



GDG Rzeszów

Krzysztof Sowa

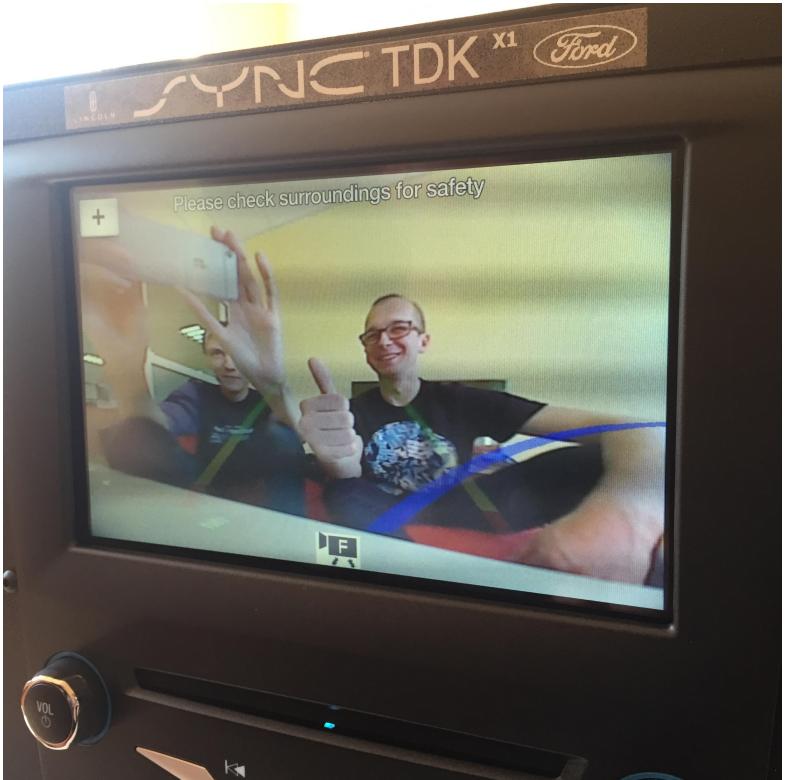
Android Auto

GDG Rzeszów #5

12 X 2016

Cześć ;)

Concise Software

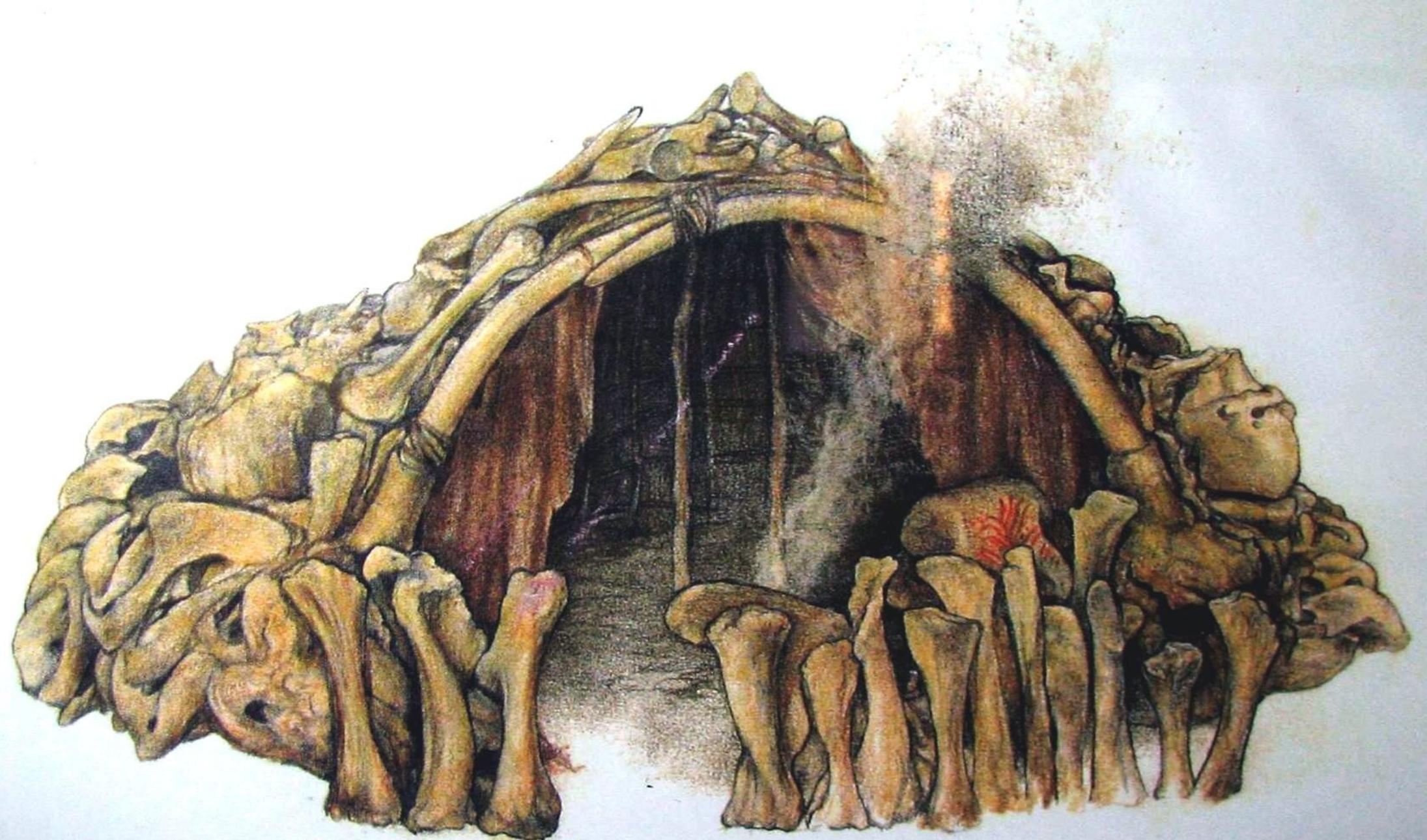


Concise Software



Concise Software









Mercedes-Benz works with Nest

Precondition your home to a comfortable temperature in time for your arrival.

Get reminded to set your Nest to Home or Away depending on your location.





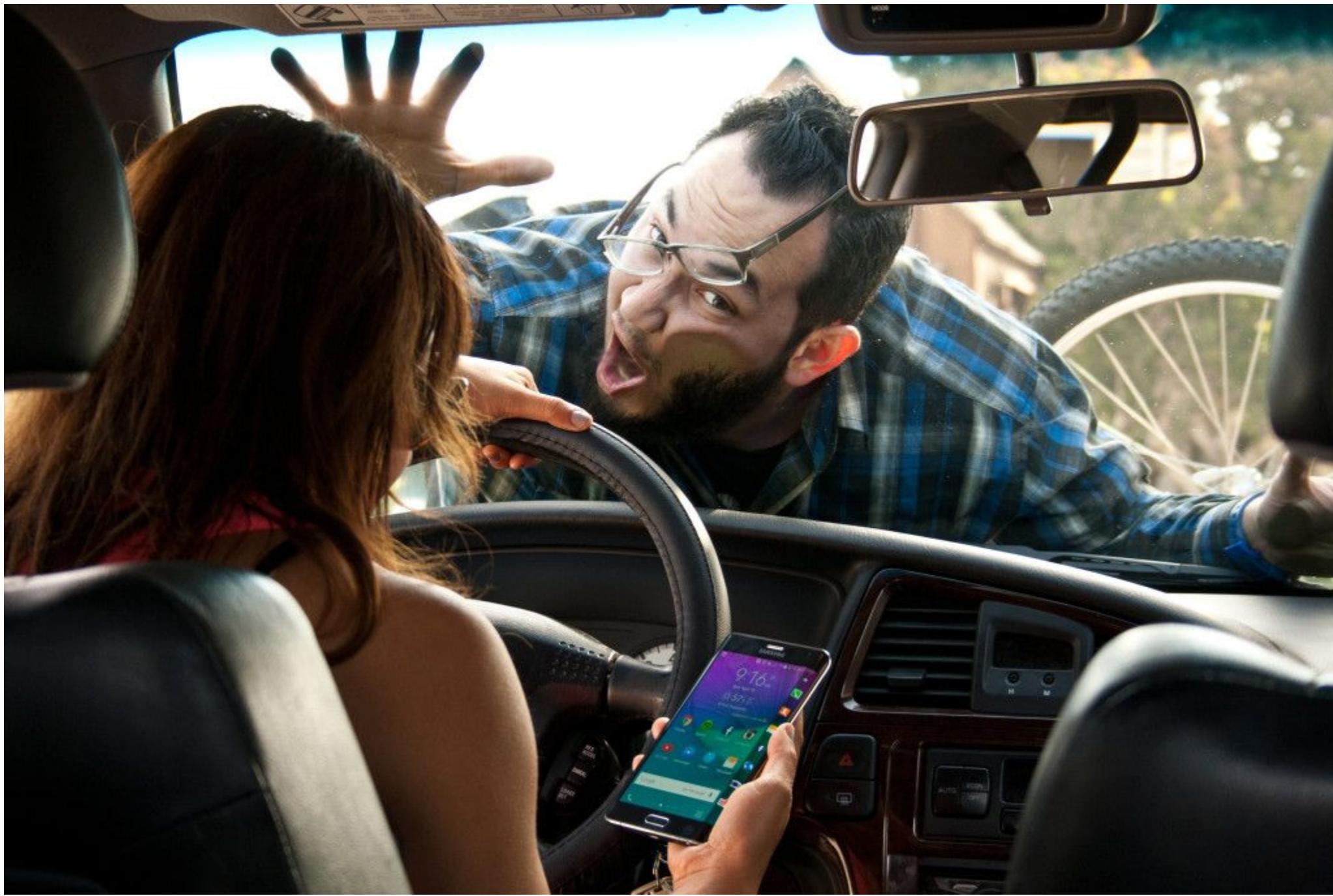
Saving lives through research and education

American Driving Survey 2014–2015

September 2016

“Drivers reported spending an average of **68.4 minutes driving** and driving an average of 43.2 miles on days when they drove.”

<https://www.aaafoundation.org/sites/default/files/AmericanDrivingSurvey2015.pdf>

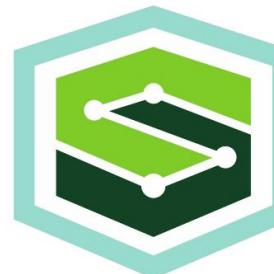


Connected car



Connected car

android
auto



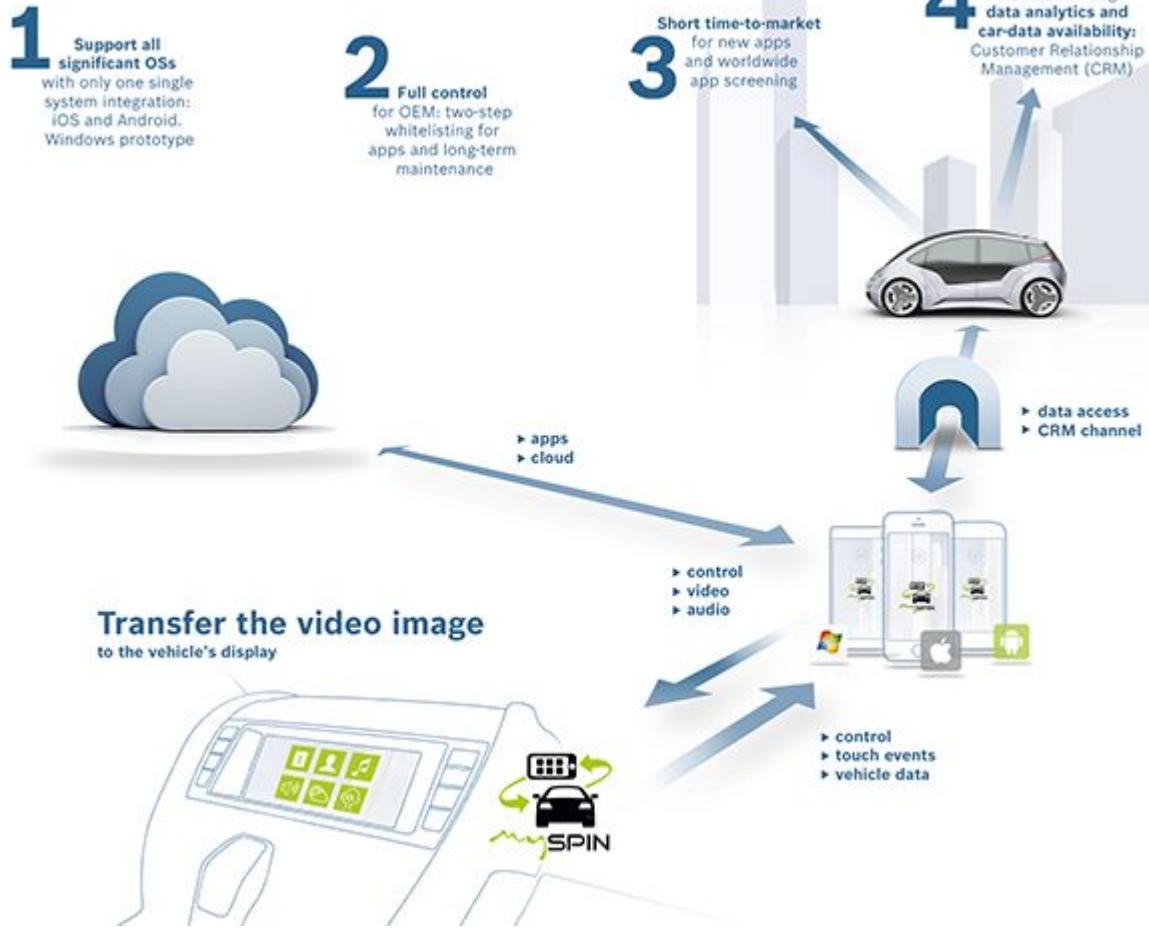
smart
device link

MirrorLink®

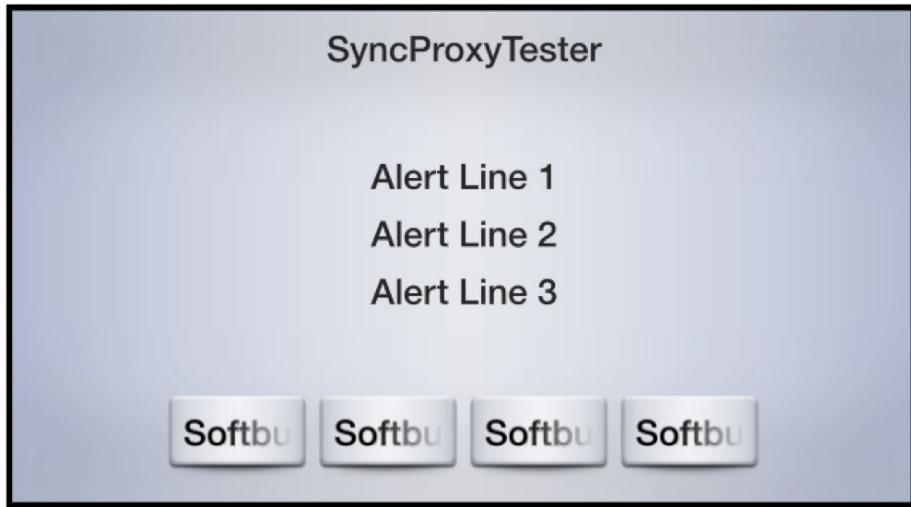


Apple CarPlay

Connected car



Connected car



Developers have challenges

- Many auto platforms
- Distribution difficulties
- Safety regulations
- Diversity of car hardware
- Connectivity + personalization

android
auto

You didn't have to be
automotive expert.

