



GUIs and Toolmaking with PowerShell



David das Neves
Premier Field Engineer, Microsoft

A black and white photograph capturing a heavy snowfall scene. In the lower-left foreground, the back of a person wearing a dark coat and a hood is visible, walking away from the viewer. The background is filled with thick, falling snow, obscuring buildings and trees. Bare tree branches are visible on the left side of the frame. A street lamp post stands on the right, its light fixture partially visible through the falling snow.

David das Neves
PFE, Microsoft

Agenda

- Introduction
- Windows Forms
- XAML
- Further advices
- Recap

Why use GUI in Powershell?

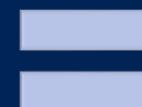
User-friendly UI



Better Information Presentation



Additional Logical Layer



Better User Experience



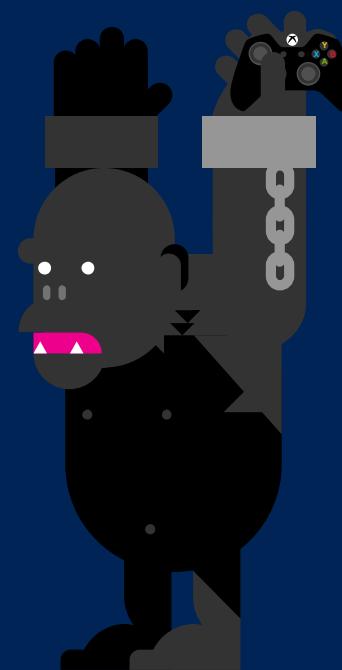
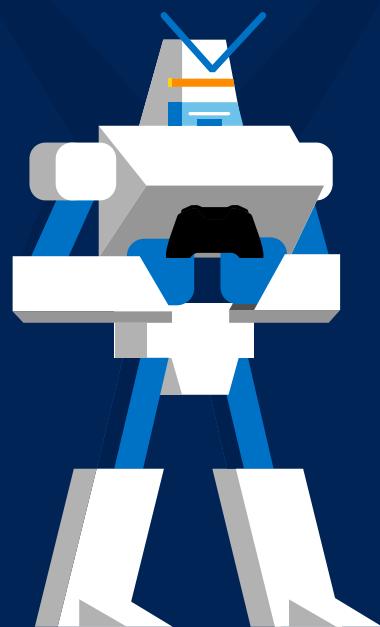
PSCONF.EU
POWERSHELL CONFERENCE EU

GUI Creation - Technical Ways

XAML

vs.

Windows Forms



PSCONF.EU
POWERSHELL CONFERENCE EU

Windows Forms

The old way.



PSCONF.EU
POWERSHELL CONFERENCE EU

Windows Forms - Workflow

Loading assemblies
Forms, Drawing

Adding Form

Creating and
Adding Objects

Adding Events
with Eventhandlers

Showing the
Form



PSCONF.EU
POWERSHELL CONFERENCE EU

```
# Load the Winforms assembly
[reflection.assembly]::LoadwithPartialName( "System.Windows.Forms")

# Create the form
$form = New-Object Windows.Forms.Form

#Set the dialog title
$form.text = "PowerShell WinForms Example"

# Create the label control and set text, size and location
$label = New-Object Windows.Forms.Label
$label.Location = New-Object Drawing.Point 50,30
$label.Size = New-Object Drawing.Point 200,15
$label.text = "Enter your name and click the button"

# Create TextBox and set text, size and location
$textfield = New-Object Windows.Forms.TextBox
$textfield.Location = New-Object Drawing.Point 50,60
$textfield.Size = New-Object Drawing.Point 200,30
```

```
# Create Button and set text and location
$button = New-Object Windows.Forms.Button
$button.text = "Greeting"
$button.Location = New-Object Drawing.Point 100,90

# Set up event handler to extarct text from TextBox and display it on
the Label.
$button.add_click({

    $label.Text = "Hello " + $textfield.text
})

# Add the controls to the Form
$form.controls.add($button)
$form.controls.add($label)
$form.controls.add($textfield)

# Display the dialog
$form.ShowDialog()
```

Tools

- Visual Studio
- PowerShell Pro Tools for Visual Studio
- <https://poshgui.com/>
- Sapien Powershell Studio



PSCONF.EU
POWERSHELL CONFERENCE EU

Windows Forms Tools Demos



PSCONF.EU
POWERSHELL CONFERENCE EU

XAML

The new way.



PSCONF.EU
POWERSHELL CONFERENCE EU

Introduction - XAML

- Declarative XML-like Markup-language
- Open and End-Tags
- Properties inline or nested
- Listing and nesting of objects
- Bindings
- Styles
- Templates

XAML - Workflow

Creation of
XAML

Modifications
to XAML

Loading
assemblies
Presentation
Framework

Loading
XAML

Creating
variables for
all used
objects

Events and
showing the
window /
dialog



PSCONF.EU
POWERSHELL CONFERENCE EU

XAML

ShowDialog (used in most examples in Internet)

- Modal
- Returns value

Show + ApplicationContext + Application.Run

- Sometimes more stable
- Runs the main parent form, and makes that form the main form.

