

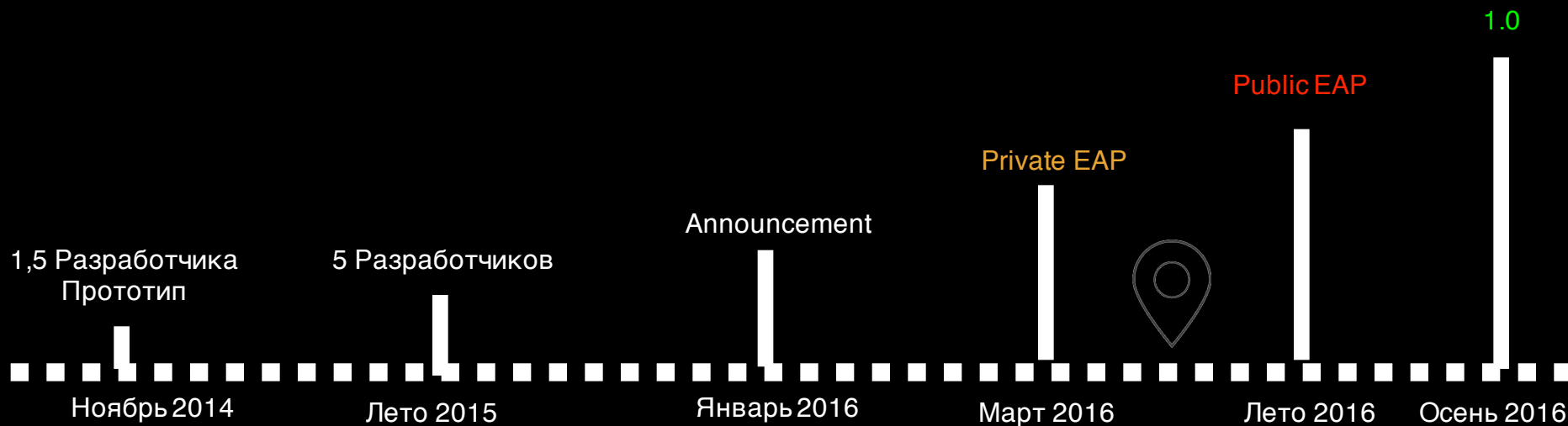


**Rider**

Кирилл Скрыган

«А почему бы вам не  
написать свою IDE?!»

