Large Scale Testing in an Agile World

Delivering enterprise quality in the IBM SDK, Java Technology Edition 7.0

Important Disclaimers

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.

WHILST EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

ALL PERFORMANCE DATA INCLUDED IN THIS PRESENTATION HAVE BEEN GATHERED IN A CONTROLLED ENVIRONMENT. YOUR OWN TEST RESULTS MAY VARY BASED ON HARDWARE, SOFTWARE OR INFRASTRUCTURE DIFFERENCES.

ALL DATA INCLUDED IN THIS PRESENTATION ARE MEANT TO BE USED ONLY AS A GUIDE.

IN ADDITION, THE INFORMATION CONTAINED IN THIS PRESENTATION IS BASED ON IBM’S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM, WITHOUT NOTICE.

IBM AND ITS AFFILIATED COMPANIES SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION.

NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, OR SHALL HAVE THE EFFECT OF:

- CREATING ANY WARRANT OR REPRESENTATION FROM IBM, ITS AFFILIATED COMPANIES OR ITS OR THEIR SUPPLIERS AND/OR LICENSORS

IBM Java Technology Edition Version 7.0

- General Availability 19th September 2011
  - Improved throughput
  - Faster startup
  - Smaller footprint
  - Introduces Balanced GC
  - Soft Real Time capabilities
  - Improved consumability
- Operating systems
  - AIX, Linux, z/OS, Windows, Solaris
- Platforms
  - Power, System Z, Intel, AMD, SPARC

Introduction to the speakers:

Iain Lewis

- 10 years experience developing and deploying Java SDKs
- Recent work focus:
  - Testing the IBM SDK using runtime byte-code modification
- Contact information:
  - ilewis@uk.ibm.com

Andrew Taylor

- 5 years experience in the IBM Java team
- Based in Hursley, UK
- Recent work focus:
  - System test infrastructure lead developer and maintainer.
  - Test developer for the Java 7 NIO.2 JSR (203)
- Contact information:
  - andytaylor@uk.ibm.com

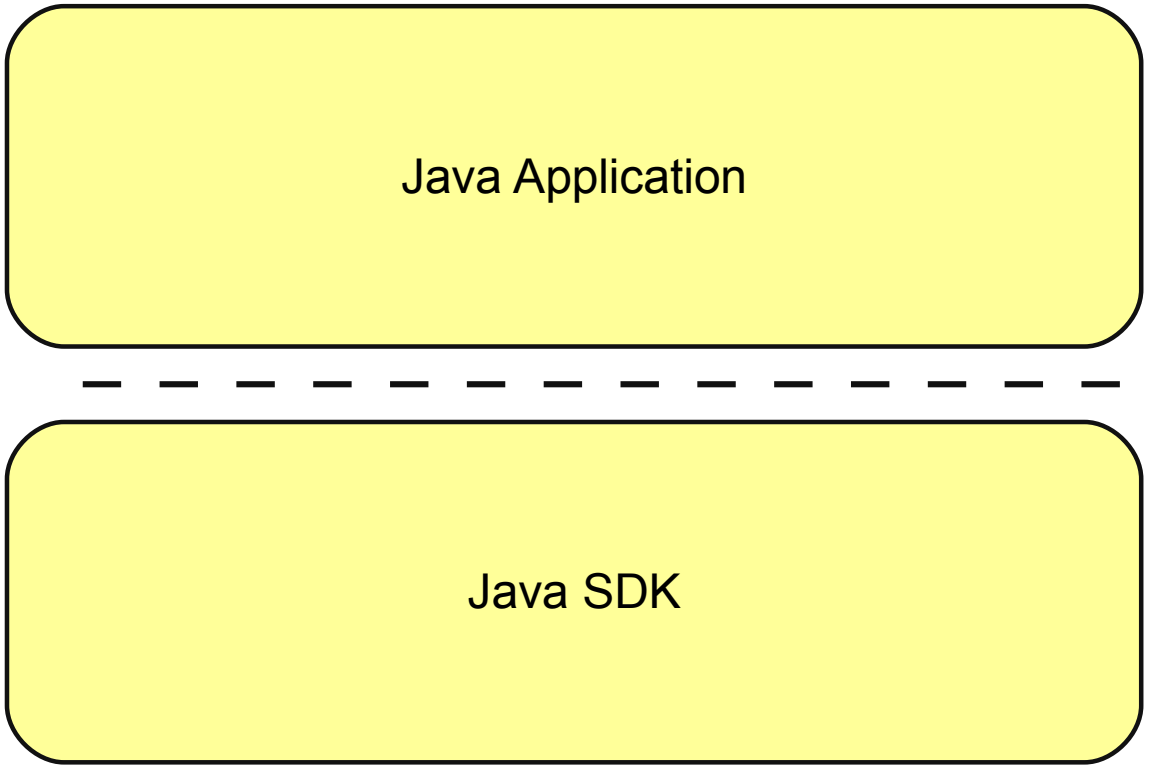
Agenda

- Iain
  - Problems/Challenges in testing a multi-site multi component product
    - Introduction to the SDK
    - What drives our testing (applications, platforms releases)
    - What testing we do
    - How we cover all scenarios
- Andy
  - Practical implementation of large scale multi-platform testing
    - How to implement an automatable/verifiable test
    - How to schedule tests
    - How to triage all those tests
    - Number of tests actually executed
    - How to report test status
- Wrap-up (Iain/Andy)

