

Environment Modeling with JavaPathfinder (JPF) Part 2 - JPF Introduction

Oksana Tkachuk

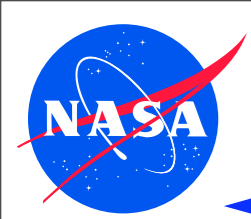
SGT / NASA Ames Research Center

<oksana.tkachuk@nasa.gov>

Peter C. Mehlitz

SGT / NASA Ames Research Center

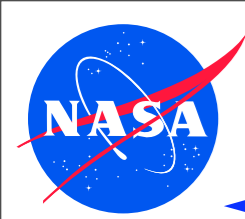
<peter.c.mehlitz@nasa.gov>



Roadmap



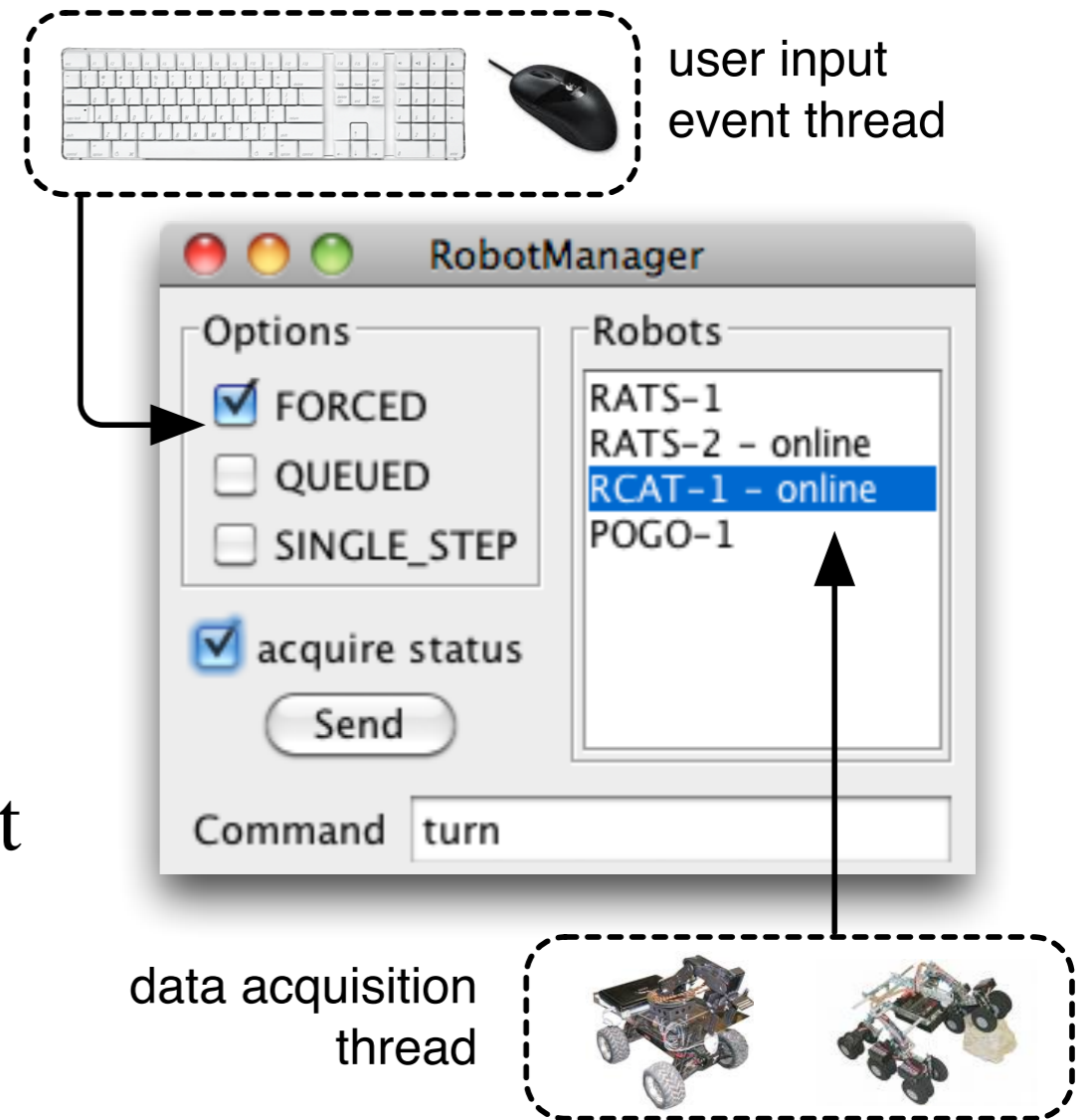
- ◆ Motivating Example
- ◆ What is JPF?
- ◆ Major Extension Mechanisms
- ◆ Anatomy of a JPF Installation
- ◆ Documentation



Example - RobotManager



- ◆ this is where we want to go
“not your usual model checking”
- ◆ small system under test
(7 classes, ~500 loc)
- ◆ graphical user interface app
⇒ uses huge framework (Swing, AWT)
400+ classes
- ◆ user input and concurrency ⇒ hard to test
- ◆ needs tool that supports
environment modeling and
faithful execution

