BC/NAV TECH DAYS 2022

mibuso.com

10 YEAR ANNIVERSARY

www.bctechdays.com

Advanced topics in test automation

Nikola Kukrika – Microsoft
Luc van Vugt – fluxxus.nl


BC/NAV
TECH
DAYS
2022
mibuso.com
Welcome
10 YEAR ANNIVERSARY
www.bctechdays.com

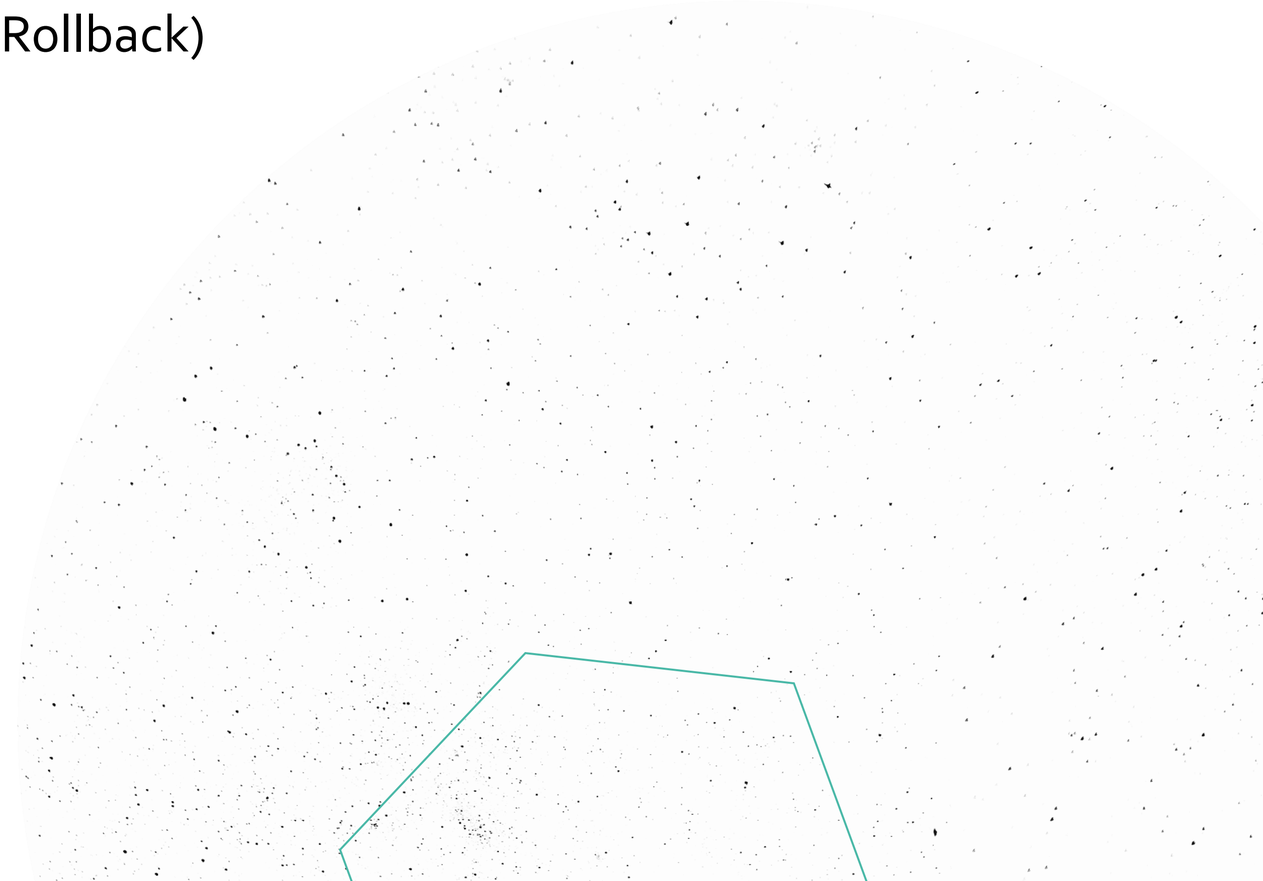
Agenda

BC/NAV TECH DAYS 2022
mibuso.com

- Advanced topics in test automation

  Mocking and Testability, Testing integrations (APIs, job queue, sessions), Demodata, Code coverage, Reusing Microsoft Tests, TransactionModel (AutoRollback)

- Being more efficient
- Good tests
- Internals of the test tooling



Us

Nikola
Kukrika

Microsoft Development Center
Copenhagen

Luc
van Vugt

fluxxus.nl

Why?

test automation

Less bugs

Code coverage

Regression prevention

ATDD

TDD

Quality Metrics

Better code design

CI/CD

Documentation

Code Quality

Best practice

Quality Assurance

Regression prevention

Manual Testing

Quality Metrics

Efficient

Risk Management

Simply a must

Tools

Agile

Modern Software

Why NOT?

test automation

No 1. Reason: No time (money) to write tests

Costs are too high and will make us uncompetitive.

We are not used to doing it this way and our sales and management team don’t see the benefits and cannot be "convinced."

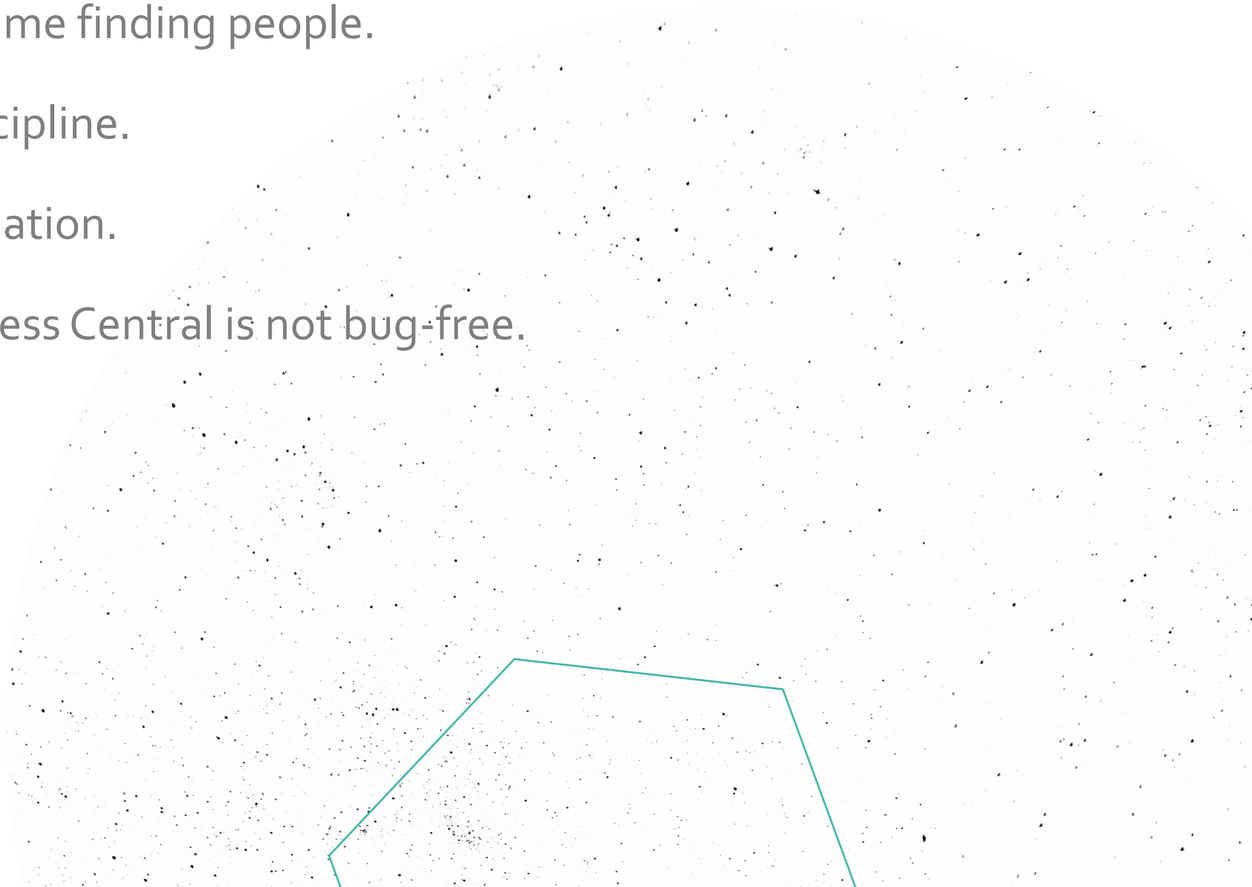
Who's going to write the test code? We already have a hard time finding people.

Our everyday business does not leave room to add a new discipline.

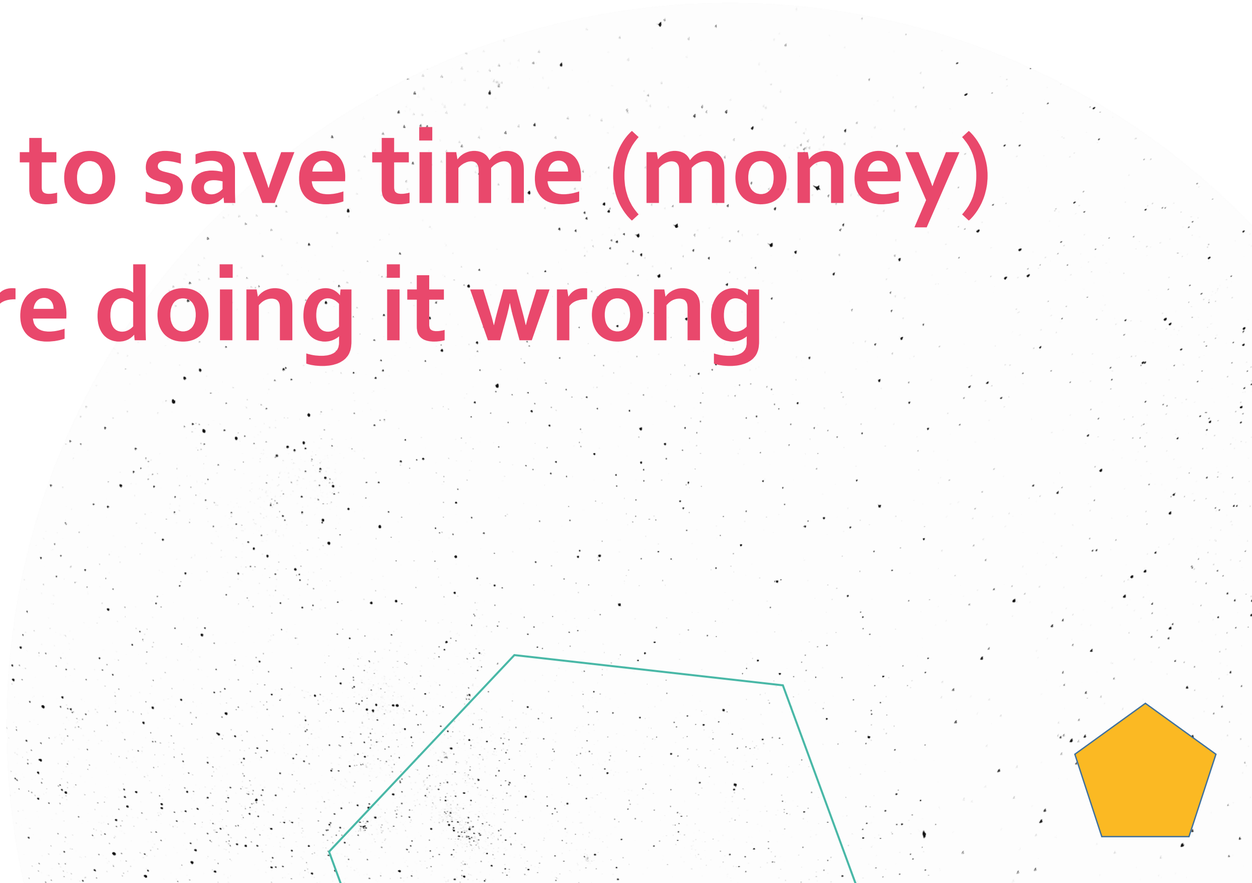
There are too many different projects to allow for test automation.

Microsoft has tests automated, and still, Dynamics 365 Business Central is not bug-free.

Customers do the testing, so why should we bother?



Test automation needs to save time (money)
If it is not, you are doing it wrong

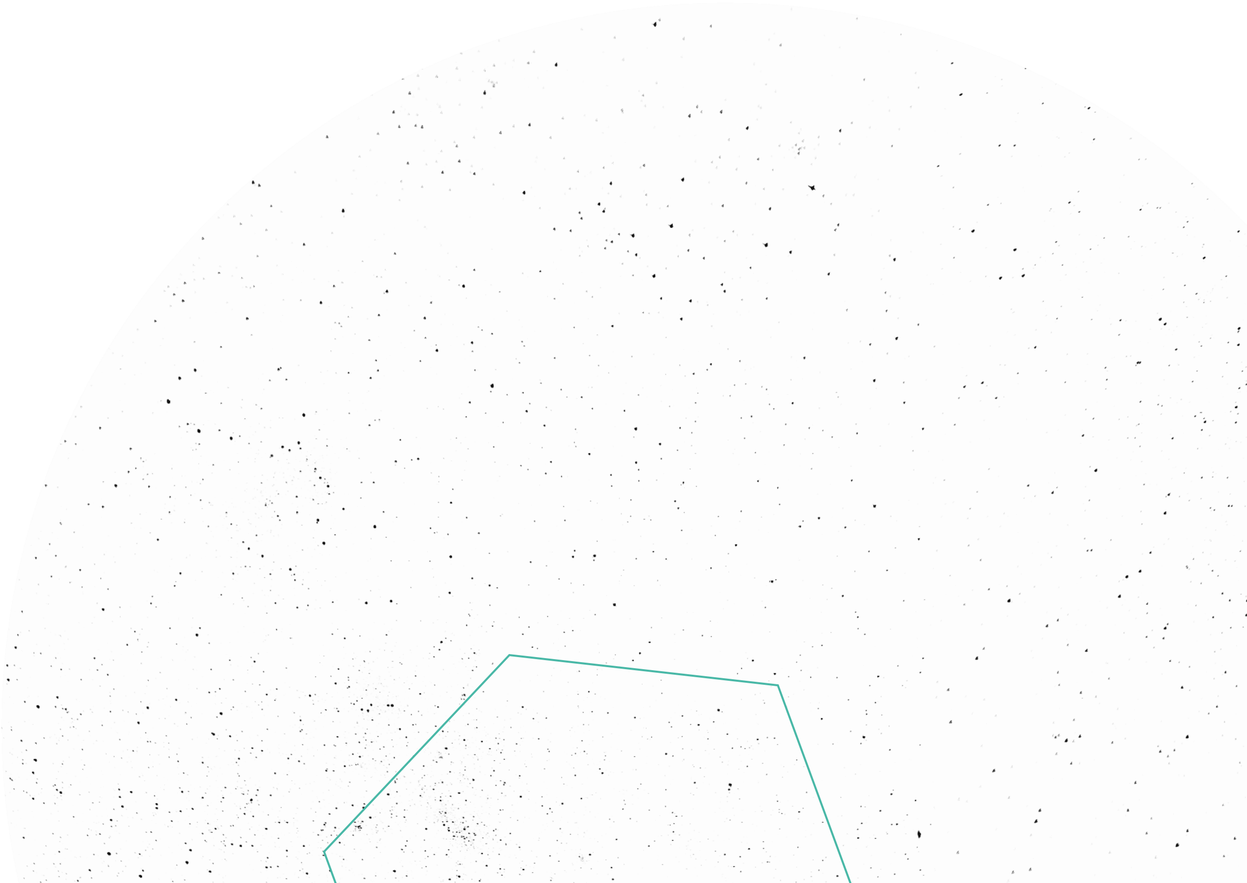


New feature

BC/NAV TECH DAYS 2022
mibuso.com

No testing

Test

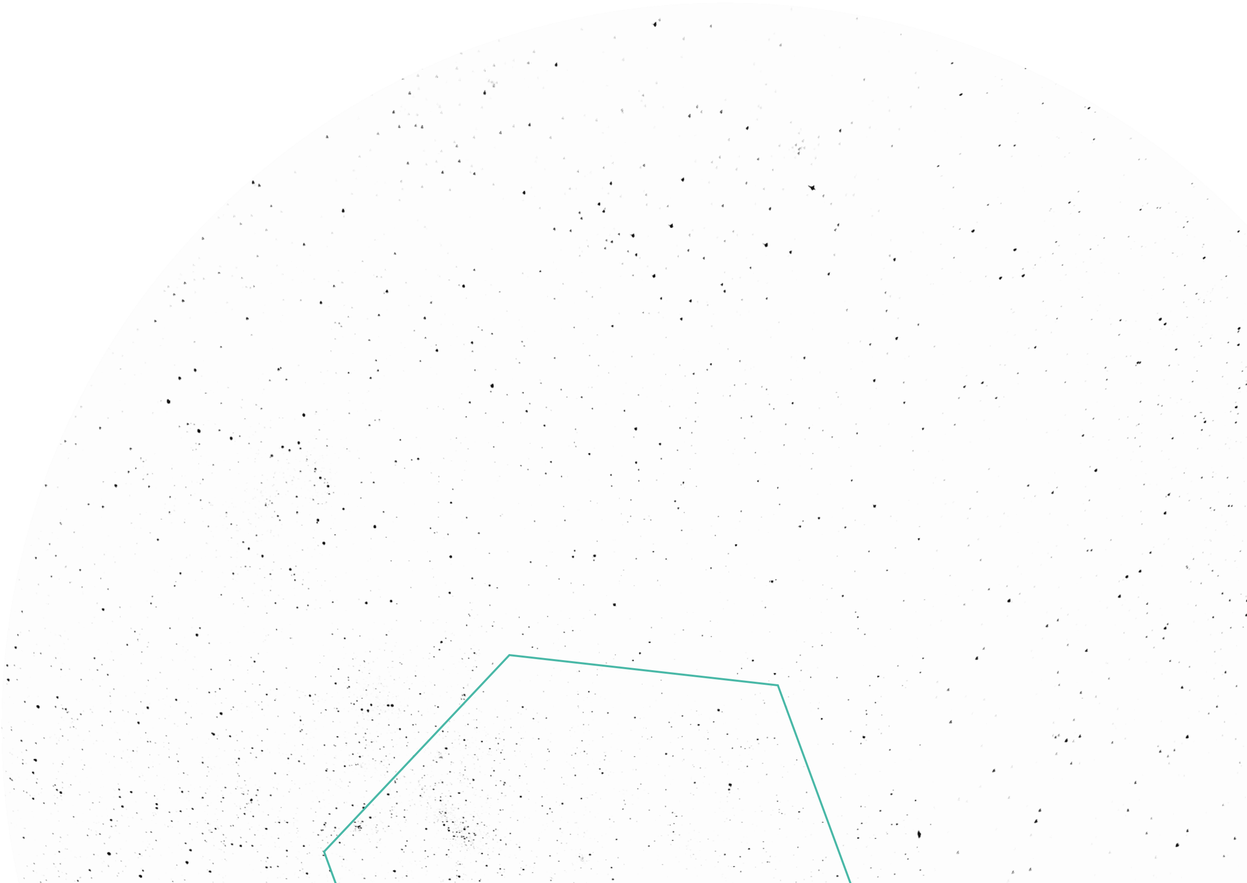


New feature

No testing

Test

Test



New feature

BC/NAV TECH DAYS 2022
mibuso.com

No testing

Manual Testing

Scenario testing
Ad hoc testing (exploratory…)
Specialized testing

No replacement for human eyes

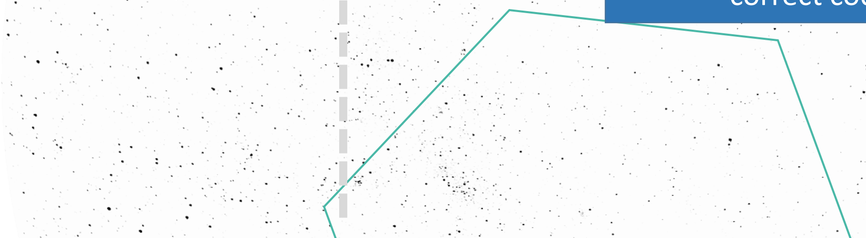
Automated Testing

TDD
Test Driven Development

Ensures that the code runs correctly

ATDD
Acceptance Test Driven Development

(Given, When, Then)
Ensures that you write the correct code



New feature

End User Testing

SDD
Scream Driven Development

It is not fixed until customer stops screaming

Manual Testing

Scenario testing
Ad hoc testing (exploratory…)
Specialized testing

No replacement for human eyes

Automated Testing

TDD
Test Driven Development

Ensures that the code runs correctly

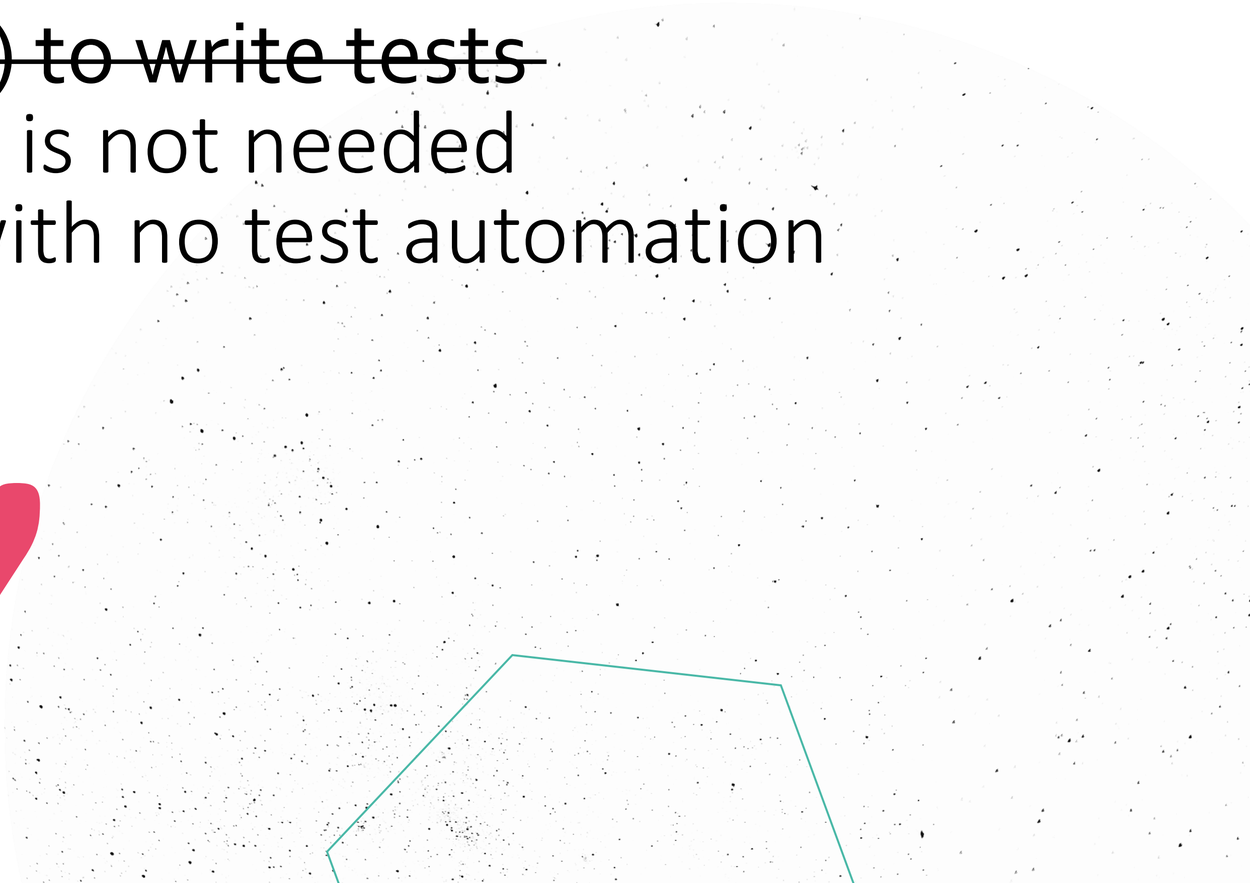
ATDD
Acceptance Test Driven Development

(Given, When, Then)
Ensures that you write the correct code

“

~~No time (money) to write tests~~
Test automation is not needed
We believe we can live with no test automation

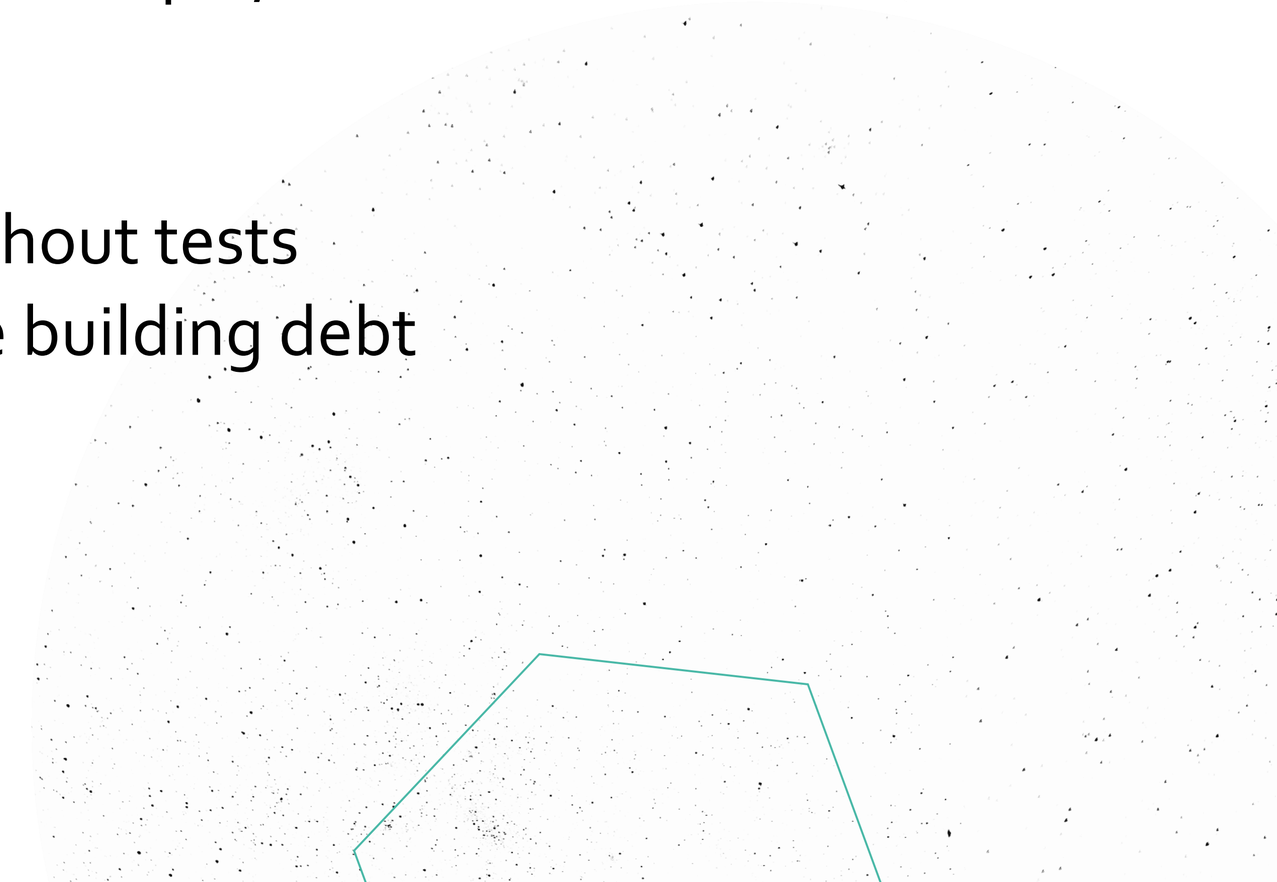
”



BC Test suite

- More than 30.000 AL Tests
- More than 80.000 Platform tests
- Dedicated tests (UI Rendering, API Test Scripts, Load/Performance…)

- We do not have enough
- It would not be possible to ship BC without tests
- If you decide not to automate, you are building debt



