Useful features of the SAP Cloud SDK for any Cloud-native application

PUBLIC

THE BEST RUN SAP

Disclaimer

The information in this presentation is confidential and proprietary to SAP and may not be disclosed without the permission of SAP. Except for your obligation to protect confidential information, this presentation is not subject to your license agreement or any other service or subscription agreement with SAP. SAP has no obligation to pursue any course of business outlined in this presentation or any related document, or to develop or release any functionality mentioned therein.

This presentation, or any related document and SAP's strategy and possible future developments, products and or platforms directions and functionality are all subject to change and may be changed by SAP at any time for any reason without notice. The information in this presentation is not a commitment, promise or legal obligation to deliver any material, code or functionality. This presentation is provided without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. This presentation is for informational purposes and may not be incorporated into a contract. SAP assumes no responsibility for errors or omissions in this presentation, except if such damages were caused by SAP’s intentional or gross negligence.

All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.

> **“SAP S/4HANA Cloud SDK significantly enhanced the development … we have been able reduce to development by up to 50 percent.”**

[Leonardo De Araujo, CTO, Beyond Technologies]

The Benefits when Connecting to SAP Cloud Platform destination service

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

```
// 1. Get destination credentials
JsonObject destinationCredentials = EnvironmentValueReader.getDestinationCredentials();
String destinationClientId = destinationCredentials.get("clientid").getAsString();
String destinationClientSecret = destinationCredentials.get("clientsecret").getAsString();
```

Retrieve config

```
// 2. Get jwt token to fetch destination configuration
UaaContextFactory destinationUaaContextfactory = UaaContextFactory.factory(xsUaaUrl);
TokenRequest destinationTokenRequest = destinationUaaContextfactory.tokenRequest();
destinationTokenRequest.setGrantType(GrantType.CLIENT_CREDENTIALS);
destinationTokenRequest.setClientId(destinationClientId);
destinationTokenRequest.setClientSecret(destinationClientSecret);
UaaContext destinationXsuaaContext = destinationUaaContextfactory.authenticate(destinationT
CompositeAccessToken destAccessToken = destinationXsuaaContext.getToken();
```

Authenticate

```
// 3. Connecting to the URL to access destination configurations
String destEnv = Constant.DESTINATION_SERVICE_NAME;
URL urlToAccessDestConfig = new URL(destinationCredentials.get("uri").getAsString()
      + "/destination-configuration/v1/subaccountDestinations/" + destEnv);
```

Build request

```
HttpURLConnection urlConnToAccessDestConfig = (HttpURLConnection) urlToAcessDestConfig.openConnection();
urlConnToAccessDestConfig.setConnectTimeout(10000);
urlConnToAccessDestConfig.setReadTimeout(60000);
urlConnToAccessDestConfig.setRequestProperty("Authorization", "Bearer " + destAccessToken);
urlConnToAccessDestConfig.connect();
int status = urlConnToAccessDestConfig.getResponseCode();
```

Connect

```
StringBuilder destSB = new StringBuilder();
// Reading the destination configurations
switch (status) {
case 200:
   BufferedReader br = new BufferedReader(new InputStreamReader(urlConnToAccessDestConfig.getInputStream()));
   String line;
   while ((line = br.readLine()) != null) {
      destSB.append(line + "\n");
   }
   br.close();
}
```

Handle response

Without SDK

```
// Connecting to the destination configurations URL
JSONObject destJSON = new JSONObject(destSB.toString());
String resultUrl = destinationJSONConfiguration.getString("URL");
```

