Creating your own

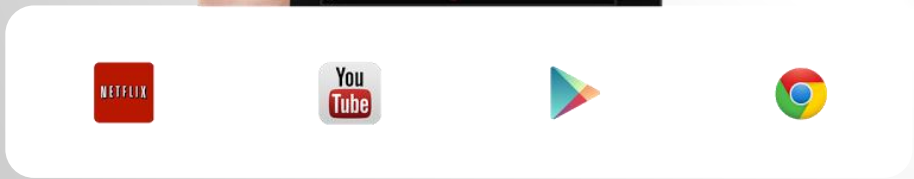
Google Cast App

Paul Lammertsma
CTO, Pixplicity

.Pixplicity®

Google Cast?

Google Cast?



.Pixplicity®

Google Cast?

- **Netflix, YouTube, Google Play Movies, Music & TV**
- **Stream on the TV**
- **Control from Android, iOS or Chrome**

- ***Build your own apps***

You’ll need

- TV with HDMI
- WiFi with Internet connection
- USB power
- A sender device

Senders & receivers

Android or iOS app using Chromecast SDK
Chrome using Chromecast extension

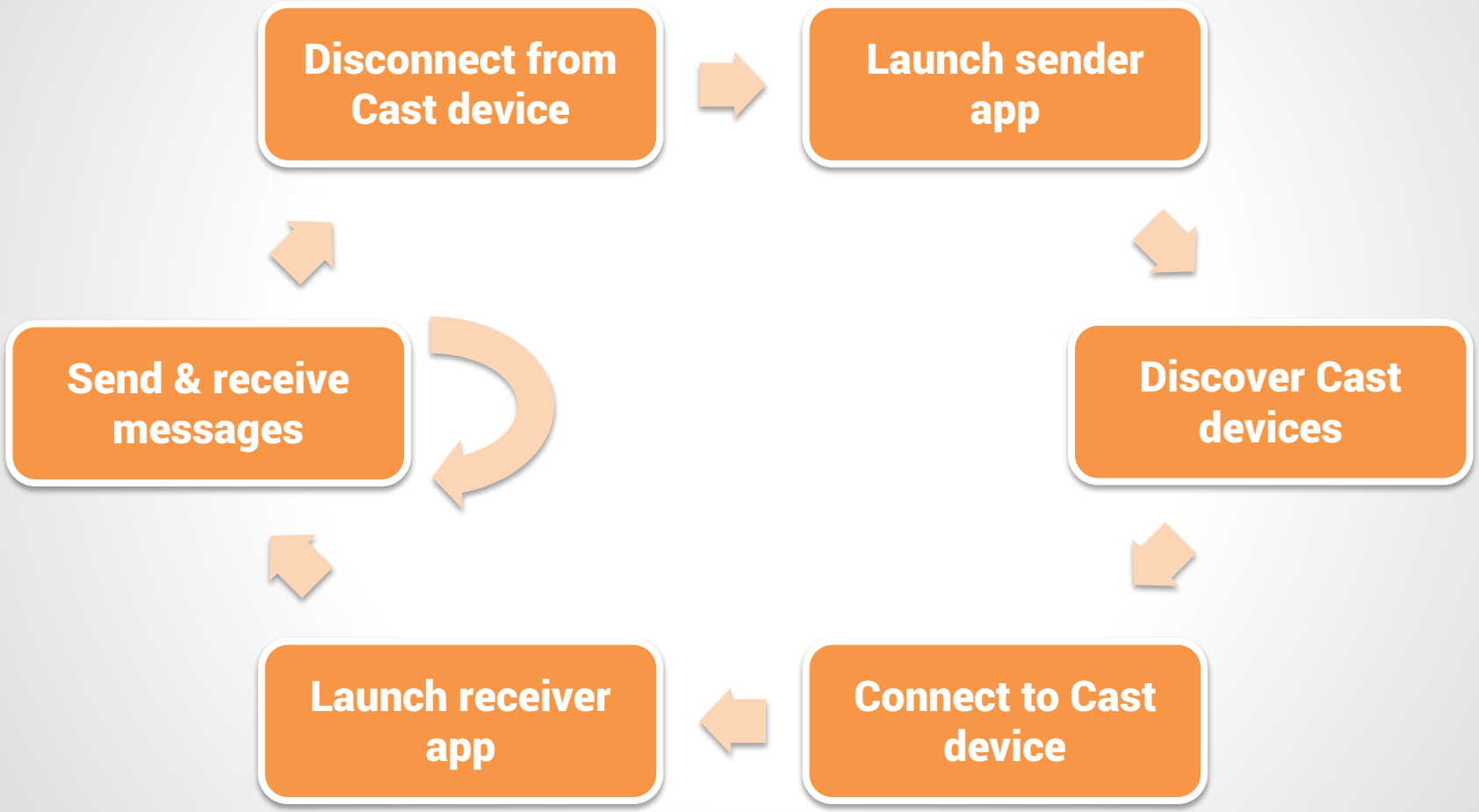
Single-page HTML app

.Pixplicity®

Sending & receiving

- Communication channels
- Multiple sessions per receiver app

Flow



Senders

- Discover Cast devices
- Provide Cast button in UI
- Start or join a session with a Cast device

Receivers

- Single page HTML apps
- Content loaded from the network
  - Internet or local WiFi
- Identified by ID
  - Needs to talk to Google over HTTPS
- Communicates with sender over message bus

Receiver types

- **Default media receiver**
  - Media playback
- **Styled media receiver**
  - Media playback + CSS
- **Custom receiver**
  - Whatever you like
  - Needs media player lib

€5 registration fee per developer

The docs

https://developers.google.com/cast/

- **Design checklist**
  Understand basic flow and avoid common pitfalls
- **Developer guide: sender apps**
  Choose the type of sender app and get started with relevant samples
- **Developer guide: receiver apps**
  Choose the type of receiver app and get stared with relevant samples
- **API Reference**

The samples

https://github.com/googlecast

- **Various samples for all platforms**
- **By Google**

The community

https://plus.google.com/communities/115742157569103585 4

- *Google Cast Developers* community on Google+
- Google engineers at the ready!

