Four Productive Ways to Use Open Source JFR and JMC

DEFENCE AND SPACE

Martin Klähn
05 Feb 2020

AIRBUS

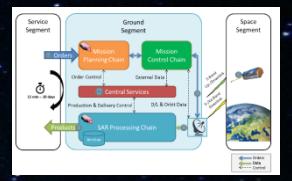
About Me

- Software Engineer @ Airbus Defence and Space
- Co-Founder of JUG Bodensee
- Member of Apache NetBeans PMC
- NetBeans Community Acceptance Testing (NetCAT)
- JCP Associate Member

About Us

Radar Upstream Ground Segment & Payload Data Processing
Your Connectivity between Satellites and Users

Turn Key Systems



Operational Data Processing
with high performance computing

Crypto & Security Engineering
for space and ground

Coherent Engineering for Systems
incl. end-to-end verification

Analysis and Design
of complex space ground segments

AIRBUS

Agenda

- Introduction to JFR and JMC
- Have fun
  - On the command line
  - Using a nice UI
  - Using your favourite IDE
  - With unit tests

Introduction to JFR and JMC

- What is it?
  - JVM tools suite
    - JDK Flight Recorder
    - JDK Mission Control
  - Helps finding problems in, and optimizing, programs running on the JVM in production

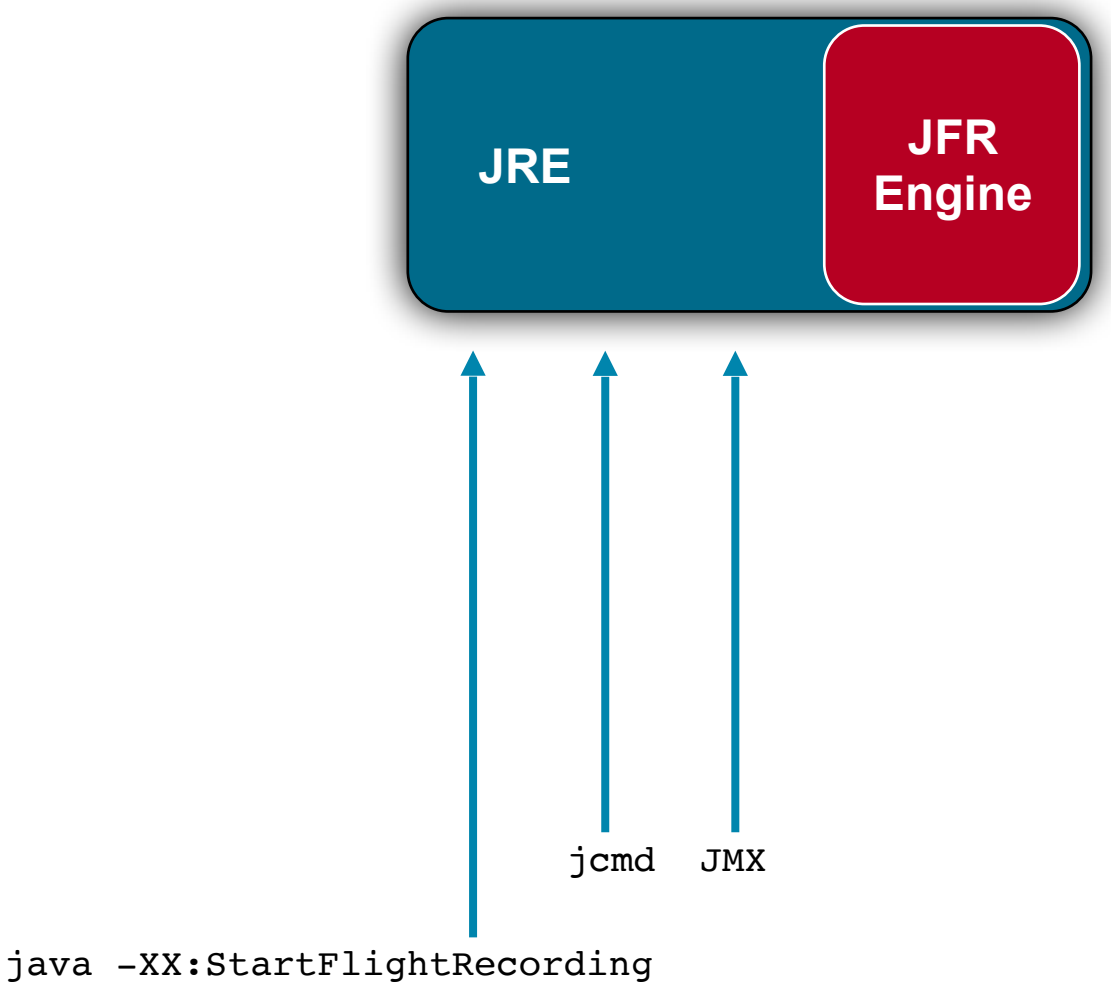
- Where does it come from?
  - Started out as JRockit Mission Control as a part of the JRockit JVM distribution by Appeal Virtual Machines, a Swedish company created in 1998 by students from the Royal Institute of Technology in Stockholm
  - Commercial feature of Oracle JDK up to JDK 10
  - Open sourced and rebranded for JDK 11

JDK Flight Recorder

- What is it?
  - Event recorder built into the JVM
  - Low overhead, always on, runtime monitoring and profiling information gathering part in the JVM
  - Designed to minimize the Observer Effect
  - Able to many multiple recordings

