

Spring Skills

Giving your Spring apps a voice with Alexa

Craig Walls

craig@habuma.com

Twitter: [@habuma](https://twitter.com/habuma)

GitHub: github.com/habuma



Who is this guy?

Engineer with Pivotal

Author Spring in Action & Spring Boot in Action

**Presenter at NFJS, DevNexus, SpringOne, and
countless Java user groups**

**Geeked out by “computing outside of the
computer”**



Why voice UIs?

The natural next step in UI evolution

More natural than typing, clicking, tapping, etc

Fulfils the promises of Sci-Fi

Because it's so darn cool



The power of voice

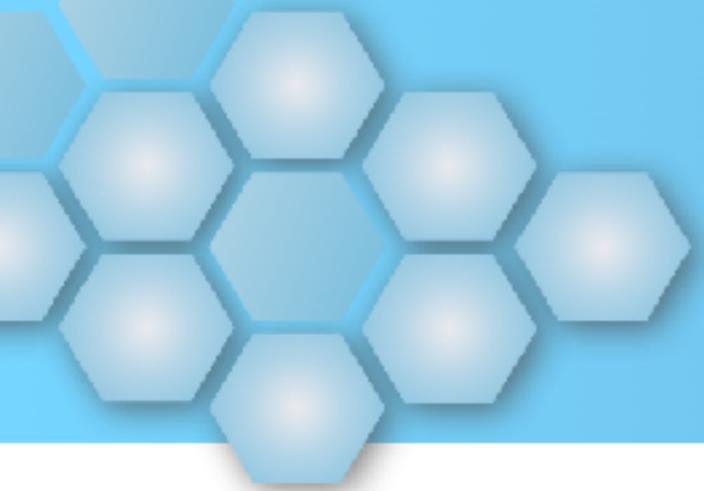
“Alexa, play some Van Halen”

“Alexa, preheat oven to 350”

“Alexa, set an alarm for 5am”

“Alexa, play Jeopardy”

“Alexa, what is the airspeed of an unladen swallow?”



The voice UI landscape

Amazon Alexa (Echo, Echo Dot, Echo Show, Echo Spot, FireTV, etc)

Google Home

Apple Siri (iOS, HomePod)



ASK (Alexa Skills Kit)

APIs, tools, and docs for building custom skills

Skill == Alexa app

Types of skills:
Custom, smart home, flash briefing, video, list



Anatomy of a Skill

Utterances - Things a human might say. They are mapped to Intents.

Intents - Discretely defined actions to perform.

Slots - Parameters to an Intent.

Card - A visual presentation of a skill's output. (For mobile app or Echo Show/Spot)

Function - An AWS Lambda function that will handle requests, including Intent requests and perform the actual work.

Different types of requests: Launch, Intent, SessionEnded



SSML

Speech Synthesis Markup Language

- <amazon:effect>
- <audio>
- <break>
- <emphasis>
- <p>
- <phoneme>

- <prosody>
- <s>
- <say-as>
- <speak>
- <sub>
- <w>

```
<speak>
  <prosody rate="x-slow">
    <say-as interpret-as="spell-out" th:text="${word}"></say-as>
  </prosody>
</speak>
```



