/* ACADEMY SOFTWARE FOUNDATION

# Annual Review for Zero Trust Working Group

Oct 29th 2025

# Zero Trust Working Group

**Brief Description:**
*This working group is looking at ways of improving security in ASWF projects by allowing them to operate in zero trust environments. We will develop educational materials and best practices, collect and document requirements, and consider frameworks that could be useful for ASWF software.*

**TSC Chairperson:**
*Daryll Strauss <dstrauss@movielabs.com>*

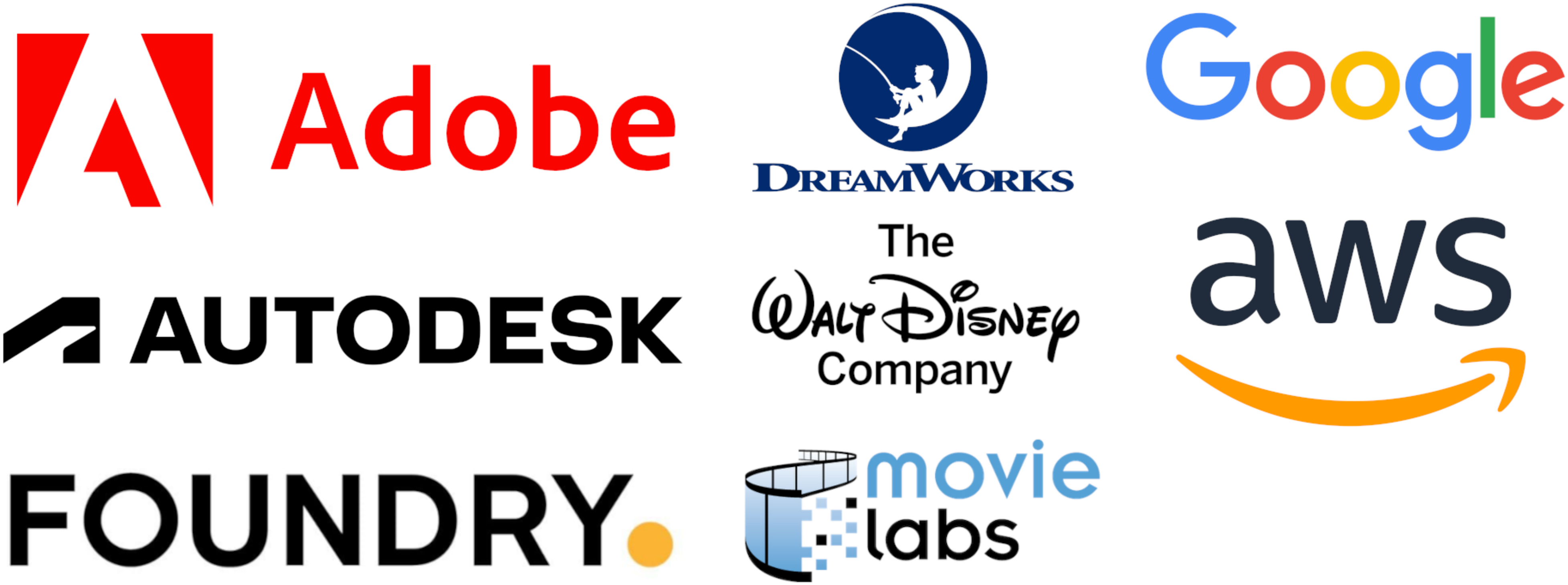
**TSC Members and Affiliations:**
*Jim Helman <jhelman@movielabs.com>*
*Spencer Stephens <sstephens@movielabs.com>*

**Contributed by:**
*Movielabs*

**Key Links:**
**Github:** In process
**Website:**
https://lf-aswf.atlassian.net/wiki/spaces/ZTWG
**Artwork:**
https://lf-aswf.atlassian.net/wiki/spaces/ZTWG/pages/715685906/Artwork
**Mailing list:** zero-trust-wg@lists.aswf.io

**OpenSSF Best Practice Badge URL:** *N/A*

# Participation



# Key Achievements in the past year

- Discussions between software vendors and studios on zero trust requirements
- Discussion of commercial zero trust solutions that could meet some requirements
- Identification of identity management in desktop applications, plugins, and rendering as a key issue for users, vendors, and studios
- Initial design of a proxy identity server for desktops
- In process security reviews of proxy identity server

# Proxy Identity Server

Applications (apps) include GUIs, Plugins, and Rendering. Some have user present some do not.

