DRAFT - IT Project Guidance

On Entra ID Identity Access Management (IdAM)

Version:

0.1

## Purpose

This document aims to provide both non-technical and technical stakeholders with a clear overview of the fundamental principles and key considerations involved in using Entra ID for managing access to services. It will explain how Entra ID supports secure identity and access management (IdAM), outline its core features, and describe its role in enabling authorised users to access relevant systems and resources. The content is designed to be accessible for readers with varying levels of technical expertise, ensuring a broad understanding of how Entra ID facilitates secure and efficient service access within an organisation.

## Synopsis

TODO

## Contents

[Purpose 1](#_Toc195791704)

[Synopsis 1](#_Toc195791705)

[Contents 2](#_Toc195791706)

[Introduction 3](#_Toc195791707)

[Background of this work 3](#_Toc195791708)

[Context 3](#_Toc195791709)

[Background of issue 3](#_Toc195791710)

[Problem Statement 3](#_Toc195791711)

[H3 4](#_Toc195791712)

[H4 4](#_Toc195791713)

[Appendices 5](#_Toc195791714)

[Appendix A - Document Information 5](#_Toc195791715)

[Versions 5](#_Toc195791716)

[Images 5](#_Toc195791717)

[Tables 5](#_Toc195791718)

[References 5](#_Toc195791719)

[Review Distribution 5](#_Toc195791720)

[Audience 5](#_Toc195791721)

[Structure 5](#_Toc195791722)

[Diagrams 6](#_Toc195791723)

[Acronyms 6](#_Toc195791724)

[Terms 6](#_Toc195791725)

# Context

Ssytems require a number of core capabilities in order to in turn support the dvelopemnt of business specific capabilities. User Management, Access Management, Authorisation Management are an example of such.   
These core capabilities either come with a purchased product, or if a custom system have to be developed, delegated, offloaded, or omitted, the last being a design failure.

With a pivoting towards Agile delivery, with a poor grasp of Agile doesn’t obviate engineering, the development of minimum viable projects (MVPs) put pressure on skiping as much as possible to get products in front of stakeholders as soon as possible.   
As such offloading as much as possible is considered as a good outcome, so as to not have to repeat what is already built.

However, there is a big difference between offloading, and delegating. Offlading/Delegating means not doing it, and instead passing it to an external service to do it.   
Federating is different – it means doing the work, but taking on board external inputs.

We have previously written about the distinction between the core aspects of identity and access management being Association, Identity, using in-system or external IdP based Authentication, Access, Authorisation, and Auditing.

Access is often conflated with Authorisation however they are distinct. Access is Access To a system and should be based on coarse classification of the end user (Employee, Region, time of day, etc.).

Whereas Authorisation is Authorisation Within the system, where the system manages its own System’s Users, Roles Responsibilities and Permissions.

However we also discussed that there are pressures that push development of MVPs towards skipping as much development as possible, to rely on external enterprise services that provide user and role management.

It can even so far as to not develop Users and Roles in a system, relying soley on an external IdAM to manage this. This is a design mistake.   
An service must always manage its own users, and roles and permissions.   
  
Where they go wrong is thinking that if it is managed externally, one can rely solely on the external system. And develop tokens that are in turn used to develop cached or roundtripped messages defining the users roles.   
This is wrong on multiple counts. One can rely on an external system’s management of users and roles to hint to the system what roles are assigned to each.

Appendices

Appendix A - Document Information

Authors & Collaborators

* Sky Sigal, Solution Architect

### Versions

* 1. Initial Draft

### Images

[Figure 1: TODO Image 2](#_Toc144995112)

### Tables

[Table 1: TODO Table 3](#_Toc145048484)

[Table 2: TODO Table 2 3](#_Toc145048485)

### References

**There are no sources in the current document.**

### Review Distribution

The document was distributed for review as below:

|  |  |
| --- | --- |
| Identity | Notes |
|  |  |
|  |  |
|  |  |

### Audience

The document is technical in nature, but parts are expected to be read and/or validated by a non-technical audience.

### Structure

Where possible, the document structure is guided by either ISO-\* standards or best practice.

### Diagrams

Diagrams are developed for a wide audience. Unless specifically for a technical audience, where the use of industry standard diagram types (ArchiMate, UML, C4), is appropriate, diagrams are developed as simple “box & line” monochrome diagrams.

### Acronyms

API

: [Application Programming Interface](#Term_ApplicationProgrammingInterface).

DDD

: Domain Driven Design

GUI

: [Graphical User Interface](#Term_ApplicationProgrammingInterface). A form of [UI](#Acronym_UI).

ICT

: acronym for Information & Communication Technology, the domain of defining Information elements and using technology to automate their communication between entities. [IT](#Acronym_IT) is a subset of ICT.

IT

: acronym for Information, using Technology to automate and facilitate its management.

UI

: User Interface. Contrast with [API](#Acronym_API).

### Terms

Refer to the project’s Glossary.

Application Programming Interface

: an Interface provided for other systems to invoke (as opposed to User Interfaces).

Capability

: a capability is what an organisation or system must be able to achieve to meet its goals. Each capability belongs to a domain and is realised through one or more functions that, together, deliver the intended outcome within that area of concern.

Domain

: a domain is a defined area of knowledge, responsibility, or activity within an organisation or system. It groups related capabilities, entities, and functions that collectively serve a common purpose. Each capability belongs to a domain, and each function operates within one.

Entity

: an entity is a core object of interest within a domain, usually representing a person, place, thing, or event that holds information and can change over time, such as a Student, School, or Enrolment.

Function

: a function is a specific task or operation performed by a system, process, or person. Functions work together to enable a capability to be carried out. Each function operates within a domain and supports the delivery of one or more capabilities.

Person

: a physical person, who has one or more Personas. Not necessarily a system User.

Persona

: a facet that a Person presents to a Group of some kind.

Quality

: a quality is a measurable or observable attribute of a system or outcome that indicates how well it meets expectations. Examples include reliability, usability, and performance. Refer to the ISO-25000 SQuaRE series of standards.

User

: a human user of a system via its UIs.

User Interface

: a system interface intended for use by system users. Most computer system UIs are Graphics User Interfaces ([GUI](#Acronym_GUI)) or Text/Console User Interfaces (TUI).