

Project Roslyn

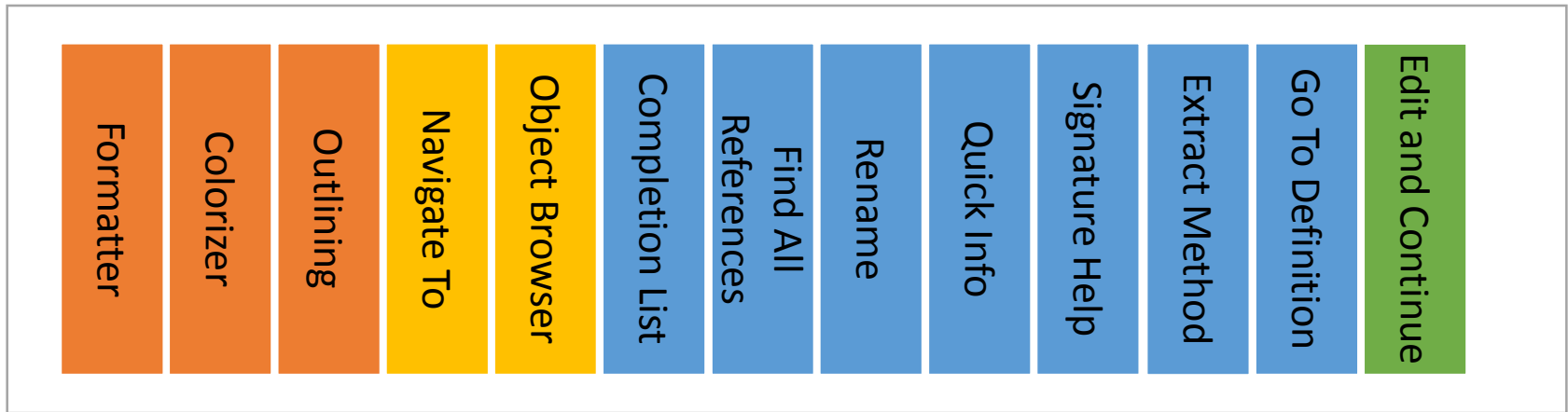
Exposing the C# and Visual Basic compilers' code analysis

Dustin Campbell
Senior Program Manager
Microsoft

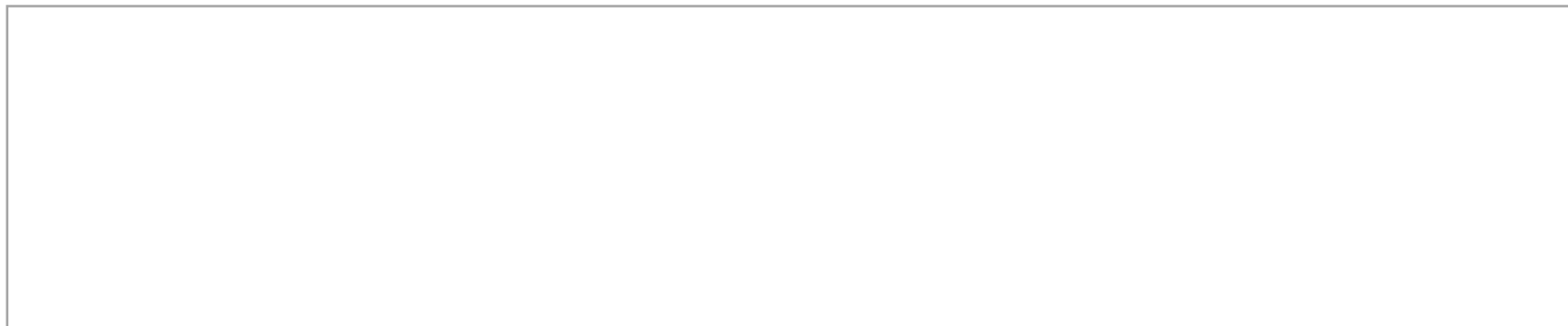
Project Codename “Roslyn”

- Scope
 - Managed compilers & language services
 - Code analysis APIs
 - Scripting APIs
 - Language service extensibility
 - Read-Eval-Print-Loop (REPL)
- Ship Date
 - v.Next + n, where $n > 0$

