



z/TPF support for Java

Welcome to a new era in z/TPF language support!

Chris Filachek

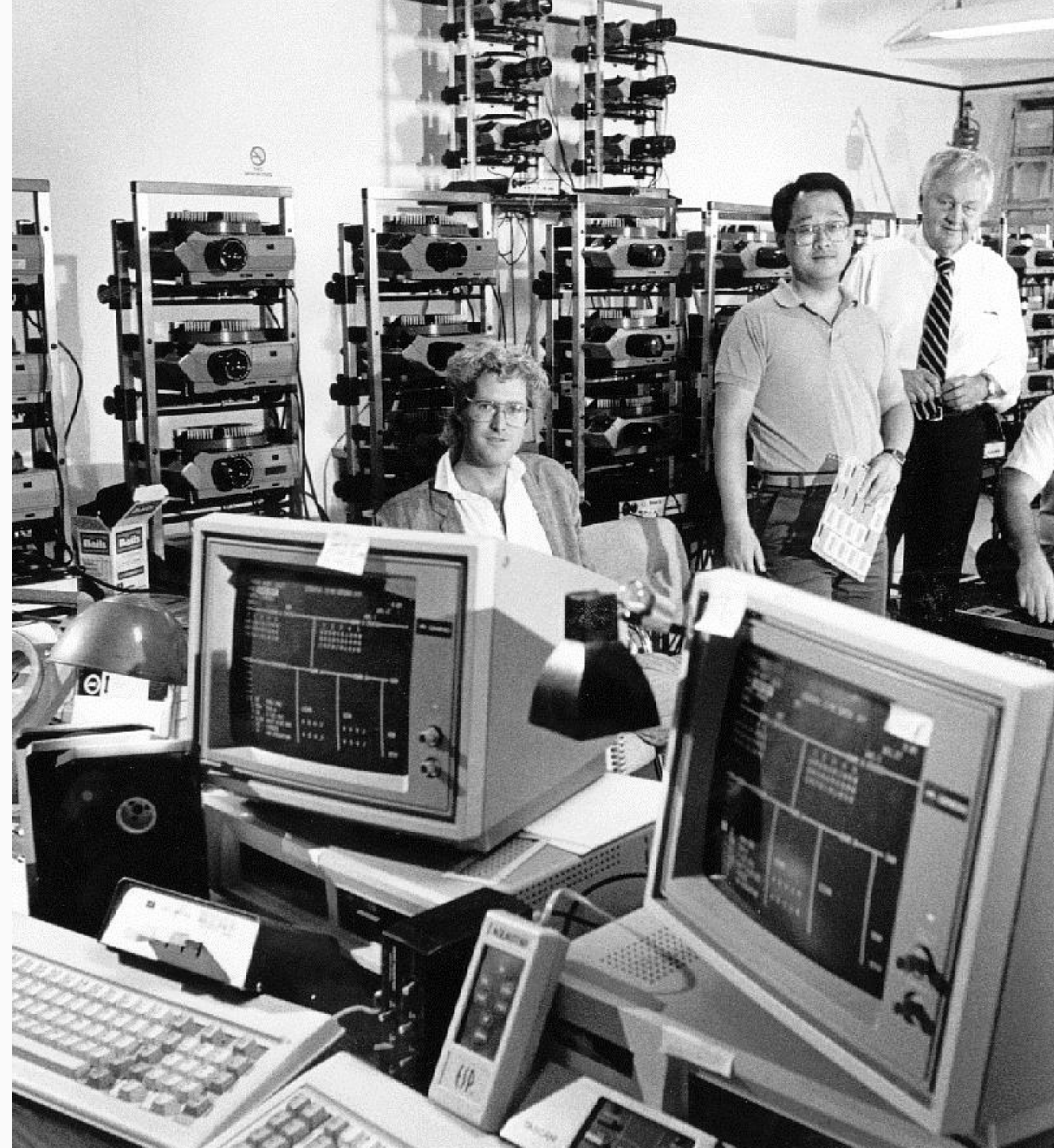
z/TPF and z/TPFDF Architecture & Development

IBM z/TPF

April 4, 2017

Maintaining the vitality of our development organizations

- Experts are retiring
- Time needed to build new expertise



A **Java programmer** can create new or extend existing z/TPF applications using Java **without requiring any z/TPF knowledge.**

z/TPF



IBM 64-bit Runtime
Environment for z/TPF,
Java™ Technology Edition,
Version 8

z/TPF



IBM 64-bit Runtime
Environment for z/TPF,
Java™ Technology Edition,
Version 8

What kind of Java is this?

Java-Lite

“Java”

