my life before and after meeting

CLEAN ARCHITECTURE

Roc Boronat

a **pragmatic** clean architecture lover

my life before

…was young, need the money

logic everywhere

a smart **.jsp** with a smart **.js** calling a smarty **java** that is calling a **stored procedure**

high coupling

«pass the Context, it’s funny!»

low cohesion

classes that recieve an httprequest, check the database and fill the cache before answering

patches everywhere

android «magic» fixes and my own business rules are totally mixed

test? what test?

working with architectures driven by people who doesn’t test
run the app to test eeeeeeeeeeeeeeverything

my life after

thank you InfoJobs!

Robert C. Martin

SOLID
Clean Architecture
Software Craftmanship
Agile Manifesto

SOLID?

SOLID

single responsability principle

do one thing well

SOLID

open / closed principle

open for extension, closed for modification

SOLID

liskov substitution principle

sorry! cannot write a good short definition!

SOLID

interface segregation principle

lots of small interfaces instead of a big one

SOLID

dependency inversion principle

depend upon abstractions, not upon concretions

CLEAN ARCHITECTURE?

…let the party start!

what?

independent of frameworks

frameworks are tools, let’s focus on the **business rules**.

what?

testable

business rules can be tested without UI, databases, web servers, a phone, a banana

what?

independent of ui

the UI can change ‘cause the **business rules** remain the same

what?

independent of database

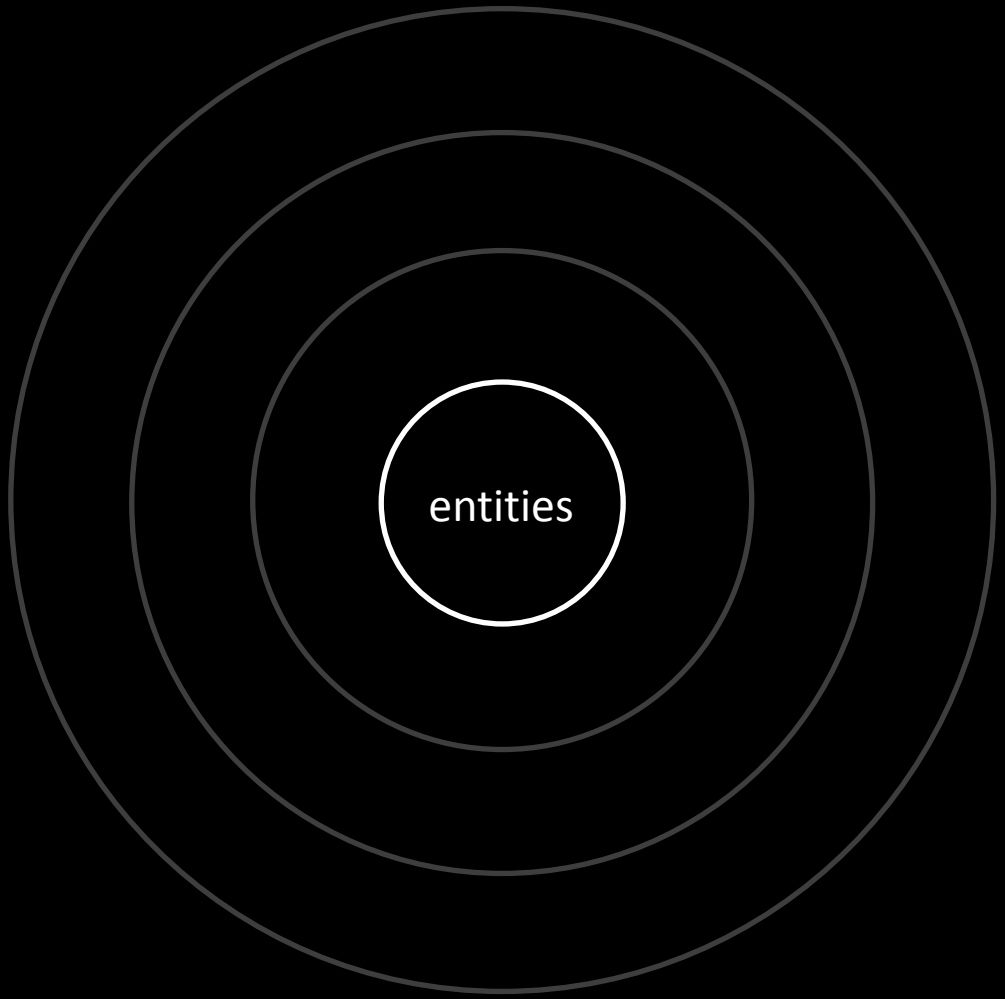
business rules are not bound to the database, so you can change it

what?