Android Auto



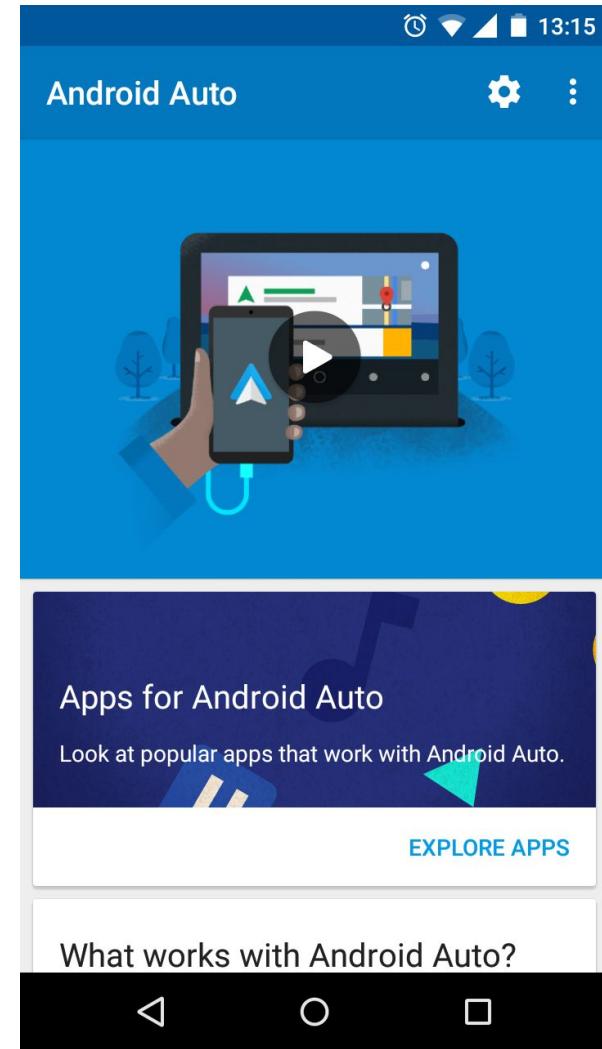
Designing interactive apps for cars is **fundamentally different than designing for handheld devices**. App content and interactions must augment the driving experience **without being distracting**.

Because driving is the primary activity in the car, the **UI must be simplified to keep the driver's eyes on the road and hands on the wheel**. Apps should never demand the user's attention, unless they are critical to the driving task.



Android Auto app

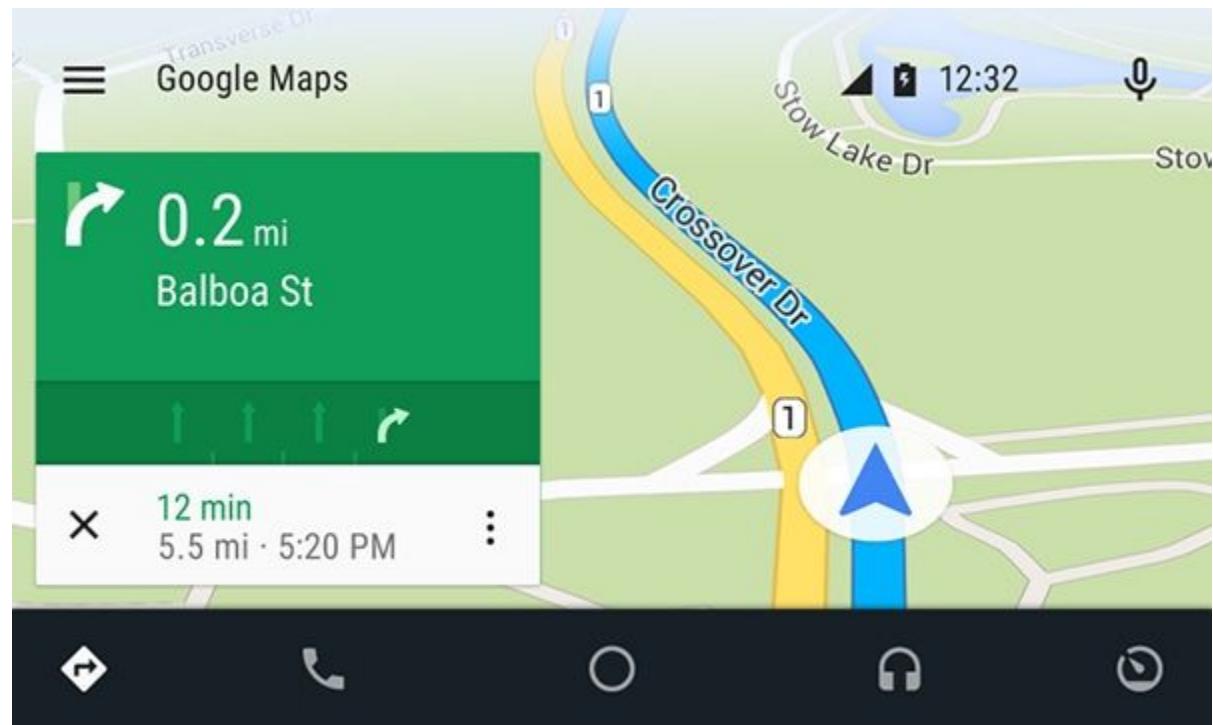
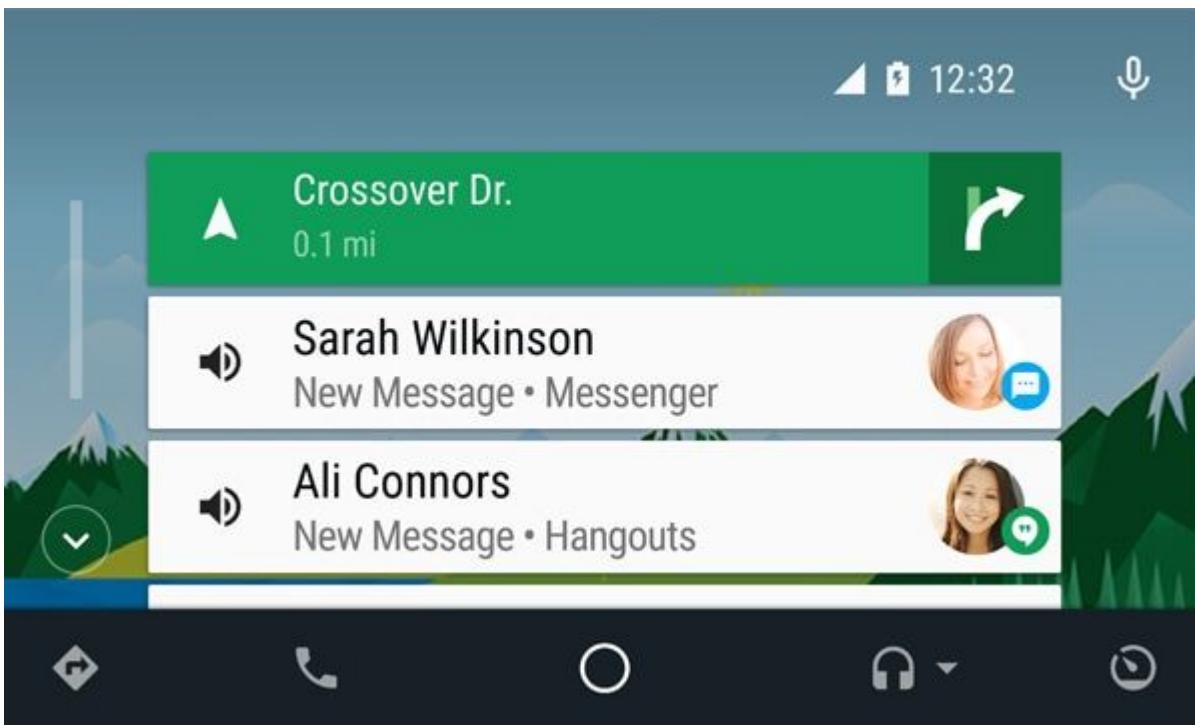
<https://play.google.com/store/apps/details?id=com.google.android.projection.gearhead>



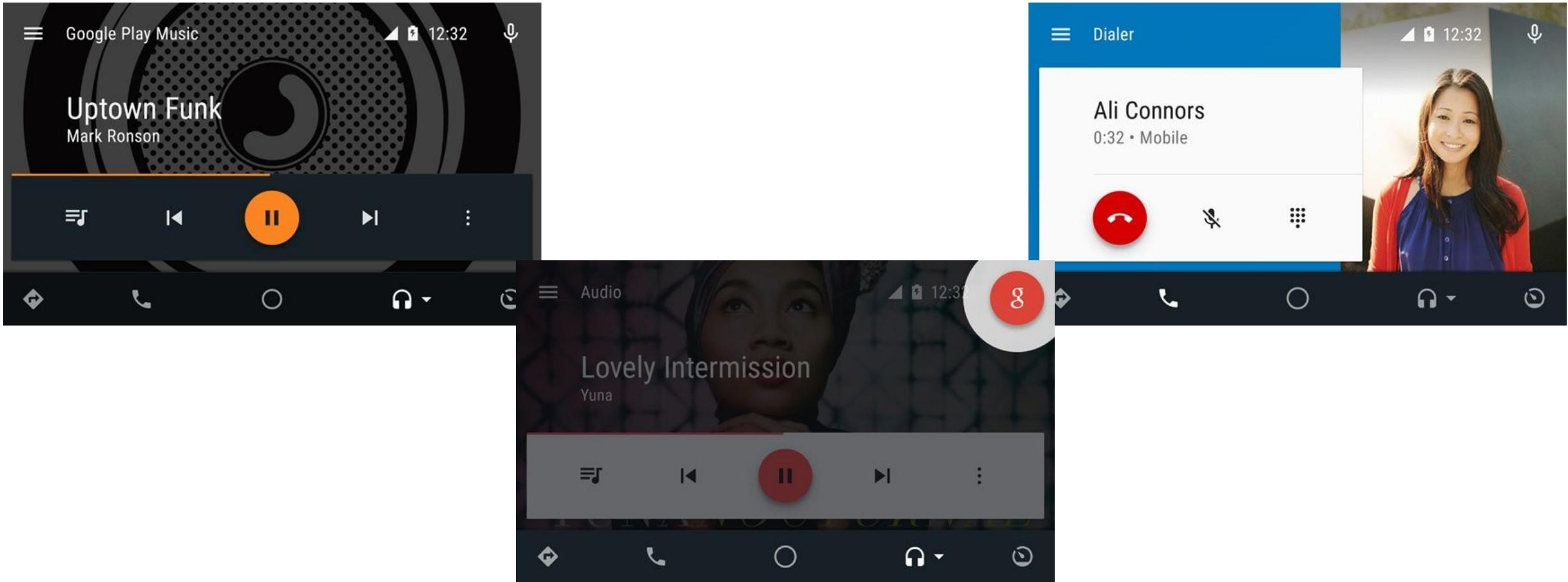
Android Auto - Lock screen



Android Auto

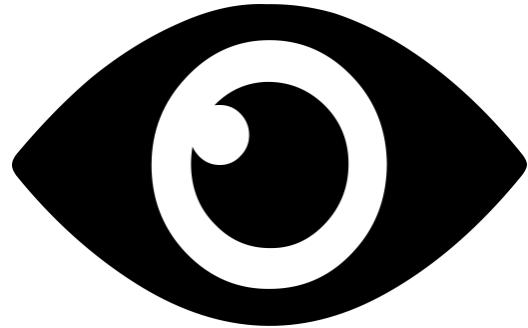


Android Auto

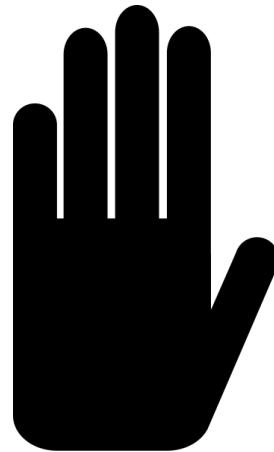


Driver distraction

Google takes driver distraction very seriously. The Android Auto framework is designed to adhere to safety and distraction guidelines from around the world.



Visual



Manual



Cognitive

We can't eliminate but we can mitigate.

Can Your Phone Actually Make You A Better Driver?



“We, as designers and researchers, we have ideas of what is the right solution for inside the car, but it’s amazing how often we’re surprised.”