TPF-Java

Decaf

Mostly Java

Sorta-Kinda-Maybe-Java

z/TPF



...is compatible with
Oracle Platform
Java Standard Edition
(Java SE) 8

IBM 64-bit Runtime
Environment for z/TPF,
Java™ Technology Edition,
Version 8

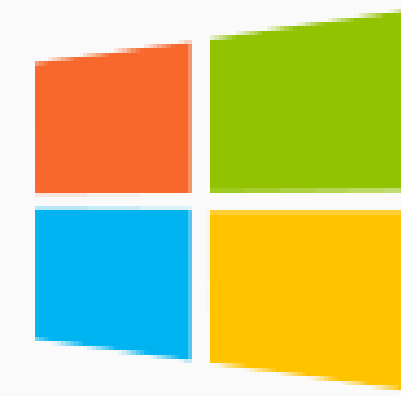


Java Compatible – Familiar development environment



Code

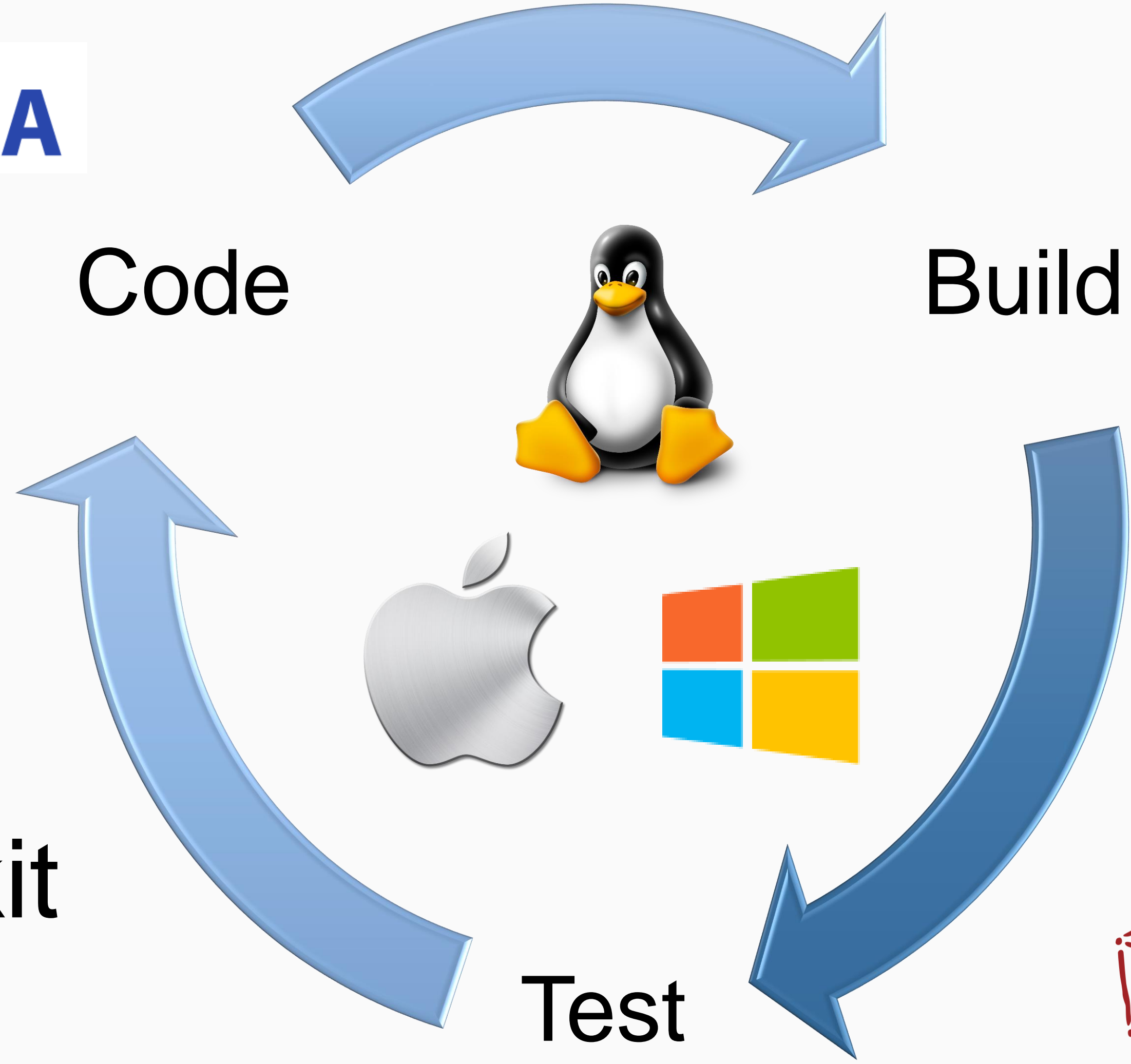
Build



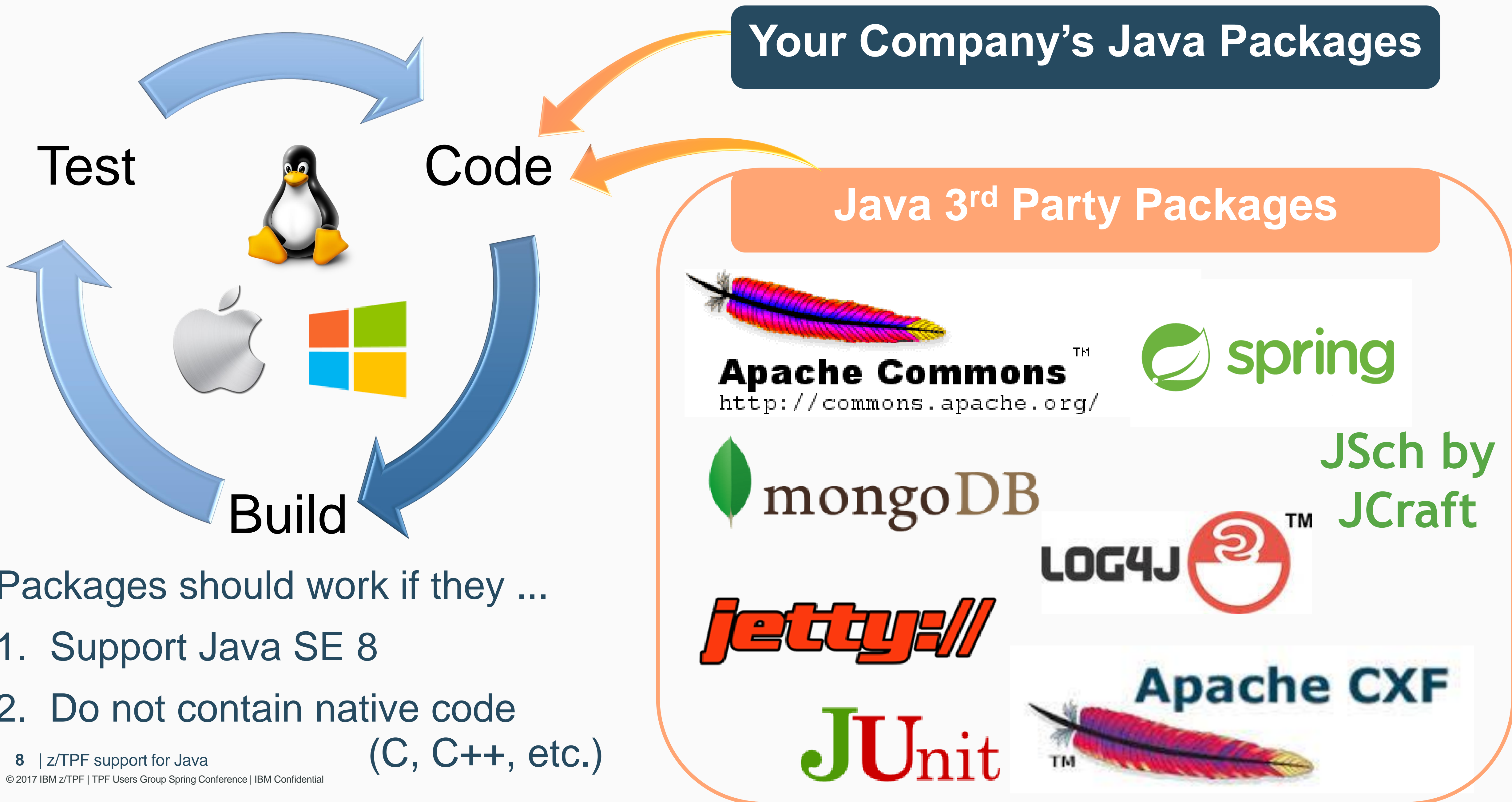
Test



NetBeans



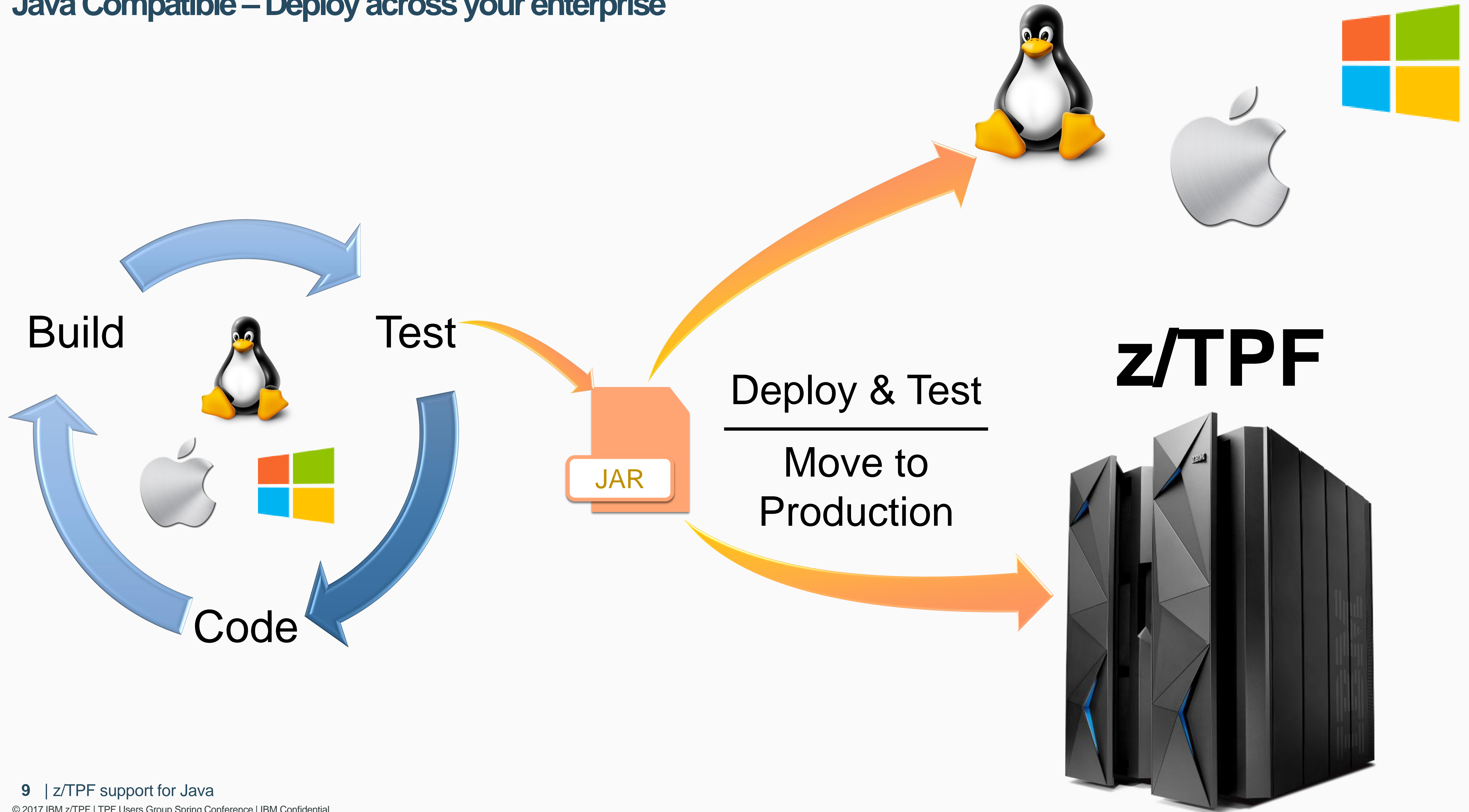
Java Compatible – Easily use existing Java packages



Packages should work if they ...

1. Support Java SE 8
2. Do not contain native code (C, C++, etc.)

Java Compatible – Deploy across your enterprise



Java Compatible – Familiar Java commands

Display Java version

```
AAES0008I 00 ==> zfile java -version
CSMP0097I 14.43.01 CPU-B SS-BSS SSU-HPN IS-01
FILE0002I 14.43.01 START OF ERROR DISPLAY FROM java -version
java version "1.8.0"
Java(TM) SE Runtime Environment (build pxz6480sr3fp21-20170116_01(SR3 FP21))
IBM J9 VM (build 2.8, JRE 1.8.0 TPF s390x-64 20170118_333732 (JIT enabled, AOT enabled)
J9VM - R28_Java8_SR3_20170118_1821_B333732
JIT - tr.r14.java.green_20161111_127476
GC - R28_Java8_SR3_20170118_1821_B333732
J9CL - 20161108_325074)
