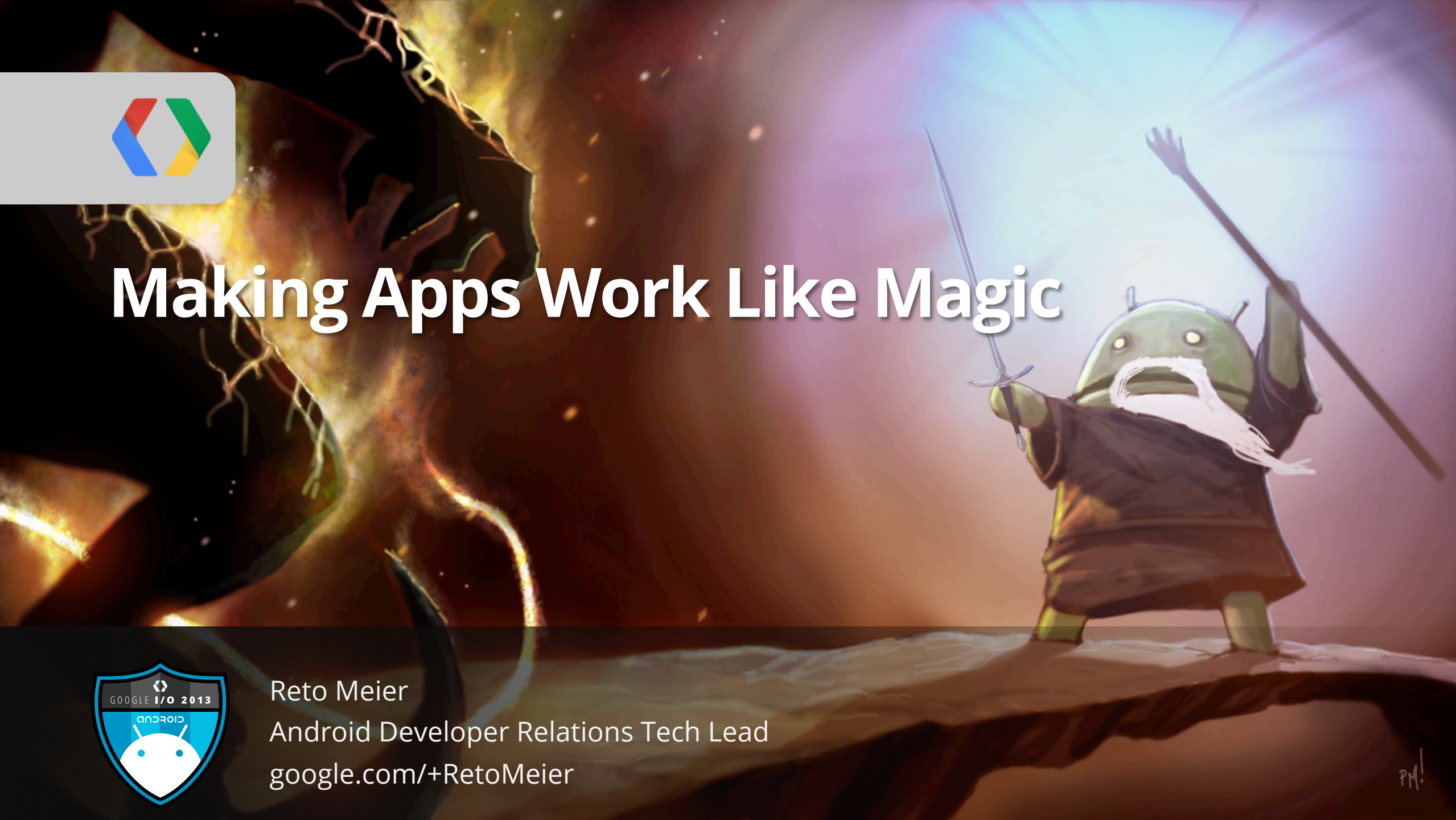
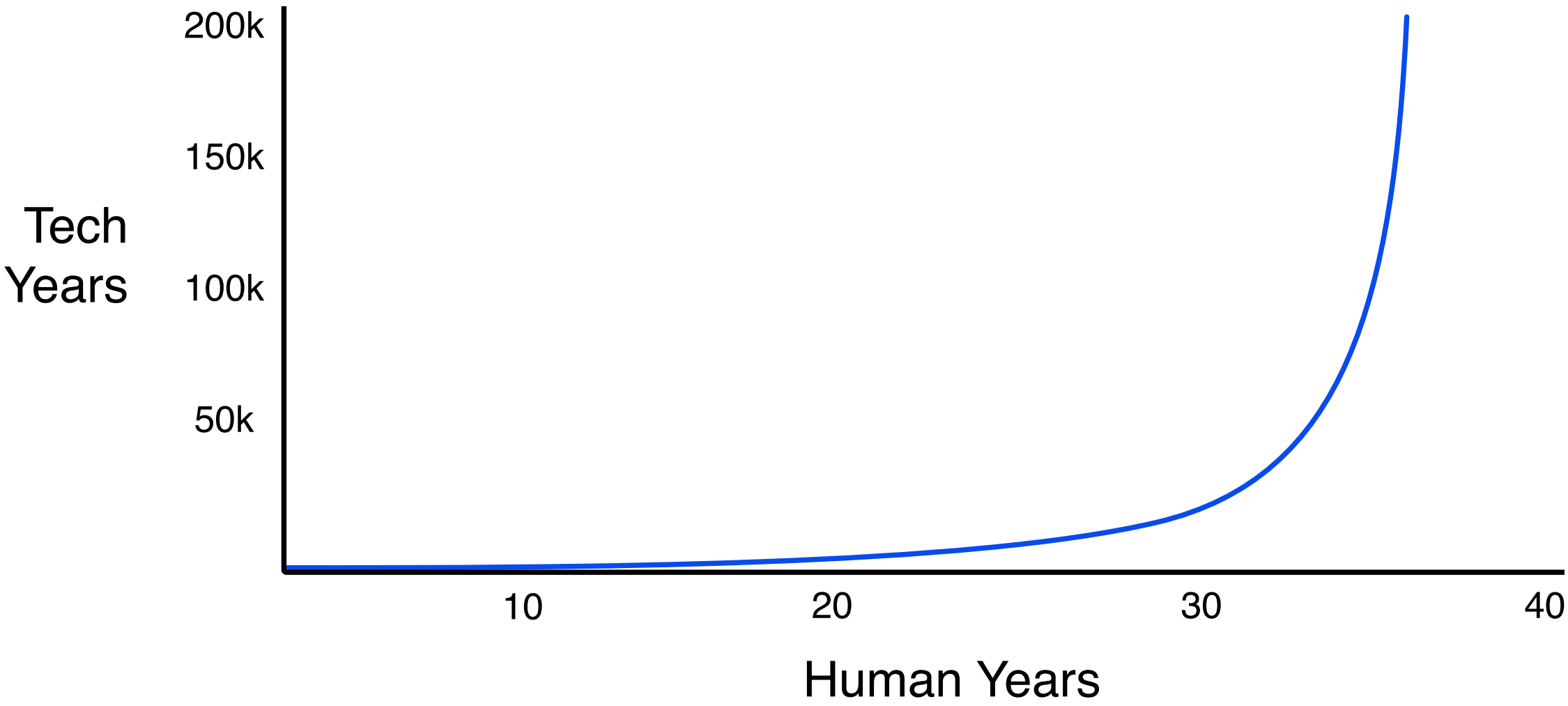
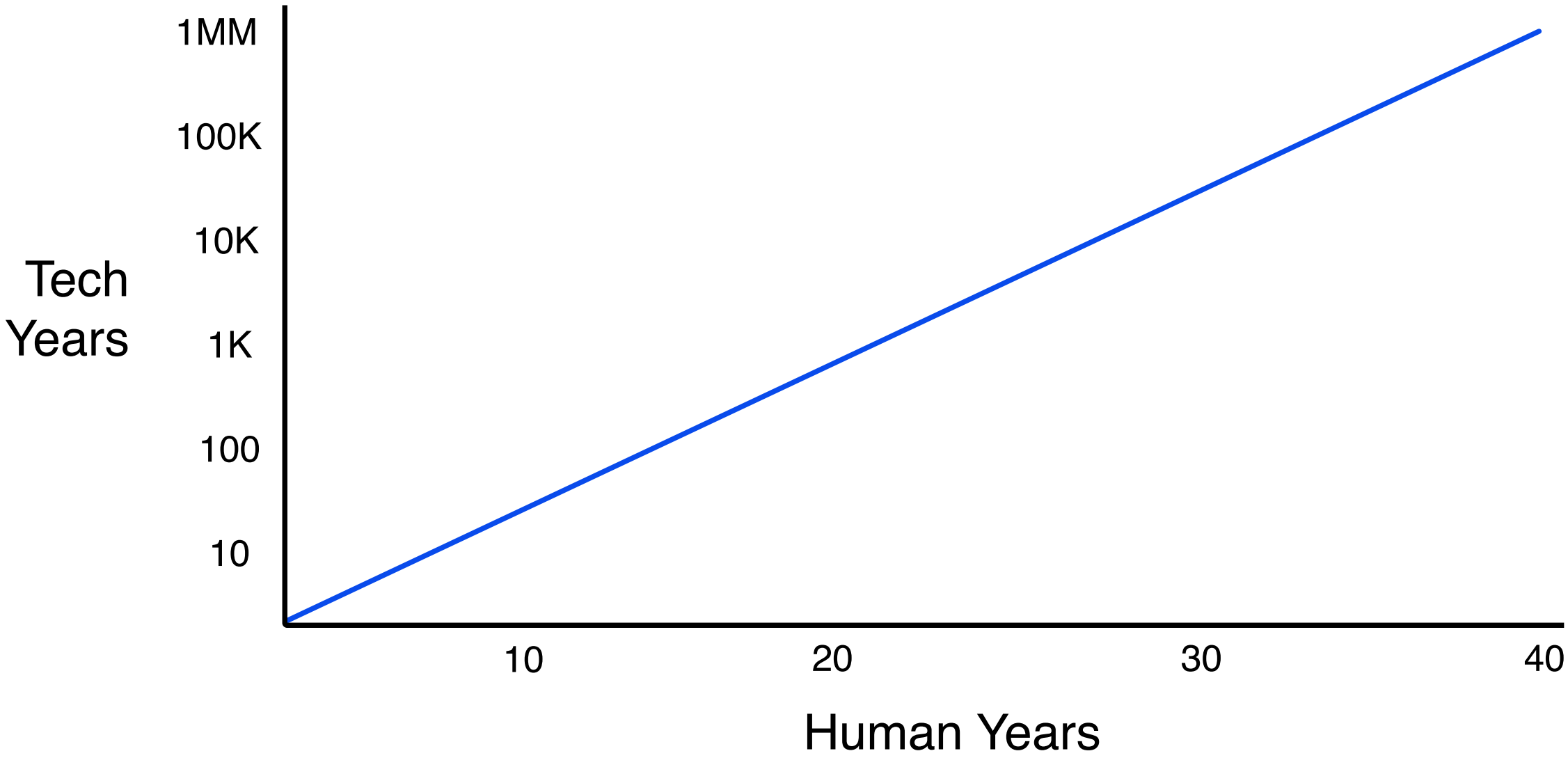


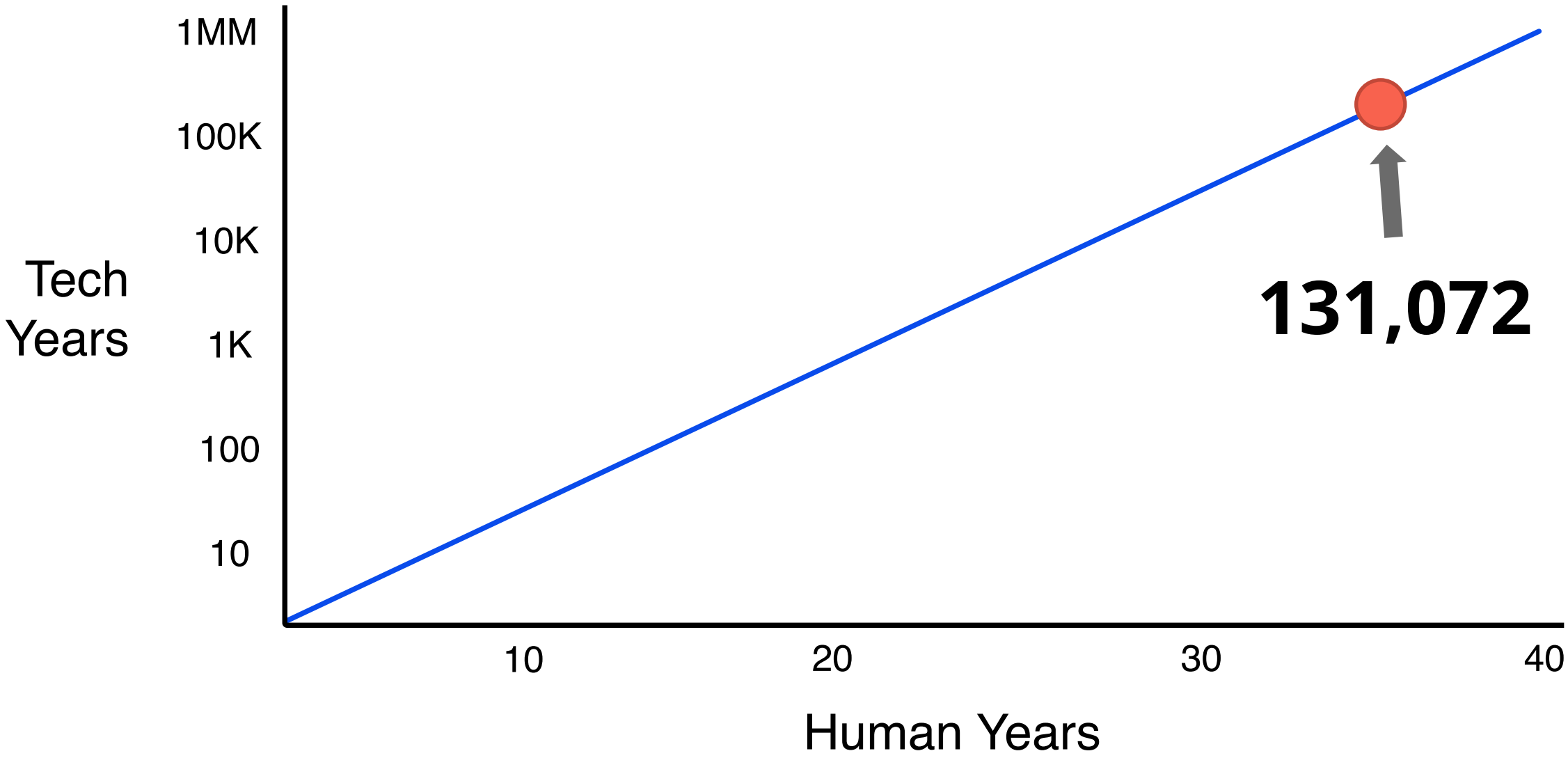
Making Apps Work Like Magic

Reto Meier
Android Developer Relations Tech Lead
google.com/+RetoMeier

Moore's Age




1MM
100K
10K
1K
100
10
Tech Years
10
20
30
40
Human Years

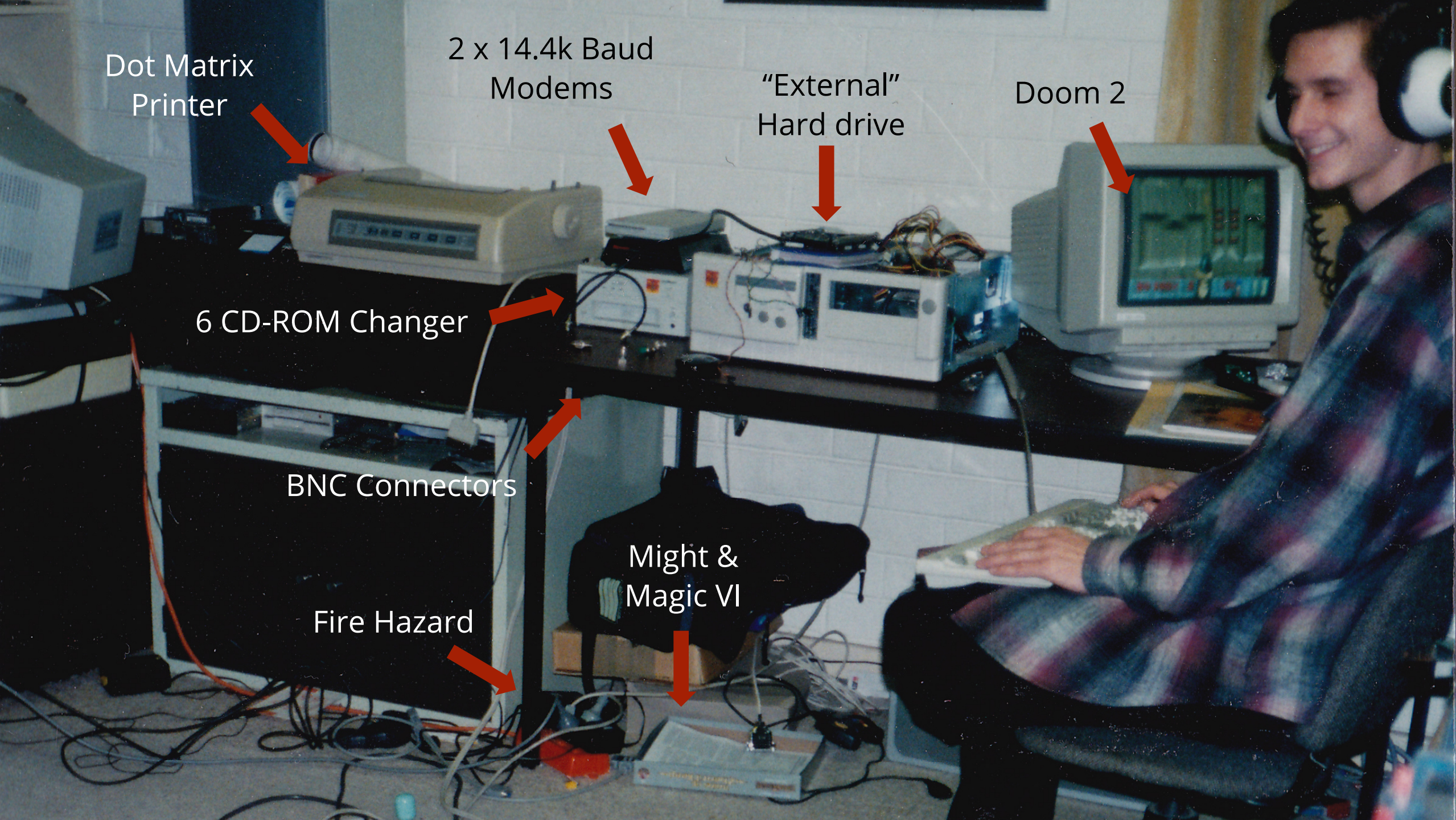

1MM
100K
10K
1K
100
10
Tech Years
131,072
10
20
30
40
Human Years

IBM
SOMCO

	12 Pitch		Proportional Spacing
RATOR	COURIER	DUAL GOTHIC	MODERN

“Any significantly advanced technology is indistinguishable from magic.”

Arthur C. Clarke


Dot Matrix Printer
2 x 14.4k Baud Modems
"External" Hard drive
Doom 2
6 CD-ROM Changer
BNC Connectors
Might & Magic VI
Fire Hazard

The Radioactive Yak

MONDAY, AUGUST 09, 2010

The Future of Mobile: Invisible, connected devices with infinite screens

In 10 years...[I'll] put on my glasses...to see an augmented version of reality.

and embarrassment. But I won't let that stop me.

Bigger screens are better

Mobile devices are morphing. Tablets have been talked about for years, and the iPad and Kindle provide the kind of experience people have been waiting for. Browsing pictures, watching videos, and reading books work really well on a screen that size.

Still, I find the iPad heavy and bulky. The ultimate device would be the size and weight of my mobile but include a screen that could be unfolded or rolled out to provide a better display for watching movies and playing games.

Actually, the *ultimate* device would be entirely virtual. I'd put on my glasses (or contact lenses) and look at any surface to see an augmented version of reality. Anything from interactive holographs, to augmented reality, or a cinema screen that stretches across the horizon. Everyone could see their own version of reality on a screen the size of their visual field.

...virtual keyboards and voice input [will] eliminate physical keyboards entirely.

For those who want to write something longer than an email, gesture recognition (capable of tracking fingers), combined with eye-focus tracking will provide virtual full-size keyboards.

If we're thinking long-term, we can look forward to research like this letting us control our devices using our minds.

- **1 year** Wireless keyboards, voice input, and gestures.
- **5 years** Larger multitouch screens, better gesture input, and flawless voice recognition.
- **10 years** Full virtual keyboards and voice input eliminate physical keyboards entirely.
- **20 years** Mind control.

Smaller devices that last longer

+1 209

Search

but this is my personal blog. The views expressed on these pages are mine alone and not those of my employer.

Follow me on

FOLLOW ME ON twitter

- Upcoming Public Appearances
- Reto Meier's Reading List
- Reto Meier's Gadget Compendium

Buy from Amazon US
Buy from Amazon UK
Buy from Google Play Books
Buy for Kindle
Buy DRM-free eBook
Read at Safari Books online

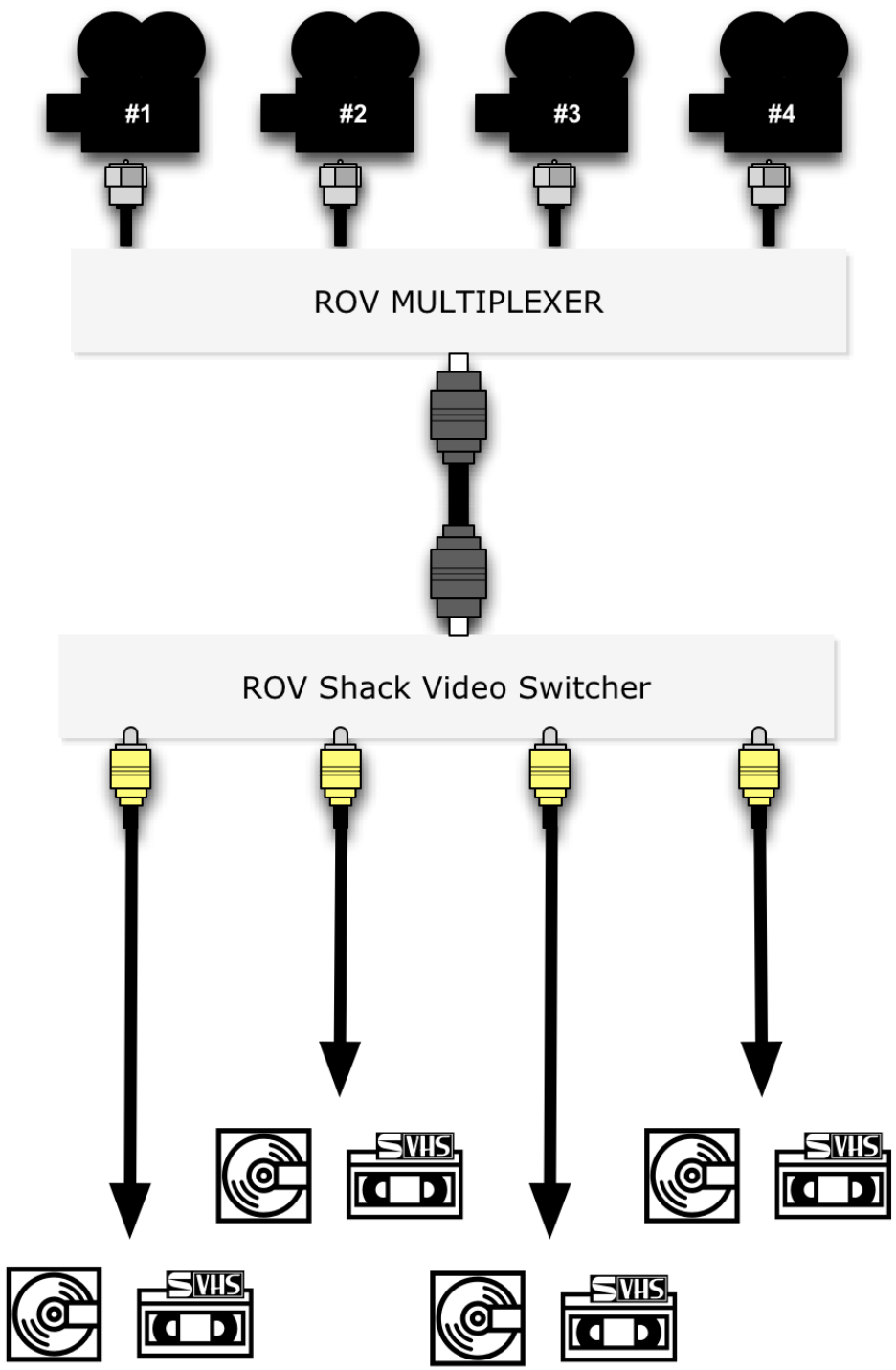
“The only way of discovering the limits of the possible is to venture a little way past it into the impossible.”