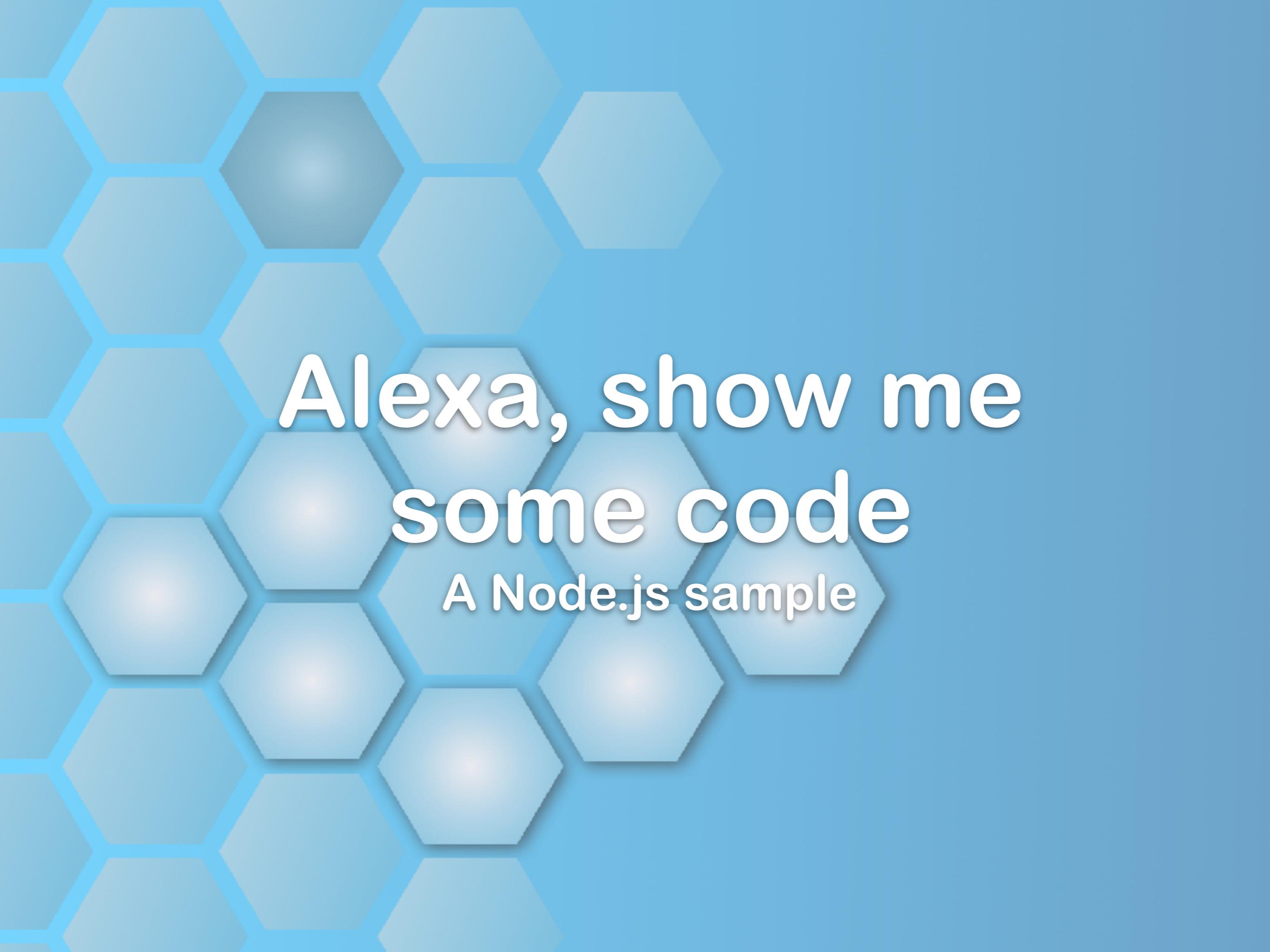
Developing a Skill

Typically developed with Node.js

JS function with same name as Intent

Deployed to AWS Lambda

Skill defined using ASK



Alexa, show me some code

A Node.js sample



But wait...



I'm not that good at JavaScript

Skill-to-Lambda interaction is just JSON



But wait...

```
{  
  "session": {  
    "new": false,  
    "sessionId": "SessionId.ebe629d5-b7ba-409c-8430-b09230e98cd1",  
    "application": {  
      "applicationId": "amzn1.ask.skill.729b861f-21b8-4210-b5ba-401f5d5806f4"  
    },  
    "attributes": {},  
    "user": {  
      "userId": "amzn1.ask.account.AE..."  
    }  
  },  
  "request": {  
    "type": "IntentRequest",  
    "requestId": "EdwRequestId.5fb3dbb8-0693-4ad8-a5c6-dae50fdcb45",  
    "intent": {  
      "name": "SayHello",  
      "slots": {}  
    },  
    "locale": "en-US",  
    "timestamp": "2018-02-06T06:09:55Z"  
  },  
  "context": {  
    "AudioPlayer": {  
      "playerActivity": "IDLE"  
    },  
    "System": {  
      "application": {  
        "applicationId": "amzn1.ask.skill.729b861f-21b8-4210-b5ba-401f5d5806f4"  
      },  
      "user": {  
        "userId": "amzn1.ask.account.AE..."  
      },  
      "device": {  
        "supportedInterfaces": {}  
      }  
    },  
    "version": "1.0"  
  }  
}
```

