SCHIBSTED

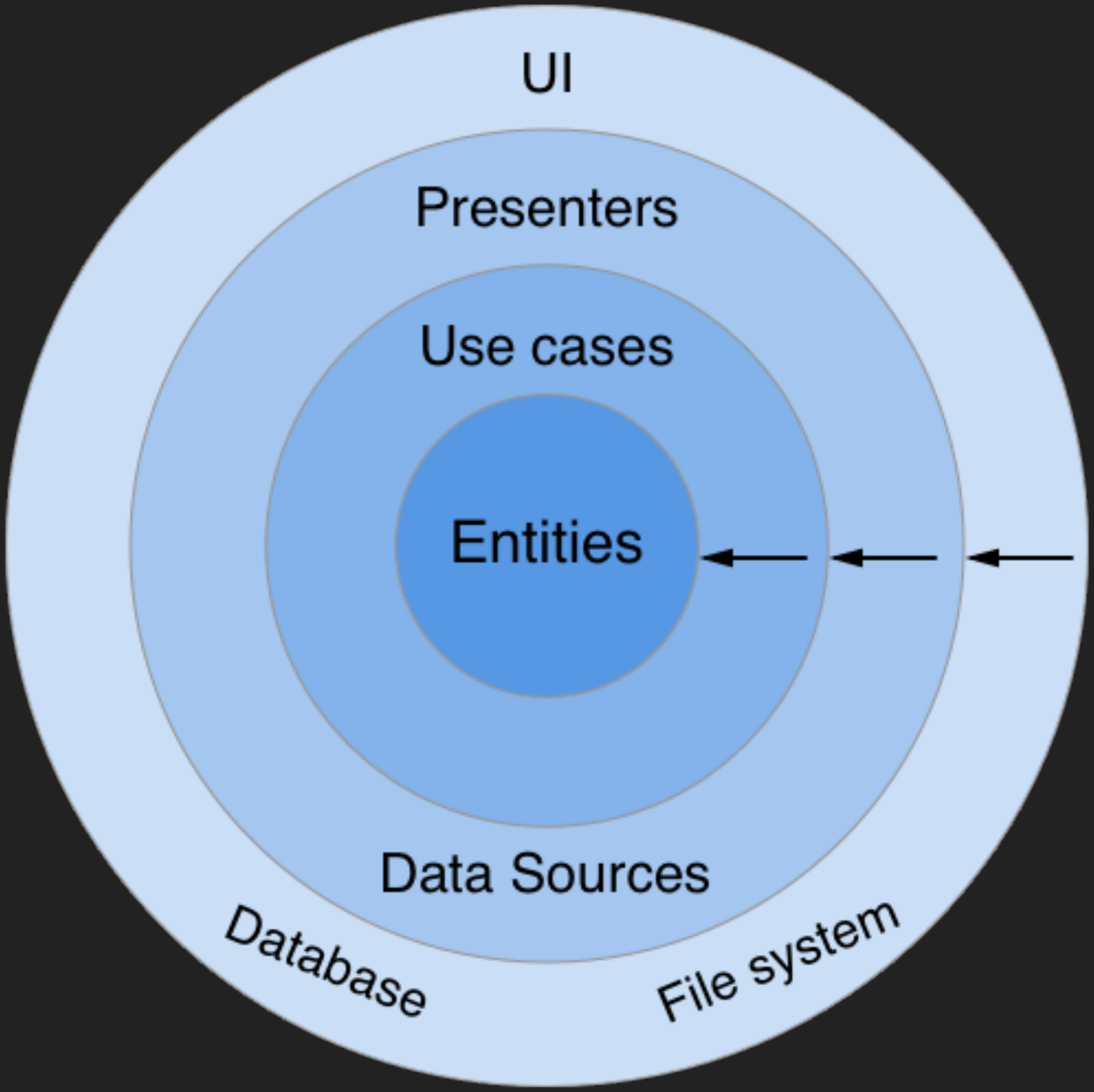
WORKSHOP ANDROID VI

CLEAN APP

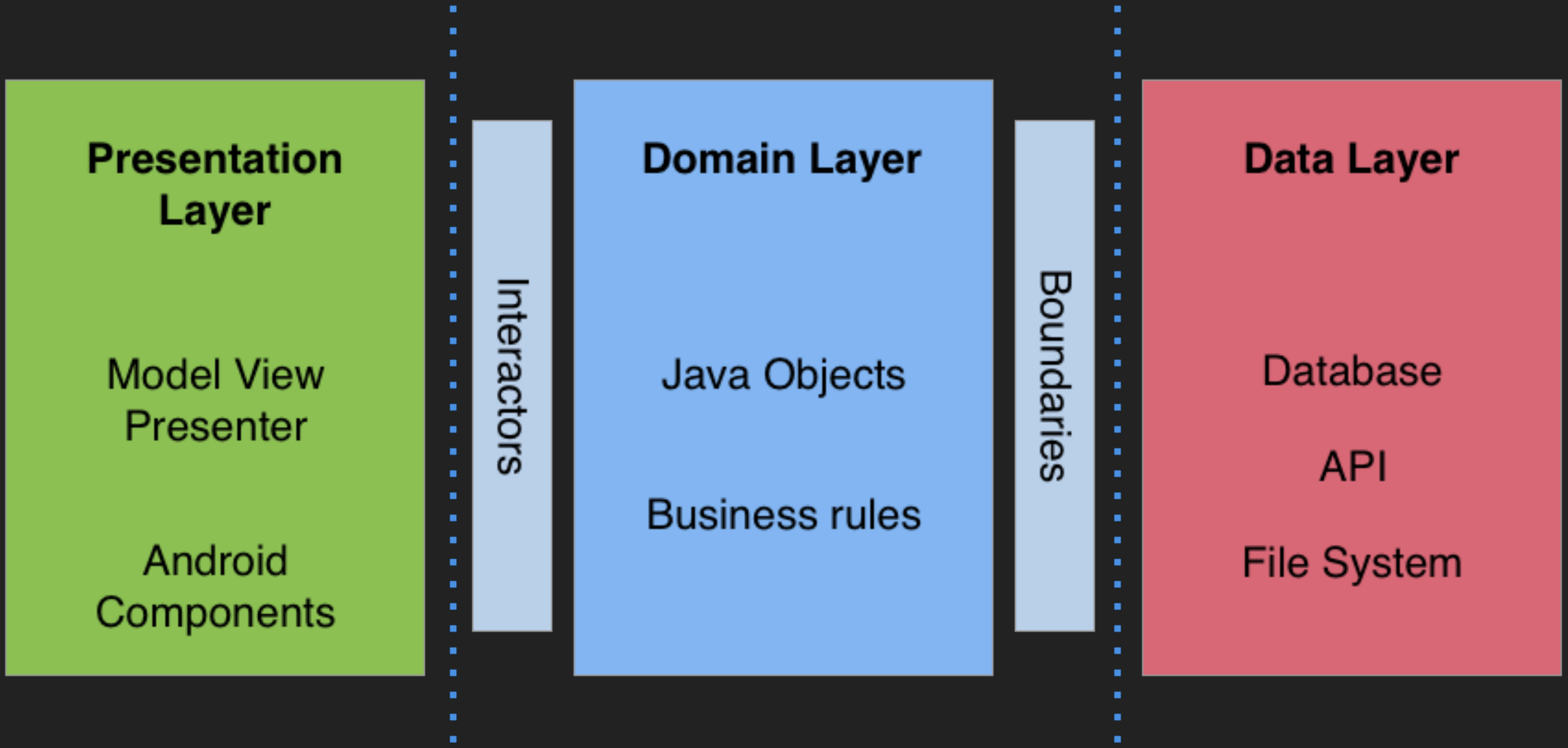
REMEMBER, REMEMBER

REMEMBER

