



# Что такое legacy?

# Что такое legacy?

- То, что написано не нами

# Что такое legacy?

~~То, что написано не нами~~

# Что такое legacy?

~~То, что написано не нами~~

Архитектуры нет или плохая архитектура

# Что такое legacy?

~~То, что написано не нами~~

Архитектуры нет или плохая архитектура

Слабоструктурированный или слишком запутанный код



# Что такое legacy?

~~То, что написано не нами~~

Архитектуры нет или плохая архитектура

Слабоструктурированный или слишком запутанный код

WTF/ в час превышает 9000

Как это Legacy мерять?



# Как это Legacy мерять?

$$L = T * H^1$$

---

<sup>1</sup> Т - время техдолга в часах, Н - часовая ставка разработчика

Но как определить  $T$ ?

# Что есть тех.долг?

# Что есть тех.долг?

- Ошибки и уязвимости

# Что есть тех.долг?

- Ошибки и уязвимости
- Проблемы сопровождаемости

# Что есть тех.долг?

- Ошибки и уязвимости
- Проблемы сопровождаемости
- Дублирование кода



# Что есть тех.долг?

- Ошибки и уязвимости
- Проблемы сопровождаемости
- Дублирование кода
- Низкий процент покрытия тестами

# Тулы для оценки

SonarQube



Bugs & Vulnerabilities

960 E

Bugs

85 E

Vulnerabilities

Code Smells

122d A

Debt

started 3 минуты назад

8.9k

Code Smells

Duplications



4.8%

Duplications

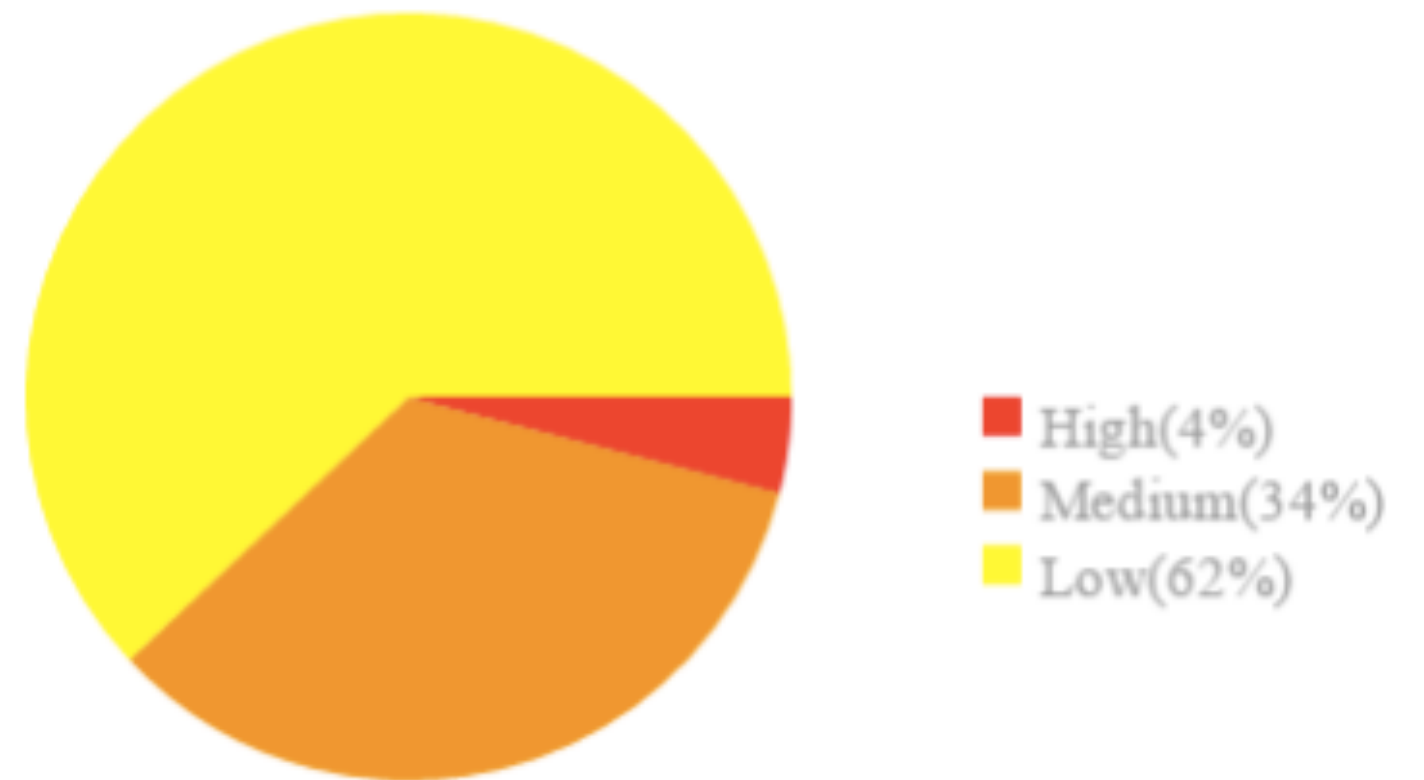
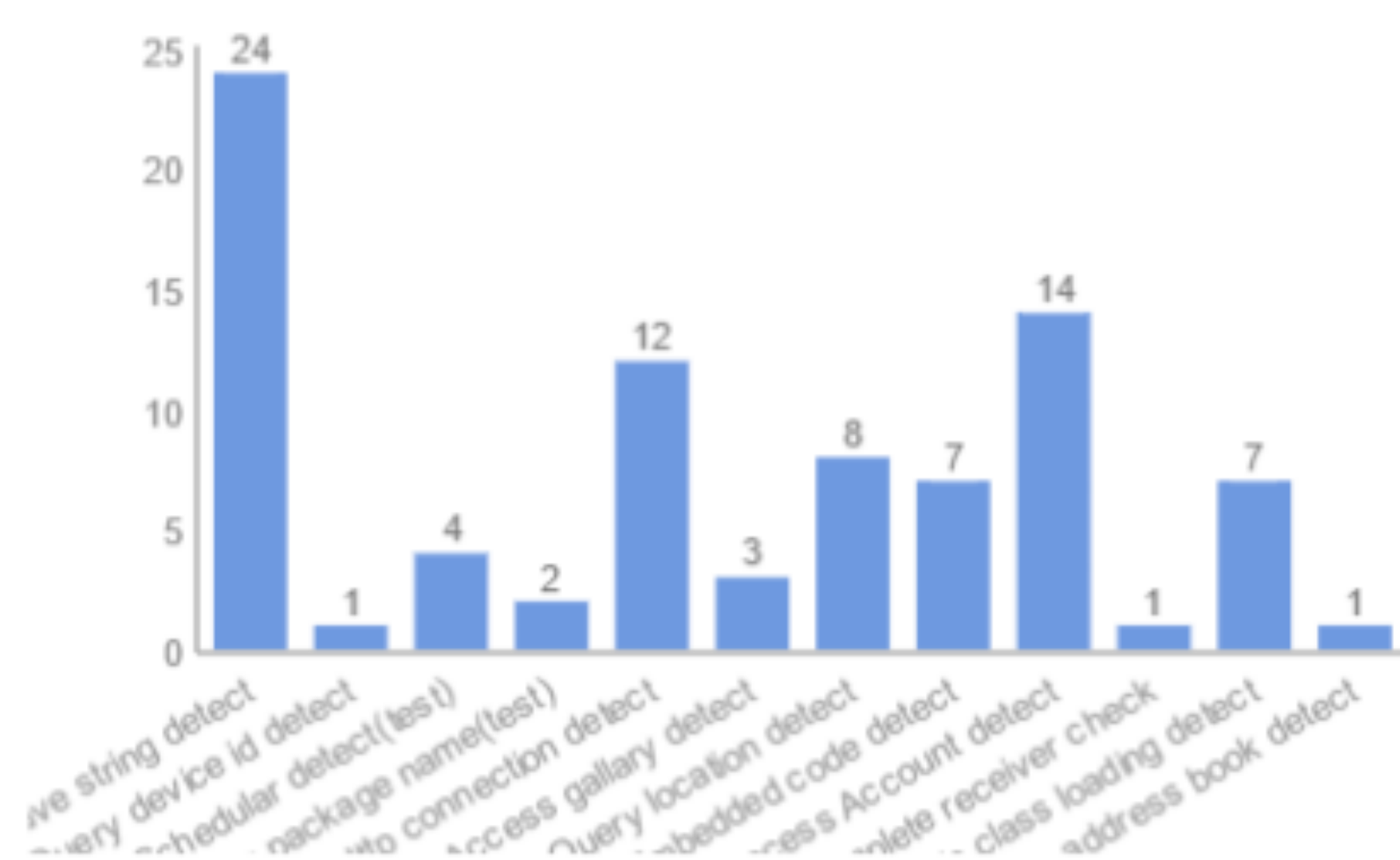
620

