JDK 9 New Features

杨晓峰(felix.yang@oracle.com)

甲骨文首席工程师，Java核心类库组

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.
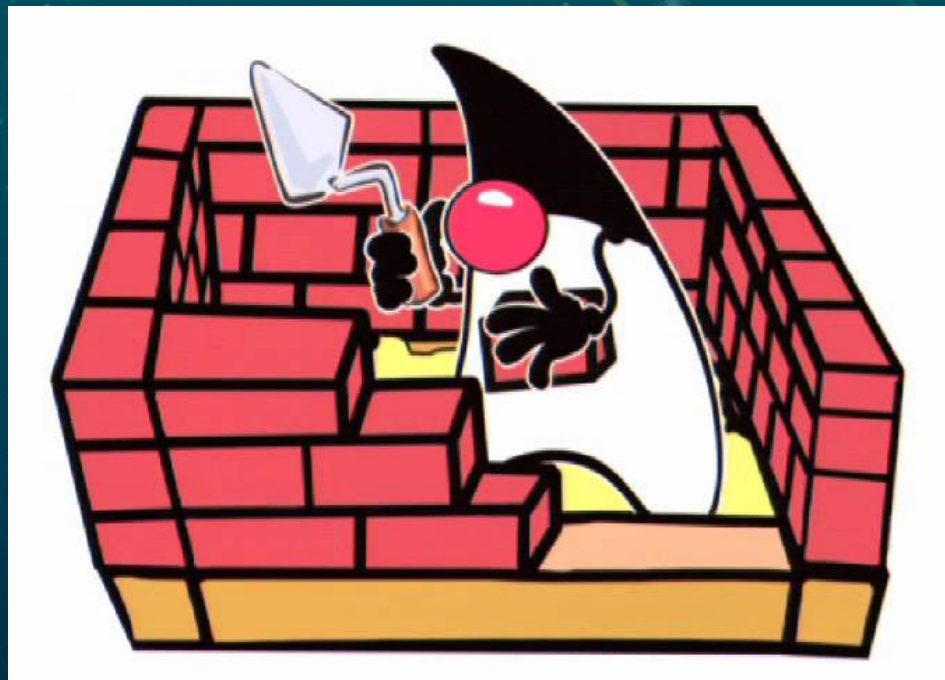
Agenda

- Java Platform Module System

- Java Virtual Machine

- Libraries and tools

Project Jigsaw

OpenJDK - http://openjdk.java.net/projects/jigsaw/

Why we need modules?

-classpath

classpath can be...

