

# KOTLIN ADOPTION @ SCALE

---

Sergei Rybalkin & Sergey Ryabov, Facebook

# TYPICAL PROJECT



# TYPICAL PROJECT

---

- ◆ Tens/hundreds modules

# TYPICAL PROJECT

---

- ◆ Tens/hundreds modules
- ◆ Hundreds thousands lines of code

# TYPICAL PROJECT

---

- ◆ Tens/hundreds modules
- ◆ Hundreds thousands lines of code
- ◆ Tens of developers

# ADOPTION IN A TYPICAL PROJECT



# ADOPTION IN A TYPICAL PROJECT

---

- ◆ **Write** code in Android Studio

# ADOPTION IN A TYPICAL PROJECT

- ◆ **Write** code in Android Studio
- ◆ Add Kotlin Gradle plugin and **Build** with Gradle



# ADOPTION IN A TYPICAL PROJECT

- ◆ **Write** code in Android Studio
- ◆ Add Kotlin Gradle plugin and **Build** with Gradle
- ◆ **Ship** slightly bigger APK to Play Store

# REALLY BIG PROJECT



# REALLY BIG PROJECT

---

- ◆ Hundreds of thousands modules

# REALLY BIG PROJECT

---

- ◆ Hundreds of thousands modules
- ◆ Tens of millions of lines of code

# REALLY BIG PROJECT

- ◆ Hundreds of thousands modules
- ◆ Tens of millions of lines of code
- ◆ Thousands developers

# ADOPTION AT SCALE



# ADOPTION AT SCALE

---

◆ **Write** code in Android Studio?

# ADOPTION AT SCALE

---

- ◆ **Write** code in Android Studio?
- ◆ **Build** with Gradle?



# ADOPTION AT SCALE

- ◆ **Write** code in Android Studio?
- ◆ **Build** with Gradle?
- ◆ **Ship** slightly(?) bigger APK to Play Store

**WRITE CODE**



# WRITE CODE

---

# WRITE CODE

◆ IDE

# WRITE CODE

- ◆ IDE
- ◆ Tools: formatters, linters, static analysers

# WRITE CODE

- ◆ IDE
- ◆ Tools: formatters, linters, static analysers
- ◆ Libraries

# IDE: PROBLEMS



# IDE: PROBLEMS

- ◆ Facebook has a monorepo with 100k+ modules



# IDE: PROBLEMS

- ◆ Facebook has a monorepo with 100k+ modules
- ◆ Non-typical IDE issues pop up at this scale

# IDE: PROBLEMS

- ◆ Facebook has a monorepo with 100k+ modules
- ◆ Non-typical IDE issues pop up at this scale
- ◆ Android Studio can have 1k+ Java modules focused

# IDE: PROBLEMS

- ◆ Facebook has a monorepo with 100k+ modules
- ◆ Non-typical IDE issues pop up at this scale
- ◆ Android Studio can have 1k+ Java modules focused
- ◆ But only several hundred Kotlin modules – **too few**

# IDE: PROBLEMS

- ◆ Facebook has a monorepo with 100k+ modules
- ◆ Non-typical IDE issues pop up at this scale
- ◆ Android Studio can have 1k+ Java modules focused
- ◆ But only several hundred Kotlin modules – **too few**
  - Some issues with Kotlin IDE Plugin

Annotations  
ore  
ore-kotlin  
ain  
st  
kotlin  
com  
facebook  
litho [fbandroid\_libraries\_compo  
testing  
ViewTesting.kt  
BUCK  
CommonStylesTest  
KCachedTest  
KEventHandlerTest  
KStateTest  
KTreePropsTest.kt  
StyleTest

```
6 load(  
7     "//tools/build_defs/oss:litho_defs.bzl",  
8     "LITHO_ANDROIDSUPPORT_TESTING_CORE_TARGET",  
9     "LITHO_ASSERTJ_TARGET",  
10    "LITHO_BUILD_CONFIG_TARGET",  
11    "LITHO_JUNIT_TARGET",  
12    "LITHO_KOTLIN_TARGET",  
13    "LITHO_RENDERERCORE_TESTING_TARGET",  
14    "LITHO_ROBOLECTRIC_V4_TARGET",  
15    "LITHO_SOLOADER_TARGET",  
16    "LITHO_WIDGET_KOTLIN_TARGET",  
17    "LITHO_YOGA_TARGET",  
18    "litho_roboelectric4_test",  
19    "make_dep_path",  
20 )  
21  
22 ► litho_roboelectric4_test(  
23     name = "litho",  
24     srcs = glob(["**/*.kt"]),  
25     contacts = ["oncall+components_for_android@xmail.facebook.com"],  
26     is_androidx = True,  
27     language = "KOTLIN",  
28     provided_deps = [  
29         LITHO_ROBOLECTRIC_V4_TARGET,  
30     ],  
31     source = "8",  
32     target = "8",  
33     deps = [  
34         LITHO_ANDROIDSUPPORT_TESTING_CORE_TARGET,  
35         LITHO_ASSERTJ_TARGET,  
36         LITHO_BUILD_CONFIG_TARGET,  
37         LITHO_JUNIT_TARGET,  
38         LITHO_KOTLIN_TARGET,
```

Annotations  
ore  
ore-kotlin  
ain  
st  
kotlin  
com  
facebook  
litho [fbandroid\_libraries\_compo  
testing  
ViewTesting.kt  
BUCK  
CommonStylesTest  
KCachedTest  
KEventHandlerTest  
KStateTest  
KTreePropsTest.kt  
StyleTest

```
6 load(  
7     "//tools/build_defs/oss:litho_defs.bzl",  
8     "LITHO_ANDROIDSUPPORT_TESTING_CORE_TARGET",  
9     "LITHO_ASSERTJ_TARGET",  
10    "LITHO_BUILD_CONFIG_TARGET",  
11    "LITHO_JUNIT_TARGET",  
12    "LITHO_KOTLIN_TARGET",  
13    "LITHO_RENDERERCORE_TESTING_TARGET",  
14    "LITHO_ROBOLECTRIC_V4_TARGET",  
15    "LITHO_SOLOADER_TARGET",  
16    "LITHO_WIDGET_KOTLIN_TARGET",  
17    "LITHO_YOGA_TARGET",  
18    "litho_robolectric4_test",  
19    "make_dep_path",  
20 )  
21  
22 ► litho_robolectric4_test(  
23     name = "litho",  
24     srcs = glob(["**/*.kt"]),  
25     contacts = ["oncall+components_for_android@xmail.facebook.com"],  
26     is_androidx = True,  
27     language = "KOTLIN",  
28     provided_deps = [  
29         LITHO_ROBOLECTRIC_V4_TARGET,  
30     ],  
31     source = "8",  
32     target = "8",  
33     deps = [  
34         LITHO_ANDROIDSUPPORT_TESTING_CORE_TARGET,  
35         LITHO_ASSERTJ_TARGET,  
36         LITHO_BUILD_CONFIG_TARGET,  
37         LITHO_JUNIT_TARGET,  
38         LITHO_KOTLIN_TARGET,
```

# IDE: HOW TO DETECT?

# IDE: HOW TO DETECT?

- ◆ IDE profiling



# IDE: HOW TO DETECT?

- ◆ IDE profiling
- ◆ Custom analytics & tracing

# IDE: HOW TO DETECT?

- ◆ IDE profiling
- ◆ Custom analytics & tracing
- ◆ Internal fork to iterate fast

# TOOLBAR ICONS FREEZE



**Sergei Rybalkin**  
@lightdelay

В IntelliJ Idea скоро будет отменен налог на роскошь!  
Можно будет не платить за использование Touch Bar на Mac

❤️ @dolzhenko



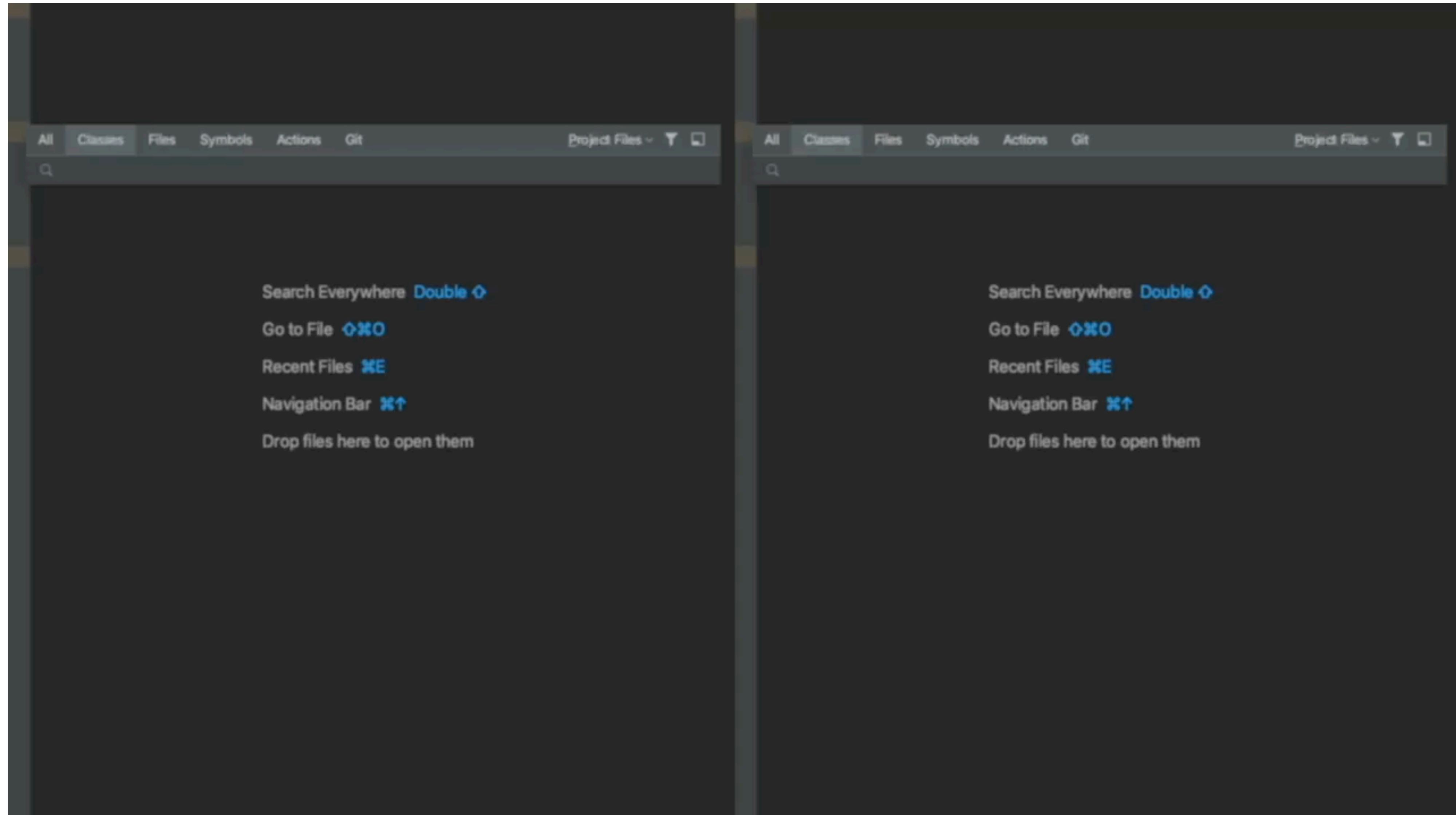
**IntelliJ IDEA**  
Capable and Ergonomic  
Java IDE

JET  
BRAINS

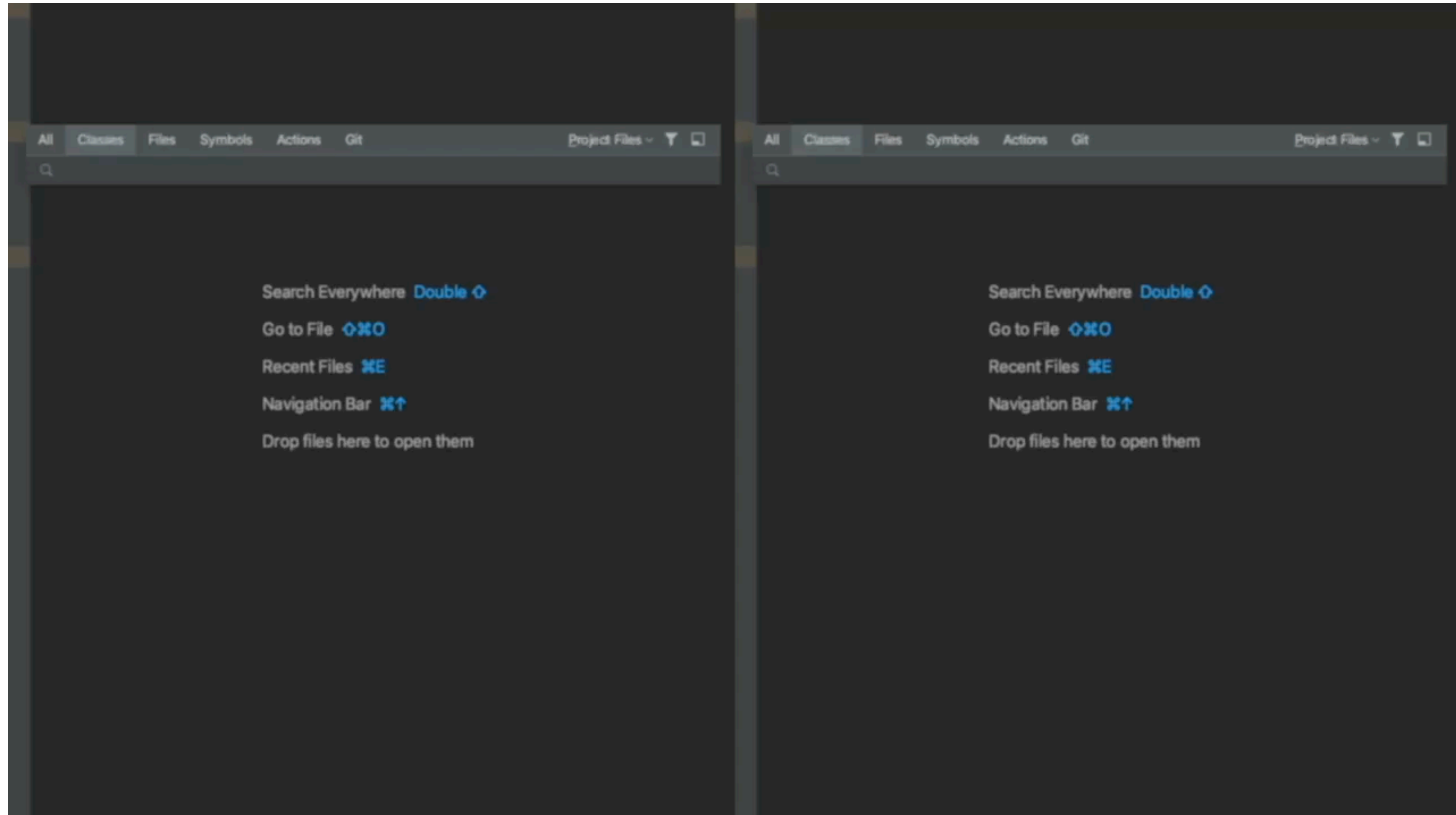
IDEA-225039: Add registry key to force disable touchbar · JetBrains/int...  
GitOrigin-RevId: de958c5c98e0e28a85d26ee19b4c94c1d8c0ed3d  
[github.com](#)

10:50 AM · Jan 19, 2021

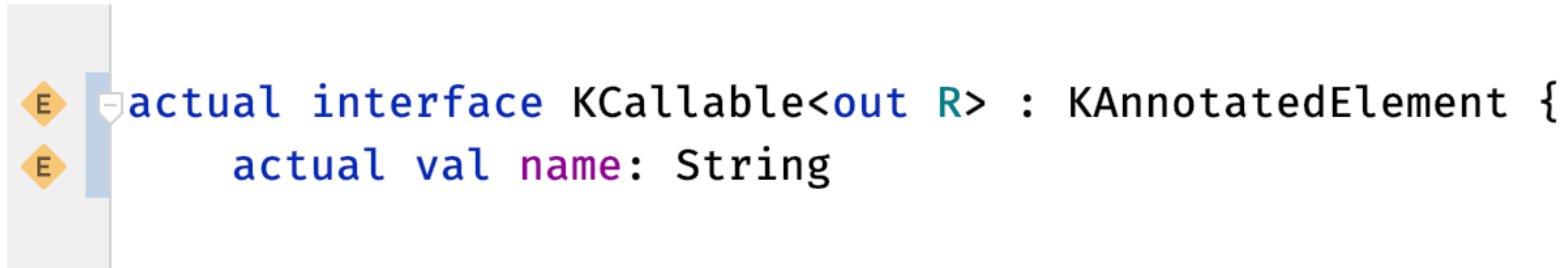
# TOOLBAR ICONS FREEZE



# TOOLBAR ICONS FREEZE



# EXPECT/ACTUAL GUTTER ICON FREEZE



## Kotlin

- DSL markers
- Implemented declaration
- Implementing declaration
- Multiplatform actual declaration
- Multiplatform expect declaration
- Overridden declaration
- Overriding declaration

# TOOLS

---

# TOOLS: LINTERS



# TOOLS: LINTERS

---

- ◆ Code style

# TOOLS: LINTERS

- ◆ Code style
- ◆ Performance

# TOOLS: LINTERS

- ◆ Code style
- ◆ Performance
- ◆ Safety

# TOOLS: LINTERS

- ◆ Code style
- ◆ Performance
- ◆ Safety
- ◆ Race-conditions

# TOOLS: LINTERS

# TOOLS: LINTERS

46

```
resourceResolver.resolveStringRes(id, formatArgs)!!
```



Lint (ANDROIDLINT) raised a **UnsafeDereference** warning on **line 46**



Please avoid using the !! operator. For ideas and options see: <https://fburl.com/kotlin-null-safety>



File New Task



# TOOLS: LINTERS

46

```
resourceResolver.resolveStringRes(id, formatArgs)!!
```



Lint (ANDROIDLINT) raised a **UnsafeDereference** warning on **line 46**

Please avoid using the `!!` operator. For more information, see <https://developer.android.com/studio/write/lint>.

File New Task



infer commented on **line 18**

There may be a **Execution Time Complexity Increase**: Time complexity of constructor has **increased** from `O(1)` to `O(points.length)`. (error trace: TV115306648)  
Feedback: complexity increase is [good catch | expected | wrong].

Sunday 10:20am • Like • Reply • Restore • Formatted

19

```
pointsToWait = new HashSet<>();
```

20

```
this.pointsToWait.addAll(Arrays.asList(points));
```

# TOOLS: LINTERS

46

```
resourceResolver.resolveStringRes(id, formatArgs)!!
```



Lint (ANDROIDLINT) raised a **UnsafeDereference** warning on **line 46**

Please avoid using the `!!` operator. For more information, see <https://developer.android.com/studio/write/lint.html#unsafe-dereference>.



File New Task



infer commented on **line 18**

There may be a **Execution Time Complexity Increase**: Time complexity of constructor **increased** from `O(1)` to `O(points.length)`. (error trace: TV115306648)

Feedback: complexity increase is [good catch | expected | wrong].

