I

Identity Services Support Tools

Technical Design Documents : Version 3

Table of Contents

[Summary 4](#_Toc29363434)

[Revision History 4](#_Toc29363435)

[Purpose 4](#_Toc29363436)

[Intended Audience 4](#_Toc29363437)

[Team 4](#_Toc29363438)

[Overview 4](#_Toc29363439)

[Architecture and Design 6](#_Toc29363440)

[Overview 6](#_Toc29363441)

[Activity Diagram 7](#_Toc29363442)

[Deployment Diagram 8](#_Toc29363443)

[Component Diagram 9](#_Toc29363444)

[Components 9](#_Toc29363445)

[Amazon Web Services Web Application Firewall 9](#_Toc29363446)

[Amazon Web Service API Gateway 9](#_Toc29363447)

[Amazon Web Service Lambda 9](#_Toc29363448)

[Amazon Web Service S3 and CloudFront 9](#_Toc29363449)

[Amazon Web Service Cognito 10](#_Toc29363450)

[Amazon Web Service System Manager 10](#_Toc29363451)

[Okta 10](#_Toc29363452)

[Active Directory 10](#_Toc29363453)

[Business Requirements 10](#_Toc29363454)

[Business Requirements Summary 10](#_Toc29363455)

[Business Requirements Detail 11](#_Toc29363456)

[Authorization and Granular Access 11](#_Toc29363457)

[Streamlined Access 12](#_Toc29363458)

[Enable Rapid Updates 12](#_Toc29363459)

[Enable Operations Support Tools Updates 12](#_Toc29363460)

[Technical Requirements 12](#_Toc29363461)

[General Technical Requirements 12](#_Toc29363462)

[Authentication and Authorization Requirements 13](#_Toc29363463)

[Deprovisioning Requirements 14](#_Toc29363464)

[Web Application Firewall 14](#_Toc29363465)

[Identity Services API Requirements 15](#_Toc29363466)

[Client Application 16](#_Toc29363467)

[Security Requirements 16](#_Toc29363468)

[High Availability Requirements 16](#_Toc29363469)

[Monitoring Requirements 17](#_Toc29363470)

[Environments Design 17](#_Toc29363471)

[Production 17](#_Toc29363472)

[Preview 17](#_Toc29363473)

[QA 18](#_Toc29363474)

[Development 19](#_Toc29363475)

[Infrastructure/DevOps/Terraform (TODO) 19](#_Toc29363476)

[CI/CD – TO DO LATER 19](#_Toc29363477)

[OpenID Connect Provider Design (AWS Cognito) 19](#_Toc29363478)

[SAML Identity Provider Design (Okta) 19](#_Toc29363479)

[Production Configuration Information 20](#_Toc29363480)

[Preview Configuration Information 20](#_Toc29363481)

[QA Configuration Information 20](#_Toc29363482)

[Development Configuration Information 20](#_Toc29363483)

[Group Authorization Design 20](#_Toc29363484)

[Group Hierarchy 20](#_Toc29363485)

[API’s and Endpoints Design 21](#_Toc29363486)

[Endpoint Configuration Information 21](#_Toc29363487)

[Bearer Token Validation 22](#_Toc29363488)

[Activate User Endpoint 22](#_Toc29363489)

[Description 22](#_Toc29363490)

[Endpoint Details 22](#_Toc29363491)

[Business Logic 23](#_Toc29363492)

[Orchestration 23](#_Toc29363493)

[Group Assignment Endpoint 24](#_Toc29363494)

[Description 24](#_Toc29363495)

[Endpoint Details 24](#_Toc29363496)

[Business Logic 25](#_Toc29363497)

[Orchestration 25](#_Toc29363498)

[MFA Reset Endpoint 28](#_Toc29363499)

[Description 28](#_Toc29363500)

[Endpoint Details 28](#_Toc29363501)

[Business Logic 28](#_Toc29363502)

[Orchestration 28](#_Toc29363503)

[MFA Enrollment Exception (Email) Endpoint 32](#_Toc29363504)

[Description 32](#_Toc29363505)

[Business Logic 32](#_Toc29363506)

[Orchestration 32](#_Toc29363507)

[Identity Proofing 35](#_Toc29363508)

[Business Logic 35](#_Toc29363509)

[Orchestration 35](#_Toc29363510)

[Getting user by Email : 35](#_Toc29363511)

[List factors enrolled by user: 37](#_Toc29363512)

[Appendix 47](#_Toc29363513)

[WAF 48](#_Toc29363514)

[References: 48](#_Toc29363515)

[HA & DR 48](#_Toc29363516)

[OKTA API 49](#_Toc29363517)

[References: 49](#_Toc29363518)

[Architecture 49](#_Toc29363519)

[References: 49](#_Toc29363520)

# Summary

This document is a technical design document (TDD) for the development of the second version of the Identity Services Support Tools. This document provides architecture and design and developer information for the development of the application and associated services.

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision # | Revision Date | Responsible Person | Description of Change |

## Purpose

This document provides the technical and business requirements for the EA Identity Services Support tools. This document provides the requirements for the second iteration of the services which leverages an updated solution architecture and OpenID Connect in lieu of custom EA developed authentication services.

## Intended Audience

This document is intended to be used by the project and technical teams to scope out details, and for IT leadership to receive updates on the progress as well as provide feedback on additions/removals/alterations of the problem domain specifications and requirements in addition to feedback on the viability of any proposed solutions.

## Team

|  |  |  |
| --- | --- | --- |
| Team Member | Role | Contact |

# Overview

The Identity Services Support Tools provide consolidated support services for the most common Identity Services tasks aimed at simplifying the Identity Services support process. The Support Tools are aimed at providing support services for administrators that do not have and do not need specialized Identity Services administrator training. Additionally, the Identity Support tools provide an additional layer of security abstraction where EA does not need to provide Tier 1 and Tier 2 service desk support technicians access to the Okta administrative console.

The Support Tools provide the following functionality.

1. Audit a user’s account for known issues
2. Activate a user’s account
3. Assign the following groups to a user’s account (individual assignment)
   1. Okta All EA Employees
   2. Okta All EA Employees O365
4. Reset a user’s multi-factor authentication (MFA) types
5. Grant an MFA enrollment exception on an individual and mass basis
6. Ensure the tool is only functional when used on EA network (VPN...)

The primary changes in this release are:

1. Updating the application architecture to use only client-side JavaScript and the OpenID Connect (OIDC) implicit flow for authentication
2. Migrating the server-side services to API based backend as a service (BaaS)
3. Laying the framework for easier and more secure future updates, including
   1. Enable MFA authentication prompt for users when calling the service desk or site IT for assistance
   2. Enable granular access to administrative functionality within the support tools by utilizing claims-based authorization

# Architecture and Design

## Overview

The diagram below provides a high-level overview of the Identity Services Support Tools.

A screenshot of a cell phone

Description automatically generated

# Activity Diagram

C:\Documents and Settings\Administrator\Desktop\Image2.EMF

# Deployment Diagram

A close up of a map

Description automatically generated

# Component Diagram

A screenshot of a cell phone

Description automatically generated

## Components

The Identity Service Support Tools is comprised of multiple services and components. This section outlines the components that comprise their solution and provides a description of their services.

### Amazon Web Services Web Application Firewall

AWS Web Application Firewall (WAF) provides an inline security tier between clients and an application. For all net new public facing applications, a WAF is required for protection against DDOS and other types of attacks. The WAF solution will be implemented inline between the Identity Support Tools API’s and the clients.

### Amazon Web Service API Gateway

AWS’s API Gateway is a solution providing API routing, authentication, and related services. All API’s outlined in this will leverage an API Gateway for authentication (custom authorizers) and for logical group as a singular service. The API’s outlined in this solution will be Identity Support Tools specific and will be implementing a BaaS (backend as a service) architecture.

### Amazon Web Service Lambda

AWS Lambda is AWS’s function-oriented solution focused on real-time and highly scalable execution of discrete functions. AWS Lambda integrates seamlessly with API Gateway for the deployment of function-oriented API’s that can be orchestrated to provide discrete functionality.

### Amazon Web Service S3 and CloudFront

AWS S3 is Amazon’s asset storage and retrieval service which can host static assets for web applications. Amazon S3 integrates seamlessly with Amazon’s CloudFront solution, which provides global CDN capabilities. Amazon S3 and CloudFront will be leveraged to store static JavaScript, CSS, HTML and associated static assets for the Identity Services Support Tools.

### Amazon Web Service Cognito

AWS Cognito is an OpenID Connect (OIDC) authentication provider and user repository service. This service is used to store user accounts, authenticate users via OIDC to clients, manage authentication workflows, provide API’s, and provide other identity related services. AWS Cognito is used as the primary authentication service for the Identity Service Support Tools, both the API’s and the client interface.

### Amazon Web Service System Manager

The AWS System Manager provides a mechanism for storing and retrieving sensitive information, such as passwords and API keys. The AWS System Manager will be used to store the Okta API key and any AWS Cognito OIDC secrets used within this service.

### Okta

Okta is a SaaS Identity Management solution where the core product and the infrastructure is managed by Okta (<https://www.okta.com>). Okta’s Identity SaaS solution connects to EA’s on-premise Active Directory instance via Okta Active Directory “agents” installed on Windows Servers within EA’s network. These agents synchronize EA’s Active Directory information (excluding passwords) to Okta’s SaaS solution.

