IT Global Meetup #6

Sandboxing in .NET CLR

Mikhail Shcherbakov
IntelliEgg

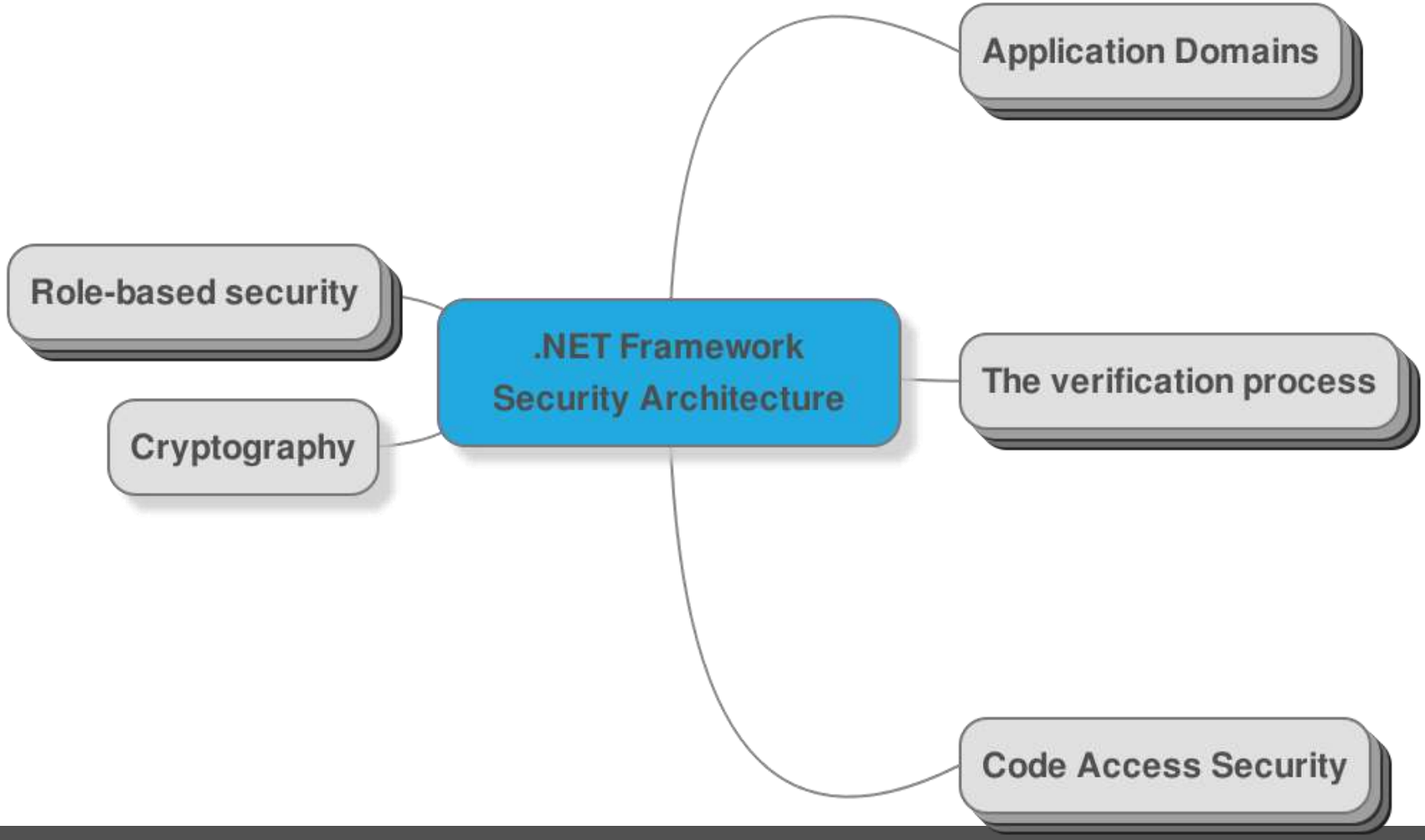
About me

- Creator of IntelliDebugger project
- Coordinator of SPB .NET Community
- Former Product manager and Team lead at Cezurity, Positive Technologies, Acronis, Luxoft, Boeing

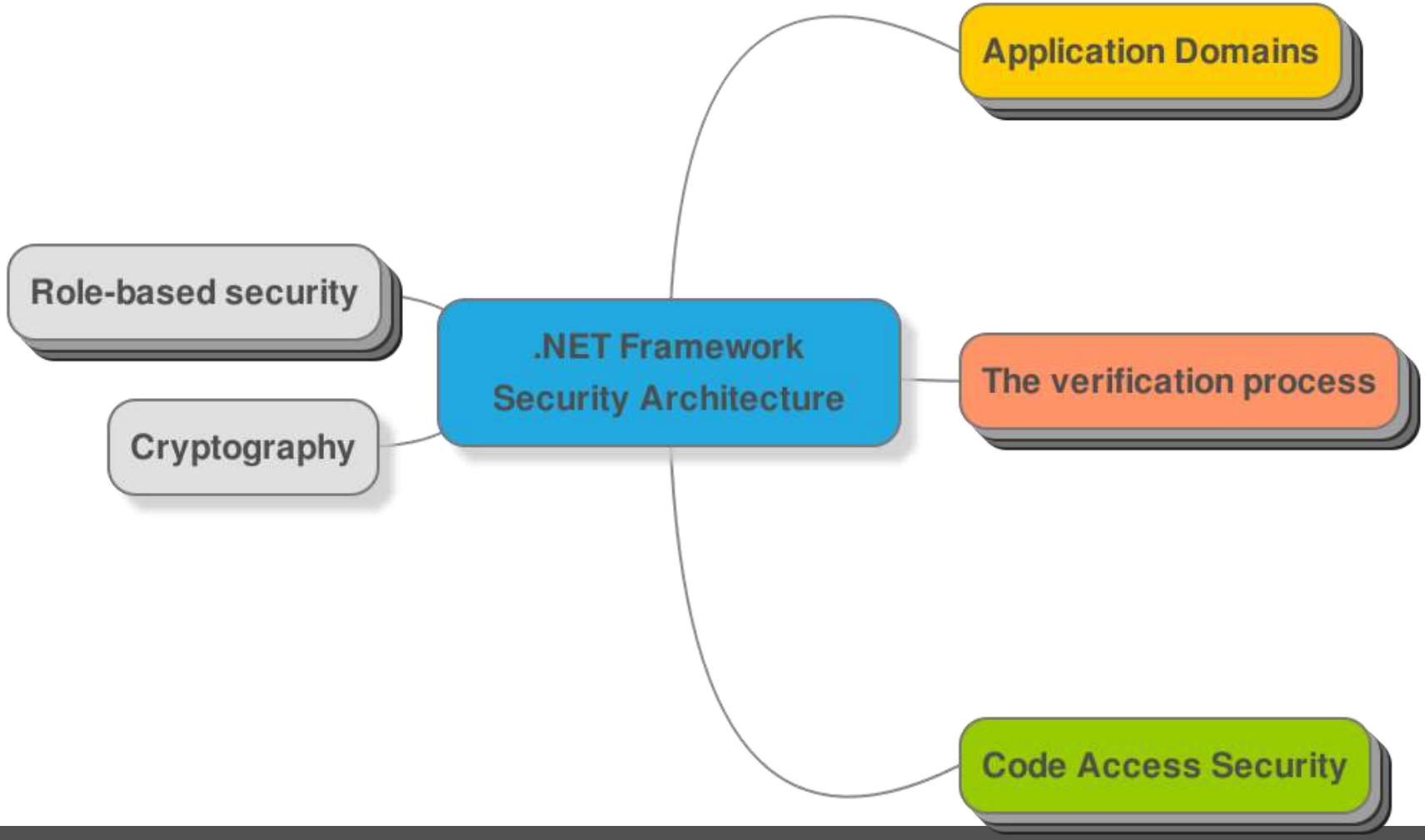
Knowledge in Practice

- Sandboxing is the base of security
  - ASP.NET / IIS
  - SQL CLR
  - ClickOnce
  - Silverlight
  - XBAP
  - Sharepoint
- Development of extensible and security-sensitive applications
- Troubleshooting and knowledge about the internals