Sunday 10:20am • Like • Reply • Restore • Formatted

313

```
final LayoutResultHolder container =
```

314

```
Layout.createAndMeasureComponent(c, this, widthSpec, heightSpec);
```



infer commented on **line 314**

There may be a **Thread Safety Violation**: Unprotected write. Non-private method `Component.measure(...)` indirectly writes to field `this.mLayoutCreatedInWillRender` outside of synchronization.

Reporting because a superclass `class com.facebook.litho.EventTriggerTarget` is annotated `@ThreadSafe`, so we assume that this method can run in parallel with other non-private methods in the class (including itself).



# TOOLS: CODE HIGHLIGHTER



# TOOLS: CODE HIGHLIGHTER

## *Problem*

```
package com.facebook.litho

@HooksDsl
fun ComponentScope.useEffect(vararg deps: Any, onAttach: () -> Unit) {
    val entries = useEffectEntries ?: ArrayList()
    useEffectEntries = entries
    val uniqueId = "${context.globalKey}:${entries.size}"
    val persistence =
        if (deps.isEmpty()) {
            EffectPersistence.ALWAYS_UPDATE
        } else {
            EffectPersistence.USE_DEPS
        }
    entries.add(UseEffectAttachable(uniqueId, persistence, deps, onAttach))
}
```

# TOOLS: CODE HIGHLIGHTER

## Problem

```
package com.facebook.litho

@HooksDsl
fun ComponentScope.useEffect(vararg deps: Any, onAttach: () -> Unit) {
    val entries = useEffectEntries ?: ArrayList()
    useEffectEntries = entries
    val uniqueId = "${context.globalKey}:${entries.size}"
    val persistence =
        if (deps.isEmpty()) {
            EffectPersistence.ALWAYS_UPDATE
        } else {
            EffectPersistence.USE_DEPS
        }
    entries.add(UseEffectAttachable(uniqueId, persistence, deps, onAttach))
}
```

## Solution

```
package com.facebook.litho

@HooksDsl
fun ComponentScope.useEffect(vararg deps: Any, onAttach: () -> Unit) {
    val entries = useEffectEntries ?: ArrayList()
    useEffectEntries = entries
    val uniqueId = "${context.globalKey}:${entries.size}"
    val persistence =
        if (deps.isEmpty()) {
            EffectPersistence.ALWAYS_UPDATE
        } else {
            EffectPersistence.USE_DEPS
        }
    entries.add(UseEffectAttachable(uniqueId, persistence, deps, onAttach))
}
```

# TOOLS: CODE HIGHLIGHTER

Since  
Pygments 2.8.0

## Problem

```
package com.facebook.litho

@HooksDsl
fun ComponentScope.useEffect(vararg deps: Any, onAttach: () -> Unit) {
    val entries = useEffectEntries ?: ArrayList()
    useEffectEntries = entries
    val uniqueId = "${context.globalKey}:${entries.size}"
    val persistence =
        if (deps.isEmpty()) {
            EffectPersistence.ALWAYS_UPDATE
        } else {
            EffectPersistence.USE_DEPS
        }
    entries.add(UseEffectAttachable(uniqueId, persistence, deps, onAttach))
}
```

## Solution

```
package com.facebook.litho

@HooksDsl
fun ComponentScope.useEffect(vararg deps: Any, onAttach: () -> Unit) {
    val entries = useEffectEntries ?: ArrayList()
    useEffectEntries = entries
    val uniqueId = "${context.globalKey}:${entries.size}"
    val persistence =
        if (deps.isEmpty()) {
            EffectPersistence.ALWAYS_UPDATE
        } else {
            EffectPersistence.USE_DEPS
        }
    entries.add(UseEffectAttachable(uniqueId, persistence, deps, onAttach))
}
```

# TOOLS: FORMATTER

# TOOLS: FORMATTER

- ◆ *Problem:* KtLint fails to consistently produce nice-looking code that fits 100 chars width

## TOOLS: FORMATTER

- ◆ *Problem:* KtLint fails to consistently produce nice-looking code that fits 100 chars width
- ◆ *Solution:* Ktfmt – better Kotlin code formatter

# TOOLS: FORMATTER

- ◆ *Problem:* KtLint fails to consistently produce nice-looking code that fits 100 chars width
- ◆ *Solution:* Ktfmt – better Kotlin code formatter
  - Based on Google Java Formatter











# LIBRARIES



# LIBRARIES

◆ Better codegen: KAPT - 🙄, compiler plugins - 🤘

# LIBRARIES

- ◆ Better codegen: KAPT - 🙄, compiler plugins - 🙌
- ◆ No codegen: code generation - 🙄, language features - 🙌

# LIBRARIES

- ◆ Better codegen: KAPT - 🙄, compiler plugins - 🙌
- ◆ No codegen: code generation - 🙄, language features - 🙌
- ◆ Better APIs: Java Kotlin - 🙄, idiomatic Kotlin - 🙌



# NO CODEGEN: LITHO KOTLIN

---

```
@LayoutSpec
public class PlaygroundComponentSpec {
    @OnCreateInitialState
    static void onCreateInitialState(
        @Prop int startCount, StateValue<Integer> counter) {
        counter.set(startCount);
    }

    @OnCreateLayout
    static Component onCreateLayout(ComponentContext c, @State int counter) {
        return Column.create(c)
            .paddingDip(YogaEdge.ALL, 16)
            .clickHandler(PlaygroundComponent.onClickEvent(c))
            .child(Text.create(c).text("Hello, World!").textSizeSp(20))
            .child(
                Text.create(c)
                    .text("with " + repeat("❤️", counter) + " from London")
                    .textStyle(Typeface.ITALIC))
            .build();
    }

    @OnUpdateState
    static void onUpdateState(StateValue<Integer> counter) {
        counter.set(counter.get() + 1);
    }

    @OnEvent(ClickEvent.class)
    static void onClickEvent(ComponentContext c) {
        PlaygroundComponent.onUpdateState(c);
    }
}
```

# NO CODEGEN: LITHO KOTLIN

```
@LayoutSpec
public class PlaygroundComponentSpec {
    @OnCreateInitialState
    static void onCreateInitialState(
        @Prop int startCount, StateValue<Integer> counter) {
        counter.set(startCount);
    }

    @OnCreateLayout
    static Component onCreateLayout(ComponentContext c, @State int counter) {
        return Column.create(c)
            .paddingDip(YogaEdge.ALL, 16)
            .clickHandler(PlaygroundComponent.onClickEvent(c))
            .child(Text.create(c).text("Hello, World!").textSizeSp(20))
            .child(
                Text.create(c)
                    .text("with " + repeat("❤️", counter) + " from London")
                    .textStyle(Typeface.ITALIC))
            .build();
    }

    @OnUpdateState
    static void onUpdateState(StateValue<Integer> counter) {
        counter.set(counter.get() + 1);
    }

    @OnEvent(ClickEvent.class)
    static void onClickEvent(ComponentContext c) {
        PlaygroundComponent.onUpdateState(c);
    }
}
```

```
class PlaygroundComponent(val startCount: Int) : KComponent() {

    override fun ComponentScope.render(): Component {
        val counter = useState { startCount }

        return Column(style = Style
            .padding(16.dp)
            .onClick { counter.update { value -> value + 1 } }) {

            child(Text(text = "Hello, World!", textSize = 20.sp))
            child(
                Text(
                    text = "with ${"❤️".repeat(counter.value)} from London",
                    textStyle = Typeface.ITALIC)
            )
        }
    }
}
```

# NO CODEGEN: LITHO KOTLIN

```
@LayoutSpec
public class PlaygroundComponentSpec {
    @OnCreateInitialState
    static void onCreateInitialState(
        @Prop int startCount, StateValue<Integer> counter) {
        counter.set(startCount);
    }

    @OnCreateLayout
    static Component onCreateLayout(ComponentContext c, @State int counter) {
        return Column.create(c)
            .paddingDip(YogaEdge.ALL, 16)
            .clickHandler(PlaygroundComponent.onClickEvent(c))
            .child(Text.create(c).text("Hello, World!").textSizeSp(20))
            .child(
                Text.create(c)
                    .text("with " + repeat("❤️", counter) + " from London")
                    .textStyle(Typeface.ITALIC))
            .build();
    }

    @OnUpdateState
    static void onUpdateState(StateValue<Integer> counter) {
        counter.set(counter.get() + 1);
    }

    @OnEvent(ClickEvent.class)
    static void onClickEvent(ComponentContext c) {
        PlaygroundComponent.onUpdateState(c);
    }
}
```

```
class PlaygroundComponent(val startCount: Int) : KComponent() {

    override fun ComponentScope.render(): Component {
        val counter = useState { startCount }

        return Column(style = Style
            .padding(16.dp)
            .onClick { counter.update { value -> value + 1 } }) {

            child(Text(text = "Hello, World!", textSize = 20.sp))
            child(
                Text(
                    text = "with ${"❤️".repeat(counter.value)} from London",
                    textStyle = Typeface.ITALIC)
            )
        }
    }
}
```

# NO CODEGEN: LITHO KOTLIN

```
@LayoutSpec
public class PlaygroundComponentSpec {
    @OnCreateInitialState
    static void onCreateInitialState(
        @Prop int startCount, StateValue<Integer> counter) {
        counter.set(startCount);
    }

    @OnCreateLayout
    static Component onCreateLayout(ComponentContext c, @State int counter) {
        return Column.create(c)
            .paddingDip(YogaEdge.ALL, 16)
            .clickHandler(PlaygroundComponent.onClickEvent(c))
            .child(Text.create(c).text("Hello, World!").textSizeSp(20))
            .child(
                Text.create(c)
                    .text("with " + repeat("❤️", counter) + " from London")
                    .textStyle(Typeface.ITALIC))
            .build();
    }

    @OnUpdateState
    static void onUpdateState(StateValue<Integer> counter) {
        counter.set(counter.get() + 1);
    }

    @OnEvent(ClickEvent.class)
    static void onClickEvent(ComponentContext c) {
        PlaygroundComponent.onUpdateState(c);
    }
}
```

```
class PlaygroundComponent(val startCount: Int) : KComponent() {

    override fun ComponentScope.render(): Component {
        val counter = useState { startCount }

        return Column(style = Style
            .padding(16.dp)
            .onClick { counter.update { value -> value + 1 } }) {

            child(Text(text = "Hello, World!", textSize = 20.sp))
            child(
                Text(
                    text = "with ${"❤️".repeat(counter.value)} from London",
                    textStyle = Typeface.ITALIC)
            )
        }
    }
}
```

# BUILD CODE



# BUILD SYSTEMS

---

# **BUILD SYSTEMS**

---

- ◆ Facebook uses BUCK, not Gradle

# BUILD SYSTEMS

- ◆ Facebook uses BUCK, not Gradle
  - Multi-language support



# BUILD SYSTEMS

- ◆ Facebook uses BUCK, not Gradle
  - Multi-language support
  - More explicit and side-effect free configuration

# BUILD SYSTEMS

- ◆ Facebook uses BUCK, not Gradle
  - Multi-language support
  - More explicit and side-effect free configuration
  - Better reproducibility

# BUILD SYSTEMS

- ◆ Facebook uses BUCK, not Gradle
  - Multi-language support
  - More explicit and side-effect free configuration
  - Better reproducibility
  - Better parallelism and scalability

# **BUILD SYSTEMS: BUCK VS GRADLE**

---

# BUILD SYSTEMS: BUCK VS GRADLE

- ◆ Different modules structure

# BUILD SYSTEMS: BUCK VS GRADLE

- ◆ Different modules structure
- ◆ Different configuration language

# BUILD SYSTEMS: BUCK VS GRADLE

- ◆ Different modules structure
- ◆ Different configuration language
- ◆ It's just different

# BUILD SYSTEMS: BUCK VS GRADLE

- ◆ Different modules structure
- ◆ Different configuration language
- ◆ It's just different
- ◆ But... It's open-source!



# BUILD SYSTEMS: BUCK VS GRADLE

- ◆ Different modules structure
- ◆ Different configuration language
- ◆ It's just different
- ◆ But... It's open-source!
- ◆ **Initial Kotlin support by OSS – Uber**

# BUCK + KOTLIN

# BUCK + KOTLIN

- ◆ General slowness of Kotlin Compiler and notoriously slow KAPT

## BUCK + KOTLIN

- ◆ General slowness of Kotlin Compiler and notoriously slow KAPT
- ◆ For our codebase: **2-2.5x slower** to compile Kotlin than Java

# BUCK + KOTLIN

- ◆ General slowness of Kotlin Compiler and notoriously slow KAPT
- ◆ For our codebase: **2-2.5x slower** to compile Kotlin than Java
- ◆ Is this The End?

**WE NEED A HERO**



# WE NEED A HERO

Buck can compile against ABIs Jars,  
instead of Full Jars

# WHAT IS ABI?

---



# WHAT IS ABI?

---

Application Binary Interface – public interface of your module; resources & class interfaces

# WHAT IS ABI?

---

```
package com.facebook.rendercore;

public class RenderUnit<MOUNT_CONTENT> {

    public enum RenderType {
        DRAWABLE,
        VIEW,
    }

    private final RenderUnit.RenderType renderType;
    private final Extension mountUnmountExtension;

    public RenderUnit(
        RenderType renderType, Extension mountUnmountExtension) {
        this.renderType = renderType;
        this.mountUnmountExtension = mountUnmountExtension;
    }

    public RenderType getRenderType() {
        return renderType;
    }
}
```

# WHAT IS ABI?

---

```
package com.facebook.rendercore;

public class RenderUnit<MOUNT_CONTENT> {

    public enum RenderType {
        DRAWABLE,
        VIEW,
    }

    private final RenderUnit.RenderType renderType;
    private final Extension mountUnmountExtension;

    public RenderUnit(
        RenderType renderType, Extension mountUnmountExtension) {
        this.renderType = renderType;
        this.mountUnmountExtension = mountUnmountExtension;
    }

    public RenderType getRenderType() {
        return renderType;
    }
}
```

# WHAT IS ABI?

---

```
package com.facebook.rendercore;

public class RenderUnit<MOUNT_CONTENT> {

    public enum RenderType {
        DRAWABLE,
        VIEW,
    }

    public RenderUnit(
        RenderType renderType, Extension mountUnmountExtension);

    public RenderType getRenderType();
}
```

# ABI BENEFITS

---

# ABI BENEFITS

---

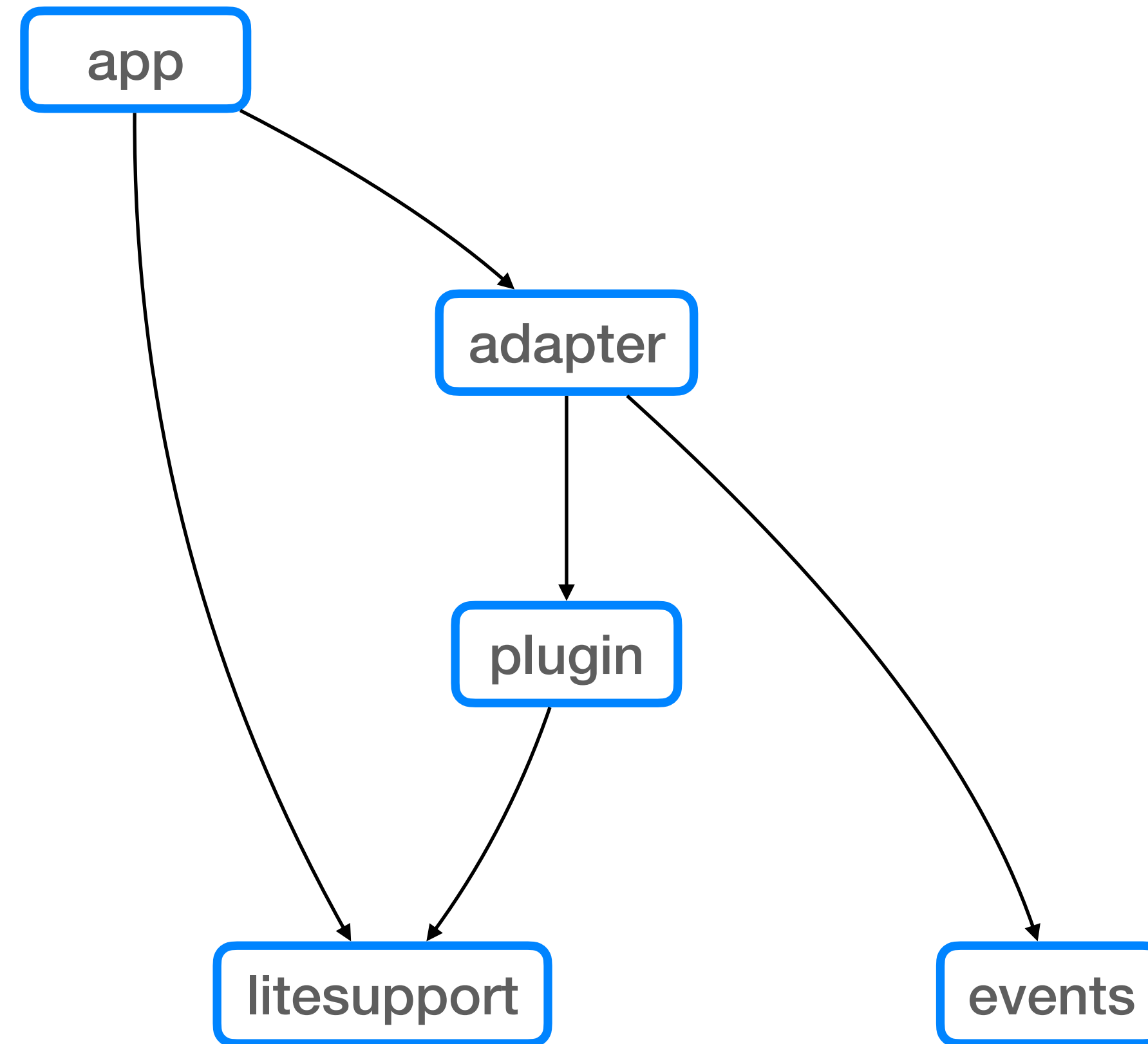
- ◆ ABI jars help determine which modules need to be rebuilt during incremental build

