BPM MIGRATIONS

A Guide to Migrate IBM BPM Applications to BAW on Cloud

A Publication of Salient Process

TABLE OF CONTENTS

INTRODUCTION

Migrating from an on-prem BPM/BAW environment to the IBM Cloud is typically an exercise in configuration, adaptation, and testing:

- Configuration of security, connections, and resource references
- Adaptation of design artifacts (process apps, toolkits, and other solution components) that may not be cloud-compatible as-is
- Testing of the migrated solution – including possible cloud-specific adaptations – in the cloud

A common expectation is for both applications and process instances to be migrated to the cloud, but on the IBM BPM/BAW platform, process instances are not movable entities. For this reason, a migration can also require federation - which gives users the illusion of working in a single environment while both the on-prem and cloud environment operate together relatively seamlessly during the transition period until the on-prem environment is drained of all its instances.

Another expectation may be that very minimal disruption will occur in reporting. This is only true in limited cases. Historical data continuity is only seamlessly possible under certain scenarios. In some cases, a migration can entail bridging between two Performance Data Warehouse instances, or even consolidating both instances post-transition, to help reduce migration impact on the reporting function.

In any case, the “exercise in configuration, adaptation, and testing” summarizes an approach that has worked well in migration projects for Salient’s customers, both in its sequence and in the detail of its implementation – much of which is discussed in this document.

OVERALL MIGRATION SEQUENCE

There is an optimal sequence to a migration that generally progresses from 1) infrastructure setup on the cloud, to 2) application migration, to 3) migrated solution testing, to 4) production.

The depiction of this sequence/process is provided below and (in the context of the information in this document) can be used to create a migration plan and to estimate the extent and the nature of the effort involved:

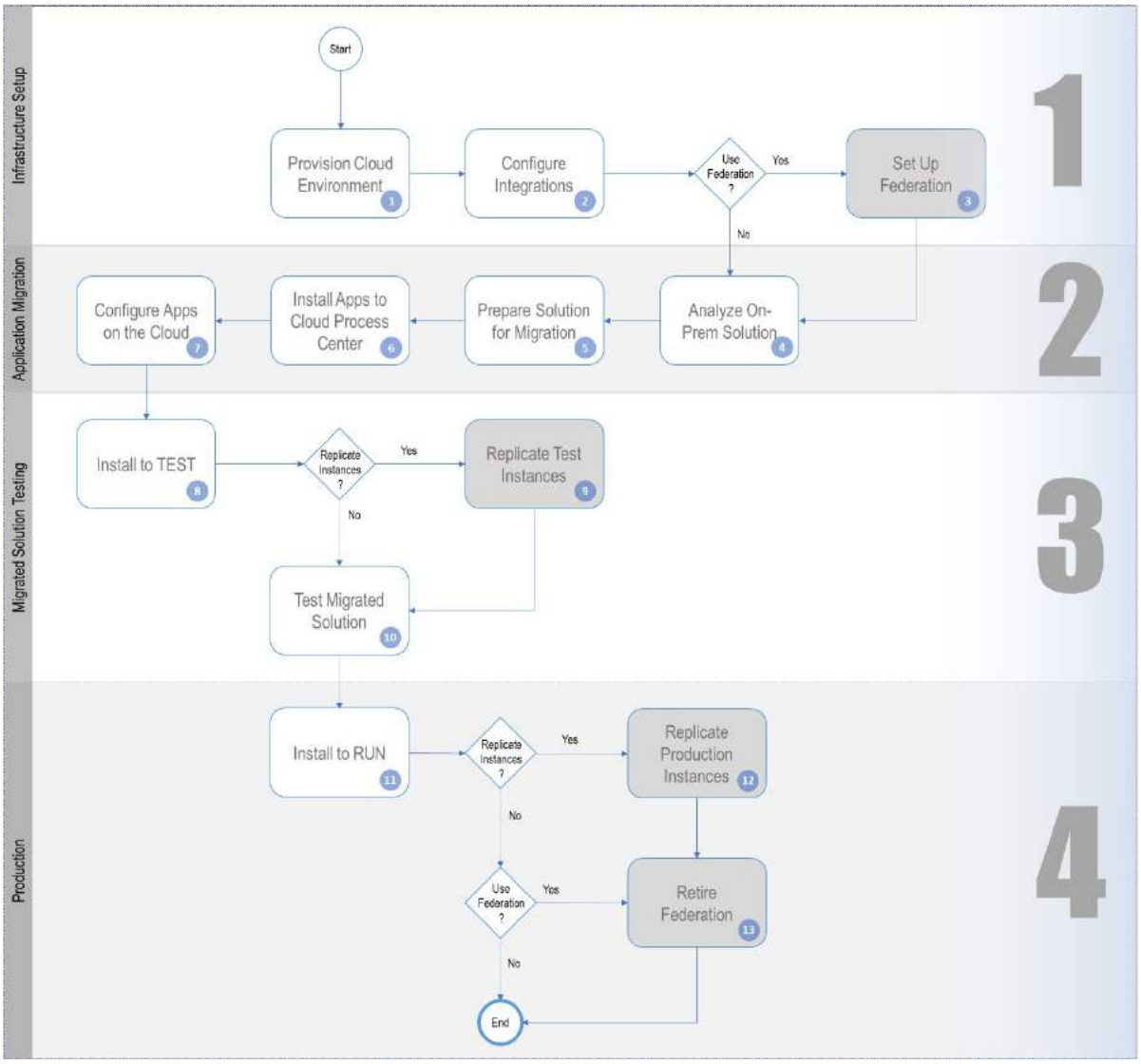

Figure 1 General sequence for BPM/BAW migration to the cloud

Note on optional activities: The gray-shaded activities are optional and depend on a customer's need for federation and the potential replication of some instances to the cloud (more on both topics later).

This document focuses on expertise, practices, and accelerators that support the overall migration sequence.

CONFIGURING THE CLOUD ENVIRONMENT

There are several aspects of cloud configuration that must be implemented before processes, tasks, and services can run and/or be used.

Security

On the cloud, users can log in with a cloud-specific user-id & password, or they can seamlessly log in using SSO if their current user security environment provides this capability.

SSO

SSO in BAWoC is SAML-based and requires the customer's system administrator to share details of the Identity Provider with IBM Cloud Support. The following procedure could be followed when using Microsoft Azure AD as Identity Provider [1]:

1. Create a New application:

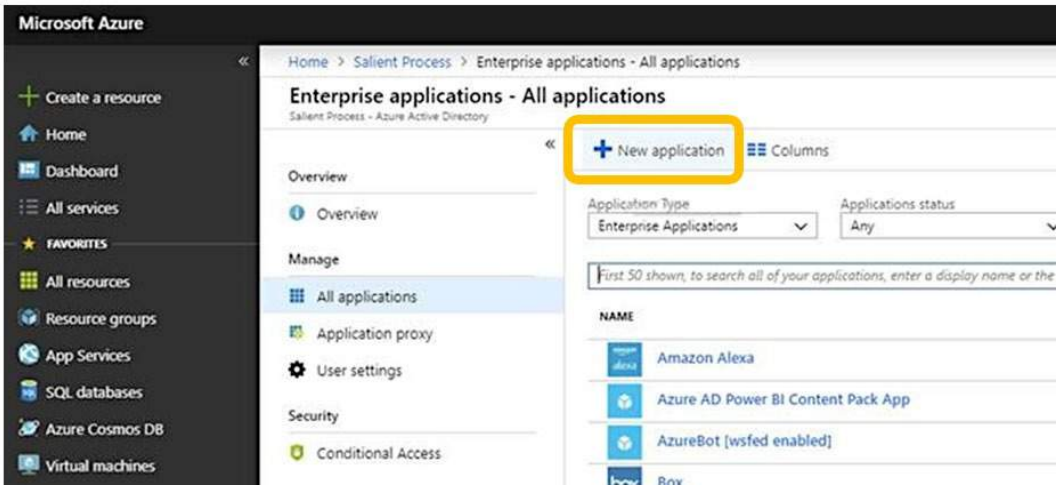

Figure 2 Setting up SSO for the cloud: Microsoft Azure - Management Console

[1] See Configuring Azure Active Directory as an Identity Provider in the IBM Knowledge Center

2. Get the IDP descriptor from Azure AD:

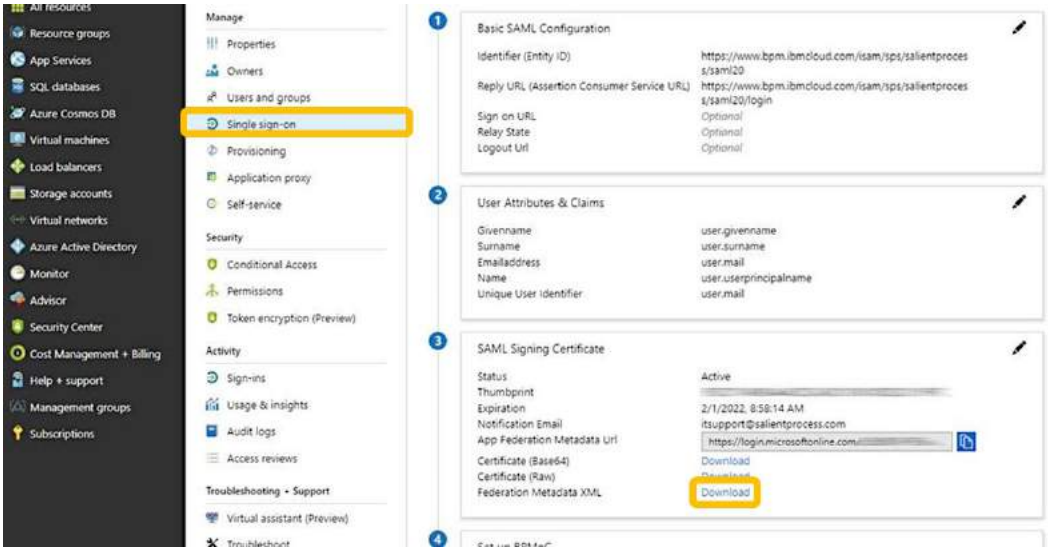

Figure 3 Setting up SSO for the cloud: Microsoft Azure - Configuring Single Sign-On

3. Send the downloaded descriptor (XML) file to IBM support

User setup

Even after SSO is set up, users still need to be defined in the IBM cloud user repository (as of BAW 19.0.0.2 on the IBM Cloud, there is no option to simply point BAWoC to a customer's LDAP).

Users can be manually invited by a customer user designated as an administrator:

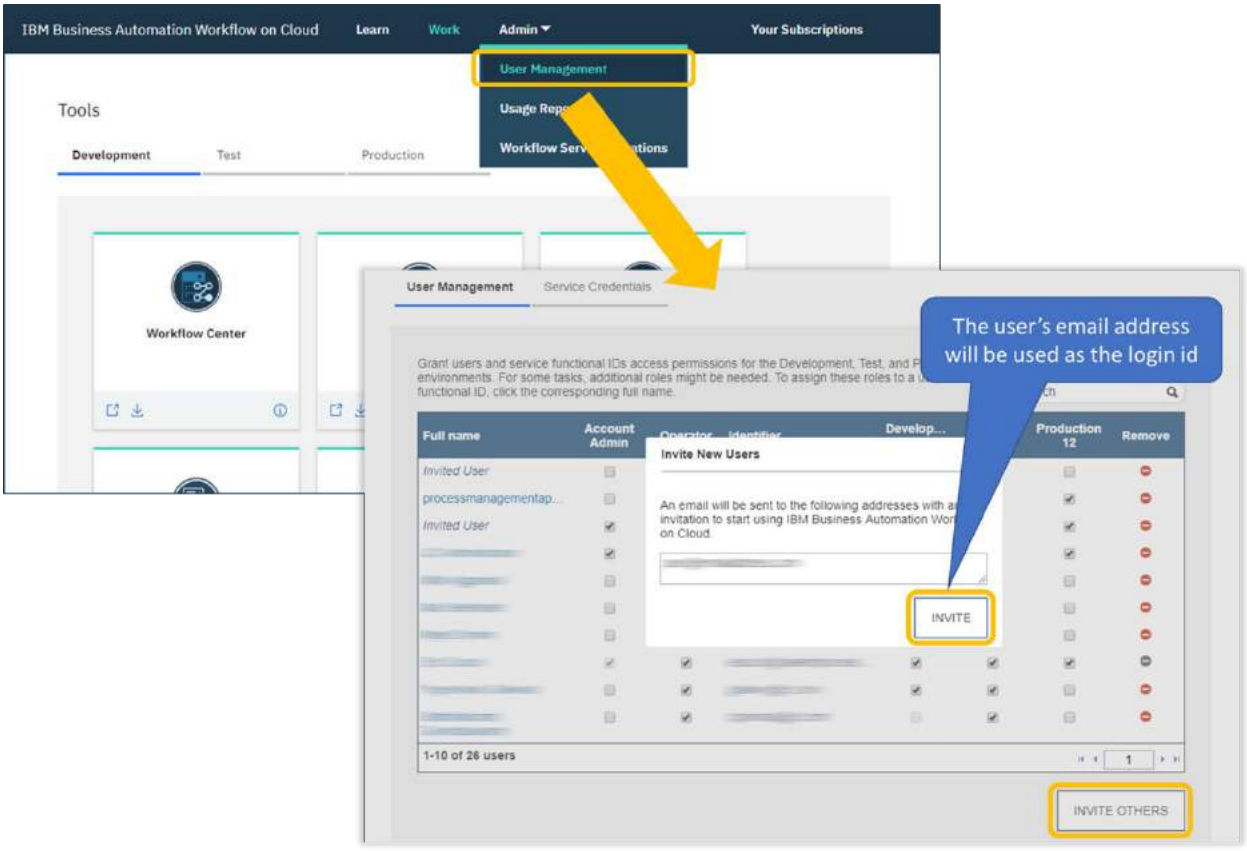

Figure 4 Inviting new BAWoC users

Once invited, users can activate their access from the IBM Cloud link provided in the registration email:

Figure 5 BAWoC registration email example

Alternatively, IBM provides a REST API that can be used to define users and groups programmatically. The description of the various API capabilities & interfaces (including for user and group management) can be accessed from https://<customer>/bpm.ibmcloud.com/api/explorer - as shown below:

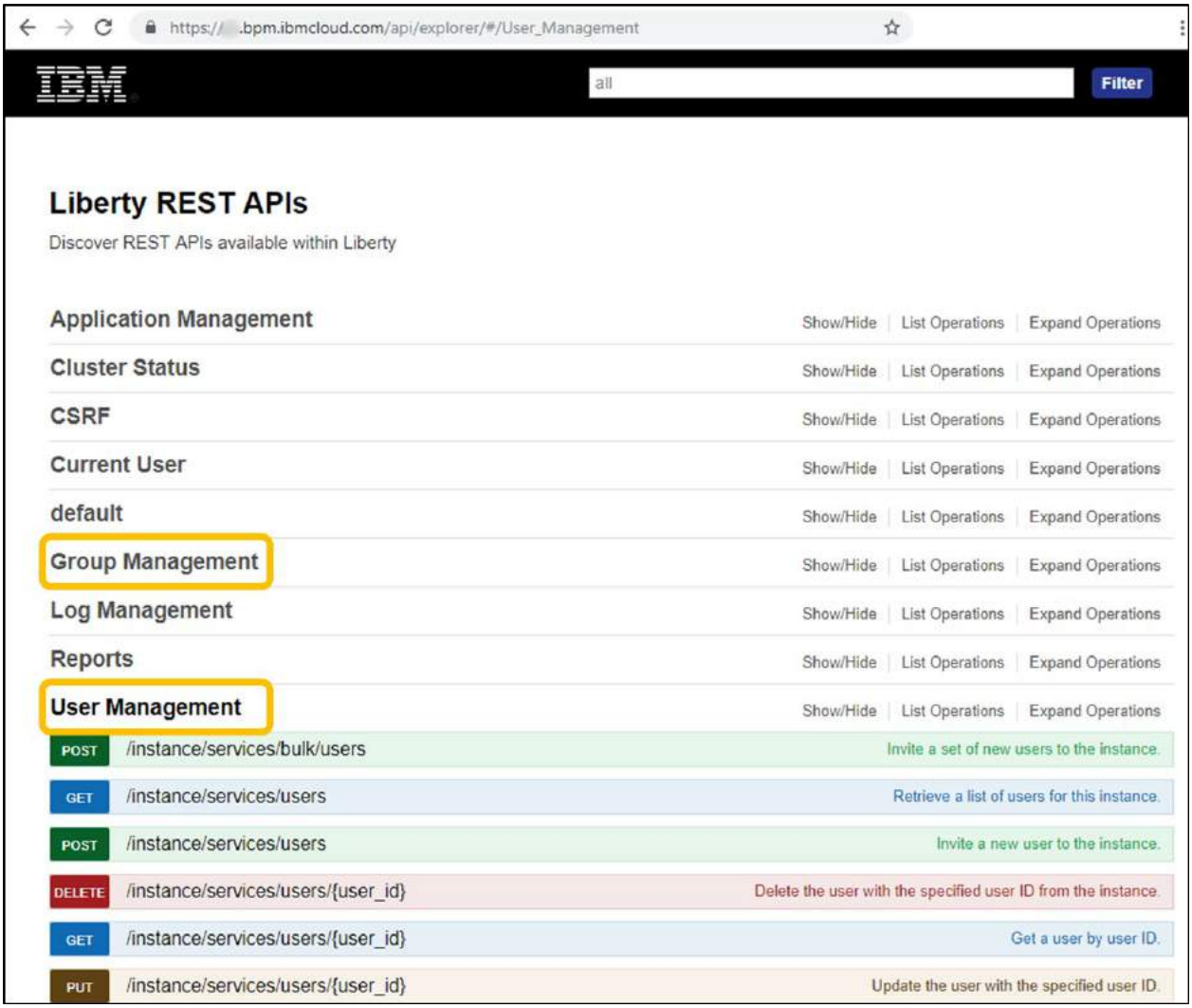

Figure 6 Application Management REST API Explorer [2]

Note on API and user registration process: Provisioning a user through the REST API provides the same email-based registration experience for the end-user. If an organization broadly uses IBM BPM/Workflow and defines users in mass through the REST API, it might make sense to precede defining those users with an email notifying them of the upcoming need for cloud registration, to reduce potential confusion.