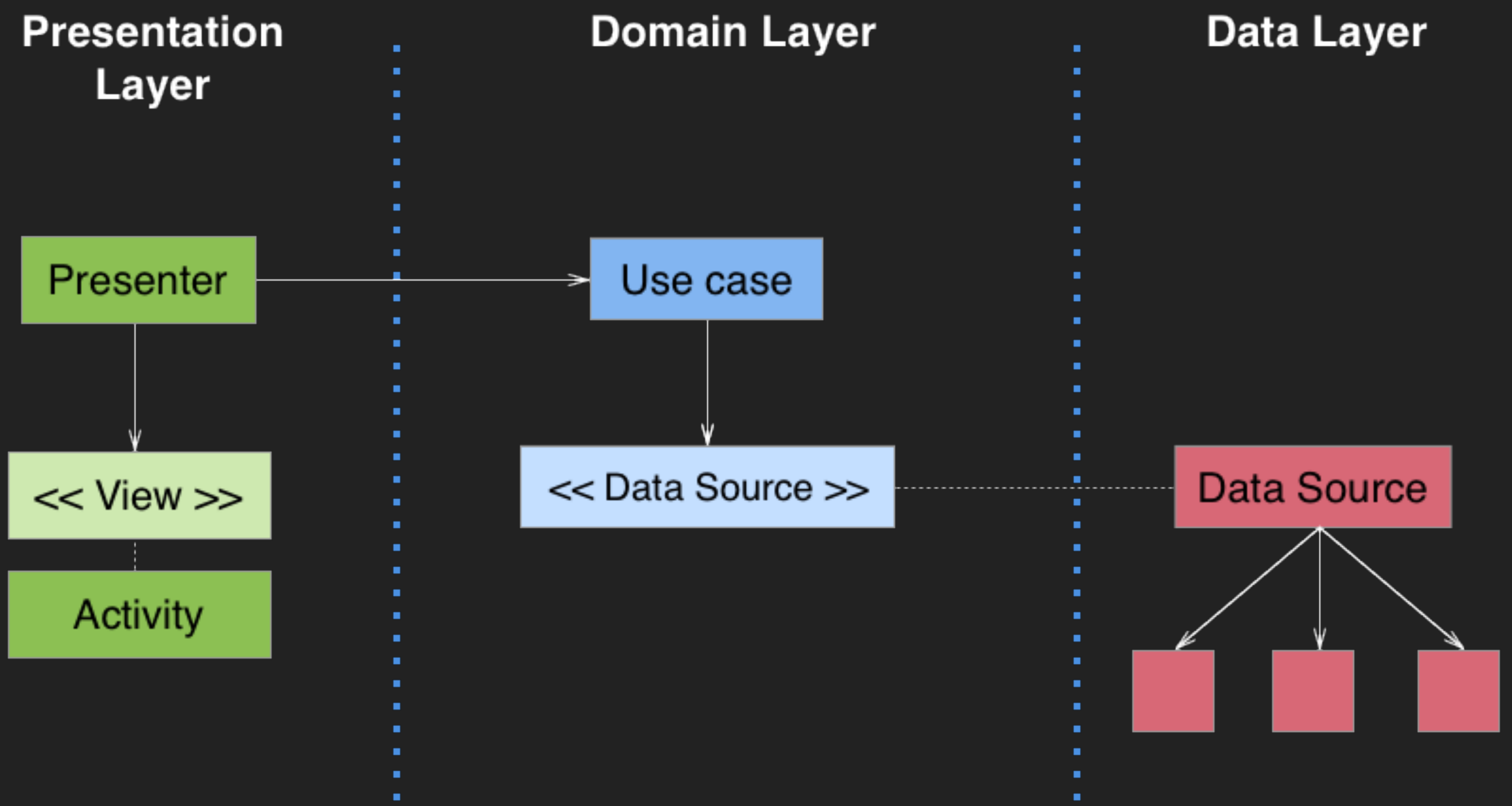
DEPENDENCY RULE

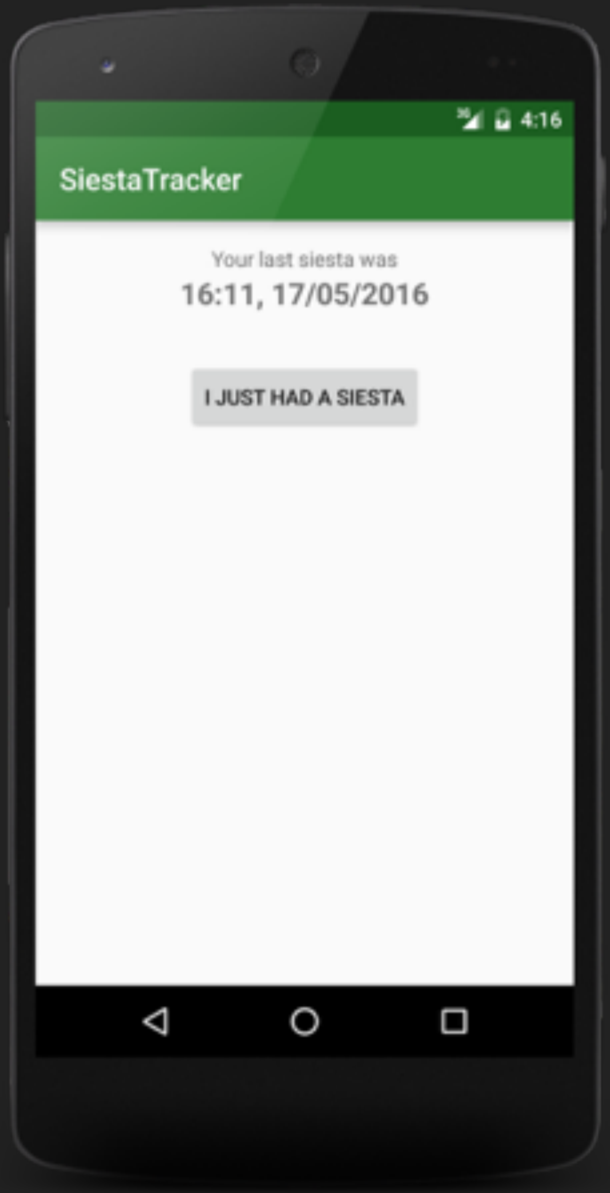


[OUR] IMPLEMENTATION



[OUR] IMPLEMENTATION





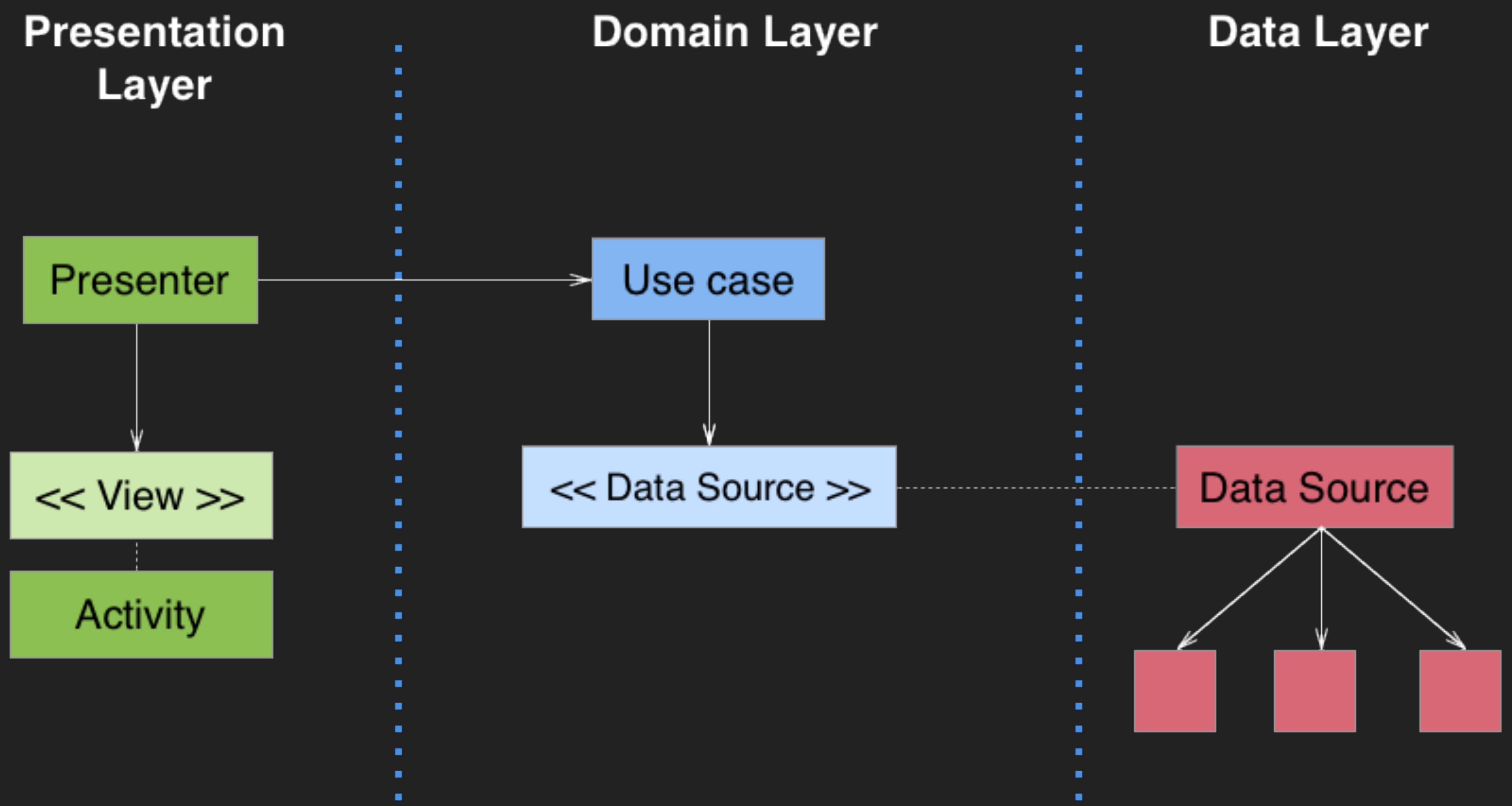