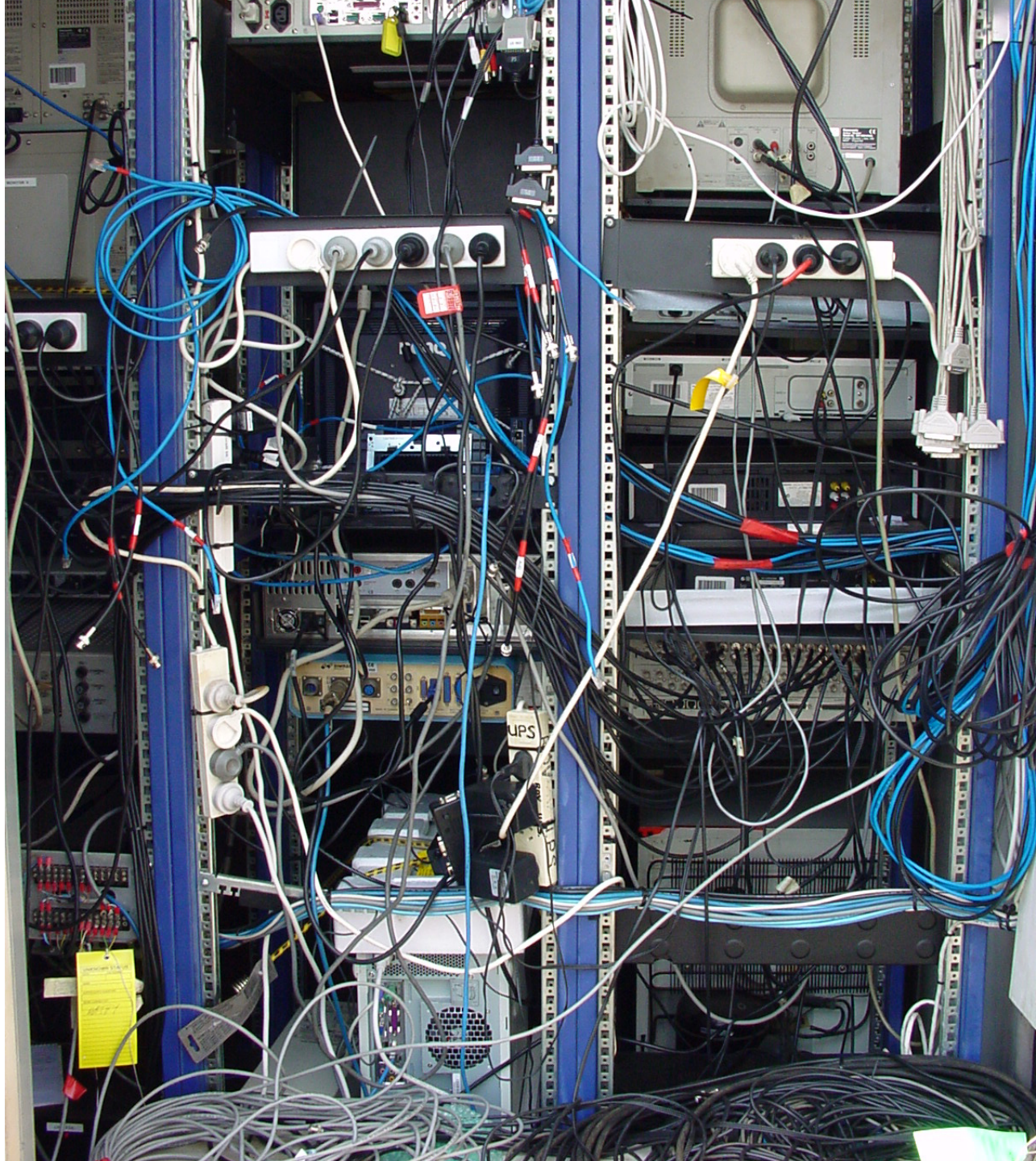
Arthur C. Clarke

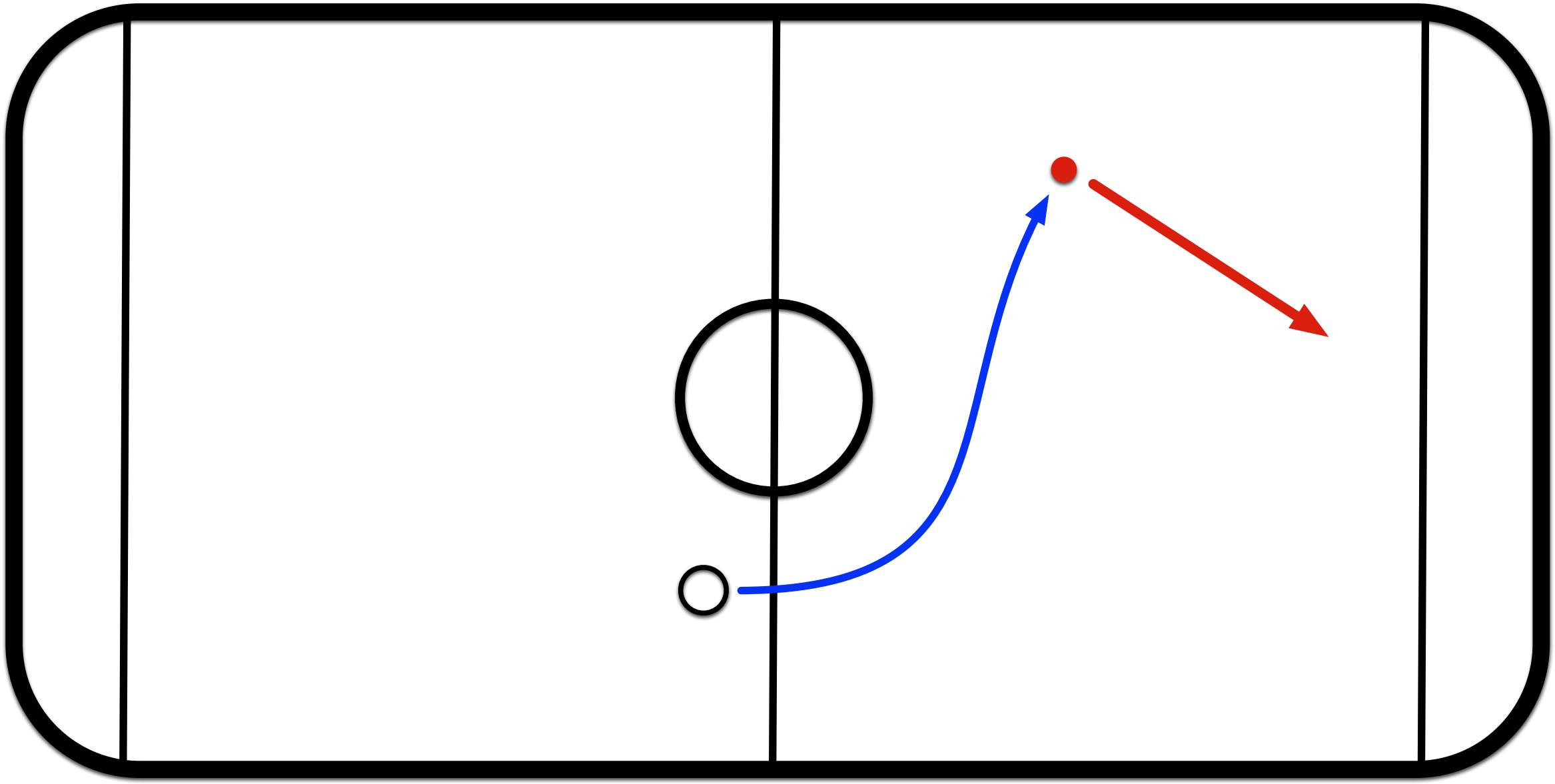
Developing the Impossible

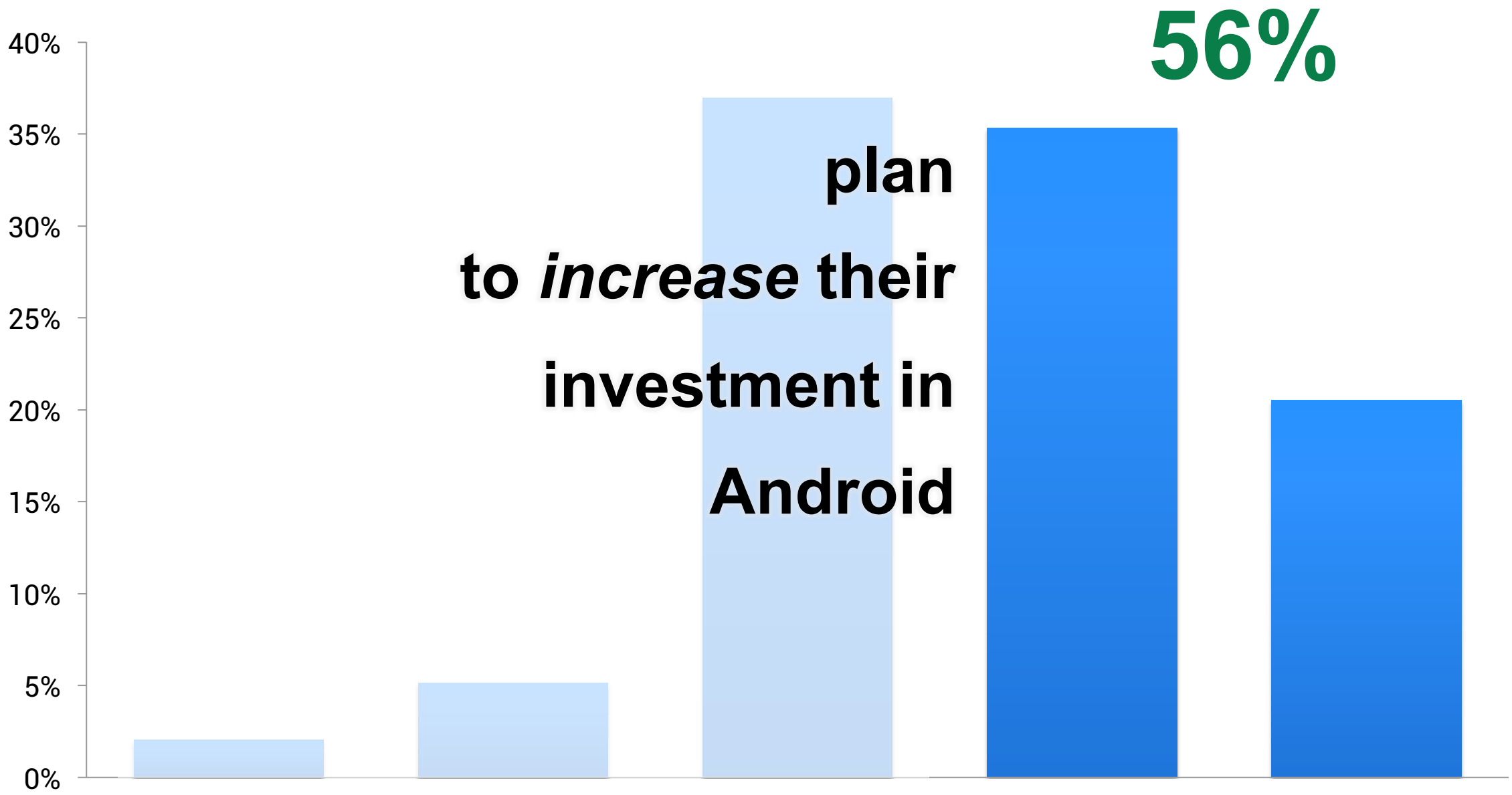
“Is that actually doing anything, or is it just a mock?”

