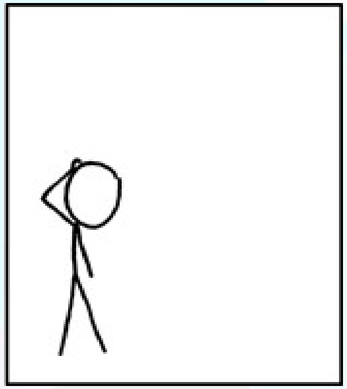
inside

jGridstart

getting to know the application and its components,
dependencies, philoshopy, ideas, bugs, adoption,
problems encountered, their workarounds, solutions,
etc. etc. etc. etc. etc. etc. etc. etc. etc. etc. etc. etc.

Willem van Engen, Nikhef, 13th of December 2012



http://xkcd.com/508/

jGridstart ... what?

goal

use the grid

jGridstart's goal

get ready to use the grid

& stay

jGridstart's goal

get ready & stay to use the grid

Keyholder is indeed
Jan Klaassen
Dutch Grid

(your secret!)

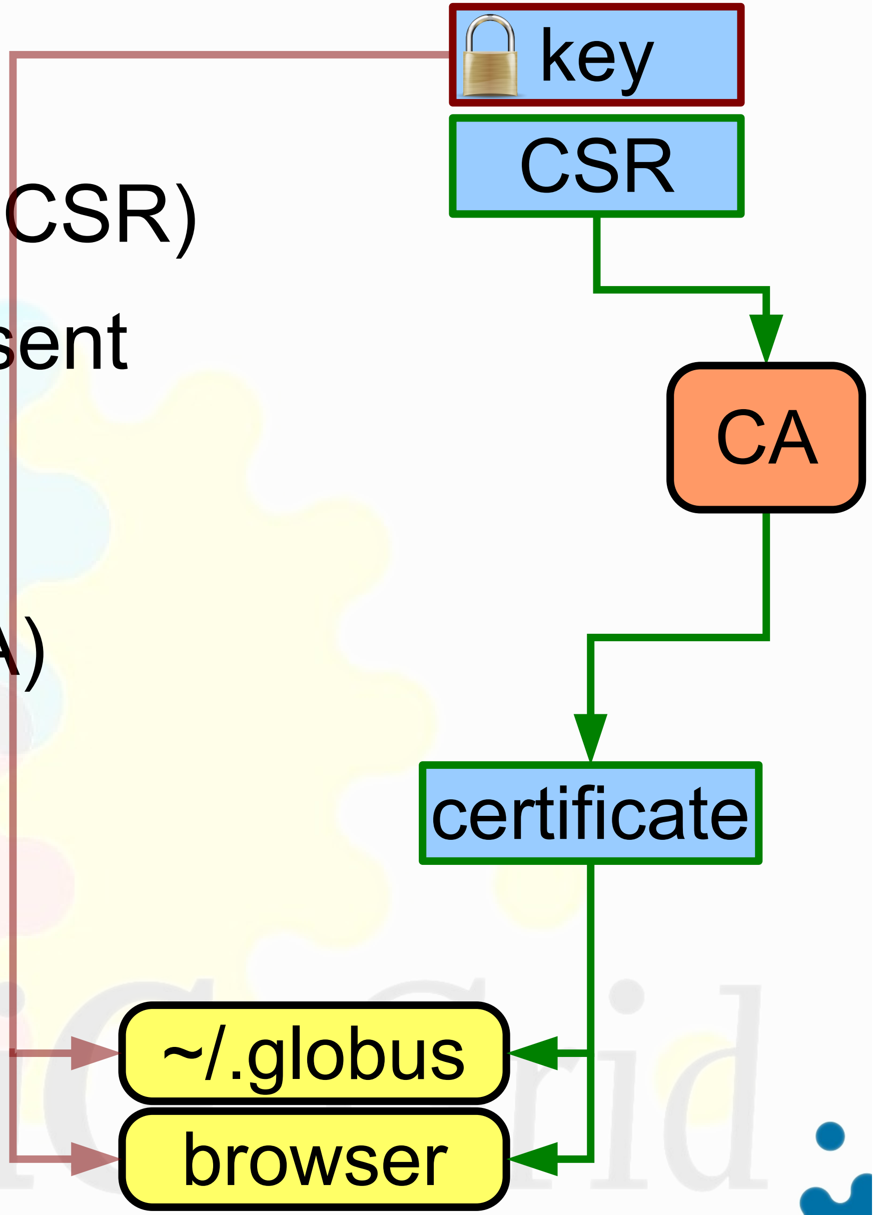
```
-----BEGIN CERTIFICATE-----
MZIEdyCCA12gAwIBAgICCmcVDQYJKoZxXxcNAQEFBQ
DzANBgN6BFoABk5JS0hFRjSDMDAGA1UEAxM6TklLS8
eSBjZXJGa1Zbv2F0aW9uIGF1dGgwHhcNMDkwbvIxMD
NjM4WjBQMRmwzAYDVQQdDzgkdXRjaGdyaWQxDjAMBg
VQQKDAZuaWtoZWYxGTAXBgNVBAMMEFdpbGxlbSB2YW
```

```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,26c639f0a01b0186

1DEdWeCQldFVZCsSAsBqC2+Tbo2DzQB0/4nFGDvka8
ODp7iG5ZxVAV/43Z2dUGYvkciO37/6dArm3b0e/Cru
```

The process

- User generates key & certificate signing request (CSR)
- Certificate signing request is sent to certificate authority (CA)
- User brings proof-of-identity to registration authority (RA)
- CA signs the CSR
- User retrieves certificate and installs it



Desired user experience

1. Easily discoverable entry point (like web page)
2. Enter details (once, if needed)
3. Submit request
4. Verify identity
5. Start using the grid

Desired user experience

1. Easily discoverable entry point (like web page)
2. Enter details (once)
3. Generate certificate and submit request
4. *Print and sign PDF form*
5. Verify identity at RA
6. *Install certificate into ~/.globus & browser*
7. *Signup for VOs*
8. Start using the grid

philosophy

user

philosophy

- The user is always right
  - ...
  - when he doesn't know what to do next
  - when it is unclear what an action would do
  - when it is unclear what an action has done

philosophy

- The user is always right
  - ...
  - when he doesn't know what to do next
  - when it is unclear what an action would do
  - when it is unclear what an action has done

we aren't perfect

but I think we could use a little more of this philosophy

The User's Bill of Rights.

