Google Developers
Experts

Make it fast

Denis Nek
Google Developer Expert
@nekdenis

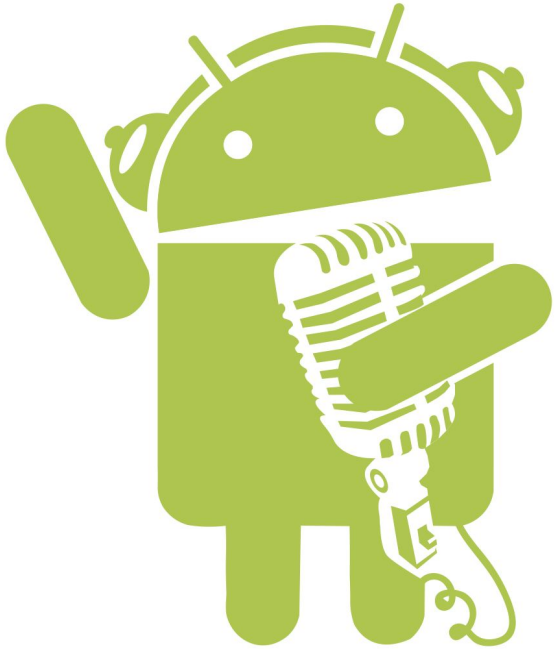
Google Developers Experts

90Seconds.com

100+ Countries 17,000+ Videos 16,000+ Creators 2,400+ Brands

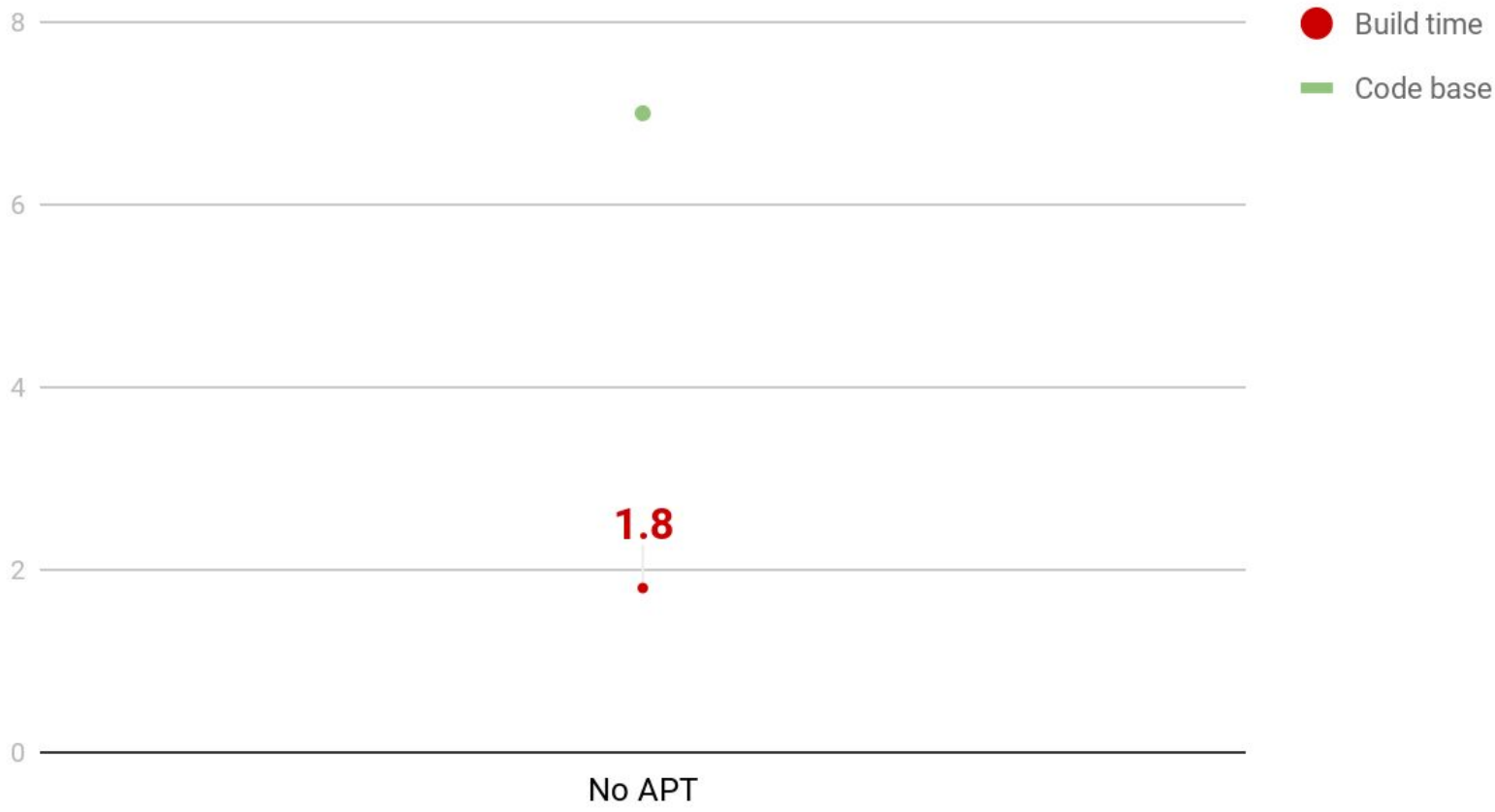
AndroidDev Podcast

Google Developers Experts

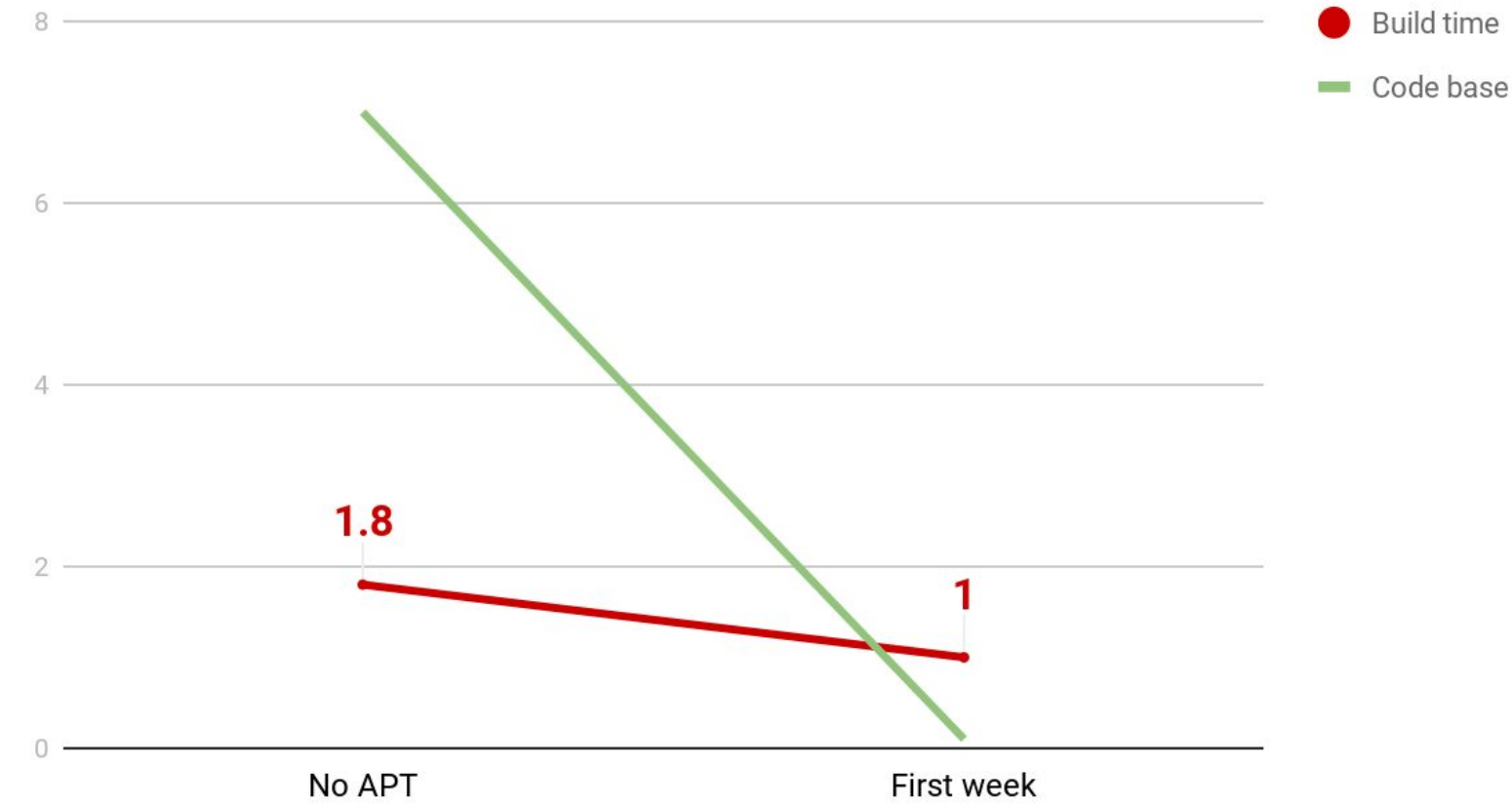


#AndroidDevPodcast

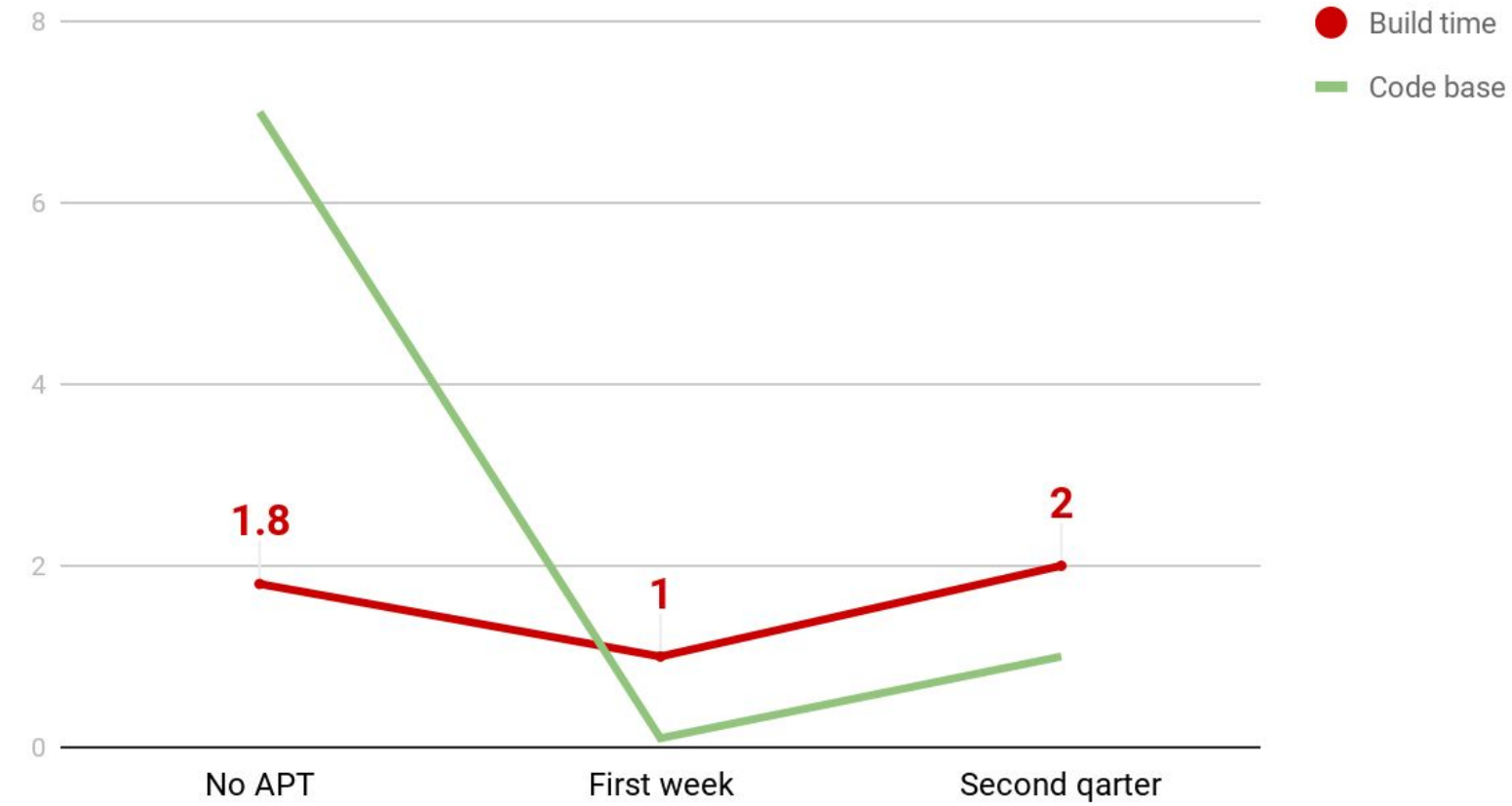
Clean build time minutes



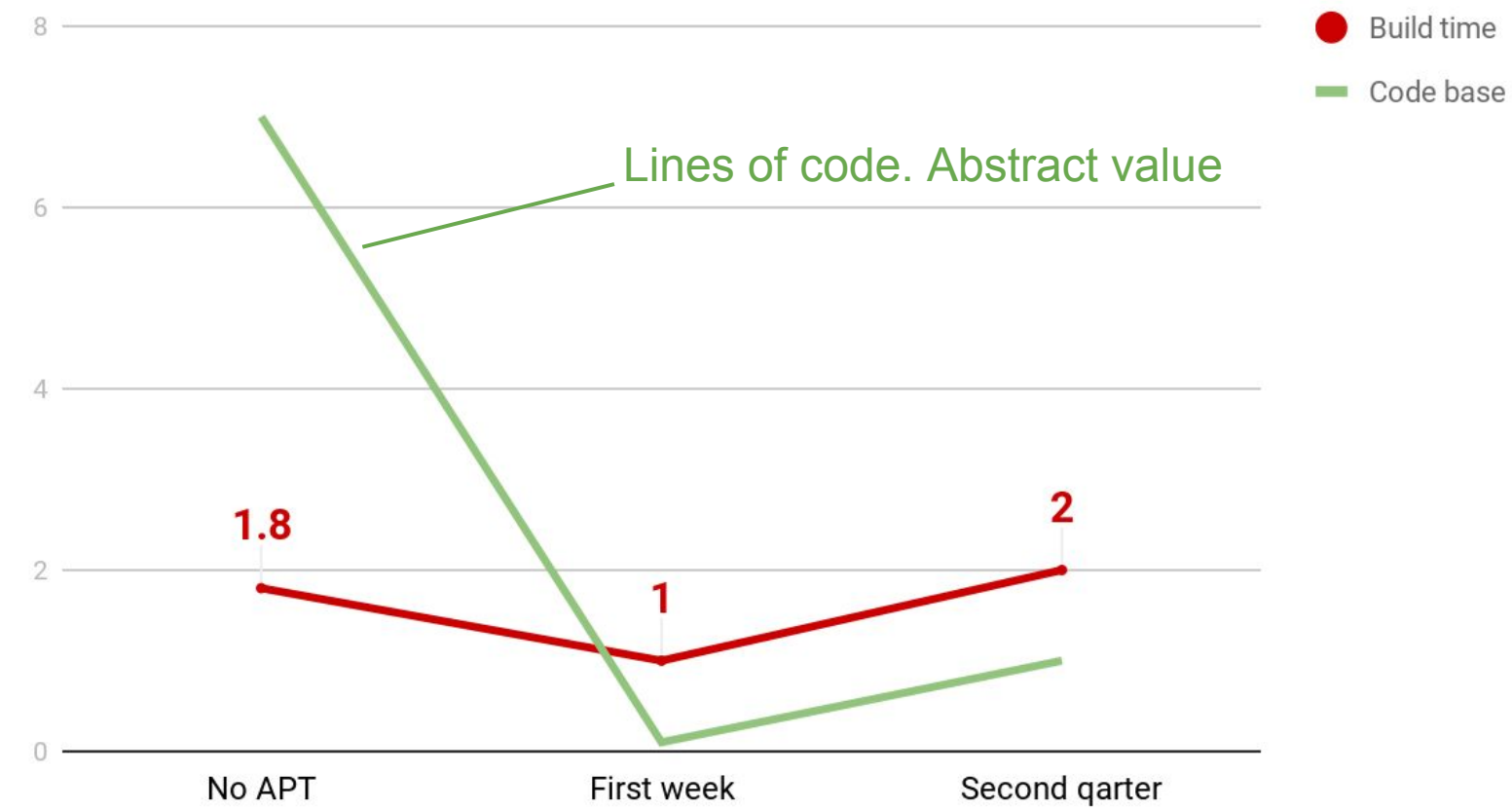
Clean build time minutes



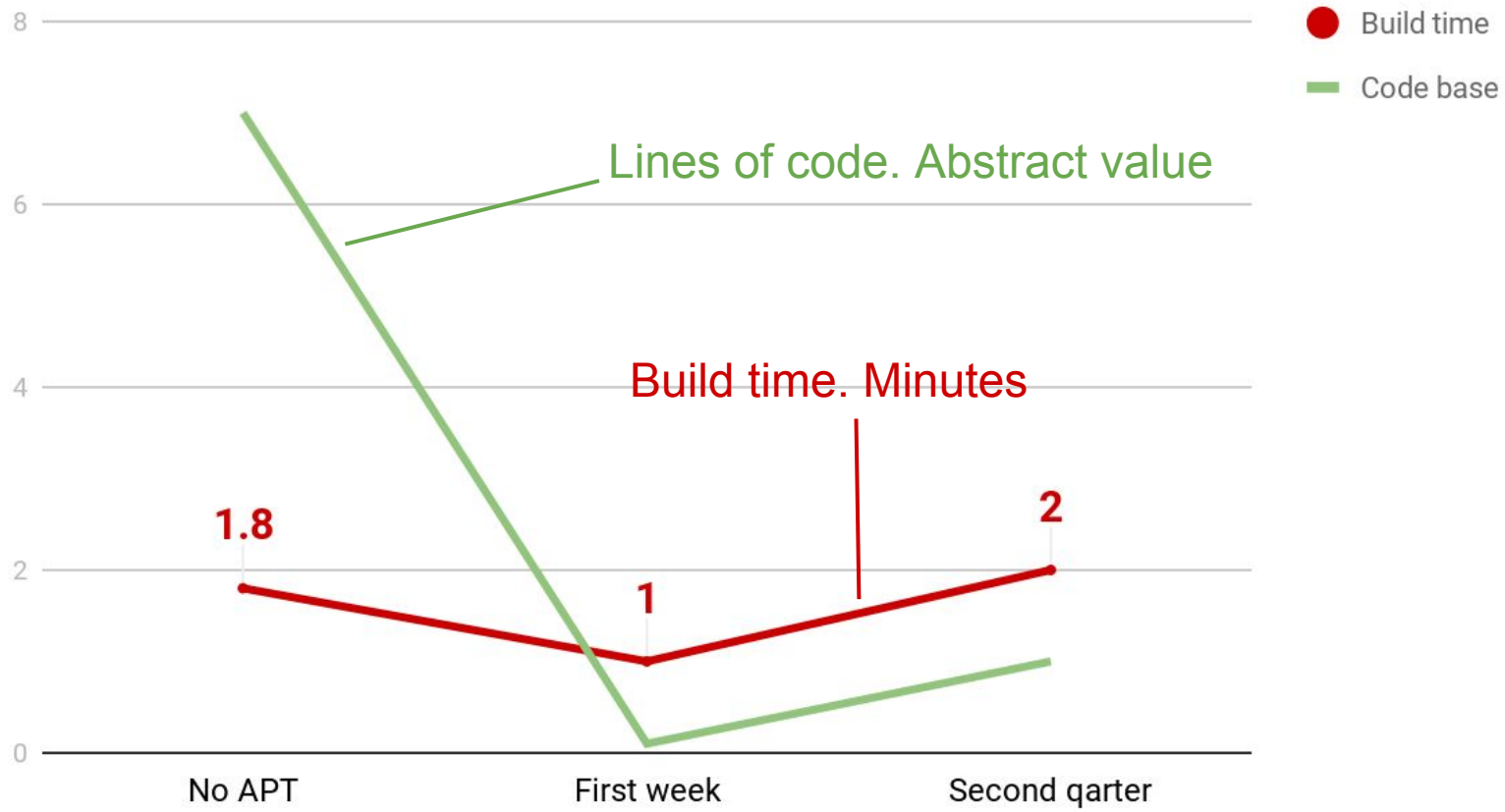
Clean build time minutes



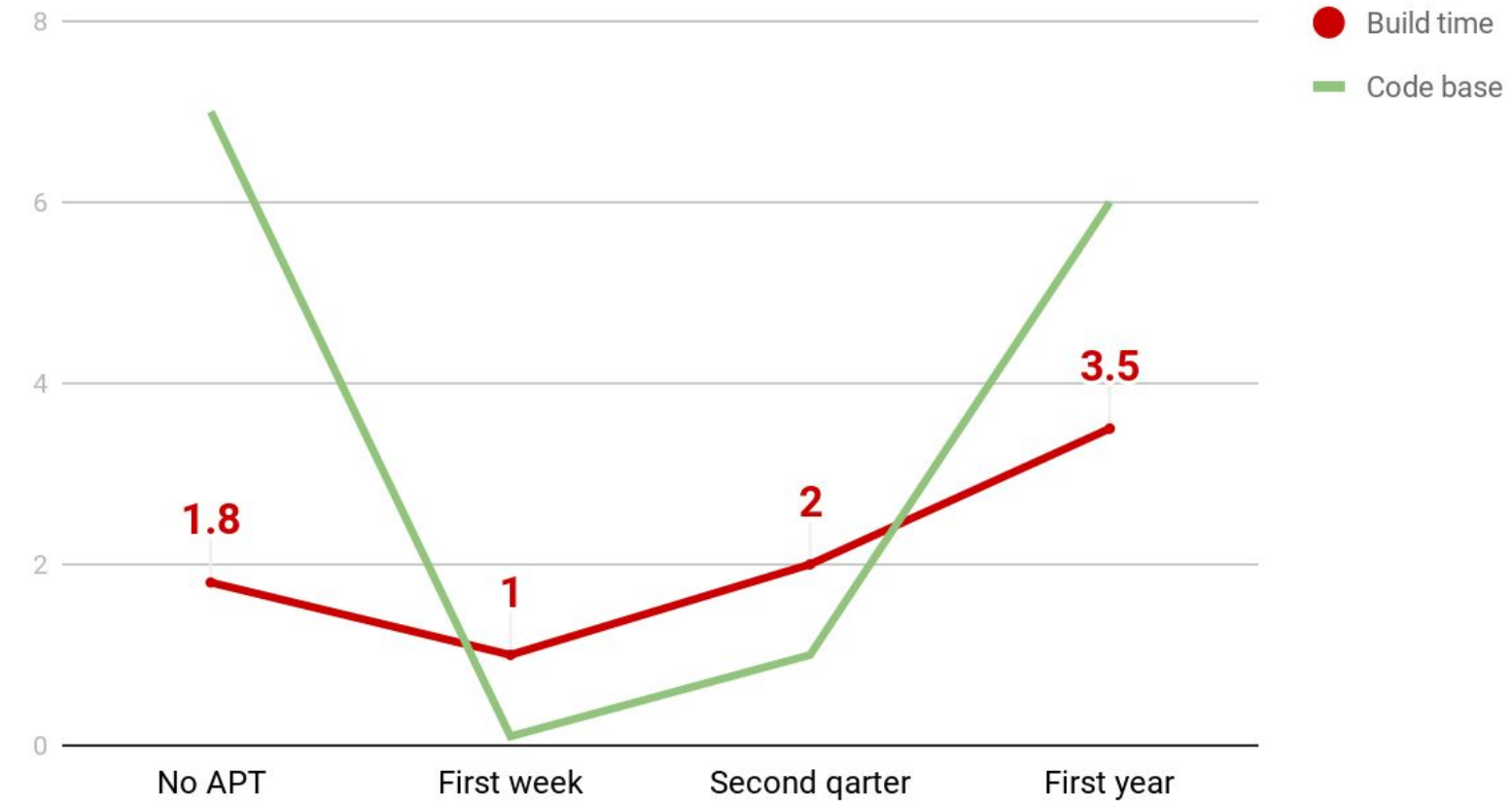
Clean build time minutes



Clean build time minutes



Clean build time minutes



Clean build time **before APT**: 1.8 min

What’s common?

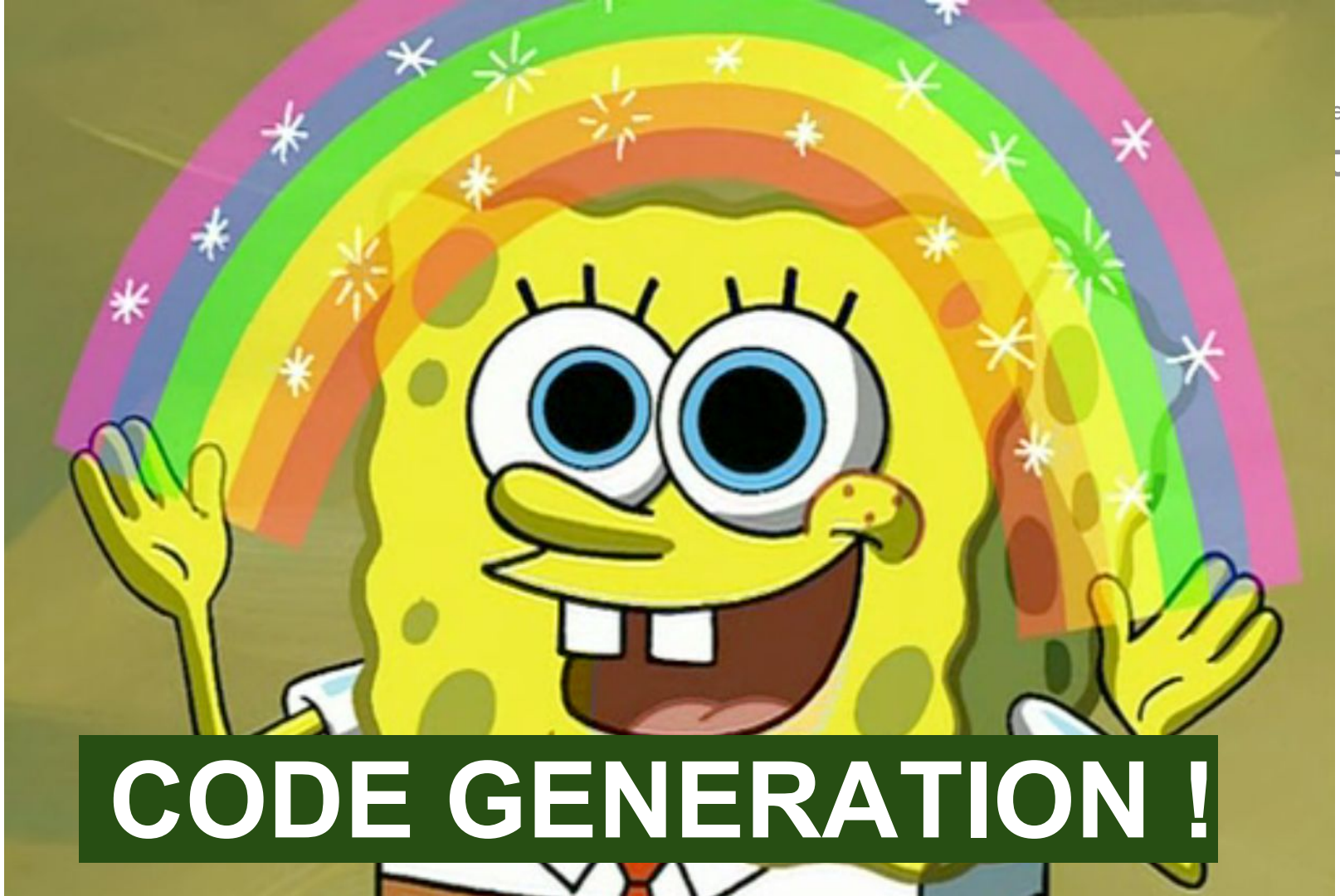
Dagger 2

Butter Knife

DataBinding

Room

APT and KAPT


CODE GENERATION !

