T-REX

Elkhan is a principal cloud architect and 14 times MSFT MCT with experience in cloud solutions, enterprise healthcare systems and healthcare data, and API standards (HL7 FHIR, EDI X12).

He is an active MCT, a Microsoft Tech Community Contributor and Azure SME (Architecture, DevOps, Security).

He is a frequent speaker at tech meetups and has volunteered for Give Camps, Azure Data Fest, SQL Saturday and Global Azure events.

The Cloud Marathoner:

www.thecloudmarathoner.com

Elkhan Yusubov
Principal Cloud Architect

@ElYusubov

ElkhanYusubov

ElYusubov

aws

3

Company Snapshot: T- Rex Is…

▸ **An Innovative NGIT Provider:** a leading solutions provider of innovative and modernizing Next Generation IT capabilities and enabling skills to the federal government

▸ **A first to market in migrating large federal mission applications and data** into a citizen-facing public cloud, giving the Company the capabilities and past performance to bid and win future NGIT opportunities in adjacent federal agencies

▸ **A Prime Contractor** on **2020 Census,** one of the largest (scale) and most complex mission critical cloud/cyber/data analytics integration projects within the federal government

Concurrent Web Respondents (Peak)	Mobile Census Enumerators	Regional & Local Census Offices	Business Systems	Application Components
👤 600,000+	📱 500,000+	🏢 254 / 20K Workers	52	285+

▸ **Integrated 52 mission-critical systems comprised of 200+ applications** operating in a hybrid hyper-converged data center and Amazon Web Service ("AWS") public facing GovCloud protected by a world class active cyber defense (ACD) solution. Also includes a VDI-based Office 365 solution supporting 40k+ users and MS Azure AD managing 350k+ mobile users

