



WEBINAR

Micronaut and Alexa

LEARN HOW TO BUILD ALEXA SKILLS FOR MICRONAUT

- Software Engineer on Grails/Micronaut team at Object Computing (Home to Grails)
- Father of 2 kiddos 7 and 14
- Tinkerer of gadgets, automobiles, and beer and wine
- Talk to me if you need Grails, Groovy or Micronaut support



WE ARE SOFTWARE ENGINEERS.

We deliver mission-critical software solutions that accelerate innovation within your organization and stand up to the evolving demands of your business.

AREAS OF EXPERTISE



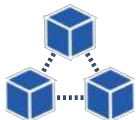
INDUSTRIAL IoT

We equip industrial environments with seamless connectivity and real-time analytics that reduce operating costs and deliver on customer demands.



MACHINE LEARNING

We can modernize your legacy applications and enable scalable, integrated AI capabilities tailored to your unique sets of data.



BLOCKCHAIN CONSULTING

We are at the forefront of blockchain technology, and we have practical, real-world experience with its implementation.



CLOUD SOLUTIONS

We combine software engineering expertise with cloud-native architecture to accelerate innovation within your organization.

Agenda



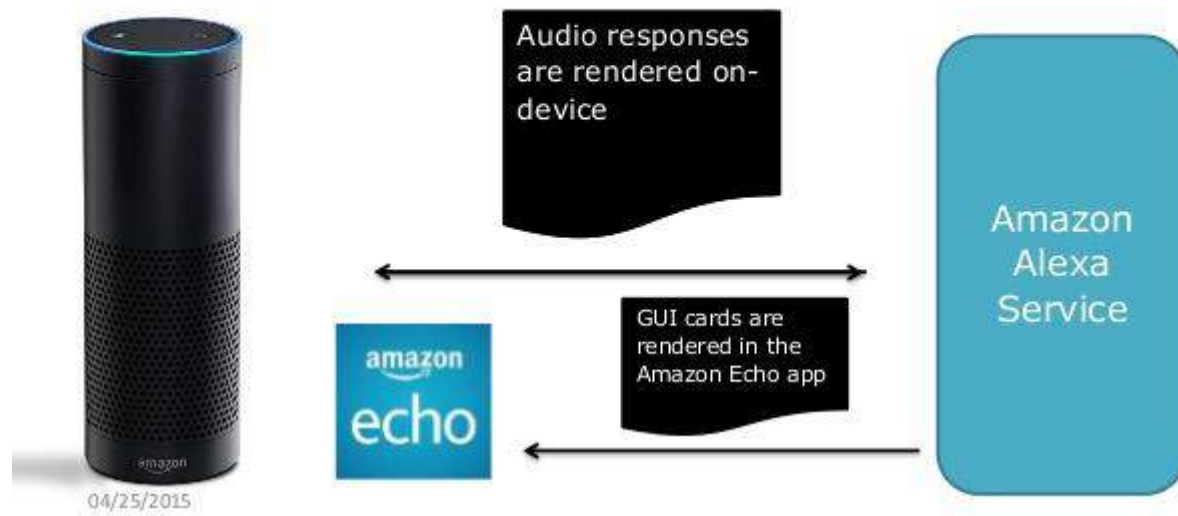
- What is Alexa and what devices it covers
- How does Alexa work?
- Evolution of UI
- Alexa software concepts and components v2 API
- Brief Lambda Overview
- Micronaut Lambda Groovy App Setup
- Hello World Examples Walkthrough
- More advanced - Star Wars Quiz Skill

What is Alexa?



Alexa is a cloud based voice recognition service

Alexa Architecture



There are 3 SDKS - Skills(3 subsets), AVS and Home API

The Devices



ECHO V1/V1+, V2/V2+

Things Alexa Can Do



The Internals





Echo(V1)



- The first one
- Multi dimensional mic array
- Nice speaker with downfiring subwoofer
- Some simple home automation ability
- Wifi and Bluetooth capabilities(can only accept BT)

Echo Dot (V1/V2/V3)

- Baby brother - has the fancy mics but minimal speaker for voice
- Has Wifi and bluetooth capabilities (both ways)
The big deal: a 3.5mm audio output jack!
- Always on listening for wake word (Alexa or Amazon)
- Gen 2 has +/- volume buttons on top instead of 'ring'
- v2 under \$50US



Echo Dot Insides (V1)



Tap



Echo Tap



- On the go - portable bluetooth speaker with battery and simple mic
 - Has Wifi and bluetooth capabilities
 - A 3.5mm audio INPUT jack!
- Now always listening or push button- have to 'tap' button to wake

Echo Look

TECHNICAL DETAILS



Echo Look



- Style check - get second opinion on your outfit (Alexa, how do I look?)
- No calling, bluetooth, or messaging abilities
- Has flash for pictures
- Can record pictures or videos on command

FireTV



FireTV



- Latest versions have Alexa on the remote
- Can work with all FireTV sticks, box and cube
- Mostly meant for TV apps - cheapest option for Alexa
- Recent OS updates support Alexa on both V1 and V2
- Must push voice remote button to start apps
The same skills work here too!
- Some built in functions don't work like alarm

