
Google™
11
I/O

The YouTube Caption API, Speech Recognition, and WebVTT Captions for HTML5

Naomi Black, Cynthia Boedihardjo, and Jeffrey Posnick

May 11, 2011

Welcome!

Hashtags: #io2011 #youtube

Feedback: http://goo.gl/NLAu0

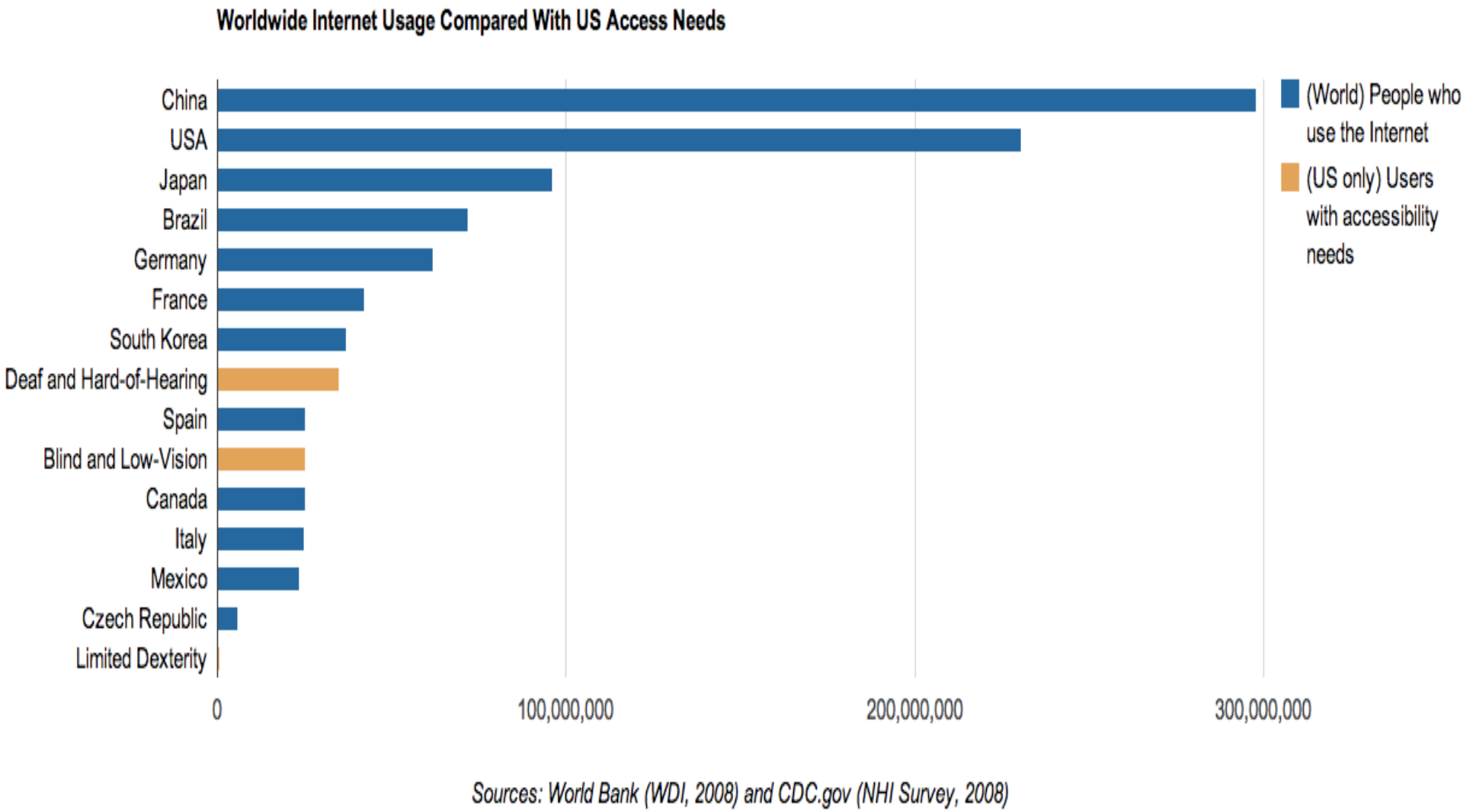
email: captioning@google.com

Overview

- Captions for I/O Live
- WebVTT (Code & Demos)
- The YouTube Captions API (Code & Demos)

Why Captions?

