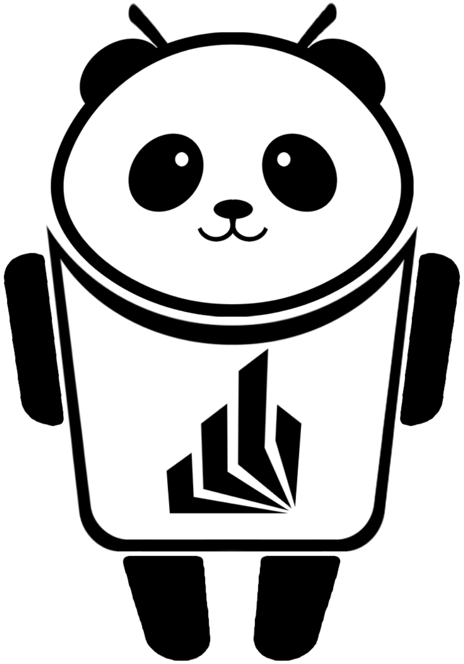


PAndA^2

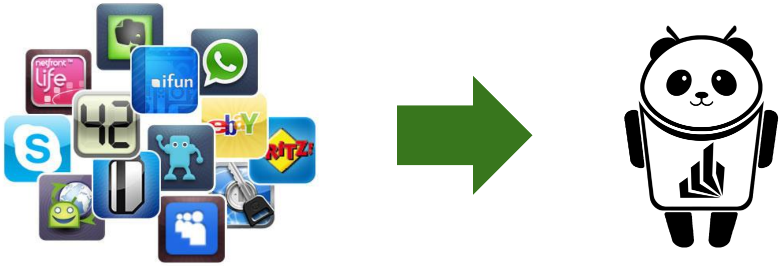
Paderborn **And**roid **A**pp **A**nalysis

Outline

- Introduction to PAndA^2
- Component Architecture
- Architecture Details
  - General Analysis
  - Datastructure
    - Enhancer
    - GraphGenerator
    - Analyzer
  - 3 Analysis Levels
- Milestones
  - Status
  - Prototype demo
  - Soot

Introduction to PAndA²

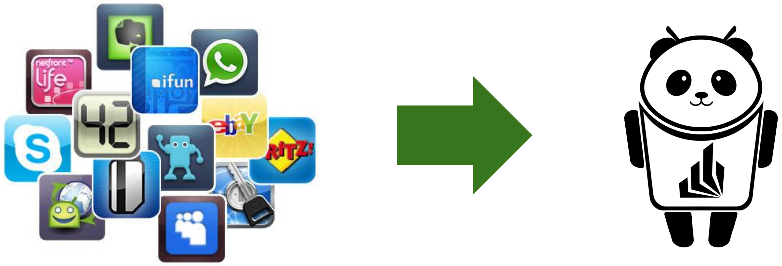
- Tool for analyzing Android Apps



Introduction to PAndA^2

- Tool for analyzing Android Apps

- 3 Analysis Levels:
    - Level 1: Intra-App Resource Usage
    - Level 2a: Intra-App Information Flow Control
    - Level 2b: Inter-App Resource Usage

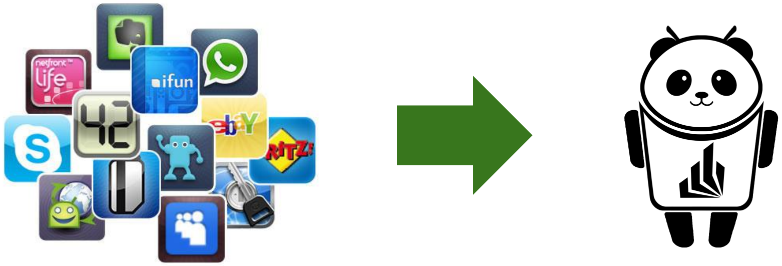


Introduction to PAndA^2

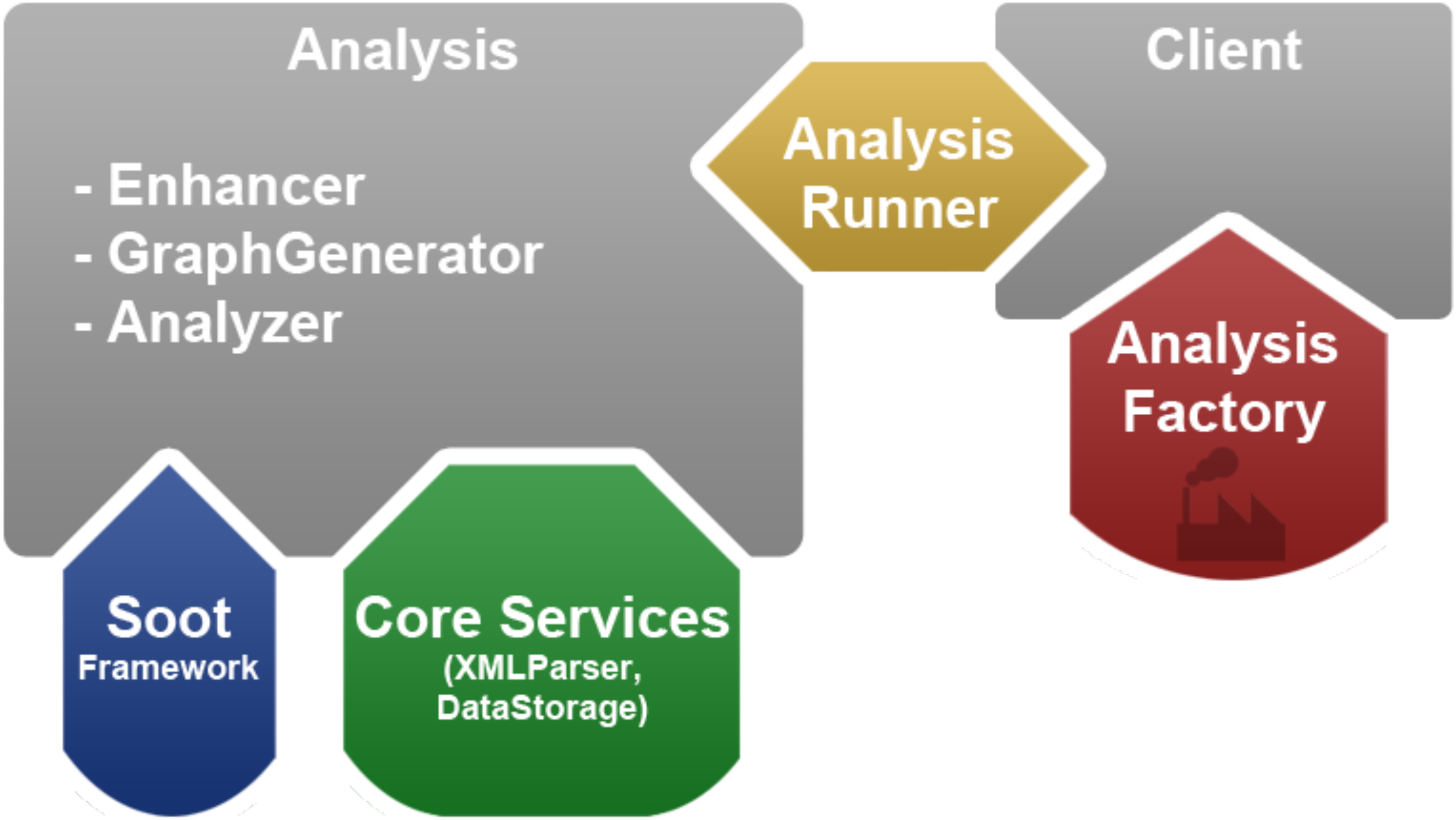
- Tool for analyzing Android Apps

- 3 Analysis Levels:
  - Level 1: Intra-App Resource Usage
  - Level 2a: Intra-App Information Flow Control
  - Level 2b: Inter-App Resource Usage

