/MODERNIZE IAM

ACCELERATORS

ForgeRock Solution Guide

Table of Contents

/ Author	/ Action	/ Date	/ Version
Andrei Dumitru	Draft - Template	2019-11-20	0.1
Andrei Dumitru	Updated based on team feedback Added toolkit explanation	2019-12-16	0.2
Andrei Dumitru	Changed document structure to map each individual toolkit	2019-12-16	0.3
Daniel Coman	Update documentation for v7.0	2021-01-18	1.1

1 Introduction

The purpose of this document is to provide guidance for customers and partners to accelerate migration projects from legacy IAM systems to the ForgeRock Identity Platform.

The target audience for this document is technical staff (enterprise architect, solution architect, integration architect) with a general understanding of identity and access management systems.

1.1 Glossary

The following terms and abbreviations are used in this guide:

Term	Description
AM	ForgeRock Access Management
SSO	single sign-on
IG	Forge Rock Identity Gateway
RP	reverse proxy
legacy IAM	legacy module (commercial or purpose-built) that handles user authentication and/or authorization
API	application programmable interface
OIDC	OpenID Connect
RS	resource server
PEP	policy enforcement point
PDP	policy definition point
TLS	transport layer security
REST	representational state transfer

1.2 Key Requirements of Migration Projects

The following key requirements are typically seen in migration projects:

- User profile migration
- User password migration
- LDAPv3 custom schema migration
- Group membership migration
- Restricting unauthenticated user access to protected resources
- Protecting applications using a PEP
- SSO between legacy IAM and ForgeRock IAM
- Migrating from legacy web Agents and legacy J2EE agents to IG
- Asserting user identity information to protected applications
- Authentication module migration (including MFA)
- Application migration
- Authorization policies migration
- Migrating federation agreements, COTs, SP and IDP definitions
- Migrating legacy IAM authentication UI pages to use ForgeRock authentication APIs

The Modernize IAM accelerators currently cover the following requirements:

- User profile migration
- User password migration
- LDAPv3 custom schema migration
- SSO between legacy IAM and ForgeRock IAM

1.3 Legacy IAM Architecture

The following typical legacy solution architecture has been considered as a baseline for the Modernize IAM Accelerators:

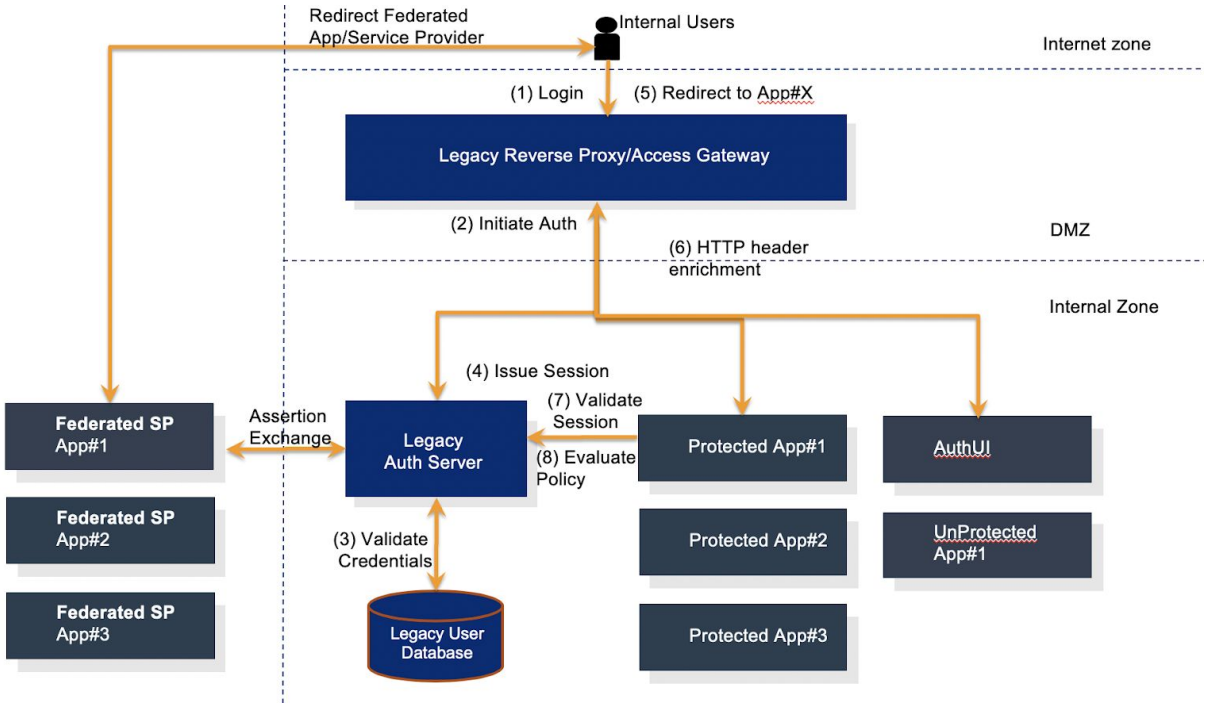

Figure 1 - Legacy Deployment

System	Role
Legacy auth server	Validates authentication and authorization requests
Legacy reverse proxy	Unified point of entry inside the domain
Legacy user database	Typical LDAP or RDBMS user store
Protected apps	Existing application integrated with legacy IAM
Auth UI	Legacy UI pages for authentication (login, logout)
Unprotected apps	Existing application not protected by legacy IAM

2 Migration Strategy

2.1 Strategy Assessment

Complex migration projects typically require a thoroughly detailed design engagement requiring a 360 degree view on the baseline deployment and the target envisioned system.

The following project specific parameters might impact the migration strategy best suited to a Customer deployment:

1. Number of applications integrated in SSO with legacy IAM
2. Existing applications technical capabilities for a standard based integration via OIDC or SAMLv2
3. Availability of APIs/SDK in legacy IAM to expose authentication and profile retrieval methods
4. Frequency of user changes (attributes changed, group membership changes, authorization information changes)
5. Existing EOL applications that cannot be adapted to a new IAM system
6. Identity assertion being sent to applications in a proprietary format
7. Existing applications supported by third-party vendors and the associated impact of the migration in such scenario (timeline, scope and other types of dependencies)
8. Custom LDAP schema
9. Security standards deployed (OAuth2, OIDC, SAMLv2) in legacy IAM
10. Existing SAMLv2 federation agreements in place with third-party entities
11. Custom authentication modules or custom point-to-point integrations
12. SLA – maximum planned maintenance available for the legacy IAM service
13. Rollback option available

2.1.1 Number of Applications Integrated in SSO with legacy IAM

As the number of applications grows, so does the complexity of the migration project, as IAM is typically the entry gate in multiple business critical applications that require 24x7 availability.

This is usually due to multiple factors:

- Changes need to be correlated across multiple systems
- The business impact is larger
- Unavailability and app SLA needs to be determined
- Applications are being supported by different vendors and different platforms

The best practice is to create and maintain an application registry together with appropriate business and technical contact details, and to start the assessment with each application. Doing this identifies the roadmap and the technical solution. During the detailed design phase, a dedicated assessment for each application is typically required.

The impact of migrating from legacy IAM to ForgeRock IAM depends on the deployment specifics. There could be little or no impact, or there could be important changes that require development and coordinated testing.

With deployments of tens or hundreds of applications, migration waves or chunks may be required to minimize the operational impact of production systems. With this type of use case, coexistence and SSO between legacy IAM and ForgeRock IAM is often needed.

2.1.2 Existing OIDC or SAMLv2 Support

When legacy IAM systems support OIDC or SAMLv2 standards, a typical IDP to SP or IDP to IDP integration could facilitate the migration:

1. With OIDC-ready or SAMLv2-ready applications, the switch to ForgeRock IAM can be designed to follow the standard and minimize impact.
2. With applications that use proprietary tokens, orchestrated SSO flows need to be designed and implemented in ForgeRock IAM to support coexistence.In this case, the AM-based SSO toolkit can help customers accelerate migration.

