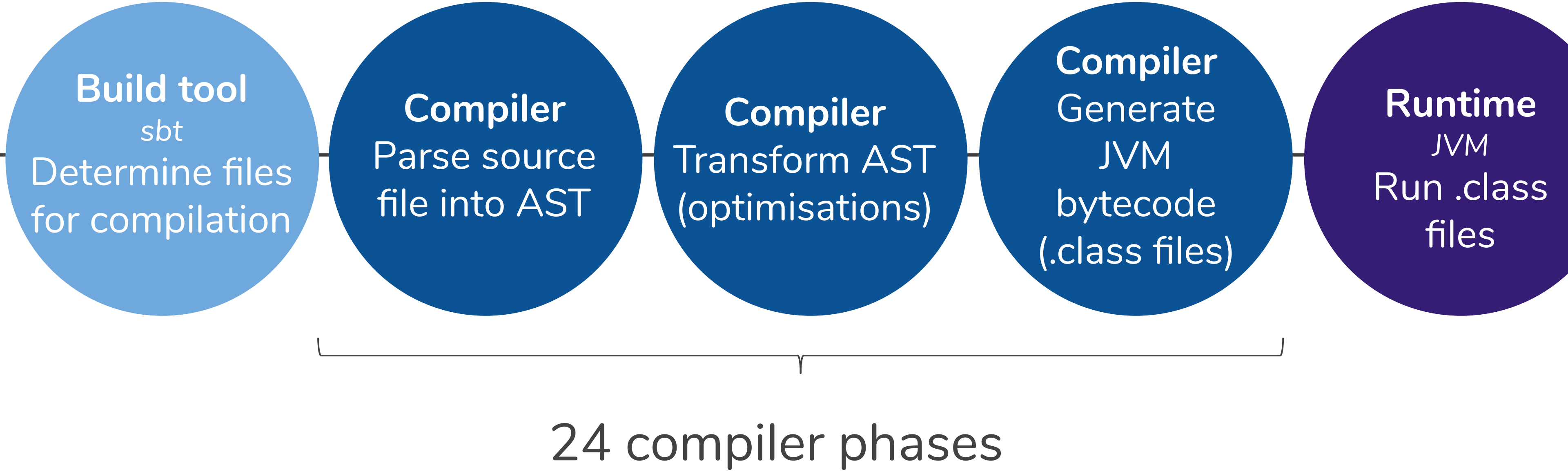
Modern cross-platform builds with Scala

Tim Nieradzik

Roadmap

- Standard build process
- Alternative targets
  - Scala.js
  - Scala Native
- Cross-platform builds
  - sbt
  - Bloop, Seed

Standard Build Process


Build tool
sbt
Determine files for compilation
Compiler
Parse source file into AST
Compiler
Transform AST (optimisations)
Compiler
Generate JVM bytecode (.class files)
Runtime
JVM
Run .class files
24 compiler phases

AST - Abstract Syntax Tree

Standard Build Process Compiler phases

```
$ scalac -Xshow-phases
    phase name  id  description
    ----------  --  -----------
        parser   1  parse source into ASTs, perform simple desugaring
         namer   2  resolve names, attach symbols to named trees
packageobjects   3  load package objects
         typer   4  the meat and potatoes: type the trees
        patmat   5  translate match expressions
superaccessors   6  add super accessors in traits and nested classes
[...]
         mixin  20  mixin composition
       cleanup  21  platform-specific cleanups, generate reflective calls
    delambdafy  22  remove lambdas
           jvm  23  generate JVM bytecode
      terminal  24  the last phase during a compilation run
```

Alternative Compilation Targets

Scala.js

Scala Native

Benefits

- Single-language code base
- Developers can do full-stack development
- Code sharing
  - Protocols
  - Templates
  - Validation logic
  - Business logic
- Interfacing with existing libraries (FFI)
  - Strongly typed
- Platform-agnostic code
  - Write logic for one platform, run/test on another

