

<http://poloclub.gatech.edu/cse6242>

CSE6242 / CX4242: **Data** & **Visual** Analytics

Data Collection

Duen Horng (Polo) Chau

Assistant Professor

Associate Director, MS Analytics

Georgia Tech

Partly based on materials by

Professors Guy Lebanon, Jeffrey Heer, John Stasko, Christos Faloutsos

How to Collect Data?

Method

Effort

Download

Low

API

(Application program interface)

Medium

Scrape/Crawl

High

How to Collect Data?

Method

Effort

Download

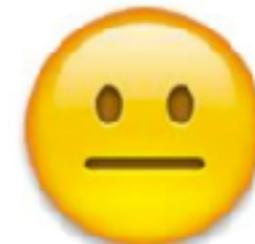
Low



API

(Application program interface)

Medium



Scrape/Crawl

High



Data you can just download

NYC Taxi data: Trip (11GB), Fare (7.7GB)

StackOverflow (xml)

Wikipedia (data dump)

Atlanta crime data (csv)

Soccer statistics

Data.gov

...

Data you can just download

If you have leads, let us know on Piazza!

More datasets on course website:

<http://poloclub.gatech.edu/cse6242/2017spring/#datasets>

Schedule

Homework

Project

Datasets

Readings & Resources

Prerequisites

All students must first review prerequisites & course expectation.

CSE6242 / CX4242, Spring 2017

Data and Visual Analytics

Collect Data via APIs

Google Data API (e.g., Google Maps Directions API)

<https://developers.google.com/gdata/docs/directory>

Twitter (small subset)

<https://dev.twitter.com/streaming/overview>

Last.fm (Pandora has unofficial API)

Flickr

data.nasa.gov

data.gov

Facebook (your friends only)

iTunes

Data that needs scraping

Amazon (reviews, product info)

ESPN

eBay

Google Play

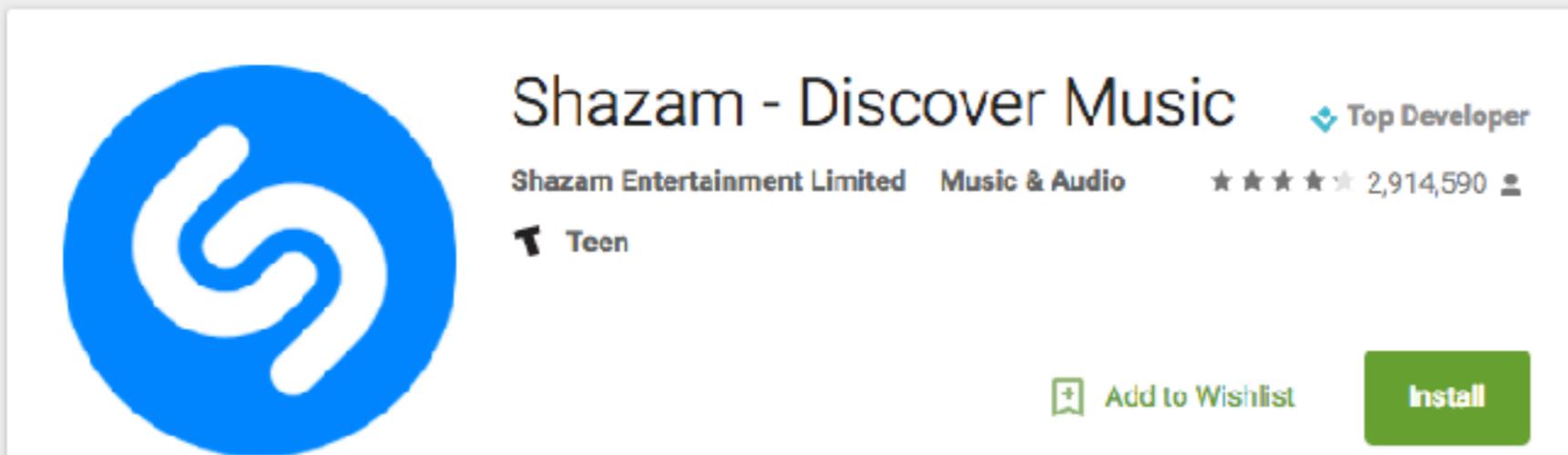
Google Scholar

...

