

Finding Resume and Restart Errors in Android Applications

[OOPSLA'16]

Iulian Neamtiu

New Jersey Institute of Technology

joint work with Zhiyong Shan (Univ. of Central Missouri), Tanzirul Azim (MSR)

Smartphone apps have a fundamentally different lifecycle, compared to desktop/server programs



Desktop/server programs never killed by the system during proper operation



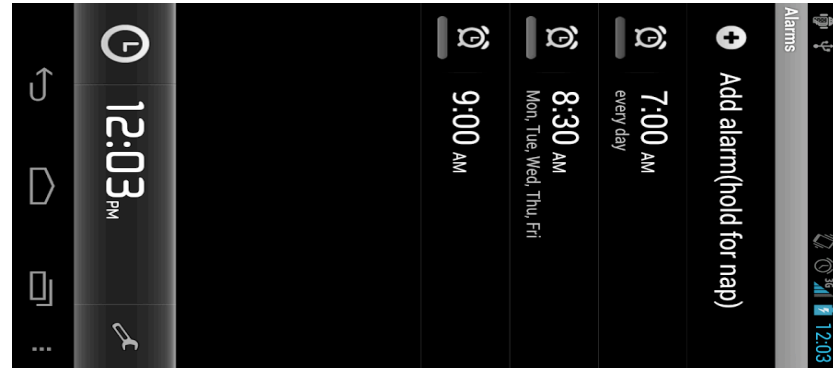
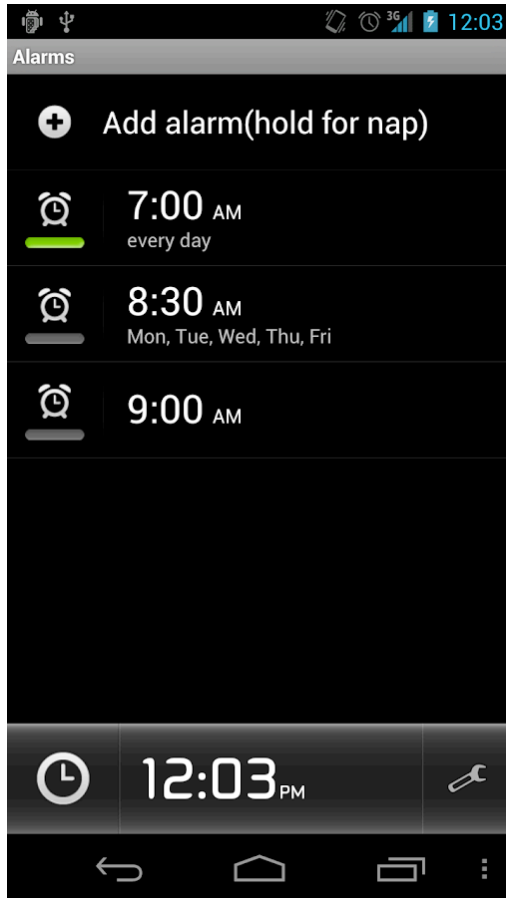
Android apps are stopped when

- Pressing *Home* button
- Pressing *Back* button
- Changing phone's orientation
- Receiving phone call
- Receiving text message
- etc.

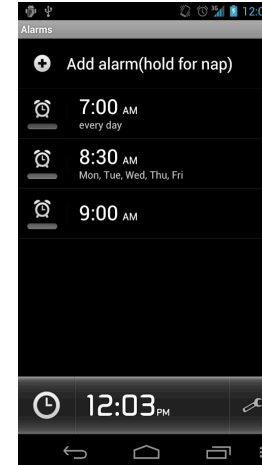
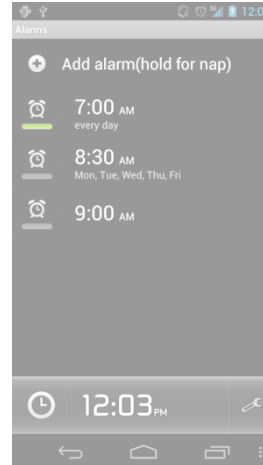
Rationale: energy, security, etc.
iOS has similar behavior

Earlier version of Alarm Clock Plus app,
more than 5 million installs

Change phone orientation, **alarm is reset !**



Or send app to background, then bring to foreground
alarm is reset!