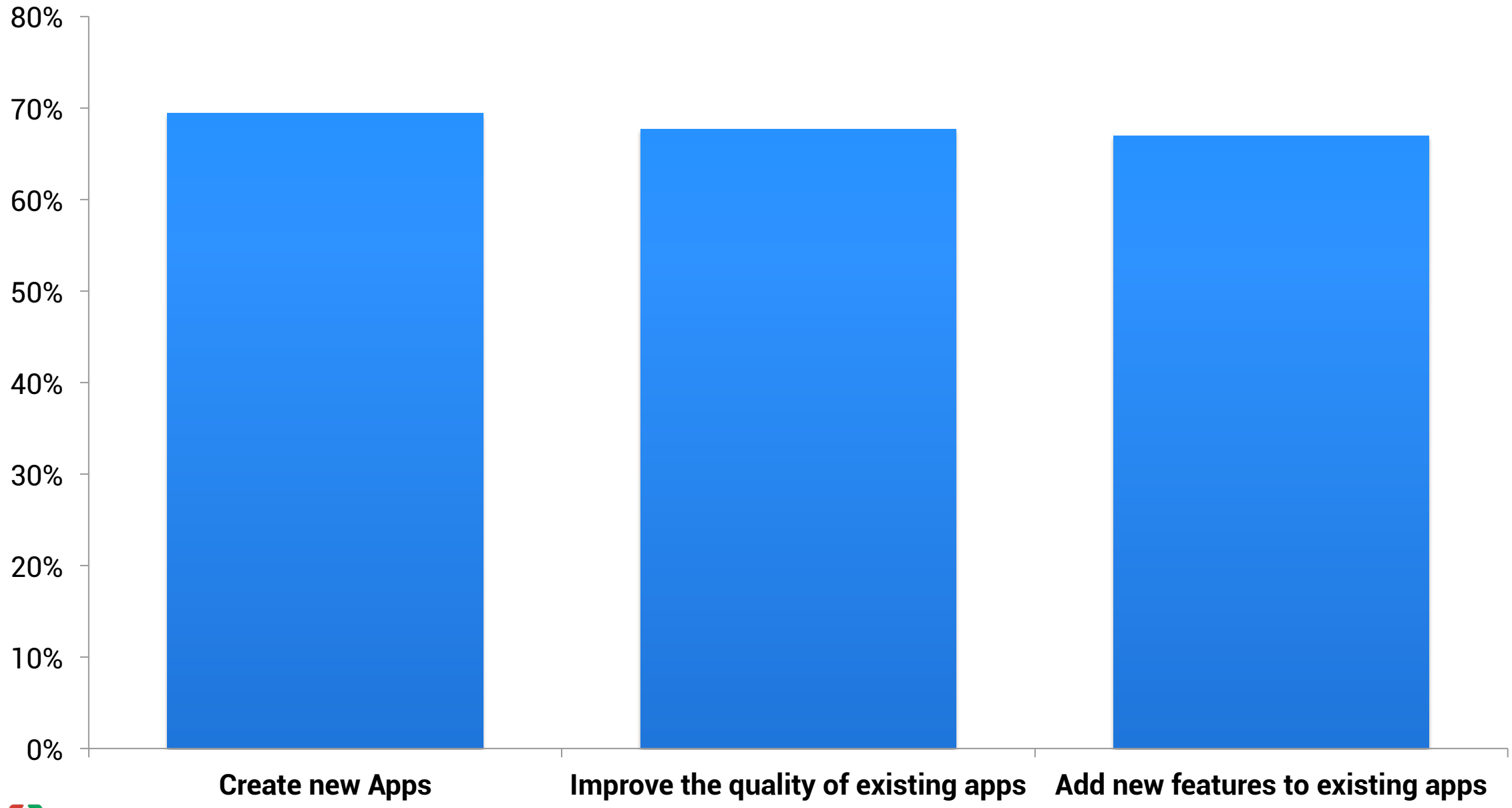
A. N. Engineer

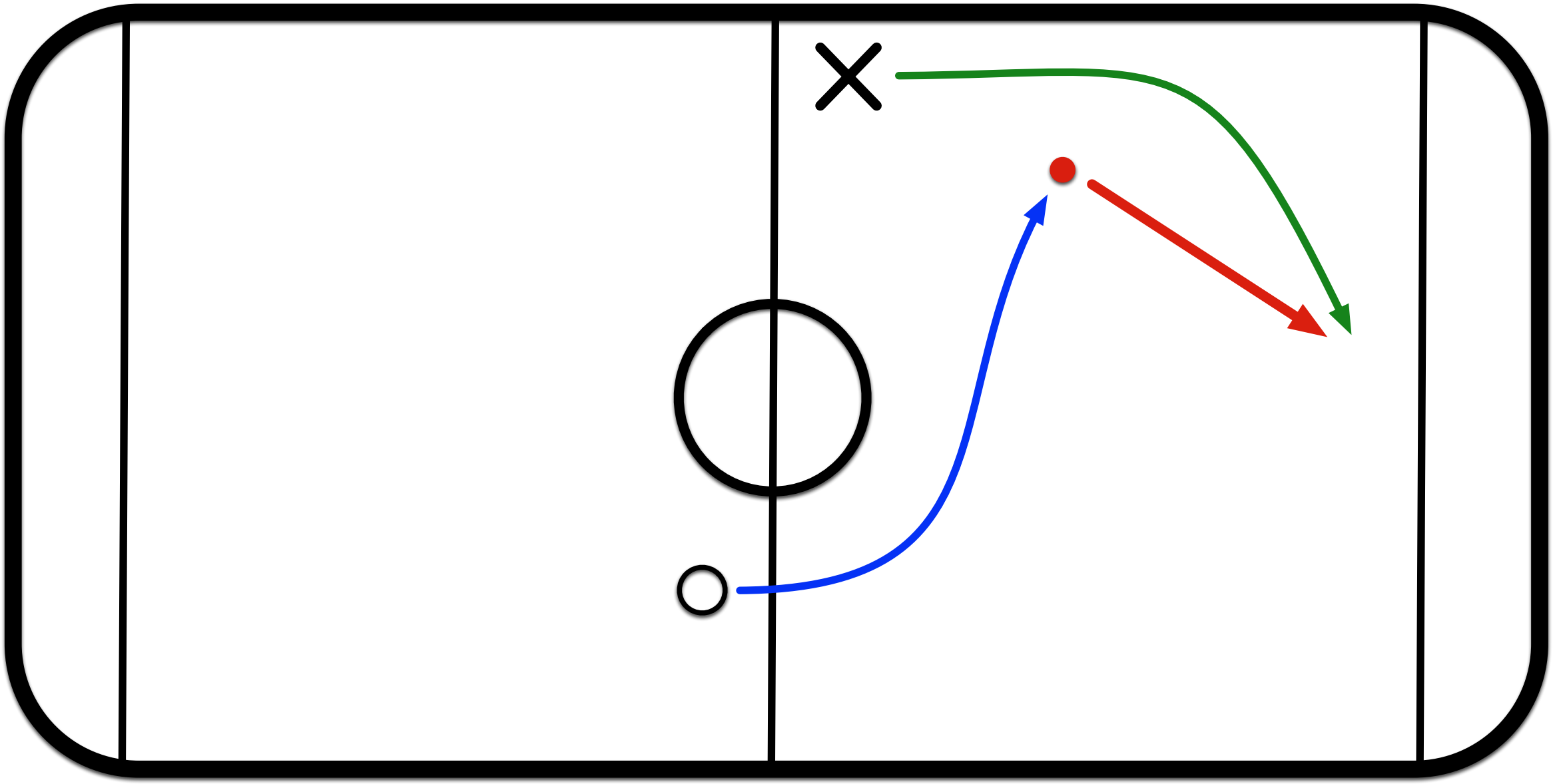

#1
#2
#3
#4
ROV MULTIPLEXER
ROV Shack Video Switcher
SVHS
SVHS
SVHS
SVHS

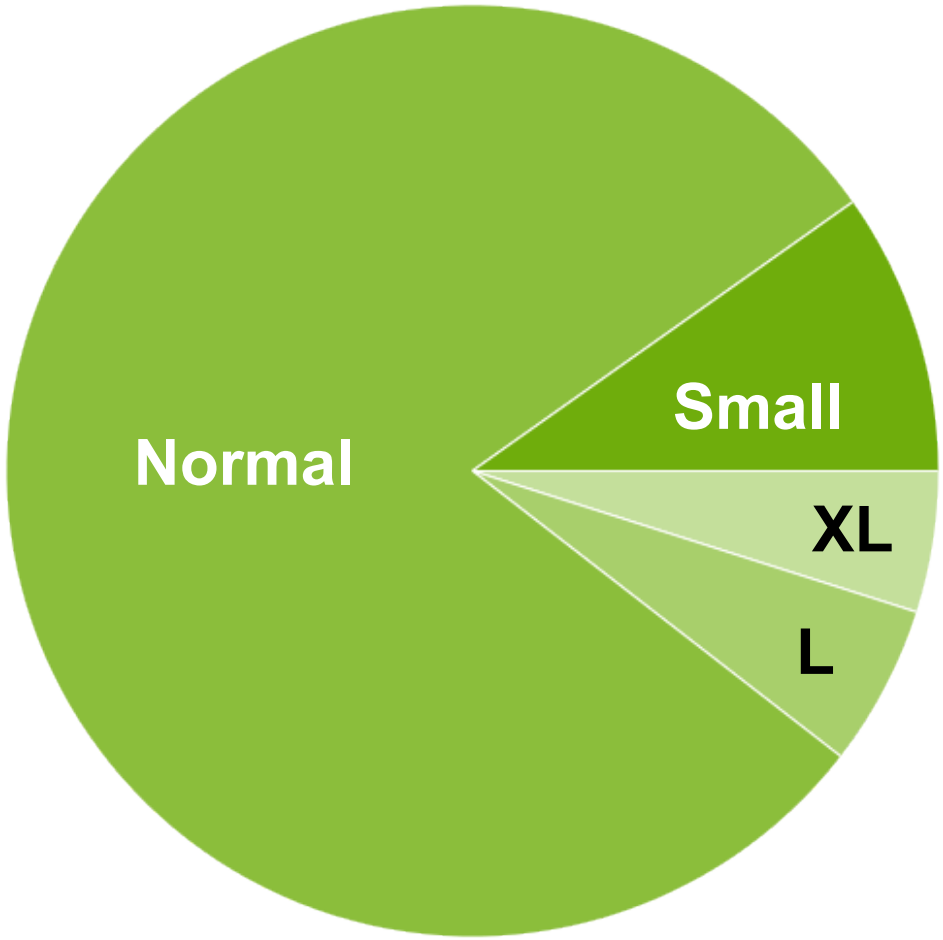

UPS

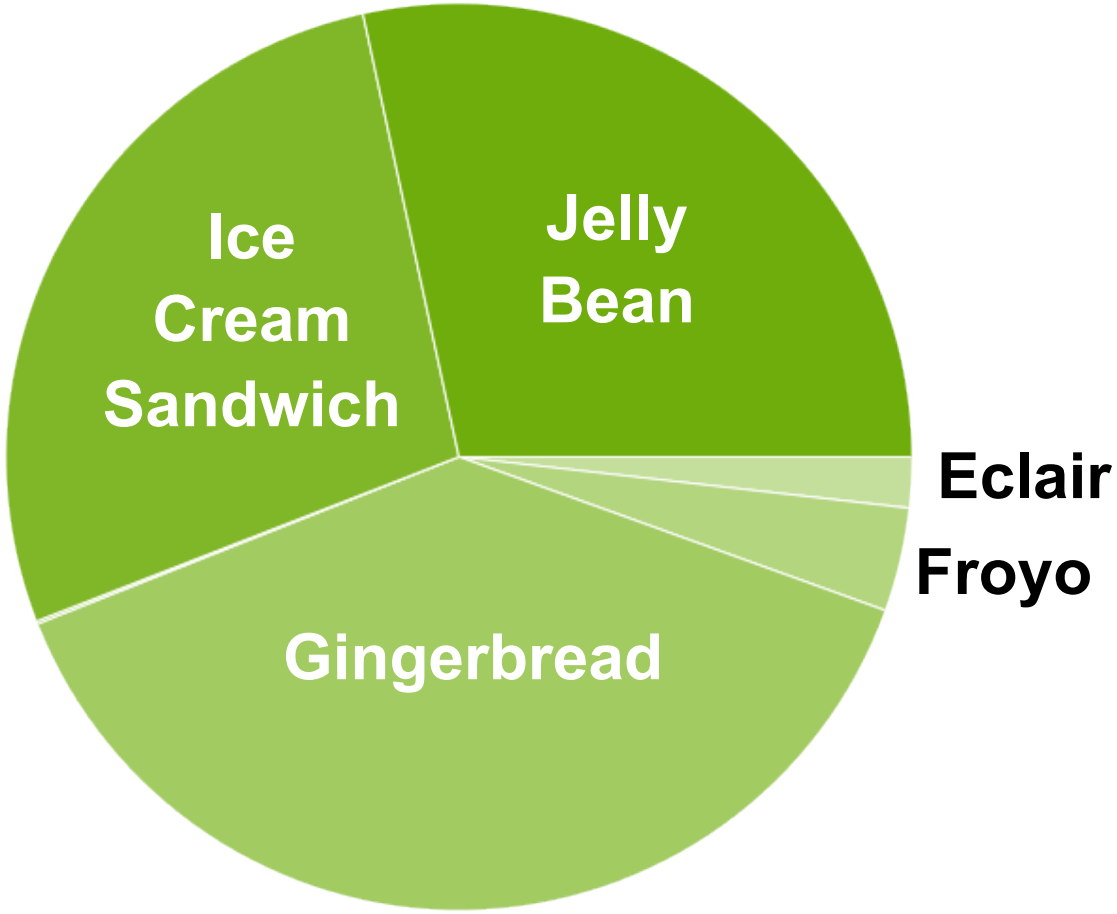


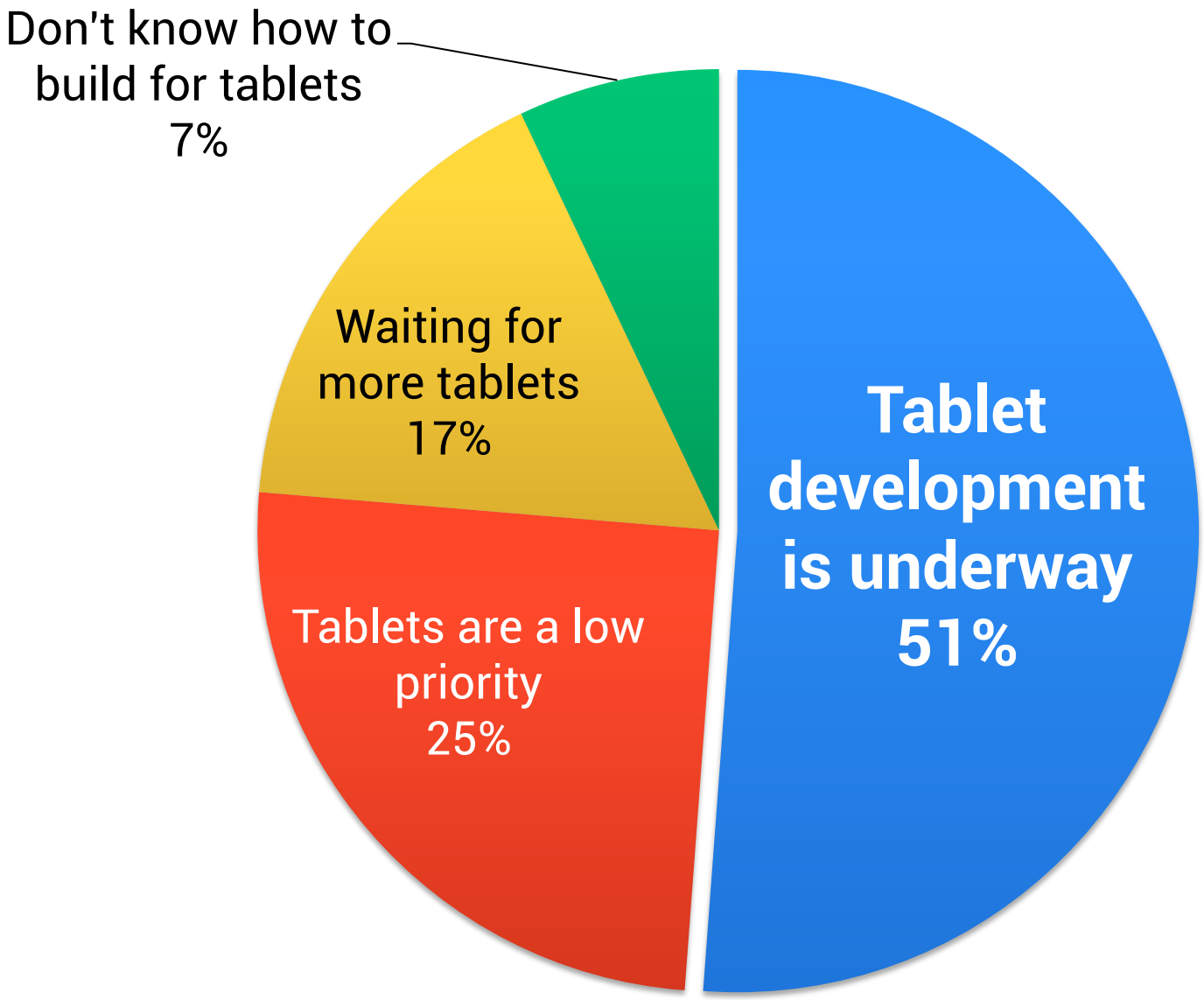

56%
plan
to *increase* their
investment in
Android
40%
35%
30%
25%
20%
15%
10%
5%
0%

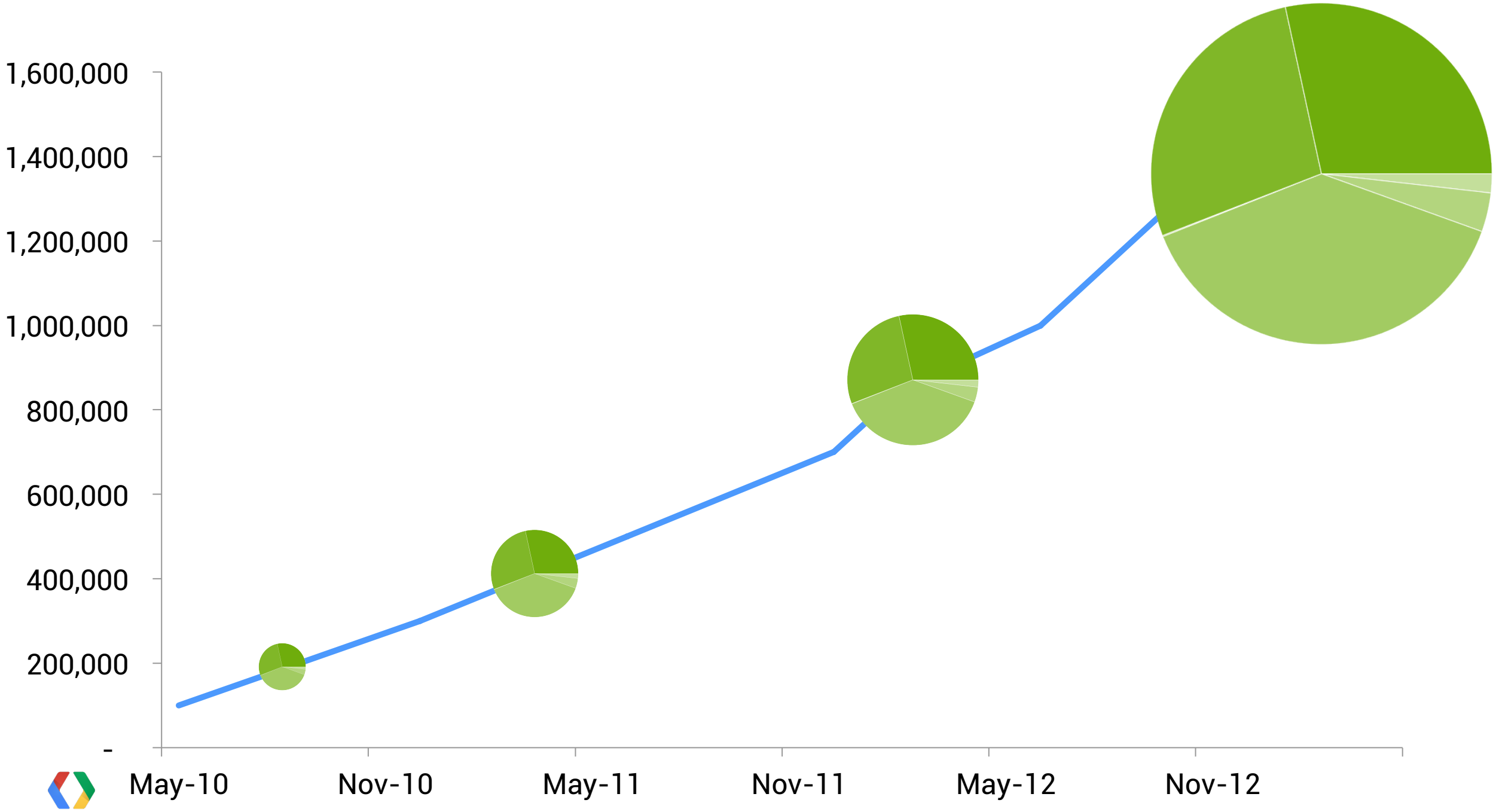

80%
70%
60%
50%
40%
30%
20%
10%
0%
Create new Apps
Improve the quality of existing apps
Add new features to existing apps




Normal
Small
XL
L


Ice Cream Sandwich
Jelly Bean
Eclair
Froyo
Gingerbread


Don't know how to build for tablets 7%
Waiting for more tablets 17%
Tablets are a low priority 25%
Tablet development is underway 51%


1,600,000
1,400,000
1,200,000
1,000,000
800,000
600,000
400,000
200,000
-
May-10
Nov-10
May-11
Nov-11
May-12
Nov-12

Create the best possible app
for *every* user

Android Beam

MyActivity

```
@Override
protected void onResume() {
  super.onResume();
  NfcAdapter nfcAdapter = NfcAdapter.getDefaultAdapter(this);
  nfcAdapter.setNdefPushMessageCallback(new CreateNdefMessageCallback() {
    public NdefMessage createNdefMessage(NfcEvent event) {
      NdefMessage message = createMessage();
      return message;
    }
  }, this);
  ...
}
```

Android Beam

MyActivity

```
private void createMessage() {
  String mimeType = "application/com.myapp.nfcbeam";
  byte[] mimeBytes = mimeType.getBytes(Charset.forName("US-ASCII"));

  String payLoad = // TODO Contextualized payload.

  NdefMessage nfcMessage = new NdefMessage(new NdefRecord[] {
    new NdefRecord(ndefRecord.TNF_MIME_MEDIA, mimeByes, new byte[],
    payload.getBytes()),

    NdefRecord.createApplicationRecord("com.myapp.nfcbeam")
  });
}
```

Android Beam

Manifest.xml

```
<intent-filter>
  <action android:name="android.nfc.action.NDEF_DISCOVERED"/>
  <category android:name="android.intent.category.DEFAULT"/>
  <data android:mimeType="application/com.myapp.nfcbeam"/>
</intent-filter>
```

Android Beam

MyActivity

```
private void extractPayload(Intent beamIntent) {
  Parcelable[] messages = beamIntent.getParcelableArrayExtra(
    NfcAdapter.EXTRA_NDEF_MESSAGES);

  NdefAdapter message = (NdefMessage)message[0];
  NdefRecord record = message.getRecords()[0];

  String payload = new String(record.getPayload());
  navigateTo(payload);
}
```

Lockscreen Widgets

Manifest.xml

```
<appwidget-provider ..."
   ...
   android:widgetCategory="keyguard|home_screen"
</appwidget-provider>
```

Lockscreen Widgets

MyAppProvider

```
Bundle opt = appWidgetManager.getAppWidgetOptions(widgetId);

int cat = opt.getInt(AppWidgetManager.OPTION_APPWIDGET_HOST_CATEGORY, -1);

boolean isKeyguard = cat == AppWidgetProviderInfo.WIDGET_CATEGORY_KEYGUARD;
int baseLayout = isKeyguard ? R.layout.keyguard_widget_layout :
                              R.layout.widget_layout;
```

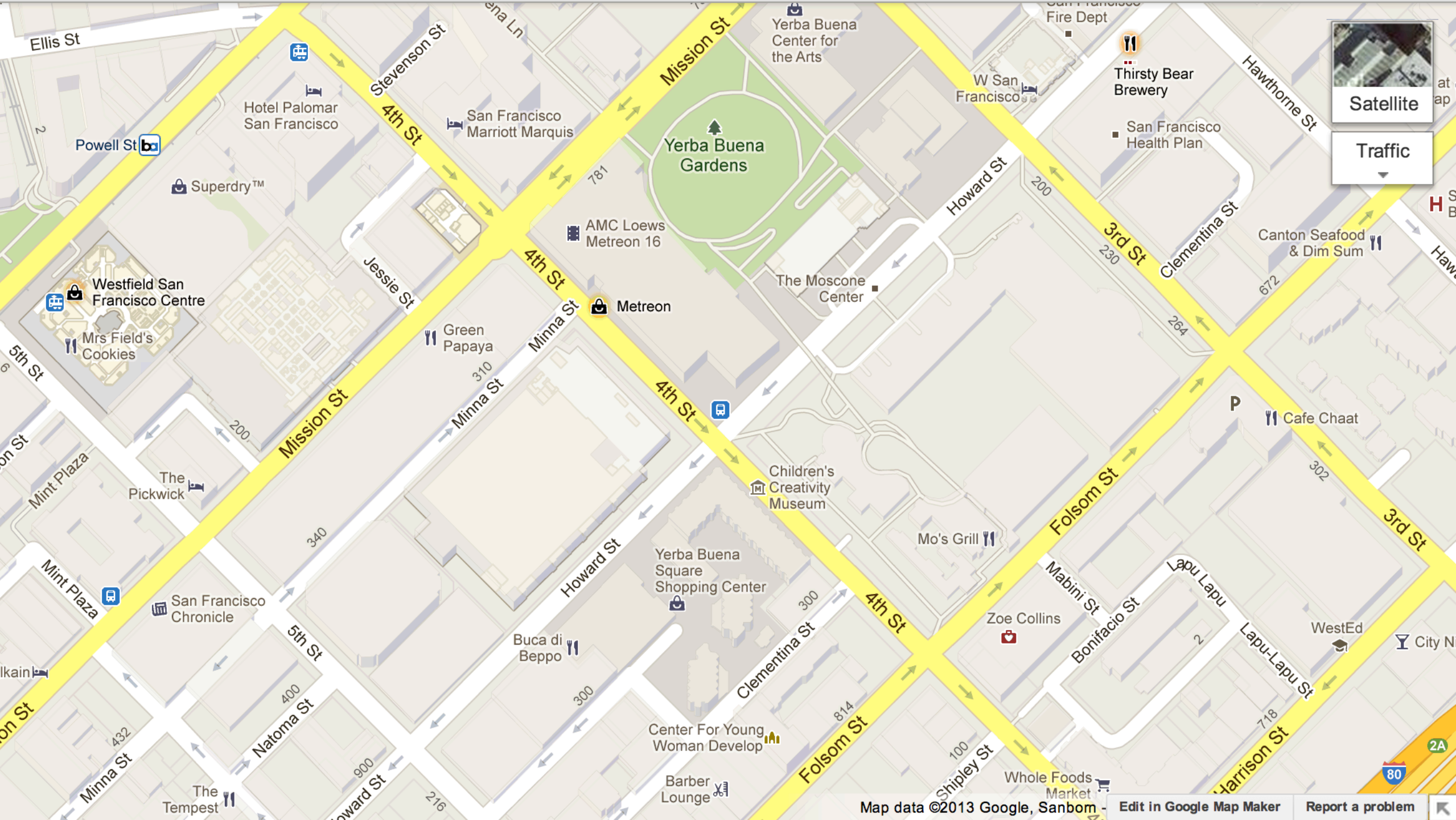
“Context isn’t important…”

“Context isn’t important, it’s critical.”

Hardware
OPEN
Reto

Hardware
OPEN
EXIT

ANDROID



Ellis St
Stevenson St
Mission St
Yerba Buena Center for the Arts
Fire Dept
Thirsty Bear Brewery
W San Francisco
Hawthorne St
Satellite
Traffic
Hotel Palomar San Francisco
San Francisco Marriott Marquis
4th St
Powell St
Yerba Buena Gardens
San Francisco Health Plan
Superdry™
Howard St
AMC Loews Metreon 16
3rd St
Clementina St
Canton Seafood & Dim Sum
Jessie St
Westfield San Francisco Centre
The Moscone Center
Metreon
Green Papaya
Minna St
Mrs Field's Cookies
5th St
Cafe Chaat
Mint Plaza
The Pickwick
Children's Creativity Museum
Folsom St
Mo's Grill
Yerba Buena Square Shopping Center
Mabini St
Lapu Lapu
San Francisco Chronicle
Zoe Collins
Bonifacio St
WestEd
Buca di Beppo
Lapu-Lapu St
Natoma St
Center For Young Woman Develop
Shipley St
Whole Foods Market
Harrison St
The Tempest
Barber Lounge
Map data ©2013 Google, Sanborn
Edit in Google Map Maker
Report a problem

Using Intents to monitor location changes

- Specify a Pending Intent to broadcast when the location changes.
- Location is stored as an extra: `KEY_LOCATION_CHANGED`.
- Useful for multiple Activities / Services that track location.

```
final int resultCode = 0;
final String locAction = "com.ioApp.LOCATION_UPDATE_
int flags = PendingIntent.FLAG_UPDATE_CURRENT;

Intent intent = new Intent(locAction);
PendingIntent pi = PendingIntent.getBroadcast(this,
  resultCode, intent, flags);

locationManager.requestLocationUpdates(provider, minT
```

Monitor inactive providers for a better option

```
locationManager.getBestProvider(criteria, false);

public void onProviderEnabled(String provider){
  // Switch providers!
}
```

Check the last known location

```
List<String> providers = lm.getProviders(criteria, false);
for (String provider: providers) {
 Location location = lm.getLastKnownLocation(provider);
 l
 ion.getAccuracy();
 on.getTime();
 his the best previously known location?
```

Use Interfaces for backwards compatibility

```
private static boolean newSensorAPIsSupported =
  Build.VERSION.SDK_INT >= Build.VERSION_CODES.CUPCAKE;

boolean gyroExists =
  getPackageManager().hasSystemFeature(
  PackageManager.FEATURE_SENSOR_GYROSCOPE);

IOrientationSensorListener myListener;
if (gyroExists)
  myListener = new GyroOrientationSen
else if (newSensorAPIsSupported)
  myListener = new AccOrientationSens
else
  myListener = new AccOldOrientationS

myListener.setOrientationChangeListen
```

Using Intents to passively detect location changes

- Your app updates even if it's not running.
- No incremental battery cost.
- Start a Service to refresh the data without updating a UI.

```
<receiver android:name=".locReceiver" android:enabled="true">
  <intent-filter>
    android:name="com.ioApp.LOCATION_UPDATE_RECEIVED"/>
  r>
```

- Receives location updates iff another application requests them.
- Requires `ACCESS_FINE_LOCATION` permission.
- `Location.getProvider` reveals the underlying Location Provider.

```
String passiveProvider = LocationManager.PASSIVE_PROVIDER;
locationManager.requestLocationUpdates(passiveProvider,
                                       minTime, minDistance,
                                       myLocationListener);
```

Use Intents to update location when your app isn't running

Using Intents to monitor location changes

- Specify a Pending Intent to broadcast when the location changes.
- Location is stored as an extra: `KEY_LOCATION_CHANGED`.
- Useful for multiple Activities / Services that track location.

```
BroadcastReceiver locReceiver = new BroadcastReceiver() {
  @Override
  public void onReceive(Context context, Intent intent) {
    String key = LocationManager.KEY_LOCATION_CHANGED;
    Location location = (Location)intent.getExtras().get(key);
     // [... Do something with the new location ...]
  }
};
IntentFilter locIntentFilter = new IntentFilter(locAction);
registerReceiver(locReceiver, locIntentFilter);
```

Using Intents to monitor location changes

- Specify a Pending Intent to broadcast when the location changes.
- Location is stored as an extra: KEY_LOCATION_CHANGED.
- Useful for multiple Activities / Se

```
final int resultCode = 0;
final String locAction = "com.io
int flags = PendingIntent.FLAG_U

Intent intent = new Intent(locAc
PendingIntent pi = PendingIntent
  resultCode, intent, flags);

locationManager.requestLocationU
```

Check the last known location

```
List<String> providers = lm.getProviders(criteria, false);
for (String provider: providers) {
  Location location = lm.getLa
  ion.getAccuracy
  on.get
```

```
nLocation(provider);

location?
```

Android Developers Blog

Developers

SEARCH

Search

ARCHIVE

- 2013 (9)
- 2012 (43)
- 2011 (68)
  - December (7)
  - November (7)
  - October (5)
  - September (5)
  - August (3)
  - July (7)
  - June (3)
    - A Deep Dive Into Location

23 JUNE 2011

A Deep Dive Into Location

[This post is by Reto Meier, Tech Lead for Android Developer Relations, who wrote the book on Android App development. — Tim Bray]

I'm a big fan of location-based apps, but not their seemingly-inevitable startup latency.

Whether it's finding a place to eat or searching for the nearest Boris Bike, I find the delay while waiting for the GPS to get a fix, and then for the results list to populate, to be interminable. Once I'm in a venue and ready to get some tips, check-in, or review the food, I'm frequently thwarted by a lack of data connection.

Rather than shaking my fist at the sky, I've written an open-source reference app that incorporates all of the tips, tricks, and cheats I know to reduce the time between opening an app and seeing an up-to-date list of nearby venues - as well as providing a reasonable level of offline support — all while keeping the impact on battery life to a minimum.

