Android Clean Architecture

Cauê Ferreira
@CaueFerreira

Cauê Ferreira

Developer

@CaueFerreira

caueferreira

moip

elo7

doare

+100k clientes
+300 parceiros integrados
+1 bi/ ano
+200 colaboradores
+ 8 anos de operação

AssinaMe

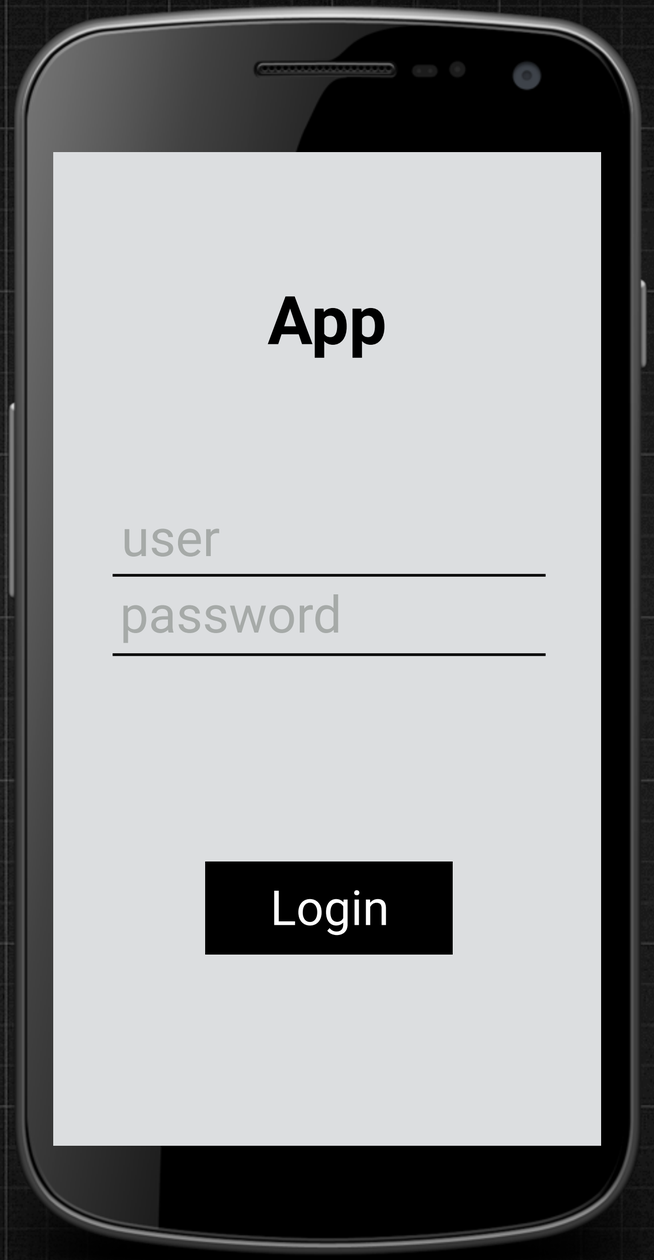
PORTA DOS FUNDOS

CLUBE DO MALTE

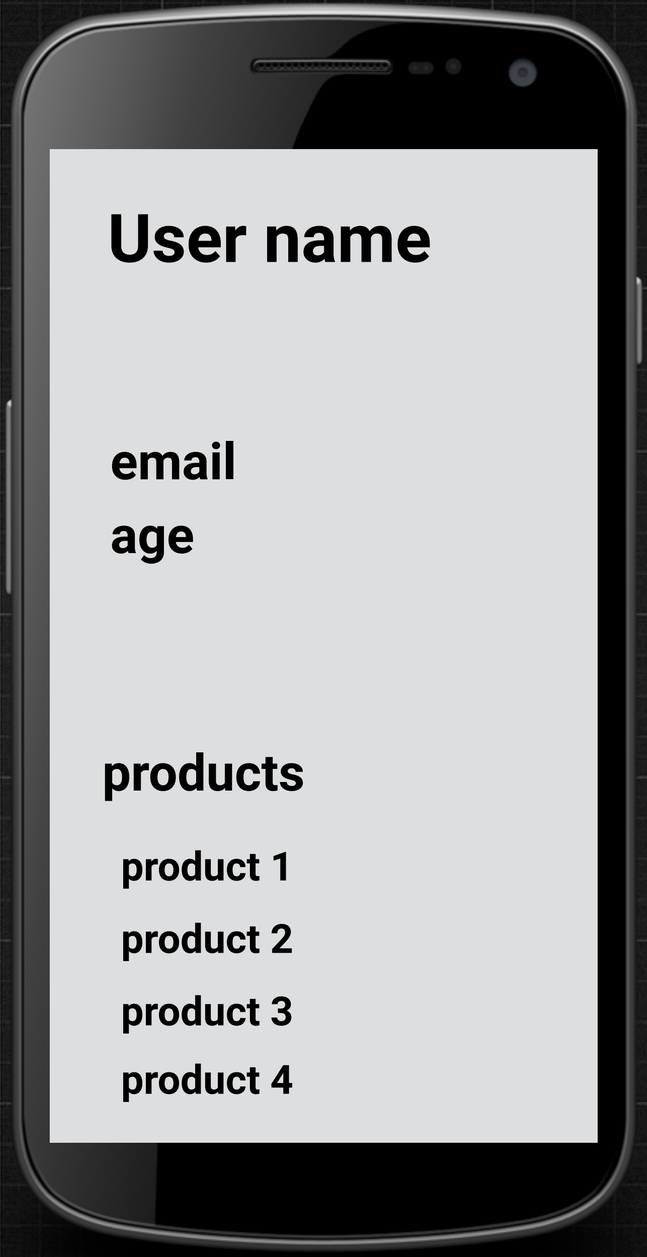
estante virtual

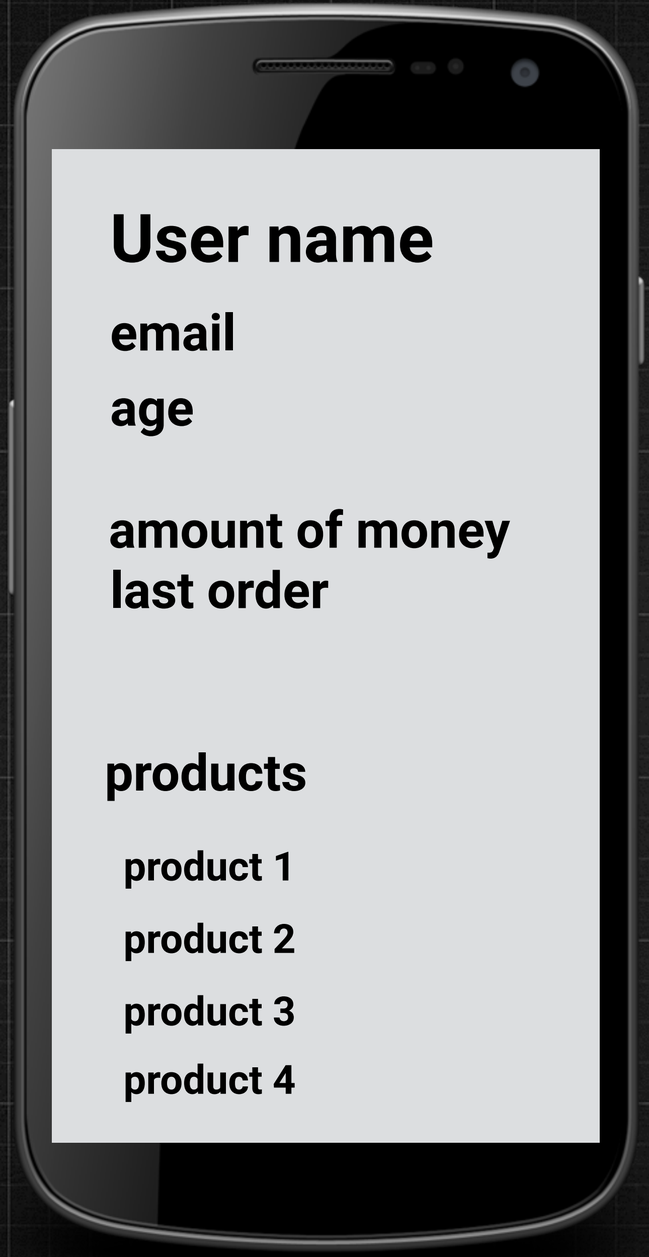
natura

Ipiranga


App
user
password
Login

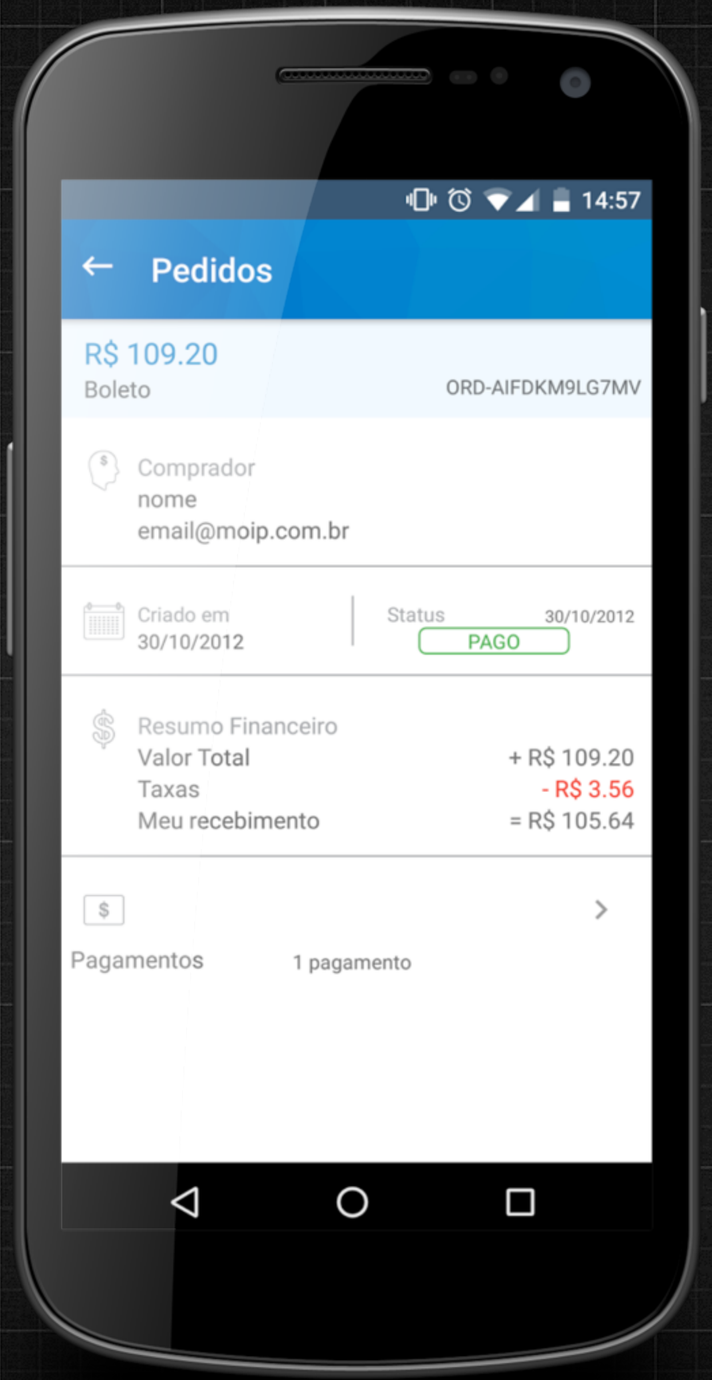

User name
email
age


User name
email
age
products
product 1
product 2
product 3
product 4


User name
email
age
amount of money
last order
products
product 1
product 2
product 3
product 4


14:42
moip
Saldo Disponível
R$ 92,188.88
Performance
MAR/ 16
comparado com
MAR/ 15
dia 21
R$ 0.00
R$ 4,447.00
R$ 10K
R$ 8K
R$ 6K
R$ 4K
R$ 2K
R$ 0
1
5
9
13
17
21
25
29
Vendas
74
74
Ticket Médio
R$ 107.07
R$ 52.04
Acumulado
R$ 3,054.96
R$ 3,054.96


5:06
Semanal
Mensal
Anual
90
Pedidos
R$ 136.56
Ticket médio
R$ 6,152.11
Acumulativo
R$ 13,571.80
R$ 0.00


14:57
Pedidos
R$ 109.20
Boleto
ORD-AIFDKM9LG7MV
Comprador
nome
email@moip.com.br
Criado em
30/10/2012
Status
30/10/2012
PAGO
Resumo Financeiro
Valor Total
+ R$ 109.20
Taxas
- R$ 3.56
Meu recebimento
= R$ 105.64
Pagamentos
1 pagamento


5:07
victor@idea.me
01/11/2012
Valor da cobrança
R$ 10.00
Taxa Moip
R$ 0.68
Valor recebido
R$ 9.32
Data de recebimento
01/11/2012

