Java™
Community Process
Executive Committee
Member

IBM

Please note

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

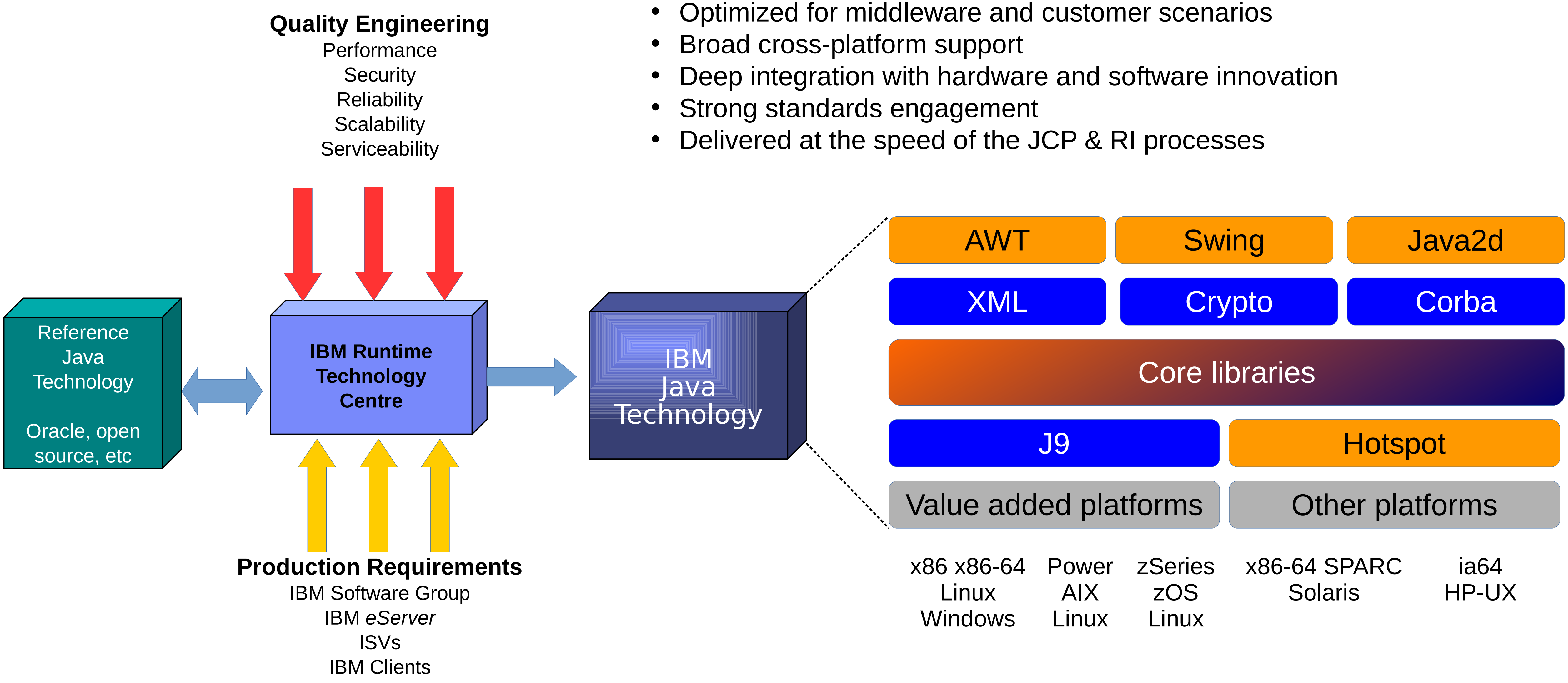
The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.

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

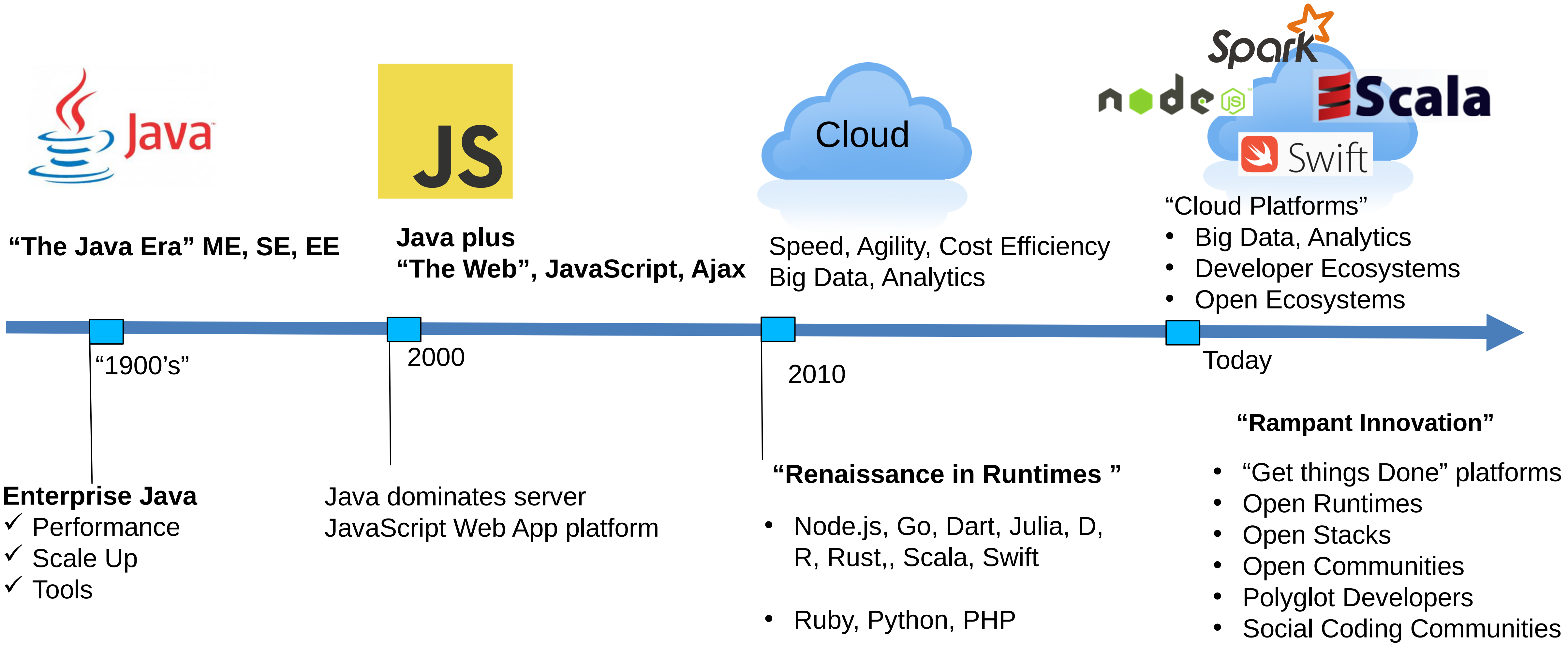
Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user’s job stream, 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.