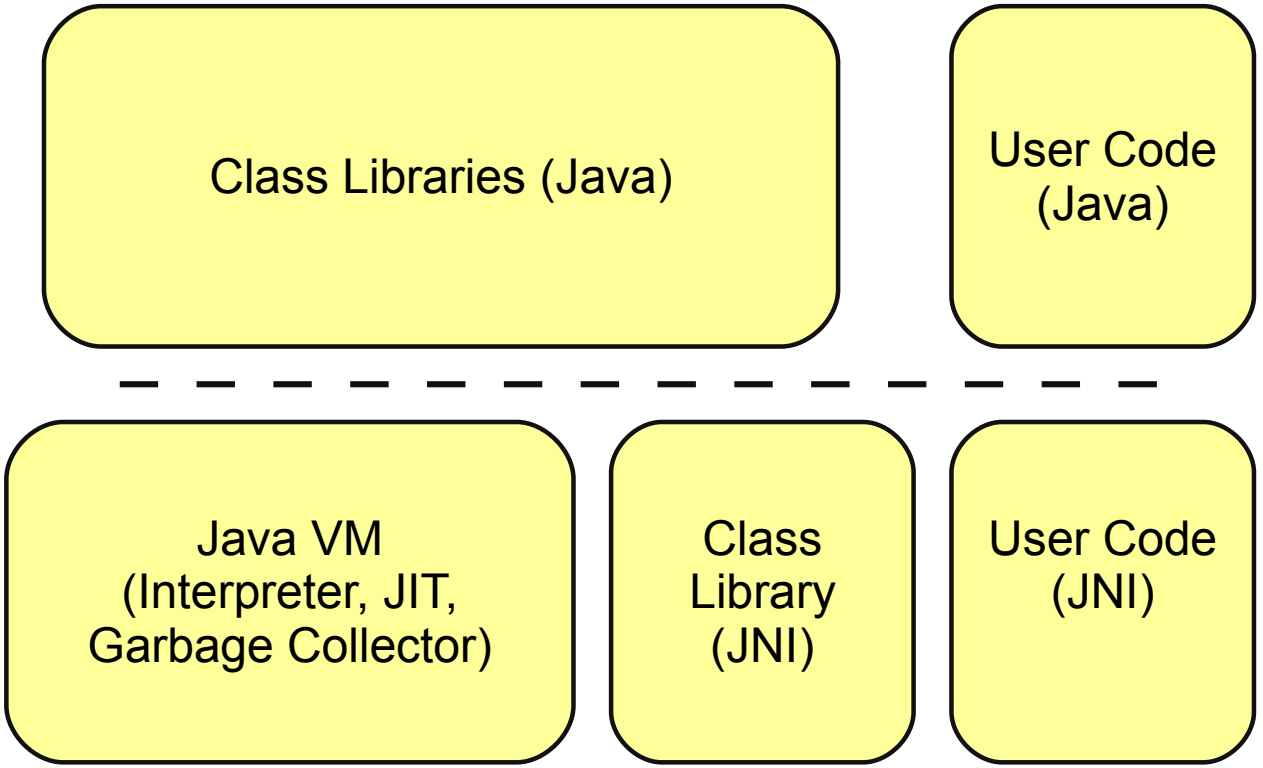
Why is Java important to IBM

- Java™ is critically important to IBM
  - Provides fundamental infrastructure to hundreds of products within IBM software portfolio
  - Delivers standard development environment
  - Delivered to Independent Software Vendors (ISVs) supporting IBM server platforms
  - Continuous investment in performance, reliability and serviceability

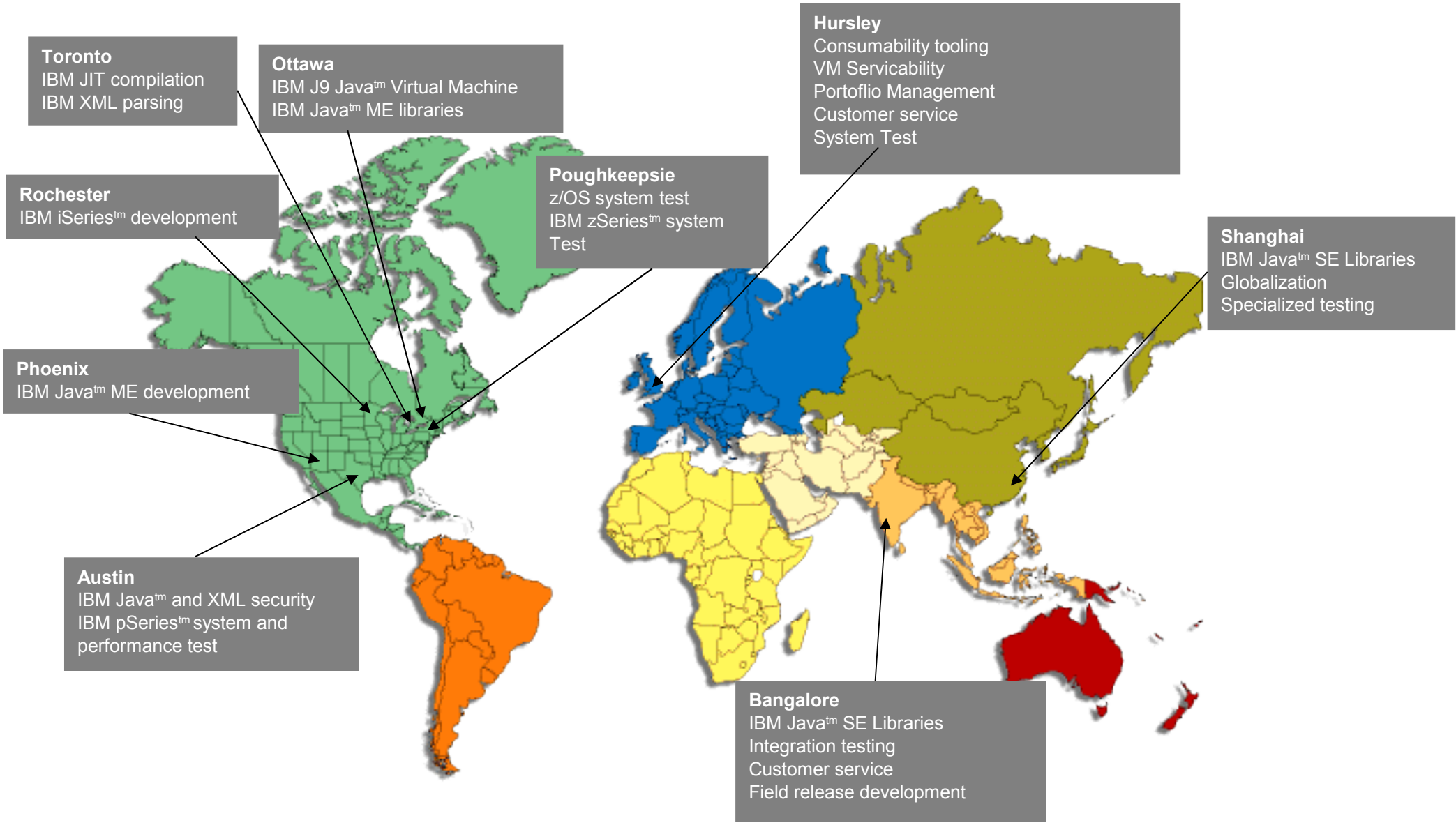
What is the IBM SDK for Java?



