OpenArchitectureModel Specification

Release 0.1.0 (Draft 2024-11-6)

Sheikh Mohammad Sajid

Sheikh Mohammad sajid (editor)

Nov 06, 2024

Contents

Introduction

1.1 Introduction

OpenArchitectureModel is a *language for defining and querying software architecture models*. Its main goal is to enable developers to define architecture elements in a standard way.

OpenArchitectureModel is an open standard developed by *Sheikh Mohammad Sajid*.

This document describes version 0.1.0 (Draft 2024-11-6) of the core OpenArchitectureModel standard. It is intended that it will be superseded by new incremental releases with additional features in the future.

1.1.1 Design Goals

The design goals of OpenArchitectureModel are the following:

- Fast, safe, and portable *semantics*:

    - **Well-defined**: fully and precisely defines the syntax.

    - **Language-independent**: does not privilege any particular language, programming model, or object model.

    - **Open**: programs can interoperate with their environment in a simple and universal manner.

- Efficient and portable *representation*:

    - **Modular**: programs can be split up in smaller parts that can be transmitted, cached, and consumed separately.

    - **Portable**: makes no architectural assumptions that are not broadly supported across modern hardware.

1.1.2 Scope

At its core, OpenArchitectureModel is a model for storing and retrieving data about software elements. To encompass their variety and enable maximum reuse, the OpenArchitectureModel specification is split and layered into several documents.

This document is concerned with the core layer of OpenArchitectureModel. It defines the base model, query syntax, validation, and execution semantics. It does not, however, define how OpenArchitectureModel data are stored within a specific environment they execute in, nor how they are invoked from such an environment.

1.2 Overview

1.2.1 Concepts

OpenArchitectureModel is a high-level, SQL-like programming language. This language is structured around the following concepts:

Organization An organization operates as a single tenant, maintaining its own set of architectural information that is versioned collectively. Transactions are restricted to the scope of one organization and cannot extend across multiple organizations. User access and permissions are likewise confined to a single organization, ensuring isolated data governance and security at the tenant level.

Space Spaces are functional areas within an organization, designed to group and manage related information specific to teams, departments, groups, or sections. These spaces follow a hierarchical structure, supporting multiple sub-levels to reflect organizational complexity. The hierarchical model ensures that information is organized according to business needs and roles, promoting streamlined workflows and role-based access management.

Application

> An application is a distinct software component or module that resides within a single Space. Each application is associated exclusively with one Space, allowing it to inherit the permissions, hierarchy, and structural organization of that Space. Applications serve as functional units that provide specific capabilities or services within the organization's system, aligned with the scope and access controls of the Space to which they belong. This ensures that applications operate within defined organizational boundaries, facilitating modular deployment, isolated data handling, and efficient role-based access management.

Framework

Profile

Element

Extensions

Release

Deployment

Architecture Decision Records

Fitness Functions

C

D

F

I

M

N

P

T

U

V