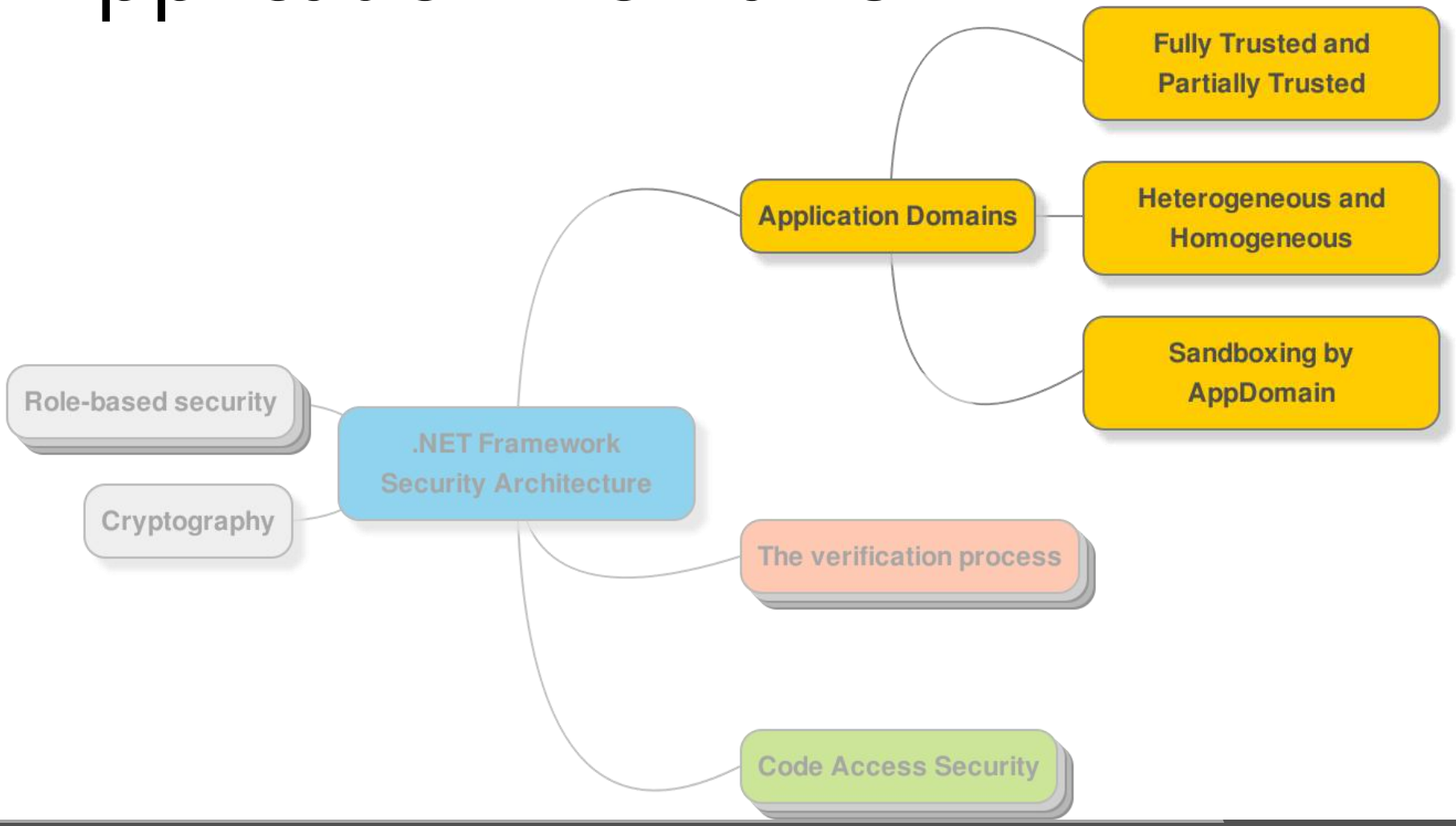
Security Architecture



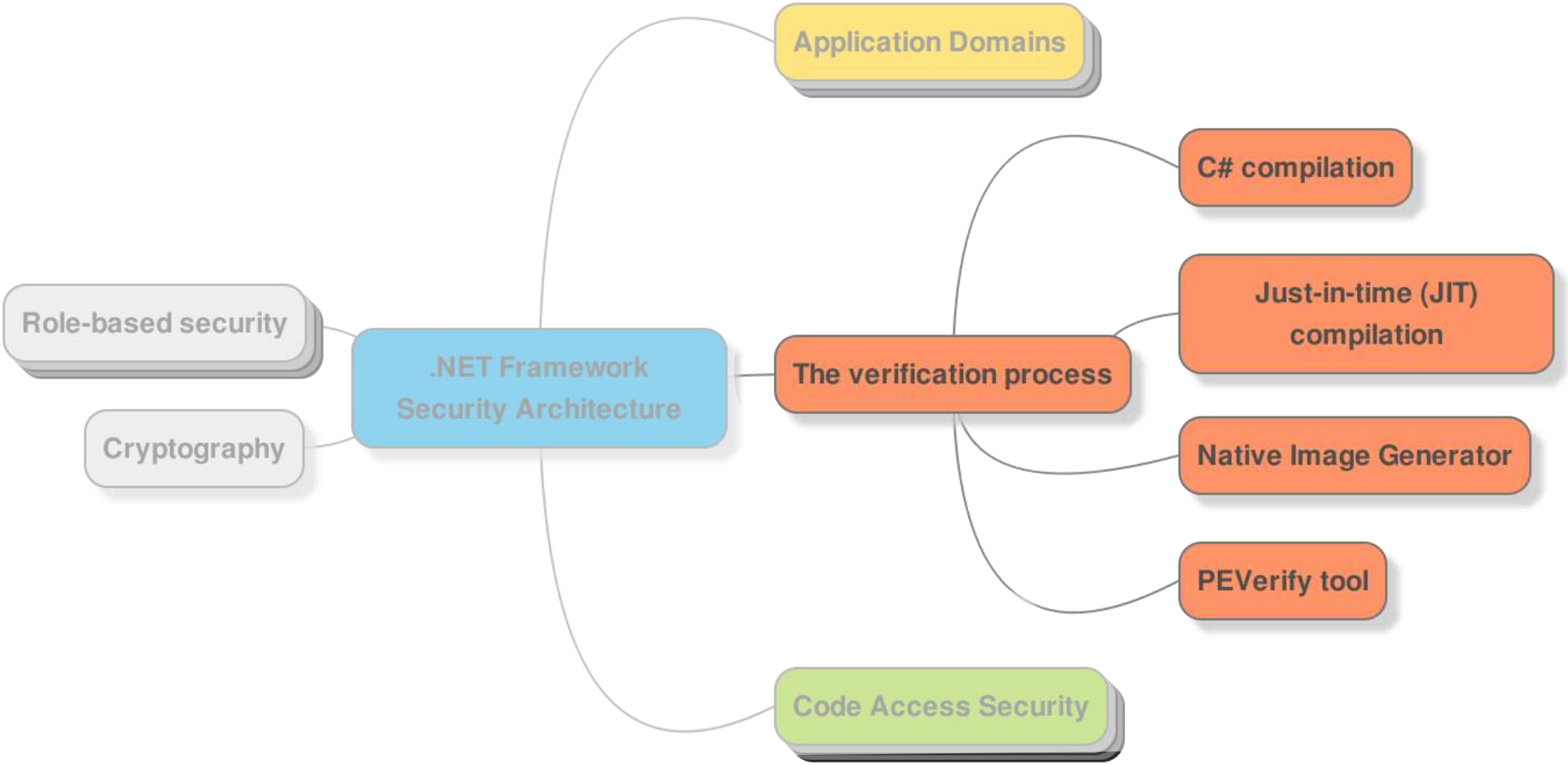
Security Architecture



Application Domains

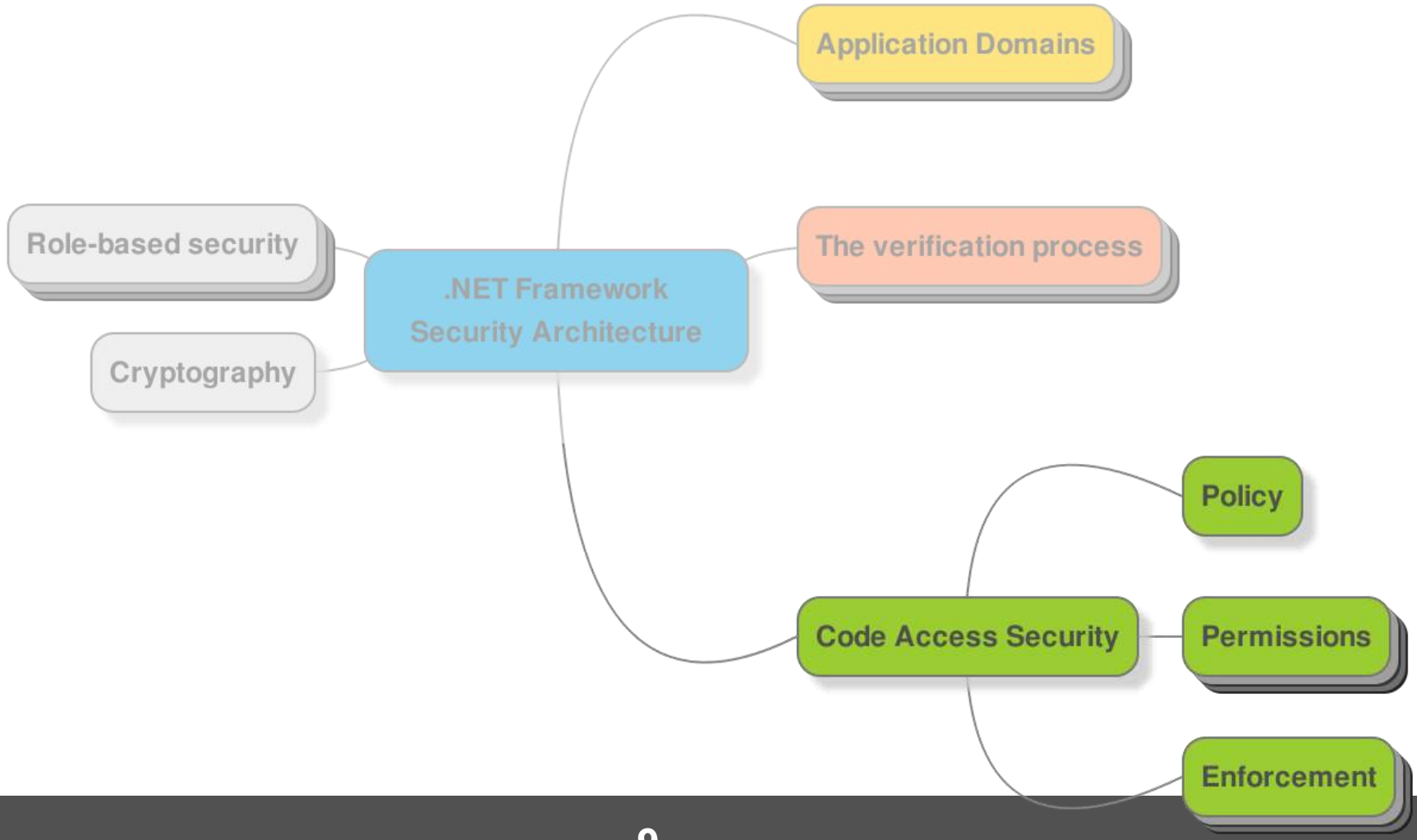


The verification process

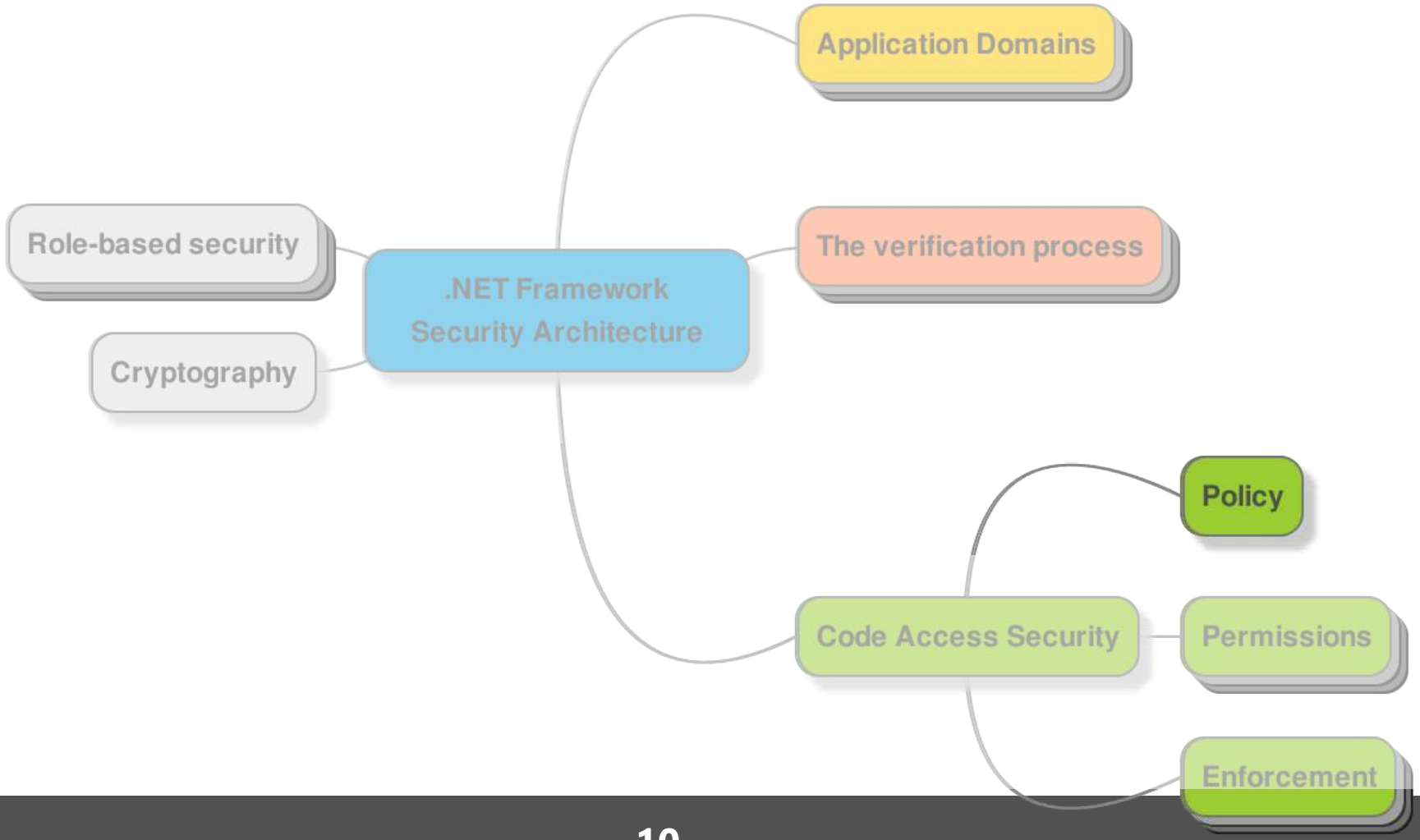