Why captions?



Why captions?

- Accessibility!
- A captioned video is searchable.
- Captioned video is text and can be translated.
- Same-language subtitles help comprehension.

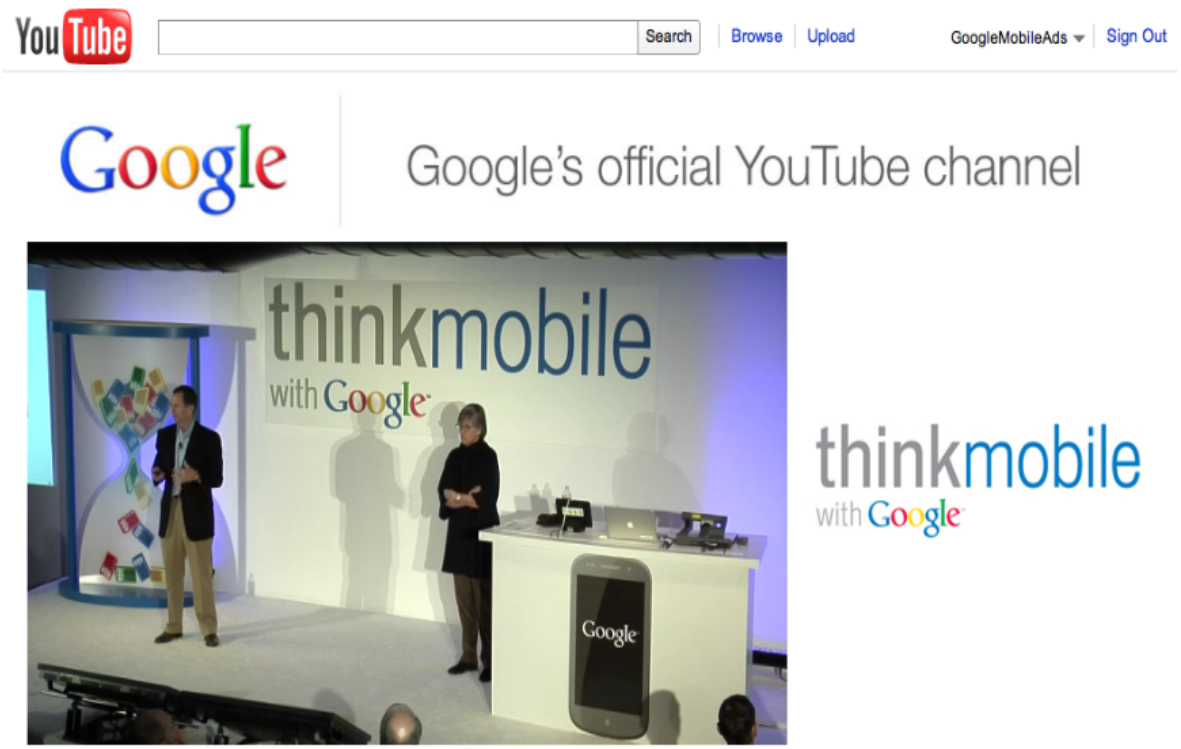
Real-Time Captions for I/O Live

Real-time Captions for I/O Live

- CART provider (a person!) types the text in real-time.
- Text is sent over TCP/IP to a StreamText Server.
- StreamText uses .NET.
- We provision with App Engine and stream to I/O Live Viewers.

