

Why Have The OSGi Specifications Been Based On Java™ Technology ?

By

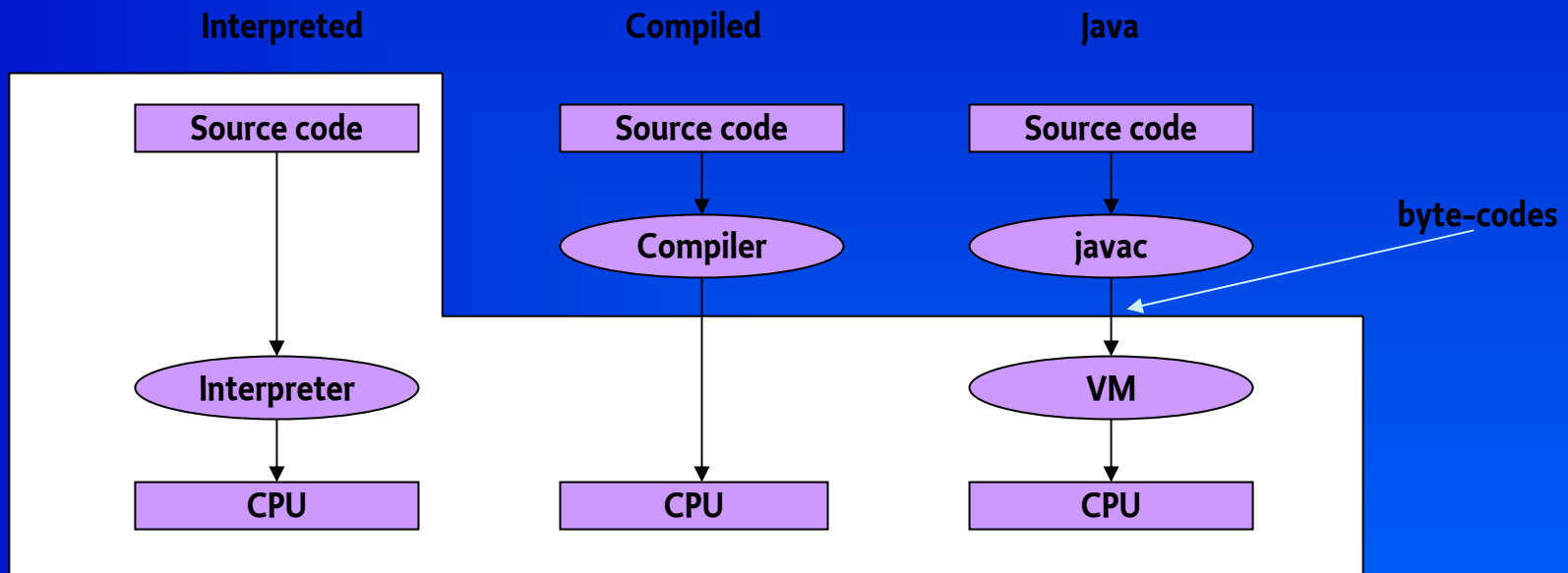
Peter Kriens, CEO aQute
OSGi Technology Officer

