POSITIVE TECHNOLOGIES

POSITIVE TECHNOLOGIES

Security Model in .NET Framework

Mikhail Shcherbakov
senior software developer
Positive Technologies

About me

— Senior software developer at Positive Technologies

— Working on Application Inspector - source code analysis product

— Former team lead at Acronis and Luxoft

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

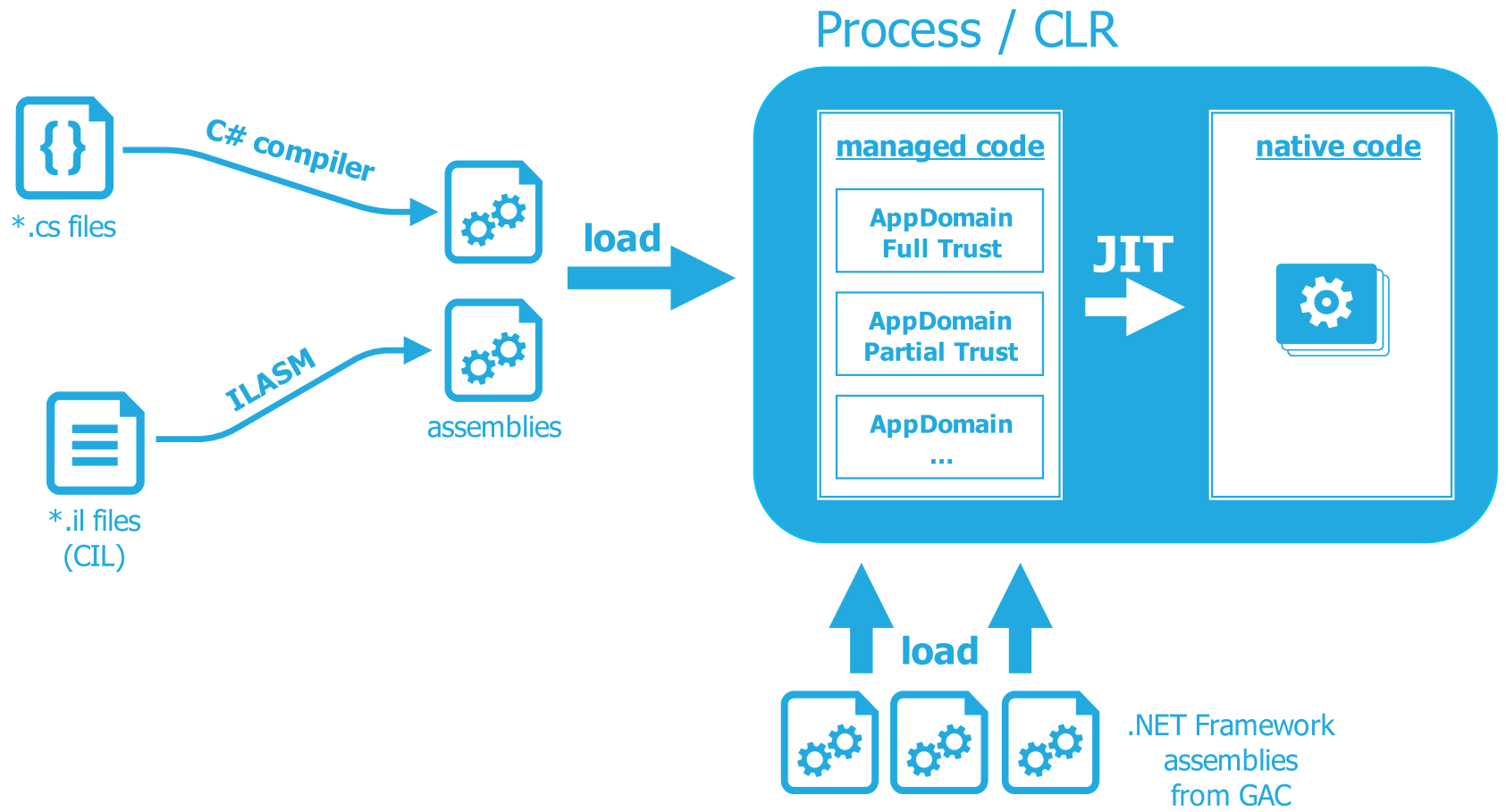
— Troubleshooting and knowledge about the internals

Knowledge in Practice

> — Are there some security features in Paint.NET that restrict what a plugin can do and what it can access?
>
> — There are no security features. And no, there is no guarantee of safety…
>
> — If there are no security features, then … whenever Paint.NET was running, it could look for interesting files and send them off to Russia.

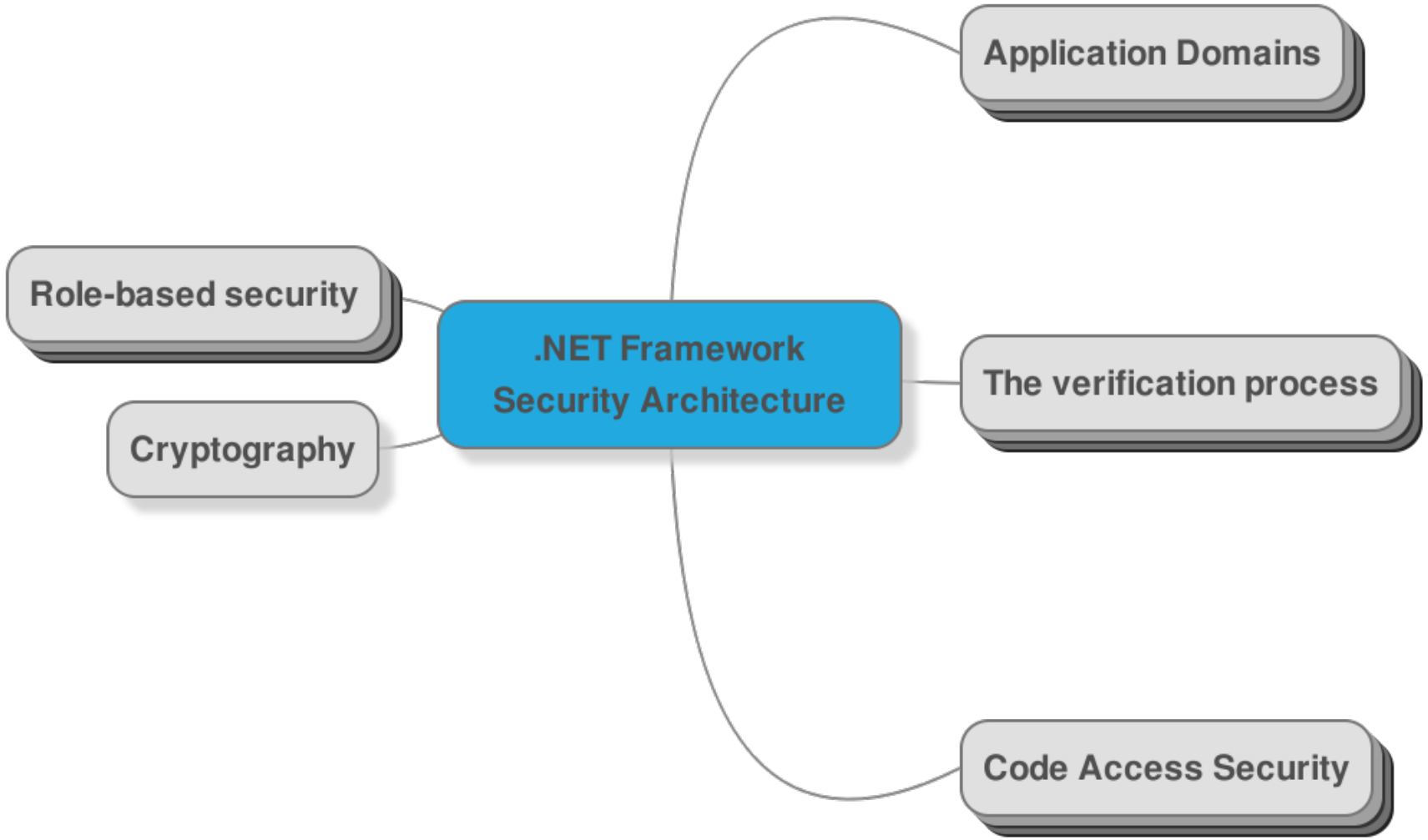
"Plugins & Security?" topic, Paint.NET Forum
http://bit.ly/1ABI3sH
#send2Russia

