Software development with Scala

Vladyslav Pekker

Hochschule Konstanz
University of Applied Sciences

What is this course about?

80% Programming in Scala

- Syntax
- Semantics

What is this course about?

80% Programming in Scala

- Syntax
- Semantics

20% Software development

- Programming languages
- Programming disciplines

What is the goal of this course?

The goal is to give a head start on learning Scala.

What is the goal of this course?

The goal is to give a head start on learning Scala.

Scala's learning curve

Part I

The Java Ecosystem

Outline

1 Fight metal programming

2 A History Lesson

Once upon a time...

Once upon a time...

in the land of computers there were 10101110010101101...

Once upon a time...

in the land of computers there were 10101110010101101...
then assemblers came along...

Once upon a time...

in the land of computers there were 10101110010101101...
then assemblers came along...
followed by compilers...

Once upon a time...

in the land of computers there were 10101110010101101...
then assemblers came along...
followed by compilers...
and they lived happily ever after...

Once upon a time...

in the land of computers there were 10101110010101101...
then assemblers came along...
followed by compilers...
and they lived happily ever after... NOT!

Portability

Platform Dependency

```
for(os <- List("linux", "mac", "windows"))
  println("If you compile a C program under " + os
          + "you can run it only on " + os + ".")
```

Portability

Platform Dependency

```
for(os <- List("linux", "mac", "windows"))
  println("If you compile a C program under " + os
          + "you can run it only on " + os + ".")
```

Sometimes a recompilation is not sufficient. The code has to be adapted!

Portability

Platform Dependency

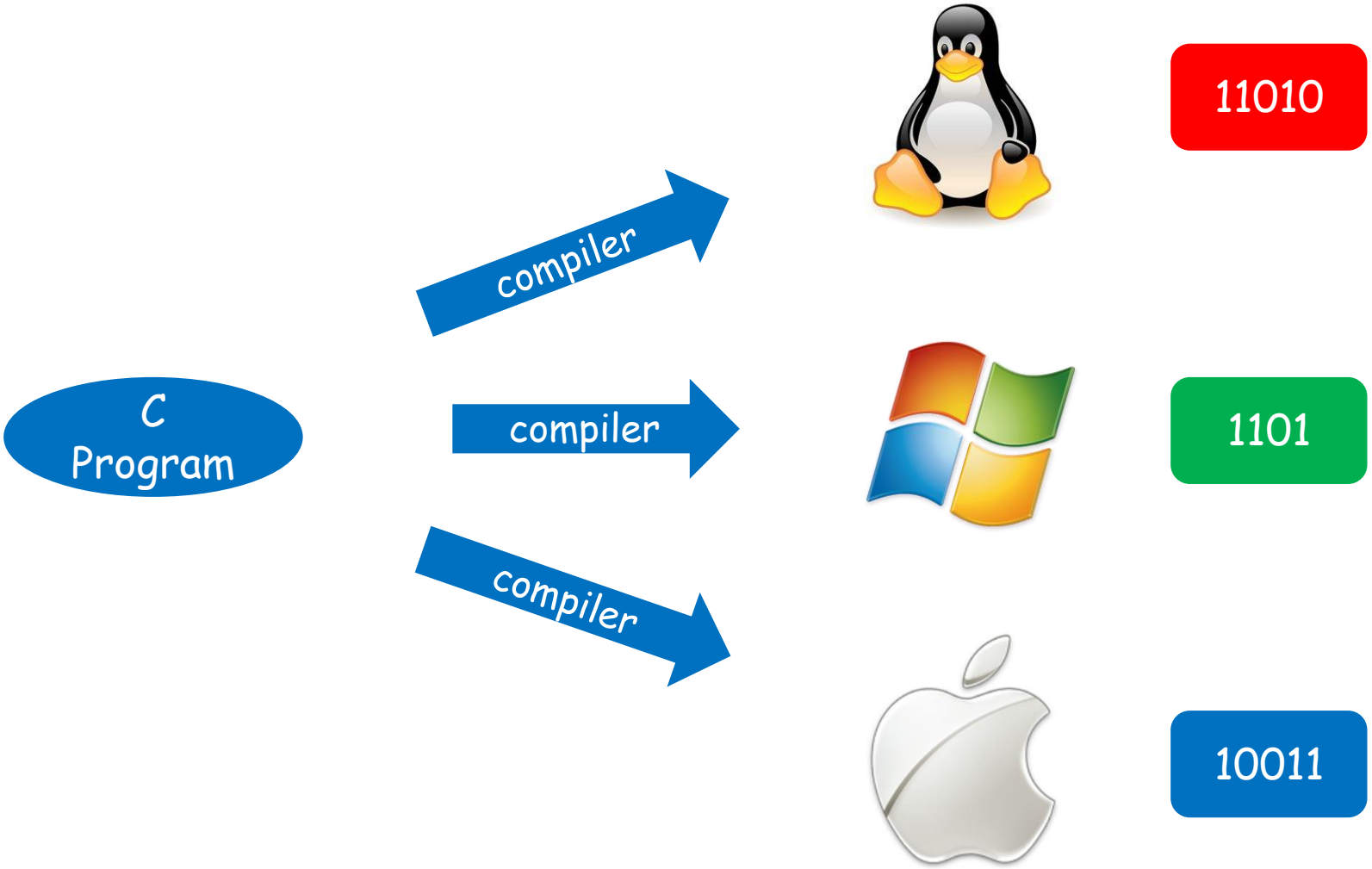
```
for(os <- List("linux", "mac", "windows"))
  println("If you compile a C program under " + os
          + "you can run it only on " + os + ".")
```

