

# The road from C/AL to AL

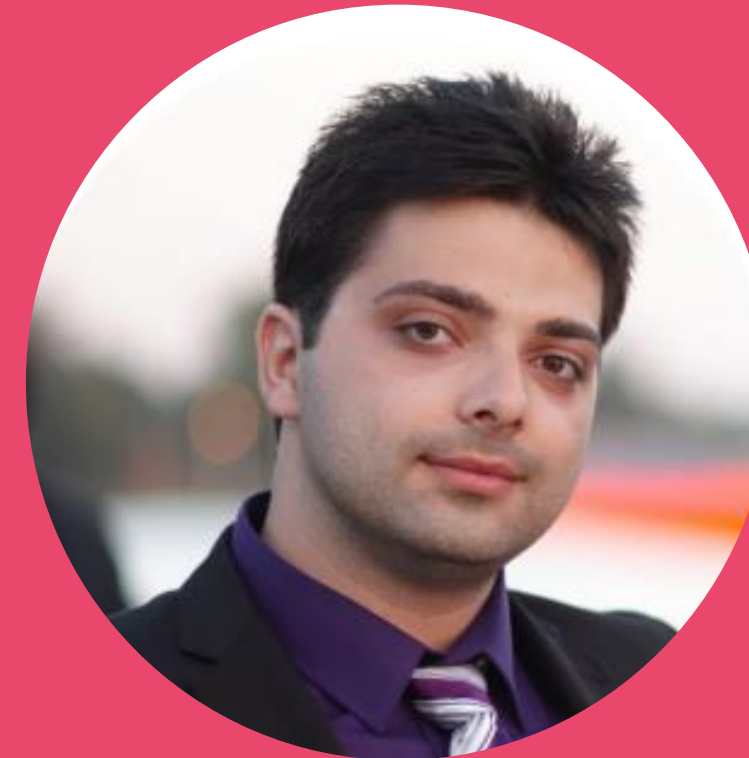
Esben Nyhuus Kristoffersen, Alex Toader, Jesper Schulz-Wedde, Wael AbuSeada  
MICROSOFT DEVELOPMENT CENTER COPENHAGEN

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

# Today's presenters are...



Esben Nyhuus Kristoffersen  
Principal Architect  
Microsoft



Alex Toader  
Senior Engineering Manager  
Microsoft



Wael AbuSeada  
Software Engineer  
Microsoft



Jesper Schulz-Wedde  
Senior Engineering Manager  
Microsoft



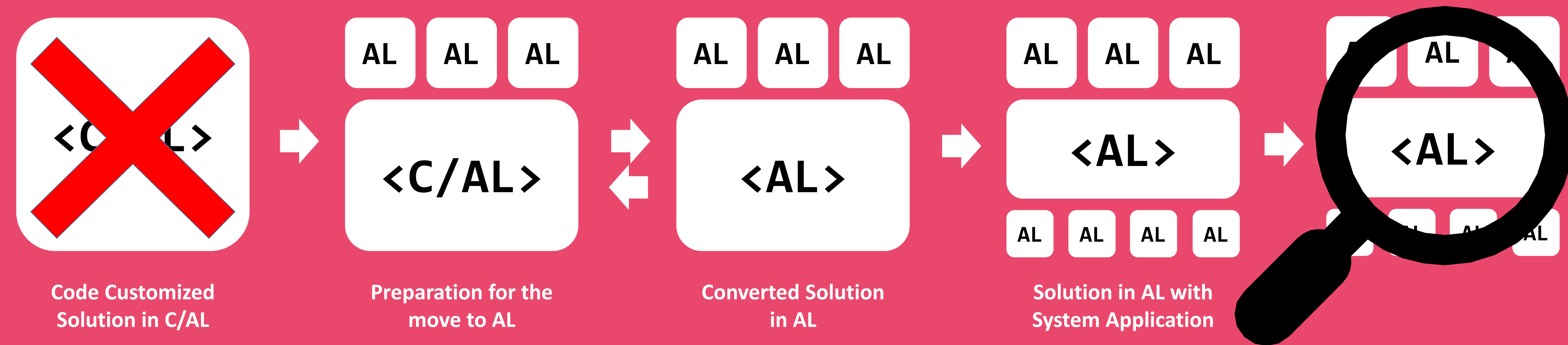
# The Road From C/AL to AL



*Difficult roads often lead to beautiful destinations*



# The Road From C/AL to AL



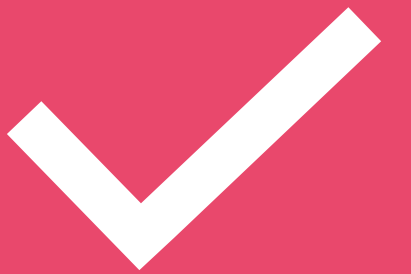
# Related Sessions

## What this session covers:

The move from C/SIDE and C/AL to VSCode and AL

The change of application architecture from a monolith to modules and extensions

The latest additions to AL



## For more info on the actual code conversion and upgrade, please join the related session:

“Using Docker and the ContainerHelper to convert your C/AL solution to an AL solution”

Freddy and Nikola are going to dive into these topics later today at 16:00.

Don't miss it! There will be wales, crocodiles, mountains and even a zeppelin!



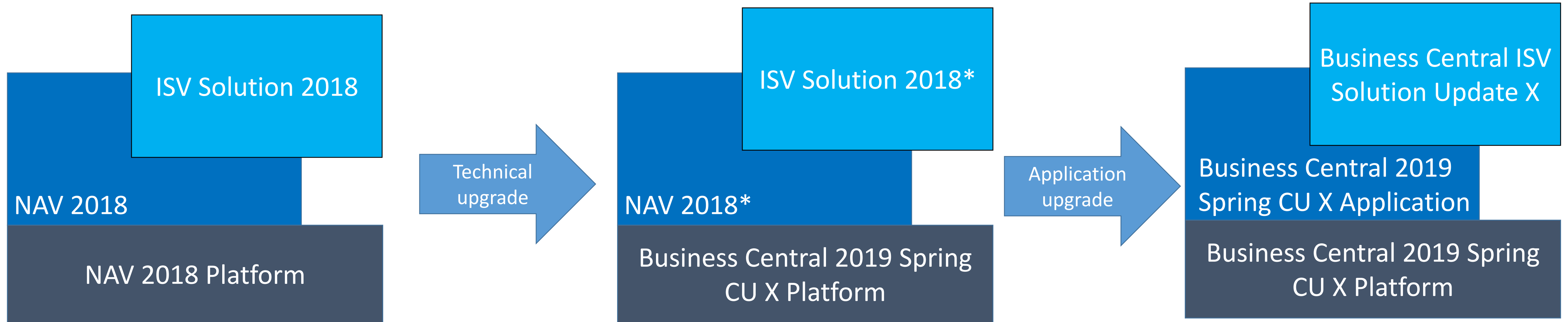
# Agenda

- Convert your solution to AL
- Uptake the System Application
- Secure your apps
- After the move to AL
- Session Wrap Up
- Q&A

# The conversion from C/AL to AL

...for this part we'll relax the pink theme for a while

# Convert to AL – Preparation





# Convert to AL – Prepare translations

- Multi-language properties in C/AL code
  - Will be extracted to translation files automatically
- Non-UTF8 translations
  - Import them in C/SIDE using the standard Tools->Translate->Import command.
  - Compile the project in C/SIDE before converting to AL
- UTF-8 translations
  - Use the finsql CreateLanguage command to generate placeholders

# Convert to AL – Export to new syntax

- finsql.exe command=ExportToNewSyntax
- Export-NavApplicationObject -ExportToNewSyntax

# Convert to AL – Export to new syntax

```
OBJECT Table 3 Payment Terms
{
  OBJECT-PROPERTIES
  {
    Date=11/14/19;
    Time=12:00:00 PM;
    Version List=NAVW114.00;
  }
  PROPERTIES
  {
    DataCaptionFields=Code,Description;
    OnInsert=BEGIN
      SetLastModifiedDateTime;
    END;

    OnModify=BEGIN
      SetLastModifiedDateTime;
    END;

    OnDelete=VAR
      PaymentTermsTranslation@1000 : Record 462;
      0365SalesInitialSetup@1001 : Record 2110;
```

```
OBJECT Table 3 Payment Terms
{
  OBJECT-PROPERTIES
  {
    Date=20191114D;
    Time=120000T;
    Version List=NAVW114.00;
  }
  PROPERTIES
  {
    DataCaptionFields=Code,Description;
    OnInsert=BEGIN
      SetLastModifiedDateTime;
    END;

    OnModify=BEGIN
      SetLastModifiedDateTime;
    END;

    OnDelete=VAR
      PaymentTermsTranslation@1000 : Record "Payment Term Translation";
      0365SalesInitialSetup@1001 : Record "0365 Sales Initial Setup";
```



# Convert to AL – Export to new syntax

<pre>{ 7301;      ;Qty. Picked (Base) ;Decimal      ;CaptionML=ENU=Qty. Picked (Base); DecimalPlaces=0:5; Editable=No }</pre>	←	<pre>{ 7301;      ;Qty. Picked (Base) ;Decimal      ;CaptionML=ENU=Qty. Picked (Base); DecimalPlaces=0:5; Editable=false }</pre>
<pre>{ 7302;      ;Completely Picked ;Boolean      ;CaptionML=ENU=Completely Picked; Editable=No }</pre>	←	<pre>{ 7302;      ;Completely Picked ;Boolean      ;CaptionML=ENU=Completely Picked; Editable=false }</pre>
<pre>{ 7303;      ;Pick Qty. (Base) ;Decimal      ;CaptionML=ENU=Pick Qty. (Base); DecimalPlaces=0:5; Editable=No }</pre>	←	<pre>{ 7303;      ;Pick Qty. (Base) ;Decimal      ;CaptionML=ENU=Pick Qty. (Base); DecimalPlaces=0:5; Editable=false }</pre>

# Convert to AL – Export to new syntax

```
IF IdentityManagement.IsInvAppId THEN
  IF 0365SalesInitialSetup.GET AND
    (0365SalesInitialSetup."Default Payment Terms Code" = Code)
  THEN
    ERROR(CannotRemoveDefaultPaymentTermsErr);

  WITH PaymentTermsTranslation DO BEGIN
    SETRANGE("Payment Term",Code);
    DELETEALL
  END;
```

```
if IdentityManagement.IsInvAppId then
  if 0365SalesInitialSetup.Get and
    (0365SalesInitialSetup."Default Payment Terms Code" = Code)
  then
    Error(CannotRemoveDefaultPaymentTermsErr);

  with PaymentTermsTranslation do begin
    SetRange("Payment Term",Code);
    DeleteAll
  end;
```

# Convert to AL – Export to new syntax

```
    ODataKeyFields=Id;
}
CONTROLS
{
    { 1 ;0 ;Container ;
//////////////////////////////////////////////////
        ContainerType=ContentArea }

    { 2 ;1 ;Group ;
        Name=Group;
        GroupType=Repeater }

    { 3 ;2 ;Field ;
        Name=number;
        CaptionML=[@@@={Locked};
            ENU=No.];
        ApplicationArea=#All;
        SourceExpr="No." }
//////////////////////////////////////////////////

    { 4 ;2 ;Field ;
        Name=name;
        CaptionML=[@@@={Locked};
            ENU=Name];
        ApplicationArea=#All;
        SourceExpr=Name }
//////////////////////////////////////////////////

    { 5 ;2 ;Field ;
        Name=searchName;
        CaptionML=[@@@={Locked};
            ENU=Search Name];
        ApplicationArea=#All;
        SourceExpr="Search Name" }
//////////////////////////////////////////////////

    { 6 ;2 ;Field ;
        Name=name2;
        CaptionML=[@@@={Locked};
            ENU=Name 2];
```

```
    ODataKeyFields=Id;
}
CONTROLS
{
    { 1 ;0 ;Container ;
        Name=Control1;
        ContainerType=ContentArea }

    { 2 ;1 ;Group ;
        Name=Group;
        GroupType=Repeater }

    { 3 ;2 ;Field ;
        Name=number;
        CaptionML=[@@@={Locked};
            ENU=No.];
        ApplicationArea=#All;
        SourceExpr="No.";
        ImplicitType=Code20 }

    { 4 ;2 ;Field ;
        Name=name;
        CaptionML=[@@@={Locked};
            ENU=Name];
        ApplicationArea=#All;
        SourceExpr=Name;
        ImplicitType=Text50 }

    { 5 ;2 ;Field ;
        Name=searchName;
        CaptionML=[@@@={Locked};
            ENU=Search Name];
        ApplicationArea=#All;
        SourceExpr="Search Name";
        ImplicitType=Code50 }

    { 6 ;2 ;Field ;
        Name=name2;
        CaptionML=[@@@={Locked};
            ENU=Name 2];
```



# Convert to AL – Export to new syntax

```
}  
}  
CONTROLS  
{  
  { 1900000001;0;Container;  
    ContainerType=ContentArea }  
  
  { 1 ;1 ;Group ;  
    GroupType=Repeater }  
  
  { 4 ;2 ;Field ;  
    ToolTipML=ENU=Specifies whether the item in the item ledger entry is  
    ApplicationArea=#ItemTracking;  
    SourceExpr=Positive }  
  
  { 6 ;2 ;Field ;  
    ToolTipML=ENU=Specifies the document number on the entry. The document is the  
    ApplicationArea=#ItemTracking;  
    SourceExpr="Document No." }  
  
  { 8 ;2 ;Field ;  
    ToolTipML=ENU=Specifies the number of the item in the entry.;  
    ApplicationArea=#ItemTracking;  
    SourceExpr="Item No." }  
  
  { 10 ;2 ;Field ;  
    ToolTipML=ENU=Specifies the variant of the item on the line.;  
    ApplicationArea=#Planning;  
    SourceExpr="Variant Code";  
    Visible=FALSE }
```

```
}  
}  
CONTROLS  
{  
  { 1900000001;0;Container;  
    Name=Control1900000001;  
    ContainerType=ContentArea }  
  
  { 1 ;1 ;Group ;  
    Name=Control1;  
    GroupType=Repeater;  
    ShowCaption=false }  
  
  { 4 ;2 ;Field ;  
    Name=Positive;  
    ToolTipML=ENU=Specifies whether the item in the item ledger entry is positive  
    ApplicationArea=#ItemTracking;  
    SourceExpr=Positive;  
    ImplicitType=Boolean }  
  
  { 6 ;2 ;Field ;  
    Name="Document No.";  
    ToolTipML=ENU=Specifies the document number on the entry. The document is the  
    ApplicationArea=#ItemTracking;  
    SourceExpr="Document No.";  
    ImplicitType=Code20 }  
  
  { 8 ;2 ;Field ;  
    Name="Item No.";  
    ToolTipML=ENU=Specifies the number of the item in the entry.;  
    ApplicationArea=#ItemTracking;  
    SourceExpr="Item No.";  
    ImplicitType=Code20 }  
  
  { 10 ;2 ;Field ;  
    Name="Variant Code";  
    ToolTipML=ENU=Specifies the variant of the item on the line.;  
    ApplicationArea=#Planning;  
    SourceExpr="Variant Code";  
    Visible=FALSE;
```

