Interstitial Technical & Best Implementation practices

Content

I. Common mistakes & best practices to fix

1. Interstitial on opening app
2. Interstitial between pages
3. Interstitial on exiting app
4. Interstitial after suspending or exiting app
5. Best practice to control interstitial (preload & show)

II. Optimize Interstitial performance with Firebase

1. Control Interstitials frequency
2. Find the best placements for your Interstitials

III. Signals of Interstitial problem on Admob Report

1. Problems about match rate
2. Problems about show rate

I. Common mistakes

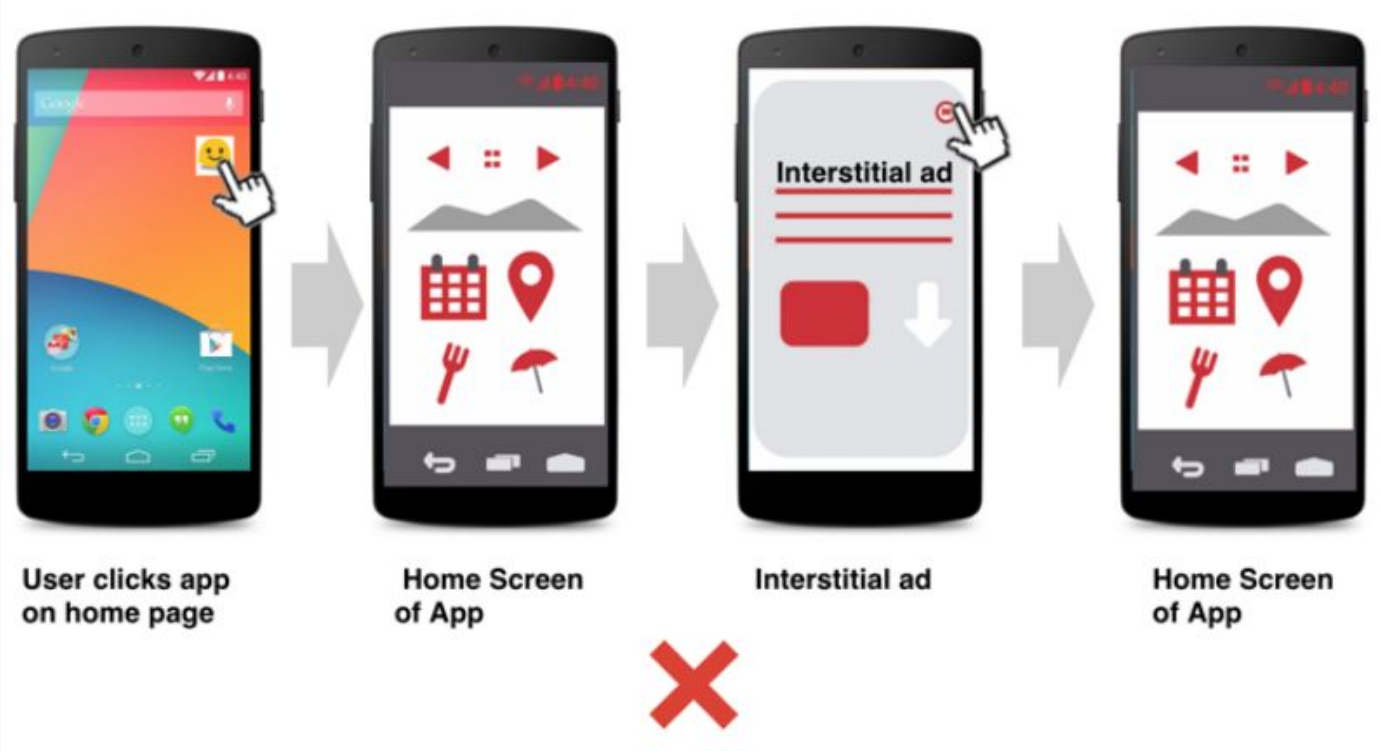
1. Interstitial on opening app
2. Interstitial between pages
3. Interstitial on exiting app
4. Interstitial after suspending or exiting app
5. Best practice to control interstitial

