# Implementing CAS

Adam Rybicki

2009 Jasig Conference, Dallas, TX

March 1, 2009

1. Introduction
2. Problems CAS solves
3. CAS protocol
4. Building from sources
5. Customizing the presentation
6. Authentication handlers
7. CAS-enabling uPortal
8. CAS-enabling Tomcat

# Introduction

Who are we?

What is CAS?

Brief history of CAS

# Adam’s Involvement with CAS

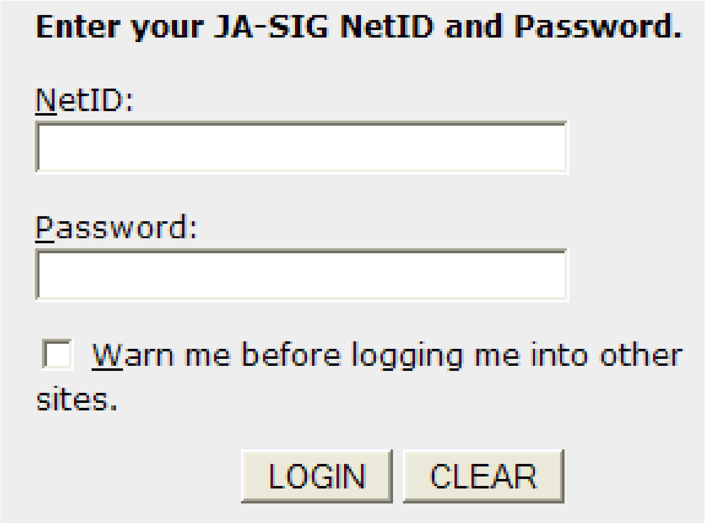
- Got interested.
- Worked with several clients helping them to CASify their applications.
- Asked many questions of the CAS mail list
- Wrote a CAS self-study guide for Unicon developers. (https://confluence.unicon.net/confluence/x/XgZi) (authentication required)
- Answered some questions on the CAS list.
- Currently working with Unicon clients on CAS server implementations and CAS-enabling their Web applications.

# Introductions

- Who are you?
  - Name
  - Institution
  - Role
  - Why interested in CAS?

# What is CAS?

- CAS is enterprise single-sign-on for the Web.
  - Free
  - Open source
  - Server implemented in Java
  - Clients implemented in a plethora of languages



**Enter your JA-SIG NetID and Password.**

□ Warn me before logging me into other sites.

# Some of the people involved as the project has evolved

- Scott Battaglia
- Shawn Bayern
- Susan Bramhall
- Marc-Antoine Garrigue
- Howard Gilbert
- Dmitriy Kopylenko
- Arnaud Lesueur
- Drew Mazurek
- Benn Oshrin
- Jan Van der Velpen (Velpi)

# Many CAS deployers

- Appian Corporation
- Athabasca University
- Azusa Pacific University
- BCcampus
- California Polytechnic Institute
- California State University, Chico
- Campus Crusade for Christ
- Case Western Reserve University
- Columbia
- Employers Direct
- GET-INT
- Hong Kong University of Science and Technology
- Indiana
- Karlstad University, Sweden
- La Voz de Galicia, Spain
- Memorial University of Newfoundland
- Nagoya University
- NHMCCD
- Northern Arizona University
- Plymouth State University (used with SunGardHE Luminis)
- Roskilde University
- Rutgers, The State University of New Jersey
- SunGard HE Luminis
- Simon Fraser University (Vancouver, B.C.)
- Suffield Academy
- Tollpost Globe AS

# … and more

- Universita degli Studi di Parma
- Universite de Bourgogne - France
- Universite de La Rochelle, France
- Universite de Pau et des Pays de l'Adour, France
- University of Nancy 1, France
- Universite Nancy 2, France
- Universite Pantheon Sorbonne
- Universiteit van Amsterdam
- University of Bristol, England
- University of California Merced
- University of California, Riverside
- University of Crete, Greece
- University of Delaware
- University of Geneva
- University of Hawaii
- University of New Mexico
- University of Rennes1
- University of Technology, Sydney
- Uppsala University
- Valtech
- Virginia Tech
- Yale University

- And likely more not well-enumerated…

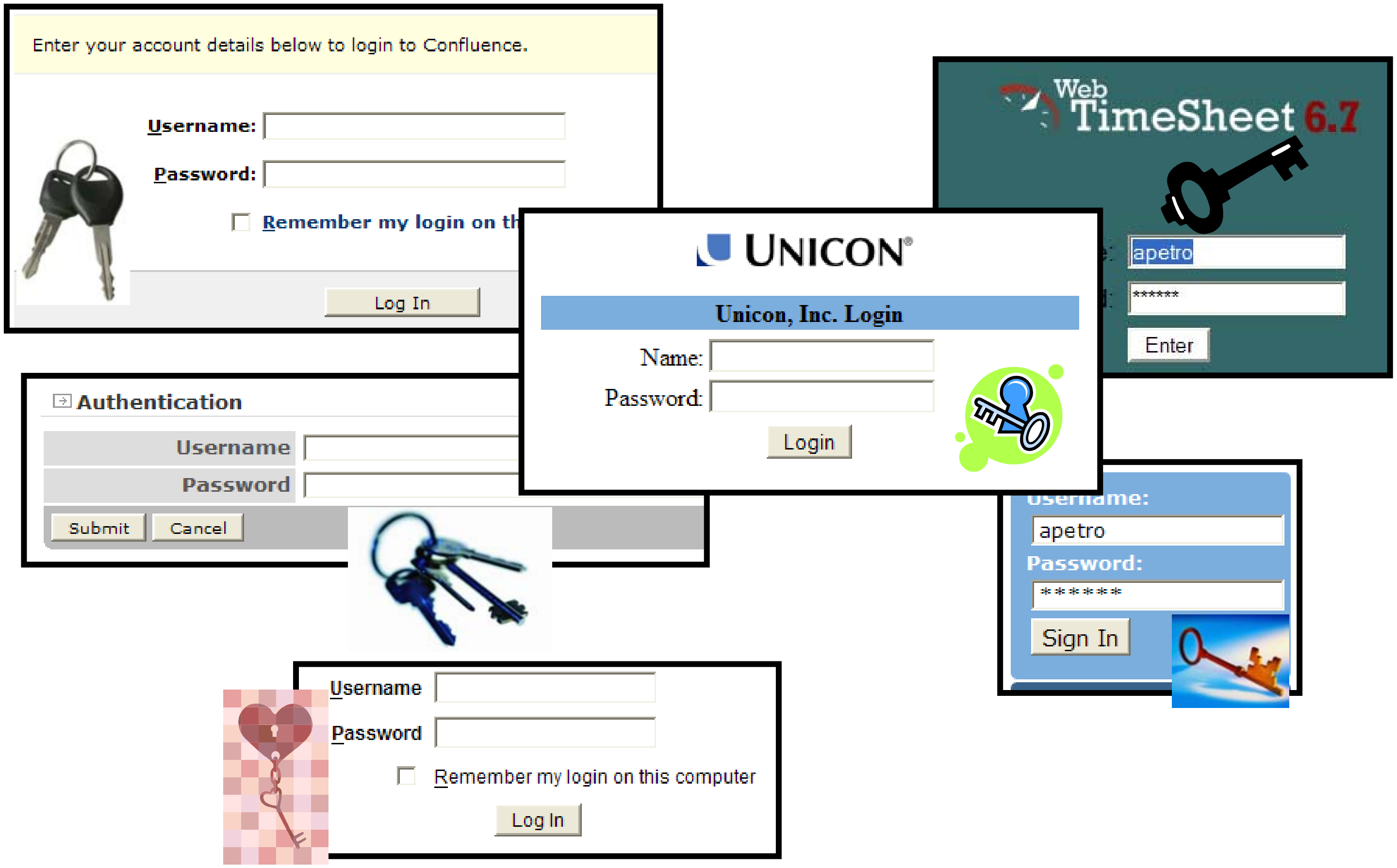
# Problems CAS solves

Disparate credentials and name spaces

Too many Web applications dealing with credentials

CAS creates new challenges, too

# Multi-sign-on for the Web



Enter your account details below to login to Confluence.

