



10 YEAR ANNIVERSARY  
10 YEAR ANNIVERSARY

# Universal Code – Target must be cloud

Esben Nyhuus Kristoffersen  
Microsoft

Michael Nielsen  
ForNAV



# Objectives

- 1 **New customers** should be able to purchase Dynamics 365 Business Central online and find the apps they need on Microsoft AppSource
- 2 **Existing customers** want to see a vision of how their partners will support them in their digital transformation journey
- 3 Cloud-first partners spend less time on upgrade projects and have more capacity **to provide industry value and serve more customers**
- 4 When partner code is isolated from Microsoft base app code, **we open doors for innovation**
- 5 **Replace legacy programs** 'Registered Solution Program' and 'Certified for Microsoft Dynamics program'





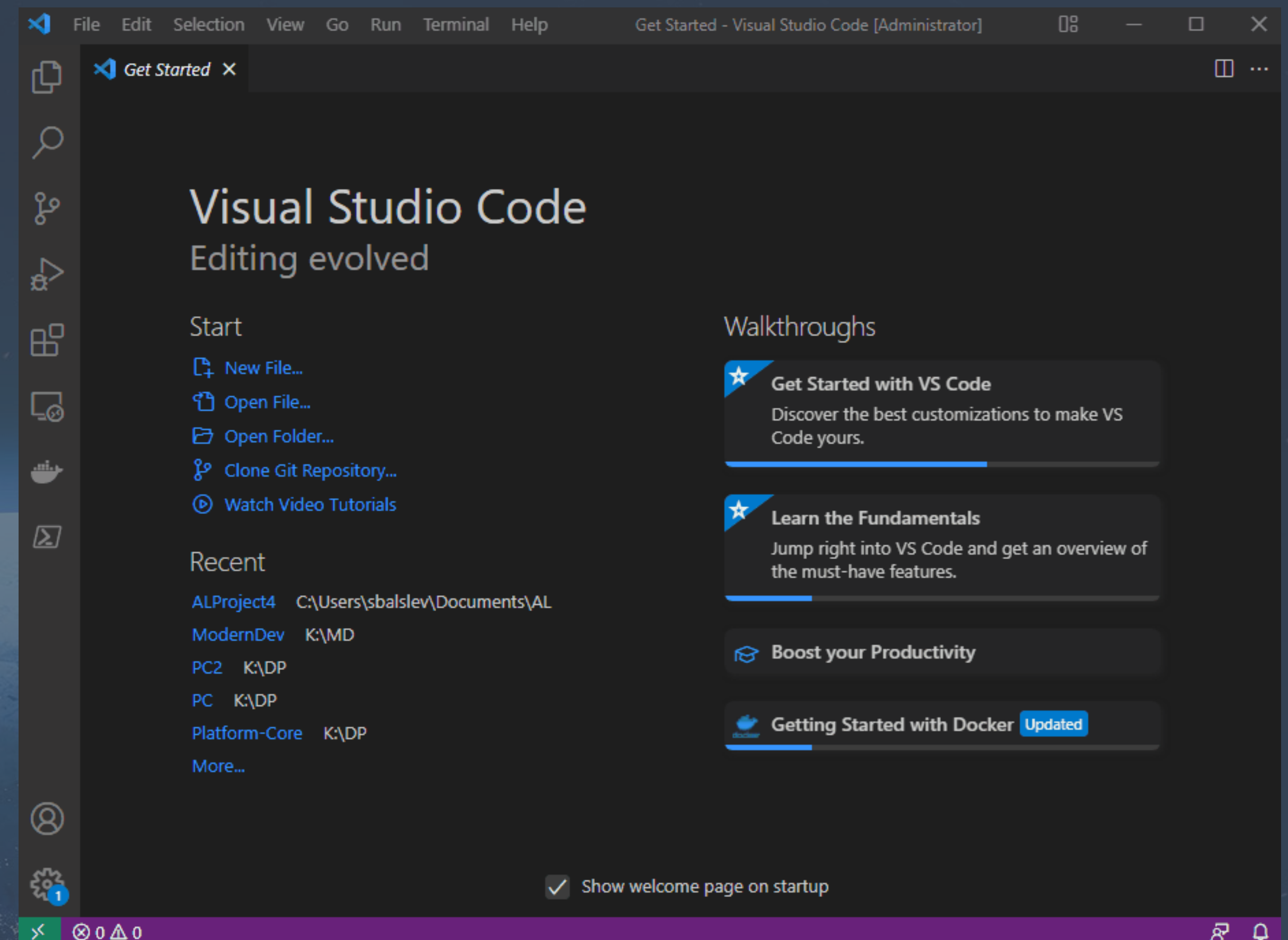
# 'Universal Code' are Cloud-optimized extensions (Target: "Cloud")

**Universal code** can be  
Implemented everywhere

Reach more customers by  
listing **universal code** on  
Microsoft AppSource

Add customer stickiness by  
bundling **universal code** with  
other Microsoft cloud services

**Universal code** frees up  
capacity and allows for greater  
innovation and increased market  
share

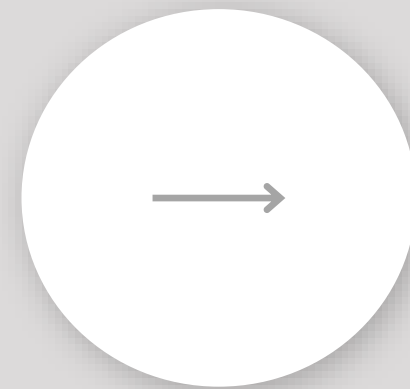


# What is the “Universal Code” initiative?

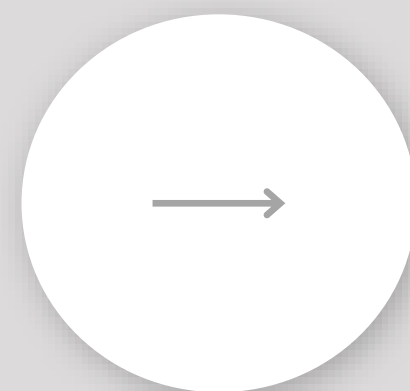
## Initiative details



Q4CY22



Business Central on-premise platform will only **allow “Universal Code” for customizations** in customer environments. This will be enforced by the platform (i.e., technical enforcement). \*



**Use of non-“Universal Code” modules** will require additional modules to be licensed.

\* Any Business Central implementations created prior to October 3, 2022, as indicated by the implementation’s initial product line registration date of Business Central in an implementation’s license key, do NOT require the non-‘Universal Code’ module (or fee). This exemption includes Business Central implementations created prior to October 3, 2022, as a result of a transition from NAV/GP/SL.

