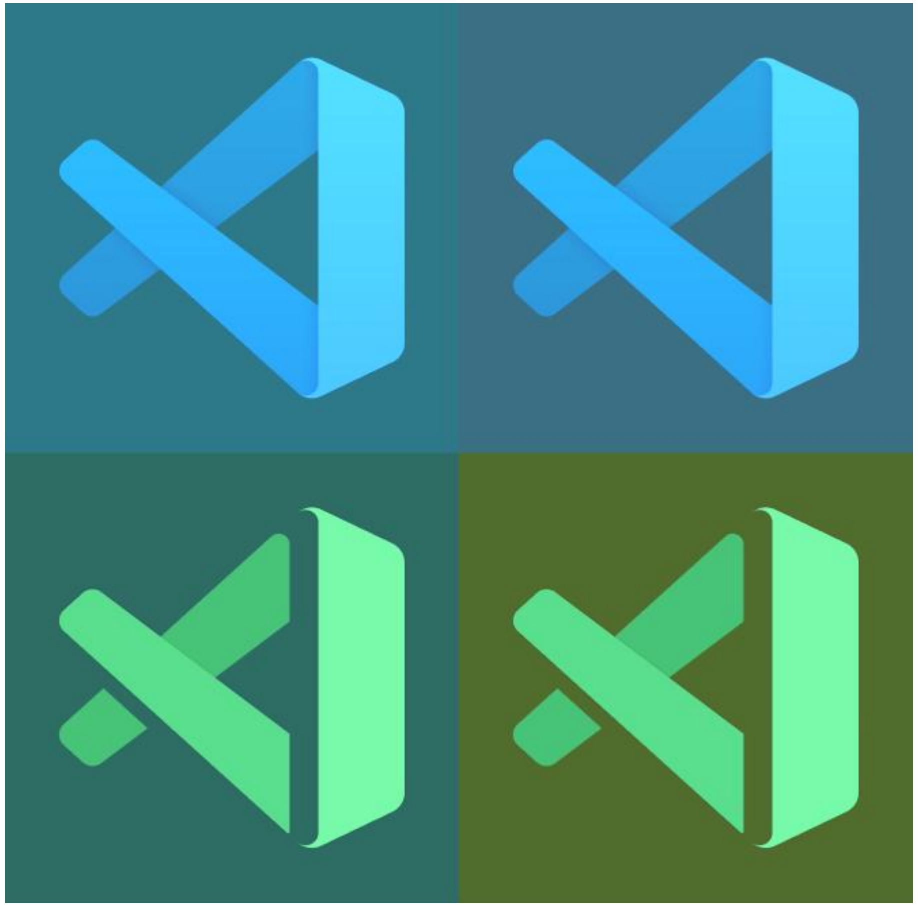
Visual Studio Code 基础架构与源码构建

魏兆华
723889427@qq.com

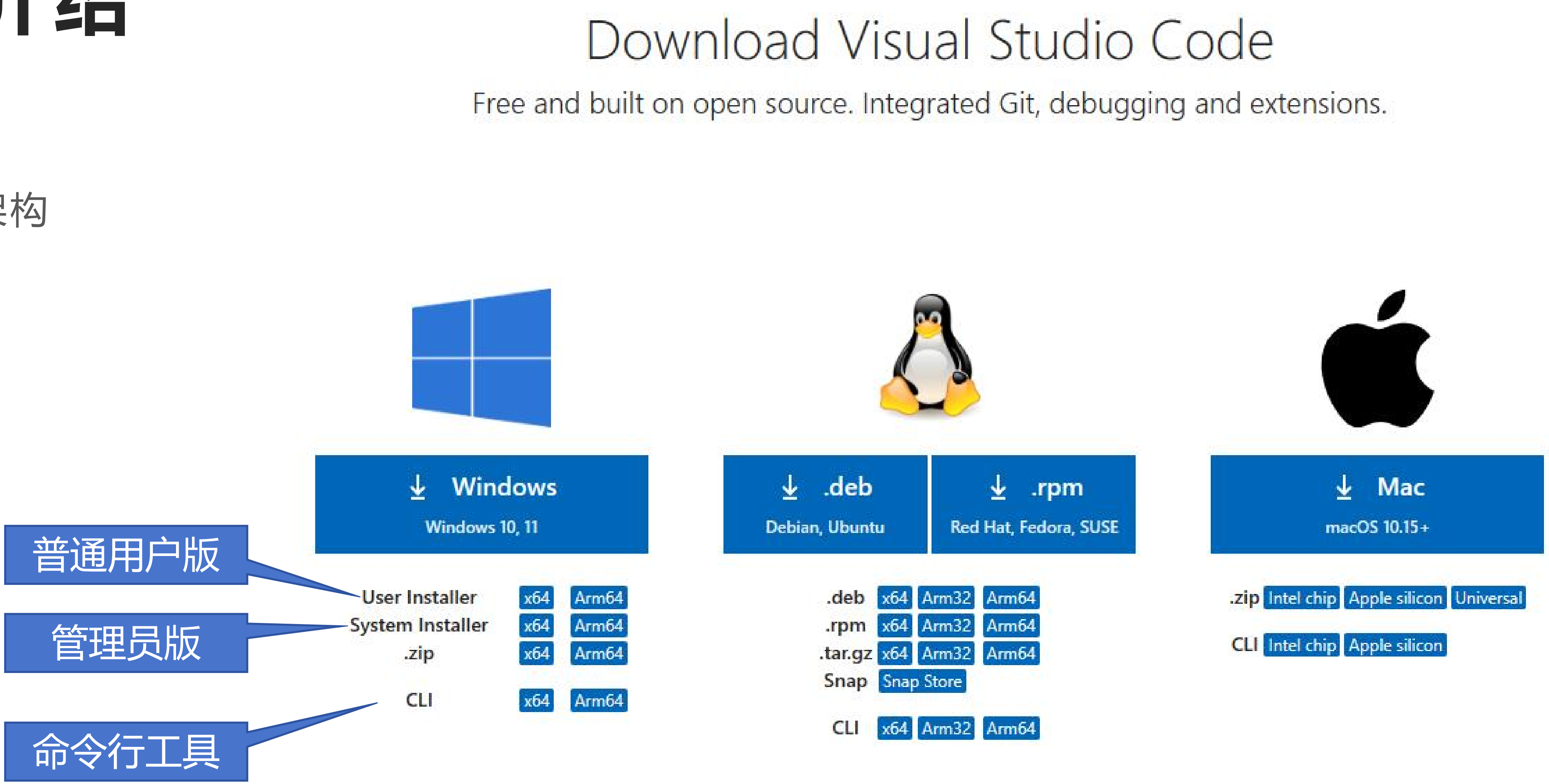
主要内容

- 版本介绍
- 源码结构
- 体系架构
- 源码构建



版本介绍

- 平台
- 处理器架构
- 安装包



Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

Windows
Windows 10, 11

User Installer	x64	Arm64
System Installer	x64	Arm64
.zip	x64	Arm64
CLI	x64	Arm64

.deb
Debian, Ubuntu

.rpm
Red Hat, Fedora, SUSE

.deb	x64	Arm32	Arm64
.rpm	x64	Arm32	Arm64
.tar.gz	x64	Arm32	Arm64
Snap	Snap Store		
CLI	x64	Arm32	Arm64

Mac
macOS 10.15+

.zip	Intel chip	Apple silicon	Universal
CLI	Intel chip	Apple silicon	

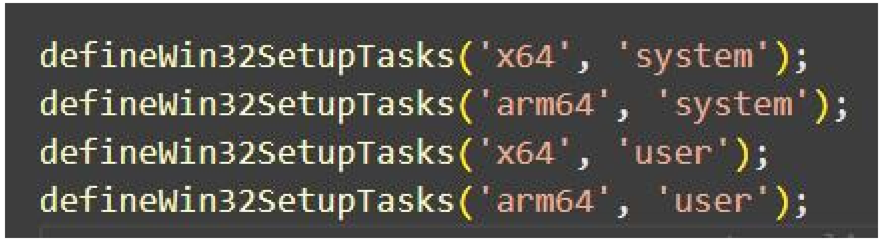
普通用户版

管理员版

命令行工具

版本介绍

- Windows 用户版与管理员版的区别

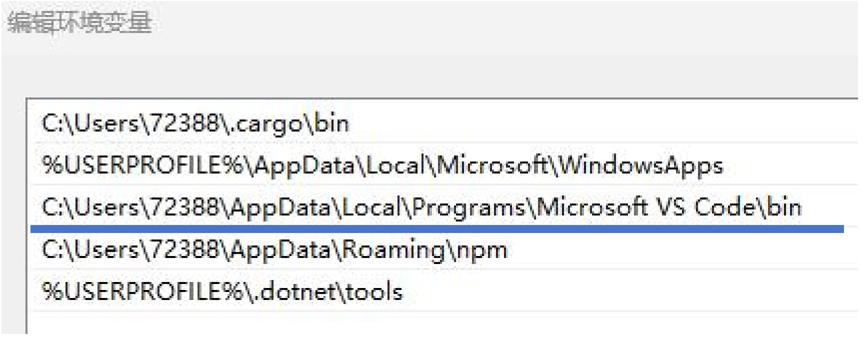


```
defineWin32SetupTasks('x64', 'system');
defineWin32SetupTasks('arm64', 'system');
defineWin32SetupTasks('x64', 'user');
defineWin32SetupTasks('arm64', 'user');
```

	user 普通用户	system 管理员
ProductID	win32x64UserAppId win32arm64UserAppId	win32x64AppId win32arm64AppId
DefaultDirName	{userpf}/{#DirName}	{pf}/{#DirName}
PrivilegesRequired	lowest	
SoftwareClassesRootKey EnvironmentRootKey	HKCU	HKLM
Permissions	Users Authenticated Users 右侧管理员权限	Local System Administrators Owners

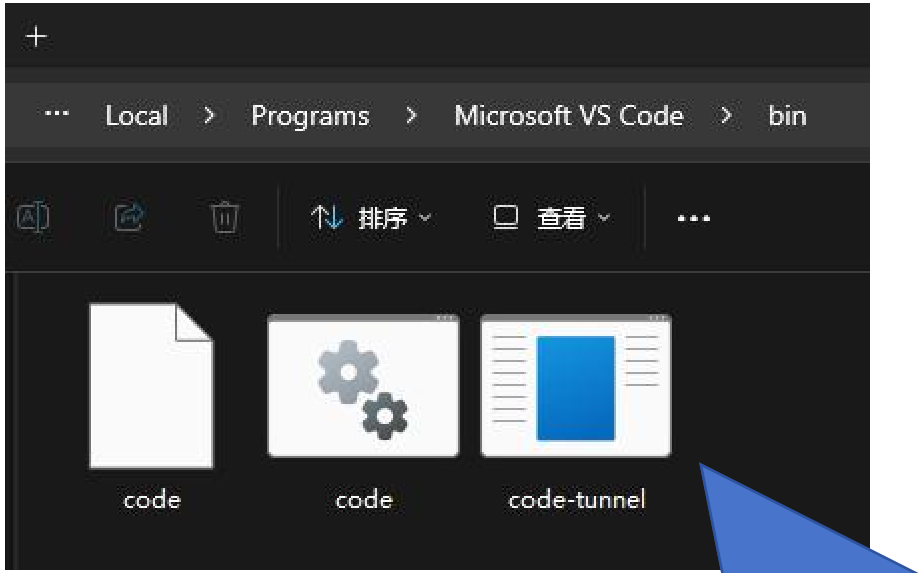
版本介绍

- CLI：Command Line Interface



编辑环境变量

```
C:\Users\72388\.cargo\bin
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\Users\72388\AppData\Local\Programs\Microsoft VS Code\bin
C:\Users\72388\AppData\Roaming\npm
%USERPROFILE%\.dotnet\tools
```



单独下载的 CLI 包里，名称为 code.exe

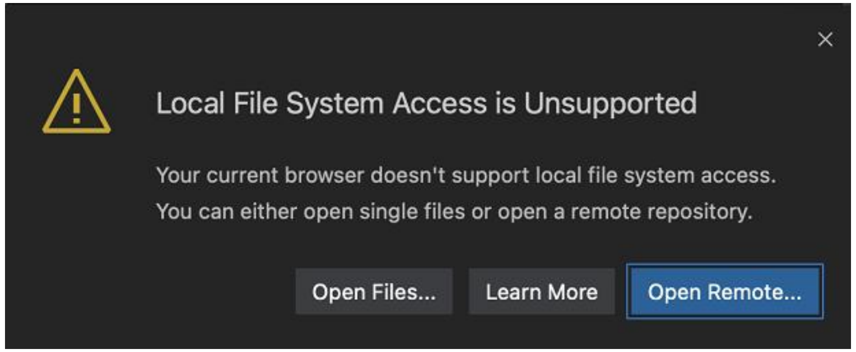
- Code.exe
- bin/code[.sh]
- bin/code.cmd
- bin/code-tunnel.exe

