

《Android Kotlin协程实战》

项目实战

让人人都能享受到高品质的教育服务

协程项目实战

目 录

1 Paging3 RemoteMediator

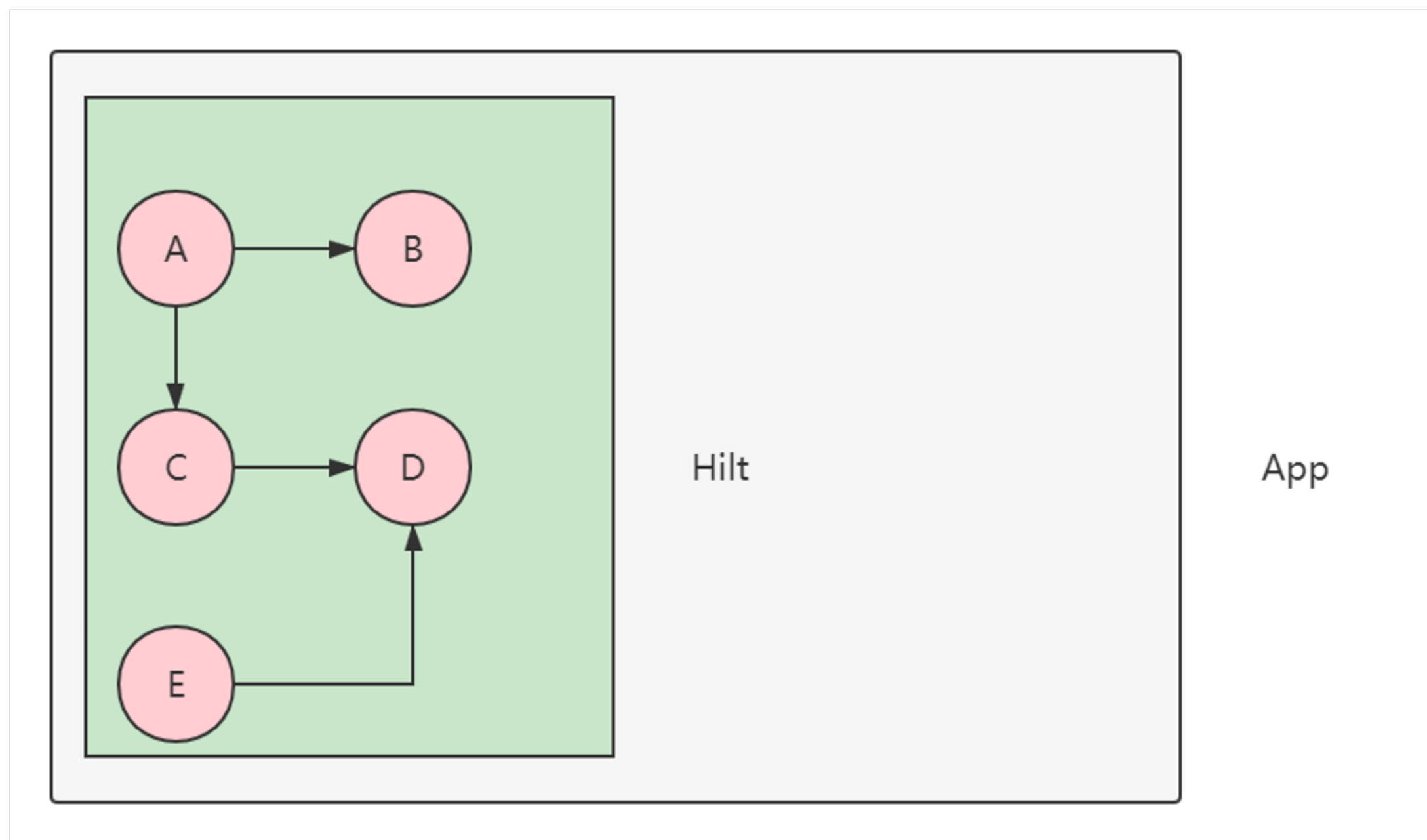
2 Room、Retrofit

3 Flow

4 Hilt

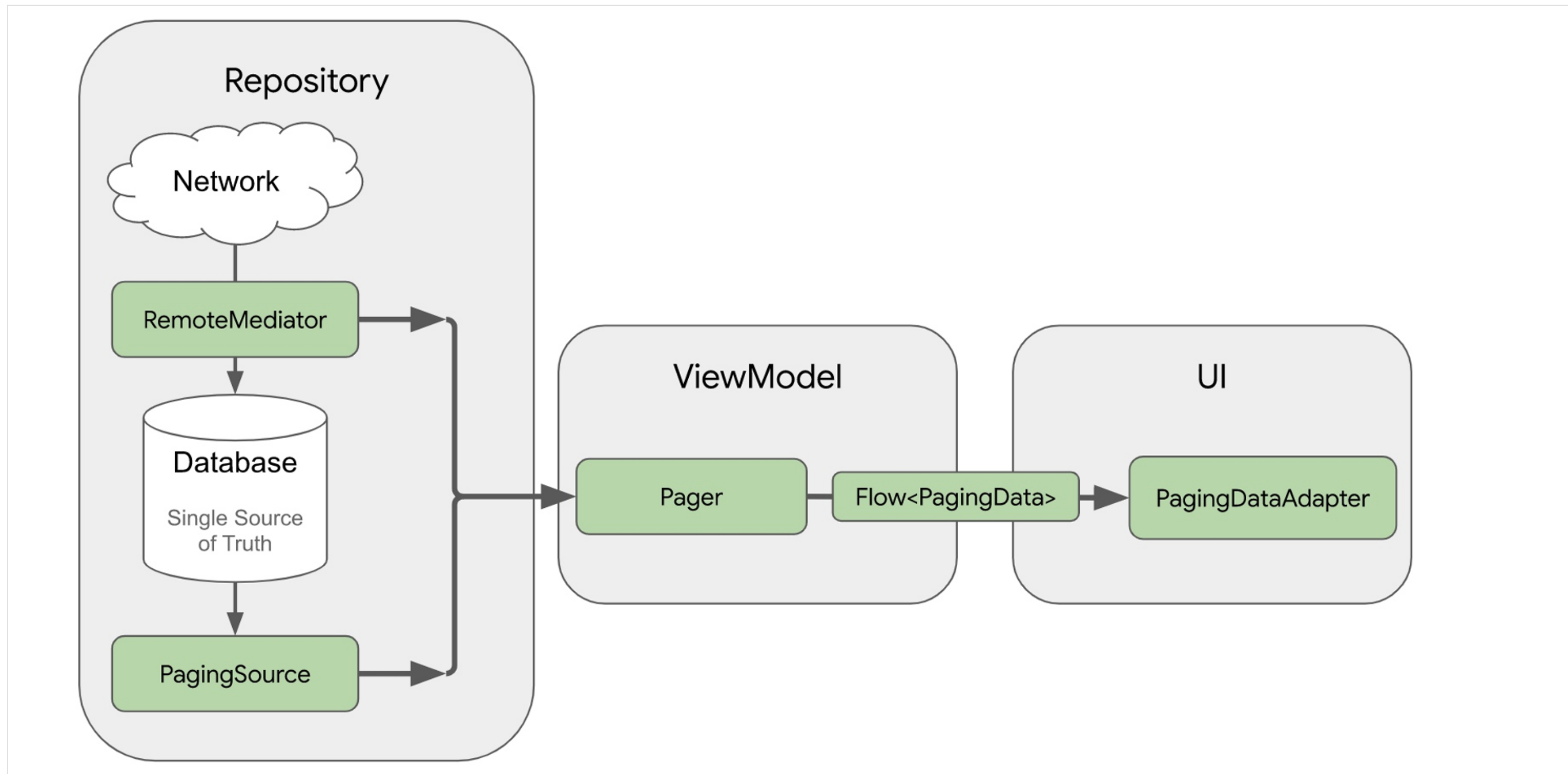
