Biml for Beginners: Speed up your SSIS development

Cathrine Wilhelmsen · SQLBits XV · May 7th 2016

Session Description

SSIS is a powerful tool for extracting, transforming and loading data, but creating and maintaining a large number of SSIS packages can be both tedious and time-consuming. Even if you use templates and follow best practices you often have to repeat the same steps over and over and over again. Handling metadata and schema changes is a manual process, and there are no easy ways to implement new requirements in multiple packages at the same time.

It is time to bring the Don't Repeat Yourself (DRY) software engineering principle to SSIS projects. First learn how to use Biml and BimlScript to generate SSIS packages from database metadata and implement changes in all packages with just a few clicks. Then take the DRY principle one step further and learn how to update all packages in multiple projects by separating and reusing common code.

Speed up your SSIS development by using Biml and BimlScript, and see how you can complete in a day what once took more than a week!

The Basics

Tools & Projects

Code Management

Cathrine Wilhelmsen

Data Warehouse Architect
Business Intelligence Developer

@cathrinew

cathrinewilhelmsen.net

MVP
Microsoft®
Most Valuable Professional

You…

SSIS developer?

Easily bored?

Tired of repetitive work?

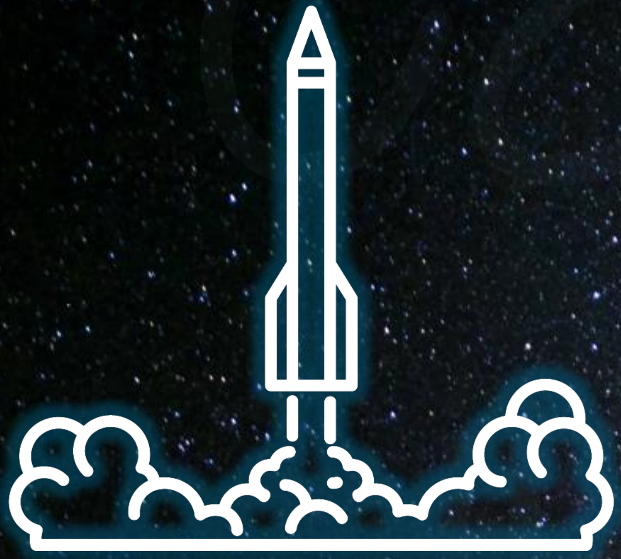
Work...

Long development time?

Many SSIS packages?

Frequent requirement changes?

Ever experienced this?



takeoff!

new standards

crash landing!

Ready for a change?

Biml™

What is Biml?