# ABI BENEFITS

---

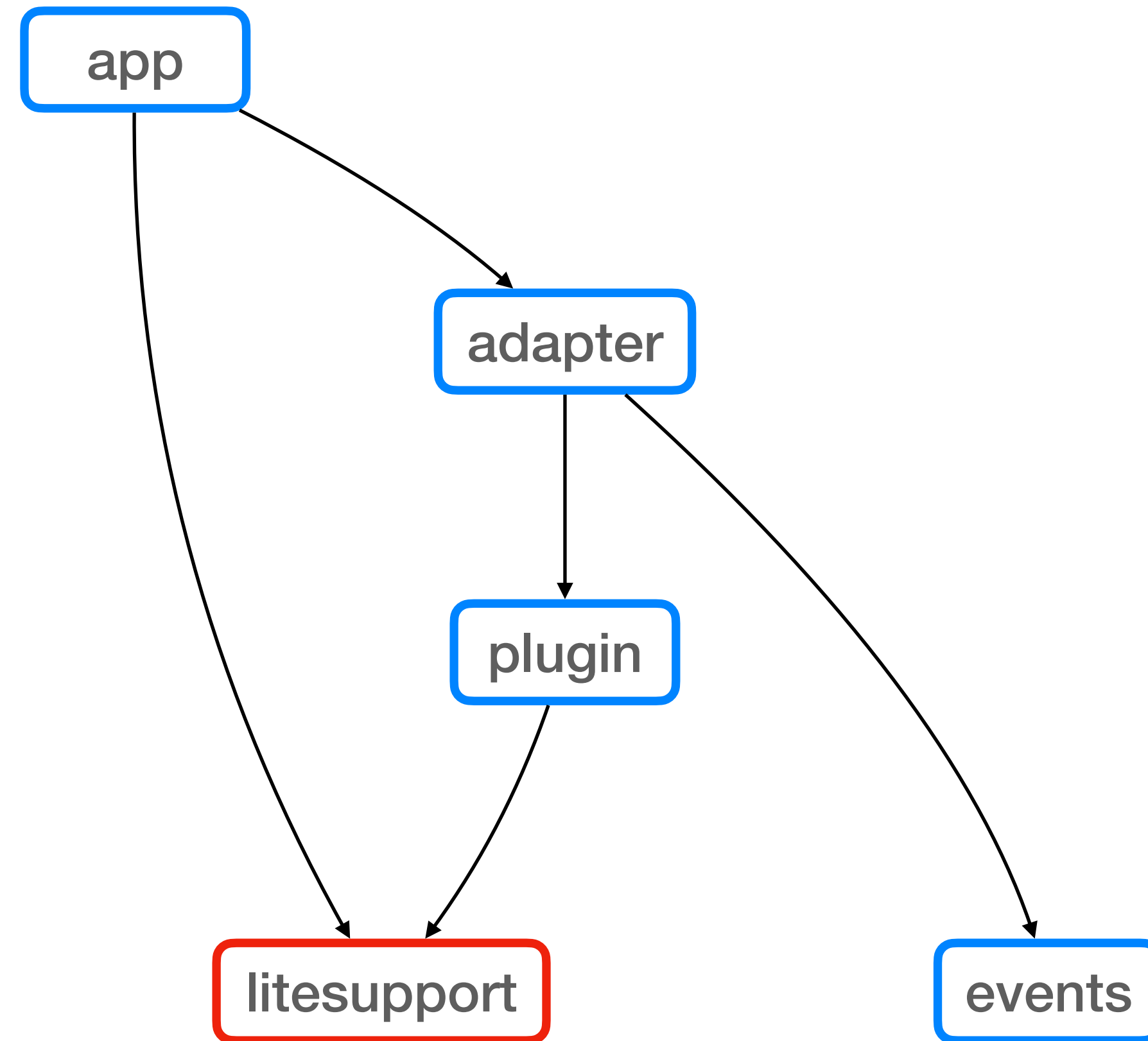
- ◆ ABI jars help determine which modules need to be rebuilt during incremental build
- ◆ Compiler can use ABI Jars in the compilation classpath instead of full jars to decrease resource usage