Just-in-time verification

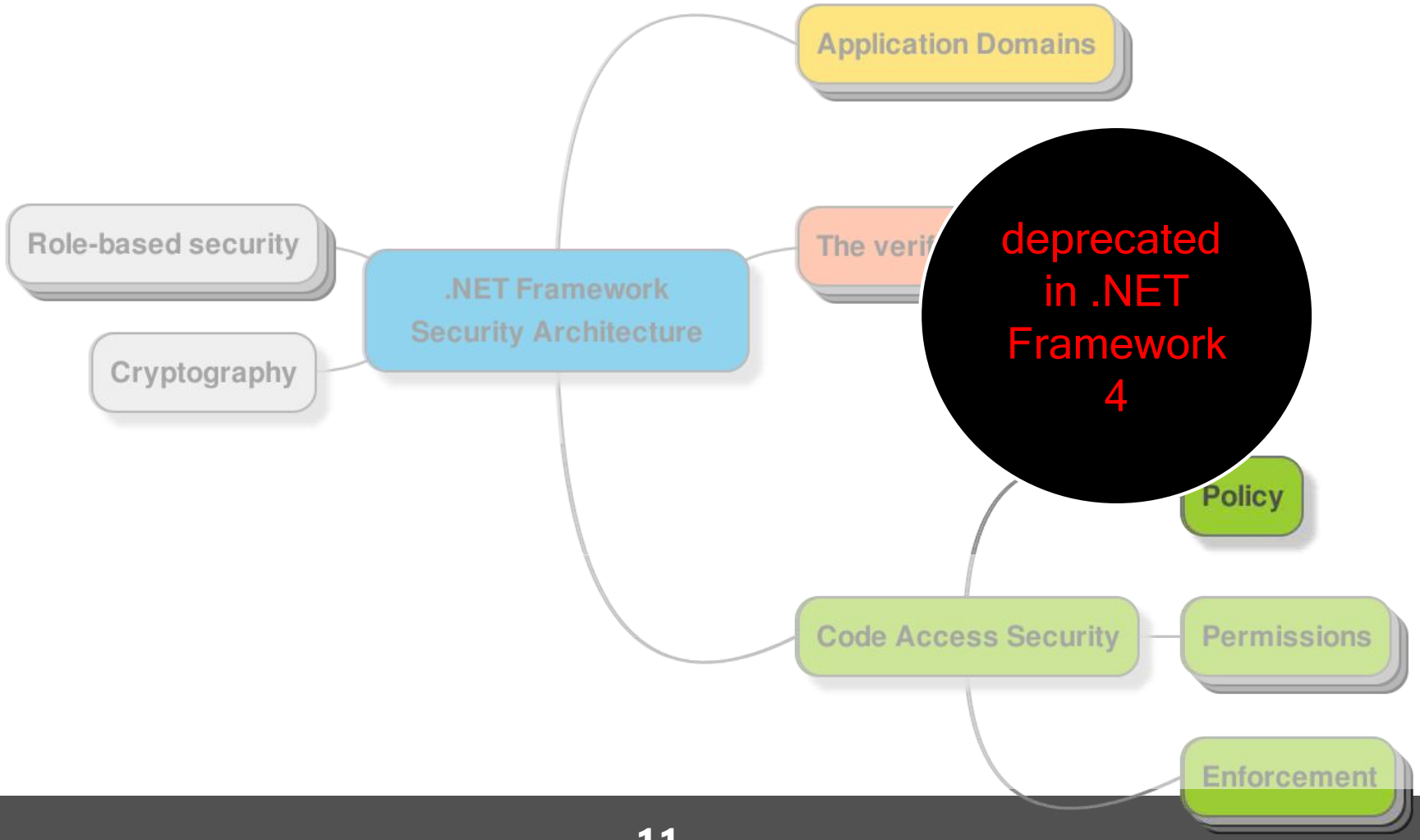
Code Access Security



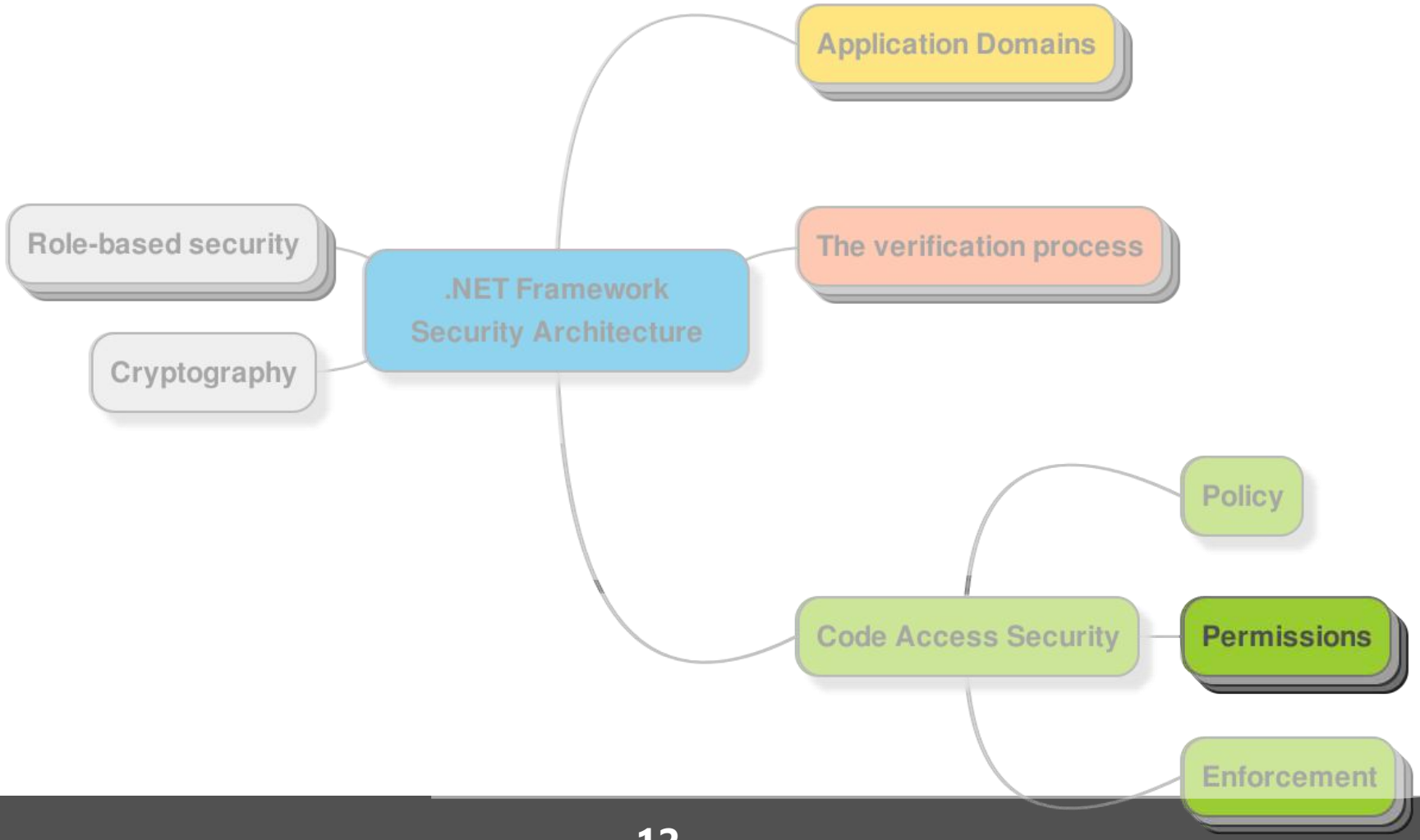
Policy


Application Domains
Role-based security
.NET Framework
Security Architecture
The verification process
Cryptography
Policy
