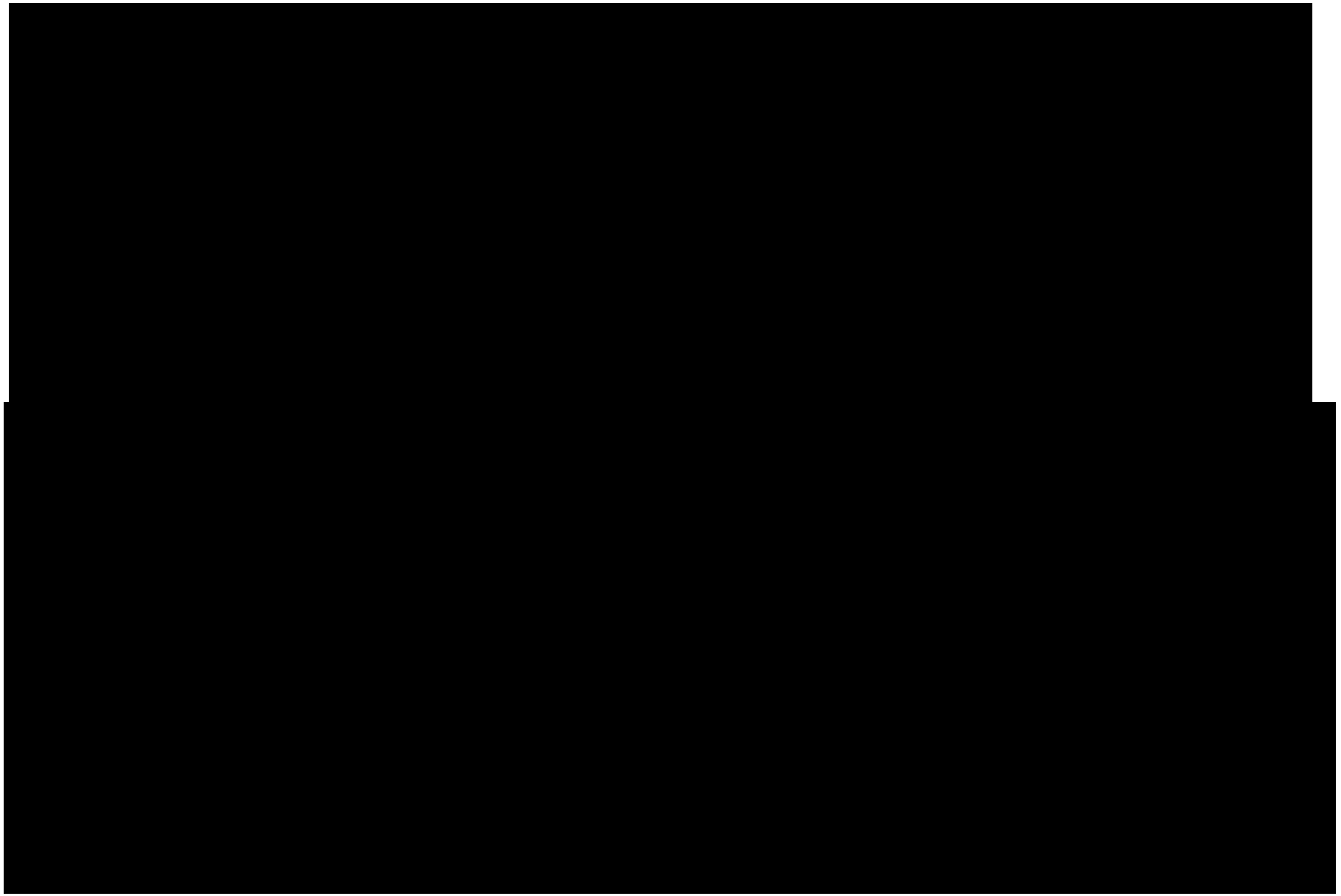
Authentication

- ***Principles***
- ***Architecture***
- ***Functionality***
- ***Configuration***
- ***Future plans***

Principles

- In mutual authentication the mechanism is symmetric, both sides do the same steps
  - Load credentials – cert, private key
  - Load trust settings – CA certs, CRLs
  - Send credentials
  - Verify received credentials