2.1.3 Availability of legacy IAM APIs/SDKs

Unidirectional or bidirectional SSO between legacy IAM and ForgeRock IAM requires that the legacy IAM exposes the following capabilities:

- Validate existing legacy IAM session
- Authenticate user with credentials (username, password)
- Retrieve user profile

During the orchestrated SSO flows in ForgeRock IAM, these APIs are called in order to facilitate bidirectional SSO. In environments where session synchronization is required between the legacy IAM and ForgeRock IAM, the ability to refresh sessions might also be required in the legacy IAM.

2.1.4 Frequency of user profile changes

User profiles and credentials migration is a very important step in any migration project. The requirements usually are:

- Accommodate LDAP or RDBMS user stores
- Allow full or incremental migration
- Migrate identity attributes
- Migrate user passwords
- Migrate group membership
- Map identity information based on specific rules (eg. organizationalUnit, user status, user preferences)
- Adapt directory schema
- Allow monitoring/preview of the migration process

In typical customer deployments, user changes are pushed via the legacy identity lifecycle systems to the user store (either LDAP or RDBMS).

Some use cases, such as HR for employee IAM and CRM processes for CIAM, trigger updates in user profile identity. If these changes are frequent, a one-time (or even batch) migration process of several hundred thousand identities from legacy IAM to ForgeRock IAM could require switching the legacy IAM to operate in read-only mode during the migration.

In the case where bi-directional SSO is used, with customers being able to login both directly in legacy IAM but also in Forgerock IAM in parallel, synchronizing user profile and password information between the two systems can be very complex.

This is one scenario where a centralized JIT provisioning can massively ease the deployment process and simplify the rollout process while minimizing risk.

In the case of a cut-off migration (one time move from legacy IAM to Forgerock IAM with no coexistence) a bulk migration process for the user profiles can help improve the speed of the deployment and keep the downtime window to minimum. This is where ForgeRock IDM comes into play, orchestrating the migration of large data sets between source and target systems, while still being flexible to allow identity information mapping. ForgeRock IDM reconciliation, either on demand or scheduled, can also help achieve on-going direct sync if such a requirement exists.

2.1.5 Identity Assertion in Proprietary Format

Applications integrated in SSO often require identity based attributes in order to understand who the logged-in customer is, so they can personalize the journey and the user's experience in the application.

The following set of attributes are usually mandatory for the applications:

1. Username/UserId
2. Groups
3. Profile Attributes (email, first name, last name)

The traditional approach to this has been HTTP header enrichment at the legacy policy agent level-adding a set of headers with a specified name to the HTTP header. The additional headers are then handled in the application.

For example:

- IBM ISAM standard headers - **iv-user, iv-groups**
- RSA standard header - **ct-remote-user**
- OAM standard header **- oam_remote_user**

In the early phases of migration projects, a typical approach is to preserve the header names and format in the next generation IAM, so that applications are less impacted by the change.

2.1.6 SLAs and Maximum Planned Maintenance Windows

Business critical systems protected by IAM, with 24x7 and 99% availability target, might not allow for more than two or four hour maintenance windows. Therefore, production deployment for next generation IAM should consider these restrictions and allow risk mitigation techniques like phased migration, coexistence and partial migration or JIT provisioning.

2.1.7 Approach to EOL Applications

In typical customer deployments, EOL applications connected to legacy IAM systems can no longer be adapted for next generation IAM systems, but cannot be decommissioned. Their functions need to be preserved, and the

protection level offered by the legacy IAM needs to be migration to the next generation IAM system.

As with traditional SSO integration, these systems are integrated via web agents or J2EE agents that are part of the legacy IAM stack. Agents typically enforce SSO and add user profile attributes via HTTP header enrichment in downstream HTTP calls to integrated applications.

The following options are available for customers in this situation:

- Implement SSO from ForgeRock IAM to legacy IAM by generating the proprietary legacy SSO token. This lets existing legacy agents validate the legacy tokens and function in the same way they're currently doing it.
- Remove the web or J2EE agent functionality from the application and replace the agent functions with ForgeRock Identity Gateway. Include HTTP header-based enrichment integration from the ForgeRock Identity Gateway.

3 ForgeRock IG-Based SSO & JIT Toolkit

3.1 Target Customer Deployment

The ForgeRock SSO & JIT User Migration Toolkit accelerates migration activities in customer scenarios with one or more of the following requirements:

Topic	Answer
Bi-directional SSO between legacy IAM and ForgeRock IAM; legacy IAM does not expose programmatic authentication APIs.	YES
Legacy IAM modules or configuration can be modified or changed to integrate using SSO with ForgeRock IAM.	NO
Application migration needs to be performed in chunks; legacy IAM and ForgeRock IAM need to coexist.	YES

The application DNS domain needs to change during migration, but SSO with legacy IAM needs to be preserved.	YES
User profiles can be exported from the legacy IAM store in bulk.	NO
User passwords are stored with custom hashing in the legacy IAM user store, making migration using bulk export impossible.	YES
User passwords can be exported and imported between legacy AM and ForgeRock AM in LDIF format.	NO
Applications involved in SSO should be migrated in phases between legacy IAM to ForgeRock IAM.	YES
User profiles and passwords can be synced bi-directionally between legacy IAM and ForgeRock IAM	YES
The authentication module used is user and password.	YES

3.2 High Level Reference Architecture

To accelerate migration of complex deployments, similar to the one described above in chapter 1.2.1, ForgeRock is introducing the following JIT reference architecture solution based on the following ForgeRock Identity Platform core components:

- ForgeRock Identity Gateway
- ForgeRock Access Manager
- ForgeRock Directory Server
- ForgeRock Identity Manager

ForgeRock Identity Gateway has unique capabilities for both reverse proxy, including transparent proxy, and Access Management-enabled functions. IG can perform complex tasks, such as **JIT provisioning** between legacy IAM and ForgeRock IAM and also **Extensible migration framework** for Bi-directional Single Sign On.

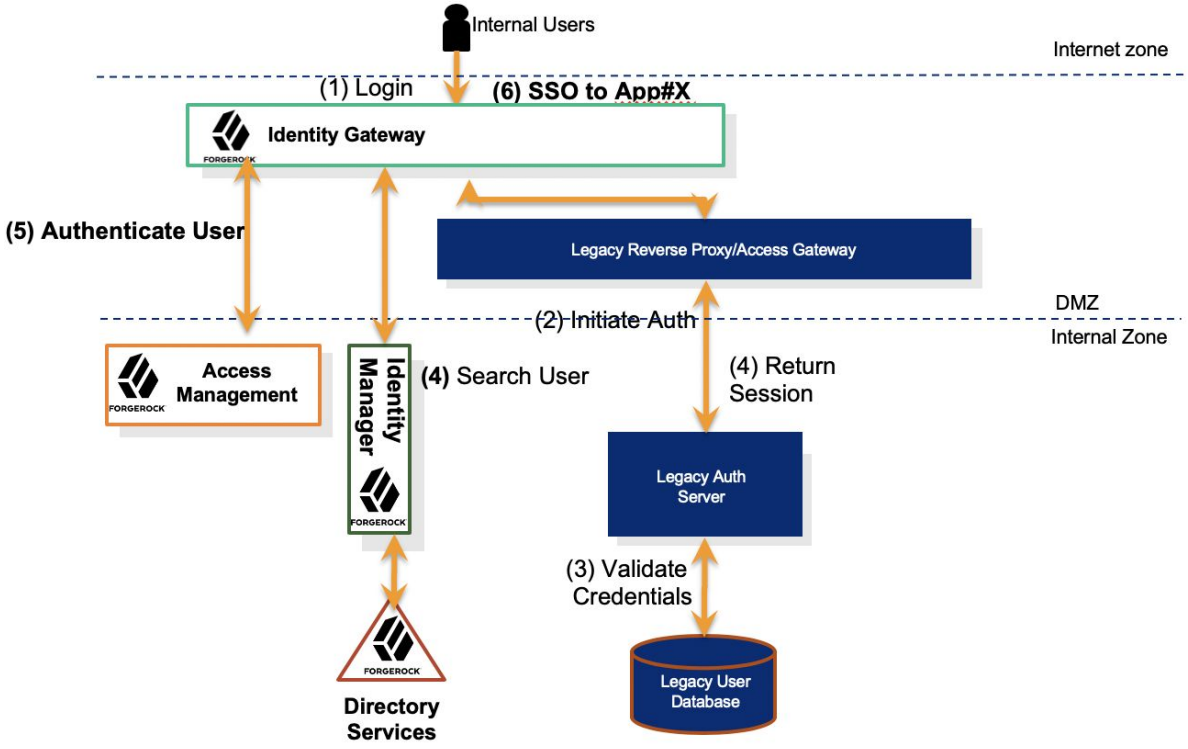

Figure 2 - JIT & SSO Reference Architecture

All interactions between ForgeRock Identity Gateway and legacy AM are performed in vendor-specific plugins using Java Provider interface, which allows the framework to be easily extended.