Okta also provides identity API’s for managing MFA, group membership, and associated identity services. The Identity Support Tools will leverage Okta’s API’s to provide administrative identity functionality.

### Active Directory

Electronic Arts Active Directory is used to authenticate users and obtain role-based access information from Active Directory groups. Oka is integrated with Active Directory for credential and Integrated Windows Authentication (IWA) authentication. Additionally, all user application roles are mapped back to an Active Directory group. The group name is passed through to the client application and used for authorization to specific API’s.

# Business Requirements

This section outlines the business cases and business requirements for the updated Identity Support Tools.

## Business Requirements Summary

The original Identity Support tools was quickly built as a stop-gap solution to address specific business and security requirements incurred as part of the Retina go-live process. As a result of the rapid response to the business need, the standard architecture and design process was not completed. As a result, the solution requires an update to ensure EA has the ability to quickly and stably deploy new features in response to business and technical needs.

Additionally, the original tool was deployed with a custom authentication solution and lacked granular authorization controls. The updates outlined in this document address the authentication and authorization issues and update the architecture and design to adhere to new Identity Services standards.

The updates outlined in this document include but are not limited to:

* Reduce the development time for deploying new solutions
* Reduce the number of authorization schemas used within EA
* Abstract the complexity of the EA Identity Services
* Reduce the number of authentication related services used by EA employees
* Enhance self-service capabilities, specifically as it pertains to multi-factor authentication
* Deploy tools for EA operations team to effectively and securely support EA employees
* Rebrand the Retina Support Tools to EA’s Identity Services Support Tools
* Utilize current versions of development software and components.
* Remove dependencies on no longer supported libraries and technologies (i.e Joystick)
* Provide contextual access to EA Identity Services Support Tools
* Update the Identity Service Support tools UI to be consistent with the EA employee branding
* Adopt the new Identity Services OIDC Implicit Flow and BaaS application architecture

## Business Requirements Detail

This section outlines the detailed business requirements from which the technical requirements are based.

* Easier access
* More granular access control to functionality
* Update solution so we can more easily iterate new development
* Operationalize solution development
* Lay the foundation for new enhancements
* Streamline the deployment pipeline
* Remove legacy UI/UIX design library: Joystick (no longer supported)

### Authorization and Granular Access

The Identity Services support tools currently have no in application authorization controls. As a result, the functionality within the tools cannot be scoped to a specific role or responsibility. The addition of OIDC and OAuth 2.0 to the EA Identity Services portfolio enables the use of OAuth 2.0 scopes to control the granular access. The integration of Identity Services with Active Directory also allows scopes to be mapped to Active Directory groups and the use of DLManager to manage the groups.

Scopes are defined and mapped in Cognito and mapped Active Directory (AD) groups and the groups can be reviewed in [DL Manager](https://dlmanager.ea.com). They follow a general naming convention: “*App Assignment - Identity Support Tools -* ***{sub-claim-name}***”, where {sub-claim-name} is the specific function within the Identity Support tools. Access to the app itself is granted with the claim: “*App Assignment - Identity Support Tools*”.

### Streamlined Access

The Identity Support Tools are publicly hosted on AWS however the Identity Support Tools API’s are hosted internally in EA’s network on Windows Servers. The access to the Identity Services tools is not streamlined and results in difficult user access. Streamlining the Identity Support tools access and enabling secure access from outside of EA’s network is required.

### Enable Rapid Updates

The Identity Support tools must allow the ability for the IT Ops EE team to rapidly respond to business and technical needs and update the solution accordingly. The update outlined in this document will allow for the rapid and secure addition of new features to the Identity Support Tools.

### Enable Operations Support Tools Updates

The existing solution is complex to update and not fully integrated with EA’s Gitlab and standard CI/CD pipeline deployment process. The updates to the Identity Services outlined in this document must enable easy and rapid update of the Identity Support Tools using standard engineering processes.

# Technical Requirements

This section outlines the technical requirements supporting the business requirements and ensuing the updates.

## General Technical Requirements

|  |  |
| --- | --- |
| Requirement Number | Client Application Requirement |
| GATR-1 | One of the primary objectives of this version of the app is to deprecate the custom authentication mechanism implemented specifically for the older version of the app with a more secure standardized/generic approach that can be reused with any app supporting OIDC. |
| GATR-2 | Users must have OIDC claims based authorization to each discrete piece of functionality within the Identity Support Tools |
| GATR-3 | All Retina text and branding must be removed from the application |
| GATR-4 | Joystick should not be used in this project |
| GATR-5 | The application must support the EA SSO user experience |
| GATR-6 | The Identity Services Support tools session must be 1 minute less than the expires at time contained within the OIDC JWT access token |
| GATR-7 | The Identity Services Support tools must automatically redirect to the IDP when the session expires. The redirect must honor all deep linking (e.g. return the user to the current location / route / page the user was originally using within the application) |
| GATR-8 | The EA Identity Service Support tools must load within 1.5 seconds maximum (only application, not authentication workflow) |

## Authentication and Authorization Requirements

All authentication must be completed via Okta, with AWS Cognito acting as the intermediate OIDC to SAML authentication broker. The Identity Services Support Tools will leverage the OIDC implicit flow for authentication and will use the OIDC JWT obtained during the authentication process, for authenticating and authorizing the user to access the Identity Service Support Tools APIs. This section outlines the authentication requirements for the Identity Service Support Tools.

|  |  |
| --- | --- |
| Requirement Number | Authentication and Authorization Requirements |
| Auth-1 | All client authentication must be completed via the OpenID Implicit flow |
| Auth-2 | Users must leverage the standard EA SSO login workflow services. These services include the use of IWA authentication on network and credential + MFA authentication off of EA’s network |
| Auth-3 | The Identity Services Support Tools must integrate directly with AWS Cognito via OIDC for authentication. All OIDC tokens must be minted by Cognito. |
| Auth-4 | The Identity Services Support Tools Cognito instance must integrate with Okta via SAML 2.0 for authentication. No authentication directly to Cognito will be allowed |
| Auth-5 | A custom Cognito trigger must be used to add the appropriate Identity Services Support Tools AD groups to the Cognito Identity Token using the post authentication trigger |
| Auth-6 | Access to the Identity Services Support Tools must be tightly controlled by specific Active Directory security groups that limit access to specific personnel |
| Auth-7 | The Identity Services Support Tools provisioning is dynamic and completed just-in-time based on Active Directory security group membership. |
| Auth-8 | Authentication to the Identity Services Support Tools must be completed with a valid EA Active Directory account |
| Auth-9 | The SAML integration between Cognito and Okta must leverage a user’s employee number as the primary SAML name identifier |
| Auth-10 | The SAML assertion generated by Okta must include the following attributes:   * Current Active Directory email address * Employee number * First name * Last name * Active Directory group memberships for all groups starting with “App Assignment - Identity Support Tools” |
| Auth-11 | AWS Cognito must be used to authenticate the Cognito user pool with Okta (IDP) via SAML 2.0 |
| Auth-12 | The AWS Cognito user pool must have a one to one mapping between the integrated client applications and the Identity Provider |

## Deprovisioning Requirements

The Identity Services Support Tools leverages just-in-time provisioning based on a user’s Active Directory group membership. As such, user accounts must be regularly deprovisioned using a regularly scheduled AWS Lambda service. This service must execute regularly and be updated and fully integrated with the Identity Services Support Tools CI/CD pipeline.

|  |  |
| --- | --- |
| Requirement Number | Authentication and Authorization Requirements |
| Deprov-1 | Deprovisioning must be executed on a regular basis for all accounts that have not access the Identity Services Support Tools in the last 30 calendar days |
| Deprov-2 | Deprovisioning must be executed via a daily scheduled AWS Lambda function that removes users from the Cognito User Pool |
| Deprov-3 | The deprovisioning service must log every account removed, the date and time the account was removed, and the unique identifier for the account being removed |

## Web Application Firewall

AWS WAF is a web application firewall that helps you protect your websites and web applications against various attack vectors at the HTTP protocol level. AWS’s WAF solution will be deployed via Terraform to provide additional protection between clients and the Identity Support Tool’s API’s. The WAF rules will include WAF protection for the top 10 OWASP application vulnerabilities. It supports rate-based rules which mitigate XSS attacks, one can use a cross-site scripting match condition to deploy rules to mitigate these attacks, AWS WAF filters dangerous HTTP request patterns that can indicate path traversal attempts, or remote and local file inclusion (RFI/LFI). A key advantage of AWS WAF is its programmability. You can configure and modify AWS WAF web access control lists (ACLs), rules, and conditions by using a programmatic API at any time. Any change normally take effect within a minute, even for our global service that’s integrated with Amazon CloudFront.

|  |  |
| --- | --- |
| Requirement Number | WAF Requirements |
| WAF-1 | Protect against SQL Injection attacks |
| WAF-2 | Protect against broken authentication and session management attacks |
| WAF-3 | Protect against cross-site scripting attacks |
| WAF-4 | Protect against broken access control attacks |
| WAF-5 | Protect against common security misconfigurations |
| WAF-6 | Protect against sensitive data exposure |
| WAF-7 | Protect against insufficient attack protection |
| WAF-8 | Protect against using components with known vulnerabilities |
| WAF-9 | Protect against cross-site forgery requests |
| WAF-10 | Protect against unprotected APIs |