Terms

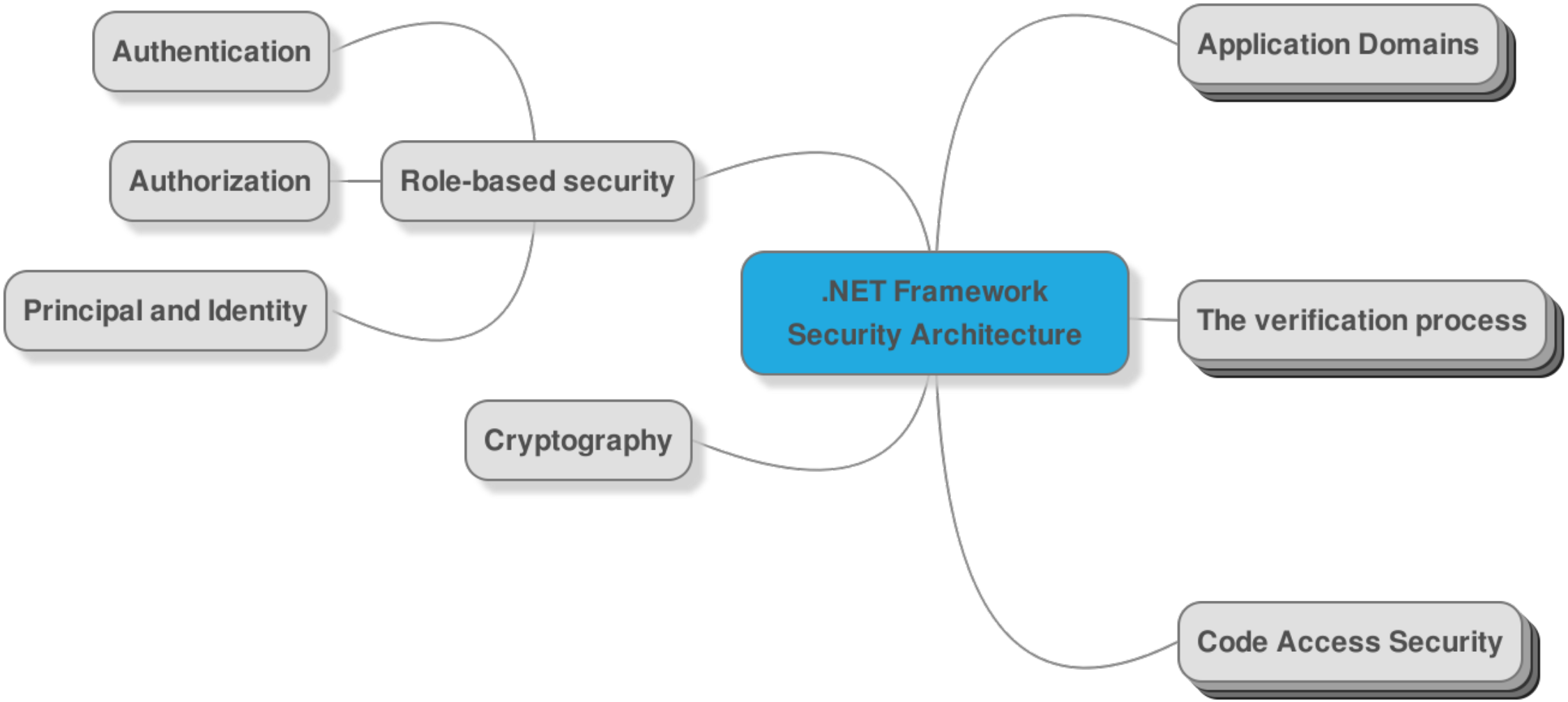


C# 5.0 Language Specification http://bit.ly/1tXdOI2
Common Language Infrastructure (CLI) Standard ECMA-335 http://bit.ly/1IesnAK

