SAMSUNG

Bixby Developers

Port an Alexa Skill to Bixby

Office Hours - August 14, 2020

SAMSUNG

Bixby Developers

Introductions

@rogerkibbe

Roger Kibbe

- Senior Developer Evangelist
- Father of two daughters
- UC Berkeley Graduate – Go Bears!

@JohnWithoutTheH

Jonathan Pan

- Developer Evangelist
- Has 4 too many cats (meow)

The following slides are for the video:

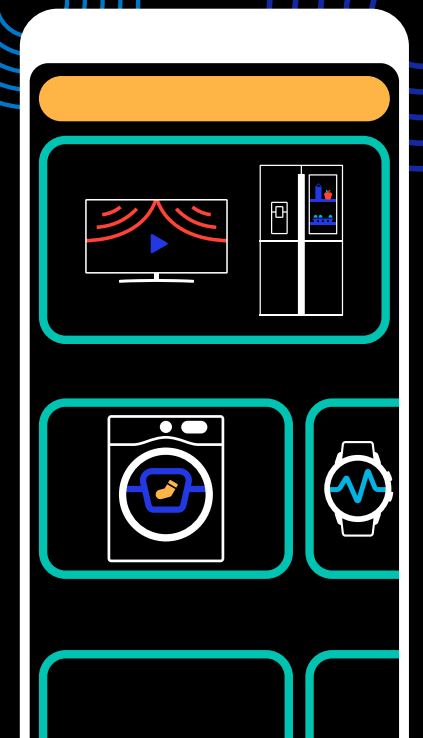
https://youtu.be/fgp5ACSfTxA

SAMSUNG

Bixby Developers

Port an Alexa Skill to Bixby

Roger Kibbe
Senior Developer Evangelist
Viv Labs / Samsung
@rogerkibbe



Pet Match Skill

alexa

github.com/alexa/
skill-sample-nodejs-petmatch

Bixby

github.com/bixbydevelopers/
capsule-sample-petmatch-port
(look in direct-port)

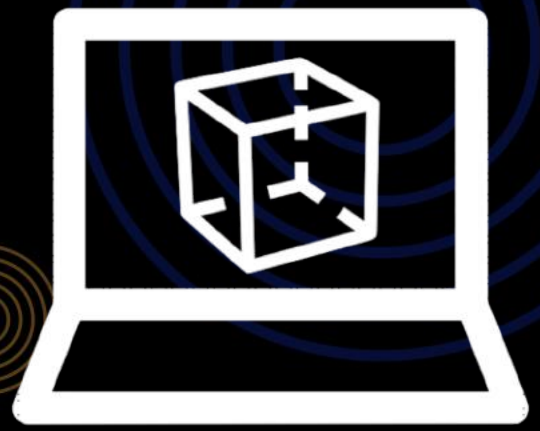
Why Pet Match?

- Popular Alexa Repo
- Highlights more advanced Alexa features such as Dialog Management & Entity Resolution
- Dialog Management similar to Bixby (Conceptually)
- Calls an API – common in Voice Apps
- Conversational, not overly simple, medium complexity – just complex enough

There is also an Alexa Conversations version of Pet Match which is likely a little closer to Bixby but given how new Alexa Conversations are and lack of developer experience with the the original Pet Match seemed like a better fit