Instance data that has to persist across runs -> **KR data**
KR data lost upon restart/resume -> **KR error**

Click one of the buttons to select a date/time.

Select DefaultDate

Select AlternativeDate

Select CustomDate

Select Month/Year

Select Time

Select DateTime

The chosen date and time:
11. March 2016
01:15

Before restart

Click one of the buttons to select a date/time.

Select DefaultDate

Select AlternativeDate

Select CustomDate

Select Month/Year

Select Time

Select DateTime

?

After restart

Example: KR error in DateSlider app

```

1  public abstract class DateSlider extends Dialog {
2      protected Calendar mTime;
3      .....
4      public void onCreate(Bundle savedInstanceState) {
5          super.onCreate(savedInstanceState);
6          if ( savedInstanceState!=null) {
7              long time = savedInstanceState.getLong(
8                  "time", mTime.getTimeInMillis());
9              mTime.setTimeInMillis(time);
10         }
11         .....
12     }
13     .....
14     public void updateCalendar(Calendar calendar) {
15         mTime.setTimeInMillis( calendar . getTimeInMillis ());
16     }
17     .....
18     public Bundle onSaveInstanceState() {
19         Bundle savedInstanceState =
20             super.onSaveInstanceState();
21         if ( savedInstanceState==null)
22             savedInstanceState = new Bundle();
23         savedInstanceState.putLong("time",
24             mTime.getTimeInMillis());
25         return savedInstanceState;
26     }
27     protected void onDestroy() {
28         ...
29     }} ...

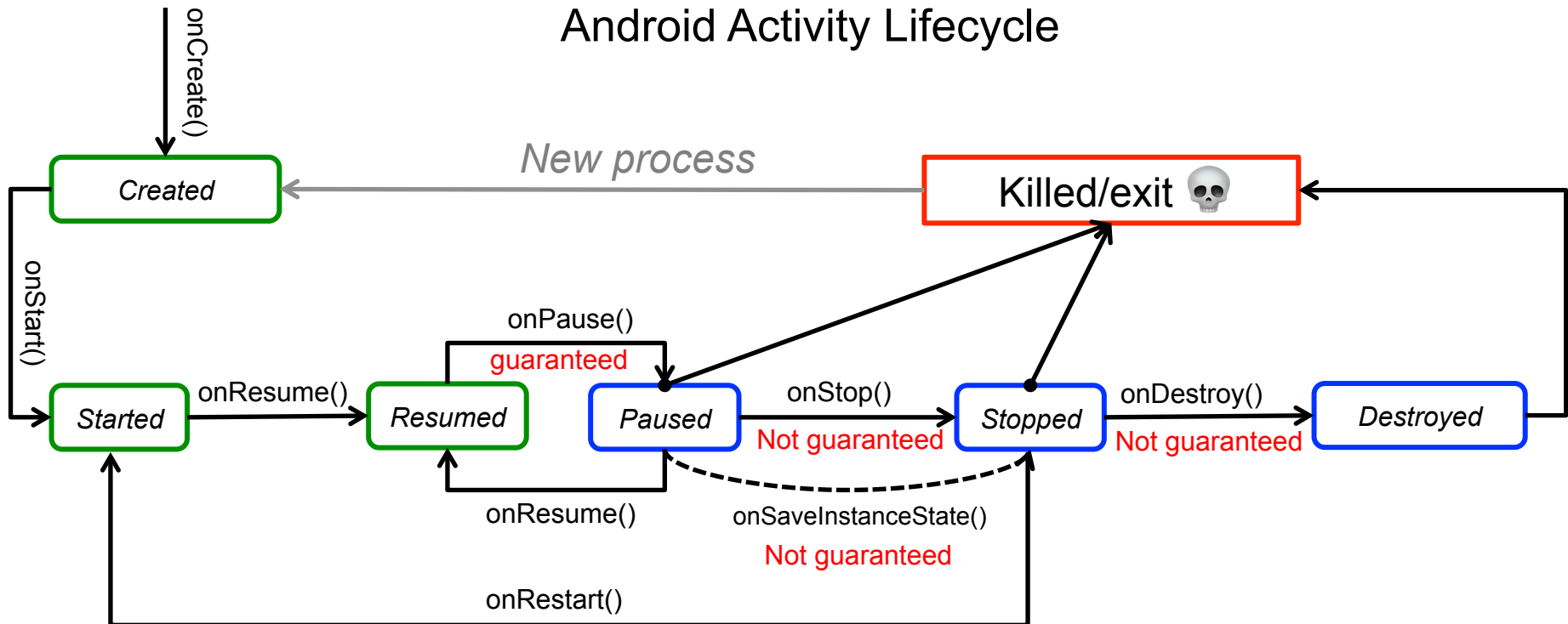
```

mTime is modified

When pressing the *Back* button, `onSaveInstanceState()` is not called by the system → the save operation `putLong()` is not called