independent of external *

business rules don’t know anything about the outside world

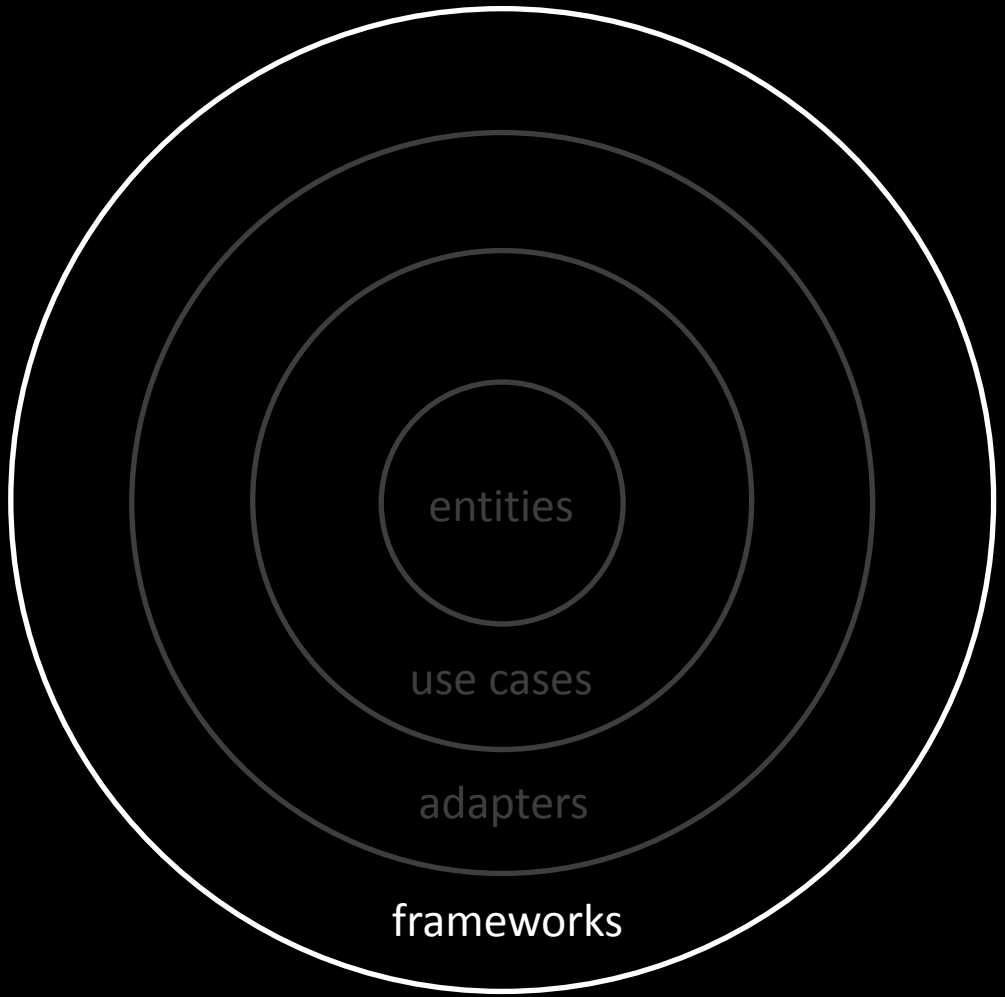
show me something **please**



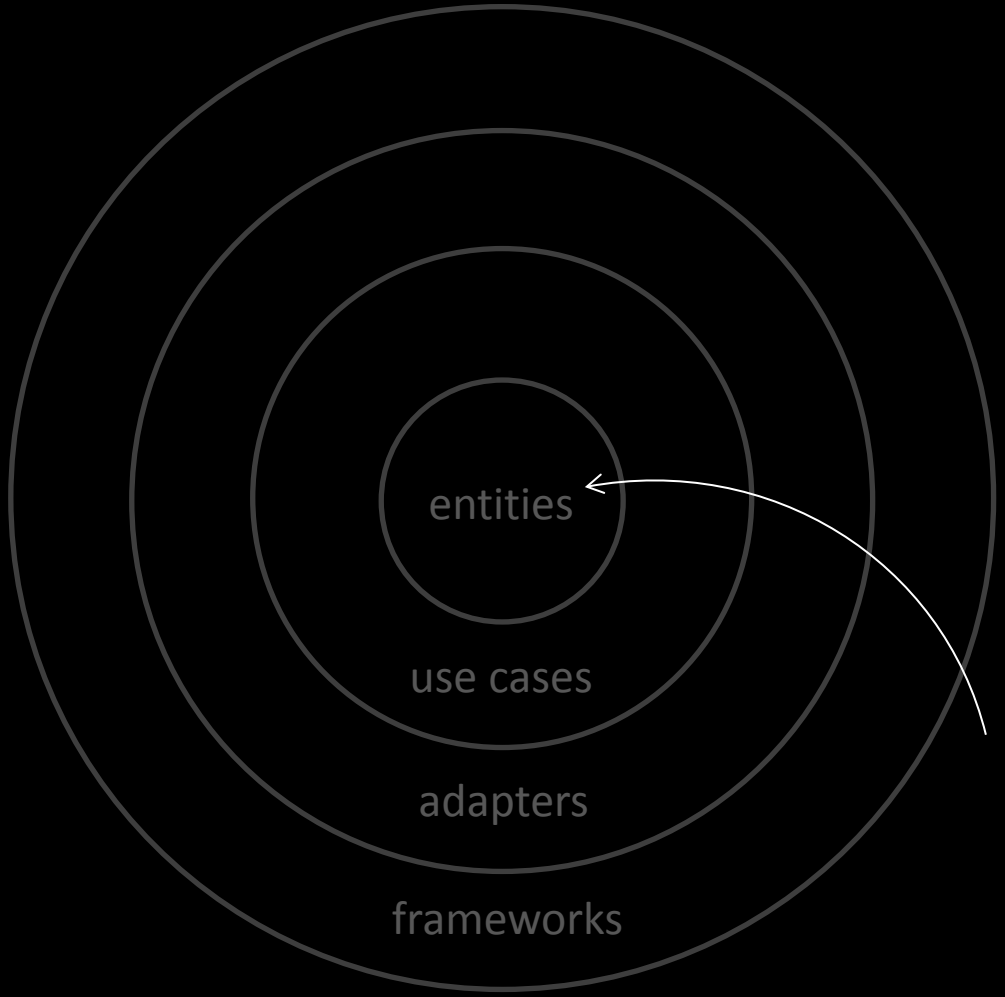
- **enterprise wide** business rules
- could be used by **different applications** in the enterprise
- these are the less likely to change when some external changes