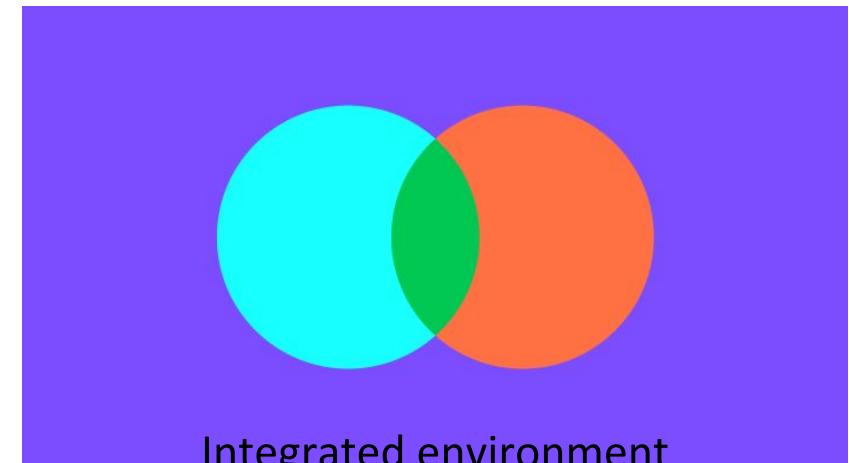
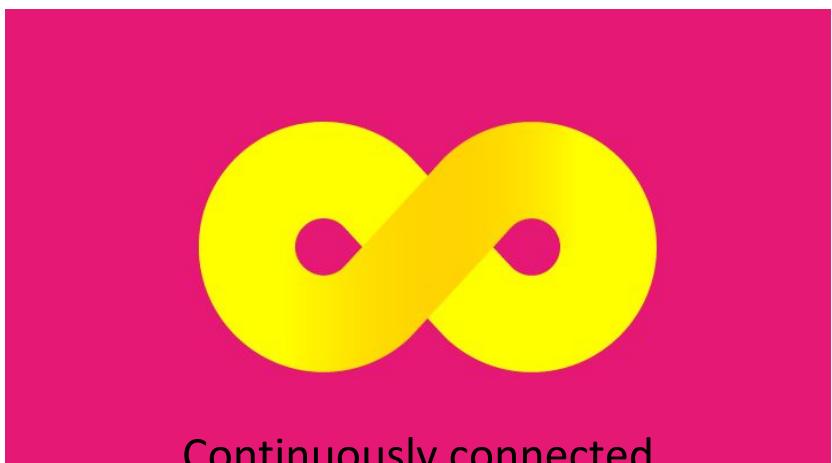
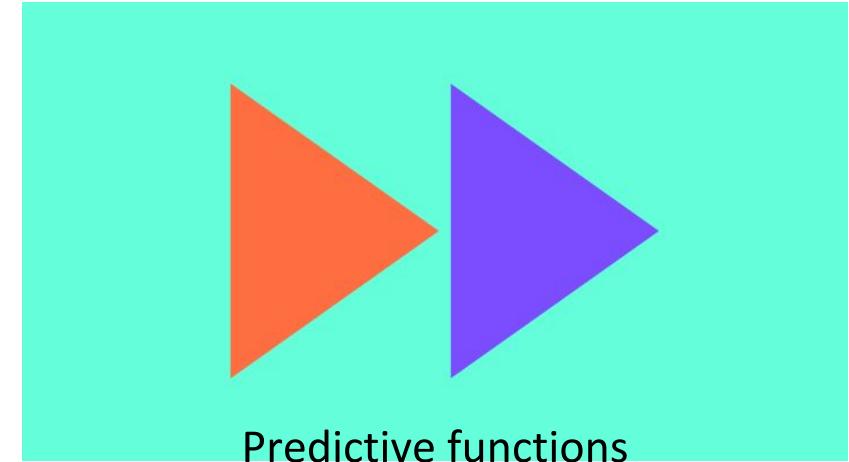
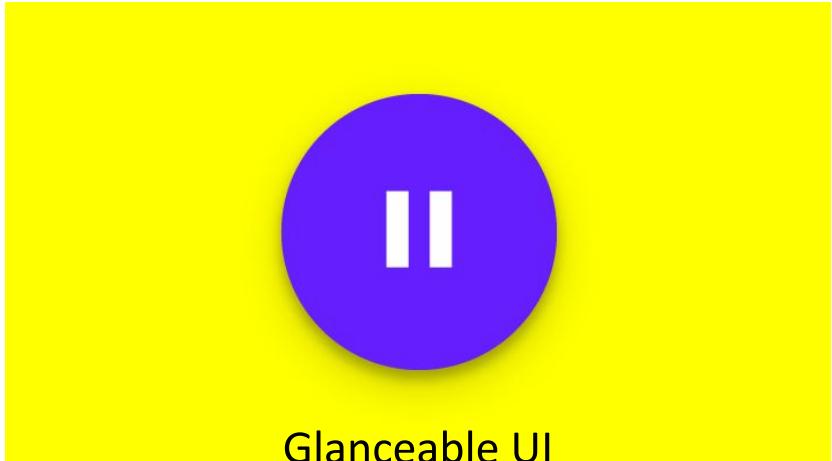
Design principles

App interfaces need to be **quick** and **easy** to navigate:

- Focus on primary actions and content
- Avoid interfaces that require focused attention

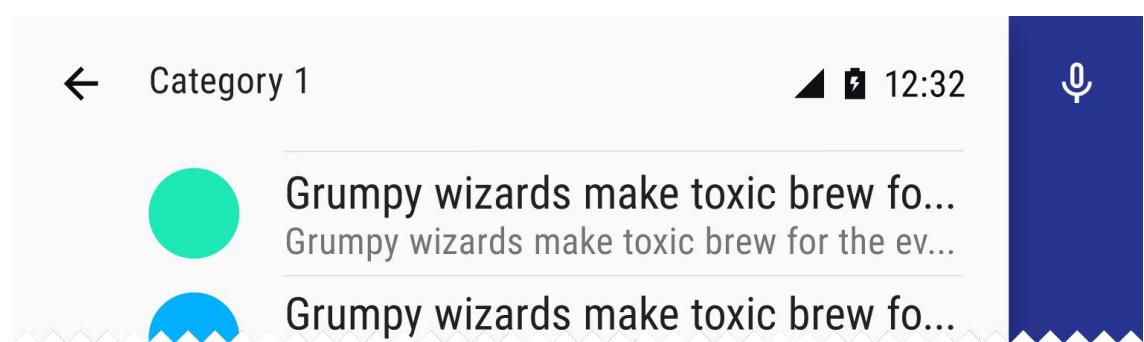
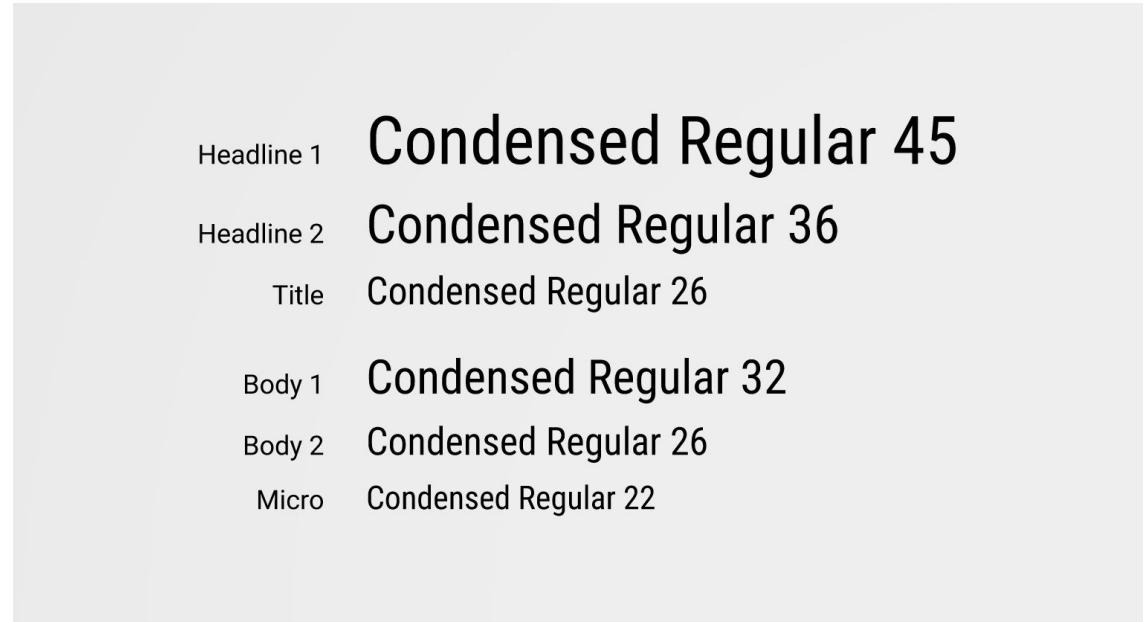
All Android Auto platform apps must pass a formal review and meet minimum safety and driver distraction requirements and regulations.

Auto apps must be predictable and intuitive so that drivers can keep their eyes on the road!



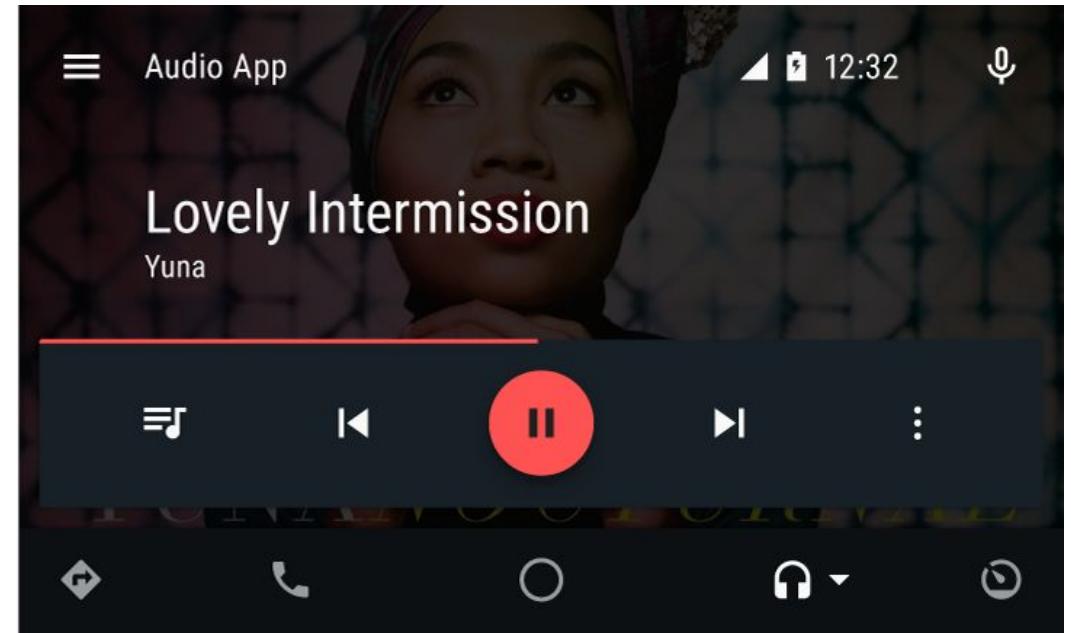
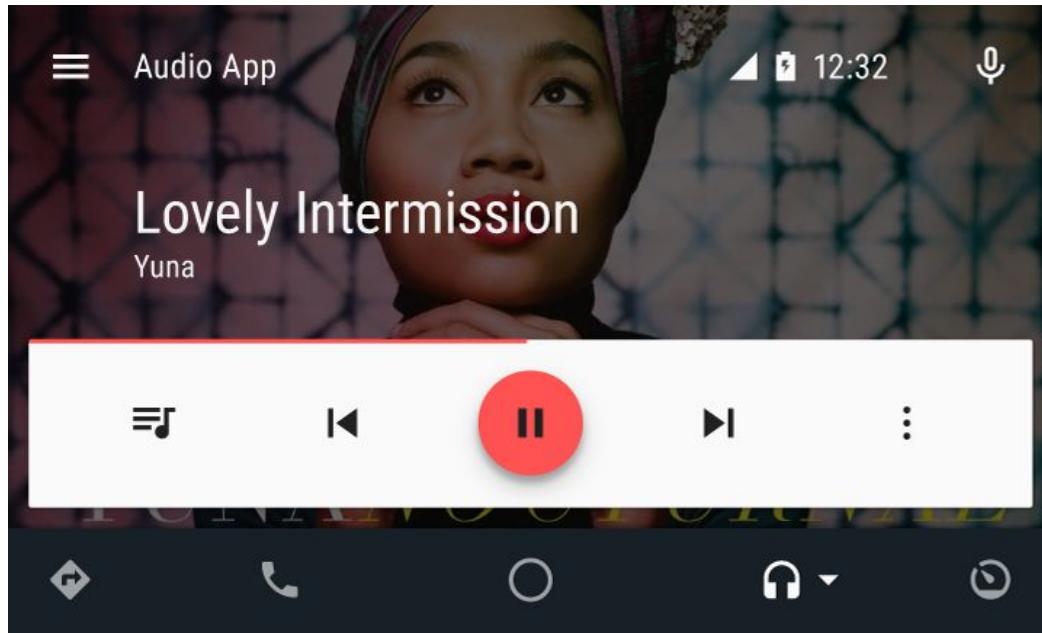
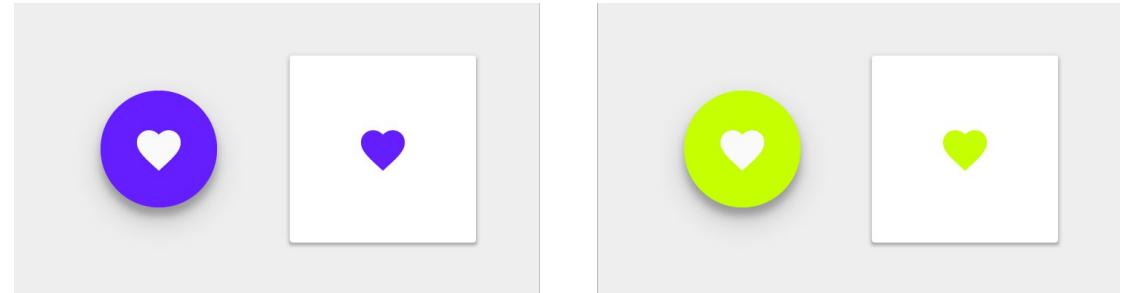
Minimizing driver distraction - Typography

- Fonts & sizing
- Text length
- Autoscrolling



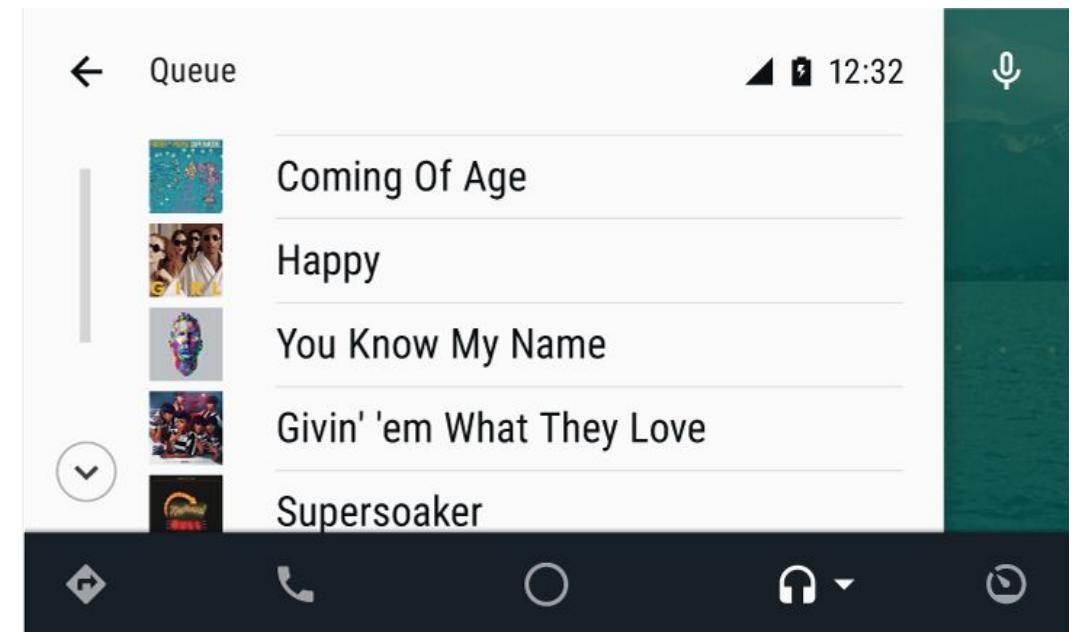
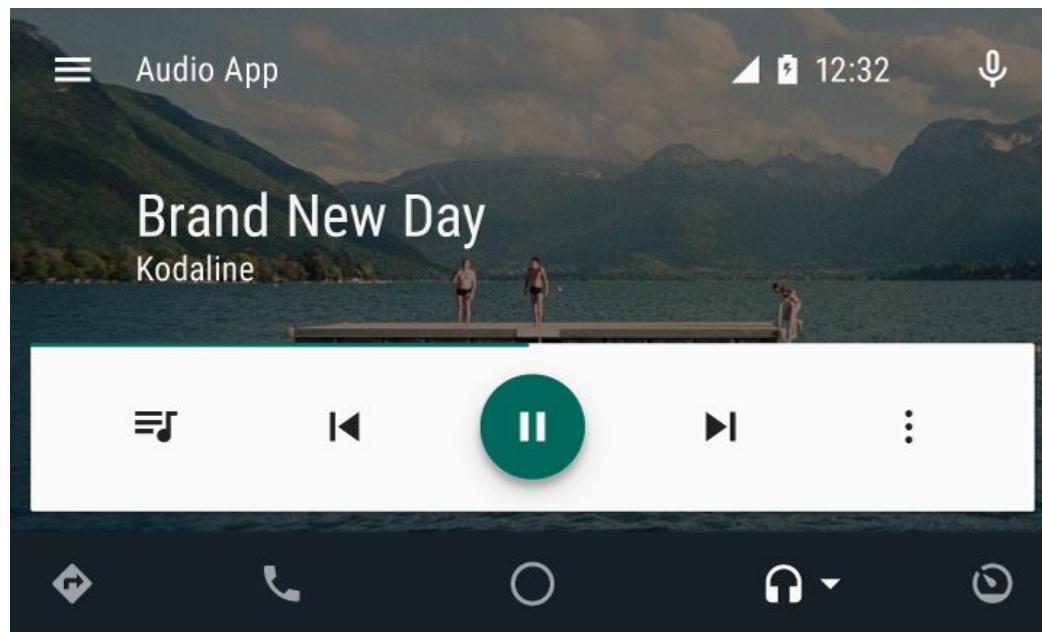
Minimizing driver distraction - Colors and contrast

