From Java 11 to 17

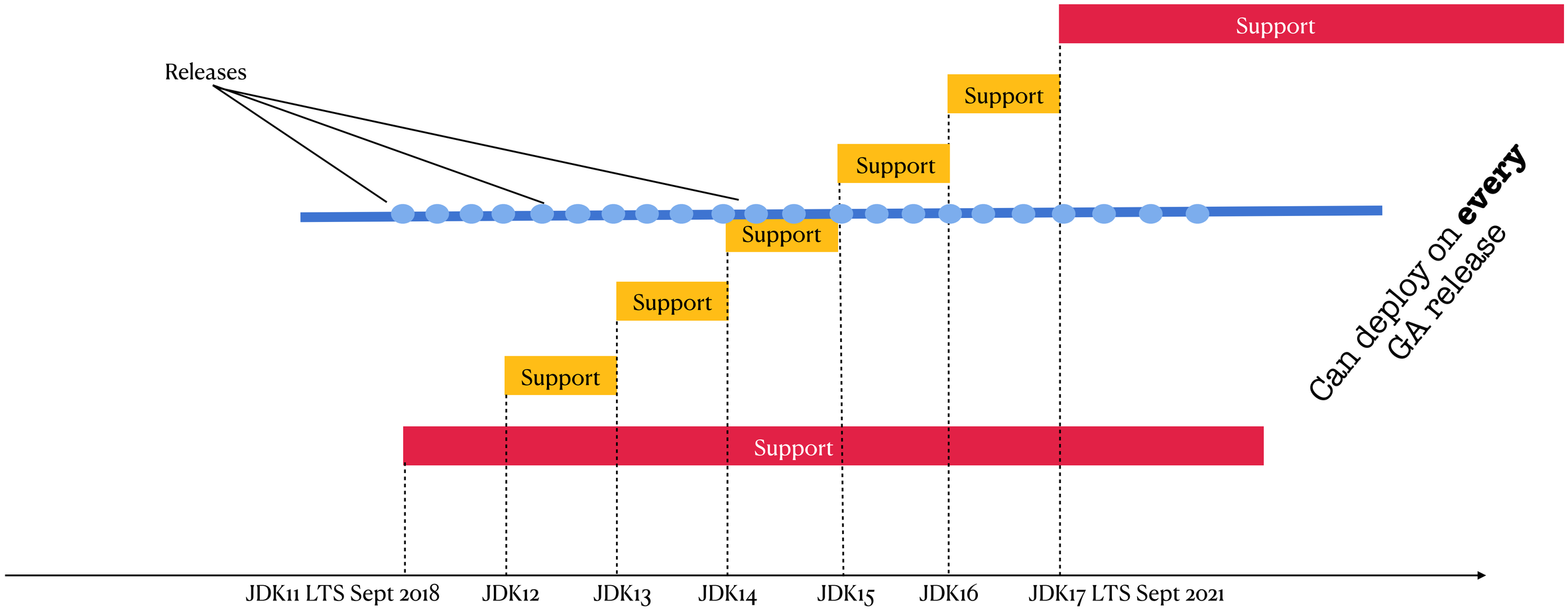
Mini talk

Ivan Krylov
JokerConf Program Commette

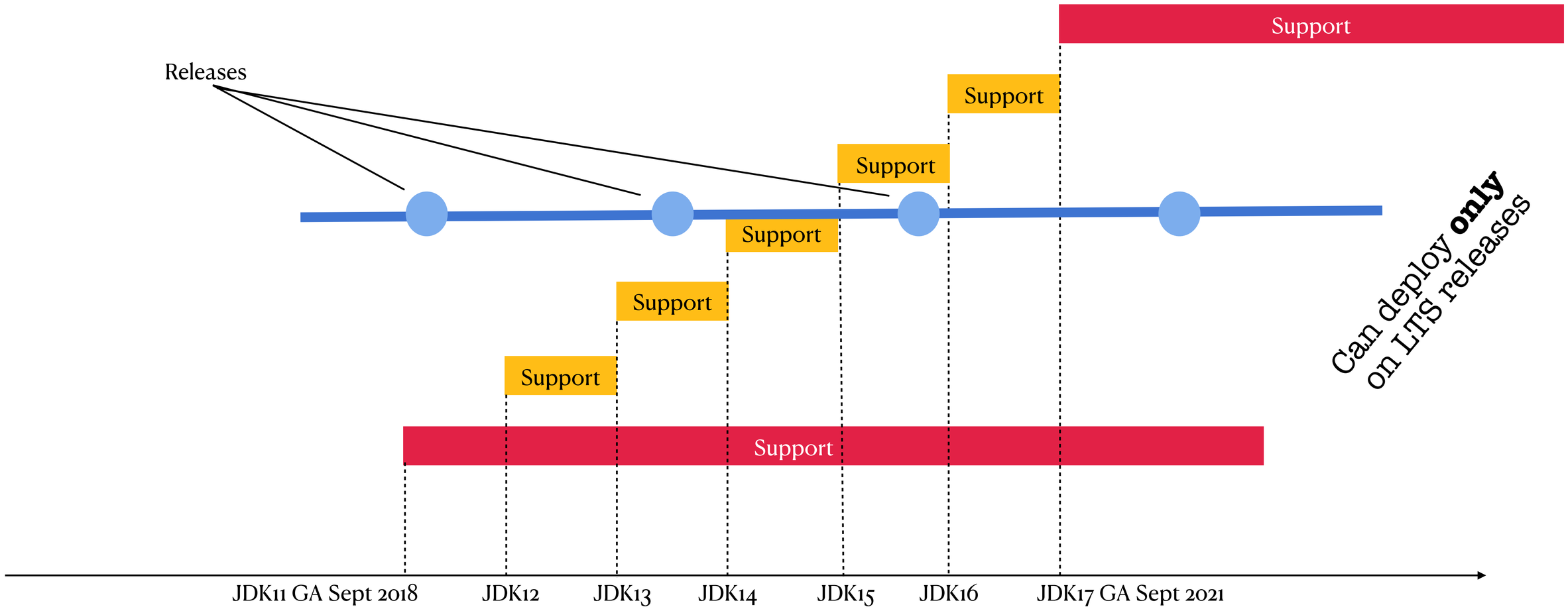
Agenda

- Why JDK 17 is important
  - Migration paths and adoption of interim releases
- Overview of breaking API changes
- JVM changes
- Recall of major API and language features
- What JDK 17 does not deliver
  - Valhalla, Panama, (final version of) Vector API and Loom

