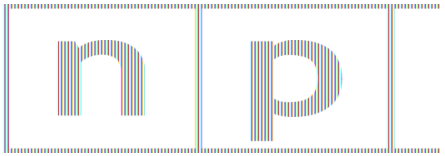


Nara Kasbergen (@xiehan) | jsDay Italy | May 10, 2018

Finding Your Voice:

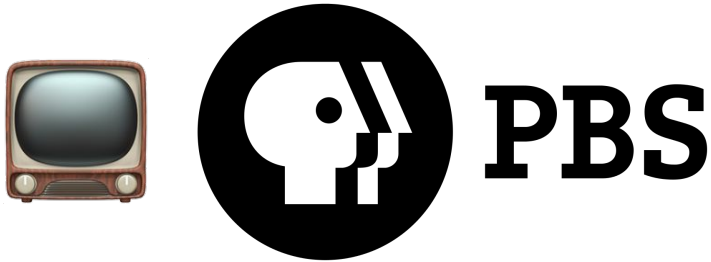
Building Screenless Interfaces with Node.js

Who am I?

- Senior full-stack web developer
- At NPR since March 2014
- Part of a 5-member skunkworks team focused 100% on voice UI development
  - Formed in September 2017

What is NPR?

A quick explainer for the Italians in the audience:



Why voice UI development?

Then:

Now:

Hey Siri, tune to NPR

“Alexa, what's in the news?”

Ok Google, play the latest news from NPR

“smart speakers”


“smart speakers”

“voice assistants”

How Amazon views Alexa

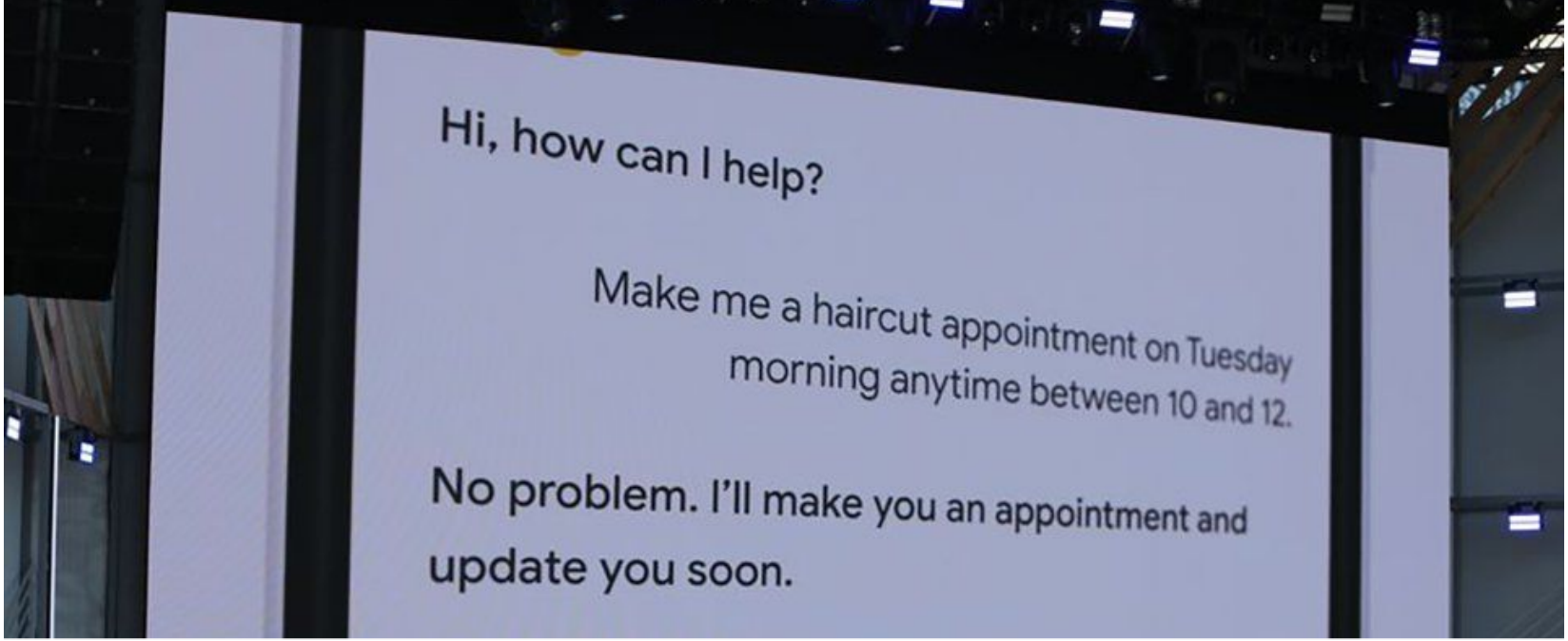
“Alexa, set the thermostat to 25 degrees.”