- Contrast ratio
- Night vs. day modes

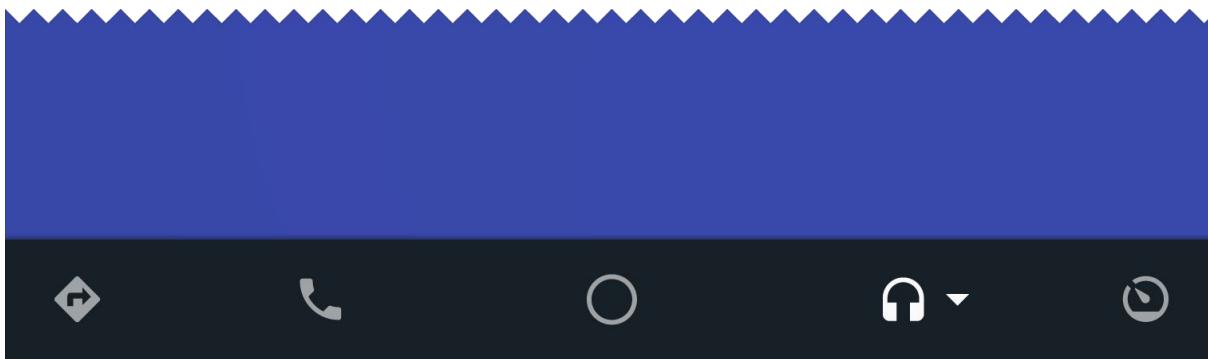


Minimizing driver distraction - Images

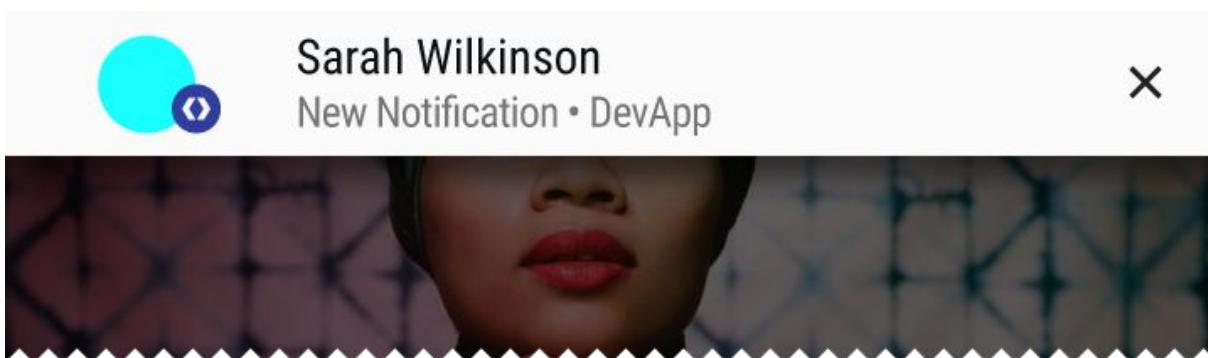
- Imagery & video
- Advertising



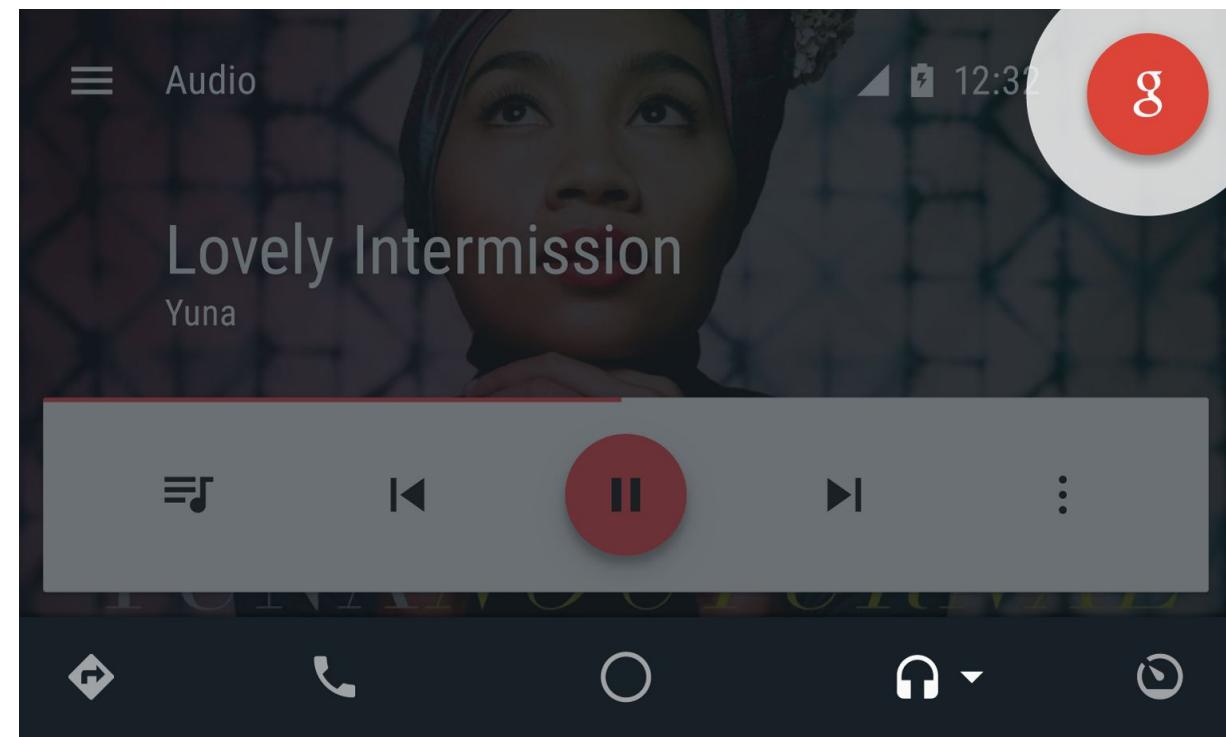
System overview



Activity bar and app switching



Notifications



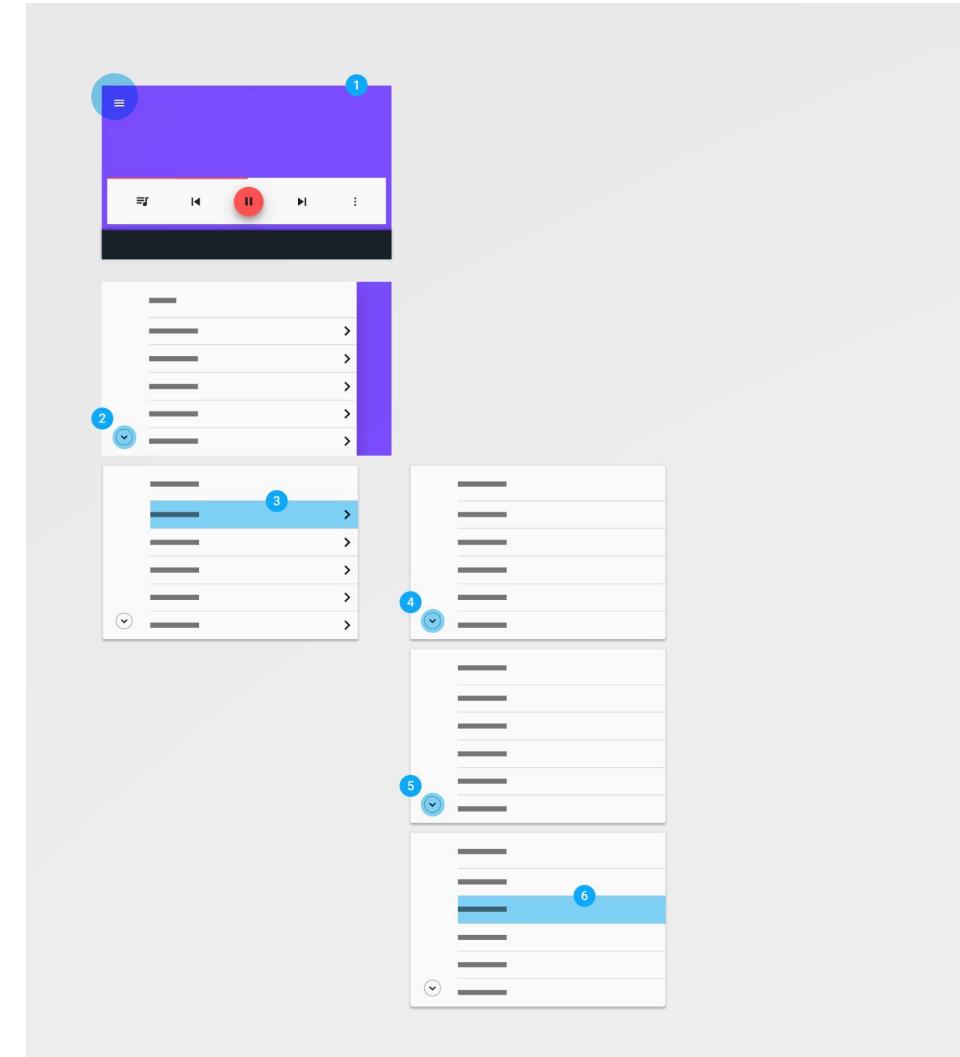
Voice
Demand layer

System overview - Drawer

- Focus on contextual, fresh, and useful items
 - Simplify your top-level drawer
 - Show most items without scrolling
 - Use icons to help identify items
-
- Complete actions in six steps (This number may differ for some countries. For example, four steps is the maximum in Japan.)
 - Show content after two levels of hierarchy
 - Drawers contain five items maximum

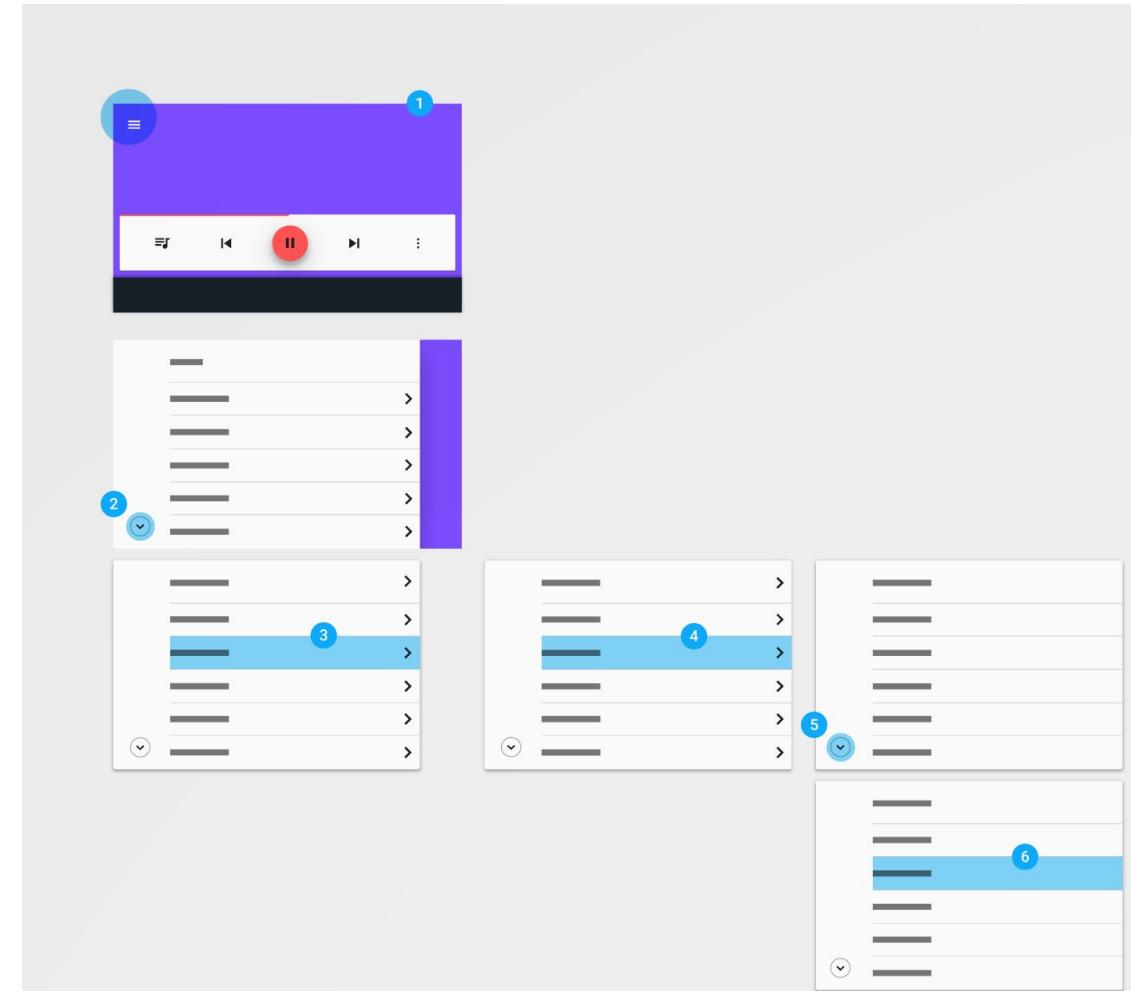
System overview - Drawer

- Focus on contextual, fresh, and useful items
- Simplify your top-level drawer
- Show most items without scrolling
- Use icons to help identify items



System overview - Drawer

- Complete actions in six steps (This number may differ for some countries. For example, four steps is the maximum in Japan.)
- Show content after two levels of hierarchy
- Drawers contain five items maximum

