Release date 04-01-2015

Software Engineering 2: MyTaxiService

Inspection Document

V1.0

Dimitar Anastasovski, Marco Colombo

Contents

Chapter 1

Assigned classes

We only have one class called **PolicyParser.java**

Methods:

- **Name:**getStorePassURL( )
  Start Line:293
  Location:appserver/security/core-ee/src/main/java/com/sun/enterprise/security/provider/PolicyParser.java

- **Name:**parseKeyStoreEntry( )
  Start Line:358
  Location:appserver/security/core-ee/src/main/java/com/sun/enterprise/security/provider/PolicyParser.java

- **Name:**writeKeyStoreEntry(PrintWriter out)
  Start Line:397
  Location:appserver/security/core-ee/src/main/java/com/sun/enterprise/security/provider/PolicyParser.java

- **Name:**parseGrantEntry( )
  Start Line:420
  Location:appserver/security/core-ee/src/main/java/com/sun/enterprise/security/provider/PolicyParser.java

Chapter 2

Functional role

The policy for a Java runtime (specifying which permissions are available for code from various principals) is represented as a separate persistent configuration. The configuration may be stored as a flat ASCII file, as a serialized binary file of the Policy class, or as a database.
The Java runtime creates one global Policy object, which is used to represent the static policy configuration file. It is consulted by a ProtectionDomain when the protection domain initializes its set of permissions.
The Policy init method parses the policy configuration file, and then populates the Policy object. The Policy object is agnostic in that it is not involved in making policy decisions. It is merely the Java runtime representation of the persistent policy configuration file.
When a protection domain needs to initialize its set of permissions, it executes code such as the following to ask the global Policy object to populate a Permissions object with the appropriate permissions:

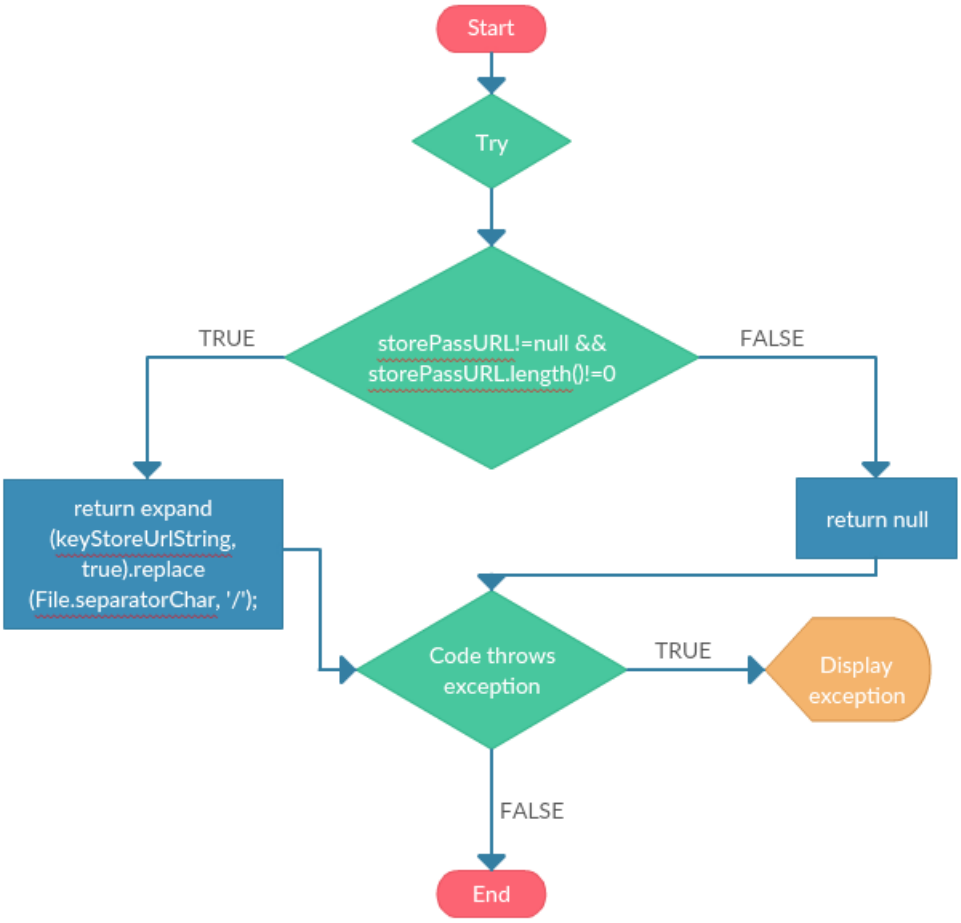
```
policy = Policy.getPolicy();

Permissions perms = policy.getPermissions(protectiondomain)
```

