

Android Media Hacks

@KeithYokoma - Drivemode, Inc.

potatotips #18

@KeithYokoma    

Keishin Yokomaku at Drivemode, Inc. as Android Engineer

Experience

- 1.SNS client and photo book application for Android
- 2.Driver's application for Android

Publications

- 1.Android Training
- 2.Mixi official smartphone application development guide

Like

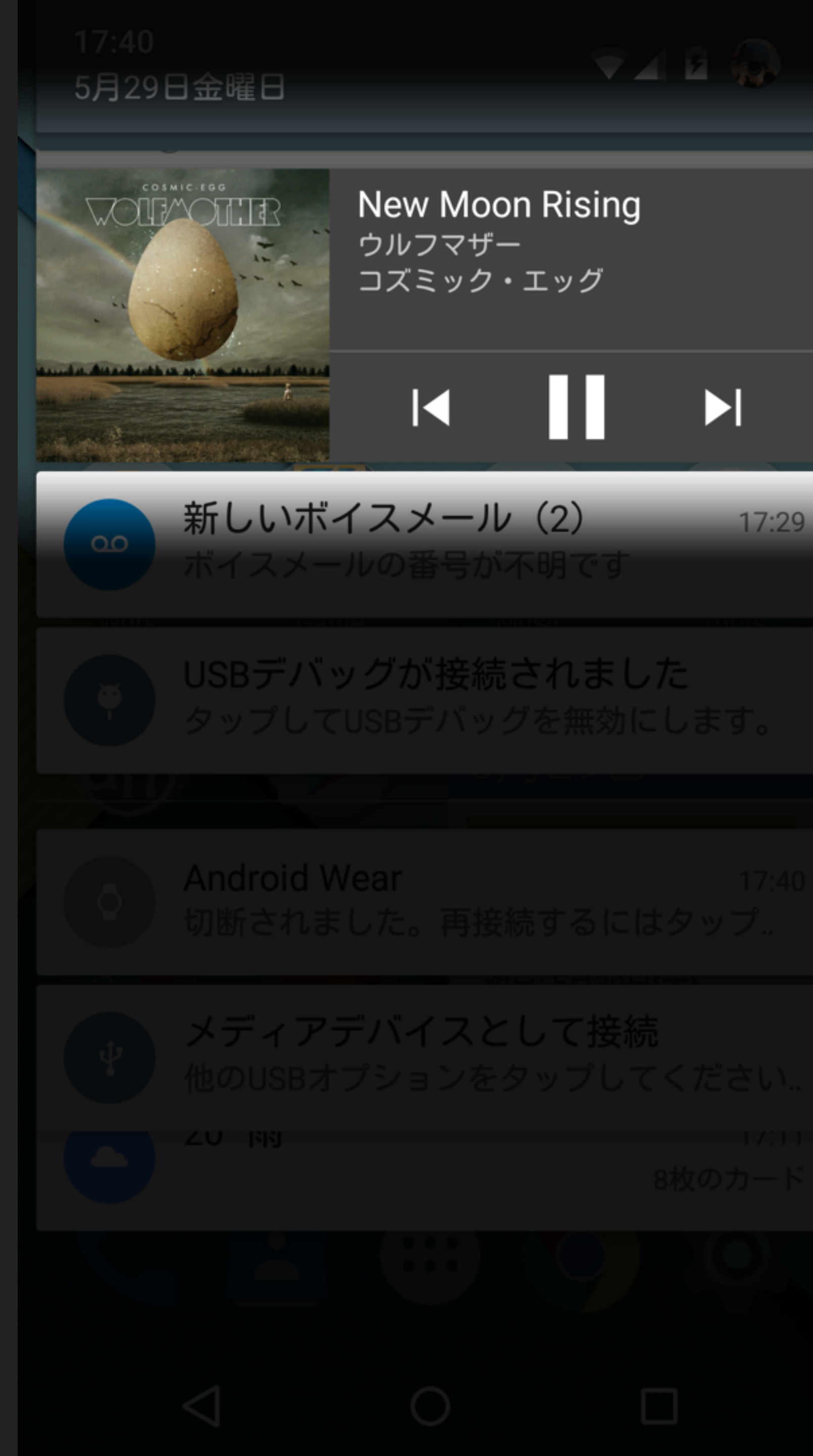
Motorsport, Bicycle, Photography, Tumblr



Media Playback Control

Put media playback controller on...