# Convert to AL – Export to new syntax

```
{ 1102601007;2 ;Action      ;
//
CaptionML=ENU=Service Dis&counts;
ToolTipML=ENU=View or edit the discounts that you grant for
ApplicationArea=#Service;
RunObject=Page 6058;
RunPageLink=Contract Type=FIELD(Contract Type),
Contract No.=FIELD(Contract No.);
Image=Discount }
{ 1102601008;2 ;Action      ;
//
CaptionML=ENU=Service &Hours;
ToolTipML=ENU=View the service hours that are valid for the
ApplicationArea=#Service;
RunObject=Page 5916;
RunPageLink=Service Contract No.=FIELD(Contract No.),
Service Contract Type=FILTER(Contract);
Image=ServiceHours }
{ 1102601010;2 ;Action      ;
//
CaptionML=ENU=Co&mments;
ToolTipML=ENU=View or add comments for the record.;
ApplicationArea=#Service;
RunObject=Page 5911;
RunPageLink=Table Name=CONST(Service Contract),
Table Subtype=FIELD(Contract Type),
No.=FIELD(Contract No.),
Table Line No.=CONST(0);
```

```
{ 1102601007;2 ;Action      ;
Name="Service Dis&counts";
CaptionML=ENU=Service Dis&counts;
ToolTipML=ENU=View or edit the discounts that you grant for the contra
ApplicationArea=#Service;
RunObject=Page "Contract/Service Discounts";
RunPageLink="Contract Type"=FIELD("Contract Type"),
"Contract No."=FIELD("Contract No.");
Image=Discount }
{ 1102601008;2 ;Action      ;
Name="Service &Hours";
CaptionML=ENU=Service &Hours;
ToolTipML=ENU=View the service hours that are valid for the service co
ApplicationArea=#Service;
RunObject=Page "Service Hours";
RunPageLink="Service Contract No."=FIELD("Contract No."),
"Service Contract Type"=FILTER(Contract);
Image=ServiceHours }
{ 1102601010;2 ;Action      ;
Name="Co&mments";
CaptionML=ENU=Co&mments;
ToolTipML=ENU=View or add comments for the record.;
ApplicationArea=#Service;
RunObject=Page "Service Comment Sheet";
RunPageLink="Table Name"=CONST("Service Contract"),
"Table Subtype"=FIELD("Contract Type"),
"No."=FIELD("Contract No."),
"Table Line No."=CONST(0);
```

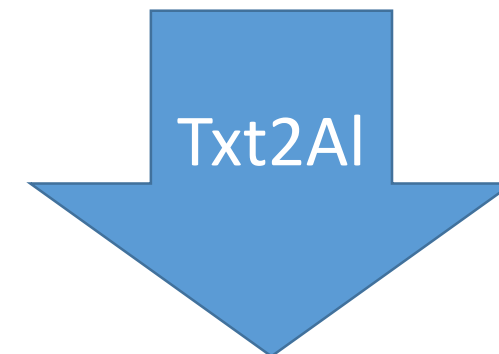
# Convert to AL – Txt2Al

```
Txt2Al.exe --source="C:\Temp\NewSyntaxFolder"  
           --target="C:\Temp\ALFolder"  
           --runtime="4.0"
```



# Convert to AL – Txt2AI

XMLDoc@1001 : DotNet "System.Xml, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089".System.Xml.XmlDocument"



```
dotnet
{
    assembly(System.Xml)
    {
        Version = '4.0.0.0';
        Culture = 'neutral';
        PublicKeyToken = 'b77a5c561934e089';

        type("System.Xml.XmlDocument"; XmlDocument) { }
    }
}
```

```
codeunit 50100 MyCodeunit
{
    var
        XMLDoc: DotNet XmlDocument;
}
```

# Convert to AL – Txt2Al

```
Txt2Al.exe --source="C:\Temp\NewSyntaxFolder"  
           --target="C:\Temp\ALFolder"  
           --runtime="4.0"  
           --dotNetTypePrefix="My_"
```

# Convert to AL – Txt2AI

XMLDoc@1001 : DotNet "System.Xml, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089".System.Xml.XmlDocument"

Txt2AI

dotnet

```
{
  assembly(System.Xml)
  {
    Version = '4.0.0.0';
    Culture = 'neutral';
    PublicKeyToken = 'b77a5c561934e089';

    type("System.Xml.XmlDocument"; My_XmlDocument) { }
  }
}
```

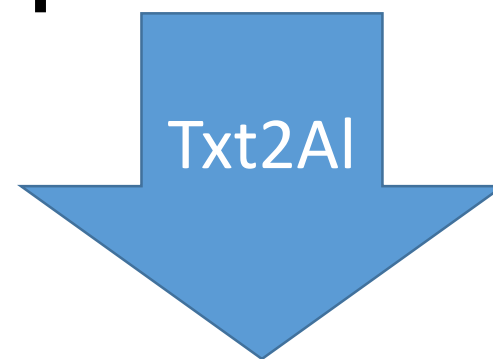
codeunit 50100 MyCodeunit

```
{
  var
    XMLDoc: DotNet My_XmlDocument;
}
```



# Convert to AL – Txt2Al

```
{ 3 ; 2 ;Field ;  
    Name=Value;  
    CaptionML=ENU=Amount;  
    ToolTipML=ENU=Specifies the summarized amount;  
}
```



```
field(Value; Value)  
{  
    Caption = 'Amount';  
    ToolTip = 'Specifies the summarized...';  
}
```

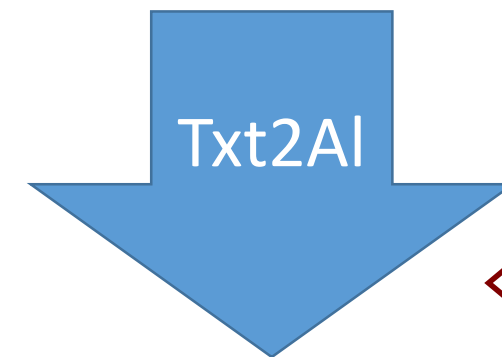
```
<trans-unit id="Page 1455895736 - Control 1131619133 -  
    Property 2879900210" size-  
unit="char" translate="yes" xml:space="preserve">  
    <source>Amount</source>  
    <target>Amount</target>  
    <note from="Developer" annotates="general" priority="2"></  
note>  
    <note from="Xliff Generator" annotates="general" priority=  
"3">Page 0365 Sales Year Summary - Control Value -  
    Property Caption</note>  
</trans-unit>
```

# Convert to AL – Txt2Al

```
Txt2Al.exe --source="C:\Temp\NewSyntaxFolder"  
           --target="C:\Temp\ALFolder"  
           --runtime="4.0"  
           --dotNetTypePrefix="My_"  
           --addLegacyTranslationInfo
```

# Convert to AL – Txt2AI

```
{ 3 ; 2 ;Field ;  
    Name=Value;  
    CaptionML=ENU=Amount;  
    ToolTipML=ENU=Specifies the summarized amount;  
}
```

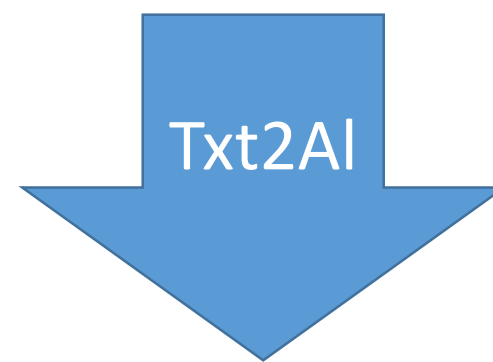


```
field(Value; Value)  
{  
    Caption = 'Amount';  
    ToolTip = 'Specifies the summarized amount';  
}
```

```
<trans-unit id="Page 1455895736 - Control 1131619133 -  
    Property 2879900210" size-  
unit="char" translate="yes" xml:space="preserve">  
    <source>Amount</source>  
    <target>Amount</target>  
    <note from="Developer" annotates="general" priority="2"></  
note>  
    <note from="Xliff Generator" annotates="general" priority=  
"3">Page 0365 Sales Year Summary - Control Value -  
    Property Caption</note>  
    <note from="Transitional" annotates="general" priority="4"  
>N2100-C3-P8629</note>  
</trans-unit>
```

# Convert to AL – Txt2AI

```
{ 2 ;1 ;Field ;  
    Name=WebPageViewer;  
    ApplicationArea=#Basic,#Suite;  
    ControlAddIn=[Microsoft.Dynamics.Nav.Client.WebPageViewer;PublicKey  
Token=31bf3856ad364e35]}
```



```
usercontrol(WebPageViewer;"Microsoft.Dynamics.Nav.Client.WebPageViewer")  
{  
    ApplicationArea = Basic, Suite;  
}
```

# Convert to AL – Txt2Al

```
Txt2Al.exe --source="C:\Temp\NewSyntaxFolder"  
--target="C:\Temp\ALFolder"  
--runtime="4.0"  
--dotNetTypePrefix="My_"  
--addLegacyTranslationInfo  
--dotNetAddInsPackage="C:\Temp\MyCustomAddIns.al"
```



# Convert to AL – Txt2Al

dotnet

```
{
    assembly("Microsoft.Dynamics.Nav.Client.WebPageViewer")
    {
        type("Microsoft.Dynamics.Nav.Client.WebPageViewer.IWebP
ageViewer"; "Microsoft.Dynamics.Nav.Client.WebPageViewer")
        {
            IsControlAddIn = true;
        }
    }
}
```

# Convert to AL – Compiling your AL project

- Create an app.json and add it to the root of your project
- Configure the “al.packageCachePath” to point to a directory containing the System.app
- Configure your “al.assemblyProbingPaths” to point to a set of directories containing referenced .NET assemblies
- Compile

# Convert to AL – Compiling your AL project

```
AccountantActivities.Page.al
AccountantActivities.Page.al > Page 9037 "Accountant Activities" > layout > {} Area content > {} CueGroup "Product Videos" > Action Action32
241 begin
242     UserTaskList.SetPageToShowMyPendingUserTasks;
243     UserTaskList.Run;
244 end;
245 }
246 }
247 cuegroup("Product Videos")
248 {
249     Caption = 'Product Videos';
250     Visible = ShowProductVideosActivities;
251
252     actions
253     {
254         action(Action32)
255         {
256             ApplicationArea = Basic, Suite;
257             Caption = 'Product Videos';
258             Image = TileVideo;
259             RunObject = Page "Product Videos";
260             Tooltip = 'Open a list of videos that showcase some of the product capabilities.';
261         }
262     }
263 }
264 cuegroup("Get started")
265 {
266     Caption = 'Get started';
267 }
```

PROBLEMS 1K+ OUTPUT DEBUG CONSOLE TERMINAL

Filter. E.g.: text, \*\*/\*.ts, !\*\*/node\_modules/\*\*

- AccountantRoleCenter.Page.al 2
  - 'ToolTip' is being deprecated in the versions: '5.0' or greater. This property does not have any effect on parts. This warning will become an error in a future release. AL(AL0667) [40, 17]
  - 'ToolTip' is being deprecated in the versions: '5.0' or greater. This property does not have any effect on parts. This warning will become an error in a future release. AL(AL0667) [45, 17]
- AccountEntity.Page.al 2
  - Field 'Id' is marked for removal. Reason: This functionality will be replaced by the systemID field AL(AL0432) [12, 22]
  - Field 'Id' is marked for removal. Reason: This functionality will be replaced by the systemID field AL(AL0432) [23, 27]
- ApplicationAreaMgmt.Codeunit.al 7
  - Field 'Invoicing' is marked for removal. Reason: Microsoft Invoicing is not supported on Business Central AL(AL0432) [366, 80]
  - Field 'Invoicing' is marked for removal. Reason: Microsoft Invoicing is not supported on Business Central AL(AL0432) [661, 36]
  - Field 'Invoicing' is marked for removal. Reason: Microsoft Invoicing is not supported on Business Central AL(AL0432) [662, 73]
  - Field 'Invoicing' is marked for removal. Reason: Microsoft Invoicing is not supported on Business Central AL(AL0432) [748, 33]
  - Field 'Invoicing' is marked for removal. Reason: Microsoft Invoicing is not supported on Business Central AL(AL0432) [751, 46]
  - Field 'Invoicing' is marked for removal. Reason: Microsoft Invoicing is not supported on Business Central AL(AL0432) [936, 37]
  - Field 'Invoicing' is marked for removal. Reason: Microsoft Invoicing is not supported on Business Central AL(AL0432) [937, 39]
- AssembletoOrderLink.Table.al 1
  - Method 'CreateReservEntryFor' is marked for removal. Reason: AL(AL0432) [587, 35]
- AssemblyPost.Codeunit.al 1
  - Method 'CreateReservEntryFor' is marked for removal. Reason: AL(AL0432) [1296, 43]
- AssistedCompanySetup.Codeunit.al 1

# Convert to AL – Compiling your AL project

```
alc.exe -project:"C:\Temp\MyProject"  
-out:"C:\Temp\MyApp.app"  
-generatereportlayout-  
-packageCachePath:"C:\Temp\ReferencePackages"  
-assemblyprobingpaths:...  
-features:translationfile  
-parallel  
-errorLog:"C:\Temp\errors.json"
```