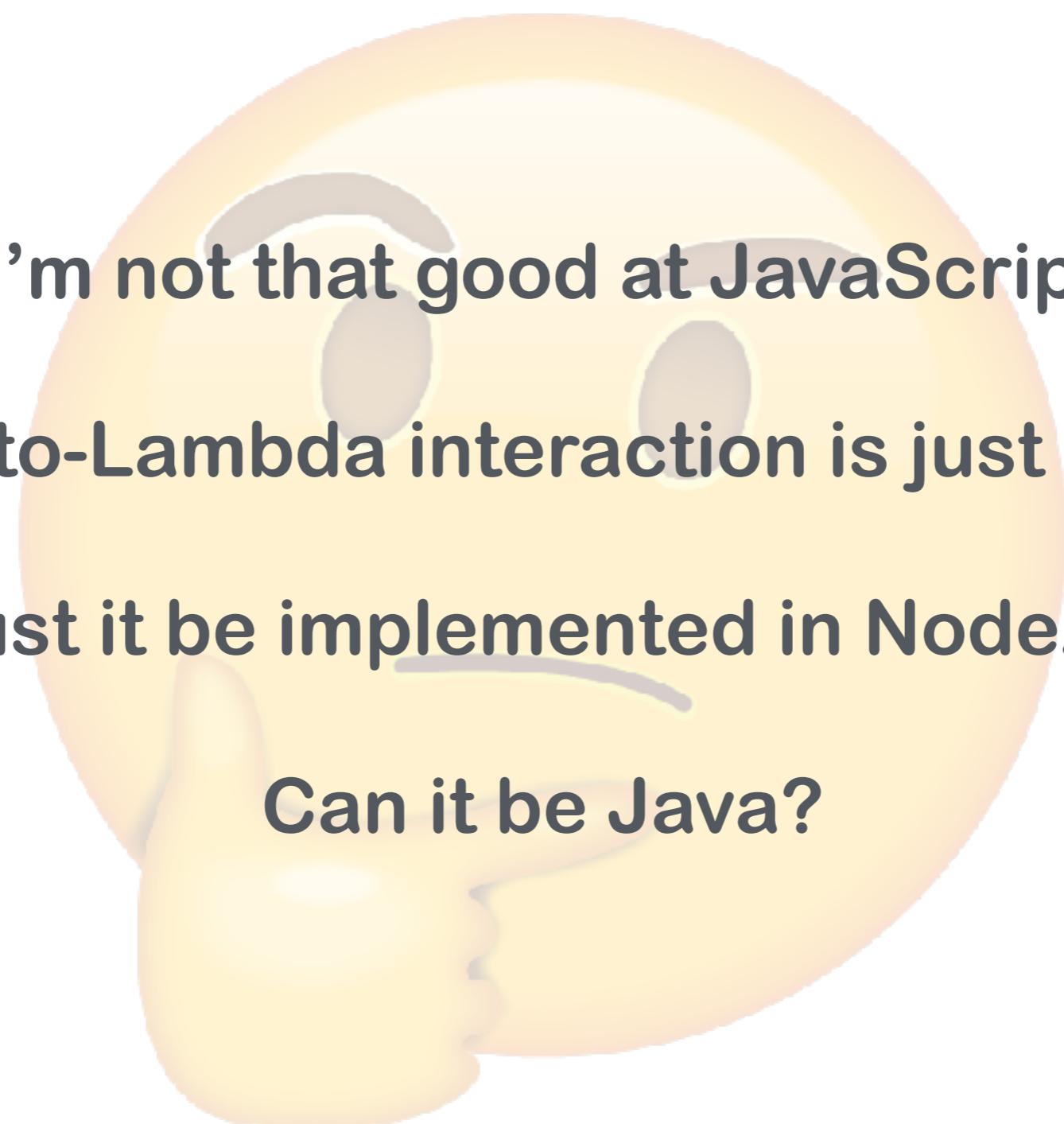


But wait...

```
{  
  "response": {  
    "outputSpeech": {  
      "ssml": "<speak>Hello Spring World!</speak>",  
      "type": "SSML"  
    },  
    "card": {  
      "content": "Hello Spring World!",  
      "title": "Hello"  
    },  
    "speechletResponse": {  
      "outputSpeech": {  
        "ssml": "<speak>Hello Spring World!</speak>",  
        "id": "hello"  
      },  
      "card": {  
        "content": "Hello Spring World!",  
        "title": "Hello"  
      },  
      "shouldEndSession": true  
    }  
}
```



But wait...



I'm not that good at JavaScript

Skill-to-Lambda interaction is just JSON

Must it be implemented in Node.js?

Can it be Java?



Alexa Java SDK

Open Source (Apache 2.0)

**Provides SpeechletV2, a Servlet-like interface
which can be used to create Java-based skills**

**Provides Java representations of requests, cards,
intents, etc.**



Alexa, show me
some Java code



But wait...

A “simple” and “most basic” Java Speechlet example is non-trivial

I’m lazy—I don’t want to write that much code for a “simple” hello world skill

Can’t it be simpler?

Can Spring help?



Spring Cloud Function

Function as a Service (FaaS) for Spring

Functions as the primary concept

```
@Bean  
public Function<String, String> lowercase() {  
    return value -> value.toLowerCase();  
}
```

Can deploy to AWS Lambda



Alexa Java SDK

Two core types...

`SpeechletRequestEnvelope`

`SpeechletResponseEnvelope`

Suitable for JSON (de-)serialization with Jackson



What if...?