- Notification
- AppWidgets
- Android Auto App
- Lock Screen(KeyGuard)



Media Playback Control

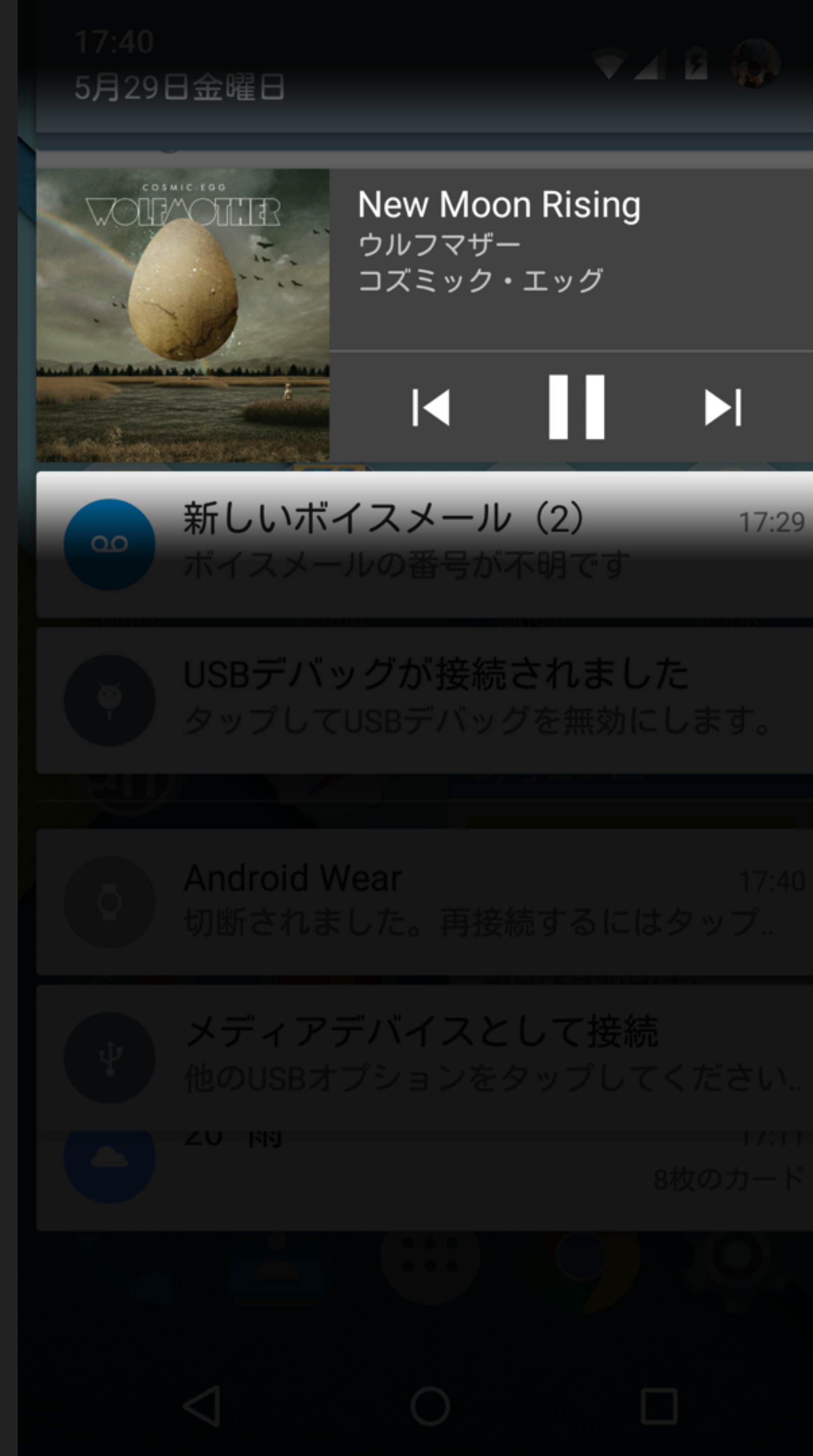
Different implementations for each API level