# Convert to AL –Why am I getting errors?

C/SIDE was lenient → Sloppy code

AL is more strict → Better code



# Convert to AL – Why am I getting errors?

## Duplicate case-lines

```
805     CASE XMLSchemaElement."Node Type" OF
806         XMLSchemaElement."Node Type"::Element:
807             IF (SchemaPrefix <> '') AND (PrefixLength = 0) THEN
808                 XMLSchemaElement."Node Name" := STRSUBSTNO('%1:%2',SchemaPrefix,XMLSchemaElement."Node Name");
809                 XMLSchemaElement."Node Type"::Element:
810                     IF PrefixLength > 0 THEN
811                         XMLSchemaElement."Node Name" := COPYSTR(XMLSchemaElement."Node Name",PrefixLength + 1);
812             END;
813     END;
```

# Convert to AL – Why am I getting errors?

## Non-existing objects

```
67      }  
68      { 8      ;2      ;Action      ;  
        CaptionML=ENU=Code Coverage;  
        ApplicationArea=#All;  
        RunObject=Page 130002;  
        Promoted=Yes;  
        PromotedCategory=Report }  
68      }
```

# Convert to AL – Why am I getting errors?

## Permission on non-existing objects

```
11 Permissions=TableData 21=rm,  
12     TableData 25=rm,  
13     TableData 81=rim,  
14     TableData 356=ri;  
15     TableData 81=rim;
```

# Convert to AL –Why am I getting errors?

## .NET type-checking at compile-time

```
1404 PROCEDURE InitializeProxyVersionList@40(VAR TempStack@1000 : TEMPORARY Record 9160);
1405 VAR
1406     CRMHelper@1005 : DotNet "'Microsoft.Dynamics.Nav.CrmCustomizationHelper, Version=13.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35'.Mi
1407     IList@1004 : DotNet "'mscorlib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089'.System.Collections.IList";
1408     IList@1004 : DotNet "'mscorlib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089'.System.Collections.Generic.List`1";
1409     i@1003 : Integer;
1410     ProxyCount@1002 : Integer;
1411 BEGIN
1412     IList := CRMHelper.GetProxyIdList;
1413     ProxyCount := IList.Count;
```

# Convert to AL – Why am I getting errors?

## Symbols are evaluated from innermost to outermost

```
200 PROCEDURE Evaluate@16(Model@1002 : Text;VAR Quality@1001 : Decimal);
201 VAR
202     OutputValue@1003 : Text;
203     CallAzureEndPoint@1000 : Boolean;
204 BEGIN
205     CallAzureEndPoint := TRUE;
206     OnBeforeEvaluate(Model,Quality,RecordVar,CallAzureEndPoint);
207     IF NOT CallAzureEndPoint THEN
208         EXIT;
209     AzureMLConnector.Initialize(ApiKey,ApiUri,ApiTimeout);
210     TestInitialized;
211     AzureMLConnector.AddParameter('method','evaluate');
212     AzureMLConnector.AddParameter('model',Model);
213     CreateInput;
214     IF NOT AzureMLConnector.SendToAzureML(TRUE) THEN // hard coded to TRUE so that ML does update the usage stats
215         ERROR(SomethingWentWrongErr,GetLastDetailedError);
216
217     AzureMLConnector.GetOutput(1,1,OutputValue);
218     EVALUATE(Quality,OutputValue,9);
219     SYSTEM.EVALUATE(Quality,OutputValue,9);
219 END;
```



# Convert to AL –Why am I getting errors?

## Events from variables previously declared with WITHEVENTS

```
247 VAR
248     NoReportsAvailableErr@1011 : TextConst 'ENU=There are no reports available from Power BI.';
249     PowerBIUserConfiguration@1018 : Record 6304;
250     IntelligentCloud@1035 : Record 2000000146;
251     TempPowerBiReportBuffer@1015 : TEMPORARY Record 6302;
252     PowerBiServiceMgt@1013 : Codeunit 6301;
253     AzureAdMgt@1012 : Codeunit 6300;
254     ConfPersonalizationMgt@1014 : Codeunit 9170;
255     SetPowerBIUserConfig@1017 : Codeunit 6305;
256     ClientTypeManagement@1077 : Codeunit 4;
257     JObject@1019 : DotNet "'Newtonsoft.Json'.Newtonsoft.Json.Linq.JObject";
258     JObjecttemp@1020 : DotNet "'Newtonsoft.Json'.Newtonsoft.Json.Linq.JObject";
```

677

```
EVENT JObject@1019::PropertyChanged@117(sender@1001 : Variant;e@1000 : DotNet "'System, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c56
BEGIN
END;

EVENT JObject@1019::PropertyChanging@118(sender@1001 : Variant;e@1000 : DotNet "'System, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c5
BEGIN
END;
```



# Convert to AL – Why am I getting errors?

## Leftover properties

```
1  OBJECT Page 6510 Item Tracking Lines
2  {
3    OBJECT-PROPERTIES
4    {
5      Date=;
6      Time=;
7      Version List=;
581
582      { 61 ;2 ;Field ;
583          Name=AvailabilitySerialNo;
584          CaptionML=ENU=Availability, Serial No.;
585          TooltipML=ENU=Specifies a warning icon if the sum of the quantities of the item in outbound documents is greater than the serial number;
586          OptionCaptionML=ENU=Bitmap45;
587          ApplicationArea=#ItemTracking;
588          SourceExpr=TrackingAvailable(Rec,0);
589          Editable=False;
590          OnDrillDown=BEGIN
591              LookupAvailable(0);
592          END;
593
2357
2358  LOCAL PROCEDURE TrackingAvailable@45(VAR TrackingSpecification@1000 : Record 336;LookupMode@1001 : 'Serial No.,Lot No.,CD No.') : Boolean;
2359  BEGIN
2360      EXIT(ItemTrackingDataCollection.TrackingAvailable(TrackingSpecification,LookupMode));
2361  END;
2362
```

# Convert to AL – Why am I getting errors?

## Leftover properties

```
1 OBJECT Report 5875 Phys. Invt. Order Diff. List
2 {
3     OBJECT-PROPERTIES
4     {
5         Date=;
6         Time=;
```

```
220
221     { 105 ;3 ;Column ;Text1000_Control105 ;
222         DecimalPlaces=0:5;
223         SourceExpr=Text1000Lb1 }
224     { 106 ;3 ;Column ;Text1000_Control106 ;
225         DecimalPlaces=0:5;
226         SourceExpr=Text1000Lb1 }
```

# Convert to AL –Why am I getting errors?

## Subscriber signature must match Publisher signature

```
32
33     [EventSubscriber(Codeunit,1660,OnImportPayroll)]
34     [External]
35     [PROCEDURE ImportPayrollTransactions@19(VAR TempServiceConnection@1000 : TEMPORARY Record 1400;VAR GenJournalLine@1001 : Record 81);]
36     PROCEDURE ImportPayrollTransactions@19(VAR TempServiceConnection@1000 : TEMPORARY Record 1400;GenJournalLine@1001 : Record 81);
37     BEGIN
38         TempNewGenJournalLine.RESET;
39         IF TempNewGenJournalLine.FINDSET THEN
```

```
633
634     [HyperlinkHandler]
635     [PROCEDURE Action62HyperlinkHandler@93(Message@1000 : Text[1000]);]
636     PROCEDURE Action62HyperlinkHandler@93(Message@1000 : Text[1024]);
637     BEGIN
638     END;
639     BEGIN
640     END.
```

# Convert to AL – Why am I getting errors?

## DataCaptionExpr must be Text

```
15      ModifyAllowed=No;  
16      SourceTable=Table823;  
17      DataCaptionExpr=DATE2DMY(WORKDATE,3);  
17      DataCaptionExpr=FORMAT(DATE2DMY(WORKDATE,3));  
18      PageType=CardPart;  
19      SourceTableTemporary=Yes;  
20      RefreshOnActivate=Yes;  
21      OnInit=BEGIN  
22          MonthlyDataVisible := FALSE;
```

# Convert to AL – Why am I getting errors?

## Test attributes are only allowed in Test codeunits

```
1 OBJECT Codeunit 8800 Custom Layout Reporting
2 {
3     OBJECT-PROPERTIES
4     {
5         Date=;
6         Time=;
```

```
868 [Normal]
869 PROCEDURE SetTestMode@35(TestMode@1000 : Boolean);
870 VAR
871     TempBlob@1002 : Record 99008535;
872     BlobOutStream@1001 : OutStream;
873 BEGIN
874     // Sets test mode for this codeunit - enables unit test scenarios
875     TempBlob.INIT;
876     TempBlob.RESET;
877     TempBlob."Primary Key" := CODEUNIT::"Custom Layout Reporting";
878     IF TempBlob.FIND THEN
879         TempBlob.DELETE;
```



# Convert to AL – Why am I getting errors?

## Invalid attribute / Missing handler

```
380 [HandlerFunctions(Hnd1Message)]
381 [External]
382 PROCEDURE InitAssemblyBackwayScenario@65(VAR TimeSheetHeader@1016 : Record 950;VAR AssemblyHeader@1015 : Record 900;VAR AssemblyLine@1009 : Recor
383 VAR
384     Location@1010 : Record 14;
385     UserSetup@1011 : Record 91;
```



# Convert to AL – Why am I getting errors?

## Name collisions

```
1  OBJECT Report 11304 Financial Ledger
2  {
3    OBJECT-PROPERTIES
4    {
5      Date=;
6      Time=;

74
75      { 49 ;3 ;Column ;GenJnlTemplateName ;
76        SourceExpr=Text11301 + "Gen. Journal Template".Name }
77

194 { 194 ;3 ;Column ;GenJnlTemplateNameFormat;
195 { 194 ;3 ;Column ;GenJnlTemplateNameFmt;
195      SourceExpr=Text11301 + FORMAT("Journal Template Name") }
```

# Convert to AL – Why am I getting errors?

"<option>"::"<number>" is no longer supported.

```
255      CASE "Account Type" OF
256      [ "Account Type"::"0":
257        0:
258          BEGIN
259            GLAcc."No." := "Account No.";
260            PAGE.RUN(PAGE::"G/L Account Card",GLAcc);
261          END;
```

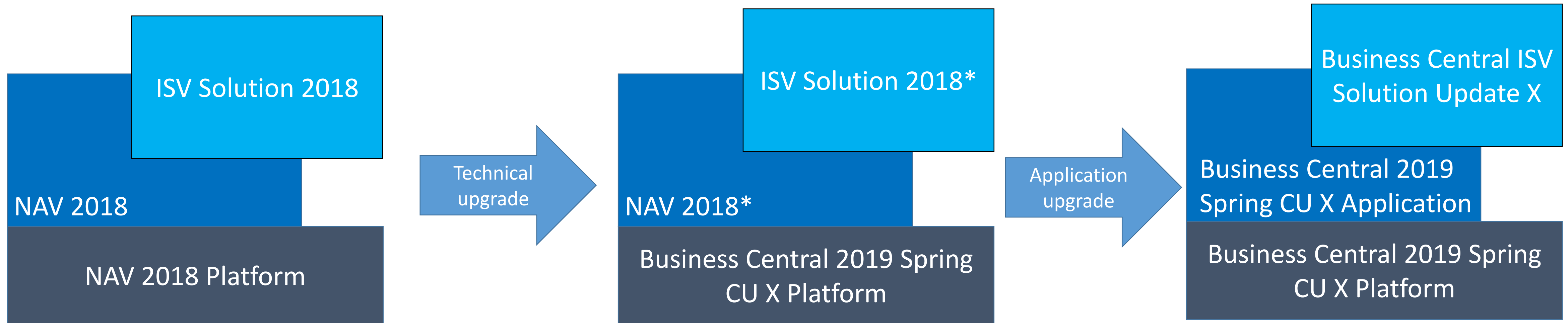
# Convert to AL –Why am I getting errors?

## Auto-generated label caption may be too long

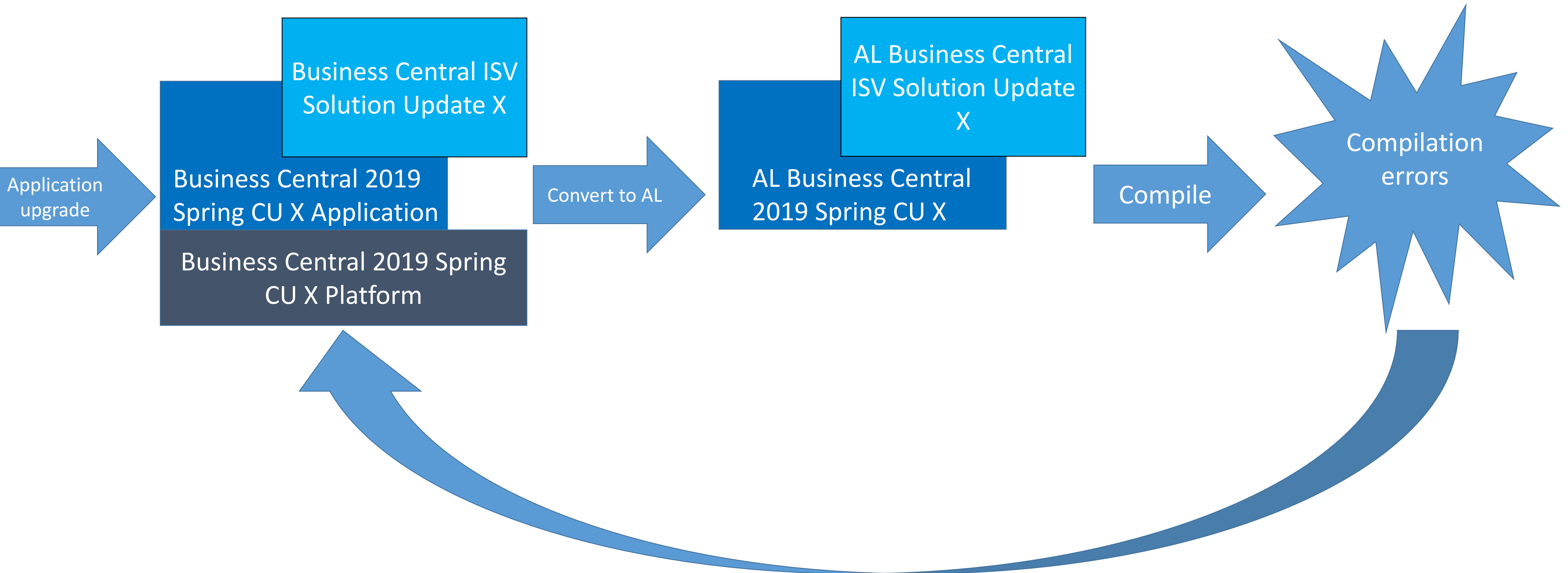
```
199 { 1500048;1;Column ;V1_purpose_of_payment_eg_rental__audit__lawyer__knowledge_of_doctor__building_expenses__prize__discount_or_any_other_benefit
200 [ SourceExpr=V1_purpose_of_payment_eg_rental__audit__lawyer__knowledge_of_doctor__building_expenses__prize__discount_or_any_other_benefit
201 [ SourceExpr=V1_PurposeOfPaymentLbl ] }
```

```
288 Total_income_and_tax_paying__including_with_Phor_Ngor_Dor_3_if_any_CaptionLbl@1504689 : TextConst 'ENU=Total income and tax paying (including w
289 [ V1_purpose_of_payment_eg_rental__audit__lawyer__knowledge_of_doctor__building_expenses__prize__discount_or_any_other_benefit_Lbl@1505692 : Text
289 [ V1_PurposeOfPaymentLbl@1505692 : TextConst 'ENU=1 indicate the purpose of payment eg rental, audit, lawyer, knowledge of doctor, building expen
290 V2_indicate_the_number_accordinglyCaptionLbl@1509114 : TextConst 'ENU=2 indicate the number accordingly';
```