## Identity Services API Requirements

This section outlines general requirements for the Identity Services Support Tools API’s. As the Identity Support Tools leverages a client application and BaaS architecture, the OIDC JWT used for authentication is leveraged for authentication and authorization to Identity Service Support Tools API’s to obtain data and execute actions.

|  |  |
| --- | --- |
| Requirement Number | API Requirements |
| API-1 | The Identity Services Support Tools must leverage EA’s Identity Services Support Tools API’s for all execution logic |
| API-2 | The Identity Services Support Tools may have API services orchestration which orchestrates the EA Identity Services API’s. The Identity Services Support Tools API orchestration must leverage OIDC implicit code flow for API authentication and authorization |
| API-3 | Authentication and authorization to all Identity Services and related API’s leveraged by this solution must be completed via an OIDC identity token. |
| API-4 | Token should be passed along to backend services endpoints within the Bearer token header |
| API-5 | All API’s must leverage OIDC identity or access tokens for authorization |
| API-6 | All API’s must be configured to leverage a set scope mapped to an AD group for authorization |
| API-7 | All API’s must validate the token in real time |
| API-8 | Token validation endpoints must be whitelisted within the API configuration |
| API-9 | All API configurations must be obtained from an AWS DynamoDB global tables instance |
| API-10 | All API keys and other credentials leveraged by the Identity Support Tools APIs must be stored in AWS KMS. Under no circumstances should any secrets be stored in the Identity Services Support Tools source code |
| API-11 | The Identity Services Support Tools APIs will leverage AWS API Gateway, AWS Lambda, and AWS DynamoDB Global tables |
| API-12 | The Identity Services Support Tools must be deployed to at least 2 AWS regions. Preferably deployed to the AWS Oregon and AWS Virginia regions. |
| API-13 | All Identity Services Support Tools API’s must leverage a single hostname regardless of the region where the support tools are deployed to. |

## Client Application

The Identity Services Support Tools user interface is a client-side JavaScript application built using the [StencilJS](https://stenciljs.com/) component compiler. The generated Web Components are coded with Typescript and JSX. The application and all assets are hosted on AWS’s S3 and CloudFront for low-latency, global high availability, and reduced operational overhead. This section outlines the requirements for the client-side application. Note, this client will be protected by WAF.

|  |  |
| --- | --- |
| Requirement Number | Client Application Requirements |
| App-1 | The Identity Services Support Tools must leverage Stencil as the basis for the user interface Web Components |
| App-2 | All assets must be stored in AWS S3 and be served from a S3 bucket |
| App-3 | CloudFront or another content proxy will be used to serve and cache all static resources (**Note**: There appears to be an issue with CloudFront being a proxy for SPAs hosted on S3 buckets) |
| App-4 | The user interface must leverage all DWS font, styling, and UI/UX frameworks. Because the UI is very minimalist, no external CSS libraries will be used. However, specific CSS class implementations will be “borrowed” from the DWS project. |
| App-5 |  |

## Security Requirements

Note: Will be finalized in V3.2

|  |  |
| --- | --- |
| Requirement Number | Security Requirements |
| Sec-1 | SSL with forward secrecy must be used in the communication between all services |
| Sec-2 | No secrets can be stored in source code |
| Sec-3 | All authorization and access controls must leverage AD |
| Sec-3 | Single sign-on (SSO) is required to access the Identity Services Support Tools |
| Sec-4 | All access must be Active Directory group based |
| Sec-5 | Only authorized EA employees and contractors are authorized to use this application |

## High Availability Requirements

Note: Will be finalized in V3.2

|  |  |
| --- | --- |
| Requirement Number | High Availability Requirements |
| HA-1 | All services must be highly available and deployed to at least two physically separate locations |
| HA-2 | All services must dynamically shift from a datacenter or region experiencing an outage to regions not experiencing an outage |
| HA-3 | All regions must be configured in an active-active load balancing configuration driven by latency-based DNS routing |
| HA-4 | All involved components such as WAF, CloudFront, API GW, Lambda, Cognito, Cloudwatch, Okta etc.., must have monthly uptime percentage of at least 99.0% |

## Monitoring Requirements

Note: Will be finalized in V3.2

|  |  |
| --- | --- |
| Requirement Number | Security Requirements |
| M-1 | Each Identity Services Support Tools API must be monitored individually |
| M-2 | The Identity Services Support Tools application must be monitored via synthetic monitoring and alerts raised after 3 unsuccessful synthetic monitoring authentication attempts |
| M-3 | Identity Services Support Tools monitoring must be incorporated into the Identity Support Services |

# Environments Design

This section outlines the environments configuration and design. Multiple environments exist, including production, preview, QA, and development.

## Production

The production environment details are listed below.

|  |  |
| --- | --- |
| Key | Value |
| Hostname | tools.work.ea.com |
| SSL cert required | True |
| Available off network | False |
| Gitlab Repo Branch | Master |
| AWS Environment | Itett-atlas-admin |
| Okta Environment | Ea.okta.com |
| Logging Environment | The app will support logging to the Console. By default, there will not be any logging to the console. In future versions, we may evaluate the option to log to the cloud.  Backend APIs will log to AWS CloudWatch. |

## Preview

|  |  |
| --- | --- |
| Key | Value |
| Hostname | tools**-staging**.work.ea.com |
| SSL cert required | True |
| Available off network | False |
| Gitlab Repo Branch | Preview |
| AWS Environment | Itett-atlas-admin |
| Okta Environment | Ea.oktapreview.com |
| Logging Environment | The app will support logging to the Console. By default, there will not be any logging to the console. In future versions, we may evaluate the option to log to the cloud.  Backend APIs will log to AWS CloudWatch. |
| APP URL | Please keep adding all the relevant and required parameters needed such as db endpoints, api gw endpoints, dns record sets, cognito urls, lamdas, |
| APP URL | Application URL |
| Cognito URL | Call back URL (SAML Endpoint) |
| AWS WAF | WAF Rules |
| API Gateway | Api Enpoints(URL) |
| AWS Lambda | Lambda Function Endpoints which respectively connected to API endpoints |
| S3 | S3 Obect Endpoint for Landing page |
| CloudFront | Cloudfront endpoint associated with WAF |
| Active Directory | EA- Active Directory Server Endpoint |

## QA

|  |  |
| --- | --- |
| Key | Value |
| Hostname | tools**-qa**.work.ea.com |
| SSL cert required | True |
| Available off network | False |
| Gitlab Repo Branch | QA |
| AWS Environment | Itett-atlas-preprod-admin |
| Okta Environment | Ea.oktapreview.com |
| Logging Environment | The app will support logging to the Console. By default, there will not be any logging to the console. In future versions, we may evaluate the option to log to the cloud.  Backend APIs will log to AWS CloudWatch. |
| APP URL | Application URL |
| Cognito URL | Call back URL (SAML Endpoint) |
| AWS WAF | WAF Rules |
| API Gateway | Api Enpoints(URL) |
| AWS Lambda | Lambda Function Endpoints which respectively connected to API endpoints |
| S3 | S3 Obect Endpoint for Landing page |
| CloudFront | Cloudfront endpoint associated with WAF |
| Active Directory | EA- Active Directory Server Endpoint |

## Development

|  |  |
| --- | --- |
| Key | Value |
| Hostname | tools-dev.work.ea.com |
| SSL cert required | True |
| Available off network | False |
| Gitlab Repo Branch | Develop |
| AWS Environment | Itett-atlas-preprod-admin |
| Okta Environment | Ea.oktapreview.com |
| Logging Environment | The app will support logging to the Console. By default, there will not be any logging to the console. In future versions, we may evaluate the option to log to the cloud.  Backend APIs will log to AWS CloudWatch. |
| APP URL | Application URL |
| Cognito URL | Call back URL (SAML Endpoint) |
| AWS WAF | WAF Rules |
| API Gateway | Api Enpoints(URL) |
| AWS Lambda | Lambda Function Endpoints which respectively connected to API endpoints |
| S3 | S3 Obect Endpoint for Landing page |
| CloudFront | Cloudfront endpoint associated with WAF |
| Active Directory | EA- Active Directory Server Endpoint |

# Infrastructure/DevOps/Terraform (TODO)

## CI/CD – TO DO LATER

Infra pipeline this is more about how we use terraform/cloud formation

Code pipeline, what all repos we are using and how they get build on changes

We need to think more from stack perspective whether all the components are resilient and HA across multi regions.

Ideally we need to be having a CI/CD pipeline to spin up the whole stack on a click of a button

# OpenID Connect Provider Design (AWS Cognito)

Information to be added in V3.2

# SAML Identity Provider Design (Okta)

AWS Cognito integrates with Okta via SAML 2.0 for seamlessly translating the authentication from OIDC to SAML 2.0 and ensuring the integrity of the Identity Services SSO experience. Okta’s OIDC application integration does not natively support EA’s custom user interface in the configuration EA has deployed. Integrating AWS Cognito with Okta via SAML 2.0, ensures the integrity of the user experience and security of SSO.

The sections below outline the configuration information for integrating AWS Cognito with Okta via SAML 2.0, in support of the Identity Services Support Tools.