- RemoteController (from KitKat)
- RemoteControlClient (from ICS)

These system are based on PendingIntent

From Lollipop...

- MediaSession and MediaController



MediaController

Standardized media playback control API from Lollipop

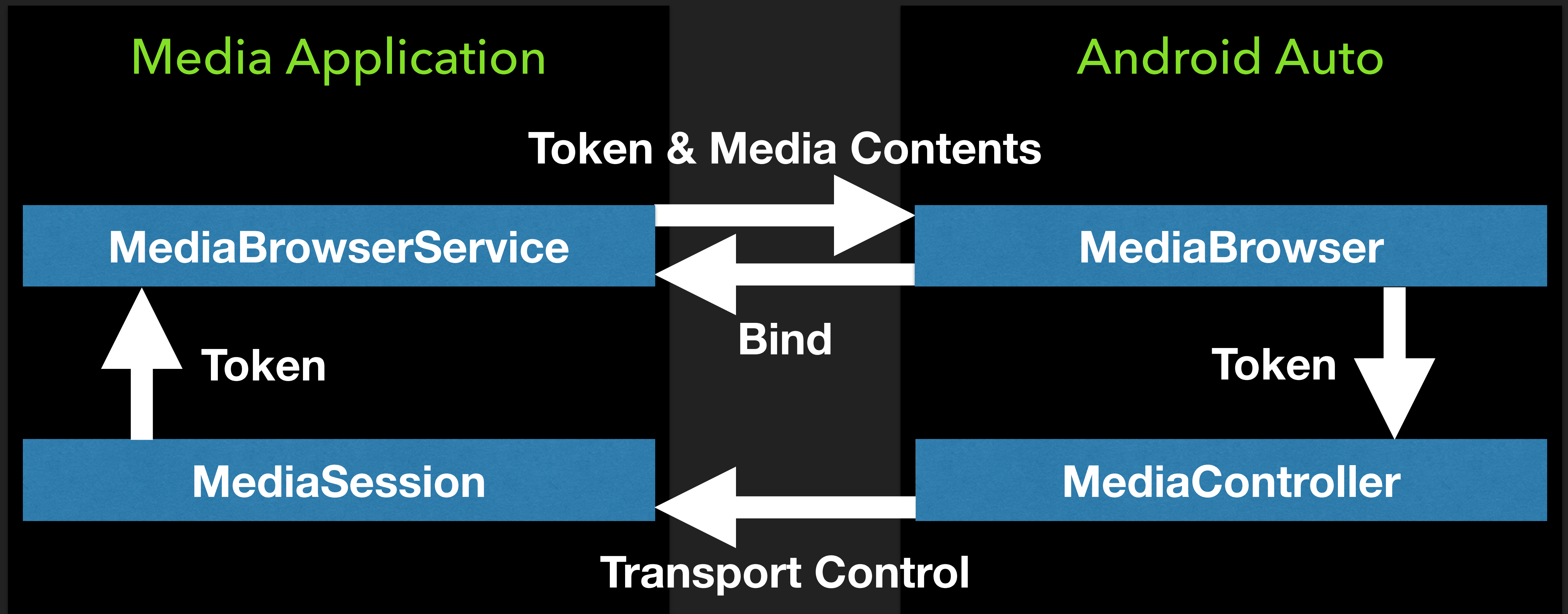
System will use this API in...

