WIBU SYSTEMS

SECURITY
LICENSING
PERFECTION IN PROTECTION

Protection and Licensing .NET Applications

CodeMeter Protection Suite

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

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

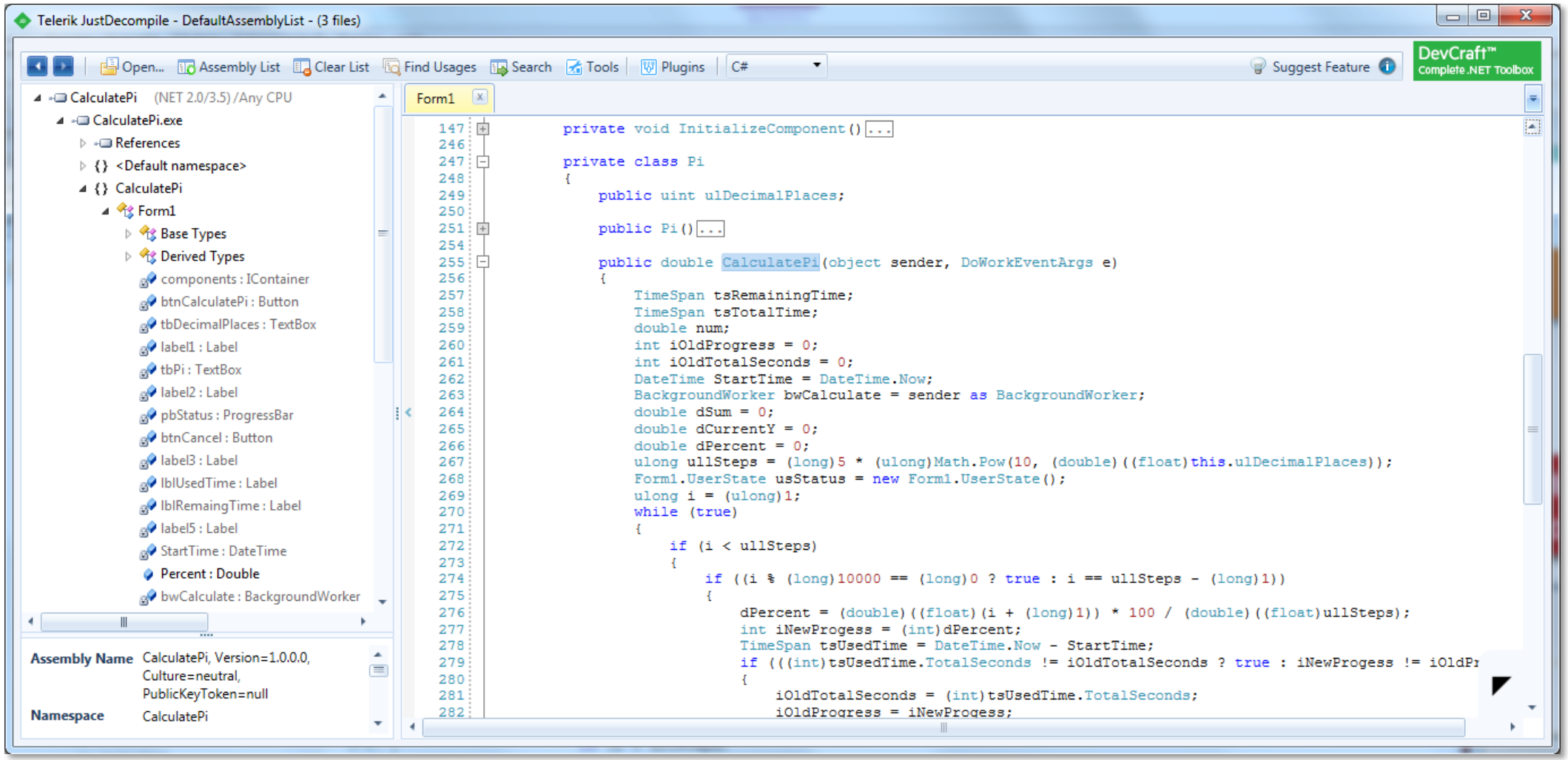
Where to find the accompanying audio

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

Disassembling a .NET Application (Just Decompile)



