

MASTER'S THESIS DEFENSE

Understanding and Detecting Platform-Specific Violations in Android Auto Apps

Moshood A. Fakorede

Advisor: Dr. Umar Farooq

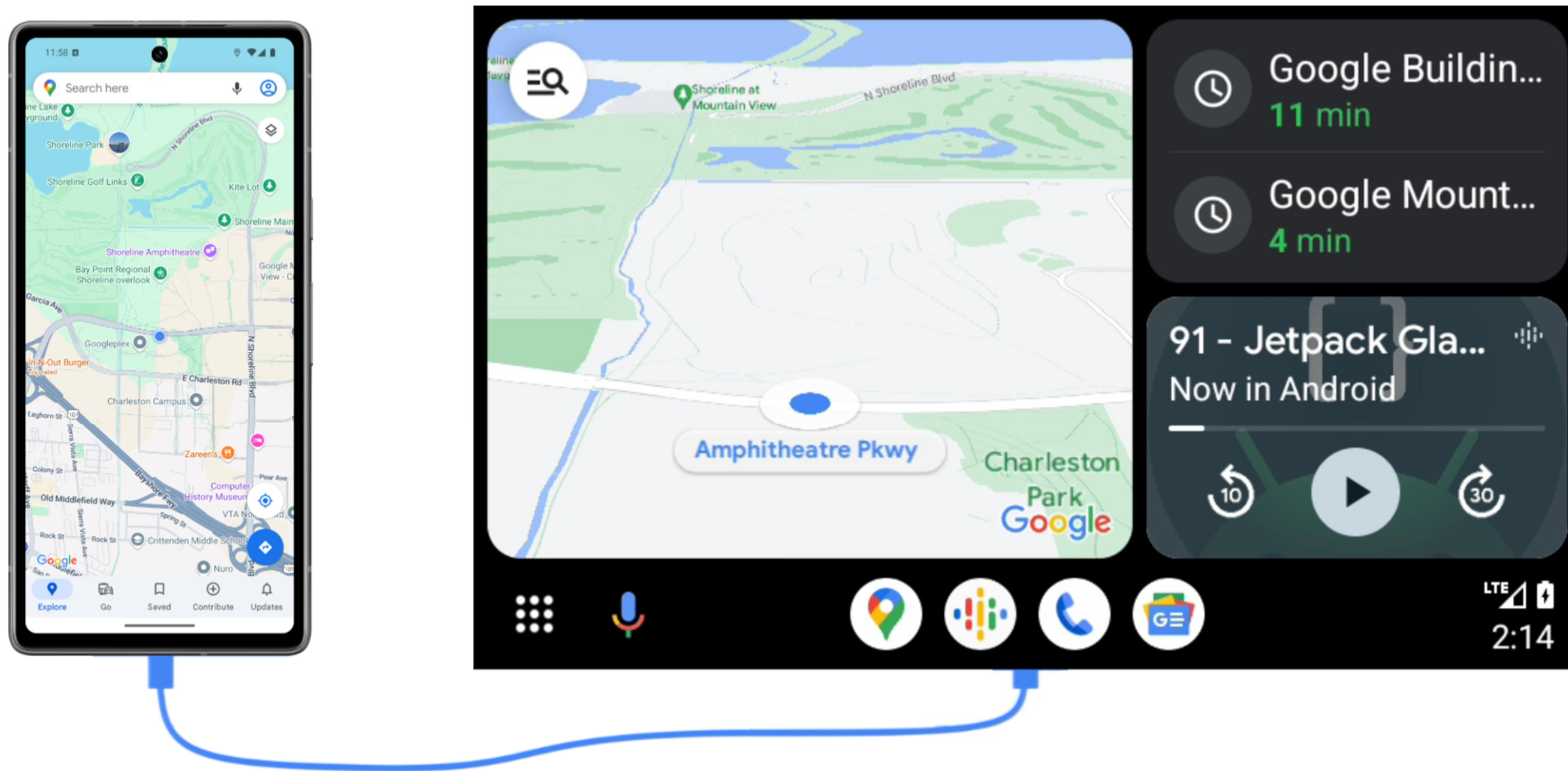
Committee: Dr. Gerald Baumgartner · Dr. David Shepherd