Speechlet request/response types

+

Spring Cloud Function



SC Function Skill

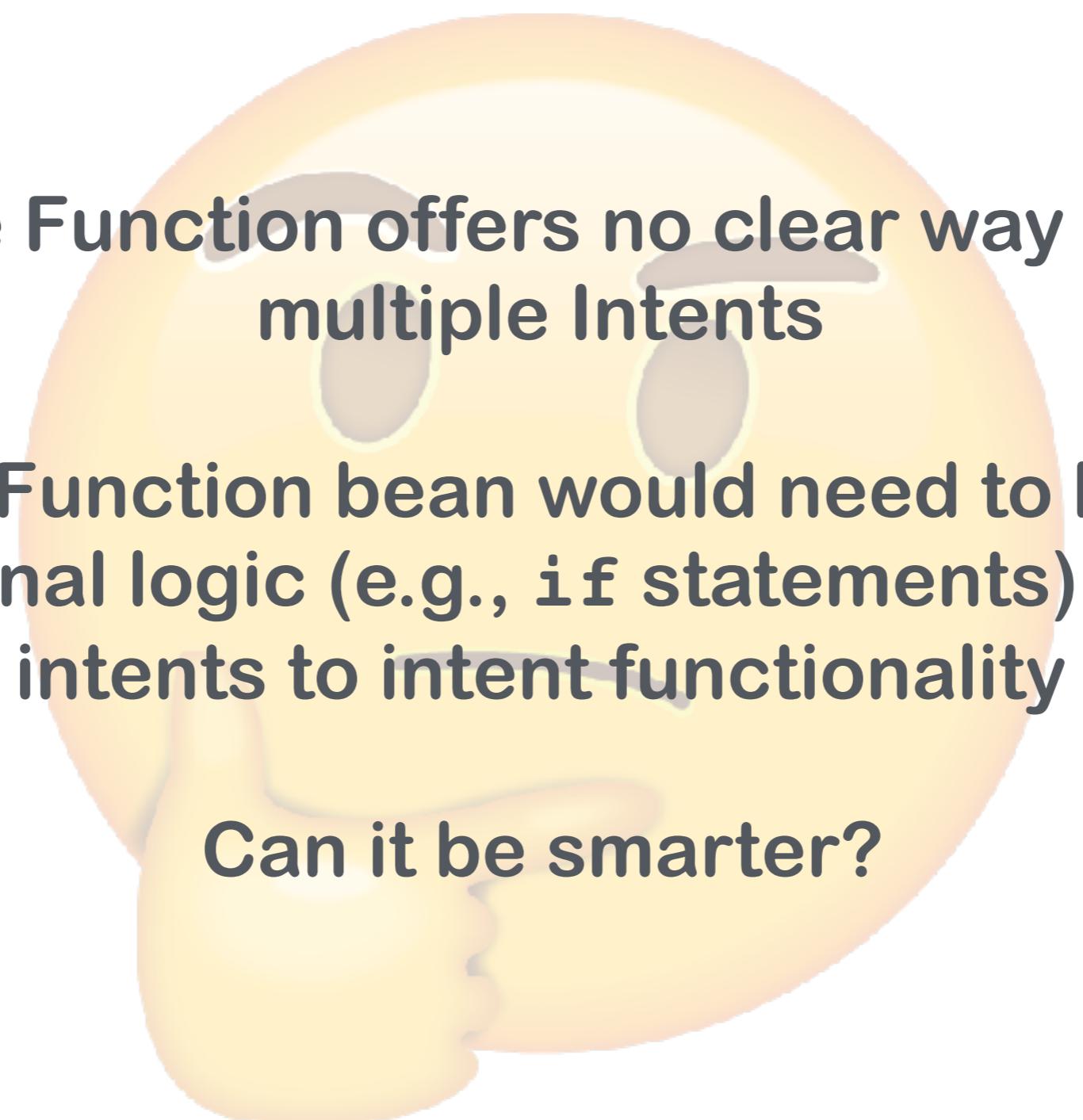
```
@Bean
public Function<SpeechletRequestEnvelope, SpeechletResponseEnvelope> sayHello() {
    return (requestEnv) -> {
        SpeechletResponseEnvelope responseEnv = new SpeechletResponseEnvelope();
        SpeechletResponse response = new SpeechletResponse();
        SsmlOutputSpeech outputSpeech = new SsmlOutputSpeech();
        outputSpeech.setId("hello");
        outputSpeech.setSsml("<speak>Hello World!</speak>");
        response.setOutputSpeech(outputSpeech);
        SimpleCard card = new SimpleCard();
        card.setTitle("Hello");
        card.setContent("Hello World!");
        response.setCard(card);
        responseEnv.setResponse(response);
        return responseEnv;
    };
}
```

In application.yml:
(for the AWS adapter)

function:
name: sayHello



But wait...



A single Function offers no clear way to route multiple Intents

The Function bean would need to have conditional logic (e.g., if statements) to route intents to intent functionality

Can it be smarter?



Spring Skills

Based on Spring Cloud Function

Provides routing for speechlet requests and intents

Supports Amazon's built-in intents

<https://github.com/habuma/spring-skills>



Spring Skills: DISCLAIMER!!!

This is just a personal project

This is not an official Spring portfolio project

**It is not supported by the Spring team nor by
Pivotal**

