SINGLE SIGN-ON

& Authentication Services

TABLE OF CONTENTS

INTRODUCTION

Most universities use Single Sign-On as a method of one-time access for users. These users authenticate once and then gain access to resources of multiple software systems.

Universities must also maintain a single source of authentication information for all users on campus. Multiple users validate this information against a common repository.

Tk20 software is designed to meet these needs.

The architecture is flexible for integration with external web portals. The software is designed to support authentication against different authentication sources. It can also be configured to authenticate various types of users against different authentication sources on a single deployment.

ARCHITECTURE

The Tk20 system uses an n-tier, Java 2 Enterprise Edition Architecture. It is built using a proven, open-source operating system and software components. It is written in Java, and comprised of three tiers: a web tier, middle tier, and database tier. Tiers function independently with their own interfaces for communication. They can be physically located in the same server, or have multiple servers allocated to each one (depending on the load and configuration).

One component deployed as part of the Tk20 installation in the application server (JBOSS) is the Pluggable Authentication Module (PAM). PAM is the heart and soul of this architecture.

Different types of users exist in the university environment, such as Student, Faculty, Cooperating teachers, Program coordinators, etc. In the Tk20 system, each user is assigned a role.

Possible Campus Situations

- Some or all users have campus accounts and are required to access campus services with a web portal
- The campus maintains only one source of authentication information and all clients needing access must access the central repository.
- There is a set of users needing access to the Tk20 system, but their authentication information is not centrally managed by campus.

Tk20 software is designed to meet all of the above needs through configuration management.

In the Tk20 system, it is possible to specify the authentication source for each user type. For example, it is possible to direct students and faculty to an external LDAP (Lightweight Directory Access Protocol) server for authentication and direct cooperating teachers to a Tk20 local database for authentication. Depending on the configuration data, PAM initiates the implementation class and carries out remote authentication against any authentication source.

CENTRAL SERVER AUTHENTICATION

For universities maintaining a central repository of authentication information and needing clients to access the repository, Tk20 is designed to communicate with external authentication sources to look up the authentication information.

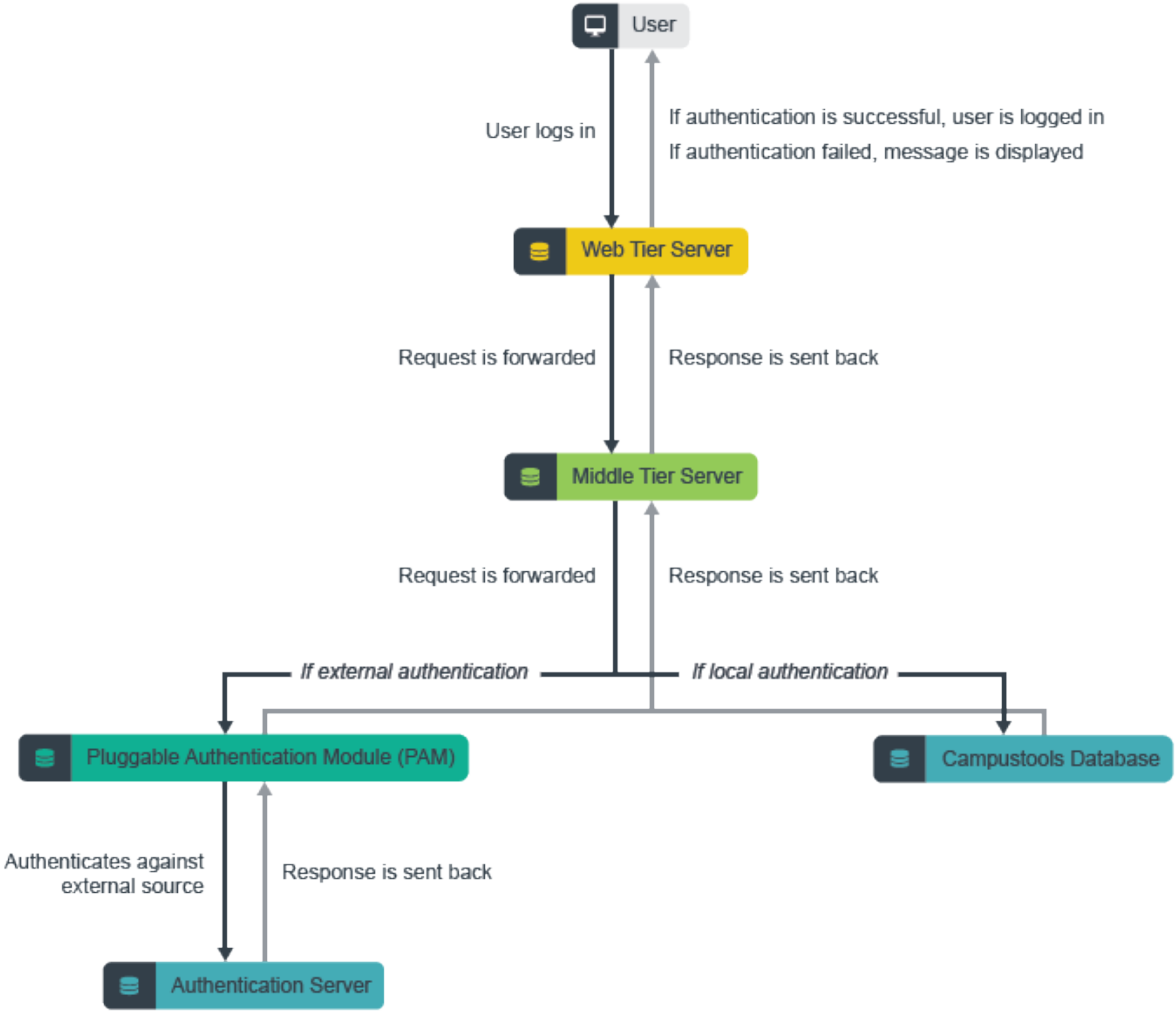
A few examples of external authentication sources that Tk20 software currently authenticates users against include:

- LDAP servers
- Active Directory
- Radius Server
- Central Authentication Services (CAS)
- Banner Authentication

The architecture is flexible to enhance this ability against any other authentication source. During remote authentication, all Tk20 users access the system with the Tk20 URL.

Depending on the type of user trying to access the system and the configuration data, the PAM decides to which authentication source to connect, in order to authenticate the user.

In the case of a failed authentication, a message displays indicating a failed login. In the case of a successful authentication, the system runs the authorization checks on the user before granting access. This is depicted in the diagram below:



ACTIVE DIRECTORY/LDAP AUTHENTICATION

Project Plan

1. **Requirement gathering** – Projects begin with an initial meeting between Tk20 Engineering Services and university personnel to clearly communicate requirements and exchange any information necessary to begin the integration. To configure your system to connect and authenticate users against your Active Directory/LDAP server, Tk20 requires cooperation and information from the institution.

   Configuration for ports and firewall – The test and production instances should be able to connect to the Active Directory/LDAP server for authentication.

   > *The test and production instance should connect to your Active Directory/LDAP server through the firewall for authentication testing.*

   Connection parameters – The university must provide Tk20 with the connection parameters to the Active directory/LDAP server. The system initiates the connection request with these parameters. (Further details are mentioned in the Integration Requirements Checklist found on page 9.)

   Configuration parameters – The University must provide Tk20 with the parameters to configure the search base and search filter. (Further details are mentioned in the "Integration Requirements checklist.")

   Test account configuration – The university must provide Tk20 with test account credentials for authentication testing throughout the duration of the project.

2. **Configuration** – Engineering Services configures the Tk20 system according to the configuration and connection parameters supplied by the university.

3. **Testing** – Tk20 will try to simulate the login process for different users in the system. Tk20 will create an account in the system with the same username as the test account on Active Directory and will follow the following test cases:

	Test Case	Success Criteria
1	Users with inactive Tk20 accounts logs in with correct credentials	**Message Displays:** This user account is inactive
2	User with unpaid Tk20 account logs in with correct credentials	**Message Displays:** Your login was unsuccessful. Your student account has not been activated in the system
3	User with expired Tk20 account logs in with correct credentials	**Message Displays:** The user account is inactive
4	User logs in with correct credentials but does not exist in Tk20 system	**Message Displays:** Invalid username/password
5	User logs in with incorrect username	**Message Displays:** Invalid username/password
6	User logs in with correct username; incorrect password	**Message Displays:** Use remote password
7	User logs in with correct login credentials; has active/paid account	**Message Displays:** Successful login
8	User with active/paid account logs in to Tk20 and is authenticated against external	**Message Displays:** Successful login

9	User with active/paid account logs into Tk20 and is authenticated locally against the Tk20 database	**Message Displays:** Successful login
10	User logs into Tk20 and is authenticated for the first time	User is presented with a password change screen
11	Superadmin Login Test	**Message displays:** Successful login
12	Faculty log in via link in the notification email	User is directed to the appropriate login page, based on their portal

Possible Campus Situations

The built-in support for Active Directory/LDAP authentication process is explained below.