[2] See Cloud operations APIs in the IBM Knowledge Center

Note on on-prem to cloud user-id consistency: To minimize potential impact to existing services and systems of records that might be using on-prem user-ids as keys or for reporting purposes, the set of users defined on the cloud should reflect, and have the same login names as, the set of users for BPM on-prem.

Examples of REST API usage

One important aspect of IBM's REST API is the requirement of a Cross-Site Request Forgery (CSRF) -related token. The simple examples below show how one might use the management REST APIs to work with users.

Obtaining a CSRF token:

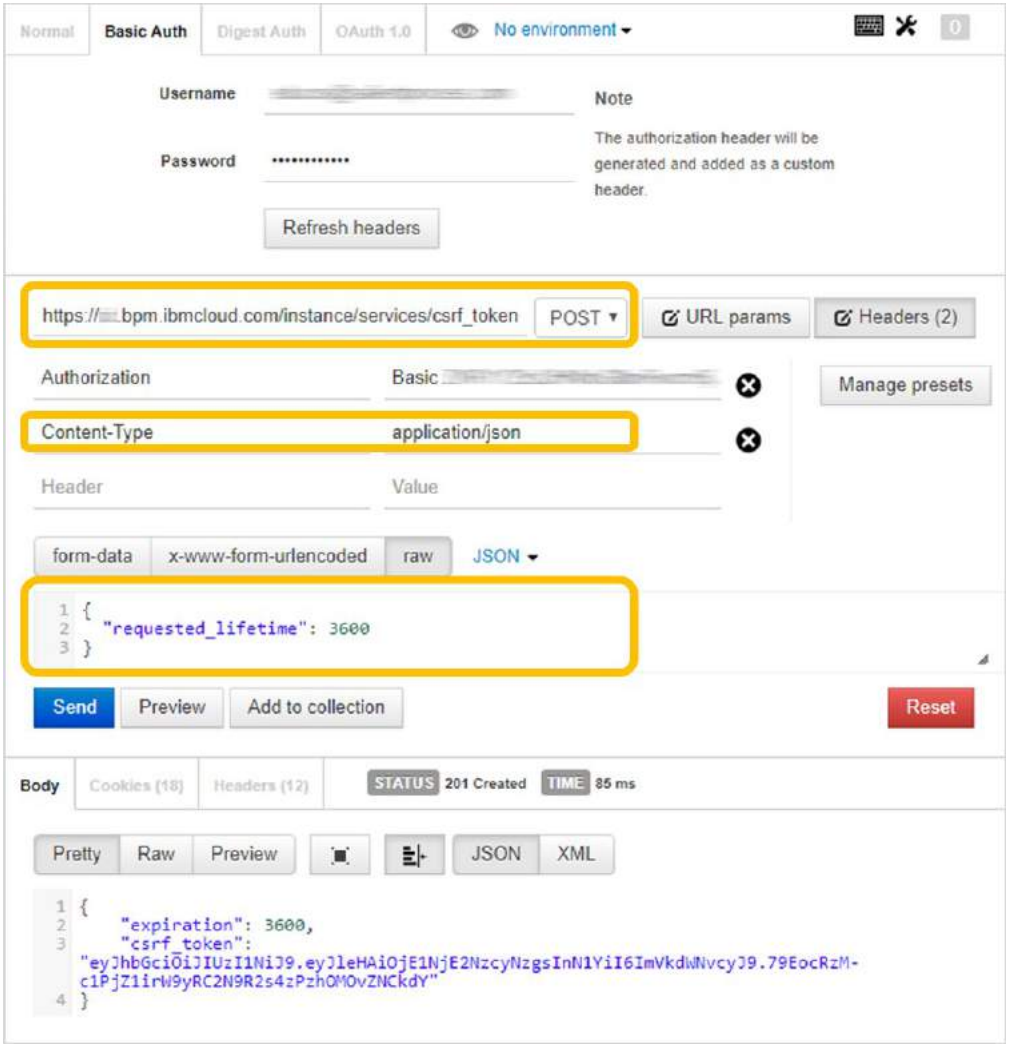

Figure 7 Postman example to get management API CSRF token

Creating a user:

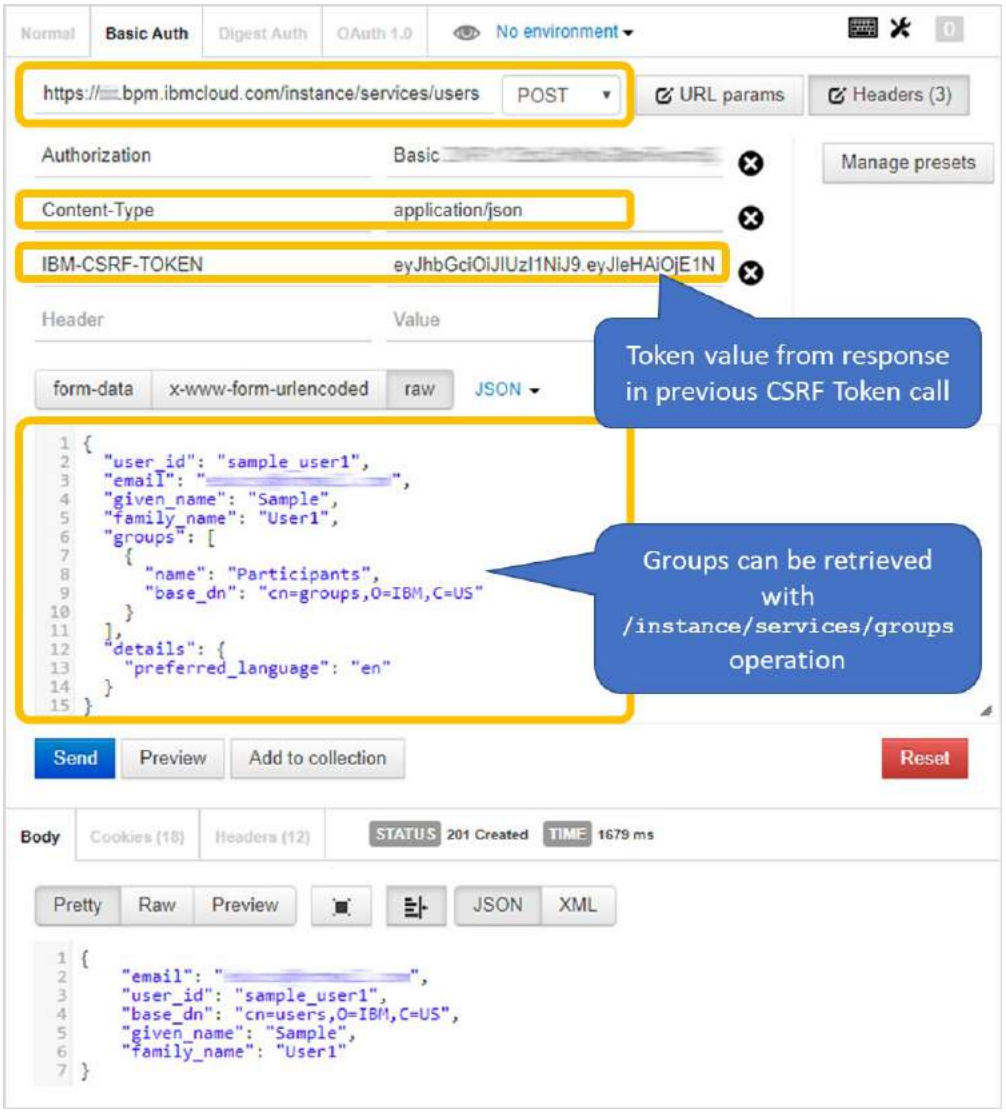

Figure 8 Postman example to define a user under the BAW instance

Service user ids

User ids defined under the BAW Cloud instance are counted against a maximum allocation that is managed by the customer's license/subscription. Sometimes however, user ids for BAW are simply used to access the system programmatically (likely for integration purposes) and have no other productive purpose.

A special type of user can be defined, under BAW on Cloud, that doesn't count against the licensed allocation but can be used to access BAW and make REST API calls (for example to start a process, or to make a query).

Such service user IDs can be defined through one of the administrative sections of the cloud instance (Admin > User Management, then Service Credentials tab) – as shown below:

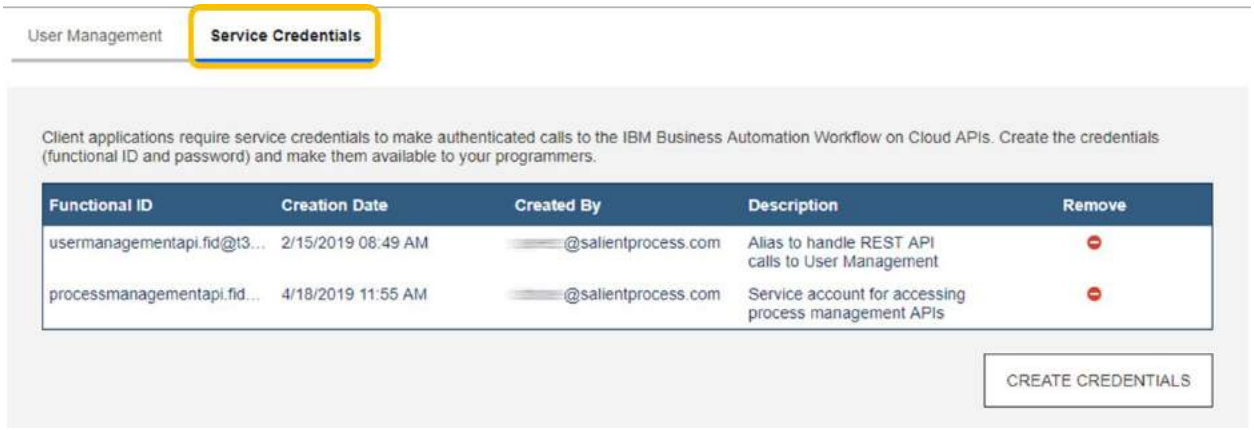

Figure 9 Integration "service" user ids

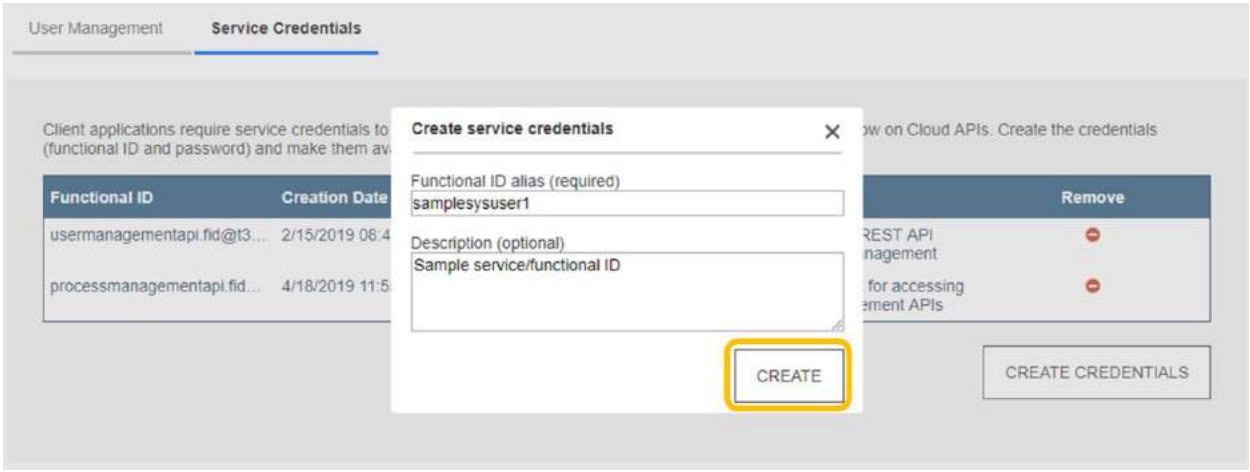

Figure 10 Defining a service user-id

REST calls and SSO

Sometimes, customers might make calls to BPM from client-side applications to (for example) invoke the REST API. Those client-side applications may share the same security context and session management method as the WebSphere cell running BPM, or they may use SSO to prevent the need for re-authentication on the client browser making the call.

When using BAW on the cloud however, two restrictions prevent this type of interaction and may require adaptations:

1. BAW on Cloud does not use LTPA token-based session management
2. BAW REST APIs ignore SSO and instead require explicit Basic Authentication

The above constraints imply the caller's identity cannot be propagated to the BAW server on the cloud.

To work around this limitation, customers might consider the following options:

- Server-side "wrapper" logic for the BAW REST API call, which requires adaptation to client-side logic
- A Web proxy to transparently mediate the interaction by relaying calls generically to the cloud, which may only require a configuration change instead of client-side logic adaptations

In either case, the relay logic should use pre-emptive Basic Authentication with pre-configured user credentials (potentially a Service user id) to successfully invoke the BAW REST API.

Network setup

Moving Process Applications and Toolkits to the cloud often requires connectivity to and from those applications to be adjusted. Properly configuring the network and the endpoints it connects is critical in a cloud scenario, especially since the distance between the BAW servers running Process Applications and the endpoints they used to access on-prem will have likely increased as a result of the migration.

Outbound (cloud to customer-internal) connectivity

IBM provides an outbound VPN option for services and processes that need to connect to services/endpoints hosted on customer premises. The standard IPSec VPN provided by IBM Cloud Support puts no restrictions on the type of connectivity supported, but the customer is responsible for the configuration of firewall rules for traffic from the cloud. Possible service/endpoints might include:

1. LDAP servers
2. Web Servers and/or Application Servers
3. Databases (other than the BAW-DBs, which are hosted on the cloud)
4. Email servers
5. Content Stores (such as FileNet P8, Microsoft SharePoint, etc.

Note on environments: Although there are three environments by default on a BAW on Cloud instance, only one VPN needs to be set up. Because applications in each BAW environment (DEV, TEST, RUN) likely use different endpoints, firewall rules generally need to allow connectivity for all those endpoints.

Note on VPN types: Although IBM Cloud support usually suggests a specific default IPSec VPN, a customer may request a different type of VPN, as long as it is compatible with the IBM Cloud.

Security Certificates

For SSL traffic (for example with HTTPS or TLS) security certificates must first be installed on the cloud. Adding a certificate is another administrative function accessible through the Admin > Workflow Server Operations menu:

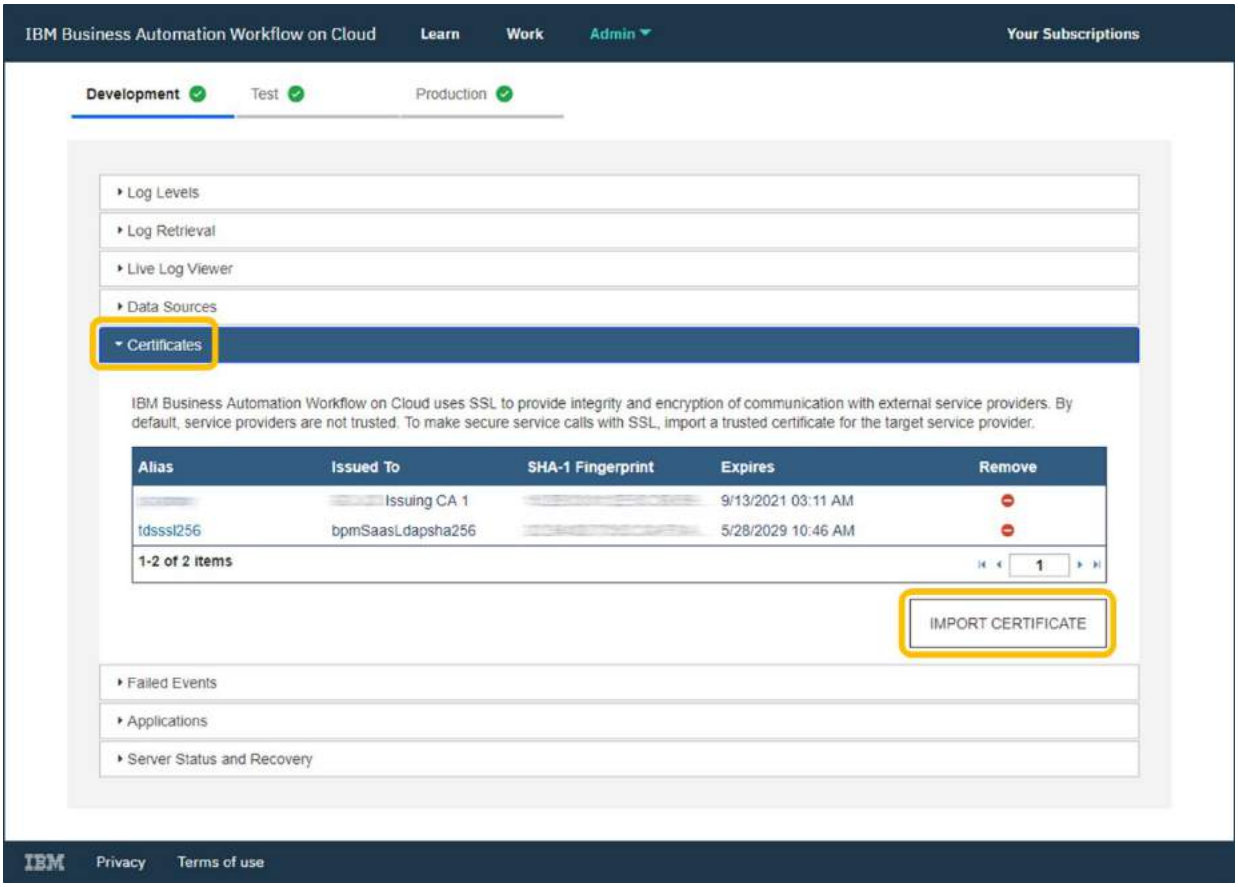

Figure 11 SSL certificates for various integration endpoints

Adding a new SSL certificate is only a matter of providing a URL for the target Web resource and importing the certificate:

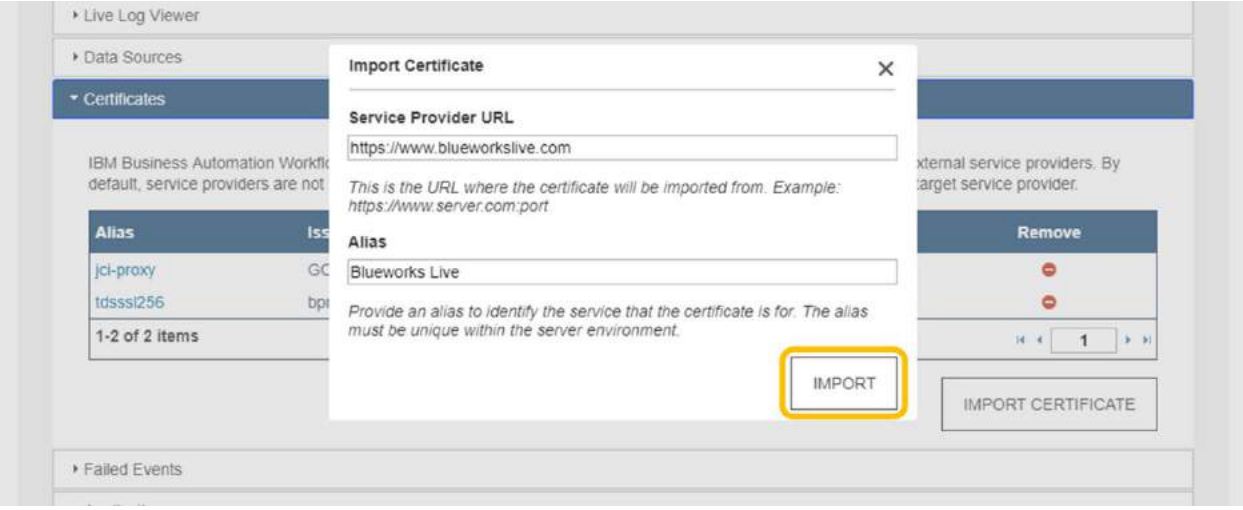

Figure 12 Importing an SSL certificate from an integration endpoint URL

Data Sources

Data sources must be defined to access databases hosted outside the BAW instance (only DB2, Oracle, and SQL Server databases are supported as of BAW 19.0.0.2). Data source definitions are accessed (and configured) from the Admin > Workflow Server Operations administrative menu:

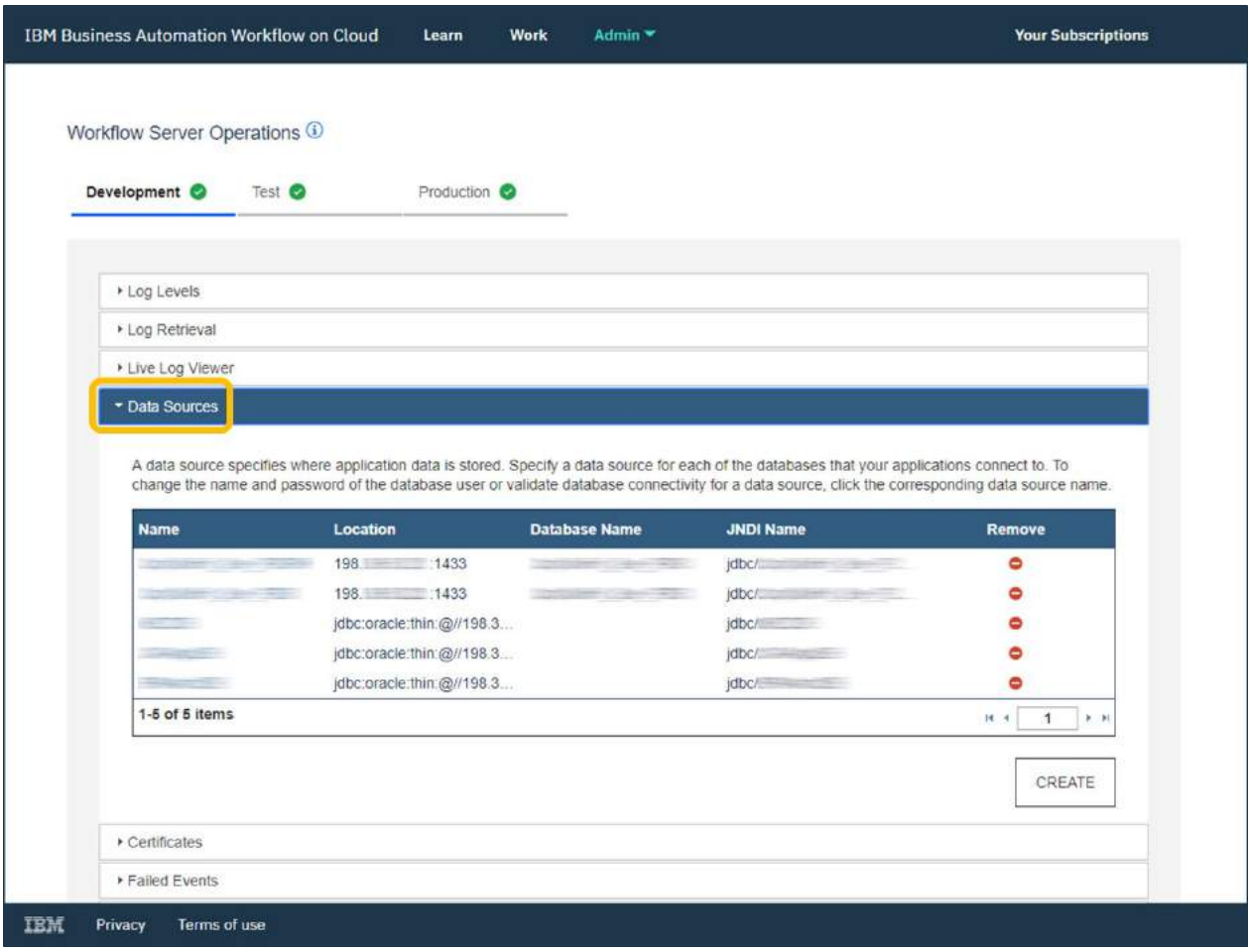

Figure 13 Data source definitions

A new data source can be created as follows:

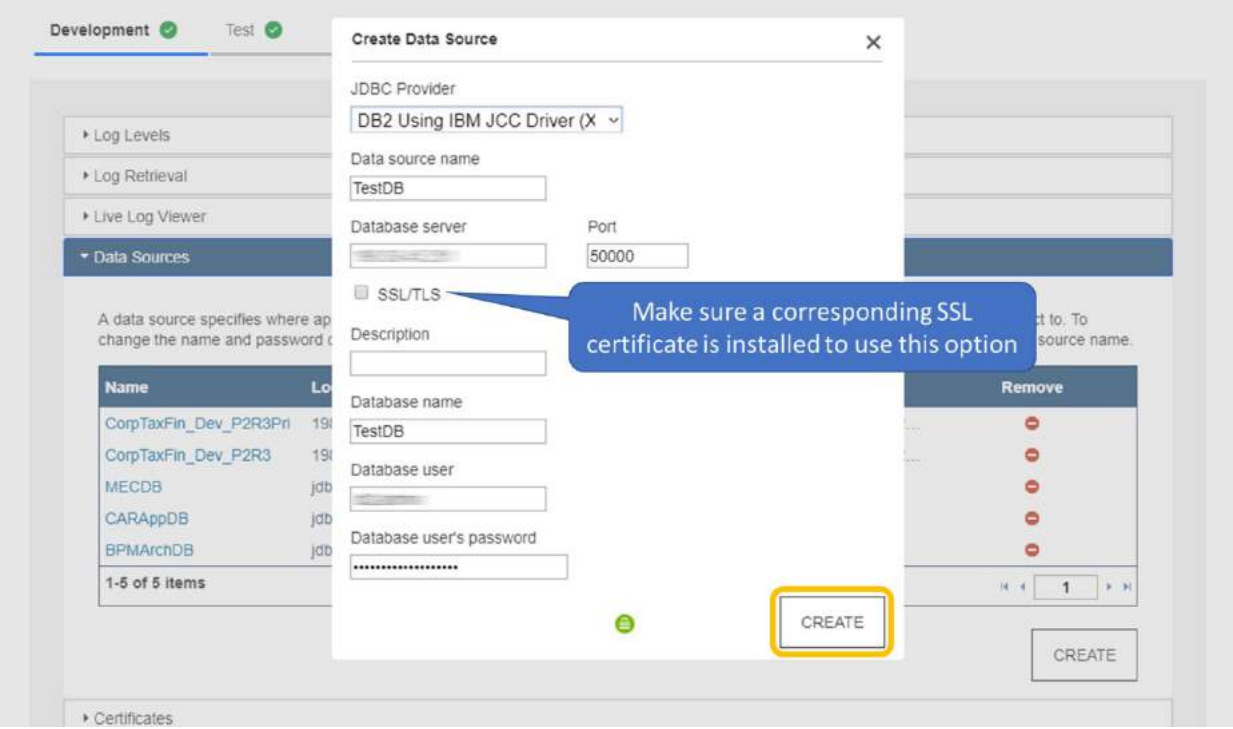

Figure 14 Creating a new data source

DNS mapping for on-prem resources accessible from cloud

Host names for endpoints can be used in many contexts, including in:

- Data sources
- Process Applications and Toolkits for resources such as:
    - Server definitions
    - Environment Variables
    - Exposed Process Values

If no public DNS entries exist for the host names associated with customer-internal endpoints (which is likely the case), DNS entries (to resolve the NAT IP address for a particular host name) can be added on the IBM Cloud through a ticket to IBM Cloud Support.

Adding a DNS entry for IBM cloud support simply means adding a NAT IP address-to-host name mapping to the /etc/hosts file of the BAW server.

Note on IP addresses: In general, any IP address used with the VPN should be the VPN NAT IP address for the endpoint, and not its private IP address (which could create unintended IP address collisions). In other words, if a data source points to a database server whose private IP address is 10.23.44.5 and whose NAT address is 198.23.43.221, the 198.* IP address should be used.

Connectivity diagnostics

Sometimes, a set of process apps running on the cloud may connect to dozens of endpoints. But activating each specific part of the application to exercise all the endpoints can be very cumbersome and require complex test scenarios.

Automated/streamlined endpoint connectivity testing can save significant time in troubleshooting service integrations by providing an instant summary of connectivity status from the cloud:

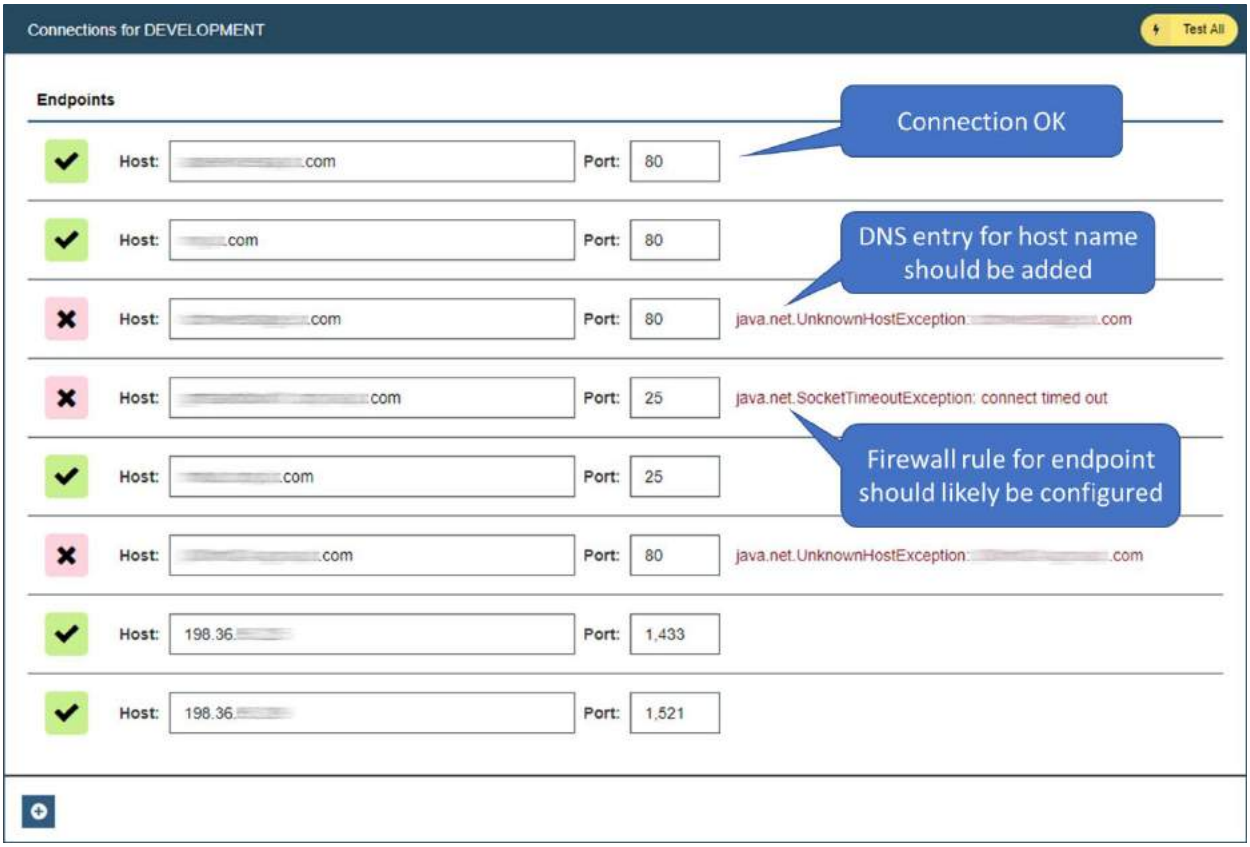

Figure 15 "Test Service Connectivity" Salient Process utility service for cloud migration support

Inbound (customer-internal to cloud) connectivity

Unlike for outbound connectivity, where IP traffic can be mediated through VPN, inbound connections (for an external system to call *into* the IBM cloud) have no such option. Instead, all calls to the BAW instance on the IBM Cloud must occur through HTTPS.

Any logic that might have previously connected to a BPM server in any other way (e.g. using RMI/IIOP or JMS) will need to be adapted (for example by using REST).

File Storage

Some applications make use of the file system (e.g. the BPM Server's file system, or a shared volume) to read from or write to files. If these applications cannot be modified to remove reliance on a file system, a cloud storage option can be requested from IBM Cloud Support. IBM usually allocates 10GB of cloud storage per environment, available to server-run logic (scripts, Java integrations, etc.) as /tenant.

No user interface or file transfer option is available to manage the /tenant directory. However, Salient provides a simple cloud storage File Management application (installable as a WebSphere Enterprise Application) to manage the allocated storage. The application allows:

- Creating, renaming and deleting directories
- Uploading, downloading, renaming, and deleting files
- Unzipping archives to the directory containing them (preserving the archive's directory structure)

An annotated example of the Cloud storage File Manager application is shown below:

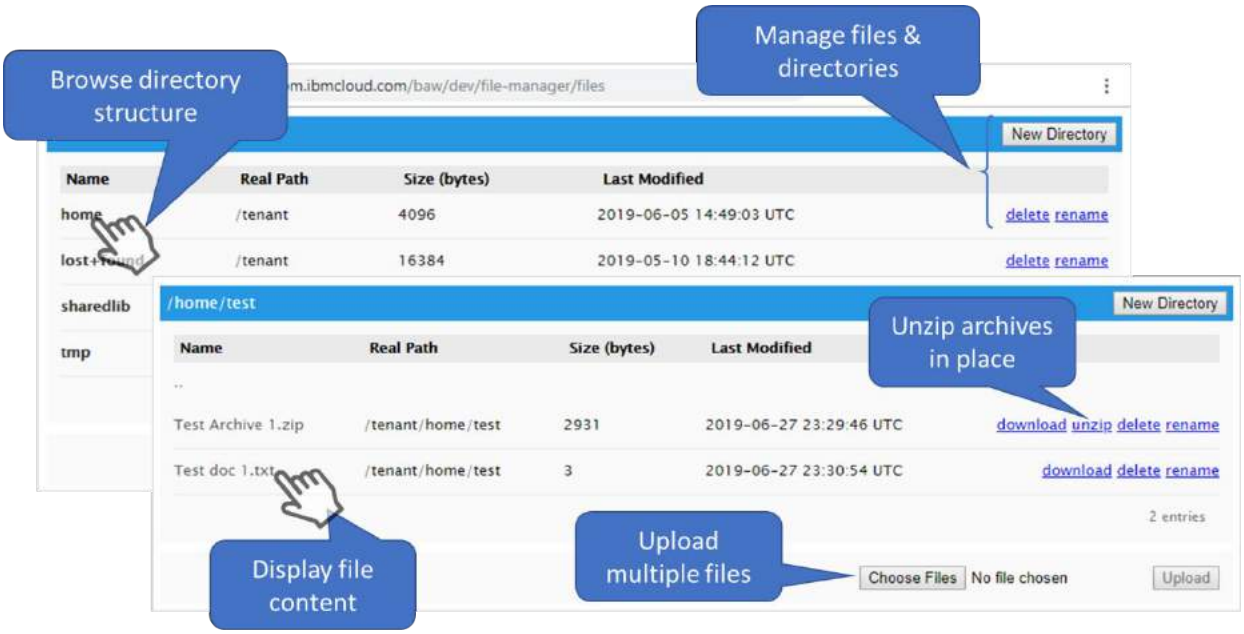

Figure 16 "Cloud storage File Manager" Salient Process utility for cloud migration support

Note on file management in Production: Configuration settings can be used to deactivate key functions in a production environment to prevent accidental deletions and modifications of files and directories.