Sometimes a recompilation is not sufficient. The code has to be adapted!

Platform Independency

Write Once, Run Anywhere.

Problem - Unmanaged code



Problem - Unmanaged code 2

Why is unmanaged code a problem?

Problem - Unmanaged code 2

Why is unmanaged code a problem?

1. portability issues

Problem - Unmanaged code 2

Why is unmanaged code a problem?

1. portability issues
2. low level hazards like buffer overflows, segmentation faults etc.

Problem - Unmanaged code 2

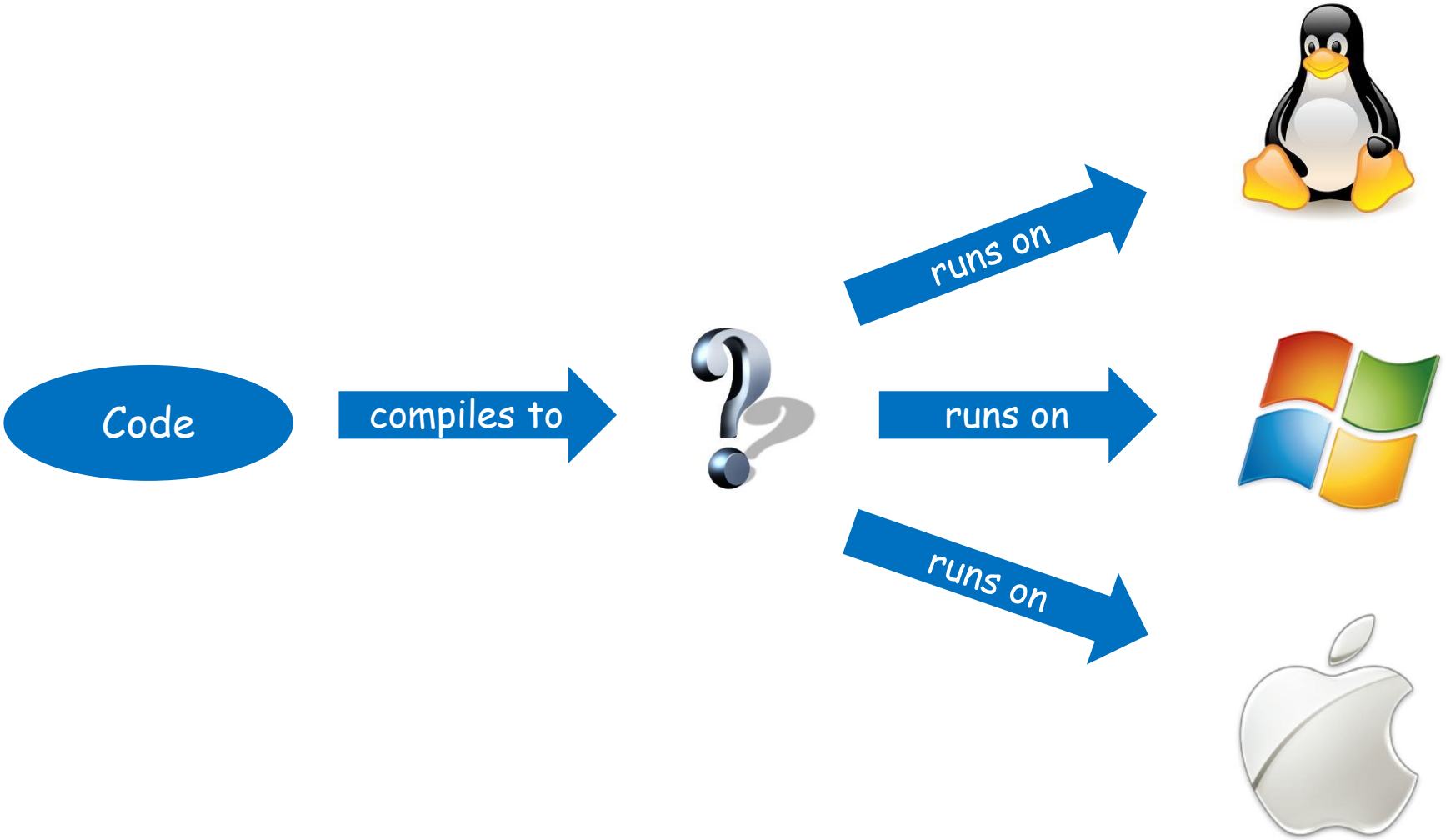
Why is unmanaged code a problem?