Duplicated Blocks

# Тулы для оценки

SonarQube

MobSF/Akana/Whatever Security Analysis tool



# Тулы для оценки

SonarQube

MobSF/Akana/Whatever Security Analysis tool

Architecture analysis tool

# Тулы для оценки

SonarQube

MobSF/Akana/Whatever Security Analysis tool

~~Architecture analysis tool~~

# Что такое архитектура?

Архитектура - это набор структур, состоящих из компонентов и связей между ними, а также свойств одних и вторых, определяющих свойства системы <sup>2</sup>

---

<sup>2</sup> Software Architecture in Practice, 3rd Edition, Bass, Clements, Kazman, Addison-Wesley



Что это к чертям значит?

# Что это к чертям значит?

- Архитектура - ключ к свойствам системы, которые важны пользователю



# Что это к чертям значит?

- Архитектура - ключ к свойствам системы, которые важны пользователю
- Или владелец продукта думает, что они важны

# Что это к чертям значит?

- Архитектура - ключ к свойствам системы, которые важны пользователю
- Или владелец продукта думает, что они важны
- То есть плохих или хороших архитектур не бывает, они либо подходят, либо не подходят под требования

# Когда нужно думать про архитектуру?

# Когда нужно думать про архитектуру?

- Если вы проверяете технологию, об архитектуре можно не думать

# Когда нужно думать про архитектуру?

- Если вы проверяете технологию, об архитектуре можно не думать
- Если вы пишете скрипт для отсылки почты по списку адресов, об архитектуре можно не думать