History

90s - The age of Software Testing

Difficult to roll out HF-es

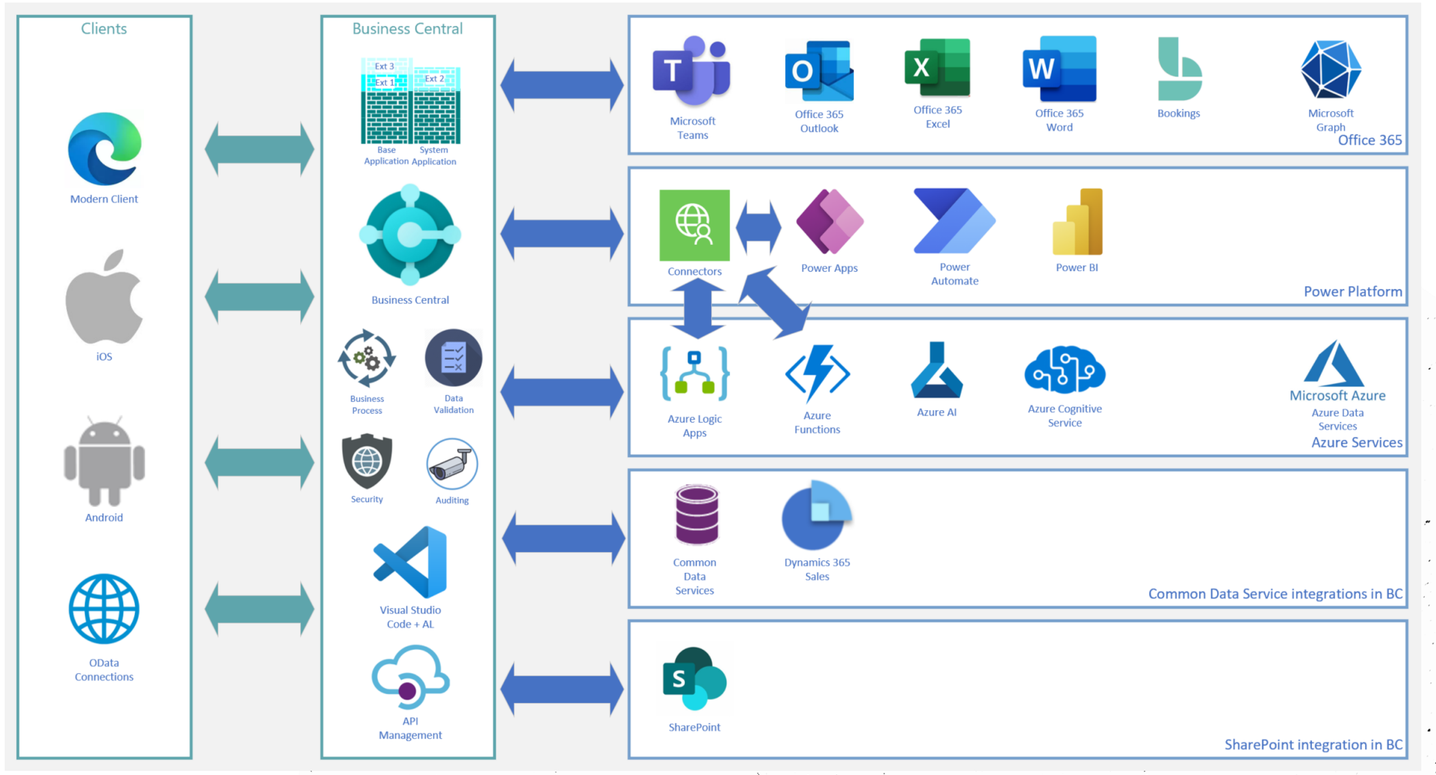
$100.000 - $200.000 cost per bug

Today

Complex Systems

Easy HF Rollout is a must

Early detection (telemetry)
Disaster Recovery



“

Only Dogma to follow is:
Do not follow any Dogmas

”

Good Tests

Good Tests

BC/NAV TECH DAYS 2022
mibuso.com

1. Cover the risk

Good Tests

BC/NAV TECH DAYS 2022
mibuso.com

1. Cover the risk

Good Tests

BC/NAV TECH DAYS 2022
mibuso.com

1. Cover the risk

Good Tests

BC/NAV TECH DAYS 2022
mibuso.com

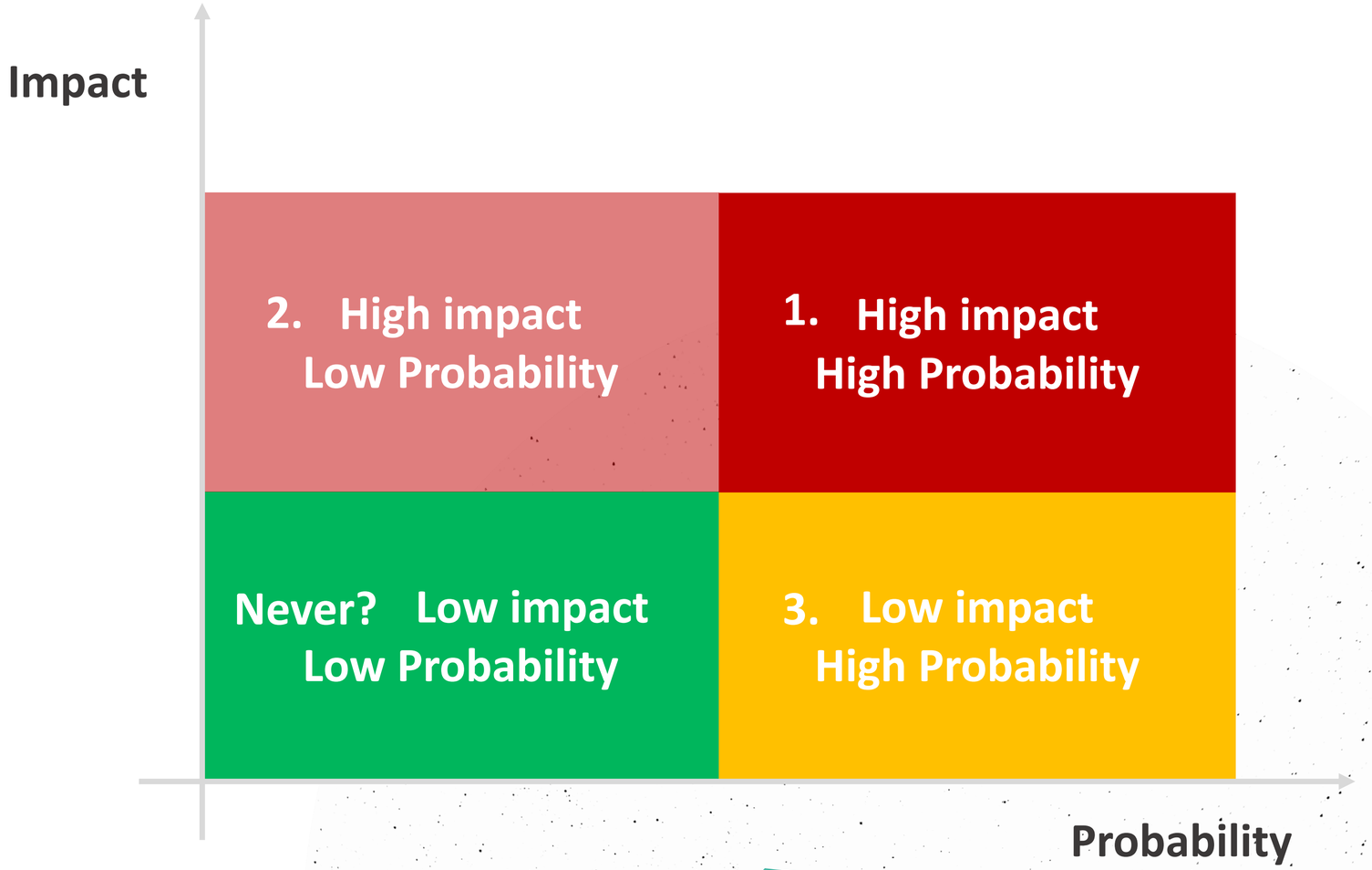
1. Cover the risk

Cover the risk

Cover all high impact

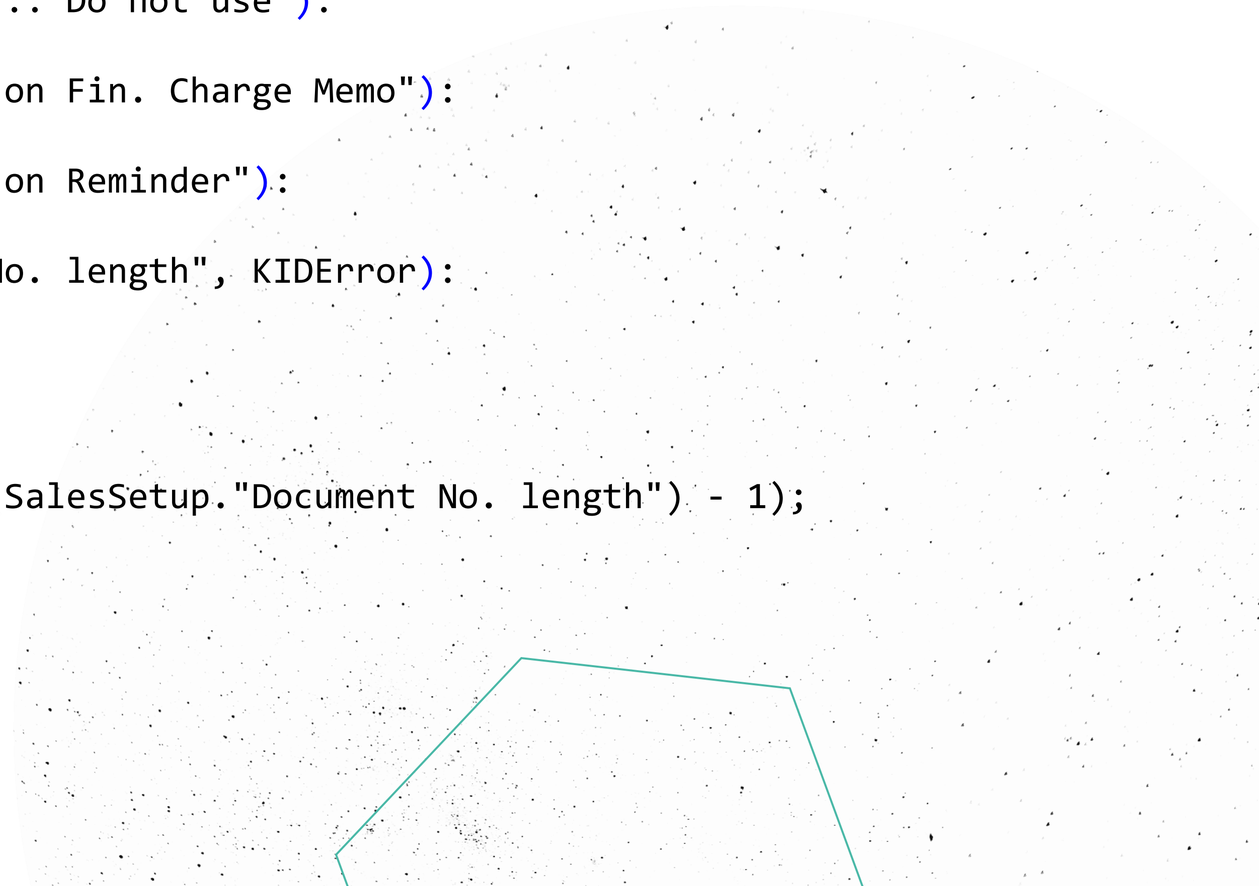
Test as close to the risk as possible

Test needs to add value



Tests need to add value


```
procedure GenerateGiroKID(DocumentNo: Code[20]; var GiroKID: Text[25])
var
    SalesSetup: Record "Sales & Receivables Setup";
begin
    SalesSetup.Get();
    case true of
        (SalesSetup."KID Setup" = SalesSetup."KID Setup"::"Do not use"):
            GiroKID := '';
        (DocumentType = 2) and (not SalesSetup."Use KID on Fin. Charge Memo"):
            GiroKID := '';
        (DocumentType = 3) and (not SalesSetup."Use KID on Reminder"):
            GiroKID := '';
        not KIDTextOK(DocumentNo, SalesSetup."Document No. length", KIDError):
            GiroKID := '';
        else
            // Format KID
            SalesSetup.TestField("Document No. length");
            GiroKID := CopyStr(DocumentNo, 1, MaxStrLen(SalesSetup."Document No. length") - 1);
      end;
end;
```



Tests need to add value

BC/NAV TECH DAYS 2022
mibuso.com

```
[Test]
procedure TestGiroKidBlankWhenSalesSetupSetToDontUse()
var
    SalesSetup: Record "Sales & Receivables Setup";
    DocumentTools: Codeunit DocumentTools;
    Any: Codeunit Any;
    Assert: Codeunit "Library Assert";
    GiroKID: Text[25];
    DocumentNo: Code[20];
begin
    // [GIVEN] Sales Setup marked as not use
    if not SalesSetup.Get() then
        SalesSetup.Insert();

    SalesSetup."KID Setup" := "KID Setup"::"Do not use";
    SalesSetup.Modify();

    // [GIVEN] A document number
    DocumentNo := UpperCase(Any.AlphabeticText(MaxStrLen(DocumentNo)));

    // [WHEN] We call generate KID
    DocumentTools.GenerateGiroKID(DocumentNo, GiroKID);

    // [THEN] Blank GiroKID is returned
    Assert.AreEqual('', GiroKID, 'Unexpected value for GiroKID');
end;
```

Tests need to add value


```
procedure GenerateGiroKID(DocumentNo: Code[20]; var GiroKID: Text[25])
var
    SalesSetup: Record "Sales & Receivables Setup";
begin
    SalesSetup.Get();
    case true of
(SalesSetup."KID Setup" = SalesSetup."KID Setup"::"Do not use"):
        GiroKID := '';
(DocumentType = 2) and (not SalesSetup."Use KID on Fin. Charge Memo"):
        GiroKID := '';
(DocumentType = 3) and (not SalesSetup."Use KID on Reminder"):
        GiroKID := '';
not KIDTextOK(DocumentNo, SalesSetup."Document No. length", KIDError):
        GiroKID := '';
    else
        // Format KID
        SalesSetup.TestField("Document No. length");
        GiroKID := CopyStr(DocumentNo, 1, MaxStrLen(SalesSetup."Document No. length") - 1);
    end;
end;
```

1 Test

1 Test

1 Test

1 Test

2 Tests

Tests need to add value

BC/NAV TECH DAYS 2022
mibuso.com

```
procedure GenerateGiroKID(DocumentNo: Code[20]; var GiroKID: Text[25])
var
    SalesSetup: Record "Sales & Receivables Setup";
begin
    SalesSetup.Get();
    case true of
        // [GIVEN] A document number
        DocumentNo := UpperCase(Any.AlphabeticText(MaxStrLen(DocumentNo)));

        // [WHEN] We call generate KID
        DocumentTools.GenerateGiroKID(DocumentNo, GiroKID);

        // [THEN] Blank GiroKID is returned
        Assert.AreEqual('', GiroKID, 'Unexpected value for GiroKID');
    end;
            SalesSetup.TestField("Document No. length");
            GiroKID := CopyStr(DocumentNo, 1, MaxStrLen(SalesSetup."Document No. length") - 1);
        end;
end;
```

1
1
1
1
2

Tests need to add value


```
procedure GenerateGiroKID(DocumentNo: Code[20]; var GiroKID: Text[25])
var
    SalesSetup: Record "Sales & Receivables Setup";
begin
    SalesSetup.Get();
    case true of
        (Sa
        (Dc
        (Dc
        not
        els
            // Format KID
            SalesSetup.TestField("Document No. length");
            GiroKID := CopyStr(DocumentNo, 1, MaxStrLen(SalesSetup."Document No. length") - 1);
    end;
end;
```

1 Test

1 Test

1 Test

1 Test

2 Tests

Cemented in place ☑

Tests need to add value

BC/NAV TECH DAYS 2022
mibuso.com

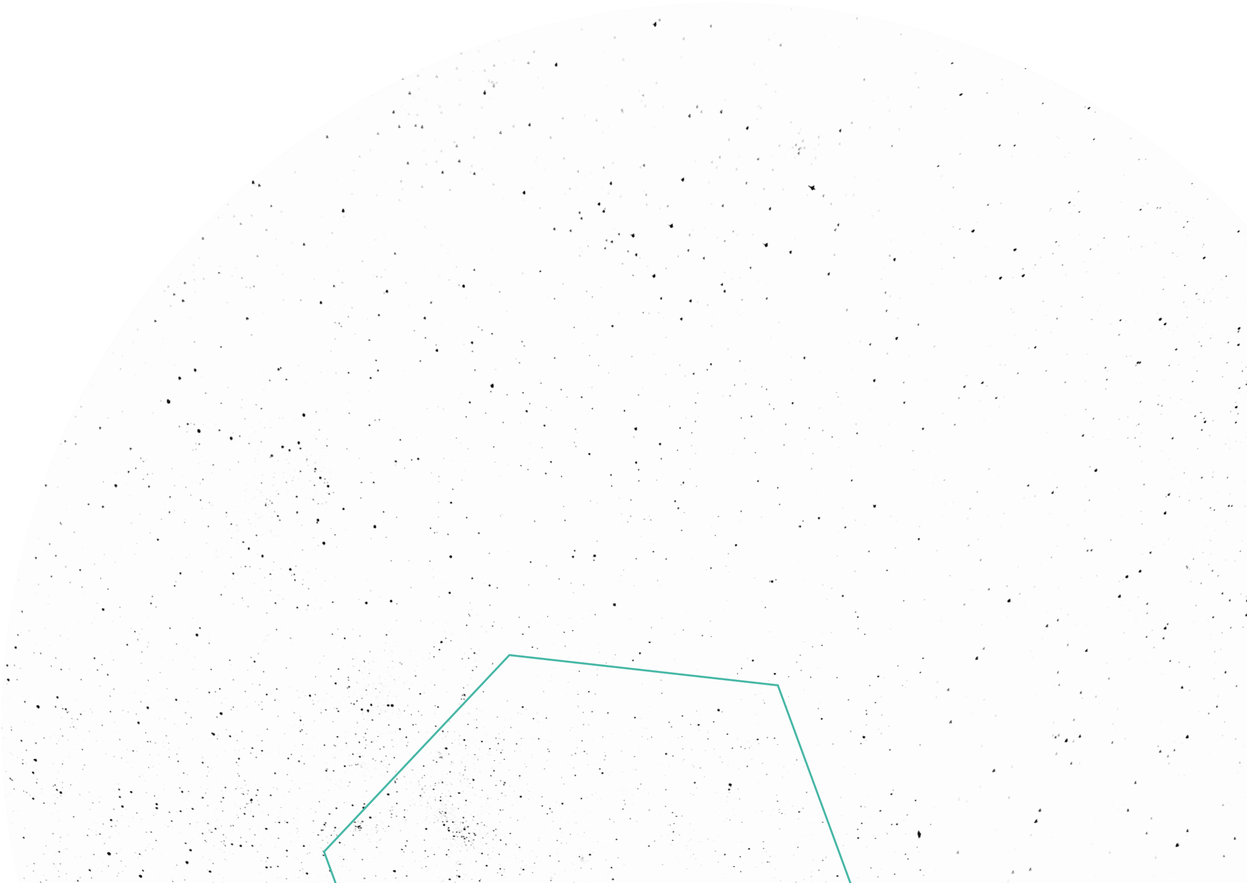
Proper way:

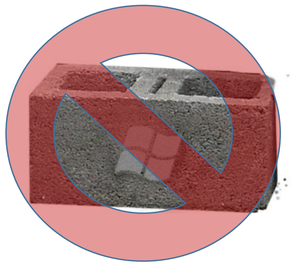
TestGiroKIDSalesInvoicePrivateCustomer
TestGiroKIDSalesInvoiceBusinessCustomer
TestGiroKIDReminder
…

Or even better:

TestElectronicInvoicingPrivateCustomerSalesInvoice
TestElectronicInvoicingBusinessCustomerSalesInvoice

…





Avoiding Unit Test Cement

BC/NAV TECH DAYS 2022
mibuso.com

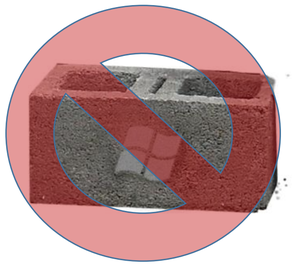
Unit tests should test complex logic

Need to know proper inputs/outputs

Messages in Assert are the key

Person looking at the test failure will most likely not know what the test is about

```
codeunit 80 "Sales-Post"
codeunit 12 "Gen. Jnl.-Post Line“
codeunit 1255 "Match Bank Payments“
…
```



Avoiding Unit Test Cement 2

BC/NAV TECH DAYS 2022
mibuso.com

Unit tests must match method description 1-1

Everything written in method documentation should have a unit test

Every unit test should be reflected in the public method definition

TDD will ensure you test how good the interface is before coding it

```
codeunit 1284 "Password Handler"
{
    /// <summary>
    /// Generates a password that consists of a user-defined number of characters, and meets the <see
cref="IsPasswordStrong"/> conditions.
    /// </summary>
    /// <param name="Length">The number of characters in the password. Passwords must contain at least eight
characters.</param>
    /// <error>The length is less than the minimum defined in <see cref="OnSetMinPasswordLength"/>
event.</error>
    /// <returns>The generated password.</returns>
    procedure GeneratePassword(Length: Integer): Text;
    begin
        exit(PasswordHandlerImpl.GeneratePassword(Length));
    end;

    /// <summary>
    /// Check whether the password meets the following conditions:
    /// - Contains at least the number characters defined by <see cref="OnSetMinPasswordLength"/> event, but it
cannot be less than eight.
    /// - Contains uppercase and lowercase characters, digits, and special characters.
    /// - Does not contain sequences of characters. For example, aaa or 123.
    /// </summary>
    /// <param name="Password">The password to check.</param>
    /// <returns>True if the password meets the conditions for strong passwords.</returns>
    procedure IsPasswordStrong(Password: Text): Boolean;
    begin
        exit(PasswordHandlerImpl.IsPasswordStrong(Password));
    end;
```

No tests? Start with Scenario testing

BC/NAV TECH DAYS 2022
mibuso.com

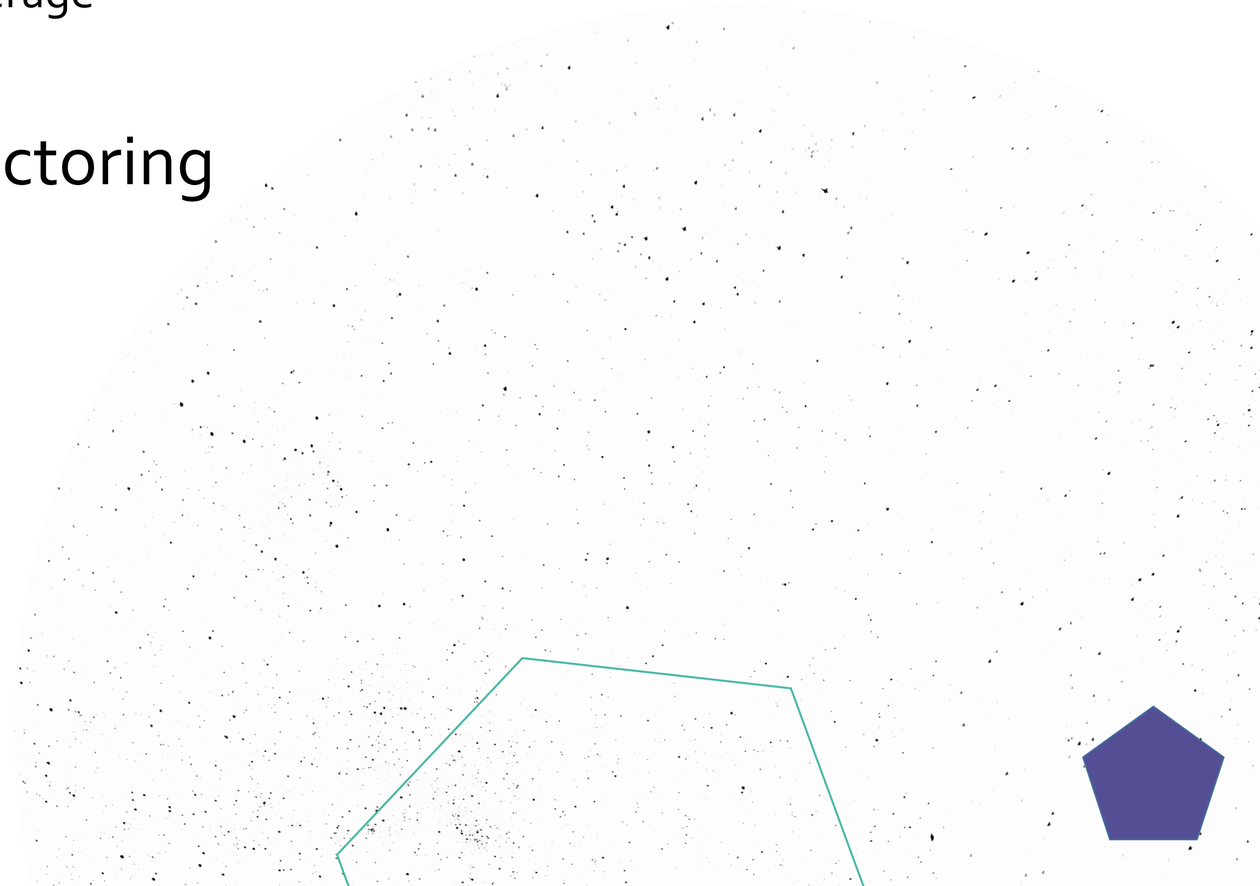
Cover the most important scenarios first

You will automate the documentation

You will document the code

Even few scenario tests will give you a good code coverage

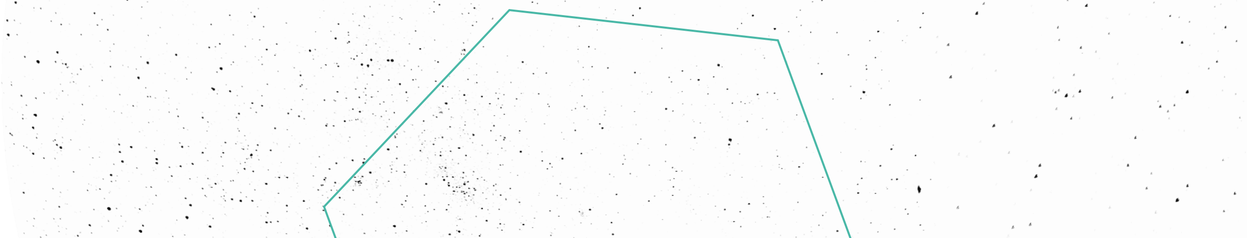
Slowly introduce unit tests with refactoring



TDD Demo

You can pick up these bugs or a new feature. What do you preferer?

Feature of course



Good Tests

Cover the risk

Document the code

Fast to execute

Easy to read

Test one thing

Fast to execute – numbers please?

BC/NAV TECH DAYS 2022
mibuso.com

Test – few seconds, max. 2 minutes

Codeunit – max. 5 minutes. No more than 50 tests.

