

# The Future of IDE

## How software development paradigm is shifted



Dmitry Ivanov

 [korifey\\_ad](#)

October 24, 2020



Page 1

# About speaker

- **Huawei, Russian Research Institute, Principal Expert, Leader of Software Analysis Team** - develops code analysis tools, IDE plugins for static analysis, tooling for DevCloud.
- **ex. Technical leader of JetBrains Rider** IDE for .NET, the most fast-growing JetBrains product. Designed a desktop and cloud IDEs architecture.
- **Speaker of several conferences:** Topics are related to IDE design, concurrency, multithreading, optimizations



# What is IDE?

- **IDE**

- **Integrated** – integrates other tools separated before; each year more tools are inside
- **Development** – initially IDEs were used for only for code, but now it's much more
- **Environment** – not only editor but bunch of tool windows interconnected with each other

**IDE** is a main tool for contemporary developer; single access point for code navigation, reading and writing, project building, running, testing, debugging, profiling, versioning and **all** other activities developer may need



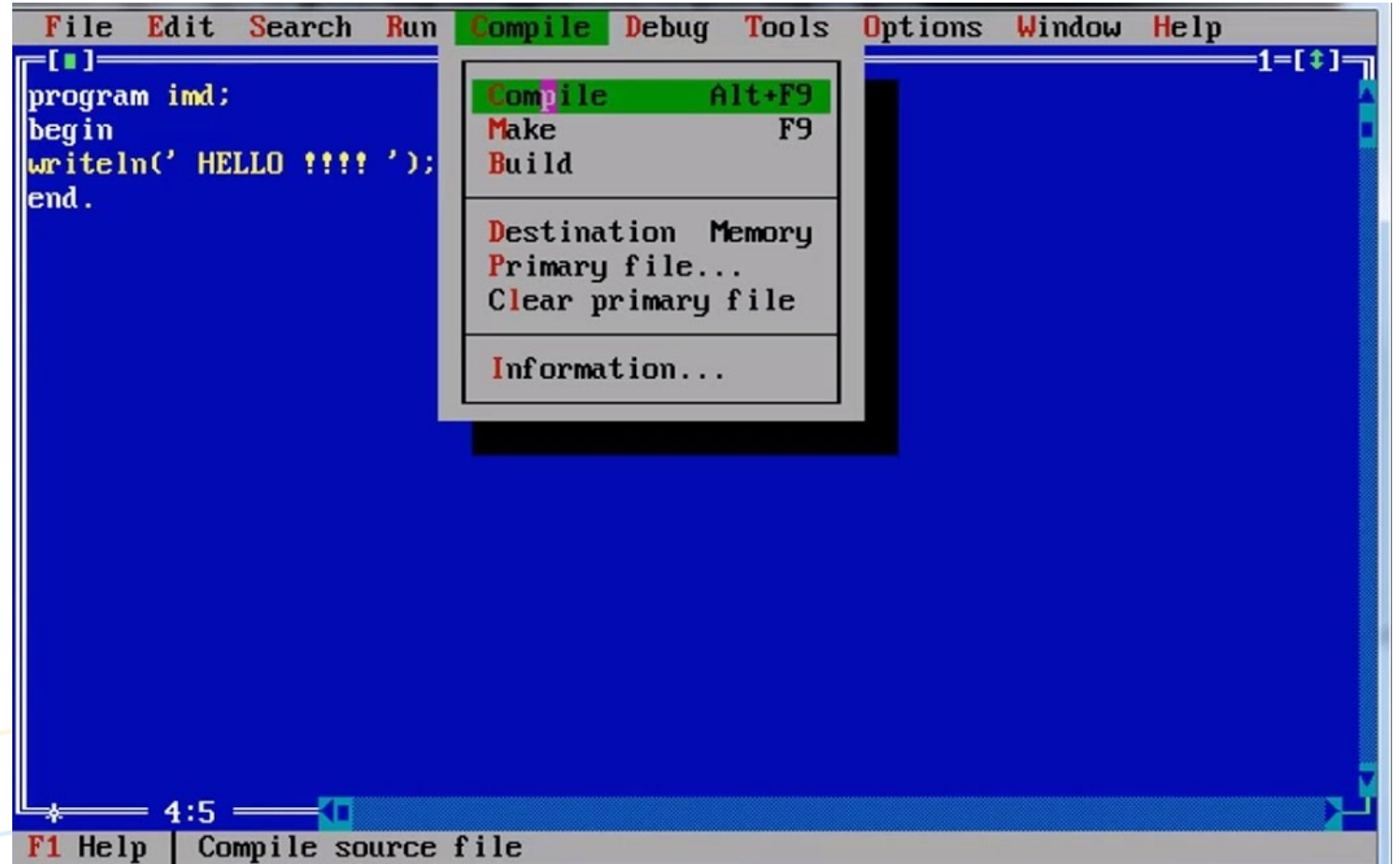
# LIFE DEMO

WORLD WITHOUT IDE



# IDE of 1<sup>st</sup> generation: 1970s-1990s

- **Examples**