common/hadoop-common-3.0.0-SNAPSHOT.jar:common/hadoop-nfs-3.0.0-SNAPSHOT.jar:common/lib/activation-1.1.jar:common/lib/apacheds-i\
18n-2.0.0-M15.jar:common/lib/apacheds-kerberos-codec-2.0.0-M15.jar:common/lib/api-asn1-api-1.0.0-M20.jar:common/lib/api-util-1.0\
.0-M20.jar:common/lib/asm-3.2.jar:common/lib/avro-1.7.4.jar:common/lib/commons-beanutils-1.7.0.jar:common/lib/commons-beanutils-\
core-1.8.0.jar:common/lib/commons-cli-1.2.jar:common/lib/commons-codec-1.4.jar:common/lib/commons-collections-3.2.1.jar:common/l\
ib/commons-compress-1.4.1.jar:common/lib/commons-configuration-1.6.jar:common/lib/commons-digester-1.8.jar:common/lib/commons-ht\
tpclient-3.1.jar:common/lib/commons-io-2.4.jar:common/lib/commons-lang-2.6.jar:common/lib/commons-logging-1.1.3.jar:common/lib/c\
ommons-math3-3.1.1.jar:common/lib/commons-net-3.1.jar:common/lib/curator-client-2.7.1.jar:common/lib/curator-framework-2.7.1.jar\
:common/lib/curator-recipes-2.7.1.jar:common/lib/gson-2.2.4.jar:common/lib/guava-11.0.2.jar:common/lib/hadoop-annotations-3.0.0-\
SNAPSHOT.jar:common/lib/hadoop-auth-3.0.0-SNAPSHOT.jar:common/lib/hamcrest-core-1.3.jar:common/lib/htrace-core4-4.0.1-incubating\
.jar:common/lib/httpclient-4.2.5.jar:common/lib/httpcore-4.2.5.jar:common/lib/jackson-core-asl-1.9.13.jar:common/lib/jackson-jax\
rs-1.9.13.jar:common/lib/jackson-mapper-asl-1.9.13.jar:common/lib/jackson-xc-1.9.13.jar:common/lib/java-xmlbuilder-0.4.jar:commo\
n/lib/jaxb-api-2.2.2.jar:common/lib/jaxb-impl-2.2.3-1.jar:common/lib/jcip-annotations-1.0.jar:common/lib/jersey-core-1.9.jar:com\
mon/lib/jersey-json-1.9.jar:common/lib/jersey-server-1.9.jar:common/lib/jets3t-0.9.0.jar:common/lib/jettison-1.1.jar:common/lib/\
jetty-6.1.26.jar:common/lib/jetty-util-6.1.26.jar:common/lib/jsch-0.1.51.jar:common/lib/json-smart-1.1.1.jar:common/lib/jsp-api-\
2.1.jar:common/lib/jsr305-3.0.0.jar:common/lib/junit-4.11.jar:common/lib/log4j-1.2.17.jar:common/lib/mockito-all-1.8.5.jar:commo\
n/lib/netty-3.6.2.Final.jar:common/lib/nimbus-jose-jwt-3.9.jar:common/lib/paranamer-2.3.jar:common/lib/protobuf-java-2.5.0.jar:c\