Purveyors of Fine Java^{tm} Runtimes Since 1995

- 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

“1900’s”

Enterprise Java

- ✓ Performance
- ✓ Scale Up
- ✓ Tools

Java plus
“The Web”, JavaScript, Ajax

2000

Java dominates server
JavaScript Web App platform

Cloud

Speed, Agility, Cost Efficiency
Big Data, Analytics

2010

“Renaissance in Runtimes ”

- Node.js, Go, Dart, Julia, D, R, Rust,, Scala, Swift
- Ruby, Python, PHP

“Cloud Platforms”

- Big Data, Analytics
- Developer Ecosystems
- Open Ecosystems

Today

“Rampant Innovation”

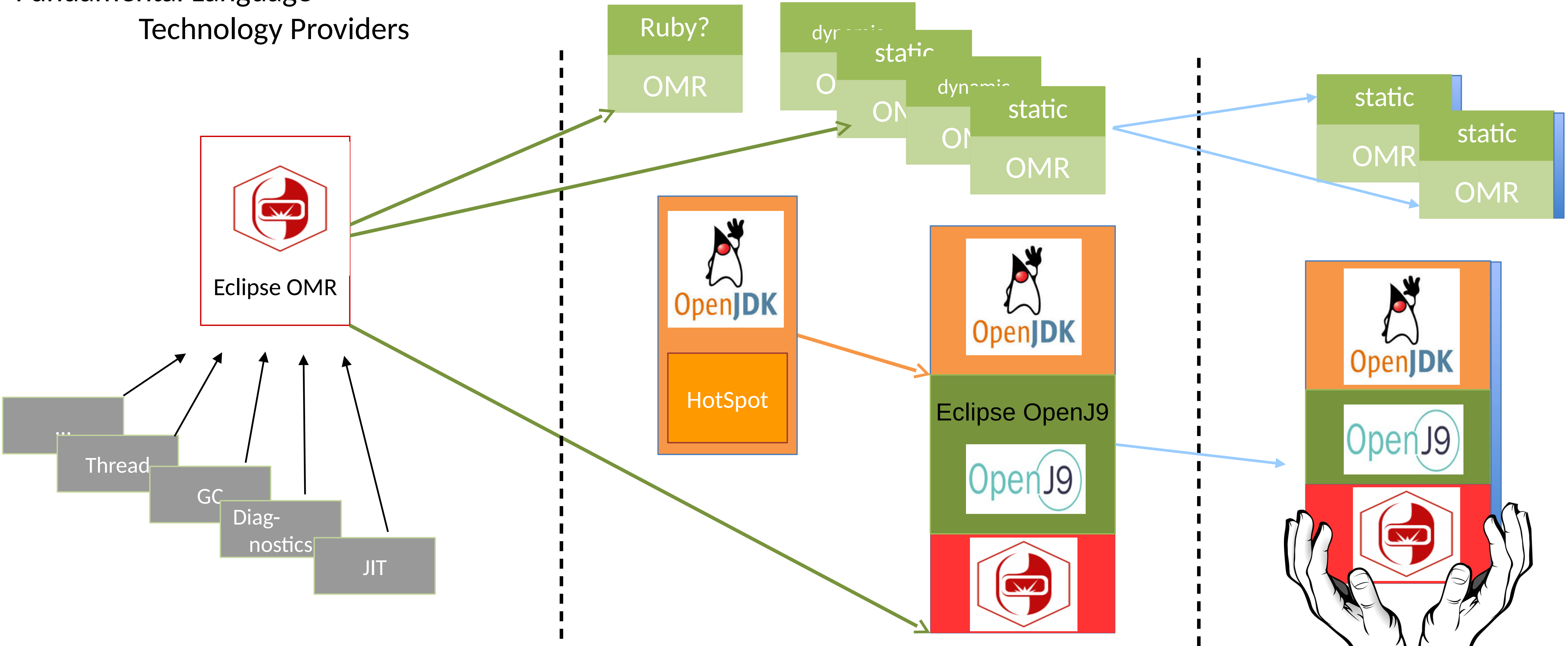
- “Get things Done” platforms
- Open Runtimes
- Open Stacks
- Open Communities
- Polyglot Developers
- Social Coding Communities

