mibuso.com

Leveraging the power of the cloud

Bardur Knudsen, Gert Robyns, Vincent Nicolas, Esben Nyhuus Kristoffersen
MICROSOFT DEVELOPMENT CENTER COPENHAGEN

When you are passionate about
Microsoft Dynamics NAV/365 Business Central

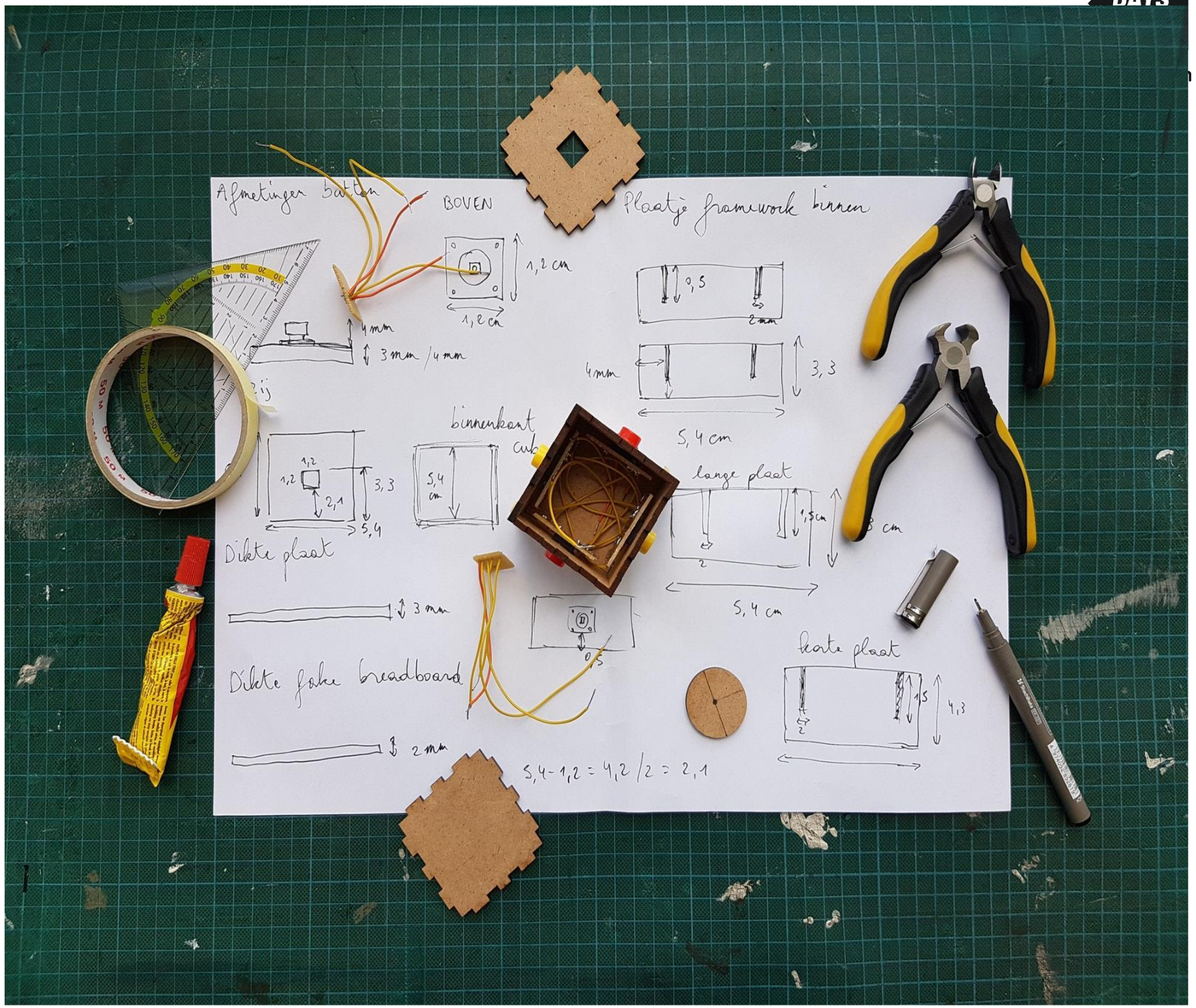
NAV TECH DAYS 2019
mibuso.com

Leveraging the power of the cloud

Bardur Knudsen
Gert Robyns
Esben Nyhuus Kristoffersen
Vincent Nicolas


TDK
IEC I/TYPE I NORMAL POSITION
AD90
SONY
HF-S90
NORMAL BIAS 120μs EQ
Crystal Gamma
90
TYPE I (NORMAL)

Prototype vs Production



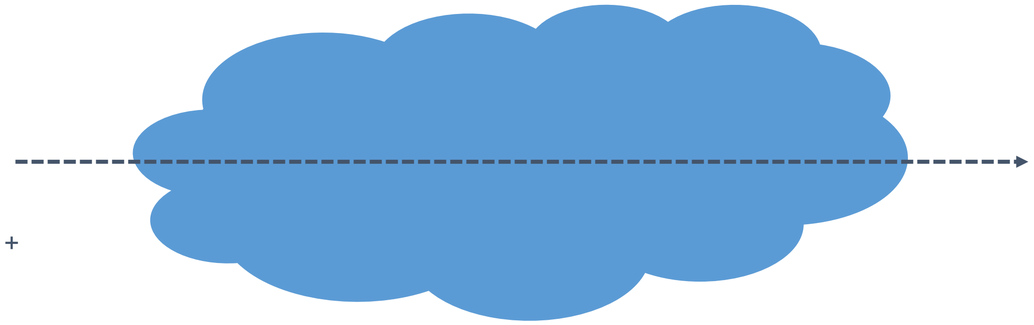
Shared Key Access

Shared Access Key Principles

NAV TECH DAYS 2019
mibuso.com

Client

Service REST API



Sign the request

Request Signature = Hash(Key Request Attributes + Timestamp, Shared Key)

Verify the request

Ok := Request Signature = Hash(Key Request Attributes + Timestamp, Shared Key)

Azure Service Bus + Shared Access Signatures

SharedAccessSignature sig=<signature-string>&se=<expiry>&skn=<keyName>&sr=<URL-encoded-resourceURI>

Key	Value
se	Token expiry instant. Integer reflecting seconds since the epoch 00:00:00 UTC on 1 January 1970 (UNIX epoch) when the token expires.
skn	Name of the authorization rule.
sr	URI of the resource being accessed.
sig	Signature. SHA-256 hash computed using the key for the selected policy over the resource URI and the string representation of the token expiry instant, separated by LF. E.g. SHA-256('https://<yournamespace>.servicebus.windows.net/'+'\n'+ 1438205742)

Files & Storage

NAV TECH DAYS 2019
mibuso.com

DEMO

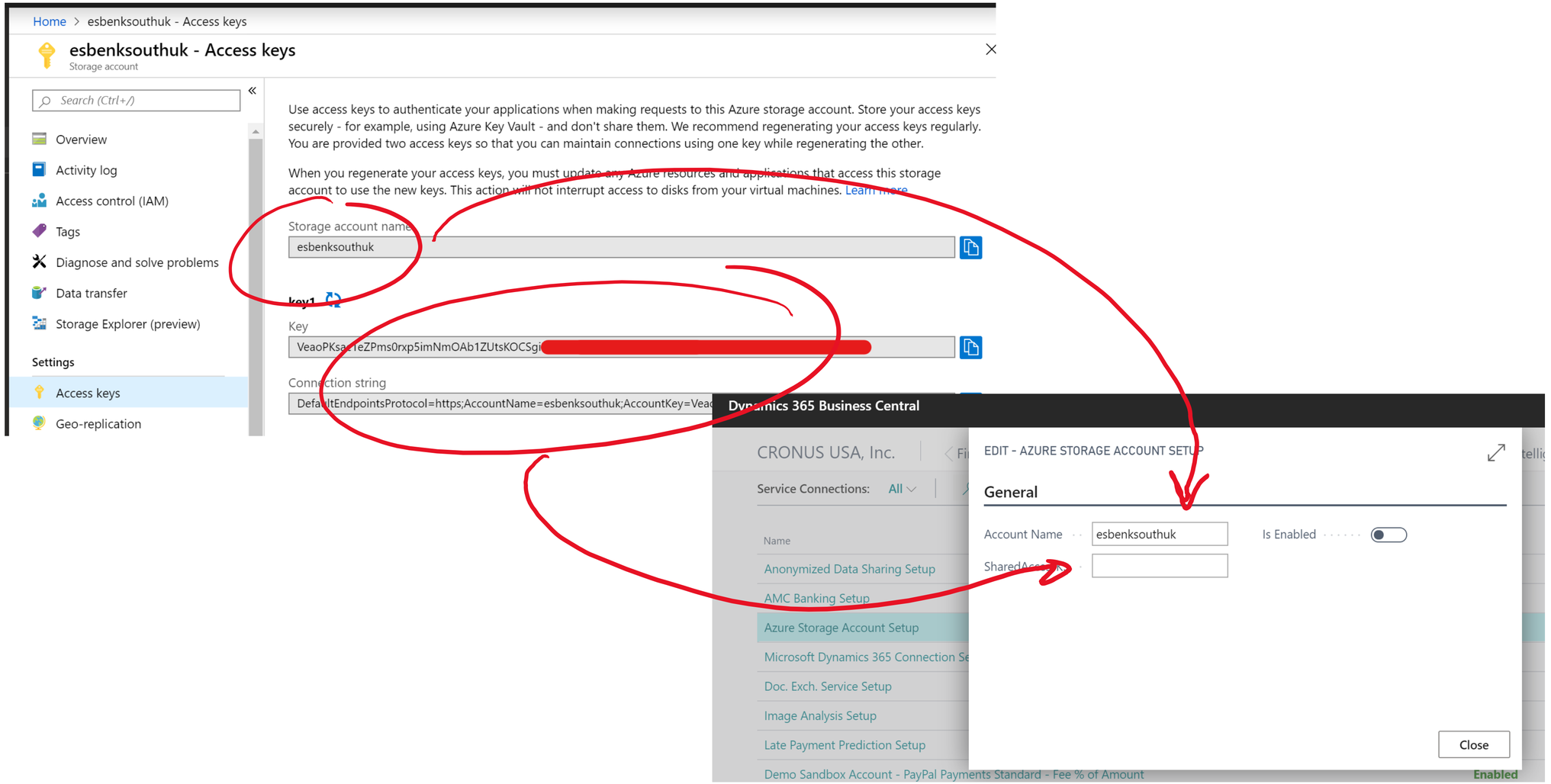
Azure BLOB Storage

Supported BLOB Service REST API's

Operation	Ressource Type	Description
List Containers	Account	Lists all of the containers in a storage account.
Create Container	Container	Creates a new container in a storage account.
Delete Container	Container	Deletes the container and any blobs that it contains.
List Blobs	Container	Lists all of the blobs in a container.
Put Blob	Block, append, and page blobs	Creates a new blob or replaces an existing blob within a container.
Get Blob	Block, append, and page blobs	Reads or downloads a blob from the Blob service, including its user-defined metadata and system properties.
Copy Blob	Block, append and page blobs	The Copy Blob operation copies a blob to a destination within the storage account.
Delete Blob	Block, append and page blobs	Marks a blob for deletion.
Delete Blobs	Block, append and page blobs	Marks blobs for deletion based on a prefix.

*Thanks to Bart van Beek

Setting Up the BLOB Storage Service



Example: How to upload a file to BLOB Storage

NAV TECH DAYS 2019
mibuso.com

```
procedure UploadFile()
var
    AzureBlobStorage: codeunit AzureBlobStorage;
    ins: InStream;
    name: Text;
begin
    UploadIntoStream('', '', '', name, ins);
    AzureBlobStorage.PutBlob('MyContainer/' + name,
        ins,
        AzureBlobStorage.GetContentTypeFromFileName(name));
end;
```

External functionality

NAV TECH DAYS 2019
mibuso.com

DEMO

Azure Functions

MakeMonochrome Azure Function

NAV TECH DAYS 2019
mibuso.com

```
[FunctionName("MakeMonochrome")]
public static async Task<IActionResult> Run(
    [HttpTrigger(AuthorizationLevel.Function, "get", "post", Route = null)] HttpRequest req,
    ILogger log)
{
    try
    {
        MemoryStream memoryStream = new MemoryStream();

        using (Image<Rgba32> image =
            Image.Load(Convert.FromBase64String(new StreamReader(req.Body).ReadToEnd())))
        {
            image.Mutate(x => x.Grayscale());
            image.SaveAsJpeg(memoryStream);
        }
        return (ActionResult)new OkObjectResult(Convert.ToBase64String(memoryStream.GetBuffer()));
    }
    catch (System.Exception e)
    {
        return new BadRequestObjectResult(e.Message);
    }
}
```

Image Conversion

Calling MakeMonochrome Azure Function


```
local procedure MakeMonochrome(TempBlob: codeunit "Temp Blob");
var
    Base64Convert: codeunit "Base64 Convert";
    ImageAsString: Text;
    Content: HttpContent;
    Client: HttpClient;
    Response: HttpResponseMessage;
    Ins: InStream;
    Outs: OutStream;
    len: Integer;
begin
    len := TempBlob.Length();
    TempBlob.CreateInStream(Ins);
    ImageAsString := Base64Convert.ToBase64(Ins);

    Content.WriteFrom(ImageAsString);
    Client.Post(
        'https://imagehelper2019.azurewebsites.net/api/MakeMonochrome?code=aMmhBXEgbYsTFI0Br9vqzpjxdfhlkdaG/BL8V81JbY2qikoBhng==',
        Content,
        Response);
    if not Response.IsSuccessStatusCode() then
        Error('MakeMonochrome: ' + Response.ReasonPhrase());
    Response.Content().ReadAs(ImageAsString);

    TempBlob.CreateOutStream(Outs);
    Base64Convert.FromBase64(ImageAsString, Outs);
end;
```

