

Guidelines in designing & operating **Enterprise Grade Integration Artifacts**

Raj Chintam – Raj.Chintam@sap.com

PUBLIC

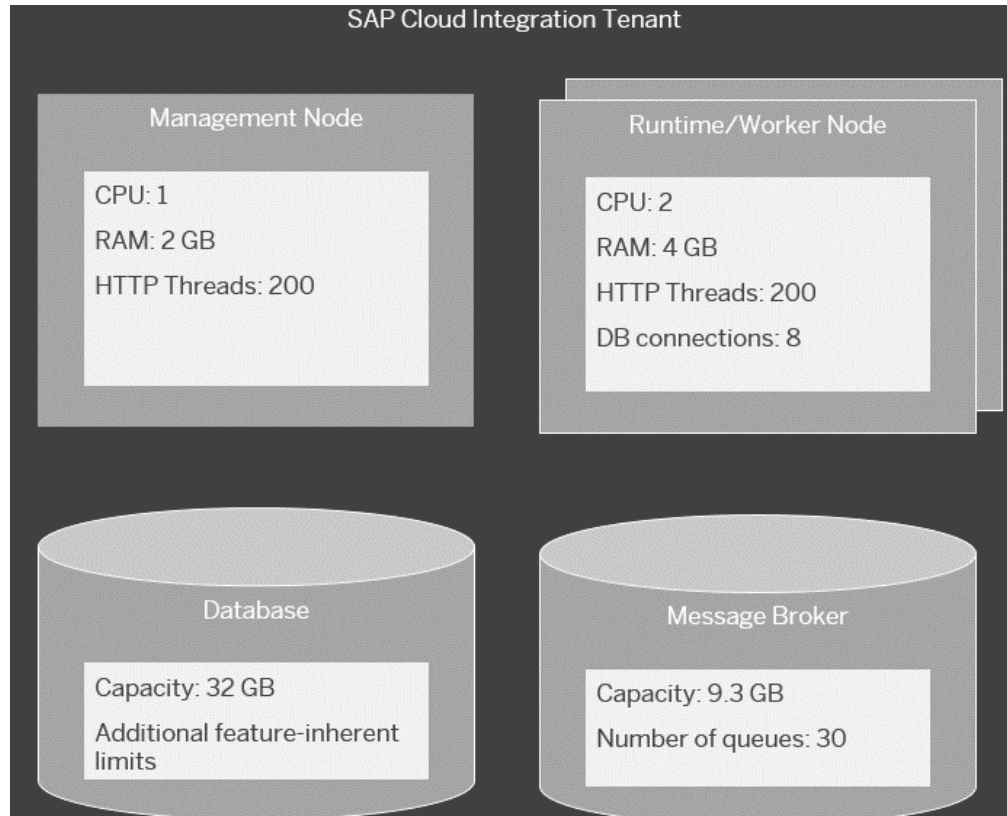
Agenda

- Get strong on Basics
 - What are Basics
 - Understand Boundary Conditions
- Design Guide
 - Fundamentals of Design Tenets
 - Optimize Performance
 - Watchout for memory footprint
 - So Much more.
- Resiliency, Troubleshooting Guide And Use Cases for API's
 - Where to find guide?
 - How to use it?
- How to Automate Failed Messages Re-Processing
 - Alert on Failure
 - Understand available options
 - Store for later pickup

Get Strong on Basics. **And What are Basics ?**

- Exchange Headers & Exchange Properties
- Learn how to handle attachments – in multiples
- Adapters & Properties
- Decouple senders from Processing
- Delta Synchronisation – Local & Global Variables
- Exception Handling
- Persistence – Several Options available including EDA
- Content Modification
- Content Transport
- How communication works between external & internal objects
- How to Transfer Files

Understand Boundaries



Management Node	<ul style="list-style-type: none"> ▪ CPU: 1 ▪ RAM: 2 GB ▪ HTTP threads: 200
Runtime/Worker Node	<ul style="list-style-type: none"> ▪ CPU: 2 ▪ RAM: 4 GB ▪ HTTP threads: 200 ▪ Database connections: 8
Database	<ul style="list-style-type: none"> ▪ Capacity: 32 GB ▪ Additional feature-inherent limits
Message Broker	<ul style="list-style-type: none"> ▪ Capacity: 9.3 GB ▪ Number of queues: 30

Design Guide – Fundamentals of Design Tenets

- High Availability
- Resilience
- Resource Management
- Loose Coupling
- Gracefully handle failures & exceptions
- Ensure maintainability & readability

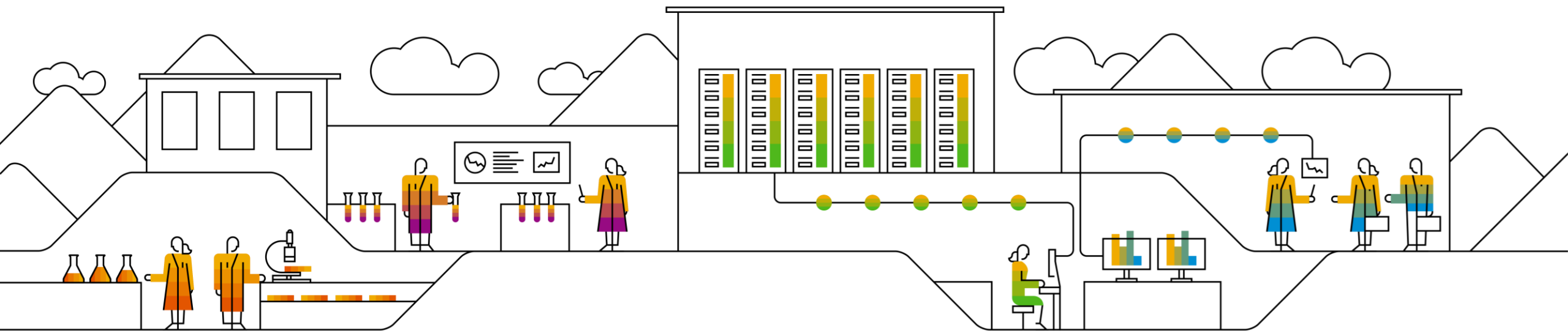
Design Guide – Optimize Performance

- Watchout for Payload Sizes
- Push Messages versus Timer Triggered Messages
- Watch out for Number of Flow steps – every step adds overhead
- Choose Right Adapter. SOAP takes time to syntax validate Vs Http
- Avoid External Calls in Loop. Do Lookup's. Use Hashmap
- Consider Network Latency – choose system at nearest DC
- Weigh options for Memory Vs DB calls
- Use Pagination. Never load all-at-once
- Parallel Vs Sequential Multicast and Splitters
- Authentication is expensive – use Session Reuse
- Use batch API Processing calls whenever possible

Design Guide – Optimize Memory Footprint

- If there are more branches – always end branch via join & gather, empty or reset holders
- If you expect large payload – avoid multicasting
- Large Payloads - Avoid memory intensive transformation – disk space errors
- For large Payloads - Use ByteArray instead of String
- Global Variables – cleanup headers after use
- DOM parsers are memory hogs. Use SAX parser if possible
- To avoid OOM - [Use message streaming](#)