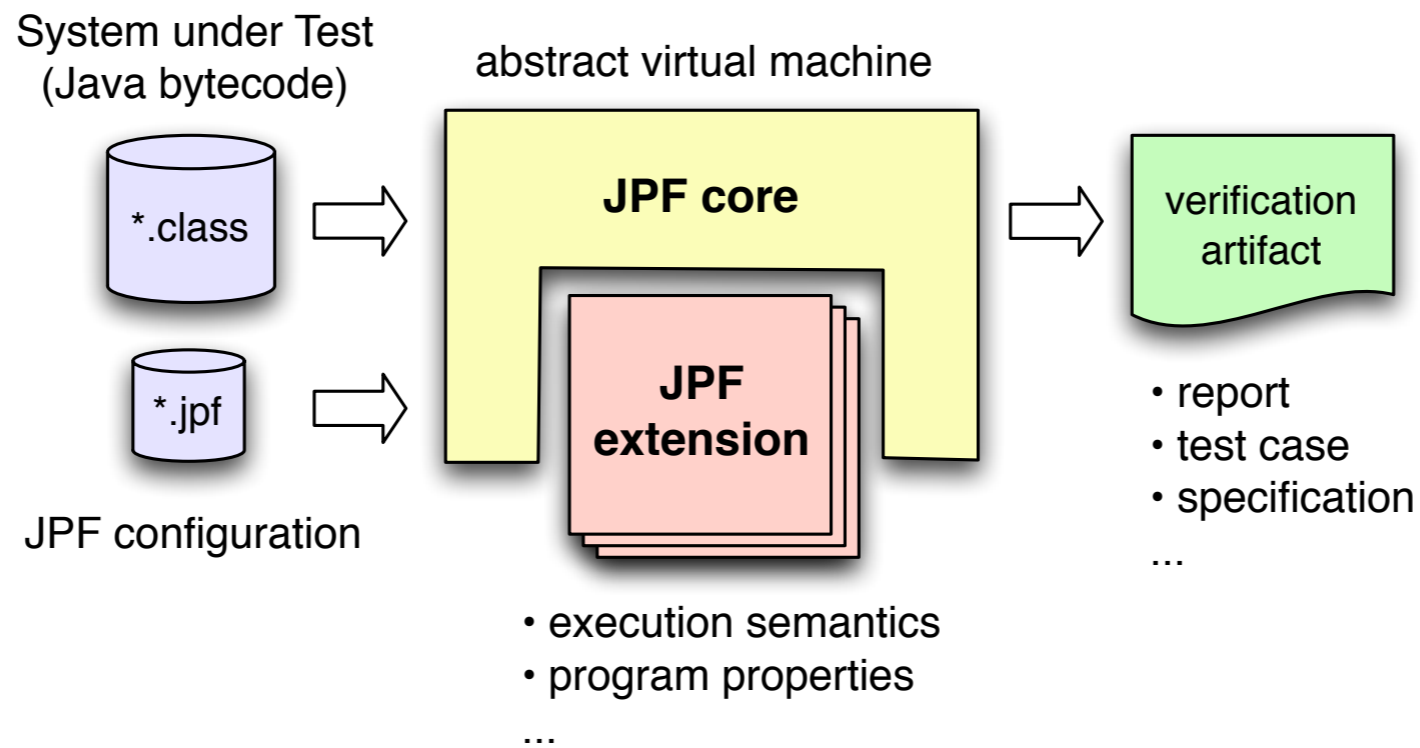


```

jpf src/examples/RobotManager-thread.jpf
...
===== error #1
NoUncaughtExceptionsProperty
java.lang.NullPointerException: Calling 'processSequence(String)' on null object
    at RobotManager.sendSequence(RobotManager.java:265)
    at RobotManagerView.sendSequence(RobotManager.java:565)
...

```

◆ a model checker? a virtual machine? ..

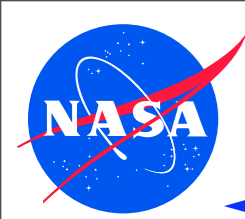


◆ ..and the answer is: both, and more - it depends on you

◆ *not* a monolithic, black box tool

No “one size fits all” - Extensibility is Paramount

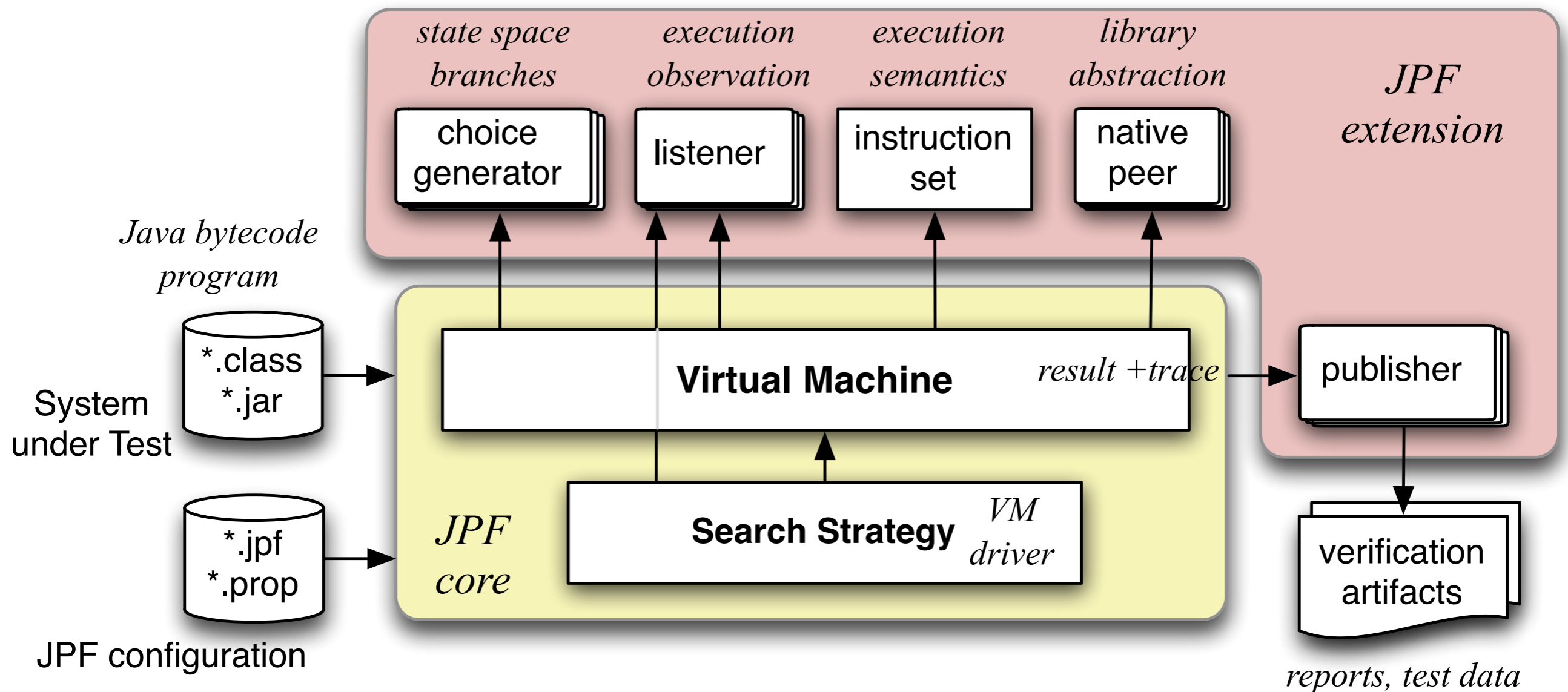
◆ the quest of today: learn what is in the toolbox to find out how you can adapt JPF to *your* needs