perspective

installation

compliance

instruction

control

dependencies

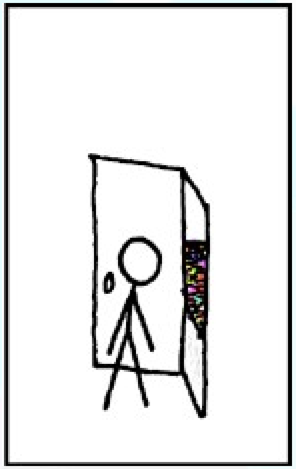
feedback

scope

assistance

usability

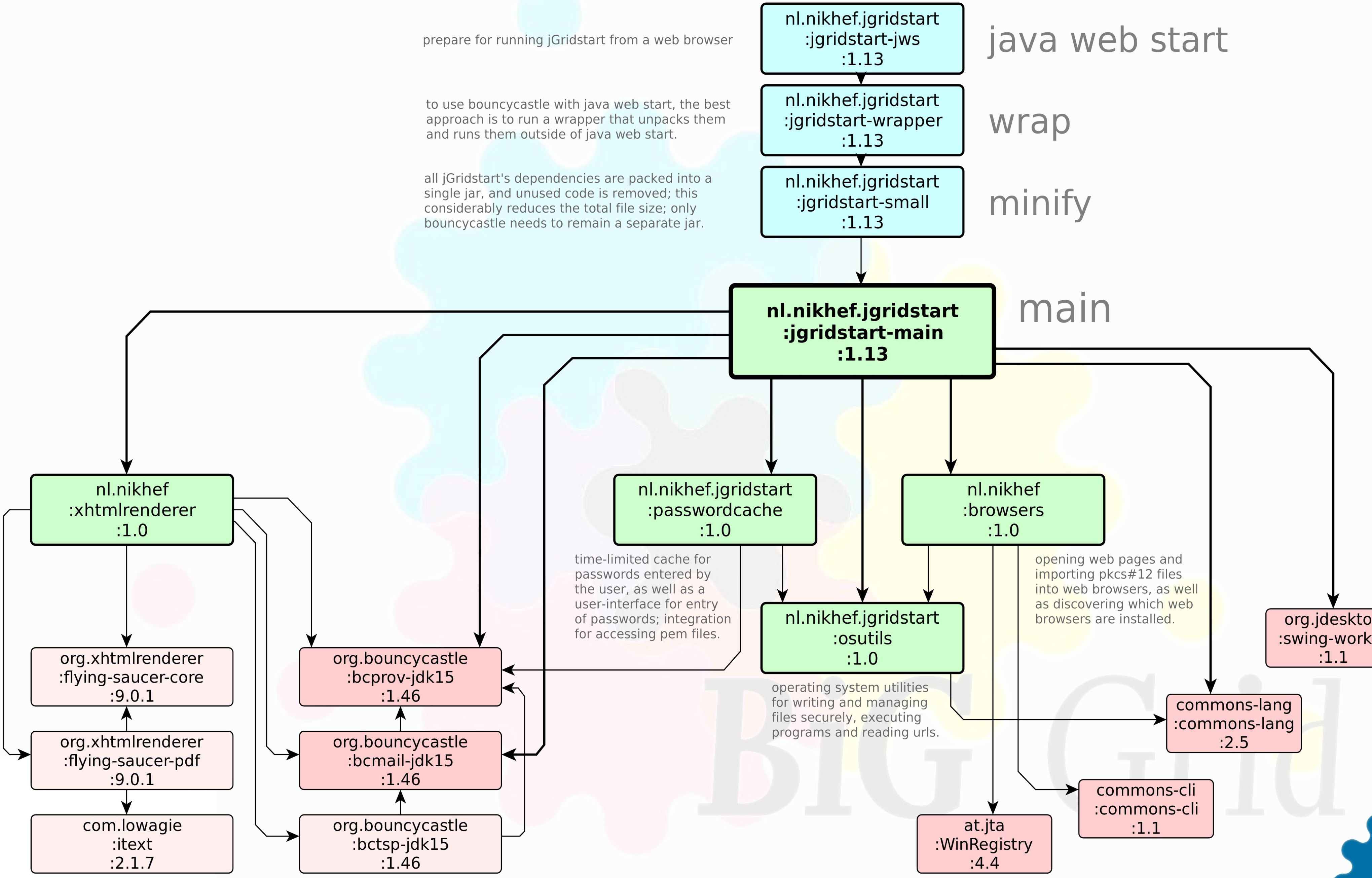
http://doi.acm.org/10.1145/290133.290144
http://www-01.ibm.com/software/ucd/designconcepts/userrights.html



http://xkcd.com/150/

now show me what's inside!

modules & dependencies



prepare for running jGridstart from a web browser

nl.nikhef.jgridstart
:jgridstart-jws
:1.13

java web start

to use bouncycastle with java web start, the best approach is to run a wrapper that unpacks them and runs them outside of java web start.

nl.nikhef.jgridstart
:jgridstart-wrapper
:1.13

wrap

all jGridstart's dependencies are packed into a single jar, and unused code is removed; this considerably reduces the total file size; only bouncycastle needs to remain a separate jar.

nl.nikhef.jgridstart
:jgridstart-small
:1.13

minify

nl.nikhef.jgridstart
:jgridstart-main
:1.13

main

nl.nikhef
:xhtmlrenderer
:1.0

nl.nikhef.jgridstart
:passwordcache
:1.0

time-limited cache for passwords entered by the user, as well as a user-interface for entry of passwords; integration for accessing pem files.

nl.nikhef
:browsers
:1.0

opening web pages and importing pkcs#12 files into web browsers, as well as discovering which web browsers are installed.

nl.nikhef.jgridstart
:osutils
:1.0

operating system utilities for writing and managing files securely, executing programs and reading urls.

org.jdesktop
:swing-worker
:1.1

org.xhtmlrenderer
:flying-saucer-core
:9.0.1

org.xhtmlrenderer
:flying-saucer-pdf
:9.0.1

com.lowagie
:itext
:2.1.7

org.bouncycastle
:bcprov-jdk15
:1.46

org.bouncycastle
:bcmail-jdk15
:1.46

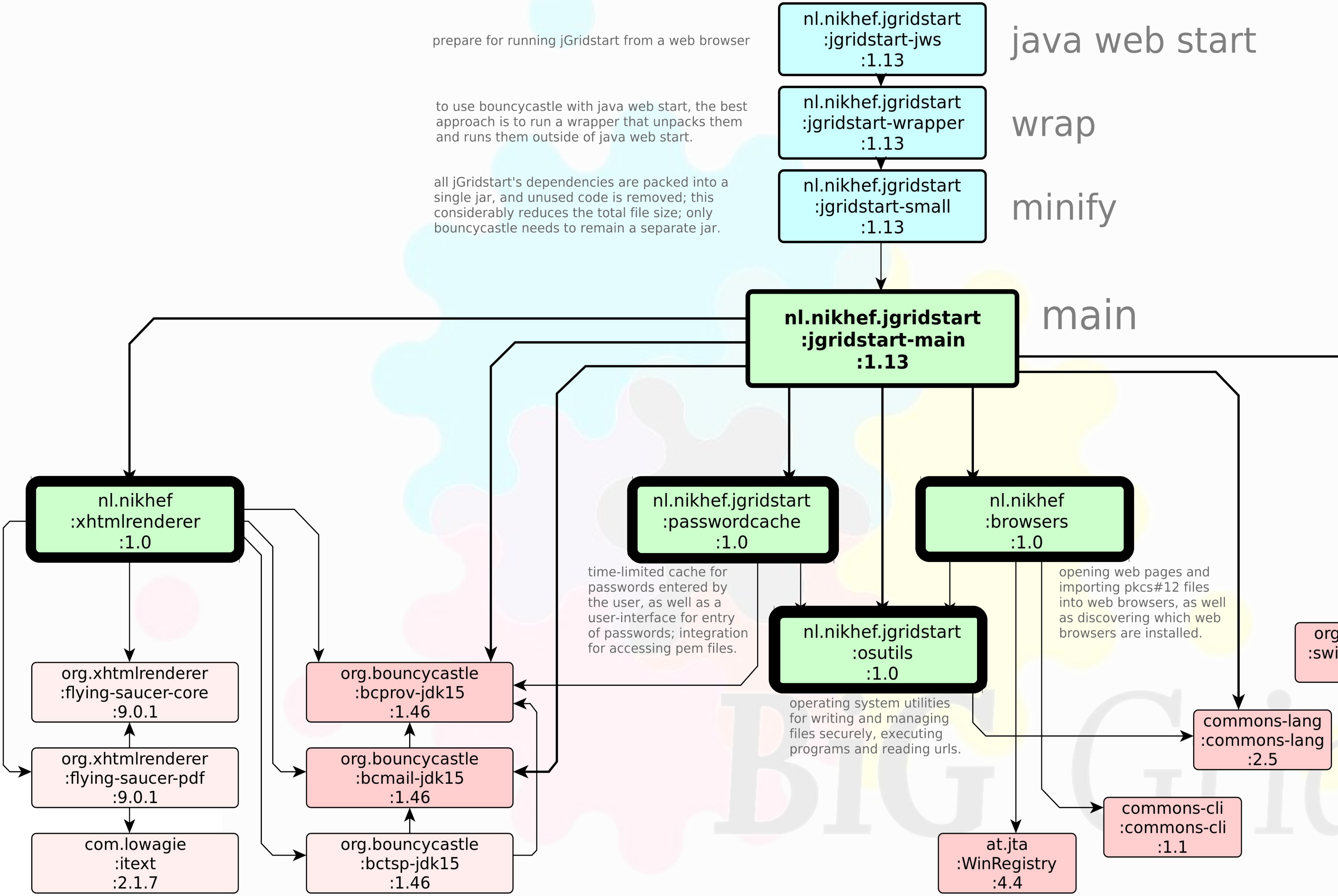
org.bouncycastle
:bctsp-jdk15
:1.46

commons-lang
:commons-lang
:2.5

commons-cli
:commons-cli
:1.1

at.jta
:WinRegistry
:4.4

modules & dependencies



prepare for running jGridstart from a web browser

nl.nikhef.jgridstart :jgridstart-jws :1.13

java web start

to use bouncycastle with java web start, the best approach is to run a wrapper that unpacks them and runs them outside of java web start.

nl.nikhef.jgridstart :jgridstart-wrapper :1.13

wrap

all jGridstart's dependencies are packed into a single jar, and unused code is removed; this considerably reduces the total file size; only bouncycastle needs to remain a separate jar.