Test extension – max. 30 minutes

Long runs consume resources. Issues with cross test dependencies (Singleton codeunits…)

Group all long running methods into dedicated test extensions

Single Instance codeunits

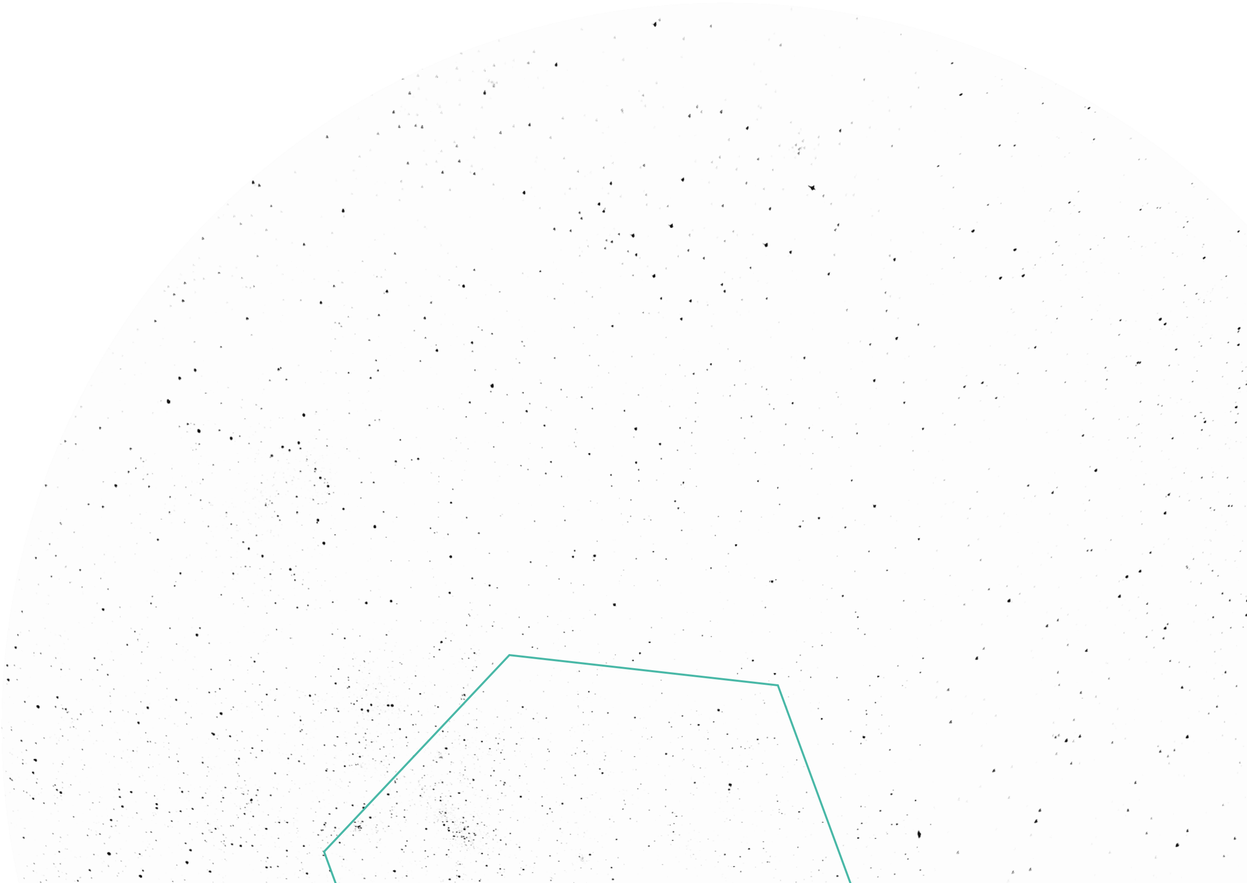
BC/NAV TECH DAYS 2022
mibuso.com

Very difficult to test

Due to Cross-Test dependency

Clear from initialize method

Try to avoid when possible



Good Tests

Cover the risk
Document the code
Fast to execute
Easy to read
Test one thing
Should not modify the environment

Rolling back data – Test isolation

PerCodeunit Recommended, most of the tests

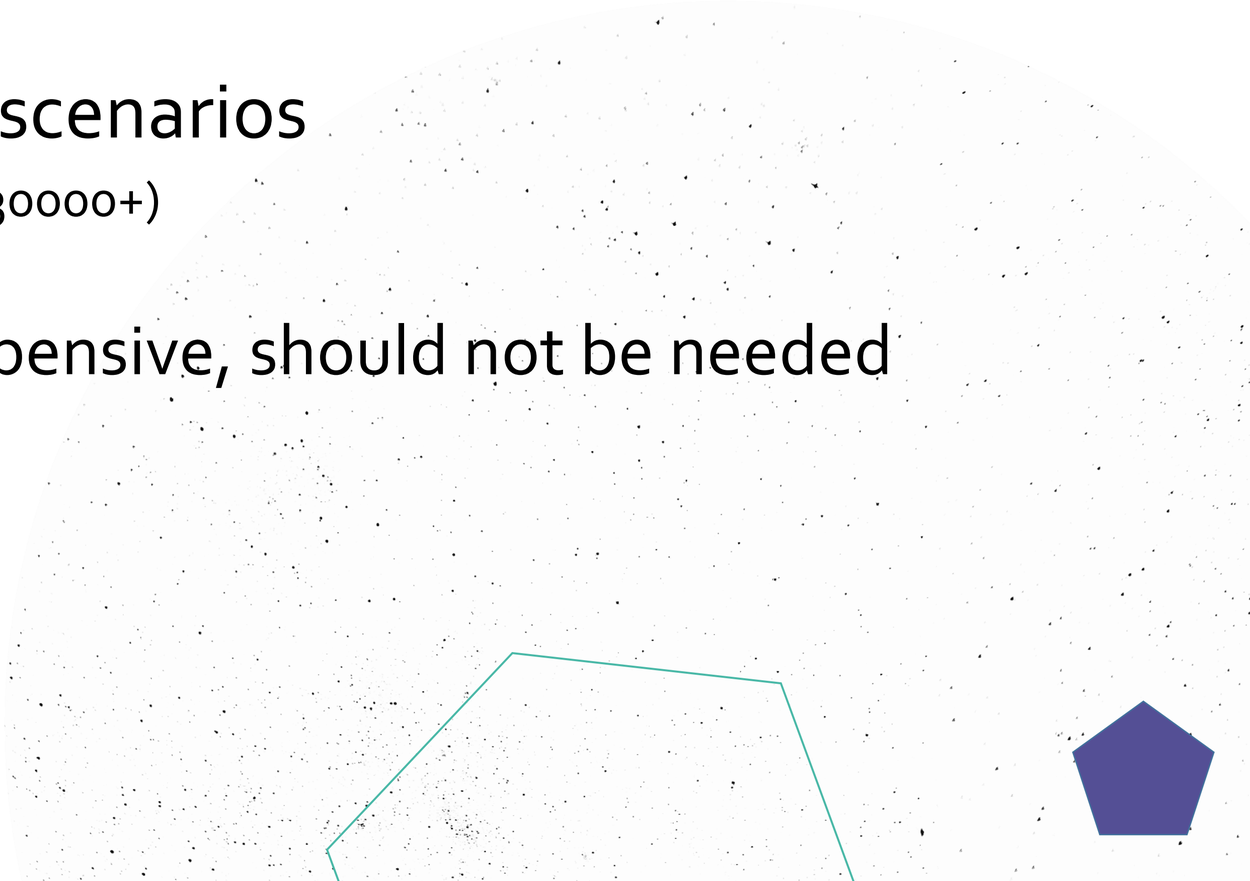
Needed because of - number series issues, different warnings…

Keep the tests simple

Disabled Needed for specific scenarios

Around 200-300 tests (out of 30000+)

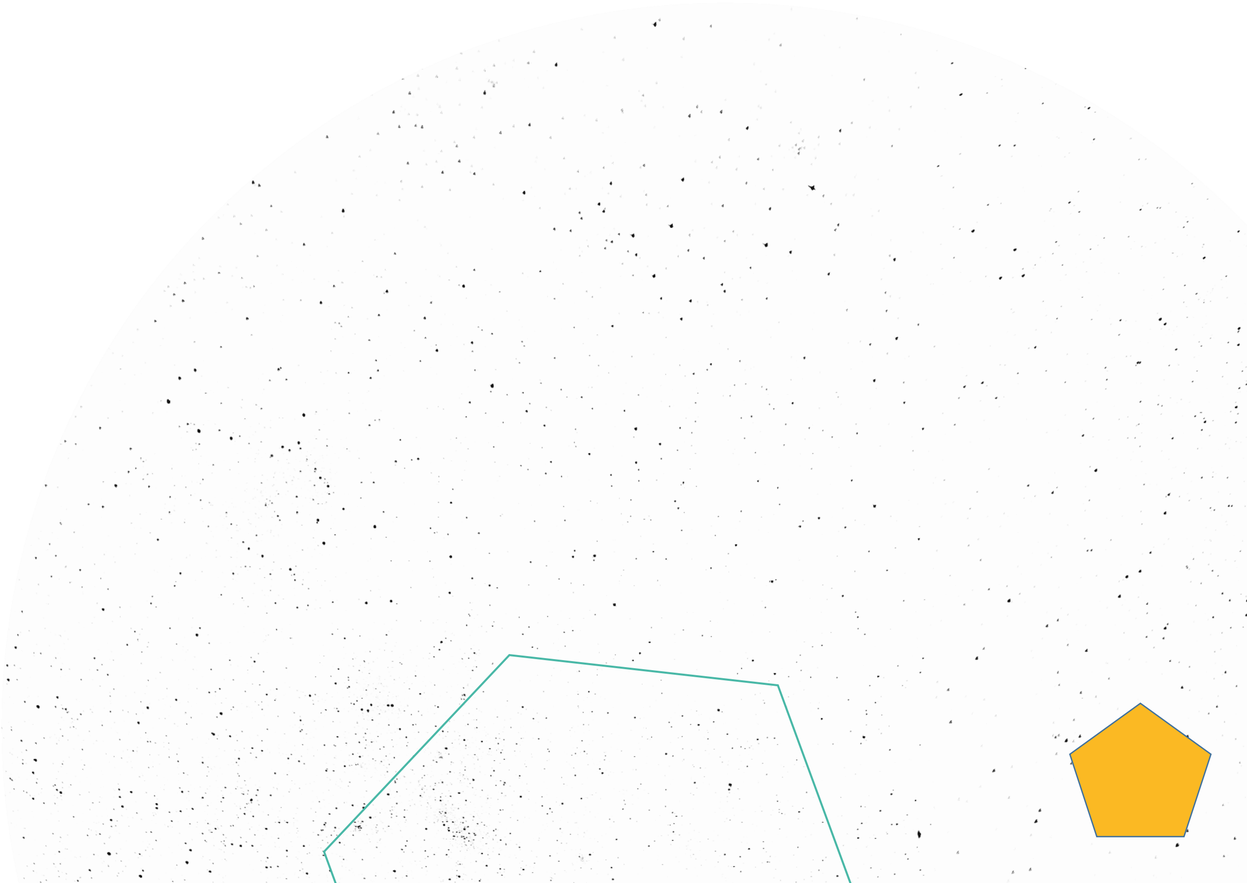
PerMethod Never – Rollback is expensive, should not be needed



Roll back data – TransactionModel:: AutoRollback

When it should be used:

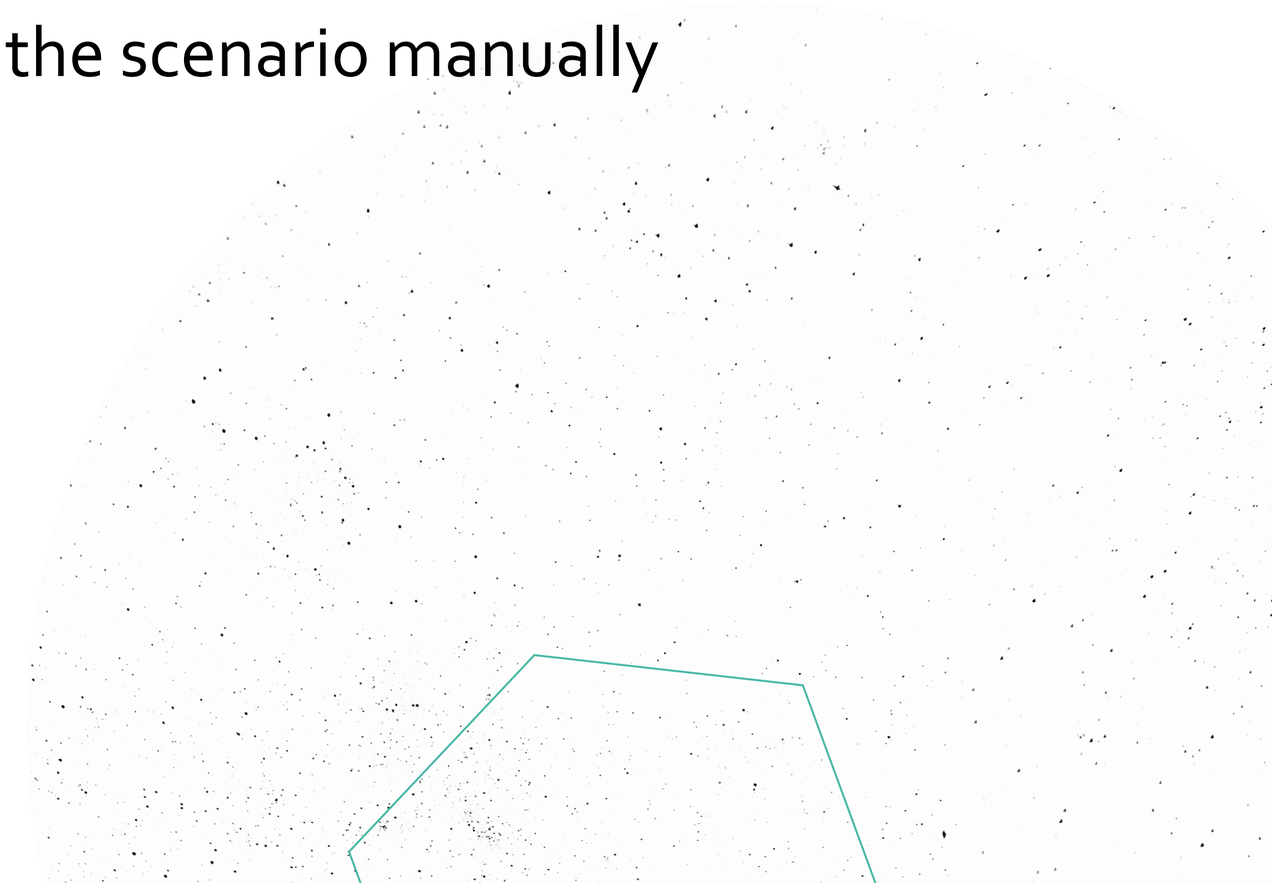
Never



Test Event Subscribers

All Test Event Subscribers should be manual

Could affect other tests
You will need to uninstall tests to verify the scenario manually



Mocking
aka Test Doubling

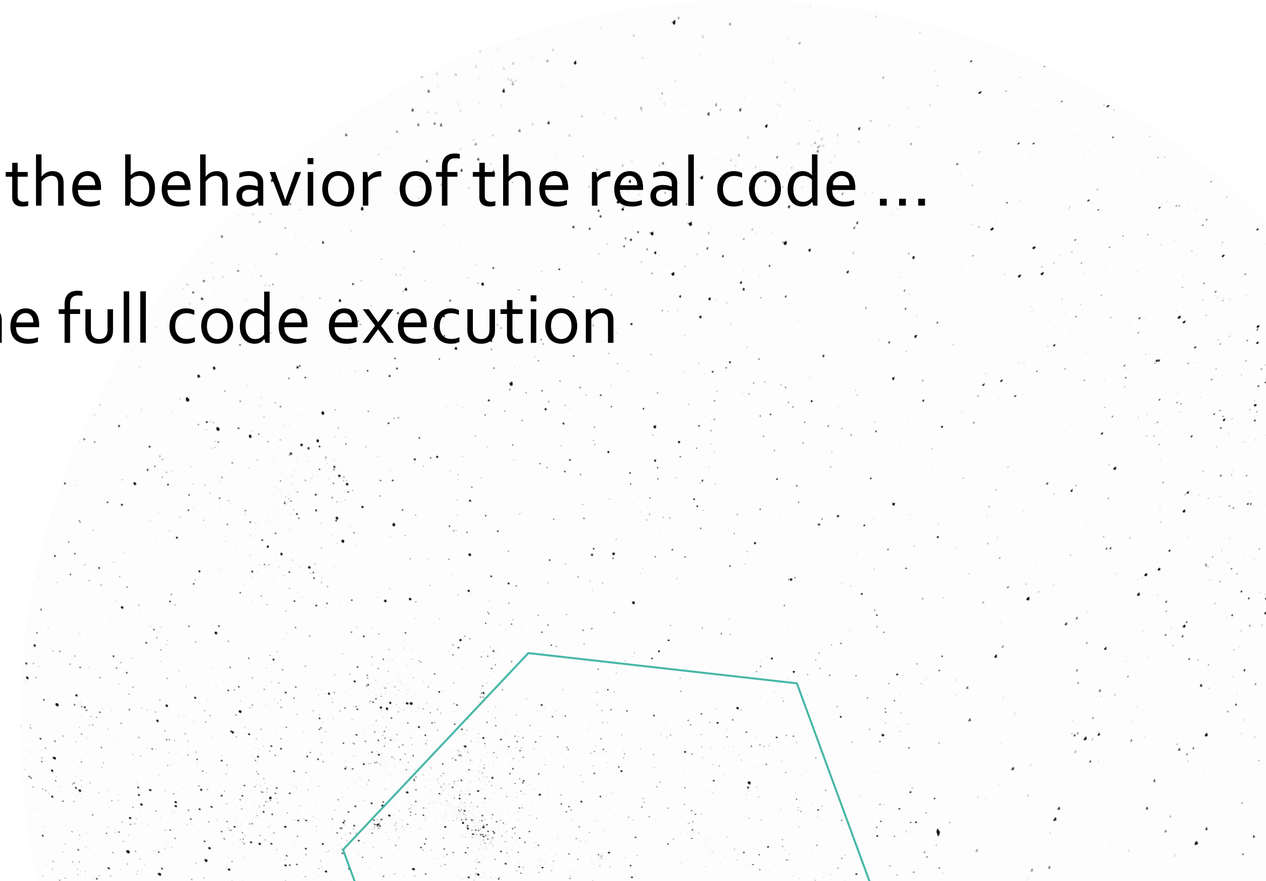
Mocking aka Test Doubling

With mocking we change the code execution flow …

- … by introducing a *stand-in* for – a part – of the real code

Such a stand-in, aka *test double*, mimics the behavior of the real code …

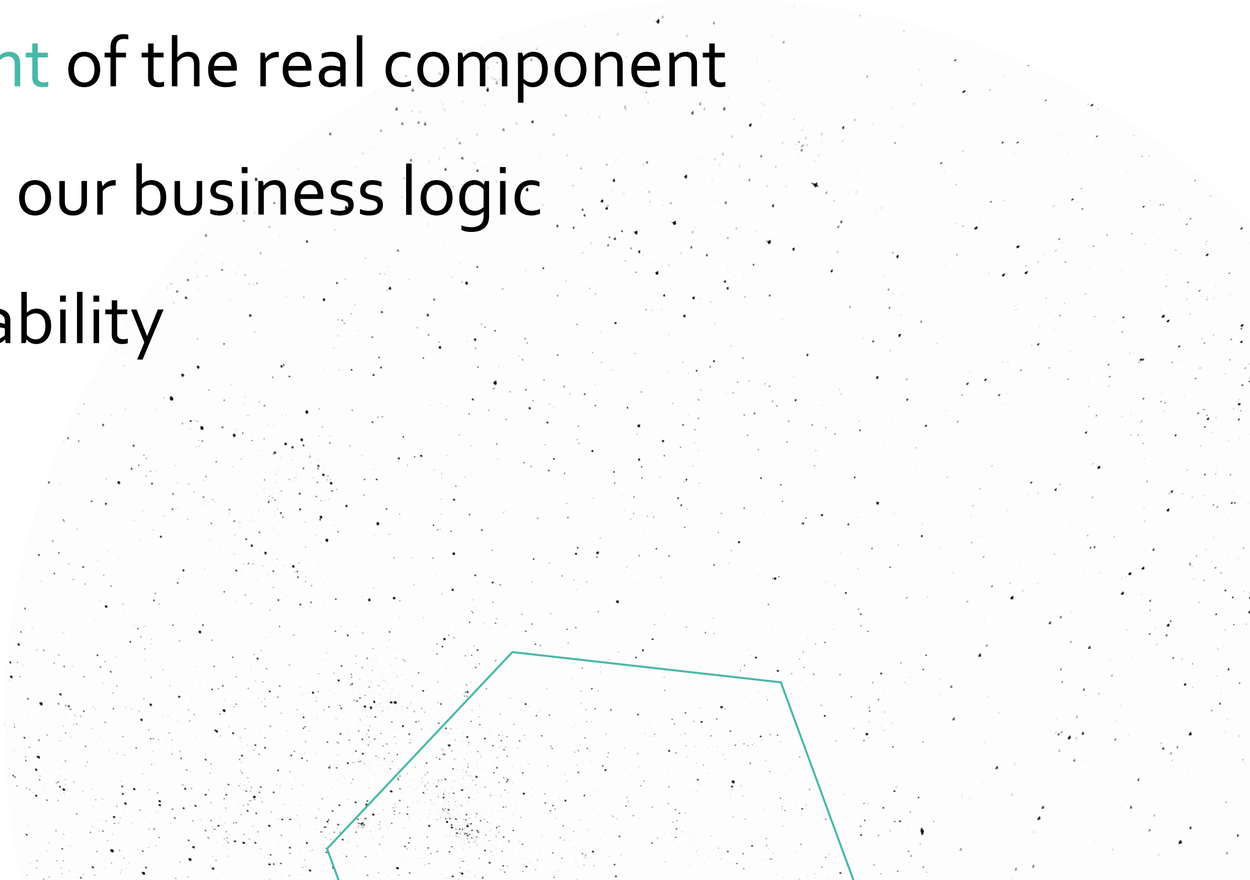
- … as such introducing a shortcut in the full code execution



Mocking aka Test Doubling

A test double is applied to replace calls to external components like web services

- as we want our tests to be independent of the real component
- as we are only interested in validating our business logic
- as we have no control over their availability
- to save cost on paid services
- and also, to save test execution time



Mocking aka Test Doubling

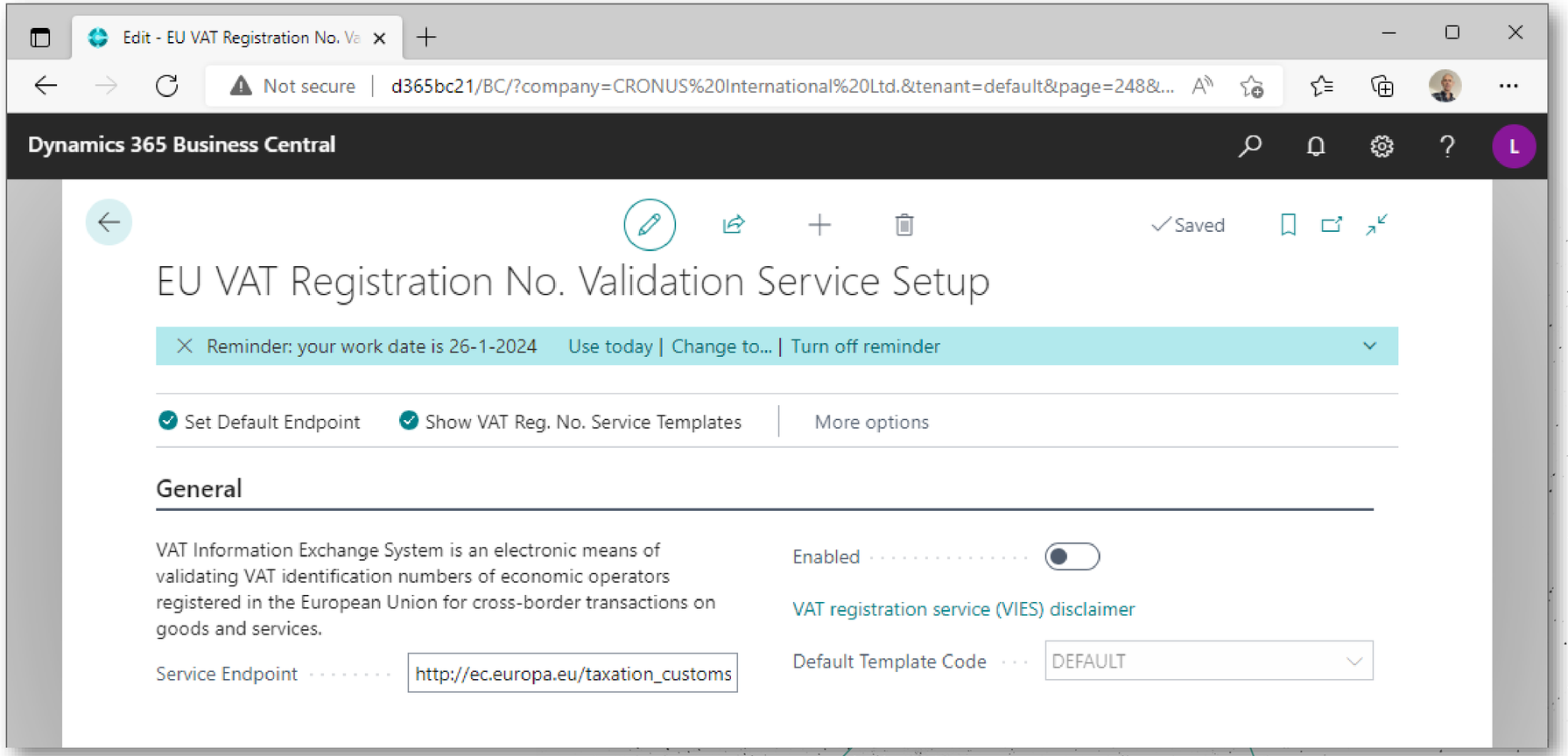
A test double is applied to replace calls to external components like web services

- as we want our tests to be independent of the real component
- as we are only interested in validating our business logic
- as we have no control over their availability
- to save cost on paid services
- and also, to save test execution time – remember that TA should be fast!
- alternatively, it could also be applied to datasets

Test Doubling – example

BC/NAV TECH DAYS 2022
mibuso.com

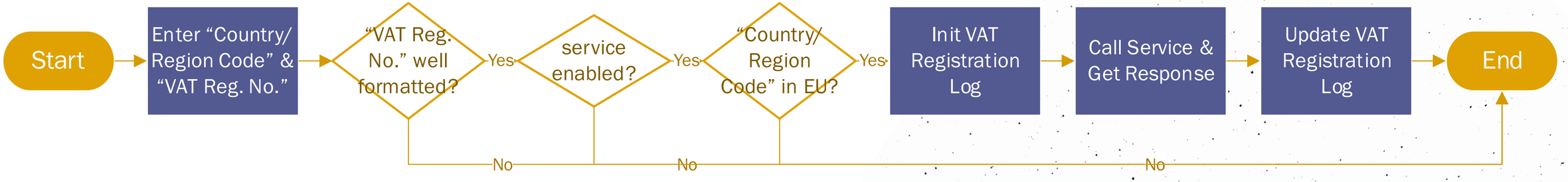
Validation VAT registration number



Test Doubling – example

BC/NAV TECH DAYS 2022
mibuso.com

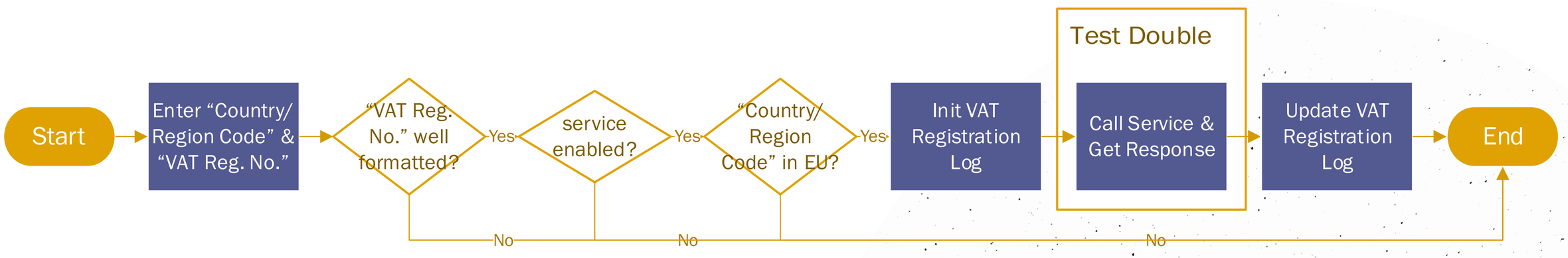
Validation VAT registration number



Test Doubling – example

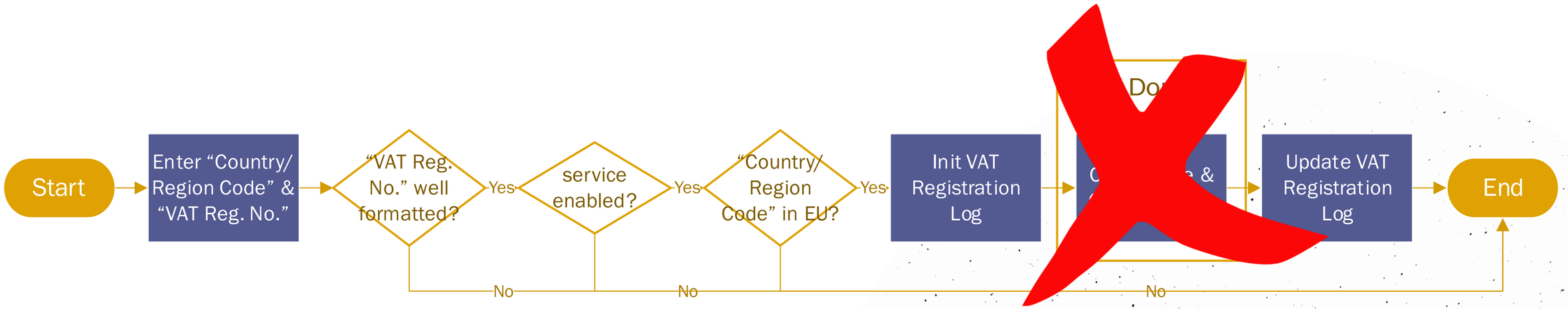
BC/NAV TECH DAYS 2022
mibuso.com

Validation VAT registration number



Test Doubling – bad example

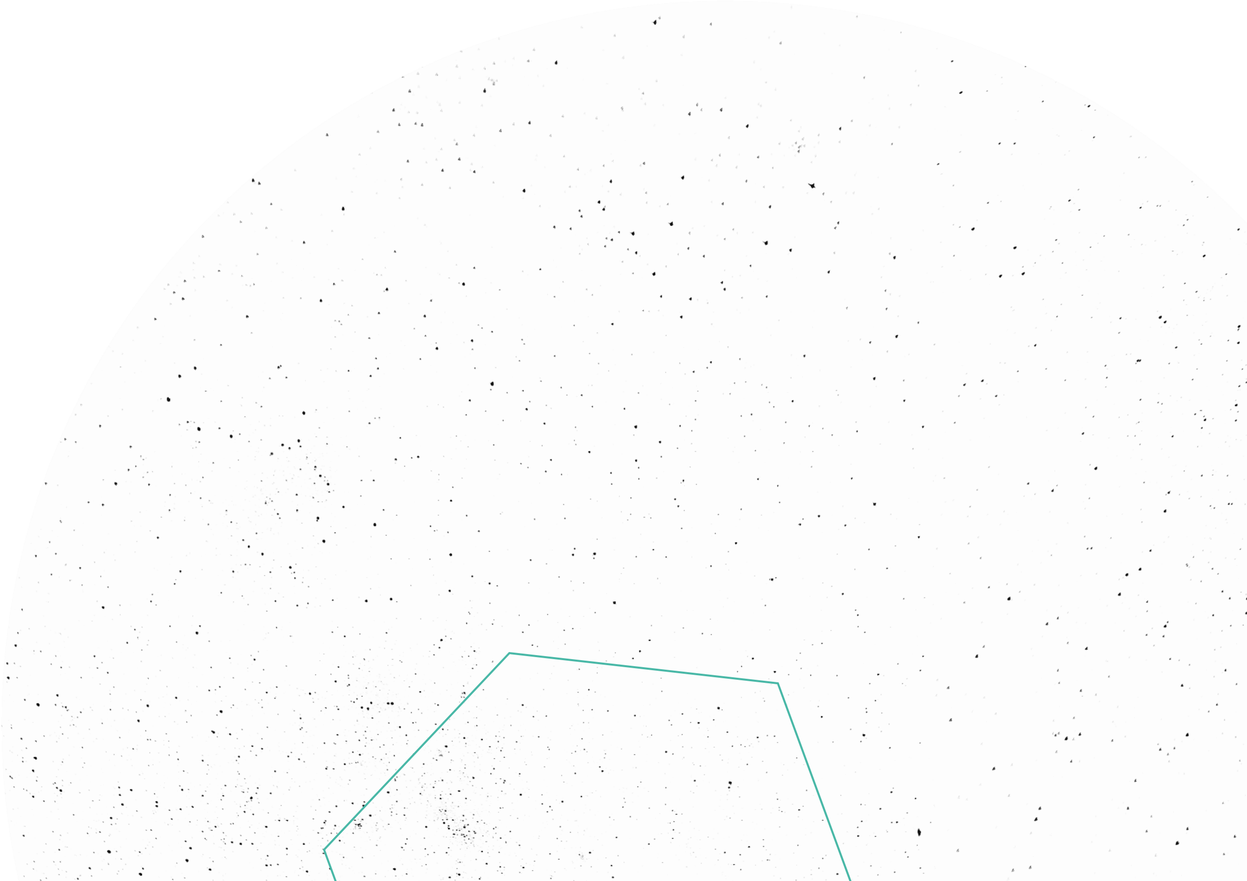
Validation VAT registration number



- Bad, as there is no direct way to replace dependency on service call
- Yes, we can replace service endpoint, but no way to mimic service call