Pet Match Demo

Bixby and Alexa Development Styles



Alexa*

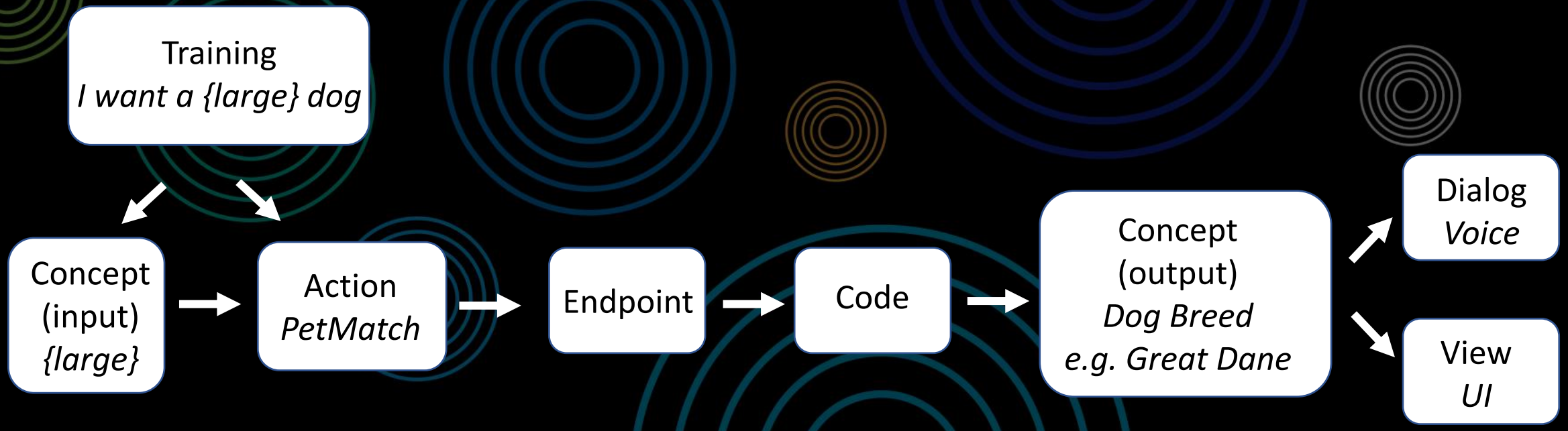
- Imperative
- Code Driven
- “How to do”

Bixby

- Declarative/Opinionated
- Model Driven
- “What to do”

* Alexa Dialog Management and the new Conversations are much more declarative

Bixby Components



Pet Match Basics – Slots/Concepts

- Uses 3 primary slots/concepts
  - size
  - temperament
  - energy
- Additional Slots
  - pet: Animal type (AMAZON.animal), used for humor
  - IWantType: Various ways to being phrase (not needed by Bixby NLU)
  - articleType: a, an, the (not needed by Bixby NLU)
  - atTheType: at the, around the, in the etc (not needed by Bixby NLU)
- Defined but not used by skill except for examples/training phrases
  - comparisonType: greater than/less than
  - amount: Number (AMAZON.NUMBER)
  - unitsType: Height/Weight
  - sheddingType: How much dog sheds
  - locationType: Various locations

Pet Match Basics – Intents/Actions

- PetMatch: Find an appropriate dog breed using these slots/concepts:
  - size
  - temperament
  - energy
- Help: Show Help
- Alexa Only: Cancel, Stop, Fallback
- Bixby Only: Intro