Android sender

First a bit of preparation…

Google

Google Cast SDK Developer Console

Overview
Applications
Devices
Settings

Welcome to the Google Cast SDK Developer Console

The Google Cast Developer Console enables developers to register applications and authorize devices for testing.

Applications

Application ID	Application Name	Status	
	Group Gallery	Unpublished	Remove
916A9C15	Cast Demo	Unpublished	Remove

ADD NEW APPLICATION

Devices

Serial Number	Description	Status	
	Paul & Thif	Ready For Testing	Remove
	Dylan	Ready For Testing	Remove

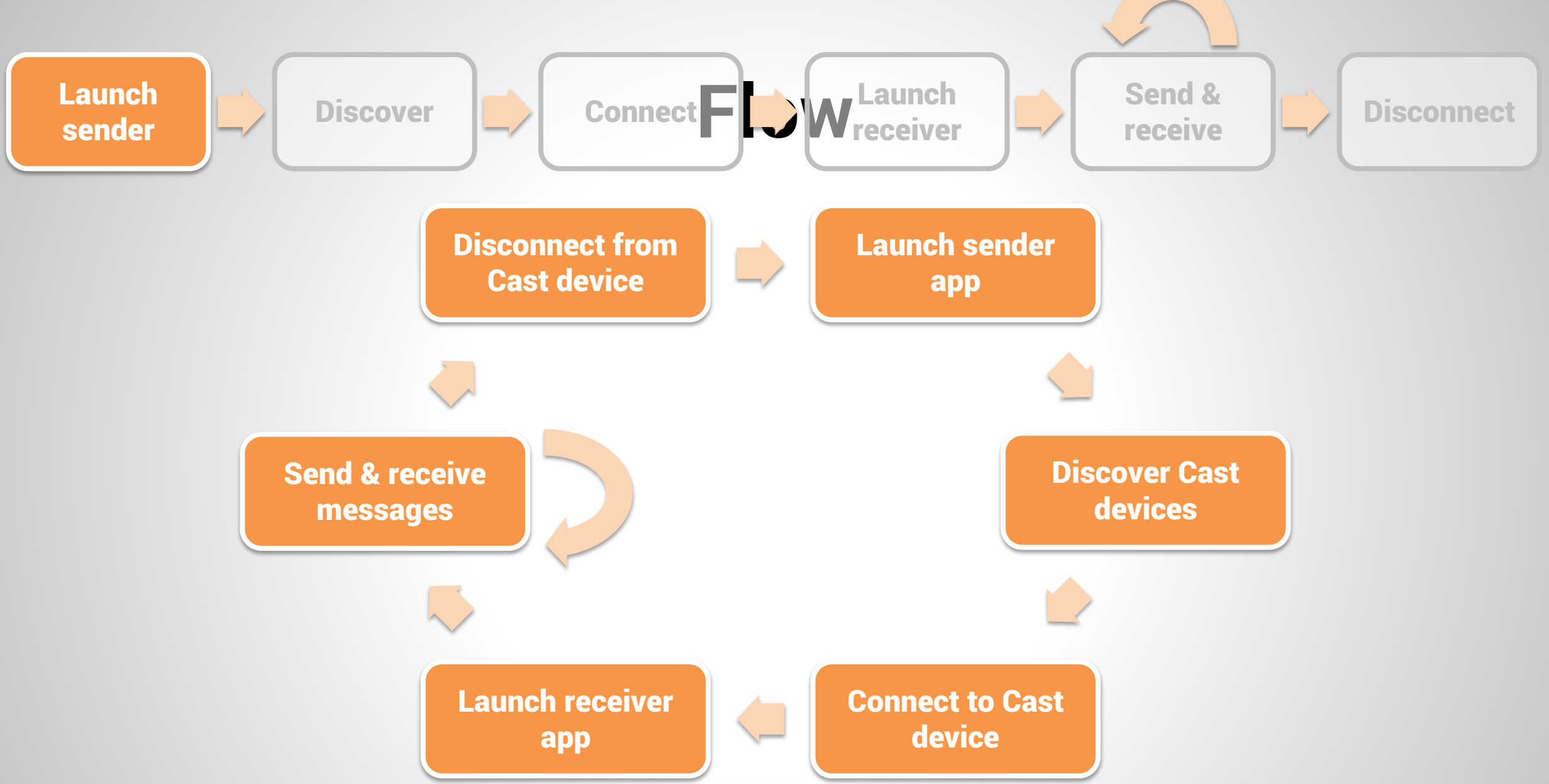
ADD NEW DEVICE

Pixplicity®

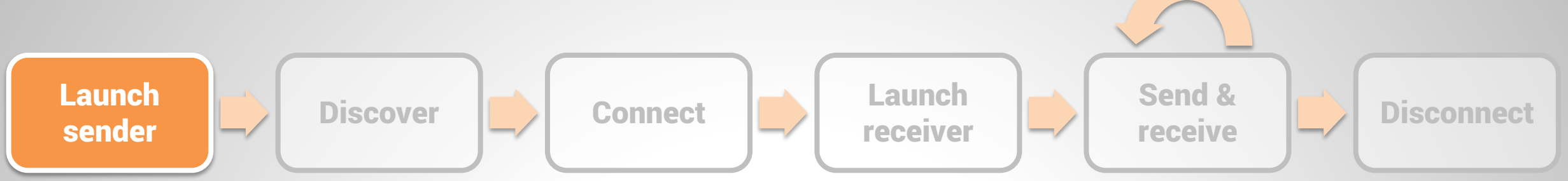
...and more preparation...