Open Technology: Eclipse OMR, Eclipse OpenJ9, and OpenJDK

Fundamental Language Technology Providers

Language communities

IBM

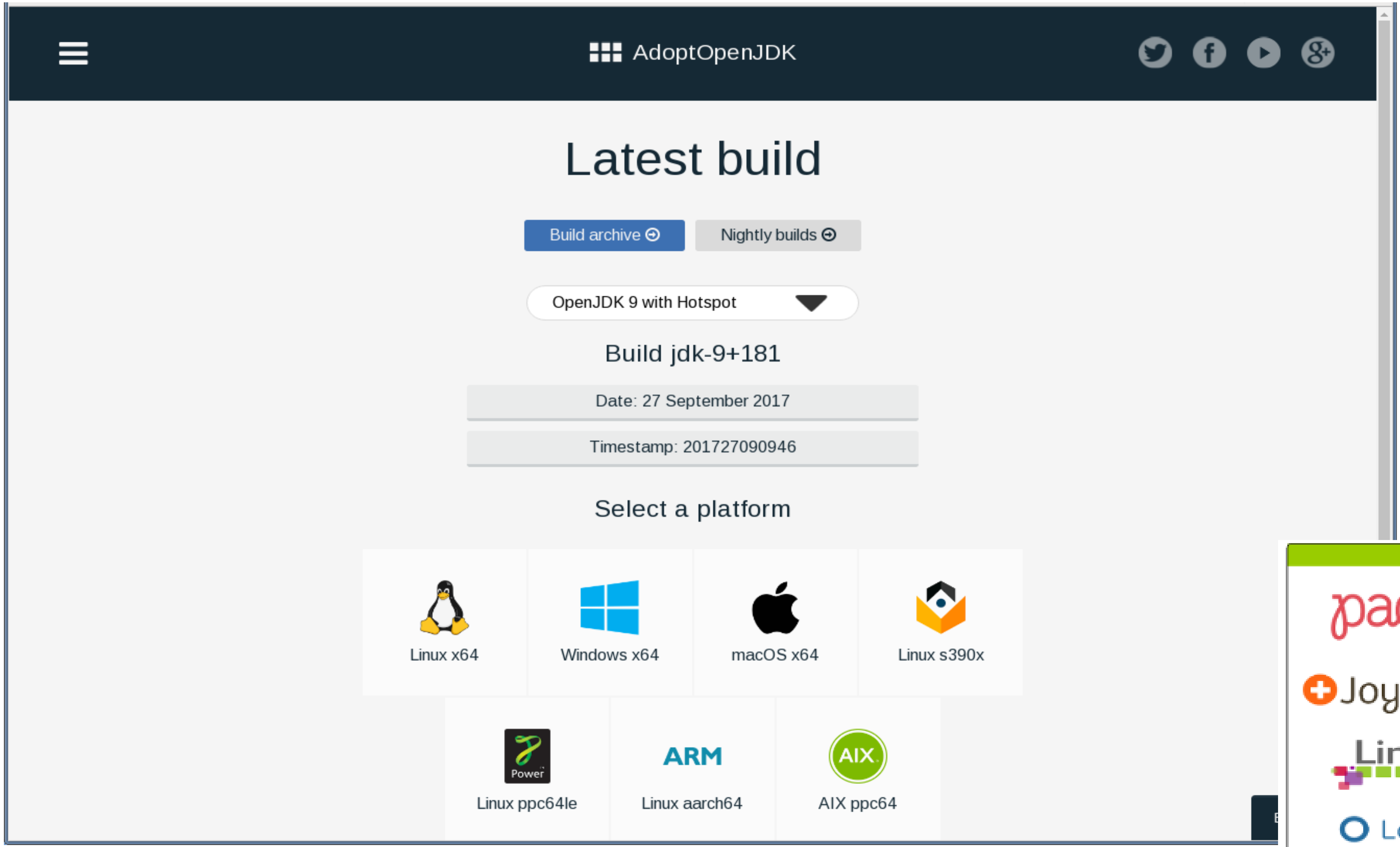


Eclipse OMR (EPLv1 & AL2.0)
Components for current and future language ecosystems