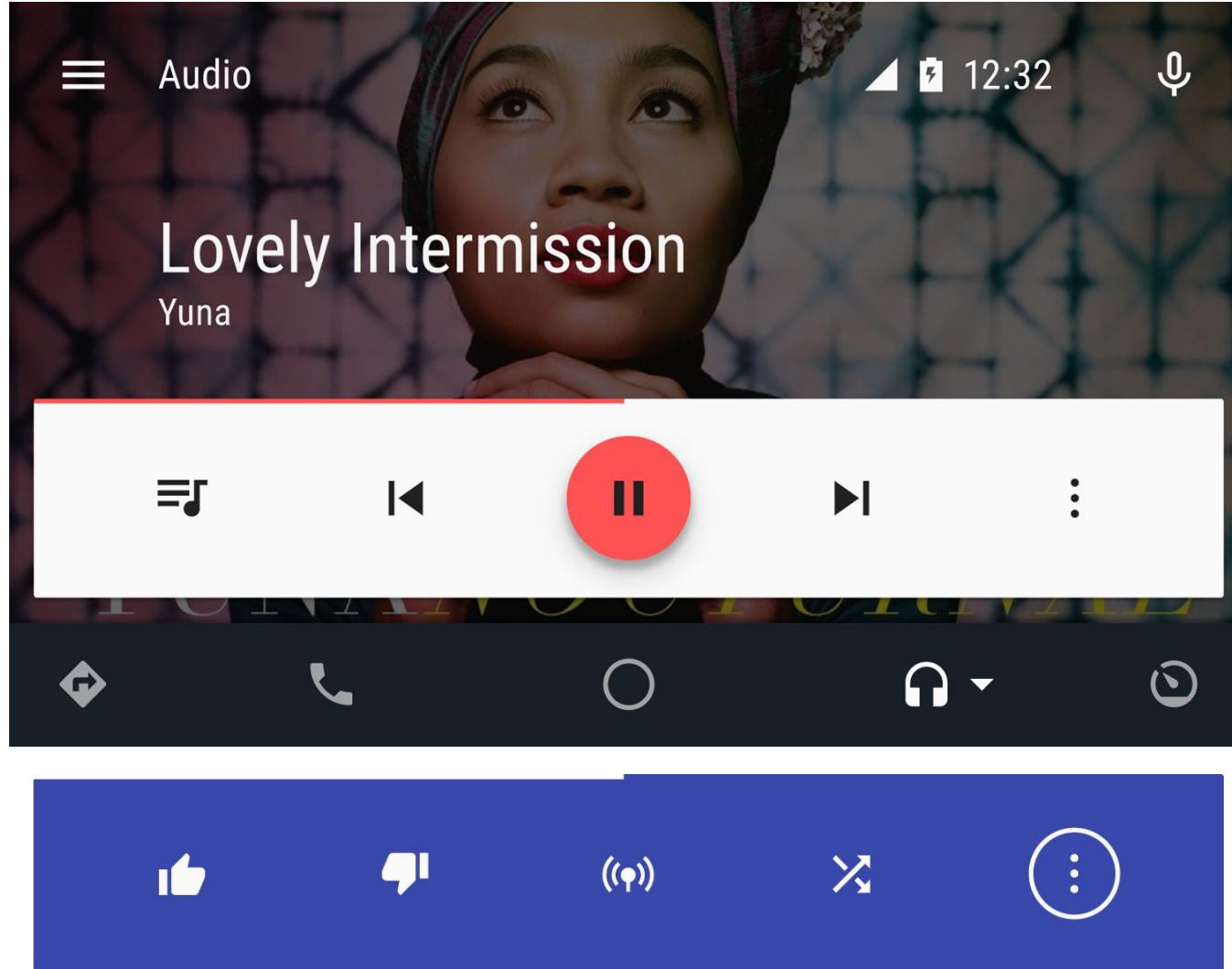


Android Auto - Apps

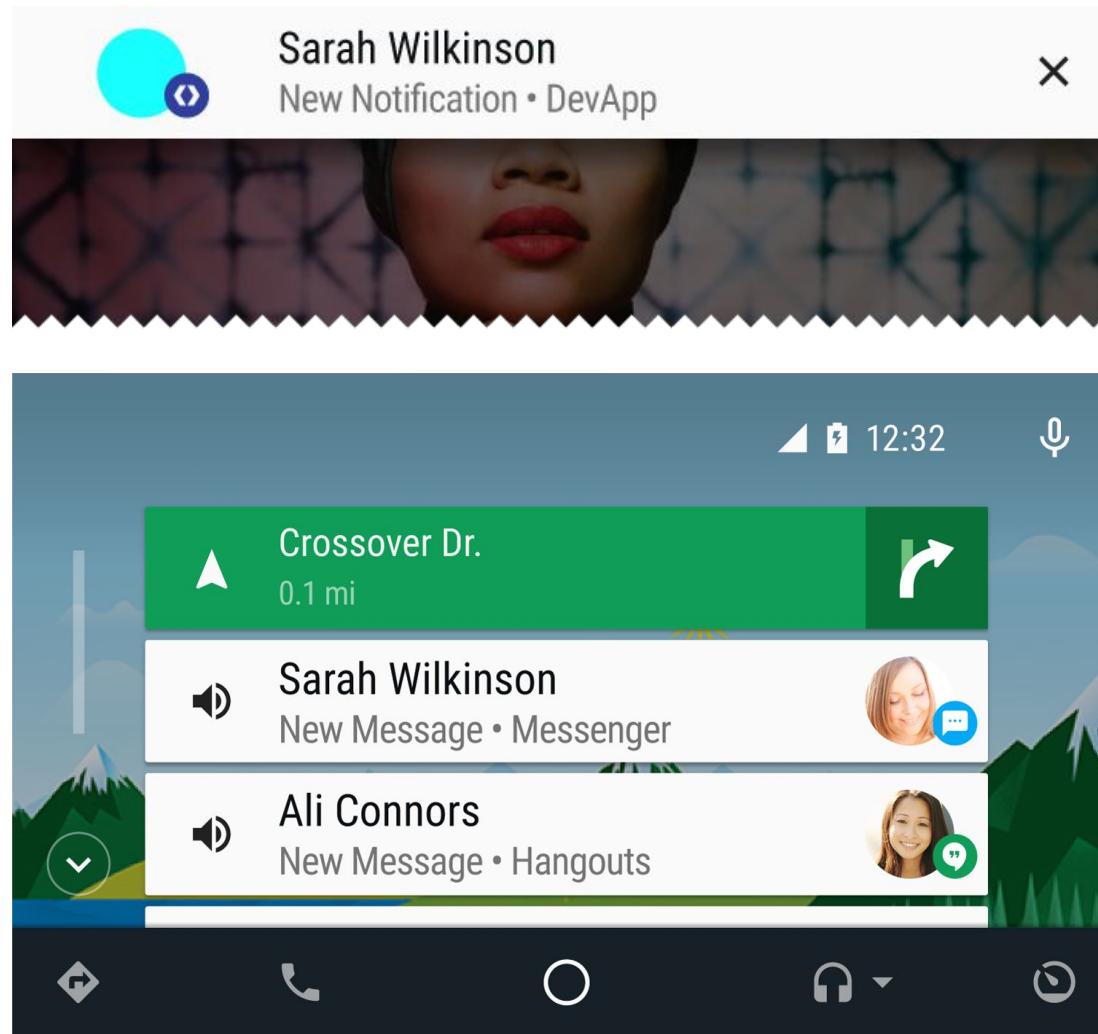
Auto currently supports two types of apps:

- **Audio apps** that allow users to browse and play music and spoken audio content in the car.
- **Messaging apps** that receive incoming notifications, read messages aloud via text-to-speech, and send replies via voice input in the car.

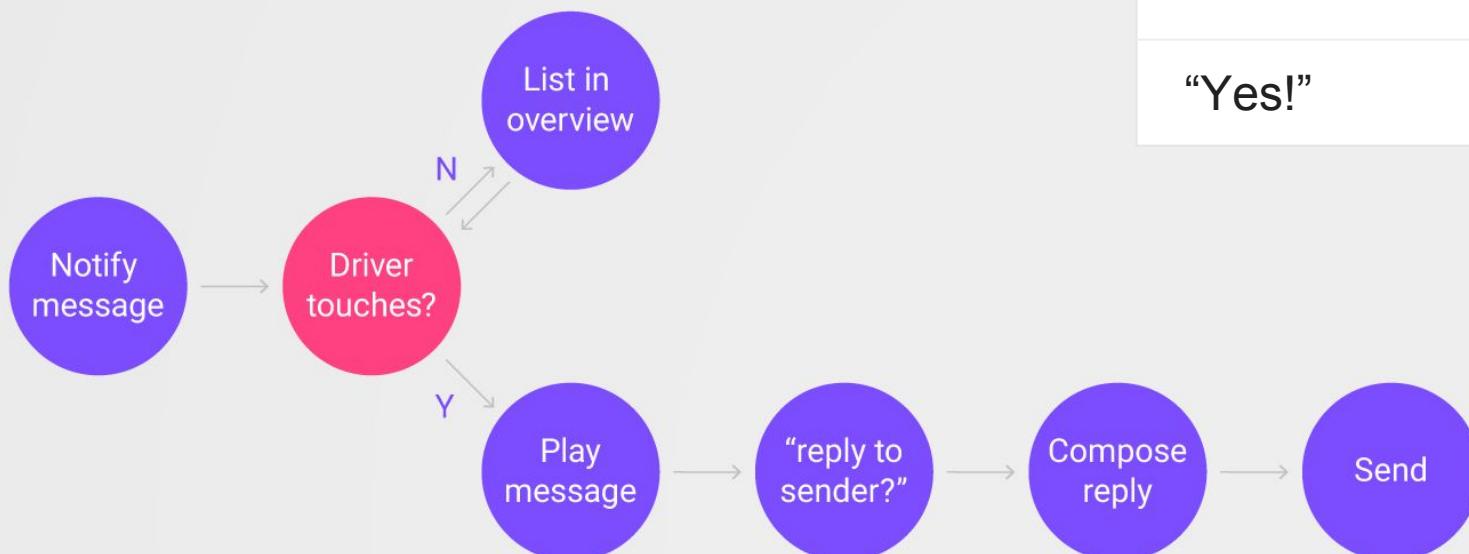
Audio apps



Messaging apps



Messaging apps



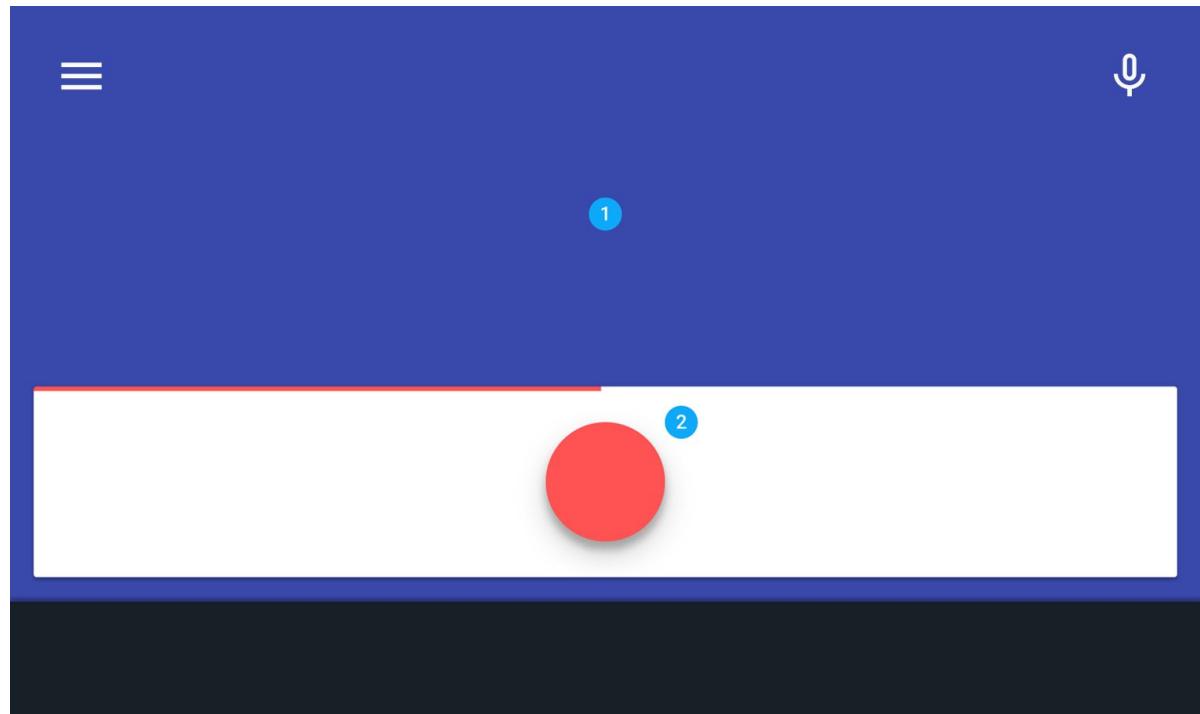
Driver	Google Voice
“Reply!”	“What’s the message?”
“I’ll be 10 minutes late. Please start without me. I’ll find you when I get there.”	“Here is your message to Firstname Lastname: ‘I’ll be 10 minutes late. Please start without me. I’ll find you when I get there.’
“Yes!”	Do you want to send it? “Message sent to xxx”