# Disclaimer

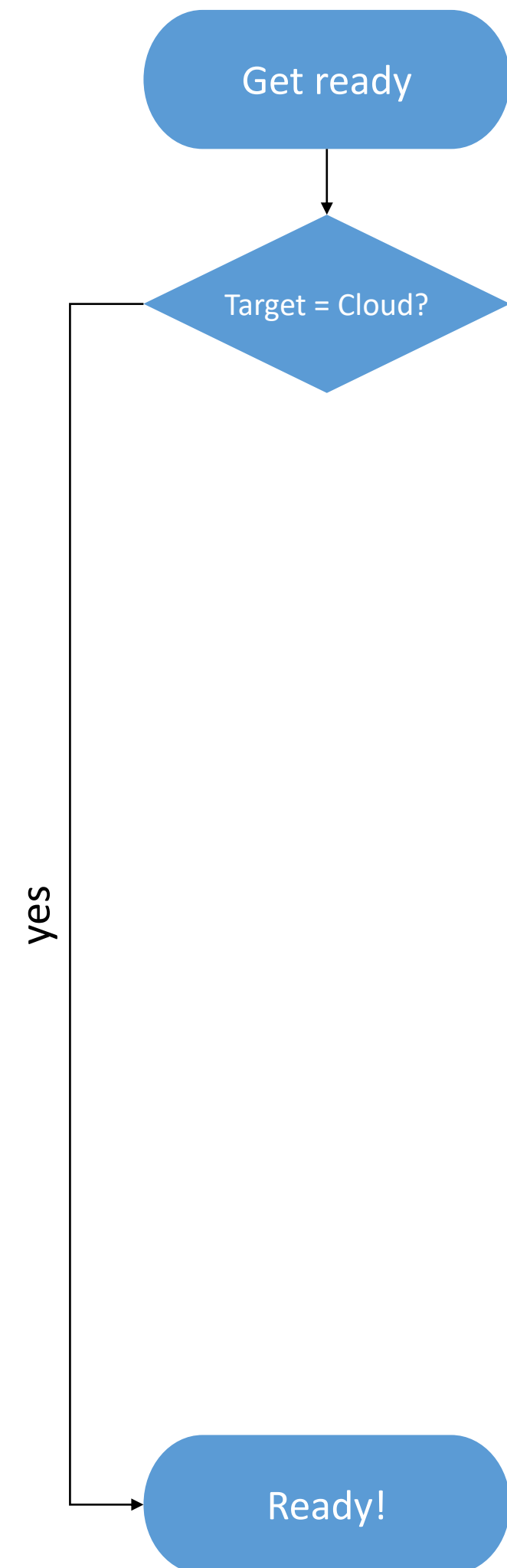
There is **not one easy fix**

Many **different obstacles** for old on-prem solutions

This is just **some possible solutions** to solve some problems



# Getting to Universal Code



Get ready for the Universal Code Initiative

Check the “Target” setting in app.json

You are ready!



A close-up photograph of a glass filled with beer. The glass is covered in condensation droplets, and the beer has a thick, white head of foam. The word "BEERHAUS" is printed in bold, black letters on the lower part of the glass. The background is dark and out of focus.

# The Beer Organizer

A not so business-like kind of extension



# Features

The Virtual Beer Gallery: All Search Import Delete Edit List Rate More options

20180410\_171459529\_iOS.jpg

20190330\_115435875\_iOS.jpg

20190717\_11002713

20190722\_155541722\_iOS.jpg

201904

20190414\_094814616\_iOS.jpg

20180410\_171459529\_iOS.jpg

20190414\_094814616\_iOS.jpg

20190414\_094814616\_iOS.jpg

Extension Management | Work

Search Details Man.

View

Microsoft Application

Microsoft Base Application

Microsoft DemoTool

Microsoft System Application

BCTech The Beer Organizer

Delete Edit List Rate

⚡ ★

⚡ ★ ★

⚡ ★ ★ ★

⚡ ★ ★ ★ ★

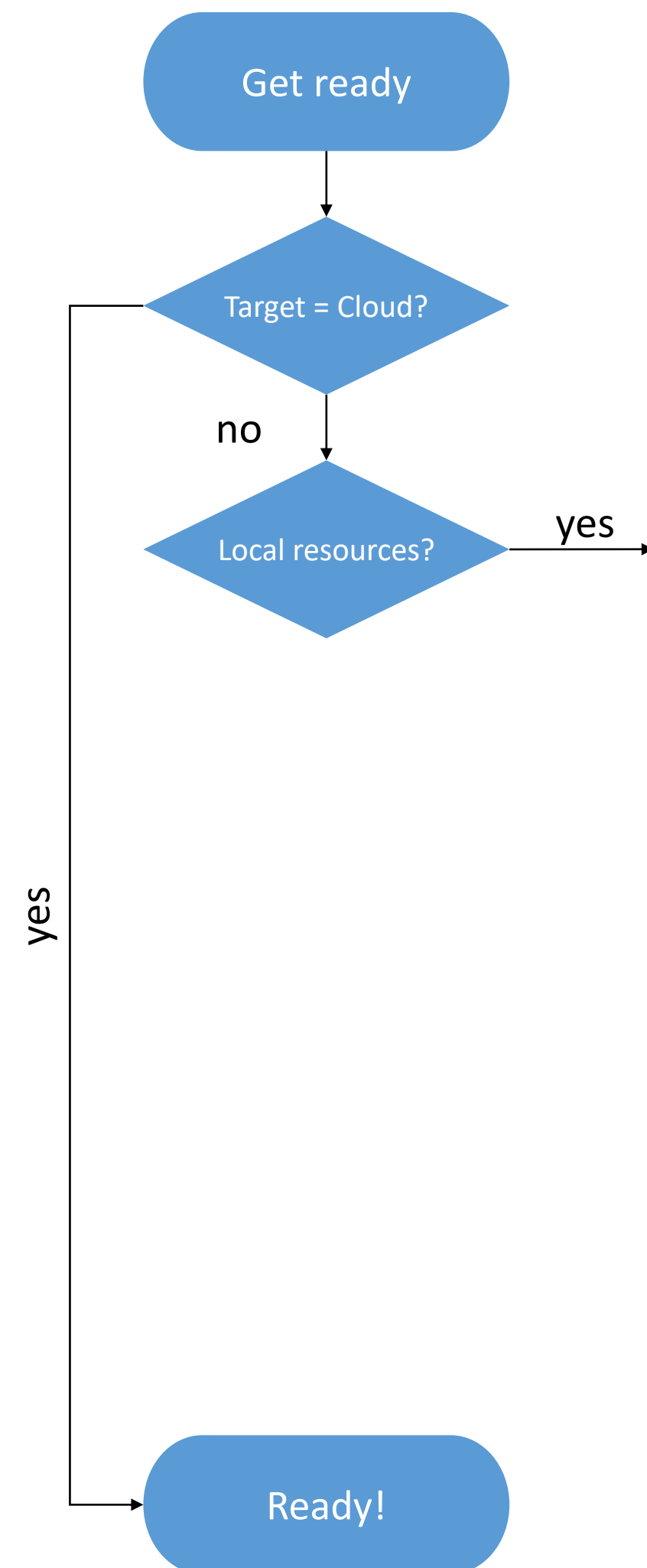
⚡ ★ ★ ★ ★ ★

c:\temp\beers\20190620\_13314...

20180410\_171459529\_iOS.jpg



# Getting to Universal Code



Accessing local resources (files, network, local hardware, ...)



# “Brew First Batch”



The Virtual Beer Gallery: All Search Import Delete



c:\temp\beers\20161129\_1...



c:\temp\beers\20190620\_1...



c:\temp\beers\20190515\_1...



# "Brew First Batch" – OnPrem version

```
procedure BrewFirstBatch()
var
    FileManagement: Codeunit "File Management";
    NameValueBuffer: record "Name/Value Buffer" temporary;
    FileNo: Integer;
    MyDialog: Dialog;
    Text000: Label 'Loading beers: #1';
begin
    FileManagement.GetServerDirectoryFilesList(
        NameValueBuffer, FirstBatchServerPath);

    if NameValueBuffer.FindSet() then begin
        FileNo := 0;
        MyDialog.Open(Text000, FileNo);
        repeat
            Brew(NameValueBuffer.Name, NameValueBuffer.Name);
            FileNo := FileNo + 1;
            MyDialog.Update();
        until NameValueBuffer.Next() = 0;
        MyDialog.Close();
    end;
end;
```

```
local procedure Brew(Path: Text; Name: Text)
var
    File: File;
    Ins: InStream;
    BeerRec: record BeerTable;
begin
    File.TextMode := false;
    File.Open(Path);
    File.CreateInStream(Ins);
    BeerRec.Id := CreateGuid();
    BeerRec.Name := Name;
    BeerRec.Image.ImportStream(Ins, Name, 'image/jpeg');
    BeerRec.Insert();
    File.Close();
end;
```