Building a Caption Gadget for Live Events

- New LIVE platform on youtube.com/live
- Live events include:
  - Product launches
  - Concerts
  - Sporting Events
  - Google I/O
- Increase Live events on YouTube



Building the Caption Gadget

- Contracted help to build gadget
  - Psycle Interactive Limited - independent production company
  - StreamText.net - realtime streaming text service provider
- Gadget requirements
  1. Take real time text feed and serve transcript to gadget
  2. Translate real time feed into multiple languages using Google Translate API
  3. Make the code open source

Features of Caption Gadget

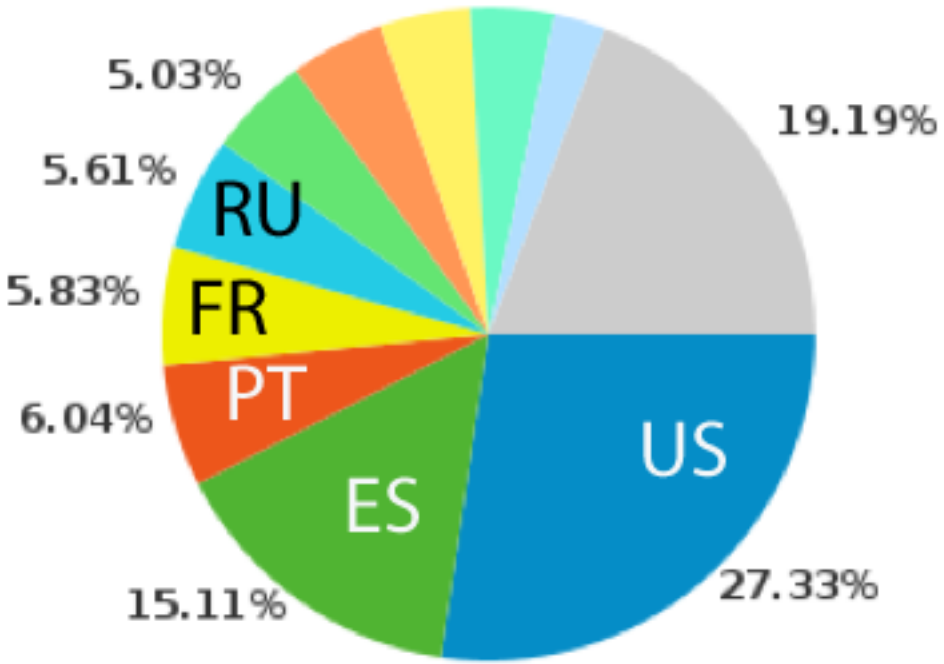
- View gadget on google.com/io
- Technical Challenges
  - Handle large amount of viewership traffic
  - Delay between live event to streaming player
- Features include:
  - Translates to 57 languages powered by Google Translate
  - Ability to add delay
  - Streams word by word in English
  - Streams 35 characters at a time for translated languages