JCL - 20161111_01 based on Oracle jdk8u111-b14
END OF DISPLAY+
```

Run a JAR file

```
AAES0008I 00 ==> zfile java -jar /sys/tpf_pbfiles/bin/HelloWorld.jar
CSMP0097I 14.54.45 CPU-B SS-BSS SSU-HPN IS-01
FILE0001I 14.54.45 START OF DISPLAY FROM java -jar /sys/tpf_pbfiles/bin/Hell...
Hello World from Java(TM)
END OF DISPLAY+
```

A **Java programmer** can extend existing z/TPF applications with Java **without writing any transformation code or TPF-to-Java linkage.**

Create New ✓

Java application
(ZFILE java)

Existing z/TPF applications

Extend existing Apps ?

???

Java application

z/TPF calling Java – Programmer's View

z/TPF
Programmer

Call a C
function

```
struct pricingRequest
{
  unsigned int ffid;
  struct flightData
  {
    char departDate[6];
    char origCity[4];
    char departTime[6];
    char destCity[4];
  };
  char addons[3];
};
```

z/TPF
ECB

Java
Process

Convert C
structures
to Java
Objects

```
class pricingRequest
{
  public:
    getFfid();
    setFfid();
    ...
    setAddons();

  private:
    long ffid;
    String addons;
};
```

Java
Programmer

Create a
REST
Service

Existing z/TPF applications

```
tpf_srvcInvoke("pricingLookup",
               &requestData,
               ...);
```