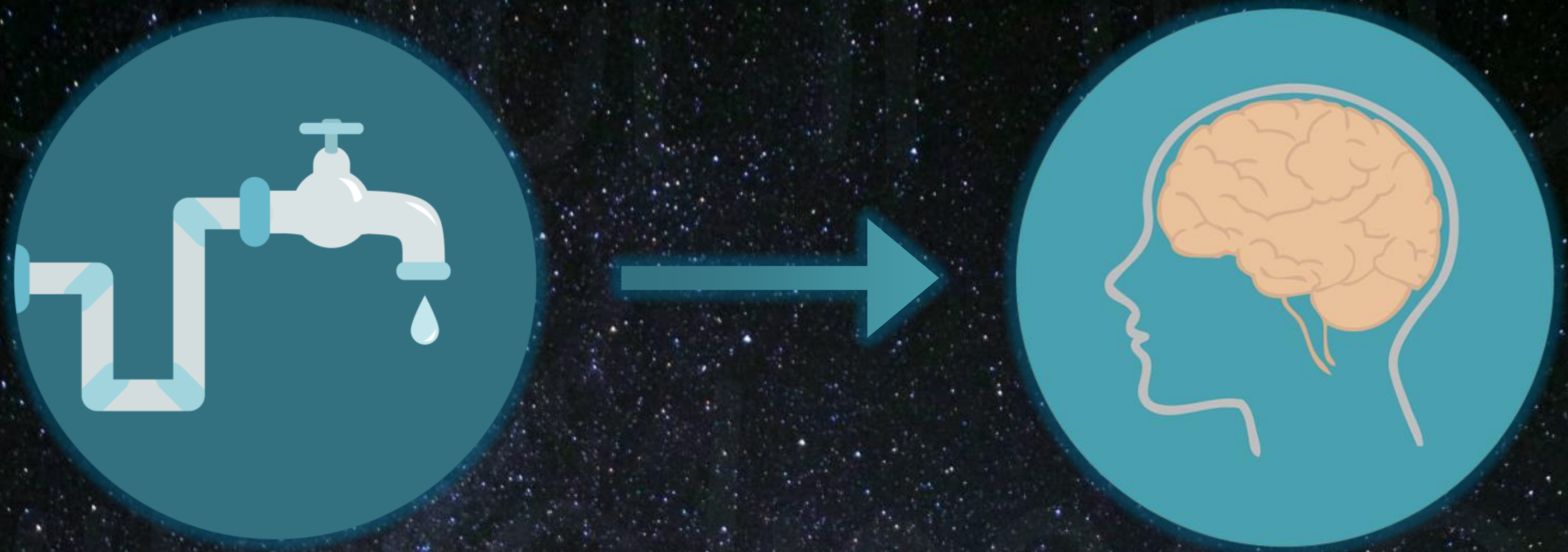
Business Intelligence Markup Language

Easy to read and write XML language

Describes business intelligence objects:

- Databases, Schemas, Tables, Views, Columns
- SSIS Packages
- SSAS Cubes

From Traditional SSIS to Agile SSIS



Traditional SSIS: Plumbing

Agile SSIS: Business Logic

Will Biml solve *all* your problems?

Probably not…

Biml is a tool for generating SSIS packages

Biml is not a pre-defined ETL framework

Biml is not a tool for automated deployment

…but it will solve *many* problems!

How can Biml help you?

Timesaving: Many SSIS Packages from one Biml file

Reusable: Write once and run on any platform

Flexible: Start simple, expand as you learn

What do you need?

Visual Studio®

BI BIDS Helper

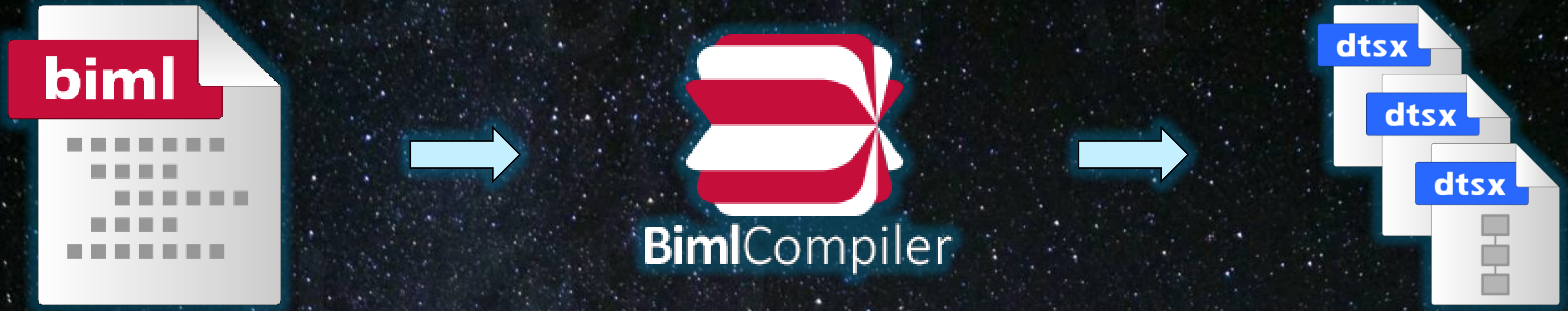
...or you can use the new Biml tools

BimlExpress

BimlOnline

BimlStudio

How does it work?



...generated packages look exactly like manually created packages

Biml syntax

```
<Biml xmlns="http://schemas.varigence.com/biml.xsd">
    <Packages>
        <Package Name="EmptyPackage1"></Package>
        <Package Name="EmptyPackage2"/>
    </Packages>
</Biml>
```

Biml syntax: Root Element

```
<Biml xmlns="http://schemas.varigence.com/biml.xsd">
    <Packages>
        <Package Name="EmptyPackage1"></Package>
        <Package Name="EmptyPackage2"/>
    </Packages>
</Biml>
```

Biml syntax: Collections of Root Objects

```
<Biml xmlns="http://schemas.varigence.com/biml.xsd">
    <Packages>
        <Package Name="EmptyPackage1"></Package>
        <Package Name="EmptyPackage2"/>
    </Packages>
</Biml>
```

Biml syntax: Elements

```
<Biml xmlns="http://schemas.varigence.com/biml.xsd">
   <Packages>
      <Package Name="EmptyPackage1"></Package>
      <Package Name="EmptyPackage2"/>
   </Packages>
</Biml>
```

Biml syntax: Attributes

```
<Biml xmlns="http://schemas.varigence.com/biml.xsd">
    <Packages>
        <Package Name="EmptyPackage1"></Package>
        <Package Name="EmptyPackage2"/>
    </Packages>
</Biml>
```