Platform support and save protocol

Android saves GUI state, invokes user-defined callbacks
Developers *must do the rest*, e.g., fill out callbacks
Confusing and error-prone

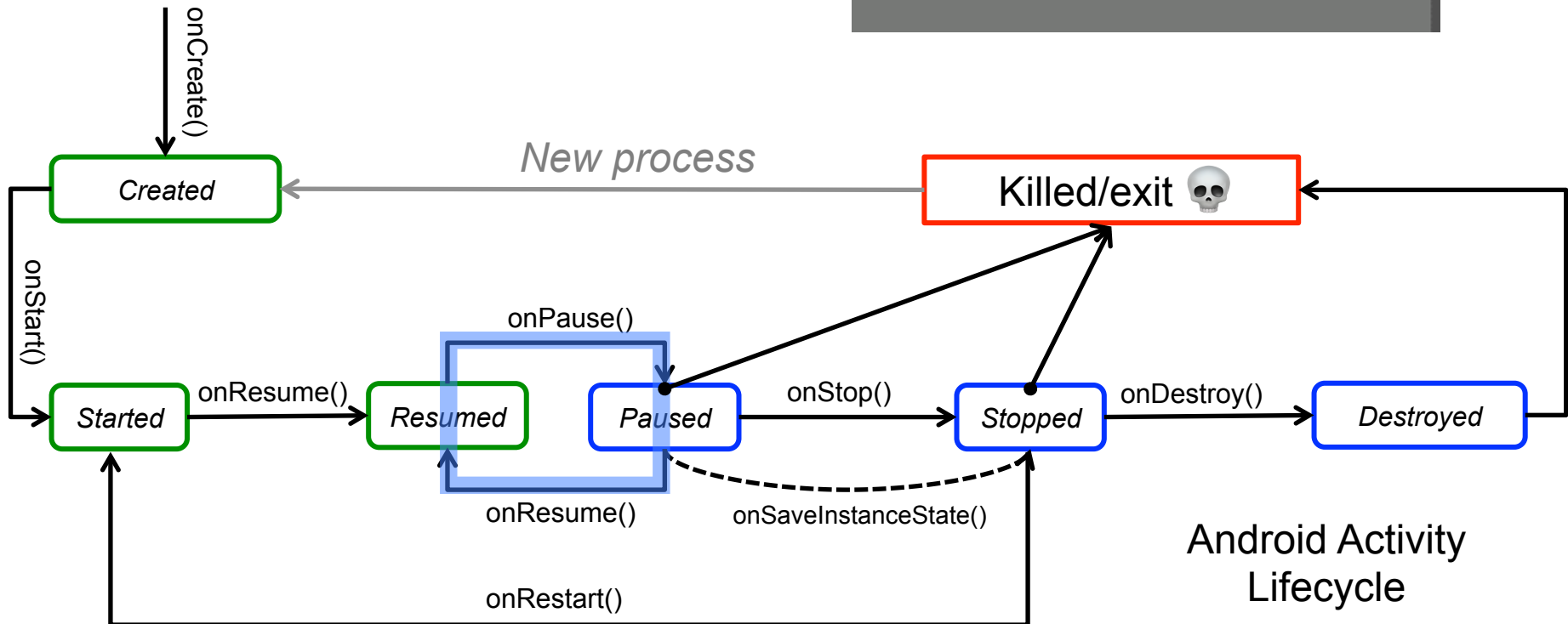
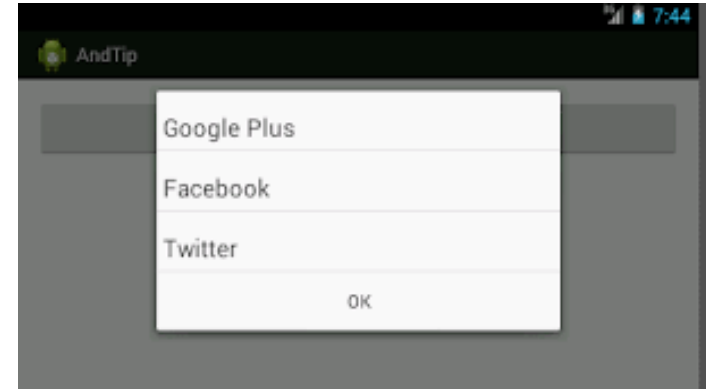