Q：从命令行启动时真正的入口是哪一个？

版本介绍

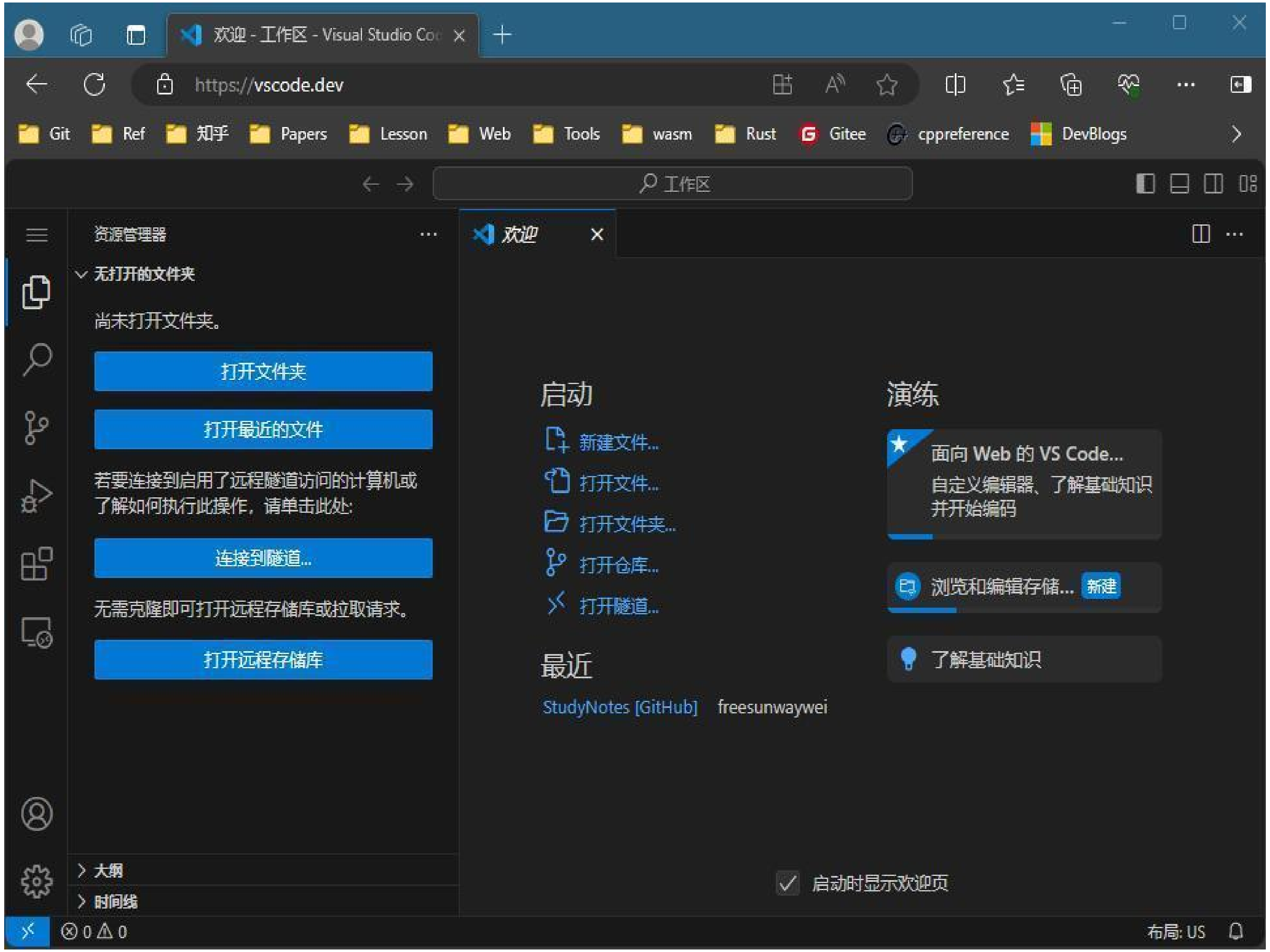
https://vscode.dev

- 本地文件/目录/仓库
- 远程仓库
- Tunnels



Local File System Access is Unsupported

Your current browser doesn't support local file system access.
You can either open single files or open a remote repository.



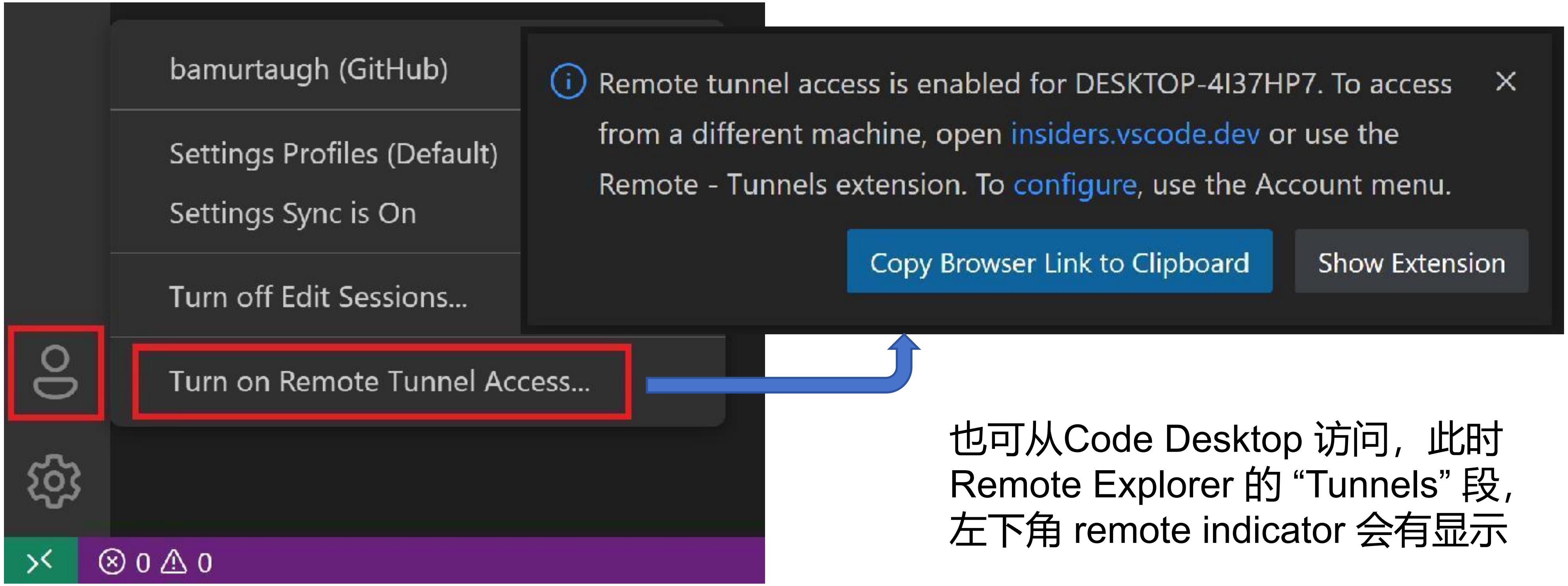
尚未打开文件夹。

若要连接到启用了远程隧道访问的计算机或了解如何执行此操作，请单击此处:

无需克隆即可打开远程存储库或拉取请求。

版本介绍

- 命令行启动 code tunnel [args]
- https://vscode.dev/tunnel/<machine_name>/<folder_name>
- 界面



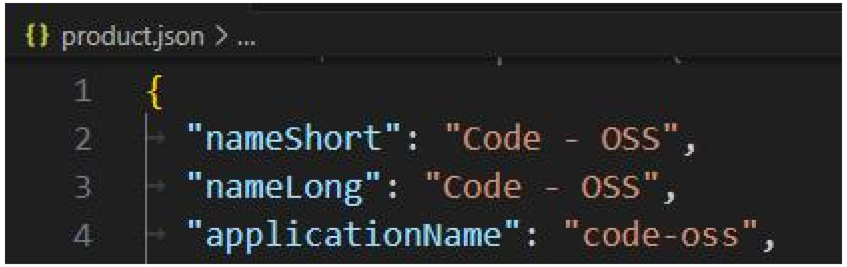
bamurtaugh (GitHub)
Settings Profiles (Default)
Settings Sync is On
Turn off Edit Sessions...
Turn on Remote Tunnel Access...
Copy Browser Link to Clipboard
Show Extension
⊗ 0 ⚠ 0

Remote tunnel access is enabled for DESKTOP-4I37HP7. To access from a different machine, open insiders.vscode.dev or use the Remote - Tunnels extension. To configure, use the Account menu.

也可从Code Desktop 访问，此时 Remote Explorer 的 “Tunnels” 段，左下角 remote indicator 会有显示

版本介绍

VS Code OSS vs. VS Code Distro



{} product.json > ...

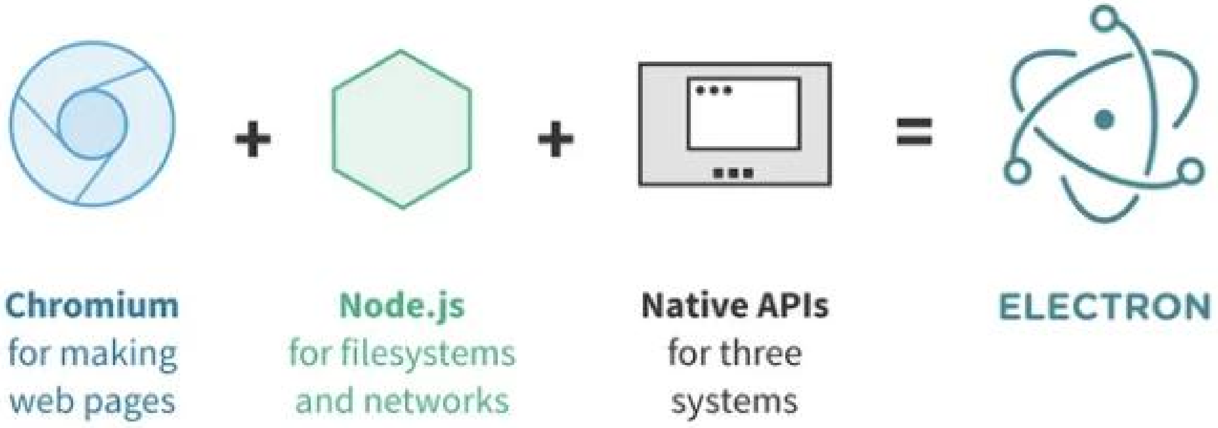
```
{
	"nameShort": "Code - OSS",
	"nameLong": "Code - OSS",
	"applicationName": "code-oss",
```

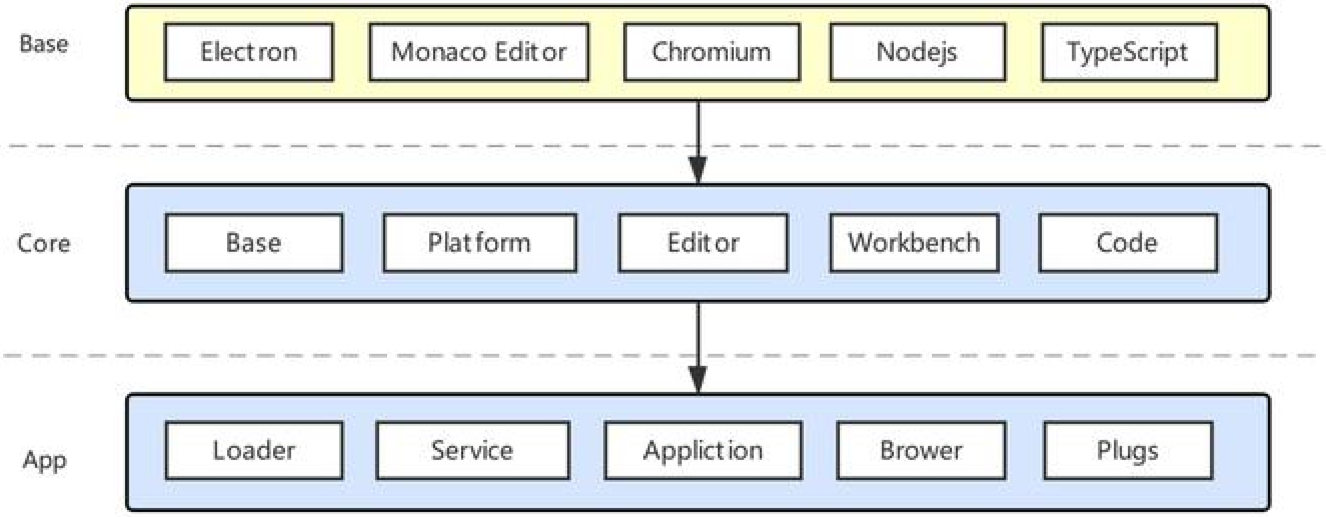
VS Code **OSS**	VS Code **Distro**
Everybody can build from github repo	Builds on VS Code
Not branded	MS branded (logo, telemetry, gallery)
MIT License	MS Standard License
Supported by Community	Supported by MS