## Production Configuration Information

|  |  |
| --- | --- |
| Key | Value |
| Okta Application Name |  |
| Okta Application ACS URL |  |
| Okta Application SAML Cert |  |
| Base64 Encoded Okta SAML Metadata |  |
| AWS Cognito SP URL |  |
| Base64 Encoded AWS Cognito SAML 2.0 Metadata |  |

## Preview Configuration Information

|  |  |
| --- | --- |
| Key | Value |
| Okta Application Name |  |
| Okta Application ACS URL |  |
| Okta Application SAML Cert |  |
| Base64 Encoded Okta SAML Metadata |  |
| AWS Cognito SP URL |  |
| Base64 Encoded AWS Cognito SAML 2.0 Metadata |  |

## QA Configuration Information

## Development Configuration Information

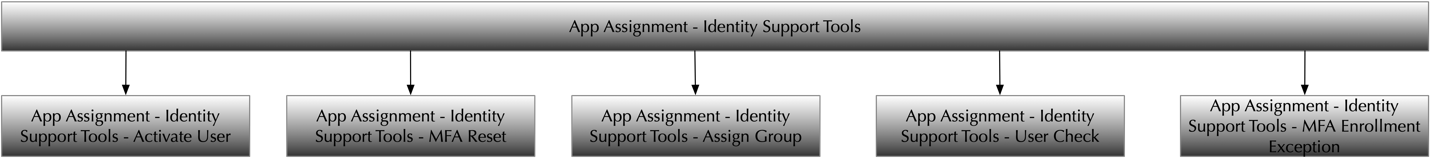
Information to be added in V3.2

# Group Authorization Design

Active Directory groups are used to display specific in application functions and authorize users to execute these functions within the API services. The groups are nested to allow for separate authentication and authorization. The client application must only display the functional component outlined in the “group” claim within the OIDC identity token.

## Group Hierarchy

The parent Active Directory group (App Assignment – Identity Support Tools) must have each of the corresponding groups as members. Each subgroup is used for in application authorizations. A user can be a member of more than one of these groups (and typically will be). Each group corresponds to an in-application function that a user is authorized for. Okta only requires the parent group of “App Assignment – Identity Support Tools” to be assigned to it for identity provider authorization. All new groups that are created for future in-application functionality must follow this authorization model.



|  |  |
| --- | --- |
| Group Name | Relationship |
| App Assignment – Identity Support Tools | Parent group |
| App Assignment – Identity Support Tools – Activate User | Child of App Assignment – Identity Support Tools |
| App Assignment – Identity Support Tools – MFA Reset | Child of App Assignment – Identity Support Tools |
| App Assignment – Identity Support Tools – Assign Group | Child of App Assignment – Identity Support Tools |
| App Assignment – Identity Support Tools – User Check | Child of App Assignment – Identity Support Tools |
| App Assignment – Identity Support Tools – MFA Enrollment Exception | Child of App Assignment – Identity Support Tools |
| App Assignment – Identity Support Tools – Verify Identity | Child of App Assignment – Identity Support Tools |

# API’s and Endpoints Design

All API’s and endpoints outlined in this section must be highly available and be deployed to AWS Lambda, AWS API Gateway, and leverage Route53 DNS. Additionally, all endpoints outlined in this section must leverage OIDC for authentication and authorization and may not leverage static API keys for access.

## Endpoint Configuration Information

|  |  |
| --- | --- |
| Key | Value |
| okta\_domain | Options include ea.okta.com and ea.oktapreview.com |
| okta\_apikey | Must be stored and obtained from AWS KMS |
| cognito\_region | Obtained during the CI/CD pipeline build process and will need to dynamically be updated. |
| cognito\_instanceID | Obtained during the CI/CD pipeline build process and will need to dynamically be updated. |
| environment | Possible values include production, preview, QA, and develop |
|  |  |

## Bearer Token Validation

The Bearer tokens used in each API call must be validated in real time against AWS Cognito. The issuer claim in the token must match the issuer in the configuration file/service and will be used to construct the OIDC metadata URL. A real time API call must be made to this service to obtain the JWKS keys and validate the token.

The logic for the Bearer token validation is as follows.

|  |  |
| --- | --- |
| Step | Value |
|  | Obtain the bearer token from the HTTP header |
|  | Split the bearer token into the header, payload, and signature |
|  | Base64 decode the header, payload, and signature |
|  | Obtain the “iss” claim value from the base64 decoded payload |
|  | String match the “iss” claim from the JWT against the issuer configuration variable from the service configuration |
|  | Obtain the “kid” claim value from the base64 decoded header |
|  | Obtain the “jwks\_uri” claim value from the base64 decoded payload |
|  | Request the JSON payload from the “jwks\_uri” claim value URL. The resulting payload will be referred to as the JWKS payload |
|  | Iterate through the JWKS payload, looking for a “kid” match from the original bearer token in the JWKS payload. If a match is found, use the array (“key”) from the JWKS payload in which the matching key was found. |
|  | Validate the bearer token using the “key” obtained in step 9 |
|  | If the bearer token cannot be validated against the matching key obtained from the JWKS\_URI, the bearer token and resulting API call must be rejected. If the bearer token can be validated, continue to step 12 |
|  | Obtain the “group” claim value (array) from the base64 decoded bearer token body |

|  |  |
| --- | --- |
| Configuration | Value |
| Issuer Format | https://cognito-idp.{{region}}.amazonaws.com/{{instanceID}} |

## Activate User Endpoint

Description: This endpoint is responsible for activating a user in Okta.

### Endpoint Details

|  |  |
| --- | --- |
| Configuration | Configuration Value |
| Endpoint | /v1/users/${user-email}/fix/activate |
| Headers | Bearer: ${token} |
| Method | GET |
| AuthZ Claim | App Assignment - Identity Support Tools - Activate User |
| Group Name | App Assignment - Identity Support Tools - Activate User |
| Request Body | None |
| HTTP Response: Success | HTTP 200 |
| HTTP Response: Failure | HTTP 404 |
| Payload Output: Success | User is already active in Okta:  {  "data": {  "email": "oktatest161@ea.com",  "summary": ["Okta Account [oktatest161@ea.com] is already in active state"]  }  }  Invalid user email address:  {  "data": {  "email": "oktatest161aaa@ea.com",  "summary": [  "Okta Account with primary email [oktatest161aaa@ea.com] not found"  ]  }  } |
| Logging | Required. Single AWS Cloudwatch group.  /aws/lambda/identity-services-support-tools-<<environment name>> |

### Business Logic

No business logic is implemented in this API endpoint. This endpoint purely abstracts the Okta Activate user endpoint and allows users to be activated without Okta confirmation emails being sent.

### Orchestration

This Identity Services Support Tools function looks up the user by their email address, returns the Okta userId, then leverages the Okta userId in the Okta Activate user API call. The follow Okta API’s are being leveraged in the order defined below.

|  |  |
| --- | --- |
| Configuration | Configuration Value |
| Execution | First |
| Input | Email address from client application |
| Output Value | Okta userId from JSON payload |
| API URL | {{Okta Org}}/api/v1/users/{{email}} |
| HTTP Method | GET |
| Input Headers | 1. Authorization: SSWS {{Okta API Key}} 2. Content-Type: application/json 3. Accept: application/json |
| Reference | <https://developer.okta.com/docs/reference/api/users/#get-user> |

|  |  |
| --- | --- |
| Configuration | Configuration Value |
| Execution | Second |
| Input | userId from the GET User API call |
| Output | HTTP 200 indicating success |
| API URL | {{Okta Org}}/api/v1/users/{{userId}}/lifecycle/activate?sendEmail=false  NOTE: It is imperative that the sendEmail=False flag be set otherwise users will be sent emails directly from Okta and this is *not* supported or approved |
| HTTP Method | POST |
| Input Headers | 1. Authorization: SSWS {{Okta API Key}} 2. Content-Type: application/json 3. Accept: application/json |
| Body | Empty |
| Reference | <https://developer.okta.com/docs/reference/api/users/#activate-user> |

## Group Assignment Endpoint

Description: This endpoint is responsible for assigning a group membership to a user given their email address. There are two group membership supported at this time: [Okta All EA Employees] and [Okta All EA Employees Office 365]. The two groups we are adding users to should be stored and referred to from a configuration table. The solution must be easy to update with new groups. Ultimately, this configuration will be shifted to a DynamoDB table and we will create administrative API’s to update and manage these groups.