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

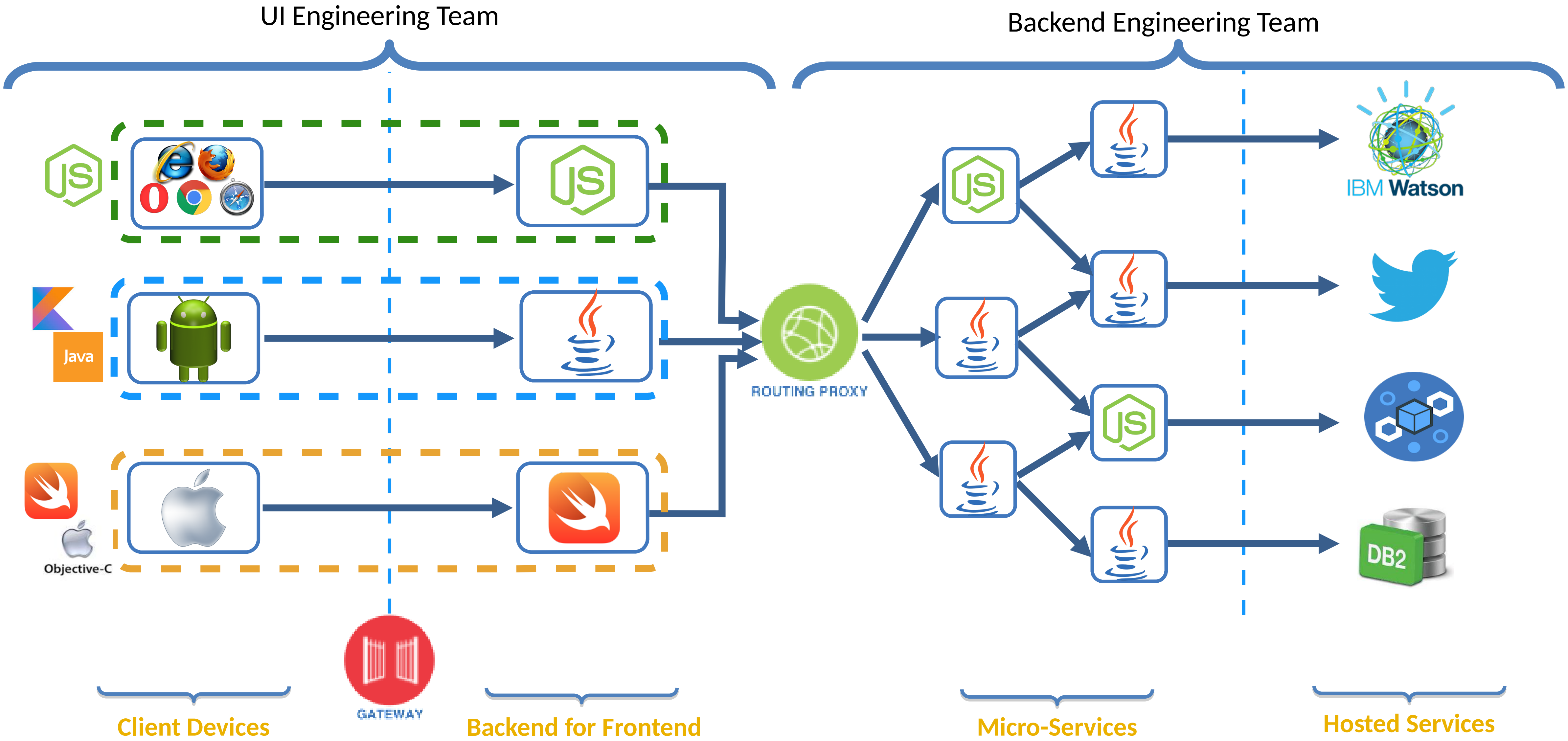
Timestamp: 201727090946

Select a platform

SPONSORS

packet LJC IBM
Joyent Scaleway CLOUDFLARE
Linaro MacStadium GitHub
Localize slack Installers by BitRock
digicert macincloud

Java in the Cloud Ecosystem



Integrating Runtimes with Cloud Analytics and Developer Tooling

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

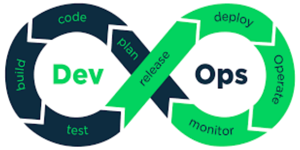
Best practices and immediate feedback of runtime metrics to help improve your application through your IDE

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

Java EE™

THENEWSTACK
THE DOCKER & CONTAINER ECOSYSTEM
EDITED & CURATED BY ALEX WILLIAMS
THENEWSTACK
APPLICATIONS & MICROSERVICES WITH DOCKER & CONTAINERS
EDITED & CURATED BY ALEX WILLIAMS
COMING SOON
AUTOMATION & ORCHESTRATION WITH DOCKER & CONTAINERS
COMING SOON
NETWORKING, SECURITY & STORAGE WITH DOCKER & CONTAINERS
COMING SOON
MONITORING & MANAGEMENT WITH DOCKER & CONTAINERS

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

Cloud Native Java Stack

eclipse

MICROPROFILE™
OPTIMIZING ENTERPRISE JAVA

Java EE

Eclipse MicroProfile 1.3:
> https://projects.eclipse.org/projects/technology.microprofile

Eclipse Enterprise for Java:
> https://projects.eclipse.org/projects/ee4j

IBM LJC redhat Tomitribe

payara SOUJava sociedade de usuários java hazelcast FUJITSU

kumuluzEE SMARTBEAR ORACLE HAMMOCK

MICROPROFILE 1.3 - NEW ENTERPRISE CAPABILITIES FOR MICROSERVICES

Configuration 1.2
Externalize configuration to improve portability

Open API 1.0
provides a Java API for the OpenAPI v3 specification that developers can use to expose their API documentation

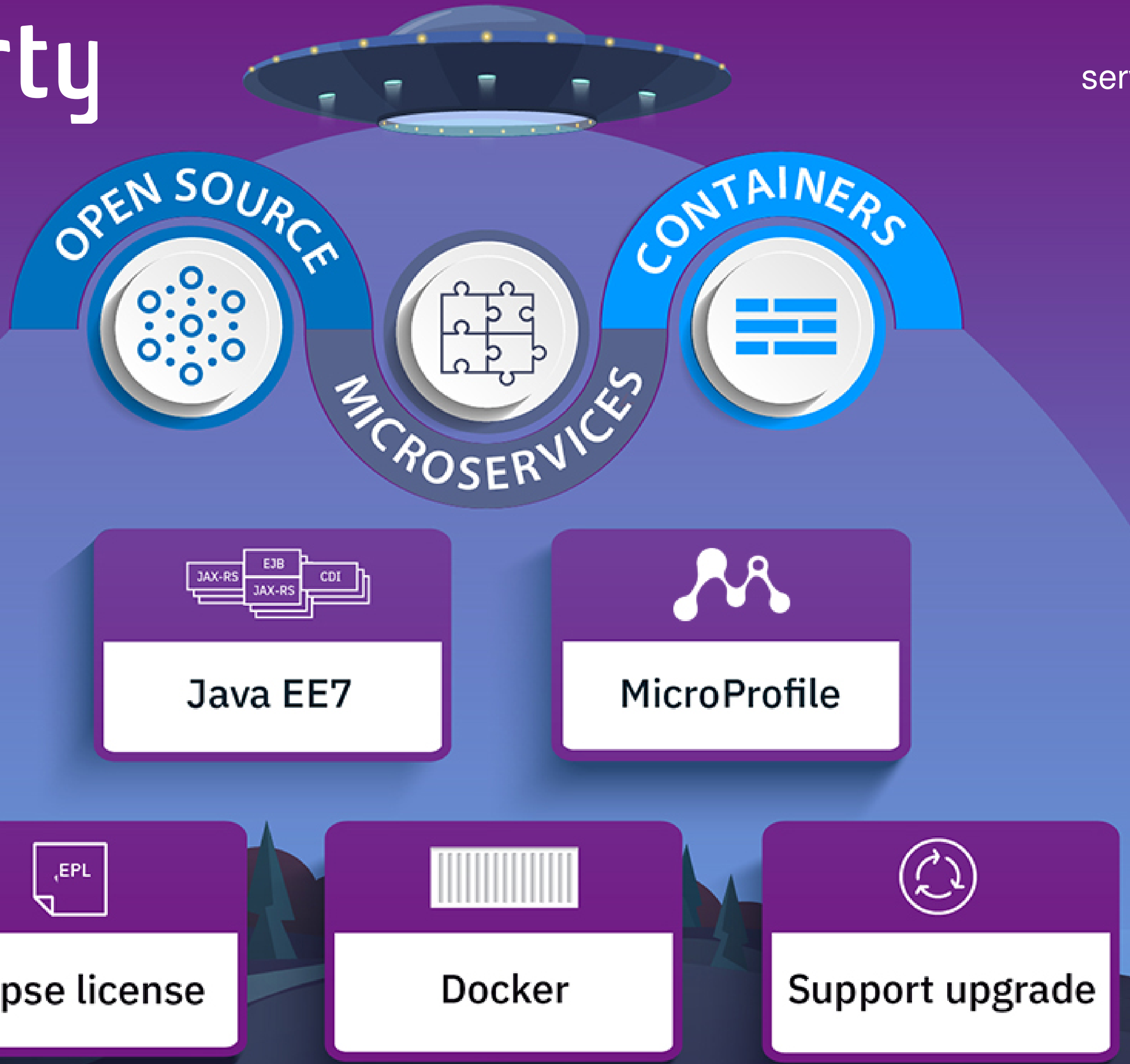
Open Tracing 1.0
Allows services to easily participate in a distributed tracing environment

Typesafe Rest Client 1.0
Provides a type-safe approach for invoking RESTful services over HTTP.

