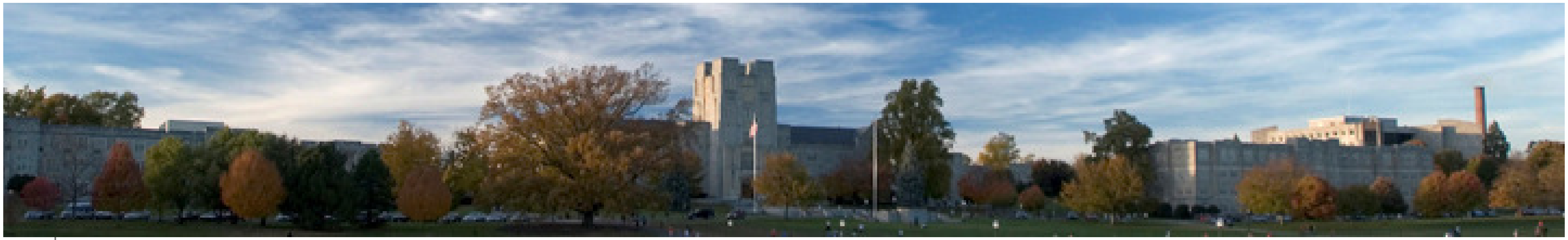
VirginiaTech
Invent the Future

CAS-anova: A University Proclaims its Love for Simplified Authentication

Ken McCrery | Project Leader, Collaborative Technologies Unit

JA-SIG 2007 Summer Conference – Denver, CO

June 25, 2007



Single Sign-on at Virginia Tech – A Look Back

Significant Events in 2002

- VT was in process of overhauling identity management
- VT moved to a central LDAP for authentication and authorization
- Internet2 issued "WebISO Web Application Agent Questionnaire"
- VT's Middleware Group Responded to WebISO Questionnaire with Authportal
- Other Responses to WebISO Questionnaire Included:
  - CAS
  - Cosign
  - Pubcookie
  - WebAuth
- VT's Portal Group adopted uPortal as its portal framework and became active in JA-SIG community

Single Sign-on at Virginia Tech – A Look Back

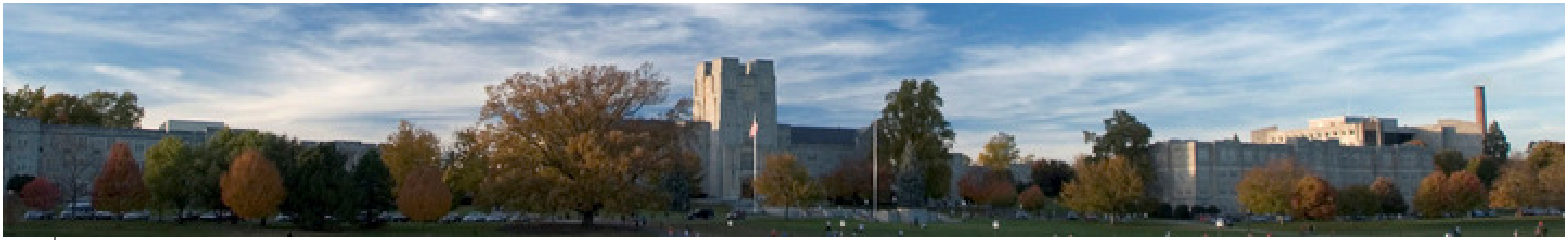
Early Implementation of Single Sign-on (SSO) at VT

- Portal Group wanted seamless transition from My VT to other enterprise applications
- Administration was very resistant to the idea of SSO due to a security vulnerability in an earlier product we used
- Approval for a limited deployment was finally made under the following conditions:
    - Sessions were to be short (30 minutes)
    - Users had to be able to log out of all applications with one click (single sign-out)
- VT's own Authportal application was selected as SSO framework
- Summer 2003 – Authportal provided SSO between My VT and Hokie SPA (Banner Self-service) and Filebox (personal file storage)

Single Sign-on at Virginia Tech – A Look Back

Significant Events in 2004

- Portal Group attended its first JA-SIG conference – Heard nothing but good things about CAS
- CAS community grew, along with support for many CAS clients
- VT's Authportal community was stagnant – Only other implementation was some high school in the US
- VT's Middleware Group was too busy and unwilling to support Authportal for Portal Group
- Portal Group made first official proposal for CAS – It was declined by management
- Portal Group began "covert" development of CAS for proof-of-concept
- Portal Group made two more official proposals – The third proposal finally got approved



Single Sign-on at Virginia Tech – Today

Customizing CAS 2.0 for VT

- **Single Sign-out**
  Provides a single URL that logs the user out of all services accessed during the session.