Echo Show V1 and V2



Echo Show



- Can create visual layouts and videos with Skills API
- V2 has much bigger screen
- You can also convert a Fire tablet into a Show via software
- Good sound
- Only bluetooth connections, physical inputs/outputs
- Has camera

Echo Spot

- Basically miniature Spot meant to be an alarm clock
- Sound quality not great
- Great for bedsides



Echo Input

- Speakerless mic only device, meant to look into as an input to another audio device, like a home stereo



Alexa Gadgets

- These device are not full fledged echo devices, but are meant to connect to and accompany one with additional functionality (usually bluetooth)
- Examples:
 - Echo buttons
 - Wall Clock
 - Microwave
 - Big Mouth Billy Gass
 - Dancing Push robots
 - Holiday lights
- Uses Alexa Gadget API (beta)



Other devices



- Google Home - and derivatives
- Facebook Portal
- Apple Homepod
- LingLong DingDong (Chinese only)

Make your own!

- All you need is a raspberry PI and a microphone!
- Official AVS tutorial on PI
<https://developer.amazon.com/docs/alexa-voice-service/register-a-product.html>

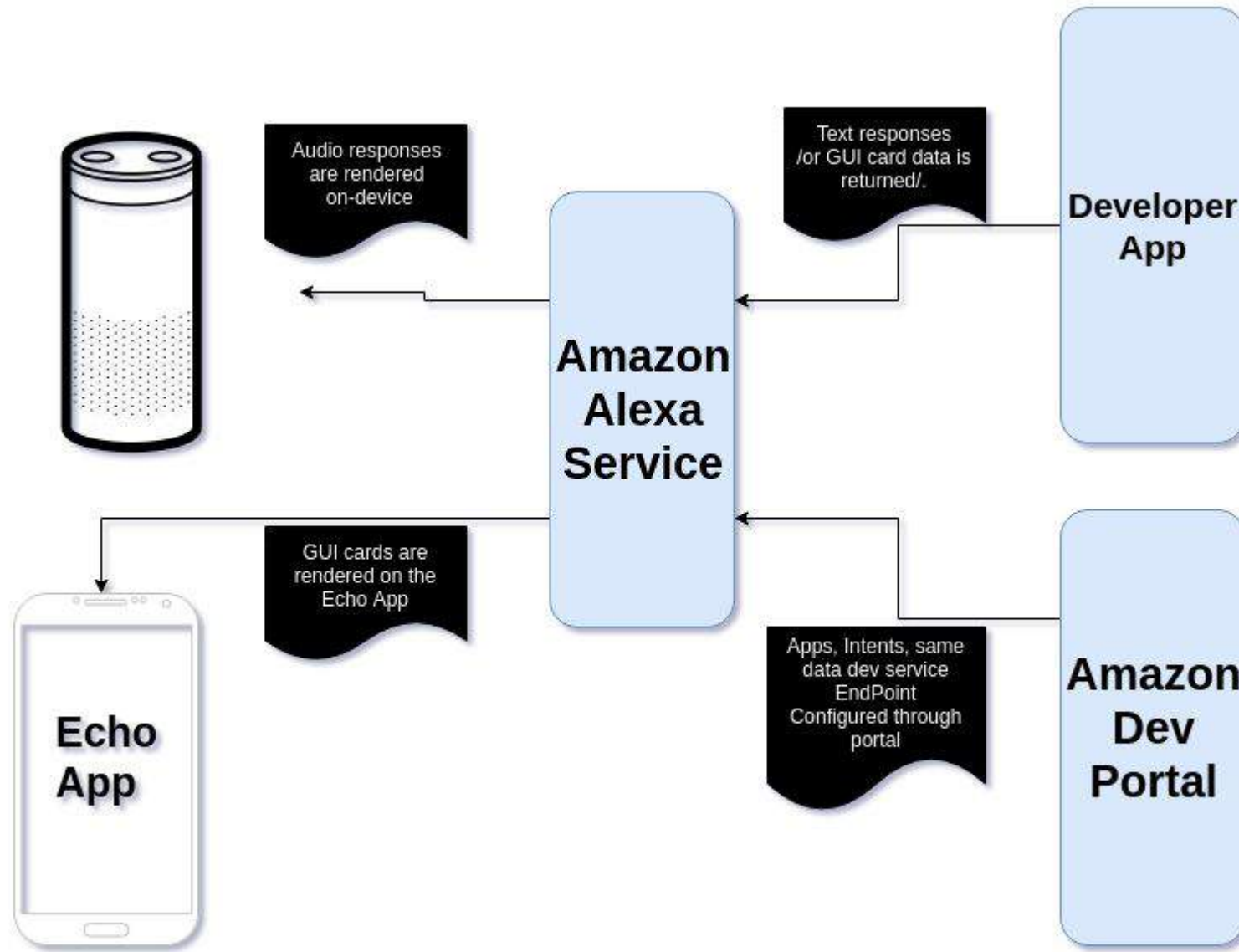


BACK TO ALEXA SKILLS - HOW DO THEY WORK?



- App developer never talks directly to device
- Device must initiate interaction
- Alexa server response with JSON body
- Text to speech, small sound clips, video, or music/sound files all supported
- Currently only works as Lambda Function on Micronaut

BACK TO ALEXA SKILLS - HOW DO THEY WORK?