# Universal Code version

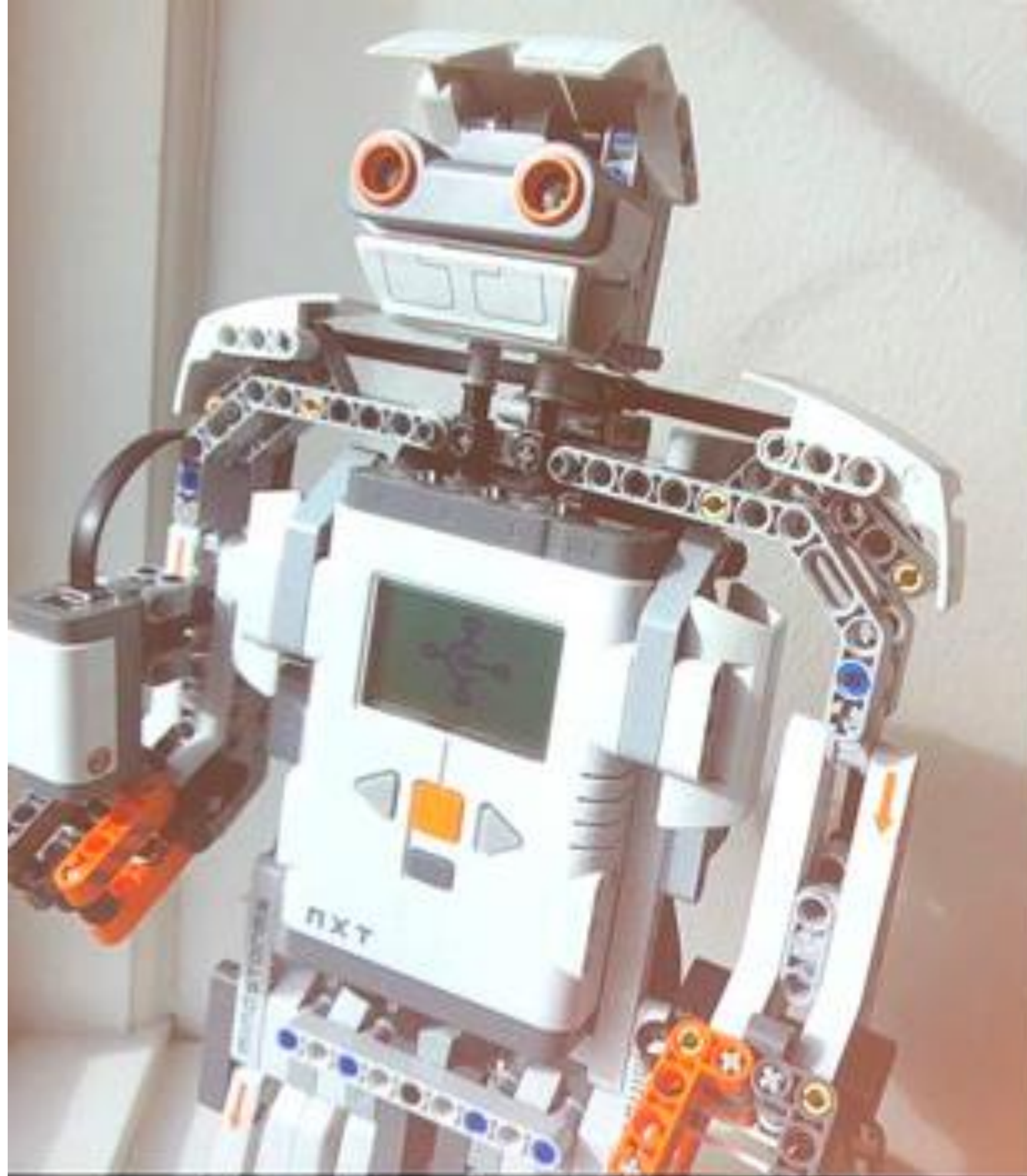
```
procedure BrewFirstBatch()
var
    ABSBlobClient: Codeunit "ABS Blob Client";
    ABSContainerContent: Record "ABS Container Content";
    StorageServiceAuthorization: Codeunit "Storage Service Authorization";
    Response: Codeunit "ABS Operation Response";
    Authorization: Interface "Storage Service Authorization";
    TempBlob: Codeunit "Temp Blob";
    Ins: InStream;
    ImageRec: Record BeerTable;
begin
    Authorization := StorageServiceAuthorization.CreateSharedKey(BeerBatchKey);

    ABSBlobClient.Initialize(
        'esbenkbeerstorage',
        'beers',
        Authorization);
    if ABSBlobClient.ListBlobs(ABSContainerContent).IsSuccessful() then begin
        if ABSContainerContent.FindSet() then begin
            repeat
                Clear(TempBlob);
                TempBlob.CreateInStream(Ins);
                if ABSBlobClient.GetBlobAsStream(ABSContainerContent.Name, ins).IsSuccessful() then begin
                    ImageRec.Id := CreateGuid();
                    ImageRec.Name := ABSContainerContent.Name;
                    ImageRec.Image.ImportStream(ins, ABSContainerContent."Full Name", 'image/jpeg');
                    ImageRec.Insert();
                end;
            until ABSContainerContent.Next() = 0;
        end;
    end;
end;
```