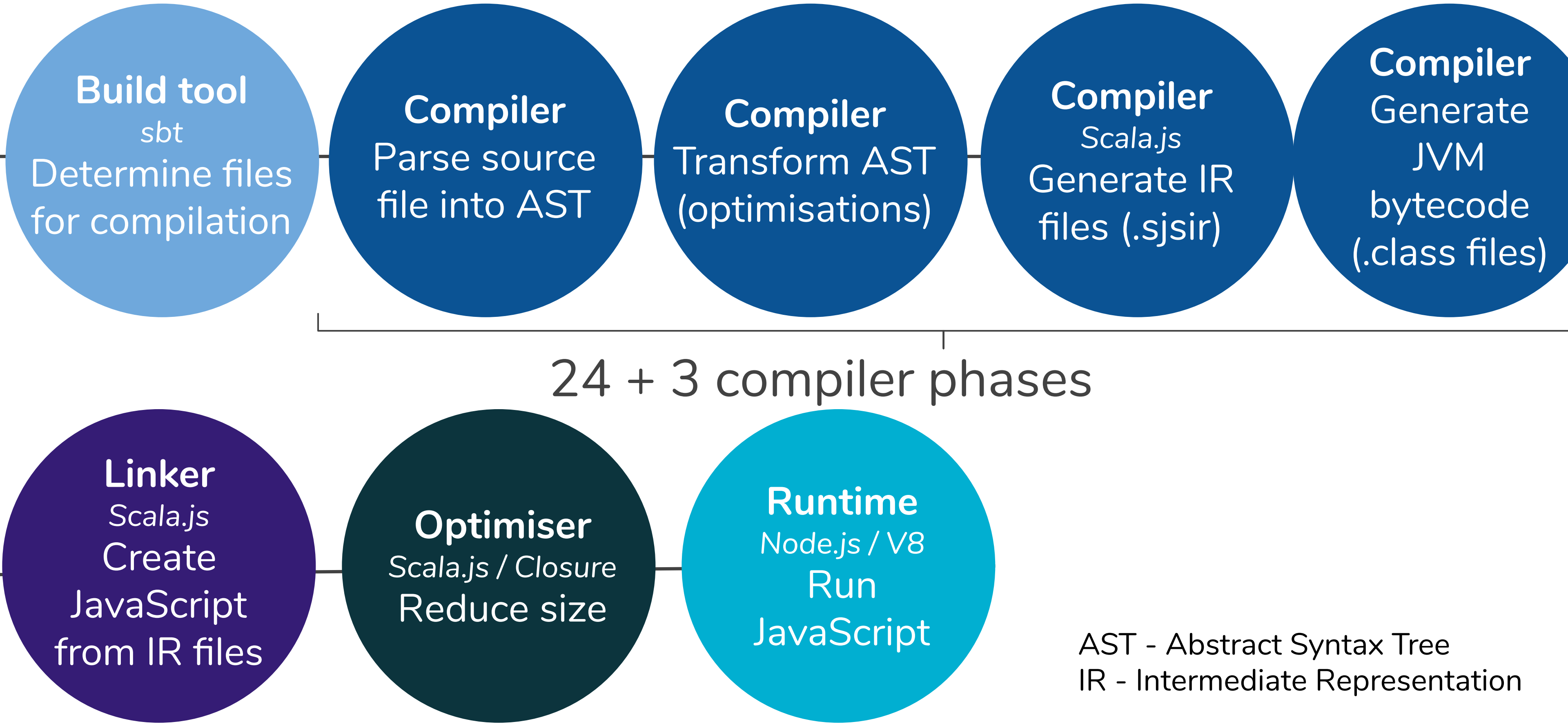
Benefits IDE support

- Import entire project in IntelliJ
- Do code refactoring across platform boundaries
- Support for auto-completions
- Jump back and forth between front- and back-end code
- Enforce uniform coding style