1. **Initial bind** –The Tk20 system tries connecting to the Active Directory/LDAP server using the connection parameters provided. If able to connect, it continues to Step 2 after the successful bind.

   If the system is unable to connect to the Active Directory/LDAP server, or if the bind is unsuccessful, the authentication process is terminated and the user is unable to log in to the system.

2. **User search** – The system enters this stage only if the initial bind is successful. Here, the system searches for the user created in the Active Directory/LDAP server within the configured search base. If it finds the user, it proceeds to Step 3. If not, authentication fails and the user is unable to log in to the system.

3. **Authentication bind** – The system reaches this stage only if the initial bind is successful and the system successfully finds the user in the Active Directory/LDAP server. Now the system tries authenticating the user with the entered username and password against the Active Directory/LDAP server.

If authentication fails, an appropriate message displays to the user. The user should contact the local Tk20 administration on campus for further assistance. If authentication is successful, the system forwards the request to the authorization module.

> *Note: Tk20 software does not support failover authentication or multiple search bases. In the case of failed authentication, the campus help desk is your first point of contact.*

Integration Requirements Checklist

Please provide the following parameters for integration of Tk20 software against your Active Directory/LDAP server for user authentication.

1. Active Directory/LDAP server.

2. Port of communication. (If the communication must occur over ssl, please provide the certificate in der format, so that Tk20 can import it. If the cert is signed by multiple CAs, please provide the certs in the entire certificate chain.)

3. Search/user base

4. Bind account DN. (Please provide full DN of account. For example, CN=Tk20 LDAP, CN=Users, DC=xxx, DC=xxx.)

5. Bind account password.

6. User attribute for searching. (Filter is created based on this attribute.)

7. Attributes that may be requested to be returned after the initial bind. (Tk20 uses these to get the DN for the user.)

8. Test account username.

9. Test account password.

SINGLE SIGN-ON

Single Sign-On provides users with access to multiple environments with a single, secure password. Most universities maintain their web portals; on-campus users would use single authentication information to log in. Once logged in, they can access various websites or services, such as Tk20.

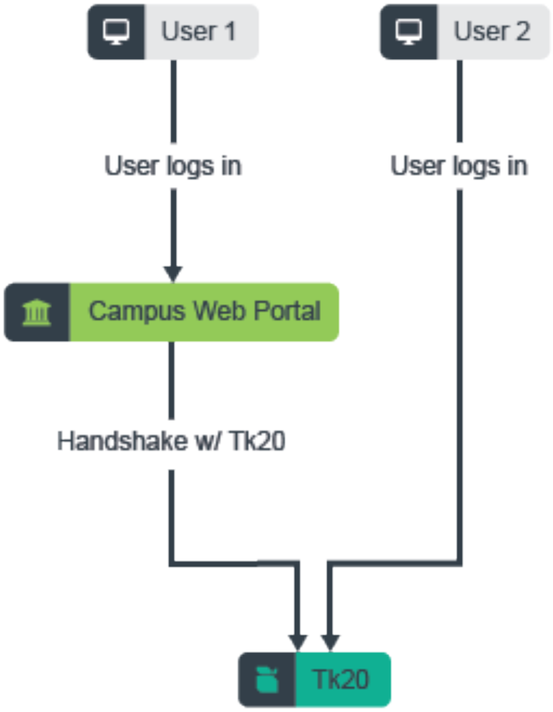
Typically, in this scenario, once the user is authenticated via the portal, a request forwards to Tk20. When the software receives the request, a secure handshake with the web portal ensures the request is legitimate.

After the request is validated, information pertaining to the user attempting access to Tk20 is exchanged. The software ensures that the user trying to access Tk20 is a valid Tk20 user.

After passing the checks, the user is immediately directed to the Tk20 account home page in the application. In this case, the user does not see the Tk20 login page because the authentication only happens once when the user logs in to the web portal on campus. Tk20 merely integrates with the web portal, ensures the proper authorization and allows the user into the Tk20 system.

The Tk20 system is designed to allow access to users who do not have access to the campus portal, but do need access to Tk20. In such cases, the authentication information of such users is maintained locally in the Tk20 database.

The sequence of events during this process is depicted below:



INTEGRATED WITH SHIBBOLETH

Tk20 Engineering Services works with the university to integrate the software with the on-campus Shibboleth web Single Sign-On authentication framework. After the successful user authentication Idp needs to pass the username of the user to SP for authorization.