# ABI: MODULES

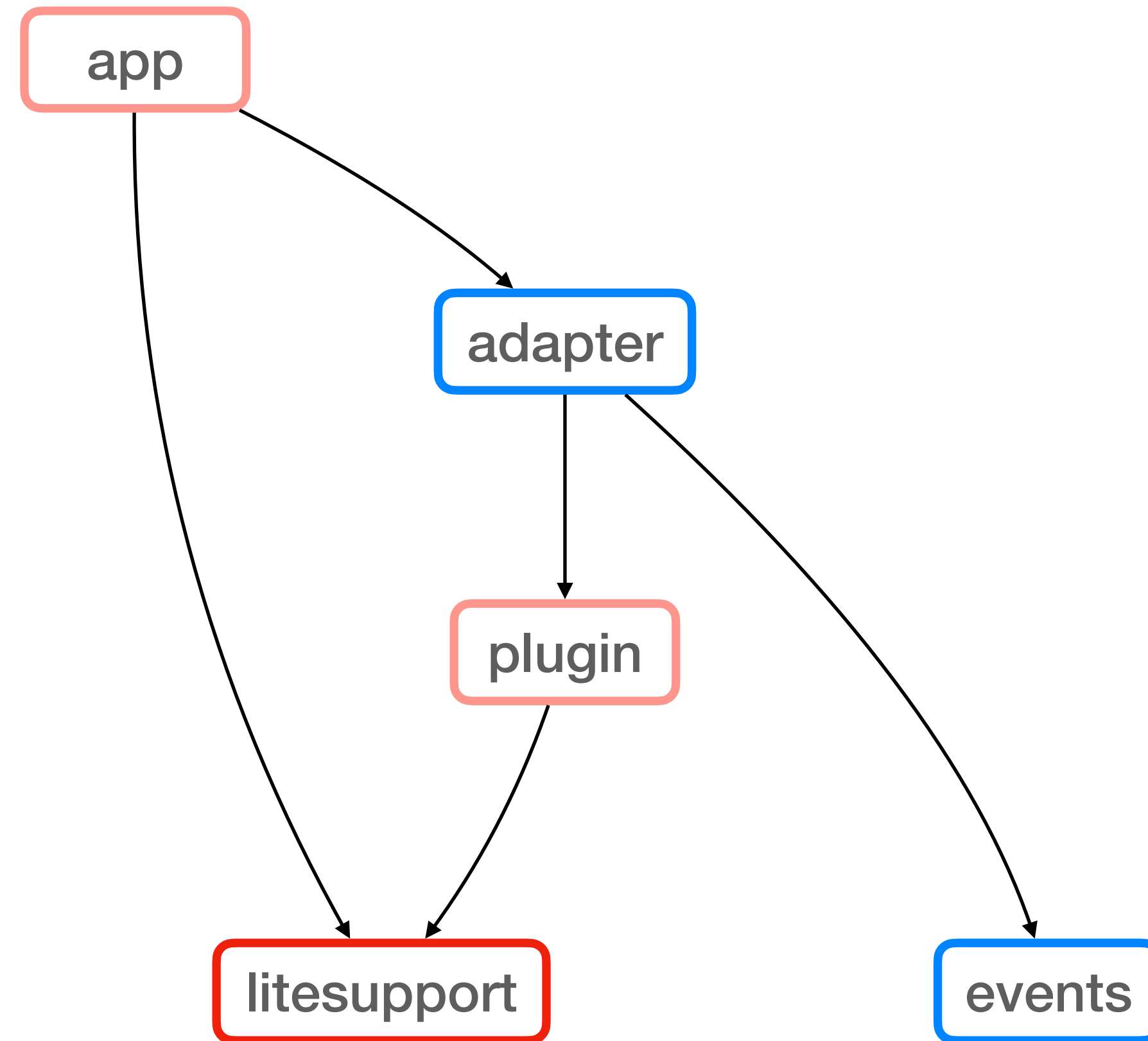




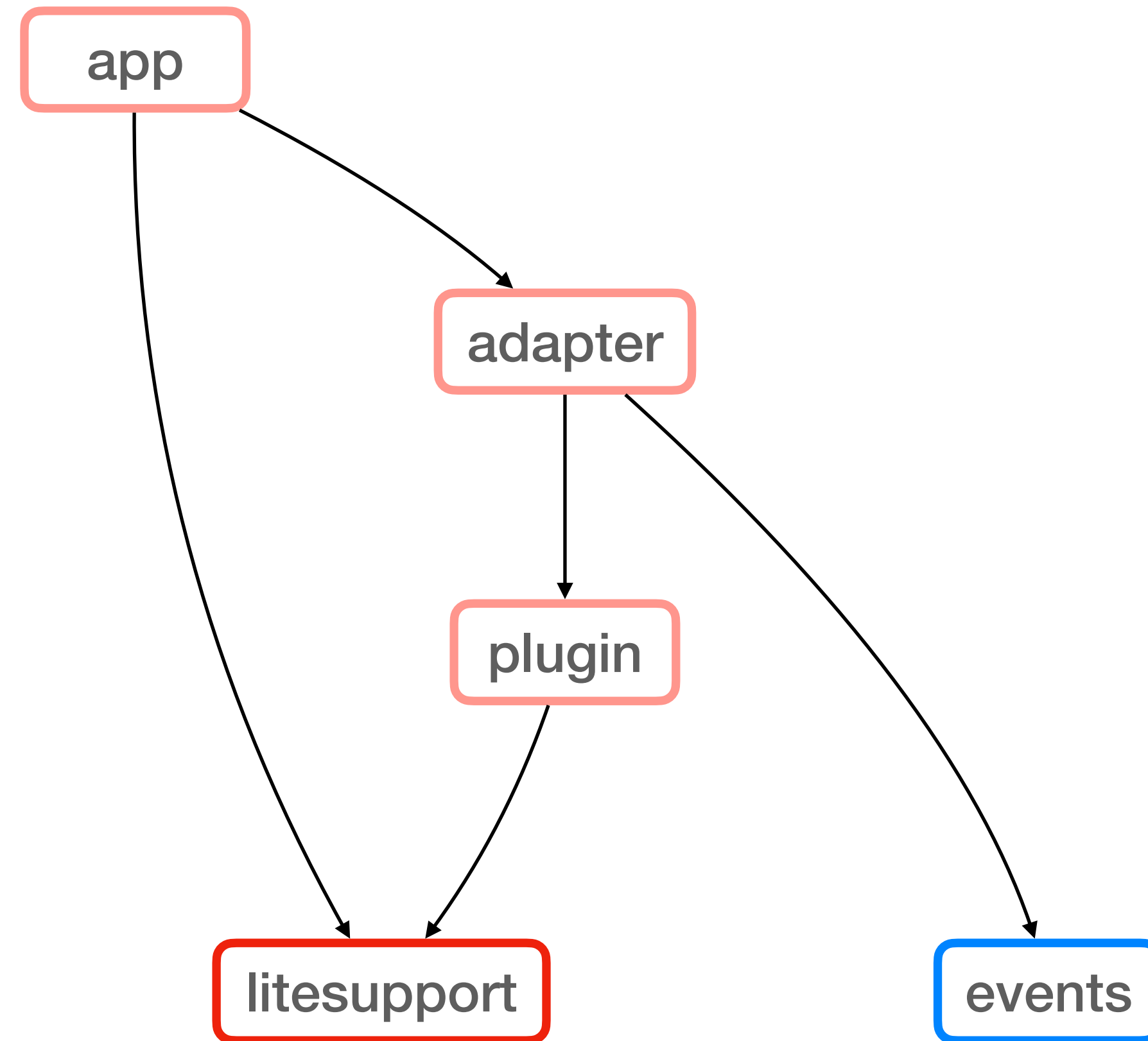
# ABI: MODULES



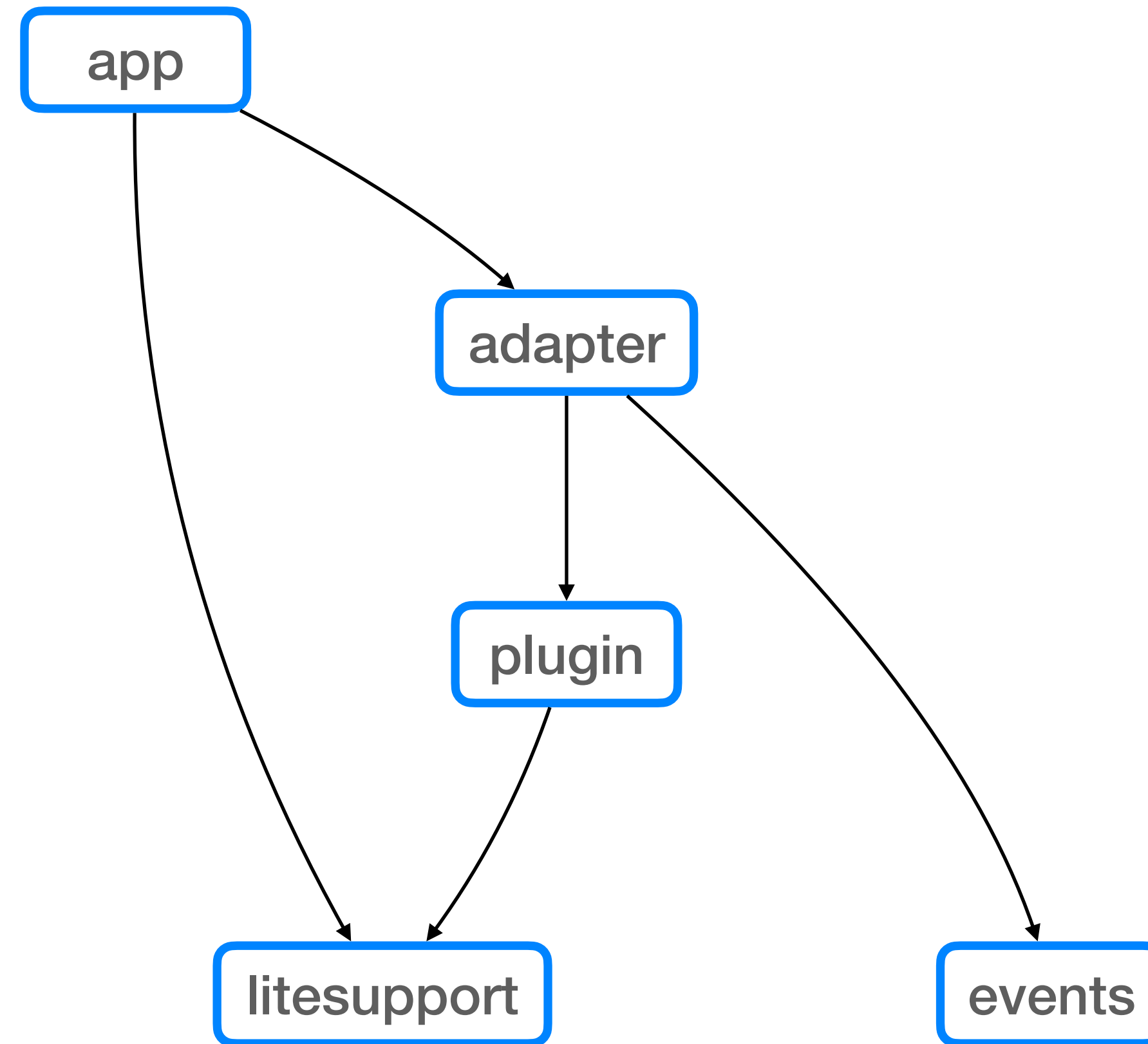
# ABI: MODULES



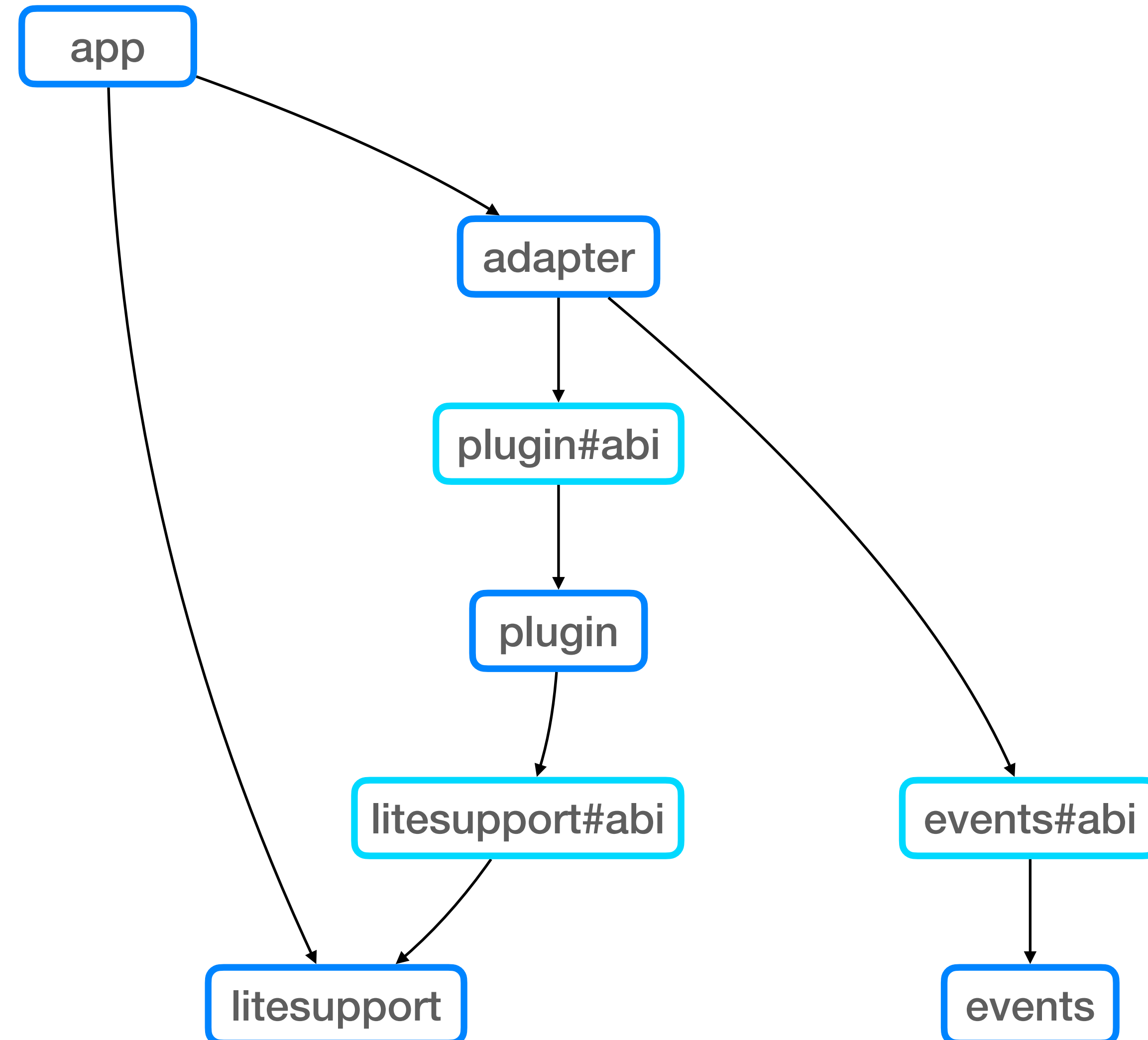
# ABI: MODULES



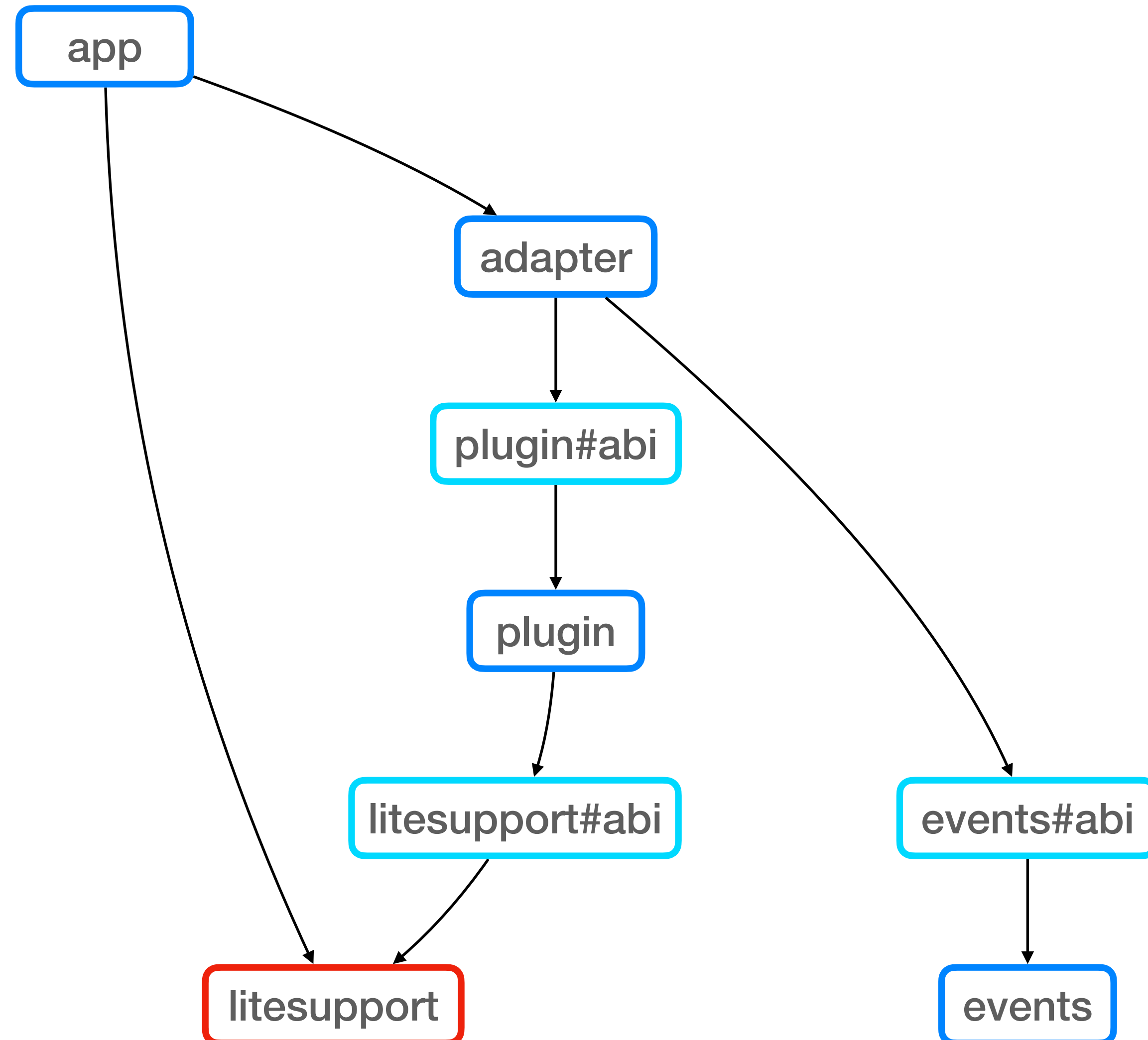
# ABI: MODULES



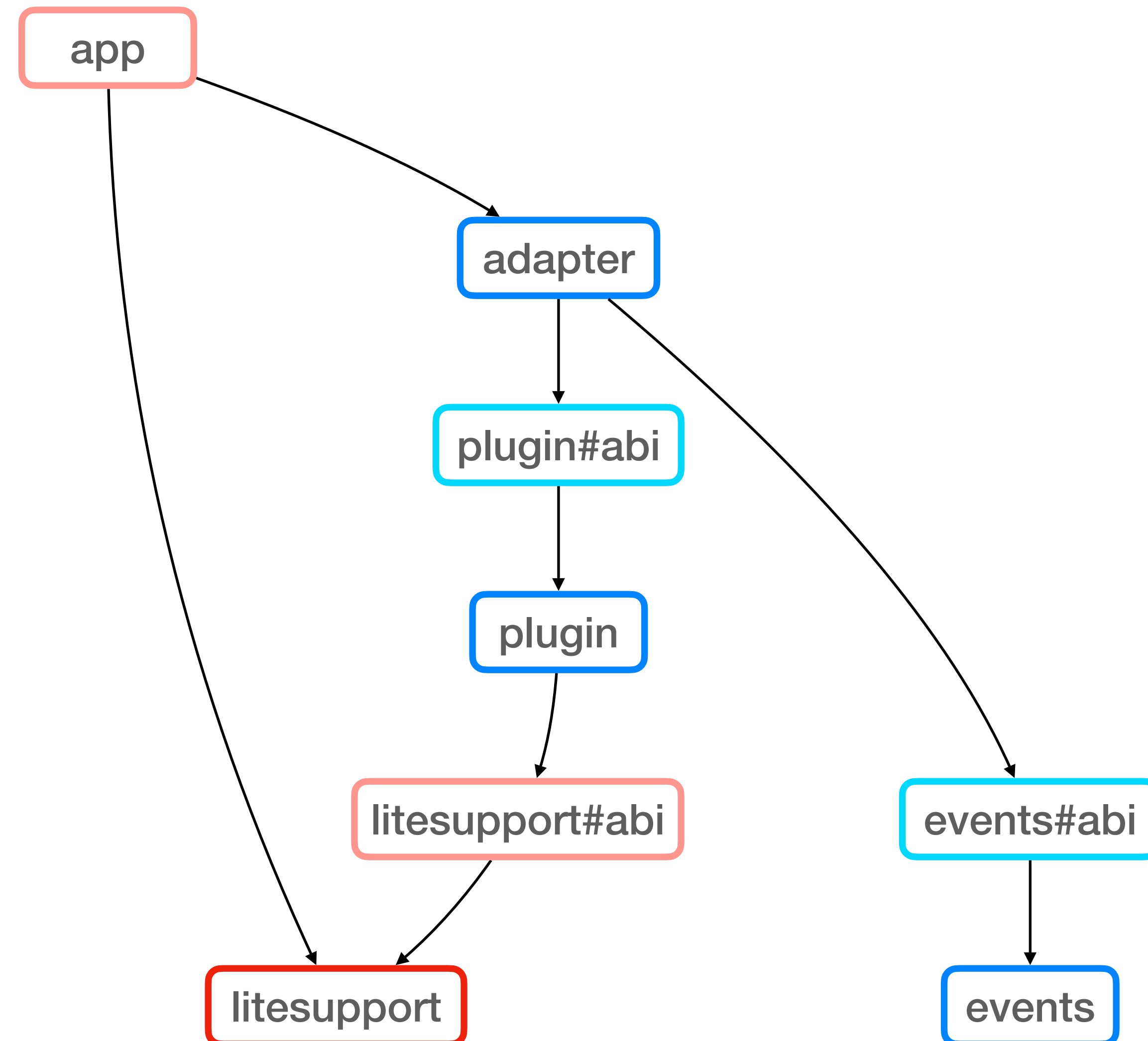
# ABI: CLASS-ABI



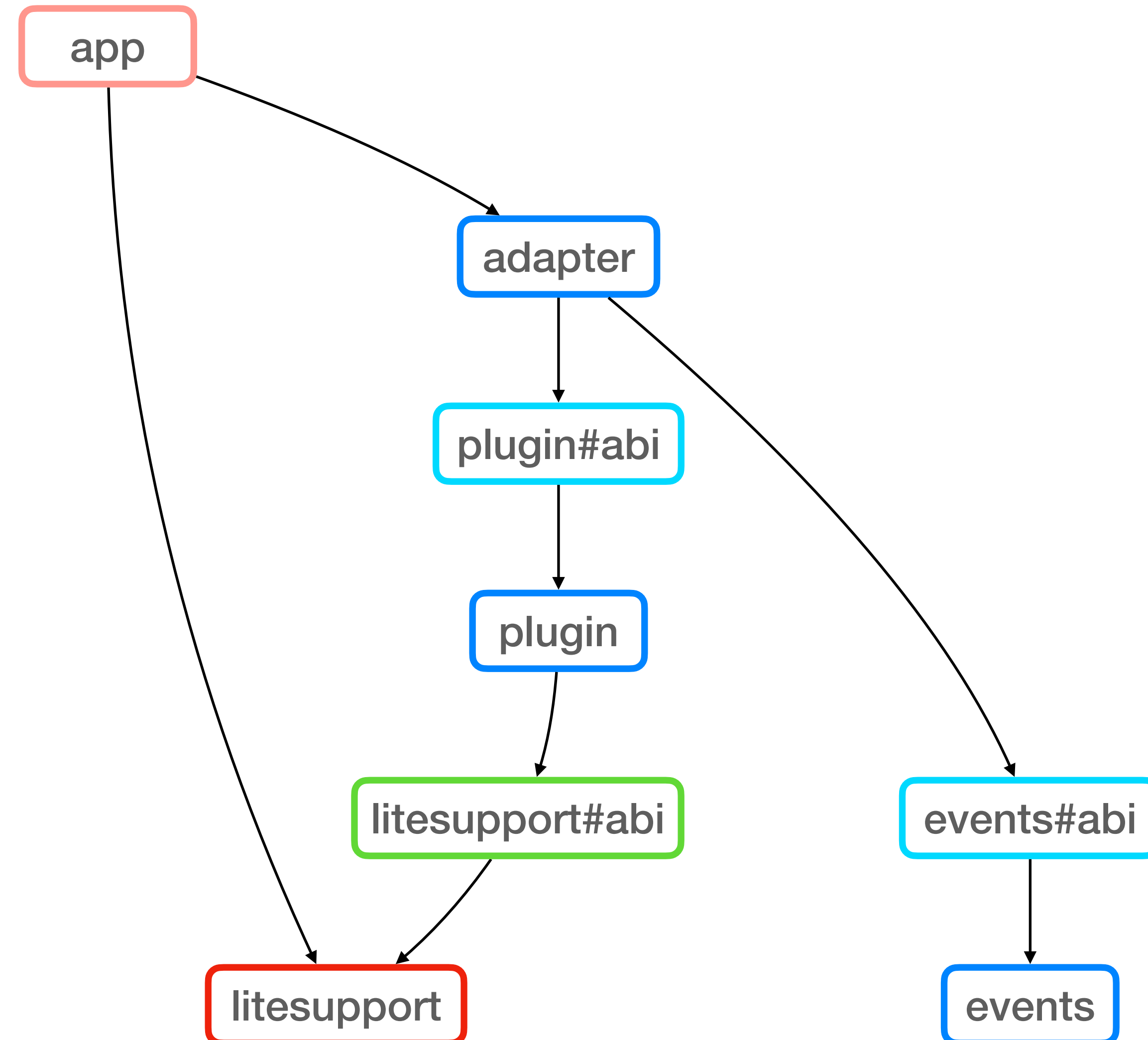