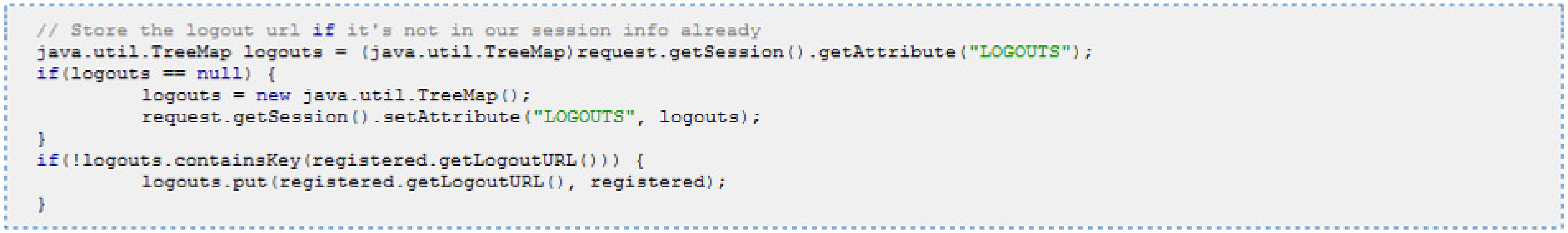
- **Registered Services**
  Restricts CAS access to specific services. Provides additional service information, such as a more user-friendly service name.

- **Service Information Includes**
  An HTML snippet that services may use to brand their CAS login page.

Single Sign-on at Virginia Tech – Today

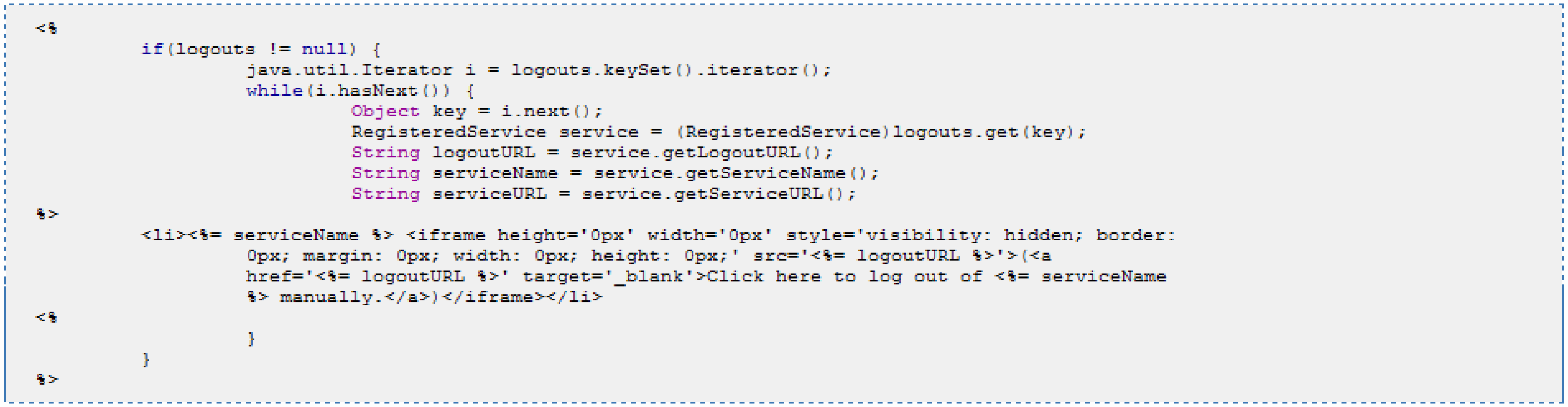
How we accomplished Single Sign-out

Single Sign-out depends on Registered Services for the logout URL. Aside from that, the only additional modifications are in JSP files. When a user is redirected to a service - either through **goService.jsp** or **redirect.jsp**, the ServiceInfo object is stored in their session. We modified **warnService.jsp** to redirect to **redirect.jsp**.



```
// Store the logout url if it's not in our session info already
java.util.TreeMap logouts = (java.util.TreeMap)request.getSession().getAttribute("LOGOUTS");
if(logouts == null) {
        logouts = new java.util.TreeMap();
        request.getSession().setAttribute("LOGOUTS", logouts);
}
if(!logouts.containsKey(registered.getLogoutURL())) {
        logouts.put(registered.getLogoutURL(), registered);
}
```