# Roadmap

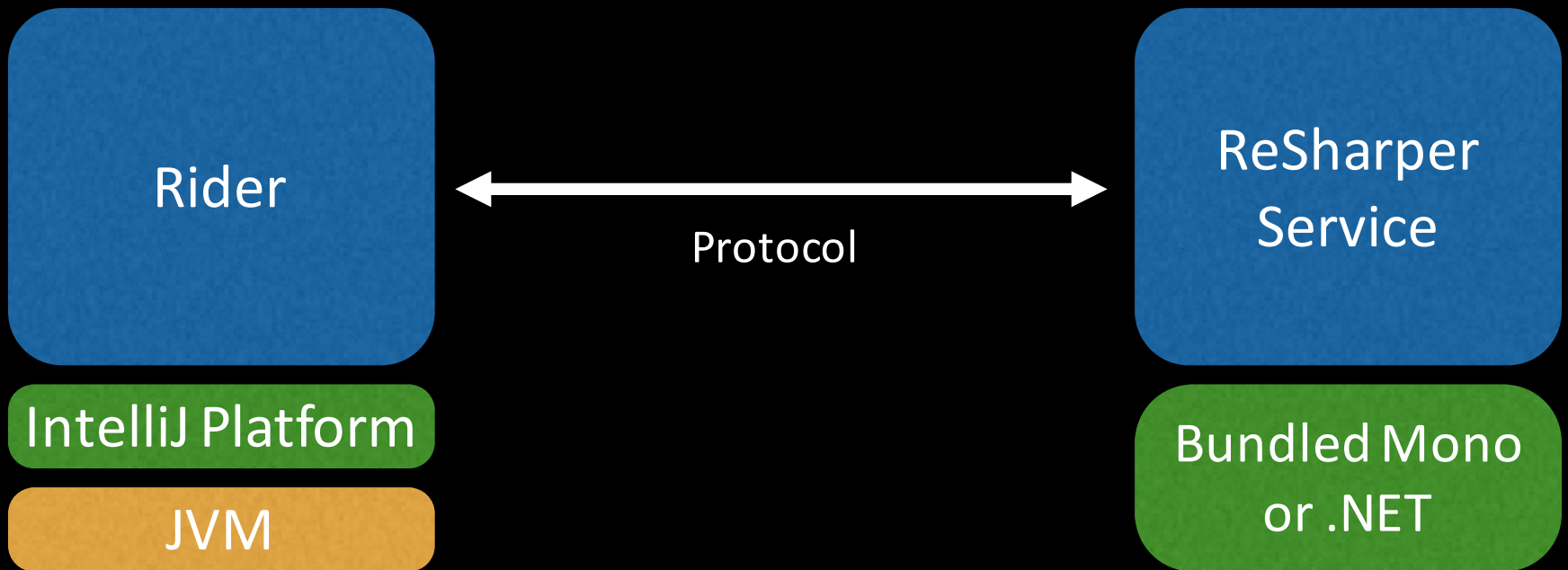


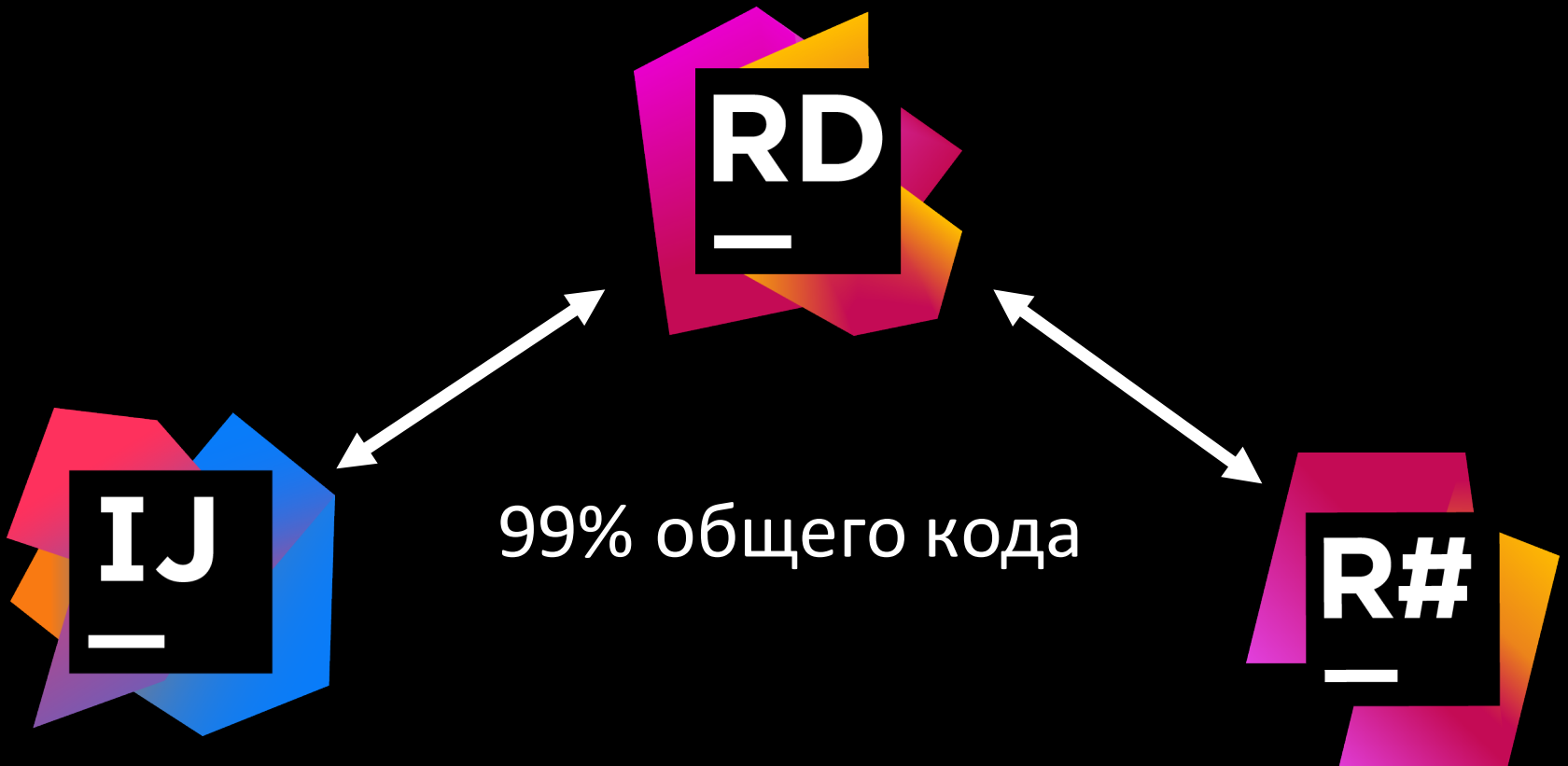
# Rider Features

- 100500 рефакторингов
- 100500 навигаций
- 100500 анализов кода
- И ещё 100500 других фичей

Rider =  + 

# Архитектура





ReSharper

File View Search Code Build Refactor Tools Internal Help

Solution Explorer

PerformanceTestsFramework  
 PerformanceTestsRunner  
 PsiStatistics  
 ReSharper  
 Services  
 References  
 actions  
 codeInsight  
 component  
 editor  
 Gen  
 ide  
 internal  
 lexing  
 parsing  
 progress  
 project  
 psi  
 caches  
 codeStyle  
 completion  
 controlFlow  
 expectedTypes  
 impl  
 interfaces  
 navigation  
 pointers  
 resolve  
 search  
 types  
 utl  
 DeclaredElementInfo.cs  
 DeclaredElementPresente  
 DeclaredElementVisitor.cs  
 DeprecatedElementVisitor  
 IDeclaredElementInfo.cs  
 IElementFactory.cs  
 ILocalScope.cs  
 IModifiersOwner.cs  
 IncorrectOperationExcep

PsiManager.cs X PsiManagerImpl.cs

Types Psi.PsiManager Members Solution

```

public abstract bool IsCommitted (IProjectItem prjItem);

public abstract Psi.Search.IFinder Finder { get; }
public abstract Psi.IElementFactory ElementFactory { get; }
public abstract Predefined Predefined { get; }

public abstract ISolution Solution { get; }
public abstract void CommitAllDocuments ();
public abstract void Commit (IProjectItem prjItem);
public abstract bool IsValid (IFile file);

Summary:
Creates PSI file from the specified text which is connected to the specified project and this manager.
But neither project item nor manager do not know anything about this file. The file can be safely used
for resolving references in it without affecting the PSI state.

Parameters:
project: The text to parse when creating the file
fileText: The text to parse when creating the file

Returns:
New PSI file

public abstract IFile CreateTemporaryFile (IProject project, char[] fileText);

Summary:
Start temporary PSI change (without committing to document Can be nested)

public abstract void StartTemporaryChange ();

