ARCHITECTURAL OPTIONS FOR SECURING THE TDM CALCULATOR APPLICATION

Abstract

This report examines architectural options for securing the Traffic Demand Management (TDM) Calculator, a Single-Page Application developed by Hack for LA for the City of Los Angeles. In 2023, the City requested integration with Angeleno, its Okta-based Workforce Identity Cloud (WIC) system, which provides single sign-on across multiple public-facing applications. Our analysis finds that Angeleno, as currently implemented, does not support secure integration with SPA architectures like the TDM Calculator because it lacks a WIC Custom Authentication/Authorization server capable of validating API-level requests. Without this capability, TDM endpoints would be vulnerable to impersonation attacks. The TDM Calculator's existing custom IAM system, which provides role-based security, per-calculation ownership, and token-based session management, remains secure and has undergone penetration testing. While integration with Angeleno is desirable for user experience and centralized citywide identity management, it will only be feasible once the City upgrades to an appropriate authentication service such as Auth0 or extends Angeleno with custom API security features. Until then, TDM will maintain its self-contained IAM implementation while continuing to monitor and remediate vulnerabilities.

Prepared for

LADOT & LOS ANGELES CITY PLANNING

Prepared by

HACK FOR LA

TDM Calculator Team

March 2024, updated with abstract in October 2025

Authors/Editors
John Darragh, Hack for LA TDM Project Architect
Bonnie Wolfe, Senior Project Manager

Summary

Hack for LA (HfLA) started building the TDM calculator in 2019. HfLA was asked to integrate Angeleno in 3rd quarter 2023 when we were approached about what the requirements would be for integrating VMT Calculator with TDM Calculator so that they could share information. This document explores the technical requirements for implementation and limitations that we have discovered. The TDM Calculator is constructed as a Single-Page Web Application (SPA), vs the Traditional Multi-Page Application (see here for an explanation).

At this time, in order to maintain a high level of security for the TDM Calculator application and its users, we cannot implement Angeleno. The current IAM (Identity and Access Management) on TDM Calculator is secure, and we will continue to make improvements as vulnerabilities are identified.

HfLA is interested in integrating with Angeleno. In order for that to be feasible, upgrades to the service will have to be made (adding of a WIC Custom Authentication/Authorization server, or moving to the Auth0 services).

Analysis of Angelo usage

The city is implementing a sign on system for its public-facing applications (Angeleno), as part of a portal strategy. About 15 applications are currently integrated with Angeleno to varying degrees. Some do not use Angeleno's authentication features at all (e.g. NavigateLA, Bureau of Engineering Customer Service Portal), and the ones that do use Angeleno authentication appear to be traditional web applications. Based on cursory experimentation, I have not been able to find any Single Page Web Applications that are currently integrated with Angeleno.

Technical Details

The TDM application is designed using a common modern web application architecture, with separate "client" and "server" software.

The **client** consists of code written to run in an internet browser, sometimes referred to as a Single-Page Application (SPA). Since the particular code library we use to build our application is called "React", it can also be called a React application. Since the browser runs on the user's machine, it is subject to manipulation of various sorts by bad actors, and is considered "untrusted".

The **server**, which we will call a "Web API server" runs in the cloud on the city-managed Azure account. It contains the critical parts of the application, including the database and functions called "endpoints" which can be accessed from the internet (i.e., by the client) to perform various operations, such as entering, modifying and viewing data. Since the endpoints are called from the internet via HTTP, they are accessed by the client across the internet, but can

also be accessed by anybody and any program that can make HTTP requests, so the endpoints are vulnerable to all sorts of attacks, and need to be constructed to prevent unauthorized usage. The API server is entrusted with keeping the database data safe on the city's Azure database.

Current TDM Calculator IAM Implementation

TDM is designed from the ground up to include a custom authentication and authorization system. The API server takes care of user registration, login (i.e. authentication) as well as authorization, and does not depend on a third-party authentication system. The main security concerns centered around securing the API server endpoints from attack.
The authentication system includes:

1. The ability for new users to self-register. The registration is open to anyone who accesses the website, and the user's email is confirmed by TDM's email confirmation process and used as their unique username. TDM also records their first and last name. The current implementation is a simple username and password authentication and does not support Multi-Factor Authentication.
2. Registered users can reset their password.
3. Registered users can save and manage their own TDM calculations.