### Endpoint Details

|  |  |
| --- | --- |
| Configuration | Configuration Value |
| Endpoint | /v1/users/${user-email}/fix/assign/${group-name} |
| Headers | Bearer: ${token} |
| Method | POST |
| AuthZ Claim | App Assignment - Identity Support Tools |
| Group | App Assignment - Identity Support Tools |
| Request Body | None |
| HTTP Response: Success | HTTP 204 |
| HTTP Response: Failure | HTTP 404 |
| Response Payload: Success | {  "data": {  "email": "oktatest161@ea.com",  "summary": ["Successfully assigned Okta User [oktatest161@ea.com] to Okta Group [Okta\_All\_EA\_Employees]"]  }  } |
| Logging | Required. Single AWS Cloudwatch group.  /aws/lambda/identity-services-support-tools-<<environment name>> |

### Business Logic

1. Lookup the users Okta userID using the Okta GET users API call
2. Lookup the group name passed in by the API call against the local configuration / mapping table
3. If the group name is not found in the local configuration / mapping table, return an error
4. If the group name matches a group name in the local configuration / mapping table, call the Okta add user to group API call passing in the user’s Okta userId and the Okta groupId from the local mapping table
5. Return confirmation of success or failure

### Orchestration

The section below outlines the underlying service orchestration that is required to successfully execute this endpoint. This function requires that a client’s Okta userId be obtained from Okta and then a subsequent API call is executed to add the user to the appropriate Okta group. The Identity Services Support Tools client application will provide the clients email address and group name to the Identity Services Support Tools API. The Identity Services Support Tools API will look up the group name against a configuration file (will be migrated to a configuration database in a future release), return the corresponding Okta groupId (or error out if the group name does not match a stored group name value), then add the user to the corresponding Okta group.

|  |  |
| --- | --- |
| Configuration | Value |
| Priority | First |
| Input | Email address from client application |
| Output | Okta user profile payload from which the userId must be extracted. HTTP 200 header (success) |
| API URL | {{Okta Org}}/api/v1/users/{{email}} |
| HTTP Method | GET |
| Input Headers | 1. Authorization: SSWS {{Okta API Key}} 2. Content-Type: application/json 3. Accept: application/json |
| Request Body | None |
| Response Payload | {  "id": "00u5h8rhqv5Ne4CmB0h7",  "status": "ACTIVE",  "created": "2015-12-08T11:03:45.000Z",  "activated": "2015-12-08T11:03:46.000Z",  "statusChanged": "2019-09-24T12:15:17.000Z",  "lastLogin": "2019-09-24T13:00:26.000Z",  "lastUpdated": "2019-09-18T15:14:06.000Z",  "passwordChanged": null,  "profile": {  "country": "Switzerland",  "lastName": "McGill",  "zipCode": "1204",  "city": "Geneva",  "samAccountName": "AMcGill",  "COA\_Department": "4453-EE Engineering",  "distinguishedName": "CN=McGill\\, Avery,OU=Users,OU=Geneva,OU=Switzerland,OU=EU Offices,DC=eu,DC=ad,DC=ea,DC=com",  "title": "Principal Solutions Architect",  "login": "AMcGill@ea.com",  "employeeNumber": "170368",  "objectguid": "Q7+AOHqqRUirmYmSjm3VXQ==",  "spgatewayUsername": "AMcGill@EU.AD.EA.COM",  "COA\_Entity": "377-EA Swiss SARL",  "state": "Geneva",  "department": "EE Engineering",  "userPrincipalName": "AMcGill@ea.com",  "email": "AMcGill@ea.com",  "LegacyUserName": "eu.ad.ea.com\\AMcGill",  "mfaAllowed": "True",  "COA\_Business\_Unit": "1010-Corporate - HQ",  "manager": "CN=Bockelman\\, Eric,OU=Pending,OU=Users,OU=User Accounts,DC=tib,DC=ad,DC=ea,DC=com",  "secondEmail": null,  "vendorName": "",  "firstName": "Avery",  "primaryPhone": "+41 22 316 1303",  "employeeType": "Employee",  "testRespawnGSuiteName": "AMcGill@ea.com",  "mobilePhone": null,  "streetAddress": "Place du Molard 8",  "location": "Geneva",  "ea\_cube": "GEVA\_231",  "pwdLastSet": "2018-06-03T12:17:08+0000",  "pwdChangeAllowed": "True"  },  "credentials": {  "recovery\_question": {  "question": "What is the food you least liked as a child?"  },  "provider": {  "type": "ACTIVE\_DIRECTORY",  "name": "eu.ad.ea.com"  }  },  "\_links": {  "suspend": {  "href": "https://ea.oktapreview.com/api/v1/users/  00u5h8rhqv5Ne4CmB0h7/lifecycle/suspend",  "method": "POST"  },  "expirePassword": {  "href": "https://ea.oktapreview.com/api/v1/users/  00u5h8rhqv5Ne4CmB0h7/lifecycle/expire\_password",  "method": "POST"  },  "changeRecoveryQuestion": {  "href": "https://ea.oktapreview.com/api/v1/users/  00u5h8rhqv5Ne4CmB0h7/credentials/change\_recovery\_question",  "method": "POST"  },  "self": {  "href": "https://ea.oktapreview.com/api/v1/users/  00u5h8rhqv5Ne4CmB0h7"  },  "resetFactors": {  "href": "https://ea.oktapreview.com/api/v1/users/  00u5h8rhqv5Ne4CmB0h7/lifecycle/reset\_factors",  "method": "POST"  },  "deactivate": {  "href": "https://ea.oktapreview.com/api/v1/users/  00u5h8rhqv5Ne4CmB0h7/lifecycle/deactivate",  "method": "POST"  }  }  } |
| Reference | <https://developer.okta.com/docs/reference/api/users/#get-user> |

|  |  |
| --- | --- |
| Configuration | Value |
| Priority | Second |
| Input | Clients Okta userId and groupId. GroupID will initially be stored in a configuration file. |
| Output | HTTP 204 (success) |
| API URL | {{url}}/api/v1/groups/{{groupId}}/users/{{userId}} |
| HTTP Method | PUT |
| Input Headers | 1. Authorization: SSWS {{Okta API Key}} 2. Content-Type: application/json 3. Accept: application/json |
| Request Body | None |
| Response Payload | None |
| Reference | <https://developer.okta.com/docs/reference/api/groups/#add-user-to-group> |

## MFA Reset Endpoint

Description: This endpoint is responsible for resetting a user’s MFA enrollment. User is removed from the MFA\_Okta\_Verify\_TOTP, MFA\_FreeOTP, MFA\_Okta\_Verify\_Push, MFA\_External\_Access\_Allowed and MFA\_SMS, MFA\_RSA groups.

### Endpoint Details

|  |  |
| --- | --- |
| Configuration | Configuration Value |
| Endpoint | /v1/users/${*user-email*}/fix/resetmfa |
| Input Headers | Bearer: ${token} |
| HTTP Method | POST |
| AuthZ Claim | App Assignment - Identity Support Tools - MFA Reset |
| Group | App Assignment - Identity Support Tools - MFA Reset |
| Request Body | None |
| Response Header: Success | HTTP 204 |
| Response Header: Failure | HTTP 404 |
| Response Payload | None |
| Logging | Required. Single AWS Cloudwatch group.  /aws/lambda/identity-services-support-tools-<<environment name>> |

### Business Logic

This API call requires that the Identity Support Tools API reset the users MFA enrollment and remove the user from specific Okta groups. These groups are used during the MFA enrollment process and must be reset at the same time as the MFA enrollment. Note that this API returns a user to a base state and any MFA enrollment exceptions (SMS, RSA, etc..) must be regranted via the normal exception requesting process (outside the scope of this document).

### Orchestration

This API call requires the administrative function to lookup a user’s Okta userId, use the userId to remove the user from the groups used in the MFA enrollment process, and then reset the users MFA if the user is successfully removed from all groups. If the user is not found in any of the groups, this process can proceed. The call to remove users from the Okta groups must be asynchronously executed.

|  |  |
| --- | --- |
| Configuration | Value |
| Priority | First |
| Input | Email address from client application |
| Output | Okta user profile payload from which the userId must be extracted. HTTP 200 header (success) |
| API URL | {{Okta Org}}/api/v1/users/{{email}} |
| HTTP Method | GET |
| Input Headers | 1. Authorization: SSWS {{Okta API Key}} 2. Content-Type: application/json 3. Accept: application/json |
| Request Body | None |
| Response Payload | {  "id": "00u5h8rhqv5Ne4CmB0h7",  "status": "ACTIVE",  "created": "2015-12-08T11:03:45.000Z",  "activated": "2015-12-08T11:03:46.000Z",  "statusChanged": "2019-09-24T12:15:17.000Z",  "lastLogin": "2019-09-24T13:00:26.000Z",  "lastUpdated": "2019-09-18T15:14:06.000Z",  "passwordChanged": null,  "profile": {  "country": "Switzerland",  "lastName": "McGill",  "zipCode": "1204",  "city": "Geneva",  "samAccountName": "AMcGill",  "COA\_Department": "4453-EE Engineering",  "distinguishedName": "CN=McGill\\, Avery,OU=Users,OU=Geneva,OU=Switzerland,OU=EU Offices,DC=eu,DC=ad,DC=ea,DC=com",  "title": "Principal Solutions Architect",  "login": "AMcGill@ea.com",  "employeeNumber": "170368",  "objectguid": "Q7+AOHqqRUirmYmSjm3VXQ==",  "spgatewayUsername": "AMcGill@EU.AD.EA.COM",  "COA\_Entity": "377-EA Swiss SARL",  "state": "Geneva",  "department": "EE Engineering",  "userPrincipalName": "AMcGill@ea.com",  "email": "AMcGill@ea.com",  "LegacyUserName": "eu.ad.ea.com\\AMcGill",  "mfaAllowed": "True",  "COA\_Business\_Unit": "1010-Corporate - HQ",  "manager": "CN=Bockelman\\, Eric,OU=Pending,OU=Users,OU=User Accounts,DC=tib,DC=ad,DC=ea,DC=com",  "secondEmail": null,  "vendorName": "",  "firstName": "Avery",  "primaryPhone": "+41 22 316 1303",  "employeeType": "Employee",  "testRespawnGSuiteName": "AMcGill@ea.com",  "mobilePhone": null,  "streetAddress": "Place du Molard 8",  "location": "Geneva",  "ea\_cube": "GEVA\_231",  "pwdLastSet": "2018-06-03T12:17:08+0000",  "pwdChangeAllowed": "True"  },  "credentials": {  "recovery\_question": {  "question": "What is the food you least liked as a child?"  },  "provider": {  "type": "ACTIVE\_DIRECTORY",  "name": "eu.ad.ea.com"  }  },  "\_links": {  "suspend": {  "href": "https://ea.oktapreview.com/api/v1/users/  00u5h8rhqv5Ne4CmB0h7/lifecycle/suspend",  "method": "POST"  },  "expirePassword": {  "href": "https://ea.oktapreview.com/api/v1/users/  00u5h8rhqv5Ne4CmB0h7/lifecycle/expire\_password",  "method": "POST"  },  "changeRecoveryQuestion": {  "href": "https://ea.oktapreview.com/api/v1/users/  00u5h8rhqv5Ne4CmB0h7/credentials/change\_recovery\_question",  "method": "POST"  },  "self": {  "href": "https://ea.oktapreview.com/api/v1/users/  00u5h8rhqv5Ne4CmB0h7"  },  "resetFactors": {  "href": "https://ea.oktapreview.com/api/v1/users/  00u5h8rhqv5Ne4CmB0h7/lifecycle/reset\_factors",  "method": "POST"  },  "deactivate": {  "href": "https://ea.oktapreview.com/api/v1/users/  00u5h8rhqv5Ne4CmB0h7/lifecycle/deactivate",  "method": "POST"  }  }  } |
| Reference | <https://developer.okta.com/docs/reference/api/users/#get-user> |