APT benefits

- Less code
- Easy to read
- Tested
- Hipsterish

Plain Java way

lopers

S

Google Developers Experts

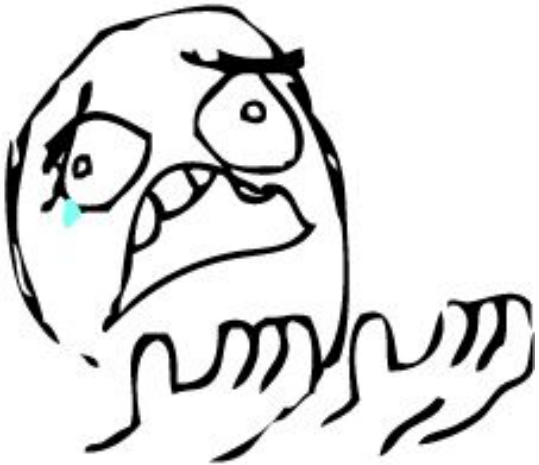
Clean build time **before APT**: 1.8 min

Google Developers Experts

Clean build time **with APT**: 3.5 min

Google Developers Experts

Incremental change of
one line of code:
3.5 min



Google Developers Experts

Annotation processing is not INCREMENTAL

Single module architecture:

Google Developers Experts

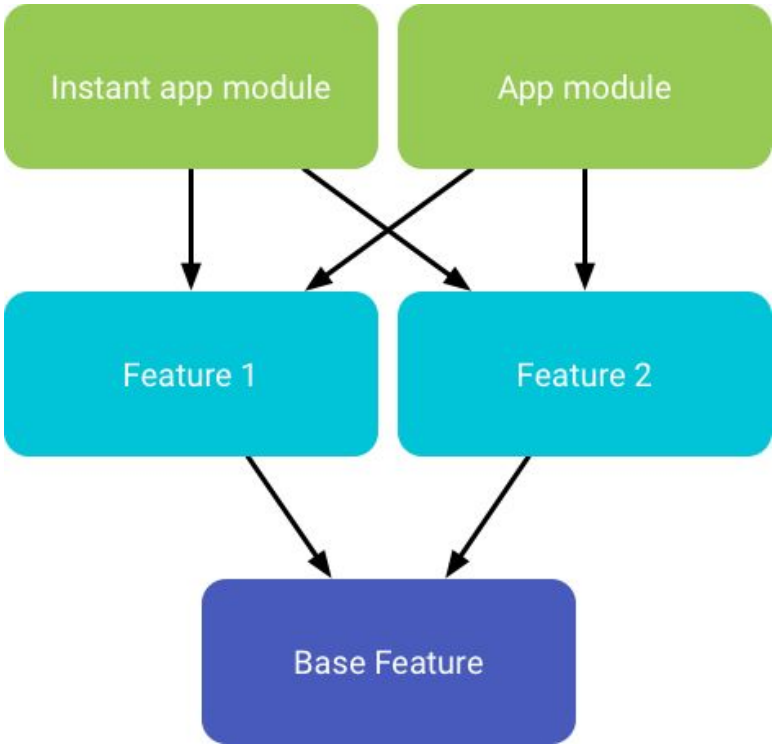
Single module architecture:

Google Developers Experts

Google Developers
Experts

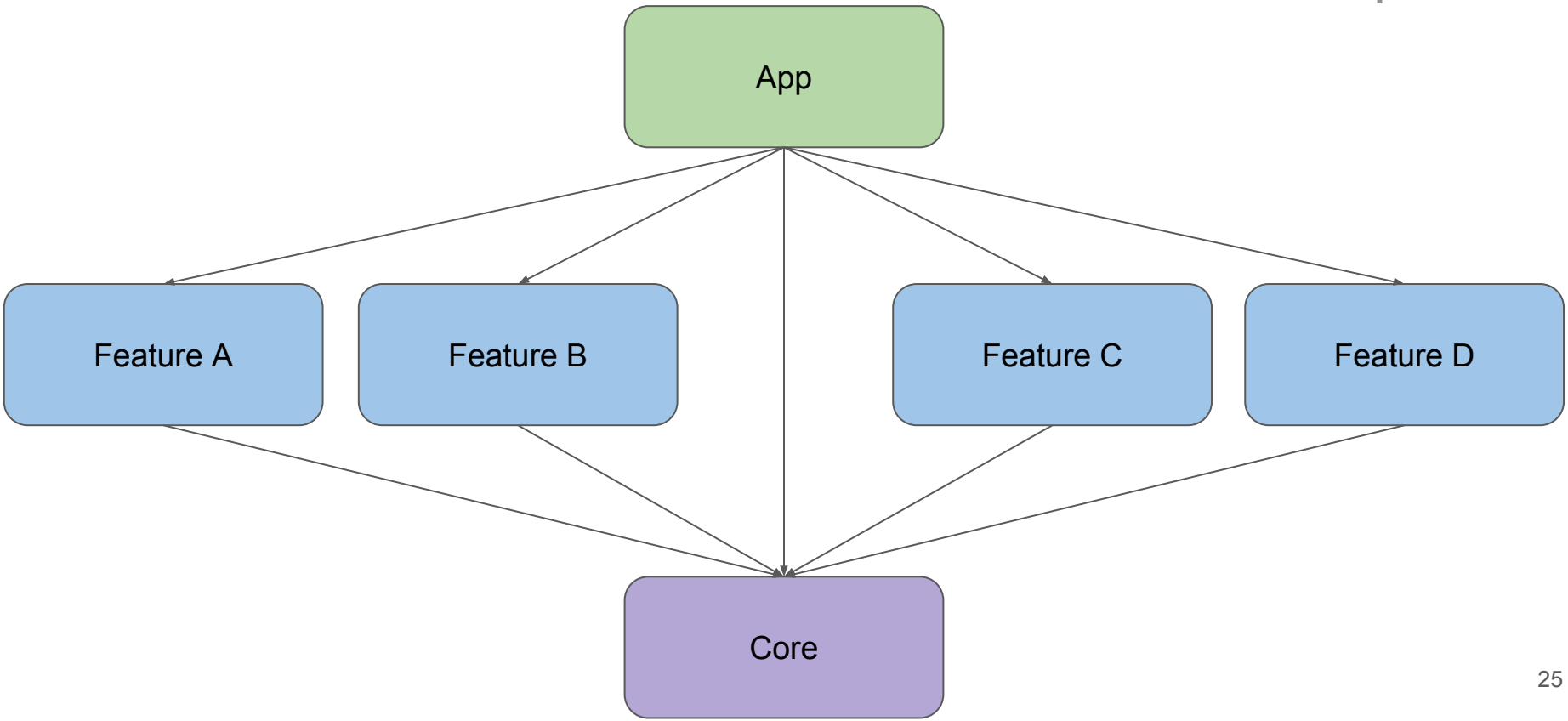
Several modules architecture:

Google Developers Experts



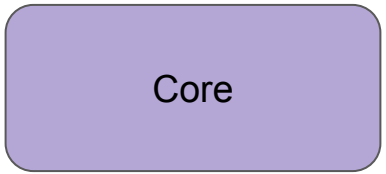
Several modules architecture:

Google Developers Experts



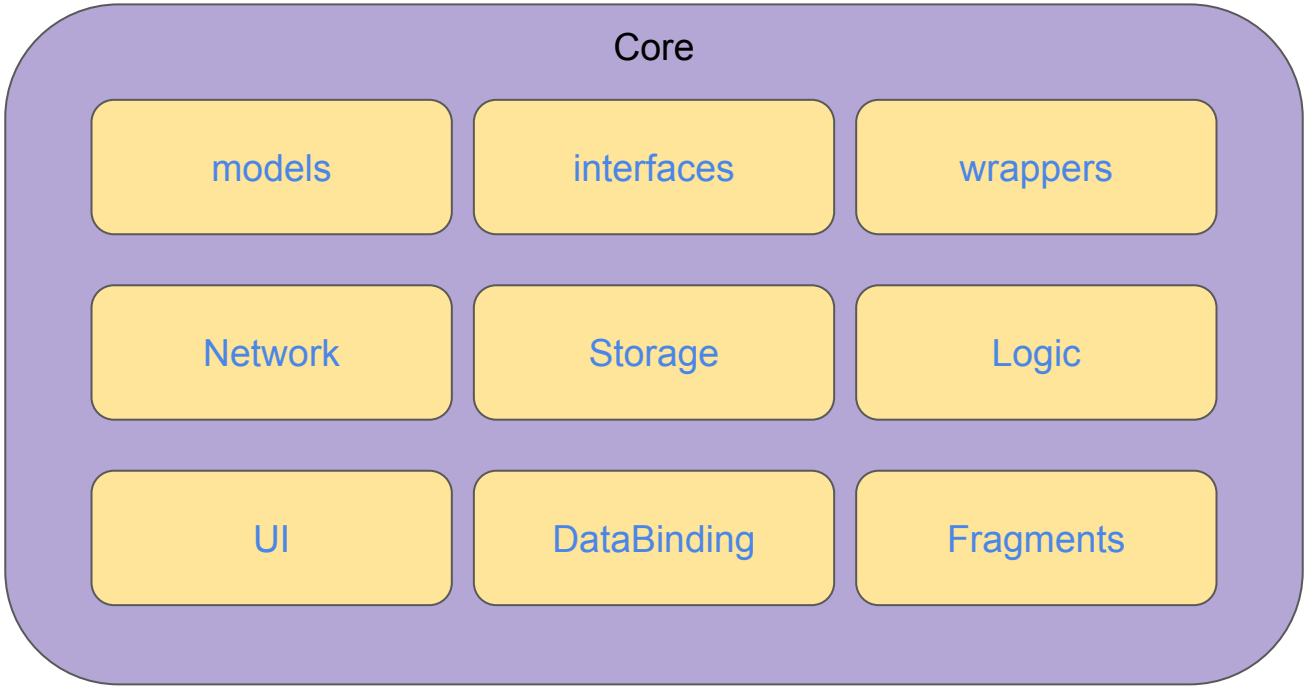
Several modules architecture:

Google Developers Experts



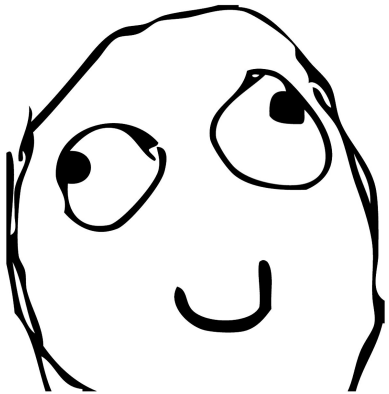
Several modules architecture:

Google Developers Experts



Google Developers Experts

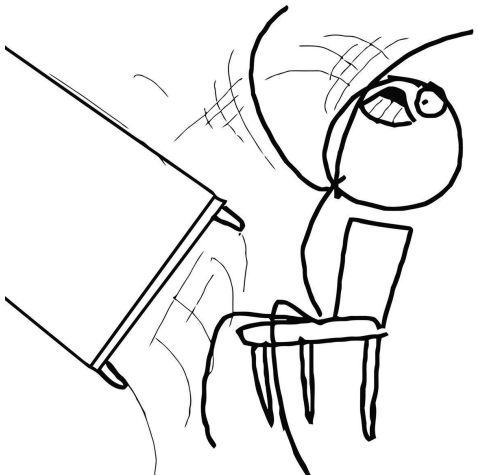
Incremental change of one line of code **in feature**: 3 min



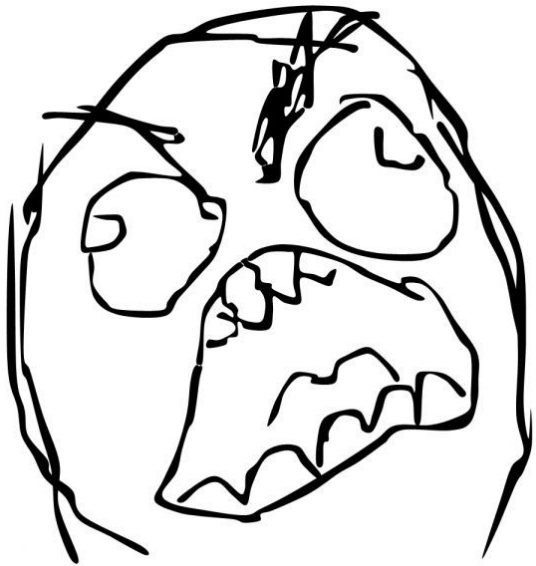
Google Developers Experts

Incremental change of one line of code **in core**:

7.5 min



Google Developers Experts

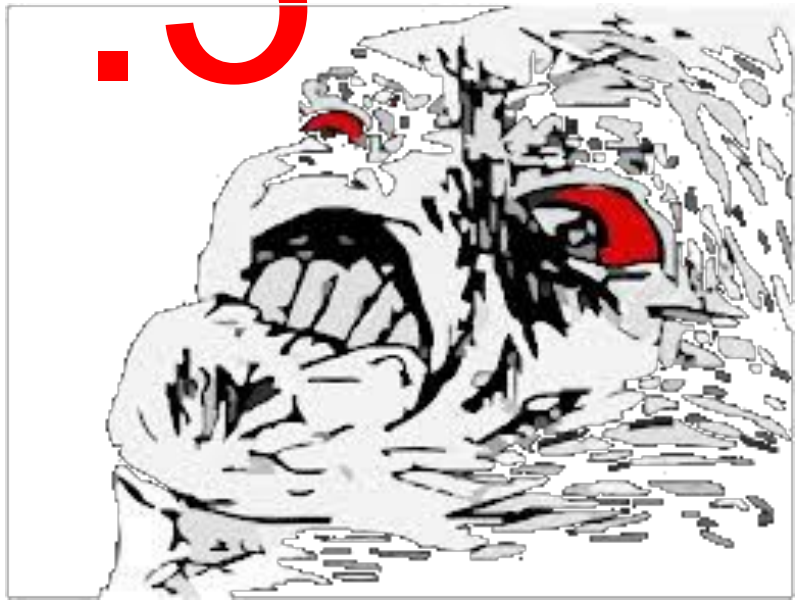


in core:

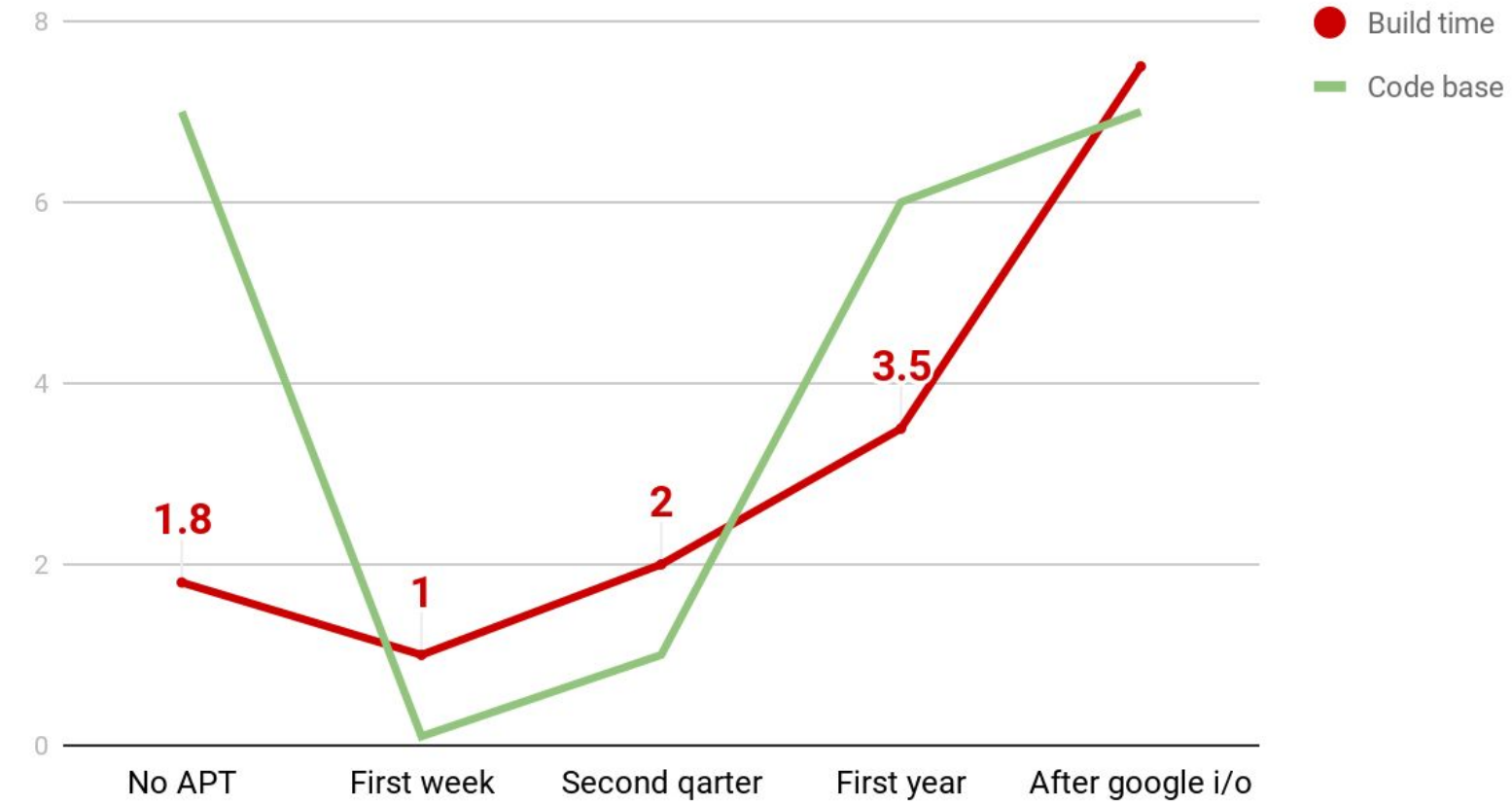
7.5 min

Google Developers Experts

7.5



Clean build time minutes

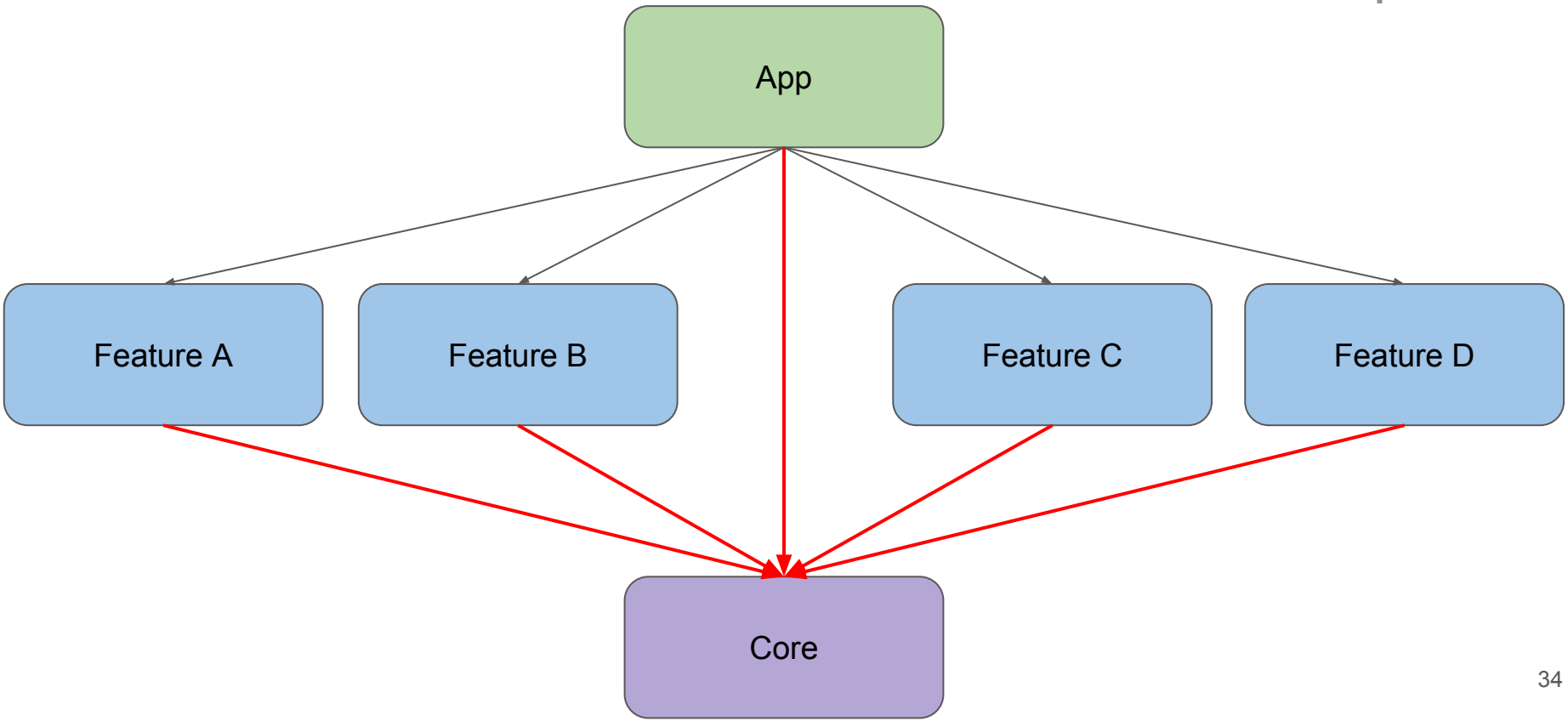


Google Developers Experts

Why that happened?

Several modules architecture:

Google Developers Experts



Modules without APT:

Google Developers Experts

Google Developers Experts

Annotation processing is not INCREMENTAL

Life has no more sense

Google Developers Experts

Small refactoring

Google Developers Experts

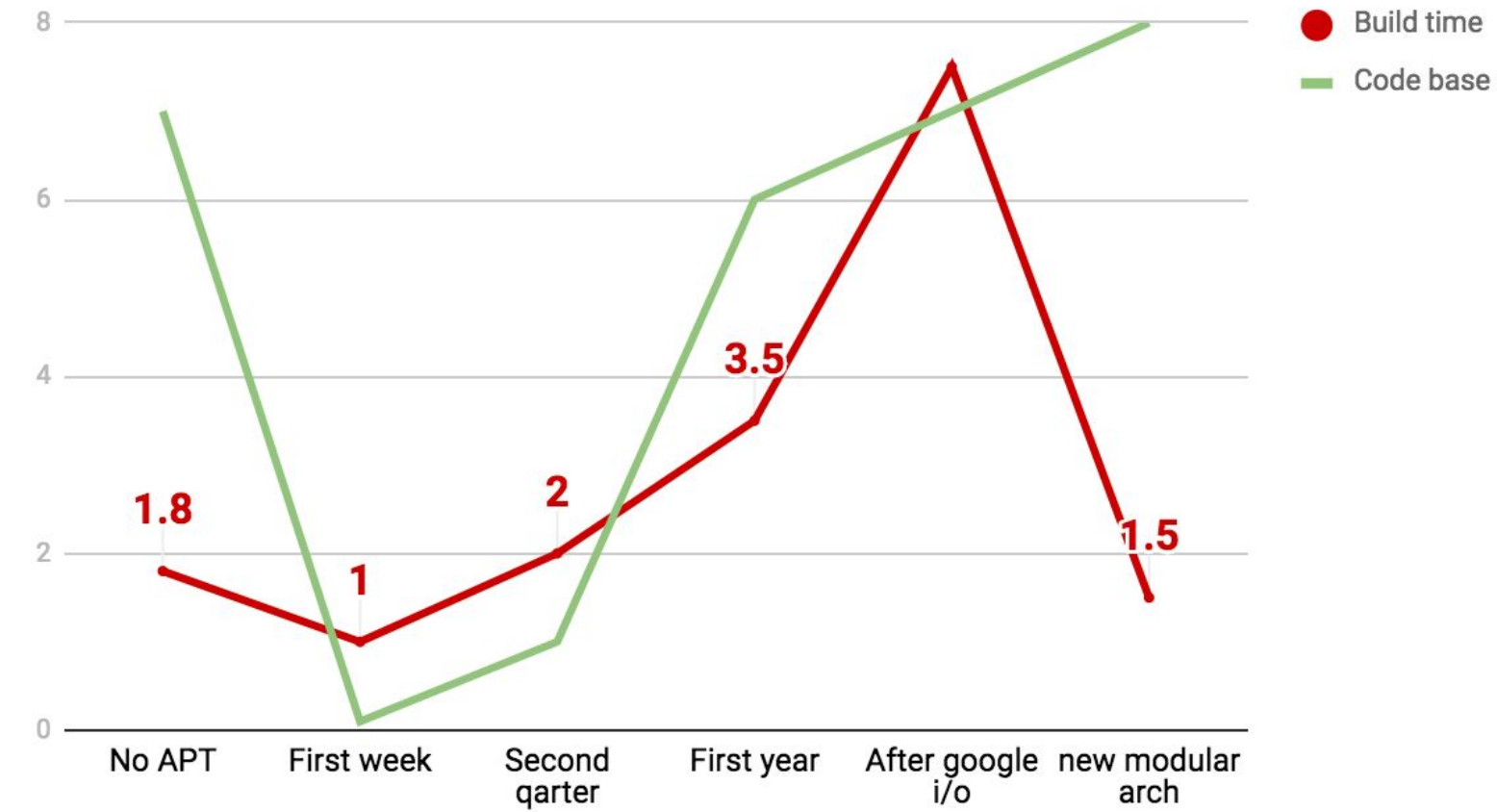
Non revertable refactoring

Google Developers Experts

Almost new project

Google Developers Experts

Clean build time minutes



Google Developers Experts

Investigation

Google Developers Experts

Gradle

Gradle --scan

Google Developers Experts

./gradlew clean assembleDebug --scan

Gradle --scan

Google Developers Experts

./gradlew clean assembleDebug --scan

Build Scan d ✓ 90Seconds_fixes cl clBC assDD Mar 17, 2018 10:45:34 PM +08

- Summary
- Console log
- Timeline
- Performance
- Projects
- Dependencies
- Plugins
- Switches
- Infrastructure

Started today at 10:45:34 PM +08, finished today at 10:53:06 PM +08
Gradle 4.4-rc-3, Build scan plugin 1.10.3, Android Gradle plugin 3.0.0

Explore console log

259 tasks executed in **7 projects** in **7m 17.766s**

:app:compileDevDebugKotlin	1m 2.069s
:app:kaptDevDebugKotlin	47.606s
:core:compileDevDebugKotlin	41.240s

Explore timeline

7 min 32 sec total build time

Initialization and configuration	14.007s
Task execution	7m 17.766s

Explore performance

Tasks count

Google Developers Experts

Path	Started after	Duration
:appinjector:clean	0.000s	6.713s
:auth:clean	0.000s	0.749s
:clean	0.000s	0.018s
:core:clean	0.000s	7.410s
:coreui:clean	0.001s	4.587s
:creator-signup:clean	0.001s	0.019s
:linkedin-sdk:clean	0.001s	0.344s
:location:clean	0.001s	3.591s
:login:clean	0.020s	4.631s
:navigation:clean	0.021s	3.175s
:network:clean	0.346s	1.468s
:repo:clean	0.749s	1.438s
:settings:clean	1.814s	4.138s
:splash:clean	2.187s	3.496s
:test-tools:clean	3.197s	0.028s
:tools:clean	3.227s	1.798s
:upload:clean	3.592s	1.087s
:user-profile:clean	4.589s	2.889s
:workflow:clean	4.658s	1.480s

Original build (incremental)

Google Developers Experts

259 tasks executed in **7 projects** in **7m 17.766s**

:app:compileDevDebugKotlin	1m 2.069s
:app:kaptDevDebugKotlin	47.606s
:core:compileDevDebugKotlin	41.240s