Code Access Security
Permissions
Enforcement

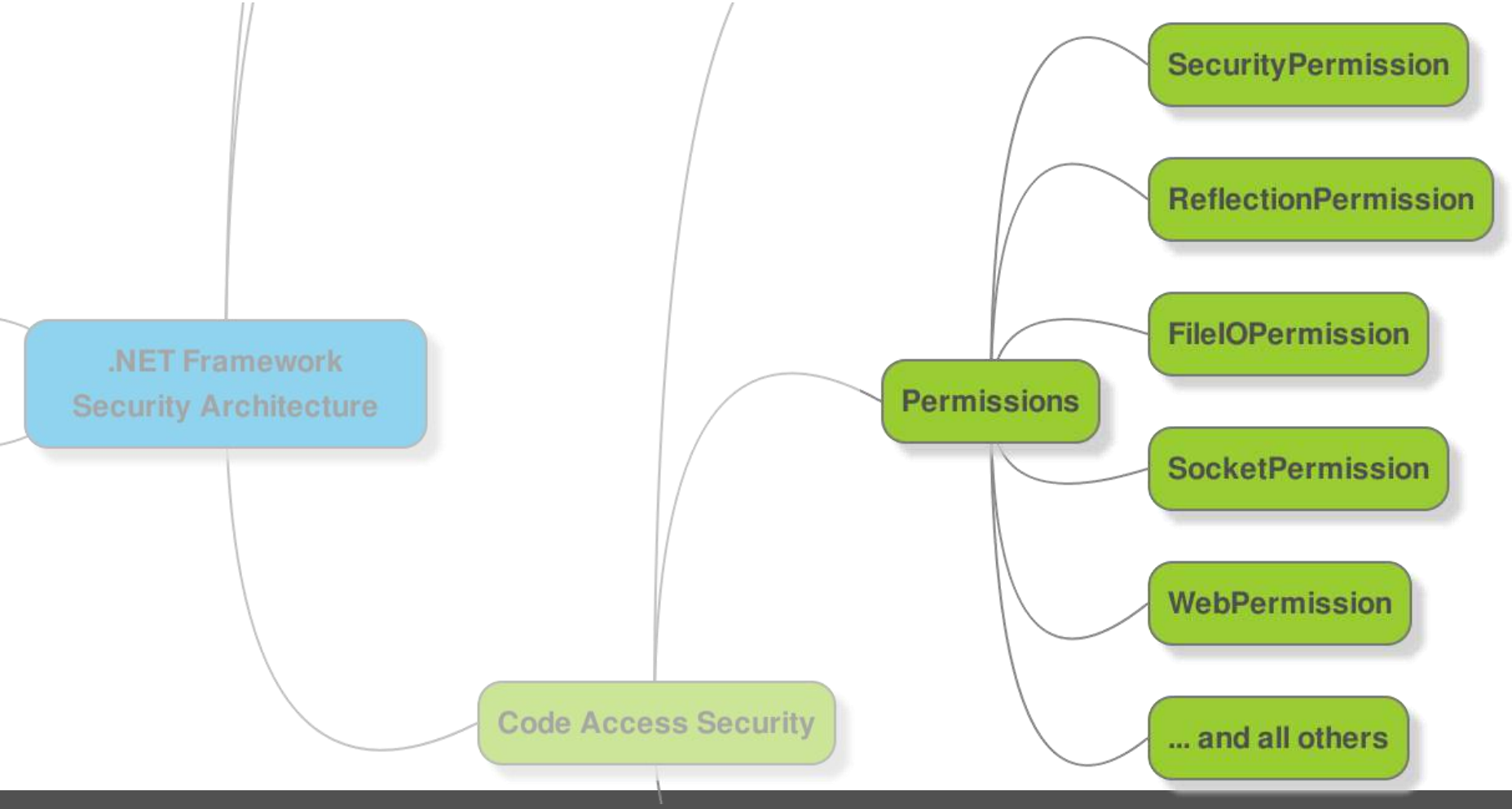
Policy


Application Domains
Role-based security
.NET Framework Security Architecture
The verif
deprecated in .NET Framework 4
Cryptography
Policy
Code Access Security
Permissions
Enforcement