Username:

Password:

□ Remember my login on th

Log In

Authentication

Username

Password

Submit Cancel

Web TimeSheet 6.7

apetro

******

Enter

UNICON®

Unicon, Inc. Login

Name:

Password:

Login

Username:

apetro

Password:

******

Sign In

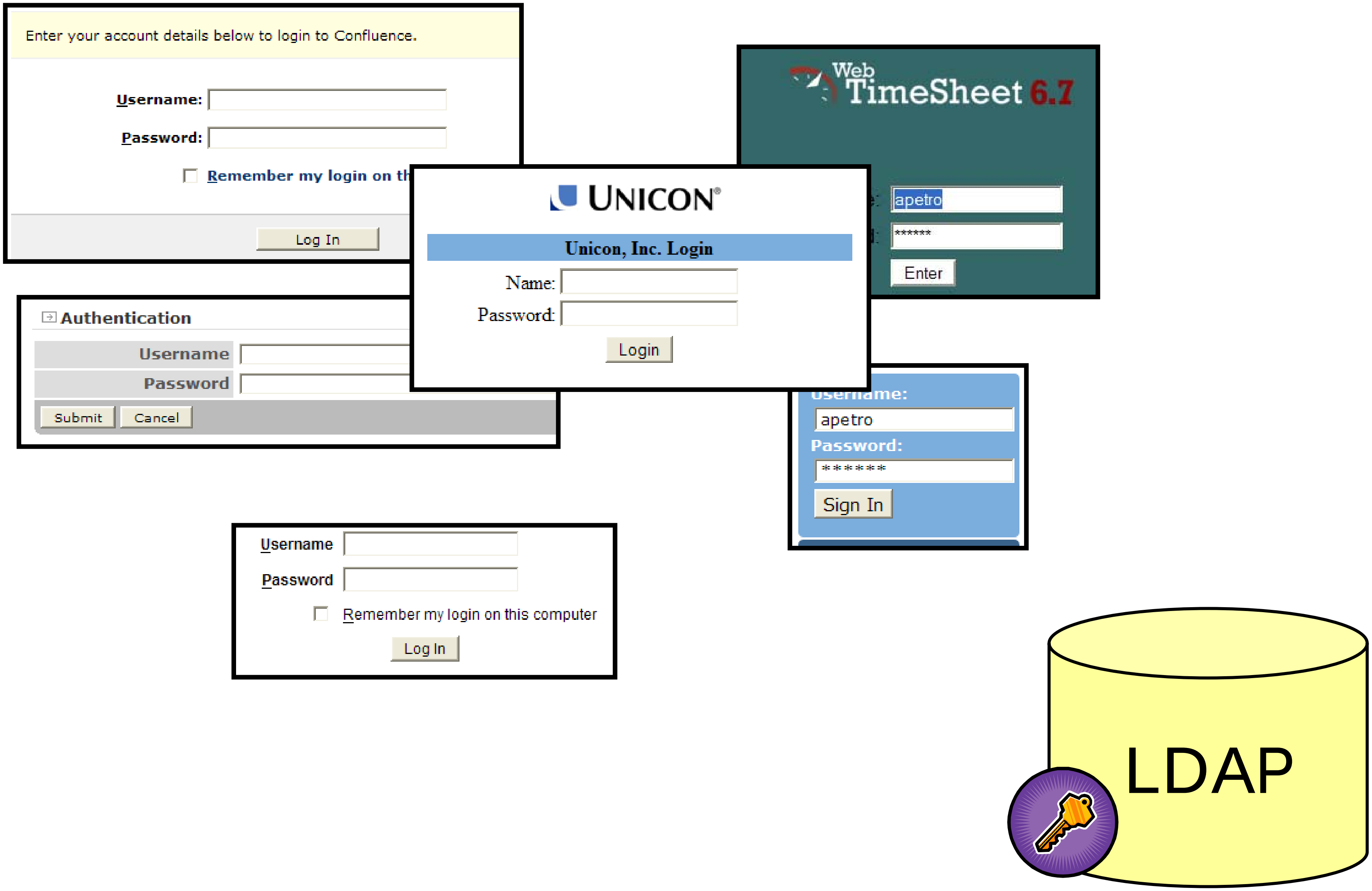
Username

Password

□ Remember my login on this computer

Log In

# At least with one username/password?



Enter your account details below to login to Confluence.