1. portability issues
2. low level hazards like buffer overflows, segmentation faults etc.

The problem, programming language designers are trying to solve:

Software development is not about programming the hardware, it is about abstracting over the hardware!

Solution - Managed code



Solution - Managed code 2

Why is managed code a solution?

Solution - Managed code 2

Why is managed code a solution?

1. portability issues are being taken care by VM vendors.

Solution - Managed code 2

Why is managed code a solution?

1. portability issues are being taken care by VM vendors.
2. low level hazards like buffer overflows, segmentation faults etc. are being handled by the VM.

Solution - Managed code 2

Why is managed code a solution?

1. portability issues are being taken care by VM vendors.
2. low level hazards like buffer overflows, segmentation faults etc. are being handled by the VM.
3. programs as-well as programming languages are changing every second; platforms, are fairly stable!

Solution - Managed code 2

Why is managed code a solution?

1. portability issues are being taken care by VM vendors.
2. low level hazards like buffer overflows, segmentation faults etc. are being handled by the VM.
3. programs as-well as programming languages are changing every second; platforms, are fairly stable!

Wind of Change

- it is feasible to change a program.
- it is hard to change a programming language.
- it is very hard to change a platform, but platforms are not required to be frequently changed.

Virtual Machines

Operating System

An operating system is an abstraction over a hardware machine .

Virtual Machines

Operating System

An operating system is an abstraction over a hardware machine .

Virtual Machine

A virtual machine is an abstraction over an operating system.

Virtual Machines

Operating System

An operating system is an abstraction over a hardware machine .

Virtual Machine

A virtual machine is an abstraction over an operating system.

Well-known VMs

Java	Java Virtual Machine (JVM)
.Net	Common Language Runtime (CLR)
Android	Dalvik virtual machine
	and *many more*

Managed Programming Languages

JVM - runs everywhere

Java Statically typed OO language

CLR - runs only on Windows

Managed Programming Languages

JVM - runs everywhere

Java	Statically typed OO language
Scala	Statically typed OO & FP language

CLR - runs only on Windows

Managed Programming Languages

JVM - runs everywhere

Java	Statically typed OO language
Scala	Statically typed OO & FP language
Clojure	Dynamically typed FP language (*Lisp* dialect)

CLR - runs only on Windows

Managed Programming Languages

JVM - runs everywhere

Java	Statically typed OO language
Scala	Statically typed OO & FP language
Clojure	Dynamically typed FP language (*Lisp* dialect)
Groovy	Dynamically typed scripting language

CLR - runs only on Windows

Managed Programming Languages

JVM - runs everywhere

Java	Statically typed OO language
Scala	Statically typed OO & FP language
Clojure	Dynamically typed FP language (*Lisp* dialect)
Groovy	Dynamically typed scripting language
	and *many more*

CLR - runs only on Windows

Managed Programming Languages

JVM - runs everywhere

Java	Statically typed OO language
Scala	Statically typed OO & FP language
Clojure	Dynamically typed FP language (*Lisp* dialect)
Groovy	Dynamically typed scripting language
	and *many more*

CLR - runs only on Windows

C#	Statically typed OO & FP language

Managed Programming Languages

JVM - runs everywhere

Java	Statically typed OO language
Scala	Statically typed OO & FP language
Clojure	Dynamically typed FP language (*Lisp* dialect)
Groovy	Dynamically typed scripting language
	and *many more*

CLR - runs only on Windows

C#	Statically typed OO & FP language
VB	Statically typed OO language

Managed Programming Languages

JVM - runs everywhere

Java	Statically typed OO language
Scala	Statically typed OO & FP language
Clojure	Dynamically typed FP language (*Lisp* dialect)
Groovy	Dynamically typed scripting language
	and *many more*