The authorization features in TDM consist of two dimensions:

1. (Role-based Security) Some users are granted an Administrator security role, which allows them to see other users' calculations, set a few properties of calculations, etc. An additional "Security Administrator" role allows a user to grant (or revoke) security roles to other users, delete user accounts, etc.
2. (Per-Calculation Authorization) Calculations are owned by a particular user, and several operations on a calculation are only granted to the owner of a calculation.

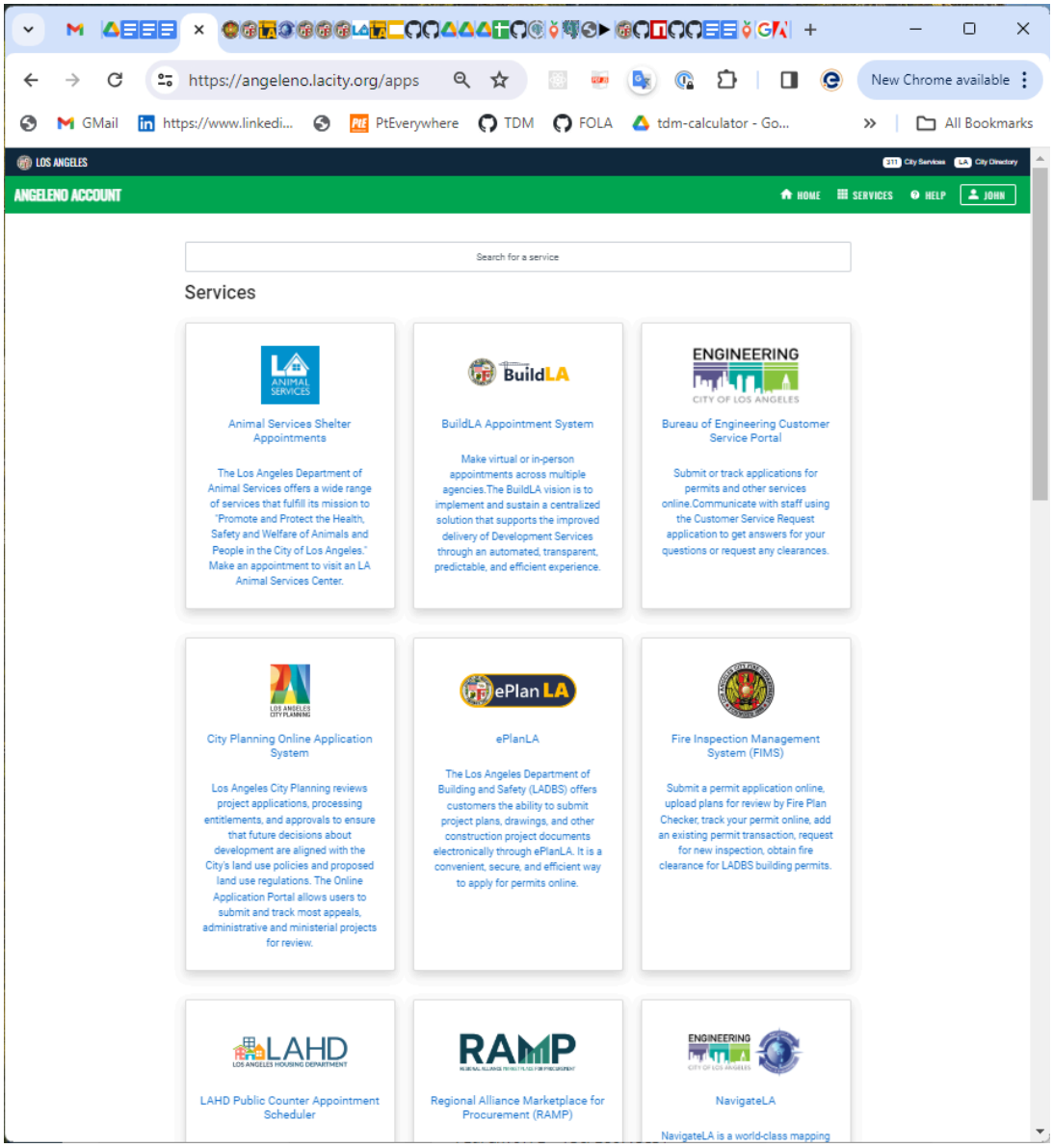
All of this is working pretty well, and ITA Security has been conducting penetration testing and reported a few vulnerabilities with the assistance of Hacker One (https://www.hackerone.com/) and possibly other white-hat hacker organizations. We have been working on addressing any reported vulnerabilities shared with us.