Test Doubling – testable code

The Validation VAT registration number feature code was not written with testability in mind …



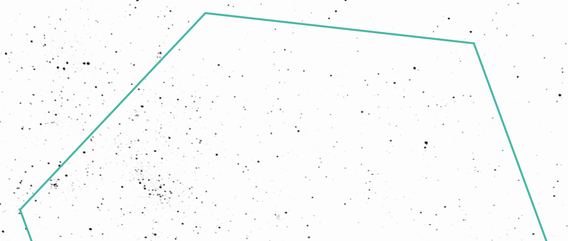
Test Doubling – testable code

The Validation VAT registration number feature code was not written with testability in mind … that is …

… code that can be tested efficiently and effectively using coded tests

Testable code …

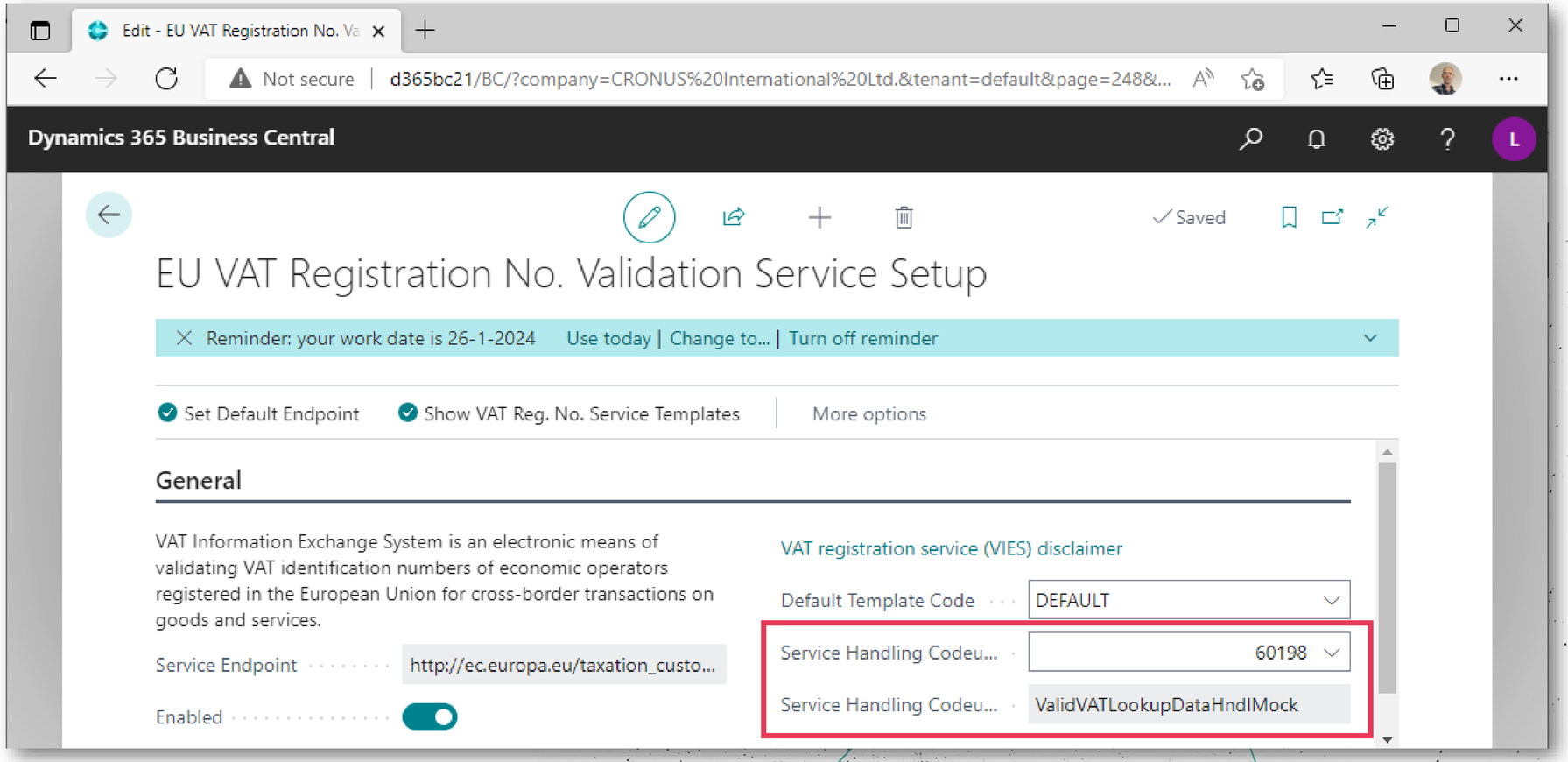
- allows you to have full control over checking its behavior from your test code
- has no direct dependency on other code in your application and outside
- is broken up into loosely coupled units



Test Doubling – testable example

BC/NAV TECH DAYS 2022
mibuso.com

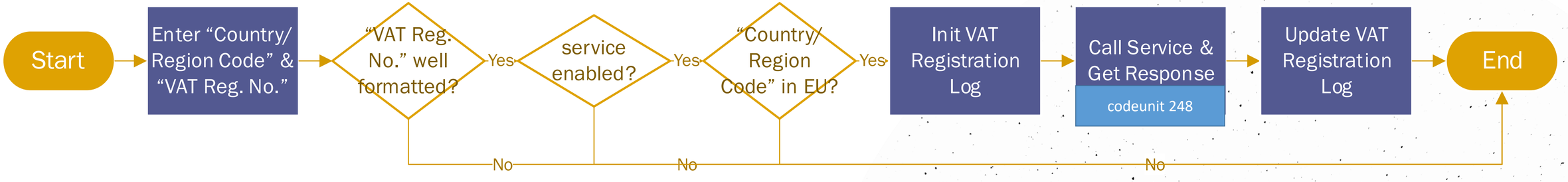
Validation VAT registration number



Test Doubling – testable example

BC/NAV TECH DAYS 2022
mibuso.com

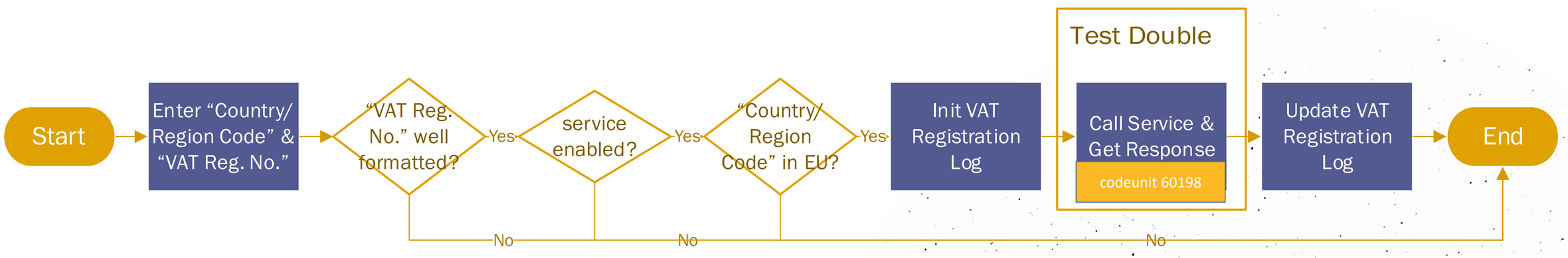
Validation VAT registration number



Test Doubling – testable example

BC/NAV TECH DAYS 2022
mibuso.com

Validation VAT registration number



Test Doubling – testable example

BC/NAV TECH DAYS 2022
mibuso.com

codeunit 248 "VAT Lookup Ext. Data Hndl"

```
trigger OnRun()
begin
    VATRegistrationLog := Rec;

    LookupVatRegistrationFromWebService(true);

    OnRunOnAfterLookupVatRegistrationFromWebService(VATRegistrationLog, Rec);

    Rec := VATRegistrationLog;
end;

local procedure LookupVatRegistrationFromWebService(ShowErrors: Boolean)
begin
    SendRequestToVatRegistrationService(TempBlobRequestBody, ShowErrors);

    InsertLogEntry(TempBlobRequestBody);

    Commit();
end;


local procedure SendRequestToVatRegistrationService(var TempBlobBody: Codeunit "Temp Blob"; ShowErrors: Boolean)
begin
end;

local procedure InsertLogEntry(TempBlobRequestBody: Codeunit "Temp Blob")
begin
end;

[IntegrationEvent(false, false)]
local procedure OnRunOnAfterLookupVatRegistrationFromWebService(
    VATRegistrationLog: Record "VAT Registration Log"; var RecVATRegistrationLog: Record "VAT Registration Log")
begin
end;
```

codeunit 60198 "ValidVATLookupDataHndlMock"

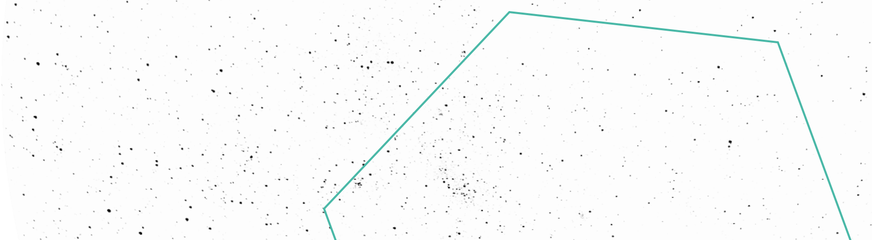
```
// Mock of "VAT Lookup Ext. Data Hndl" returning a valid log entry

trigger OnRun()
begin
    SetValidVATRegistrationLog(Rec);

    Commit(); // To allow for details page to be triggered to open after this
end;

local procedure SetValidVATRegistrationLog(VATRegistrationLog: Record "VAT Registration Log")
begin
    ValidatedName := Any.AlphanumericText(15);
    ValidatedAddress := Any.AlphanumericText(15);
    CreateValidVATCheckResponse(ValidVATResponseDoc, ValidatedName, ValidatedAddress);
    VATRegistrationLogMgt.LogVerification(VATRegistrationLog, ValidVATResponseDoc, NamespaceTxt);
end;

local procedure CreateValidVATCheckResponse(
   var XMLDoc: DotNet XmlDocument; ValidatedName: Text; ValidatedAddress: Text)
begin
end;
```

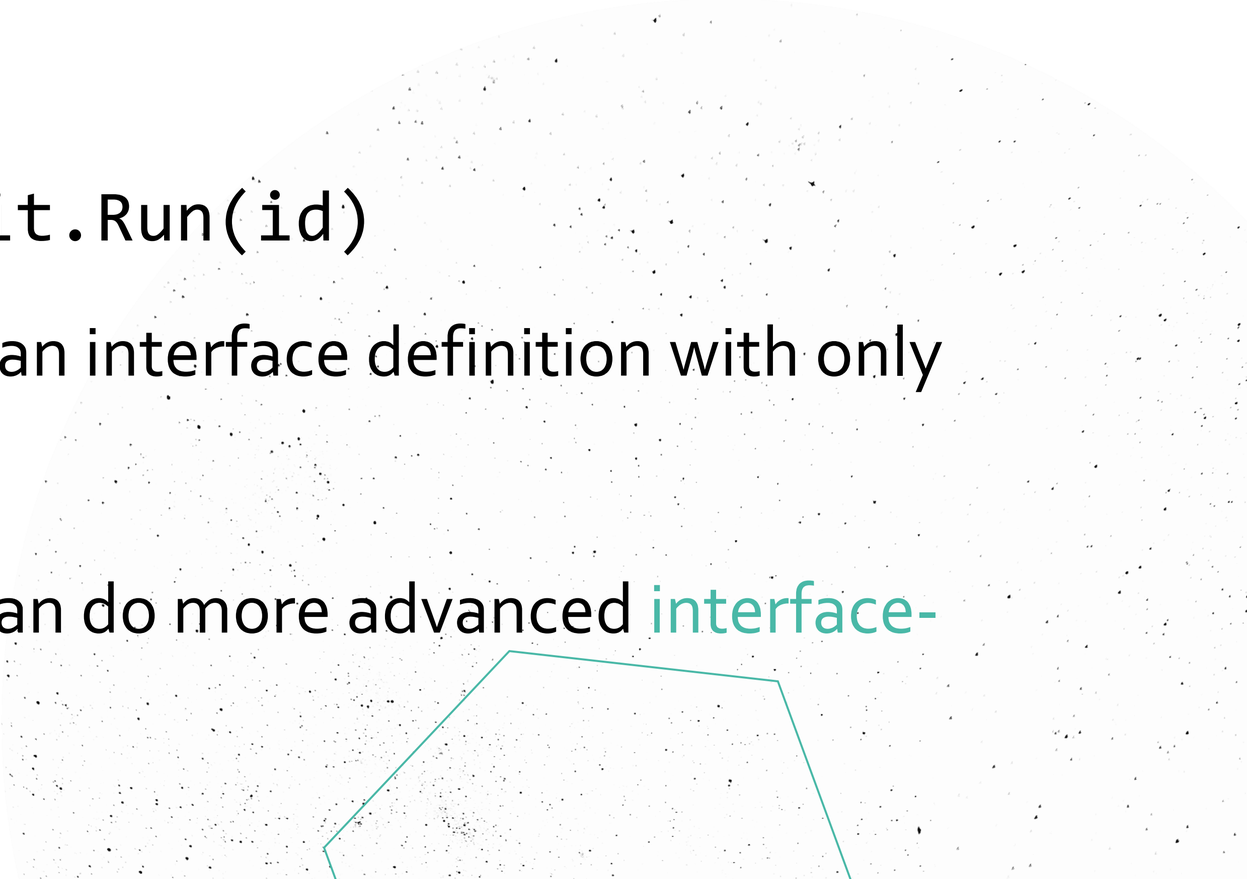


Test Doubling – dependency injection

Technique of replacing a dependency is called dependency injection (DI)

The way applied in the example is called interface-based injection

- replacing one codeunit (id) by another
- let code run based on id using `Codeunit.Run(id)`
- each codeunit is an implementation of an interface definition with only one method: `OnRun`
- having *real* interface objects in AL we can do more advanced interface-based injection



Test Doubling – interface-based injection

BC/NAV TECH DAYS 2022
mibuso.com

Using Enum extension

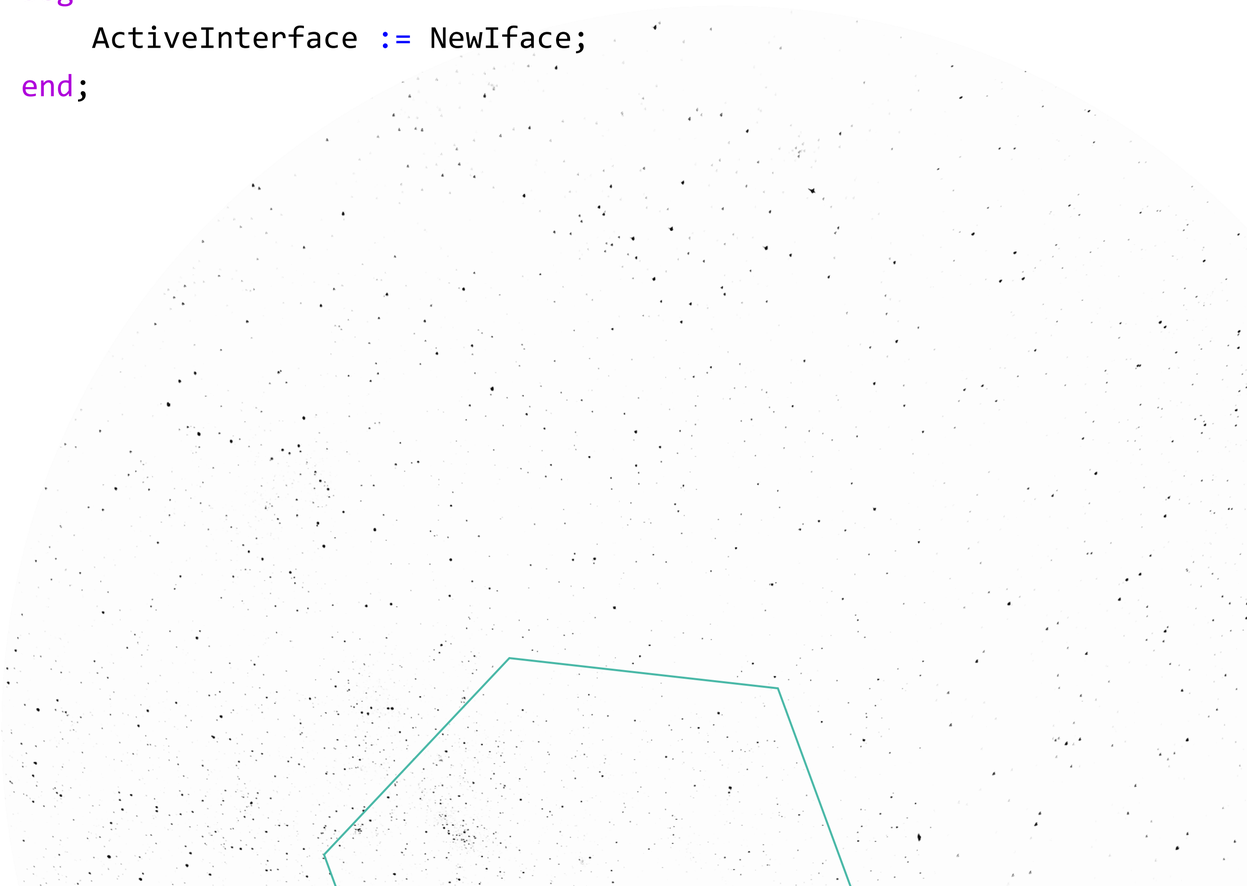
Advanced alternative of `Codeunit.Run(id)`

```
enum xxxx "Interface Provider" implements IInterface
{
    value(0; "First Implementation")
    {
        Implementation = IInterface = "First Implementation";
    }
    value(1; "Second Implementation")
    {
        Implementation = IInterface = "Second Implementation";
    }
}

enumextension yyyy “Stub Iface Prov." extends "Interface Provider"
{
    value(yyyy; "Stub Implementation")
    {
        Implementation = IInterface = "Stub Implementation";
    }
}
```

Using Setter construct

```
procedure SetInterfaceProvider(NewIface: Interface IInterface)
begin
    ActiveInterface := NewIface;
end;
```



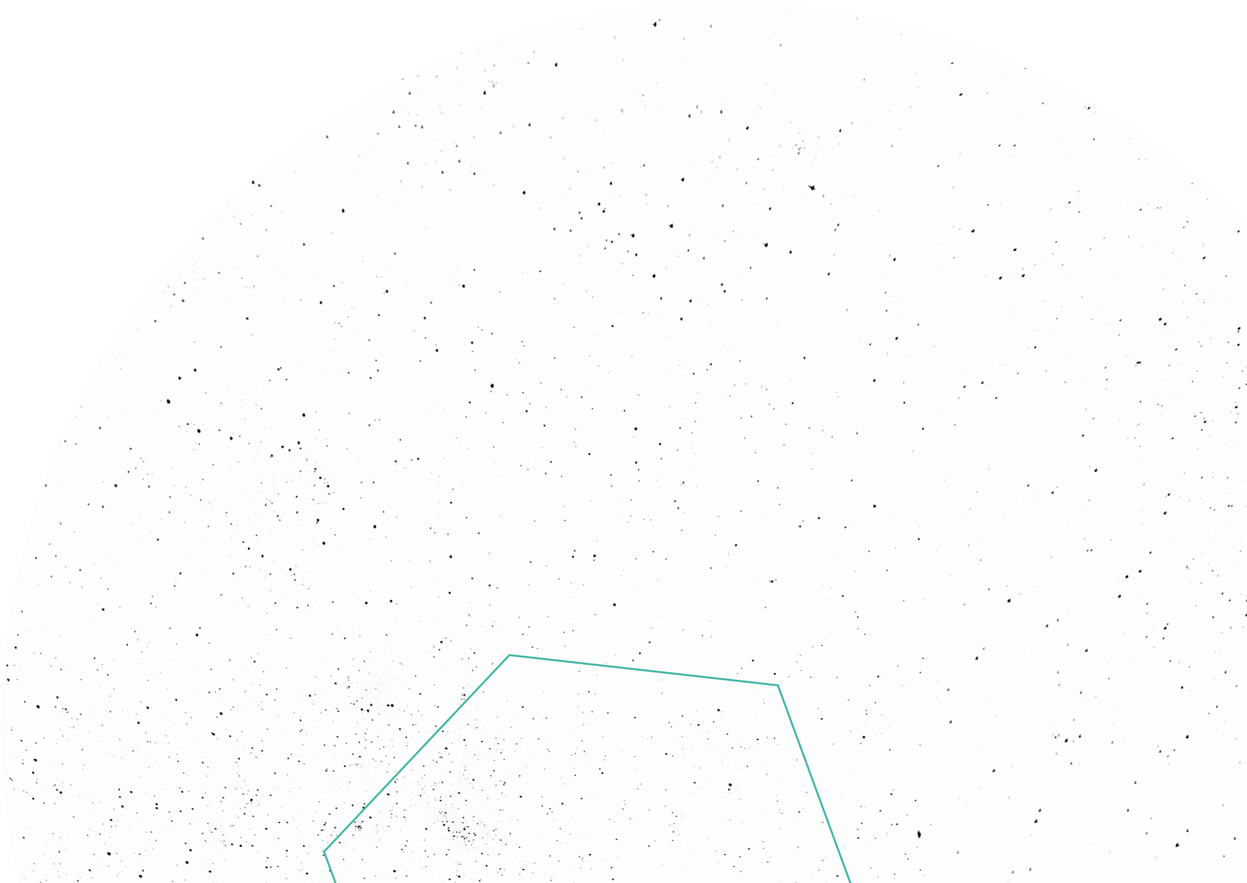
Test Doubling – handled pattern

BC/NAV TECH DAYS 2022
mibuso.com

```
codeunit xxxx "Send Request"
{
    procedure HandleRequest(...)
    begin
        OnBeforeSendRequest(..., IsHandled);
        if IsHandled then
            exit;
        SendRequest(..., IsHandled);
    end;
}

codeunit yyyy “Request Events"
{
    EventSubscriberInstance = Manual;

    [EventSubscriber(ObjectType::Codeunit,
        Codeunit::"Send Request",
        'OnBeforeSendRequest', '', false, false)]
    local procedure OnBeforeSendRequest (
        ...; var IsHandled: Boolean)
    begin
        if IsHandled then
            exit;
        ...
    end;
}
```



Test Doubling – handled pattern

```
codeunit xxxx "Send Request"
{
    procedure HandleRequest(...)
    begin
        OnBeforeSendRequest(..., IsHandled);
        if IsHandled then
            exit;
        SendRequest(..., IsHandled);
    end;
}

codeunit yyyy “Request Events"
{
    EventSubscriberInstance = Manual;

    [EventSubscriber(ObjectType::Codeunit,
        Codeunit::"Send Request",
        'OnBeforeSendRequest', '', false, false)]
    local procedure OnBeforeSendRequest (
        ...; var IsHandled: Boolean)
    begin
        if IsHandled then
            exit;
        ...
    end;
}
```

```
codeunit zzzz “Test Send Request"
{
    Subtype = Test;

    [Test]
    procedure SendRequest()
    var
        RequestEvents : Codeunit "Request Events";
    begin
        BindSubscription(RequestEvents);
        ...
        HandleRequest(...)
        ...
        UnbindSubscription(RequestEvents);
    end;
}
```

Test Doubling – handled pattern

BC/NAV TECH DAYS 2022
mibuso.com

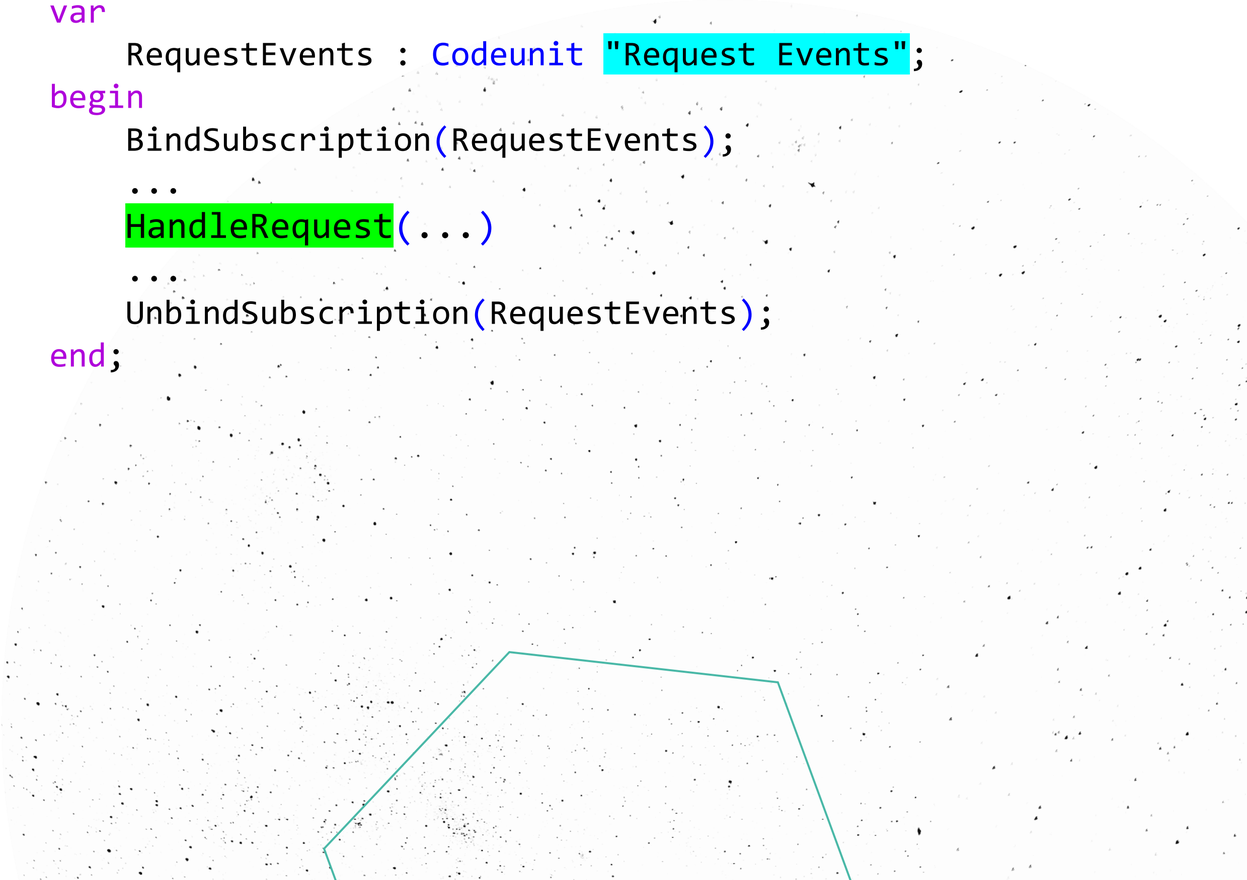
```
codeunit xxxx "Send Request"
{
    procedure HandleRequest(...)
    begin
        OnBeforeSendRequest(..., IsHandled);
        if IsHandled then
            exit;
        SendRequest(..., IsHandled);
    end;
}

codeunit yyyy "Request Events"
{
    EventSubscriberInstance = Manual;

    [EventSubscriber(ObjectType::Codeunit,
        Codeunit::"Send Request",
        'OnBeforeSendRequest', '', false, false)]
    local procedure OnBeforeSendRequest (
        ...; var IsHandled: Boolean)
    begin
        if IsHandled then
            exit;
        ...
    end;
}
```

```
codeunit zzzz "Test Send Request"
{
    Subtype = Test;

    [Test]
    procedure SendRequest()
    var
        RequestEvents : Codeunit "Request Events";
    begin
        BindSubscription(RequestEvents);
        ...
        HandleRequest(...)
        ...
        UnbindSubscription(RequestEvents);
    end;
}
```