nl.nikhef.jgridstart :jgridstart-small :1.13

minify

nl.nikhef.jgridstart :jgridstart-main :1.13

main

nl.nikhef :xhtmlrenderer :1.0

nl.nikhef.jgridstart :passwordcache :1.0

time-limited cache for passwords entered by the user, as well as a user-interface for entry of passwords; integration for accessing pem files.

nl.nikhef :browsers :1.0

opening web pages and importing pkcs#12 files into web browsers, as well as discovering which web browsers are installed.

nl.nikhef.jgridstart :osutils :1.0

operating system utilities for writing and managing files securely, executing programs and reading urls.

org.jdesktop :swing-worker :1.1

org.xhtmlrenderer :flying-saucer-core :9.0.1

org.xhtmlrenderer :flying-saucer-pdf :9.0.1

com.lowagie :itext :2.1.7

org.bouncycastle :bcprov-jdk15 :1.46

org.bouncycastle :bcmail-jdk15 :1.46

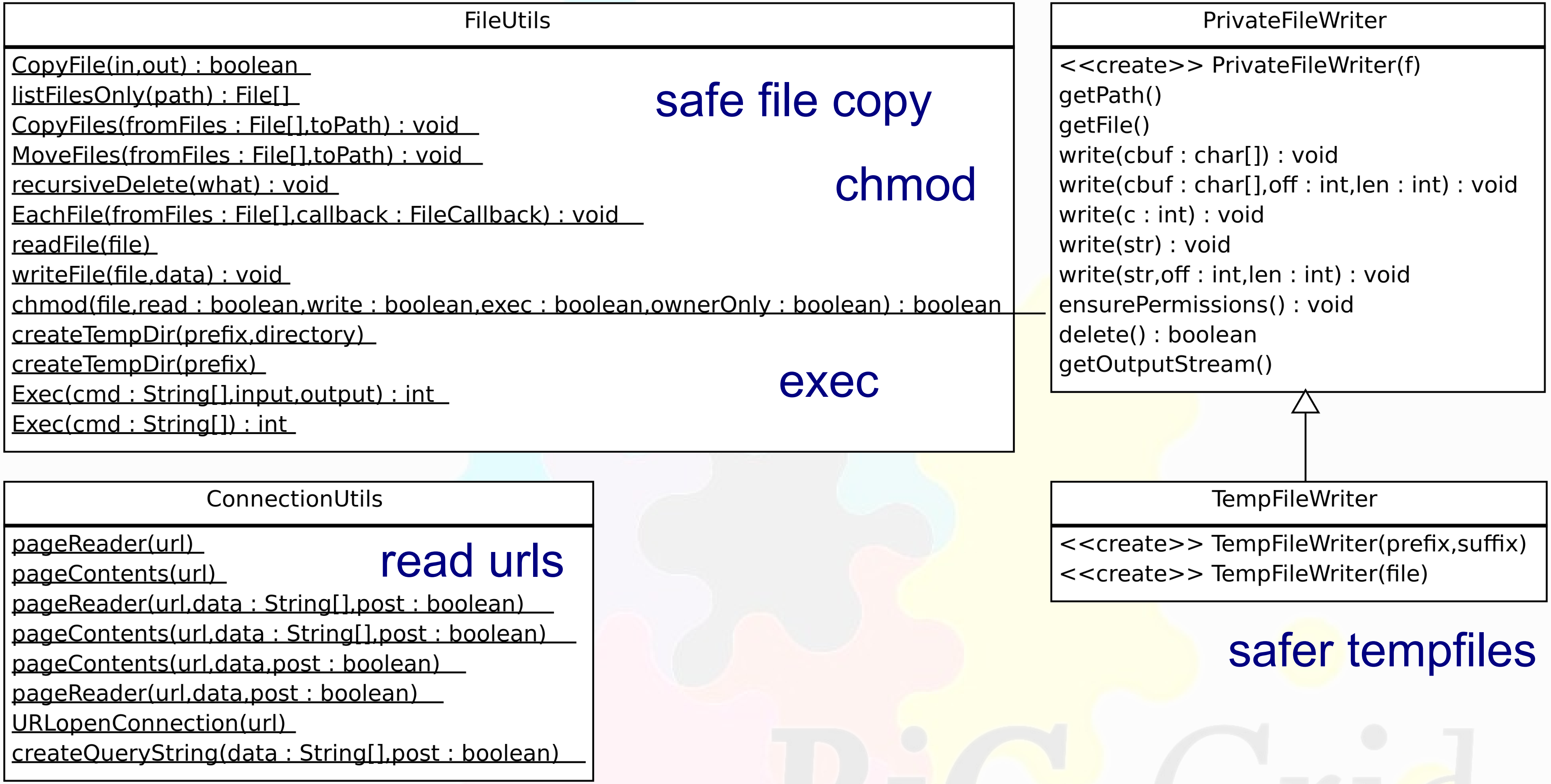
org.bouncycastle :bctsp-jdk15 :1.46

commons-lang :commons-lang :2.5

commons-cli :commons-cli :1.1

at.jta :WinRegistry :4.4

module: osutils



FileUtils

- CopyFile(in,out) : boolean
- listFilesOnly(path) : File[]
- CopyFiles(fromFiles : File[],toPath) : void
- MoveFiles(fromFiles : File[],toPath) : void
- recursiveDelete(what) : void
- EachFile(fromFiles : File[],callback : FileCallback) : void
- readFile(file)
- writeFile(file,data) : void
- chmod(file,read : boolean,write : boolean,exec : boolean,ownerOnly : boolean) : boolean
- createTempDir(prefix,directory)
- createTempDir(prefix)
- Exec(cmd : String[],input,output) : int
- Exec(cmd : String[]) : int

safe file copy

chmod

exec

PrivateFileWriter

- <<create>> PrivateFileWriter(f)
- getPath()
- getFile()
- write(cbuf : char[]) : void
- write(cbuf : char[],off : int,len : int) : void
- write(c : int) : void
- write(str) : void
- write(str,off : int,len : int) : void
- ensurePermissions() : void
- delete() : boolean
- getOutputStream()

ConnectionUtils

- pageReader(url)
- pageContents(url)
- pageReader(url,data : String[],post : boolean)
- pageContents(url,data : String[],post : boolean)
- pageContents(url,data,post : boolean)
- pageReader(url,data,post : boolean)
- URLopenConnection(url)
- createQueryString(data : String[],post : boolean)