Metrics 1.1
Common REST endpoints for monitoring service health

Open Liberty

A lightweight open source
server run time ideal for building
Java microservices and
cloud-native apps



OPEN SOURCE

MICROSERVICES

CONTAINERS

Java EE7

MicroProfile

Eclipse license

Docker

Support upgrade

Innovation Happens Elsewhere

OpenJDK

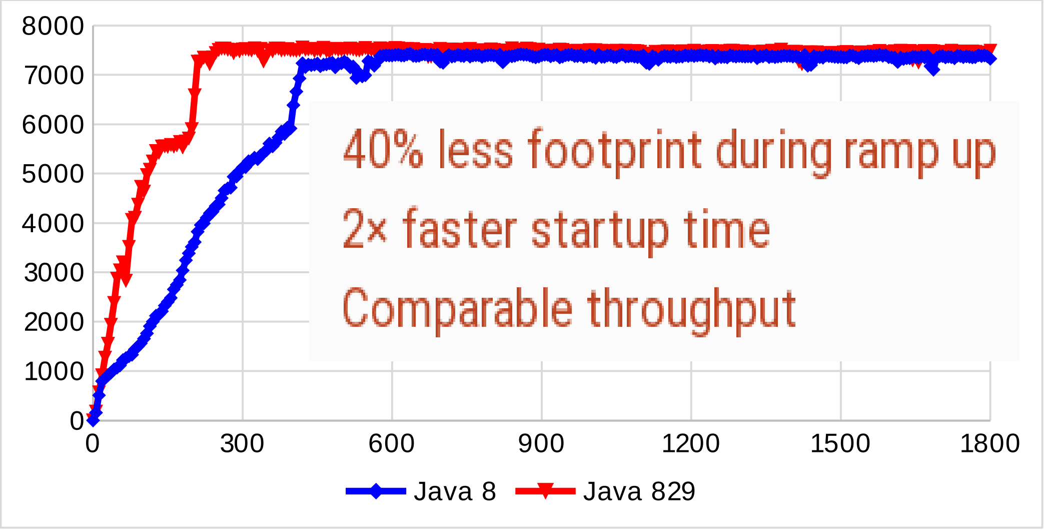
```
module example {
  requires helloworld;
}
```