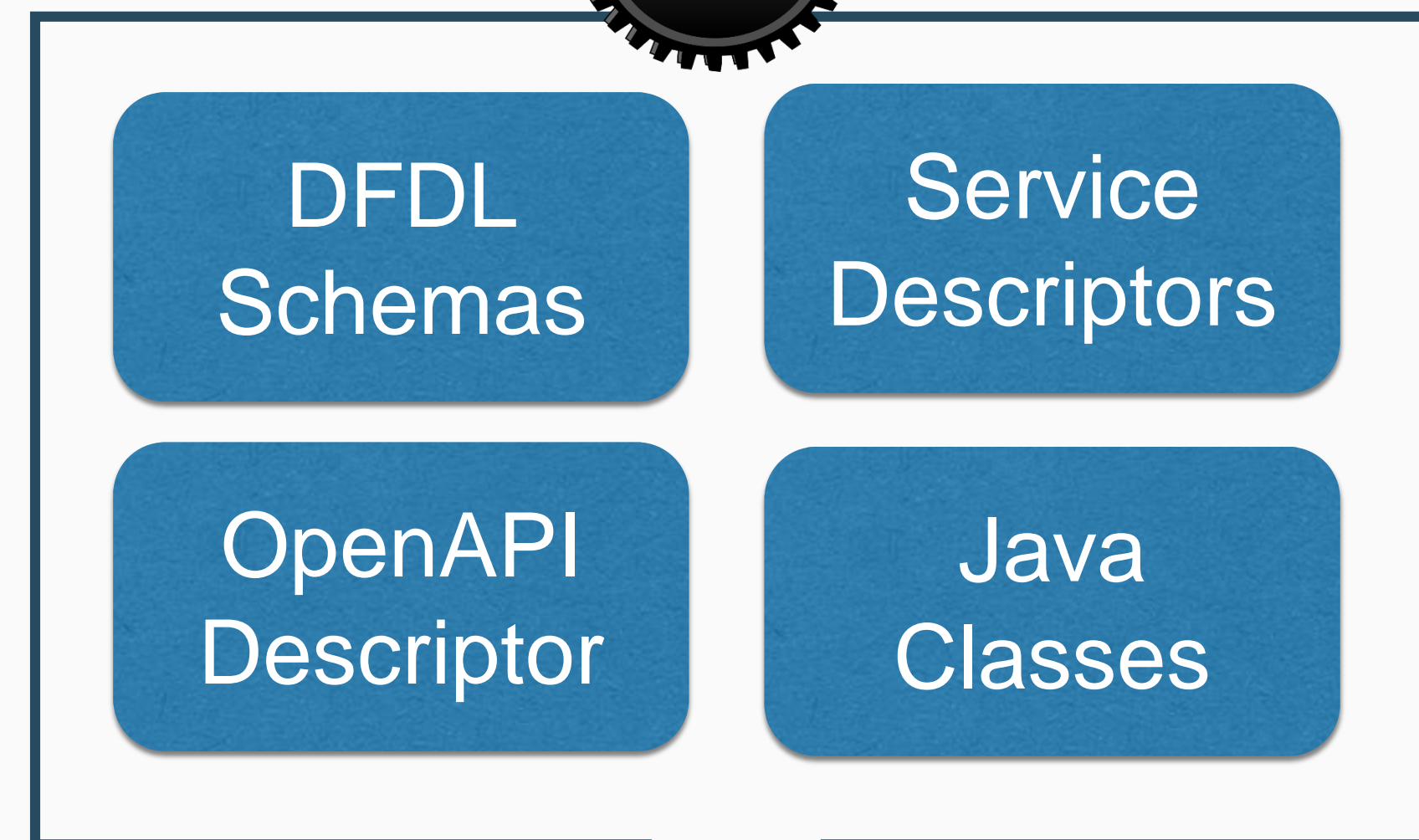
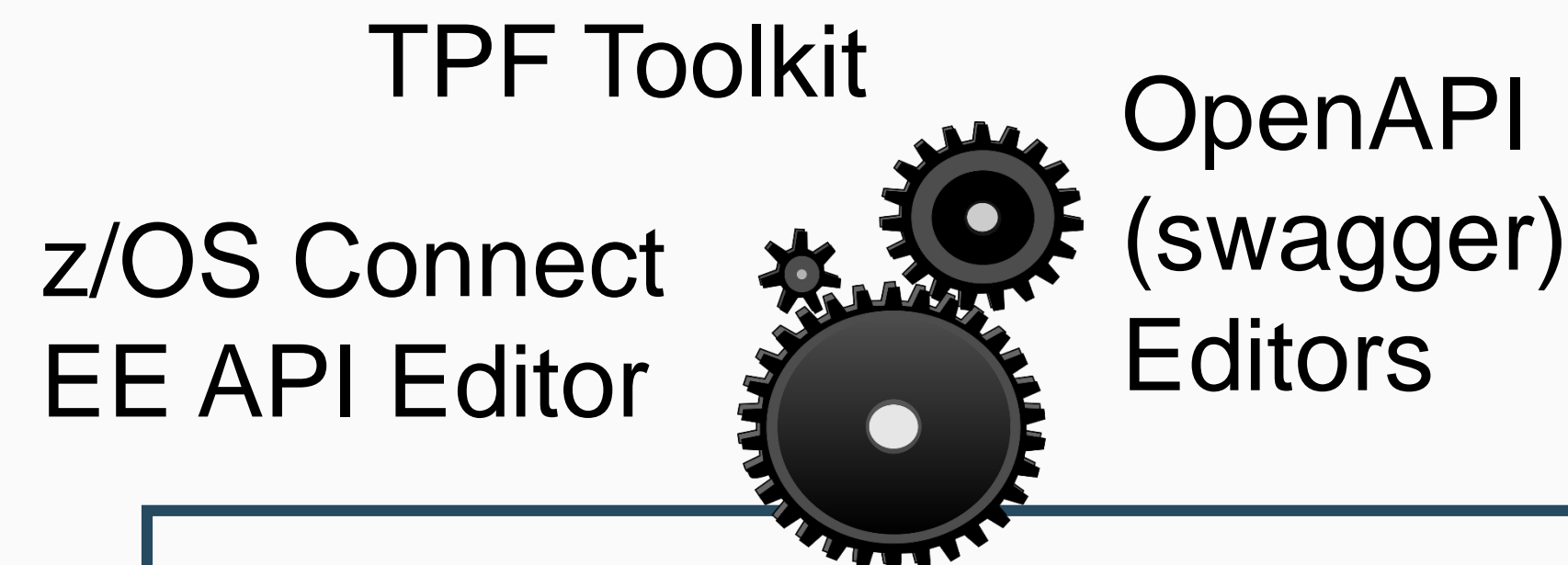
Service applications in Java

```
public class PricingServicesImpl {
  @GET
  @Path("/priceLookup")
  public Response
  priceLookup(PricingRequest
  pricingRequest)
  { ... }
```

Defining the z/TPF to Java Interface

1. DFDL schemas: Defines request / response data for each service
2. Service descriptors: Defines a service
 - Example: “pricingLookup” service
3. OpenAPI descriptors: Define service applications
 - Groups 1 or more services together
4. Generate JAX-RS CXF server classes and stubs
 - OpenAPI descriptor is input to generator
 - Write business logic in generated Java stubs
 - Packaged as a Service Application in Java

Tooling available to help create descriptors, Java classes, etc.