1. Interstitial on opening app

Mistake: Interstitial shows right after user click to open app

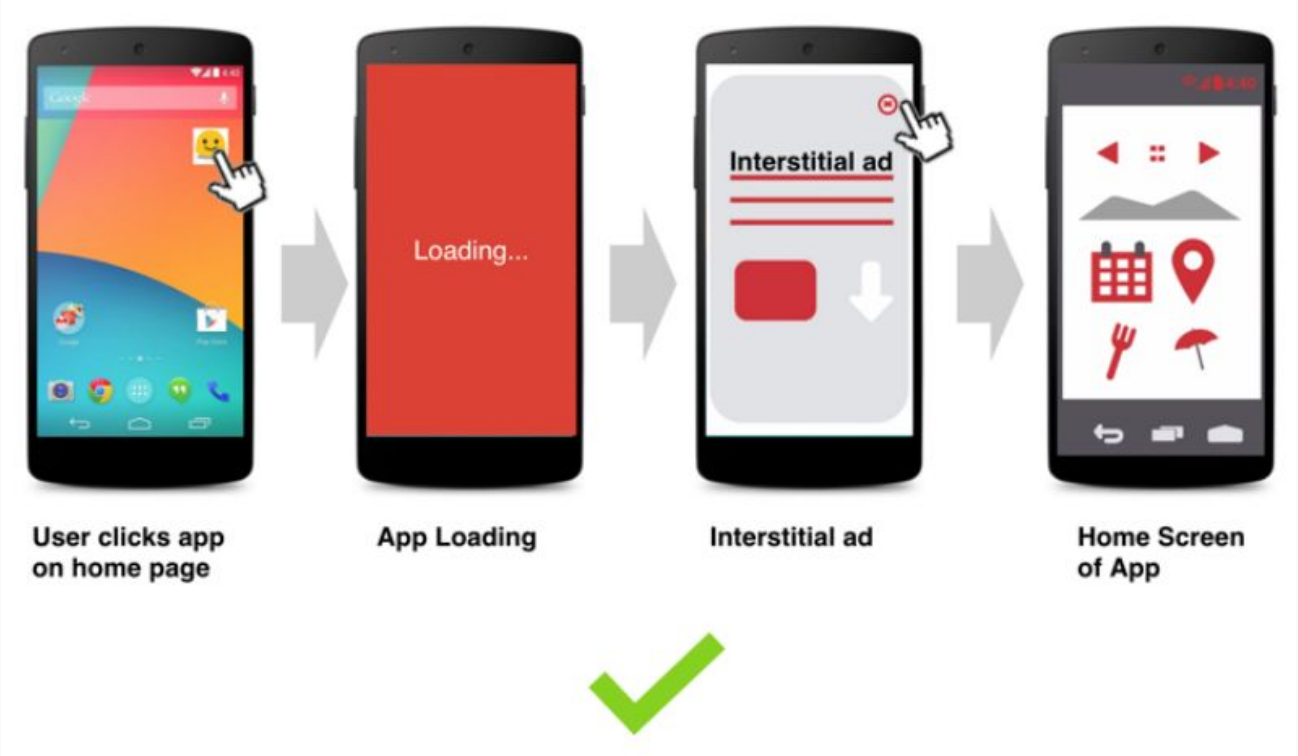
1. Interstitial on opening app

Mistake: Interstitial show AFTER users land the home page



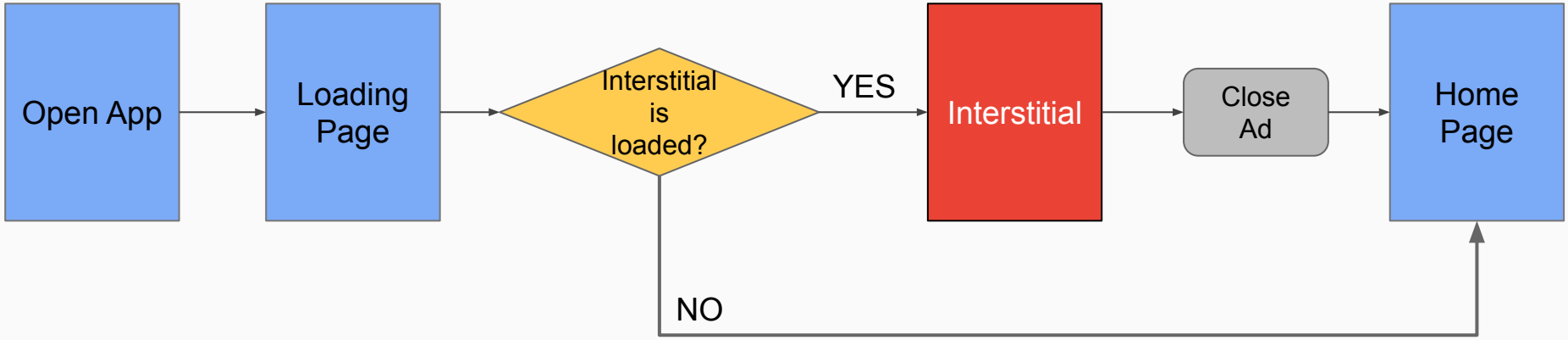
1. Interstitial on opening app

Interstitial should show between Loading Page and Home Page



1. Interstitial on opening app

Interstitial show between Loading Page and Home Page logic



Important: wait for ad closed to open Home Page

1. Interstitial on opening app

Implementation

```
private static final String APP_ID = "ca-app-pub-3940256099942544~3347511713";
private static final String INTERSTITIAL_AD_UNIT = "ca-app-pub-3940256099942544/1033173712";
private InterstitialAd interstitialAd;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_test_code);

    MobileAds.initialize( context: this, APP_ID);
    setupInterstitialAd();
    loadInterstitialAd();
    loadingResources(); //loading some resources for your app
}
```

1. Interstitial on opening app

Implementation: methods to setup and load interstitial

```
private void setupInterstitialAd() {
    interstitialAd = new InterstitialAd( context: this);
    interstitialAd.setAdUnitId(INTERSTITIAL_AD_UNIT);
    interstitialAd.setAdListener(new AdListener() {
        @Override
        public void onAdClosed() {
            super.onAdClosed();
            showHomeActivity(); //show your HomePage when ad is closed
        }
    });
}

private void loadInterstitialAd() {
    if (!interstitialAd.isLoading() && !interstitialAd.isLoaded()) {
        AdRequest adRequest = new AdRequest.Builder().build();
        interstitialAd.loadAd(adRequest);
    }
}
```

1. Interstitial on opening app

Implementation: Methods to show interstitial and show Home Page

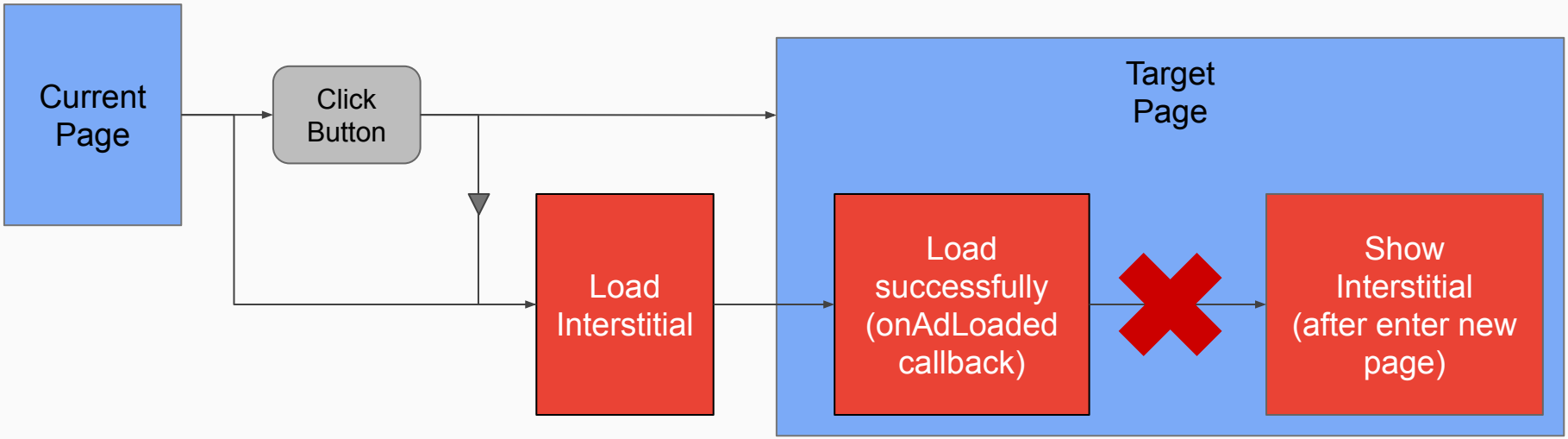
```
private void onResourcesLoaded() {  //call after you're done loading your resources
    if (interstitialAd != null && interstitialAd.isLoaded()) {
        interstitialAd.show();
    } else {
        showHomeActivity(); //show your home page if no loaded interstitial ad
    }
}

private void showHomeActivity() {   //open your HomePage
    Intent intent = new Intent( packageContext: this, HomeActivity.class);
    startActivity(intent);
}
```