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

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

jshell>
```

OpenJ9

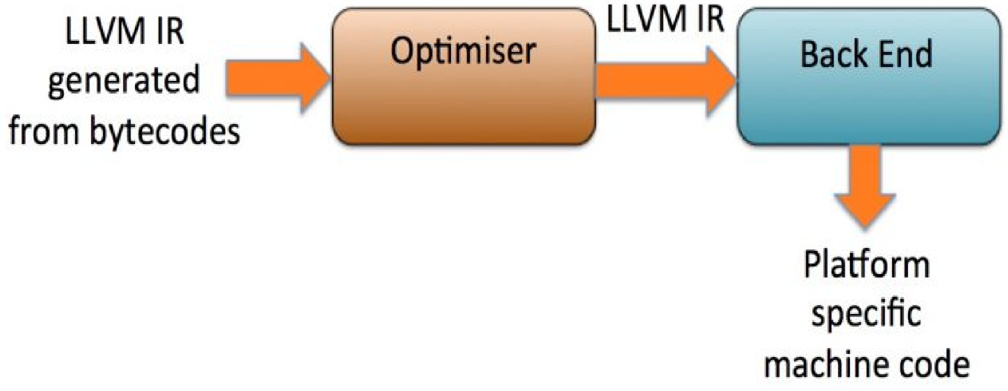


40% less footprint during ramp up

2× faster startup time

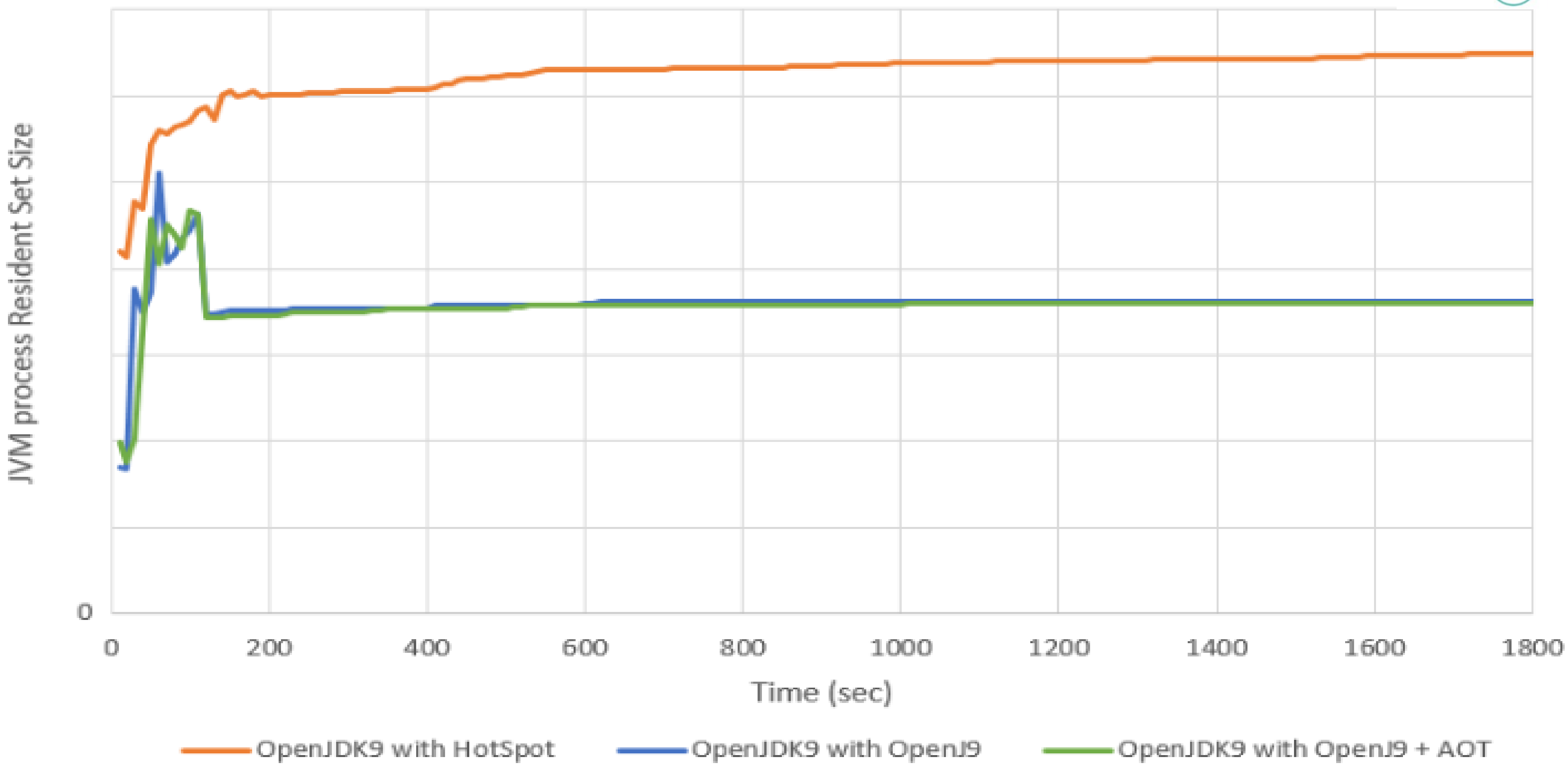
Comparable throughput

AZUL SYSTEMS

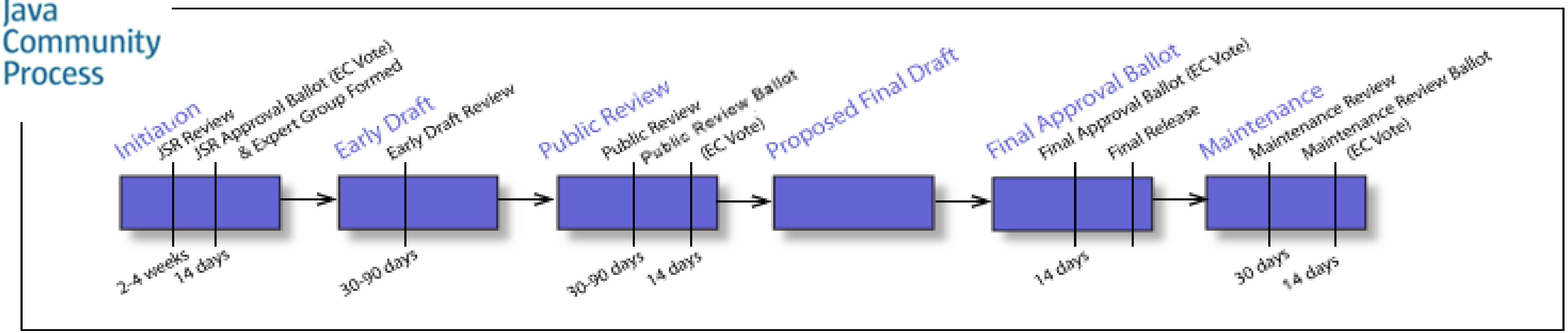


LLVM IR generated from bytecodes
Optimiser
LLVM IR
Back End
Platform specific machine code

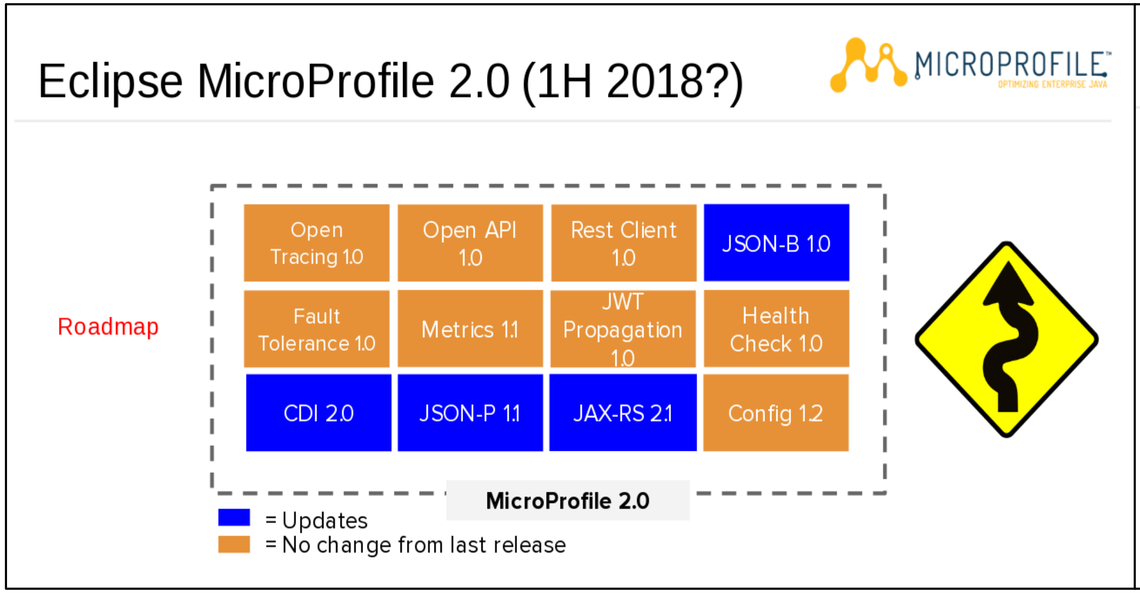
Footprint size during ramp up (with -Xmx1G)



Java Community Process



eclipse



Eclipse MicroProfile 2.0 (1H 2018?)

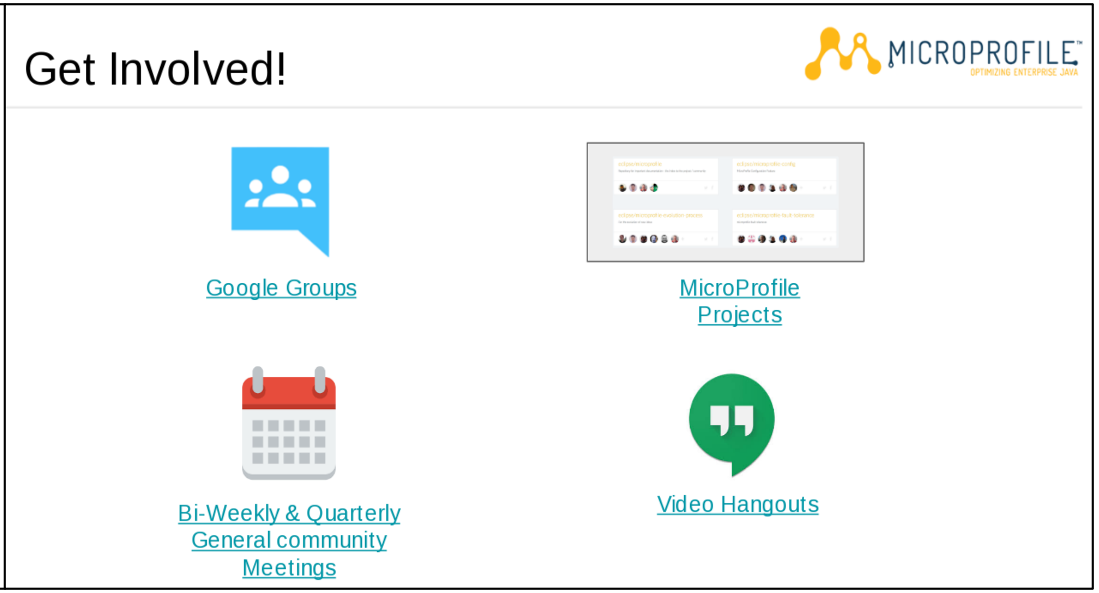