Division of Computer Science and Engineering

OCTOBER 21, 2026

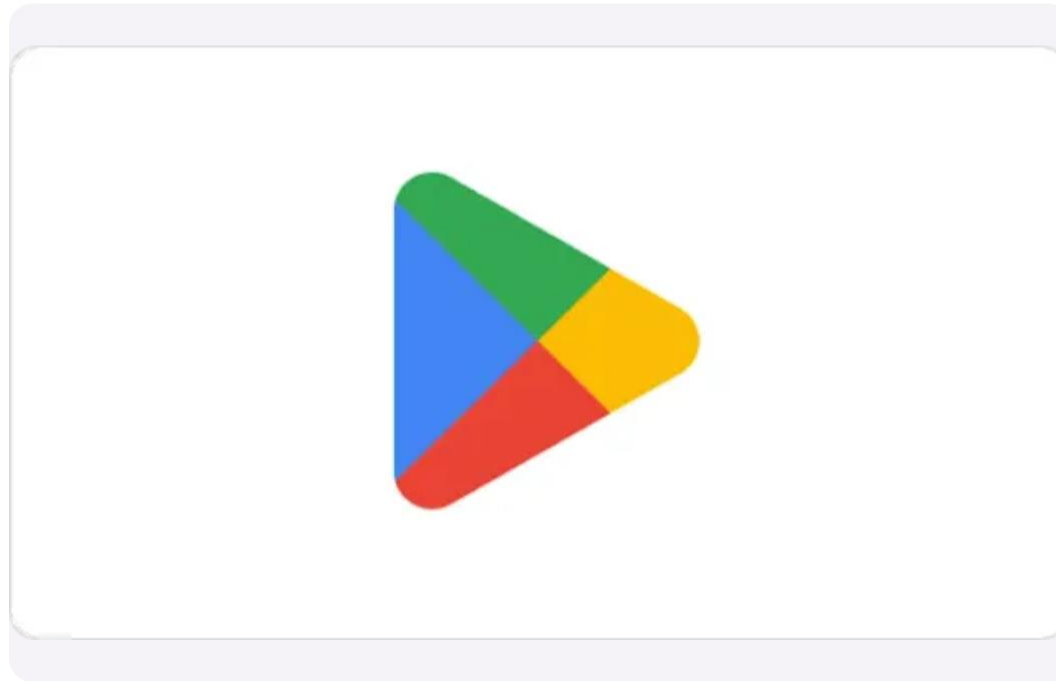
■ Published at the 7th ACM/IEEE International Conference on Automation of Software Test (AST '26), Rio de Janeiro, Brazil

Phone-Projected Apps



The app runs on the phone; the vehicle's head unit renders it and sends driver input back.

Mobile and Automotive



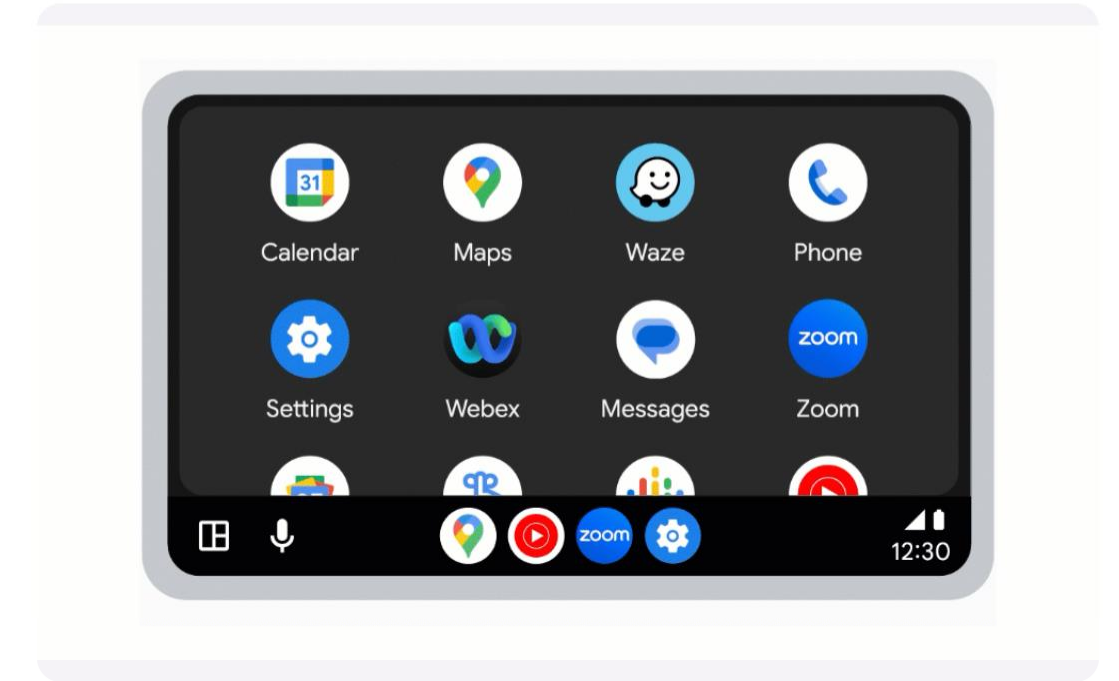
3.5M+

Android apps on Google Play



200M+

Android Auto-enabled vehicles (2024)



363

Android Auto-compliant apps (March 2025)

Developer Challenges

Android Auto: Update Refused #438

Closed



Paul Woitaschek opened on Apr 1, 2016

Owner

Rejected with following comment:

We found that voice commands break after two uses. Scenario: Audio is playing. User gives command. "Play chapter one". After your app confirms the command, audio will not resume, but the scrubber (line showing track progress) continues to move.

Further, when the voice button is tapped again the audio will play over the command prompt, which makes it impossible for the user to give a command, or stop the audio. This behavior would not be working from our users perspective.

Can someone confirm this?

How does the voice command work at all? When I use it in the emulator it always responds something like:

Don't know what you mean by 'pause music'.

I temporary removed the android auto declaration from the manifest. Does it still appear in your car?