Use cases

- Scala.js
    - Web applications
    - Browser extensions
- Scala Native
    - Desktop GUIs
    - CLI tools
    - Games
    - Embedded software (only x86-64)

Scala.js Build Process


Build tool
sbt
Determine files for compilation
Compiler
Parse source file into AST
Compiler
Transform AST (optimisations)
Compiler
Scala.js
Generate IR files (.sjsir)
Compiler
Generate JVM bytecode (.class files)
24 + 3 compiler phases
Linker
Scala.js
Create JavaScript from IR files
Optimiser
Scala.js / Closure
Reduce size
Runtime
Node.js / V8
Run JavaScript

AST - Abstract Syntax Tree
IR - Intermediate Representation

Scala.js

- Implemented as Scala plug-in
- Re-uses all JVM phases
- 3 additional phases for typer, interoperability and IR generation
- ⚠️ JVM bytecode still generated for IDE support
- Separate linking phase required for IR → JavaScript
    - Generates source maps
- Introduction to Scala.js: http://www.lihaoyi.com/hands-on-scala-js/

```
$ scalac -Xshow-phases
-Xplugin:$HOME/.cache/coursier/v1/https/repo1.maven.org/maven2/org/scala-js/scalajs-compiler_2.
12.4/0.6.26/scalajs-compiler_2.12.4-0.6.26.jar
    phase name  id  description
    ----------  --  -----------
        parser   1  parse source into ASTs, perform simple desugaring
    jspretyper   2  capture pre-typer only tree info (for Scala.js)
         namer   3  resolve names, attach symbols to named trees
packageobjects   4  load package objects
         typer   5  the meat and potatoes: type the trees
     jsinterop   6  prepare ASTs for JavaScript interop
[...]
         mixin  22  mixin composition
        jscode  23  generate JavaScript code from ASTs
       cleanup  24  platform-specific cleanups, generate reflective calls
    delambdafy  25  remove lambdas
           jvm  26  generate JVM bytecode
      terminal  27  the last phase during a compilation run
```

Scala.js example

```
$ cat Test.scala

import scala.scalajs.js
object Test {
  val console = js.Dynamic.global.console
  def main(args: Array[String]): Unit = console.log("hello")
}
```

Compile Scala.js from CLI

```
$ export MAVEN=$HOME/.cache/coursier/v1/https/repo1.maven.org/maven2

$ scalac \
  -Xplugin:$MAVEN/org/scala-js/scalajs-compiler_2.12.4/0.6.26/scalajs-compiler_2.12.4-0.6.26.jar \
  -cp $MAVEN/org/scala-js/scalajs-library_2.12/0.6.26/scalajs-library_2.12-0.6.26.jar \
  Test.scala

$ ls
Test$.class  Test$.sjsir  Test.class  Test.scala
```

Scala Native

- Similar architecture to Scala.js
- Uses LLVM to compile to assembly
- Linked programs run without VM
  - Fast startup time
- Further resources: https://github.com/tindzk/awesome-scala-native

Comparison

	Scala.js	**Scala Native**
Versions	2.11, 2.12, 2.13	2.11
Language Features	All	All
Reflection	Partial[1]	No
Interoperability	Good	Moderate
Library support	Good	Spotty

[1] https://github.com/portable-scala/portable-scala-reflect

Default directory structure

```
├── js/src/{main,test}/scala
├── jvm/src/{main,test}/scala
├── native/src/{main,test}/scala
└── shared/src/{main,test}/scala
```

Cross-compiled build with sbt

```
addSbtPlugin("org.portable-scala" % "sbt-scalajs-crossproject"      % "0.6.1")
addSbtPlugin("org.portable-scala" % "sbt-scala-native-crossproject" % "0.6.1")
addSbtPlugin("org.scala-js"       % "sbt-scalajs"                   % "0.6.23")
addSbtPlugin("org.scala-native"   % "sbt-scala-native"              % "0.3.7")
```

