Scala goes Native

by Denys Shabalin

EPFL

Prologue

Pascal

My first programming experience.

Verbose.

Low-level.

Can’t immediately shoot yourself.

Seriously, it’s really verbose.

C

I can type code so much faster now.

Low-level.

Oh, my foot is right over there.

Wait, where is the other one.

C#

You’re a class, and you, and you…

Structs and pointers are still there.

First experience with modern IDEs.

Every single thing is documented.

Python

Look mom, no types.

Productive to prototype.

Pain to maintain.

We don’t need no threads.

Can go low-level with C API.

Scala

Is it typed? It’s typed.

Where did all the annotations go?

Is it fast? Yes, just wait for it.

Is it parallel? Oh, yes it is.

Chapter 1:

The golden cage

It had four walls, food and water

It’s fast. Once warmed up.

It’s fast. If JIT likes your code.

It’s fast. If you don’t stress the GC.

It’s fast…

And it was safe

Perhaps too safe.

Can’t escape the sandbox.

Escape hatches are for the masters.

They know better.

Don’t talk to strangers

Let’s make talking to others hard.

As hard as possible.

No, even harder.

They will just stay here, it’s safe.

Chapter 2:

Daydreaming

Immediate

Your code runs right away.

No warm-up.

No interpretation.

Family values

It’s nice not to allocate every thing.

Passing values around is cheap.

Stack is your friend.

Forgetful

Memory, under your control.

Sometimes, you just know better.

GC heap is just an option.

You choose.

Talking to others is easy

Calling native code should be easy.

And not incur any overhead.

Chapter 3:

Waking up

Chapter 4:

9 months later

Chapter 5:

The one with the answers

Based on LLVM? ....

Can it optimise tail calls? ....

Same language? ....

Just a backend? ....

Which front-end? ....

Garbage collected?................

Hardware support? ....

Library support? ....

Open source? ....

When? ....

....

....

Based on LLVM?

Yes.

Can it optimise tail calls?

Yes, even the mutual ones.

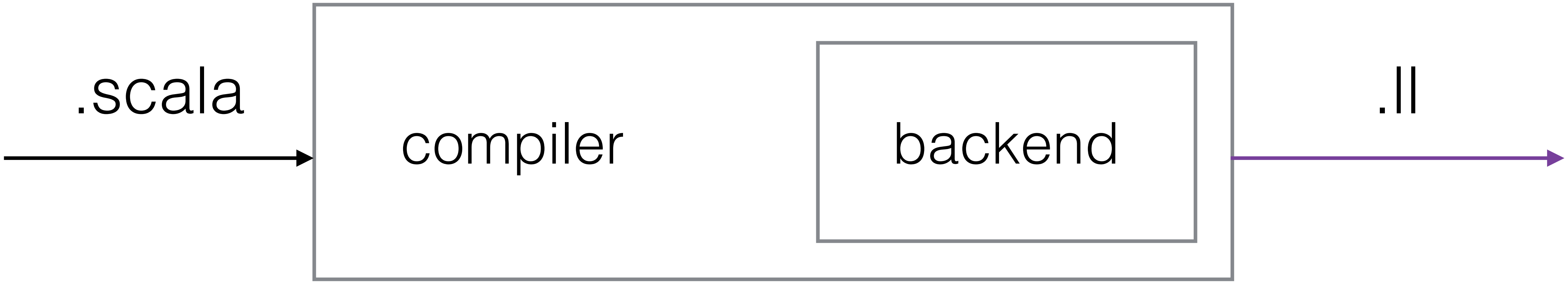
Same language?

Yes, mostly.