Principles

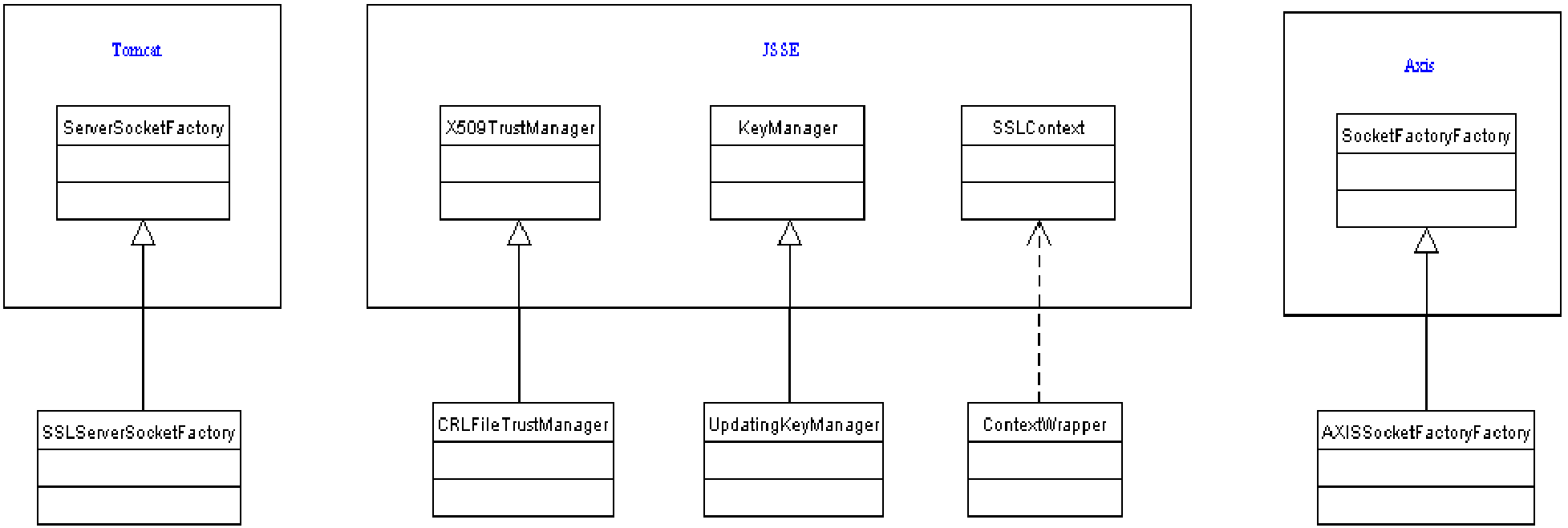


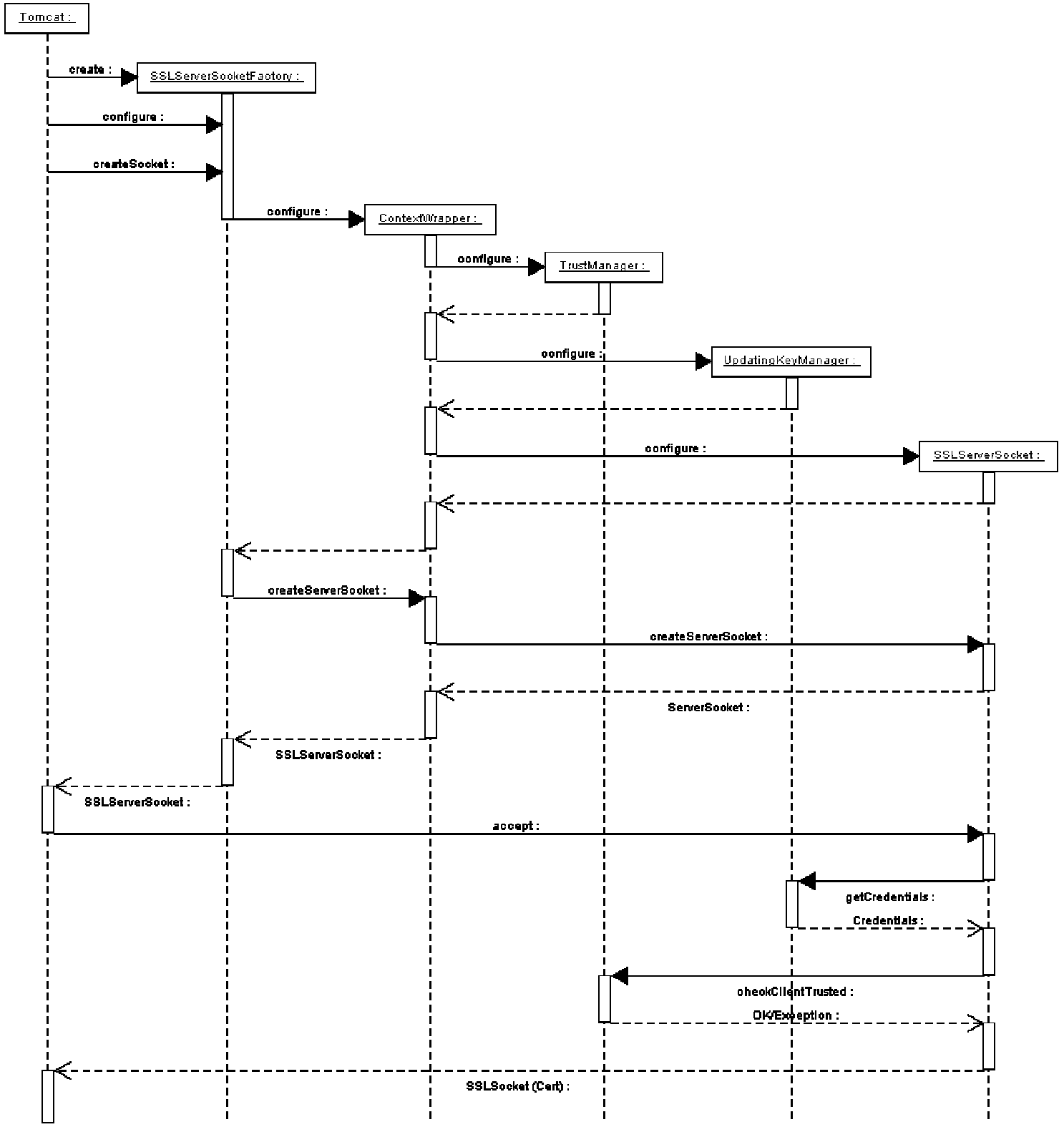
Architecture

- ***Based on Java Secure Sockets Extension, which provides good hooks for customizing***
  - ***TrustManager – Certificate checking and trust settings***
  - ***KeyManager – Credentials storage***
- Tomcat and Axis also provide mechanisms for interfacing
  - SSLServerSocketFactory for tomcat
  - AxisSocketFactoryFactory for Axis

Architecture

- The main classes used for hooking up




Tomcat :
create :
SSLServerSocketFactory :
configure :
createSocket :
configure :
ContextWrapper :
configure :
TrustManager :
configure :
UpdatingKeyManager :
configure :
SSLServerSocket :
createServerSocket :
createServerSocket :
ServerSocket :
SSLServerSocket :
SSLServerSocket :
accept :
getCredentials :
Credentials :
checkClientTrusted :
OK/Exception :
SSLSocket (Cert) :

Functionality

- Credentials (certificate path + private key)
  - Java key store (JKS), PKCS12, cert+key, proxy
  - Optionally updated periodically
  - Password callback on client side for private key file
- CA certificates
  - JKS, PKCS12, certs in a directory
- Certificate revocation lists
  - Files in a directory

Functionality

- Cert path validation, revocation cheking
- Naming constraint for proxy certs
  - First cert without CA flag considered user cert
  - The DN of the certs under user cert must contain the DN of the previous cert
    - Example: “C=CH, O=CERN, CN=Bob”
      - Can sign “C=CH, O=CERN, CN=Bob, CN=proxy”
      - Can sign “C=CH, O=CERN, CN=Bob, CN=Job xyz”
      - Can't sign “C=CH, O=CERN, CN=John”

Configuration (tomcat)

- ***Setup credentials, CA certs, CRLs***
- ***Copy the jar containing the security classes to tomcat/server/lib***
- ***Edit tomcat/conf/server.xml***

```
<!-- Define an SSL HTTP/1.1 Connector on port 8443 -->
    <Connector className="org.apache.catalina.connector.http.HttpConnector"