Original build

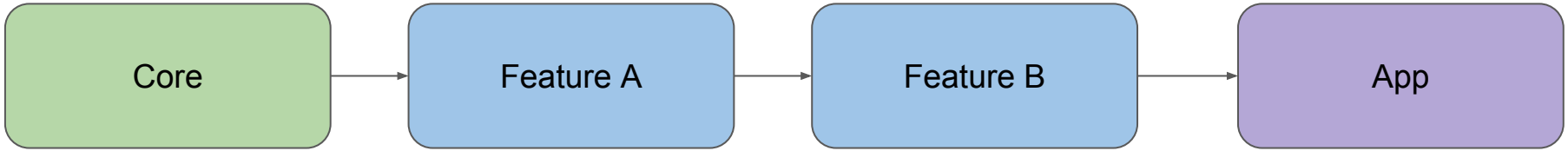
Google Developers Experts

7 projects

90Seconds_fixes +6

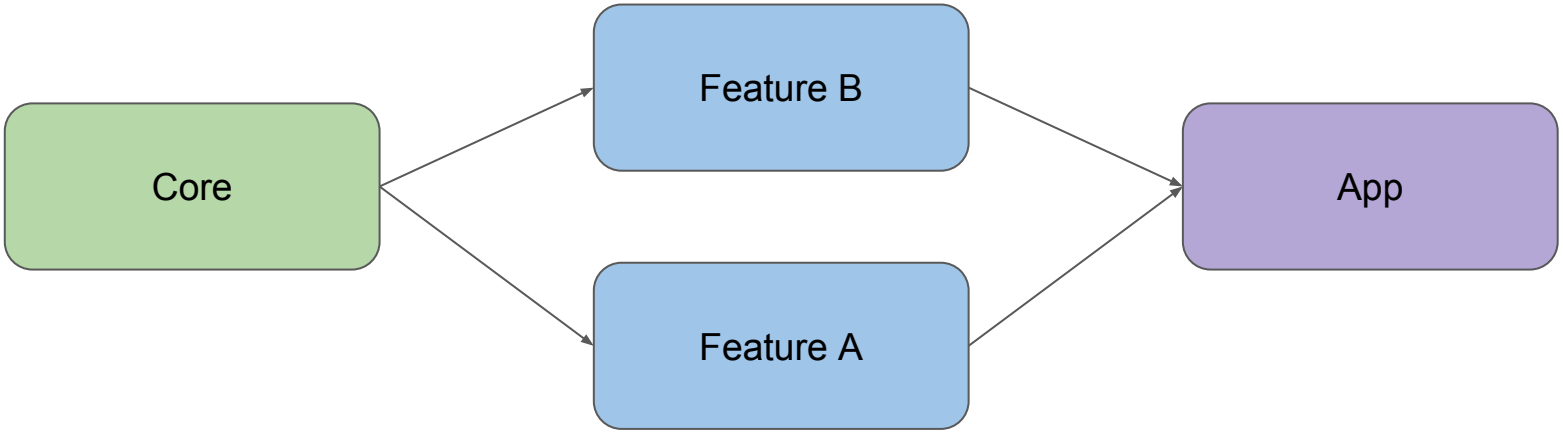
- activities
- app
- core
- package-order
- project-details
- settings

Google Developers Experts



Google Developers Experts

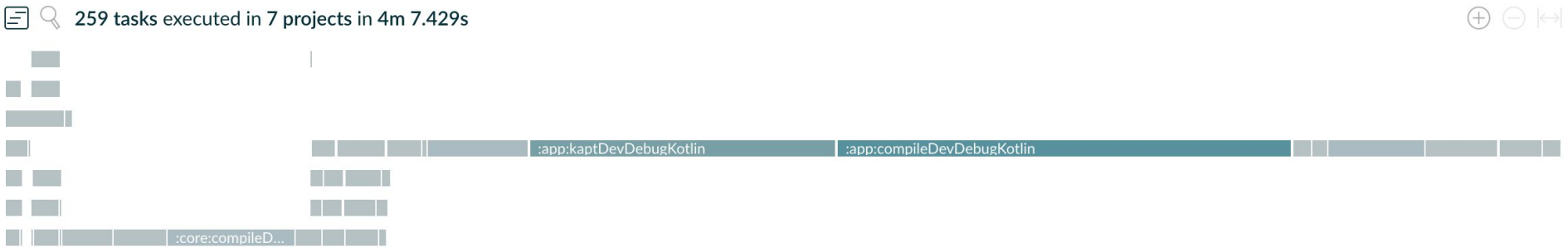
org.gradle.parallel=true



Parallel original build (incremental)

Google Developers Experts

org.gradle.parallel=true



Parallel original build (clean)

Google Developers Experts

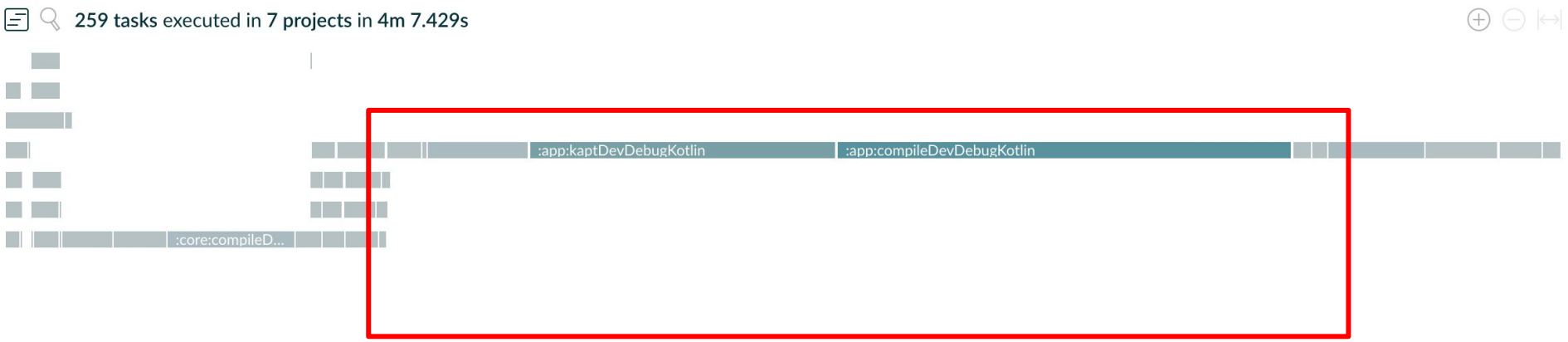
org.gradle.parallel=true

parallel features build

Parallel original build (clean)

Google Developers Experts

org.gradle.parallel=true



fat app module looooooong running build


STAR WARS
EPISODE IV
A NEW HOPE

Four ROOT PROBLEMS:

Annotation processing everywhere

Wrong module dependency

Application module DataBinding and Dagger generation

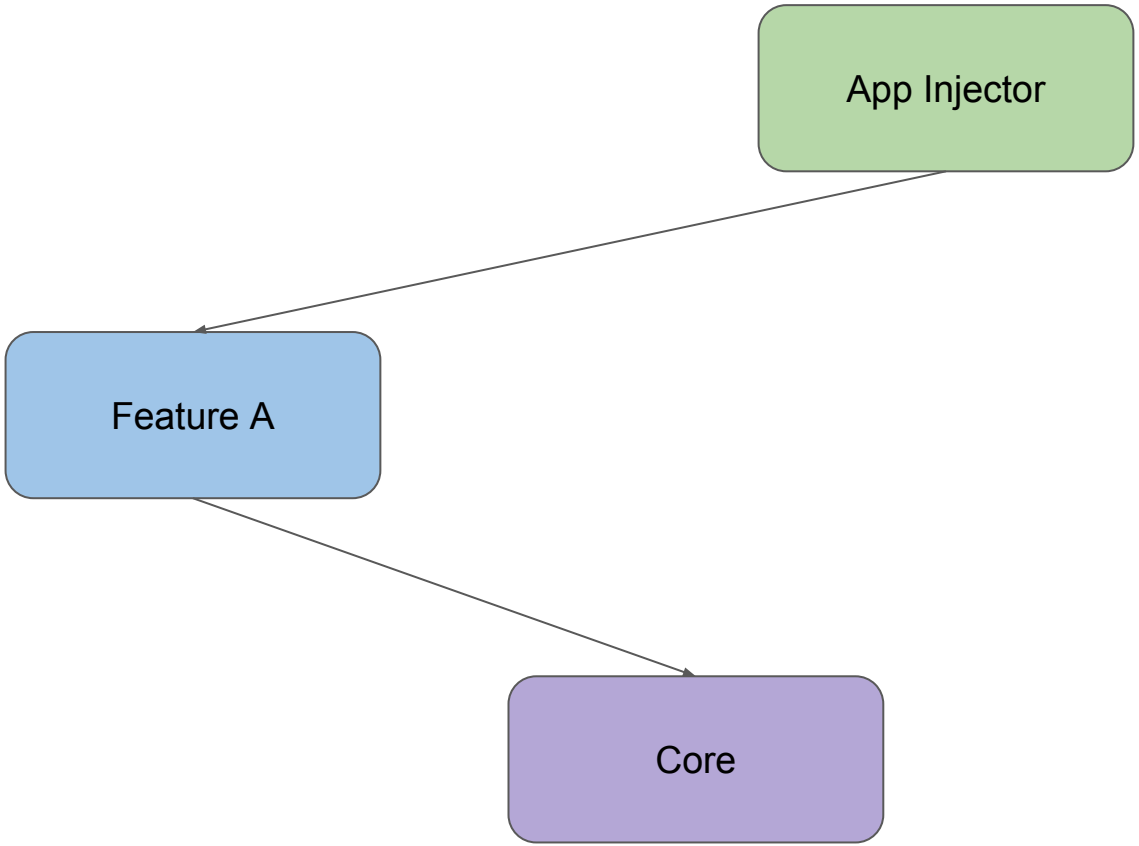
Fat core and application modules

Google Developers Experts

Solution?

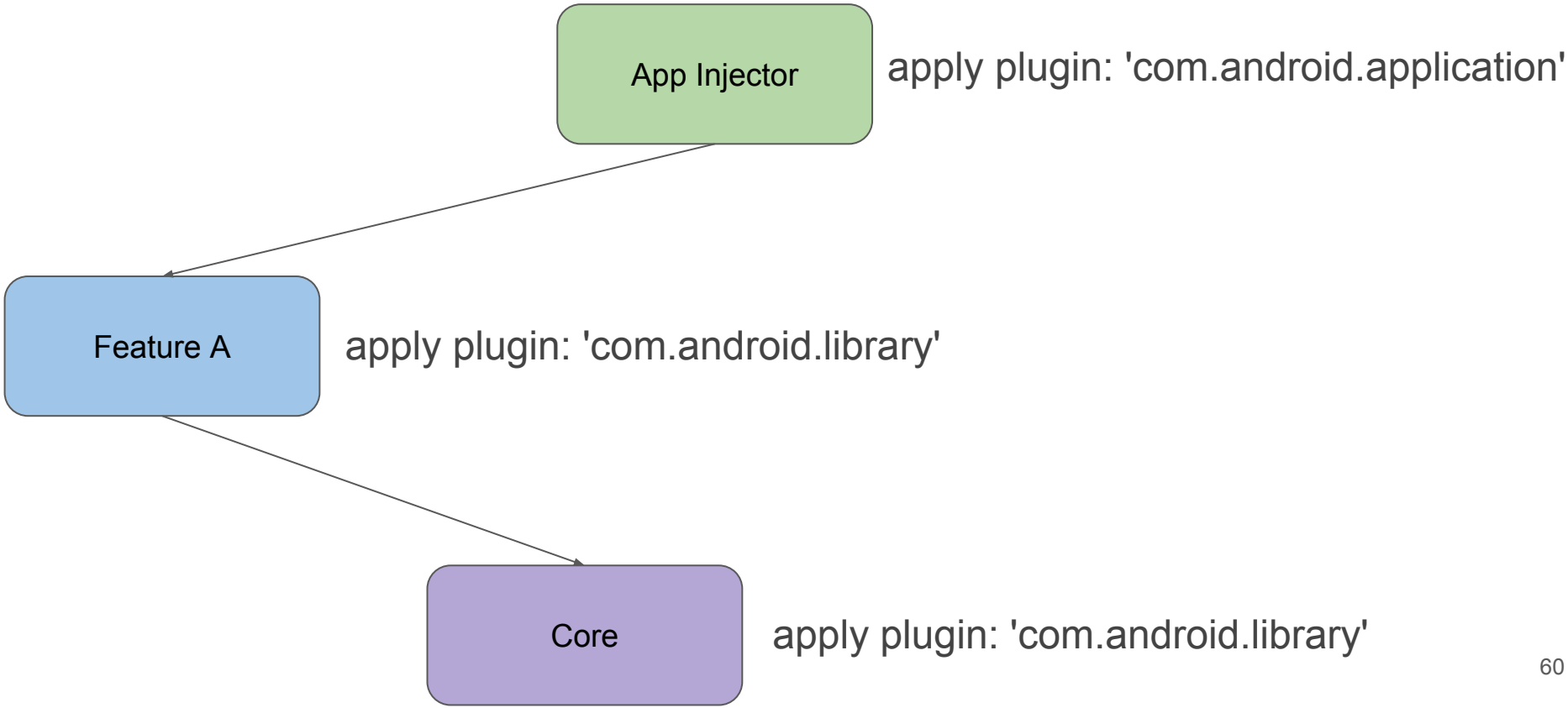
Several modules architecture:

Google Developers Experts

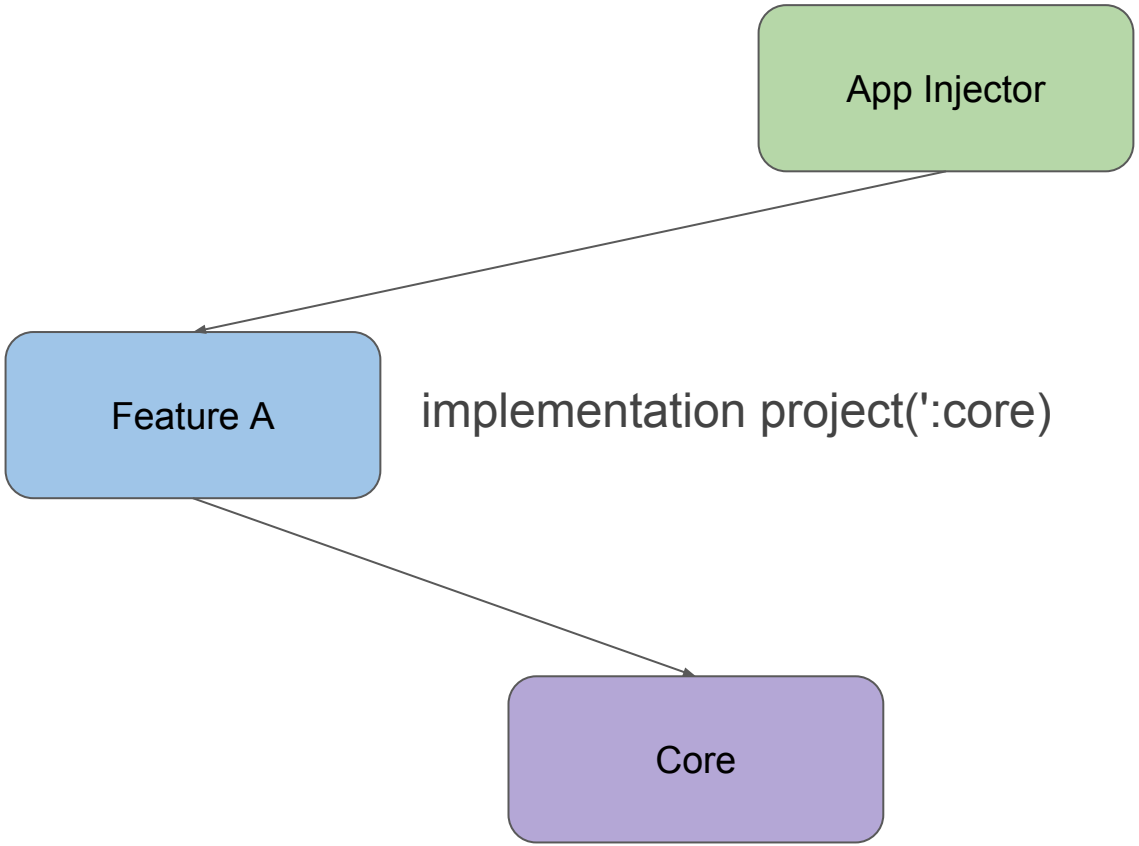


Several modules architecture:

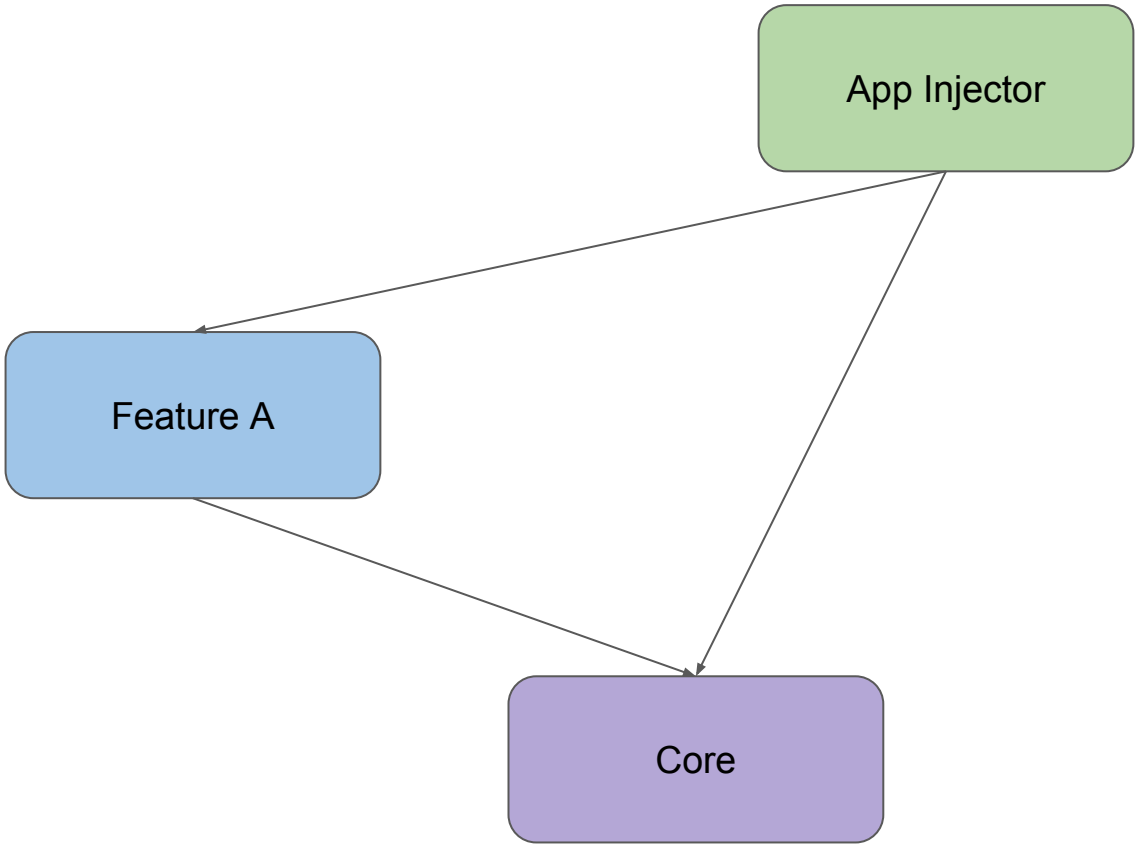
Google Developers Experts



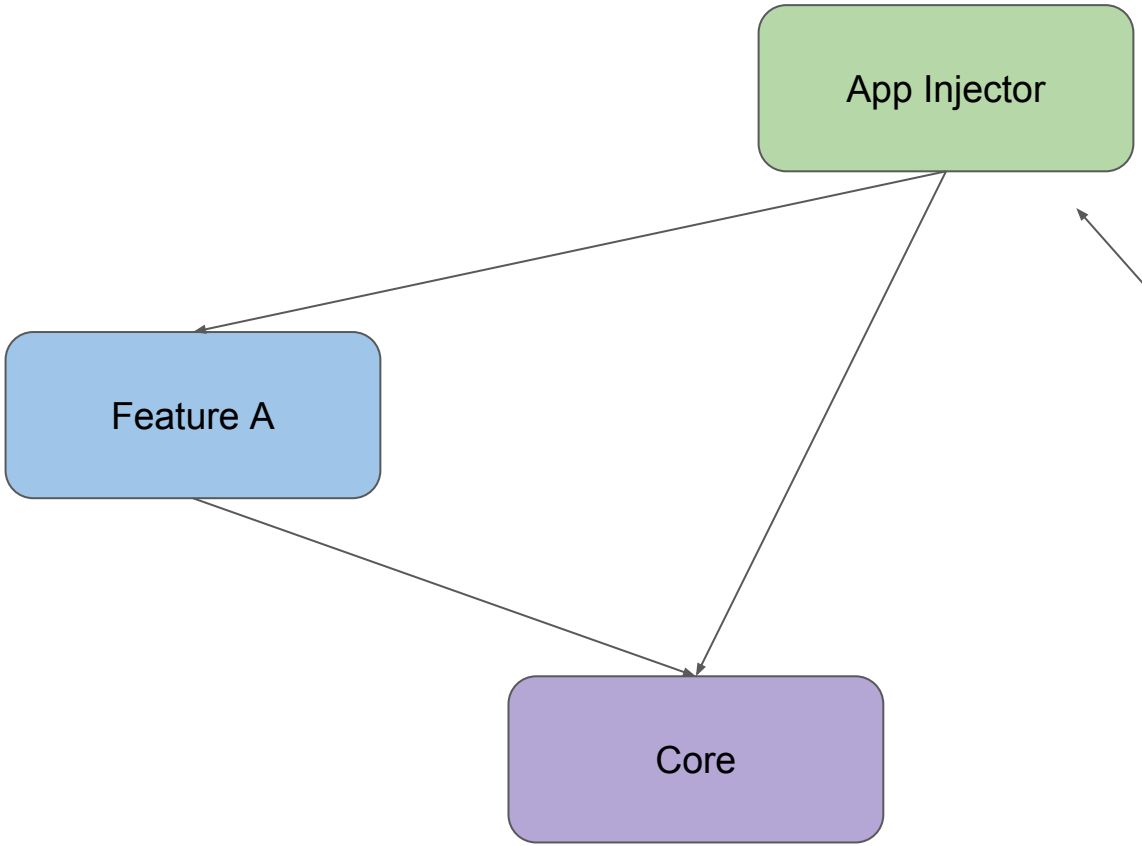
Several modules architecture:



Google Developers Experts

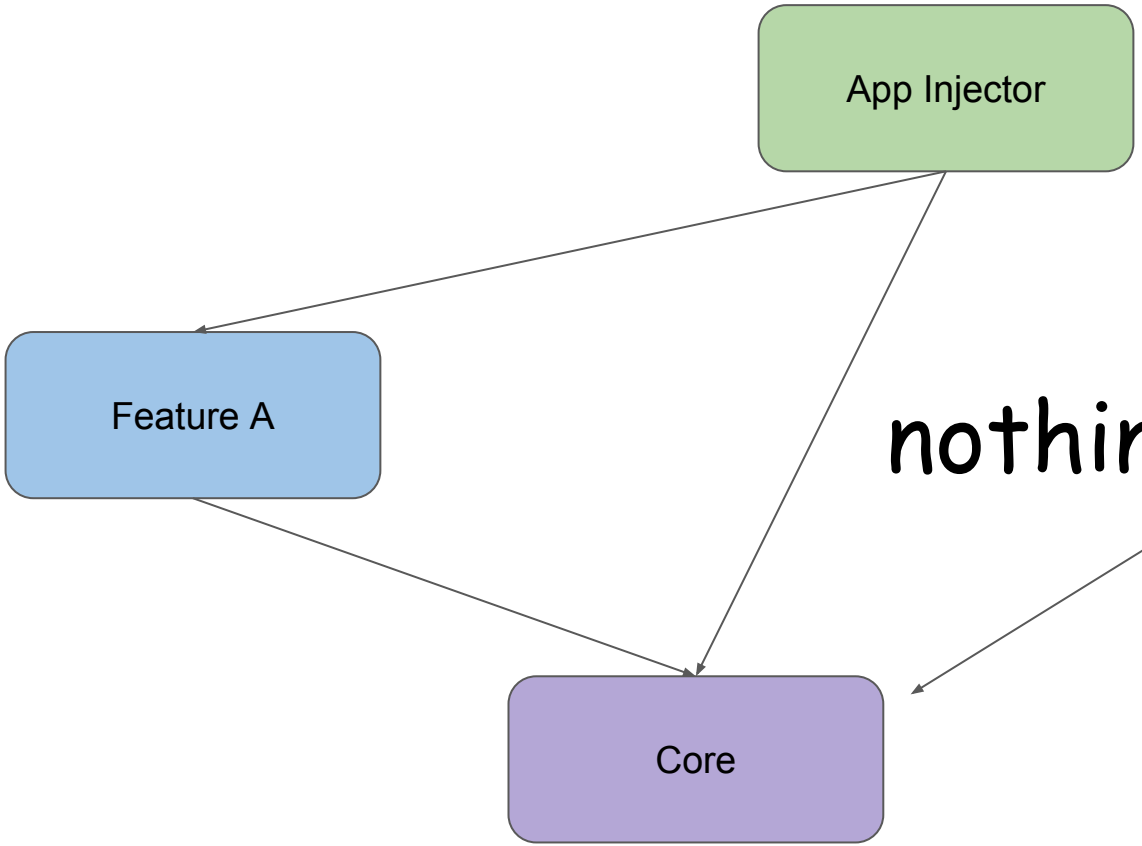


Google Developers Experts



nothing but DI

Google Developers Experts

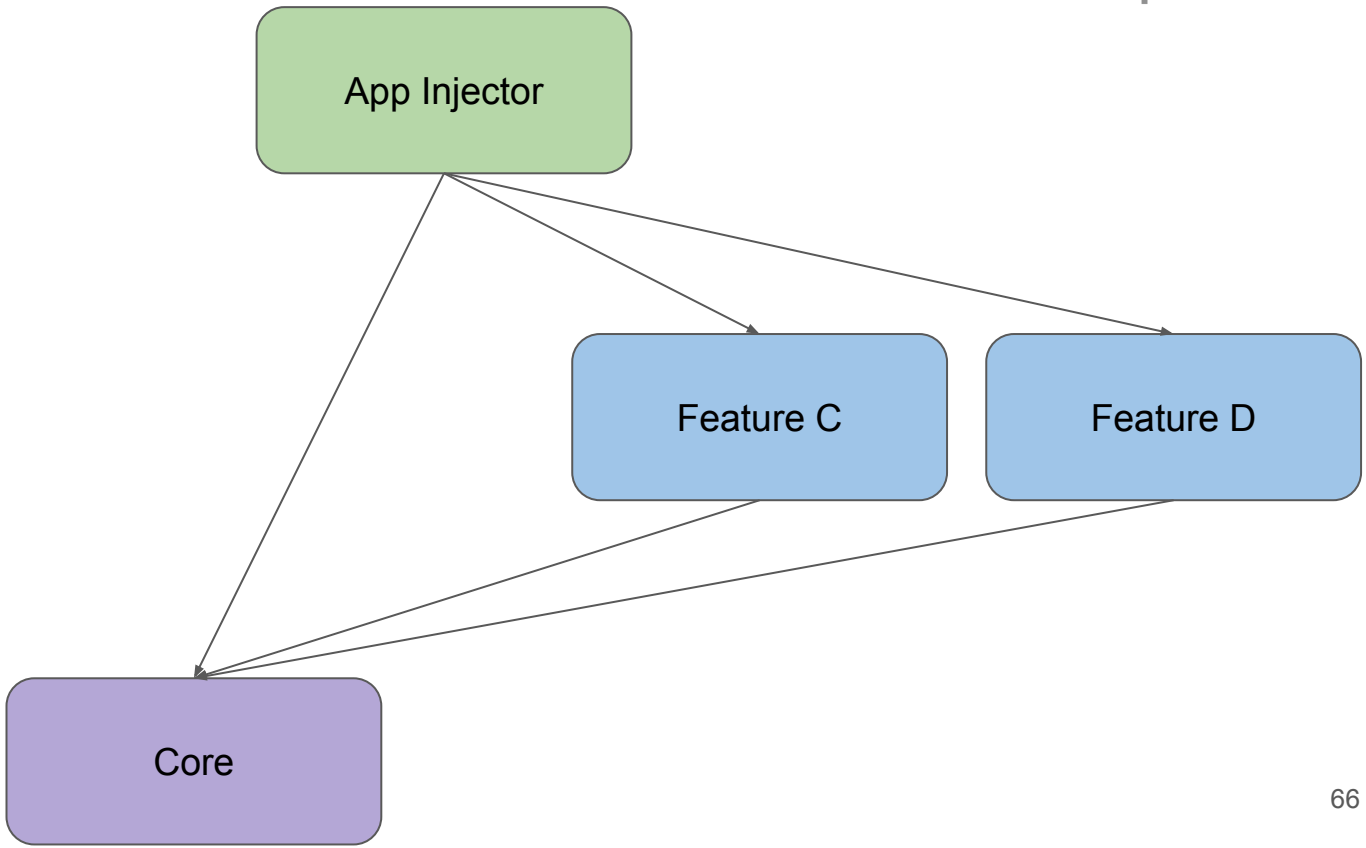


nothing but interfaces and POJO

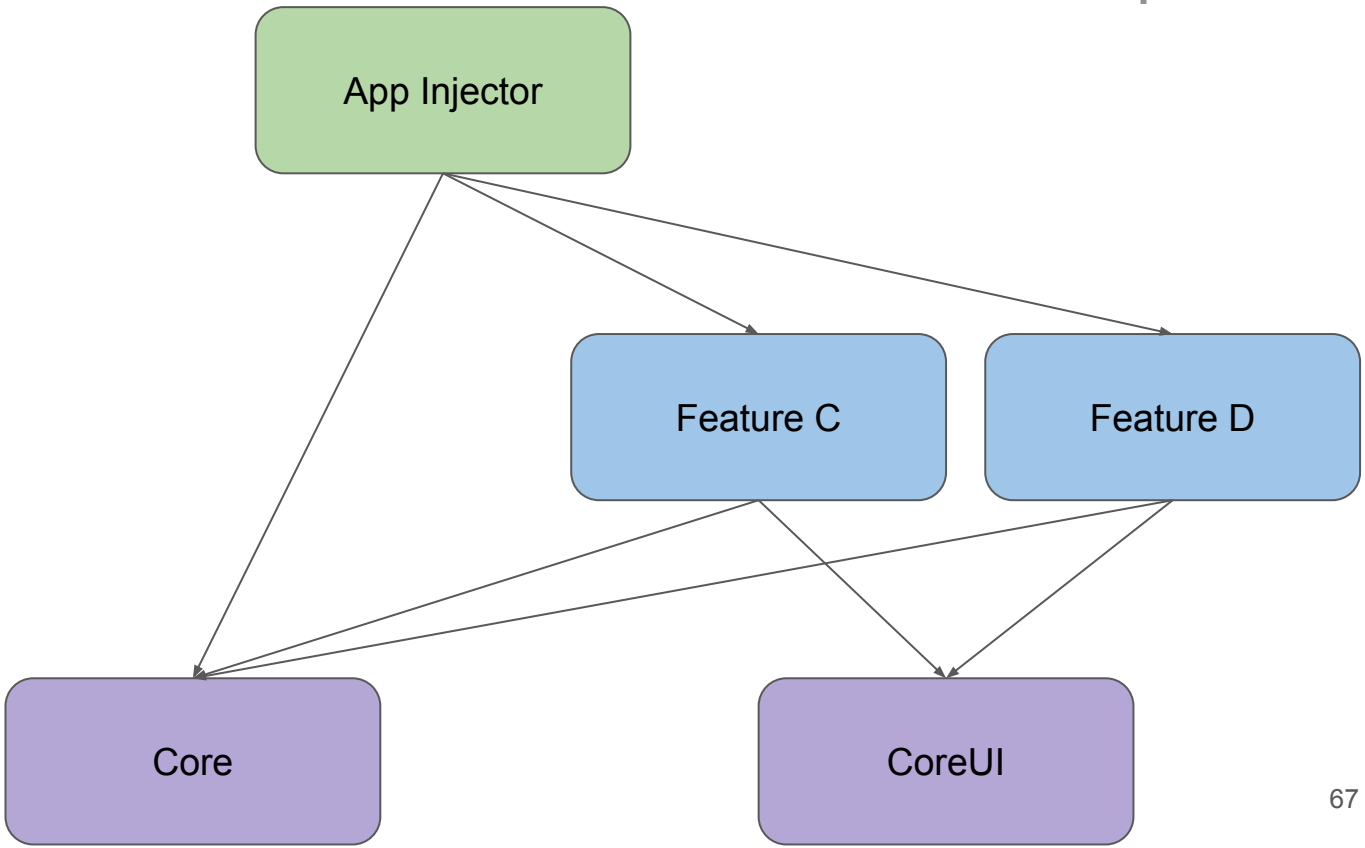
Google Developers Experts

Where is common UI logic?

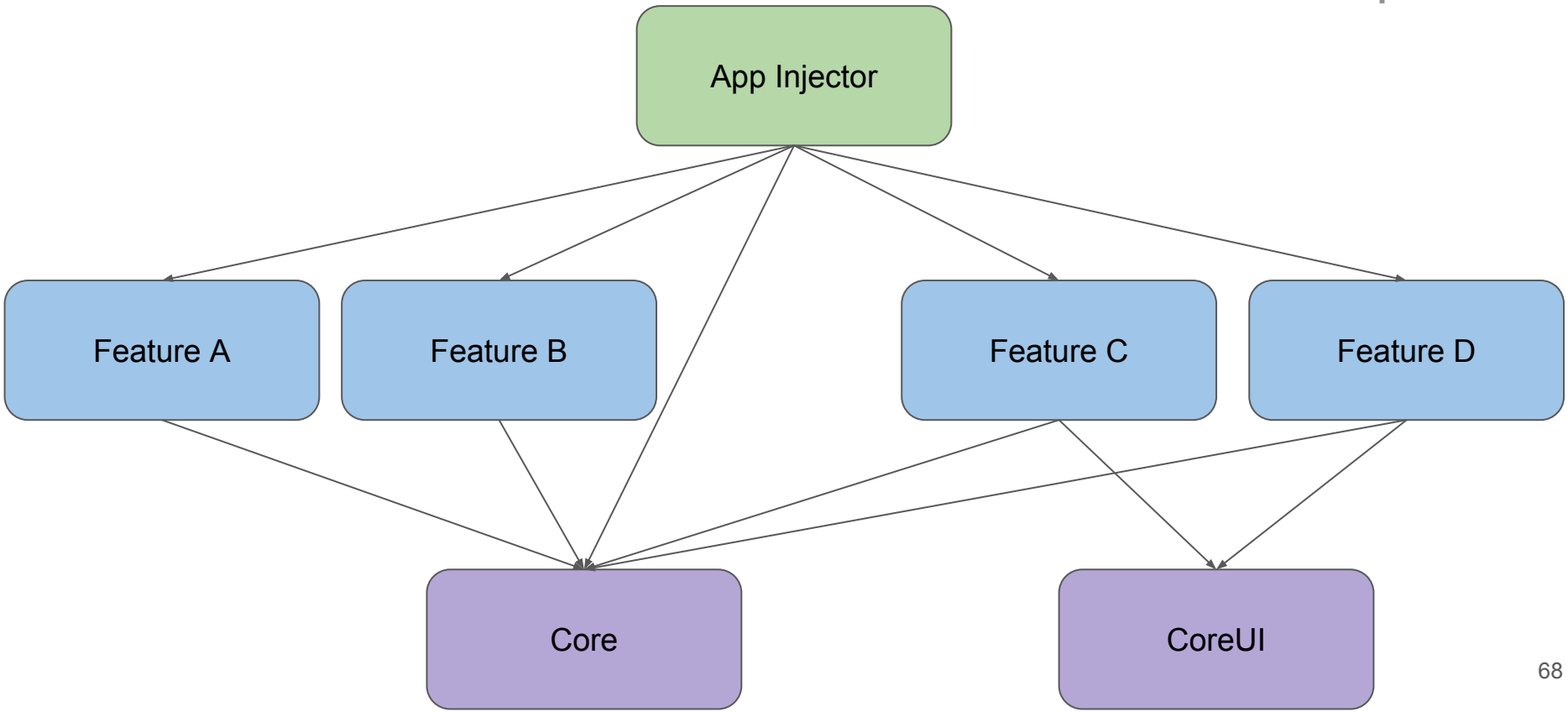
Google Developers Experts



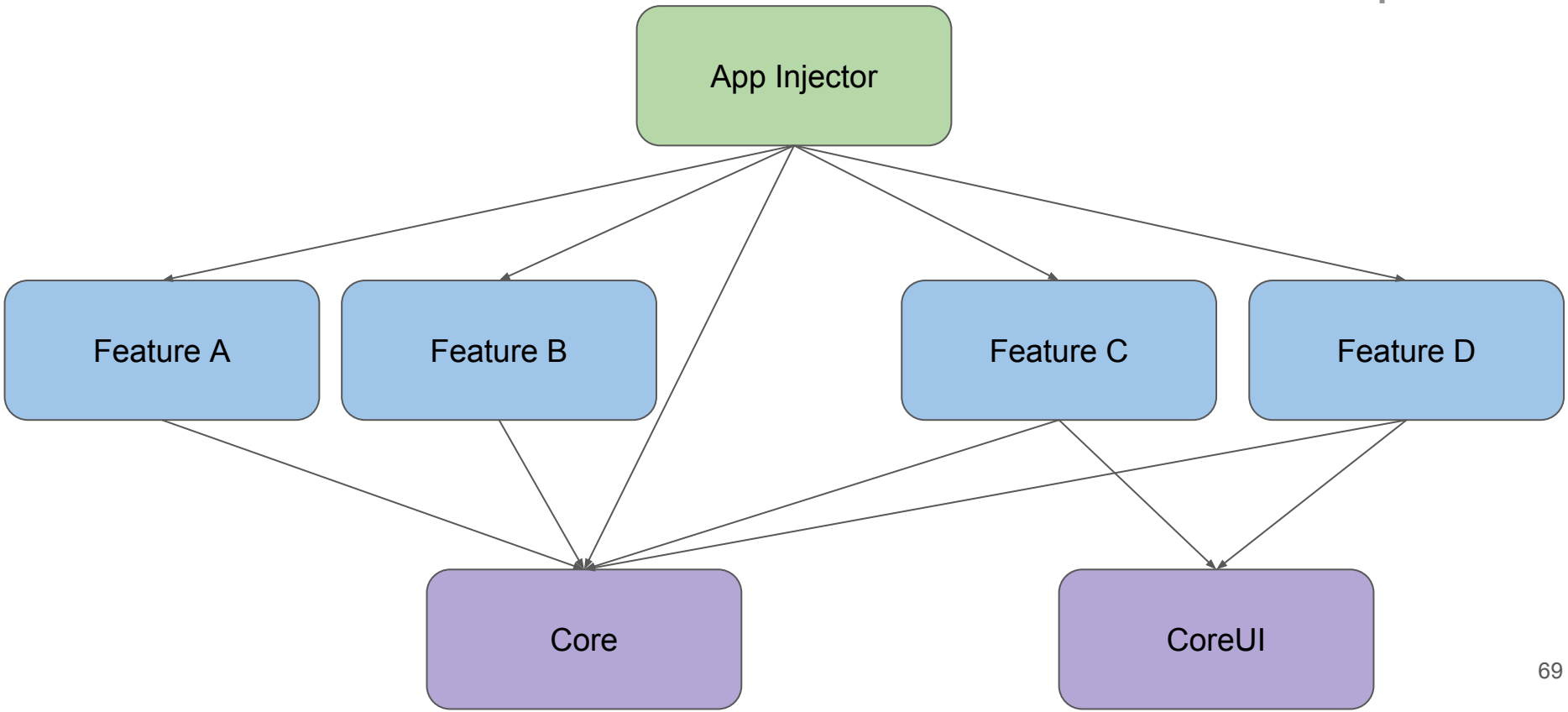
Google Developers Experts



Google Developers Experts

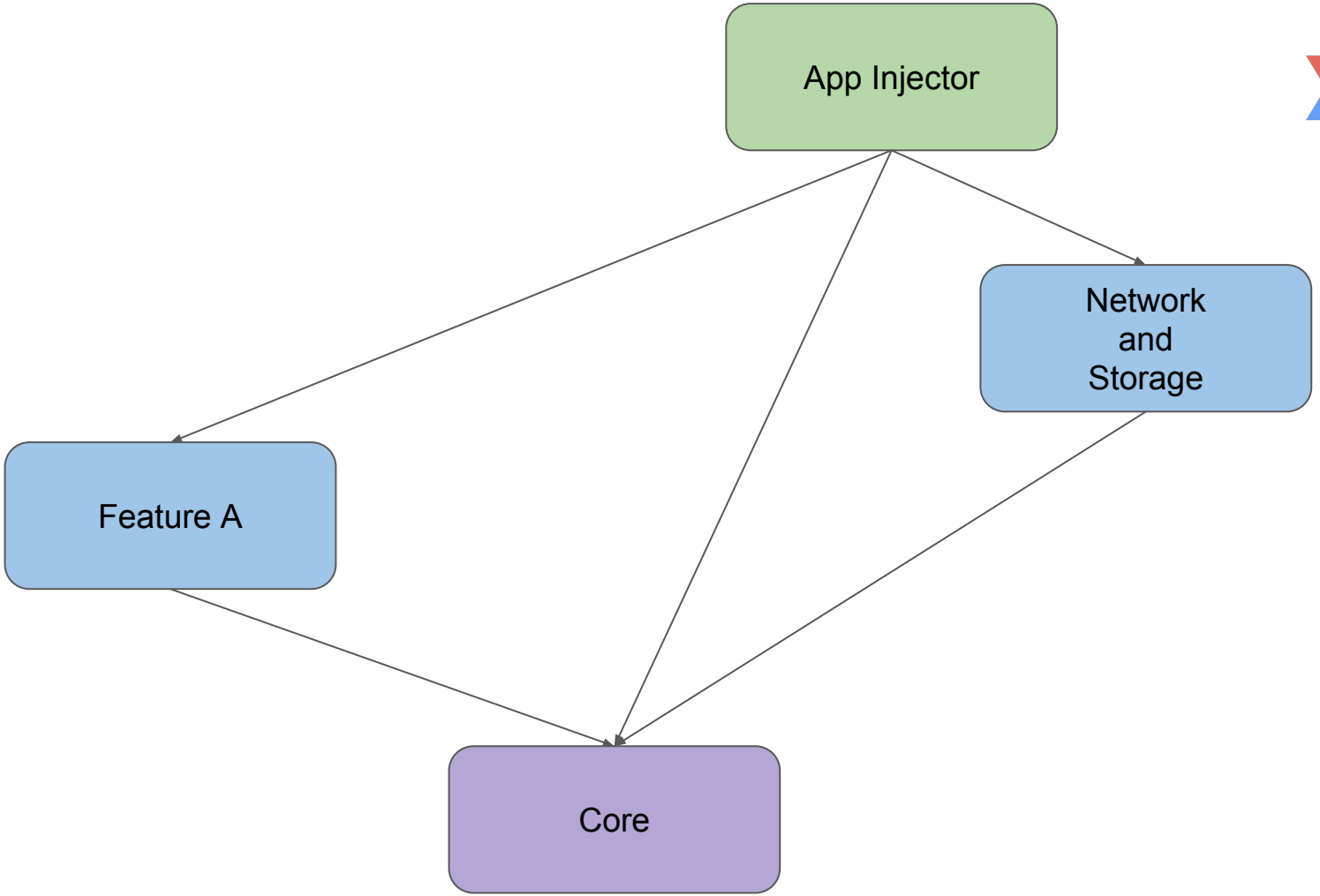


Google Developers Experts

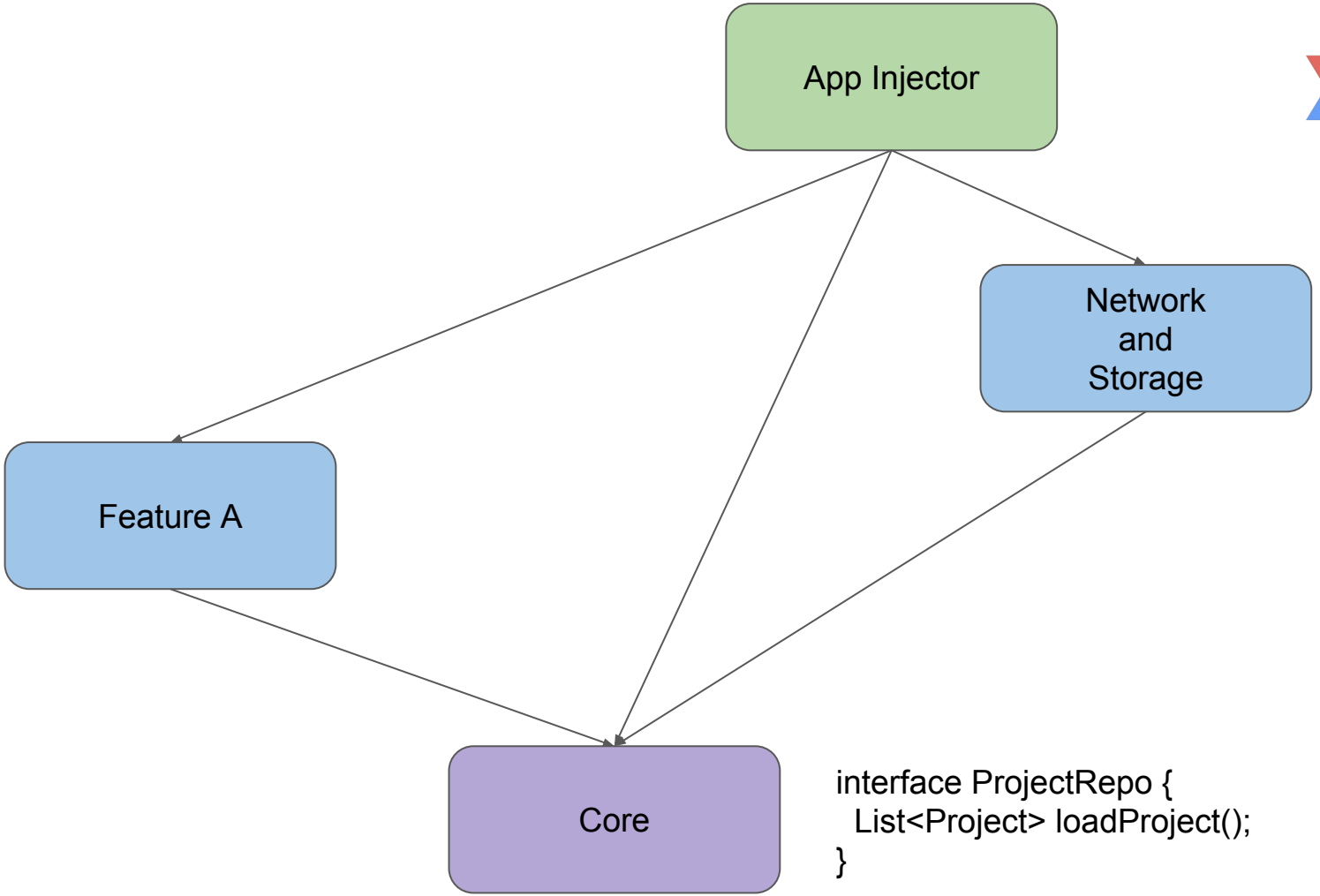


Google Developers Experts

Where is common network and storage?

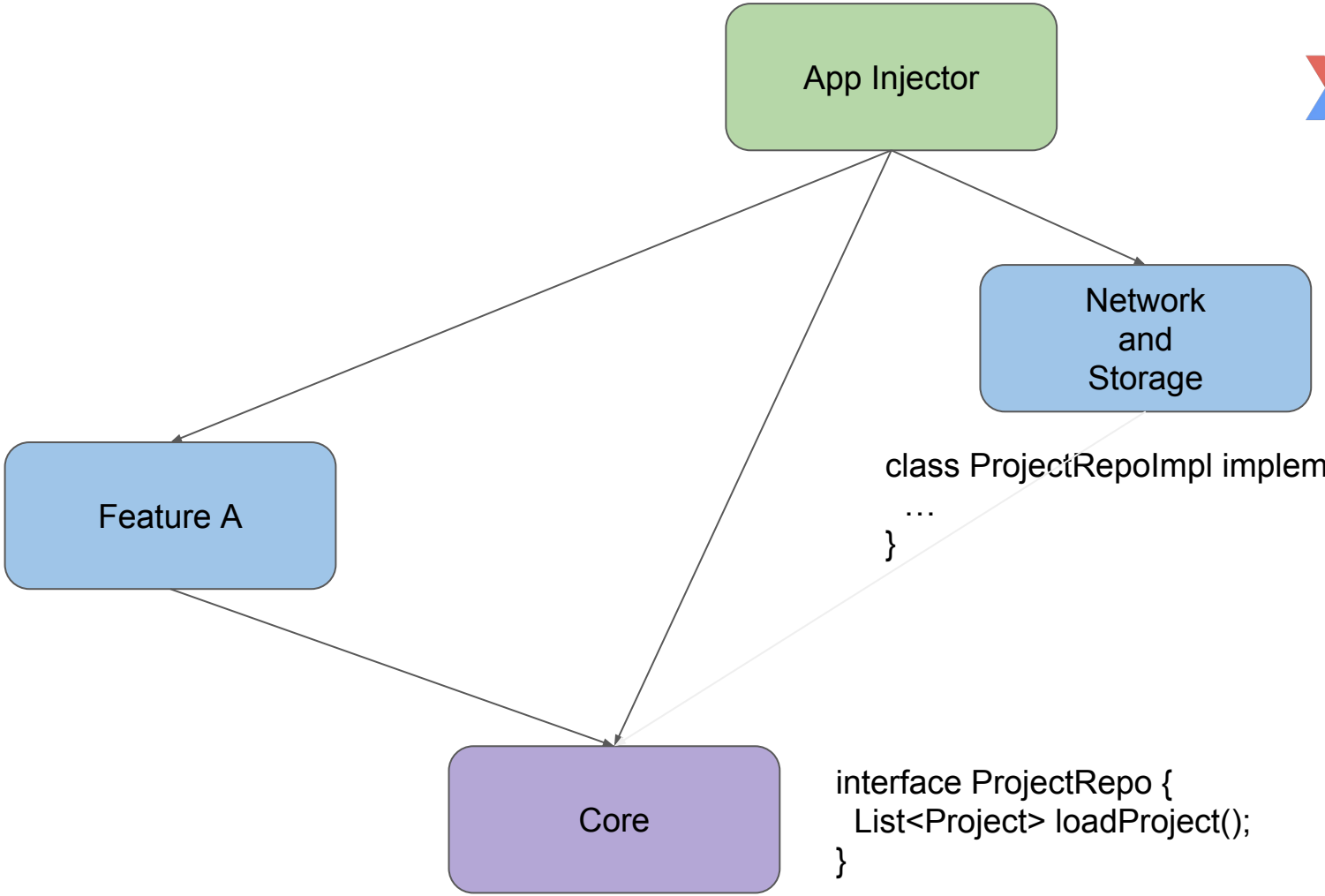


Google Developers Experts



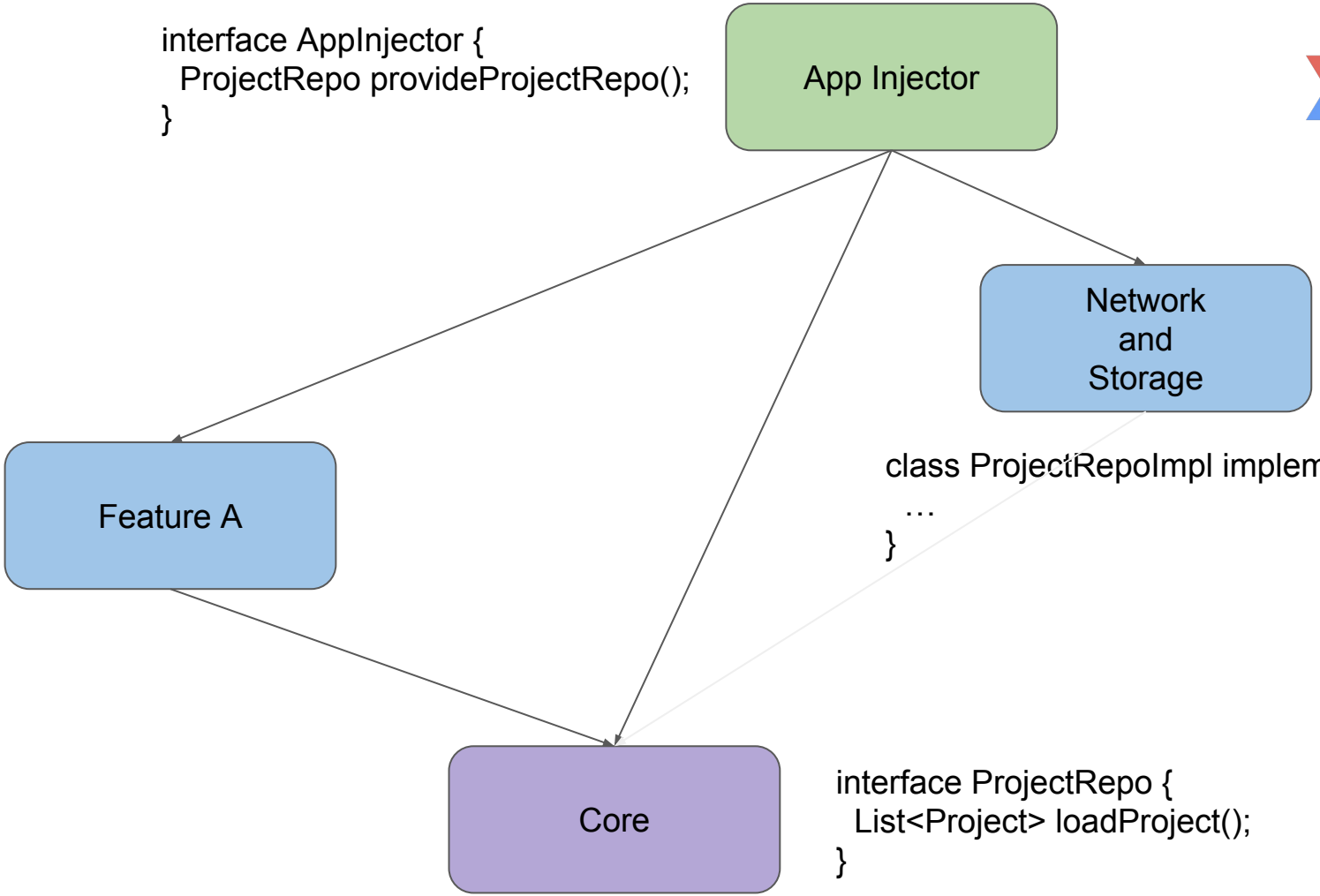
Google Developers Experts

```
interface ProjectRepo {
  List<Project> loadProject();
}
```



Google Developers Experts

```
interface AppInjector {
  ProjectRepo provideProjectRepo();
}
```