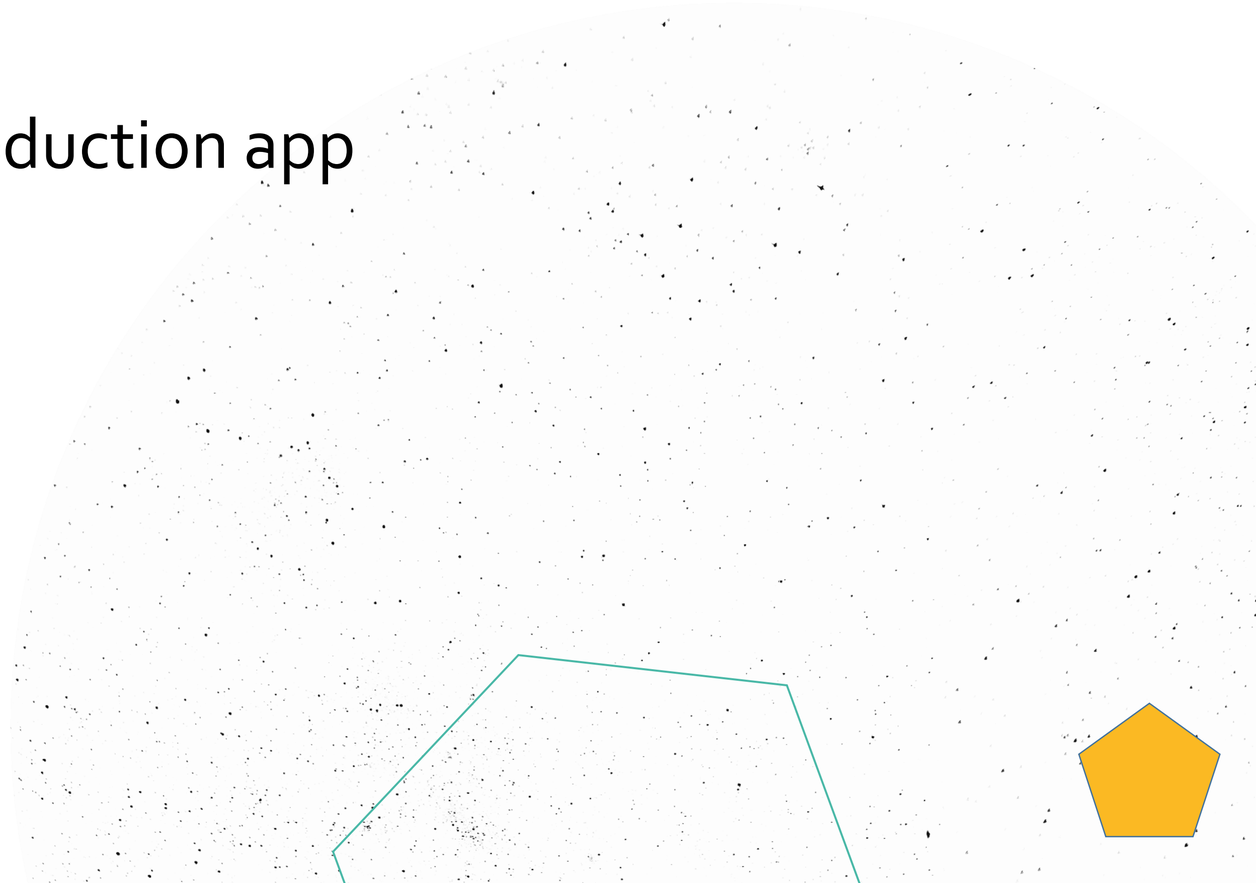


Testability

Code smell – Code should be testable

But may be needed…

Recommended – Compile out of the production app



Good Tests

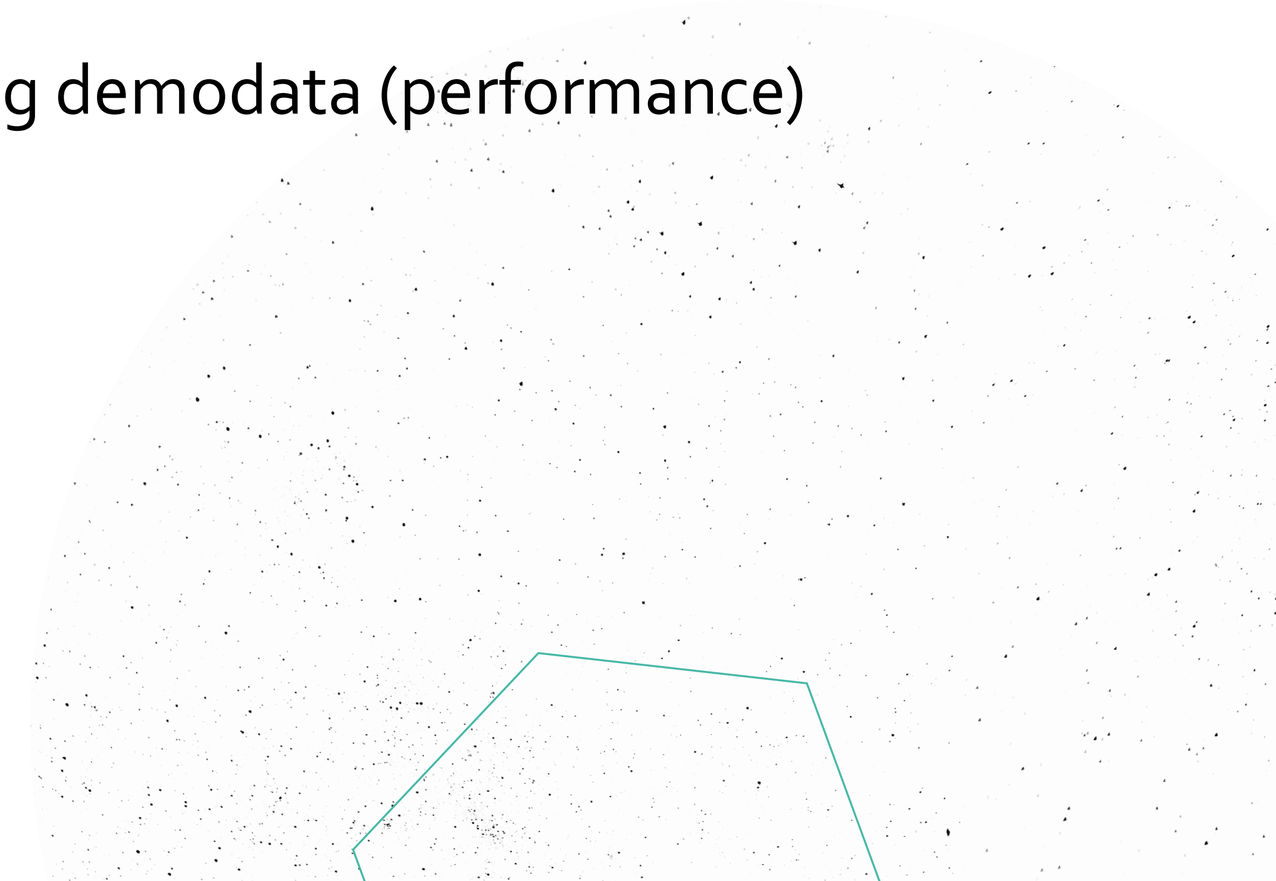
Cover the risk
Document the code
Fast to execute
Easy to read
Test one thing
Should not modify the environment
Data agnostic

Data Agnostic

Create all data that is possible (fast)

- Reuse libraries
- Use Pseudo-Random (Library Any)

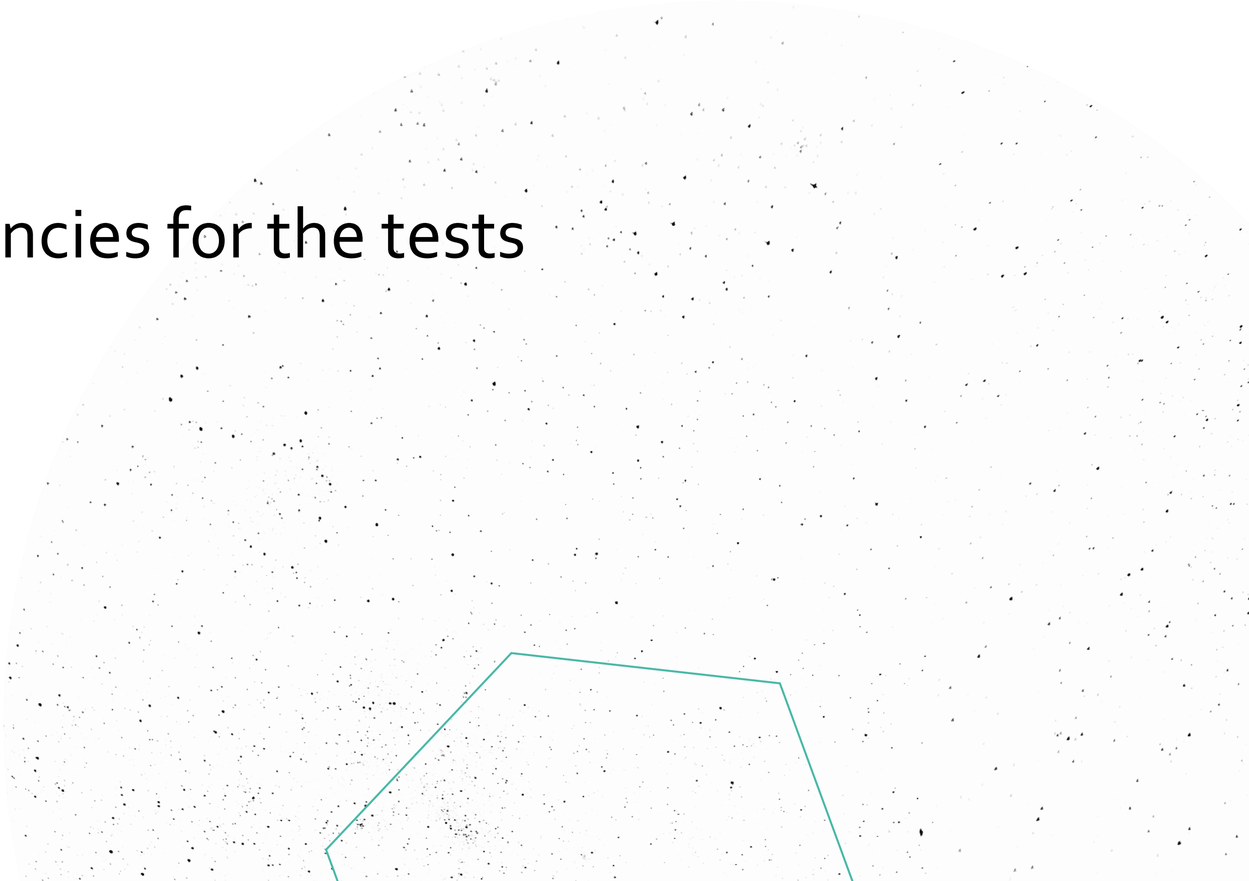
Need to take dependency to the existing demodata (performance)



Demo data

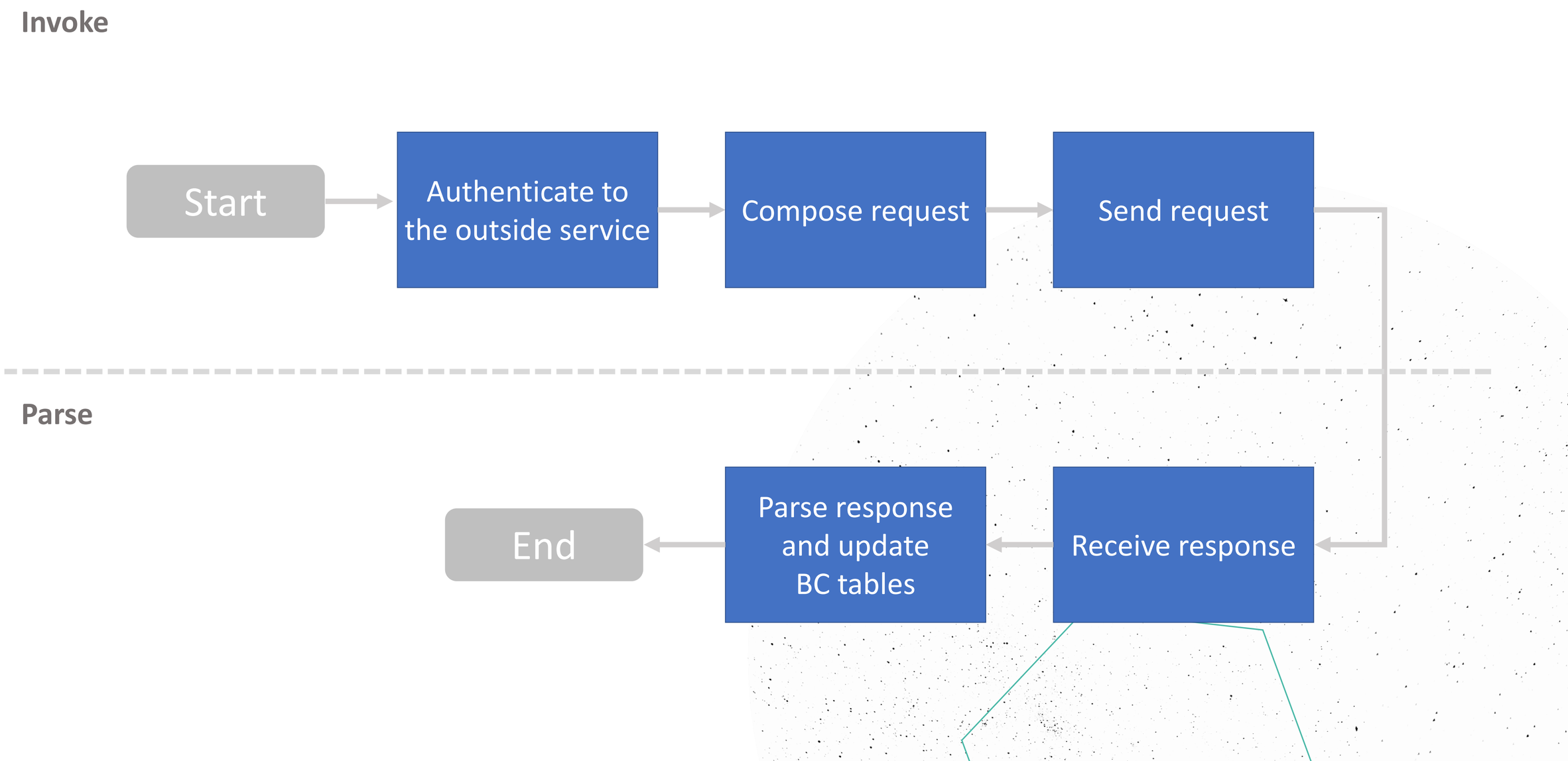
Extended – Cronus OnPrem (Most of the tests depend on this demo data)
Evaluation – Cronus SaaS
Standard – My Company