AN EVOLUTION OF UI



- Character based interfaces
- Graphical User Interfaces (GUI)
- Web Interfaces
- Mobile Interfaces
- Voice User Interfaces

CHARACTER BASED INTERFACES

- 70s
- Green screens
- Readability issues
- Mapping characters for input

```
A0C6 -- Sample JES Panel A0C6JES -- 02/10/15 20:52:00

----- JES SP00L Information -----

Overall SP00L utilization: 85.9575 %

Jobnum  Jobname  % Spool  I  Jobnum  Jobname  % Spool  ===== 1/10(15)
JOB07066 BH0LPP5   22.9090  |  JOB07056 BH0LPP1    5.4242
JOB07059 BH0LPP4   19.4121  |  JOB07081 BH0LPP2    4.3757
JOB07058 BH0LPP4   17.6666  |  STC06598 SYSLOG     0.4787
JOB07061 BH0LPP3    8.9212  |  STC06647 NET        0.0727
JOB07082 BH0LPP2    6.1272  |  STC06645 CAZO       0.0666

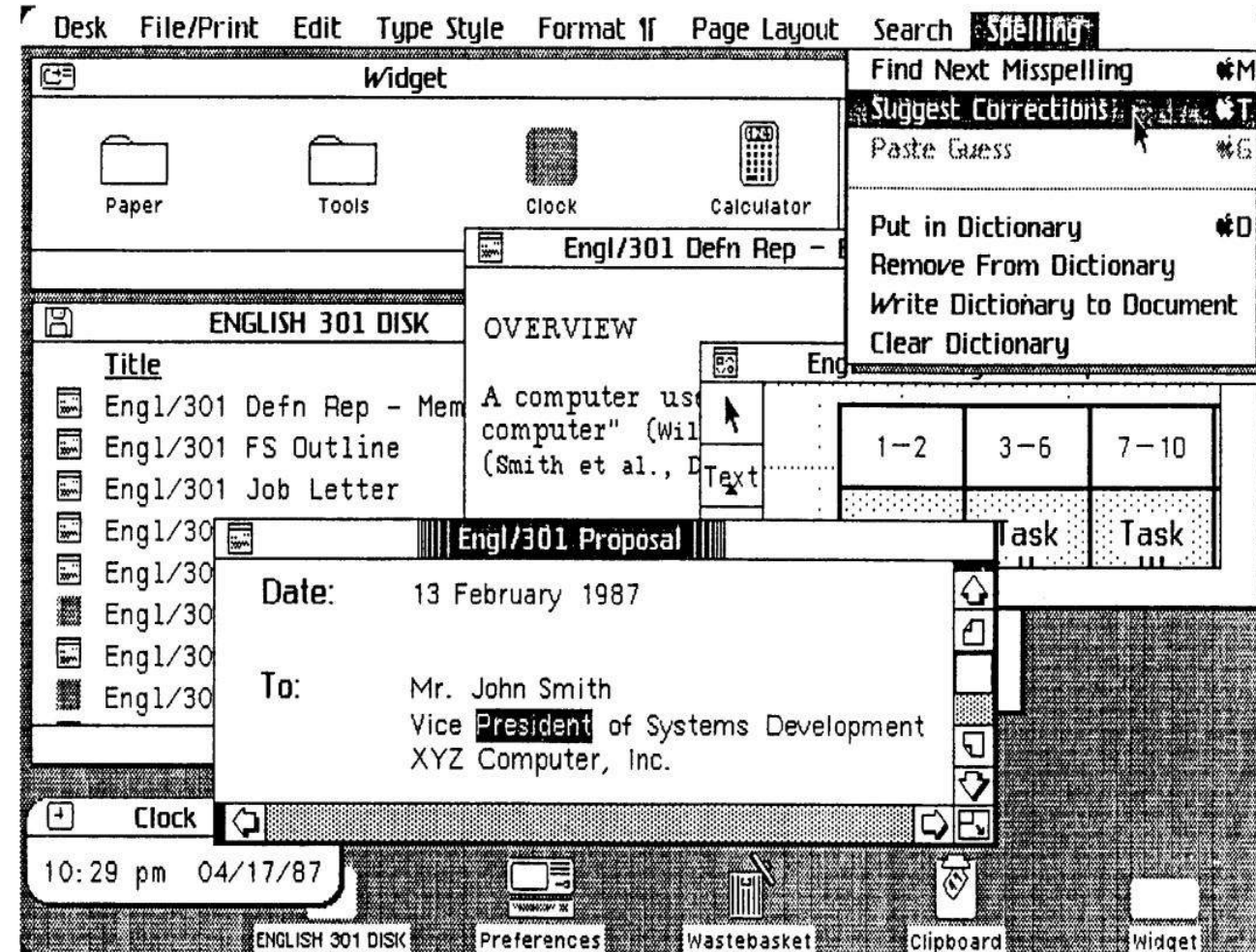
----- Other JES Information -----

Free job numbers . . . . : 9,918 (99.19%)
Free job queue elements : 5,919 (98.65%)
Free job output elements : 6,989 (99.84%)
Free BERTs . . . . . : 11,839 (97.84%)

===> 
1=Help 2=Detail 5=Purge 10=Prev 11=Next
```