- **application specific** business rules
- **they use entities** to achieve a goal
- it will change if **the operation of the application** changes
- ui? databases? **It's totally isolated**

- **translates data** from use case and entities to an external agency
- **converts requests** from the outer layers to the inner ones

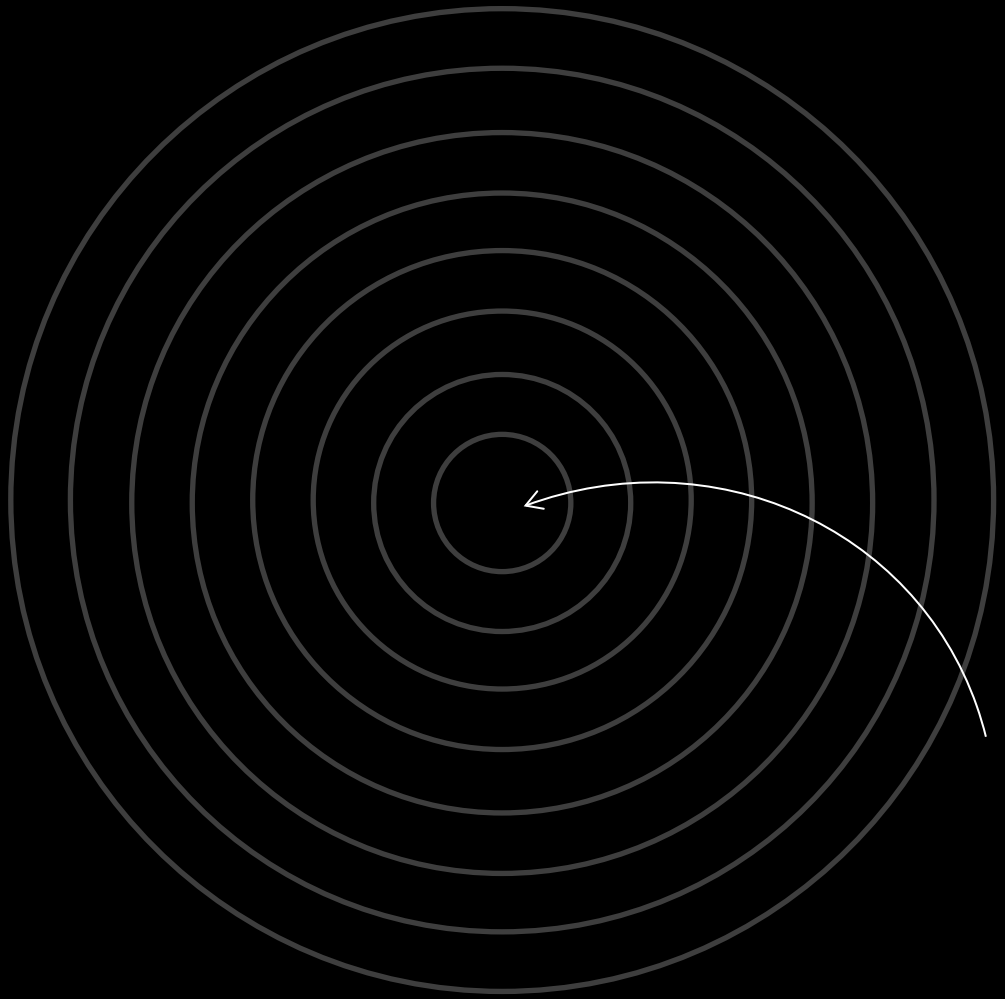


- android screens, web, database, a distributed cache… are **details**
- keeping them outside **where they can do little harm**



The Dependency Rule

- entities are alone
- use cases know entities
- adapters know use cases
- frameworks know adapters



Only four circles?

