --version
cds v
cf plugins
mbt --version
mta --version
```

> Install (line-by-line) on the respective libraries if any one of them above is missing.

```
npm install -g @sap/cds-dk
npm install -g @sap/cds
npm install -g mta
npm install -g mbt
cf install-plugin multiapps
```

README.md

Let's Get Started

1. Clone into a bookshop folder.

```
git clone https://github.com/jacobtan89/bootcamp-cap-bookshop.git bookshop
```

2. Navigate into the ***bookshop*** folder & install the ***required npm dependencies*** declared in the package.json (takes about a few minutes).

```
cd bookshop
npm install
```

3. Run it with `cds watch` in your bookshop folder.

```
cds watch
```

4. `(Optional)` Connecting to your S/4 HANA Cloud System, please follow these steps.

- Create a destination in your SAP BTP account, pointing to your S/4 HANA Cloud system.

> Destination Name: `S4HC`

> Destination URL: https://`<tenant>`.s4hana.ondemand.com

Please note that the above destination name `S4HC` will be used in the Custom Logic file `Line 3` located in srv/admin-service.js. In order to connect your app with the destination defined in your SAP BTP account, you can either create a local file `default-env.json` in your bookshop folder with the `destination` & `xsuaa` service key credentials **OR** simply just run everything in the cloud by packaging with MTA, build and deploy.

5. Package & Build with MTA then Deploy in your SAP BTP account.

- In your bookshop folder, package your app with the MTA Build Tool.

```
mbt build -t ./
```

- Once completed, let's deploy `bookshop_1.0.0.mtar` file in `bookshop` folder, to your SAP BTP Cloud Foundry environment..

```
cf deploy bookshop_1.0.0.mtar
```

If you face a problem with the example application or the description, feel free to create an issue.

Learn More

Learn more about the core concepts at SAP Developer Tutorials on CAP Topic.

Thank you.

Contact information:

Jacob Tan
SAP BTP Solution Architect

Alessandro Biagi
SAP BTP Solution Architect