Achievements & Partnerships

Silver Microsoft Partner

Silver Data Analytics
Silver Application Development
Silver Data Platform

aws partner network
Advanced Consulting Partner
Public Sector Partner

ISO 9001:2015 BUREAU VERITAS Certification

ISO 27001 BUREAU VERITAS Certification

CMMIDEV/3
Exp. 2022-03-15 / Appraisal #2472

CMMISVC/3
Exp. 2022-03-15 / Appraisal #2472

#1 Bloomberg Government Cloud Government Support Services PROVIDER 2016-20

Washington Technology 2020 Top 100 Government Contractors, Federal Market T-REX ▸ #41
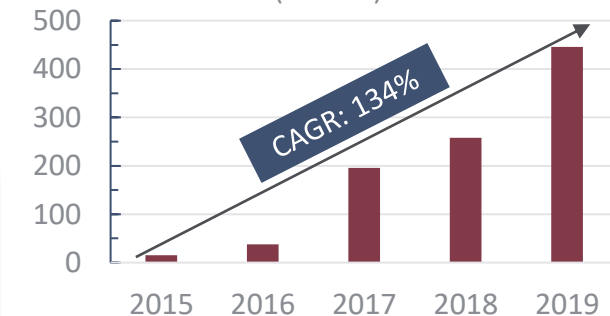
TOP WORK PLACES 2020 The Washington Post
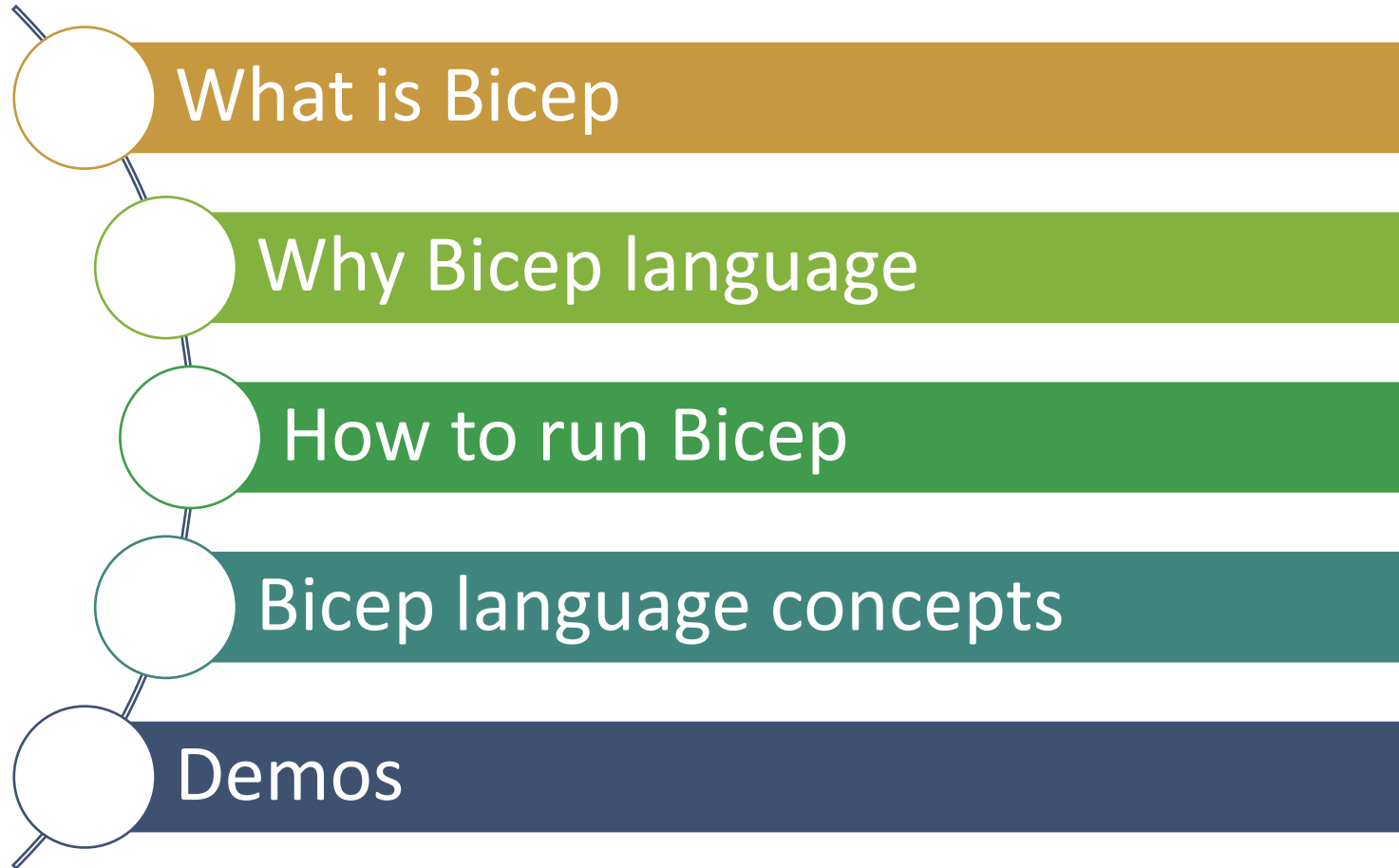
Company Info

- **Established:** 1999
- **Legal Structure:** LLC
- **Headquarters:** Greenbelt, MD
- **8(a)/HUBZone & SDVOSB Joint Ventures:** Ability to deliver services as a small business concern
- **Prime/**Sub (09/20): 98.5% Prime
- **Backlog** (09/20): $341.6MM

Prime Vehicles/BPAs:
- GSA Schedule 70
- USDA Cloud COE BPA
- Dept of Commerce ESF

Financial Summary

T-Rex Annual Revenue
(USD M)

CAGR: 134%

	2015	2016	2017	2018	2019

(Y-axis: 0, 100, 200, 300, 400, 500)

What we will cover

- What is Bicep
- Why Bicep language
- How to run Bicep
- Bicep language concepts
- Demos

T-REX

What we will not cover

ARM templates

CI/CD in piplines

3-rd party support

Integration topics

What is Bicep?

- DSL for deploying Azure resources

- Transpiler of code into ARM template

- Simple way to author Azure resources

T-REX

Why Bicep language?