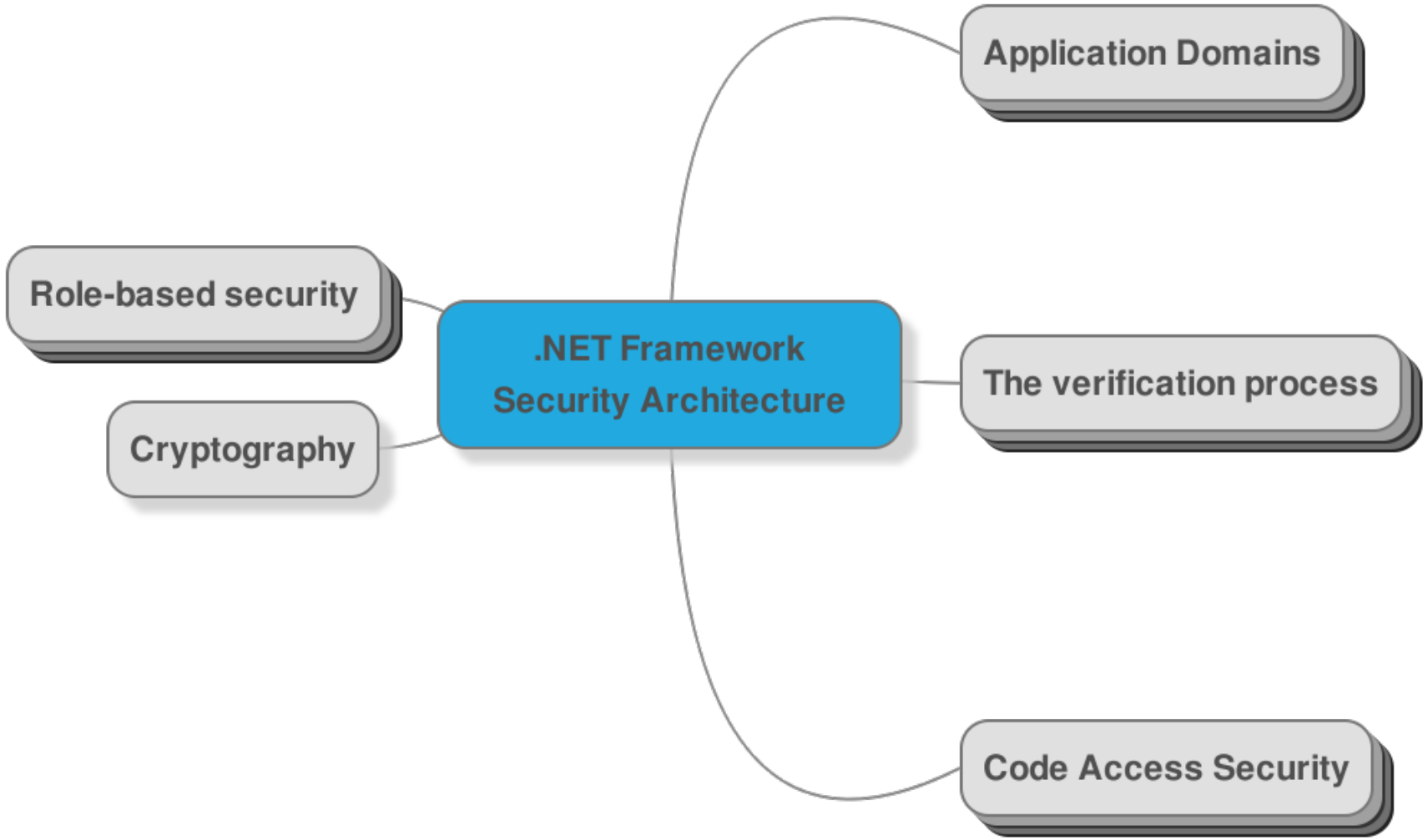
.NET Framework 4 Security Architecture



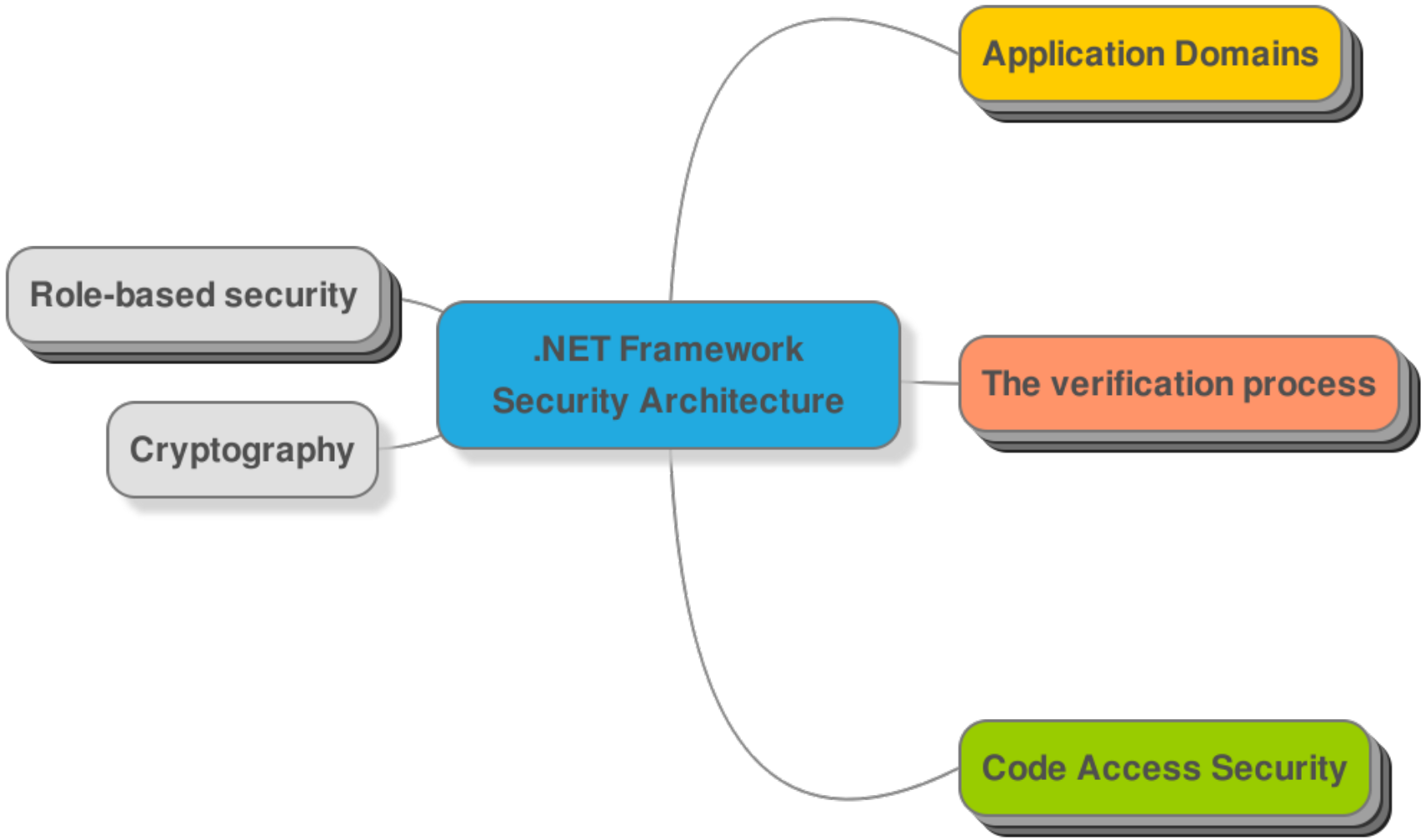
.NET Framework 4 Security Architecture



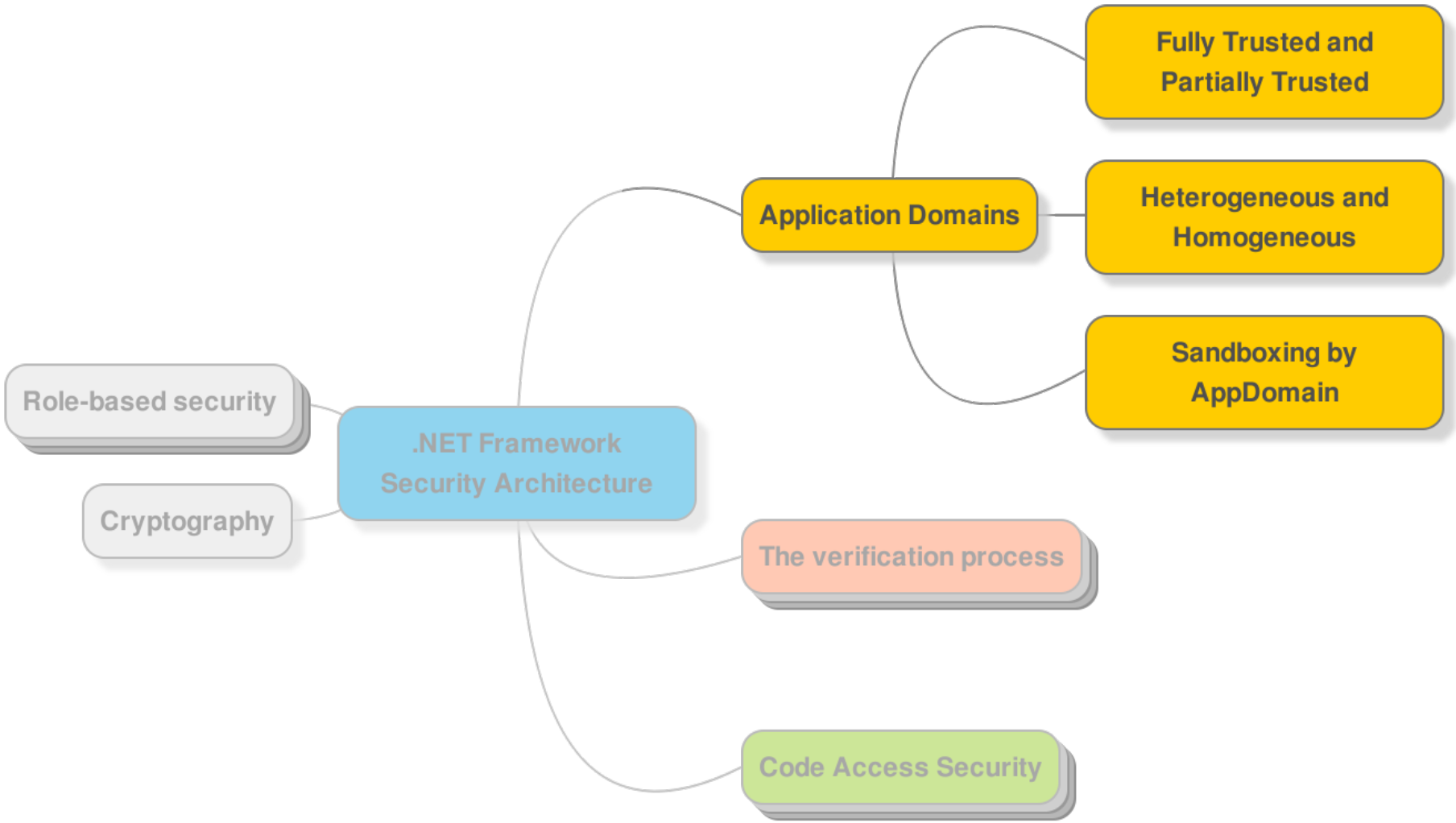
