Meta-programming with KSP and Kotlin compiler plugins

Tadeas Kriz

@TadeasKriz

KotlinConf'23

Amsterdam

Metaprogramming

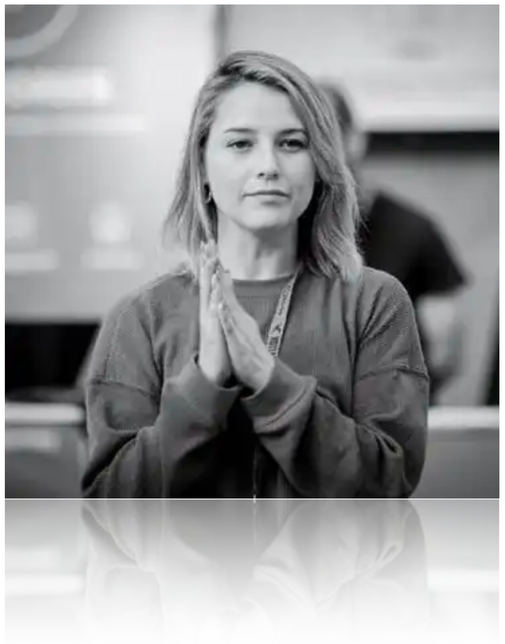
with KSP and Kotlin compiler plugins

Tadeas Kriz

Senior Mobile Developer @ Touchlab

```
val me = "Tadeas Kriz"
```

Great Scott!



Crash Course on the Kotlin Compiler

Amanda Hinchman-Dominguez

Apr 13, 11:15 - 12:00 @ Berlage Zaal

Agenda

Agenda

- How to use KSP
- How to write Kotlin compiler plugins
- How to choose which one to use

Agenda

- JVM meta-programming recap
- KSP overview
- Compiler plugins overview
- How to choose which one to use

