Humix

COMMUNITY MEETUP #2 2015/11/26

Part 1

Overview

TTS and Speech Recognition

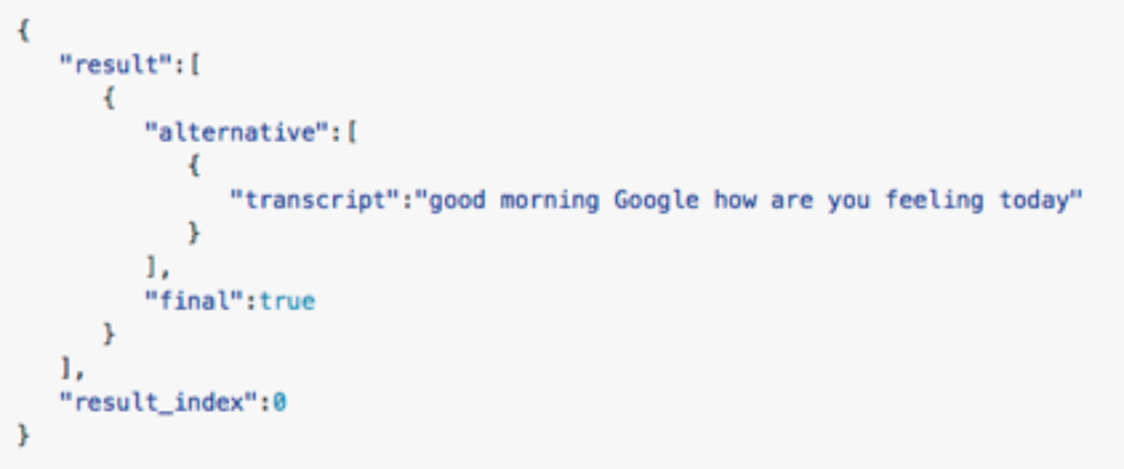
what is it ?

Google Speech Recognition and TTS

REQUEST

```
curl -X POST \
--data-binary @'audio/hello (16bit PCM).wav' \
--header 'Content-Type: audio/l16; rate=16000;' \
'https://www.google.com/speech-api/v2/recognize?output=json&lang=en-us&key=yourkey'
```

RESPONSE

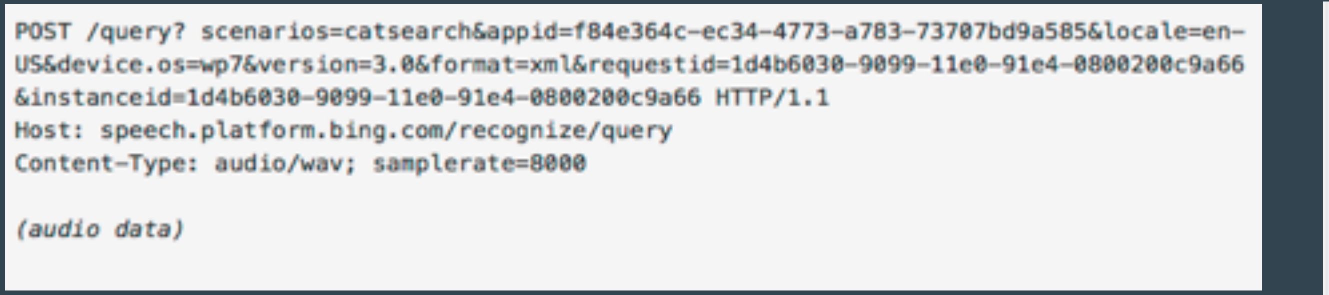


```
{
   "result":[
      {
         "alternative":[
            {
               "transcript":"good morning Google how are you feeling today"
            }
         ],
         "final":true
      }
   ],
   "result_index":0
}
```

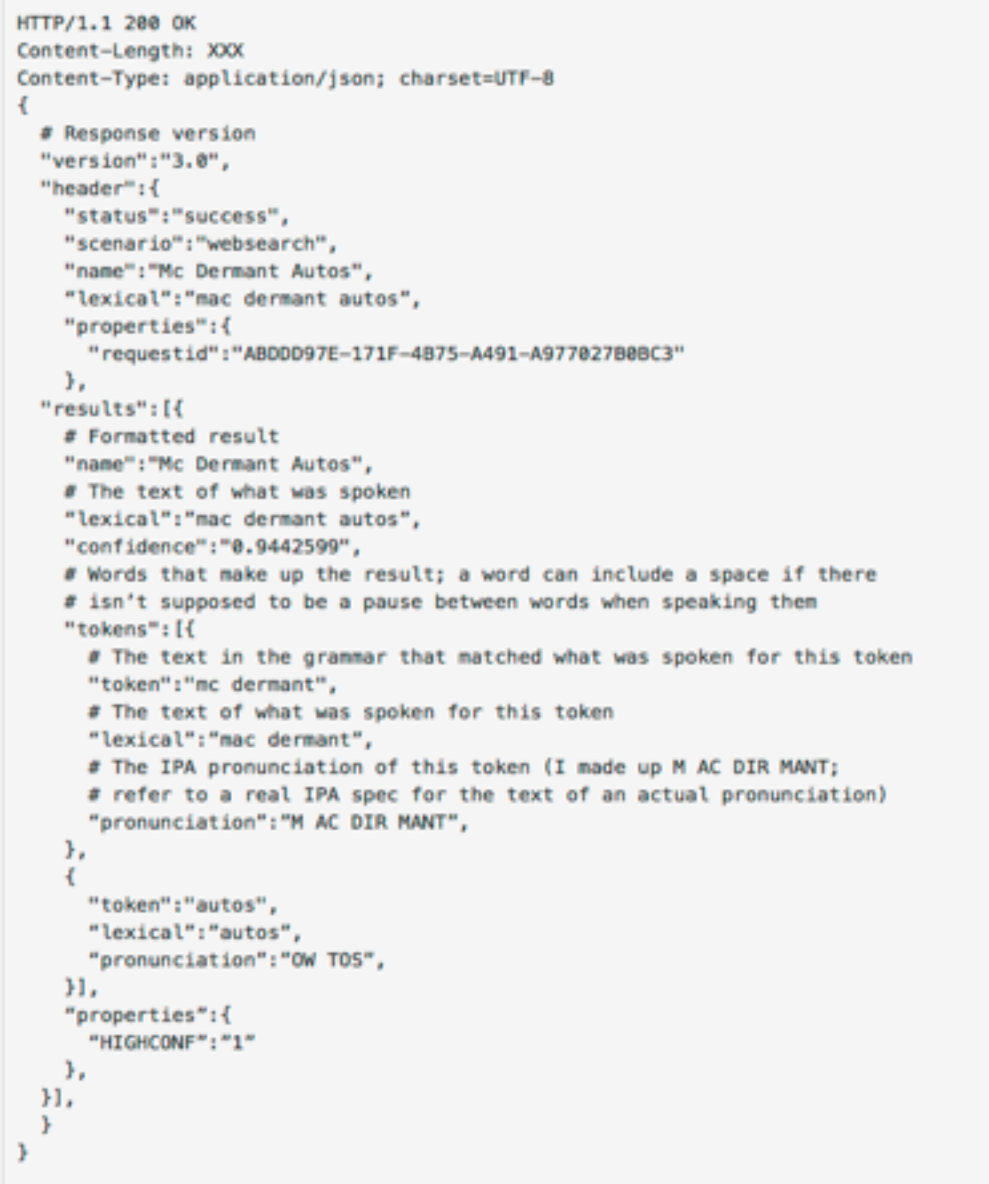
Try TTS

http://translate.google.com/translate_tts?ie=utf-8&tl=zh&q=Hello

MS Speech Recognition Service



```
POST /query? scenarios=catsearch&appid=f84e364c-ec34-4773-a783-73707bd9a585&locale=en-
US&device.os=wp7&version=3.0&format=xml&requestid=1d4b6030-9099-11e0-91e4-0800200c9a66
&instanceid=1d4b6030-9099-11e0-91e4-0800200c9a66 HTTP/1.1
Host: speech.platform.bing.com/recognize/query
Content-Type: audio/wav; samplerate=8000

(audio data)
```