.NET Framework 4 Security Architecture



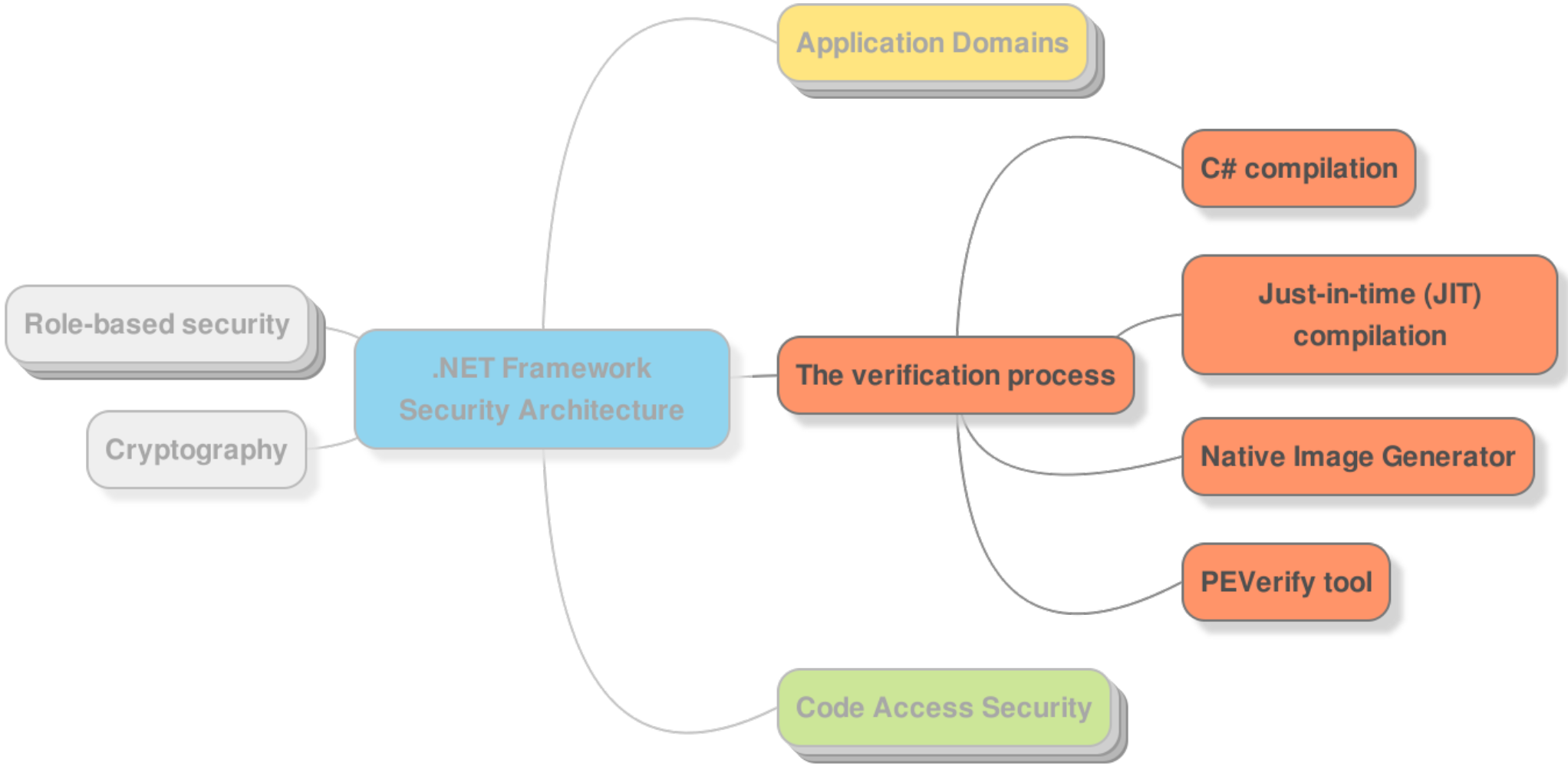
.NET Framework 4 Security Architecture



Application Domains

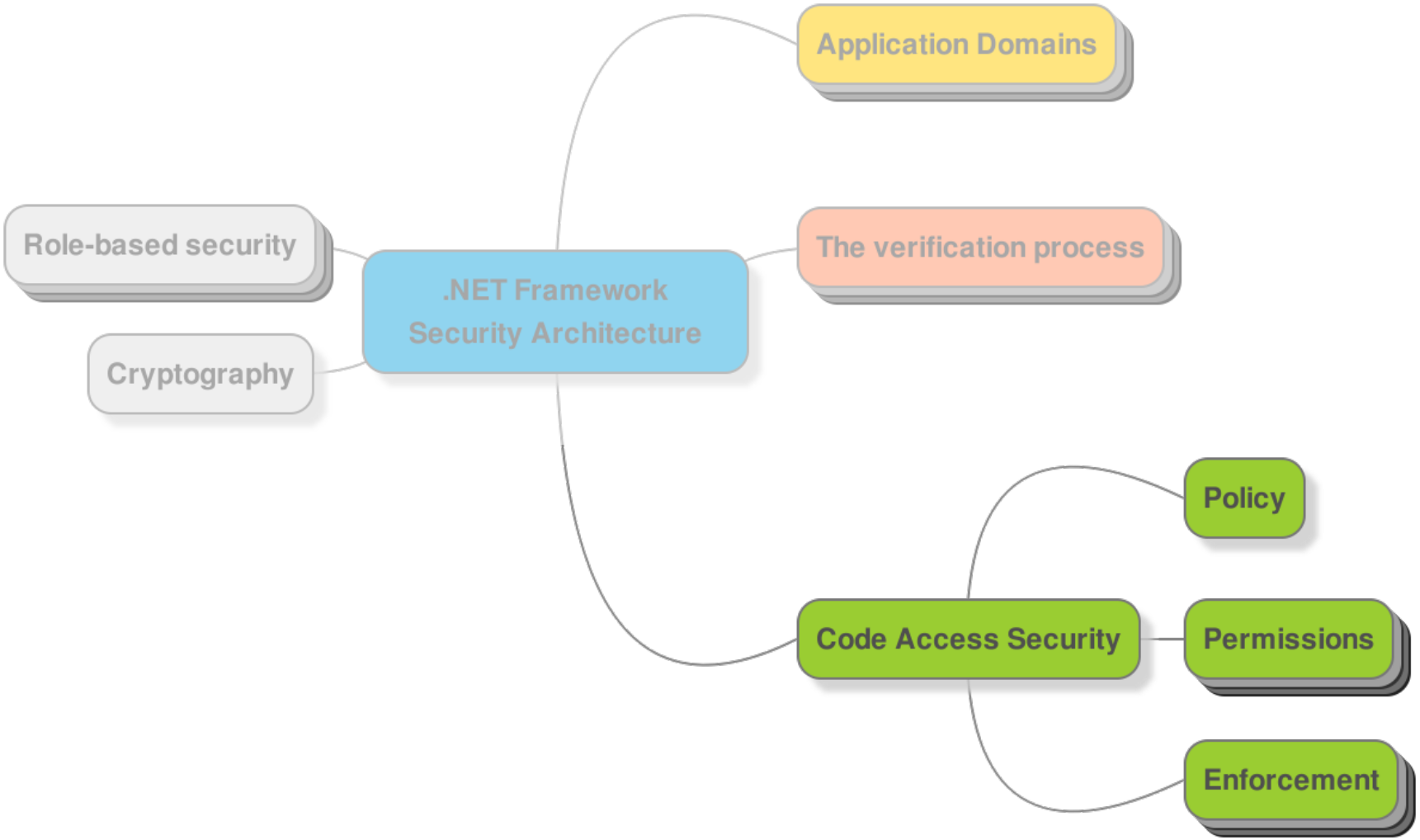


The verification process