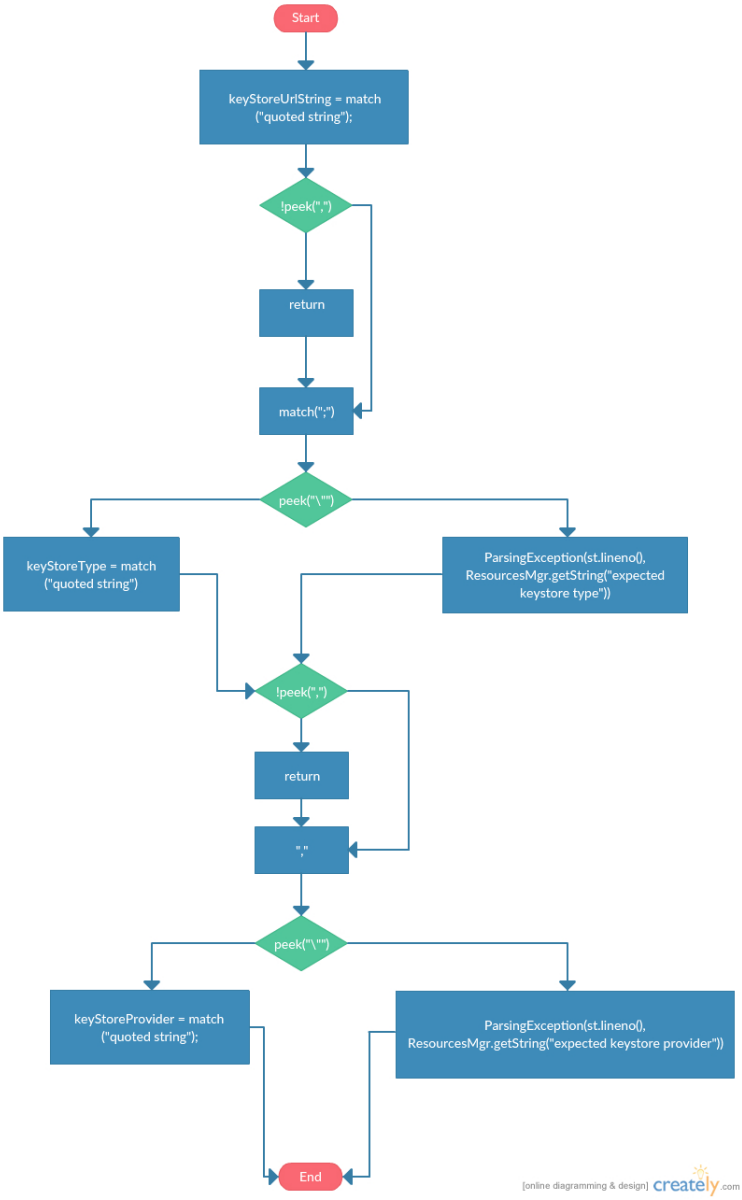
The protection domain contains CodeSource object, which encapsulates its codebase (URL) and public key attributes. It also contains the principals associated with the domain. The Policy object evaluates the global policy in light of who the principal is and what the code source is and returns an appropriate Permissions object.