- nope, do what you need

but

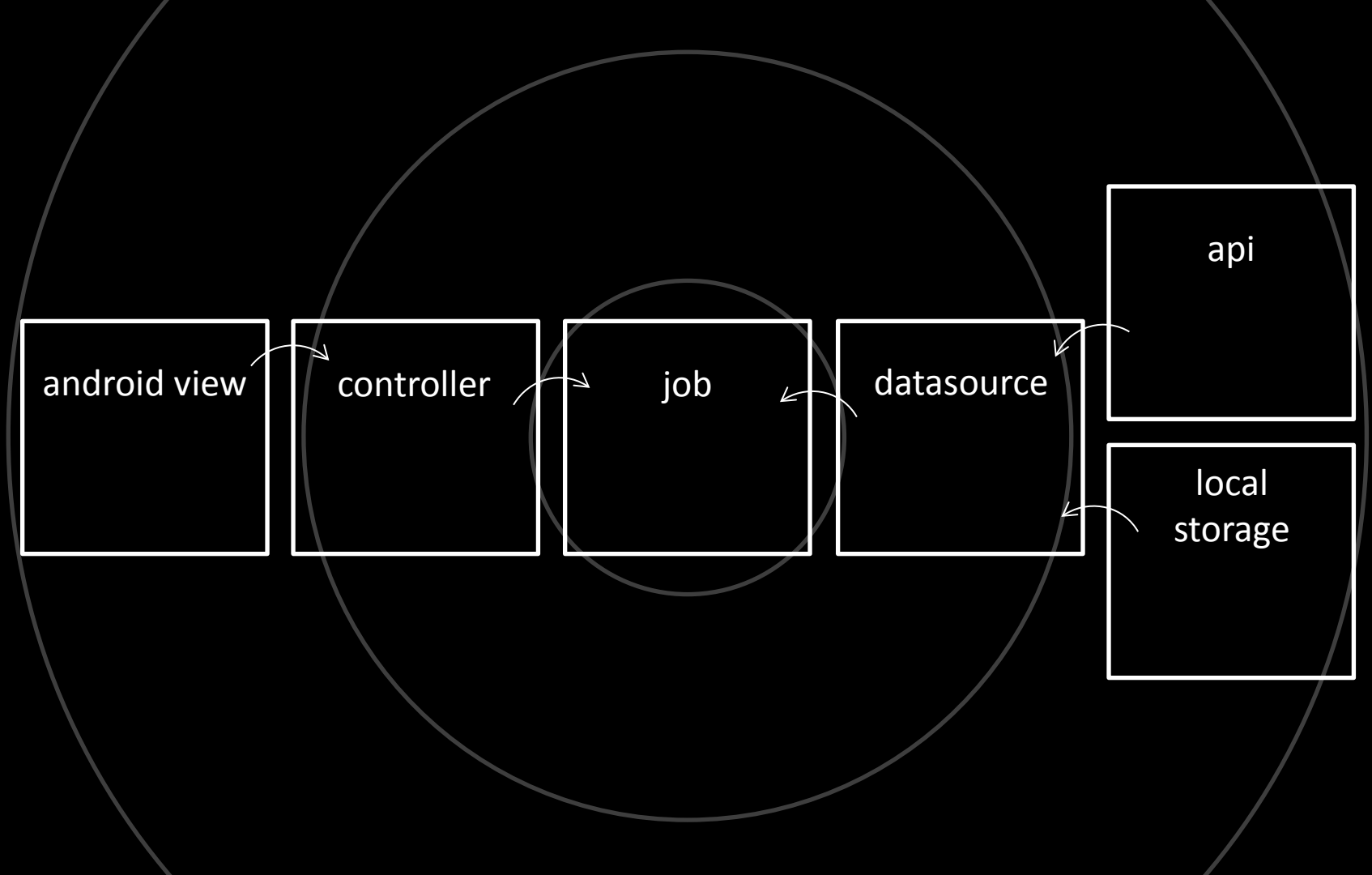
- Respect **The Dependency Rule**

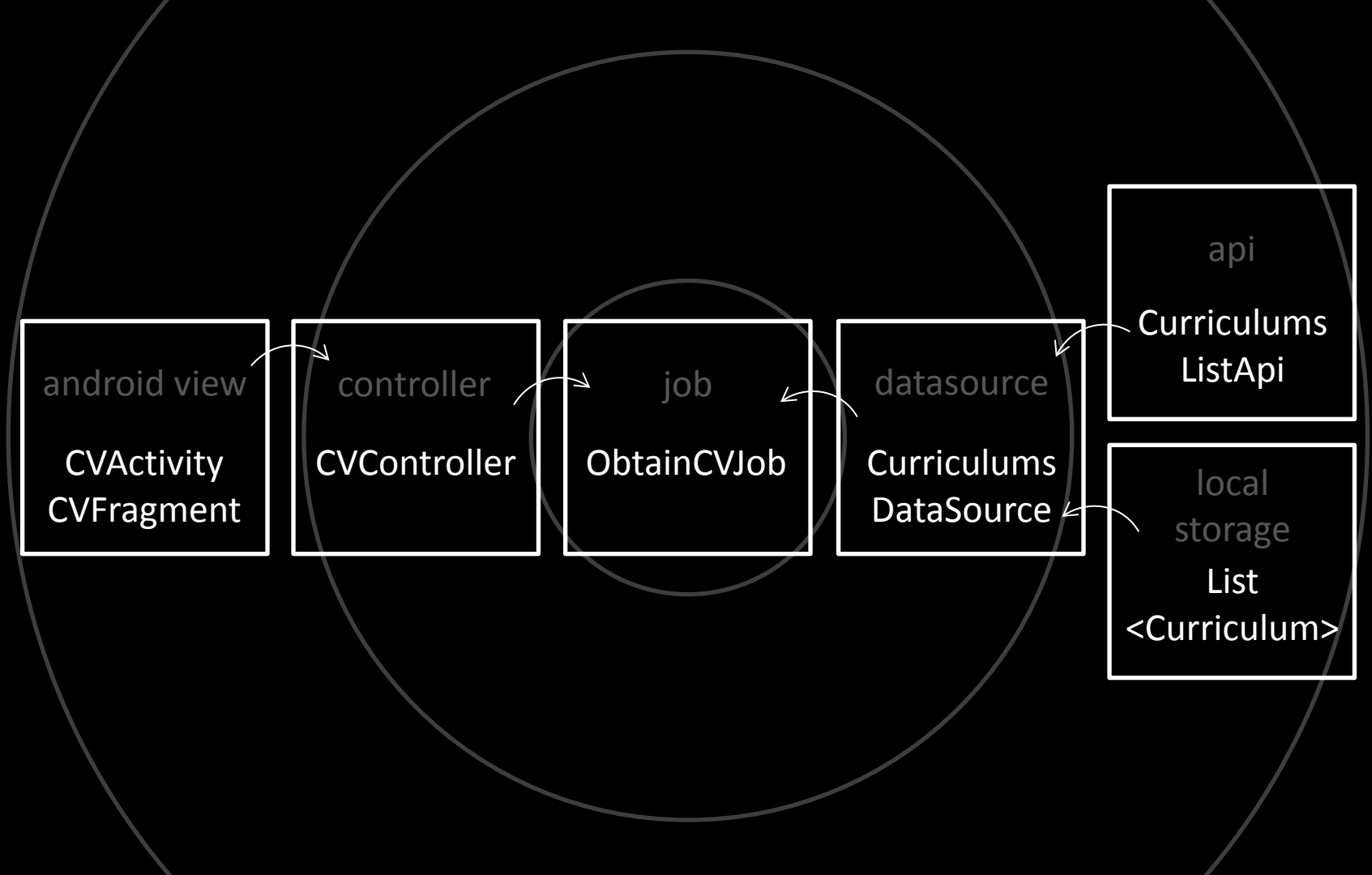
what about the code?

tired of abstractness?