Situation 1: Project with frequent releases

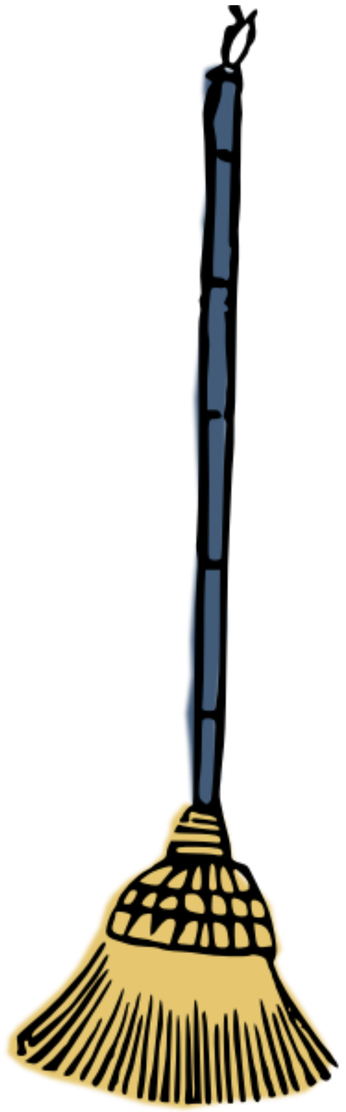


Situation 2: Project with infrequent releases



Breaking Changes - JDK Code Cleanup

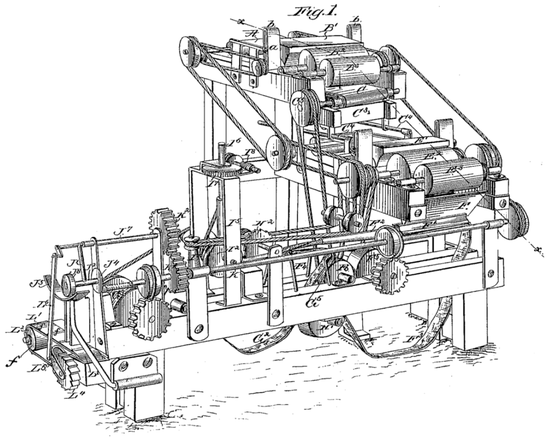
- Finalizers removed from some methods
  - `java.io.FileInput(Output)Stream`
  - `java.util.zip.(Deflater/Inflater/ZipFile)`
  - `java.awt.color.ICC_Profile`
- Package `java.util.jar`: Interfaces `Packer` and `Unpacker`, implementing class `Pack200` removed
- Constructor in `java.net.URLDecoder` removed
- A number of `AWT` and `Swing` classes changed visibility attributes and received other cleanup
- Deprecated for removal:
  - Applet API



Breaking Changes - Security and VM

Security & cryptography

- Different root certificates
- Removal of NIST EC Curves from the default TLS Algorithms
- Disabled Native SunEC Implementation by Default
- Deprecated for removal: Security manager
- Removed **`java.security.acl`** & **`java.rmi.activation`** packages



VM

- GraalVM JIT no longer bundled with JDK
- CMS GC removed
- Biased locking removed (impacting, not breaking)

Key JVM Changes

Lot's of small changes under the hood

- GC
  - CMS is gone
  - ZGC is very good and production ready
    - But don't use ZGC in JDK 11
  - Unused memory returned to the OS (G1, earlier in Shenandoah)
  - Abortable Mixed Collections for G1
  - NUMA-Aware Memory Allocation for G1
  - More concurrent ops => smaller GC pauses for all GCs
- JIT
  - GraalVM JIT is outside of OpenJDK
  - Better tuning for C2 inlining
- Runtime
  - Default CDS Archives
  - Dynamic CDS Archives
  - Elastic metaspace
- Serviceability
  - JFR Enhancements: Streaming events

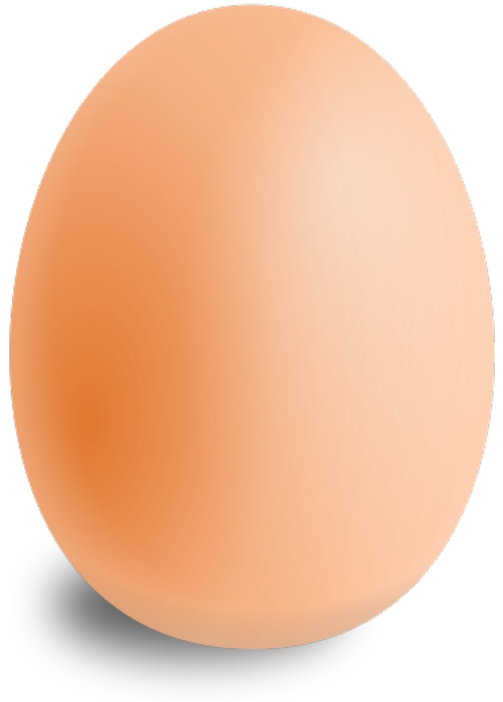
New Features

Including major ones from 12..16

- Language
  - Sealed classes
  - Switch Expressions
  - Enhanced *instanceof*
  - Text Blocks
  - Records
- Networking: Reimplemented Legacy APIs
  - DatagramSocket API
  - Socket API
- Helpful NPEs!
- Reflection & Method Handles
  - Hidden Classes
  - JVM Constants API
- Security:
  - Enhanced Modern Cryptography (EdDSA)
  - Context-Specific Deserialization Filters
  - Enhanced Pseudo-Random Number Generators
- Platform-specific changes
  - macOS: rendering using Metal API