CodeMeter Protection Suite

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	✓	✓	✓					

Variants of AxProtector .NET

- 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

AxProtector .NET NC – Targets

Windows

- Windows x86
- Windows x86_64

macOS

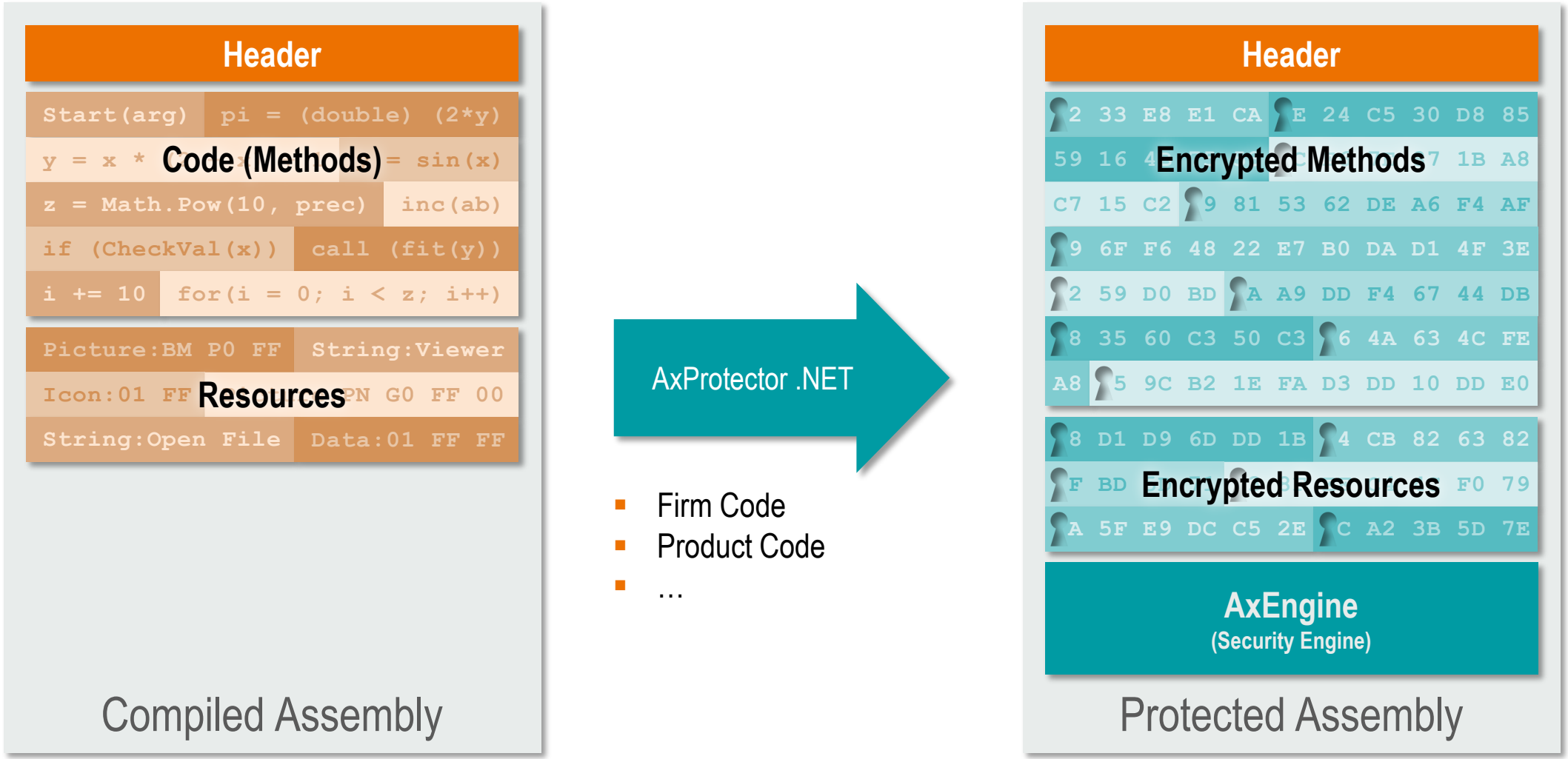
- macOS x86_64
- macOS ARM

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

Operating Principle of AxProtector .NET



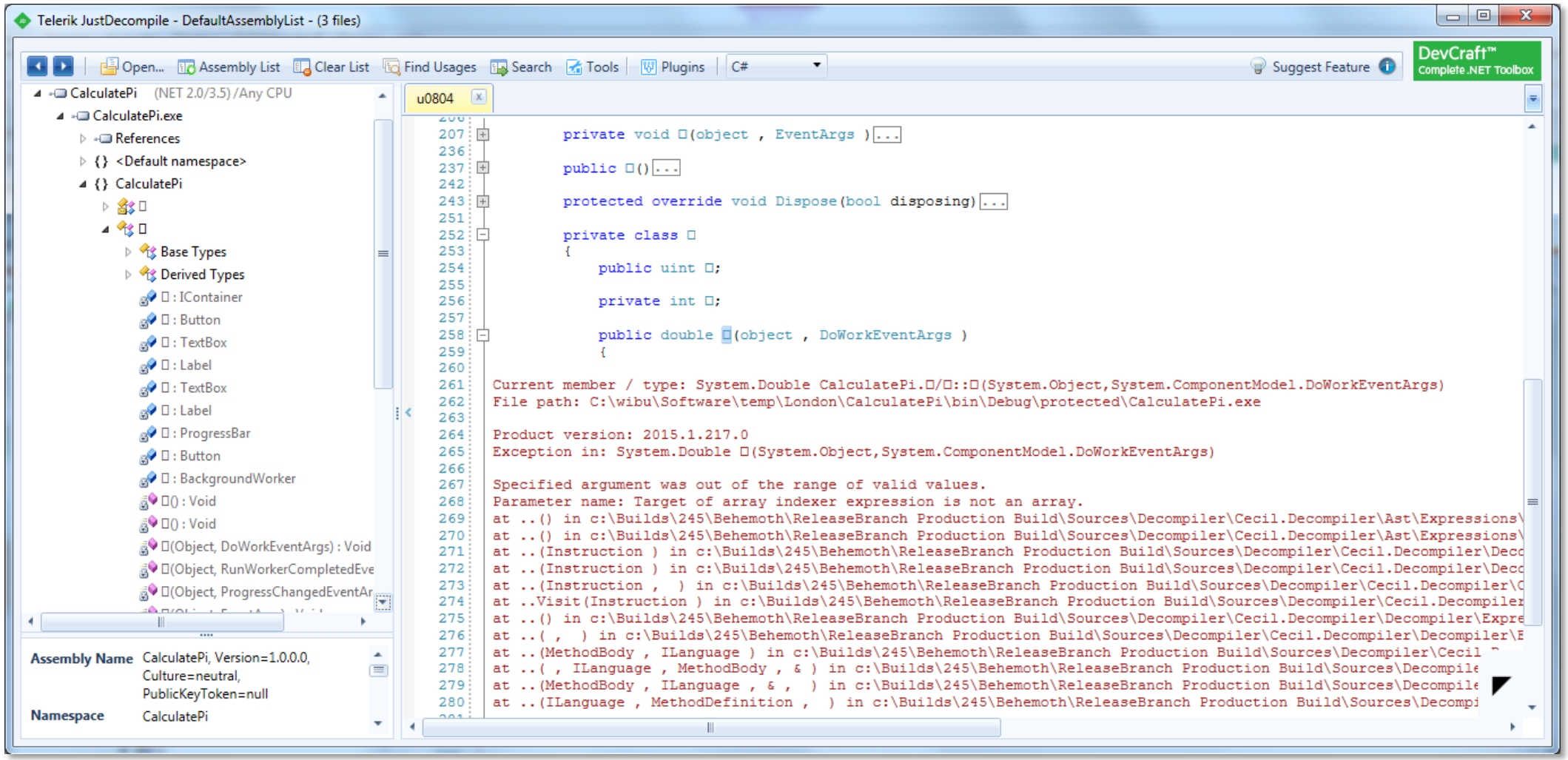