Captioning for Google I/O

Over **250k** viewers
to captioning gadget

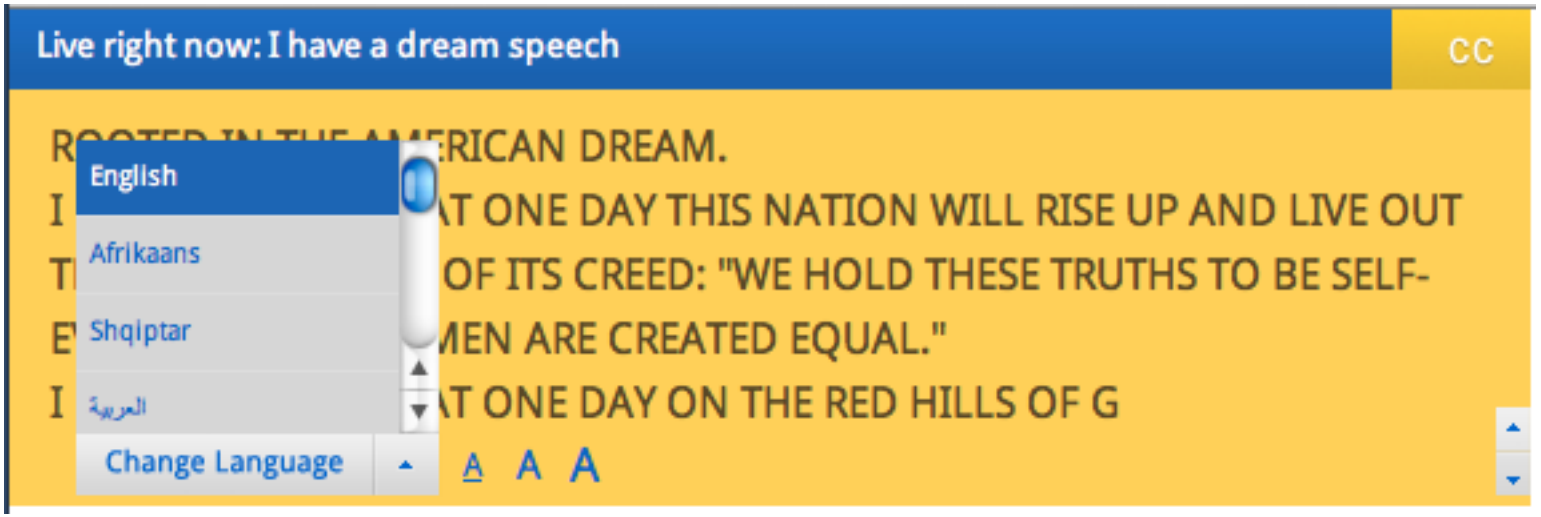
Out of **57** languages
52 were selected
for translation

Top 5



How to Use the Gadget for Your LIVE Event

- Using the Gadget
  - Code available on: http://code.google.com/p/io-captions-gadget/
  - Gadget available to select YouTube LIVE partners



WebVTT: Timed Text for HTML5

WebVTT: Timed Text for HTML5

For deaf, blind and other users

To watch video on the Web...

- Hearing-impaired people need captions
- Vision-impaired people need audio descriptions
- Non-native speakers need subtitles
- Everyone finds synchronized music lyrics useful
- Everyone finds navigation markers (like DVD chapters) useful

HTML5 has a text-based solution for all these:

WebVTT and the TextTrack API

A WebVTT example file

Styled and positioned captions

WEBVTT FILE

0:00:00.000 --> 0:00:02.000
(Can you hear me all right?)

0:00:03.040 --> 0:00:06.920 T:60% A:middle
So, I just wanted to introduce you to W3C

Styling

<u>

Positioning

Horizontal Text Position:
T:<percentage>
Alignment:
A:middle
Vertical Line position:
L:<percentage>

A WebVTT example file

Using CSS for styling

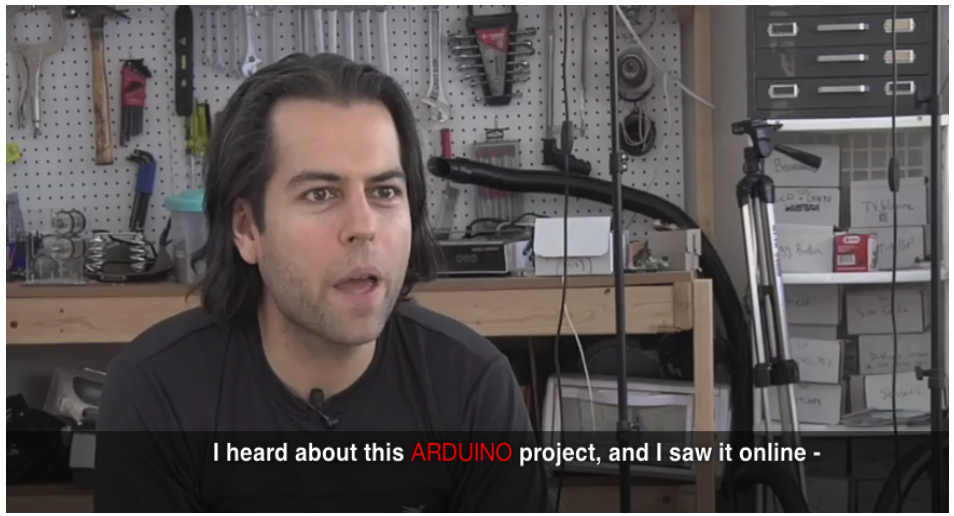
```
WEBVTT FILE

00:00:13.000 --> 00:00:16.100
I heard about this <c.arduino>arduino</c> project, and I saw it online -

00:00:16.100 --> 00:00:20.100
- and I said 'Wow! a lot of people are starting to talk about this. I should check it out!'
```