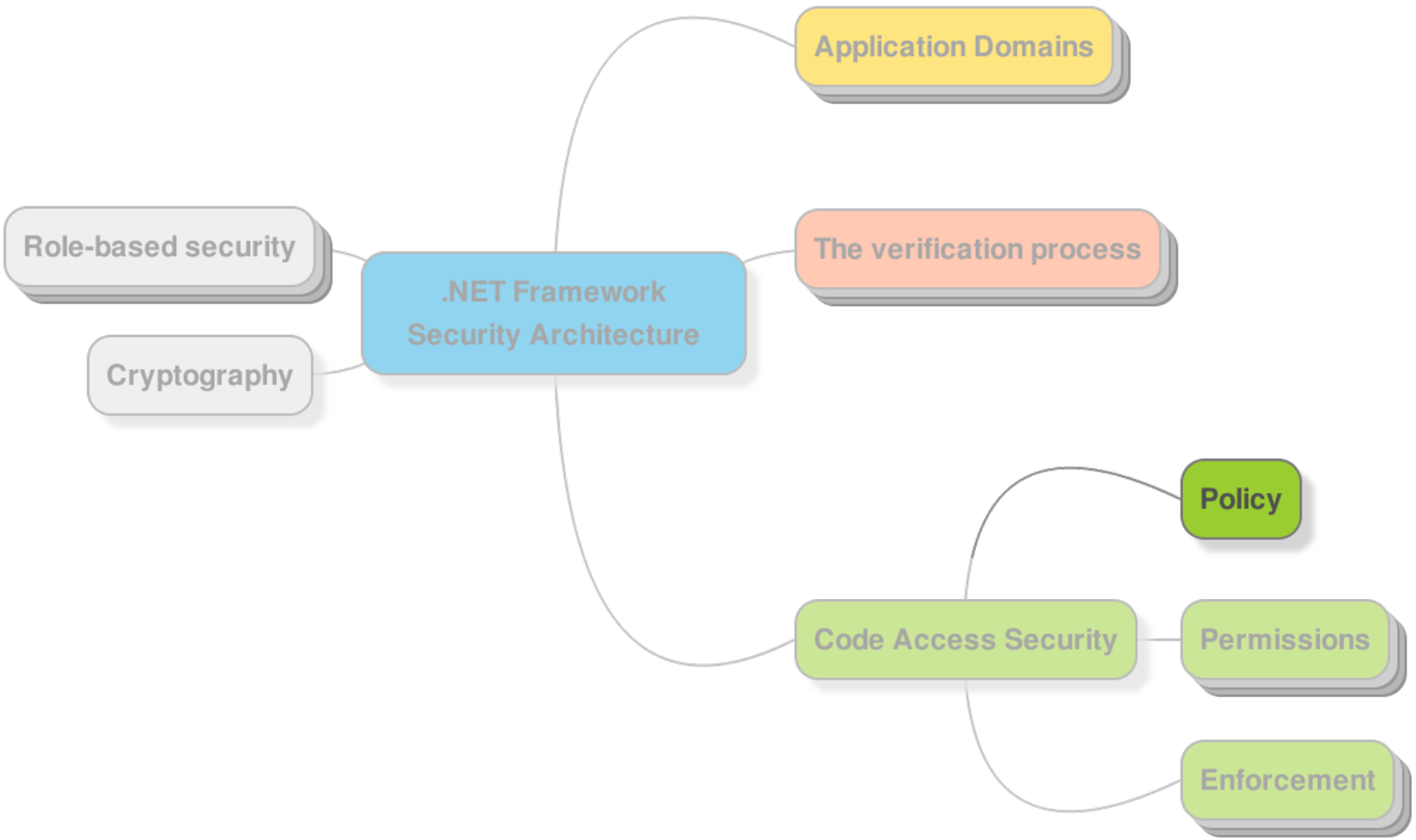


Just-in-time verification

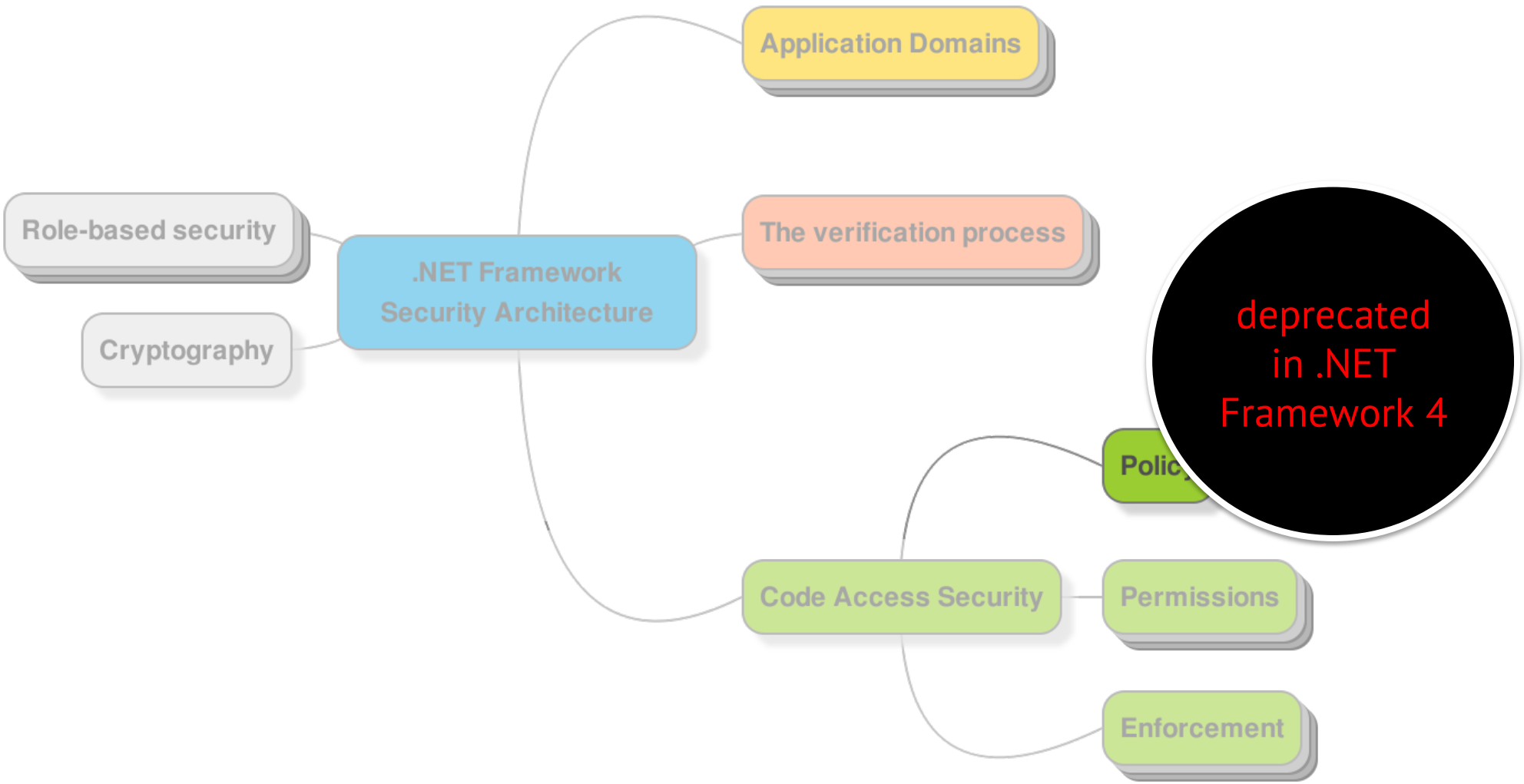
Code Access Security



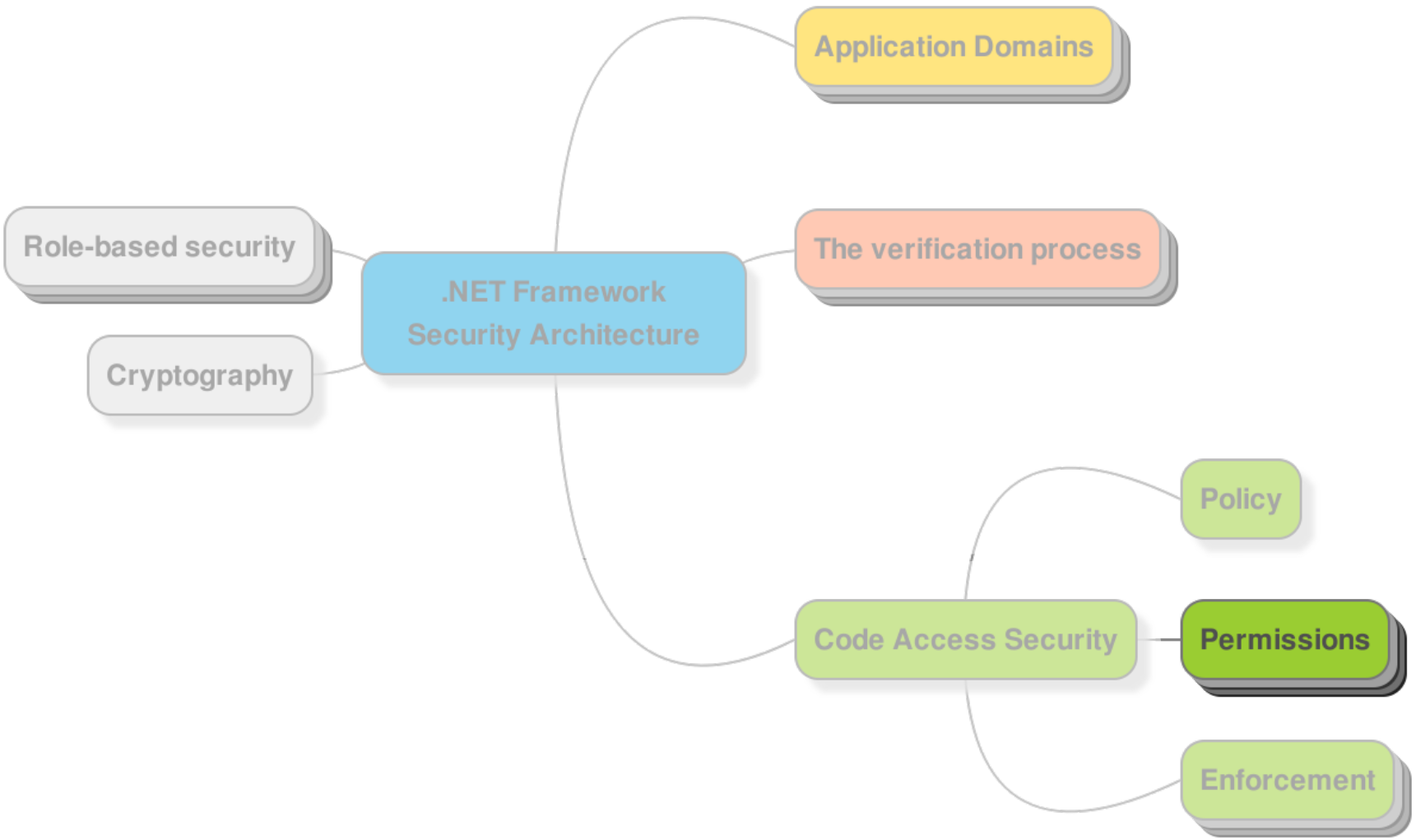
Policy