  - Turbo Pascal
  - Turbo C/C++
  - Borland Delphi
  
- **Integrate together**
  - Editor
  - Compiler
  - Runner
  - Debugger



# LIFE DEMO

## IDE 1.0: TURBO PASCAL



# IDE of 2<sup>nd</sup> generation : 2000s

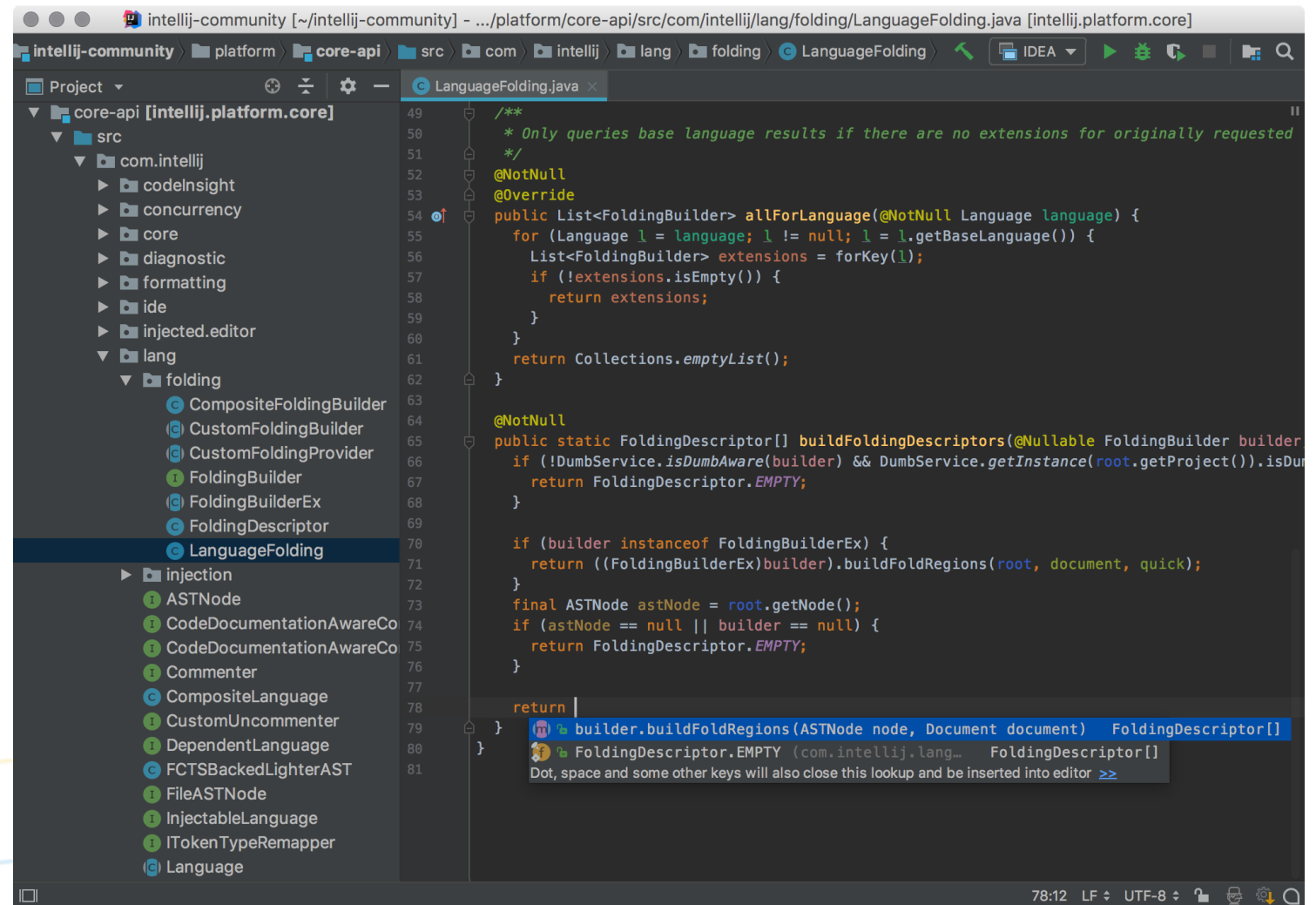
## ■ Examples

- IntelliJ IDEA
- Eclipse

## ■ Features

- Completion
- Error highlighting
- Refactoring
- Quickfix

IDE 2.0 features are based on deep code understanding - language services



```
intellij-community [~/intellij-community] - .../platform/core-api/src/com/intellij/lang/folding/LanguageFolding.java [intellij.platform.core]
intellij-community platform core-api src com intellij lang folding LanguageFolding IDEA
Project core-api [intellij.platform.core]
  src
    com.intellij
      codeinsight
      concurrency
      core
      diagnostic
      formatting
      ide
      injected.editor
      lang
        folding
          CompositeFoldingBuilder
          CustomFoldingBuilder
          CustomFoldingProvider
          FoldingBuilder
          FoldingBuilderEx
          FoldingDescriptor
          LanguageFolding
          injection
            ASTNode
            CodeDocumentationAwareCo
            CodeDocumentationAwareCo
            Commenter
            CompositeLanguage