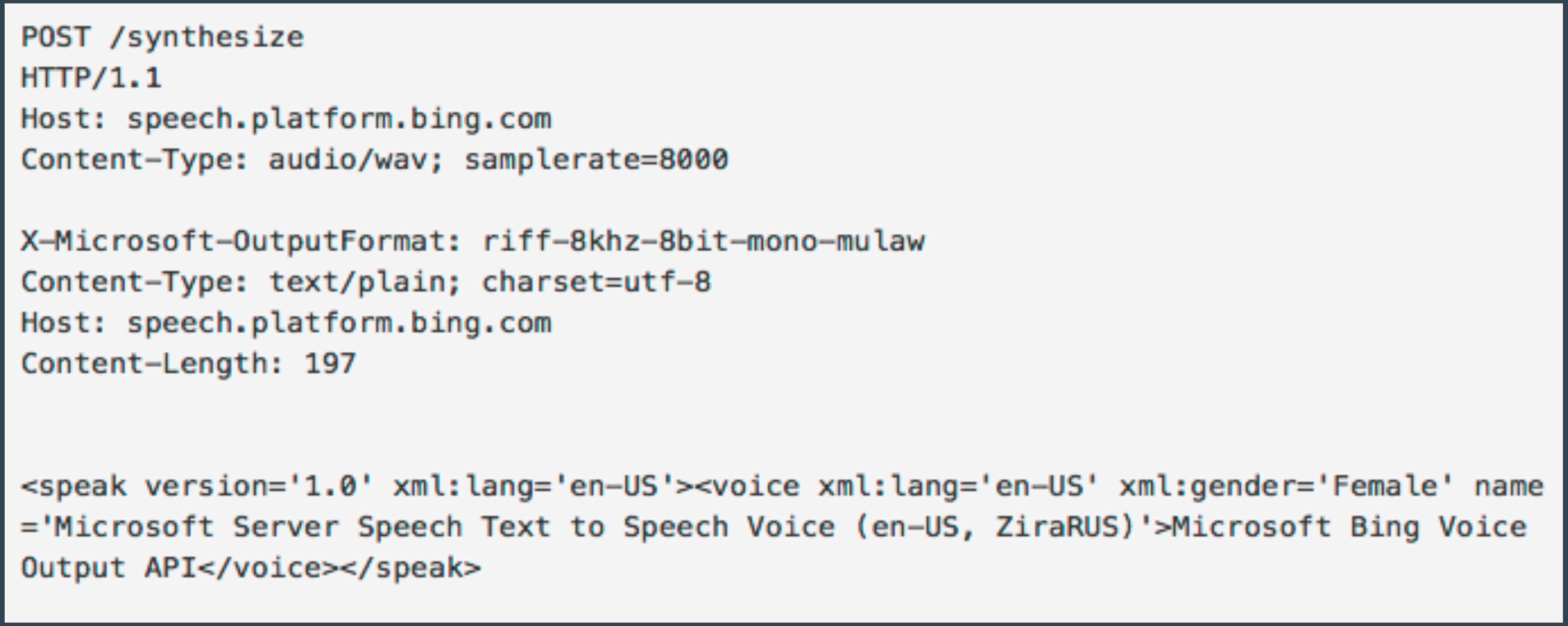


```
HTTP/1.1 200 OK
Content-Length: XXX
Content-Type: application/json; charset=UTF-8
{
  # Response version
  "version":"3.0",
  "header":{
    "status":"success",
    "scenario":"websearch",
    "name":"Mc Dermant Autos",
    "lexical":"mac dermant autos",
    "properties":{
      "requestid":"ABDDD97E-171F-4875-A491-A977027B0BC3"
    },
  "results":[{
    # Formatted result
    "name":"Mc Dermant Autos",
    # The text of what was spoken
    "lexical":"mac dermant autos",
    "confidence":"0.9442599",
    # Words that make up the result; a word can include a space if there
    # isn't supposed to be a pause between words when speaking them
    "tokens":[{
      # The text in the grammar that matched what was spoken for this token
      "token":"mc dermant",
      # The text of what was spoken for this token
      "lexical":"mac dermant",
      # The IPA pronunciation of this token (I made up M AC DIR MANT;
      # refer to a real IPA spec for the text of an actual pronunciation)
      "pronunciation":"M AC DIR MANT",
    },
    {
      "token":"autos",
      "lexical":"autos",
      "pronunciation":"OW TOS",
    }],
    "properties":{
      "HIGHCONF":"1"
    },
  }],
  }
}
```

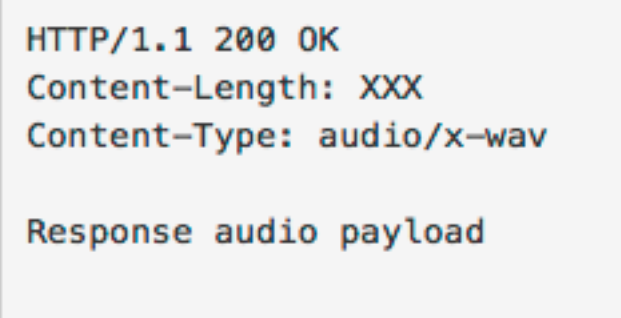
https://www.projectoxford.ai/doc/speech/overview

MS TTS Service

https://speech.platform.bing.com/synthesize



```
POST /synthesize
HTTP/1.1
Host: speech.platform.bing.com
Content-Type: audio/wav; samplerate=8000

X-Microsoft-OutputFormat: riff-8khz-8bit-mono-mulaw
Content-Type: text/plain; charset=utf-8
Host: speech.platform.bing.com
Content-Length: 197


<speak version='1.0' xml:lang='en-US'><voice xml:lang='en-US' xml:gender='Female' name
='Microsoft Server Speech Text to Speech Voice (en-US, ZiraRUS)'>Microsoft Bing Voice
Output API</voice></speak>
```



```
HTTP/1.1 200 OK
Content-Length: XXX
Content-Type: audio/x-wav

Response audio payload
```