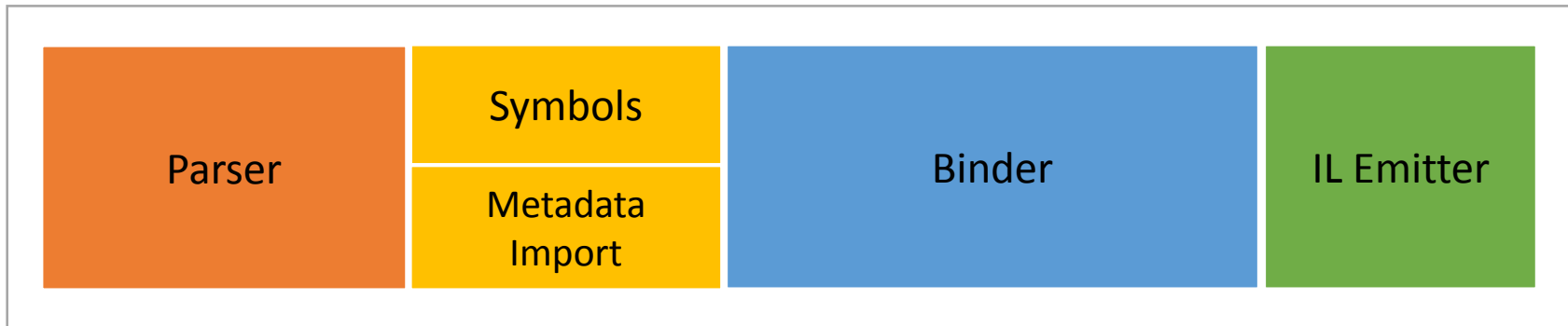
Language Service



Compiler API



Compiler Pipeline



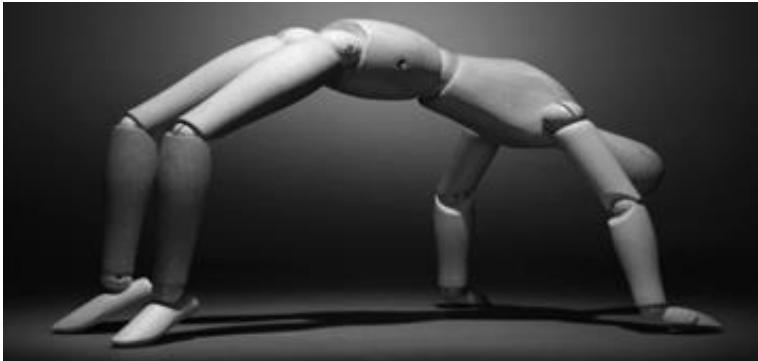
Design Choices



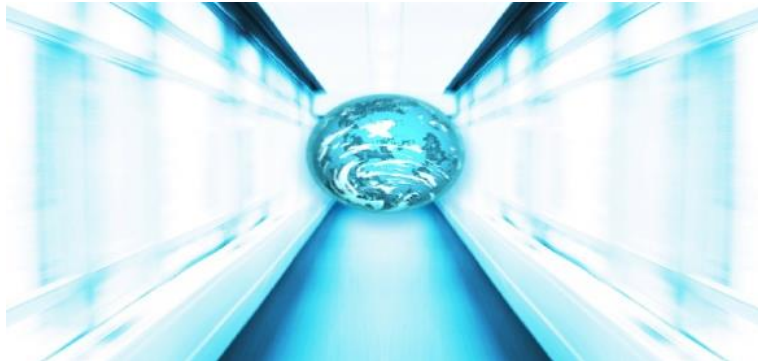
Immutable