Troubleshooting Guide



Where to find Troubleshooting Guide

The screenshot shows the SAP Integration Suite dashboard. At the top, there is a navigation bar with the SAP logo and 'Integration Suite' text. Below this is a large blue banner with the text 'Welcome to SAP Integration Suite' and 'Simplify and accelerate enterprise integration'. A dropdown menu is open in the top right corner, showing options: 'About', 'Online Guide', 'Troubleshooting Guide' (highlighted with a red box), and 'Logout'. A red arrow points from the 'Troubleshooting Guide' option to the right. Below the banner, there is a 'Recent' section with the text 'View recently accessed artifacts.' and two columns of integration artifacts. The first column is 'Integrations and APIs' and the second is 'API Proxies'. Each artifact has a status indicator (Deployed or Not Deployed). At the bottom, there is a 'Monitoring' section with the text 'Analyze the usage and performance of your artifacts.' and three summary cards: 'Integrations and APIs' (Total 63, Error 2), 'API Proxy Traffic' (Total Calls 0), and 'API Proxy Errors' (Total API Proxy Errors 0). The 'Day' tab is selected in the monitoring section.

Integrations and APIs	API Proxies
Get_Employee_Data_IntUsr01 Integration Flow Draft Deployed	HTTPBIN_Anything REST 1 Deployed
MM_CompoundEmployee Message Mapping 1.0.2 Not Deployed	Get_Users_SAML REST 1 Deployed
VM_Incoterms Value Mapping 1.0.1 Deployed	Metering_APIs_CF REST 1.0 Deployed
Number_Conversion Integration Flow 1.0.1 Not Deployed	PP_CI SOAP 1 Deployed

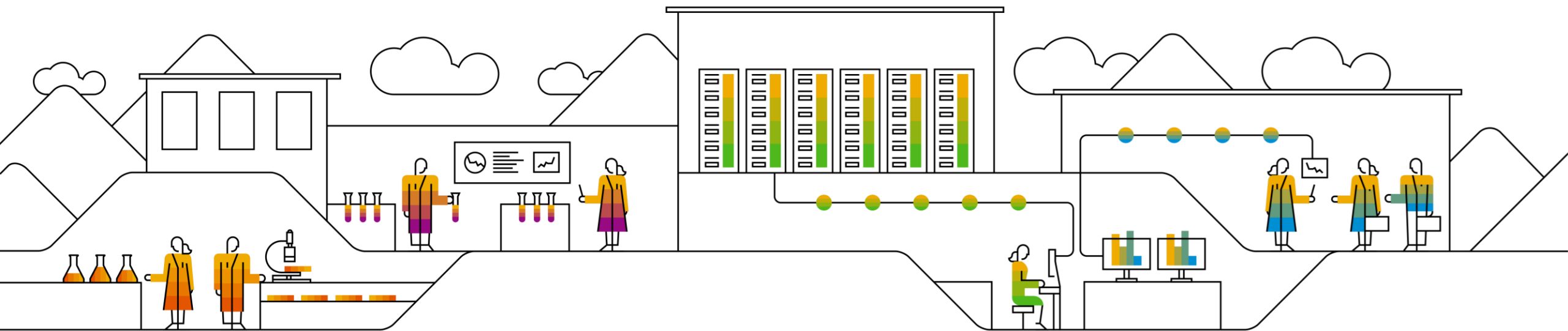
Integrations and APIs	API Proxy Traffic	API Proxy Errors
Based on message processing artifacts Total: 63 Error: 2	Based on API Proxy Calls Total Calls: 0	Based on number of failure Total API Proxy Errors: 0

<https://ga.support.sap.com/dtp/viewer/#/tree/2237/actions/28748>

Security, Credentials and Keystores

- User credentials – CRUD operations
- OAuth2 client credentials – CRUD operations
- Certificates – upload / download
- Key pair – create / upload
- Certificate chains - change / download
- Keystore entries – get metadata / rename / delete / get OpenSSH public key
- Keystore – import / backup / export public content /
- Keystore history – get backed up entries / restore backed up entries
- Certificate-to-User-mapping – CRUD operations*

Uses cases for APIs



Use Cases – DevOps / CI/CD

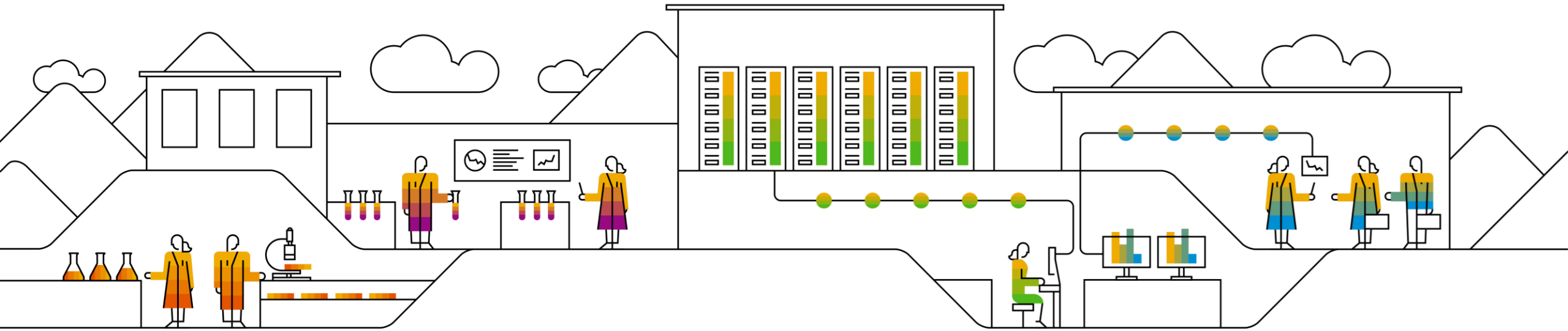
- Create an automated test pipeline for your integration content
 1. Upload content from a source repository
 2. Adapt configurations
 3. Deploy integration flows
 4. Get endpoints
 5. Send messages
 6. Check MPL status

Use Cases – **Integration Content Transport across Tenants**

- Transport across multiple tenants is possible using tools like TMS or CTS+ or via manual export / import
- Another option is to use the OData APIs for Integration Content
- Transport can be done on package level as well as on integration flow level

Use Cases – Certificate Expiry Notification

- Certificates and private keys have an expiry date. Once this is exceeded your integration scenarios will fail.
- So an upfront notification about keystore entries expiring soon might be useful.
 1. Use the keystore entries OData API to fetch all items
 2. Check the expiry date
 3. Compare against the current date
 4. In case the difference is below 14 days send a notification for example via mail



Automate Failed Message Reprocessing

Available options – JMS, Datastore, EM, Memory

PUBLIC

Alerting on Scenario Failure

- In case of message failures the sender will receive an error and message processing log is in *error* state
- But:
 - a. not every scenario is synchronous (async, scheduler, broker)
 - b. the monitor is not checked regular, as you expect a running scenario to continue being succesful
- In order to get to know about failures, build an own notification mechanism
- Use the Message Processing Log OData API to fetch failed messages and send a notification via mail