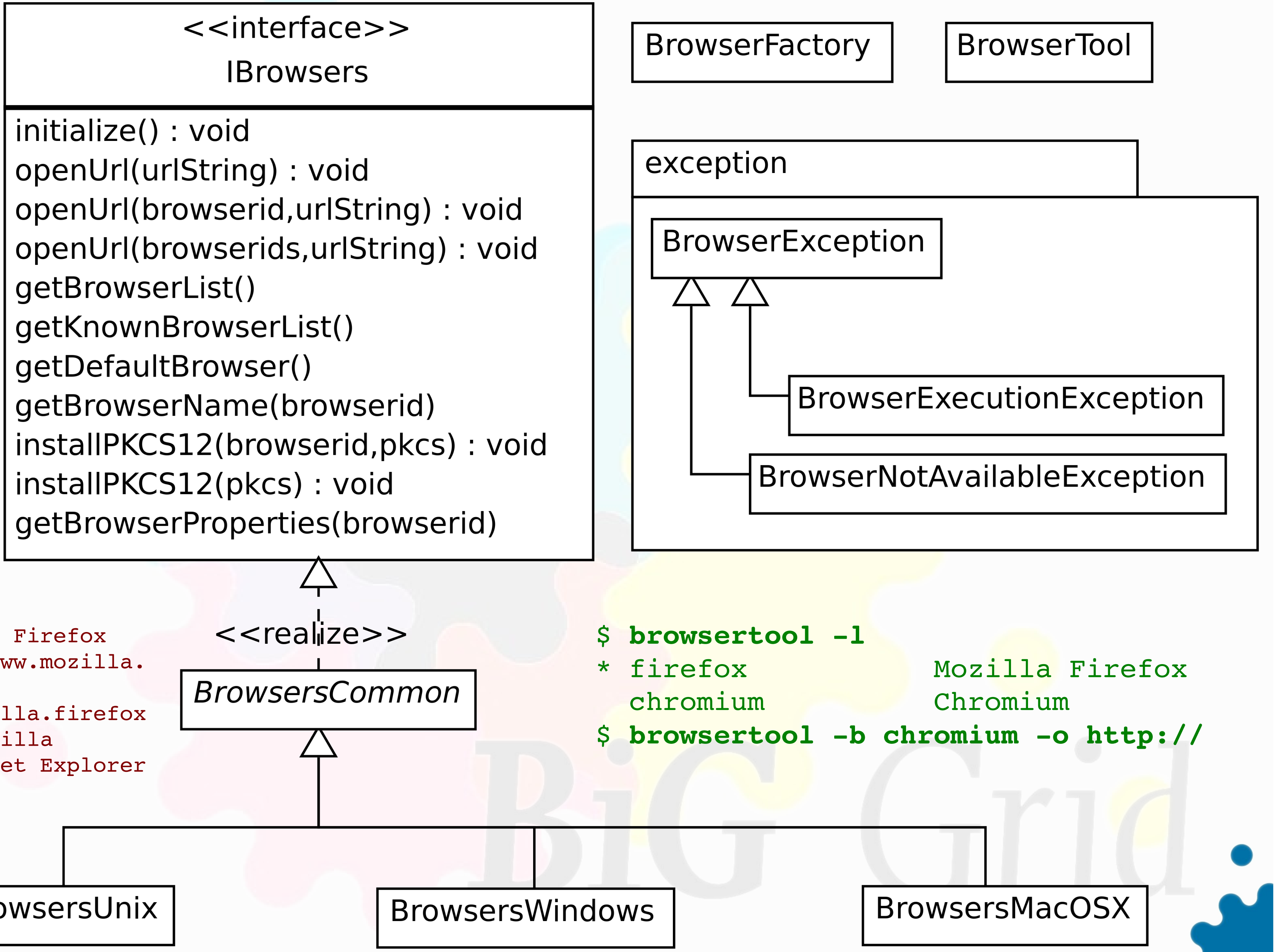
read urls

TempFileWriter

- <<create>> TempFileWriter(prefix,suffix)
- <<create>> TempFileWriter(file)

safer tempfiles

module: browsers



<<interface>>
IBrowsers

initialize() : void
openUrl(urlString) : void
openUrl(browserid,urlString) : void
openUrl(browserids,urlString) : void
getBrowserList()
getKnownBrowserList()
getDefaultBrowser()
getBrowserName(browserid)
installPKCS12(browserid,pkcs) : void
installPKCS12(pkcs) : void
getBrowserProperties(browserid)

BrowserFactory

BrowserTool

exception

BrowserException

BrowserExecutionException

BrowserNotAvailableException

<<realize>>

BrowsersCommon

BrowsersUnix

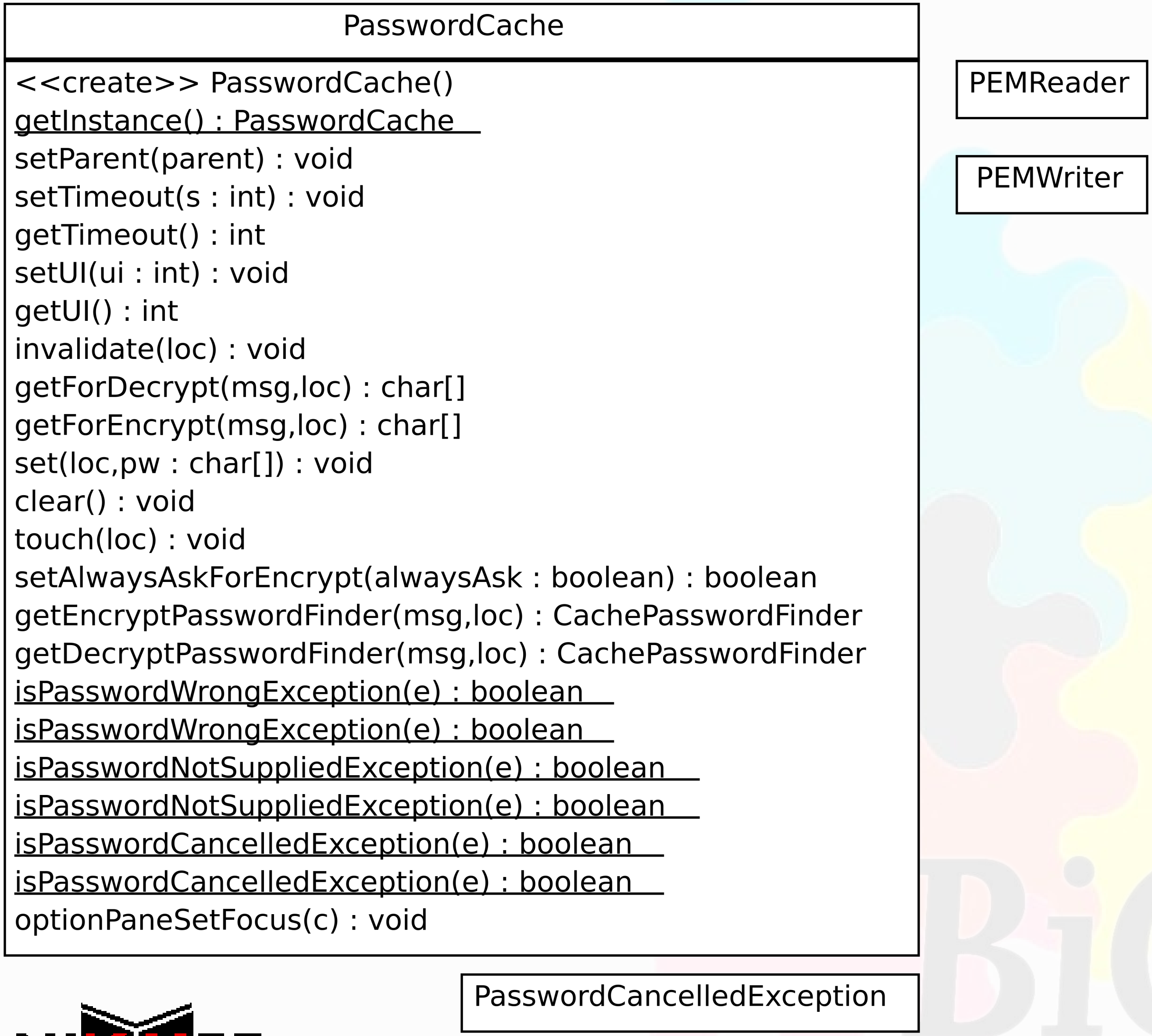
BrowsersWindows

BrowsersMacOSX

```
# browsers.properties
firefox.desc = Mozilla Firefox
firefox.url = http://www.mozilla.
firefox.exe = firefox
firefox.uti = org.mozilla.firefox
firefox.certinst = mozilla
explorer.desc = Internet Explorer
```

```
$ browsertool -l
* firefox            Mozilla Firefox
  chromium           Chromium
$ browsertool -b chromium -o http://
```