Information the Customer Provides for Integration

- **Workflow:** Describe the workflow via which the student/faculty will be accessing the Tk20 application.
- **Logout URL:** This is the landing URL for the users once they logout from Tk20. Please note this is applicable only for users that get authenticated via Shibboleth.
- **Idp Metadata:** Please provide the metadata from Idp provider.
- **EntityID :** Please provide entityID to be configured in SP shibboleth2.xml

- **Username Attribute:** Please share the attribute that will have the username information, as we will need to configure attribute-map.xml based on that information.
  - The **username** should same as what comes in through the extracts under username field or what is being released as username field in the extracts.
- **Test Credentials:** Please share ogin test credentials to perform Shibboleth authentication testing. We can disable the same after testing out the setup.
  - Please share the **test credentials** on : projects@tk20.com
- We can start installing Shibboleth 2.4.3 on https://xyz.tk20.com and share Metadata once we get the requested info.
- ***Note for the IT team:*** *For set ups with Shibboleth it is recommended that post log out the user closes the browser to end the session completely.*

Information that Tk20 Engineering Services Provides

Tk20 provides the URL to acquire SP Metadata for both the production and test instance.

INTEGRATED WITH CAS

Tk20 Engineering Services works with the university to integrate the software with your on-campus Central Authentication Services (CAS) web Single Sign-On authentication framework.

The Tk20 integration against CAS is based on authenticating the users manually with CAS Java Objects. The servlet that CAS returns to (service URL processor servlet) expects to receive a ticket parameter. If this servlet is accessed by the user directly, the system redirects the user to the CAS login page. This servlet retrieves the username of the user attempting to access the resource via the service ticket validator.

Information the Customer Provides for Integration

- **Portal login URL** – If users need to be authenticated against the portal, they should log in to Tk20 with the Tk20 login page. They will then be redirected to the portal log in URL for authentication.

- **Logout URL** – This is the landing URL for users when they logout from Tk20. Please note: This is only applicable for users authenticated via CAS. Users authenticated against the local Tk20 database are forwarded to the default Tk20 logout URL.

- **Users for CAS authentication** – The university needs to identify the sets of users that are authenticated with CAS (faculty, students, etc.). Engineering Services configures the system with those users marked for authentication with CAS.

- **CAS validator URL** – The university needs to provide the CAS validator URL to use when validating the tickets.

Information that Tk20 Engineering Services Provides

Tk20 provides the Service URL for both the testing and production instance.

INTEGRATED WITH COSIGN

Tk20 Engineering Services works with the university to integrate the software with the on-campus Cosign web Single Sign-On authentication framework.

Tk20 implements the Cosign integration based on creating and configuring the Apache filter to access a protected resource. The AJP connector must be configured so the environment variables can pass from Apache to the servlet container (Tomcat). The processor servlet reads the username with the request variable.

Information the Customer Provides for Integration

- **Portal login URL** – If users need to be authenticated against the portal, they should log in to Tk20 with the Tk20 login page. They will then be redirected to the portal log in URL for authentication.

- **Logout URL** – The landing URL for users when they logout from Tk20. Please note—this is only applicable for users authenticated via Cosign. Users authenticated against the local Tk20 database are forwarded to the default Tk20 logout URL.

- **Users for Cosign authentication** – The university needs to identify the sets of users that are authenticated with Cosign (faculty, students, etc.). Engineering Services configures the system with those users marked for authentication with Cosign.

- **Cosign host name** – The Cosign host name is used to authenticate users.

Information that Tk20 Engineering Services Provides

Tk20 provides the Service URL for both the testing and production instance.

TOKEN-BASED AUTHENTICATION

Tk20 Engineering Services works with the university to integrate the software with the on-campus portal using Token-Based authentication. The Single Sign-On framework provides a secured way for users to access the Tk20 System.

Information the Customer Provides for Integration

1. SHA1/SHA2* hashed username
2. Timestamp

Tk20 would like your portal to post the SHA1/SHA2* hashed username and current time in milliseconds (GMT format) to a Tk20 system URL that will be shared with you. Tk20 will then use the hashed username to determine the user trying to access Tk20 system and the timestamp for validating the authenticity of the request and then forwards the request to authorization module.

*Tk20 Version 8.2.0 version supports SHA2.

Information that Tk20 Engineering Services Provides

Tk20 provides the Service URL for both the testing and production instance.