**My ability to support it is limited by my available
time**



Building a skill with Spring Skills

1. Create a new Spring Boot project
2. Configure build for shaded JARs
3. Add Spring Skills to the dependencies*
4. Define one or more Function beans that are named to match your skill's intents
5. Build and deploy to AWS Lambda
6. Configure skill

* You'll need to build/install Spring Skills first, as it isn't in a public Maven repo (yet).



Shaded JARs

```
<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
      <dependencies>
        <dependency>
          <groupId>org.springframework.boot.experimental</groupId>
          <artifactId>spring-boot-thin-layout</artifactId>
          <version>${wrapper.version}</version>
        </dependency>
      </dependencies>
    </plugin>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-shade-plugin</artifactId>
      <configuration>
        <createDependencyReducedPom>false</createDependencyReducedPom>
        <shadedArtifactAttached>true</shadedArtifactAttached>
        <shadedClassifierName>aws</shadedClassifierName>
      </configuration>
    </plugin>
  </plugins>
</build>
```



Define handler beans

```
@Bean  
public Function<SpeechletRequestEnvelope, SpeechletResponseEnvelope> sayHello() {  
    return (requestEnv) -> {  
        SpeechletResponseEnvelope responseEnv = new SpeechletResponseEnvelope();  
        ...  
        return responseEnv;  
    };  
}
```