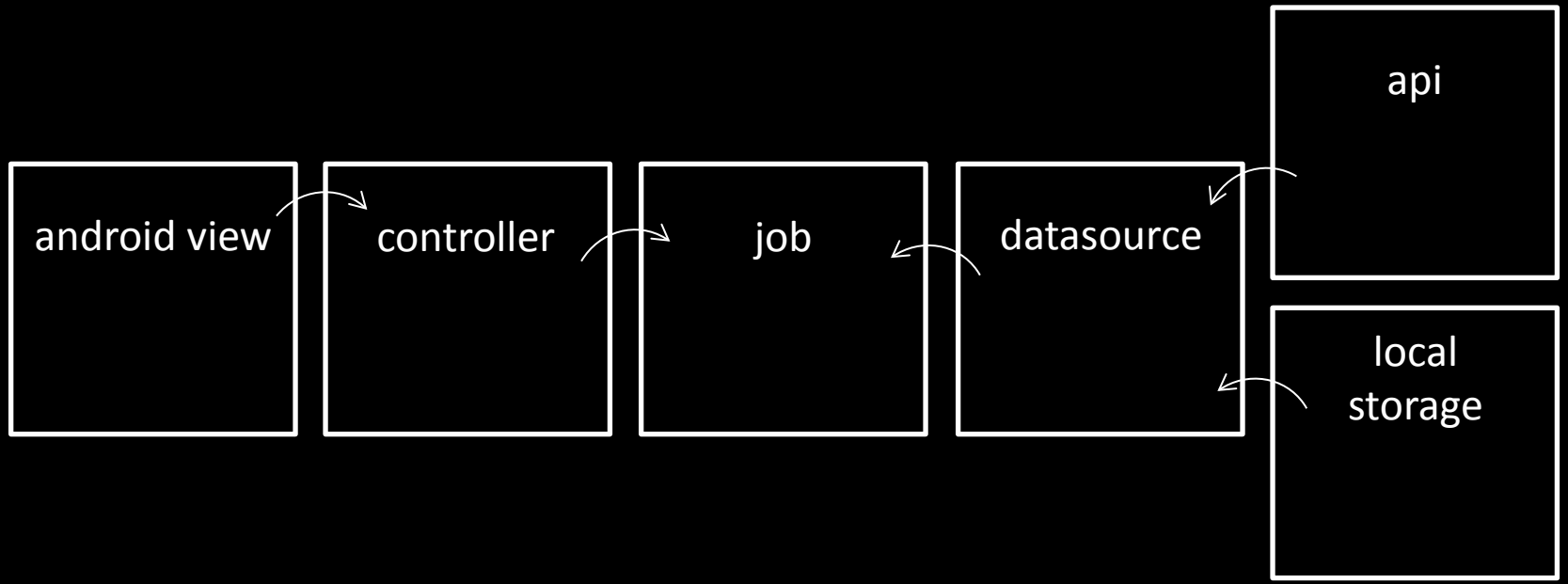
Jorge Barroso

InfoJobs' architecture **father**
Karumi Android Expert

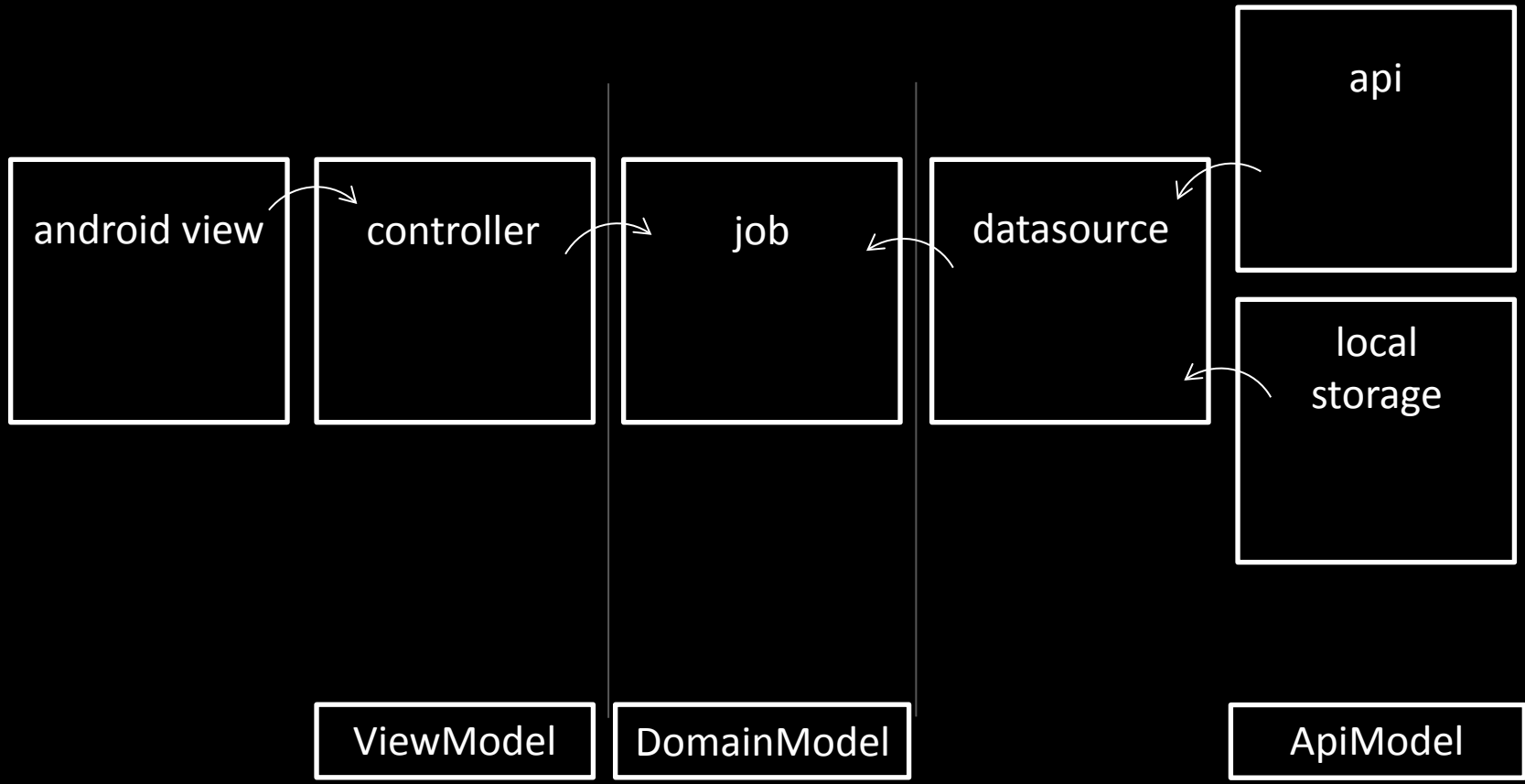

android view
controller
job
datasource
api
local
storage


android view