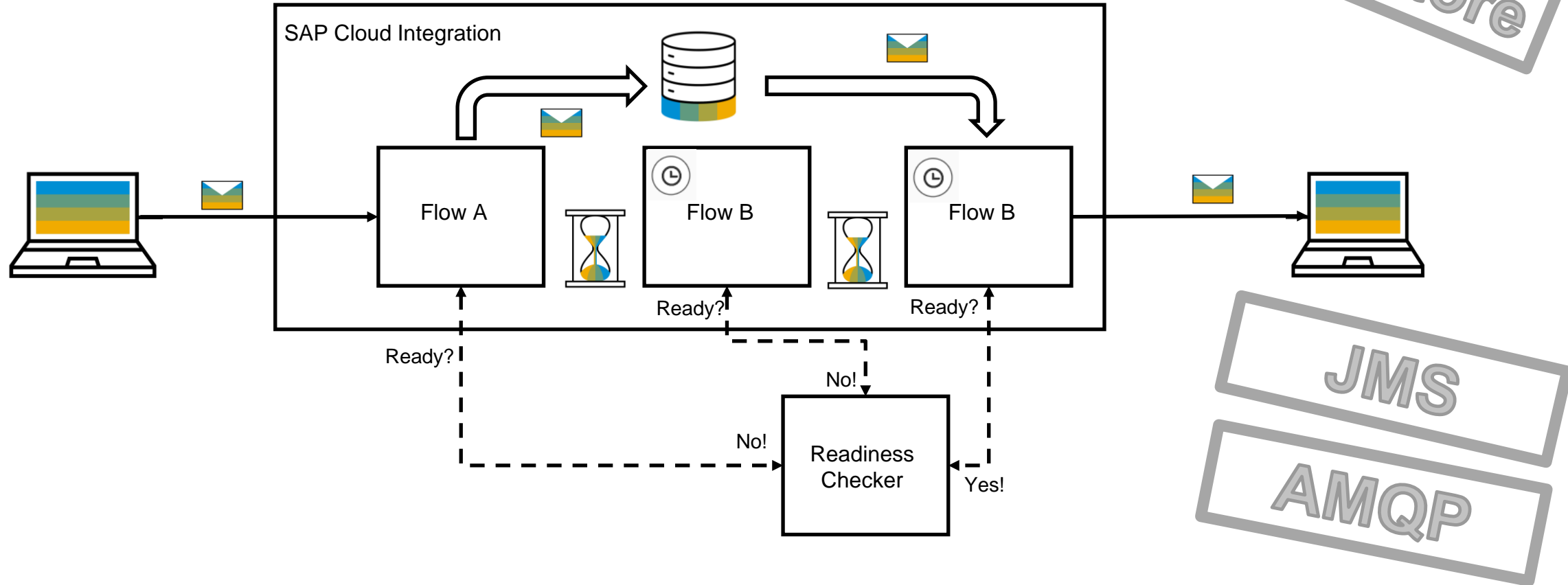
Retry of failed Messages

- Build an integration flow with two endpoints, one for productive usage, the other one for retry (secured by a separate role)
- At runtime, in case of error store the incoming message as MPL Attachment
- Via the MPL OData API check for failed integration flow executions
- Check the error information using the MPL OData API to see if a retry makes sense
- If yes, download the Attachment using the MPL Attachment OData API
- Get the retry endpoint of the failed integratoin flow and push the message again to the flow

Store for later pickup / delay messages

Variables

Datastore

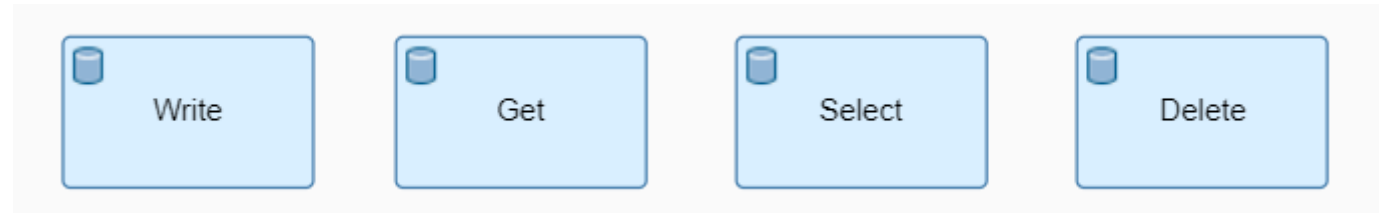


JMS

AMQP

- Parking of message because backend is not ready yet
- Waiting for a state change before continue message processing

Datastore



Storage location:

CI database

What data is stored:

Body (+ headers optionally)

Access by:

1. One certain integration flow
2. Any integration flow

Retention period:

Up to 180 days, default 90 days

Explicit deletion possible:

1. Via Delete operation
2. Via Monitor

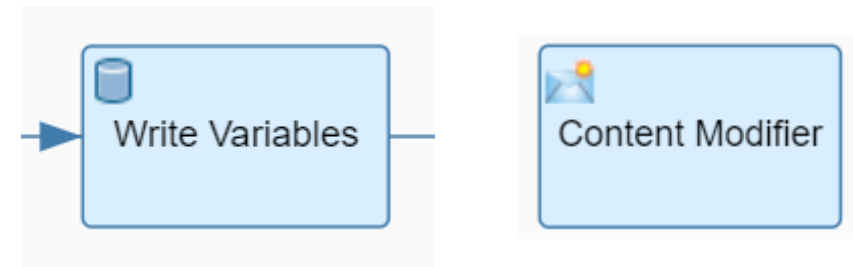
Storage size

32GB total storage size of CI tenant database

Specialities

Requires a database connection

Global & Local Variables



Storage location:

CI database

What data is stored:

Anything

Access by:

1. One certain integration flow (= Local Variable)
2. Any integration flow on the tenant (= Global Variable)

Retention period:

400 days

Explicit deletion possible:

Via Monitor

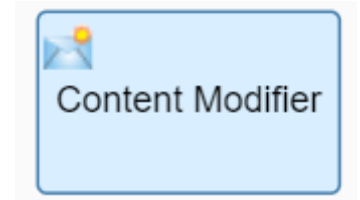
Storage size

32GB total storage size of CI tenant database

Specialities

Requires a database connection

Header & Properties



Storage location:	Memory
What data is stored:	Anything
Access by:	The current integration flow execution
Retention period:	Integration flow execution duration
Explicit deletion possible:	Via Content Modifier
Storage size	4GB memory per worker node

AMQP Adapter

Storage location:

Message Queue/Topic on any external Message Broker

What data is stored:

Body

Access by:

1. Any integration flow with an AMQP channel pointing to the queue/topic
2. Every external client with access to the Broker

Retention period:

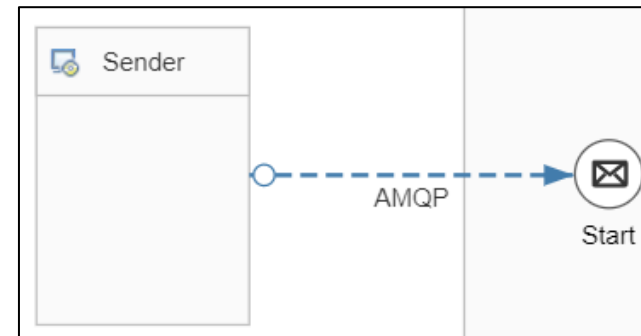
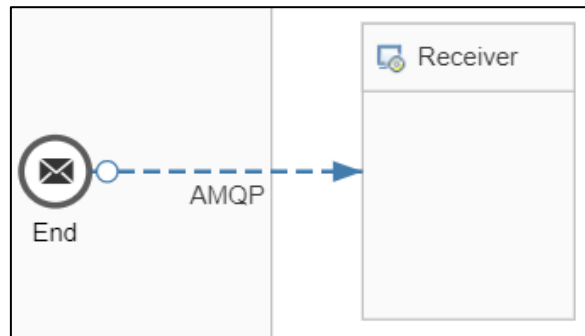
Depends on Broker configuration

Explicit deletion possible:

1. Removed when consumed successfully
2. Via Broker if possible

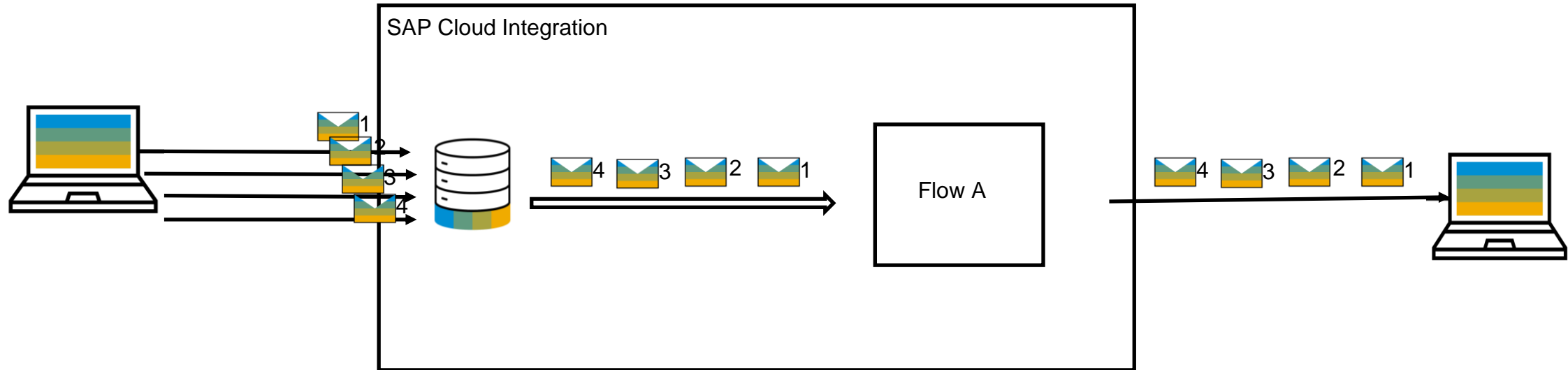
Storage size

Depends on Broker configuration



At Least Once in Order II

AMQP



- Message Brokers like SAP Event Mesh do have a capability called Exclusive queues
- Consumers can receive those messages in Order
- No parallelization applicable
- Exactly once not guaranteed as Flow A could run into errors after sending a message

Re-Process – Recap Slides

0. If you don't want completed status - You can mark the message as failed by using Error End Event
1. Automatic Retry mechanism is available for certain adapters like - JMS queues & XI adapter
2. You can Use Odata V2 API to retrieve Failed message and send notifications
3. you can use Datastore to store headers & body of incoming payload to reprocess again. message available for 90 days

Persist step – audit log

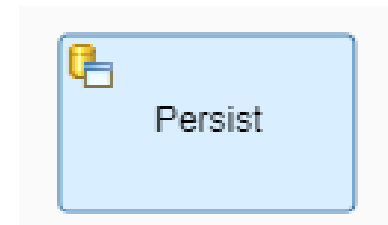
- Persistence on the CI tenant database
- Storage of messages for auditing purposes
- Only successful messages are stored
- Only body is stored
- No storage in case of errors (except node crash like OOM)
- Independent of JDBC transaction
- Stays persisted until expired
- Retention time is bound to the message processing log, per default 30 days
- No monitor available
- OData API available

Persist Message

General **Processing**

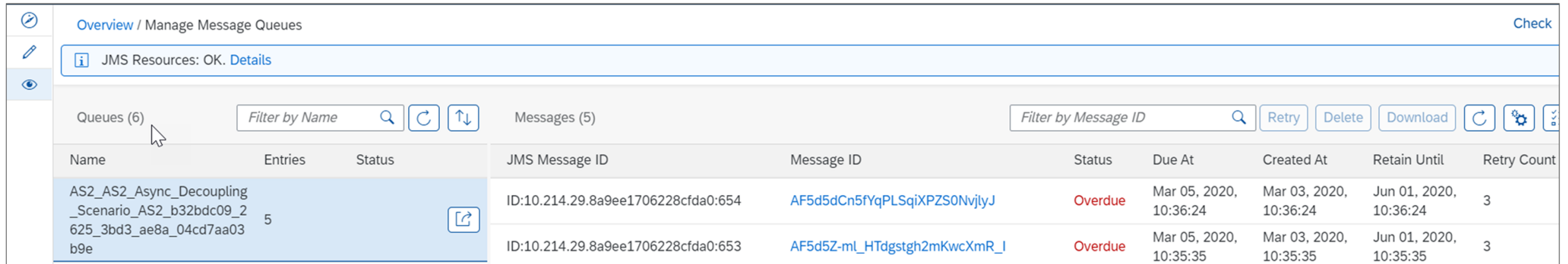
*Step ID:

Encrypt Stored Message



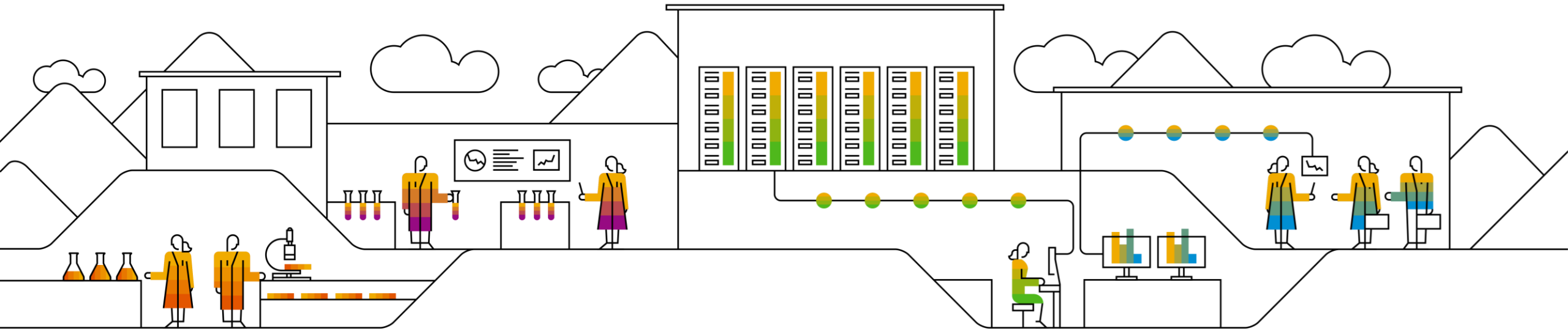
JMS Monitor

- Queues can be checked for the messages
- Check number of retries and next retry
- Trigger an immediate retry of the message
- Queues and Entries can be deleted manually
- [JMS Resource Limits and Optimizing their Usage](#)