Biml syntax: Full vs. Shorthand Syntax

```
<Biml xmlns="http://schemas.varigence.com/biml.xsd">
   <Packages>
      <Package Name="EmptyPackage1"></Package>
      <Package Name="EmptyPackage2"/>
   </Packages>
</Biml>
```

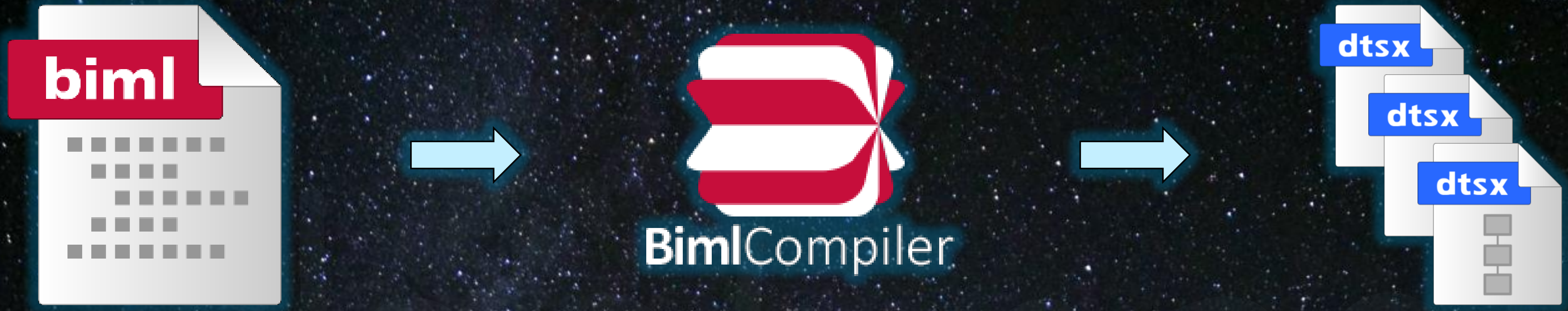
Biml syntax: Full vs. Shorthand Syntax

```
<Biml xmlns="http://schemas.varigence.com/biml.xsd">
    <Packages>
        <Package Name="EmptyPackage1">...</Package>
        <Package Name="EmptyPackage2"/>
    </Packages>
</Biml>
```

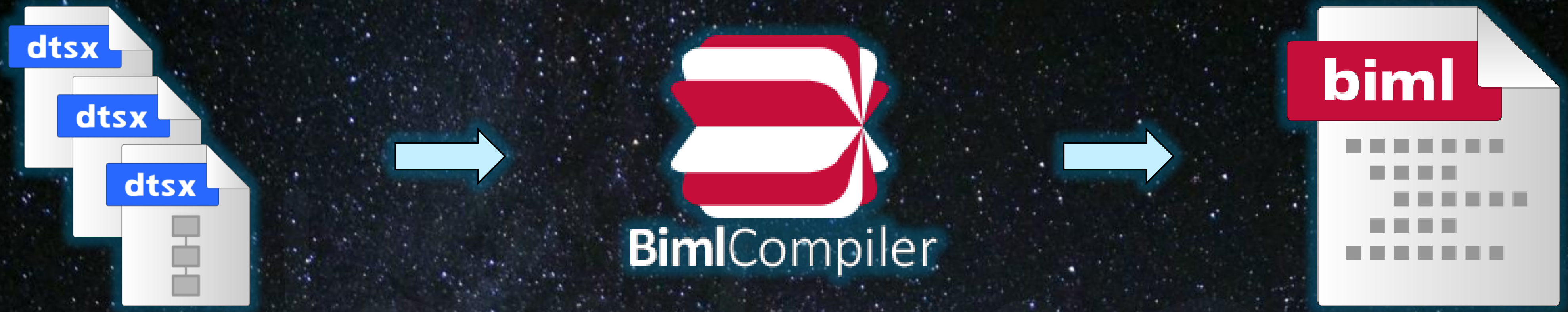
Let's generate some packages!

(demo time!)

Ok, so we can go from Biml to SSIS...



...can we go from SSIS to Biml?



Yes! :)

BimlOnline BETA

Let's reverse-engineer some packages!

The magic is in the

BimlScript

What is BimlScript?