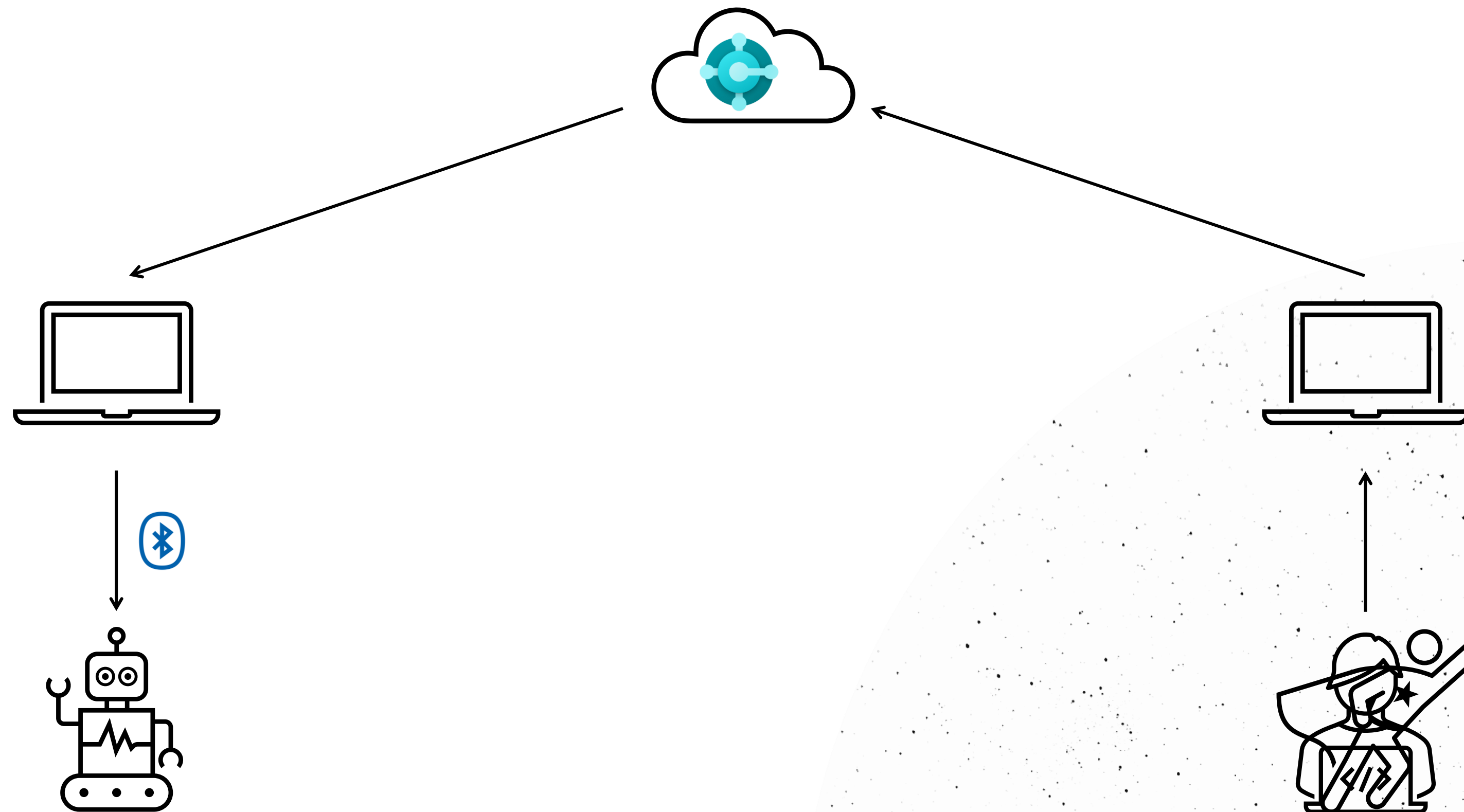
# BC Local Agent

Access to local resources from  
cloud



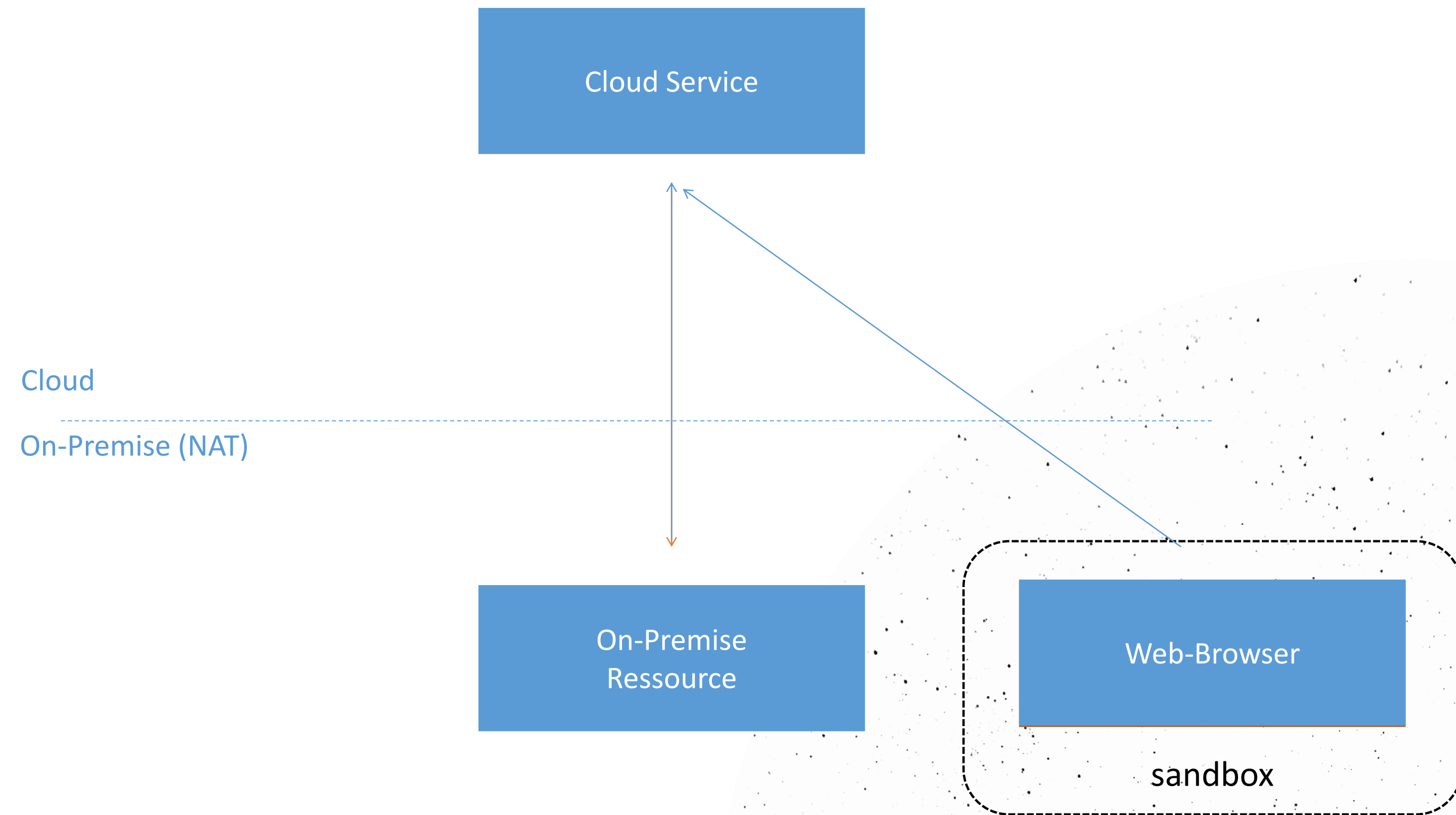


# The Robot Demo



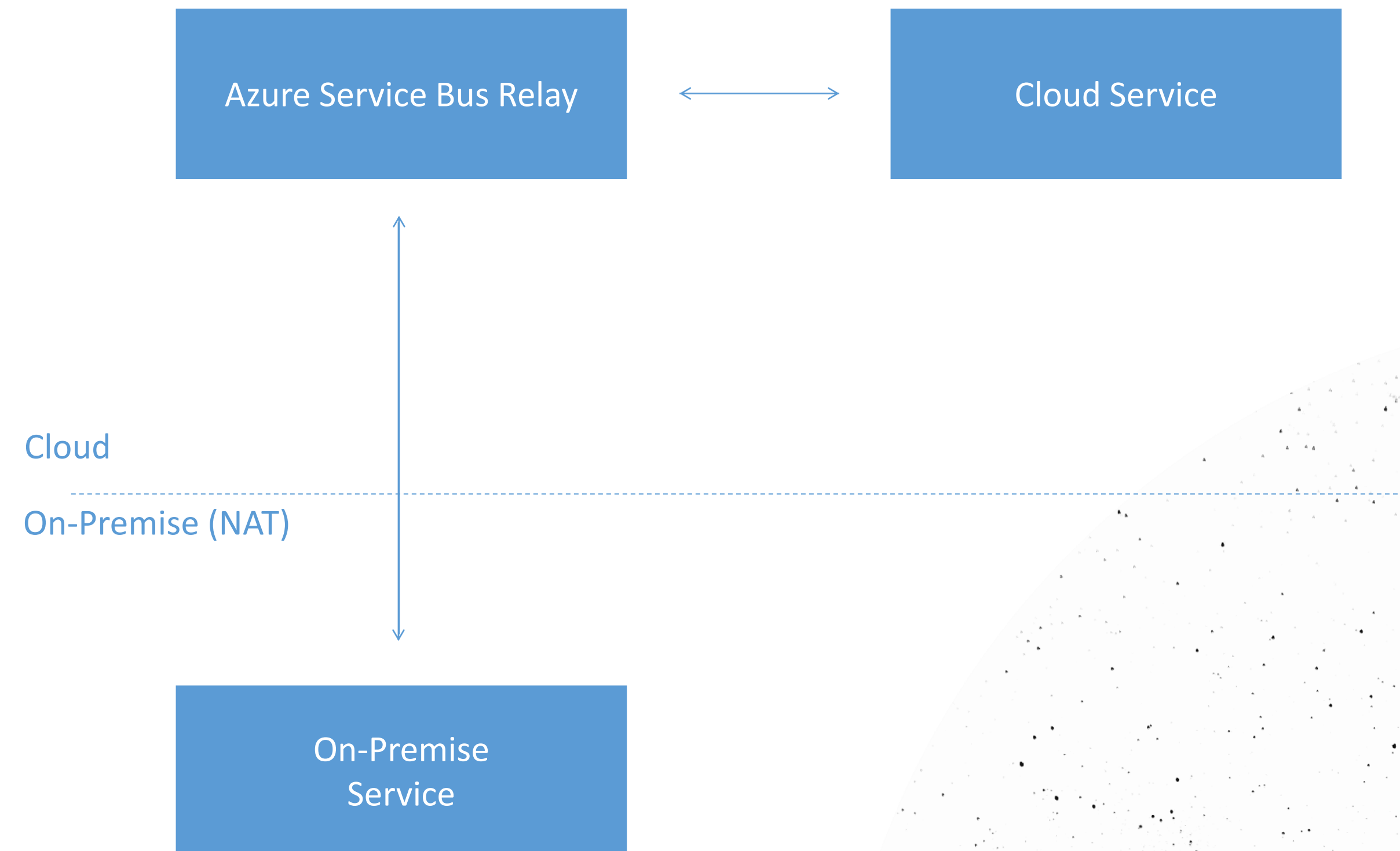


# "The problem"



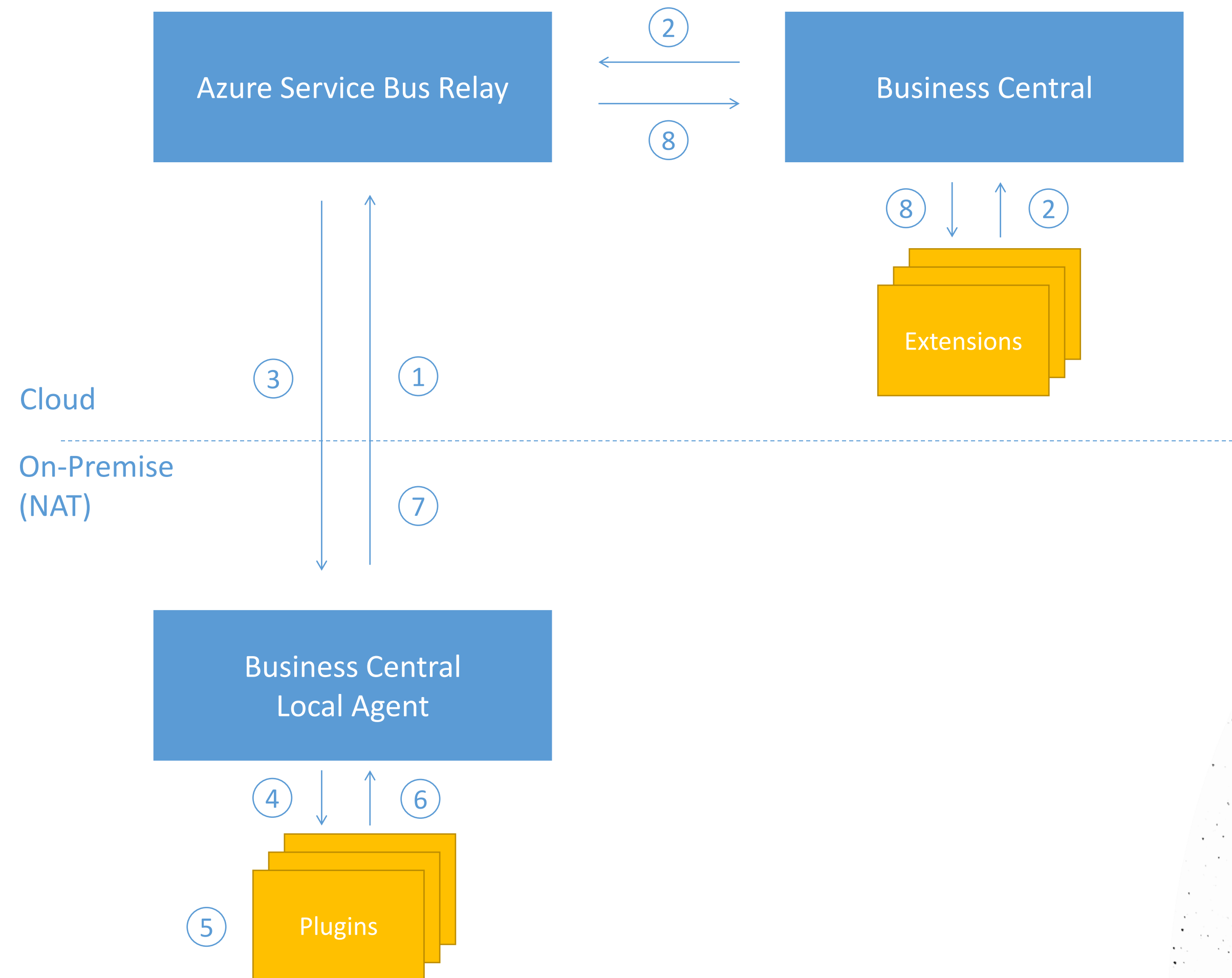