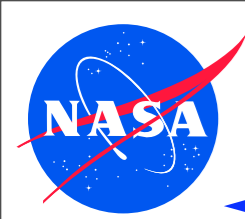


What is JPF - Core and Extensions



- ◆ partitioned into core and extensions
- ◆ core consists of Search and Virtual Machine
- ◆ several extension mechanism

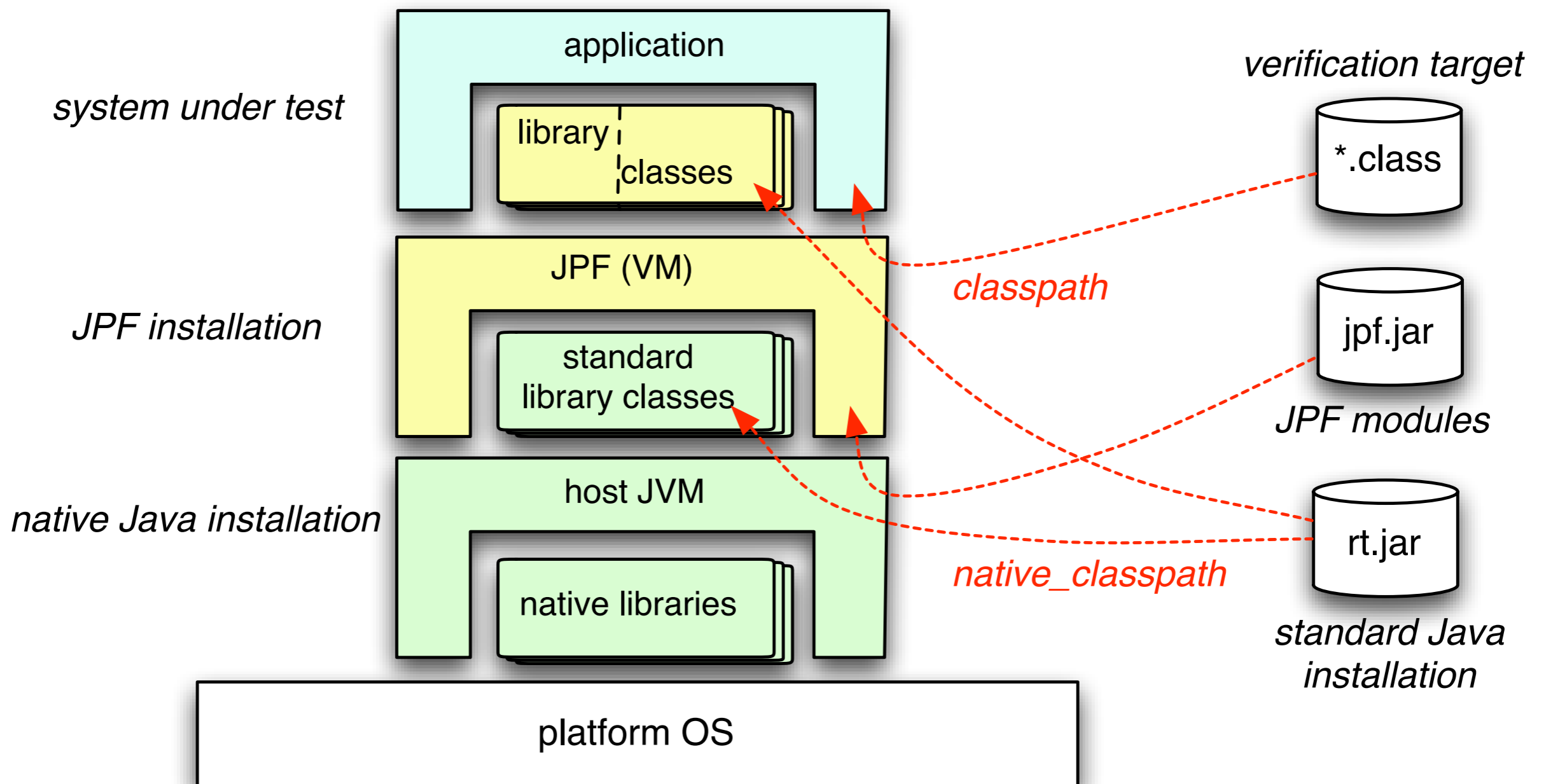


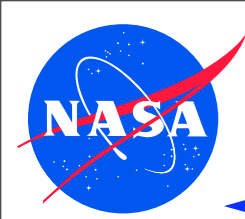


What is JPF - VM running in VM



- ◆ main stumbling block is recursive nature of JPF
- ◆ JPF's Virtual Machine (VM) runs as a normal Java application on top of a host Java VM





JPF Extensions - Main Mechanisms

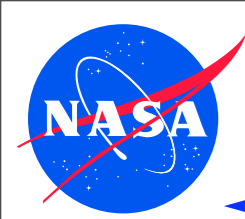


- ◆ ChoiceGenerators - Transitions and States
- ◆ VM and Search Listeners - the JPF “plugins”
- ◆ Native Peers - Model Java Interface (MJI)

used in lab

-
- ◆ Object Attributes - data flow
 - ◆ Instruction Factories - execution semantics
 - ◆ ... and many more (Serializers, Publishers, PublisherExtensions, ...)