MICROPROFILE OPTIMIZING ENTERPRISE JAVA

Roadmap

Open Tracing 1.0	Open API 1.0	Rest Client 1.0	JSON-B 1.0
Fault Tolerance 1.0	Metrics 1.1	JWT Propagation 1.0	Health Check 1.0
CDI 2.0	JSON-P 1.1	JAX-RS 2.1	Config 1.2

MicroProfile 2.0

🟦 = Updates
🟧 = No change from last release



Get Involved!

MICROPROFILE OPTIMIZING ENTERPRISE JAVA

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

OpenJDK

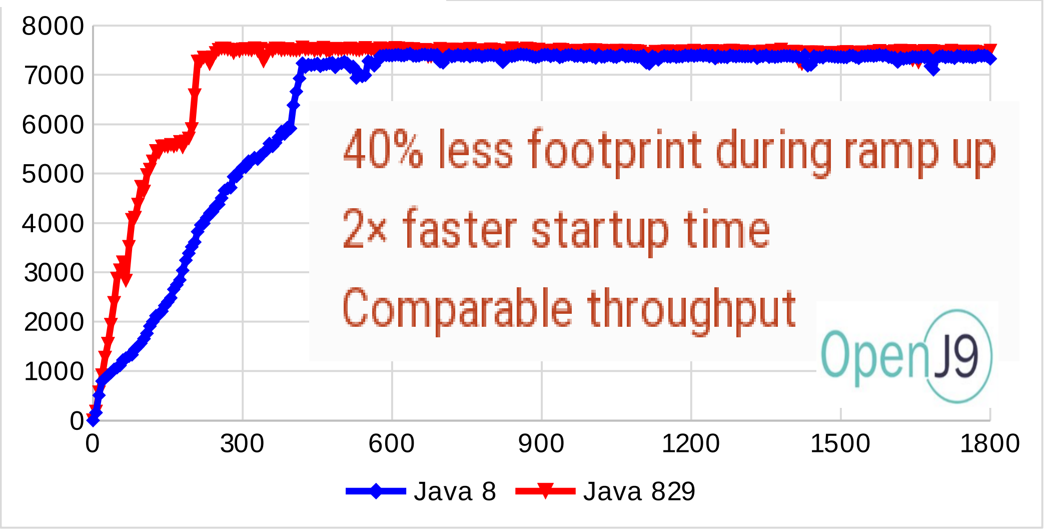
```
module example {
  requires helloworld;
}
```

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

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

jshell>
```