Speech Services available today

NUANCE

http://www.nuance.com/for-developers/dragon/index.htm

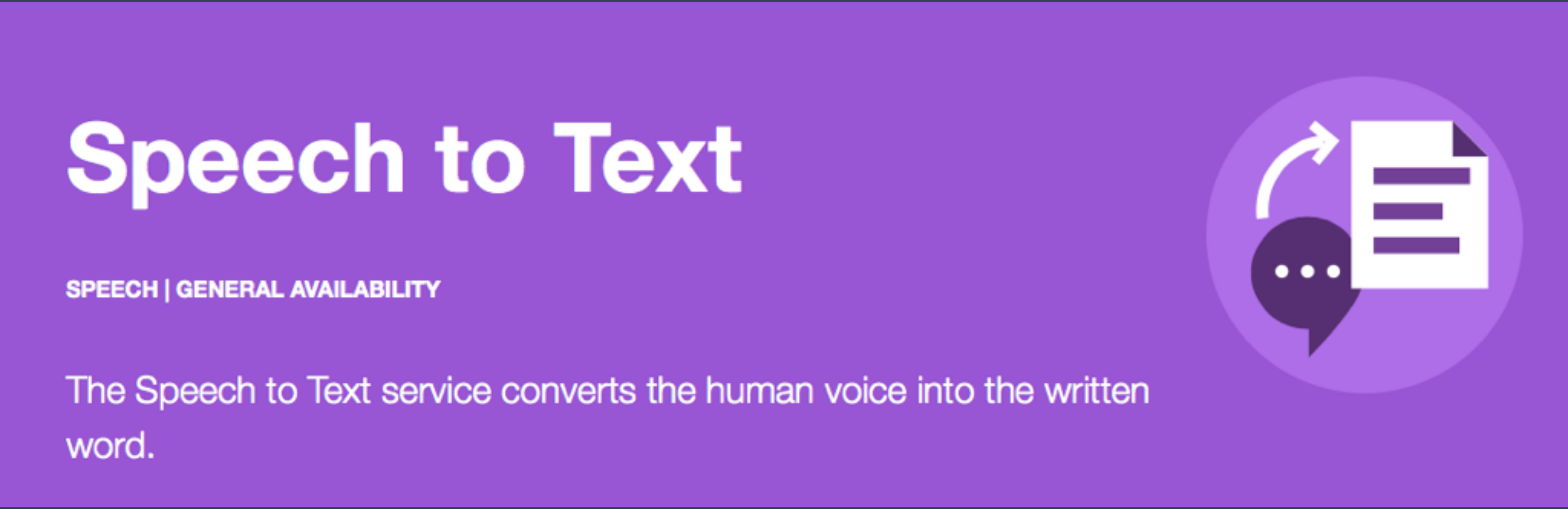
iSpeech

http://www.ispeech.org/#/home

http://tts.itri.org.tw

Part 2

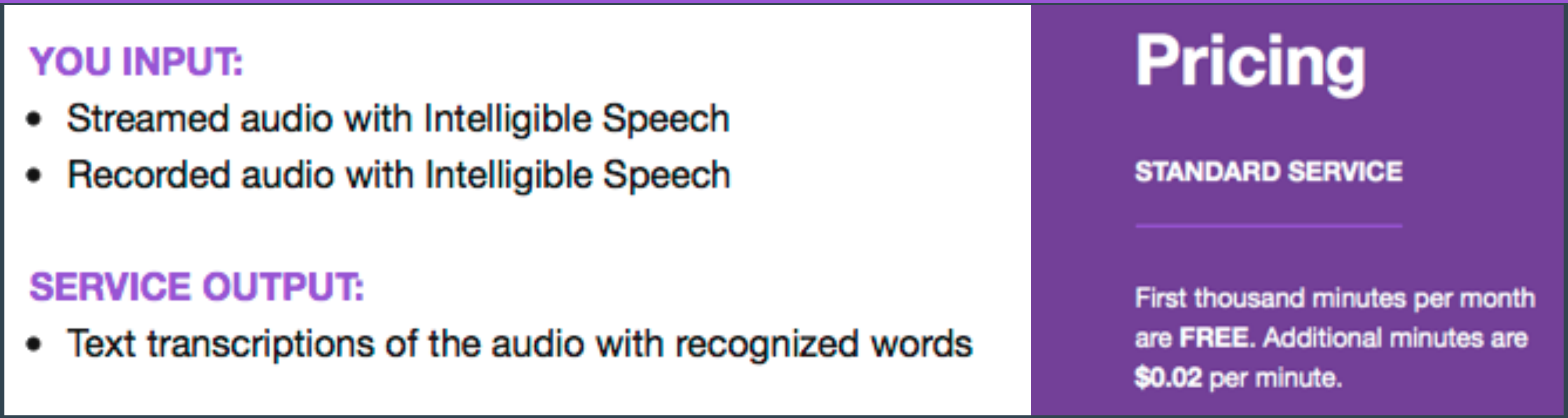
Watson TTS & Speech



Speech to Text

SPEECH | GENERAL AVAILABILITY

The Speech to Text service converts the human voice into the written word.



YOU INPUT:

- Streamed audio with Intelligible Speech
- Recorded audio with Intelligible Speech

SERVICE OUTPUT:

- Text transcriptions of the audio with recognized words

Pricing

STANDARD SERVICE

First thousand minutes per month are **FREE**. Additional minutes are **$0.02** per minute.

1.3 min / hour

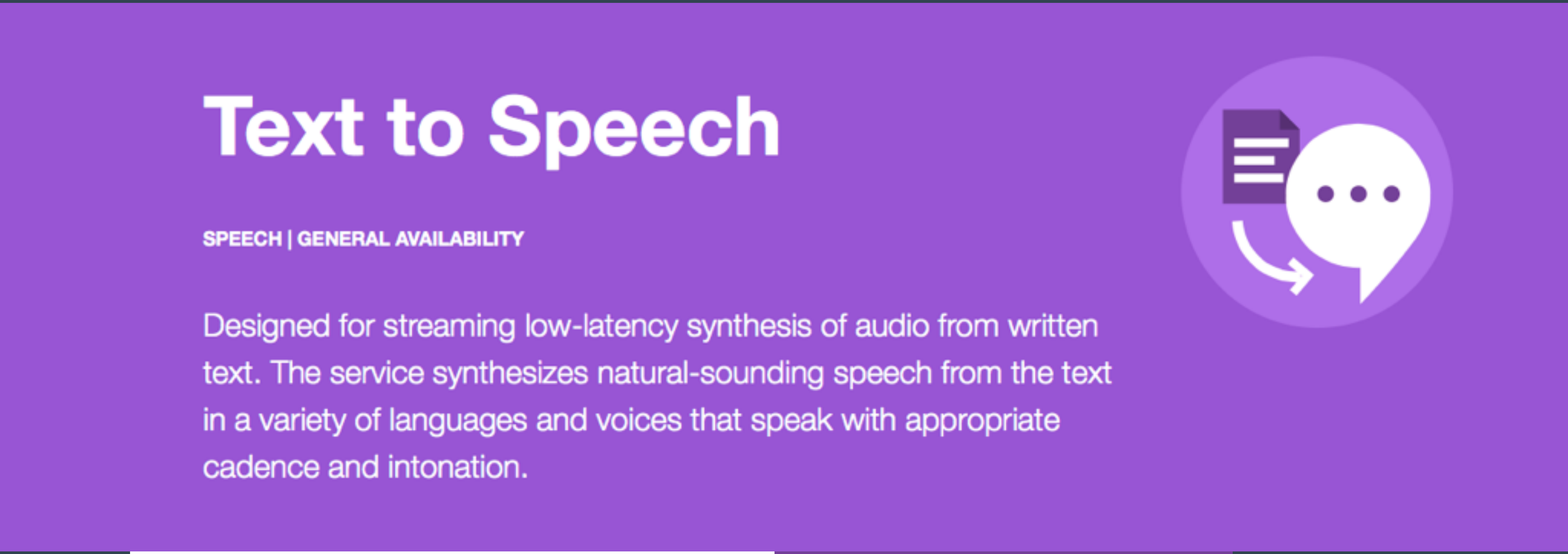
https://youtu.be/JWnLgZ58zsw

Documentation

http://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/speech-to-text/api/v1/#introduction