Android Auto feature is not compliant #4205

Open



bmarty opened on Oct 11, 2021

Member

Android Auto feature has been added in 1.3.2, but The PlayStore has rejected the upgrade with this reason:

The app reads out the outgoing messages, the app should only read out the incoming messages

1.3.3 will disable Android Auto support, to unblock release 1.3.2, in the mean time let's investigate the problem

Related: [#4078](#)



Android Auto/Play Store submission issue #376

Closed



soncaokim opened on Feb 20, 2021 · edited by soncaokim

Edits

Collaborator

Submission of version 1.1.0 to the Play Store got rejected due to Android Auto issues. That's what it says:

Voice commands

Your app does not support all of the required voice commands. You can read more about how to support Voice Actions in the [Android Developers Help Center](#).

For example, your app recognizes the voice command, but does not start the requested track in the Android Auto environment



Make app full compliant Android Auto to publish in Play Store

Closed

Issue created Oct 3, 2022 by Óscar García Amor

Version 4.1.0 has been rejected in the Google Play Store.

As usual Google surprises us with another app rejection. This time they indicate the following:

Your app has not been accepted into Android Auto.

Voice commands

Your app does not support all of the required voice commands. You can read more about how to support Voice Actions in the [Android Developers Help Center](#).

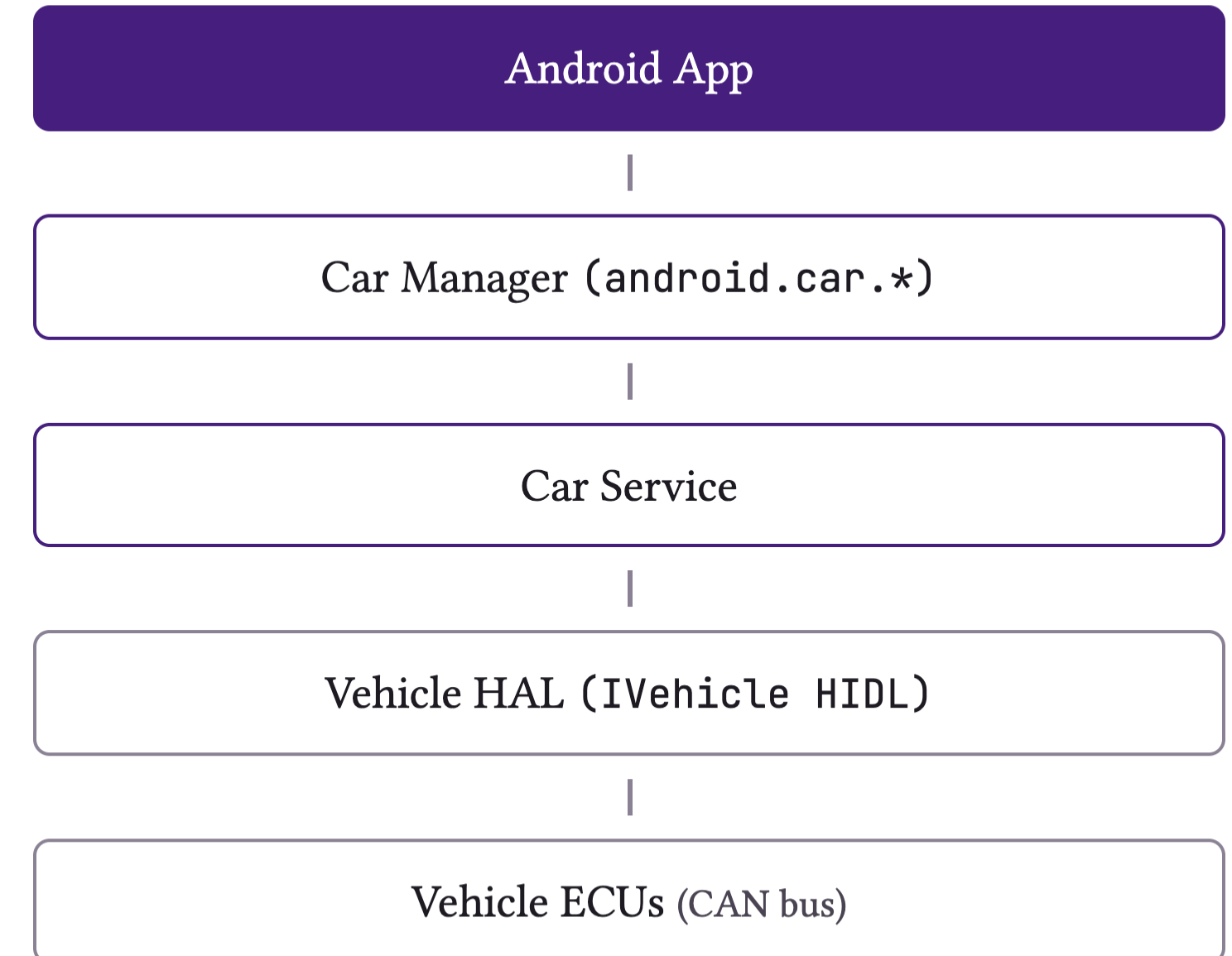
For example, your app does not switch audio tracks when recognizing and responding to a voice command intent. When being asked to play "X," the app remains on "Y." The app should switch over and play "X."

As a result of this rejection, version 4.1.0 cannot be published in the Play Store. The proposed solution is to either support voice commands or directly remove Android Auto.