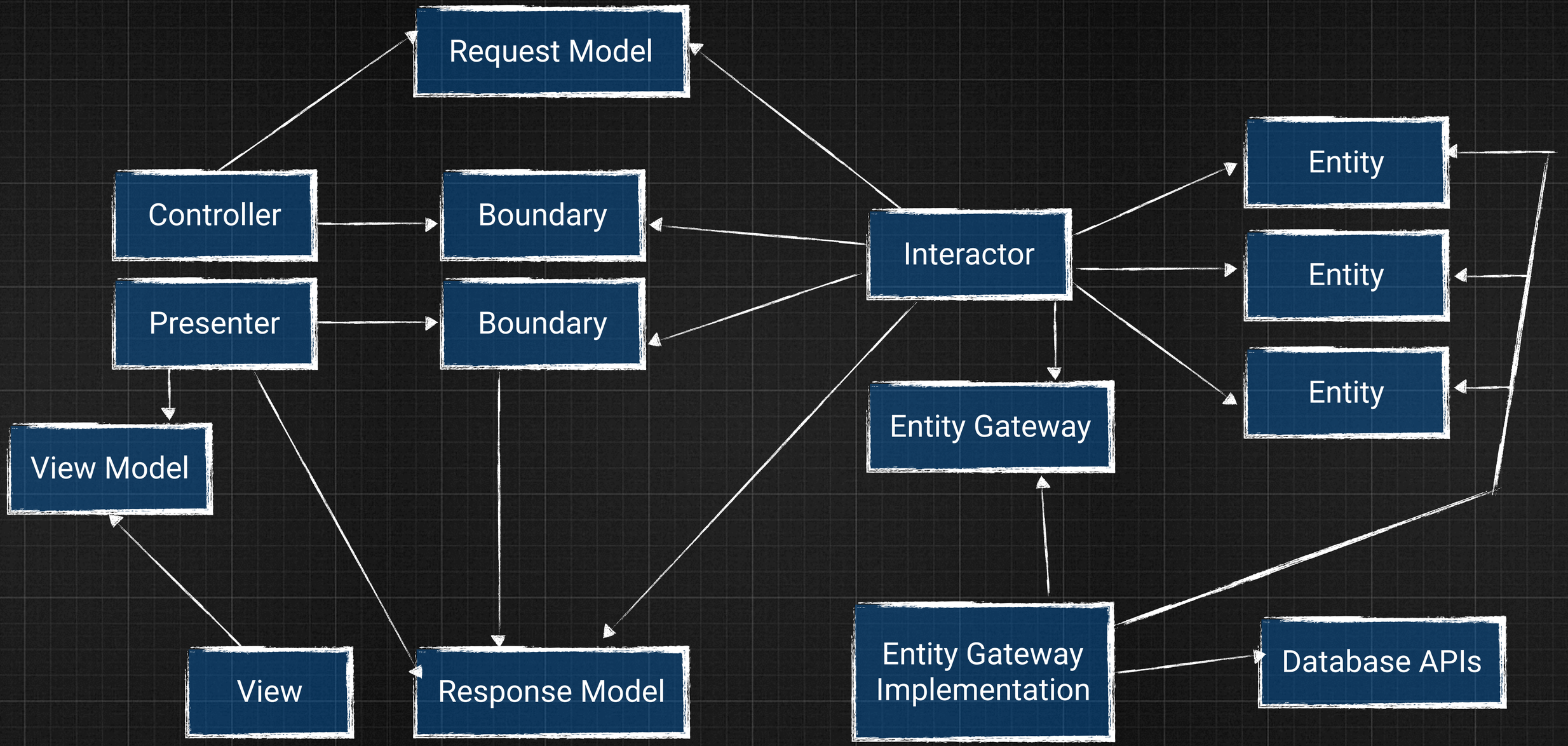

14:43
Transferência
16917 transferências totalizando R$ 2.2M
Nenhum filtro selecionado
R$ 11.99
conta Moip
integracao@labs.moip.com.br
18/04/2016
Completo
R$ 11.99
conta Moip
integracao@labs.moip.com.br
18/04/2016
Completo
R$ 11.99
conta Moip
integracao@labs.moip.com.br
17/04/2016
Completo
R$ 11.99
conta Moip
integracao@labs.moip.com.br
17/04/2016
Completo
R$ 11.99
conta Moip
integracao@labs.moip.com.br
17/04/2016
Completo
R$ 11.99
conta Moip
integracao@labs.moip.com.br
17/04
Comp


5:07
R$ 11.99
Conta Moip
integracao@labs.moip.com.br
04/09/2016 15:33:34
AUTORIZADO
R$ -11.99
Conta Moip
nxaqu6pjaw@labs.moip.com.br
04/09/2016 15:33:29
AUTORIZADO
R$ -11.99
Conta Moip
integracao-transparente@labs.m
04/09/2016 15:33:26
AUTORIZADO
R$ -5.00
BANCO BRADESCO S.A.
Ag: 12345-0 | CC: 12345678
04/09/2016 14:27:30
AUTORIZADO
R$ -5.00
BANCO BRADESCO S.A.
Ag: 12345-0 | CC: 12345678
04/09/2016 14:27:21
AUTOR

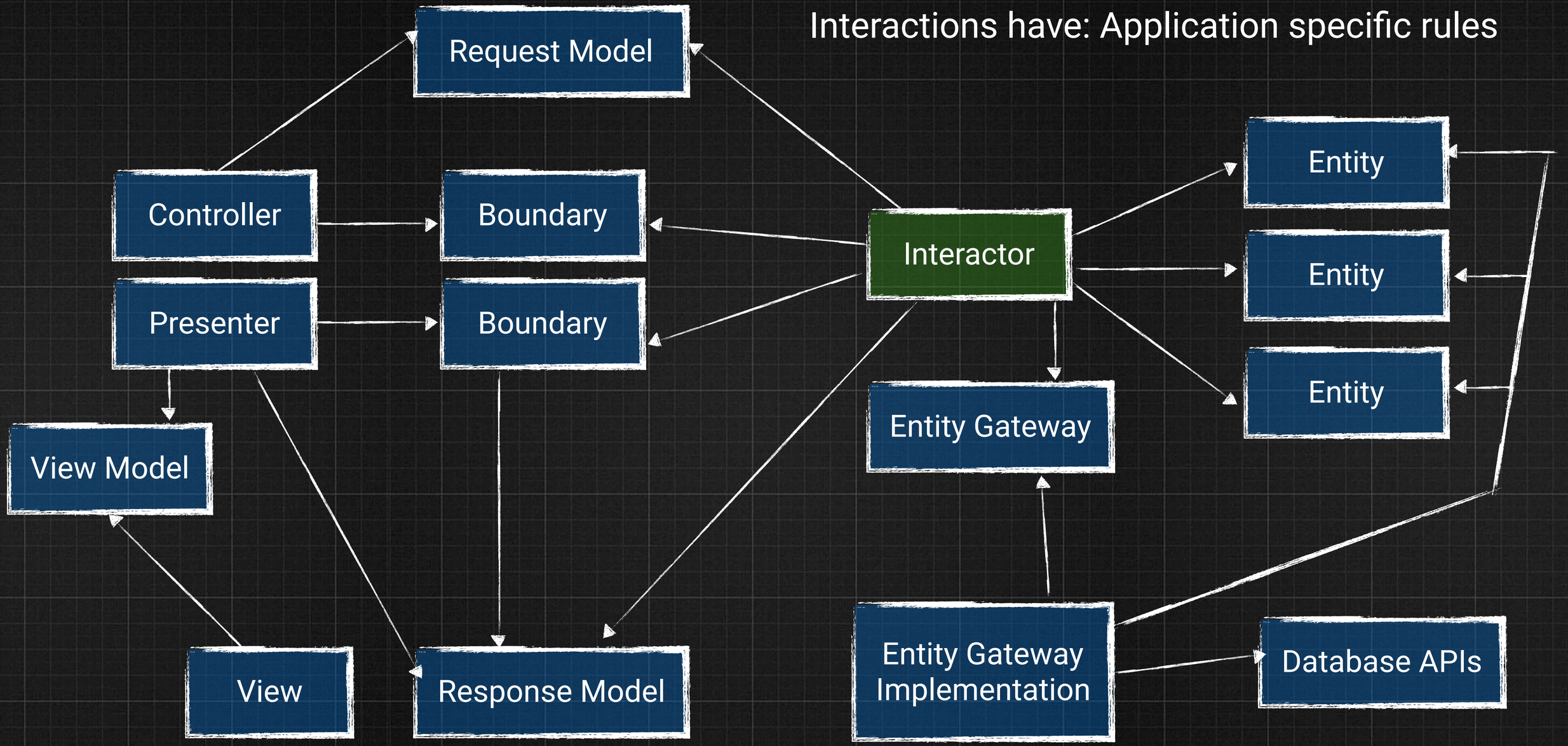
```
MoipAccountManager.retrieveOauth(getActivity(), externalId -> {
        loginProgressDialog = setUpProgressDialog(getActivity());
        loginProgressDialog.show();

        AccountApi.getAccount(MoipSharedPreferences.getInstance().getExternalId(), account -> {
            MoipSharedPreferences.getInstance().saveAccount(account);
            ContaMoipApplication.getInstance().identifyPerson(account);
            toDashboard();
            loginProgressDialog.hide();

        }, error -> {
            loginProgressDialog.hide();
            MoipSharedPreferences.getInstance().clear();
        });

    }, () -> {
        MoipSharedPreferences.getInstance().clear();
    });
```

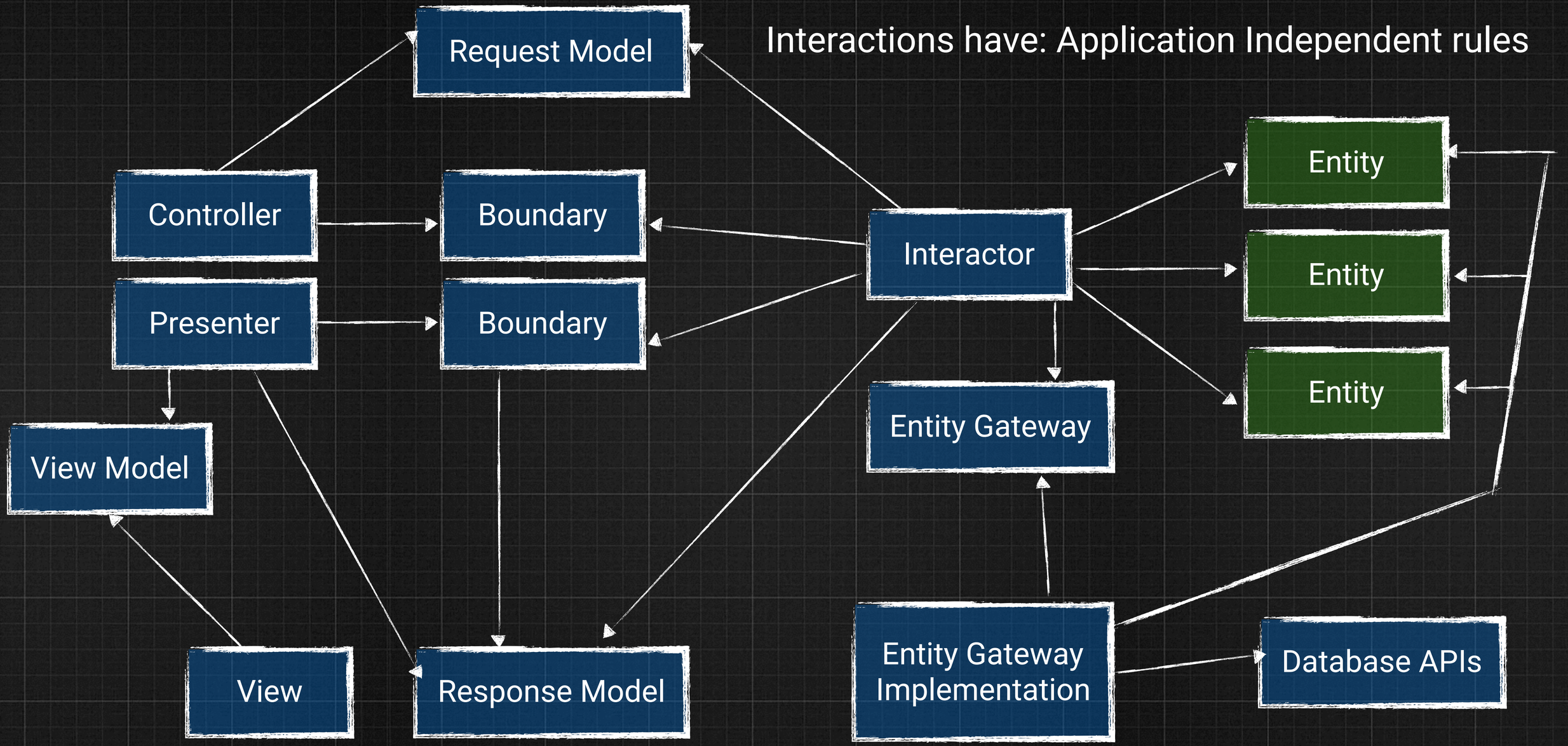
```
private void toggleViews(final RelativeLayout parent, final boolean enable) {
        parent.setEnabled(enable);
        for (int i = 0; i < parent.getChildCount(); i++) {
            View child = parent.getChildAt(i);
            if (child instanceof ViewGroup) {
            } else {
                child.setEnabled(enable);
                if (child.isClickable())
                    child.setClickable(enable);
            }
        }
    }
```