Application run window

- Crashes if used Forms + WPF

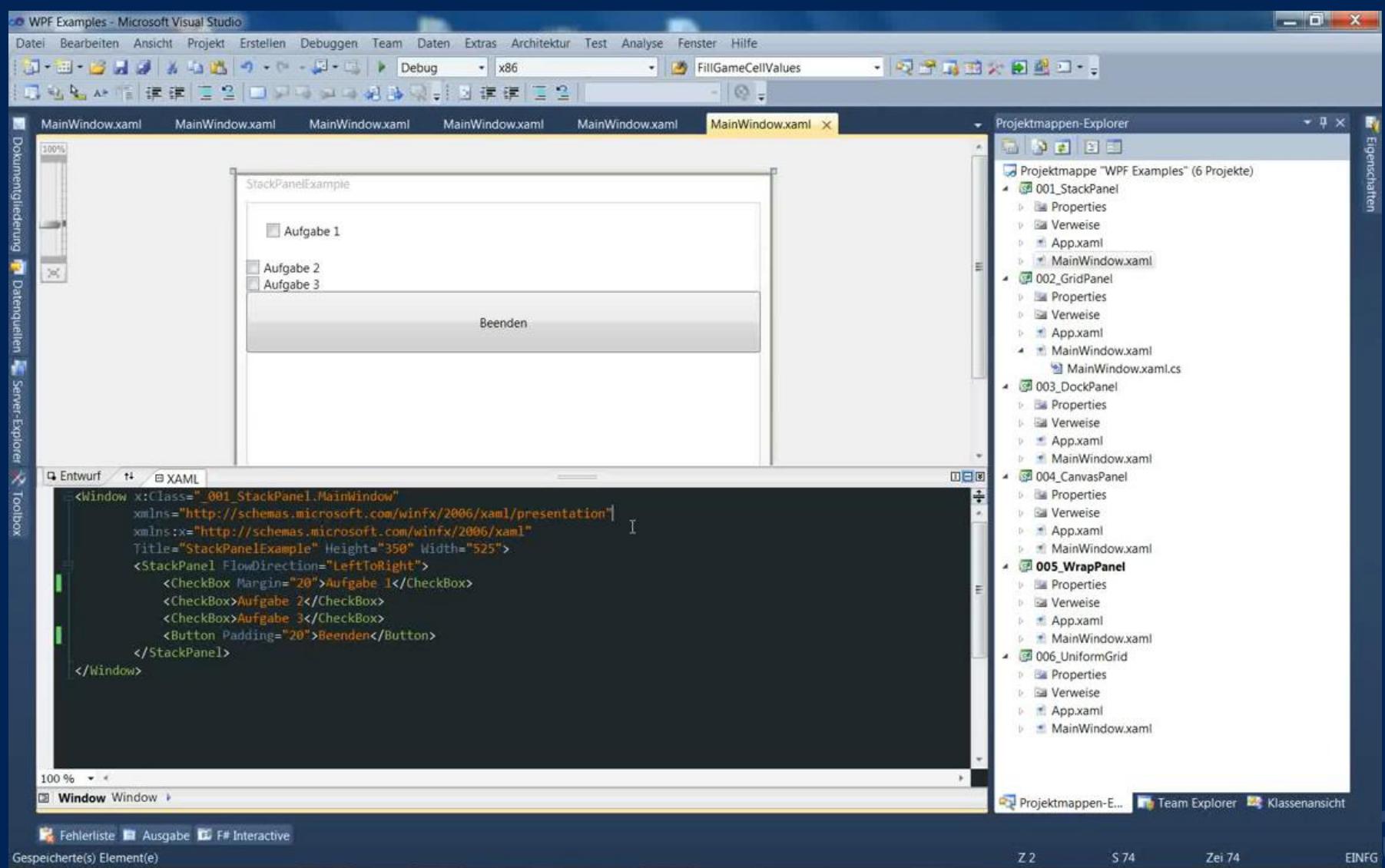
Using Runspace



#Loading window – Best Practice

```
$window.Dispatcher.InvokeAsync{  
    $window.ShowDialog()  
}.Wait()
```

XAML - Container



XAML - Container

- Canvas
 - Simplest, placement relative to two edges
- StackPanel
 - Horizontal or vertical stacking
- Grid
 - Uses rows and columns
- DockPanel
 - Dock to top, right, bottom, left, and all else fills remaining space
- WrapPanel
 - Horizontal stacking with wrap on overflow
- All of these can be nested, any one in another

XAML - Visualization

- In WPF there is only (usually) one window
 - Controls are not windows!
 - No handles - really, no handles
 - A button is a shape with border, fill, text, animation, and events, like click.
 - There is a Button class, but it is not a .Net control in the traditional sense nor an ActiveX control.
 - Just markup, lines, fills, and events.

XAML – Parse Tree

- XAML gets rendered into a parse tree, just like XML – it is XML
 - Inherited properties are based on parent child relationships in the markup tree
 - Events bubble based on those relationships as well
 - You have direct and simple control over that structure

XAML - Routed Events

- WPF maps markup elements to UIElements, which derive from ContentControl
 - That means that almost everything can hold content – only one thing unless it's a panel.
 - How does a mouse click *event* on any one of a control's content elements get *routed* to the control?
 - By walking the XAML parse tree until it finds a parent that handles that event.



XAML - Control Templates

- Use Control Templates to change the look of existing controls or create your own controls

```
<Button.Template>
  <ControlTemplate>
    <Grid><Rectangle /></Grid>
  </ControlTemplate>
</Button.Template>
```

XAML - Special UIElements

- **ViewBox**
 - Resizes content to fit available space
- **UserControl**
 - Way to build custom controls as collections of elements on a panel
- **Animatable**
 - Provides hooks for DirectX to change elements properties over time, e.g.: position, size, color
- **MediaElement**
 - Play media on load or on request, e.g., wma, wmv, mp3

Demo

Simple XAML GUI

Operating System Details	
Hostname	[REDACTED]
Operating System Name	Microsoft Windows 10 Pro
Available Memory	11815 MB
OS Architecture	64-Bit
Windows Directory	C:\WINDOWS
Windows Version	10.0.10240
System Drive	C:

[Exit](#)

```

=====
# XAML Demo-Applikation
=====

# XAML from visual studio without class
=====
[xml]$XAML = @'
<window
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Title="OS Details" Height="306" Width="525" WindowStartupLocation="CenterScreen" WindowStyle='None' ResizeMode='NoResize'>
    <Grid Margin="0,0,-0.2,0.2">
        <TextBox HorizontalAlignment="Center" Height="23" TextWrapping="Wrap" Text="Operating System Details" VerticalAlignment="Top" Width="525" Margin="0,-1,-0.2,0" TextAlignment="Center" Foreground="white" Background="#FF98D6EB"/>
        <Label Content="Hostname" HorizontalAlignment="Left" Margin="0,27,0,0" VerticalAlignment="Top" Height="30" Width="170" Background="#FF98D6EB" Foreground="white"/>
        <Label Content="Operating System Name" HorizontalAlignment="Left" Margin="0,62,0,0" VerticalAlignment="Top" Height="30" Width="170" Background="#FF98D6EB" Foreground="white"/>
        <Label Content="Available Memory" HorizontalAlignment="Left" Margin="0,97,0,0" VerticalAlignment="Top" Height="30" Width="170" Background="#FF98D6EB" Foreground="white"/>
        <Label Content="OS Architecture" HorizontalAlignment="Left" Margin="0,132,0,0" VerticalAlignment="Top" Height="30" Width="170" Background="#FF98D6EB" Foreground="white"/>
        <Label Content="Windows Directory" HorizontalAlignment="Left" Margin="0,167,0,0" VerticalAlignment="Top" Height="30" Width="170" Background="#FF98D6EB" Foreground="white"/>
        <Label Content="Windows Version" HorizontalAlignment="Left" Margin="0,202,0,0" VerticalAlignment="Top" Height="30" Width="170" Background="#FF98D6EB" Foreground="white"/>
        <Label Content="System Drive" HorizontalAlignment="Left" Margin="0,237,0,0" VerticalAlignment="Top" Height="30" Width="170" Background="#FF98D6EB" Foreground="white"/>
        <Button Name="btnExit" Content="Exit" HorizontalAlignment="Left" Margin="0,272,0,0" VerticalAlignment="Top" Width="525" Height="34" BorderThickness="0"/>
        <TextBox Name="txtHostName" HorizontalAlignment="Left" Height="30" Margin="175,27,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="343" IsEnabled="False"/>
        <TextBox Name="txtOSName" HorizontalAlignment="Left" Height="30" Margin="175,62,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="343" IsEnabled="False"/>
        <TextBox Name="txtAvailableMemory" HorizontalAlignment="Left" Height="30" Margin="175,97,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="343" IsEnabled="False"/>
        <TextBox Name="txtOSArchitecture" HorizontalAlignment="Left" Height="30" Margin="175,132,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="343" IsEnabled="False"/>
        <TextBox Name="txtWindowsDirectory" HorizontalAlignment="Left" Height="30" Margin="175,167,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="343" IsEnabled="False"/>
        <TextBox Name="txtWindowsVersion" HorizontalAlignment="Left" Height="30" Margin="175,202,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="343" IsEnabled="False"/>
        <TextBox Name="txtSystemDrive" HorizontalAlignment="Left" Height="30" Margin="175,236,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="343" IsEnabled="False"/>
    </Grid>
</window>
'@

```

```

# Assembly
#=====
[void][System.Reflection.Assembly]::LoadWithPartialName('presentationframework')

# Loading XAML with XAML-Reader
#=====
$reader=(New-Object System.Xml.XmlNodeReader $xaml)
try{$Form=[Windows.Markup.XamlReader]::Load( $reader )}
catch{write-Host 'Unable to load Windows.Markup.XamlReader. Some possible
causes for this problem include: .NET Framework is missing PowerShell must be
launched with PowerShell -sta, invalid XAML code was encountered.'; exit}

# Variables for all objects
#=====
$xaml.SelectNodes('//*[@Name]') | %{$_.Name}
-value $Form.FindName($_.Name)}

# Adding events
#=====
$btnExit.Add_Click({$form.Close()})

```

```
# Fill GUI
# WMI
$owMIOS = Get-WmiObject win32_OperatingSystem

# Filling Textboxes
$txtHostName.Text = $owMIOS.PSComputerName

#Formats and displays OS name
$aOSName = $owMIOS.name.Split('|')
$txtOSName.Text = $aOSName[0]

#Formats and displays available memory
$sAvailableMemory = [math]::round($owMIOS.freephysicalmemory/1000,0)
$sAvailableMemory = "$sAvailableMemory MB"
$txtAvailableMemory.Text = $sAvailableMemory

#Displays OS Architecture
$txtOSArchitecture.Text = $owMIOS.OSArchitecture

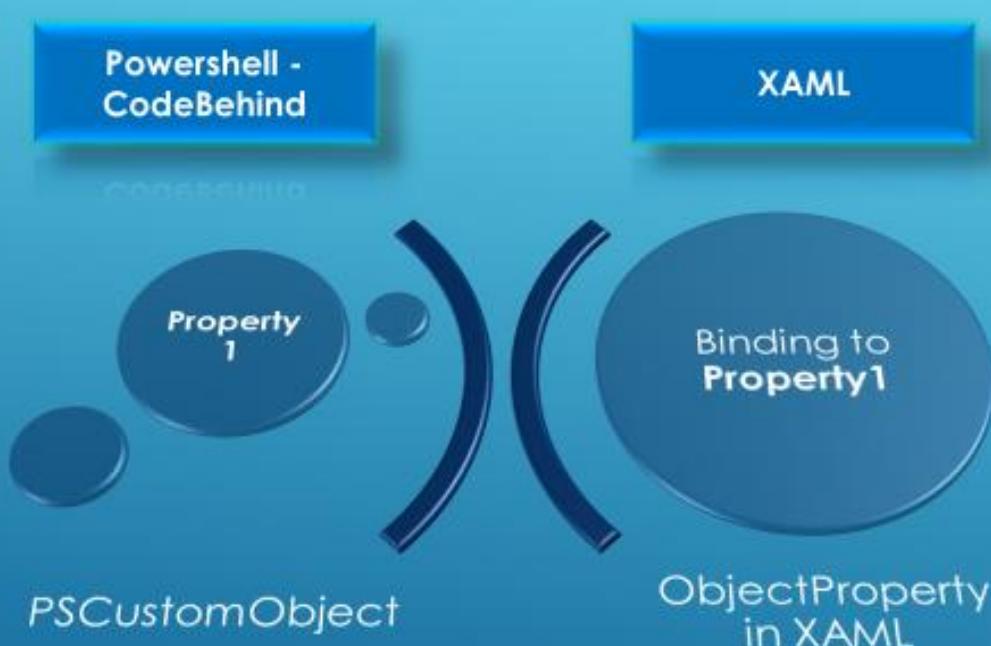
#Displays Windows Directory
$txtwindowsDirectory.Text = $owMIOS.windowsDirectory

#Displays Version
$txtwindowsVersion.Text = $owMIOS.Version

#Displays System Drive
$txtSystemDrive.Text = $owMIOS.SystemDrive

#Formular anzeigen
$Form.ShowDialog() | Out-Null
```