            CustomUncommenter
            DependentLanguage
            FCTSBackedLighterAST
            FileASTNode
            InjectableLanguage
            ITokenTypeRemapper
            Language
49 /**
50  * Only queries base language results if there are no extensions for originally requested
51  */
52 @NotNull
53 @Override
54 public List<FoldingBuilder> allForLanguage(@NotNull Language language) {
55     for (Language l = language; l != null; l = l.getBaseLanguage()) {
56         List<FoldingBuilder> extensions = forKey(l);
57         if (!extensions.isEmpty()) {
58             return extensions;
59         }
60     }
61     return Collections.emptyList();
62 }
63
64 @NotNull
65 public static FoldingDescriptor[] buildFoldingDescriptors(@Nullable FoldingBuilder builder
66     if (!DumbService.isDumbAware(builder) && DumbService.getInstance(root.getProject()).isDumb
67     return FoldingDescriptor.EMPTY;
68 }
69
70 if (builder instanceof FoldingBuilderEx) {
71     return ((FoldingBuilderEx)builder).buildFoldRegions(root, document, quick);
72 }
73 final ASTNode astNode = root.getNode();
74 if (astNode == null || builder == null) {
75     return FoldingDescriptor.EMPTY;
76 }
77
78 return
79 }
80 }
81
78:12 LF UTF-8
```



