

Implementing CAS

Adam Rybicki

2010 Jasig Conference, San Diego, CA

March 7, 2010

© Copyright Unicon, Inc., 2009. This work is the intellectual property of Unicon, Inc. Permission is granted for this material to be shared for non-commercial, educational purposes, provided that this copyright statement appears on the reproduced materials and notice is given that the copying is by permission of Unicon, Inc. To disseminate otherwise or to republish requires written permission from Unicon, Inc.



1. Introduction
2. Problems CAS solves
3. CAS protocol
4. Building from sources
5. Customizing the presentation
6. Authentication handlers
7. CAS-enabling applications
8. Using Proxy CAS
9. Advanced Topics
 - Clustering
 - Service Registry
 - Single Sign-Out

Introduction

Who are we?

What is CAS?

Brief history of CAS

Adam's Involvement with CAS

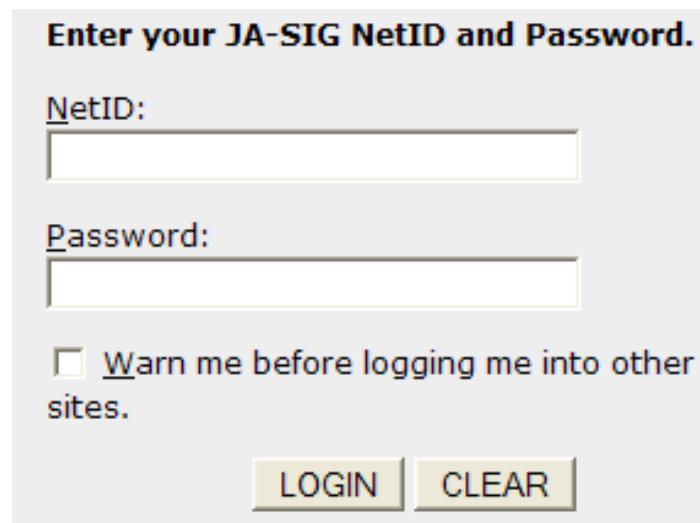
- Got interested.
- Worked with several clients helping them to CASify their applications.
- Asked many questions of the CAS mail list
- Wrote a CAS self-study guide for Unicon developers.
(<https://confluence.unicon.net/confluence/x/XgZi>) (authentication required)
- Answered some questions on the CAS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

NetID:

Password:

Warn me before logging me into other sites.

Some of the people involved as the project has evolved

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

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 this computer


Log In

Web TimeSheet 6.7




apetro

Enter

 UNICON®

Unicon, Inc. Login

Name:
Password:


Login 


Authentication

Username
Password

Submit Cancel


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.

Username:

Password:

Remember my login on this computer

Log In

Web TimeSheet 6.7

apetro

Enter

UNICON®

Unicon, Inc. Login

Name:

Password:

Login

Authentication

Username

Password

Submit Cancel

Username: apetro

Password: *****

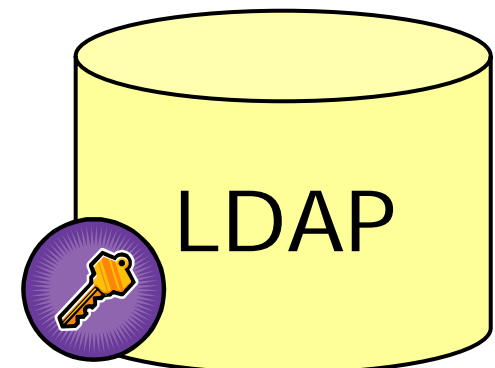
Sign In

Username

Password

Remember my login on this computer

Log In



All applications touch passwords

Enter your account details below to login to Confluence.

Username:

Password:

Remember my login on this computer

Log In

Web TimeSheet 6.7

apetr

Enter

UNICON®

Unicon, Inc. Login

Name:

Password:

Authentication

Username

Password

Submit Cancel

apetr

Pass *****

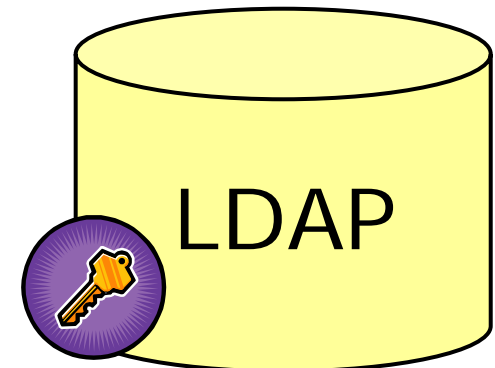
Sign In

Username

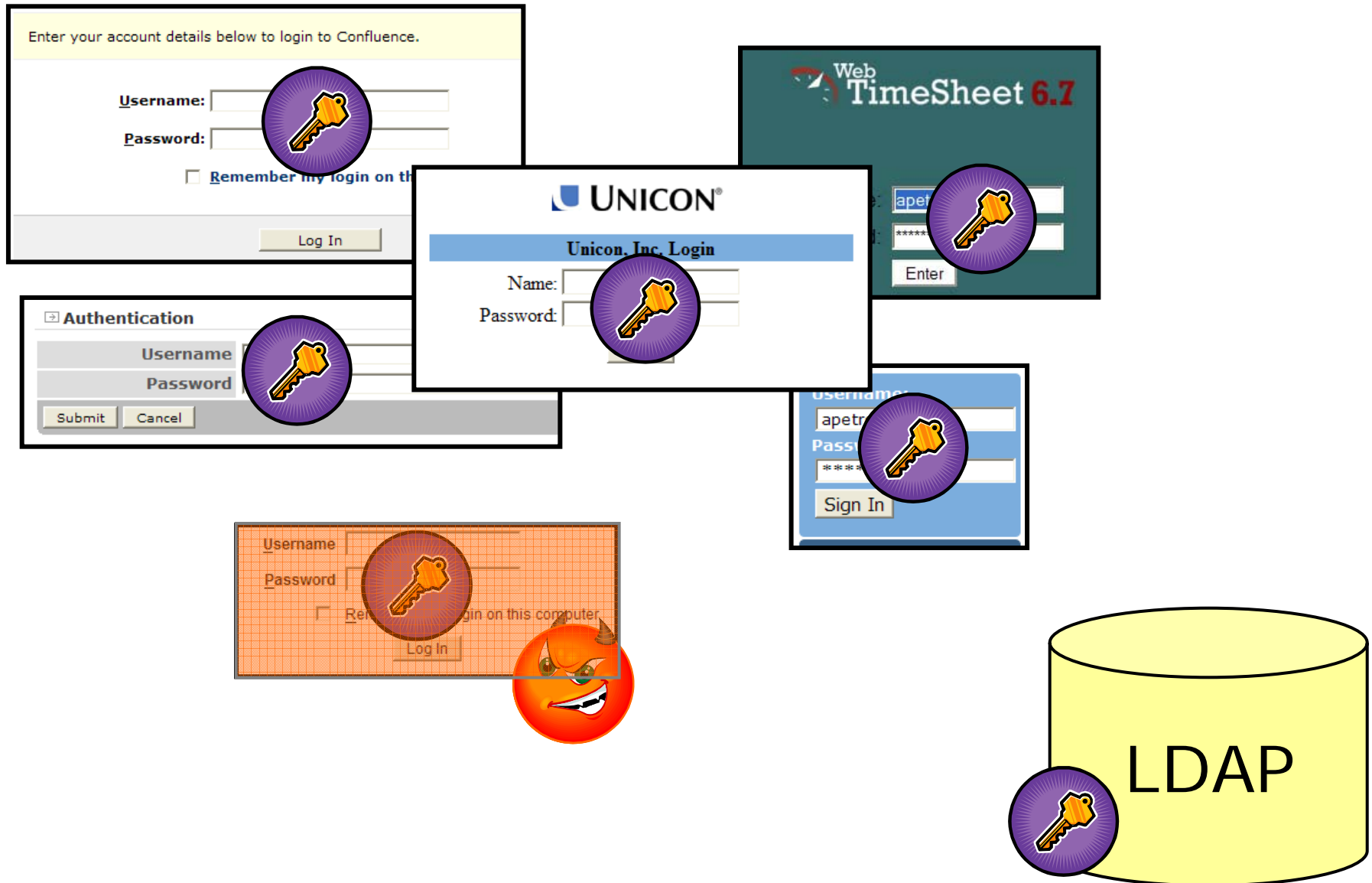
Password

Remember my login on this computer

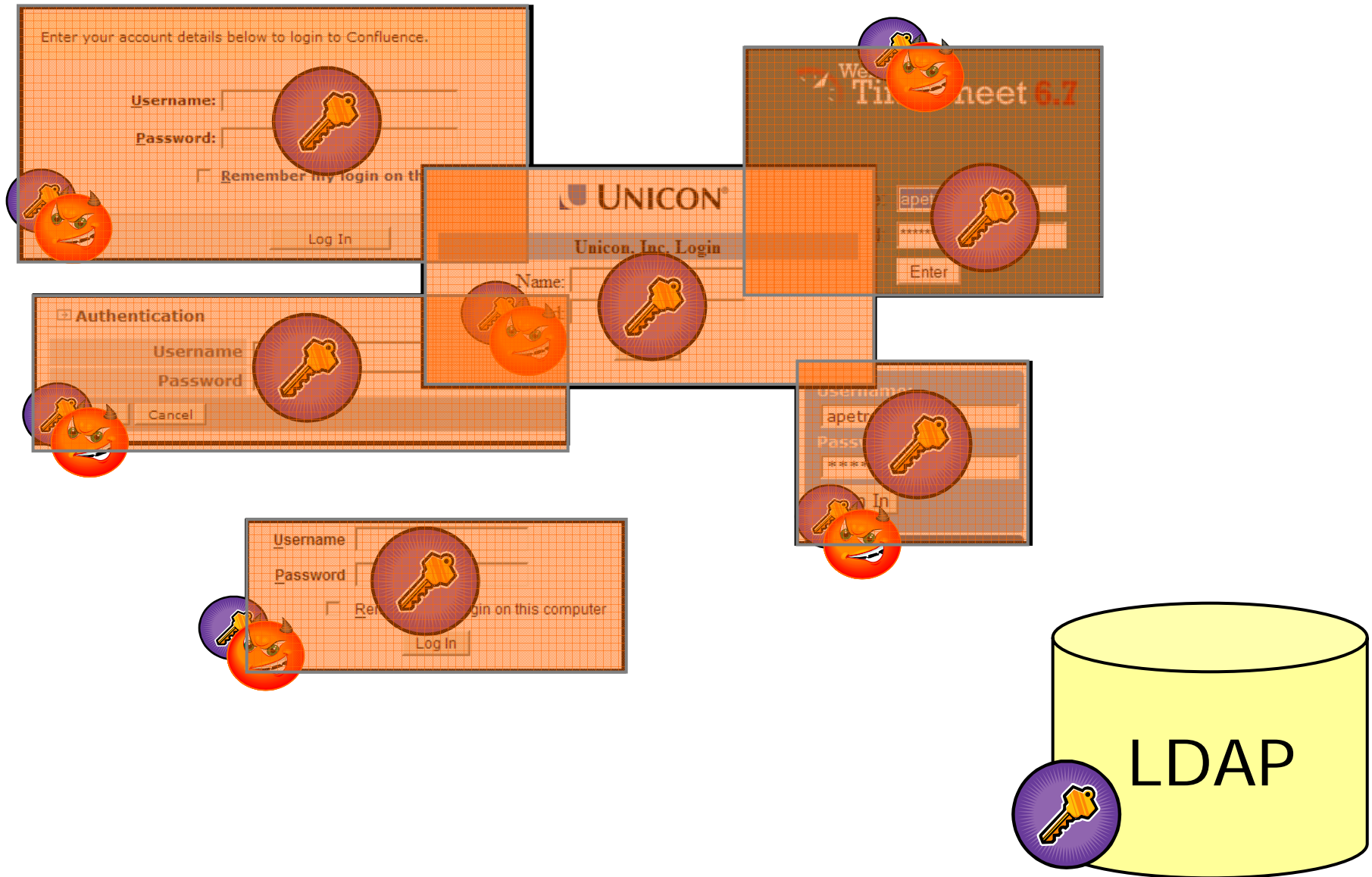
Log In



Any compromise leaks primary credentials



Adversary then can run wild



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.

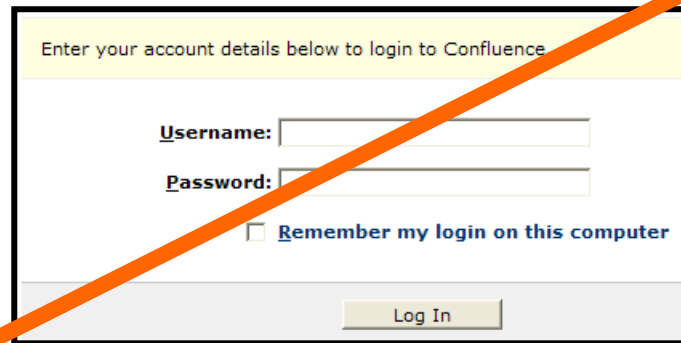
NetID:

Password:

Warn me before logging me into other sites.