Prepare Arguments

Azure Function Call

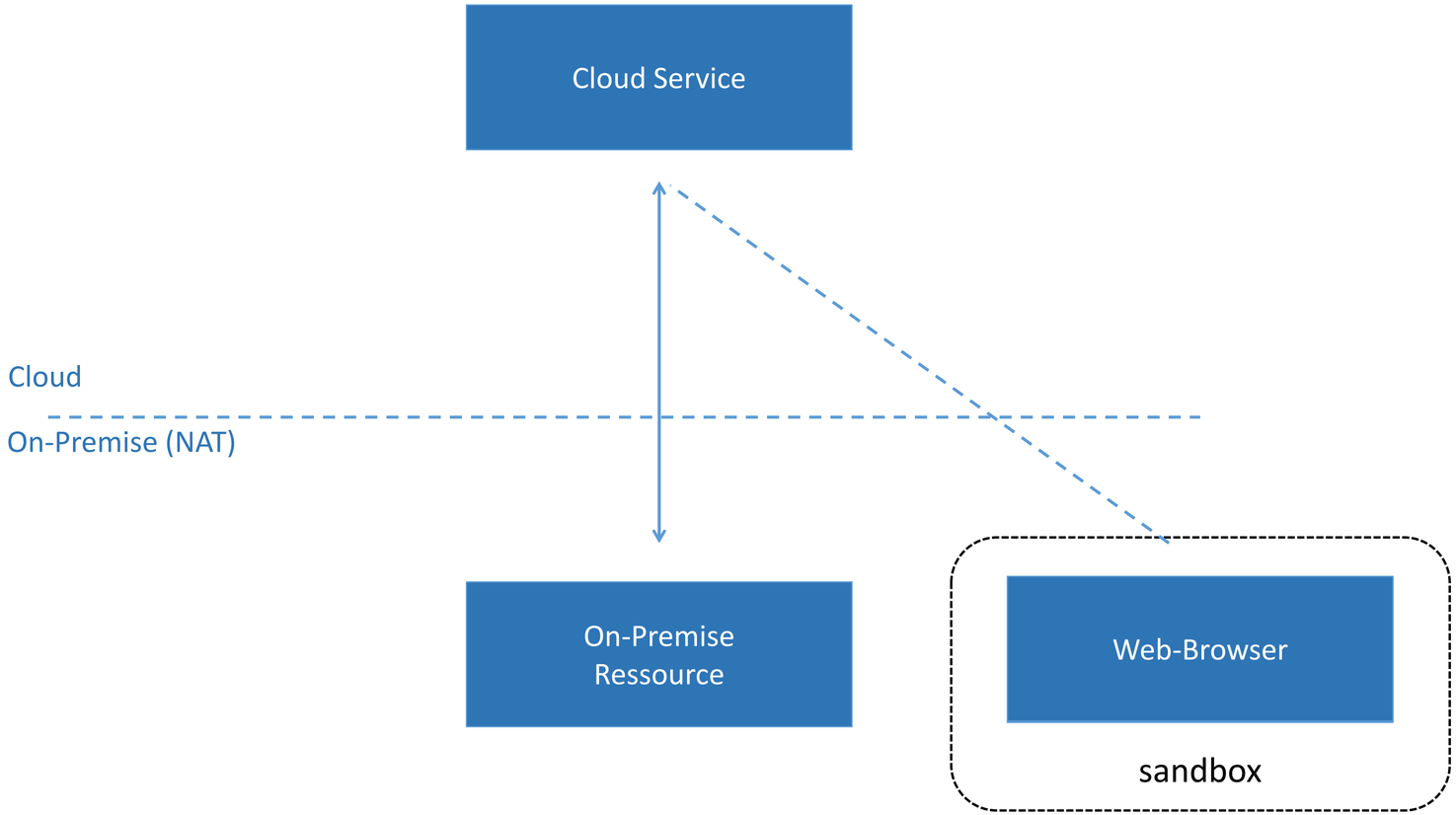
Return Result

Examples of integration to the “real” world

Barcode scanners, Payment terminals

Scales

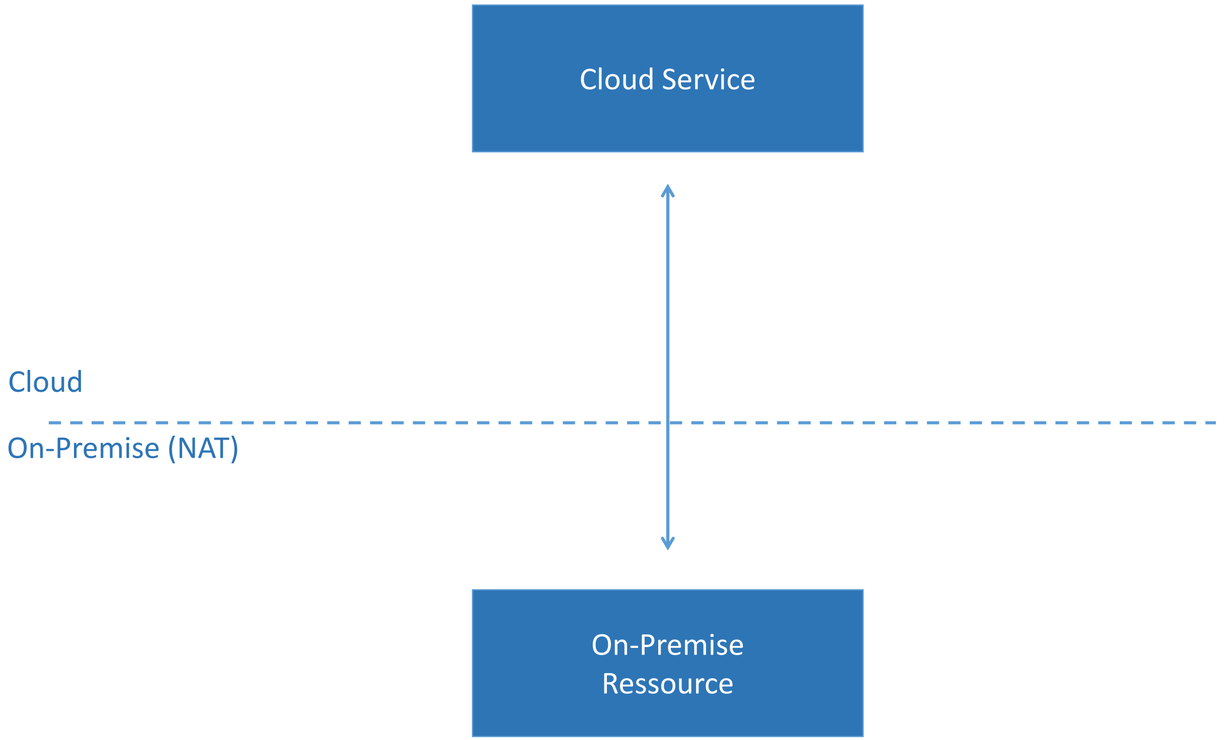
Production equipment


Cloud Service
Cloud
On-Premise (NAT)
On-Premise Ressource
Web-Browser
sandbox

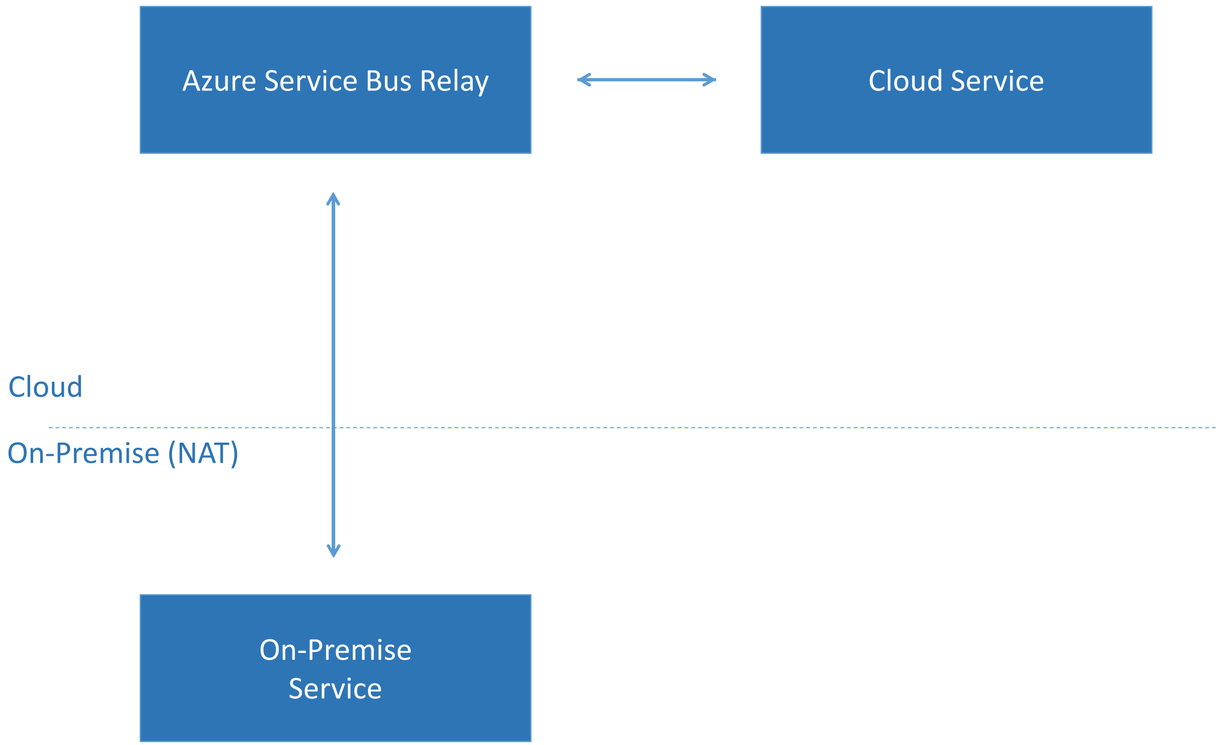
DEMO

Local File Browsing

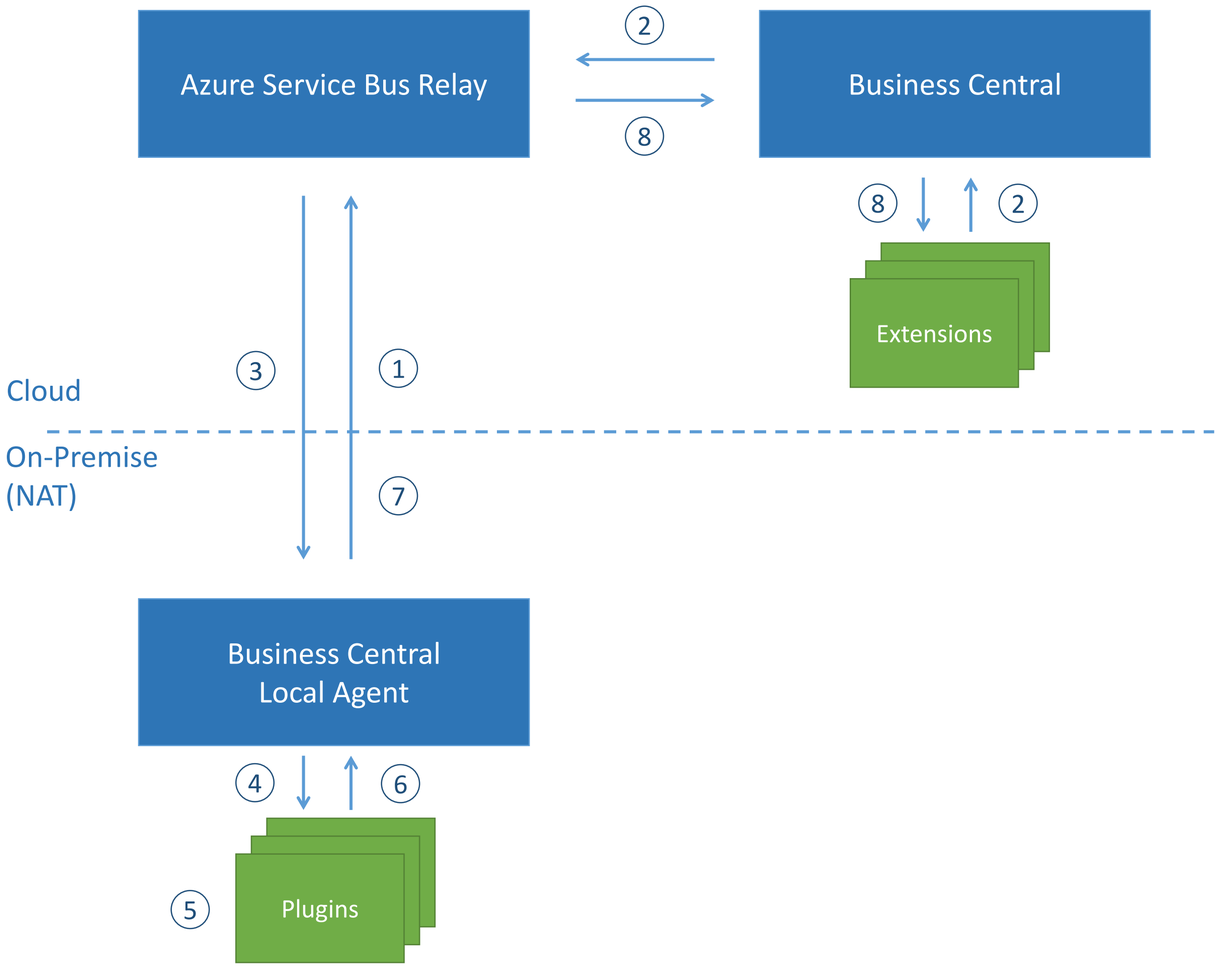
“The Problem”



Azure Service Bus Relay



BC Local Agent



1. The Agent establish a listener connection to the Relay endpoint.
2. Extension X uses HttpClient to send a request to Plugin X
3. Azure Relay forwards request to the local Agent
4. The local Agent forwards the request to the right plugin based on the relative Uri
5. Plugin X is processing the request with access to local resources.
6. Plugin X is returning the result to the local Agent
7. The Agent returns the result in the Response content
8. Azure Relay returns the result to the extensions