Pending: Release the list of the dependencies for the tests

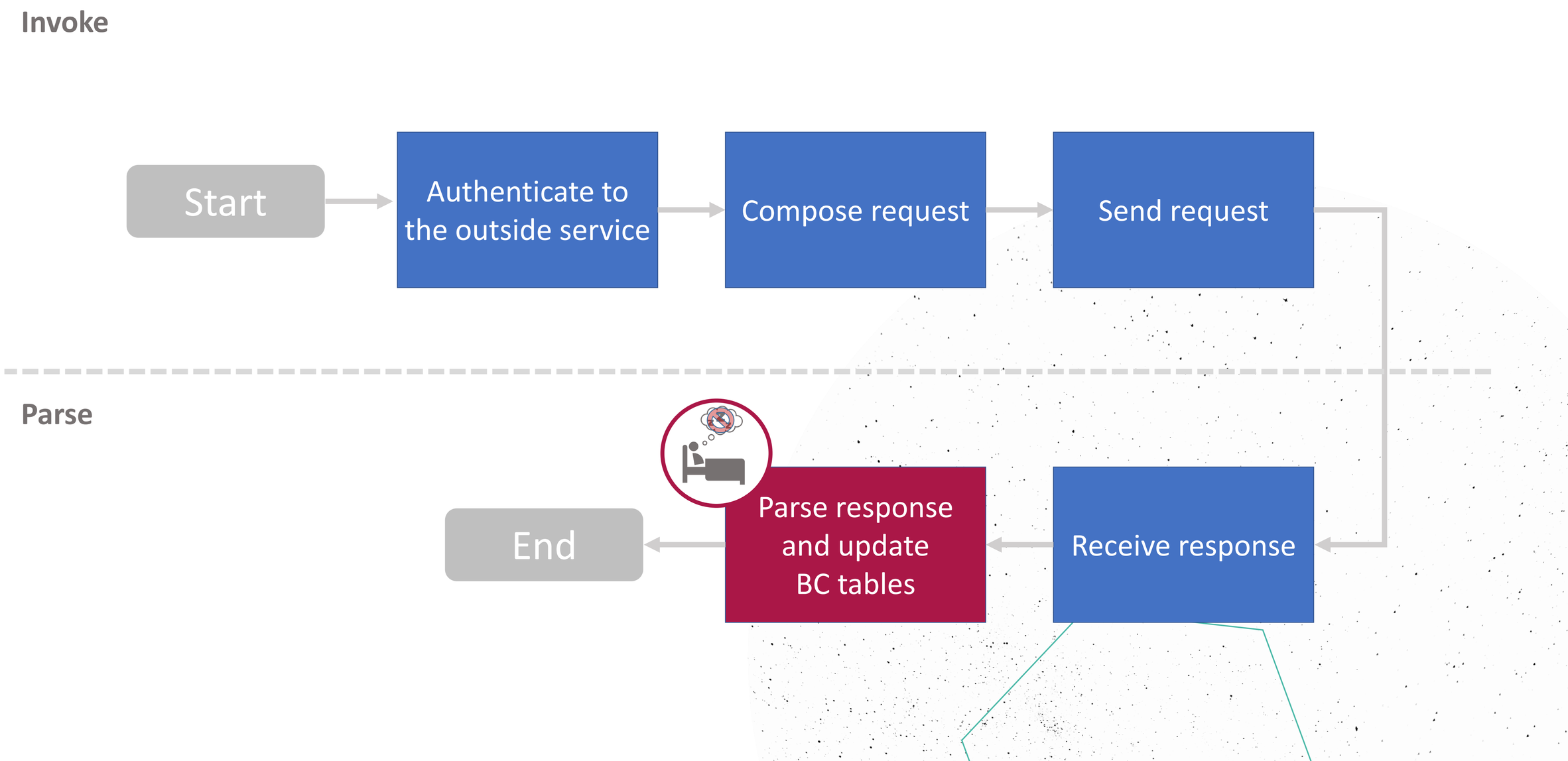


Testing integrations

Testing integrations



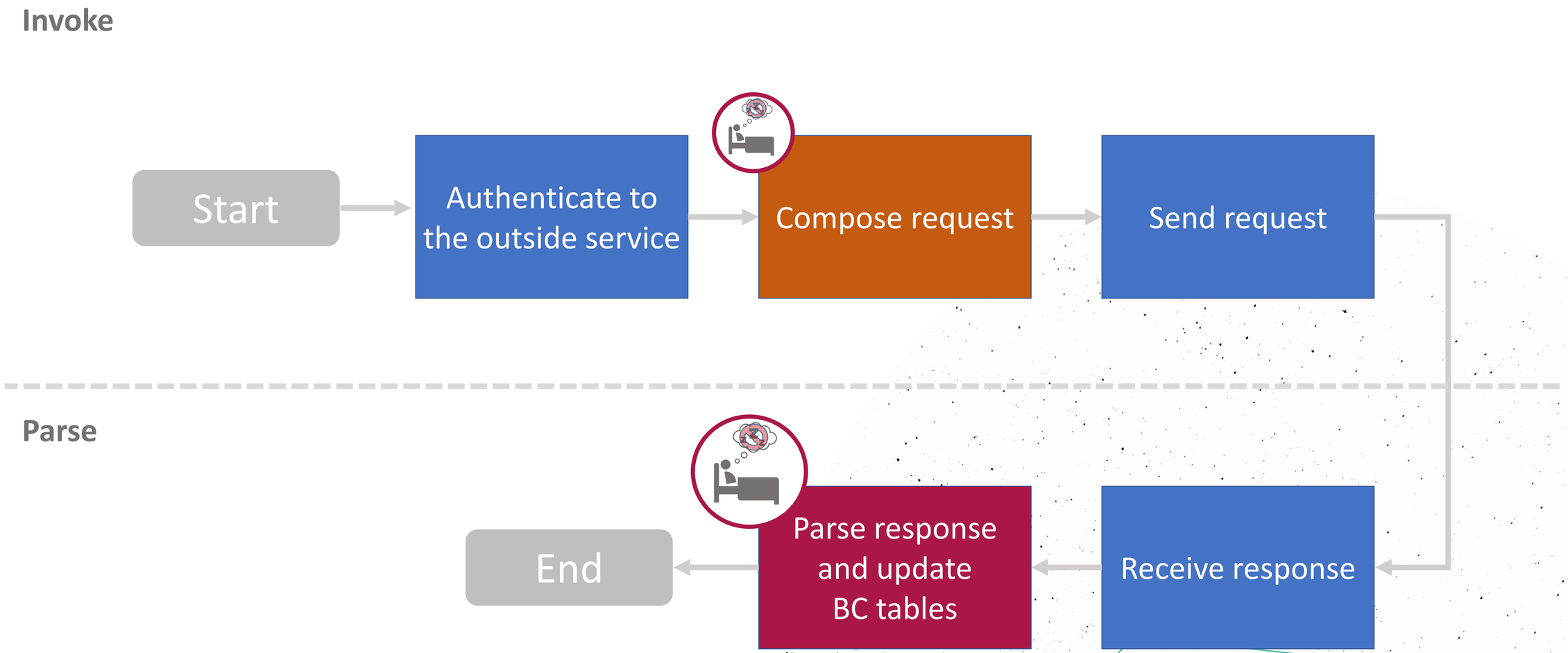
Testing integrations



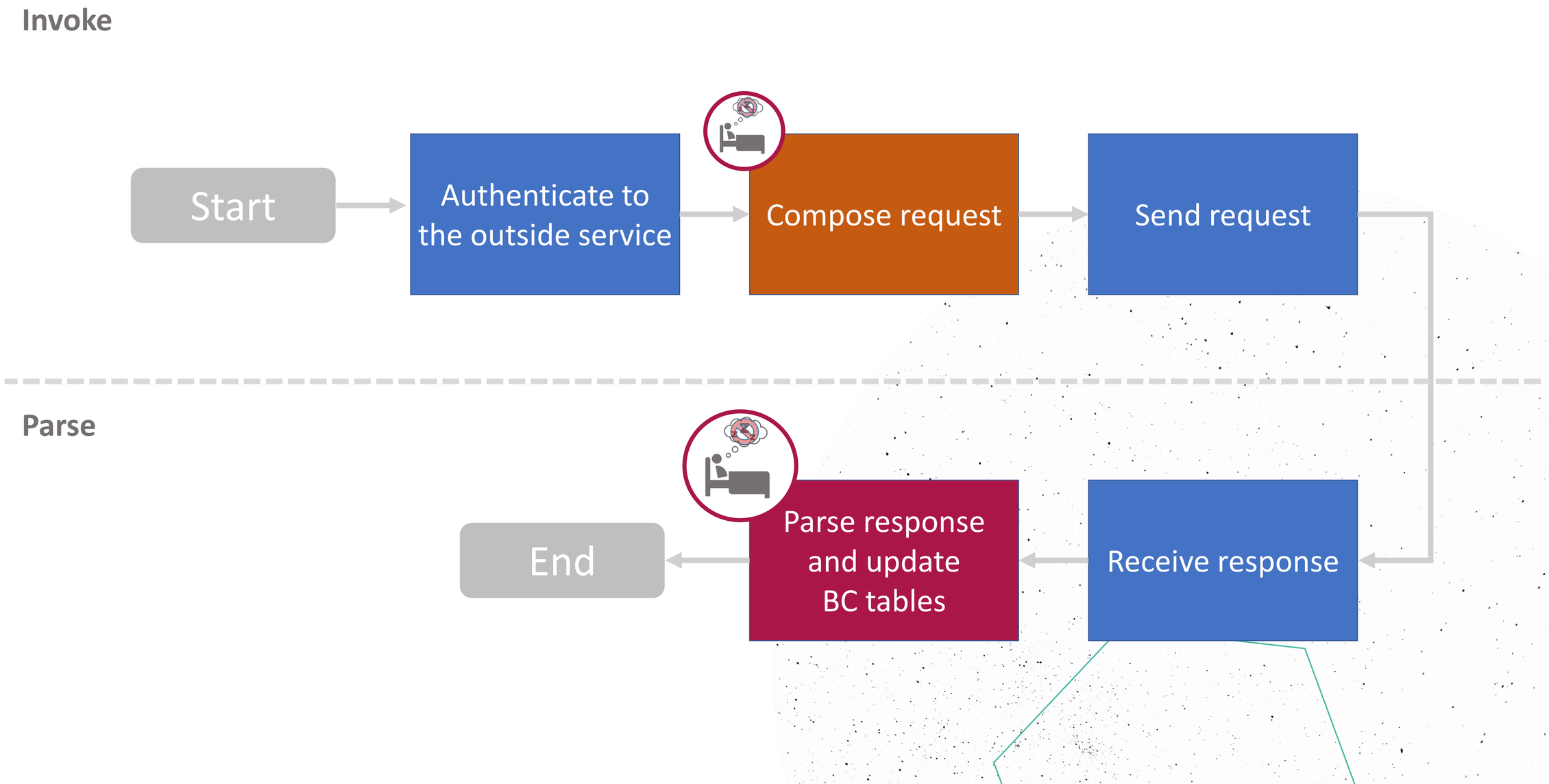
BC/NAV TECH DAYS 2022
mibuso.com

Testing integrations the simple way
Code example

Testing integrations



Testing integrations

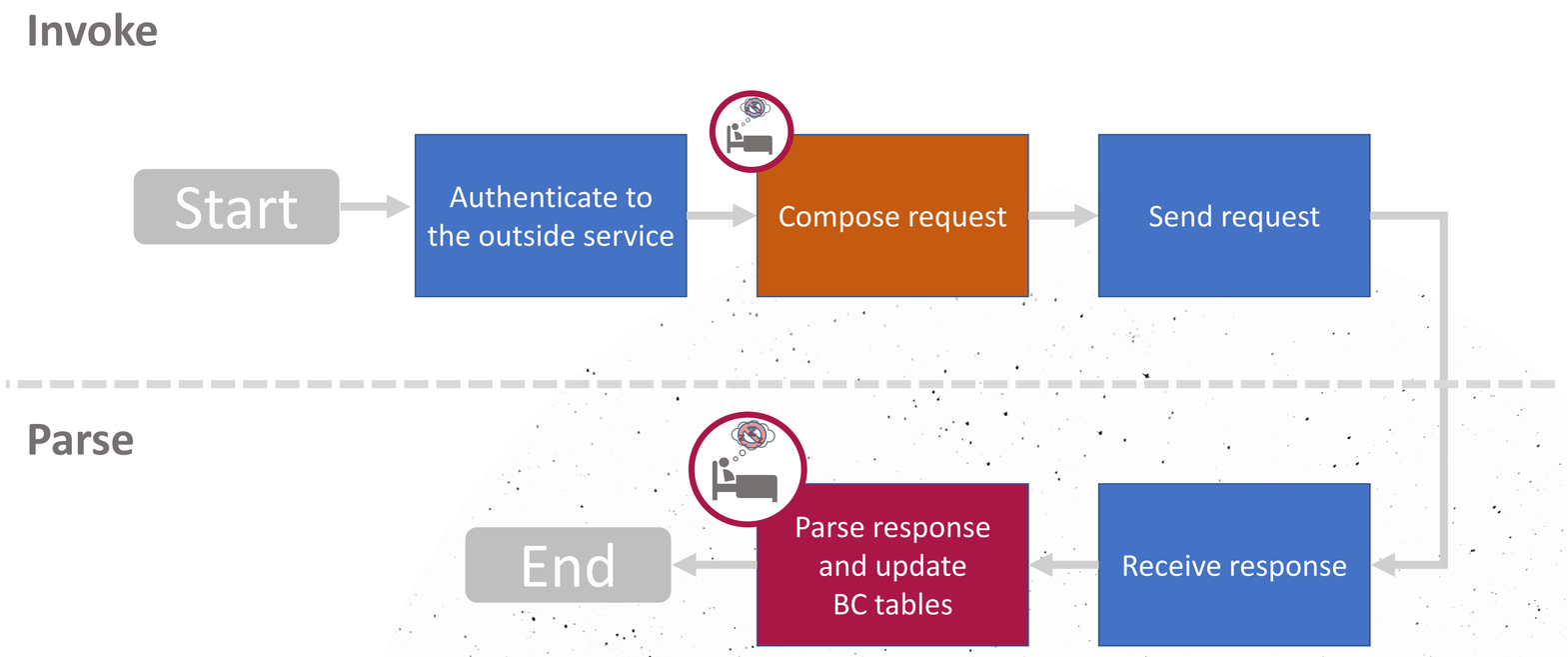


Testing integrations

Manual or
Semi-Automated
E2E tests

Don’t call actual service from automated tests

MOQ or similar framework if you need to automate E2E

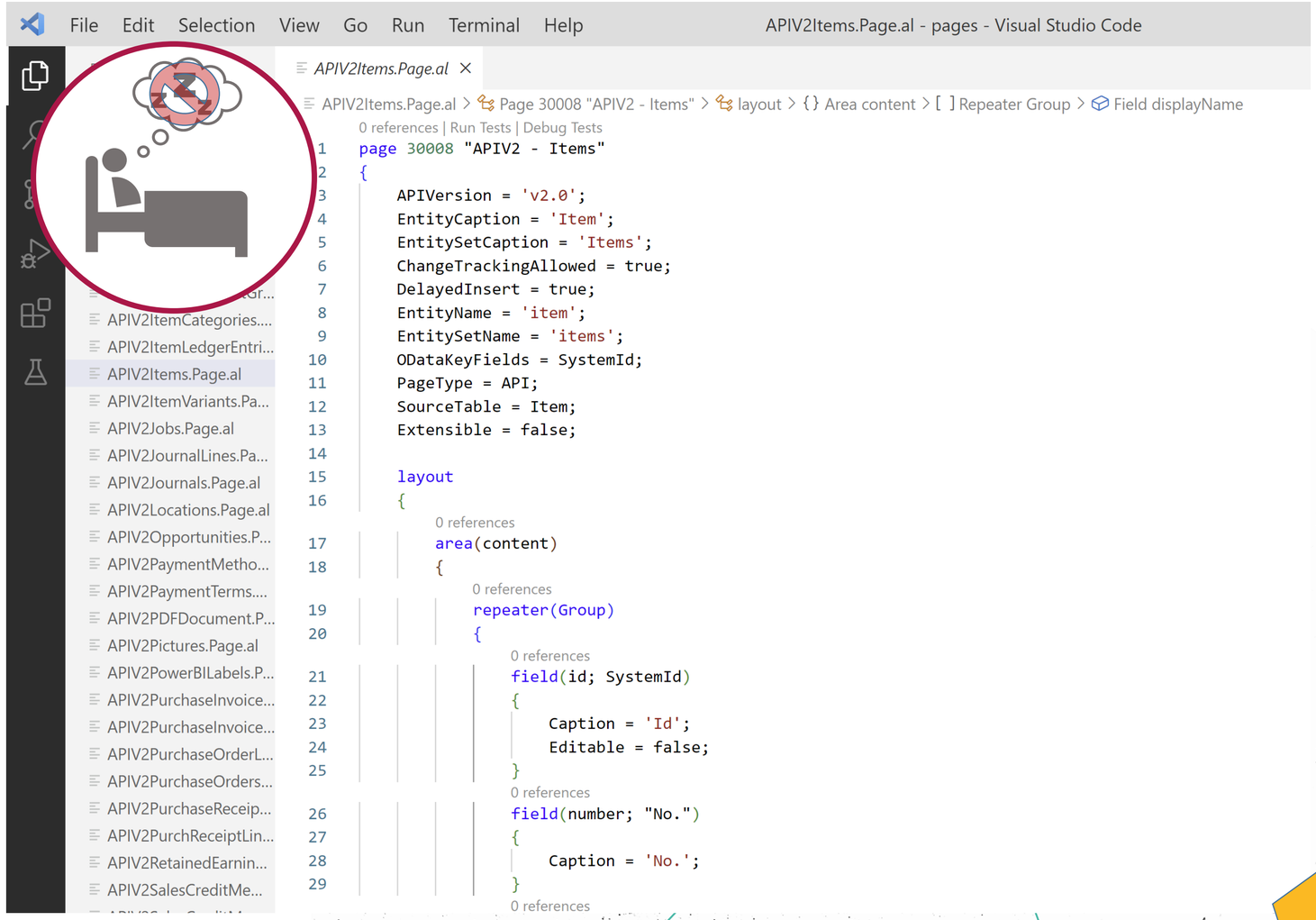


Testing APIs

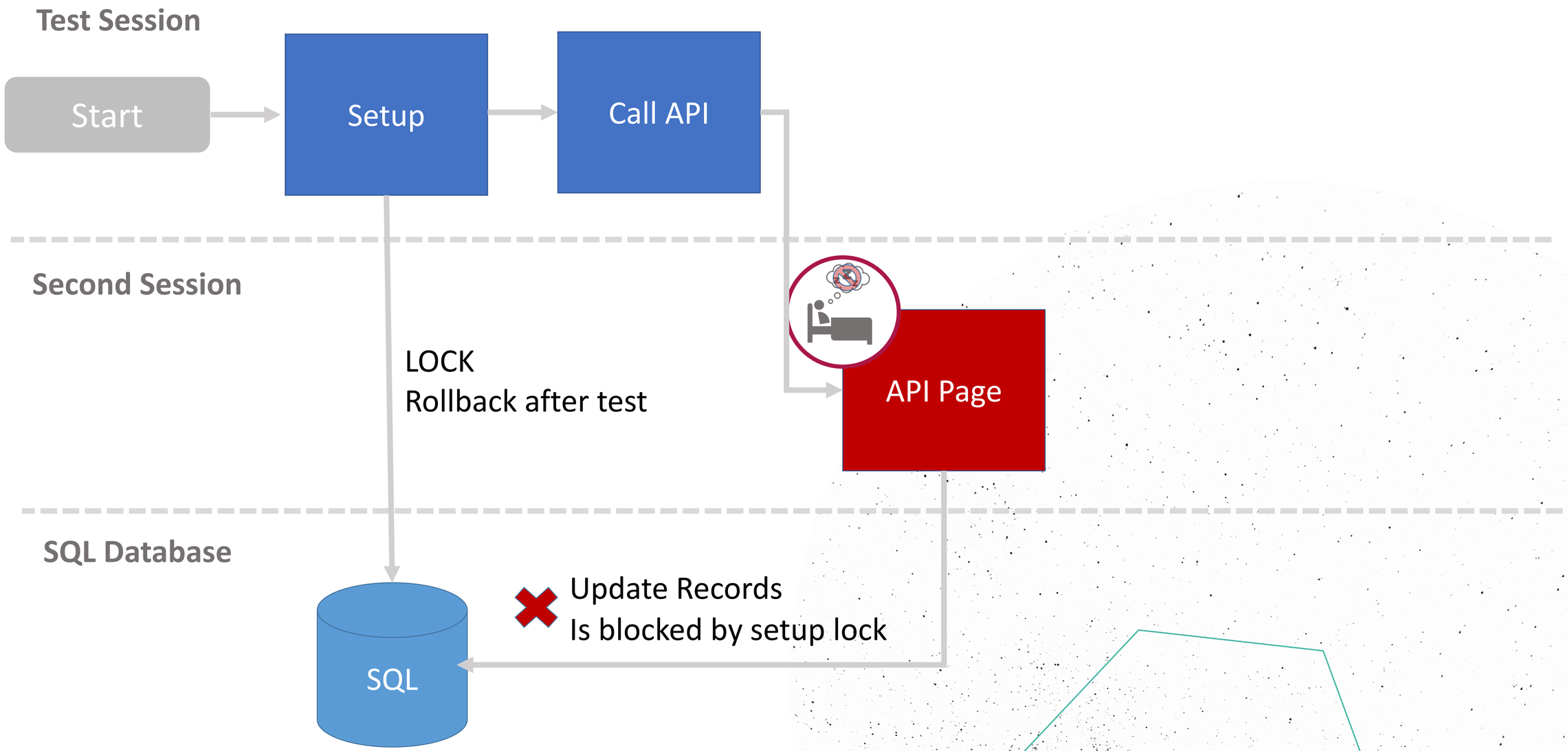
BC/NAV TECH DAYS 2022
mibuso.com

Risk is that API page behaves differently than UI Page

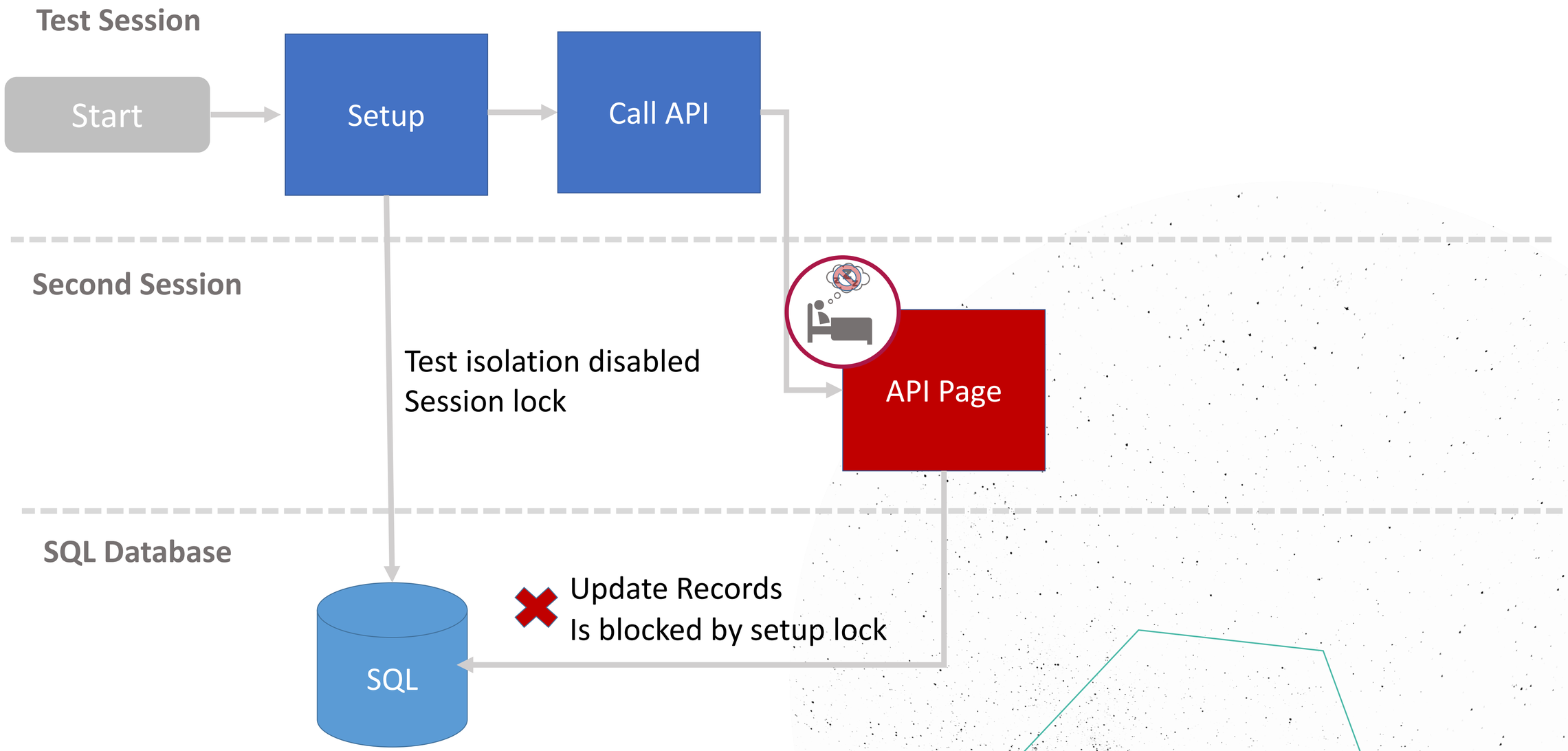
- Different flow of the triggers
- Different validation
- Any UI will fail the Web Service call



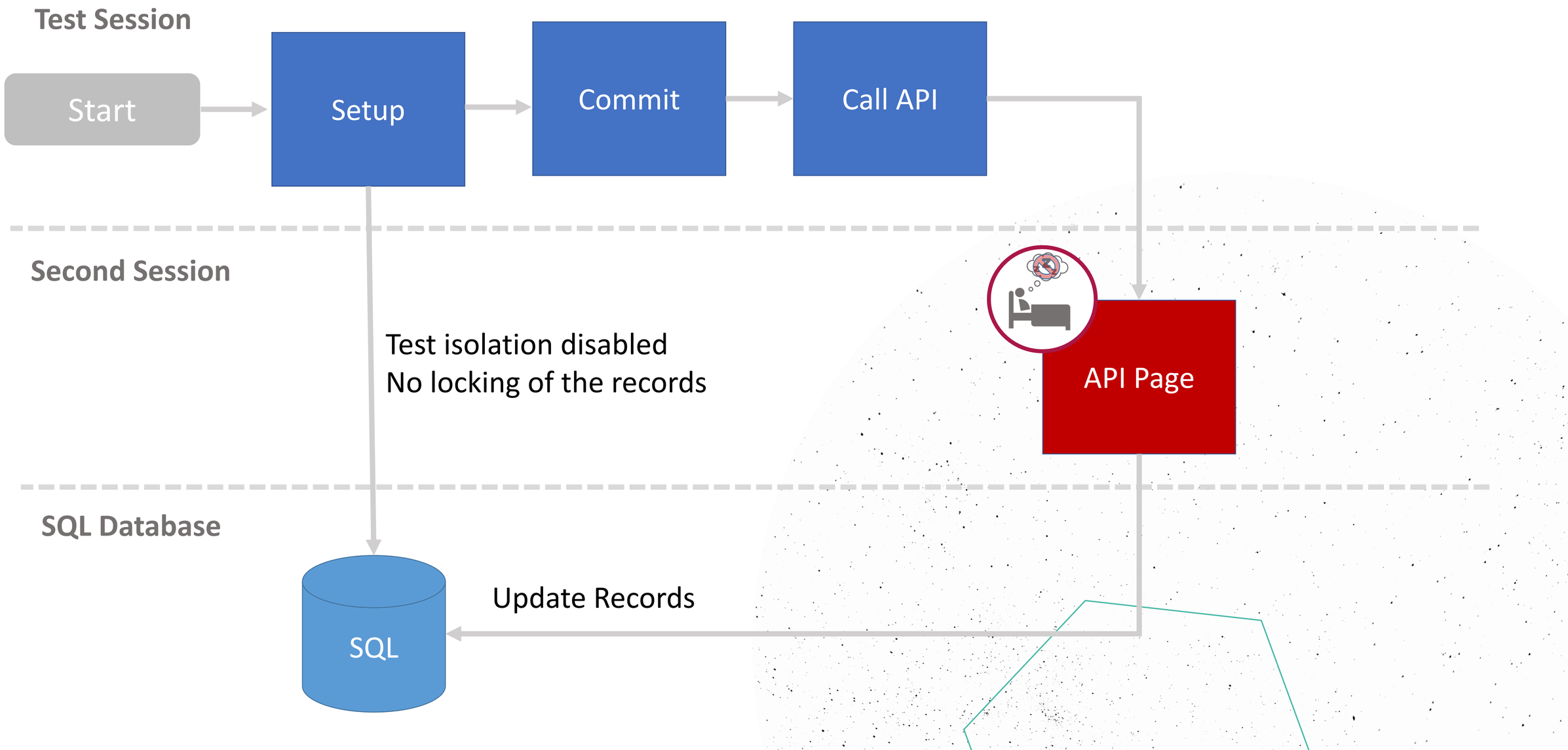
Testing APIs



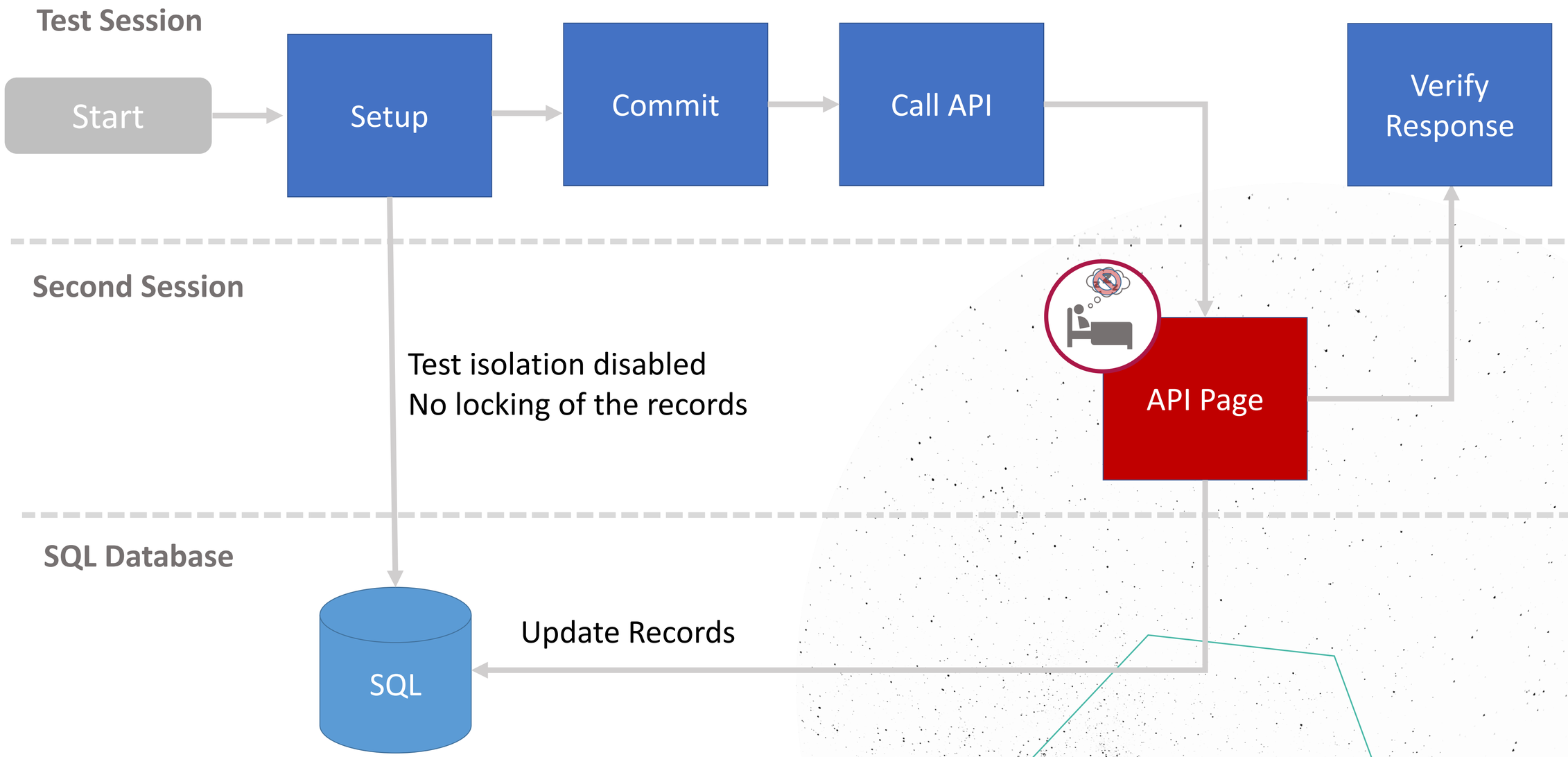
Testing APIs



Testing APIs



Testing APIs



Testing APIs - Authentication

MS Test use Windows authentication

Secret trick

With no Users in the Database there is no authentication

But due to community work we can also use basic authentication

[Simplifying API testing using basic authentication]

Testing TASKSCHEDULER/Job Queue

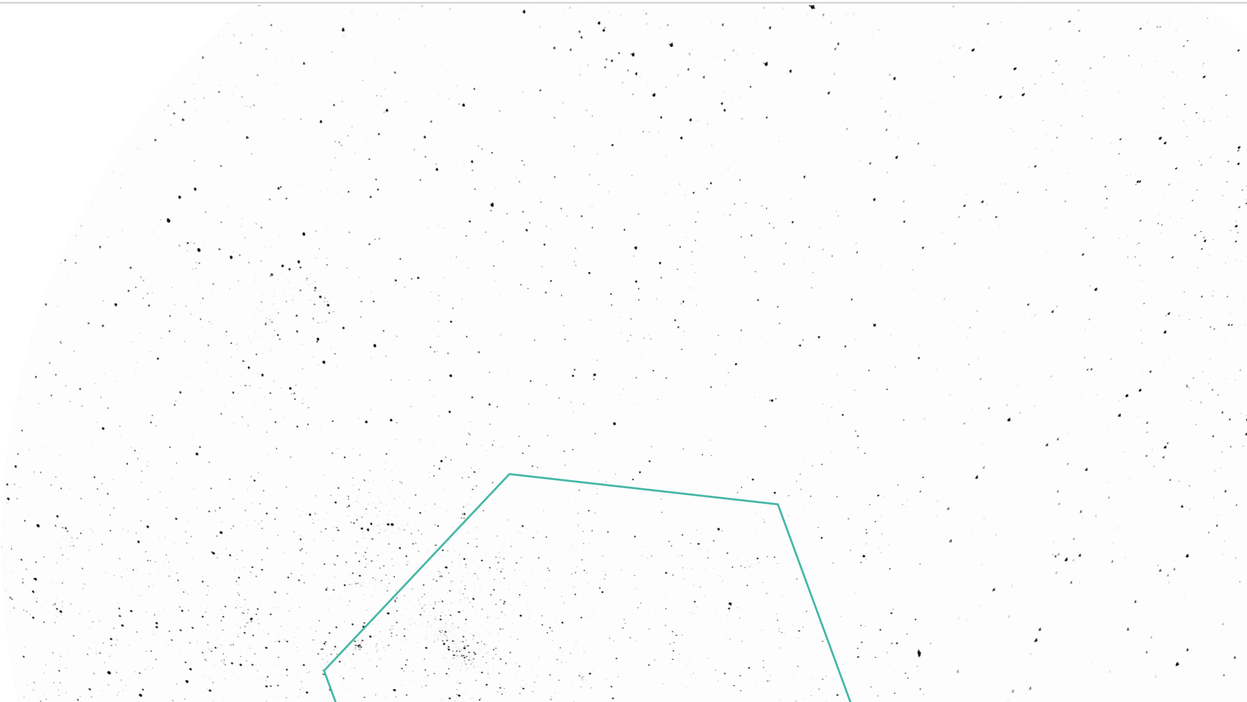
Disable Tasks when running tests

Tasks can lead to test instabilities

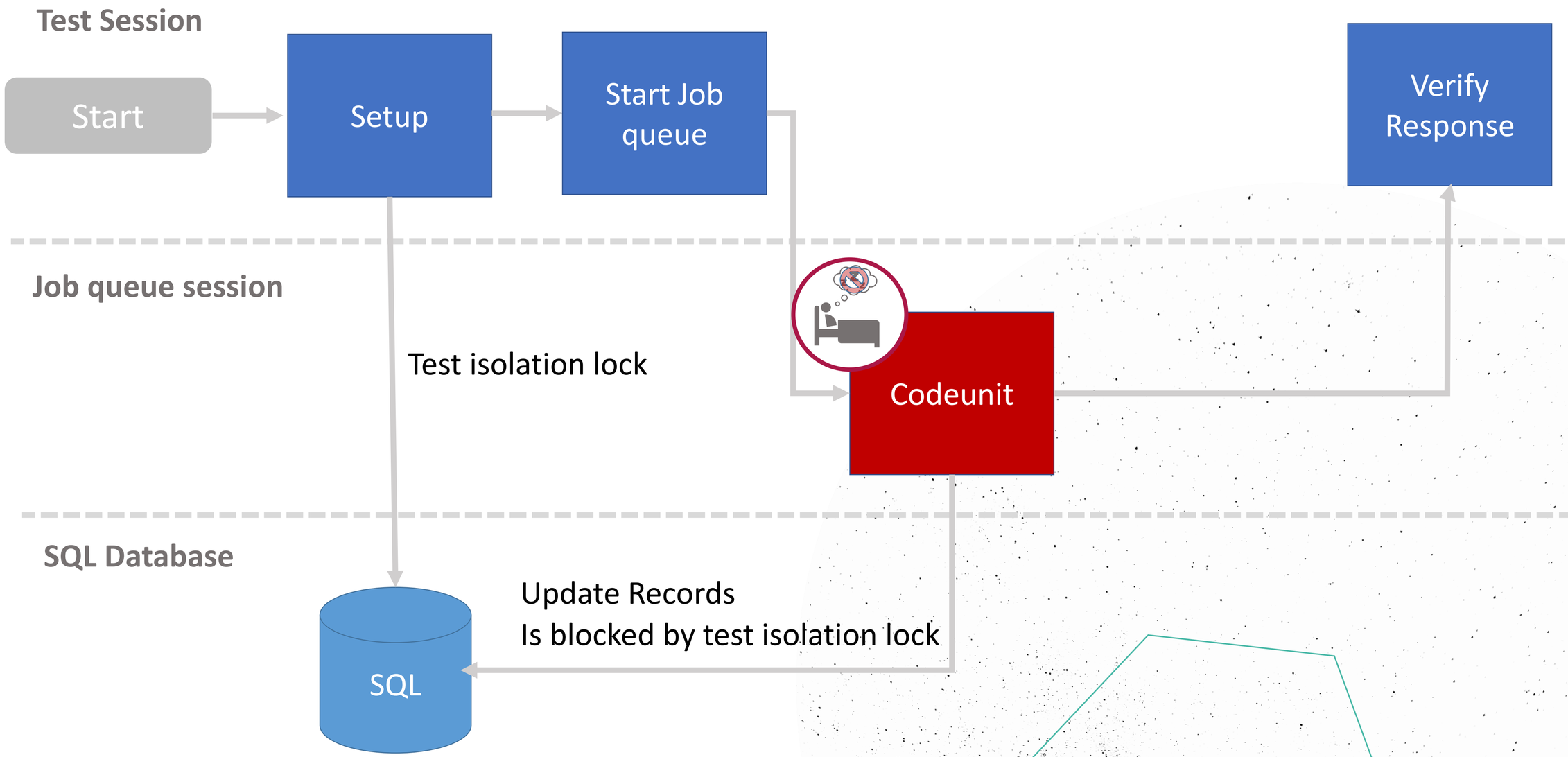
How do we test?