How to Scrape?

Google Play example

Goal: collect the network of similar apps

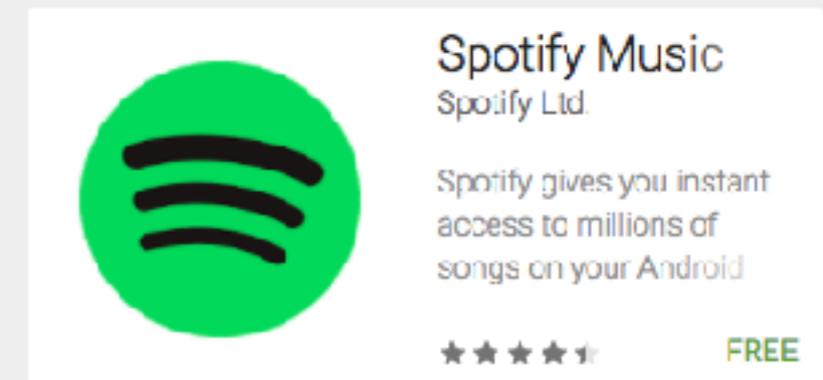


Shazam - Discover Music Top Developer
Shazam Entertainment Limited Music & Audio ★★★★★ 2,914,590
Teen

[Add to Wishlist](#) [Install](#)

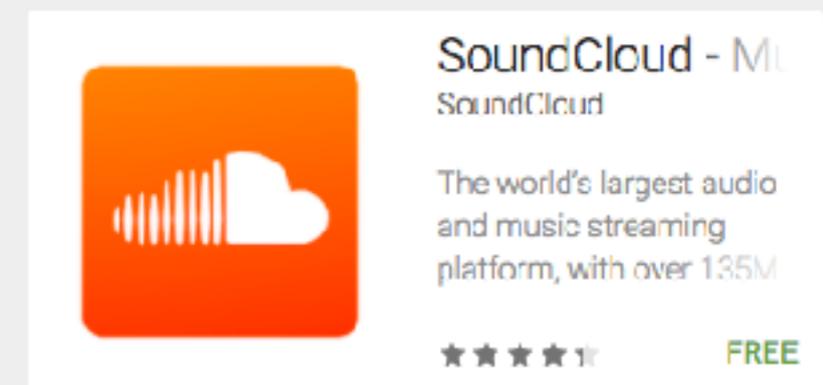
Similar

[See more](#)



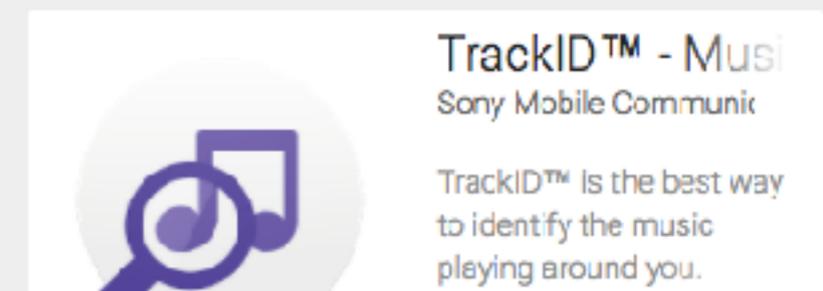
Spotify Music
Spotify Ltd.
Spotify gives you instant access to millions of songs on your Android