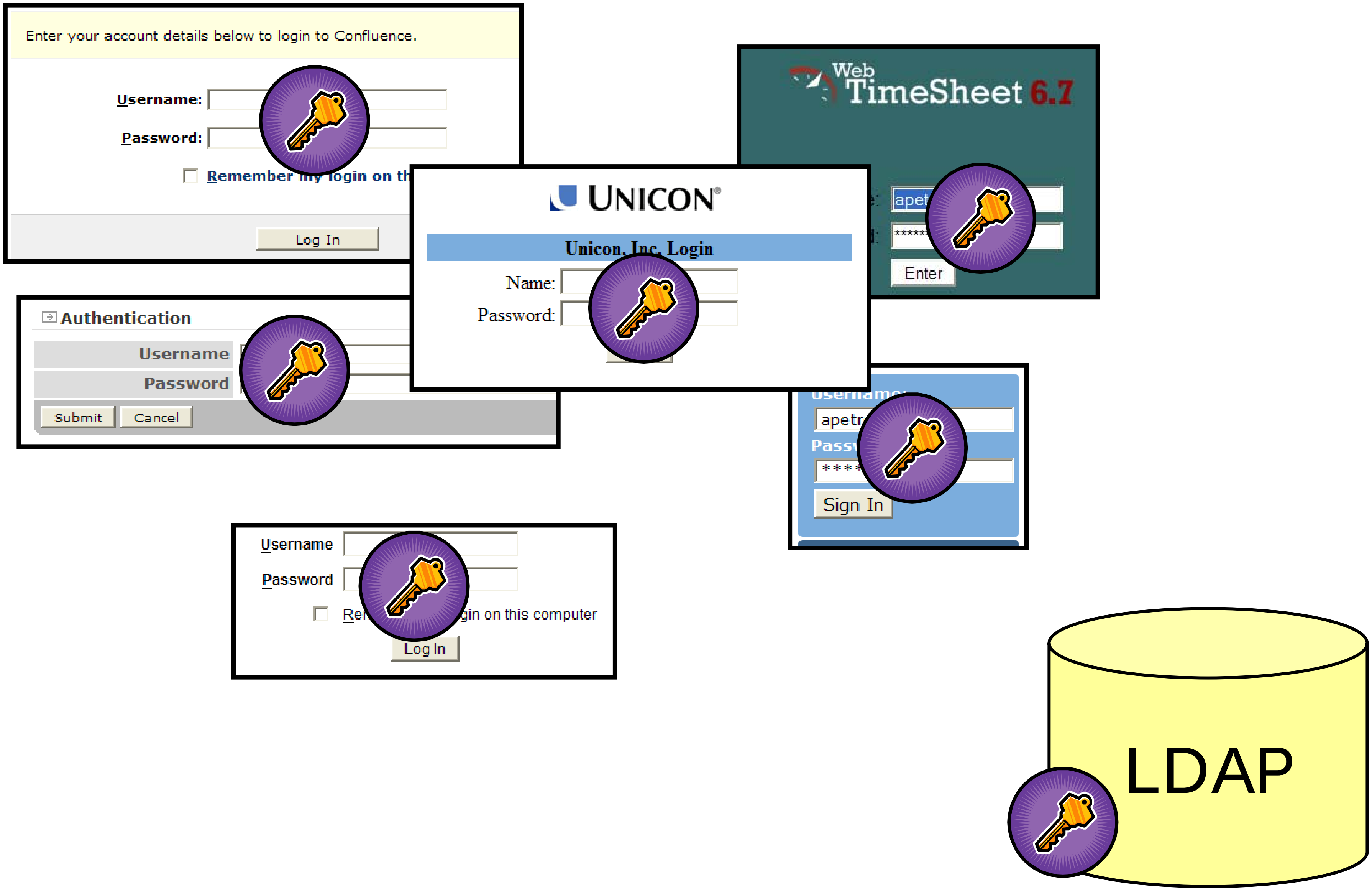
apetro

******

apetro

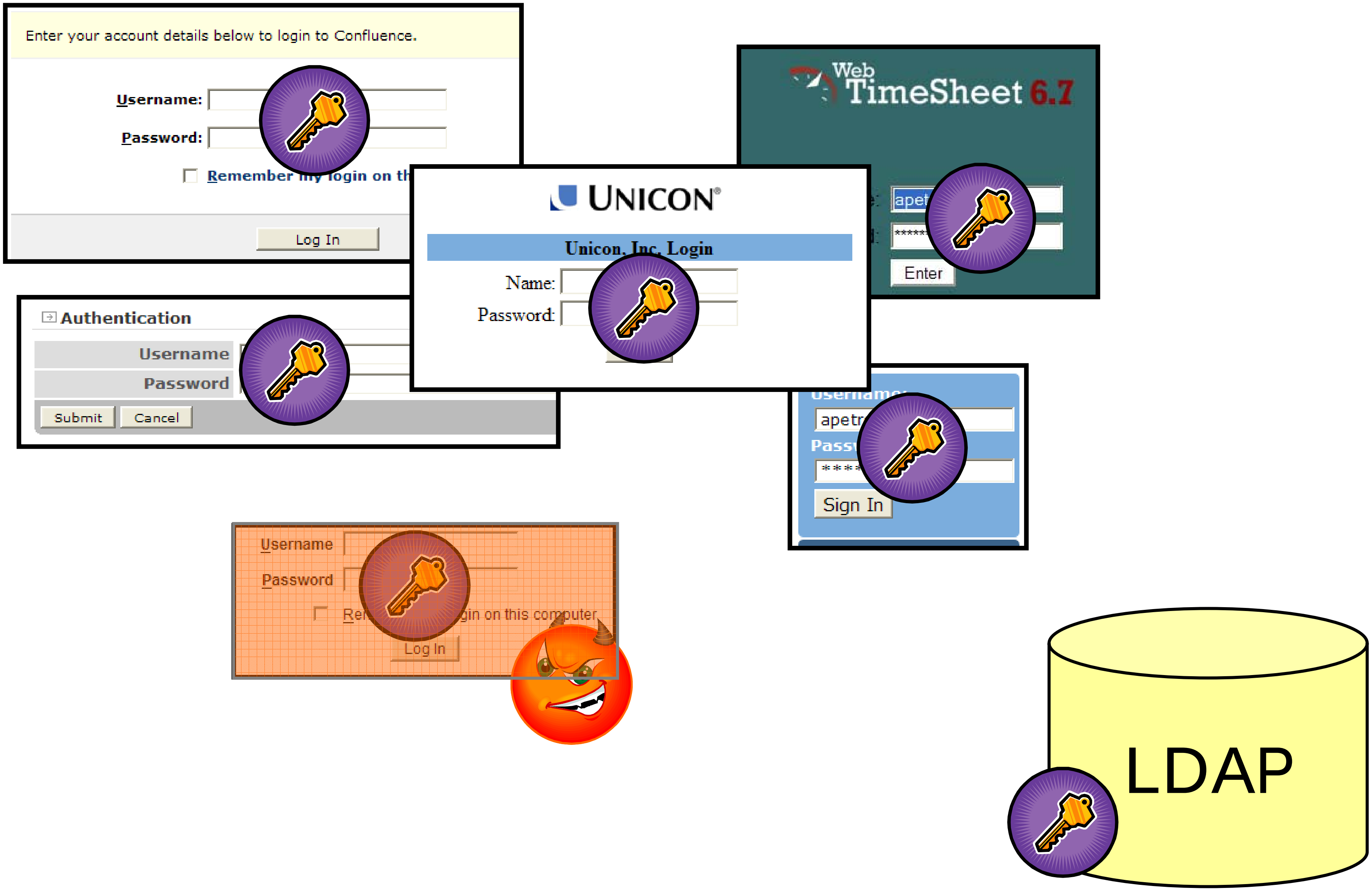
******

# All applications touch passwords

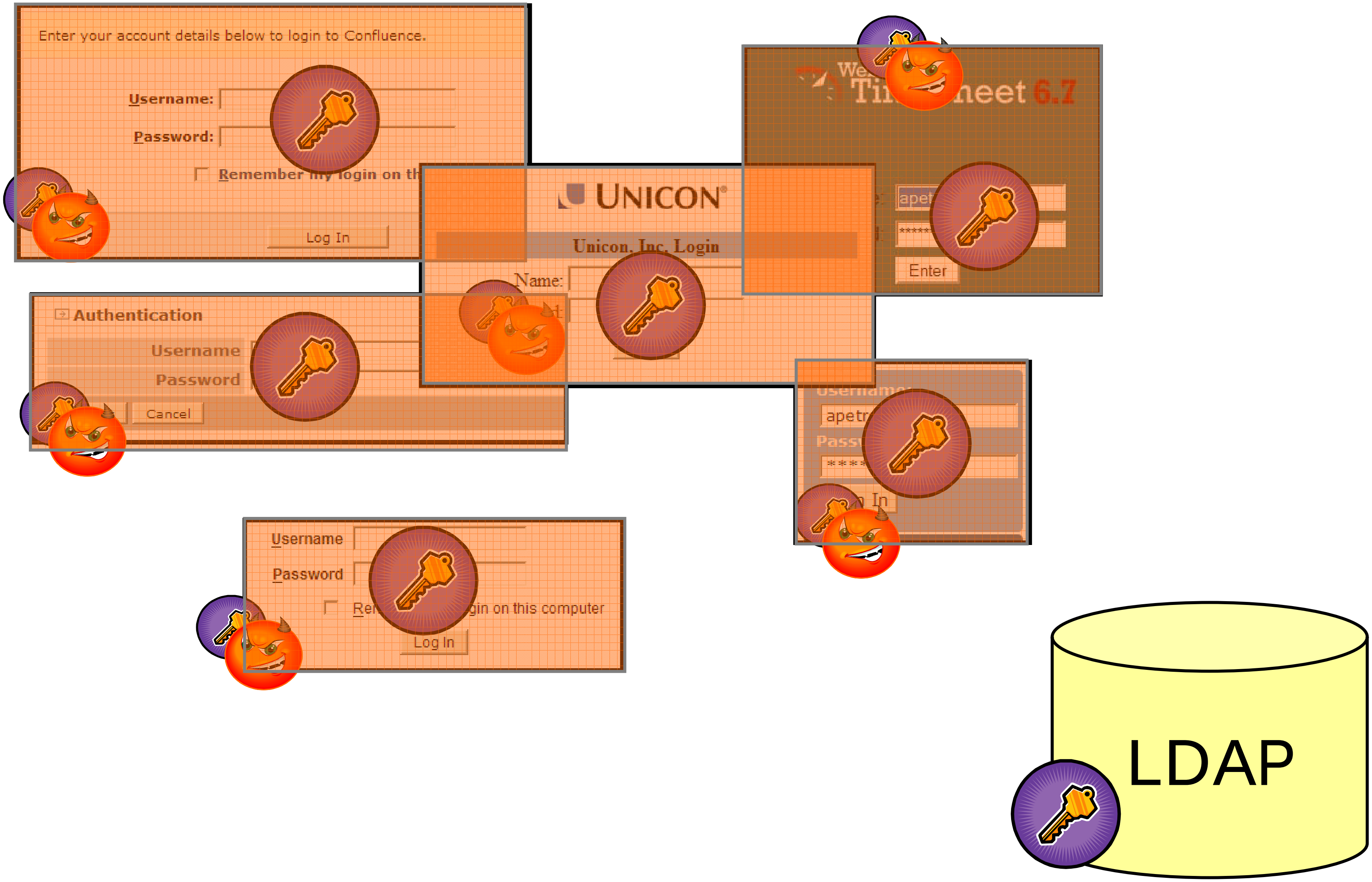


Enter your account details below to login to Confluence.

# Any compromise leaks primary credentials



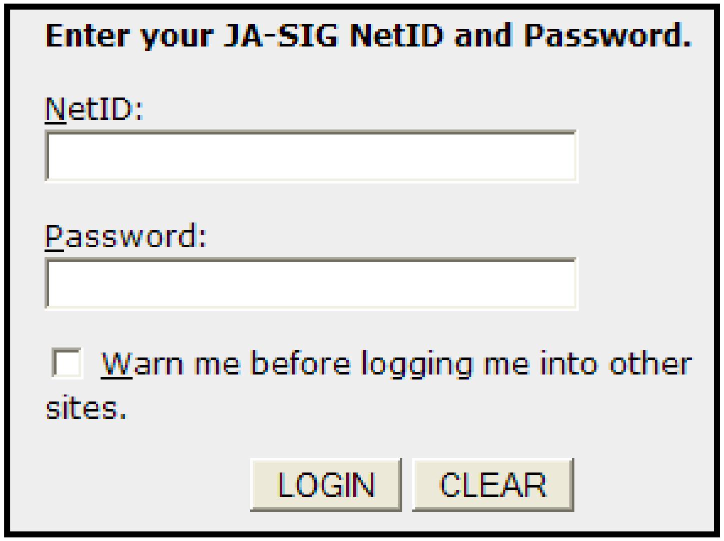
# Adversary then can run wild



Enter your account details below to login to Confluence.

# What to do about this?

- What if there were only one login form, only one application trusted to touch primary credentials?



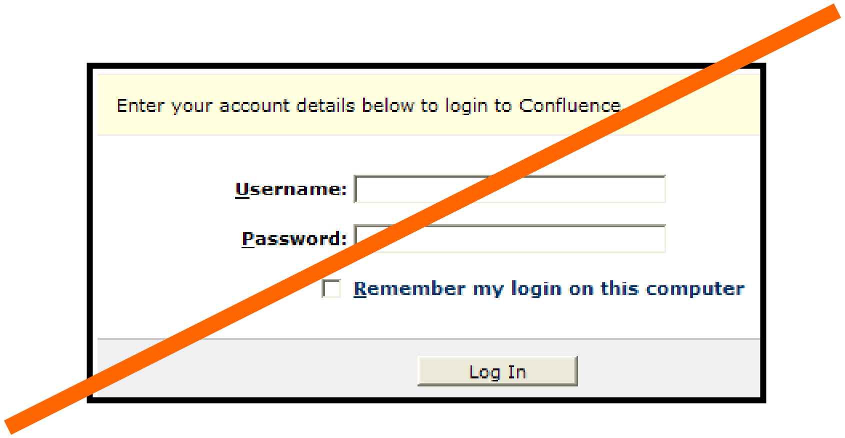
**Enter your JA-SIG NetID and Password.**

NetID:

Password:

□ Warn me before logging me into other sites.