“Okay.”

“I'd like to reorder paper towels please.”

“Alexa, thank you!”

“No problem.”


Hi, how can I help?
Make me a haircut appointment on Tuesday morning anytime between 10 and 12.
No problem. I'll make you an appointment and update you soon.


User
"Get me an appointment at..."
Google Assistant
Google Duplex
Call
Business
"You're all set!"

“What would you want to know about voice UI development?”

What can I actually make?

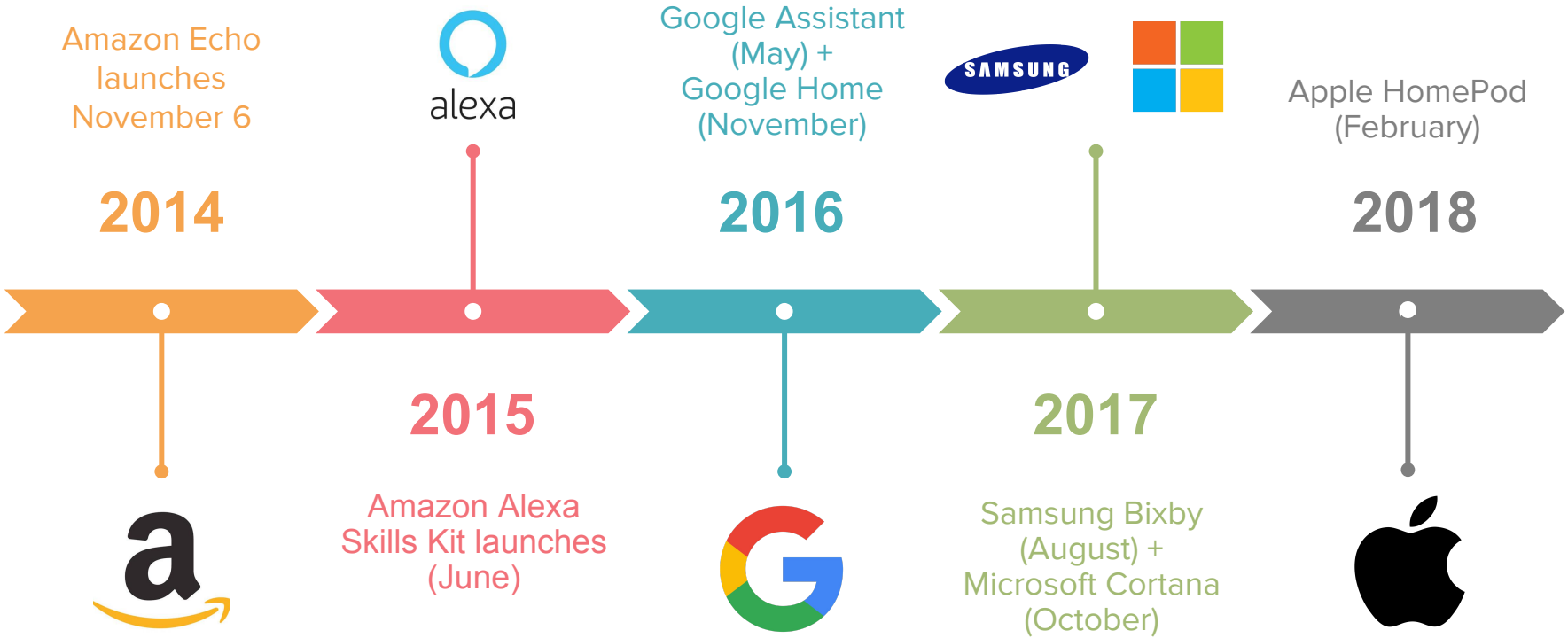
Is it possible to build one app for Amazon Echo, Google Home, and Apple HomePod?

