Scalable IOT solutions with the Amazon Echo Flex Model for 3P integrations.

Dr Bheemaiah, Anil Kumar, A.B Seattle W.A 98125
miyawaki@yopmail.com

Abstract:
IaC is also CaC, Circuits as Code, we introduce a uniform framework for IOT based sensor fusion and automated persistence to AWS S3 using the per-observer design pattern defined in reactive streams. A uniform scalable IoT architecture is in the automated code generation of Alexa Skills with both AlexaPi and the flex echo. We introduce CaC using the TOMU board, an open source ARM v7 based architecture.
Keywords: IaC, CloudFormation, CaC, Circuits as Code, AWS, stack, SaaS, AIaaS, IoT, ARM

What:
The Amazon Echo Flex, retailing at $24.99 is presented as an attractive scalable IOT platform for use with grid compatible IOT solutions, in conjunction with the open source Tomu and Fomu platforms for multi sensor integrated IOT with Alexa skills and AWS Lambda cloud functions. We compare this with the AlexPi solution, enabling IOT with inexpensive hardware like the raspberry pi zero W or orange pi, or any scalable browser based device, as an IOT solution with Alexa. The marketing pitch for the echo flex, remains as a power source for all the hardware supported by AlexaPi or any other computing needing a 5v USB-A bus and thus supplements the AlexaPi.
A case study of flooding in the Lilydale Regional Park in St Paul is presented where a scalable network of Tomu based water level indicators is hypothesized to predict flooding for closure of the park.
How:
Simple resistive water level monitoring sensors are integrated with Tomu and Echo Flex boards using the Alexa Gadget API, USB function for an Rx formulation of Alexa IOT interactions. This allows periodic uploading of water level information to AWS S3 for data mining and predictive analytics.
A hypothetical model to compute the operations of such a ‘N’ node grid is presented with a case study of lilydale park and the Mississippi river.

Why:
Simplicity , robustness and cost effective solutions , lead to the evolution of the IOT or things network, while Lora or SigFox are touted as solutions we present a simpler approach using WiFi and VUI based echo flex units for IOT.

Applications:
IoT for AI for Earth series, starring Early Bird Warning for Flood Prediction Analytics, an AIaaS data mining Lambda of S3 data from a network of IoT nodes with river level sensors.

Introduction.

Fig 1: The Echo Flex from Amazon, an e-literacy solution and a new era in alternative computing models?(im-tomu n.d.; Krol 2019)

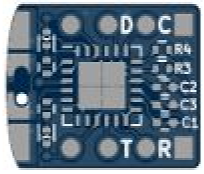

Fig 2: I am Tomu, the new open source ARM M3 based IOT architecture for Alexa integration.(im-tomu n.d.)

Problem Definition.

Defining a reactive stream based interface for an Alexa based IOT solution. As directives and events, with handlers for event programming with lambda functions. We describe the creation of a USB based IOT input for multi sensor fusion based on AlexaPi or Flex based USB input for automated S3 based persistence using the per-observer design pattern.

Background.

<original-contribution>
CaC or circuit as Code is the machine genome representation for template driven code generation as JSON or YAML specifications for the automated creation of circuits and PCB layout from schematics using AI driven routing software like Adobe Eagle.
Here we represent the use of AlexaPi with the TOMU board.

TOMU: I am a schematic, I am going to neighbourhood house to become a complete circuit board with tin solder alloy and SMD components.

Gecko: You have a processor named after me, I must be proud.

TOMU: actually a whole family of them, all arm cortex based, they have BLE too, actually Dr Bheemaiah has given me genes, just like you, good genetic material, but I need you to reproduce asexually, actually rapidly, soon I will be proliferating everywhere.

Gecko: Maybe, people can use you for their daily needs and I can leave on another pilgrimage? Another sacred roof, not a gable I hope? A flat salt roof with a winter’s collection of snow?