3.2.1 Scope Definition

The toolkit provides a mechanism for IG to intercept legacy IAM authentication requests and implement JIT user provisioning to ForgeRock IAM during this legacy IAM authentication transaction.

When legacy IAM authentication has been successful, the toolkit programmatically authenticates the user to ForgeRock Access Manager and sends the ForgeRock SSO token to the user agent.

The JIT requires that the user is successfully authenticated against the legacy IAM system and that a legacy token has been issued.

JIT orchestration is performed in ForgeRock Identity Gateway. The provisioning operation is executed via ForgeRock Identity Management standard managed user provisioning APIs.

The framework can be easily extended to support migration from any legacy IAM platform that is capable of exposing client SDKs/APIs for operations such as:

- Validating existing legacy IAM tokens
- Authenticating with a username and password

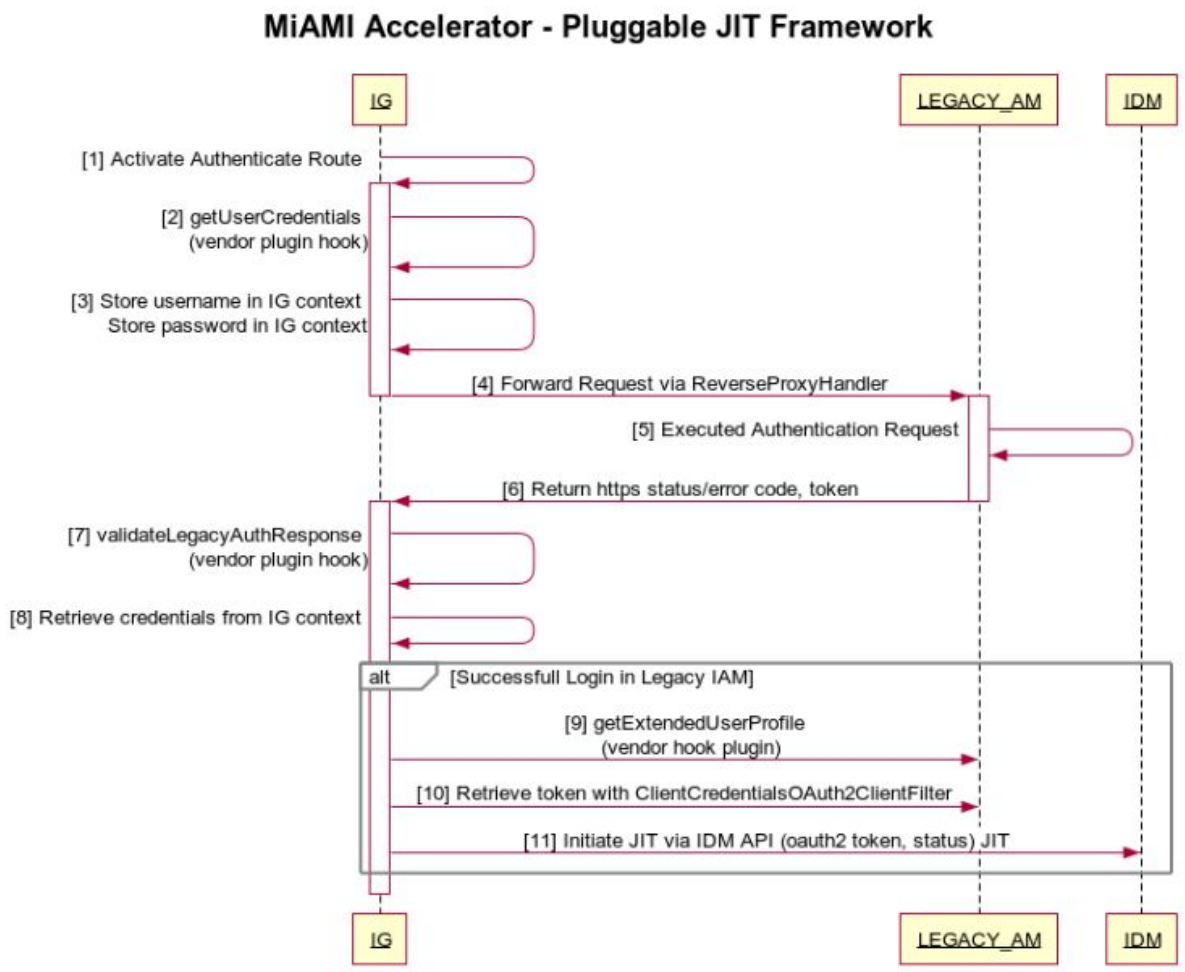

Figure 3 - JIT Framework

The following user profile attributes are provisioned:

- UserID (unique identifier) – captured from the form filled in by the user
- UserPassword – captured from the form filled in by the user

The following high-level sequence of execution occurs:

- The user navigates to the legacy IAM provider's standard login page
- The page is presented to the user via ForgeRock IG in a transparent way
- The user fills in their credentials and submits the form to ForgeRock IG
- ForgeRock IG captures the credentials and stores them in the HTTP request context (non-persistent, only in memory), and then forwards the request to the backend legacy IAM for authentication
- Upon receiving the response from the legacy IAM system, ForgeRock Identity Gateway checks the status of the authentication event; if

authentication was successful. IG provisions the user in ForgeRock IAM using the ForgeRock IDM APIs protected by an oauth2 token

- ForgeRock IG then allows the response to pass through, allowing normal execution of the legacy IAM event

The JIT process developed for the toolkit provides an easy, extensible mechanism for custom legacy IAM systems. Customers or partners seeking to customize the JIT for their environment need to implement a Java API Provider interface as described below:

```
/*****************************************************************************
 *  Copyright 2019 ForgeRock AS
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *    http://www.apache.org/licenses/LICENSE-2.0
 *
 *  Unless required by applicable law or agreed to in writing, software
 *  distributed under the License is distributed on an "AS IS" BASIS,
 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *  See the License for the specific language governing permissions and
 *  limitations under the License.
 ****************************************************************************/
package org.forgerock.openig.miami;

import org.forgerock.http.protocol.Request;
import org.forgerock.http.protocol.Response;
import org.forgerock.openig.miami.common.User;

public interface LegacyIAMProvider {

       /**
        *
        * Implementation must read the user credentials from the ForgeRock HTTP
        * Request. The HTTP request gives flexibility to capture the user's credentials
        * from the request headers or from the request body. Should output a User
        * object with the intercepted username and password.
        *
        * <br>
        * <br>
        * <b>Example for getting the request body:</b>
        *
        * <code>
          request.getEntity().getString()
```

```
     *   </code>
     *
     * <br>
     * <b>Example for getting the request headers:</b>
     *
     * <code>
       request.getHeaders()
       </code>
     *
     * @param request - ForgeRock HTTP {@link Request}
     *
     * @return {@link User} - An user object with userName and userPassword set.
     * @throws Exception - in case of any error
     *
     */
    User getUserCredentials(Request request) throws Exception;

    /**
     *
     * Get user profile attributes from the legacy IAM, with userName as input.
     *
     * @param response - ForgeRock HTTP {@link Response}
     * @param userName - The user for which to retrieve the profile attributes
     * @return {@link User} - An user object with userName and userPassword set.
     */
    User getExtendedUserAttributes(Response response, String userName);

    /**
     *
     * Validate if the authentication response from the legacy system is
     * successfull.
     *
     * @param response - ForgeRock HTTP {@link Response}
     * @return true if the authentication is successfull, false if authentication
     *         failed
     */
    boolean validateLegacyAuthResponse(Response response);
}
```