Extension

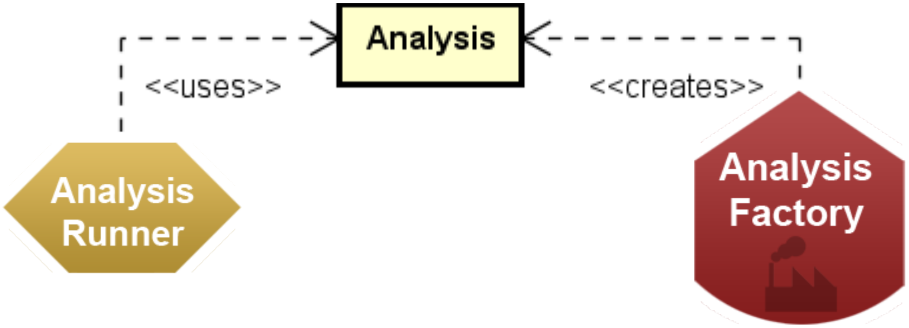


Component Architecture



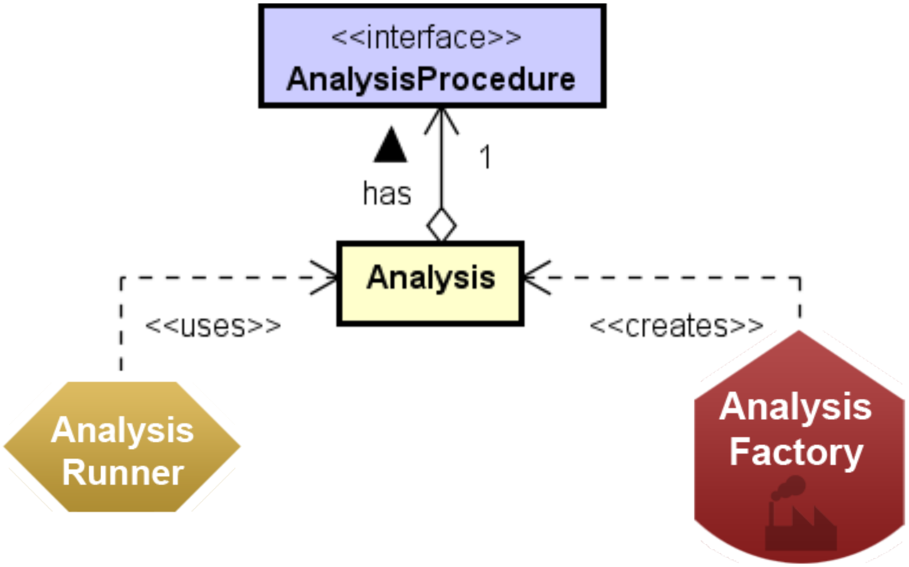
Architecture Details: Analysis

- Created by **AnalysisFactory**
- Executed by **AnalysisRunner**

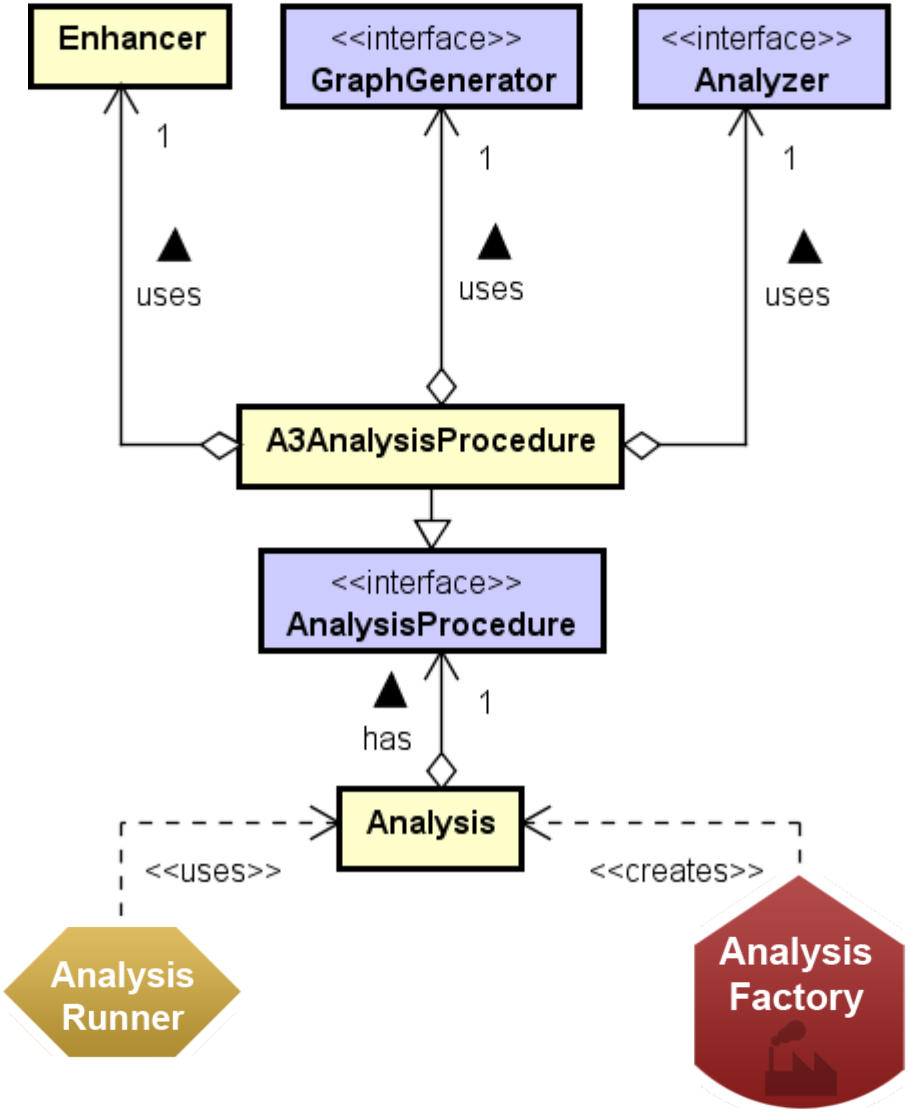


Architecture Details: Analysis

- Created by **AnalysisFactory**
- Executed by **AnalysisRunner**
- **Strategy pattern** to ensure extendability (different analyses)

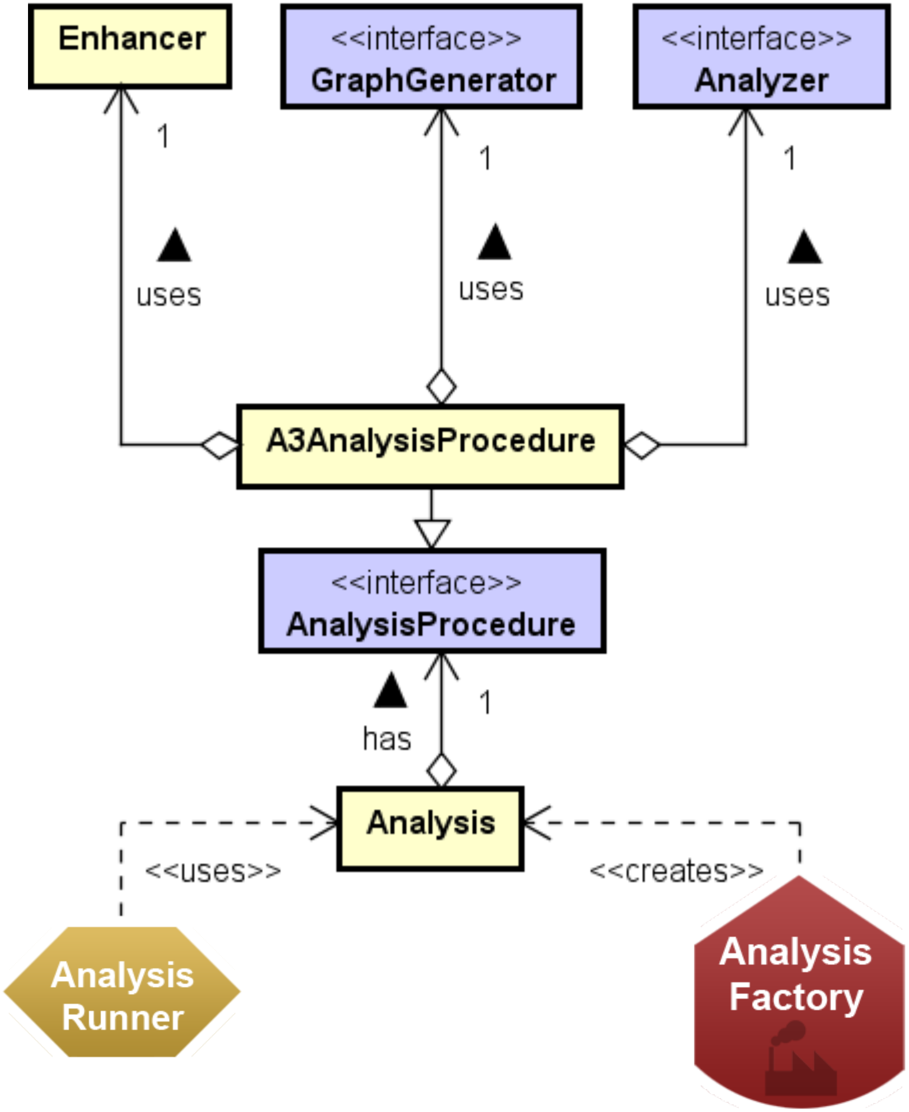


Architecture Details: Analysis



- Created by **AnalysisFactory**
- Executed by **AnalysisRunner**
- **Strategy pattern** to ensure extendability (different analyses)
- **A3AnalysisProcedure**:
  - **Enhancer** (general)
  - **GraphGenerator** (specific)
  - **Analyzer** (specific)