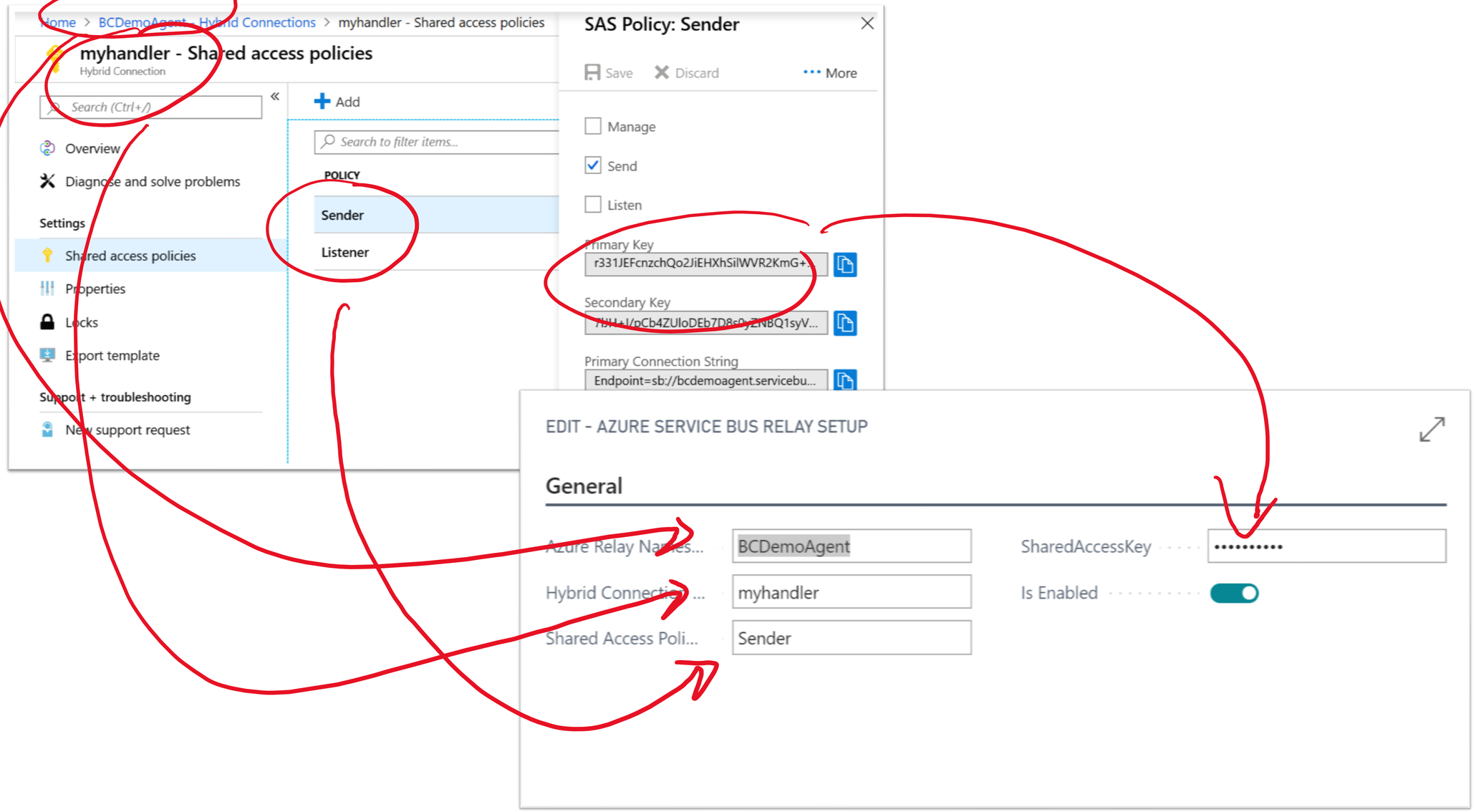
Plugins + Extensions

NAV TECH DAYS 2019
mibuso.com

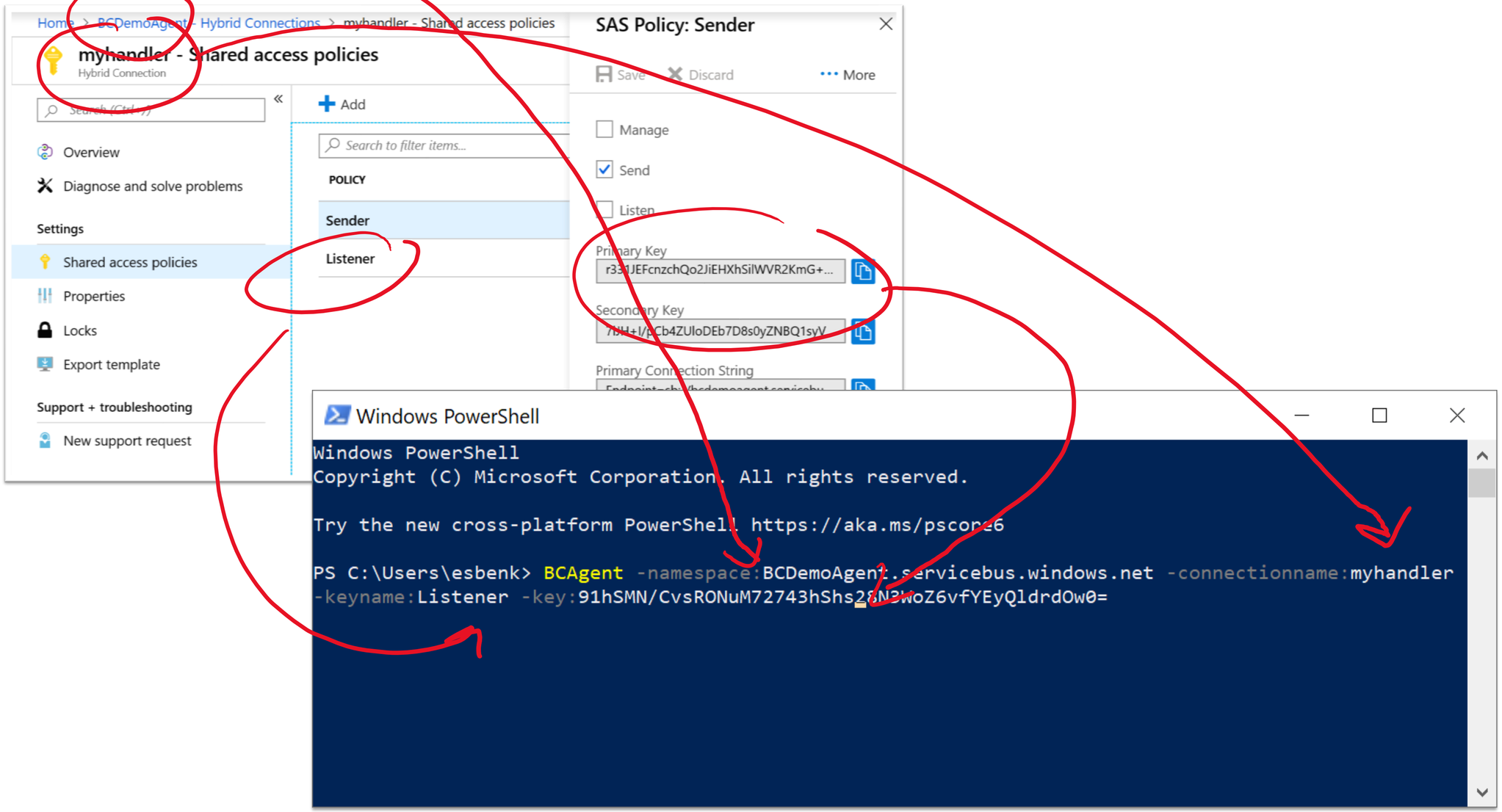
```
namespace CalculatorPlugin
{
    using Microsoft.Dynamics.BusinessCentral.Agent.Common;

    [AgentPlugin("calculator/V1.0")]
    public class Calculator : IAgentPlugin
    {
        [PluginMethod("GET")]
        public decimal Add(decimal a, decimal b)
        {
            return a + b;
        }

        [PluginMethod("GET")]
        public decimal Subtract(decimal a, decimal b)
        {
            return a - b;
        }
    }
}
```

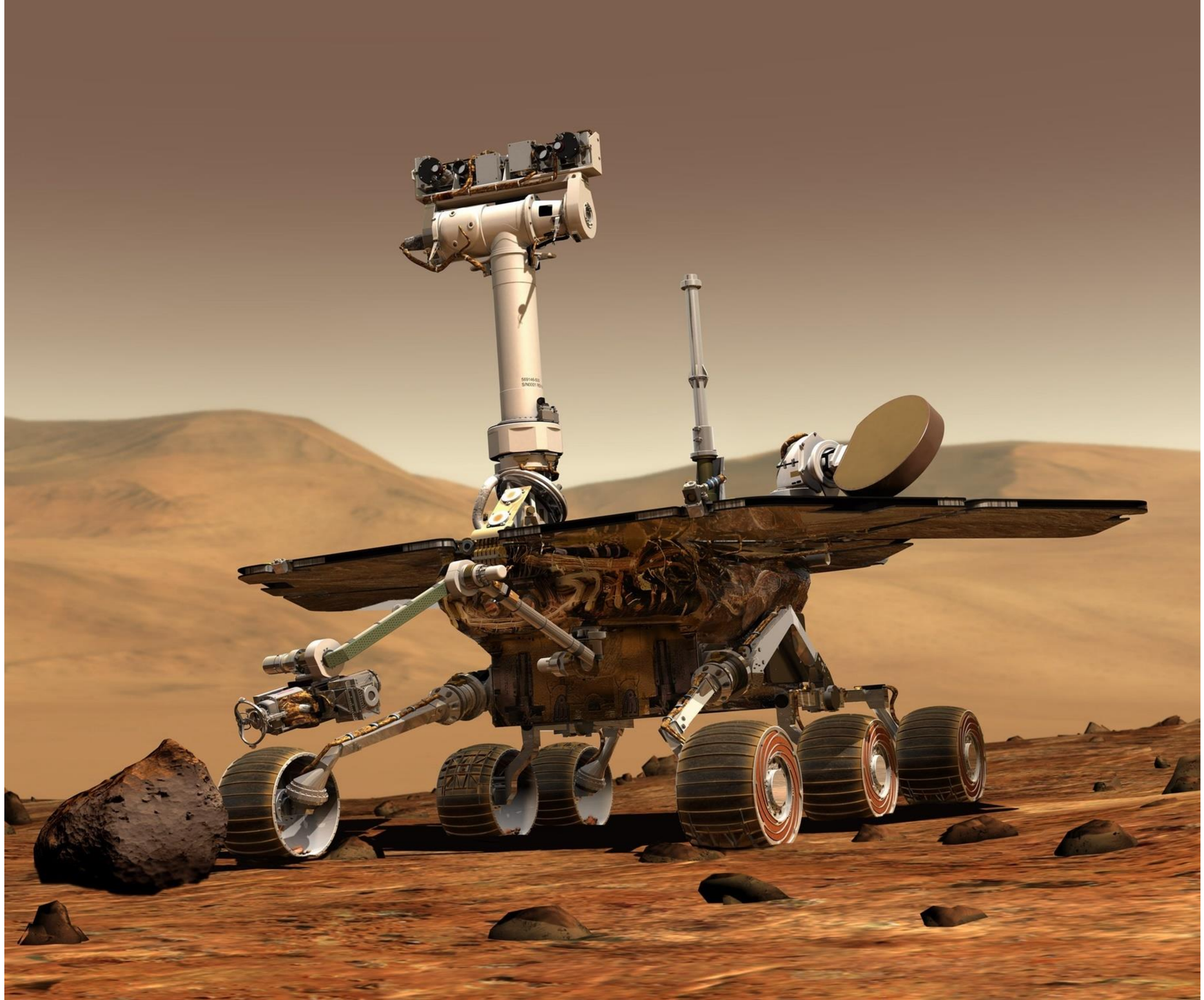
```
codeunit 50130 Calculator
{
  var
    ServiceBusRelay: codeunit ServiceBusRelay;
    CalculatorPluginName: Label '/calculator/V1.0', Locked = true;
    AddFuncDef: Label '/Add?a=%1&b=%2', Locked = true;
    SubtractFuncDef: Label '/Subtract?a=%1&b=%2', Locked = true;

  procedure Initialize(ServiceRelayUri: Text; keyName: Text; keyValue: Text;
    ttl: Integer);
  begin
      ServiceBusRelay.Initialize(ServiceRelayUri + CalculatorPluginName,
        keyName, keyValue, ttl);
  end;

  procedure Add(a: Decimal; b: Decimal) Result: Decimal;
  var
    ResultText: Text;
  begin
     ServiceBusRelay.Get(StrSubstNo(AddFuncDef, Format(a, 0, 9), Format(b, 0, 9)),
       ResultText);
     Evaluate(Result, ResultText, 9);
  end;

  procedure Subtract(a: Decimal; b: Decimal) Result: Decimal;
  var
    ResultText: Text;
  begin
    ServiceBusRelay.Get(StrSubstNo(SubtractFuncDef, Format(a, 0, 9),
      Format(b, 0, 9)), ResultText);
    Evaluate(Result, ResultText, 9);
  end;
}
```

Configuration – Business Central



Configuration - BCAgent





NAV TECH DAYS 2019

mibuso.com

LEGO Mindstorms

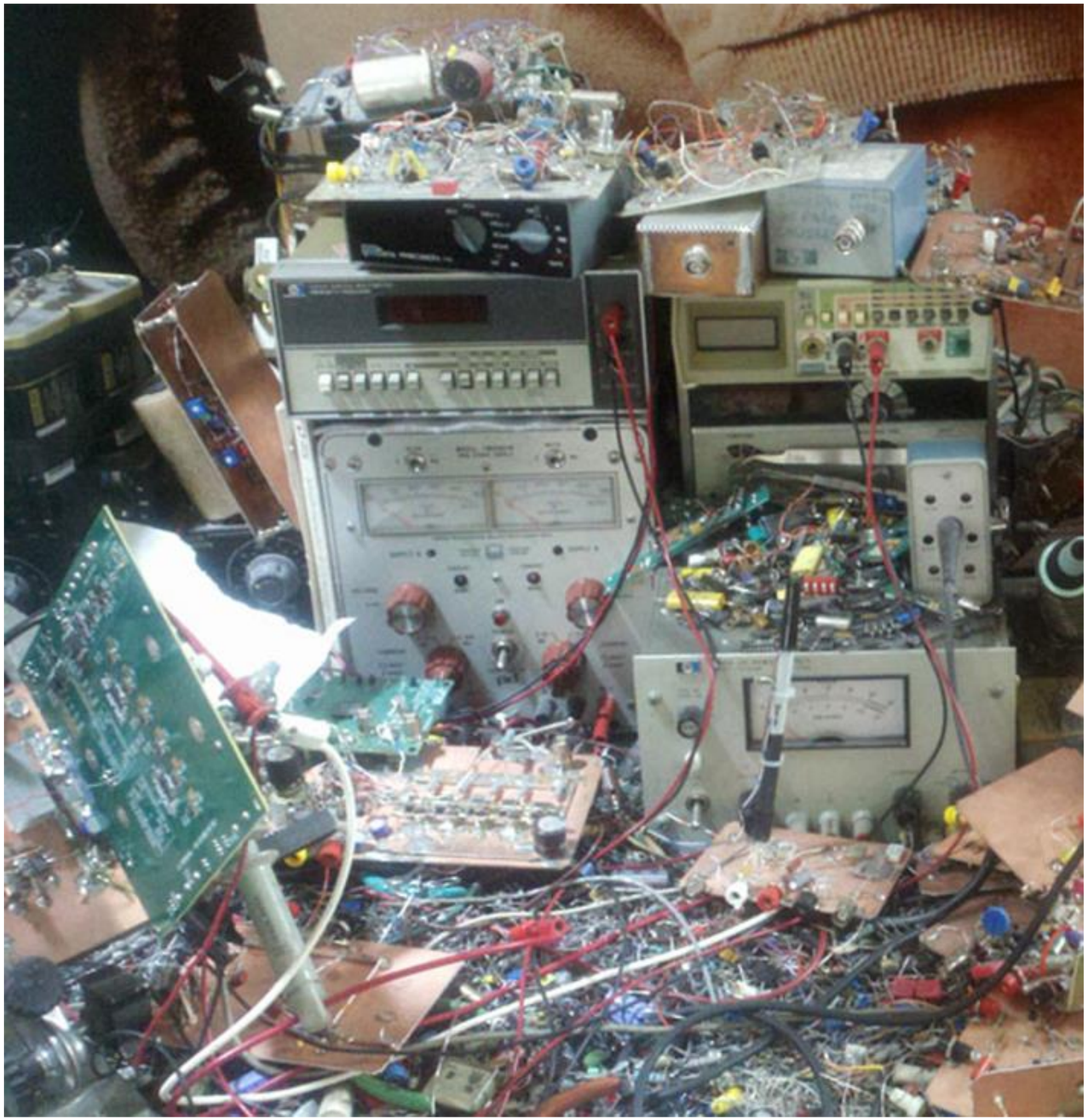
NXT 2.0
Released in 2009
Has been hiding in my basement

NAV TECH DAYS 2019
mibuso.com

DEMO

LEGO Robot Demo

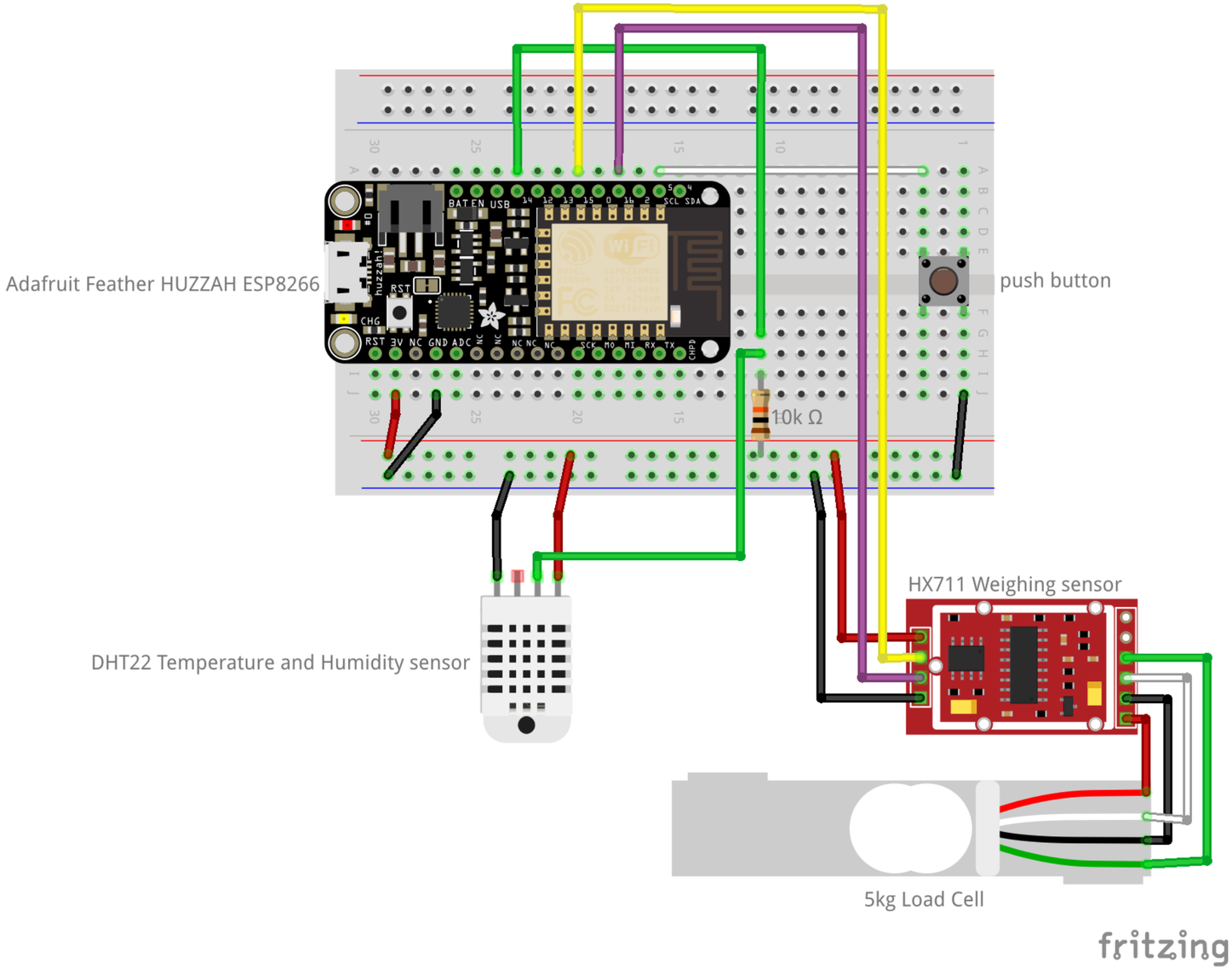
Internet of Things



NAV TECH DAYS 2019
mibuso.com

NAV TECH DAYS 2019
mibuso.com

Cookie jar wiring diagram:

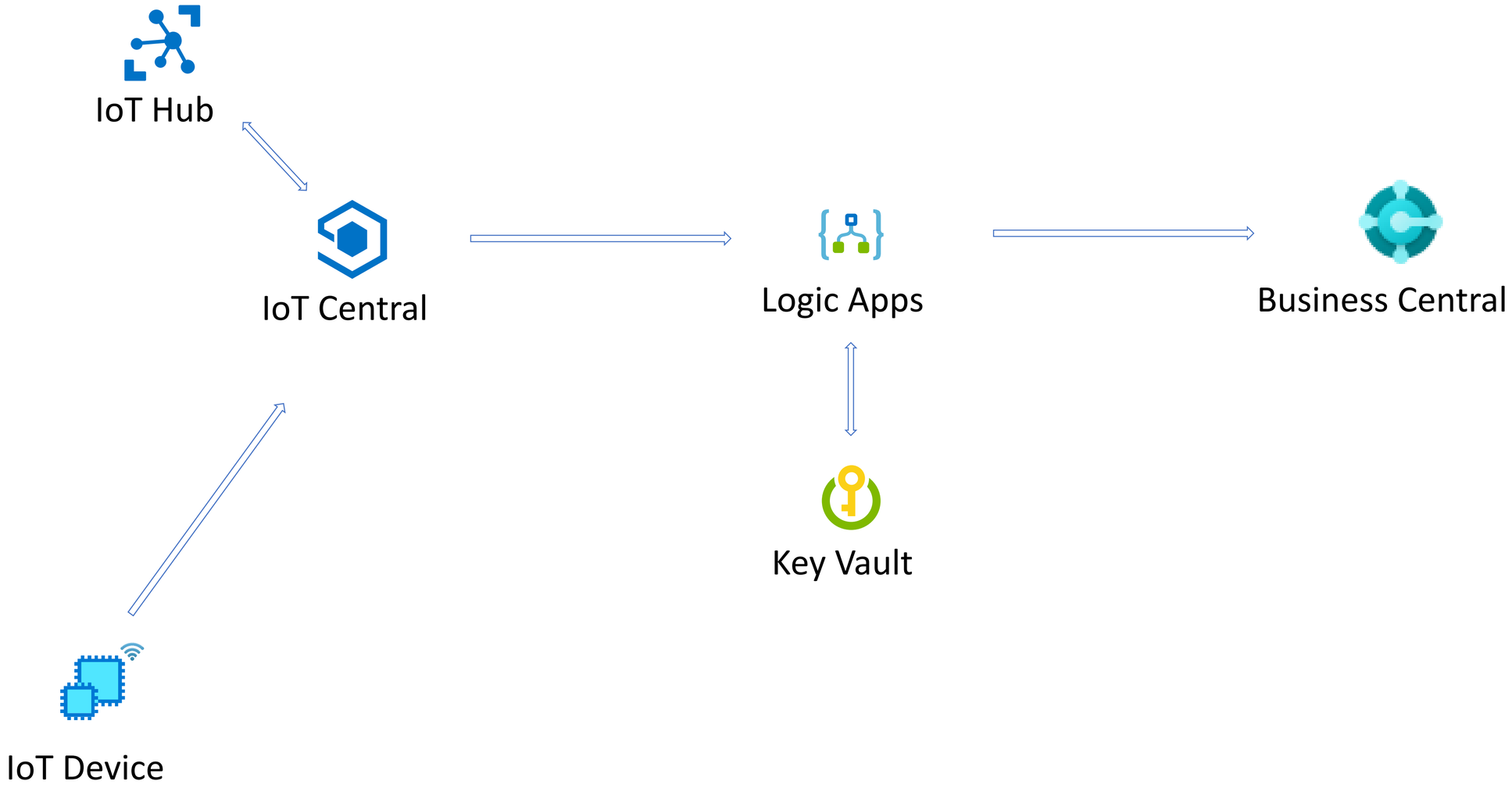


NAV TECH DAYS 2019
mibuso.com

DEMO

IoT Central to Business Central

IoT device to Business Central

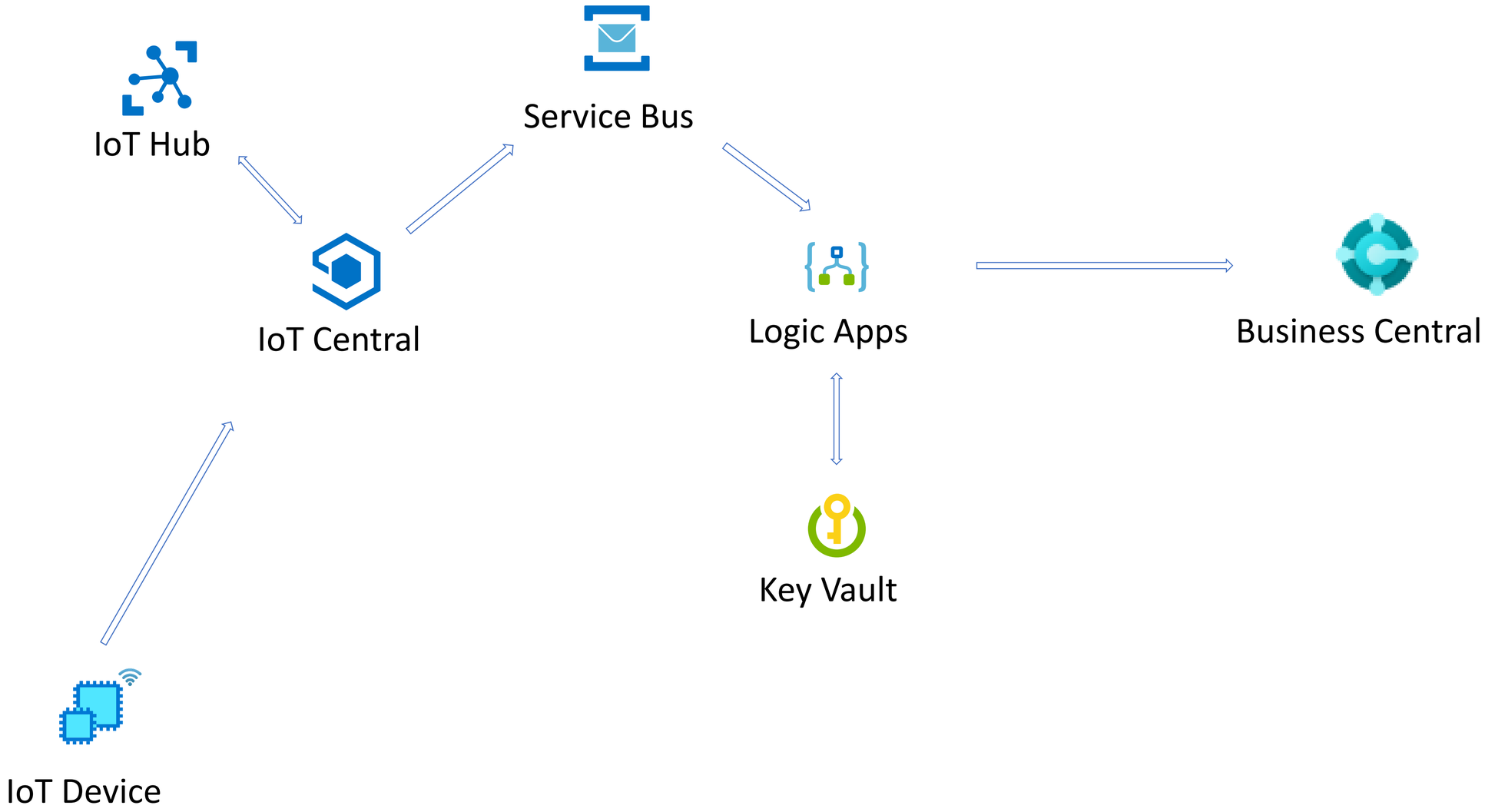


NAV TECH DAYS 2019
mibuso.com

DEMO

IoT Central continuous export

IoT device info to Business Central

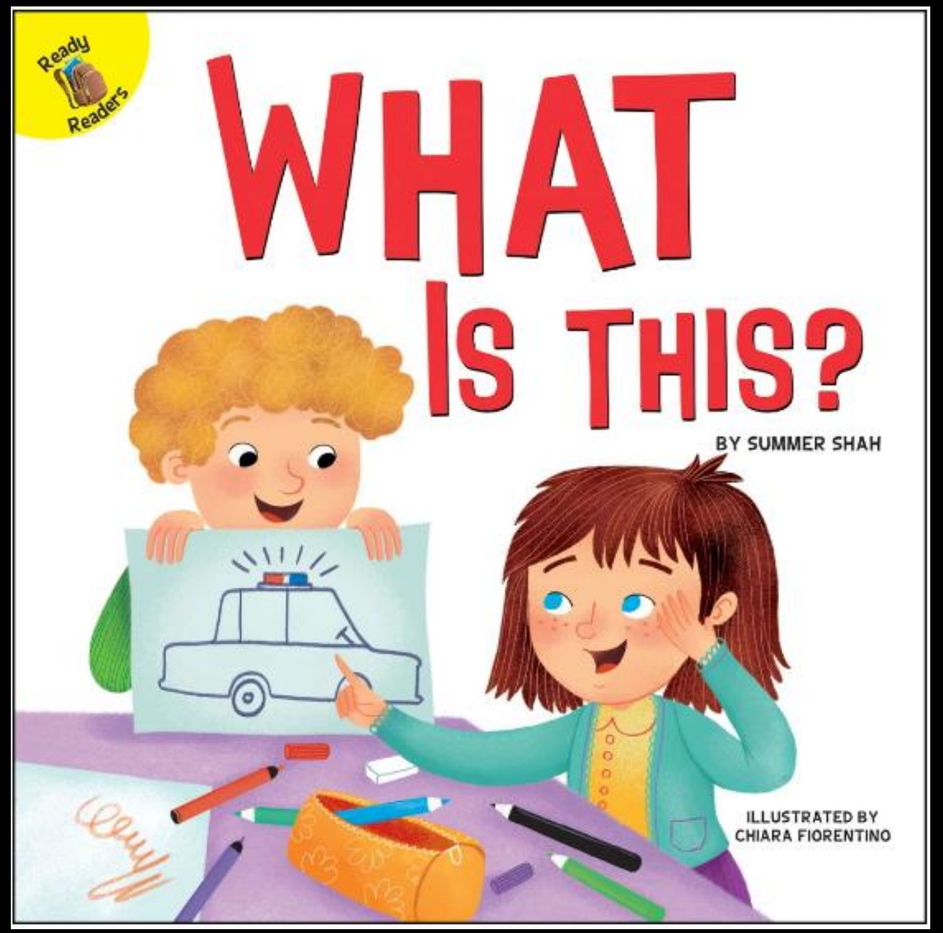


Other IoT & ERP scenarios

NAV TECH DAYS 2019
mibuso.com

- Time registration:
  automate using smart devices
- Shipping:
  track your shipment in real-time
- Inventory Mgmt:
  accurate real-time inventory, track consumption, scrap etc…
- WMS:
  automate pick lists using RFID tags / scanners
- Service Mgmt:
  use device telemetry on equipment to investigate outages or even detect issues before a customer is affected

Azure Cognitive Services – Vision / OCR



Azure AI + Machine Learning

NAV TECH DAYS 2019
mibuso.com

AI + Machine Learning

Create the next generation of applications using artificial intelligence capabilities for any developer and any scenario

Cognitive Services

Add smart API capabilities to enable contextual interactions

Azure Bot Service

Intelligent, serverless bot service that scales on demand

Machine Learning

Build, train, and deploy models from the cloud to the edge

Azure Databricks

Fast, easy, and collaborative Apache Spark-based analytics platform

Azure Cognitive Search

AI-powered cloud search service for mobile and web app development

Cognitive Services

NAV TECH DAYS 2019

Vision

Recognize, identify, caption, index, and moderate your pictures, videos, and digital ink content.

- Computer Vision
- Custom Vision
- Face
- Form Recognizer (Preview)
- Ink Recognizer (Preview)
- Video Indexer

Language

Allow your apps to process natural language with pre-built scripts, evaluate sentiment and learn how to recognize what users want.

- Immersive Reader (Preview)
- Language Understanding (LUIS)
- QnA Maker
- Text Analytics
- Translator Text

Speech

Convert speech into text and text into natural-sounding speech. Translate from one language to another and enable speaker verification and recognition.

- Speech Services
- Speaker Recognition (Preview)
- Bing Speech (Retiring)

Decision

Build apps that surface recommendations for informed and efficient decision-making.

- Anomaly Detector (Preview)
- Content Moderator
- Personalizer

Search

Add Bing Search APIs to your apps and harness the ability to comb billions of webpages, images, videos, and news with a single API call.

- Autosuggest Search
- Custom Search
- Entity Search
- Image Search
- Local Business Search (Preview)
- News Search
- Spell Check
- Video Search
- Visual Search
- Web Search

Cognitive Service Containers

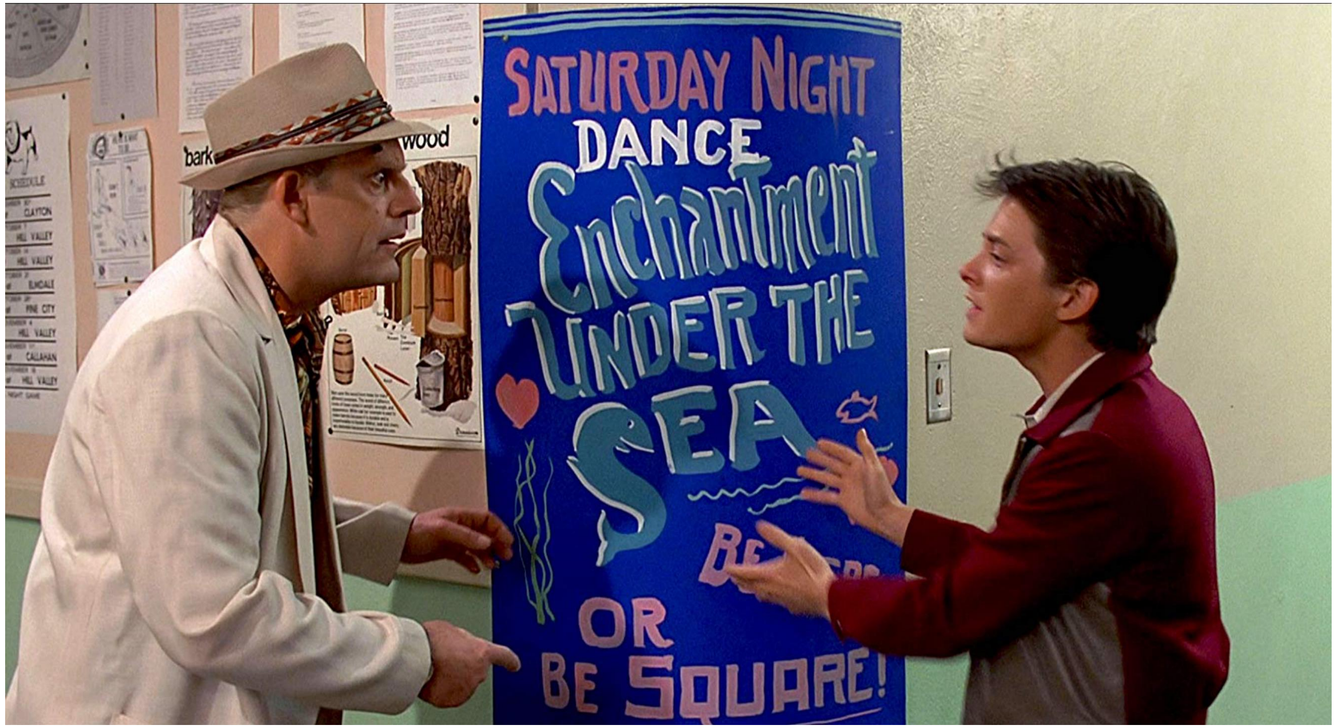
Container support in Azure Cognitive Services allows developers to use the same rich APIs that are available in Azure, and enables flexibility in where to deploy and host the services that come with Docker containers.