5 Coil、App Startup

- 负责托管对象与对象之间的注入关系。



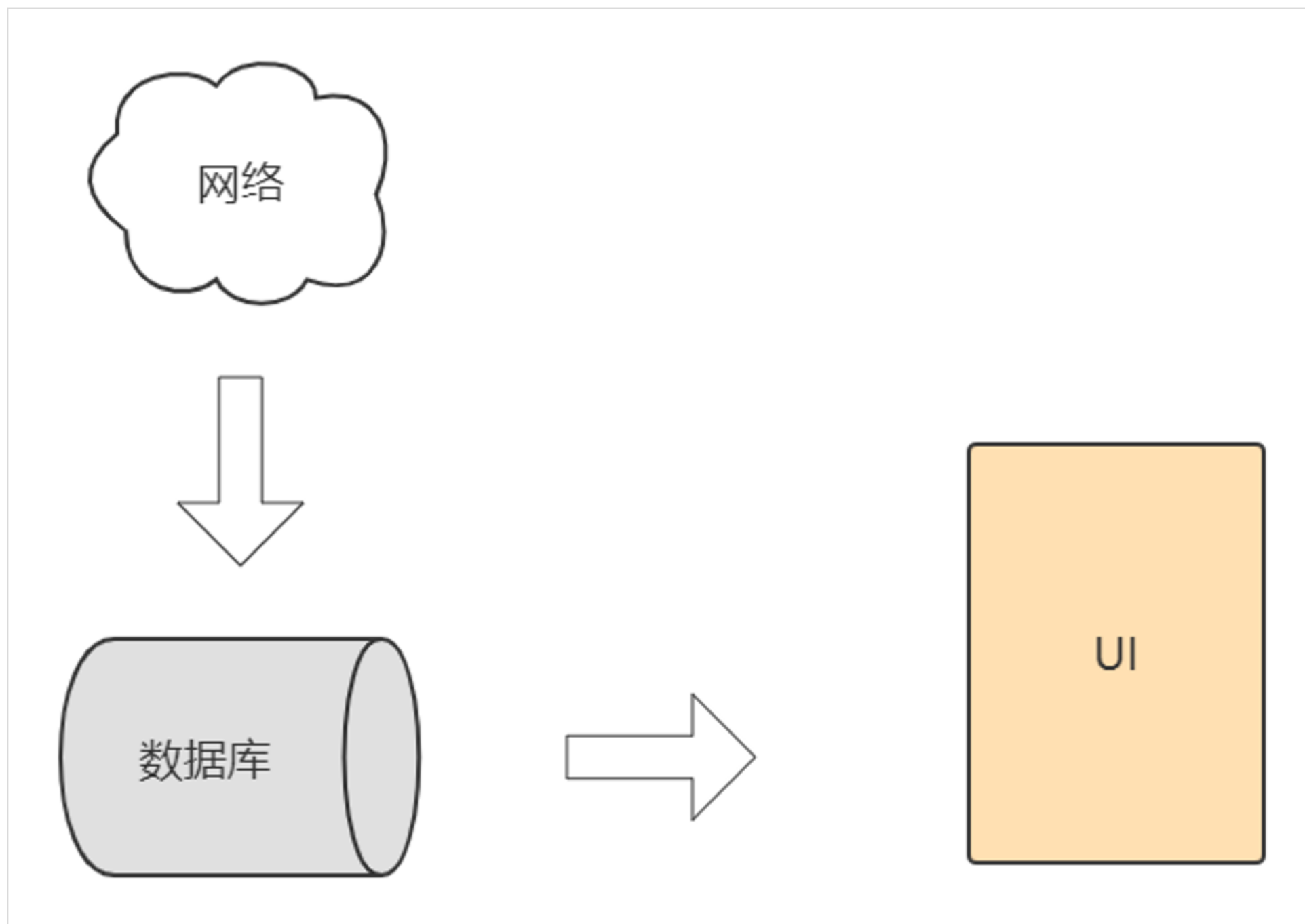
- **@HiltAndroidApp**: 触发Hilt的代码生成。
- **@AndroidEntryPoint**: 创建一个依赖容器，该容器遵循Android类的生命周期。
- **@Module**: 告诉 Hilt 如何提供不同类型的实例。
- **@InstallIn**: Install 用来告诉 Hilt 这个模块会被安装到哪个组件上。
- **@Provides**: 告诉Hilt如何获得具体实例。
- **@Singleton**: 单例。
- **@ViewModelInject**: 通过构造函数，给ViewModel注入实例。