File: project/plugins.sbt

See also https://github.com/portable-scala/sbt-crossproject

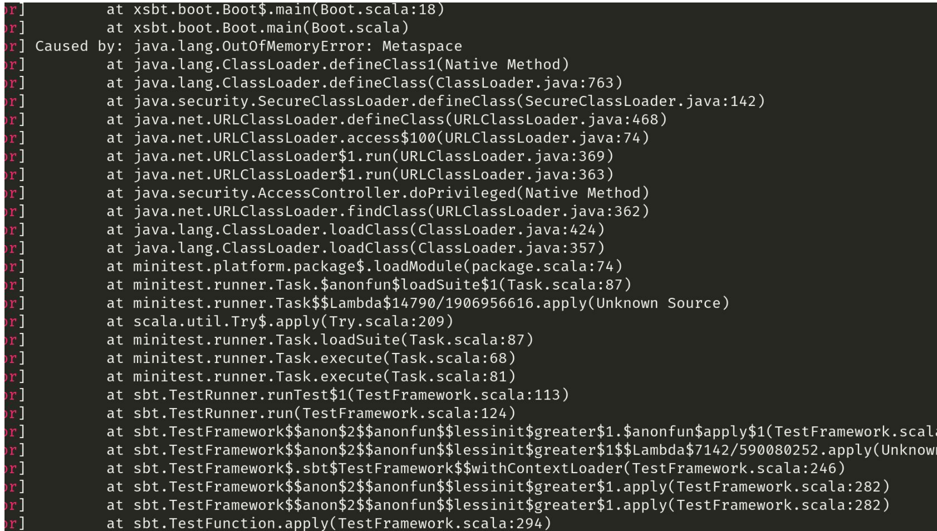
Cross-compiled build with sbt

```scala
// shadow sbt-scalajs' crossProject and CrossType from Scala.js 0.6.x
import sbtcrossproject.CrossPlugin.autoImport.{crossProject, CrossType}
val sharedSettings = Seq(scalaVersion := "2.11.12")

lazy val demo =
  crossProject(JSPlatform, JVMPlatform, NativePlatform)
    .settings(sharedSettings)
    .jsSettings(/* ... */)
    .jvmSettings(/* ... */)
    .nativeSettings(/* ... */)
```

File: build.sbt

Problems with sbt

- Not designed with cross-platform builds in mind
- Slow start-up
- High memory consumption
- Frequent OOMs
- Many concepts to grasp
  - Tasks, graphs, streams, ...
- Convoluted DSL



```
r]         at xsbt.boot.Boot$.main(Boot.scala:18)
r]         at xsbt.boot.Boot.main(Boot.scala)
r] Caused by: java.lang.OutOfMemoryError: Metaspace
r]         at java.lang.ClassLoader.defineClass1(Native Method)
r]         at java.lang.ClassLoader.defineClass(ClassLoader.java:763)
r]         at java.security.SecureClassLoader.defineClass(SecureClassLoader.java:142)
r]         at java.net.URLClassLoader.defineClass(URLClassLoader.java:468)
r]         at java.net.URLClassLoader.access$100(URLClassLoader.java:74)
r]         at java.net.URLClassLoader$1.run(URLClassLoader.java:369)
r]         at java.net.URLClassLoader$1.run(URLClassLoader.java:363)
r]         at java.security.AccessController.doPrivileged(Native Method)
r]         at java.net.URLClassLoader.findClass(URLClassLoader.java:362)
r]         at java.lang.ClassLoader.loadClass(ClassLoader.java:424)
r]         at java.lang.ClassLoader.loadClass(ClassLoader.java:357)
r]         at minitest.platform.package$.loadModule(package.scala:74)
r]         at minitest.runner.Task.$anonfun$loadSuite$1(Task.scala:87)
r]         at minitest.runner.Task$$Lambda$14790/1906956616.apply(Unknown Source)
r]         at scala.util.Try$.apply(Try.scala:209)
r]         at minitest.runner.Task.loadSuite(Task.scala:87)
r]         at minitest.runner.Task.execute(Task.scala:68)
r]         at minitest.runner.Task.execute(Task.scala:81)
r]         at sbt.TestRunner.runTest$1(TestFramework.scala:113)
r]         at sbt.TestRunner.run(TestFramework.scala:124)
r]         at sbt.TestFramework$$anon$2$$anonfun$$lessinit$greater$1.$anonfun$apply$1(TestFramework.scal
r]         at sbt.TestFramework$$anon$2$$anonfun$$lessinit$greater$1$$Lambda$7142/590080252.apply(Unknow
r]         at sbt.TestFramework$.sbt$TestFramework$$withContextLoader(TestFramework.scala:246)
r]         at sbt.TestFramework$$anon$2$$anonfun$$lessinit$greater$1.apply(TestFramework.scala:282)
r]         at sbt.TestFramework$$anon$2$$anonfun$$lessinit$greater$1.apply(TestFramework.scala:282)
r]         at sbt.TestFunction.apply(TestFramework.scala:294)
```