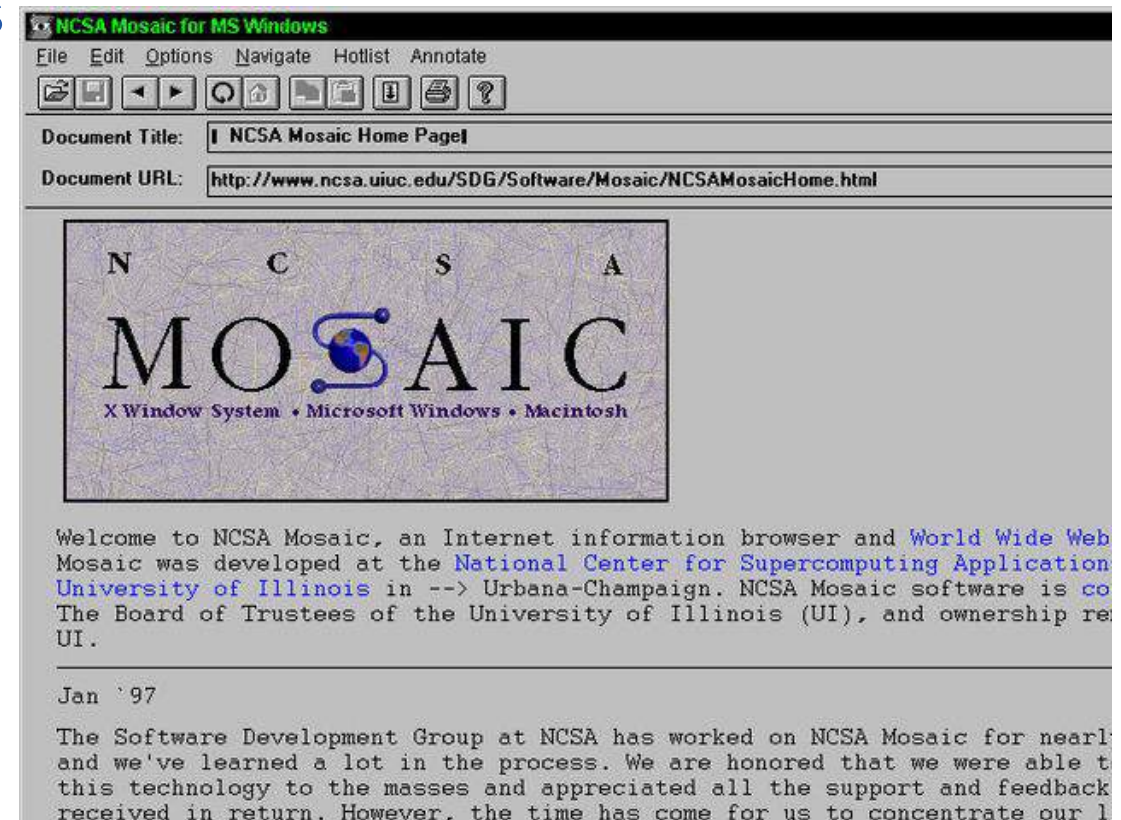
GRAPHICAL USER INTERFACES

- 80s
- Much more powerful
- heavy weight
- started leveraging metaphors
- coded for intuitive user actions
- fit A LOT more data on the screen



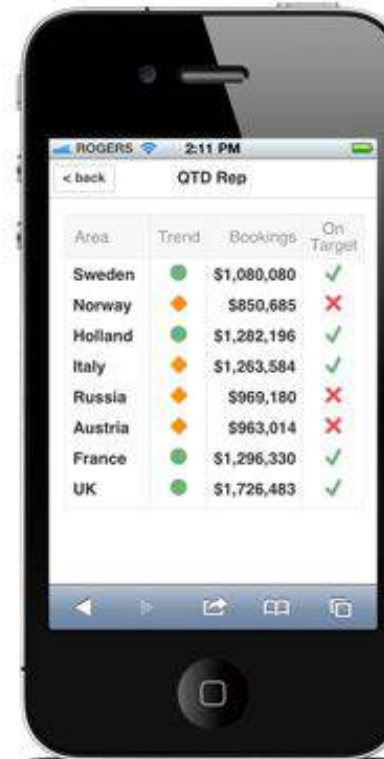
WEB INTERFACES

- 90s
- Changed the way we organize information
- mapped actions by number of clicks



MOBILE INTERFACES

- 2000s
- discovered more information visible isn't necessarily better
- introduced gestures



VOICE USER INTERFACES

- Now
- speaking patterns are complex
- Many ways of saying the same thing



- Skills SDK
 - Custom
 - Flash Briefing
 - Smart Home
 - Music
 - Video
 - Baby Activity
- Voice SDK - build a device like and Echo
- We will focus on the Skills Custom API today



SKILLS SDK - DON'T CALL US, WE'LL CALL YOU



- We host this as a Lambda function that parses JSON requests
- Responds in kind with JSON
- Alexa Java SDK v2 now gets us there
- V2 Java SDK completely different than V1 fixes a lot of old problems supporting display and sound



SKILLS SDK - DON'T CALL US, WE'LL CALL YOU



- Does some things for us like validate the request and call events
- Gives us decent interfaces that use an API form SpeechResponses
- Also helps us build Card responses and stream Audio
- Let's dig in to the specifics!