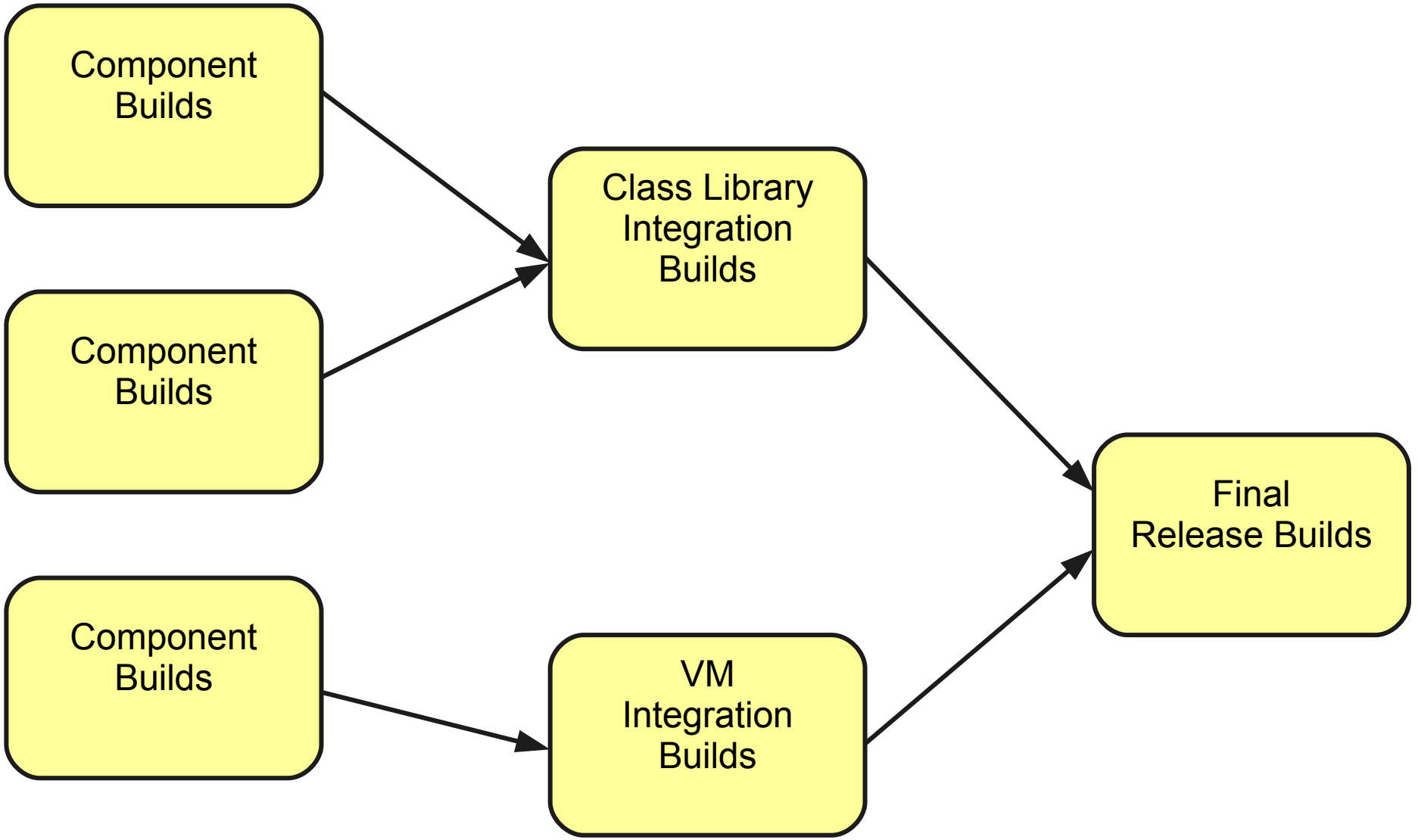
What is the IBM SDK for Java?



Java Technologies World Wide Organisation



How Does It all fit together?



Build Process Summary

- 6 independent component builds (at least)
- Final Release Build runs on average 2 times daily

Build Process Summary

- 6 independent component builds (at least)
- Final Release Build runs on average 2 times daily
- Builds on 20 platforms (OS + architecture combinations)
- Build on 5 supported Java releases
- Possibility of 200 builds a day...

All applications are the same aren't they?

All applications are the same aren't they?

- Heap size?

All applications are the same aren't they?

- Heap size from 32MB to 100s of GB
- Thread counts?

All applications are the same aren't they?