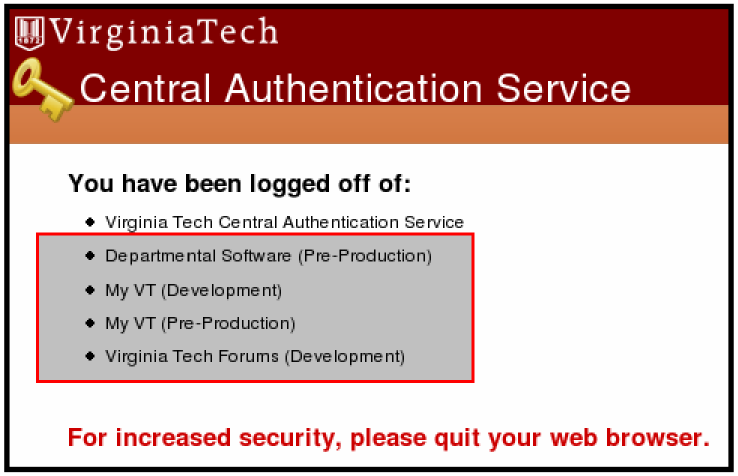
Then the **logout.jsp** is modified to read all the logout links and output inline frames with no height, so the user's browser actually hits the pages and logs off normally.



```
<%
        if(logouts != null) {
                java.util.Iterator i = logouts.keySet().iterator();
                while(i.hasNext()) {
                        Object key = i.next();
                        RegisteredService service = (RegisteredService)logouts.get(key);
                        String logoutURL = service.getLogoutURL();
                        String serviceName = service.getServiceName();
                        String serviceURL = service.getServiceURL();
%>
        <li><%= serviceName %> <iframe height='0px' width='0px' style='visibility: hidden; border:
                0px; margin: 0px; width: 0px; height: 0px;' src='<%= logoutURL %>'>(<a
                href='<%= logoutURL %>' target='_blank'>Click here to log out of <%= serviceName
                %> manually.</a>)</iframe></li>
<%
                }
        }
%>
```

Single Sign-on at Virginia Tech – Today

How we accomplished Single Sign-out



You have been logged off of:

- Virginia Tech Central Authentication Service
- Departmental Software (Pre-Production)
- My VT (Development)
- My VT (Pre-Production)
- Virginia Tech Forums (Development)

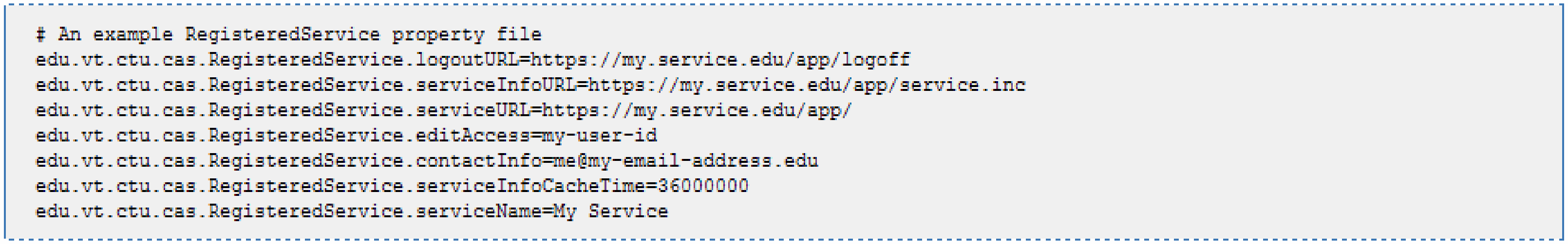
For increased security, please quit your web browser.

The log-off page informs the user of those systems for which he/she is being signed out. If the browser does not support inline frames, he/she is presented with logout links for each service.

Single Sign-on at Virginia Tech – Today

How we accomplished Registered Services

We used a combination of JSP modifications and additional Java classes. Registered services consist of the following information defined in **.properties** files somewhere. By default these files live in WEB-INF/registered/. An example properties file is as follows:



```
# An example RegisteredService property file
edu.vt.ctu.cas.RegisteredService.logoutURL=https://my.service.edu/app/logoff
edu.vt.ctu.cas.RegisteredService.serviceInfoURL=https://my.service.edu/app/service.inc
edu.vt.ctu.cas.RegisteredService.serviceURL=https://my.service.edu/app/
edu.vt.ctu.cas.RegisteredService.editAccess=my-user-id
edu.vt.ctu.cas.RegisteredService.contactInfo=me@my-email-address.edu
edu.vt.ctu.cas.RegisteredService.serviceInfoCacheTime=36000000
edu.vt.ctu.cas.RegisteredService.serviceName=My Service
```