FYI

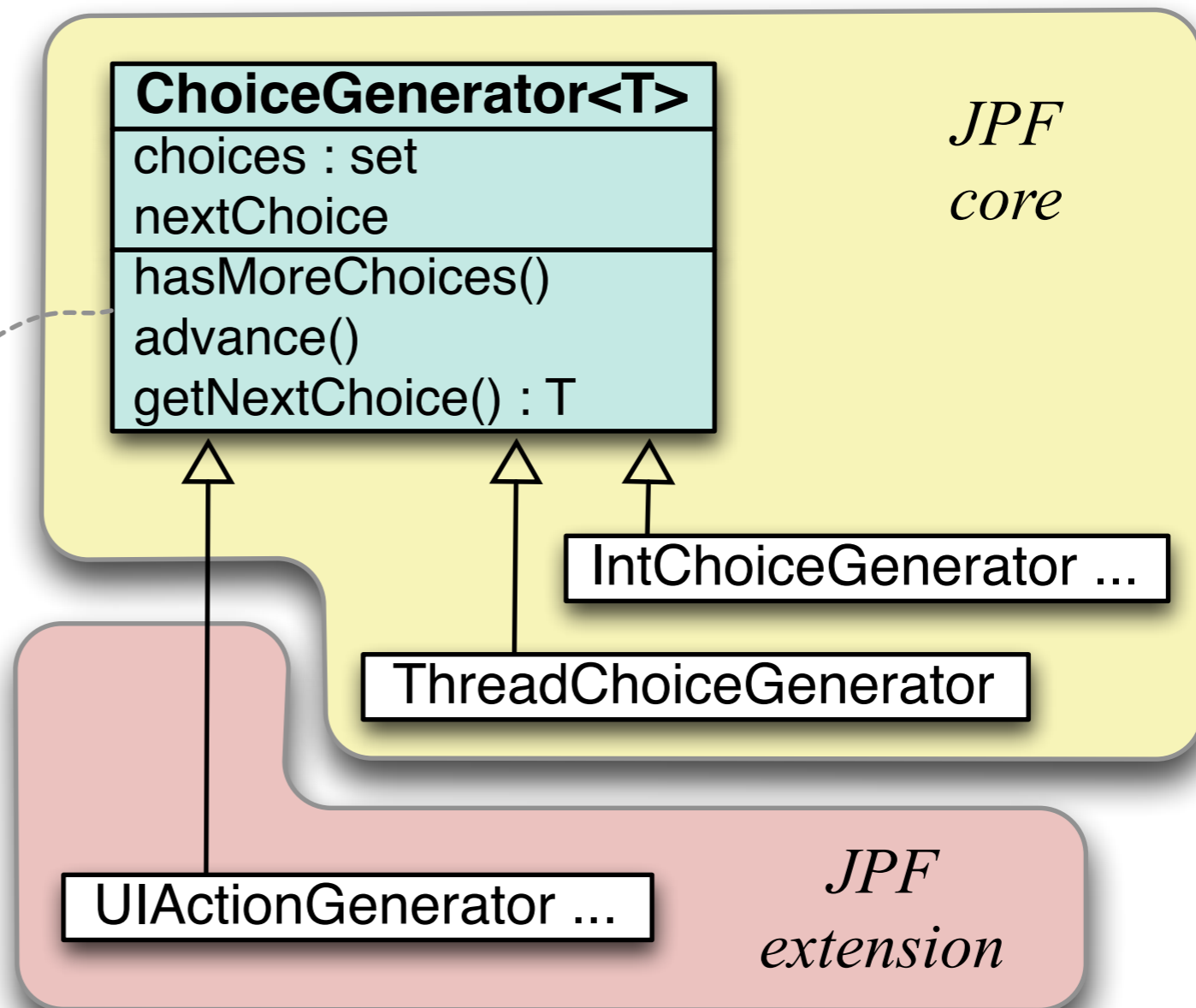
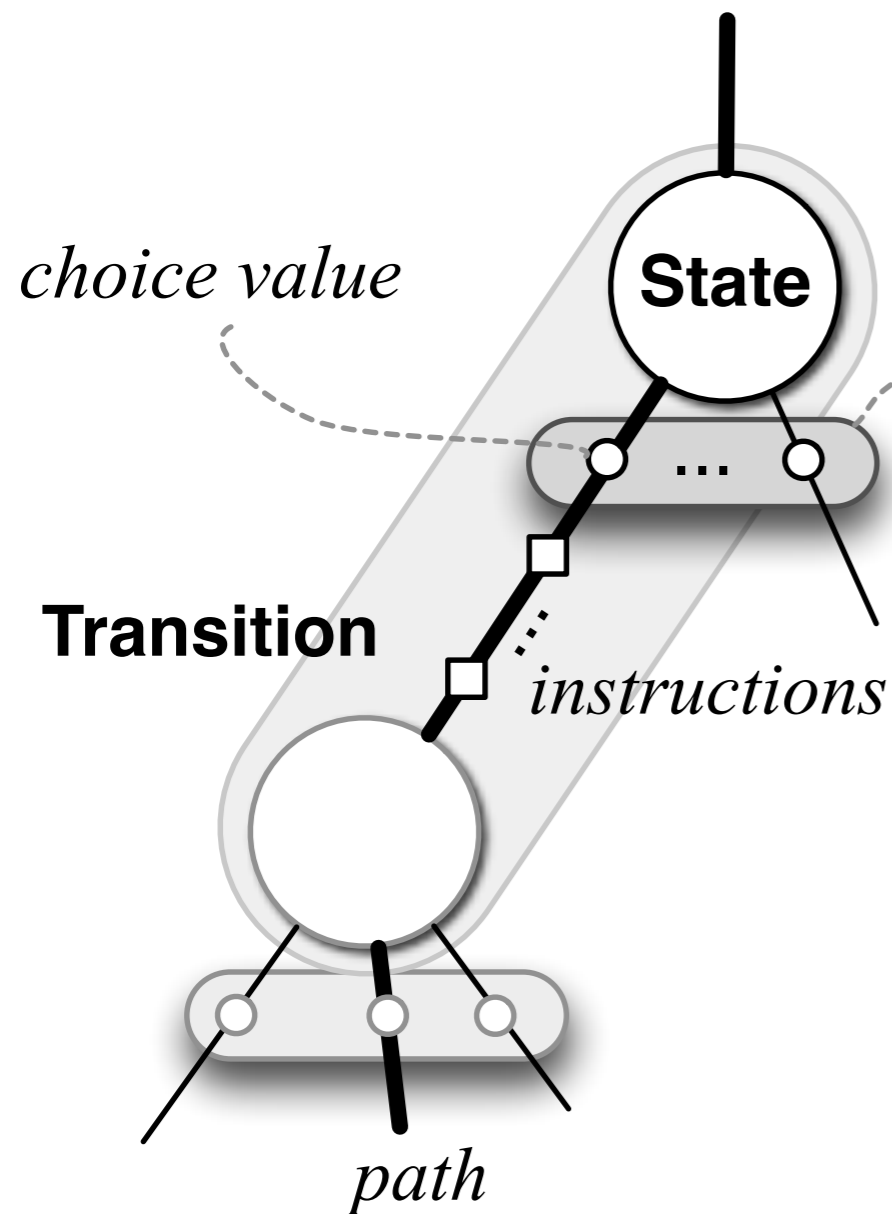


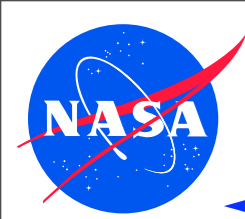
JPF Extensions - ChoiceGenerators



- ◆ nature of state space branches not hardwired into JPF
- ◆ user can introduce new choice types by providing ChoiceGenerators