Your last siesta was
16:11, 17/05/2016

SIESTA TRACKER APP

FEATURES

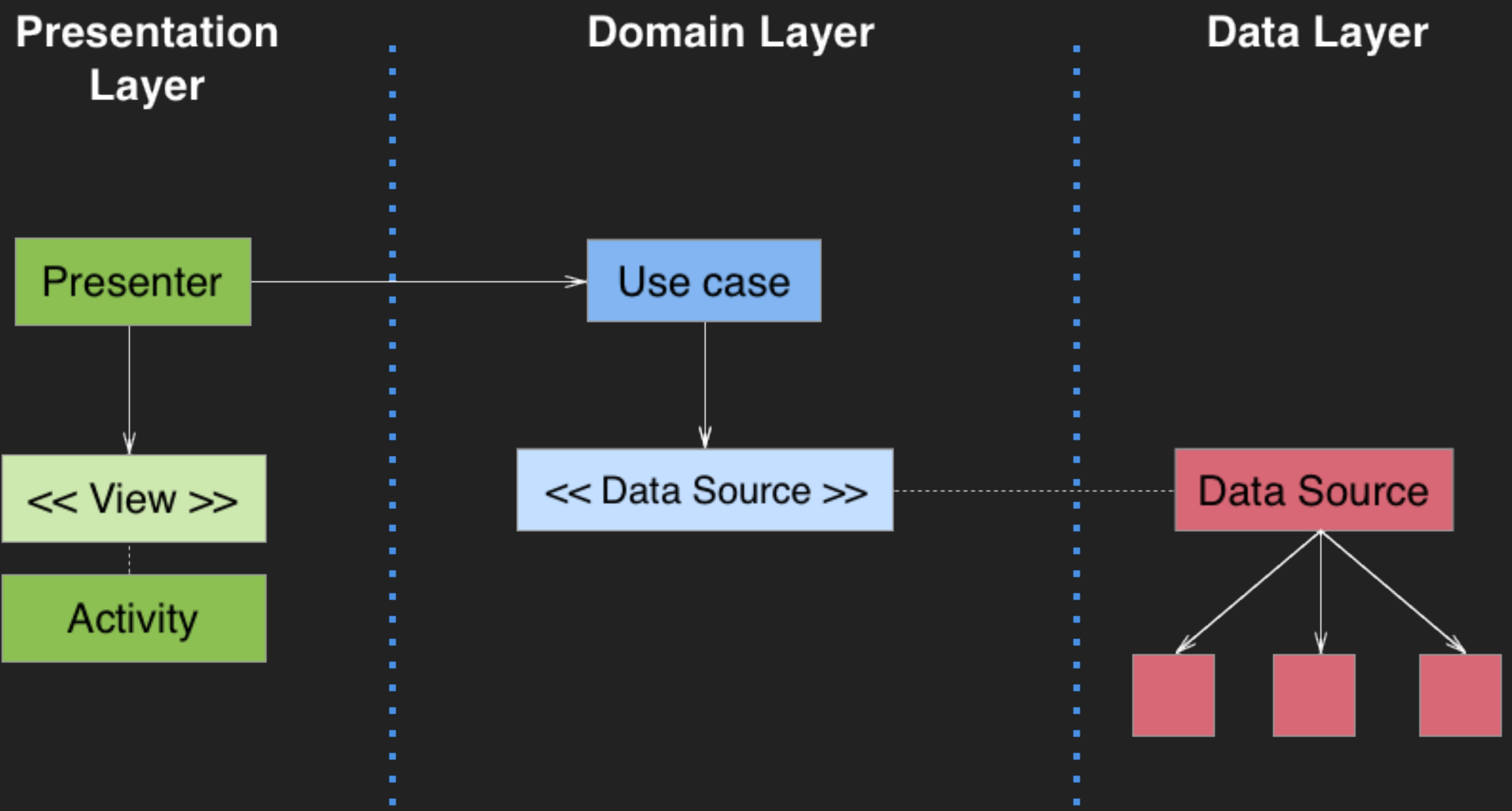
- 1. Shows the date and time of your last siesta
- 2. You can set a new siesta with a button

SKELETON


Presentation Layer
Domain Layer
Data Layer
Presenter
Use case
<< View >>
Activity
<< Data Source >>
Data Source