2. Interstitial between pages

Mistake: Interstitial show after page changes

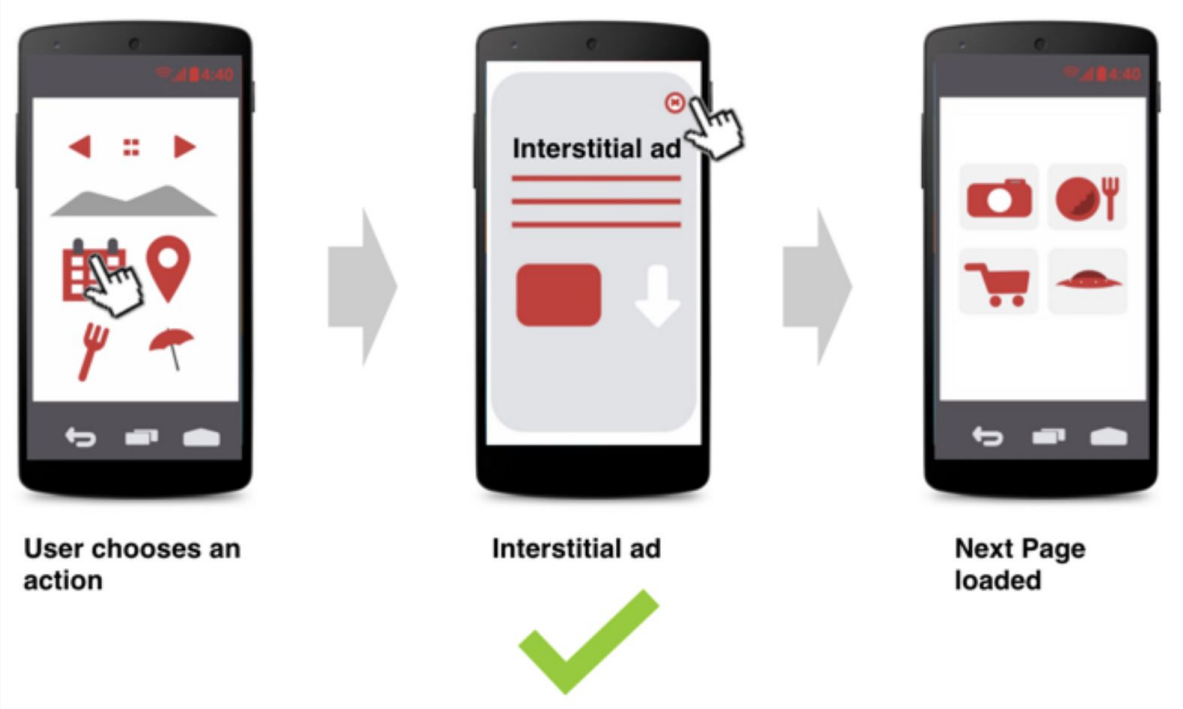
2. Interstitial between pages

Reason: show Interstitial on onAdLoaded callback



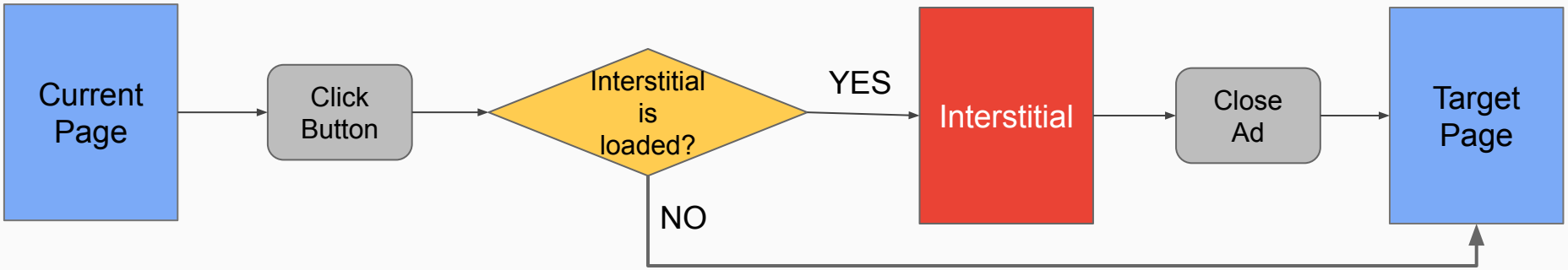
2. Interstitial between pages

Interstitial should show before a new page shows



2. Interstitial between pages

Interstitial show between pages logic



Important: wait for ad closed to show Target Page

2. Interstitial between pages

Implementation

```
private static final String ADMOB_AD_UNIT_ID = "ca-app-pub-3940256099942544/1033173712";
private static final String ADMOB_APP_ID = "ca-app-pub-3940256099942544~3347511713";

private InterstitialAd interstitialAd;
private Button showInterstitialAdButton;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main_activity);

    MobileAds.initialize( context: this, ADMOB_APP_ID);
    setupInterstitialAd();
    setupButtons();
    loadInterstitialAd();
}
```

2. Interstitial between pages

Implementation: Method to setup Interstitial Ad

```
private void setupInterstitialAd() {
    interstitialAd = new InterstitialAd( context: this);
    interstitialAd.setAdUnitId(ADMOB_AD_UNIT_ID);
    interstitialAd.setAdListener(new AdListener() {
        @Override
        public void onAdClosed() {
            super.onAdClosed();
            showTargetActivity();
        }

        @Override
        public void onAdLoaded() {
            super.onAdLoaded();
            showInterstitialAdButton.setVisibility(View.VISIBLE);
        }
    });
}
```

Show the next page after the ad is closed

Show the button after the Ad is Loaded

2. Interstitial between pages

Implementation: Methods to setup button and show ad

```
private void setupButtons() {
    showInterstitialAdButton = findViewById(R.id.retry_button);
    showInterstitialAdButton.setVisibility(View.INVISIBLE);
    showInterstitialAdButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            showInterstitial();
        }
    });

}

private void showInterstitial() {
    if (interstitialAd != null && interstitialAd.isLoaded()) {
        interstitialAd.show();
    } else {
        Toast.makeText( context: this,  text: "Ad did not load", Toast.LENGTH_SHORT).show();
        showTargetActivity();
    }
}
```

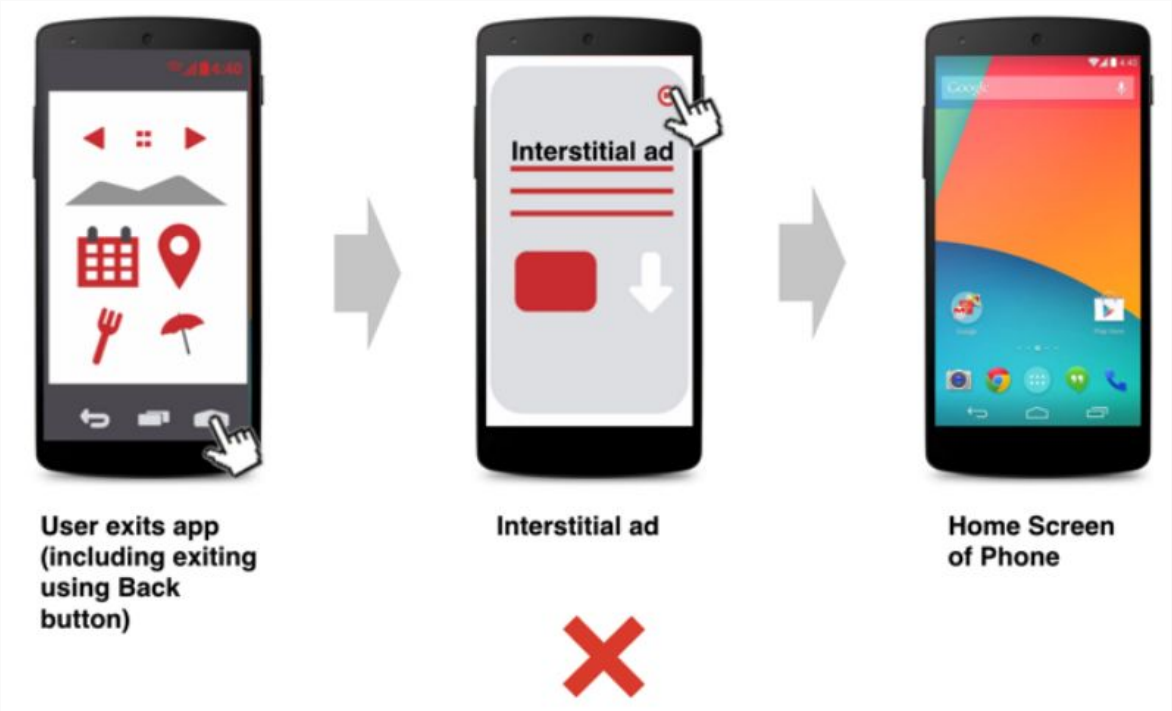
2. Interstitial between pages

Implementation: Methods to load Interstitial and show Target Activity

```
private void loadInterstitialAd() {
    if (!interstitialAd.isLoading() && !interstitialAd.isLoaded()) {
        AdRequest adRequest = new AdRequest.Builder().build();
        interstitialAd.loadAd(adRequest);
    }

    showInterstitialAdButton.setVisibility(View.INVISIBLE);
}

private void showTargetActivity() {
    Intent intent = new Intent( packageContext: this, TargetActivity.class);
    startActivity(intent);
}
```

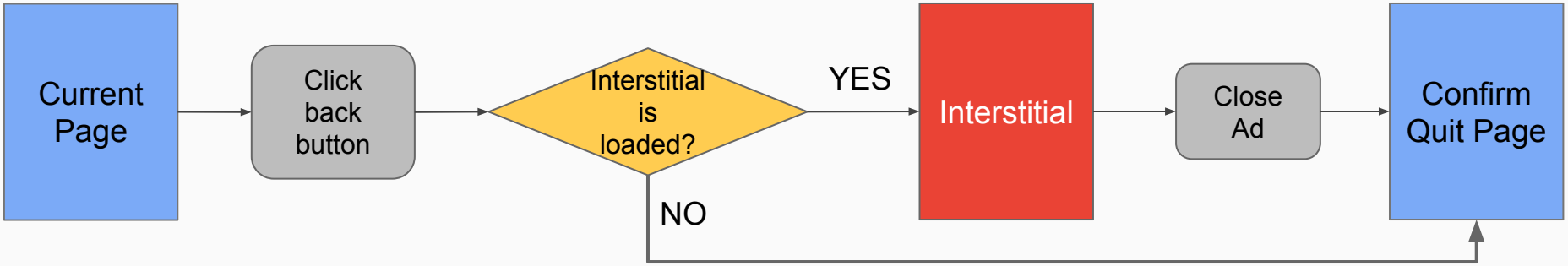
3. Interstitial on exiting app

Mistake: Interstitial show after users confirm to quit app



3. Interstitial on exiting

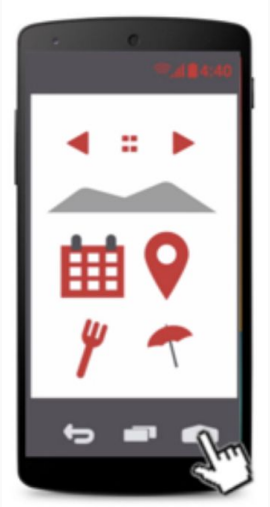
Interstitial show on app exit logic



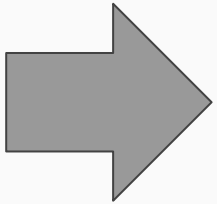
Important: wait for ad closed to show Confirm Quit Page

4. Interstitial after suspending or exiting app

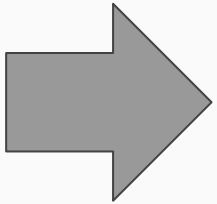
Mistake: Interstitial show after users suspend or exit app



Users suspend app by clicking home button (or exit by click back button)