# ABI: CLASS-ABI



# ABI: CLASS-ABI

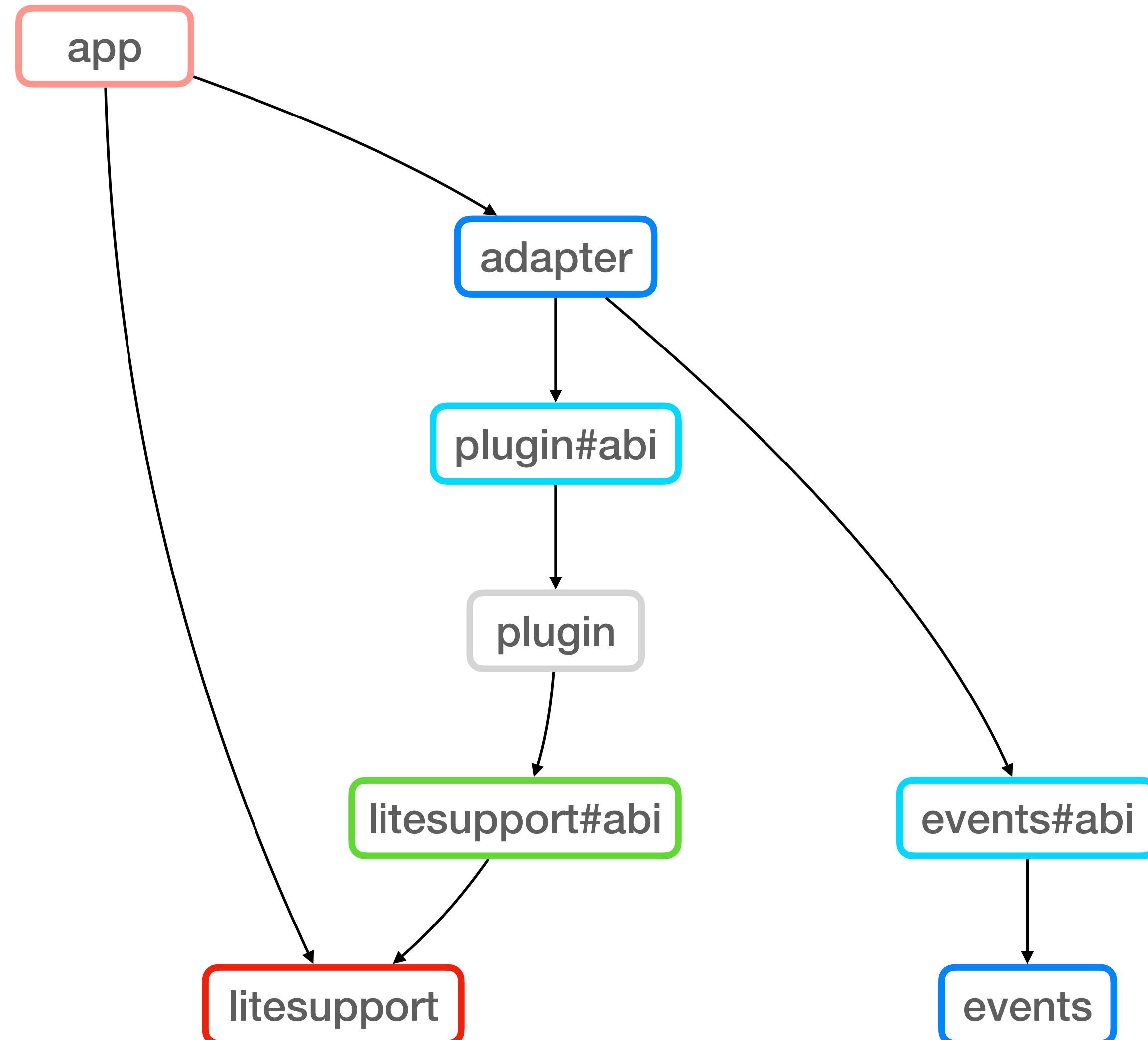


# ABI: CLASS-ABI

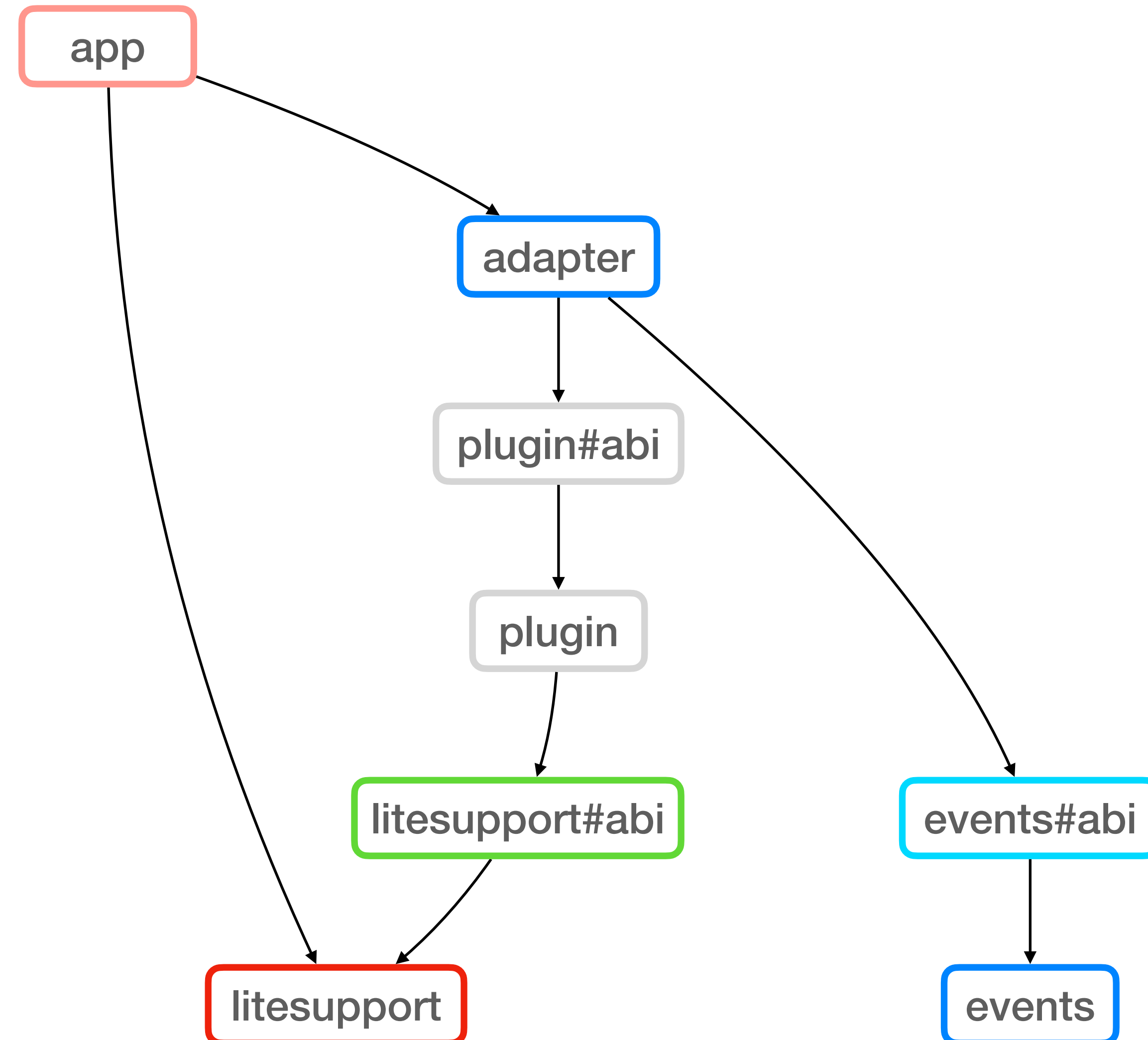




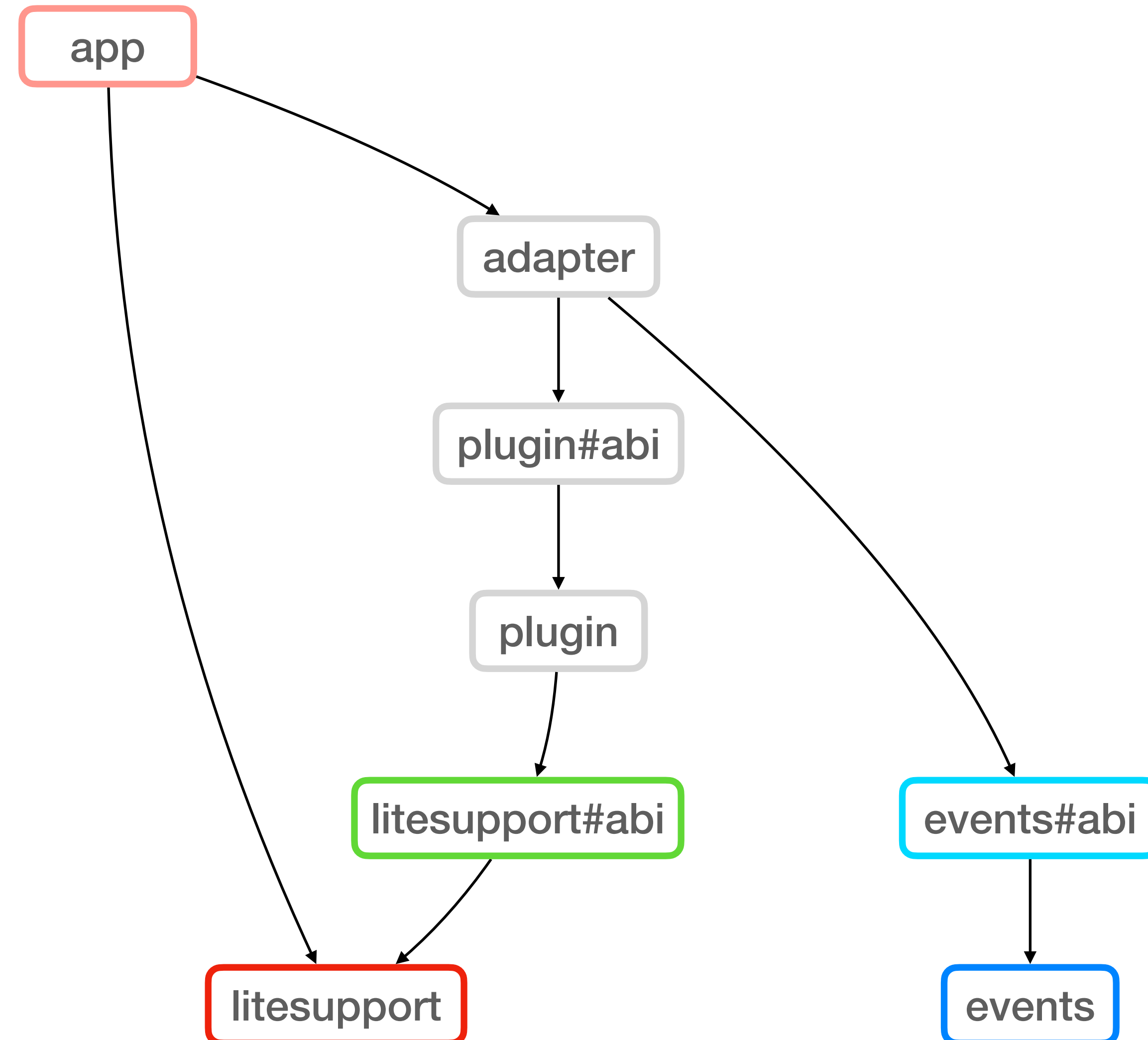
# ABI: CLASS-ABI



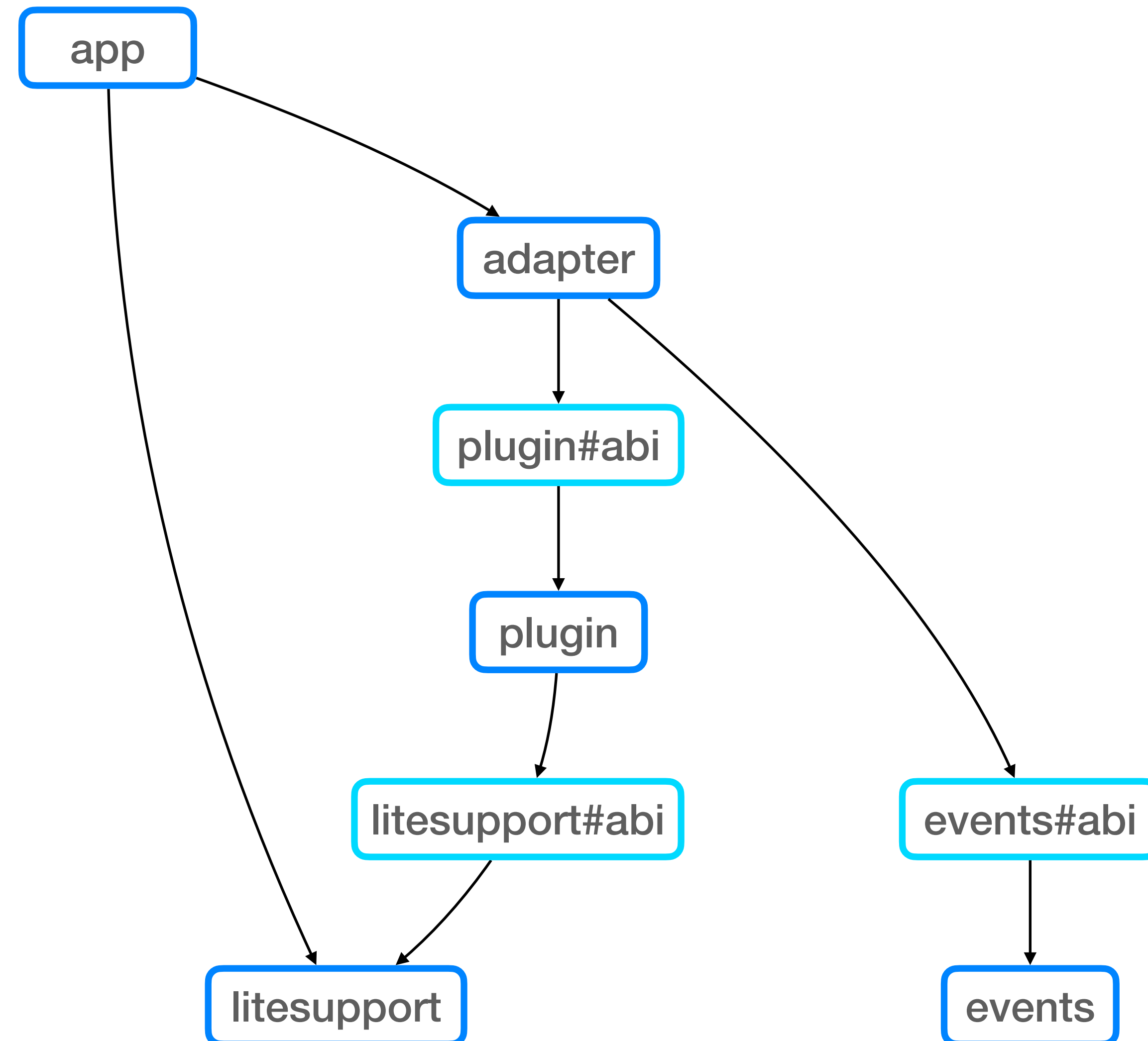
# ABI: CLASS-ABI



# ABI: CLASS-ABI



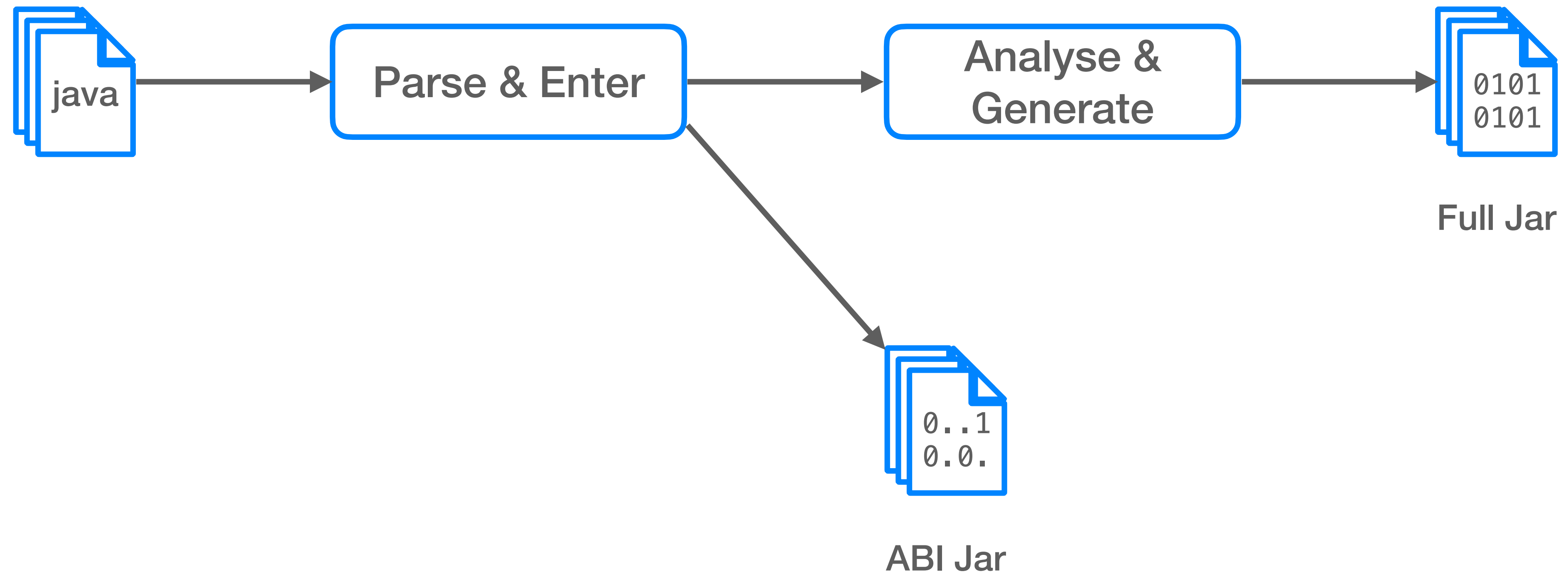
# ABI: CLASS-ABI



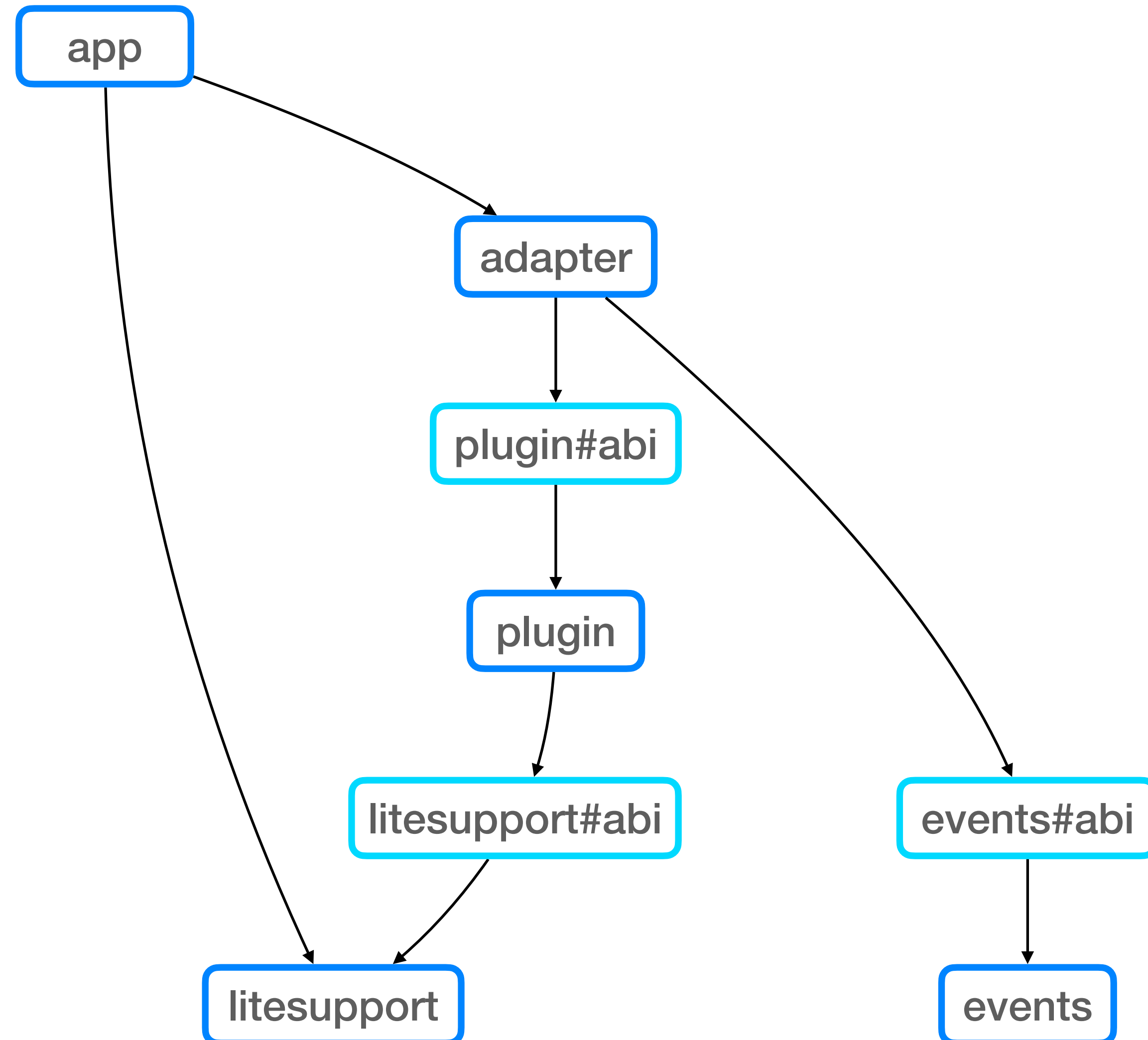
# JAVA COMPILATION



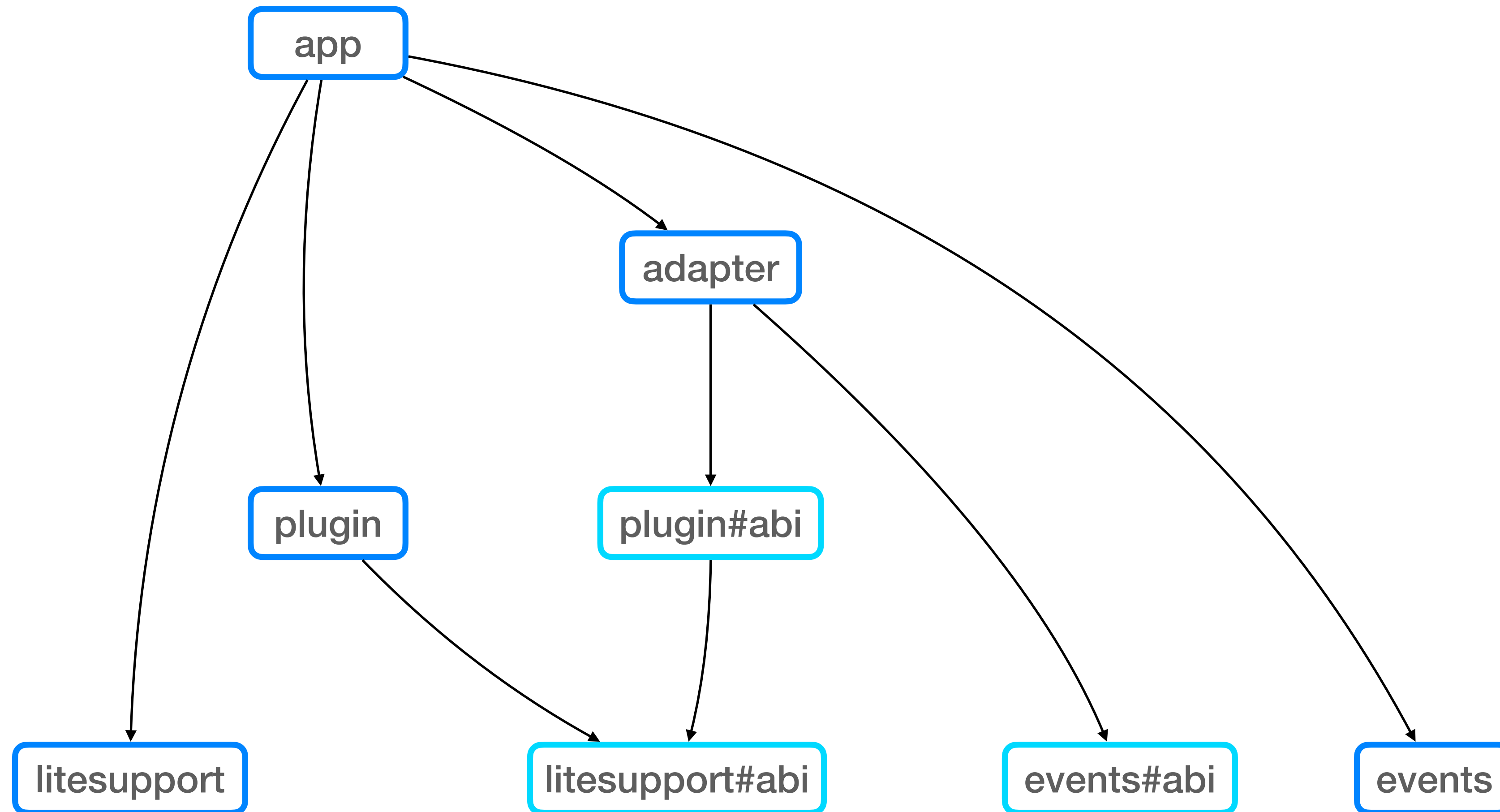