# Delete your login forms.



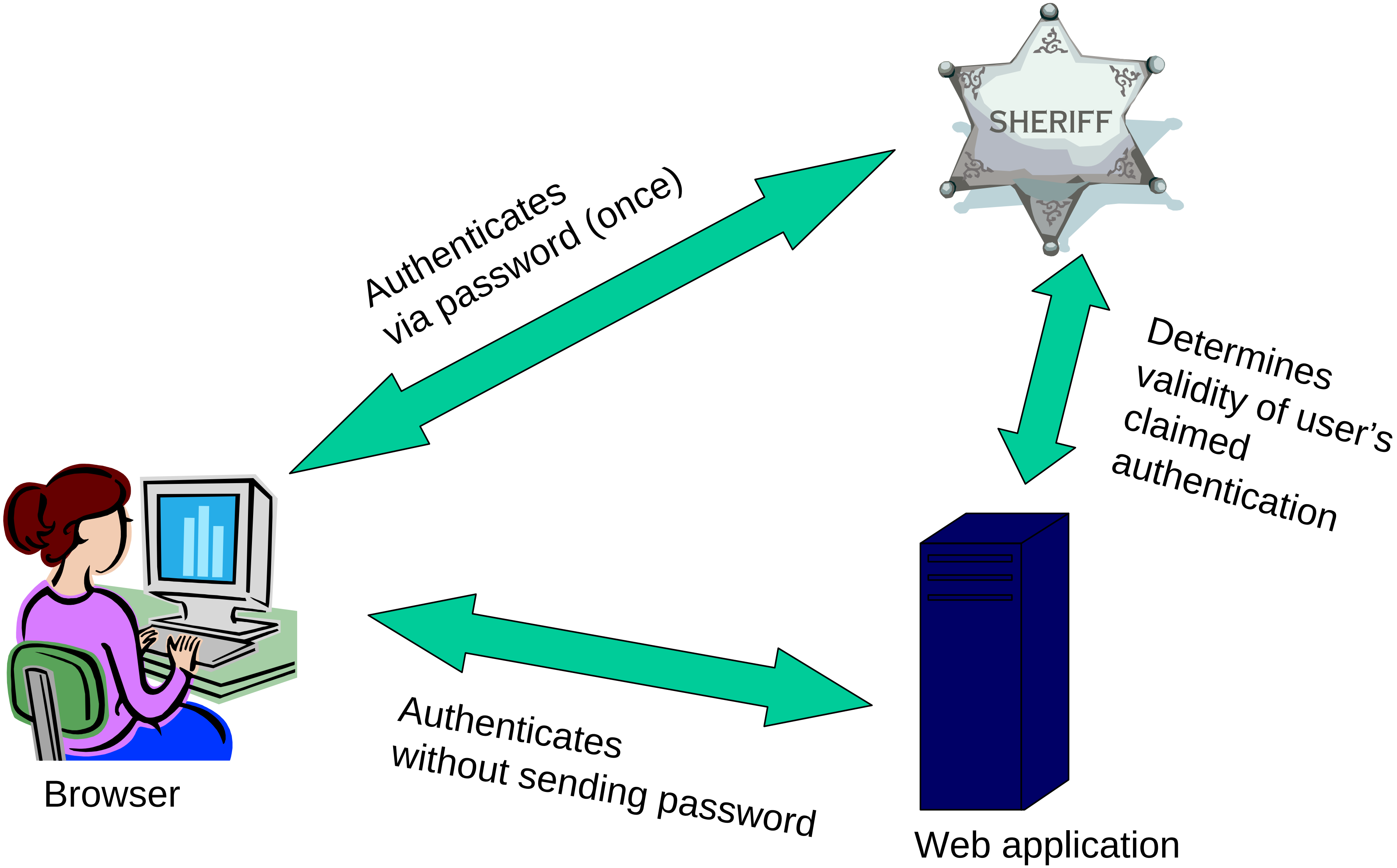
Enter your account details below to login to Confluence

Username:

Password:

□ Remember my login on this computer

# CAS in a nutshell



Authenticates
via password (once)

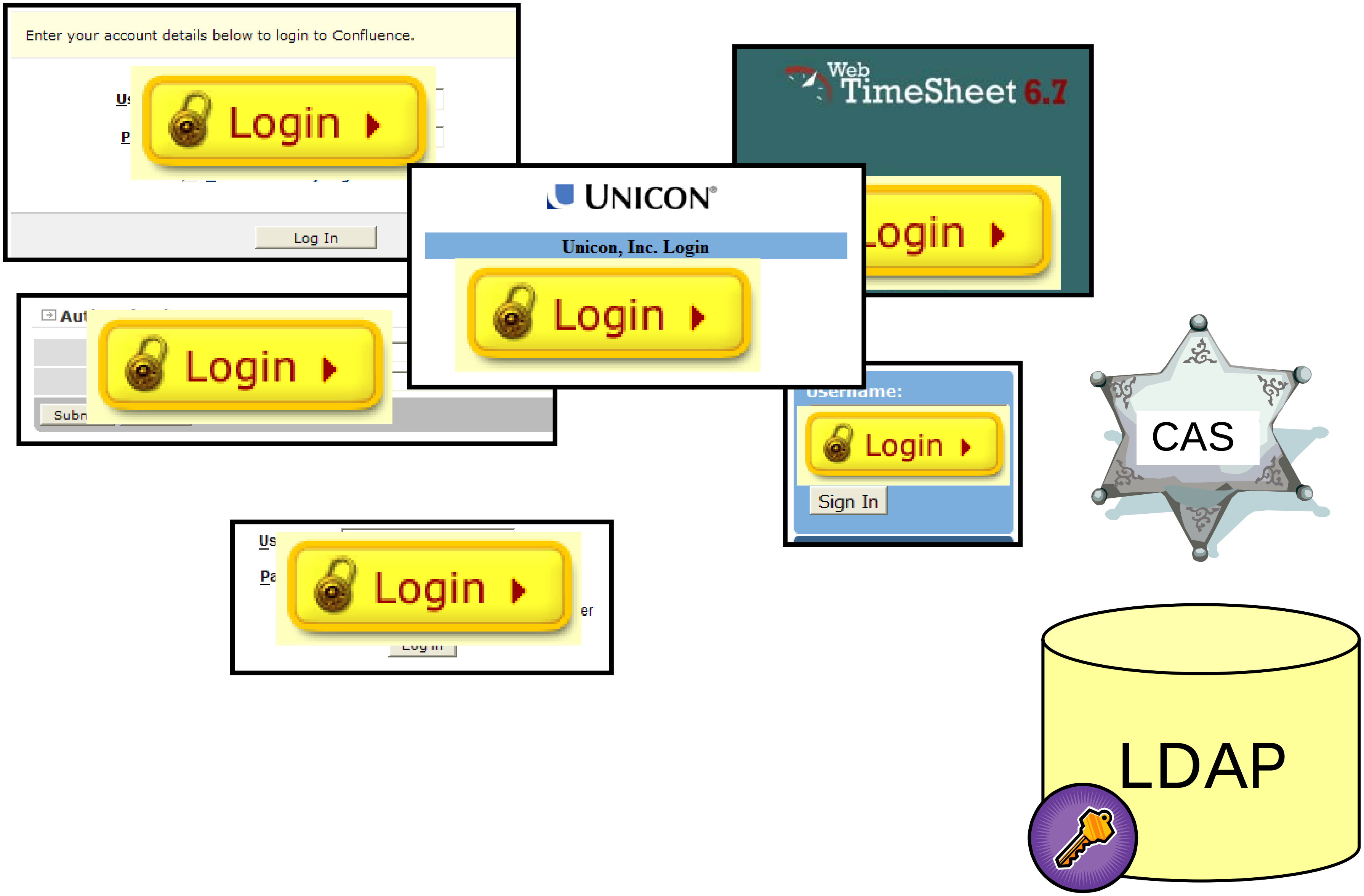
Determines
validity of user's
claimed
authentication