源码结构

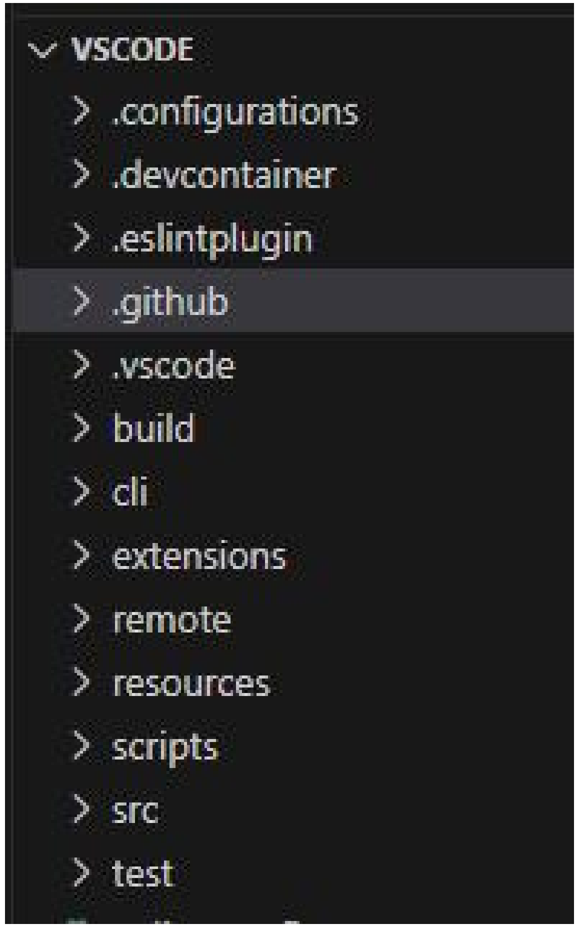
基础知识

- Web/WebAPI
- Javascript
- TypeScript
- NodeJS
- Electron
- [V8、C++]
- [Rust]



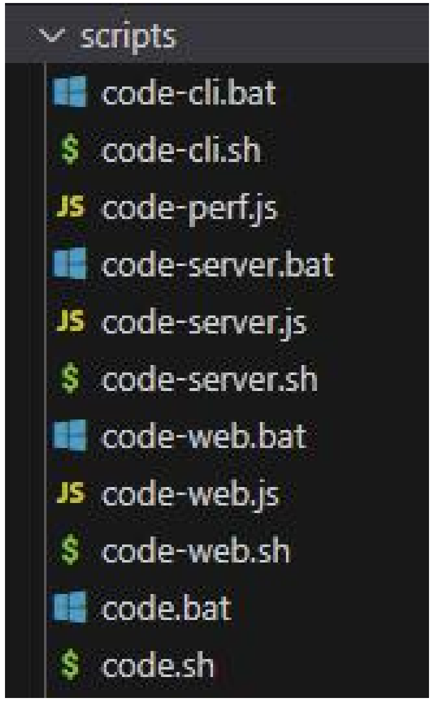


源码结构



```
VSCODE
  .configurations
  .devcontainer
  .eslintplugin
  .github
  .vscode
  build
  cli
  extensions
  remote
  resources
  scripts
  src
  test
```

- /.github：代码仓库工作流
- /.vscode：配置
- /build：构建代码及资源
- /cli：code-tunnel 源码，用 Rust 写成
- /extensions：内置扩展
- /remote：构建 reh 配置
- /resources：构建资源
- /scripts：测试/调试脚本
- /src：源码
- /test：测试

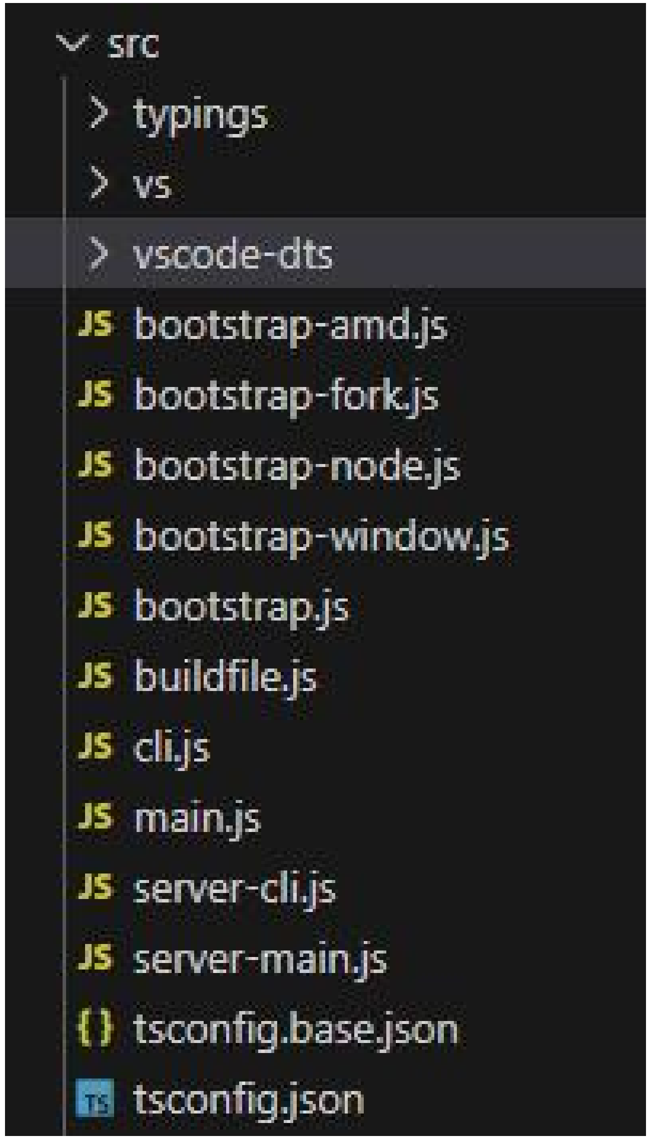


```
scripts
  code-cli.bat
  code-cli.sh
  code-perf.js
  code-server.bat
  code-server.js
  code-server.sh
  code-web.bat
  code-web.js
  code-web.sh
  code.bat
  code.sh
```

源码结构

/src 目录

- 引导文件：bootstrap 系列
- 入口文件：cli main server-cli server-main
- 配置文件：tsconfig 系列
- 协助构建：buildfile
    - base Desktop Web entrypoint ...
- /typings：Typescript 类型定义
- /vs：主目录
- /vscode-dts：扩展开发接口定义
    - vscode.d.ts vscode.proposed.*.d.ts

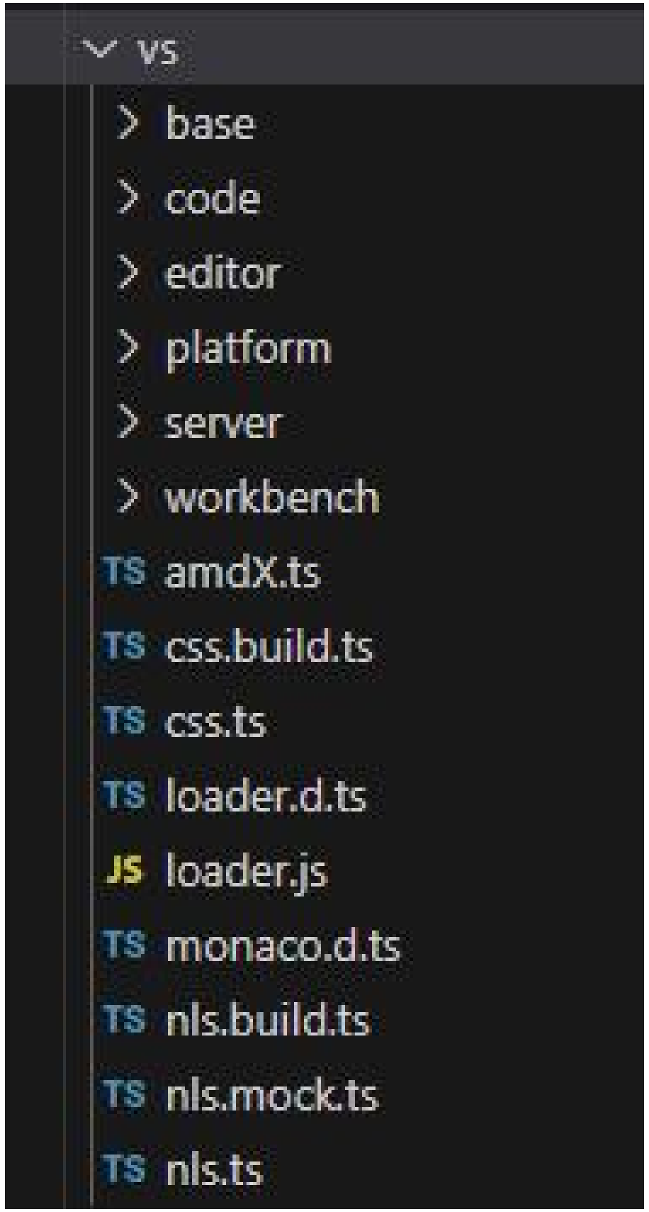


```
⌄ src
  > typings
  > vs
  > vscode-dts
  JS bootstrap-amd.js
  JS bootstrap-fork.js
  JS bootstrap-node.js
  JS bootstrap-window.js
  JS bootstrap.js
  JS buildfile.js
  JS cli.js
  JS main.js
  JS server-cli.js
  JS server-main.js
  {} tsconfig.base.json
  TS tsconfig.json
```

源码结构

/src/vs 目录

- 加载器：loader.js loader.d.ts
  - https://github.com/microsoft/vscode-loader
- css/nls 系列
  - CSS及语言包支持，区分构建/运行时
- /base：基础组件
- /code：主要入口
- /editor：编辑器
- /platform：平台支撑
- /server：服务端
- /workbench：工作台



```
∨ vs
  > base
  > code
  > editor
  > platform
  > server
  > workbench
  TS amdX.ts
  TS css.build.ts
  TS css.ts
  TS loader.d.ts
  JS loader.js
  TS monaco.d.ts
  TS nls.build.ts
  TS nls.mock.ts
  TS nls.ts
```

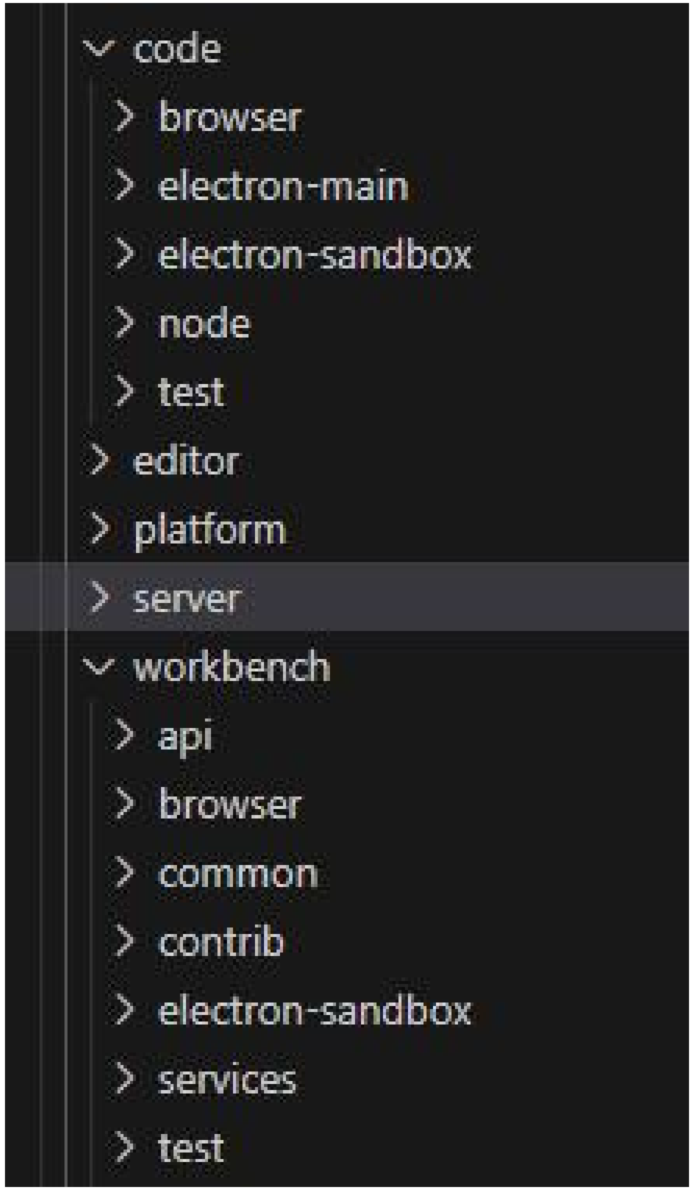
源码结构

环境区分

● common	公用代码
● browser	浏览器环境
● node	NodeJS 环境
● electron-sandbox	Electron 渲染器沙盒
● electron-main	Electron 主进程

功能性支持

- platform 侧重平台
- services 侧重工作台
- contrib 贡献点



```
⌄ code
  > browser
  > electron-main
  > electron-sandbox
  > node
  > test
> editor
> platform
> server
⌄ workbench
  > api
  > browser
  > common
  > contrib
  > electron-sandbox
  > services
  > test
```

源码结构

进程入口（for loader/bootstrap）

server/node/server.main 服务端
server/node/server.cli
浏览器
workbench/workbench.web.main