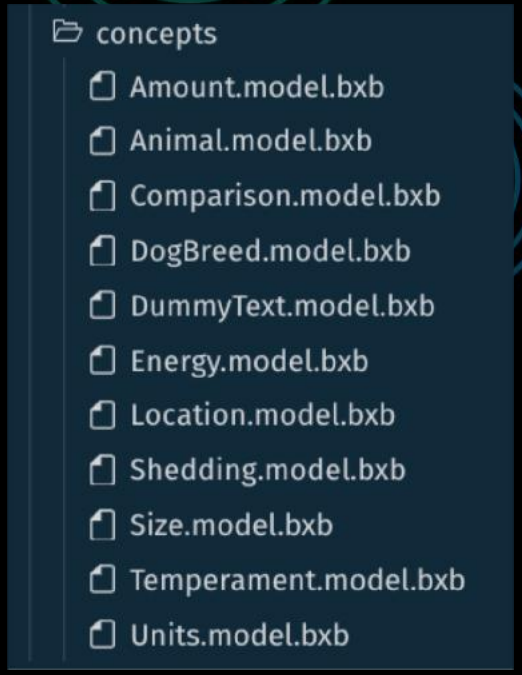
Slots and Concepts

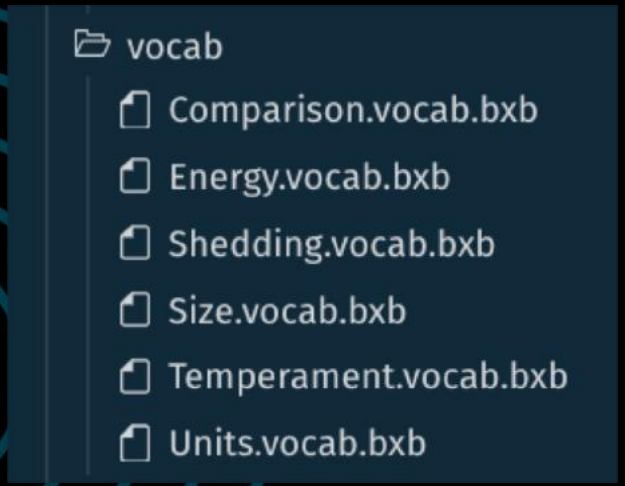
Alexa

- Single Interaction Model JSON file – en-US.json for US skill

Bixby

- Concepts in separate files
- Synonyms = Bixby Vocabulary in separate files





SAMSUNG

Bixby Developers

Slots and Concepts

Alexa

- temperamentType Slot Type
- temperament Slot

```
{
    "name": "temperamentType",
    "values": [
        {
            "name": {
                "value": "watch",
                "synonyms": [
                    "adult",
                    "barks at people"
                ]
            }
        },
        {
            "name": {
                "value": "guard",
                "synonyms": [
                    "to protect me",
                    "protective"
                ]
            }
        },
        {
            "name": {
                "value": "family",
                "synonyms": [
                    "is kid friendly",
                    "good with kids",
                    "family friends",
                    "gentle with kids"
                ]
            }
```

```
{
    "name": "temperament",
    "type": "temperamentType",
    "samples": [
        "{temperament} {pet}",
        "{size} {temperament} {at_the} {location}"
    ]
},
```

Bixby

- Temperament Concept
- Temperament Vocabulary

Temperament.model.bxb

```
enum (Temperament) {
  description (Pet temperament)
  symbol (watch)
  symbol (family)
  symbol (guard)
}
```

Temperament.vocab.bxb

```
vocab (Temperament) {
  "watch" {
    "watch"
    "adult"
    "barks"
    // "barks at people" - Not needed "barks" handles
  }
  "family" {
    "family"
    "kid"
    "kids"
    "family friends"
    // Below not need, "kids" handles
    //"good with kids"
    //"hang out with kids"
    //"gentle with kids"
  }
  "guard" {
    "guard"
    "protective"
    "protect"
    // "to protect me" - not need "protect" handles
```

Pro Tip: If you want to reuse concepts in Bixby (2 or more slots with the same type) use *role-of or* to extend, use *extends*

SAMSUNG

Bixby Developers

Slots and Concepts

Alexa	Bixby
values	enum
value	symbol
synonyms	vocab

```
{
    "name": "temperamentType",
    "values": [
        {
            "name": {
                "value": "watch",
                "synonyms": [
                    "adult",
                    "barks at people"
                ]
            }
        },
```

```
enum (Temperament) {
  description (Pet temperament)
  symbol (watch)
```

```
vocab (Temperament) {
  "watch" {
    "watch"
    "adult"
    "barks"
    // "barks at people" - Not needed "barks" handles
  }
```

Built In Slots and Concepts

```
{
    "name": "amount",
    "type": "AMAZON.NUMBER"
},
```

=

Amount.model.bxb

```
integer (Amount) {
  description (Amount)
}
```

- Bixby does not have all the rich List Types (Named Entities) that Alexa has
- Bixby Library Capsules provide some equivalence/different/enhanced functionality
  - viv.time, viv.money, viv.geo

Pro Tip: Become familiar with the Library capsules – using them can save significant development time and effort

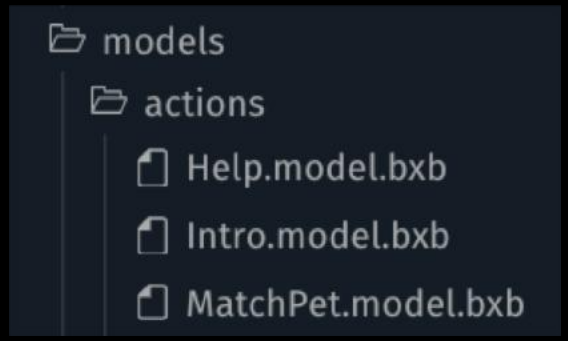
Intents/Actions

Alexa Intents:

- Intent definition, metadata and samples stored in single Interaction Model file e.g. en-US.json is a used for US skills
- Behavior in Node.js code (Dialog management has some behavior in the interaction model file)

Bixby Actions

- Separate files for each action
- Action = Definition and Metadata plus basic "housekeeping behavior
- NLU Training (samples) stored in separate training files