Methods:

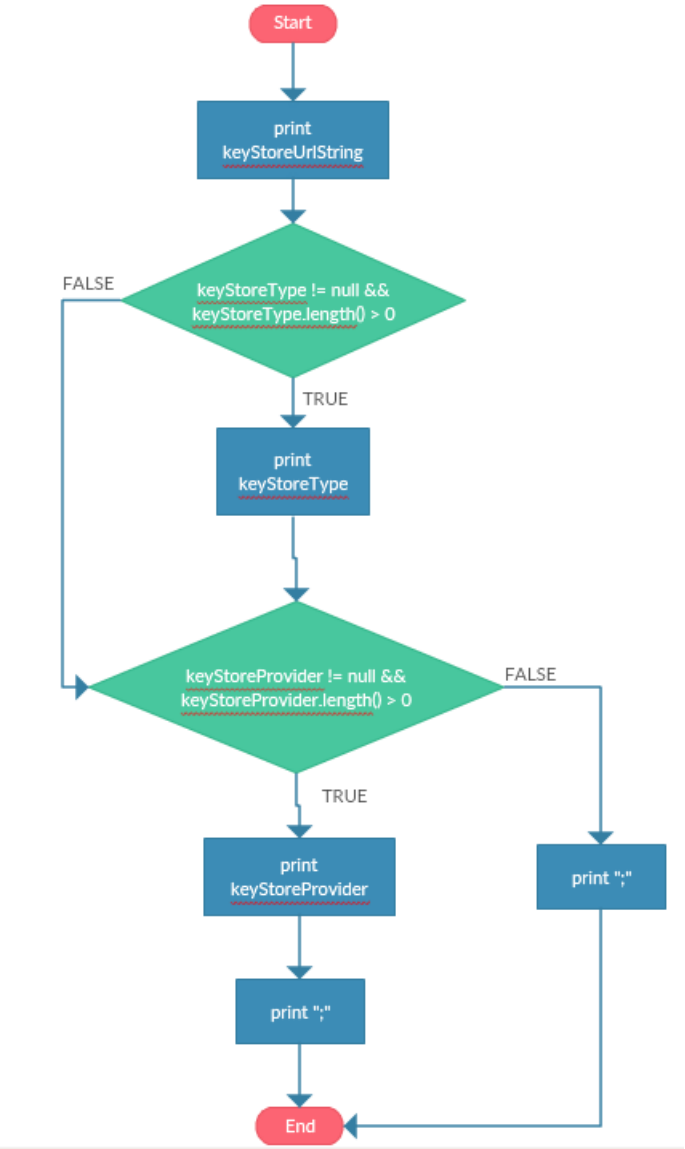
- getStorePassURL( )



- parseKeyStoreEntry( )



- writeKeyStoreEntry(PrintWriter out)



Chapter 3

Issues

3.1 Naming Conventions

No issues here

3.2 Indention

1. Line 432 more than 9 spaces applied
   Line 444 more than 9 spaces applied

2. No issues

3.3 Braces

Keringhan and Ritchie styke is used in whole class

3.4 Wrapping lines

Line 355 line break not correct
Line 364 line break not correct
Line 443 line break not correct
Line 463 line break not correct

3.5 File Organization

No issues here

3.6 Wrapping lines

No issues here

3.7 Comments

Comments styles are different in the whole code. For example:

// parse keystore type

or

/**
* parses a keystore entry
*/

3.8 Java Source File

No issues here

3.9 Package and import statements

No issues here

3.10 Class and interface declaration

No issues here

3.11 Initialization and Declarations

No issues here

3.12 Method Calls

No issues here

3.13 Arrays

No issues here

3.14 Object Comparison

No issues here

3.15 Output Format

Line 410 output formal is not clear and formal

3.16 Computation, Comparisons and Assignments

Line 401,549,575 brutish programing is present

3.17 Exceptions

Line 542 exception is thrown but not catch properly in the function.

3.18 Flow of Control

No issues here

3.19 Files

No issues here

Chapter 4

References

4.1 Software and used Tools

- TexShop (http://pages.uoregon.edu/koch/texshop/), to redact this document

4.2 Working hours

Dimitar Anastasovski: $\sim$ hours

Marco Colombo: $\sim$ hours