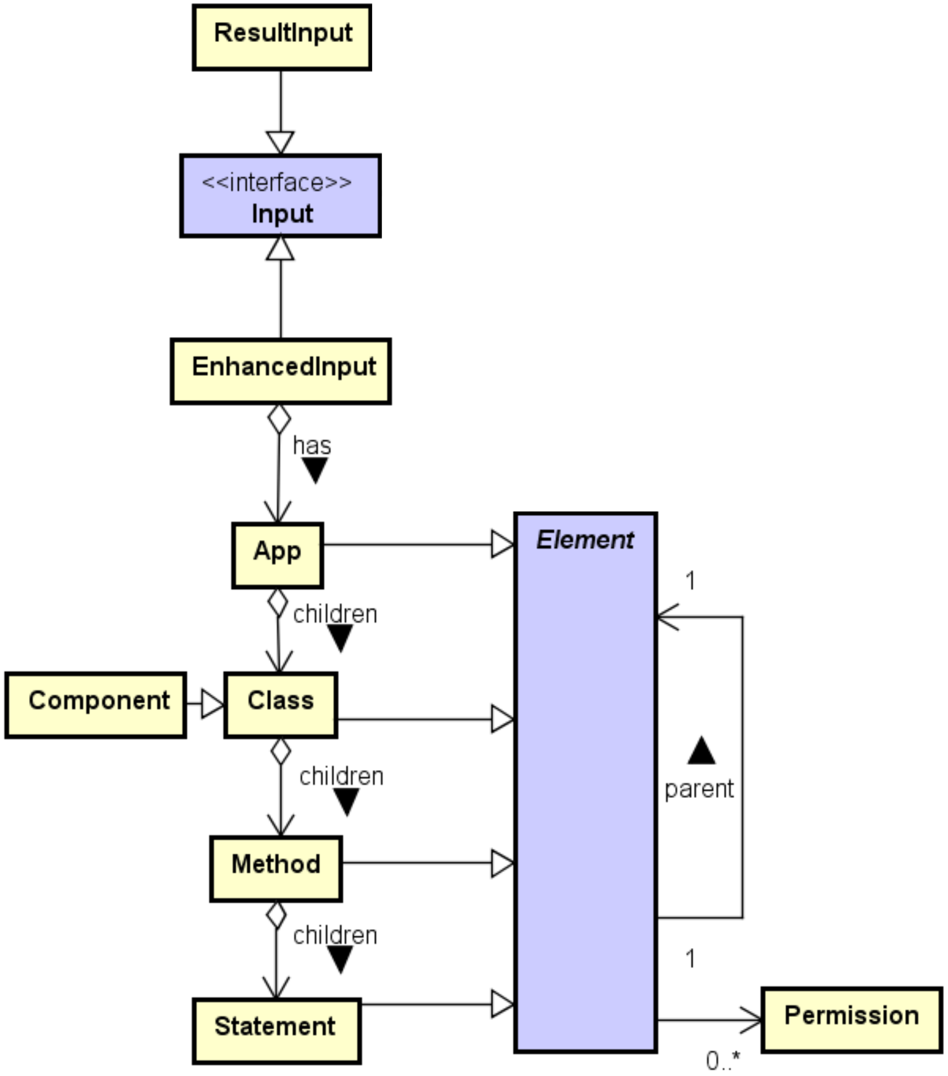
Architecture Details: Analysis



- Created by **AnalysisFactory**
- Executed by **AnalysisRunner**
- **Strategy pattern** to ensure extendability (different analyses)
- **A3AnalysisProcedure**:
  - **Enhancer** (general)
  - **GraphGenerator** (specific)
  - **Analyzer** (specific)

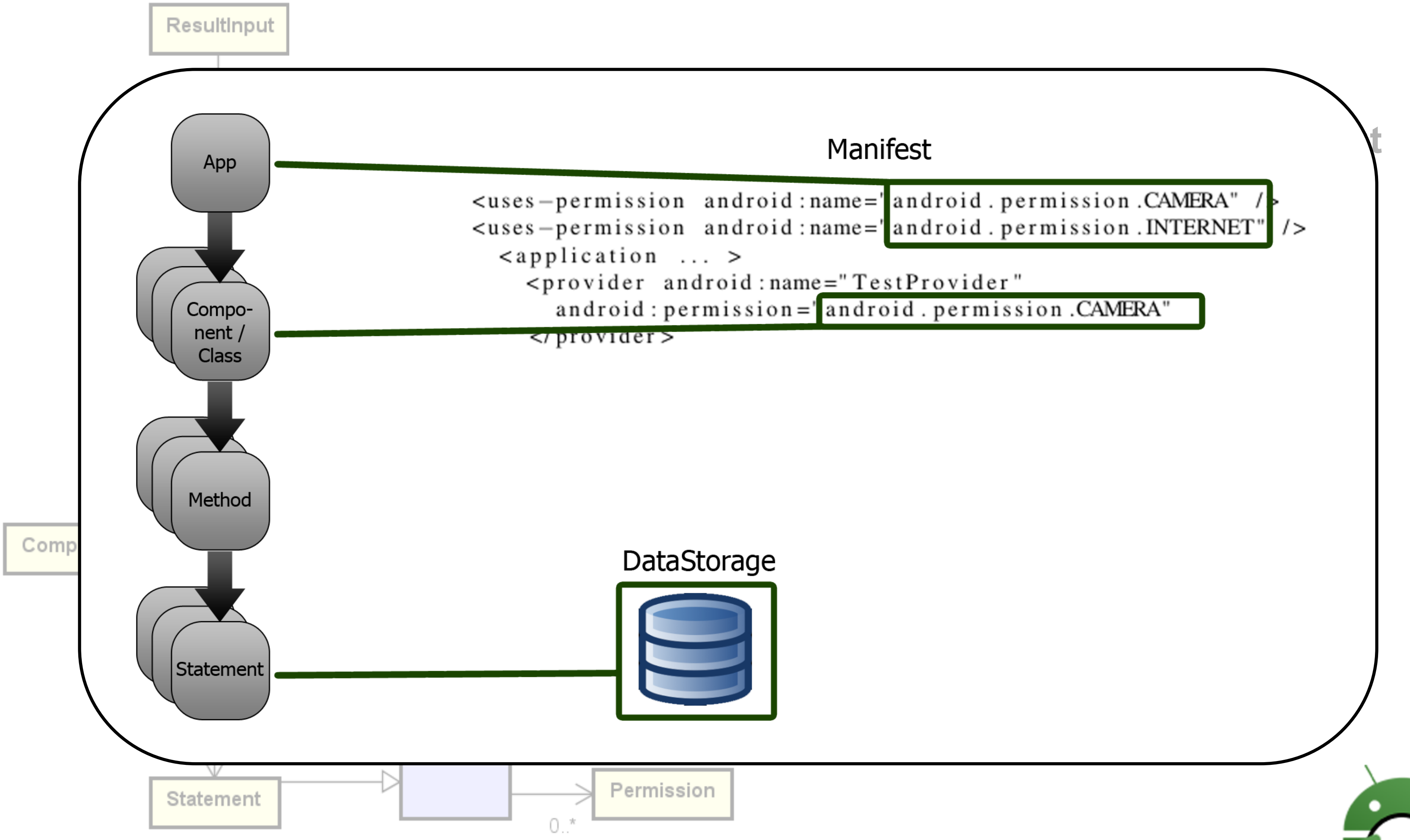
➡ Common intermediate representations needed