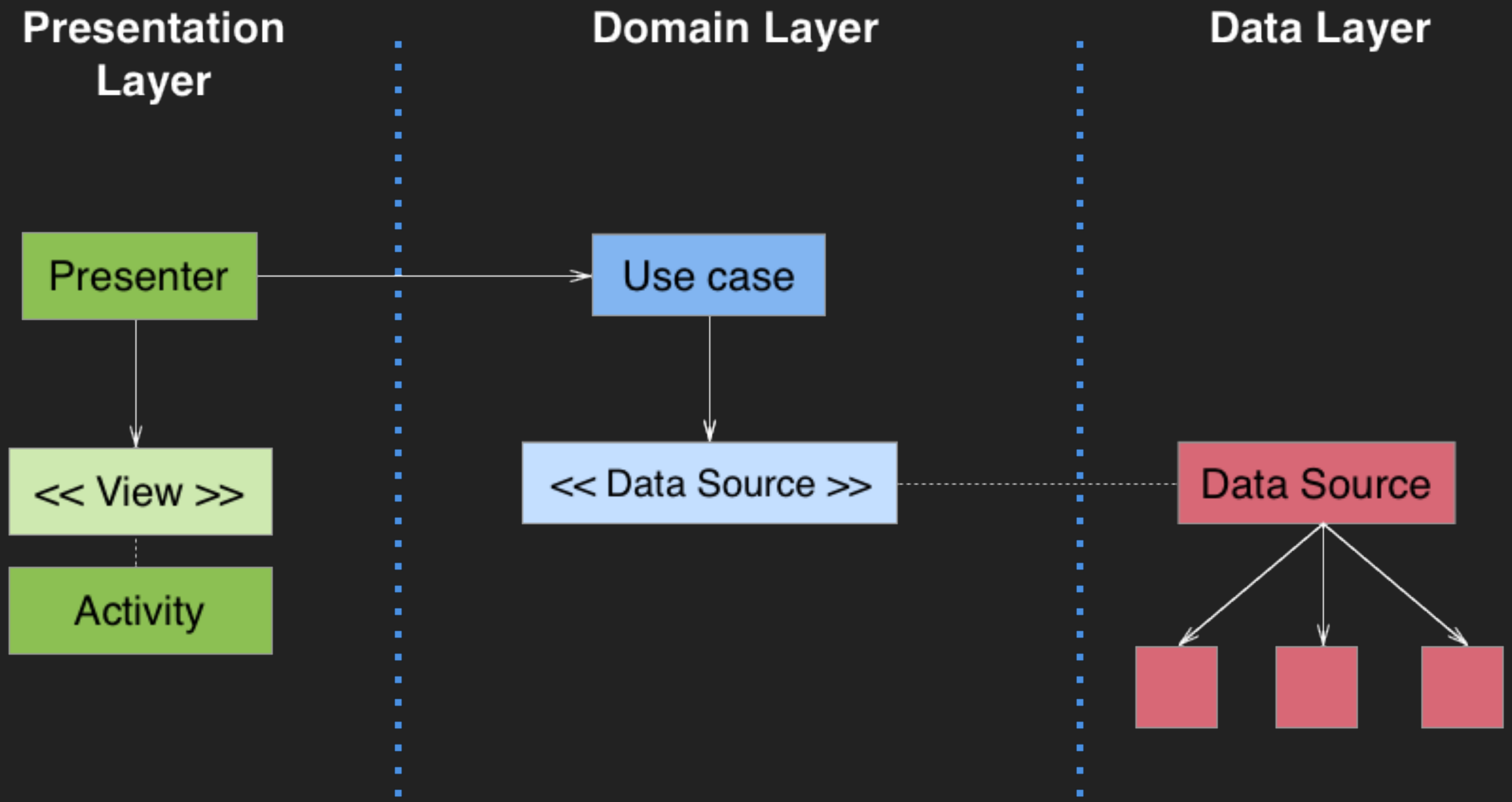
PRESENTER

- When received a date -> view shows the date as text



INTERACTOR

- When received a date -> returns that date
- When no date -> throw IllegalStateException