Delete your login forms.



Enter your account details below to login to Confluence

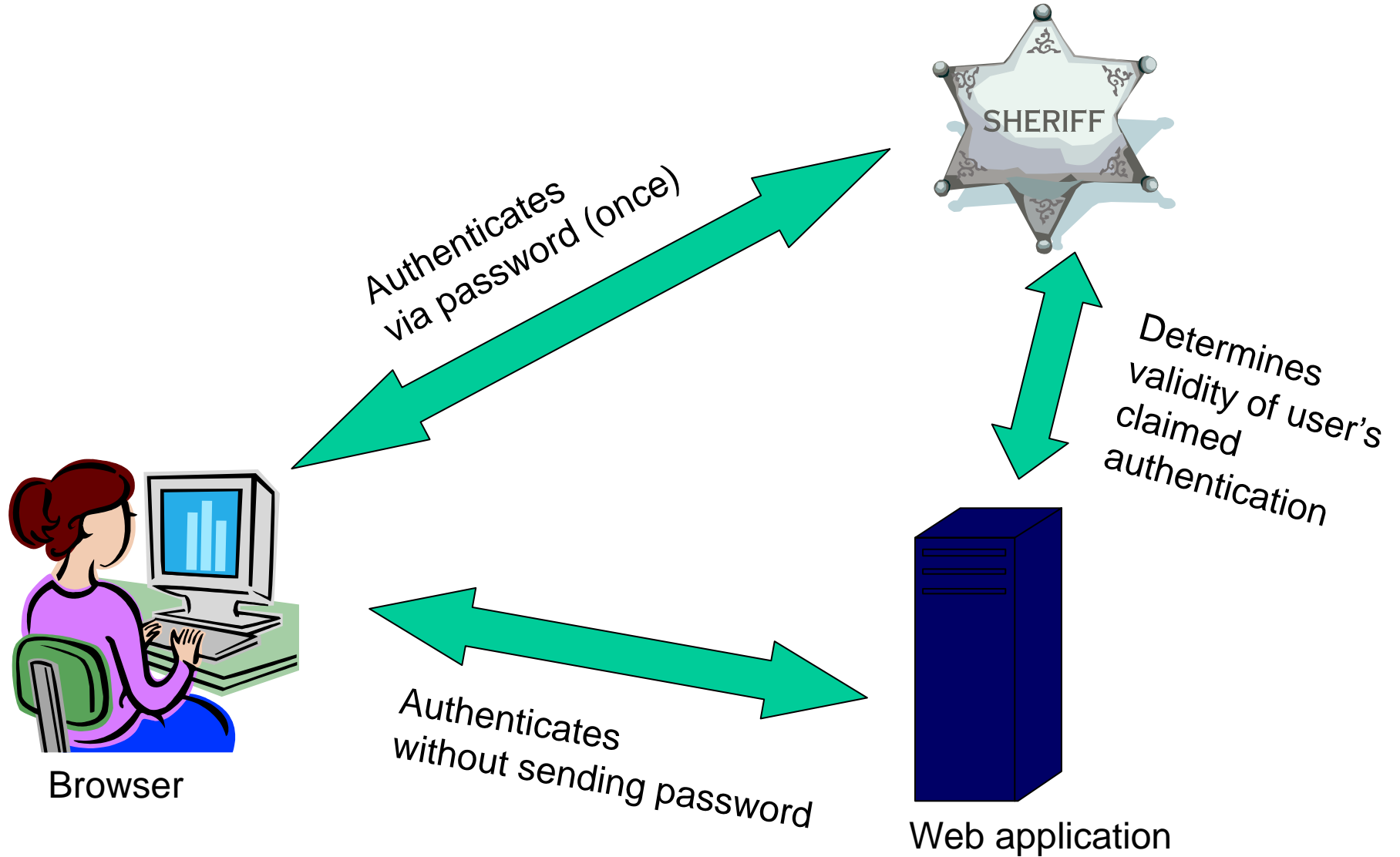
Username:

Password:

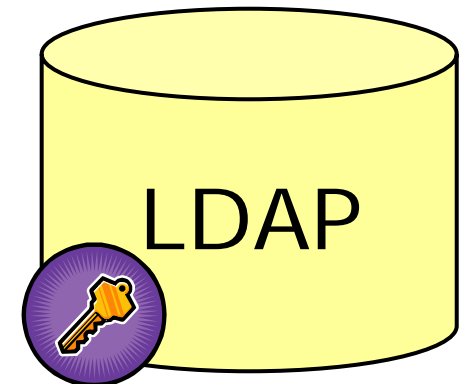
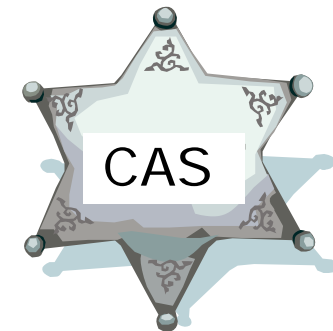
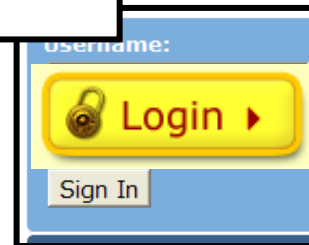
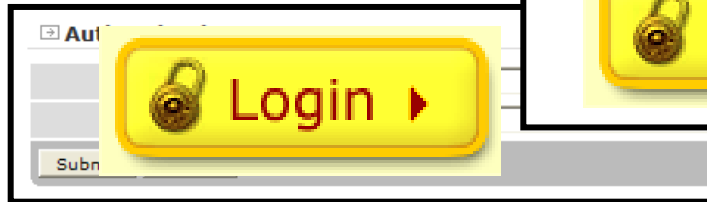
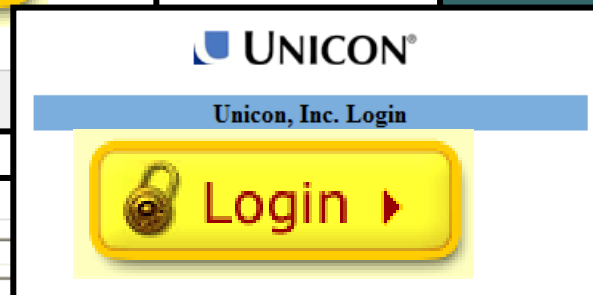
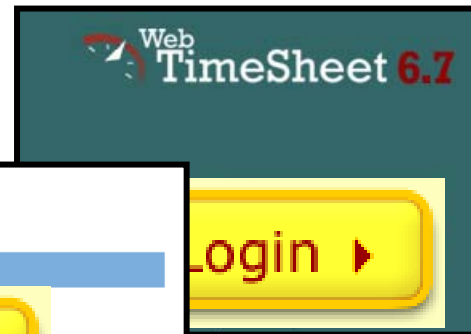
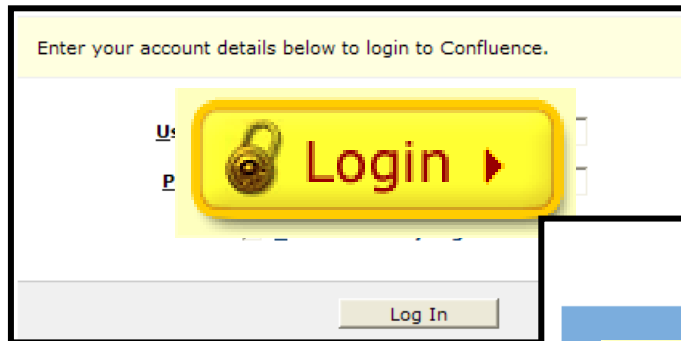
Remember my login on this computer