The TDM authentication system is self-contained within the Web API server. When a login is successful, an encrypted authentication and authorization token is returned to the client. When the client submits any subsequent web API request back to the server that requires authentication and/or authorization, the request must include this token in the request, and the server is able to verify that the request is legitimate before servicing the request.

City Required Angeleno IAM Implementation

The ITA Identity Management department has implemented a commercial identity management product from Okta called Workforce Identity Cloud (WIC), dubbed "Angeleno". Their intent is to:

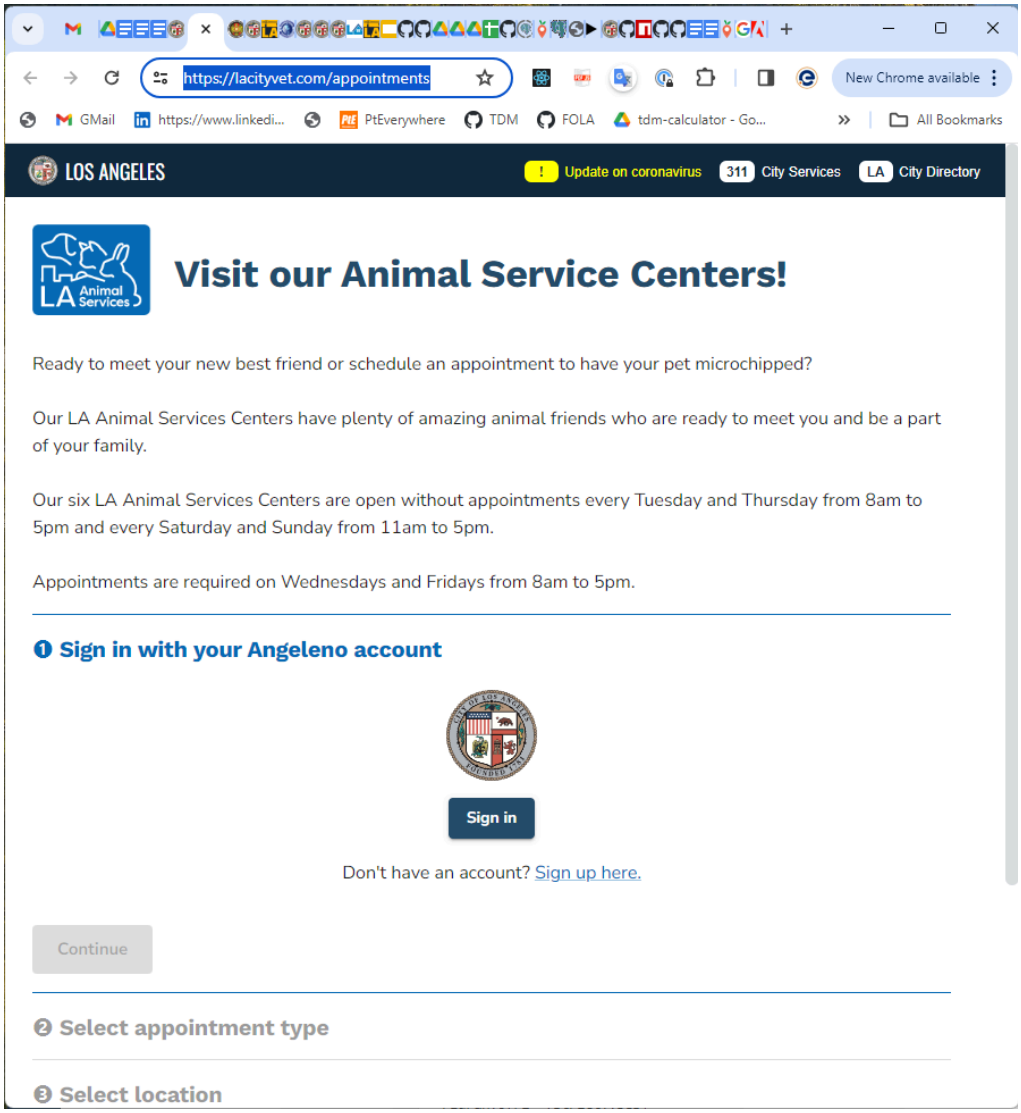
1. Allow external (public) users to register and login to Angeleno, then directly open any of the Angeleno-integrated applications from the Service tab within Angeleno by clicking on the desired application:



When the application is opened, the user will already be logged in. This Single Sign On system allows a user to switch among supported applications without having to re-login.