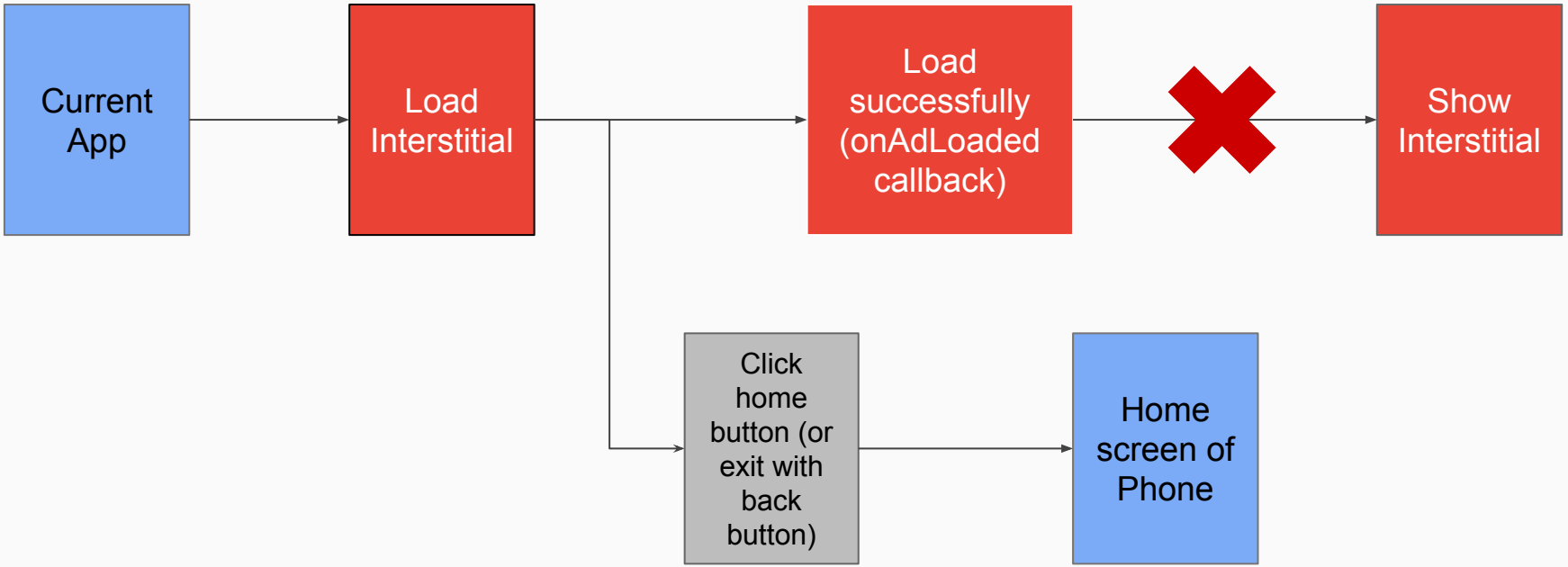
Home screen of Phone



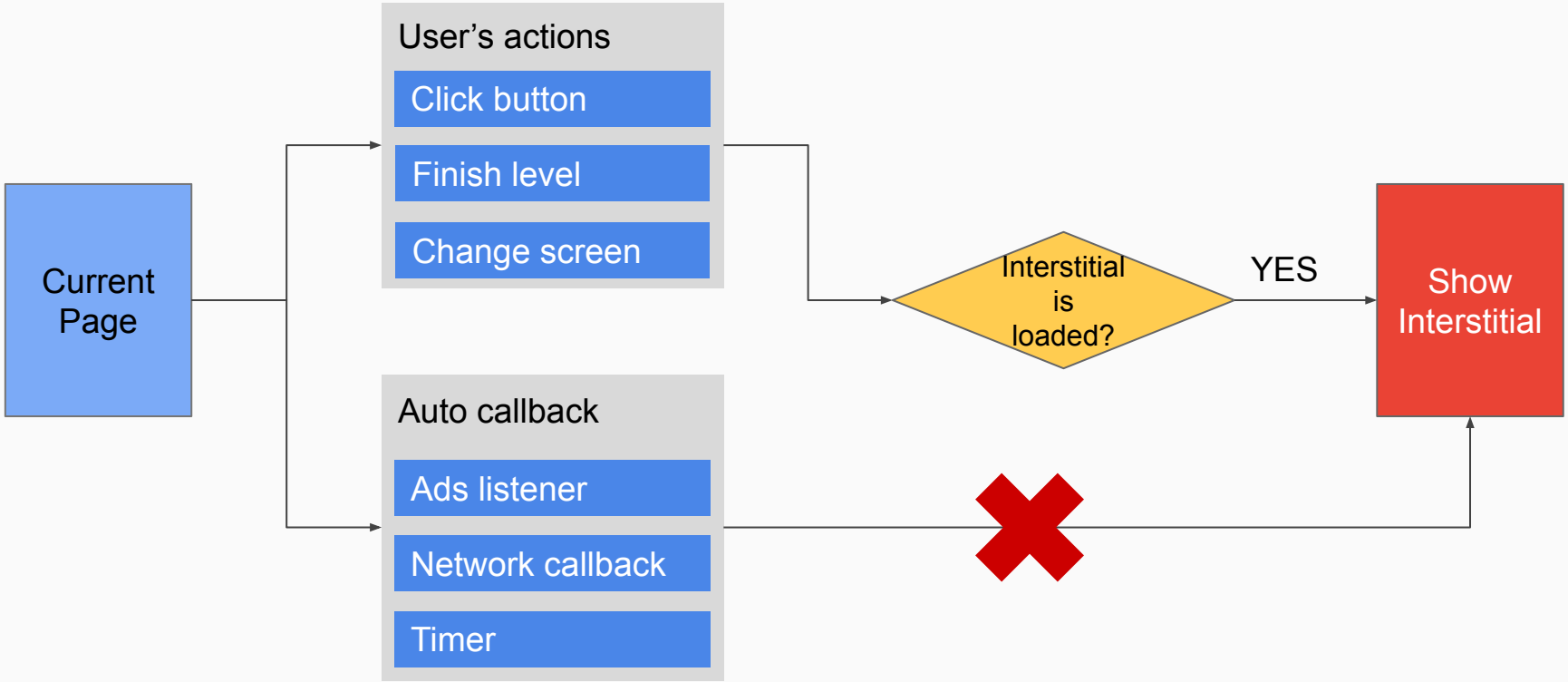
Interstitial ad

4. Interstitial after suspending or exiting app

Reason: show Interstitial on onAdLoaded callback



4. Interstitial after suspending or exiting app

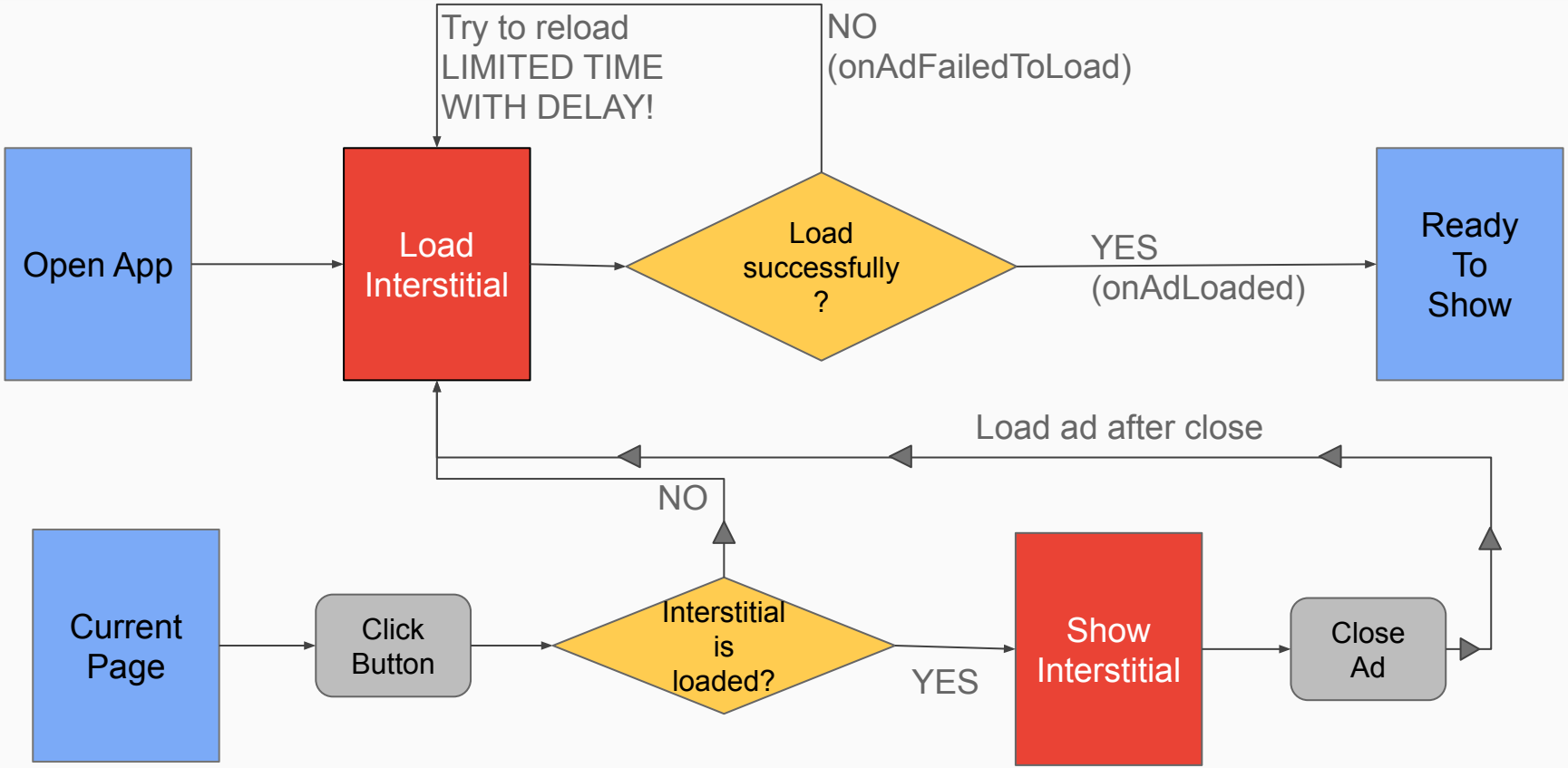


Important: only show Interstitial on user's actions

5. Best practice to preload ad

- Always preload ad by loading ad when:
  - App is opening
  - Ad is closed
  - Need to show but don’t have available ad
- Can try to reload ad but don’t do too many times (1 or 2 times is enough)

5. Best practice to preload ad



5. Best practice to preload ad

- Need a way to control the Interstitial:
  - Load one play, show every place
  - Do the auto-preload
  - Auto do the callback:
    - After ad close
    - If there isn’t no ad to show

Use Singleton method to create one class to control all the Interstitial logic

5. Best practice to preload ad

Implementation: Use Singleton to control the Interstitial

```
public class InterstitialUtils {
    private final String APP_ID = "ca-app-pub-3940256099942544~3347511713";
    private final String INTERSTITIAL_AD_UNIT = "ca-app-pub-3940256099942544/1033173712";
    private static InterstitialUtils sharedInstance;
    private InterstitialAd interstitialAd;
    private AdCloseListener adCloseListener;
    private boolean isReloaded = false;

    public static InterstitialUtils getSharedInstance() {
        if (sharedInstance == null) {
            sharedInstance = new InterstitialUtils();
        }
        return sharedInstance;
    }
```

Used for the onAdClosed callback

Used for controlling reload ad

5. Best practice to preload ad

Implementation: Function to init the Singleton Interstitial module

Do callback when ad closed (Eg: change page)

Auto load new ad when ad closed

Try to reload ONE TIME if failed to load

Load first interstitial ad on initing Interstitial module

```
public void init(Context context) {
    MobileAds.initialize(context, APP_ID);
    interstitialAd = new InterstitialAd(context);
    interstitialAd.setAdUnitId(INTERSTITIAL_AD_UNIT);
    interstitialAd.setAdListener(new AdListener() {
        @Override
        public void onAdClosed() {
            super.onAdClosed();
            if (adCloseListener != null) {
                adCloseListener.onAdClosed();
            }
            loadInterstitialAd();
        }
        @Override
        public void onAdFailedToLoad(int i) {
            super.onAdFailedToLoad(i);
            if (isReloaded == false) {
                isReloaded = true;
                loadInterstitialAd();
            }
        }
    });
    loadInterstitialAd();
}
```

5. Best practice to preload ad

Implementation: function to show Interstitial ad with a callback

```
public void showInterstitialAd(AdCloseListener adCloseListener) {
    if (canShowInterstitialAd()) {
        isReloaded = false;
        this.adCloseListener = adCloseListener;
        interstitialAd.show();
    } else {
        loadInterstitialAd();
        adCloseListener.onAdClosed();
    }
}

private boolean canShowInterstitialAd() {