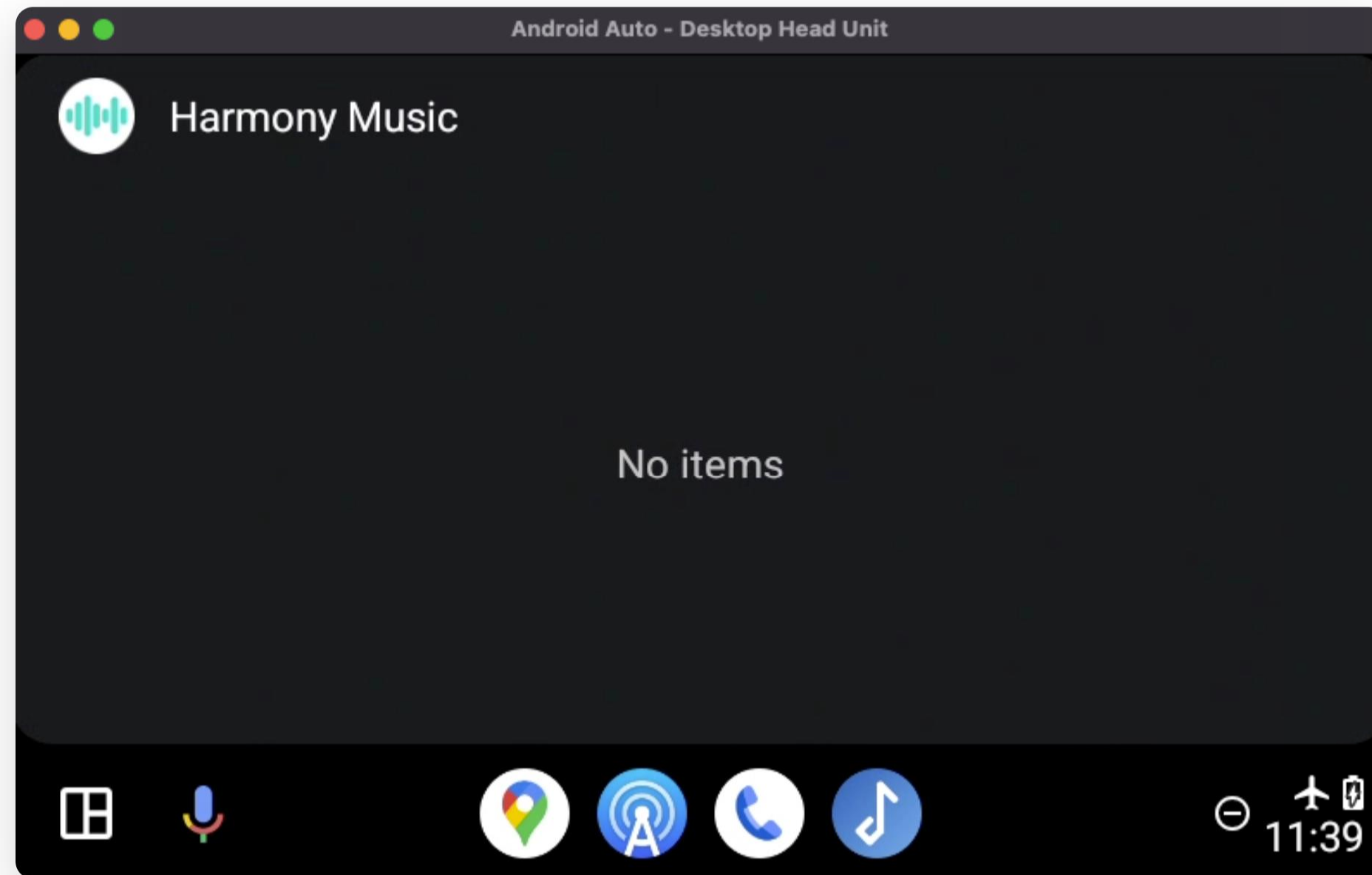
- Play Store rejections, and no tooling to catch them before submission

Android Auto Requirements

- 1 Discoverability**
Manifest declares automotive metadata and service registrations
- 2 Voice Integration**
Required voice command APIs for hands-free use
- 3 UI Compliance**
Designated templates; content hierarchy via MediaBrowserService
- 4 Functional Requirements**
Category-specific callbacks, especially for media playback



Anatomy of a Silent Failure



Harmony-Music #111 on the Desktop Head Unit

Build

Compiles, no crash

Phone

Library displays

Car screen

Empty

Motivating Example

Problematic: blank car screen

```
public class MusicService
    extends MediaBrowserServiceCompat {
    @Override
    public void onCreate() {
        super.onCreate();
        // Initializes MediaSession only
    }
    // CRITICAL: Missing onGetRoot()
    // and onLoadChildren()
    // RESULT: no UI hierarchy
}
```