Show Me the Code

```
    }
};
IntentFilter locIntentFilter = new IntentFilter(locAction);
registerReceiver(locReceiver, locIntentFilter);
```

Use Interf

```
private stati
  Build.VERSI

boolean gyroE
  getPackageM
  PackageMana

IOrientationS
if (gyroExist
  myListener
else if (newS
  myListener
else
  myListener

myListener.se
```

ect location changes

g.

out updating a UI.

```
android:enabled="true">

CATION_UPDATE_RECEIVED"/>
```

dates iff another application requests them.

INE_LOCATION permission.

vider reveals the underlying Location Provider.

```
String passiveProvider = LocationManager.PASSIVE_PROVIDER;
locationManager.requestLocationUpdates(passiveProvider,
                                       minTime, minDistance,
                                       myLocationListener);
```

Use Intents to update location
app isn't running

en your

android-protips-location

Android Protips: A Deep Dive Into Location

Search projects

Project Home | Issues | **Source** | Administer

Checkout | **Browse** | Changes | Request code review

Source path: svn/

‹r4 **r5**

Directories

- svn
  - trunk
    - res
    - src
      - com
        - radioactiveyak
          - location_best_practices
            - UI
              - fragments
            - content_providers
            - receivers
            - services
            - utils
              - base

Filename	Size	Rev	Date	Author
FroyoLocationUpdateRequester.java	1.7 KB	r1	Jun 21, 2011	retome...@google.com
FroyoSharedPreferenceSaver.java	1.3 KB	r1	Jun 21, 2011	retome...@google.com
GingerbreadLastLocationFinder.java	6.0 KB	r3	Jun 22, 2011	retome...@google.com
GingerbreadLocationUpdateRequester.java	1.6 KB	r1	Jun 21, 2011	retome...@google.com
GingerbreadSharedPreferenceSaver.java	1.2 KB	r1	Jun 21, 2011	retome...@google.com
HoneycombStrictMode.java	1.3 KB	r1	Jun 21, 2011	retome...@google.com
LegacyLastLocationFinder.java	5.6 KB	r3	Jun 22, 2011	retome...@google.com
LegacyLocationUpdateRequester.java	2.7 KB	r1	Jun 21, 2011	retome...@google.com
LegacySharedPreferenceSaver.java	1.2 KB	r1	Jun 21, 2011	retome...@google.com
LegacyStrictMode.java	1.2 KB	r1	Jun 21, 2011	retome...@google.com
PlatformSpecificImplementationFactory.java	2.8 KB	r1	Jun 21, 2011	retome...@google.com

Create or upload a new file

```java
@Override
public void onCreate(Bundle savedInstanceState) {
  super.onCreate(savedInstanceState);

  // Inflate the layout
  setContentView(R.layout.main);

  // Get a handle to the Fragments
  placeListFragment = (PlaceListFragment)getSupportFragmentManager().f
  checkinFragment = (CheckinFragment)getSupportFragmentManager().findF

  // Get references to the managers
  packageManager = getPackageManager();
  notificationManager = (NotificationManager)getSystemService(Context.
  locationManager = (LocationManager)getSystemService(Context.LOCATION

  // Get a reference to the Shared Preferences and a Shared Preference
  prefs = getSharedPreferences(PlacesConstants.SHARED_PREFERENCE_FILE,
  prefsEditor = prefs.edit();

  // Instantiate a SharedPreferenceSaver class based on the available
  // This will be used to save shared preferences
  sharedPreferenceSaver = PlatformSpecificImplementationFactory.getSha

  // Save that we've been run once.
  prefsEditor.putBoolean(PlacesConstants.SP_KEY_RUN_ONCE, true);
  sharedPreferenceSaver.savePreferences(prefsEditor, false);

  // Specify the Criteria to use when requesting location updates whil
  criteria = new Criteria();
  if (PlacesConstants.USE_GPS_WHEN_ACTIVITY_VISIBLE)
    criteria.setAccuracy(Criteria.ACCURACY_FINE);
  else
    criteria.setPowerRequirement(Criteria.POWER_LOW);

  // Setup the location update Pending Intents
  Intent activeIntent = new Intent(this, LocationChangedReceiver.class
  locationListenerPendingIntent = PendingIntent.getBroadcast(this, 0,

  Intent passiveIntent = new Intent(this, PassiveLocationChangedReceiv
  locationListenerPassivePendingIntent = PendingIntent.getBroadcast(th

  // Instantiate a LastLocationFinder class.
  // This will be used to find the last known location when the applic
  lastLocationFinder = PlatformSpecificImplementationFactory.getLastLo
  lastLocationFinder.setChangedLocationListener(oneShotLastLocationUpd

  // Instantiate a Location Update Requester class based on the availa
  // This will be used to request location updates.
  locationUpdateRequester = PlatformSpecificImplementationFactory.getL
      locationManager.requestLocationUpdates(prov
  }

  /**
   * {@inheritDoc}
   */
  @Override
  public void requestPassiveLocationUpdates(long
    // Pre-Froyo there was no Passive Location Pr
    // that will trigger once the minimum time be
    // than simple passive alarms, however the Re
    // distance before initiating a background ne
    alarmManager.setInexactRepeating(AlarmManager

  }
}
```

```java
listener that receives updates from the {@link LastLocationFinder}.
d where the last known location is outside the bounds of our maximum
ency
```

```java
/**
 * Returns the most accurate and timely previously detected location.
 * Where the last result is beyond the specified maximum distance or
 * latency a one-off location update is returned via the {@link LocationListener}
 * specified in {@link setChangedLocationListener}.
 * @param minDistance Minimum distance before we require a location update.
 * @param minTime Minimum time required between location updates.
 * @return The most accurate and / or timely previously detected location.
 */
public Location getLastBestLocation(int minDistance, long minTime) {
  Location bestResult = null;
  float bestAccuracy = Float.MAX_VALUE;
  long bestTime = Long.MIN_VALUE;

  // Iterate through all the providers on the system, keeping
  // note of the most accurate result within the acceptable time limit.
  // If no result is found within maxTime, return the newest Location.
  List<String> matchingProviders = locationManager.getAllProviders();
  for (String provider: matchingProviders) {
    Location location = locationManager.getLastKnownLocation(provider);
    if (location != null) {
      float accuracy = location.getAccuracy();
      long time = location.getTime();

      if ((time > minTime && accuracy < bestAccuracy)) {
        bestResult = location;
        bestAccuracy = accuracy;
        bestTime = time;
      }
      else if (time < minTime && bestAccuracy == Float.MAX_VALUE && time > bestTime) {
        bestResult = location;
        bestTime = time;
      }
    }
  }

  // If the best result is beyond the allowed time limit, or the accuracy of the
  // best result is wider than the acceptable maximum distance, request a single update.
  // This check simply implements the same conditions we set when requesting regular
  // location updates every [minTime] and [minDistance].
  if (locationListener != null && (bestTime < minTime || bestAccuracy > minDistance)) {
    IntentFilter locIntentFilter = new IntentFilter(SINGLE_LOCATION_UPDATE_ACTION);
    context.registerReceiver(singleUpdateReceiver, locIntentFilter);
    locationManager.requestSingleUpdate(criteria, singleUpatePI);
  }

  return bestResult;
}
```

```java
ner = new LocationListener() {

ationChangedReceiver"/>
siveLocationChangedReceiver"/>
nectivityChangedReceiver" enabled="false">

conn.CONNECTIVITY_CHANGE" />
iveyak.places.retry_queued_checkins" />

e) {

upportFragmentManager().findFragmentById(R.id.list_fragment);
rtFragmentManager().findFragmentById(R.id.checkin_fragment);

getSystemService(Context.NOTIFICATION_SERVICE);
mService(Context.LOCATION_SERVICE);

s and a Shared Preference Editor.
s.SHARED_PREFERENCE_FILE, Context.MODE_PRIVATE);

s based on the available platform version.
nces
lementationFactory.getSharedPreferenceSaver(this);

EY_RUN_ONCE, true);
Editor, false);

ing location updates while the application is Active

ISIBLE)
NE);

ER_LOW);

s
tionChangedReceiver.class);
nt.getBroadcast(this, 0, activeIntent, PendingIntent.FLAG_UPDATE_CURRENT);

siveLocationChangedReceiver.class);
ingIntent.getBroadcast(this, 0, passiveIntent, PendingIntent.FLAG_UPDATE_CURRENT);

 location when the application starts.
entationFactory.getLastLocationFinder(this);
er(oneShotLastLocationUpdateListener);

// Instantiate a Location Update Requester class based on the available platform version.
// This will be used to request location updates.
locationUpdateRequester = PlatformSpecificImplementationFactory.getLocationUpdateRequester(locationManager);
```

```java
};
```

Connect to the Location Services Client

MyActivity

```
private void connectLBS() {
  int gpsExists = GooglePlayServicesUtil.isGooglePlayServicesAvailable(this);
  if (gpsExists == ConnectionResult.SUCCESS) {
    mLocationClient = new LocationClient(this, this, this);
    mlocationClient.connect();
  }
}

@Override
public void onConnected(Bundle connectionHint) {
  requestUpdates(mlocationClient);
}
```

Location Update Requests

MyActivity

```
LocationRequest request = LocationRequest.create();
request.setInterval(minTime);
request.setPriority(lowPowerMoreImportantThanAccuracy ?
  LocationRequest.PRIORITY_BALANCED_POWER_ACCURACY :
  LocationRequest.PRIORITY_HIGH_ACCURACY);

mlocationClient.requestLocationUpdates(request, new LocationListener() {
  @Override
  public void onLocationChanged(Location location) {
    updateLocation(location);
  }
});
```

Geofencing

MyActivity

```
List<Geofence> fenceList = new ArrayList<Geofence>();

// TODO Repeat for all Geofences
Geofence geofence = new Geofence.Builder()
  .setRequestId(myKey)
  .setTransitionTypes(Geofence.GEOFENCE_TRANSITION_ENTER |
                      Geofence.GEOFENCE_TRANSITION_EXIT)
  .setCircularRegion(latitude, longitude, GEOFENCE_RADIUS)
  .setExpirationDuration(Geofence.NEVER_EXPIRE)
  .build();

fenceList.add(geofence);

mLocationClient.addGeofences(fenceList, pendingIntent, addGeofencesResultListener);
```

Activity Recognition

MyActivity

```
Intent intent = new Intent(this, ActivityRecognitionIntentService.class);
intent.setAction(MyActivity.ACTION_STRING);

PendingIntent pi =
  PendingIntent.getService(this, 0, intent,
                           PendingIntent.FLAG_UPDATE_CURRENT);

mActivityRecognitionClient.requestActivityUpdates(interval, pi);
```

Activity Recognition

ActivityRecognitionIntentService

```
@Override
protected void onHandleIntent(Intent intent) {
  if (intent.getAction() == MyActivity.ACTION_STRING) {
    if (ActivityRecognitionResult.hasResult(intent)) {
      ActivityRecognitionResult result = ActivityRecognitionResult.extractResult(intent);
      DetectedActivity detectedActivity = result.getMostProbableActivity();
      int activityType = detectedActivity.getType();
      if (activityType == DetectedActivity.STILL)
        setUpdateSpeed(PAUSED);
      else if (activityType == DetectedActivity.IN_VEHICLE)
        setUpdateSpeed(FASTER);
      else
        setUpdateSpeed(REGULAR);
    }
  }
}
```

Activity Detection

ActivityRecognitionIntentService

```
@Override
protected void onHandleIntent(Intent intent) {
  if (intent.getAction() == MyActivity.ACTION_STRING) {
    if (ActivityRecognitionResult.hasResult(intent)) {
      ActivityRecognitionResult result = ActivityRecognitionResult.extractResult(intent);
      DetectedActivity detectedActivity = result.getMostProbableActivity();
      int activityType = detectedActivity.getType();
      if (activityType == DetectedActivity.STILL)
        setUpdateSpeed(FASTER);
      else if (activityType == DetectedActivity.ON_BICYCLE)
        setUpdateSpeed(PAUSED);
      else
        setUpdateSpeed(REGULAR);
    }
  }
}
```

What are my friends doing?

MyActivity

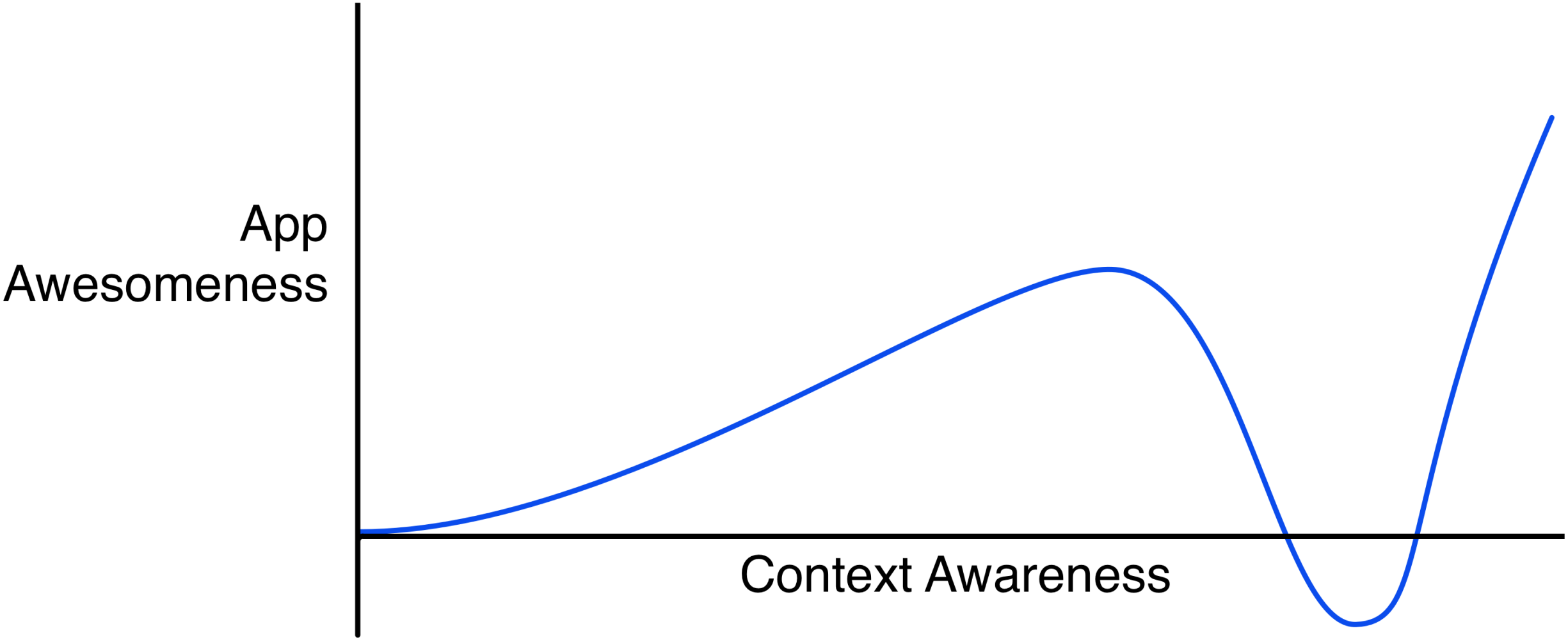
```
@Override
public void onConnected() {
  mPlusClient.loadPeople(this, Person.Collection.VISIBLE);
}
```

What are my friends doing?

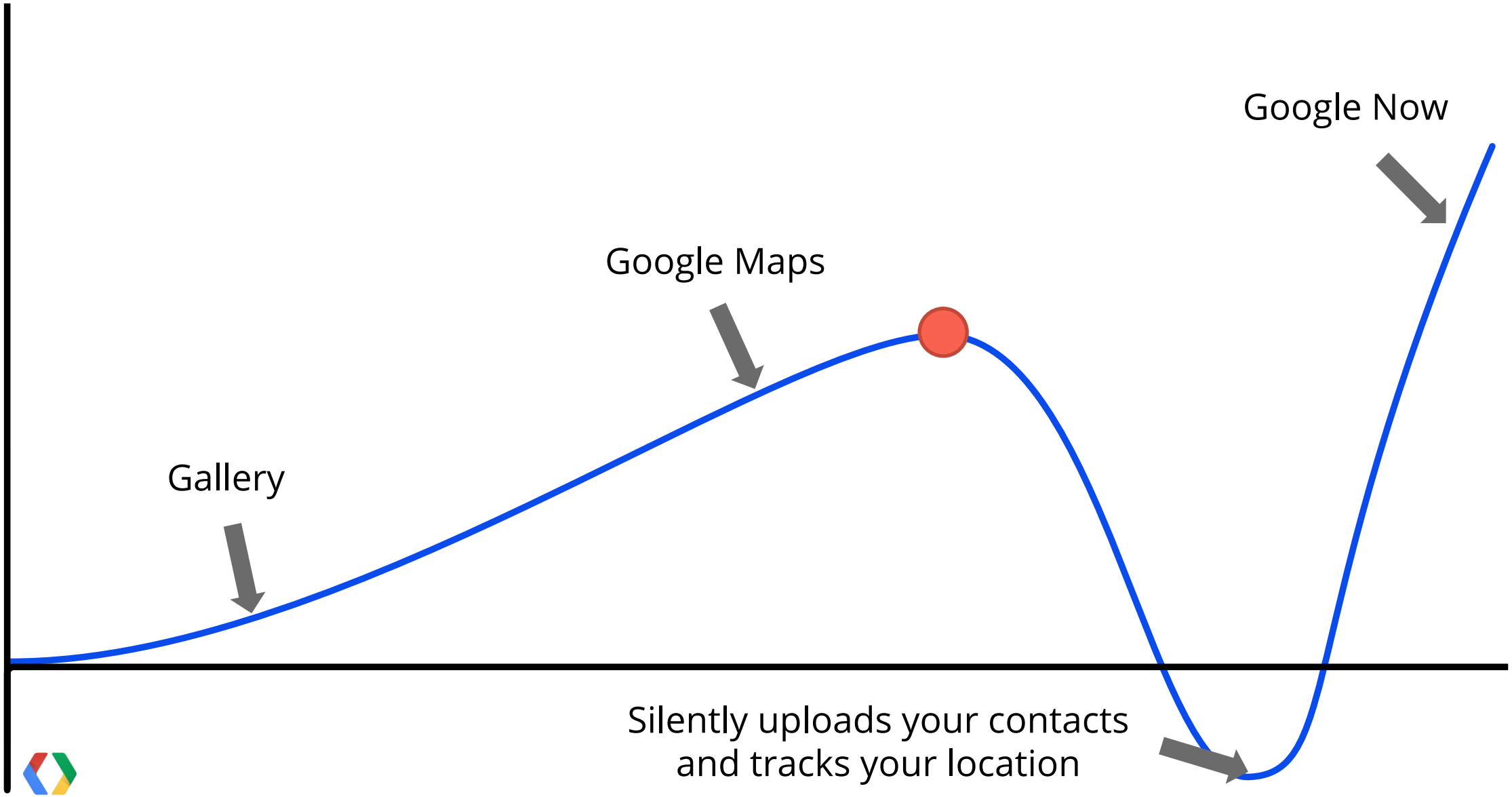
MyActivity

```
@Override
public void onPeopleLoaded(ConnectionResult status,
                           PersonBuffer personBuffer, String nextPageToken) {
  if (status.getErrorCode() == ConnectionResult.SUCCESS) {
    try {
      // TODO Things with personBuffer
    } finally {
      personBuffer.close();
    }
  } else
    Log.e(TAG, "Error listing people: " + status.getErrorCode());
}
```