Note on using files in a server cluster: Servers in a cluster are physically distinct entities with their own local file system. In the Production (RUN) environment, where more than one server exists in the cluster, one server may write a file to its own file system while another in the cluster attempts (but fails) to read that same file because it is not locally available. This problem is easily resolved by requesting a shared volume from IBM Cloud Support (still mapped to "/tenant") instead of the locally allocated one.

Without a Cloud storage File Manager application (like the one provided by Salient), files and directory structures can be uploaded/created either by request to IBM Cloud Support or by importing a process app that creates/unpacks files and/or creates directories through server-side logic.

Adjusting for non-globally unique identifiers

Various ids in IBM BPM/BAW are not universally unique. Such (often-used) ids include Process Instance and Task ids.

Both ids are simple integers backed by corresponding BPM database-managed sequences that make them locally unique (i.e. unique in the scope of the BPM and PDW database).

Because the cloud database for BAW also uses locally unique identifiers, it is virtually certain that some process instance and task ids will be the same as ids that previously existed (or still exist) in the on-prem environment.

If those non-globally unique ids are used as keys/correlation data in business data stores and other systems used with the solution, collisions and false positives may occur.

This problem can be avoided by:

1. Finding the latest id values used for process instances and tasks*
2. Adding a buffer for each id that accounts for new ids created on-prem during the transition and provides ample room to avoid collisions/overlap of ids between the on-prem and cloud environments
3. Deriving new minimum values for each id
4. Requesting IBM Cloud Support to set corresponding id sequences to the newly computed values
5. Repeating the process for each environment (Development, Test, Production)

Deriving suitable identifiers for cloud environment

Current ids for tasks and process instances can be derived by running the Salient Process "Migration Tooling" application ("Determine Process & Task Ids" process) in each on-prem environment (corresponding to DEV, TEST, and RUN on the cloud).

Running the "Estimate Target Ids" task provide the following configurable information:

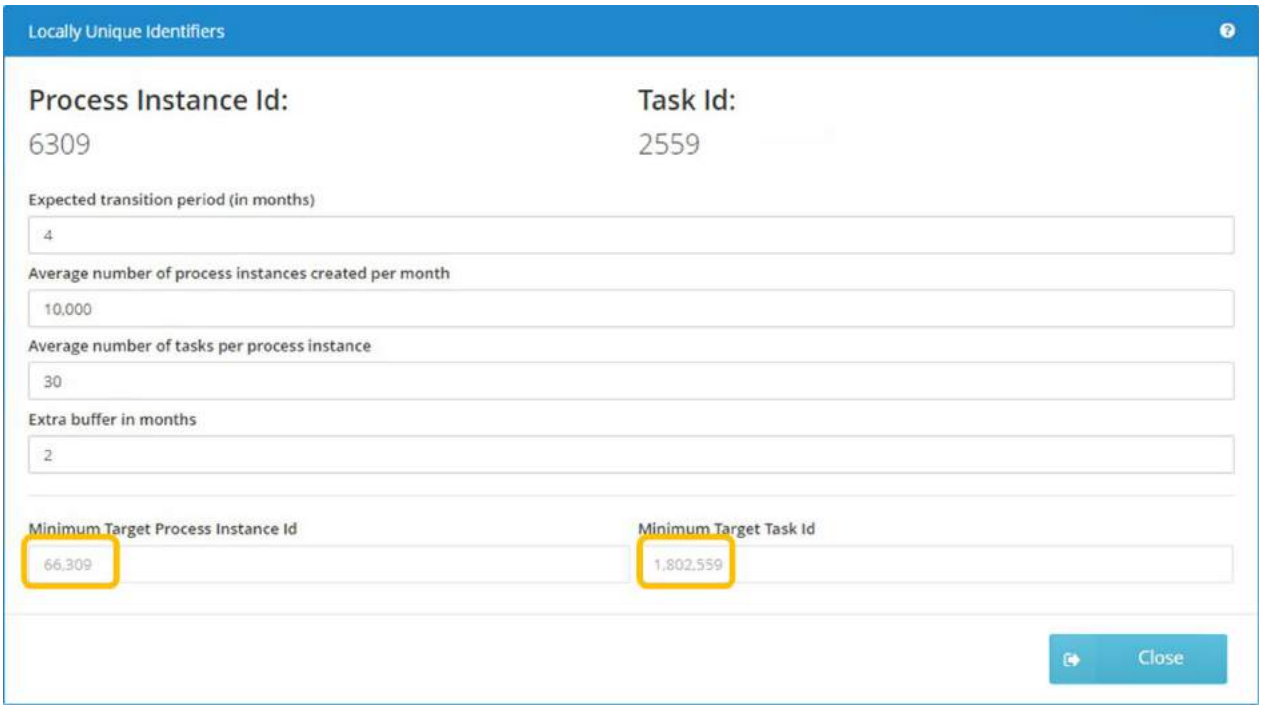

Figure 17 Salient Process Id value estimator for migrations

Note on estimating id values: The default values provided in the "Estimate Target Ids" task should be adjusted to reflect customer expectations and historical information regarding tasks and process volumes. When changes are made, the resulting id values for process instance and tasks are re-computed automatically.

Alternatively, process and task id values could be computed/derived as follows:

1. Create a new process app containing one BPD/process itself containing a task assigned to the tester
2. Create an instance of the new process and user Process Inspector to take note of the process instance and task ids using Process Inspector
   If (for example):
   - Observed current process instance id = 256,327
   - # ids expected to be created during 4-month transition = 8,000
   - Safety buffer = 100,000

Then the minimum computed value could be ~ 400,000

PROCESS INSTANCE MIGRATION

Process instances are distinct "live" entities that are separate from the source/modeling artifacts that are used to build them. Because process instances live inside the BPM engine and can last a long time (over a year in extreme cases), an approach is needed to deal with those instances during the migration.

IBM currently provides no option to physically move process instances from one BPM database to another. Transferring BPM database records from the on-prem database to the cloud database is both an unsupported scenario and is a very complex endeavor due to identifiers and relationships (some encoded in opaque Java serialized binary objects) that cannot merely be moved from one environment to another without numerous adaptations.

Potential options

Of the many aspects to consider for process instance migration, the following are often the most impactful:

- Disruption for end-users
- Licensing cost for on-prem and cloud environments
- Disruption to reporting continuity

The above considerations can also be influenced by:

- The duration of the on-prem sunset period (set by the longest living process instance running on-prem)
- The current solution's tendency to feed itself new instances (i.e. process instances that create other instances during their lifecycle – thus preventing a drain/sunset scenario)
- IBM's ability to agree on licensing accommodations during the transition period (or part of it)
- The degree of tolerance from the end-user community in working with two side-by-side BPM front ends (e.g. BPM portal) during the transition
- The effort and cost associated with the seamlessness of the transition

Customers could consider the options on the next page (in the context of associated pros and cons) in choosing an acceptable approach to process instance migration. Note some options are not mutually exclusive.

Side-by-side transition

The side-by-side transition scenario relies on end-users working on two BPM Portals to work on tasks, processes, and to access UI services and dashboards – as depicted below:

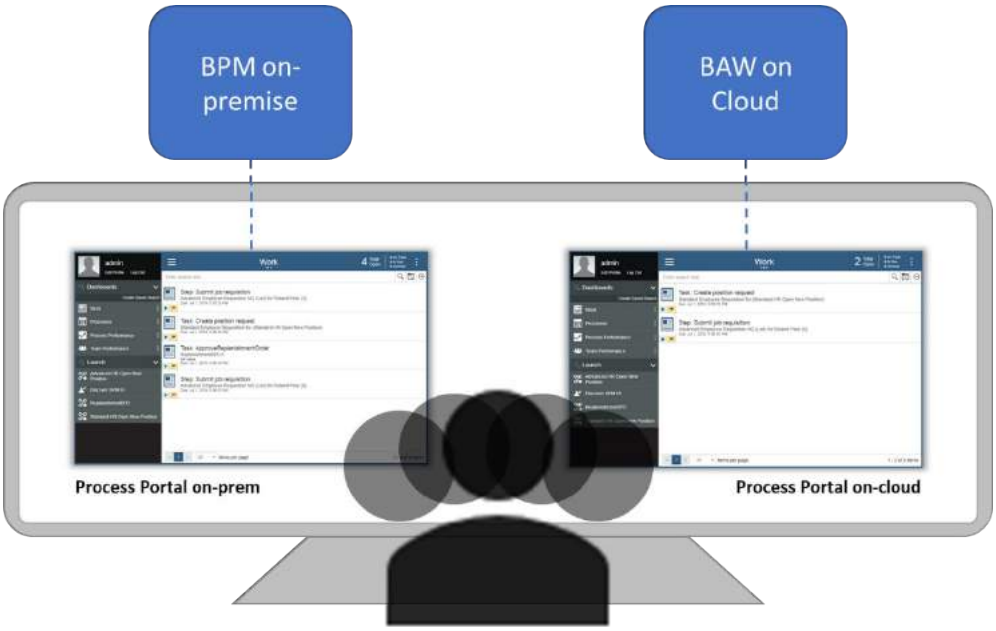

Figure 18 Using side-by-side portals during environment transition

Pros	Cons
• Technically the simplest of all as it pushes the overhead of dealing with two separate systems to the end-user • Lowest technical cost and effort • Simple post-transition finalization (users just stop using the old on-prem Portal)	• Potentially disruptive/distracting for end-users, especially if they must work in this mode for a long time • Requires multiple logins if not using SSO

Federation with cloud

While most process instances cannot be moved from the on-prem environment to the cloud, a federated portal can make two BPM/BAW systems look like one to the end-user.

The federated portal approach currently relies on 3rd party offerings such as Salient Process Federated Portal to allow IBM BPM Portal (or potentially other custom portals) to display task lists, work on tasks, processes, UI services, and dashboards seamlessly aggregated from two systems (one on-prem, one on-cloud).

Note on IBM Process Federation Server: IBM provides a product called "Process Federation Server," which is designed to federate on-prem environments. The product, however, cannot be used currently to aggregate content from BAW on Cloud due to significant security/session management differences between on-prem BPM/BAW installations and BAW On Cloud.

Salient's Federated Portal offering relies on proxies and on the definition of virtual hosts that cannot be done in a Cloud environment. Accordingly, Salient's Federated Portal must be installed on-prem. After the transition, end-users simply point their browser to the cloud-hosted BPM Portal to work exclusively with tasks, instances, services, and dashboards on the cloud and the entire on-prem environment (including Federated Portal) can be retired.

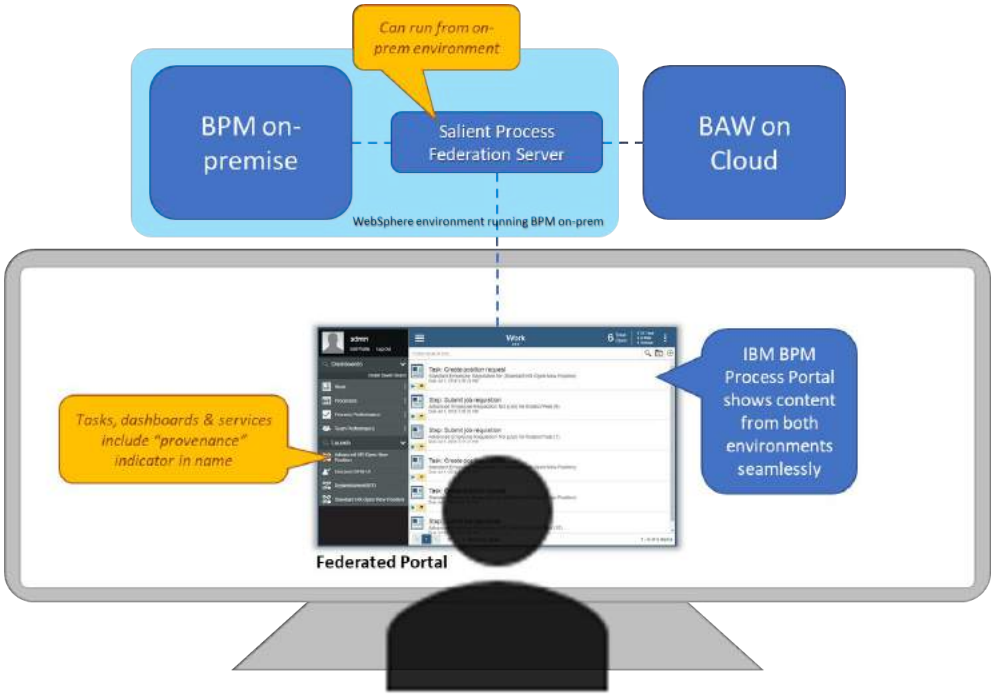

Figure 19 Using a federated portal during environment transition

Pros	Cons
• Comparatively less distracting/disruptive for end-users than side-by-side alternative • Simple post-transition finalization (users just point to the cloud Portal)	• Setup somewhat involved and tooling license required • SSO cannot be enabled on the cloud for the duration of the transition

Deciding between the side-by-side and federated portal approach

Customers need to evaluate the impact of split focus on end-users, in the context of the expected duration of the transition/sunset. The relationship between the value of federation to the transition duration represented below in generally true:

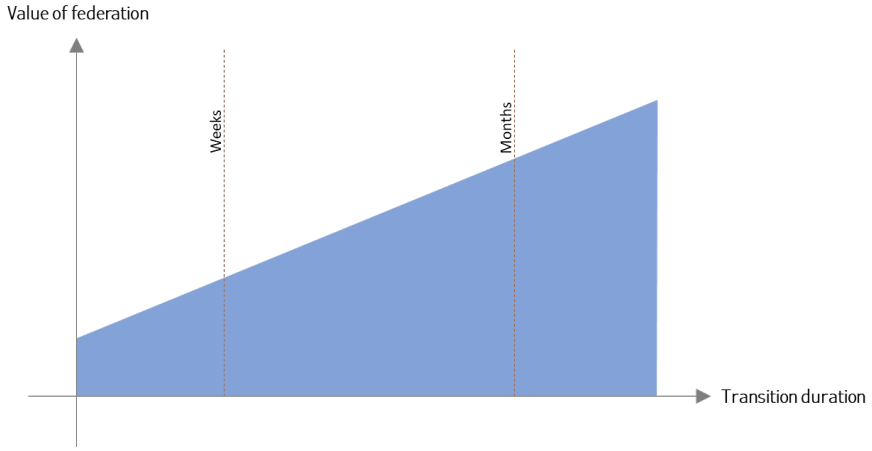

Figure 20 Chart: Value of federation to transition duration

Ultimately, customers must weigh the cost of federation setup and potential licensing against the cost of disruption – especially for long transition periods. The immediate risk of side-by-side vs. federation is usually not significant unless immediate action on high-priority items is critical (and the end-user is not looking at the portal that lists the highest-priority task at the time of choosing what to work on next).

Assuming process instances do not generate other equally long-running instances in the same system, the duration of the transition simply depends on the longest-living process instance. However, if only a small number of straggling instances make the transition untenably long, additional approaches may be considered to effectively "move" those very-long lived instances to the cloud. This option is discussed below.

Process instance transfer to the cloud

As mentioned previously, physically moving a process instance to the cloud is simply not a capability of IBM's BPM/Workflow platform. But the options discussed here might come, in some cases, close enough to an acceptable result:

- Scripting process navigation: A few BPM testing tools have emerged from IBM and other business partners to run process instances and their associated activities based on certain data scenarios and scripted interactions. Those tools use a mix of BPM/BAW API invocations and RPA-like capabilities to reproduce data context and control flow navigation. With such capabilities, the state of a live process instance could be encoded as a test case in the tool. Driven by the test script, a new process instance on the cloud could

potentially be manipulated to have the same data and to be "walked" to an equivalent navigation state

- Creating migration process variants: Existing processes (i.e. models) may be decorated with new entry points that make automated/scripted navigation less cumbersome. For example, with a few entry points modeled as Start Message Events and judiciously placed along the path(s) of a process, a process instance might be started in the middle and loaded with the right data. There are many limitations to this scenario, but the need to avoid double licensing costs (due to long transition durations) may provide enough motivation for a customer to consider creative approaches
- Scripting navigation of a migration process variant: This hybrid approach, which mixes the previous two options, addresses the valid and common use case of automatically walking a process to a specified state while skipping the re-invocation of integrations that should only be called once for a process instance. In this scenario, duplicate invocations can be avoided by having the script use mid-process entry points purposefully modeled in the migration variant of the process

Note on limitations: Certain process constructs (such as Intermediate Message Events, Timers, Multi-Instance Loops, Sub-processes) and certain business requirements (related – for example – to reporting and audit trail, or time-based escalations) can significantly complicate the scripted reproduction of a process instance's state and should be examined before committing to instance transfer options.

