Using Roslyn to improve your codebase

Jiří Činčura | @cincura_net

Before

- Blackbox
- Compiler vs. Visual Studio language services
- Written in unmanaged code ☺
- Refactorings, renames, references, …

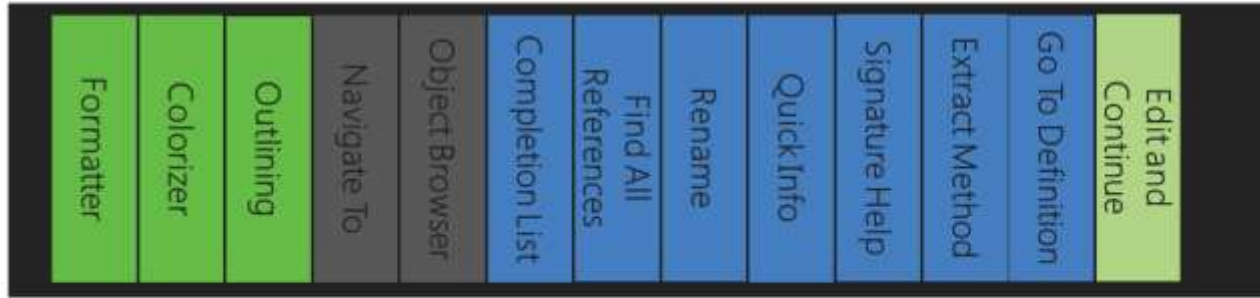# Roslyn

- Next (currently last ☺) generation C# (etc.) compiler
- Open platform
- i.e. access to syntax and semantic models
- Rich API
- Open source

Roslyn

- Almost completely immutable
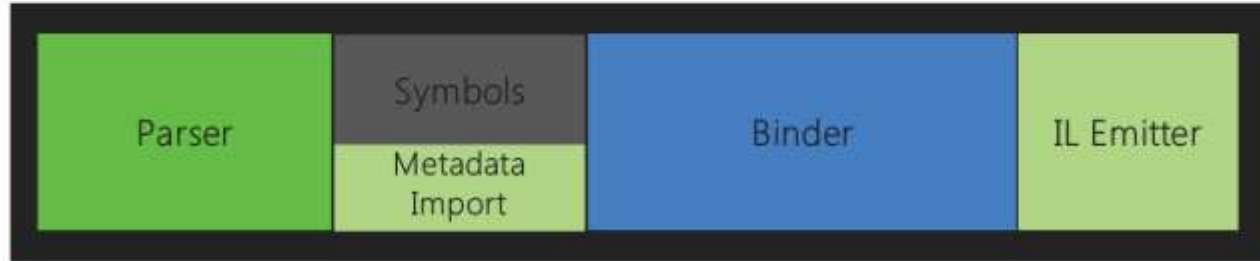- Lazy initialization
- Lock-free

Language Service

Formatter | Colorizer | Outlining | Navigate To | Object Browser | Completion List | Find All References | Rename | QuickInfo | Signature Help | Extract Method | Go To Definition | Edit and Continue

Compiler API

Syntax Tree API | Symbol API | Binding and Flow Analysis APIs | Emit API

Compiler Pipeline

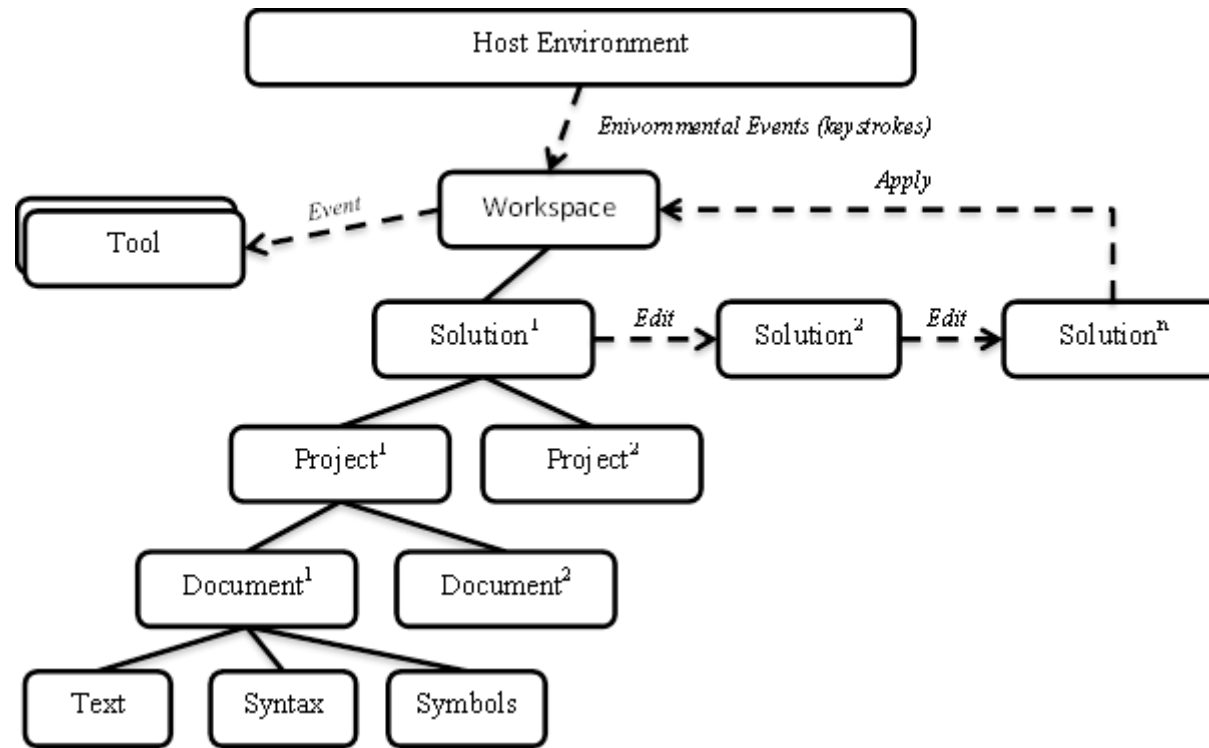Parser | Symbols / Metadata Import | Binder | IL Emitter

Roslyn

- Syntax Tree API
- lexical and syntax information
- Symbol API
- i.e. Find All References
- Binding and analysis API
- i.e. casting, name conflicts
- Emit API

Demo

- https://roslynquoter.azurewebsites.net/
- https://sharplab.io/

- https://source.dot.net/

Workspaces

Demo

Q & A