Complete



Resilient

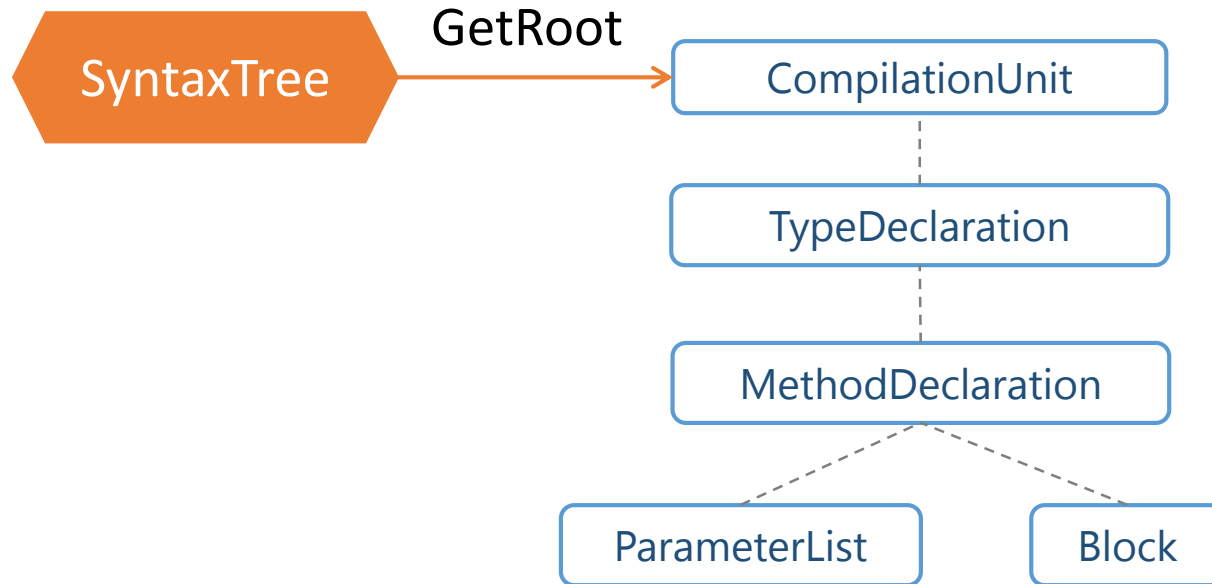


Efficient

Demo: Compiler APIs

Major Players

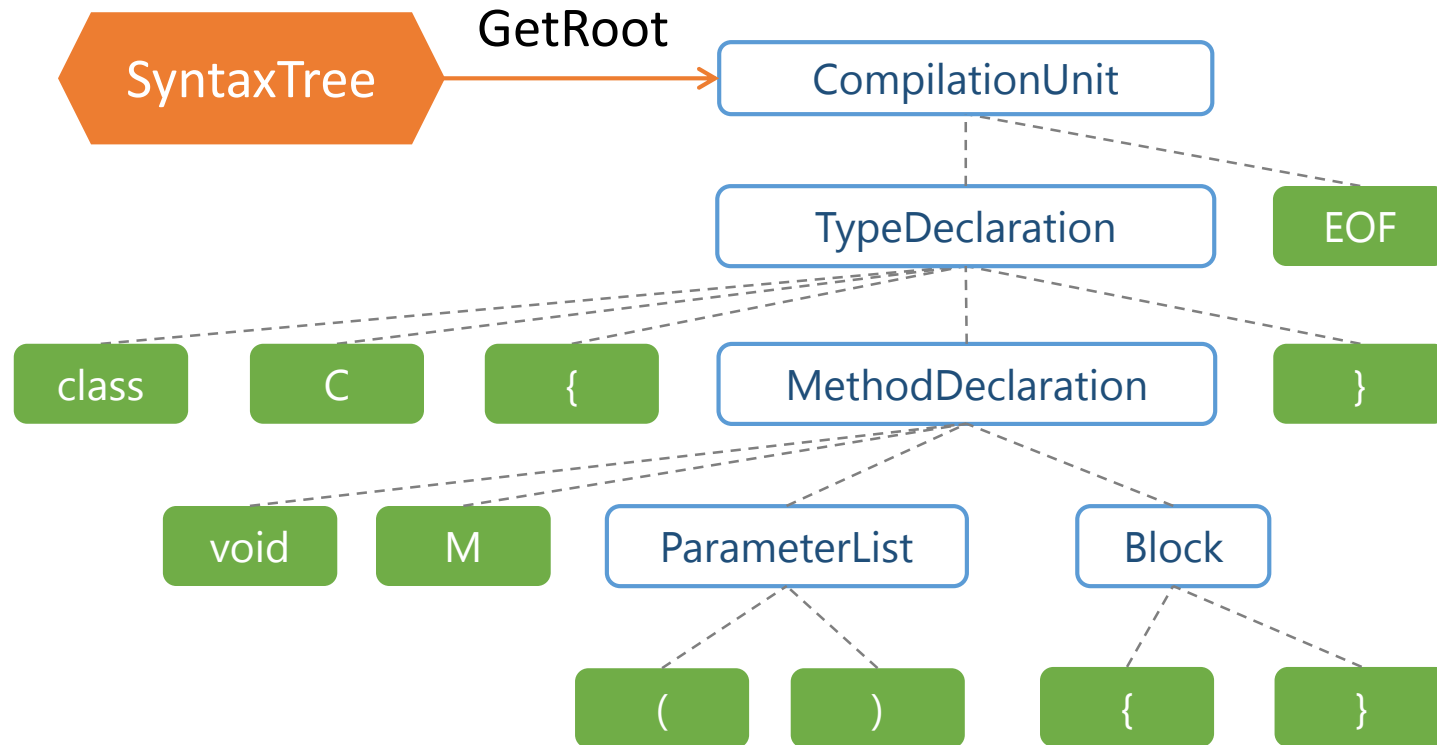
Syntax Nodes



```
class C
{
    void M()
    {
    }
} // C
```