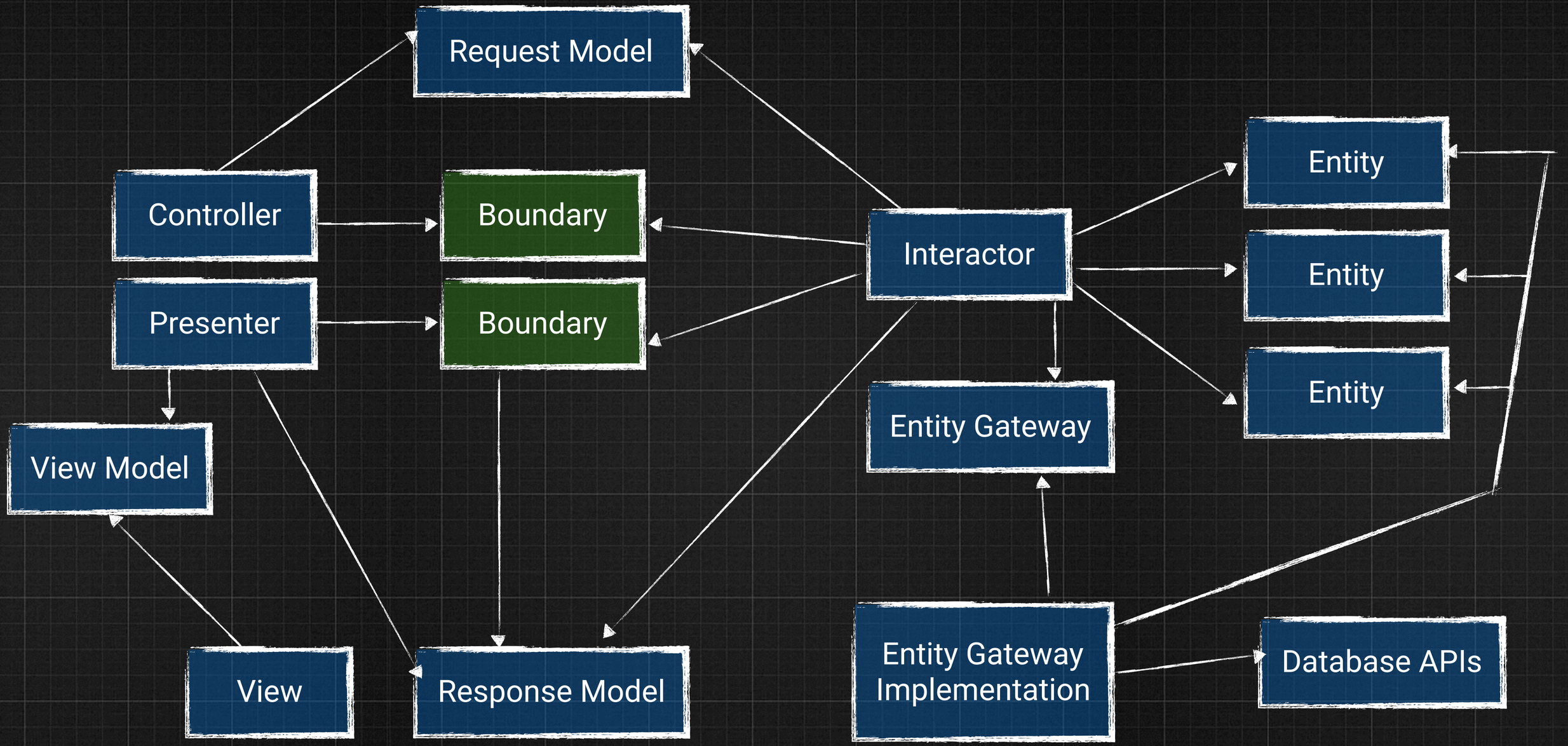
```
private void loadData() {
        String fakeEmail = "";

      if (order.getCustomer().getEmail() != null) { ContaMoipApplication.getInstance().loadGravatar(order.getCustomer().getEmail(), profileImage);
          txtEmail.setText(order.getCustomer().getEmail());
      } else {
          ContaMoipApplication.getInstance().loadGravatar(fakeEmail, profileImage);
          txtEmail.setText(fakeEmail);
      }

      txtData.setText(Date.parse(order.getCreatedAt(), Date.BRAZILIAN_DATE));
      txtDetail.setText(" ");
      txtOrderValue.setTextWithCleanAmount(order.getAmount().getTotal());
      txtMoipTax.setTextWithCleanAmount((long) order.getAmount().getFees());
      txtLiquidValue.setTextWithCleanAmount(order.getAmount().getTotal() - order.getAmount().getFees());
      txtDueDate.setText(Date.parse(order.getUpdatedAt(), Date.BRAZILIAN_DATE));
    }

    private int getLayoutStatusColor(final OrderStatus status) {
        switch (status) {
            case PAID:
                return getActivity().getResources().getColor(R.color.paid_order);
            case REVERTED:
                return getActivity().getResources().getColor(R.color.refunded_order);
            case NOT_PAID:
                return getActivity().getResources().getColor(R.color.not_paid_order);
            default:
                return getActivity().getResources().getColor(R.color.other_order_status);
        }
    }
```

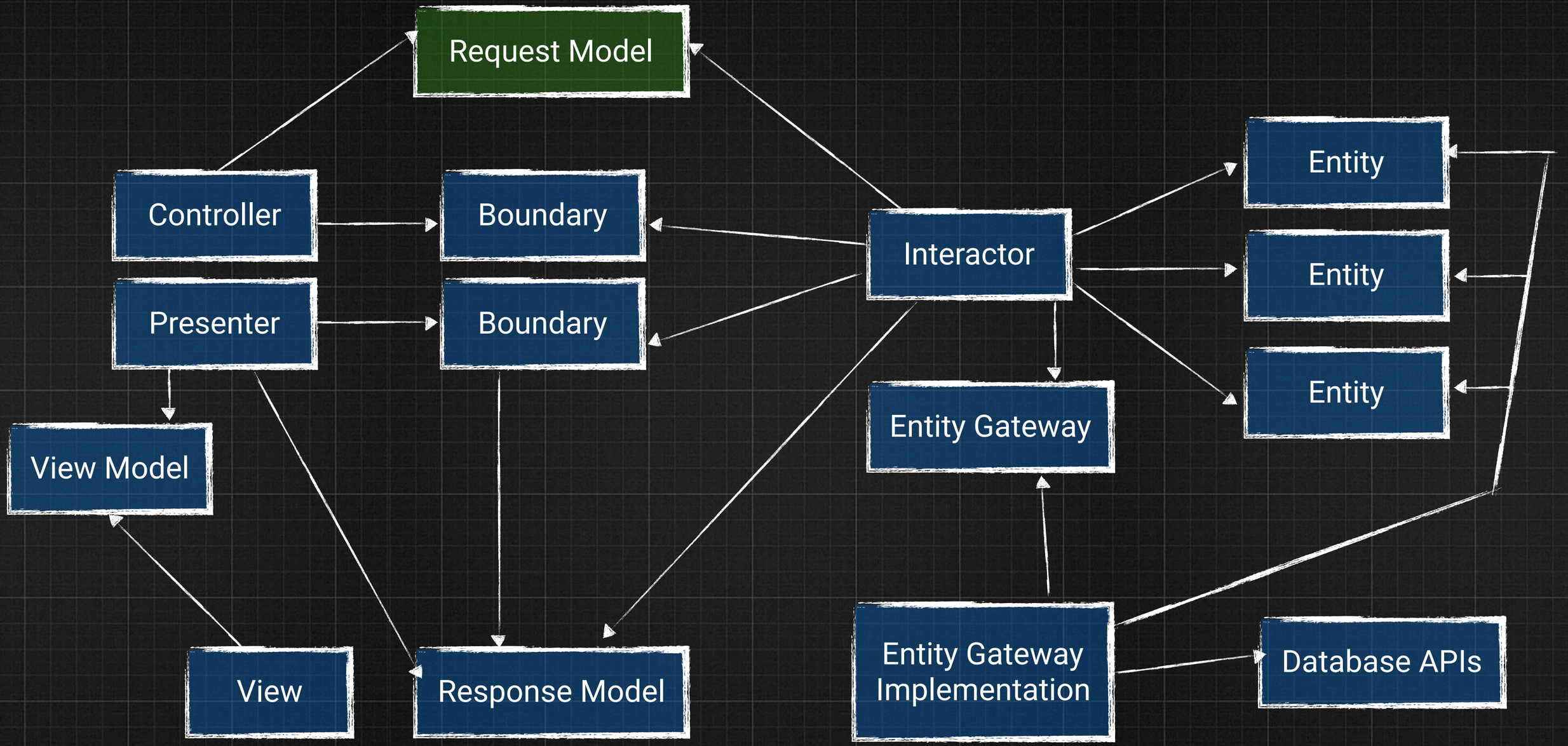

Request Model
Controller
Boundary
Interactor
Entity
Entity
Presenter
Boundary
Entity
Entity Gateway
View Model
View
Response Model
Entity Gateway Implementation
Database APIs

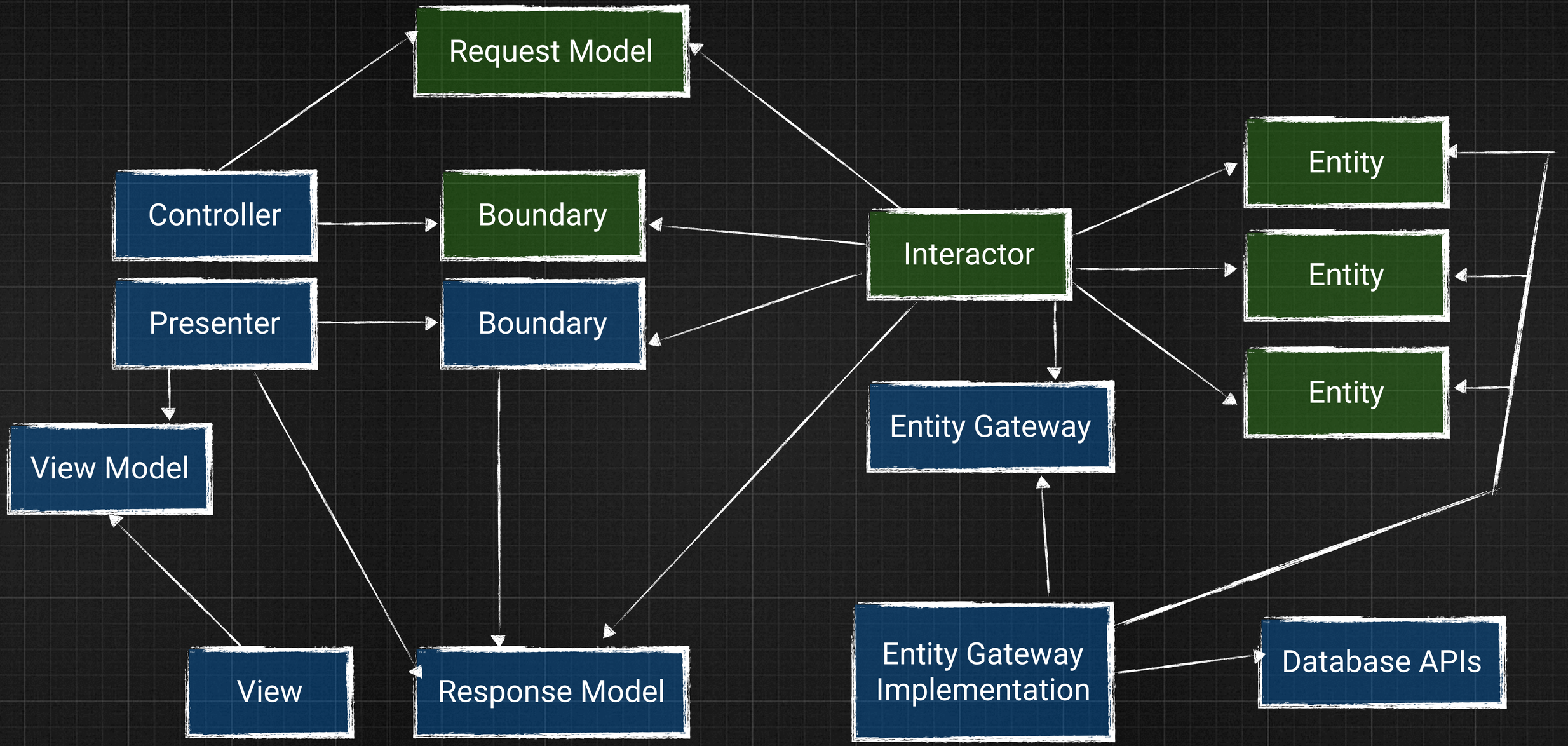
Interactions have: Application specific rules