AxProtector .NET – ISV's Side

- 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

AxProtector .NET – User's Side

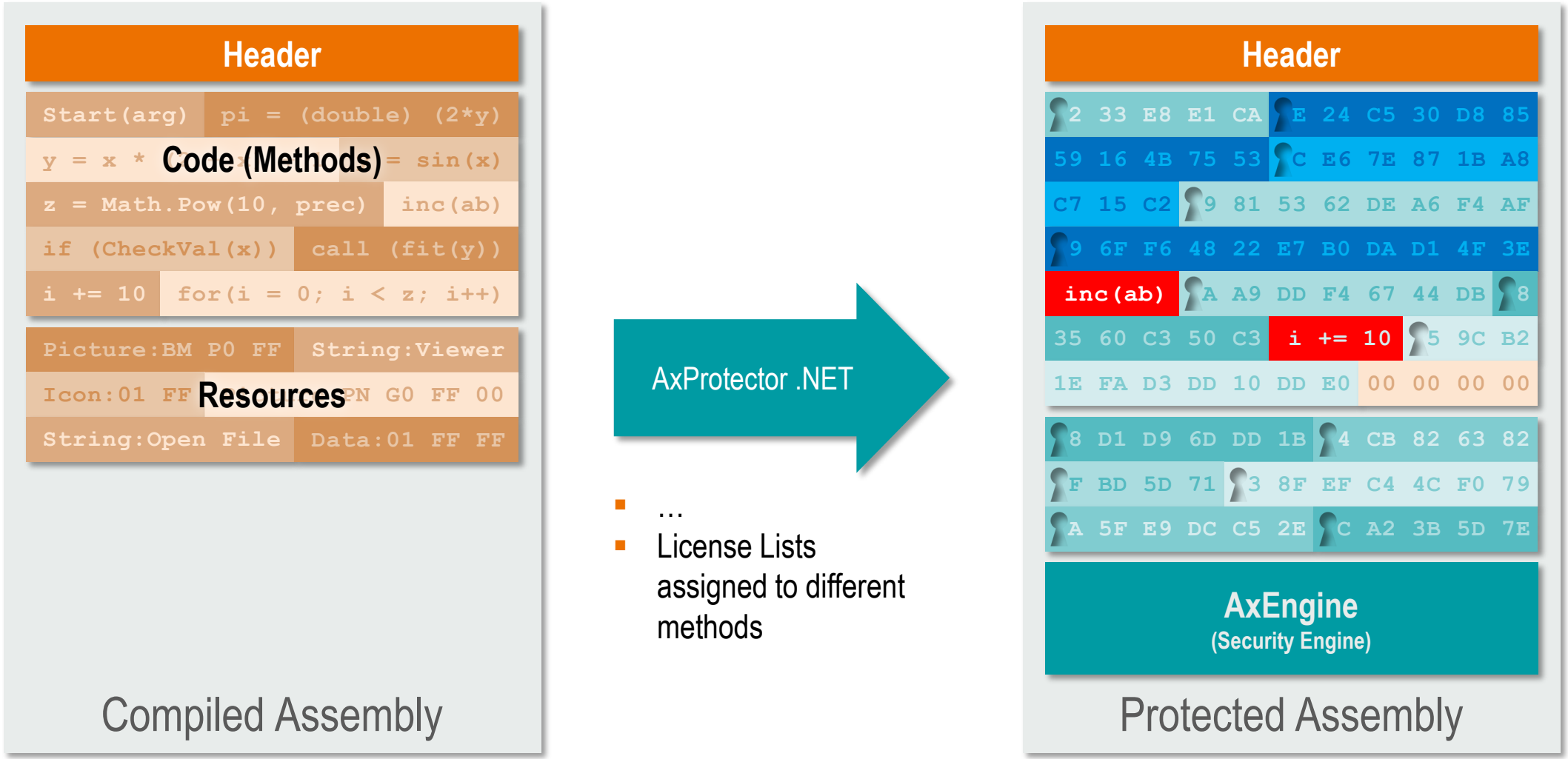
- Decryption during runtime
  - When calling the method, its code will be automatically decrypted
  - Only if the suitable license is available
  - On-Demand Decryption of the executable code
  - Decrypted IL code is removed automatically from memory after JIT compilation
- CpsEvents adopts flexible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