It should be clear from the above the idea of approximating the transfer of process instances to the cloud is technically more complicated than federation, and the cost of implementing such an approach may even outweigh licensing cost avoidance benefits. But if a transition period is made artificially long because of a few process instances, and if maintenance of the on-prem environment itself is onerous, then these options should be duly considered – and, in some cases, be a reasonable choice for runtime process migration.

MANAGING MIGRATION INHIBITORS

Process Applications and Toolkits have varying degrees of transferability to the cloud. This because the cloud environment can differ from BPM/BAW on-prem in a few important ways:

- All environments in a BAW instance are accessible through the same host; the only difference is in the path for each environment:
  - Development: https://<customer>.bpm.ibmcloud.com/baw/dev
  - Test: https://<customer>.bpm.ibmcloud.com/baw/test
  - Production: https://<customer>.bpm.ibmcloud.com/baw/run
- User-ids by default are a user's email address instead of a simple login name
- The host name for the cloud environment is not the same as on-prem (obviously)
- The domain name for the cloud environment is not <customer domain name>; instead it is (<customer>.bpm.)ibmcloud.com
- BAW's (LDAP) directory is essentially not customizable compared to a customer-managed directory (where standard and/or custom attributes can be defined, populated, queried)
- The apps/systems previously accessed by processes and services running on the on-prem BPM server likely don't share the same security and session management context
- Identifiers for process instances and tasks will – by default – restart at 0
- The Performance Data Warehouse on the cloud will start empty

In consequence of the above differences, some applications/toolkits could experience the following (example) runtime issues after relocating to the cloud:

- Links and URLs to various systems (including BPM resources and REST API) could become invalid due to path differences
- User/staffing-related operations could fail due to login name inconsistencies
- Cross-site restrictions could cause errors in browser-based logic that depends on iframes or makes REST calls because domain names are different
- Integrations may stop working because a once common security context is no longer shared

Identifying inhibitors

Analyzing applications and toolkits in an overall solution to be migrated cannot reasonably be done without specialized tooling, or at least advanced knowledge of the export format, structure, and entity relationships behind BPM "projects" (the high-level conceptual container of a Process Application or Toolkit).

This is because Process Center only provides surface insight into the projects it contains, and – for the most part - Process Designer only lets authors examine what is in a project manually, artifact by artifact, and functional area by functional area.

Salient Process provides several specialized tools (categorized below as “TWX API” and “TWX Analyzer”) to systematically explore projects associated with a migration candidate solution and identify areas that may require adjustments.

TWX API

TWX API supports the programmatic access of TWX files to read, modify, or even create artifacts in a TWX file (i.e. a project exported out of Process/Workflow Center).

Through TWX API, a project can be expanded into its full dependency tree, queried for its various components, and searched for programmer-defined patterns to identify, report on, and potentially modify its migration-relevant aspects. TWX API also helps maintain the relational integrity of project parts and manage the versions of artifacts if changes are made. TWX API can open or create process applications and toolkits from BPM 8.5.6 through BAW 19.0.0.2.

TWX Analyzer

TWX Analyzer exercises TWX API intelligently to extract migration-relevant characteristics of projects and their dependencies. This, in turn, informs a migration by helping target potential areas of change and by organizing and sequencing how those changes should be made.

Below is a sampling of the kind of information and reports that are produced and/or supported by TWX Analyzer.

Asset inventory & general solution makeup

This TWX Analyzer function works on a set of TWX files to provide a high-level report of:

- How many apps and toolkits are used in a solution
- How many snapshots of the same toolkit are used through the entire solution dependency tree
- What is the make-up of each app/project in artifacts and toolkit dependencies

App and Toolkit breakdown example

This report quickly summarizes the number or apps and toolkits to be migrated. It also breaks down the toolkit dependencies by snapshots used, helping assess areas of changes that could trigger extra reconciliation and/or re-testing effort – should changes in toolkits need to be made:

Level 0 solution breakdown	Level 1 solution breakdown
Analysis result for 11 projects: Project: BPM Job Router (Snapshot: V0.34) Project: Util2 Service Delivery (Snapshot: V7.0.9) Project: Data Delivery Process (Snapshot: V6.8.03.01) Project: Util1 Service Gateway (Snapshot: V2.1.6) Project: App 2 Plant Process (Snapshot: V1.2.08) Project: Ruleset Process (Snapshot: V1.01.1) Project: App 3 Proposal Process (Snapshot: V8.1.31) Project: Work Approval (Snapshot: V2.0.95) Project: WL Service Delivery (Snapshot: V5.0.29) Project: WS High Volume Service Delivery Process (Snapshot: V0.2.9) Project: WS Service Delivery (Snapshot: V3.22.6) 41 toolkits in use – Snapshots/versions across all projects: Toolkit: XYZ Custom Controls Toolkit (8 snapshots) Toolkit: XYZ Quest Toolkit (4 snapshots) Toolkit: Task Master Toolkit (2 snapshots) Toolkit: Dojo Charting Toolkit (2 snapshots) Toolkit: XYZ Email Toolkit (4 snapshots) Toolkit: XYZ MT Ecosystem Toolkit (1 snapshot) Toolkit: 3rd Party LDAP (2 snapshots) Toolkit: XYZ SPX Toolkit (1 snapshot) Toolkit: XYZ Global Traceability Toolkit (1 snapshot) Toolkit: XYZ Master Data Toolkit (2 snapshots) Toolkit: XYZ Maximo Toolkit (1 snapshot) Toolkit: Dashboards (1 snapshot) Toolkit: XYZ Util4 Toolkit (1 snapshot) Toolkit: XYZ CMM Toolkit (1 snapshot) Toolkit: XYZ Excel Toolkit (2 snapshots) Toolkit: Responsive Coaches (1 snapshot) Toolkit: XYZ Util1 Toolkit (5 snapshots) Toolkit: XYZ CCQI Toolkit (1 snapshot) Toolkit: System Data (2 snapshots) Toolkit: XYZ LKMS Toolkit (1 snapshot) Toolkit: XYZ Common BPM Artifacts (8 snapshots) Toolkit: SWP Toolkit (1 snapshot) Toolkit: XYZ C4C Toolkit (1 snapshot) Toolkit: XYZ Remedy Toolkit (1 snapshot) Toolkit: XYZ SPARK Artifacts (4 snapshots) Toolkit: Content Management (2 snapshots) …	Analysis result for 11 projects: Project: BPM Job Router (Snapshot: V0.34) Project: Util2 Service Delivery (Snapshot: V7.0.9) Project: Data Delivery Process (Snapshot: V6.8.03.01) Project: Util1 Service Gateway (Snapshot: V2.1.6) Project: App 2 Plant Process (Snapshot: V1.2.08) Project: Ruleset Process (Snapshot: V1.01.1) Project: App 3 Proposal Process (Snapshot: V8.1.31) Project: Work Approval (Snapshot: V2.0.95) Project: WL Service Delivery (Snapshot: V5.0.29) Project: WS High Volume Service Delivery Process (Snapshot: V0.2.9) Project: WS Service Delivery (Snapshot: V3.22.6) 41 toolkits in use – Snapshots/versions across all projects: Toolkit: XYZ Custom Controls Toolkit (8 snapshots) Snapshot: V1.9.15.03 (2018/06/07 10:35:28) Snapshot: V1.9.8 (2017/11/14 05:56:29) Snapshot: V1.9.18 (2019/02/13 03:00:02) Snapshot: V1.9.9 (2018/01/24 12:56:20) Snapshot: V1.0.8.6 (2015/11/30 11:41:59) Snapshot: V1.9.2 (2017/10/31 13:17:08) Snapshot: V1.9.13 (2018/03/01 20:49:55) Snapshot: V1.9.15.01 (2018/04/27 11:23:30) Toolkit: XYZ Quest Toolkit (4 snapshots) Snapshot: V2.6.1 (2019/02/07 04:05:12) Snapshot: V2.6.2 (2019/03/26 22:58:45) Snapshot: V2.6 (2018/06/03 23:46:51) Snapshot: 1.6.1 (2015/10/15 15:09:26) Toolkit: Task Master Toolkit (2 snapshots) Snapshot: V0.5.2 (2015/04/02 14:10:44) Snapshot: V0.8.1 (2017/09/29 14:36:23) Toolkit: Dojo Charting Toolkit (2 snapshots) Snapshot: V3.5 (2017/09/29 14:41:04) Snapshot: V3.3.1 (2014/12/09 09:10:32) …

Figure 21 Overall solution breakdown report from TWX Analyzer

Per project (i.e. application or toolkit), a toolkit dependency and contained artifacts breakdown are also provided:

High-level project asset inventory

Report for "Data Delivery Process" (DDTS) / V6.8.03.01

Toolkit Dependencies (15 toolkits)
- 3rd Party E-Mail (SMTP) / V0.11
- 3rd Party UI (UI2) / V3.12
- 3rd Party LDAP (LDAP) / V1.6
- XYZ Trace Toolkit (XYZETTK) / V0.11.01
- XYZ Email Toolkit (XYZETK) / V1.0.7
- XYZ Util1 Toolkit (XYZIDTK) / V0.15.25
- Coaches (SYSC) / 8.6.0.0
- Content Management (SYSCM) / 8.6.0.0
- XYZ Master Data Toolkit (XYZMDTK) / V0.15
- XYZ Custom Controls Toolkit (XYZCCTK) / V1.9.9
- System Data (TWSYS) / 8.6.0.0
- DDTS Toolkit (DDTSTK) / V0.62
- XYZ Common BPM Artifacts (XYZCMTK) / V1.10
- XYZ Util4 Toolkit (XYZQTTK) / V0.6.2
- Util2 Artifacts (DMTK) / V0.56

Artifacts summary:
Total artifact count: 292
- process: 225
  - Client-Side Human Service: 5
  - Decision Service: 3
  - General System Service: 144
  - Integration Service: 11
  - Ajax Service: 18
  - Content Integration Service: 1
  - Service Flow: 5
  - Deployment Service: 1
  - Heritage Human Service: 37
- coachView: 10
- environmentVariableSet: 1
- SmartFolder: 2
- bpd: 3
  - Event Sub-process BPD: 1
  - Heritage BPD: 2
- twClass: 36
- managedAsset: 2
- participant: 10
- resourceBundleGroup: 1
- projectDefaults: 1
- underCoverAgent: 1

Figure 22 Project-specific breakdown report from TWX Analyzer

A key benefit of TWX Analyzer in the above reports, beyond the core consolidated data provided, is its ability to report on all Process Applications and toolkits at once. Contrast this with the painstaking manual process of obtaining the information from Process/Workflow Center, app by app and toolkit by toolkit, for the potentially numerous snapshots of the apps and toolkits involved in a solution. In many cases, reports can also be output to CSV files to facilitate further analysis.

Visualizing dependencies

Because toolkits dependencies can sometimes be dense and intricate, TWX Analyzer also provides a user-friendly and interactive way of visualizing and drilling up and down the toolkit dependency hierarchy:

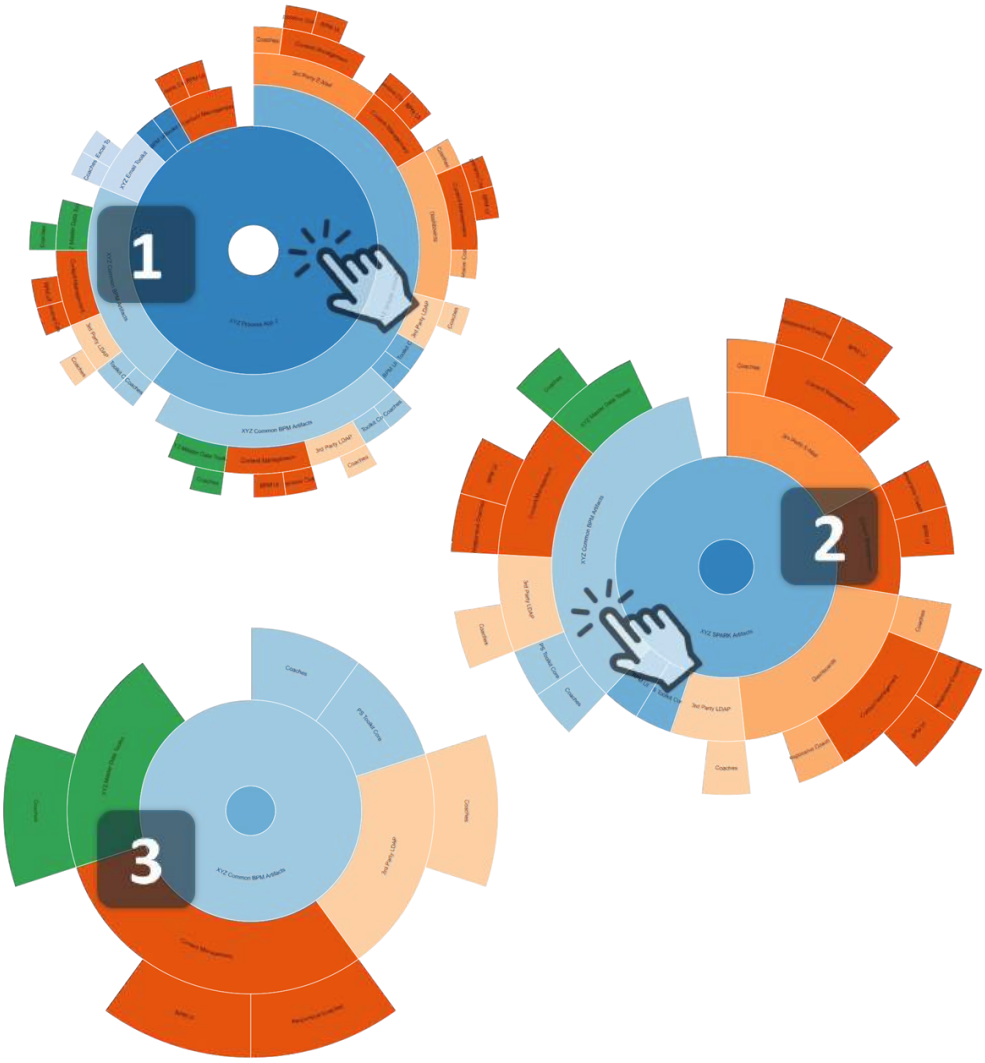

Figure 23 Interactive dependency visualization "sunburst" charts with TWX Analyzer

Solution structure and dependency tree

Sequence planning is a key component of a migration effort if toolkits in the dependency hierarchy require changes. Without proper sequencing, the propagation of updated toolkit snapshots – on one or more tracks – through the dependency tree can be onerous when multiple changes must be done in multiple toolkits.

The visualization below shows the dependency tree for one example application (represented as the leftmost node):

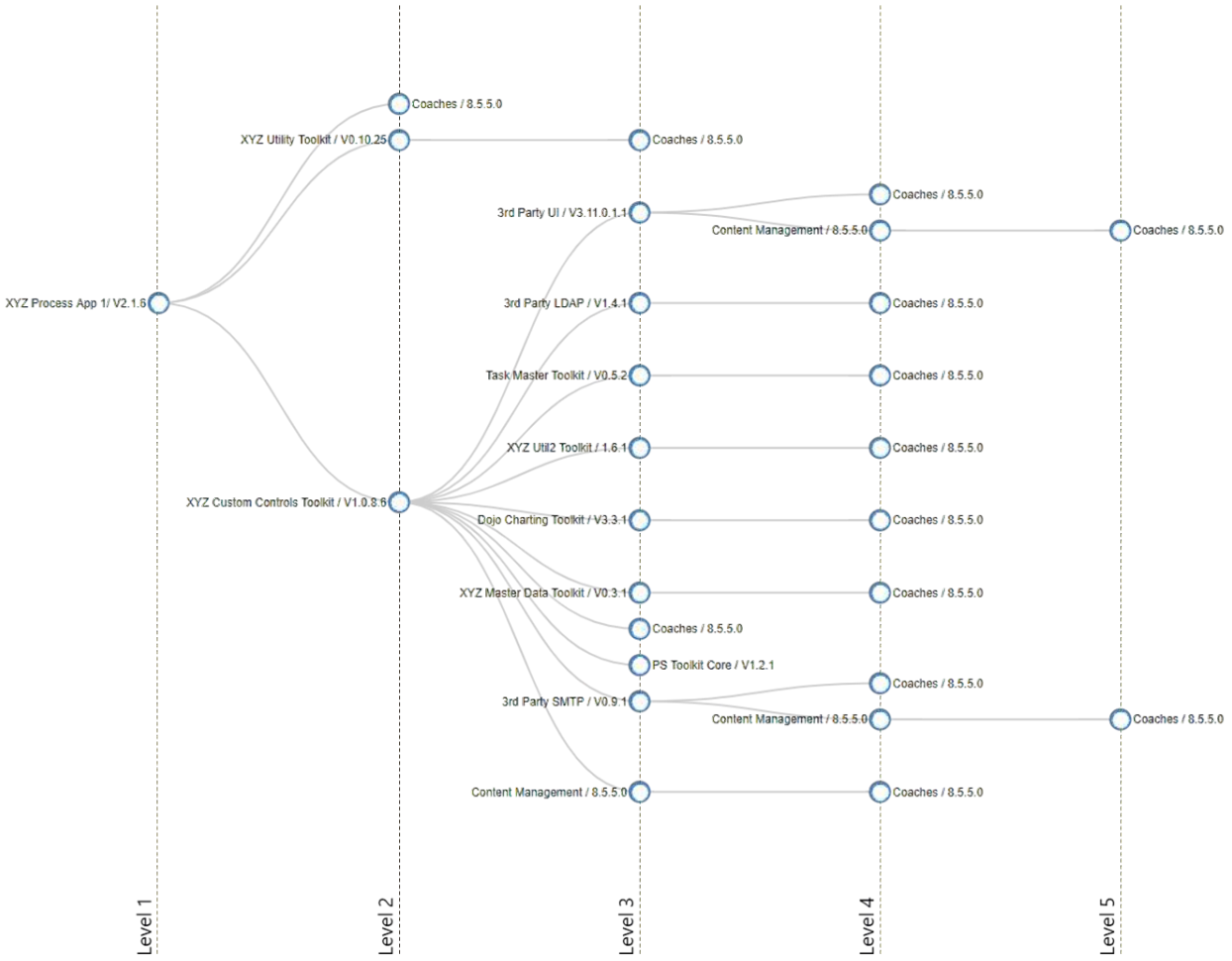

Figure 24 Project dependency tree from TWX Analyzer

Note on System Data toolkit: Because the System Data toolkit is a leaf for every branch in the dependency tree, it is not shown in the hierarchy to avoid unnecessary diagram clutter.