The Uncanny App Valley

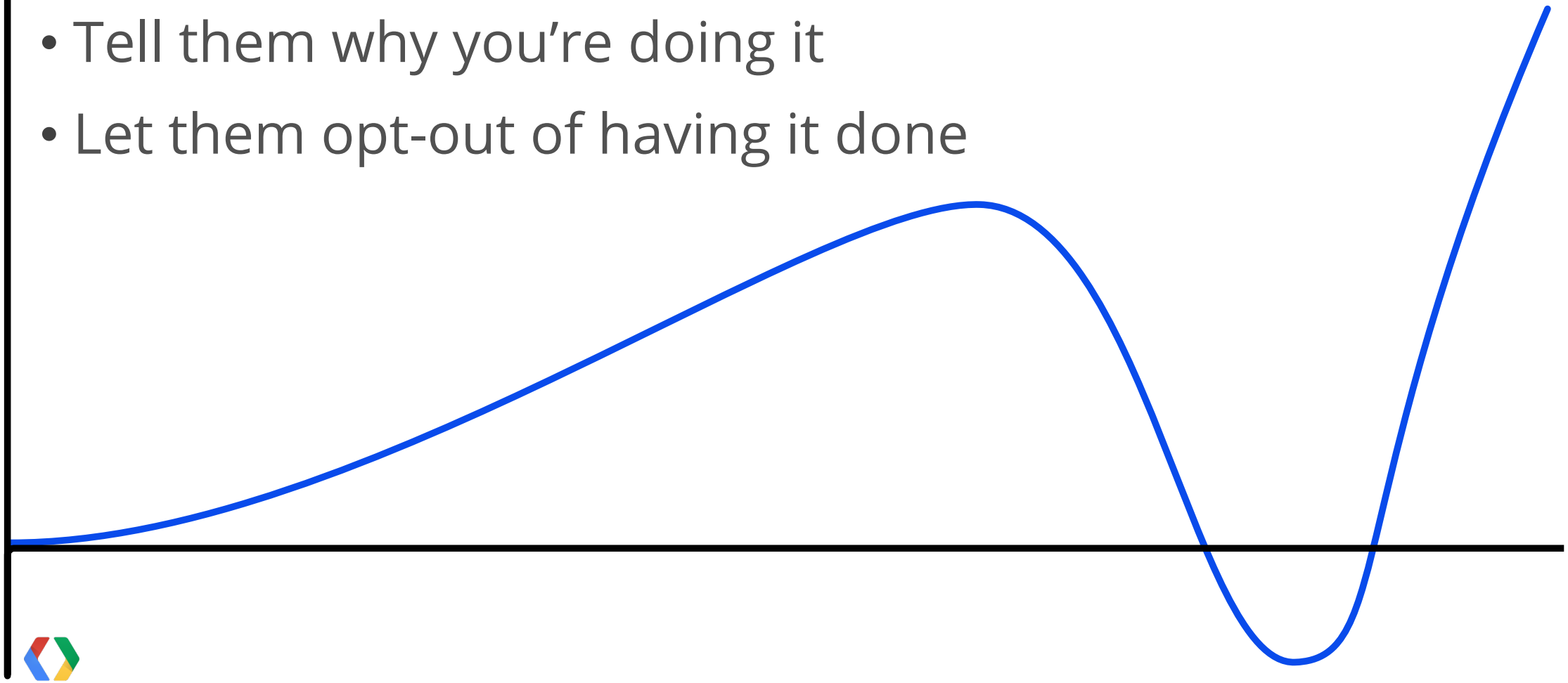


The Uncanny App Valley



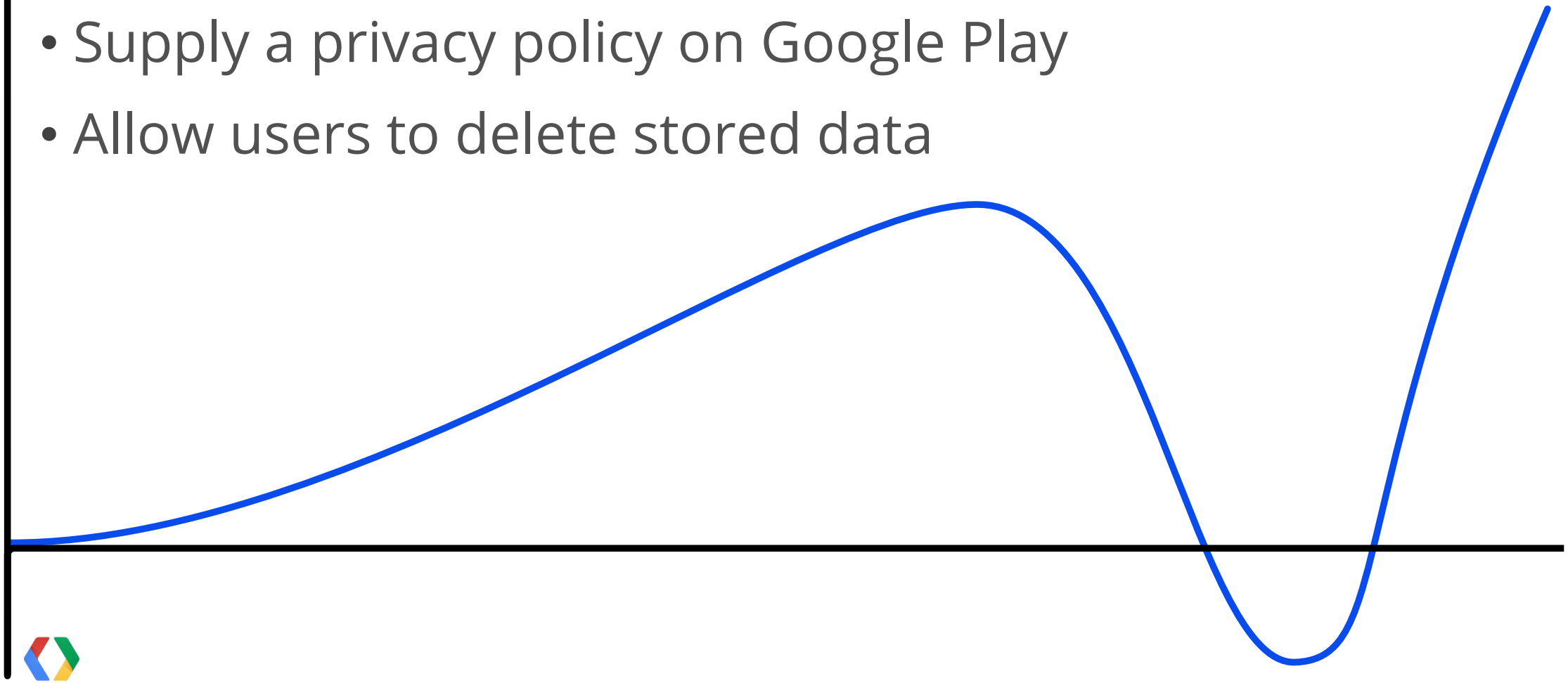
Avoiding the Uncanny App Valley

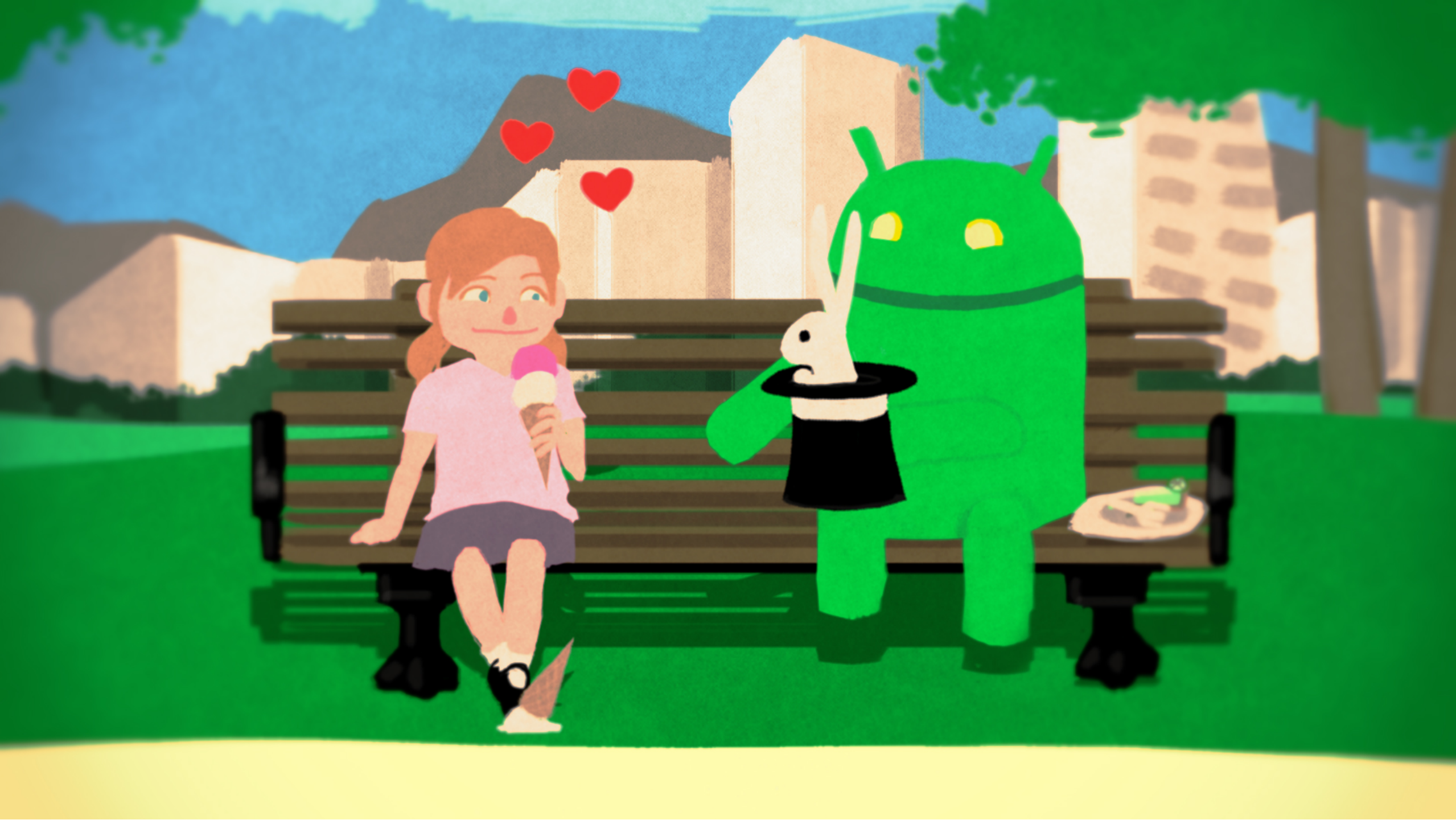
- Tell them what you're doing
- Tell them why you're doing it
- Let them opt-out of having it done



Avoiding the Uncanny App Valley

- Don’t transmit or store contact details or location
- Supply a privacy policy on Google Play
- Allow users to delete stored data





“Introducing visceral elements into an app . . will make it speak to the subconscious.”

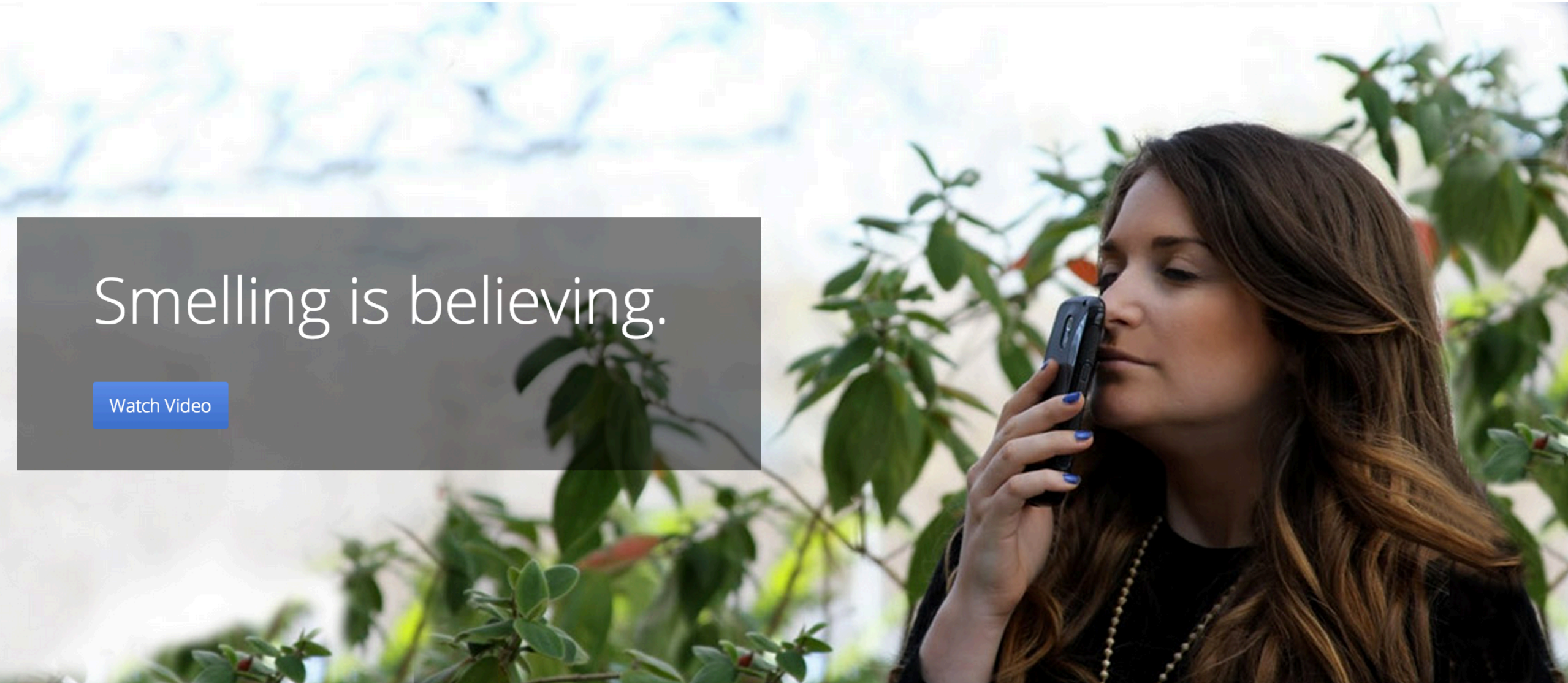
Rob Foster

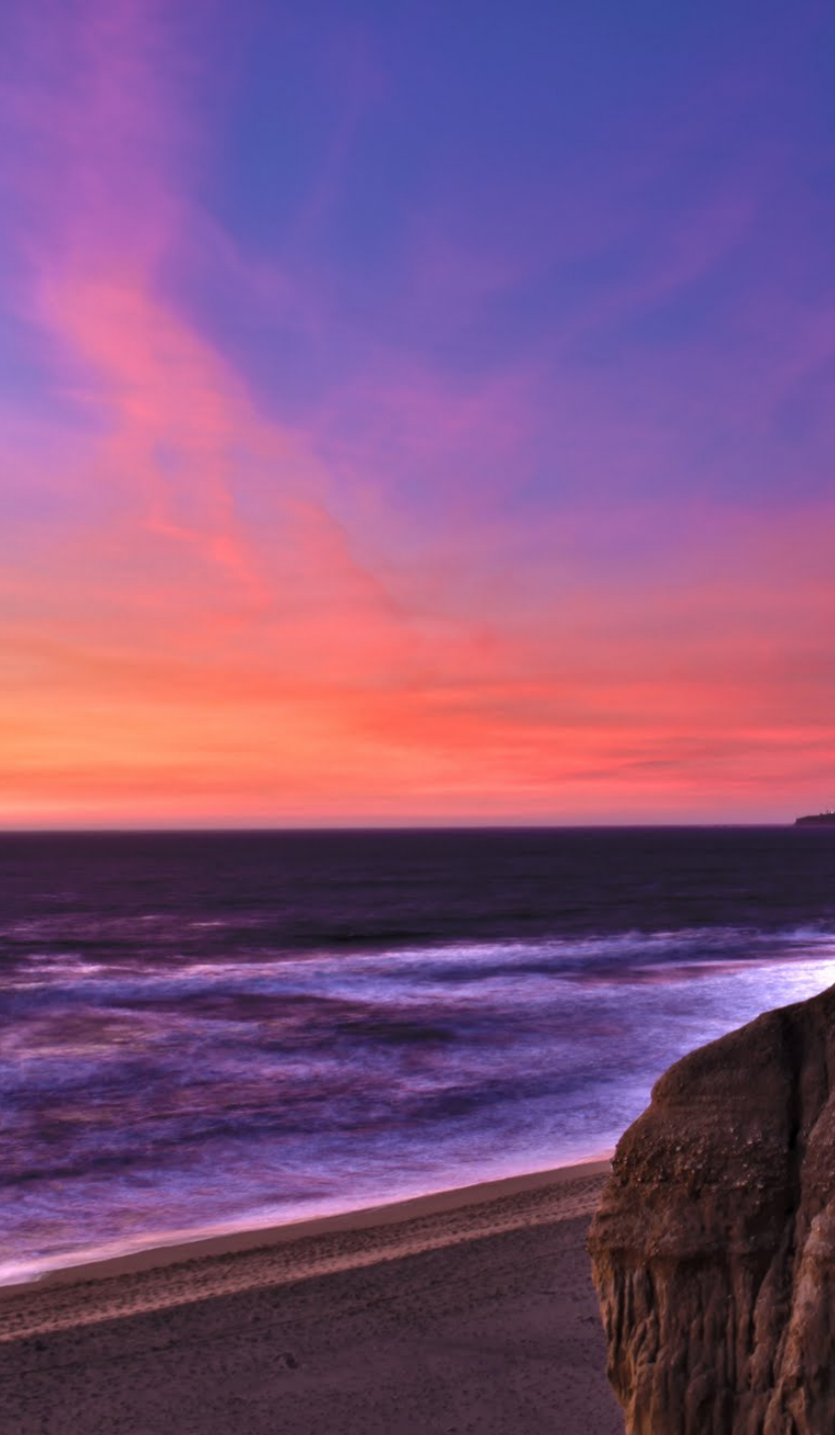
Scent has huge muscles and a giant brain

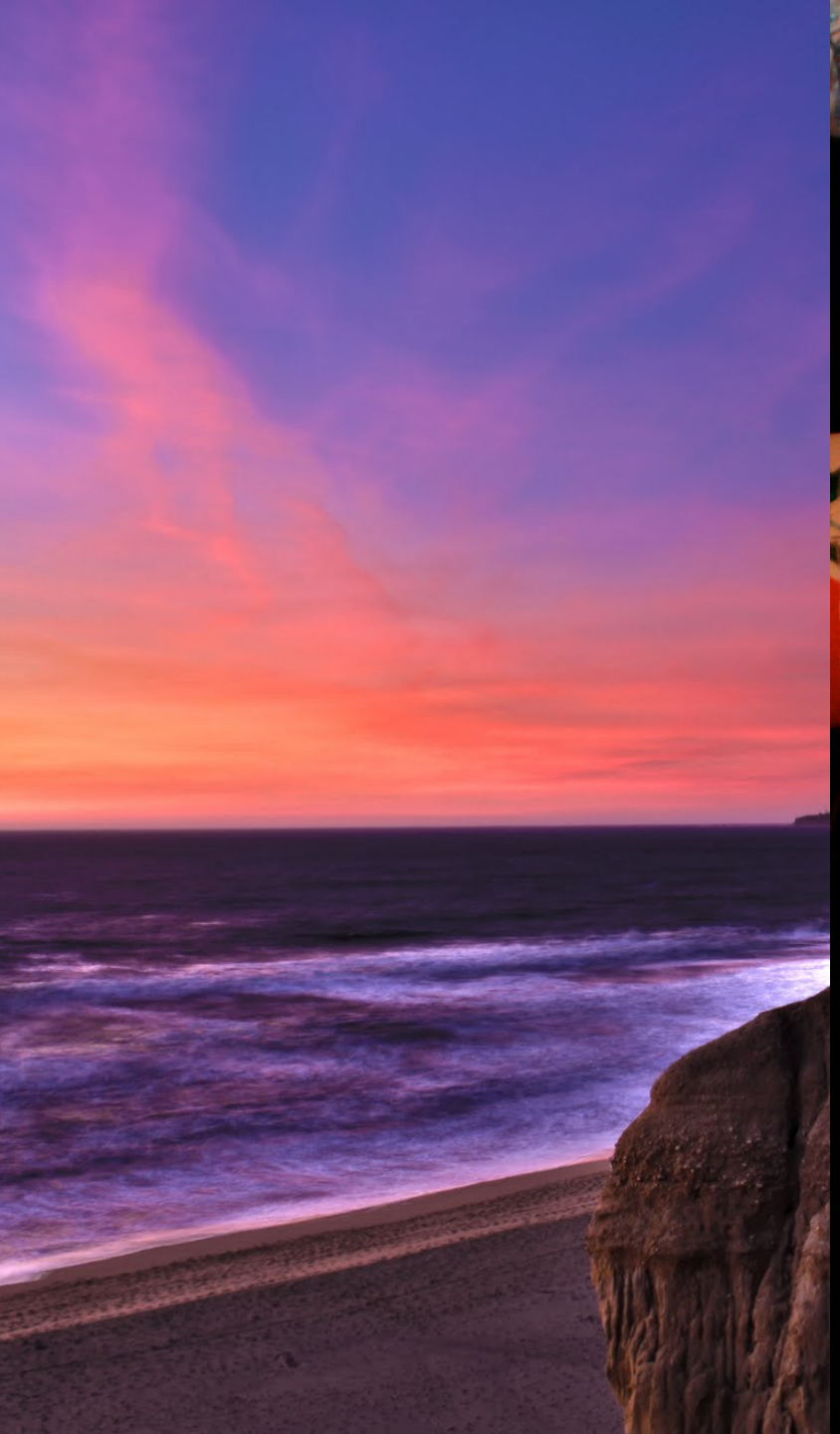
Let us show you how to take advantage of the most powerful sense available

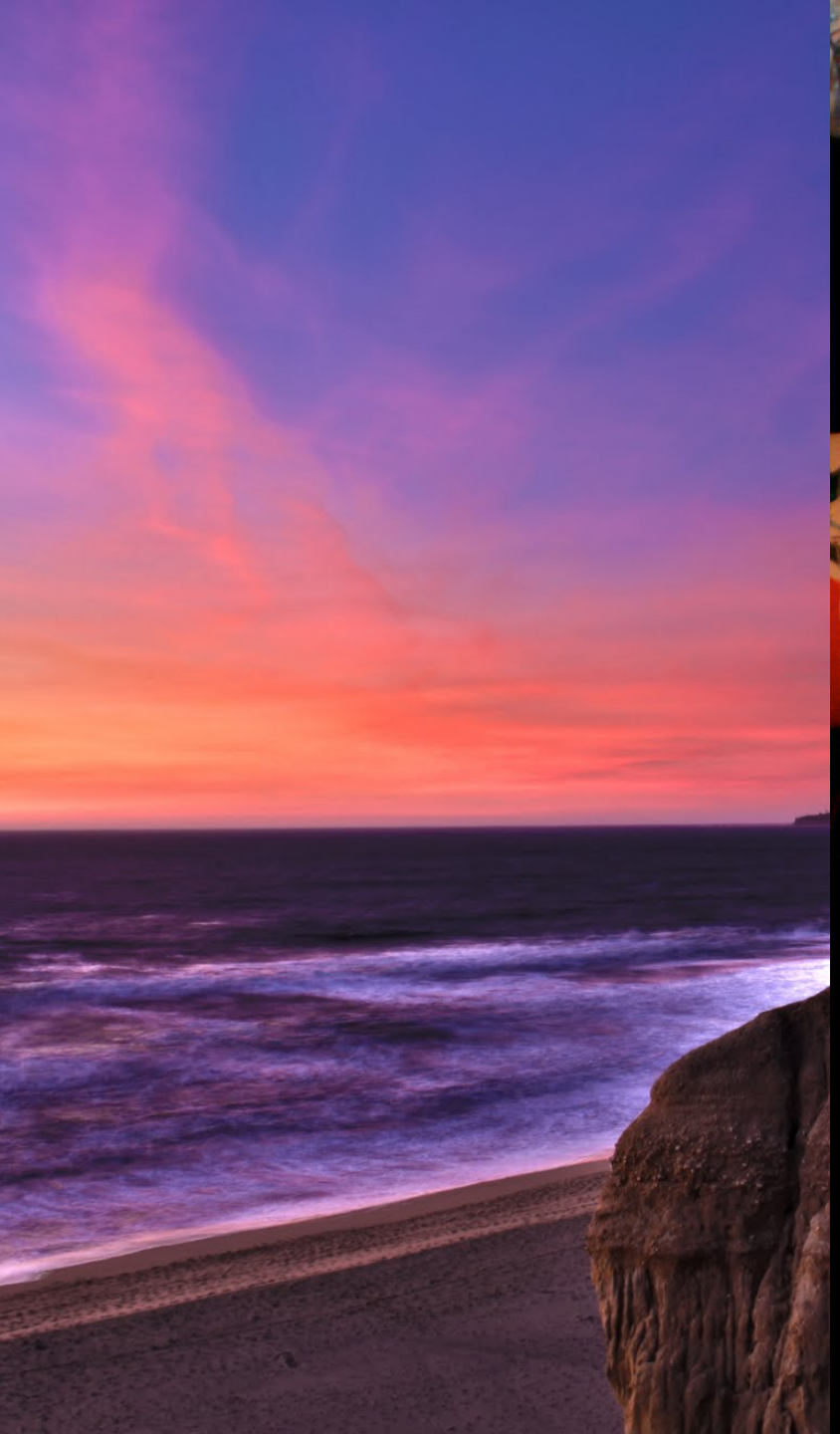
Now make it work for you

Brand identity is more critical today than ever before, as more and more businesses and products compete for consumer attention across an ever-increasing variety of channels. Our senses play a vital and complex role in forming our thoughts, impressions and behaviors. By targeting the senses, brands establish a stronger and


Smelling is believing.
Watch Video







Developers | Design | Develop | Distribute

- Get Started
  - Creative Vision
  - Design Principles
  - UI Overview
- Style
- Patterns
- Building Blocks
- Downloads
- Videos

Welcome to **Android Design**, your place for learning how to design exceptional Android apps.