Architecture Details: Data Structure



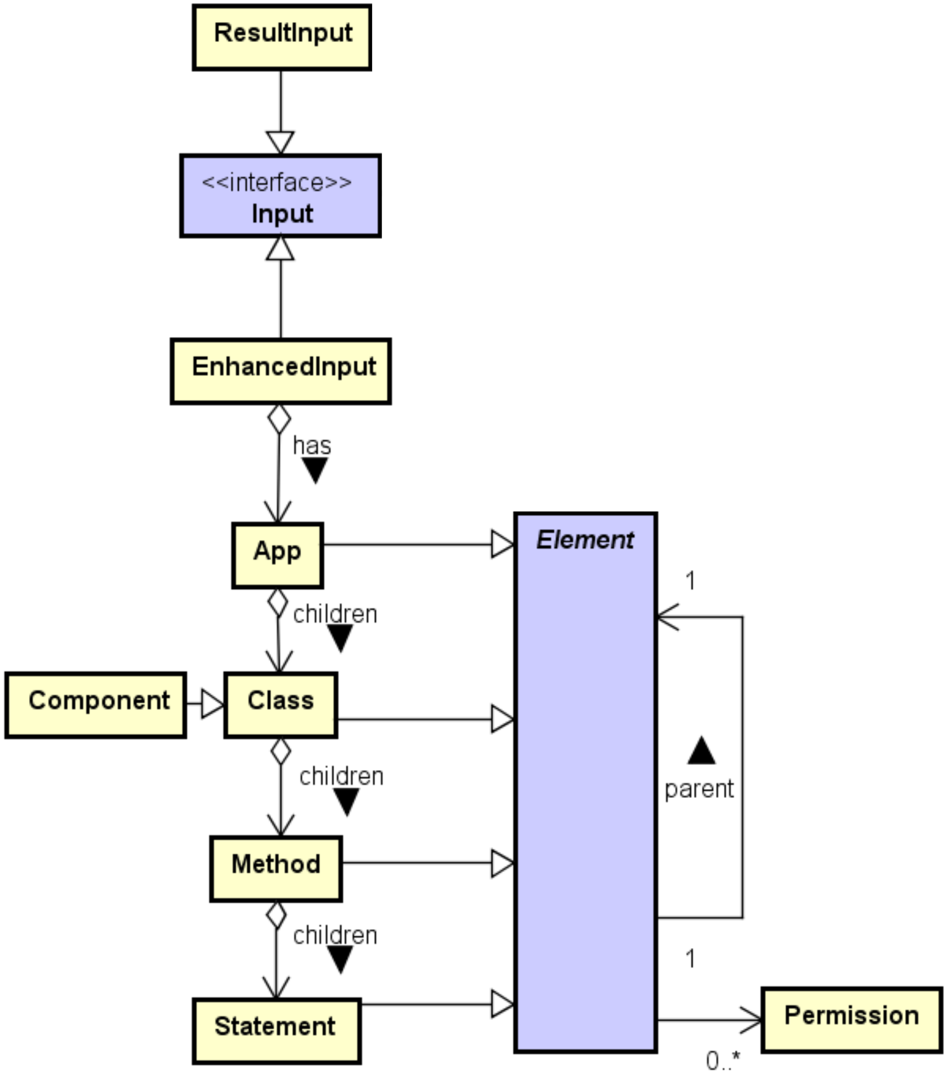
- **Enhancer** creates **Input**

Architecture Details: Data Structure



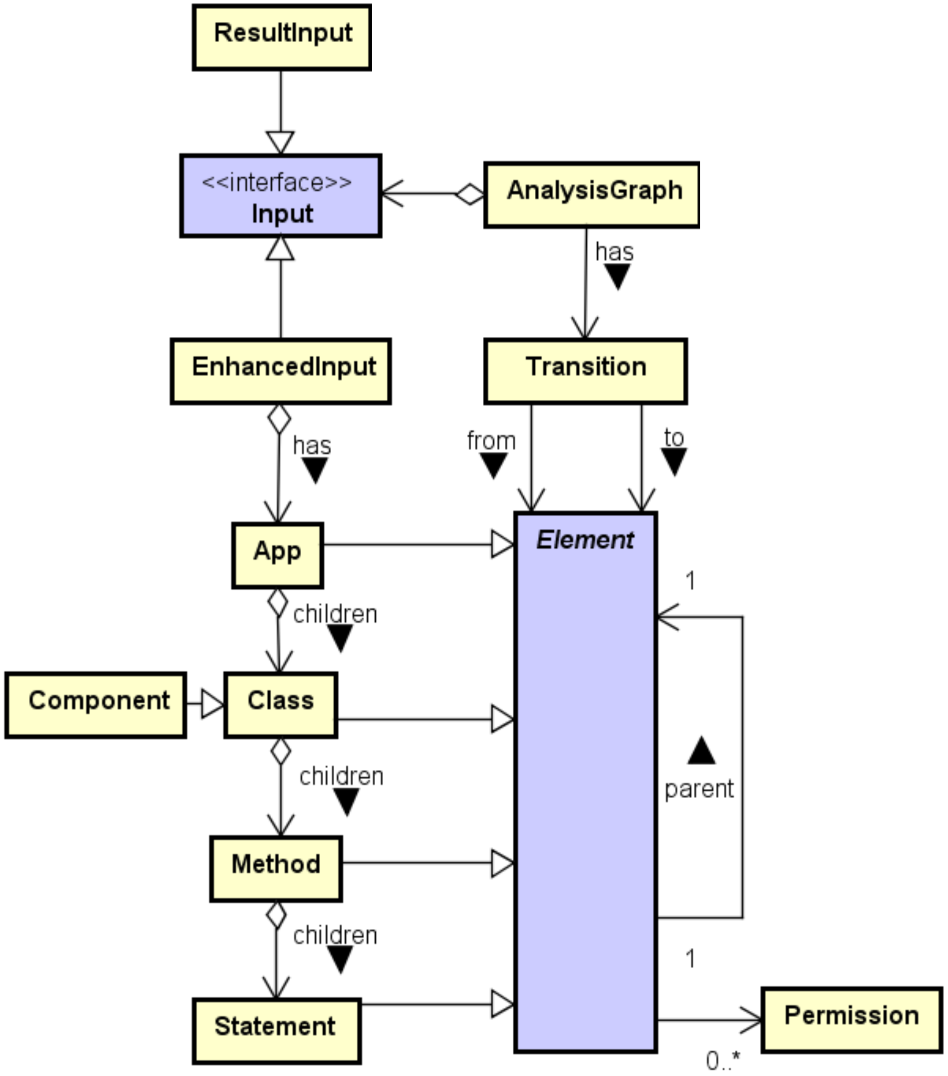
```
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.INTERNET" />
  <application ... >
    <provider android:name="TestProvider"
      android:permission="android.permission.CAMERA"
    </provider>
```