- android-support-v7-mediarouter
  - Requires android-support-v7-appcompat
- Google Play Services
- Optional: CastCompanionLibrary
  - Will offload a lot of work
  - Loads of code; could be intimidating

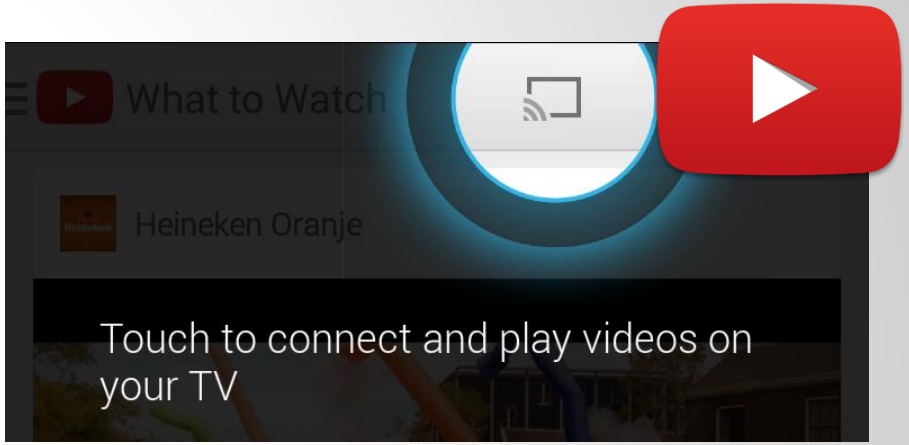
Flow

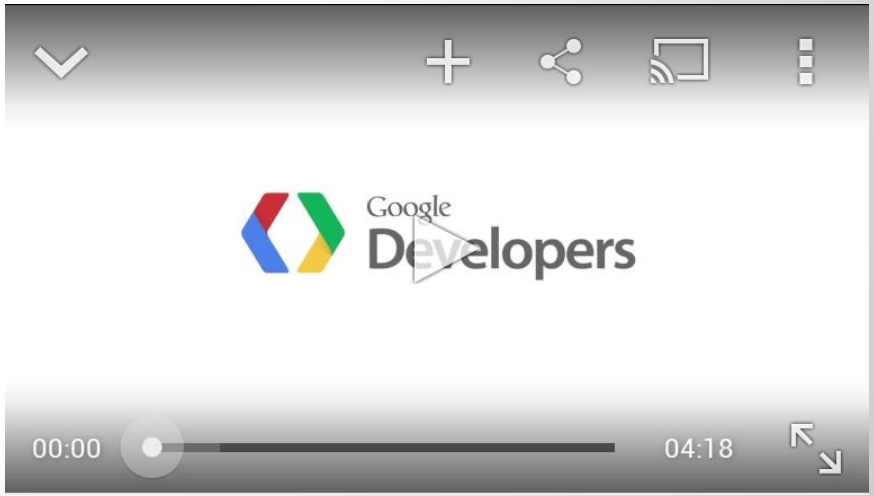


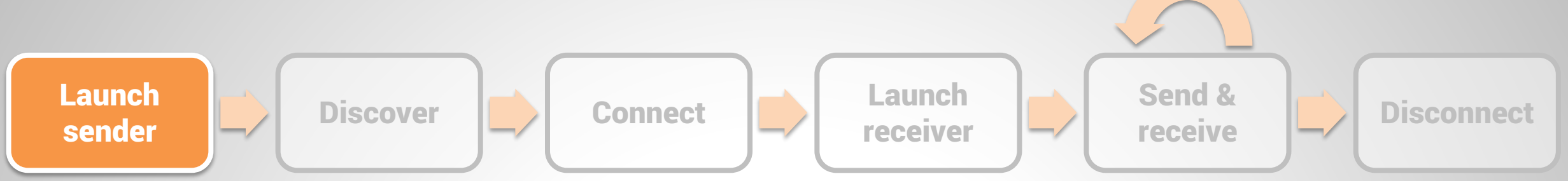
Pixplicity®



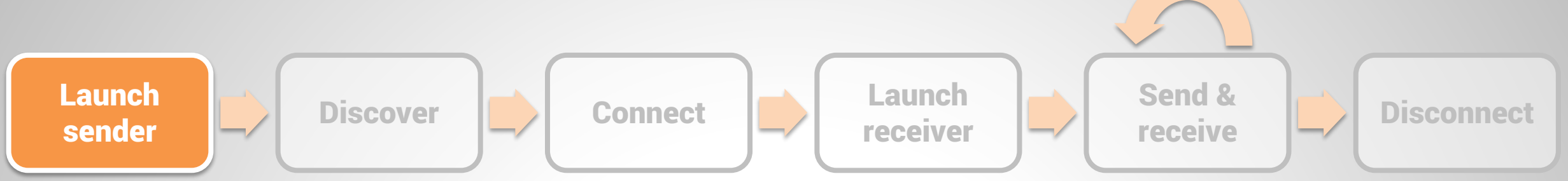
- **Add button for casting:**
  - **MediaRouterActionProvider**
  - **MediaRouter Button**
  - **Your own**
- **Only show it when Cast devices are discovered**



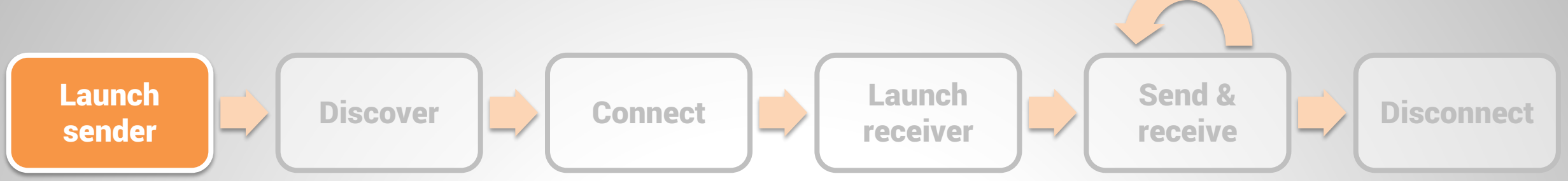




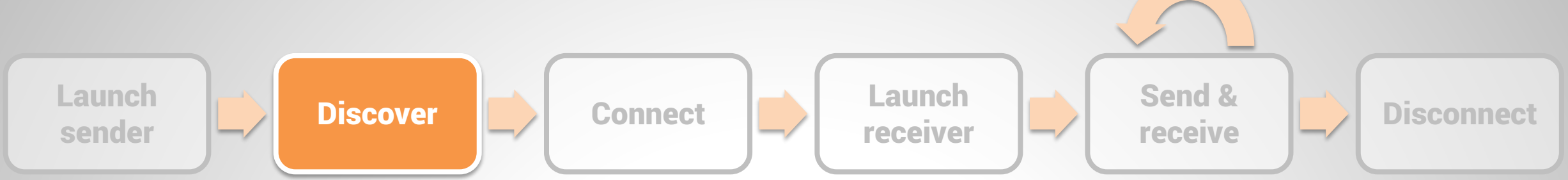

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto">

    <item android:id="@+id/media_route_menu_item"
        android:title="@string/media_route_menu_title"
        app:showAsAction="always"
        app:actionProviderClass="android.support.v7.app.MediaRouteActionProvider"/>