# Когда нужно думать про архитектуру?

- Если вы проверяете технологию, об архитектуре можно не думать
- Если вы пишете скрипт для отсылки почты по списку адресов, об архитектуре можно не думать
- Во всех остальных случаях, про архитектуру надо думать



# Какие наши целевые свойства?

# Какие наши целевые свойства?

- Быть производительным



# Какие наши целевые свойства?

- Быть производительным
- Быть локализуемым

# Какие наши целевые свойства?

- Быть производительным
- Быть локализуемым
- Работать оффлайн

# Какие наши целевые свойства?

- Быть производительным
- Быть локализуемым
- Работать оффлайн
- Быть тестируемым

# Какие наши целевые свойства?

- Быть производительным
- Быть локализуемым
- Работать оффлайн
- Быть тестируемым
- Быть расширяемым

# Тестируемость

# Тестируемость

- Аргументы и возвращаемые значения должны передаваться явно



# Тестируемость

- Аргументы и возвращаемые значения должны передаваться явно
- Зависимости должны быть переданы явно

# Тестируемость

- Аргументы и возвращаемые значения должны передаваться явно
- Зависимости должны быть переданы явно
- Зависимости должны быть заменяемы в тестах



# Расширяемость

# Расширяемость

- Изменения неизбежны

# Расширяемость

- Изменения неизбежны
- Сколько стоят реализация изменений:
  - Сколько стоит создание механизма для осуществления изменений?
  - Сколько стоит осуществление изменения с помощью этого механизма?

# Расширяемость

Для N похожих изменений примерное уравнение равно:

$$N * ChangeCost \leq MechanismCost + N * ( Change2Cost )$$

# Какие наши целевые свойства?

Быть производительным

Быть локализуемым

Работать оффлайн

Быть тестируемым

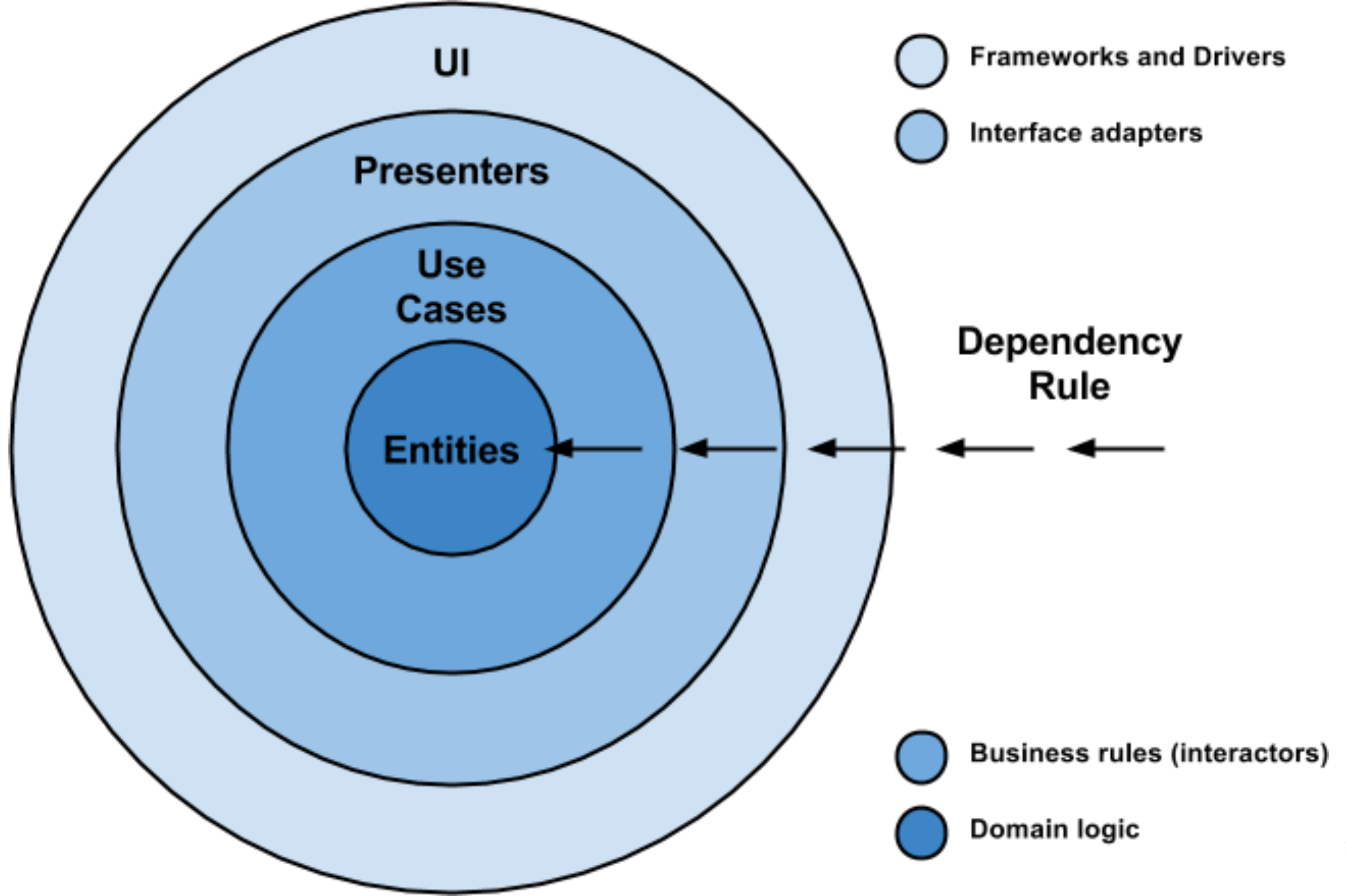
Быть расширяемым

Какая архитектура отвечает этим  
требованиям?