- Heap size from 32MB to 100s of GB
- Thread counts from 1 to thousands

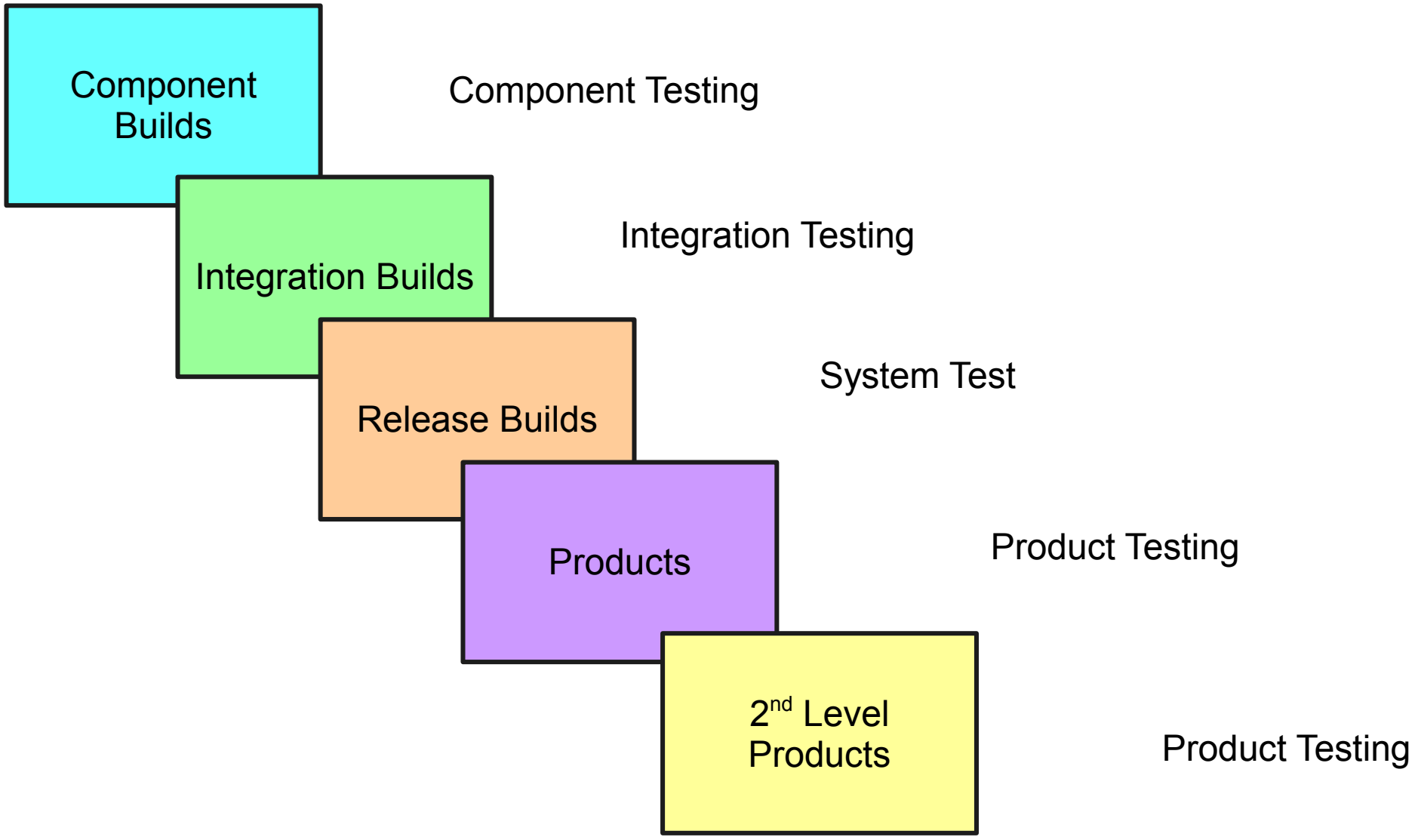
All applications are the same aren't they?

- Heap size from 32MB to 100s of GB
- Threads counts from 1 to thousands
- Running on 1 to 128+ cores
- 4 different GC policies
- 1000s command line option combinations

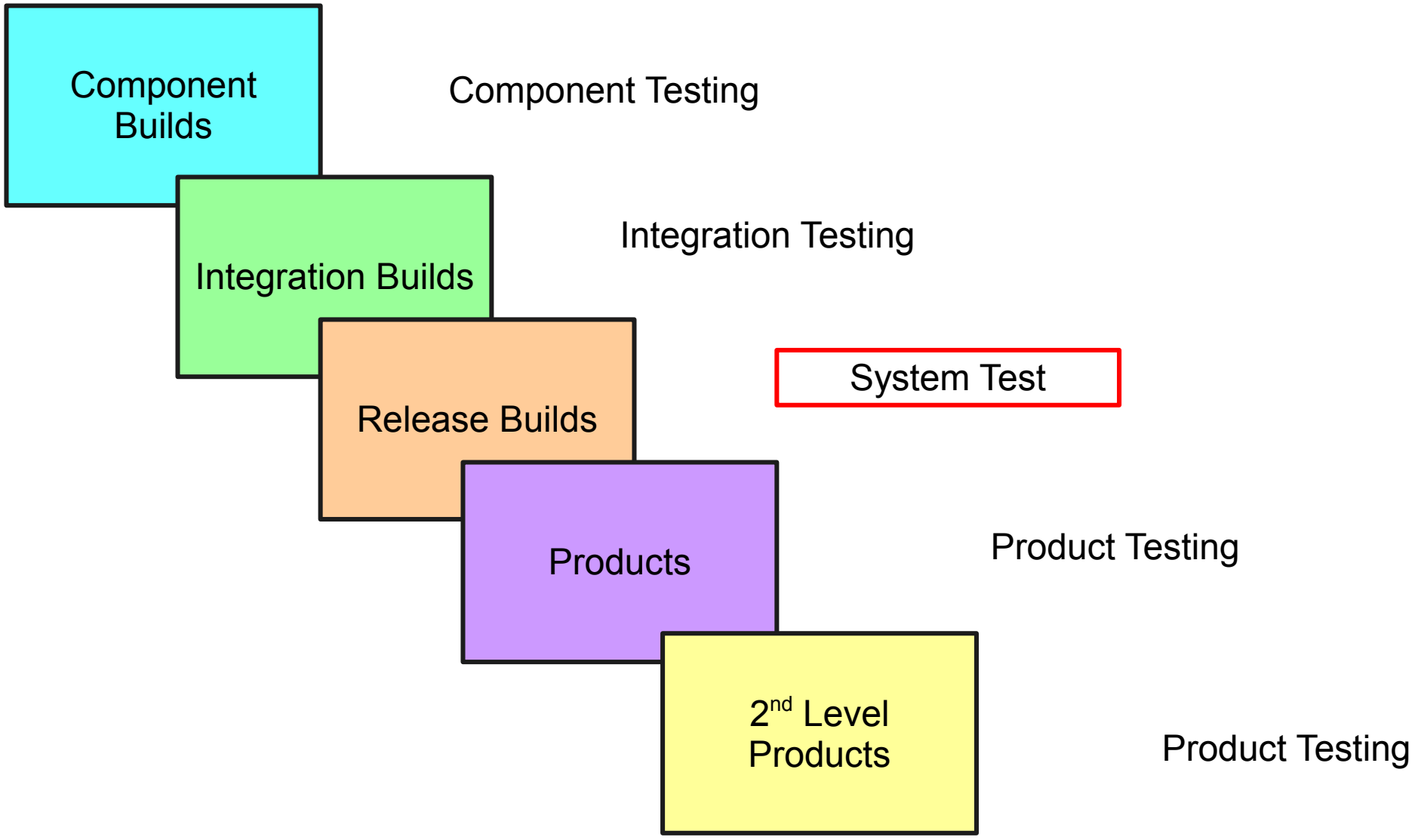
What Does our testing need to achieve?