module: passwordcache



PasswordCache

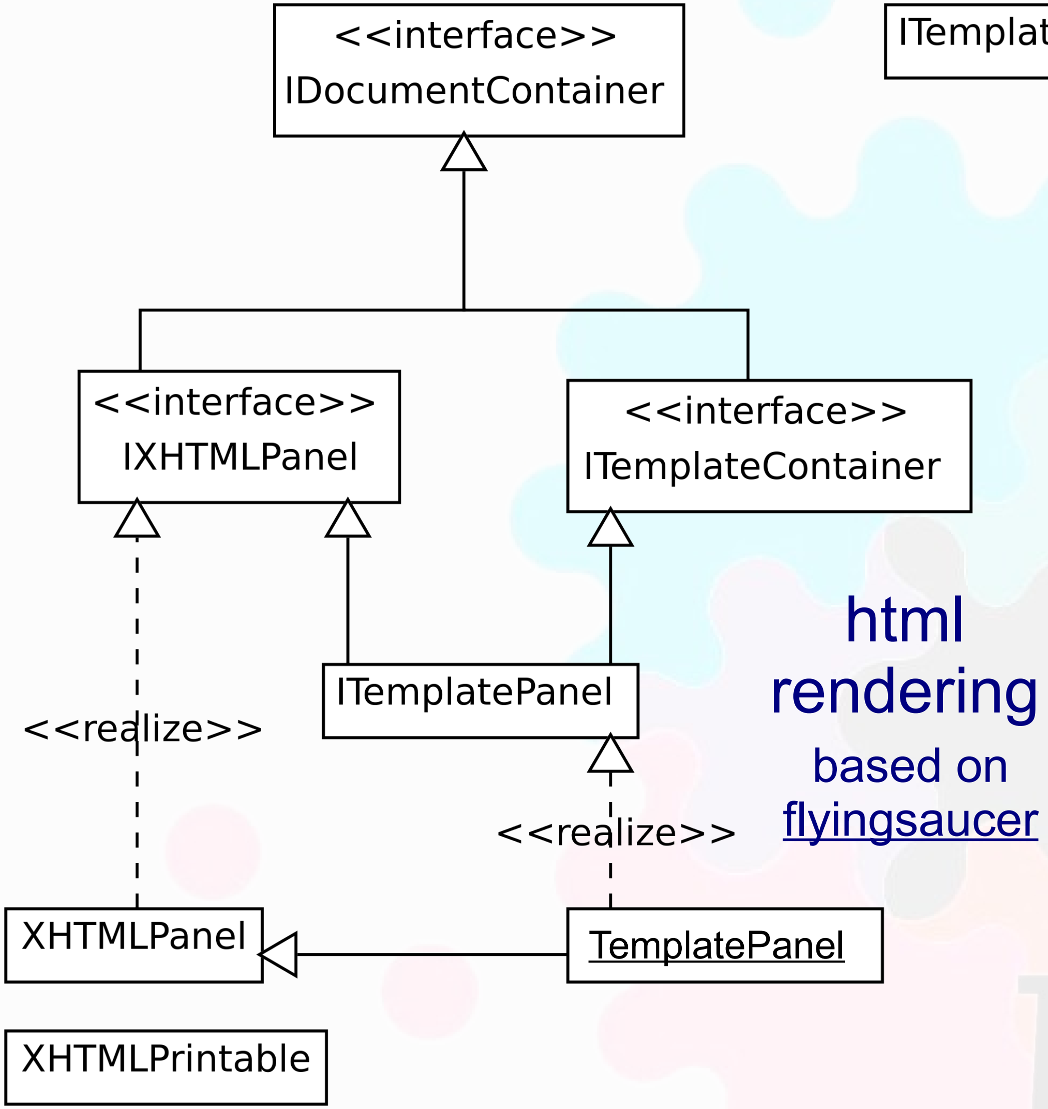
<<create>> PasswordCache()
getInstance() : PasswordCache
setParent(parent) : void
setTimeout(s : int) : void
getTimeout() : int
setUI(ui : int) : void
getUI() : int
invalidate(loc) : void
getForDecrypt(msg,loc) : char[]
getForEncrypt(msg,loc) : char[]
set(loc,pw : char[]) : void
clear() : void
touch(loc) : void
setAlwaysAskForEncrypt(alwaysAsk : boolean) : boolean
getEncryptPasswordFinder(msg,loc) : CachePasswordFinder
getDecryptPasswordFinder(msg,loc) : CachePasswordFinder
isPasswordWrongException(e) : boolean
isPasswordWrongException(e) : boolean
isPasswordNotSuppliedException(e) : boolean
isPasswordNotSuppliedException(e) : boolean
isPasswordCancelledException(e) : boolean
isPasswordCancelledException(e) : boolean
optionPaneSetFocus(c) : void

PEMReader

PEMWriter

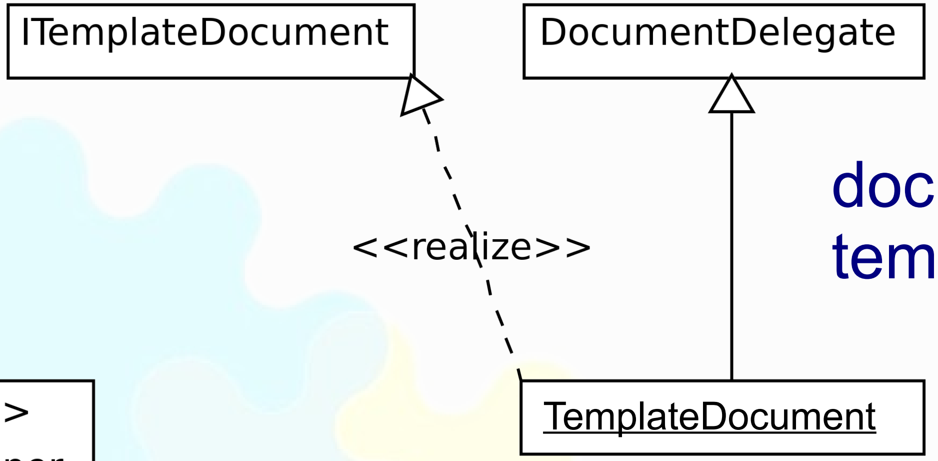
PasswordCancelledException

module: xhtmlrenderer



html rendering

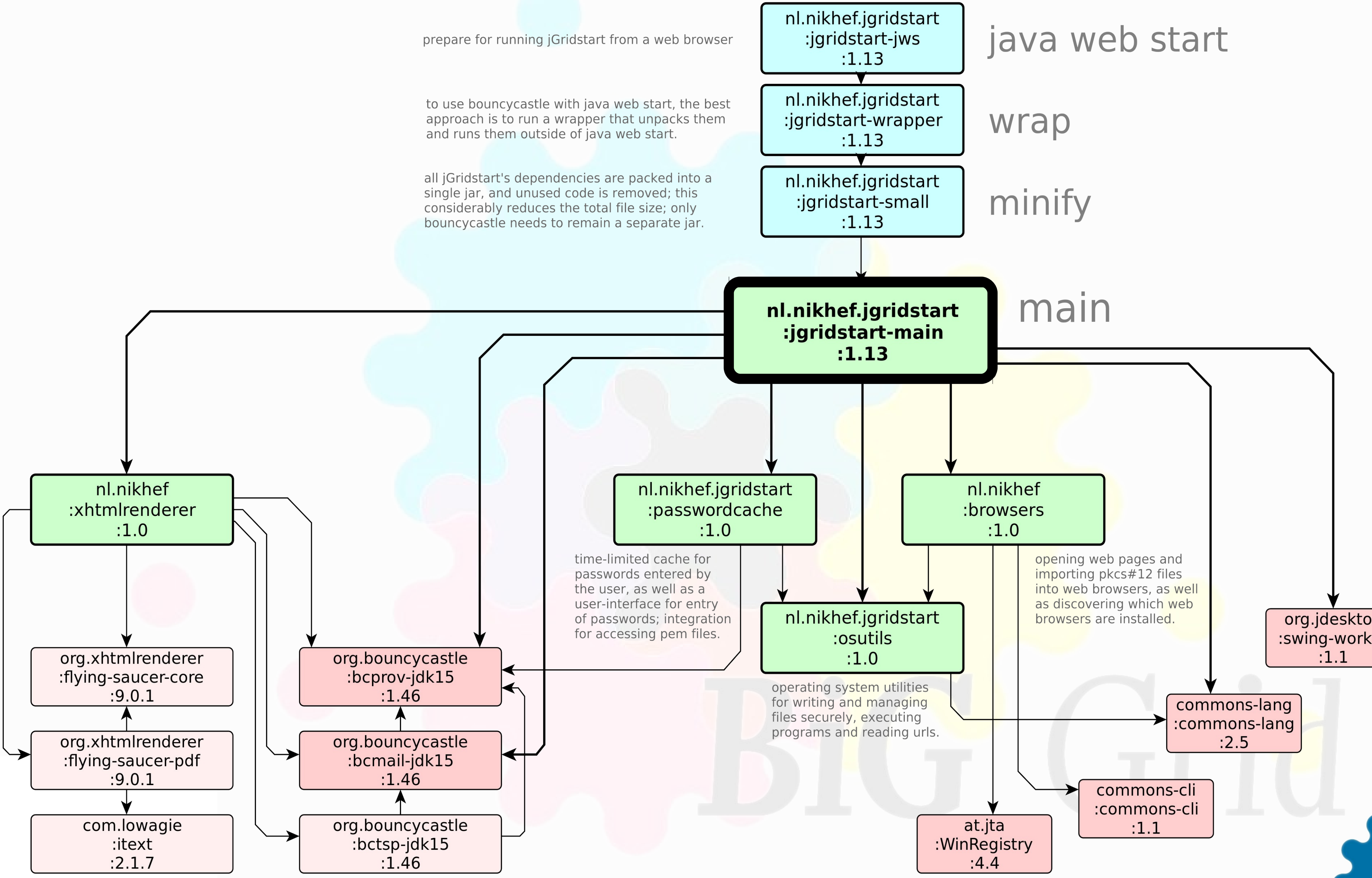
based on flyingsaucer



document templates

```
<html>
  <head><title>Hi there</title></head>
<body>
  <h1 c="Cool ${thing}">Cool foo</h1>
  <p if="!${renewal}">
    This is a new <span c="${thing}"/>
  </p>
  <form><input name="thing"/></form>
</body>
</html>
```

modules & dependencies



prepare for running jGridstart from a web browser

nl.nikhef.jgridstart
:jgridstart-jws
:1.13

java web start

to use bouncycastle with java web start, the best approach is to run a wrapper that unpacks them and runs them outside of java web start.