|  |  |
| --- | --- |
| Configuration | Value |
| Priority | Second (Executed asynchronously) |
| Input | Clients Okta userId and groupId’s. All group names and their associated ID’s are listed below.   1. Group Name: M\_F\_A\_Off\_Network\_Exceptions\_Enrollment\_Group    1. GroupID: 00geqi59f0N839fAy0x7 2. Group Name: MFA\_External\_Access\_Allowed    1. GroupID: 00gd90mzu4A46Yw3m0x7 3. Group Name: MFA\_FreeOTP    1. GroupID: 00gexwl1uoPwx8c9l0x7 4. Group Name: MFA\_Okta\_Verify\_Push    1. GroupID: 00gc14b465EsEnPYQ0x7 5. Group Name: MFA\_Okta\_Verify\_TOTP    1. GroupID: 00gc14gqwp6q0TRmg0x7 6. Group Name: MFA\_RSA    1. GroupID: 00gbo92unanRae3tN0x7 7. Group Name: MFA\_SMS    1. GroupID: 00gc14dx3p4df8K9F0x7 |
| API URL | {{url}}/api/v1/groups/{{groupId}}/users/{{userId}} |
| HTTP Method | DELETE |
| Input Headers | 1. Authorization: SSWS {{Okta API Key}} 2. Content-Type: application/json 3. Accept: application/json |
| Request Body | None |
| Response Header: Success | HTTP 204 |
| Response Header: Fail | HTTP 404 (user not found) |
| Response Payload | None |
| Reference | <https://developer.okta.com/docs/reference/api/groups/#remove-user-from-group> |

|  |  |
| --- | --- |
| Configuration | Value |
| Priority | Third |
| Input | Clients Okta userId |
| API URL | {{url}}/api/v1/users/{{userId}}/lifecycle/reset\_factors |
| HTTP Method | DELETE |
| Input Headers | 1. Authorization: SSWS {{Okta API Key}} 2. Content-Type: application/json 3. Accept: application/json |
| Request Body | None |
| Response Payload | None |
| Reference | <https://developer.okta.com/docs/reference/api/factors/#reset-factor> |

## MFA Enrollment Exception (Email) Endpoint

Description: This endpoint grants a user an off-network MFA enrollment exception. This allows a user to enroll in MFA while not on an EA whitelisted network.

NOTE: *The next version of the Identity Support Tools will send a Slack message to the end user once the MFA enrollment exception granting process is complete. The message will contain information to reach out to the service desk (IT Ops EE?) if the user did not request the exception.*

|  |  |
| --- | --- |
| Configuration | Configuration Value |
| Endpoint | /v1/users/{{email}]/fix/assign/M\_F\_A\_Off\_Network\_Exceptions\_Enrollment\_Group |
| Headers | Bearer: ${token} |
| AuthZ Claim | App Assignment – Identity Support Tools – Verify Identity |
| Group | App Assignment – Identity Support Tools – Verify Identity |
| Method | GET |
| Scopes | The API will validate the Bearer token and check the scopes returned in the payload. Only if a proper scope was provided will the API perform the requested action. |
| Payload |  |

### Business Logic

This endpoint provides an MFA enrollment exception allowing users to enroll in MFA off of EA’s network. The logic for this endpoint is as follows:

1. Obtain the client’s Okta userId by getting the user profile via looking the user up via their email address
2. Add the user to the MFA enrollment exception group

### Orchestration

|  |  |
| --- | --- |
| Configuration | Value |
| Priority | First |
| Input | Email address from client application |
| Output | Okta user profile payload from which the userId must be extracted. HTTP 200 header (success) |
| API URL | {{Okta Org}}/api/v1/users/{{email}} |
| HTTP Method | GET |
| Input Headers | 1. Authorization: SSWS {{Okta API Key}} 2. Content-Type: application/json 3. Accept: application/json |
| Request Body | None |
| Response Payload | {  "id": "00u5h8rhqv5Ne4CmB0h7",  "status": "ACTIVE",  "created": "2015-12-08T11:03:45.000Z",  "activated": "2015-12-08T11:03:46.000Z",  "statusChanged": "2019-09-24T12:15:17.000Z",  "lastLogin": "2019-09-24T13:00:26.000Z",  "lastUpdated": "2019-09-18T15:14:06.000Z",  "passwordChanged": null,  "profile": {  "country": "Switzerland",  "lastName": "McGill",  "zipCode": "1204",  "city": "Geneva",  "samAccountName": "AMcGill",  "COA\_Department": "4453-EE Engineering",  "distinguishedName": "CN=McGill\\, Avery,OU=Users,OU=Geneva,OU=Switzerland,OU=EU Offices,DC=eu,DC=ad,DC=ea,DC=com",  "title": "Principal Solutions Architect",  "login": "AMcGill@ea.com",  "employeeNumber": "170368",  "objectguid": "Q7+AOHqqRUirmYmSjm3VXQ==",  "spgatewayUsername": "AMcGill@EU.AD.EA.COM",  "COA\_Entity": "377-EA Swiss SARL",  "state": "Geneva",  "department": "EE Engineering",  "userPrincipalName": "AMcGill@ea.com",  "email": "AMcGill@ea.com",  "LegacyUserName": "eu.ad.ea.com\\AMcGill",  "mfaAllowed": "True",  "COA\_Business\_Unit": "1010-Corporate - HQ",  "manager": "CN=Bockelman\\, Eric,OU=Pending,OU=Users,OU=User Accounts,DC=tib,DC=ad,DC=ea,DC=com",  "secondEmail": null,  "vendorName": "",  "firstName": "Avery",  "primaryPhone": "+41 22 316 1303",  "employeeType": "Employee",  "testRespawnGSuiteName": "AMcGill@ea.com",  "mobilePhone": null,  "streetAddress": "Place du Molard 8",  "location": "Geneva",  "ea\_cube": "GEVA\_231",  "pwdLastSet": "2018-06-03T12:17:08+0000",  "pwdChangeAllowed": "True"  },  "credentials": {  "recovery\_question": {  "question": "What is the food you least liked as a child?"  },  "provider": {  "type": "ACTIVE\_DIRECTORY",  "name": "eu.ad.ea.com"  }  },  "\_links": {  "suspend": {  "href": "https://ea.oktapreview.com/api/v1/users/  00u5h8rhqv5Ne4CmB0h7/lifecycle/suspend",  "method": "POST"  },  "expirePassword": {  "href": "https://ea.oktapreview.com/api/v1/users/  00u5h8rhqv5Ne4CmB0h7/lifecycle/expire\_password",  "method": "POST"  },  "changeRecoveryQuestion": {  "href": "https://ea.oktapreview.com/api/v1/users/  00u5h8rhqv5Ne4CmB0h7/credentials/change\_recovery\_question",  "method": "POST"  },  "self": {  "href": "https://ea.oktapreview.com/api/v1/users/  00u5h8rhqv5Ne4CmB0h7"  },  "resetFactors": {  "href": "https://ea.oktapreview.com/api/v1/users/  00u5h8rhqv5Ne4CmB0h7/lifecycle/reset\_factors",  "method": "POST"  },  "deactivate": {  "href": "https://ea.oktapreview.com/api/v1/users/  00u5h8rhqv5Ne4CmB0h7/lifecycle/deactivate",  "method": "POST"  }  }  } |
| Reference | <https://developer.okta.com/docs/reference/api/users/#get-user> |

|  |  |
| --- | --- |
| Configuration | Value |
| Priority | Second |
| Input | Clients Okta userId, Okta MFA enrollment exceptions groupId |
| Output | None |
| API URL | {{url}}/api/v1/groups/{{groupId}}/users/{{userId}} |
| HTTP Method | PUT |
| Input Headers | 1. Authorization: SSWS {{Okta API Key}} 2. Content-Type: application/json 3. Accept: application/json |
| Request Body | None |
| Response Headers: Success | HTTP 204 |
| Response Payload | None |
| Reference | <https://developer.okta.com/docs/reference/api/groups/#add-user-to-group> |

# Identity Proofing

The Identity Services Support Tools Identity Proofing functionality enables the EA Service Desk, Site IT, operations teams, and anyone else with access to these tools, to issue an MFA challenge and verify the identity of the person they are interacting with. This section outlines the development business logic and orchestration for developing the API services to fulfill this functionality.