- Provide the high quality base on which business critical applications are deployed
  - 100s of IBM Products
  - Support applications from business partners and Independent Software Vendor Applications
  - Customer applications (IBM SDK ships with both z/OS and AIX)
  - Open source applications
    - Hadoop (distributed processing framework)
    - Lucene (search engine library)
    - Tomcat/Geronimo (application servers)
    - Scripting languages (Jruby, Jython, Scala, Closure, Groovy...)
- Deliver coverage across the whole Java API
  - (Around 7000 classes java/* and javax/* classes in rt.jar alone)

What testing do we do?



What testing do we do?



What Testing do we do in the System Test Team

- Cover as many applications scenarios as possible:
  - 3rd Party test suites
    - Jruby, Scala, Jython, Eclipse, Derby (more being added..)
  - IBM Application testing
- Run as much Java code as possible
  - Apache Harmony testsuite (1000s)
  - Mauve test suite (100s)
  - 100s of other unit tests
- Internal testcases:
  - New tests written each release for
    - JSRs (new Java functionality)

      For Java7: Project Coin, NIO.2, JSR292 (java.lang.invoke)...
    - New functionality in the IBM SDK
    - Customer scenarios
  - Regression testing covering features of previous releases.
- Custom load test harness run much of the above in parallel at varying thread levels
- Cover as many usage patterns as possible
  - Run tests in > 200 test modes (combinations of various JVM options)
  - Run all tests across all platforms (where possible)
  - Run across varying machine configurations (OS, memory levels, core count)
- Test every build that is produced

What testing do we do?

Usage Patterns

Jython

IBM Application Testing

Scala

IBM Features

Project Coin

Apache Harmony

Derby

Customer Scenarios

Load Testing

New tests each release

Application Scenarios

JRuby

JSR292 (java.lang.invoke)

NIO.2

> 200 Test Modes

Eclipse

As much as possible!

Mauve

20 Platforms

Regression Testing

Iain's Summary

- We have a lot of builds to test
- We have a lot of tests to run
- How do we go about running all of that?
- Over to Andy..

The Need For Automation – An Overview

- Challenges
  - The need to provide continual verification of the quality of the Java SDK builds
  - Large matrix of possible combinations of command line options
  - Large number of individual testcases
  - The Java VM is a complex system, and defects are sometimes highly intermittent and hard to reproduce.
  - With teams located across the world, tests and their environments need to be portable and reproducible.
- Our Solutions
  - Continual running of tests 24/7 with tools and services to increase throughput and reduce manual effort required

Scale

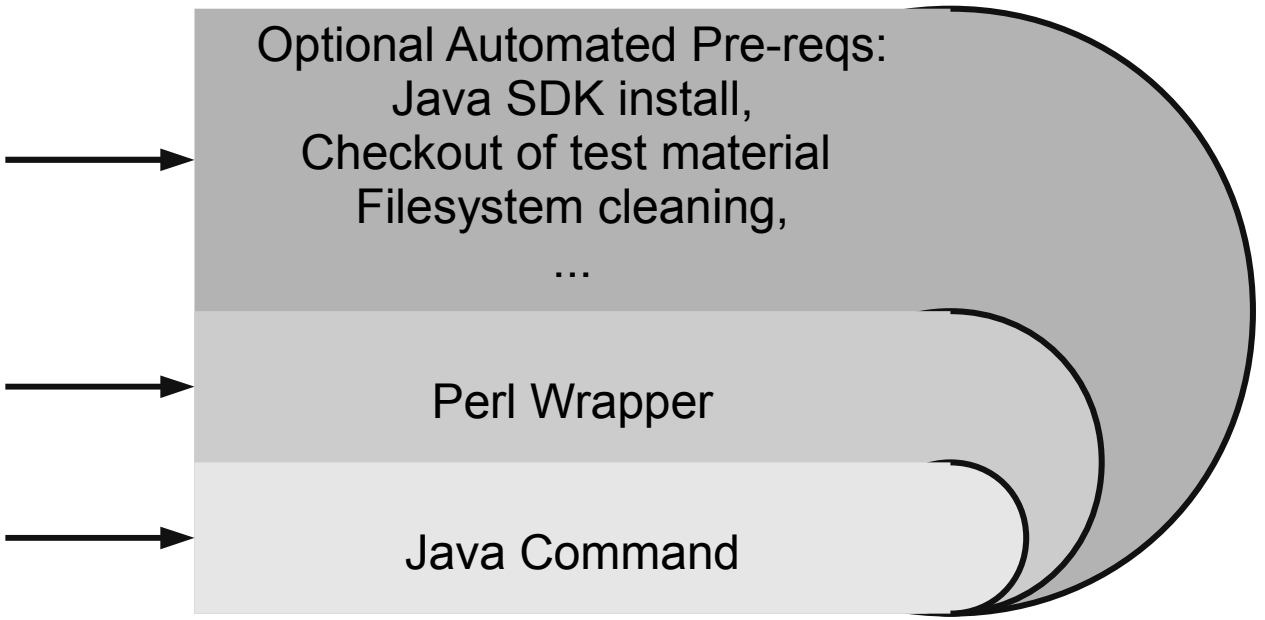
- 1000s of test machines
- Millions of tests executed monthly
- 10,000s of hours of tests run per day
- 1000s of packaged Java builds tested across 5 releases
- 20 platform environments tested

The Onion – Layers of Automation

How to build up functionality without tying yourself to one test environment

Individual testcases should be able to be initiated from any level.