By implementing the `LegacyAMProvider` interface, with the following methods contained in the framework, customers can easily adapt the JIT framework:

- `getUserCredentials` – pulls the username and password from an incoming request; adapt this for scenarios in which credentials are sent via headers, parameters or JSON

- `getExtendedUserAttributes` – retrieves the user profile from legacy IAM based on the legacy IAM token;used to enrich the JIT object
- `validateLegacyAuthResponse` – determines if the call to legacy IAM ended with successful authentication; the credentials were valid, which should trigger the JIT

3.2.2 Just In Time Provisioning using ForgeRock Identity Gateway

ForgeRock IG orchestrates existing legacy IAM authentication with IDM-based user provisioning activities. This helps customers accelerate the migration process without degrading user experience.

ForgeRock Identity Gateway is non-intrusive and can be deployed in front of existing legacy access gateways or reverse proxy.

3.2.3 SSO using ForgeRock Identity Gateway

ForgeRock Identity Gateway has unique capabilities for both reverse proxy, including transparent proxy, and access management-enabled functions. IG is the ideal module in the architecture to perform complex tasks such as SSO initiation between legacy IAM and ForgeRock IAM.

3.2.4 Migration Accelerators Package

The following high-level configuration of modules and extensions are included in the ForgeRock reference architecture:

System	Type	Name	Description
IG	IG Route	migration-assets-authentication-route.json	IG route that intercepts the legacy IAM authentication requests and provisions the user.
IG	Custom Filter	openig-miami-filters-1.0-SNAPSHOT.jar	Custom IG filters that are used in the migration authentication route

The following IG filters are combined to implement the authentication rules migration accelerators:

Filter Name	Filter Type	Description
MigrationSsoFilter	Custom	Used on Request and Response. The filter does the following actions: ● Intercepts the user' credentials from the authentication request by calling the framework method implementation **getUserCredentials**. ● Verifies if the user is migrated in ForgeRock IDM ○ If the user is **migrated**: he is authenticated in ForgeRock AM; the request is passed through to the legacy IAM and the user is authenticated there also; when legacy IAM responds, the user will have on the HTTP response a Set-Cookie header representing the legacy SSO token. The filter also adds a Set-Cookie header with the value of the SSO token obtained after authentication to ForgeRock AM. As a result, the user will have in his browser two tokens, one for the legacy IAM, and one for the ForgeRock AM. ○ If the user is **not migrated**: the filter allows the request to pass directly to legacy IAM to validate the credentials; on the response from legacy IAM, the filter verifies if the authentication succeeded by calling the framework method implementation **validateLegacyAuthResponse**. On successful authentication in legacy IAM, the filter attempts to retrieve the user profile details by calling the framework method implementation **getExtendedUserAttributes**. With the user profile attributes retrieved, the filter provisions the user in ForgeRock IDM.

The following IG route definitions implement the filter and handler chains:

	Route Execution (top to bottom on request, bottom up on response)

Route Name	
migration-assets-authenticate.json	HeaderFilter-ChangeHostFilter ClientCredentialsOAuth2ClientFilter MigrationSsoFilter ChangeLocationFilter ReverseProxyHandler

3.3 Solution Design

3.3.1 Technical Design JIT provisioning

The following sequence diagram depicts the sequence of execution of the JIT provisioning solution between legacy IAM and ForgeRock IAM:

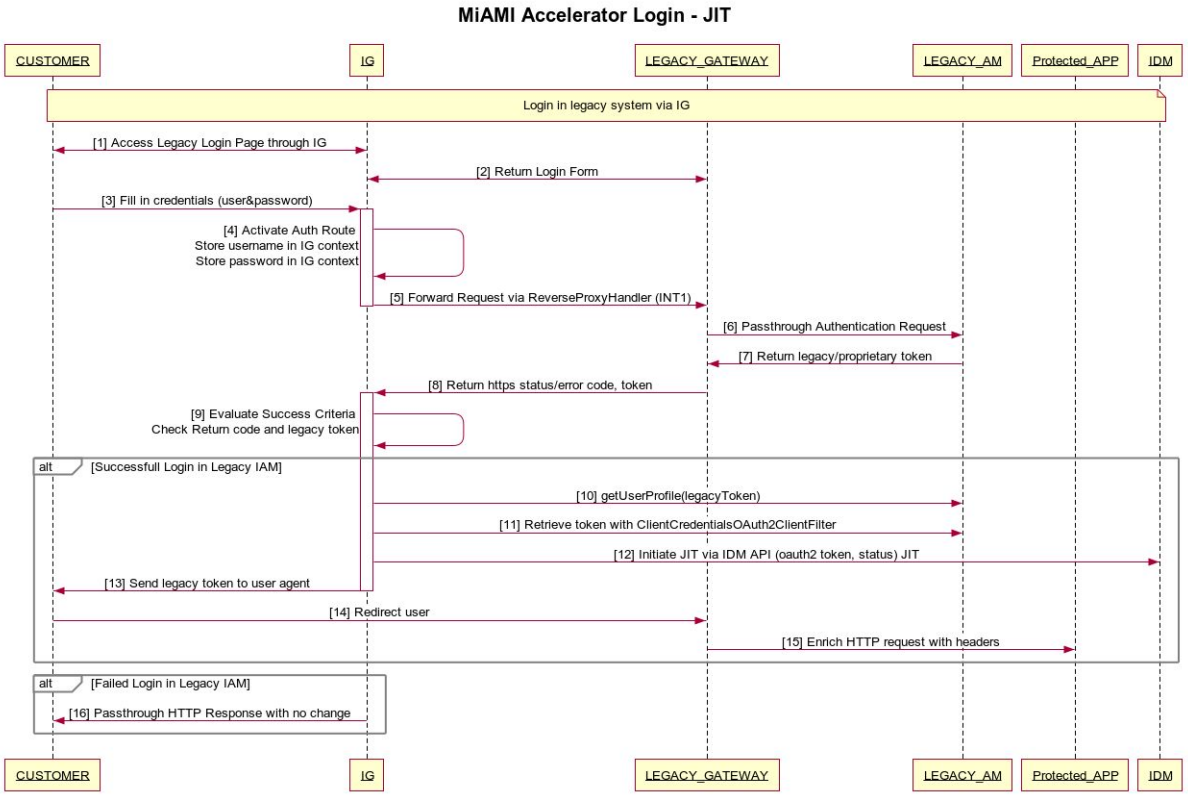

Figure 4 – JIT orchestrated in IG

Provisioning

1. The customer tries to access the protected resource
2. The legacy IAM agent redirects the user to the standard legacy login page
3. ForgeRock IG intercepts the requests and routes the login request via the transparent proxy
4. The customer enters their credentials, username and password; the user experience is identical
5. ForgeRock Identity Gateway intercepts the requests and stores the username and password in the HTTP request context in memory
6. ForgeRock Identity Gateway allows the request to continue to the backend legacy IAM

7. Upon receiving the HTTP response, ForgeRock IG intercepts the status code and response elements and evaluates if the authentication transaction in the legacy system was successful
    - If the authentication transaction was successful, JIT provisioning is triggered using the ForgeRock IDM API; username, password are retrieved from the request context
    - If the authentication transaction was not successful, JIT is not triggered
8. The response is allowed to pass through back to the user agent
9. The user is allowed access to the resource based on the legacy token existing in the user agent