Define Kill & Restart (KR) hierarchy

Level 1: Pause activity

Causes: Display dialog
Show drop-down menu

Callbacks: onPause()/onResume()

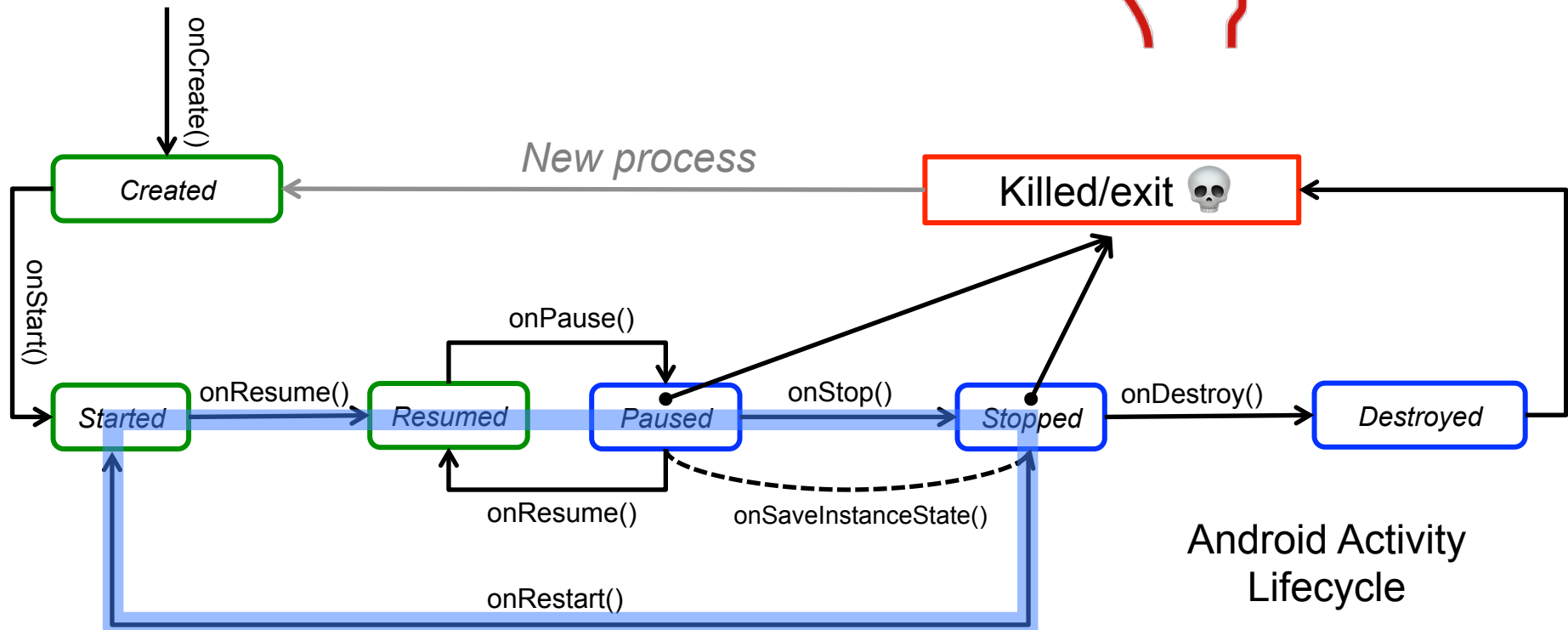
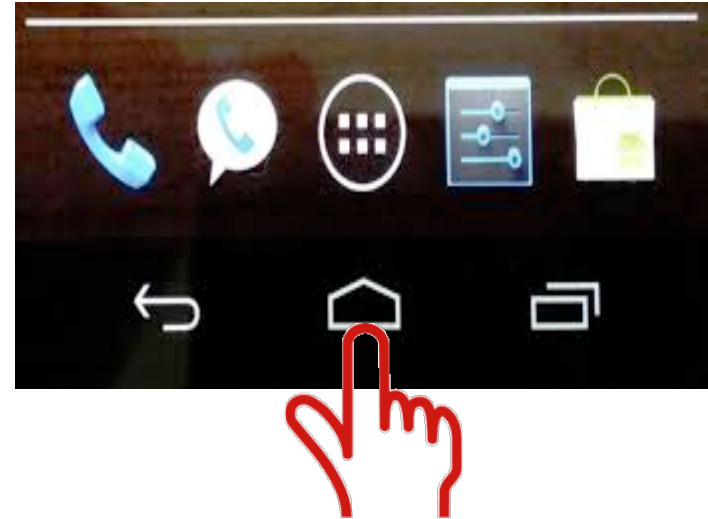