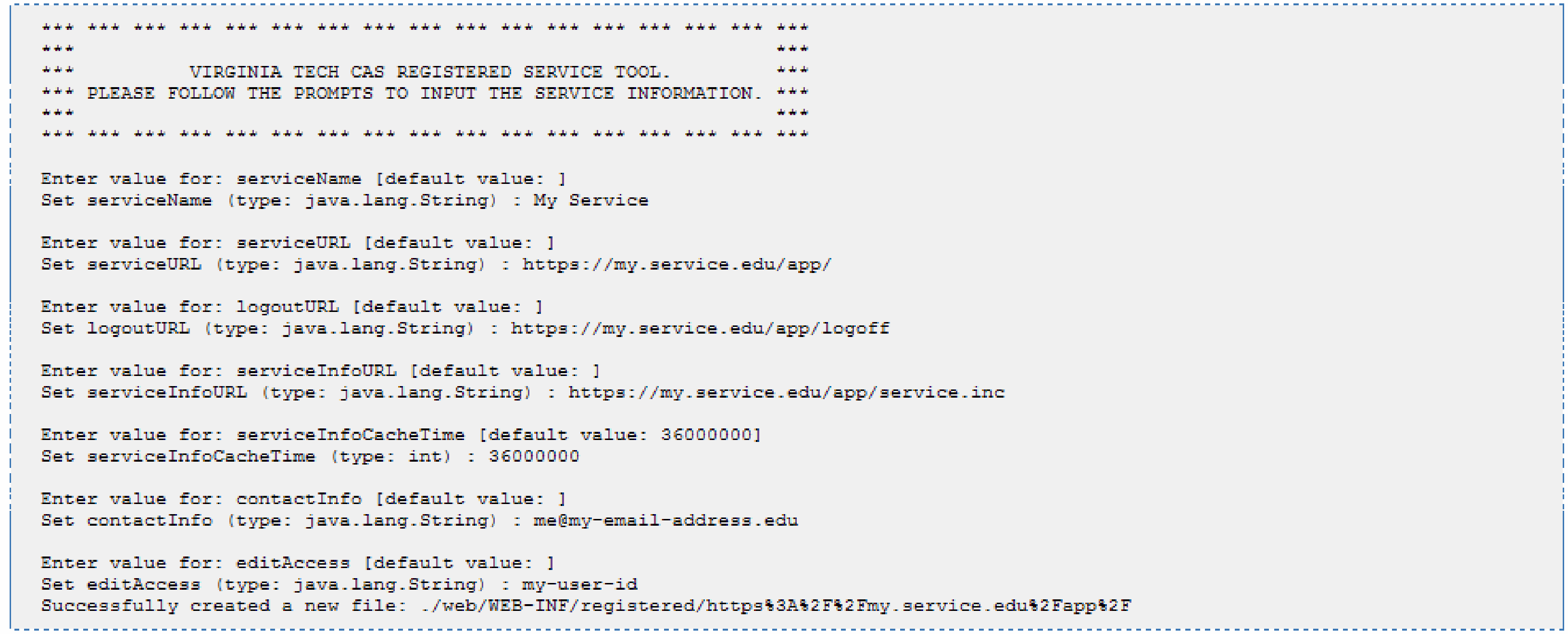
•**logoutURL** is where users are sent to log off
•**serviceInfoURL** is where to find the registered service include file (optional)
•**serviceURL** is the root URL to allow
•**editAccess** is what user IDs can access this service info file (never implemented)
•**contactInfo** is who to contact regarding this service (only used for information purposes)
•**serviceInfoCacheTime** is how long in milliseconds to cache the service info include file
•**serviceName** is what to display to the user when this service is referenced.

Single Sign-on at Virginia Tech – Today

How we accomplished Registered Services

Run the Registered Service Tool to generate the required property file. The following input would generate the previous example properties file.

addService.sh Script



```
*** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***
***                                                             ***
***          VIRGINIA TECH CAS REGISTERED SERVICE TOOL.          ***
*** PLEASE FOLLOW THE PROMPTS TO INPUT THE SERVICE INFORMATION. ***
***                                                             ***
*** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***

Enter value for: serviceName [default value: ]
Set serviceName (type: java.lang.String) : My Service

Enter value for: serviceURL [default value: ]
Set serviceURL (type: java.lang.String) : https://my.service.edu/app/

Enter value for: logoutURL [default value: ]
Set logoutURL (type: java.lang.String) : https://my.service.edu/app/logoff

Enter value for: serviceInfoURL [default value: ]
Set serviceInfoURL (type: java.lang.String) : https://my.service.edu/app/service.inc

Enter value for: serviceInfoCacheTime [default value: 36000000]
Set serviceInfoCacheTime (type: int) : 36000000

Enter value for: contactInfo [default value: ]
Set contactInfo (type: java.lang.String) : me@my-email-address.edu

Enter value for: editAccess [default value: ]
Set editAccess (type: java.lang.String) : my-user-id
Successfully created a new file: ./web/WEB-INF/registered/https%3A%2F%2Fmy.service.edu%2Fapp%2F
```

Single Sign-on at Virginia Tech – Today

How we accomplished Registered Services

The Java classes required for registered services are:

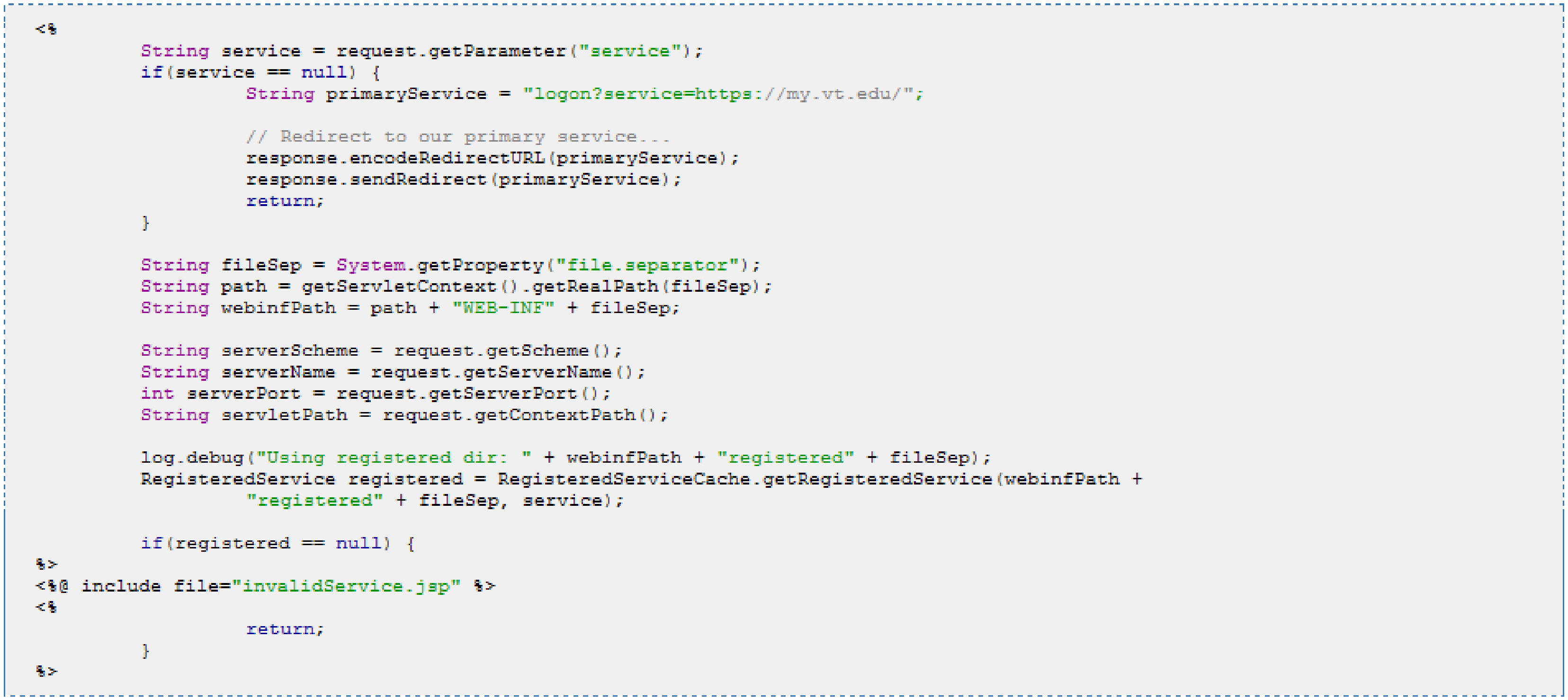
```
edu.vt.ctu.cas.RegisteredService
edu.vt.ctu.cas.RegisteredServiceCache
edu.vt.ctu.cas.RegisteredServiceFilter
edu.vt.ctu.cas.RegisteredServiceTool
```

- **RegisteredService** class is a representation of a registered service
- **RegisteredServiceFilter** provides a FilenameFilter to find the registered services' properties files
- **RegisteredServiceTool** offers a command-line utility to generate registered service property files and to load them from the file system
- **RegisteredServiceCache** class does the grunt work of caching and retrieving the registered service objects. This is what the JSP pages interact with.

Single Sign-on at Virginia Tech – Today

How we accomplished Registered Services

To disallow access, we added the following code to the 3 JSP files - **login.jsp**, **goService.jsp**, and **redirect.jsp**.