Just Decompile with a Protected .NET Assembly



Modular Protection of a Single Assembly

Concept of AxProtector .NET (Modular Protection)



Modular Protection of a Single Assembly

Demo

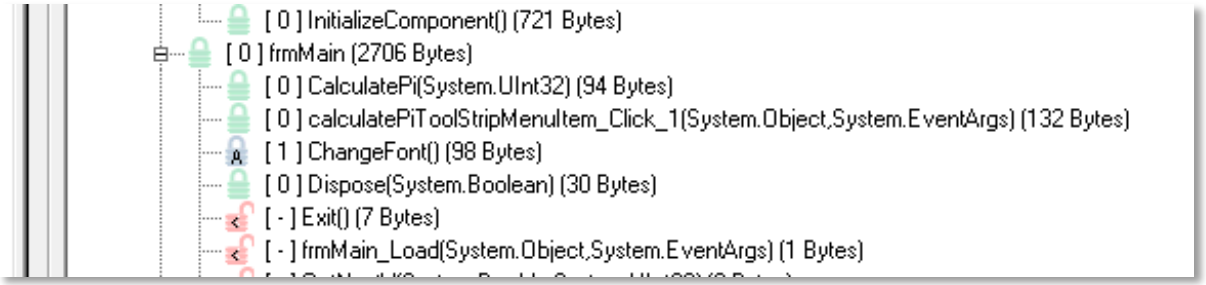
Modular Protection of a Single Assembly

- 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

Modular Protection of a Single Assembly


```
/// <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;
```



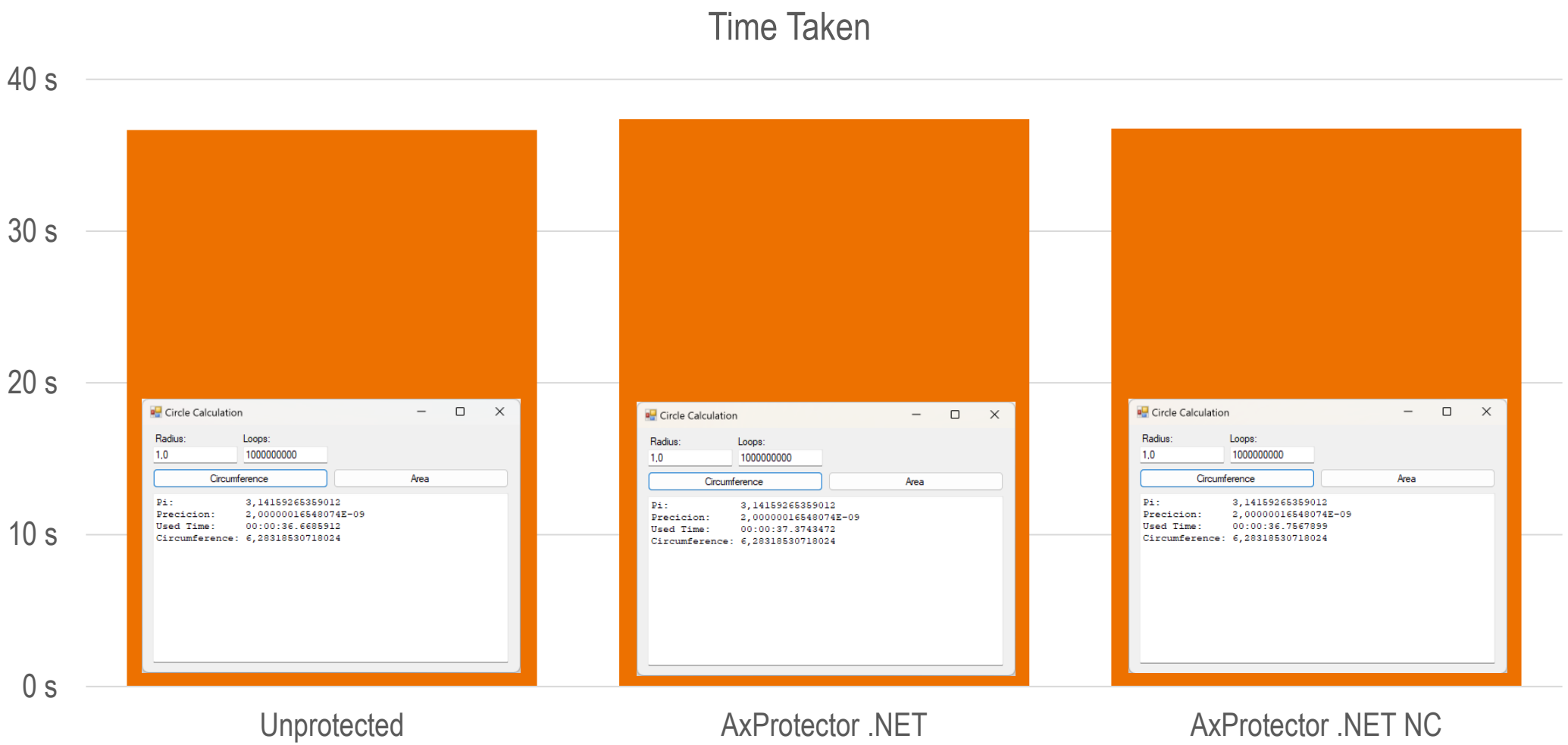
Modular Protection of a Single Assembly

- 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

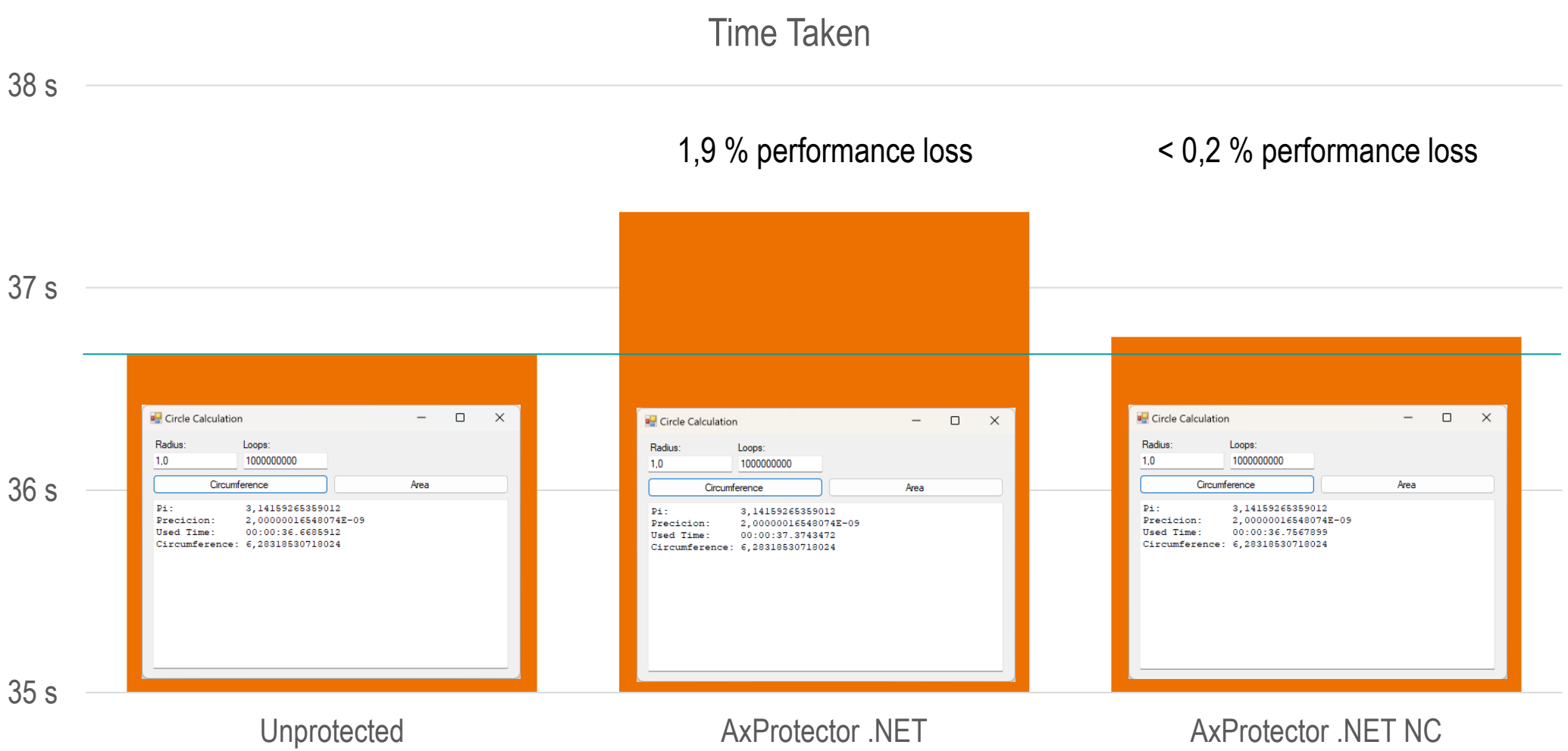
```
/// <summary>  ...
private void mViewFont_Click(object sender, EventArgs e)
{
    try
    {
        ChangeFont();
    }
    catch (WupiEngine.WupiException ex)
    {
        MessageBox.Show("No License found!\r\n" + ex.Message, "License Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}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

Fine Tuning the Performance

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



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



Modular Protection of Multiple Assemblies

Application by one ISV with different modules

- 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

Application by different ISVs (Plugins)

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

Multiplatform Protection

Multiplatform Protection

Demo

WIBU SYSTEMS

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