Colors & branding

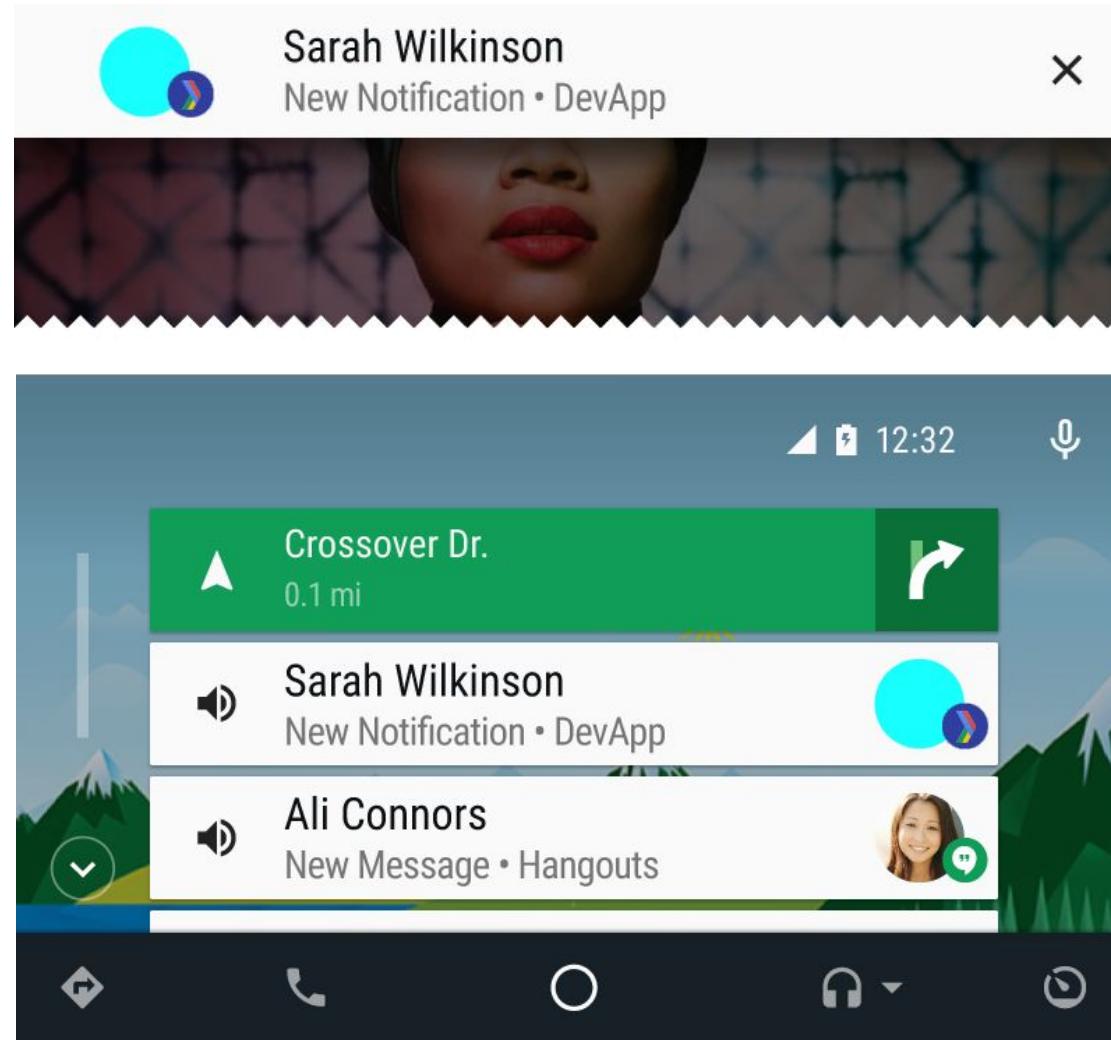
Android Auto apps may specify two colors for the system palette:

- colorPrimaryDark
- colorAccent

The app has the option to specify an additional set of nighttime colors



Colors & branding



Android Auto App Quality

- **Visual Design and User Interaction**
 - App does not display on the Auto screen animated elements.
 - App does not include games or other features outside of the intended app types supported by Auto.
 - App does not display automatically scrolling text.
 - App provides colors that the system can optimize for easy in-vehicle readability.
 - App must support voice commands in audio apps.
- **Functionality**
 - App loads content in no more than 10 seconds.
 - No tasks in the app take more than six steps to complete.
 - App does not present advertisements through notifications.



ACURA



Audi



BENTLEY



BORGWARD



BUICK



Cadillac



CHEVROLET



DODGE



Ford



GMC



HOLDEN



HONDA



HYUNDAI



Jeep

KENWOOD



Koenigsegg



LADA



THE LINCOLN
MOTOR COMPANY

Mahindra



mazda

Mercedes-Benz



mitsubishi
motors



NISSAN



PEUGEOT

Pioneer

RAM

RENAULT



SEAT

ŠKODA



SONY



SUBARU



SUZUKI

TATA
MOTORS



VAUXHALL





J JOYRIDE



tunein

talkray



STITCHER
RADIO ON DEMAND



textPlus



Threema.

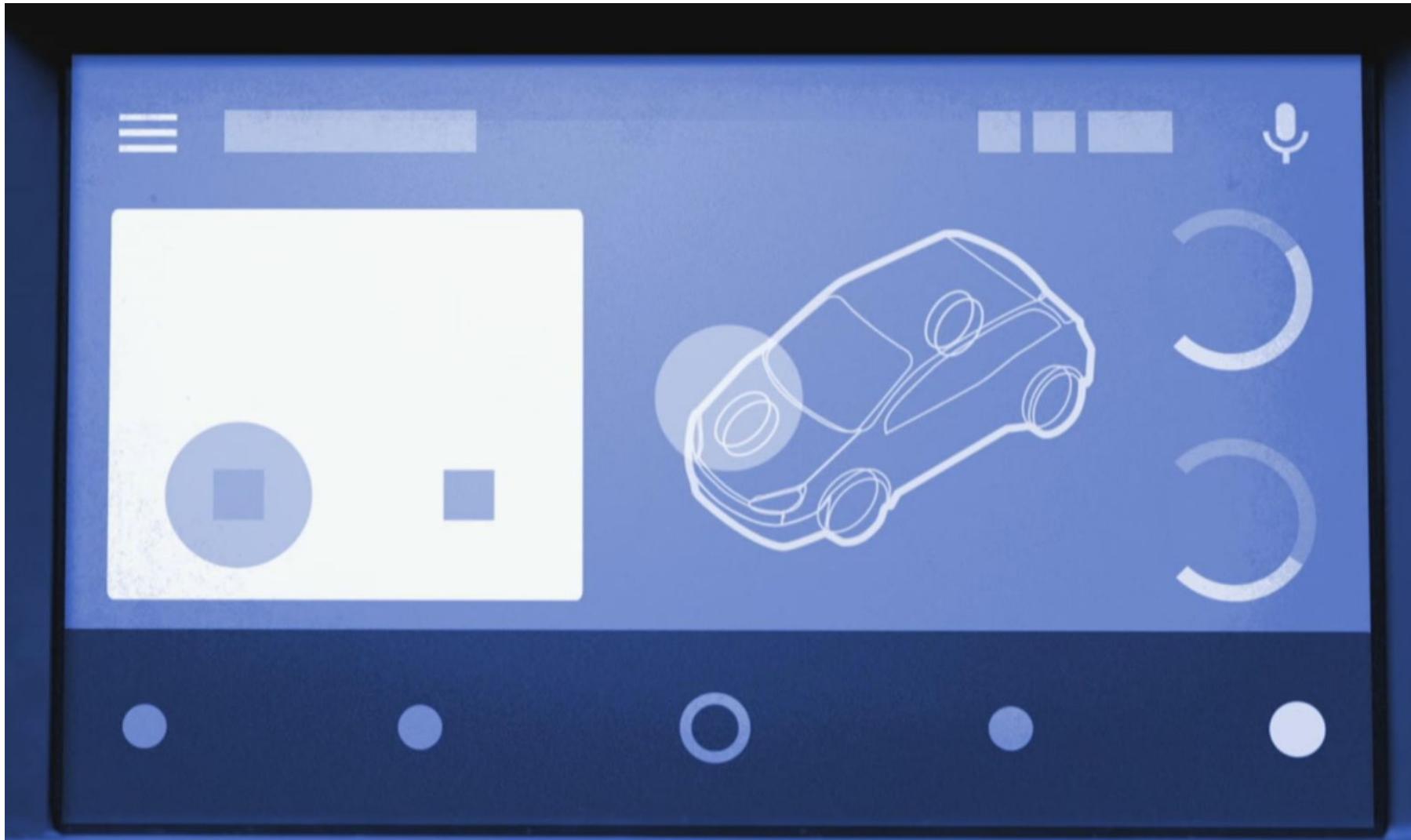
skype™



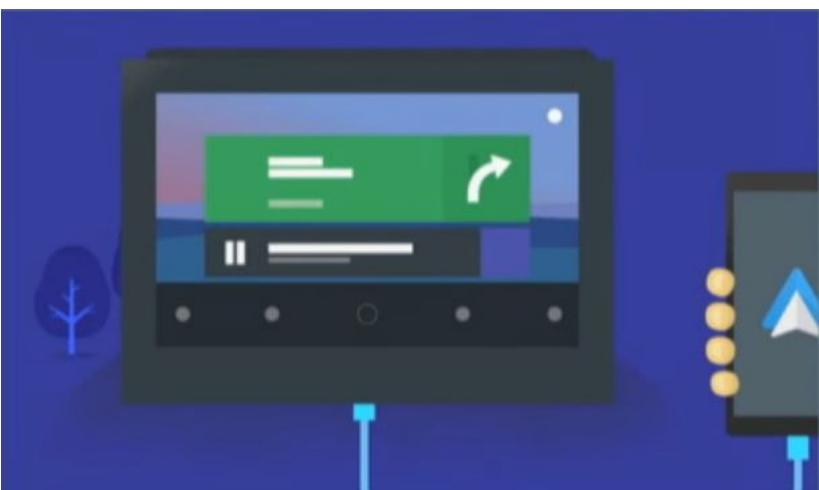
contacts+



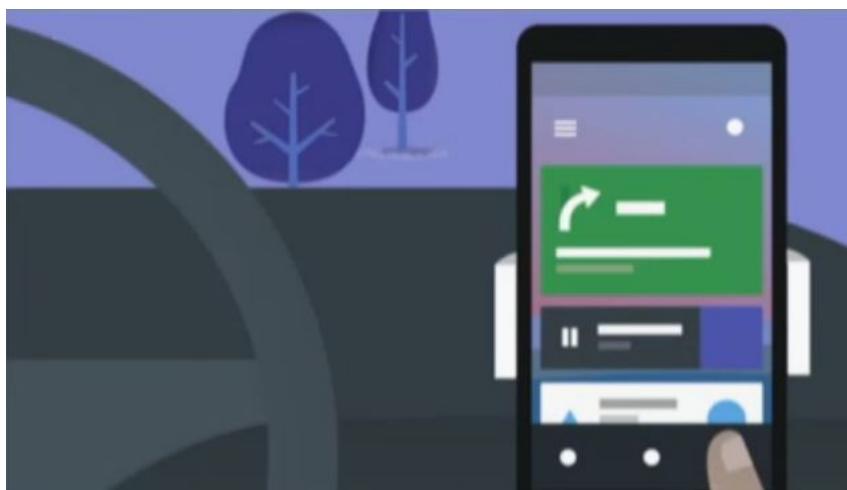
Android Auto - Future



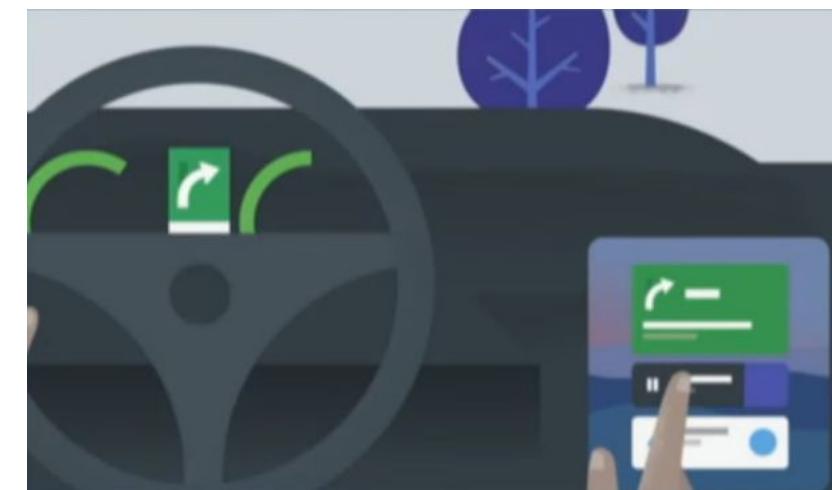
Android Auto - Future



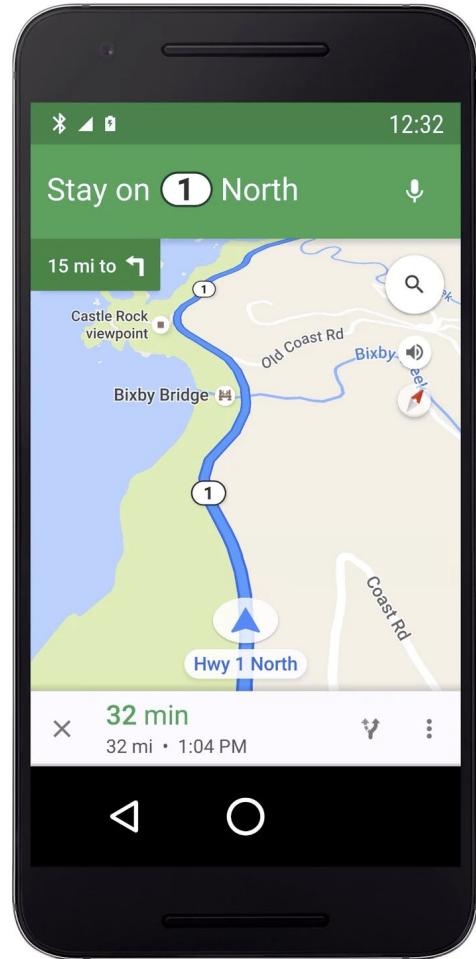
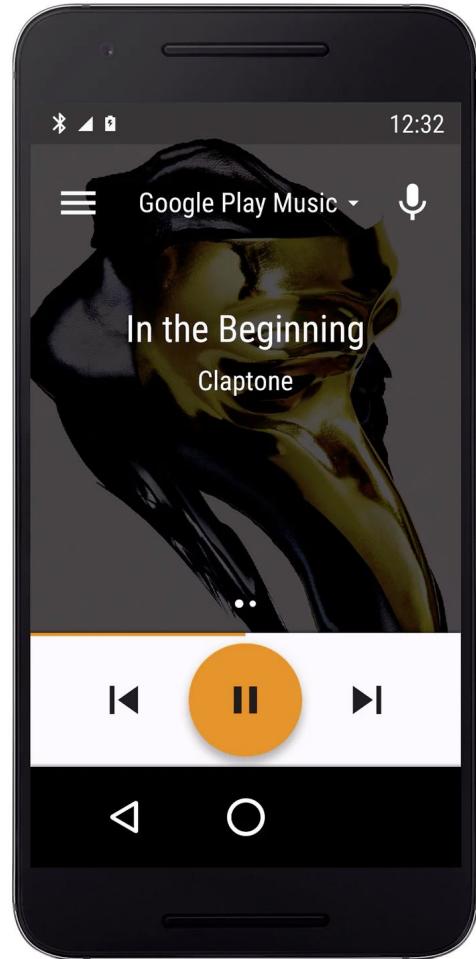
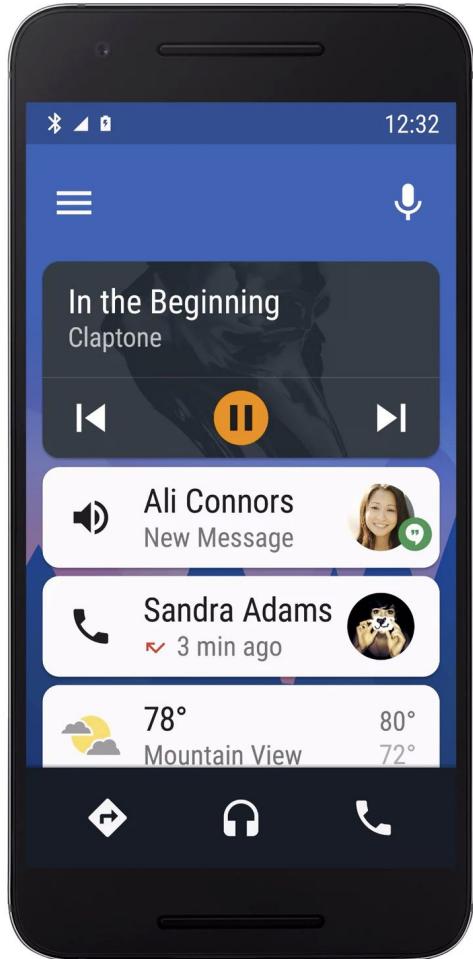
Connected to the phone