# Azure Service Bus Relay





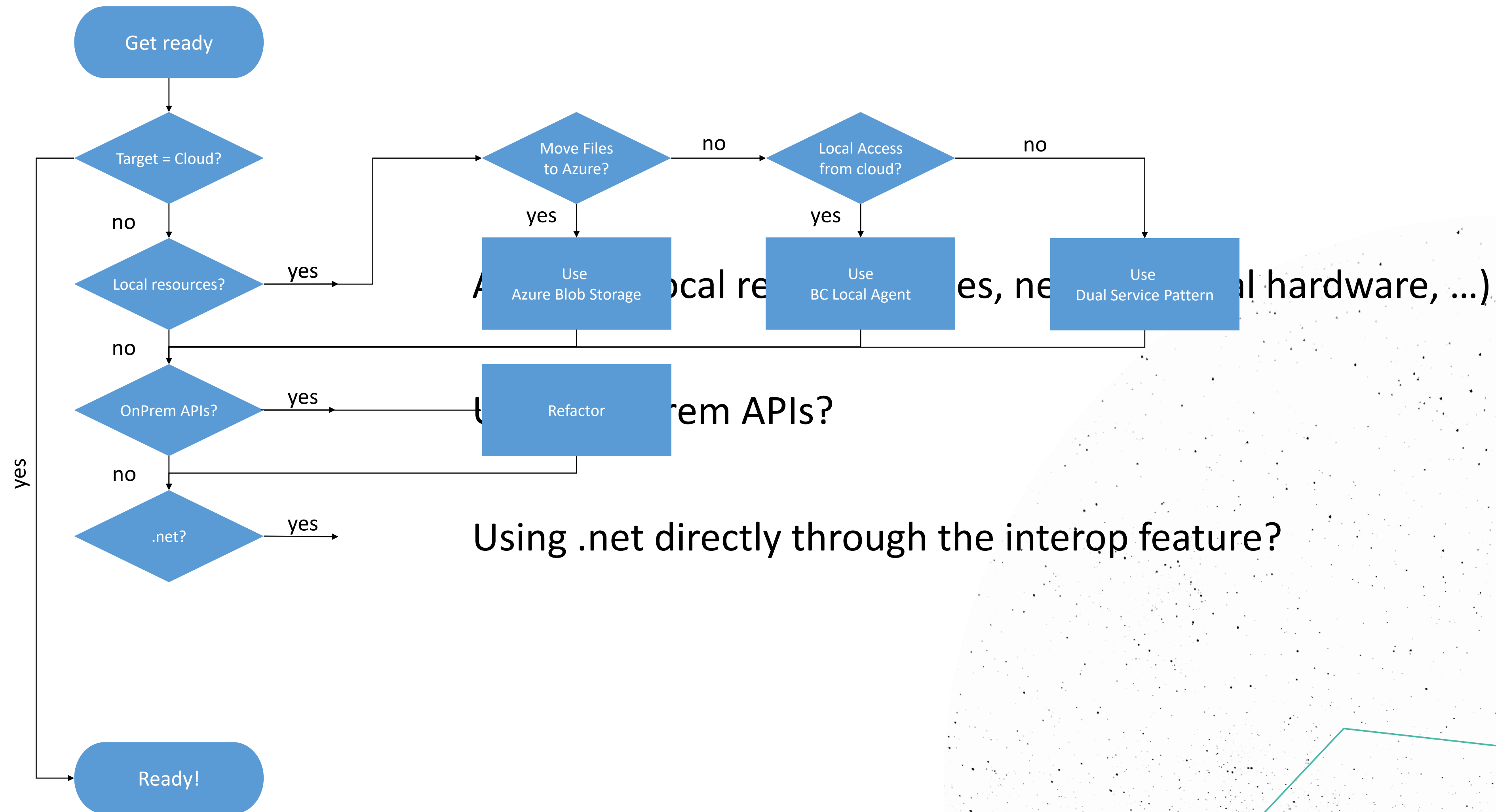
# BC Local Agent



- 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



# Getting to Universal Code







# Rate Beers

The Virtual Beer Gallery: **All** ▾ | 🔍 Search Import ▾ | 🗑 Delete ✎ Edit List **Rate** ▾ | More options