```

Find

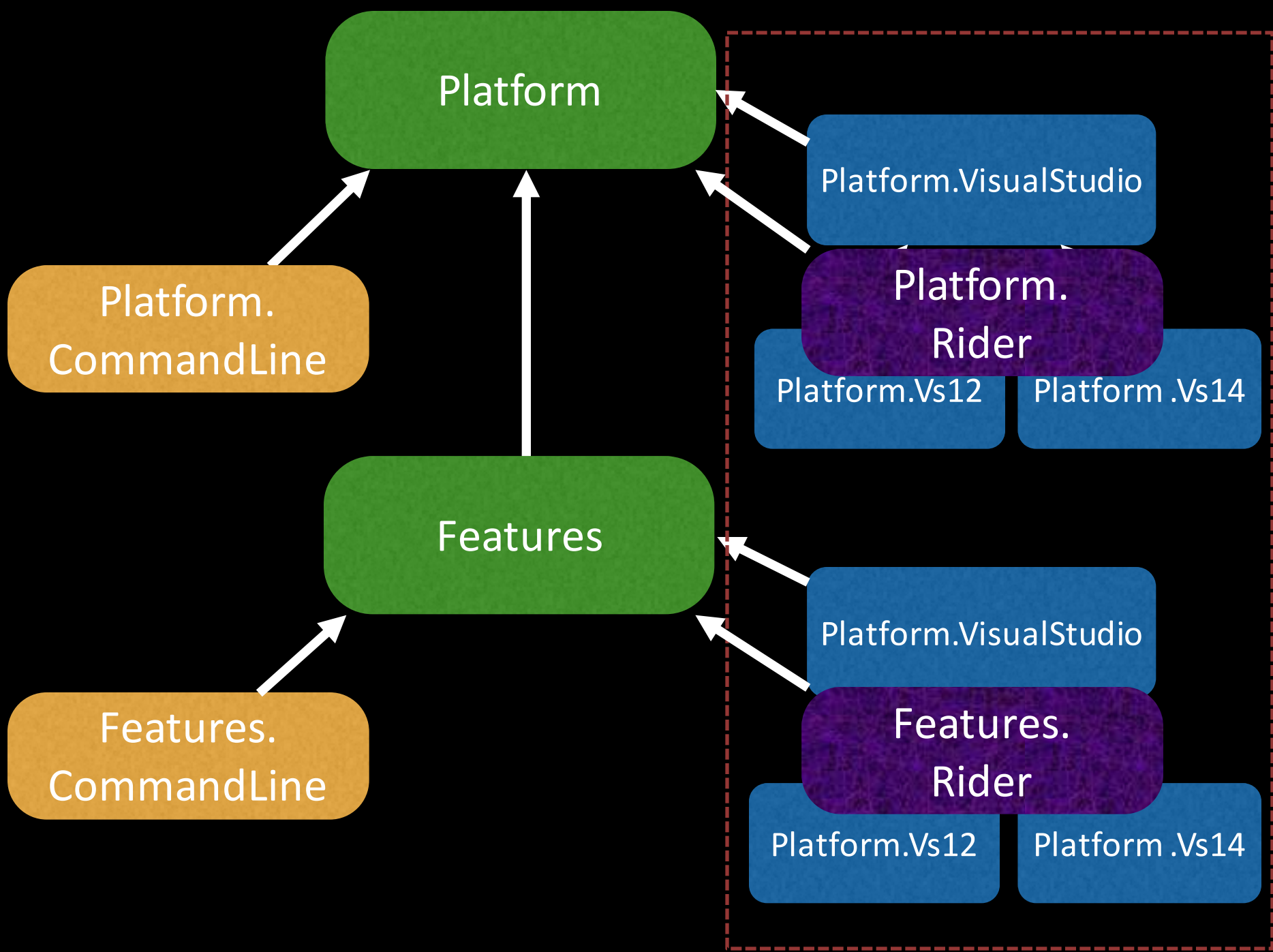
Usages of IsValid

Grouping: None

Targets

- PsiManager.IsValid
- 2 occurrences in 1 files
  - PsiManagerImpl.IsValid
    - (163,54) return myProjectItem.IsValid && GetManager().IsValid (this);