SAMSUNG

Bixby Developers

Alexa Intents

```
{
    "name": "PetMatchIntent",
    "slots": [
        {
            "name": "pet",
            "type": "AMAZON.Animal"
        },
        {
            "name": "size",
            "type": "sizeType",
            "samples": [
                "{I_Want} {article} {size} {pet}",
                "{I_Want} {size}",
                "{comparison} than a {size}",
                "the {size}",
                "Something i can {size}",
                "{size} size",
                "{I_Want} {article} {size} {pet} that {energy}",
                "{I_Want} {article} {size} {temperament} {pet}",
                "{I_Want} {article} {size} {temperament} to {energy}",
                " {temperament} {pet}",
                "{energy} energy",
                "{size}"
            ]
        },
        {
            "name": "temperament",
            "type": "temperamentType",
            "samples": [
```

```
"dialog": {
    "intents": [
        {
            "name": "PetMatchIntent",
            "confirmationRequired": false,
            "prompts": {},
            "slots": [
                {
                    "name": "pet",
                    "type": "AMAZON.Animal",
                    "confirmationRequired": false,
                    "elicitationRequired": false,
                    "prompts": {}
                },
                {
                    "name": "size",
                    "type": "sizeType",
                    "confirmationRequired": false,
                    "elicitationRequired": true,
                    "prompts": {
                        "elicitation": "Elicit.Intent-PetMatchIntent.IntentSlot-size"
                    }
                },
```

```
"prompts": [
    {
        "id": "Elicit.Intent-PetMatchIntent.IntentSlot-size",
        "variations": [
            {
                "type": "PlainText",
                "value": "There are dogs that are tiny, small, medium, and large. Which would you like?"
            },
            {
                "type": "PlainText",
                "value": "What size of a dog would you like?"
            }
        ]
    },
```

SAMSUNG

Bixby Developers

Bixby Actions

```
input (energy) {
  type (Energy)
  min (Required)
  max (One)
}
input (locationType) {
  type (LocationType)
  min (Optional)
  max (One)
}
input (sheddingType) {
  type (SheddingType)
  min (Optional)
  max (One)
}
```

```
action (MatchPet) {
  type (Search)
  description (Find a pet match)
  collect {
    input (animal) {
      type (Animal)
      min (Required)
      max (One)
      default-init {
        intent {
          value: Animal ("dog")
          goal: Animal
        }
      }
      validate {
       if (regexAllMatch(animal, 'unicorn|chimera|dragon')) {
          halt {
            dialog {
              choose (Random) {
                template ("I'm sorry, but I'm not qualified to match
                template ("Ah yes, #{value(animal)}s are splendid cre
                template ("I'm sorry I can't match you with a #{value
              }
            }
```

SAMSUNG

Bixby Developers

Slot Validation

Alexa: Validation done in code

```
{
    "name": "AMAZON.Animal",
    "values": [
        {
            "name": {
                "value": "mythical_creatures",
                "synonyms": [
                    "unicorn",
                    "chimera",
                    "dragon"
                ]
            }
        }
    ]
}
```

```
const MythicalCreaturesHandler = {
  canHandle(handlerInput) {
    if (handlerInput.requestEnvelope.request.type !== 'IntentRequest'
      || handlerInput.requestEnvelope.request.intent.name !== 'PetMat
      return false;
    }

    let isMythicalCreatures = false;
    if (handlerInput.requestEnvelope.request.intent.slots.pet
      && handlerInput.requestEnvelope.request.intent.slots.pet.resolu
      && handlerInput.requestEnvelope.request.intent.slots.pet.resolu
      && handlerInput.requestEnvelope.request.intent.slots.pet.resolu
      && handlerInput.requestEnvelope.request.intent.slots.pet.resolu
      && handlerInput.requestEnvelope.request.intent.slots.pet.resolu
      && handlerInput.requestEnvelope.request.intent.slots.pet.resolu
      const attributesManager = handlerInput.attributesManager;
      const sessionAttributes = attributesManager.getSessionAttribute
      sessionAttributes.mythicalCreature = handlerInput.requestEnvelo
      attributesManager.setSessionAttributes(sessionAttributes);
      isMythicalCreatures = true;
```

```
const slotsMeta = {
  pet: {
    invalid_responses: [
      "I'm sorry, but I'm not qualified to match you with {0}s.",
      'Ah yes, {0}s are splendid creatures, but unfortunately owni
      "I'm sorry I can't match you with {0}s.",
    ],
    error_default: "I'm sorry I can't match you with {0}s.",
  },
};
```

```
sMythicalCreatures;
```

Bixby: Validation done in Action (could also do in JS)

```
action (MatchPet) {
  type (Search)
  description (Find a pet match)
  collect {
    input (animal) {
      type (Animal)
      min (Required)
      max (One)
      default-init {
        intent {
          value: Animal ("dog")
          goal: Animal
        }
      }
      validate {
       if (regexAllMatch(animal, 'unicorn|chimera|dragon')) {
          halt {
            dialog {
              choose (Random) {
                template ("I'm sorry, but I'm not qualified to match
                template ("Ah yes, #{value(animal)}s are splendid cre
                template ("I'm sorry I can't match you with a #{value
              }
            }
```