- Cognitive Service Containers
- Create containers for reuse
- Deploy and run container on Azure Container Instance
- Deploy to Azure Kubernetes Service
- Use Docker Compose to deploy multiple containers
- Cognitive Services containers FAQ

https://docs.microsoft.com/en-us/azure/cognitive-services/

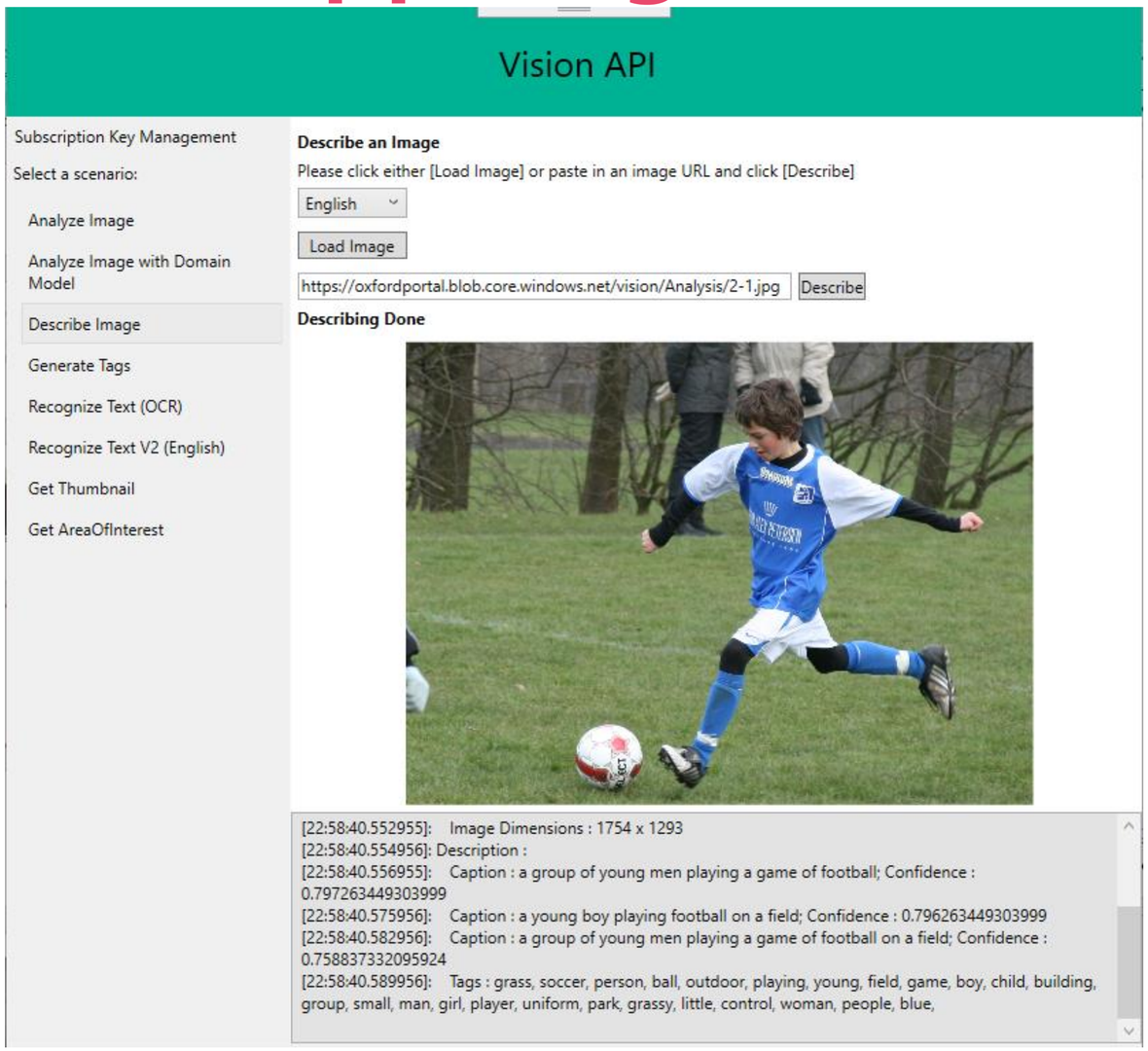
Computer Vision

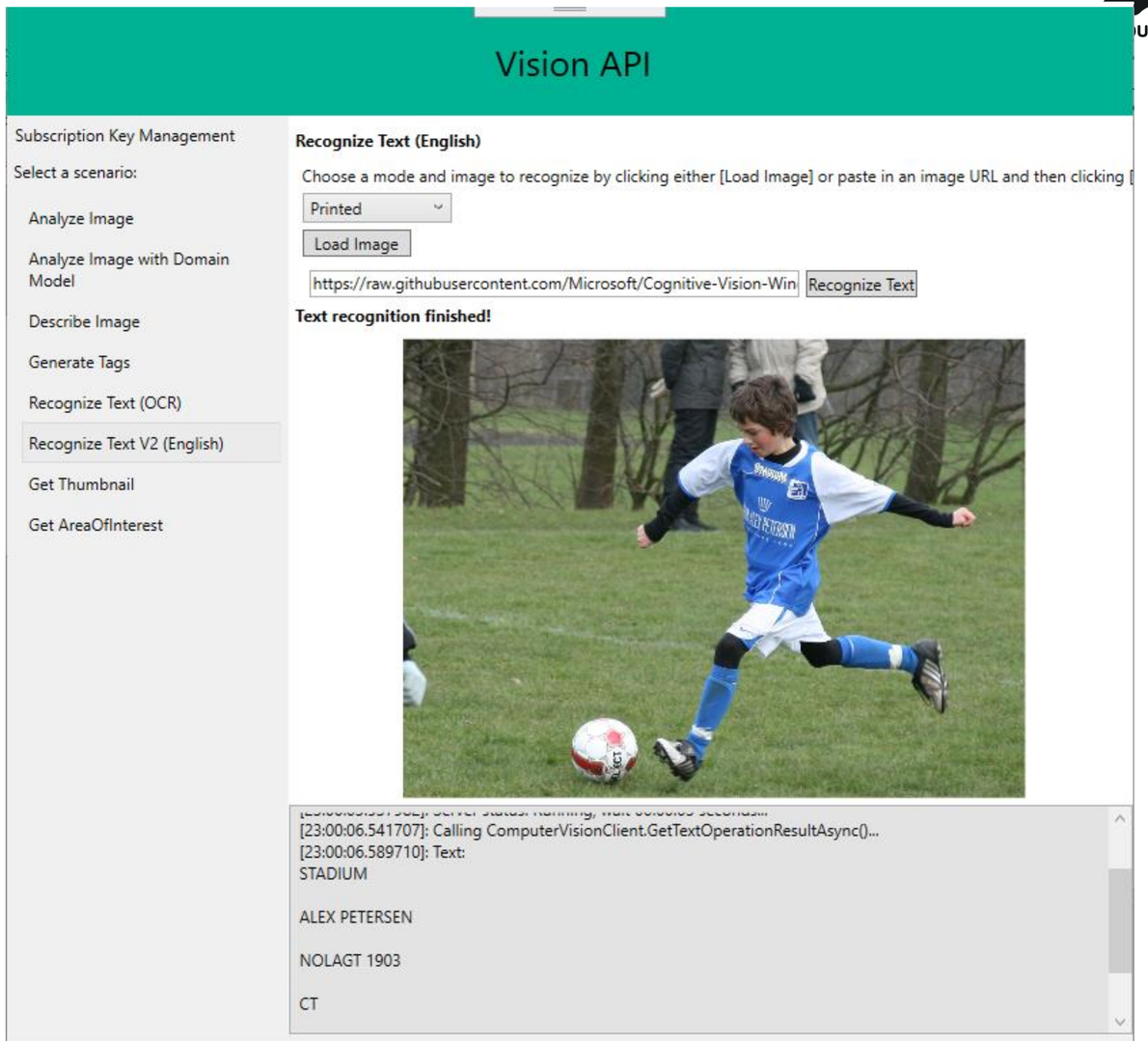
- Analyze Images
- Moderate content
- Extract text
  - OCR of printed text
  - Recognize handwritten and printed text in a picture



6 wood SATURDAY NIGHT bark DANCE Enchantment UNDER THE SEA RE OR BE SQUARE!

C# app on github





https://docs.microsoft.com/en-us/azure/cognitive-services/Computer-vision/tutorials/csharptutorial

git clone --recurse-submodules https://github.com/Microsoft/Cognitive-Face-Windows.git

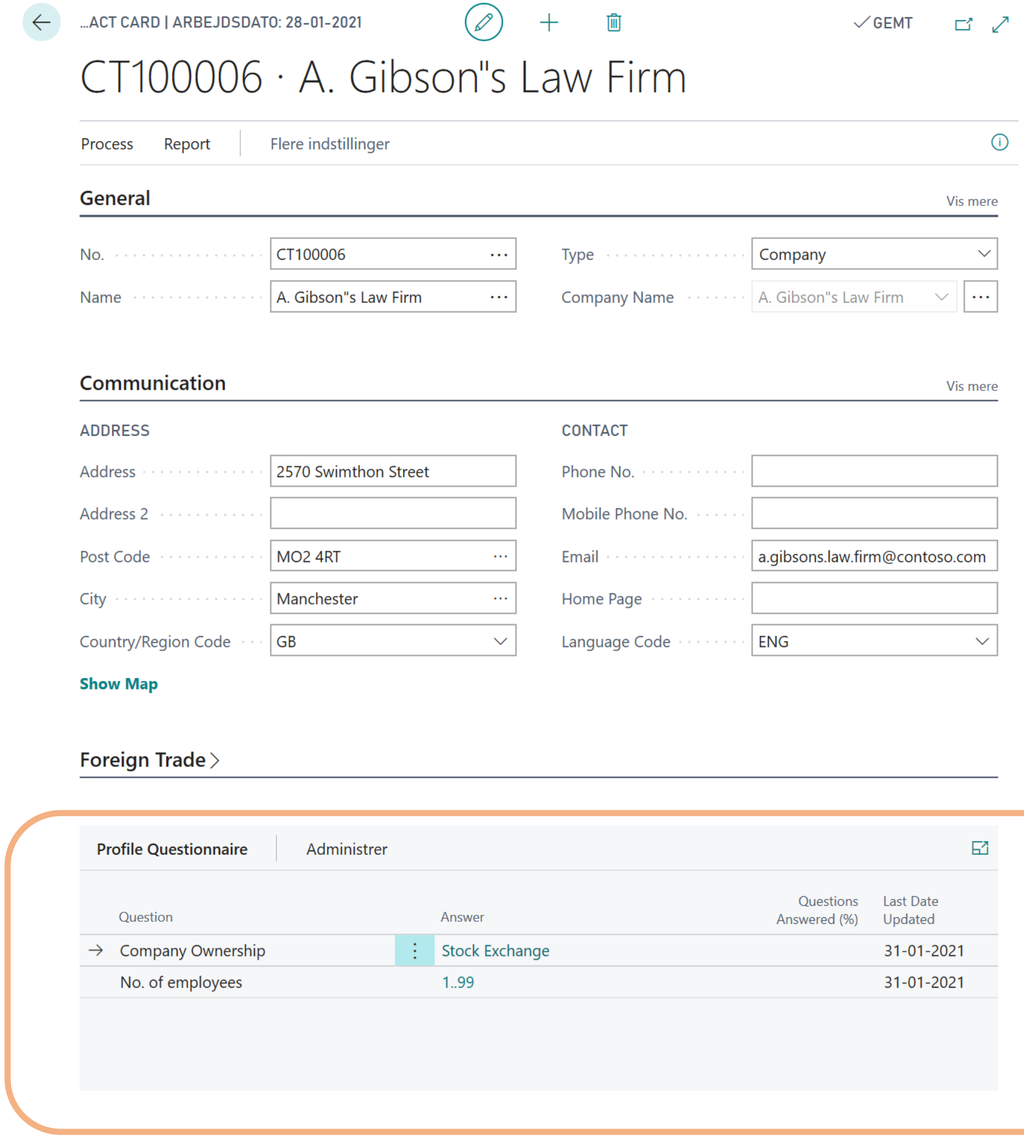
Microsoft Dynamics 365

NAV TECH DAYS 2019

mibuso.com

User Story

Contacts – Profile Questionnaire



Questionnaire - Handouts

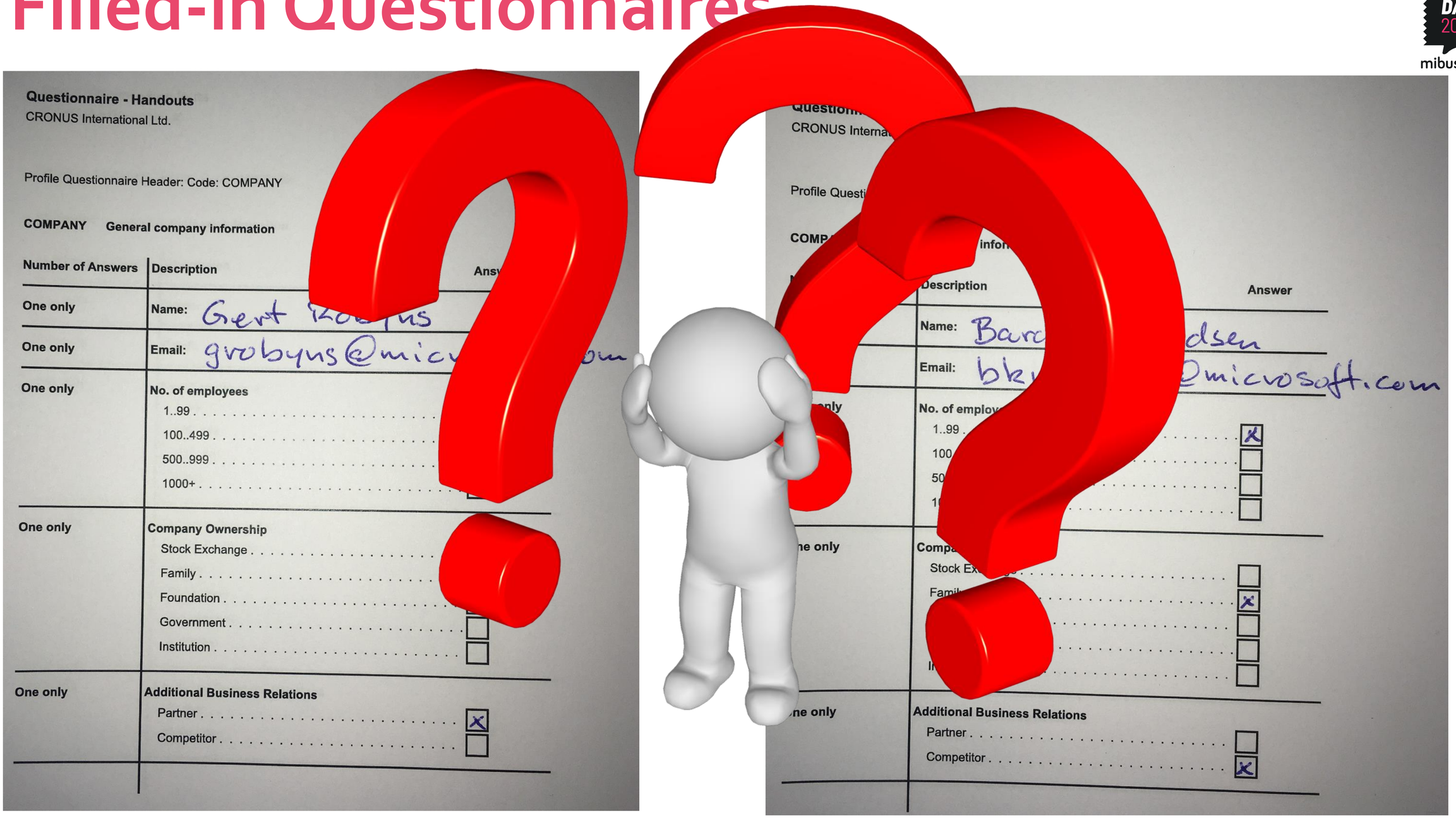
CRONUS International Ltd.