- Easy to understand and maintain

- Day zero support

- Transparent abstraction

- Awesome tolling in VS Code

- Clean code syntax

- Code re-use and modularity

- Depp integration with Azure

- Preflight validation

- No state management

- Production support


```bicep
param appName string = uniqueString(resourceGroup().id)
param accountName string = toLower('cosmos-${appName}')
param location string = resourceGroup().location
param databaseName string = toLower('db-${appName}')

resource cosmos 'Microsoft.DocumentDB/databaseAccounts@2020-04-01' = {
  name: accountName
  location: location
  properties: {
    enableFreeTier: true
    databaseAccountOfferType: 'Standard'
    consistencyPolicy: {
      defaultConsistencyLevel: 'Session'
    }
    locations: [
      {
        locationName: location
      }
    ]
  }
}

resource cosmosdb 'Microsoft.DocumentDB/databaseAccounts/sqlDatabases@2020-0
  name: '${cosmos.name}/${databaseName}'
  properties: {
    resource: {
      id: databaseName
    }
    options: {
      throughput: 400
    }
  }
}
```

```json
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploym
  "contentVersion": "1.0.0.0",
  "metadata": {
    "_generator": {
      "name": "bicep",
      "version": "0.3.255.40792",
      "templateHash": "17325859092750274113"
    }
  },
  "parameters": {
    "appName": {
      "type": "string",
      "defaultValue": "[uniqueString(resourceGroup().id)]"
    },
    "accountName": {
      "type": "string",
      "defaultValue": "[toLower(format('cosmos-{0}', parameters('appName')))
    },
    "location": {
      "type": "string",
      "defaultValue": "[resourceGroup().location]"
    },
    "databaseName": {
      "type": "string",
      "defaultValue": "[toLower(format('db-{0}', parameters('appName')))]"
    }
  },
  "functions": [],
  "resources": [
    {
      "type": "Microsoft.DocumentDB/databaseAccounts",
      "apiVersion": "2020-04-01",
      "name": "[parameters('accountName')]",
      "location": "[parameters('location')]",
      "properties": {
        "enableFreeTier": true,
        "databaseAccountOfferType": "Standard",
        "consistencyPolicy": {
          "defaultConsistencyLevel": "Session"
        },
        "locations": [
          {
```

T-REX

Relationship between Bicep and ARM templates
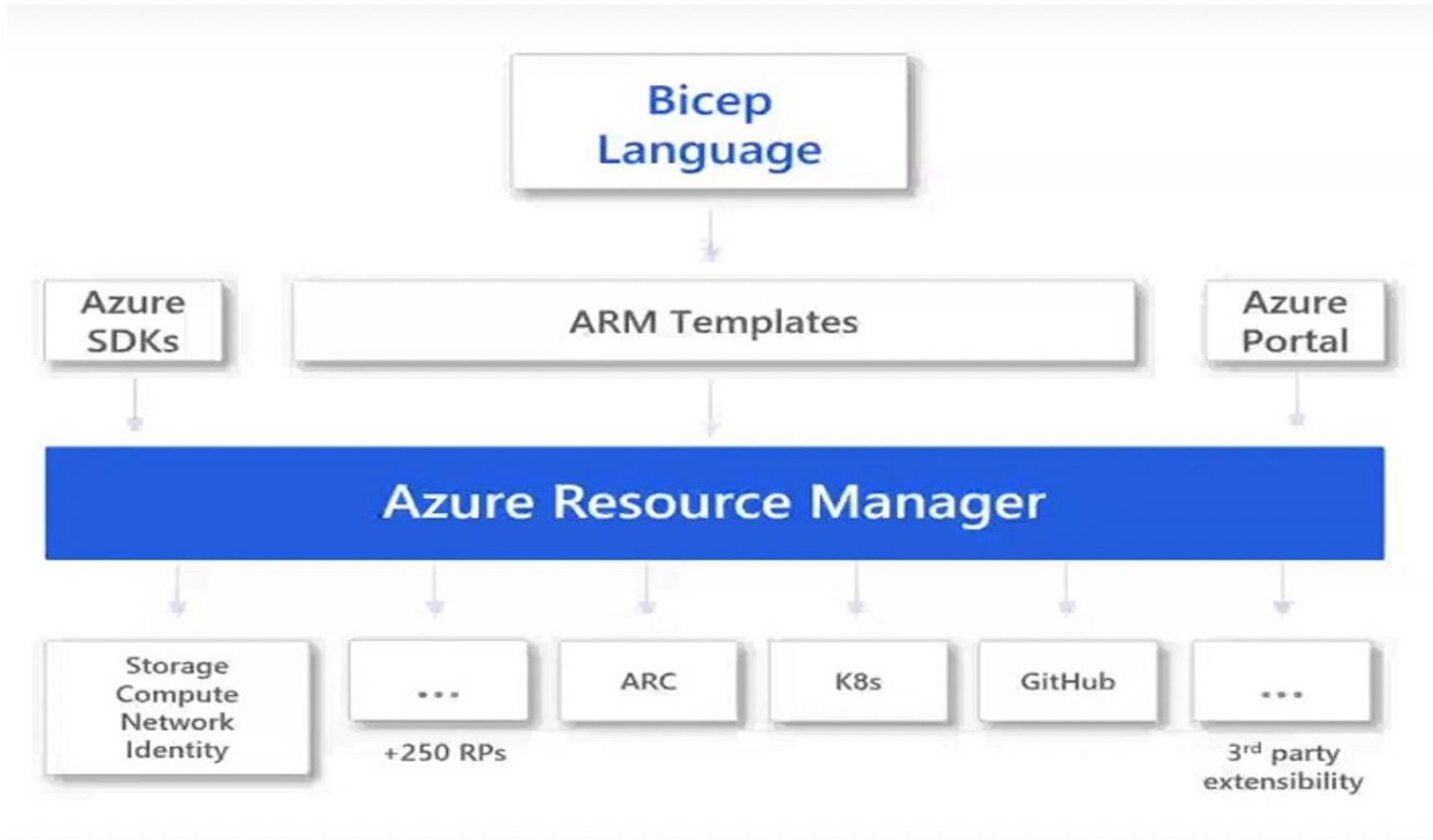
How to start with Bicep?