Meta programming

Reflection

Lower performance

Limited in native

Compile-time Introspection

Longer compile times

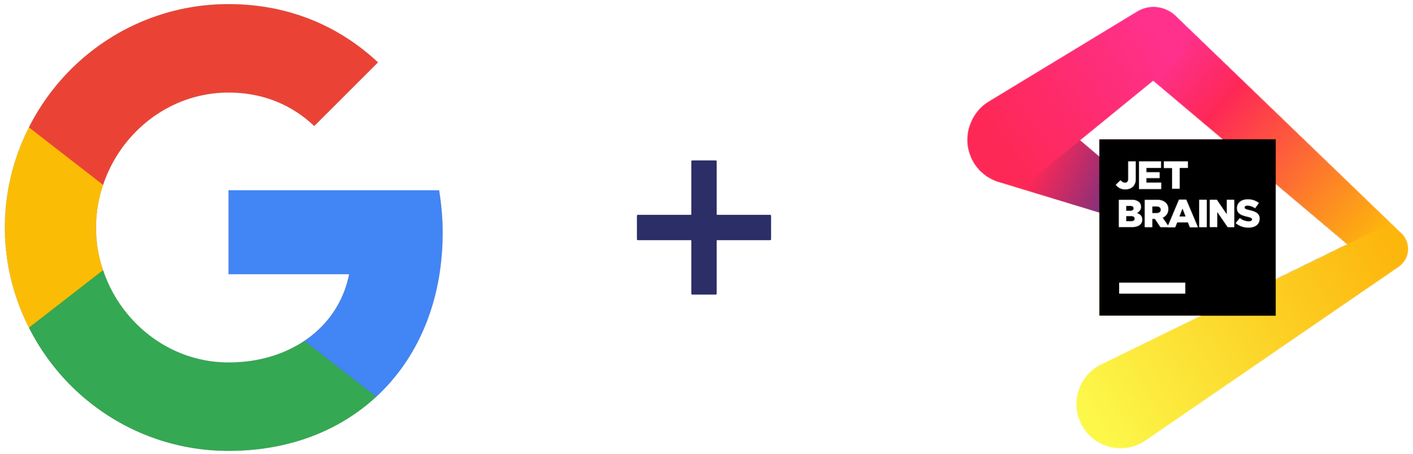
Larger binaries

Necessity in Kotlin Multiplatform

apt

kapt

KSP


JET
BRAINS

Faster than kapt

Stable processor API

Supports Java sources

Well documented

Moshi

Auto Dagger

Room

Kotshi

DeeplinkDispatch

Glide

Lyricist

Growing support

EasyAdapter

Epoxy

Paris

Koin Annotations

gRPC Dekorator