# LIFE DEMO

IDE 2.0: INTELLIJ IDEA

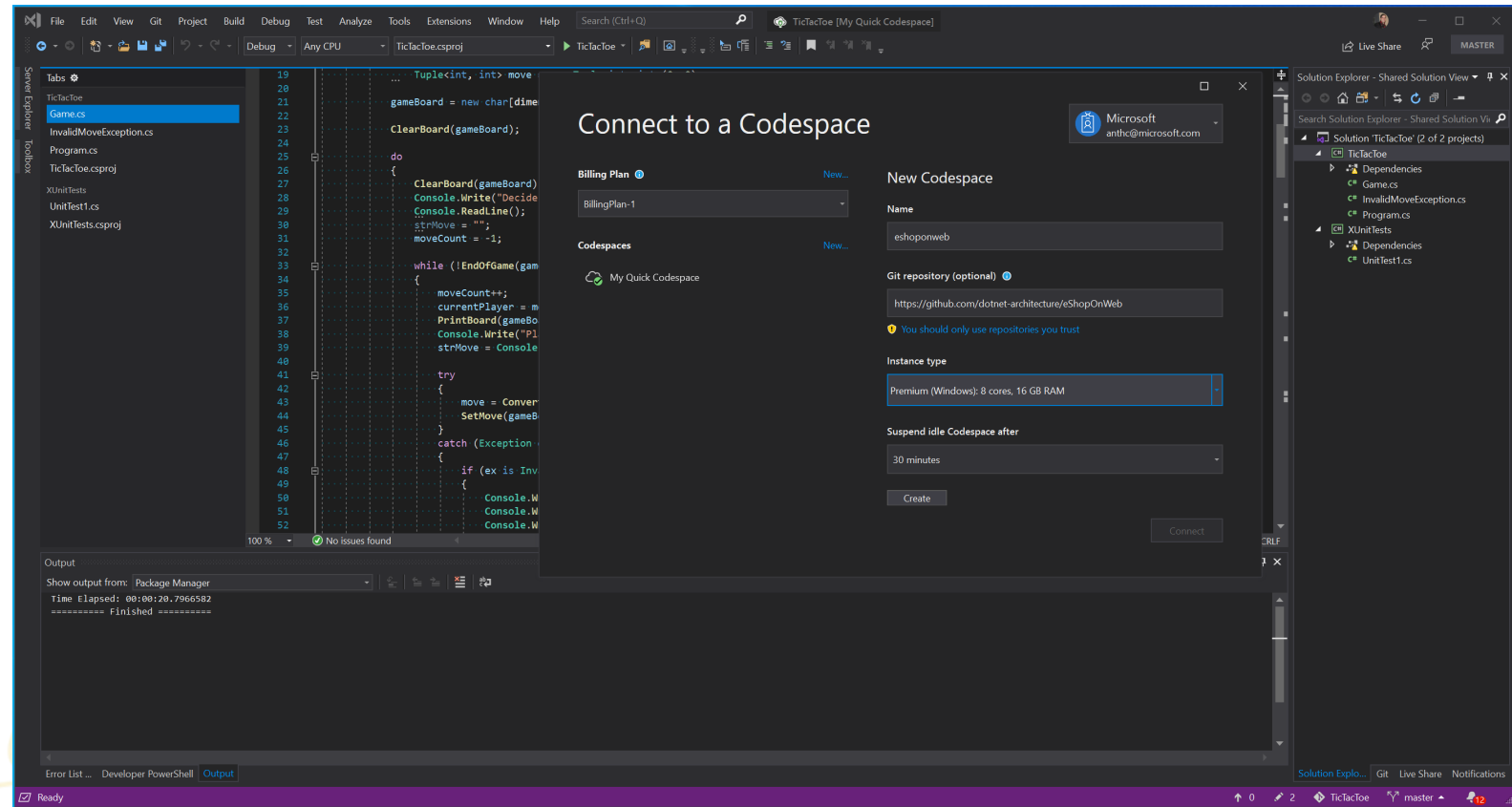




# IDE 2.5: 2010s

**Lightweight IDEs** – separate editor and language server

- **Vim**
- **Emacs**
- **Sublime**
  
- **Visual Studio Code**
  - **LSP** - Language server protocol
  - **Browser/Desktop** - Visual Studio Codespaces
  - **Collaborative editing** – Live Share