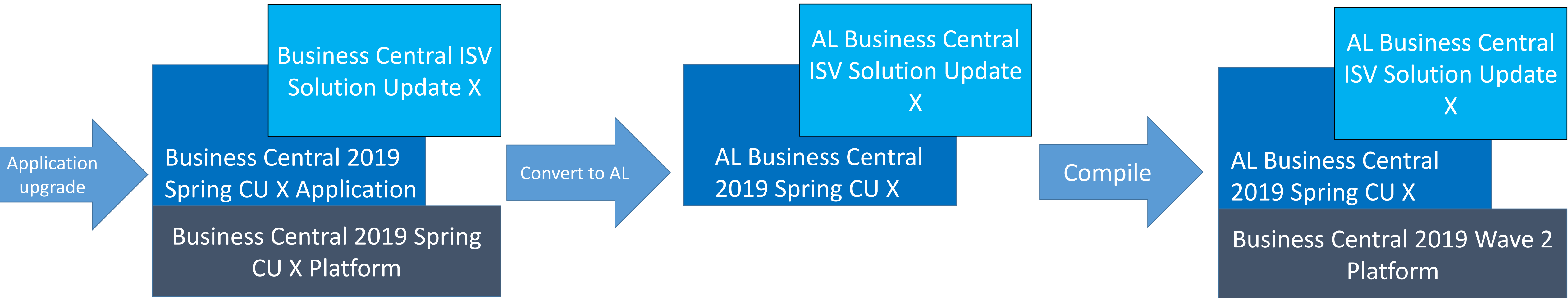
# Convert to AL - Compiling your AL project



# Convert to AL - Compiling your AL project



# Convert to AL - Compiling your AL project





# Convert to AL – Convert translations

- Multi-language properties in C/AL code ✓
- Non-UTF8 translations ✓
- UTF-8 translations
  - Download  
<https://github.com/microsoft/AL/blob/master/scripts/Txt2Xliff.ps1>
  - Call it on the generated XLIFF files and your translation TXT files ✓

# Convert to AL – Convert profiles

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Profiles>
  <Profile ID="SALES MANAGER" Description="Sales Manager" RoleCenterID="900
5" DefaultRoleCenter="false">
    <ProfileMetadata ProfileID="SALES MANAGER" PageID="25" PersonalizationI
D="0806C9F3-14A7-4377-975A-DA1392632923">
      <PageMetadata>&lt;?xml version="1.0"?&gt;&lt;delta xmlns:xsd="http://
www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"&gt;
        &lt;changes&gt;
          &lt;update id="{00000019-0000-0001-0008-
0000836BD2D2}" uri="" name="FreezeColumnID" value="-1" /&gt;
        &lt;/changes&gt;
      &lt;/delta&gt;</PageMetadata>
    </ProfileMetadata>
  </Profiles>
```

# Convert to AL – Convert profiles

```
profile "SALES MANAGER"  
{  
    RoleCenter = 9005;  
    Description = 'Developer comment';  
    Caption = 'Sales Manager';  
    ProfileDescription = 'Functionality for managers...';  
    Enabled = true;  
    Promoted = true;  
}
```







# Optimize Visual Studio Code performance

**A large base app, Visual Studio Code, multiple containers and compile/deploy require a lot of resources!**

Recommend minimum 32GB memory and remote Docker containers

## Turn off Code Analyzers, Code Lens, and Code Actions

Fine to use on for smaller projects, but consumes a lot of memory for large projects like current monolithic AL base application

## Use new Incremental Build (`al.incrementalBuild`)

Reuses cached compilations from IntelliSense etc in Visual Studio Code

## Use RAD

Partial deploy with Alt+Ctrl+F5 and Alt+F5. Require a full deploy first.

```
{ settings.json x
1  {
2    "al.enableCodeAnalysis": false,
3    "editor.codeLens": false,
4    "al.enableCodeActions": false,
5    "al.incrementalBuild": true,
6    "al.packageCachePath": ".alpackages",
7    "al.assemblyProbingPaths": [
8      ".netpackages",
9      "C:\\ProgramData\\NavContainerHelp
10 ]
11 }
```

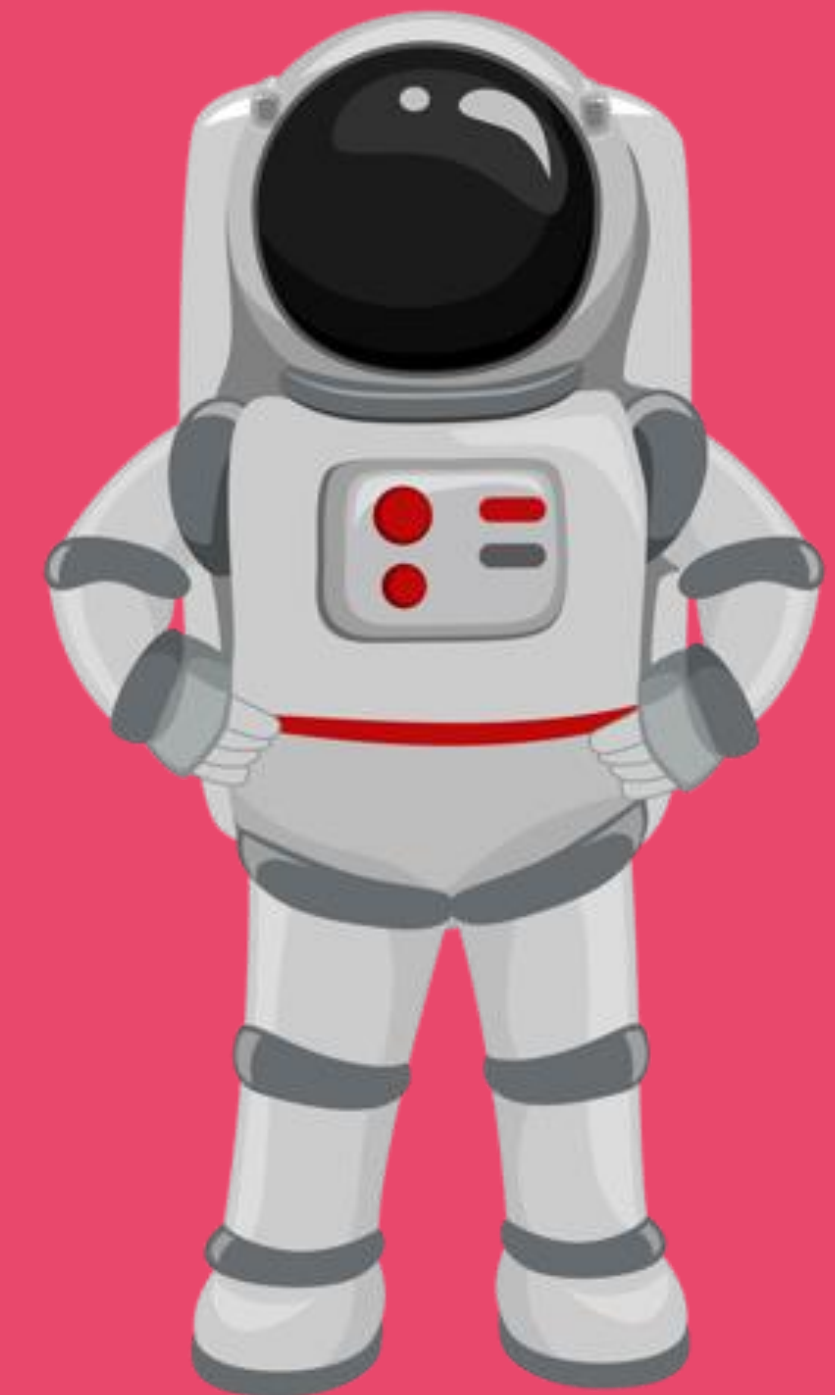
# Uptaking the System Application



# Uptake of System Application

That's one small step for a developer...