ommon/lib/servlet-api-2.5.jar:common/lib/slf4j-api-1.7.10.jar:common/lib/slf4j-log4j12-1.7.10.jar:common/lib/snappy-java-1.0.4.1\
.jar:common/lib/stax-api-1.0-2.jar:common/lib/xmlenc-0.52.jar:common/lib/xz-1.0.jar:common/lib/zookeeper-3.4.6.jar:hdfs/hadoop-h\
dfs-3.0.0-SNAPSHOT.jar:hdfs/hadoop-hdfs-nfs-3.0.0-SNAPSHOT.jar:hdfs/lib/commons-daemon-1.0.13.jar:hdfs/lib/hadoop-hdfs-client-3.\
0.0-SNAPSHOT.jar:hdfs/lib/hpack-0.11.0.jar:hdfs/lib/leveldbjni-all-1.8.jar:hdfs/lib/netty-all-4.1.0.Beta5.jar:hdfs/lib/okhttp-2.\
4.0.jar:hdfs/lib/okio-1.4.0.jar:hdfs/lib/xercesImpl-2.9.1.jar:mapreduce/hadoop-mapreduce-client-app-3.0.0-SNAPSHOT.jar:mapreduce\
/hadoop-mapreduce-client-common-3.0.0-SNAPSHOT.jar:mapreduce/hadoop-mapreduce-client-core-3.0.0-SNAPSHOT.jar:mapreduce/hadoop-ma\
preduce-client-hs-3.0.0-SNAPSHOT.jar:mapreduce/hadoop-mapreduce-client-hs-plugins-3.0.0-SNAPSHOT.jar:mapreduce/hadoop-mapreduce-\
client-jobclient-3.0.0-SNAPSHOT.jar:mapreduce/hadoop-mapreduce-client-nativetask-3.0.0-SNAPSHOT.jar:mapreduce/hadoop-mapreduce-c\
lient-shuffle-3.0.0-SNAPSHOT.jar:mapreduce/hadoop-mapreduce-examples-3.0.0-SNAPSHOT.jar:yarn/hadoop-yarn-api-3.0.0-SNAPSHOT.jar:\
yarn/hadoop-yarn-applications-distributedshell-3.0.0-SNAPSHOT.jar:yarn/hadoop-yarn-applications-unmanaged-am-launcher-3.0.0-SNAP\
SHOT.jar:yarn/hadoop-yarn-client-3.0.0-SNAPSHOT.jar:yarn/hadoop-yarn-common-3.0.0-SNAPSHOT.jar:yarn/hadoop-yarn-registry-3.0.0-S\
NAPSHOT.jar:yarn/hadoop-yarn-server-applicationhistoryservice-3.0.0-SNAPSHOT.jar:yarn/hadoop-yarn-server-common-3.0.0-SNAPSHOT.j\
ar:yarn/hadoop-yarn-server-nodemanager-3.0.0-SNAPSHOT.jar:yarn/hadoop-yarn-server-resourcemanager-3.0.0-SNAPSHOT.jar:yarn/hadoop\
-yarn-server-sharedcachemanager-3.0.0-SNAPSHOT.jar:yarn/hadoop-yarn-server-web-proxy-3.0.0-SNAPSHOT.jar:yarn/lib/aopalliance-1.0\
.jar:yarn/lib/commons-math-2.2.jar:yarn/lib/curator-test-2.7.1.jar:yarn/lib/fst-2.24.jar:yarn/lib/guice-3.0.jar:yarn/lib/guice-s\
ervlet-3.0.jar:yarn/lib/javassist-3.18.1-GA.jar:yarn/lib/javax.inject-1.jar:yarn/lib/jersey-client-1.9.jar:yarn/lib/jersey-guice\
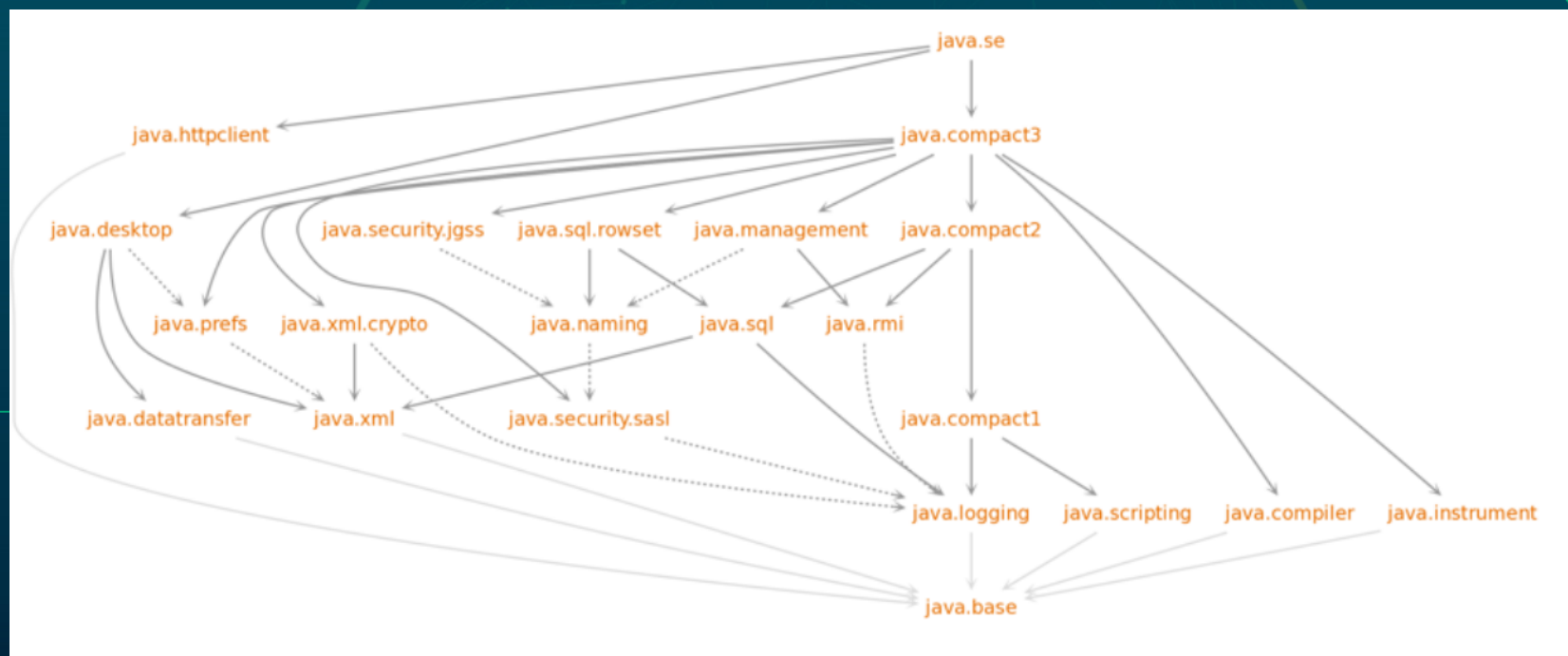-1.9.jar:yarn/lib/objenesis-2.1.jar