# LIFE DEMO



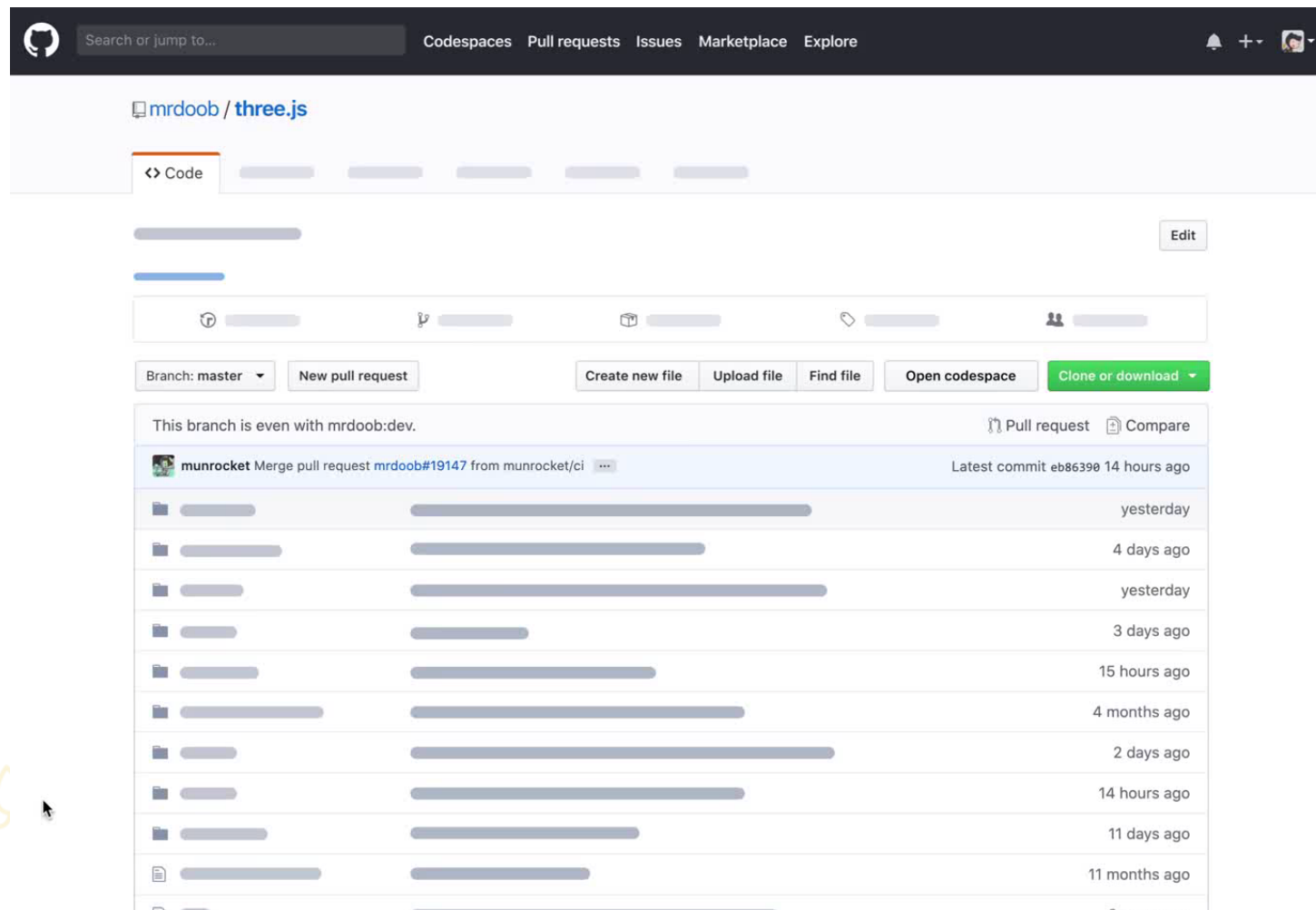
## IDE 2.5: Visual Studio Code



# Current state of the art: Cloud IDE

## Team collaboration tools

- Github (+ Codespaces)
- Gitlab
- JetBrains Space
  
- Cloud IDE
  - AWS Cloud 9
  - Eclipse Theia