RxHttp

Lich SavedState

https://kotlinlang.org/docs/ksp-overview.html#supported-libraries

No code modifications

No expression level introspection

KSP quickstart

KSP quickstart

https://kotlinlang.org/docs/ksp-quickstart.html

Kotlin Compiler Plugins

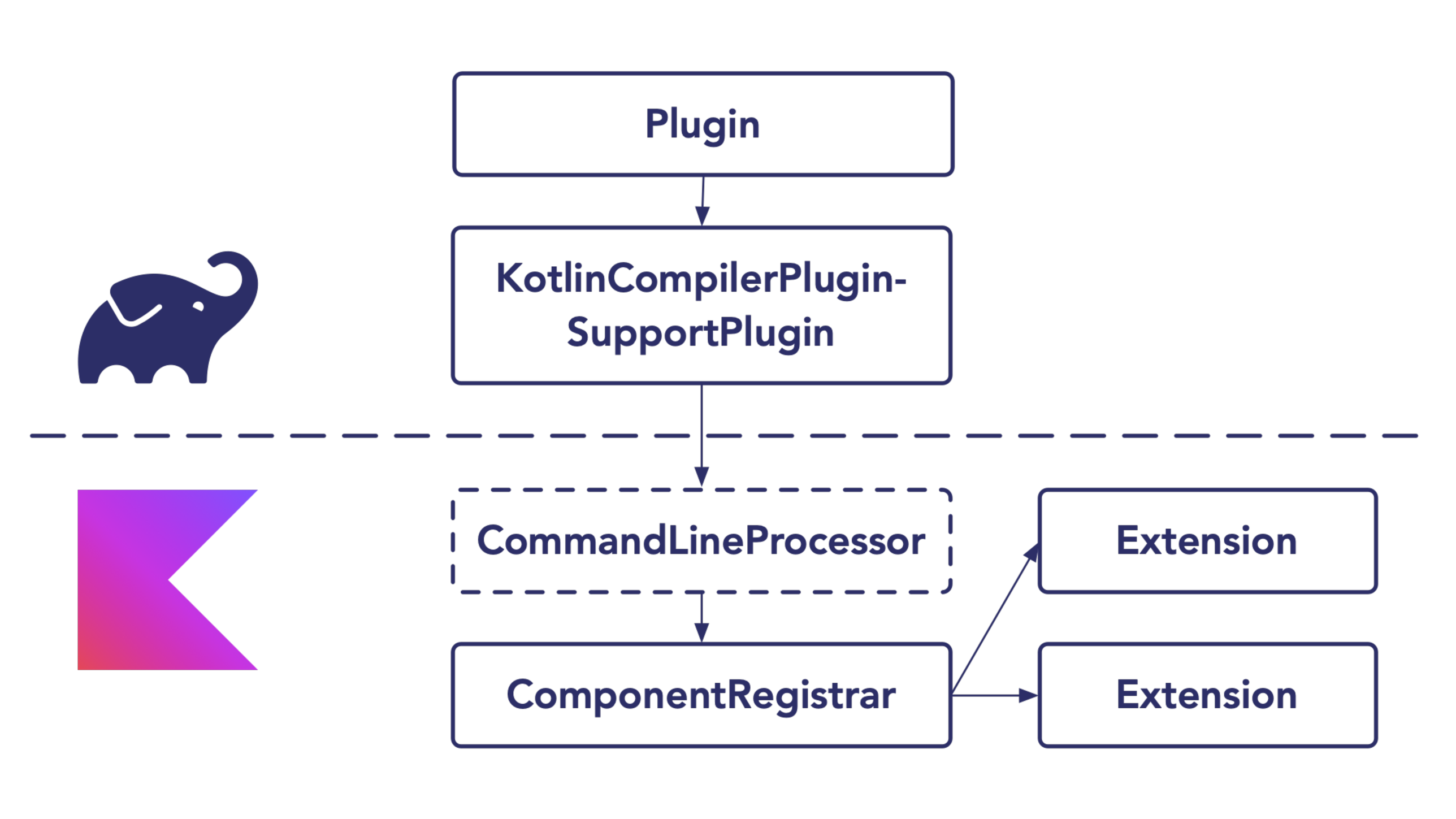
Compiler plugins ❤️

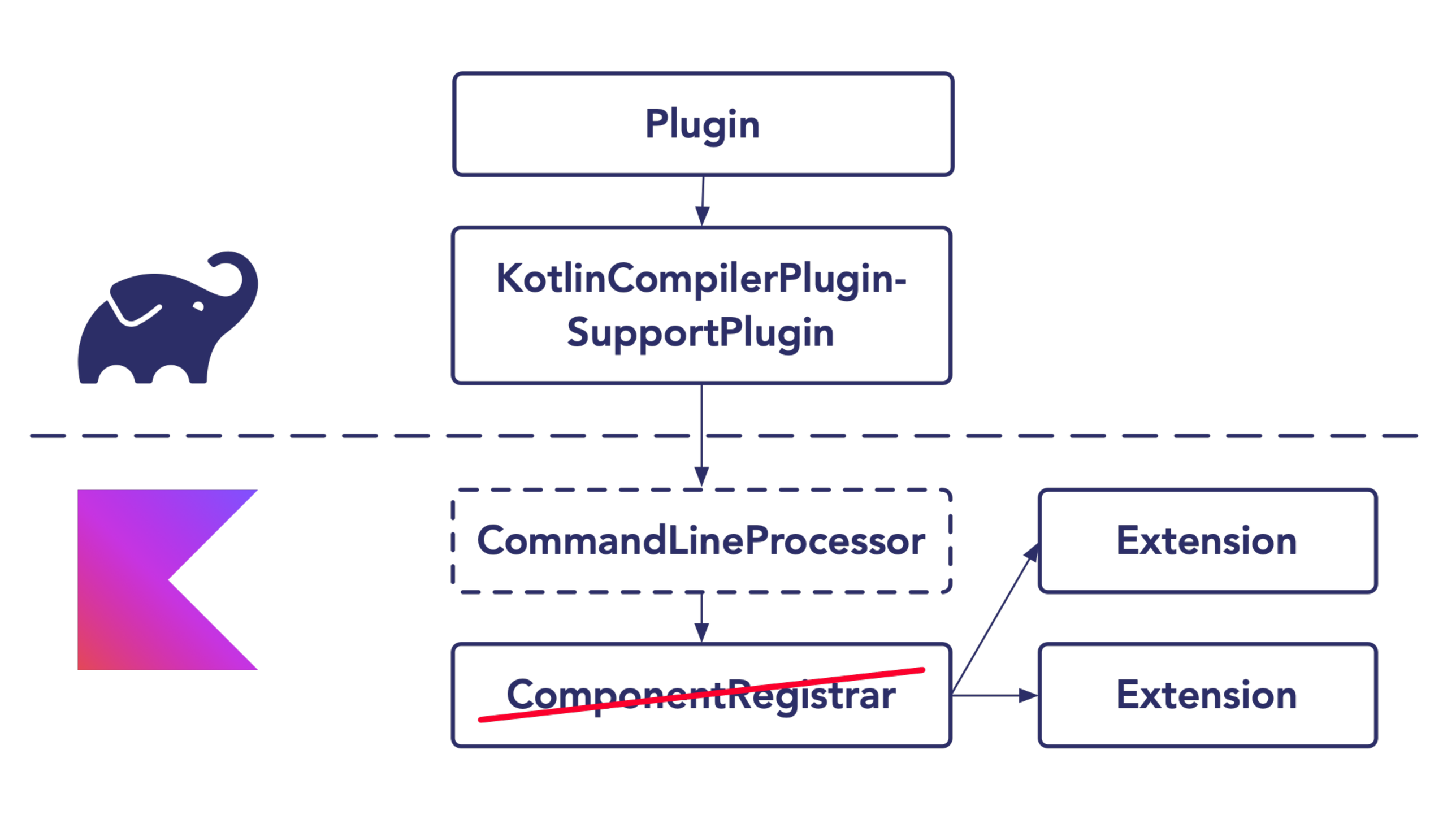
- Extremely powerful
- IR transformations
- Synthetic declarations
- Expression-level validation