## Business Logic

The below endpoints obtain the client’s Okta userId by getting the user profile via looking the user up via their email address. This user id will then be leveraged to get the list of factors enrolled by an user. And user will be challenged to use one of the enrolled factors appearing in order of RSA, Okta Verify Push, Okta Verify TOTP, TOTP(FreeOTP), SMS. User then needs to verify the factor which will then return confirmation of success or failure.

## Orchestration

This Identity Services Support Tools function looks up the user by their email address, returns the Okta userId, then leverages the Okta user id to get the factors enrolled by the user and verify the same. The following Okta API’s are being leveraged in the order defined below.

### Getting user by Email :

|  |  |
| --- | --- |
| Configuration | Value |
| Priority | First |
| Input | Email address from client application |
| Output | Okta user profile payload from which the userId must be extracted. HTTP 200 header (success) |
| API URL | {{Okta Org}}/api/v1/users/{{email}} |
| HTTP Method | GET |
| Input Headers | 1. Authorization: SSWS {{Okta API Key}} 2. Content-Type: application/json 3. Accept: application/json |
| Request Body | None |
| Response Header: Failure | HTTP 404 (user not found) |
| Response Payload: Failure | {  "errorCode": "E0000007",  "errorSummary": "Not found: Resource not found: test@tafzs.com (User)",  "errorLink": "E0000007",  "errorId": "oaeIUqMGEbdQmmjzNP1xGM\_cQ",  "errorCauses": []  } |
| Response Header: Success | HTTP 200 |
| Response Payload: Success | [      {          "id": "00u23krrqiOs0EVAu357",          "status": "STAGED",          "created": "2019-12-04T03:45:18.000Z",          "activated": null,          "statusChanged": null,          "lastLogin": null,          "lastUpdated": "2019-12-04T03:45:18.000Z",          "passwordChanged": null,          "profile": {              "firstName": "Isaac",              "lastName": "Brock",              "mobilePhone": null,              "secondEmail": null,              "login": "isaac@example.com",              "email": "isaac@example.com"          },          "credentials": {              "provider": {                  "type": "OKTA",                  "name": "OKTA"              }          },          "\_links": {              "self": {                  "href": "https://tafzstafzs.okta.com/api/v1/users/00u23krrqiOs0EVAu357"              }          }      }  ] |
| Reference | <https://developer.okta.com/docs/reference/api/users/#get-user> |

### List factors enrolled by user:

|  |  |
| --- | --- |
| Configuration | Configuration Value |
| Execution | Second |
| Input | UserID of the User |
| Output Value | Okta user profile payload from which the factorId of enrolled factors must be extracted. HTTP 200 header (success) |
| API URL | {{url}}/api/v1/users/{{factorId}}/factors |
| HTTP Method | GET |
| Input Headers | 1. Authorization: SSWS {{Okta API Key}} 2. Content-Type: application/json 3. Accept: application/json |
| Request Body | None |
| Response Header: Failure | HTTP 404 (user not found) |
| Response Payload: Failure | {  "errorCode": "E0000007",  "errorSummary": "Not found: Resource not found: 00u2e7zxrzybDhBh937 (User)",  "errorLink": "E0000007",  "errorId": "oaedkw6oFgLQuOi\_k7t4N7FyA",  "errorCauses": []  } |
| Response Header: Success | HTTP 200 |
| Response Payload: Success | [{  "id": "ufs2bysphxKODSZKWVCT",  "factorType": "question",  "provider": "OKTA",  "vendorName": "OKTA",  "status": "ACTIVE",  "created": "2014-04-15T18:10:06.000Z",  "lastUpdated": "2014-04-15T18:10:06.000Z",  "profile": {  "question": "favorite\_art\_piece",  "questionText": "What is your favorite piece of art?"  },  "\_links": {  "questions": {  "href": "https://${yourOktaDomain}/api/v1/users/00u15s1KDETTQMQYABRL/factors/questions",  "hints": {  "allow": [  "GET"  ]  }  },  "self": {  "href": "https://${yourOktaDomain}/api/v1/users/00u15s1KDETTQMQYABRL/factors/ufs2bysphxKODSZKWVCT",  "hints": {  "allow": [  "GET",  "DELETE"  ]  }  },  "user": {  "href": "https://${yourOktaDomain}/api/v1/users/00u15s1KDETTQMQYABRL",  "hints": {  "allow": [  "GET"  ]  }  }  }  },  {  "id": "ostf2gsyictRQDSGTDZE",  "factorType": "token:software:totp",  "provider": "OKTA",  "status": "PENDING\_ACTIVATION",  "created": "2014-06-27T20:27:33.000Z",  "lastUpdated": "2014-06-27T20:27:33.000Z",  "profile": {  "credentialId": "dade.murphy@example.com"  },  "\_links": {  "next": {  "name": "activate",  "href": "https://${yourOktaDomain}/api/v1/users/00u15s1KDETTQMQYABRL/factors/ostf2gsyictRQDSGTDZE/lifecycle/activate",  "hints": {  "allow": [  "POST"  ]  }  },  "self": {  "href": "https://${yourOktaDomain}/api/v1/users/00u15s1KDETTQMQYABRL/factors/ostf2gsyictRQDSGTDZE",  "hints": {  "allow": [  "GET"  ]  }  },  "user": {  "href": "https://${yourOktaDomain}/api/v1/users/00u15s1KDETTQMQYABRL",  "hints": {  "allow": [  "GET"  ]  }  }  },  "\_embedded": {  "activation": {  "timeStep": 30,  "sharedSecret": "HE64TMLL2IUZW2ZLB",  "encoding": "base32",  "keyLength": 16  }  }  },  {  "id": "sms2gt8gzgEBPUWBIFHN",  "factorType": "sms",  "provider": "OKTA",  "status": "ACTIVE",  "created": "2014-06-27T20:27:26.000Z",  "lastUpdated": "2014-06-27T20:27:26.000Z",  "profile": {  "phoneNumber": "+1-555-415-1337"  },  "\_links": {  "verify": {  "href": "https://${yourOktaDomain}/api/v1/users/00u15s1KDETTQMQYABRL/factors/sms2gt8gzgEBPUWBIFHN/verify",  "hints": {  "allow": [  "POST"  ]  }  },  "self": {  "href": "https://${yourOktaDomain}/api/v1/users/00u15s1KDETTQMQYABRL/factors/sms2gt8gzgEBPUWBIFHN",  "hints": {  "allow": [  "GET",  "DELETE"  ]  }  },  "user": {  "href": "https://${yourOktaDomain}/api/v1/users/00u15s1KDETTQMQYABRL",  "hints": {  "allow": [  "GET"  ]  }  }  }  }  ] |
| Reference | <https://developer.okta.com/docs/reference/api/factors/> |

#### Verify Factors Order of Priority:

Depending on the enrolled factors user will be challenged in the order below. (TODO: List out the order)

1. RSA
2. Okta Verify Push
3. Okta Verify TOTP
4. TOTP(FreeOTP)
5. SMS

#### RSA: TODO

#### Okta Verify Push: (validated using POSTMAN)

Below demonstrates 2 endpoints one to issue the challenge and the other to verify the same.

##### Issue push challenge

|  |  |
| --- | --- |
| Configuration | Configuration Value |
| Execution | Third (Push verification step 1) |
| Input | UserId and FactorId |
| Output Value | HTTP Status 201 |
| API URL | {{url}}/api/v1/users/${userId}/factors/${factorId}/verify |
| HTTP Method | POST |
| Input Headers | 1. Authorization: SSWS {{Okta API Key}} 2. Content-Type: application/json 3. Accept: application/json |
| Request Body | None |
| Response Header: Success | HTTP 201 |
| Response Payload: Success | {  "expiresAt": "2015-04-01T15:57:32.000Z",  "factorResult": "WAITING",  "\_links": {  "poll": {  "href": "https://${yourOktaDomain}/api/v1/users/00u15s1KDETTQMQYABRL/factors/opfh52xcuft3J4uZc0g3/transactions/mst1eiHghhPxf0yhp0g",  "hints": {  "allow": [  "GET"  ]  }  },  "cancel": {  "href": "https://${yourOktaDomain}/api/v1/users/00u15s1KDETTQMQYABRL/factors/opfh52xcuft3J4uZc0g3/transactions/mst1eiHghhPxf0yhp0g",  "hints": {  "allow": [  "DELETE"  ]  }  }  }  } |
| Reference | <https://developer.okta.com/docs/reference/api/factors/> |