CLR - runs only on Windows

C#	Statically typed OO & FP language
VB	Statically typed OO language
F#	Statically typed FP language

Managed Programming Languages

JVM - runs everywhere

Java	Statically typed OO language
Scala	Statically typed OO & FP language
Clojure	Dynamically typed FP language (*Lisp* dialect)
Groovy	Dynamically typed scripting language
	and *many more*

CLR - runs only on Windows

C#	Statically typed OO & FP language
VB	Statically typed OO language
F#	Statically typed FP language
	and *many more*

Managed Programming Languages 2

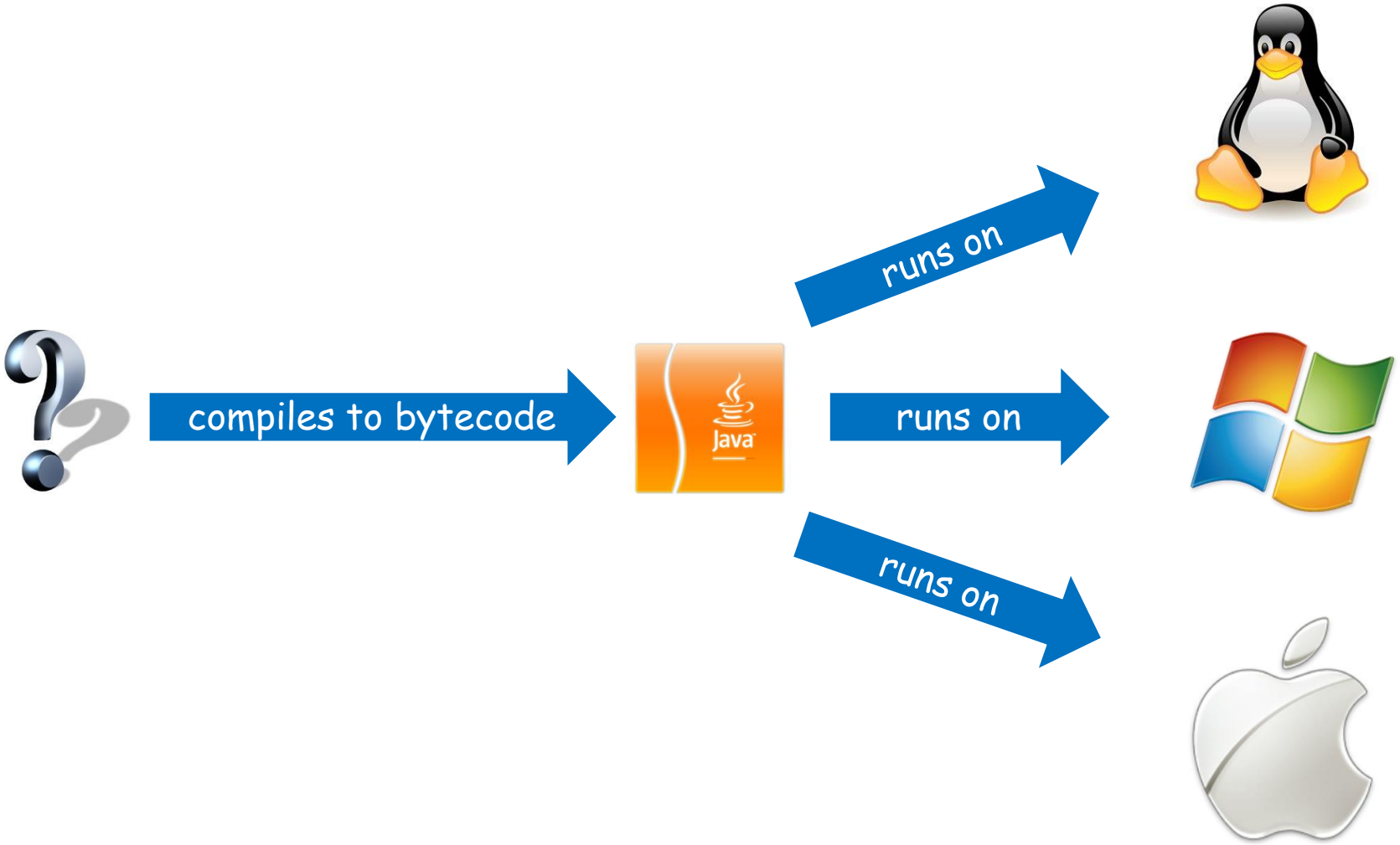
Dalvik - runs only on Android

Java is officially supported. Every JVM language should theoretically work. Scala works. *Few others* work as-well.