Bindings



XAML Demos



PSCONF.EU
POWERSHELL CONFERENCE EU

Tools

- PSGUI
- ISE + ISEsteroids
- Visual Studio

PSGUI - Main Aims

Project PSGUI

PSGUI

- Loading Framework
- Automation integrated

PSGUIManager

- Managing / Creation of multiple GUIs
- Examples
- Debugging
- Deployment



PSCONF.EU
POWERSHELL CONFERENCE EU

PSGUI - Main Aims

Easy Usage

Structured and separated Code

Easy to **create**, **load**, **manage**
and **deploy** Dialogs

Free Code

Examples



PSGUI - Technical Parts

Removing class of XAML

NameSpace integration of XAML

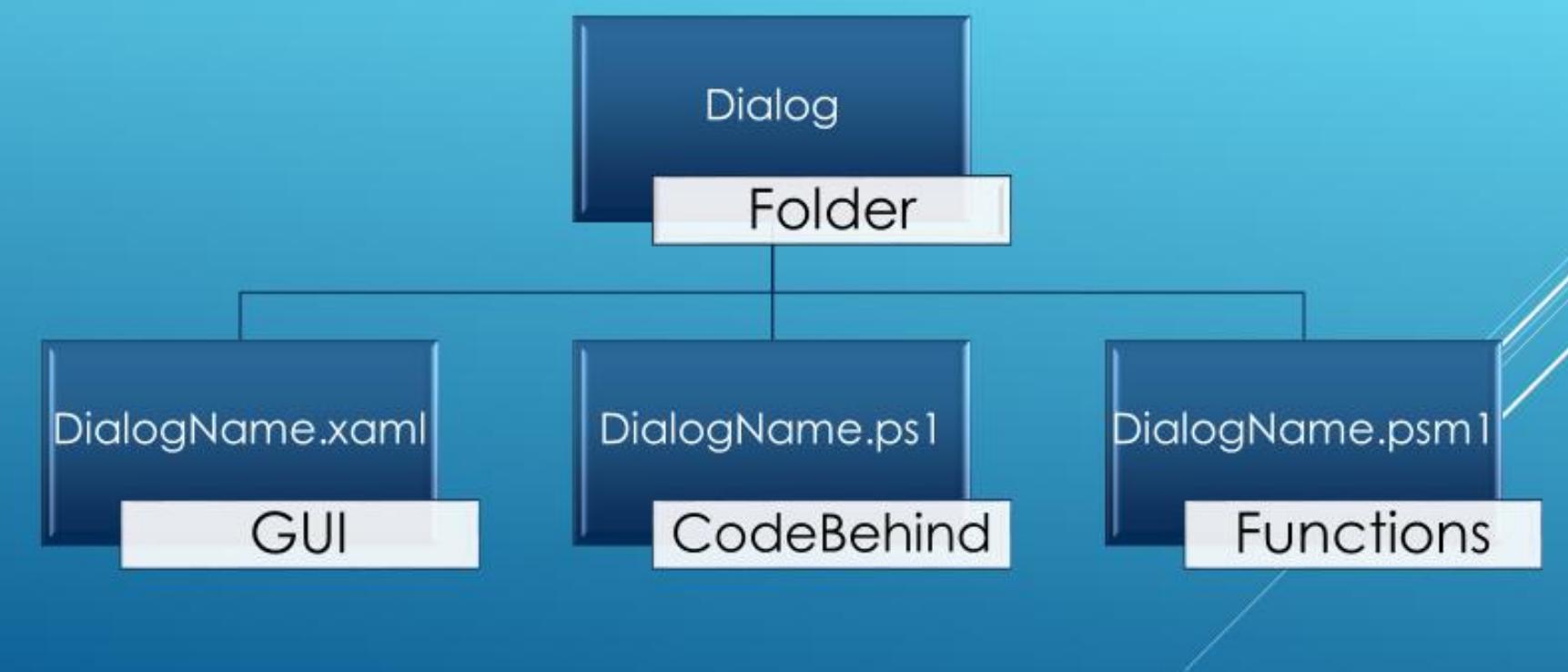
Fixing relative and absolute links

Folder Resources

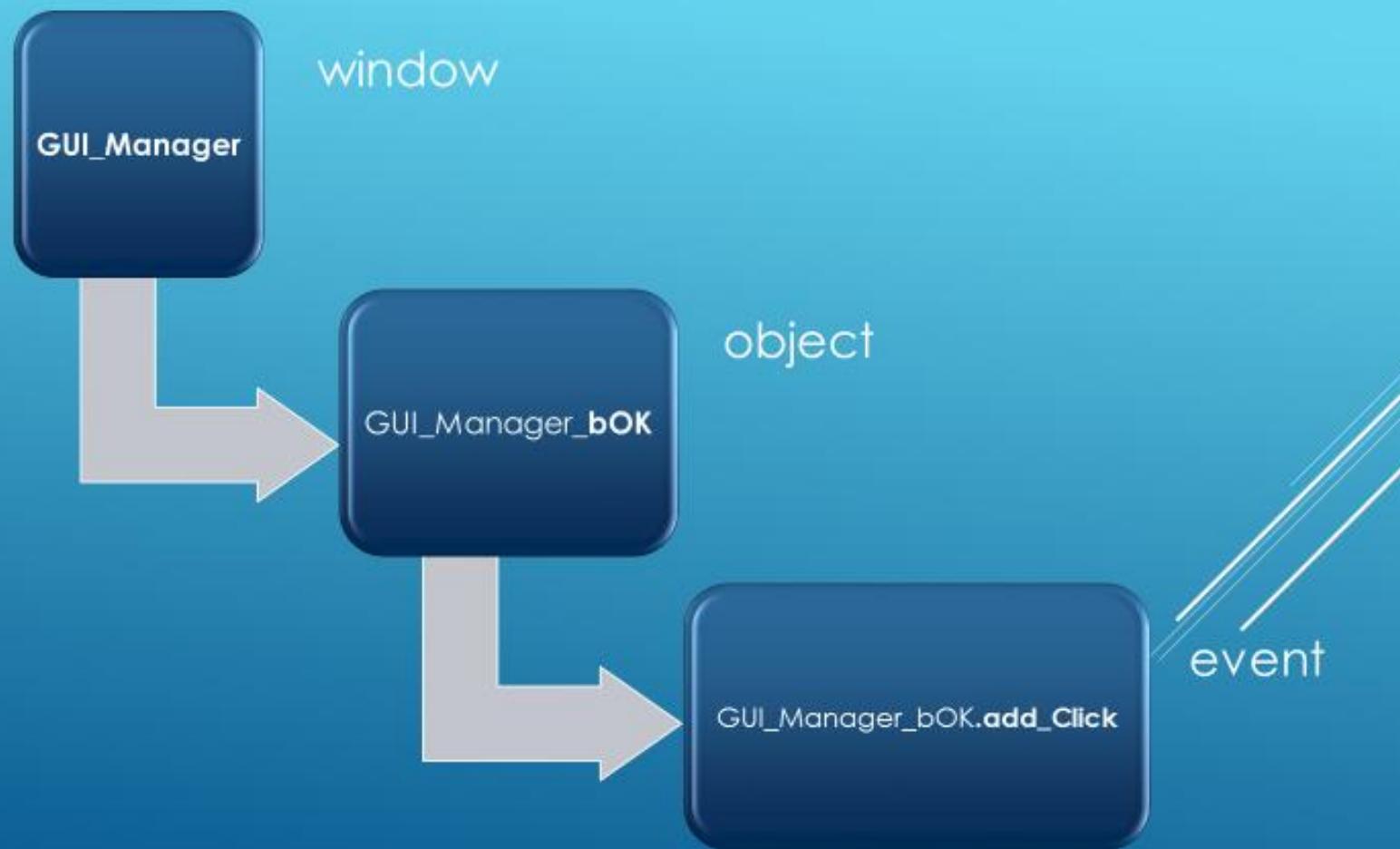
Debugging Assistance

Stable GUIs

PSGUI - Folder Structure



PSGUI - Window Structure



PSGUI - PSGUIManager

The screenshot shows the PSGUIManager application window. The top navigation bar includes 'File', 'Dialogs', and 'About' tabs. The 'Dialogs' tab is active, displaying a list of dialog examples: 01_UserInput, 02_PCIInformation, 03_ShowDialogResult, 04_Bindings1, 05_Bindings2, 06_Bindings3, 07_Bindings_Complete, 08_AddColumnToListviewwithnewBinc, and 09_Styles_Property_Trigger. Below this is a 'Variables' section with a table:

Name	Objekt
SCCM_LogAnalyzer	Window
SCCM_LogAnalyzer_Fenster	Window
SCCM_LogAnalyzer_lvData	ListView Items.Count:0

The 'Events' section is currently empty. At the bottom, there are tabs for 'DialogFiles', 'XAML' (which is selected), 'CodeBehind', and 'Functions'. The 'XAML' tab displays the following XAML code:

```
1 <window