🔗 🔍 🖨 ↗




20190330\_103724215\_iOS.jpg



20190620\_133149946\_iOS.jpg



20190414\_094814616\_iOS.jpg



20190731\_122125308\_iOS.jpg

⚡ ★

⚡ ★ ★

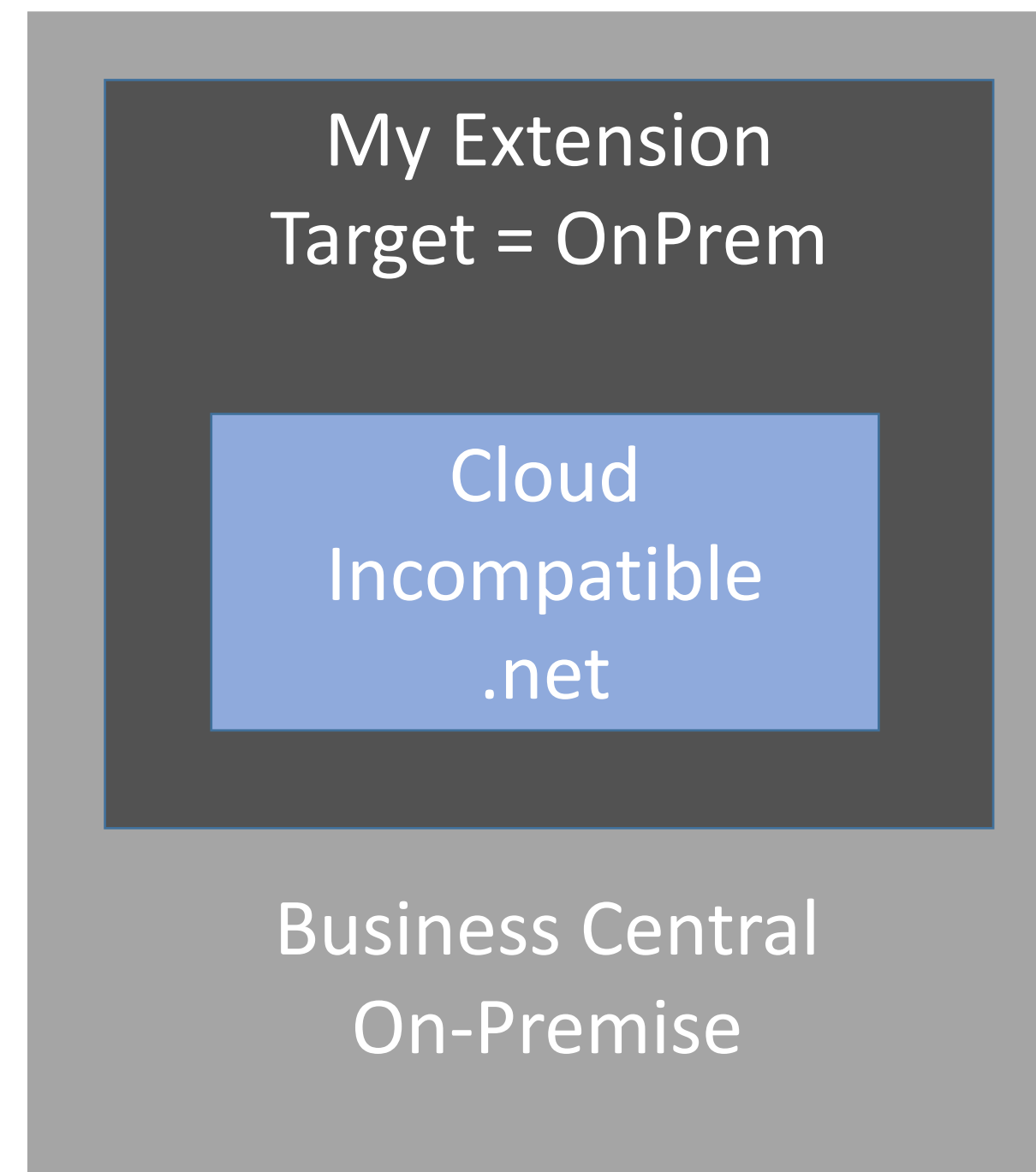
⚡ ★ ★ ★

⚡ ★ ★ ★ ★

⚡ ★ ★ ★ ★ ★

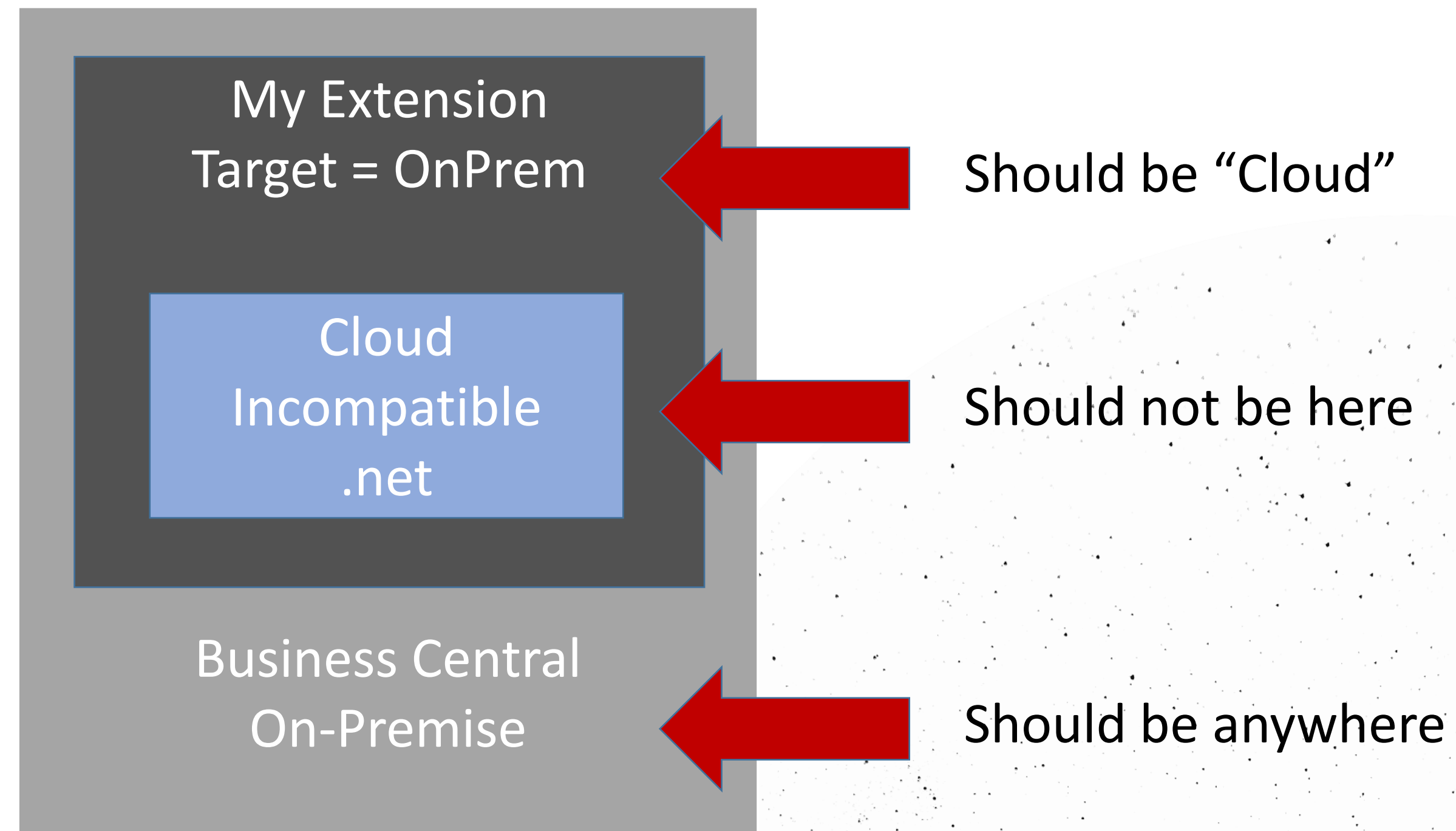


# “The problem”

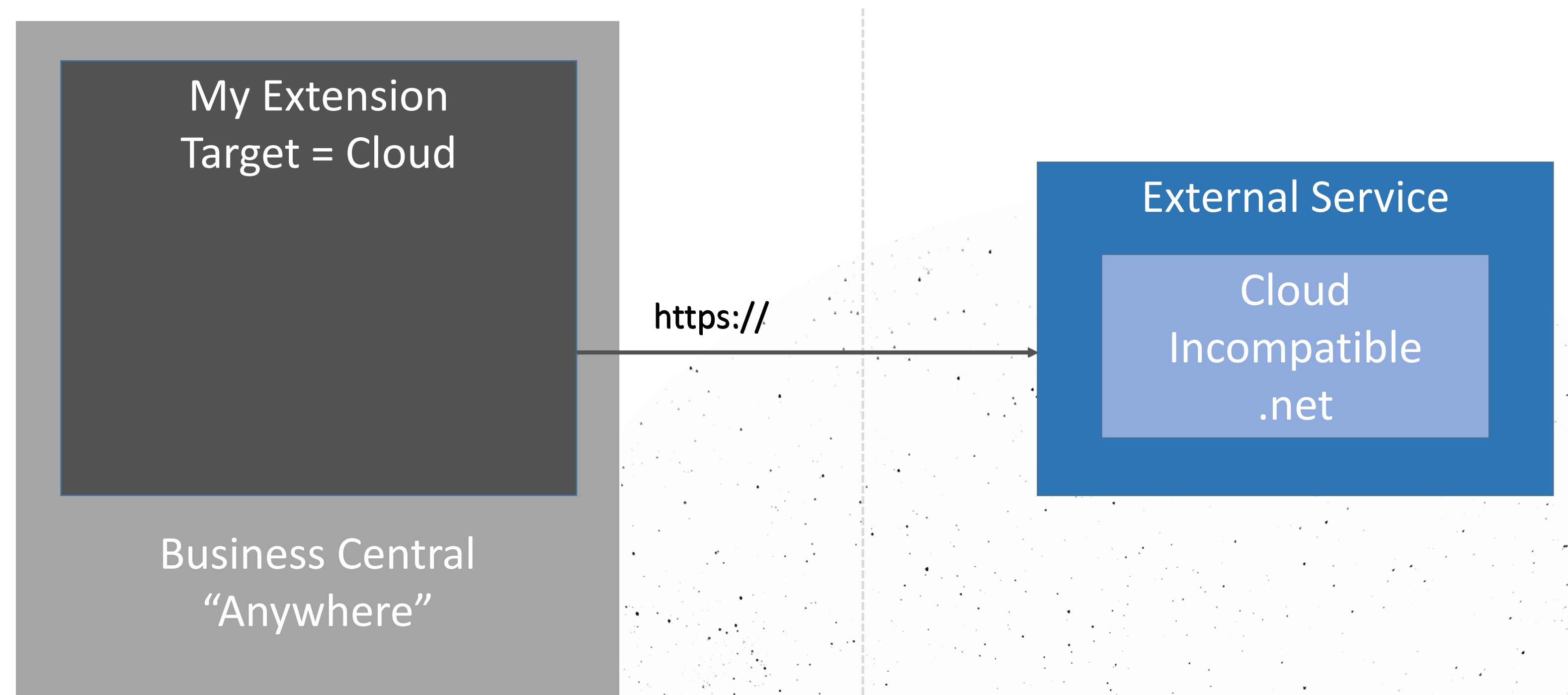




# “The problem”

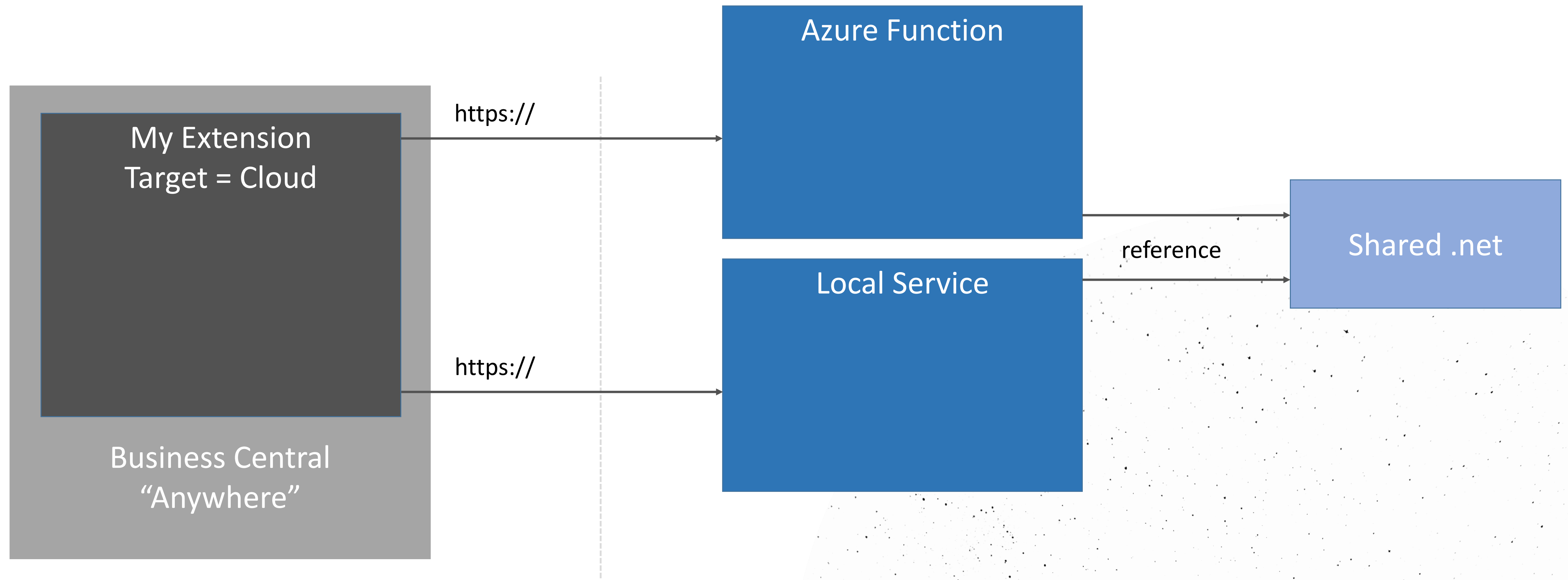


# A solution





# The dual service pattern



# 2 Services – 1 Interface

```
interface IService
{
    /// <summary>
    /// Adds a star-rating watermark with <paramref name="Rating"/> of
    /// start to a Base64 encoded jpeg image and returns
    /// the updated image as Base64 encoded string.
    /// </summary>
    /// <param name="ImageStr">The original picture encoded as Base64</param>
    /// <param name="Rating">The number of stars</param>
    /// <returns>The image with rating as a Base64 encoded string.</returns>
    procedure AddRating(ImageStr: Text; Rating: Integer): Text;
}
```



# 2 Services – 1 Interface

```
codeunit 50141 Service
{
    SingleInstance = true;

    var
        IsInitialized: Boolean;
        Service: Interface IService;

    /// <summary>
    /// Gets environment dependent implementation
    /// of the IService interface
    /// </summary>
    /// <returns>An IService implementation</returns>
    procedure Instance(): interface IService;
    begin
        if not IsInitialized then begin
            Initialize();
            IsInitialized := true;
        end;

        exit(Service);
    end;

    local procedure Initialize();
    var
        Environment: Codeunit "Environment Information";
        SaaSServiceImpl: codeunit SaaSServiceImpl;
        OnPremServiceImpl: Codeunit OnPremServiceImpl;
    begin
        if Environment.IsSaaS() then
            Service := SaaSServiceImpl
        else
            Service := OnPremServiceImpl;
    end;
}
```