```
"dependencies": [  
  {  
    "id": "63ca2fa4-4f03-4f2b-a480-172fef340d3f",  
    "name": "System Application",  
    "publisher": "Microsoft",  
    "version": "15.0.0.0"  
  },  
]
```



One gigant leap for... Ehrm wait!

# "YOU BROKE US!"





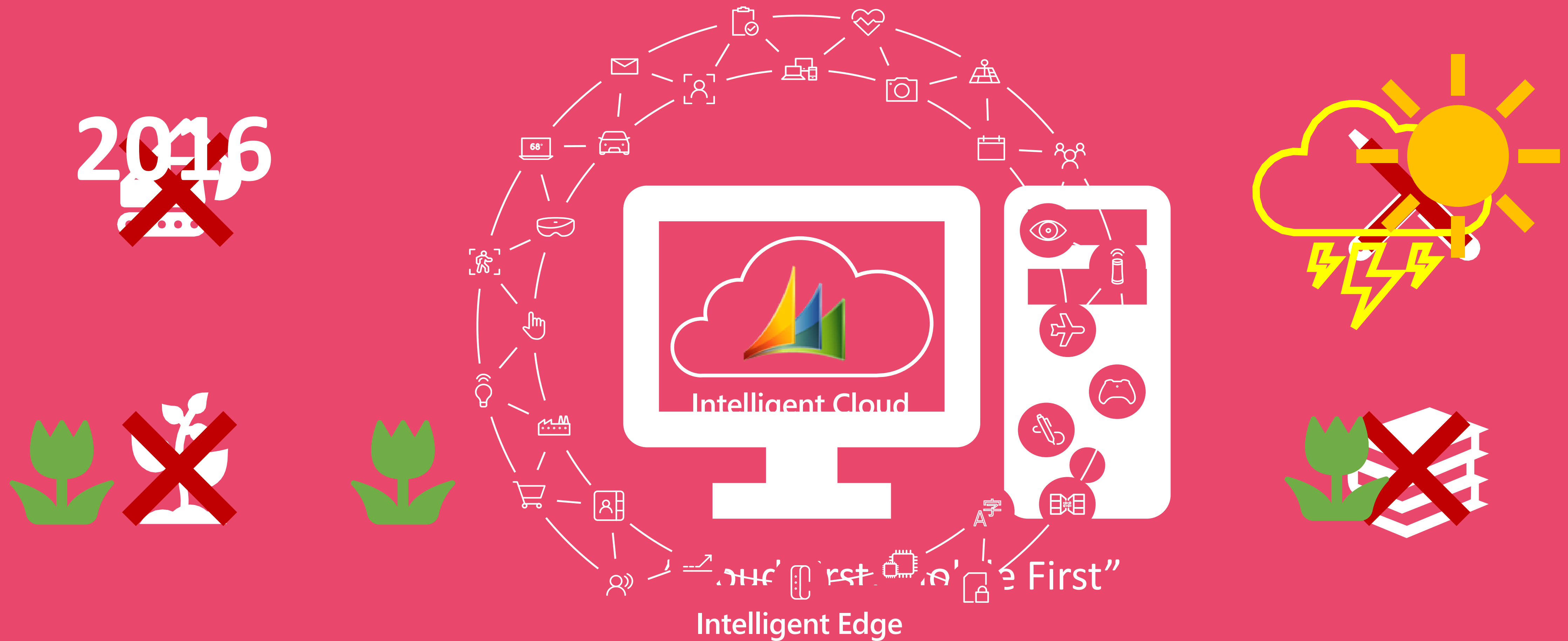


**“You can't build a great building on a weak foundation. You must have a solid foundation if you're going to have a strong superstructure.”**

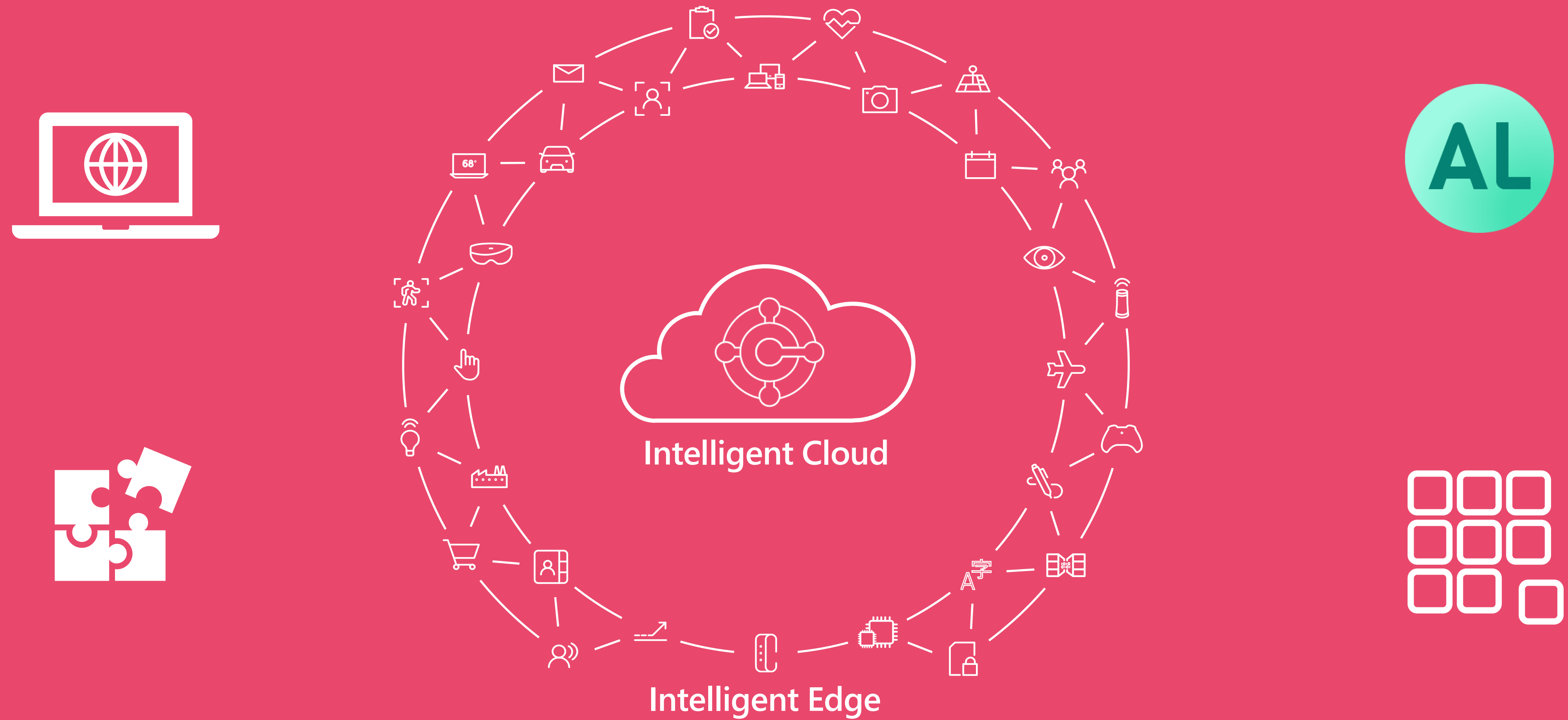
Gordon B. Hinckley (1910 – 2008)



# Why now?

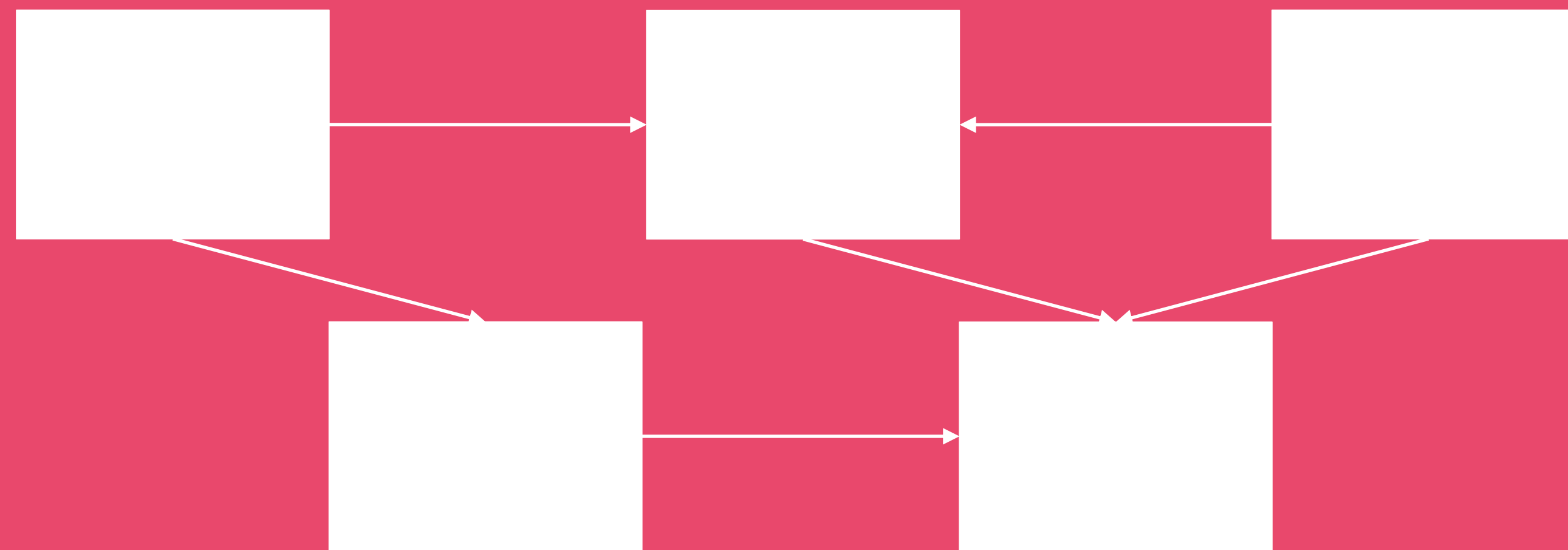


# Why now?

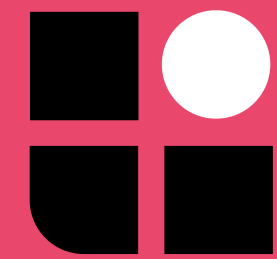




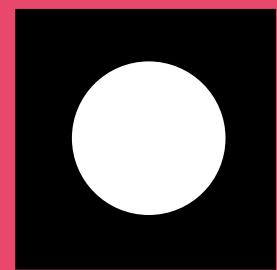
# A module's architecture



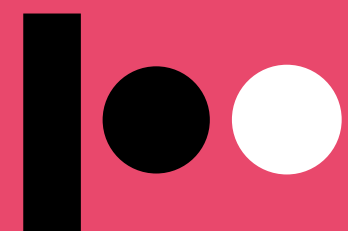
# A module's architecture



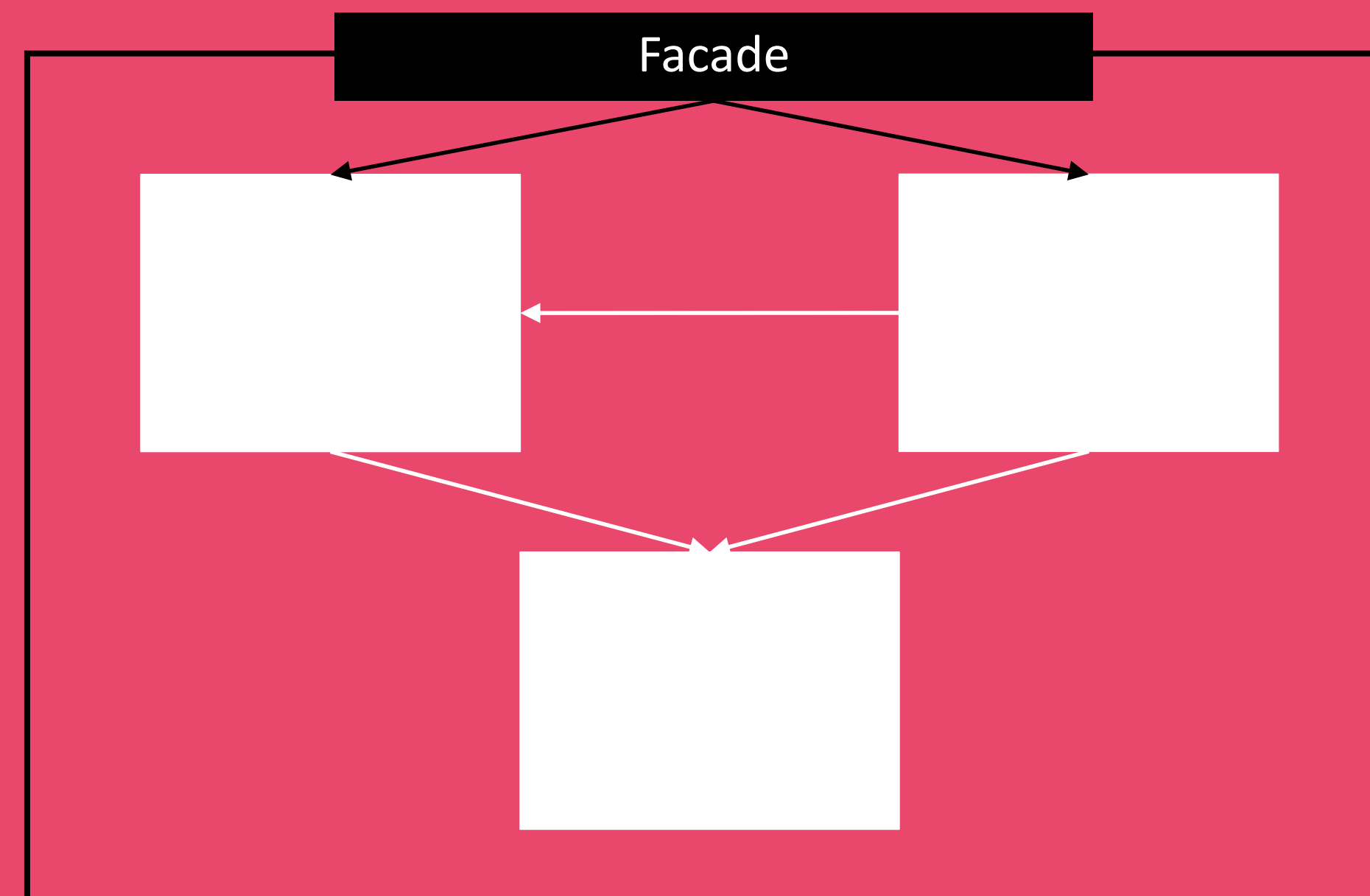
Coherence



Encapsulate



Stable API



Separate Extension



Access = Internal



XML Code Doc

# Detour into access modifiers

```
Access = Internal;
```

**Definition:** The Access property is an access modifier which allows controlling the accessibility level of methods and objects.



## Object Level:

- Internal: Access only by code in the same module
- Public (Default): Access by any other code in same module and in other modules that reference it

## Field/Function Level:

- Internal: Access only by code in the same module, but not from another module.
- Local: Access only by code in the same table or table extension where the field is defined.
- Protected: Access only by code in the same table or table extensions of that table.
- Public (Default): Access by any other code in same module and in other modules that reference it.

# Controlled extensibility