★★★★★ **FREE**

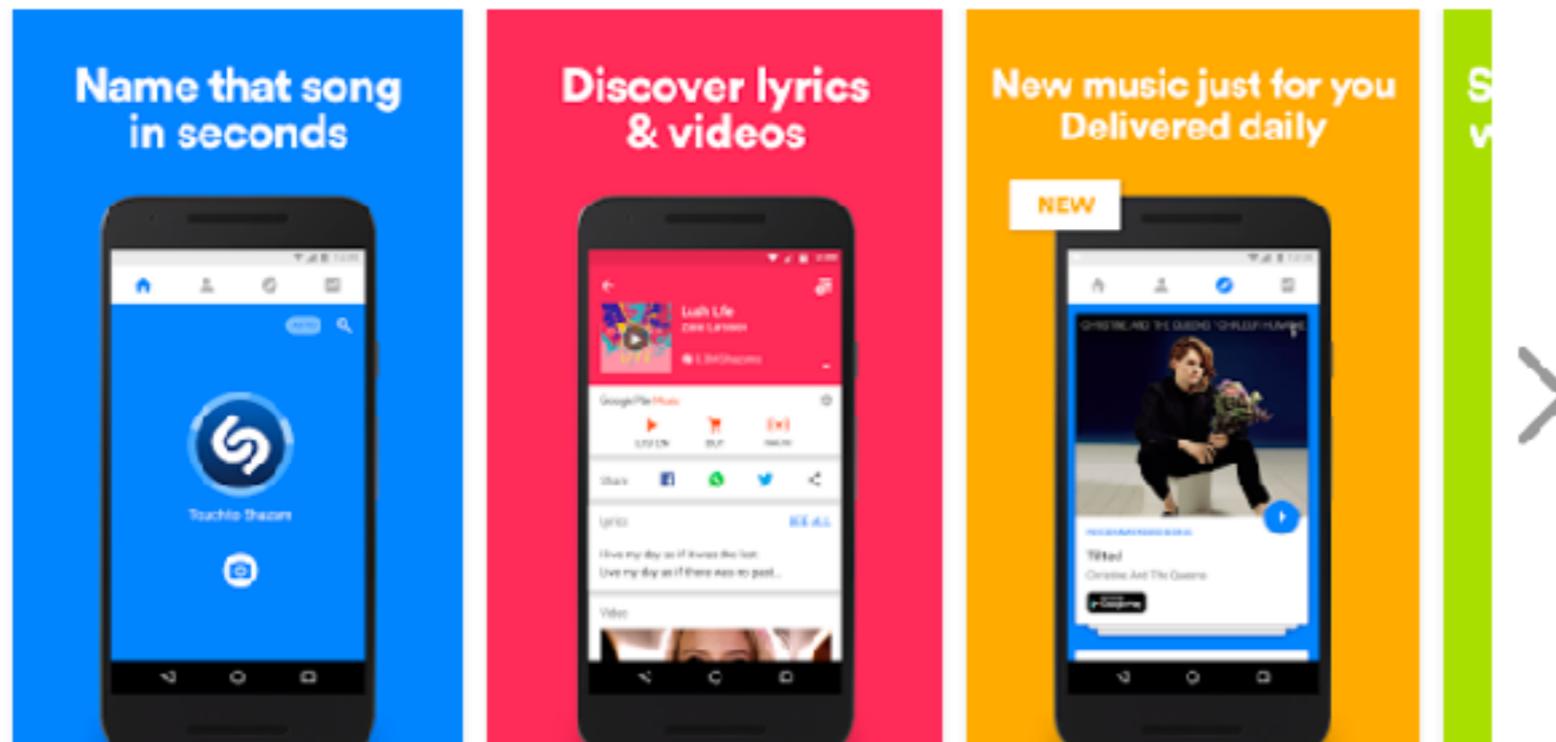


SoundCloud - Music
SoundCloud
The world's largest audio and music streaming platform, with over 135M

★★★★★ **FREE**



TrackID™ - Music
Sony Mobile Communic
TrackID™ is the best way to identify the music playing around you.