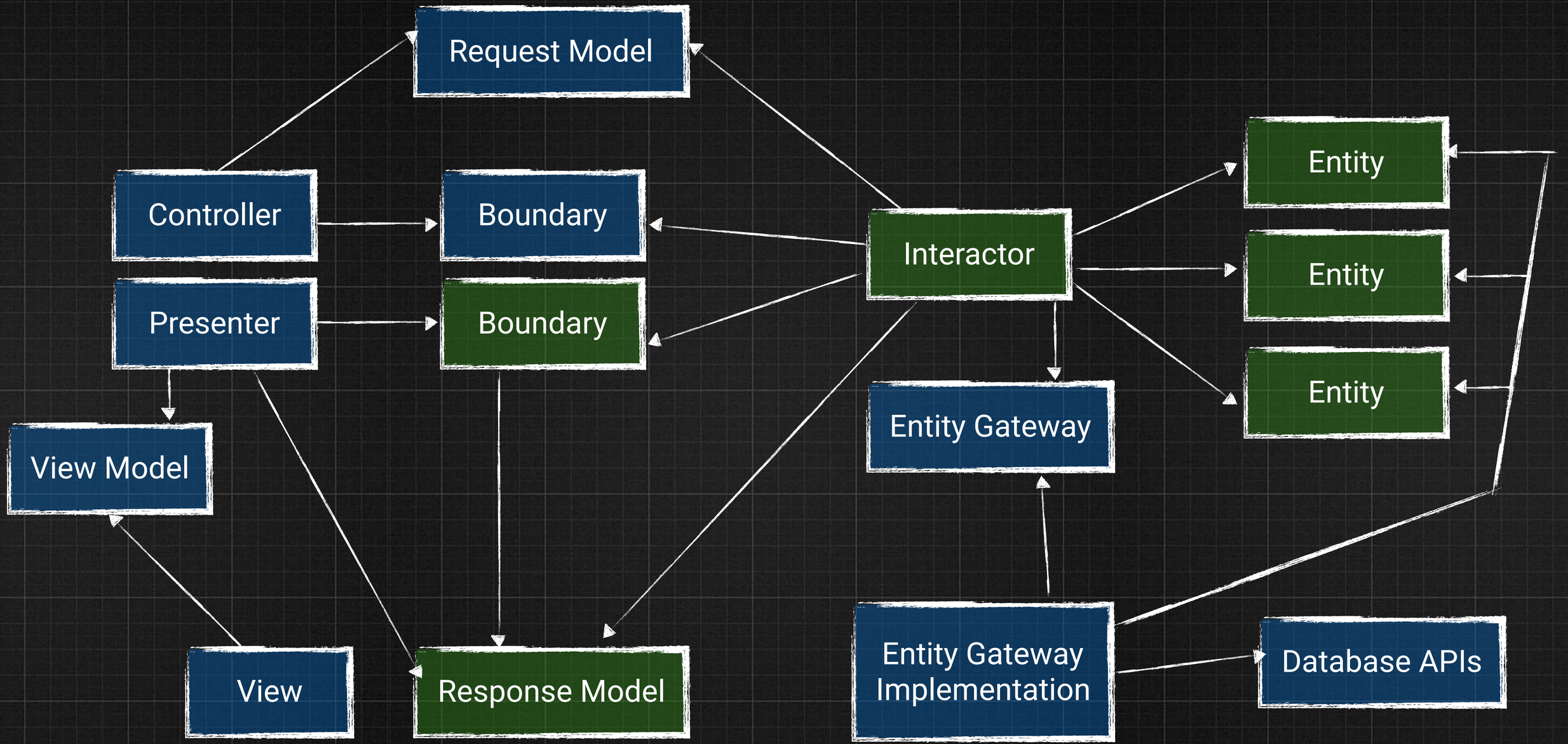



Interactions have: Application Independent rules
Request Model
Controller
Boundary
Interactor
Entity
Entity
Entity
Presenter
Boundary
Entity Gateway
View Model
View
Response Model
Entity Gateway Implementation
Database APIs

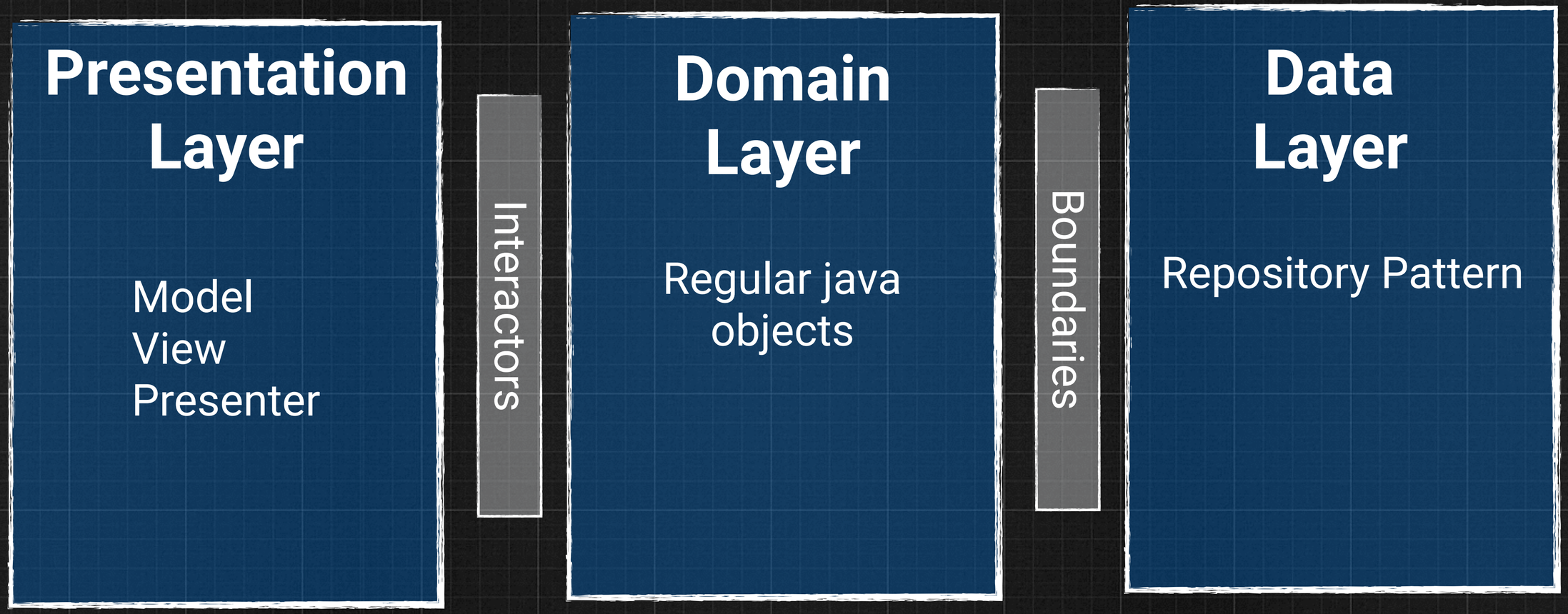

Request Model
Controller
Boundary
Interactor
Entity
Presenter
Boundary
Entity
Entity
View Model
Entity Gateway
View
Response Model
Entity Gateway Implementation
Database APIs

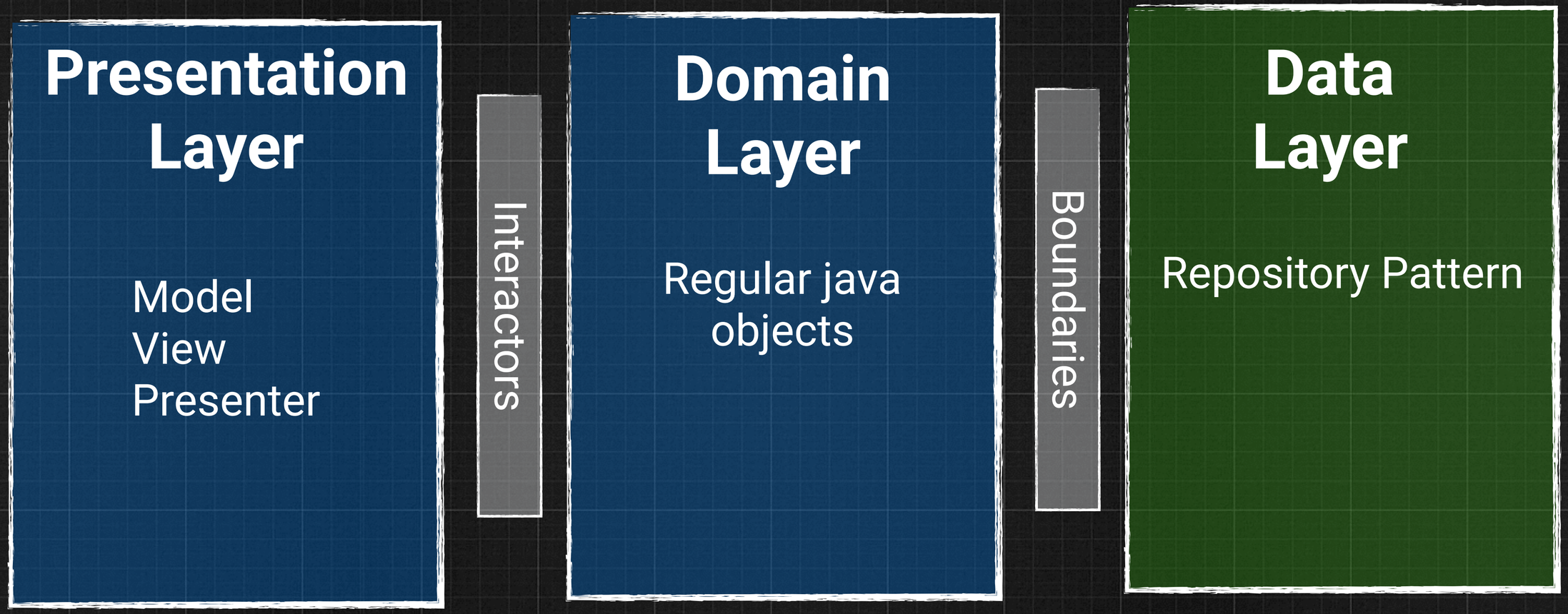

Request Model
Controller
Boundary
Interactor
Entity
Presenter
Boundary
Entity
Entity
View Model
Entity Gateway
View
Response Model
Entity Gateway Implementation
Database APIs

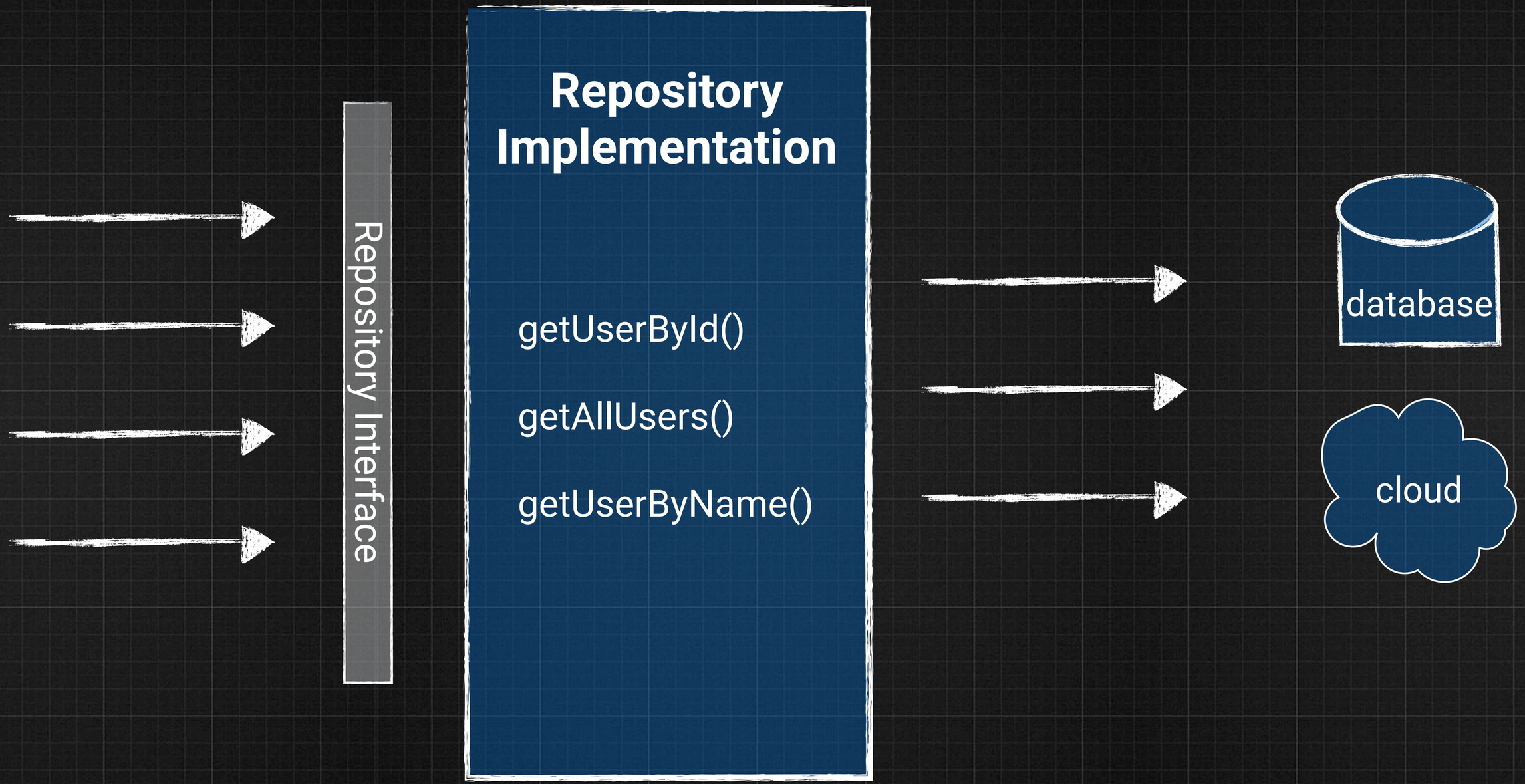

Request Model
Controller
Boundary
Interactor
Entity
Entity
Entity
Presenter
Boundary
Entity Gateway
View Model
View
Response Model
Entity Gateway Implementation
Database APIs