Convert C structures to Java Objects

z/TPF Application Manager for Java (JAM)

- Each JAM descriptor defines and manages a set of service applications in Java
- Runs the service applications in long-running Java processes (JVMs)
 - Each JVM runs all service applications for a specific JAM
 - Application threads wait for requests and can process any service for that JAM
- Defines Java environment for JVMs
 - Number of JVMs
 - Number of application threads per JVM
 - Java options (classpath, memory, etc.)

z/TPF Application Manager for Java (JAM)

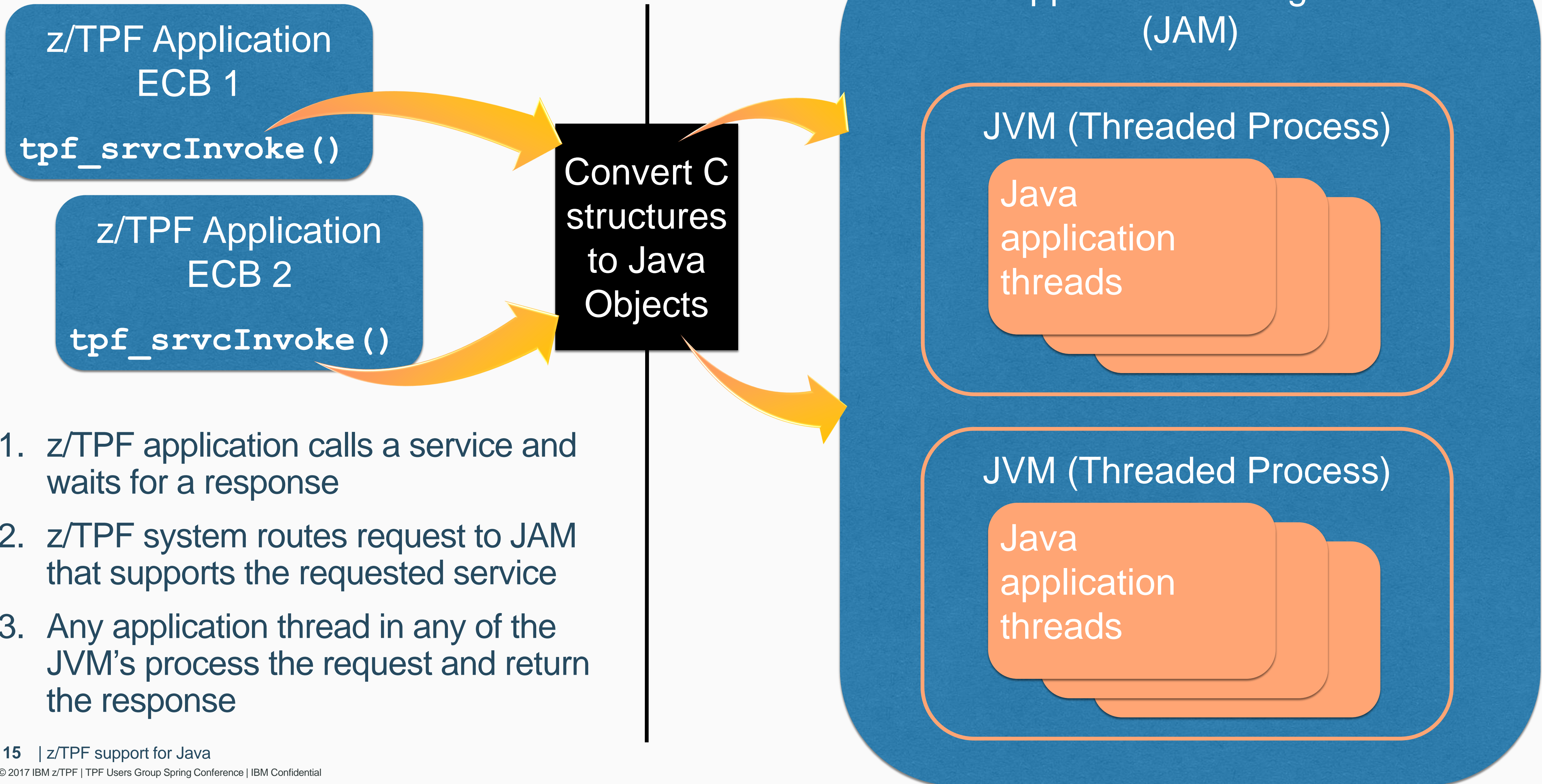
JVM (Threaded Process)

Java application threads

JVM (Threaded Process)

Java application threads

z/TPF calling Java – Calling services in a JAM



Managing JAMs

- ZJAMC command
 - Display, start, and stop a JAM
 - Recycle a JAM
 - Serially start new JVMs and stop old JVMs
 - New JVMs started with current ECB activation number
 - Nondisruptive - Services are available
- Automatic processing options in JAM descriptor
 - Start on cycle to NORM
 - Recycle on ECB activation number changes (ZOLDR ACT, DEAC, etc.)
- Scalability to match your workload
 - Update JAM descriptor to change number of JVMs or threads per JVM

z/TPF Application Manager for Java (JAM)

JVM (Threaded Process)

Java application threads



JVM (Threaded Process)

Java application threads



JVM (Threaded Process)

Java application threads



ZJAMC DISPLAY

```
AAES0008I 00 ==> ZJAMC DISPLAY NAME-*
CSMP0097I 11.08.26 CPU-B SS-BSS SSU-HPN IS-01
JAMC0008I 11.08.26 START OF ZJAMC SUMMARY DISPLAY
JAM NAME          #JVMS #THDS SHARED CLASS CACHE
ted2              3      11 N/A
fib               2       4 fib00000002
MYJAVAAPP        1       3 N/A
REGJAM1          1       3 N/A
resCluster       4       2 N/A
END OF DISPLAY+
```

STATE
DEPLOYED
ACTIVE
STARTING
UNDEPLOYED
ACTIVE

Displays state
of all JAMs

```
AAES0008I 00 ==> ZJAMC DISPLAY NAME-fib
JAM NAME          #JVMS #THDS SHARED CLASS CACHE
fib               2       4 fib00000002
```