Corrected: host-invoked callbacks

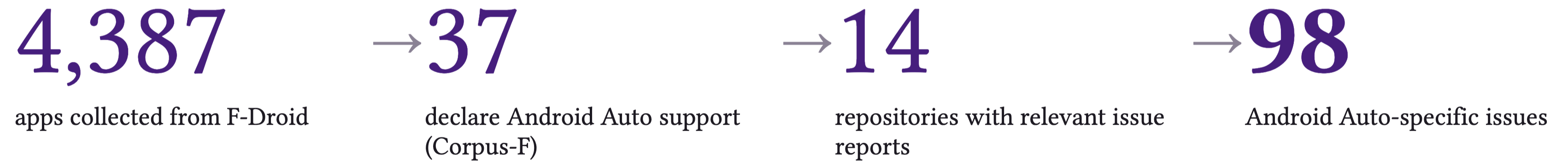
```
public class MusicService
    extends MediaBrowserServiceCompat {
    public void onCreate() { ... }
    @Override
    public BrowserRoot onGetRoot(...) {
        if (isValidAutoClient(...))
            return new BrowserRoot(MEDIA_ROOT_ID, null);
        return new BrowserRoot(EMPTY_ROOT_ID, null);
    }
    @Override
    public void onLoadChildren(...) {
        ...
        result.sendResult(mediaItems);
    }
}
```

PART 1

Formative Study

RQ1 What are the most prevalent types of Android Auto compliance violations, and what platform-specific factors contribute to their occurrence?

Study Corpus



Issue search keywords

Android Auto

car screen

car app

automotive

MediaBrowserService

MediaSession

voice command

Google Assistant

Analysis Methodology

RETENTION CRITERIA

Fails only in the Android Auto environment

Not reproducible on phones

Involves Android Auto APIs

MediaBrowserService, MediaSessionCompat, voice intents

Concerns a mandatory requirement

Not an optional feature

CLASSIFICATION BY TWO AUTHORS

- 1 Propose categories independently
- 2 Merge and refine definitions together
- 3 Re-classify all 98 issues independently
- 4 Resolve disagreements by discussion

Violation Taxonomy



T1 · 59 issues

Media Playback

MediaSessionCompat callbacks and playback control



T2 · 31 issues

User Interface

MediaBrowserService callbacks and content hierarchy



T3 · 8 issues

Voice Commands

Voice action intents and Google Assistant integration

Root Causes

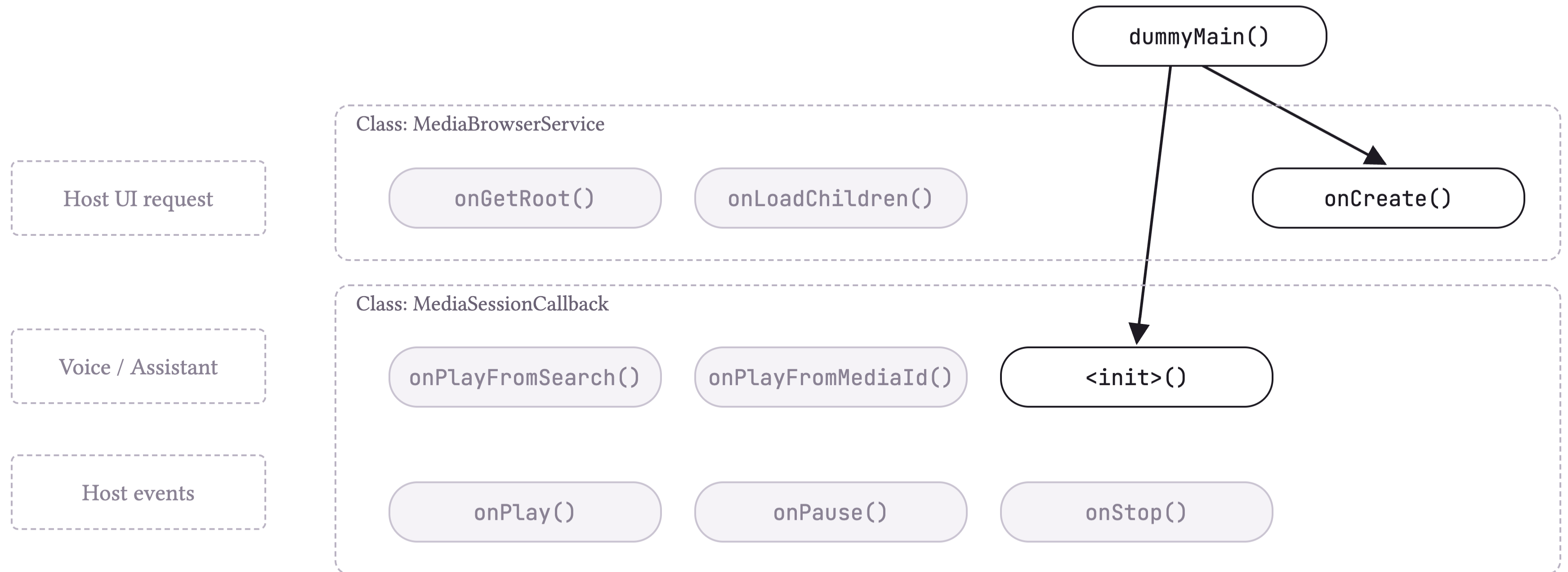
TYPE	ROOT CAUSE	EXAMPLE	SYMPTOM
T1 Media	Incomplete callbacks	AntennaPod #2380	Episode list shown instead of the next episode
	State management errors	Vinyl #348	Crash when liking a song from the car
T2 UI	Missing callbacks	Harmony-Music #111	No user interface on the car screen
	Incorrect hierarchy structure	Podcini #57	Car UI broken for several releases
T3 Voice	Missing intent filter	Ultrasonic #827	Voice command does not switch track
	Unimplemented callback	Vinyl #376	Command recognized, track never starts

Answer to RQ1

Violations fall into three categories matching the platform's core requirements: media playback control (60.2%), UI content provision (31.6%), and voice command handling (8.2%).