More than merely for visualization, TWX Analyzer can also prescribe an optimal migration order/sequence across all apps and toolkits to minimize rework inherent to changes in a complex dependency tree – as shown below:

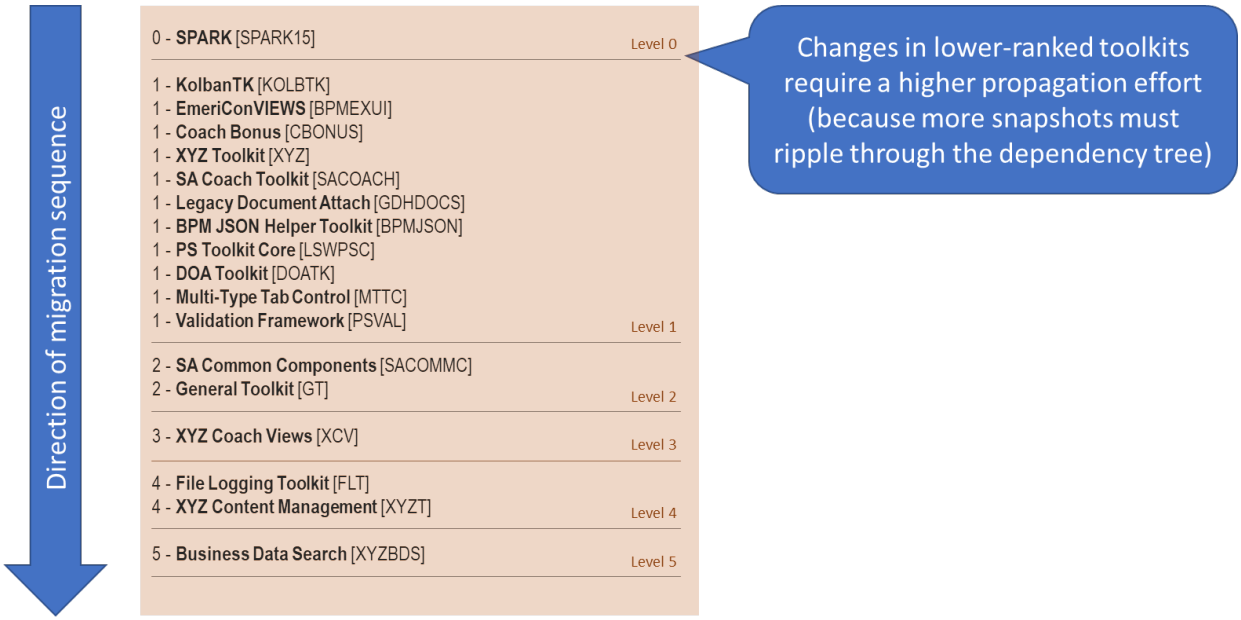

Figure 25 Optimal toolkit migration sequence from TWX Analyzer

Note on snapshots and dependencies: The toolkit precedence view above is simplified because, in this specific migration example, dependencies did not change between snapshots of the same toolkit. If dependencies changed between snapshots, then several snapshots of the same toolkit would figure in the sequence (each at its corresponding level).

Problematic patterns and implementation approaches

Several situations can cause errors or at least different behavior when an on-prem application is moved to the cloud – for example:

- URLs in client or server-side code containing /teamworks, /portal, /rest, etc.
- Direct access to the BPM database (with or without an explicit schema name)
- References to tw.system.user_loginName
- Customized XSL in Heritage Coaches
- IFrames loading non-BPM/BAW URLs
- Access to file system by server-side code

TWX Analyzer can analyze the entire solution's apps and dependency tree for telltale signs of the above issues (and others) to both report if problematic areas were found and provide the precise location of the report trigger, including:

- Project name & snapshot information
- Asset name and type containing the issue
- Line number if the issue comes from JavaScript code

Such reporting helps plan the nature and number of changes to be made as part of the migration effort. It also helps the development team address the issues detected much more efficiently.

Snapshot/track reconciliation support

Over time, toolkit versions and particularly tracks can drift apart if strict governance isn't applied to force reconciliation. Reconciliation of toolkit snapshots in the same track can be cumbersome in IBM BPM/BAW due to the lack of fine-grained artifact-level merging support in the product. Such merging can be even more challenging across tracks.

The practice of using several toolkit snapshots and tracks in a given solution can add complexity to a migration effort if migration-related changes must be made to those toolkits. The following questions should be considered:

- Should the changes be made to all snapshots and should new respective snapshots be taken (on new tracks) and propagated to their dependent projects?
- Should the different snapshots be normalized to a new "cumulative" snapshot and reflected on all dependent projects?
- What are the testing implications of either approach given the potential for regressions?

When creating a "cumulative fix" toolkit snapshot (i.e. one that doesn't introduce regressions), the ability to identify what has changed in the toolkit between snapshots on the same track or across tracks becomes very important. TWX Analyzer provides reports that show version differences across tracks and snapshots at the artifact level for a toolkit. It also allows for fine-grained comparisons at the artifact level to help create a cumulative merge result across artifact versions, across 2 or more snapshot versions.

Toolkit versioning summary

The artifact versioning report provides, for each toolkit, a roll-up of all the snapshots used (including track information), then it lists each artifact changed, with detail for each artifact version. A sample report for three artifacts changed in a toolkit is shown below:

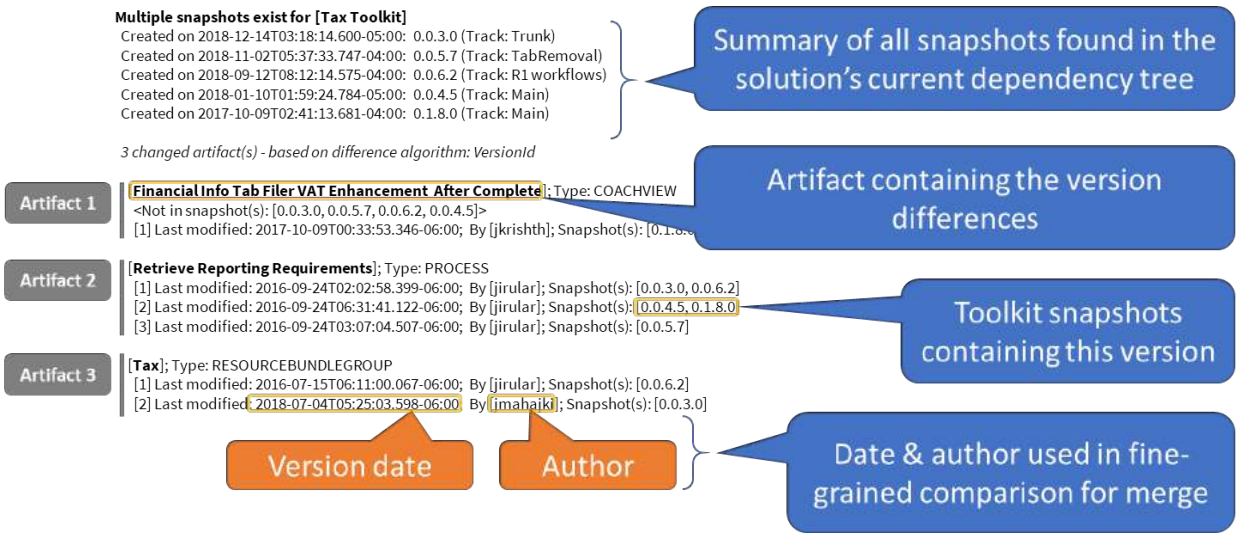

Figure 26 Toolkit versioning report from TWX Analyzer

Fine-grained reconciliation

In addition to the versioning report, TWX Analyzer further facilitates artifact comparison by copying each file associated with the artifact version to a directory structure. The structure is organized as follows:

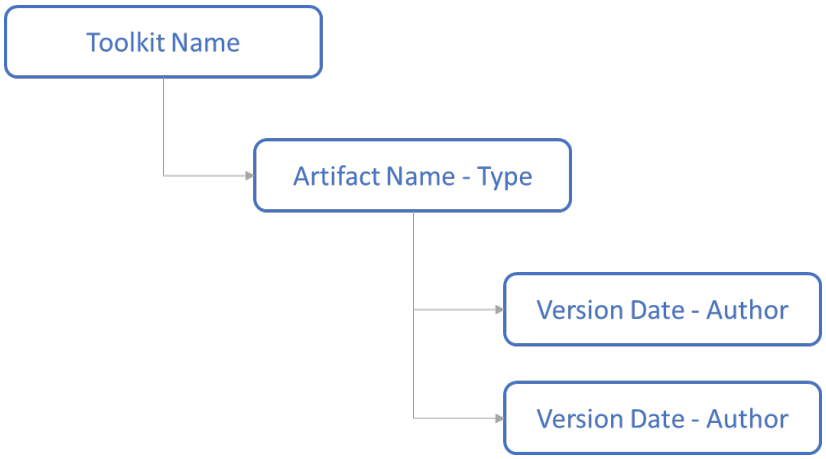

Figure 27 Fine-grained artifact comparison directory structure

The illustration below shows how the versioning report relates to the version comparison directory structure created and populated by TWX Analyzer:

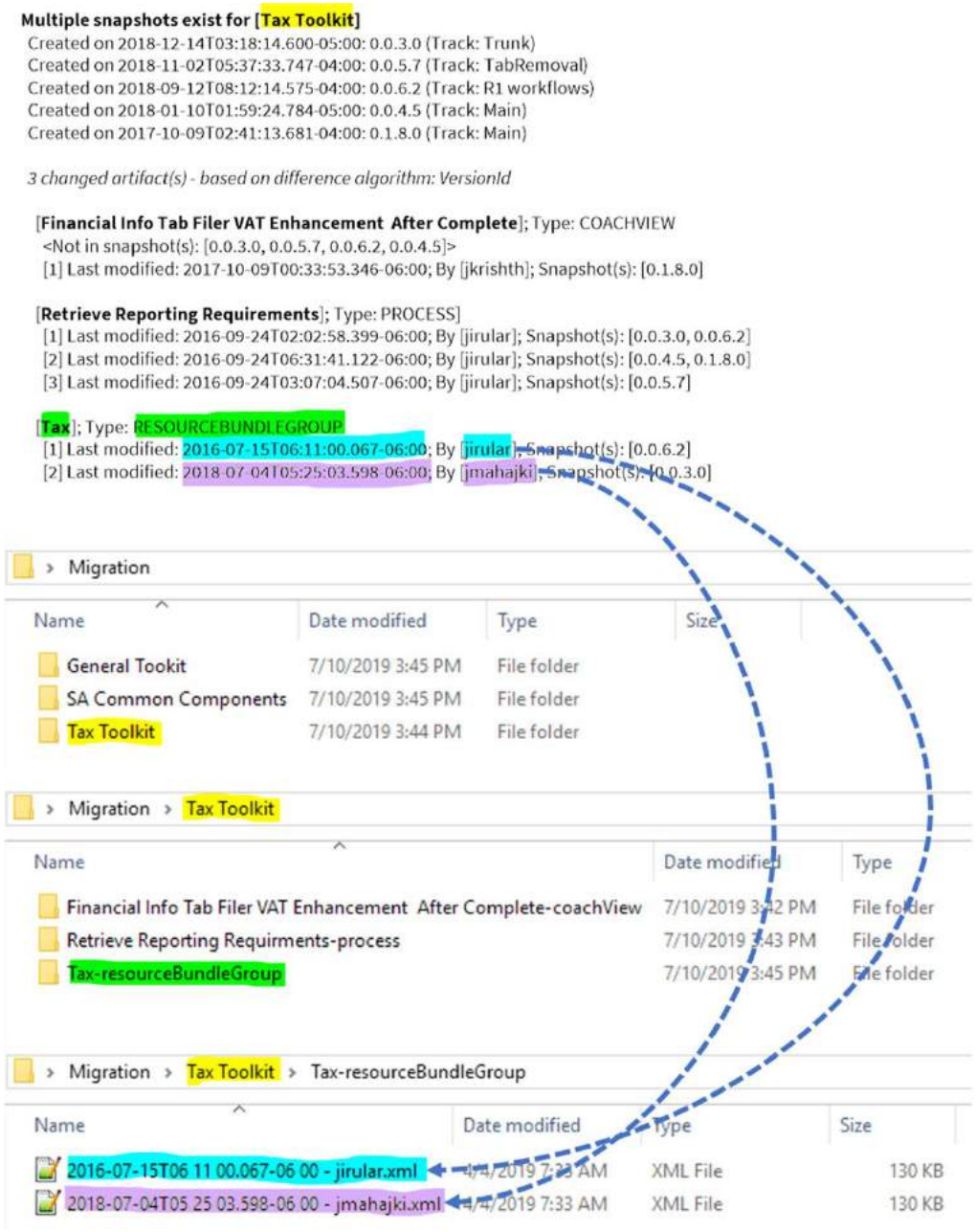

Figure 28 Mapping of TWX Analyzer versioning report with artifact comparison directory structure

Lastly, BPM/Workflow developers can review the fine-grained changes in a diff/merge tool to understand how the artifact has specifically changed over time and can – based on the changes made evident by the comparison, update the artifacts in Process Designer by manually merging

artifact implementation details or by reverting the appropriate artifact version to the tip, if appropriate. The illustration below an example of such a diff/merge-facilitated comparison:

Figure 29 Fine-grained artifact comparison to facilitate version reconciliation

Note on element ordering: IBM BPM/BAW controls the ordering of elements in the XML files that contain the metadata and implementation details of an artifact. Across versions, the ordering of elements is sometimes not preserved (though it remains semantically equivalent). Accordingly, in some cases, it may be necessary to reorder certain XML elements between two files to further facilitate the comparison.

Note on artifact detail merging: IBM BPM/BAW does not support artifact detail-level merging. Merge tools at the moment can only be used for comparison – to inform the BPM developer of what should be merged and where. The actual merging must be done manually in Process Designer.

RESOLVING APPLICATION-SPECIFIC MIGRATION ISSUES

Changes to applications and toolkits should be kept to the minimum necessary during the migration to avoid compounding testing/debugging complexity. The "minimum necessary" is the set of changes without which the solution would not run on the cloud.

The migration inhibitors discussed in "Problematic patterns and implementation approaches" provide a sampling of the kinds of issues that must be addressed before running migrated applications in the cloud. Although TWX Analyzer can pinpoint the location of a potential issue, the effort of making changes is the BPM developer's responsibility and can be time-consuming and error-prone.

Such changes might include:

- Making updates to widely duplicated code
- Adjusting how logging is implemented
- Removing artifacts that cause unnecessary maintenance challenges
- Replacing problematic/deprecated constructs with working/supported ones
- Normalizing toolkit versions across a solution's application dependency tree

The idea behind automatically analyzing large numbers of TWX files as a solution is to accelerate the process of locating potential issues and to overlook fewer of them. In the best (and not uncommon) case however, BPM developers can be saved from likely spending hundreds of hours making thousands of repetitive changes.

TWX Transformer

TWX Transformer is a TWX API-based tool that makes BPM developer-reviewed changes, based on TWX Analyzer reports, through automated scripts.

Most of the time, scripting changes means taking a source pattern/construct and mapping it to a somewhat equivalent target pattern/construct. The potential for completeness of the mapping depends on the extent to which the source and target are conceptually/semantically mappable.

At a high level, the expectation in using TWX Transformer is as follows:

- Batch changes are made as directed by TWX Analyzer and as specified by BPM developer-created adaptation scripts (the scripts typically use TWX API to make changes)
- A detailed change report is provided to BPM developers to review at a glance the location and nature of the changes made (optionally including before and after comparisons)
- After reviewing the adaptation report, a BPM developer can import the changes to BPM/BAW or make additional ones (manually, or through another script-driven batch)

Changes made through TWX Transformer scripts are (obviously) only as good as the adaptation scripts allow, and certain categories of changes may still require additional human-reviewed modifications after the scripts run. But the ability to make targeted changes and make mass repeatable updates across Process Applications and Toolkits, leaving a trace for the developer that pinpoints the exact location of areas that still require attention has proven extremely valuable.