桌面版专用

code/electron-main/main
code/node/cli
code/node/cliProcessMain
code/node/sharedProcess/sharedProcessMain
code/electron-sandbox/issue/issueReporterMain
code/electron-sandbox/processExplorer/processExplorerMain

workbench/workbench.desktop.main
workbench/contrib/debug/node/telemetryApp

公用进程

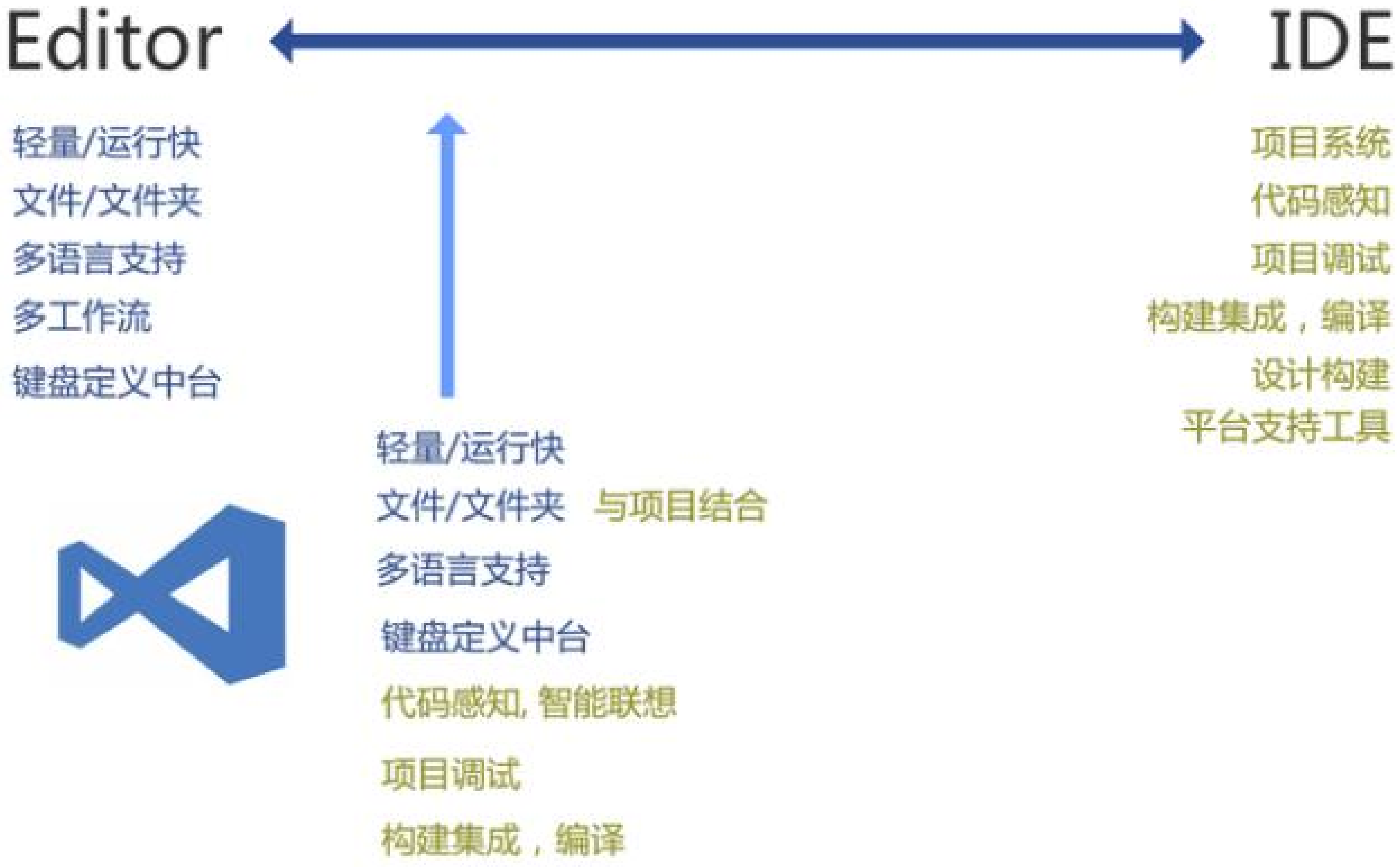
platform/files/node/watcher/watcherMain
platform/terminal/node/ptyHostMain
workbench/api/node/extensionHostProcess

公用Worker

editor/common/services/editorSimpleWorker
base/common/worker/simpleWorker
workbench/api/worker/extensionHostWorker
workbench/contrib/notebook/common/services/notebookSimpleWorker