::cue pseudo-element

```
CSS ::cue .arduino {
  color: red;
  text-transform: uppercase;
  font-family: "Helvetica Neue";
  font-weight: lighter;
}
```



A WebVTT example file

Internationalization for subtitles

```
WEBVTT FILE

00:00:15.042 --> 00:00:18.042 D:vertical A:start
<ruby>左<rt>ひだり</rt></ruby>に<ruby> 見<rt>み</rt></ruby>えるのは…

00:00:18.750 --> 00:00:20.333 D:vertical A:start
<ruby>右<rt>みぎ</rt></ruby>に<ruby> 見<rt>み</rt></ruby>えるのは…

00:00:20.417 --> 00:00:21.917 D:vertical A:start
……首刈り機

00:00:22.000 --> 00:00:24.625 D:vertical A:start
すべて安全|完璧に安全だ
```

UTF-8 character encoding
ruby text
vertical / horizontal rendering
alignment start / middle / end

Text Track Captions and Subtitles in HTML5

Markup: kind="captions", kind="subtitles"

```
<video controls>
   <source src="video.mp4" type="video/mp4">
   <source src="video.webm" type="video/webm">
   <track label="English Captions" kind="captions"
          srclang="en" src="video_cc_en.vtt">
   <track label="English Subtitles" kind="captions"
          srclang="en" src="video_cc_en.vtt">
   <track label="German Subtitles" kind="subtitles"
          srclang="de" src="video_sub_de.vtt">
</video>
```

first implementations in Webkit

Audio Description example

Markup: kind="description"

```
WEBVTT FILE

1
00:00:00.000 --> 00:00:05.000
The orange open movie project presents

2
00:00:05.010 --> 00:0:12.000
Introductory titles are showing on the background of a water pool with fishes swimming and
mechanical objects lying on a stone floor.

3
00:00:12.010 --> 00:00:14.800
title: elephants dream
```

calculate length with average reading rate

demo (ChromeVox ARIA Live using JS Lib)

Navigation example

Markup: kind="chapters"

```
WEBVTT

Chapter1
00:00:00.000 --> 00:00:10.700
Title Slide

Chapter2
00:00:10.700 --> 00:00:47.600
Introduction by Naomi Black

Chapter3
00:00:47.600 --> 00:01:50.100
Impact of Captions on the Web
```

demo

JavaScript API examples

Turn on French subtitles

```
for(i=0; i < video.textTracks.length; i++) {
  if (textTracks[i].kind == "subtitles" && textTracks[i].language == "fr") {
    textTracks[i].track.mode = SHOWING;
  }
}
```

Register an event handler on all cue changes

```
video.textTrack[0].addEventListener("cuechange", function() {
  alert("A cue just started or ended.");
}, false);
```

Future of media on the Web

- WebVTT's simplicity makes it easy for desktop players and browsers to implement
- WebVTT's simplicity makes it easy for authors to create timed text and work with cues

New exciting video applications are possible and part of it will be more captions, subtitles and text descriptions

More on WebVTT?
http://www.youtube.com/watch?v=gK72pcu3cpk
(short: http://goo.gl/VFPFv)

The YouTube Captions API

YouTube Captions API

- One part of the larger YouTube Data API.
- REST-ful interface for creating, retrieving, updating, and deleting caption tracks using HTTP requests.
- Normal YouTube Data API access restrictions apply:
  - Authentication needed for modification and retrieval of caption tracks.
  - Developer key needed for all requests.