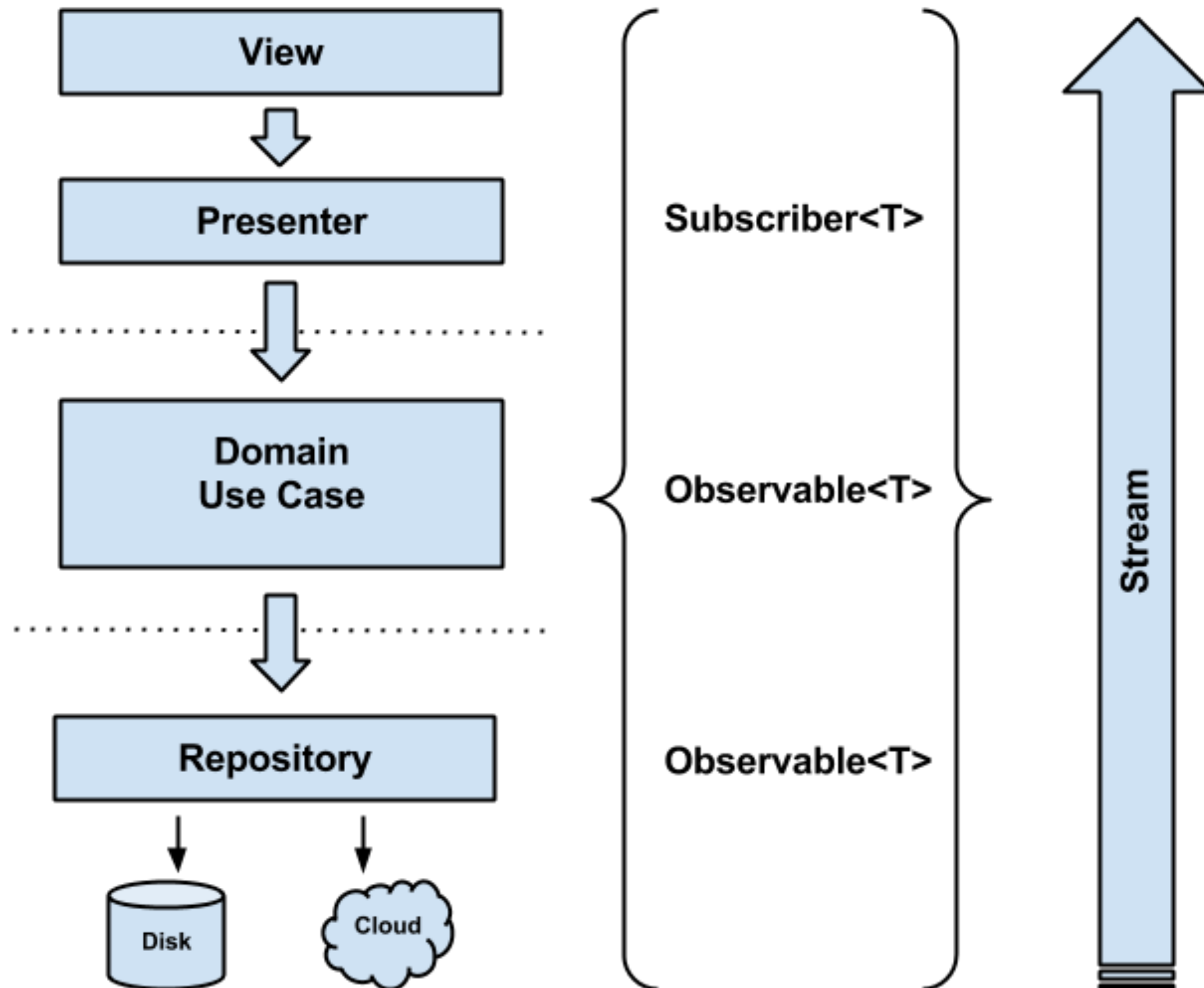


# Какая архитектура отвечает этим требованиям?

- Clean architecture







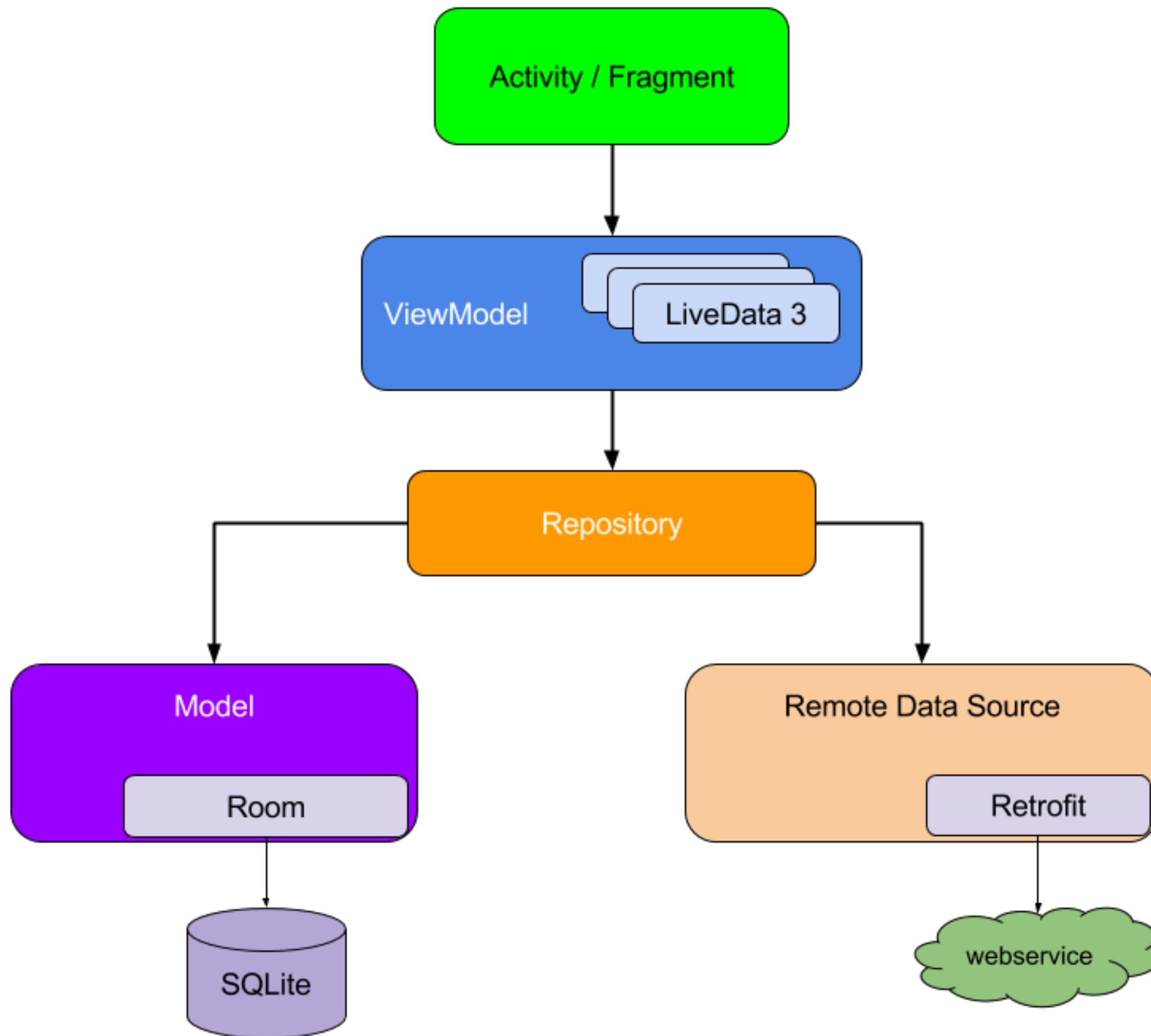
Какая архитектура отвечает этим  
требованиям?

# Какая архитектура отвечает этим требованиям?

- Clean architecture

# Какая архитектура отвечает этим требованиям?

- Clean architecture
- Google Android architecture



Какая архитектура отвечает этим  
требованиям?



# Какая архитектура отвечает этим требованиям?

- Clean architecture

# Какая архитектура отвечает этим требованиям?

- Clean architecture
- Google Android architecture



# Какая архитектура отвечает этим требованиям?

- Clean architecture
- Google Android architecture
- VIPER

Что общего?

# Что общего?

В строго поделенных на слои системах, слою позволено использовать только слой, расположенный непосредственно под ним<sup>3</sup>

---

<sup>3</sup> Software Architecture in Practice, 3rd Edition, Bass, Clements, Kazman, Addison-Wesley

# Пример с GreenDao

# Пример с GreenDao

## 1. 4 базы данных

# Пример с GreenDao

1. 4 базы данных

2. 30 таблиц



# Пример с GreenDao

1. 4 базы данных
2. 30 таблиц
3. 278 классов для доступа к данным

# Пример с GreenDao

1. 4 базы данных
2. 30 таблиц
3. 278 классов для доступа к данным
4. 20,000 строк кода для доступа к данным



# Пример с GreenDao

1. 4 базы данных
2. 30 таблиц
3. 278 классов для доступа к данным
4. 20,000 строк кода для доступа к данным
5. 0% фактического покрытия



# Пример из Entity

```
public class Entity {  
  
    private String name;  
    private Long id;  
  
    ...  
  
    public String getName();  
    ...  
  
}
```

# Пример - запрос Entity

```
@AutoFactory
public class GetEntityRequest extends SqlRequest {

    @Override
    protected String[] getColumns() {
        return new String[] {
            F + "." + ID_FIELD + " AS " + F_ID_FIELD,
            F + "." + NAME_FIELD + " AS " + F_NAME_FIELD,
            ...
            STORAGE + "." + ID_FIELD + " AS " + STORAGE_ID_FIELD,
            ...
            SYNC + "." + ID_FIELD + " AS " + SYNC_ID_FIELD,
            ...
        }
    }
}
```

