Please note

Copyright © 2018 by International Business Machines Corporation (IBM). No part of this document may be reproduced or transmitted in any form without written permission from IBM.

IBM’s statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM’s sole discretion.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.

The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including 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 results similar to those stated here.

IBM and the IBM logo are trademarks of International Business Machines Corporation, 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 at “Copyright and trademark information” at: www.ibm.com/legal/copytrade.shtml.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. ‘Eclipse’ is a trademark of Eclipse Foundation, Inc.
Other names and logos depicted in this presentation may be trademarks of their respective owners.
IBM has always derived runtimes from best of breed technology
- Optimized for middleware and customer scenarios
- Broad cross-platform support
- Deep integration with hardware and software innovation
- Strong standards engagement
- Delivered at the speed of the JCP & RI processes
Runtimes Evolution

“The Java Era” ME, SE, EE

Enterprise Java
✓ Performance
✓ Scale Up
✓ Tools

“The 1900’s”

Java dominates server
JavaScript Web App platform

2000

Java plus
“The Web”, JavaScript, Ajax

Speed, Agility, Cost Efficiency
Big Data, Analytics

2010

“Renaissance in Runtimes”
• Node.js, Go, Dart, Julia, D, R, Rust, Scala, Swift
• Ruby, Python, PHP

Today

“Cloud Platforms”
• Big Data, Analytics
• Developer Ecosystems
• Open Ecosystems

“Cloud Platforms”

“Rampant Innovation”
• “Get things Done” platforms
• Open Runtimes
• Open Stacks
• Open Communities
• Polyglot Developers
• Social Coding Communities
Open Technology: Eclipse OMR, Eclipse OpenJ9, and OpenJDK

Fundamental Language Technology Providers

Eclipse OMR (EPLv1 & AL2.0)
Components for current and future language ecosystems

Language communities

- Ruby?
- OMR
- static
- OMR
- dynamic
- OMR

IBM

Accelerated innovation and community health across multiple languages incl. OpenJDK and Eclipse OpenJ9 VM

Long term support, etc. focused on IBM customer value
AdoptOpenJDK: Open, Trusted, High-quality Binary Distributions

Latest build

Build jdk-9+181
Date: 27 September 2017
Timestamp: 201727090946

Select a platform

- Linux x64
- Windows x64
- macOS x64
- Linux s390x
- Linux ppc64le
- Linux aarch64
- AIX ppc64
Java in the Cloud Ecosystem

UI Engineering Team

Client Devices

Backend for Frontend

Micro-Services

Hosted Services

Backend Engineering Team

Java in the Cloud Ecosystem

Client Devices

Backend for Frontend

Micro-Services

Hosted Services
**Microclimate** an end to end development platform for the creation of **cloud native applications** and **microservices**.

https://microclimate-dev2ops.github.io/

1. **Containerized Development**
   Start to from scratch using lightweight containers that are easily reproducible to match your production environment locally or on IBM Cloud Private.

2. **Rapid Iteration**
   Lightning fast round-tripping through edit, build, and run for real-time performance insights, with an integrated IDE or editor of choice supporting Language Server Protocol.

3. **Intelligent Feedback**
   Best practices and immediate feedback of runtime metrics to help improve your application through your IDE.

4. **Diagnostic Services**
   Rapid problem determination at development time with diagnostic monitoring in the IDE.

5. **Integrated Dev-Ops Pipeline**
   Deploy into production fast with a preconfigured DevOps pipeline that can be tailored to your needs.
Microservices and cloud platforms have changed the role of the application container.

Developers need well-defined application-centric capabilities:

- Packaged by a continuous integration and continuous deployment (CI/CD) pipeline, running in lightweight virtualization containers.
- Wired to cloud platform capabilities for routing, management, scaling, and fault tolerance.
MICROPROFILE 1.3 - NEW ENTERPRISE CAPABILITIES FOR MICROSERVICES

- **Configuration 1.2**: Externalize configuration to improve portability
- **Open Tracing 1.0**: Allows services to easily participate in a distributed tracing environment
- **Metrics 1.1**: Common REST endpoints for monitoring service health
- **Open API 1.0**: Provides a Java API for the OpenAPI v3 specification that developers can use to expose their API documentation
- **Typesafe Rest Client 1.0**: Provides a type-safe approach for invoking RESTful services over HTTP
A lightweight open source server run time ideal for building Java microservices and cloud-native apps.
module example {
    requires helloworld;
}

var list = new ArrayList<String>();
var stream = list.stream();

% jshell
| Welcome to JShell -- Version 9
| For an introduction type: /help intro

jshell>
Java Community Process

Eclipse MicroProfile 2.0 (1H 2018?)

Get Involved!

- Google Groups
- MicroProfile Projects
- Bi-Weekly & Quarterly General Community Meetings
- Video Hangouts
Summary? It's Black and White!

- IBM has all the legacy responsibilities you'd expect from an org that has been in Java for so long.
  - We take long term support, security, serviceability, etc very seriously for existing code streams.

- Processes and procedures have to evolve quickly, drawing on experience to ensure we retain the good parts (well-defined specs, test suites, reference implementations, etc), while dropping the bad parts (hub-and-spoke IP flow, long ballots on waterfall processes, exclusivity, etc).

- Thought-leaders are collaborating in various locations.
  - This is as true for current technology as it is for new technology.
    - Indeed, Oracle "does not recommend or support use the JCP process for any future Java EE 8 functional enhancements".

- IBM is increasingly moving our Java innovation into community-empowering forums.
  - The community need places to discuss strategy, focus, and plans for a modern Java platform.
END
Innovation Happens Elsewhere

module example {
    requires helloworld;
}

var list = new ArrayList<String>();
var stream = list.stream();

% jshell
| Welcome to JShell -- Version 9
| For an introduction type: /help intro

jshell>
Microclimate: seamless cloud development experience

Rapid on-boarding

- Web-based IDE supporting full live sandbox lifecycle, or BYOE
- Instant monitoring and feedback
- Consistent tools from dev to ops

Powered by Language Server Protocol (LSP)