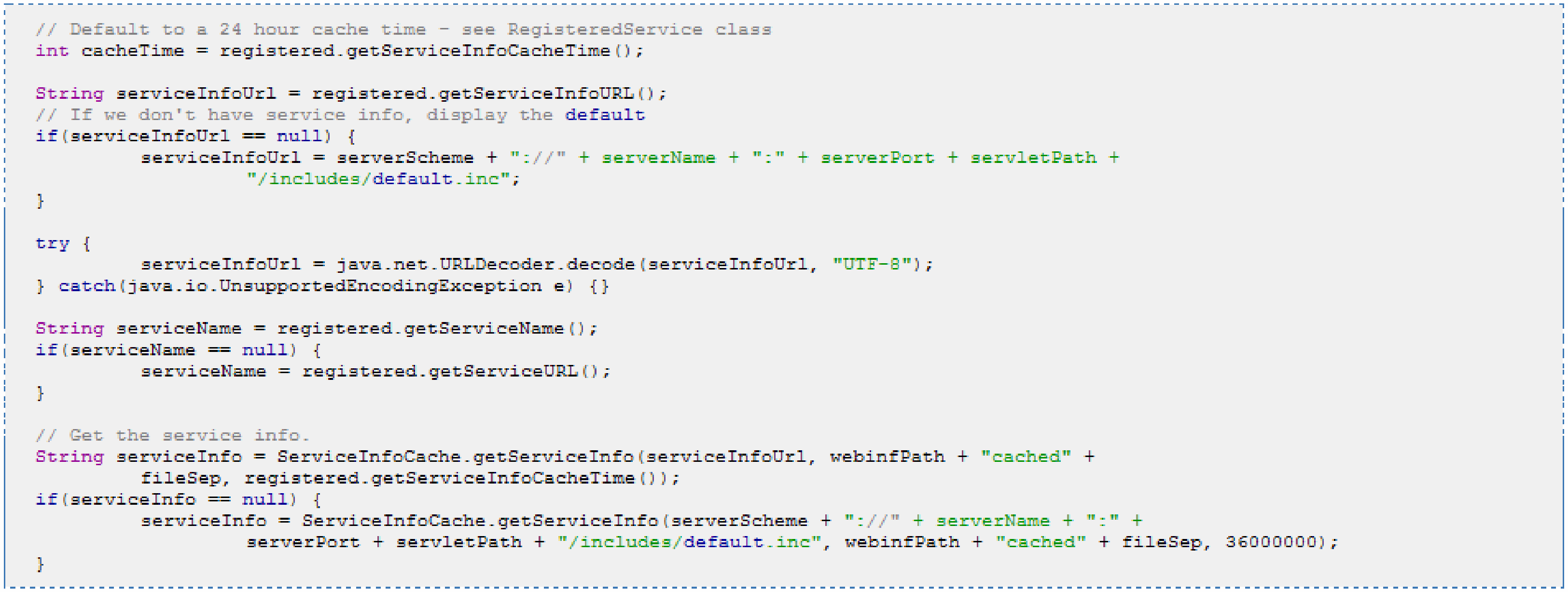
```
<%
         String service = request.getParameter("service");
         if(service == null) {
                  String primaryService = "logon?service=https://my.vt.edu/";

                  // Redirect to our primary service...
                  response.encodeRedirectURL(primaryService);
                  response.sendRedirect(primaryService);
                  return;
         }

         String fileSep = System.getProperty("file.separator");
         String path = getServletContext().getRealPath(fileSep);
         String webinfPath = path + "WEB-INF" + fileSep;

         String serverScheme = request.getScheme();
         String serverName = request.getServerName();
         int serverPort = request.getServerPort();
         String servletPath = request.getContextPath();

         log.debug("Using registered dir: " + webinfPath + "registered" + fileSep);
         RegisteredService registered = RegisteredServiceCache.getRegisteredService(webinfPath +
                  "registered" + fileSep, service);

         if(registered == null) {
%>
<%@ include file="invalidService.jsp" %>
<%
                  return;
         }
%>
```

Single Sign-on at Virginia Tech – Today

How we accomplished Service Information Includes

The service information includes files that are dependent on the services being registered so that a unique URL may be specified regarding where to find the service information. This requires modifications to the JSP files and additional Java classes.

Following the validation of the requested service, the login.jsp includes:

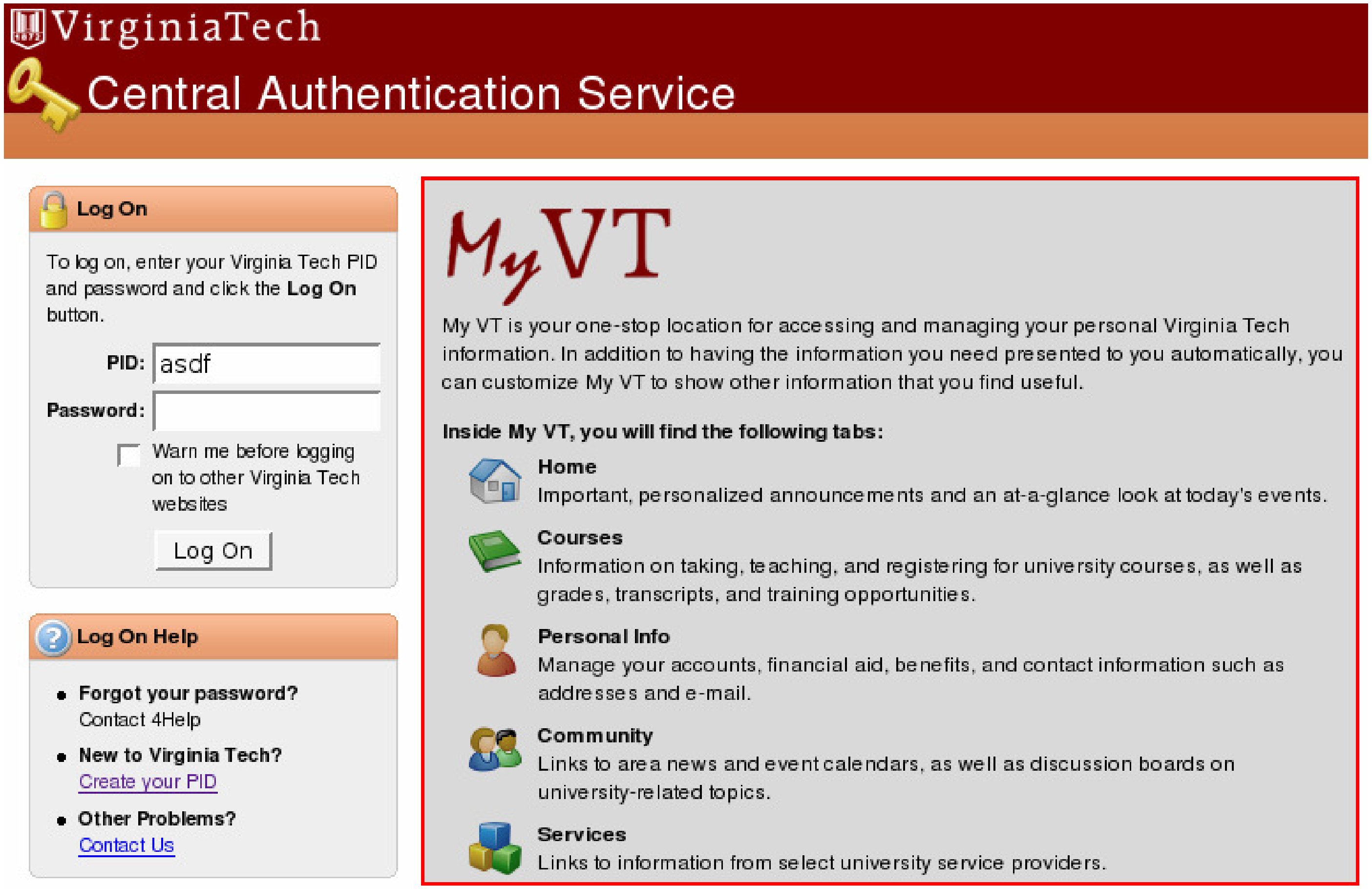


```
// Default to a 24 hour cache time - see RegisteredService class
int cacheTime = registered.getServiceInfoCacheTime();

String serviceInfoUrl = registered.getServiceInfoURL();
// If we don't have service info, display the default
if(serviceInfoUrl == null) {
        serviceInfoUrl = serverScheme + "://" + serverName + ":" + serverPort + servletPath +
                "/includes/default.inc";
}

try {
        serviceInfoUrl = java.net.URLDecoder.decode(serviceInfoUrl, "UTF-8");
} catch(java.io.UnsupportedEncodingException e) {}

String serviceName = registered.getServiceName();
if(serviceName == null) {
        serviceName = registered.getServiceURL();
}

// Get the service info.
String serviceInfo = ServiceInfoCache.getServiceInfo(serviceInfoUrl, webinfPath + "cached" +
        fileSep, registered.getServiceInfoCacheTime());
if(serviceInfo == null) {
        serviceInfo = ServiceInfoCache.getServiceInfo(serverScheme + "://" + serverName + ":" +
                serverPort + servletPath + "/includes/default.inc", webinfPath + "cached" + fileSep, 36000000);
}
```

Single Sign-on at Virginia Tech – Today

How we accomplished Service Information Includes

Example for My VT Registered Service:



VirginiaTech

Central Authentication Service

Log On

To log on, enter your Virginia Tech PID and password and click the **Log On** button.

PID: asdf

Password:

□ Warn me before logging on to other Virginia Tech websites

Log On Help

- **Forgot your password?** Contact 4Help
- **New to Virginia Tech?** Create your PID
- **Other Problems?** Contact Us

MyVT

My VT is your one-stop location for accessing and managing your personal Virginia Tech information. In addition to having the information you need presented to you automatically, you can customize My VT to show other information that you find useful.

Inside My VT, you will find the following tabs:

Home
Important, personalized announcements and an at-a-glance look at today's events.

Courses
Information on taking, teaching, and registering for university courses, as well as grades, transcripts, and training opportunities.

Personal Info
Manage your accounts, financial aid, benefits, and contact information such as addresses and e-mail.