# MONO

- Да, есть баги
- Но оно работает и даже более или менее стабильно
- Проблемы с I/O
- MockWpf в помощь
- XBuild - в нашем случае самое слабое звено

# devenv.exe

## Visual Studio

## ReSharper

Text Buffer



Document

Caret



Caret

Focused Editor



Focused Editor

Рисуем сами 😊

Features UI

# IntelliJ

IntelliJ document

Caret

Focused Editor

Features UI

Rider.exe



# ReSharper

R# document

Caret

Focused Editor

Features UI

ReSharper.Host.exe

# Проблемы производительности IDE

- GC
  - Так как много аллокаций связано с парсингом и резолвом
- Visual Studio
  - STA COM
  - Roslyn
  - Непонятно как сделанный встроенный MSBuild
- Прерываемость
- А что насчёт пересылки по протоколу между двумя процессами?

Fixed

# Frontend

# Backend

Нажали Alt+Enter

ActionId : String

Запустили движок  
Alt+Enter


Посчитали все  
элементы


Приготовили View  
Model


Открыли реальный  
popup


IconId : int  
DisplayName : int  
ItemId : int


(ContextTracker contextTracker)


 Make parameter type IDisposable


 Check parameter for null

 Check if parameter is not valid

 Create overload without parameter

 Can be null (inferred)

 Not null

 Inspection "Parameter can be declared with base type" ▶

ccess());

Host] Sending {0} {1} items to frontend...", log, wrap

rapper.ShortName, wrapper.DispalyName, myIconHost.Trans  
) .ToList();

# Особенности

- IntelliJ, даже без Language Services – это очень большая система, со своей жизнью
- Синхронизовать lifecycle сложного frontend'а с backend'ом оказалось нетривиальной задачей



Frontend

Открывается редактор!