# Пример - запрос Entity

```
@Override
public Entity createEntity() {
    if (!cursor.moveToFirst()) {
        return null;
    }

    ...
    String name = cursor.getString(cursor.getColumnIndex(F_NAME_FIELD));
    ...
    return new Entity(id, name, ...);
}
```

# Пример - запрос Entity

@AutoFactory

```
public class GetEntityRequest extends SqlRequest {
```

```
    ...
```

```
}
```



# Пример - запрос Entity

@AutoFactory

```
public class GetEntityRequest extends SqlRequest {  
    ...  
}
```

@Generated

```
public class GetEntityRequestFactory {  
    public GetEntityRequest create() {  
        return new GetEntityRequest();  
    }  
}
```



# Вызов запроса

```
public class MyEntityActivity extends Activity {  
  
    @Inject GetEntityRequestFactory requestFactory;  
  
    @Override  
    protected void onCreate(Bundle instance) {  
        super.onCreate(instance);  
        Entity entity = requestFactory.create().createEntity();  
        show(entity);  
    }  
}
```



# В чем проблема?

# В чем проблема?

1. Чтение диска на UI

# В чем проблема?

1. Чтение диска на UI
2. Большое количество кода, чтобы работало

# В чем проблема?

1. Чтение диска на UI
2. Большое количество кода, чтобы работало
3. Слой представления имеет доступ к данным, затрудняя юнит-тестирование

# В чем проблема?

1. Чтение диска на UI
2. Большое количество кода, чтобы работало
3. Слой представления имеет доступ к данным, затрудняя юнит-тестирование
4. Вы не знаете свою модель данных







# Выделение отдельного слоя

# Выделение отдельного слоя

## 1. Отдельный модуль



# Выделение отдельного слоя

1. Отдельный модуль
2. Покрытый на 100% тестами

# А зачем?

# А зачем?

1. Вы поймете свою модель

# А зачем?

1. Вы поймете свою модель
2. Обеспечите ее гарантированное функционирование

# А зачем?

1. Вы поймете свою модель
2. Обеспечите ее гарантированное функционирование
3. Защитите от неверного использования

# Как осознать модель данных



# Как осознать модель данных

Stetho<sup>4</sup>

---

<sup>4</sup> <http://facebook.github.io/stetho/>

# Как осознать модель данных

Stetho<sup>4</sup>  
Db files

---

<sup>4</sup> <http://facebook.github.io/stetho/>



# Как осознать модель данных

Stetho <sup>4</sup>  
or Android file Explorer  
Db files  
sqldelight <sup>5</sup>

---

<sup>4</sup> <http://facebook.github.io/stetho/>

<sup>5</sup> <https://github.com/square/sqldelight>

# Stetho

```
compile 'com.facebook.stetho:stetho:1.5.0'
```

# Stetho

```
public class MyApplication extends Application {  
    ...  
    public void onCreate() {  
        super.onCreate();  
        ...  
        Stetho.initializeWithDefaults(this);  
    }  
}
```

▶

Frames

▼

Web SQL

▼

apod.db

📄

android\_metadata

📄

rss\_items

📄

sqlite\_sequence

📄

IndexedDB

▶

Local Storage

▶

Session Storage

🍪

Cookies

📄

Application Cache

>

SELECT \* FROM rss\_items;

| _... | title                   | description_image_u... | description_text       |
|------|-------------------------|------------------------|------------------------|
| 43   | Our Galaxy's Magneti... | http://antwrp.gsfc...  | What does the magne... |
| 44   | The Milky Way over ...  | http://antwrp.gsfc...  | You may have heard ... |
| 45   | A Twisted Solar Eru...  | http://antwrp.gsfc...  | A Twisted Solar Eru... |
| 46   | Light from Cygnus A     | http://antwrp.gsfc...  | Celebrating astrono... |
| 47   | Interior View           | http://antwrp.gsfc...  | Interior View          |
| 48   | Launch to Lovejoy       | http://antwrp.gsfc...  | Launch to Lovejoy      |
| 49   | The Complex Ion Tai...  | http://antwrp.gsfc...  | What causes the str... |

>

SELECT \_id, title FROM rss\_items WHERE description\_text CONTAINS '%comet%';

✖

near "CONTAINS": syntax error (code 1): , while compiling: SELECT \_id, title FROM rss\_items WHERE description\_text CONTAINS '%comet%';

>

SELECT \_id, title FROM rss\_items WHERE description\_text LIKE '%comet%';

| _id | title                                 |
|-----|---------------------------------------|
| 49  | The Complex Ion Tail of Comet Lovejoy |