Major Players

Syntax Tokens



```
class C
{
    void M()
} // C
```

Major Players

Syntax Tokens

class

C

{

void

M

(

)

{

}

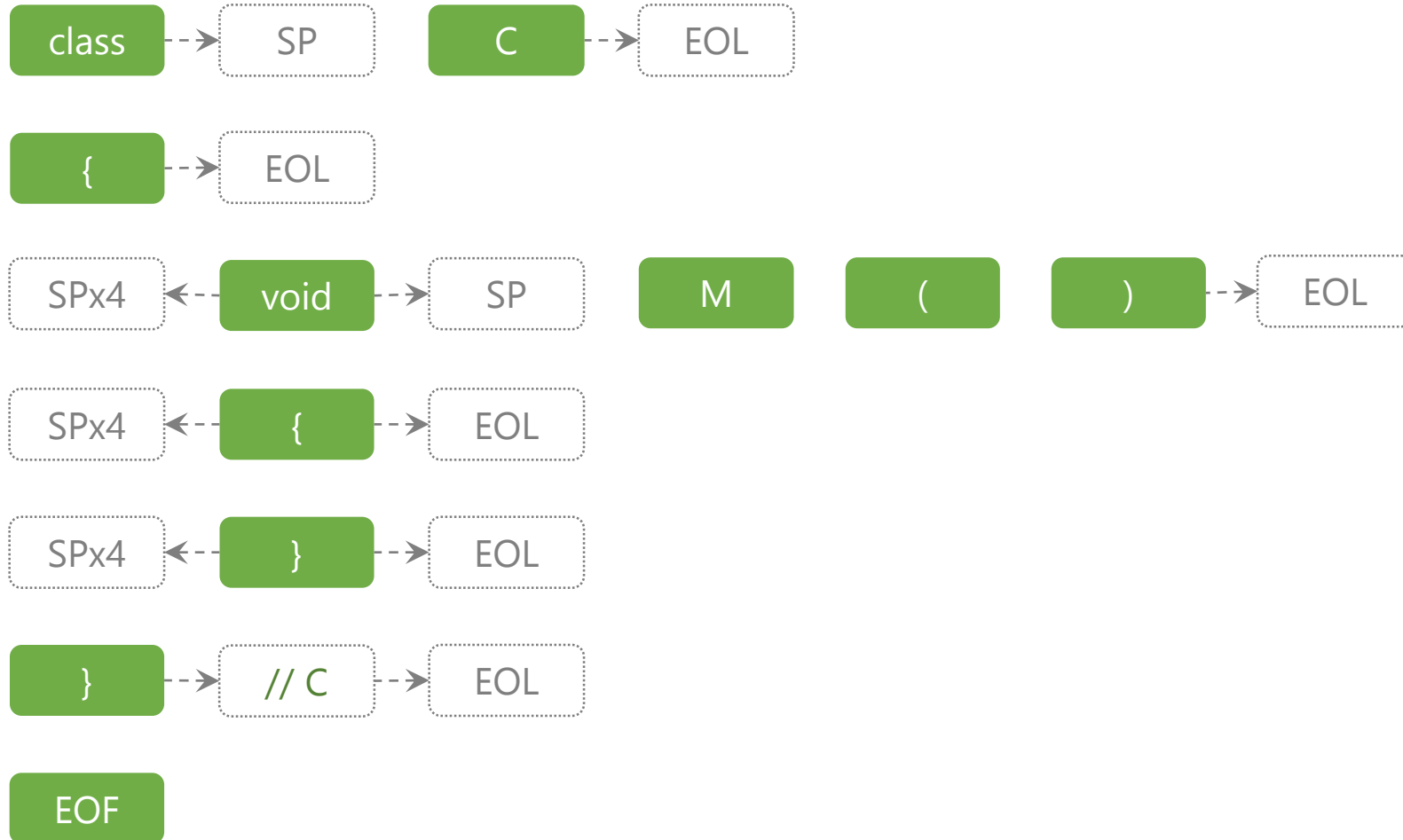
}

EOF

```
class C  
{  