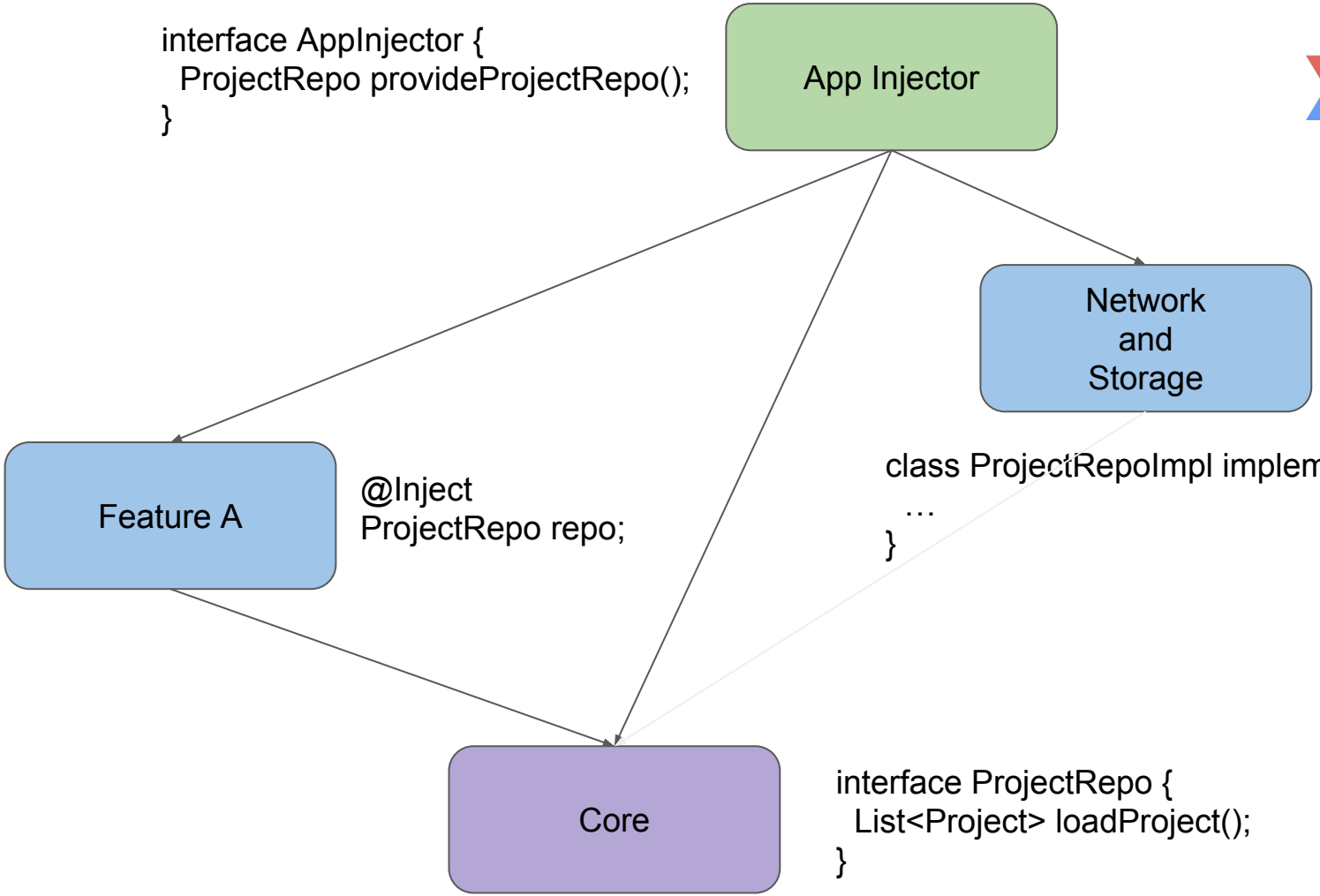



```
class ProjectRepoImpl implements ProjectRepo {
  …
}
```

```
interface ProjectRepo {
  List<Project> loadProject();
}
```

Google Developers Experts

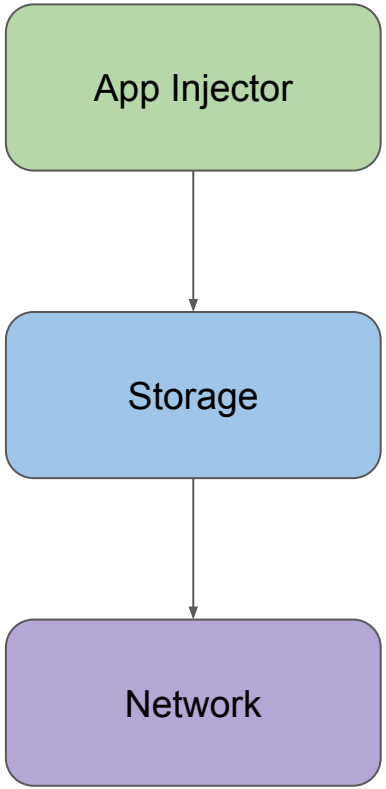
```
interface AppInjector {
  ProjectRepo provideProjectRepo();
}
```



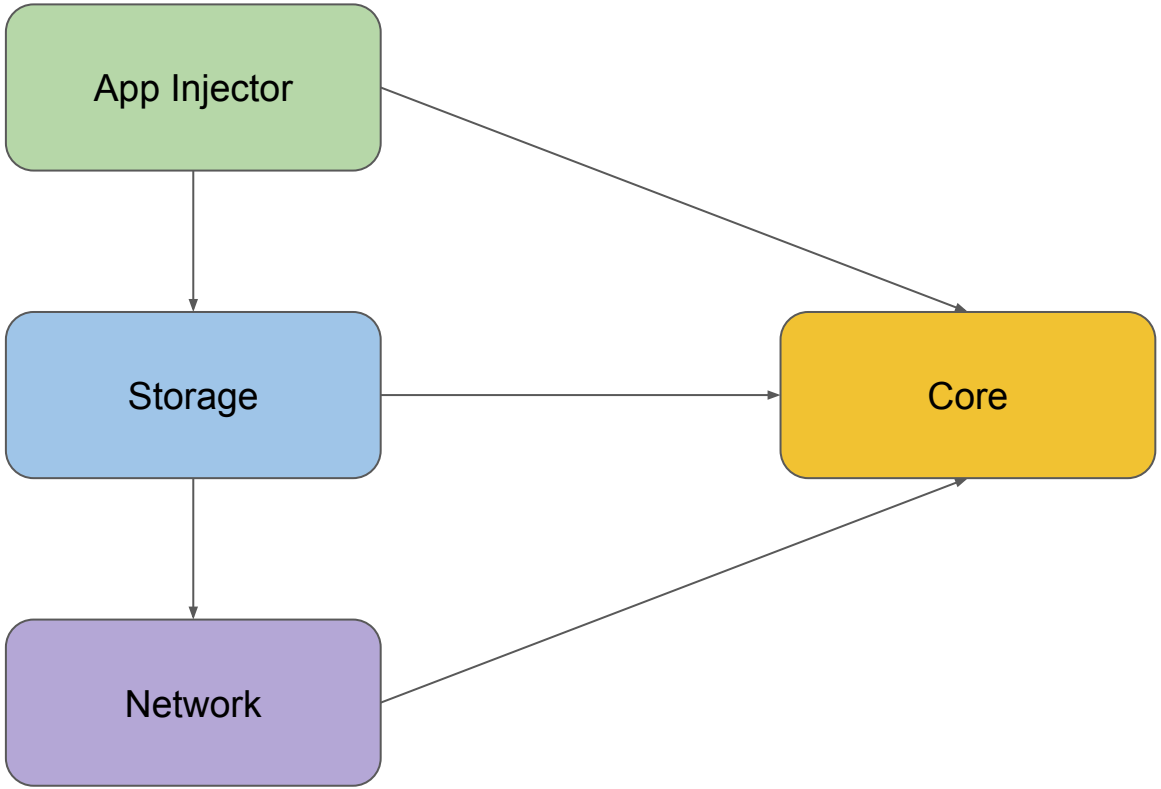
Google Developers Experts

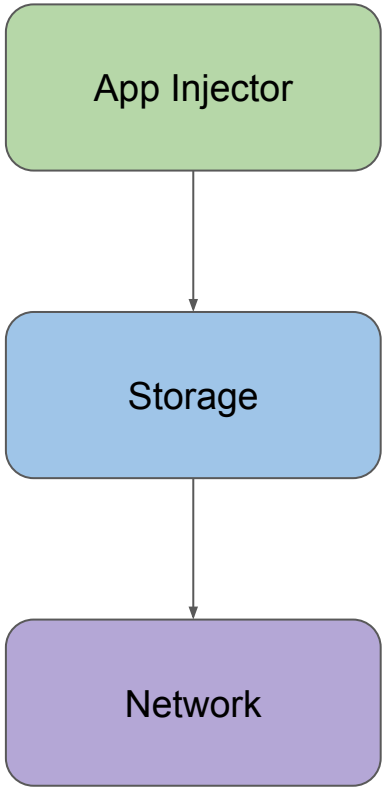
Google Developers Experts

Let’s improve data layer


App Injector
Storage
Network

Google Developers Experts





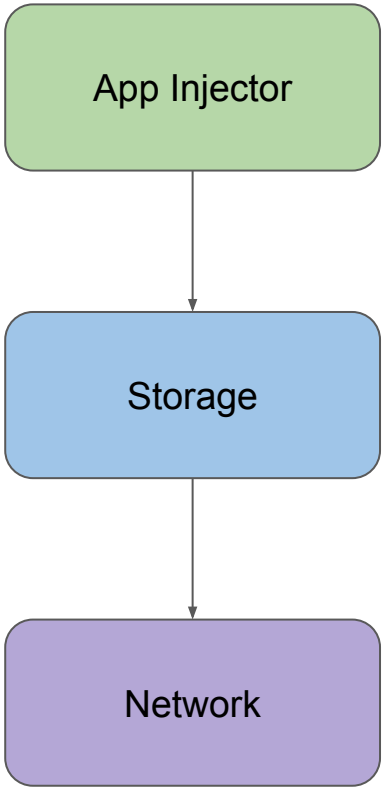
Google Developers Experts

Google Developers Experts

Gradle *implements* vs *api*

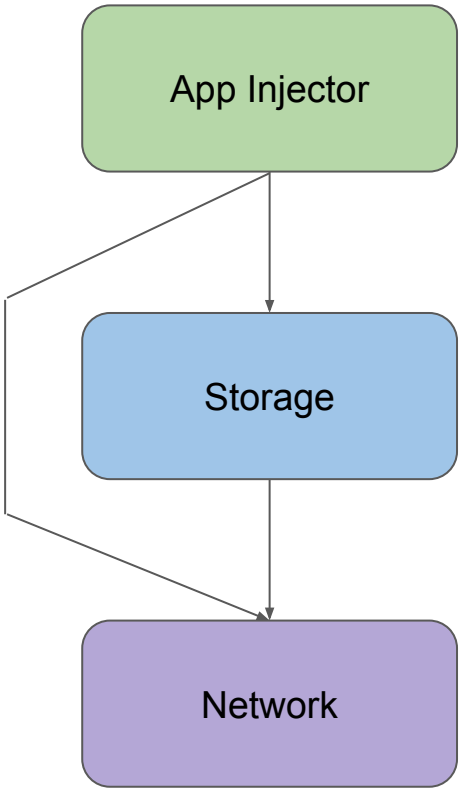
Google Developers Experts

Gradle < 4	Gradle 4+
compile	implements, api



api project(':storage')

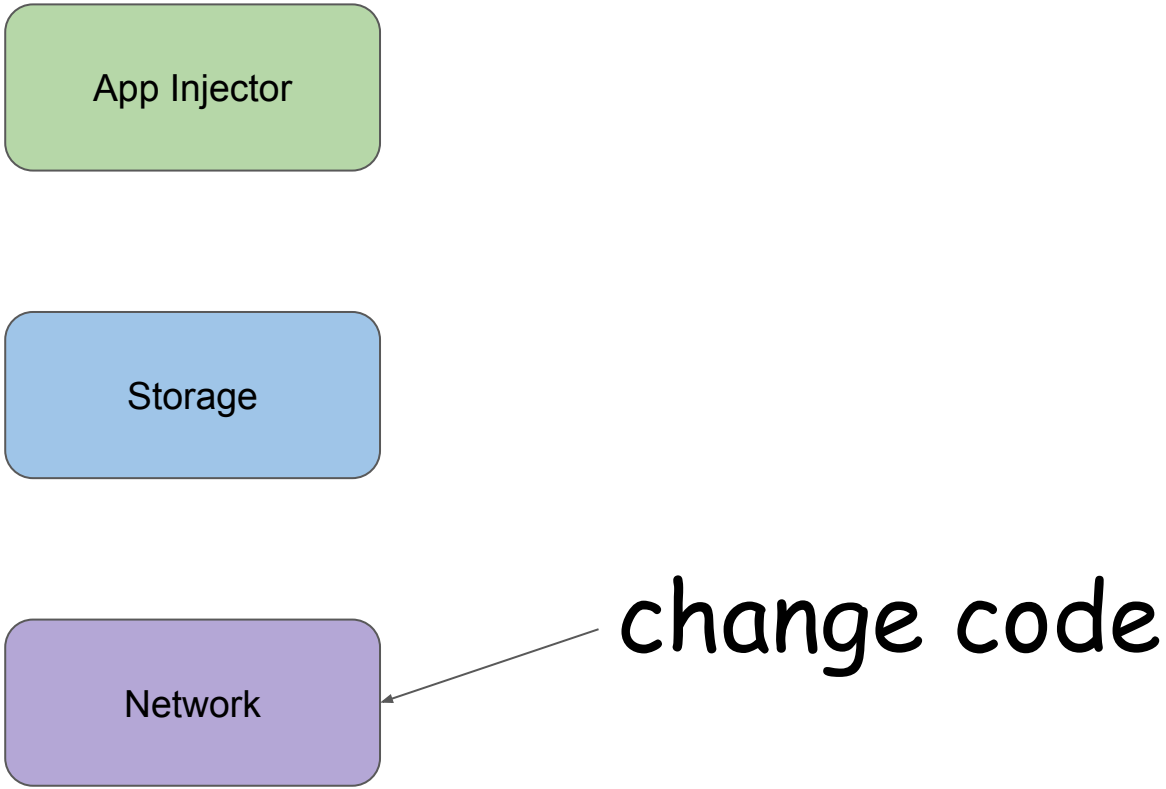
api project(':network)



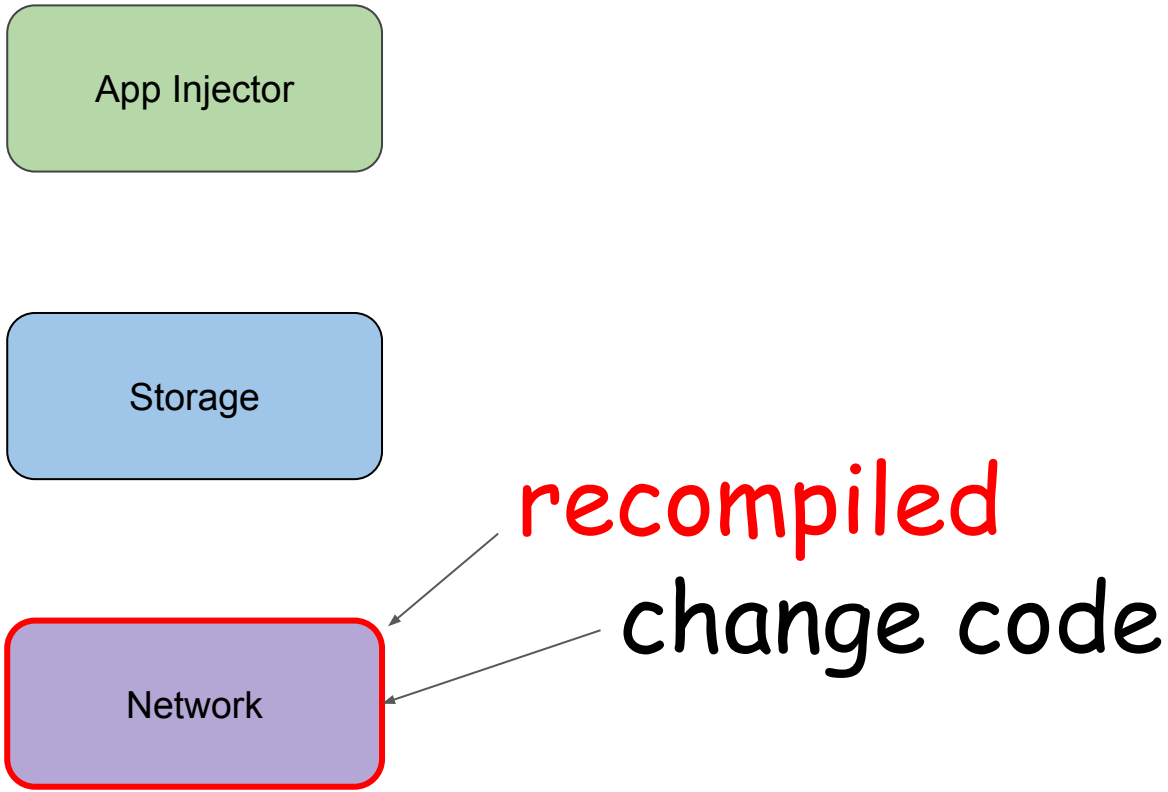
api project(':storage')

api project(':network)

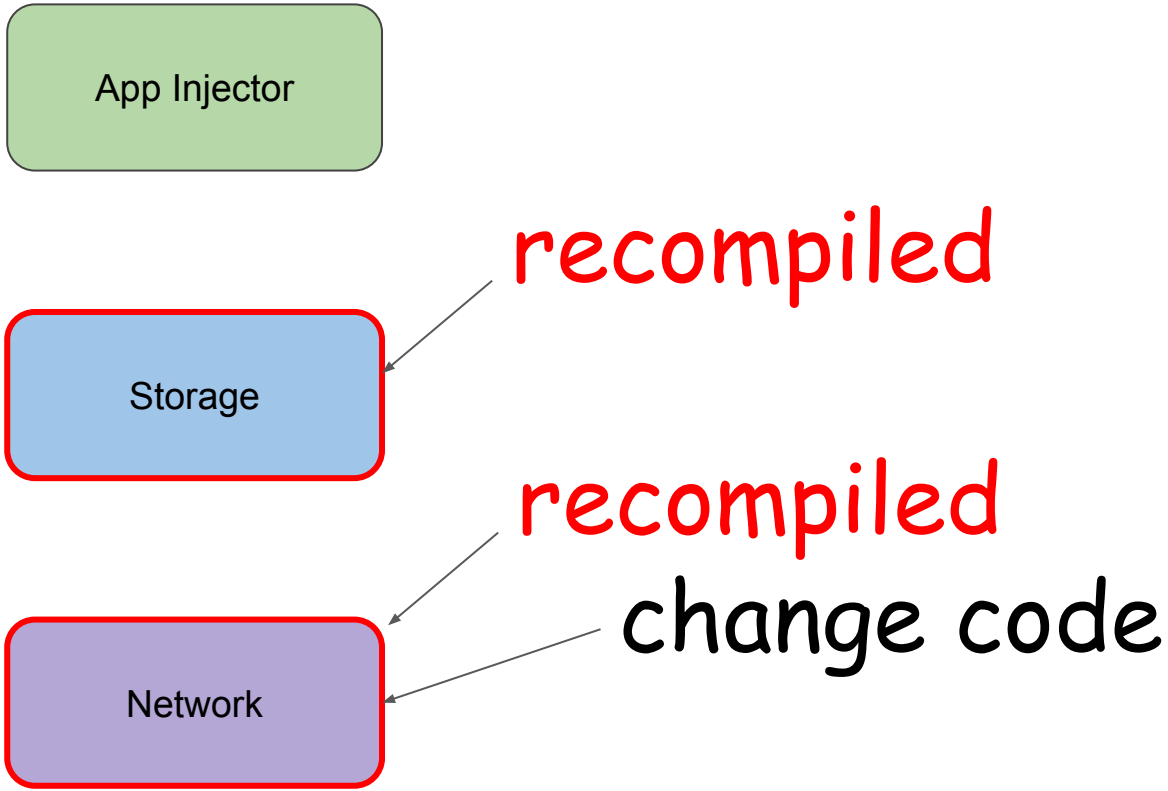
Google Developers Experts

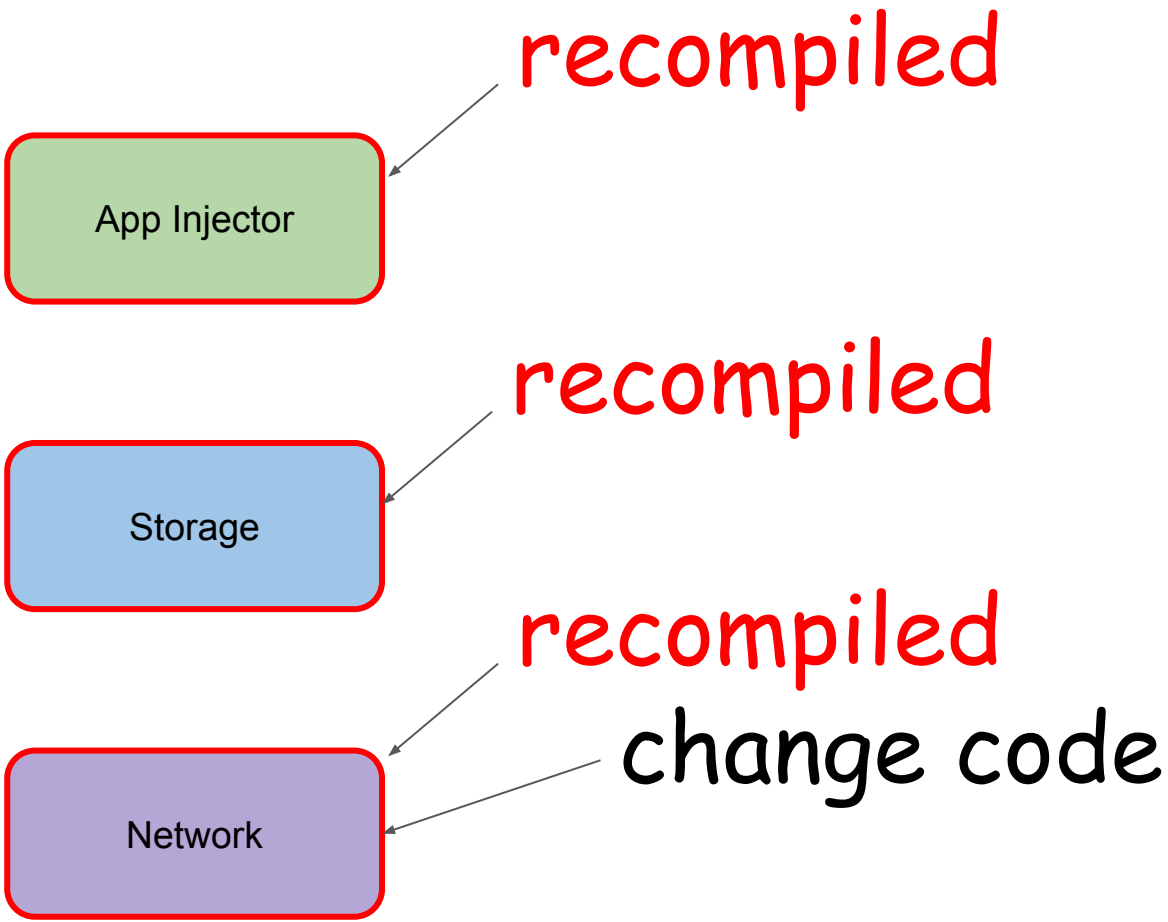


Google Developers Experts

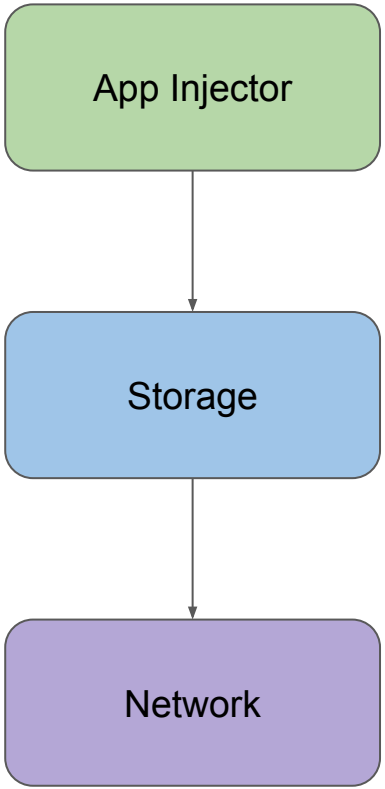


Google Developers Experts





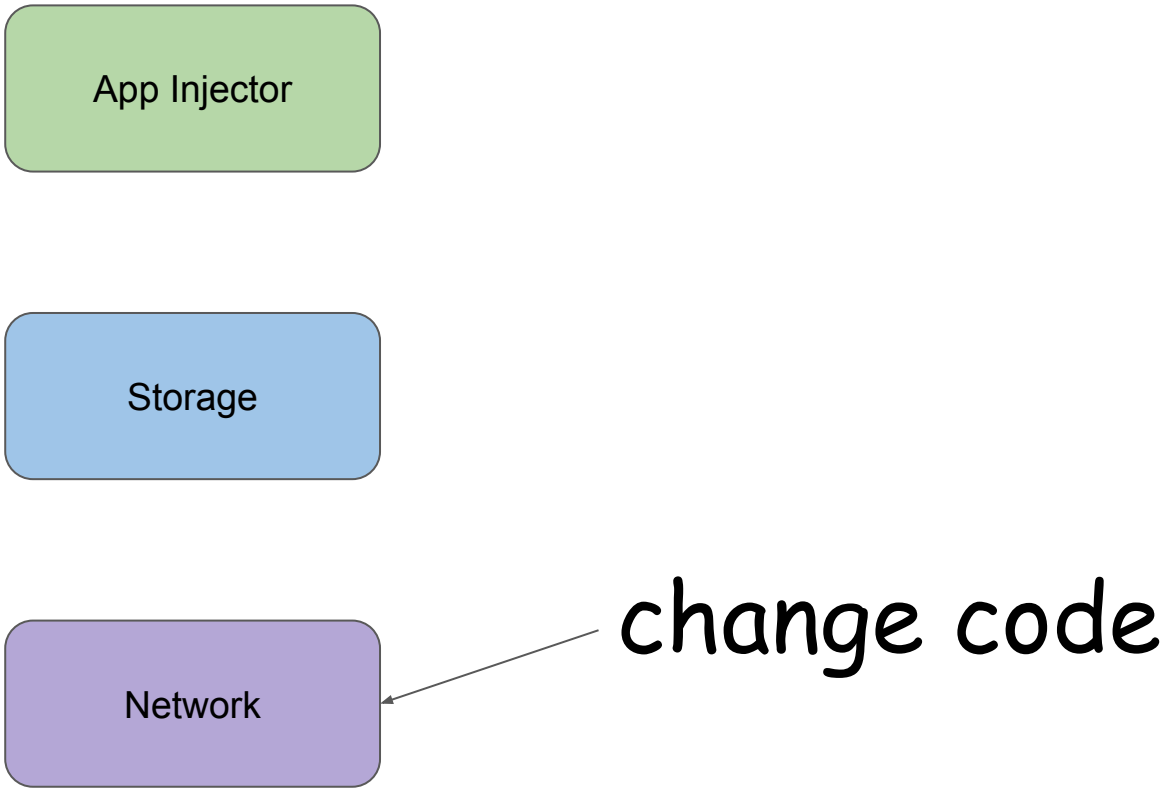
Google Developers Experts



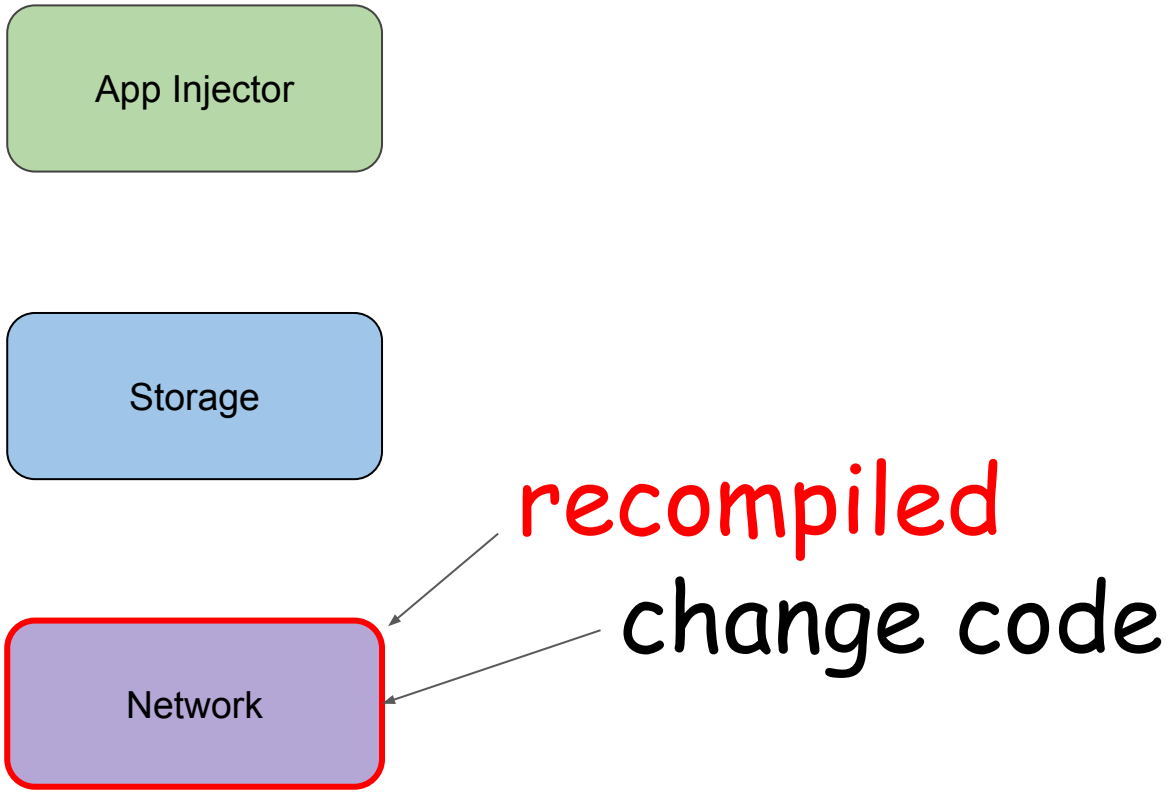
implementation project(':storage')

implementation project(':network)