</menu>
```



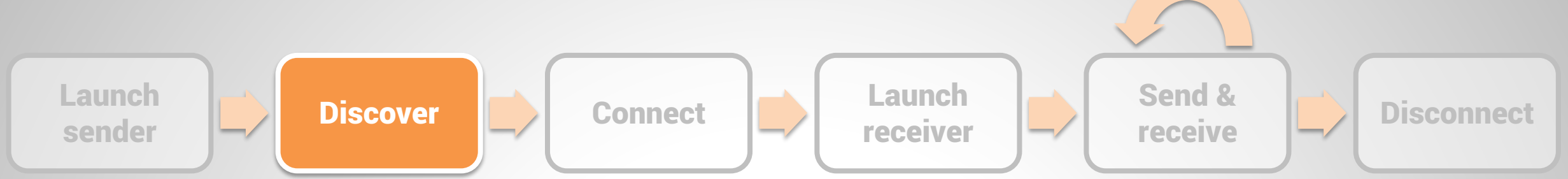
- **Get the MediaRouter instance**
- **Create a MediaRouteSelector for Cast apps**
- **Create a MediaRouter.Callback instance**



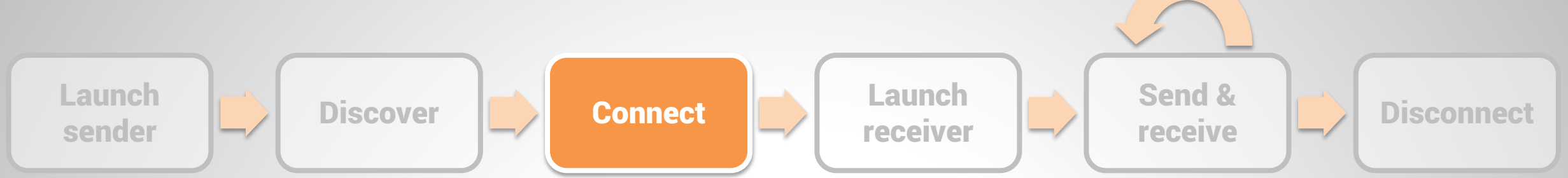

```
mMediaRouter = MediaRouter.getInstance(getApplicationContext());

// Create a MediaRouteSelector
mMediaRouteSelector = new MediaRouteSelector.Builder()
        .addControlCategory(CastMediaControlIntent.categoryForCast(CAST_APP_ID))
        .build();

// Create a MediaRouter callback for discovery events
mMediaRouterCallback = new MyMediaRouterCallback();
```



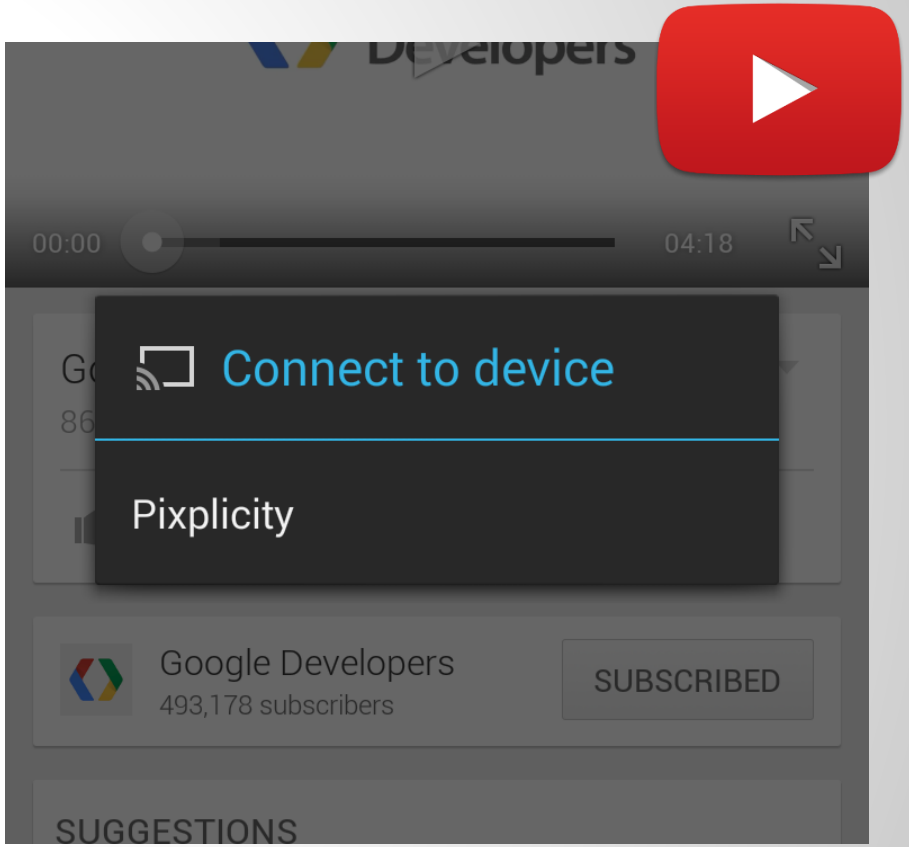
- **Add MediaRouter.Callback in onResume()**
- **Remove MediaRouter.Callback in onPause()**



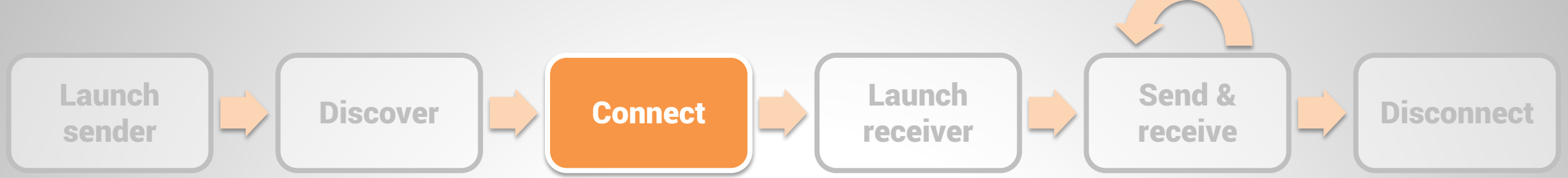