nl.nikhef.jgridstart
:jgridstart-wrapper
:1.13

wrap

all jGridstart's dependencies are packed into a single jar, and unused code is removed; this considerably reduces the total file size; only bouncycastle needs to remain a separate jar.

nl.nikhef.jgridstart
:jgridstart-small
:1.13

minify

nl.nikhef.jgridstart
:jgridstart-main
:1.13

main

nl.nikhef
:xhtmlrenderer
:1.0

nl.nikhef.jgridstart
:passwordcache
:1.0

time-limited cache for passwords entered by the user, as well as a user-interface for entry of passwords; integration for accessing pem files.

nl.nikhef
:browsers
:1.0

opening web pages and importing pkcs#12 files into web browsers, as well as discovering which web browsers are installed.

nl.nikhef.jgridstart
:osutils
:1.0

operating system utilities for writing and managing files securely, executing programs and reading urls.

org.jdesktop
:swing-worker
:1.1

org.xhtmlrenderer
:flying-saucer-core
:9.0.1

org.xhtmlrenderer
:flying-saucer-pdf
:9.0.1

com.lowagie
:itext
:2.1.7

org.bouncycastle
:bcprov-jdk15
:1.46

org.bouncycastle
:bcmail-jdk15
:1.46

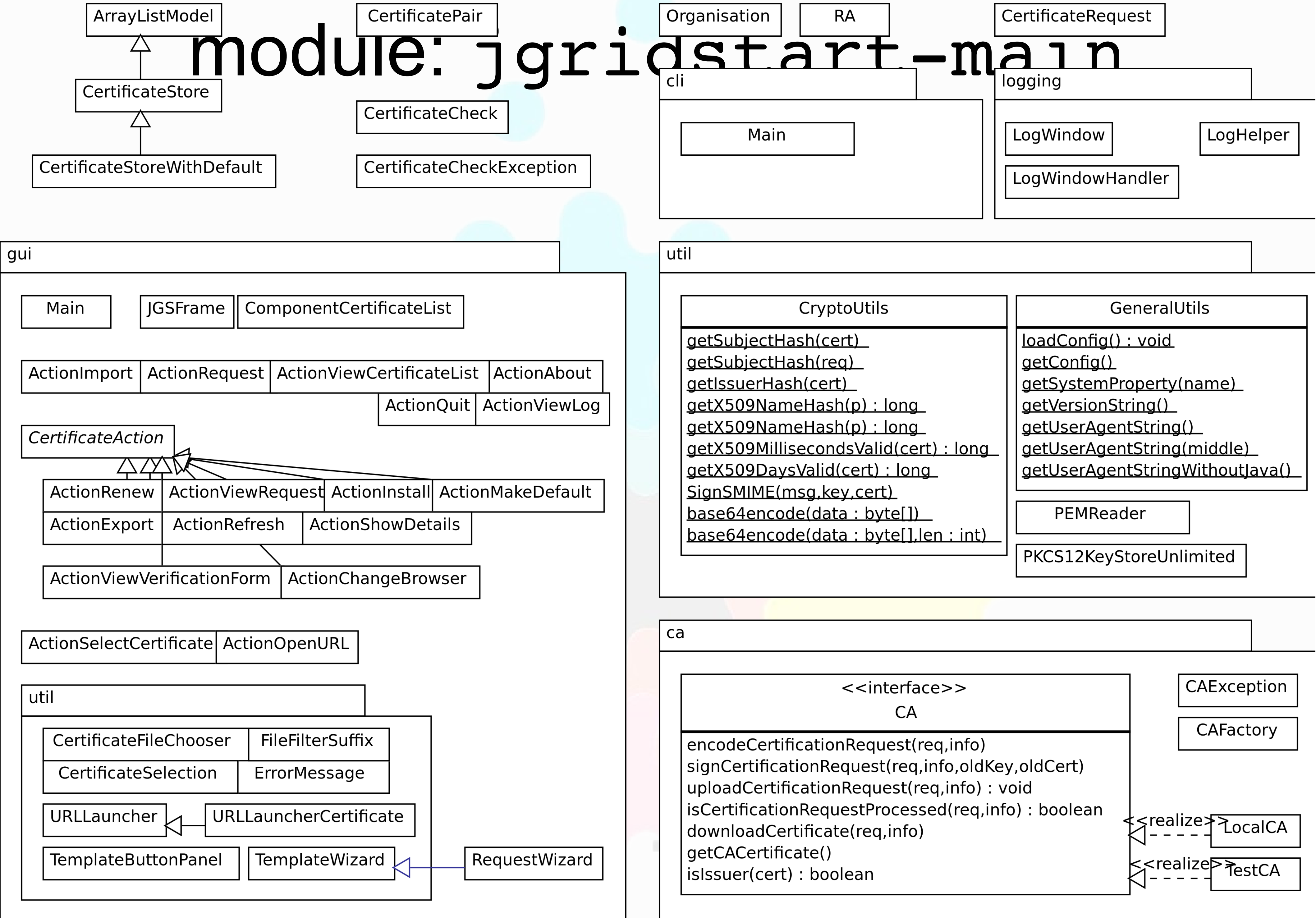
org.bouncycastle
:bctsp-jdk15
:1.46

commons-lang
:commons-lang
:2.5

commons-cli
:commons-cli
:1.1

at.jta
:WinRegistry
:4.4

module: jgridstart-main



ArrayListModel

CertificateStore

CertificateStoreWithDefault

CertificatePair

CertificateCheck

CertificateCheckException

Organisation

RA

CertificateRequest

cli

Main

logging

LogWindow

LogHelper

LogWindowHandler

gui

Main

JGSFrame

ComponentCertificateList

ActionImport

ActionRequest

ActionViewCertificateList

ActionAbout

ActionQuit

ActionViewLog

CertificateAction

ActionRenew

ActionViewRequest

ActionInstall

ActionMakeDefault

ActionExport

ActionRefresh

ActionShowDetails

ActionViewVerificationForm

ActionChangeBrowser

ActionSelectCertificate

ActionOpenURL

util

CertificateFileChooser

FileFilterSuffix

CertificateSelection

ErrorMessage

URLLauncher

URLLauncherCertificate

TemplateButtonPanel

TemplateWizard

RequestWizard

util

CryptoUtils

getSubjectHash(cert)
getSubjectHash(req)
getIssuerHash(cert)
getX509NameHash(p) : long
getX509NameHash(p) : long
getX509MillisecondsValid(cert) : long
getX509DaysValid(cert) : long
SignSMIME(msg,key,cert)
base64encode(data : byte[])
base64encode(data : byte[],len : int)

GeneralUtils

loadConfig() : void
getConfig()
getSystemProperty(name)
getVersionString()
getUserAgentString()
getUserAgentString(middle)
getUserAgentStringWithoutJava()

PEMReader

PKCS12KeyStoreUnlimited

ca

<<interface>>
CA

encodeCertificationRequest(req,info)
signCertificationRequest(req,info,oldKey,oldCert)
uploadCertificationRequest(req,info) : void
isCertificationRequestProcessed(req,info) : boolean
downloadCertificate(req,info)
getCACertificate()
isIssuer(cert) : boolean