2   xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3   xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4   Name="Fenster"
5   ResizeMode="CanResizeWithGrip"
6   Title="SCCM_LogAnalyzer"
7   Topmost="True" Width="1200" Height="600">
8     <Window.Resources>
9       <Style TargetType="{x:Type ListViewItem}">
10         <Setter Property="HorizontalContentAlignment" Value="Stretch"/>
11       </Style>
12     </Window.Resources>
13     <Grid>
14       <Grid.ColumnDefinitions>
15         <ColumnDefinition Width="1*"/>
16       </Grid.ColumnDefinitions>
17       <Grid.RowDefinitions>
18         <RowDefinition Height="1*"/>
19       </Grid.RowDefinitions>
20       <!--<Button Name="btnStop" Content="Stop" Grid.Column="0" HorizontalAlignment="Right" Margin="10" Grid.Row="0" VerticalAlignment="Bottom" /-->
21       <ListView Name="lvData" Grid.Column="0" Grid.Row="0" Margin="15" Height="Auto" ScrollViewer.HorizontalScrollBarVisibility="Hidden">
22         <ListView.View>
23           <GridView>
24             <GridViewColumn Header="#" Width="40">
25               <GridViewColumn.CellTemplate>
26                 <DataTemplate>
27                   <TextBlock Text="{Binding Path=RowNum}" TextAlignment="Left" Width="Auto" Background="{Binding Path=Set<span style="color: red;">erColor}"/>
28                 </DataTemplate>
29               </GridViewColumn.CellTemplate>
30             </GridViewColumn>
31             <GridViewColumn Header="LogEntry" Width="450">
32               <GridViewColumn.CellTemplate>
```

On the right side of the XAML editor, there is a 'Open in ISE' button.