Java Module System

- JSR 376: Java Platform Module System

- Support modules deep into Java:
  - compilation, linking and runtime phases.

- Javac/java:
  - Support to work with modules, such as module path
  - A module can be jar, jmod or exploded files

- Add jlink tool and introduced new **optional** phase of linking

Modularized JDK libraries

Have a look at Java module

- Module sources:
  - Module descriptor (module-info.java)
  - Source codes
- Can be jar, jmod or exploded files
- A typical module descriptor:


```
module java.naming {
    requires java.security.sasl;
    exports javax.naming;
// …
    exports javax.naming.spi;
    exports com.sun.jndi.toolkit.ctx to jdk.naming.dns;
    exports com.sun.jndi.toolkit.url to java.corba, jdk.naming.dns, jdk.naming.rmi;
    uses javax.naming.ldap.StartTlsResponse;
// …
    provides java.security.Provider with sun.security.provider.certpath.ldap.JdkLDAP;
}
```
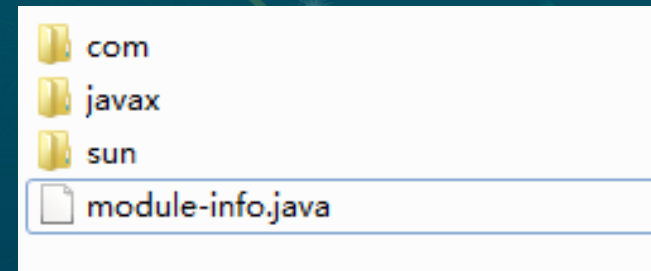
Kinds of Java modules