# JAVA COMPILATION



# ABI: CLASS-ABI

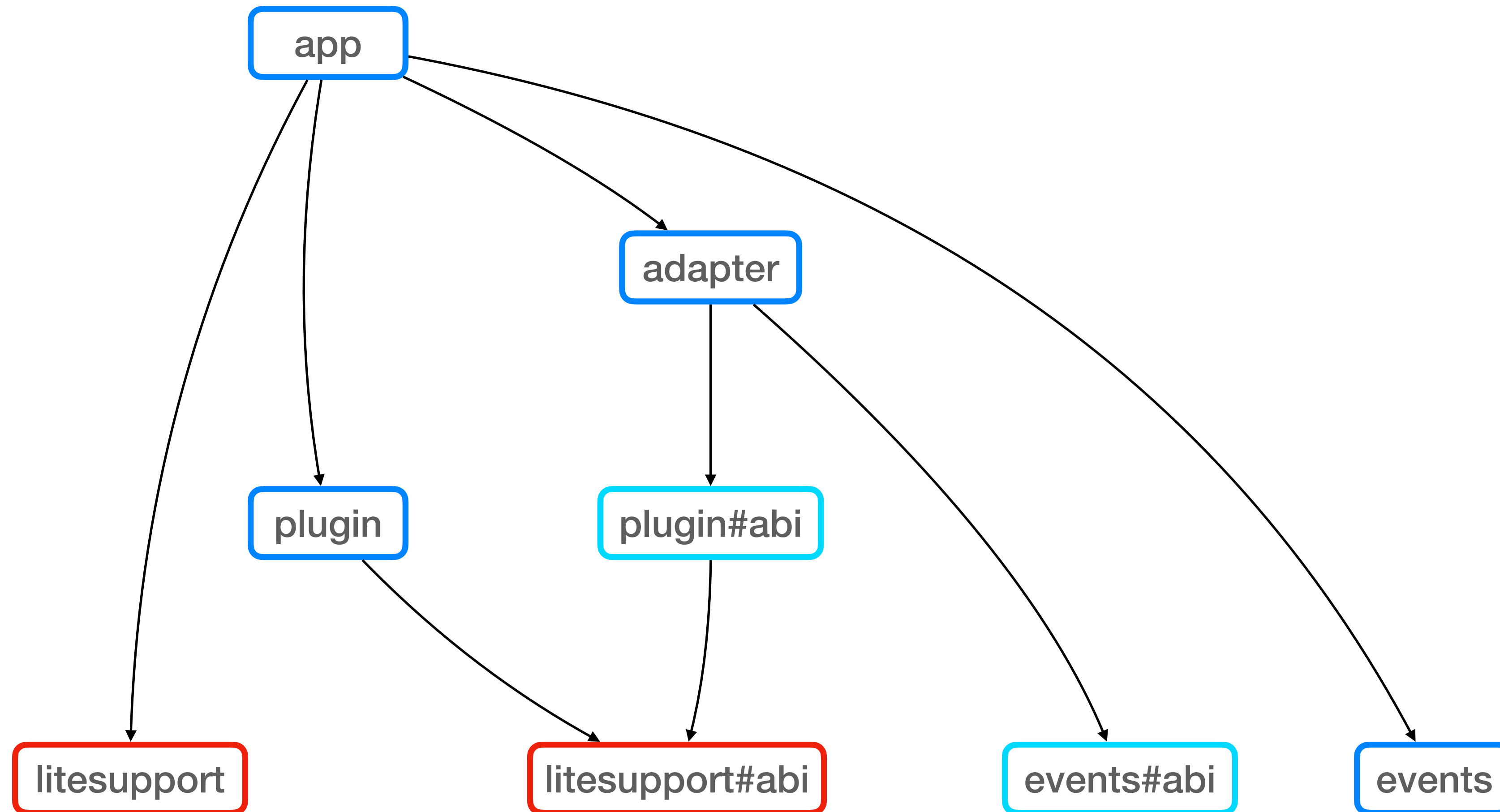


# ABI: SOURCE-ABI

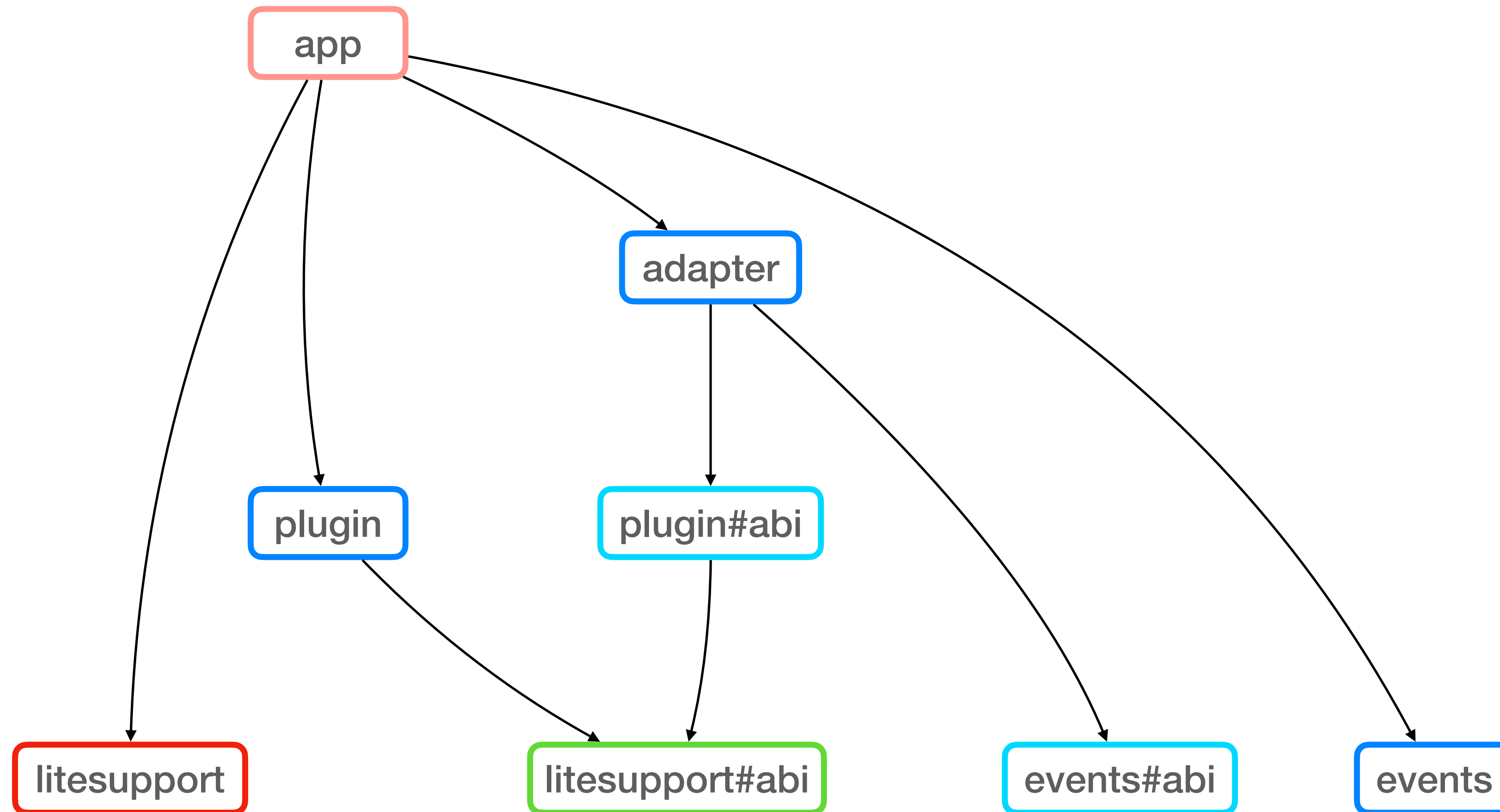




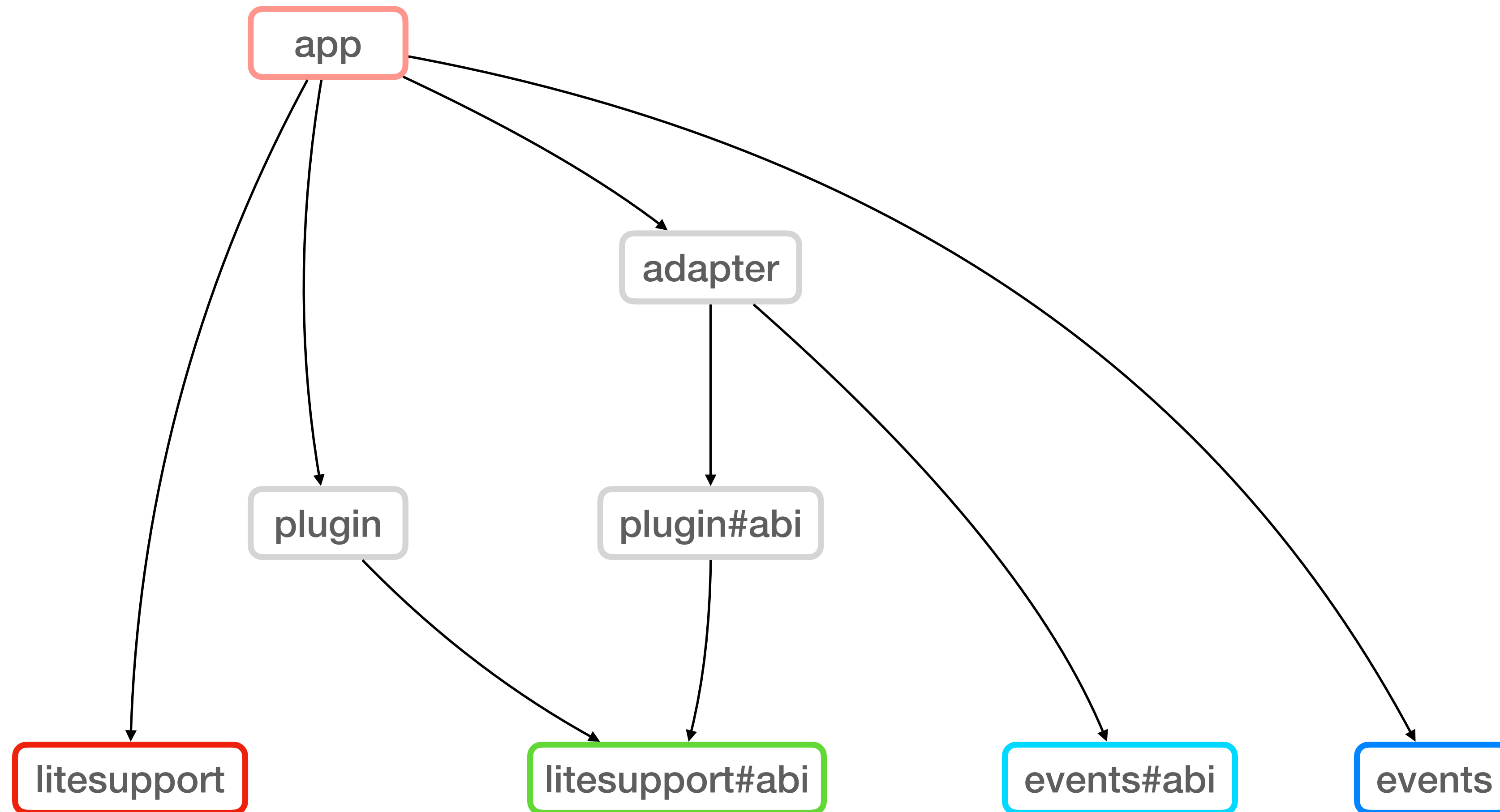
# ABI: SOURCE-ABI



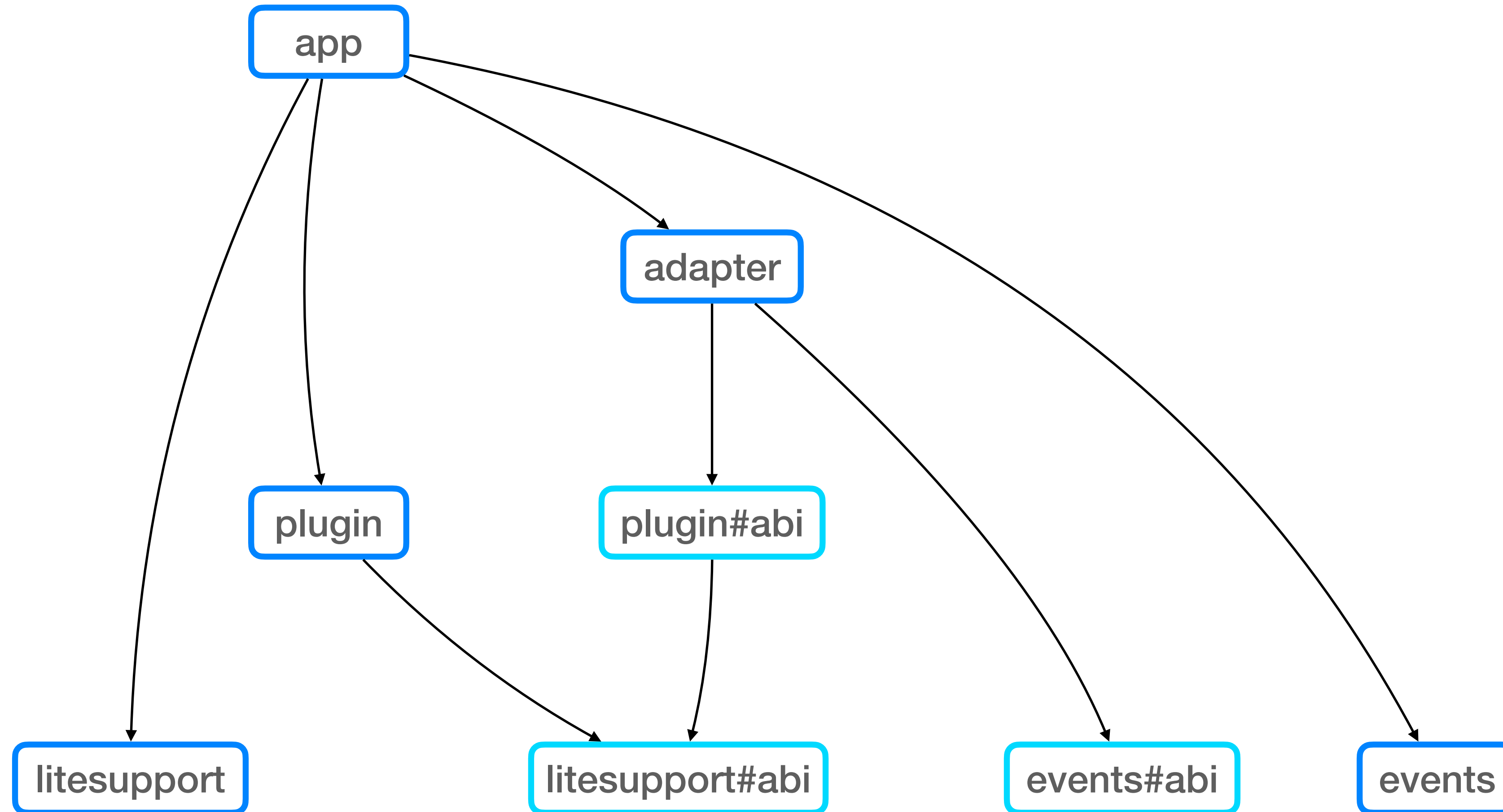
# ABI: SOURCE-ABI



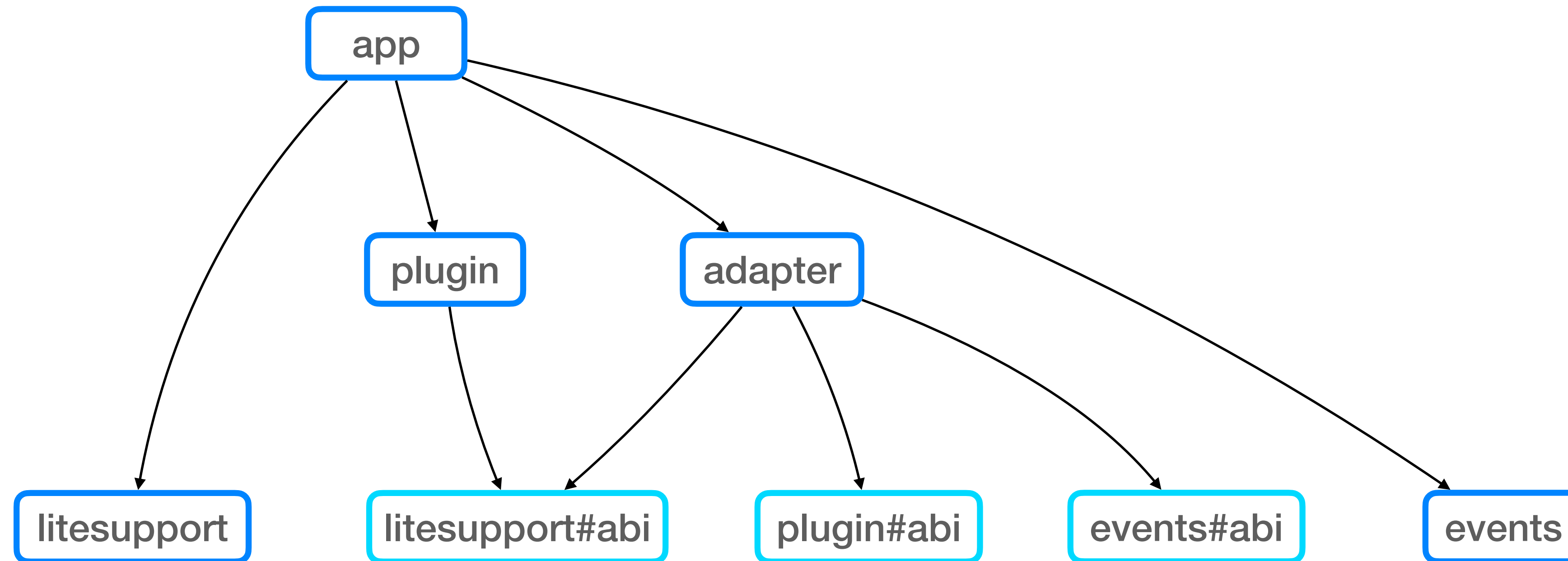
# ABI: SOURCE-ABI



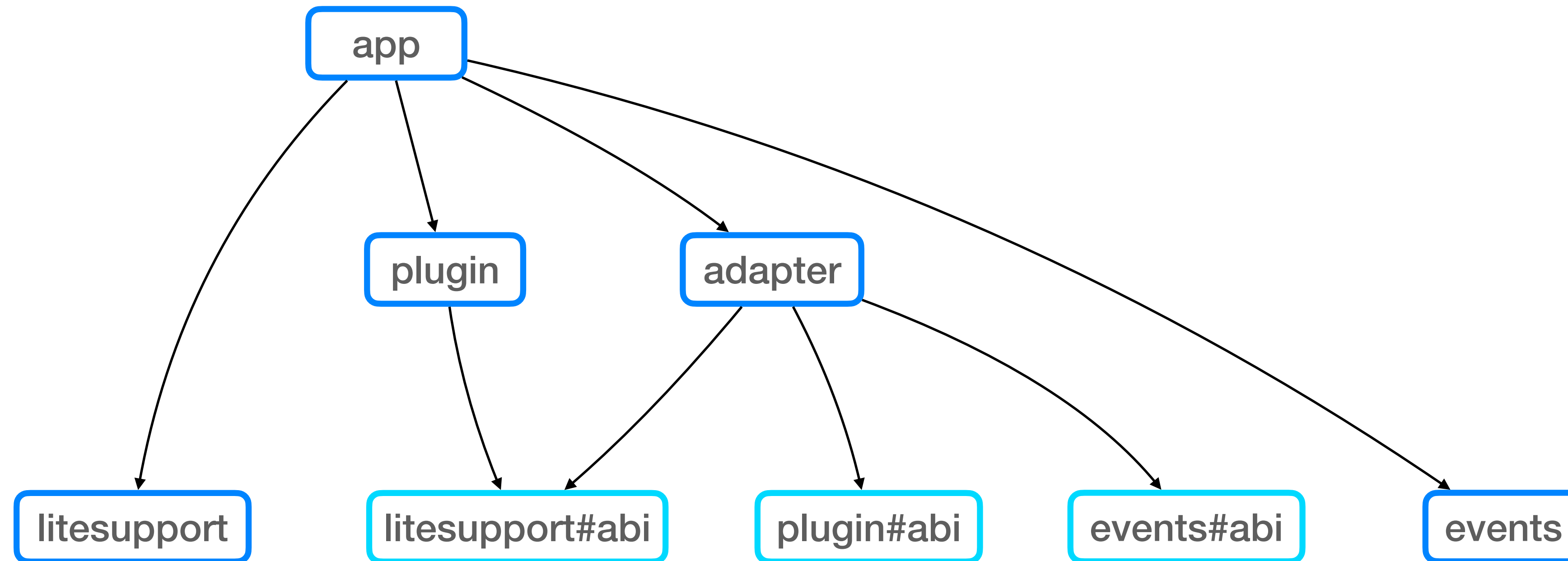
# ABI: SOURCE-ABI – CAN WE DO BETTER?



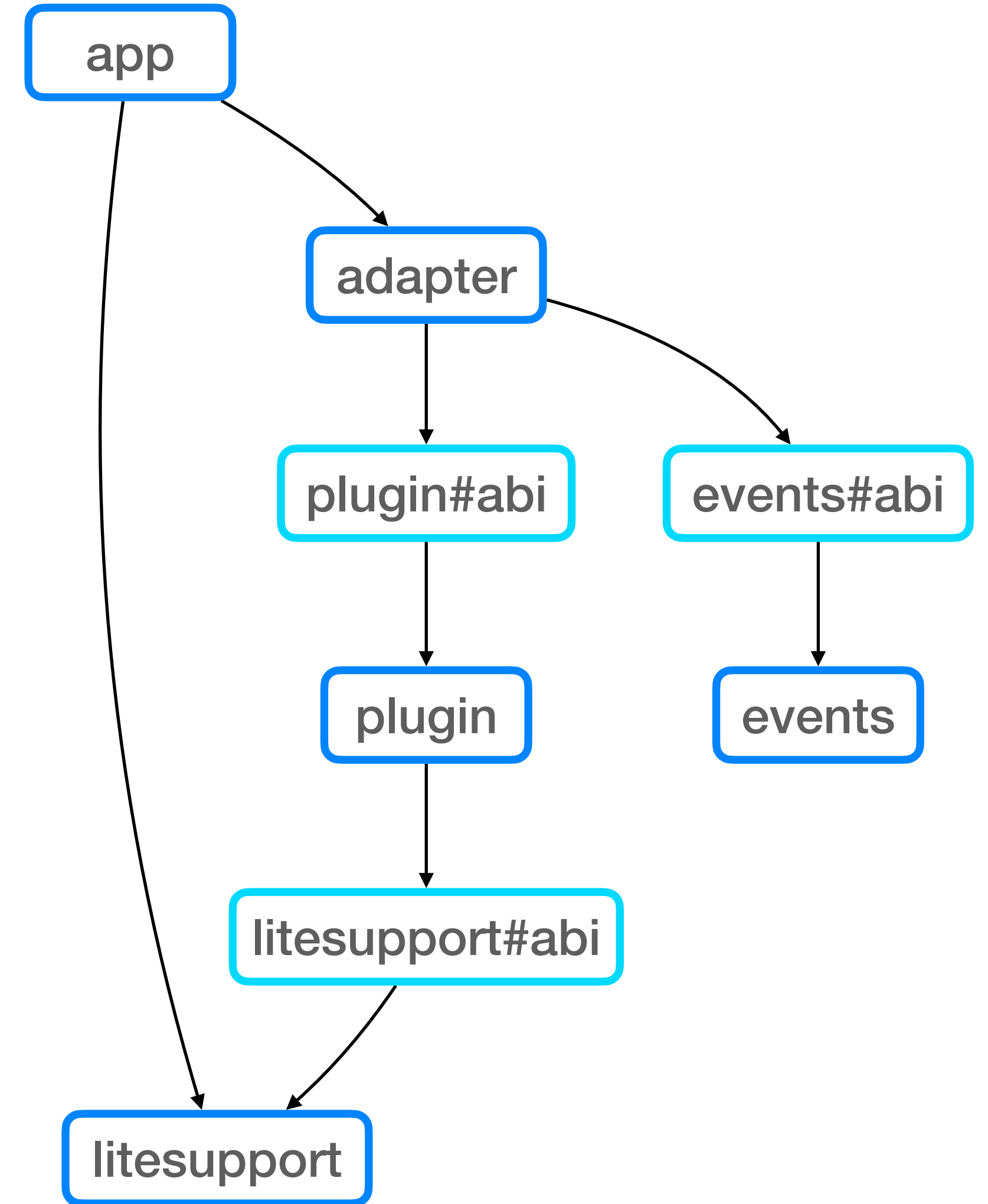
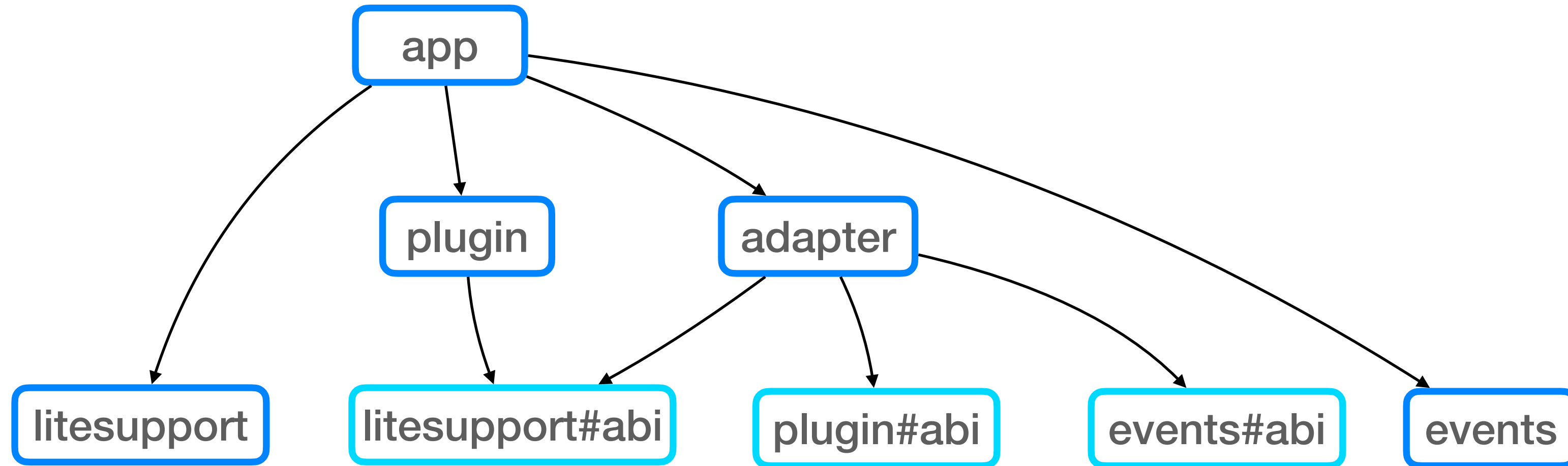
# ABI: SOURCE-ONLY-ABI



# ABI: SOURCE-ONLY-ABI



# ABI: SOURCE-ONLY-ABI



# ABI: WINS



# ABI: WINS

- ◆ **class-abi** – reduced the number of rules Buck rebuilds by **35%**

## ABI: WINS

- ◆ **class-abi** – reduced the number of rules Buck rebuilds by **35%**
- ◆ **source-abi** – reduced build times by **10%**

## ABI: WINS

- ◆ **class-abi** – reduced the number of rules Buck rebuilds by **35%**
- ◆ **source-abi** – reduced build times by **10%**
- ◆ **source-only-abi** – reduced graph depth for IG by **77%**, cache fetches by **50%** and build times by **30%**

# ABI: KOTLIN?

# ABI: KOTLIN?

- ◆ **class-abi** – possible to strip from Full Jars

# ABI: KOTLIN?

- ◆ **class-abi** – possible to strip from Full Jars
- ◆ **source-abi** – already quite problematic: type inference, inline methods, ...

# ABI: KOTLIN?

- ✦ **class-abi** – possible to strip from Full Jars
- ✦ **source-abi** – already quite problematic: type inference, inline methods, ...
  - Can use Kotlin `jvm-abi-gen` compiler plugin

# ABI: KOTLIN?

- ✦ **class-abi** – possible to strip from Full Jars
- ✦ **source-abi** – already quite problematic: type inference, inline methods, ...
  - Can use Kotlin `jvm-abi-gen` compiler plugin
  - Still under development



# ABI: KOTLIN?

- ✦ **class-abi** – possible to strip from Full Jars
- ✦ **source-abi** – already quite problematic: type inference, inline methods, ...
  - Can use Kotlin `jvm-abi-gen` compiler plugin
  - Still under development
- ✦ **source-only-abi** – ...

# ABI: KOTLIN?

- ◆ **class-abi** – possible to strip from Full Jars
- ◆ **source-abi** – already quite problematic: type inference, inline methods, ...
  - Can use Kotlin `jvm-abi-gen` compiler plugin
  - Still under development
- ◆ **source-only-abi** – ...make a wish for Santa

# SHIP CODE



# ANDROID PIPELINE



# ANDROID PIPELINE

---

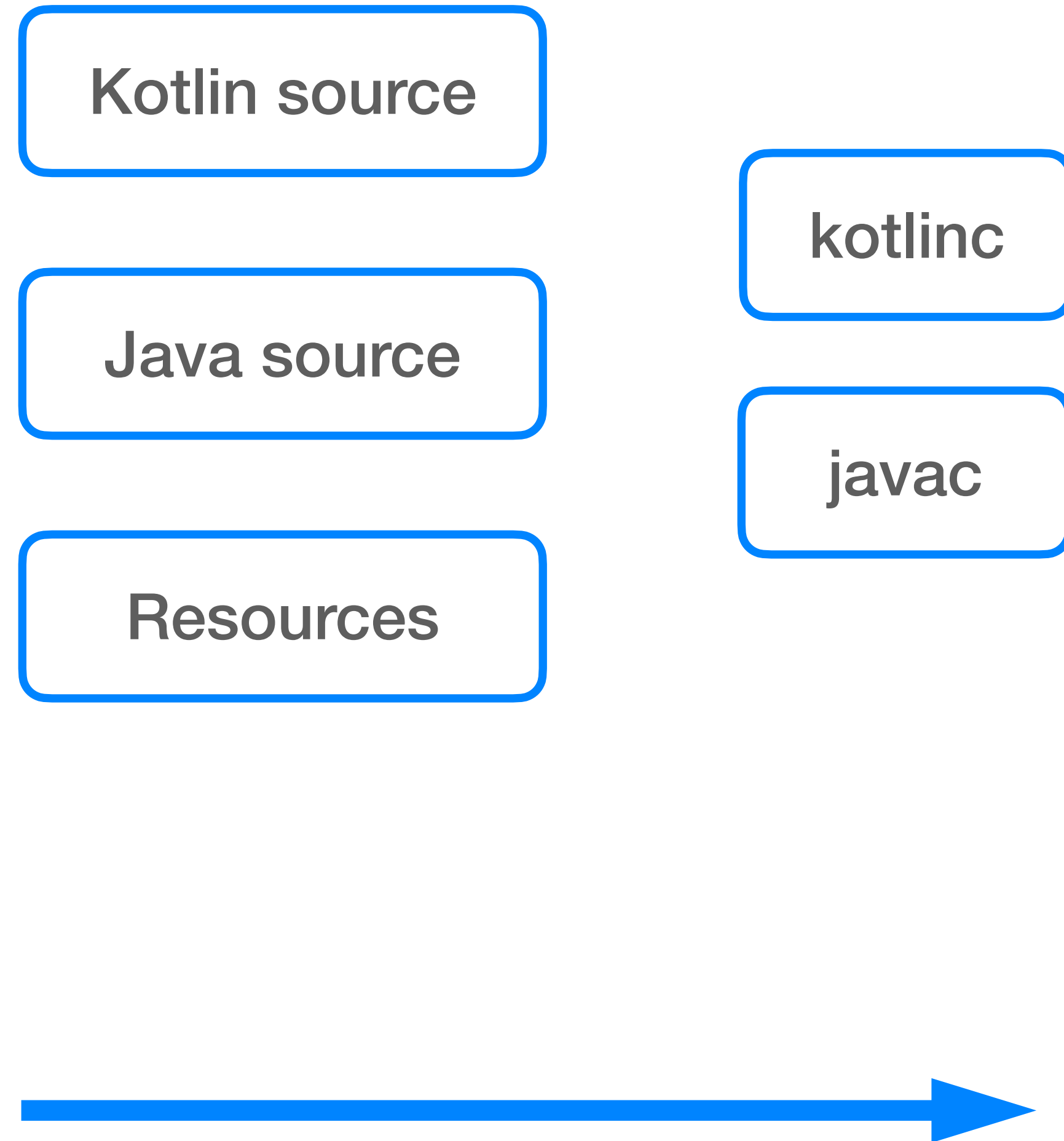
Kotlin source

Java source

Resources

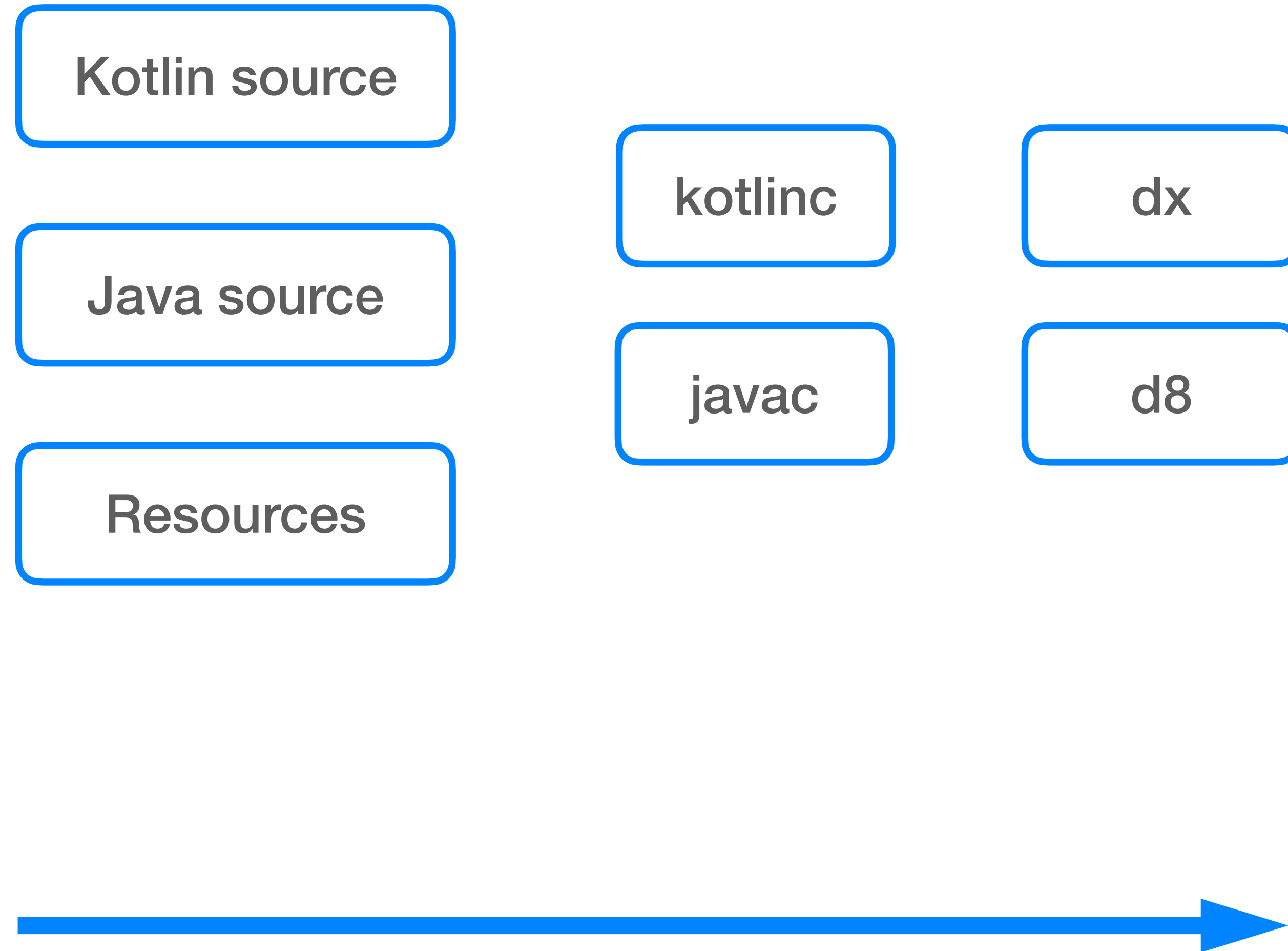
# ANDROID PIPELINE

---



# ANDROID PIPELINE

---



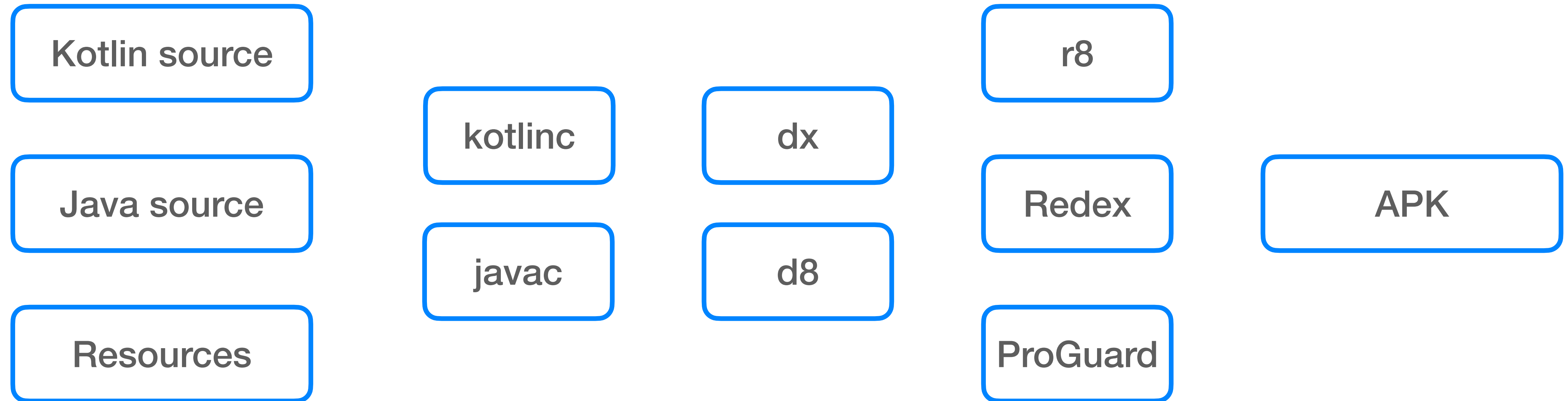
# ANDROID PIPELINE

---



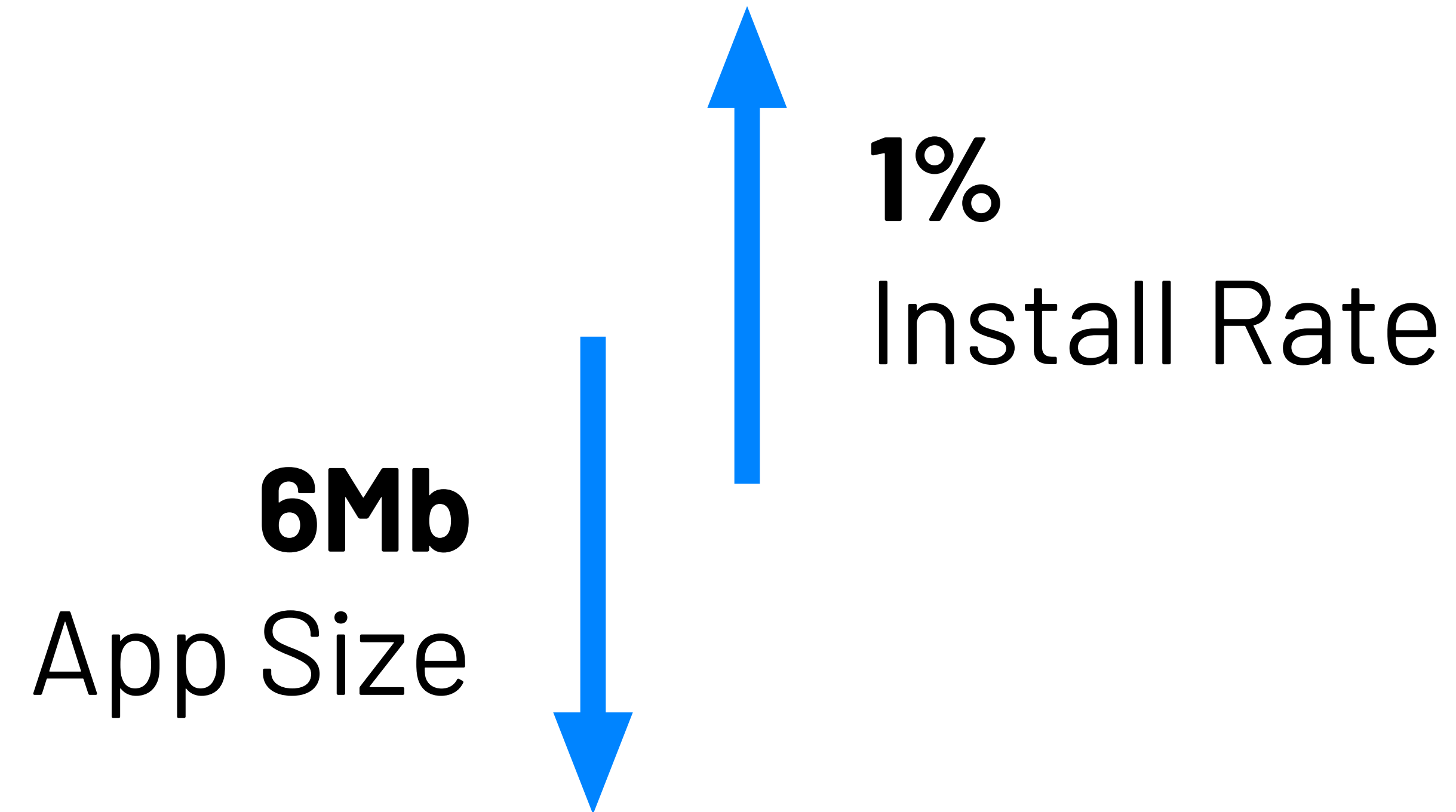


# ANDROID PIPELINE



# ANDROID BYTECODE OPTIMIZERS

---



# ANDROID BYTECODE OPTIMIZERS



R8

ProGuard



Redex

# BYTECODE OPTIMIZATIONS. INLINING

```
fun foo(): Int = 42
```

```
class Bar {  
    fun baz(): Int {  
        foo()  
    }  
}
```

# BYTECODE OPTIMIZATIONS. INLINING

---

```
class Bar {  
    fun baz(): Int = 42  
}
```

# BYTECODE OPTIMIZATIONS. REMOVE UNREACHABLE

```
fun foo(): Int = 42
fun bar(): String = "Hello, World!"

fun main() {
    println( bar() )
}
```

# BYTECODE OPTIMIZATIONS. REMOVE UNREACHABLE

```
fun bar(): String = "Hello, World!"
```

```
fun main() {  
    println( bar() )  
}
```

# BYTECODE OPTIMIZATIONS. INLINING AGAIN

---

```
fun main() {  
    println( "Hello, World!" )  
}
```



# KOTLIN SUGAR

---

## Lambda Expressions

```
fun foo(init: () -> String) { ... }
```

## Property delegates

```
val p: String by Delegate()
```

## Nullability

```
val x: Any? = null
```

## Data Classes

```
data class P(id: Int, name: Str)
```

## Companion objects

```
class A { companion object }
```

# KOTLIN SUGAR

---

## Lambda Expressions

```
fun foo(init: () -> String) { ... }
```

## Property delegates

```
val p: String by Delegate()
```

## Nullability

```
val x: Any? = null
```

## Data Classes

```
data class P(id: Int, name: Str)
```

## Companion objects

```
class A { companion object }
```

# LAMBDA EXPRESSIONS

---

```
fun lambda(foo: () -> Unit): Unit {  
    foo()  
}
```

## LAMBDA EXPRESSIONS. BYTECODE


```
public final class Placeholder {
    public final static lambda(Lkotlin/jvm/functions/Function0;)V
        @Lorg/jetbrains/annotations/NotNull;()
        L0
        ALOAD 0
        LDC "foo"
        INVOKESTATIC kt/j/i/Intrinsics.checkNotNullParameter
            (Ljava/lang/Object;Ljava/lang/String;)V
        L1
        LINENUMBER 1 L1
        ALOAD 0
        INVOKEINTERFACE kotlin/jvm/functions/Function0.invoke ()
            Ljava/lang/Object; (itf)
        POP
        RETURN
}
```


## LAMBDA EXPRESSIONS. DEX CODE


```
Placeholder.lambda:(Lkotlin/jvm/functions/Function0;)V
const-string v0, "foo"
invoke-static
    {v1, v0},
    Lkotlin/jvm/internal/Intrinsics;.checkNotNullParameter:
        (Ljava/lang/Object;Ljava/lang/String;)V
invoke-interface
    {v1},
    Lkotlin/jvm/functions/Function0;.invoke:()Ljava/lang/Object;
return-void
```


# LAMBDA EXPRESSIONS. AT SCALE

---


 **com.facebook.a.a.a**  
Lambda { callback\_0001() }

 **com.facebook.a.a.z**  
Lambda { callback\_0002() }

 **com.facebook.x.y.z**  
Lambda { callback\_0003() }

 **com.facebook.s.o.s**  
Lambda { callback\_0004() }

 **com.facebook.w.t.k**  
...

 **com.facebook.z.z.z**  
Lambda { callback\_9999() }

# LAMBDA EXPRESSIONS. AT SCALE

```
Lambda { callback_0001() } ... Lambda { callback_9999() }  
10'000xClass + 30'000xMethod + 10'000xInstance
```

```
class PlaceholderKt$fun$1  
    extends kotlin/Lambda  
    implements kotlin/Function0 {  
  
    public final invoke()Ljava/lang/Object;  
    <init>()V  
    static LPlaceholderKt$fun$1; INSTANCE  
    static <clinit>()V  
    @Lkotlin/Metadata; { meta }  
  
}
```

# LAMBDA EXPRESSIONS. FIXING IT

`fun$1 ... fun$10000 => Uber$fun`

`<clinit>$1 ... <clinit>$10000 => Uber$<clinit>`

`INSTANCE$1 ... INSTANCE$10000 => Uber$INSTANCE`

`invoke$1 ... invoke$10000 => Uber$invokeWithSwitch`

R8: Lambda Grouping

<https://r8.googlesource.com/r8/+/-/fd9fcdf19cb6600145852215dd45f7ecbb949255/src/main/java/com/android/tools/r8/ir/optimize/lambda/kotlin/KotlinLambdaGroup.java>

Redex: Class Merging

<https://github.com/facebook/redex/blob/379e926cd41e4f18b69ac1445b70e331ba01c0b1/opt/class-merging/ClassMergingPass.cpp>



# LAMBDA EXPRESSIONS. FIXING IT

```
val type: Int

fun Uber.<init>(int type) = when {
    1 -> fun$1.<init>(type)
    2 -> fun$2.<init>(type)
    ...
}
```

R8: Lambda Grouping

<https://r8.googlesource.com/r8/+/-/fd9fcdf19cb6600145852215dd45f7ecbb949255/src/main/java/com/android/tools/r8/ir/optimize/lambda/kotlin/KotlinLambdaGroup.java>

Redex: Class Merging

<https://github.com/facebook/redex/blob/379e926cd41e4f18b69ac1445b70e331ba01c0b1/opt/class-merging/ClassMergingPass.cpp>

# LAMBDA EXPRESSIONS. FIXING IT

```
fun Uber.invoke() {  
    val type = this.type  
    when(type) {  
        1 -> fun$1.invoke(type)  
        2 -> fun$2.invoke(type)  
  
        ...  
        else -> super.invoke()  
    }  
}
```

R8: Lambda Grouping

<https://r8.googlesource.com/r8/+/-/fd9fcdf19cb6600145852215dd45f7ecbb949255/src/main/java/com/android/tools/r8/ir/optimize/lambdakotlin/KotlinLambdaGroup.java>

Redex: Class Merging

<https://github.com/facebook/redex/blob/379e926cd41e4f18b69ac1445b70e331ba01c0b1/opt/class-merging/ClassMergingPass.cpp>

## LAMBDA EXPRESSIONS. STILL FIXING IT

```
val foo = Uber$fun()  
  
if (foo is fun$0001) {  
    ...  
}  
  
val bar = foo as fun$0001
```

# LAMBDA EXPRESSIONS. WHY NOT JUST

```
inline fun lambda(foo: () -> Unit): Unit {  
    foo()  
}
```

0xClass + 0xMethod + 0xInstance

# DATA CLASSES

---

```
data class Person(  
    val id: Long,  
    val name: String,  
    val job: Job?  
)  
  
p1 == p2  
set.add(p1)  
val p3 = p1.copy(id = 42)
```

## DATA CLASSES. BYTECODE

```
class Person
  <init>(Ljava/lang/String;LJob;)V
  getId() | getName() | getJob()
  component1()J
  component2()Ljava/lang/String;
  component3()LJob;
  copy()
  synthetic copy$default
  toString()Ljava/lang/String;
  hashCode()I
  equals(Ljava/lang/Object;)Z
```

## DATA CLASSES. BYTECODE

```
class Person  
  getId() | getName() | getJob()
```

Inlining Pass (R8 / Redex)

## DATA CLASSES. BYTECODE

```
class Person  
—component1()J  
—component2()Ljava/lang/String;  
—component3()LJob;
```

Inlining Pass

Remove Unreachable Pass (R8 / Redex)



## DATA CLASSES. BYTECODE

```
class Person  
—copy()  
—synthetic copy$default
```

Remove Unreachable Pass (R8 / Redex)

## DATA CLASSES. HARD ONES

```
class Person
  toString()Ljava/lang/String;
  hashCode()I
  equals(Ljava/lang/Object;)Z
```

# DATA CLASSES. TOSTRING

```
#9          : (in LPerson;)
name       : 'toString'
type      : '()Ljava/lang/String;'
access    : 0x0001 (PUBLIC)
code      : -
registers : 4
ins       : 1
outs      : 3
insns size : 45 16-bit code units
0004d0:      |[0004d0] Person.toString:()Ljava/lang/String;
0004e0: 2200 0600 |0000: new-instance v0, Ljava/lang/StringBuilder; // type@0006
0004e4: 7010 0f00 0000 |0002: invoke-direct {v0}, Ljava/lang/StringBuilder;<init>:()V // method@000f
0004ea: 1a01 1b00 |0005: const-string v1, "Person(id=" // string@001b
0004ee: 6e20 1200 1000 |0007: invoke-virtual {v0, v1}, Ljava/lang/StringBuilder;.append:(Ljava/lang/String;)Ljava/lang/StringBuilder; //
0004f4: 5331 0000 |000a: iget-wide v1, v3, LPerson;.id:J // field@0000
0004f8: 6e30 1000 1002 |000c: invoke-virtual {v0, v1, v2}, Ljava/lang/StringBuilder;.append:(J)Ljava/lang/StringBuilder; // method@0010
0004fe: 1a01 0a00 |000f: const-string v1, ", name=" // string@000a
000502: 6e20 1200 1000 |0011: invoke-virtual {v0, v1}, Ljava/lang/StringBuilder;.append:(Ljava/lang/String;)Ljava/lang/StringBuilder; //
000508: 5431 0200 |0014: iget-object v1, v3, LPerson;.name:Ljava/lang/String; // field@0002
00050c: 6e20 1200 1000 |0016: invoke-virtual {v0, v1}, Ljava/lang/StringBuilder;.append:(Ljava/lang/String;)Ljava/lang/StringBuilder; //
000512: 1a01 0900 |0019: const-string v1, ", job=" // string@0009
000516: 6e20 1200 1000 |001b: invoke-virtual {v0, v1}, Ljava/lang/StringBuilder;.append:(Ljava/lang/String;)Ljava/lang/StringBuilder; //
00051c: 5431 0100 |001e: iget-object v1, v3, LPerson;.job:LJob; // field@0001
000520: 6e20 1100 1000 |0020: invoke-virtual {v0, v1}, Ljava/lang/StringBuilder;.append:(Ljava/lang/Object;)Ljava/lang/StringBuilder; //
000526: 1a01 0800 |0023: const-string v1, ")" // string@0008
00052a: 6e20 1200 1000 |0025: invoke-virtual {v0, v1}, Ljava/lang/StringBuilder;.append:(Ljava/lang/String;)Ljava/lang/StringBuilder; //
000530: 6e10 1300 0000 |0028: invoke-virtual {v0}, Ljava/lang/StringBuilder;.toString:()Ljava/lang/String; // method@0013
000536: 0c00 |002b: move-result-object v0
000538: 1100 |002c: return-object v0
```

# DATA CLASSES. HARD ONES

```
@DataClassGenerate(  
    toString = Mode.NO,  
    equalsHashCode = Mode.YES  
)  
data class Person(  
    val id: Long,  
    val name: String,  
    val job: Job?  
)
```

## GUARD METRICS. BENCHMARKING

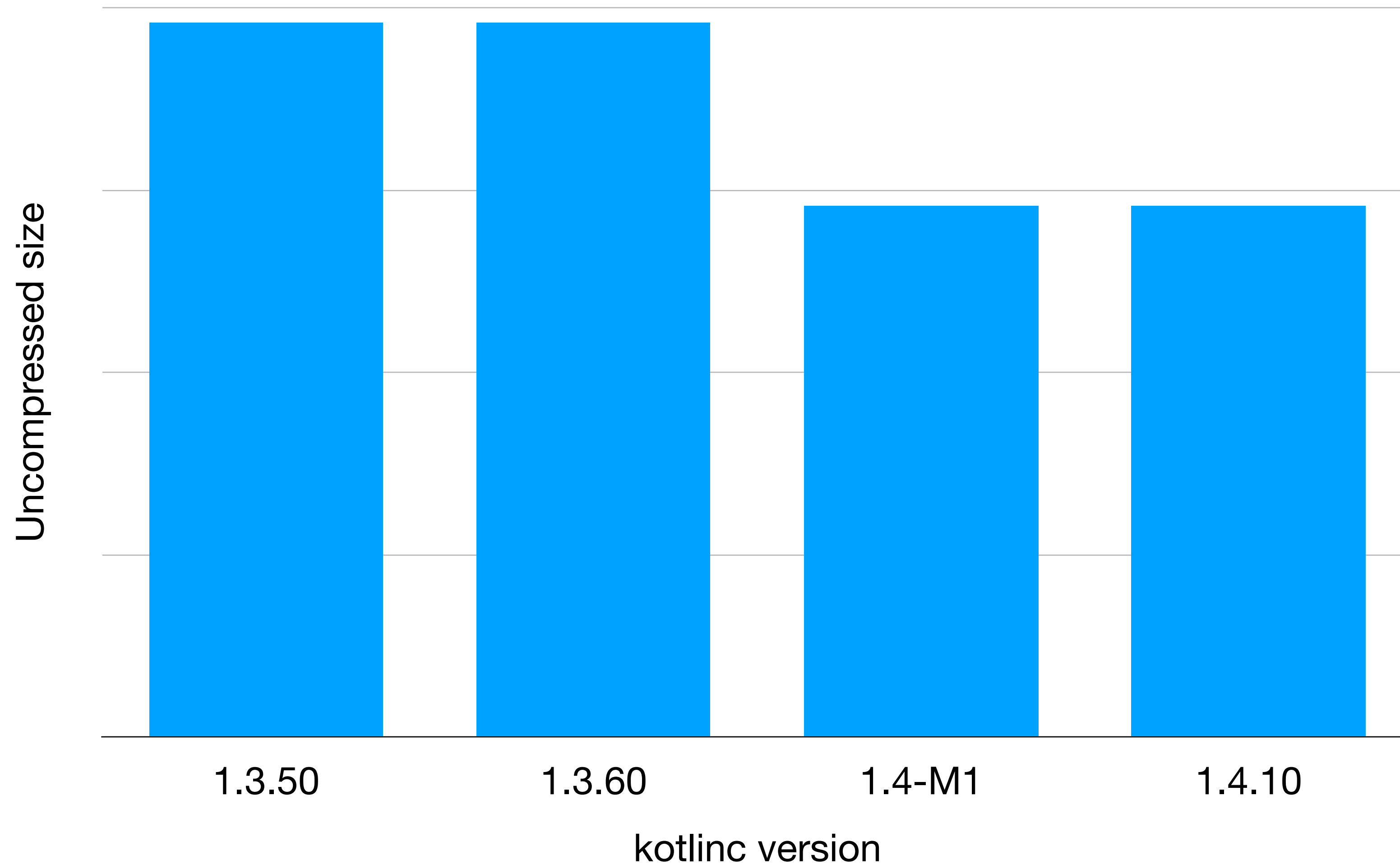
```
val lazyProp: String by lazy { "lazy string" }
```

## GUARD METRICS. BENCHMARKING

```
val lazyProp: String by lazy { "lazy string" }
{
  "normal": {
    "test_name": "Kotlin Lazy Delegate",
    "compiler": "Kotlinc 1.X.YY",
    "optimizer": "Redex"
  },
  "int": {
    "source_code_loc": 14,
    "compiler_time": 2271,
    "optimizer_time": 1702,
    "optimizer_dex_size_compressed": 481,
    "optimizer_dex_size_uncompressed": 764,
    "optimizer_method_ref_count": "3",
    "optimizer_class_count": 2
  },
}
```

# GUARD METRICS. BENCHMARKING

```
val lazyProp: String by lazy { "lazy string" }
```



# TAKEAWAYS

---

- ◆ Kotlin adoption at scale is very different – expect it to be a marathon, not a sprint
- ◆ Any small inefficiency at scale has huge impact
- ◆ Developer Happiness is worth it! Hiring becomes easier



# LINKS

---

- ◆ Ktfmt: [github.com/facebookincubator/ktfmt](https://github.com/facebookincubator/ktfmt)
- ◆ BUCK & ABI optimisations: [engineering.fb.com/2017/11/09/android/rethinking-android-app-compilation-with-buck](https://engineering.fb.com/2017/11/09/android/rethinking-android-app-compilation-with-buck)
- ◆ Redex Optimisations: [fbredex.com/docs/passes](https://fbredex.com/docs/passes)

# THANK YOU

---

Sergei Rybalkin  
@lightdelay

Sergey Ryabov  
@colriot

FACEBOOK