YouTube Captions API

Formats & Conversions

- There is support for submitting tracks in a number of formats:
  - RealText, SAMI, SubRip, SubViewer, etc.
  - ...or submit a block of text, and let us auto-sync.
- When requesting captions, the `fmt` parameter specifies which format to convert to.
  - `srt` (SubRip) and `sbv` (SubViewer) are currently supported conversions.

YouTube Captions API

Auto-synchronization

- Simplifies the process of adding captions.
- Uses speech recognition to generate timecodes automatically.
- Just upload a plain-text transcript using the API.
- English and Japanese are currently supported.

YouTube Captions API

Automatic Speech Recognition (ASR) Tracks

- On-demand availability via the API when authenticated as the owner.
- Uses speech recognition to generate both caption text and timecodes.
- Identified by the `yt:derived` tag in the entry captions response entry.
- English and Japanese are currently supported.

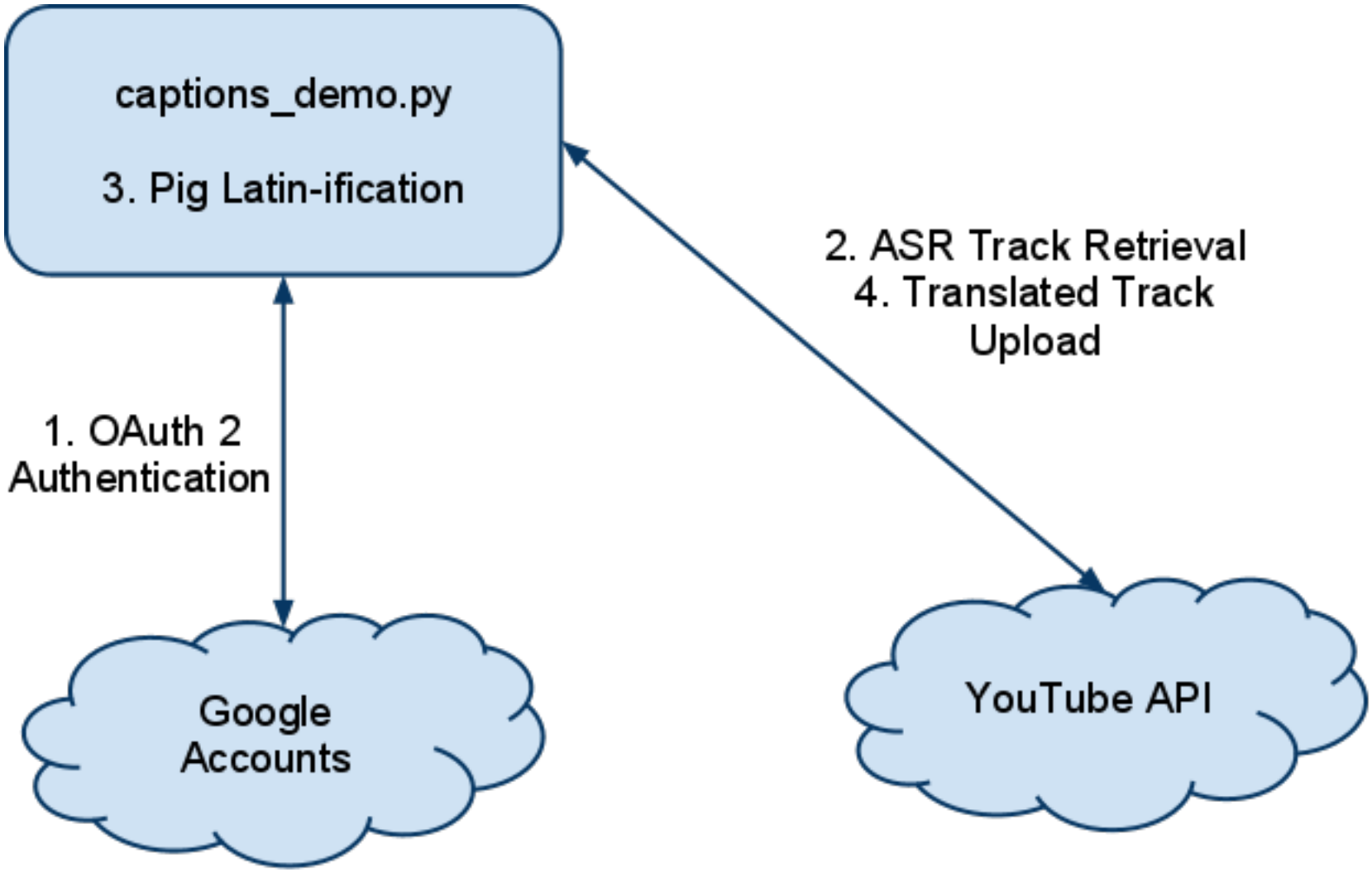
YouTube Captions API

Demo Code

- Given a YouTube video ID, retrieves the ASR caption track for that video, translates it into Pig Latin, and uploads the translation.
- Python, command line code.