- [Install Azure PowerShell](#)

- [Install Azure CLI on Windows](#)

- [Install Azure CLI on Linux](#)

- [Install Azure CLI on macOS](#)

Bicep language concepts

Parameters

Variables

Modules

Loops

Scopes

Conditionals

T-REX

Quick look into Bicep concepts

Parameters

```
samples > ≡ parameterized-storage.bicep > ◈ storageSKU
  1    // parameterized-storage.bicep
  2    @minLength(3)
  3    @maxLength(24)
  4    param storageName string
  5
  6    @allowed([
  7      'Standard_LRS'
  8      'Standard_ZRS'
  9    ])
 10    param storageSKU string = 'Standard_LRS'
```

Modules

```
resource computeRg 'Microsoft.Resources/resourceGroups@2020-06-01' = {
  name: concat('bicep-azglobal-compute-v1-', azureRegion)
  location: azureRegion
}

var computeDeployment = 'vmLinuxDeploy'
var uniqueComputeDeployment = '${computeDeployment}${uniqueString(computeRg.id)}'

module vmWinMod './linux-vm.bicep' = {
  name: uniqueComputeDeployment
  scope: resourceGroup(computeRg.name)
  params: {
    adminUsername: 'azureuser'
    vmSize: 'Standard_B2s'
    vmName: 'myLinuxVm'
  }
}

resource databaseRg 'Microsoft.Resources/resourceGroups@2020-06-01' = {
  name: concat('bicep-azglobal-database-v1-', azureRegion)
  location: azureRegion
```

Template function and Variables

```
param azureRegion string = resourceGroup().location

var uniqueStorageName = '${storageName}${uniqueString(resourceGroup().id)}'
```

T-REX

Demo 1 – Getting started with Bicep on VS Code

Quick starter demo with storage account and parameterization

T-REX

Demo 2 – Parametrized Bicep modules (re-use)