Android Activity
Lifecycle

Level 2: Stop activity

Causes: Press **Home** button
Switch to another app
Start a new activity
Receive a phone call

Callbacks: onStop()/onRestart()

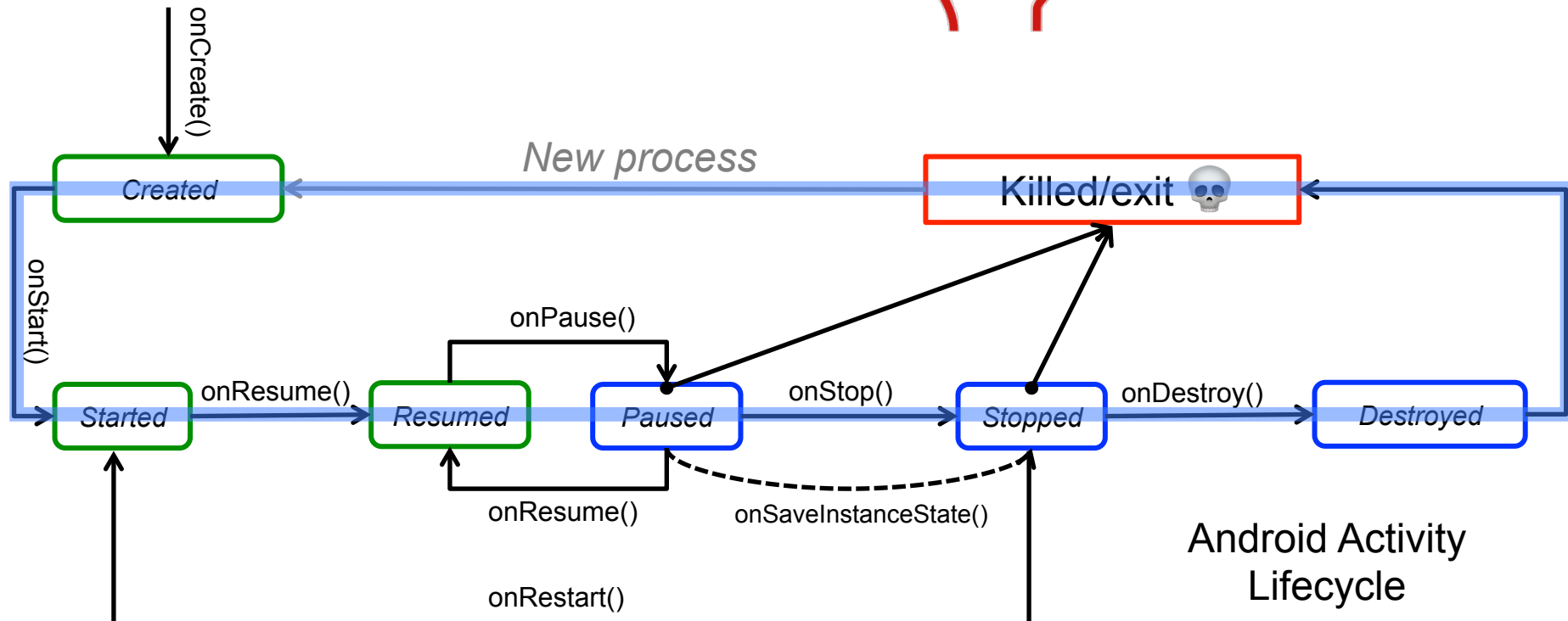


Android Activity
Lifecycle

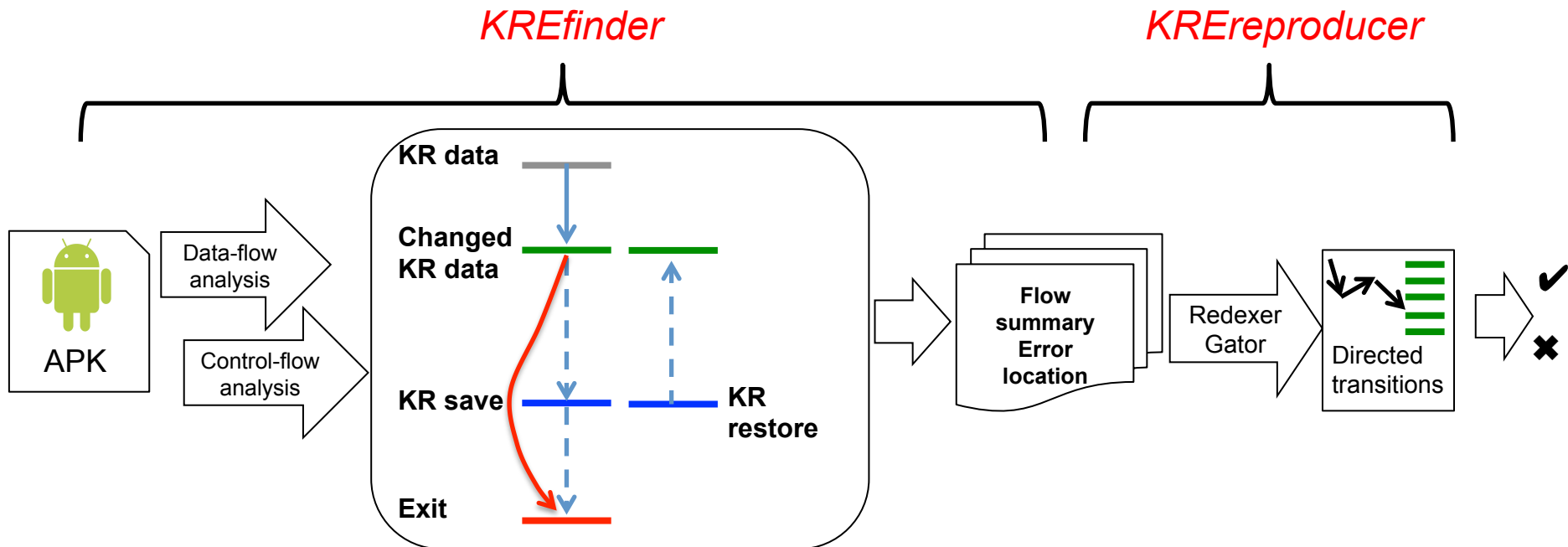
Level 3: Destroy activity

Causes: Press **Back** button
Killed by user

Callbacks: onDestroy()/onCreate()



Our approach: a complete toolchain to find and reproduce KR errors



- Find KR data, KR changes, KR saves, KR restores and exits
- Track data and control flow from changed KR data to KR saves, exits
- Track data and control flow between KR changes and KR restores

- Report potential bugs and their location
- Generate a list of targeted methods

- Construct input sequence to reach error location
- User verifies error

KREfinder: data flow analysis points

```
1 public abstract class DateSlider extends Dialog {
2     protected Calendar mTime;
3     .....
4     public void onCreate(Bundle savedInstanceState) {
5         super.onCreate(savedInstanceState);
6         if ( savedInstanceState!=null ) {
7             long time = savedInstanceState.getLong(
8                 "time", mTime.getTimeInMillis());
9             mTime.setTimeInMillis(time);
10        }
11        .....
12    }
13    .....
14    public void updateCalendar(Calendar calendar) {
15        mTime.setTimeInMillis(calendar.getTimeInMillis());
16    }
17    .....
18    public Bundle onSaveInstanceState() {
19        Bundle savedInstanceState =
20            super.onSaveInstanceState();
21        if ( savedInstanceState==null )
22            savedInstanceState = new Bundle();
23        savedInstanceState.putLong("time",
24            mTime.getTimeInMillis());
25        return savedInstanceState;
26    }
27    protected void onDestroy() {
28        ...
29    } ...
```