The principles of clean architecture


Presentation Layer
Model
View
Presenter
Interactors
Domain Layer
Regular java objects
Boundaries
Data Layer
Repository Pattern


Presentation Layer
Model
View
Presenter
Interactors
Domain Layer
Regular java objects
Boundaries
Data Layer
Repository Pattern

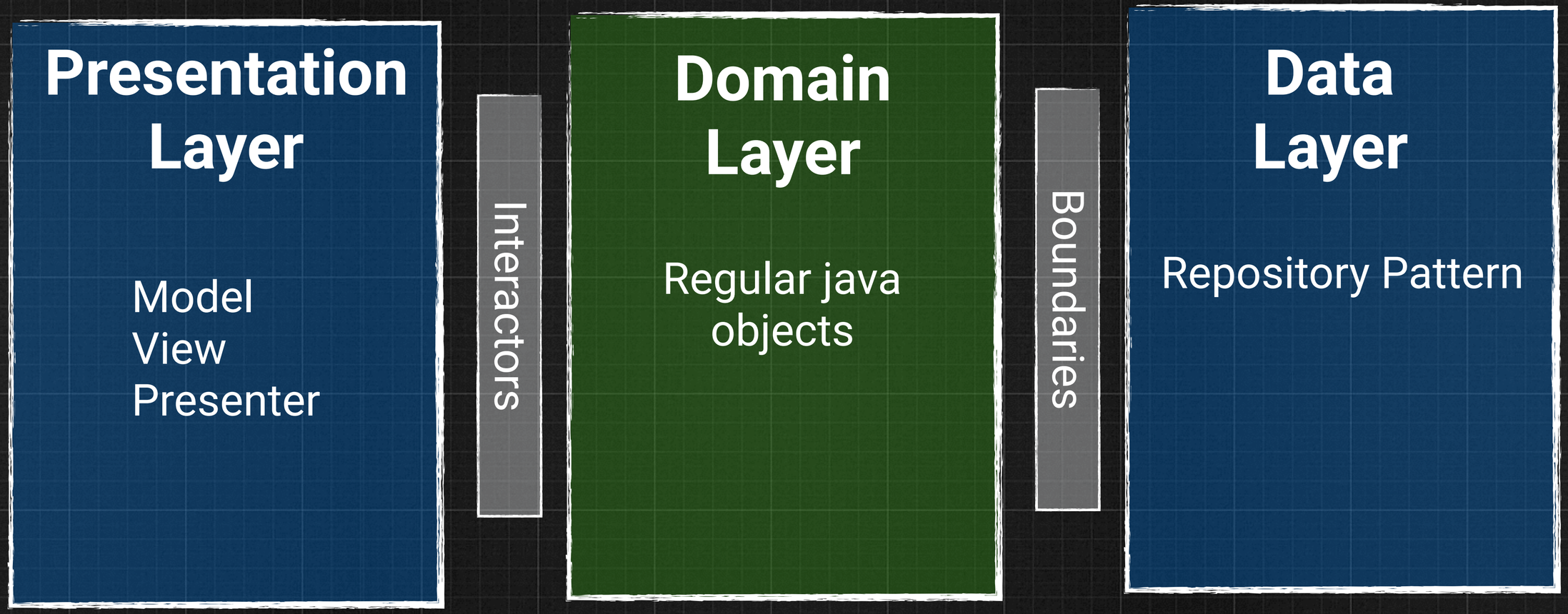

Repository Interface
Repository Implementation
getUserById()
getAllUsers()
getUserByName()
database
cloud

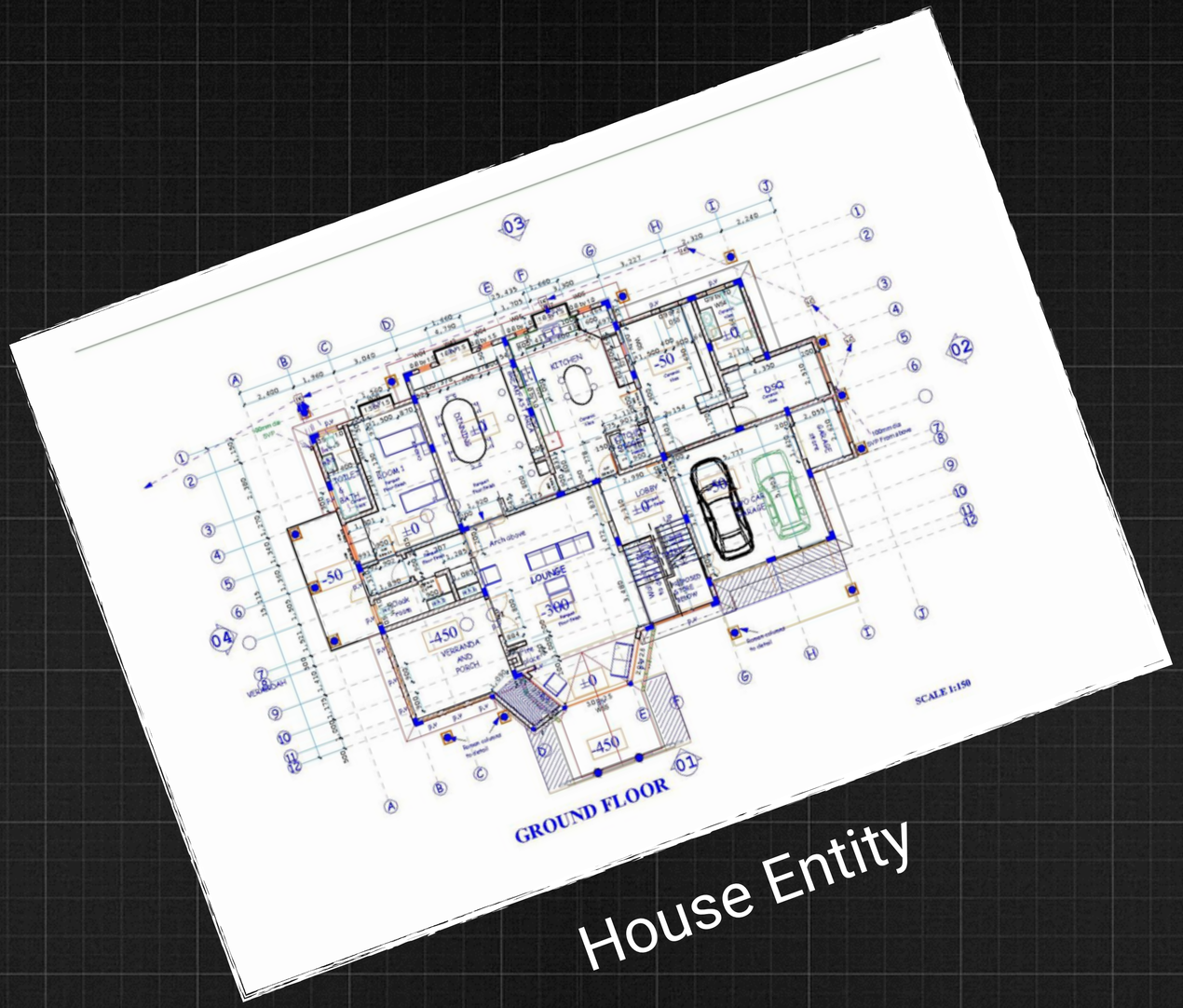
A glimpse about Repository Pattern

* It contains some business rules
* Should be named after an object that holds its content

moneyRepository
documentsRepository
cardsRepository

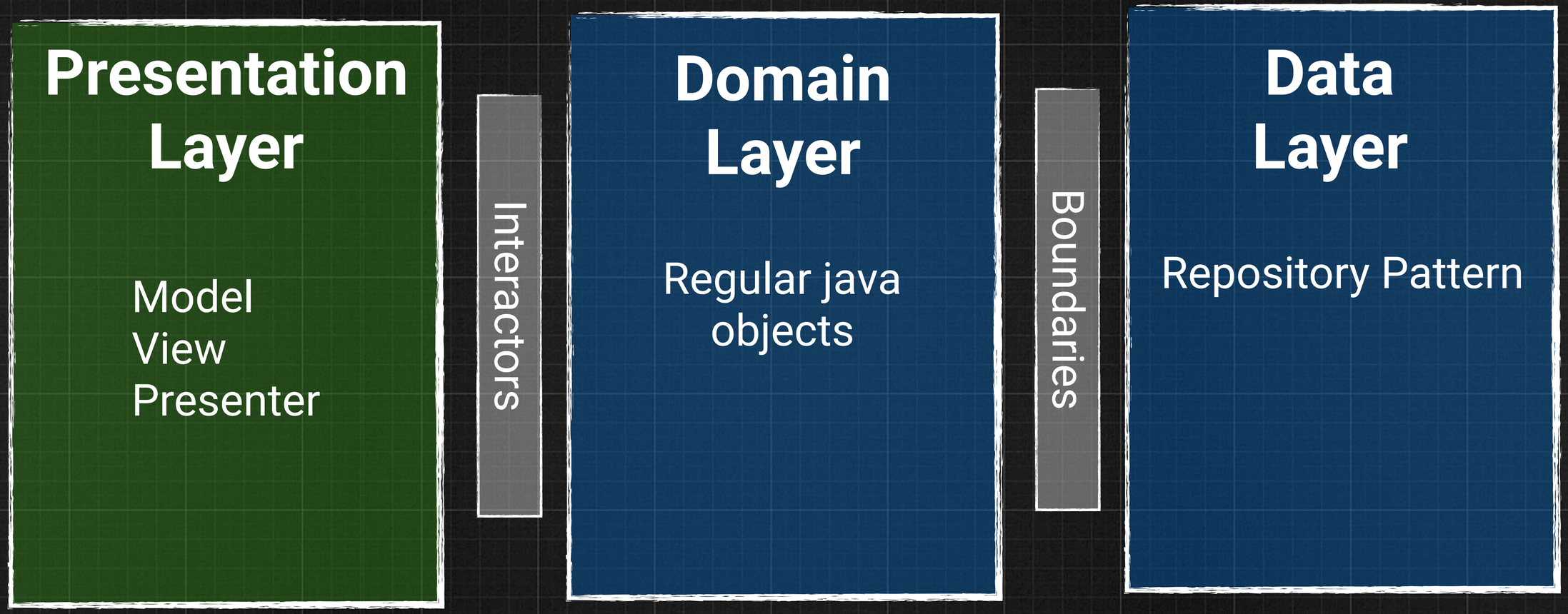
wallet
getMoney
getDocuments
getCards


Presentation Layer
Model
View
Presenter
Interactors
Domain Layer
Regular java objects
Boundaries
Data Layer
Repository Pattern