Browser

Authenticates
without sending password

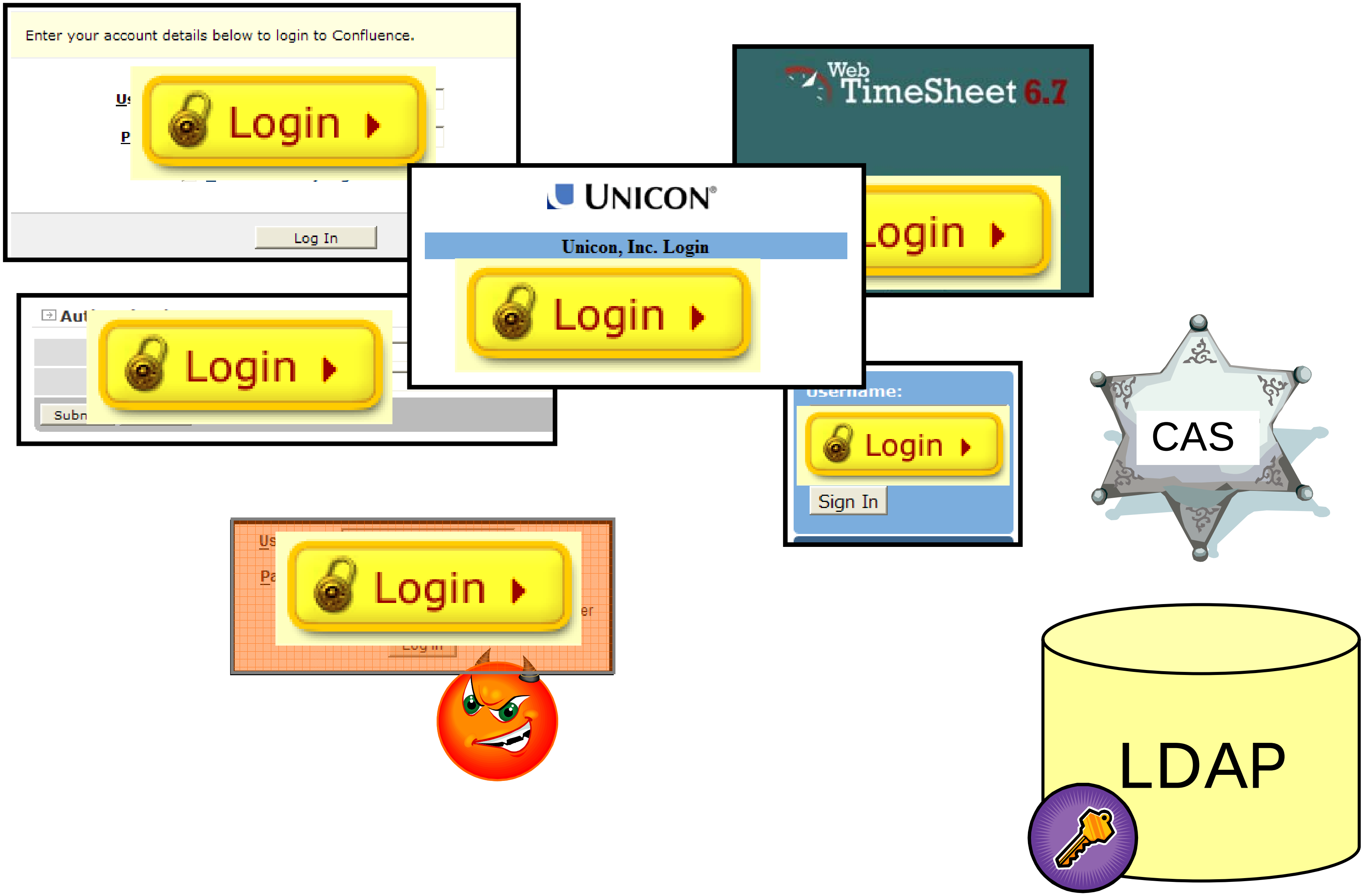
Web application

# Webapps no longer touch passwords



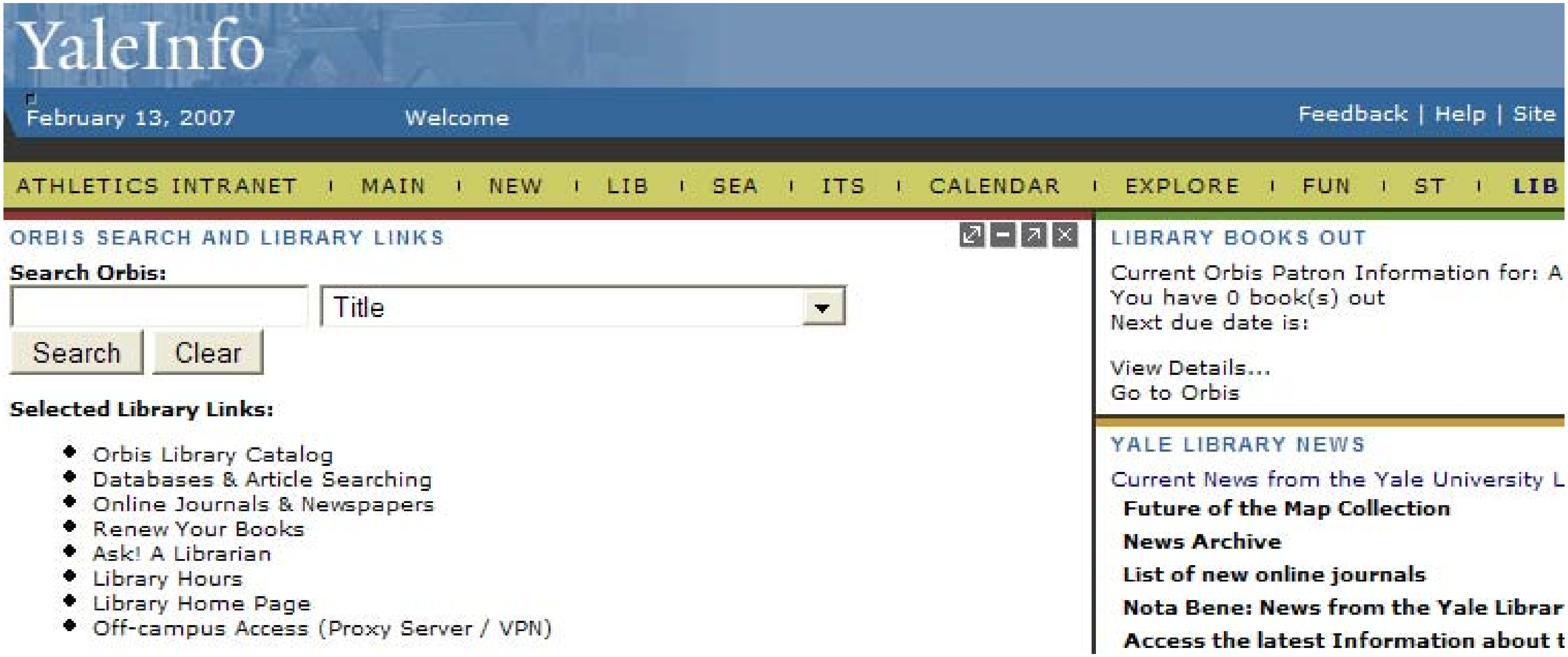
Enter your account details below to login to Confluence.

# Adversary compromises only single apps



Enter your account details below to login to Confluence.

# What about portals?



YaleInfo
February 13, 2007
Welcome
Feedback | Help | Site
ATHLETICS INTRANET | MAIN | NEW | LIB | SEA | ITS | CALENDAR | EXPLORE | FUN | ST | LIB

ORBIS SEARCH AND LIBRARY LINKS

**Search Orbis:**

Title

Search Clear

**Selected Library Links:**

- Orbis Library Catalog
- Databases & Article Searching
- Online Journals & Newspapers
- Renew Your Books
- Ask! A Librarian
- Library Hours
- Library Home Page
- Off-campus Access (Proxy Server / VPN)

LIBRARY BOOKS OUT

Current Orbis Patron Information for: A
You have 0 book(s) out
Next due date is:

View Details...
Go to Orbis

YALE LIBRARY NEWS

Current News from the Yale University L

**Future of the Map Collection**

**News Archive**

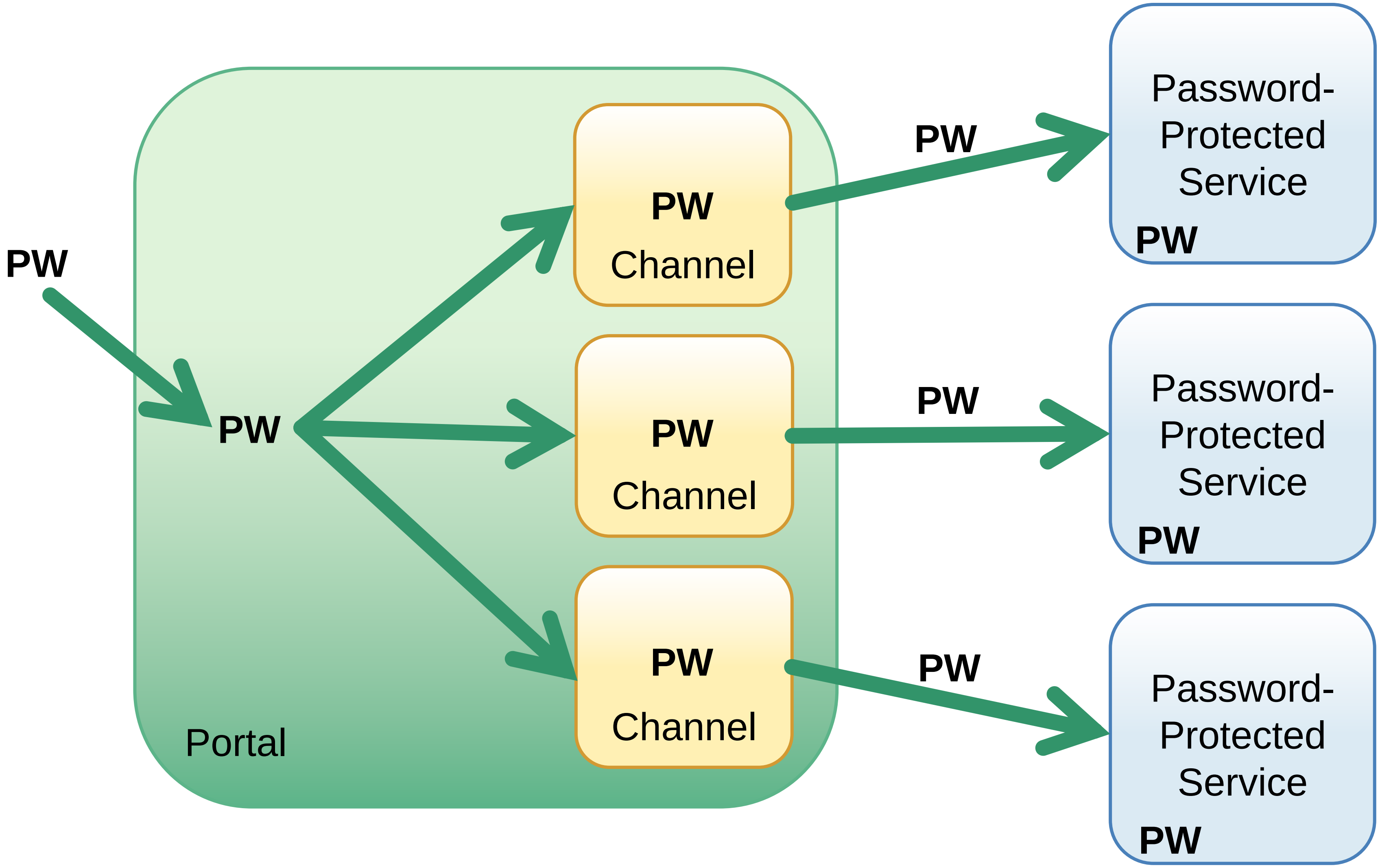
**List of new online journals**

**Nota Bene: News from the Yale Librar**

**Access the latest Information about t**

Need to go get interesting content from different systems.