KR data

Fields that are mutable and modified

KR restore

API calls that restore data, e.g.,
Android.OS.Bundle.get*()

KR data change

Assignment statement where a KR
data (or alias thereof) is written to

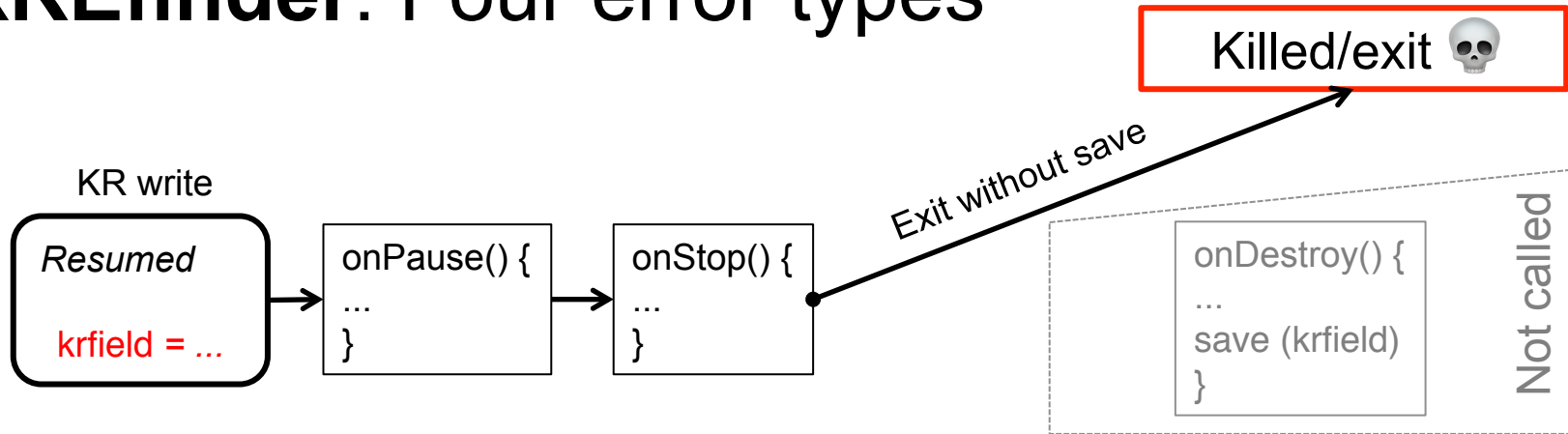
KR save

API calls that save data, e.g.,
Android.OS.Bundle.put*()

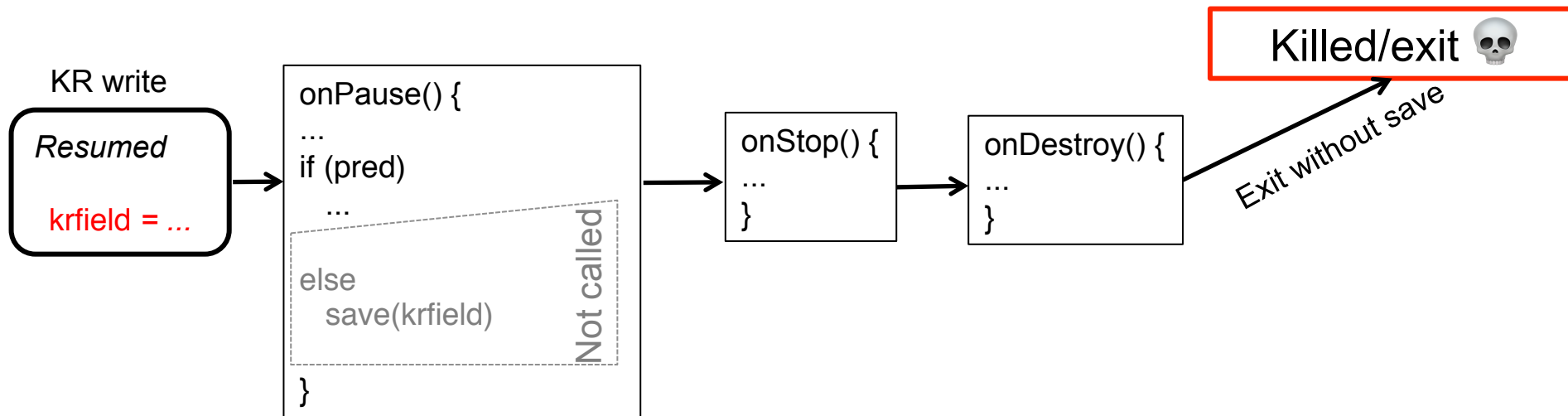
The end of onDestroy() or onStop(),
java.lang.system.exit() call