Apps increasingly need to access network based services (both corporate and SaaS)

Apps should use zero trust security to authenticate users and gain authorized access to services

# Proxy Identity Server

Apps rely on OS login, which authenticates the process not the user, and doesn’t work well for network services.

Network services ask the user to login to each service separately which is awkward for the user and often stores credentials locally.

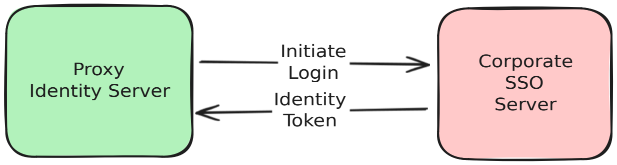
Apps use API Keys which are awkward to setup, long lasting, difficult to track, store credentials, and are difficult to deauthorize.

Doing OAuth flows in apps is awkward and requires each app to interact with a browser and embed a web server.

# Proxy Identity Server

## Step 1: User Login

After logging into their workstation, the user uses the proxy identity server to login to the corporate identity server. The proxy identity server caches the identity token. Note that this is an identity token which does not provide access to any services.



Proxy Identity Server

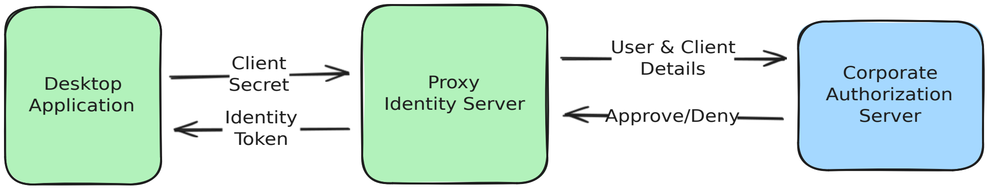
Initiate Login

Identity Token

Corporate SSO Server

## Step 2: Identity Request

When a desktop application requires user authentication it requests the identity from the proxy identity server. Each desktop application provides a client secret that is used to control access to the identity. The proxy identity server can be configured to pass forward the request to a corporate authorization server to allow fine grained control of which applications can request the identity



Desktop Application

Client Secret

Identity Token

Proxy Identity Server

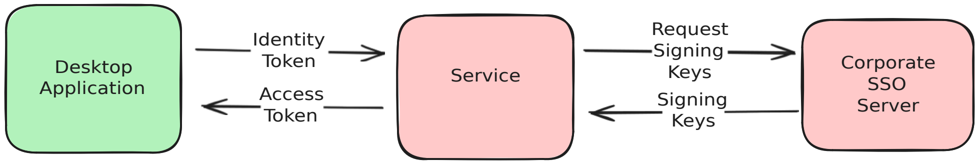
User & Client Details

Approve/Deny

Corporate Authorization Server

## Step 3: Access Request

When the desktop application needs to access a service, they exchange the identity token for the service's access token. The service must use validate the identity token by checking the signature. Once validated the service returns an access token that is used by the desktop application to access the service



Desktop Application

Identity Token

Access Token

Service

Request Signing Keys

Signing Keys

Corporate SSO Server

### Legend

🟩 Applications that run on the desktop

🟦 Applications that run inside the corporate network

🟥 External SaaS applications run by the corporation

# Proxy Identity Server

Apps don’t need to deal with SSO flows, just REST calls

Users only login once for multiple apps

Users can have different versions of their identity (eg. to work on different shows)

Organizations control which apps can access which services.

Identity tokens and access tokens can be short lived

# Areas the project could use help on

- We are moving from early design documents to more formal requirements. Additional review of the designs and requirements would be helpful by security teams and other ASWF software developers.
- We are preparing to do more collaborative development of documents, requirements, and prototypes. We are in the process of setting up a github repository.

# Feedback on working with ASWF

- We appreciate the high quality members we have working with our group which would have been more difficult without ASWF.
- We’re new to running projects under ASWF, so we’ve stumbled on the “right” way to do some things.

# TAC Open Discussion