XAML Tools Demos

JEA Helper

JEA Helper Tool

Create or Edit Role Capability Role Capabilities Design Configurations Listing, Mapping and Testing SDDL Helper

In this tab, you can create the VisibleCmdlets section of Role Capabilities, and copy/paste them in your files or the first tab

You can start from... Existing role capability... Audit log Replace grid

Or you can pick a cmdlet and - optionally - properties Add to Grid

Or you can add a full/partial module, or use it to filter the cmdlets list Add to Grid Add Get-* only Filter Cmdlets Remove Filter

Module to import Import Module

Or you can pick SMA Runbook(s) Add to Grid

Module	Name	Parameter	ValidateSet	ValidatePattern
	Get-Event			
	Add-Computer	ComputerName		
	Add-Computer	Credential		
	Add-Computer	DomainName		
	Add-Computer	OUPath		
AppLocker	Get-*			
Defender	*			
	Get-AppLockerFileInfo			

Add Row Remove Selected Row(s) Remove All Rows Refresh Role Capability Output

```
VisibleCmdlets='Get-Event',  
@{Name = 'Add-Computer'; Parameters=@{Name='ComputerName'}, @{Name='Credential'}, @{Name='DomainName'}, @{Name='OUPath'}},  
'AppLocker\Get-*',  
'Defender*',  
'Get-AppLockerFileInfo'  
  
VisibleFunctions=
```

Copy to Clipboard



Advanced Demo – PSConfEU 2017 Session List

PSConfEU 2017 - Sessionlist by David das Neves

Filtering: Grouping: [Clear](#) [Day Start](#) [Speaker](#) [Day Track](#) [Audience Day Start](#)

Day	Date	Start	End	Title	Speaker	Room	Audience	Track
Tuesday (10 items)								
Tuesday	2017-05-02	08:00	09:00	Preconf Registration	All None	Workshop Area	German,English	Commons
Tuesday (9:00 (4 items))								
Tuesday	2017-05-02	09:00	12:00	DSC Introduction	Ravikanth Chaganti	24+26	English	DSC and DevOps
Tuesday	2017-05-02	09:00	12:00	PowerShell Remoting	Aleksandar Nikolic	11+13	English	PowerShell
Tuesday	2017-05-02	09:00	12:00	PowerShell Security Einführung	David das Neves	25	German	Security
Tuesday	2017-05-02	09:00	12:00	Nanoversion und Container	Rinon Belegu	12+14	German	NanoServer and Po...
Tuesday (12:00 (1 item))								
Tuesday	2017-05-02	12:00	13:00	Lunch	All None	Lunch Areas	German,English	Food
Tuesday (13:00 (4 items))								
Tuesday	2017-05-02	13:00	16:00	Writing world-class DSC resource modules	Ravikanth Chaganti	24+26	English	DSC and DevOps
Tuesday	2017-05-02	13:00	16:00	JEA and Security	Aleksandar Nikolic	11+13	English	Security
Tuesday	2017-05-02	13:00	16:00	GUIs und Toolmaking mit PowerShell	David das Neves	25	German	PowerShell
Tuesday	2017-05-02	13:00	16:00	Chocolatey, PowerShellGet und Paketverwaltung	Rinon Belegu	12+14	German	Cross-Platform and...
Wednesday (28 items)								
Wednesday	2017-05-03	08:00	09:00	Registration		Foyer Leibnitz S...	English	Commons
Wednesday (09:00 (1 item))								
Wednesday	2017-05-03	09:00	09:15	Opening Ceremony	Tobias Weltner	Leibnitz Saal	English	Automation
Wednesday (09:15 (1 item))								
Wednesday	2017-05-03	09:15	10:00	Keynote: State of the Union	Jeffrey Snover	Leibnitz Saal	English	PowerShell Team
Wednesday (10:00 (1 item))								
Wednesday	2017-05-03	10:00	10:15	Coffee Break	All None	Leibnitz Saal	English	Coffee
Wednesday (10:15 (1 item))								
Wednesday	2017-05-03	10:15	11:00	PowerShell Warm-Up: Quiz & Quirks	Tobias Weltner	Leibnitz Saal	English	PowerShell
Wednesday (11:00 (1 item))								
Wednesday	2017-05-03	11:00	11:45	Catch Me If You Can - PowerShell Red vs. Blue	Will Schroeder	Leibnitz Saal	English	Security



PSCONF.EU
POWERSHELL CONFERENCE EU

May 3-5, 2017, Hannover
40+ Speakers
80 Sessions

Session: Preconf Registration

Audience: German,English

Speaker: All None

Room: Workshop Area **Day:** Tuesday

Start: 08:00 **End:** 09:00

Description

We meet in the main workshop area where you get coffee and can register. This day is open for delegates who registered for the preconference day. It is not open for anyone else.