www.aQute.se

Content

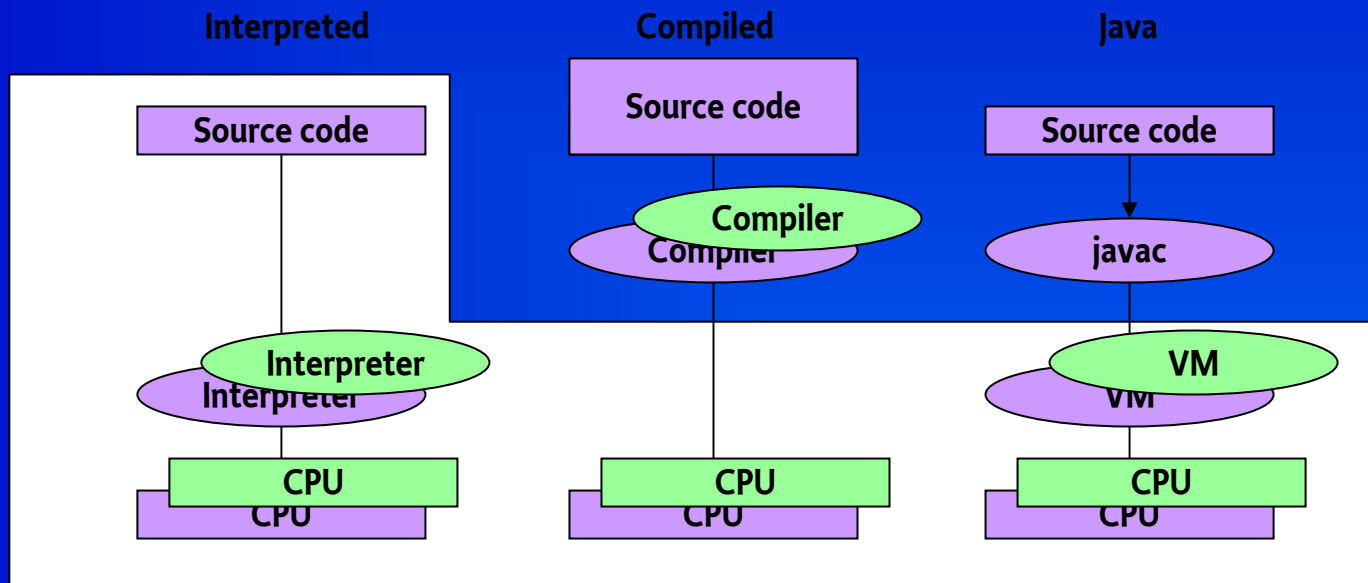
- Interpreted and Compiled languages
- Detailed look at Issues
- Comparison with .NET
- Conclusion

Interpreted Languages



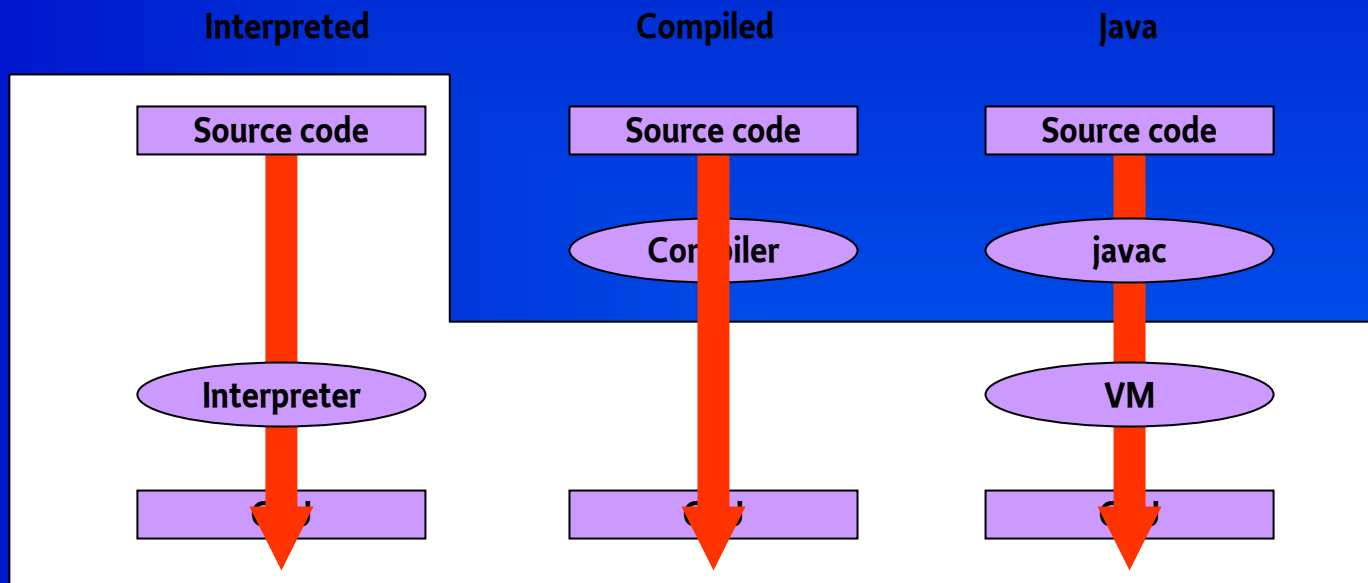
- Interpreter/VM is between source code and CPU
- Java is an interpreter with a defined executable format: VM

Portability



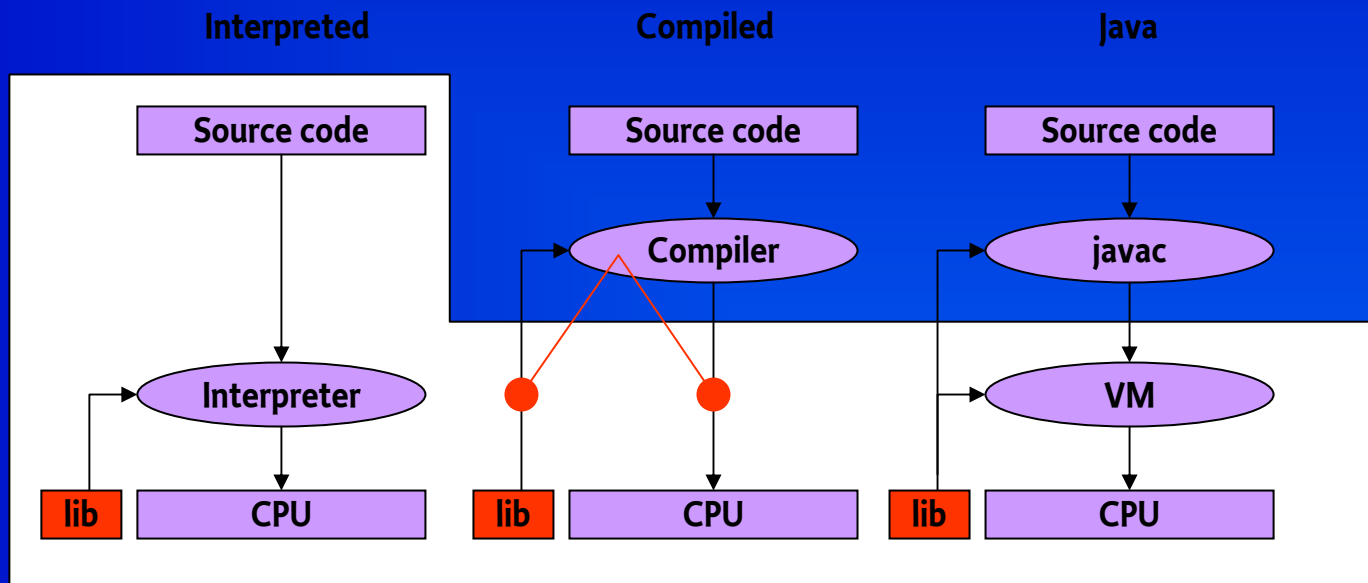
- Interpreters are easier to port to other environments
- No Source Code changes required

Illegal Access To Memory



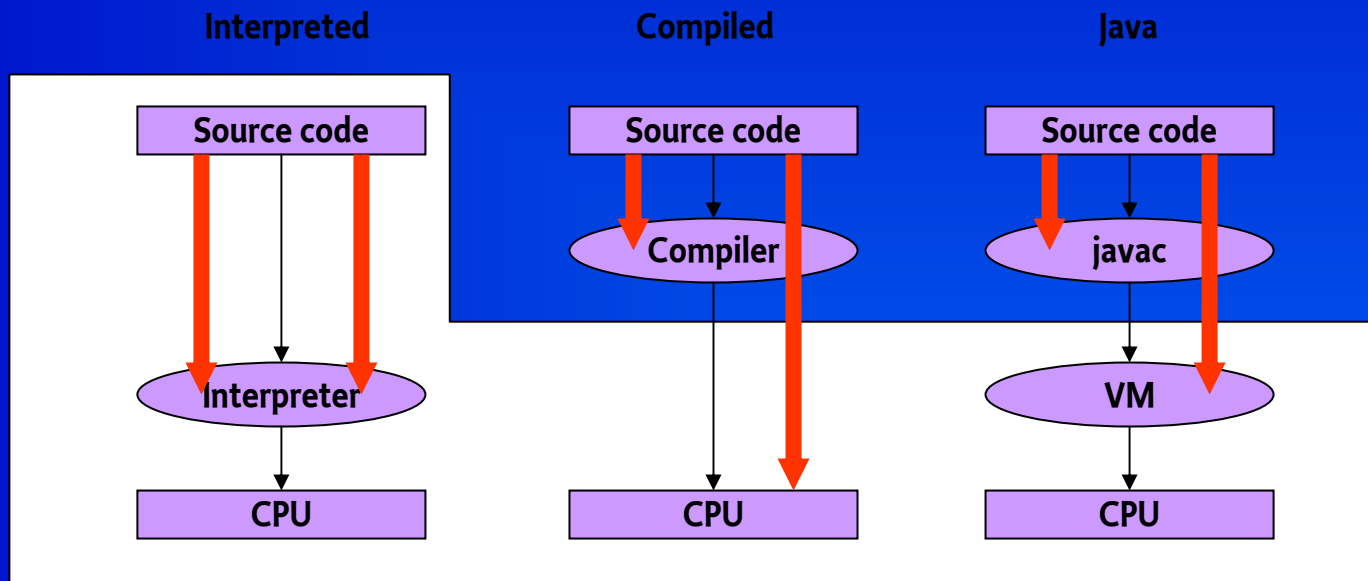
- No Illegal Access: Reduces security threats, Improves stability
- No (more) memory exceptions and buffer overruns

Versioning



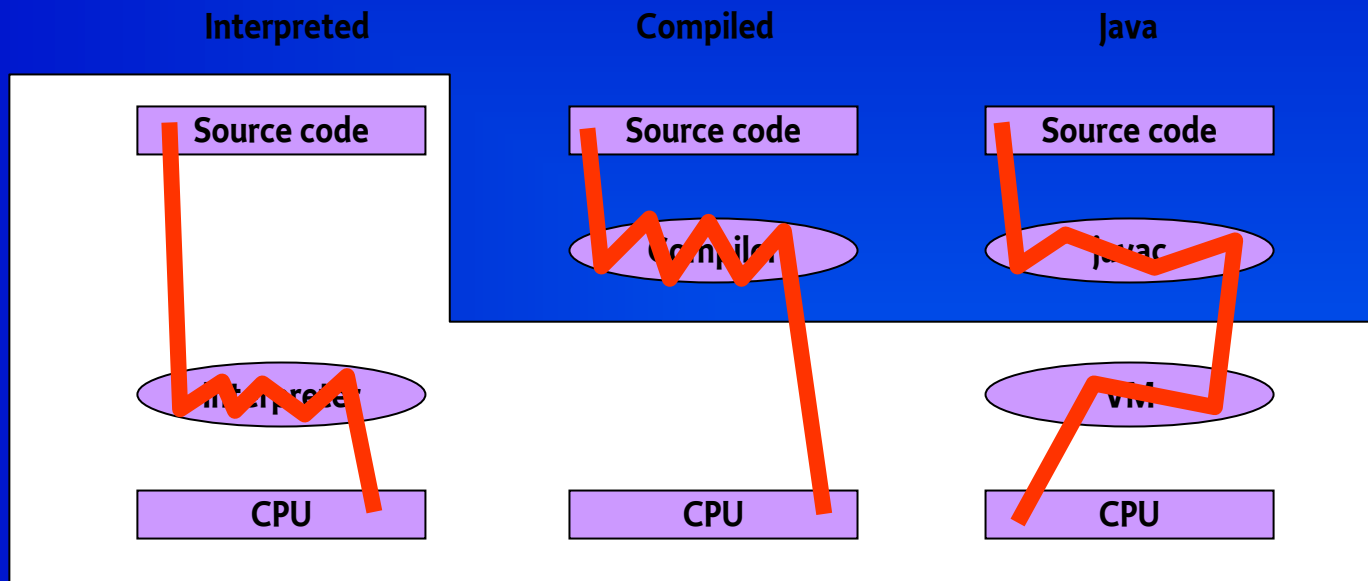
- Solves many version issues: Simpler deployment
- No More: Requires version 5.12.1.4-kwt
- Interpreters use symbolic linking: more robust