Unit test the codeunit

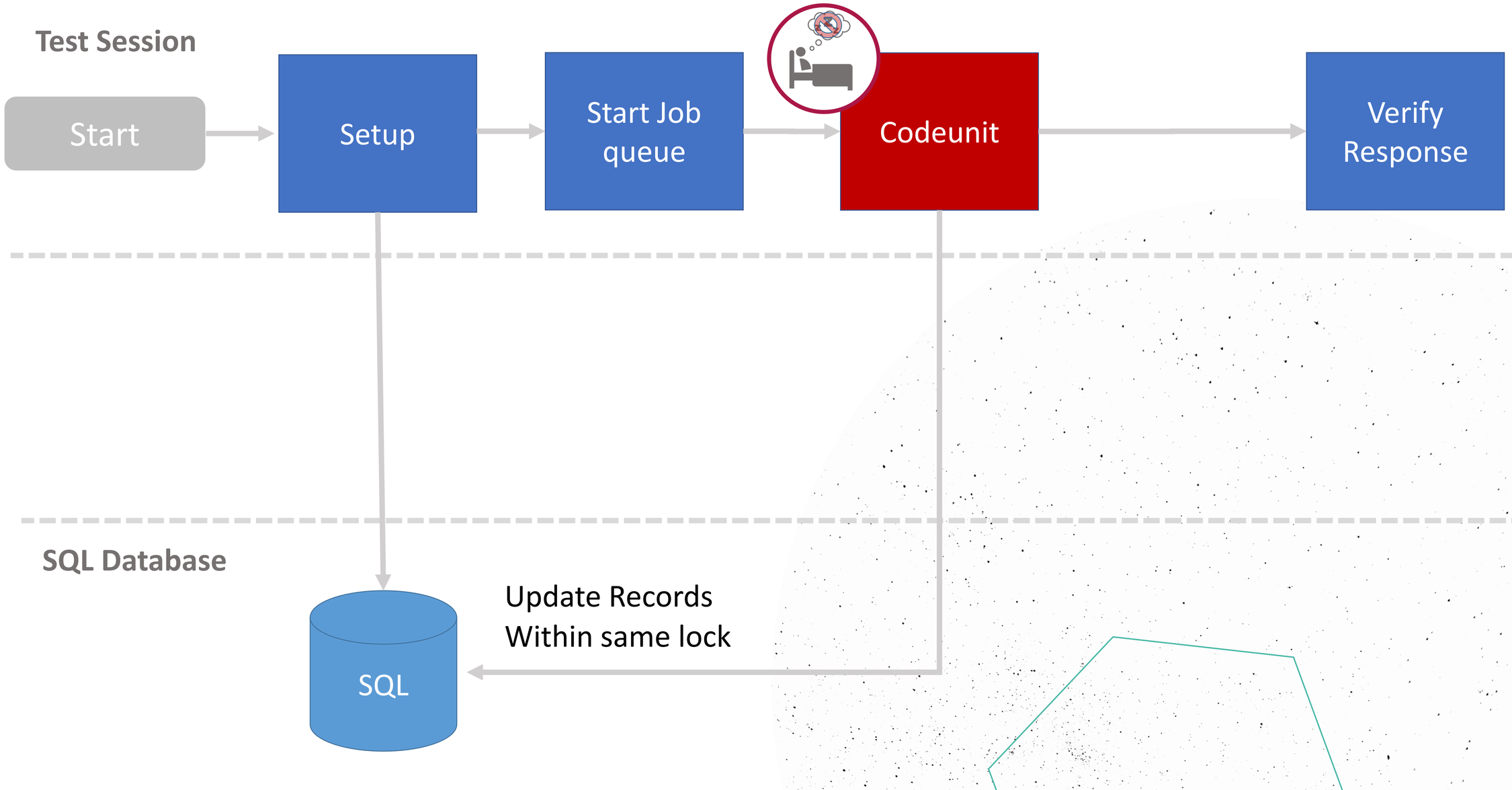
Key Name	Description
EnableTaskScheduler	Specifies whether the server instance starts with the task scheduler enabled. If this option is enabled, the server instance will process scheduled tasks. Default: Enabled Dynamically Updatable: No



Testing Tasks/Job Queue E2E



Testing Tasks/Job Queue E2E

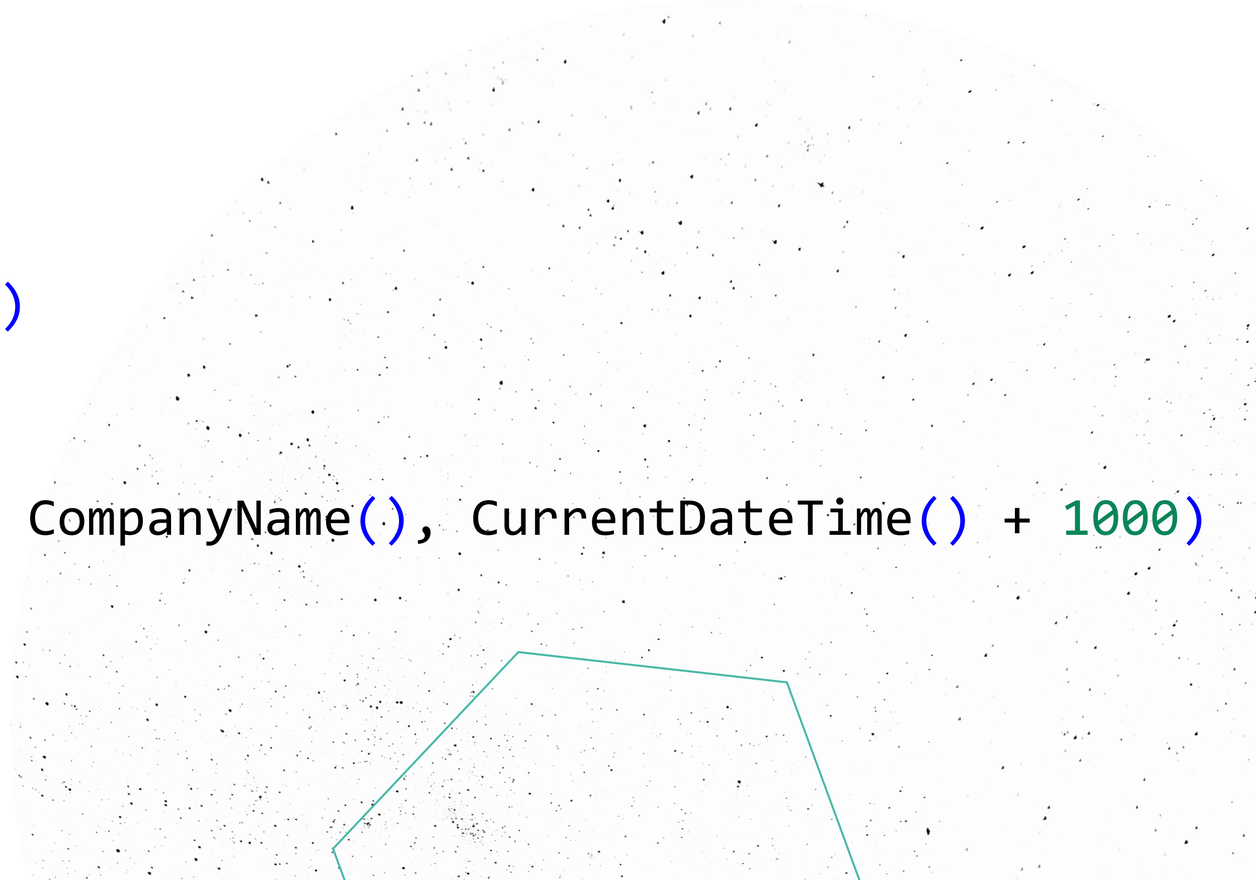


Testing Task Scheduler

BC/NAV TECH DAYS 2022
mibuso.com

```
local procedure TestTaskSchedulerExample()
var
    RunInSameSession: Boolean;
begin
    OnRunInSameSession(RunInSameSession);

    if RunInSameSession then
        Codeunit.Run(Codeunit::MyCodeunit)
    else
        TaskScheduler.CreateTask(
           Codeunit::MyCodeunit, 0, true, CompanyName(), CurrentDateTime() + 1000)
end;
```

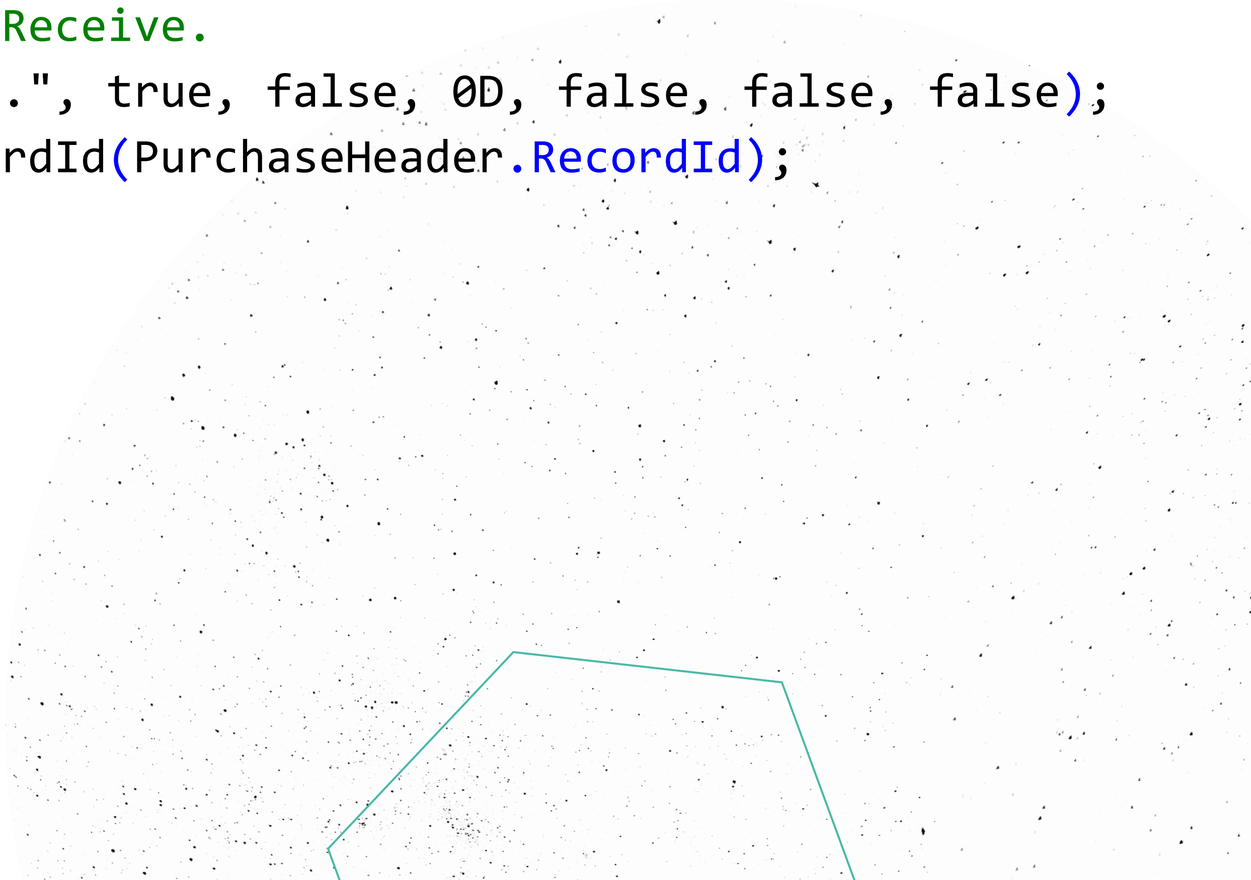


Testing Job Queues

BC/NAV TECH DAYS 2022
mibuso.com

```
// Setup.
BindSubscription(LibraryJobQueue);
LibraryJobQueue.SetDoNotHandleCodeunitJobQueueEnqueueEvent(true);

// [WHEN] Run Batch Post Purchase Order with Receive.
RunBatchPostPurchaseOrders(PurchaseHeader."No.", true, false, 0D, false, false, false);
LibraryJobQueue.FindAndRunJobQueueEntryByRecordId(PurchaseHeader.RecordId);
```



Good Tests

Cover the risk
Document the code
Fast to execute
Easy to read
Test one thing
Should not modify the environment
Data Agnostic

Test as close to the risk as possible

BC/NAV TECH DAYS 2022
mibuso.com

Testing Permissions

Test tooling internals

And some more …

Code Coverage

Testing Permissions

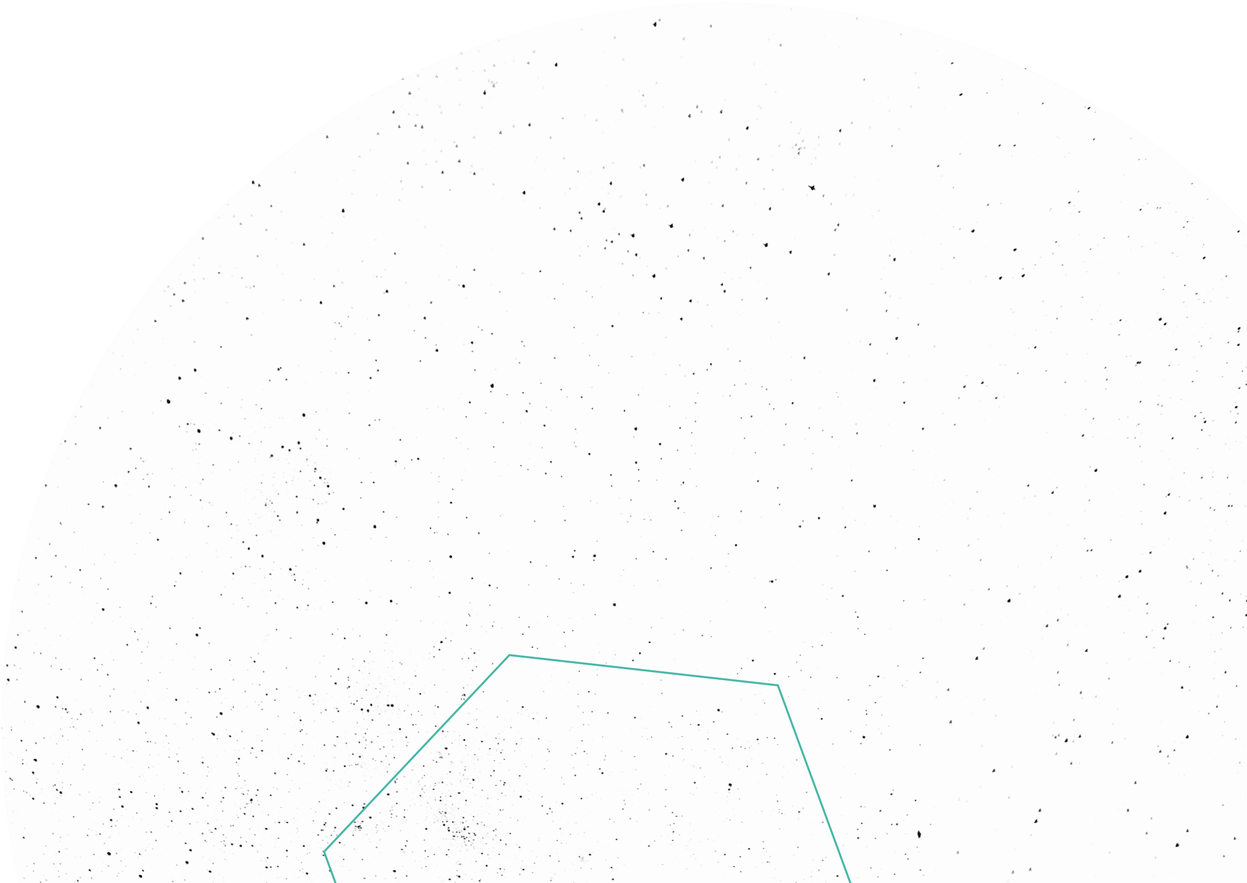
Testing Permissions

BC/NAV TECH DAYS 2022
mibuso.com

Why?

Basics of permission testing

Example scenarios

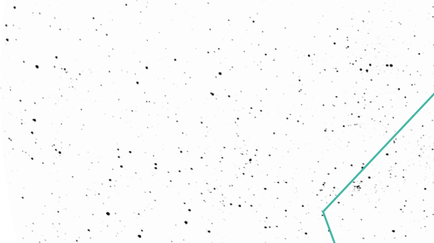


Testing Permissions – Why?

BC/NAV TECH DAYS 2022
mibuso.com

With testing …
… we verify if (new) functionality is working as intended

This does, however, not necessarily mean that all BC users will be able to operate this new functionality …
… as this also depends on the **permission sets** that have been assigned to them

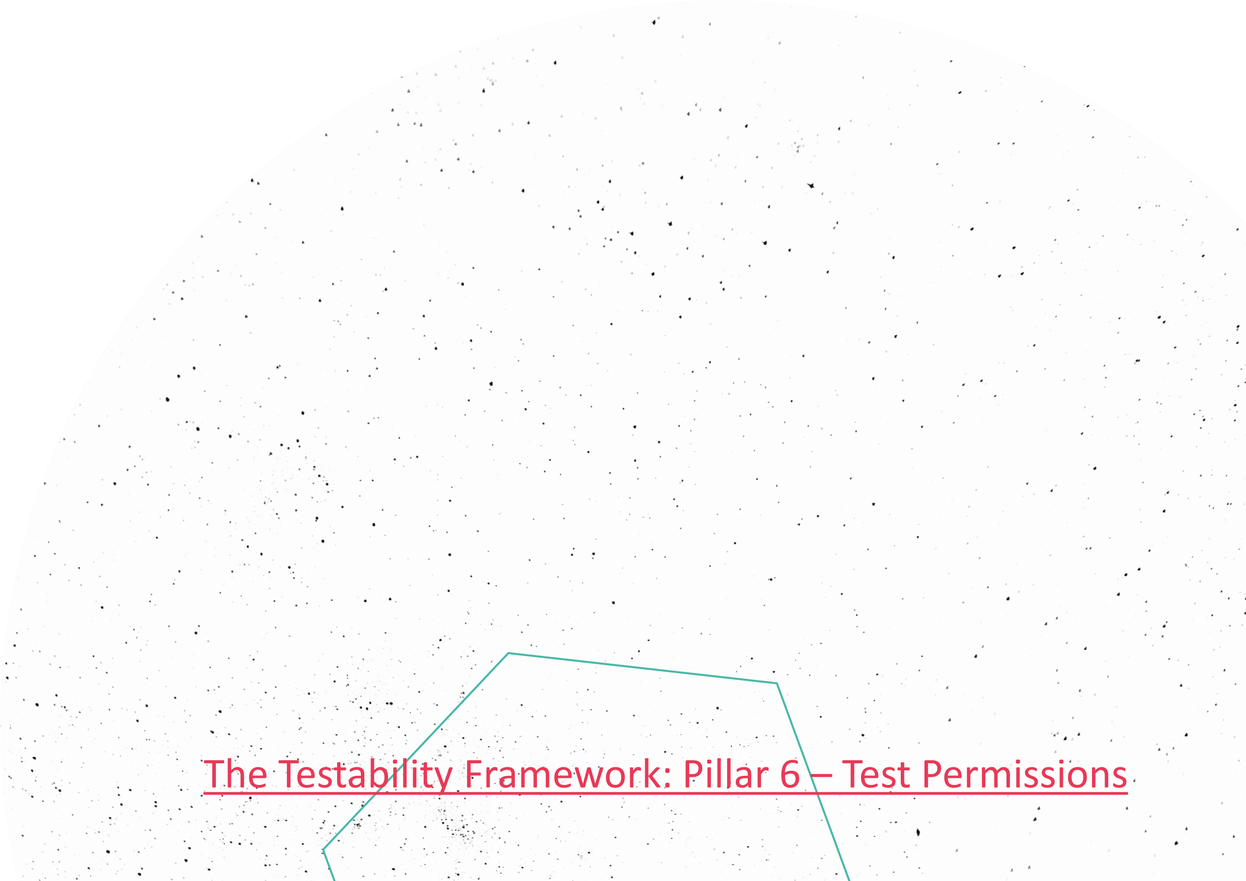


Testing Permissions – basics

TestPermissions property

Lowering permissions

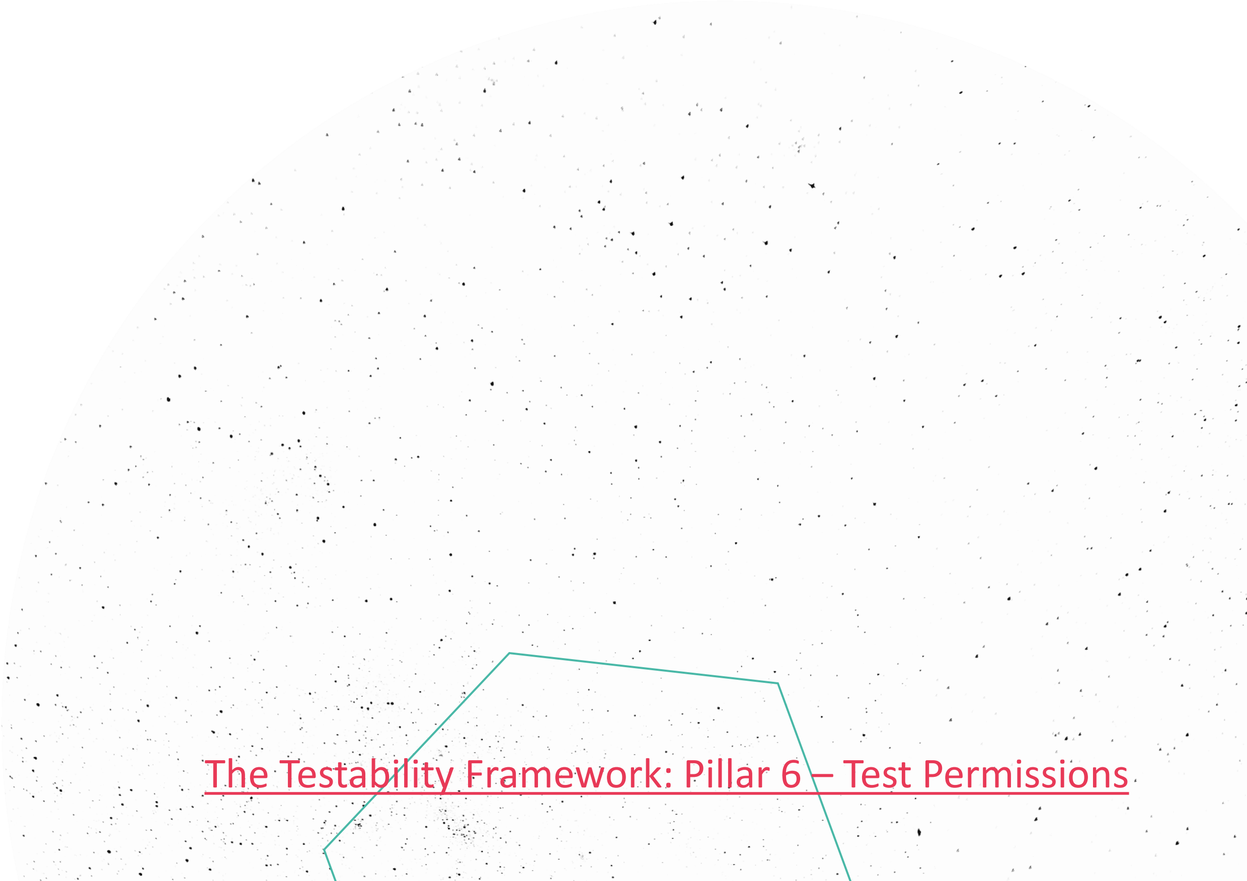
Permission set



The Testability Framework: Pillar 6 – Test Permissions

Testing Permissions – TestPermissions property

- Test Runner
- Test Codeunit
- Test Method



The Testability Framework: Pillar 6 – Test Permissions

Testing Permissions – TestPermissions property

Based on the value of the TestPermissions property of a Test Codeunit the

Test Runner manages which permission set is used on tests that are run

- For default value *Restrictive* Test Runner uses D365 BUS FULL ACCESS
- The TestPermissions attribute on a test method has the same effect
  - as default, it will inherit value from test codeunit TestPermissions

Note that TestPermissions as such does not do anything, it's the Test Runner

The Testability Framework: Pillar 6 – Test Permissions

Testing Permissions – TestPermissions property

From BC21 on, if you haven't set the TestPermissions property various tests will fail with an error like the following:

Sorry, the current permissions prevented the action.

(TableData <table name> Insert: <app name>)

Solution

Set TestPermissions property on relevant test codeunits to *Disabled*

Testing Permissions – Lowering permissions

Making use of .NET component *PermissionTestHelper.dll* …

- we can change permissions of current session on the fly
- but only lower permissions compared to the current users' permissions settings in database
  - therefore, current user should be SUPER
- can be done gradually
- can always revert to original permissions of current user
- does not change permission setup of user in database

Testing Permissions – Lowering permissions

PermissionTestHelper.dll is wrapped into *Permissions Mock.app* …

- both will be installed in docker container using New-BCContainer
  - with -includeTestToolkit (and -includeTestLibrariesOnly)

Note that …

- tests can only run in on-prem environment
- need to set the target in app.json to *OnPrem*

Testing Permissions – Library - Lower Permissions

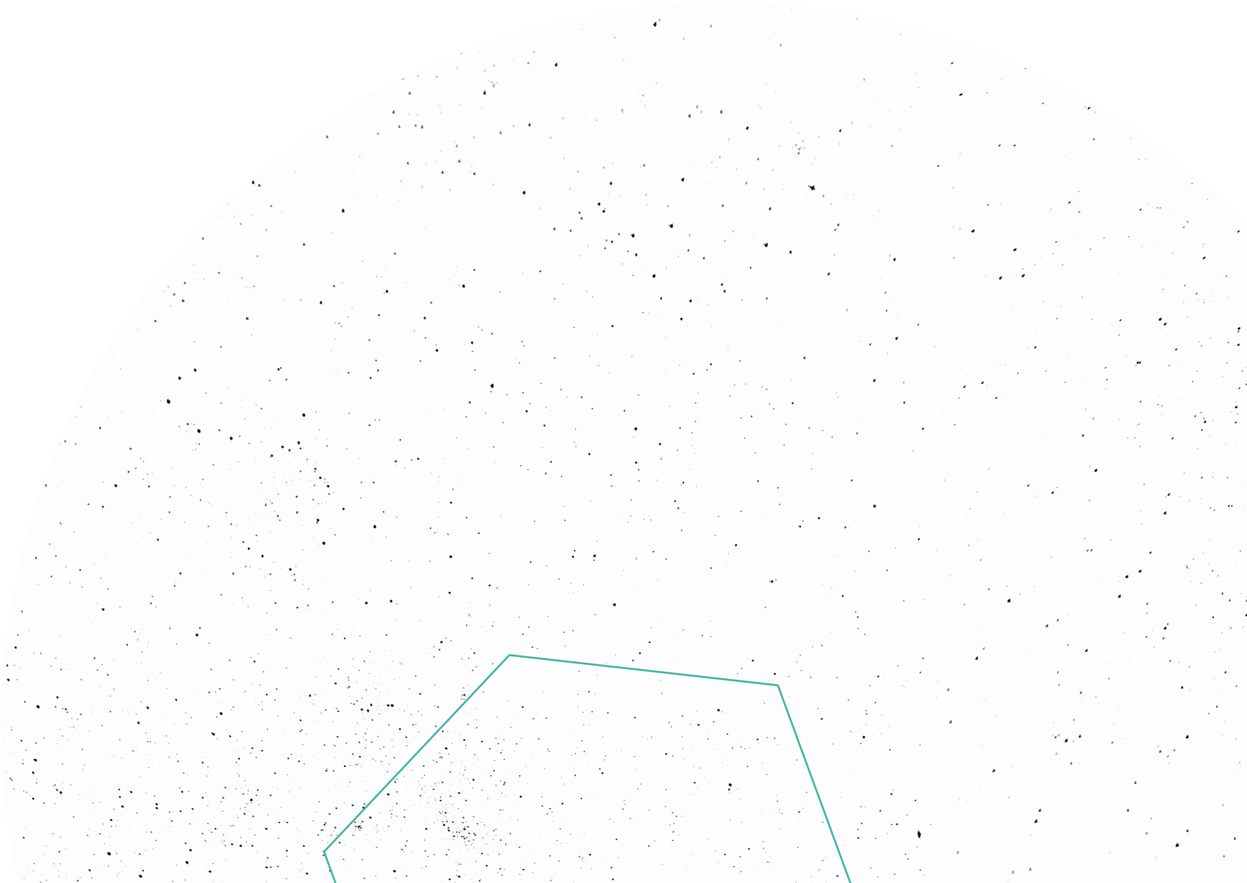
Library - Lower Permissions calls upon Permissions Mock.app and wraps its basic methods in various easy to use test helper methods for us, like …

- `StartLoggingNAVPermissions`
  - to (re)start permission logging (and set current user permission role)
- `PushPermissionSet`
  - to set current user permission role
- `AddPermissionSet`
  - to extend current user permissions with an additional set

Testing Permissions – permission sets

BC/NAV TECH DAYS 2022
mibuso.com

You can only Lower Permissions …



Testing Permissions – permission sets

BC/NAV TECH DAYS 2022
mibuso.com

You can only lower permissions …

… with permission sets that have been defined by permission set objects

```
permissionset 50000 "Lookup Value"
 {
     Assignable = true;
     Caption = 'Lookup Value';
     Permissions = tabledata LookupValue = RIMD;
 }
```

Testing Permissions – example scenario #0041

BC/NAV TECH DAYS 2022
mibuso.com

[FEATURE] LookupValue Permissions

[SCENARIO #0041] Create lookup value **without** permissions

[GIVEN] Full base starting permissions

[WHEN] Create lookup value

[THEN] Insert permissions error thrown

Testing Permissions – example scenario #0041

BC/NAV TECH DAYS 2022
mibuso.com

```
[Test]
procedure CreateLookupValueWithoutPermissions()
begin
    //[SCENARIO #0041] Create lookup value without permissions

    //[GIVEN] Full base starting permissions
    // Full base starting permissions automatically set based on
    // TestPermissions property

    //[WHEN] Create lookup value
    asserterror CreateLookupValueCode();

    //[THEN] Insert permissions error thrown
    VerifyPermissionsErrorThrown('Insert');
end;
```