Recap

Summary and Additional Advices

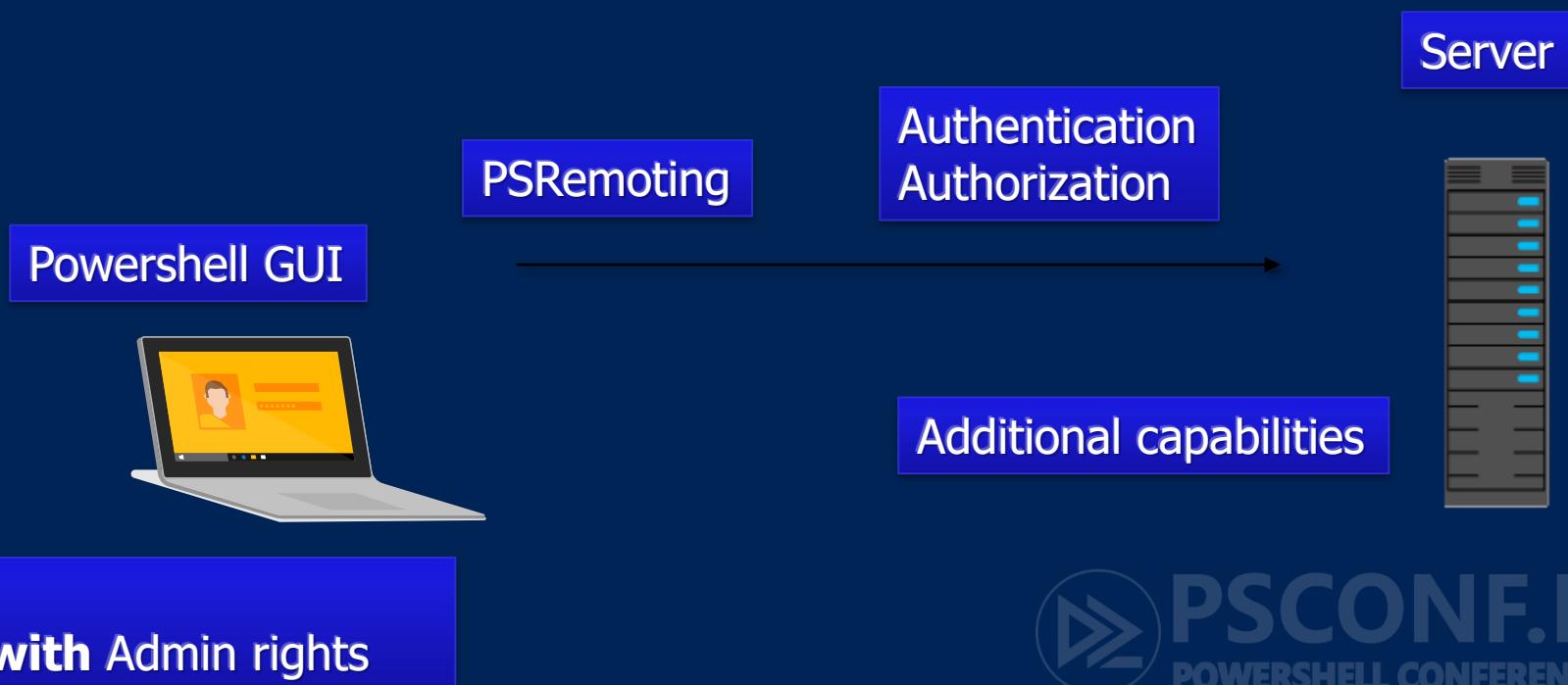
Code Design – some hints

- Sort / manage Code
 - Keep file structured with #regions
- Use Module to show/load window
- Separate external code
- Beware of unmanageable code / plan your resulting window
- Beware of unstable GUIs