```
Extensible = true;
```

**Definition:** Sets whether an object can be extended or not.

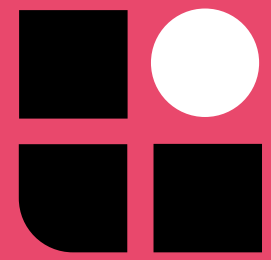
Extensibility needs to be designed into your code.  
Occasionally, it is better to prevent extensibility.

The property works on all extensible objects, which currently are:

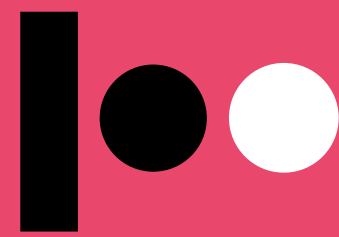
- Tables
- Pages
- Enums



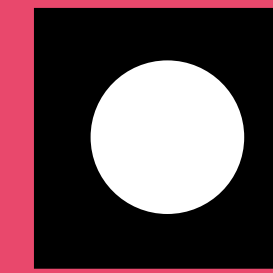
# Advantages of modules



They separate concerns and are functionally coherent



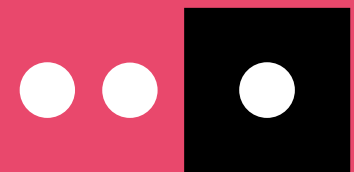
They have stable, well documented APIs that are exposed through facades



They encapsulate complexity and hide implementation details



They have a small attack surface, which makes them easier to secure



They are faster to compile and publish, which supports a good developer experience



Their size and clear purpose allow for a code contribution model and focused design discussions

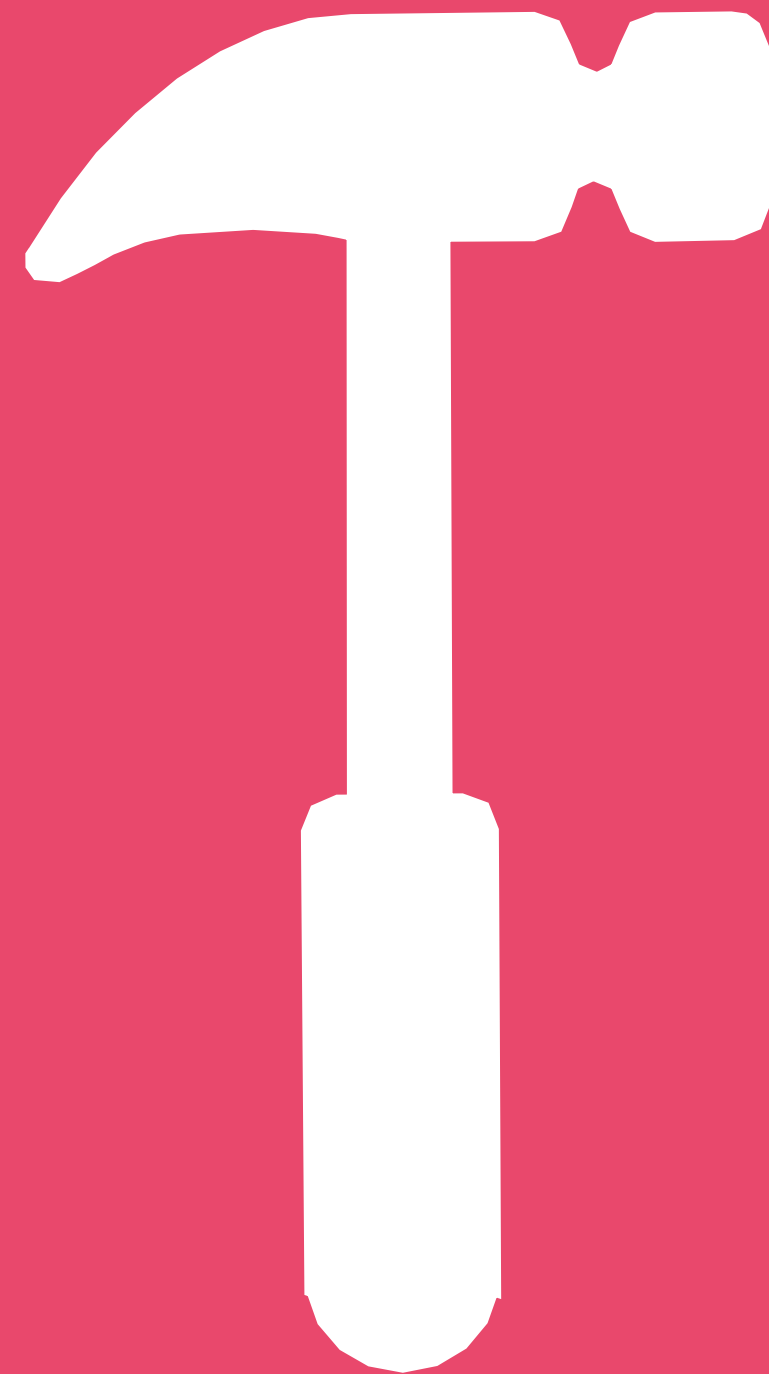


They are easier to performance test and monitor



They allow for individual versioning and effective upgrading

# Introducing the System Application



Base Application

} System Application



# Idea behind the System Application



**KEEP  
CALM  
AND**

**GET YOUR NOSE OUT OF  
MY BUSINESS**

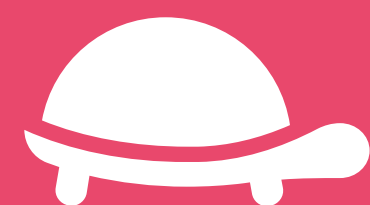
# ...jumping into GitHub

Everything you need to know  
about the System Application,  
is on GitHub!

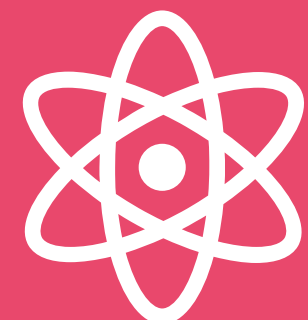
Let's go there!



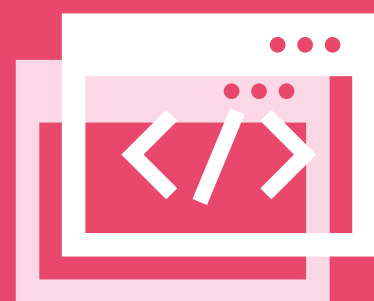
# Speaking of Breaking Changes... Why not like that in this release?



Ensuring backwards compatibility through deprecation is costly and slows down progress considerably.



The System Application needed a "critical mass" to establish itself



The new model will require complex "side by side" implementations



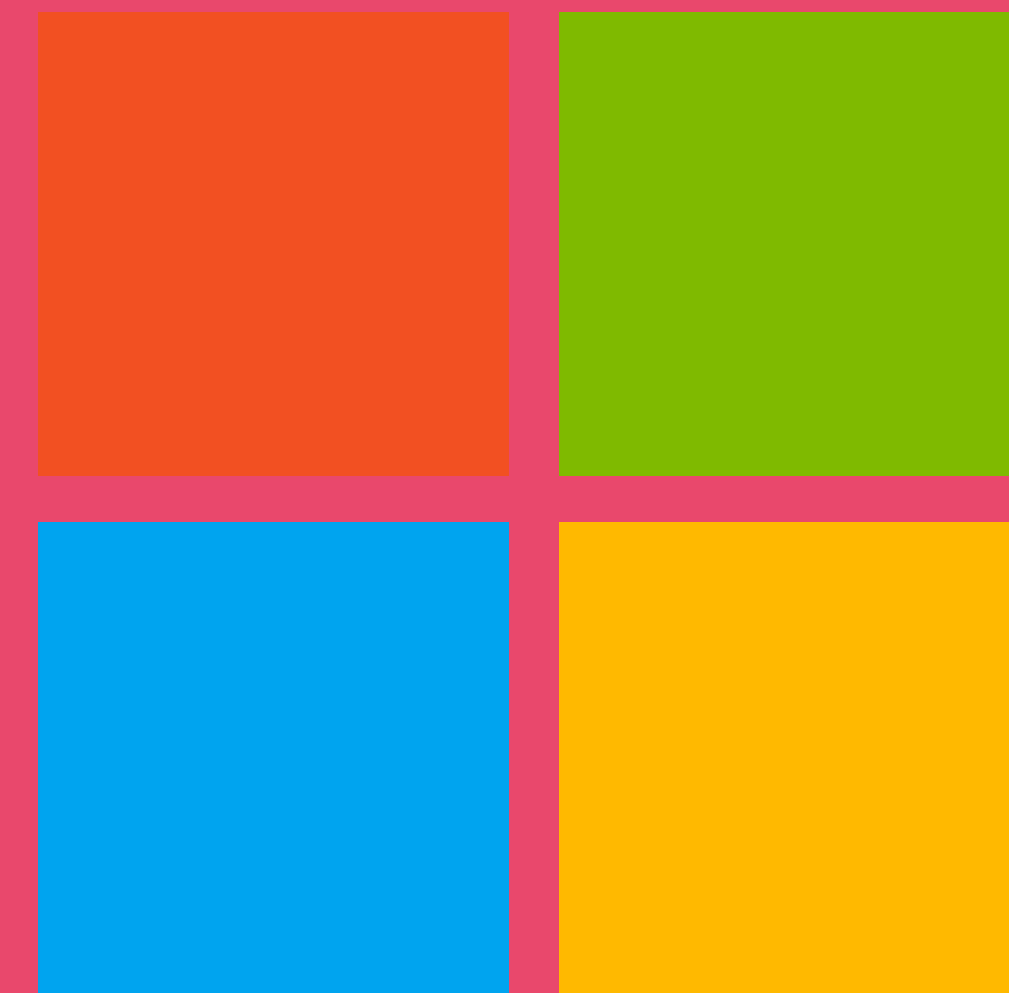
The Modules had to be "clean" to set proper examples\*

\* = A few exceptions were made to ensure backwards compatibility on some of the most commonly used APIs

# How do I participate or request change?



[github.com/Microsoft/alappextensions](https://github.com/Microsoft/alappextensions)

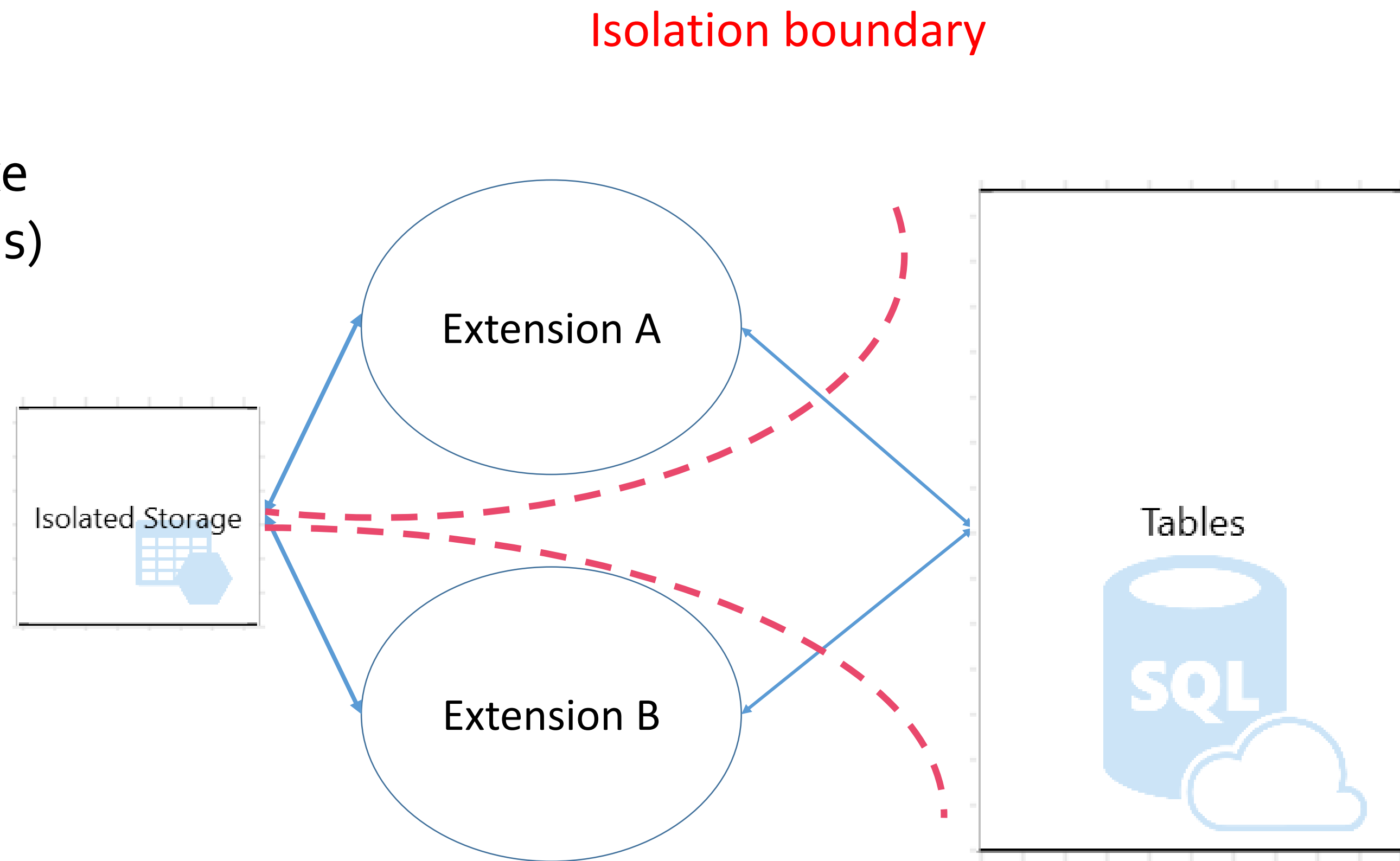


[aka.ms/bcideas](https://aka.ms/bcideas)

# Security Best Practices when Building Extensions

# Storing data

- Non-Sensitive data
- Sensitive customer data (like social security number or credit cards)
  - Use Isolated Storage
  - It offers data isolation between extensions.





# Isolated Storage - sample code

## DataScope:

- User: only current user
- Company: only current company
- CompanyAndUser: only current user within current company
- Module: this extension (by default)

```
codeunit 50101 "Isolated Storage Mgmt - v1.0"
{
    0 references
    procedure StoreSecret(SecretKey: Text; SecretValue: Text)
    begin
        IsolatedStorage.Set(SecretKey, SecretValue, DataScope::Module)
    end;

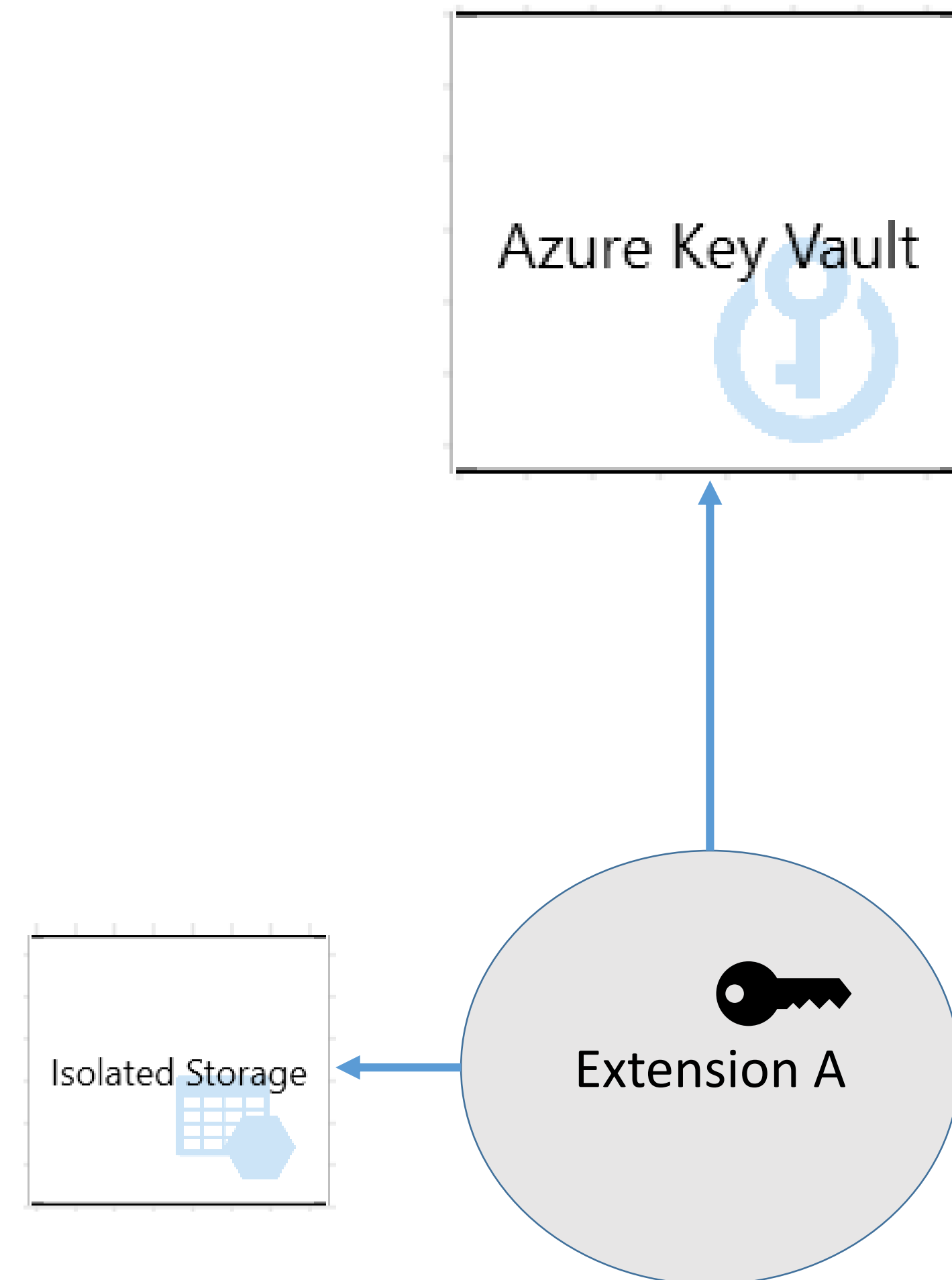
    0 references
    procedure StoreEncryptedSecret(SecretKey: Text; SecretValue: Text)
    begin
        IsolatedStorage.SetEncrypted(SecretKey, SecretValue, DataScope::Module)
    end;