Most stem from the inverted control model: the vehicle, not the app, initiates the callbacks.

Why Traditional Analyses Fail



Host callbacks are not modeled as entry points

Platform invocation order is ignored

Compliance violations go undetected

Inverted Control Model

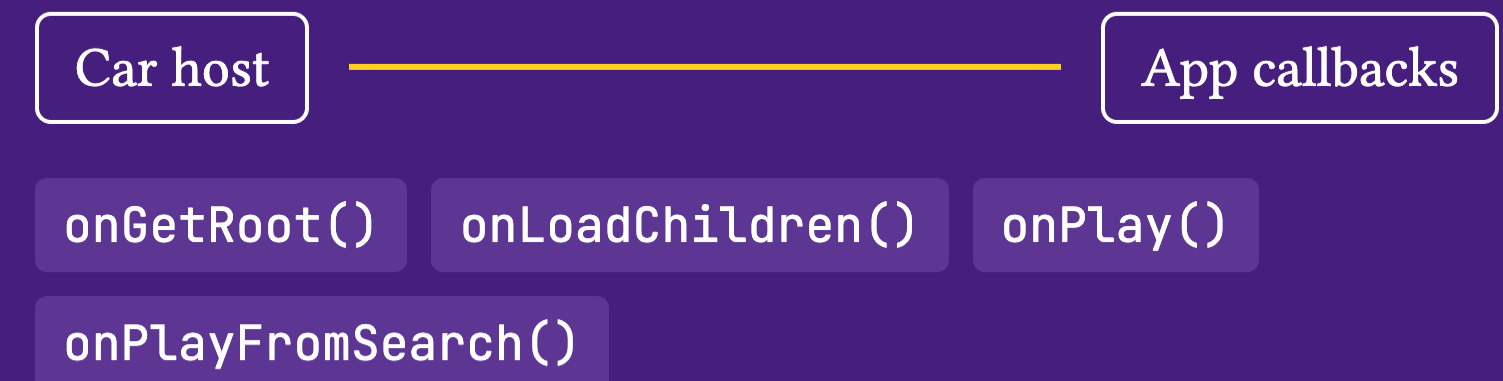
STANDARD ANDROID

The app drives its own UI through Activities



ANDROID AUTO

The vehicle invokes app callbacks to build the UI



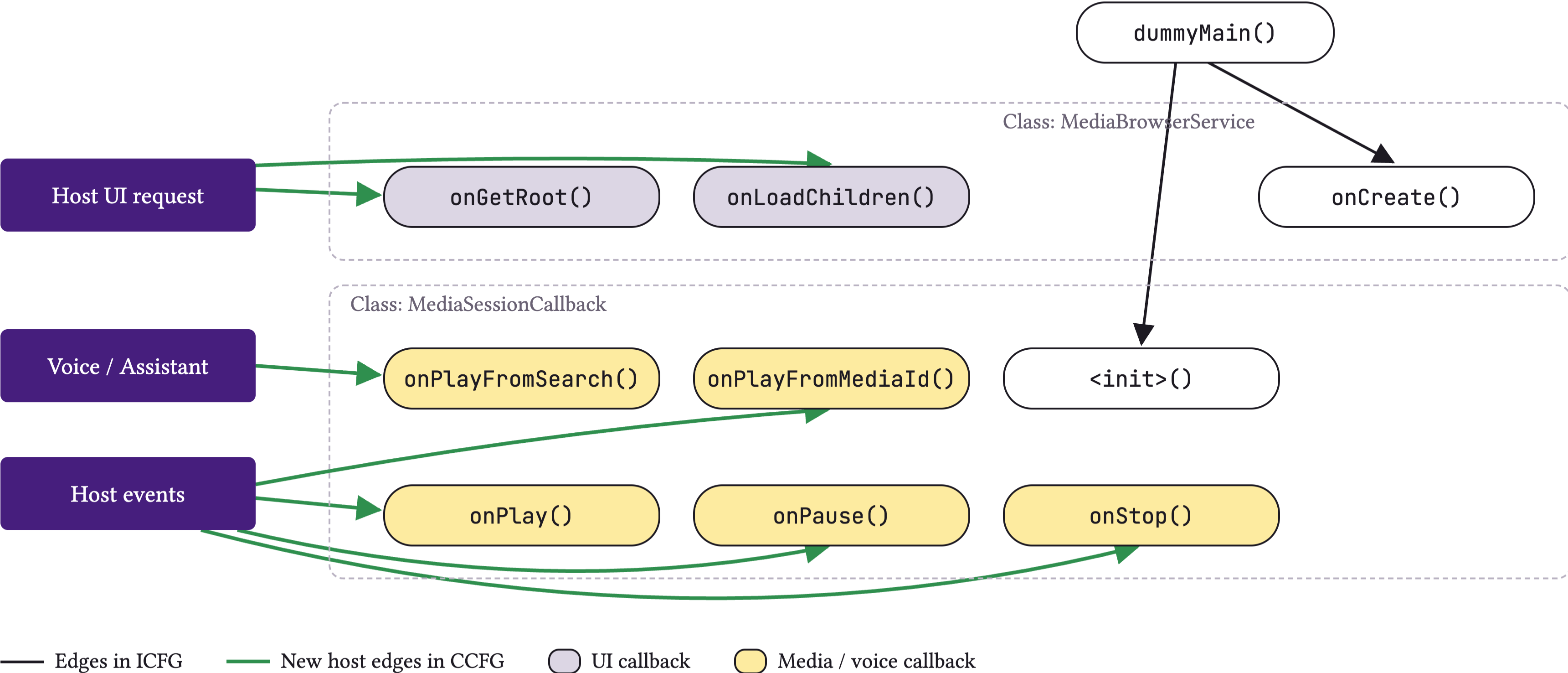
Callbacks are invoked even when the app's own UI is not visible.