This facilitates any mode of operation, from fully-automated regression testing, all the way down to customising the java command for defect debugging



Extra arguments can be added in at any level, passed down to the appropriate command lines.
Large, monolithic, test environments are not flexible enough to be passed between teams and do not lend themselves to a quick defect turnaround.

The Onion – Layers of Automation

The Java command

- At the heart of any test is the raw test command:
  - java <options> <class>
- The <options> are varied to change the JIT configuration, GC policy or other VM settings.
- Extra options may need to be added to trigger diagnostics whilst debugging a defect, or perhaps increase the likelihood of certain internal operations occurring
- A pure java command may be sufficient (and in some cases advantageous) when running a test manually, but in an automated system it is important to have something else checking the system under test...

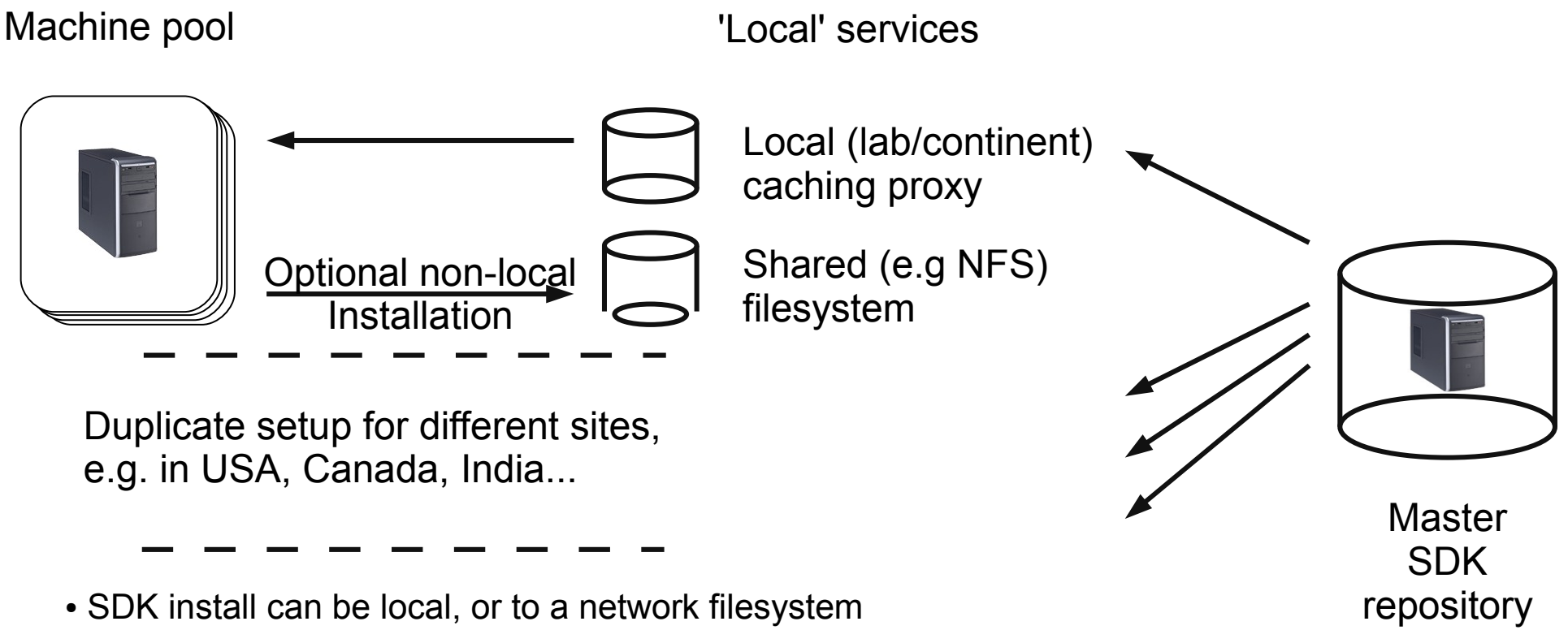
The Onion – Layers of Automation

The Perl Wrapper

- Tests are run on Linux, AIX, Windows and z/OS machines, requiring platform specific configuration (e.g. path separators)
- Watches the Java process for unexpected behaviour – e.g. creation of core dumps, hanging processes.
- Determines if the test has passed – additional checking of output files may be done once the Java command has completed.
- Provides the ability to upload diagnostic and output files to a central server if required (e.g. if the test fails)