Access Endpoint

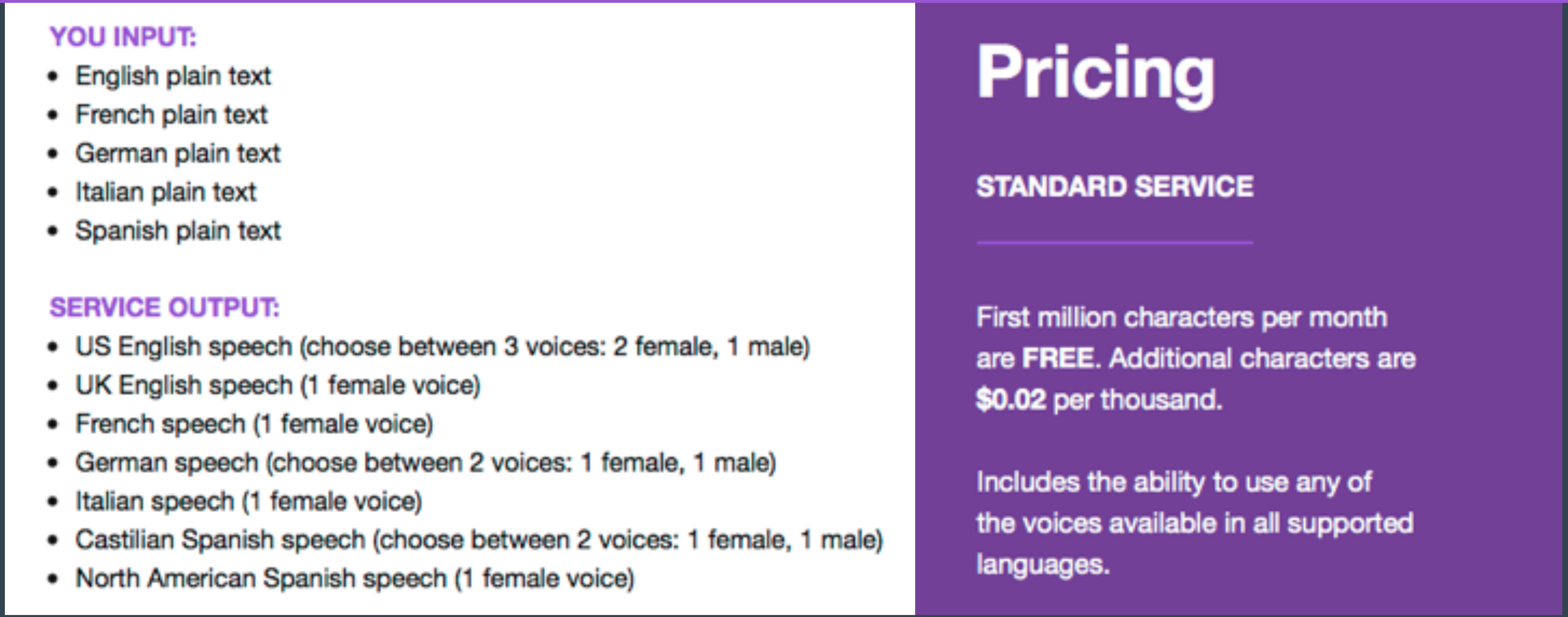
http://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/speech-to-text/api/v1/



Text to Speech

SPEECH | GENERAL AVAILABILITY

Designed for streaming low-latency synthesis of audio from written text. The service synthesizes natural-sounding speech from the text in a variety of languages and voices that speak with appropriate cadence and intonation.



YOU INPUT:

- English plain text
- French plain text
- German plain text
- Italian plain text
- Spanish plain text

SERVICE OUTPUT:

- US English speech (choose between 3 voices: 2 female, 1 male)
- UK English speech (1 female voice)
- French speech (1 female voice)
- German speech (choose between 2 voices: 1 female, 1 male)
- Italian speech (1 female voice)
- Castilian Spanish speech (choose between 2 voices: 1 female, 1 male)
- North American Spanish speech (1 female voice)

Pricing

STANDARD SERVICE

First million characters per month are **FREE**. Additional characters are **$0.02** per thousand.

Includes the ability to use any of the voices available in all supported languages.

23 char / min

Documentation

http://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/doc/text-to-speech/

Access Endpoint

https://stream.watsonplatform.net/text-to-speech/api

Part 3

Wit.ai for Natural Language Processing

https://wit.ai

Natural Language for Developers

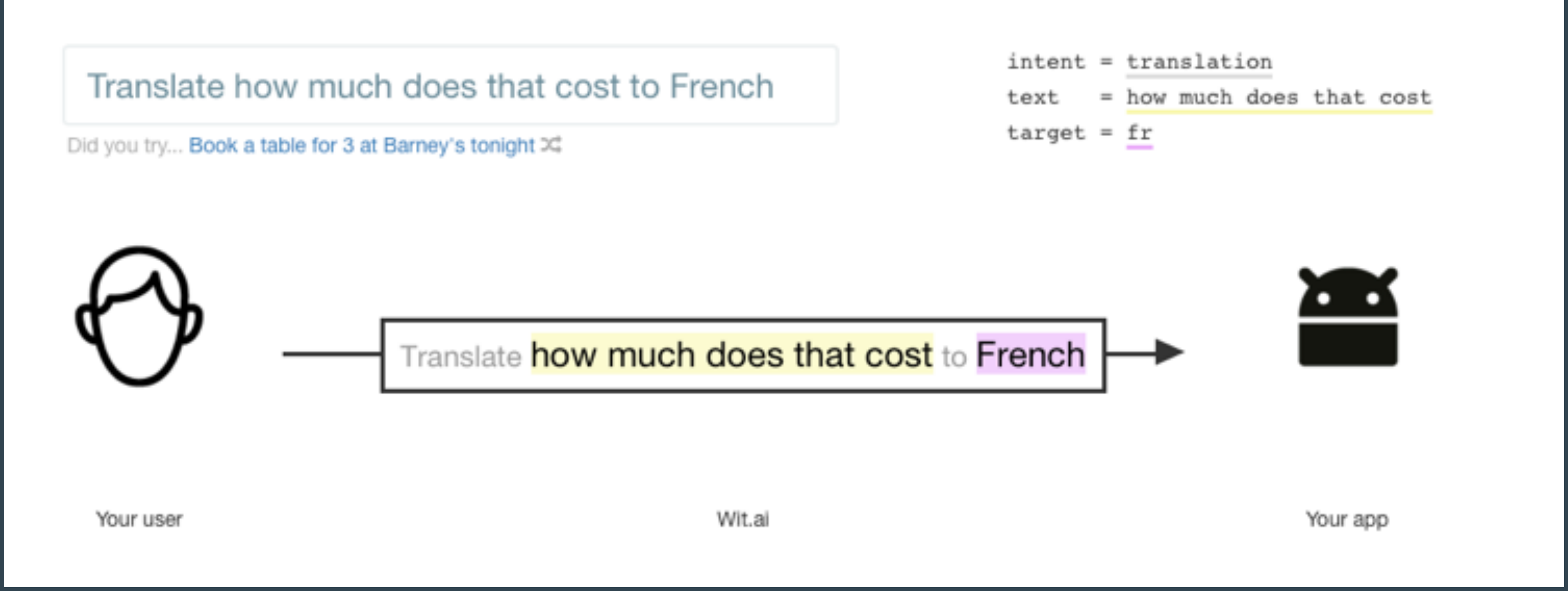
Mobile apps

Home automation

Wearable devices

Robots

Messenger Agents



Translate how much does that cost to French

Did you try... Book a table for 3 at Barney's tonight

```
intent = translation
text   = how much does that cost
target = fr
```