Creative Vision >

http://d.android.com/design

Text to Speech (TTS)

MyActivty

```
private void initTextToSpeech() {
  Intent intent = new Intent(Engine.ACTION_CHECK_TTS_DATA);
  startActivityForResult(intent, TTS_DATA_CHECK);
}
```

Text to Speech (TTS)

MyActivty

```
protected void onActivityResult(int request, int result, Intent data) {
  if (request == TTS_DATA_CHECK &&
      result == Engine.CHECK_VOICE_DATA_PASS) {
    tts = new TextToSpeech(this, new OnInitListener() {
      public void onInit(int status) {
        if (status == TextToSpeech.SUCCESS)
          ttsIsInit = true;
      }
    });
  } else
    startActivity(new Intent(Engine.ACTION_INSTALL_TTS_DATA);
}
```

Text to Speech (TTS)

MyActivty

```
private void say(String text) {
  if (tts != null && ttisIsInit)
    tts.speak(text, TextToSpeech.QUEUE_ADD, null);
}
```

Speech Recognition

MyActivty

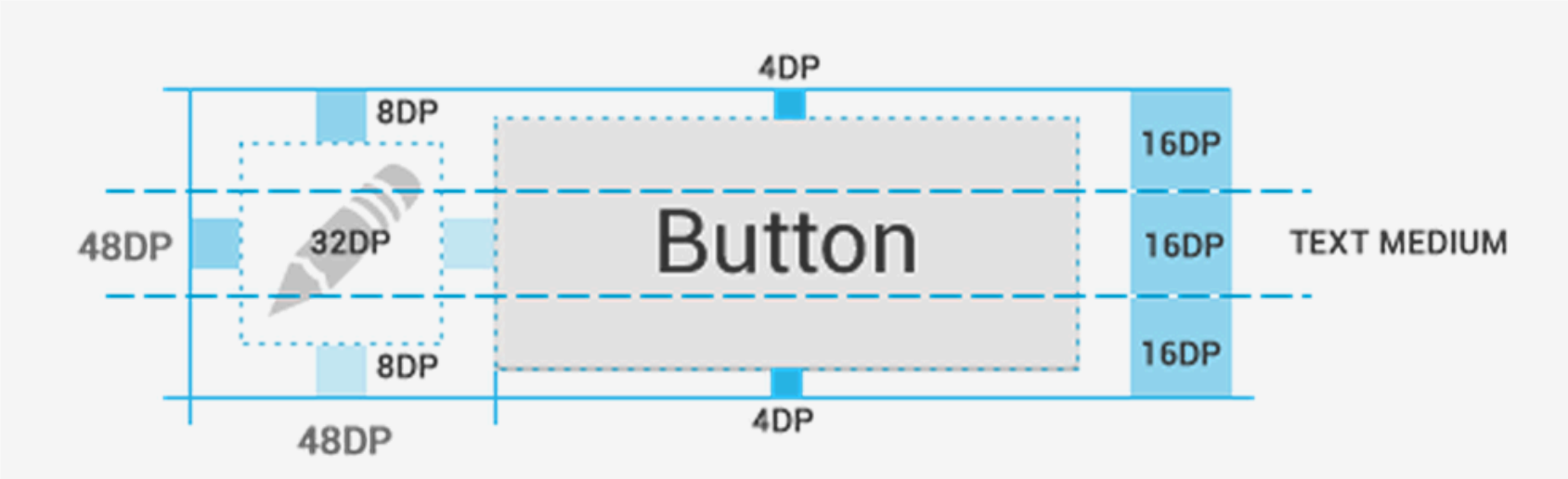
```
private void requestVoiceInput() {
  Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
  intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
                  RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
  intent.putExtra(RecognizerIntent.EXTRA_PROMPT,
                  getString(R.String.voice_input_prompt);
  intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE,
                  Locale.ENGLISH);
  startActivityForResult(intent, VOICE_RECOGNITION);
}
```

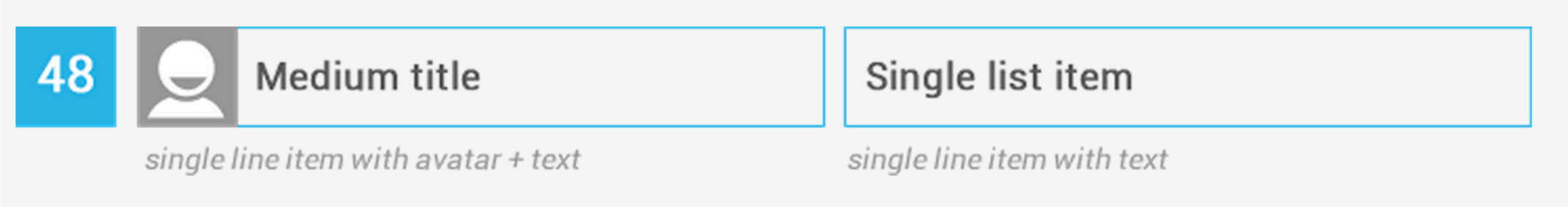
Speech Recognition

MyActivty

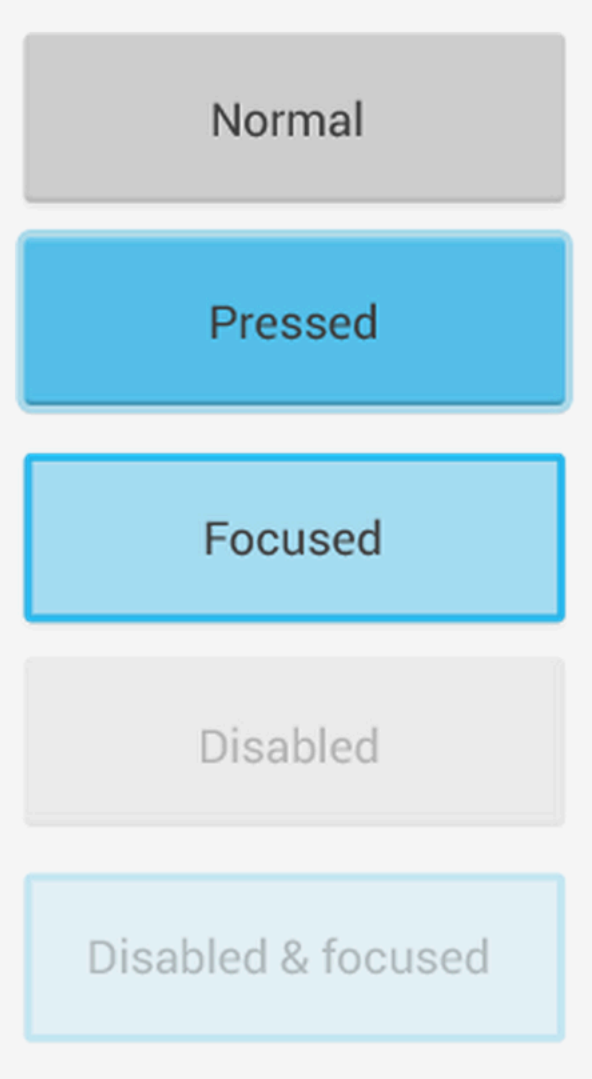
```
@Override
protected void onActivityResult(int request, int result, Intent data) {
  if (request == VOICE_RECOGNITION && result == RESULT_OK) {
    ArrayList<String> results =
      data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);

    String mostLikelyResult = results[0];
    useSpeechInput(mostLikelyResult);
  }
}
```


4DP
8DP
16DP
48DP
32DP
Button
16DP
TEXT MEDIUM
8DP
16DP
48DP
4DP


48
Medium title
Single list item
single line item with avatar + text
single line item with text

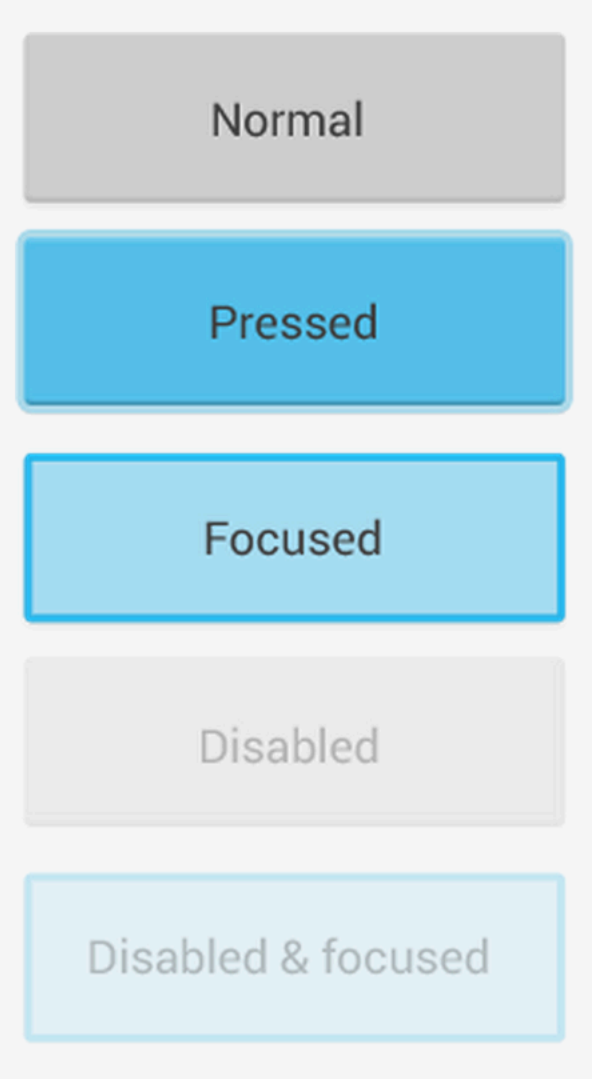
Touch Feedback



MyBGDrawable.xml

```
<selector xlmns:android="...">
  <item android:state_pressed="true"
        android:drawable="@drawable/bg_pressed"/>
  <item android:state_enabled="false"
        android:state_focussed="true"
        android:drawable="@drawable/bg_disabled_foc"/>
  <item android:state_focussed="true"
        android:drawable="@drawable/bg_focussed"/>
  <item android:drawable="@drawable/bg_normal"/>
</selector>
```

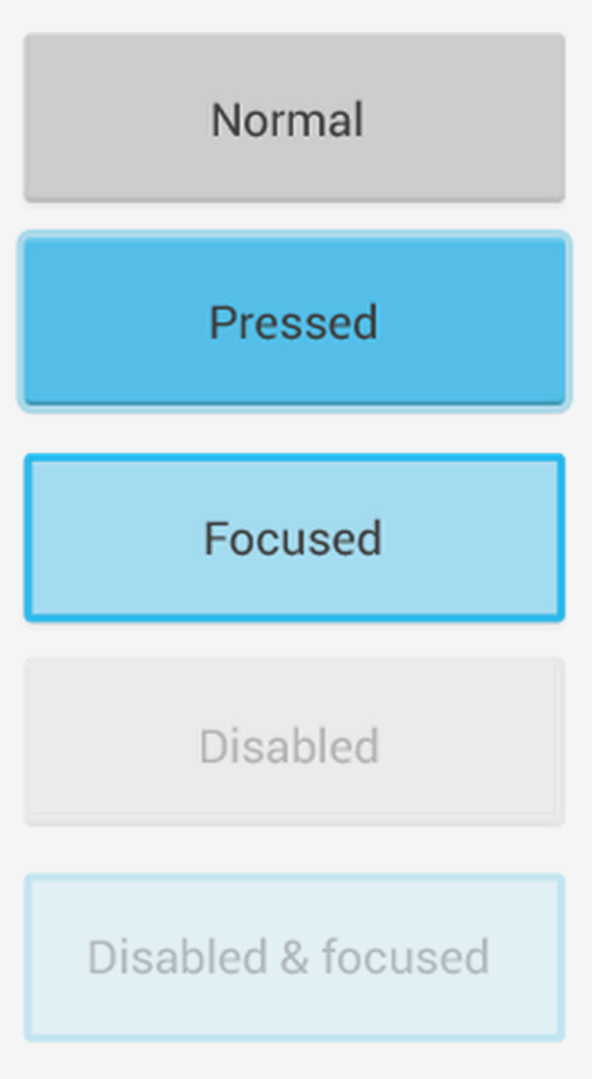
Touch Feedback



MyViewLayout.xml

```
...
android:background="?android:selectableItemBackground"
...
```

Touch Feedback



MyViewLayout.xml

```
...
android:foreground="?android:selectableItemBackground"
...
```

Simple Accessibility Support

MyViewLayout.xml

```
<Button
  ...
  android:contentDescription="@string/my_button_description"
/>
```

Custom Control Accessibility Support

MyCompass

```
public void setHeading(float heading) {
  mHeading = heading;
  sendAccessibilityEvent(AccessibilityEvent.TYPE_VIEW_TEXT_CHANGED);
}

@Override
public boolean dispatchPopulateAccessibilityEvent(final AccessibilityEvent e) {
  super.dispatchPopulateAccessibilityEvent(e);
  String heading = String.valueOf(mheading);
  if (heading.length() > AccessibilityEvent.MAX_TEXT_LENGTH))
    heading = heading.subString(0, AccessibilityEvent.MAX_TEXT_LENGTH);
  event.getText().add("Heading is " + heading + " degrees");
  return true;
}
```

Getting Started

Building Apps with Multimedia

Building Apps with Graphics & Animation

Building Apps with Connectivity & the Cloud

Building Apps with User Info & Location

Best Practices for User Experience & UI

Best Practices for User Input

Using Touch Gestures

- Detecting Common Gestures
- Tracking Movement
- Animating a Scroll Gesture
- Handling Multi-Touch Gestures
- Dragging and Scaling

Using Touch Gestures

This class describes how to write apps that allow users to interact with an app via touch gestures. Android provides a variety of APIs to help you create and detect gestures.

Although your app should not depend on touch gestures for basic behaviors (since the gestures may not be available to all users in all contexts), adding touch-based interaction to your app can greatly increase its usefulness and appeal.

To provide users with a consistent, intuitive experience, your app should follow the accepted Android conventions for touch gestures. The Gestures design guide shows you how to use common gestures in Android apps. Also see the Design Guide for Touch Feedback.

Lessons

Detecting Common Gestures

Learn how to detect basic touch gestures such as scrolling, flinging, and double-tapping, using `GestureDetector`.

Tracking Movement

Learn how to track movement.

GET STARTED >

DEPENDENCIES AND PREREQUISITES

- Android 1.6 (API Level 4) or higher

YOU SHOULD ALSO READ

- Input Events API Guide
- Sensors Overview
- Making the View Interactive
- Design Guide for Gestures
- Design Guide for Touch Feedback

TRY IT OUT

Download the sample

InteractiveChart.zip

Jazz Hands

MyView

```
@Override
public boolean onTouchEvent(MotionEvent event) {
  int action = event.getAction();

  if (event.getPointerCount() > 1) {
    int actionPointerId = action & MotionEvent.ACTION_POINTER_ID_MASK;
    int actionEvent = action & MotionEvent.ACTION_MASK;

    int pointerIndex = event.findPointerIndex(actionPointerId);
    int xPos = (int)event.getX(pointerIndex);
    int yPos = (int)event.getY(pointerIndex);
    // TODO Magic.
  }
}
```

Professional
Android 4
Application Development
Android 4
ANDROID
SWERVE

Google play | Developer Console

Android Market Test android.market.test@gmail.com Sign out

ANGRY CARROTS – com.happycompany.angrycarrots

- Statistics
- Ratings & Reviews
- Rankings
- Crashes & ANRs
- **APK**
- Store Listing
- Pricing and Distribution
- In-app Products
- Services & APIs

APK Published

Switch to simple mode

PROD ✓	BETA TESTING –	ALPHA TESTING ✓
Versions 1000, 1010	Set-up Beta testing for your app	Versions 2000, 2010

ALPHA CONFIGURATION Upload new APK

Supported devices
Currently live See list
4079

Excluded devices
Manage excluded devices
2

VERSION		UPLOADED ON	STATUS	ACTIONS
2010 (2.0)	**Tablet version 2**	Mar, 10 2012	✓ **in Alpha**	Move...
2000 (2.0)	**Phone version 2**	Mar, 10 2012	✓ **in Alpha**	Move...

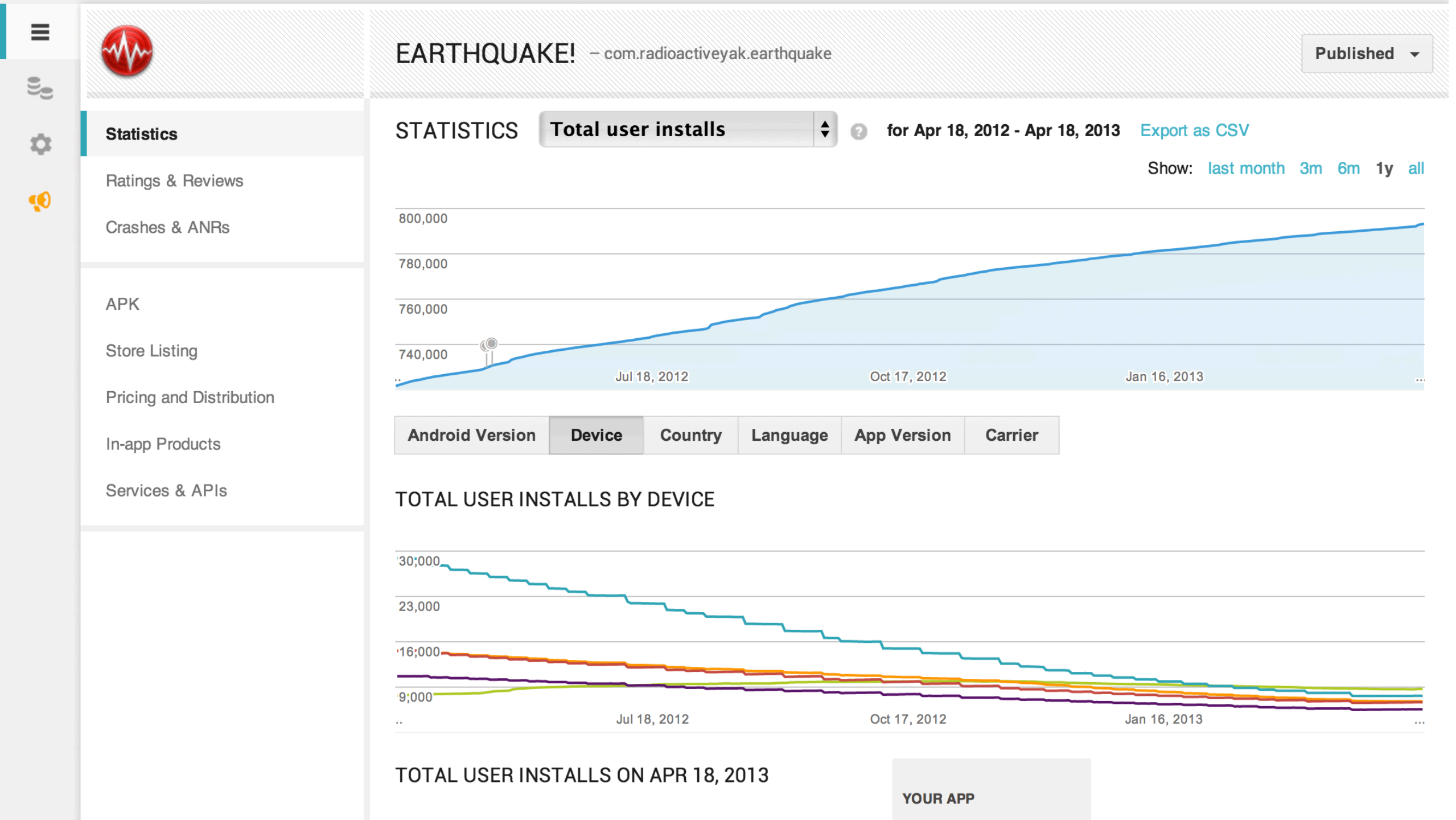
Move all Alpha APKs...

OTHER APKs Show

Not All Users Are Created Equal

Mobile Device Info	Total Visits
Samsung GT-I9300 Galaxy S3	825,780
Samsung GT-I9100 Galaxy S II	537,478
Motorola MOTXT912B Droid Razr 4G	486,950
Motorola Xoom	328,762
Motorola DroidX	229,418
Samsung GT-I9000 Galaxy S	210,994
Samsung GT-I9300 Galaxy SIII	185,329
Google Nexus 7	132,242
Asus TF101 Eee Pad Transformer TF101	125,363
Samsung SPH-D710 Galaxy SII Epic 4G Touch	123,806

Mobile Device Info	Avg. Visit Duration
Motorola Xoom	1:21:05
Google Nexus 7	1:05:53
Asus TF101 Eee Pad Transformer TF101	1:05:42
Samsung GT-I9300 Galaxy S3	53:09
Samsung GT-I9100 Galaxy S II	49:01
Samsung GT-I9300 Galaxy SIII	47:29
Motorola DroidX	42:17
Motorola MOTXT912B Droid Razr 4G	37:33
Samsung SPH-D710 Galaxy SII Epic 4G Touch	34:54
Samsung GT-I9000 Galaxy S	33:05


EARTHQUAKE! – com.radioactiveyak.earthquake
Published
Statistics
Ratings & Reviews
Crashes & ANRs
APK
Store Listing
Pricing and Distribution
In-app Products
Services & APIs
STATISTICS
Total user installs
for Apr 18, 2012 - Apr 18, 2013
Export as CSV
Show: last month 3m 6m 1y all
800,000
780,000
760,000
740,000
Jul 18, 2012
Oct 17, 2012
Jan 16, 2013
Android Version
Device
Country
Language
App Version
Carrier
TOTAL USER INSTALLS BY DEVICE
30,000
23,000
16,000
9,000
Jul 18, 2012
Oct 17, 2012
Jan 16, 2013
TOTAL USER INSTALLS ON APR 18, 2013
YOUR APP

Analytics Code

MyView

```
@Override
public void onClick(View v) {
  myTracker.sendEvent("ui_action", "button_press", "play_button", opt_value);
  ...
}
```