Alternatives?

Bloop is a Scala build server.

Compile, test and run Scala fast.

Bloop

- Build server with focus on performance
- Reads project specification from JSON files
- Comes with sbt plug-in to generate JSON files
- Benefits
  - Supports Scala.js and Scala Native out-of-the-box
  - No start-up time
  - Shorter compilation cycles
  - No OOMs

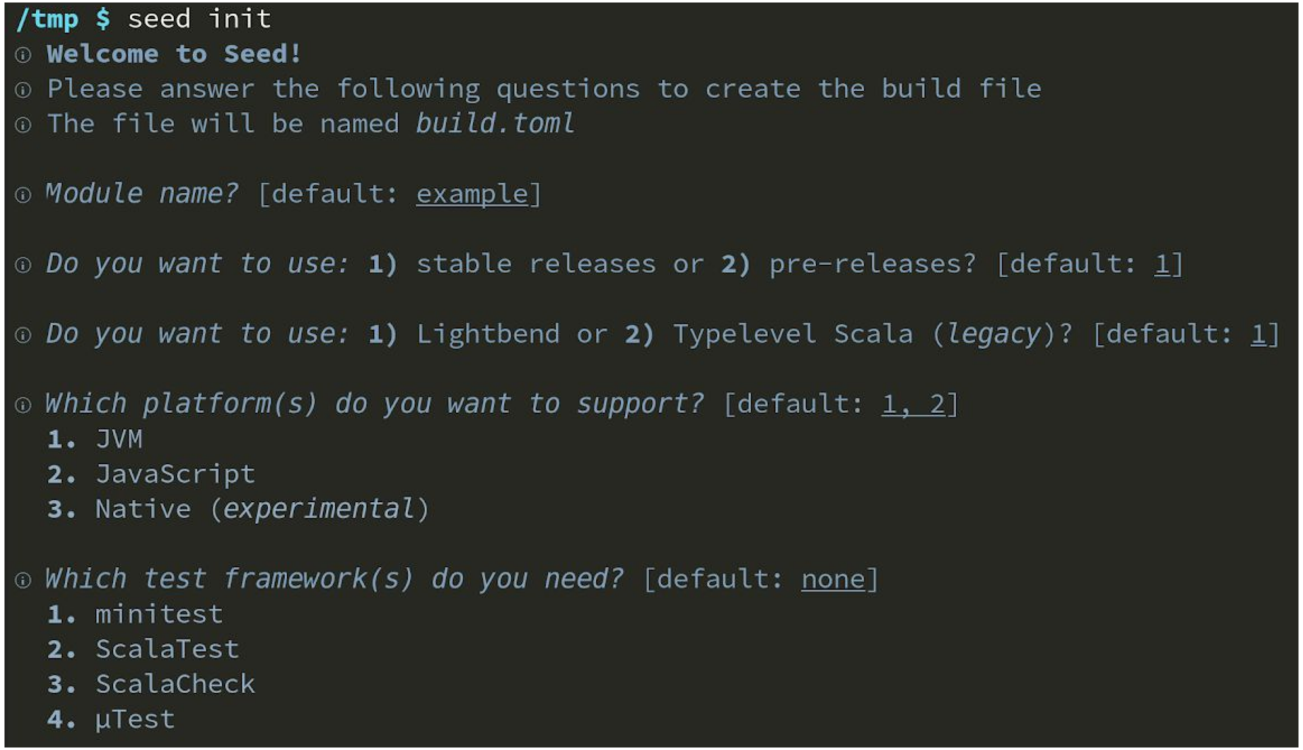
Seed

- Bloop and IDEA configuration generator
- < 10K LOC
- Readable build definitions
    - TOML instead of custom Scala DSL
    - Cross-compiled projects are a first citizen
- Coursier for dependency resolution
- Available as Docker image

https://github.com/tindzk/seed

Seed: Project creation wizard

```
$ seed init
```



```
/tmp $ seed init
ⓘ Welcome to Seed!
ⓘ Please answer the following questions to create the build file
ⓘ The file will be named build.toml

ⓘ Module name? [default: example]