>

PRAGMA user\_version;

| user_version |
|--------------|
| 1            |

>

Samsung SM-G930F Android 7.0, API 24

| Name                                                        | Permissions |
|-------------------------------------------------------------|-------------|
| ▶ acct                                                      | dr-xr-xr-x  |
| ▶ cache                                                     | drwxrwx---  |
| ▶ config                                                    | drwxr-xr-x  |
| ▶ cpefs                                                     | drwxrwx--x  |
| ▶ d                                                         | lrwxrwxrwx  |
| ▼ data                                                      | drwxrwx--x  |
| ▶ app                                                       | drwxrwx--x  |
| ▼ data                                                      | drwxrwx--x  |
| ▶ android                                                   | drwxrwx--x  |
| ▶ com.android.apps.tag                                      | drwxrwx--x  |
| ▶ com.android.backupconfirm                                 | drwxrwx--x  |
| ▶ com.android.bluetooth                                     | drwxrwx--x  |
| ▶ com.android.bluetoothmidiservice                          | drwxrwx--x  |
| ▶ com.android.bookmarkprovider                              | drwxrwx--x  |
| ▼ com.android.calendar                                      | drwxrwx--x  |
| run-as: Could not set capabilities: Operation not permitted |             |
| ▶ com.android.callogbackup                                  | drwxrwx--x  |

# sqldelight

```
buildscript {  
    repositories {  
        mavenCentral()  
    }  
    dependencies {  
        classpath 'com.squareup.sqldelight:gradle-plugin:0.6.1'  
    }  
}  
  
apply plugin: 'com.squareup.sqldelight'
```



# sql delight

```
CREATE TABLE entity (  
  _id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,  
  name TEXT NOT NULL  
);
```

# sql delight

```
public class Entity {  
    private Long id;  
  
    private String name;  
  
    public Long getId() { ... }  
    public String getName() { ... }  
};
```



# sql delight

```
@Table(name = "Entity")
public class Entity {

    @Column(name = "_id")
    private Long _id;

    @Column(name = "NAME")
    private String name;

    public Long getId() { ... }
    public String getName() { ... }
};
```

# Интеграция в приложение - CIDR!

# Интеграция в приложение - CIDR!

1. Create unit test

# Интеграция в приложение - CIDR!

1. Create unit test
2. Implement

# Интеграция в приложение - CIDR!

1. Create unit test
2. Implement
3. Deprecate

# Интеграция в приложение - CIDR!

1. Create unit test
2. Implement
3. Deprecate
4. Replace



# Интеграция в приложение - CIDR!

# Интеграция в приложение - CIDR!

1. Create unit test



# Create unit test

```
@Override
protected void onCreate(Bundle instance) {
    super.onCreate(instance);
    Entity entity = requestFactory.create().createEntity();
    show(entity);
}
```

# Create unit test

```
@Override  
protected void onCreate(Bundle instance) {  
    super.onCreate(instance);  
    Entity entity = requestFactory.create().createEntity();  
    show(entity);  
}
```

# Create unit test

```
@Override
protected void onCreate(Bundle instance) {
    super.onCreate(instance);
    GetEntityRequest request = requestFactory.create();
    Entity entity = request.createEntity();
    show(entity);
}
```

# Create unit test

```
@VisibleForTesting  
protected Entity loadEntity() {  
    return requestFactory.create().createEntity();  
}
```

# Create unit test

```
@VisibleForTesting
protected Observable<Entity> loadEntity() {
    return Single.just(
        requestFactory.create().createEntity()
    );
}
```

# Create unit test

```
@Override
protected void onCreate(Bundle instance) {
    super.onCreate(instance);
    loadEntity()
        .subscribeOn(Schedulers.io())
        .observeOn(AndroidSchedulers.mainThread());
    .subscribe({entity, throwable} -> {
        show(entity);
    });
}
```



# Create unit test

```
@Override
protected void onCreate(Bundle instance) {
    super.onCreate(instance);
    viewModel.loadEntity()
        .subscribeOn(Schedulers.io())
        .observeOn(AndroidSchedulers.mainThread());
    .subscribe({entity, throwable} -> {
        show(entity);
    });
}
```

# Create unit test

```
@Test
public void shouldLoadEntity() {
    ViewModel viewModel = new ViewModel();

    viewModel.loadEntity()
        .subscribe({entity, throwable} -> {
            checkEntity(entity);
        });
}
```



# Create unit test

```
public void checkEntity(Entity entity) {  
    assertEquals(NAME, entity.getName());  
    assertEquals(ID, entity.getId());  
    ...  
}
```

# Интеграция в приложение - CIDR!

# Интеграция в приложение - CIDR!

1. Create unit test

# Интеграция в приложение - CIDR!

1. Create unit test
2. Implement

# Implement

```
public interface EntityDAO {  
    Observable<Entity> loadEntity();  
}
```

# Implement

```
public class ViewModel {  
  
    @Inject EntityDao entityDao;  
  
    public Observable<Entity> loadEntity() {  
        return entityDao.loadEntity();  
    }  
}
```



# Implement

```
public class GreenDaoEntityDAO implements EntityDAO {  
  
    public Observable<Entity> loadEntity() {  
        RxDao<DbEntity> rxDao = daoSession.getEntityDao().rx();  
        return rxDao.load();  
    }  
}
```

# Проблема №1



# Проблема №1

1. Объект из базы - это не то, что удобно использовать в представлении

# Проблема №1

1. Объект из базы - это не то, что удобно использовать в представлении
2. Скорее у вас есть Entity и DbEntity.

# Проблема №1

1. Объект из базы - это не то, что удобно использовать в представлении
2. Скорее у вас есть Entity и DbEntity.
3. Что же делать?