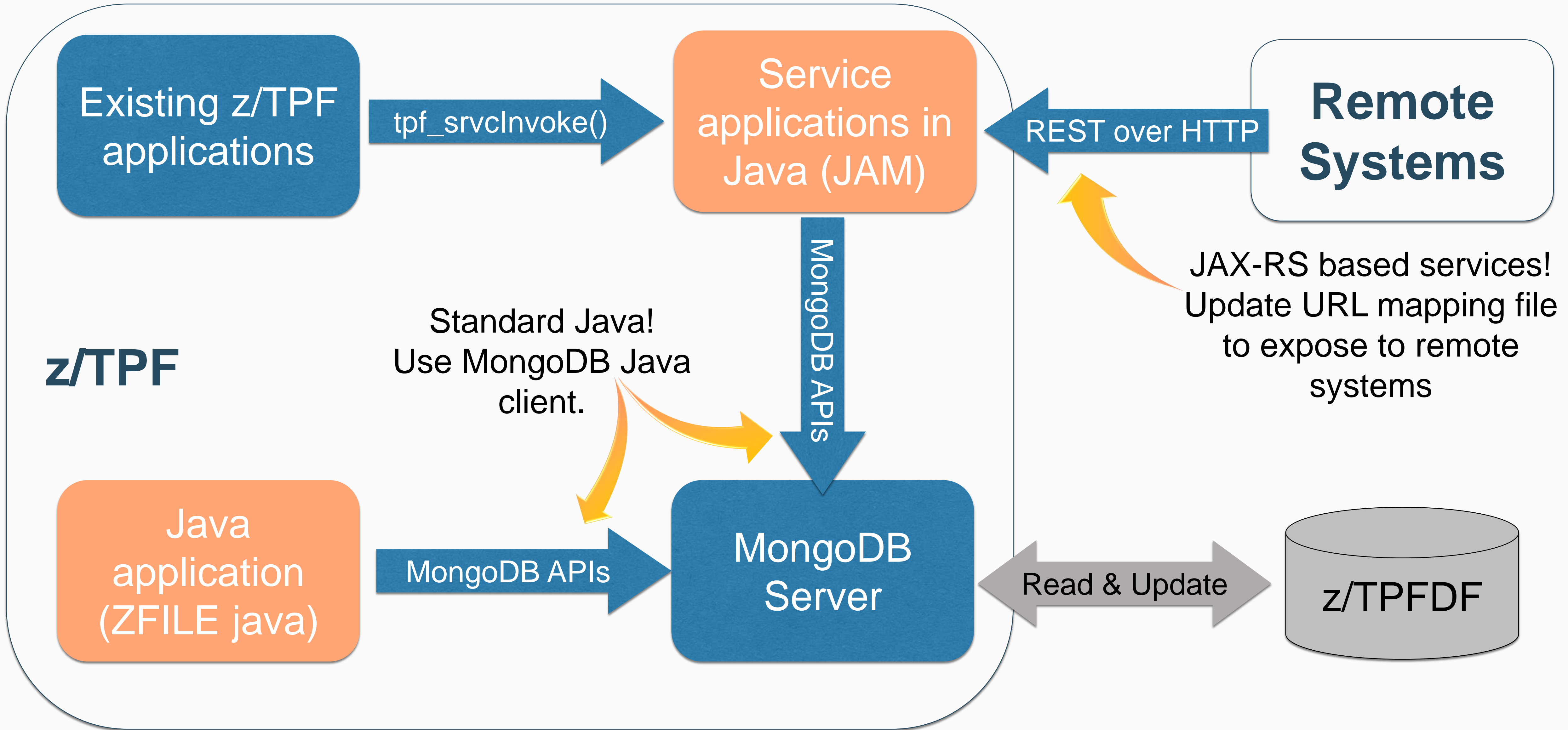
STATE
ACTIVE

```
DEPLOYMENT DESCRIPTOR FILE NAME
/sys/tpf_pfiles/tpf-fdes/fib.jam.xml
```

```
JVM PID          JVM STATE          JVM LOG FILE DIRECTORY
1084031110      ACTIVE          /tpfjam/fib/1084031110
1082523780      ACTIVE          /tpfjam/fib/1082523780
END OF DISPLAY+
```

Displays state of a JAM
and associated JVMs

What does the big picture look like?