Transient signal



Backend



Solution загрузился, мы готовы  
обрабатывать открытие редакторов



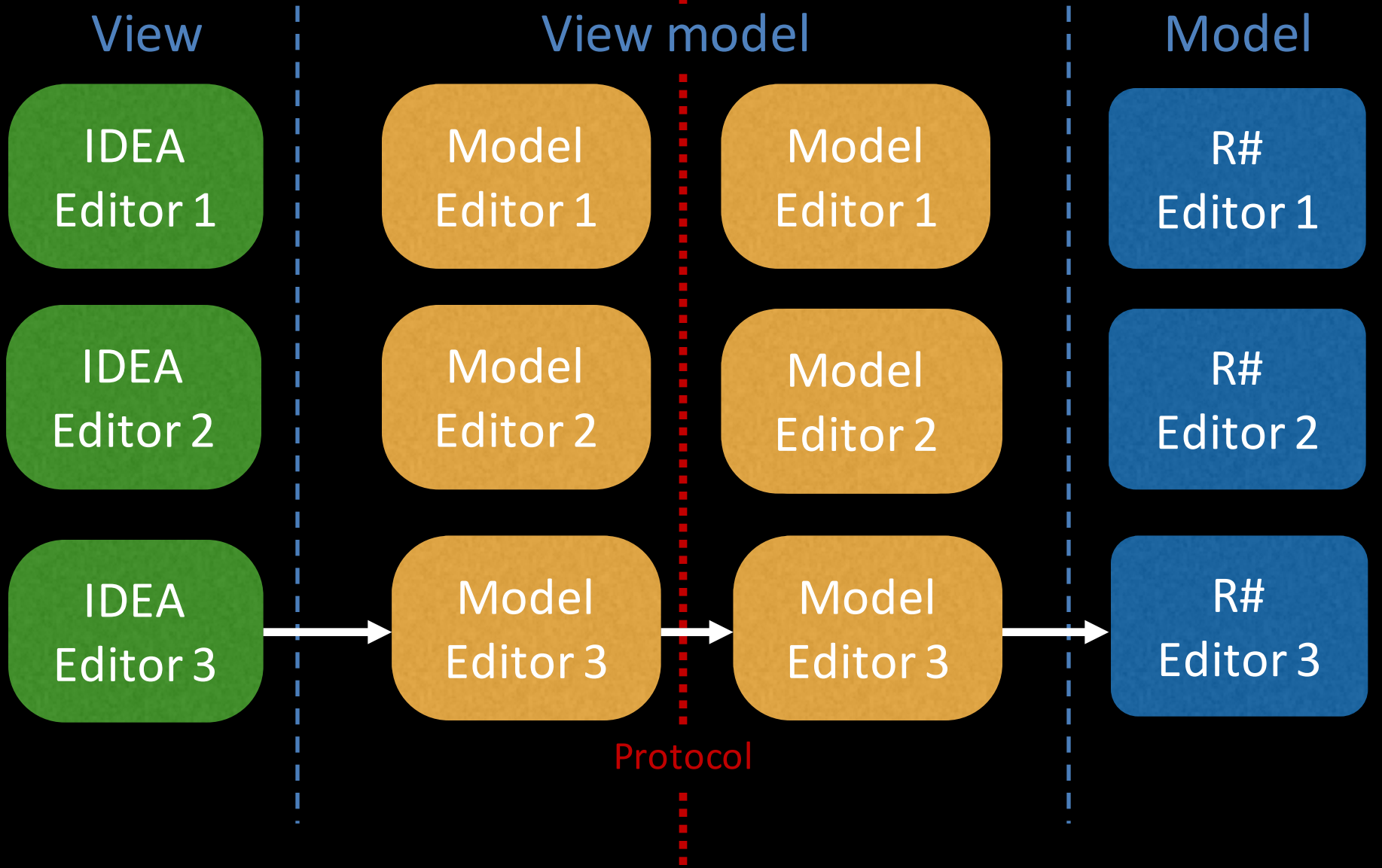
# Решение

- MVVM архитектура
- Реактивный протокол с **состоянием**, синхронизированным по обе стороны провода



# Frontend

# Backend



Frontend

Открывается редактор!