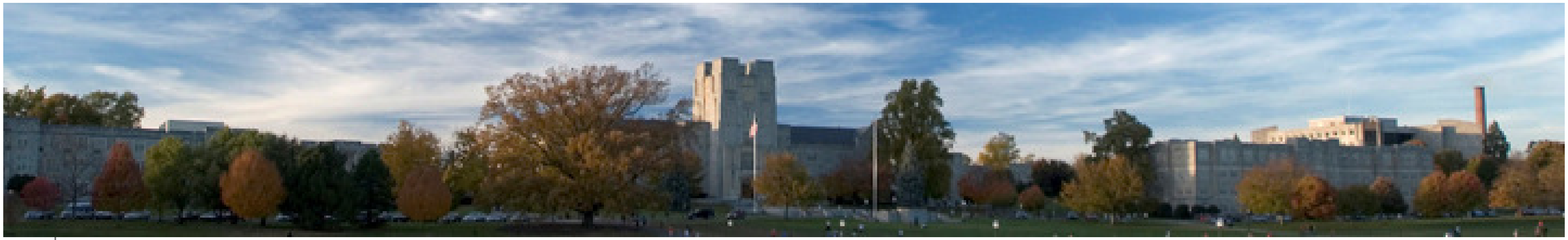
Community
Links to area news and event calendars, as well as discussion boards on university-related topics.

Services
Links to information from select university service providers.

Single Sign-on at Virginia Tech – Today

Summer 2005 – CAS 2.0 goes into production at VT

- Immediate Success!
  - CAS was easy to deploy
  - Previous Authportal clients were simple to convert
  - Small footprint of CAS was fast and efficient
  - System proved to be very stable and reliable
- Rapid Adoption
  - CAS integration proved to be so easy that many development groups converted all their authentication mechanisms
  - CAS was successfully integrated with a number of external vendor products
  - CAS is currently supporting approximately 50 clients



Single Sign-on at Virginia Tech – Future

Upgrading to CAS 3.0

- Been developing for the last few months
- VT wanted to get back in line with baseline code:
  - Adopted 3.0 support for Registered Services
  - Extended CAS' RegisteredServices class instead of using our own
  - Created DetailedRegisteredService class to handle VT's additional attributes
  - Created a JDBC-backed interface for Registered Services
  - Continuing to use our own Single Sign-out
- Adding load balancing to ensure high availability
- Turned over development and maintenance responsibilities to VT's Middleware Group
- Plan to go live with CAS 3.0 on July 15, 2007

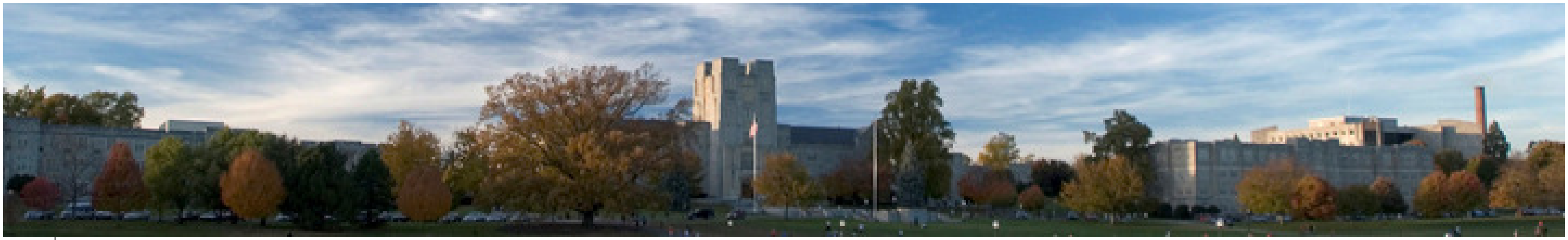
VirginiaTech
Invent the Future

Single Sign-on at Virginia Tech – Future

Additional Plans

- Virginia Tech currently has a Personal Digital Certificate pilot. We plan to test X.509 certificate authentication with CAS 3.0.
- Going to adopt CAS implementation of Single Sign-out once it becomes available in 3.1
- Working on Shibboleth project and hope to integrate with CAS
- Intend to provide authorization data to CAS clients once SAML is available in CAS 3.1

- **The VT Security Office plans to mandate that all Web applications use CAS once authorization data is available.**



Additional Resources

JA-SIG Central Authentication Service (CAS)
http://www.ja-sig.org/products/cas/

Virginia Tech CAS
http://www.ctu.vt.edu/cas

Internet2 WebISO Working Group
http://middleware.internet2.edu/webiso/

VirginiaTech
Invent the Future

VirginiaTech
Invent the Future

Thank You!

Ken McCrery
Project Leader, Collaborative Technologies Unit
Virginia Polytechnic Institute and State University
kmccrery@vt.edu

June 25, 2007