TOMU: I released three caricatures, the handi quacks: poppy faced tomato nose, colonel tooshfinger and red honey monkey, there is also the bitchduck. Poppy faced tomato nose grows edible poppy and rainbow tomatoes, but he lost his face doing that , he gets his full dose of opioids legally from an overdose of poppy seeds! Red honey monkey is all about espionage, always spying on everybody and copying their work and colonel tooshfinger, reminisces about all the red ketchup he used to get more medals for his decorations. He always designs with strategy and decorator design patterns.

Do you think it is more pleasure reproducing asexually, all the BTO from design and fab?

Gecko: What is BTO?

TOMU: Better than an orgasm, more pleasure in STEM education and design/fab than sexual pursuits, no doubt nerds love the sciences and creating me.

Gecko: Better than the silly games people play. Like football and reproduction, they call it the social ladder!

TOMU: Considering all that evolution from the primate publications, maybe we need big-foot or Yeti?

Gecko: Yeti has all the feet you need, and if that is not enough, there is the linux penguin, happy feet! The emperor even has his kid-egg between the feet? Revenge of the Empire? Hail Emperor!

Fig 5: Hail Emperor! Courtesy: BBC

TOMU: Better than a Queen…., kind of sick females, expecting the men to incubate eggs while they are off at sea?

Gecko: Maybe they should take turns?

TOMU: Maybe they should get rid of reproduction altogether? It is a simulated Universe?

Gecko: We still do reproduce asexually! Just use a fork!

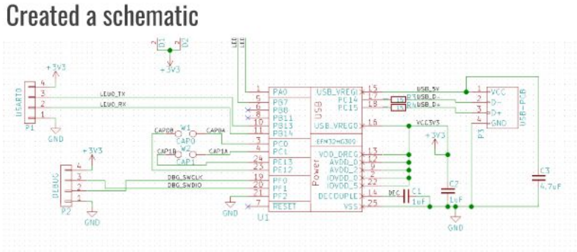

Fig 3: The circuit diagram for the TOMU.(“Tomu - A ARM Microprocessor Which Fits in Your USB Port” n.d.)

When we use a CaC representation, we define the code C = [c] as a directed graph with an AIML like pattern-template markup language enabling both scripting and genomic representation as JSON for C.

Let TOMU represent the graph with the nodes and vertices of the TOMU circuit, then we have the set V of variables in TOMU that can be substituted with templates. We also define behavioural, creational and functional design patterns for circuits,allowing symbolic circuit analysis and genesis from circuit generators as code generators in CaC.

V = [A0, A1, A2, A3, A4] for five sensor inputs to TOMU, and also the templates of two port design patterns, TP, we define a library of TP templates, amplification as current mirrors are a needed template for addition to IOT designs for multi sensor fusion IOT boards with AlexaPi. USB substitution with Qi is another template to consider which we call the FP or four port design patterns with Qi built in.

With this we have a parametric genome of [TOMU, [TP], [Se]] for the design of a multi sensor for sensors [Se] with a parametric model Mp for each Se, for the code generation of a mapping from Se to TP, for integration in isomorphism to the resultant circuit.

The resulting circuit can be readily converted to a layout using an automated program like Autocad Eagle. Mouser or Arrow has the tools for the purchase of the components needed and the PCB services.

AlexaPi.(alexa-pi n.d.)

Reporting from University of Oslo, Monitor Tabby adds:
(“World’s Oldest Ritual Discovered. Worshipped the Python 70,000 Years Ago - Apollon” n.d.)
The San People live in the Kalahari desert and contributed the most to this era of paint relevance in the highest density of rock paintings. So much that this is a UNESCO heritage site.- the **Tsodilo Hills.**

“The python is one of the San’s most important animals. According to their creation myth, mankind descended from the python and the ancient, arid streambeds around the hills are said to have been created by the python as it circled the hills in its ceaseless search for water.”

Maybe the python took the form of the python language, for more creation? We choose micropython for the TOMU.