- Explicit named modules
- Automatic named modules
  - put jars to module path
- Unnamed modules
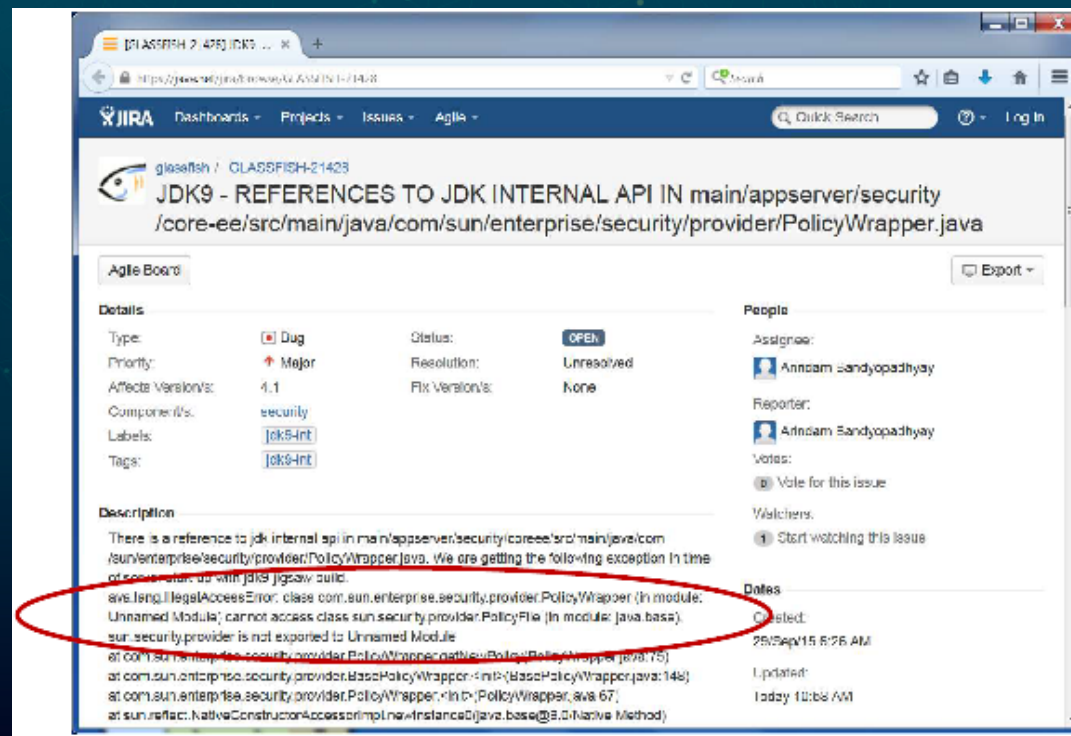 - those loaded by class path

10

Java Modules behaves

- Accessibility strongly encapsulates module internals
- Accessibility relies readability
- Fidelity across compiling, linking or runtime

11

Accessibility (JDK 9)

- *public to everyone*

- *public but only to specific modules*

- *public only within a module*

- protected

- <package>

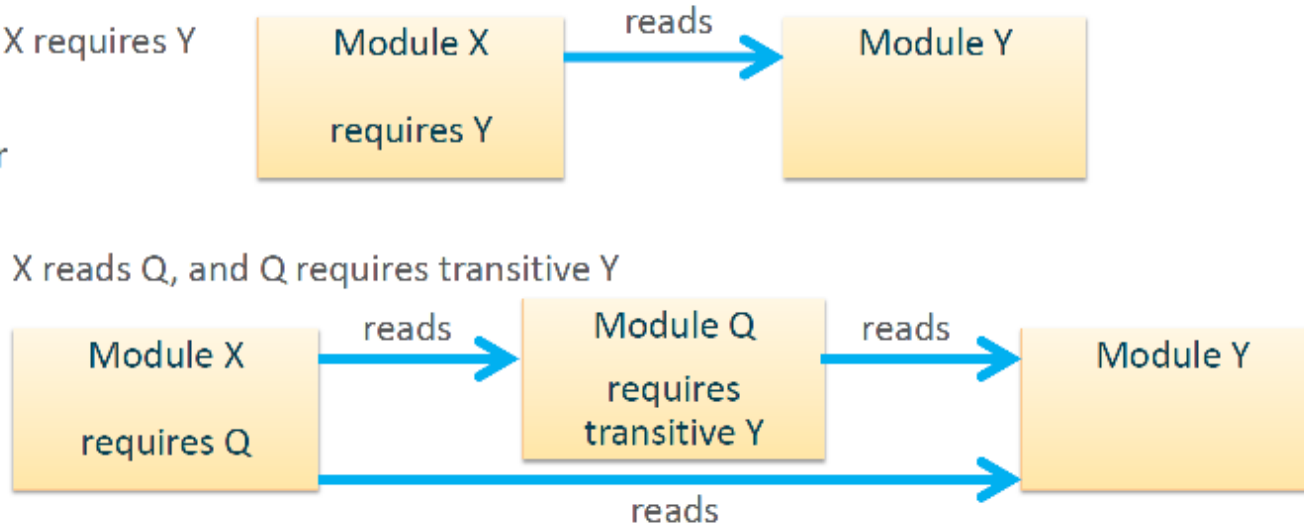- private

Readability – direct or implied

X reads Y if:

— X requires Y

```
┌──────────────┐      reads      ┌──────────────┐
│  Module X    │ ──────────────▶ │  Module Y    │
│  requires Y  │                 │              │
└──────────────┘                 └──────────────┘
```

or

— X reads Q, and Q requires transitive Y

```
┌──────────────┐  reads  ┌──────────────┐  reads  ┌──────────────┐
│  Module X    │ ──────▶ │  Module Q    │ ──────▶ │  Module Y    │
│  requires Q  │         │  requires    │         │              │
│              │         │  transitive Y│         │              │
└──────────────┘         └──────────────┘         └──────────────┘
        │                                                ▲
        └────────────────────── reads ──────────────────┘
```

Core Reflection APIs for Modules

- New API:
  - java.lang.reflect.Layer/Module
  - Extends: Java.lang.Class::getModule()
- Layers control the relationship between modules and class loaders
  - Non-hierarchical
  - Boot layer created at vm startup
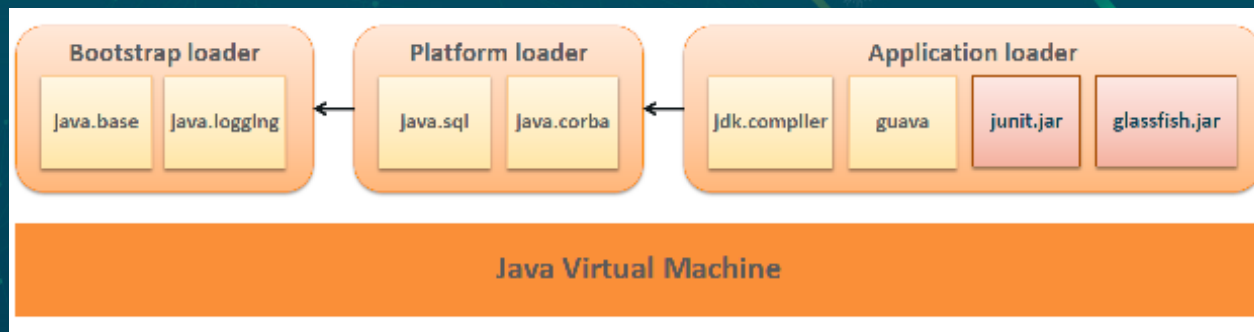 - A container application can create a new layer

Class loaders changed?