CAException

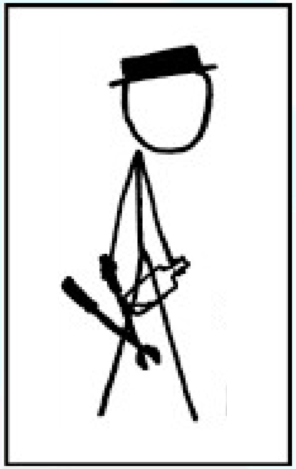
CAFactory

<<realize>>

LocalCA

<<realize>>

TestCA



http://xkcd.com/801/

important tools & libraries

tool Maven

tool Maven: artifacts

com.example.app : my-app : 1.0
group : artifact : version

OR

```
<groupId>com.example.app</groupId>
<artifactId>my-app</artifactId>
<version>1.0</version>
```

tool Maven: default lifecycle

```
process-resources
compile
process-test-resources
test-compile
test
package
install
deploy
```

tool Maven: simple pom.xml

```xml
<project>

  <modelVersion>4.0.0</modelVersion>

  <groupId>com.example</groupId>
  <artifactId>my-app</artifactId>
  <version>1.0</version>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>

</project>
```

tool Maven: standard layout

pom.xml

src/main/java

src/main/resources

src/test/java

src/test/resources

target/classes

target/**my-app-1.0.jar**

library BouncyCastle and JCA

- JCE = Java Cryptography Extensions
  encryption, key generation, key agreement, MAC
- JCA = Java Cryptography Architecture
  framework for developing and accessing cryptographic functionality in Java
- BouncyCastle = external Java Crypto Provider

library BouncyCastle and JCA

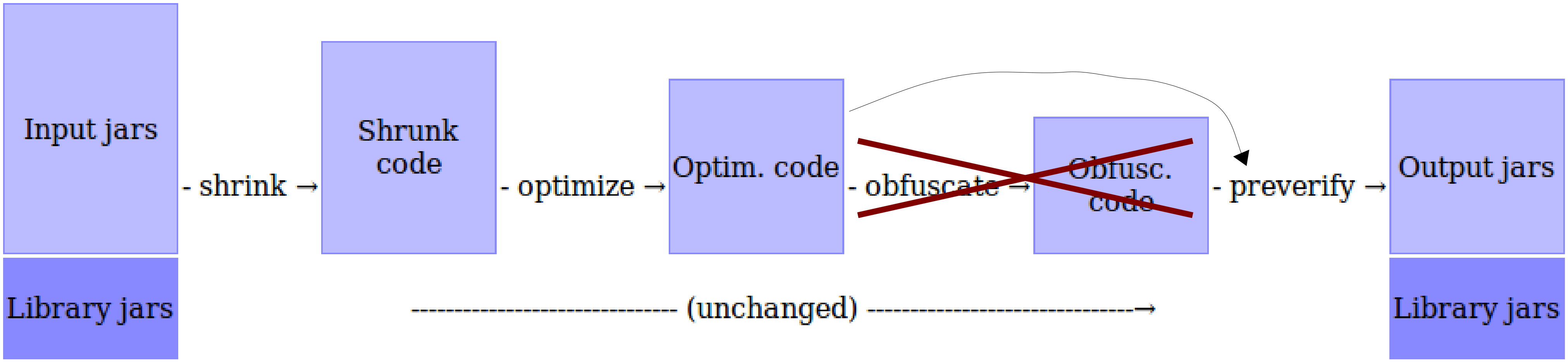
```
KeyGenerator kg = KeyGenerator.getInstance("DES");
Key key = kg.generateKey();

Cipher cipher = Cipher.getInstance("DES");
byte[] data = "Hello World!".getBytes();

cipher.init(Cipher.ENCRYPT_MODE, key);
byte[] result = cipher.doFinal(data);
```

```
cipher.init(Cipher.DECRYPT_MODE, key);
byte[] original = cipher.doFinal(result);
```

tool ProGuard



Input jars

- shrink →

Shrunk code

- optimize →

Optim. code

~~- obfuscate →~~

~~Obfusc. code~~

- preverify →

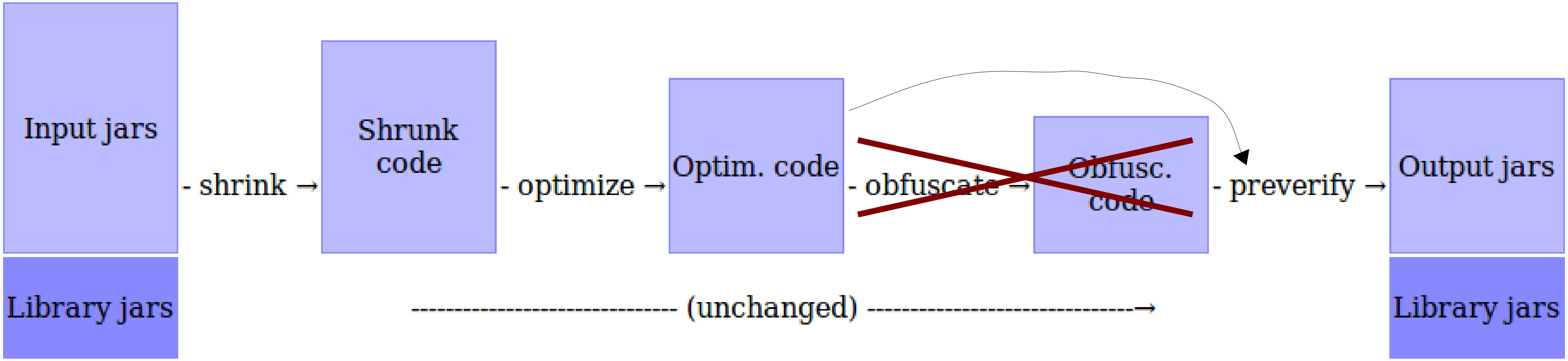
Output jars

Library jars

------------------------------ (unchanged) ------------------------------→

Library jars

tool ProGuard



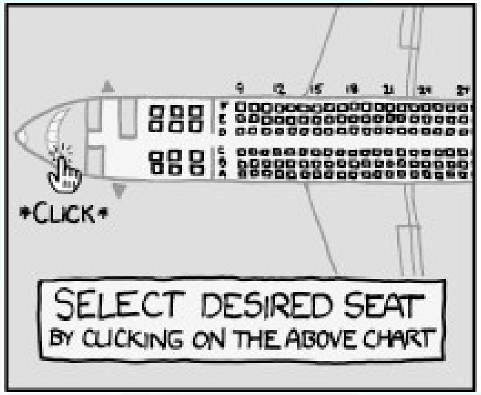
jGridstart main	207	kB
main + most deps	3.4	MB
small	1.7	MB
BouncyCastle dep.	1.8	MB
Wrapper (total)	**3.0**	**MB**

incl. jGridstart, dependencies, BouncyCastl

tool Java Web Start

tool Java Web Start

```xml
<?xml version="1.0" encoding="utf-8"?>
<jnlp spec="1.0+" href="jgridstart.jnlp" codebase="http://ca.dutchgrid.nl/start/">
 <information>
  <title>jGridstart 1.13</title>
  <vendor>NIKHEF / Stichting FOM</vendor>
  <homepage href="http://www.nikhef.nl/"/>
  <description>Setup your computer to work with the grid</description>
 </information>
 <security>
  <all-permissions/>
 </security>
 <resources>
  <!-- program defaults, customize these for your site (properties are prefixed
       with "jnlp." to avoid the need for a signed jnlp) -->
  <property name="jnlp.jgridstart.defaults.country" value="NL"/>
  <property name="jnlp.jgridstart.keysize" value="2048"/>
  <property name="jnlp.jgridstart.org.href" value="cert_signup.conf"/>
  <property name="jnlp.jgridstart.ca.provider" value="DutchGridCA"/>
  <!-- end of program defaults -->
  <j2se href="http://java.sun.com/products/autodl/j2se" version="1.5+"/>
  <jar href="jgridstart-wrapper-1.13.jar" main="true"/>
 </resources>
 <application-desc main-class="nl.nikhef.jgridstart.wrapper.Wrapper"/>
</jnlp>
```



http://xkcd.com/726/

hacks

only try this at work

hack PKCS12KeystoreUnlimited

```
FileInputStream in = new FileInputStream("test.p12");
KeyStore store = KeyStore.getInstance("PKCS12", "BC");
store.load(in, "longpassworduhoh".toCharArray());
Certificate cert = store.getCertificate("cert alias");
```

hack PKCS12KeystoreUnlimited

```
FileInputStream in = new FileInputStream("test.p12");
KeyStore store = KeyStore.getInstance("PKCS12", "BC");
store.load(in, "longpasswordhuhoh".toCharArray());
Certificate cert = store.getCertificate("cert alias");
```

BOOM

Exception

hack PKCS12KeystoreUnlimited

```
FileInputStream in = new FileInputStream("test.p12");
KeyStore store = PKCS12KeyStoreUnlimited.getInstance();
store.load(in, "longpasswordhuhoh".toCharArray());
Certificate cert = store.getCertificate("cert alias");
```