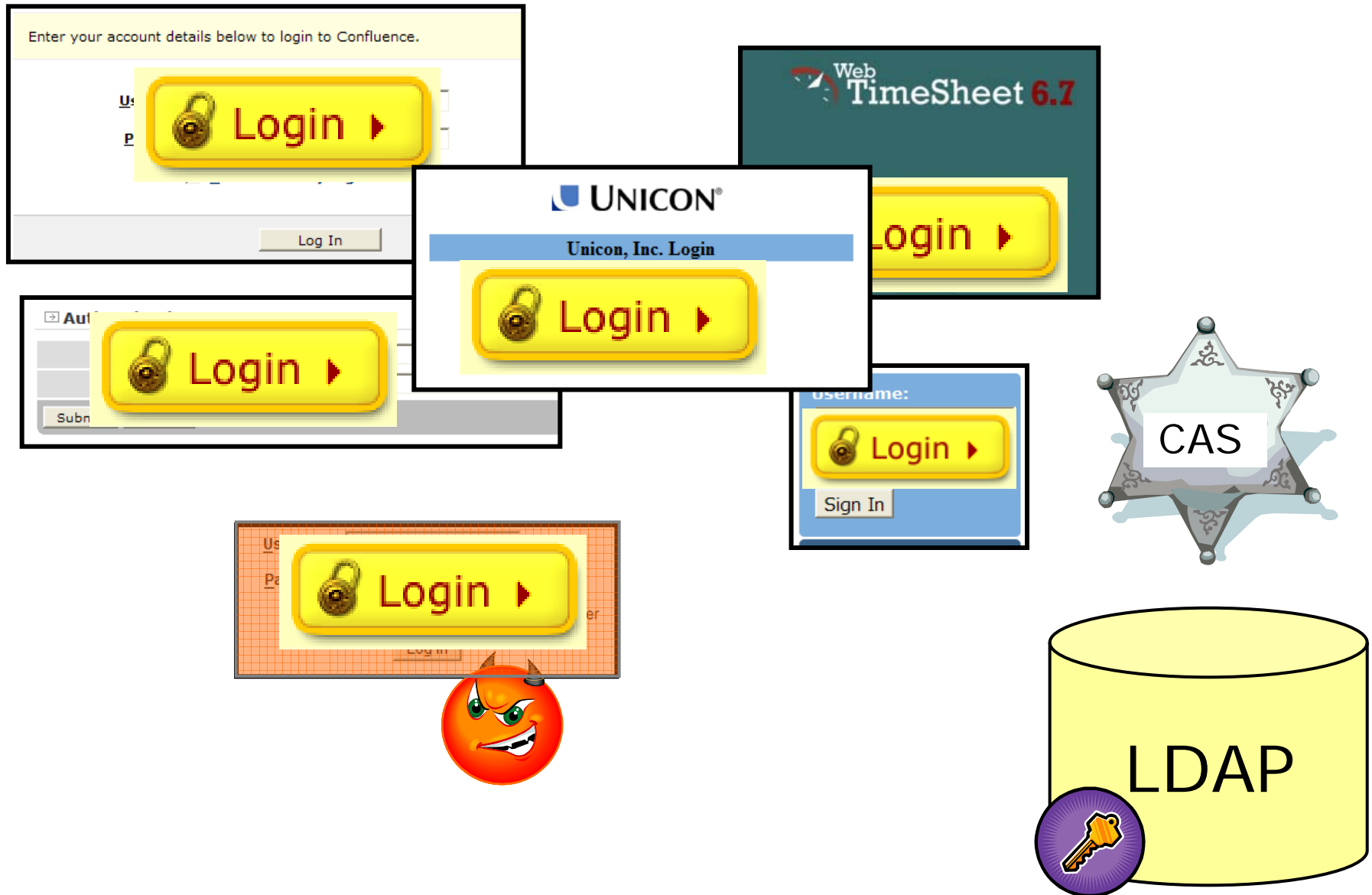
CAS in a nutshell



Webapps no longer touch passwords



Adversary compromises only single apps

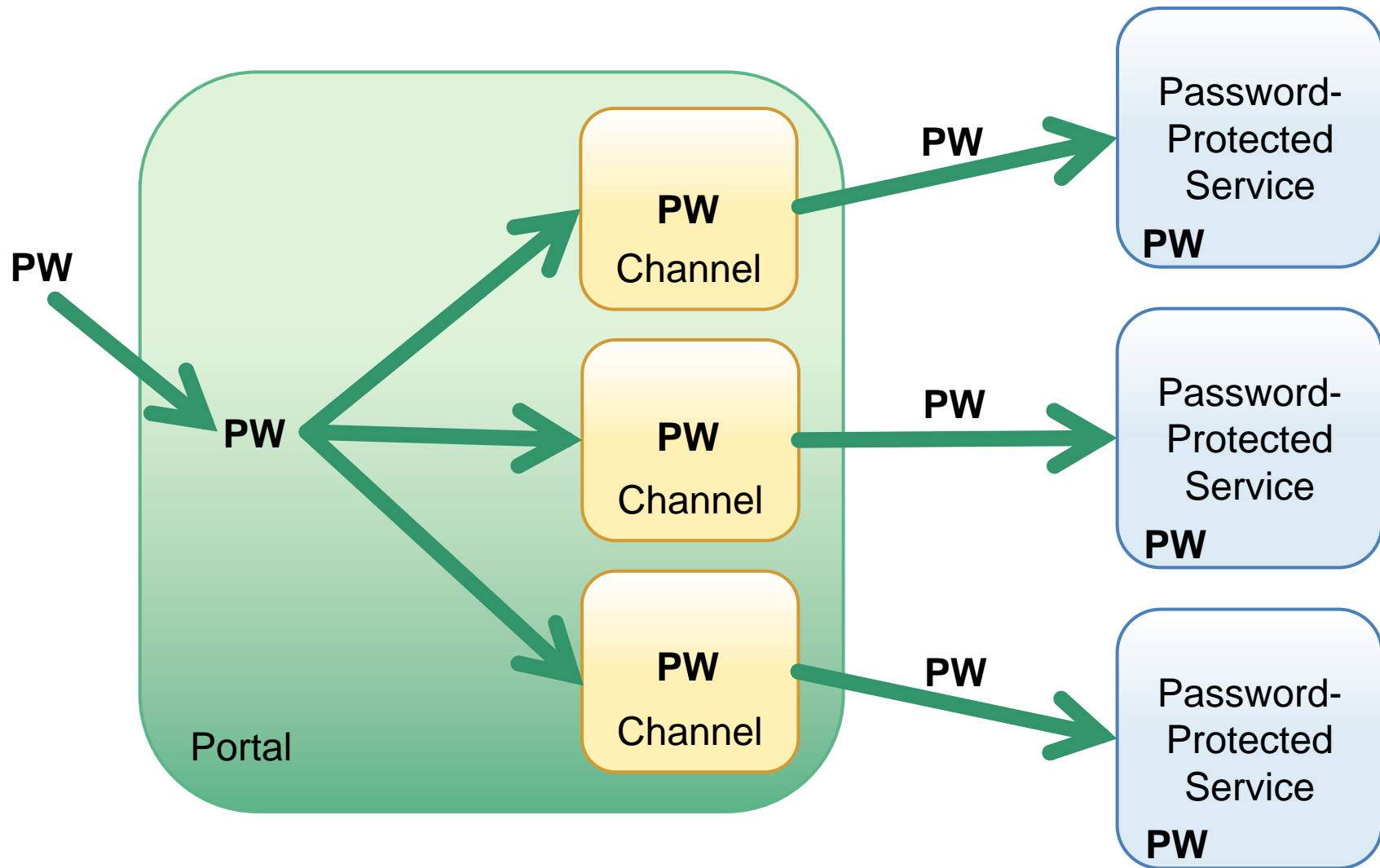


What about portals?

The screenshot shows the YaleInfo library portal. At the top left is the 'YaleInfo' logo. Below it, the date 'February 13, 2007' and the word 'Welcome' are displayed. On the top right, there are links for 'Feedback | Help | Site'. A horizontal navigation bar contains links: 'ATHLETICS INTRANET | MAIN | NEW | LIB | SEA | ITS | CALENDAR | EXPLORE | FUN | ST | LIB'. The main content area is divided into two columns. The left column is titled 'ORBIS SEARCH AND LIBRARY LINKS' and contains a search box with the text 'Title' and buttons for 'Search' and 'Clear'. Below the search box is a section titled 'Selected Library Links:' with a list of links: 'Orbis Library Catalog', 'Databases & Article Searching', 'Online Journals & Newspapers', 'Renew Your Books', 'Ask! A Librarian', 'Library Hours', 'Library Home Page', and 'Off-campus Access (Proxy Server / VPN)'. The right column is titled 'LIBRARY BOOKS OUT' and displays 'Current Orbis Patron Information for: A', 'You have 0 book(s) out', and 'Next due date is:'. Below this are links for 'View Details...' and 'Go to Orbis'. At the bottom of the right column is a section titled 'YALE LIBRARY NEWS' with links for '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', and 'Access the latest Information about t'.

Need to go get interesting content from different systems.