Architecture Details: Data Structure



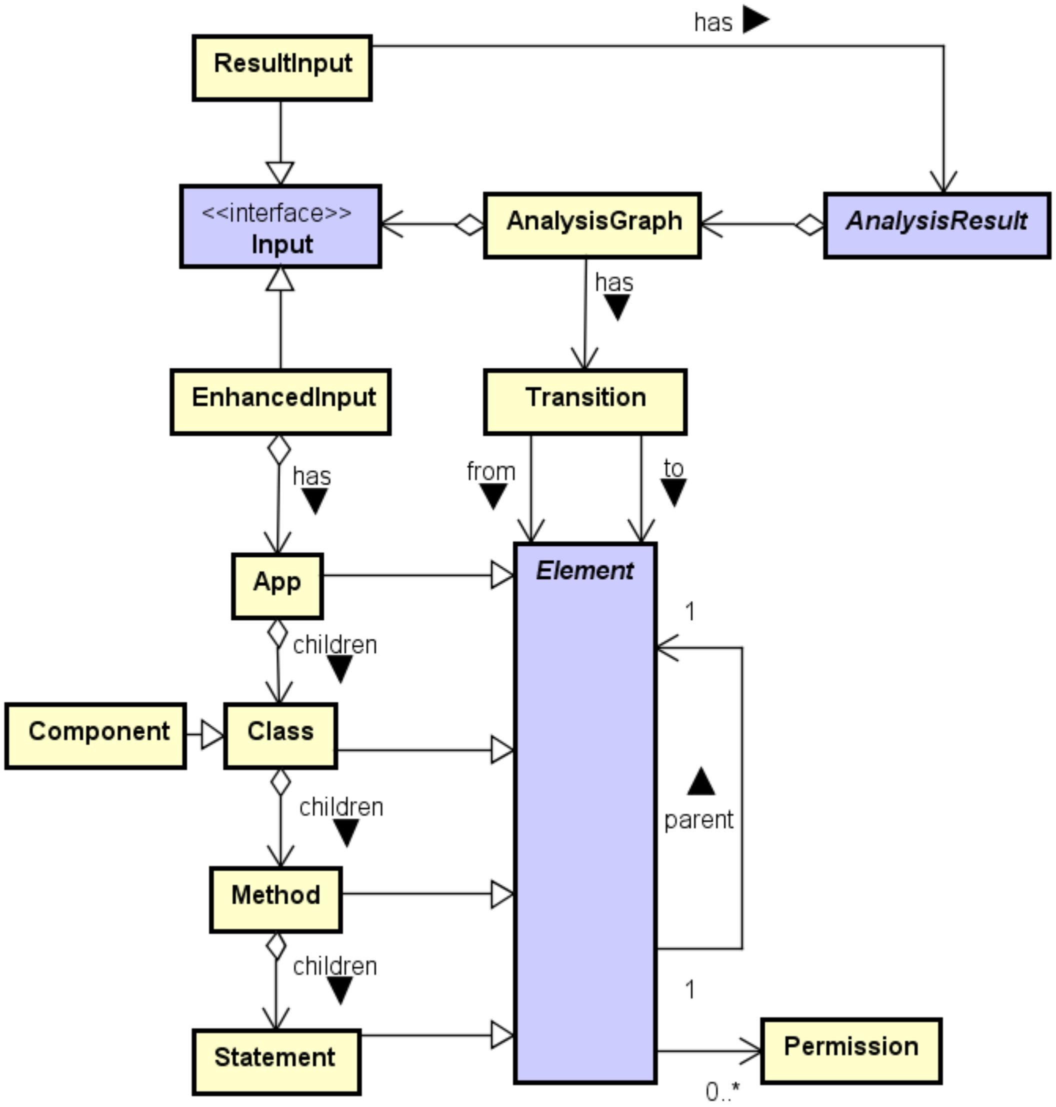
- **Enhancer** creates **Input**

Architecture Details: Data Structure



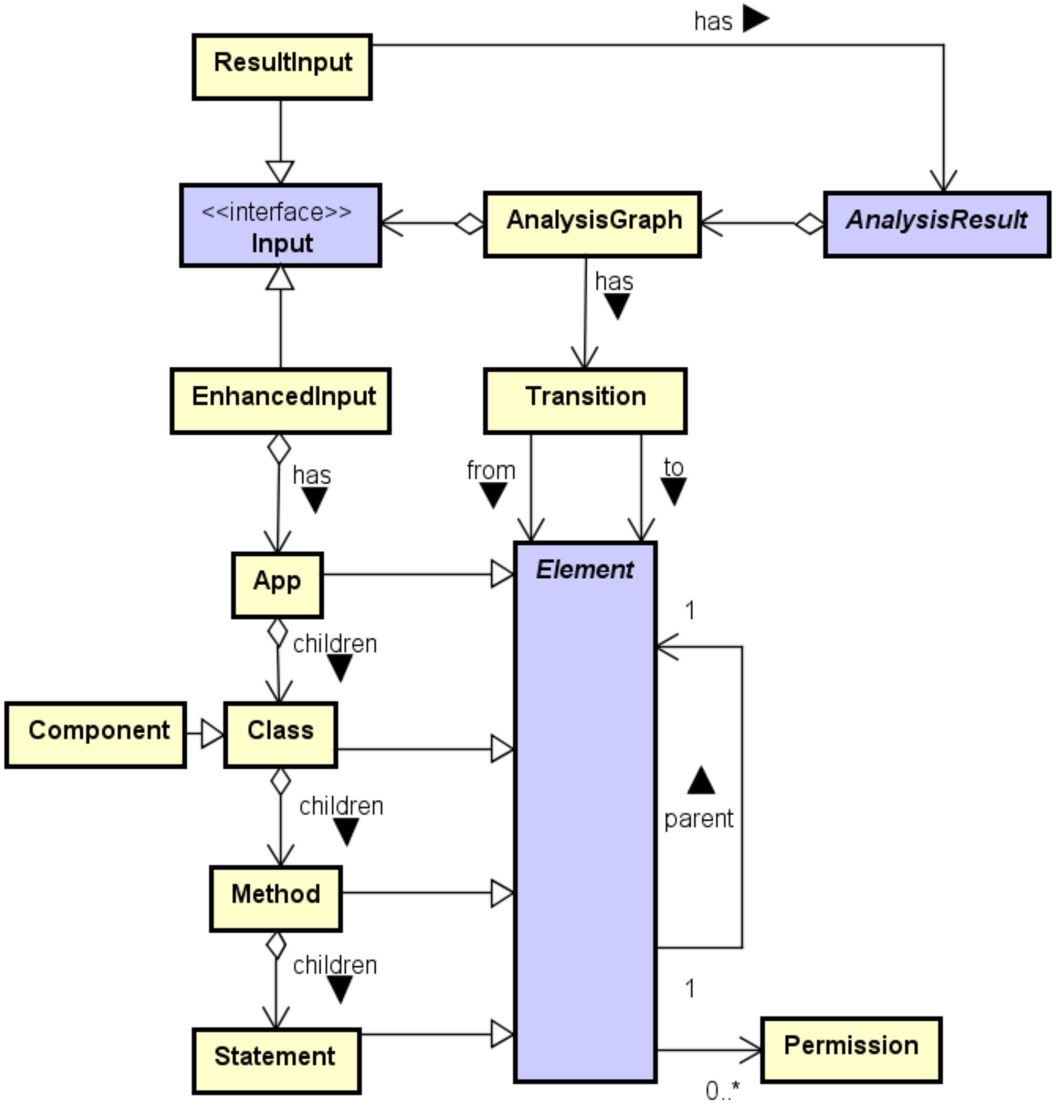
- **Enhancer** creates **Input**
- **GraphGenerator** creates **AnalysisGraph**

Architecture Details: Data Structure



- **Enhancer** creates **Input**
- **GraphGenerator** creates **AnalysisGraph**
- **Analyzer** creates **AnalysisResult**

Architecture Details: Data Structure



- **Enhancer** creates **Input**
- **GraphGenerator** creates **AnalysisGraph**
- **Analyzer** creates **AnalysisResult**
  - **HTML5 (textual)**
  - **Dot (graphical)**

Architecture Details: Level 1

- **Enhancer**

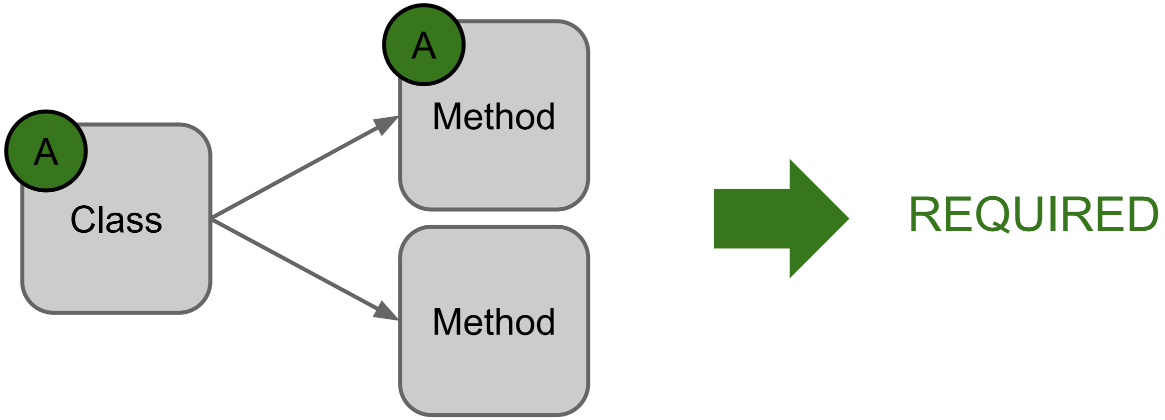
- **GraphGeneratorLvl1**
  - Adding Transitions for explicit Intents.