Backend



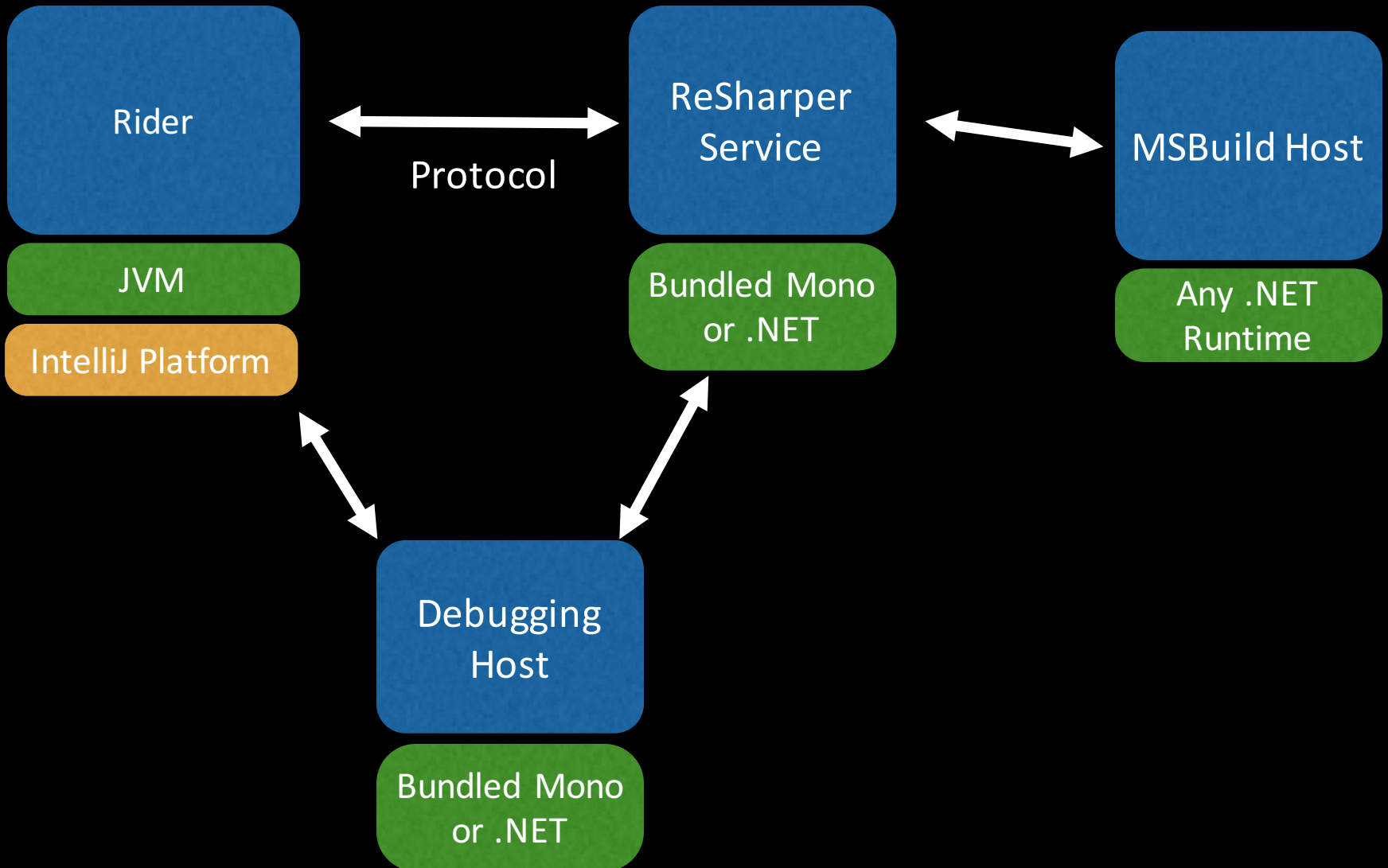
Solution загрузился, мы готовы  
обрабатывать открытие редакторов

Frontend может  
присоединиться к Backend  
в любой момент времени  
и наоборот

# Стоит ли бояться Interprocess взаимодействия?

- Вообще говоря, наш MVVM очень «гибридный» 😊
- Передать 5000 completion items - < 100 ms!

# Архитектура



# Итого

- Мощный бэкэнд
- Фронтенд, почти не зависящий от проблем бэкэнда
- По сути мы переходим к эпохе распределённых IDE



# Web

- JS, TS, CSS, Razor, ...
- На данный момент все поддерживается силами решарпера.
- Думаем над гибридными вариантам: Idea + R#

# Плагины

- Плагины для решарпера будут работать
- Плагины для идеи уже работают
- В планах сделать возможность для кросс протокольных плагинов

```
fun testKeywords() {
  doTestWithDocumentsEnableCompletion({
    withOpenedEditor("ConsoleApplication8/Program.cs", {
      moveCaretRelatively(3, 2)
      pressEnter()
      pressEnter()
      moveUp()

      pressTab()
      typeWithLatency("p")
      waitForCompletion()
      require(lookupItemsCount == 4) { "found $lookupItemsCount items" }

      typeWithLatency("ub")
      wait(100)
      require(lookupItemsCount == 1) { "found $lookupItemsCount items" }

      require(currentLookupItemString.equals("public"))
      completeWithEnter()
      wait(100)
    })
  })
}
```



Program.cs X

```
namespace ConsoleApplication8
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}
```



# Вопросы?

- [twitter.com/kskrygan](https://twitter.com/kskrygan)
- [Kirill.Skrygan@jetbrains.com](mailto:Kirill.Skrygan@jetbrains.com)