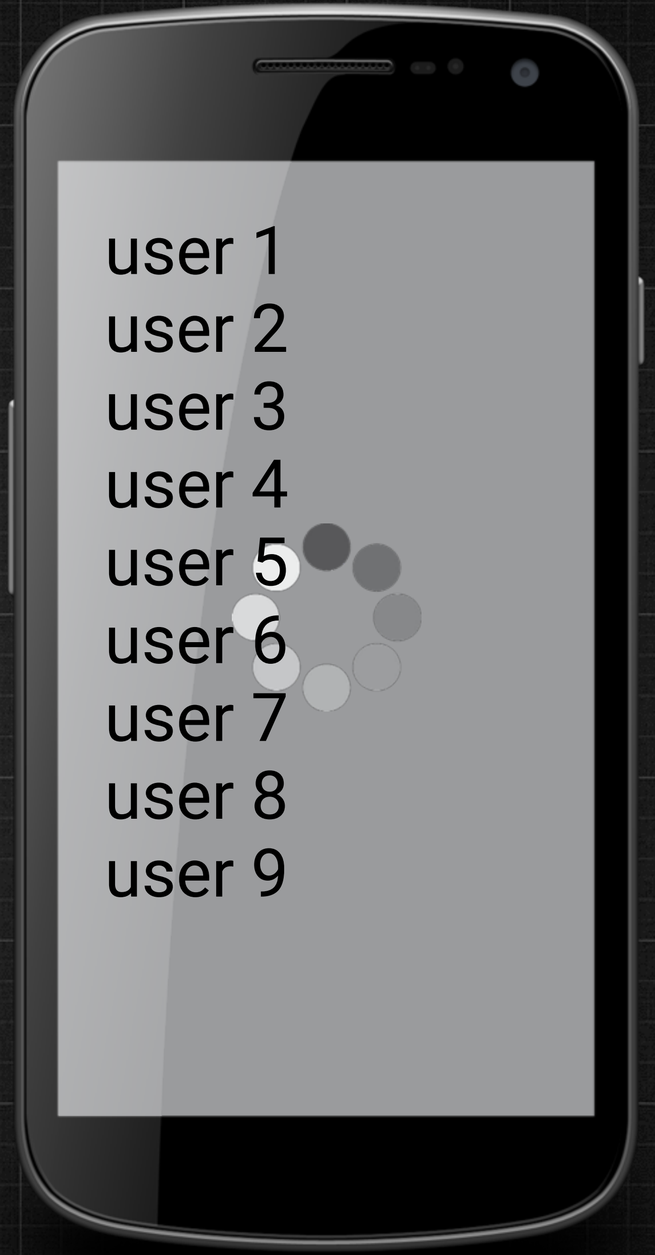
House Entity

House

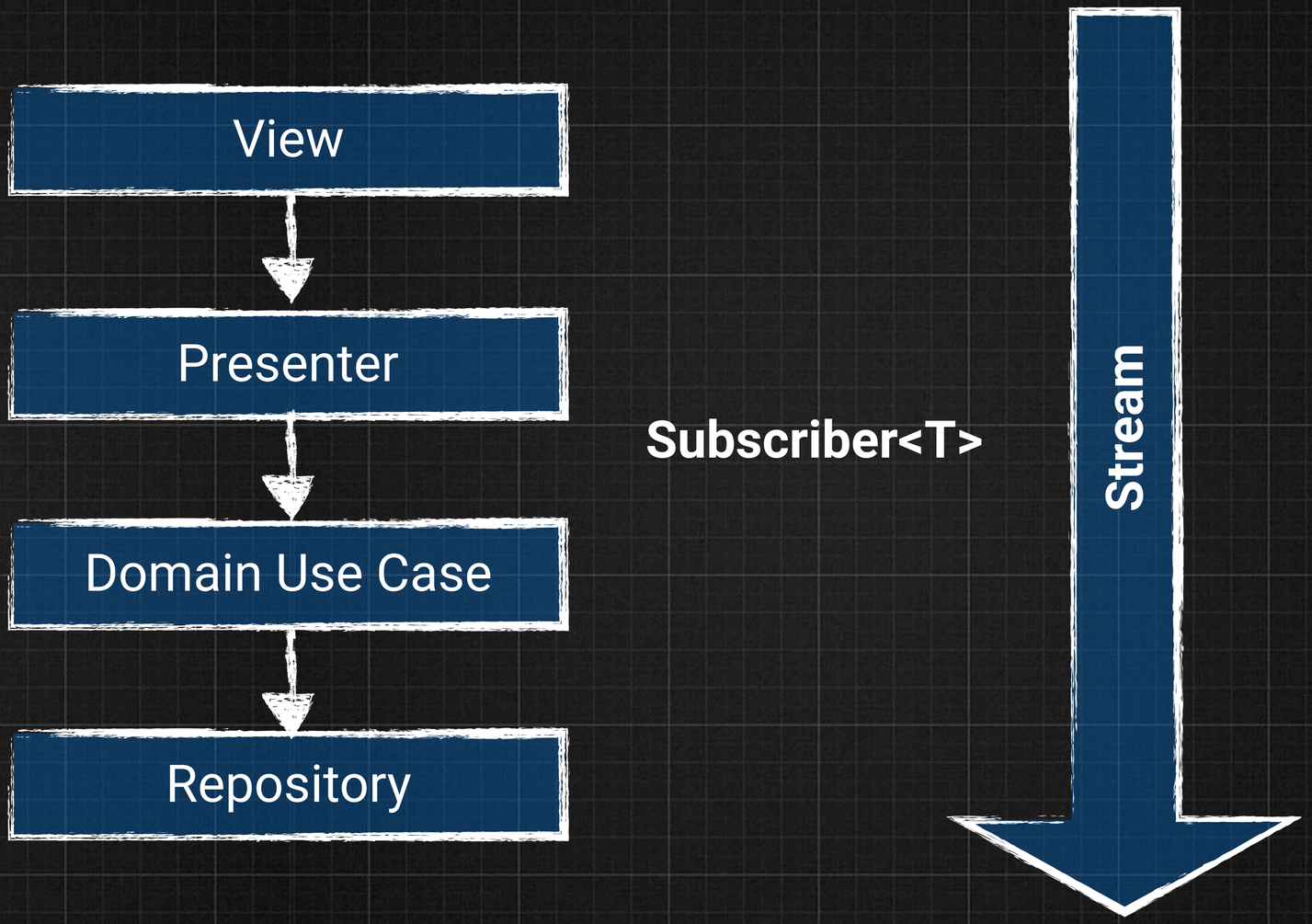

Presentation Layer
Model
View
Presenter
Interactors
Domain Layer
Regular java objects
Boundaries
Data Layer
Repository Pattern

Presenter

renderUserList
hideLoading
showLoading



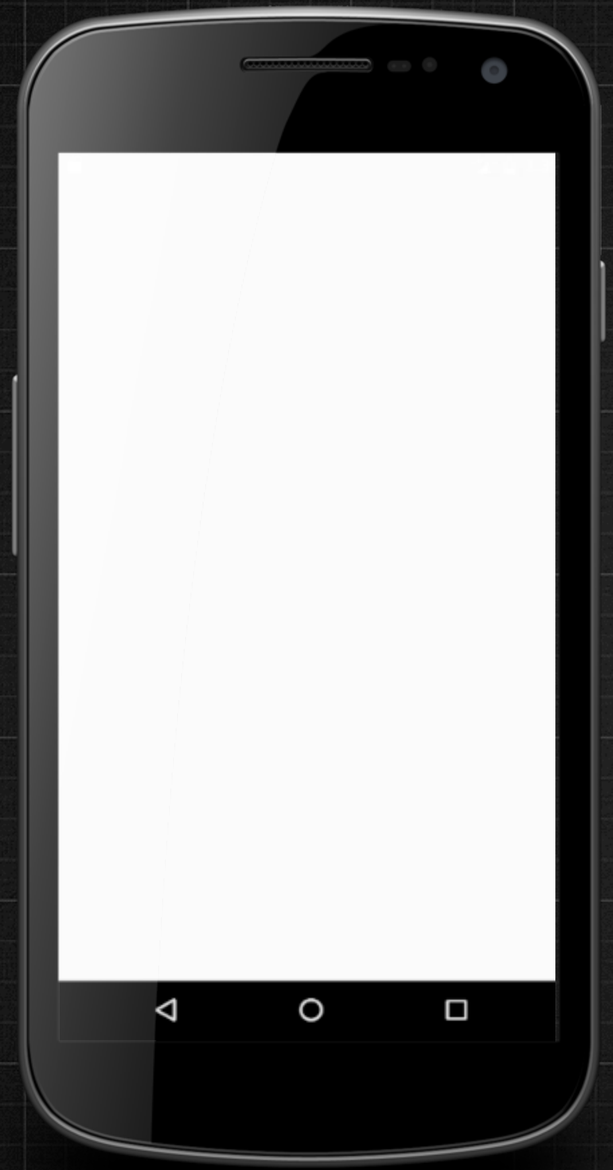
Reactive Approach

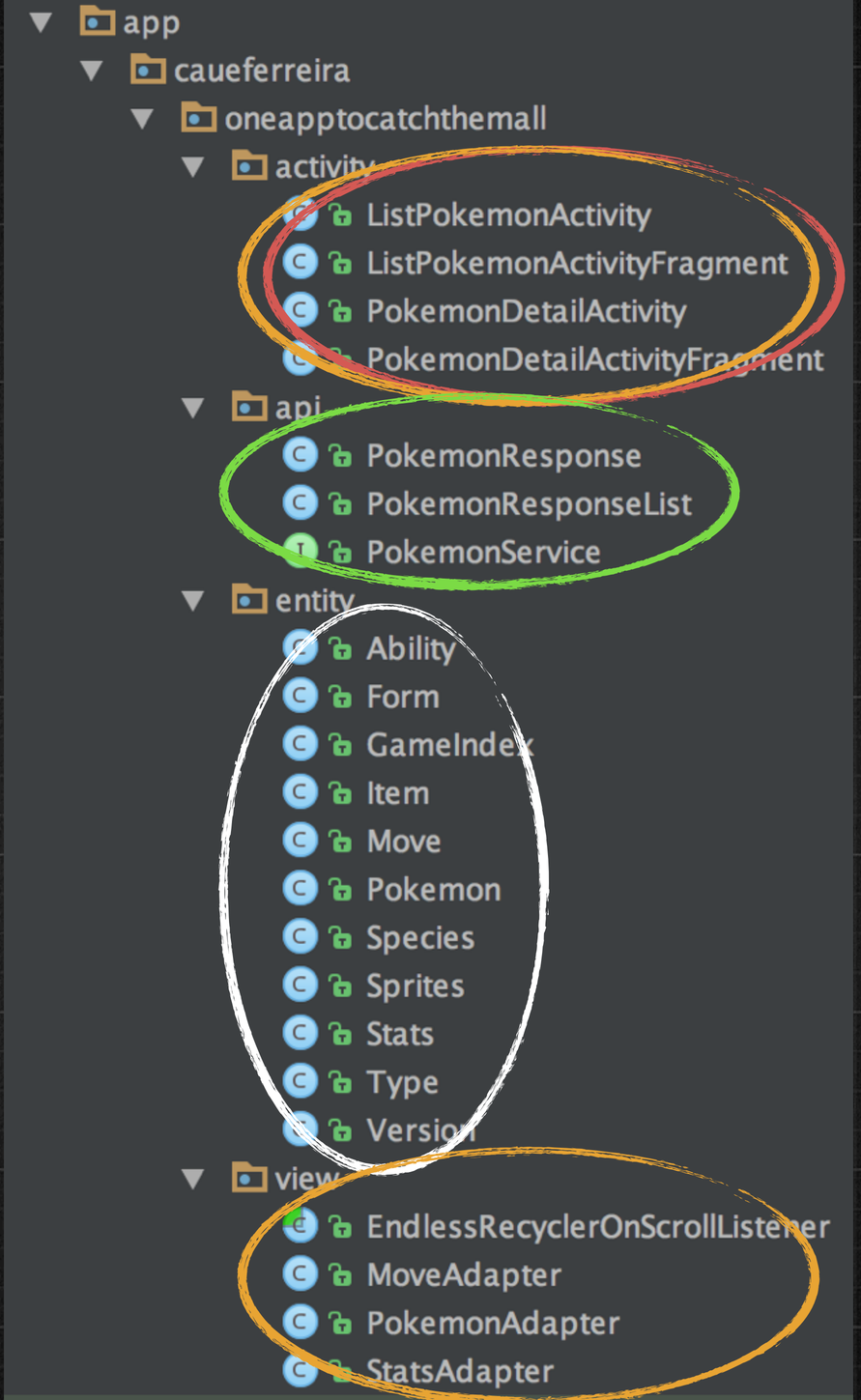


Our Case

https://pokeapi.co/

* list all pokemons
* show some details of the chosen pokemon





You've already seen it?

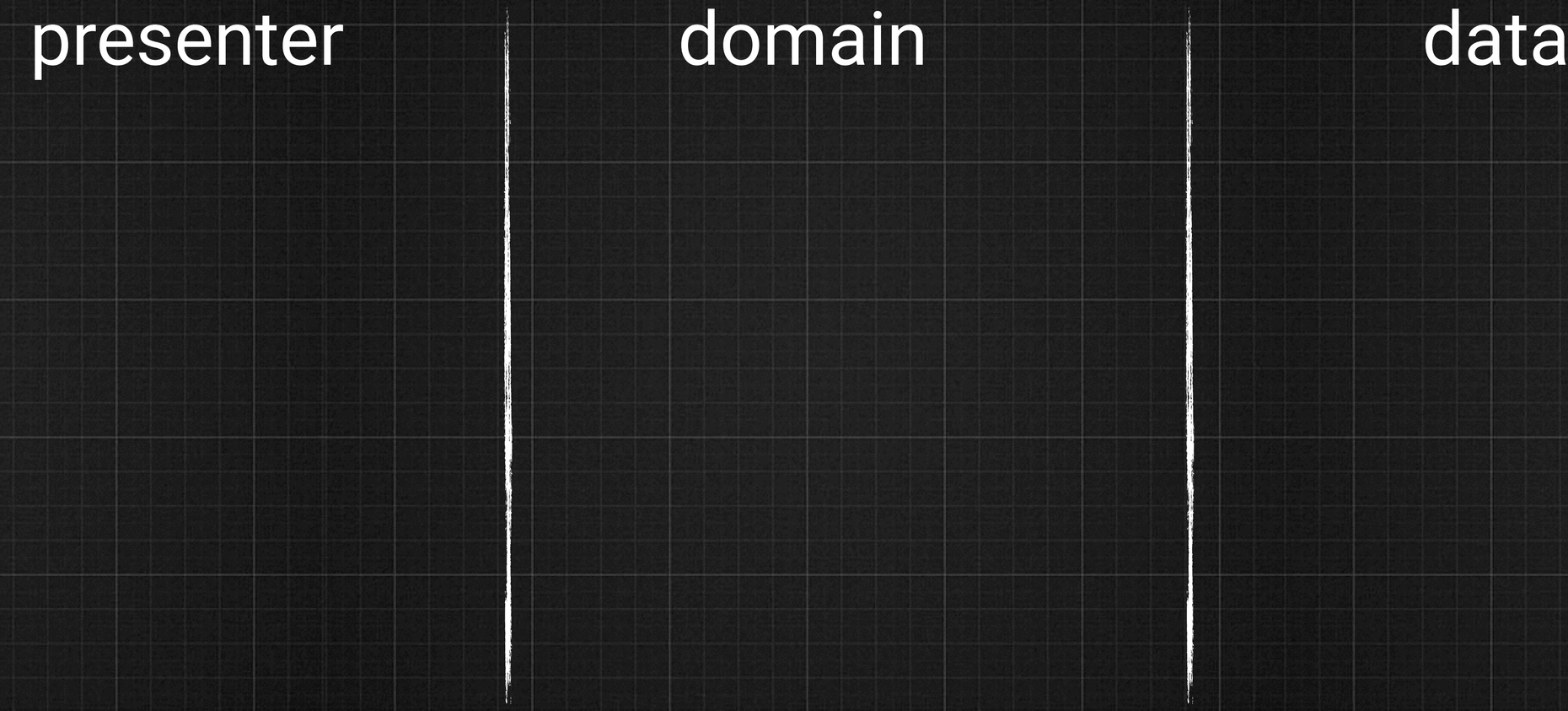
probably all logic from our app

wtf?