- **AnalyzerLvl1**
  - Filling up remaining permissions
  - Assigning permission groups to Elements&Permissions:
    REQUIRED
    MAYBE_REQUIRED
    UNUSED
    MAYBE_MISSING
    MISSING

- **AnalysisResultLvl1**

Architecture Details: Level 1

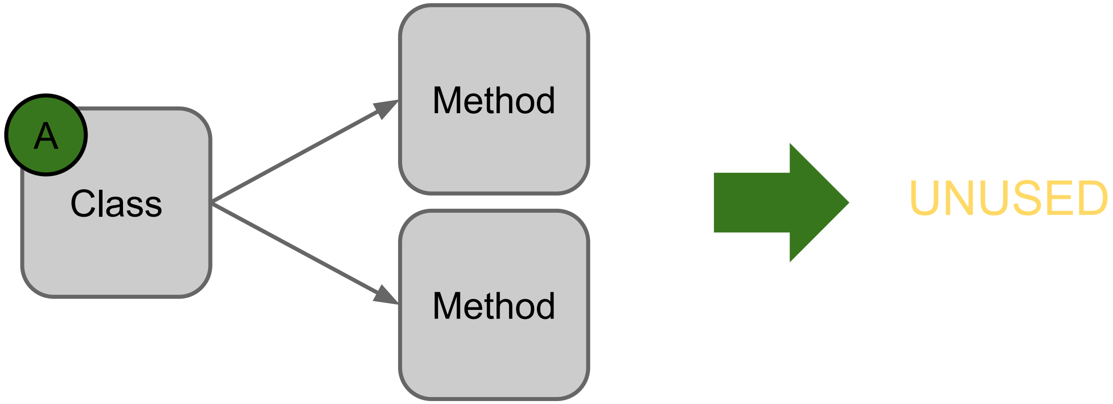
- **Permission** Ⓐ **assigned to *visited* Element?** ✓
- **Permission** Ⓐ **assigned to any child?** ✓
- **Element in maybeMore list?** ✗



REQUIRED

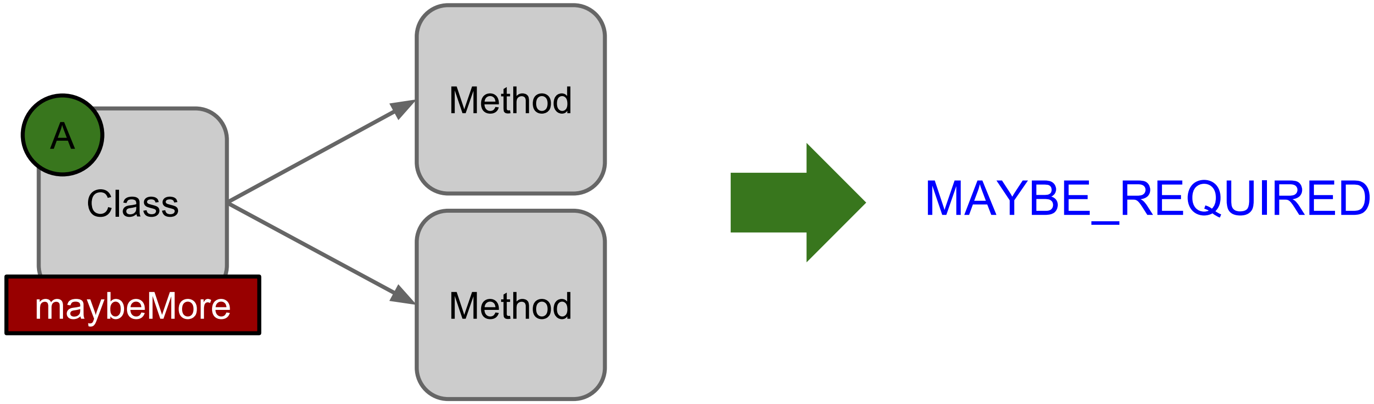
Architecture Details: Level 1

- **Permission** Ⓐ **assigned to *visited* Element?** ✓
- **Permission** Ⓐ **assigned to any child?** ✗
- **Element in maybeMore list?** ✗



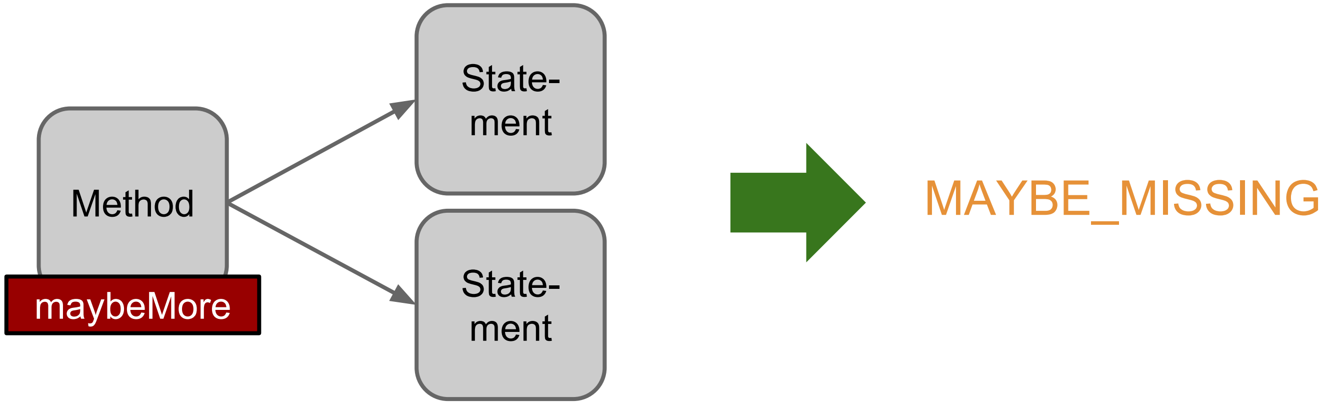
Architecture Details: Level 1

- **Permission** A **assigned to *visited* Element?** ✓
- **Permission** A **assigned to any child?** ✗
- **Element in maybeMore list?** ✓



Architecture Details: Level 1

- **Permission** A **assigned to *visited* Element?** ✗
- **Permission** A **assigned to any child?** ✗
- **Element in maybeMore list?** ✓



Architecture Details: Level 1

- **Enhancer**
- **GraphGeneratorLvl1**
    - Adding Transitions for explicit Intents.
- **AnalyzerLvl1**
    - Filling up remaining permissions
    - Assigning permission groups to Elements&Permissions:

        REQUIRED
        MAYBE_REQUIRED
        UNUSED
        MAYBE_MISSING
        MISSING
- **AnalysisResultLvl1**

Architecture Details: Level 2b

- **Enhancer**
  - Collects previous results
- **GraphGeneratorLvl2b**
  - Adding Transitions for implicit Intents. → Intent-Filter
- **AnalyzerLvl2b**
  - Assigning permission groups to Elements&Permissions:

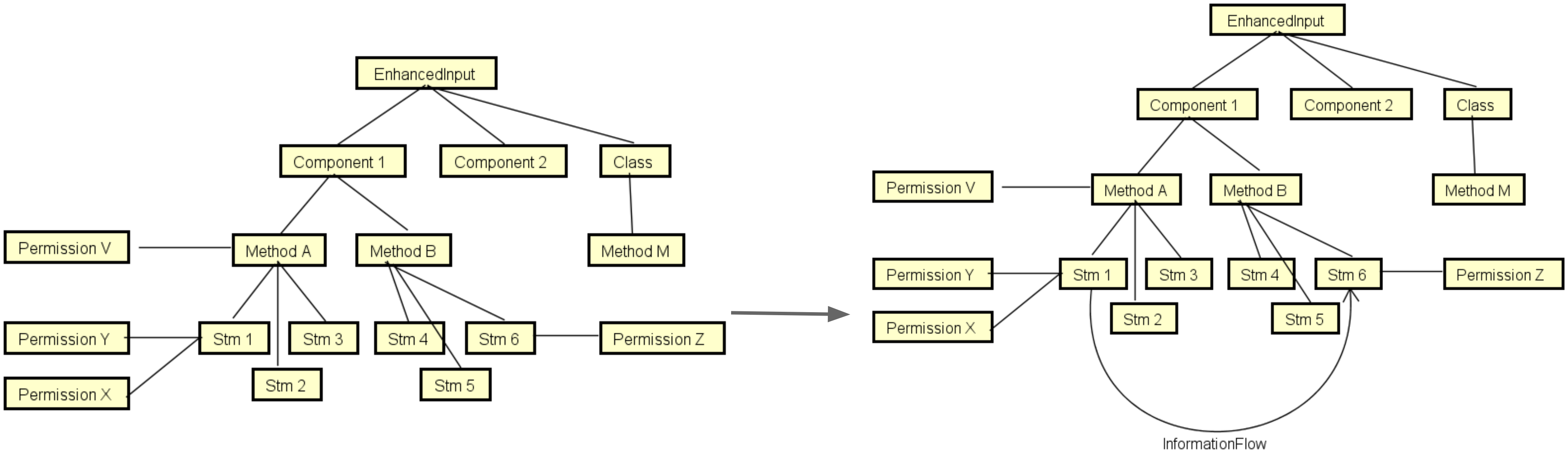
    REQUIRED
    MAYBE_REQUIRED
    UNUSED
    MAYBE_MISSING
    MISSING

    } direct / indirect
- **AnalysisResultLvl2b**

Architecture Details: Level 2a

```
...
Stm1:   v = readContactData();
...
Stm6:   upload(v);
```

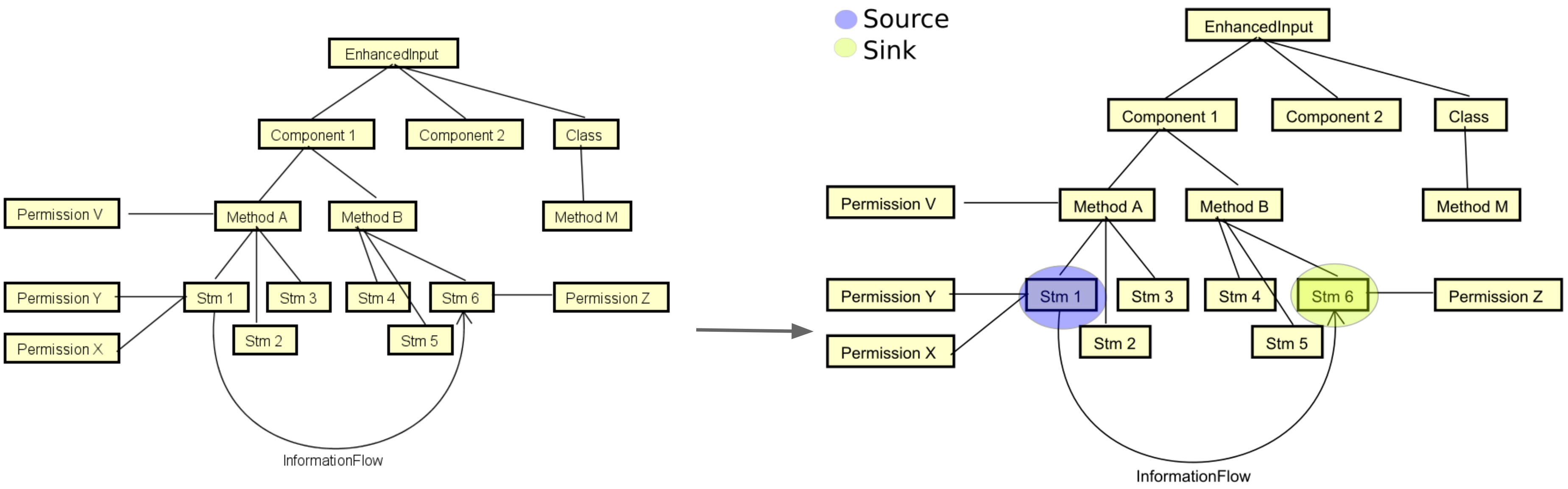
protected by permission



Architecture Details: Level 2a

```
...
Stm1:   v = readContactData();
...
Stm6:   upload(v);
```

protected by permission



Milestones

- **Milestone 1** (external - 4th September)
  - Constructed EnhancedInput
  - Basic Level 1 Analysis
  - Basic CMD-Line

- **Milestone 2** (internal - 8th October)
  - Enhancer
  - Advanced Level 1, 2a, 2b
  - Result-Loading/Storing
  - UI

- **Milestone 3** (external - 14th December)
  - Finished Level 1, 2a, 2b
  - ...

Testing

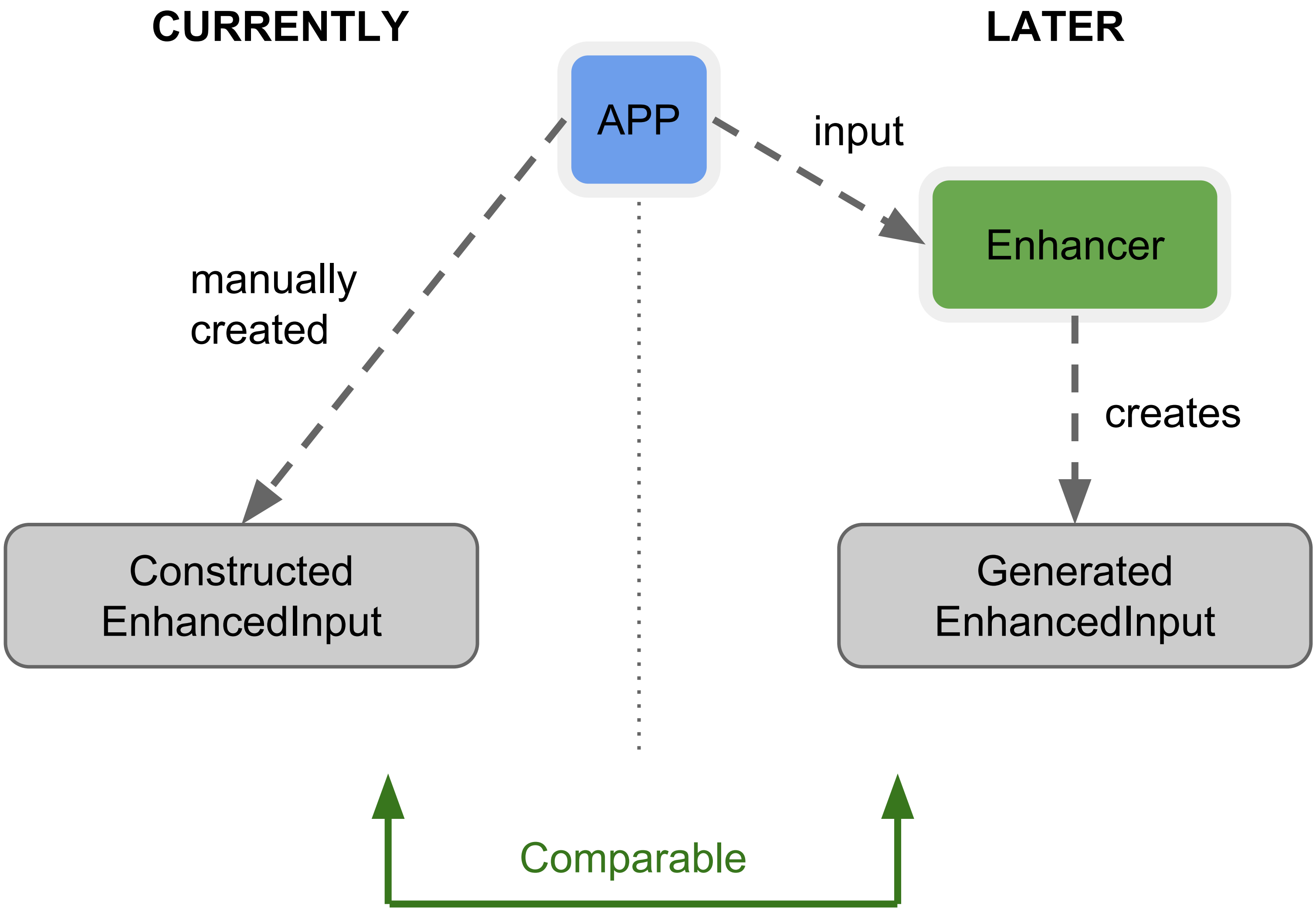
- Created dummy APPs covering different scenarios as follows:
  - Example for Level 1 including 5 groups of permissions
  - Examples for Android Components

- Prepared document describing the functionality of each APP along with the important features of the Android component that is being used.