- Notification.MediaStyle
- Android Auto App

How it works on Notification



How it works on Android Auto



Watching Notification...

NotificationListenerService

NotificationListenerService

Event receiver for status bar notification available from JB-MR2

We can access following data in the notification...

- Title
- Message
- ContentView
- HeadsUpContentView
- Extras

NotificationListenerService

Event receiver for status bar notification available from JB-MR2

We can access following data in the notification...

- Title
- Message
- ContentView
- HeadsUpContentView
- Extras

Notification.MediaStyle

String

EXTRA_MEDIA_SESSION

`extras` key: A `MediaSession.Token` associated with a `Notification.MediaStyle` notification.

Notification.MediaStyle

String

EXTRA_MEDIA_SESSION

`extras` key: A MediaSession.Token associated with a `Notification.MediaStyle` notification.

MediaController

Public Constructors

MediaController ([Context](#) context, [MediaSession.Token](#) token)

Create a new MediaController from a session's token.

MediaController

Public Constructors

MediaController ([Context](#) context, [MediaSession.Token](#) token)

Create a new MediaController from a session's token.

*"Yes, you can steal a media session token
without invading private APIs"*

-Me

“And you can set the token to your own `MediaController`”

-Me

“This is more like session hijacking”

-Someone

Take control of music playback

```
public class MusicNotificationWatcher extends NotificationListenerService {  
    private Bus eventBus;  
    @Override  
    public void onNotificationPosted(StatusBarNotification sbn) {  
        MediaSession.Token token =  
            sbn.getNotification().extras.getParcelable(Notification.EXTRA_MEDIA_SESSION);  
        eventBus.post(new NewMusicNotification(token));  
    }  
}
```

```
public class MyActivity extends Activity {  
    private MediaController controller;  
    @Subscribe  
    public void onNewMusicNotificationAdded(NewMusicNotification event) {  
        controller = new MediaController(this, event.getToken())  
    }  
}
```

Take control of music playback

```
public class MusicNotificationWatcher extends NotificationListenerService {
    private Bus eventBus;
    @Override
    public void onNotificationPosted(StatusBarNotification sbn) {
        MediaSession.Token token =
            sbn.getNotification().extras.getParcelable(Notification.EXTRA_MEDIA_SESSION);
        eventBus.post(new NewMusicNotification(token));
    }
}

public class MyActivity extends Activity {
    private MediaController controller;
    @Subscribe
    public void onNewMusicNotificationAdded(NewMusicNotification event) {
        controller = new MediaController(this, event.getToken())
    }
}
```


Take control of music playback

```
public class MusicNotificationWatcher extends NotificationListenerService {
    private Bus eventBus;
    @Override
    public void onNotificationPosted(StatusBarNotification sbn) {
        MediaSession.Token token =
            sbn.getNotification().extras.getParcelable(Notification.EXTRA_MEDIA_SESSION);
        eventBus.post(new NewMusicNotification(token));
    }
}

public class MyActivity extends Activity {
    private MediaController controller;
    @Subscribe
    public void onNewMusicNotificationAdded(NewMusicNotification event) {
        controller = new MediaController(this, event.getToken())
    }
}
```




It only works for Google Play Music

Conclusion

- Do not put PRIVATE things on your notification
 - Crackers are watching you
- Consider using `MediaSession` and `MediaController`
 - For better interaction with Auto, Notification and your partners' applications

Conclusion

- For a OS compatibility reason, you can use `RemoteControlClient` and `AudioManager`
 - Still you need some dirty hacks...
- For an application compatibility reason, you should use `RemoteController`
 - Even though it is deprecated now, it works on Lollipop

Android Media Hacks

@KeithYokoma - Drivemode, Inc.

potatotips #18