```
@Override
protected void onResume() {
    super.onResume();
    // Add the callback to start device discovery
    mMediaRouter.addCallback(mMediaRouteSelector, mMediaRouterCallback,
            MediaRouter.CALLBACK_FLAG_PERFORM_ACTIVE_SCAN);
}
@Override
protected void onPause() {
    // Remove the callback to stop device discovery
    mMediaRouter.removeCallback(mMediaRouterCallback);
    super.onPause();
}
```



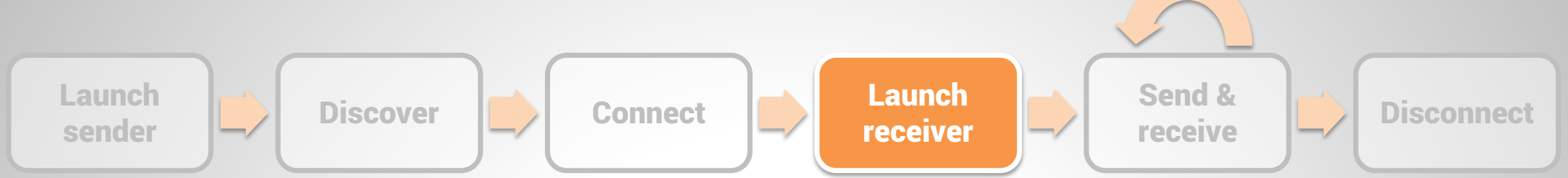
- **When a route is selected:**
  - **Get a CastDevice from the route**
  - **Connect a Google API Client**



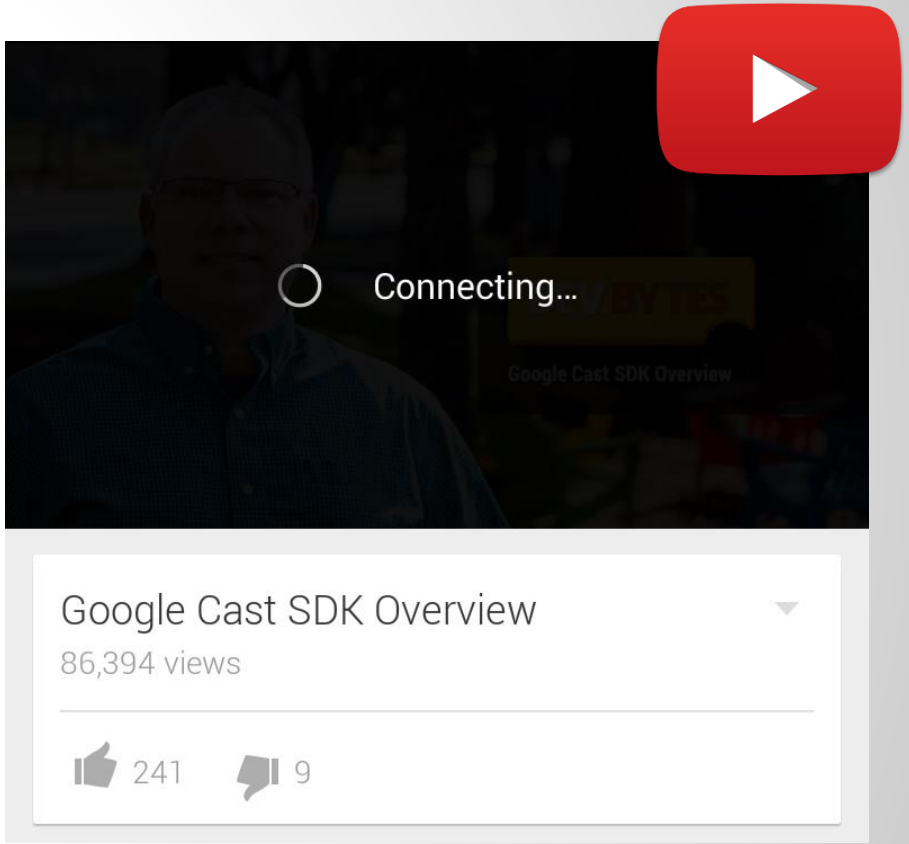
Pixplicity®

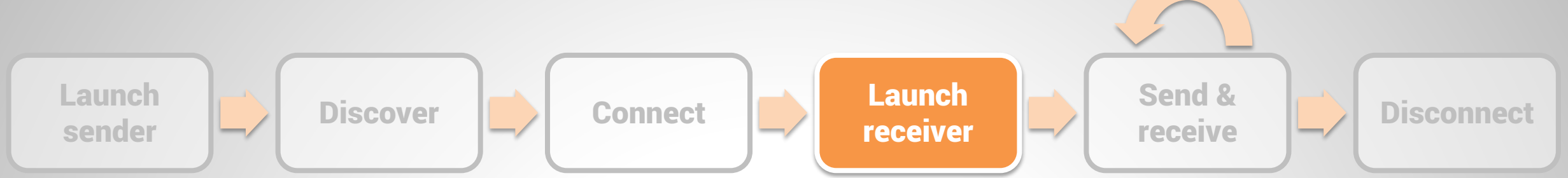



```
CastDevice device = CastDevice.getFromBundle(route.getExtras());

Cast.CastOptions.Builder apiOptionsBuilder = Cast.CastOptions
        .builder(device, mCastListener);
Builder builder = new GoogleApiClient.Builder(mContext)
        .addApi(Cast.API, apiOptionsBuilder.build())
        .addConnectionCallbacks(mConnectionCallbacks)
        .addOnConnectionFailedListener(...);