    return interstitialAd != null && interstitialAd.isLoaded();
}
```

Reset the reload-flag everytime showing an ad

Set callback for the onAdClosed event

Load new ad if no ad is ready to show

Call the callback directly if we can't show ad

5. Best practice to preload ad

Implementation: function to load interstitial

```
private void loadInterstitialAd() {
    if (interstitialAd != null && !interstitialAd.isLoading()
            && !interstitialAd.isLoaded()) {
        AdRequest adRequest = new AdRequest.Builder().build();
        interstitialAd.loadAd(adRequest);
    }
}

public interface AdCloseListener {
    public void onAdClosed();
}
```

Interface for the callback of onAdClosed event

5. Best practice to preload ad

Implementation: init the Singleton module when opening app

```
public class LoadingActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        InterstitialUtils.getSharedInstance().init(getApplicationContext());
        loadResources();
    }
```

5. Best practice to preload ad

Implementation: Only need to call showInterstitial with a callback

```
private void onResourcesLoaded() {
    InterstitialUtils.getSharedInstance().showInterstitialAd(new InterstitialUtils.AdCloseListener() {
        @Override
        public void onAdClosed() {
            openGameActivity();
        }
    });
}

private void openGameActivity() {
    Intent intent = new Intent( packageContext: this, GameActivity.class);
    startActivity(intent);
}
```

When load resource done, call show interstitial with a callback to open GameActivity
If can show ad, callback will be automatically called when ad close
If not, callback will be called immediately

5. Best practice to preload ad

Implementation: Only need to call showInterstitial with a callback

```
public class GameActivity extends AppCompatActivity {

    private void onClickRetryButton() {
        InterstitialUtils.getSharedInstance().showInterstitialAd(new InterstitialUtils.AdCloseListener() {
            @Override
            public void onAdClosed() {
                startGame();
            }
        });
    }
```

When click retry button, call show interstitial with a callback to open start new game

5. Best practice to preload ad

Implementation: Singleton method

- Pros:
  - Easy to use
  - Make sure the preload and closed callback work well
- Cons:
  - Need modify to use multiple ad units

II. Optimize Interstitial performance with Firebase

1. Control Interstitial frequency
2. Choose your best Interstitials placements

1. Control Interstitial Frequency

- Showing too many ads
  - bad user experience
  - decrease active users
  - decrease ad requests
  - decrease ad revenue
- Showing too little ads
  - low impression per user
  - low ARPDAU
  - low ad revenue

It is important to control
frequency of showing ads
to balance UX and Monetization

Use a fixed interval time inside your code

- Pros: easy to implement
- Cons:
  - need to update app to change the interval value
  - hard to find the best value

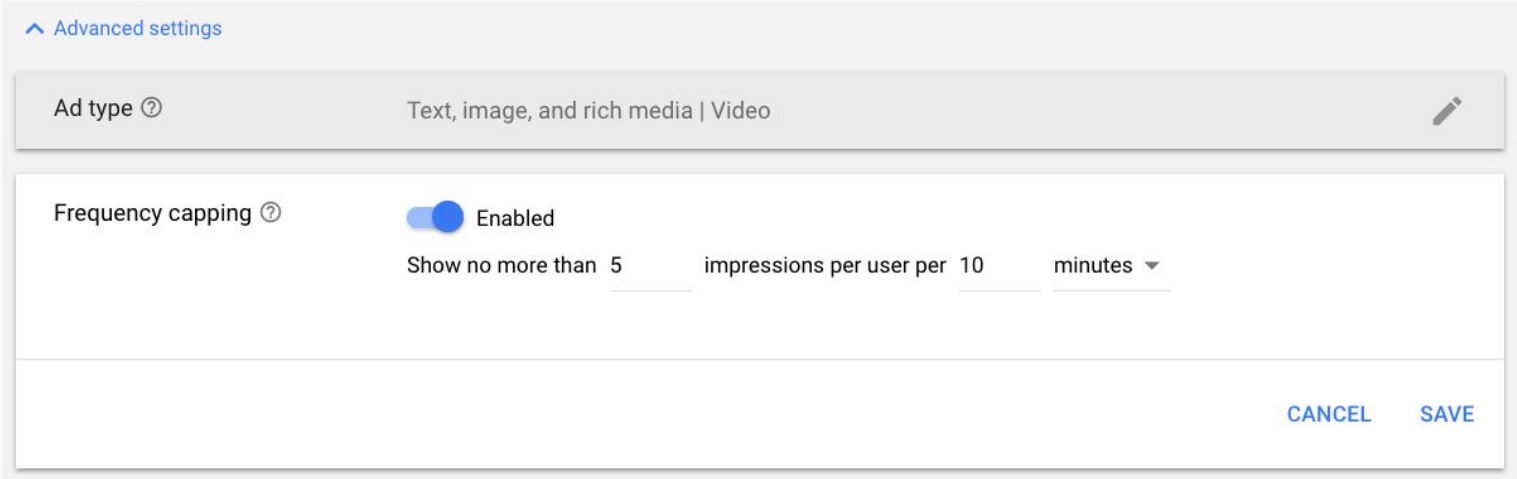
Only show interstitial after at least two minutes