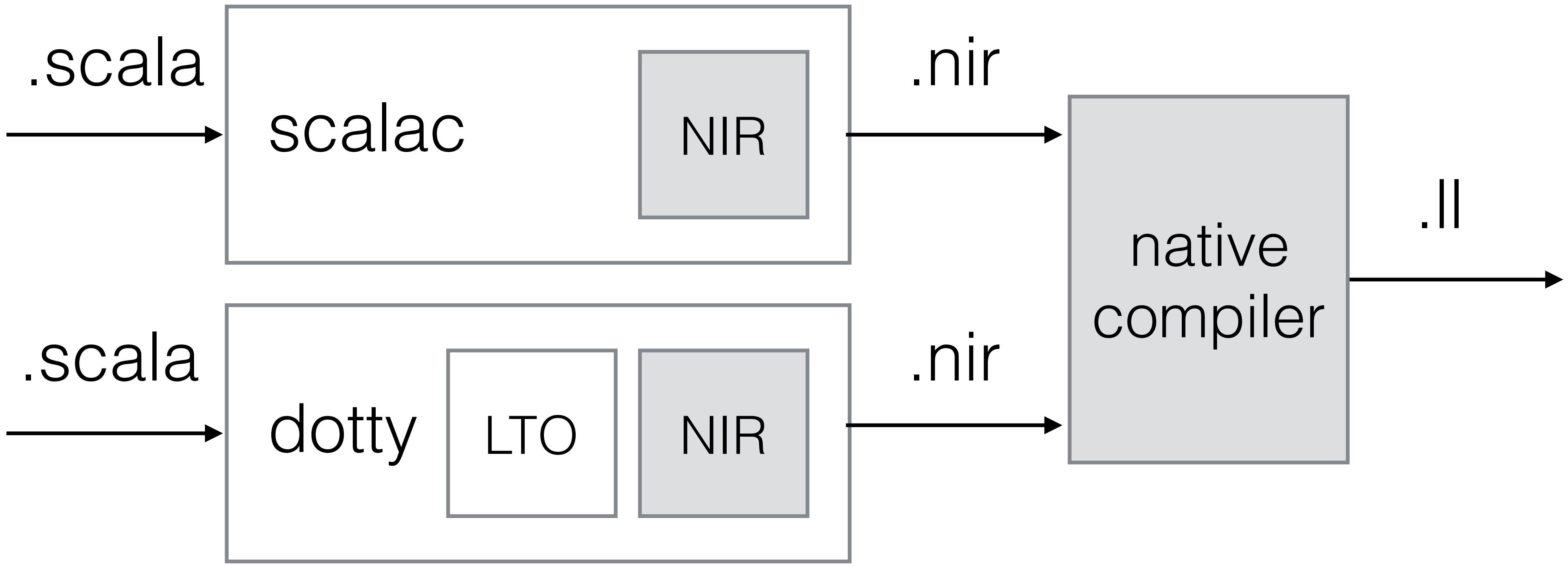
With extra low-level primitives,

and better native interoperability.

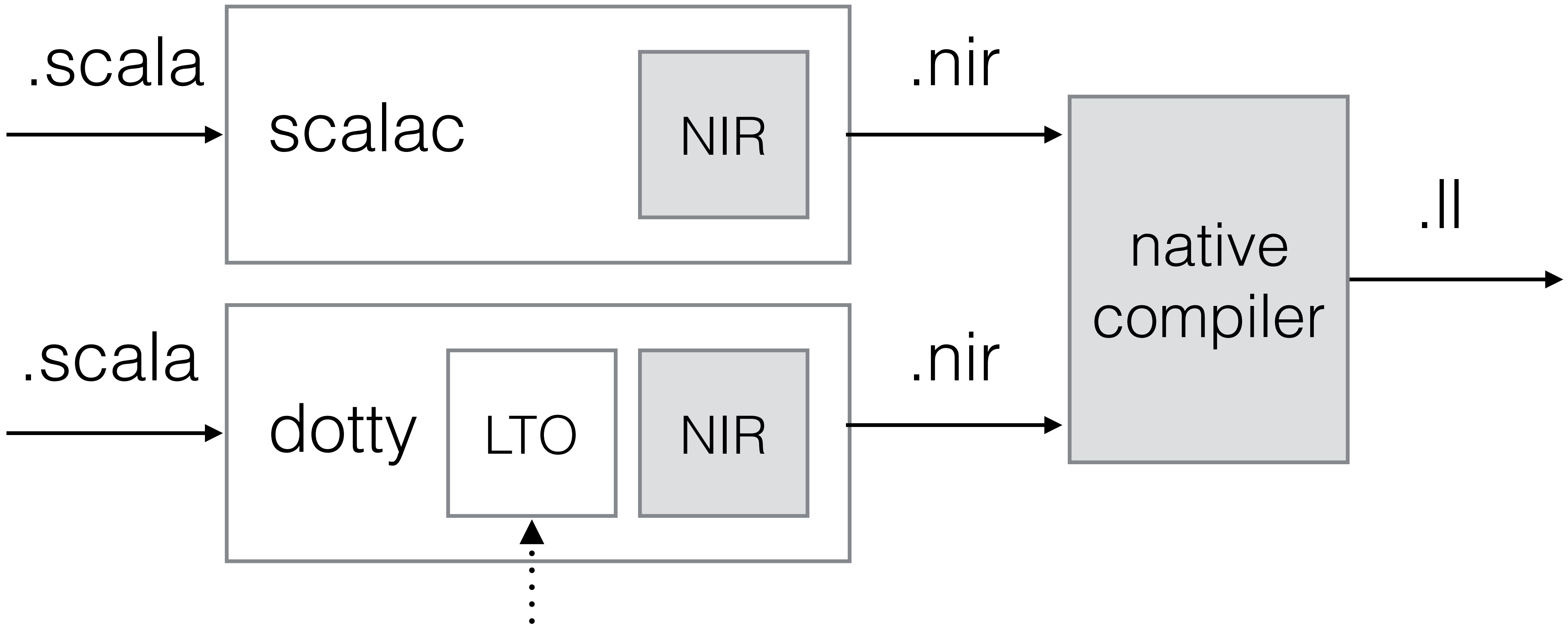
Just a backend? Not quite.



Which front-end? All of them.



Which front-end? All of them.



github.com/dotty-linker by @darkdimius

Garbage-collected?

Yes.

First release will use Boehm GC.

Can only get better from there.

Hardware support?

Developed on 64-bit Intel.

People expressed interest in

{ARM, Intel} x {32-bit, 64-bit}.

Library support?

Subset of java.* together with

C’s stdlib bindings are built-in.

Open source?

Yes.

https://github.com/scala-native

When?

When it's ready.

http://twitter.com/scala_native

The end