3.3.2 SSO orchestrated by ForgeRock Identity Gateway

The following sequence diagram depicts the ForgeRock reference architecture solution for IG-based SSO between legacy IAM and ForgeRock IAM:

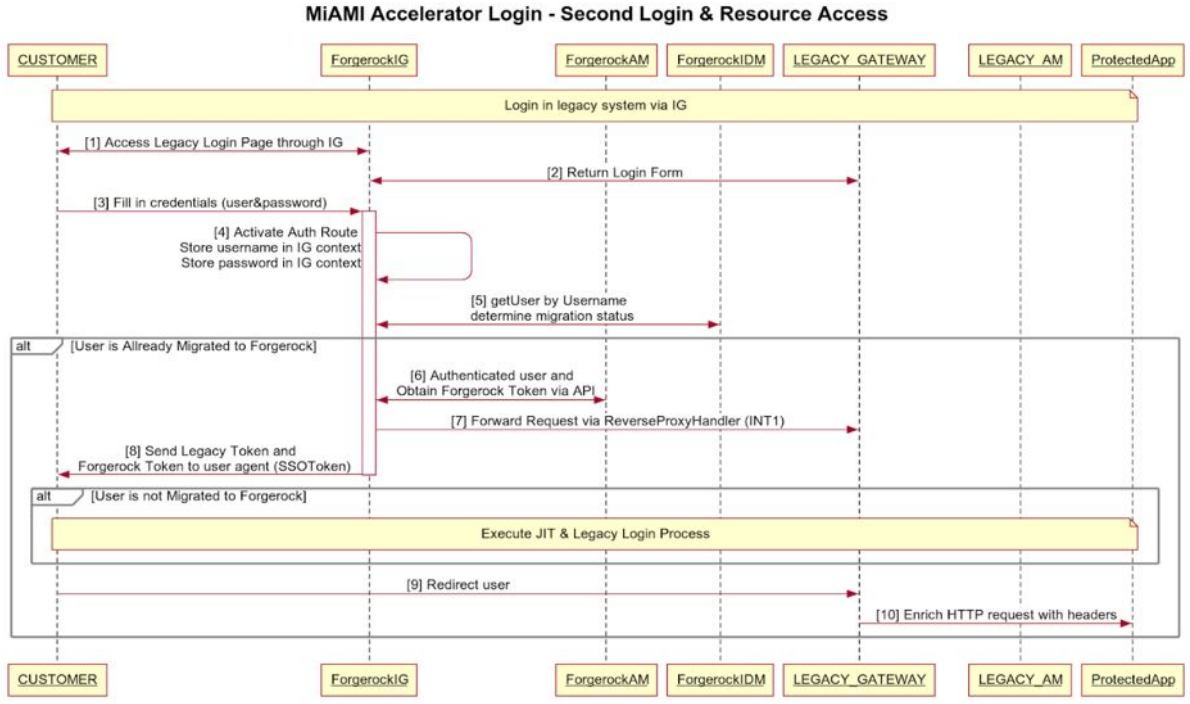

Figure 5 – SSO orchestrated in IG

SSO

1. The customer tries to access the protected resource

2. The legacy IAM Agent redirects the user to the standard login page in the legacy IAM system
3. ForgeRock IG intercepts the requests, and routes the login request using the transparent proxy
4. The customer enters their credentials, username and password; the user experience is the same
5. ForgeRock Identity Gateway intercepts the requests and stores the username and password in the HTTP request context in memory
6. ForgeRock Identity Gateway searches the IDM repository for an existing identity mapped to the associated username.
7. ForgeRock Identity Gateway allows the request to continue to the backend legacy IAM
8. On the HTTP response, ForgeRock IG intercepts the status code and response elements. It then evaluates if the authentication transaction in the legacy IAM system was successful.
    - If authentication was successful, the ForgeRock authentication flow is triggered from IG using the ForgeRock authentication treeAPI based on the username and password retrieved from the request context. The ForgeRock IAM SSO token is added to the HTTP response.
    - If authentication was not successful, ForgeRock IAM authentication is not attempted
9. The response is allowed to pass through back to the user agent
10. At the end of the sequence, in the case of a successful authentication event, both the legacy IAM system's token and the ForgeRock IAM SSO token are present in the user agent, thus allowing SSO to be achieved
11. With the legacy IAM system's token and ForgeRock's token present in the user agent, the user can access resources protected by either the legacy IAM system or by ForgeROck Access Management.

4 ForgeRock AM-Based SSO & JIT Toolkit

4.1 Target Customer Deployment

The ForgeRock AM-based SSO & JIT Toolkit accelerates migration activities in customer scenarios with one or more of the following requirements:

Topic	Answer
Bi-directional SSO between legacy IAM and ForgeRock IAM is a requirement.	YES
User passwords are stored with custom hashing in the legacy IAM user store, making migration using bulk export impossible.	YES
User passwords can be exported and imported between legacy AM and ForgeRock AM in LDIF format.	NO
Applications involved in SSO should be migrated in phases between legacy IAM to ForgeRock IAM.	YES
Legacy IAM modules or configuration can be modified or changed to integrate using SSO with ForgeRock IAM.	NO
Existing user sessions in legacy IAM need to be preserved in SSO.	YES
Partial user profile migration (no password migration) has been performed.	YES

4.2 High Level Reference Architecture

The reference architecture for the AM-based SSO Toolkit consists of:

- ForgeRock Access Manager
- ForgeRock Directory Server
- ForgeRock Identity Manager

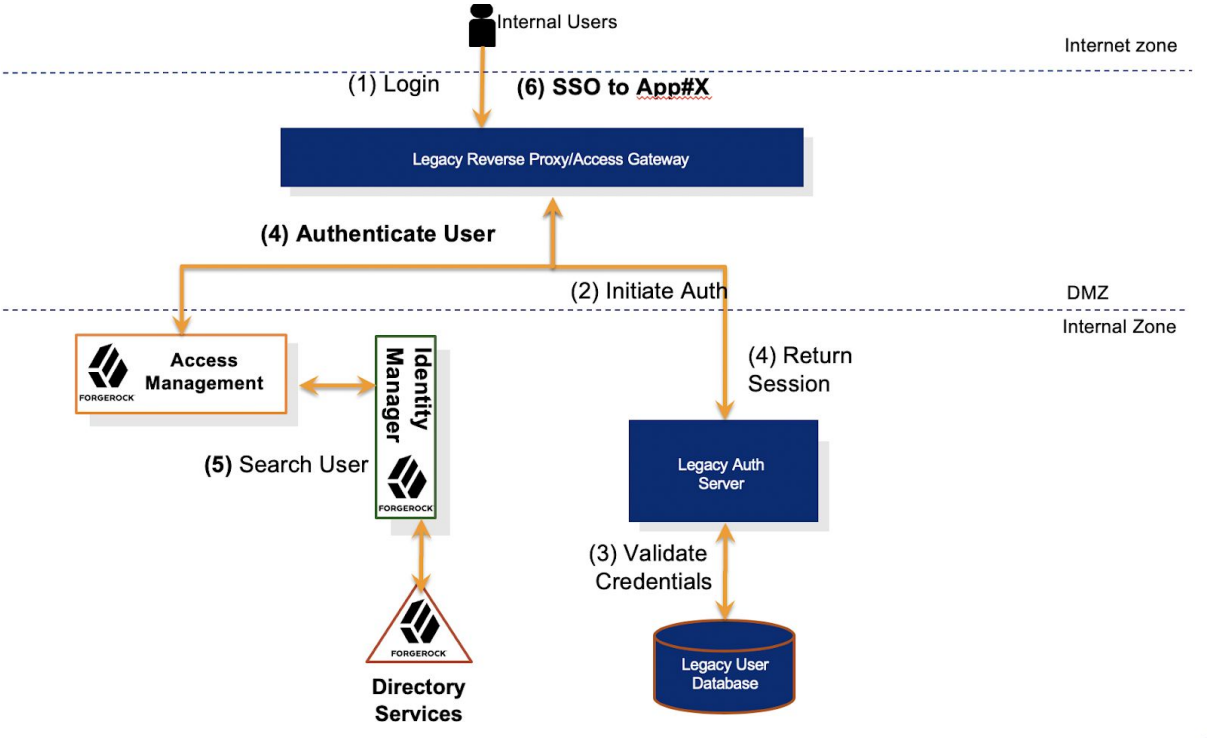

Figure 5 - AM based SSO Framework

4.2.1 Scope Definition

The toolkit provides a collection of custom nodes and a migration tree that can handle very complex migration scenarios, including bi-directional SSO between legacy IAM and ForgeRock AM.