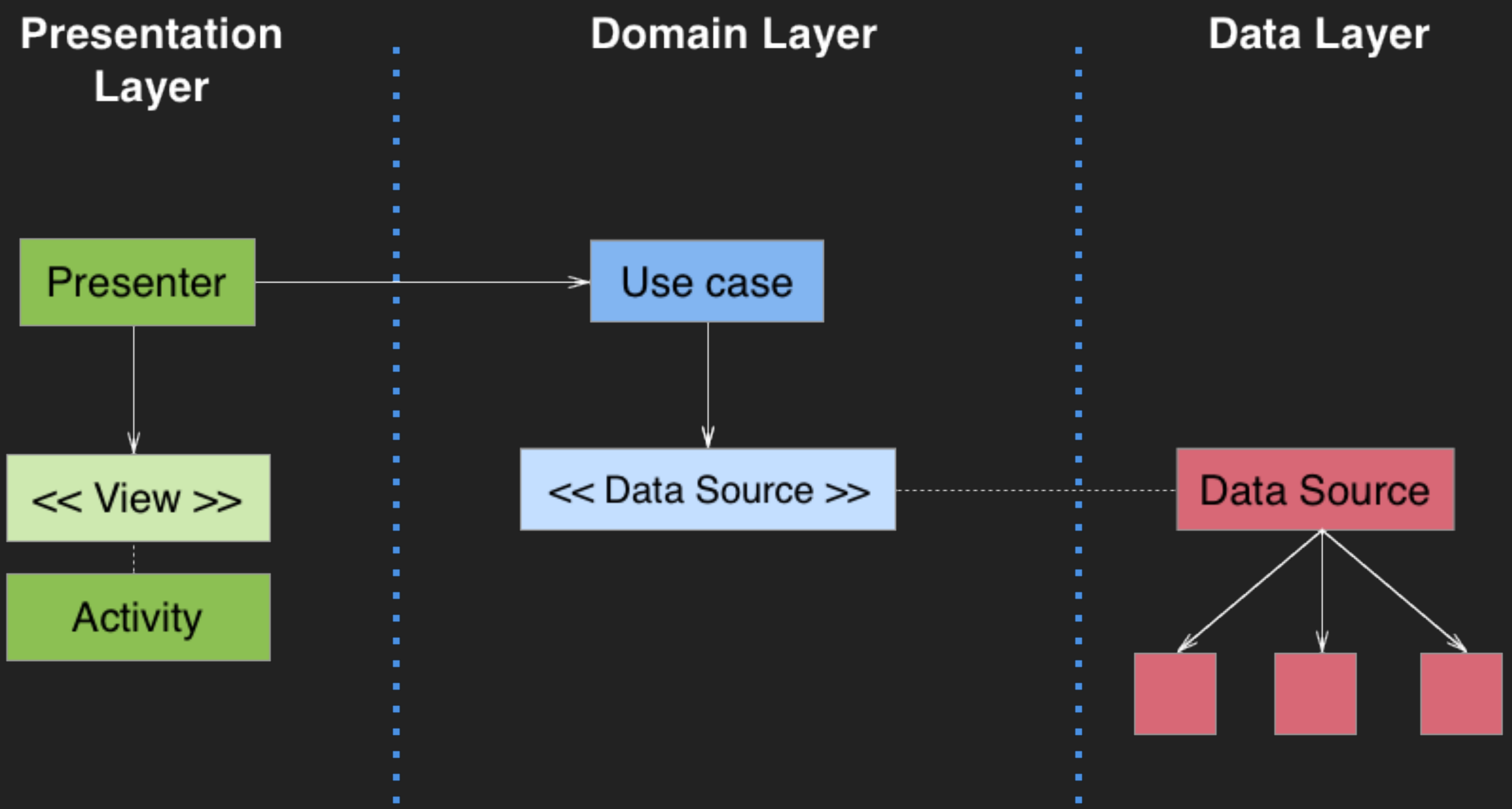


FEATURE 2:

SAVE NEW SIESTA

PRESENTER

- When update button click -> Save siesta date