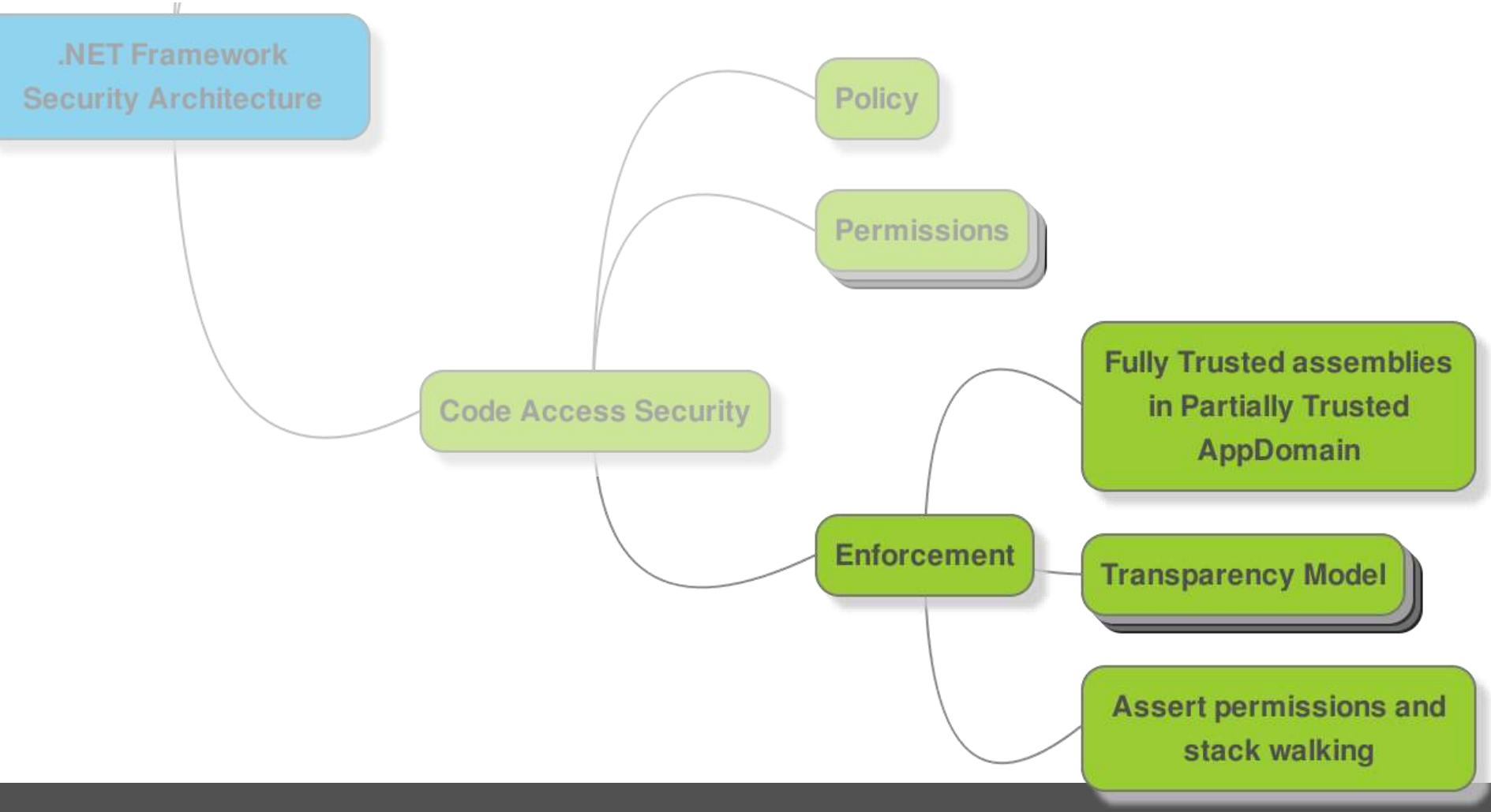
Permissions



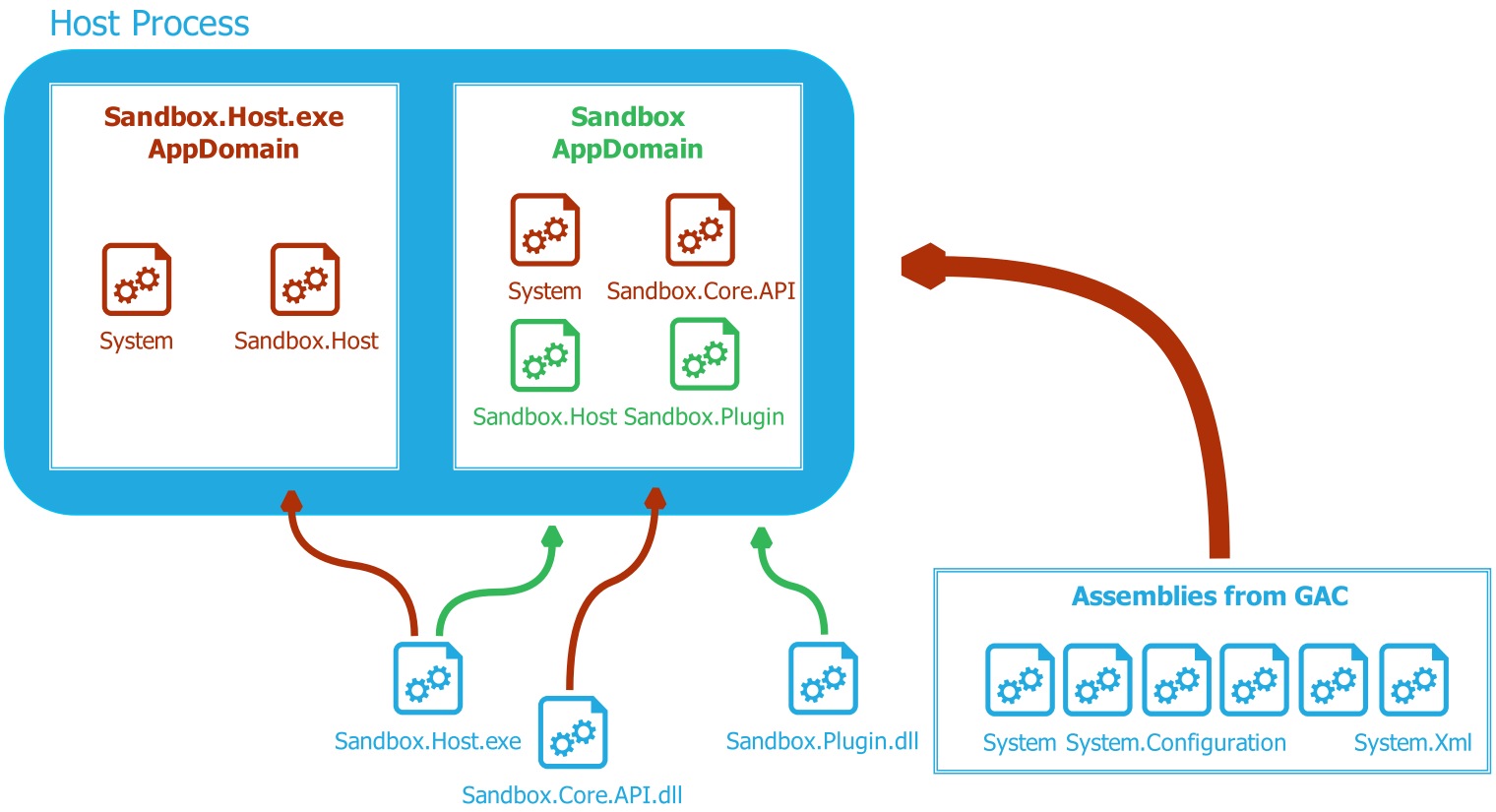
Permissions



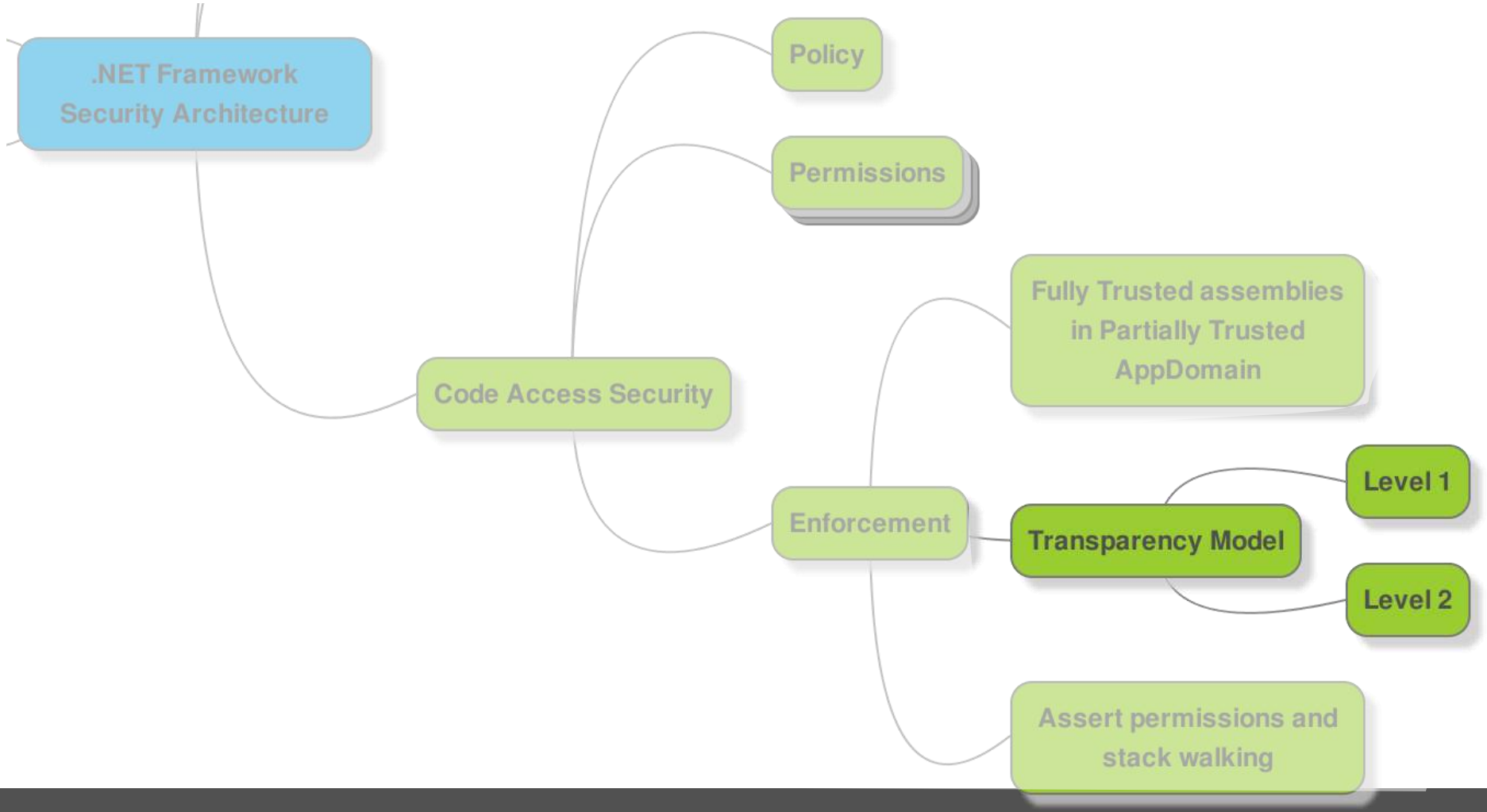
Enforcement



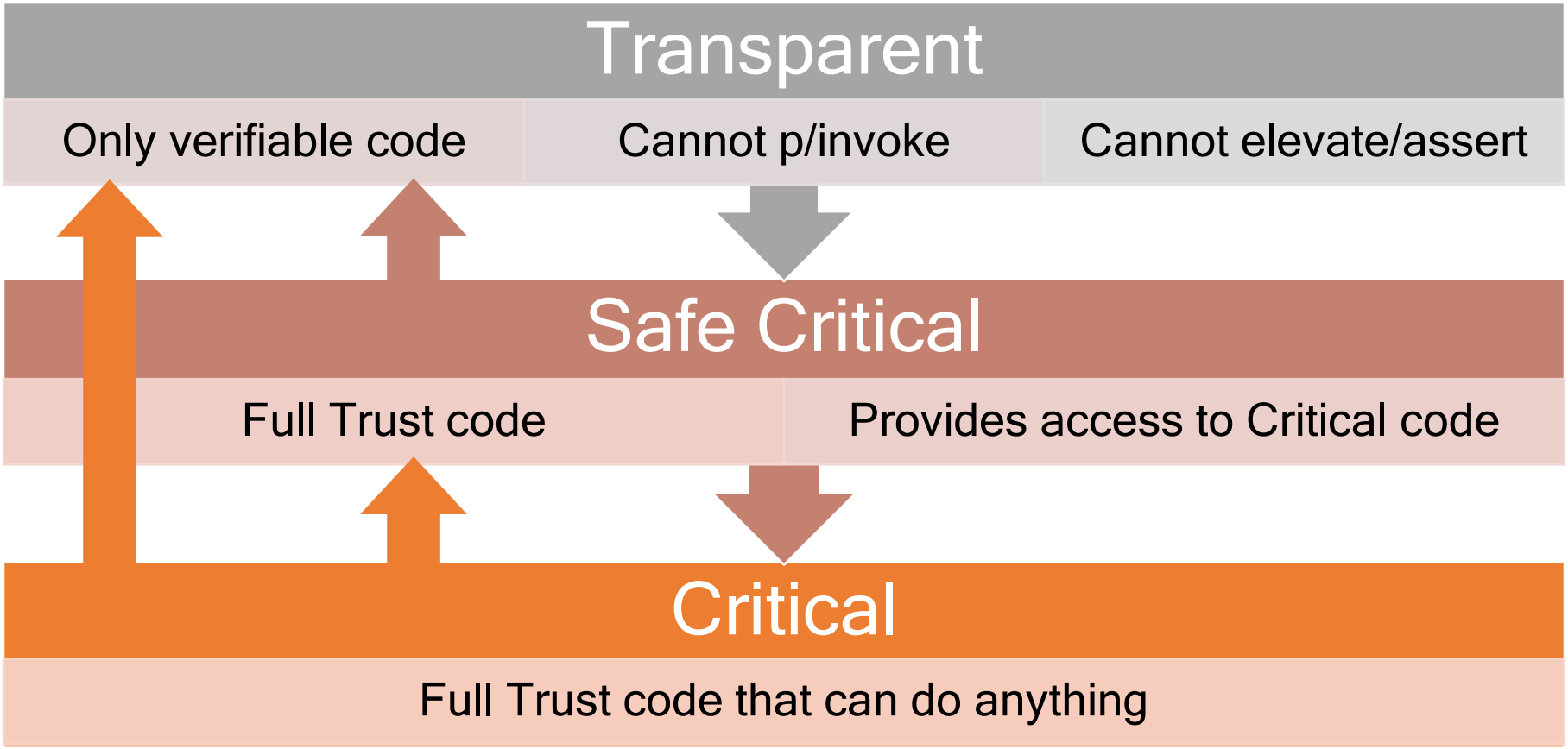
Fully Trusted code in Partially Trusted AppDomain



Transparency Model



Level 2 Security Transparency