- Yes and no...
- Now:
  - Bootstrap class loader
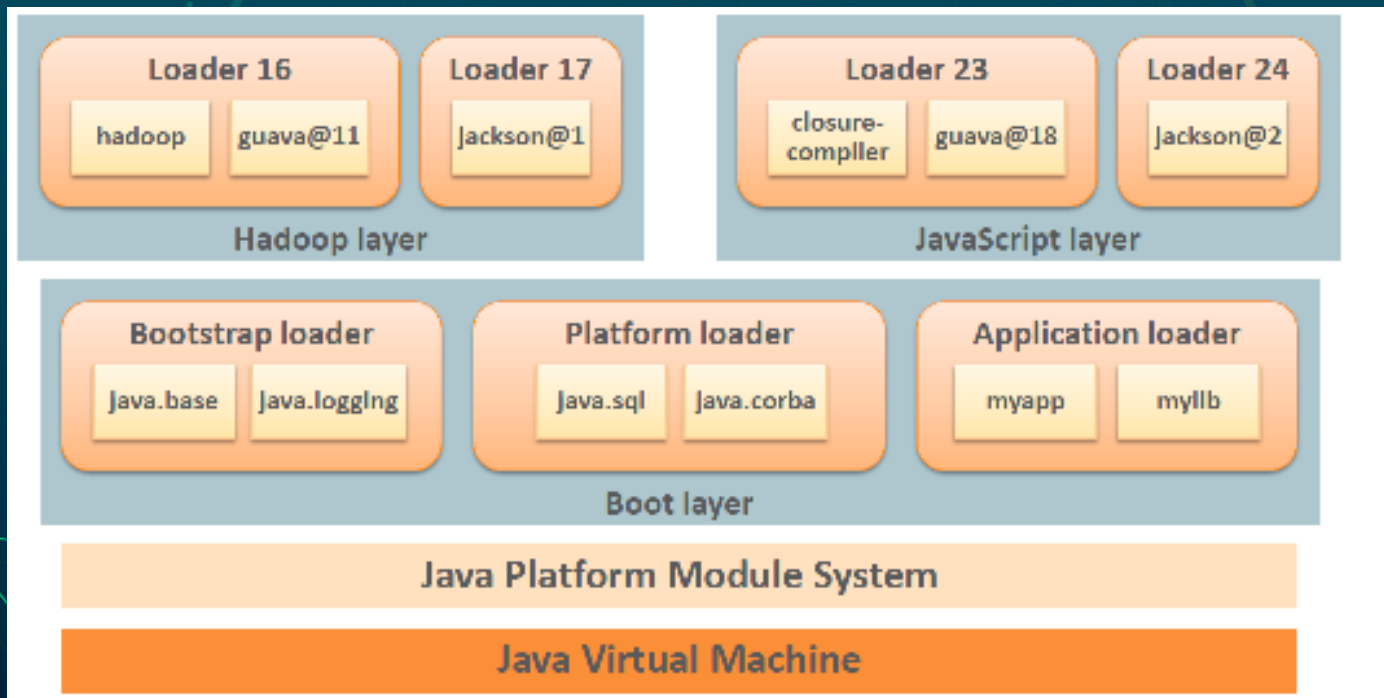 - Platform class loader
    - original extension class loader
    - Retire extension mechanism
    - Modules de-privileged for security improvement
      - Side effect: NOT all Jave SE types are visible to boot loader
  - Application class loader

Layers, class loaders and modules

Sample usage

- List modules or a specific module

  java --list-modules

  java --list-modules java.se

- Compile with module path

  javac --module-path mod_dir \
         --add-modules java.desktop \
         --module-source-path java_files

- Launch a modular application

  java --module-path mod_dir -m my_mod/MainClass

Sample usage

- Easily create a minimized Java runtime:

  $ jlink --module-path jmods/ --add-modules java.desktop --output myimage

  $ myimage/bin/java --list-modules

  java.base@9

  java.datatransfer@9

  java.desktop@9

  java.prefs@9

  java.xml@9

  $ myimage/bin/java -jar jedit.jar

Benefits from Modules

Scalability + Maintainability + Security + improved-performance

19

Agenda

- Java Platform Module System
- Java Virtual Machine
- Libraries and tools

Ahead-of-Time Compilation

- Currently experimental
- Accelerate startup by providing a library-like mechanism for saving and loading compiled code
- A new tool for compilation: jaotc
- Sample usage:
  - Compile:
    jaotc --output libHelloWorld.so HelloWorld.class
  - Then run with:
    java -XX:AOTLibrary=./libHelloWorld.so HelloWorld