SKILLS SDK - DON'T CALL US, WE'LL CALL YOU



- This tells Alexa what actions your app can do
- Describes intents and slots used
- When you upload this(and utterances), Alexa calculates speech variations to launch them
- See sample IntentSchema.json



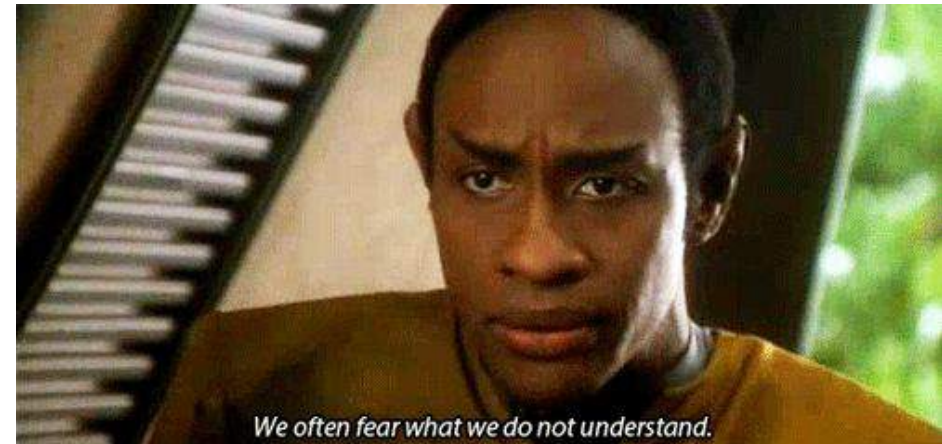
SKILLS SDK - SLOTS

- This is how Alexa parameterizes commands
- Very simple list of options
- Slots don't work well for variable/word parameter responses
- Define custom slots for types not built into Alexa
- Built in slot types always growing list is at <http://amzn.to/2fHnHd5>



SKILLS SDK - SAMPLE UTTERANCES

- Train Alexa on what the commands it will understand
- This is the phrases that activate the intents
Use {} and | to use parameters
- When you upload this, Alexa calculates speech variations to launch Intents
- Protip: use redundant and misspelled word variations to help Alexa understand



SKILLS SDK - THE ALEXA APP



- This is where you can do non-speech interactions
- See a log of what you've done
- See cards returned by skill
- Install Skills / browse Skills library by category
- Can use mobile app or go to echo.amazon.com



SKILLS SDK - CARDS

- This is similar to Android cards but more basic
- You can launch a separate card and speech response for intents
- There are 3 main cards: Simple(text only), Standard(1 pic), and LinkAccount
- Amazon limits what kind of content can be on each type of card (no HTML/js/css)



SKILLS SDK - INTENTS

- Micronaut uses concept of annotations tagged as @IntentHandler
- Built-in Intents are in AlexaIntents class: STOP, CANCEL, FALLBACK and HELP
- Any other intents are annotated like so: @IntentHandler("SomeIntent")
- Can be inside one class or individual handler classes that implement IntentHandler interface



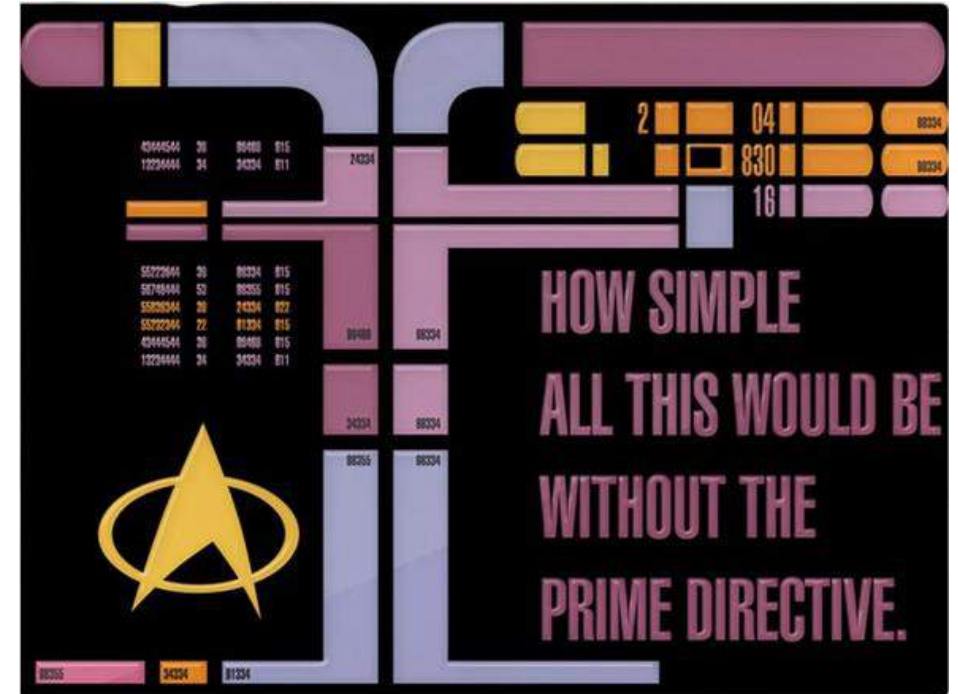
SKILLS SDK - INTENTS

- Micronaut uses concept of annotations tagged as `@IntentHandler`
- Built-in Intents are in `AlexaIntents` class: `STOP`, `CANCEL`, `FALLBACK` and `HELP`
- Any other intents are annotated like so: `@IntentHandler("SomeIntent")`
- Can be inside one class or individual handler classes that implement `IntentHandler` interface