The framework can be easily extended to support migration from any legacy IAM platform that expose client SDKs/APIs for operations such as:

- Validating existing legacy IAM tokens
- Authentication API, with username and password as input

4.2.2 Extensible Framework for Bi-Directional SSO

Powered by ForgeRock Intelligent Authentication and the powerful capabilities of the authentication trees, the framework has built-in capabilities to detect:

- Existing session from legacy IAM sessions
- Whether users are provisioned or partially provisioned in ForgeRock Directory Server
- Whether users have already been migrated , but are missing passwords

Validation of a user-entered password in the Legacy IAM is used as a decision point that determines whether the password is ready to be provisioned in ForgeRock IAM.

The Migration authentication tree provides these capabilities. Successful authentication using this tree results in a valid ForgeRock Access Manager SSO token that enables subsequent execution of outbound SSO flows using OIDC or SAMLv2.

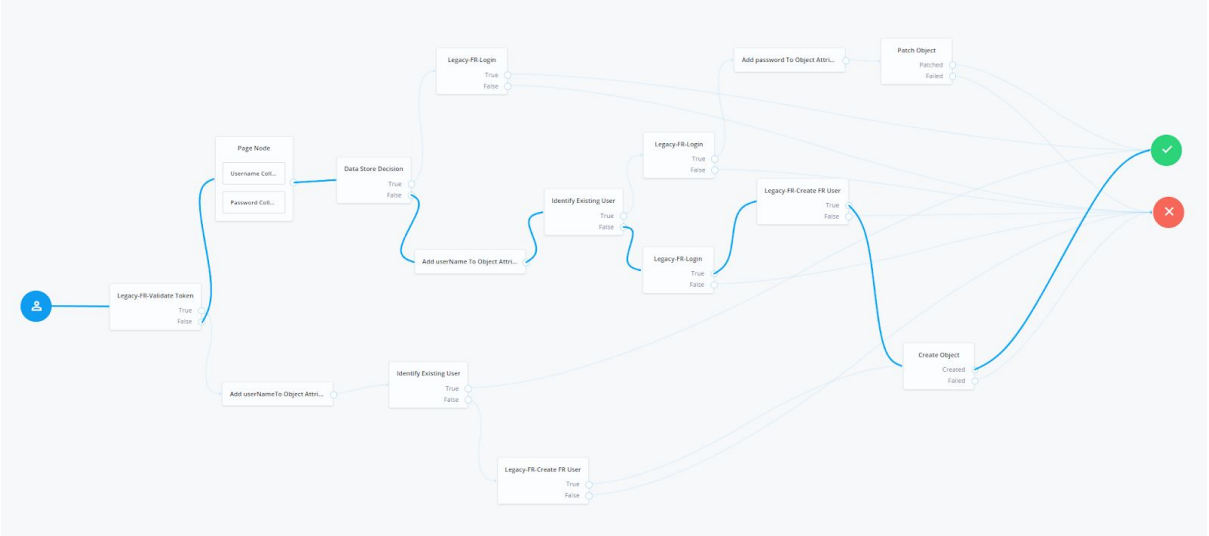

Figure 6 - Migration Authentication Tree

4.2.3 Migration Accelerators Package

The following high-level configuration of modules and extensions are included in the package:

System	Type	Name	Description
AM	Node	Legacy-FR-Validate Token	Retrieves a token from an existing cookie, validates the token against legacy IAM and provides as output in the shared state the username and outcome.
AM	Node	Legacy-FR-Create FR User	Calls the ForgeRock IDM API to provision the managed user.
AM	Node	Legacy-FR-Login	Based on the username and password from the shared state, executes the legacy IAM login API call.
AM	Node	AddAttributesToObjectAttributesNode	Constructs objectAttributes object on the Tree's shared state which will be used to call IDM or to create/patch IDM objects
AM	Tree Hook	LegacySessionTreeHook	Manages cookies if a successful login is performed into legacy IAM by the tree
AM	Authentication Tree	migrationTree	Implements the migration login and bi-directional SSO
AM	Custom Nodes	openam-modernize-auth-nodes-7.0.1.jar	Custom AM nodes that are used in the migration authentication tree.
AM	Service	LegacyFRService	Custom Service that holds the configuration service for the Legacy IAM platform

4.2.3.1 External libraries needed for (re)building the code

- n/a

4.3 Solution Design

4.3.1 Migration Authentication Tree

The execution of the Migration tree depends on the current state of the user profile:

- Existing user profile and password – no provisioning is required
- Existing user profile but no password – authentication to legacy IAM and provisioning of the user password is required
- Existing legacy SSO session – only provisioning of the user profile can be executed
- No existing user profile – upon successful authentication to legacy IAM, both the user profile and user password are provisioned and SSO is triggered