Make G1 the Default GC

- Usually limiting gc pause time is more important than maximizing throughput
- G1 is a robust and well-tested collector:
  - Able to meet the latency SLAs
  - Worse case pause times are expected to be better than CMS
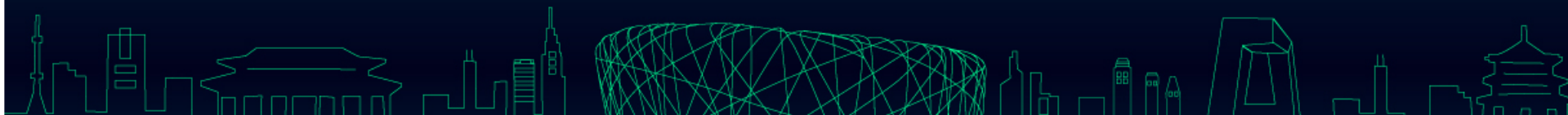
Drop deprecated iCMS GC

- Remove incremental CMS (iCMS, deprecated in 8)
- Dropped following gc combinations

Flags	GC Configuration
-XX:-UseParNewGC -XX:+UseConcMarkSweepGC	DefNew + CMS
-XX:+UseParNewGC	ParNew + SerialOld
-Xincgc	ParNew + iCMS
-XX:+CMSIncrementalMode -XX:+UseConcMarkSweepGC	ParNew + iCMS
-XX:+CMSIncrementalMode -XX:+UseConcMarkSweepGC -XX:-UseParNewGC	DefNew + iCMS

Unified logging for JVM and GC

- Provide fine-grained, easy to configure JVM logging

- Make -Xlog:gc to be similar with "-XX:PrintGC"

- Sample usage:
  -Xlog:help
  -Xlog:disable
  -Xlog:gc
  -Xlog:gc=trace:file=gctrace.txt:uptimemillis,pids:filecount=5,filesize=1024

Agenda

- Java Platform Module System
- Java Virtual Machine
- **Libraries and tools**

Process API Updates

- On java.lang.Process new methods to get the PID, direct children, and all descendants

- New java.lang.ProcessHandle interface for better control

```
ProcessHandle current = ProcessHandle.current();
current.info().totalCpuDuration().ifPresent(d -> {
                System.out.println("Total cpu duration :" + d);
        });
current.children().forEach(p -> System.out.println("Pid:" +
p.getPid()));
```

Convenience Factory Methods for Collections

- How many times have you written code like:
  ```
  Set<String> set = new HashSet<>();
  set.add("a");
  set.add("b");
  set.add("c");
  set = Collections.unmodifiableSet(set);
  ```
- Now:

  Set<String> set = Set.of("a", "b", "c");

- New static factory methods named "of" on Set, List, and Map

- Randomized iteration for set and maps.

- JEP 254: Compact Strings
  - Replace String-internal char[] representation (16 bits/char) with a byte[] array plus encoding field
  - Transparent to users with better memory density
- JEP 193: Variable Handles
  - Fine-grained fence operations for memory-reordering and strongly-reachable

jshell

- As a new command in $JDK/bin in JDK 9
- Less formal way for experienced developers to
  - Explore using a new API
  - Experiment with new language features
- Open CMD to try following:

```
jdk-9\bin\jshell
Jshell>/help
jshell> ProcessHandle ph = ProcessHandle.current();
jshell> ph.getPid();
jshell> ph.info().command();
```

JDK 9 Information

- Current proposed schedule: GA July 2017, weekly updates:
  - Early access binaries + docs: https://jdk9.java.net/
  - Early access binaries with cutting edge Jigsaw: https://jdk9.java.net/jigsaw/
- OpenJDK:
  - Project: http://openjdk.java.net/projects/jdk9/
  - Mailing list: http://mail.openjdk.java.net/mailman/listinfo/jdk9-dev
  - Source code: http://hg.openjdk.java.net/jdk9/dev/
  - Adoption: http://mail.openjdk.java.net/pipermail/adoption-discuss/
    https://wiki.openjdk.java.net/display/quality/Quality+Outreach/

  - JEPs (JDK Enhancement Proposals) used for project tracking