SKILLS SDK - AUDIOPLAYER DIRECTIVES

- This tells Alexa to queue, play, or stop playback
- Directives are `PlayBehavior.ENQUEUE`, `REPLACE_ALL`, `REPLACE_ENQUEUED`
- These are tacked on as `.addAudioPlayerPlayDirective` on the response builder
- These are only suitable for long duration playback like music or podcasts - Use SSML for audio snippets in conversation



SKILLS SDK - THAT'S COOL, BUT CAN IT ONLY TO TEXT TO SPEECH?

- Nope! We can use SSML to play sounds clips!
- Limited to 90s, 48kbit SSL hosted mp3 only and picky about format
- Can do other commands like pronounce words
- Let's check out the SSML Reference <http://amzn.to/1OaLmAZ>
- Audio and Video Player support
 - <http://amzn.to/2fRikwA> and <http://amzn.to/2sbO3v3>

SKILLS SDK - SSML AUDIO PROTIP



- Use ffmpeg to resample your mp3 so amazon likes it (s3 is easiest)
 - `ffmpeg -y -i -ar 16000 -ab 48k -codec:a libmp3lame -ac 1 .mp3`

SKILLS SDK - SSML MARKUP



< speak >

< audio

src="\"https://s3.amazonaws.com/vanderfox-sounds
/test.mp3\" / "> \${speechText}

< / audio >

< / speak >

SKILLS SDK - SSML MARKUP



< speak >

Here is a number spoken as a cardinal number:

```
< say-as interpret-as="cardinal">12345</ say-as >.
```

Here is the same number with each digit spoken separately:

```
< say-as interpret-as="digits">12345</ say-as >.
```

Here is a word spelled out: < say-as

```
interpret-as="spell-out">hello</ say-as >
```

</ speak >

TESTING OPTIONS



- Using the developer.amazon.com test tab
- Use echosim.io browser tester!(No Audio skills support)
- Use a physical device that is attached to your account (share with beta feature!)

MICRONAUT HELLO WORLD EXAMPLE APPS



- Starting with Micronaut 1.1 Alexa Support Built In
- Example in Kotlin, Java, and Groovy
- Let's look!

TESTING OPTIONS



- Using the developer.amazon.com test tab
- Use echosim.io browser tester!(No Audio skills support)
- Use a physical device that is attached to your account (share with beta feature!)

VISUALS ON DISPLAYS

- Enable directives on console

NAME	DESCRIPTION	
Audio Player	The AudioPlayer interface provides directives and requests for streaming audio and monitoring playback progression. Learn more about the Audio Player Interface.	<input type="checkbox"/>
Display Interface	Echo Show allows skill developers to create skills for Alexa that use both screen and voice interaction. Learn more about the Display Interface.	<input type="checkbox"/>
Video App	The VideoApp interface provides the VideoApp.Launch directive for streaming native video files in Echo Show. Learn more about the VideoApp Interface.	<input type="checkbox"/>
Alexa Gadget BETA	Create Alexa Gadget skills using the Gadget Controller Directives or Game Engine Inputs. Learn More about creating Alexa Gadget Skills. Gadget Controller: Enable your skill to control an Alexa Gadget. Game Engine: Enable your skill to receive input from an Alexa Gadget.	<input type="checkbox"/> <input type="checkbox"/>
CanFulfillIntentRequest BETA	Support runtime checks for whether a customer's request can be fulfilled. Learn More about the benefits of adding support for the CanFulfillIntentRequest interface to your Alexa skill.	<input type="checkbox"/>
Alexa Presentation Language BETA	The Alexa Presentation Language (APL) allows skill developers to build multimodal skills for Echo Show, Echo Spot, and Fire TV. Use the authoring tool and simulator to create and test APL documents. Please refer to the API Reference for the beta limitations. Additionally, you can learn more about this feature in our Blog post .	<input type="checkbox"/>