GOOGLE
QUEST
CHEAT
CODES

Amongst the Reasons to Transfer Data

Client Updates

Amongst the Reasons to Transfer Data

Client Updates

Server Updates

Amongst the Reasons to Transfer Data

Client Updates

Server Updates

On-Demand Downloads

Amongst the Reasons to Transfer Data

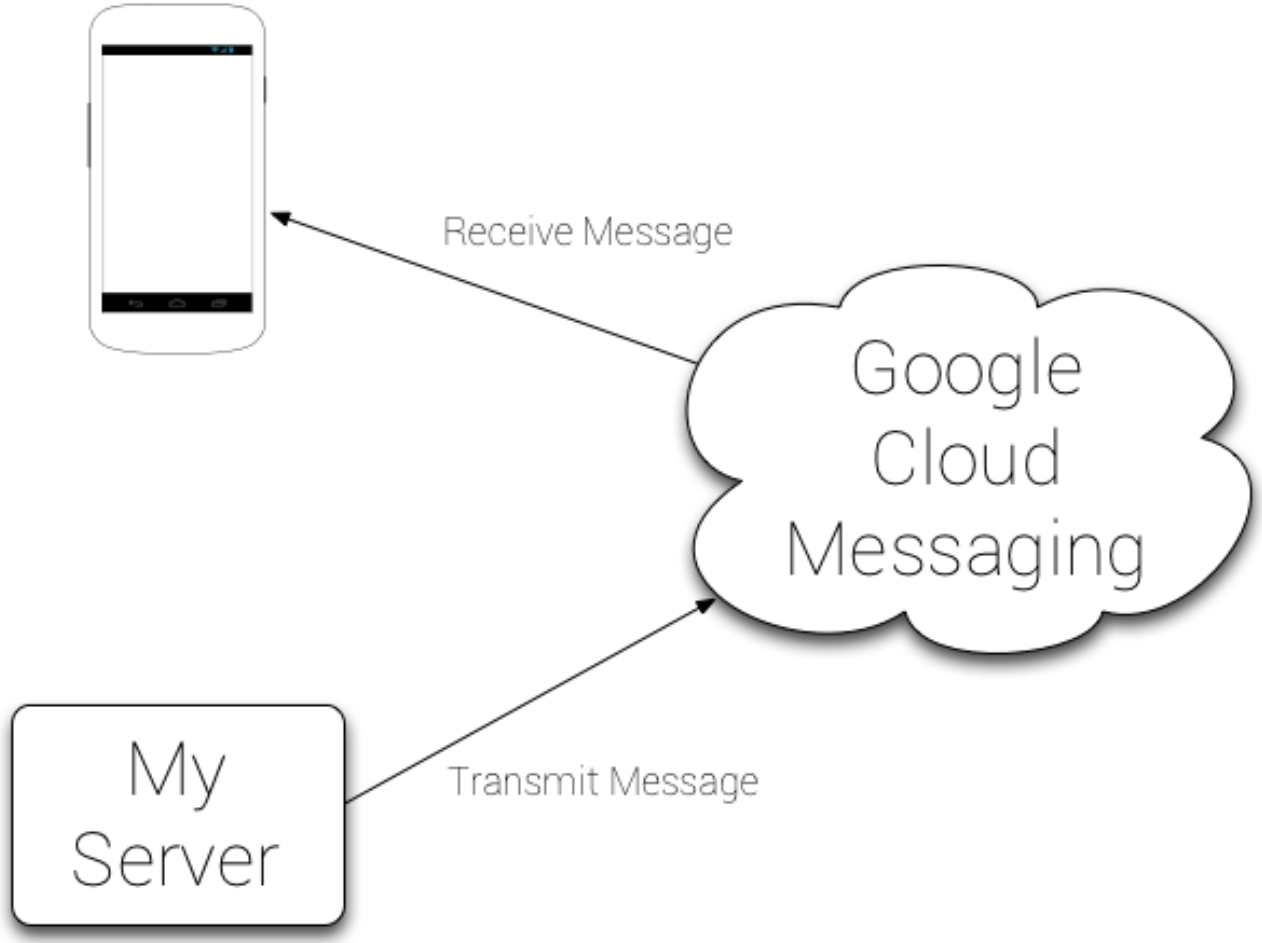
Client Updates

Server Updates

On-Demand Downloads

Cross-Device Updates

Google Cloud Messaging



Amongst the Ways to Transfer Data

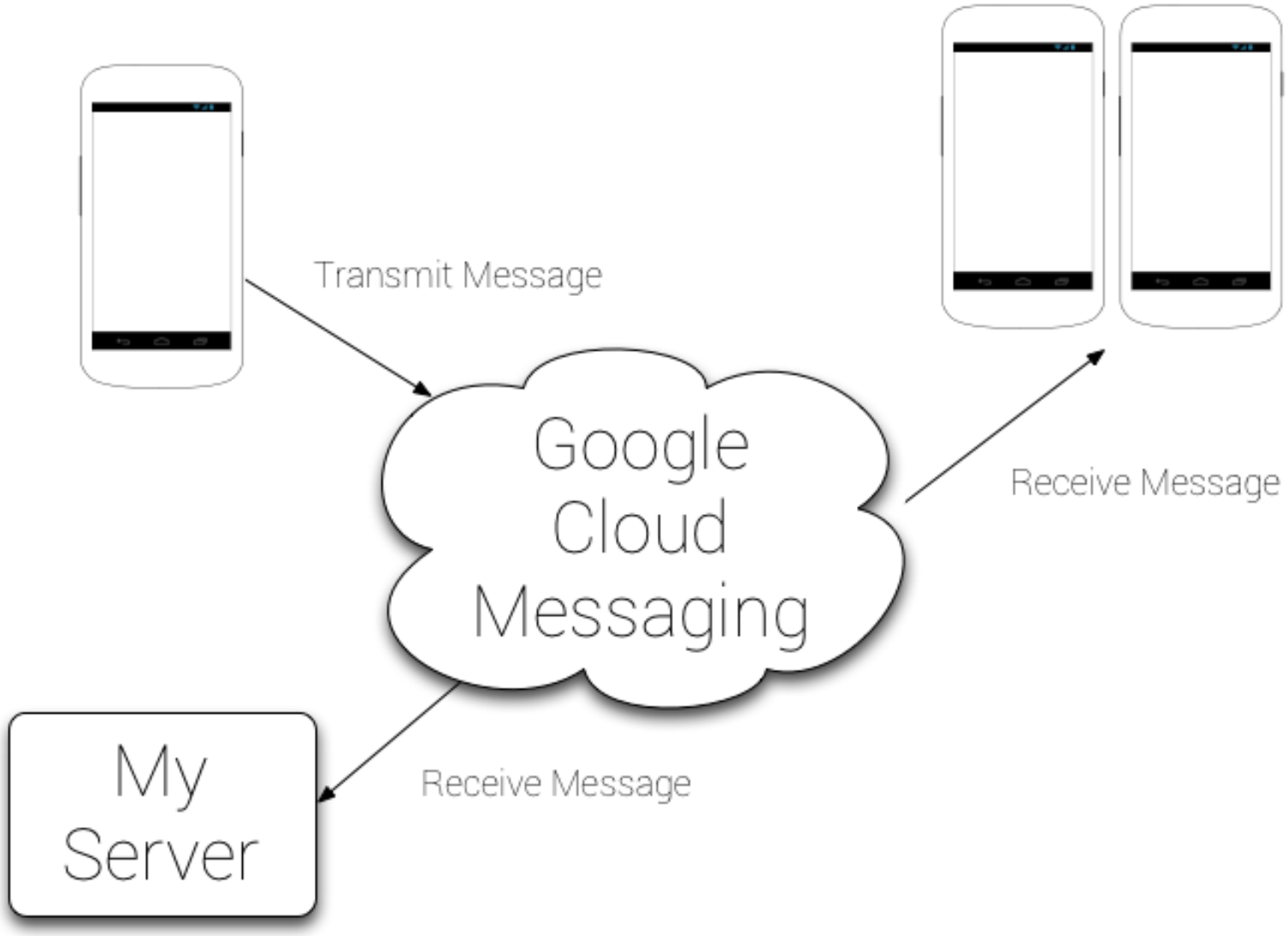
Client Updates

GCM

On-Demand Downloads

Cross-Device Updates

Google Cloud Messaging: Upstream



Google Cloud Messaging: Upstream

MyService

```
GoogleCloudMessaging gcm = GoogleCloudMessaging.get(context);

gcm.send(to, msgId, data);
```

Amongst the Ways to Transfer Data

Client Updates

GCM

On-Demand Downloads

Implementing a SyncAdapter

MySyncAdapter

```
public class MySyncAdapter extends AbstractThreadedSyncAdapter {
  public MySyncAdapter(Context context, boolean autoInitialize, boolean allowParallelSyncs) {
    super(context, autoInitialize, allowParallelSyncs);
  }

  public MySyncAdapter(Context context, boolean autoInitialize) {
    super(context, autoInitialize);
  }

  @Override
  public void onPerformSync(Account account, Bundle extras, String authority,
    ContentProviderClient provider, SyncResult syncResult) {
    // TODO Synchronize your data between client and server.
  }
}
```

SyncAdapter Service

MySyncService

```
public class MySyncService extends Service {
  private static final Object sSyncAdapterLock = new Object();
  private static MySyncAdapter sSyncAdapter = null;

  @Override
  public void onCreate() {
    synchronized (sSyncAdapterLock) {
      if (sSyncAdapter == null)
        sSyncAdapter = new MySyncAdapter(getApplicationContext(), true);
    }
  }

  @Override
  public IBinder onBind(Intent intent) {
    return sSyncAdapter.getSyncAdapterBinder();
  }
}
```

Abstract Account Manager

MyAuthenticationService

```
public class MyAccountAuthenticator extends AbstractAccountAuthenticator {
  public static final String ACCOUNT_TYPE = "com.mycompany.myapp";
  public static final String ACCOUNT_NAME = "MY STUB ACCOUNT";

 @Override
  public Bundle addAccount(AccountAuthenticatorResponse response, String accountType,
    String authTokenType, String[] requiredFeatures, Bundle options)
    throws NetworkErrorException {

    AccountManager manager = AccountManager.get(activity);
    final Account account = new Account(ACCOUNT_NAME, ACCOUNT_TYPE);
    manager.addAccountExplicitly(account, null, null);
    ContentResolver.setIsSyncable(account, authority, 1);
    ContentResolver.setSyncAutomatically(account, authority, true);
    return null;
  }
  ...
}
```

Account Manager Service

MyAuthenticationService

```
public class MyAuthenticationService extends Service {
  MyAccountAuthenticator mAuthenticator;

  @Override
  public void onCreate() {
    mAuthenticator = new MyAccountAuthenticator(this);
  }

  @Override
  public IBinder onBind(Intent intent) {
    return mAuthenticator.getIBinder();
  }
}
```

Account Manager XML Config

accountauth.xml

```
<account-authenticator xmlns:android="..."
    android:accountType="com.mycompany.myapp"
    android:icon="@drawable/miniicon"
    android:smallIcon="@drawable/miniicon"
    android:label="@string/app_name"
/>
```

Content Provider

MyContentProvider

```
public class MyContentProvider extends ContentProvider {
  @Override
  public boolean onCreate() { return true; }

  @Override
  public String getType(Uri uri) { return "vnd.android.cursor.dir/vnd.myapp.items”; }

  @Override
  public Cursor query(Uri uri, String[] projection, String selection,
    String[] selectionArgs, String sort){ return null; }

  @Override
  public Uri insert(Uri uri, ContentValues initialValues) { return null; }

  @Override
  public int delete(Uri uri, String where, String[] whereArgs) { return 0; }

  @Override
  public int update(Uri uri, ContentValues values, String where, String[] whereArgs) {  return 0; }
}
```

Manifest

AndroidManifest.xml

```
<provider android:authorities="com.mycompany.myappp.myauthority"
          android:name=".content_providers.PlaceDetailsContentProvider" />
```

Sync Adapter XML Config

sync_myapp.xml

```
<sync-adapter xmlns:android="http://schemas.android.com/apk/res/android"
  android:contentAuthority="com.mycompany.myappp.myauthority"
  android:accountType="com.mycompany.myapp"
  android:userVisible="false"
/>
```

Manifest

AndroidManifest.xml

```
<provider android:authorities="com.mycompany.myappp.myauthority"
          android:name=".content_providers.PlaceDetailsContentProvider" />

<service android:name=".MyAuthenticationService" android:exported="true">
  <intent-filter>
    <action android:name="android.accounts.AccountAuthenticator" />
  </intent-filter>
  <meta-data
    android:name="android.accounts.AccountAuthenticator"
    android:resource="@xml/accountauth" />
</service>

<service android:name=".MySyncService" android:exported="true">
  <intent-filter>
    <action android:name="android.content.SyncAdapter" />
    </intent-filter>
    <meta-data
      android:name="android.content.SyncAdapter"
      android:resource="@xml/sync_myapp" />
</service>
```

Triggering Syncs

MyService

```
final Account account =
  new Account(null, MyAccountAuthenticator.ACCOUNT_TYPE);
String authority = "com.mycompany.myapp.myauthority"

mContentResolver.requestSync(account, authority, null);
```

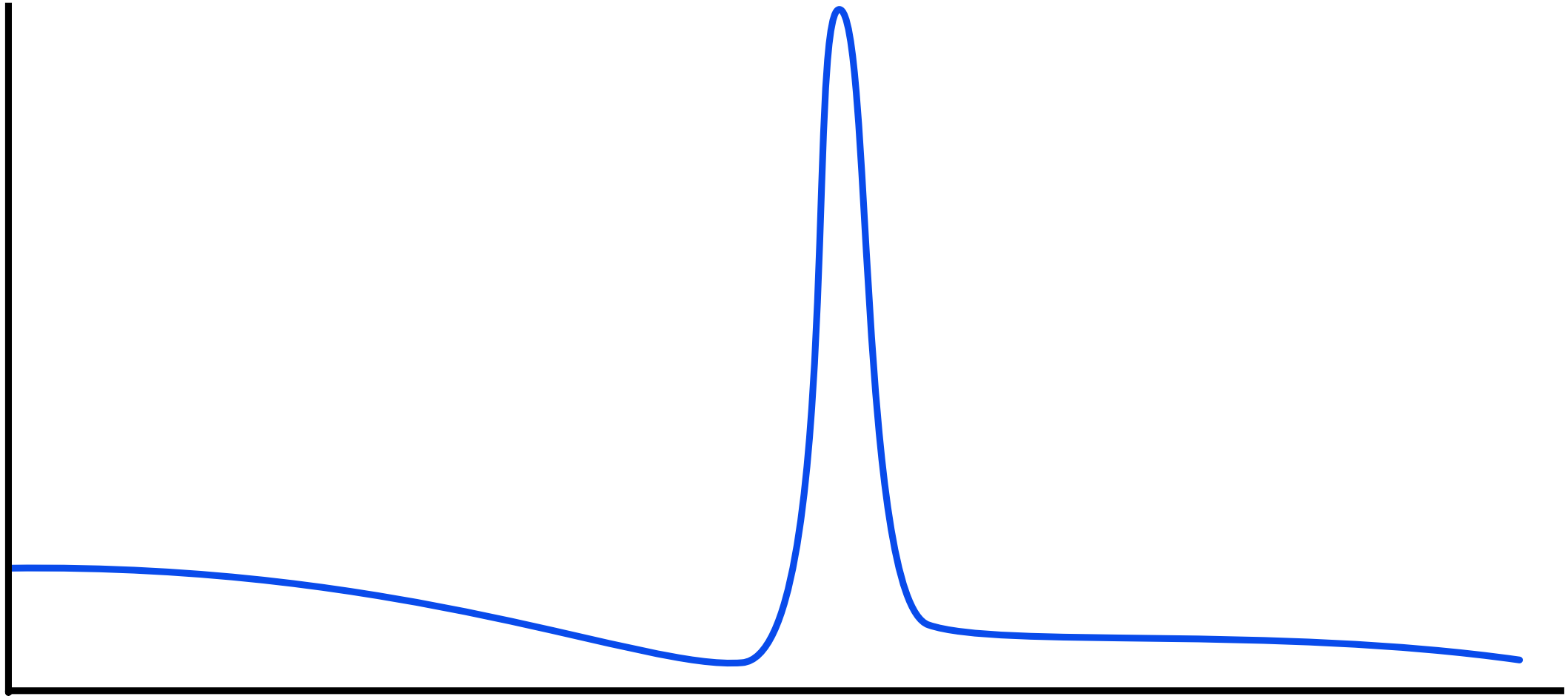
Periodic Repeating Syncs

MyService

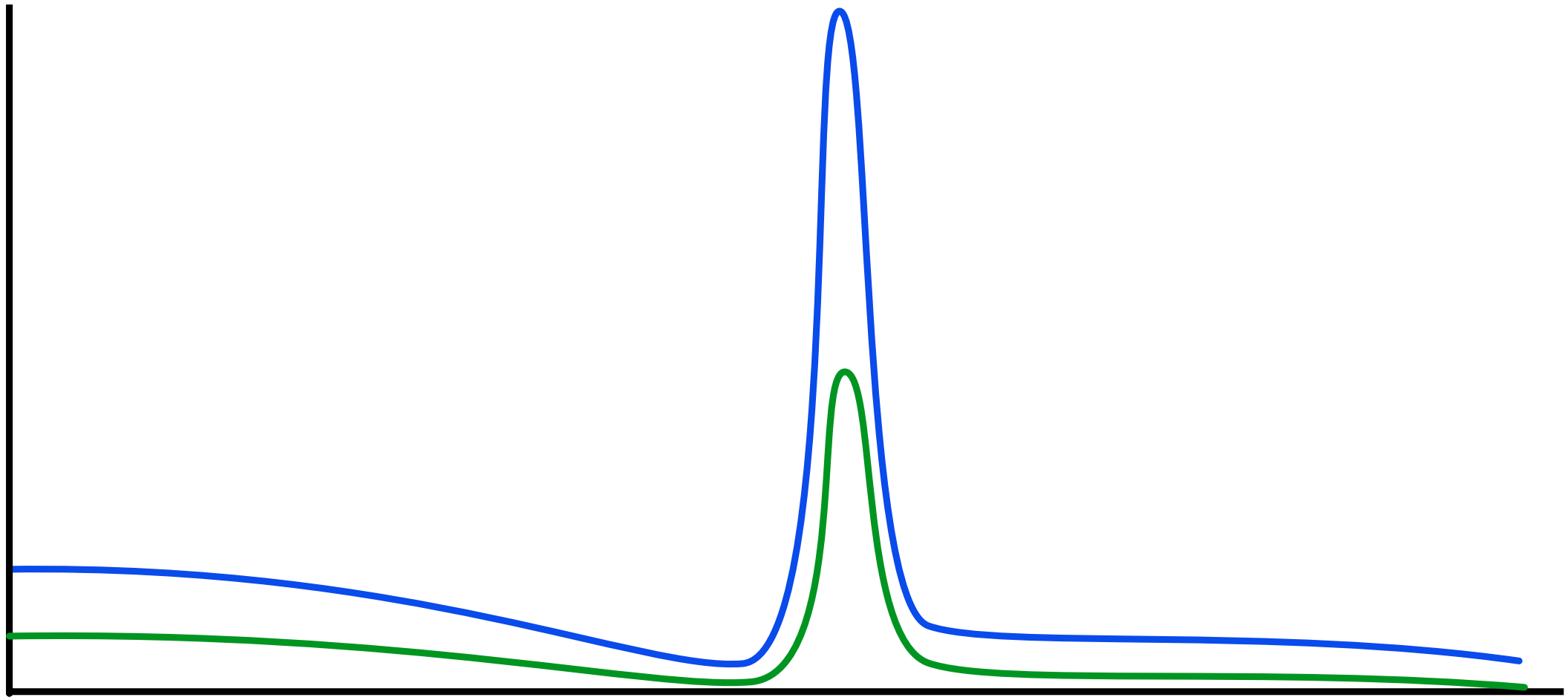
```
final Account account =
  new Account(null, MyAccountAuthenticator.ACCOUNT_TYPE);
String authority = "com.mycompany.myapp.myauthority"
long interval = 24*60*60; // 12 hours (24hr by default).

mContentResolver.addPeriodicSync(account, authority, null, interval);
```

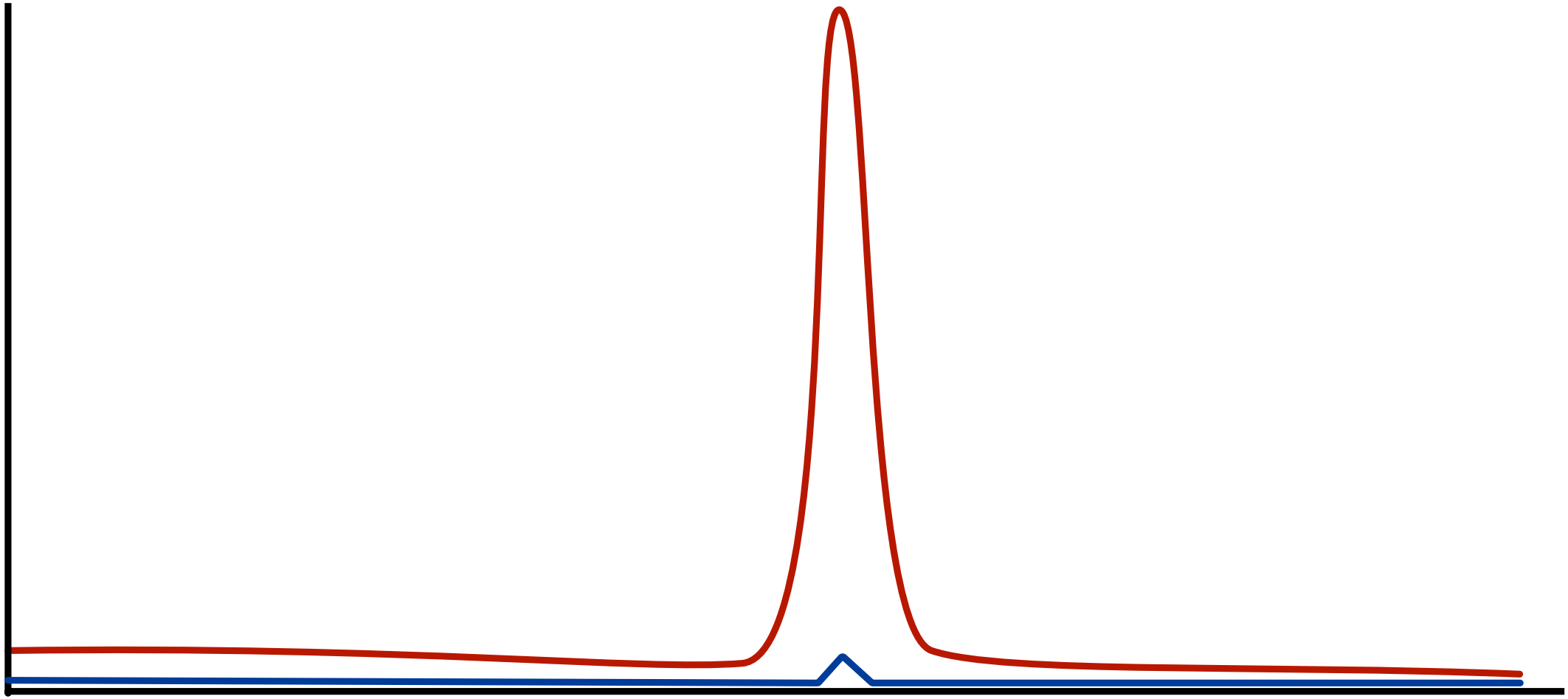
Your Server Spike



Do You Even Scale?



Carrier Service Spike



Update Window

MyService

```
int alarmType = AlarmManager.ELAPSED_REALTIME;
long interval = AlarmManager.INTERVAL_HOUR;

// int jitter = 3 // (Chosen by fair dice roll. Guaranteed to be random.
Random random = new Random();
int jitter = random.nextInt(60);

long start = SystemClock.elapsedRealtime() + interval - ((30 + jitter)*60000);

alarmManager.setInexactRepeating(alarmType, start, interval, pi)
```

Update Window

MyService

```
int alarmType = AlarmManager.ELAPSED_REALTIME;
long interval = AlarmManager.INTERVAL_HOUR;

// int jitter = 3 // (Chosen by fair dice roll. Guaranteed to be random.
Random random = new Random();
int jitter = random.nextInt(60);

long start = SystemClock.elapsedRealtime() + interval - ((30 + jitter)*60000);

alarmManager.setInexactRepeating(alarmType, start, interval, pi)
```

Activity Recognition: Pickup Trigger

MyReceiver

```
@Override
public void onReceive(Context context, Intent intent) {
  // Extract the Activity that's been detected.
  ActivityDetectionResult activity =
    ActivityDetectionResult.extractResult(intent);

  if (activity != null) {
    ActivityType activityType = activity.getMostProbableActivity().getType();
    if (ActivityType.valueOf(activityTypeString) == ActivityType.TILTING)
      context.startService(new Intent(context, DailyUpdateService.class);
  }
}
```

SyncAdapter

GCM

On-Demand Downloads

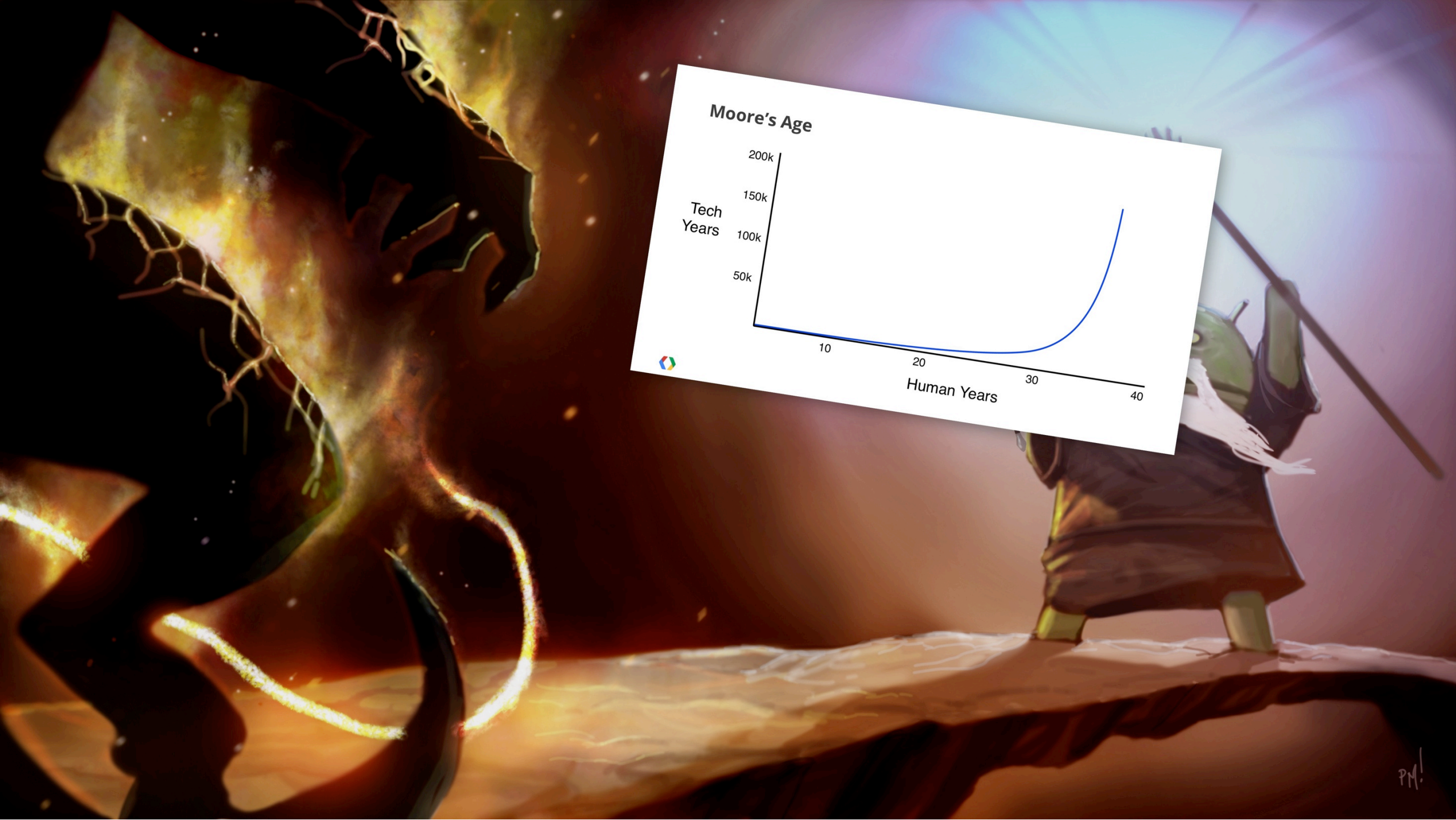
SyncAdapter with Calls to Other Updates

MySyncAdapter

```
@Override
public void onPerformSync(Account account, Bundle extras, String authority,
                ContentProviderClient provider, SyncResult syncResult) {

  downloadServerSideSync();
  uploadClientSideSync();
  transmitBatchedAnalytics();
  executePrefetch();
  retryFailedTransfers();
}
```

PM!