// Cast API requires a reference to this later on
mGoogleApiClient = builder.build();
// Do the magic
mGoogleApiClient.connect();
```



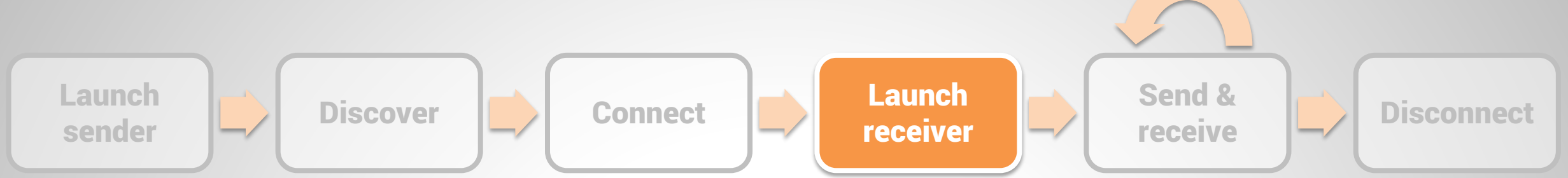
- **Launch Cast app or join an existing session**



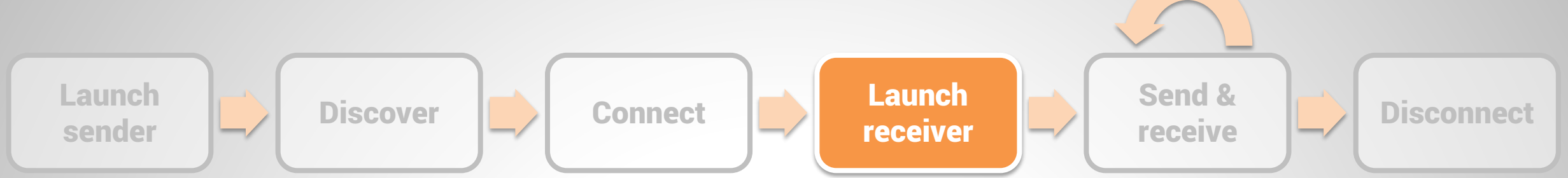


ConnectionCallbacks

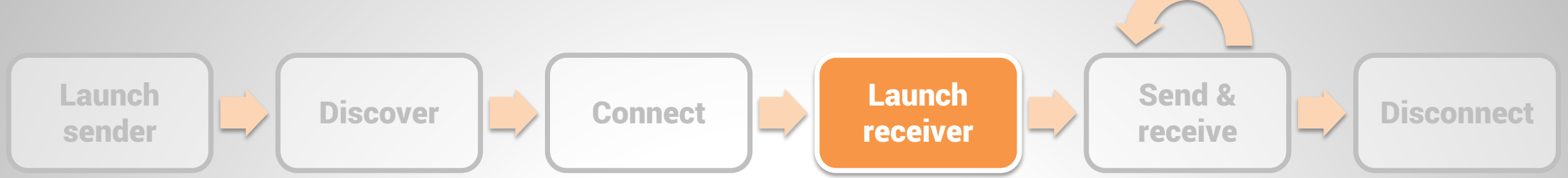
```
Builder builder = new GoogleApiClient.Builder(mContext)
        .addApi(Cast.API, apiOptionsBuilder.build())
        .addConnectionCallbacks(mConnectionCallbacks)
        .addOnConnectionFailedListener(...);
```



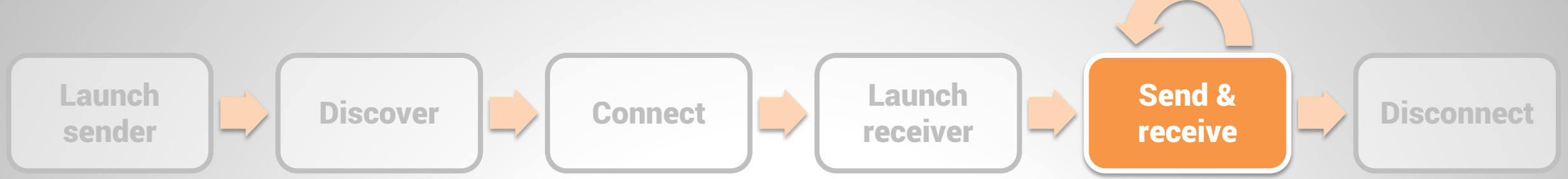

```
private class ConnectionCallbacks implements
        GoogleApiClient.ConnectionCallbacks {

    @Override
    public void onConnected(Bundle connectionHint) {
        launchApp();
    }

    @Override
    public void onConnectionSuspended(int cause) {
        // TODO Wait for next onConnected() and re-create the message channel
    }
}
```



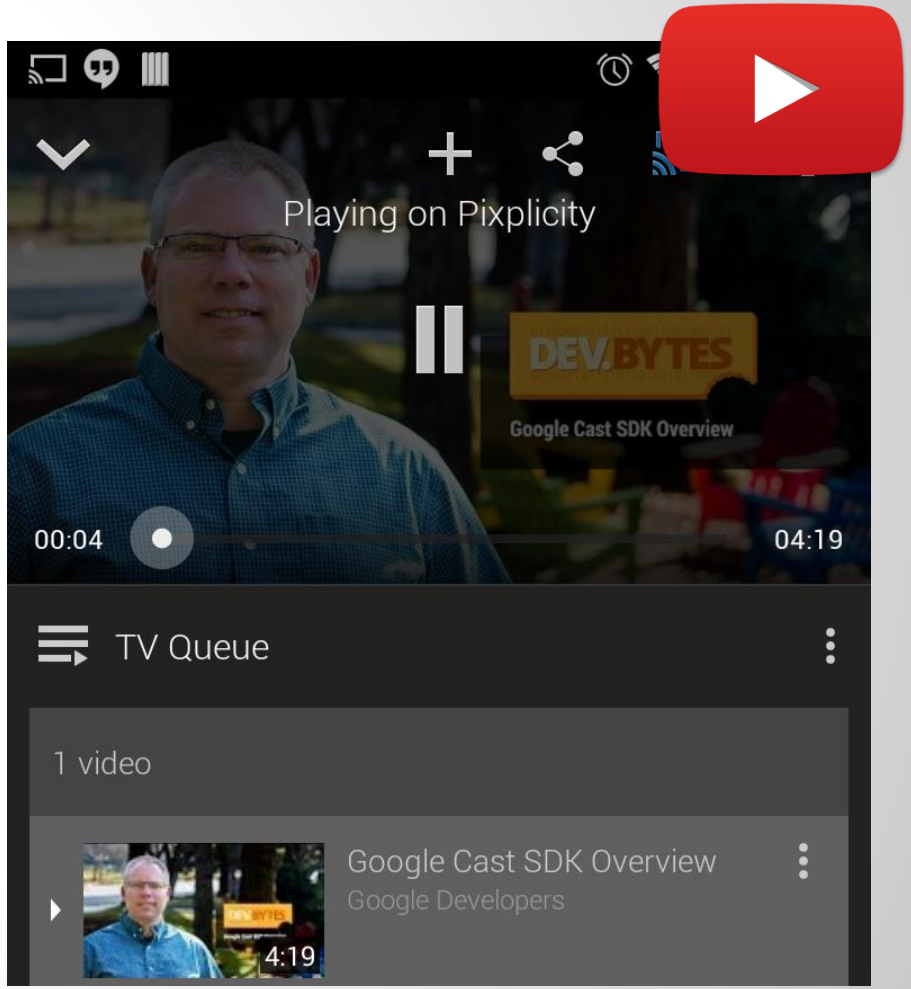