*iterator objects
for statespace branches*

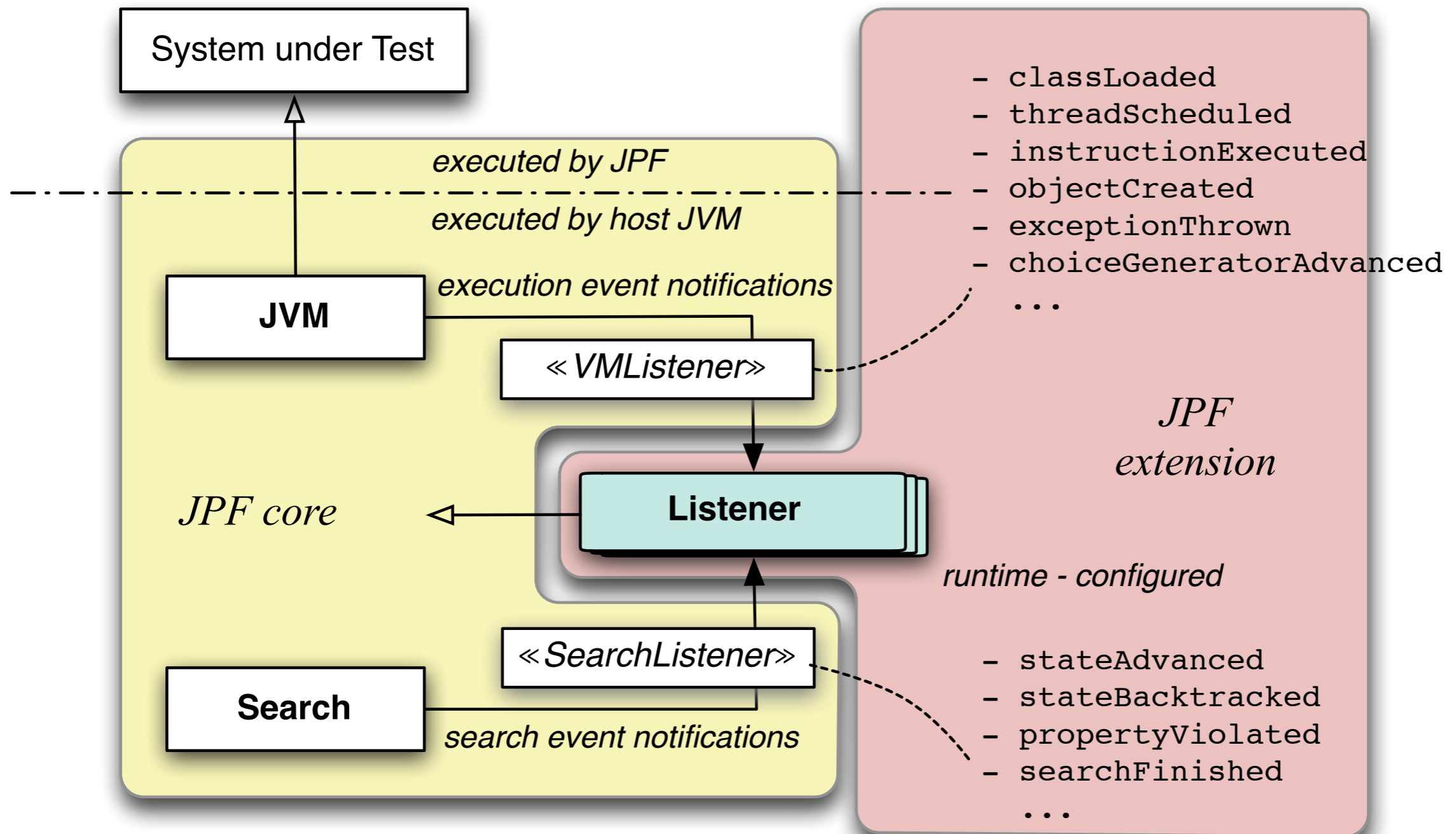


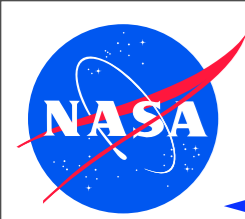


JPF Extensions - Listeners



- ◆ the primary JPF “plugins”
- ◆ used to monitor execution & search (e.g. to implement properties)
- ◆ runtime configurable