WIT.AI DEMO

Part 4

Humix TTS Module

Humix In Action

Join US

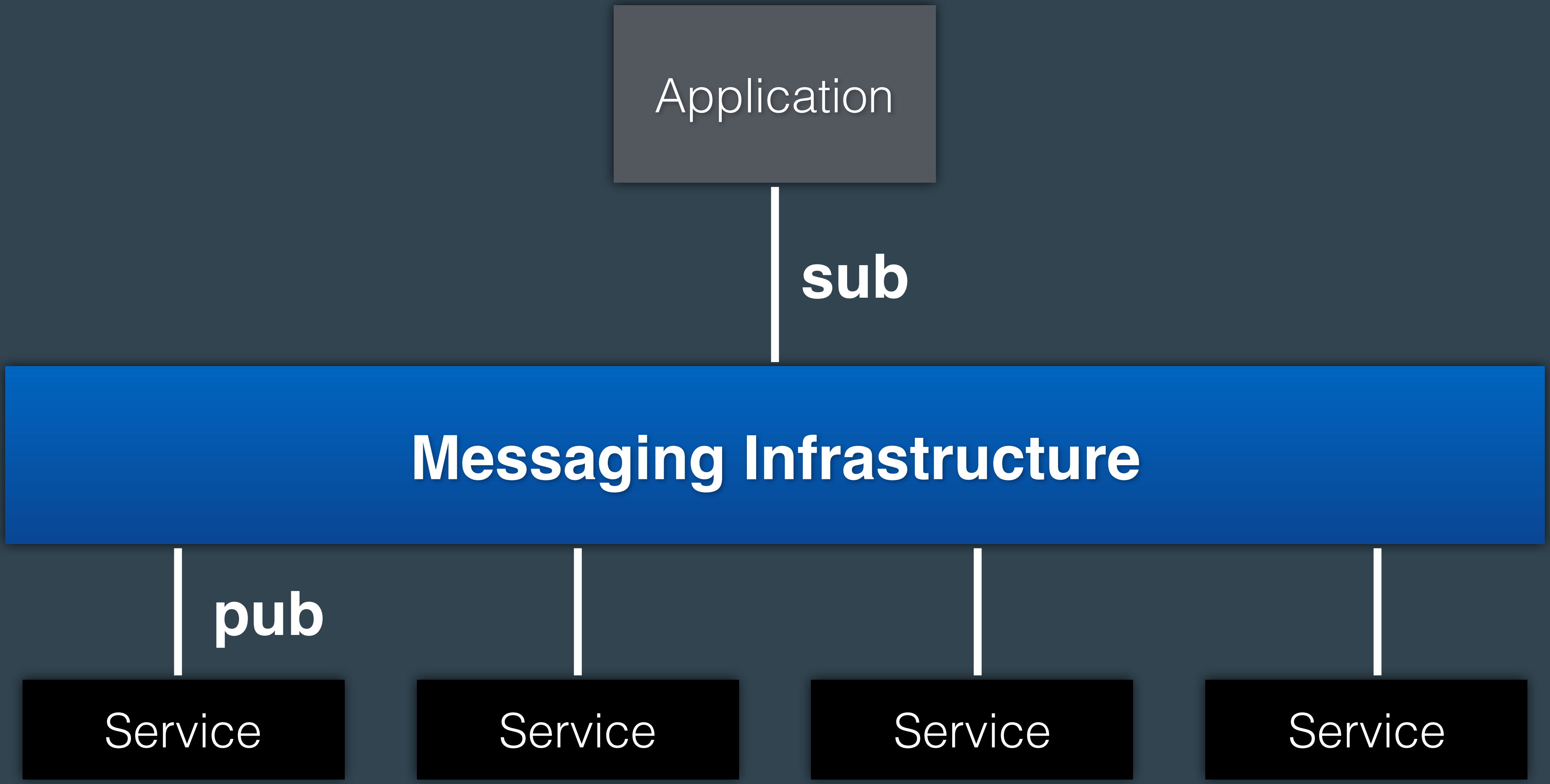
https://www.facebook.com/groups/1513552855636723/