Paging3架构



数据的流向

- 多数据源变成单一数据源。



PagingSource

- **Key**: 分页标识类型，如页码，则为Int。
- **Value**: 返回列表元素的类型。

```
abstract class PagingSource<Key : Any, Value : Any> {
```

RemoteMediator

- RemoteMediator 和 PagingSource 相似，都需要覆盖 load() 方法，但是不同的是 RemoteMediator 不是加载分页数据到 RecyclerView 列表上，而是**获取网络分页数据并更新到数据库中。**

Room支持

- 如果使用的是Room，从 2.3.0-alpha 开始，它将默认为您实现 PagingSource。
在定义 Dao 接口的 Query 语句时，返回类型要使用 PagingSource 类型。同时不需要在 Query 里指定页数和每页展示数量，页数由 PagingSource 来控制，每页数量在 PagingConfig 中定义。

RemoteMediator

- 第一步，判断 LoadType
- 第二步，请求网络分页数据
- 第三步，插入数据库。



LoadType

➤ LoadType 是一个枚举类，里面定义了三个值，如下所示：

类名	作用
LoadType.Refresh	在初始化刷新的使用，首次访问 或者调用 PagingDataAdapter.refresh() 触发
LoadType.Append	在加载更多的时候使用，需要注意的是当 LoadType.REFRESH 触发了，LoadType.PREPEND 也会触发
LoadType.Prepend	在当前列表头部添加数据的时候使用

PagingState

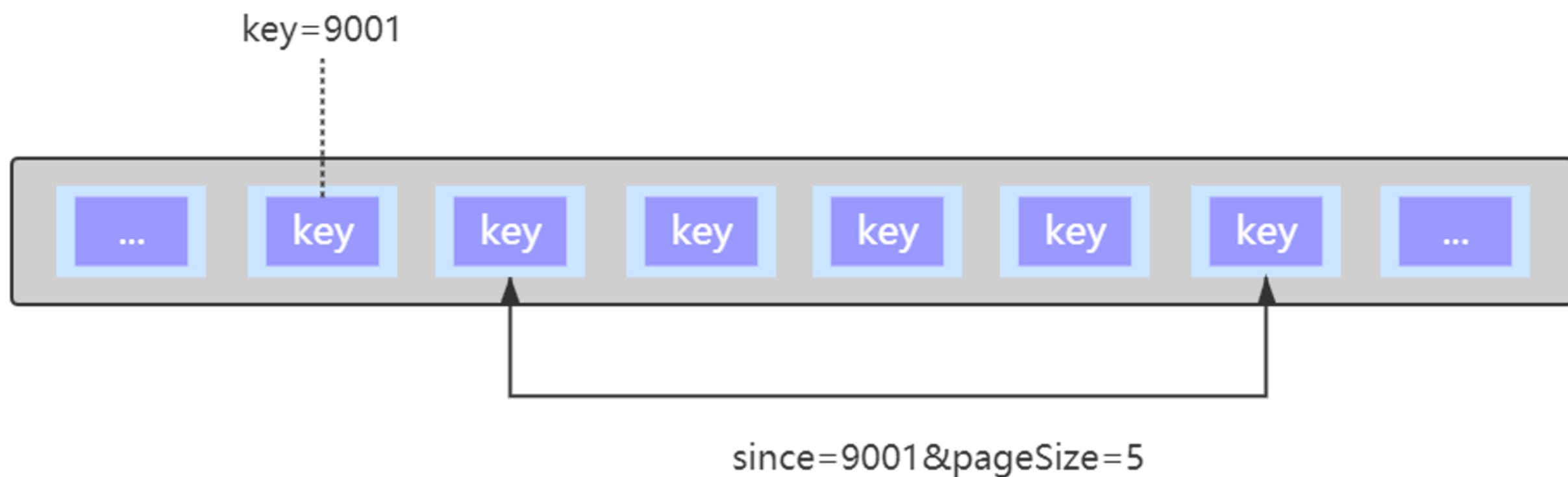
- **pages**: `List<Page<Key, Value>>` 返回的上一页的数据，主要用来获取上一页最后一条数据作为下一页的开始位置。
- **config**: `PagingConfig` 返回的初始化设置的 `PagingConfig` 包含了 `pageSize`、`prefetchDistance`、`initialLoadSize` 等等