Look into the multi-deployment scenarios

T-REX

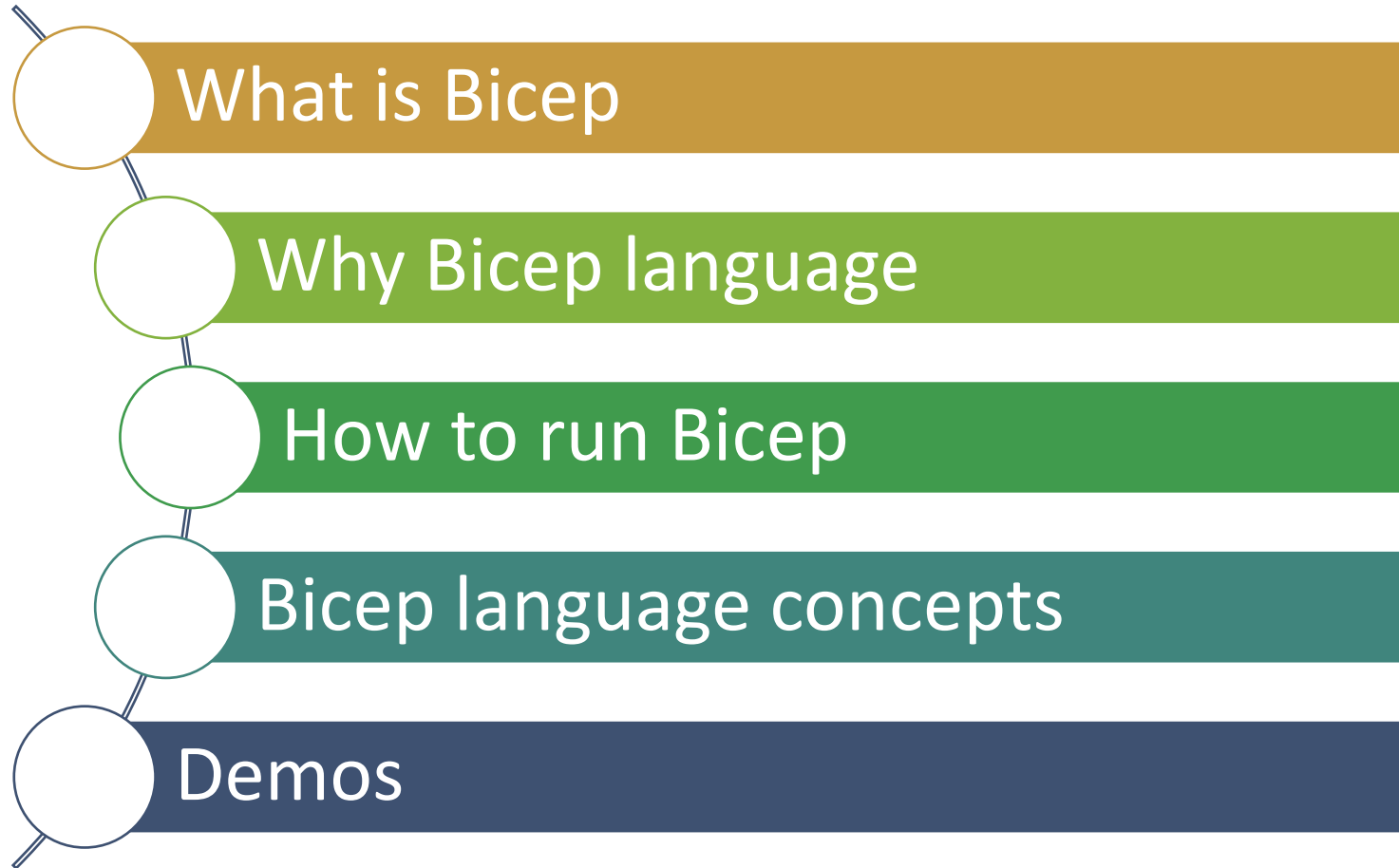# Demo 3 – Deployments (databases, security policies, etc.)

More modules and security policies for the deployment

What we have covered

What is Bicep

Why Bicep language

How to run Bicep

Bicep language concepts

Demos

T-REX

Q & A

T-REX

Elkhan Yusubov
Principal Cloud Architect
T-Rex Solutions

@ElYusubov

ElkhanYusubov

ElYusubov

Transformation journey from ARM to Azure Bicep

All you need to know about Azure Bicep and service provisioning journey