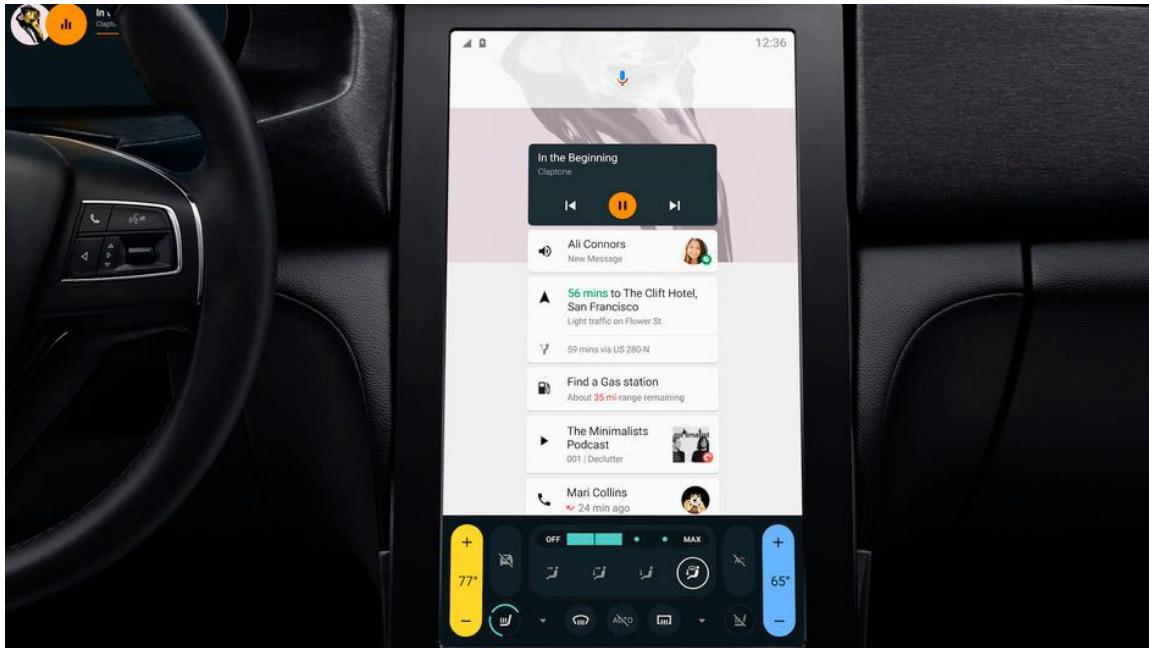


On a phone screen



Built-in to the car







ANDROID-POWERED
MASERATI

M

Android Auto Developers Google+ Community



Oren Mizrahi

where is the phone dashboard for non compatible cars ? is it out yet?

Wayne Piekarski (29.09.2016)

The standalone mode for Android Auto is **coming soon**, but sorry I don't have a release date yet. Very soon! :)

Oren Mizrahi

gimme beta please. promise not to share and give feedback

Android Auto Developers Google+ Community



Hung Chinlin

Is there any android auto API available to read vehicle data?

Wayne Piekarski

Since your app runs on your phone, you can still **access all the sensors in the phone** (accelerometer, GPS, etc). However, if you want access to the engine data, much of this is available via OBD-II protocol via a Bluetooth adaptor that you can get quite cheaply, and there are even apps that show this data on your phone screen.

Android Auto Developers Google+ Community



Wayne Piekarski

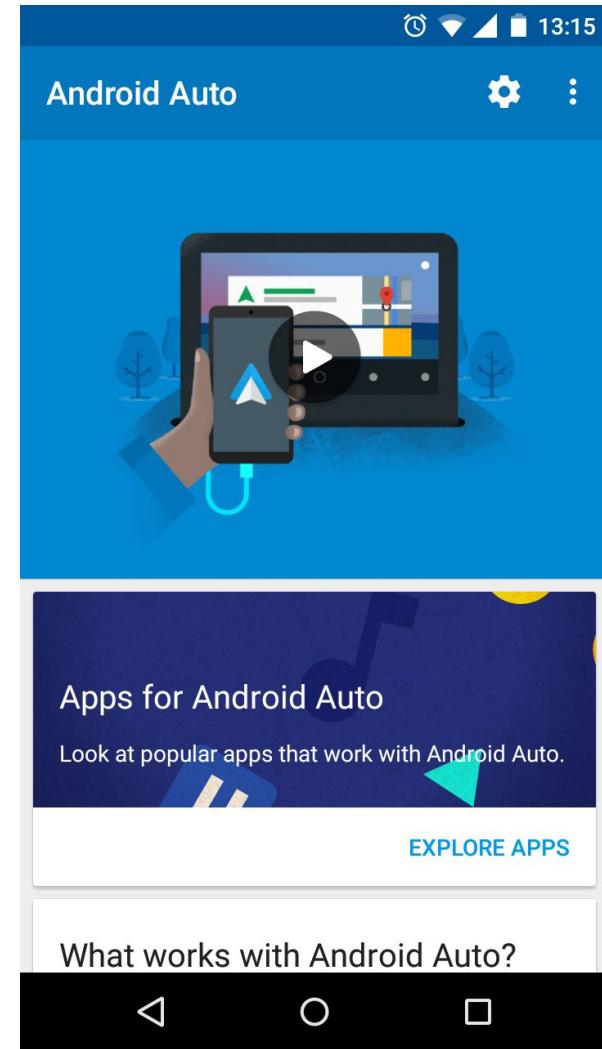
The Android Auto app on the phone checks somehow to ensure the app has been approved. I don't know how this works exactly, and you cannot work around it ... it is designed for exactly this purpose, to require a safety review.

Remember, **Android Auto runs on the phone itself. So the phone draws the pixels for everything on the display, and then sends the pixels to the head unit in the vehicle.** The DHU is basically the same thing as a vehicle display, it renders the pixels from Android Auto to your computer monitor instead. So it is not really an emulator ... it handles exactly the same protocol as a car does. So if the DHU handles it correctly, it should run in any real Android Auto vehicle exactly the same way.



Android Auto app

<https://play.google.com/store/apps/details?id=com.google.android.projection.gearhead>



Auto App Quality

When designing support for Android Auto in your app, **avoid driver distraction above all else**. Apps that work with the Auto user interface should **minimize distractions** faced by the driver through best practices such as voice commands and very simple visual design.

Great auto experiences are **predictive** and **predictable**. Apps that support Android Auto should show timely information to the driver only when it is relevant, and use simple, predictable patterns for common tasks.

<https://developer.android.com/distribute/essentials/quality/auto.html>

Auto App Quality - Visual Design and User Interaction

Type	Description
Driver Attention	<p>App does not display on the Auto screen animated elements such as animated graphics, video, or progress bars.</p> <p>App does not display any form of visual or text advertising on the Auto screen. Only audio ads are acceptable.</p>
	<p>App elements do not display any images on the Auto screen. Exceptions include: app may display a single static image for content context in the background of the consumption screen, such as album art, and app may display icons in the content navigation drawer.</p>
	<p>App does not include games or other features outside of the intended app types supported by Auto.</p>

Auto App Quality - Visual Design and User Interaction

Type	Description
Driver Attention	App never activates the phone screen to present any form of visual information (notifications, toasts, video, images, advertising, etc.) on the phone screen while the app is interacting with the Android Auto dashboard unit.
Layout	App does not display automatically scrolling text.
Visual Contrast	App supports day mode, which renders dark text and controls on a light background.
	App supports night mode, which renders light text and controls on a dark background.

Auto App Quality - Visual Design and User Interaction

Type	Description
Visual Contrast	App provides white icon sets that the system colorizes to provide automatic contrast compensation.
	App provides colors that the system can optimize for easy in-vehicle readability.
Interaction	App must support voice commands in audio apps.
	App-specific buttons respond to user actions with no more than a two-second delay.

Auto App Quality - Functionality

Type	Description
General	<p>App launches in no more than 10 seconds.</p> <p>App loads content in no more than 10 seconds.</p> <p>Auto app functionality works as expected or described in the app's Google Play Store listing.</p> <p>When the app is relaunched from the home screen, the app restores the app state as closely as possible to the previous state.</p> <p>No tasks in the app take more than six steps to complete.</p> <p>Interactive elements that are intentionally grayed-out must be non-functional.</p>

Auto App Quality - Functionality

Type	Description
Media	Media apps do not autoplay on startup of Android Auto or without user initiated action to select the app or app media.
Notifications	<p>App does not present advertisements through notifications.</p> <p>App displays notifications only when relevant to the driver's needs.</p> <p>Examples:</p> <p>Good: Notifying the user that a new message has arrived.</p> <p>Bad: Notifying the user about a new album release.</p>

Auto App Quality - Functionality

Type	Description
Messaging	<p>App is able to successfully receive incoming messages.</p>
	<p>Messages are properly grouped and displayed in the correct order.</p>
	<p>User is able to successfully reply to a message.</p>
	<p>Apps use short-form messaging app design patterns. Traditional long-form messaging apps, such as apps for email, are not permitted.</p>
	<p>Apps are peer-to-peer messaging services and not notification service apps such as weather, stocks, and sport scores apps.</p>

Getting Started with Auto

Android Auto **provide a simplified interface for apps that can be used in a car.**

Apps that work with Android Auto consoles **run on a connected device**, such as a phone or tablet. **The app communicates via specific APIs with the in-dash console**, which provides a user interface for the connected app that is designed for use in a car.

Getting Started with Auto

Android Auto extends the Android platform into the car. When users connect their handheld **devices running Android 5.0 or higher** to a compatible vehicle, the Auto user interface provides a **car-optimized Android** experience on the vehicle's screen. Users interact with compatible apps and services through **voice actions** and the **vehicle's input controls** (like a touchscreen or dashboard buttons).

Getting Started with Auto

Auto currently supports **two** types of apps:

- **Audio apps** that allow users to browse and play music and spoken audio content in the car.
(<https://developer.android.com/training/auto/audio/index.html>)
- **Messaging apps** that receive incoming notifications, read messages aloud via text-to-speech, and send replies via voice input in the car.
(<https://developer.android.com/training/auto/messaging/index.html#messaging>)

Getting Started with Auto

Create automotive_app_desc config file

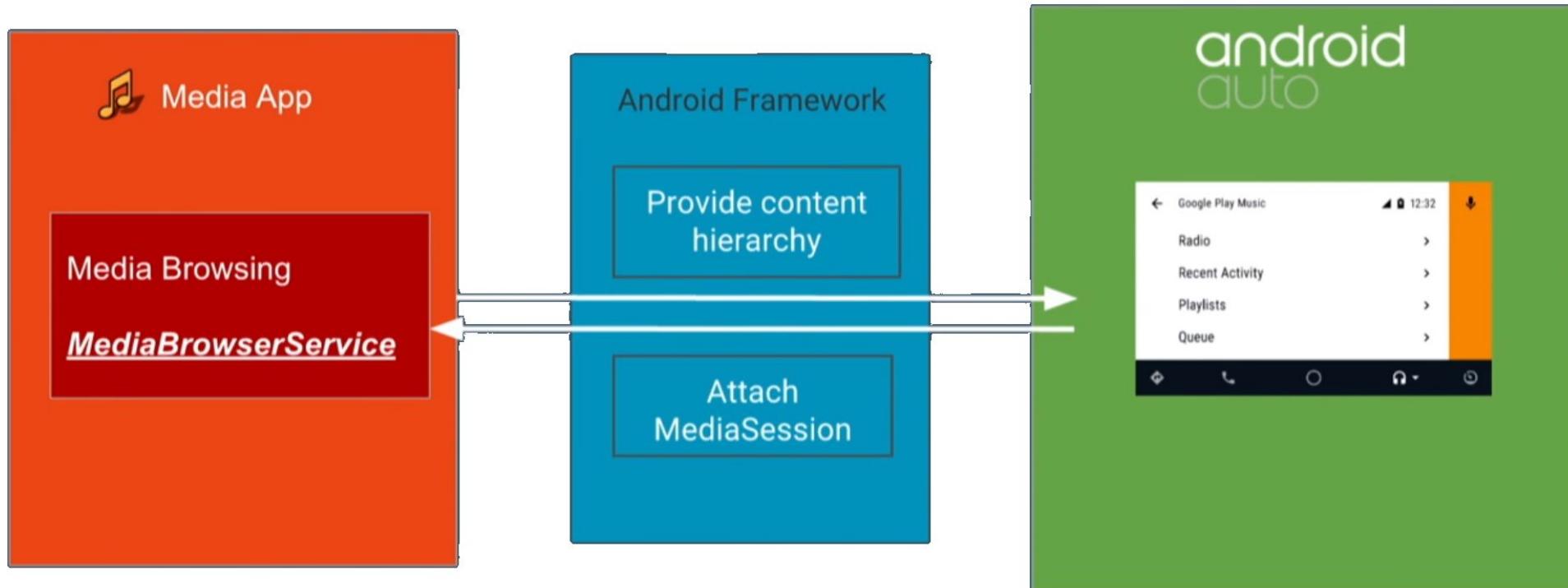
```
<automotiveApp>
    <uses name="media"/>
</automotiveApp>
```

Add to <application> in AndroidManifest

```
<meta-data
    android:name="com.google.android.gms.car.application"
    android:resource="@xml/automotive_app_desc"/>
```