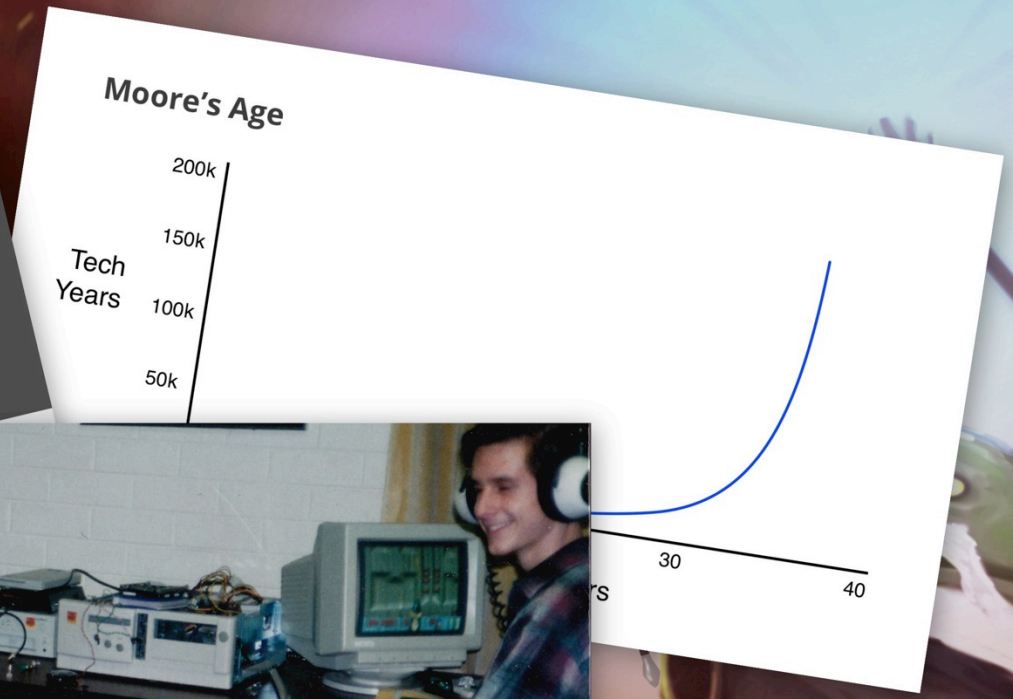

Moore's Age
200k
150k
100k
50k
Tech Years
10
20
30
40
Human Years

"Any significantly advanced technology is indistinguishable from magic."

Arthur C. Clarke

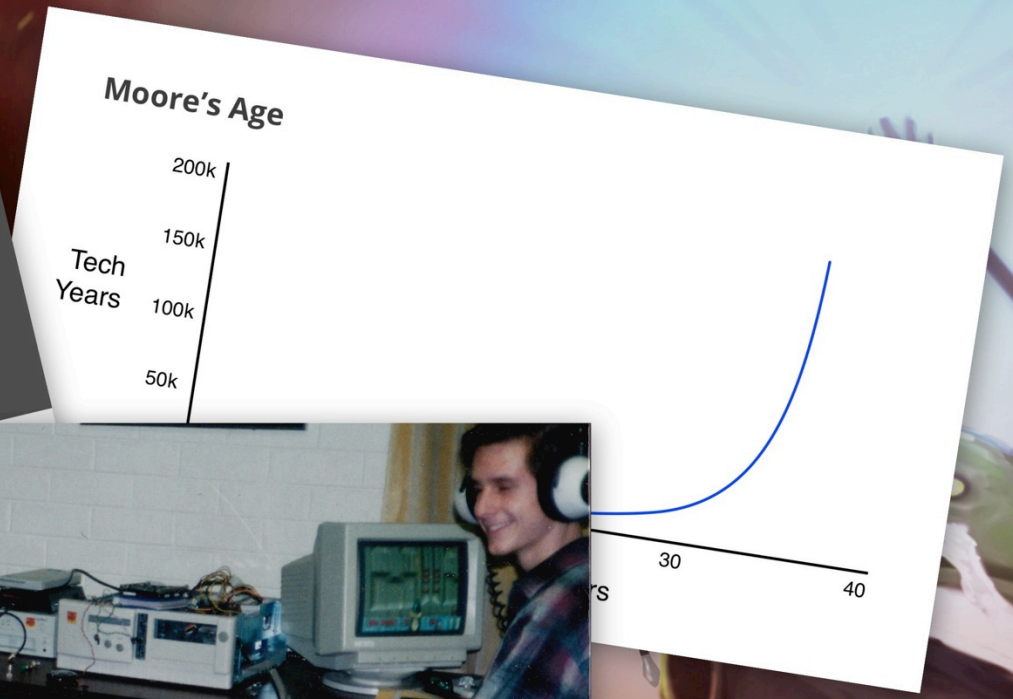
"Any significantly advanced technology is indistinguishable from magic."

Arthur C. Clarke



"Any significantly advanced technology is indistinguishable from magic."

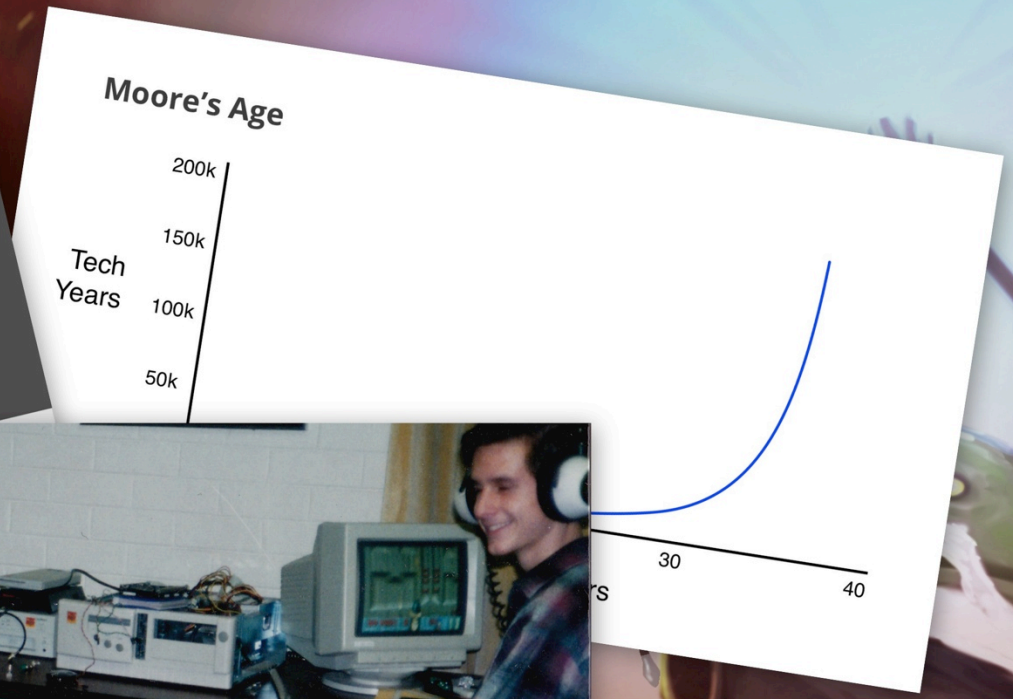
Arthur C. Clarke



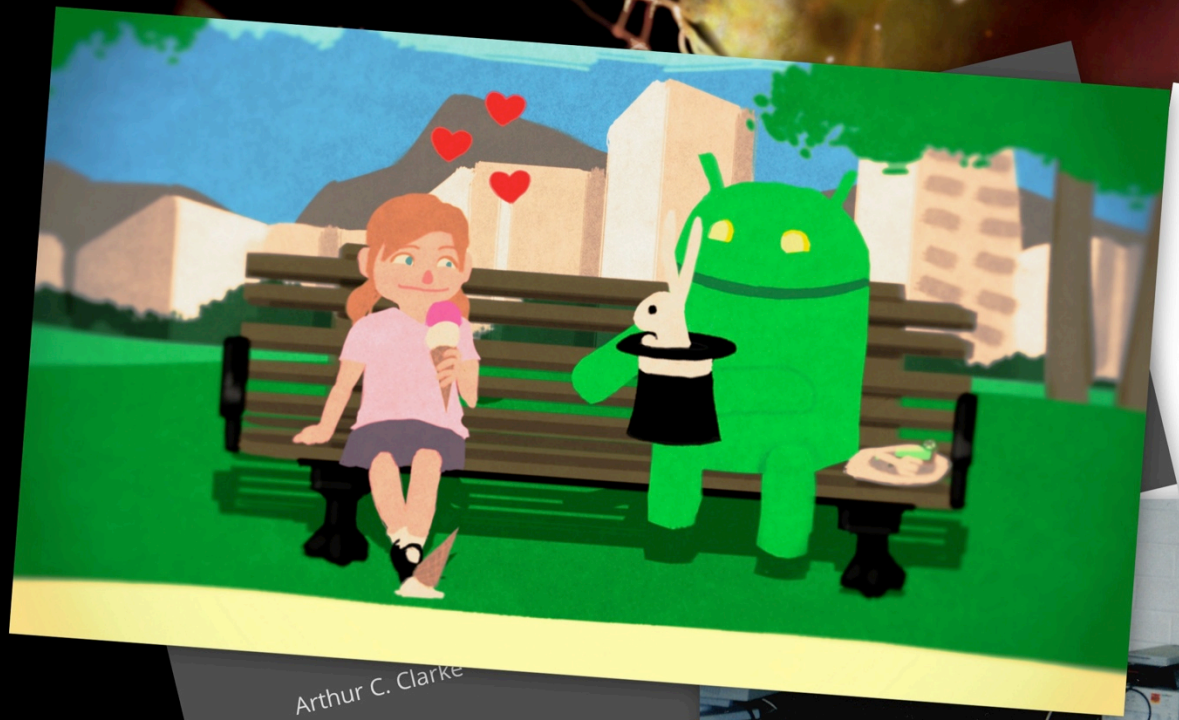
Developing the Impossible

"Any significantly advanced technology is indistinguishable from magic."

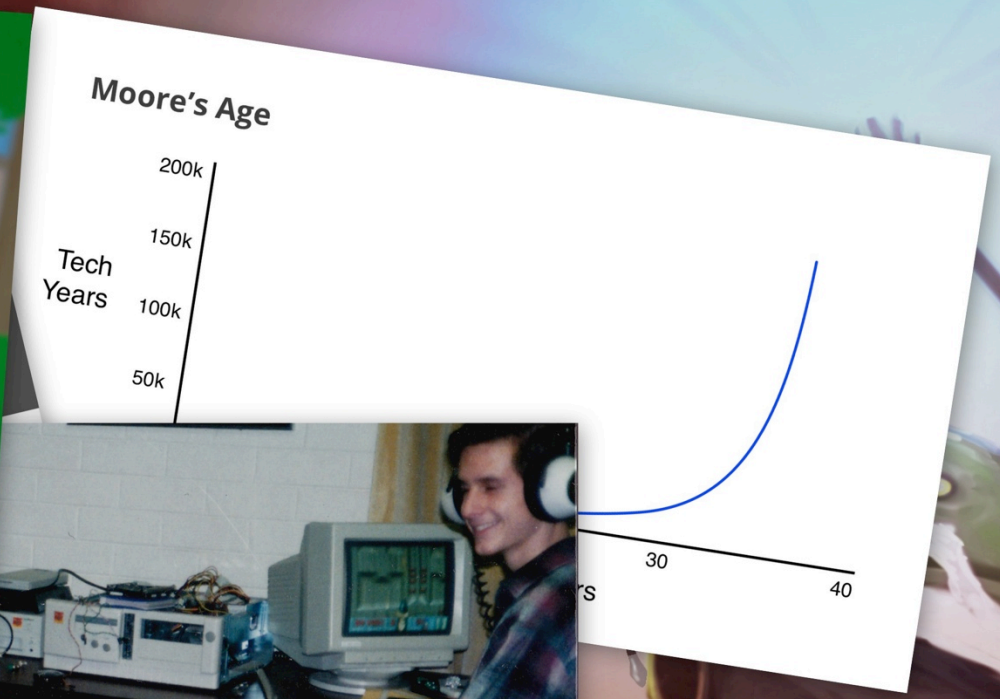
Arthur C. Clarke



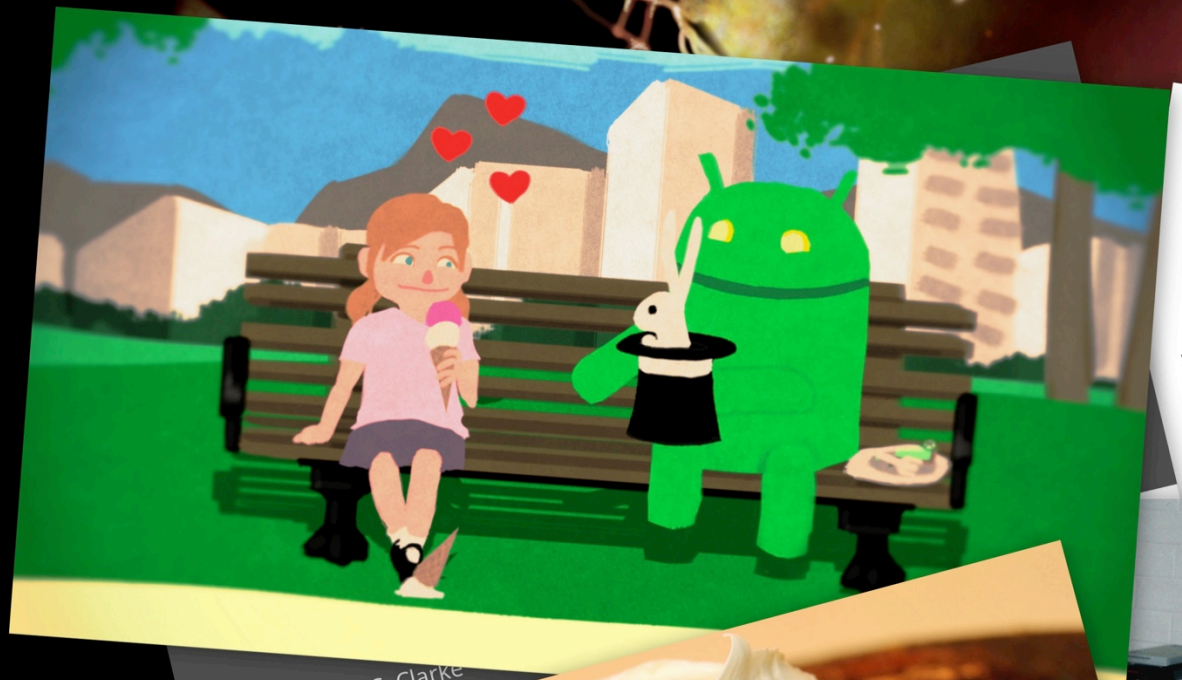
Developing the Impossible



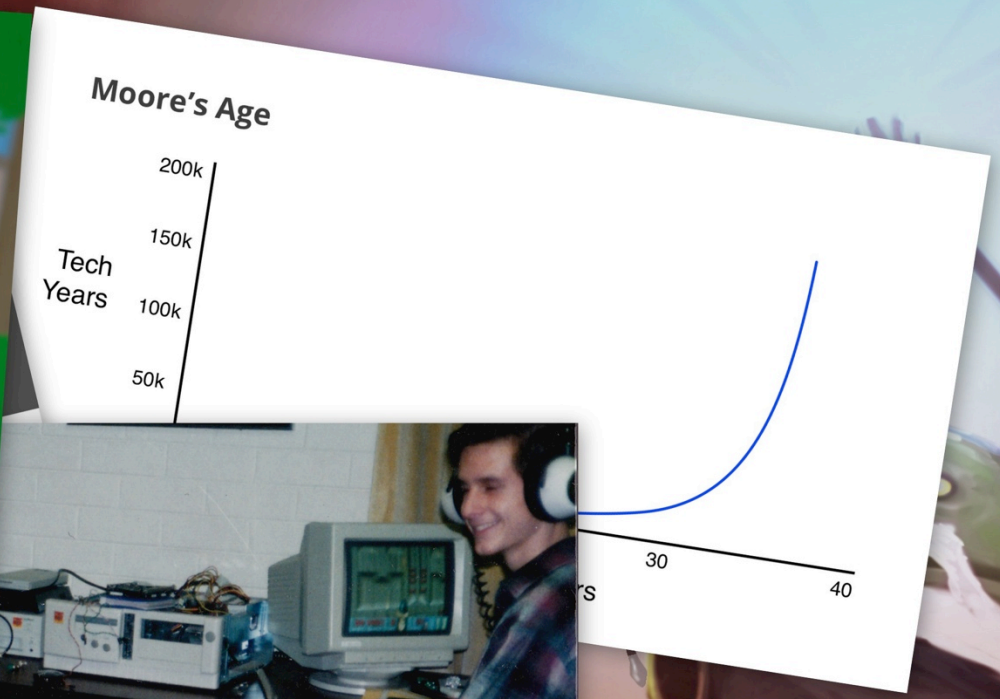
Arthur C. Clarke

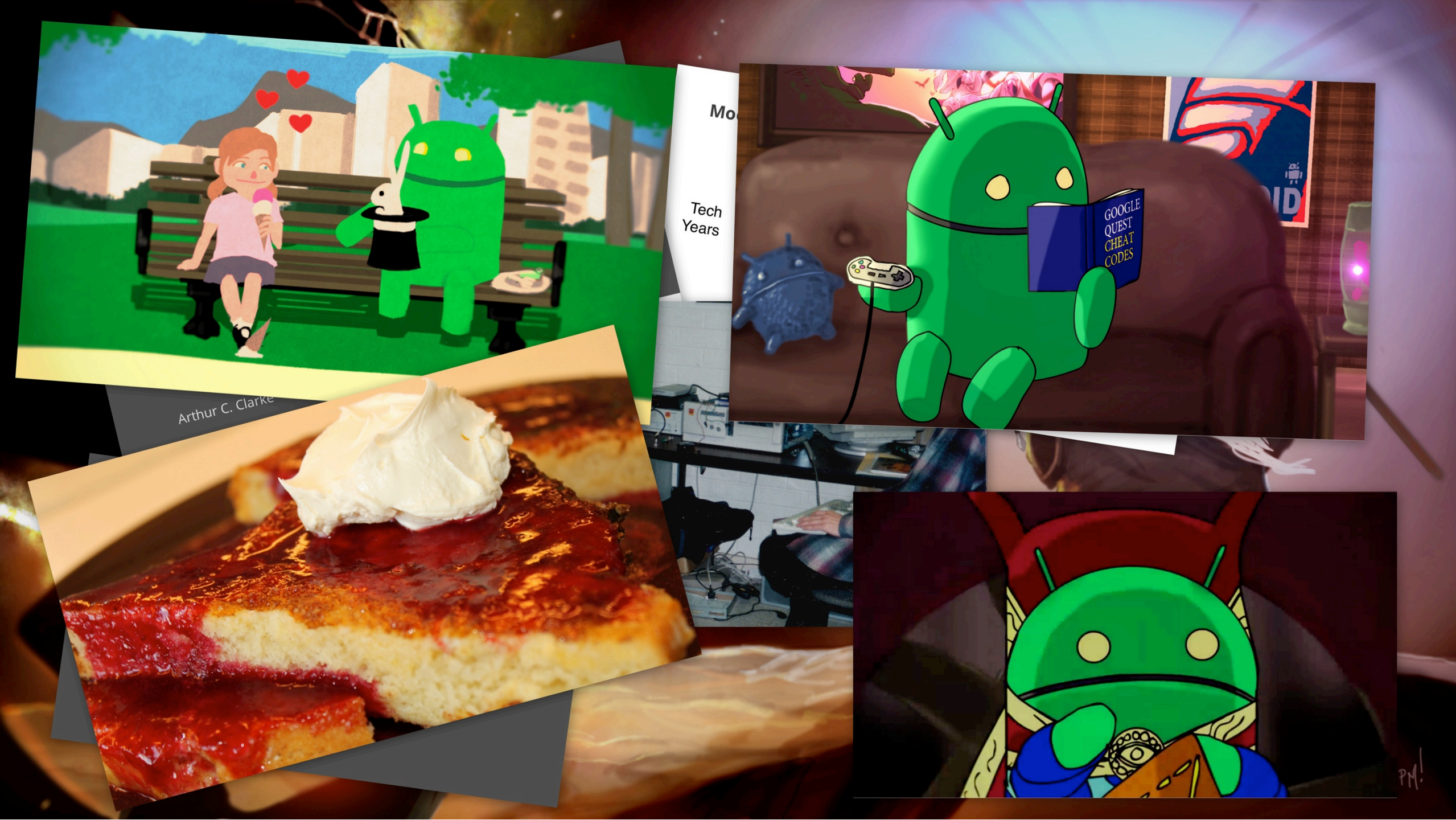


Developing the Impossible



Arthur C. Clarke




Mo
Tech
Years
Arthur C. Clarke
GOOGLE
QUEST
CHEAT
CODES

“Stop developing for the future and start defining the future itself.”

Thank You!

Created and presented by Reto Meier (+RetoMeier)

Graphics & animations by Pandamusk (@pandamusk)
Additional music by Joel Alford "Bliss" (@TheDinosauras)

Questions? Join me at the Android Developers Office Hours: 3rd floor

Google
Developers