Flexible intent/bean naming

Intent Name	Bean Name
SayHelloIntent	sayHello
SayHello	sayHello
sayHello	sayHello

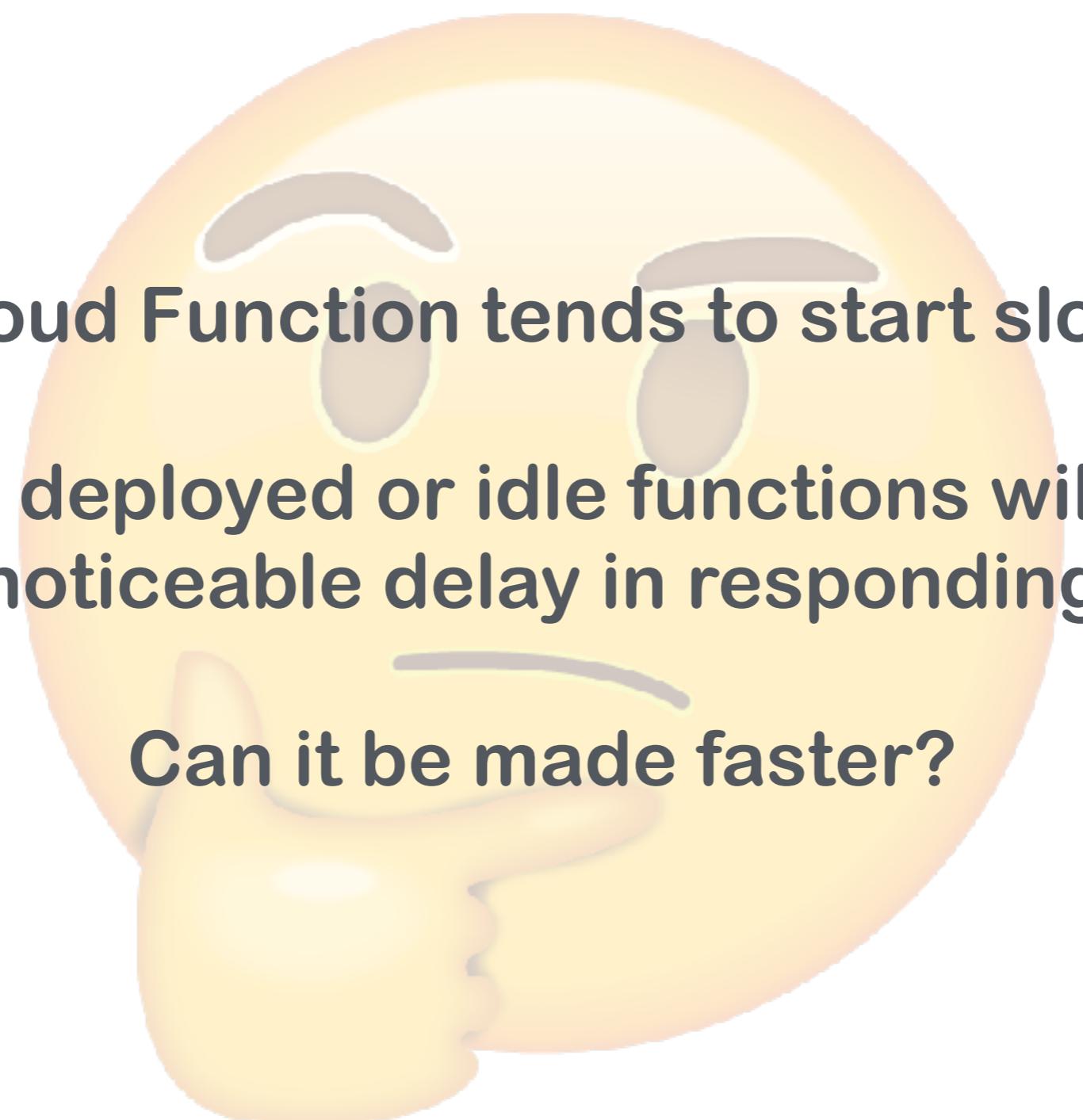


Alexa, show me some **Spring** code

A Spring Skills sample



But wait...