2. Alternatively, users can go directly to a web address for one of the applications, and click on a button to initiate the Angeleno login as for the LA Animal Services Appointment

Application:



Clicking on the Sign In button will direct them to the Angeleno login page, and then redirect back to the LA Animal Services application when login is complete. If the user then navigates to another integrated application from the same browser, they will already be logged in.

3. Angeleno also allows IDA-IDM to centrally monitor users across all the WIC-integrated applications, and apply consistent authorization policies and practices across the applications, including Multi-Factor Authentication (MFA). Security vulnerabilities related to authentication would be the responsibility of Angeleno.

The Angeleno implementation of WIC does not support any kind of authorization, so integrating TDM security with Angeleno would involve allowing Angeleno to authenticate users, but still depending on TDM's custom authorization implementation to determine what features are granted to a registered user. This also means that TDM would need to have personnel outside

of ITA-IDM (LADOT) that take responsibility for granting and revoking elevated privileges from specific users. Currently, the TDM Calculator application has that same requirement, so using Angeleno does not reduce the burden on LADOT staff.

Implementing authentication with Angeleno and authorization with TDM involves a first login to Angeleno for authentication, followed by a silent second login to TDM for authorization. The steps are:

1. The user selects the Login feature from the TDM client application, which causes a redirect to the Angeleno login page, where they enter their credentials to log in to Angeleno. Once they have successfully logged in, Angeleno redirects back to the client application, and the client application receives some unencrypted information about the user, along with a WIC encrypted token that the client cannot decode.
2. Upon receiving the encrypted token, the TDM application would log in to the TDM authorization system by making a login request to the TDM web API server, based on the user's email address with the WIC encrypted token attached. Note that the email address serves as the common identifier of a user between Angeleno's authentication system and TDM's authorization system, so the TDM login request includes the user's email address to identify which user to authorize. Since the login was done through Angeleno, the login request needs to be authenticated by passing the WIC authentication token to a WIC Custom Authentication/Authorization server. See Notes below
3. Once the TDM login is completed, the TDM web API server returns an encrypted TDM authentication/authorization token (which is similar in design to the WIC token, but includes the additional authorization information) that can be used by the client for subsequent web API server requests, and this token is sufficient for the web API server to verify that the user is authenticated and authorized for the requested operation.

Notes

A. **Angeleno does not currently include a WIC Custom Authentication/Authorization server. If the Custom Auth service is not available, there is no way for the web API server to verify that the user is valid and logged in when implementing the TDM authorization login request.**

B. Based on John Darragh's communications with ITA-IDM, it appears that the Angeleno implementation does not currently include WIC Custom Authentication/Authorization server to secure a web API interface, and without this feature, the TDM API endpoints would no longer be secure.

C. Historically the Okta Authentication product has been for internal teams and the pricing has reflected that. And recently Okta acquired Auth0, which provides authentication for public facing websites and their pricing structure is designed for scaling exponentially.

D. On 2024-01-11 Nicholas Chau provides the following background information: "To give a little background, we are using Okta's workforce identity codebase that was modified by Okta for customer identity. Auth0 was acquired by Okta in 2022, after we had

implemented Angeleno on Okta. While we are planning to migrate to Auth0, Okta is our main platform for now.”

Worst Case Scenario

Without authentication of the TDM authorization login request, anybody on the internet who knows the email of a TDM user would be able to submit a TDM Login web API request, skipping the authentication process entirely, and be logged in to TDM as that user. Moreover, if the user they are impersonating has elevated privileges, they would have unrestricted access through additional web requests to gain the emails and names of all other users, and would be able to delete other users, grant and revoke privileges to other users.