    ... void M()  
    ... {  
    ... }  
} // C
```


Major Players

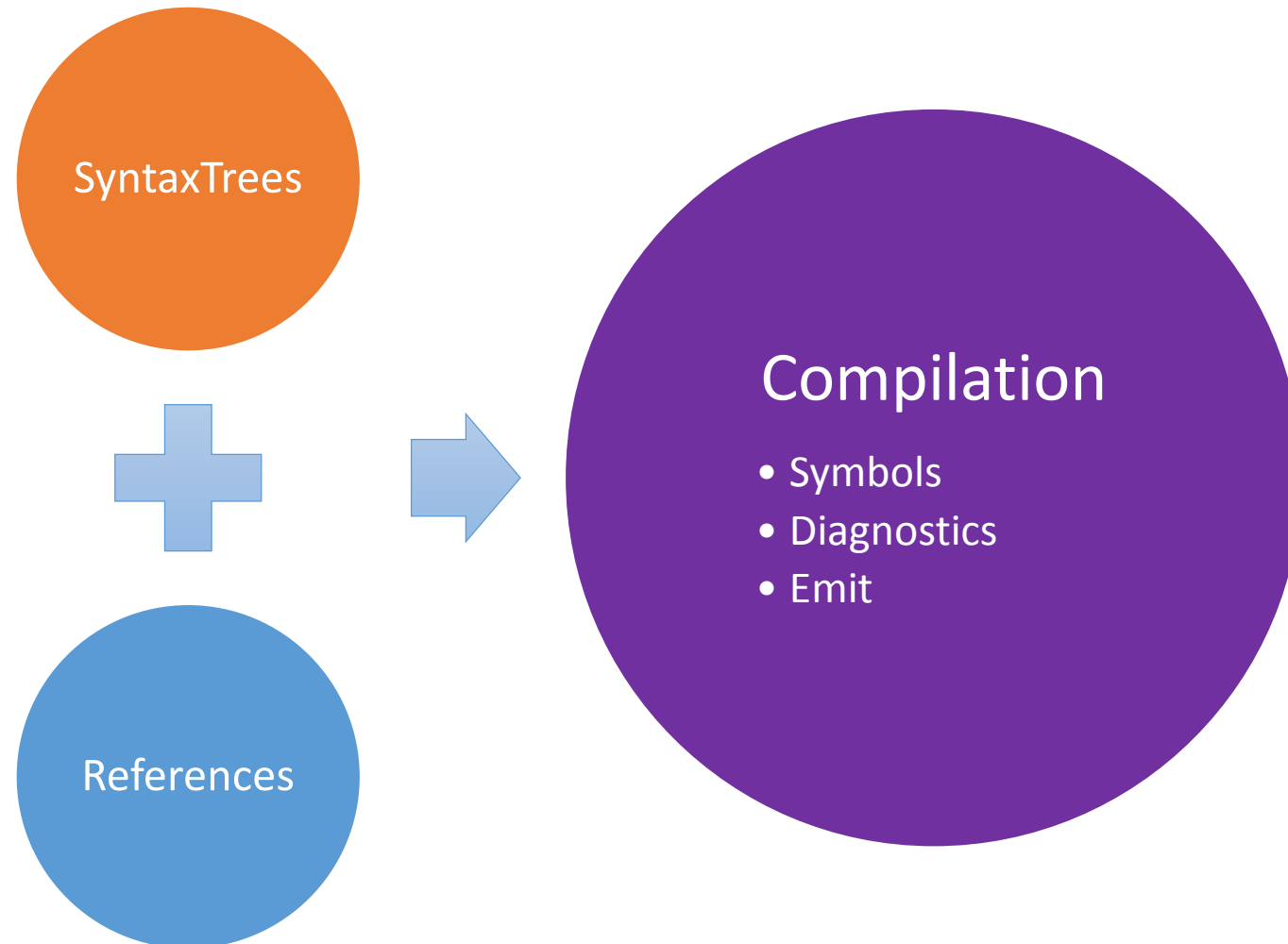
Syntax “Trivia”



```
class C  
{  
    ... void M()  
    ... {  
    ... }  
} // C
```