# Конвертеры!

# Конвертеры

```
public class GreenDaoEntityDAO implements EntityDAO {  
  
    public Observable<Entity> loadEntity() {  
        RxDao<DbEntity> rxDao = daoSession.getEntityDao().rx();  
        return rxDao.load().flatMap(dbEntity -> convert(dbEntity));  
    }  
  
}
```

# Конвертеры

```
public class GreenDaoEntityDAO implements EntityDAO {  
  
    public Observable<Entity> loadEntity() {  
        RxDao<DbEntity> rxDao = daoSession.getEntityDao().rx();  
        return rxDao.load().flatMap(dbEntity -> convert(dbEntity));  
    }  
  
}
```



# Проблема №2

# Проблема №2

1. Младшим разработчикам наплевать на эту вашу архитектуру



# Данные из модуля приложения...

```
public class GreenDaoEntityDAO implements EntityDAO {  
  
    public Observable<Entity> loadEntity() {  
        String userId = ApplicationSettings.getUserId();  
        RxDao<DbEntity> rxDao = daoSession.getEntityDao().rx();  
        return rxDao.load(userId).flatMap(dbEntity -> convert(dbEntity));  
    }  
  
}
```

Понадобилось кому почитать данные из  
модуля приложения...

# Понадобилось кому почитать данные из модуля приложения...

- Переносим слой в отдельный модуль

# Понадобилось кому почитать данные из модуля приложения...

- Переносим слой в отдельный модуль
- Покрываем его тестами

# Понадобилось кому почитать данные из модуля приложения...

- Переносим слой в отдельный модуль
- Покрываем его тестами
- Поставляем бинарно

# Интеграция в приложение - CIDR!



# Интеграция в приложение - CIDR!

1. Create unit test

# Интеграция в приложение - CIDR!

1. Create unit test
2. Implement



# Интеграция в приложение - CIDR!

1. Create unit test
2. Implement
3. Deprecate

# Deprecate

```
/**
 * @deprecated Please, use {@link EntityDao#loadEntity()}
 */
@Deprecated
@AutoFactory
public class GetEntityRequest {
    ...
}
```

# Интеграция в приложение - CIDR!

# Интеграция в приложение - CIDR!

1. Create unit test

# Интеграция в приложение - CIDR!

1. Create unit test
2. Implement

# Интеграция в приложение - CIDR!

1. Create unit test
2. Implement
3. Deprecate



# Интеграция в приложение - CIDR!

1. Create unit test
2. Implement
3. Deprecate
4. Replace

# Финальная статистика



# Финальная статистика

## 1. 4 базы данных

# Финальная статистика

1. 4 базы данных

2. 30 таблиц

# Финальная статистика

1. 4 базы данных
2. 30 таблиц
3. 84 класса для доступа к данным (было 278)

# Финальная статистика

1. 4 базы данных
2. 30 таблиц
3. 84 класса для доступа к данным (было 278)
4. 7400 строк кода вместе с тестами (было 20,000 без тестов)

# Финальная статистика

1. 4 базы данных
2. 30 таблиц
3. 84 класса для доступа к данным (было 278)
4. 7400 строк кода вместе с тестами (было 20,000 без тестов)
5. 100% фактического покрытия (было 0%)

Quality Gate

Passed

### Bugs & Vulnerabilities

0 <sup>A</sup>  
Bugs

0 <sup>A</sup>  
Vulnerabilities

### Code Smells

20min <sup>A</sup>  
Debt

1  
Code Smells

started 5 минут назад

### Duplications

○ 0.0%  
Duplications

0  
Duplicated Blocks



# Какие целевые свойства достигнуты?

# Какие целевые свойства достигнуты?

- Быть производительным 



# Какие целевые свойства достигнуты?

- Быть производительным 
- Быть локализуемым

# Какие целевые свойства достигнуты?

- Быть производительным 
- Быть локализуемым
- Работать оффлайн 

# Какие целевые свойства достигнуты?

- Быть производительным 
- Быть локализуемым
- Работать оффлайн 
- Быть тестируемым 

# Какие целевые свойства достигнуты?

- Быть производительным 
- Быть локализуемым
- Работать оффлайн 
- Быть тестируемым 
- Быть расширяемым 

Где еще можно использовать такой  
подход?

# Где еще можно использовать такой подход?

- Переделывая взаимодействие с сетью



# Где еще можно использовать такой подход?

- Переделывая взаимодействие с сетью
- Переходя на MVVM

# Где еще можно использовать такой подход?

- Переделывая взаимодействие с сетью
- Переходя на MVVM
- Вообще, вводя любой новый слой



# Где этот подход не помогает?

# Где этот подход не помогает?

- С секьюрити

# Где этот подход не помогает?

- С секьюрити
- С тулами

# Где этот подход не помогает?

- С секьюрити
- С тулами
- С любым межкомпонентным code

# О докладчике

# О докладчике

- Владимир Иванов - ведущий разработчик в ЕРАМ



# О докладчике

- Владимир Иванов - ведущий разработчик в ЕРАМ
- Больше 6 лет разработки под Android

# О докладчике

- Владимир Иванов - ведущий разработчик в ЕРАМ
- Больше 6 лет разработки под Android
- Больше 15 опубликованных приложений



# О докладчике

- Владимир Иванов - ведущий разработчик в ЕРАМ
- Больше 6 лет разработки под Android
- Больше 15 опубликованных приложений
- Широкий кругозор в мобильных технологиях