Pro Tip: Become familiar with Bixby EL (Expression Language) – using it can simplify your overall capsule

SAMSUNG

Bixby Developers

Error Handling

Alexa: Error handling in code

```
    } else {
      outputSpeech = `I am sorry I could not find a match
        for a ${slotValues.size.resolved}
        ${slotValues.temperament.resolved}
        ${slotValues.energy.resolved} dog`;
    }
  } catch (error) {
    outputSpeech = 'I am really sorry. I am unable to access part of my
    console.log(`Intent: ${handlerInput.requestEnvelope.request.intent.n
  }
```

Bixby: Error thrown in code, handled in Action

```
} catch (error) {
  console.log ("Error: " + response)
  throw fail.checkedError(response,"APIError")
}
```

```
output (DogBreed) {
  throws {
    error (APIError) {
      on-catch {
        halt {
          dialog {
            template ("I am really sorry. I am unable to access part of
          }
        }
      }
    }
  }
  on-empty {
    halt {
      dialog {
        template ("I'm sorry I couldn't find a match for a #{value(size)
      }
    }
  }
}
}
```

SAMSUNG

Bixby Developers

Samples/Training

Alexa: In Slot definition

```
{
    "name": "PetMatchIntent",
    "slots": [
        {
            "name": "pet",
            "type": "AMAZON.Animal"
        },
        {
            "name": "size",
            "type": "sizeType",
            "samples": [
                "{I_Want} {article} {size} {pet}",
                "{I_Want} {size}",
                "{comparison} than a {size}",
                "the {size}",
                "Something i can {size}",
                "{size} size",
                "{I_Want} {article} {size} {pet} that {energy}",
                "{I_Want} {article} {size} {temperament} {pet}",
                "{I_Want} {article} {size} {temperament} to {energy}",
                " {temperament} {pet}",
                "{energy} energy",
                "{size}"
            ]
        },
        {
            "name": "temperament",
            "type": "temperamentType",
            "samples": [
```

Bixby: Separate training

DogBreed	I want a large dog	Learned · modified 23 hours ago
DogBreed	I want a small family dog	Learned · modified 23 hours ago
DogBreed	I want a family friendly dog that is fun to play with	Learned · modified 23 hours ago
DogBreed	I want a medium high energy dog	Learned · modified 23 hours ago

SAMSUNG

Bixby Developers

Bixby Training

playground.petmatch/Training

playground.petmatch / en

Run on Simulator

Cancel

Save

Goal

DogBreed

No specialization

I want a small family dog

Show Aligned NL

v: value

r: route

Role

Flag

Learned

Fit

Value Animal [1] dog

Required Energy

Value Size [1] small

Value Temperament [1] family

2 Optional Inputs

input 'animal'

input 'energy'

input 'size'

input 'temperament'

Search MatchPet

Annotations

small Size

family Temperament

dog Animal

Roles

No Roles found

Flags

No Flags found

Pro Tip: Do not over train your Bixby capsule. Start with a small training set and extend.

SAMSUNG

Bixby Developers

prompts & speak/View & Dialog

Alexa: Prompts in JSON (for Dialog) or in Code

```
"prompts": [
    {
        "id": "Elicit.Intent-PetMatchIntent.IntentSlot-size",
        "variations": [
            {
                "type": "PlainText",
                "value": "There are dogs that are tiny, small, medium, and
            },
            {
                "type": "PlainText",
                "value": "What size of a dog would you like?"
            }
        ]
    },
```

```
if (response.result.length > 0) {
  outputSpeech = `So a ${slotValues.size.resolved}
    ${slotValues.temperament.resolved}
    ${slotValues.energy.resolved}
    energy dog sounds good for you. Consider a
    ${response.result[0].breed}`;
} else {
  outputSpeech = `I am sorry I could not find a match
    for a ${slotValues.size.resolved}
    ${slotValues.temperament.resolved}
    ${slotValues.energy.resolved} dog`;
}
```

Bixby: Dialog in Message (in view) or in separate Dialog

```
input-view {
  match: Size

  message {
    choose (Random) {
      template ("There are dogs that are tiny, small, medium, and lar
      template ("What size of a dog would you like?")
    }
  }
}
```