The screenshot displays the 'Overview / Manage Message Queues' interface. It features a navigation bar with a 'Check' button and a status bar indicating 'JMS Resources: OK. Details'. Below this, there are two main sections: 'Queues (6)' and 'Messages (5)'. The 'Queues (6)' section includes a search filter 'Filter by Name' and a refresh button. The 'Messages (5)' section includes a search filter 'Filter by Message ID' and buttons for 'Retry', 'Delete', 'Download', and a refresh button. The main table lists the following data:

Name	Entries	Status	JMS Message ID	Message ID	Status	Due At	Created At	Retain Until	Retry Count
AS2_AS2_Async_Decoupling_Scenario_AS2_b32bdc09_2_625_3bd3_ae8a_04cd7aa03b9e	5		ID:10.214.29.8a9ee1706228cfda0:654	AF5d5dCn5fYqPLSqjXPZS0NvjlyJ	Overdue	Mar 05, 2020, 10:36:24	Mar 03, 2020, 10:36:24	Jun 01, 2020, 10:36:24	3
			ID:10.214.29.8a9ee1706228cfda0:653	AF5d5Z-mL_HTdgstgh2mKwcXmR_I	Overdue	Mar 05, 2020, 10:35:35	Mar 03, 2020, 10:35:35	Jun 01, 2020, 10:35:35	3



Scripting in SAP Cloud Integration

Dos and Don'ts

PUBLIC

When / Why to use scripting?

- The graphical flow steps are built as generic as possible and cannot cover every use case
- In case multiple flow steps can be combined in a script the performance might be better
- Some transformations are very complex and need scripting
- External libraries provide features and functions that you want to re-use
- Scripts can be tested in external tools like Eclipse or GroovyIDE
- Some error details can be only retrieved via script
- You can hide secrets from Tracing using them in scripts
- Some Cloud Integration features require the use of scripts

Script Basics

Script Skeleton Groovy

```
import com.sap.gateway.ip.core.customdev.util.Message;

def Message processData(Message message) {

    return message;
}
```

Script Skeleton Javascript

```
importClass(com.sap.gateway.ip.core.customdev.util.Message);

function processData(message) {

    return message;
}
```

- You can define own helper methods or script functions
- *processData* is called by default if not specified otherwise in the script step

Script API and Generic API library download:

<https://tools.hana.ondemand.com/#cloudintegration>

Script documentation:

<https://help.sap.com/viewer/368c481cd6954bdfa5d0435479fd4eaf/Cloud/en-US/03b32eb2c5c249f0a59bcd27c44d1e4e.html>

Access / Modifications of headers/properties/body

```
def Message processData(Message message) {  
    //Body  
    def body = message.getBody();  
    message.setBody(body + "Body is modified");  
  
    //Headers  
    def headerMap = message.getHeaders();  
    def value = map.get("oldHeader");  
    message.setHeader("oldHeader", value + " overwritten");  
    message.setHeader("newHeader", "myNewHeaderValue");  
  
    //Properties  
    def propMap = message.getProperties();  
    value = map.get("oldProperty");  
    message.setProperty("oldProperty", value + " overwritten");  
    message.setProperty("newProperty", "myNewPropValue");  
  
    return message;  
}
```

The body will be returned in what data type it is available at runtime.

If you want to get the Body in a certain type:

```
message.getBody(java.lang.String);  
message.getBody(java.io.Reader);  
message.getBody((byte[]).class);
```

MPL properties

```
def messageLog = messageLogFactory.getMessageLog(message);  
  
messageLog.addCustomHeaderProperty("myCustHeaderProp1", "1");  
messageLog.addCustomHeaderProperty("myCustHeaderProp2", "2");  
messageLog.addCustomHeaderProperty("myCustHeaderProp2", "3");  
messageLog.addCustomHeaderProperty("aPropStartingWithA", "4");  
  
messageLog.setStringProperty("Greeting", "Hello World!");
```

```
Message Processing Log:  
StartTime           = Mon Oct 12 21:31:16.542 UTC 2020  
StopTime            = Mon Oct 12 21:31:18.826 UTC 2020  
OverallStatus       = COMPLETED  
MessageGuid         = AF-EyyQH92QsJXnQoSoQ0hYzNtgt  
ChildCount          = 0  
ChildrenCounter     = 4  
ContextName         = script_session  
CorrelationId       = AF-EyySe2Xw6BCYP6hsDbFP3Qf2k  
CustomHeaderProperties = {aPropStartingWithA=[4], myCustHeaderProp1=[1], myCustHeaderProp2=[2, 3]}  
ID                  = "12345"  
IntermediateError   = false  
Node                 = vsa7497420  
OriginComponentName = CPI_m6301  
PreviousComponentName = CPI_m6301  
ProcessId            = d9147b86cef7d4f7f504e28d86c15cf236273fde  
ReceiverId           = TEST_Receiver  
SenderId             = 'TEST_Sender'  
TransactionId        = 4fee65be01084772810d4b2e1c143b17  
ReceiverIds [       =  
  TEST_Receiver  
]
```

customHeaderProperties:

- Visible at log header
- Visible for all log levels
- Sorted by alphabetical order

Typed Properties:

- Visible in the flow step details
- Only visible for log level DEBUG and TRACE

```
Processing exchange ID-vsa7497420-46367-1600900620805-36-2:  
StartTime           = Mon Oct 12 21:32:35.209 UTC 2020  
StopTime            = Mon Oct 12 21:32:35.457 UTC 2020  
ChildCount          = 3  
Greeting             = Hello World!  
ModelStepId         = CallActivity_7  
StepId              = CallActivity_7  
Activities [         =  
  {Activity=setHeader[scriptFile], StartTime=Mon Oct 12 21:32:35.209 UTC 2020, StopTime=Mon Oct 12 21:32:35.210 UTC 2020, StepId=CallActivity_7, TransactionId=4fee65be01084772810d4b2e1c143b17, ReceiverId=TEST_Receiver, SenderId='TEST_Sender', ContextName=script_session, CorrelationId=AF-EyySe2Xw6BCYP6hsDbFP3Qf2k, MessageGuid=AF-EyyQH92QsJXnQoSoQ0hYzNtgt, OverallStatus=COMPLETED, StopTime=Mon Oct 12 21:32:35.211 UTC 2020, StartTime=Mon Oct 12 21:32:35.209 UTC 2020, ChildCount=3, ChildrenCounter=4, IntermediateError=false, Node=vsa7497420, OriginComponentName=CPI_m6301, PreviousComponentName=CPI_m6301, ProcessId=d9147b86cef7d4f7f504e28d86c15cf236273fde, ReceiverId=TEST_Receiver, SenderId='TEST_Sender', TransactionId=4fee65be01084772810d4b2e1c143b17, ID="12345"}  
  {Activity=setHeader[scriptFileType], StartTime=Mon Oct 12 21:32:35.210 UTC 2020, StopTime=Mon Oct 12 21:32:35.211 UTC 2020, StepId=CallActivity_7, TransactionId=4fee65be01084772810d4b2e1c143b17, ReceiverId=TEST_Receiver, SenderId='TEST_Sender', ContextName=script_session, CorrelationId=AF-EyySe2Xw6BCYP6hsDbFP3Qf2k, MessageGuid=AF-EyyQH92QsJXnQoSoQ0hYzNtgt, OverallStatus=COMPLETED, StopTime=Mon Oct 12 21:32:35.211 UTC 2020, StartTime=Mon Oct 12 21:32:35.210 UTC 2020, ChildCount=3, ChildrenCounter=4, IntermediateError=false, Node=vsa7497420, OriginComponentName=CPI_m6301, PreviousComponentName=CPI_m6301, ProcessId=d9147b86cef7d4f7f504e28d86c15cf236273fde, ReceiverId=TEST_Receiver, SenderId='TEST_Sender', TransactionId=4fee65be01084772810d4b2e1c143b17, ID="12345"}  
  {Activity=bean[ref:scriptprocessor method:process], StartTime=Mon Oct 12 21:32:35.211 UTC 2020, StopTime=Mon Oct 12 21:32:35.211 UTC 2020, StepId=CallActivity_7, TransactionId=4fee65be01084772810d4b2e1c143b17, ReceiverId=TEST_Receiver, SenderId='TEST_Sender', ContextName=script_session, CorrelationId=AF-EyySe2Xw6BCYP6hsDbFP3Qf2k, MessageGuid=AF-EyyQH92QsJXnQoSoQ0hYzNtgt, OverallStatus=COMPLETED, StopTime=Mon Oct 12 21:32:35.211 UTC 2020, StartTime=Mon Oct 12 21:32:35.211 UTC 2020, ChildCount=3, ChildrenCounter=4, IntermediateError=false, Node=vsa7497420, OriginComponentName=CPI_m6301, PreviousComponentName=CPI_m6301, ProcessId=d9147b86cef7d4f7f504e28d86c15cf236273fde, ReceiverId=TEST_Receiver, SenderId='TEST_Sender', TransactionId=4fee65be01084772810d4b2e1c143b17, ID="12345"}  
]  
Attachments [       =  
  {AttachmentName=attachmentName, AttachmentURI=sap-it-res:msg:c00rdjnud6:dd58bbd6-115f-4011-8000-000000000000, TransactionId=4fee65be01084772810d4b2e1c143b17, ReceiverId=TEST_Receiver, SenderId='TEST_Sender', ContextName=script_session, CorrelationId=AF-EyySe2Xw6BCYP6hsDbFP3Qf2k, MessageGuid=AF-EyyQH92QsJXnQoSoQ0hYzNtgt, OverallStatus=COMPLETED, StopTime=Mon Oct 12 21:32:35.211 UTC 2020, StartTime=Mon Oct 12 21:32:35.211 UTC 2020, ChildCount=3, ChildrenCounter=4, IntermediateError=false, Node=vsa7497420, OriginComponentName=CPI_m6301, PreviousComponentName=CPI_m6301, ProcessId=d9147b86cef7d4f7f504e28d86c15cf236273fde, ReceiverId=TEST_Receiver, SenderId='TEST_Sender', TransactionId=4fee65be01084772810d4b2e1c143b17, ID="12345"}  
]
```

Value Mappings

- Used to look up source or target value for an entry in the deployed Value Mapping for a certain context

The screenshot displays the SAP Value Mapping configuration interface. At the top, it is titled "Bi-Directional Mapping" and includes a search bar and "Add" and "Delete all" buttons. Below this, there are two columns representing source and target configurations. The source side has fields for "Agency" (mySourceAgency) and "Identifier" (mySourceSchema). The target side has fields for "Agency" (myTargetAgency), "Identifier" (myTargetSchema), and "State". A double-headed arrow indicates the bi-directional nature of the mapping. Below the configuration fields, there are tabs for "Value Mappings:" and "Default Values:". Under "Value Mappings:", there is a search bar and "Add" and "Delete all" buttons. A list shows the mapping between source and target values: "mySourceAgency, mySourceSchema" maps to "myTargetAgency, myTargetSchema". A specific mapping is highlighted: "mySourceValue" maps to "myTargetValue". To the right of this list, there is a "Usage:" section with the text: "ValueMap (Source agency, Source identifier, Source value, Target agency, Target identifier) = Target value;". Below that is an "Example:" section with two code snippets: "ValueMap (mySourceAgency, mySourceSchema, mySourceValue, myTargetAgency, myTargetSchema) = myTargetValue;" and "ValueMap (myTargetAgency, myTargetSchema, myTargetValue, mySourceAgency, mySourceSchema) = mySourceValue;".

```
import com.sap.it.spi.ITApiHandler;
import com.sap.it.api.mapping.ValueMappingApi;

def Message processData(Message message) {

    def service = ITApiFactory.getApi(ValueMappingApi.class, null);
    if( service != null) {
        def myTargetValue = service.getMappedValue("mySourceAgency", "mySourceSchema", "mySourceValue", "myTargetAgency", "myTargetSchema");
    }
    ...
    return message;
}
```


Number Range

- Receive unique and consecutive numbers from a defined range
- Number can also be shared across flows by providing an identifier, e.g. Correlation ID

Add Number Range

Name: * myNumberRange

Description:

Minimum Value: * 1

Maximum Value: * 999999

Field Length: 0

Rotate

OK Cancel

```
import com.sap.it.api.ITApiFactory;
import com.sap.it.api.nrc.NumberRangeConfigurationService;
import com.sap.it.api.nrc.exception.NumberRangeConfigException;
def Message processData(Message message) {
    def service = ITApiFactory.getApi(NumberRangeConfigurationService.class, null);
    if( service != null) {
        //Get next value from number range configured in web tooling.
        //second argument is for an identifier like correlation ID to get the same value across multiple flows
        def nextValue = service.getNextValuefromNumberRange("myNumberRange", null);
    }
    ...
return message;}

```

customFunctions for Mappings

- Script activities during message mapping
- Instead of Message object use the MappingContext
- Access of headers and properties
- Call Value Mappings
- Perform complex functions that are not possible with standard mapping functions

```
import com.sap.it.api.mapping.MappingContext;

def String myCustomFunction(String inputValue, MappingContext context) {

    ...

    return "someValue";

}
```

Access to SOAP headers

- Whenever setting a payload in the integration flow only the SOAP body gets influenced.
- The CI SOAP adapter removes the SOAP header on receiving incoming messages
- Via script you can get, set and remove the SOAP headers

```
import com.sap.gateway.ip.core.customdev.util.SoapHeader;
import java.util.List;

def Message processData(Message message) {

    //getting Soap Headers
    List incomingHeadersList = message.getSoapHeaders();

    //remove all Soap Headers
    message.clearSoapHeaders();

    // Create a Soap Header and add it
    ...
    SoapHeader myHeader = new SoapHeader(new QName("<namespace>", "<local name>"), <header object>);
    myHeader.setActor("<SOAP actor>");
    myHeader.setMustUnderstand(true);

    List headersList = new ArrayList<SoapHeader>();
    headersList.add(myHeader);
    message.setHeader("org.apache.cxf.headers.Header.list", headersList);

    return message;
}
```