Type Problems Detection



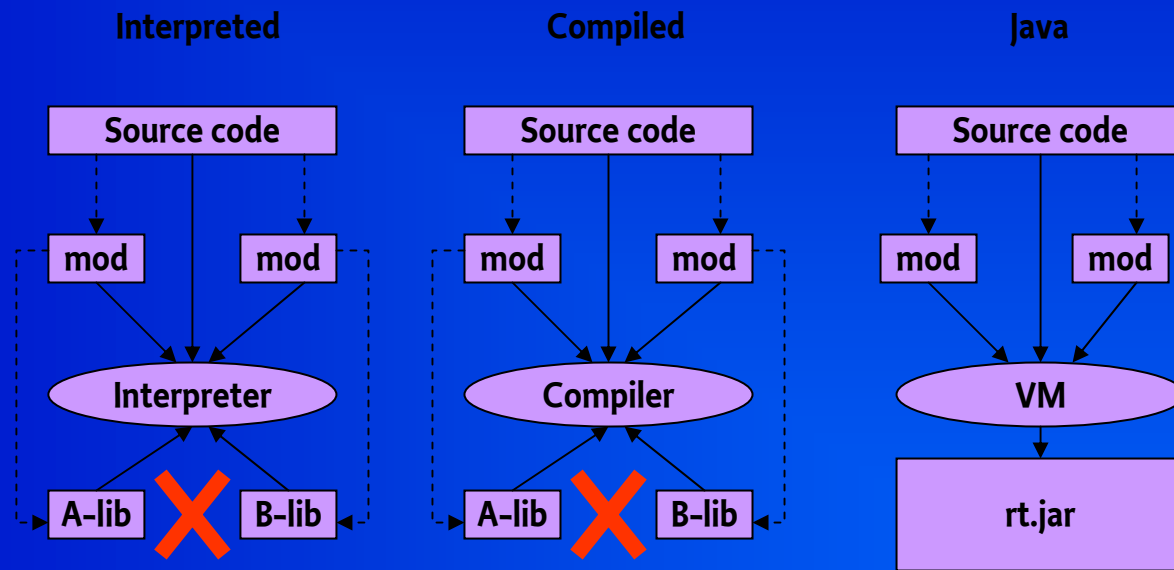
- Early detection of bugs: improved productivity
- Accidental and intended type violations
- Java finds all type errors, and many early

Performance



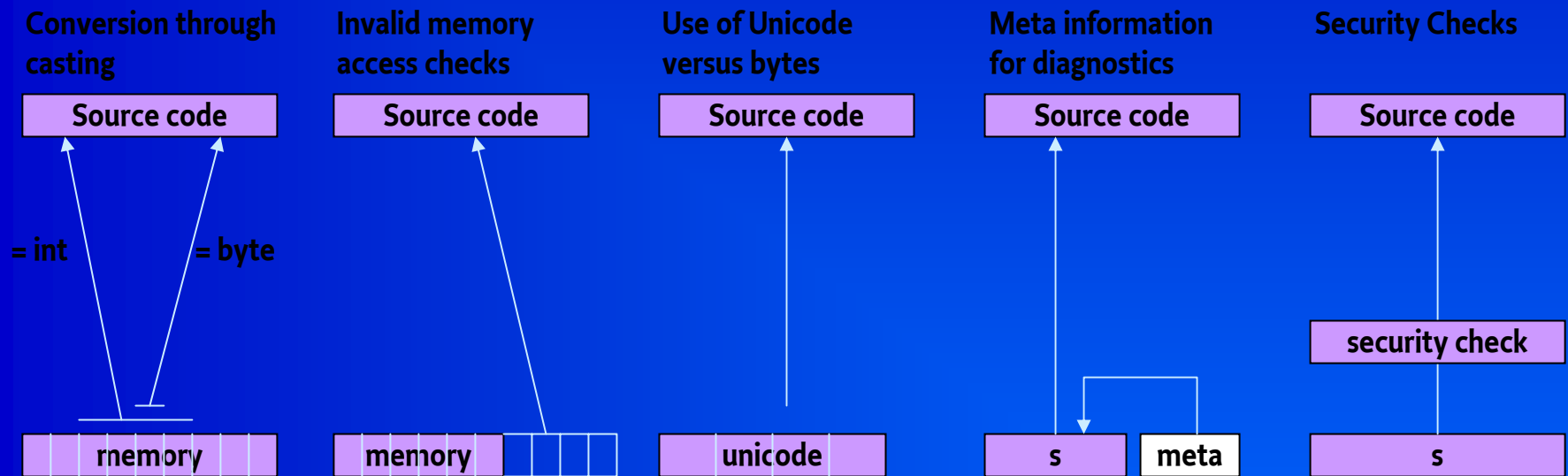
- Pre-processing reduces overhead: faster startup
- Compiled code is most efficient
- JIT techniques can reduce the performance gap

Too Many Libraries ...