    0 references
    procedure GetSecret(SecretKey: Text; var SecretValue: Text)
    begin
        IsolatedStorage.Get(SecretKey, DataScope::Module, SecretValue);
    end;

    0 references
    procedure ContainsSecret(SecretKey: Text): Boolean
    begin
        exit(IsolatedStorage.Contains(SecretKey, DataScope::Module));
    end;
}
```

# Storing Secrets?

- Use secret store like Azure KeyVault.
- Don't hardcode secrets.
- Store creds in Isolated Storage when setting up new tenant.



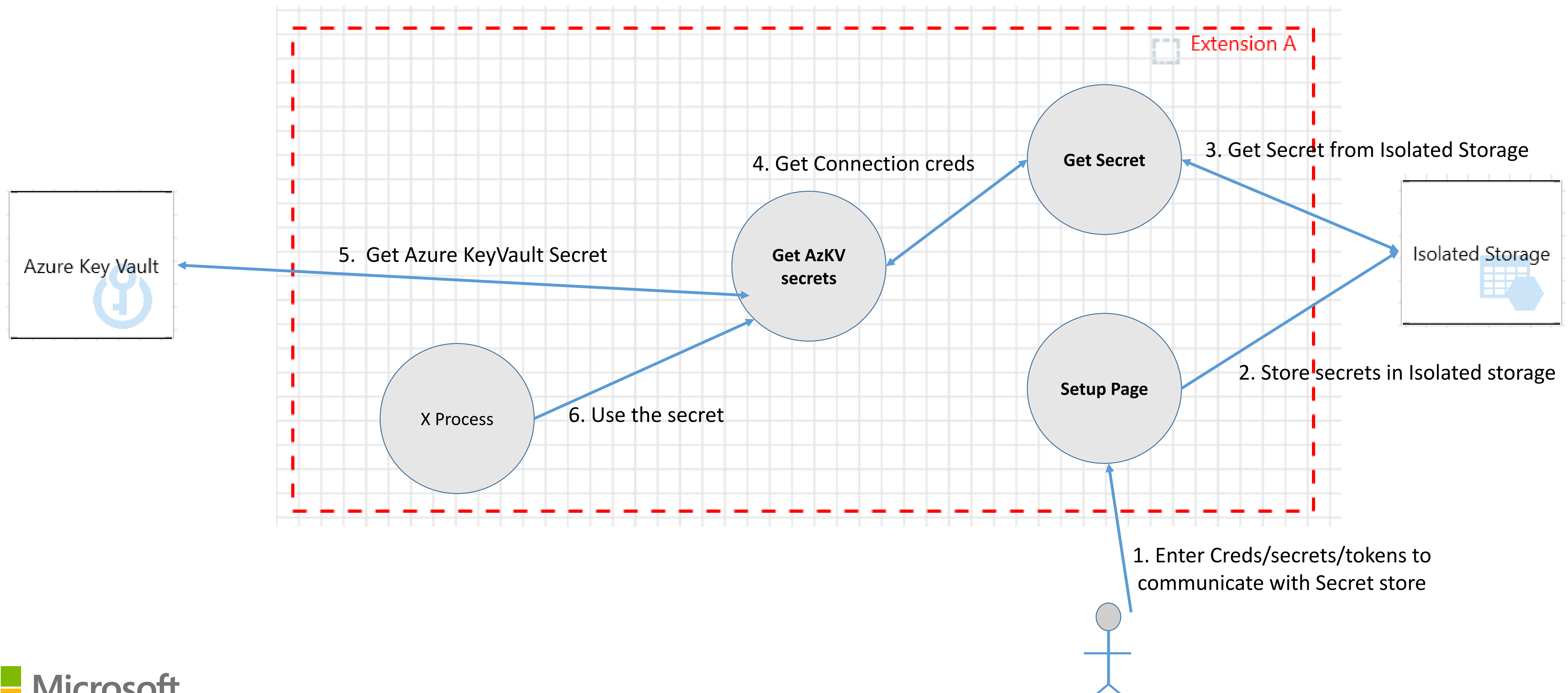
# Prototype using Azure Keyvault with an extension

# Threat Model

- How?
  - Build Data Flow diagram
  - Identify Risk/Threat
  - Mitigate
  - Validate
- Why?
  - Identify and address threats/risks
  - Increase risk awareness and understanding



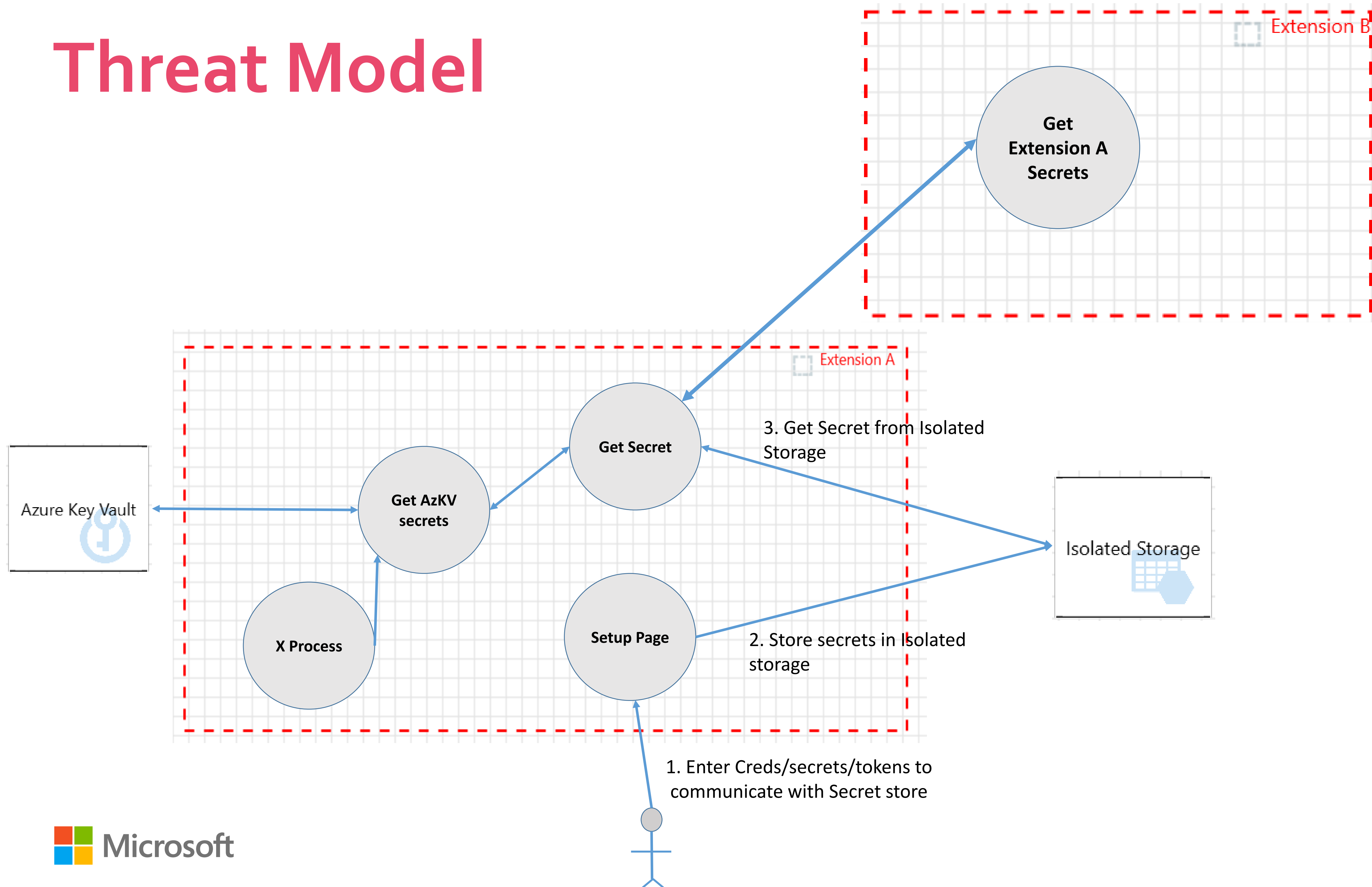
# Threat Model



# Procedure scope (access modifier)

- In AL: procedures are external by default.
- Make sure to mark any sensitive procedure as “Internal”, otherwise other extension that has dependency on yours could call these procedures.

# Threat Model



# Proper access modifiers for sensitive procedure

```
codeunit 50101 "Isolated Storage Mgmt - v1.0"
{
    0 references
    procedure StoreSecret(SecretKey: Text; SecretValue: Text)
    begin
        IsolatedStorage.Set(SecretKey, SecretValue, DataScope::Module)
    end;

    0 references
    procedure StoreEncryptedSecret(SecretKey: Text; SecretValue: Text)
    begin
        IsolatedStorage.SetEncrypted(SecretKey, SecretValue, DataScope::Module)
    end;

    0 references
    procedure GetSecret(SecretKey: Text; var SecretValue: Text)
    begin
        IsolatedStorage.Get(SecretKey, DataScope::Module, SecretValue);
    end;

    0 references
    procedure ContainsSecret(SecretKey: Text): Boolean
    begin
        exit(IsolatedStorage.Contains(SecretKey, DataScope::Module));
    end;
}
```

```
1 codeunit 50102 "Isolated Storage Mgmt - v1.1"
2 {
3     0 references
4     internal procedure StoreSecret(SecretKey: Text; SecretValue: Text)
5     begin
6         IsolatedStorage.Set(SecretKey, SecretValue, DataScope::Module)
7     end;
8
9     0 references
10    internal procedure StoreEncryptedSecret(SecretKey: Text; SecretValue: Text)
11    begin
12        IsolatedStorage.SetEncrypted(SecretKey, SecretValue, DataScope::Module)
13    end;
14
15    0 references
16    internal procedure GetSecret(SecretKey: Text; var SecretValue: Text)
17    begin
18        IsolatedStorage.Get(SecretKey, DataScope::Module, SecretValue);
19    end;
20
21    0 references
22    internal procedure ContainsSecret(SecretKey: Text): Boolean
23    begin
24        exit(IsolatedStorage.Contains(SecretKey, DataScope::Module));
25    end;
26 }
```



# Protect your IP

“ShowMyCode” property in App.json file

- False by default
- If you set it to true:
  - Your code is visible through dependency and debugging

```
app.json ×
app.json > showMyCode
13  "dependencies": [],
14  "screenshots": [],
15  "platform": "14.0.0.0",
16  "application": "14.0.0.0",
17  "idRanges": [
18    {
19      "from": 50100,
20      "to": 50149
21    }
22  ],
23  "showMyCode": true,
24  "runtime": "3.0"
25 }
```

# Protect your IP

- Compiler generates .APP file an archived .APP file
- When distributing your extension, it is recommended to create a runtime package
  - If showMyCode is false, runtime package will NOT include AL code.

The following example gets the NAV App runtime package with the provided name and version.

```
Get-NAVAppRuntimePackage -ServerInstance DynamicsNAV -AppName 'Proseware SmartApp' -Version 2.3.4.500 -ExtensionPath  
'Proseware SmartApp_2.3.4.500_runtime.app'
```

- <https://docs.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-creating-runtime-packages>

# Protect sensitive data during debugging

- If you decide to set ShowMyCode to true, mark procedures/variables that handles sensitive data as NonDebuggable

```
codeunit 50102 "Isolated Storage Mgmt - v1.1"
{
    0 references
    internal procedure StoreSecret(SecretKey: Text; SecretValue: Text)
    begin
        IsolatedStorage.Set(SecretKey, SecretValue, DataScope::Module)
    end;

    0 references
    internal procedure StoreEncryptedSecret(SecretKey: Text; SecretValue: Text)
    begin
        IsolatedStorage.SetEncrypted(SecretKey, SecretValue, DataScope::Module)
    end;

    0 references
    internal procedure GetSecret(SecretKey: Text; var SecretValue: Text)
    begin
        IsolatedStorage.Get(SecretKey, DataScope::Module, SecretValue);
    end;

    0 references
    internal procedure ContainsSecret(SecretKey: Text): Boolean
    begin
        exit(IsolatedStorage.Contains(SecretKey, DataScope::Module));
    end;
}

1 codeunit 50103 "Isolated Storage Mgmt - v1.2"
2 {
3     [NonDebuggable]
4     4 references
5     internal procedure StoreSecret(SecretKey: Text; SecretValue: Text)
6     begin
7         IsolatedStorage.Set(SecretKey, SecretValue, DataScope::Module)
8     end;
9     [[NonDebuggable]]
10    0 references
11    internal procedure StoreEncryptedSecret(SecretKey: Text; SecretValue: Text)
12    begin
13        IsolatedStorage.SetEncrypted(SecretKey, SecretValue, DataScope::Module)
14    end;
15    [NonDebuggable]
16    4 references
17    internal procedure GetSecret(SecretKey: Text; var SecretValue: Text)
18    begin
19        IsolatedStorage.Get(SecretKey, DataScope::Module, SecretValue);
20    end;
21    [NonDebuggable]
22    4 references
23    internal procedure ContainsSecret(SecretKey: Text): Boolean
24    begin
25        exit(IsolatedStorage.Contains(SecretKey, DataScope::Module));
26    end;
27 }
```

## After the move to AL...







# Interfaces & Enum Extensibility

```
/// Membership levels
enum 50130 LoyaltyLevel implements ILoyaltyBenefits
{
    Extensible = true;

    value(0; Bronze)
    {
        Implementation = ILoyaltyBenefits = BronzeLevelBenefits;
    }
    value(1; Silver)
    {
        Implementation = ILoyaltyBenefits = SilverLevelBenefits;
    }
    value(2; Gold)
    {
        Implementation = ILoyaltyBenefits = GoldLevelBenefits;
    }
}
```

```
/// Interface for a frequent-flyer member
interface ILoyaltyBenefits
{
    /// Check if member has lounge access
    procedure HasLoungeAccess(): Boolean;

    /// Get Discount % on in-flight service
    procedure GetDiscountPct(): Decimal;
}
```