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

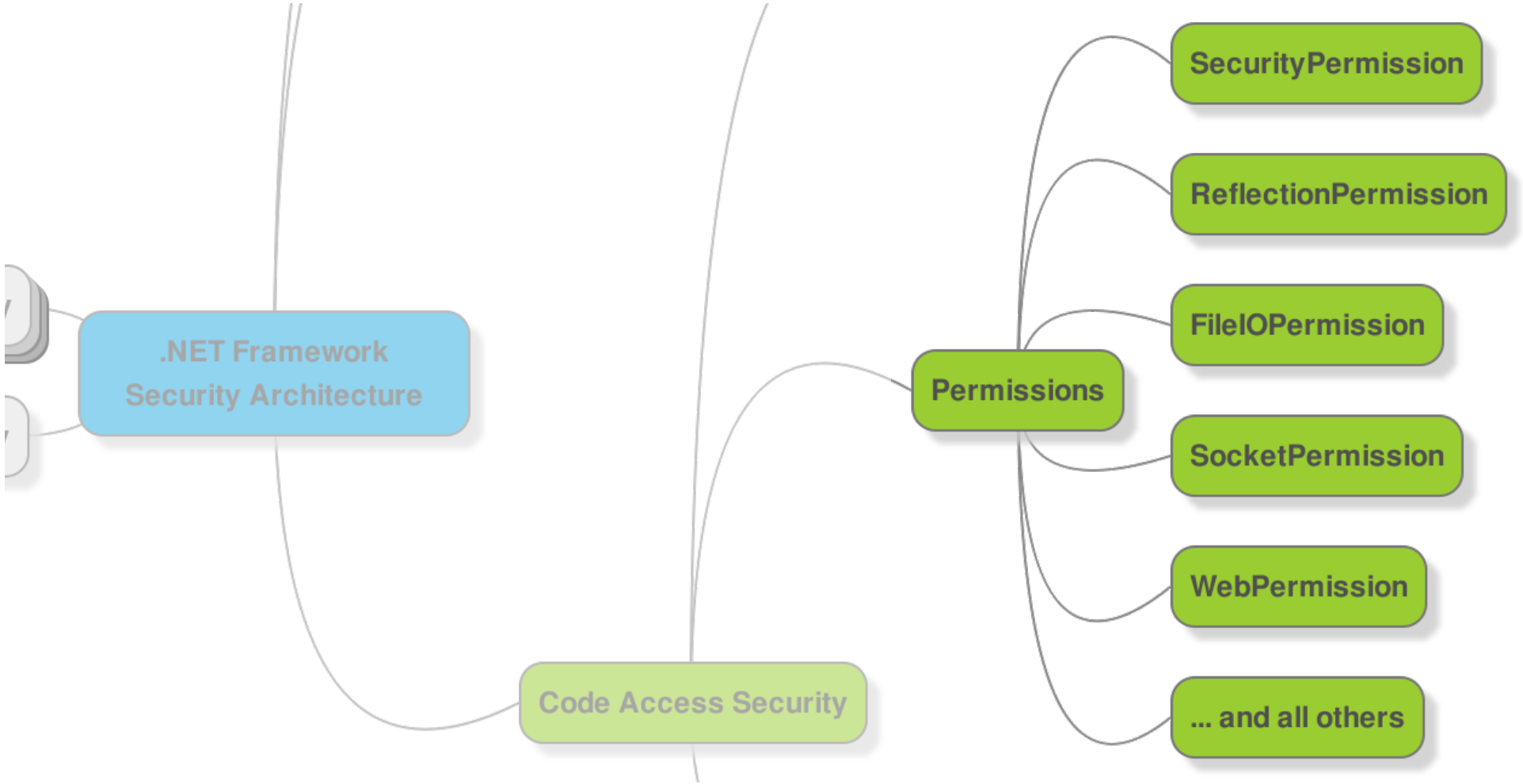
Policy


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

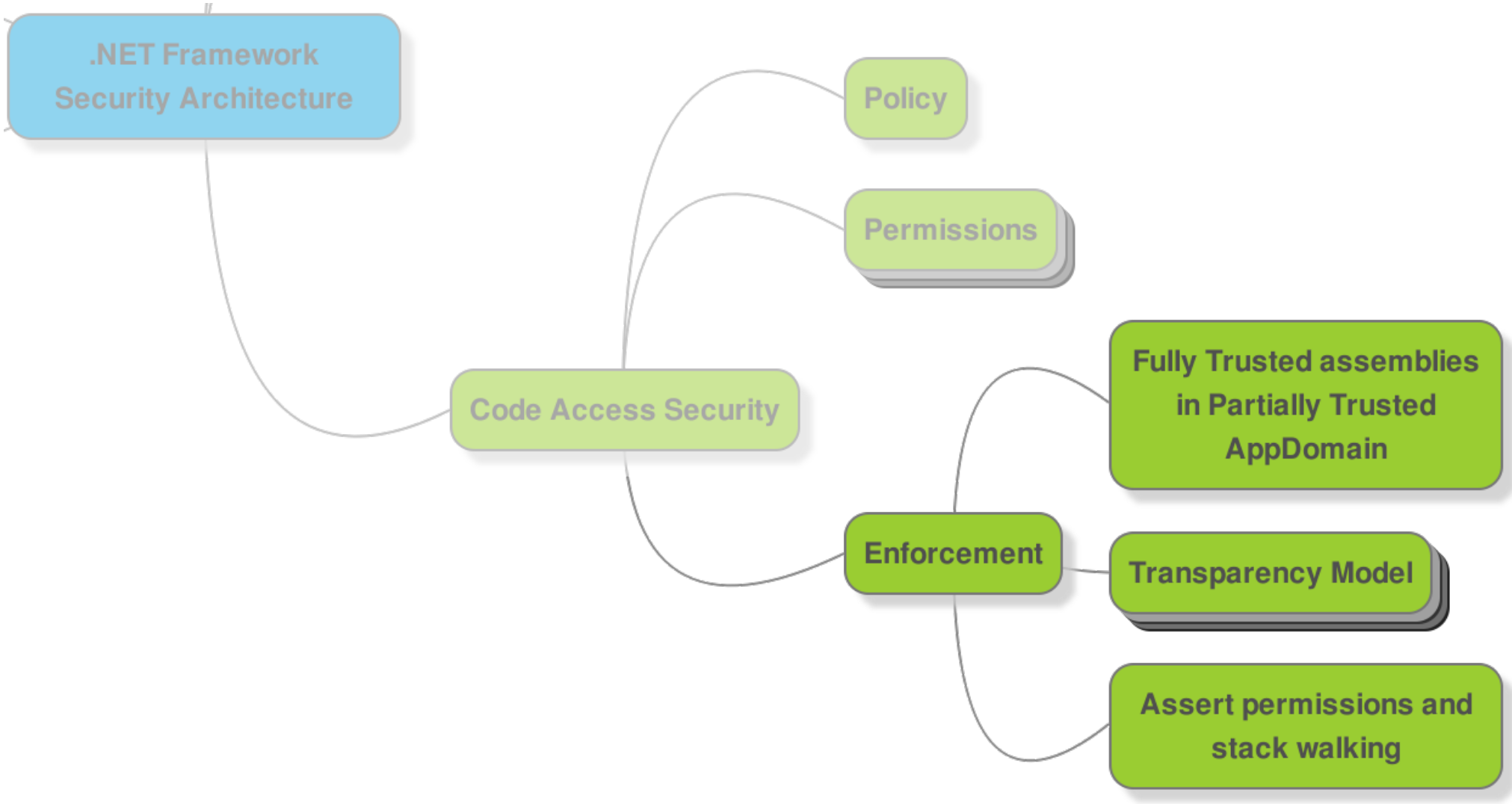
Permissions