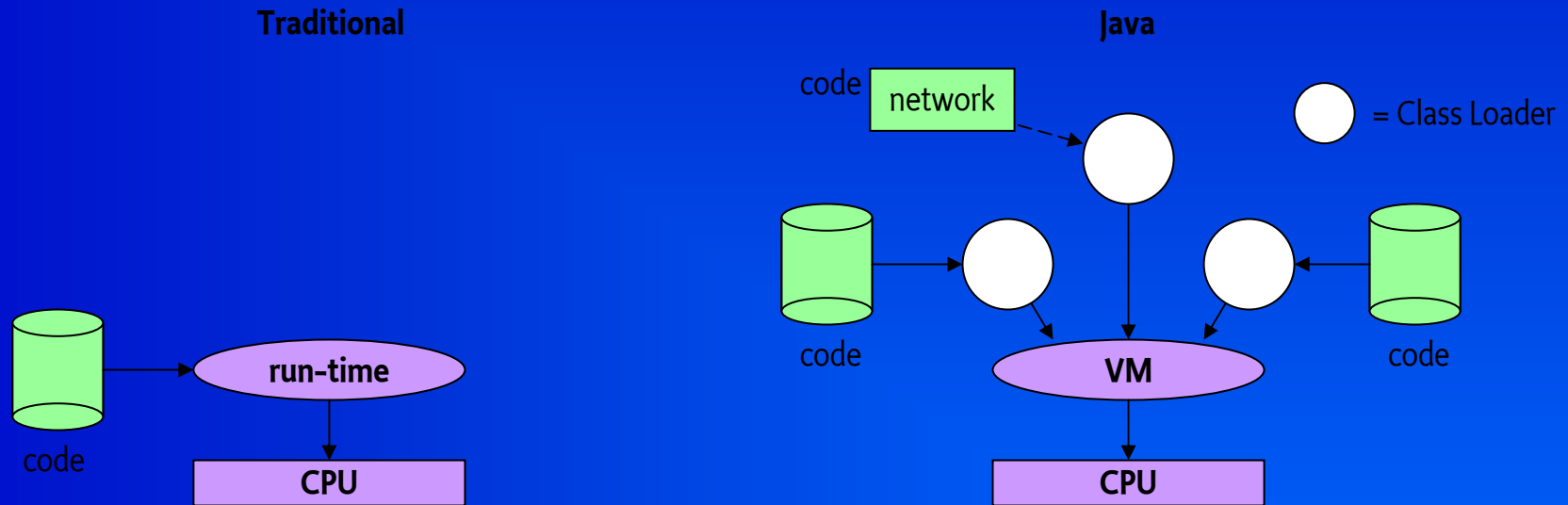
- Foundation library: One App for all platforms!
- Problems increase exponentially!
- Java very well defined, including run-time

Do It Right versus Performance



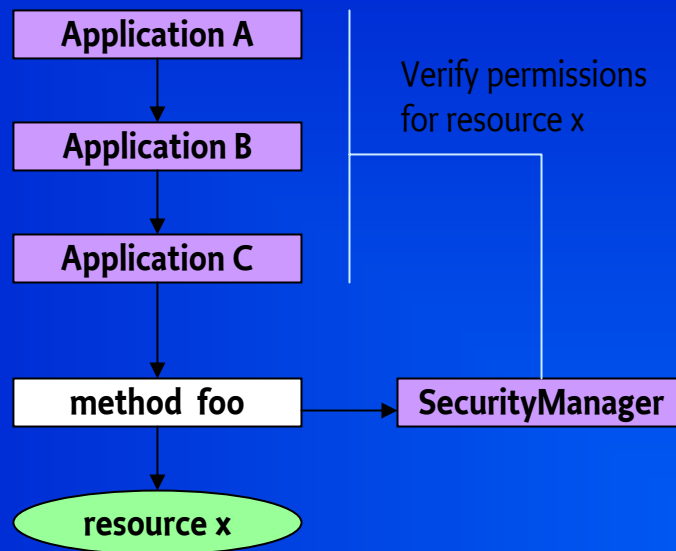
- Cannot perform *hacks*: security versus performance
- Traditional code has *hacks* that perform well
- *Hacks* increase maintenance cost and field problems

Life Cycle Management



- Loading policy defined in application code: Flexibility
- Used by the OSGi specifications
- Policies implemented by application code

Security



- Language includes security check: Safe code
- Prevents against viruses and worms
- First language with built-in security architecture

But What About Microsoft's .NET ?

What is .NET?



Java versus .NET CLI/CLR

- Java VM technology is similar to .NET C#
- But the Java environment is:
 - More mature
 - Not controlled by a single vendor: JCP
 - Applicable for embedded devices
 - Coherent environment (single syntax)
 - Exists for virtually every OS in existence, today
 - Runs the OSGi specifications!

Conclusion

- Portability
- Safety and security features
- Increased productivity
- Class loaders make OSGi possible
- Interpreted language is the right choice for the OSGi
- Should we use Java today [again]?
- YES!

Questions

?

Contact:

Peter.Kriens@aQute.se