Major Players

Compilations, References and Symbols



Major Players

Semantics

GetSemanticModel(tree)

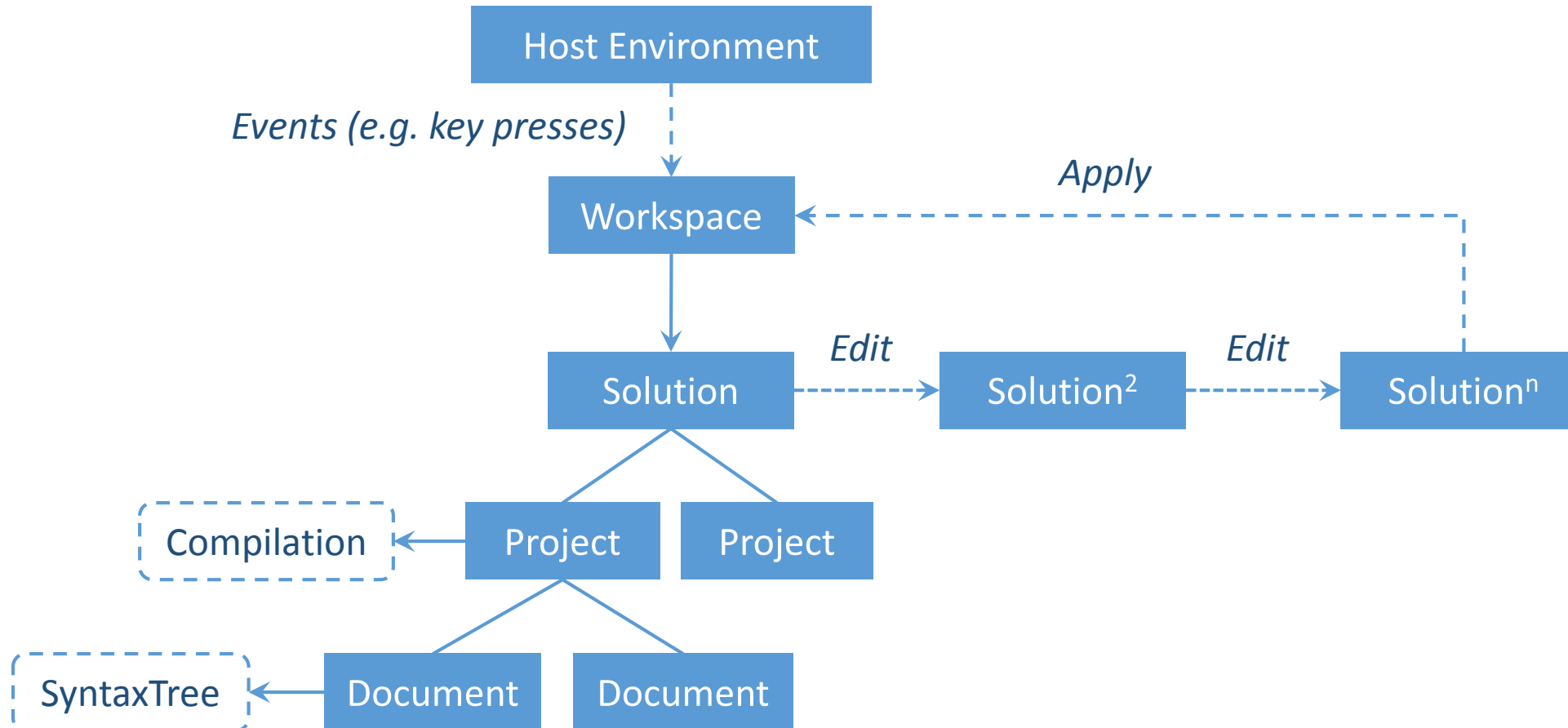


SemanticModel

- Binding Information
- Conversion Classification
- Flow Analysis
- Overload Resolution
- Etc.

Major Players

Workspace, Solutions, Projects and Documents



Demo: Building a Quick Fix

Next Steps

- Install the Roslyn CTP
- Try out the APIs
- Give us feedback!

Dev Center:

<http://msdn.com/roslyn>

Forum:

<http://social.msdn.microsoft.com/forums/en-us/roslyn>

Email:

dustinca@microsoft.com