Check elapsed time from the last interstitial impression

```
if(currentTime - lastTimeShowInterstitial >= TWO_MINUTES){
    showInterstitial();
}
```

Use frequency capping on Admob Console

- Pros: no need for additional app implementation
- Cons:
  - affects their preloading method
  - unable to adjust ads interval
  - difficult to find the best value



Use Firebase Remote Config

- Pros:
  - no need to update app
  - can A/B test to find the best value
- Cons: need to implement Firebase

Get the value from Firebase Remote Config and A/B testing this value

Check elapsed time from the last interstitial impression

A/B test the Frequency with Firebase

- Create a parameter on Firebase Remote Config
- Get this parameter in code and use it for control the Frequency
- Run experiment to A/B test the frequency value

A/B test the Frequency with Firebase

- Create a parameter on Firebase Remote Config

A/B test the Frequency with Firebase

- Implement parameter fetching in your app to control the frequency

```
private void showInterstitial(){
    long currentTime = getCurrentTime();
    if(currentTime - lastTimeShowInterstitial >= getLimitTime()){
        lastTimeShowInterstitial = currentTime;
        doShowInterstitial();
    }
}

private long getLimitTime() {
    return FirebaseRemoteConfig.getInstance().getLong( s: "interstitial_interval");
}
```

A/B test the Frequency with Firebase

- Step to run experiment to A/B test the interval value
  - Create the experiment and target users for the test
  - Select the goal metrics
  - Create the variants
  - Run the experiment
  - Wait for result then roll out the best value

Create experiment and target users for the test

Basics
name: Test Interstitial Frequency

2 Targeting

Target users

App | com.[redacted] and

Percentage of target users

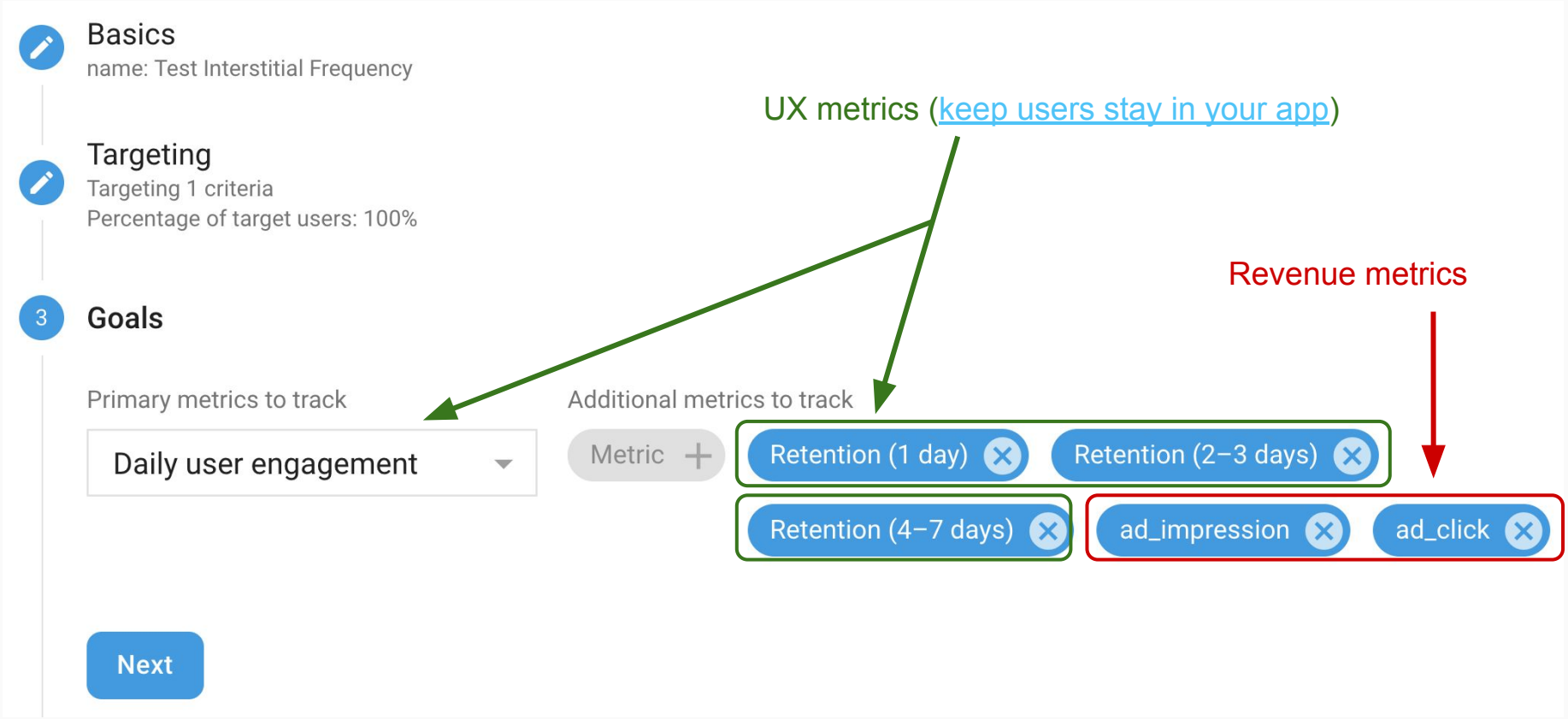
100

Activation event (optional)