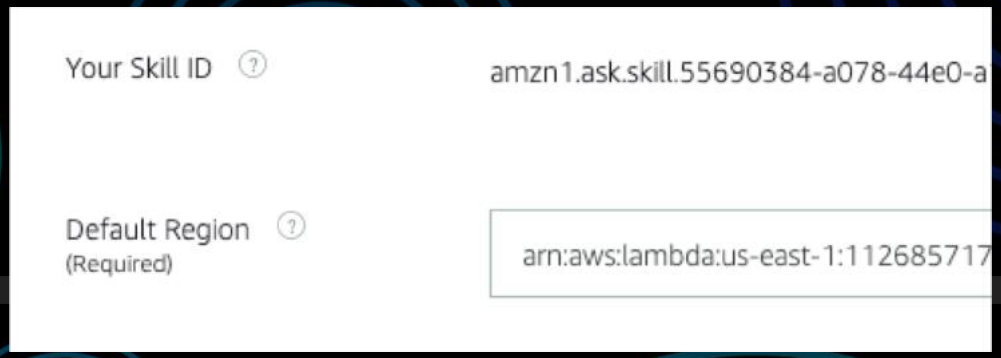
```
dialog (Result) {
  match: DogBreed (db) {
    from-output: MatchPet {
      from-input: Size (size)
      from-input: Energy (energy)
      from-input: Temperament (temperament)
    }
  }
  template("So a #{value(size)} #{value(temperament)} #{value(energy)} energy
}
```

SAMSUNG

Bixby Developers

Endpoint/Endpoints

Alexa: Define in Developer Console



Bixby: endpoint.bxb file

```
endpoints {
  action-endpoints {
    action-endpoint (MatchPet) {
      accepted-inputs (animal, energy, size, temperament, location, shedding)
      local-endpoint (MatchPet.js)
    }
  }
}
```

SAMSUNG

Bixby Developers

Code

Alexa: Lambda, most business logic is here e.g. imperative development.
Pet Match = 402 lines code

```
const Alexa = require('ask-sdk-core');
const https = require('https');

/* INTENT HANDLERS */

const LaunchRequestHandler = {
  canHandle(handlerInput) {
    return handlerInput.requestEnvelope.request.type === 'LaunchRequest';
  },
  handle(handlerInput) {
    return handlerInput.responseBuilder
      .speak('Welcome to pet match. I can help you find the best dog for you. ' +
        'What are two things you are looking for in a dog?')
      .reprompt('What size and temperament are you looking for in a dog?')
      .getResponse();
  },
};

const MythicalCreaturesHandler = {
  canHandle(handlerInput) {
    if (handlerInput.requestEnvelope.request.type !== 'IntentRequest'
      || handlerInput.requestEnvelope.request.intent.name !== 'PetMatchIntent') {
      return false;
    }

    let isMythicalCreatures = false;
    if (handlerInput.requestEnvelope.request.intent.slots.pet
      && handlerInput.requestEnvelope.request.intent.slots.pet.resolutions
      && handlerInput.requestEnvelope.request.intent.slots.pet.resolutions.resoluti
      && handlerInput.requestEnvelope.request.intent.slots.pet.resolutions.resoluti
```

Bixby: Part of Capsule, small, business logic in model, dynamic program generation e.g. declarative development
Pet Match = 37 lines of code

```
module.exports.function = function MatchPet(animal, energy, size, temperament, locationType
// animal used for validation, not in JS
// locationType and sheddingType in Alexa skill but not used for Alexa or Bixby
  var options = {
    format: "json",
    query: {
      "SSET": buildPetMatchParams(energy, size, temperament)
    }
  }

  try {
    // No promise or call back need, the Bixby JS API handles for you
    var response = http.getUrl(petMatchApi, options)
  } catch (error) {
    console.log ("Error: " + response)
    throw fail.checkedError(response,"APIError") // See the MatchPet action which handles t
  }

  //console.log ("response = " + JSON.stringify(response))
  if (response.result[0]) {
    return response.result[0].breed;
  } else return null;
}

function buildPetMatchParams(energy, size, temperament) {
  return "canine-" + energy + "-" + size + "-" + temperament
}
```

