Android:

1. There is never a need to save a context object.

       Out of convenience, save views since findViewById is an expensive call

       By not saving, we avoid memory leaks

       If code needs a context, the programmer is forced to know where in the lifecycle the context can be referenced

2. Actions that require a context must be handled by an Android managed object

       See first tenet

3. Never utilize an Android managed object outside of its lifecycle

RxJava

- Used to chain callbacks without passing a reference
- Do not use to cache values (BehaviorSubject)
- Do not use to manage threading

Dagger Injection

- Used to instantiate or retrieve classes from getters.
- In an android managed class, injection must occur after onCreate
- Do not use scoping.
- Used for singletons and instantiating a new instance of the class

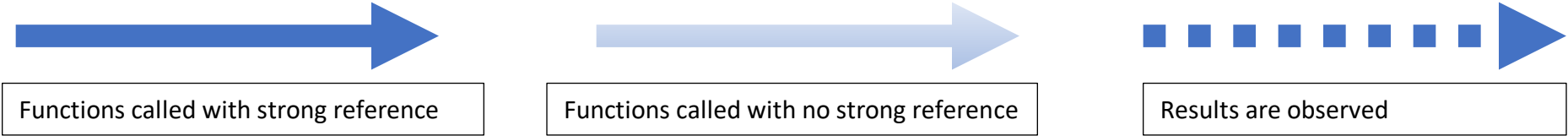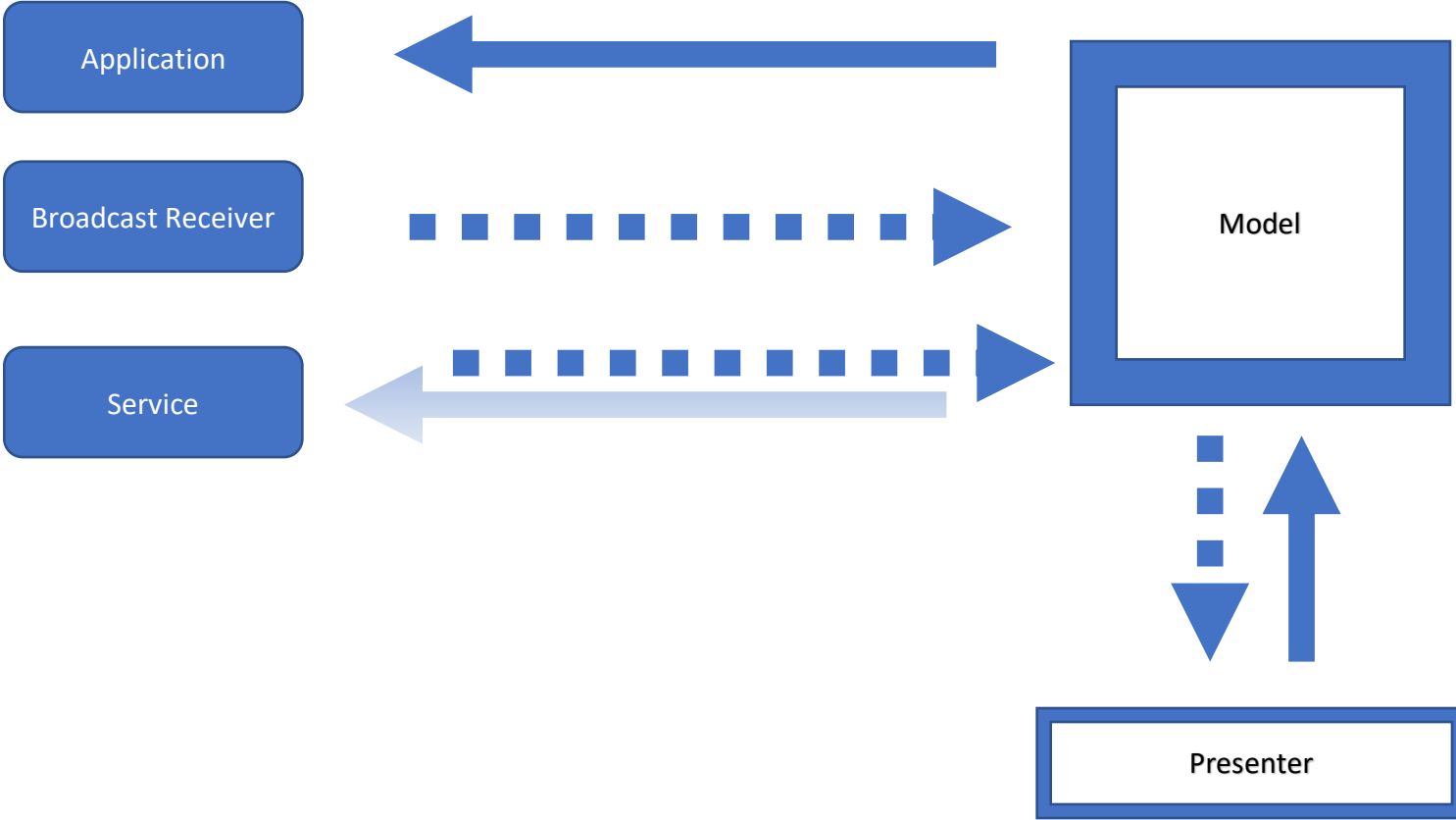| Application | | Model |
|---|---|---|
| Broadcast Receiver | | Presenter |
| Fragment | | View Wrapper |
| Activity | | RecyclerView.Adapter |
| Service | | RecyclerView.ViewHolder |
| View | | |