Profile Questionnaire Header: Code: COMPANY

COMPANY **General company information**

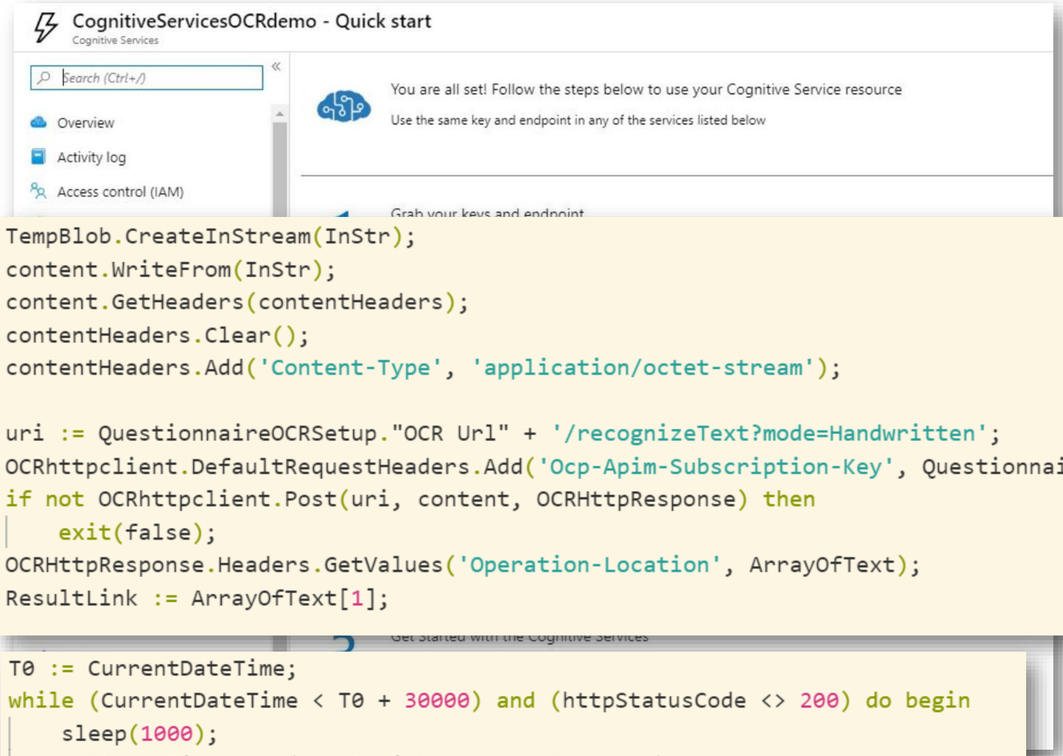
Number of Answers	Description	Answer
One only	**Name:**	
One only	**Email:**	
One only	**No. of employees** 1..99 ☐ 100..499 ☐ 500..999 ☐ 1000+ ☐	
One only	**Company Ownership** Stock Exchange ☐ Family ☐ Foundation ☐ Government ☐ Institution ☐	
One only	**Additional Business Relations** Partner ☐ Competitor ☐	

Filled-in Questionnaires



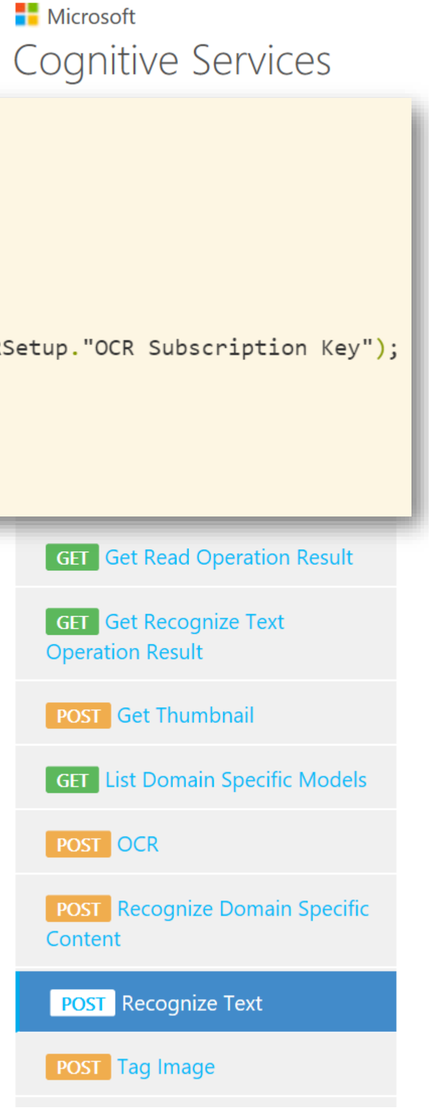
Recipe

1. Obtain a subscription
2. POST a request
3. GET the result
4. Parse the json result




```
TempBlob.CreateInStream(InStr);
content.WriteFrom(InStr);
content.GetHeaders(contentHeaders);
contentHeaders.Clear();
contentHeaders.Add('Content-Type', 'application/octet-stream');

uri := QuestionnaireOCRSetup."OCR Url" + '/recognizeText?mode=Handwritten';
OCRhttpclient.DefaultRequestHeaders.Add('Ocp-Apim-Subscription-Key', QuestionnaireOCRSetup."OCR Subscription Key");
if not OCRhttpclient.Post(uri, content, OCRHttpResponse) then
    exit(false);
OCRHttpResponse.Headers.GetValues('Operation-Location', ArrayOfText);
ResultLink := ArrayOfText[1];
```

```
T0 := CurrentDateTime;
while (CurrentDateTime < T0 + 30000) and (httpStatusCode <> 200) do begin
    sleep(1000);
    OCRhttpclient.get(ResultLink, OCRHttpResponse);
    httpStatusCode := OCRHttpResponse.HttpStatusCode;
end;
if httpStatusCode <> 200 then
    exit(false);
OCRHttpResponse.Content.ReadAs(t);
jsonresult.ReadFrom(t);
```



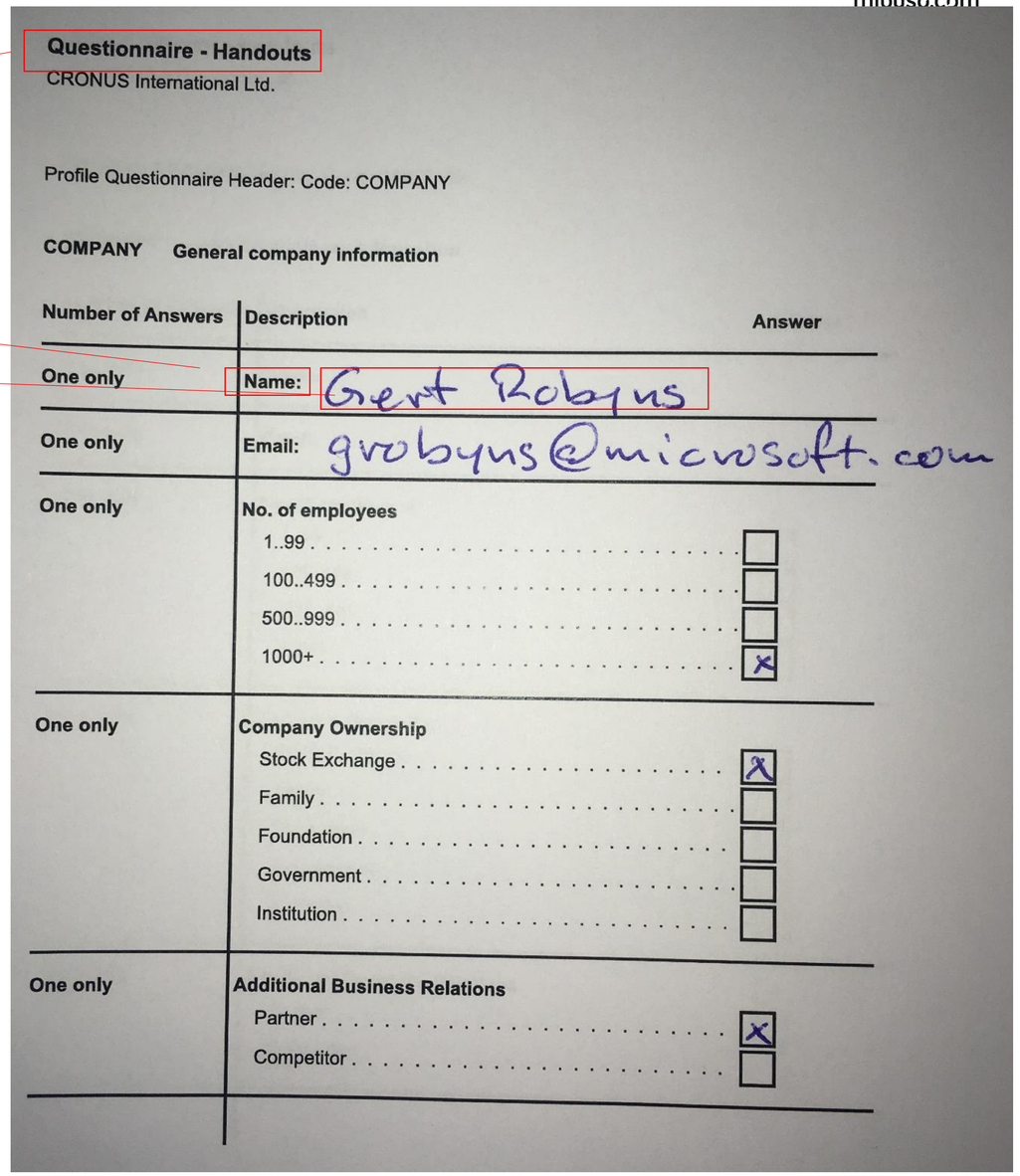

```
{
  "status": "Succeeded",
  "succeeded": true,
  "failed": false,
  "finished": true,
  "recognitionResults": [
    {
      "page": 1,
      "clockwiseOrientation": 0.82,
      "width": 874,
      "height": 826,
      "unit": "pixel",
      "lines": [
        {
          "boundingBox": [15,9,408,10,407,30,14,29],
          "text": "COMPANY General company information",
          "words": [
            {
              "boundingBox": [17,10,114,10,114,30,17,29],
              "text": "COMPANY"
            },
```

json result


```
{"status":"Succeeded",
 "recognitionResult":{
    "lines":[
       {"boundingBox":[241,483,971,502,969,569,239,549],"text":"Questionnaire - Handouts",
          "words":[
             {"boundingBox":[241,483,646,497,645,559,241,550],"text":"Questionnaire"},
             {"boundingBox":[659,497,681,497,680,560,658,559],"text":"-"},
             {"boundingBox":[695,498,968,503,965,570,693,560],"text":"Handouts"}]},

       {"boundingBox":[777,1403,952,1413,949,1475,773,1465],"text":"Name:",
          "words":[
             {"boundingBox":[781,1409,951,1413,949,1476,777,1462],"text":"Name:"}]},
       {"boundingBox":[993,1376,2003,1381,2002,1534,992,1529],"text":"Gert Robyus",
          "words":[
             {"boundingBox":[1002,1383,1393,1382,1393,1529,1005,1525],"text":"Gert","confidence":"Low"},
             {"boundingBox":[1467,1382,1997,1381,1994,1537,1467,1530],"text":"Robyus"}]},
       {"boundingBox":[773,1583,941,1585,941,1646,772,1644],"text":"Email:",
          "words":[{"boundingBox":[778,1586,940,1585,940,1647,777,1643],"text":"Email:"}]},
       {"boundingBox":[987,1547,2832,1535,2833,1700,988,1711],"text":"grobyns@ microsoft.com",
          "words":[
             {"boundingBox":[1015,1560,1722,1554,1726,1700,1020,1700],"text":"grobyns@"},
             {"boundingBox":[1777,1554,2815,1536,2816,1702,1781,1700],"text":"microsoft.com","confidence":"Low"}]},
```

json result


```
{"status":"Succeeded",
 "recognitionResult":{
    "lines":[
       {"boundingBox":[241,483,971,502,969,569,239,549],"text":"Questionnaire - Handouts",
          "words":[
             {"boundingBox":[241,483,646,497,645,559,241,550],"text":"Questionnaire"},
             {"boundingBox":[659,497,681,497,680,560,658,559],"text":"-"},
             {"boundingBox":[695,498,968,503,965,570,693,560],"text":"Handouts"}]},

       {"boundingBox":[777,1403,952,1413,949,1475,773,1465],"text":"Name:",
          "words":[
             {"boundingBox":[781,1409,951,1413,949,1476,777,1462],"text":"Name:"}]},
       {"boundingBox":[993,1376,2003,1381,2002,1534,992,1529],"text":"Gert Robyus",
          "words":[
             {"boundingBox":[1002,1383,1393,1382,1393,1529,1005,1525],"text":"Gert","confidence":"Low"},
             {"boundingBox":[1467,1382,1997,1381,1994,1537,1467,1530],"text":"Robyus"}]},
       {"boundingBox":[773,1583,941,1585,941,1646,772,1644],"text":"Email:",
          "words":[{"boundingBox":[778,1586,940,1585,940,1647,777,1643],"text":"Email:"}]},
       {"boundingBox":[987,1547,2832,1535,2833,1700,988,1711],"text":"grobyns@ microsoft.com",
          "words":[
             {"boundingBox":[1015,1560,1722,1554,1726,1700,1020,1700],"text":"grobyns@"},
             {"boundingBox":[1777,1554,2815,1536,2816,1702,1781,1700],"text":"microsoft.com","confidence":"Low"}]},
```

Questionnaire - Handouts

CRONUS International Ltd.

Profile Questionnaire Header: Code: COMPANY

COMPANY **General company information**

Number of Answers	Description	Answer
One only	Name:	Gert Robyus
One only	Email:	grobyns@microsoft.com
One only	**No. of employees**	
	1..99	☐
	100..499	☐
	500..999	☐
	1000+	☒
One only	**Company Ownership**	
	Stock Exchange	☒
	Family	☐
	Foundation	☐
	Government	☐
	Institution	☐
One only	**Additional Business Relations**	
	Partner	☒
	Competitor	☐

Extension UI

NAV TECH DAYS 2019
mibuso.com

New | More options

Process Picture | Setup

General

Profile Code ..... COMPANY

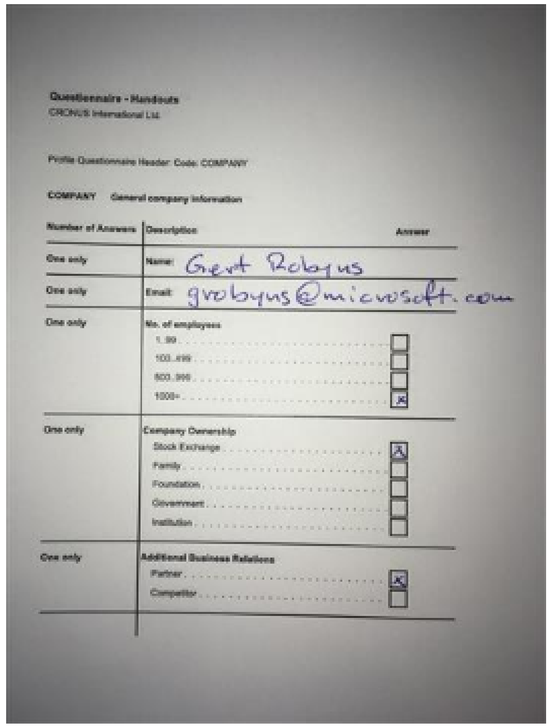
Created On ..... 11/12/2019 3:29 PM

Contact No. .....