CVActivity
CVFragment
controller
CVController
job
ObtainCVJob
datasource
Curriculums
DataSource
api
Curriculums
ListApi
local
storage
List
<Curriculum>

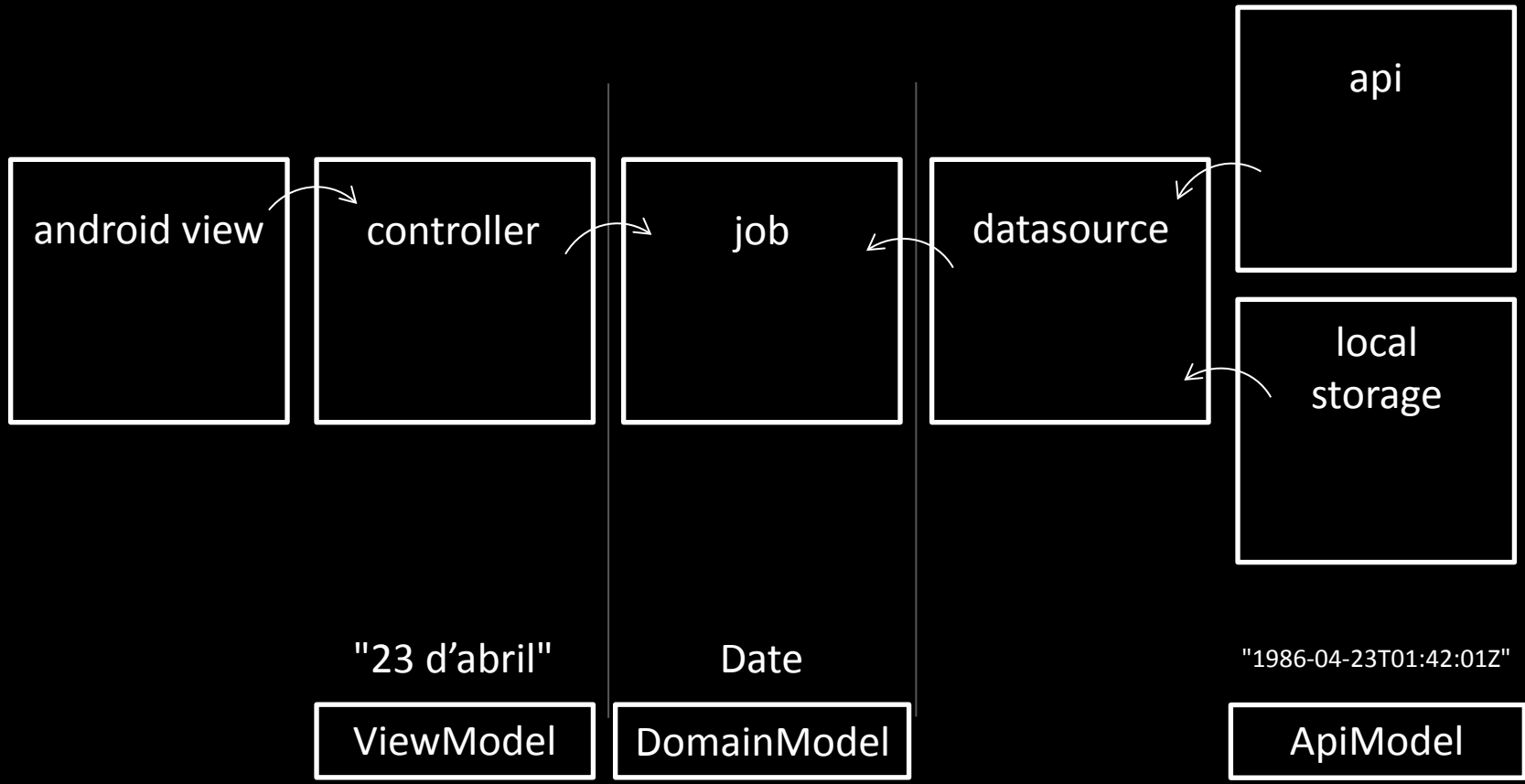
how the data travels?