ⓘ Do you want to use: 1) stable releases or 2) pre-releases? [default: 1]

ⓘ Do you want to use: 1) Lightbend or 2) Typelevel Scala (legacy)? [default: 1]

ⓘ Which platform(s) do you want to support? [default: 1, 2]
  1. JVM
  2. JavaScript
  3. Native (experimental)

ⓘ Which test framework(s) do you need? [default: none]
  1. minitest
  2. ScalaTest
  3. ScalaCheck
  4. µTest
```

Minimal cross-compiled project

```
$ tree
.
├── build.toml
└── src
    └── Main.scala
```

Minimal cross-compiled project

```scala
object Main extends App {
  println("Hello World")
}
```

File: src/Main.scala

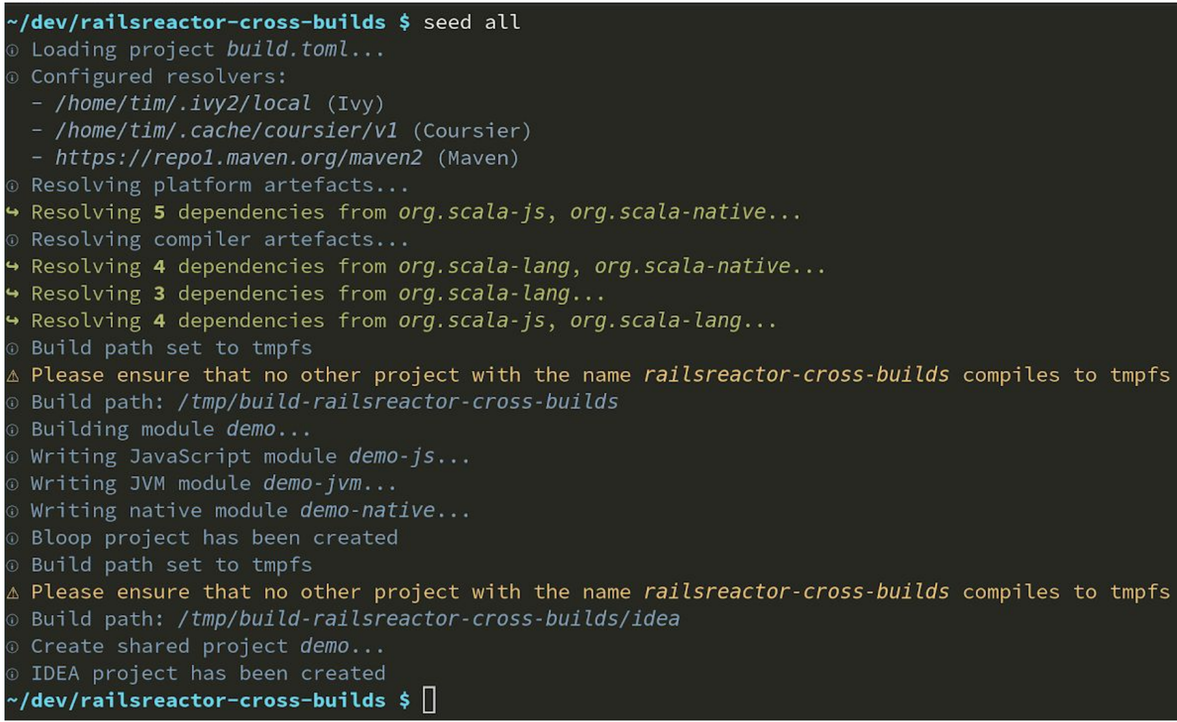
Minimal cross-compiled project

```
[project]
scalaVersion       = "2.13.0"
scalaJsVersion     = "0.6.28"
scalaNativeVersion = "0.3.9"

[module.demo]
root    = "."
targets = ["jvm", "js", "native"]
sources = ["src/"]

[module.demo.native]
scalaVersion = "2.11.12"
```