The Onion – Layers of Automation

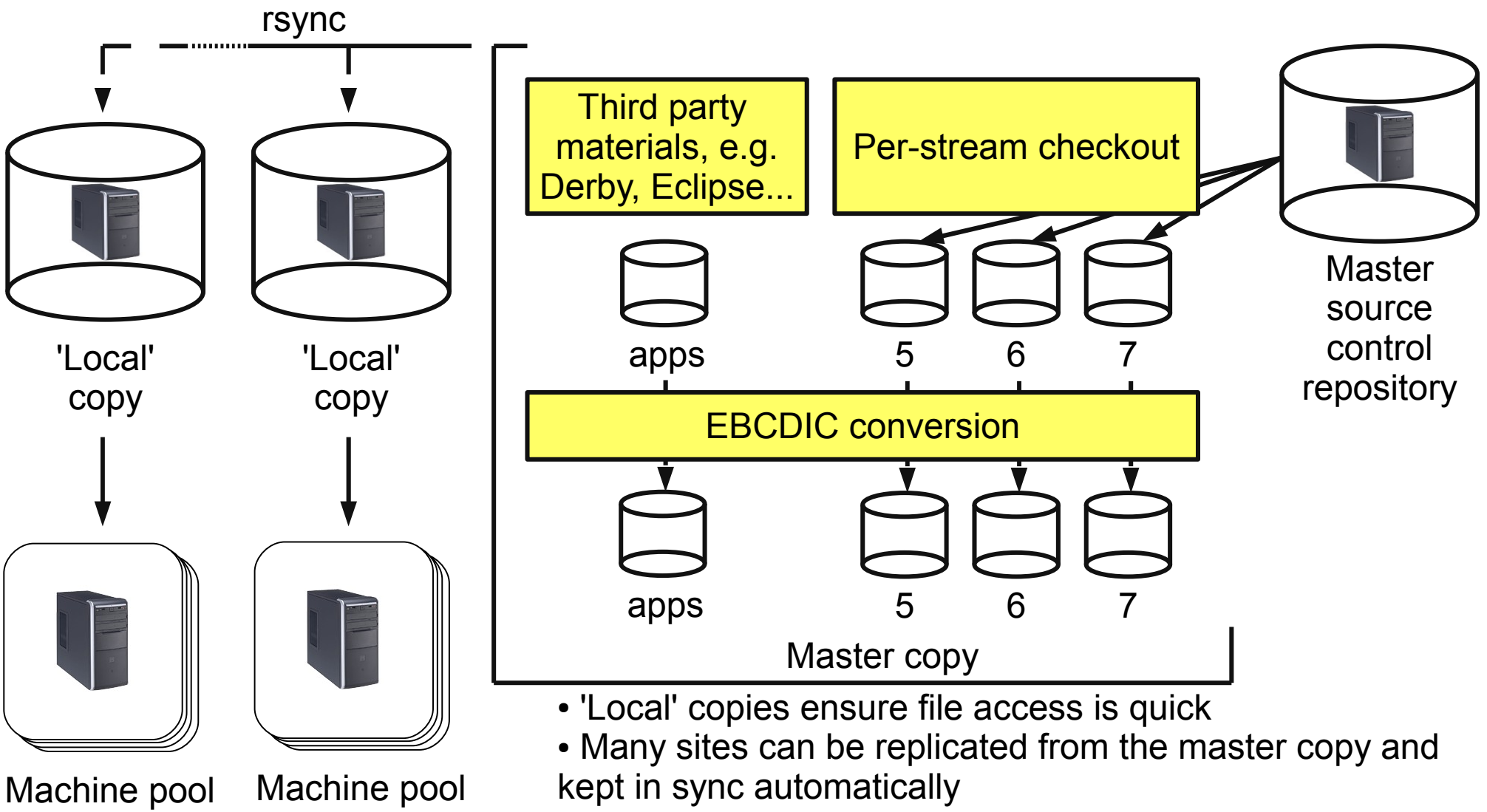
Prerequisites: Installing / Configuring a Java SDK for use on a test system



- SDK install can be local, or to a network filesystem
- 'Local' caching proxies eliminate costly transcontinental transfers
- The source of the SDK can be the master build location, local/regional proxy or on the filesystem.
- An asynchronous 'pre-stage' process can push builds to each proxy/NFS mount as they become available.

Test material availability

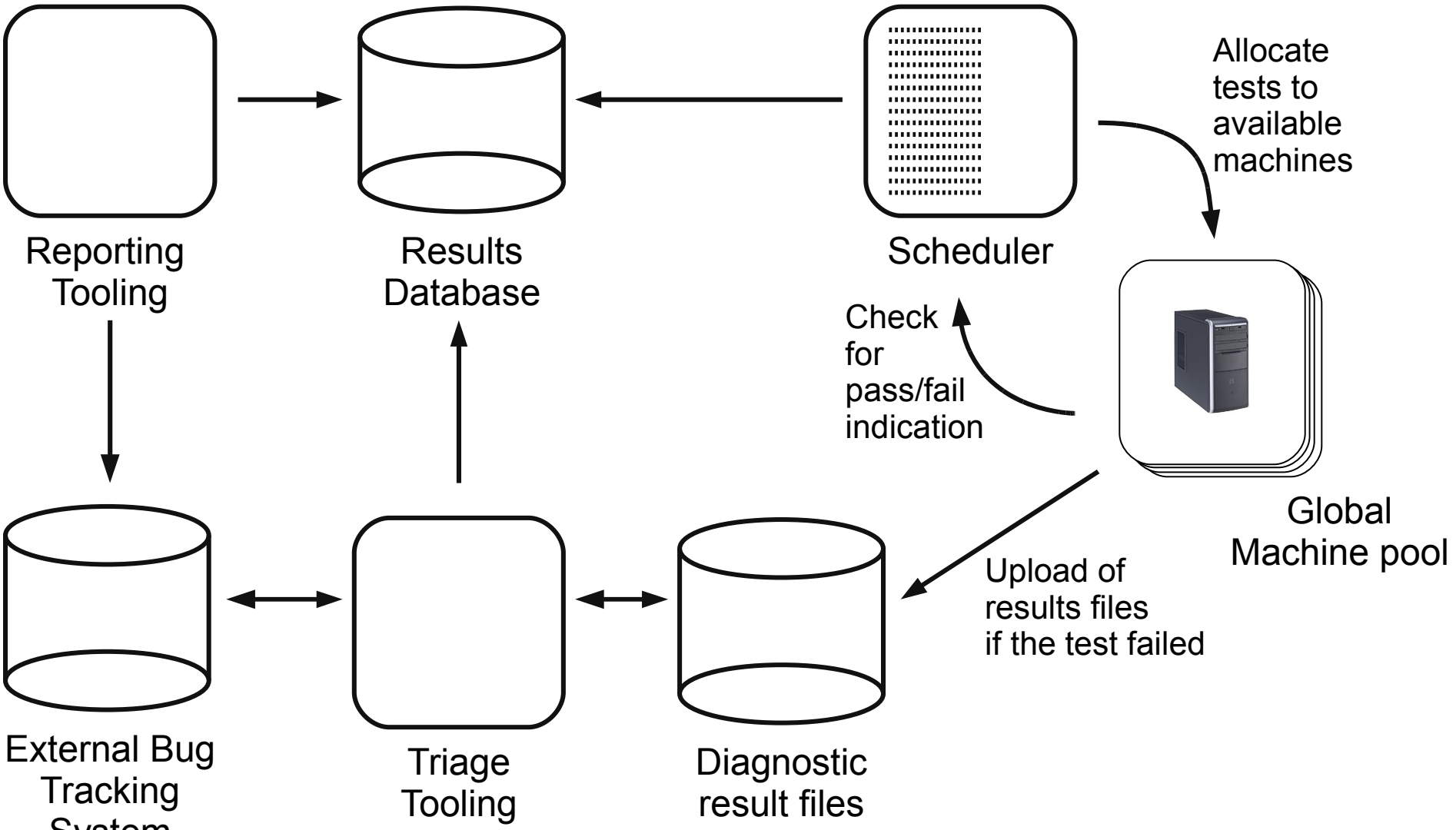
How to make sure everyone is running with the correct versions of tests