1.

What can you actually make?

To understand the present,
we must understand the past.

A brief timeline of voice assistants



A natural evolution

1. add voice activation to existing custom app ecosystem

2. add content via RSS feeds

3. add support for custom “skills”

A natural evolution

1. add voice activation to existing custom app ecosystem

2. add content via RSS feeds

3. add support for custom “skills”

Conclusions

- Amazon has a 2-year lead
- Only Amazon and Google have fully developed ecosystems
- A big focus is adding access to news and podcasts via RSS
- Home automation is secondary

2.

Can you build one “skill” to rule them all?

tl;dr yes … and no

Alexa + Google ecosystems

- Heavily leverage their existing cloud infrastructure
  - AWS Lambda + Google Cloud Functions
- Can also build a traditional REST API accessed by their services

The request/response flow

The request/response flow

The future is “serverless”

- Others can speak more eloquently on this subject than me
  - Hopefully you went to Luciano's talk
- Let's just assume we want to use Lambda or Cloud Functions...
- ... node.js wins!

The official SDKs are not bad

Alexa node.js SDK:

github.com/alexa/alexa-skills-kit-sdk-for-nodejs

Actions on Google node.js SDK:

github.com/actions-on-google/actions-on-google-nodejs

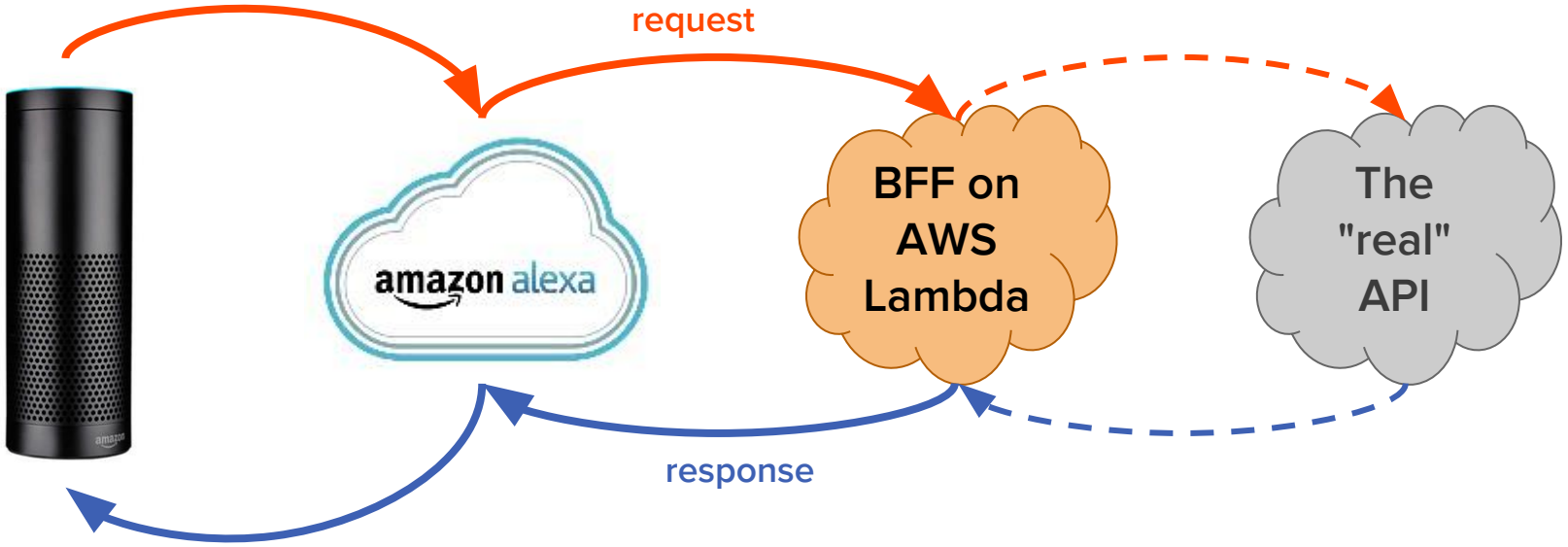
Examples from Alexa SDK

```
responseBuilder.speak("Hello!");
responseBuilder.reprompt("Hello?");
responseBuilder.withSimpleCard(
  "Card Title", "Content!");
responseBuilder.addAudioPlayerPlay
  Directive(...url);
```

