# Table of Contents

Document Control ................................................................................................................................. 2  
Objective ................................................................................................................................................. 2  
Definition ............................................................................................................................................... 2  
Single-Sign-On ........................................................................................................................................ 2  
Why CAS ............................................................................................................................................... 3  
CAS Internals ........................................................................................................................................ 3  
How CAS works at UCSB ...................................................................................................................... 5  
CAS supported environment ................................................................................................................... 6  
Sign-In Screen ....................................................................................................................................... 6  
Sign-Out Process ..................................................................................................................................... 8  
Single Logout ......................................................................................................................................... 9  
Appendix ............................................................................................................................................... 10
Objective

The objective of this document is to familiarize users at UCSB to CAS (Central Authentication Service). Although targeted for users, this document may explain and demonstrate technical matters relating to authentication and authorization. Feel free to contact us at identity@ucsb.edu in case you need further clarification.

Definition

CAS or Central Authentication Services is an enterprise Single Sign-On solution for web services. Single Sign-On (SSO) means a better user experience when running a multitude of web services, each with its own means of authentication. With an SSO solution, different web services may authenticate to one authoritative source of trust, that the user needs to log into, instead of requiring the end-user to log into each separate service.

CAS provides a friendly open source community that actively supports and contributes to the project. While the project is rooted in the higher-ed open source, it has grown to an international audience spanning Fortune 500 companies and small special-purpose installations. CAS project today is maintained by Apereo. To learn more about CAS please follow the link here https://apereo.github.io/cas/4.2.x/index.html

Single-Sign-On

Single sign on is a session/user authentication process that allows a user to provide his or her credentials once in order to access multiple applications. The single sign-on authenticates the user to access all the applications he or she has been authorized to access. It eliminates future authentication requests when the user switches applications during that particular session.
Web Single sign on works strictly with applications accessed with a web browser. The request to access a web resource is intercepted either by a component in the web server, or by the application itself. Unauthenticated users are diverted to an authentication service and returned only after a successful authentication.

Why CAS

There are many benefits of using CAS. Some of the those benefits are listed below –

- Security: Passwords are not revealed to Applications
- Effort Savings: Applications do not have to invent/re-invent their own authentication infrastructure.
- Convenience: No login for subsequent apps (if SSO is allowed)
- Consistency: One official login page for everyone

Additionally, UCSB will be upgrading its IAM backend infrastructure to UnboundID. This will include the LDAP servers that various applications depend on to authenticate. Utilizing CAS today shields these applications to no longer be concerned with that considerably big upgrade. CAS will play a middleware role between campus applications and authentication.

CAS Internals

The underlying mechanism of how CAS works is extensive. If you are interested to know how it works in detail please visit the link here https://apereo.github.io/cas/4.0.x/protocol/CAS-Protocol.html The following diagram demonstrates a CAS interaction with an application.
### Steps | Description | Example
--- | --- | ---
1 | User navigates via browser to Integration Partner's application integrated to UCSB's Central Authentication Service (CAS). | GET https://travel.ets.ucsb.edu/default.aspx

2 | If user accesses a secured page, CAS Client and Integration Partner application interrogates cookies to evaluate if pre-existing authenticated session exists:  
If valid session exists, user is granted access.  
If session does not exist, user is redirected to CAS server. | [https://cas-d.identity.ucsb.edu:8443/cas/login?service=https%3a%2f%2fesidev2.ets.ucsb.edu%2ftravelc as%2fMenus%2fMainMenu.aspx](https://cas-d.identity.ucsb.edu:8443/cas/login?service=https%3a%2f%2fesidev2.ets.ucsb.edu%2ftravelcas%2fMenus%2fMainMenu.aspx)  
This is set by the CAS Client.

3, 4, 5, 6 | Browser is redirected to login page on CAS.  
The login page is displayed if no valid CAS session is found on browser's cookies (i.e. SSO session). | 

7 | User enters UCSBNetID/Password. |
### How CAS works at UCSB

The following diagram demonstrates CAS implementation at UCSB.
CAS supported environment

CAS can be supported on almost any web application. Today, we have experience integrating CAS with –

- PHP
- Java
- Dot Net
- Cold Fusion
- JavaScript

If you need assistance please contact us at identity@ucsb.edu

Sign-In Screen
At UCSB, you can Sign-In a CAS enabled application using your UCSBNetID/Password combination. The CAS Sign-In page is developed in a responsive design so that it is compatible with your computer browsers and your favorite devices. The Sign-In screen looks like below in a computer browser –

Due to CAS authentication screen’s responsive design, the rendering of the screen is “responsive” based on your device e.g. phone, tablet etc. Below is a screen shot from an IOS device.
Sign-Out Process

The Sign-Out from CAS can be different than regular Sign-out. This is due to SSO (Single Sign On) mechanism that CAS supports. Following assumptions are true on UCSB implementation on CAS –

- At UCSB CAS supports SSO, however, it does not support Single Sign Out (also known as Single Log Out). Single Sign Out means, a user cannot sign-out from one application and assume signed out from all other applications they are currently Signed-In.
- Closing your session by clicking on the “X” icon found on browsers, may not Sign you out. You may need to exit your browser executing its “Exit” method to actually log you out.

The diagram below demonstrates a typical Sign-Out process.
### Steps

<table>
<thead>
<tr>
<th>Steps</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>User clicks on logout button/link located on Integration Partner's Application</td>
</tr>
<tr>
<td>2</td>
<td>Application clears the user's session on the local application.</td>
</tr>
<tr>
<td>3</td>
<td>Application forwards user's browser to the CAS Logout page.</td>
</tr>
<tr>
<td>4</td>
<td>CAS clears the user's session on the CAS server. The user's session is terminated on the Integration Partner's Application and CAS. You can choose to configure the logout step, to return user's browser to your application, after the CAS logout page.</td>
</tr>
</tbody>
</table>

### Single Logout (SLO)

UCSB CAS does not support Single Logout. Here is some rationale why it is so.

- University of British Columbia (UBC) has a [document](#) outlining the security issues for implementing Single Log-Out (SLO) in a distributed web environment.
- The consortium of Swiss Universities have [documented](#) security concerns for implementation of SLO.
Appendix

References

1. https://www.apereo.org/projects/cas
2. https://apereo.github.io
3. https://confluence.id.ubc.ca:8443/display/UBCCAS/Home