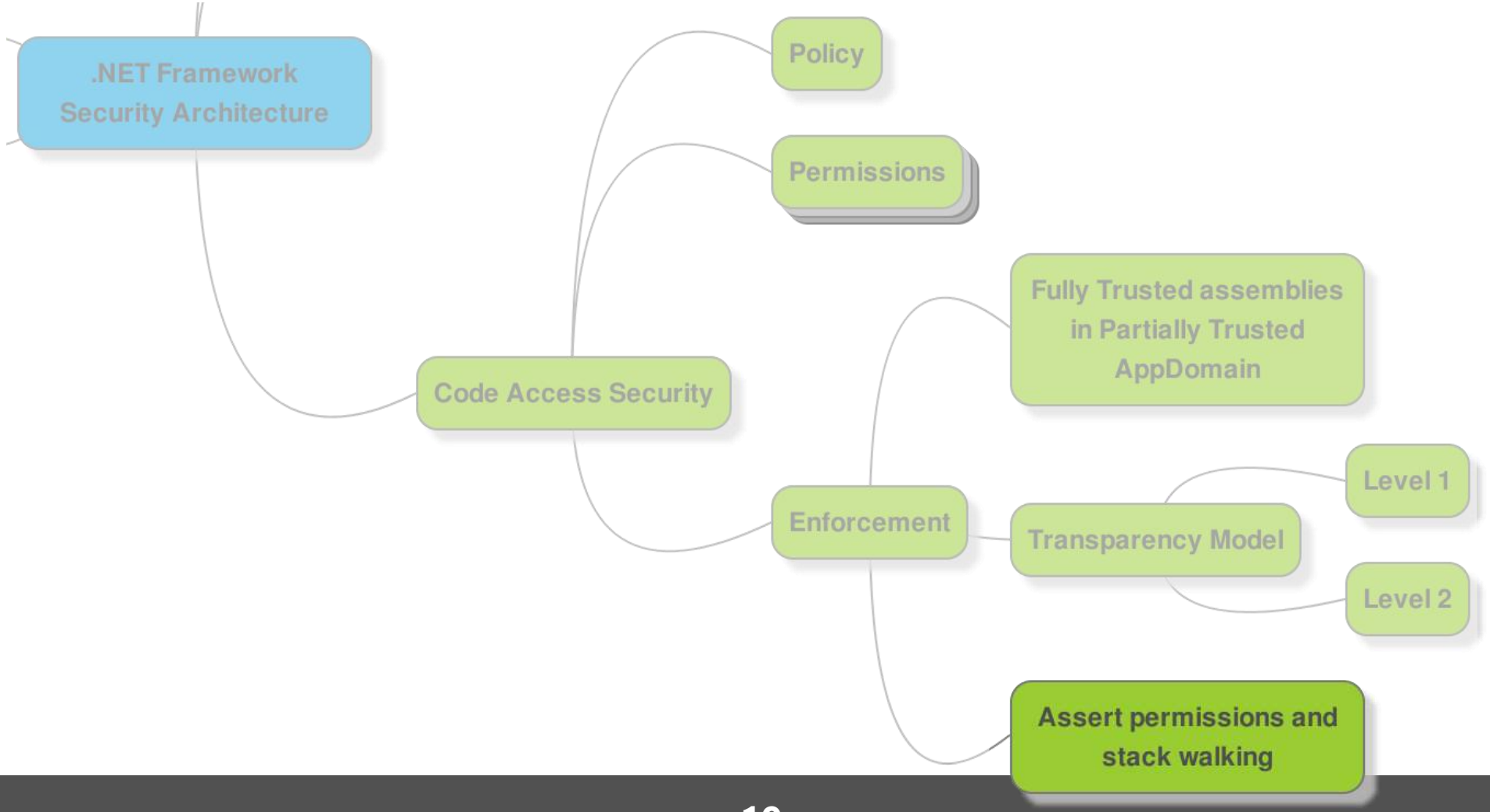


Security Transparency Attributes

	Assembly Level	Type Level	Member Level
SecurityTransparent	✓	✗	✗
SecuritySafeCritical	✗	✓	✓
SecurityCritical	✓	✓	✓
AllowPartiallyTrustedCallers	✓	✗	✗

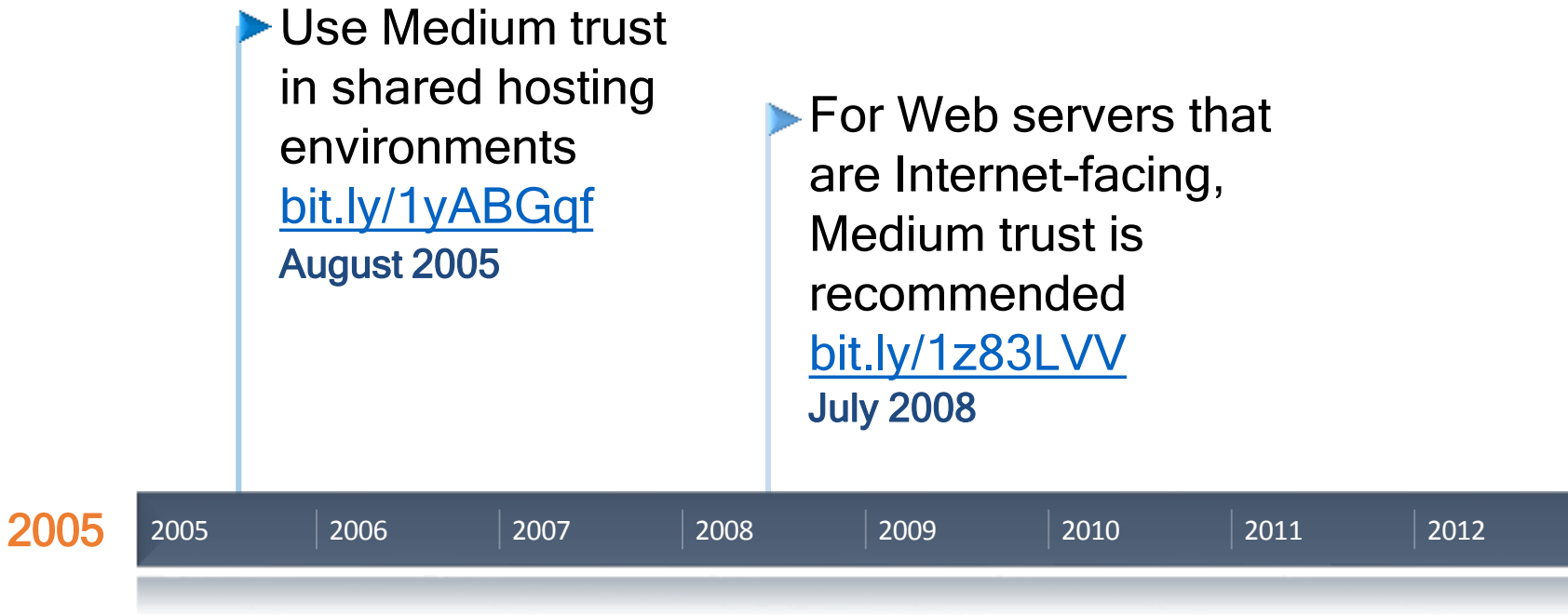
SecAnnotate.exe – .NET Security Annotator Tool
http://bit.ly/1A3vMw3