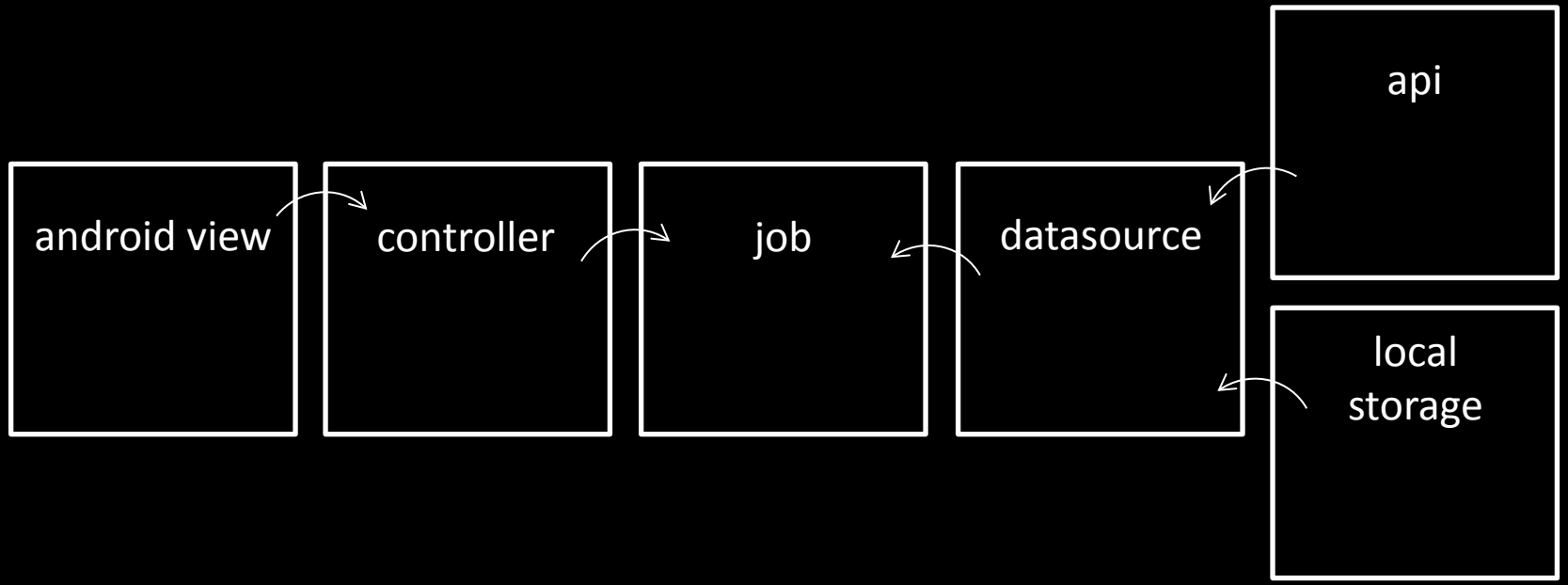
how the data travels?



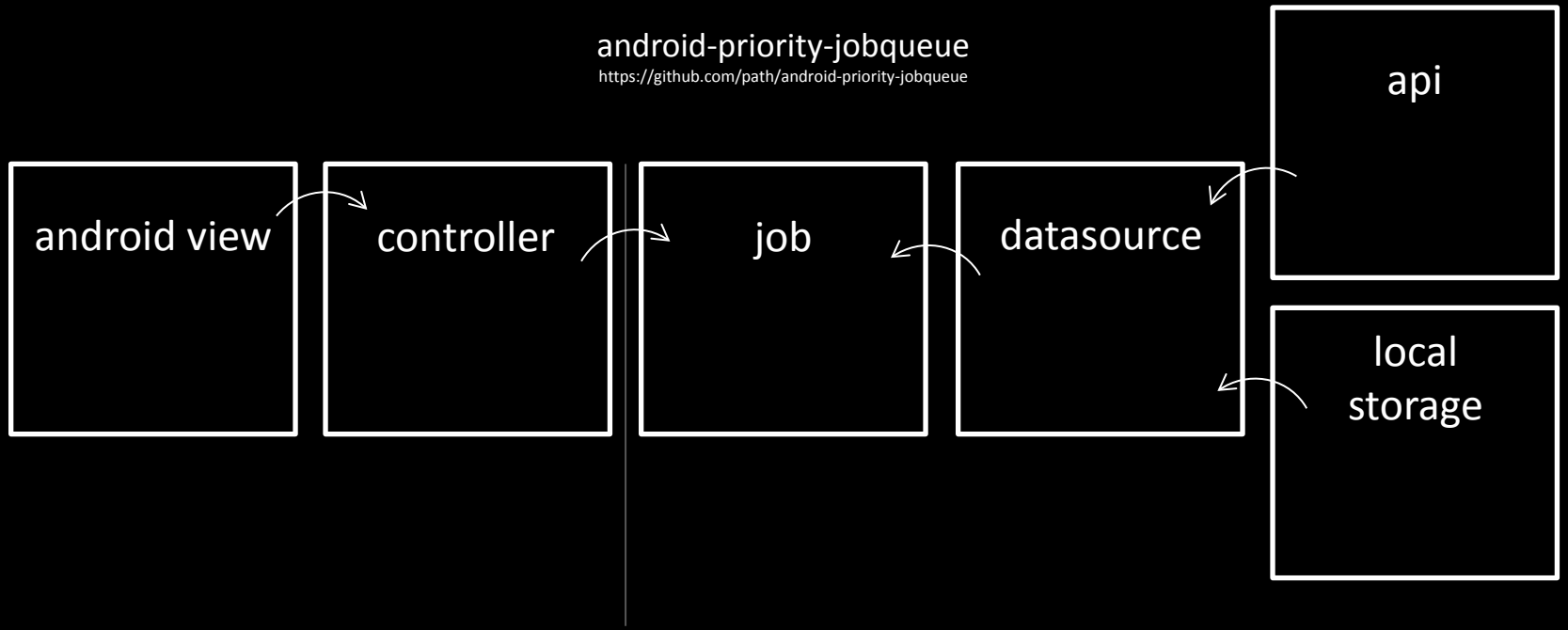
how the data travels?