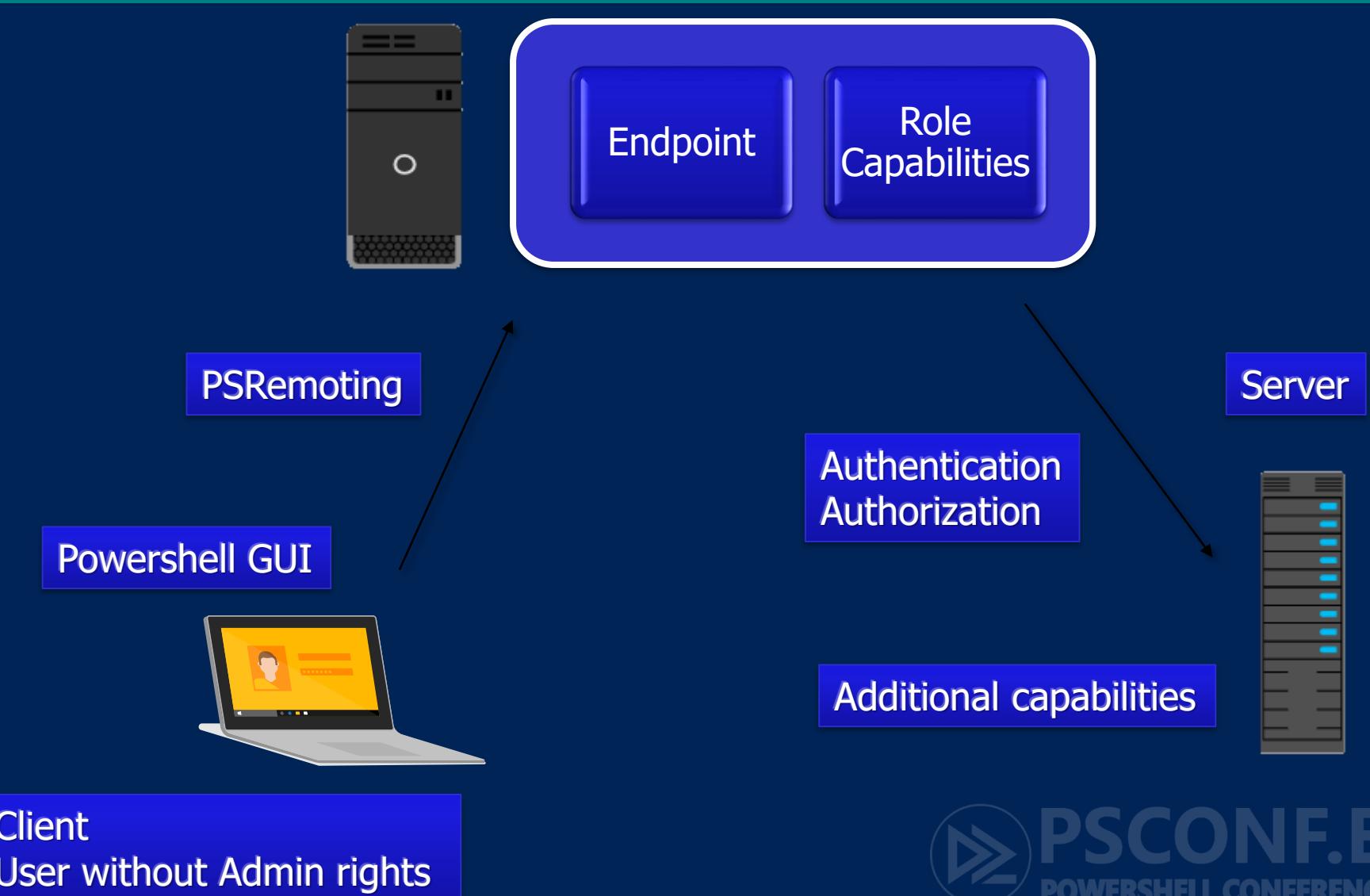


PSCONF.EU
POWERSHELL CONFERENCE EU

Extend the GUI with JEA 1/2



Extend the GUI with JEA 2/2



Open Discussion

- Pros and Cons
- Use Cases
- Examples from the Field
- Further Improvements
- Current Usage of GUIs
- Roadmap Powershell
 - Phosphor



PSCONF.EU
POWERSHELL CONFERENCE EU

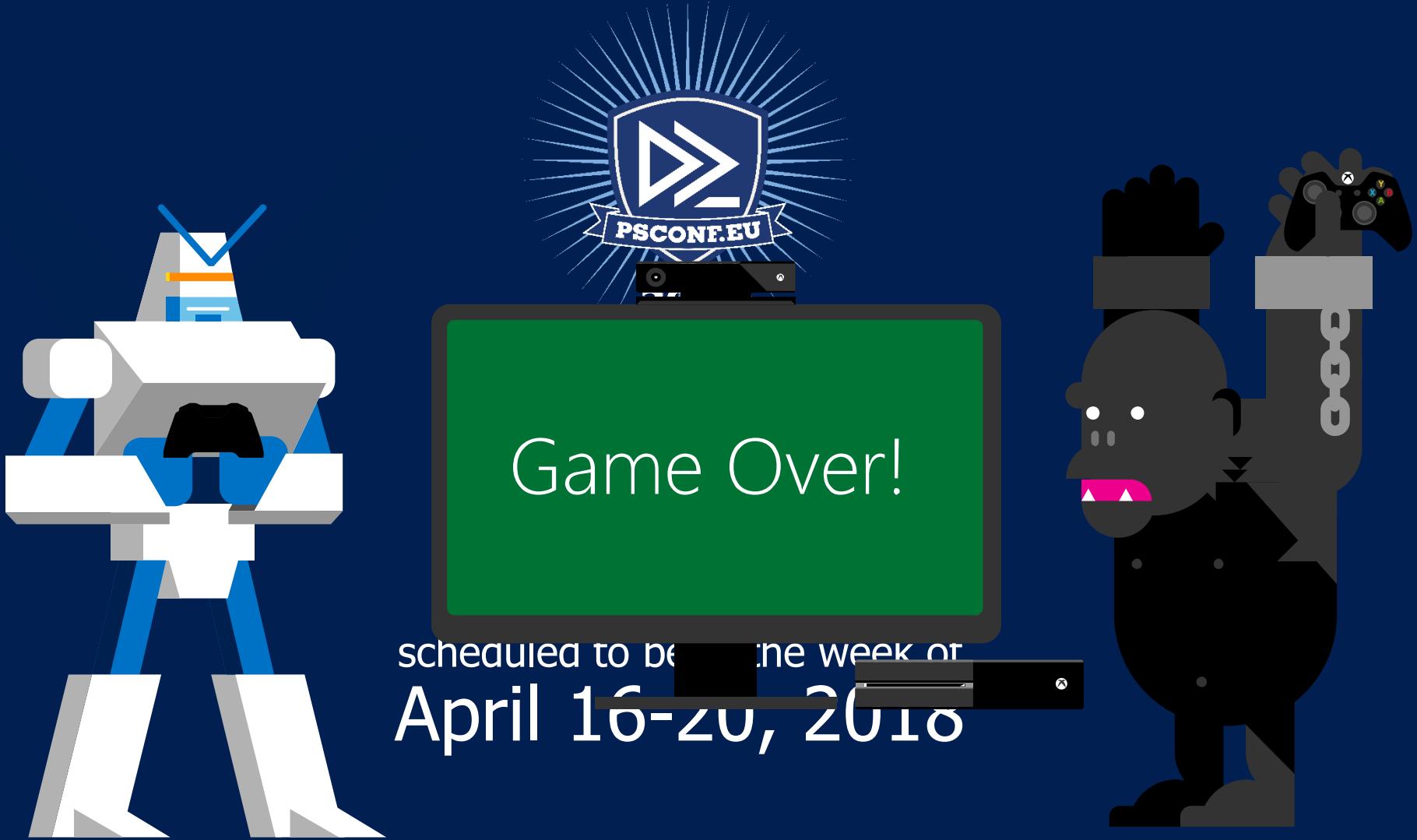
Summary

- Add a logical layer
- Use Visual Studio for easy XAML-creation
- Structure your code
- Use Bindings for effective GUI-population

Questions?

Next Steps...

- Now: 15 min break
- Grab a coffee
- Stay here to enjoy next presentation
- Change track and switch to another room
- Ask me questions or meet me in a breakout session room afterwards



scheduled to be in the week of
April 16-20, 2018

details on www.psconf.eu as they become available