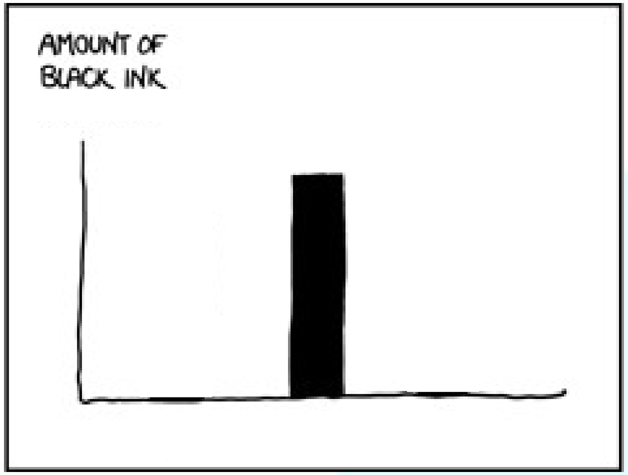
:-)

hack PKCS12KeystoreUnlimited

```
class PKCS12KeyStoreUnlimited extends JDKPKCS12KeyStore {
 static KeyStore getInstance() {
    // retrieve private BC provider
    KeyStore store = KeyStore.getInstance("PKCS12", "BC");
    Field keyStoreProvider =
        store.getClass().getDeclaredField("provider");
    keyStoreProvider.setAccessible(true);
    Provider provider = keyStoreProvider.get(store);
    // and keystore implementation
    Field keyStoreSpi =
        store.getClass().getDeclaredField("keyStoreSpi");
    keyStoreSpi.setAccessible(true);
    JDKPKCS12KeyStore bcStore =
        (JDKPKCS12KeyStore)keyStoreSpi.get(store);
    // override that by using our wrapper class
    keyStoreSpi.set(store,
        new PKCS12KeyStoreUnlimited(provider, bcStore));
  }
```

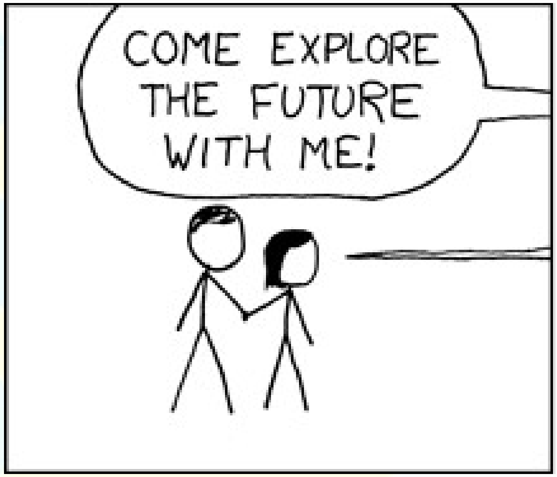
hack safe file copy

Firefox PKCS#12 installation



http://xkcd.com/688/

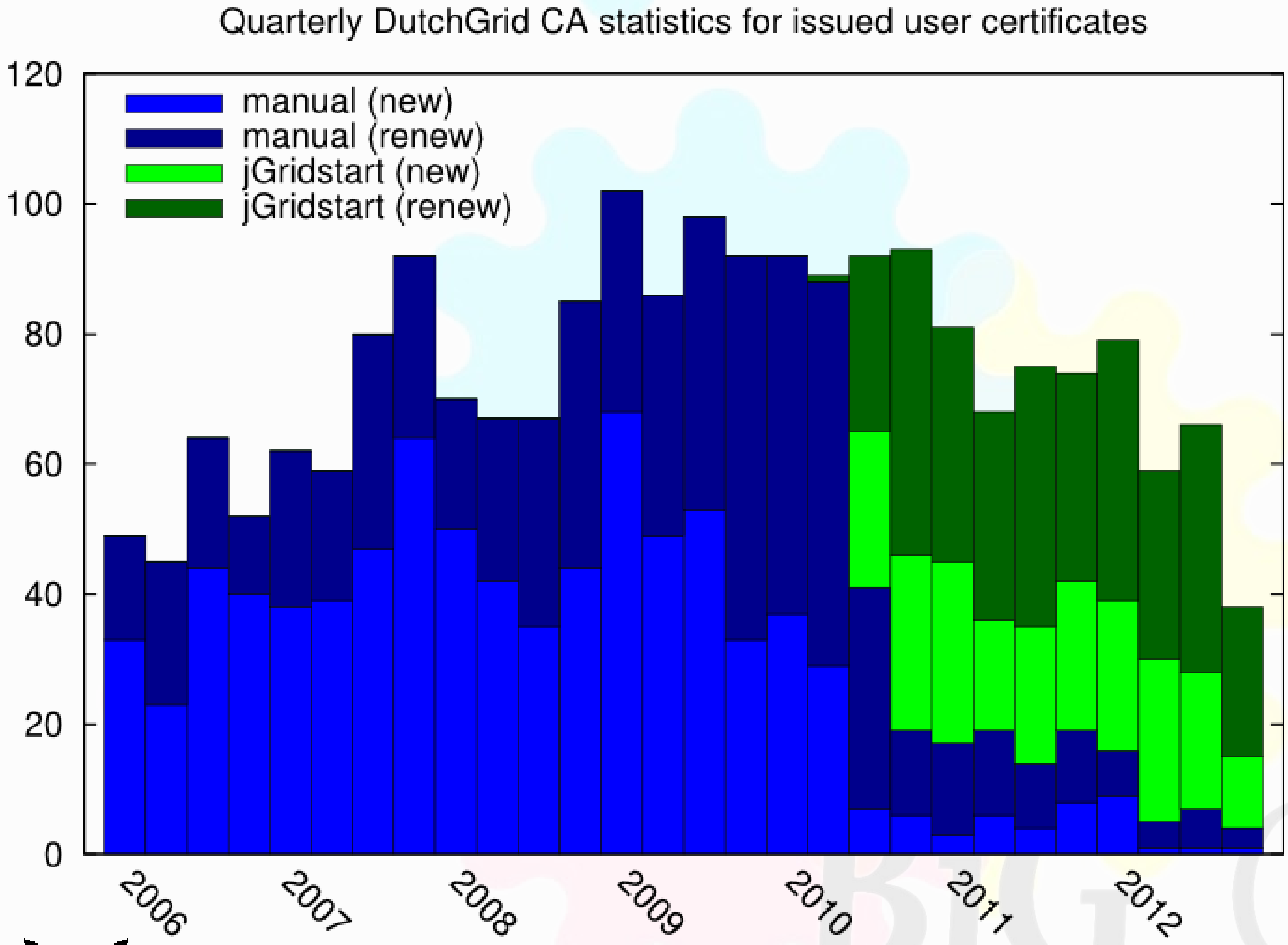
what's the value?



http://xkcd.com/338/

what next?

usage



as of Nov 2012
jGridstart grand totals

696 requests
595 granted

224 Windows
213 Macintosh
259 Linux

34 Java 1.5
626 Java 1.6
36 Java 1.7

contacts

- UK / Jens Jensen
  - Certificate Wizard (certwizard)
  - jGridstart phoneconf presentation
  - using PKCS12KeyStoreUnlimited
  - ideas for common CA interface API incl. Grix?
- Grix (AU) existed already, not friendly enough
- Some interest at EGEE UF 2012
  - poster presentation (session)
  - mostly by CAs
  - I think it's still too complicated for CAs to just go

open ends

- default file permissions on Windows?
- safe file copy works *almost* always on Windows
- new Firefox certificate installation method
  e.g. by triggering a (self-deinstalling) extension
- add *archived* certificate concept
  to handle expired certificates more intuitively
- finish Confusa / TCS connection
  get them to finish the Confusa part first
- not all dependencies in Maven Central
  WinRegistry and Abbot 1.2.0 (#4044)
- more&easier CA adaptations much done in multica branch

+ plus other bugs at
http://jgridstart.nikhef.nl/bugs

- Browser Java detection UI

pointers

- Website, and for Developers lots of extra info
  http://jgridstart.nikhef.nl/
- Code at Github; continuous integration at Travis-CI
  https://github.com/biggrid/jGridstart
- Javadoc
  http://jgridstart.nikhef.nl/javadoc/ (hosted at Github)
- Bugs at SARA's Mantis installation
  http://jgridstart.nikhef.nl/bugs
- Releases with test-drive
  http://jgridstart.nikhef.nl/Releases
  - corresponding Test CA