Spring Cloud Function tends to start slow in AWS

Newly deployed or idle functions will have
noticeable delay in responding

Can it be made faster?



The need for speed

Three options...

- Set longer idle timeouts 😠
- Improve Spring startup times 😕
- Write it as a traditional Spring web app 😊



Skills controller

```
@RestController
public class HelloSpeechletController {
    @PostMapping("/hello")
    public SpeechletResponseEnvelope sayHello(SpeechletRequestEnvelope requestEnv) {
        SpeechletResponseEnvelope responseEnv = new SpeechletResponseEnvelope();
        SpeechletResponse response = new SpeechletResponse();
        SsmlOutputSpeech outputSpeech = new SsmlOutputSpeech();
        outputSpeech.setId("hello");
        outputSpeech.setSsml(asSSML("Greetings from a Spring Boot web application."));
        response.setOutputSpeech(outputSpeech);
        SimpleCard card = new SimpleCard();
        card.setTitle("Hello");
        card.setContent("Greetings from a Spring Boot web application.");
        response.setCard(card);
        responseEnv.setResponse(response);
        return responseEnv;
    }

    private static String asSSML(String text) {
        return "<speak>" + text + "</speak>";
    }
}
```



Request envelope resolver

```
@Configuration
public class WebConfig extends WebMvcConfigurerAdapter {
    @Override
    public void addArgumentResolvers(List<HandlerMethodArgumentResolver> argumentResolvers) {
        argumentResolvers.add(new HandlerMethodArgumentResolver() {
            @Override
            public boolean supportsParameter(MethodParameter param) {
                return param.getParameterType().isAssignableFrom(SpeechletRequestEnvelope.class);
            }

            @Override
            public Object resolveArgument(MethodParameter param,
                ModelAndViewContainer mavc,
                NativeWebRequest request,
                WebDataBinderFactory binderFactory) throws Exception {
                HttpServletRequest httpServletRequest =
                    request.getNativeRequest(HttpServletRequest.class);
                ServletInputStream inputStream = httpServletRequest.getInputStream();
                byte[] bytes = IOUtils.toByteArray(inputStream);
                ...
                return SpeechletRequestEnvelope.fromJson(bytes);
            }
        });
    }
}
```