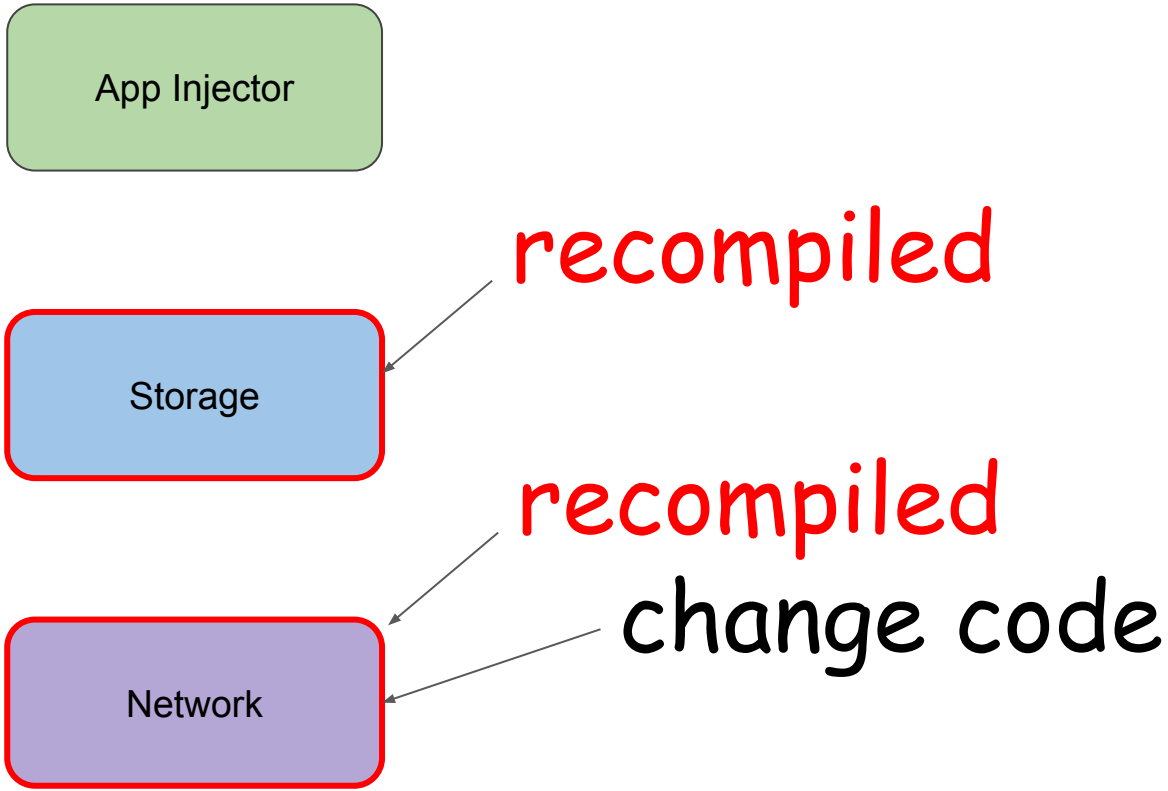
Google Developers Experts

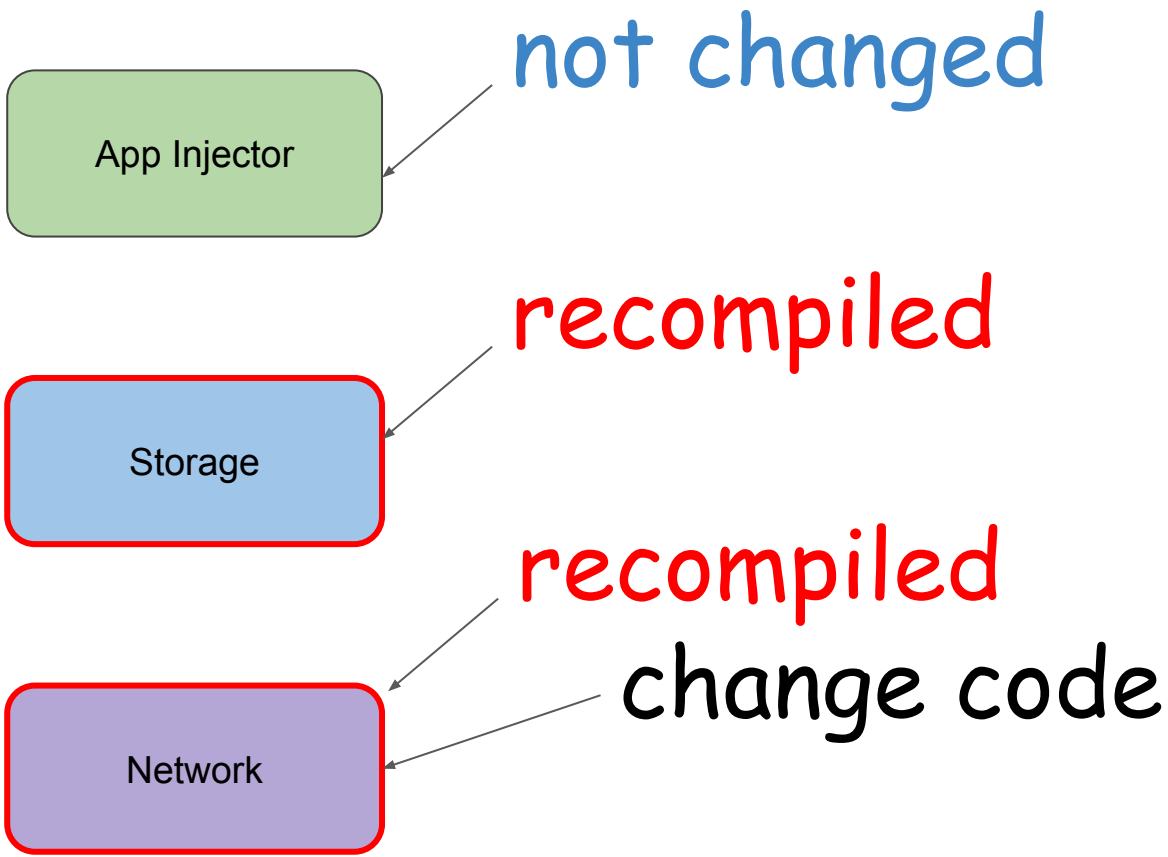


Google Developers Experts



Google Developers Experts

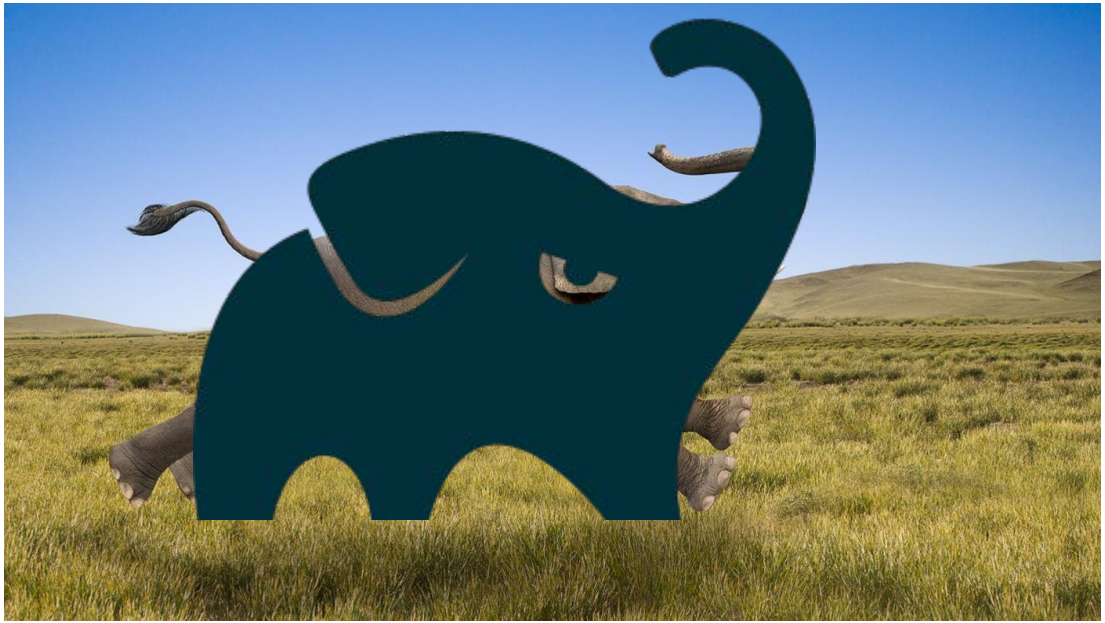




Google Developers Experts

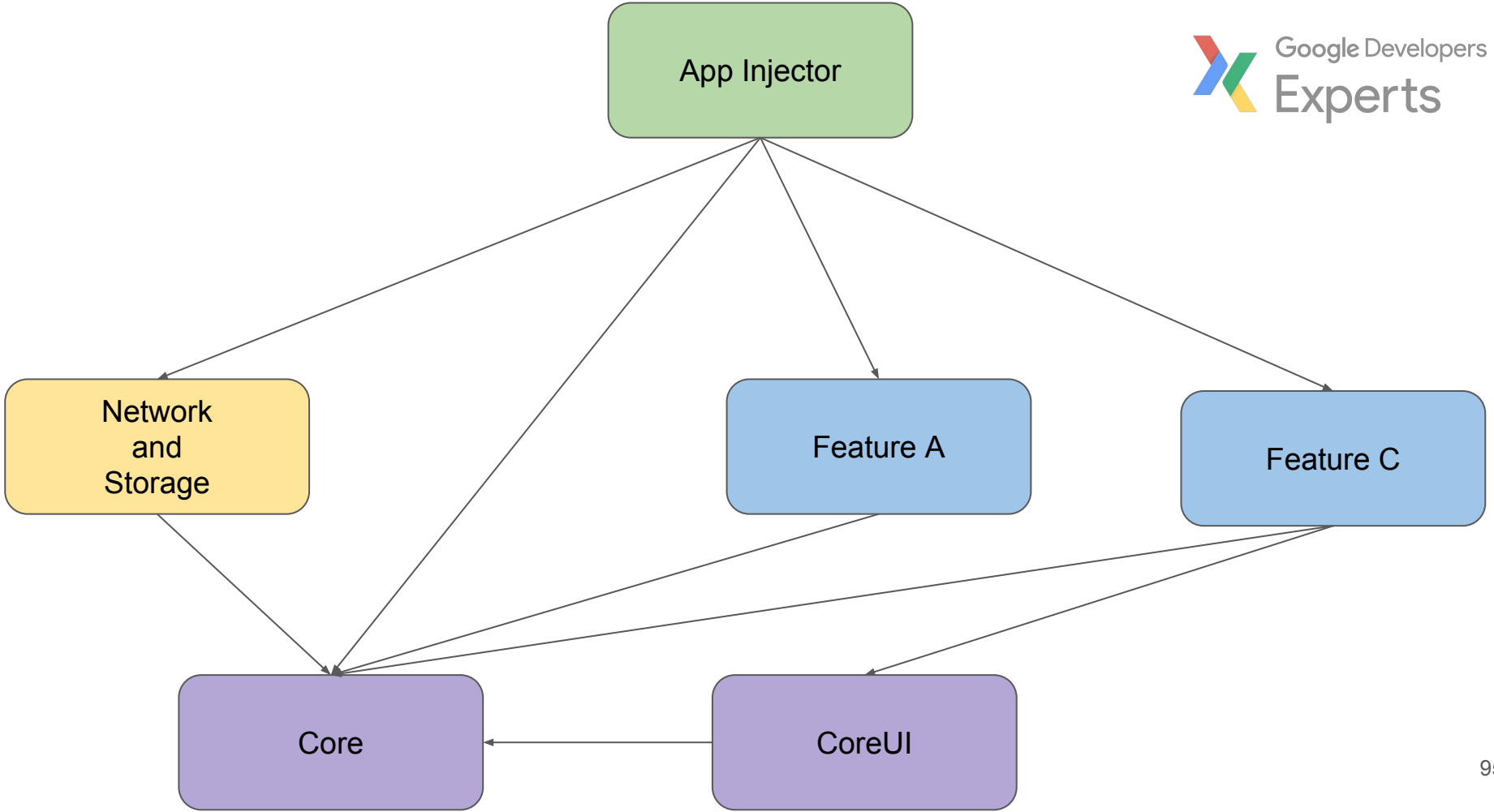
Google Developers Experts

One line change: 30 sec

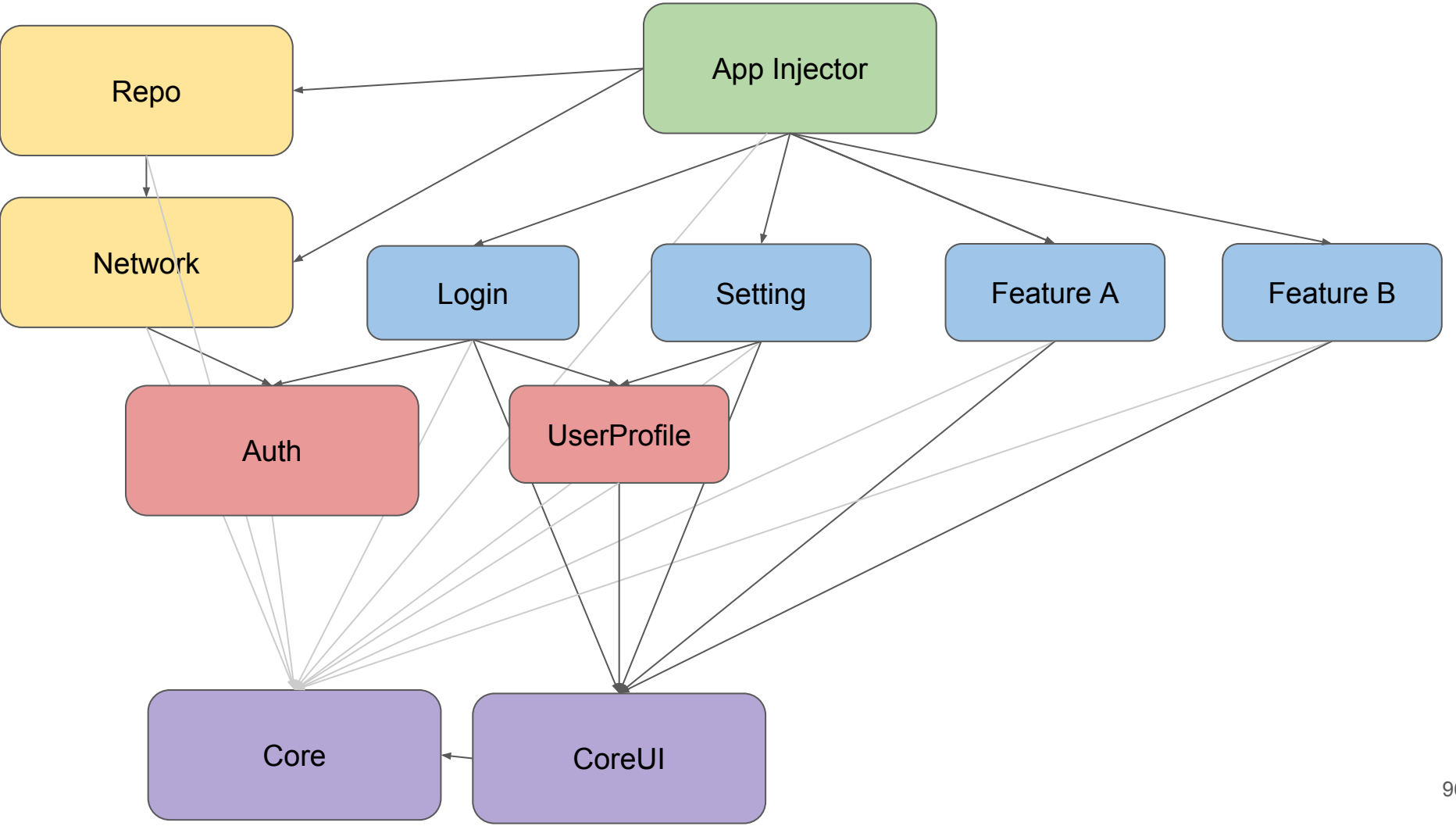


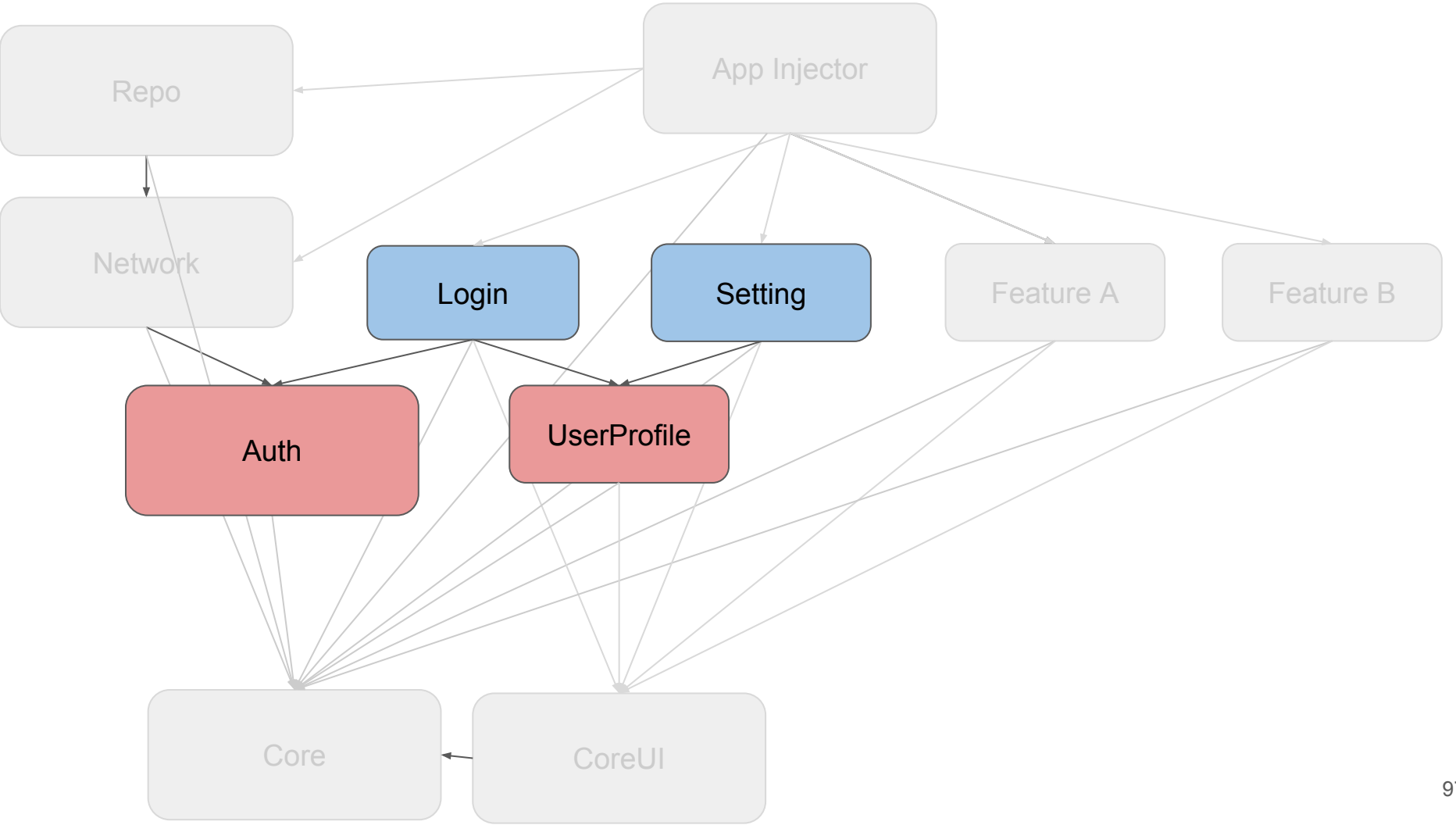
Google Developers Experts

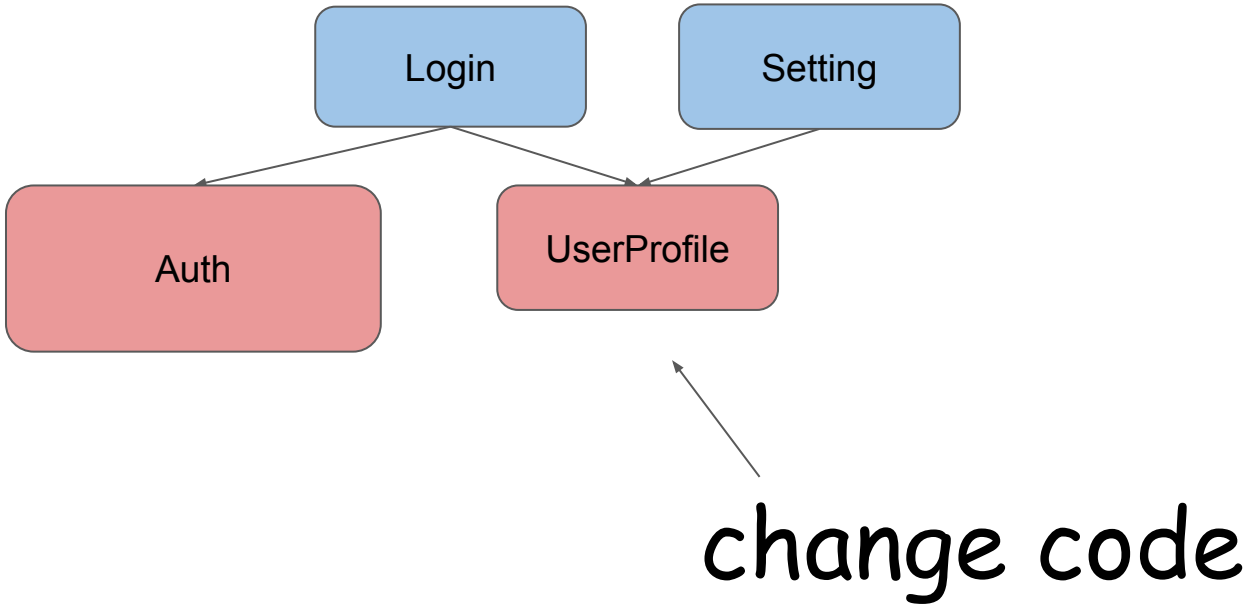
One line change in Core: 2.5 min

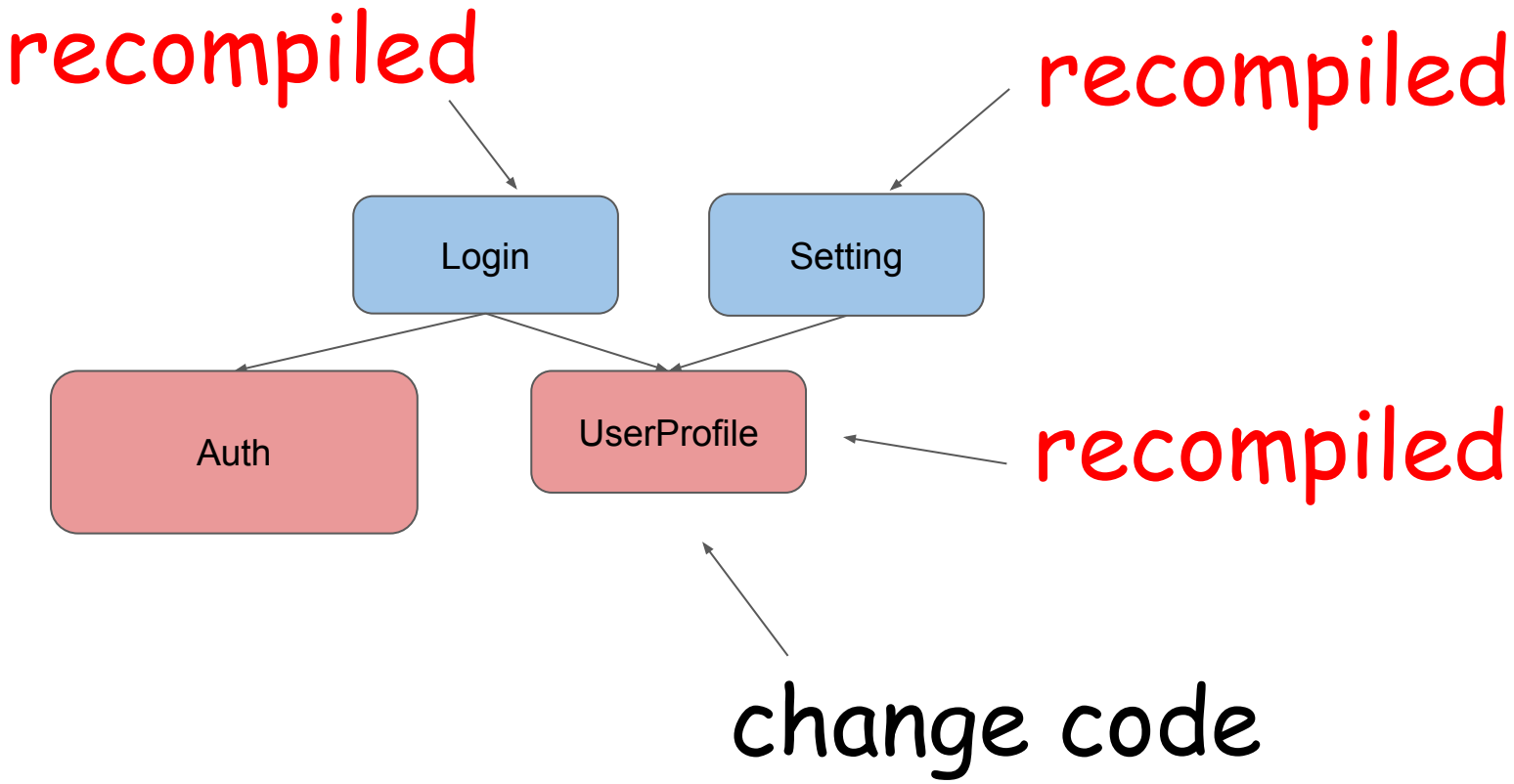


Google Developers Experts


Repo
App Injector
Network
Login
Setting
Feature A
Feature B
Auth
UserProfile
Core
CoreUI


Repo
App Injector
Network
Login
Setting
Feature A
Feature B
Auth
UserProfile
Core
CoreUI


Login
Setting
Auth
UserProfile
change code

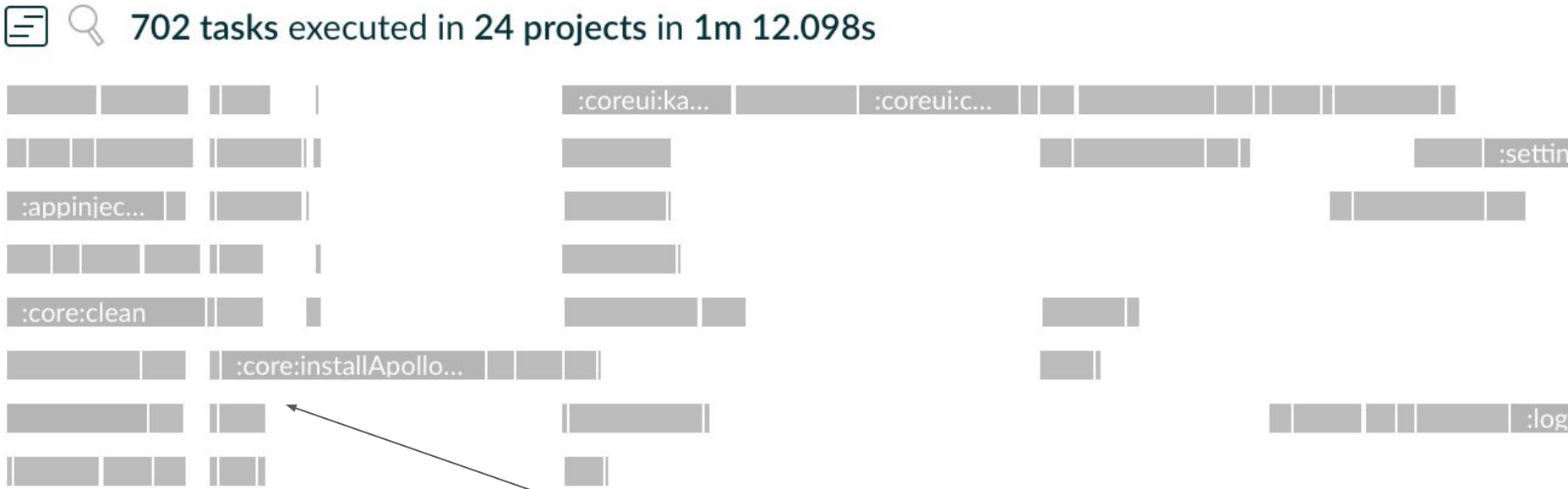

recompiled
recompiled
Login
Setting
Auth
UserProfile
recompiled
change code

Final build (clean)

Google Developers Experts

Final build

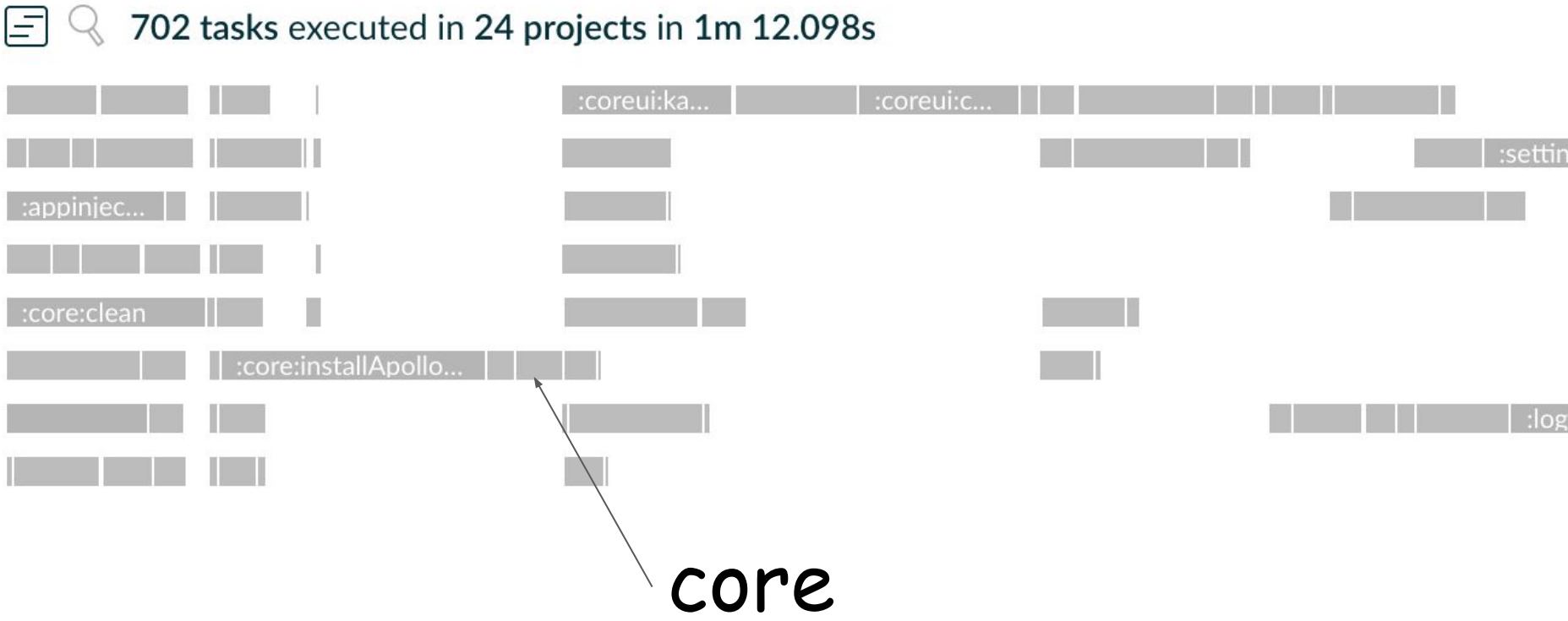
Google Developers Experts



prepare resources

Final build

Google Developers Experts



Final build

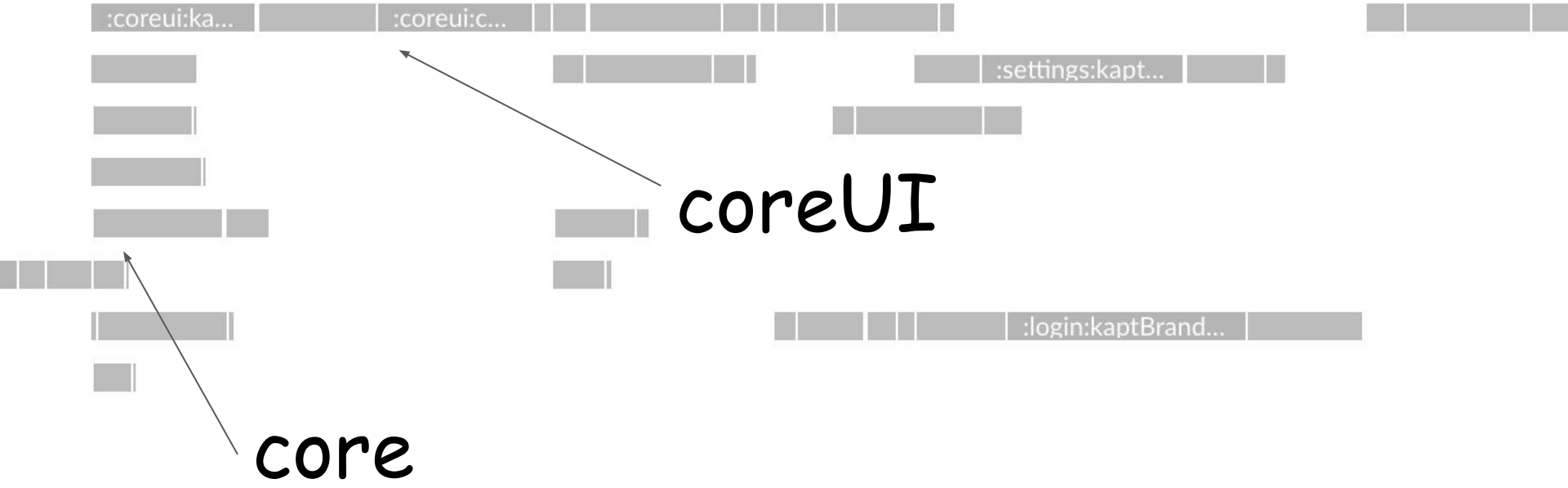
Google Developers Experts

24 projects in 1m 12.098s

Final build

Google Developers Experts

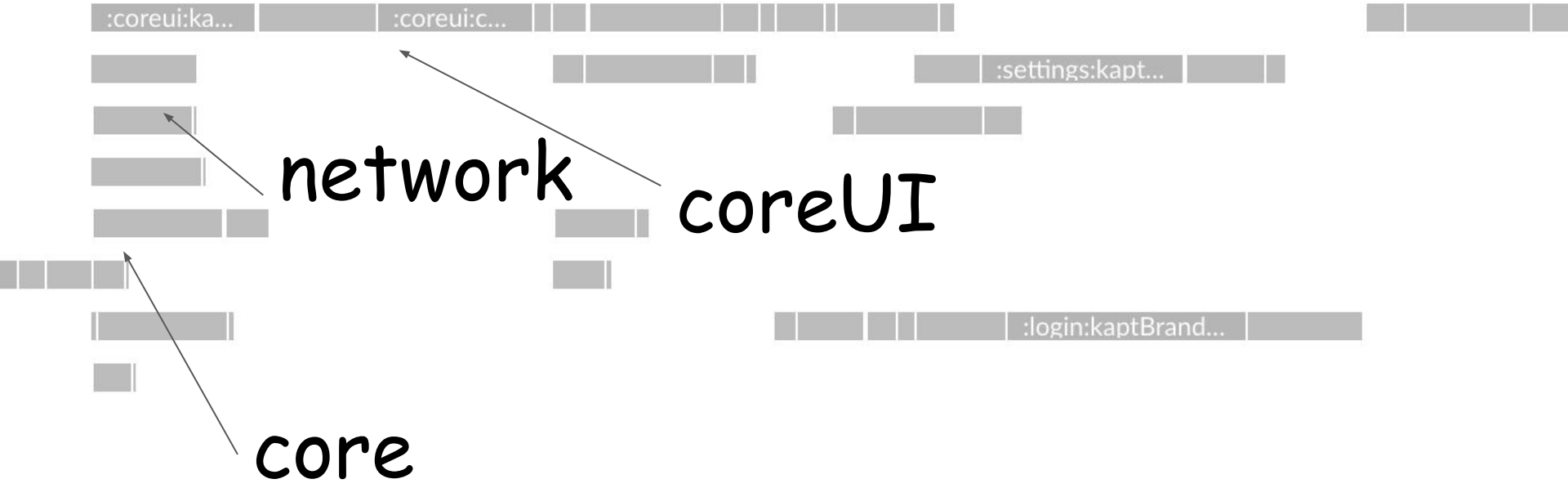
24 projects in 1m 12.098s



Final build

Google Developers Experts

24 projects in 1m 12.098s

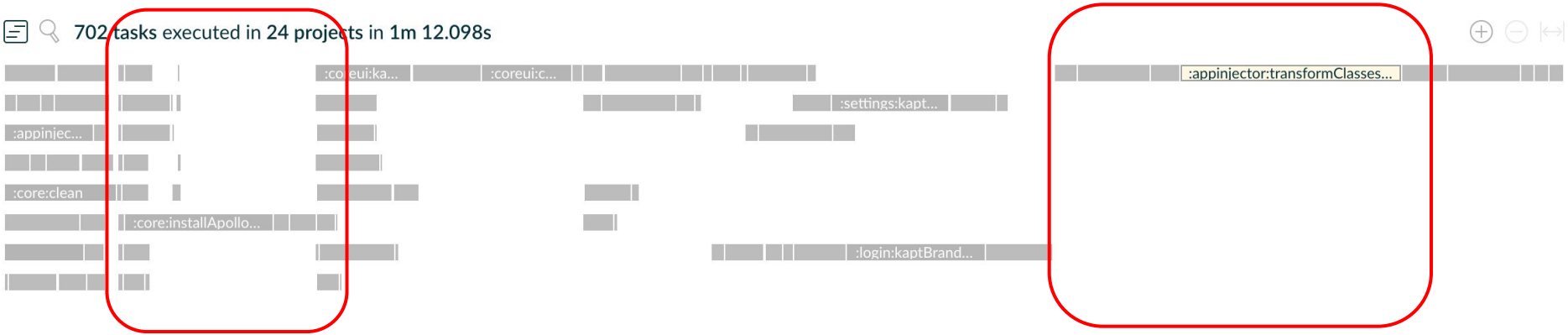


Final build


:appinjector:transformClasses...
:settings:kapt...
rk
coreUI
appinjector
:login:kaptBrand...

Final build (clean). Bottlenecks

Google Developers Experts



Original build (clean)

Google Developers Experts

259 tasks executed in **7 projects** in **7m 17.766s**

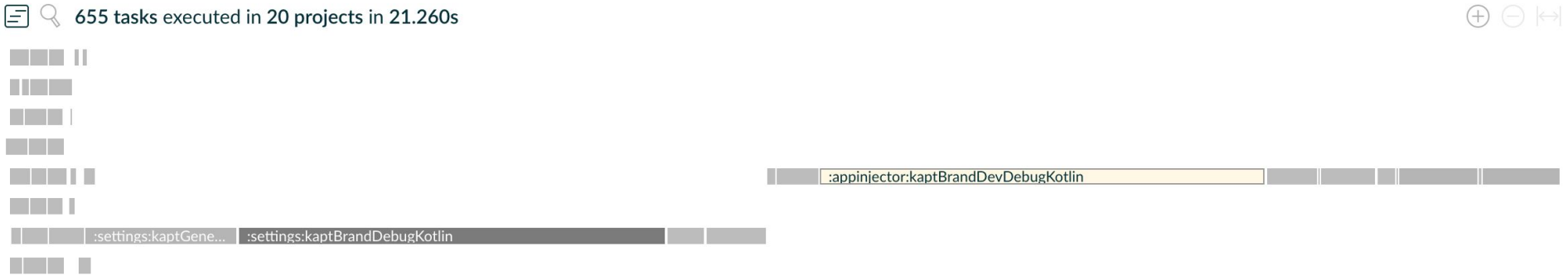
:app:compileDevDebugKotlin	1m 2.069s
:app:kaptDevDebugKotlin	47.606s
:core:compileDevDebugKotlin	41.240s

Final build (clean) ~1.5 min

Google Developers Experts

Final build (incremental). ~30 sec

Google Developers Experts



Rules of happy life

- Keep Core as thin as possible (interfaces only)

Rules of happy life

- Keep Core as thin as possible (interfaces only)
- Avoid APT in modules if possible

Rules of happy life

- Keep Core as thin as possible (interfaces only)
- Avoid APT in modules if possible
- Share code between several modules not all

Rules of happy life

- Keep Core as thin as possible (interfaces only)
- Avoid APT in modules if possible
- Share code between several modules not all
- Keep modules independent

Rules of happy life

- Keep Core as thin as possible (interfaces only)
- Avoid APT in modules if possible
- Share code between several modules not all
- Keep modules independent
- Do not interact between modules by direct calls

Rules of happy life

- Keep Core as thin as possible (interfaces only)
- Avoid APT in modules if possible
- Share code between several modules not all
- Keep modules independent
- Do not interact between modules by direct calls
- Prefer *implements* over *api*

Is that all what I need to know?

Dagger 2

Google Developers Experts

Dagger-Android

Do not use it!

more info:
https://github.com/google/dagger/issues/1092

Dagger Subcomponents

Dagger Subcomponents are initialized inside parent Component

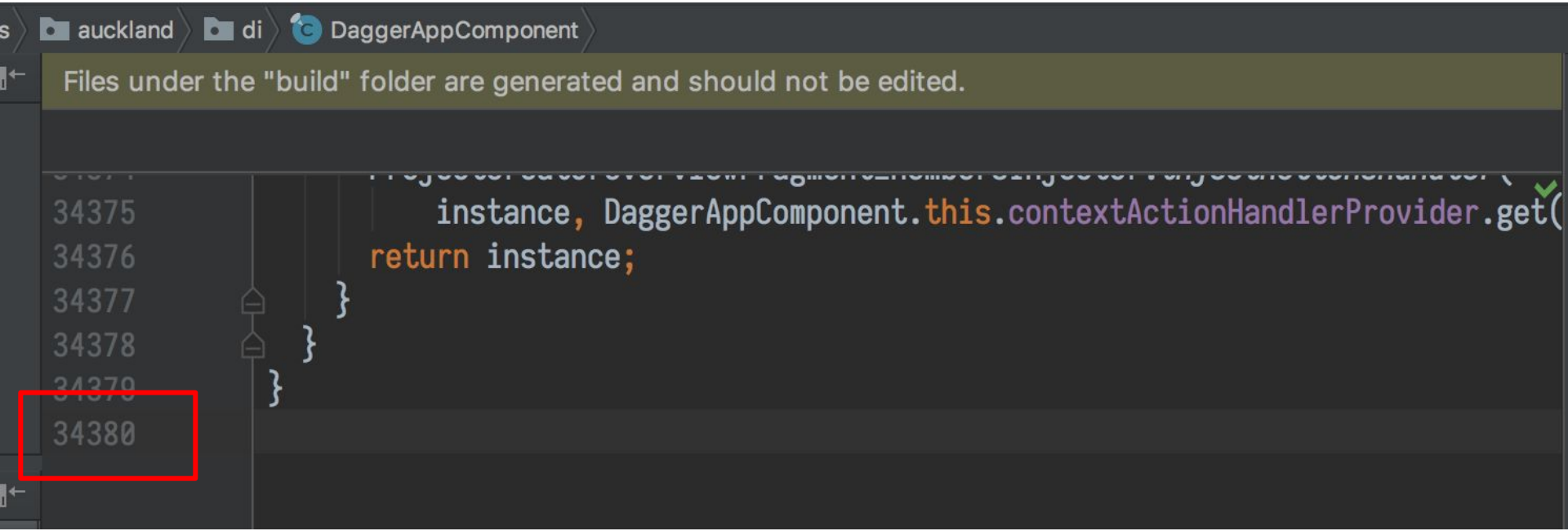
Dagger Subcomponents

Dagger Subcomponents are initialized inside parent Component

And it becomes huge!

Dagger Android

Google Developers Experts



Google Developers Experts

One Gradle module
One Dagger component


FeatureActivityComponent

FeatureActivityComponent

@Inject val toaster:Toaster
@Inject val projectRepo: ProjectRepo

FeatureActivityComponent

@Inject val toaster:Toaster
@Inject val projectRepo: ProjectRepo

AppComponent:
AppProvider

FeatureActivityComponent

@Inject val toaster:Toaster
@Inject val projectRepo: ProjectRepo

AppComponent:
AppProvider

interface AppProvider: RepoProvider, MainToolsProvider

Google Developers Experts

FeatureActivityComponent

```
@Inject val toaster:Toaster
@Inject val projectRepo: ProjectRepo
```

AppComponent:
AppProvider

```
interface AppProvider: RepoProvider, MainToolsProvider
```

```
interface RepoProvider{
        fun projectRepo()
}
```

FeatureActivityComponent

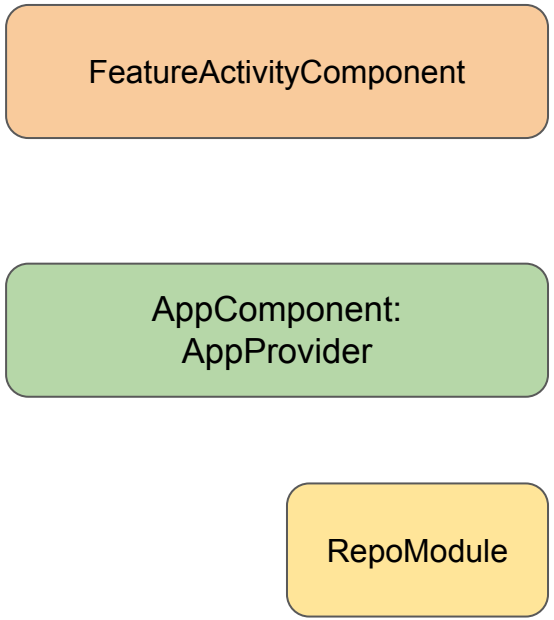
```
@Inject val toaster:Toaster
@Inject val projectRepo: ProjectRepo
```

AppComponent:
AppProvider

```
interface AppProvider: RepoProvider, MainToolsProvider
```

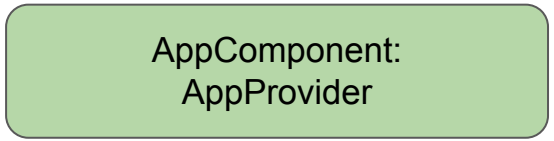
```
interface RepoProvider{
        fun projectRepo()
}

interface MainToolsProvider{
        fun toaster()
}
```

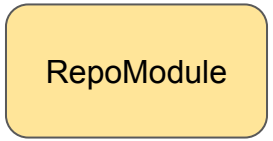


Google Developers Experts

@Inject val toaster:Toaster
@Inject val projectRepo: ProjectRepo



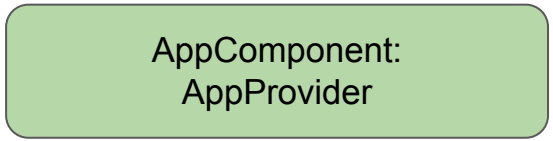
interface AppProvider: RepoProvider, MainToolsProvider



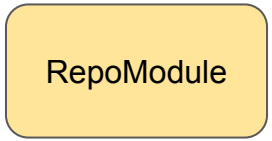
@Provides projectRepo(): ProjectRepo = ProjectRepoImpl()

Google Developers Experts

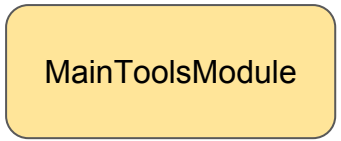
@Inject val toaster:Toaster
@Inject val projectRepo: ProjectRepo



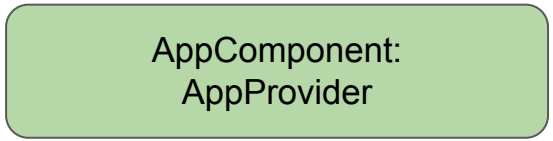
interface AppProvider: RepoProvider, MainToolsProvider



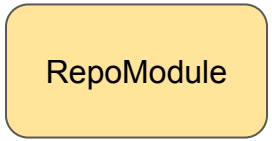
@Provides projectRepo(): ProjectRepo = ProjectRepoImpl()



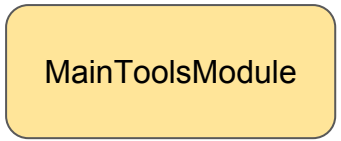
@Inject val toaster:Toaster
@Inject val projectRepo: ProjectRepo



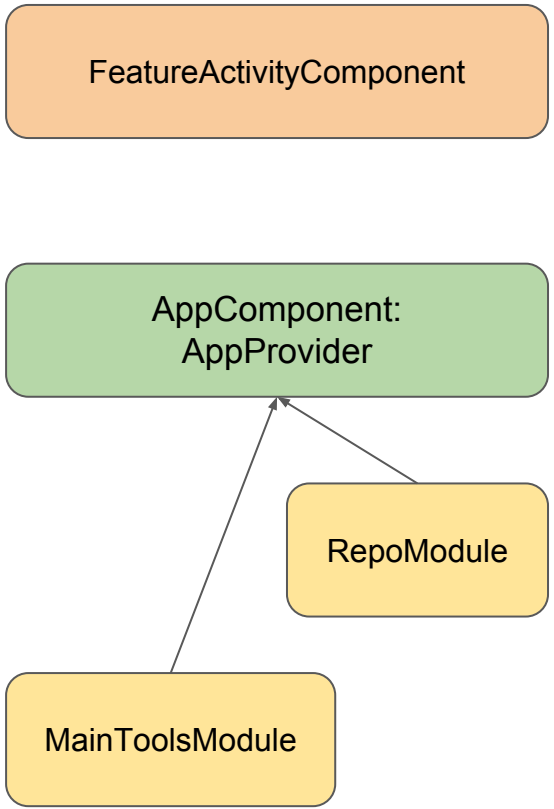
interface AppProvider: RepoProvider, MainToolsProvider



@Provides projectRepo(): ProjectRepo = ProjectRepoImpl()

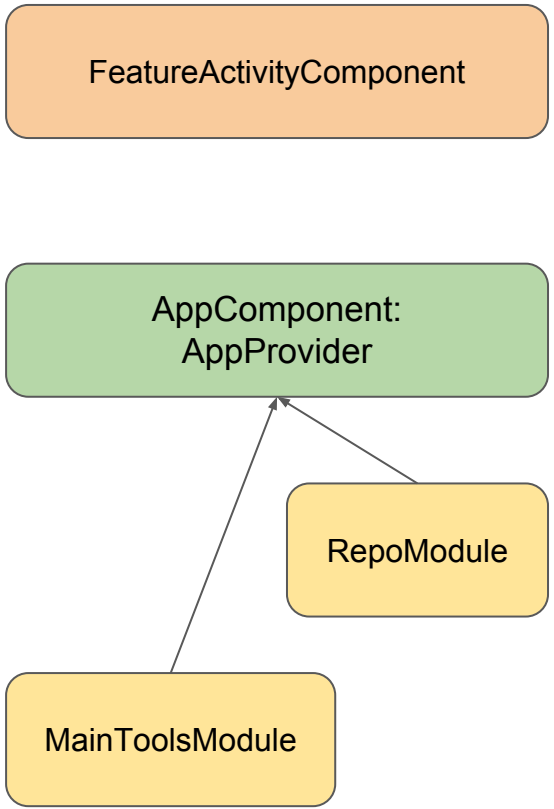


@Provides toaster(): Toaster = ToasterImpl()