Name that song in seconds

Discover lyrics & videos

New music just for you Delivered daily

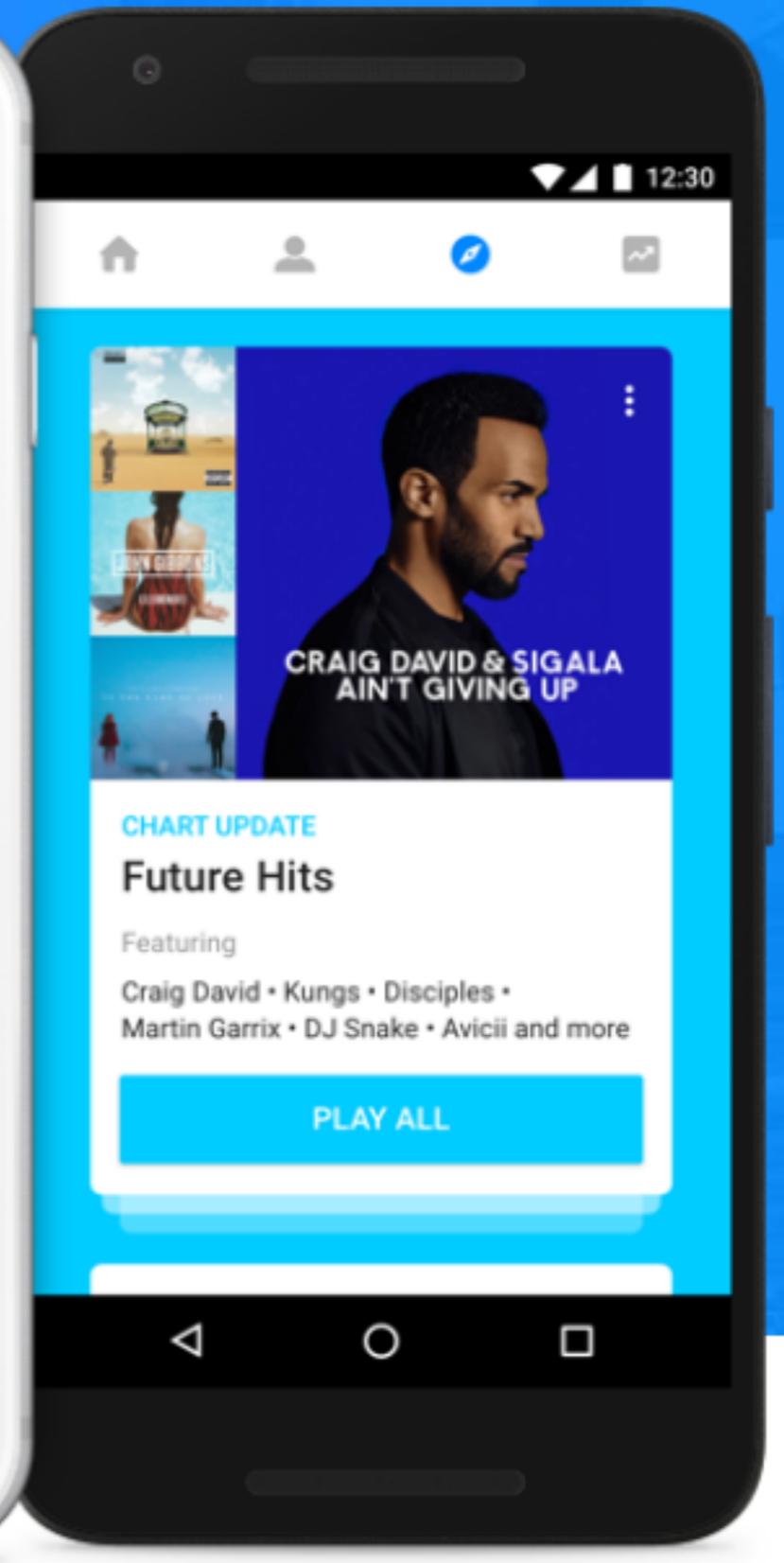
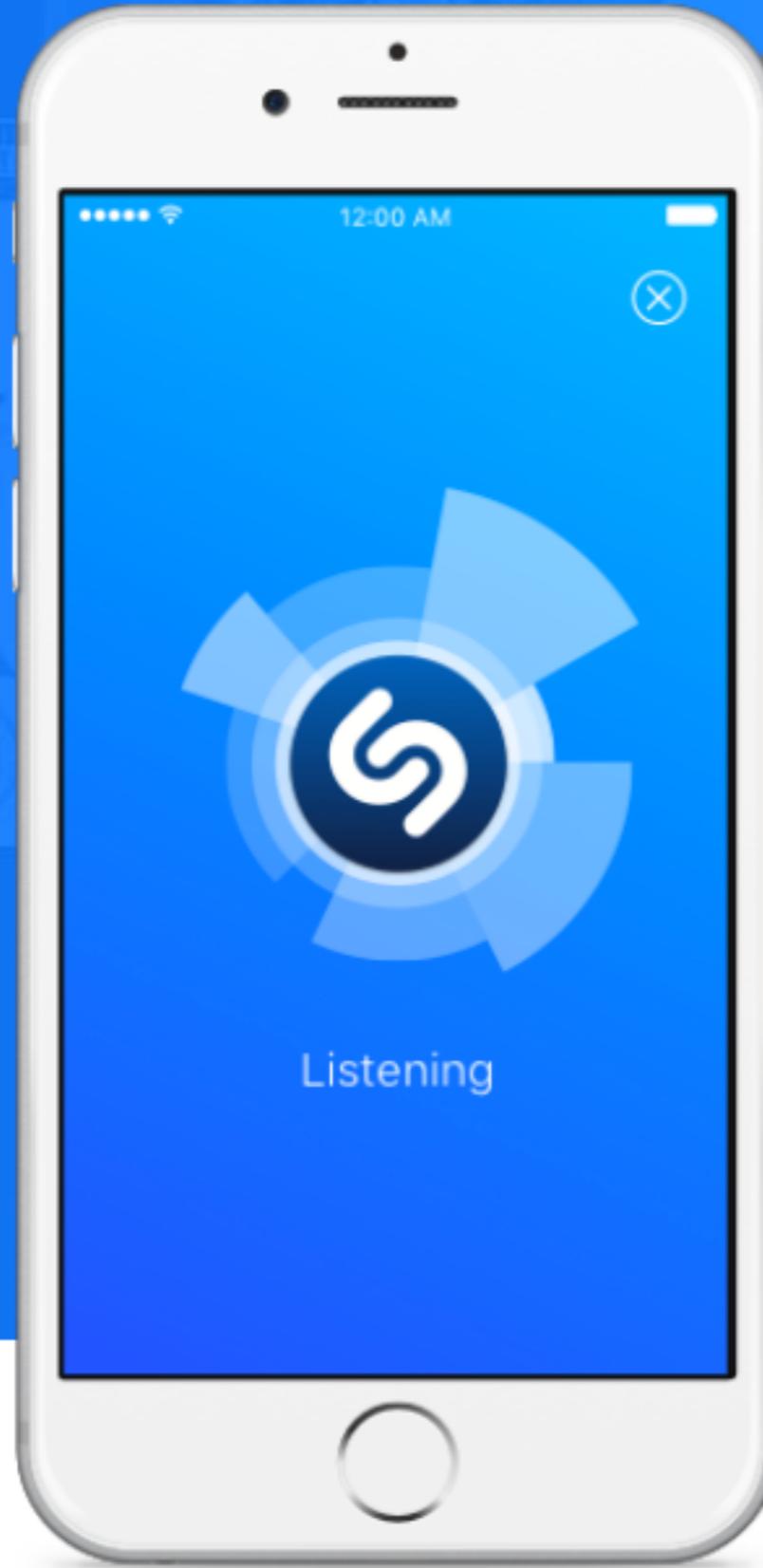
NEW

