1.1 Problem

Return to Introduction

A major problem confronting the users of DIDO Platforms are getting products that support the end-users Non-Functional requirements. The two most noteworthy Non-Functional requirements are Interoperability and Portability of products built upon the existing DIDO platforms. However, other Non-Functional requirements such as Maintainability are of concern. Maintainability covers a broad spectrum of areas:

- ***Modularity*** *- Degree to which a system or computer program is composed of discrete components such that a change to one component has minimal impact on other components.*
- ***Reusability*** *- Degree to which an asset can be used in more than one system, or in building other assets.*
- ***Analysability*** *- Degree of effectiveness and efficiency with which it is possible to assess the impact on a product or system of an intended change to one or more of its parts, or to diagnose a product for deficiencies or causes of failures, or to identify parts to be modified.*
- ***Modifiability*** *- Degree to which a product or system can be effectively and efficiently modified without introducing defects or degrading existing product quality.*
- ***Testability*** *- Degree of effectiveness and efficiency with which test criteria can be established for a system, product or component and tests can be performed to determine whether those criteria have been met.*

Often, the DIDO Platforms use proprietary languages, libraries, and interfaces which ultimately affect the Interoperability, Portability and maintainability of end user solutions built upon the proprietary solutions causing Vendor Lock-In. The response from the DIDO Platforms is often *"our product is Open Source Software (OSS)"* therefore it is a standard or our specification is a de facto Standard. While there is some legitimacy in these responses, it is not adequate for long term (30+ year) lifecycles or for Mission Critical Systems.

Additionally, the use of *"languages, libraries, and interfaces"* require a labor force that is highly trained and skilled in specialized general computing rather than the trained and skilled in the business domain. In other words, the end-user must find people who are educated and trained in their specific domain (i.e., finance, accounting, taxes, supply chain, environment, public records, etc.) as well as having the computing skills to use the languages, libraries and interfaces for each DIDO Platform.

This is not unlike the problems that have confronted other domains. For example, the DataBase Management System (DBMS) and Operating System (OS) Platform domains. Both of these domains have solved many of their problems by relying on formal, well defined and standardized Command line Interfaces developed in Technical Standards Bodies. See IEEE 1003.1-2017 - IEEE Standard for Information Technology--Portable Operating System Interface (POSIX(R)) Base Specifications and dblang-sql-part1.