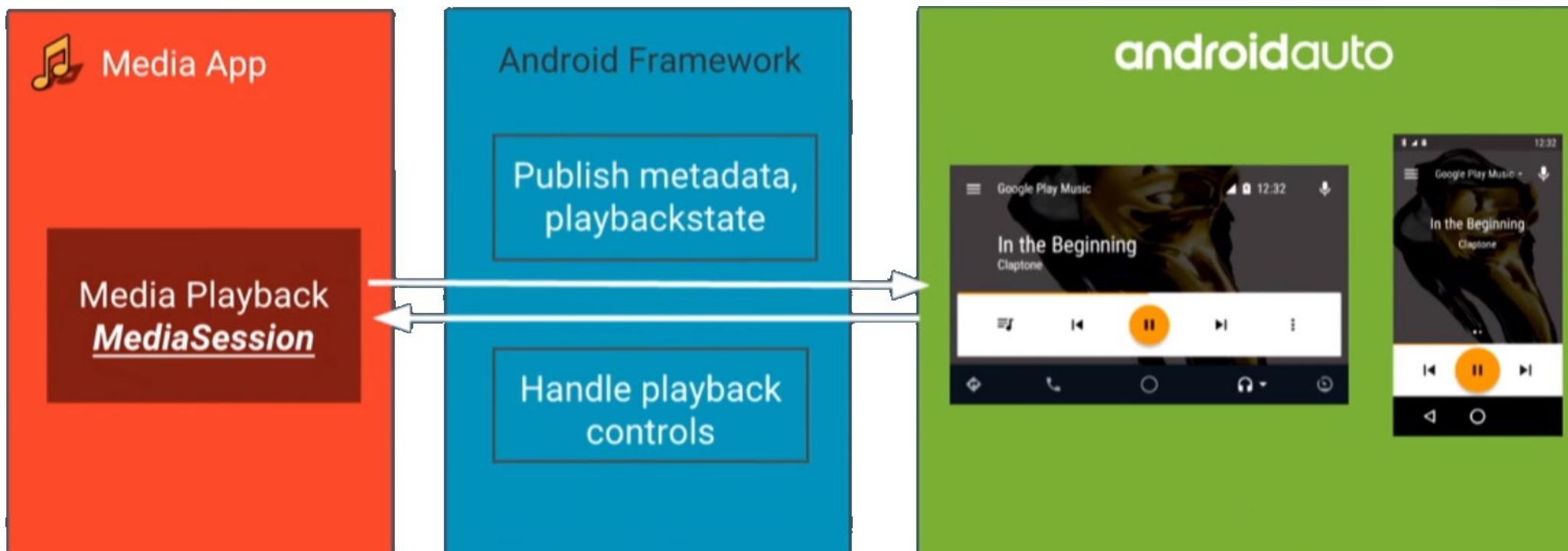
Audio Playback for Auto

- Extend MediaBrowserService to provide content hierarchy
 - `onGetRoot()`, `onLoadChildren()`



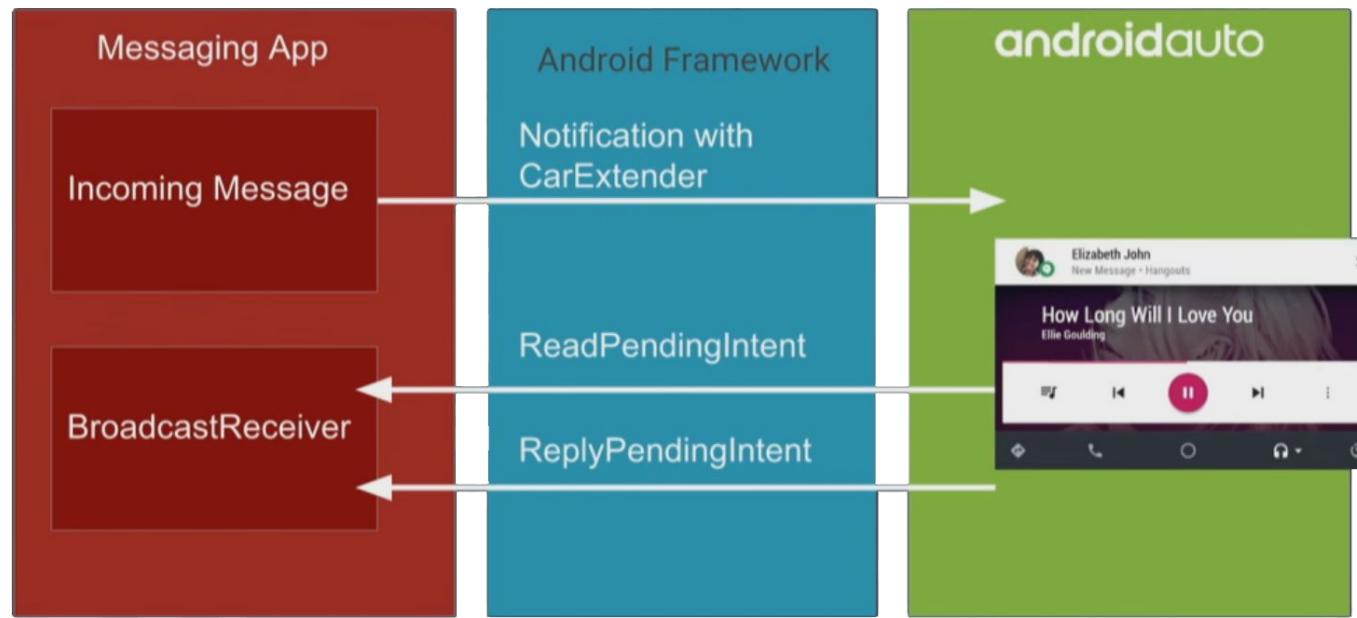
Audio Playback for Auto

- Implement playback controls through callback methods
 - `onPlay()`, `onPause()`, `onStop()`, `onSkipToNext()`, etc...
- Register for voice action with IntentFilter in AndroidManifest



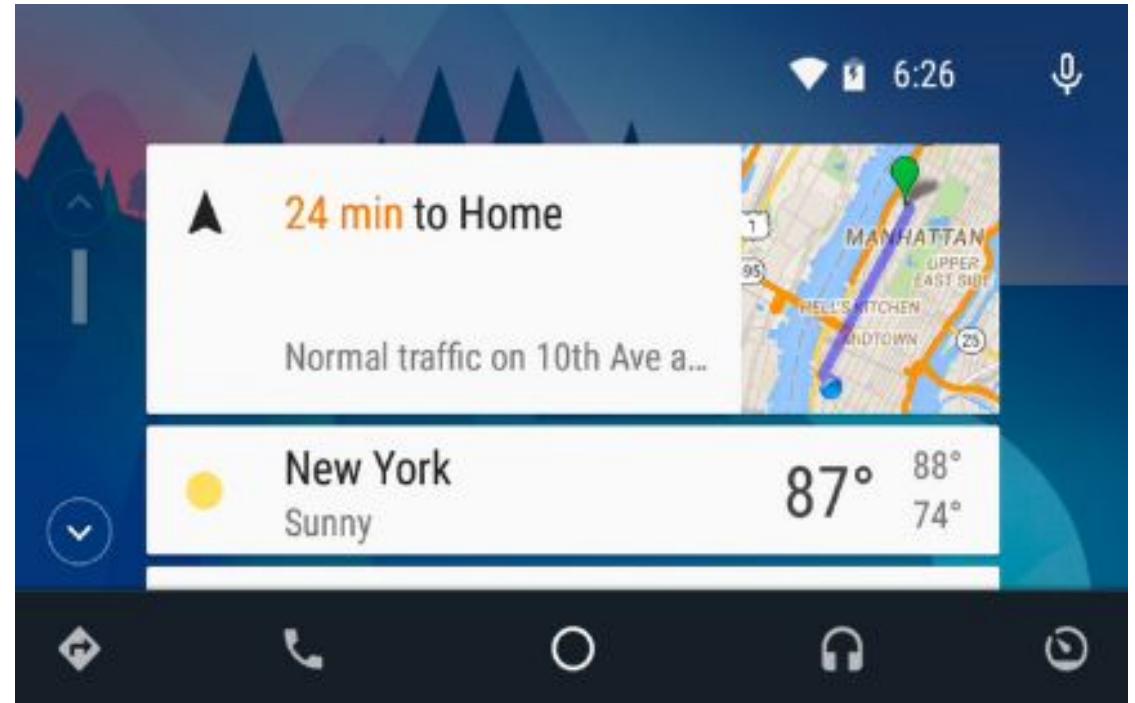
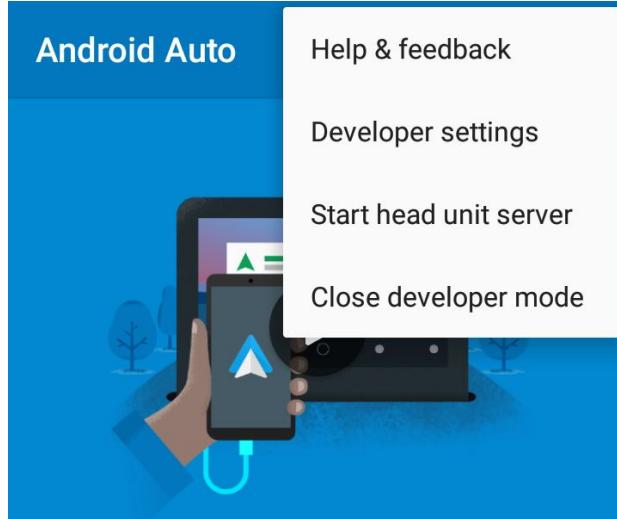
Messaging for Auto

- Extend existing notification objects with CarExtender and set UnreadConversation
- System will send intents corresponding to user actions
- Update app in BroadcastReceiver based on user action
 - Resolve and send response through your app
 - Also: mark as read, dismiss notification, etc



Testing Apps for Auto

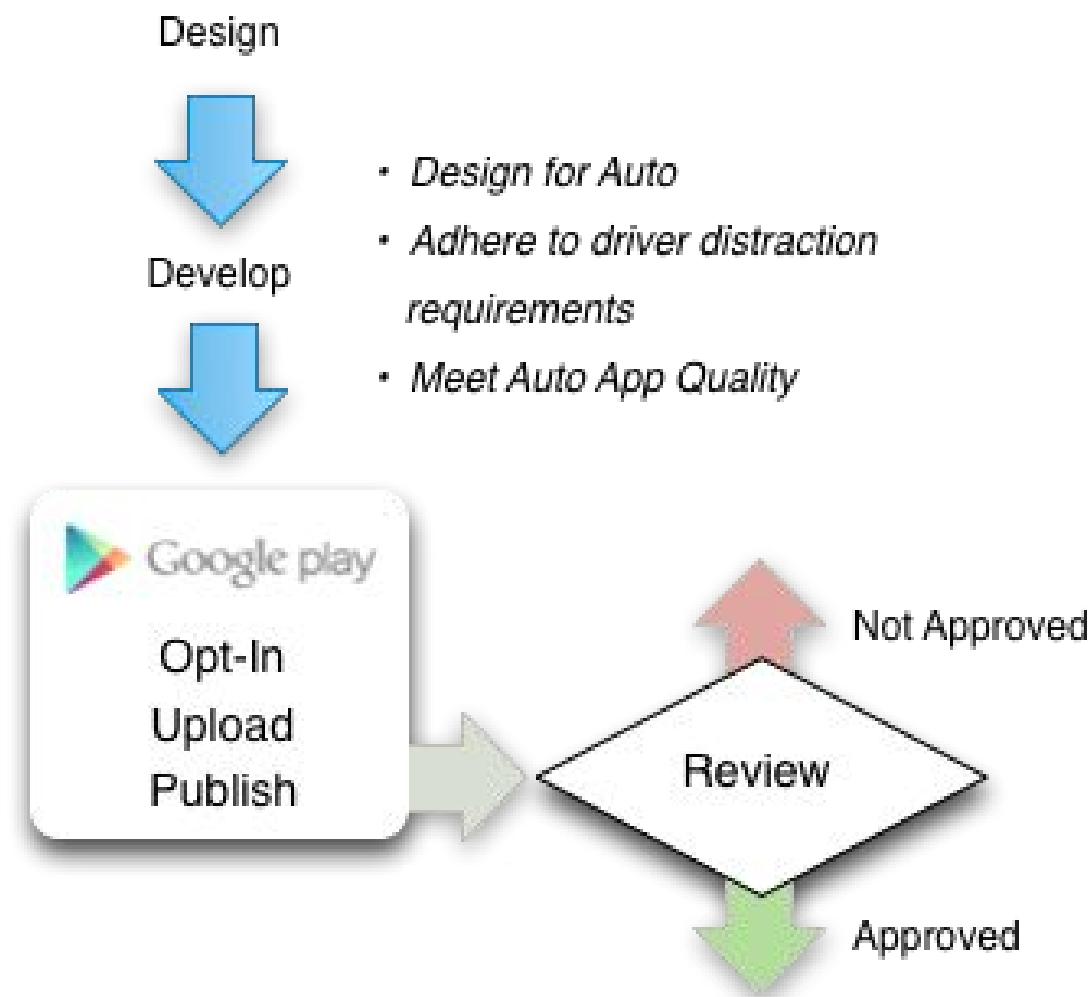
Desktop Head Unit (DHU), a **testing tool** for Auto apps that lets you test pre-released versions of your Android Auto apps without having to work from your car.



<https://developer.android.com/training/auto/testing/index.html>

Distribute to Android Auto

You must deliver your Auto experience as part of your existing app for phones, tablets, and other devices, using the same package name and store listing.



Android Auto Addendum to the Developer Distribution Agreement (March 11, 2015)

1. **Documentation.** Developer will **only access or use Android Auto as described in the documentation located at developers.android.com/auto (as may be amended from time to time)**
2. **UX Design.** Developer will adhere to the Design Guidelines specified in the documentation, which includes **not modifying the User Interface or submitting Products that could be considered a high-risk activity for a user to engage in while driving.**
3. **Intellectual Property Rights and Brand Features.** As required in the Agreement, **You are responsible for your Products, including that the Products (and any content within or brand features) are licensed for use in an automobile.**
4. **Approval and Public Availability.** If You publish your Products with a car-enabled APK, your Products will be subject to prior review and approval prior to external publication whether to Android Auto or the Store. Google will notify You of the results of your submission and any such approval is in Google's sole discretion. **Notwithstanding the foregoing, Google reserves the right to block any Products being used in a car running Android Auto at any time for any reason.**

Next step

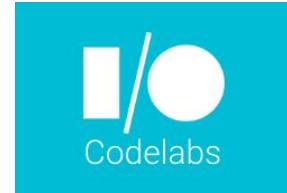


`googlesamples/android-UniversalMusicPlayer`

This sample shows how to implement an audio media app that works across multiple form factors and provide a consistent user experience on Android phones, tablets, **Auto**, Wear and Cast devices

<https://github.com/googlesamples/android-UniversalMusicPlayer>

Next step



Playing music on cars and wearables

In this codelab, you will learn how to implement an Android media app that goes beyond the basic form factors of phone and tablet. Learn how to interactively add the required pieces to support Android Auto and Android Wear.

TIMELINE: 49 min

Messaging apps on cars

In this codelab, you will learn how to extend a messaging app with Android Auto compatibility.

TIMELINE: 29 min

<https://io2015codelabs.appspot.com/>

Next step UDACITY

Android Auto Development by 

Reach your users while they're out and about by extending your Media or Messaging app to Android Auto! As you'll learn in this course, making your app available in the car is as easy as extending your existing code. This course explains how to get started and then covers Media and Messaging apps in detail.

COURSE COST: Free

TIMELINE: Approx. 1 weeks

<https://www.udacity.com/course/android-auto-development--ud875C>

Linki

- Android Auto: <https://www.android.com/auto/>
- Designing for Android Auto:
<https://www.google.com/design/spec-auto/designing-for-android-auto/>
- Android Auto Developers: <https://developer.android.com/auto/index.html>
- Android Auto Developers G+ Community:
<https://plus.google.com/communities/116320632775523824083>
- Distribute to Android Auto:
<https://developer.android.com/distribute/googleplay/auto.html>
- Auto App Quality:
<https://developer.android.com/distribute/essentials/quality/auto.html>

Dziękuję za uwagę!

Krzysztof Sowa

krzysiek.sowa90@gmail.com