Shazam

Name any song in seconds

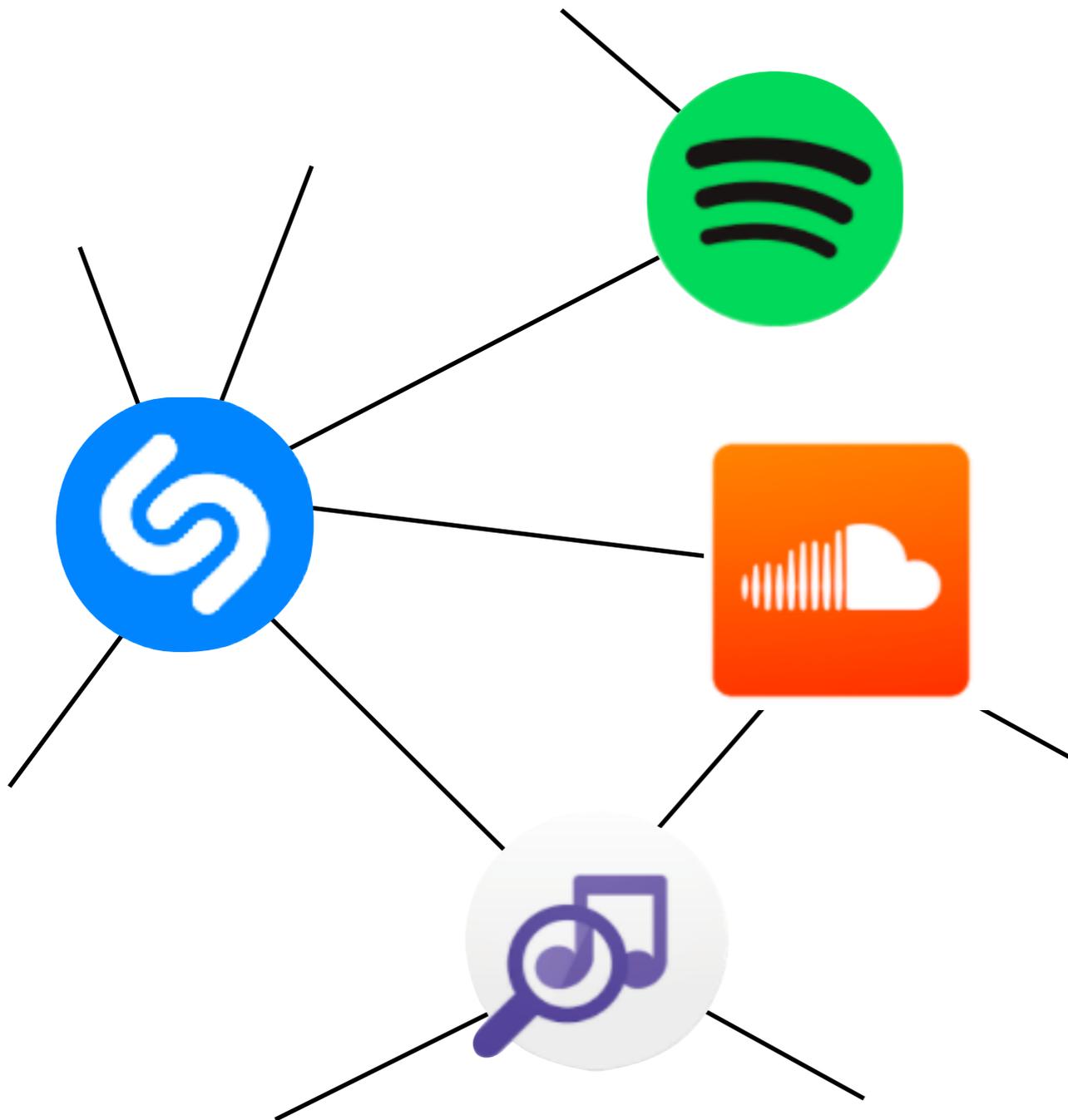
Shazam will identify any music playing around you.