SAMSUNG

Bixby Developers

Metadata

Alexa: Interaction JSON and skill.json

```
"interactionModel": {
    "languageModel": {
        "invocationName": "roger's pet match",
```

```
{
  "skillManifest": {
    "publishingInformation": {
      "locales": {
        "en-US": {
          "summary": "Find the pet that is just right for you.",
          "examplePhrases": [
            "Alexa open pet match",
            "Alexa ask pet match for a recommendation",
            "Alexa tell pet match that I want a big fluffy dog"
          ],
          "name": "Pet Match",
          "description": "Do you want a pet, but aren't sure what pet
        }
      },
      "isAvailableWorldwide": true,
      "testingInstructions": "Sample Testing Instructions.",
      "category": "EDUCATION_AND_REFERENCE",
      "distributionCountries": []
```

Bixby: capsule-info.bxb, capsule.bxb and hints.bxb

```
capsule-info {
  display-name (Pet Patch Port)
  developer-name (Roger Kibbe - @rogerkibbe)
  description ("Alexa Pet Match skill ported to Bixby")
  dispatch-name (Pet Match)
  icon-asset (images/icons/bixby_launcher.png)
  search-keywords {
    keyword (pet)
    keyword (match)
  }
}
```

```
capsule {
  id (playground.petmatch)
  version (0.1.0)
  format (3)
  targets {
    target (bixby-mobile-en-US)
  }
  runtime-version (7)
  store-countries {
    all
  }
  store-sections {
    section(EducationAndReference)
```

```
hints {
  uncategorized {
    hint (Start Pet Match)
    hint (Ask Pet Match to find a large dog)
    hint (Ask Pet Match to find a family friendly dog)
  }
}
```

Pro Tip: Fill in all metadata and ensure it follows the standards – many capsules fail review because of metadata issues

SAMSUNG

Bixby Developers

UI

Alexa: APL

```
"layouts": {
  "SolarSystemPagerItem": {
    "parameters": ["distance", "planet"],
    "items": [
      {
        "type": "TouchWrapper",
        "width": "100%",
        "height": "100%",
        "onPress": [
          {
            "type": "SendEvent",
            "arguments": [
              "exploreEvent",
              "${planet ? planet.simple : data.simple}"
            ]
          }
        ],
        "item": {
          "type": "Container",
          "width": "100%",
          "height": "100%",
          "alignItems": "center",
          "inheritParentState": false,
          "items": [
            {
              "type": "Image",
              "source": "${planet ? planet.image : data.image}",
              "height": "100%",
              "width": "100%",
```

Bixby: Bixby Views

```
content {
  layout {
    section {
      content {
        image {
          url ("[#{value(movie.posterUrl)}]")
          aspect-ratio (3:4)
          object-fit (Contain)
          lightbox-enabled (true)
        }
        paragraph {
          value ("#{value(movie.title)}")
          style (Title_M)
        }
        spacer
        hbox {
          content {
            vbox {
              grow (1)
              halign (Center)
              content {
                text {
                  value ("Released")
                  style (Detail_L_Soft)
                }
                text {
                  value ("#{dateTime(movie.releaseDate, 'MMM dd, yyyy')}")
                  style (Title_S)
```

SAMSUNG

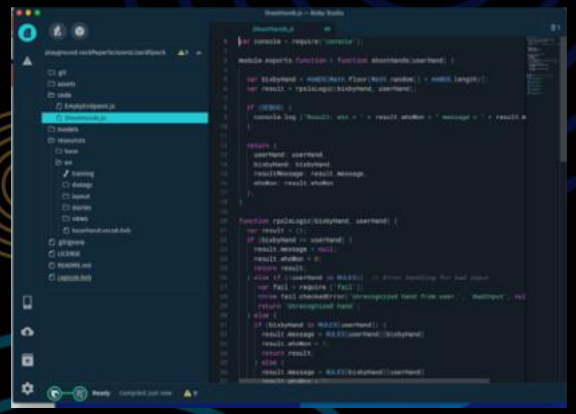
Bixby Developers

Development Environment

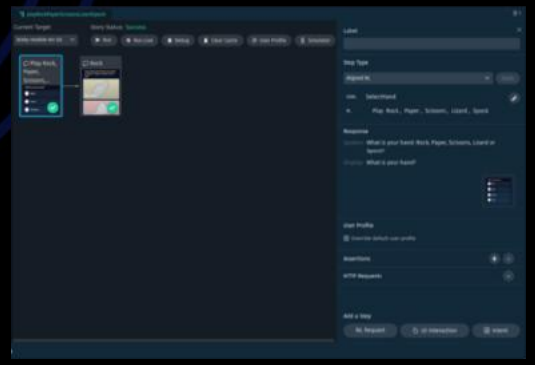
Alexa: Developer Console and/or code editor e.g. Visual Studio Code