api entities been used everywhere

two package containing view level

What we want



Fragment

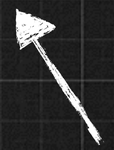
presenter | domain | data

pokemonListFragment

Interface with view methods

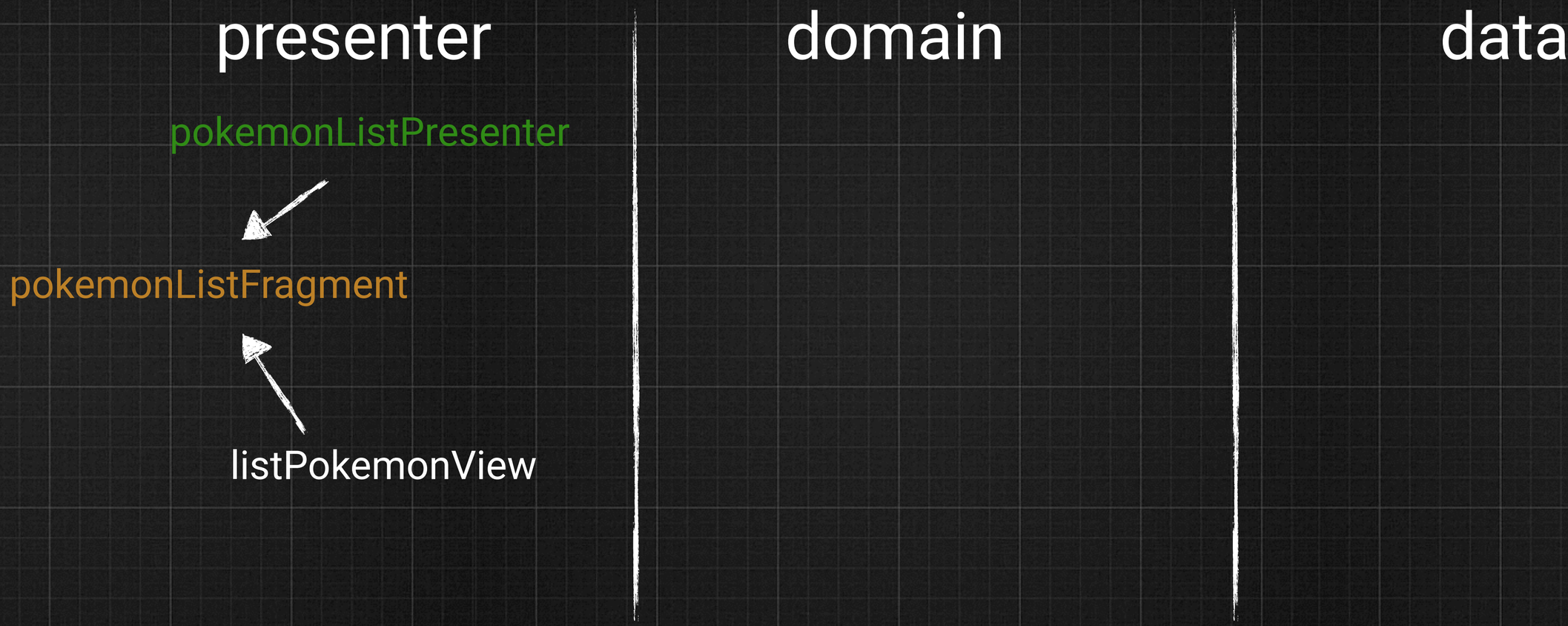
presenter | domain | data

pokemonListFragment

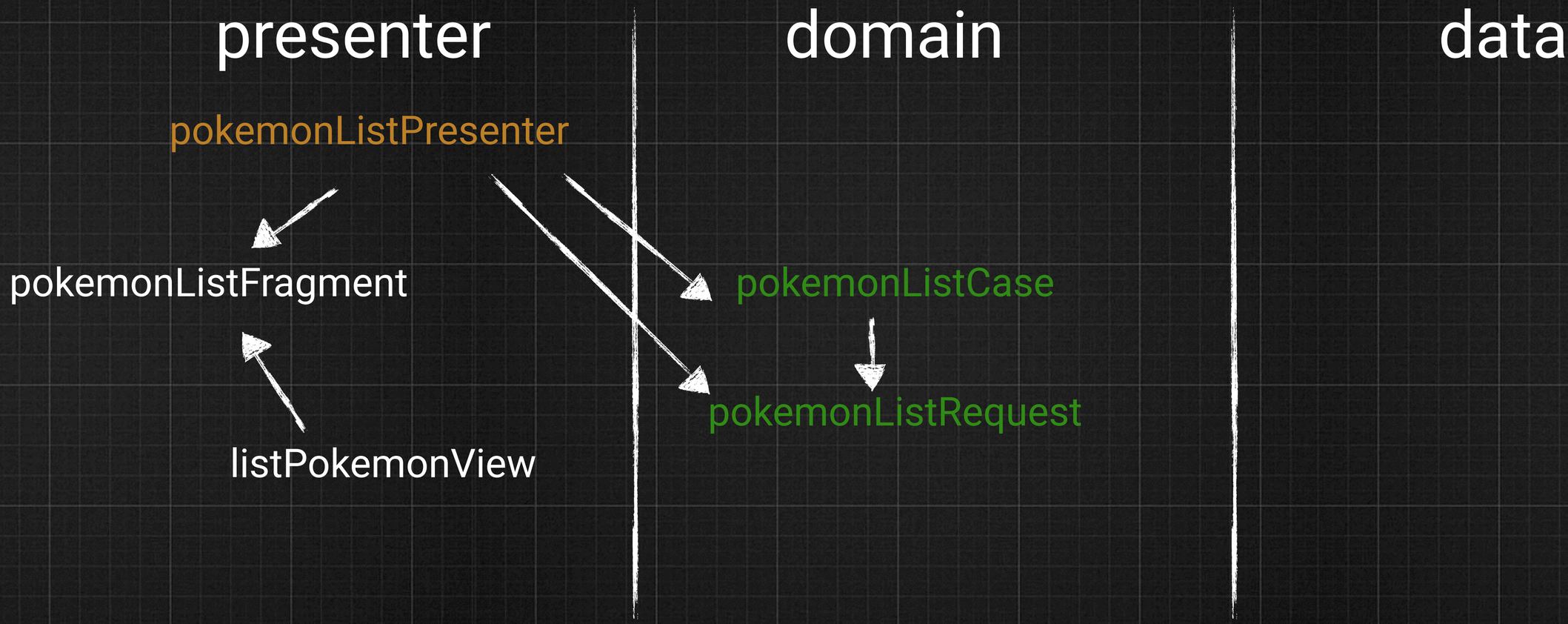


listPokemonView

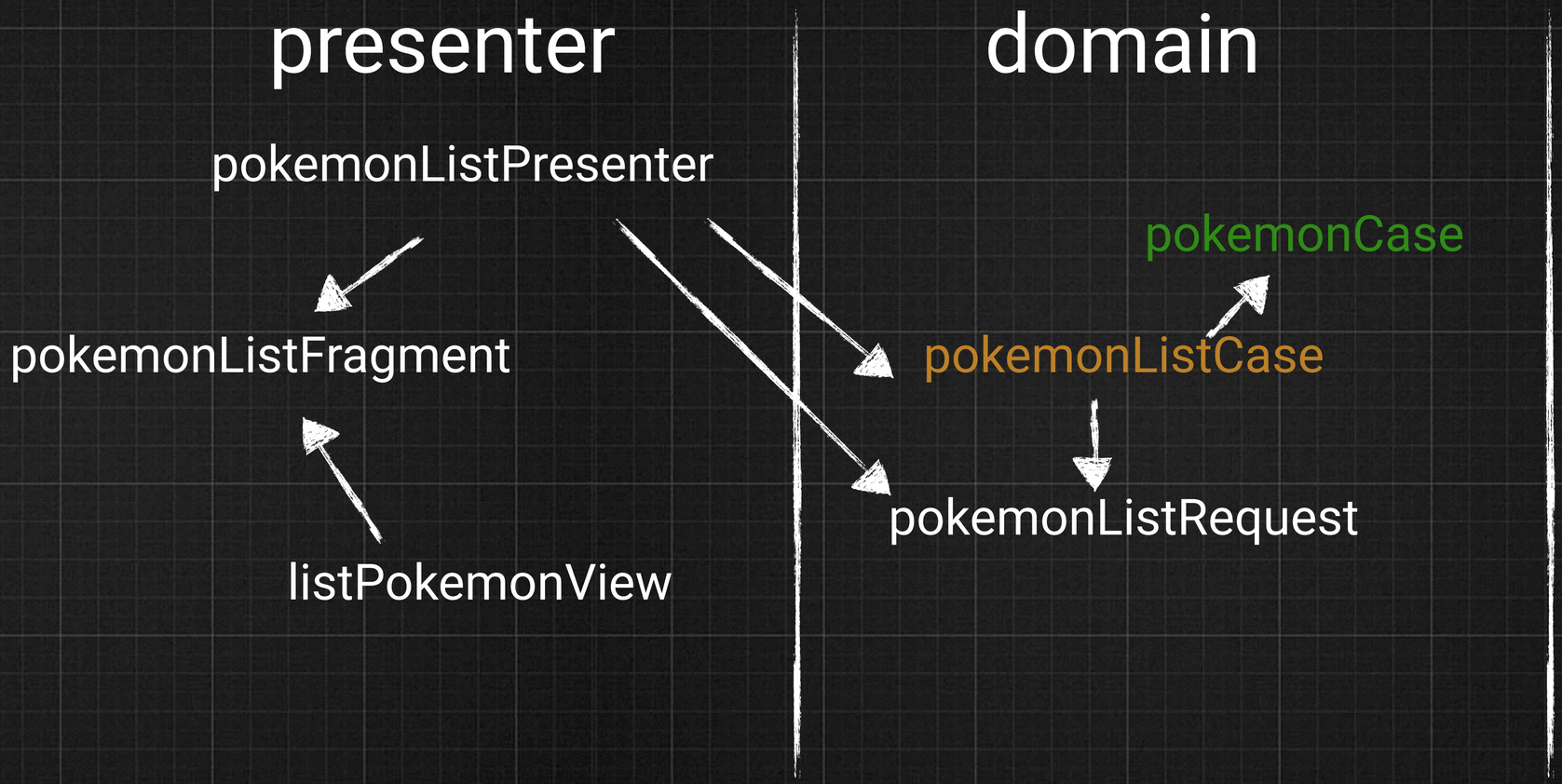
Presenter with fragment logic



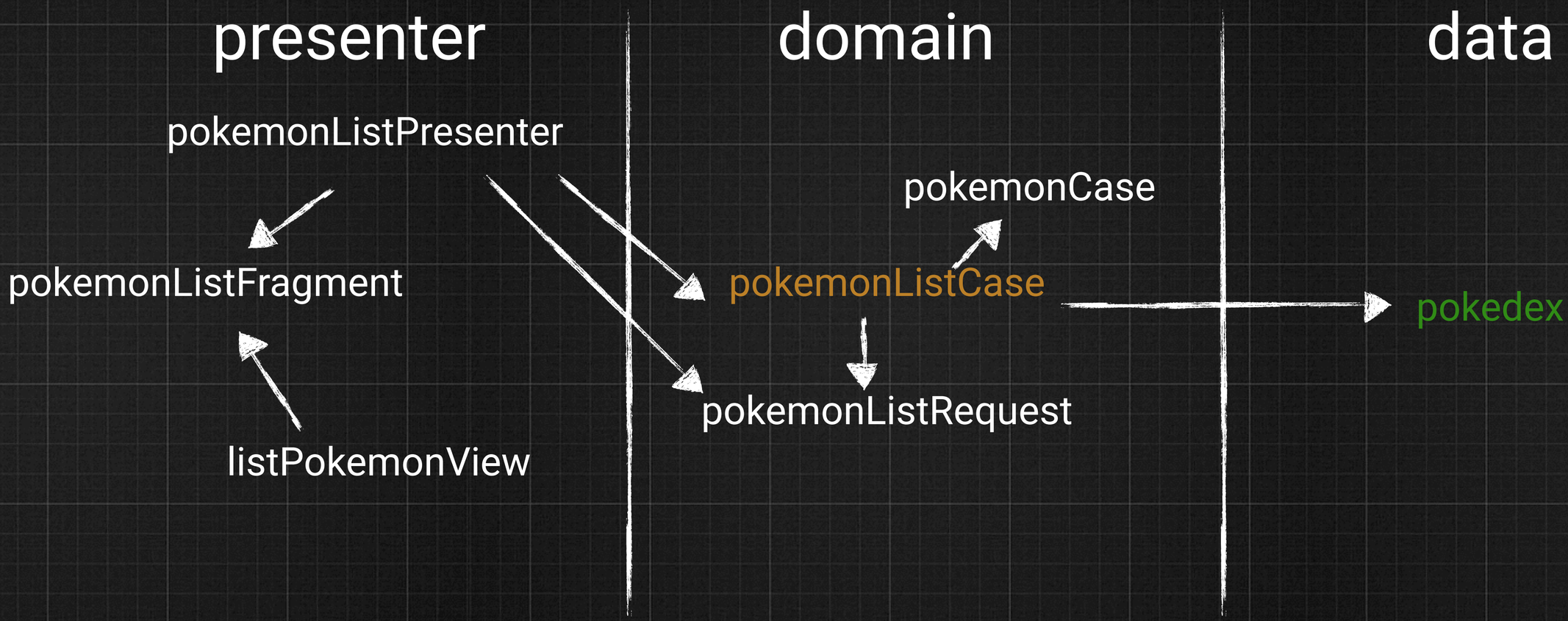
Case with business logic



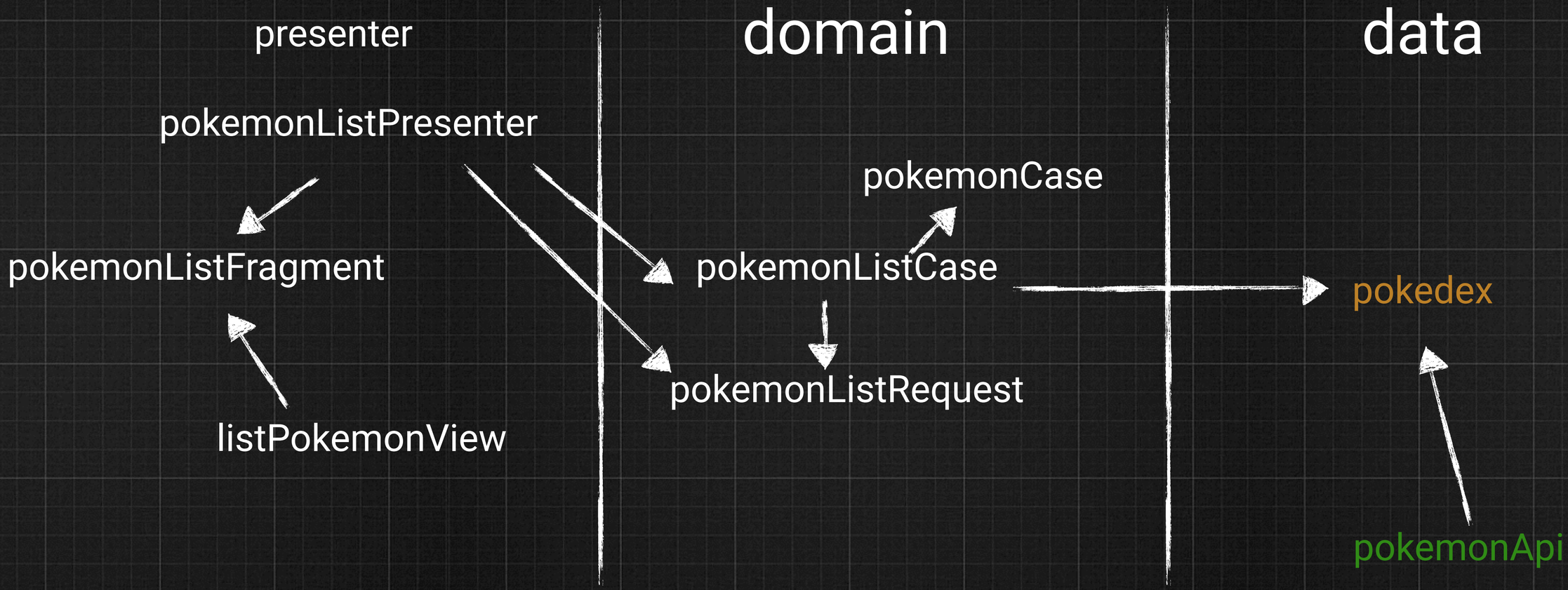
Case interface with data connection



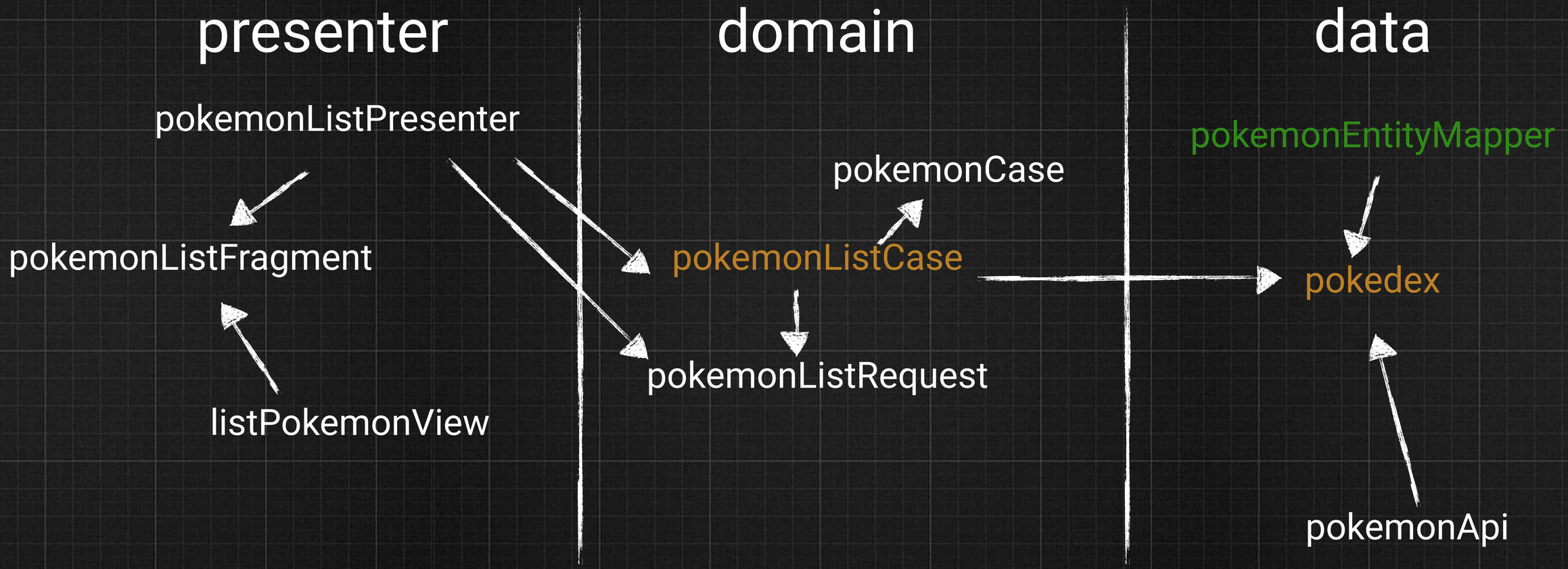
Repository


presenter
domain
data
pokemonListPresenter
pokemonListFragment
listPokemonView
pokemonCase
pokemonListCase
pokemonListRequest
pokedex

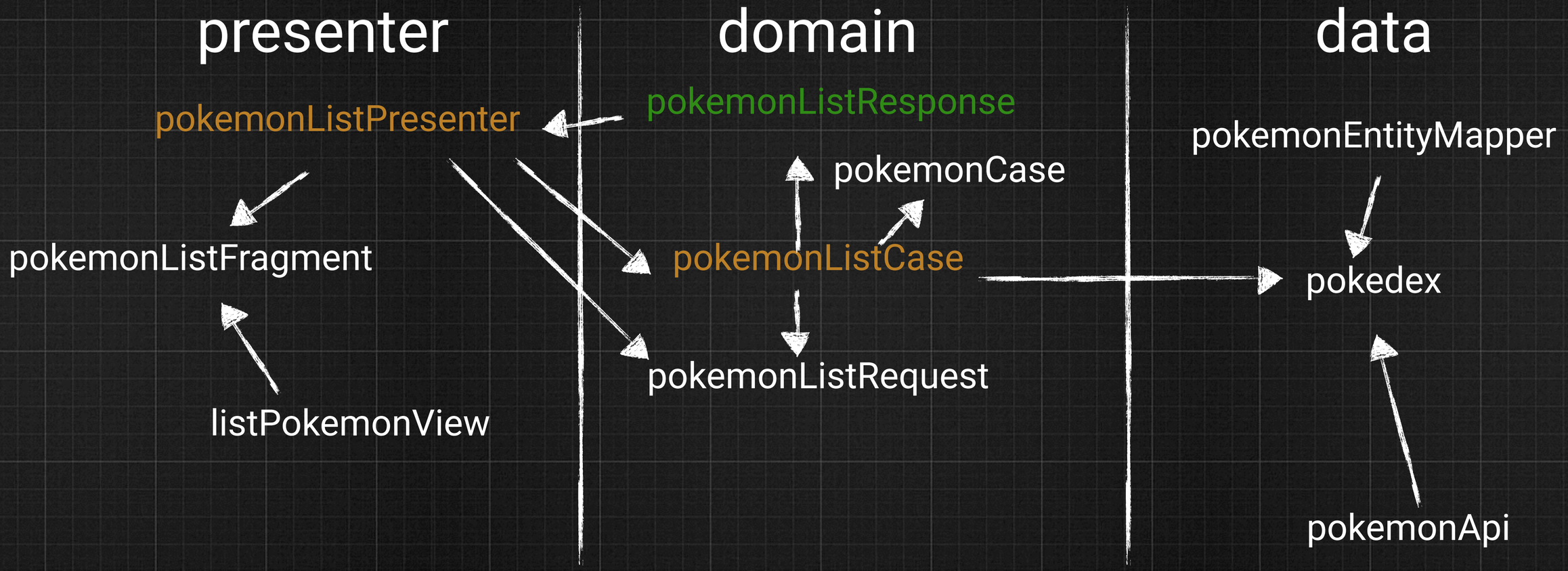