Reference

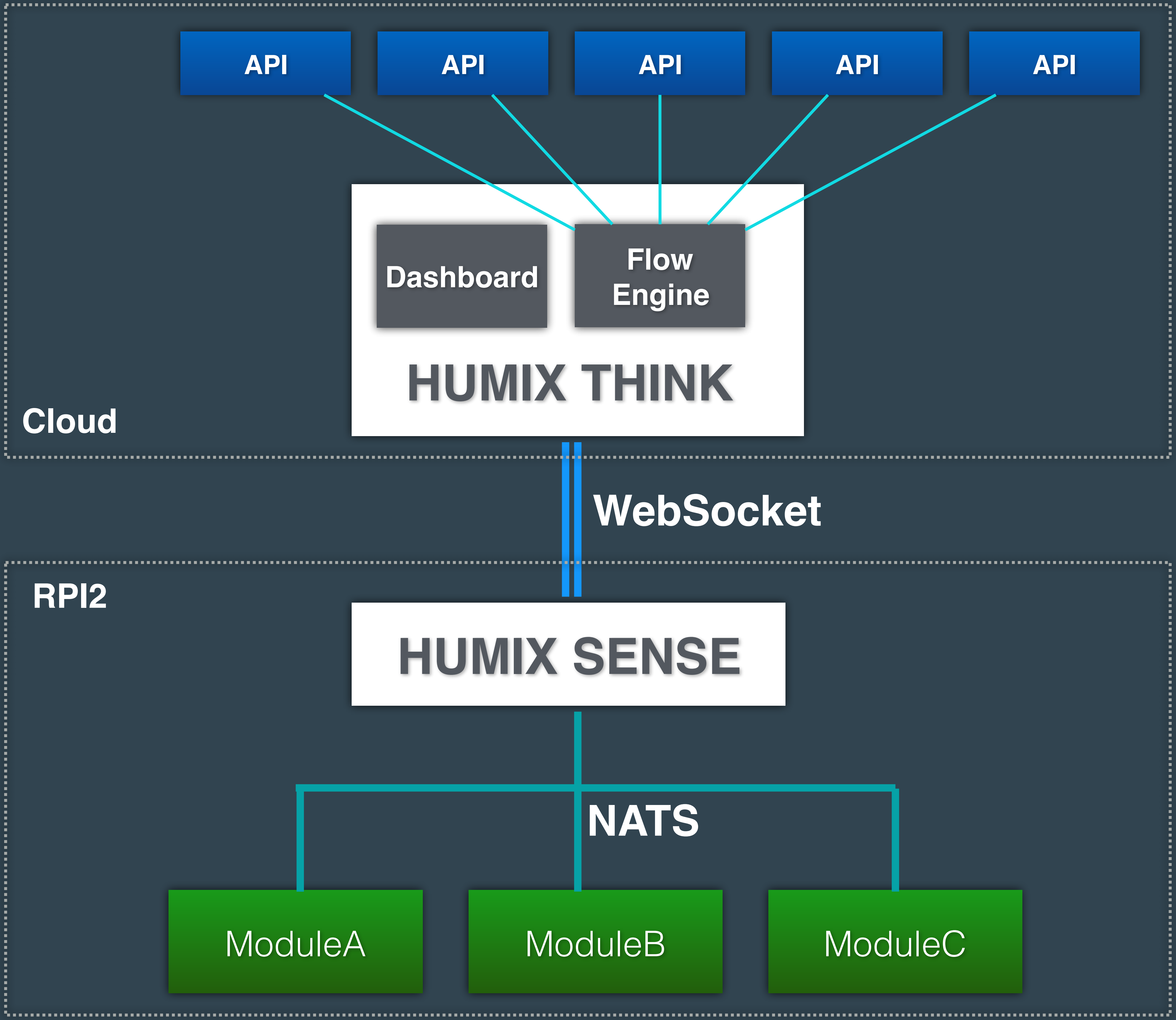
Humix Architecture

MicroService Architecture

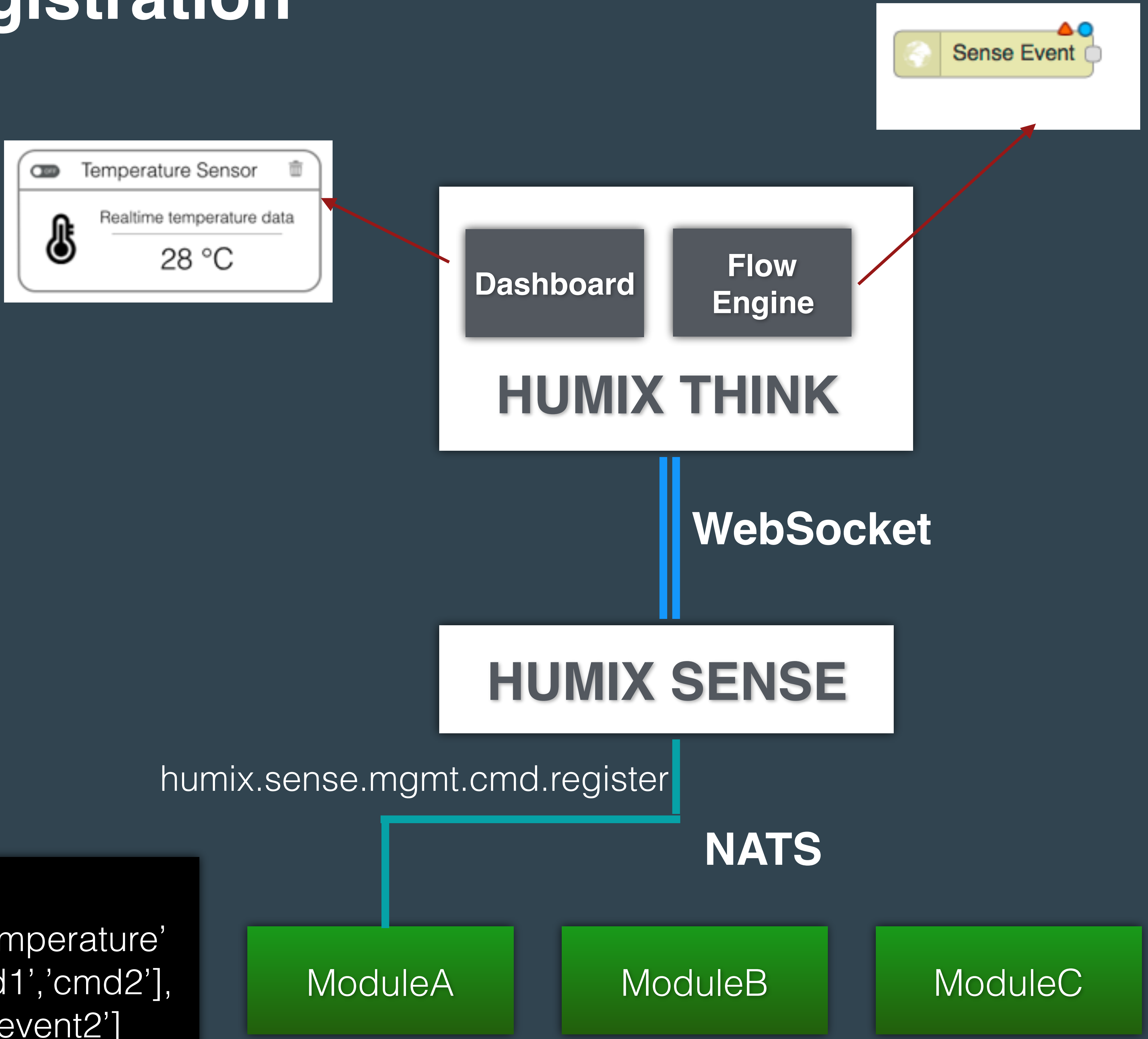
“An approach to developing a single application as a suite of small, independently deployable services”