- Model and UI defined in JSON files in Developer Console/Editor
- Code written in Lambda Function
- ASK CLI for local to remote sync and other functionality

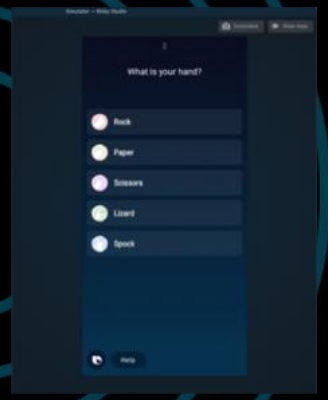
Bixby: Bixby Developer Studio
All in one IDE



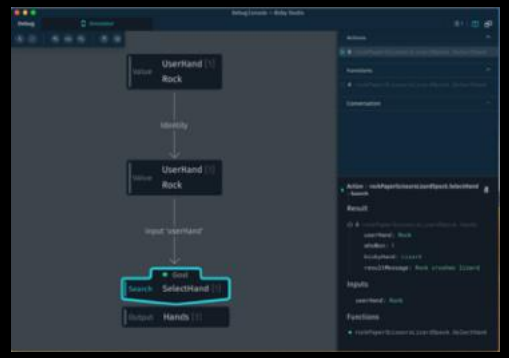
Develop



Test



Simulate



Debug

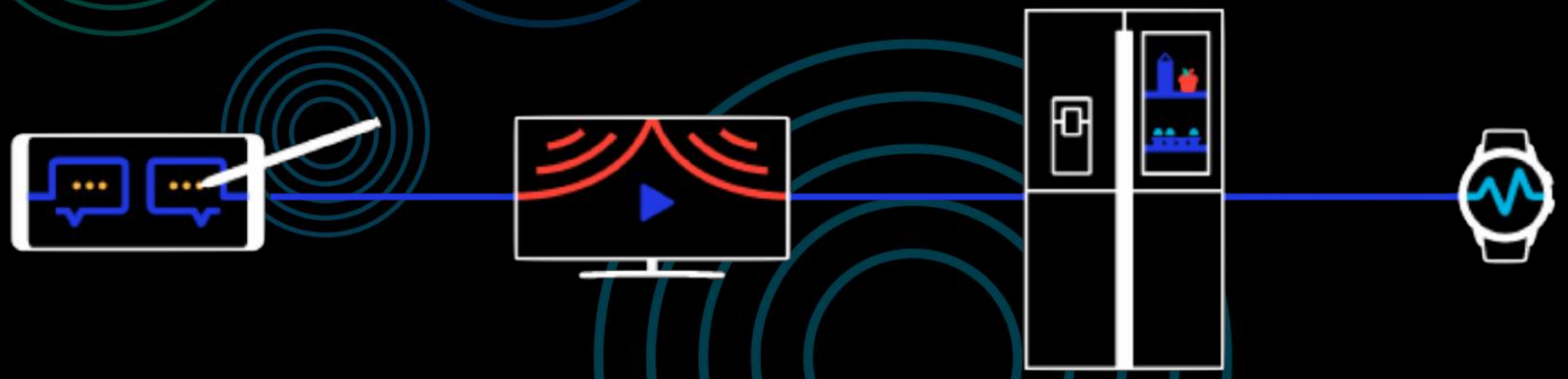
Summary

	alexa	Bixby
Voice Application	Skill	Capsule
Application Start Name	Invocation Name	Dispatch Name
NLU Training	Sample Utterances	Training
Context/Input Objects	Slot	Concept
Goal/Intent	Intent	Action
Mapping Intent to Code	Handler	Endpoint
Code/Business Logic	Lambda	JavaScript Function
Output Speech	Speak/Reprompt	Dialog
UI	Card/APL	View

SAMSUNG

Bixby Developers

Beyond the Smart Speaker
A Call to Action



Bixby Developers Resources

bixbydevelopers.com

github.com/bixbydevelopers

www.youtube.com/c/BixbyDevelopers

bixbydev.buzzsprout.com

@BixbyDevelopers

facebook.com/BixbyDevelopers

Roger Kibbe

@rogerkibbe

www.linkedin.com/in/rkibbe

End of Slides for Video

AMA Time

Roger Kibbe

@rogerkibbe

www.linkedin.com/in/rkibbe

Jonathan Pan

@JohnWithoutTheH

www.linkedin.com/in/jonmpan