# LIFE DEMO

Cloud IDE: AWS Cloud 9



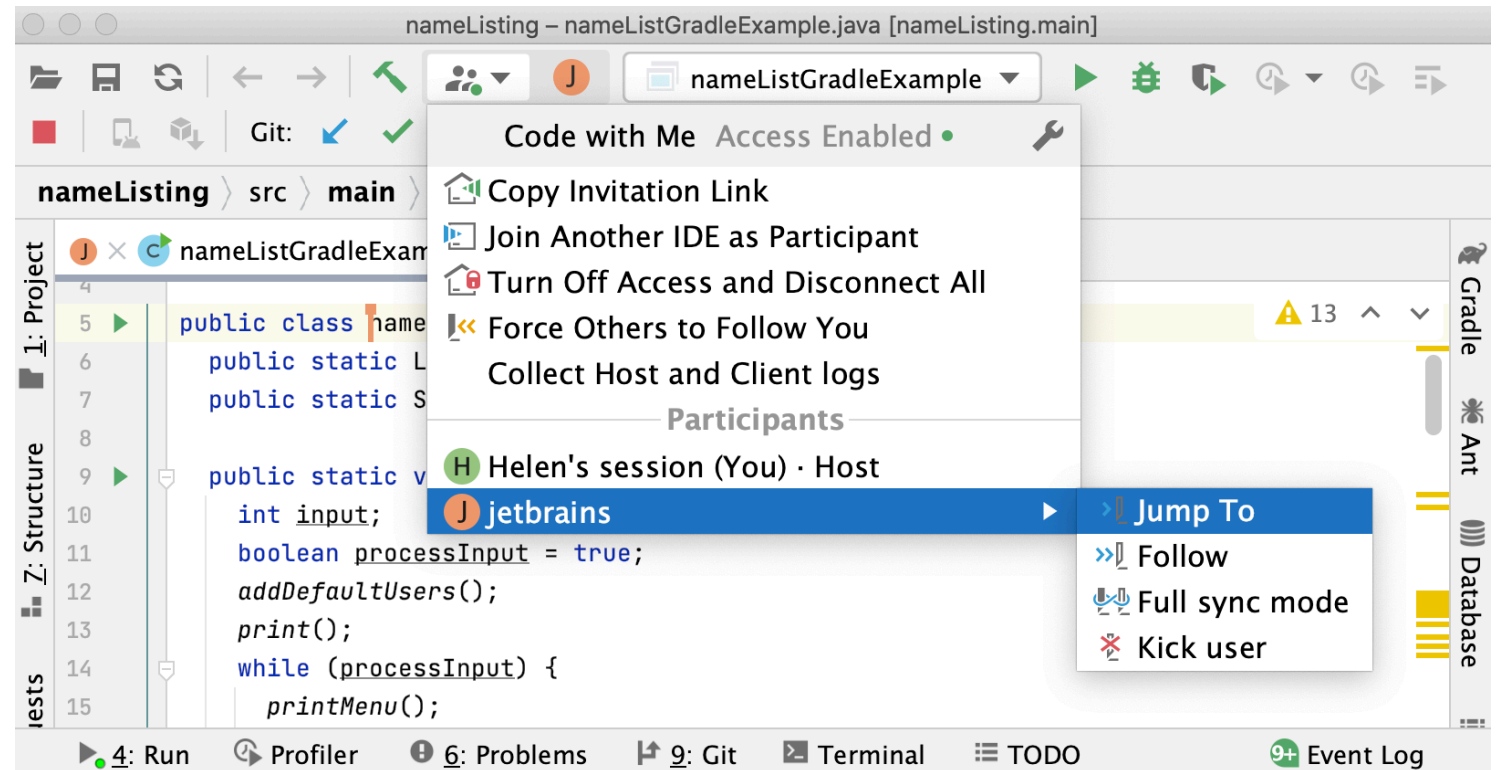
# Hybrid: Cloud + IDE 2.0

## JetBrains CodeWithMe

- User send link for collaborator to join
- Collaborator open link, downloads IntelliJClient and start working on same project