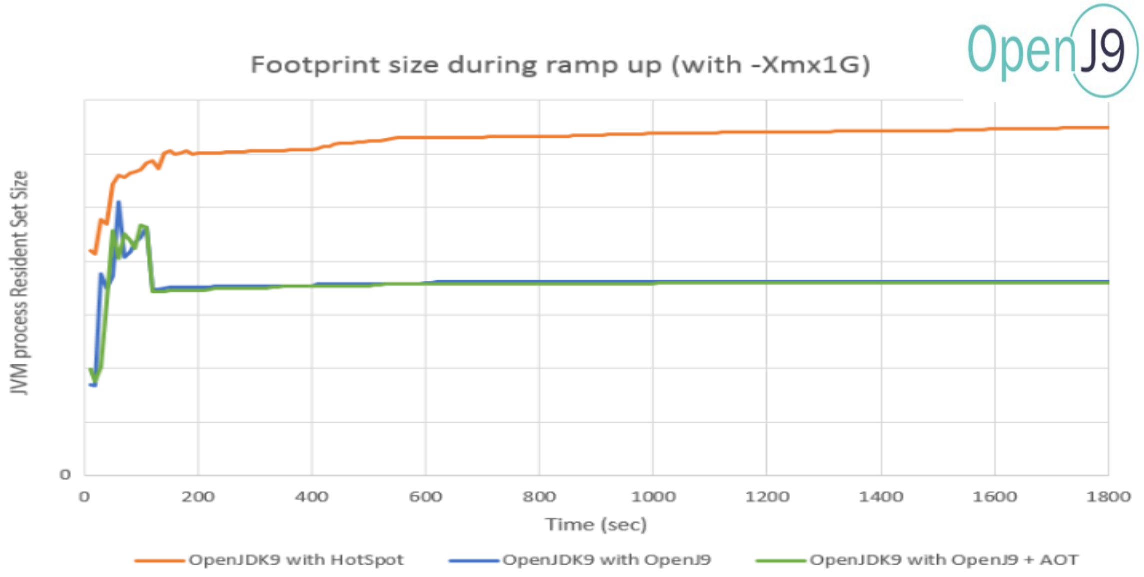
eclipse



40% less footprint during ramp up

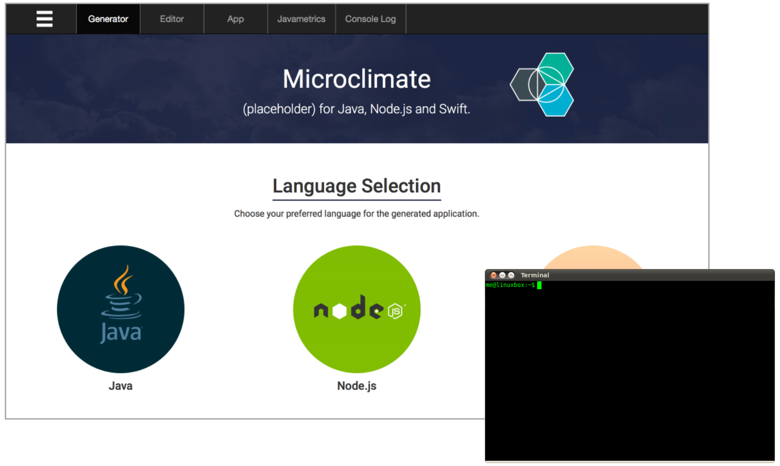
2× faster startup time

Comparable throughput



Microclimate: seamless cloud development experience

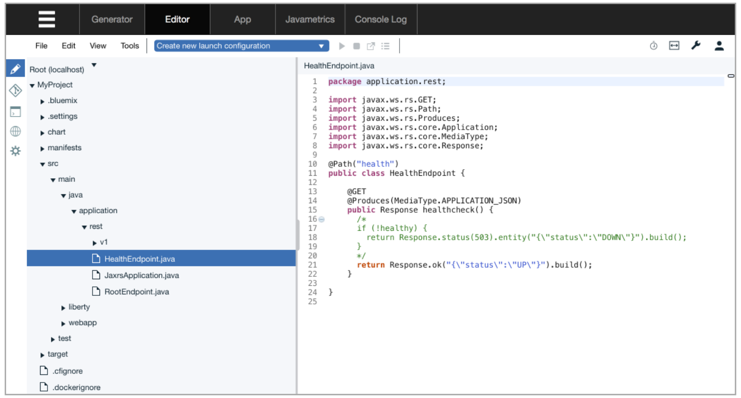
Rapid on-boarding



hosted, containerized, or command-line tools

docker

Web-based IDE supporting full live sandbox lifecycle, or BYOE



```
package application.rest;

import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.Produces;
import javax.ws.rs.core.Application;
import javax.ws.rs.core.MediaType;
import javax.ws.rs.core.Response;

@Path("health")
public class HealthEndpoint {

    @GET
    @Produces(MediaType.APPLICATION_JSON)
    public Response healthcheck() {
      /*
      if (!healthy) {
        return Response.status(503).entity("{\"status\":\"DOWN\"}").build();
      }
      */
      return Response.ok("{\"status\":\"UP\"}").build();
    }

}

```

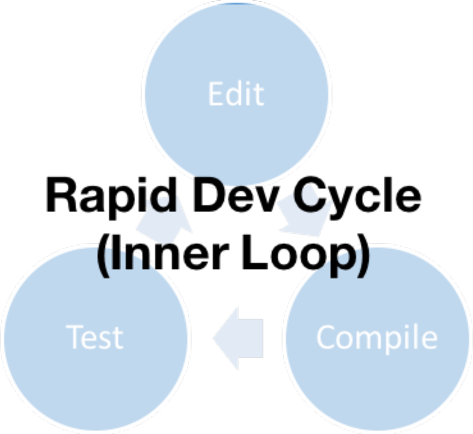
VS Code

ORION

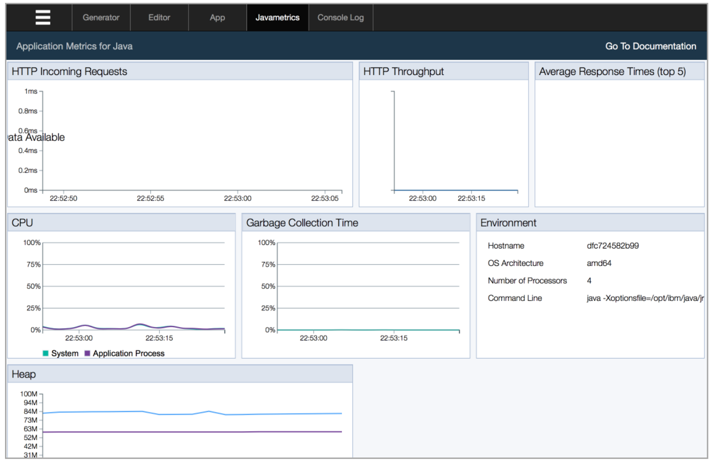
Powered by

Language Server Protocol (LSP)

Rapid Dev Cycle (Inner Loop)



Instant monitoring and feedback



Environment

Hostname	dfc724582b99
OS Architecture	amd64
Number of Processors	4
Command Line	java -Xoptionsfile=/opt/ibm/java/jr

Consistent tools from dev to ops