               port="8443" minProcessors="5" maxProcessors="75"
               enableLookups="true"
               acceptCount="10" debug="0" scheme="https" secure="true">
      <Factory
     className="org.edg.security.trustmanager.tomcat.SSLServerSocketFactory"
               caFiles="/etc/grid-security/certificates/*.0"
               clientAuth="true"
               crlFiles="/etc/grid-security/certificates/*.r0"
               crlRequired="false"
               crlUpdateInterval="1h"
               gridProxyFile="/home/hahkala/.globus/server.proxy"
               credentialsUpdateInterval="10min"
               logConf="/home/hahkala/log4j.conf"
               protocol="TLS"/>
    </Connector>
```

Configuration (HTTP client)

```
HTTPJavaClient gridclient = new HTTPJavaClient();

try {
    // load config file
    FileInputStream configStream = new FileInputStream("authentication.conf");
    Properties config = new Properties(configStream);
    gridclient.init(config);

    // get page
    result = gridclient.getURL("https://localhost:8443/gridtest.html", null);
    if(!result.getResultCode().equals("200 OK HTTP/1.1"))
       throw new Exception(“Error while connecting to server: ” + result.getResultCode());

    // get data
    InputStream body = result.getBody();

    handle(body);
} catch (Exception exc) {
    ERROR!
}
```

Configuration (HTTP client)

- ***Example configuration file***

```
# Enable use of Globus grid proxy certificate. Specify the file containing
# the grid proxy.
gridProxyFile=/tmp/x509up_u1234

# File to configure log4j
log4jConfFile=/opt/edg/edg-java-security/test/httpclient/log4j.conf

# File where to append the output of log4j
log4jFile=/tmp/authentication.log
```

Future additions

- Delegation
- CoG integration
- Tomcat versions 4.x, x>0; 5.y, y≥0
- OGSA, WS-security
- Reverse delegation?
  - For MyProxy, cert renewal, OCR?
- Site admin tool?