体系架构

产品定位



Editor

轻量/运行快

文件/文件夹

多语言支持

多工作流

键盘定义中台

IDE

项目系统

代码感知

项目调试

构建集成，编译

设计构建

平台支持工具

轻量/运行快

文件/文件夹 与项目结合

多语言支持

键盘定义中台

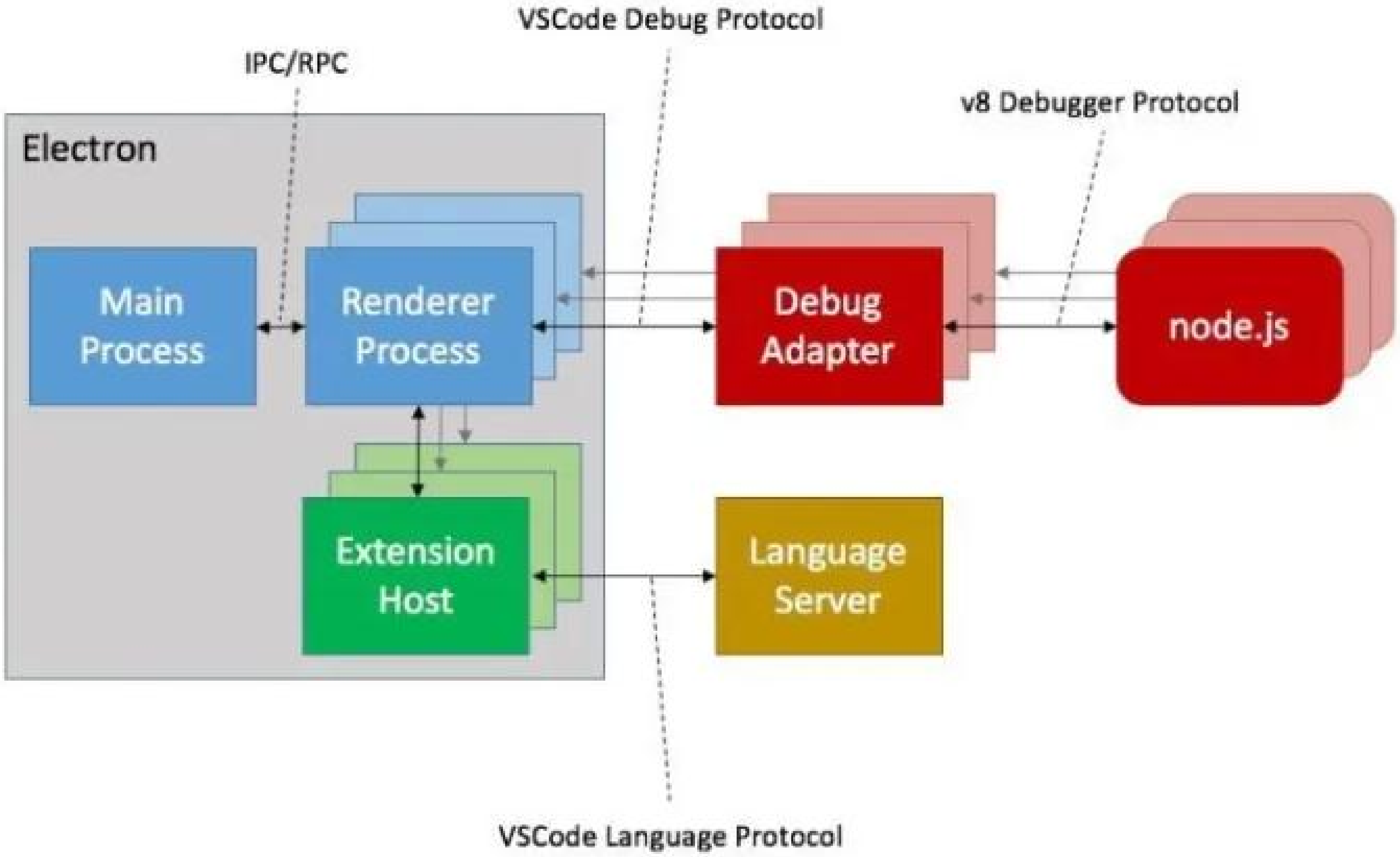
代码感知, 智能联想

项目调试

构建集成，编译

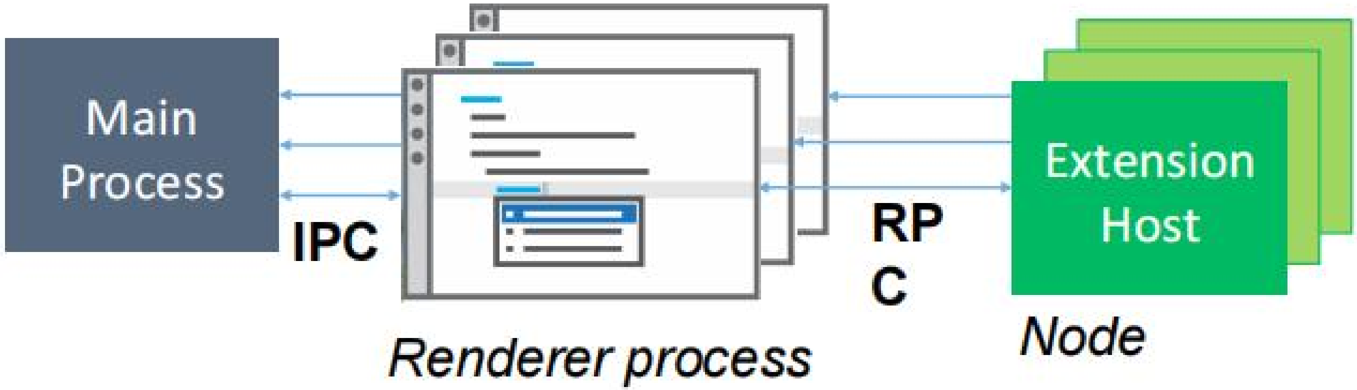
体系架构

以本地开发为例



体系架构

Extension isolation



Controlled extensibility

JavaScript or TypeScript

Reuse node modules

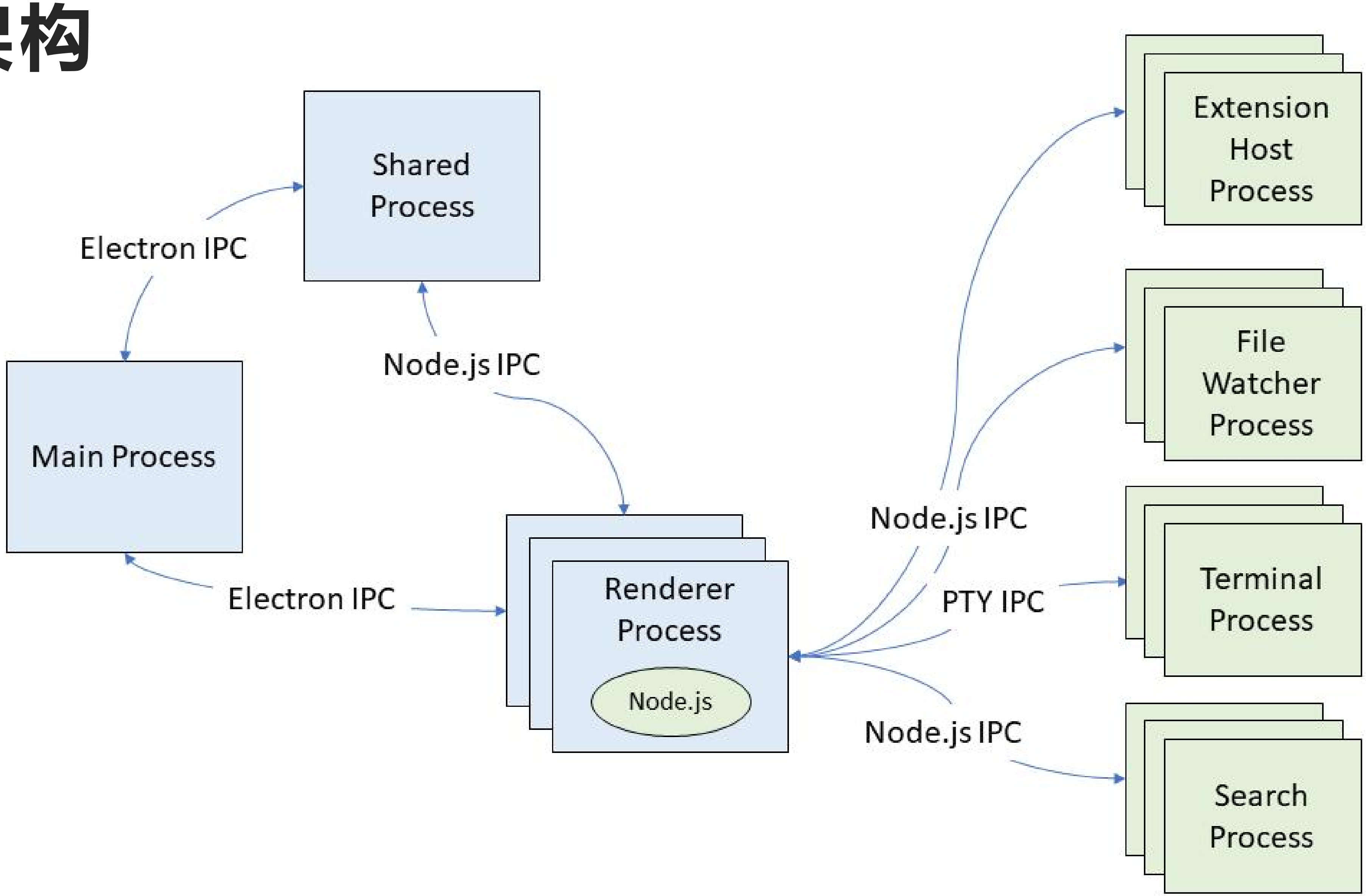
API: **vscode.d.ts**

Extension description: **package.json**

体系架构

进程模型

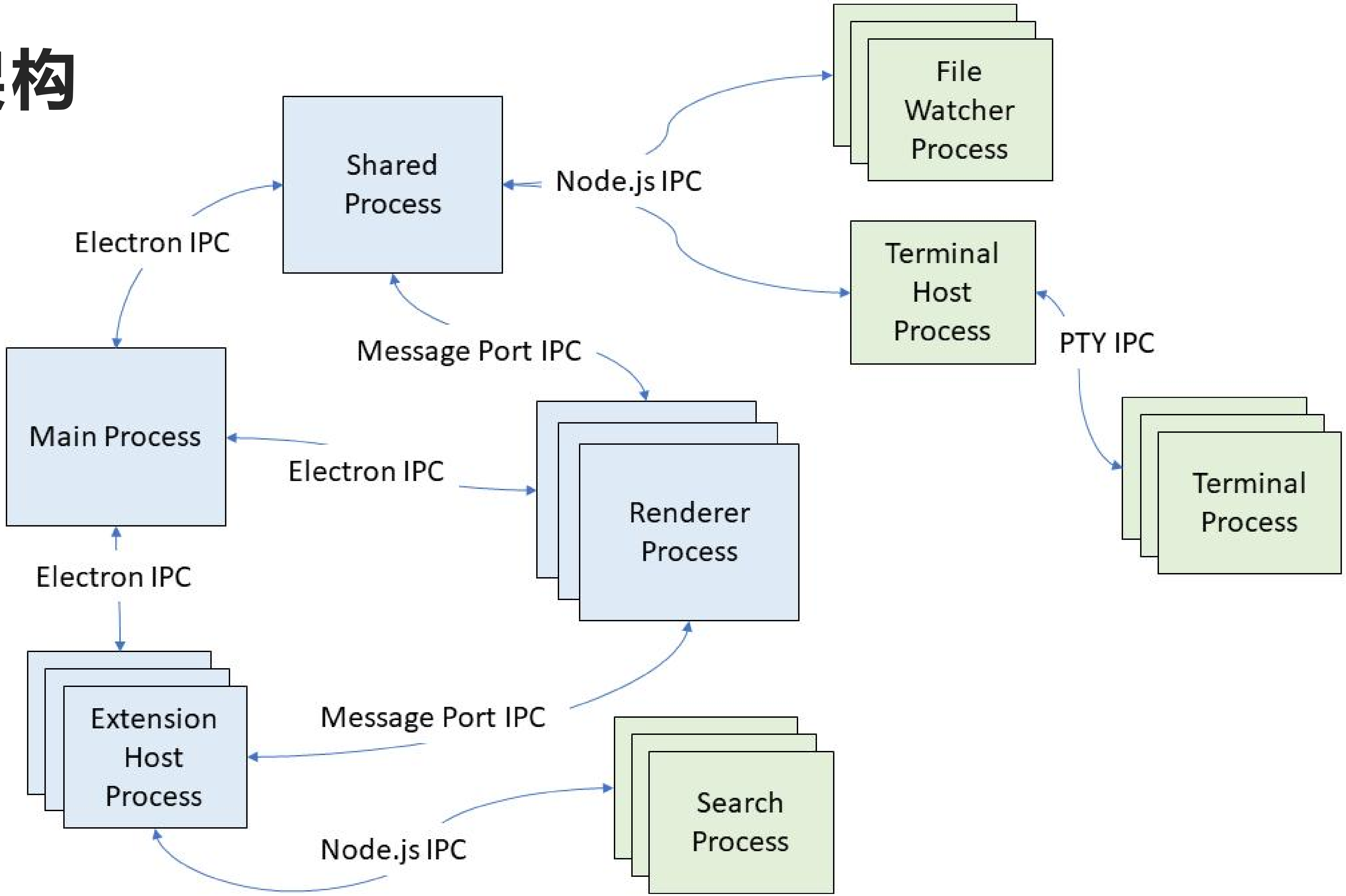
（沙盒前）



体系架构

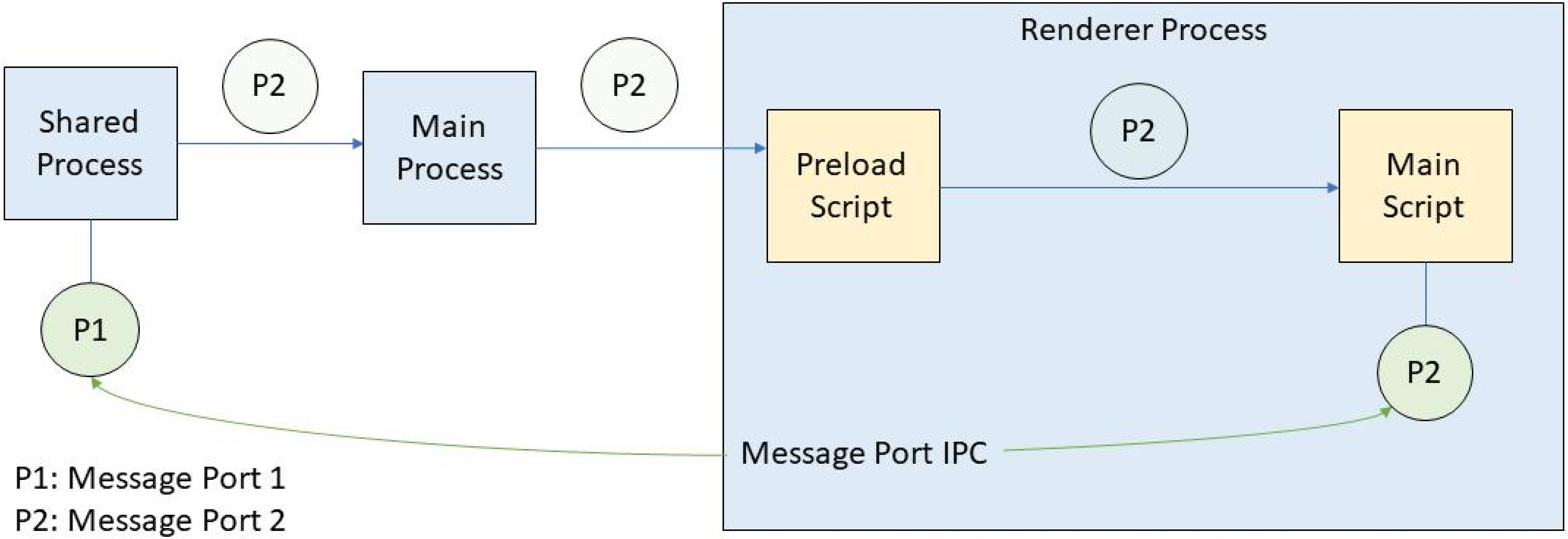
进程模型

（沙盒后）



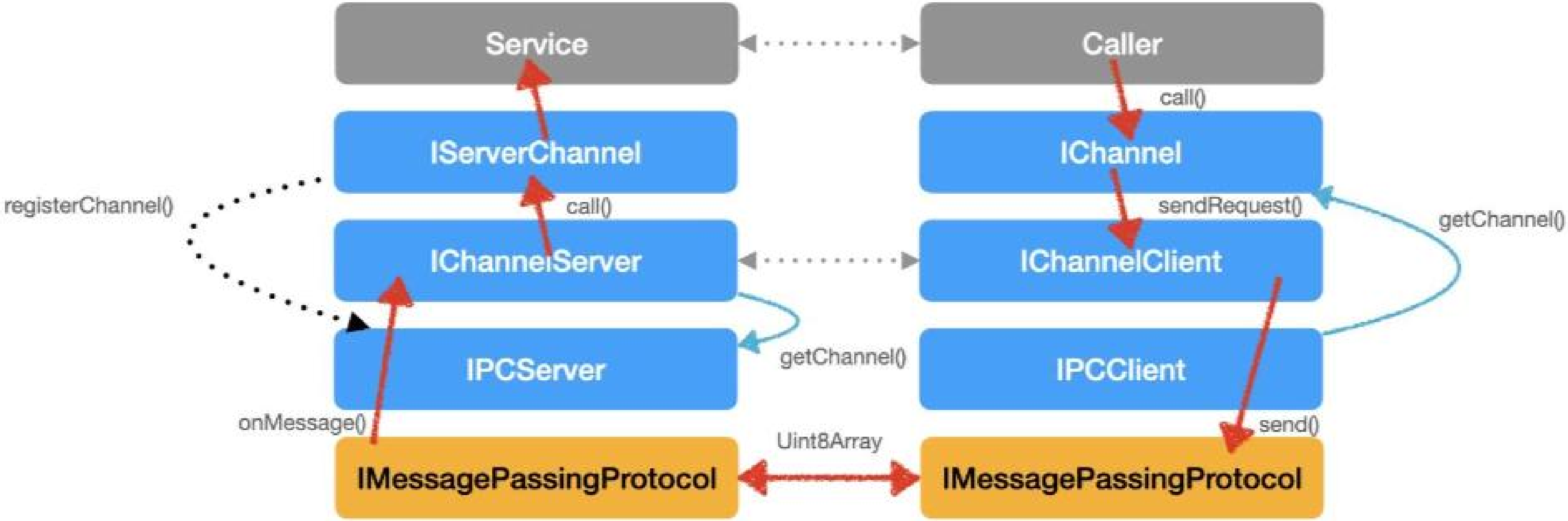
体系架构

MessagePorts



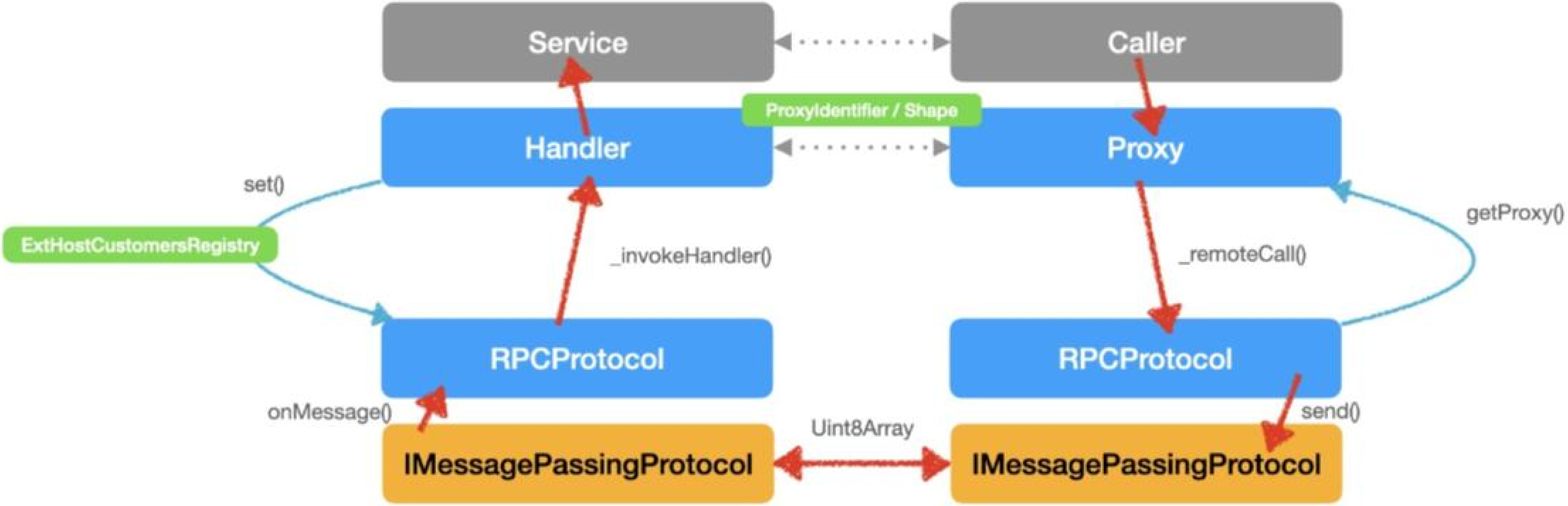
体系架构

Channel 通信



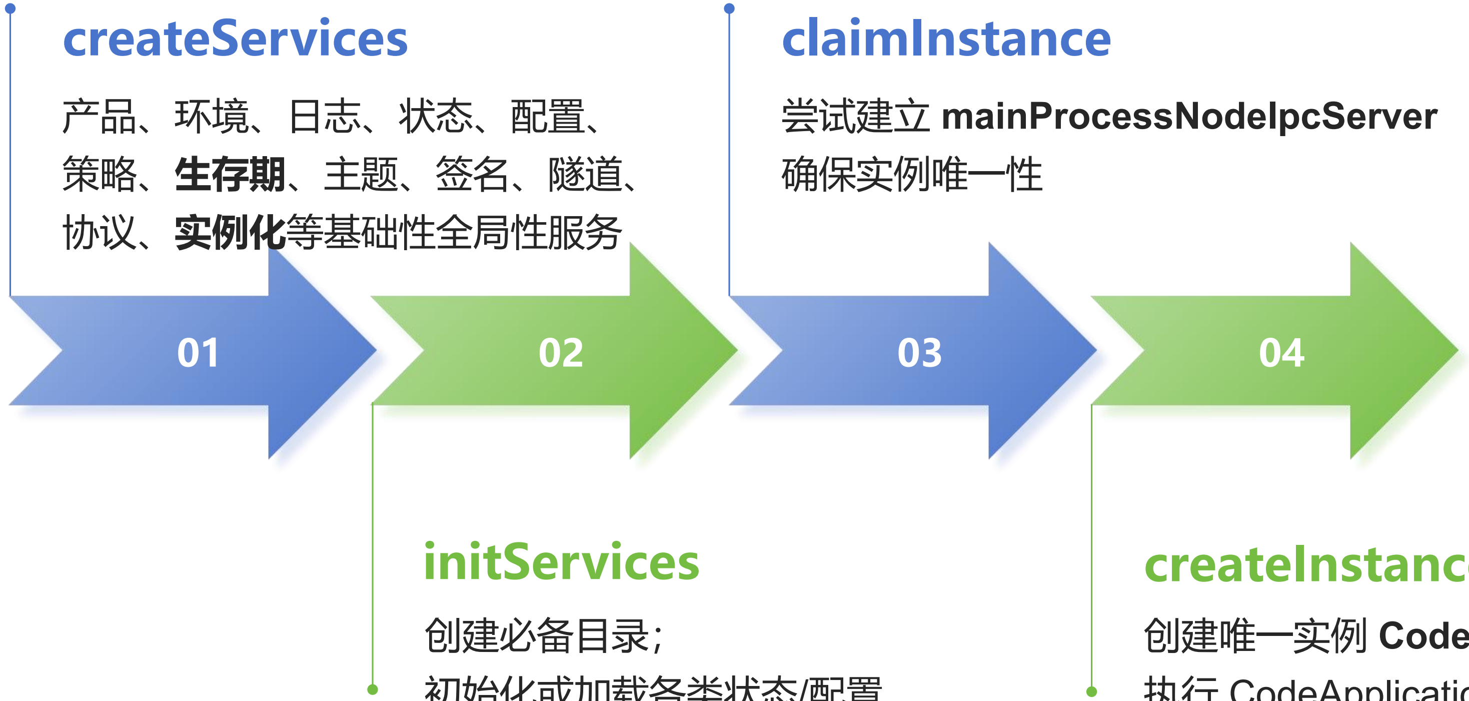
体系架构

RpcProtocol 通信



体系架构

主进程入口：code/electron-main/main.ts：**CodeMain.startup()** 方法



createServices

产品、环境、日志、状态、配置、策略、**生存期**、主题、签名、隧道、协议、**实例化**等基础性全局性服务

initServices

创建必备目录；