Humix Architecture

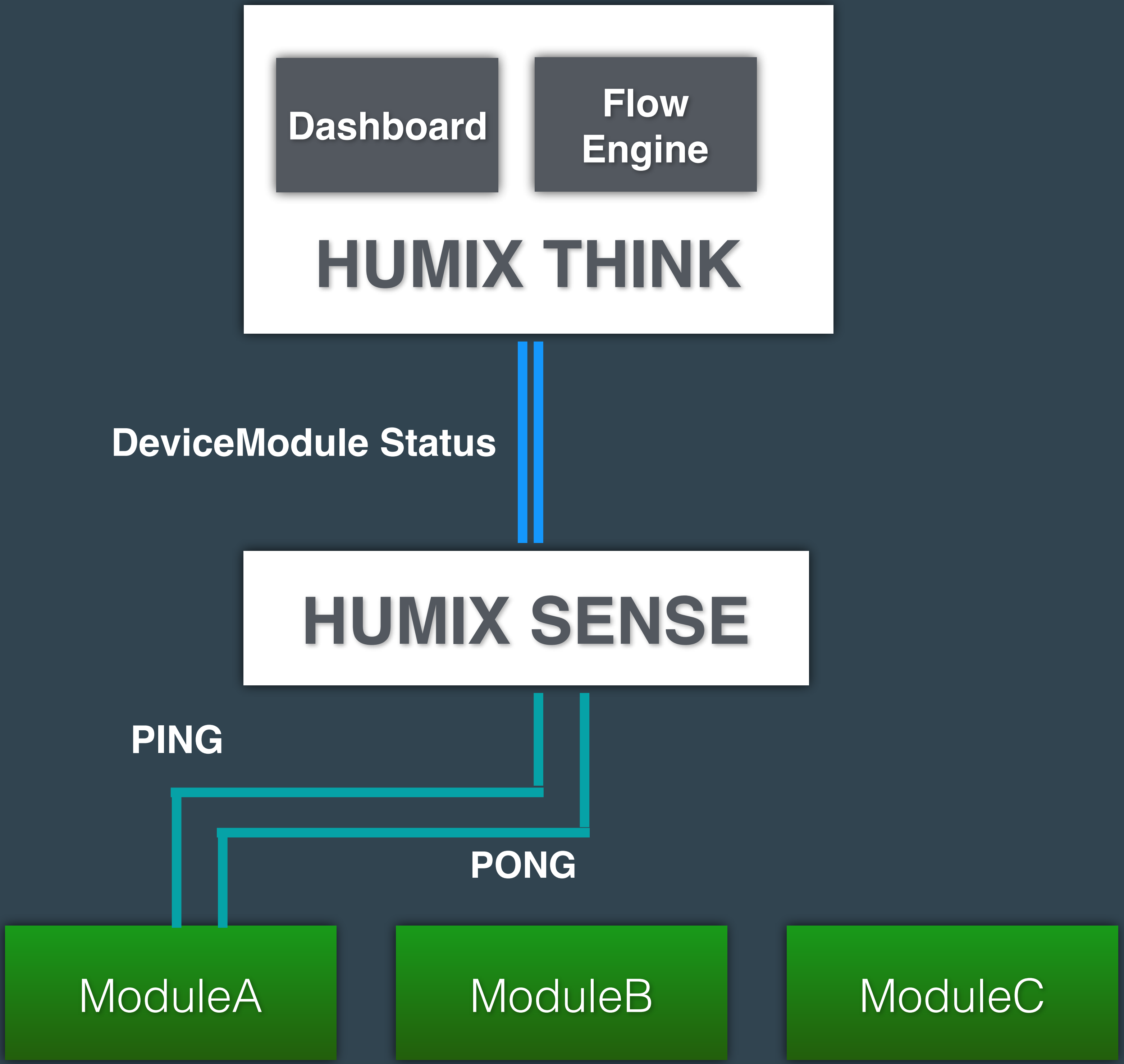


Module Registration

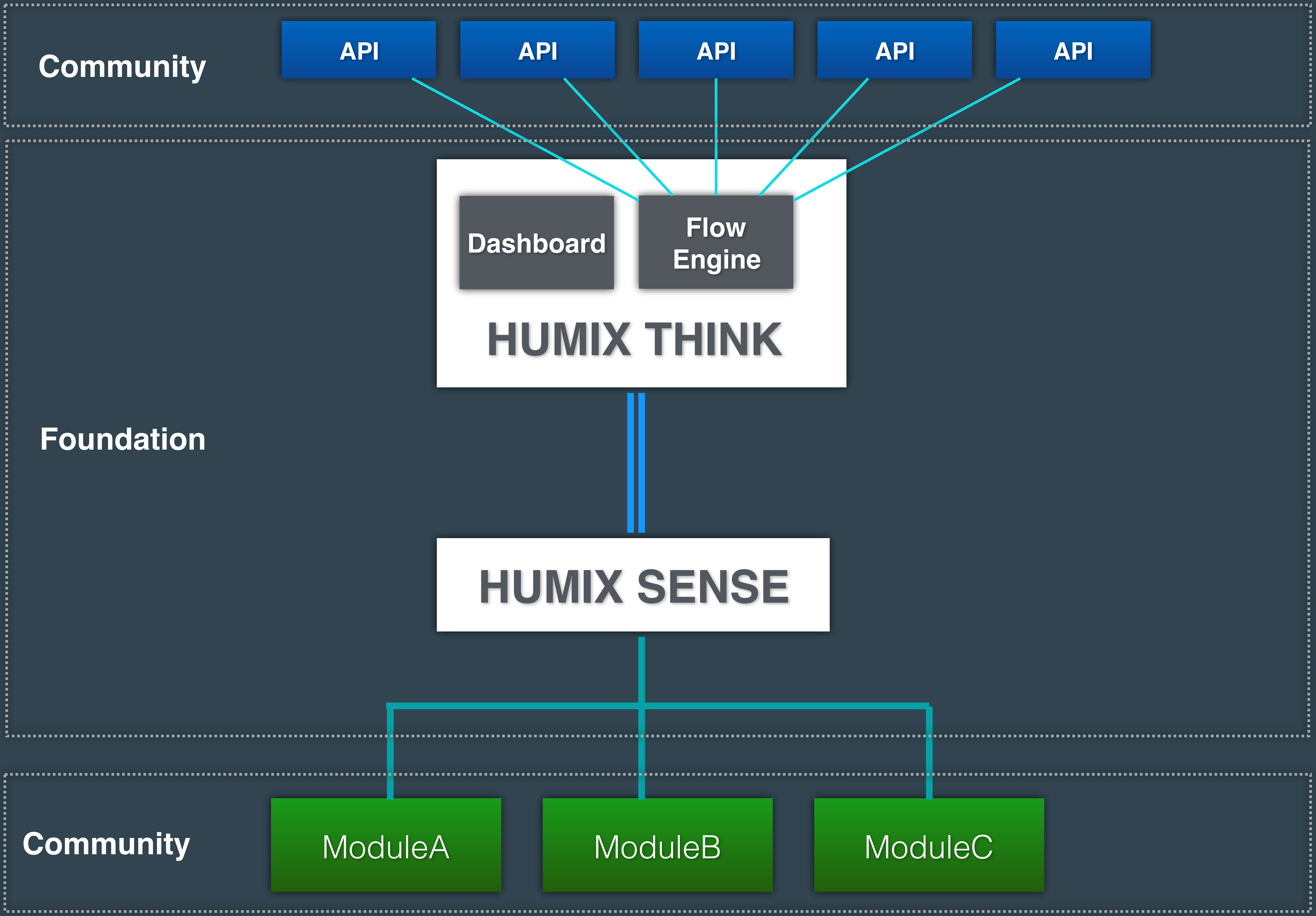


```
{
    moduleName: ‘Temperature’
    commands: [‘cmd1’,’cmd2’],
    events: [‘event1’,’event2’]
}
```