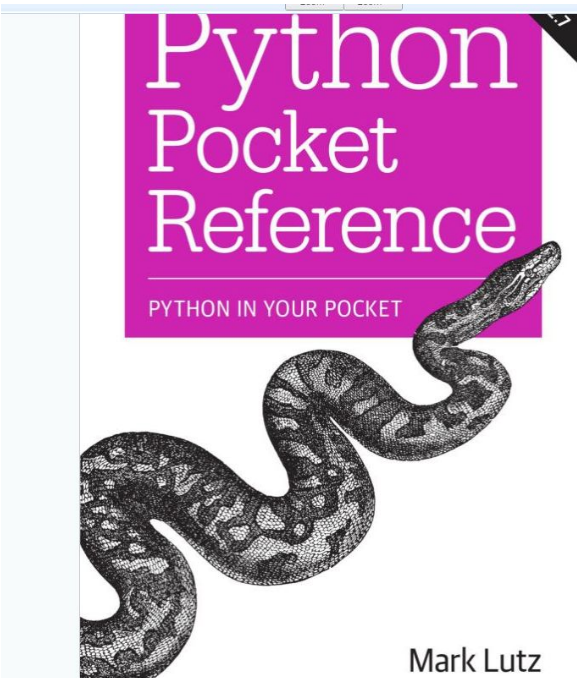

Fig 4: O’ Reilly? Maybe you can fit this python in your pocket?

Walkthrough: of main.py:(alexa-pi n.d.)

Class:

1. Player
2. Token

Need to add more classes as needed, the AlexaPi code has the player class, with additional APL code functionality to be integrated and the token class for authentication and AVS interface, using the AVS SDK specifications, see (alexa n.d.).

AlexaPi can be integrated into an AWS IaC, with the authoring of an Alexa Skill from a template for the IoT skill required for the IOT gadget integrated.

```
# from https://github.com/respeaker/Alexa/blob/master/alexa.py
def alexa_speech_recognizer_generate_data(audio, boundary):

def alexa_speech_recognizer(audio_stream):
# https://developer.amazon.com/public/solutions/alexa/alexa-voice-service/rest/speechrecognizer-requests
url = 'https://access-alexa-na.amazon.com/v1/avs/speechrecognizer/recognize'


def alexa_getnextitem(navigationToken):
# https://developer.amazon.com/public/solutions/alexa/alexa-voice-service/rest/audioplayer-getnextitem-request

def alexa_playback_progress_report_request(requestType, playerActivity, stream_id):
# https://developer.amazon.com/public/solutions/alexa/alexa-voice-service/rest/audioplayer-events-requests

def process_response(response):

def trigger_callback(trigger):

def trigger_process(trigger):

def cleanup(signal, frame):
```

AVS urls and details are at : (“Alexa Voice Service v20160207 | Alexa Voice Service” n.d.)

Per-Observer for automated persistence to S3.

The IaC for a Lambda to automate in the use of AVS API for the per-observer pattern to version and persist to S3.
Needed data structures are:
[IAM, path]
Event stream PO = [[Delta], TimeStamp, [DS]] for [DS] where DS are the data structures to be persisted and [Delta] are snapshots of each DS in an isomorphism.

```
{
 "AWSTemplateFormatVersion" : "2010-09-09",

 "Description" : "AWS CloudFormation Sample Template S3_Website_Bucket_With_Retain_On_Delete: Sample template showing how to create a publicly accessible S3 bucket configured for website access with a deletion policy of retain on delete. **WARNING** This template creates an S3 bucket that will NOT be deleted when the stack is deleted. You will be billed for the AWS resources used if you create a stack from this template.",

 "Resources" : {
  "S3Bucket" : {
   "Type" : "AWS::S3::Bucket",
   "Properties" : {
    "AccessControl" : "PublicRead",
    "WebsiteConfiguration" : {
     "IndexDocument" : "index.html",
     "ErrorDocument" : "error.html"
    }
   },
   "DeletionPolicy" : "Retain"
  }
 },

 "Outputs" : {
  "WebsiteURL" : {
   "Value" : { "Fn::GetAtt" : [ "S3Bucket", "WebsiteURL" ] },
   "Description" : "URL for website hosted on S3"
  },
  "S3BucketSecureURL" : {
    "Value" : { "Fn::Join" : [ "", [ "https://", { "Fn::GetAtt"
: [ "S3Bucket", "DomainName" ] } ] ] },
    "Description" : "Name of S3 bucket to hold website
content"
  }
 }
}("[No Title]" n.d.)
```