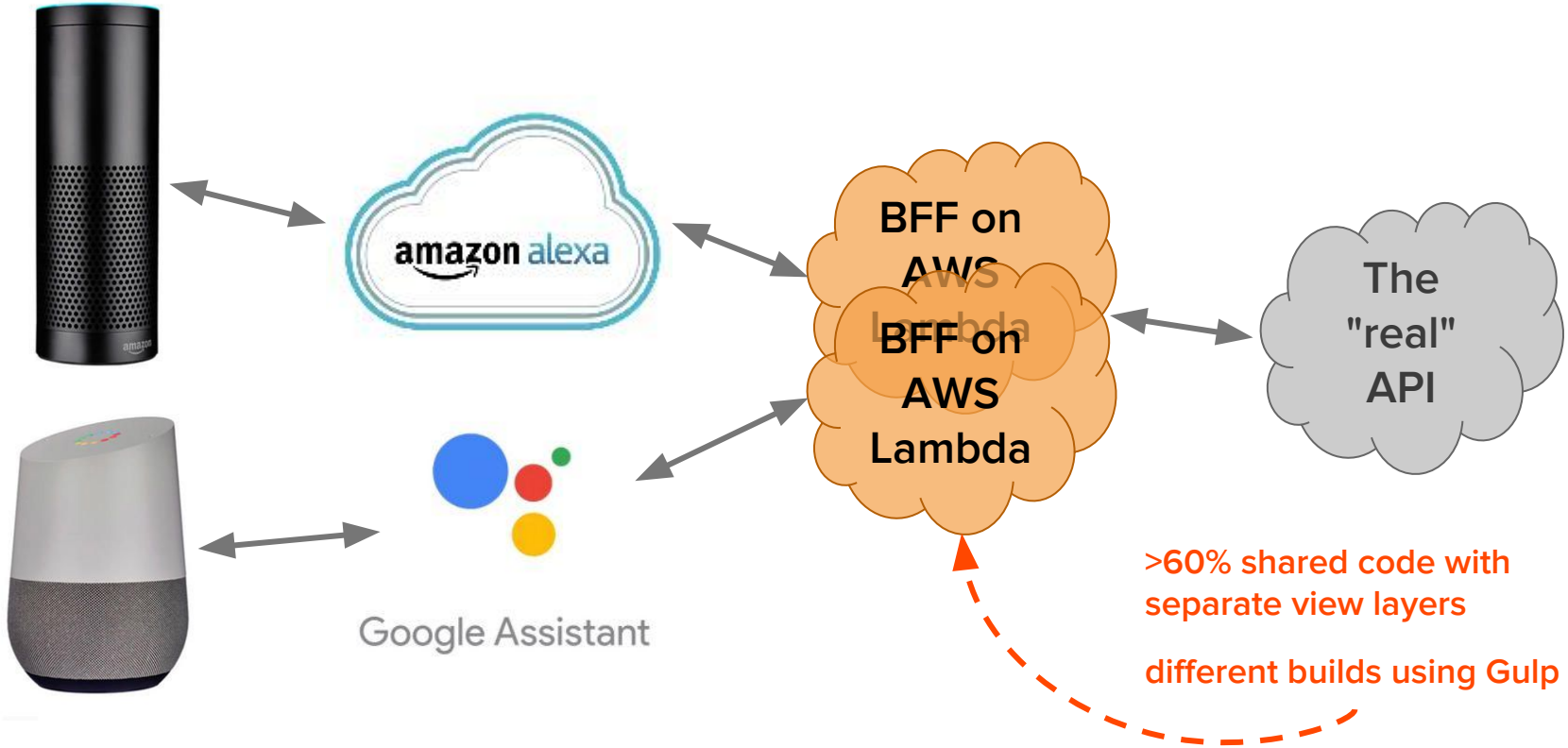
SSML: A common language

```
<say-as interpret-as="characters">WFUV</say-as> is your station.

There is a three second pause here <break time="3s"/> then I continue.

When I wake up, <prosody rate="x-slow">I speak slowly</prosody>.

I can speak with my normal pitch, and also
<prosody pitch="low">with a low pitch</prosody>.
```