- Prepared document describing the textual result for each APP.

- **Manually created the EnhancedInput instance for each APP.**

Testing


CURRENTLY
LATER
APP
input
Enhancer
manually created
creates
Constructed EnhancedInput
Generated EnhancedInput
Comparable

Status: User Interface

- GUI development using Java Swing.
- Implemented the basic business logic:
  - for saving the user input and for showing the textual results.
  - for the command line and showing and filtering the level 1 result.
    - Example:

```
-l level1 –m summary -i “c:\temp.apk” –r view –v textual
```

- Developed the codebase for the validation of the user input using JCommander library.

- Research to look up for a solution to display Graphical Output.

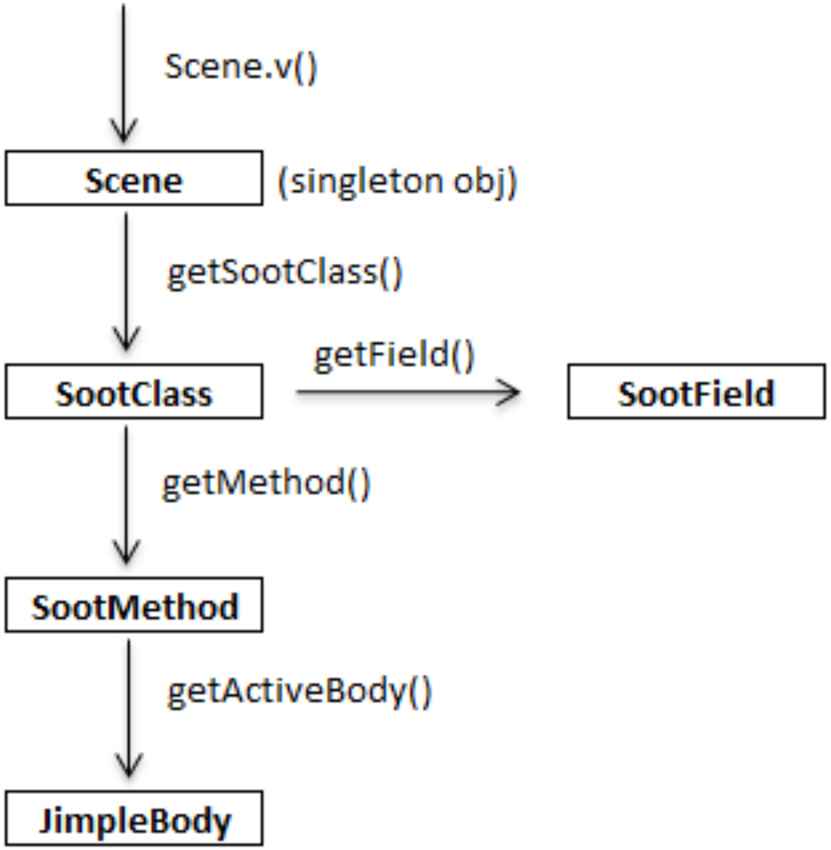
Prototype demo

Status: User Interface

```
-l level1 –m summary -i “c:\temp.apk” –r view –v textual
```

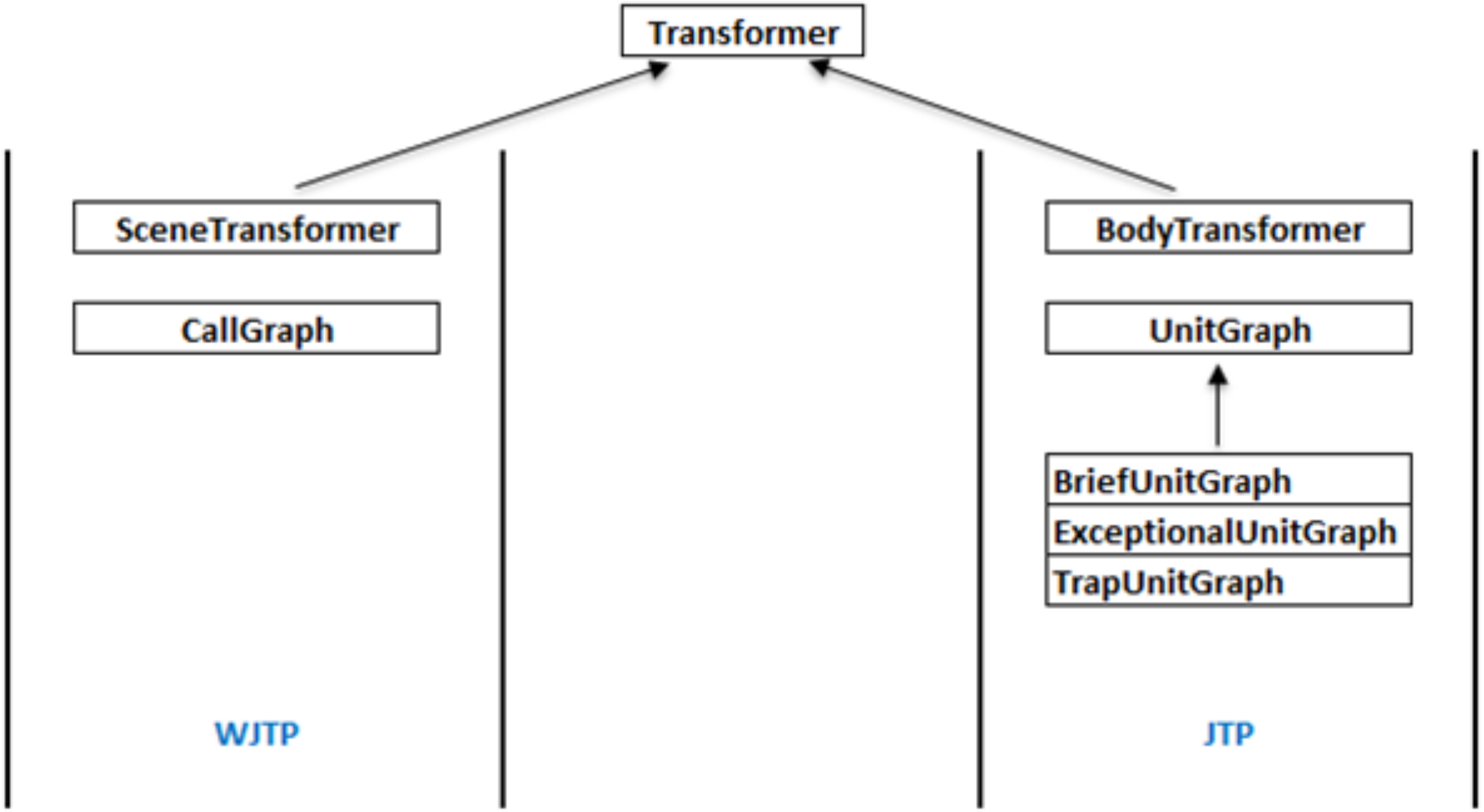
- Result Output will be shown along with the option for used to filter the results.
- User can select the any of the available detail level and the available filter by giving the same as input for filtering the result.
- With respect to current implementation user can filter the result only once using the command line. Need to extend the functionality in the next milestone.

Soot



- **Scene:** data structure for a whole program
- **SootClass:** data structure for classes
- **SootMethod:** data structure for methods
- **SootField:** data structure for fields
- Method bodies (e.g. **JimpleBody**): data structure for method body (code)

Soot - Basic diagram of relevant classes



Status: Level 2a - NodeLinkerLvl2a

- **Inter Procedural Call Graph**
  - **SceneTransformer** class
  - Create dummy main method - consists of
    - Constructors of Android component classes
    - Call back functions and lifecycle of each Android component

- **Intra Procedure Graph**
  - **BodyTransformer** class
  - Directed control flow graph