  - Uses the new Google API Python Client library for authenticated HTTP requests.
- Silly example, but intended to illustrate how you could retrieve, process, and upload your own caption tracks for a more meaningful purpose.

YouTube Captions API

Demo Code



YouTube Captions API

Demo Code

Sample Video:
http://www.youtube.com/watch?v=gWUTT-uMftM

YouTube Captions API

Demo Code

http://code.google.com/p/gdata-samples/source/browse/trunk/gdata/captions_demo.py

(or http://goo.gl/Vp5WE)

YouTube Captions API

Demo Code — GetAsrTrackUrl()

```
url = self.CAPTIONS_URL_FORMAT % self.video_id

response_headers, body = self.http.request(url,
  "GET", headers=self.headers)

if response_headers["status"] == "200":
  json_response = json.loads(body)
  for entry in json_response["feed"]["entry"]:
    if ("yt$derived" in entry and
        entry["yt$derived"]["$t"] == "speechRecognition" and
        entry["content"]["xml$lang"] == "en"):
      # This will only be set for the ASR track.
      self.track_url = entry["content"]["src"]
```

YouTube Captions API

Demo Code

```
// Snip...
"content": {
 "type": "application/vnd.youtube.timedtext",
 "src": CAPTION_TRACK_URL,
 "xml$lang": "en"
},
"yt$derived": {
 "$t": "speechRecognition"
}
// Snip...
```