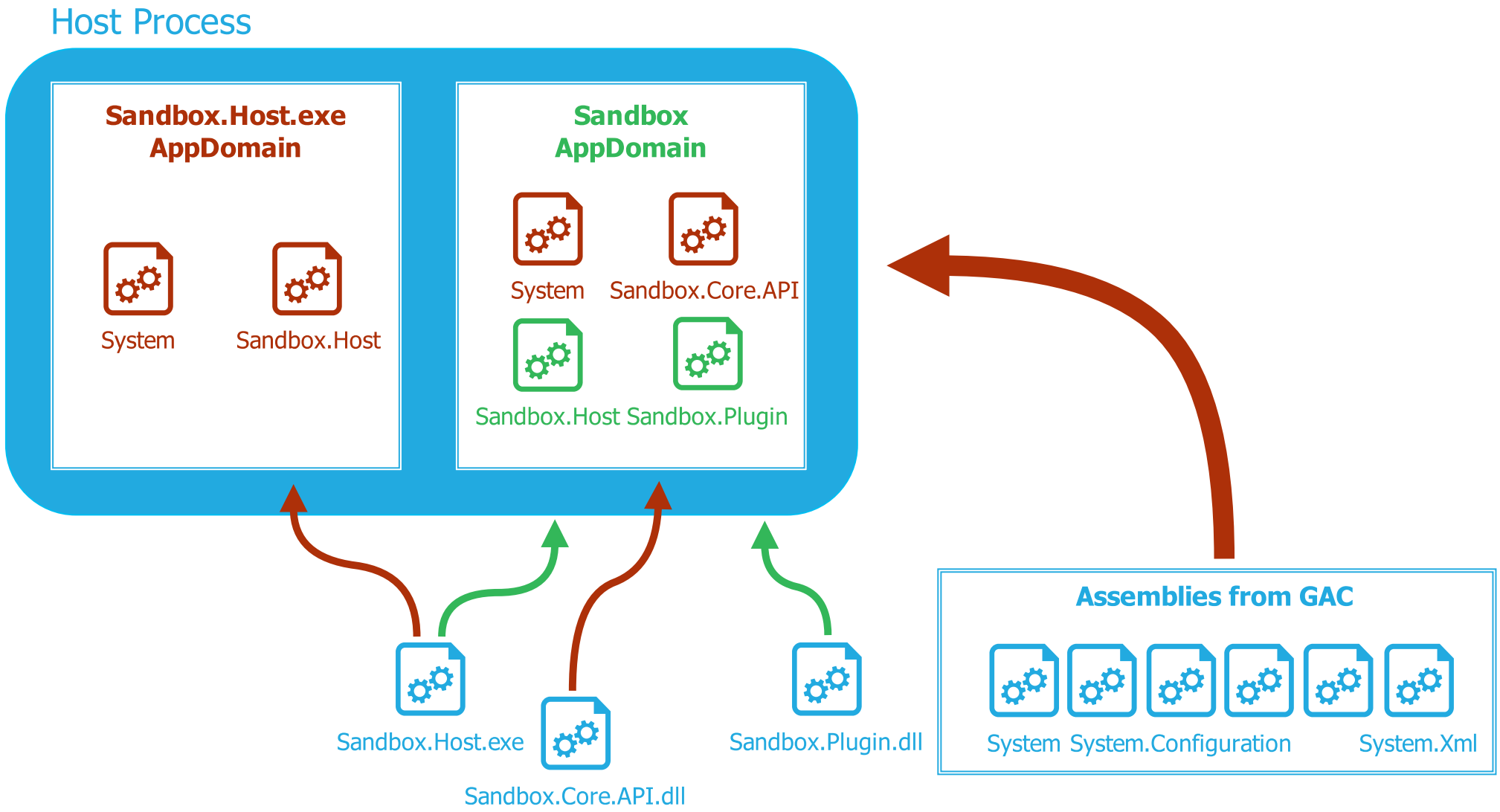
Permissions



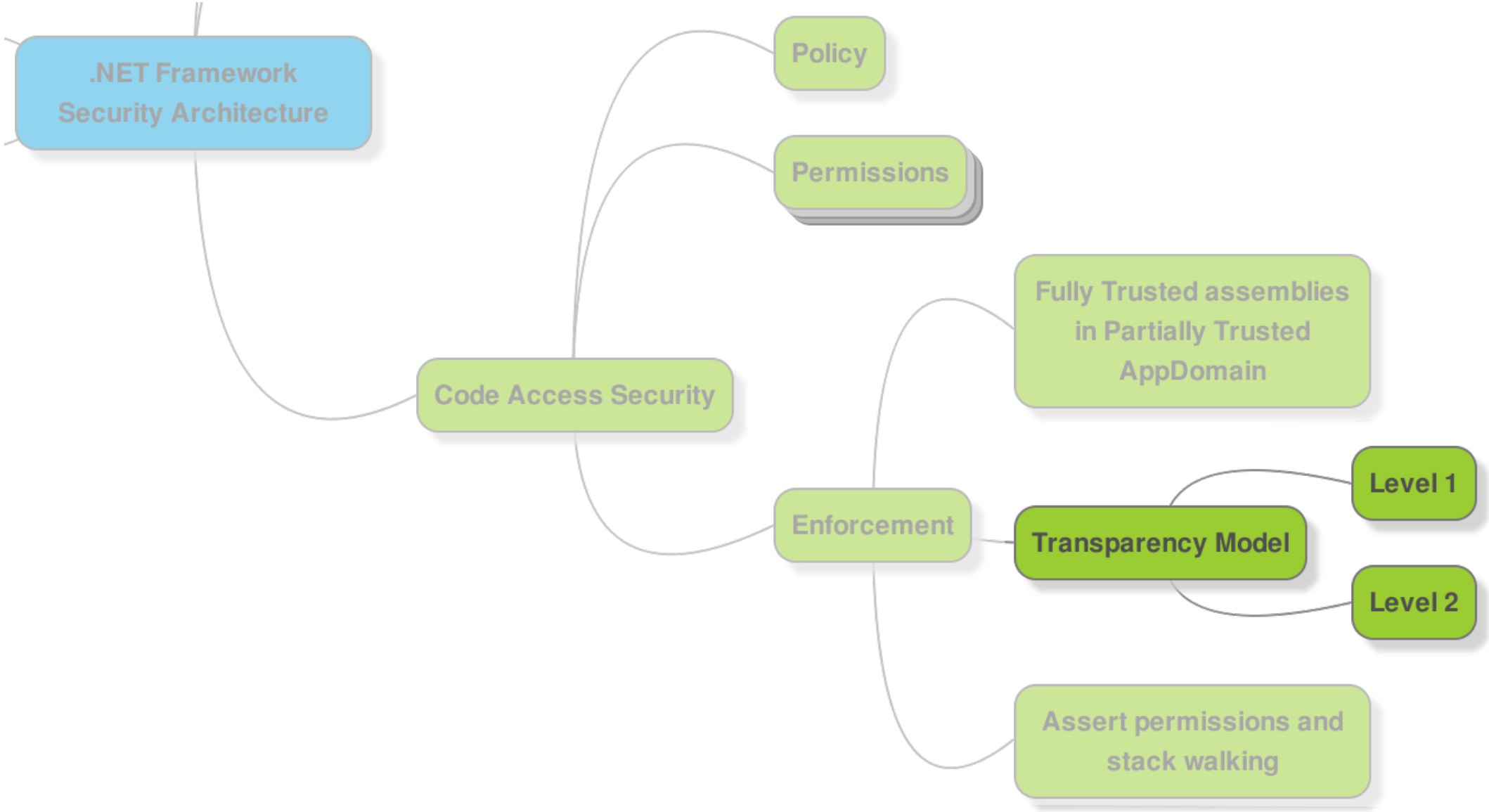
Enforcement



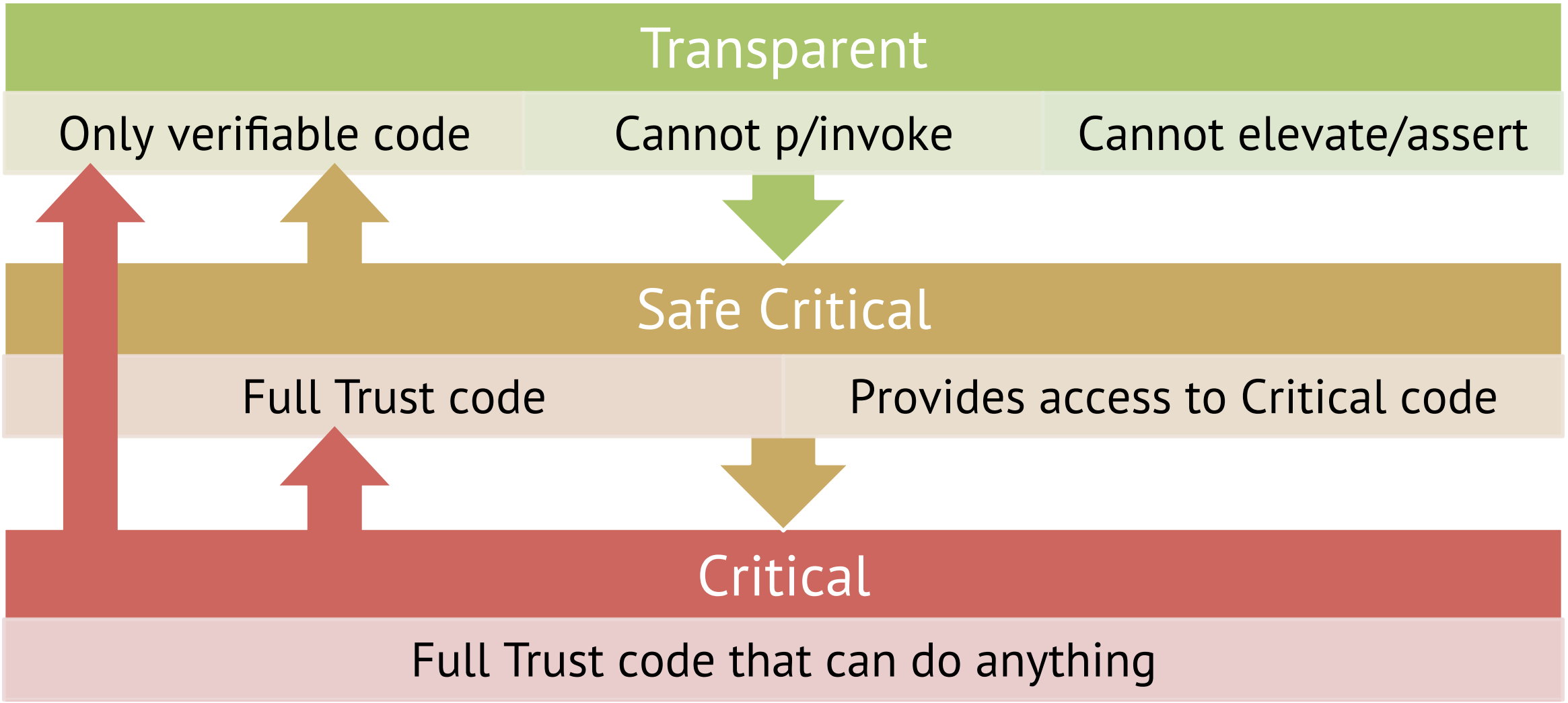
Fully Trusted code in Partially Trusted AppDomain



