Mobile Applications
Platform
Tight Coupling
Platforms Change
Platforms Change Frequently

- Android
- Petit Four
- Cupcake
- Donut
- Eclair
- Froyo
- Gingerbread
- Honeycomb
- Ice Cream
- Jelly Bean
- KitKat
- Lollipop
- Marshmallow
- Nougat
- Oreo
- Pie
- Android 10
Adaptive Maintenance
Intuition

Developer A

App A

Developer B

App B

Developer C

App C
Intuition

Developer A

App A

Developer B

App B

Developer C

App C
Intuition

Developer A

App A

Developer B

App B

Developer C

App C
API-Usage Migrations

API-Usage Changes

⚠️ This method was deprecated in API level 23.
This method does not support multiple connected networks of the same type.
Use `getAllNetworks()` and `getNetworkInfo(android.net.Network)` instead.

Old API Usage

```java
public NetworkInfo[] getAllNetworkInfo()
```

New API Usage

```java
public Network[] getAllNetworks()
public NetworkInfo getNetworkInfo(Network network)
```
API Updates

API-Usage Changes

⚠️ This method was deprecated in API level 23.
This method does not support multiple connected networks of the same type.
Use `getAllNetworks()` and `getNetworkInfo(android.net.Network)` instead.

Old API Usage

```java
class NetworkInfo[] getAllNetworkInfo()
```

New API Usage

```java
class Network[] getAllNetworks()
class NetworkInfo getNetworkInfo(Network network)
```
API Updates

API-Usage Changes

```java
public NetworkInfo[] getAllNetworkInfo()

⚠️ This method was deprecated in API level 23.

This method does not support multiple connected networks of the same type.
Use `getAllNetworks()` and `getNetworkInfo(android.net.Network)` instead.
```

Old API Usage

```java
public NetworkInfo[] getAllNetworkInfo()
```

New API Usage

```java
public Network[] getAllNetworks()
public NetworkInfo getNetworkInfo(Network network)
```
Migration Example

Migration Example Before

```java
public boolean isConnected(Context cont) {
    ConnectivityManager cm = ...;
    NetworkInfo[] info = cm.getAllNetworkInfo();
    for (int i = 0; i < info.length; i++) {
        if (info[i].isConnected()) {
            return true;
        }
    } }
    Toast.makeText(R.s.noNet).show();
    return false;
}
```

Migration Example After

```java
public boolean isConnected(Context cont) {
    ConnectivityManager cm = ...;
    if (VERSION.SDK_INT >= VERSION_CODES.M) {
        Network[] networks = cm.getAllNetworks();
        for (Network mNetwork : networks) {
            NetworkInfo networkInfo = cm.getNetworkInfo(mNetwork);
            if (networkInfo.isConnected()) {
                Log.d(networkInfo.getTypeName());
                return true;
            }
        }
    }
    else {
        NetworkInfo[] info = cm.getAllNetworkInfo();
        for (NetworkInfo anInfo : info) {
            if (anInfo.isConnected()) {
                Log.d(anInfo.getTypeName());
                return true;
            }
        }
    }
    Toast.makeText(cont.getString(...)).show();
    return false;
}
```
Migration Example

Migration Example Before

```java
public boolean isConnected(Context cont) {
    ConnectivityManager cm = ...;
    NetworkInfo[] info = cm.getAllNetworkInfo();
    for (int i = 0; i < info.length; i++) {
        if(info[i].isConnected()) {
            return true;
        }
    }
    Toast.makeText(R.s.noNet).show();
    return false;
}
```

Migration Example After

```java
public boolean isConnected(Context cont) {
    ConnectivityManager cm = ...;
    if (VERSION.SDK_INT >= VERSION_CODES.M) {
        Network[] networks = cm.getAllNetworks();
        for (Network mNetwork : networks) {
            NetworkInfo networkInfo = cm.getNetworkInfo(mNetwork);
            if(networkInfo.isConnected()) {
                Log.d(networkInfo.getTypeName());
                return true;
            }
        }
    } else {
        NetworkInfo[] info = cm.getAllNetworkInfo();
        for (NetworkInfo anInfo : info) {
            if(anInfo.isConnected()) {
                Log.d(anInfo.getTypeName());
                return true;
            }
        }
    }
    Toast.makeText(cont.getString(...)).show();
    return false;
}
```
public boolean isConnected(Context cont) {
    ConnectivityManager cm = ...;
    NetworkInfo[] info = cm.getAllNetworkInfo();
    for (NetworkInfo anInfo : info) {
        if (anInfo.isConnected()) {
            return true;
        }
    }
    Toast.makeText(R.string.noNet).show();
    return false;
}