File: build.toml

Seed: Generate and build project

```
# Create Bloop and IDEA project
$ seed all


# Link and run projects
$ bloop run demo-js
$ bloop run demo-jvm
$ bloop run demo-native
```



```
~/dev/railsreactor-cross-builds $ seed all
ⓘ Loading project build.toml...
ⓘ Configured resolvers:
  - /home/tim/.ivy2/local (Ivy)
  - /home/tim/.cache/coursier/v1 (Coursier)
  - https://repo1.maven.org/maven2 (Maven)
ⓘ Resolving platform artefacts...
↪ Resolving 5 dependencies from org.scala-js, org.scala-native...
ⓘ Resolving compiler artefacts...
↪ Resolving 4 dependencies from org.scala-lang, org.scala-native...
↪ Resolving 3 dependencies from org.scala-lang...
↪ Resolving 4 dependencies from org.scala-js, org.scala-lang...
ⓘ Build path set to tmpfs
⚠ Please ensure that no other project with the name railsreactor-cross-builds compiles to tmpfs
ⓘ Build path: /tmp/build-railsreactor-cross-builds
ⓘ Building module demo...
ⓘ Writing JavaScript module demo-js...
ⓘ Writing JVM module demo-jvm...
ⓘ Writing native module demo-native...
ⓘ Bloop project has been created
ⓘ Build path set to tmpfs
⚠ Please ensure that no other project with the name railsreactor-cross-builds compiles to tmpfs
ⓘ Build path: /tmp/build-railsreactor-cross-builds/idea
ⓘ Create shared project demo...
ⓘ IDEA project has been created
~/dev/railsreactor-cross-builds $ 
```