Transparency Model



Level 2 Security Transparency

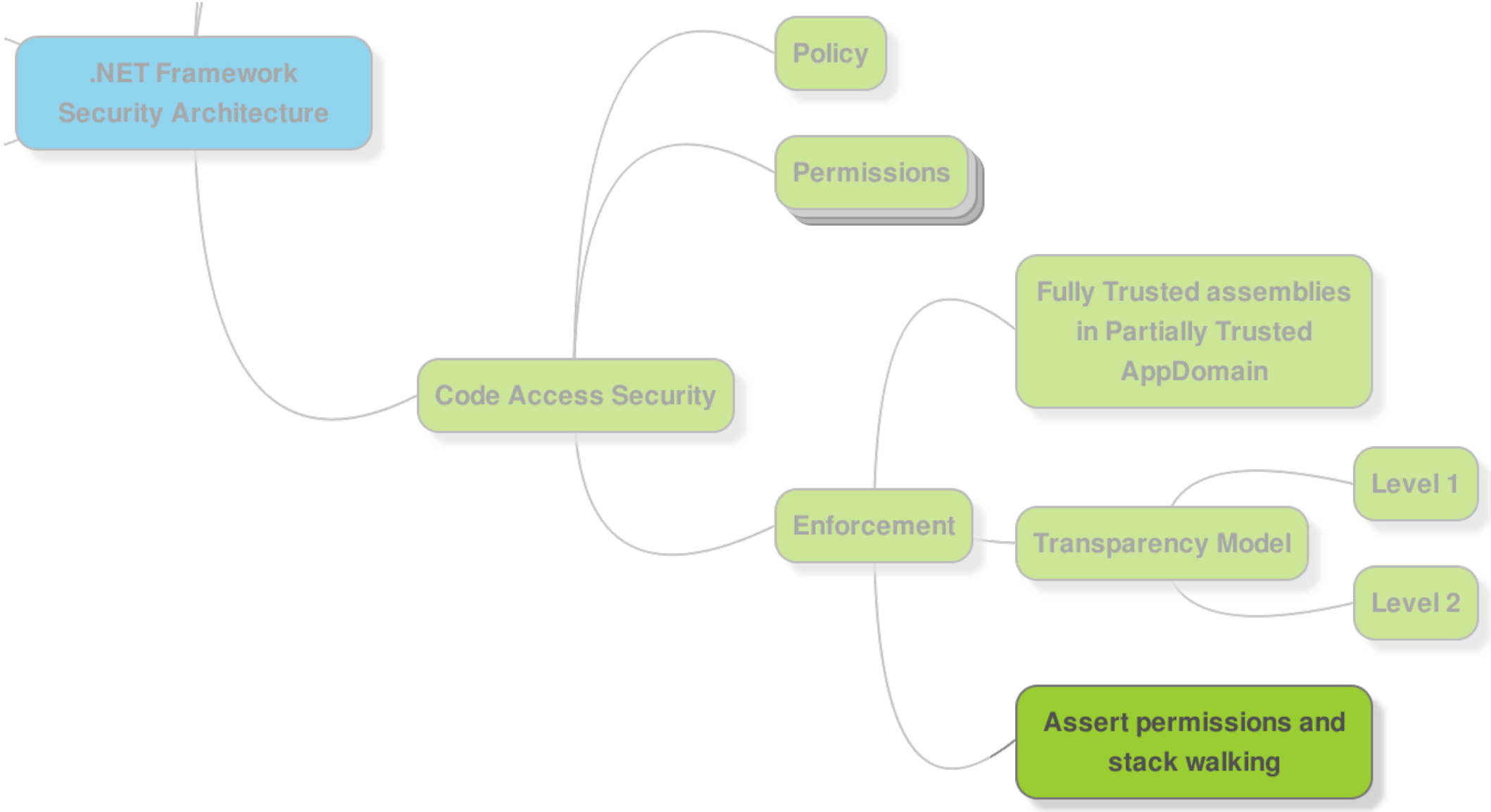


Security Transparency Attributes

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

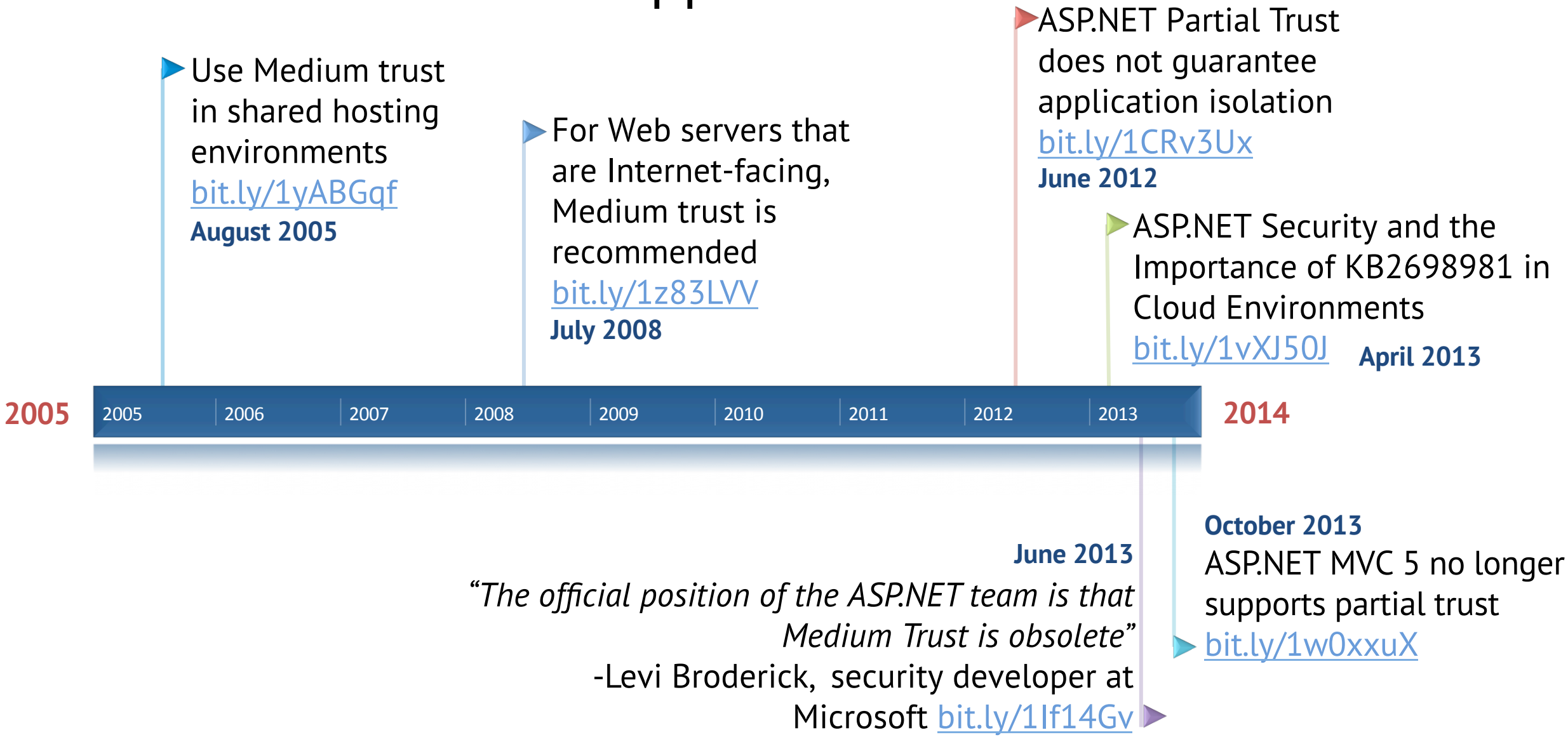
SecAnnotate.exe – .NET Security Annotator Tool http://bit.ly/1A3vMw3

Stack walking



Sandbox implementation

ASP.NET Partial Trust applications



Trusted Chain attack

— DynamicMethod class

— MS13-015 vulnerability

Could Allow Elevation of Privilege (KB2800277)

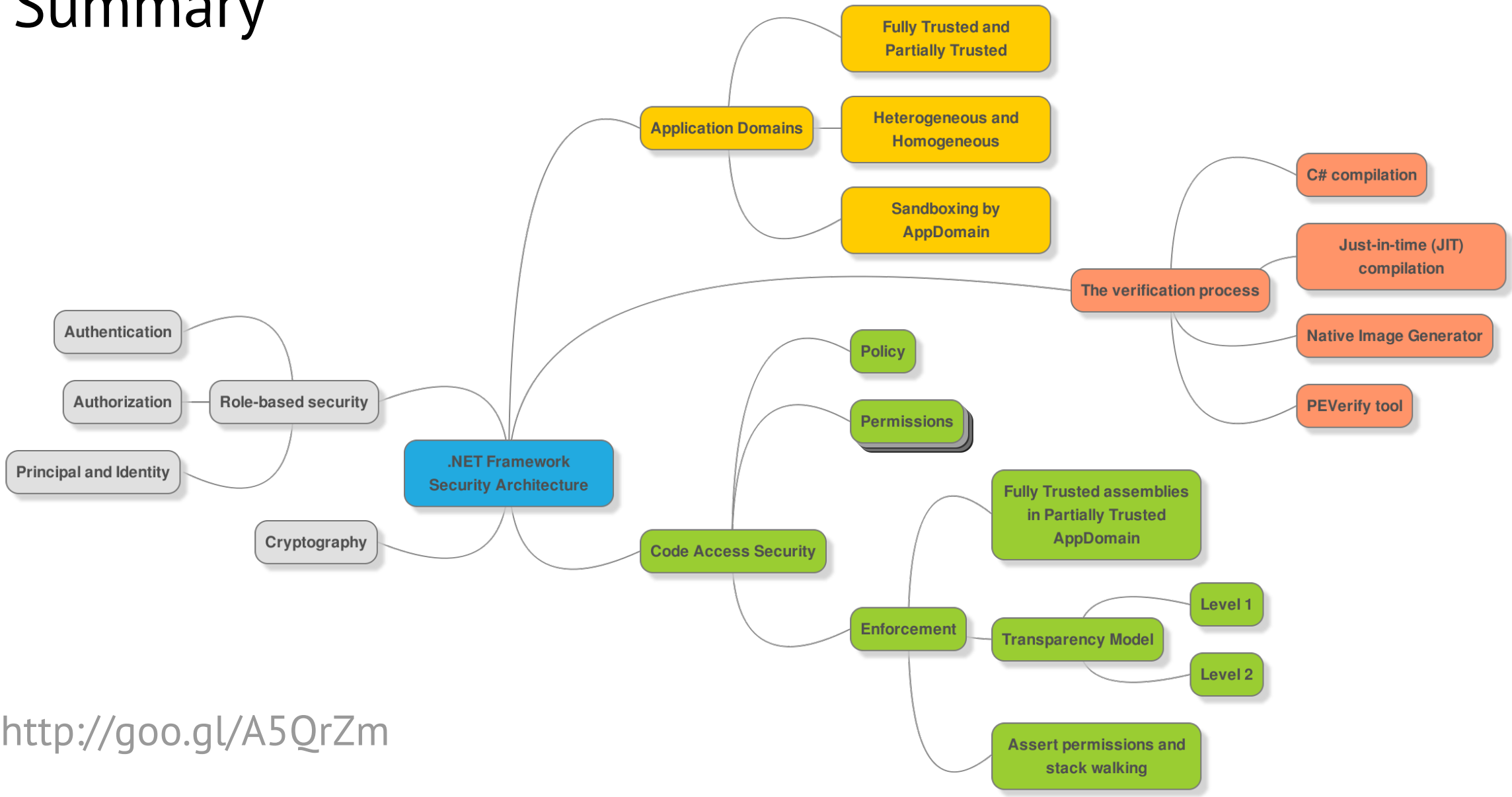
Trusted Chain attack

— DynamicMethod class

— MS13-015 vulnerability

Could Allow Elevation of Privilege (KB2800277)

Summary



http://goo.gl/A5QrZm

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

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

Thank you for your attention!

Mikhail Shcherbakov
Positive Technologies

linkedin.com/in/mikhailshcherbakov
yuske.dev@gmail.com
github.com/yuske
@yu5k3

POSITIVE TECHNOLOGIES