Verifying the signature

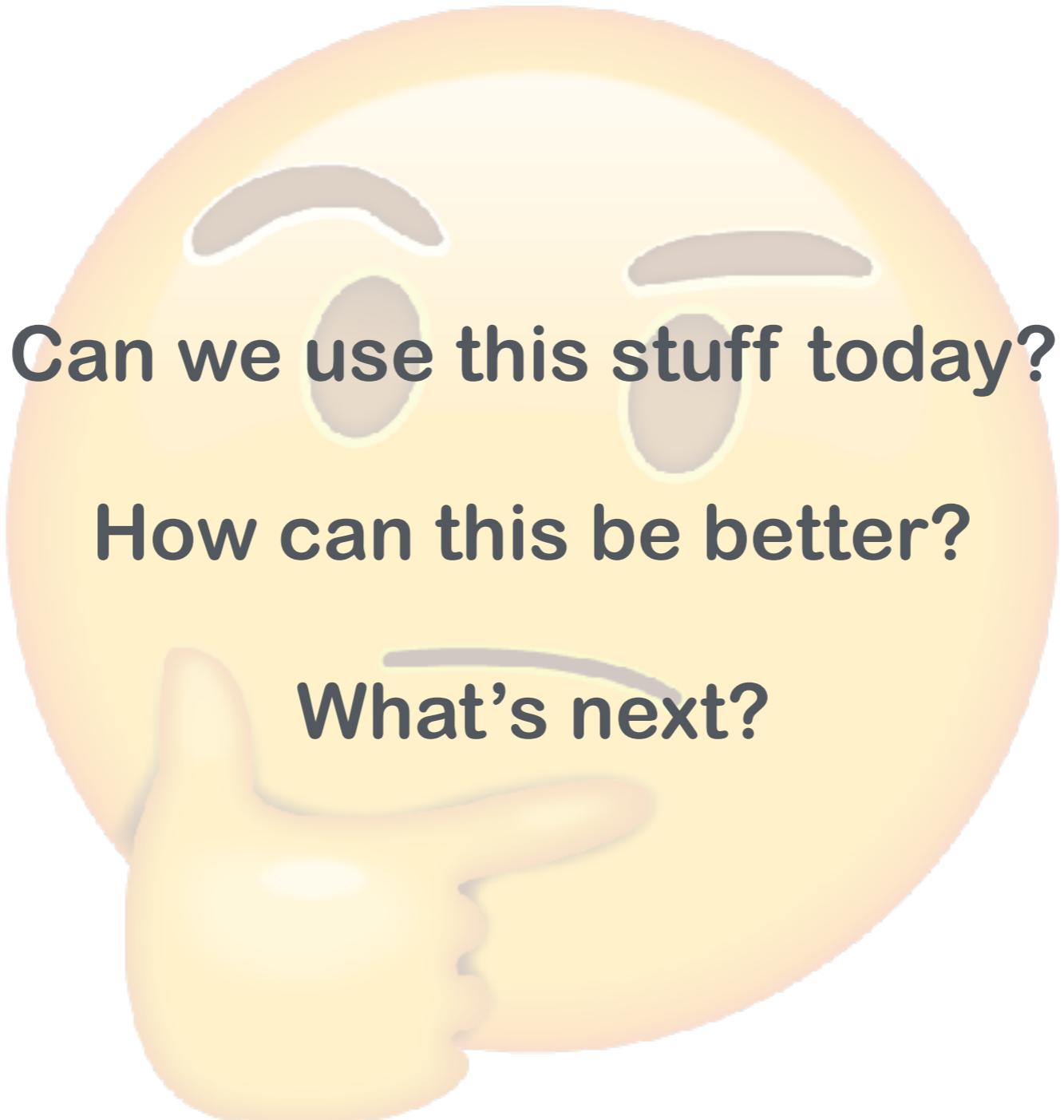
```
@Override
public Object resolveArgument(MethodParameter param,
                               ModelAndView mavc,
                               NativeWebRequest request,
                               WebDataBinderFactory binderFactory) throws Exception {
    HttpServletRequest httpServletRequest =
        request.getNativeRequest(HttpServletRequest.class);
    ServletInputStream inputStream = httpServletRequest.getInputStream();
    byte[] bytes = IOUtils.toByteArray(inputStream);

    SpeechletRequestSignatureVerifier.checkRequestSignature(bytes,
        request.getHeader(Sdk.SIGNATURE_REQUEST_HEADER),
        request.getHeader(Sdk.SIGNATURE_CERTIFICATE_CHAIN_URL_REQUEST_HEADER));

    return SpeechletRequestEnvelope.fromJson(bytes);
}
```



But wait...



Can we use this stuff today?

How can this be better?

What's next?



A work-in-progress

Consider non-Function intent handlers

Simplify creation of response envelope

Perhaps support a controller-like handler

Speechlet extension for Spring MVC/WebFlux

Expand work to support other kinds of skills

Faster startup?

More examples and docs



Just imagine... (Not yet implemented!)

A Skill controller

```
@SkillController
public MySkill {
    @IntentMapping("SayHello")
    public SpeechletResponseEnvelope sayHello(SpeechletRequestEnvelope req) {
        ...
    }

    @IntentMapping("SpellAWord")
    public String spellAWord(@Slot("word") String word, Model model) {
        model.setAttribute("word", word);
        return "spellAWord";
    }
}
```

A Thymeleaf “view”

```
<speak>
    <prosody rate="x-slow">
        <say-as interpret-as="spell-out" th:text="${word}"></say-as>
    </prosody>
</speak>
```



A work-in-progress

Consider non-Function intent handlers

Simplify creation of response envelope

Perhaps support a controller-like handler

Speechlet extension for Spring MVC/WebFlux

Expand work to support other kinds of skills

Faster startup?

More examples and docs

Thank you!

Code: <https://github.com/habuma/spring-skills>

Check out my books:

