

Protection and Licensing .NET Applications

CodeMeter Protection Suite

**SECURITY
LICENSING**

PERFECTION IN PROTECTION



Rüdiger Kügler, VP Professional Services
ruediger.kuegler@wibu.com

Axel Engelmann, Architect Protection Technologies
axel.engelmann@wibu.com

To access the on-demand replay of this masterclass, please visit

www.wibu.com//wibu-systems-webinars/protecting-and-licensing-net-applications/access.html

Disassembling a .NET Application

Telerik JustDecompile - DefaultAssemblyList - (3 files)

Assembly List: CalculatePi (NET 2.0/3.5)/Any CPU

- CalculatePi.exe
 - References
 - <Default namespace>
 - CalculatePi
 - Form1
 - Base Types
 - Derived Types
 - components : IContainer
 - btnCalculatePi : Button
 - tbDecimalPlaces : TextBox
 - label1 : Label
 - tbPi : TextBox
 - label2 : Label
 - pbStatus : ProgressBar
 - btnCancel : Button
 - label3 : Label
 - lblUsedTime : Label
 - lblRemaingTime : Label
 - label5 : Label
 - StartTime : DateTime
 - Percent : Double
 - bwCalculate : BackgroundWorker

Assembly Name: CalculatePi, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null
Namespace: CalculatePi

```

147 private void InitializeComponent() ...
246
247 private class Pi
248 {
249     public uint ulDecimalPlaces;
250
251     public Pi()...
254
255     public double CalculatePi(object sender, DoWorkEventArgs e)
256     {
257         TimeSpan tsRemainingTime;
258         TimeSpan tsTotalTime;
259         double num;
260         int iOldProgress = 0;
261         int iOldTotalSeconds = 0;
262         DateTime StartTime = DateTime.Now;
263         BackgroundWorker bwCalculate = sender as BackgroundWorker;
264         double dSum = 0;
265         double dCurrentY = 0;
266         double dPercent = 0;
267         ulong ullSteps = (long)5 * (ulong)Math.Pow(10, (double)((float)this.ulDecimalPlaces));
268         Form1.UserState usStatus = new Form1.UserState();
269         ulong i = (ulong)1;
270         while (true)
271         {
272             if (i < ullSteps)
273             {
274                 if ((i % (long)10000 == (long)0 ? true : i == ullSteps - (long)1))
275                 {
276                     dPercent = (double)((float)(i + (long)1)) * 100 / (double)((float)ullSteps);
277                     int iNewProgress = (int)dPercent;
278                     TimeSpan tsUsedTime = DateTime.Now - StartTime;
279                     if ((int)tsUsedTime.TotalSeconds != iOldTotalSeconds ? true : iNewProgress != iOldPr
280                     {
281                         iOldTotalSeconds = (int)tsUsedTime.TotalSeconds;
282                         iOldProgress = iNewProgress;

```

CodeMeter Protection Suite

	AxProtector Windows	AxProtector macOS	AxProtector Linux	AxProtector .NET	AxProtector Python	AxProtector JavaScript	AxProtector Java	AxProtector Android
Automatic Protection	✓	✓	✓	✓	✓	✓	✓	✓
Modular Licensing	✓	✓	✓	✓	✓	✓	✓	✓
IP Protection	✓	✓	✓	✓	✓	✓	✓	✓
CodeMoving	✓	✓	✓	✓	✓	✓	✓	✓
File Encryption	planned	planned	planned	✓	✓	✓	planned	planned
Compile Time Obfuscation	✓	✓	✓					

- **AxProtector .NET 10.30**
 - .NET Framework < 4.7.2
- AxProtector .NET (incl. AxProtector GUI)
 - .NET Framework 4.7.2 or higher
- AxProtector .NET Standard
 - .NET Standard 2.0 or higher
- **AxProtector .NET NC (Native Core)**
 - .NET Framework 4.7.2 or higher, .NET Standard 2.0 or higher, .NET 5 or higher

Windows

- Windows x86
- Windows x86_64

macOS

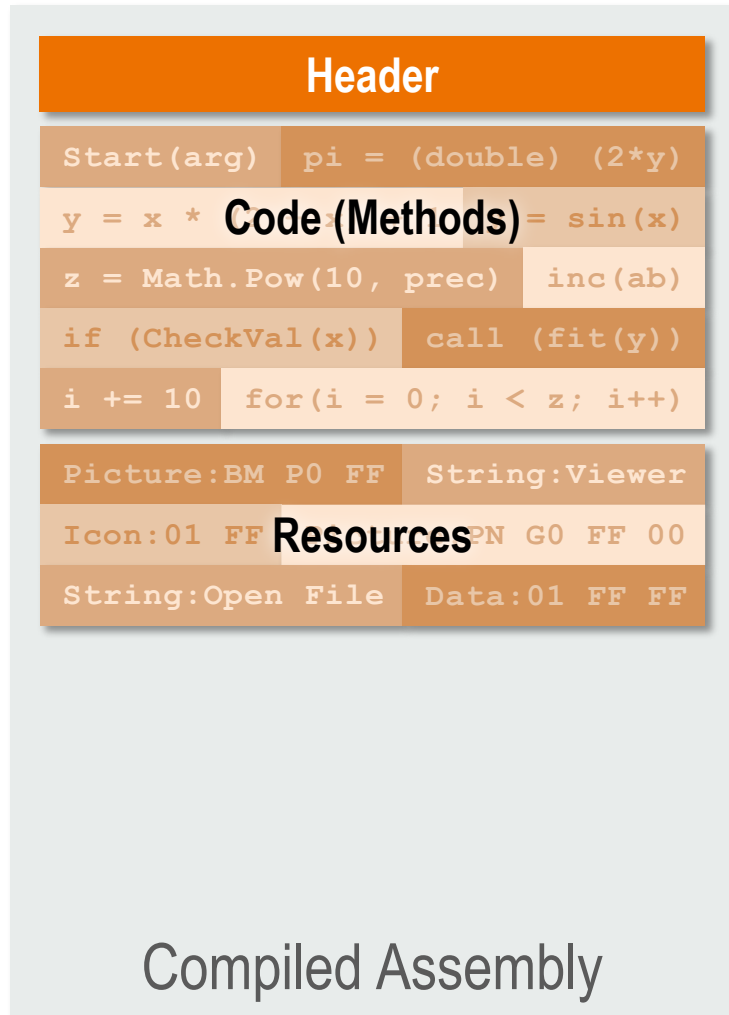
- macOS x86_64
- macOS ARM

Linux

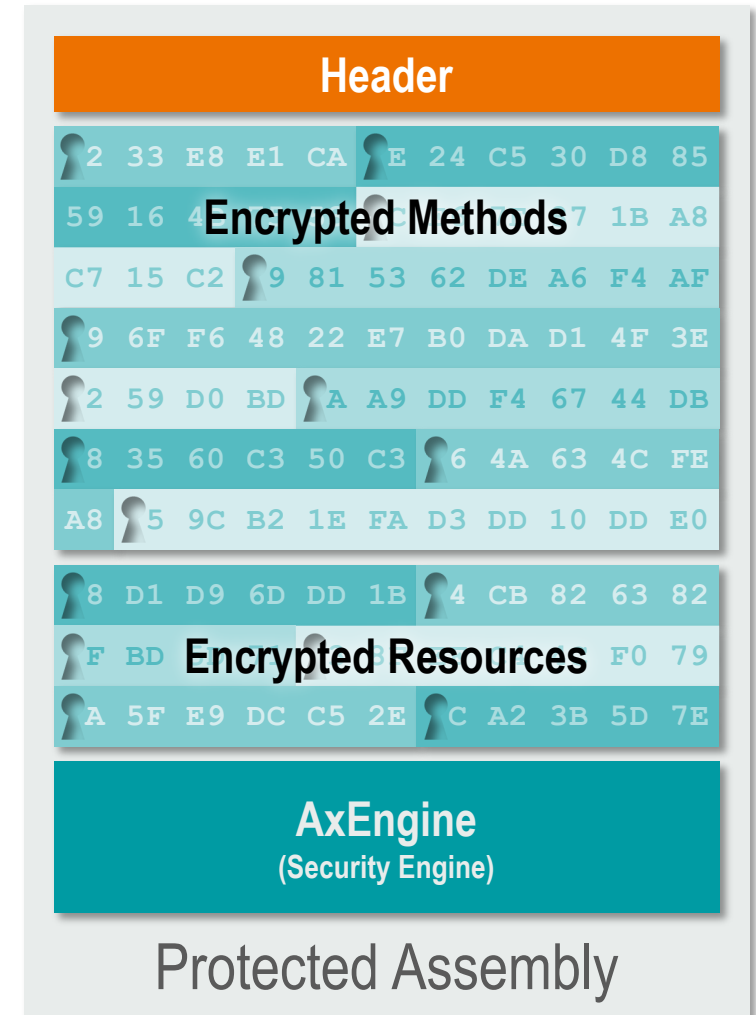
- Linux x86_64 glibc
- Linux ARMhf glibc
- Linux AARCH64 glibc
- Linux x86_64 musl
- Linux ARMhf musl
- Linux AARCH64 musl



Protecting a Single Assembly

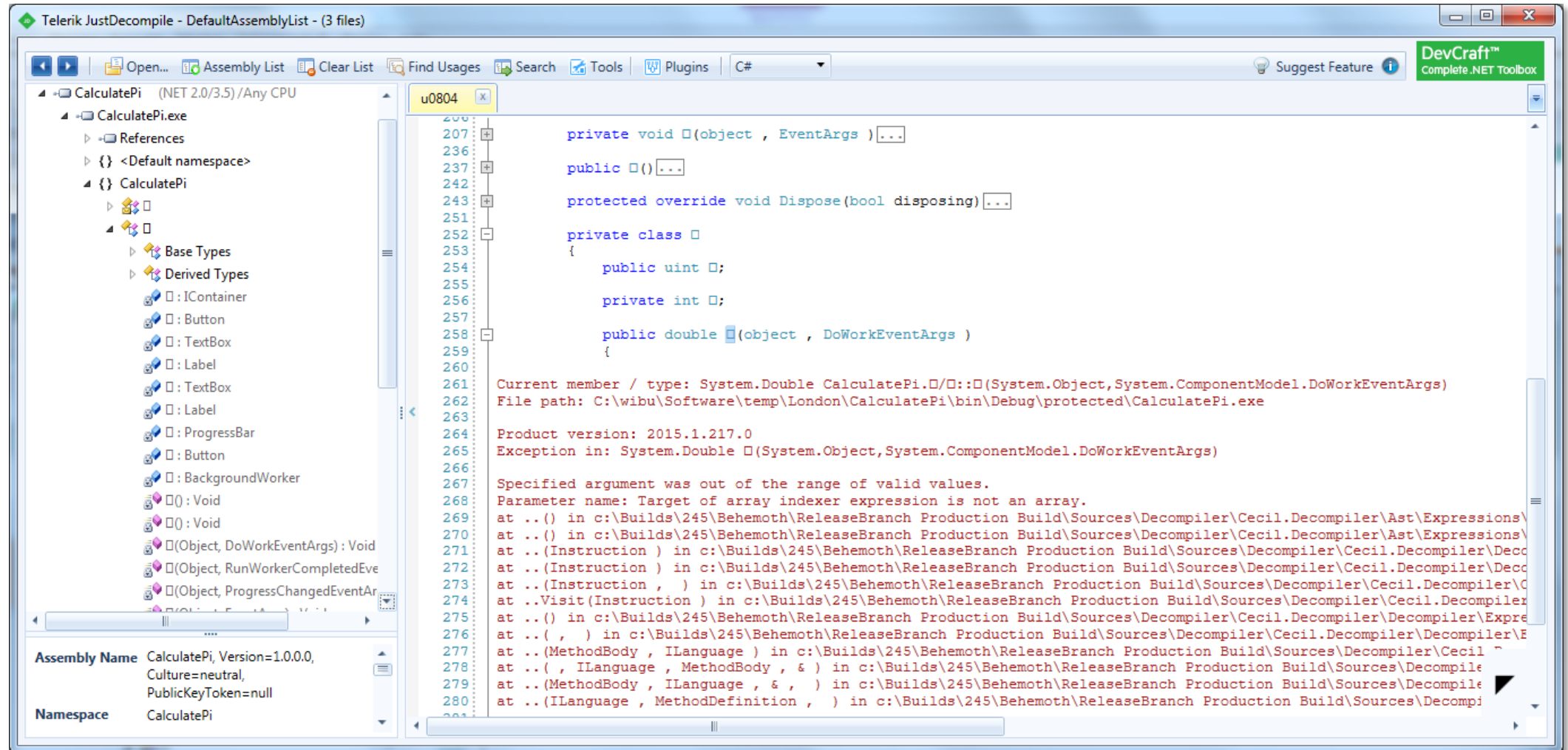


- Firm Code
- Product Code
- ...



- Encryption at vendor's side
 - Extracting the classes, the methods, and the executable code
 - Producing the new assembly (same name)
 - The new assembly looks the same, methods are encrypted individually and include some additional information
 - Resources are encrypted
 - Inserting automatic security checks
- Reflection, remoting, etc. are still possible
- Name obfuscation as additional option available

- Decryption during runtime
 - When calling the method, its code will be automatically decrypted
 - Only if the suitable license is available
 - **On-Demand Decryption** of the executable code
 - Decrypted IL code is removed automatically from memory after JIT compilation
- **CpsEvents** adopts flexible error handling



Modular Protection of a Single Assembly

Concept of AxProtector .NET (Modular Protection)

Header

Start(arg)	pi = (double) (2*y)
y = x * Code (Methods)	= sin(x)
z = Math.Pow(10, prec)	inc(ab)
if (CheckVal(x))	call (fit(y))
i += 10	for(i = 0; i < z; i++)

Resources

Picture:BM P0 FF	String:Viewer
Icon:01 FF	PN G0 FF 00
String:Open File	Data:01 FF FF

Compiled Assembly



- ...
- License Lists assigned to different methods

Header

2 33 E8 E1 CA	E 24 C5 30 D8 85
59 16 4B 75 53	C E6 7E 87 1B A8
C7 15 C2	9 81 53 62 DE A6 F4 AF
9 6F F6 48 22 E7 B0 DA D1 4F 3E	
inc(ab)	A A9 DD F4 67 44 DB 8
35 60 C3 50 C3	i += 10 5 9C B2
1E FA D3 DD 10 DD E0	00 00 00 00

8 D1 D9 6D DD 1B	4 CB 82 63 82
F BD 5D 71	3 8F EF C4 4C F0 79
A 5F E9 DC C5 2E	C A2 3B 5D 7E

AxEngine
(Security Engine)

Protected Assembly

Demo

- Encryption with different license list for highest security
 - Different effective key for code encryption
 - Markup with an attribute
 - Configuration in YAML-file
- Application behavior by WUPI calls
 - Wibu Universal Protection Interface
 - High-level API for easy integration
 - No AxEngine/CpsEvents messages are shown
- You need both – encryption and API calls

```

/// <summary> ...
private void mViewFont_Click(object sender, EventArgs e)
{
    if (WupiEngine.Wupi.CheckLicense(1))
    {
        ChangeFont();
    }
    else
    {
        MessageBox.Show("No License found!", "License Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

```

/// Changes the font
/// </summary>
[WupiEngine.Licensing(LicenseList=1)]
private void ChangeFont()
{
    FontDialog dlgFont = new FontDialog();
    dlgFont.Font = rtContent.Font;
    dlgFont.ShowColor = true;
}

```

```

[0] InitializeComponent() (721 Bytes)
[0] frmMain (2706 Bytes)
[0] CalculatePi(System.UInt32) (94 Bytes)
[0] calculatePiToolStripMenuItem_Click_1(System.Object,System.EventArgs) (132 Bytes)
[1] ChangeFont() (98 Bytes)
[0] Dispose(System.Boolean) (30 Bytes)
[-] Exit() (7 Bytes)
[-] frmMain_Load(System.Object,System.EventArgs) (1 Bytes)

```

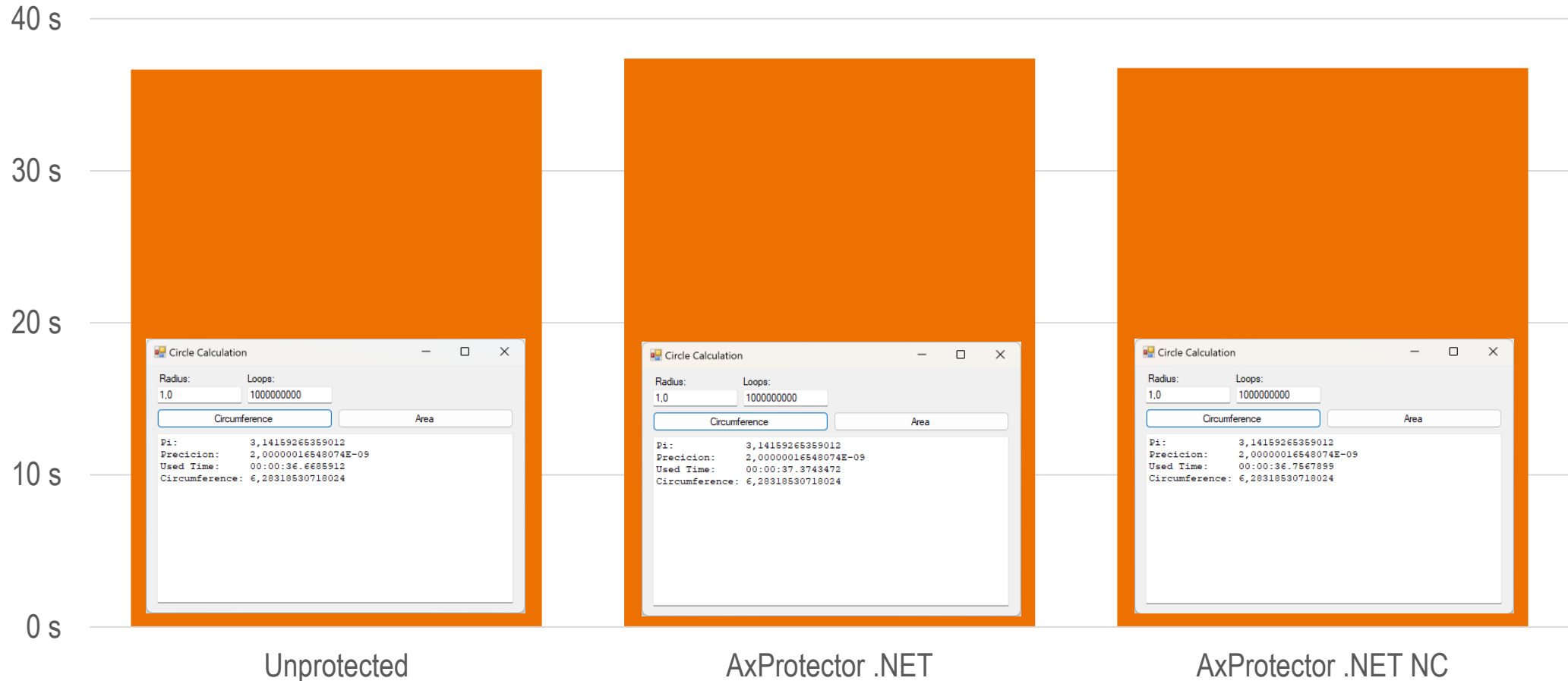
- Alternative approach for error handling (no license found)
 - Exception handling
 - Global handler around assembly entry points to catch global licensing exceptions
 - AxEngine/CpsEvents message is shown

```
/// <summary> ...  
private void mViewFont_Click(object sender, EventArgs e)  
{  
    try  
    {  
        ChangeFont();  
    }  
    catch (WupiEngine.WupiException ex)  
    {  
        MessageBox.Show("No License found!\r\n" + ex.Message, "License Error", MessageBoxButtons.OK, MessageBoxIcon.Error);  
    }  
}
```

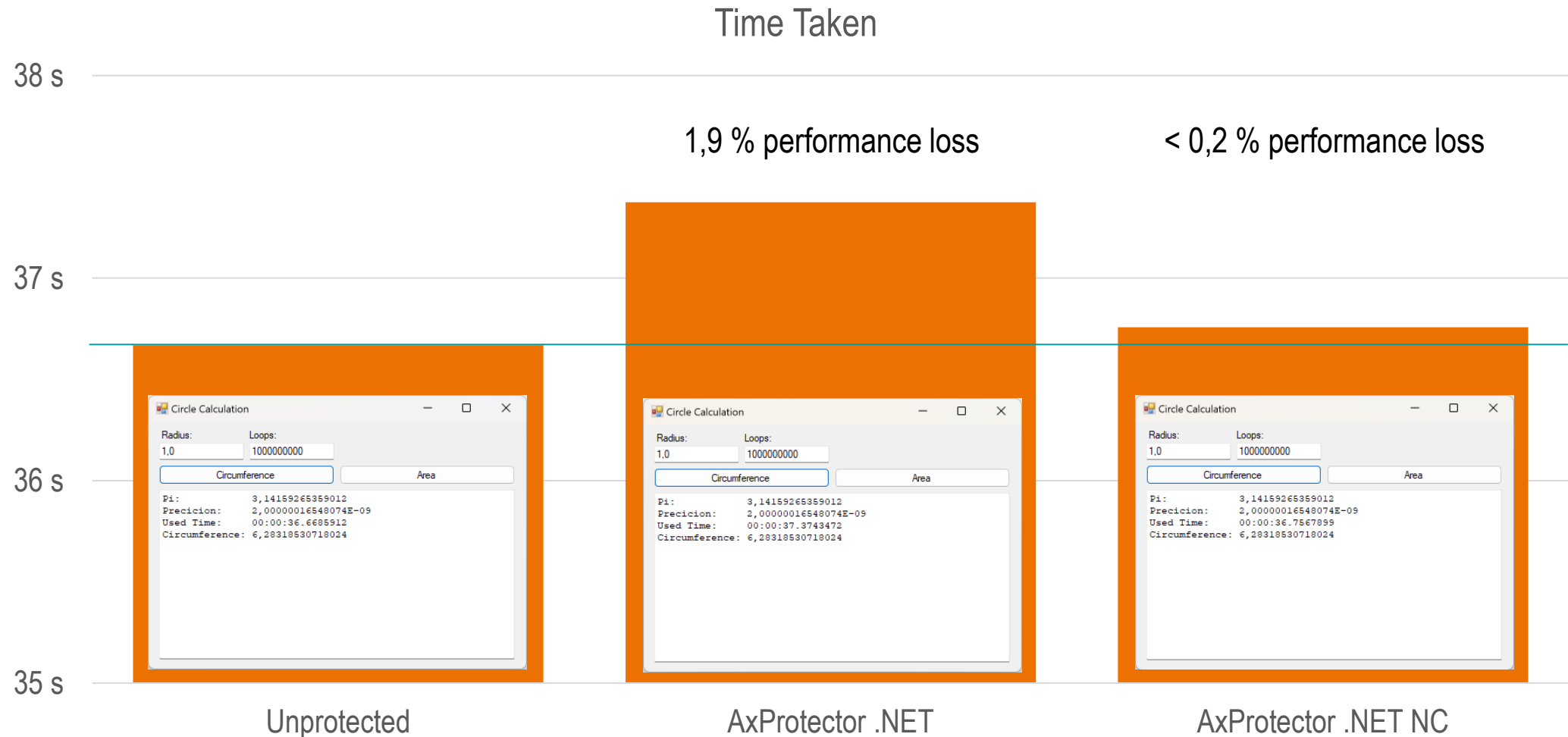
Fine Tuning the Performance

Comparing a Method Executed 1,000,000,000 Times

Time Taken



Comparing a Method Executed 1,000,000,000 Times



Modular Protection of Multiple Assemblies

- Best practice approach:
 - One assembly – one license – one **Product Code** – encryption as single assembly
 - Mixing protected and unprotected assemblies is possible
- Error handling in calling executable or in called library
 - Each entry point is unencrypted and checks its own license first
 - Library raises exception and executable catches exception
 - Executable knows and checks license before loading library
- Same technical solution as single assembly

- Encrypting plugins with individual Firm Code / Product Code
 - Mixing different Firm Codes is possible
 - Mixing with other licensing vendors is possible
- Error handling in library (plugin)
 - Error handling at entry points
 - Unencrypted entry methods

Multiplatform Protection

Demo

Thank you very much for your attention



Europe: +49-721-931720
USA: +1-425-7756900
China: +86-21-55661790
Japan: +81-45-5659710

<https://www.wibu.com>
info@wibu.com