```
private void launchApp() {

    PendingResult<ApplicationConnectionResult> result;

    // Launch the receiver app
    result = Cast.CastApi.launchApplication(mGoogleApiClient, CAST_APP_ID, false);
    // For joining an existing session
  //result = Cast.CastApi.joinApplication(mGoogleApiClient, CAST_APP_ID, mSessionId);

    result.setResultCallback(mResultCallback);

}
```



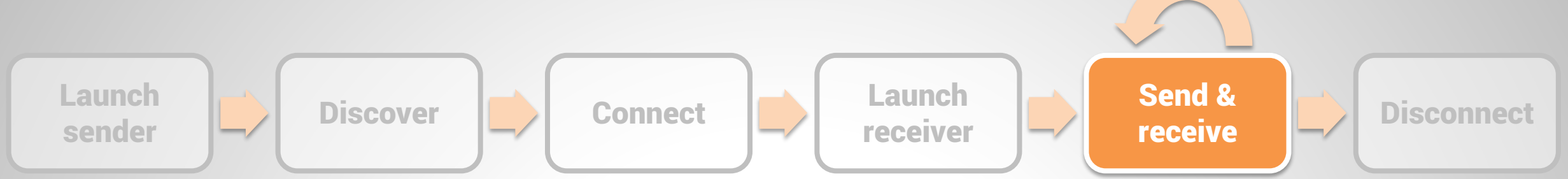

```
new ResultCallback<Cast.ApplicationConnectionResult>() {

    @Override
    public void onResult(ApplicationConnectionResult result) {
        if (result.getStatus().isSuccess()) {
            mHelloWorldChannel = new HelloWorldChannel();
            Cast.CastApi.setMessageReceivedCallbacks(
                                mApiClient,
                                mHelloWorldChannel.getNamespace(),
                                mHelloWorldChannel);
        } else { /* TODO disconnect */ }
    }
}
```



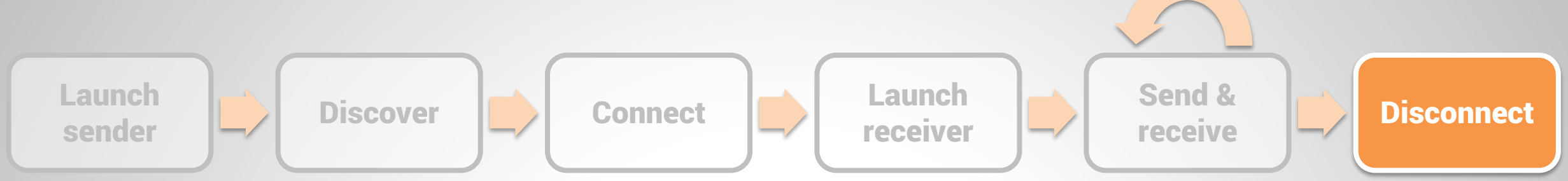
- **Exchange data over channel**
  - **Send messages**
  - **Receive messages**



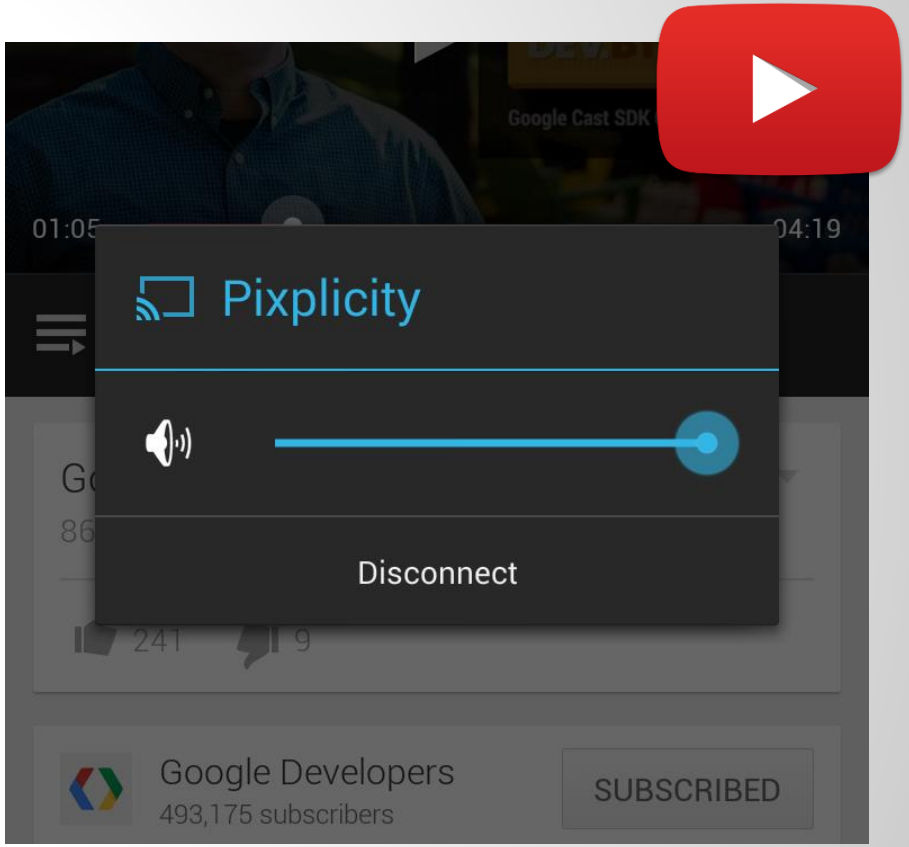
Pixplicity®



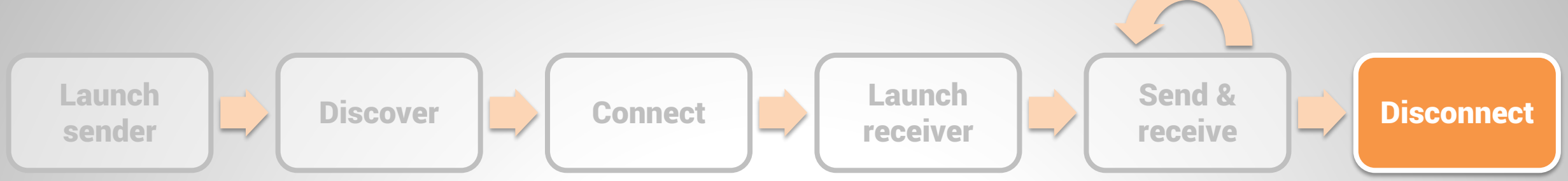