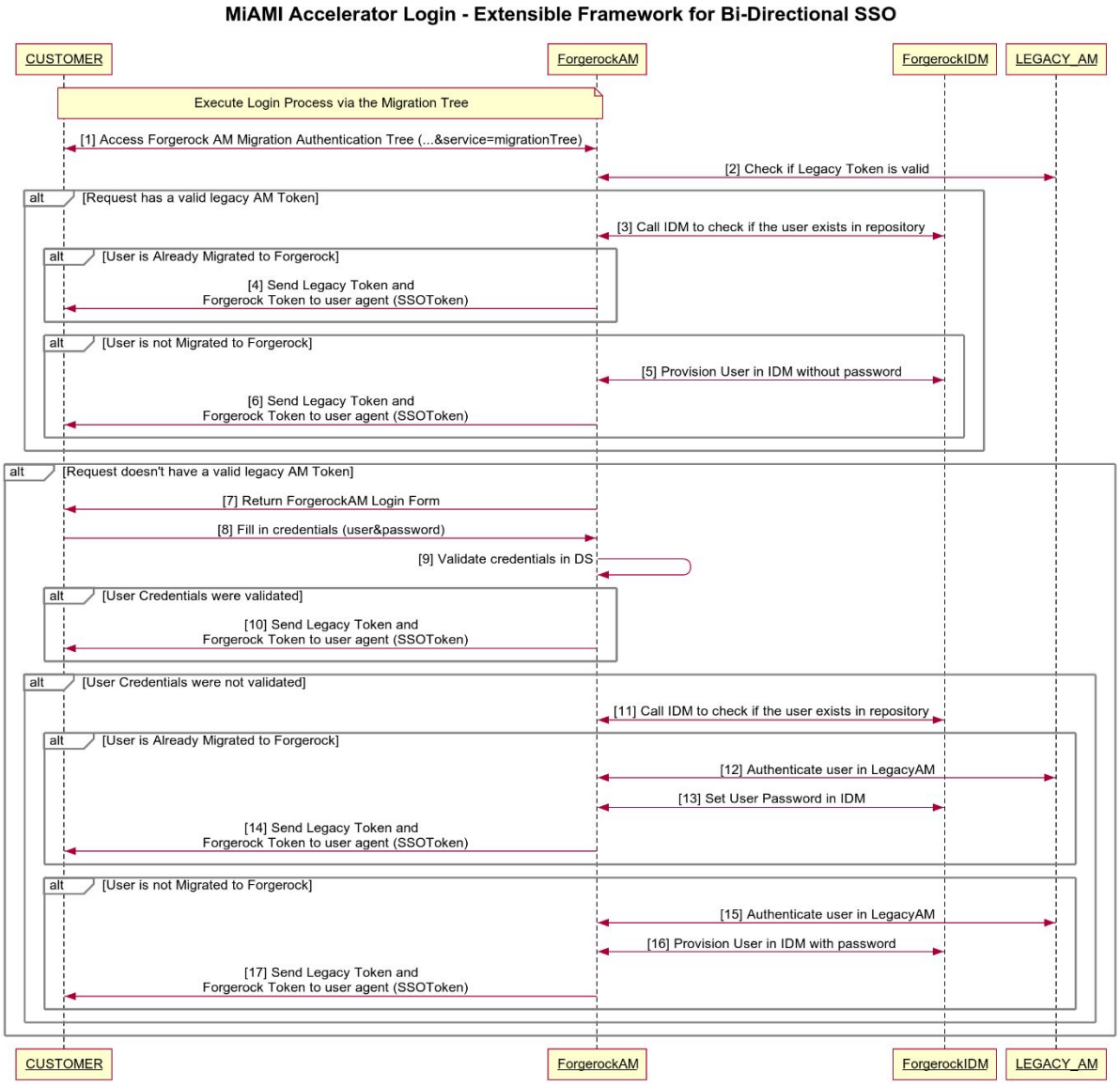

Figure 7 - Extensible Framework for Bi-Directional SSO

4.3.2 Scenarios

4.3.2.1 ***Scenario 1*** - The user has a valid legacy SSO token in the browser, and accesses the authentication tree

- The user (not previously migrated) authenticates first to the legacy IAM instance.
- The user accesses the authentication tree.

- Upon accessing the tree, the user is automatically logged in because a valid legacy SSO token is present in the browser. As a result, a user profile is created in ForgeRock IDM and AM, with no password set.

Figure 8 - Scenario1

4.3.2.2 ***Scenario 2*** - The user accesses the authentication tree, with no legacy SSO token in the browser, after previously accessing Scenario 1 - was created with no password

- The user accesses the authentication tree. The tree is prompting the user for the username and password.
- After providing credentials, the user is successfully authenticated. This happens because the user was successfully logged in to the legacy IAM system. Since the Data Store Decision node returned false but the user was already migrated, and the legacy login was successful, the password is also updated in DS.

Figure 9 - Scenario2

4.3.2.3 ***Scenario 3*** - The user is not migrated, does not have a valid legacy SSO token, and accesses the authentication tree

- The user accesses the authentication tree. The tree prompts the user for the username and password.
- After providing credentials, the user is successfully authenticated. This happens because the user was successfully logged in to the legacy IAM instance, and the user's profile was successfully provisioned in ForgeRock DS, including the password.

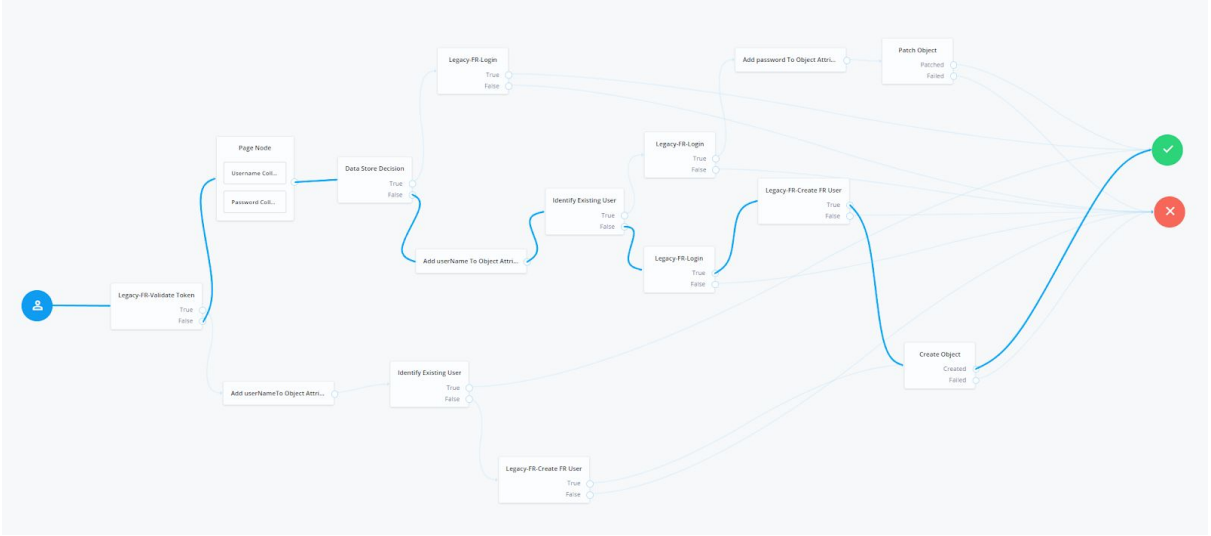

Figure 10 - Scenario3

4.3.2.4 ***Scenario 4*** - The user is already migrated, and the Data Store Decision node authenticates the user successfully

- The user accesses the authentication tree. The tree prompts the user for the username and password.
- The outcome of this scenario is that the user is authenticated automatically to both the legacy IAM instance and to ForgeRock AM after execution of the tree has completed.

Figure 11 - Scenario4

4.3.2.5 ***Scenario 5 - This scenario is triggered when the user has a valid legacy SSO token in the browser and is already migrated***

- The user (previously migrated) authenticates first to the legacy IAM.
- The user accesses the authentication tree.
- The outcome of this scenario is that the user is authenticated automatically to both the legacy IAM, and to ForgeRock AM after execution of the tree has completed.

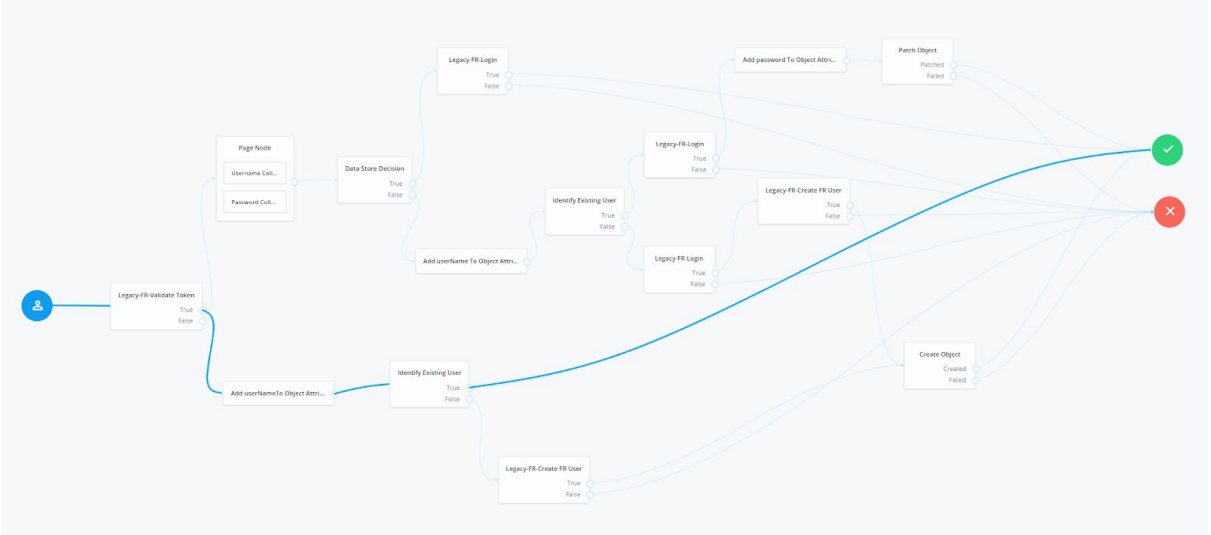

Figure 12 - Scenario5

4.3.3 Secret Stores

The passwords used in the toolkit authentication tree nodes must be saved in secret stores for security reasons.
The toolkit uses AM secret stores as described in the ForgeRock documentation.