MediatorResult

- 请求出现错误，返回 `MediatorResult.Error(e)`。
- 请求成功且有数据，返回 `MediatorResult.Success(endOfPaginationReached = true)`。
- 请求成功但是没有数据，返回 `MediatorResult.Success(endOfPaginationReached = false)`。

分页逻辑

- 适用于当目标数据的**下一页需要依赖于上一页**数据中最后一个对象中的某个字段作为key的情况，此类分页形式常见于评论功能的实现，例如，若上一页数据中最后一个对象的key为9001，那么在请求下一页时，需要携带参数`since=9001&pageSize=5`，则服务器会返回key=9001之后的5条数据。



分页逻辑

➤ API接口

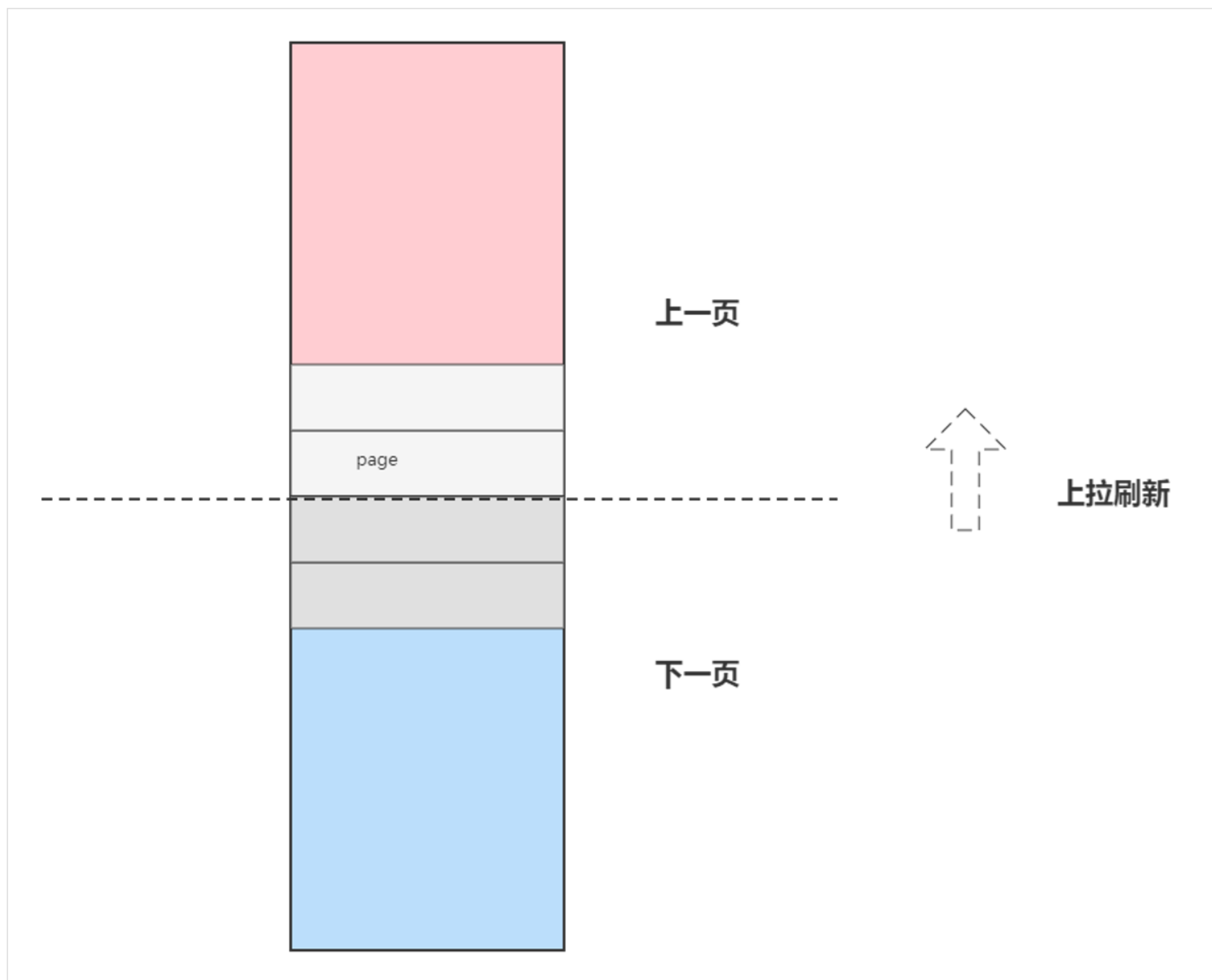
- http://192.168.0.117:8080/pagingserver_war/ikds.do?since=0&pagesize=8