##### Verify Push Challenge

|  |  |
| --- | --- |
| Configuration | Configuration Value |
| Execution | Third (Push verification step2) |
| Input | UserId , FactorId and transactionId |
| Output Value | HTTP 200 header (success) |
| API URL | /api/v1/users/${userId}/factors/${factorId}/transactions/${transactionId} |
| HTTP Method | GET |
| Input Headers | 1. Authorization: SSWS {{Okta API Key}} 2. Content-Type: application/json 3. Accept: application/json |
| Request Body | None |
| Response Header: Rejected | HTTP 200 |
| Response Payload: Rejected | {  "factorResult": "REJECTED",  "\_links": {  "verify": {  "href": "https://${yourOktaDomain}/api/v1/users/00u15s1KDETTQMQYABRL/factors/opfh52xcuft3J4uZc0g3/verify",  "hints": {  "allow": [  "POST"  ]  }  },  "factor": {  "href": "https://${yourOktaDomain}/api/v1/users/00u15s1KDETTQMQYABRL/factors/opfh52xcuft3J4uZc0g3",  "hints": {  "allow": [  "GET",  "DELETE"  ]  }  }  }  } |
| Response Header: Timeout | HTTP 200 |
| Response Payload: Timeout | {  "factorResult": "TIMEOUT",  "\_links": {  "verify": {  "href": "https://${yourOktaDomain}/api/v1/users/00u15s1KDETTQMQYABRL/factors/opfh52xcuft3J4uZc0g3/verify",  "hints": {  "allow": [  "POST"  ]  }  },  "factor": {  "href": "https://${yourOktaDomain}/api/v1/users/00u15s1KDETTQMQYABRL/factors/opfh52xcuft3J4uZc0g3",  "hints": {  "allow": [  "GET",  "DELETE"  ]  }  }  }  } |
| Response Header: Waiting | HTTP 201 Created |
| Response Payload: Waiting | {  "expiresAt": "2015-04-01T15:57:32.000Z",  "factorResult": "WAITING",  "\_links": {  "poll": {  "href": "https://${yourOktaDomain}/api/v1/users/00u15s1KDETTQMQYABRL/factors/opfh52xcuft3J4uZc0g3/transactions/mst1eiHghhPxf0yhp0g",  "hints": {  "allow": [  "GET"  ]  }  },  "cancel": {  "href": "https://${yourOktaDomain}/api/v1/users/00u15s1KDETTQMQYABRL/factors/opfh52xcuft3J4uZc0g3/transactions/mst1eiHghhPxf0yhp0g",  "hints": {  "allow": [  "DELETE"  ]  }  }  }  } |
| Response Header: Success | HTTP 200 |
| Response Payload: Success | {  "factorResult": "SUCCESS"  } |
| Reference | <https://developer.okta.com/docs/reference/api/factors/> |

#### OKTA Verify TOTP (OTP): (validated using POSTMAN)

Below demonstrates an endpoint to verify the factor py providing the OTP passcode in the request body.

|  |  |
| --- | --- |
| Configuration | Configuration Value |
| Execution | Third(TOTP verification) |
| Input | UserId and FactorId |
| Output Value | HTTP 200 header (success) |
| API URL | {{url}}/api/v1/users/${userId}/factors/${factorId}/verify |
| HTTP Method | POST |
| Input Headers | 1. Authorization: SSWS {{Okta API Key}} 2. Content-Type: application/json 3. Accept: application/json |
| Request Body | {  "passCode": "382883"  } |
| Response Header: Fail | HTTP 403 Forbidden |
| Response Payload: Failure(Rejected) | {  "errorCode": "E0000068",  "errorSummary": "Invalid Passcode/Answer",  "errorLink": "E0000068",  "errorId": "oaei\_IfXcpnTHit\_YEKGInpFw",  "errorCauses": [  {  "errorSummary": "Your passcode doesn't match our records. Please try again."  }  ]  } |
| Response Header: Success | HTTP 200 |
| Response Payload: Success | {  "factorResult": "SUCCESS"  } |
| Reference | <https://developer.okta.com/docs/reference/api/factors/> |

#### Free OTP (OTP):

FreeOTP is an open source variant of Google Authenticator. FreeOTP doesn’t leverage the Google Play services on Android devices, thus is a Google Authenticator variant that can be used by Android users in China (despite people having Android phones in China, Google Play services and Google applications are not allowed in China). This MFA type is just Google Authenticator and is also equivalent of Okta Verify TOTP Factor which is documented above.

#### SMS: (from documentation, not validated using POSTMAN)

Below demonstrates 2 endpoints one to issue the SMS challenge and the other to verify the same.

##### Issue SMS challenge

|  |  |
| --- | --- |
| Configuration | Value |
| Priority | Third(SMS verification step1) |
| Input | Userid ,factorId and Passcode |
| Output | HTTP 200 header (success) |
| API URL | {{url}}/api/v1/users/${userId}/factors/${factorId}/verify |
| HTTP Method | POST |
| Input Headers | 1 .Authorization: SSWS {{Okta API Key}}  2. Content-Type: application/json  3. Accept: application/json |
| Request Body | {  "passCode": "382883"  } |
| Response Header: Failure(Too Many Requests) | HTTP 429 Too Many Requests |
| Response Payload: Too Many Requests | {  "errorCode": "E0000109",  "errorSummary": "An SMS message was recently sent. Please wait 30 seconds before trying again.",  "errorLink": "E0000109",  "errorId": "oaeneEaQF8qQrepOWHSkdoejw",  "errorCauses": []  } |
| Response Header: Success | HTTP 200 |
| Response Payload: Success | {  "factorResult": "CHALLENGE",  "profile": {  "phoneNumber": "+12532236986"  },  "\_links": {  "verify": {  "href": "https://${yourOktaDomain}/api/v1/users/00u15s1KDETTQMQYABRL/factors/smsszf1YNUtGWTx4j0g3/verify",  "hints": {  "allow": [  "POST"  ]  }  },  "factor": {  "href": "https://${yourOktaDomain}/api/v1/users/00u15s1KDETTQMQYABRL/factors/smsszf1YNUtGWTx4j0g3",  "hints": {  "allow": [  "GET",  "DELETE"  ]  }  }  }} |
| Reference | <https://developer.okta.com/docs/reference/api/factors/> |

##### Verify SMS challenge

|  |  |
| --- | --- |
| Configuration | Value |
| Priority | Third(SMS verification step2) |
| Input | Userid ,factorId and Passcode |
| Output | HTTP 200 header (success) |
| API URL | {{url}} /api/v1/users/${userId}/factors/${factorId}/verify |
| HTTP Method | POST |
| Input Headers | 1 .Authorization: SSWS {{Okta API Key}}  2. Content-Type: application/json  3. Accept: application/json |
| Request Body | {  "passCode": "382883"  } |
| Response Header: Failure | HTTP 403 Forbidden |
| Response Payload: Failure (Forbidden) | {  "errorCode": "E0000068",  "errorSummary": "Invalid Passcode/Answer",  "errorLink": "E0000068",  "errorId": "oaei\_IfXcpnTHit\_YEKGInpFw",  "errorCauses": [  {  "errorSummary": "Your passcode doesn't match our records. Please try again."  }  ]  } |
| Response Header: Success | HTTP 200 |
| Response Payload: Success | {  "factorResult": "SUCCESS"  } |
| Reference | <https://developer.okta.com/docs/reference/api/factors/> |

# Appendix

## WAF

### References:

* <https://github.com/aws-samples/aws-waf-sample/blob/master/waf-owasp-top-10/owasp_10_base.yml>
* <https://d0.awsstatic.com/whitepapers/Security/aws-waf-owasp.pdf>
* <https://s3.us-east-2.amazonaws.com/awswaf-owasp/owasp_10_base.yml>

## HA & DR

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Component | Scope | | | SLA(Service Level Agreement) | Links |
| Region | Availability Zone | |
| WAF | YES | YES | | Monthly Uptime Percentage : Less than 99.95% but greater than or equal to 99.0% | <https://aws.amazon.com/waf/sla/> |
| CloudFront | YES | N/A | | Amazon CloudFront available with a Monthly Uptime Percentage (defined below) of at least 99.9% | <https://aws.amazon.com/cloudfront/sla/> |
| API Gateway | YES | N/A | | API Gateway available with a Monthly Uptime Percentage of at least 99.95% for each AWS region | <https://aws.amazon.com/api-gateway/sla/> |
| Lambda | N/A | | YES | Lambda available with a Monthly Uptime Percentage for each AWS region, during any monthly billing cycle, of at least 99.95% | <https://aws.amazon.com/api-gateway/sla/> |
| Amazon Cognito | N/A | | N/A | AWS will use commercially reasonable efforts to make Cognito available with a Monthly Uptime Percentage for each AWS region, during any monthly billing cycle, of at least 99.9% | <https://aws.amazon.com/cognito/sla/> |
| CloudWatch | YES | | N/A | AWS will use commercially reasonable efforts to make each Function of Amazon CloudWatch available with a Monthly Uptime Percentage for each AWS region, during any monthly billing cycle, of at least 99.9% | <https://aws.amazon.com/cloudwatch/sla/> |
| OKTA | N/A | | N/A |  | <https://support.okta.com/help/s/article/Okta-Support-Service-Level-Agreements-by-Customer-Success-Package> |

## OKTA API

### References:

<https://developer.okta.com/docs/reference/api/factors/#getting-started-with-the-factors-api>

[https://developer.okta.com/docs/reference/api/factors/#issue-an-sms-factor-challenge](https://developer.okta.com/docs/reference/api/factors/" \l "issue-an-sms-factor-challenge)

## Architecture

### References:

**A picture containing screenshot

Description automatically generated**