- 'Local' copies ensure file access is quick
- Many sites can be replicated from the master copy and kept in sync automatically
- Costly checkout / conversion steps done only once

Test Execution Automation

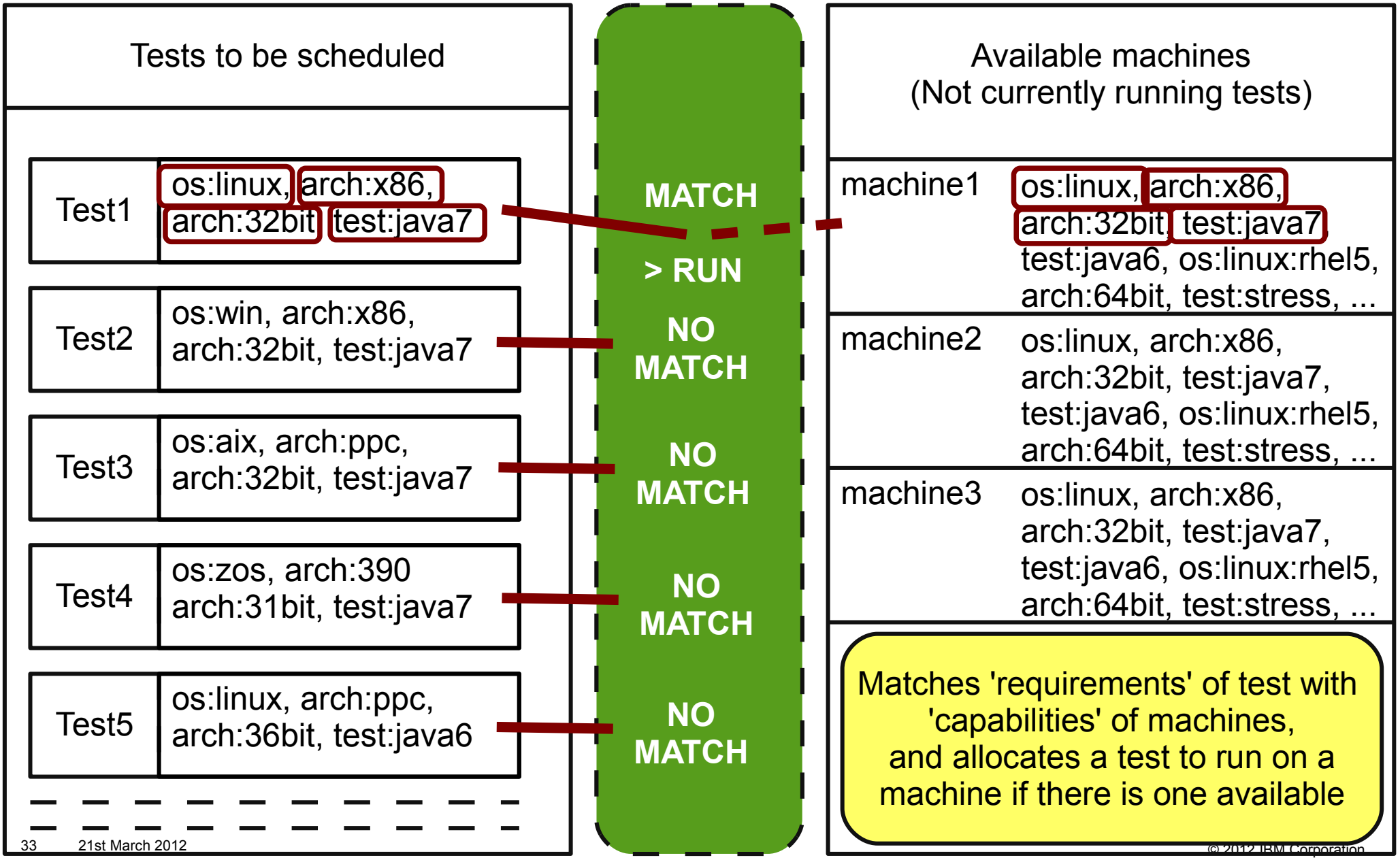
An overview



The Scheduler

- Runs 24/7
- Assigns tests to idle machines based on requirements of the test and the capabilities of the machines.
- New runs of tests are submitted daily as new Java SDKs are built.
- Tests are assigned to an available, applicable, machine - leading to coverage of all machine environments over time.

Test scheduling



Project Status Reporting

- For a release build all tests are run against it before it can be shipped – this is relatively straightforward to summarise.
- Custom in-house reporting tools can give an overall view a project by showing the most recent results for each test (possibly going back several days).

Issues with scaling up with automation

- Challenges:
  - More tests may fail leading to more manual triage / debugging effort required
  - Elimination of false-failures due to intermittent transient issues e.g. loss of network connectivity.
- Further improvements:
  - Continue effort to reduce triage and debugging time.
  - Provide tools to raise defects quickly, and mark test failures against already raised defects with minimal effort.

References

- **Get Products and Technologies:**
  - IBM Java Runtimes and SDKs:
    - https://www.ibm.com/developerworks/java/jdk/
  - IBM Monitoring and Diagnostic Tools for Java:
    - https://www.ibm.com/developerworks/java/jdk/tools/
- **Learn:**
  - IBM Java InfoCenter:
    - http://publib.boulder.ibm.com/infocenter/javasdk/v6r0/index.jsp
- **Discuss:**
  - IBM Java Runtimes and SDKs Forum:
    - http://www.ibm.com/developerworks/forums/forum.jspa?forumID=367&start=0

Copyright and Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., and registered in many jurisdictions worldwide.

Other product and service names might be trademarks of IBM or other companies.

A current list of IBM trademarks is available on the Web – see the IBM “Copyright and trademark information” page at URL: www.ibm.com/legal/copytrade.shtml