- What is a recording?
  - Binary chunks of binary data
  - Each chunk is self describing and self contained
  - Efficiently stored (binary format, ring buffer, thread local storage, …)

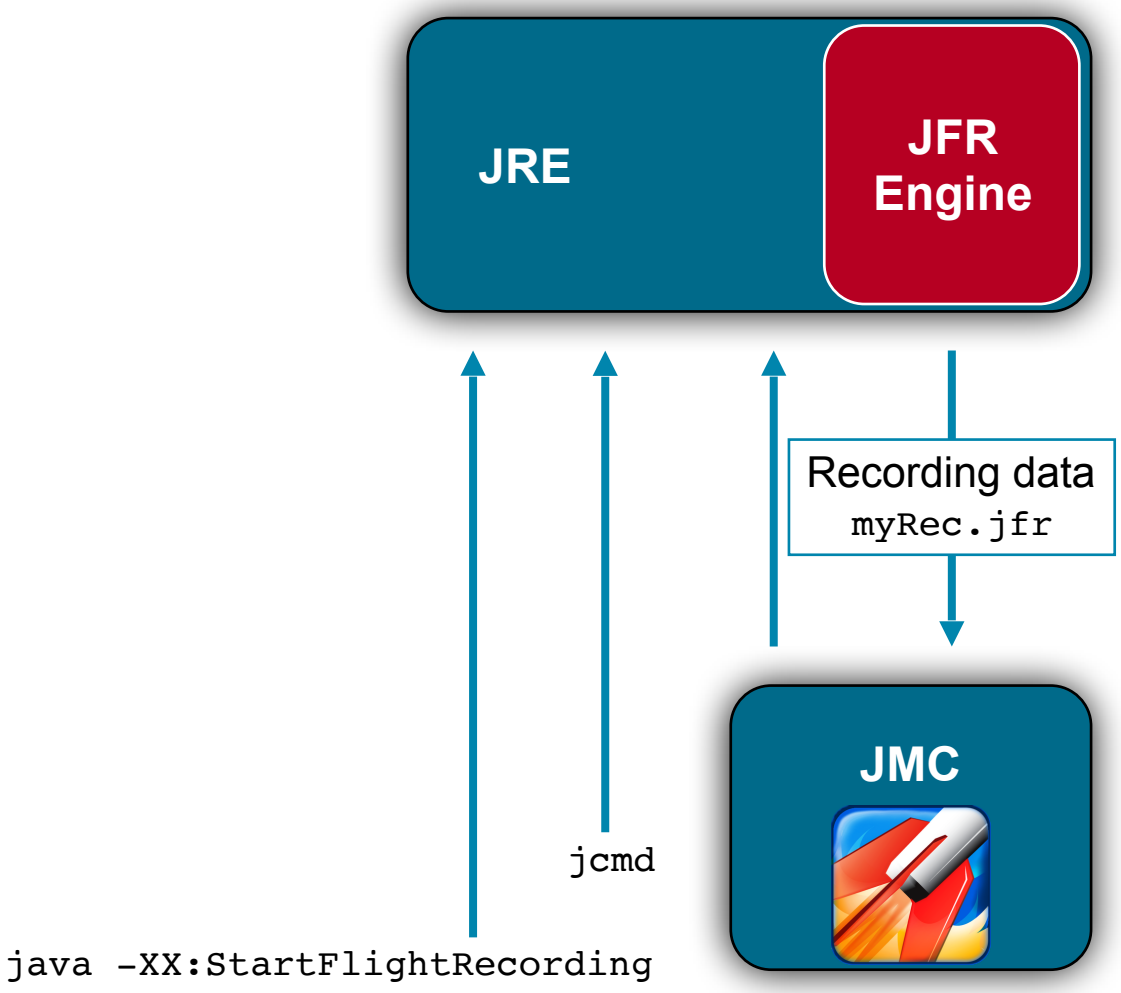
- How do I get recordings?
  - JCMD / Commandline Options / JMX / JDK Mission Control / JVisualVM



AIRBUS

JDK Mission Control

- What is it?
  - Analysis & Visualisation tool for JFR recordings
- Where can I get it?
  - https://www.oracle.com/technetwork/java-mission-control-1998576.html
- What else does it provide?
  - APIs to read and process JFR recording
  - Artefacts
    - Not yet available from Maven Central yet
    - Available from https://adoptopenjdk.jfrog.io/adoptopenjdk



Fun with the Commandline

- What you need
  - Shell
  - A bit of JIGSAW-Fu
  - Some of the JFR Libraries

- Links:
  - jmc-jshell
    - Wrapper for jshell handling classpath configuration to use JFR APIs within JShell
    - https://github.com/thegreystone/jmc-jshell

Fun Creating a Standalone UI

- What you need
  - JavaFX
  - A bit of JIGSAW-Fu (always good)
  - Some of the JFR Libraries
  - Some of the JMC Libraries

Fun Using Your Favourite IDE

- What you need
  - JavaFX
  - A bit of JIGSAW-Fu (always good)
  - Some of the JFR Libraries
  - Some of the JMC Libraries
  - Your favourite IDE

Fun With Unit Tests

- What you need
  - JUnit
  - Some of the JFR Libraries
  - Writing your own JFR Events
    - See https://www.morling.dev/blog/rest-api-monitoring-with-custom-jdk-flight-recorder-events/

Thank you