BODYTEMPLATE1

Header Text



Ut enim ad minima veniam,
quis nostrum

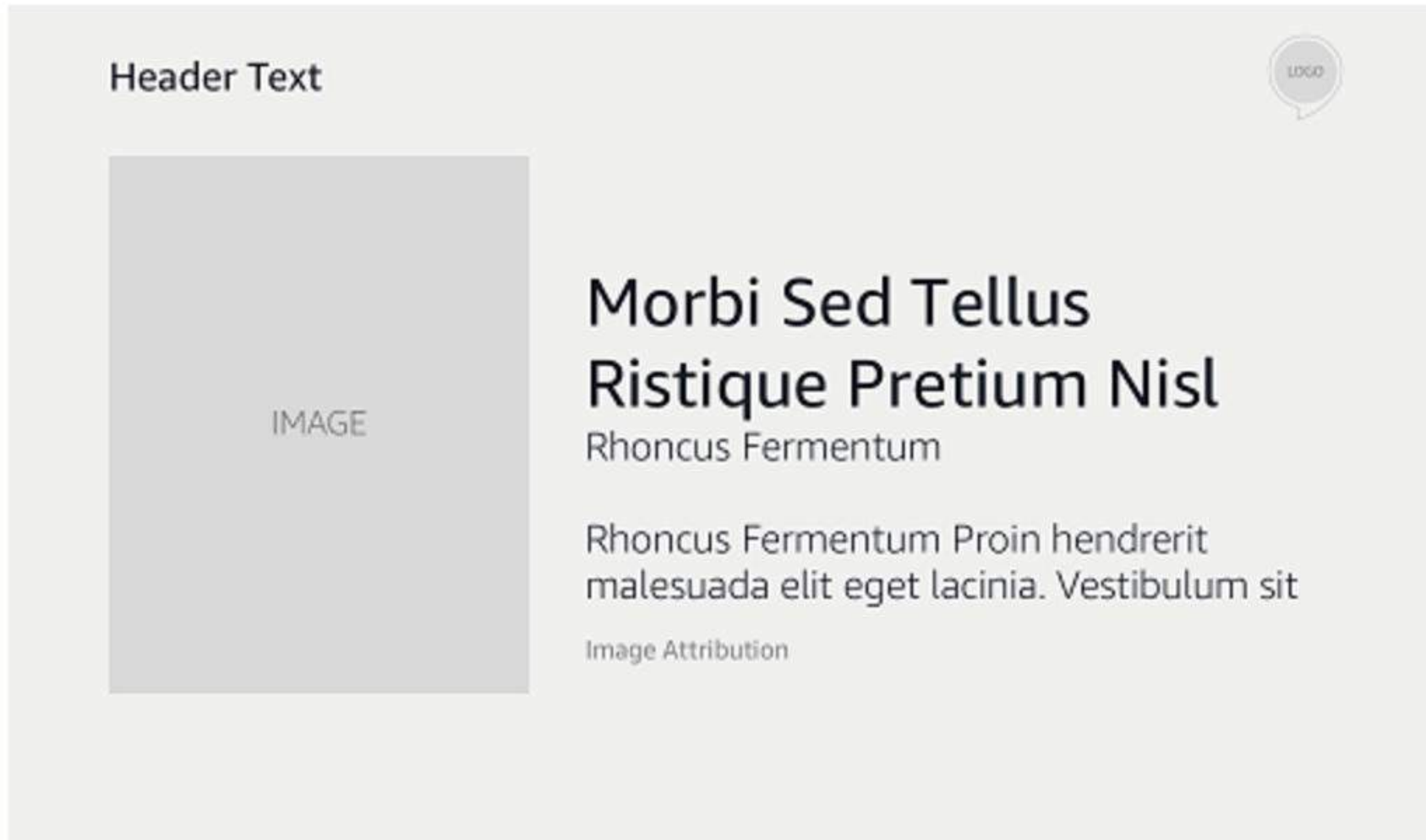
BodyTemplate1

BODYTEMPLATE2



BodyTemplate2

BODYTEMPLATE3



BodyTemplate3

BODYTEMPLATE 6-7



There are also more templates in the SDK,
BODYTEMPLATE6 and BODYTEMPLATE7

LISTTEMPLATE1

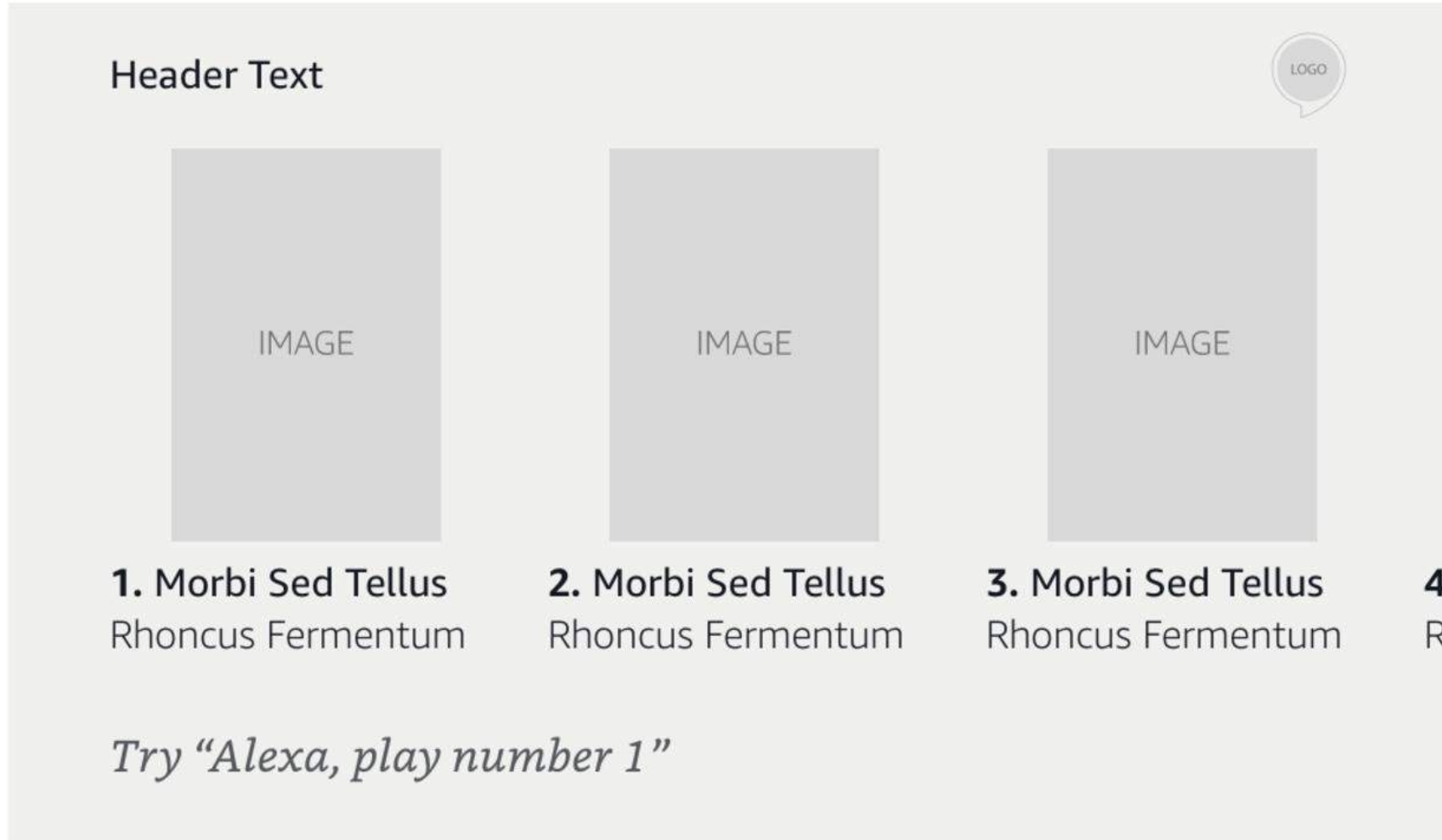
Header Text



- | | | |
|---|------------------|-----|
| 1 | Morbi Sed Tellus | 000 |
| 2 | Morbi Sed Tellus | 000 |
| 3 | Morbi Sed Tellus | 000 |
| 4 | Morbi Sed Tellus | 000 |

ListTemplate1

LISTTEMPLATE2



ListTemplate2

THE CODE



```
Template buildBodyTemplate1(String cardText) {  
    return BodyTemplate1.builder()  
  
        .withBackgroundImage(getImageInstance("https://media.giphy.com/media/  
YJNOIvcwG1NcY/giphy.gif"))  
        .withTitle("Unofficial Star Wars Quiz")  
        .withTextContent(getTextContent(cardText, cardText))  
        .build()  
}
```

SEE IT IN ACTION



Star Wars Quiz

PUBLISHING



In order to publish your app you must...

- Supply an icon of 108x108px and large icon 512x512
- Valid Recognized Cert (For non Lambda)
- Must have a VALID privacy policy and terms of use
- Must supply proper HELP anytime during session
- Must supply ability to stop skill by saying "STOP" or "CANCEL"
- Must be clear it doesn't violate any trademark or IP
- A real person will fire up the app and use it to test
- Process takes about 2-7 days for feedback

Here are some tips to help the user have a good experience

- Try to make sample utterances as specific as possible
- More sample utterances are better
- Use custom slots wherever possible
- Make sure Alexa always responds to a request when prompted
- Be Specific - Guide user during prompts tell the user what to say
- Use misspellings or phonetics in your code to help Alexa understand

Here are some gotchas

- Use number type for numbers- it will translated spoken word to numbers not words
- Sometimes you must have Amazon Literal Type to parse better
- Some invocation don't work - Hello, Amazon, Echo. Grails is often mis-interpreted
- SSL Self-signed certs - common name must match hostname of app
- SSML Audio - follow tips given, it's very picky and needs valid CA cert
- Audio player skills - audio content must be streamed in SSL as well slot values don't have to be complete just enough to guide it

SOURCES

- Micronaut Star Wars Quiz:
<https://github.com/rvanderwerf/micronaut-heroQuiz>
- Micronaut Hello World examples:
<https://github.com/micronaut-projects/micronaut-aws>
- Run Skills in Browser: <http://echosim.io>
- Amazon Developer Portal to register Skills: <https://developer.amazon.com>
- API v2 Migration Guide:
<https://alexa-skills-kit-sdk-for-java.readthedocs.io/en/latest/Migrating-To-ASK-SDK-v2-For-Java.html>

SPECIAL THANKS



- Graeme Rocher for fixing Alexa Micronaut issues
- Lee Fox for helping develop original Grails/Groovy talk
- Benoit Hédiard for his awesome Groovy Lamba code sample from GGX

THANK YOU

I HOPE YOU HAVE ENJOYED THE SESSION!



Free free to contact me on twitter
@RyanVanderwerf or email
vanderwerfr@objectcomputing.com



OBJECT
COMPUTING

CONNECT WITH US



1+ (314) 579-0066



@objectcomputing



objectcomputing.com