Convert result

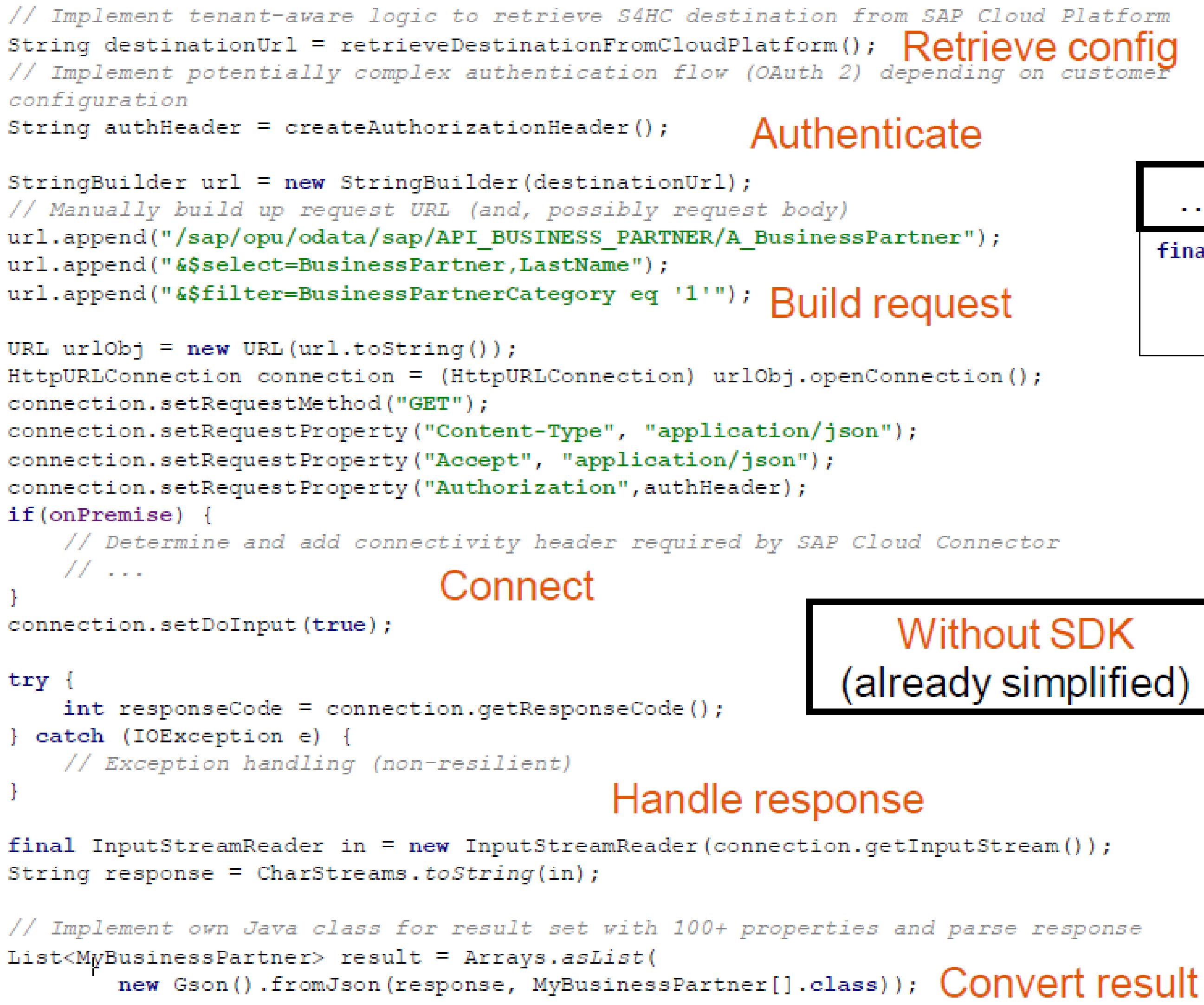
… let developers focus on solving business problems:

```
Destination destination = DestinationAccessor.getDestination(
            Constant.DESTINATION_SVC_MARKETING_CLOUD);
String resultUrl = destination.getUri().toString();
```

With SDK

The Benefits when Connecting to OData Services

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



```
// Implement tenant-aware logic to retrieve S4HC destination from SAP Cloud Platform
String destinationUrl = retrieveDestinationFromCloudPlatform();
// Implement potentially complex authentication flow (OAuth 2) depending on customer
configuration
String authHeader = createAuthorizationHeader();

StringBuilder url = new StringBuilder(destinationUrl);
// Manually build up request URL (and, possibly request body)
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());
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);

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));
```

Retrieve config

Authenticate

Build request

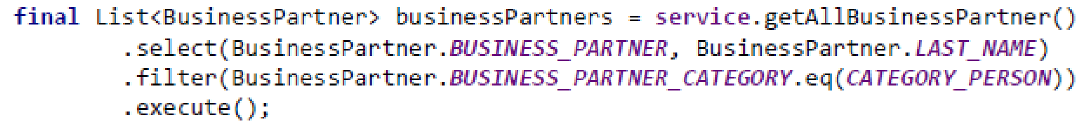
Connect

Handle response

Convert result

Without SDK
(already simplified)

… 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();
```

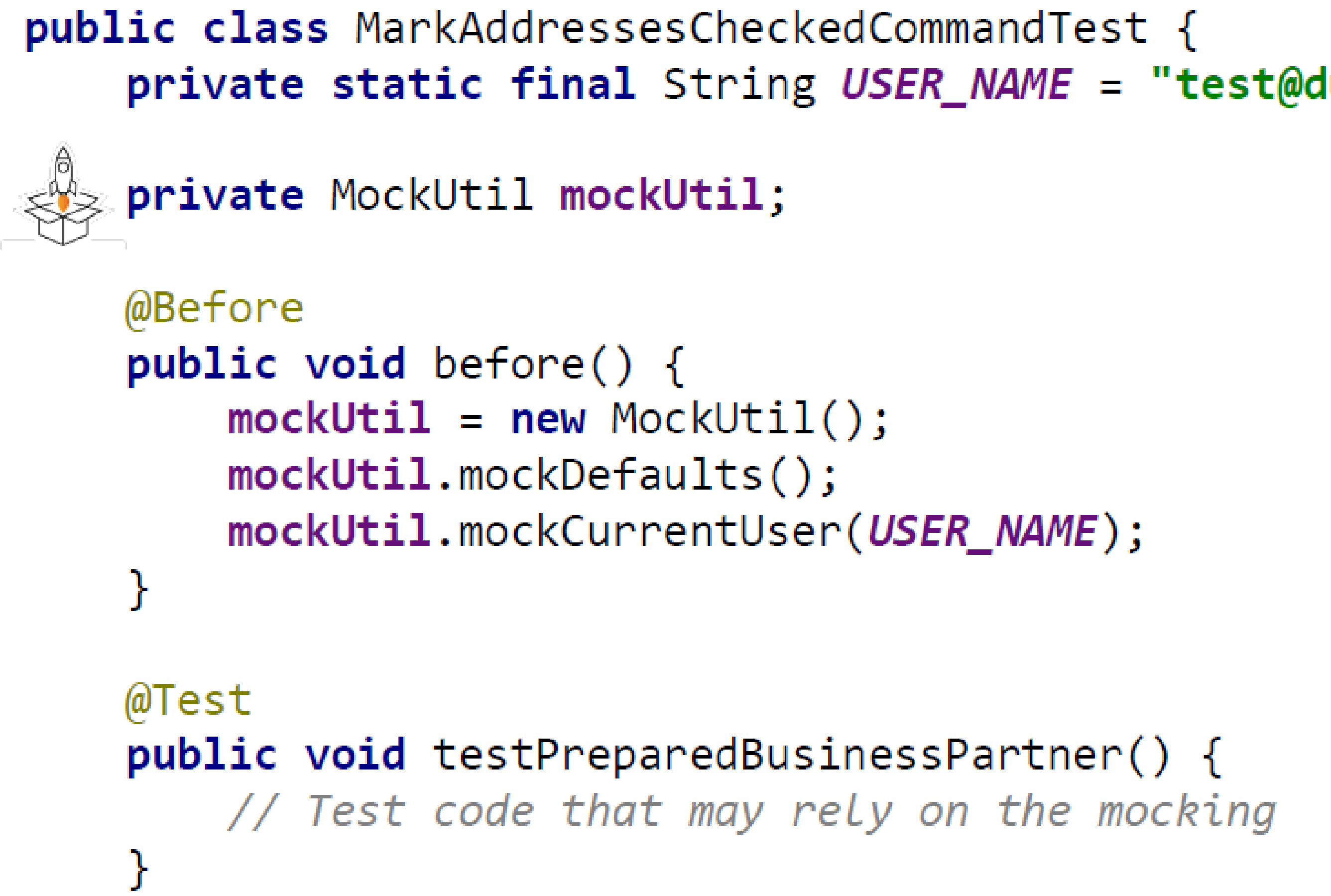
With SDK

Easy, type-safe, and fluent access to CRUD APIs

plus additional advanced features:

- Dependency injection & mocking
- Optimistic concurrency control
- Extensibility
- …

Unit Tests – Basic Setup with the SAP S/4HANA Cloud SDK



```
public class MarkAddressesCheckedCommandTest {
    private static final String USER_NAME = "test@dummy.com";

    private MockUtil mockUtil;

    @Before
    public void before() {
        mockUtil = new MockUtil();
        mockUtil.mockDefaults();
        mockUtil.mockCurrentUser(USER_NAME);
    }

    @Test
    public void testPreparedBusinessPartner() {
        // Test code that may rely on the mocking
    }
```

Unit Tests – Dependencies to External Systems

Mocking the destination service

- SAP S/4HANA Cloud SDK offers extensive support for mocking functionality of the destination service
- MockUtil can be used to return mocked destination that redirects to …
  - Actual test system of external dependency (e.g., test instance of SAP S/4HANA)
  - External mock system that mimics API of external dependency

    ```
    mockUtil.mockErpDestination()
    ```

  - Mock server spun up by test suite (WireMock)

    ```
    mockUtil.mockErpServer()
    ```

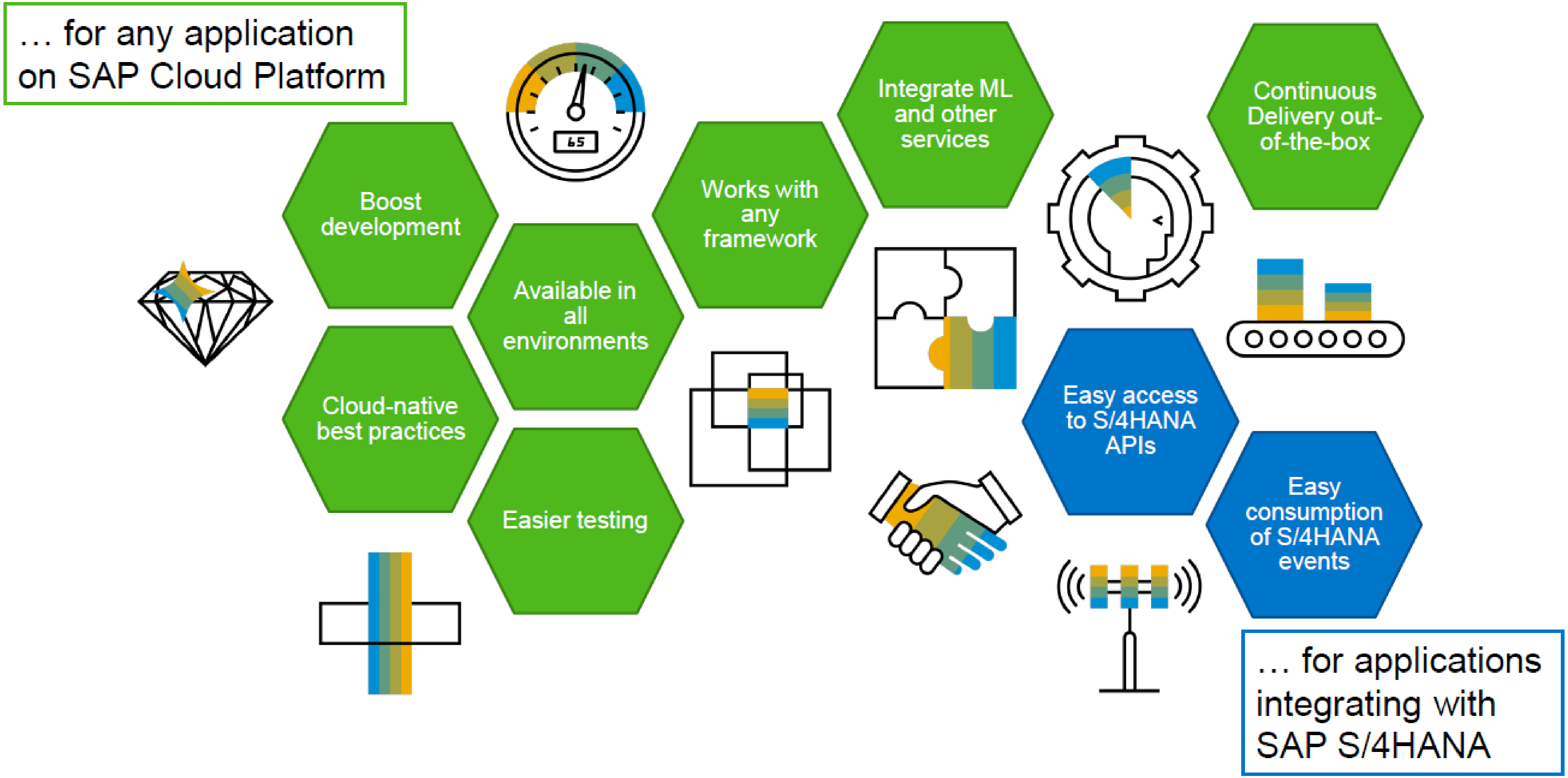
  - No system

    ```
    mockUtil.mockDestination("DestinationName", URI.create(""))
    ```

Mocking the external system

- Independence from an actual SAP system
- Either by pointing the destination service to a test / mock system (see above)
- … or by mocking the calls to the external system itself (easy when using the Java Virtual Data Model of the SAP S/4HANA Cloud SDK)

Benefits of the SAP S/4HANA Cloud SDK



… for any application on SAP Cloud Platform

Boost development

Available in all environments

Cloud-native best practices

Easier testing

Works with any framework

Integrate ML and other services

Continuous Delivery out-of-the-box

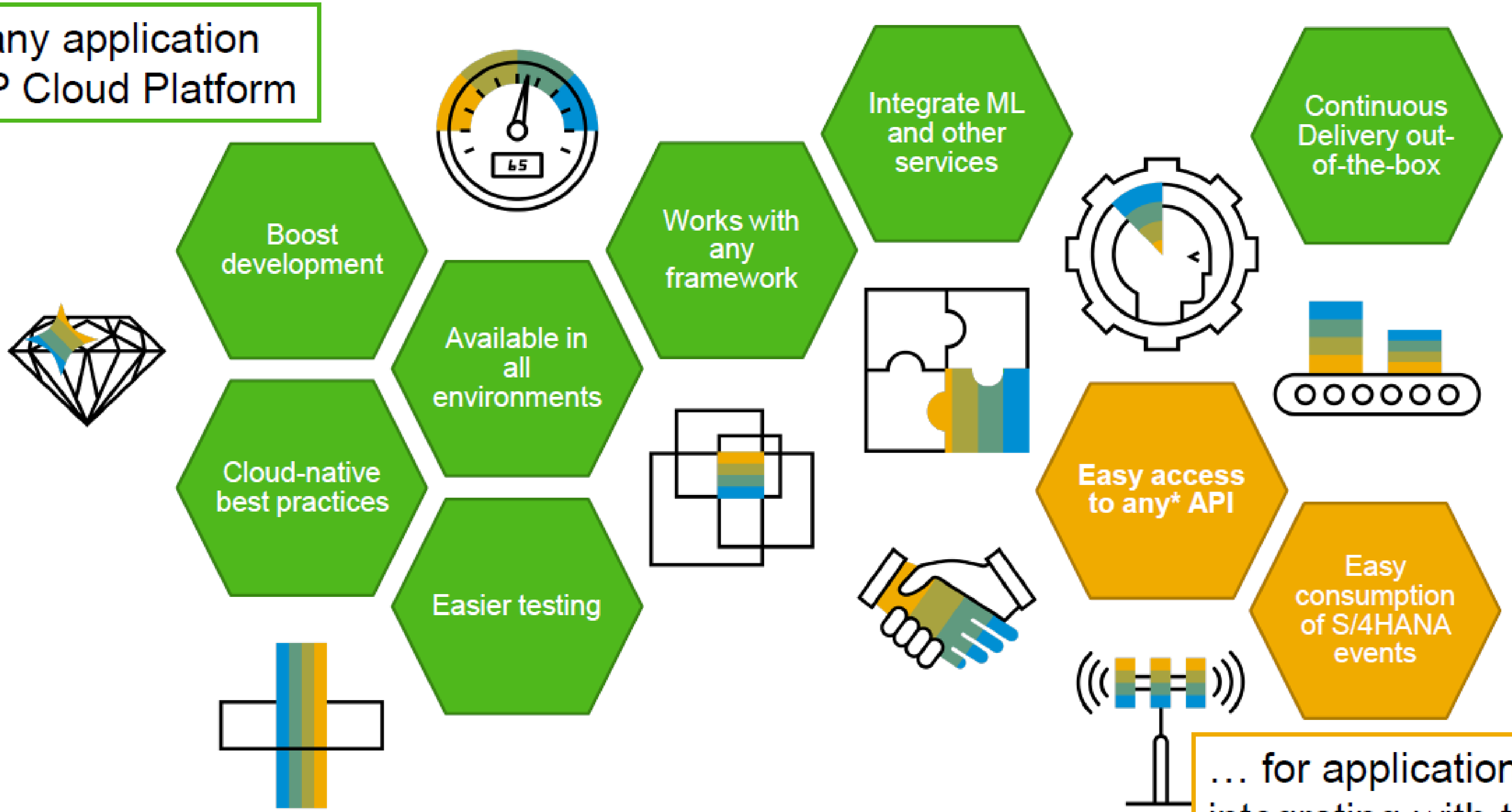
Easy access to S/4HANA APIs

Easy consumption of S/4HANA events

… for applications integrating with SAP S/4HANA

Evolution of the SAP S/4HANA Cloud SDK

… for any application
on SAP Cloud Platform



Boost development

Available in all environments

Cloud-native best practices

Easier testing

Works with any framework

Integrate ML and other services

Continuous Delivery out-of-the-box

Easy access to any* API

Easy consumption of S/4HANA events

… for applications
integrating with the
intelligent enterprise

* Any API, e.g. SuccessFactors, C/4HANA, …

This is the current state of planning and may be changed by SAP at any time without notice.

Demo

Questions & answers

Thank you.

Run Simple

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.
See http://global.sap.com/corporate-en/legal/copyright/index.epx for additional trademark information and notices.