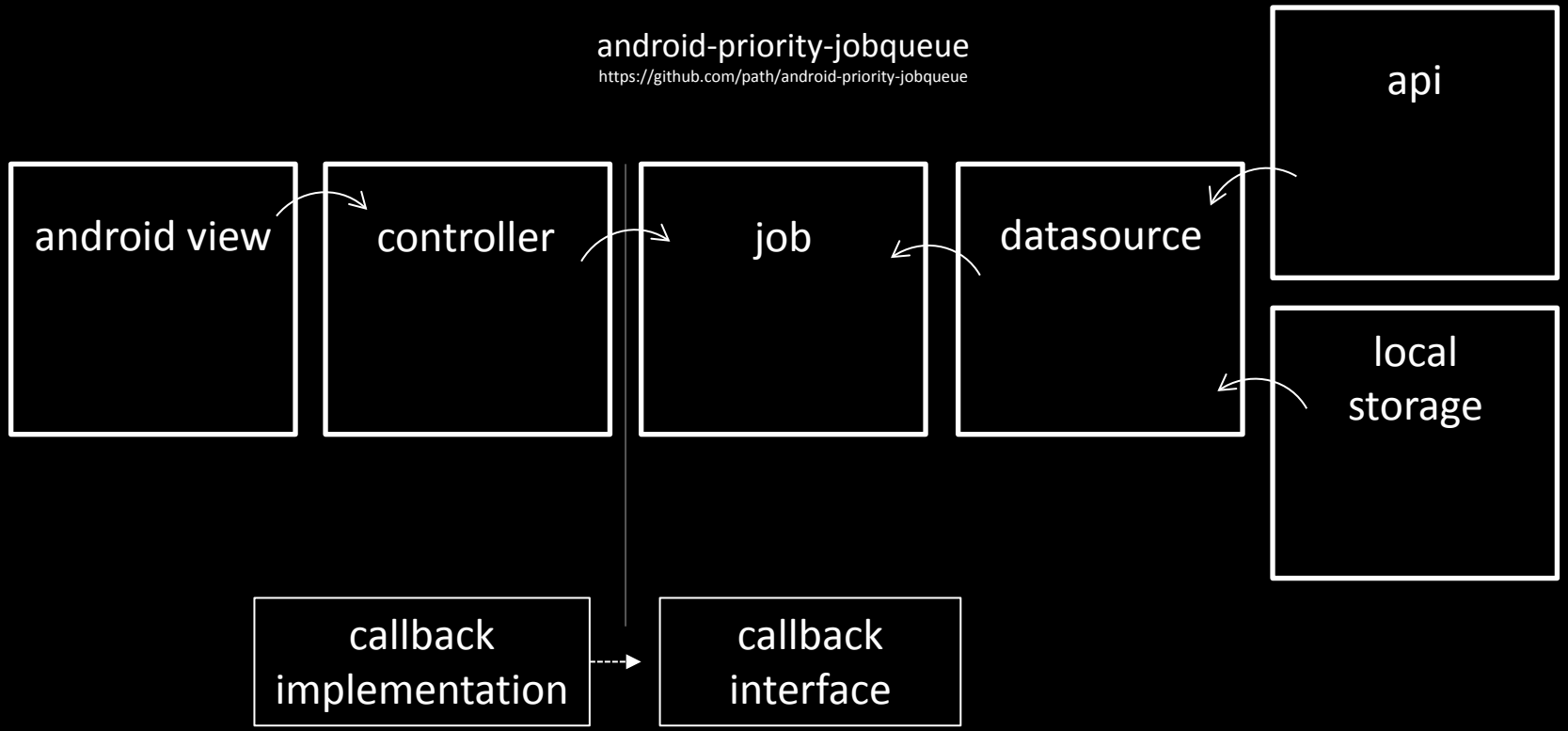
how to manage async?



how to manage async?



how to manage async?



it’s question time!

references

“Clean Architecture and Design” by **Uncle Bob**
https://www.youtube.com/watch?v=Nsjsiz2A9mg

“The Clean Architecture” by **Uncle Bob**
https://blog.8thlight.com/uncle-bob/2012/08/13/the-clean-architecture.html

“Forgetting Android” by **Jorge Barroso**
https://www.youtube.com/watch?v=ROdIvrLL1ao
http://www.slideshare.net/flipper83/forgetting-android-v2

“Architecting Android… the clean way?” by **Fernando Cejas**
http://fernandocejas.com/2014/09/03/architecting-android-the-clean-way

“What is all this Clean Architecture jibber-jabber about?” by **Pablo Guardiola**
http://pguardiola.com/blog/clean-architecture-part-1/
http://pguardiola.com/blog/clean-architecture-part-2/

“Introducción a Clean Architecture” by **Tempos21**
http://www.tempos21.com/web/blog/introduccion-clean-architecture