➤ 接口返回的数据

```
[
  {
    "id": 35076714,
    "title": "扎克·施奈德版正义联盟",
    "cover": "https://img9.doubanio.com/xxx.webp",
    "rate": "8.9"
  },
  {
    ...
  },
  .
  .
  .
]
```

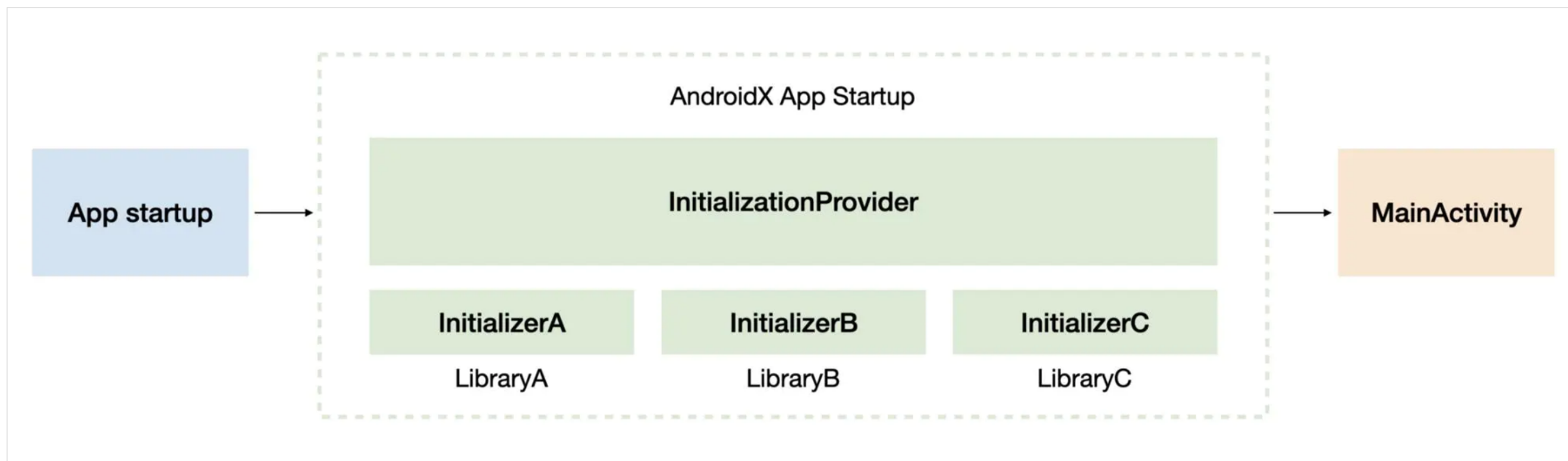
分页逻辑

➤ 如何计算得到下一个页的Page值



App Startup

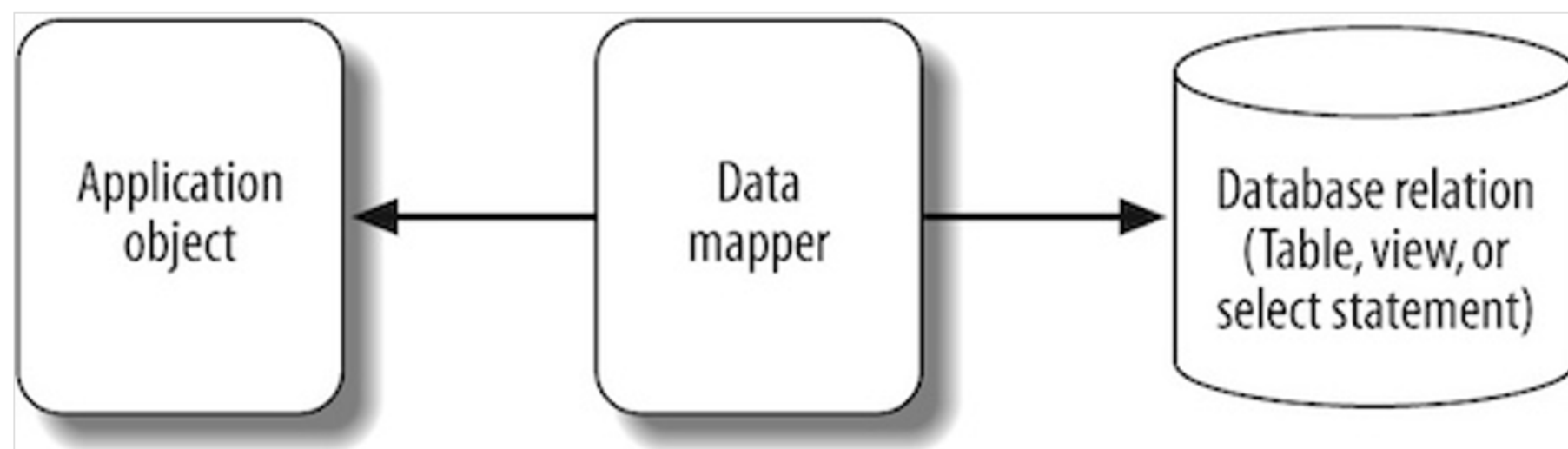
- App Startup 是 Android Jetpack 最新成员，提供了在 App 启动时初始化组件简单、高效的方法，无论是 library 开发人员还是 App 开发人员都可以使用 App Startup 显示的设置初始化顺序。



- **性能优秀**
- **体积较小**：其包体积与Picasso相当，显著低于Glide和Fresco，仅仅只有1500个方法，但是在功能上却不输于其他同类库
- **简单易用**：配合Kotlin扩展方法等语法优势，API简单易用
- **技术先进**：基于Coroutine、OkHttp、Okio、AndroidX等先端技术开发，确保了技术上的先进性
- **丰富功能**：缓存管理（MemCache、DiskCache）、动态采样（Dynamic image sampling）、加载中暂停/终止等功能有助于提高图片加载效率

Data Mapper

- 使用 Data Mapper 分离数据源的 Model 和 页面显示的 Model，不要因为数据源的增加、修改或者删除，导致上层页面也要跟着一起修改。



Data Mapper

- CarBrandEntity与CarBrandItemModel互相转换。