# Password Replay


PW
PW
Portal
PW
Channel
PW
Channel
PW
Channel
PW
PW
PW
Password-Protected Service
PW
Password-Protected Service
PW
Password-Protected Service
PW

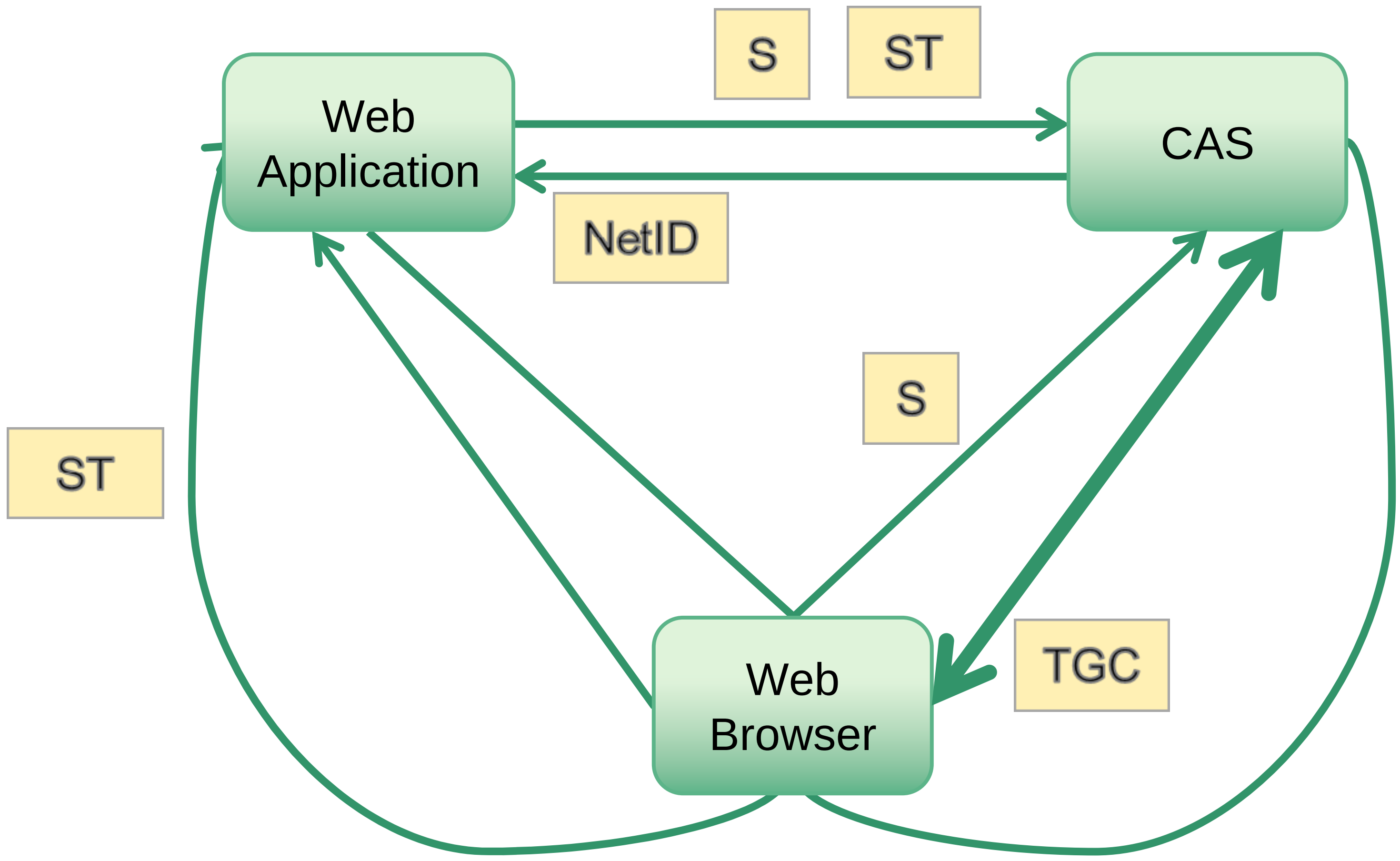
# CAS Protocol

Tickets and services

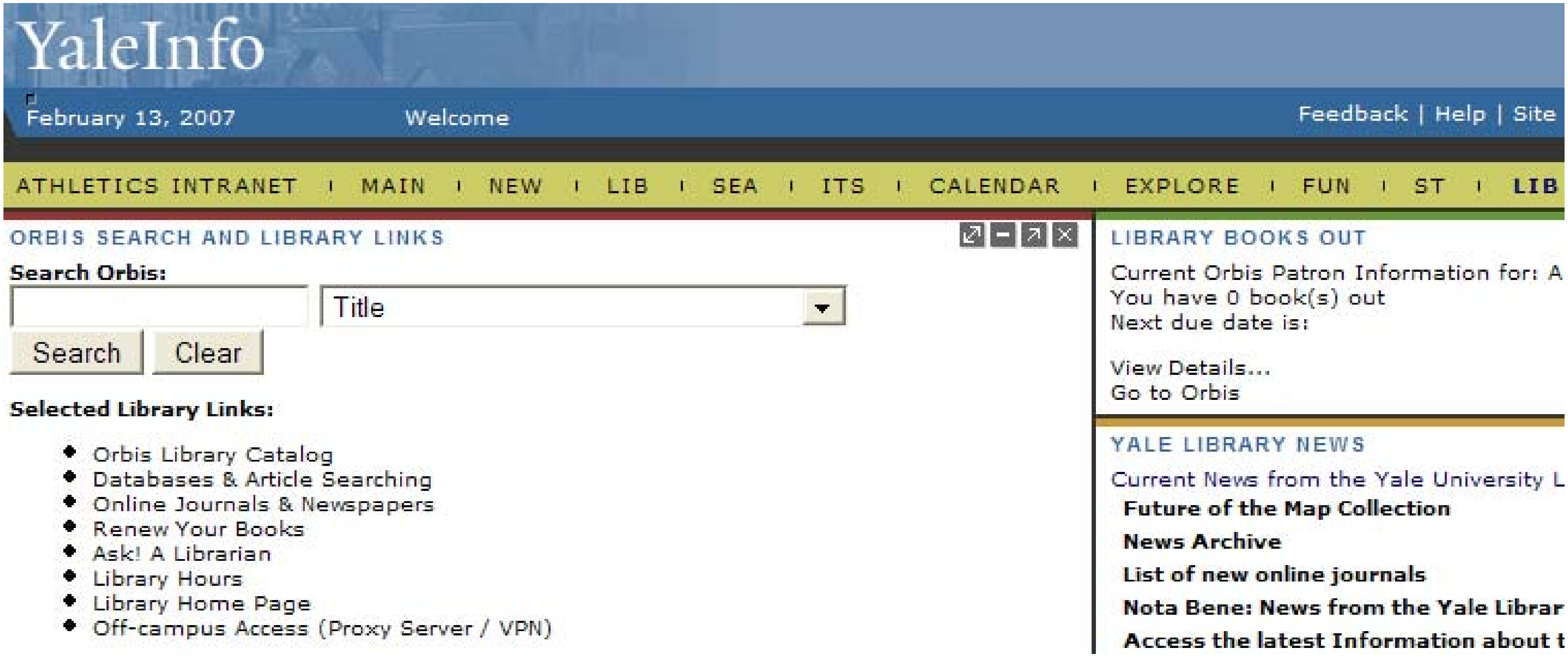
Ticket validation

Proxy authentication

# How CAS Works



# What about portals?


YaleInfo
Feedback | Help | Site
ATHLETICS INTRANET | MAIN | NEW | LIB | SEA | ITS | CALENDAR | EXPLORE | FUN | ST | LIB
Title
Search
Clear

February 13, 2007 Welcome

ORBIS SEARCH AND LIBRARY LINKS

Search Orbis:

Selected Library Links:

- Orbis Library Catalog
- Databases & Article Searching
- Online Journals & Newspapers
- Renew Your Books
- Ask! A Librarian
- Library Hours
- Library Home Page
- Off-campus Access (Proxy Server / VPN)

LIBRARY BOOKS OUT

Current Orbis Patron Information for: A
You have 0 book(s) out
Next due date is:

View Details...
Go to Orbis

YALE LIBRARY NEWS

Current News from the Yale University L

**Future of the Map Collection**

**News Archive**

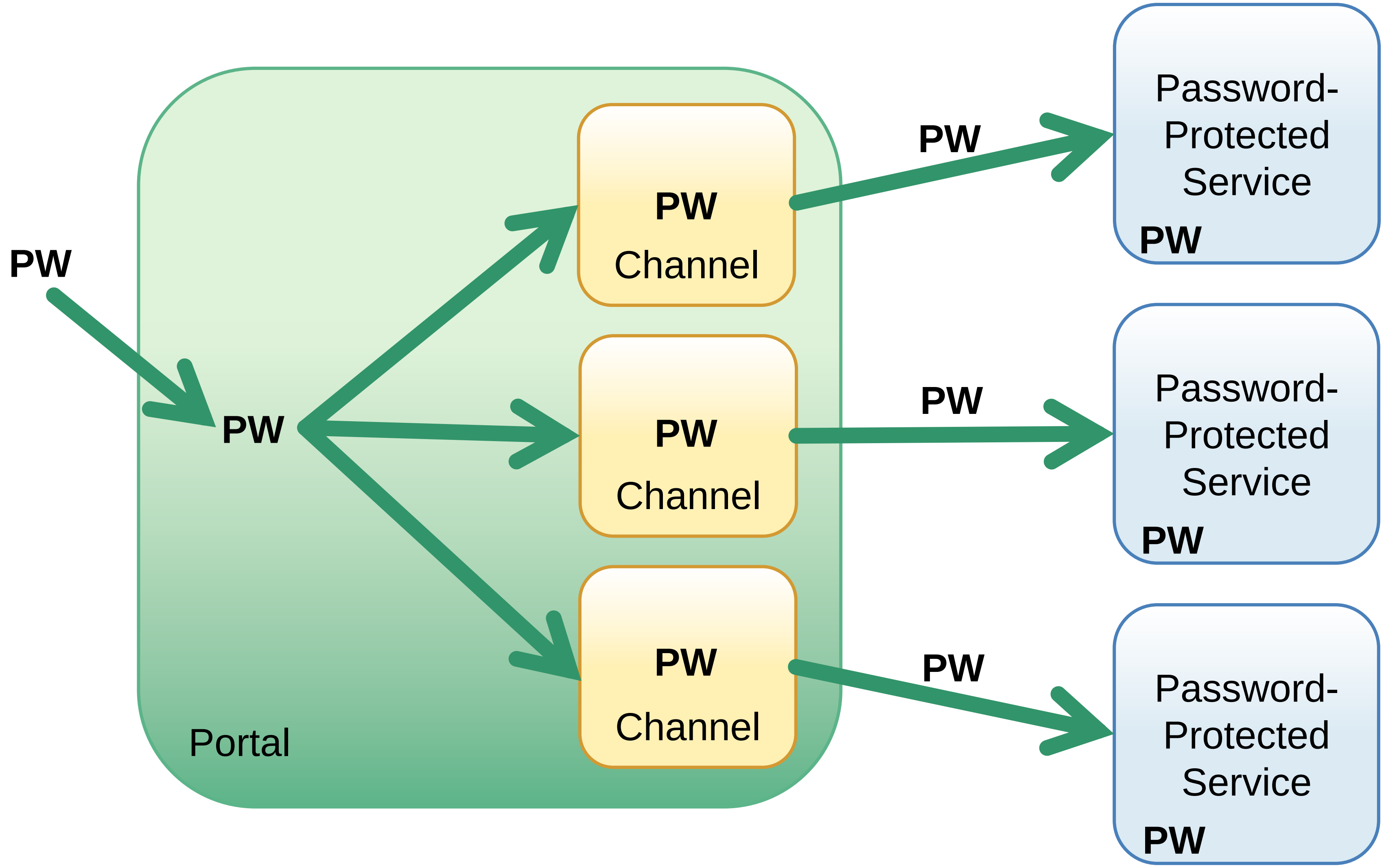
**List of new online journals**

**Nota Bene: News from the Yale Librar**

**Access the latest Information about t**

Need to go get interesting content from different systems.

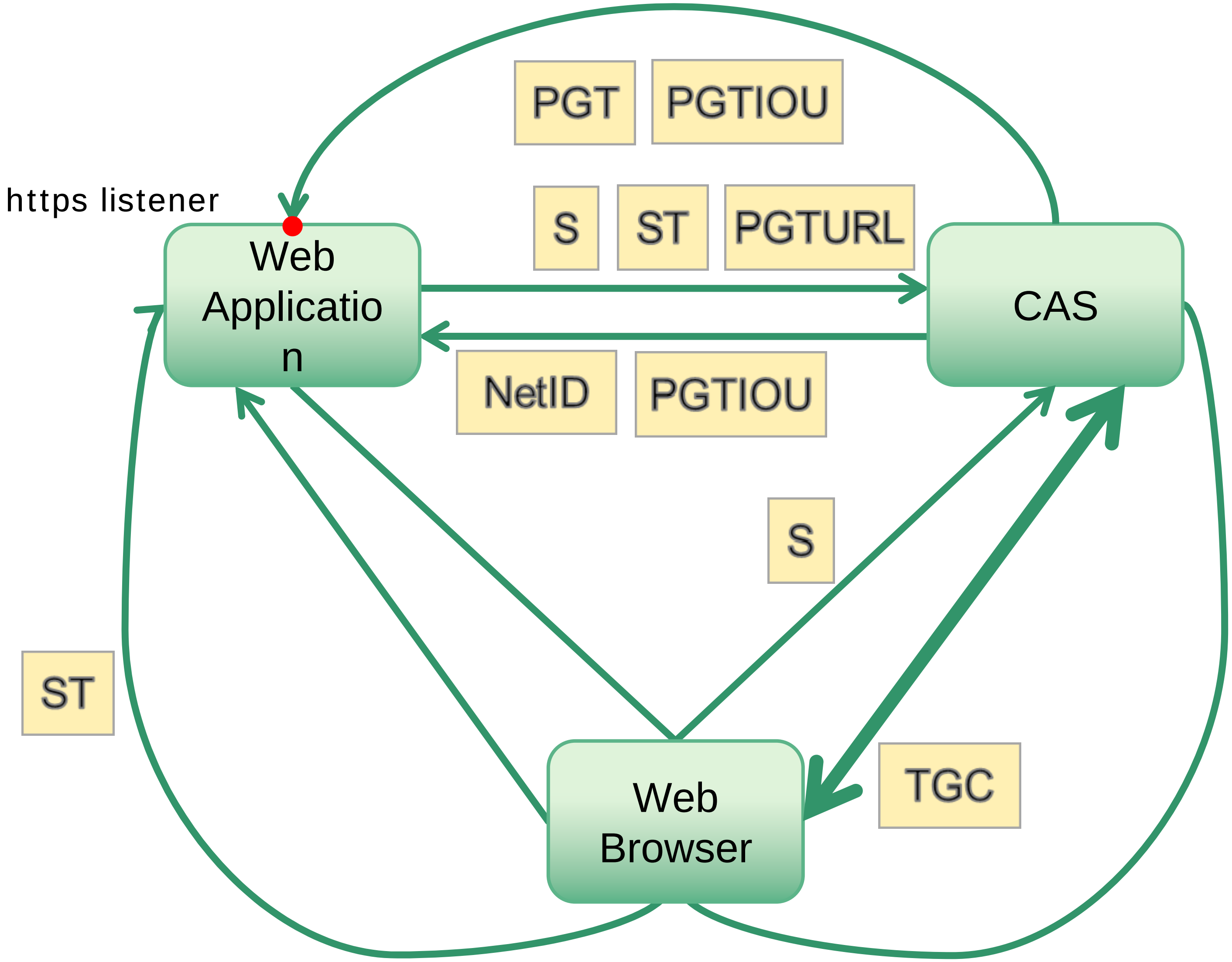
# Password Replay


PW
PW
PW
Channel
PW
Channel
PW
Channel
Portal
PW
PW
PW
Password-Protected Service
PW
Password-Protected Service
PW
Password-Protected Service
PW

# Look ma, no password!

- Without a password to replay, how am I going to authenticate my portal to other applications?

# Proxy CAS


PGT
PGTIOU
https listener
S
ST
PGTURL
Web Applicatio n
CAS
NetID
PGTIOU
S
ST
TGC
Web Browser

# Proxy CAS

- Feature unique to CAS among most of SSO systems
- Allows some Web applications to act as proxies on behalf of the users
- Proxied Web applications may act as N-th level proxies

http://www.jasig.org/cas/protocol

# Provided Authentication Handlers

- LDAP
  - Fast bind
  - Search and bind
- Active Directory
  - LDAP
  - Kerberos (JAAS)
- JAAS
- JDBC
- RADIUS
- SPNEGO
- Trusted
- X.509 certificates
- Writing a custom authentication handler is easy

# CAS – More than Authentication

- Return attributes of logged on users
- Adding support for standards
  - OpenID
  - SAML
- Single Sign-Out
- RESTful API
- Support for clustering
  - Implements distributed ticket registry
  - Requires session replication
  - Must guarantee cross-server ticket uniqueness
- Services management (white listing)
- Remember me (long-term SSO)

# Short Term Goals

- Service Registry Enhancements:
  - Self Registration Page
  - Service Priority
  - LDAP implementation of Service Registry
- InfoCard Support
- Auditing, Logging etc.
- More Internationalization

# Long Term Goals

- Re-architecture to support emerging use cases
  - Account Management integration
  - Password Expiration Policies/Password Change Integration
  - SAML, OAuth, OpenID2, etc.
  - Levels of assurance / multifactor authentication / second-level
- Better online / realtime administration
  - Installer / configurer
  - Information about CAS server (open SSO sessions, etc.)
- Hardening / anti-phishing

http://www.ja-sig.org/wiki/display/CASST/CAS+4+Roadmap

# Building from sources

Obtaining the distribution

Requirements and tools

File structure and dependencies

# Obtaining the distribution

- http://www.jasig.org/cas/
- SVN at developer.ja-sig.org

```
svn checkout https://www.ja-
  sig.org/svn/cas3/tags/cas-3-3-1-final/
  cas-server
```

- Import and maintain in your source control's vendor branch

# Requirements to build CAS

- Required
  - Java Development Kit 5 or 6
  - Maven 2
- Optional
  - SVN
  - Eclipse (with SVN and Maven plugins)
  - EasyEclipse Server (plus Maven plugin)
  - Tomcat (gotta test it somewhere!)