Password Replay



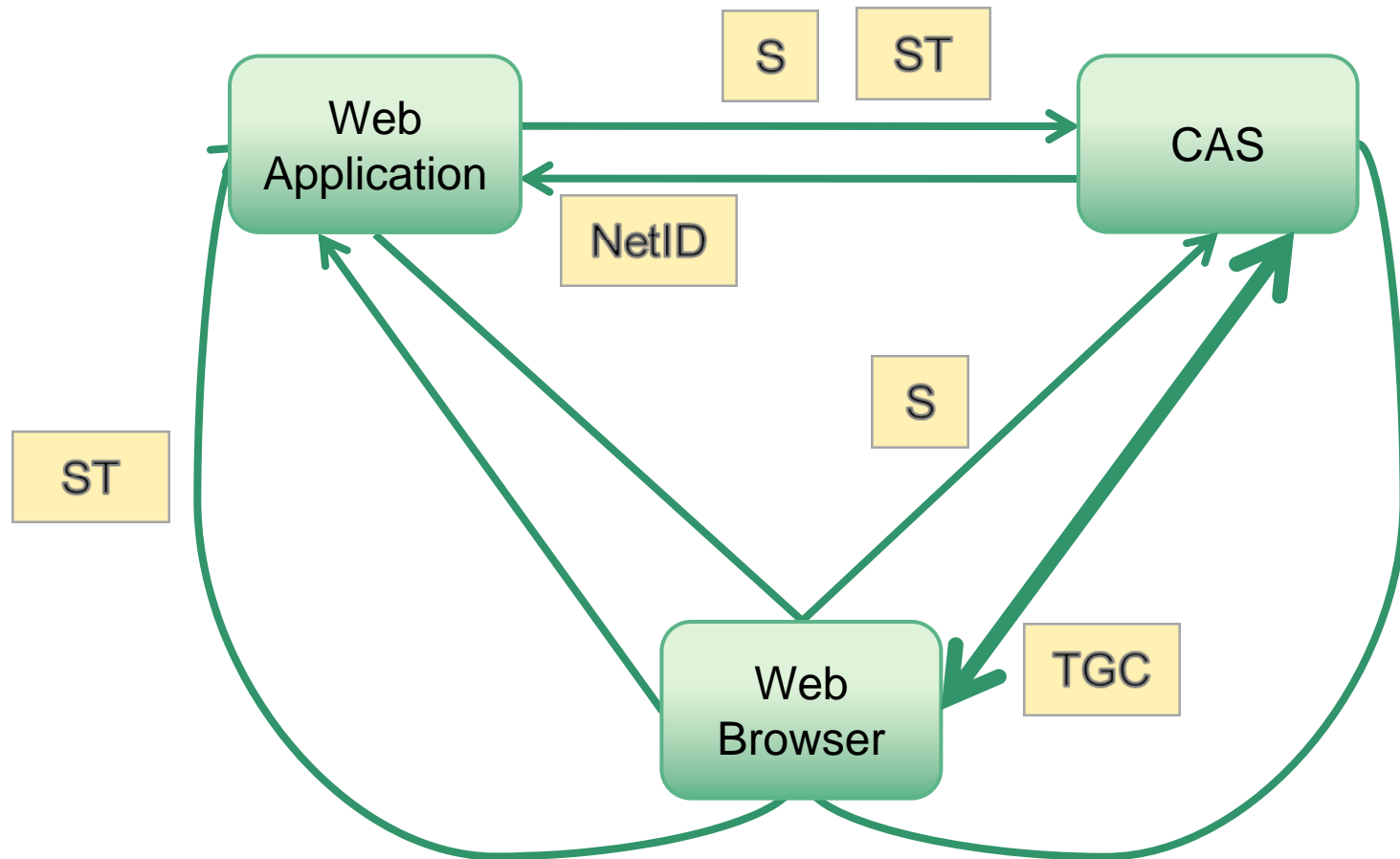
CAS Protocol

Tickets and services

Ticket validation

Proxy authentication

How CAS Works



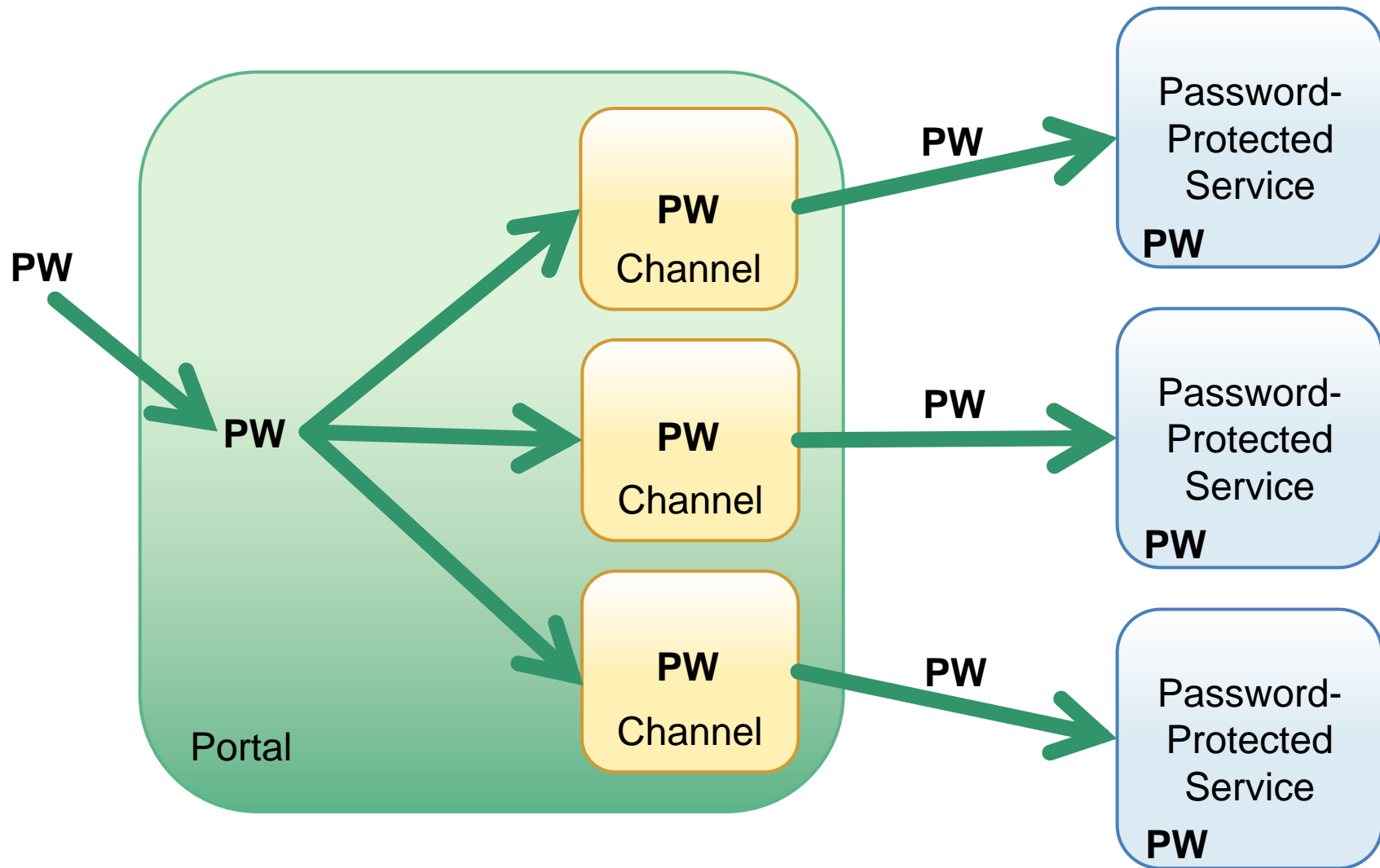
What about portals?

The screenshot shows the YaleInfo library portal. At the top left is the 'YaleInfo' logo. Below it, the date 'February 13, 2007' and the word 'Welcome' are displayed. On the top right, there are links for 'Feedback | Help | Site'. A horizontal navigation bar contains links for 'ATHLETICS INTRANET | MAIN | NEW | LIB | SEA | ITS | CALENDAR | EXPLORE | FUN | ST | LIB'. The main content area is divided into several sections:

- ORBIS SEARCH AND LIBRARY LINKS**: Includes a search box with the text 'Search Orbis:', a search input field containing 'Title', and 'Search' and 'Clear' buttons.
- Selected Library Links:** A list of links with diamond icons:
 - 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**: Displays 'Current Orbis Patron Information for: A', 'You have 0 book(s) out', and 'Next due date is:'. It includes links for 'View Details...' and 'Go to Orbis'.
- YALE LIBRARY NEWS**: Lists 'Current News from the Yale University L', 'Future of the Map Collection', 'News Archive', 'List of new online journals', and 'Nota Bene: News from the Yale Librar'. It concludes with 'Access the latest Information about t'.

Need to go get interesting content from different systems.