Extend Biml with C# or VB code blocks

- **Import** database structure and metadata
- **Loop** over tables and columns
- **Expressions** replace static values

Allows you to control and manipulate Biml code

BimlScript Code Nuggets

<#@ … #>	Directives	*(Compiler instructions)*
<# … #>	Control Nuggets	*(Control logic)*
<#= … #>	Text Nuggets	(*Returns string*)
<#+ … #>	Class Nuggets	*(Create helper classes)*

BimlScript Syntax

```
<Biml xmlns="http://schemas.varigence.com/biml.xsd">
  <Packages>
    <# foreach (var table in RootNode.Tables) { #>
      <Package Name="Load_<#=table.Name#>"></Package>
    <# } #>
  </Packages>
</Biml>
```

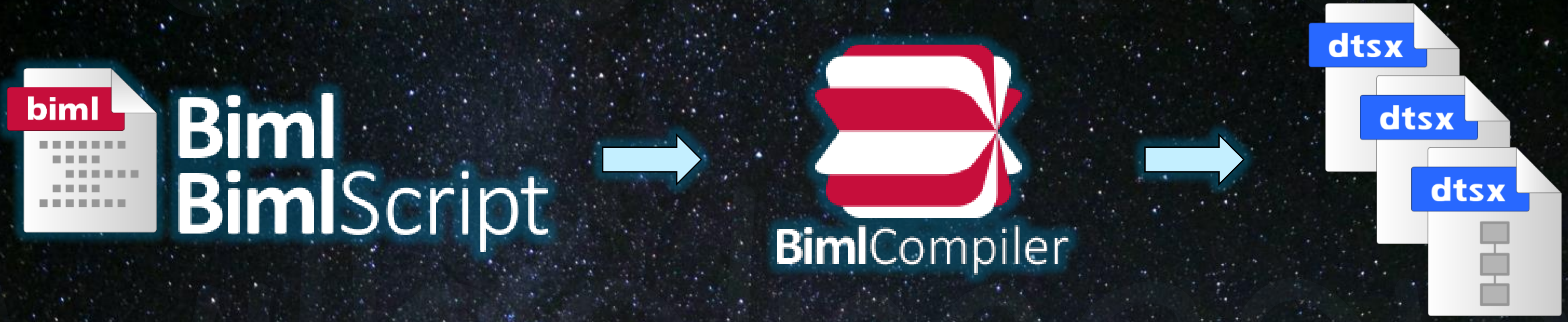
BimlScript Syntax: Control Nuggets

```
<Biml xmlns="http://schemas.varigence.com/biml.xsd">
  <Packages>
    <# foreach (var table in RootNode.Tables) { #>
      <Package Name="Load_<#=table.Name#>"></Package>
    <# } #>
  </Packages>
</Biml>
```

BimlScript Syntax: Text Nuggets

```
<Biml xmlns="http://schemas.varigence.com/biml.xsd">
  <Packages>
    <# foreach (var table in RootNode.Tables) { #>
      <Package Name="Load_<#=table.Name#>"></Package>
    <# } #>
  </Packages>
</Biml>
```

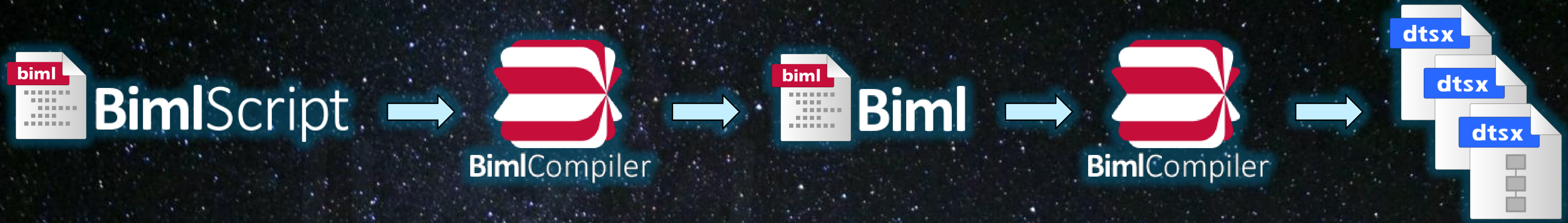
How does it work?



Biml
BimlScript

BimlCompiler

Yes, but how does it work?



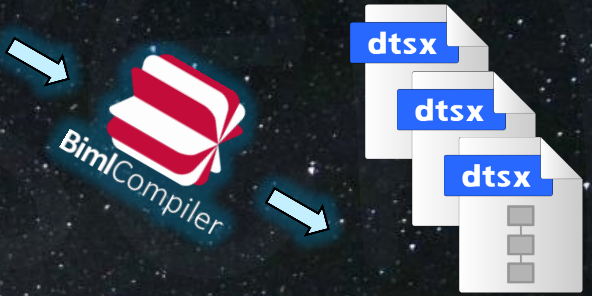
BimlScript → BimlCompiler → Biml → BimlCompiler → dtsx