PART 2

AutoComply

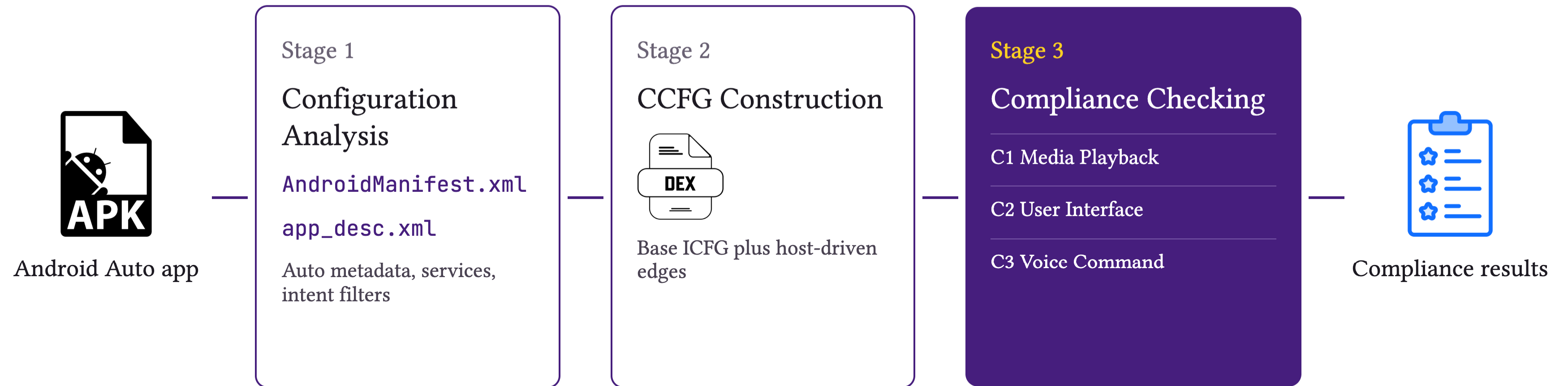
A static analysis framework for Android Auto compliance

Car-Control Flow Graph (CCFG)



$$G_{car} = (V_{base} \cup V_{host}, E_{base} \cup E_{host})$$

AutoComply Overview



Each checker verifies presence, connectivity, and semantic correctness of the required callbacks.

Compliance Analysis Algorithm

Input: APK · Output: compliance issues

```
1  autoMetadata ← ExtractMetadata(apk.manifest)
2  G_base ← ConstructBaseICFG(apk)
3  ccfg ← InitializeCCFG(apk)
4  requiredCallbacks ← InitializeCallbacks()
5  foreach autoComponent in autoMetadata.components
6      callbacks ← GetCallbacks(autoComponent)
7      AddCallbackNodes(ccfg, callbacks)
8      ConnectCallbackEdges(ccfg, callbacks)
9  end
10 foreach cb in requiredCallbacks
11     if ¬IsCompliant(ccfg, autoComponent, cb)
12         report auto compliance issue
13-14 end
```

Lines 1–9

Build the CCFG

Lines 10–14

Check each required
callback

C1 – Media Playback Checker

Invoked by dashboard and steering-wheel controls

REQUIRED CALLBACKS

- `onPlay()`
- `onPause()`
- `onStop()`
- `onSkipToNext()`
- `onSkipToPrevious()`
- `onPlayFromMediaId()`

CHECKS

All callbacks present in `MediaSessionCompat.Callback`

Each callback is connected in the CCFG

Dataflow reaches playback and resource-management logic

Error conditions and edge cases are handled

C2 – User Interface Checker

Invoked by host UI requests

REQUIRED CALLBACKS

- `onGetRoot()`
- `onLoadChildren()`

CHECKS

Both callbacks present with correct signatures

Every path in `onGetRoot()` returns a valid `BrowserRoot`

`sendResult()` is called in `onLoadChildren()`

Hierarchy is non-empty and well-structured

Client validation and error handling

C3 – Voice Command Checker

Invoked by Google Assistant voice commands

REQUIRED CALLBACKS

- `onPlayFromSearch()`
- `MEDIA_PLAY_FROM_SEARCH`

CHECKS

Intent filter declared in the manifest

Callback present with correct signature

Command parameters are validated

Playback state is updated

Unrecognized requests are handled

PART 3

Evaluation

- RQ2** Effectiveness: can AutoComply precisely detect compliance issues in real-world apps?
- RQ3** Comparison: how does it compare with existing tools in coverage and accuracy?
- RQ4** Usefulness: do developers acknowledge and fix the detected issues?