GET IT NOW



How to Scrape?

Goal: Write a **program/algorithm** to scrape Google Play to **collect a million-node network** of similar apps



Each **node** is an app

An **edge** connects two similar apps

Hint: start with some apps (e.g., Shazam), and go from there.

How to Scrape?

Google Play example

Goal: collect the network of similar apps

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



<https://play.google.com/store/apps/details?id=com.spotify.music>

How to store the data?
What's the easiest way?

Easiest Way to Store Data

As comma-separated files (CSV)

But may not be easy to parse. Why?

```
1997,Ford,E350
```

Easiest Way to Store Data

```
1997,Ford,E350
```

- Any field *may* be *quoted* (that is, enclosed within double-quote characters). Some fields *must* be quoted.

```
"1997","Ford","E350"
```

- Fields with embedded commas or double-quote characters must be quoted.

```
1997,Ford,E350,"Super, luxurious truck"
```

- Each of the embedded double-quote characters must be represented by a pair of double-quote characters.

```
1997,Ford,E350,"Super, ""luxurious"" truck"
```

- Fields with embedded line breaks must be quoted (however, many CSV implementations do not support this).



Most popular embedded database in the world

Well-known users: <http://www.sqlite.org/famous.html>
iPhone (iOS), Android, Chrome (browsers), Mac, etc.

Self-contained: one file contains data + schema

Serverless: database right on your computer

Zero-configuration: no need to set up!

SQL Refresher: create table

```
>sqlite3 database.db
```

```
sqlite> create table student(ssn integer, name text);
```

```
sqlite> .schema
```

```
CREATE TABLE student(ssn integer, name text);
```

ssn	name

SQL Refresher: insert rows

```
insert into student values(111, "Trump");  
insert into student values(222, "Johnson");  
insert into student values(333, "Obama");  
select * from student;
```

ssn	name
111	Trump
222	Johnson
333	Obama

SQL Refresher: create another table

```
create table takes  
(ssn integer, course_id integer, grade integer);
```

```
sqlite> .schema
```

```
CREATE TABLE student(ssn integer, name text);
```

```
CREATE TABLE takes (ssn integer, course_id integer,  
grade integer);
```

ssn	course_id	grade

SQL Refresher: joining 2 tables

More than one tables - **joins**

E.g., create roster for this course (6242)

ssn	name
111	Trump
222	Johnson
333	Obama

ssn	course_id	grade
111	6242	100
222	6242	90
222	4000	80

SQL Refresher: joining 2 tables + filtering

```
select name from student, takes
where
    student.ssn = takes.ssn and
    takes.course_id = 6242;
```

ssn	name
111	Trump
222	Johnson
333	Obama

ssn	course_id	grade
111	6242	100
222	6242	90
222	4000	80

Summarizing data:

Find `ssn` and `GPA` (a summary) for each student

```
select ssn, avg(grade)
from takes
group by ssn;
```

ssn	course_id	grade
111	6242	100
222	6242	90
222	4000	80

ssn	avg(grade)
111	100
222	85

Filtering Summarized Results

```
select ssn, avg(grade)
from takes
group by ssn
having avg(grade) > 90;
```

ssn	course_id	grade
111	6242	100
222	6242	90
222	4000	80

ssn	avg(grade)
111	100
222	85

SQL General Form

```
select a1, a2, ... an  
from t1, t2, ... tm  
where predicate  
[order by ...]  
[group by ...]  
[having ...]
```

A lot more to learn!
Highly recommend taking
CS 4400 Introduction to Database Systems

SQLite easily scales to multiple GBs. What if slow?

Important sanity check: Have you (or someone) created appropriate **indexes**?

SQLite's indices use **B-tree** data structure.
 $O(\log n)$ speed for adding/finding/deleting an item.

```
create index student_ssn_index on  
student(ssn);
```