Yes, but how does it *actually* work?

```
<Biml xmlns="http://schemas.varigence.com/biml.xsd">
  <Packages>
    <# foreach (var table in RootNode.Tables) { #>
      <Package Name="Load_<#=table.Name#>"></Package>
    <# } #>
  </Packages>
</Biml>
```

BimlCompiler

```
<Biml xmlns="http://schemas.varigence.com/biml.xsd">
    <Packages>
        <Package Name="Load_Customer"/>
        <Package Name="Load_Product"/>
        <Package Name="Load_Sales"/>
    </Packages>
</Biml>
```



Biml vs. BimlScript

Biml

Flat XML
"Just text"

BimlScript

Automate, control and
manipulate Biml with C#

Let's generate *a lot* of packages!

(demo time!)

Don't Repeat Yourself

Don't Repeat Yourself

Move common code to separate files

Centralize and reuse in many projects

Update code once for all projects

1. Include files
2. CallBimlScript with parameters
3. Tiered Biml files

BimlExpress vs. BimlOnline

BimlExpress

"Black Box"

Only SSIS packages visible

BimlOnline

Visual Editors

All in-memory objects visible

Include Files

Include common code in multiple files and projects

Can include many file types: .biml .txt .sql .cs

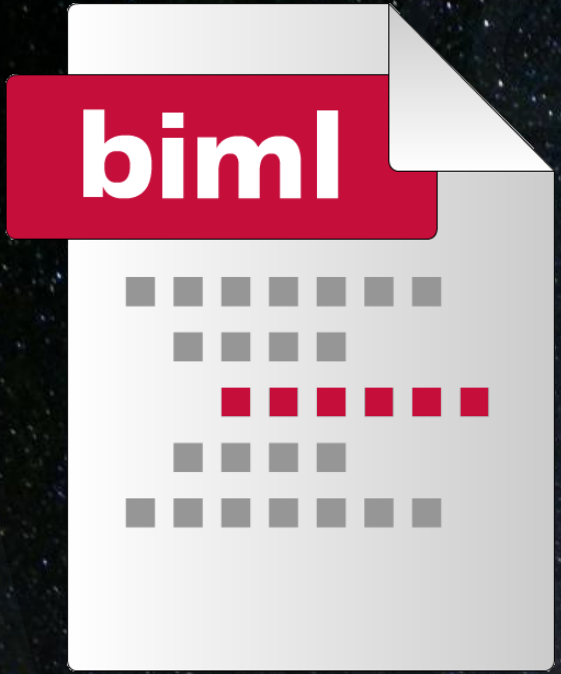
Use the include directive

```
<#@ include file="CommonCode.biml" #>
```

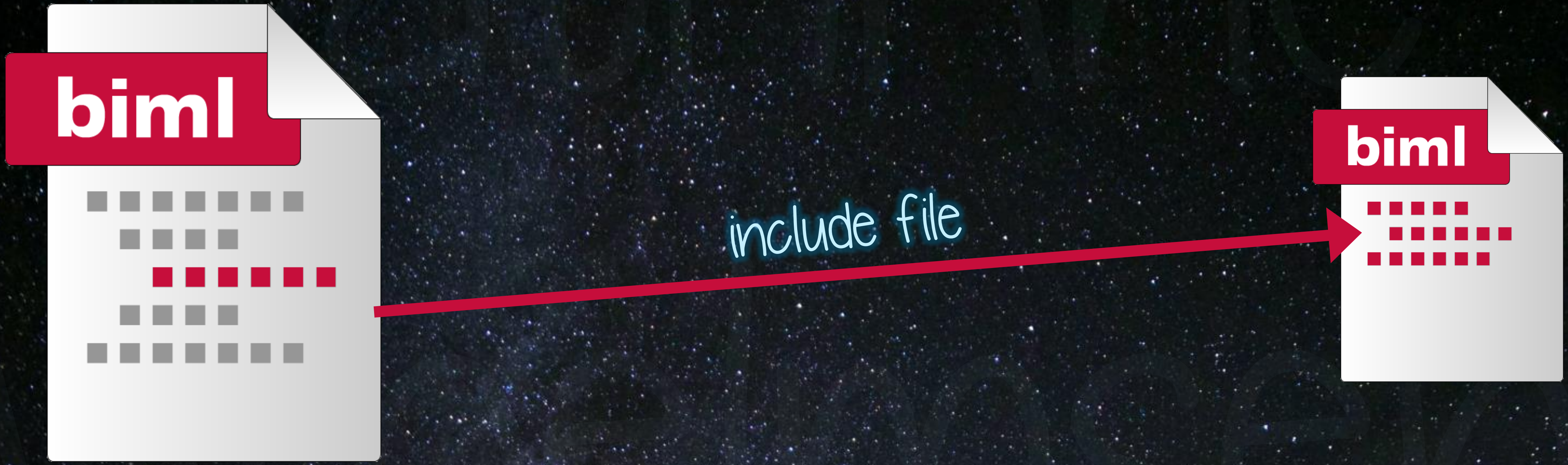
The directive will be replaced by the included file