Select Activation Event

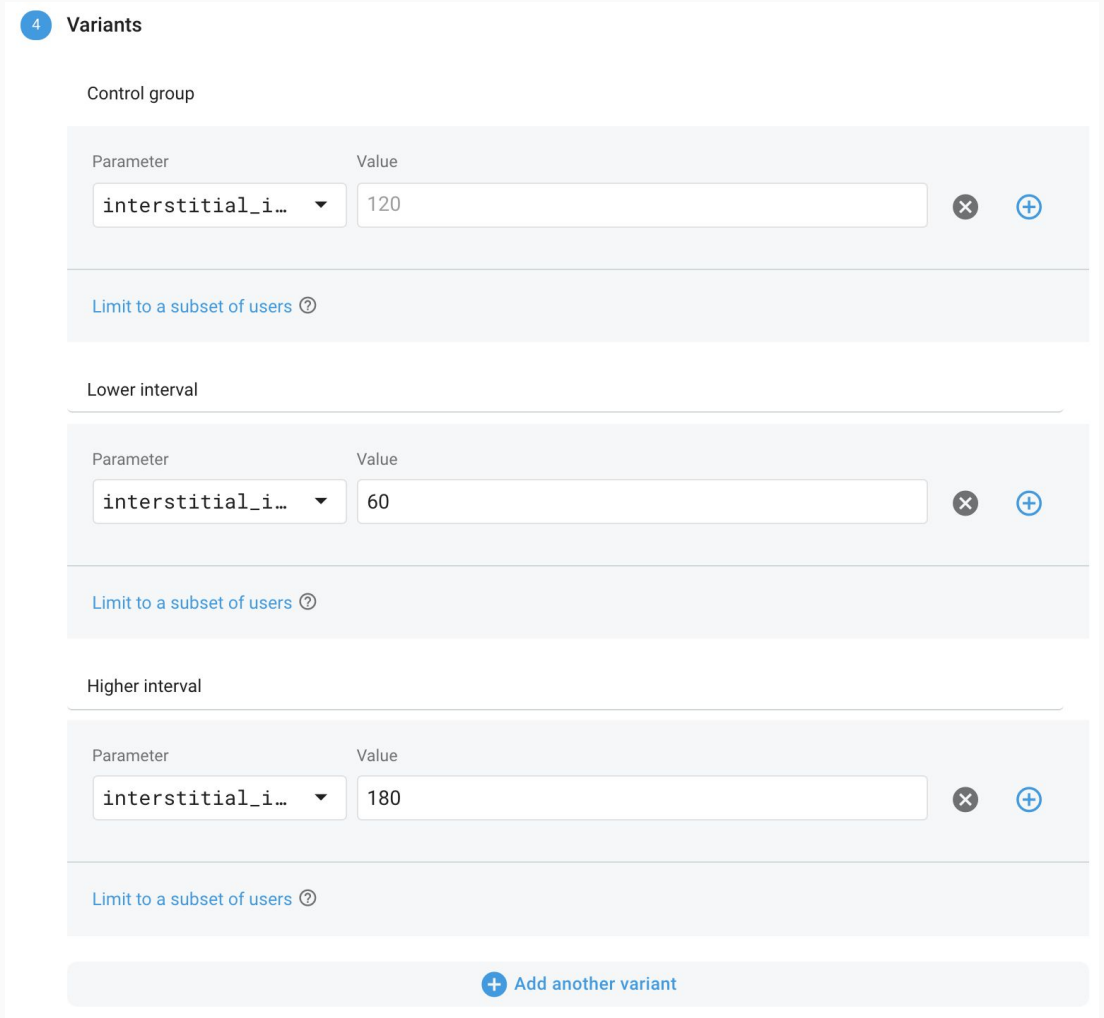
Next

Select the goal metrics

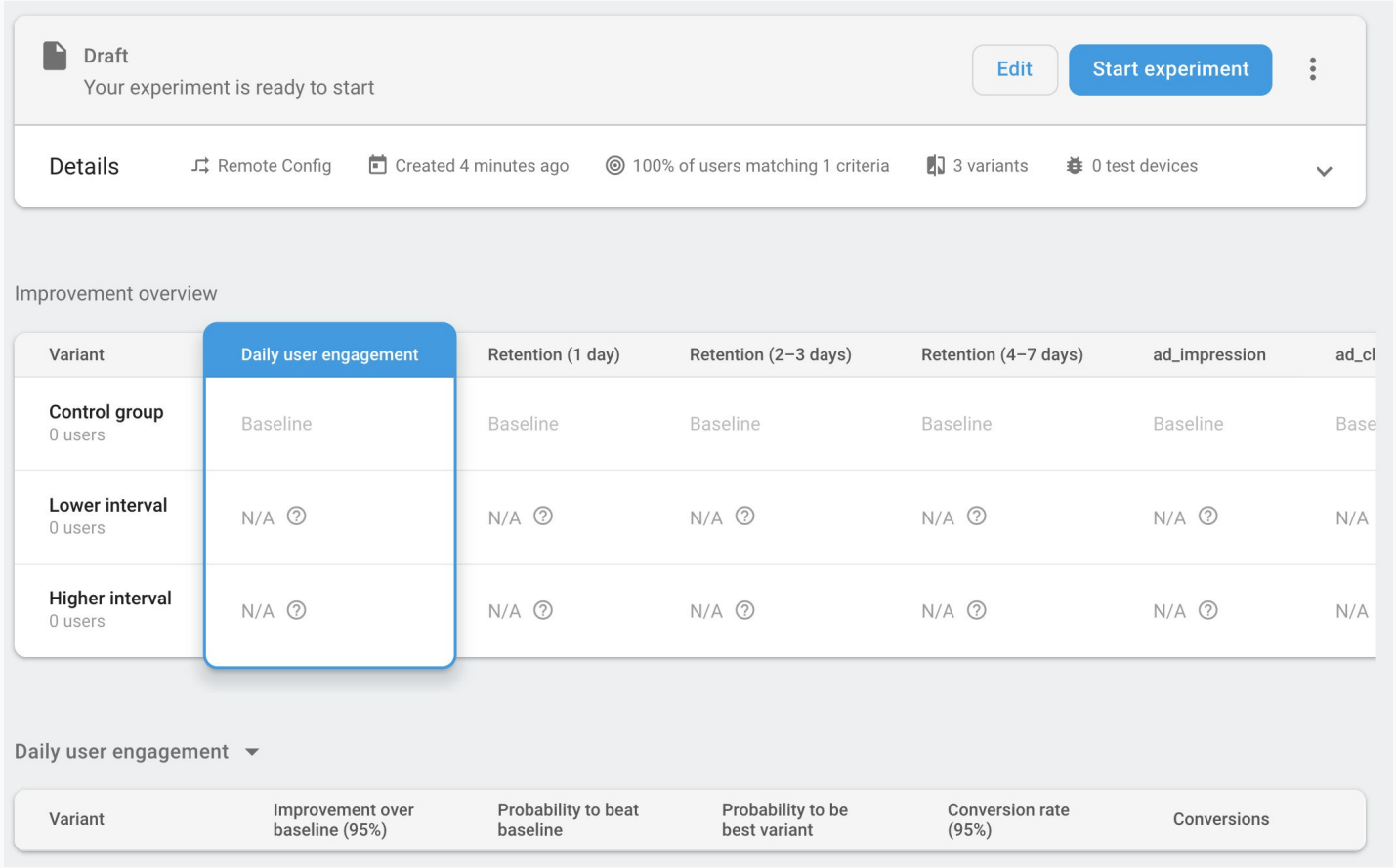


Create the variants

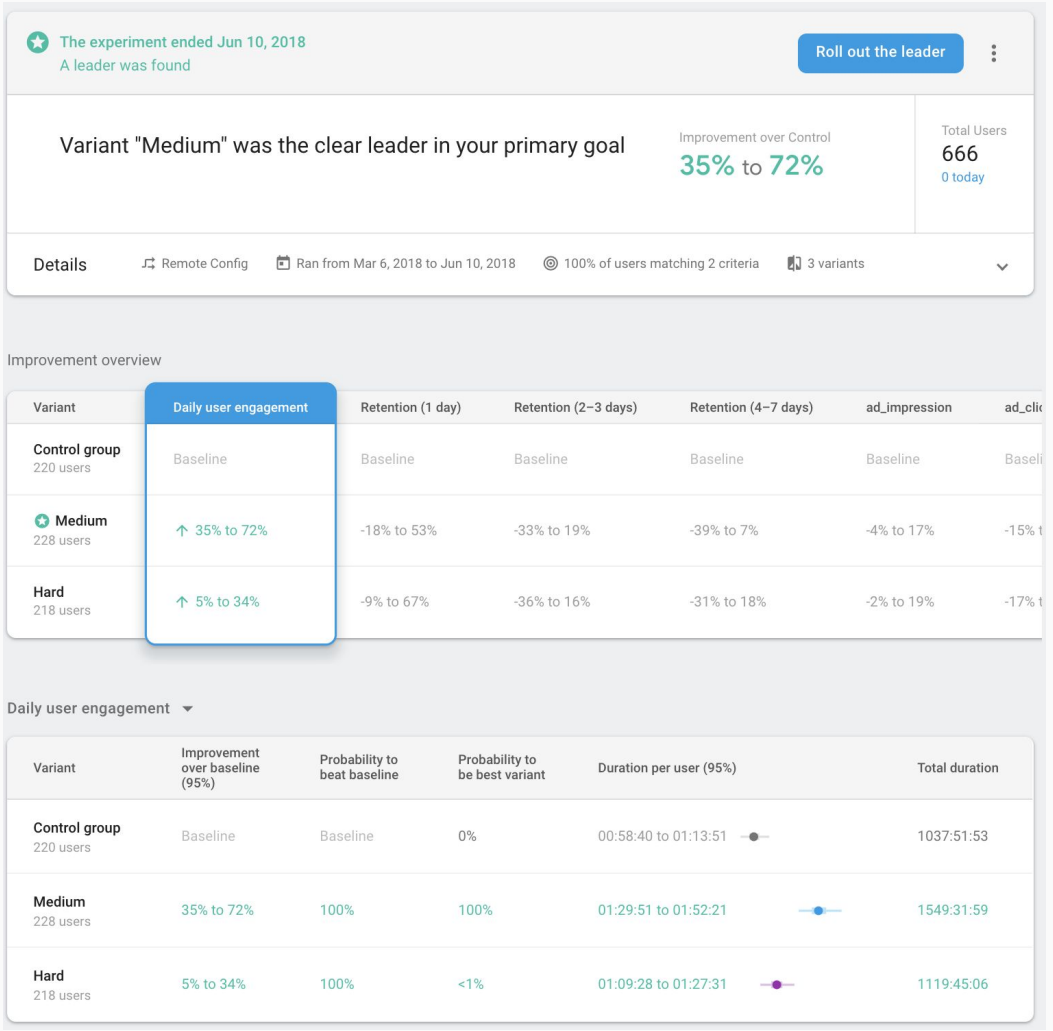
- Randomly divide users to three groups
- Change the value (higher, lower) of the interval
- You can use more conditions to limit the group to a subset of users (eg: use higher interval - less ads for tier 1 countries: Us, UK)