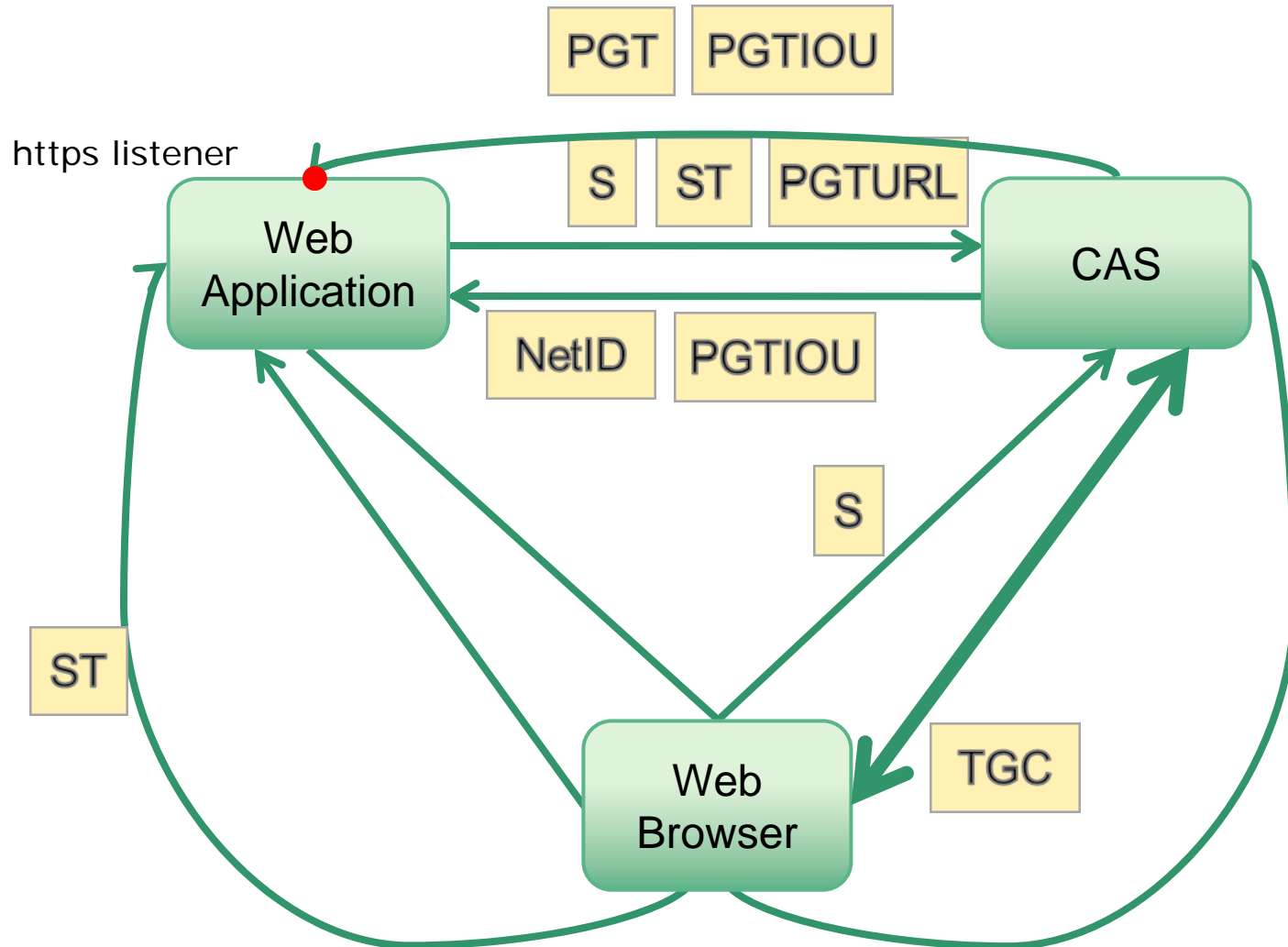
Password Replay



Look ma, no password!

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

Proxy CAS



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
 - Must guarantee cross-server ticket uniqueness
- Services management (white listing)
- Remember me (long-term SSO)

CAS Roadmap

CAS 3.4 Release

- Upgrades to "core" libraries including Spring (to 3.0), Spring Web Flow (to 2.0), Spring Security (to 3.0)
- Updates to Web Flow-related classes to conform with Web Flow 2.0 model
- Mobile CAS UI

CAS 3.5 Release

- Upgrades to core storage mechanisms. Most importantly, the API
- Introduction of core Factories for creating tickets
- Update to Ticket terminology to support future protocols
- Replacement of Jasig License with Apache 2 License

CAS 3.6 Release

- Rewrite of Services Management Tool
- Extraction of Services Management Tool into its own Web Application
- Addition of Registration Tool

CAS Roadmap (cont.)

CAS 3.7 Release

- Rewrite of two Core Interfaces: CentralAuthenticationService, AuthenticationManager to support additional use cases:
 - message passing to users
 - better throttling
 - CAPTCHA,
 - integration with password management tools
- Updated UI for:
 - message returning
 - reflect recent UI trends (immediate feedback on validation, etc.)
- Enable Advanced Use Cases including Session Id switching per request, etc.

CAS 3.8 Release

- Monitoring: JMX, Statistics publishing, support for Nagios, etc.

CAS 3.9 Release

- Support for OpenID2. This would be the first test of the new APIs to ensure we can support additional protocols

CAS 4.0 Release

- Basic SAML 2 support. "Basic" is defined as the minimal subset of required profiles to actually do something useful

CAS 4.x Releases

- Support for additional SAML 2 profiles, additional useful protocols, etc.