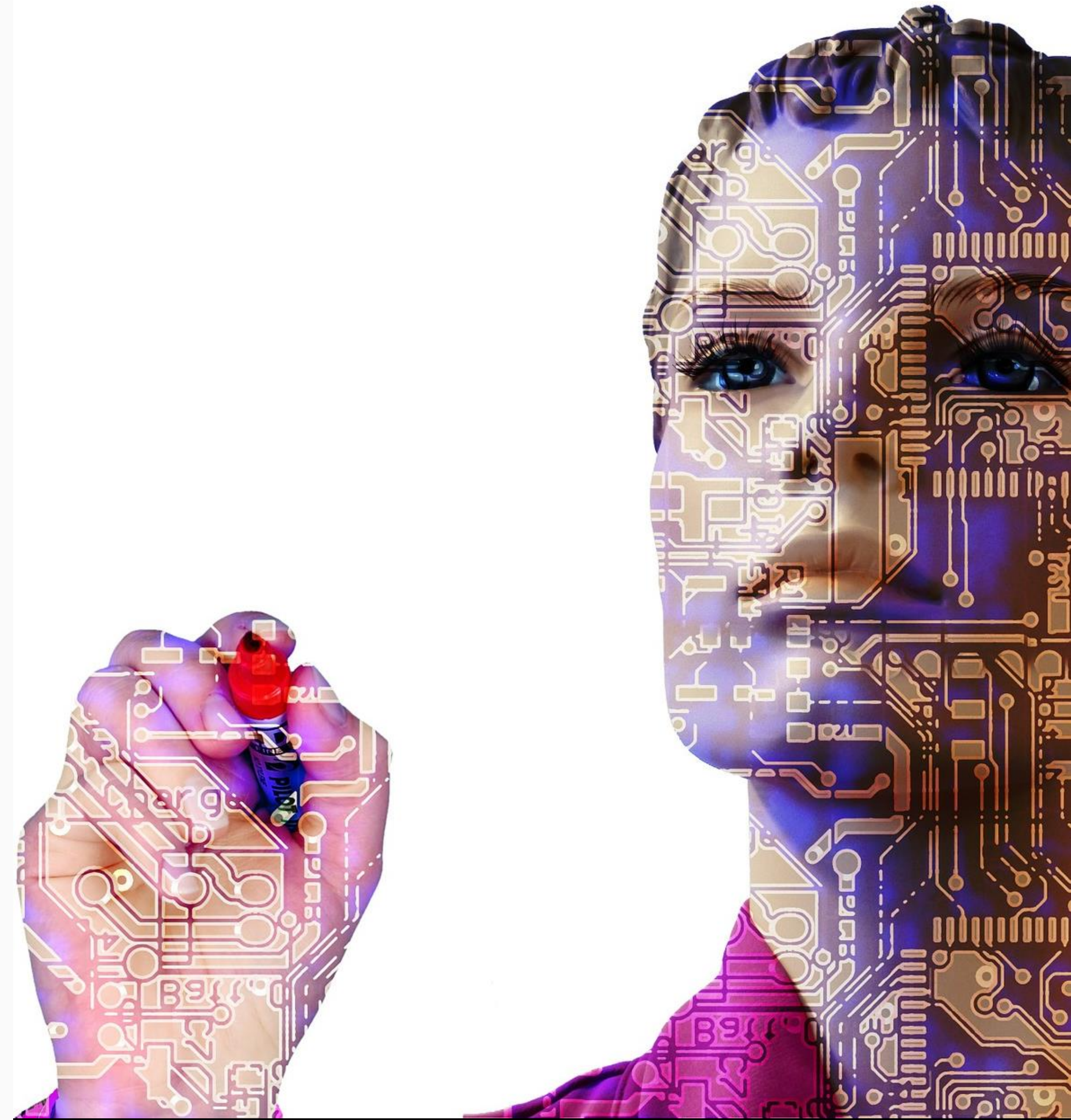


Security

- Load Java archive (JAR) files loaded using ZOLDR and ZTPLD
 - Traditional z/TPF loaders and controls
- z/TPF security manager for Java
 - Customized Java security manager
 - Control which classes can dynamically load Java code to your z/TPF system

Transformation Engine (TE) Eligible

- JVM and all Java archive (JAR) files executed by the JVM
- Includes both system and application JARs



Getting Started - Resources

[IBM Knowledge Center for z/TPF](#)

- [z/TPF support for Java](#)
- [Application Programming](#)
 - [z/TPF service applications in Java](#)

[z/TPF Blog](#) on IBM developerWorks

- [Using MongoDB in your Java Applications on z/TPF](#)
- [z/TPF support for Java](#)

[TPF Family Products: Driver downloads](#)

- [Java sample applications](#)
- [Service applications in Java](#)

YouTube video channels for [z/TPF](#) and [TPF Toolkit](#)

- [Creating JAM service artifacts for a native TPF application](#)



Disclaimer

Any reference to future plans are for planning purposes only.

IBM reserves the right to change those plans at its discretion.

Any reliance on such a disclosure is solely at your own risk.

IBM makes no commitment to provide additional information in the future.

Quarterly Refresh of Runtime

- Refreshes needed to ...
 - Maintain currency with Java 8 runtime
 - Provide fixes to security vulnerabilities
- Refresh is full replacement of runtime
 - Shared objects, JAR files, etc.
 - Download through z/TPF product download site
 - Should be loadable via ZOLDR

IBM 64-bit Runtime
Environment for z/TPF,
Java™ Technology Edition,
Version 8



APARs and Runtime Downloads

- Refresh always has new download
- Refresh may have a matching z/TPF APAR
 - APAR would contain z/TPF updates required for the runtime environment
 - No APAR will be issued if there are no required z/TPF updates
- Refresh announced through Blog entry and potentially security bulletin
- First refresh planned for 1st Half 2017