[Detailed blog](#)

Access of exception objects

- Some adapters don't provide error responses in the message body or message processing log
- After an error occurred the exception object is stored as property

```
import com.sap.gateway.ip.core.customdev.util.Message;

def Message processData(Message message) {
    def map = message.getProperties();

    // get an exception java class instance
    def ex = map.get("CamelExceptionCaught");
    if (ex!=null) {

        // an http adapter throws an instance of org.apache.camel.component.ahc.AhcOperationFailedException
        if (ex.getClass().getCanonicalName().equals("org.apache.camel.component.ahc.AhcOperationFailedException")) {

            def responseBody = ex.getResponseBody();
            def statusCode = ex.getStatusCode();
            def statusText = ex.getStatusText();

        }

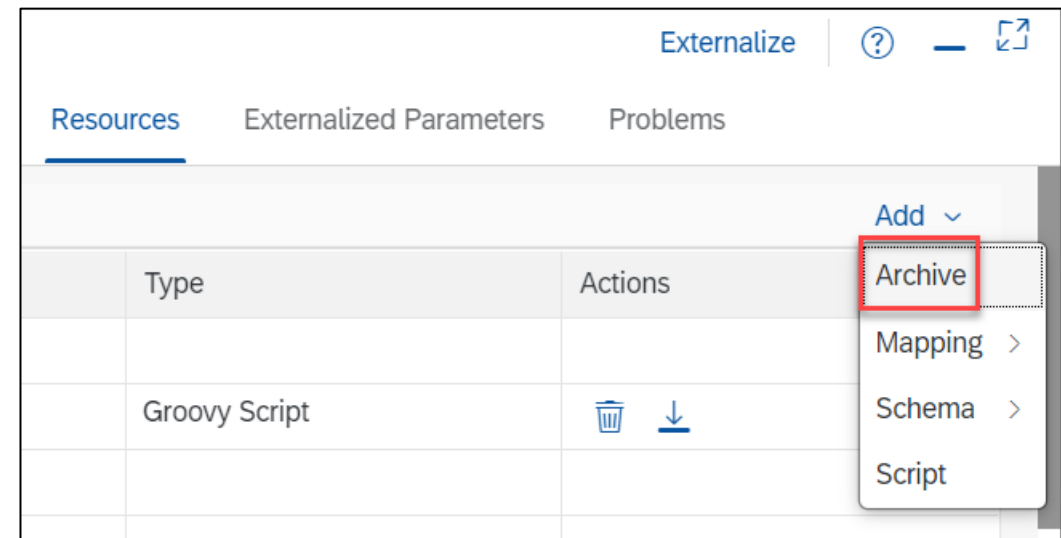
        // an odata adapter throws an instance of com.sap.gateway.core.ip.component.odata.exception.OsciException
        if (ex.getClass().getCanonicalName().equals("com.sap.gateway.core.ip.component.odata.exception.OsciException")) {
            def requestUri = ex.getRequestUri();
        }

    }

    return message;
}
```

Usage of external libraries

- Many problems have already been solved by open source libraries
- How to consume them?
 - Add the jar file under resources
 - Use the import statement in your script to load your class
- Don't use libraries that are already available in Cloud Integration



Buffering of data as Hashmap

- ID mapping for many records can be done via value mapping, but this needs to be created at Design time level
- What if you fetch a list of records and then need to change the identifier on the fly?
- Separate external calls per record is very expensive
- Instead, fetch the lookup data upfront, store everything in a hashmap and access it at runtime

```
import com.sap.gateway.ip.core.customdev.util.Message;
import java.util.HashMap;
import groovy.xml.*;
import java.io.*;

def Message processData(Message message) {

    def body = message.getBody(java.io.Reader);

    HashMap<String, String> myHashMap = new HashMap<String, String>();

    def parsedBody = new XmlSlurper().parse(body);

    parsedBody.Record.each{
        try{
            myHashMap.put( "recordKey", "recordValue");
        }
        catch(Exception ex){
            //decide what to do
        }
    }
    message.setProperty("lookupTable", myHashMap);
    return message;
}
```

```
import com.sap.gateway.ip.core.customdev.util.Message;
import java.util.HashMap;

def Message processData(Message message) {

    def properties = message.getProperties();

    def myHashMap = properties.get("lookupTable");
    String myNewValue = myHashMap.get("myOldValue");

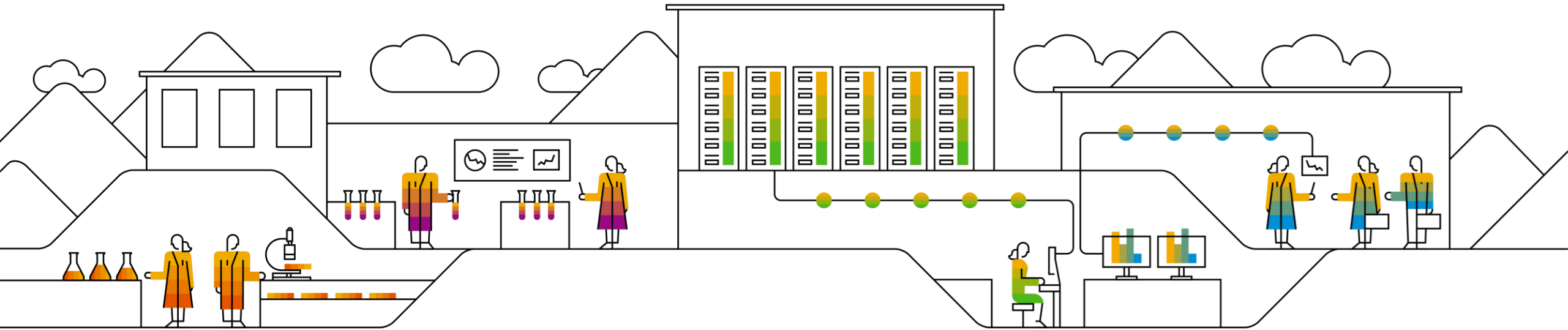
    ...

    return message;
}
```

Other use cases

Everything which is technically possible in Groovy / Javascript can be done, but not everything is recommended.

Considerations when using scripts



Disadvantages of using scripts

- No self-explaining design. In order to understand what's going on the script has to be opened and the code has to be understood
- In case of updates of Java or Groovy the author is responsible to keep the coding working
- When bad coding is used issues like Out-of-memory can occur which will impact the whole tenant
- No support by SAP for custom coding

Recommendations

- Use a graphical flow step wherever possible
- Use a [preview tenant](#) to test the compatibility of your scripts against the latest CI version
- Apply [best practices](#).
- If you want to re-use scripts in your tenant, use script collections
- Avoid Encoding issues. See details [here](#)
- Protect your script against any kind of attacks when parsing payload
- Apply security scans before using 3rd party libraries
- Always define your variables in Groovy scripts. [Blog](#)



Did you know??