Rest API


presenter
domain
data
pokemonListPresenter
pokemonCase
pokemonListFragment
pokemonListCase
pokedex
pokemonListRequest
listPokemonView
pokemonApi

Mapper from data object to business object



Response from our request



Mapper from business object to request model

presenter

domain

data

pokemonListPresenter

pokemonListResponse

pokemonCase

pokemonEntityMapper

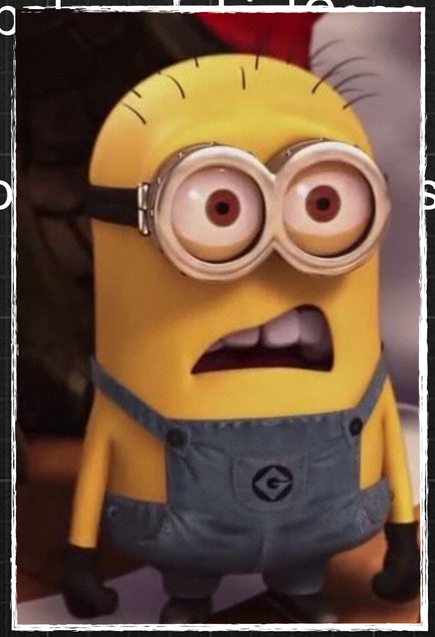
pokemonListFragment

pokedex

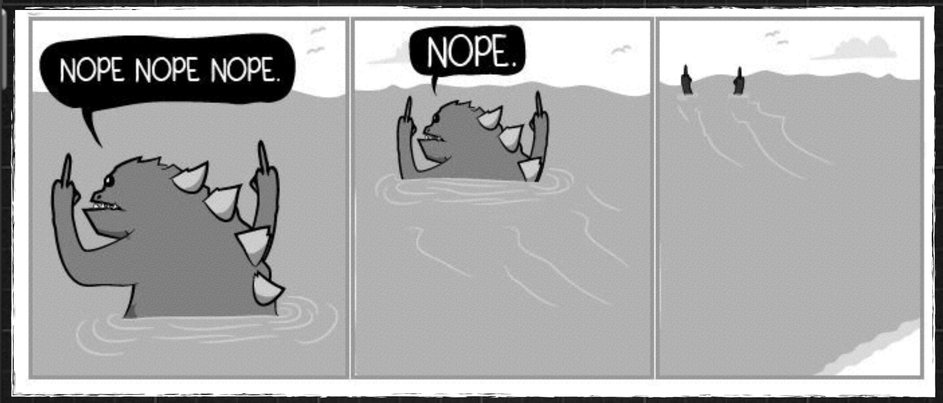
listPokemonView

pokemonModelMapper

pokemonApi



Zero to hero



"By the Flames of the Faltine!"

now

after

Why should I care?

* Makes changes less impactful
* Easier to change the project
* Increase productivity with multiple people
* Enhance test environment


KEEP
CALM
AND
SHOW ME
THE CODE