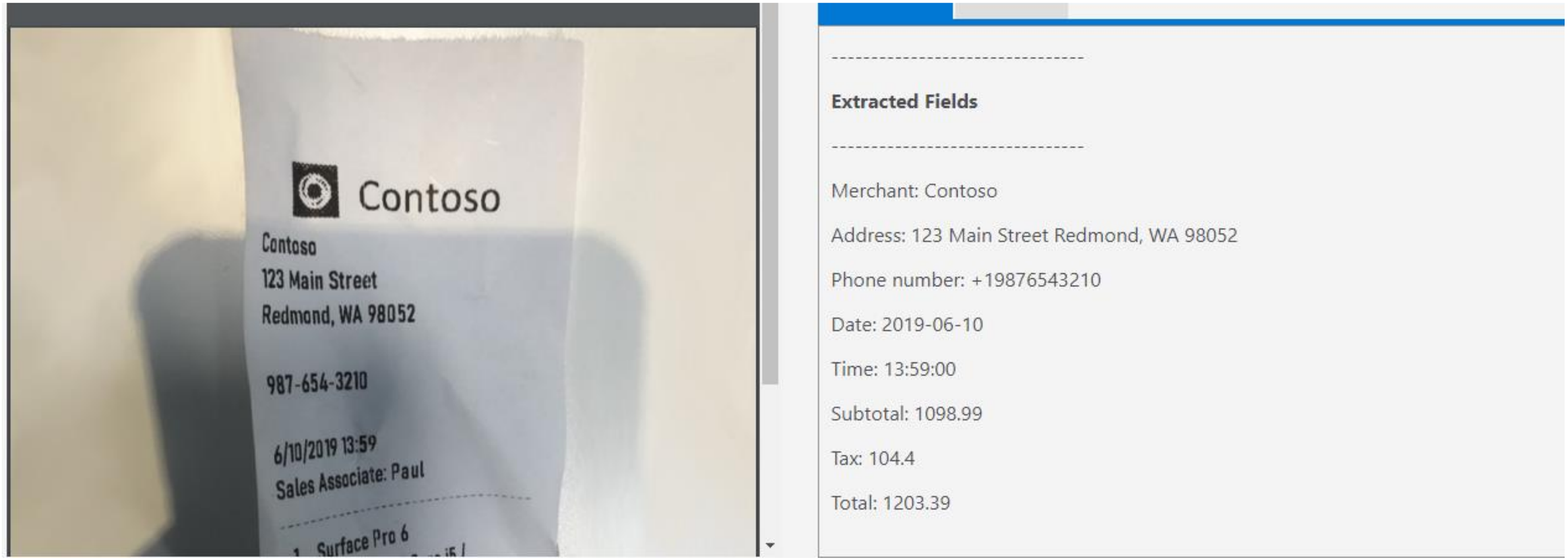
Questionnaire Response Subpage | Manage

	Question	Response Text	Response Selection
→	Name:	Gert Robyus	☐
	Email:	grobyns@ microsoft.com	☐
	No. of employees	1000+	☑
	Company Ownership	Stock Exchange	☑
	Additional Business Relations	Partner	☑

Questionnaire Response Picture

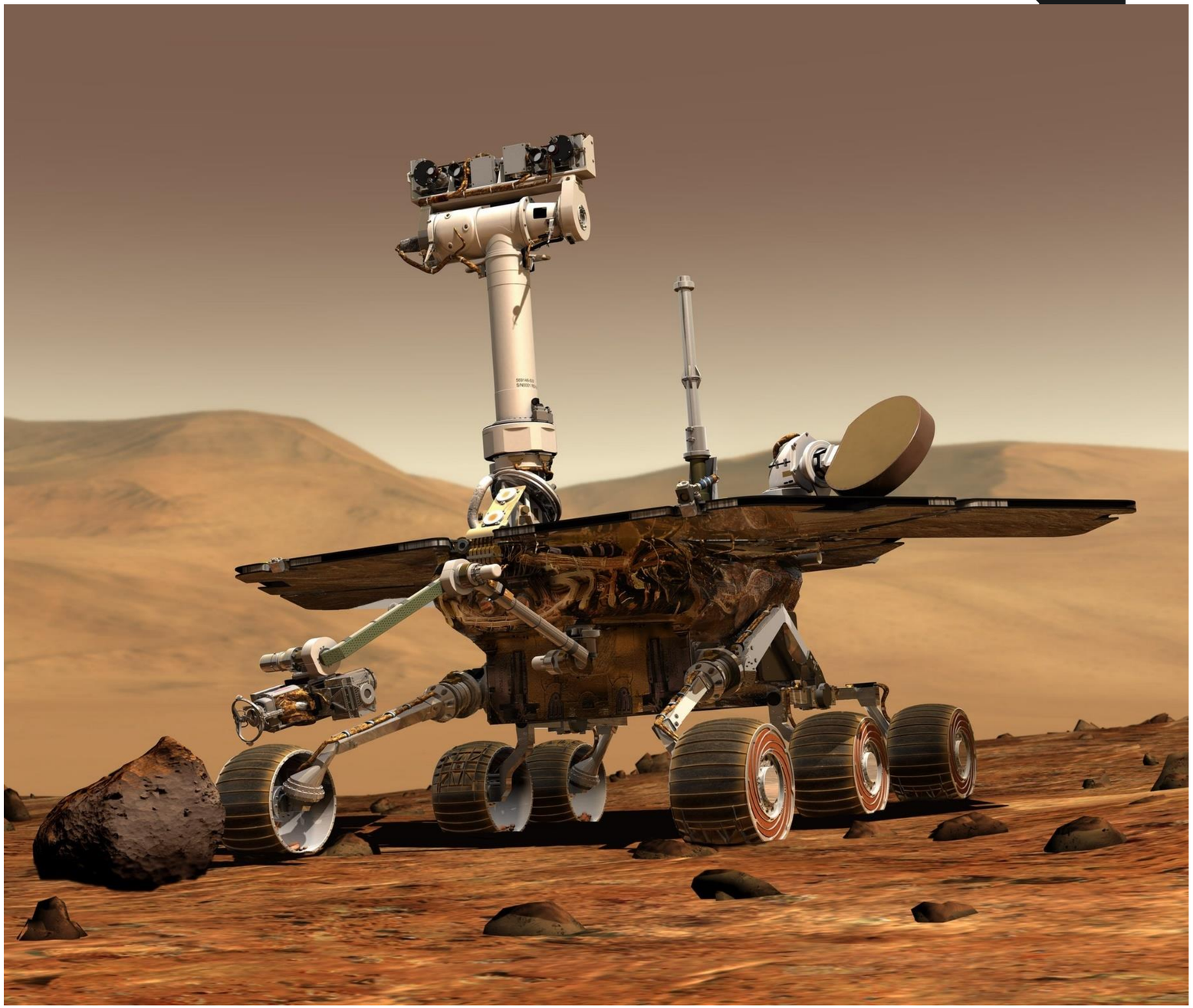


(Preview) Form Recognizer



https://azure.microsoft.com/en-us/services/cognitive-services/form-recognizer/

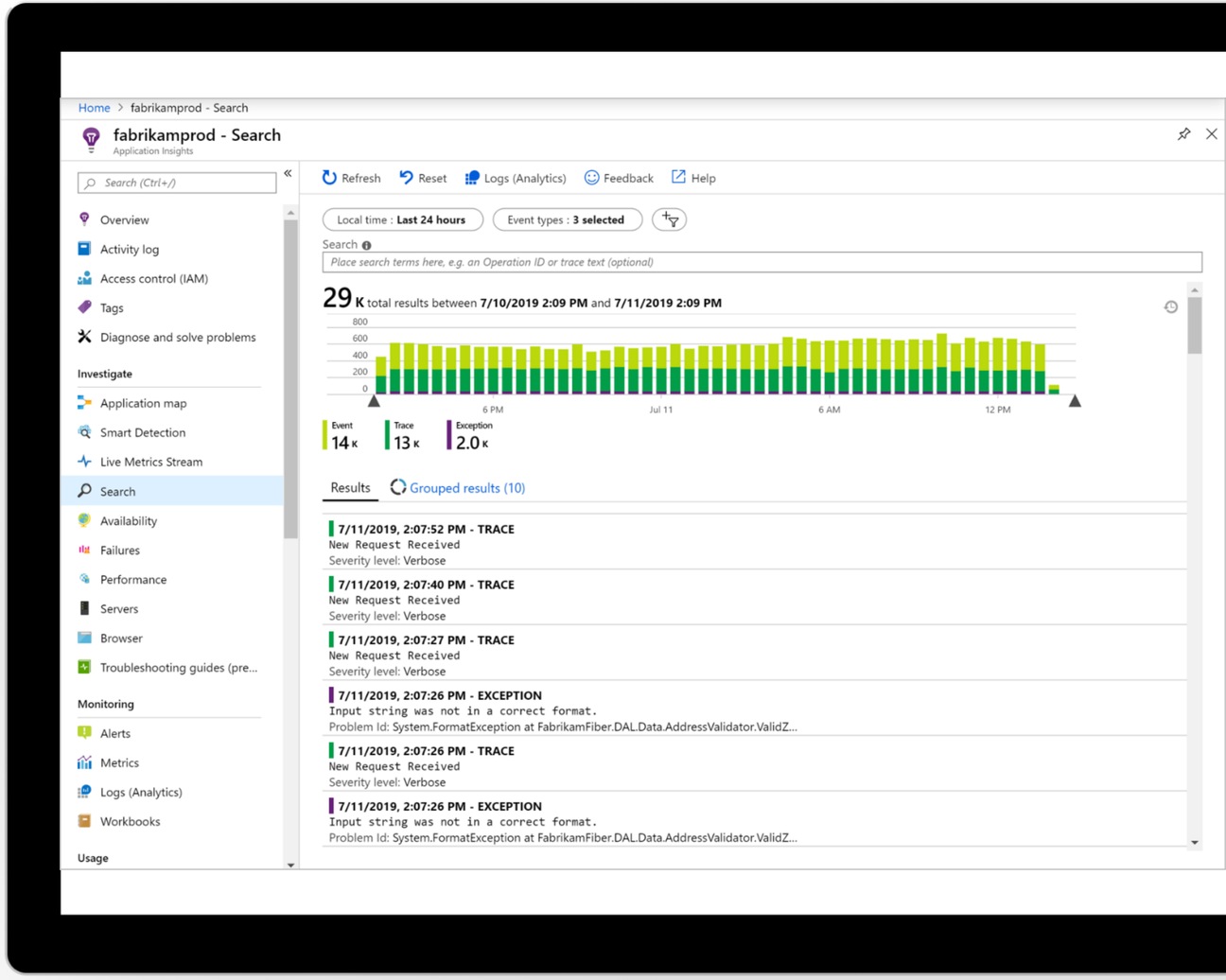
Telemetry



What is telemetry?

From Wikipedia:

“Telemetry is the collection of measurements or other data at remote or inaccessible points and their automatic transmission to receiving equipment for monitoring”



Different types of telemetry

Monitoring and Alerting

“What is going on”

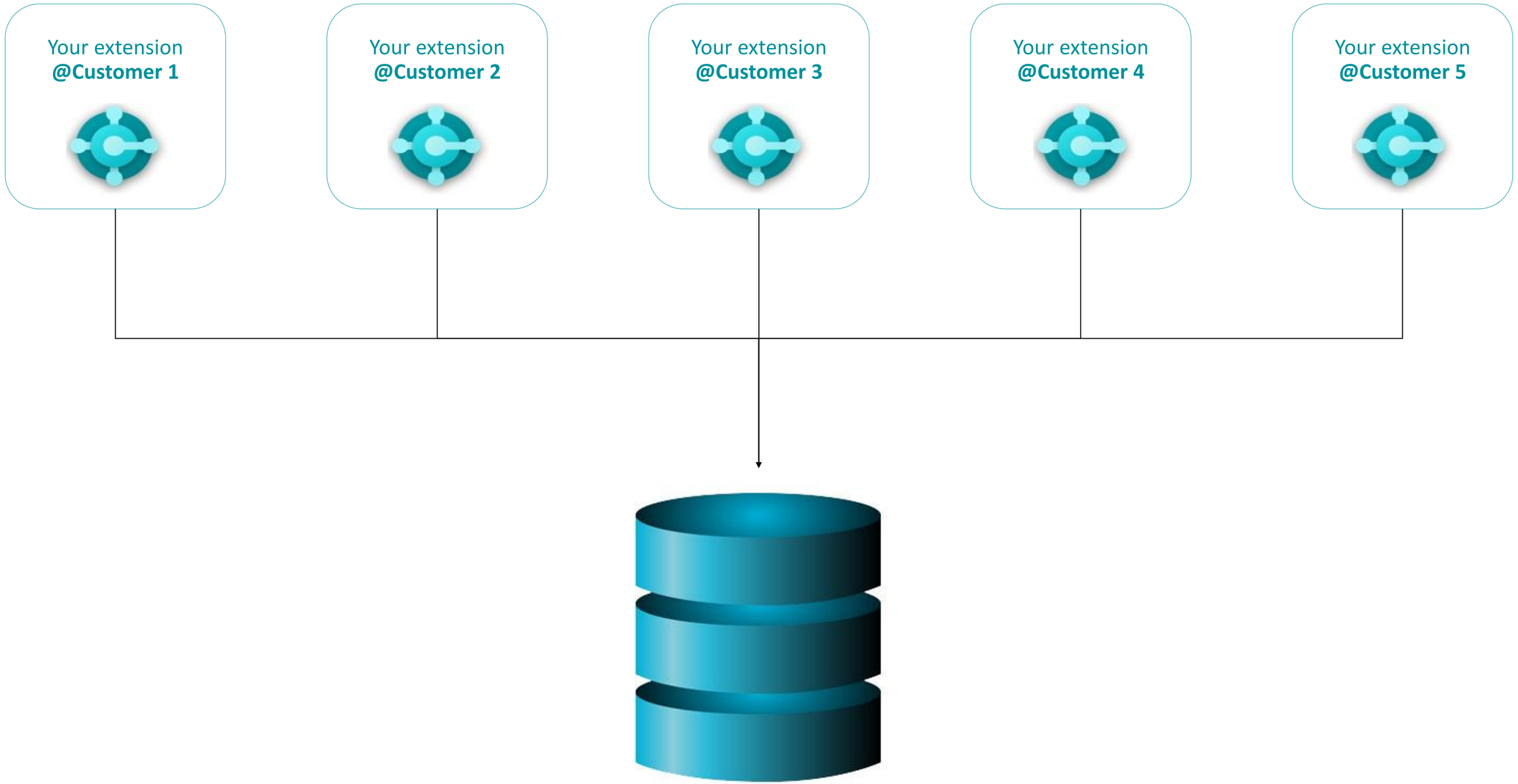
Diagnostics and troubleshooting

“What happened”

Usage

“What features of my application are used and how ”

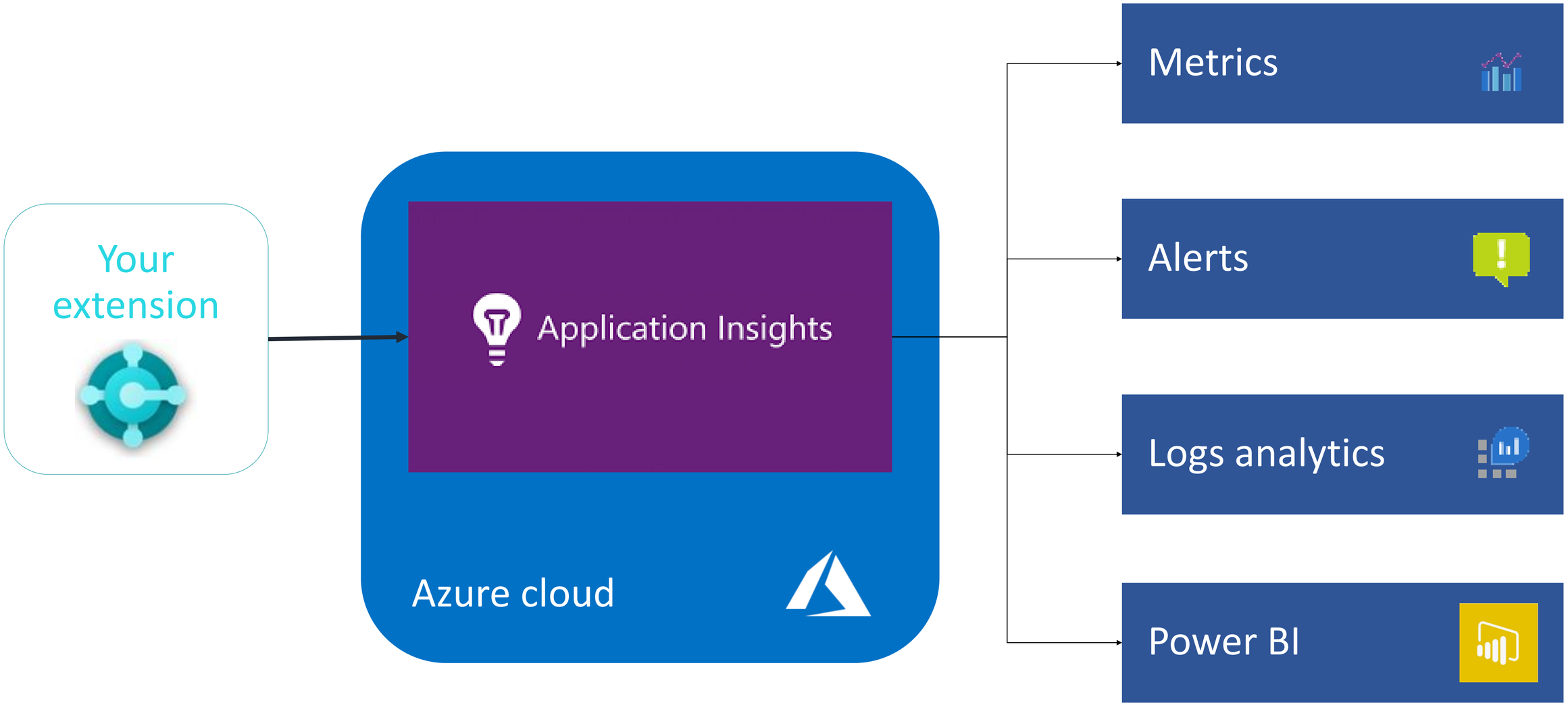
Data aggregation across users and tenants



Introducing:

Application Insights

How does Application Insights works?