## ■ Possible integration with DevCloud

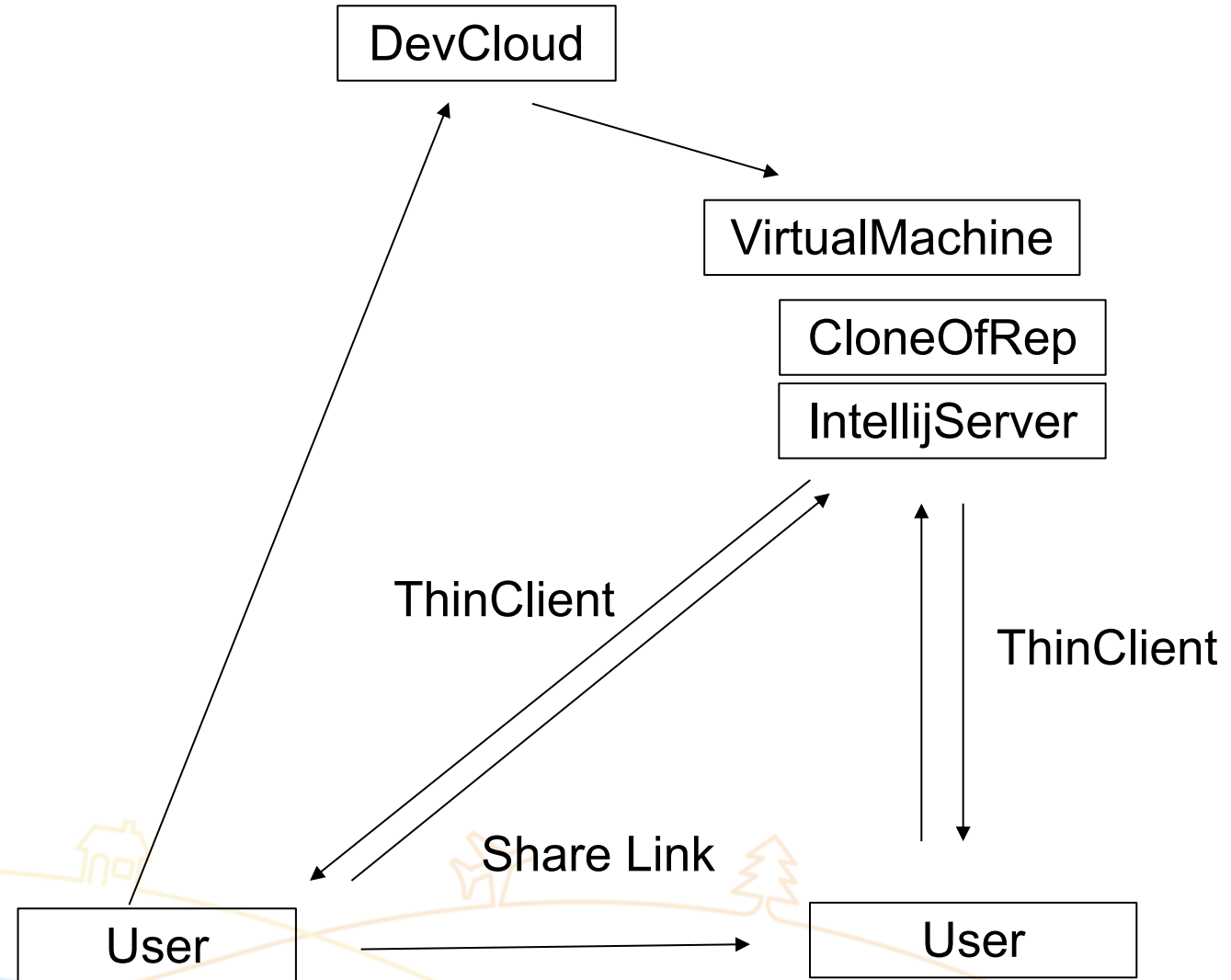
- Button (Open codespace) in DevCloud that creates cloud virtual machine
- IntelliJ Server is started on this VM
- Everyone who have link can connect (multiple users)
- Full power of JetBrains IDE is available in Cloud



# Hybrid: Cloud + IDE 2.0

## JetBrains CodeWithMe

- User send link for collaborator to join
- Collaborator open link, downloads IntelliJClient and start working on same project
- **Possible integration with DevCloud**
  - Button (Open codespace) in DevCloud that creates cloud virtual machine
  - IntelliJ Server is started on this VM
  - Everyone who have link can connect (multiple users)
  - Full power of JetBrains IDE is available in Cloud



DEMO



# Hybrid IDE: JetBrains CodeWithMe



# What is IDE of 3<sup>rd</sup> generation ?



## ■ What is 5<sup>th</sup> generation jet fighter?

- All aspect stealth – even when armed (LPIR)
- High maneuverability
- Advanced avionics
- Multirole capabilities
- **Highly integrated** computer systems capable of **networking with other** elements within **battlespace**





# What is IDE 3.0 ?

- **Distribution: Separated client(s) and server(s)** – each agent can work either locally / in intranet / or in cloud
- **Safety - Can continue local work without internet connection** – almost all necessary info is downloaded lazily to local machine as Git version control system does.
- **Isomorphic experience** – where code is running, building, debugging
- **Zero setup time** – between button click and start working
- **Team collaboration tools integrated** - Tracker, Code review, Code history browsing
- **Easy adoption to different clouds**
- **Responsive, low latency editor that works on any device:** Desktop/Browser/Tablet/Phone
- **Open protocol for 3<sup>rd</sup> party developers to write extensions easily** – LSP should be supported by default but must be enhanced in future
- **Code quality** - using of long-running linters, static analysis, deep code analysis, code metrics
- **ML adoption** – every heuristics can be improved by ML



# Questions and Answers

Dmitry Ivanov

 korifey\_ad

korifey@gmail.com