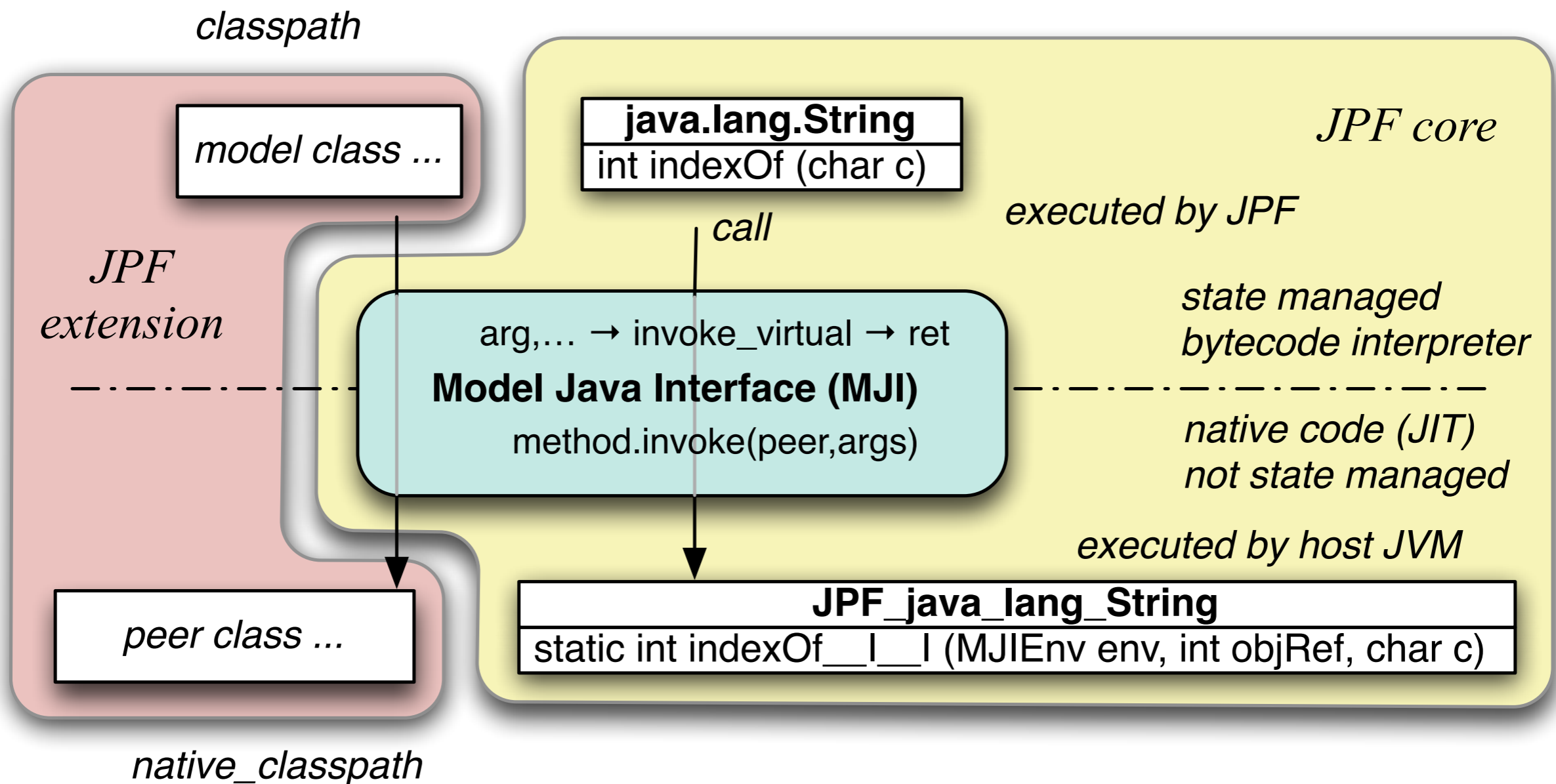


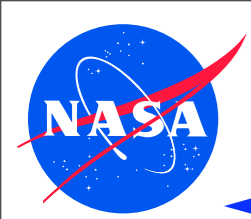


JPF Extensions - Native Peers



- ◆ preferred way to model libraries that use platform specific code (I/O, GUI frameworks etc.)
- ◆ JPF's Model Java Interface (MJI) resembles Java Native Interface (JNI)
- ◆ lowers execution from JPF VM into host VM

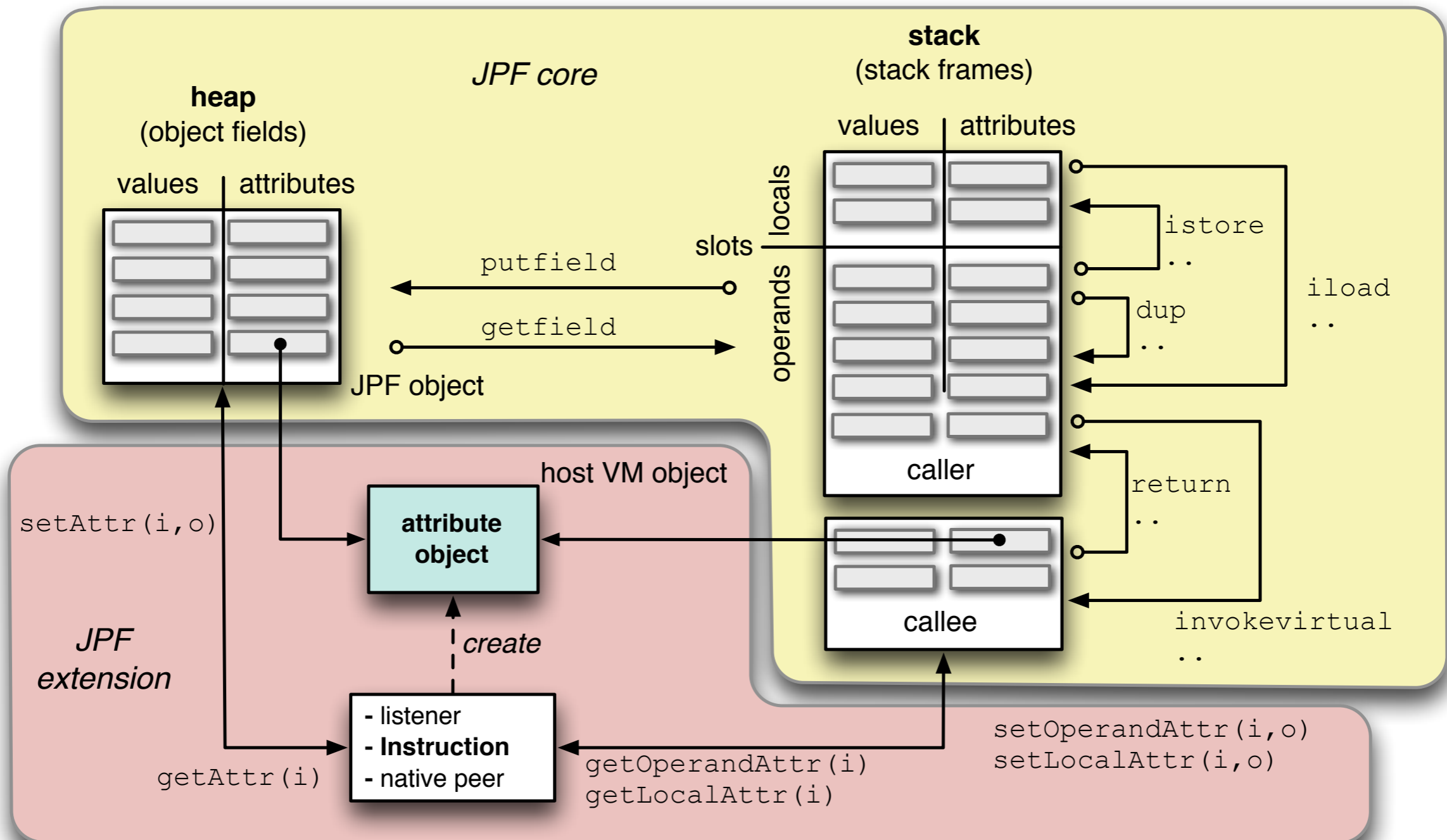


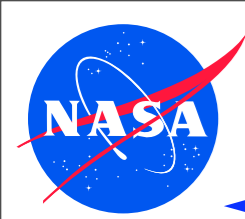


JPF Extensions - Attributes



- ◆ POJO's that can be attached to many types of JPF objects
- ◆ travel with data - good to implement data flow properties