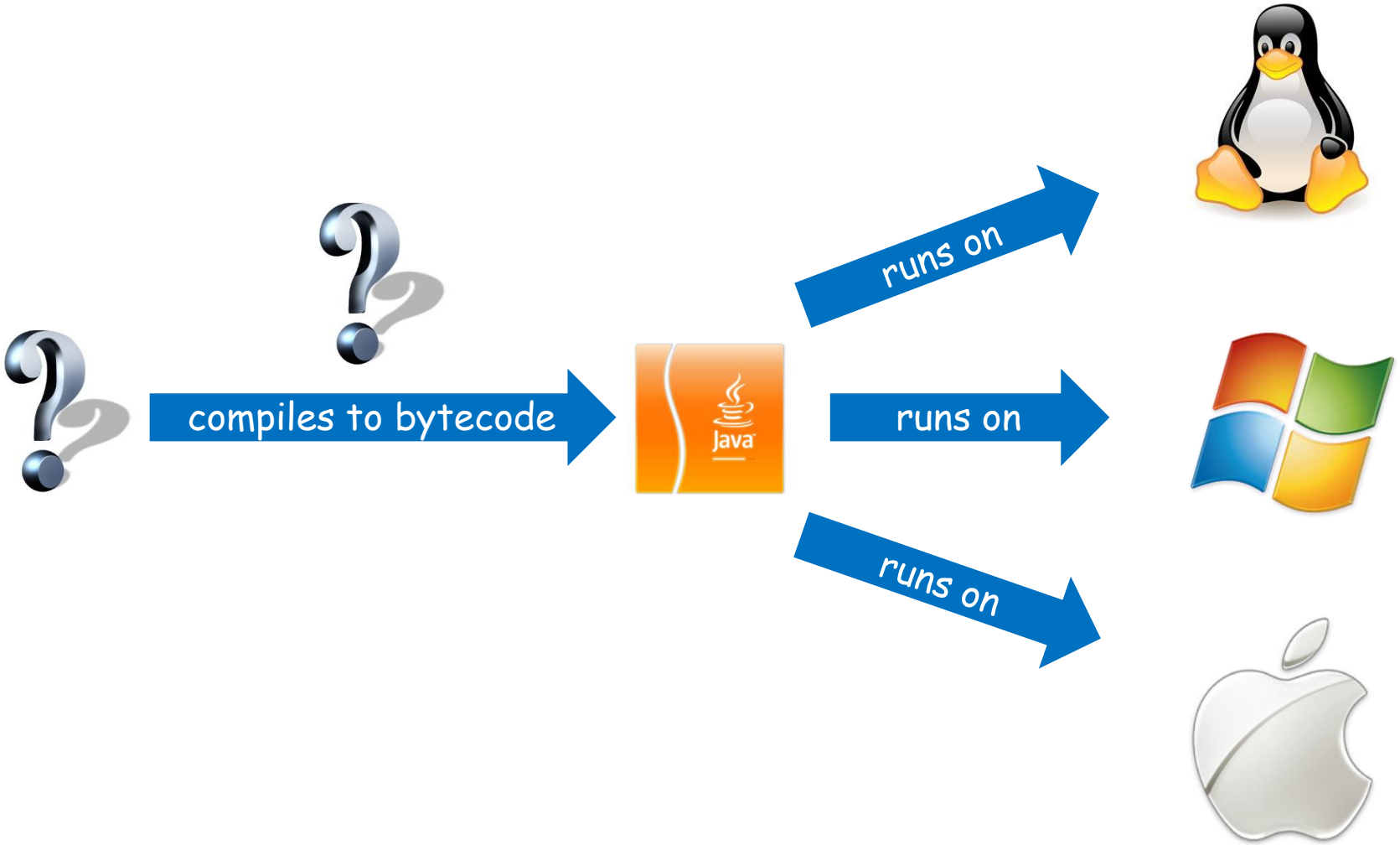
JVM - People's Choice

The code is not compiled to binary .
The code is compiled to the so called byte-code .
The JVM is a software component, which runs the byte-code .
A JVM implementation exists for almost every platform .
Java is the most used language on the JVM .
The most used Java compiler is called javac .

What language should we use for the JVM?



Can we reuse the java compiler for this language?



Outline

1. Fight metal programming
2. A History Lesson

Part II

Introducing Scala

Part III

Programming Paradigms

Part IV

Scala's Development Environment

Part V

Programming Styles

Part VI

Higher-Order Functions

Part VII

Growing The Language

Part VIII

Test-driven Development & Clean Code

Part IX

Pattern Matching

Part X

Traits

Part XI

Collections

Part XII

Concurrency

Part XIII

Domain Specific Languages

Part XIV

Higher-Kinded Types