Evaluation Setup

31 6.5M

apps (Corpus-G) lines of code

SELECTION CRITERIA

Commits after January 2024

At least 30 stars, not a fork

Public issue tracker

No duplicates, libraries, or plugins

BASELINE

Android Lint, the only tool with built-in Auto checks

ANALYSIS MACHINE

MacBook Pro, Intel Core i7 2.7 GHz, 16 GB

VALIDATION

Google Pixel 7a (Android 14) in a real vehicle

RQ2: Effectiveness

27

violations detected

15

apps affected

0

false positives

4.2_s

geomean time per app

Answer to RQ2 AutoComply correctly identified 27 issues with no false positives or false negatives, all validated in a real vehicle.

RQ3: Comparison with Android Lint

	Android Lint	AutoComply	
Violations detected	2	27	13× more
False positives	0	0	
Geomean analysis time	11.1s	4.2s	2.6× faster

AutoComply also detected both issues reported by Android Lint.

Per-App Results

#	Repository	Stars	KLOC	Lint TP	Lint (s)	AutoComply TP	AutoComply (s)
1	Harmony-Music	1.4K	35.0	0	1	3	1
2	Satunes	38	138.8	0	1	2	18
3	refreezer	433	34.7	0	1	1	4
4	Musify	2.4K	9.5	0	1	3	2
5	jellyfin-android	1.7K	158.9	0	140	2	7
6	spotube	33.9K	68.2	0	1	3	2
7	LISTEN.moe android-app	258	90.4	0	30	2	3
8	namida	2.9K	92.9	0	1	2	3
9	nextcloud news-android	705	223.5	0	18	1	5
10	sweyer	209	27.5	0	1	3	2
11	Auxio	2.3K	47.9	1	198	1	6
12	quran_android	2.1K	58.2	0	138	1	6
13	Simple-Music-Player	1.3K	215.3	0	64	1	8
14	CuteMusic	363	30.4	1	119	1	2
15	Shuttle2	208	295.5	0	315	1	21
Total / geomean time				2	11.1	27	4.2

RQ4: Usefulness

27

violations reported on GitHub



14

acknowledged by developers



8

fixed and merged; 6 more scheduled

Reported violations by category

10 UI (T2)

7 Media (T1)

10 Voice (T3)

All fixes landed within one to four weeks; re-analysis confirmed zero remaining violations.

Developer Responses

Immediate fixes

Released updated versions and asked us to verify the fix

Acknowledged, planned

Appreciated the documented reports; fixes pending time and resources

Community help

Requested pull requests, using our working examples

Answer to RQ4 Developers found the violations actionable, and every fix was verified to resolve the detected issue.

Threats to Validity

EXTERNAL

Evaluated on Java and Kotlin apps

Cross-platform code (Flutter, React Native, C/C++) may hold compliance logic
AutoComply does not analyze

CONSTRUCT

Static checks only

Runtime performance and interaction patterns are not covered; obfuscation can affect accuracy

CONCLUSION

Corpus representativeness

Open-source apps may not reflect the full Android Auto ecosystem; runtime grows with app complexity

Future Work

Future Work

◆ Journal extension

Selected for the JSS special issue: *Bridging AI, Automation, and Sustainability in Software Testing*

SELECTED PAPER

Future Work

◆ Journal extension

Selected for the JSS special issue: *Bridging AI, Automation, and Sustainability in Software Testing*

SELECTED PAPER

◆ Lintbench

Evaluate LLMs on Generating Android Lint Checks

ACCEPTED · ICSE 2027

Future Work

◆ Journal extension

Selected for the JSS special issue: *Bridging AI, Automation, and Sustainability in Software Testing*

SELECTED PAPER

◆ Lintbench

Evaluate LLMs on Generating Android Lint Checks

ACCEPTED · ICSE 2027

◇ Privacy practices in Android Auto apps

Study sensitive data access and disclosure

IN PROGRESS

Contributions

Study	First study of Android Auto compliance challenges	98 issues · 3 categories
Model	Car-Control Flow Graph for host-invoked callbacks	UI, media, voice flows
Tool	AutoComply static analysis, evaluated on 31 apps	27 found · 14 confirmed · 8 fixed

Published at AST '26 · Selected for JSS special issue · Artifact: github.com/fakorede/autocomply

Services/Other Research Activities

PROGRAM COMMITTEES

ICSE 2027
Research Track Shadow PC

MSR 2027
Junior PC

Technical Debt 2027
Junior PC

ISSTA 2026
Artifact Evaluation PC

AWARDS

Distinguished Reviewer Award
ISSTA 2026 Artifact Evaluation

NSF Student Travel Grant

Thank You

Questions?

Moshood A. Fakorede · mfakor1@lsu.edu
github.com/fakorede/autocomply

SUPPORTED BY