```
/// Interface/Enum use
table 50214 Member
{
    fields
    {
        field(1; No; Integer) { }
        field(2; Name; Text[100]) { }
        field(3; LoyaltyLevel; enum LoyaltyLevel) { }
    }

    procedure CalcMemberPrice(Price: Decimal): Decimal
    var
        Benefits: interface ILoyaltyBenefits;
    begin
        Benefits := LoyaltyLevel; // Cast enum -> interface
        exit(1 - Benefits.GetDiscountPct() / 100) * Price);
    end;

    procedure HasLoungeAccess(): Boolean;
    var
        Benefits: interface ILoyaltyBenefits;
    begin
        Benefits := LoyaltyLevel; // Cast enum -> interface
        exit(Benefits.HasLoungeAccess());
    end;
}
```

# Interfaces & Enum Extensibility

```
/// Membership levels
enum 50130 LoyaltyLevel implements ILoyaltyBenefits
{
    Extensible = true;

    value(0; Bronze)
    {
        Implementation = ILoyaltyBenefits = BronzeLevelBenefits;
    }
    value(1; Silver)
    {
        Implementation = ILoyaltyBenefits = SilverLevelBenefits;
    }
    value(2; Gold)
    {
        Implementation = ILoyaltyBenefits = GoldLevelBenefits;
    }
}
```

```
/// Interface for a frequent-flyer member
interface ILoyaltyBenefits
{
    /// Check if member has lounge access
    procedure HasLoungeAccess(): Boolean;

    /// Get Discount % on in-flight service
    procedure GetDiscountPct(): Decimal;
}
```

## Adding Diamond Level

```
/// Diamond level enum value
enumextension 50143 DiamondLoyaltyLevel extends LoyaltyLevel
{
    value(5140; Diamond)
    {
        Implementation = ILoyaltyBenefits = DiamondLevelBenefits;
    }
}
```

```
/// Diamond level implementation
codeunit 50143 DiamondLevelBenefits implements ILoyaltyBenefits
{
    procedure HasLoungeAccess(): Boolean;
    begin
        exit(true)
    end;

    procedure GetDiscountPct(): Decimal;
    begin
        exit(30);
    end;
}
```

# Application Extensibility Example

```
// Interface for a shipping provider
interface IShipping
{
    // Configure the Provider
    procedure Configure();
    procedure IsConfigured(): Boolean;
    procedure GetQuote(SalesHeader: record "Sales Header"): Decimal;
    procedure RequestPickup(Location: record Location): Boolean;
    procedure GetProviderCompanyUrl(): Text;
    procedure TrackShipment(ShipmentId: Text[100];
        TrackingRecords: record ShipmentTracking);
    procedure GetShipmentStatus(ShipmentId: Text[100]): enum ShipmentStatus;

    // More shipping ...
}
```

```
enum 60000 ShippingProvider implements IShipping
{
    Extensible = true;
```

# Application Extensibility Example

```
procedure RequestPickup(Location: Record Location): Boolean;  
procedure GetProviderCompanyId(): Text;  
procedure TrackShipment(ShipmentId: Text[100];  
    TrackingRecords: Record ShipmentTracking)  
procedure GetShipmentStatus(ShipmentId: Text[100]): enum ShipmentStatus;  
  
    // More shipping ...  
}
```

```
enum 60000 ShippingProvider implements IShipping  
{  
    Extensible = true;  
  
    value(0; None)  
    {  
        Caption = 'Select a shipping provider';  
        Implementation = IShipping = NoneShippingProviderImpl;  
    }  
}
```

```
codeunit 60000 NoneShippingProviderImpl implements IShipping  
{  
    procedure Configure();  
    begin  
        Error('No Shipping Provider configured');  
    end;
```



```
Caption = 'Select a shipping provider';  
Implementation = IShipping = NoneShippingProviderImpl;
```

# Application Extensibility Example



```
codeunit 60000 NoneShippingProviderImpl implements IShipping  
{  
    procedure Configure();  
    begin  
        Error('No Shipping Provider configured');  
    end;  
  
    procedure IsConfigured(): Boolean;  
    begin  
        exit(false);  
    end;  
  
    procedure GetQuote(SalesHeader: Record "Sales Header"): Decimal;  
    begin  
    end;  
  
    // More None implementation ...  
}
```

# Application Extensibility Example

```
begin
end;
// None None Implementation...
}

table 60000 ShippingSetup
{
    fields
    {
        field(1; PK; Integer) { }
        field(2; DefaultShippingProvider; enum ShippingProvider) { }
    }

    procedure IsConfigured(): Boolean;
    var
        Shipping: interface IShipping;
    begin
        Shipping := DefaultShippingProvider;
        exit(Shipping.IsConfigured());
    end;
}
```

```
page 60000 ShippingSetup
{
    SourceTable = ShippingSetup;

    layout {
        area(Content) {
```

# Application Extensibility Example

```
exit(Shipping.IsConfigured());
end;
}
page 60000 ShippingSetup
{
    SourceTable = ShippingSetup;

    layout {
        area(Content) {
            group(General) {
                field(DefaultShippingProvider; DefaultShippingProvider)
                {
                    ApplicationArea = All;
                    Caption = 'Default Shipping Provider';
                }
            }
        }
    }

    trigger OnOpenPage()
    begin
        if not Rec.Get() then begin
            Rec.Init();
            Rec.Insert();
        end;
    end;
}
```

# Application Extensibility Example

```
trigger OnOpenPage()
begin
    if not Rec.Get() then begin
        Rec.Init();
        Rec.Insert();
    end;
end;
}
```

```
enumextension 70000 LSAShippingProvider extends ShippingProvider
{
    value(70000; LSA)
    {
        Caption = 'Land, Sea & Air Shipping Service';
        Implementation = IShipping = LSAShippingProviderImpl;
    }
}
```

```
codeunit 70000 LSAShippingProviderImpl implements IShipping
{
    procedure Configure();
    begin
        Page.RunModal(Page::LSAShippingSetup)
    end;

    procedure IsConfigured(): Boolean;
    begin
```



# Application Extensibility Example

```
codeunit 70000 LSAShippingProviderImpl implements IShipping
{
    procedure Configure();
    begin
        Page.RunModal(Page::LSAShippingSetup)
    end;

    procedure IsConfigured(): Boolean;
    begin
        exit(true);
    end;

    procedure GetQuote(SalesHeader: Record "Sales Header"): Decimal;
    begin
        exit(CalculateQuotePrice(SalesHeader));
    end;

    local procedure CalculateQuotePrice(SalesHeader: Record "Sales Header"): Decimal;
    begin
        // LSA implementation ...
    end;

    // More LSA implementation ...
}
```



# Obsoleting objects

Process to control breaking changes and ensure adequate time to adopt

We will be following this for the base app

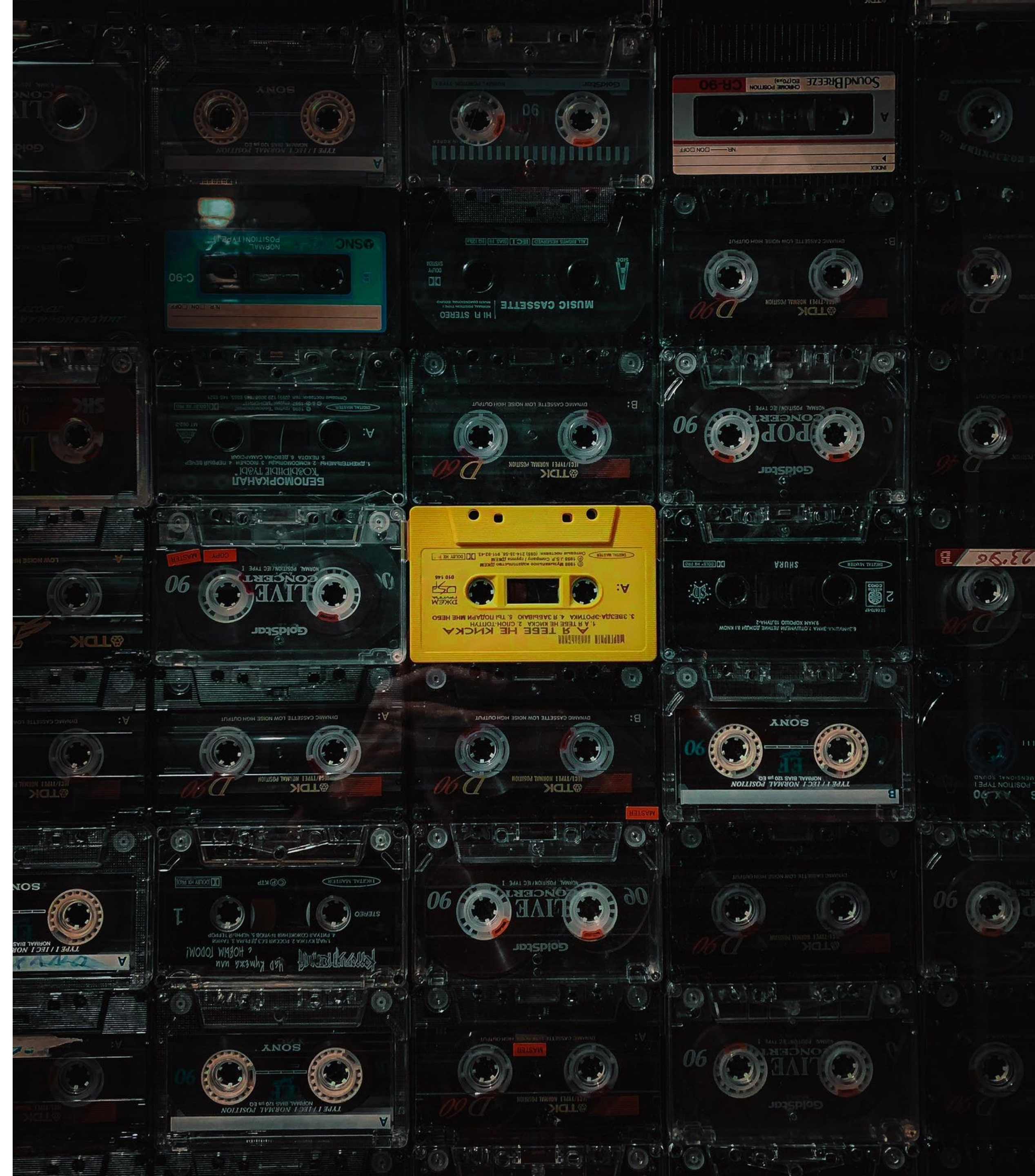
Extending Obsolete to cover all objects

Property on Codeunit, Page, Query, Report, XmlPort, Enum, etc.

Attribute on Procedure and Variable

Only obsolete removed for schema (table and field) due to support during upgrade

Note that we are still not removing fields from db, target for 2020 Wave 1 or later





# *DEMO*

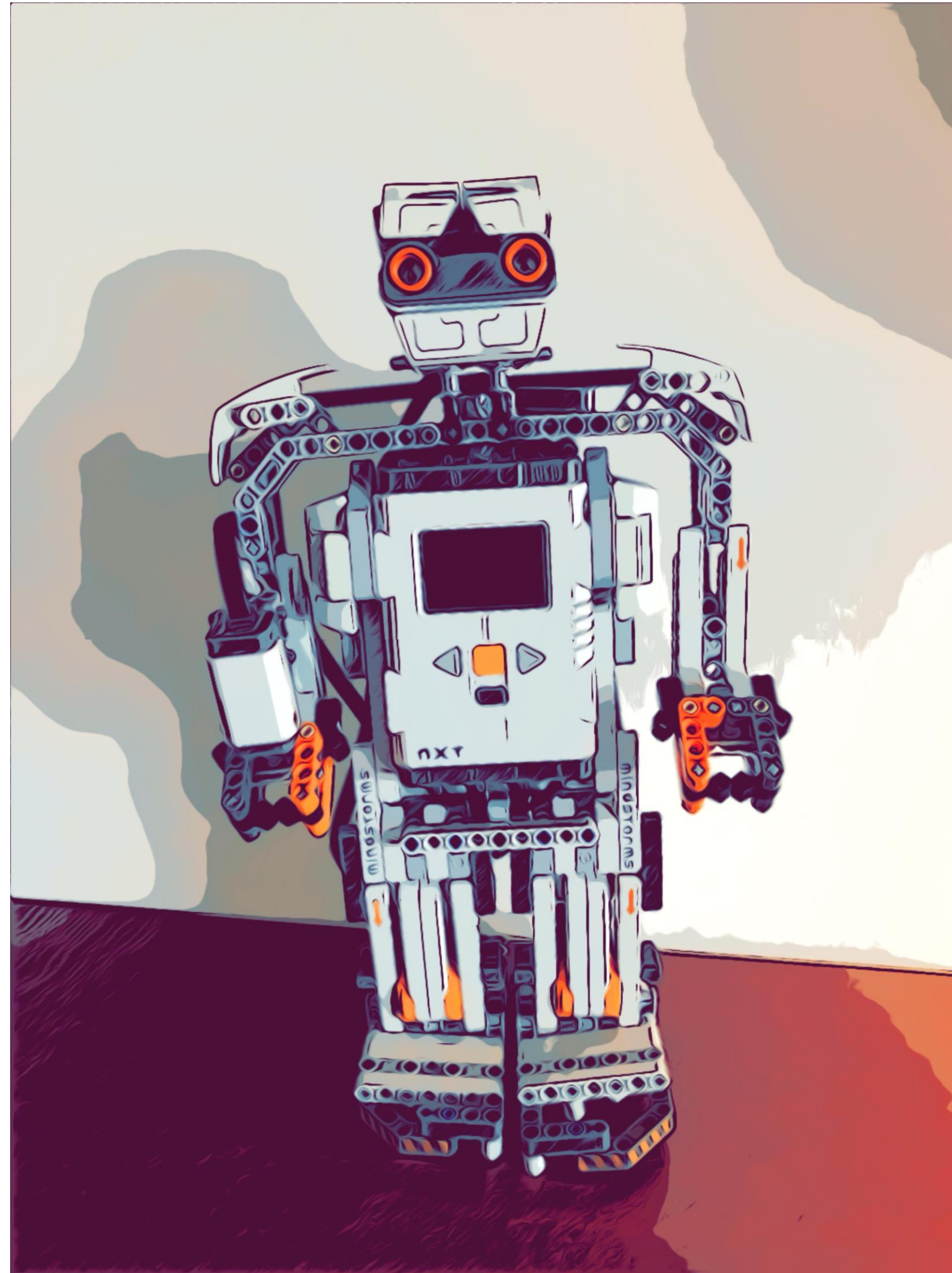
---

*?*





# Leveraging the Power of the Cloud



# Q&A

## Any Questions?

*Thank  
You!*