# File structure and dependencies

- Top-level Project Object Model (POM or pom.xml) used for all builds and to build dependent sub-projects.
- The top-level POM builds all the sub-projects, but by default they are NOT included in the resulting war file.
- To add dependent sub-projects or additional external libraries to the war file, you need to add dependencies to pom.xml in cas-server-webapp.

# Adding a dependency to pom.xml

```xml
<!-- ... -->
<dependency>
  <groupId>ognl</groupId>
  <artifactId>ognl</artifactId>
  <version>2.6.9</version>
  <scope>runtime</scope>
</dependency>

<dependency>
  <groupId>${project.groupId}</groupId>
  <artifactId>cas-server-support-ldap</artifactId>
  <version>${project.version}</version>
</dependency>

<dependency>
  <groupId>log4j</groupId>
  <artifactId>log4j</artifactId>
  <version>1.2.14</version>
  <type>jar</type>
  <scope>runtime</scope>
</dependency>
<!-- ... -->
```

# Building using Maven overlay method

Requirements

Project Structure

Dependencies

# Requirements to build CAS

- Required
  - Java Development Kit 5 or 6
  - Maven 2
- Optional
  - SVN
  - Eclipse (with SVN and Maven plugins)
  - EasyEclipse Server (plus Maven plugin)
  - Tomcat (gotta test it somewhere!)

# File structure and dependencies

- Start with just Project Object Model (pom.xml) in an empty project directory.
- Add files, as needed, to “overlay” those in the standard WAR file.
    - Your own deployerConfigContext.xml would be the first such file.
    - May want to add institutional images and CSS modifications.
- Add dependencies, as needed, to additional CAS modules.

# Configuring CAS

deployerConfigContext.xml

web.xml

log4j.properties

# deployerConfigContext.xml

- Located in cas-server-webapp/src/main/webapp/WEB-INF
- Deployer-specific configuration file
- This is the first and possibly the only file you have to modify
- Replace the default authentication handler with the one your deployment needs
- Add configuration options that your authentication handler requires

# deployerConfigContext.xml example

```
<bean id="authenticationManager" class="org.jasig.cas.authentication.AuthenticationManagerImpl">
 <!-- ... -->
   <property name="authenticationHandlers">
     <list>
       <!--
         | This is the authentication handler that authenticates services by means of callback via SSL, thereby validating
         | a server side SSL certificate.
         +-->
       <bean class="org.jasig.cas.authentication.handler.support.HttpBasedServiceCredentialsAuthenticationHandler" />

       <bean class="org.jasig.cas.adaptors.ldap.BindLdapAuthenticationHandler">
         <property name="filter" value="uid=%u" />
         <property name="searchBase" value="ou=People,dc=training" />
         <property name="contextSource" ref="contextSource" />
       </bean>
     </list>
   </property>
 </bean>

 <bean id="contextSource" class="org.jasig.cas.adaptors.ldap.util.AuthenticatedLdapContextSource">
   <property name="anonymousReadOnly" value="true" />
   <property name="password" value="{password_goes_here}" />
   <property name="pooled" value="true" />
   <property name="urls">
     <list>
        <value>ldap://localhost/</value>
     </list>
   </property>
   <property name="userName" value="{username_goes_here}" />
   <property name="baseEnvironmentProperties">
     <map>
       <entry>
         <key><value>java.naming.security.authentication</value></key>
         <value>simple</value>
       </entry>
     </map>
   </property>
 </bean>
```

# web.xml

- Located in cas-server-webapp/src/main/webapp/WEB-INF
- Standard JEE deployment descriptor
- All endpoints defined as mapped to one servlet
- Uses Spring WebMVC
- This is the “root” of the CAS Web application configuration
- Re-enable the user-friendly error reporting
- Lists all the Spring context configuration files
- My need to add auditTrailContext.xml

# log4j.properties

- Located in cas-server-webapp/src/main/webapp/WEB-INF/classes
- Log4j periodically re-reads this file (no Tomcat restart needed after editing)
- Add fully-qualified path to “cas.log”
- May want to increase the log level for troubleshooting
- Warning: setting the log level to DEBUG or higher will log users’ passwords

# Additional Features

Service registry

Single sign-out

OpenID

# Service registry

- Allows to maintain a list of services authorized to authenticate to CAS
- Off by default
- When turned on, only registered services will be allowed to authenticate to CAS
- Implementing service registry introduces a database requirement to CAS (using Hibernate)
- Service registry’s management interface requires authentication (CAS, of course)
- Must add the service registry URL as the first service to avoid locking out access to the service registry management interface

# Enabling service registry

Find a section of deployerConfigContext.xml that looks like this:

```
<bean id="userDetailsService" class="org.acegisecurity.userdetails.memory.InMemoryDaoImpl">
  <property name="userMap">
    <value>

    </value>
  </property>
</bean>
```

and make it look like this:

```
<bean id="userDetailsService" class="org.acegisecurity.userdetails.memory.InMemoryDaoImpl">
  <property name="userMap">
    <value>
      adam=notused,ROLE_ADMIN
    </value>
  </property>
</bean>
```

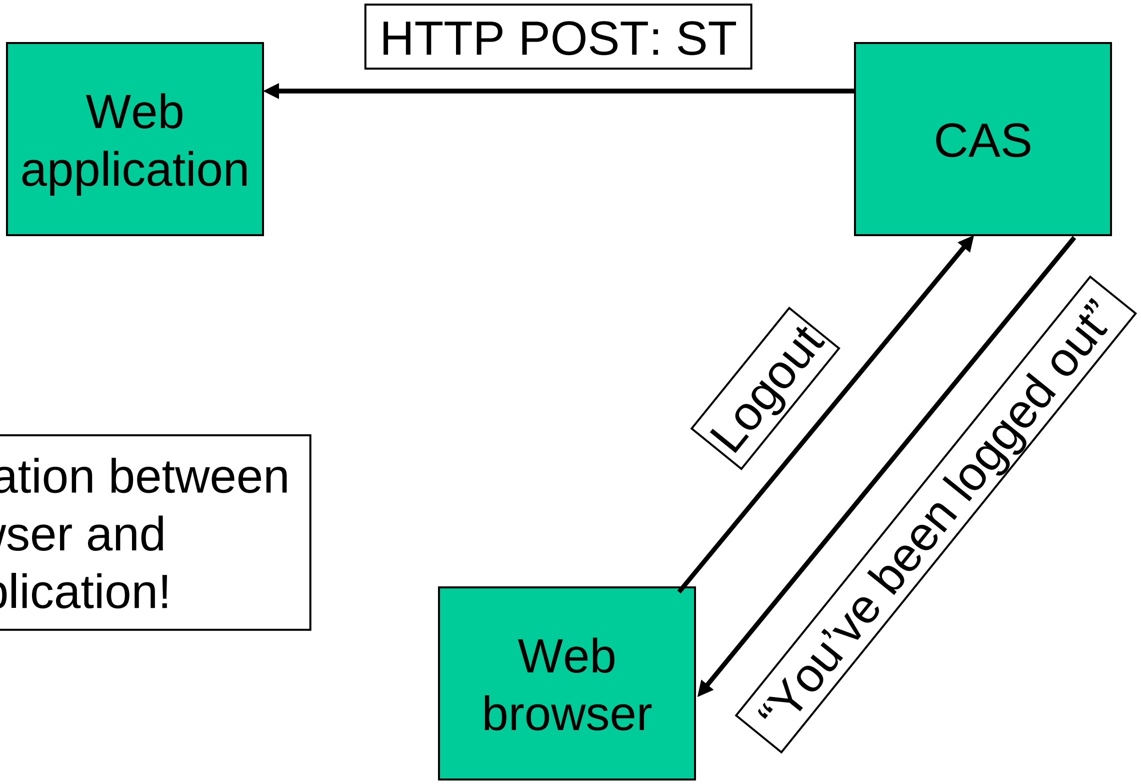
Now user “adam” is authorized to manage services.

Need to enable the database persistence, too.

# Single sign-out

- Allows CAS to post a “user signed out” message to all services that have previously authenticated to CAS
- On by default
- Services receive the SSOut message as an HTTP POST to the same endpoint identified during authentication
- Service Ticket identifies the SSO session that was terminated

# Single sign-out



HTTP POST: ST

Web application

CAS

Logout

"You've been logged out"

Web browser

No communication between the browser and Web application!

# OpenID (http://openid.net/)

- Allows to use a single digital identity across the Internet
- Web applications delegate authentication to an OpenID provider
- CAS can be configured to be an OpenID provider
- Useful if you have Web applications that support OpenID authentication and not CAS

# CAS-enabling (or CASifying) Web applications

uPortal

Tomcat Manager

????

# uPortal

- Edit `properties/security.properties`
- Edit `webpages/WEB-INF/web.xml`
- Edit (uPortal 2.x only) `webpages/stylesheets/org/jasig/portal/channels/CLogin/html.xsl`
- Deploy the changes
- Restart uPortal

# Tomcat Manager

- Tomcat Manager relies on container authentication
- This example illustrates how CAS authentication can replace Tomcat’s BASIC Authentication without having to write or modify any code
- Locate the Manager applications deployment descriptor (web.xml)
- Replace its original authentication section with CAS filter-based authentication
- Add simple authorization
- http://www.ja-sig.org/wiki/x/5yM

# Questions?

Adam Rybicki

arybicki@unicon.net

www.unicon.net

UNICON