5 ForgeRock Bulk User Migration Toolkit

5.1 Target Customer Deployment

The ForgeRock Bulk User Migration Toolkit accelerates migration activities in customer scenarios with one or more of the following requirements:

Topic	Answer
Complexity of mapping rules requirements between legacy IAM identity source and ForgeRock Directory Server as target IAM	**Medium or High**
Legacy IAM identity source type is LDAP-based; LDIF based export/import is available	**NO**
Must preview or monitor the bulk migration process	**YES**
Must schedule the migration process or provide a chunked migration	**YES**
Requiress a cut-off from legacy IAM to ForgeRock IAM.	**YES**

5.2 High Level Reference Architecture

The Bulk User Migration Toolkit reference architecture solution is based on the following ForgeRock Identity Platform core components:

- ForgeRock Directory Server
- ForgeRock Identity Manager

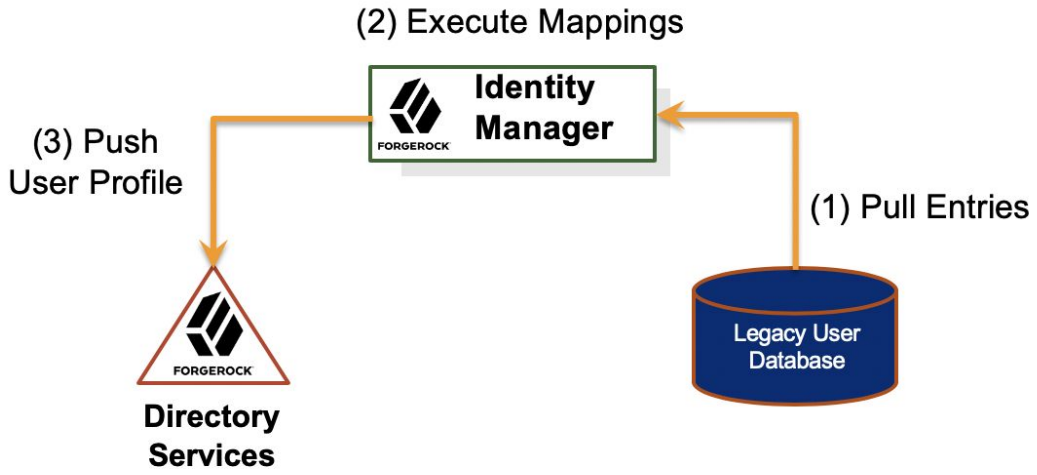

Figure 11 - Bulk Users Migration

5.2.1 Scope Definition

This toolkit implements one-way synchronization from an external legacy IAM user store to the ForgeRock IDM repository. After that, identities and groups are synchronized to ForgeRock Directory Server as the next generation user store.

The toolkit can be extended to work with any compliant source connector. User and group objects in the source system file are synchronized with the managed users and groups in the ForgeRock IDM repository, and then pushed to ForgeRock DS based on mappings that you provide.

Inbound and outbound mappings can be extended for the specific customer scenarios.

The sample source connector type is LDAPv3, but you can also use Adapter if needed.

5.2.2 Identity Management Bulk Reconciliation Process

One-time and incremental import of user profiles and groups from the legacy LDAPv3 store or another similar user store, followed by export to ForgeRock DS, is usually a requirement in the migration process.

With custom schema being used by legacy IAM systems, such as custom object classes, attributes, naming and organization units/suffix, synchronizing information can be complex to design and implement.

Mapping the extended schema, such as attributes, object classes and group membership used for core IAM transactions, can be cumbersome and lengthy.

The following assets have been included in the Migration Accelerators for this purpose:

- A template for LDAPv3 to LDAPv3 user reconciliation from legacy IAM to ForgeRock DS;
- Mapping for common group information: cn, description, uniqueMember;
- Mapping for common identity information: UID, password, common name, group membership, status, mail, telephone number, given name, last name, department details, description, employee details, last login, account locked features, number of wrong attempts

5.2.3 Bulk Migration Toolkit Package

The accelerators assets described below are delivered in a single ready-for-deployment package, making it easy for customers or partners to deploy in a pre-existing or new ForgeRock Identity Platform implementation.

The following high-level configuration of modules and extensions are included in the ForgeRock Bulk Migration Toolkit Package:

System	Type	Name	Description
IDM	Managed Object	managed.json	Enhanced user object definition that brings several other typical attributes in the IDM definition.
IDM	Managed Object	managed.json	New group managed object definition

IDM	Mapping	sync.json	Source mapping set for legacy IAM to IDM managed object (user, group)
IDM	Mapping	sync.json	Source mapping set for IDM managed object (user, group) to Forgerock Directory Server
IDM	Connector	provisioner.openicf-legacyIAM.json	Source connector that pulls user identities and groups from legacy IAM (LDAPv3 connector)
IDM	Connector	provisioner.openicf-ldap.json	Target connector that pushes identity information and groups inside ForgeRock Directory Server (LDAPv3 connector)

5.3 Solution Design

5.3.1 IDM User Managed Object Definition

The following custom fields have been added to the IDM managed object definition:

Attribute Name	Type	Description
uniqueId	String	External unique identifier
passwordSha512	String	Password SSHA512 encoded from external system
groups	String	The static groups in which the user belongs
lastFailedLogin	String	Last failed login timestamp
lastSuccessfulLogin	String	Last successful login timestamp

lockoutTime	String	Timestamp when the account was locked
loginTryCount	String	Number of invalid username/password login attempts
employeeNumber	String	For internal based IAM, the unique identifier of the employee
employeeType	String	For internal based IAM, the type of the employee
organization	String	For internal based IAM, the organization of the employee
departmentNumber	String	For internal based IAM, the unique identifier of the user department

5.3.2 IDM Group Managed Object Definition

The following new managed object and fields have been added to the IDM managed object definition:

Attribute Name	Type	Description
cn	String	Common name of the group
description	String	Description of the group
displayName	String	Display name of the group
uniqueMember	String	The static users that are members of this group

5.3.3 User mappings: Legacy IAM -> ForgeRock IDM mappings -> ForgeRock DS

Source Attribute (legacyIAM ldap)	Target Attribute (IDM)	Target Attribute (FR DS)	Description
uid	userName	uid	Unique user identifier
userPassword	passwordSha512	userPassword	External user password
cn	cn	cn	Full name
givenName	givenName	givenName	First name
inetUserStatus	accountStatus	inetUserStatus	User status
sn	sn	sn	Last name
mail	mail	mail	Email address
telephoneNumber	telephoneNumber	telephoneNumber	Telephone number
description	description	description	Description
employeeType	employeeType	employeeType	Employee type
employeeNumber	employeeNumber	employeeNumber	Employee number
o	organization	-	Organization
title	title	-	User title
displayName	displayName	-	User display name
lastFailedLogin	lastFailedLogin	sunAMAuthInvalidAttemptsData	Invalid authentication information
lastSuccessfulLogin	lastSuccessfulLogin		
lockoutTime	lockoutTime		
loginTryCount	loginTryCount		
isMemberOf	groups	isMemberOf	User groups
uniqueId	uniqueId	-	External unique identifier

5.3.4 Group mappings: Legacy IAM -> ForgeRock IDM mappings -> ForgeRock DS

Source Attribute (legacyIAM ldap)	Target Attribute (IDM)	Target Attribute (FR DS)	Description
cn	cn	cn	Common name of the group

description	description	description	Description of the group
displayName	displayName	-	Display name of the group
uniqueMember	uniqueMember	uniqueMember	The static users that are members of this group

Note: Please see Plug in Documentation for Detailed information on Plug Ins.