Module Status Check

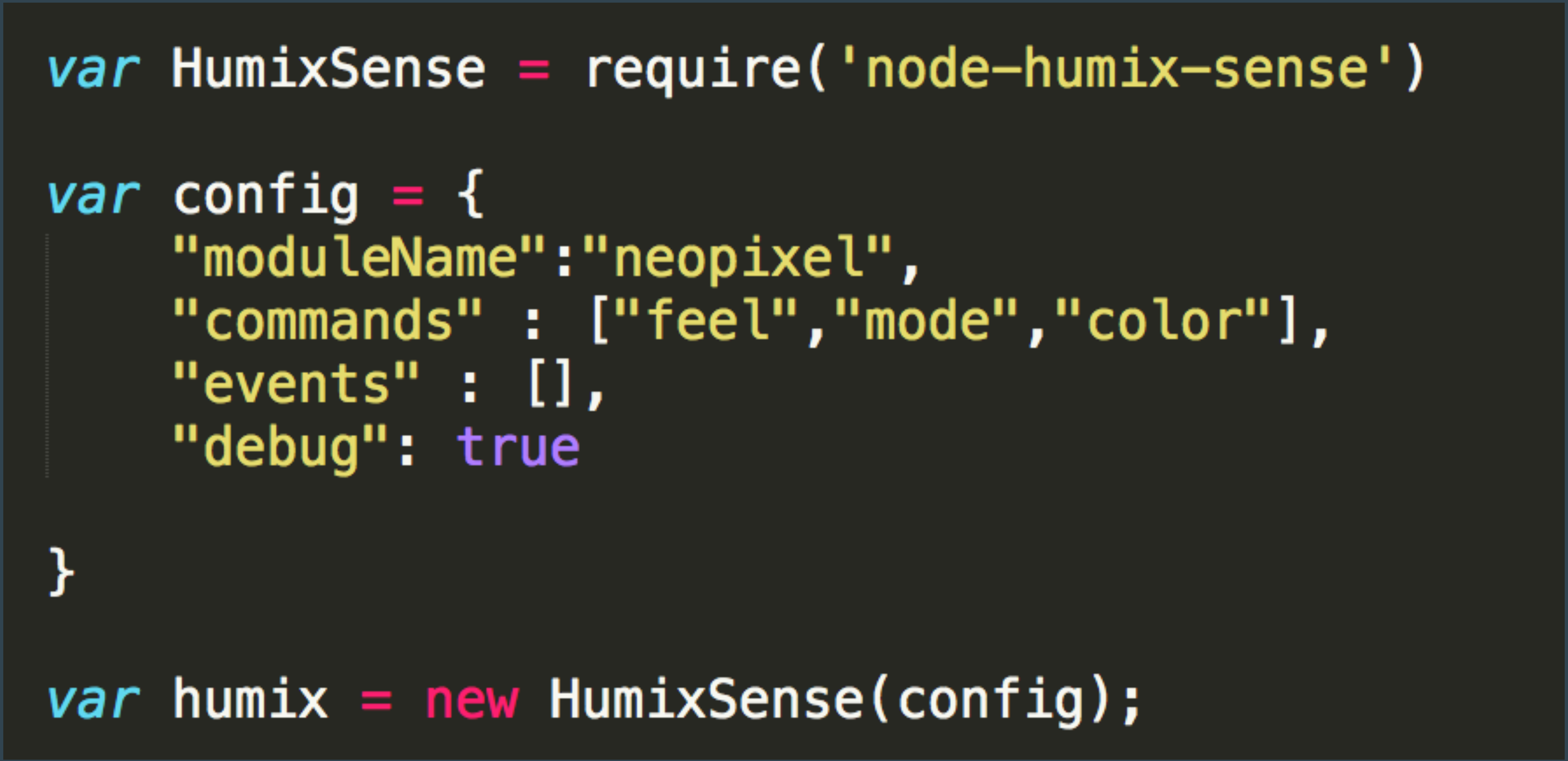


Humix Architecture



Programming Model

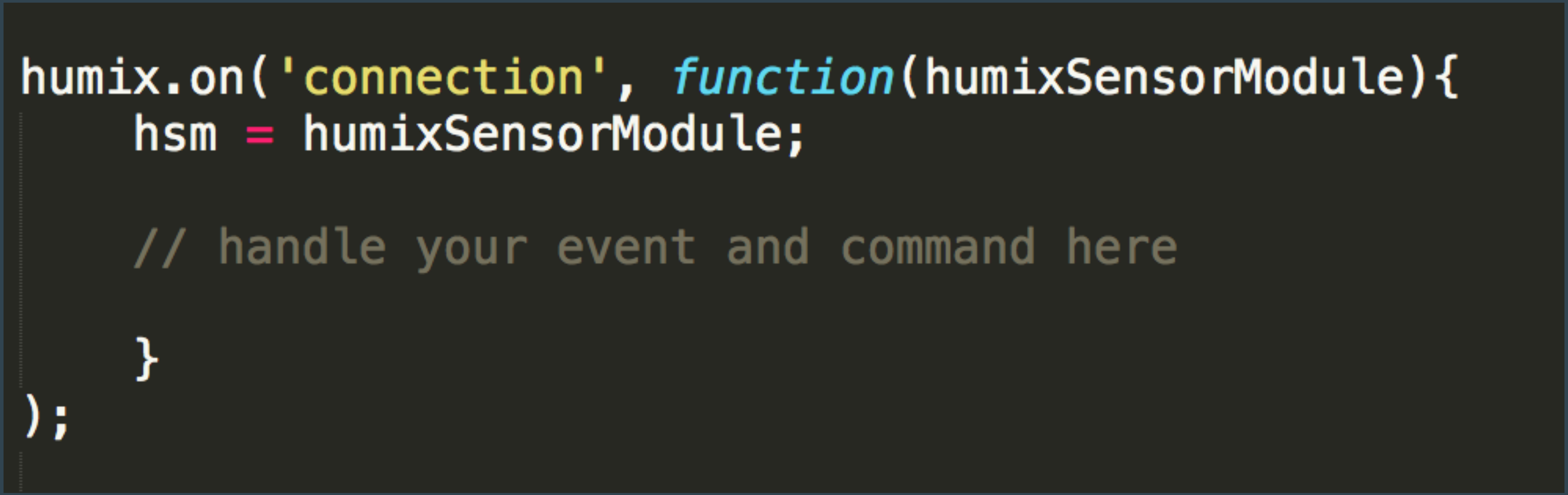
1. Require Humix-Sense and provide module config



```
var HumixSense = require('node-humix-sense')

var config = {
    "moduleName":"neopixel",
    "commands" : ["feel","mode","color"],
    "events" : [],
    "debug": true

}

var humix = new HumixSense(config);
```

Programming Model

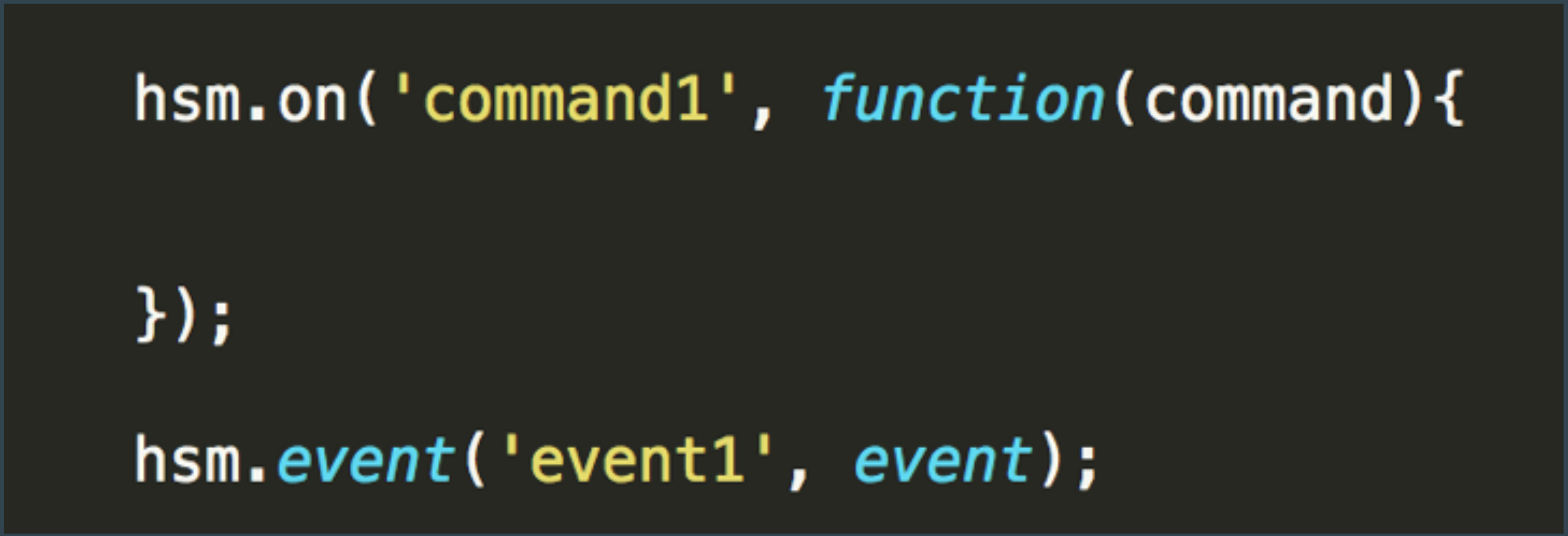
2. Connect to “Think”



```
humix.on('connection', function(humixSensorModule){
    hsm = humixSensorModule;

    // handle your event and command here

    }
);
```

Programming Model

3. Process “commands”, send “events” on device



```
hsm.on('command1', function(command){


});

hsm.event('event1', event);
```