Include *pulls* code from the included file into the main file

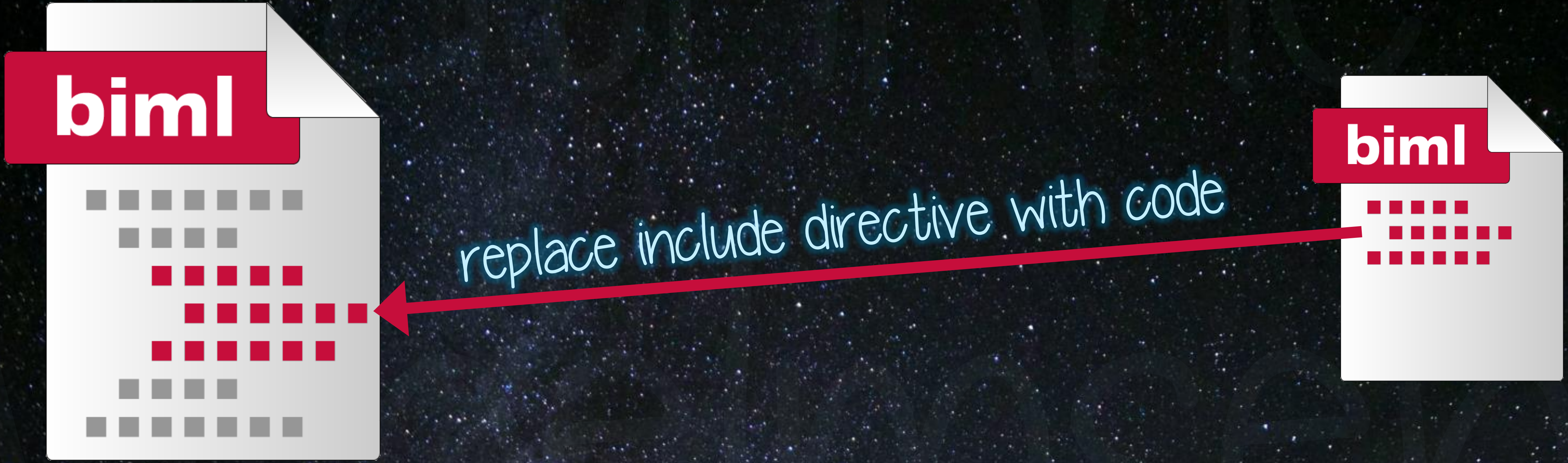
Include Files



Include Files



Include Files



CallBimlScript with Parameters

Works like a parameterized include

File to be called (*callee*) specifies input parameters it accepts

```
<#@ property name="Table" type="AstTableNode" #>
```

File that calls (*caller*) passes input parameters

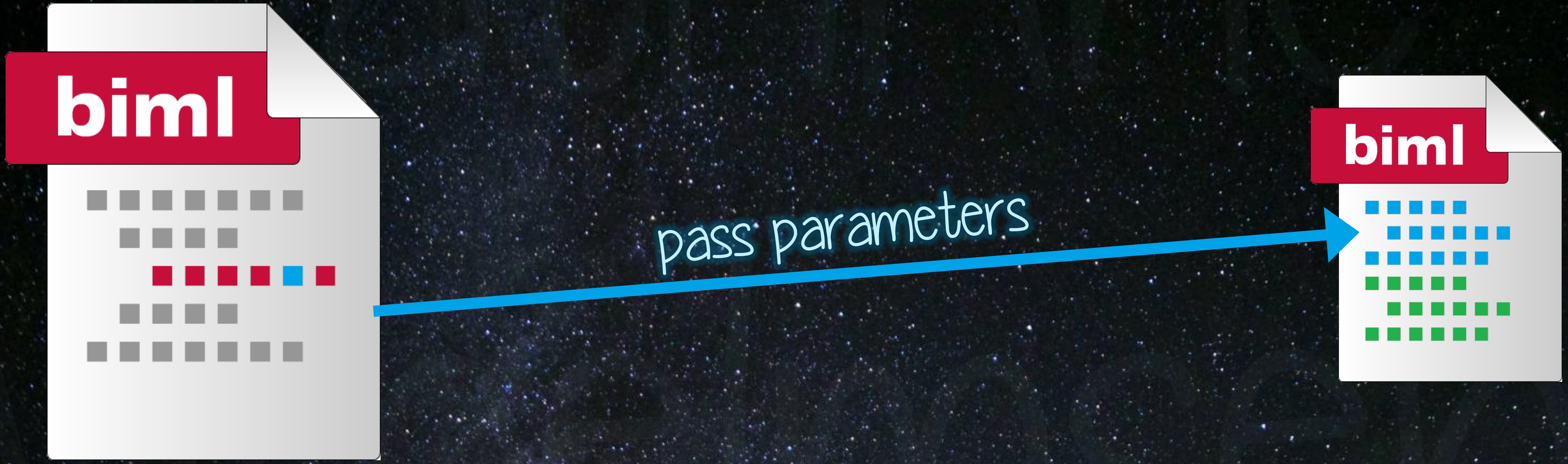
```
<#=CallBimlScript("CommonCode.biml", Table)#>
```