Stack walking

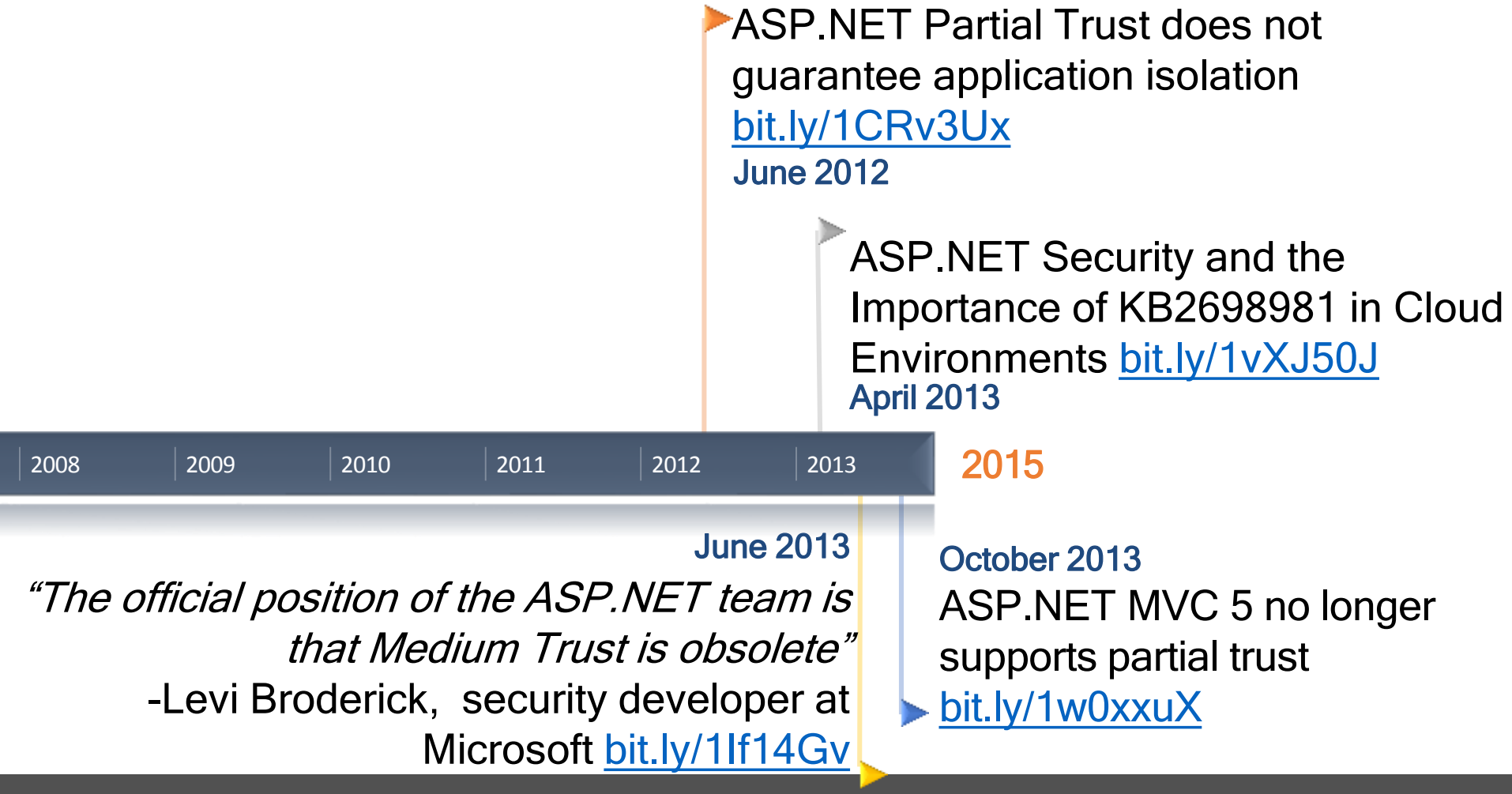


Sandbox implementation

ASP.NET Partial Trust applications



ASP.NET Partial Trust applications

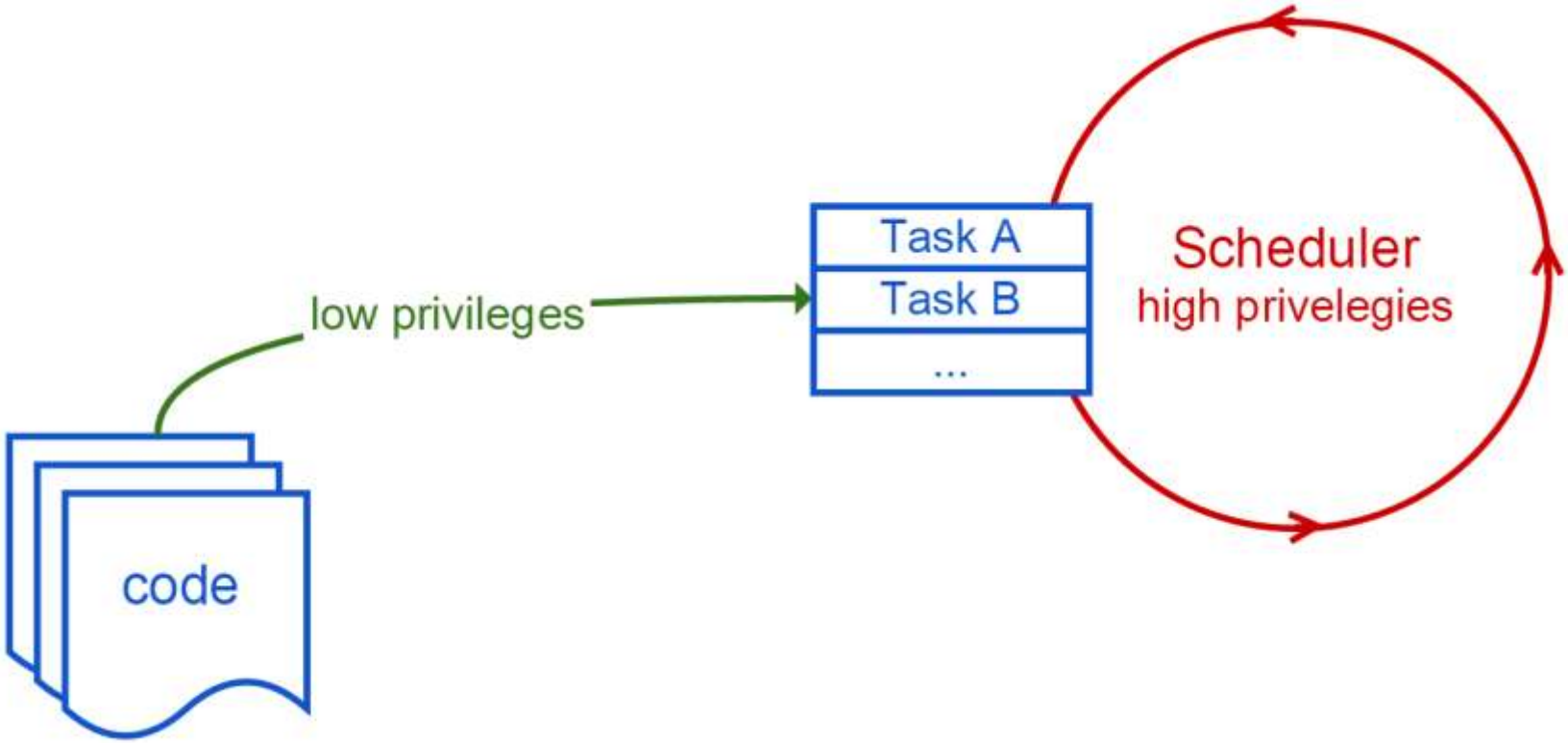


Trusted Chain Attack

- DynamicMethod class
- MS13-015 vulnerability

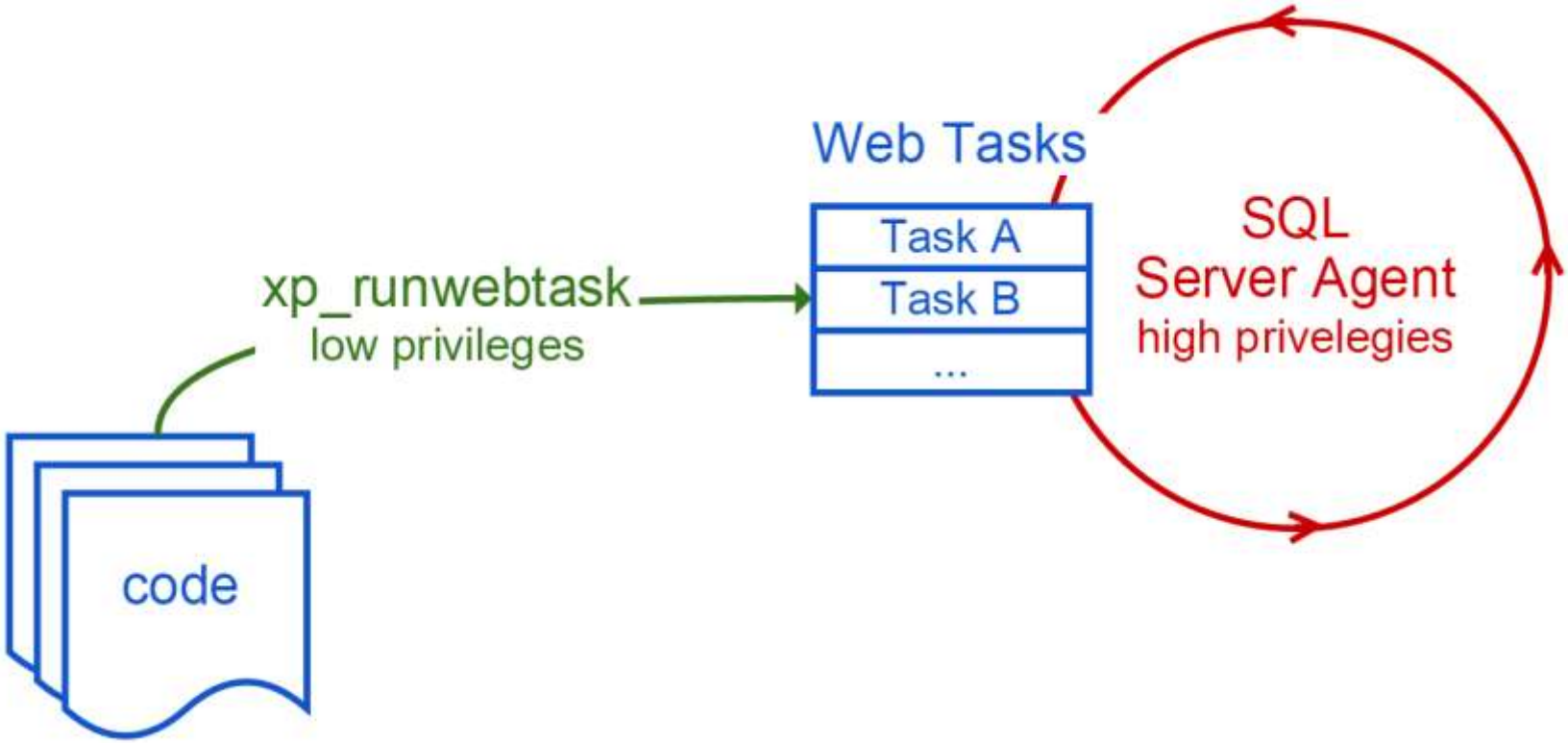
Could Allow Elevation of Privilege (KB2800277)

Luring Attack



Luring Attack

MS02-061 “Elevation of Privilege in SQL Server Web Tasks”



Exception Filter Attack

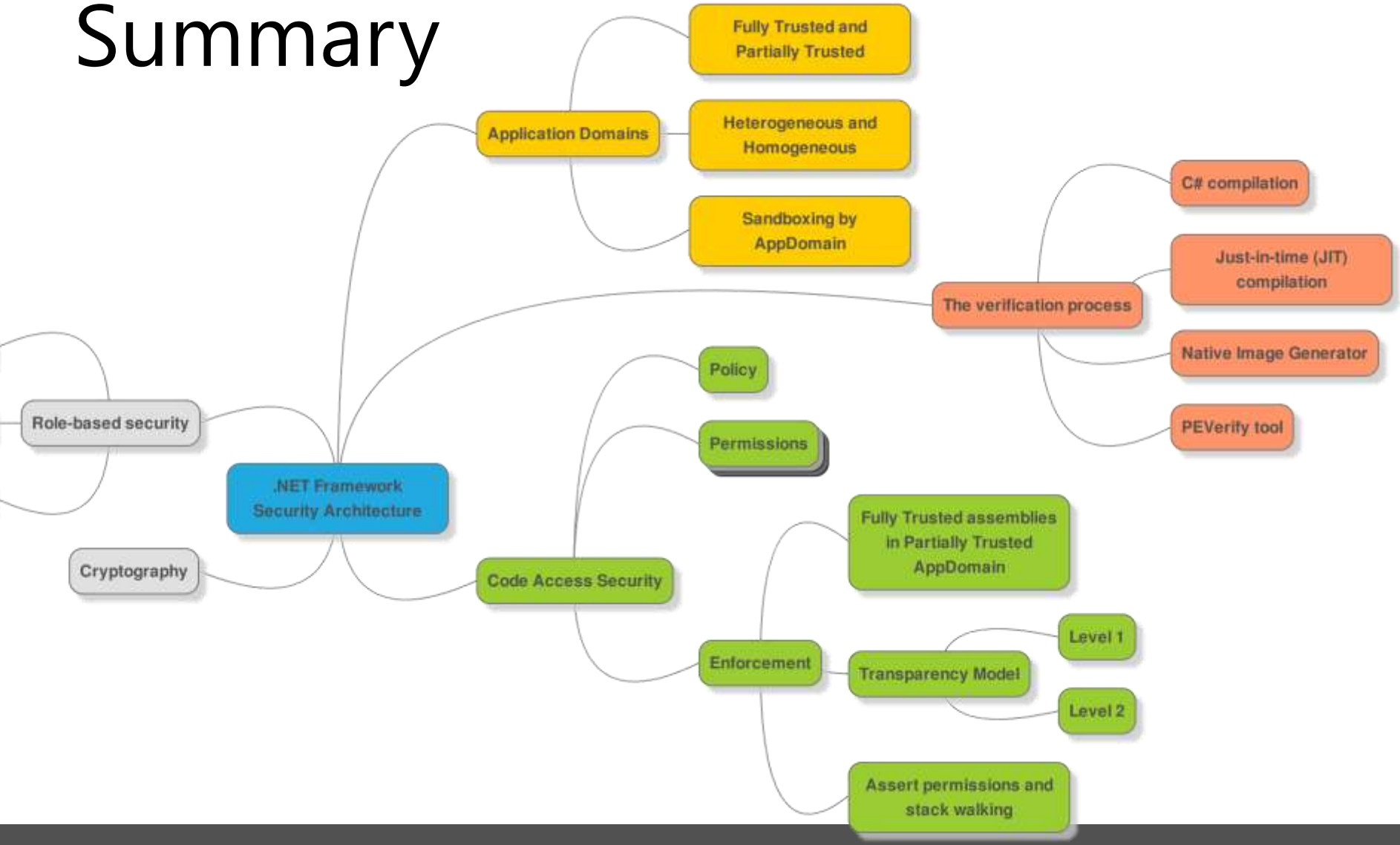
Exception Filter Attack

```
WindowsIdentity clientId = SqlContext.WindowsIdentity;
WindowsImpersonationContext impersonatedUser = null;

try
{
    impersonatedUser = clientId.Impersonate();
    if (impersonatedUser != null)
        return GetFileDetails(directoryPath);
    else return null;
}
finally
{
    if (impersonatedUser != null)
        impersonatedUser.Undo();
}
```

Exception Filter Attack

```
WindowsIdentity clientId = SqlContext.WindowsIdentity;
WindowsImpersonationContext impersonatedUser = null;

try
{
    try
    {
        impersonatedUser = clientId.Impersonate();
        if (impersonatedUser != null)
            return GetFileDetails(directoryPath);
        else return null;
    }
    finally