```
Cast.CastApi.sendMessage(mApiClient,
    mHelloWorldChannel.getNamespace(), message)
    .setResultCallback(new ResultCallback<Status>() {

        @Override
        public void onResult(Status result) {
            if (!result.isSuccess()) {
                Log.e(TAG, "Sending message failed");
            }
        }

    });
```



- **Handle disconnecting**
  - **From route unselection**
  - **From connection failure**
  - **When receiver app is stopped**
  - **onDestroy()**

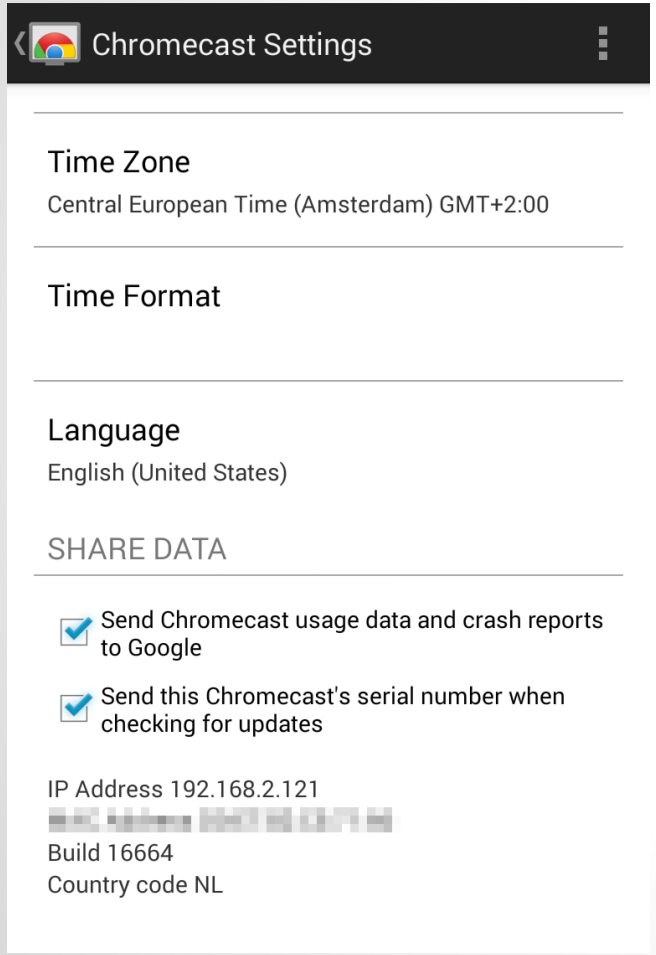


Pixplicity®




```
private void disconnect() {

    if (mApiClient != null && mApiClient.isConnected()) {
        Cast.CastApi.stopApplication(mApiClient, mSessionId);
        Cast.CastApi.removeMessageReceivedCallbacks(
                mApiClient,
                mHelloWorldChannel.getNamespace());
        mApiClient.disconnect();
        mHelloWorldChannel = null;
    }
    mApiClient = null;

}
```

Debugging



http://<ip-address>:9222

- **Console**
- **DOM inspector**

Pixplicity®

Gotchas!

Google

Google Cast SDK Developer Console

Overview
Applications
Devices
Settings

Welcome to the Google Cast SDK Developer Console

The Google Cast Developer Console enables developers to register applications and authorize devices for testing.

Applications

Application ID	Application Name	Status
	Group Gallery	Unpublished
916A9C15	Cast Demo	Unpublished

ADD NEW APPLICATION

Devices

Serial Number	Description	Status
	Paul & Thif	Ready For Testing
	Dylan	Ready For Testing

ADD NEW DEVICE

Chromecast Settings

Time Zone
Central European Time (Amsterdam) GMT+2:00

Time Format

Language
English (United States)

SHARE DATA

- [x] Send Chromecast usage data and crash reports to Google
- [x] Send this Chromecast's serial number when checking for updates

IP Address 192.168.2.121

Build 16664
Country code NL

Pixplicity®

Gotchas!

- Coping with the lifecycle asynchronously
- Unexpected disconnects
- Forgetting to start the MediaRouter scan

Useful stuff

- **The docs**
  https://developers.google.com/cast/
- **The samples**
  https://github.com/googlecast
- **The community**
  http://tiny.cc/castcommunity

Enjoy Creating your own

Google Cast App!

Paul Lammertsma
CTO, Pixplicity

Pixplicity®