- When update button click -> Show updated date



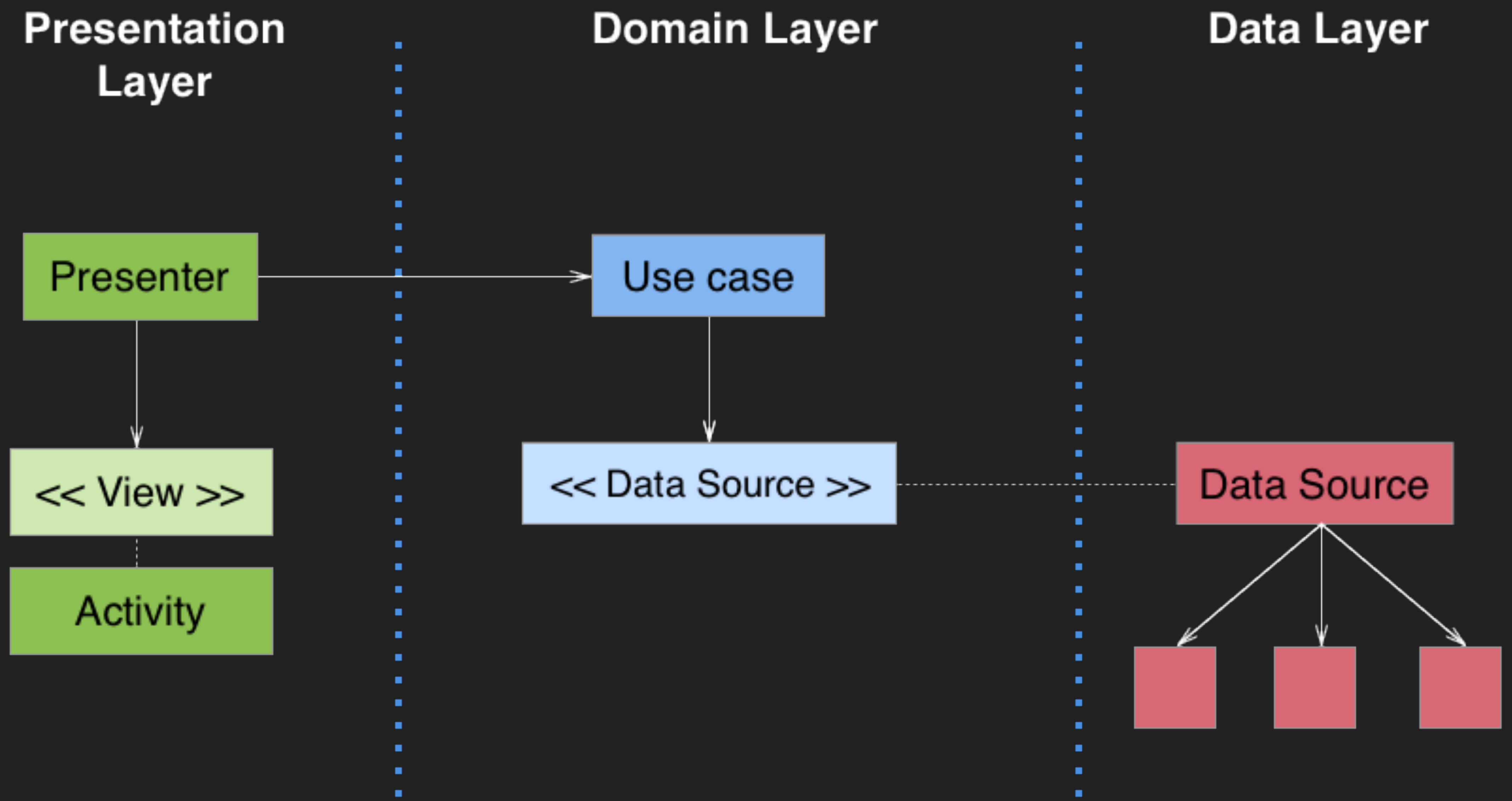
FEATURE 2

Do we need to pass the new date to the Interactor?