# 2 Services – 1 Interface

```
codeunit 50143 OnPremServiceImpl implements IService
{
    Access = Internal;

    var
        ServiceUri: Label 'http://localhost:5172/AddRating?rating=%1', Locked = true;
        HttpErr: Label OnPrem: AddRating request failed with error: %1, reason: %2';

    procedure AddRating(ImageStr: Text; Rating: Integer) Result: Text;
    var
        client: HttpClient;
        Content: HttpContent;
        Response: HttpResponseMessage;
    begin
        Content.WriteFrom(ImageStr);
        client.Post(StrSubstNo(ServiceUri, Rating), Content, Response);
        if not Response.IsSuccessStatusCode then
            Error(HttpErr, Response.HttpStatusCode, Response.ReasonPhrase);
        Response.Content.ReadAs(Result);
    end;
}
```



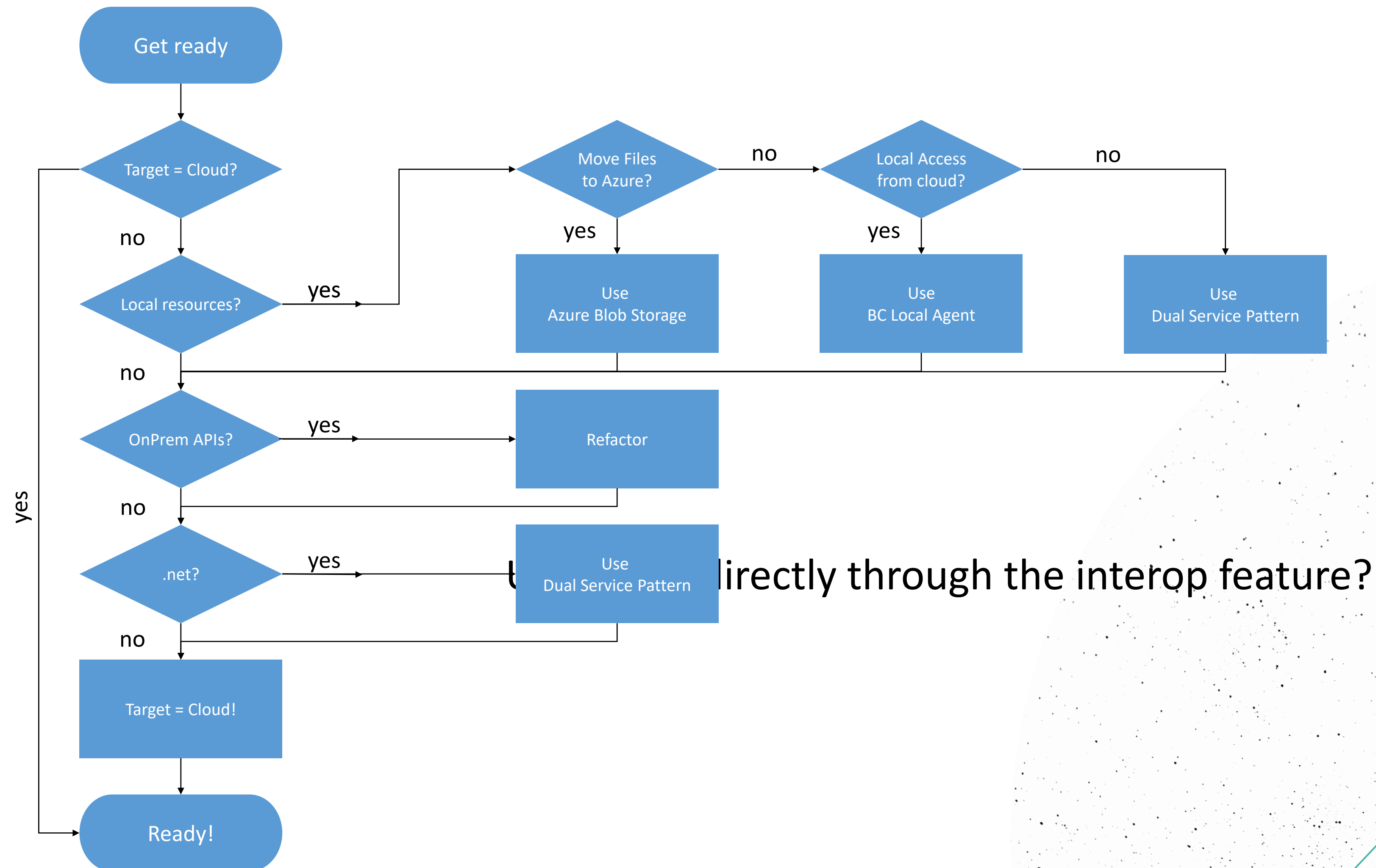
# 2 Services – 1 Interface

```
procedure RateBeer(BeerRec: Record BeerTable; Rating: Integer)
var
    Service: Codeunit Service;
begin
    BeerRec.SetImageFromBase64(Service.Instance().AddRating(BeerRec.GetImageAsBase64(), Rating));
    BeerRec.Modify();
end;
```

# DEMO: The Beer Organizer



# Getting to Universal Code



directly through the interop feature?

# A community of ISV Development Centers is available to accelerate your Universal code success



Alletech has proven to be a reliable partner with a lot of know-how on Business Central and surrounding Microsoft technologies in a cloud environment. Their team of professionals is engaged with helping out in our challenges and are happy to go the extra mile to achieve the best result in our cooperation.

**Benny Dor, Aptean**



We collaborated as one team together with our coach Innova Advanced Consulting who helped us perform automated deployments from Azure Devops, improve agility and quality in developments and migrate our customers from Onprem to Online much faster. Microsoft's ISV Invest. Program has helped us accelerate the digitization of our customers in the sheet metal transformation industry to take them to the next level in their business processes

**David Escudero, Kurago**



From the initial kick-off through the iterative design, development process and lastly, publication in AppSource, the Ciellos team was responsive, professional, and understanding. We could not have done this project at such a rapid pace without their expertise and guidance. Thanks to them, we released our first Business Central product to AppSource more than six months earlier than planned, and we were able to connect with and onboard 50% more users than we had expected at this point. Our experience with Ciellos has exceeded our expectations on every front.

**Tim Kahne, SKGlobal**



Cloud Ready Software has literally shifted our focus from on-prem to SaaS, and then helped us to put our big vertical on AppSource and to align our reseller channel. This past year has been an amazing journey together.

**Miguel Flaminio, Tegos Group**



Building visibility in the online market is as important as building an actual cloud product. This journey starts on the website and goes through marketing, sales, and pricing. Overall, it's about finding the right mindset for effective and efficient customer onboarding. Neural Impact was a huge booster for us in every phase of our project. We have been able to reduce our average customer onboarding effort by over half. To every ISV out there, we strongly recommend taking this journey with Neural Impact.

**Mark Fahrni, SwissSalary**



Simplanova is our long-term partner while doing upgrades from the oldest Dynamics version 2, up to the latest versions of Dynamics 365 Business central. We recommend Simplanova as a reliable and professional partner for doing upgrades together.

**Abeße Hamdi, Groupe Calliope**



Companial helped us to tackle the architecture and design approach with their great experience and some outside the box thinking. They also were very valuable in solving performance issues we saw

**Johan Bakker, 4PS**



Enavate has been a great partner. Their team helped us streamline our efforts in bringing more value to the clients. Enavate's innovative approach during the ISV Acceleration Program gave us a boost and increased our team's efficiency. We are grateful for their dedication to the delivering results and the high quality of their work. We are more than happy to recommend Enavate as a reliable partner to other ISVs

**Khaled Nassra, Binary Stream**



We've been very impressed with the level of involvement from Stratos Cloud Alliance (SCA) as we prepared for product launch. SCA has put together focus groups for us, given us feedback on the product itself, recommendations on bundling, and advice on how to get our product on Microsoft AppSource.

**Andrew Kurtz, Kopis**



# Further reading

PPT Decks, FAQs, Operational guidelines

<http://aka.ms/BCUniversalCode>

Business Central Local Agent sample

<https://github.com/microsoft/BCTech/tree/master/samples/BCAgent>

Moving .net to an external service

<https://github.com/microsoft/BCTech/tree/master/samples/.NETAddin>

Azure Blob Storage in System Application

<https://github.com/microsoft/ALAppExtensions/tree/main/Modules/System/Azure%20Blob%20Services%20API>



**Any Questions?**



Thank  
You!