Functions called with strong reference
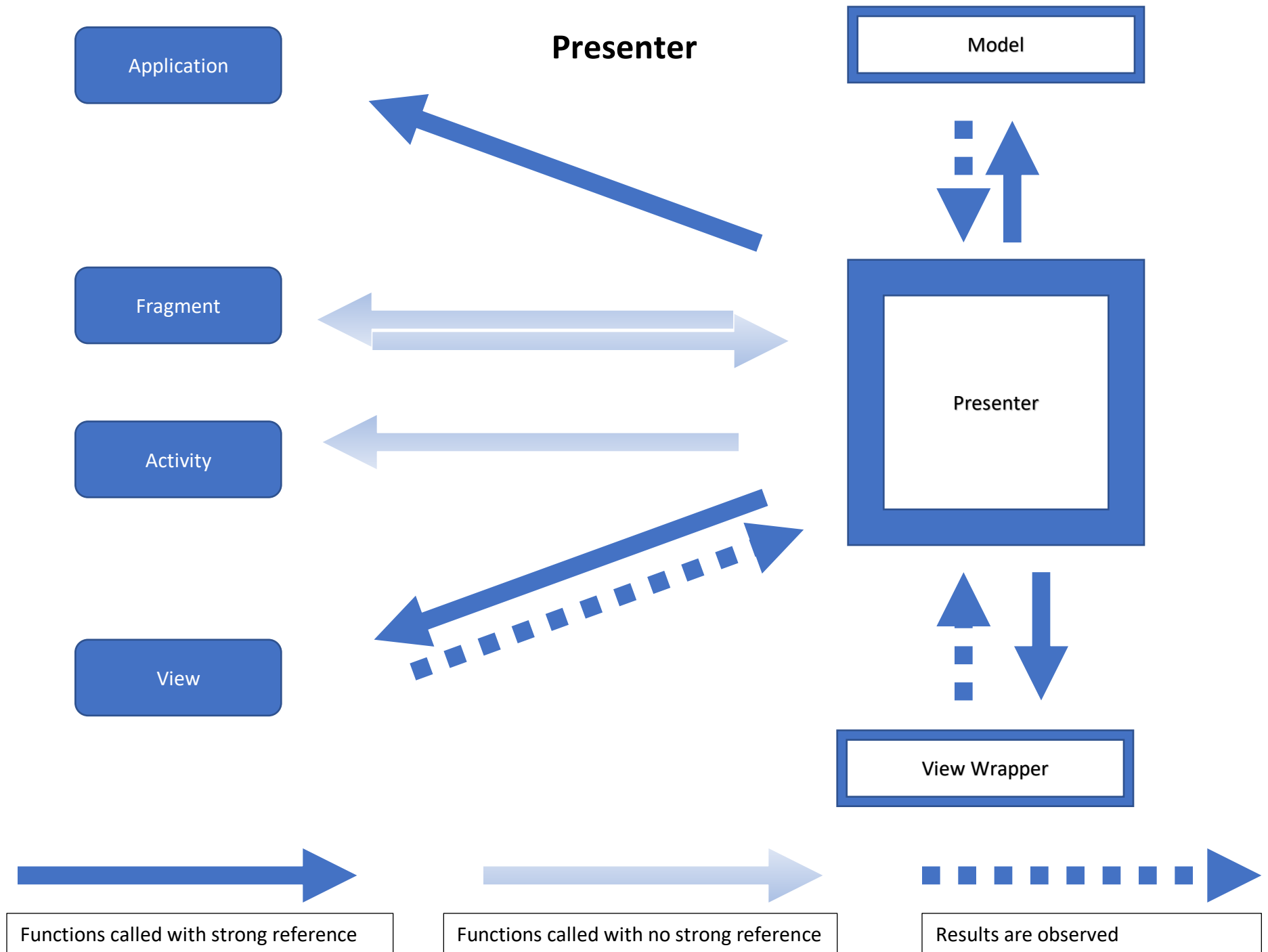
Functions called with no strong reference, using intent or supplier

Results are observed. A strong reference may exist only through a subscription.

# Model



| Application | | Model |
| --- | --- | --- |
| Broadcast Receiver | | |
| Service | | |
| | | Presenter |

Functions called with strong reference

Functions called with no strong reference

Results are observed

**Presenter**

Application

Model

Fragment

Activity

Presenter

View

View Wrapper

Functions called with strong reference

Functions called with no strong reference

Results are observed

**View Wrapper**

Presenter

View

View Wrapper

RecyclerView.Adapter

RecyclerView.ViewHolder

Model

Functions called with strong reference

Functions called with no strong reference

Results are observed