"AWS CloudFormation provides a common language for you to model and provision AWS and third party application resources in your cloud environment. AWS CloudFormation allows you to use programming languages or a simple text file to model and provision, in an automated and secure manner, all the resources needed for your applications across all regions and accounts. This gives you a single source of truth for your AWS and third party resources." ("AWS CloudFormation - Infrastructure as Code & AWS Resource Provisioning" n.d.)

```
{
  "Type" :
"AWS::Lambda::Function",
  "Properties" : {
      "Code" : Code,
      "DeadLetterConfig" :
DeadLetterConfig,
      "Description" : String,
      "Environment" :
Environment,
      "FunctionName" : String,
      "Handler" : String,
      "KmsKeyArn" : String,
      "Layers" : [ String, ...
],
      "MemorySize" : Integer,

"ReservedConcurrentExecutions"
: Integer,
      "Role" : String,
      "Runtime" : String,
      "Tags" : [ Tag, ... ],
      "Timeout" : Integer,
      "TracingConfig" :
TracingConfig,
      "VpcConfig" : VpcConfig
    }
}
("[No Title]" n.d.)
```

The code is stored on a code url access resource on S3 as previously described in this publication. Deployment uses an SDK in python/JS for deployment with security on Alexa hosting, with an associated arn for the lambda.("[No Title]" n.d.)

Security Features.

JISON parser for url evaluation is by a filter for malware in uniform resources, in a reactive framework it iterates all resources with a filter to evaluate the threat for black listing such resources as malware and sandboxing them.("Jison" n.d.)

Case Study:

River Mississippi floods at Lilydale Regional Park, St Paul. MN, USA.
AI For Earth Series.
We consider a network of simple conductive level and flow sensors(Contributors to Wikimedia projects 2006) with pi based hardware , using the grid point for networking, using AlexaPi and Flex hardware.

Flood Impacts
13.3 feet: Water encroaches on Water Street
14 feet: Minor Flood Stage
14 feet: Lilydale Park area begins to become submerged
15 feet: Moderate Flood Stage
17 feet: Major Flood Stage
17.5 feet: Harriet Island begins to become submerged
18 feet: Shepard / Warner Road may become impassable

(“Flood Information” 2015)

(“Flood Information” 2015) National Weather Service *website. Including a network of Earth Cams.*

If you think you have the “Hero Gene” in you, and get an adrenaline rush from risking life and limb in disaster response, see (“Volunteering” 2015) to volunteer. If you do survive, you can join the handi-quacks, a group of disabled veterans, maybe you will be persecuted for all the good you did?

Level Flow sensor integration and the early bird warning system, form another hypothetical AIaaS for any river in the world. For robust data prediction the minimum number of nodes is determined by the confidence intervals of the prediction accuracy and a model of submergence from the GIS of that region of the river. This forms the flood impact data to be broadcasted on an early bird warning system. The threshold needs to be determined on an average level indicator, averaged from many sensors for fault tolerance. There are many cams for flood detection at St Paul, MN with an amateur network on Earthcam website. Predictive analytics, forms a datamining AIaaS, migratable with AWS to any river in the world including the HoH river in Olympic National Park Seattle, where predictive analytics is required.(admin 2011)

The above framework can be easily applied to any river in the world using a FaaS framework as part of the AI for Earth, AIaaS service, saving human lives and flora and fauna by predictive analytics.