z/TPF Java GA

z/TPF APAR
PJ43892

Runtime
Environment
8.0.3.21

z/TPF Java Refresh 1

z/TPF APAR
PJ44655

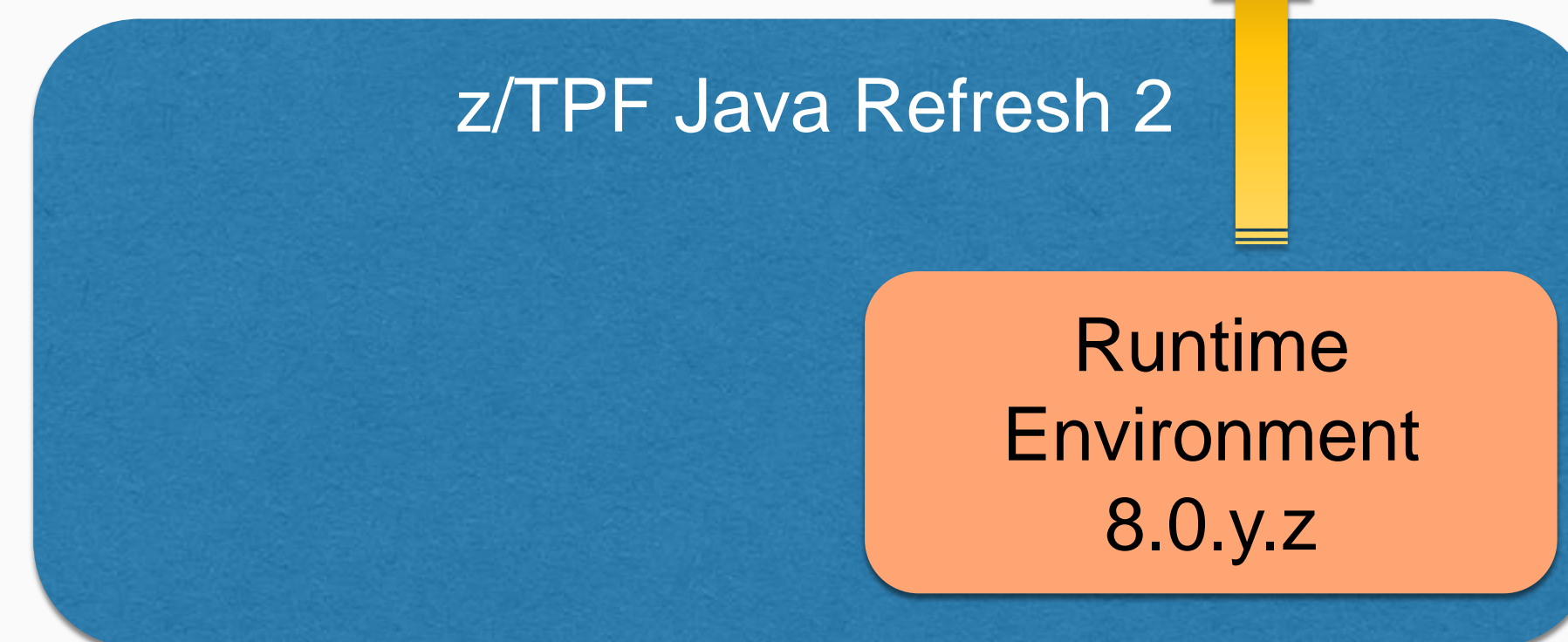
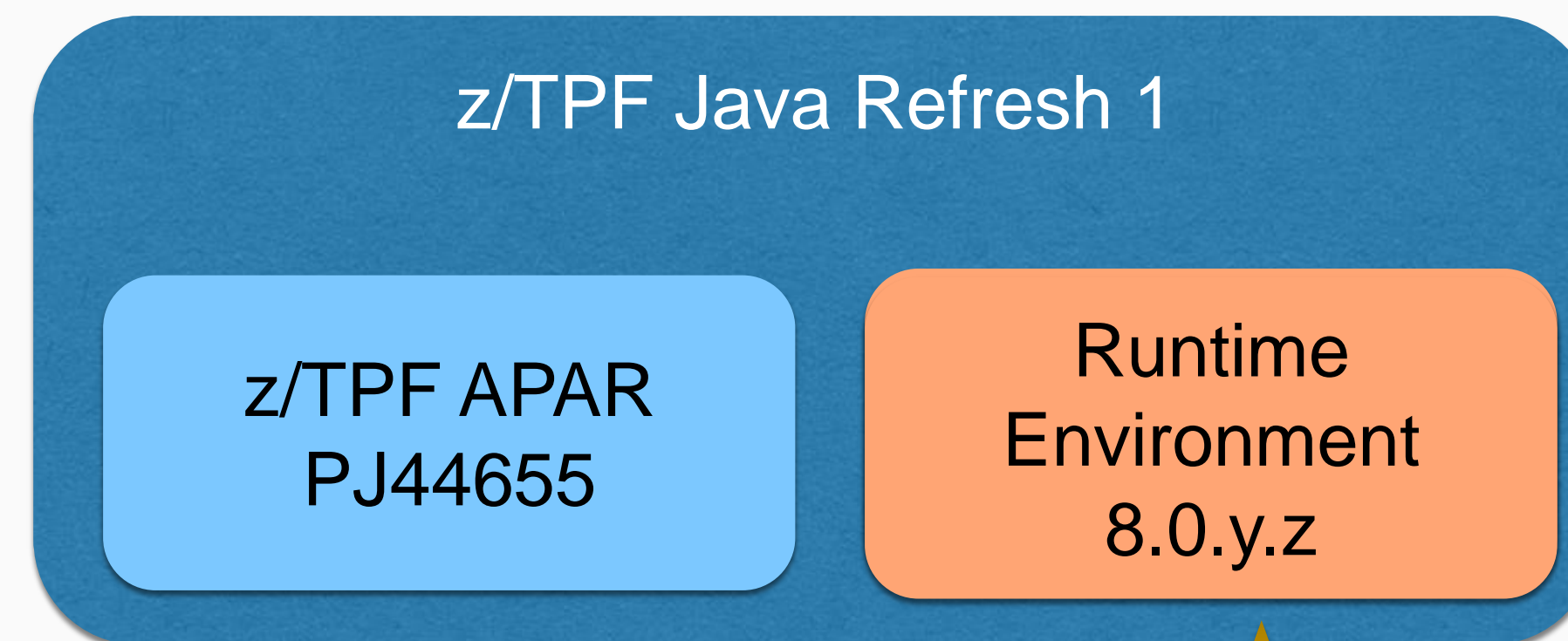
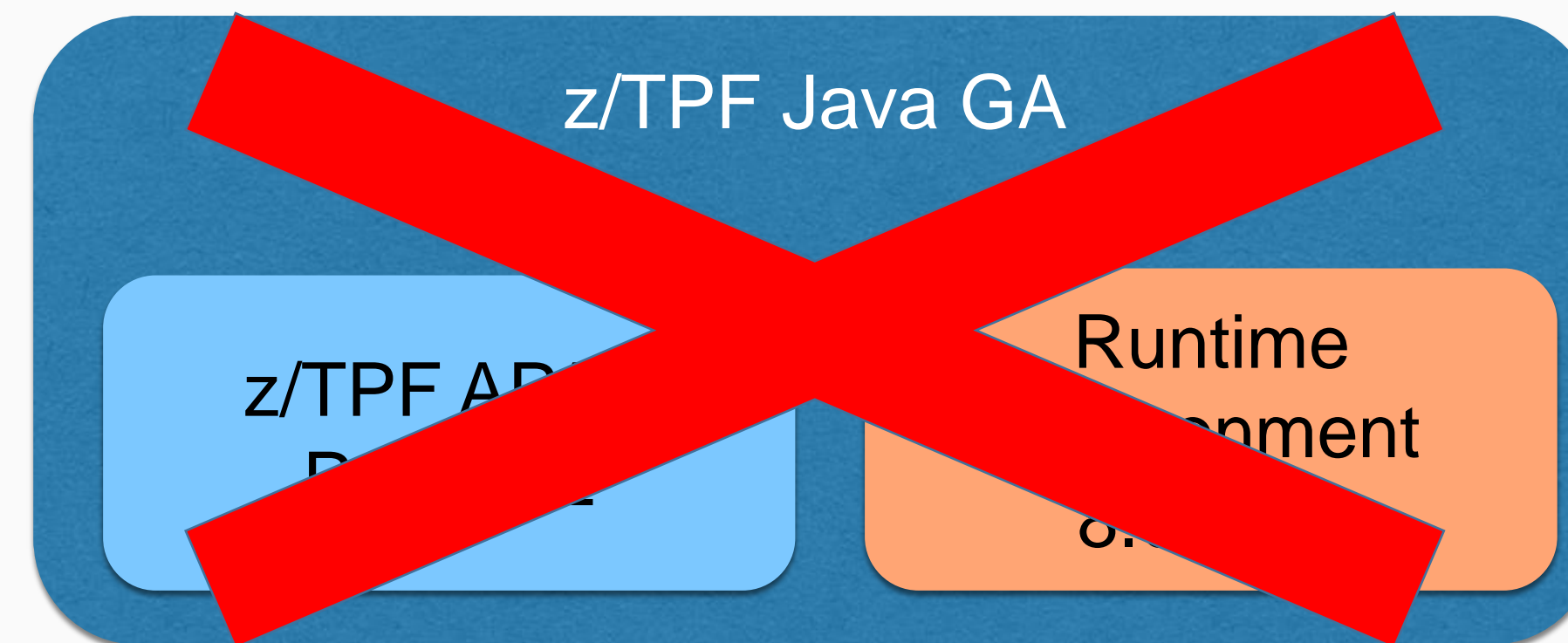
Runtime
Environment
8.0.w.x

z/TPF Java Refresh 2

Runtime
Environment
8.0.y.z

Availability of Runtime Downloads

- Download with no APAR is compatible with previous APAR/download pair
- Replaces download for most recent APAR/download pair
- Example:
 - 8.0.y.z replaces 8.0.w.x
 - 8.0.w.x is no longer available
- Downloads will be available for one year
 - In early 2018, the 8.0.3.21 download will be removed





THANK YOU

Questions or comments?

Chris Filachek

z/TPF and z/TPFDF Architecture & Development

© 2017 IBM z/TPF | TPF Users Group Spring Conference | IBM Confidential

IBM **z/TPF**
April 4, 2017

IBM, the IBM logo, ibm.com and Rational are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product names, service names, and logos might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “[Copyright and trademark information](#)” at www.ibm.com/legal/copytrade.shtml.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Notes

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

This presentation and the claims outlined in it were reviewed for compliance with US law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.