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-3-3-5-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, Spring, and Maven plugins)
 - 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

```
<!-- ... -->
<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, Spring, and Maven plugins)
 - Tomcat (gotta test it somewhere!)

Maven overlay build

- Retrieves the CAS war file from a repository and “overlays” your customizations on top of it.
- You only have to track changes to a handful of files.
- Upgrading to a newer version of CAS is as simple as changing its version in pom.xml.
- You will likely only have to overlay the CAS war file, or cas-server-webapp.
- Be careful when upgrading: the files you are overlaying may have been modified by Jasig in the upgraded version, too.

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.adapters.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.springframework.ldap.core.support.LdapContextSource">
  <property name="anonymousReadOnly" value="true" />
  <property name="password" value="{password_goes_here}" />
  <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, possibly like this:
`${catalina.base}/logs/cas.log`
- May want to increase the log level for troubleshooting
- Warning: setting the log level to DEBUG or higher will log users' passwords

CAS-enabling (or CASifying) Web applications

uPortal

Tomcat Manager

uPortal 2.x

- **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**

uPortal 3.x

- **Edit** `uportal-impl/src/main/resources/properties/security.properties`
 - https
 - Fully-qualified domain names
- **Edit** `uportal-war/src/main/webapp/WEB-INF/web.xml`
 - https
 - fully-qualified domain names
 - remove the `BROKEN_SECURITY_ALLOW_NON_SSL` hack
- **Deploy the changes with** `ant deploy-war`
- **Restart uPortal**
- **Details at:** <http://www.ja-sig.org/wiki/x/zwSDAQ>

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>

Proxy CAS examples

- Use CWebProxy channel to access Tomcat's Manager app
 - Publish a new CWebProxy channel and point it at `https://adam3:8443/manager/status`
 - Enter `org.jasig.portal.security.provider.cas.CasConnectionContext` in the LocalConnectionContext Implementation field
- Use WebProxy Portlet to access Tomcat's Manager app
 - Build WebProxy Portlet overlay with documented changes
 - Follow the instructions here: <http://www.jasig.org/wiki/x/uICuAQ>

Advanced Topics

Clustering

Service Registry

Single Sign-Out

CAS Clustering

- Needed mostly for redundancy, not load-handling
- No need for HttpSession replication
 - The complex instructions on the Clustering CAS page can be replaced by adding `repository-type="client"` attribute to the `flow:executor` element in `cas-servlet.xml`
- Enter each node's FQDN in `cas.properties`
- Must use distributed ticket registry
 - `JpaTicketRegistry`
 - `JBossCacheTicketRegistry`
 - `MemCacheTicketRegistry`
- Must take care of the registry cleaner
 - Default cleaner insufficient
 - Distributed cleaner available with CAS 3.4
- **Details:** <http://www.ja-sig.org/wiki/x/mYJc>

Service Registry

- “White List” of applications allowed to authenticate to CAS
- Administered using a Web UI (CAS-enabled itself)
- Requires a database
- Allows controlling of which attributes will be released to which services
- Must add the service registry URL as the first service to avoid locking out access to the service registry management interface
- **Details:** <http://www.ja-sig.org/wiki/x/5gI1>

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:

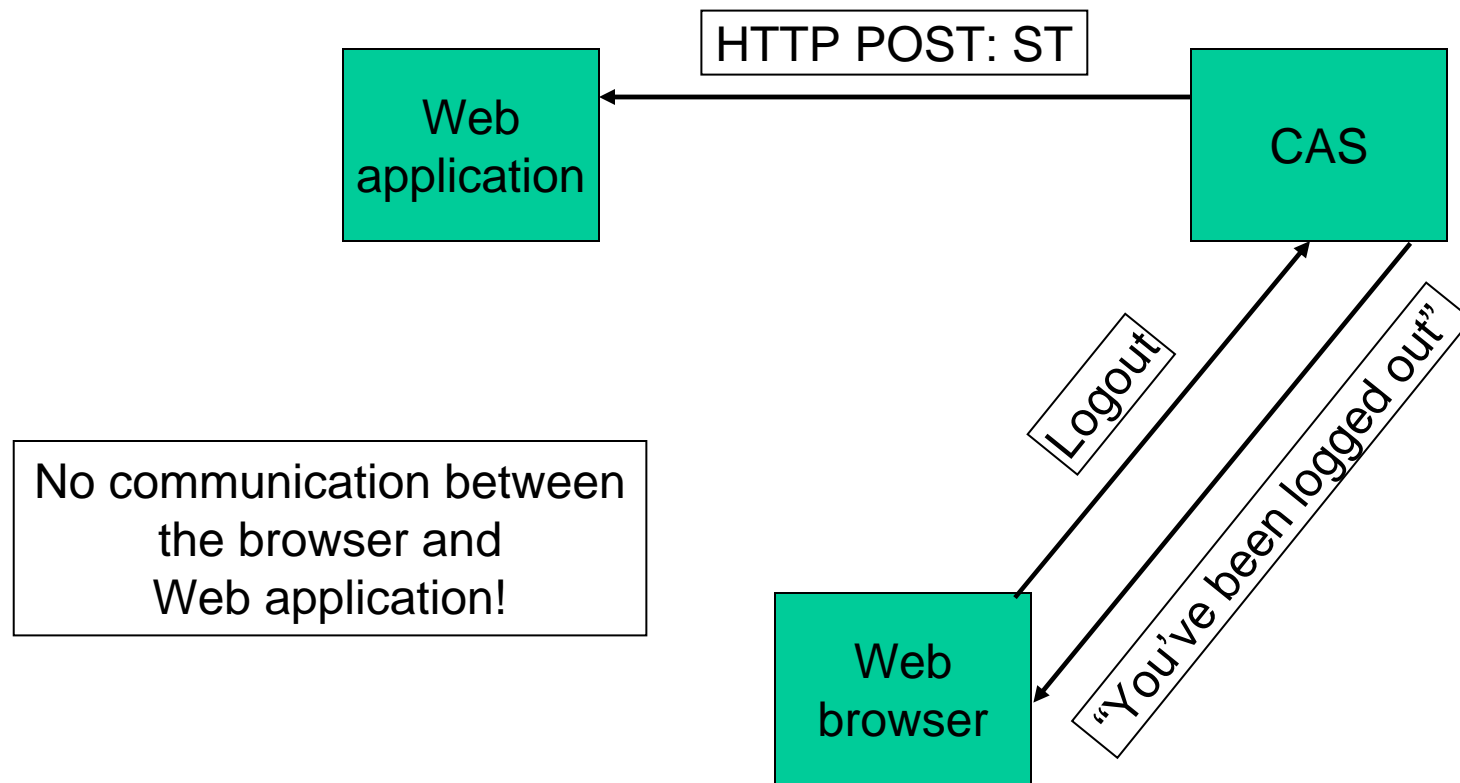
```
<bean id="userDetailsService" class="org.acegisecurity.userdetails.memory.InMemoryDaoImpl">
  <property name="userMap">
    <value>
      adam=notused,ROLE_ADMIN
    </value>
  </property>
</bean>
```

Now user "adam" is authorized to manage services.
Need to enable the database persistence, too.

Single Sign-Out

- CAS notifies services that a user has signed out of CAS
- Services must implement CAS SSOOut by “reacting” to CAS SSOOut events
- Identifies a signed out user by a service ticket that was used to log in that user
- **Details:** <http://www.ja-sig.org/wiki/x/6QN1>

Single sign-out



Questions?



Adam Rybicki

arybicki@unicon.net

www.unicon.net



UNICON