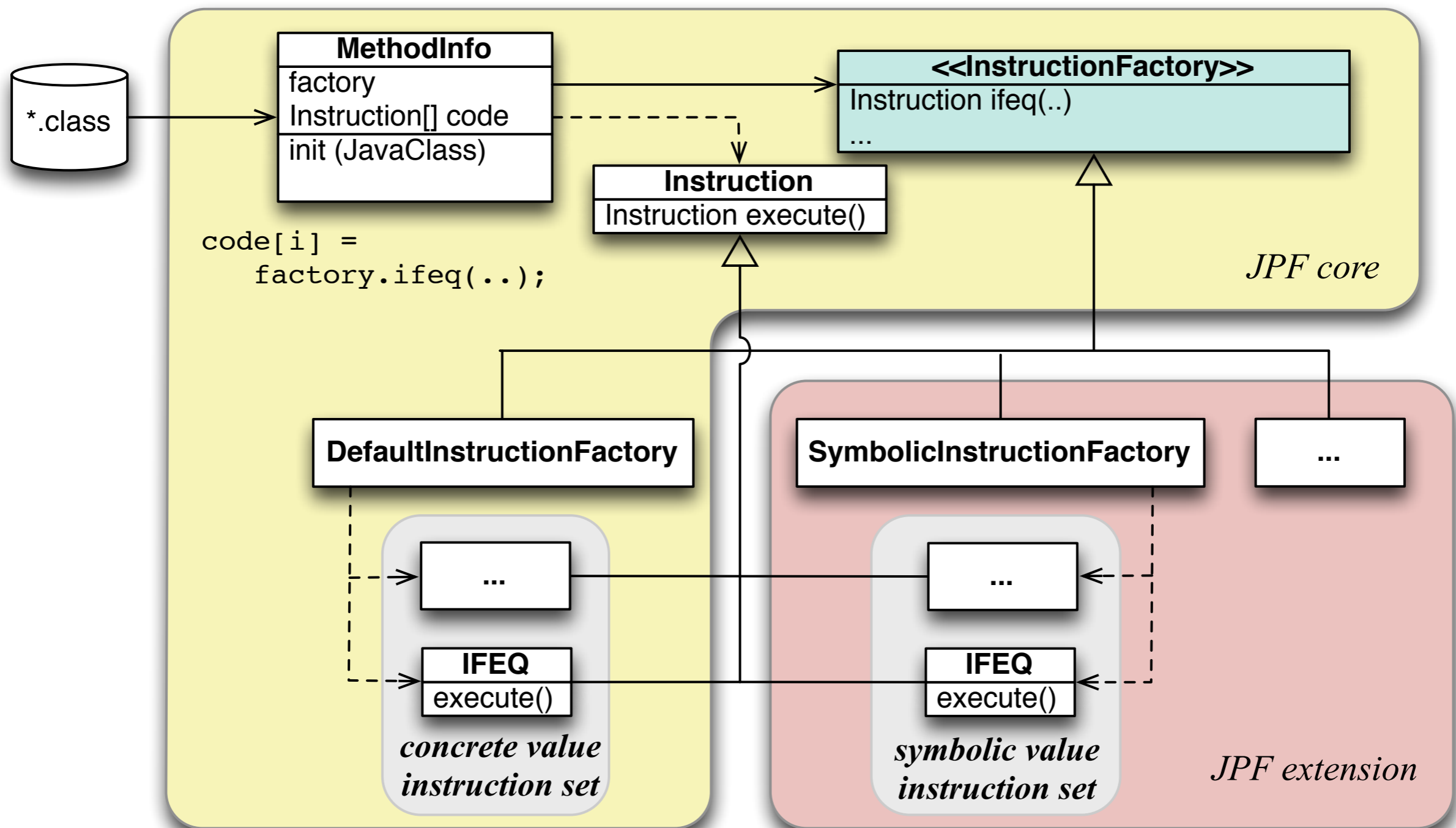


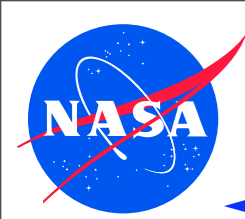


JPF Extensions - Instruction Factories



- ◆ execution semantics encapsulated in JPF Instruction objects
- ◆ can be replaced, e.g. to add additional operations