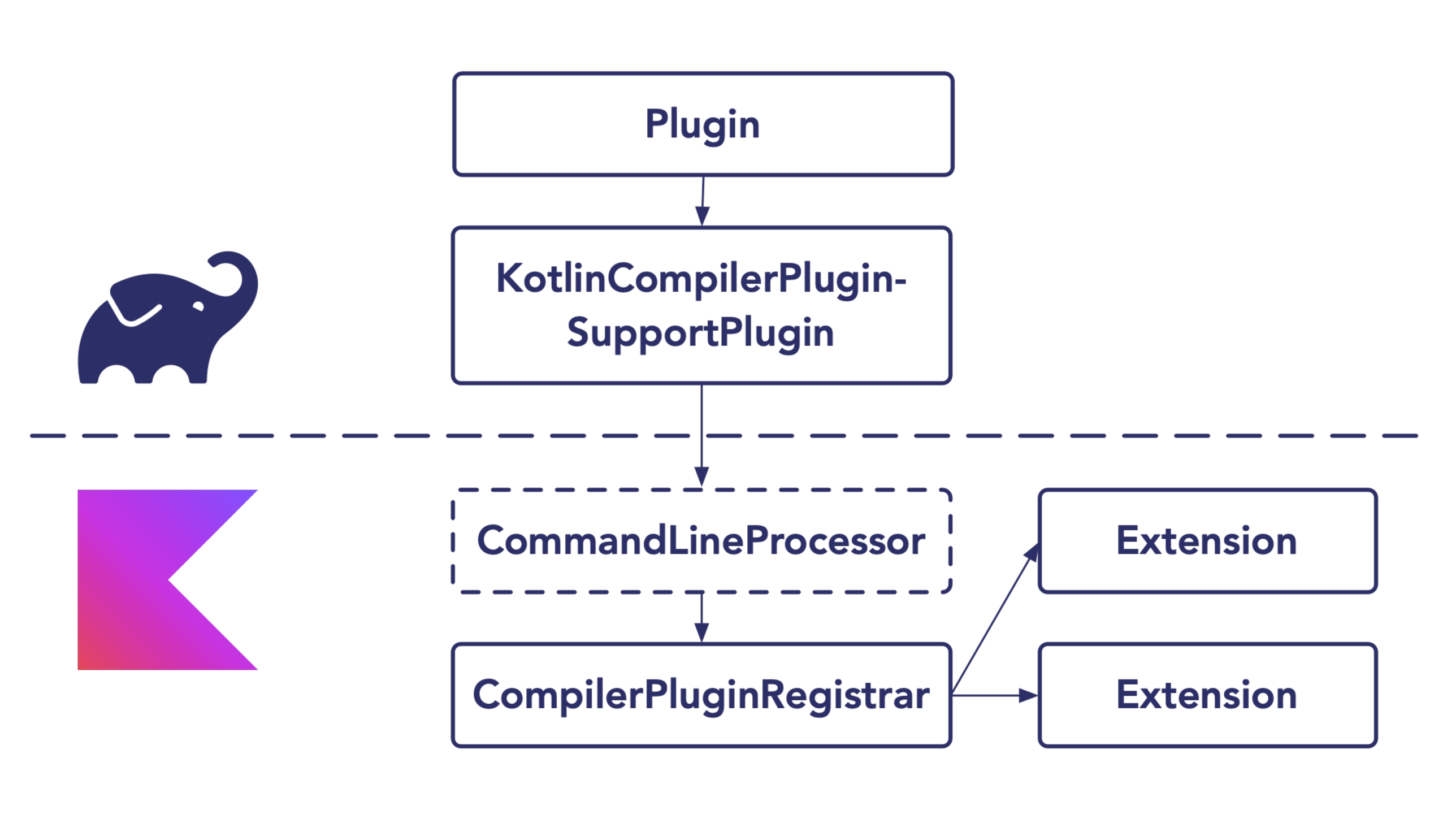
Compiler plugins 💔

- Little to no documentation
- Complex setup
- Unstable API
- May require IntelliJ IDEA plugin

Plugin structure


Plugin
KotlinCompilerPlugin-
SupportPlugin
CommandLineProcessor
ComponentRegistrar
Extension
Extension


Plugin
KotlinCompilerPlugin-
SupportPlugin
CommandLineProcessor
ComponentRegistrar
Extension
Extension


Plugin
KotlinCompilerPlugin-
SupportPlugin
CommandLineProcessor
CompilerPluginRegistrar
Extension
Extension

Extensions 😱

```
fun registerPluginExtensionPoints(project: MockProject)
```

```
SyntheticResolveExtension
SyntheticJavaResolveExtension
ClassFileFactoryFinalizerExtension
PackageFragmentProviderExtension
DeclarationAttributeAltererExtension
JsSyntheticTranslateExtension
CollectAdditionalSourcesExtension
ExpressionCodegenExtension
ShellExtension
CandidateInterceptor
FirExtensionRegistrarAdapter
```

```
IrGenerationExtension
ClassBuilderInterceptorExtension
AnalysisHandlerExtension
StorageComponentContainerContributor
PreprocessedVirtualFileFactoryExtension
CompilerConfigurationExtension
ExtraImportsProviderExtension
ScriptEvaluationExtension
TypeResolutionInterceptor
DescriptorSerializerPlugin
TypeAttributeTranslatorExtension
```

Extensions 🧐

```
fun registerPluginExtensionPoints(project: MockProject)
```

SyntheticResolveExtension

SyntheticJavaResolveExtension

ClassFileFactoryFinalizerExtension

PackageFragmentProviderExtension

DeclarationAttributeAltererExtension

JsSyntheticTranslateExtension