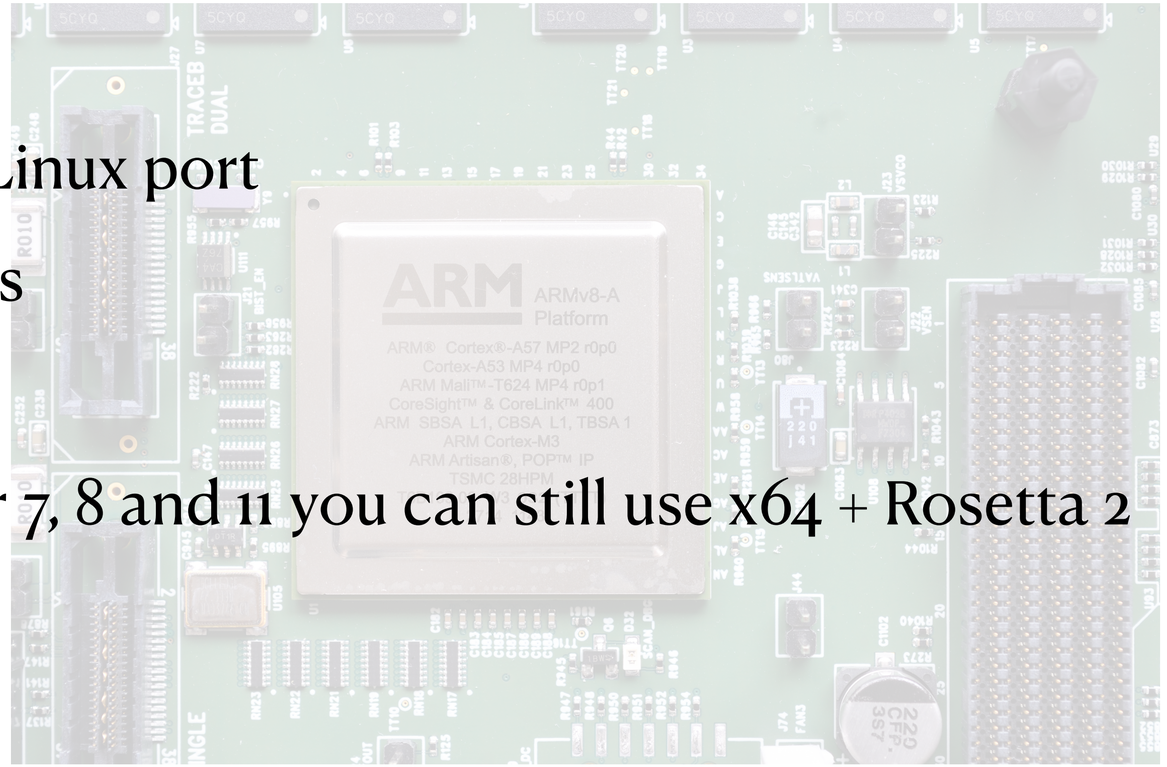
Inclubating and Preview Features

- Pattern Matching for switch
- Vector API
- Foreign Functions and Memory API



New and Developing Platforms

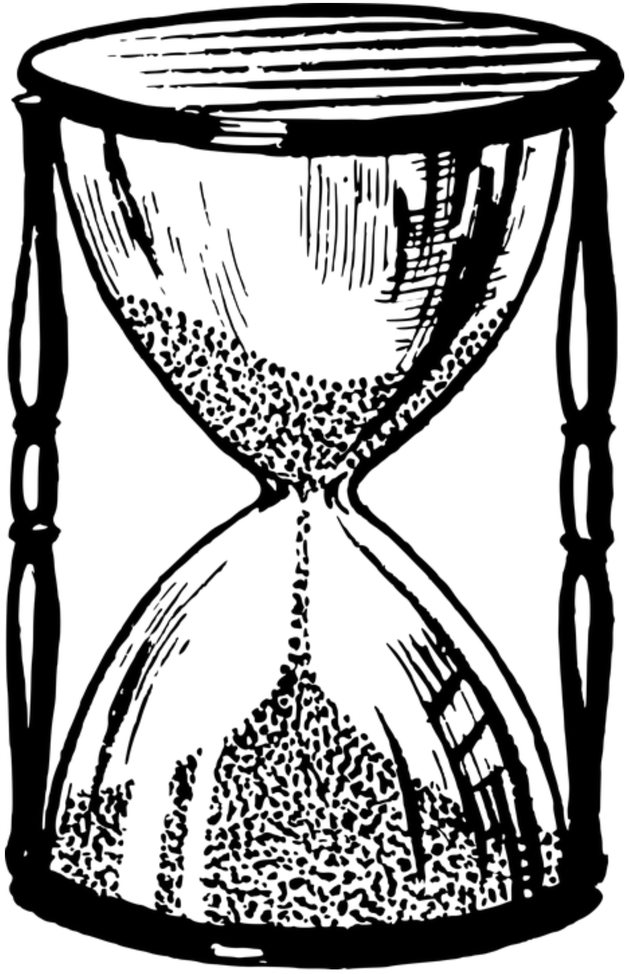
- Alpine Linux
  - Containers getting more space-efficient
- ARM
  - Unified Linux port
  - Windows
  - macOS
    - Yet for 7, 8 and 11 you can still use x64 + Rosetta 2



	x64	ARM
Linux	✓	✓
Windows	✓	✓
macOS	✓	✓

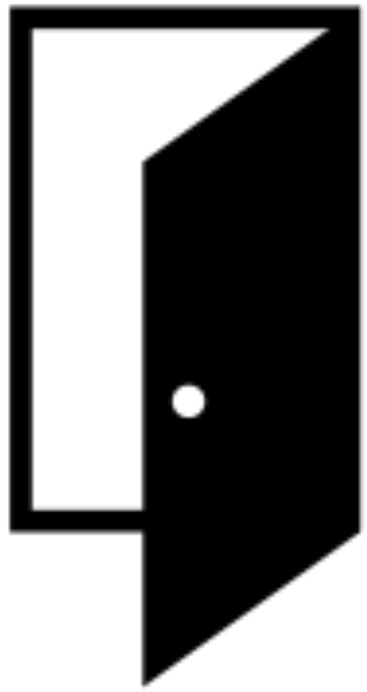
We Waited and Waited and Will Wait More

- Loom
  - Light-weight threads.... lots of them
- Panama
  - Native and java code with no borders
- Valhalla
  - “Codes like class, works like an *int*”



Summary

- Now is the time to migrate to JDK 17
  - But not because of new features
- Migration should be painless
- Know your dependancies
- JVM is faster, revolution in GCs
- IDEs are ready, most tools are unto date
- Vendors can vary the content of releases, e.g. Certificates, GraalVM presence, etc.
- All relevant platforms supported, small improvements across all platforms



Thank you

Image credits:

https://commons.wikimedia.org/wiki/File:ARMCortexA57A53.jpg

https://freesvg.org/broom-image

https://pixabay.com/vectors/lock-locked-metal-protection-tool-24269/

https://freesvg.org/egg

https://nl.wikipedia.org/wiki/Machine

https://pixabay.com/vectors/hourglass-sand-clock-sand-timer-2027541/

https://thenounproject.com/term/open-door/370565/