初始化或加载各类状态/配置

claimInstance

尝试建立 **mainProcessNodeIpcServer**

确保实例唯一性

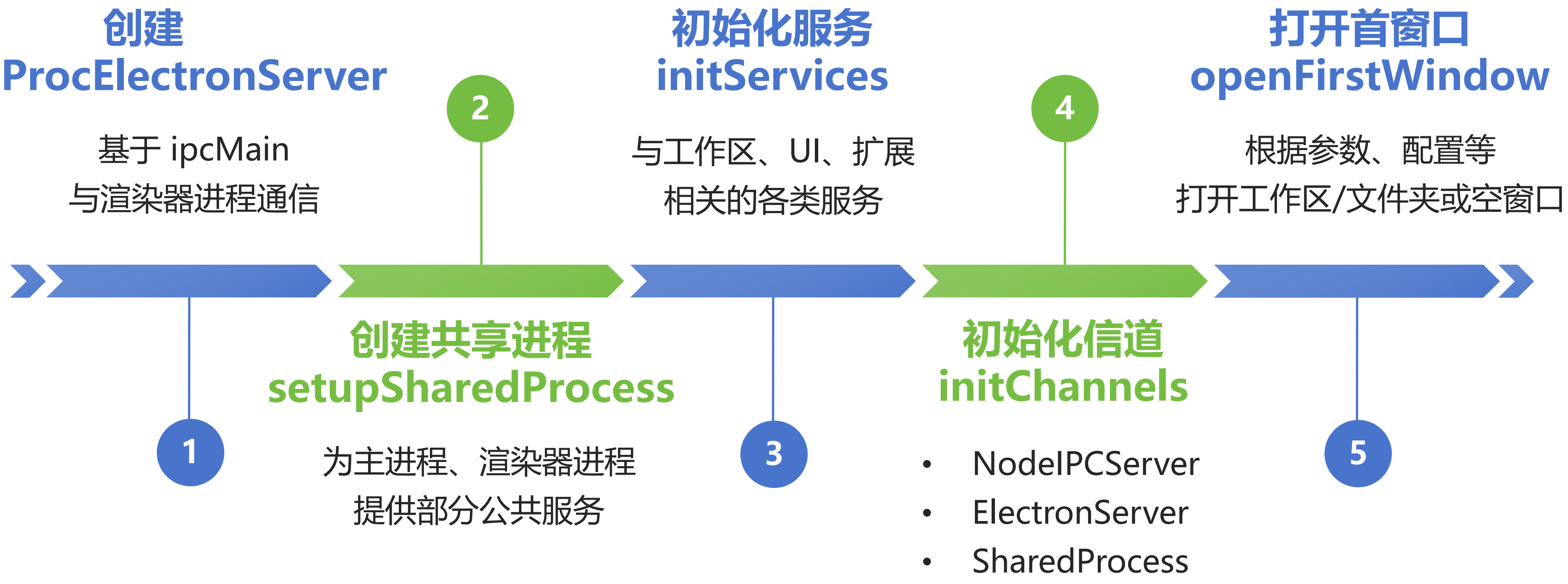
createInstance

创建唯一实例 **CodeApplication**；

执行 CodeApplication.startup()

体系架构

主进程唯一实例：code/electron-main/app.ts：CodeApplication.startup() 方法



1

创建 mainProcElectronServer

基于 ipcMain
与渲染器进程通信

2

创建共享进程 setupSharedProcess

为主进程、渲染器进程
提供部分公共服务

3

初始化服务 initServices

与工作区、UI、扩展
相关的各类服务

4

初始化信道 initChannels

- NodeIPCServer
- ElectronServer
- SharedProcess

5

打开首窗口 openFirstWindow

根据参数、配置等
打开工作区/文件夹或空窗口

体系架构

共享进程：code/node/sharedProcess/sharedProcessMain.ts

基于 Electron UtilityProcess

初始化服务及信道。

主要服务：诊断、扩展管理/提示、隧道、遥测、检验、语言包、用户数据同步等

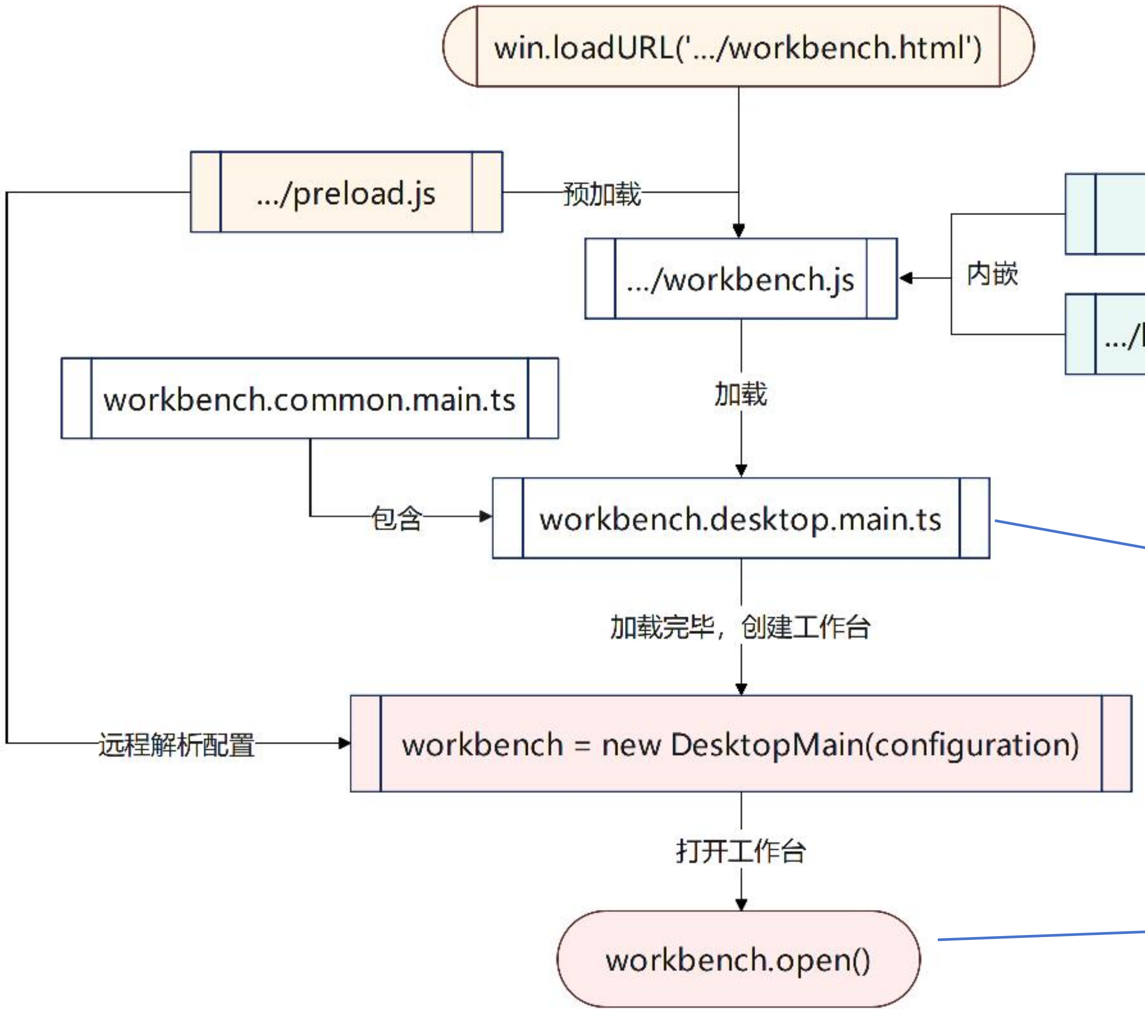
01

实例化工具类：

- 代码缓存清理
- 语言包缓存清理
- 工作区存储数据清理
- 日志数据清理
- 本地化更新
- 扩展迁移与过期清理
- 用户配置清理

02

体系架构 渲染器进程入口：workbench/workbench.desktop.main.ts



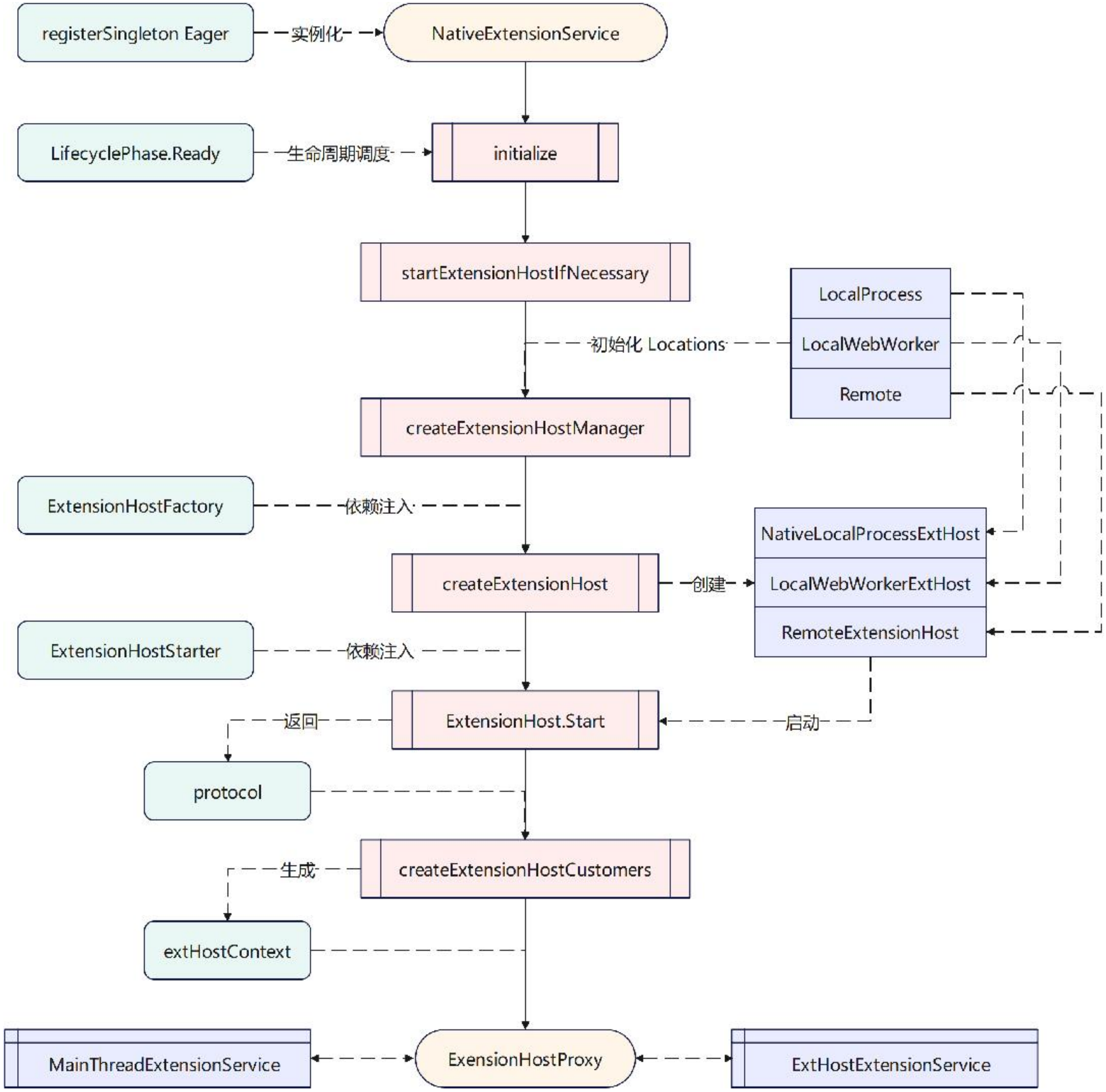
- 公共部分：动作、工作台组成、工作台服务、工作台贡献点、注册单例
- 桌面专有：services、contribs、singletons