Migration Example

Migration Example Before

public boolean isConnected(Context cont) {
    ConnectivityManager cm = ...;
    NetworkInfo[] info = cm.getAllNetworkInfo();
    for (NetworkInfo anInfo : info) {
        if (anInfo.isConnected()) {
            return true;
        }
    }
    Toast.makeText(R.string.noNet).show();
    return false;
}

Migration Example After

public boolean isConnected(Context cont) {
    ConnectivityManager cm = ...;
    if (VERSION.SDK_INT >= VERSION_CODES.M) {
        Network[] networks = cm.getAllNetworks();
        for (Network mNetwork : networks) {
            NetworkInfo networkInfo = cm.getNetworkInfo(mNetwork);
            if (networkInfo.isConnected()) {
                Log.d(networkInfo.getTypeName());
                return true;
            }
        }
    } else {
        NetworkInfo[] info = cm.getAllNetworkInfo();
        for (NetworkInfo anInfo : info) {
            if (anInfo.isConnected()) {
                Log.d(anInfo.getTypeName());
                return true;
            }
        }
    }
    Toast.makeText(cont.getString(...)).show();
    return false;
}
API-Usage Change Specification

Target App

API-Usage Analysis

Migration Examples Search

Update Examples Analysis

Generic Migration Patches

API-Usage Migration

Evolved Target App

API-Usage Migration Report

1. Identify API usages requiring migration in target app
2. Find migration examples for identified API usages
3. Abstract migration examples into generic migration patches and rank them
4. Migrate and validate API usages in target app based on patches
APIMigrator Overview

1. Identify API usages requiring migration in target app
2. Find migration examples for identified API usages
3. Abstract migration examples into generic migration patches and rank them
4. Migrate and validate API usages in target app based on patches
APIMIGRATOR Overview

1. Identify API usages requiring migration in target app
2. Find migration examples for identified API usages
3. Abstract migration examples into generic migration patches and rank them
4. Migrate and validate API usages in target app based on patches

Flowchart:
- API-Usage Change Specification
- Target App
- API-Usage Analysis
- API-Usage Change Specification
- API-Usage Report
- Migration Examples Search
- Migration Examples
- Update Examples Analysis
- Generic Migration Patches
- API-Usage Migration
- Evolved Target App
- API-Usage Migration Report
API-Usage Analysis

Find migration examples for identified API usages

Abstract migration examples into generic migration patches and rank them

Migrate and validate API usages in target app based on patches
APIMigrator Overview

1. Identify API usages requiring migration in target app
2. Find migration examples for identified API usages
3. Abstract migration examples into generic migration patches and rank them
4. Migrate and validate API usages in target app based on patches
root@apimigrator-VirtualBox:/home/apimigrator/Desktop/apimigrator/run/API-Usage_Analysis# ./run.sh A02
root@apimigrator-VirtualBox:/home/apimigrator/Desktop/apimigrator/run/API-Usage_Analysis#/run.sh A02
Empirical Evaluation

Research Questions

**RQ1 (EFFECTIVENESS):** Can APIMIGRATOR migrate API usages in real-world apps?

**RQ2 (EFFICIENCY):** What is the cost of running APIMIGRATOR?
Evaluation

RQ1 (EFFECTIVENESS): Can APIMIGRATOR migrate API usages in real-world apps?

- 17/20 (85%) successful update rate (for API usages)
- 37/41 (90%) successful update rate (for API-usage occurrences)
- 25/37 (68%) automatic validation rate (for API-usage occurrences)

APPEVOLVE is effective in automatically updating API usages.

RQ2 (EFFICIENCY): What is the cost of running APIMIGRATOR?

<table>
<thead>
<tr>
<th>Average Execution Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>API-Usage Analysis</td>
</tr>
<tr>
<td>28s</td>
</tr>
</tbody>
</table>

The cost of the update examples search phase dominates the cost of the other phases.
Summary

Platform
Tight Coupling

APIMIGRATOR Overview
1. Identify API usages requiring migration in target app
2. Find migration examples for identified API usages
3. Abstract migration examples into generic migration patches and rank them
4. Migrate and validate API usages in target app based on patches

Empirical Evaluation
Research Questions

RQ1 (Effectiveness): Can APP EVOLVE update API usages in real-world apps?

RQ2 (Efficiency): What is the cost of running APP EVOLVE?
Summary

API MIGRATOR VM
https://zenodo.org/record/3668385

**API MIGRATOR VM**

**Platform**
Tight Coupling

---

**APIMigrator Overview**

Abstract migration examples into generic migration patches and rank them

Migrate and validate API usages in target app based on patches

---

**Search Questions**

**RQ1 (Effectiveness):** Can APPEVOLVE update API usages in real-world apps?

**RQ2 (Efficiency):** What is the cost of running APPEVOLVE?