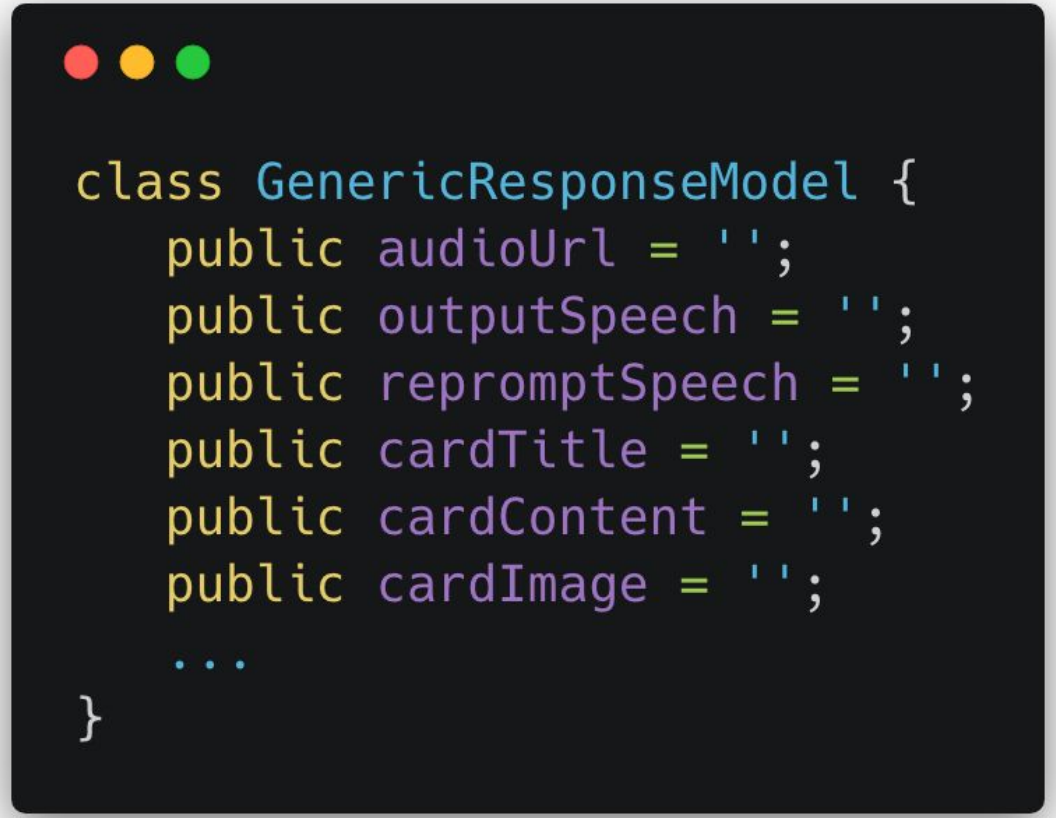
Backends-for-Frontends (BFFs)