Question

INTERACTOR

- When save new siesta -> Store new date



NEW FEATURE:

PERSIST DATA

FEATURE 1

> Abstractions should not depend on details. Details should depend upon abstractions.

Dependency Inversion Principle - SOLID

FEATURE 1

Depend upon abstractions, not implementations

Dependency Inversion Principle - Rafa

NEW FEATURE:

DEFAULT DATE

DEFAULT DATE

BUSINESS RULE

- Show a message when no previous date available

INTERACTOR:

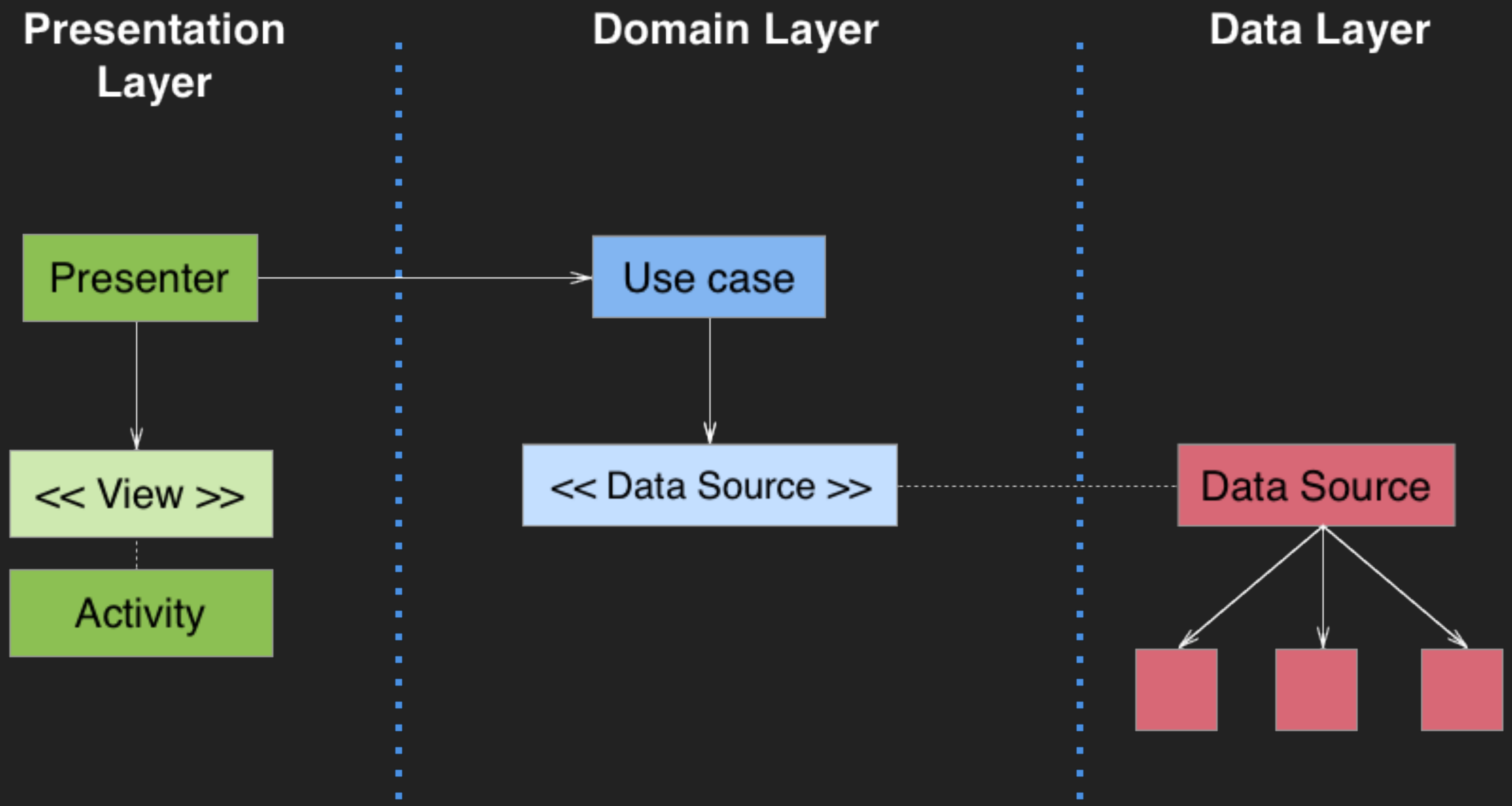
- When stored date is null -> Return null

PRESENTER:

- When date is null -> Show empty date message

DATASOURCE:

- When doesn’t contain date -> return null


Presentation Layer

Domain Layer

Data Layer

Presenter

<< View >>

Activity

Use case

<< Data Source >>

Data Source

CHALLENGE

CHALLENGE

- Feature that counts hours since last siesta
- Make code asynchronous with callbacks
- Add mappers between layers
- Don’t return null, use optional
- Wrap siesta date with our own model

THAT’S ALL, FOLKS!