CollectAdditionalSourcesExtension

ExpressionCodegenExtension

ShellExtension

CandidateInterceptor

FirExtensionRegistrarAdapter

IrGenerationExtension

ClassBuilderInterceptorExtension

AnalysisHandlerExtension

StorageComponentContainerContributor

PreprocessedVirtualFileFactoryExtension

CompilerConfigurationExtension

ExtraImportsProviderExtension

ScriptEvaluationExtension

TypeResolutionInterceptor

DescriptorSerializerPlugin

TypeAttributeTranslatorExtension

SyntheticResolveExtension

- Add synthetic declarations
  - Create new classes
  - Extend existing classes
- Needed for declarations to be visible from code
- Needs IntelliJ IDEA plugin for code completion

IrGenerationExtension

- Add and modify declarations
- Implement synthetic declarations
- Modify IR down to expression level

My approach

Crack open Kotlin

Setting up

- `git clone git@github.com:JetBrains/kotlin.git`
- Open in IntelliJ IDEA
- Create `local.properties` file

  ```
  kotlin.build.isObsoleteJdkOverrideEnabled=true
  kotlin.native.enabled=true
  ```

- Update `gradle.properties` file 😬

  ```
  org.gradle.dependency.verification=lenient
  ```

- Synchronize Gradle

More setup 😭

- In KMM project, run Kotlin/Native Gradle task with **`--info`**

  `./gradlew compileKotlinIosArm64 --info`

- Look for **Transformed arguments = [konanc ...]**
  - Copy everything except konanc
- To local.properties add:

  konan.debug.args=-Xverbose-phases ALL {arguments}

Poking around 👈👆👇👉

- Add breakpoints
- In Kotlin project run in debug:

  :kotlin-native:dist :kotlin-native:backend.native:debugCompiler
- ??
- Profit

Hard-learned lessons

IntelliJ IDEA won't stop on breakpoints

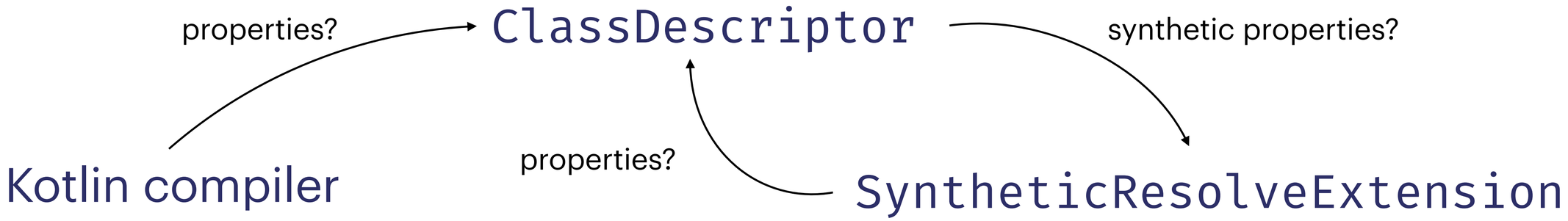
IntelliJ IDEA won't stop on breakpoints

```
kotlin.compiler.execution.strategy=in-process
```

Crashes in SyntheticResolveExtension

Crashes in SyntheticResolveExtension

Class descriptors are lazily resolved



Kotlin/Native is STRICT

Kotlin/Native is STRICT

- Calling generic methods requires passing generic types
- IR declarations require correct start and end offsets

Maintenance is HARD

Maintenance is HARD

- Kotlin releases change API
- Supporting multiple Kotlin versions

Would you like to know more?

Would you like to know more?

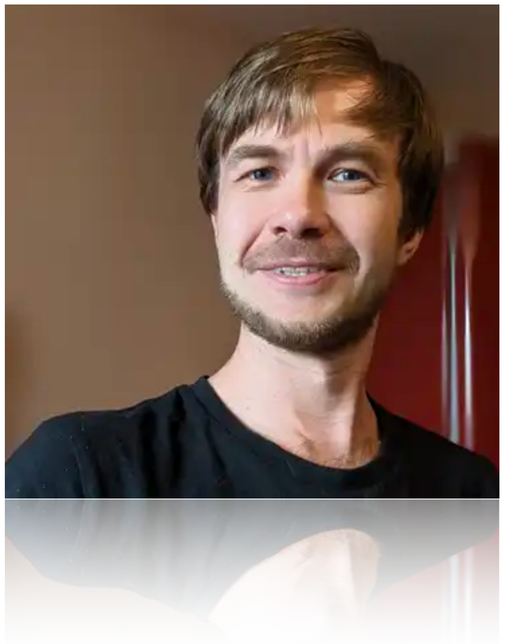
- Kotlin repository in the **plugins** directory
- AndroidX repository in **compose/compiler** directory
- KSP repository in **compiler-plugin** directory
- Zipline repository in **zipline-kotlin-plugin** directory
- Hyperdrive repository in **plugin**, **krpc/plugin** and **multiplatformx/plugin**

K2?

No.

No… but!

No... but!