Exit

KREfinder: Four error types



Type 1: save operation in a callback not guaranteed to execute



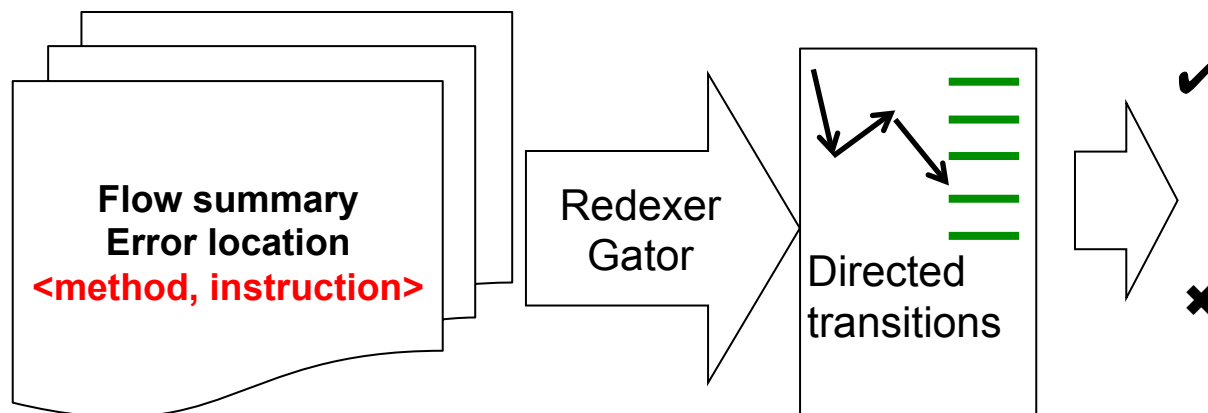
Type 2: save operation in a callback guaranteed to execute, but not on the path to exit

Type 3: no save or restore of KR data

Type 4: restore KR data but no save operation

KREreproducer

- Given an error point **<method, instruction>** generate a directed transition to reach that point
- Undecidable in general, but works well on Android
 - Use **Redexer** to generate a method transition list that ends at **method**
 - Identify GUI callbacks
 - Use **Gator** to build the UI-event mapping that leads to **<method, instruction>**



Evaluation

Dataset

- 324 apps from Google Play, AppsApk, GitHub
- Categories
Utilities, Email & SMS, Games, Health & Fitness
- Size
3KB—23MB, on average 377KB
- Popularity (installs)
11 apps with more than 1 million installs, e.g., Facebook, Yahoo Weather, NPR News

Effectiveness

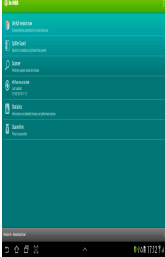
49 confirmed bugs in 37 apps (1.4 bugs per app)

210 potential bugs in 287 apps

Efficiency

Time (seconds)			
Min	Max	Average	Median
1	2,184	61	22

Error examples and categories

App	Error Description	App	Error Description
	Norton snap QR User turns on the flashlight, then stops the app; after resuming, flashlight is off		Dr.Web Anti-virus Light The custom scan check box setting is lost after restart
	OpenSudoku Game state lost		Motorola Camera The user switches from image mode to video mode, then restarts. The camera returns to image mode again

- Losing GUI state
 - Losing user's progress
 - Losing device settings
- (see paper for more examples)

Conclusions

- Smartphone apps are restarted frequently
 - When restarts not handled properly → KR errors
- Our approach finds and reproduces KR errors
 - Effective: 324 apps, found 49 reproducible bugs, 210 other potential bugs
 - Efficient: 61 seconds per app
- Wanna know more?
 - See our OOPSLA'16 paper
 - Try the open source toolchain, available at <http://spruce.cs.ucr.edu/kre/>