@Inject val toaster:Toaster
@Inject val projectRepo: ProjectRepo

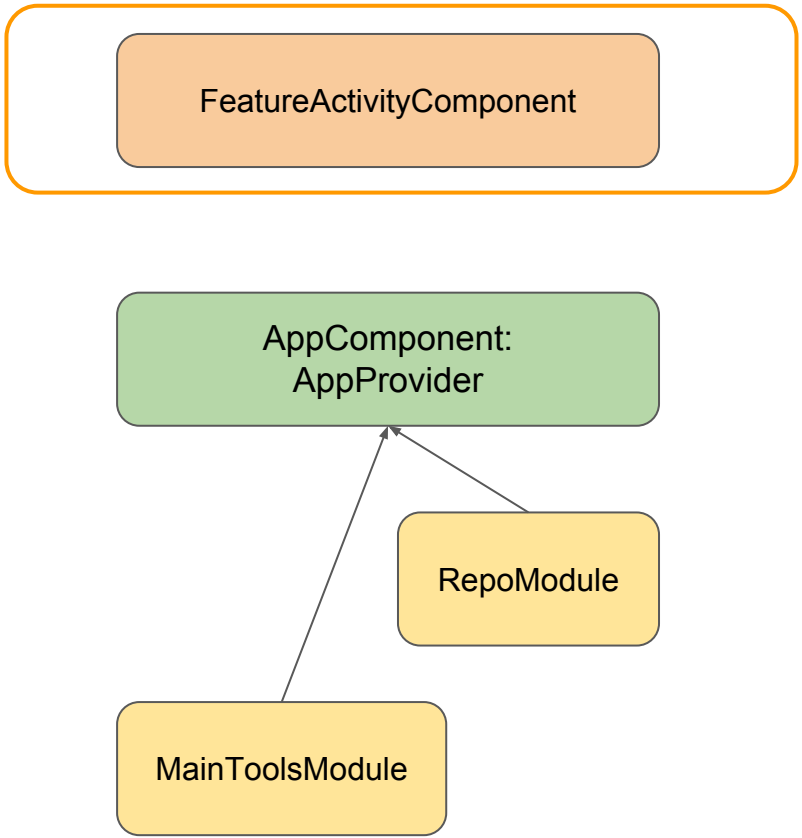
Google Developers
Experts



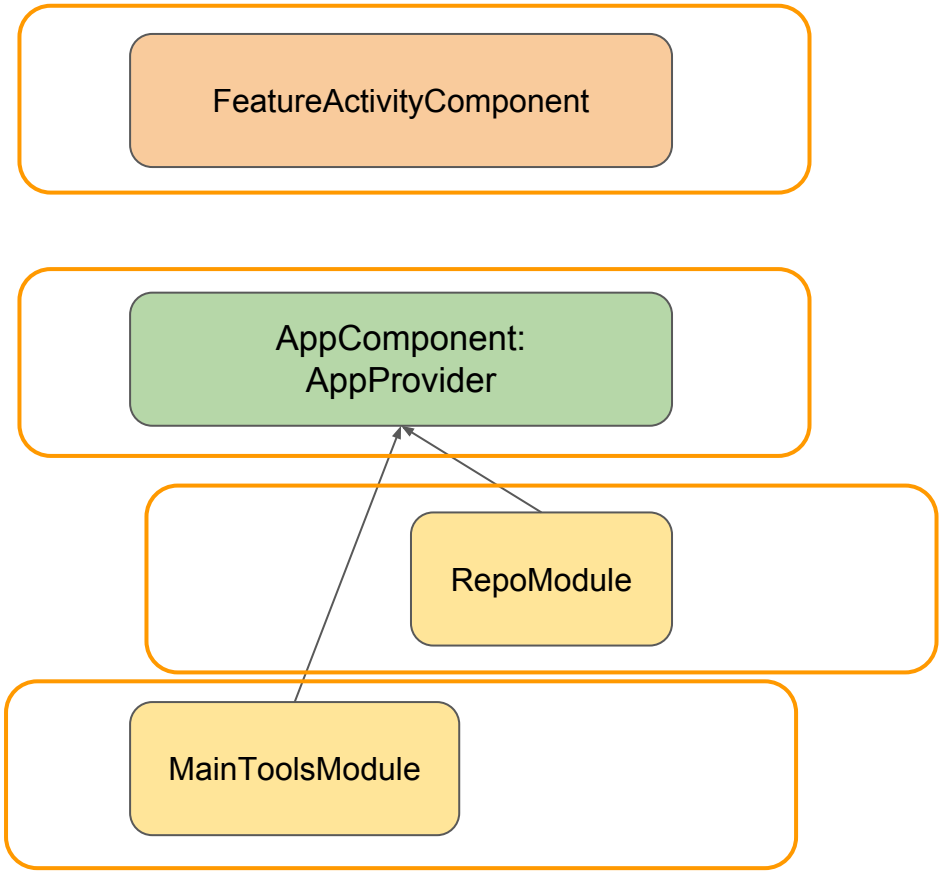

```
@Inject val toaster:Toaster
@Inject val projectRepo: ProjectRepo
```

```
@Component(modules=[
                                MainToolsModule::class,
                                RepoModule::class
                                ])
interface AppCompoment AppProvider
```

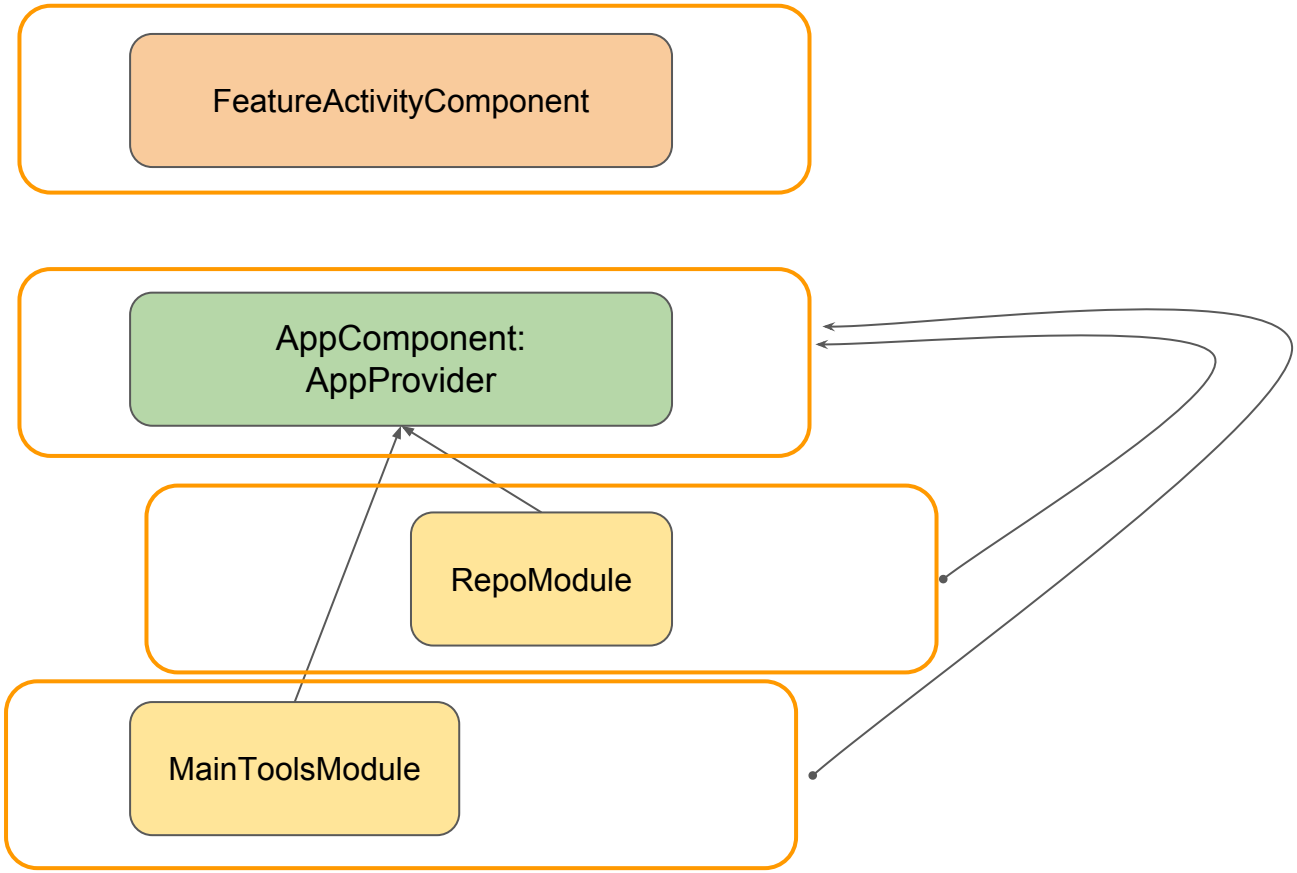
Google Developers Experts



Google Developers
Experts



Google Developers
Experts



Providers generated in AppComponent

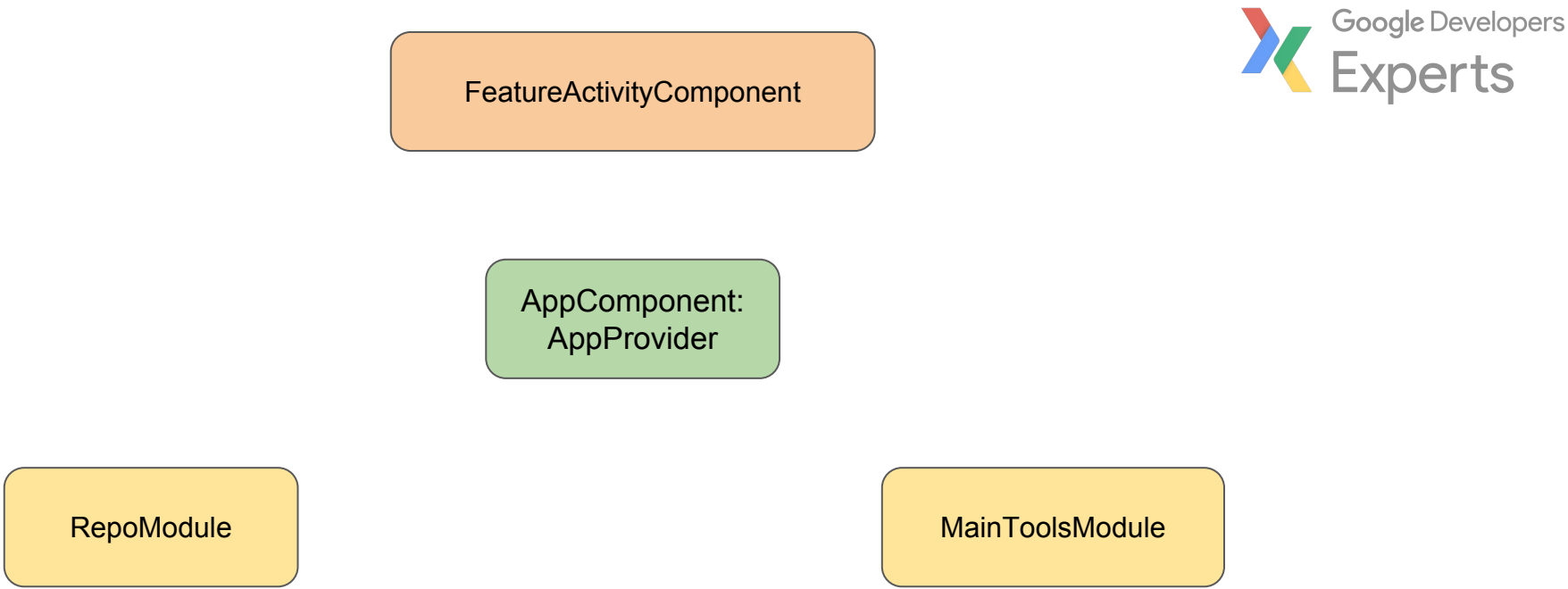
What is wrong?

Modules are generating in AppComponent gradle module (AppInjector)

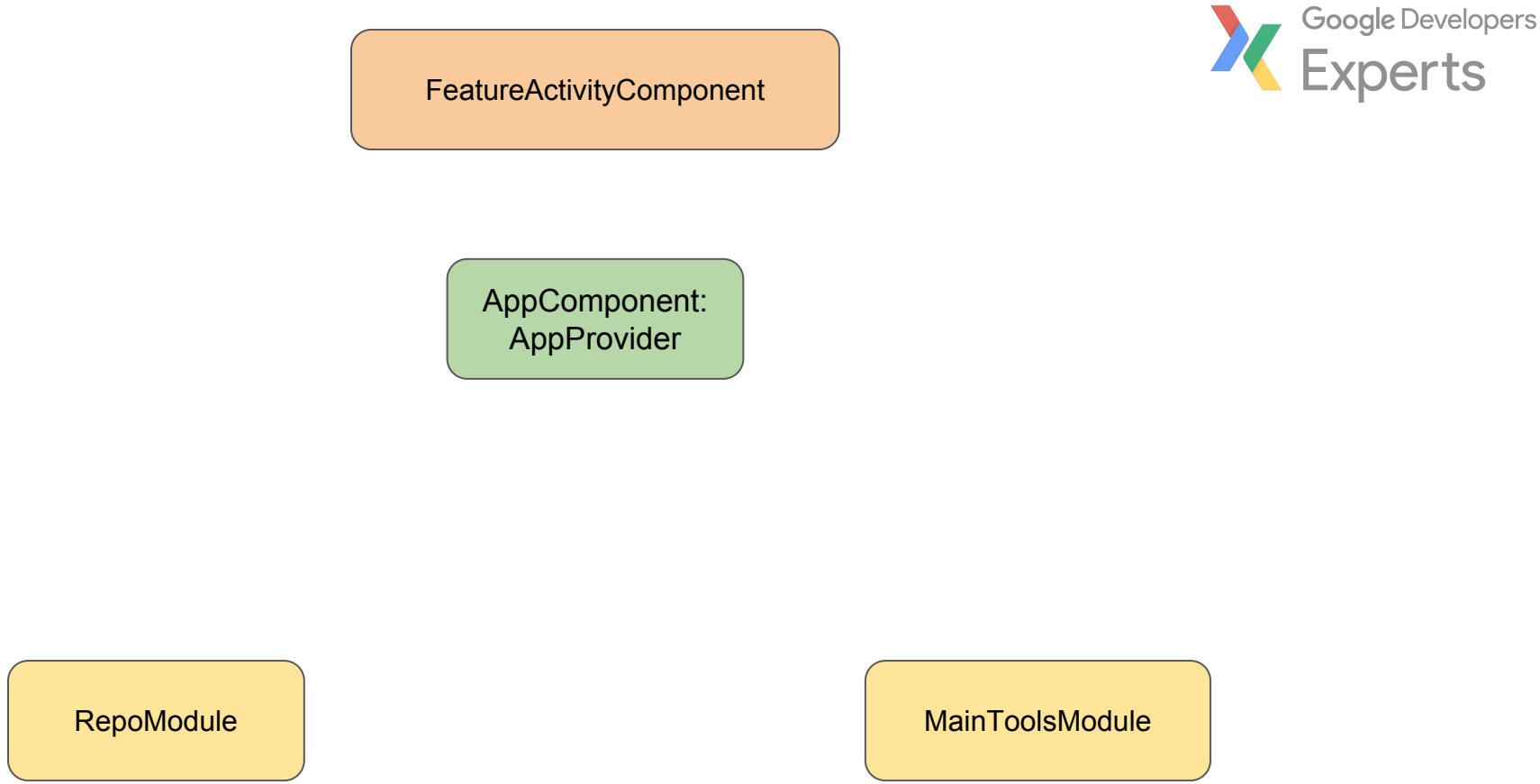
What is wrong?

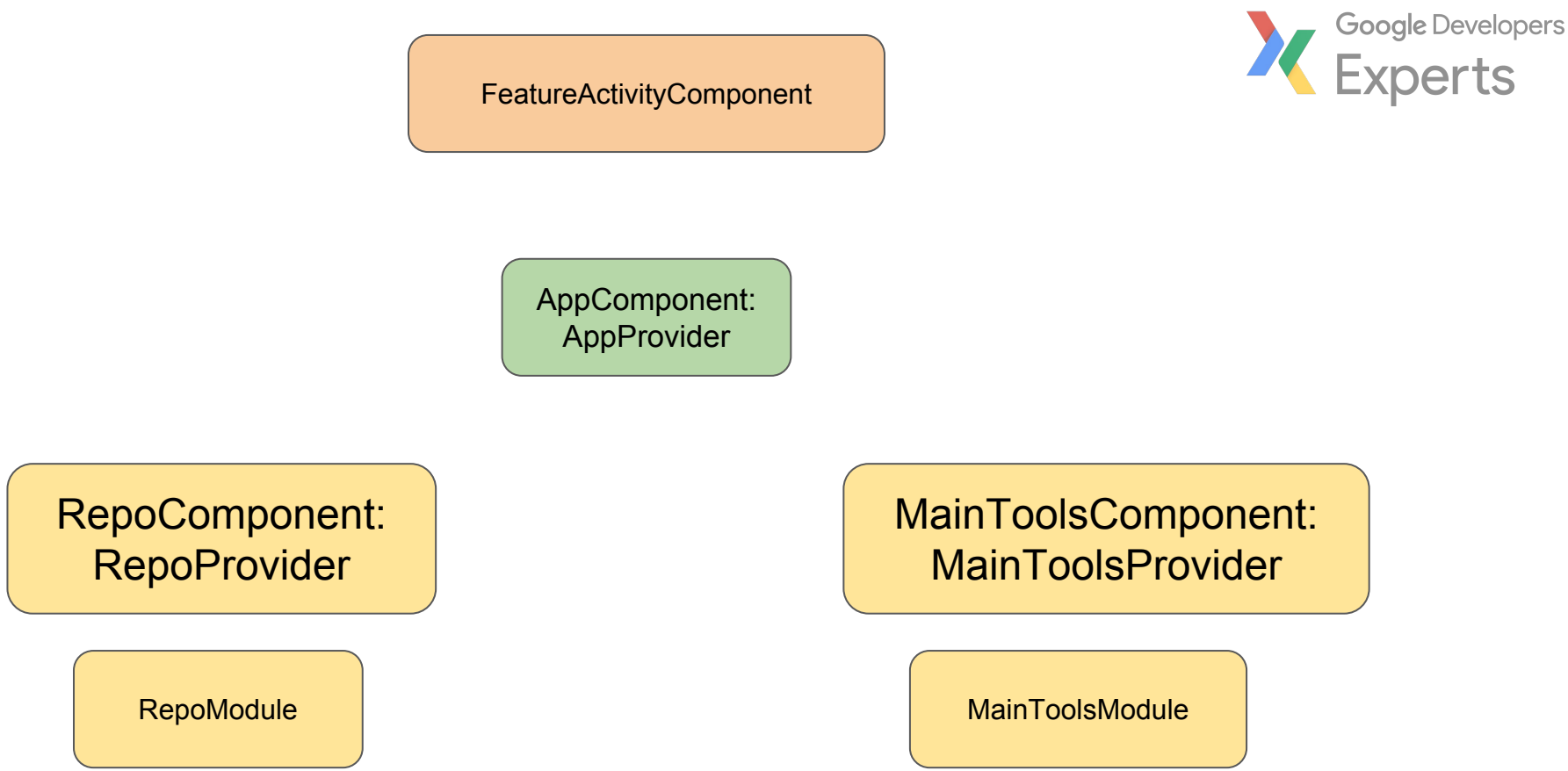
Modules are generating in AppComponent’s gradle module

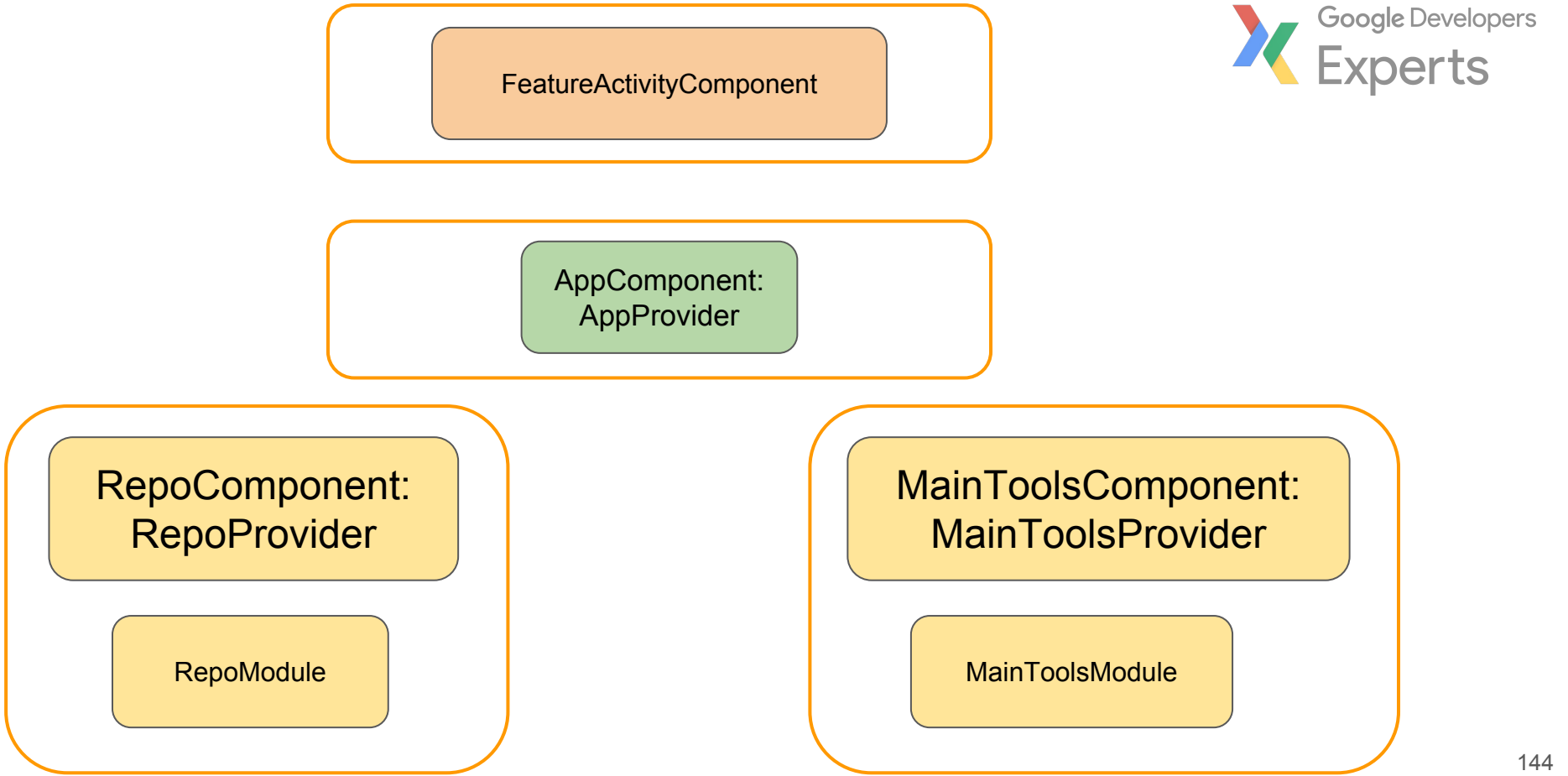
AppComponent knows about each dependency implementation



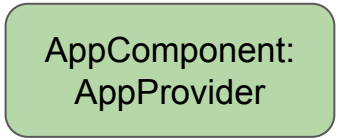
Google Developers
Experts


FeatureActivityComponent
AppComponent:
AppProvider
RepoModule
MainToolsModule


FeatureActivityComponent
AppComponent:
AppProvider
RepoComponent:
RepoProvider
MainToolsComponent:
MainToolsProvider
RepoModule
MainToolsModule


FeatureActivityComponent
AppComponent:
AppProvider
RepoComponent:
RepoProvider
RepoModule
MainToolsComponent:
MainToolsProvider
MainToolsModule

Holds reference to AppProvider



Holds reference to AppProvider

FeatureActivityComponent

Holds reference to RepoProvider and MainToolsProvider

AppComponent: AppProvider

RepoComponent: RepoProvider

MainToolsComponent: MainToolsProvider

Google Developers Experts

Holds reference to AppProvider

FeatureActivityComponent

Holds reference to RepoProvider and MainToolsProvider

AppComponent: AppProvider

RepoComponent: RepoProvider

MainToolsComponent: MainToolsProvider

implements RepoProvider

implements RepoProvider

Benefits

AppComponent becomes lightweight

AppComponent knows about each dependency implementation

Benefits

AppComponent becomes lightweight

AppComponent does not know about dependencies implementations

Dynamic module switching

settings.gradle

Google Developers Experts

```
include ':core'
project(':core').projectDir = new File(rootDir, 'sources/base/core')
include ':injector'
project(':injector').projectDir = new File(rootDir, 'sources/base/injector')
include ':repo'
project(':repo').projectDir = new File(rootDir, 'sources/base/repo')
include ':network'
project(':network').projectDir = new File(rootDir, 'sources/base/network')

include ':notifications'
project(':notifications').projectDir = new File(rootDir,'sources/features/notifications')

include ':githubbrowser'
project(':githubbrowser').projectDir = new File(rootDir, 'sources/features/githubbrowser')
include ':mainscreen'
project(':mainscreen').projectDir = new File(rootDir, 'sources/features/mainscreen')
```

settings.gradle


```
include ':notifications'
project(':notifications').projectDir = new File(rootDir,'sources/features/notifications')
```

settings.gradle

Google Developers Experts

```
include ':notifications'
project(':notifications').projectDir = new File(rootDir,'sources/features/notifications/notifications-fake')
```

Benefits?

Each module take up to 30 seconds on clean build.
After changing to fake it takes around 5 seconds.

For incremental in most cases it does not make sense

Google Developers Experts

Other solutions for fast build

Mainframer?

Google Developers Experts

https://github.com/gojuno/mainframer

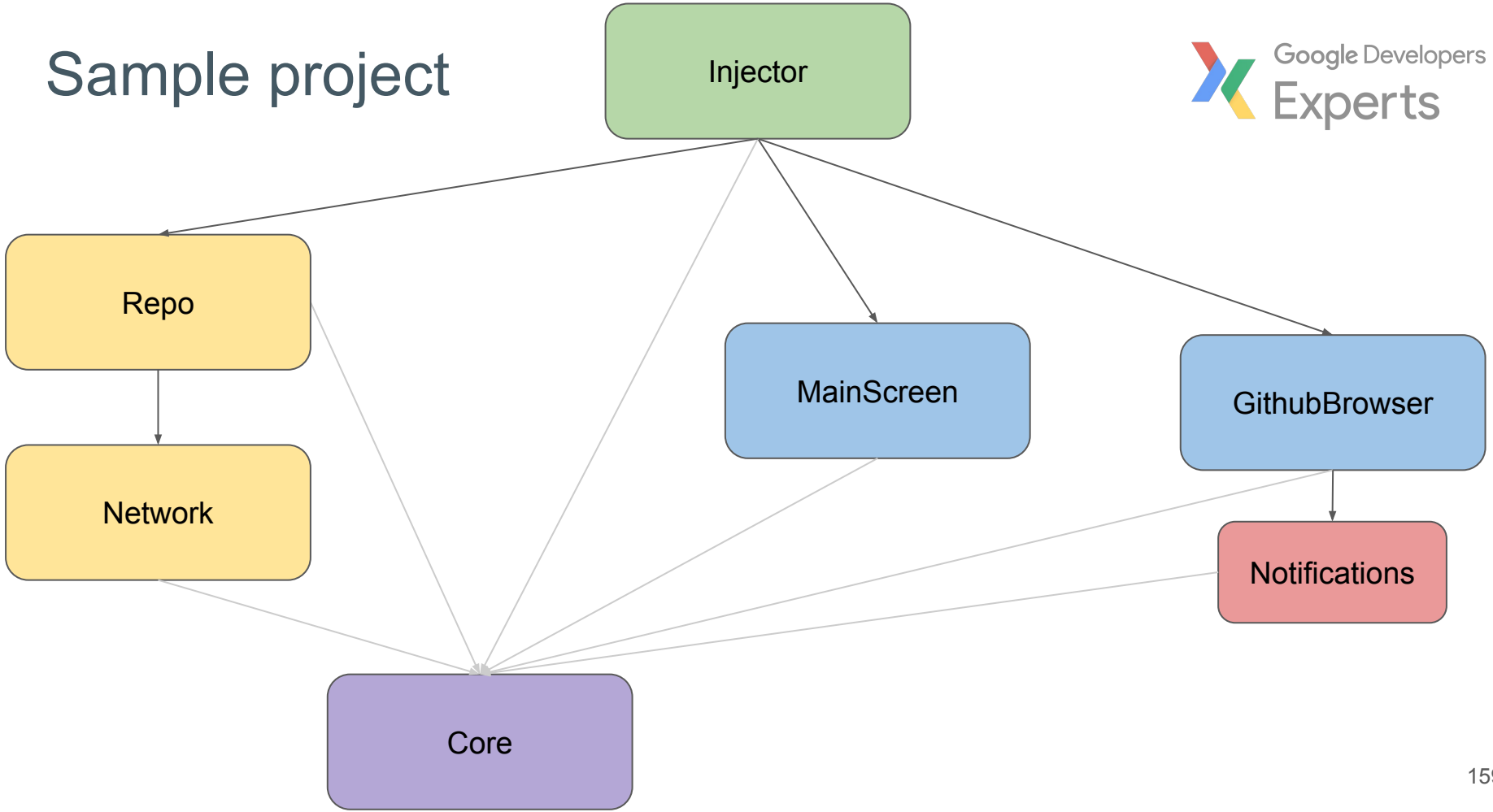
Google Developers Experts

Buck or Bezel?

Google Developers Experts

Sample project

Sample project



Google Developers Experts

Google Developers Experts

Sample project:

github.com/kotlinsg/modular

To read:

Google Developers Experts

atscaleconference.com/videos/app-modularization-and-module-lazy-loading/ - модуляризация в Instagram

medium.com/@manuelvicnt/dagger-android-thoughts-dependency-injection-in-android-d50e799affe3 - почему dagger-android не идеален

guides.gradle.org/performance/ - превосходный материал по профилированию сборки Gradle

Thank You!

Google Developers Experts

Denis Nek
Google Developer Expert
@nekdenis

Большой вклад в проект внесли:

Степан Гончаров
Кирилл Бояршинов
Андрей Мищенко

Сергей Боиштян