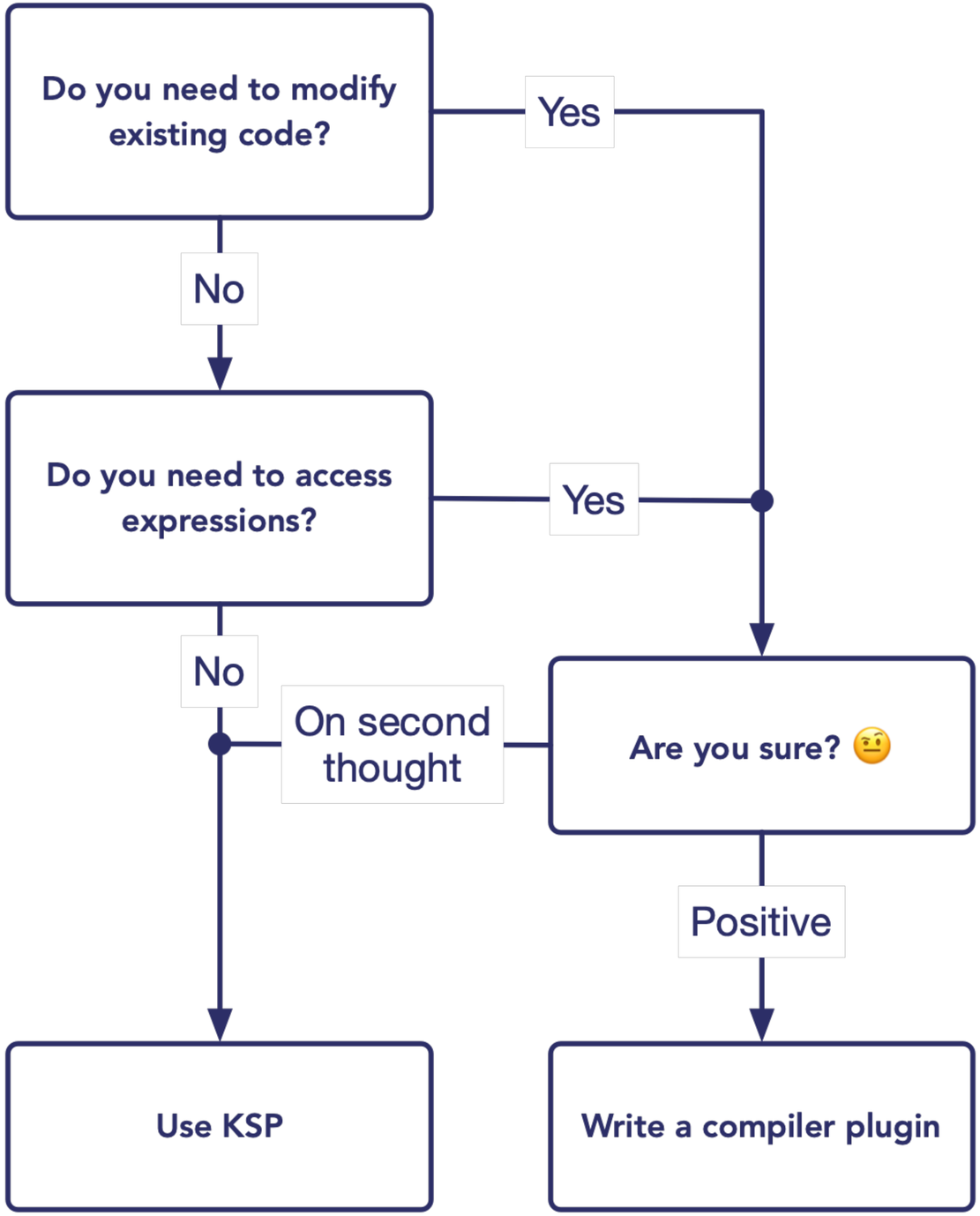


K2 Compiler plugins

MIKHAIL GLUKHIKH

Apr 14, 10:15 - 11:00 @ Berlage Zaal

Follow the diagram



Testing?

Kotlin Compile Testing

https://github.com/tschuchortdev/kotlin-compile-testing

Supports KSP!

https://github.com/tschuchortdev/kotlin-compile-testing#kotlin-symbol-processing-api-support

Slow

Kotlin Multiplatform

Serialization

Dependency Injection

Jetpack Compose

Zipline

https://github.com/cashapp/zipline

SKIE

"Swift-friendly" API

Get a SKIE Demo!

https://skie.touchlab.co

Recap

Recap

- Stop using **kapt**
- **Prefer KSP** where possible
- Compiler plugins - **powerful**, but **costly**
- Future is **bright**?

Questions?

Get in touch!

@TadeasKriz

Get in touch!

@TadeasKriz@mastodon.social

Thank you, and don’t forget to vote

@TadeasKriz

KotlinConf’23

Amsterdam