    {
        if (impersonatedUser != null)
            impersonatedUser.Undo();
    }
}
catch
{
    throw;
}
```

Summary


Fully Trusted and Partially Trusted
Application Domains
Heterogeneous and Homogeneous
Sandboxing by AppDomain
C# compilation
Just-in-time (JIT) compilation
The verification process
Native Image Generator
PEVerify tool
Role-based security
Policy
Permissions
.NET Framework Security Architecture
Fully Trusted assemblies in Partially Trusted AppDomain
Cryptography
Code Access Security
Level 1
Enforcement
Transparency Model
Level 2
Assert permissions and stack walking

Summary

Sandboxing:

- Exploring the .NET Framework 4 Security Model bit.ly/1zBHDl7
- New Security Model: Moving to a Better Sandbox bit.ly/1qdLTYf
- How to Test for Luring Vulnerabilities bit.ly/1G5asdG
- Using SecAnnotate to Analyze Your Assemblies for Transparency Violations bit.ly/12AtGZF

Summary

.NET Security:

- OWASP Top 10 for .NET developers bit.ly/1mpvG9R
- OWASP .NET Project bit.ly/1vCfknm
- Troy Hunt blog www.troyhunt.com
- The WASC Threat Classification v2.0 bit.ly/1G5d8rM

Thank you for your attention!

Mikhail Shcherbakov

IntelliEgg

spbdotnet.org

linkedin.com/in/mikhailshcherbakov

github.com/yuske

@yu5k3