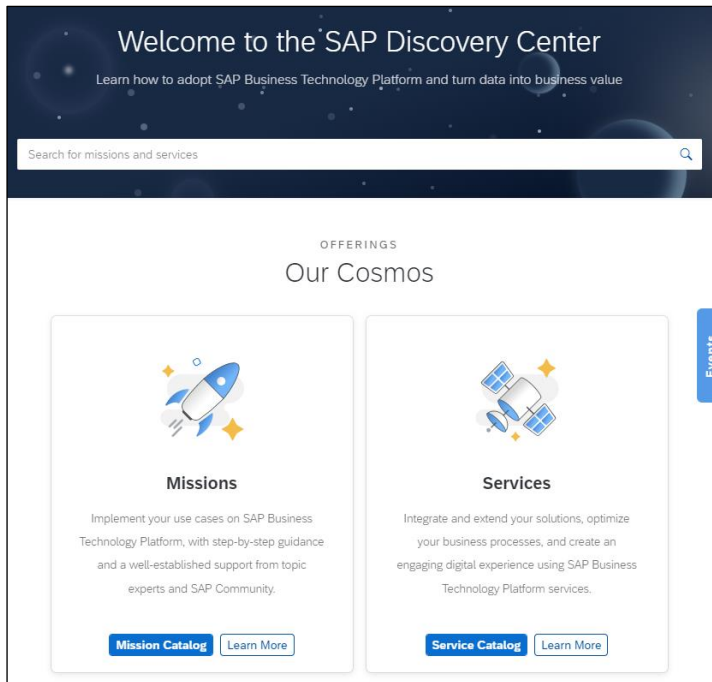
SAP Discovery Center

The Resource for SAP BTP Information & Guidance

Discovery Center
Demo Video



Explore BTP Use Cases & Services



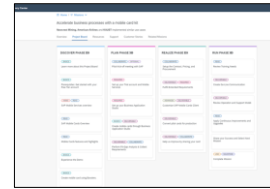
[SAP Discovery Center](#)

Find all SAP BTP related information

- [Use cases](#)
- [Free Tier Services](#)
- [Business Services](#)

Realize the Use Cases as a Mission

- Step-by-step guidance and tutorials
- Structured project board with phases and tasks
- Collaboration with SAP experts and partners



Be Inspired by Customer Stories

Take a look at [other customers' success](#)

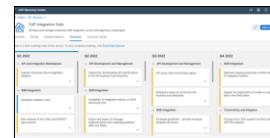
Adapt the Architecture

Adjust your architecture and create your own solution diagram



Gain More Insights Into the Service(s)

- [Roadmaps](#)
- [Service plans](#)
- [Retiring services](#)



Analyze & Estimate Costs

Check [pricing](#) with the integrated [estimator](#)



Did you know??

SAP BTP: Customer Value Network

What is the SAP BTP CVN?

Dear SAP Customers and Partners,

Welcome to the SAP BTP Customer Value Network !

Our Customer Value Network is dedicated to provide you with latest information on SAP's strategy and innovations

It will allow you to make important connections to topic experts, peers, technical content and SAP for their professional and business success to support the sharing of insights and best practices.



The CVN will allow you to experience the passion for excellence that sets this SAP Customer community apart.

Free-Tier

Gain hands-on experience with SAP Business Technology Platform (SAP BTP)

[Watch the overview video](#)

Get started with SAP BTP for free

Try SAP BTP services for free. Start in a production environment right away — try out selected services, up to a specified capacity limit, and easily switch to the paid service plan, without losing your work.

New Users

To access your free services with no upfront commitment:

1. Click the button below to go to SAP Store
2. Add Pay-As-You-Go for SAP BTP to the cart
3. You will be asked to set up an account if you do not have one. Within three hours you will receive access via email to explore and start building with SAP BTP



[New users: Go to SAP Store](#)

<https://developers.sap.com/>

The screenshot shows the SAP Developers website interface. At the top, there's a navigation bar with various links like 'SAP Cloud Platform', 'Logon Error Message', 'ICP', 'Cloud Reporting', 'Jam', 'IPB upgrade Wiki', 'PS', 'CAM - internal', 'CSD', 'api', 'IPB', 'IPB testing', 'stocks', 'Design principals w...', 'Imported', 'UK', 'service systems', and 'Other bookmarks'. Below the navigation, the main content area is divided into several sections:

- Developer Products**: A section with a 'View All' link, containing a list of products: SAP Business Technology Platform, SAP Business Application Studio, SAP Conversational AI, SAPUI5, and Open Source. Below the list, it says 'Stay up to date with what is happening at SAP that matters to developers including new trials, tools, tutorials, and more. Subscribe to the Developer Newsletter'.
- Tutorials**: A section showing progress: '0/1873 Tutorials Completed', '0/220 Missions Completed', and '0/272 Groups Completed'. Below this is a 'Start Learning...' section with a 'Learn and Build with SAP APIs' tutorial card. The card includes a 'Beginner' badge, a '5 min.' duration, and a link to 'SAP HANA, express edition' or 'Explore Tutorials here'.
- I want to...**: A section with a dropdown menu set to 'APIs and Integration'. Below it, the text reads 'Build innovative, cloud-based enterprise solutions' and 'Trending Tutorials // View All'. This section features two 'FEATURED' tutorial cards: 'Learn and Build with SAP APIs' (35 min) and 'Create an API Proxy' (15 min).
- Upcoming Events**: A section for 'Meet the SAP Conversational AI Tutorial Challenge Winners' (Community Call, Online, May 6 2021) with a 'Learn More' button.
- Achievements**: A section showing '0/10 Achievements Completed - % completed indicated below' and a list of progress indicators for 'Completed your first tutorial', 'Completed 25 tutorials', and 'Completed 50 tutorials'.

PI Elevation | Guide and Best practices

[Migration guide](#) for SAP Process Orchestration

Asset interoperability

- Reuse of mappings, data types*, and message types*
- See [blog series](#)

Design guidelines

- See [documentation](#)
- Integration packages at api.sap.com incl. postman collections

Enterprise Integration Patterns

- See [blog series](#) comparing patterns on **SAP Process Orchestration vs. SAP Integration Suite**

* Roadmap item: this is the current state of planning and may be changed by SAP at any time without notice.

The screenshot shows the SAP Help Portal interface. At the top, it says 'SAP Help Portal' with navigation links for 'Browse by Product' and 'SAP Learning Journeys'. The main heading is 'Migration Guide for SAP Process Orchestration'. Below this is a search bar with the text 'This document' and a search icon. A 'Create Custom PDF' button is visible in the top right. On the left, there is a 'Table of Contents' sidebar with a tree view showing the current page's position. The main content area features the title 'Migration Guide for SAP Process Orchestration' with a star icon, followed by a sub-heading 'SAP Integration Suite: A Future-Proof, Hybrid Integration Platform'. Below this is a paragraph of introductory text. A 'SUITE QUALITIES' section follows, featuring a grid of seven icons and their corresponding descriptions: 'Seamless user experience', 'One workflow inbox', 'End-to-end process blueprints', 'Aligned domain models', 'Consistent security & identity mgmt', 'Coordinated lifecycle mgmt', and 'Embedded & cross-product analytics'. At the bottom, there is a 'Key Highlights' section with a bulleted list of key features and awards.

Q & A

Thank you.

Contact information:

Raj Chintam

SAP BTP Adoption & Consumption Center

Cloud Success Services

Raj.Chintam@sap.com