CallBimlScript *pushes* parameters from the caller to the callee, and the callee *returns* code

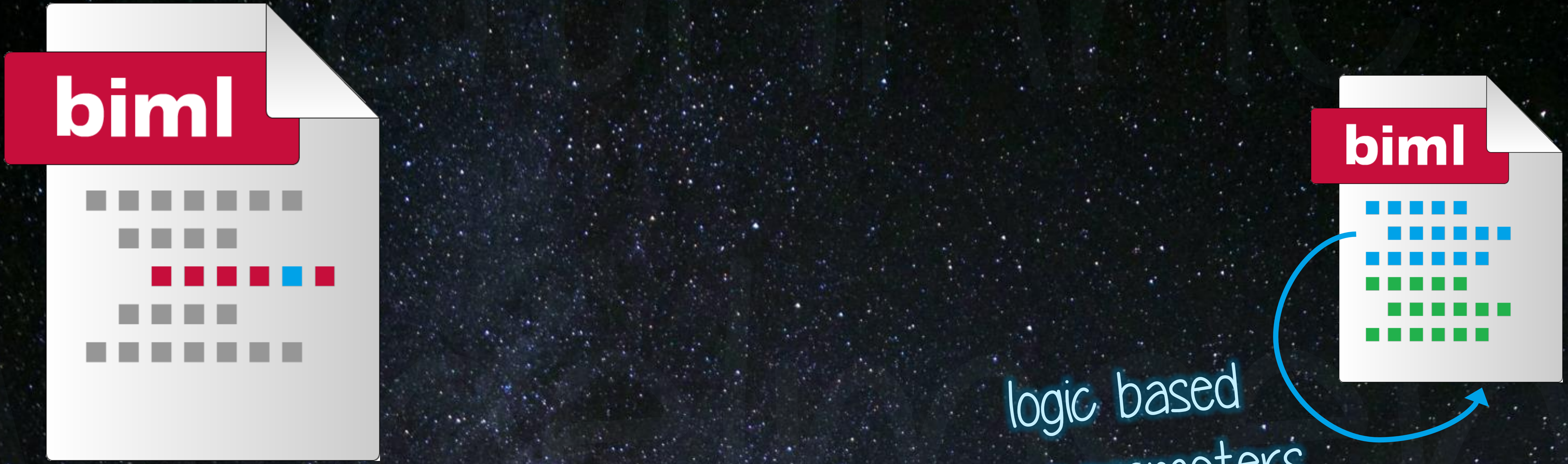
CallBimlScript with Parameters

biml

CallBimlScript with Parameters

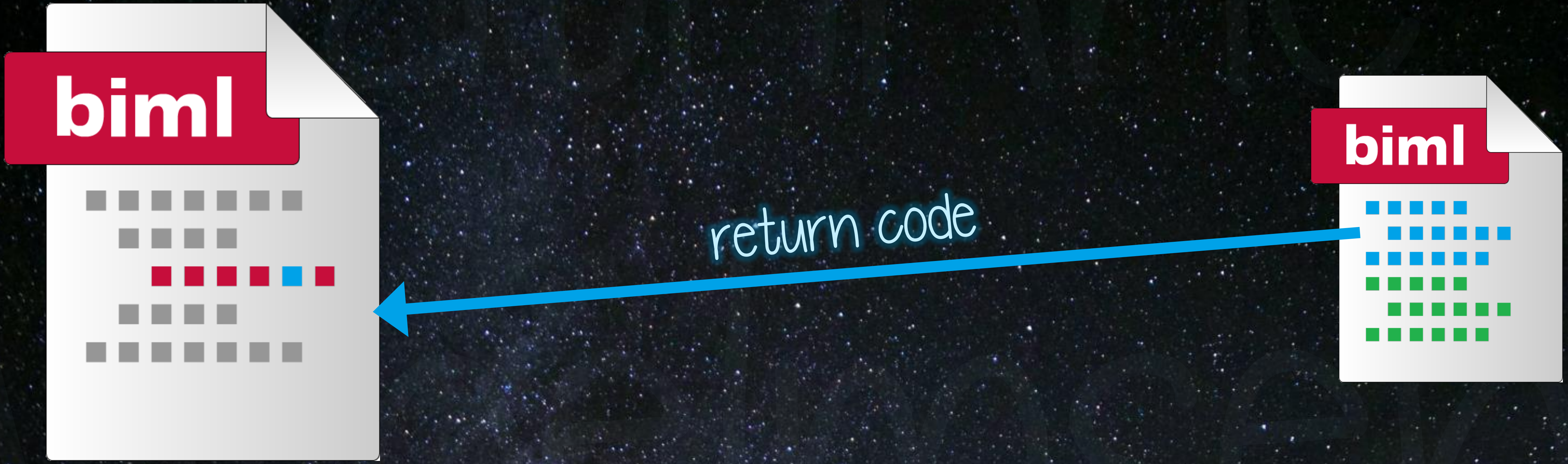


CallBimlScript with Parameters



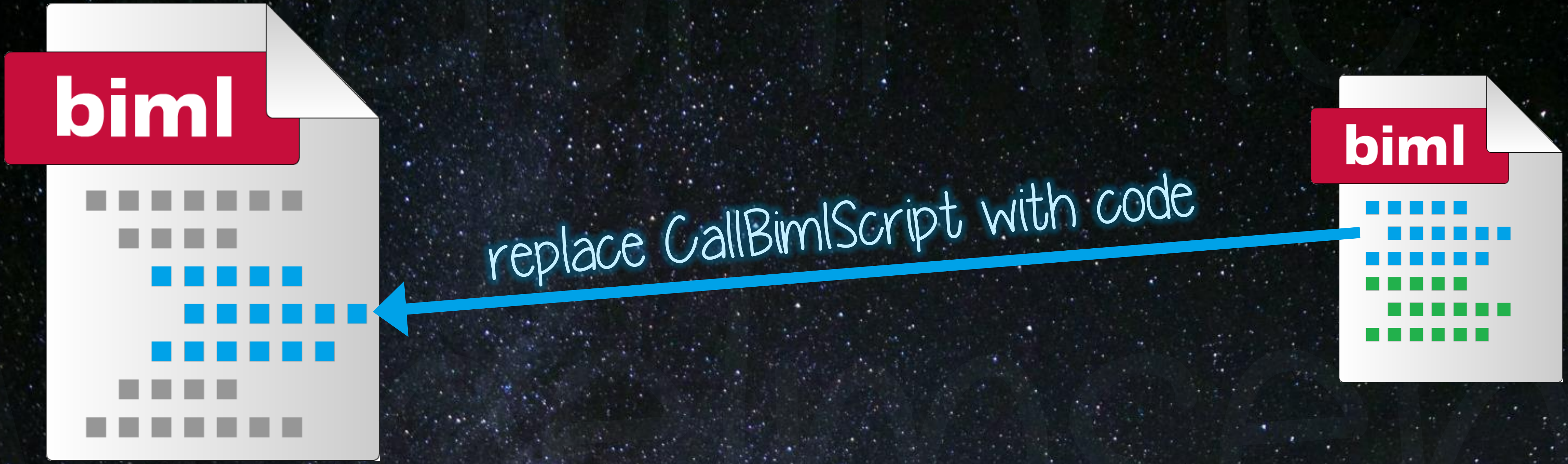
logic based
on parameters

CallBimlScript with Parameters



return code

CallBimlScript with Parameters



Tiered Biml Files

Use the template directive:

```
<#@ template tier="1" #>
```

Create objects in-memory from lowest to highest tier to:

- Solve logical dependencies
- Simulate manual workflows

In-memory objects are added to the *RootNode*

Get objects from *RootNode* in higher tiers

So how does this actually work?

(demo time!)

Get things done

Start small

Start simple

Start with ugly code

Keep going

Expand

Improve

Deliver often

Biml on Monday...

...BimlBreak the rest of the week :)

Biml resources and references:

cathrinewilhelmsen.net/biml

contact@cathrinewilhelmsen.net

@cathrinew

cathrinewilhelmsen.net

slideshare.net/cathrinewilhelmsen

linkedin.com/in/cathrinewilhelmsen