Create Drone CI pipeline

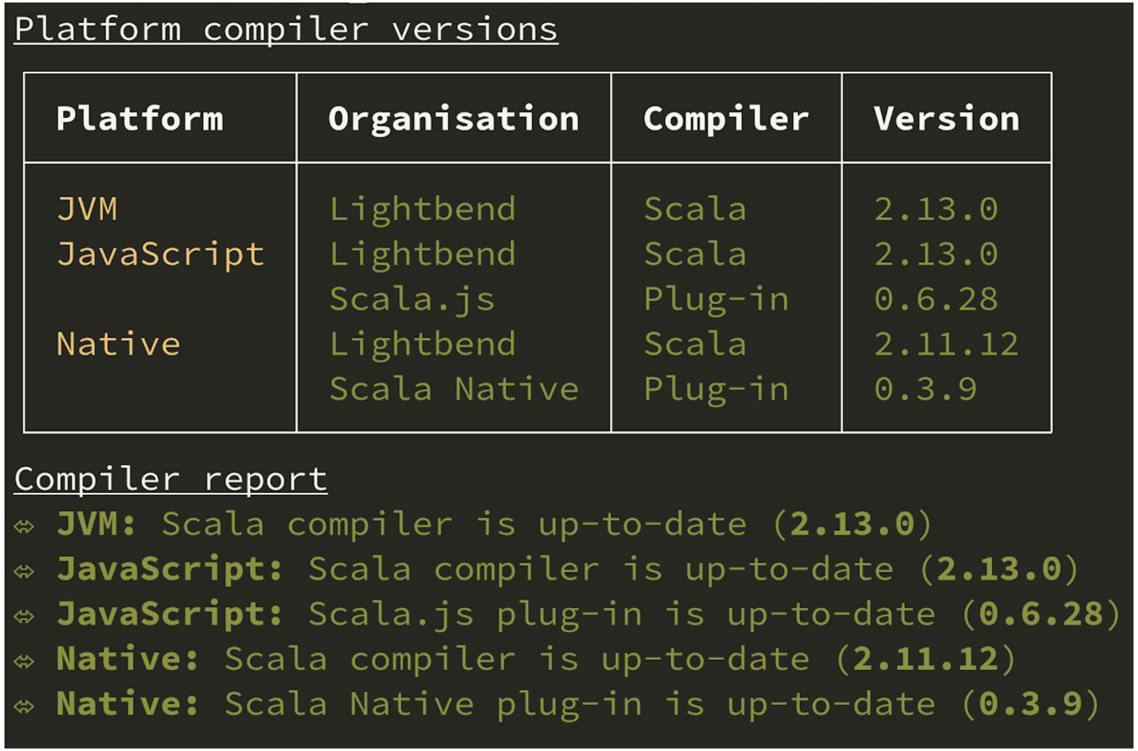
```
kind: pipeline
name: default
steps:
  - name: build
    image: tindzk/seed:0.1.5
    commands:
    - blp-server &
    - seed bloop
    - bloop run demo-js
    - bloop run demo-jvm
    - bloop run demo-native
```

File: .drone.yml

Demo:
http://ci.sparse.tech/tindzk/railsreactor-cross-builds

Seed: Check for updates

```
$ seed update
```



```
Platform compiler versions

┌────────────┬───────────────┬──────────┬─────────┐
│ Platform   │ Organisation  │ Compiler │ Version │
├────────────┼───────────────┼──────────┼─────────┤
│ JVM        │ Lightbend     │ Scala    │ 2.13.0  │
│ JavaScript │ Lightbend     │ Scala    │ 2.13.0  │
│            │ Scala.js      │ Plug-in  │ 0.6.28  │
│ Native     │ Lightbend     │ Scala    │ 2.11.12 │
│            │ Scala Native  │ Plug-in  │ 0.3.9   │
└────────────┴───────────────┴──────────┴─────────┘

Compiler report
⇎ JVM: Scala compiler is up-to-date (2.13.0)
⇎ JavaScript: Scala compiler is up-to-date (2.13.0)
⇎ JavaScript: Scala.js plug-in is up-to-date (0.6.28)
⇎ Native: Scala compiler is up-to-date (2.11.12)
⇎ Native: Scala Native plug-in is up-to-date (0.3.9)
```

Questions?

Thanks!

Code
https://github.com/tindzk/railsreactor-cross-builds

Bloop
https://scalacenter.github.io/bloop/

Seed
https://github.com/tindzk/seed

```
“Perfection is achieved, not
when there is nothing more to
add, but when there is nothing
left to take away.”

-- Antoine de Saint-Exupéry
```