创建布局，contribs 关联生存期，绑定上下文，恢复编辑器状态等

体系架构

扩展主机启动主流程

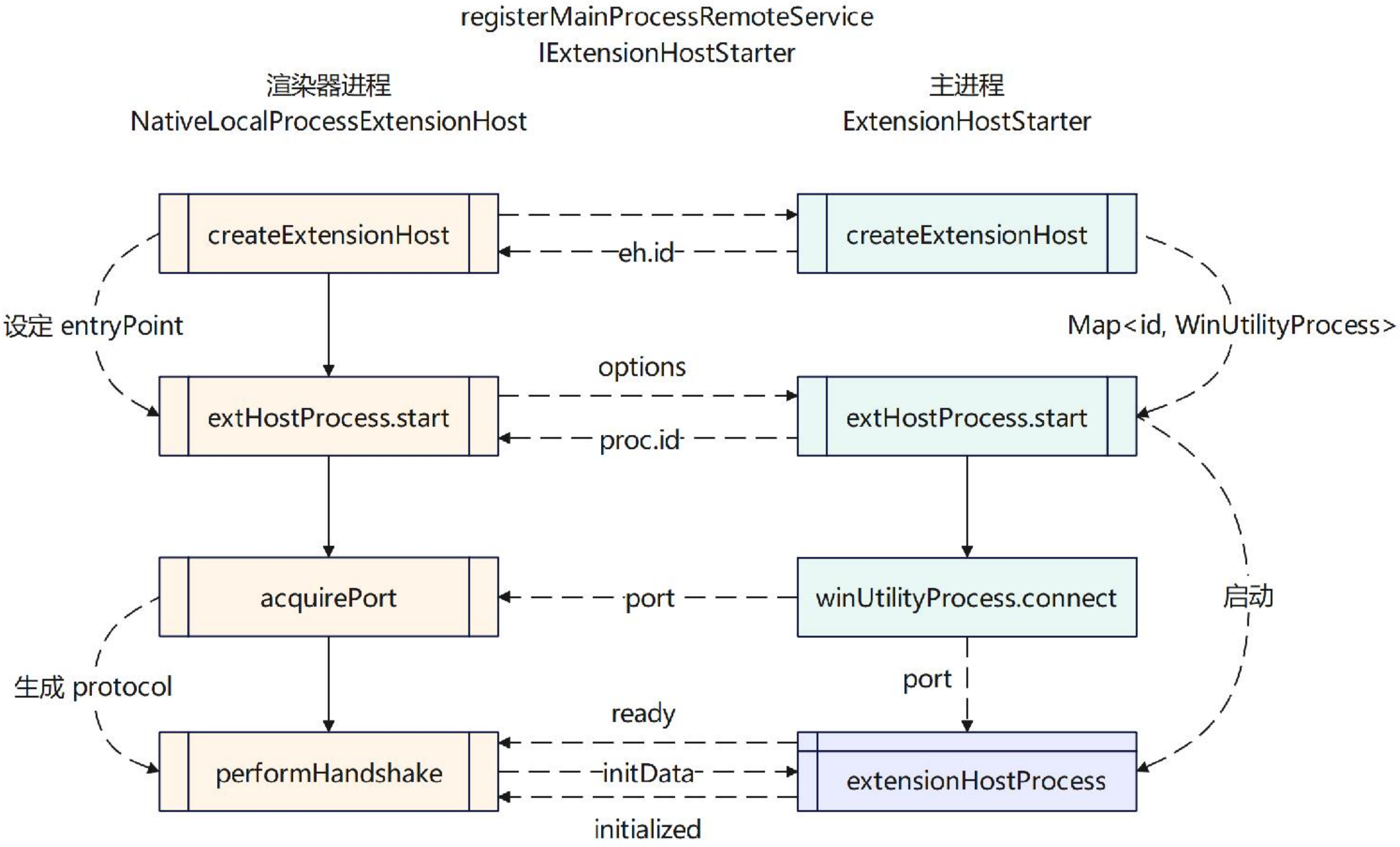
1. 主流程主要运行在渲染器进程
2. 从 ExtensionHost.Start 开始，创建扩展主机进程并频繁交互



体系架构

本地扩展主机进程

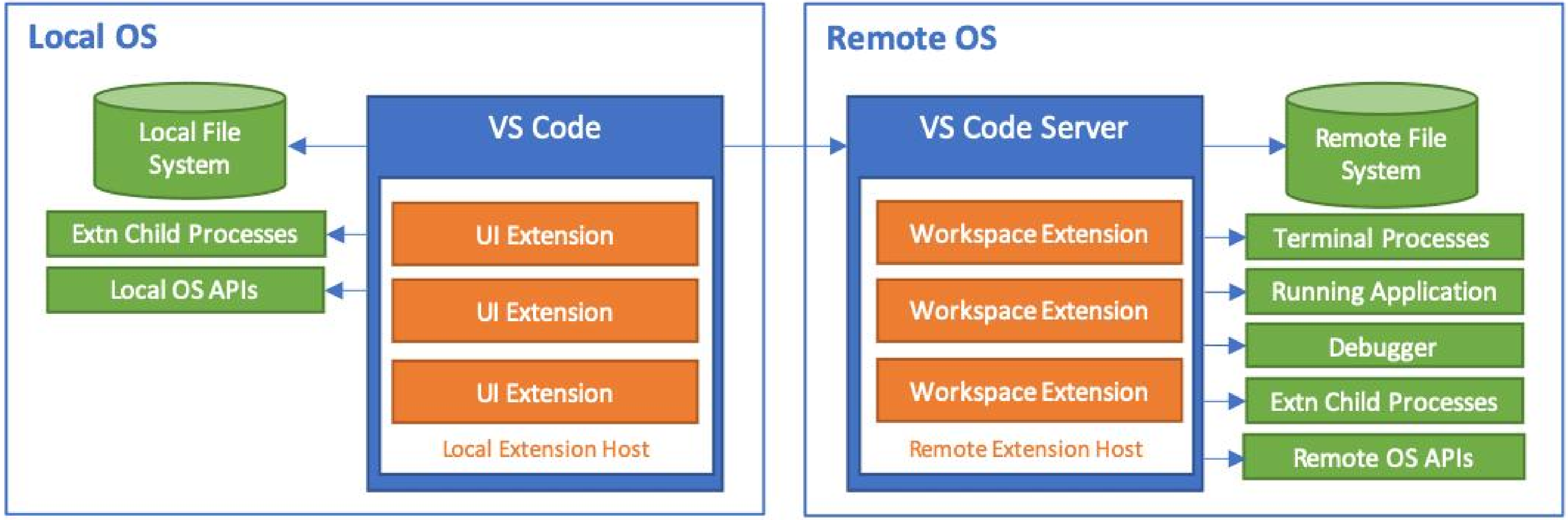
主进程生成一对消息端口，一个送给扩展主机进程，另一个发往渲染器进程，以供其通信

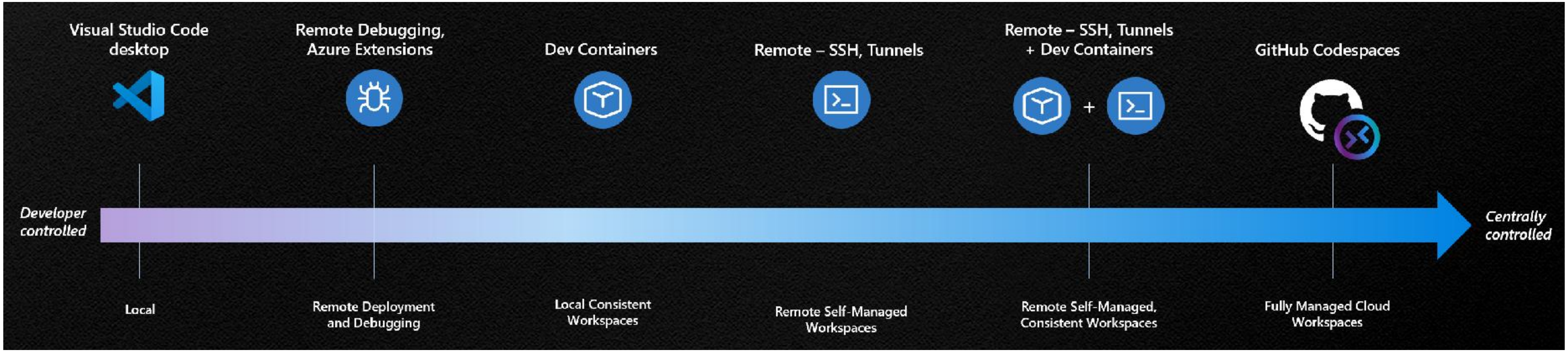


体系架构

远程开发

ExtensionHost 是在本地还是远程，并没有本质区别

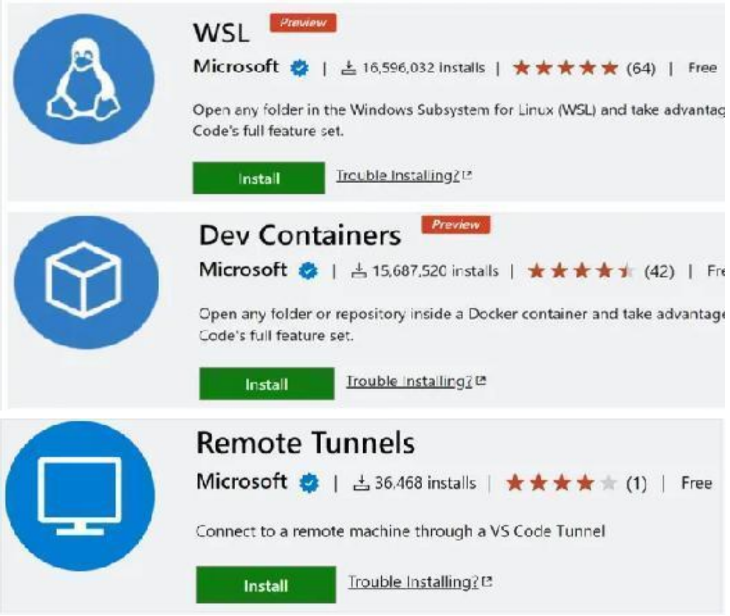




Visual Studio Code desktop
Remote Debugging, Azure Extensions
Dev Containers
Remote – SSH, Tunnels
Remote – SSH, Tunnels + Dev Containers
GitHub Codespaces
+
Developer controlled
Centrally controlled
Local
Remote Deployment and Debugging
Local Consistent Workspaces
Remote Self-Managed Workspaces
Remote Self-Managed, Consistent Workspaces
Fully Managed Cloud Workspaces