Check and run the experiment



Wait for result then roll out the best value



2. Choose the best placement

- Where (when) you show the interstitials
- Each place has different effect: viewability(show rate), CTR, UX
- Find the best place to put your interstitials

A/B test the Placement with Firebase

- Define all places you can show interstitial
- Create parameters on Firebase Remote Config and Ad units on Admob Console
- Get parameters in code and use them for control the Placements
- A/B testing the the placements with Firebase & Admob

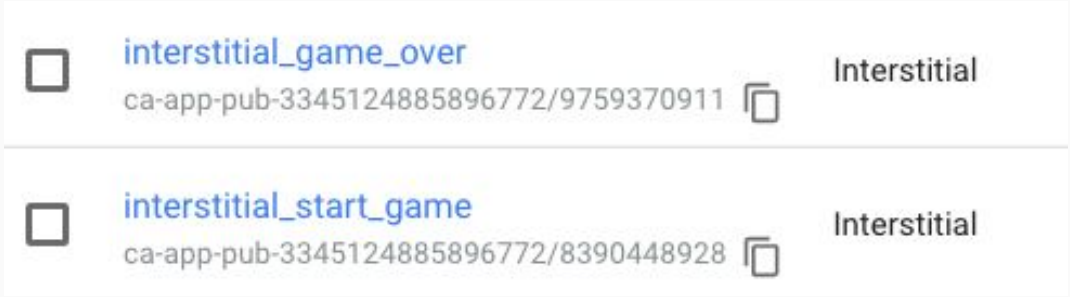
Create parameters on Firebase Remote Config

Show interstitial on two places:

- Start new game
- Game over

Create ad units

It'll be easier to check eCPM, CTR, etc for each placement if we use different Ad Unit ID for each placement.



Get parameters in code and use them for control the Placements

Check if we want to show interstitial when users start a new game

Get the control value by Firebase Remote Config

Check if we want to show interstitial when game over

Get the control value by Firebase Remote Config

```
private void onStartNewGame(){
    if(needShowAdOnStartNewGame()){
        showNewGameInterstitial();
    }
}

private void onGameOver(){
    if(needShowAdOnGameOver()){
        showGameOverInterstitial();
    }
}
```

```
private boolean needShowAdOnGameOver() {
    return FirebaseRemoteConfig.getInstance().getBoolean( s: "show_interstitial_game_over");
}

private boolean needShowAdOnStartNewGame() {
    return FirebaseRemoteConfig.getInstance().getBoolean( s: "show_interstitial_start_game");
}
```

A/B testing the the placements with Firebase & Admob

Three groups:

1. Show interstitial in both places
2. Show interstitial only when start new game
3. Show interstitial only when game over

Variants	show_interstitial_start_game	show_interstitial_game_over
Control group 33.3%	true	true
Lower frequency 33.3%	true	false
Higher frequency 33.3%	false	true

Note

1. You should test each one by one
2. Best way is test the frequency first, then test the placement
3. You can modify test for your target markets. Eg:
    a. Show less ad (higher frequency) in tier 1 country (US, UK)
    b. Show more ad in low tier countries

III. Signals of Interstitial problem on Admob Report

Match rate

Matched requests / AdMob Network requests.

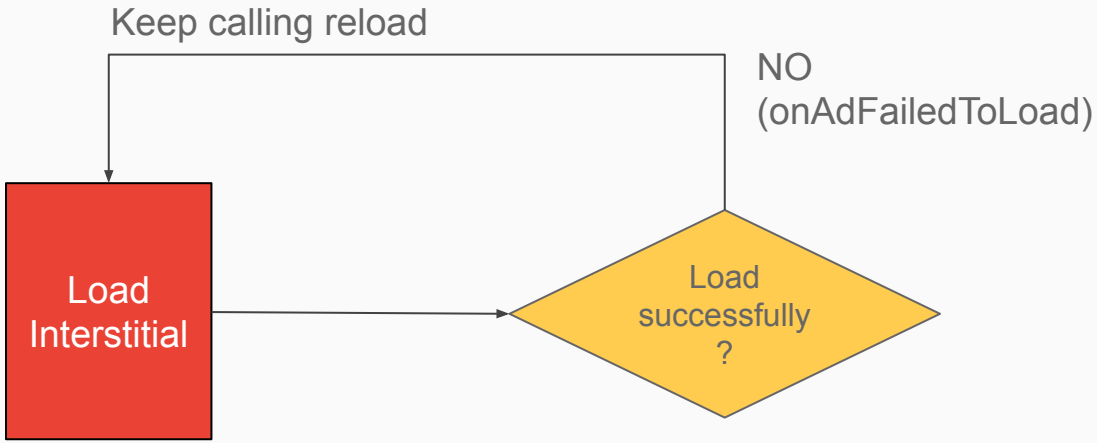
Show rate

Impressions / Matched requests.

Signals from match rate

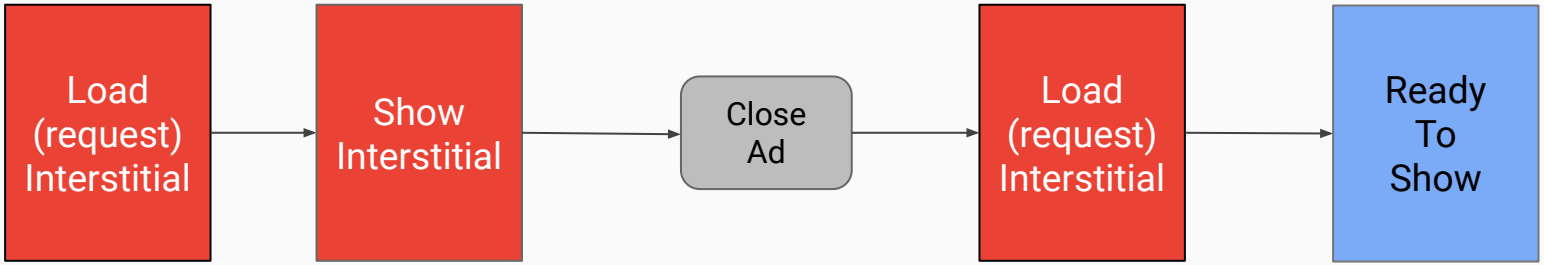
Match Rate

- Depends on the country and the format
- If the value is too low (less than 40%):
  - publishers may keep call re-load the ad too many times
  - floor settings, ad filter, MA content rating v.v.v



Should reload only one or two times

Signal from show rate



Impression = matched request - 1

Matched Requests	Impressions	Show Rate
2	1	50%
3	2	66.67%
4	3	75%
5	4	80%
6	5	83.33%

Show rate is too low

Less than 40%

- Ad has been preloaded, but users don't reach the placements (low viewability)
  - try to find more placements:
    - app open (follow the correct implementation part here)
    - app exit (follow the correct implementation part here)
    - breakpoints (between pages: follow the correct implementation part here)
  - use firebase A/B test to maintain user experience
  - take care of policy

- Use multiple ad units
  - can hurt app performance
  - should use only one ad unit for the whole app (except you need to A/B test the placement's performance). Check the best practice to control (preload & show) ad

- Preload too much (keep reload the preloaded ads)
  - check the best practice to control (preload & show) ad

Show rate is too high

More than 80%

- Show too many ads (more than 6 impressions/session)
  - should check their UX (user retention, user engagement time) by firebase
  - take care of policy
- No preloading ads: show ads right after done loading
  - show ads suddenly -> violate the policy
  - don't have ads ready to show -> reduce the chance to show ads
  - check the best practice to control (preload & show) ad
- Only show ads on app exit (load -> show -> exit app, no more load):
  - check policy
  - use Firebase to A/B test to show ads on more placements