Testing Permissions – example scenario #0042

BC/NAV TECH DAYS 2022
mibuso.com

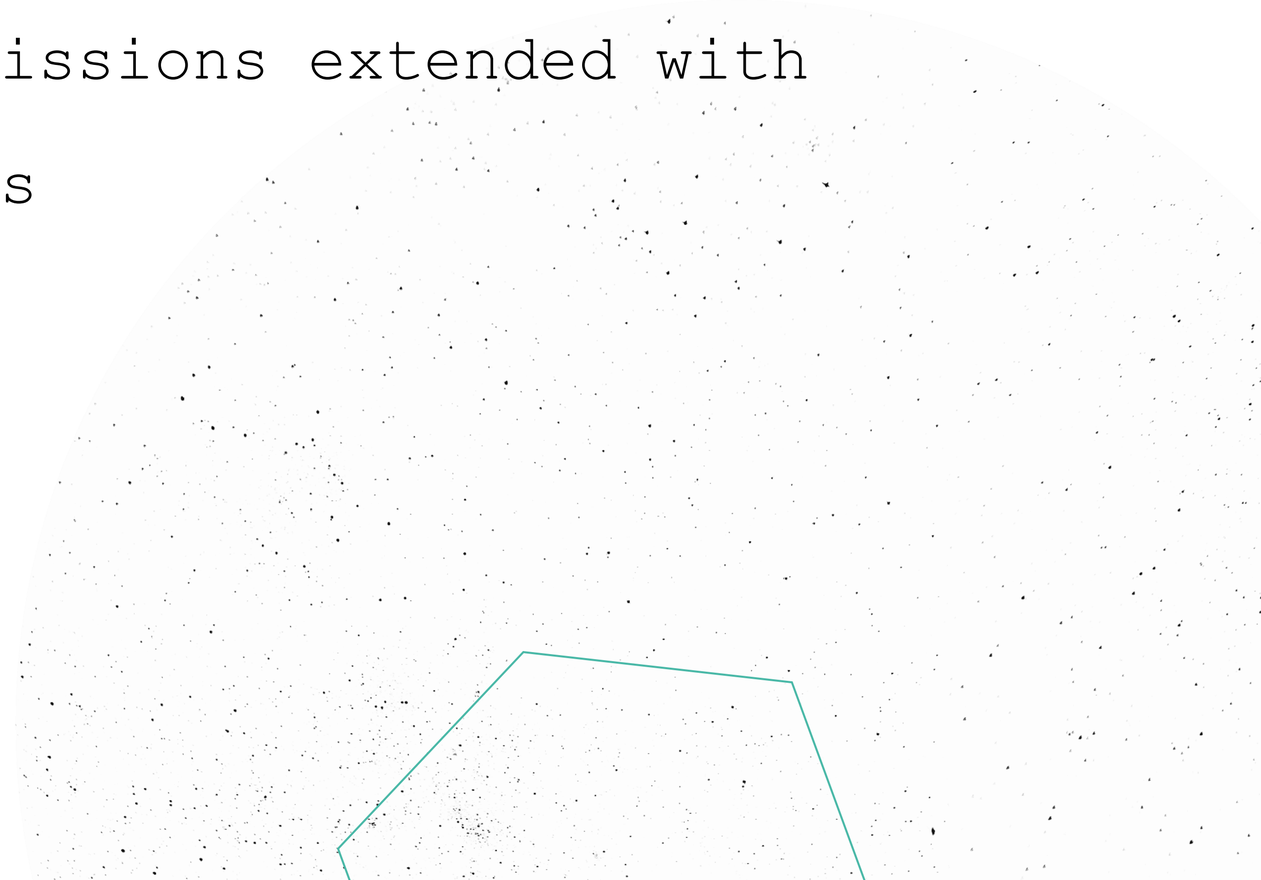
```
[FEATURE] LookupValue Permissions

[SCENARIO #0042] Create lookup value with permissions

[GIVEN] Full base starting permissions extended with
        Lookup Value permissions

[WHEN] Create lookup value

[THEN] Lookup value exists
```



Testing Permissions – example scenario #0042

BC/NAV TECH DAYS 2022
mibuso.com

```
[Test]
procedure CreateLookupValueWithPermissions()
var
    LookupValueCode: Code[10];
begin
    //[SCENARIO #0042] Create lookup value with permissions

    //[GIVEN] Full base starting permissions extended with Lookup Value permissions
    // Full base starting permissions automatically set based on
    // TestPermissions property
    AddLookupValuePermissions();

    //[WHEN] Create lookup value
    LookupValueCode := CreateLookupValueCode();

    //[THEN] Lookup value exists
    VerifyLookupValueExists(LookupValueCode);
end;
```

Testing Permissions – example scenario #0043

BC/NAV TECH DAYS 2022
mibuso.com

[FEATURE] LookupValue Permissions

[SCENARIO #0043] Read lookup value **without** permissions

[GIVEN] Unrestricted starting permissions

[GIVEN] Lookup value

[GIVEN] Full base starting permissions

[WHEN] Read lookup value

[THEN] Read permissions error thrown

Testing Permissions – example scenario #0043

BC/NAV TECH DAYS 2022
mibuso.com

```
[Test]
procedure ReadLookupValueWithoutPermissions ()
var
    LookupValueCode: Code[10];
begin
    //[SCENARIO #0043] Read lookup value without permissions

    //[GIVEN] Unrestricted starting permissions
    SetUnrestrictedStartingPermissions();    // SUPER
    //[GIVEN] Lookup value
    LookupValueCode := CreateLookupValueCode();
    //[GIVEN] Full base permissions
    SetFullBasePermissions();                // D365 BUS FULL ACCES

    //[WHEN] Read lookup value
    asserterror ReadLookupValueCode();

    //[THEN] Read permissions error thrown
    VerifyPermissionsErrorThrown('Read');
end;
```

Testing Permissions – last notes

- BC18.3 or later makes it much easier to do
- BC21: full implementation TestPermissions property

Code Coverage

Code coverage made simple

100% - extremely hard to reach in AL, questionable ROI

90%+ - most important error cases are covered

80%+ most of the scenarios covered

Getting to 70%+ requires a good test suite

Running few pages or reports will get you to around 50%

“

Code coverage tells us only that the seatbelt is fastened.

With good set of tests, we are buckled in and driving well.

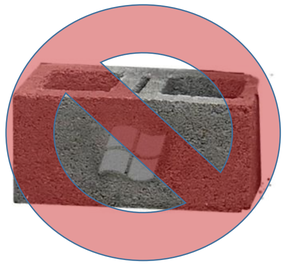
”

“

The most important metric is what is not covered.
Which scenarios are missing?

Good tests have Asserts in right places.

Don’t chase Code Coverage. Think about scenarios.



”

Test tooling internals

Tooling internals

```
PS C:\Enlistment\NAV> Invoke-NavCodeUnit -CodeunitId 134252 -ServerInstance NAV -CompanyName "CRONUS International Ltd."
[2022-09-13 19:14:34] Invoking codeunit '134252' on tenant '' on server 'NAV'.
WARNING: Test Codeunit 134252 Match Bank Reconciliation - UT
  OnRun: SUCCESS
  OneBankEntryMoreRecLinesFullMatch: SUCCESS
  OneBankEntryMoreRecLinesMatchDetails: SUCCESS
  OneBankEntryMoreRecLinesAmtVsDesc: SUCCESS
  OneBankEntryMoreRecLinesAmtVsDocNo: SUCCESS
  OneBankEntryMoreRecLinesDocNoVsDesc: SUCCESS
  OneBankEntryMoreRecLinesNoDate: SUCCESS
  OneBankEntryMoreRecLinesDateRange: FAILURE
    The following UI handlers were not executed: ConfirmOverwriteAutoMatchHandlerDefault
  OneBankEntryMoreRecLinesDateVsRange: FAILURE
    The following UI handlers were not executed: ConfirmOverwriteAutoMatchHandlerDefault
  OneBankEntryMoreRecLinesAmtVsDescDesc2: SUCCESS
  OneBankEntryMoreRecLinesAmtVsDocNoDesc2: SUCCESS
  OneBankEntryMoreRecLinesDocNoVsDescDesc2: SUCCESS
  OneBankEntryMoreRecLinesDescVsDesc2: SUCCESS
  MoreBankEntriesOneRecLineFullMatch: SUCCESS
  MoreBankEntriesOneRecLineAmtVsDescription: SUCCESS
  MoreBankEntriesOneRecLineAmtVsDocNo: SUCCESS
  MoreBankEntriesOneRecLineDocNoVsDescription: SUCCESS
  MoreBankEntriesOneRecLineDocNoVsExtDocNo: SUCCESS
  MultipleBankRec: SUCCESS
  ManyToOneIsSupported: SUCCESS
  MatchManuallyMatchDetailsOneToOne: SUCCESS
  MatchManuallyMatchDetailsManyToOne: SUCCESS
  OneToManyPartialBRL: SUCCESS
  OneToManyPartialBLE: SUCCESS
  RemoveMatchOneToMany: SUCCESS
  RemoveMatchOneToManyStatementLineDeleted: SUCCESS
  RemoveMatchFromBLE: SUCCESS
  RemoveMatchManyToOne: SUCCESS
  PostMatchManyToOne: FAILURE
    The following UI handlers were not executed: BankStatementPagePostHandler
  UTFieldsBankAccReconciliationLine: SUCCESS
  UTFieldsPostedPaymentReconLine: SUCCESS
  GenerateAppliesToID: SUCCESS
  BankAccountWithSpecialChar: SUCCESS
FAILURE
```

Server (AL Code)

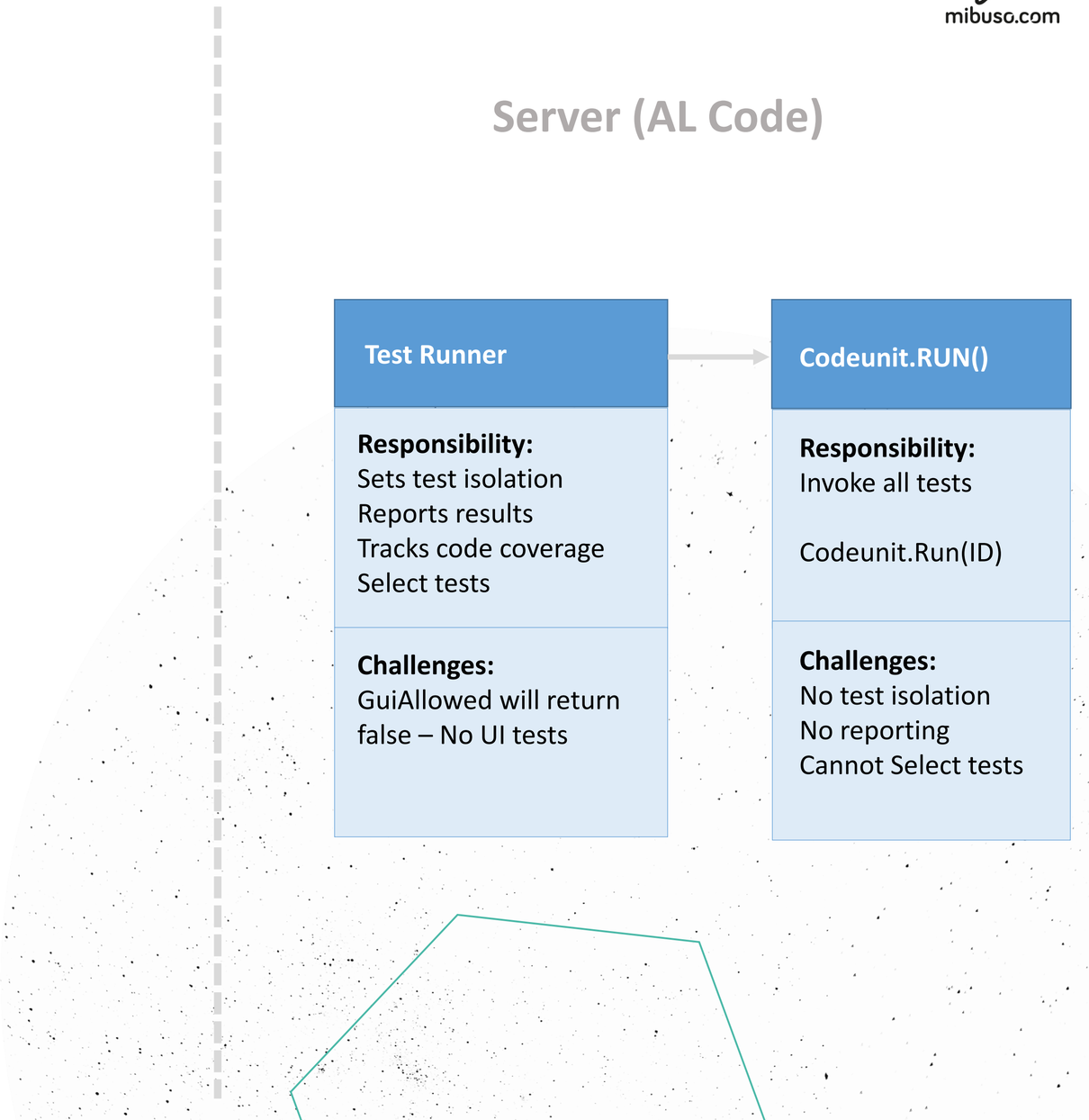
Codeunit.RUN()

Responsibility:
Invoke all tests

Codeunit.Run(ID)

Challenges:
No test isolation
No reporting
Cannot Select tests

Tooling internals



Tooling internals

Client Side

Mini Client
(AKKA ALTest)
MS Internal

Responsibility:
Create UI session
Build tests, track CC and
Many others…

Challenges:
Nobody knows all the things it does

Server (AL Code)

Test Runner

Responsibility:
Sets test isolation
Reports results
Tracks code coverage
Select tests

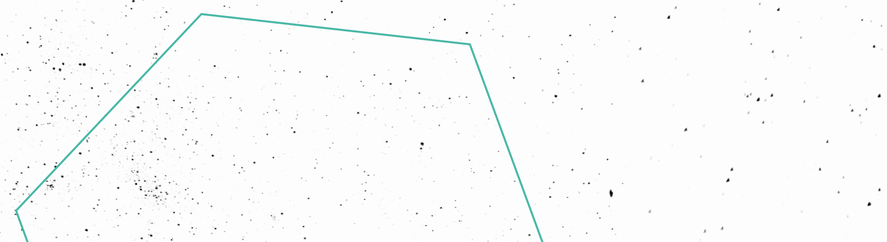
Challenges:
GuiAllowed will return false

Codeunit.RUN()

Responsibility:
Invoke all tests

Codeunit.Run(ID)

Challenges:
No test isolation
No reporting
Cannot Select tests



Tooling internals

Client Side

Mini Client
(AKKA ALTest)
MS Internal

Responsibility:
Create UI session
Build tests, track CC and
Many others…

Challenges:
Nobody knows all the things it does

ALTestInternal.ps1
PowerShell

Responsibility:
Create UI session
Wrap UI Web Services

Challenges:
UI web services to be deprecated

Server (AL Code)

Test Runner

Responsibility:
Sets test isolation
Reports results
Tracks code coverage
Select tests

Challenges:
GuiAllowed will return false

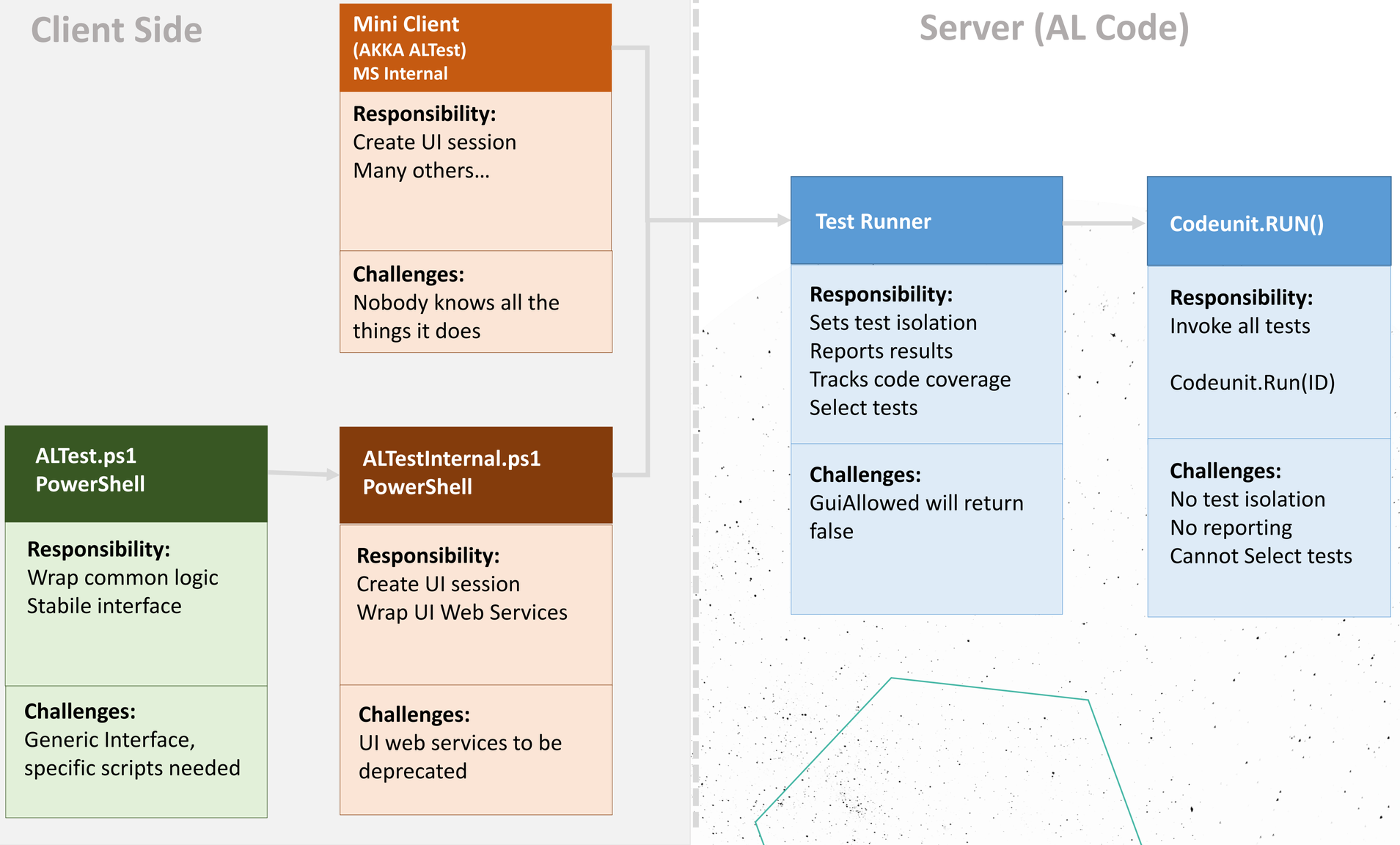
Codeunit.RUN()

Responsibility:
Invoke all tests

Codeunit.Run(ID)

Challenges:
No test isolation
No reporting
Cannot Select tests

Tooling internals



Tooling internals

Client Side

BC Container Helper

Responsibility:
Enable partners to run tests

Microsoft Test Scripts

Responsibility:
Enable MS Developers to run tests

ALTest.ps1 PowerShell

Responsibility:
Wrap common logic
Stabile interface

Challenges:
Generic Interface, specific test scripts needed

Mini Client (AKKA ALTest) MS Internal

Responsibility:
Create UI session
Many others…

Challenges:
Nobody knows all the things it does

ALTestInternal.ps1 PowerShell

Responsibility:
Create UI session
Wrap UI Web Services

Challenges:
UI web services to be deprecated

Server (AL Code)

Test Runner

Responsibility:
Sets test isolation
Reports results
Tracks code coverage
Select tests

Challenges:
GuiAllowed will return false

Codeunit.RUN()

Responsibility:
Invoke all tests

Codeunit.Run(ID)

Challenges:
No test isolation
No reporting
Cannot Select tests

Tooling internals

Client Side

BC Container Helper

Responsibility:
Enable partners to run tests

Microsoft Test Scripts

Responsibility:
Enable MS Developers to run tests

ALTest.ps1 PowerShell

Responsibility:
Wrap common logic
Stabile interface

Challenges:
Generic Interface, specific test scripts needed

ALTestInternal.ps1 PowerShell

Responsibility:
Create UI session
Wrap UI Web Services

Challenges:
UI web services to be deprecated

Server (AL Code)

Test Runner

Responsibility:
Sets test isolation
Reports results
Tracks code coverage
Select tests

Challenges:
GuiAllowed will return false

Codeunit.RUN()

Responsibility:
Invoke all tests

Codeunit.Run(ID)

Challenges:
No test isolation
No reporting
Cannot Select tests

Tooling internals

Client Side

BC Container Helper

Responsibility:
Enable partners to run tests

Microsoft Test Scripts

Responsibility:
Enable MS Developers to run tests

ALTest.ps1 PowerShell

Responsibility:
Wrap common logic
Stabile interface

Challenges:
Generic Interface, specific test scripts needed

Server (AL Code)

Management Test Endpoint (API)

Responsibility:
Create UI session
Wrap UI Web Services

Test Runner

Responsibility:
Sets test isolation
Reports results
Tracks code coverage
Select tests

Challenges:
GuiAllowed will return false

Codeunit.RUN()

Responsibility:
Invoke all tests

Codeunit.Run(ID)

Challenges:
No test isolation
No reporting
Cannot Select tests

Tooling internals

Client Side

BC Container Helper

Responsibility:
Enable partners to run tests

VS Code Test Runner

Responsibility:
Run tests from VS Code

Microsoft Test Scripts

Responsibility:
Enable MS Developers to run tests

ALTest.ps1 PowerShell

Responsibility:
Wrap common logic
Stabile interface

Challenges:
Generic Interface, specific test scripts needed

Server (AL Code)

Management Test Endpoint (API)

Responsibility:
Create UI session
Wrap UI Web Services

Test Runner

Responsibility:
Sets test isolation
Reports results
Tracks code coverage
Select tests

Challenges:
GuiAllowed will return false

Codeunit.RUN()

Responsibility:
Invoke all tests

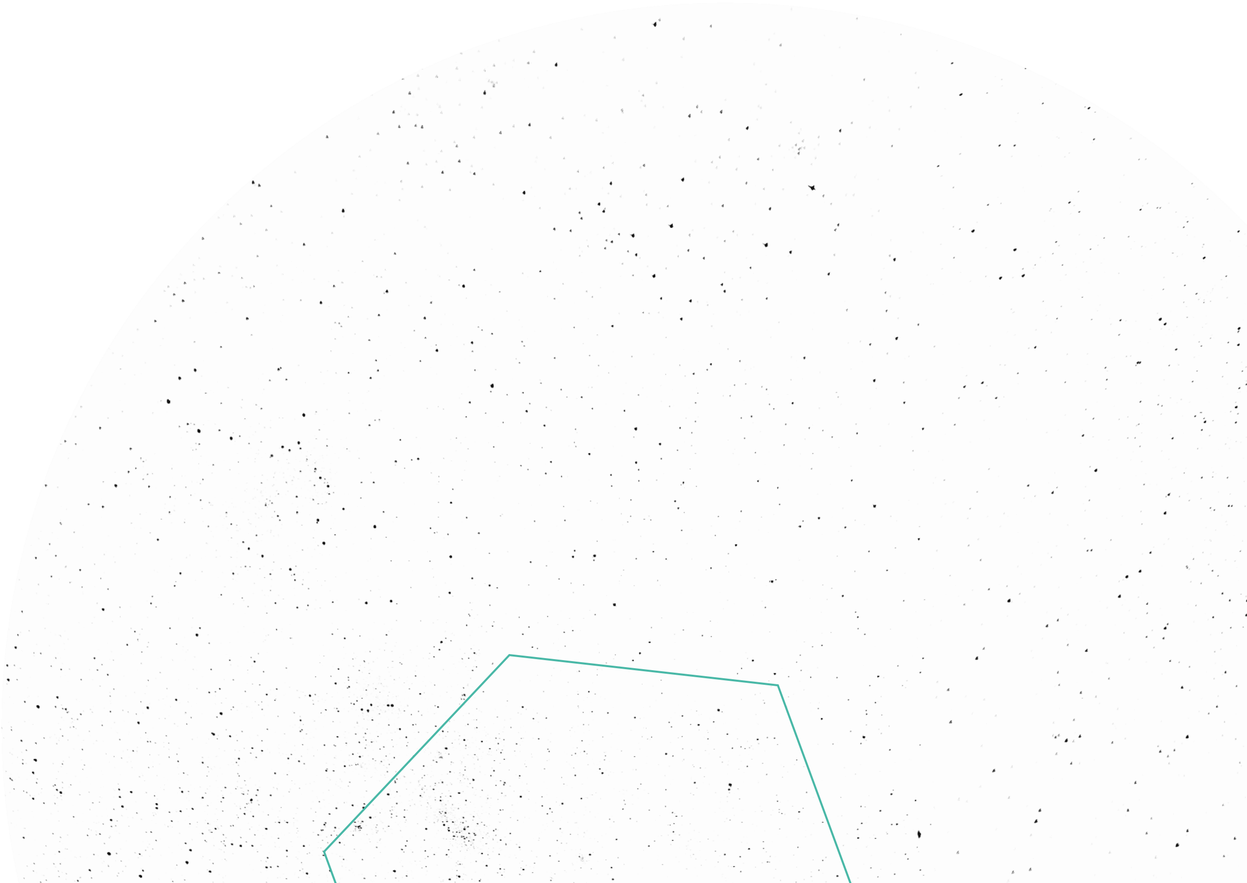
Codeunit.Run(ID)

Challenges:
No test isolation
No reporting
Cannot Select tests

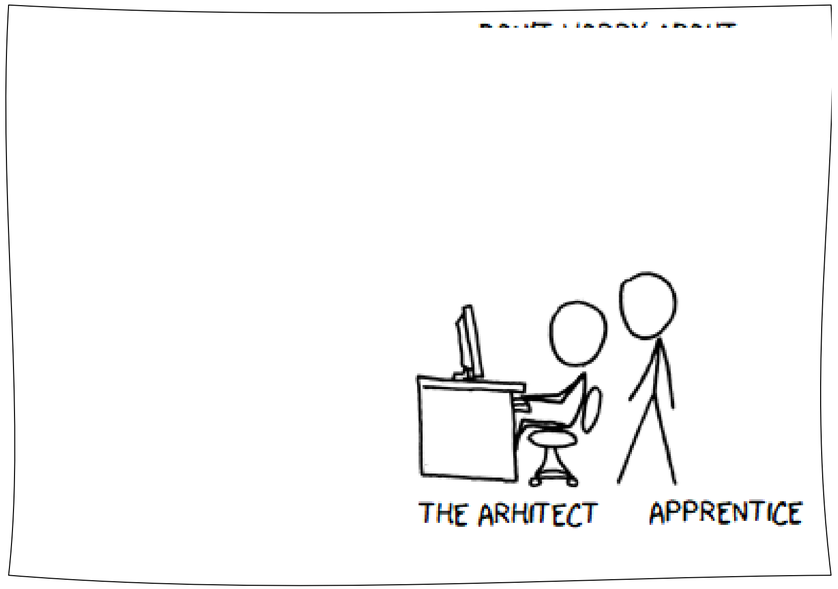
Test Discovery

Test Discovery

- Churn based execution
- Coming “soon”



Conclusion

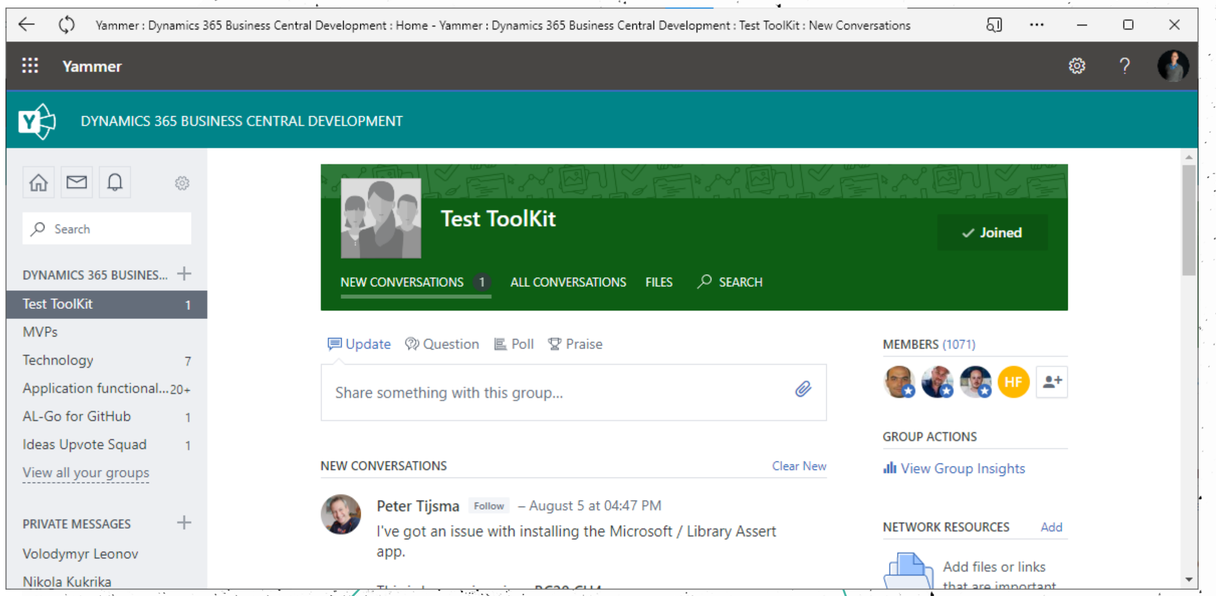

THE ARHITECT
APPRENTICE

Conclusion

- Write good tests
- Cover the risk
- Use examples
- Try to hit 80% Code Coverage at least sometimes

And … join us on Yammer

https://www.yammer.com/dynamicsnavdev/#/threads/inGroup?type=in_group&feedId=13879726



Any Questions?

BC/NAV TECH DAYS 2022
mibuso.com

Thank You!

BC/NAV TECH DAYS 2022

mibuso.com

Coffee Break

10 YEAR ANNIVERSARY

www.bctechdays.com