The example below shows the effort involved in the conversion of UI controls from an old UI toolkit to the IBM BPM UI toolkit. The table contrasts manual and scripted efforts for all controls (inventoried by TWX Analyzer) in use by the solution across applications and toolkits:

Control Type	Count	Est. Manual Effort (mins)	Est. Scripting Effort (mins)	Est. Time Savings (mins)
Text	1035	2.5	180	2407.5
Horizontal Section	1003	6	240	5778
Spacer	809	1	30	779
Output Text	588	2	60	1116
Vertical Section	469	5	60	2285
Button	418	2	45	791
Select	385	5	45	1880
Table	190	20	480	3320
Decimal	189	2.5	60	412.5
HTML Text	164	2	120	208
Date Time Picker	153	3	120	339
Text Area	145	2.5	60	302.5
Responsive Section	144	6	120	744
Collapsible Section	88	5	120	320
Image	74	2.5	60	125
Dialog Section	65	15	120	855
Integer	63	2.5	60	97.5
Checkbox	54	3	120	42
Link	35	2.5	60	27.5
Timer	18	2.5	20	25
Attachment List	17	10	240	0

Horizontal Line	17	1	30	0
Attachment Uploader	15	3	240	0
Radio Buttons	13	8	45	59
Tabs	11	15	120	45
Fast Table	9	20	480	0
Service Call	9	5	60	0
Responsive Policy	6	10	240	0
Deferred Section	3	4	60	0
Simple Dialog	3	4	60	0
Data	1	3	30	0
Totals		415hrs	63hrs	366hrs

Table 1 Manual vs. automated changes: UI Toolkit conversion comparison

Note on automated adaptations: In the example above, scripting a conversion yields a 366-hour gain for the project. Additional "manual" time is still required for adaptations (for example to adjust UI logic if the new UI controls don't have the same programmatic interface as the old ones), but the time would need to be spent making those additional adaptations regardless. Hence, scripted changes still provide a significant net gain in time and productivity.

Resolving external migration issues

Applications and toolkits can contain constructs, make assumptions, and rely on dependencies that are specific to the on-prem environment. Additionally, certain patterns in processes, such as intermediate message events, imply that the systems sending back asynchronous responses to BPM/BAW need to differentiate between sending responses to BPM on-prem and BAW on-cloud, at least during the migration transition period.

Environment specific dependencies

When a solution relies on resources, data stores and services that exist on-prem, changes might be required if 1) those are no longer available on the cloud (or at least not in the same way), or 2) the cloud environment cannot exactly replicate those interactions as they occurred from the on-prem environment.

Historical reporting through the Performance Data Warehouse

Reliance on the Performance Data Warehouse (PDW) is widespread because it is an out-of-the-box extension of the product, and it is made easy to use through built-in features such as Tracking Groups. In a migration scenario, customers should plan for the following constraints:

- During the migration period (where both on-prem and cloud environments are producing reporting data), historical data is stored in two data stores (PDW on-prem and PDW on cloud). This means that:

  - A reporting continuity break will occur for reporting windows that span the migration period
  - Data regarding on-prem and on-cloud process instances and tasks will only be available from their respective PDW databases
- IBM provides no migration tooling/scripts with the product to merge the schema and records of a PDW database into another

Solutions that depend on PDW-based reporting must plan to manage the potential disruption to end-users. That potential may be mitigated during the transition if the REST API is used for performance queries in a federated scenario. Otherwise, reporting users will likely need to consolidate/aggregate some reporting results manually from both on-prem and cloud environments.

After the on-prem environment is drained of all process instances, it *may* be possible to append the schema & records from the on-prem PDW database to its on-cloud counterpart, but such a procedure has not been attempted by Salient.

Note on migratable potential: PDW-based reporting is not migratable as-is without user/reporting disruption. The disruption can be managed with expectation setting, planning and may entail procedural workarounds and gap-bridging solution enhancements.

Historical reporting based on non-deleted process instances

Instance-based reporting is based on queries against non-deleted process instances (i.e. in an active, failed, or completed state). It is not as commonly used as PDW-based reporting because of its inherent scalability concerns. Federated solutions that issue instance-based queries for reporting should be minimally disrupted during the transition period. However, all access to historical data is permanently lost with the retirement of the on-prem environment.

Note on migratable potential: Process instance-based reporting is not migratable as-is without user/reporting disruption. The disruption can be managed with expectation setting and planning and may entail procedural workarounds and gap-bridging solution enhancements.

User Login Names

The default form of user ids/login names on the IBM cloud is the user's email address – for example "user1@company.com". This is different from on-prem, which is usually in the form "user1". The discrepancy is manifested both in client and server logic. For example, in the following attributes:

- Client-side – Coach API: context.bpm.system.user_loginName
- Server-side – Server scripts in Services: tw.system.user_loginName

...both attributes would return "user1" on-prem and "user1@company.com" on the cloud. The different behavior can cause integration, reporting, or data issues that can be most simply addressed by requesting IBM Cloud support (though a support ticket) to set up the cloud instance with *Dedicated LDAP* (instead of Shared LDAP). The customer must then redefine cloud instance users with the REST API to use "user1" as the user id instead "user1@company.com".

Process Instance and Task Ids

Process Instance and task ids are positive integers determined by database sequences. They are not universally unique (obviously). When process instance ids are (for example) used as keys in business data, and when both on-prem and cloud instances use the business data store, there is a strong potential for the cloud environment to use id values that have already been used on-prem – as illustrated below:

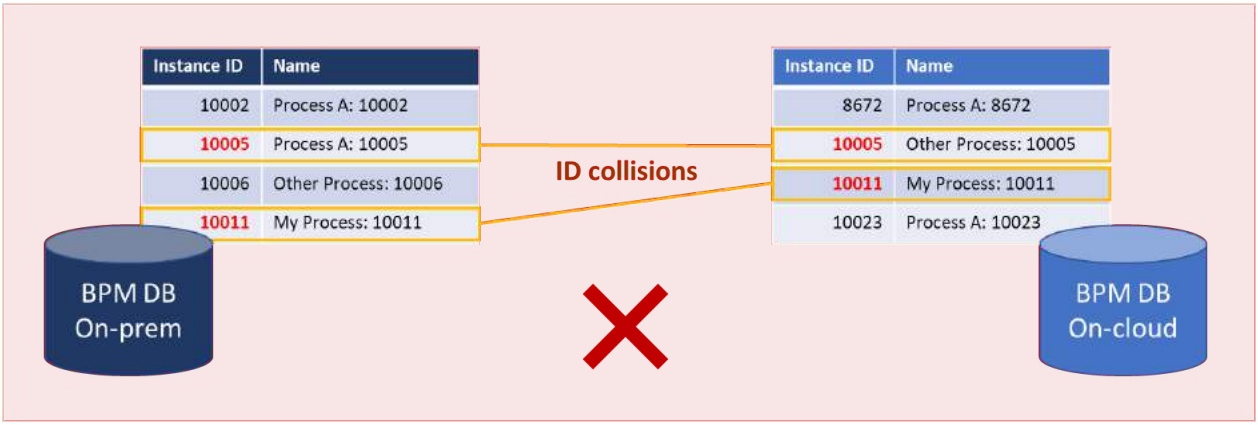

Figure 30 Id collisions between BPM/BAW instances

Overlapping/duplicate ids across BPM/BAW instances can lead to unique key violations, incorrect query results, and potential data integrity problems.

Adjusting for non-globally unique identifiers (such as process instance and task ids) allows the definition of offsets to create safe margins between BPM/BAW instances and avoid id collisions:

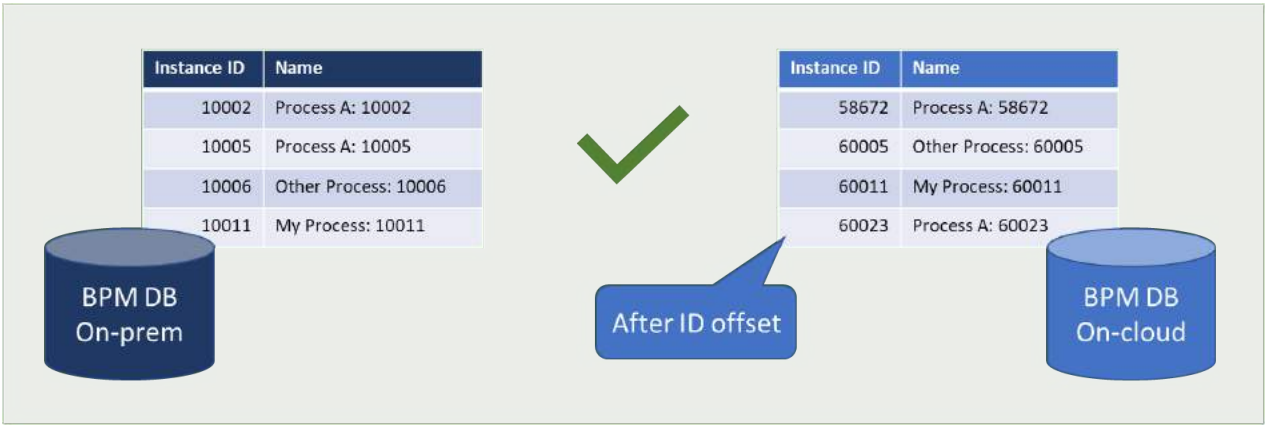

Figure 31 Id offsets to prevent cross-BPM/BAW instance collisions

Browser Integrations with non-BPM sites

Occasionally, a customer website (for example a customer portal, a SharePoint page) needs to access BAW – for example – to:

- Display a dashboard or startable human service in an iframe
- Open a link to the BAW instance (e.g. to make a service call or display a Human Service)
- Invoke the REST API on the BAW instance[†]

Such requests, however, could create a security risk (by allowing a website open in the same browser session as a BAWoC-served page to make unauthorized requests to BAW under the logged-in user's BAW credentials) for the BAW environment.

[†]Note on calling the BAW REST API: As of BAW 19.0.0.2, it is unfortunately not possible to bypass authentication from the browser for the REST API using SSO. Making REST API calls from the browser requires preemptive basic authentication, either in the browser-run logic or through a proxy that authenticates to BAW on the fly. REST API requests also require initially obtaining a CSRF token to prevent unauthorized access by unintended web pages concurrently opened in the browser.

Calls to BAW

Non-BAW-served browser content calling BAW is subject to Cross Site Request Forgery (CSRF) restrictions [3]. Under CSRF restrictions, BAW examines the referrer header and blocks non-REST API requests that are not made by its web browser-served content.

For every distinct host name in non-BAW-originated browser requests, customers can request IBM Cloud support to whitelist the originating "referrer" host (e.g. server1.company.com) to allows requests from it. If too many customer hosts serve web content that call BAW, then it may be more practical (although less secure) to whitelist the domain name (e.g. company.com).

Calls from BAW-served browser content

The reverse is also true if IBM BAW-served browser content calls non-BAW servers (assuming the non-BAW servers also impose CSRF constraints), the BAW on Cloud host (e.g. company.bpm.ibmcloud.com) should also be white-listed for those servers.

BPM Server URLs

Some customer apps and toolkits may contain logic that creates links to make HTTP(S) requests. Perhaps the logic uses hard-coded paths (relative or not) to build URLs used to make requests.

If those URLs point to the BPM/BAW server (or to a Web or Enterprise Application that was hosted on-prem on the BPM server), they may no longer be valid. For example, a Coach View might make a BAW REST API call through a URL such as "/rest/bpm/wle/v1/systems".

At runtime, on-prem, the full valid URL may resolve to:

https://bpm1.company.com/rest/bpm/wle/v1/system

However, on the cloud, the same URL will not work because each environment on the BAW instance is addressed through an added path segment ("/baw/dev" for Development, "/baw/test" for Test, and "/baw/run" for Production).

[3] See Technote: Configuring Cross-Site Request Forgery (CSRF) protection in IBM Business Process Manager (BPM)

Accordingly, the correct REST API URL for the previous "systems" REST API in the Development environment would be: "/baw/dev/rest/bpm/wle/v1/systems", resolving at runtime, on Human Service UI served by the cloud, to:

https://company.bpm.ibmcloud.com/baw/dev/rest/bpm/wle/v1/system

Developers get help from BAW to build correct, environment-relevant URLs. For example, for code in Coach Views, the <viewref>.context.contextRootMap [4] object provides properties such as:

- **rest:** Resolves to "/baw/test/rest/bpm/wle" in the cloud TEST environment
- **processPortal:** Resolves to "/baw/run/HeritagePortal" in the cloud RUN (production) environment

Using context.contextRootMap, UI developers can easily adapt on-prem specific code with logic that is both on-prem and cloud-compatible.

Asynchronous dependencies

When processes contain asynchronous interactions with outside services, the services call back to the process (for example) through an Intermediate Message Event (IME). In such a case, the IME is associated with an Undercover Agent (UCA) which itself is triggered by a service interaction (invoked with a SOAP or REST call). Services that call back to BPM/BAW must connect to the BPM/BAW server to issue the callback.

[4] See the context object in the IBM Knowledge Center

If the same outside service is used for both on-prem and on-cloud environments (as shown below), the service may not have the ability to know which server to asynchronously reply to:

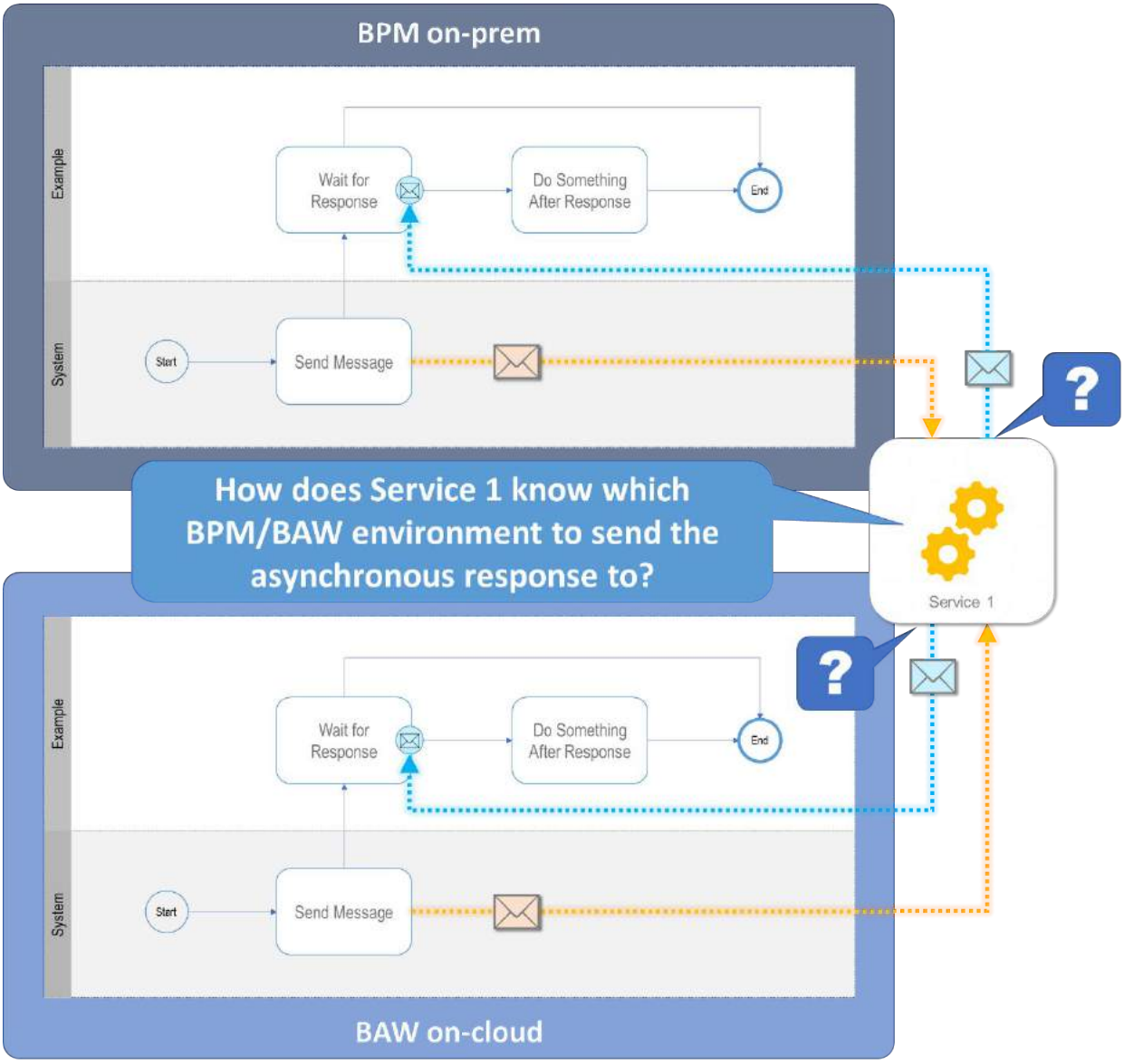

Figure 32 Async interactions with on-prem and cloud environments and shared outside service

If the outside service is not able to use the origin of a request to provide an asynchronous response, adaptations could include:

- Option 1: Making the service origin-aware and use the origin information to target the proper server for the asynchronous reply
- Option 2: Creating a new dedicated instance of the outside service that is configured to reply to the BAW on Cloud target

Option 2 is illustrated below:

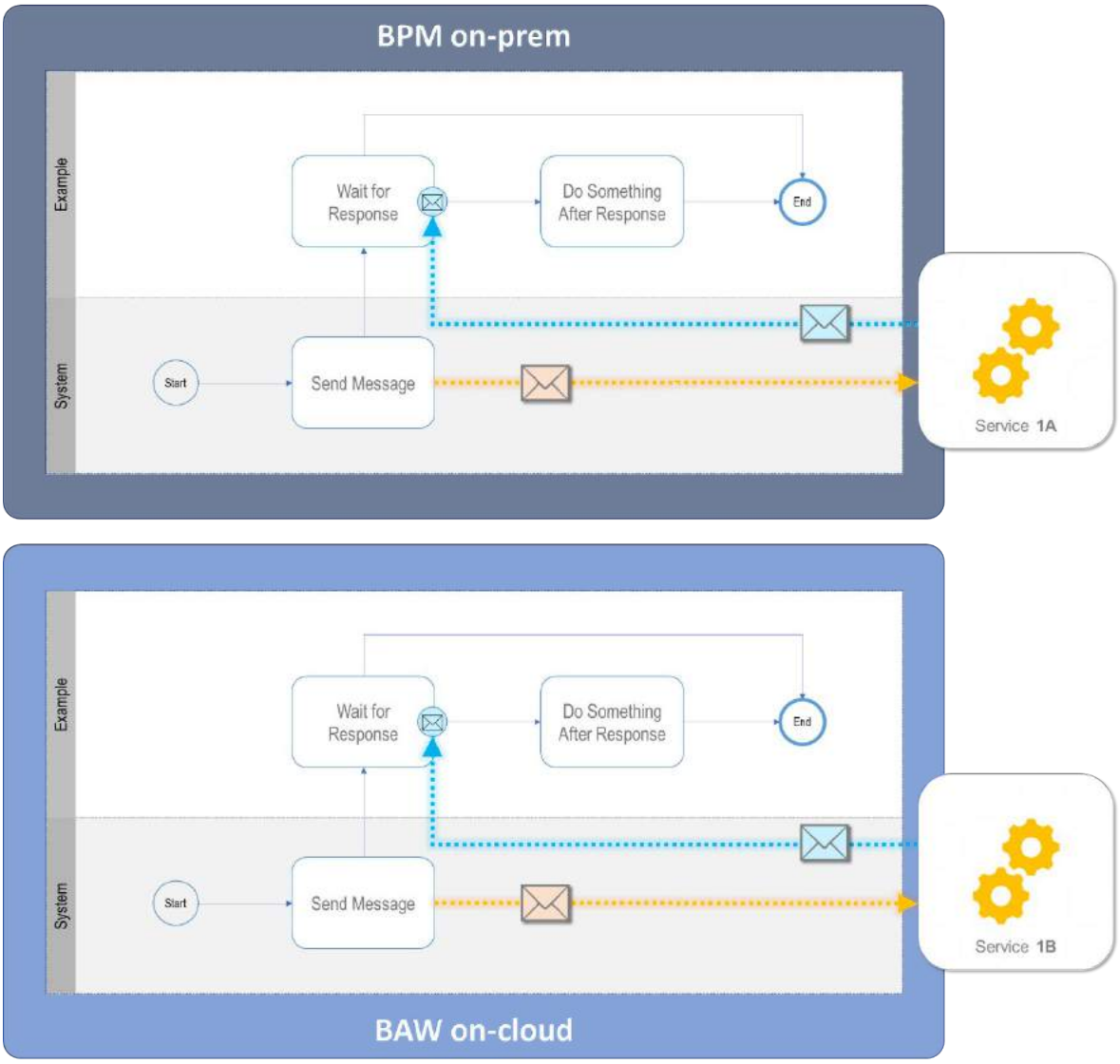

Figure 33 Async interactions with on-prem and cloud environments and dedicated outside services

BAW REST API dependencies

By default, Cross Site Request Forgery (CSRF) restrictions are enforced for BAWoC, whereas they may not be on-prem. This means that non-BPM-hosted application code that uses the BPM/BAW REST API may require minor adaptations to request an initial CSRF token (using the /system/login REST API) before making subsequent calls to the API.

PLANNING FOR REPORTING CONTINUITY

The requirement to run on-prem and on-cloud environments in parallel during the transition period can make reporting continuity challenging if a solution relies on the Performance Data Warehouse (or makes runtime queries against non-deleted instances) for reporting.

On the other hand, solutions that store historical/reporting data in a customer-managed database (populated, for example, using custom-built capabilities instead of Tracking Groups to save performance data, or using the Dynamic Event Framework to capture performance data and save it to the customer-managed database) may not be impacted at all by the migration.

Migration with Performance Data Warehouse-based reporting

Some business requirements associated with performance reporting for a workflow solution may tolerate a split in historical reporting.

For example, instance-specific reporting is also implicitly environment-specific, and the user can obtain associated reports from one of the two environments:

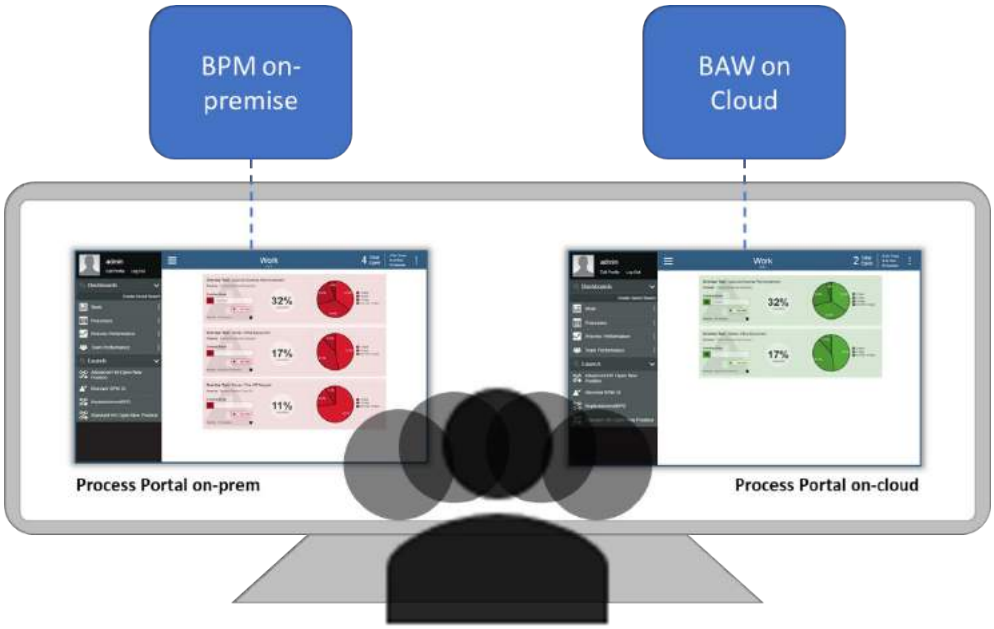

Figure 34 PDW-based reporting split between environments

However, even though the start dates of all new instances on cloud will often be more recent than those on-prem, one cannot make such an assumption about the end/completion dates. Because of the often-unavoidable reporting overlap between the on-prem and cloud environments during the transition period, end-users may need to consolidate/aggregate PDW-backed reports between environments manually.

Post transition PDW consolidation

Unless specific steps are taken to preserve the content of the on-prem PDW database, historical data (especially as accessed through BPM-served reports) will be permanently lost after the environment is decommissioned. The following mitigating options might be considered:

- Consolidated data (as accessible through tracking group-specific views) may be duplicated to a new data store and saved for later use. This is a simple option that creates a marginally usable outcome
- The on-prem PDW schema and content may be merged into the cloud PDW database. This option "magically" restores all historical data into the cloud environment. This is a complex option with an extremely usable outcome but requires heavy data manipulation (for example to adjust tracking group and tracking field ids) to create exploitable data on the cloud

Migration with custom performance reporting

If a solution's historical data is kept in a custom data store, both on-prem and cloud environments connect to and add records to the custom data store. This effectually creates a federated reporting data store, which – depending on the design of the reporting function – could immediately provide a consolidated view of both environments:

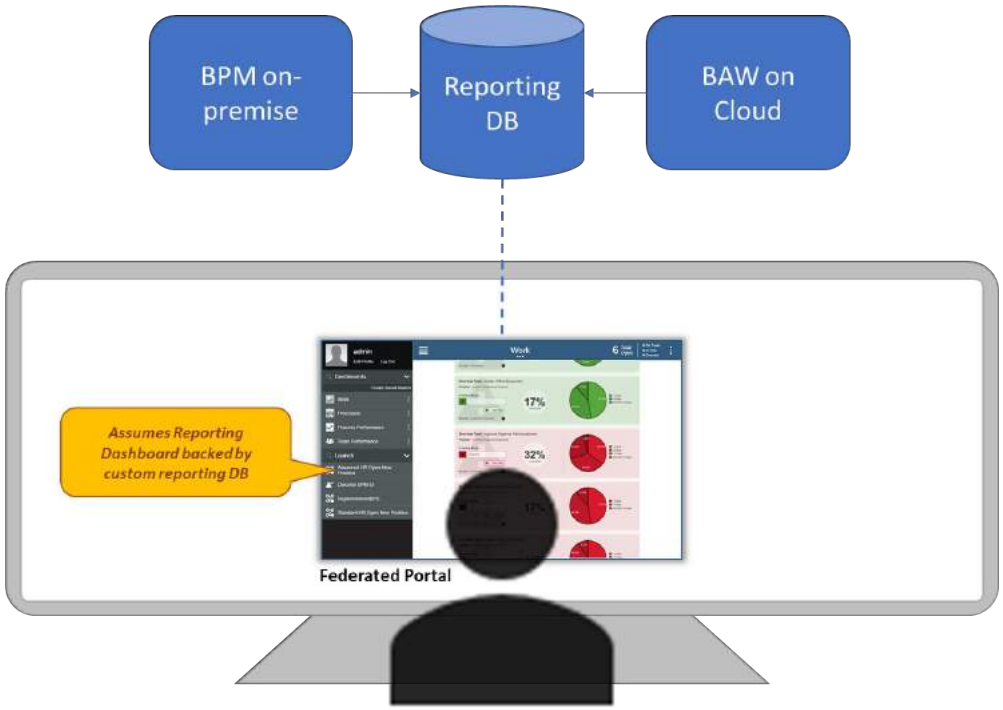

Figure 35 Custom reporting providing seamless reporting between environments

Adaptations may still be needed, especially if – for example – reports contain links that open tasks or process instance information (because those tasks/instances may be on a different environment than the one rendering the portal).

ONGOING MAINTENANCE

One of the key benefits of an IBM-managed BAW cloud environment is the significant reduction in the maintenance effort for the platform (for example in general upkeep and upgrades) compared to an on-prem installation. However – at least as of BAW on Cloud v19.0.0.2 – certain aspects of user and group management require more, not less, maintenance effort for customers.

User & group synchronization

BAW uses its own user/group directory instead of the customer's LDAP directory. Although user/group management REST APIs are provided to query, create, update, delete users and groups, no IBM-provided mechanism currently exists to automatically synchronize (initially then periodically) a customer's LDAP definitions with the cloud.

Salient's Java-based User/Group Synchronization tool bridges that gap by providing a flexible and configurable way of reconciling differences between a customer's repository (the entire directory or a subtree) and BAW's user/group directory. The tool is designed to access any javax.naming-compatible LDAP repository (such as Microsoft Azure AD, IBM Tivoli Access Manager, Sun One Directory Server) and uses the BAW on Cloud user/group management REST API to perform the actual synchronization.

A summarizing illustration of the tool is provided below:

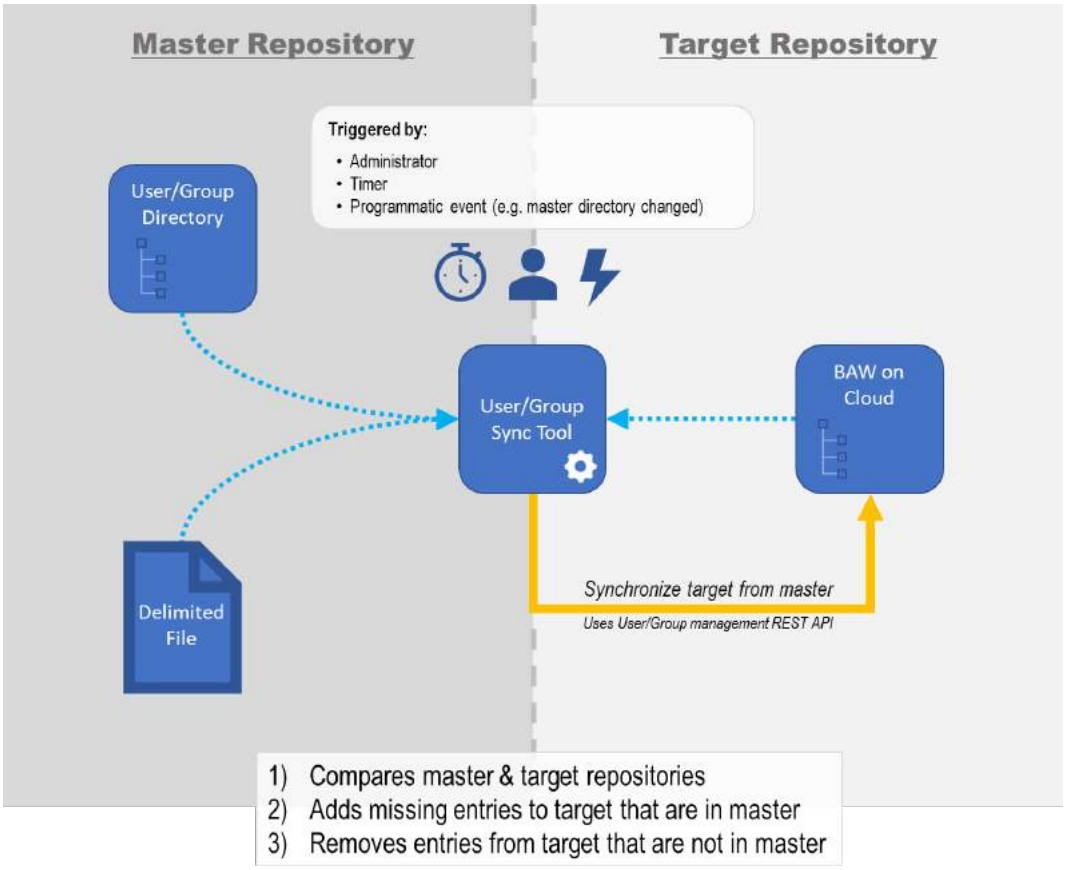

Figure 36 Salient User/Group Synchronization utility for BAW on Cloud

Appendix

Accelerator Reference

The following Salient Process accelerators are routinely used as part of migration solutions:

- Federated Portal
- TWX Analyzer
    - Solution Structure Analyzer (script)
    - Track & Snapshot Comparator (script)
    - Dependency Tree Analyzer (script)
    - Unreferenced Artifact Analyzer (script)
    - Toolkit Asset Usage Analyzer (script)
    - Global Usage Finder (script)
- TWX Transformer
    - TWX Toolkit Version Normalizer (script)
    - UI Toolkit Migrator (script)
- Cloud Storage Manager
- Cloud Connectivity Validator
- User/Group Synchronization Tool

ABOUT SALIENT PROCESS

Salient Process is a privately held award-winning company headquartered in Sacramento, CA. We are an IBM Gold Business Partner and the creators of the IBM BPM UI. Salient Process was honored to receive the 2017 IBM Cloud award for most innovative IBM Cloud partner.

We have a deep heritage in IBM BPM and ODM, dating back to the inception of those solutions. Our executive team and many of our expert consultants began their careers working with these solutions, prior to them becoming IBM products.

Our success is a direct result of our passion for your success, thus the reason for our tagline of "Your Process, Our Passion." As part of that passion and focus on your success, our goal is to work ourselves out of a job by enabling you to the point of self-sufficiency. We help your company to the point you are free to be great and enable you to stay there.

Whether it be workflow, rules, content, tasks (RPA), or data capture, we have the highly skilled resources to help you achieve your business goals.

Please contact jstange@salientprocess.com to consult with one of our experts for free.

Salient Process
6726 Fair Oaks Blvd Ste 403
Carmichael, CA 95608

www.salientprocess.com

Published in the United States of America

This document is current as of the initial date of publication and may be changed by Salient Processat any time.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary. THE INFORMATION IN THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT. IBM products are warranted according to the terms and conditions of the agreements under which they are provided.

The client is responsible for ensuring compliance with laws and regulations applicable to it. Salient Process does not provide legal advice or represent or warrant that its services or products will ensure that the client is in compliance with any law or regulation.

Statement of Good Security Practices: IT system security involves protecting systems and information through prevention, detection and response to improper access from within and outside your enterprise. Improper access can result in information being altered, destroyed, misappropriated or misused or can result in damage to or misuse of your systems, including for use in attacks on others. No IT system or product should be considered completely secure and no single product, service or security measure can be completely effective in preventing improper use or access. IBM and Salient Process systems, products and services are designed to be part of a lawful, comprehensive security approach, which will necessarily involve additional operational procedures, and may require other systems, products or services to be most effective. SALIENT PROCESS DOES NOT WARRANT THAT ANY SYSTEMS, PRODUCTS OR SERVICES ARE IMMUNE FROM, OR WILL MAKE YOUR ENTERPRISE IMMUNE FROM, THE MALICIOUS OR ILLEGAL CONDUCT OF ANY PARTY.