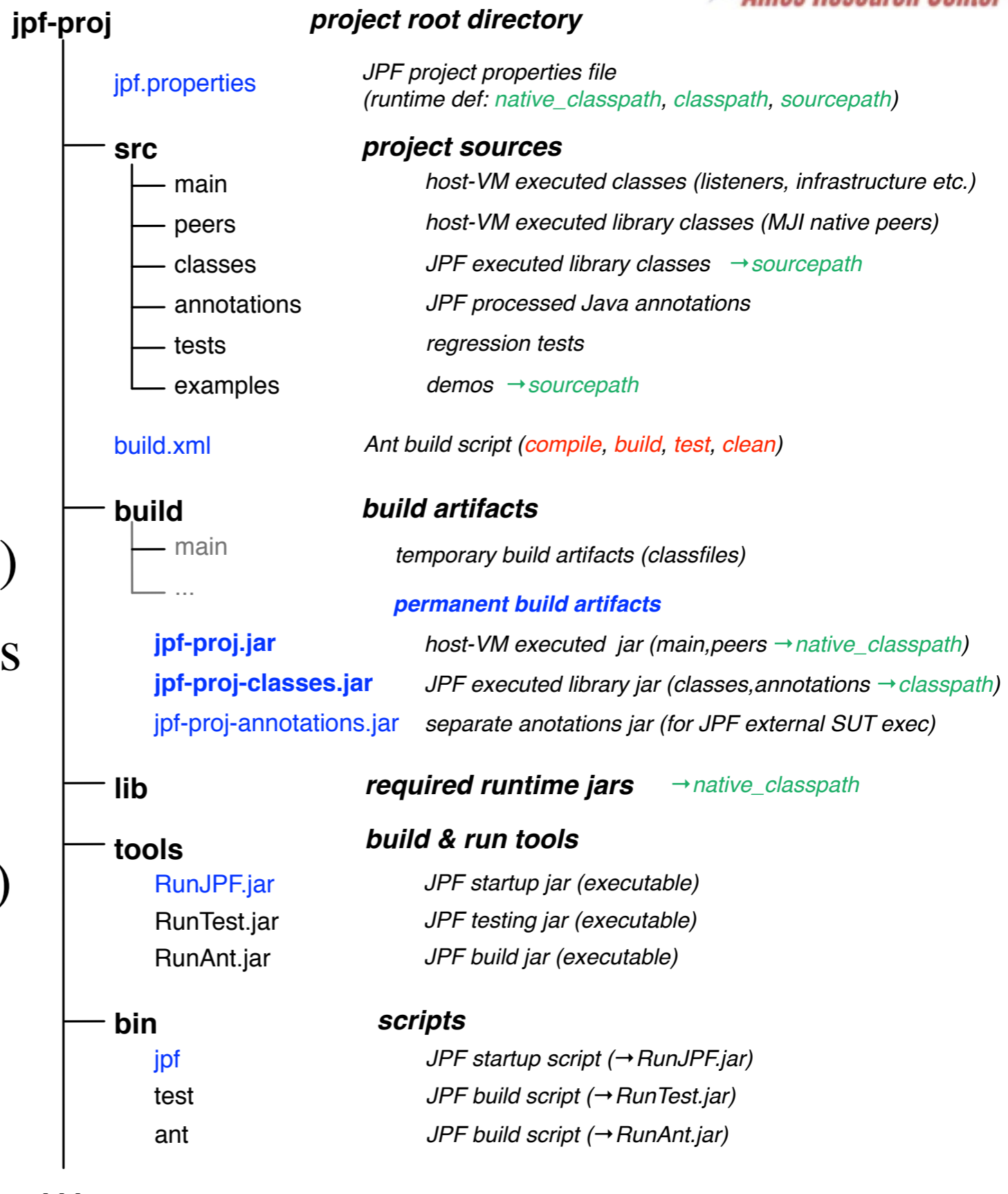


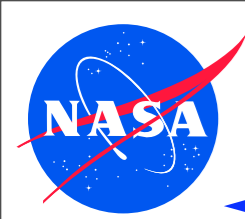


JPF Anatomy - Project layout



- ◆ all JPF projects share uniform directory layout
- ◆ use ANT based build system
- ◆ use same configuration scheme
- ◆ support binary distributions
- ◆ 3rd party tools & libraries can be included (self-contained)
- ◆ projects (should) have examples and regression test suites
- ◆ projects (should) have IDE configuration (NB, Eclipse)





JPF Anatomy - Configuration



command line property arguments

```
> bin/jpf [-log][-show] {+log.info=..} ../RobotManager.jpf
```

```
target = RobotManager
target_args = ...

@using = jpf-aprop
@import = ./my.properties

shell = .shell.basicshell.BasicShell
listener = .aprop.listener.SharedChecker
...
```

application properties

jpf.properties in current directory

```
jpf-core = ${config_path}
jpf-core.native_classpath=\
  ${jpf-core}/build/jpf.jar; \
  ...
  ${jpf-core}/lib/bcel.jar;

jpf-core.classpath=\
  build/jpf-classes.jar

jpf-core.test_classpath=\
  build/tests

jpf-core.sourcepath=\
  src/classes
...
```

all jpf.properties in order of extensions

```
jpf-awt-shell = ${config_path}
@using = jpf-awt

jpf-awt-shell.native_classpath=...
jpf-awt-shell.classpath=...
...
```

site properties

```
jpf.home = ${user.home}/projects/jpf

jpf-core = ${jpf.home}/jpf-core
jpf-awt = ${jpf.home}/awt
jpf-shell = ${jpf.home}/jpf-shell
jpf-aprop = ...
...
extensions = ${jpf-core},${jpf-shell}
```

Configuration Levels

4. command line

↑
- debugging
(+key=value pairs)

3. application properties

<project>/.../*.jpf
- system-under-test
- listeners, shells

↑

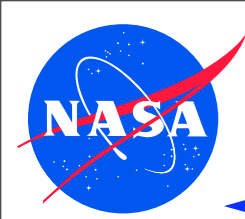
2. project properties

<project>/jpf.properties
- project class paths
- project dependencies

↑

1. site properties

~/.jpf/site.properties
- project locations
- pre-loaded projects



JPF Documentation & Help - Wiki



http://babelfish.arc.nasa.gov/trac/jpf

- public read access
- edit for account holders (also non-NASA)

bug tracking

- Trac ticket system

The screenshot shows the JPF Wiki website. The browser address bar contains the URL <http://babelfish.arc.nasa.gov/trac/jpf/wiki>. The page header includes the JPF logo and the tagline "the swiss army knife of Java™ verification". A navigation bar contains buttons for "JPF-Wiki", "Timeline", "Roadmap", "View Tickets", "New Ticket", "Search", "Admin", and "Blog". Below the navigation bar is a "Latest JPF News" section with a table of recent updates. A sidebar menu on the right lists various sections of the wiki, including "Introduction...", "Installing JPF...", "User Guide...", "Developer Guide...", "Projects...", "Change(B)log", "About...", "Papers", "FAQ", "Playground", and "Table of Context".

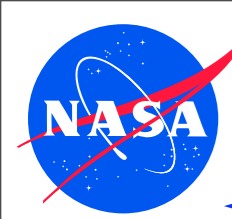
Date	News Item
02/14/2010	ISSTA 2010 Tutorial on Automated Testing with Java PathFinder announced
02/12/2010	Call for Google Summer of Code 2010 project proposals out on JPF Google group
01/30/2010	JPF Google group replaces old mailing lists
01/12/2010	Fujitsu press announcement released about using and extending Symbolic PathFinder (projects/jpf-symbc) for comprehensive testing of Java web applications
09/02/2009	JPF server on http://babelfish.arc.nasa.gov/trac/jpf goes live, featuring the JPFWiki and separate Mercurial repositories for JPF core and extension projects
07/22/2009	JPF wins the 2009 "Outstanding Technology Development Award" of the Federal Laboratory Consortium (FLC), Far West Division

project blog

- announcements
- important changes

hierarchical navigation menu

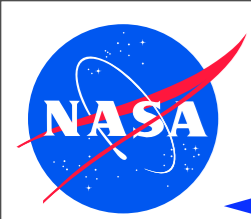
- intro
- installation
- user docu
- developer docu
- **extension projects**



JPF Documentation & Help - Mailing List



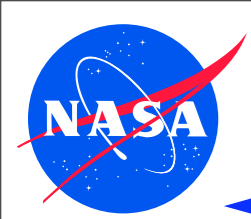
- ◆ <http://groups.google.com/group/java-pathfinder>
- ◆ anyone can join & read, member only post
- ◆ moderate volume (~50 messages/month)
- ◆ subscribe on <http://groups.google.com/group/java-pathfinder/subscribe>



Conclusions



- ◆ JPF not a black box tool
- ◆ highly extensible
- ◆ can - and has to - be customized for domain specific verification
- ◆ takes some time to master



Conclusions



- ◆ JPF not a black box tool
- ◆ highly extensible
- ◆ can - and has to - be customized for domain specific verification
- ◆ takes some time to master

Thank You!
See you at the lab