YouTube Captions API

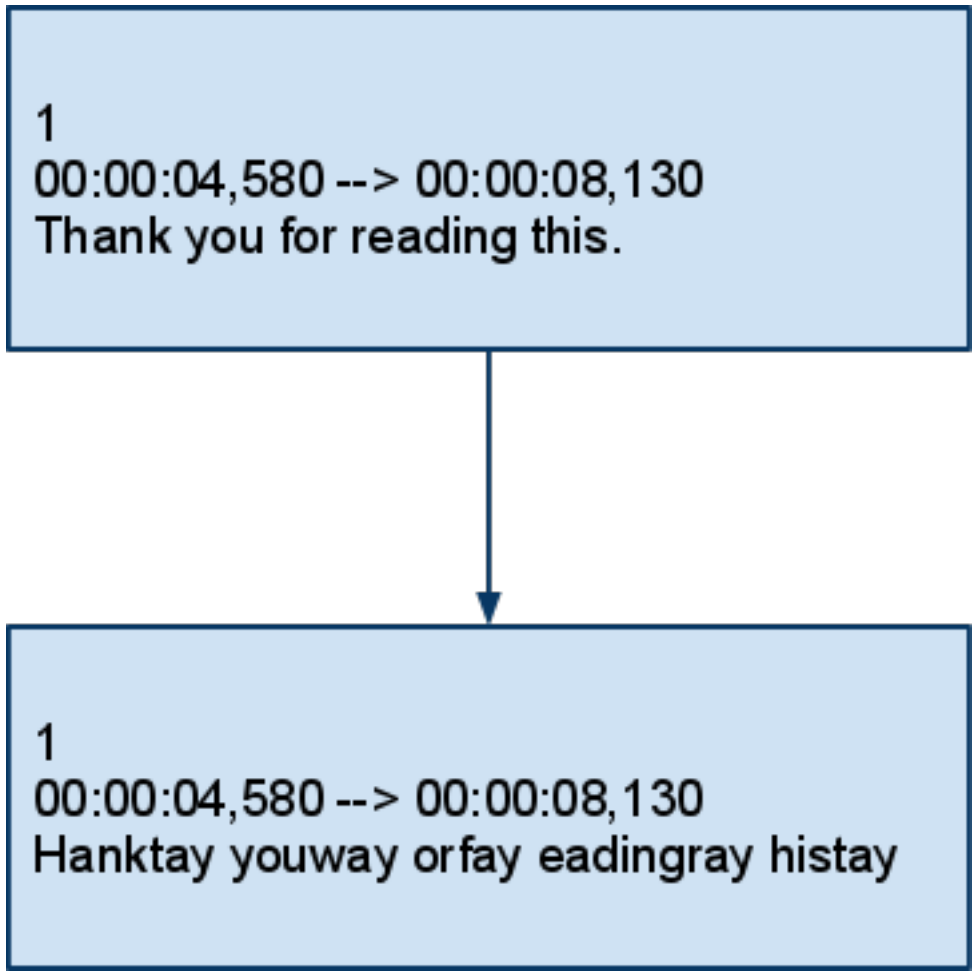
Demo Code — GetSrtCaptions()

```
response_headers, body = self.http.request(
  "%s?fmt=srt" % self.track_url,
  "GET", headers=self.headers)

if response_headers["status"] == "200":
  self.srt_captions = SubRipFile.from_string(body)
```

YouTube Captions API

Demo Code



YouTube Captions API

Demo Code — UploadTranslatedCaptions()

```
self.headers["Content-Type"] =
  self.CAPTIONS_CONTENT_TYPE
self.headers["Content-Language"] =
  self.CAPTIONS_LANGUAGE_CODE
self.headers["Slug"] = self.CAPTIONS_TITLE

url = self.CAPTIONS_URL_FORMAT % self.video_id

response_headers, body = self.http.request(
  url, "POST", body=self.translated_captions_body,
  headers=self.headers)
```

YouTube Captions API

Demo Site

http://yt-captions-uploader.appspot.com/

Java App Engine source at http://code.google.com/p/youtube-captions-uploader/

Q & A

Hashtags: #io2011 #youtube

Feedback: http://goo.gl/NLAu0

Links & Reference

- Real-time Caption Gadget Code
  - http://code.google.com/p/io-captions-gadget/
- Caption Uploader Code and Working Demo
  - http://code.google.com/p/youtube-captions-uploader/
- YouTube Caption API
  - http://code.google.com/apis/youtube/2.0/developers_guide_protocol_captions.html
- WebVTT specification:
  - http://www.whatwg.org/specs/web-apps/current-work/webvtt.html
  - http://youtu.be/gK72pcu3cpk


Google™
11
I/O