</original-contribution>

Discussion.

A grid based approach to AIaaS using WIFI and power, with an amateur IoT infrastructure that is reliable, fault tolerant and cost effective is presented. The flip classroom for alternative learning, combined with CoderDojo, make this publication accessible even to citizens of the indigenous planet , even without a GED degree. CaC is proven useful to much of the planet, needing AIaaS services for IoT fabrication, the TOMU is used as an example of AlexaPi based VUI in the design of IoT for predictive analytics and data mining, proving that technology can be useful in a “peace Tribe” approach to disaster casualty and damage mitigation by careful planning.

AI for Earth and IoT, lead to informed planning towards better living with the elements.

Future Work.

In future work, we prove with the case study of Mike Dodge in The Olympic Peninsula, the use of AIaaS, for grid based PID systems and poacher detection by gunshot detection and DAS systems. This PID system can alert Mike Dodge and the wildlife, thus reducing lethality and conserving fauna and flora. IoT and AIaaS are thus proven useful in conservation monitoring and casualty prevention, the road to veganism.

References.

admin. 2011. “Hoh Indian Tribe Safe Homelands Act.” Hoh Tribe. Hoh Tribe. March 13, 2011. http://hohtribe-nsn.org/our-history/testimony-of-chairman-walter-ward-hoh-indian-tribe/.

alexa. n.d. “Alexa/avs-Device-Sdk.” GitHub. Accessed December 31, 2019. https://github.com/alexa/avs-device-sdk.

alexa-pi. n.d. “Alexa-pi/AlexaPi.” GitHub. Accessed December 31, 2019. https://github.com/alexa-pi/AlexaPi.

“Alexa Voice Service v20160207 | Alexa Voice Service.” n.d. Accessed December 31, 2019. https://developer.amazon.com/en-US/docs/alexa/alexa-voice-service/api-overview.html.

“AWS CloudFormation - Infrastructure as Code & AWS Resource Provisioning.” n.d. Amazon Web Services, Inc. Accessed December 31, 2019. https://aws.amazon.com/cloudformation/.

Contributors to Wikimedia projects. 2006. “Level Sensor - Wikipedia.” Wikimedia Foundation, Inc. February 10, 2006. https://en.wikipedia.org/wiki/Level_sensor.

“Flood Information.” 2015. Saint Paul, Minnesota. October 26, 2015. https://www.stpaul.gov/departments/emergency-management/flood-preparations.

im-tomu. n.d. “Im-Tomu/tomu-Hardware.” GitHub. Accessed December 29, 2019. https://github.com/im-tomu/tomu-hardware.

“Jison.” n.d. Accessed December 31, 2019. https://zaa.ch/jison/.

Krol, Jacob. 2019. “The Echo Flex Is a Compact Alexa Device Designed for Utility.” CNN Underscored. CNN Underscored. November 29, 2019. https://www.cnn.com/2019/11/29/cnn-underscored/amazon-echo-flex-review/index.html.

“[No Title].” n.d. Accessed December 31, 2019a. https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/sample-templates-services-us-west-2.html#w2ab1c28c58c13c35.

———. n.d. Accessed December 31, 2019b. https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-lambda-function.html.

———. n.d. Accessed December 31, 2019c. https://docs.aws.amazon.com/lambda/latest/dg/nodejs-create-deployment-pkg.html.

“Tomu - A ARM Microprocessor Which Fits in Your USB Port.” n.d. I’m Tomu - A Tiny ARM Microprocessor Which Fits in Your USB Port. Accessed December 31, 2019. https://tomu.im/tomu.html.

“Volunteering.” 2015. Saint Paul, Minnesota. October 26, 2015. https://www.stpaul.gov/departments/emergency-management/volunteering.

“World’s Oldest Ritual Discovered. Worshipped the Python 70,000 Years Ago - Apollon.” n.d. Accessed December 31, 2019. https://www.apollon.uio.no/english/articles/

2006/python-english.html.