Type of telemetry emissions

TrackPageView	TrackTrace	TrackEvent	TrackException	TrackMetric	TrackDependency
Which pages are shown	Diagnostic log messages	User actions Other events User behavior Performance	Failures for diagnosis and trouble-shooting	Performance not related to other event	Logging the duration and frequency of calls to external components that your app depends on

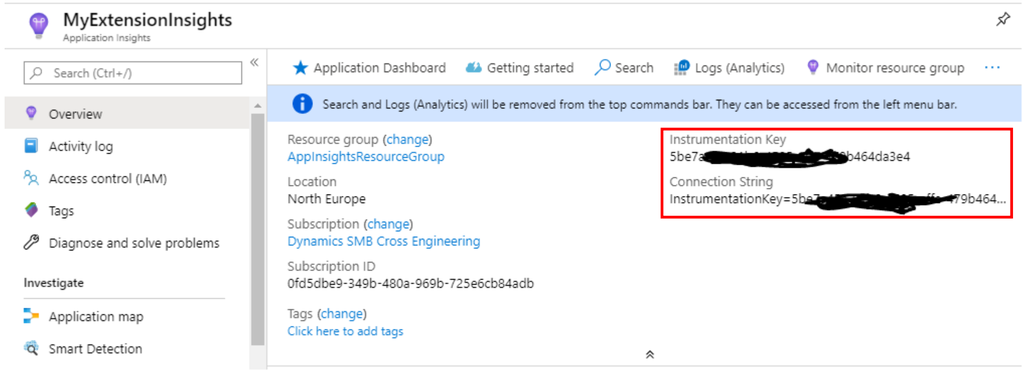
NAV TECH DAYS 2019
mibuso.com

DEMO

Application Insights

Anatomy of an Application Insights telemetry emission

Application Insights instrumentation key



HTTP Post request to https://dc.services.visualstudio.com/v2/track

JSON information in the body

```
{
    "name": "ALExtension.MyExtension",
    "time": "2019-01-21T15:17:34.5563567Z",
    "iKey": "<your instrumentation key>",
    "data": {
        "baseType": "PageViewData",
        "baseData": {
            "ver": 2,
            "name": "MyPage"
        }
    }
}
```

TrackPageView

```
"name": "ALExtension.MyExtension",
"time": "2019-01-21T15:17:34.5563567Z",
"iKey": "<your instrumentation key>",
"data": {
    "baseType": "PageViewData",
    "baseData": {
    "ver": 2,
    "name": "MyPage"
    }
}
```

TrackTrace

```
"name": " ALExtension.MyExtension ",
"time": "2019-01-21T15:17:34.5563567Z",
"iKey": "<your instrumentation key>",
"data": {
    "baseType": "MessageData",
    "baseData": {
        "ver": 2,
        "message": "Some diagnostic message",
        "properties": {
        "property 1": "Value1",
        "property 2": "Value2",
        "property 3": "Value3"
        }
    }
}
```

TrackEvent

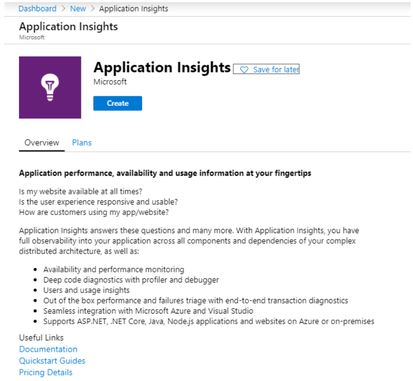
```
"name": "ALExtension.MyExtension",
"time": "2019-01-21T15:17:34.5563567Z",
"iKey": "<your instrumentation key>",
"data": {
    "baseType": "EventData",
    "baseData": {
        "name": "Some event",
        "ver": 2,
        "properties": {
            "Prop1": "Prop1Value",
            "Prop2": "Prop2Value",
            "Prop3": "Prop3Value"
        },
        "measurements": {
            "Metric1": 100.0,
            "Metric2": 11.0,
            "Metric3": 65.0
        }
    }
}
```

TrackException

```
"name": "ALExtension.MyExtension",
"time": "2019-01-21T15:17:34.5563567Z",
"iKey": "<your instrumentation key>",
"data": {
    "baseType": "ExceptionData",
    "baseData": {
        "ver": 2,
        "exceptions": [
        {
            "typeName": "MyExtensionException",
            "message": "Something wrong happened"
        }
        ]
    }
}
```

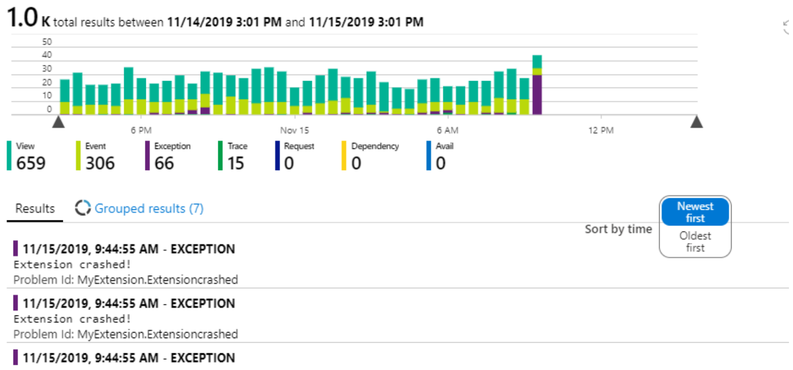
Recipe

1. Create an Application Insights service
2. In AL, create the JSON object with telemetry data
3. Invoke the Application Insights REST API
4. Create dashboards and alerts in Application Insights




```
"name": " ALExtension.MyExtension ",
"time": "2019-01-21T15:17:34.5563567Z",
"iKey": "<your instrumentation key>",
"data": {
    "baseType": "MessageData",
    "baseData": {
        "ver": 2,
        "message": "Some diagnostic message",
        "properties": {
        "property 1": "Value1",
        "property 2": "Value2",
        "property 3": "Value3"
        }
    }
}
```

Post to *https://dc.services.visualstudio.com/v2/track*



Emitting from AL


```
procedure TrackPageView(var PageName: Text) : Text
var
    AppInsightKey: Label '5be7a45c-xxxx-yyyy-zzzz-479b464da3e4';
    URL: label 'https://dc.services.visualstudio.com/v2/track';
    Client: HttpClient;
    Response: HttpResponseMessage;
    Headers: HttpHeaders;
    Content: HttpContent;
    Result: Text;
    JSON: JsonObject;
    JSONData: JsonObject;
    JSONEventData: JsonObject;
    TimeStamp: Text;
begin

    // Build the JSON data
    JSON.Add('name', 'ALExtension.MyExtension');
    TimeStamp := Format(CurrentDateTime(), 0, '<Year4>-<Month,2>-
<Day,2>T<Hours24,2>:<Minutes,2>:<Seconds,2><Second dec.>');
    JSON.Add('time', TimeStamp);
    JSON.Add('iKey', AppInsightKey);

    JSONData.Add('baseType', 'PageViewData');
    JSONEventData.Add('name', PageName);
    JSONEventData.Add('duration', '00:00:00');
    JSONEventData.Add('ver', 2);
    JSONData.Add('baseData', JSONEventData);
    JSON.Add('data', JSONData);
```

```
{
  "name": "ALExtension.MyExtension",
  "time":"2019-11-23T11:09:05.8469124Z",
  "iKey": "<your Application Insights key>",
  "data": {
      "baseType": "PageViewData",
      "baseData": {
          "ver": 2,
          "name": "My page"
      }
  }
}
```

```
    // Add the JSON data to the request content
    Content.Clear();
    Content.WriteFrom(Format(JSON));

    // Add the Content-Type to the Content headers
    Content.GetHeaders(Headers);
    Headers.Remove('Content-Type');
    Headers.Add('Content-Type', 'application/json');

    Client.Post(URL, Content, Response);
    Response.Content().ReadAs(Result);
    Exit(Result);
end;
```

BYO Application Insights key

NAV TECH DAYS 2019
mibuso.com

- Combine your telemetry with Microsoft's
- Available on version 15.x
- Configurable from the Tenant Admin Center
- First version: Long running queries
- Future:
  - Report execution time
  - OData usage

Translation
with Azure cognitive Services

Cognitive Services

NAV TECH DAYS 2019

Vision

Recognize, identify, caption, index, and moderate your pictures, videos, and digital ink content.

- Computer Vision
- Custom Vision
- Face
- Form Recognizer (Preview)
- Ink Recognizer (Preview)
- Video Indexer

Language

Allow your apps to process natural language with pre-built scripts, evaluate sentiment and learn how to recognize what users want.

- Immersive Reader (Preview)
- Language Understanding (LUIS)
- QnA Maker
- Text Analytics
- Translator Text

Speech

Convert speech into text and text into natural-sounding speech. Translate from one language to another and enable speaker verification and recognition.

- Speech Services
- Speaker Recognition (Preview)
- Bing Speech (Retiring)

Decision

Build apps that surface recommendations for informed and efficient decision-making.

- Anomaly Detector (Preview)
- Content Moderator
- Personalizer

Search

Add Bing Search APIs to your apps and harness the ability to comb billions of webpages, images, videos, and news with a single API call.

- Autosuggest Search
- Custom Search
- Entity Search
- Image Search
- Local Business Search (Preview)
- News Search
- Spell Check
- Video Search
- Visual Search
- Web Search

Cognitive Service Containers

Container support in Azure Cognitive Services allows developers to use the same rich APIs that are available in Azure, and enables flexibility in where to deploy and host the services that come with Docker containers.

- Cognitive Service Containers
- Create containers for reuse
- Deploy and run container on Azure Container Instance
- Deploy to Azure Kubernetes Service
- Use Docker Compose to deploy multiple containers
- Cognitive Services containers FAQ

https://docs.microsoft.com/en-us/azure/cognitive-services/

DEMO

XLIFF files

Microsoft Azure (Preview) | Report a bug | Search resources, services, and docs (G+/)

Dashboard > Cognitive Services > Marketplace > Translator Text

Translator Text

Microsoft

Overview

Microsoft Tra
any solution r
and more.

Extend the r

Translate text

Transliterate

Display text ir
alphabet, and

Bilingual dic

Find alternati

Add both on

With the Micr
translation ca
when it is offl

Automaticall

Easily and acc

Build custom

With Custom
vocabulary.

Useful Links
More about Translator Text API
Documentation
API reference
Pricing
Regional availability

Dashboard > BCTranslator - Keys

BCTranslator - Keys

Cognitive Services

Search (Ctrl+/)

- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems

RESOURCE MANAGEMENT

- Keys
- Quick start
- Pricing tier
- Billing By Subscription
- Properties

Regenerate Key1 | Regenerate Key2

NAME

BCTranslator

KEY 1

[redacted]

KEY 2

[redacted]

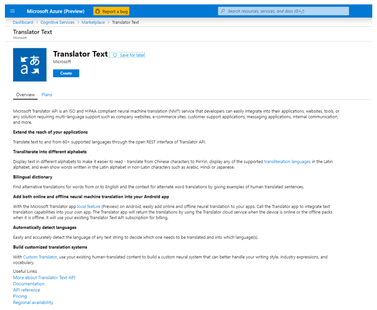
Translator Text REST API

POST *https://api.cognitive.microsofttranslator.com/translate?api-version=3.0&from=sourcelanguage&to=targetlanguage*

Body of the request

```
[
{
  "Text": "Customer name"
},
{
  "Text": "Balance"
},
{
  "Text": "Total sales"
},
{
  "Text": "Phone no."
},
{
  "Text": "Address"
},
…
```

Recipe



1. Create a Translator Text service
2. Generate the XLIFF file
3. Extract the strings from the XLIFF file
4. Invoke the Translator Text API
5. Parse the json result
6. Write the strings back to the XLIFF file

In app.json add:

```
"features": ["TranslationFile"]
```

```
{
  "Text": "Customer name"
},
{
  "Text": "Balance"
},
…
```

https://api.cognitive.microsofttranslator.com/translate?api-version=3.0&from=sourcelanguage&to=targetlanguage

```
{
  "translations": [
    {
      "text": "Kundenname",
      "to": "de"
    }
  ]
},
{
  "translations": [
    {
      "text": "Gleichgewicht",
      "to": "de"
    }
  ]
},
…
```

```
<trans-unit id="Table 3881164589 …
  <source>Customer name</source>
  <target>Kundenname</target>
</trans-unit>
```

NAV TECH DAYS 2019
mibuso.com

NAV TECH DAYS 2019
mibuso.com

DEMO

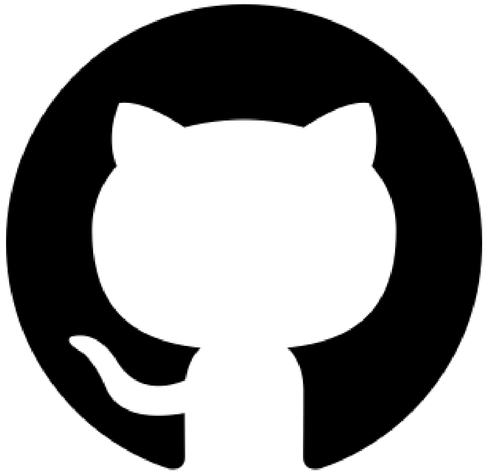
Machine translation in action in a VS Code extension

How much does all this cost?

NAV TECH DAYS 2019
mibuso.com

	Free units included	Price
Application Insights	5GB per customer per month	~€2 per GB for more than 5GB
Translator Text	2M characters per month	~€8 per additional million characters
OCR/Text recognition	-	~€1-€2 per 1000 transactions
IoT Central	5 devices	~€1.7 per device/month
Logic Apps		~€2.2 per 100k executions
Key Vault		~€2.7 per million transactions
Service Bus		~€0.043 per million operations
Blob Storage		~€0.001 per GB per month
Service Bus Relay	includes 5 GB of data per month	~€8.25 per listener

https://azure.microsoft.com/en-us/pricing/calculator/



https://www.github.com/Microsoft/BCTech

https://aka.ms/BCTech

NAV TECH DAYS 2019
mibuso.com

Thank You!