Two “skills”, one codebase



Generic Response Model



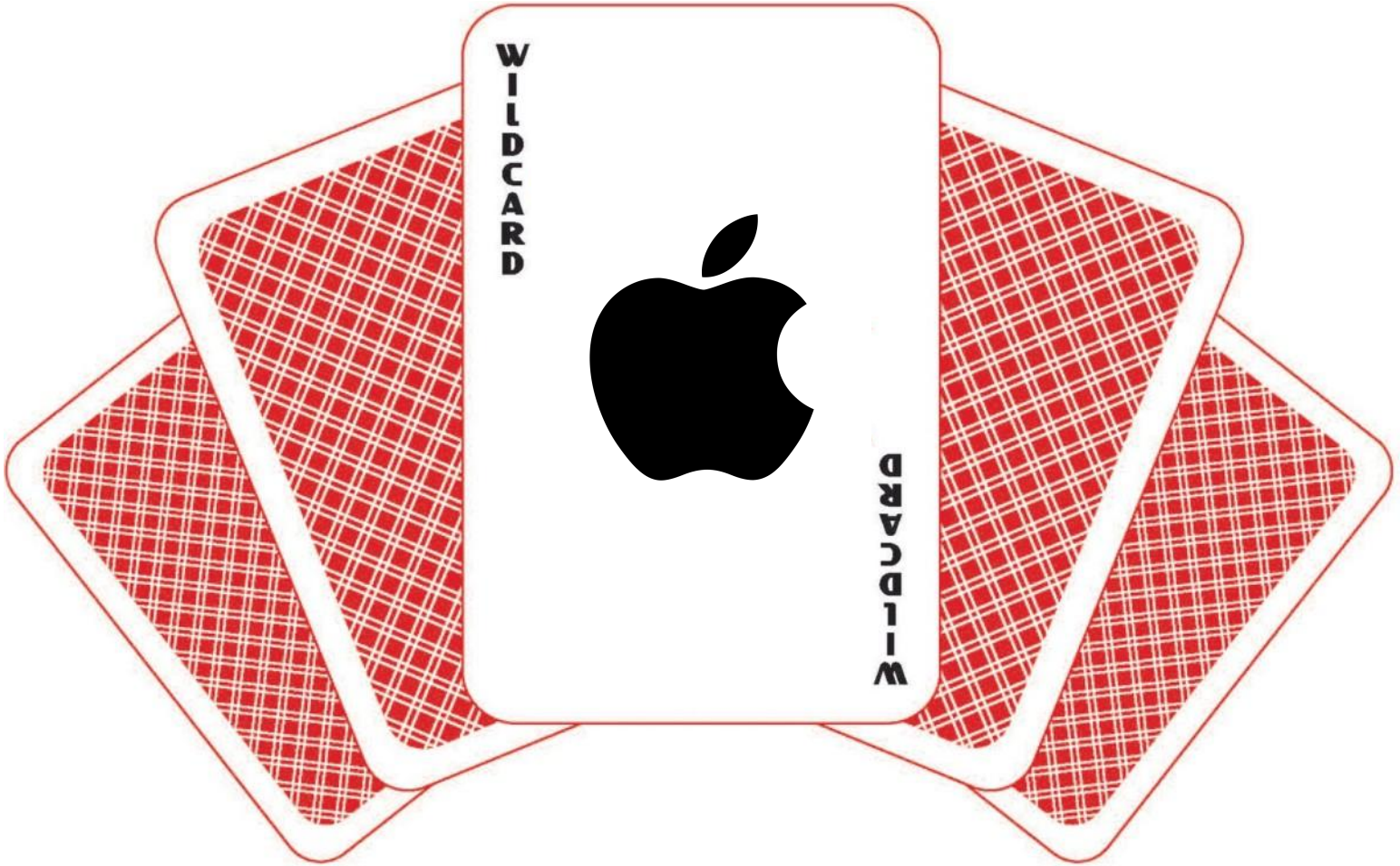
```
class GenericResponseModel {
    public audioUrl = '';
    public outputSpeech = '';
    public repromptSpeech = '';
    public cardTitle = '';
    public cardContent = '';
    public cardImage = '';
    ...
}
```

Challenges

- Text-to-Speech (TtS) is still king
  - Google didn't even add support for their native audio player until February 2018
- No access to the user's location
- Error handling is interesting!
  - User might not even trigger your skill

Conclusions

- The **code** is *not* hard
- Understanding **platform limitations** and **user expectations** are


WILDCARD
WILDCARD

Open source opportunities

- Would it be helpful to have a formalized framework?

Open source opportunities

- Would it be helpful to have a formalized framework?
  - Not really. The code is not hard.
- What we struggle with the most: QA
  - We need something like Selenium or Nightwatch.js for voice UI

Thank you!

nara@nara.codes

@xiehan

https://npr.codes