体系架构

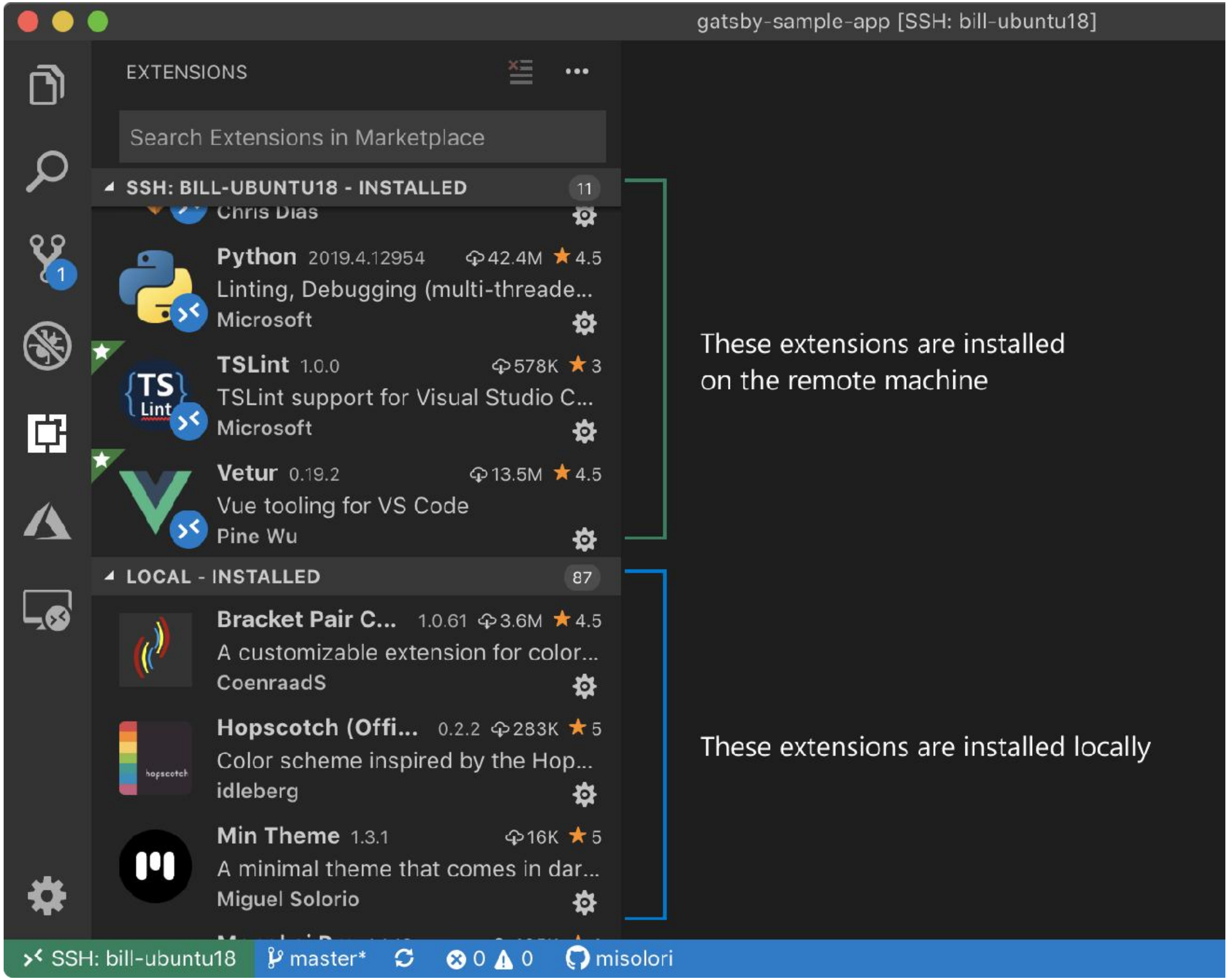
远程开发



WSL
Microsoft | 16,596,032 installs | ★★★★★ (64) | Free
Open any folder in the Windows Subsystem for Linux (WSL) and take advantag
Code's full feature set.

Dev Containers
Microsoft | 15,687,520 installs | ★★★★☆ (42) | Fre
Open any folder or repository inside a Docker container and take advantage
Code's full feature set.

Remote Tunnels
Microsoft | 36,468 installs | ★★★★☆ (1) | Free
Connect to a remote machine through a VS Code Tunnel



gatsby-sample-app [SSH: bill-ubuntu18]

EXTENSIONS

Search Extensions in Marketplace

SSH: BILL-UBUNTU18 - INSTALLED 11

Chris Dias

Python 2019.4.12954 42.4M ★4.5
Linting, Debugging (multi-threade...
Microsoft

TSLint 1.0.0 578K ★3
TSLint support for Visual Studio C...
Microsoft

Vetur 0.19.2 13.5M ★4.5
Vue tooling for VS Code
Pine Wu

LOCAL - INSTALLED 87

Bracket Pair C... 1.0.61 3.6M ★4.5
A customizable extension for color...
CoenraadS

Hopscotch (Offi... 0.2.2 283K ★5
Color scheme inspired by the Hop...
idleberg

Min Theme 1.3.1 16K ★5
A minimal theme that comes in dar...
Miguel Solorio

SSH: bill-ubuntu18 master* 0 0 misolori

These extensions are installed on the remote machine

These extensions are installed locally

体系架构

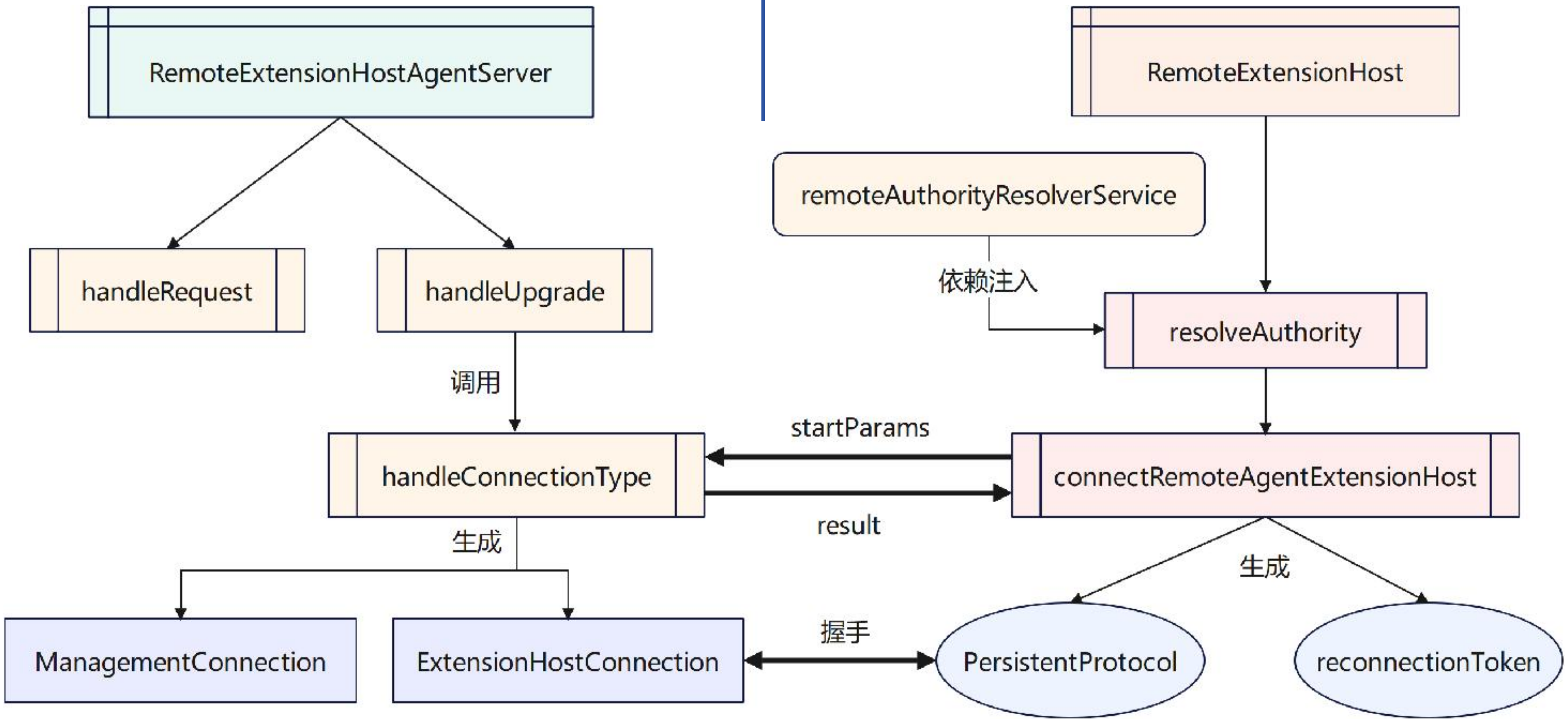
远程开发

具体连接步骤：

1. 创建端口
2. 确定协议
3. 发送认证
4. 签名验证
5. 请求连接类型

RemoteExtensionHost

渲染器进程



RemoteExtensionHostAgentServer

handleRequest

handleUpgrade

调用

handleConnectionType

生成

ManagementConnection

ExtensionHostConnection

RemoteExtensionHost

remoteAuthorityResolverService

依赖注入

resolveAuthority

startParams

result

connectRemoteAgentExtensionHost

生成

握手

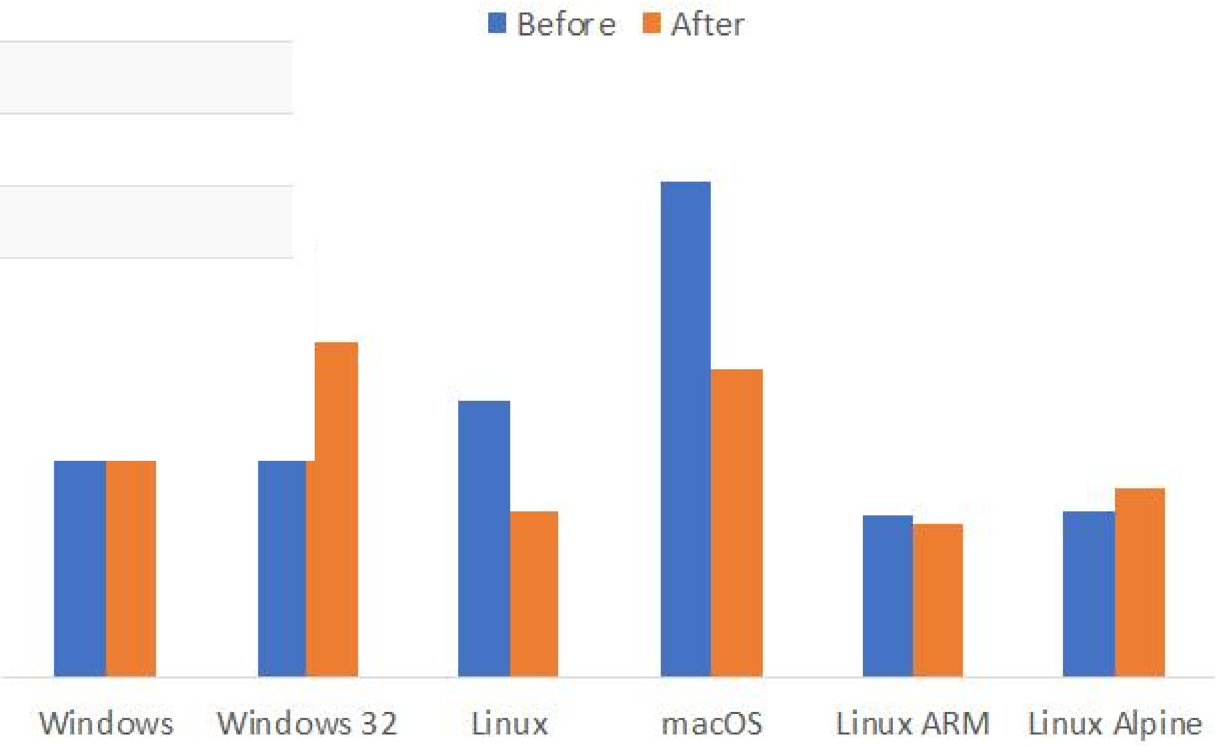
PersistentProtocol

reconnectionToken

源码构建

Platform	Before	After	Time Savings
Windows	58 min	44 min	24%
Windows 32	59 min	46 min	22%
Linux	38 min	23 min	39%
macOS	68 min	42 min	38%
Linux Arm	22 min	21 min	5%
Linux Alpine	23 min	26 min	-13%

■ Before ■ After



源码构建

源码启动 code.sh/bat：**build/lib/preLaunch.js** → launch executable

1 若无 node_modules 目录，执行 yarn [install]，期间会调用 preinstall.js 检查组件版本，安装完成后调用 postinstall.js 对指定目录执行后续安装；

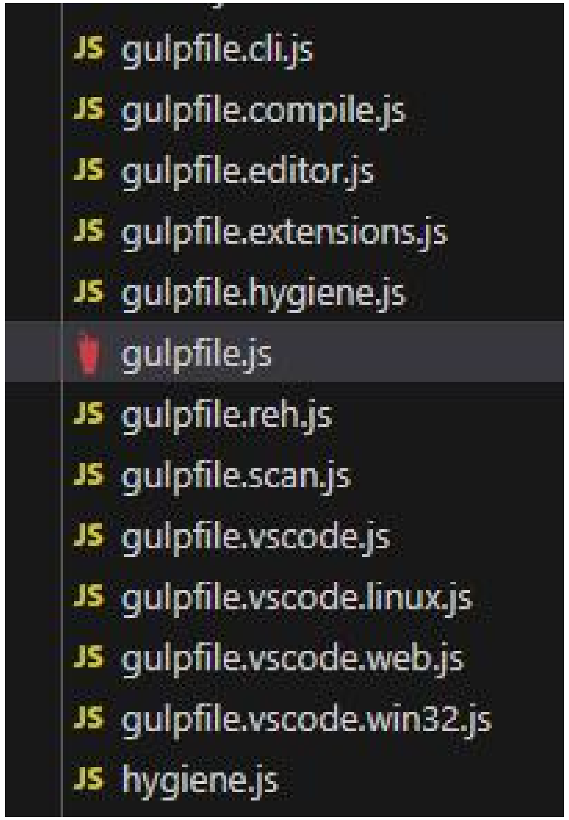
2 执行 yarn electron 检查并下载必要的 Electron 版本；

3 若无 out 目录，执行 yarn compile，启动 **build/gulpfile.js** 下定义的 compile 任务；

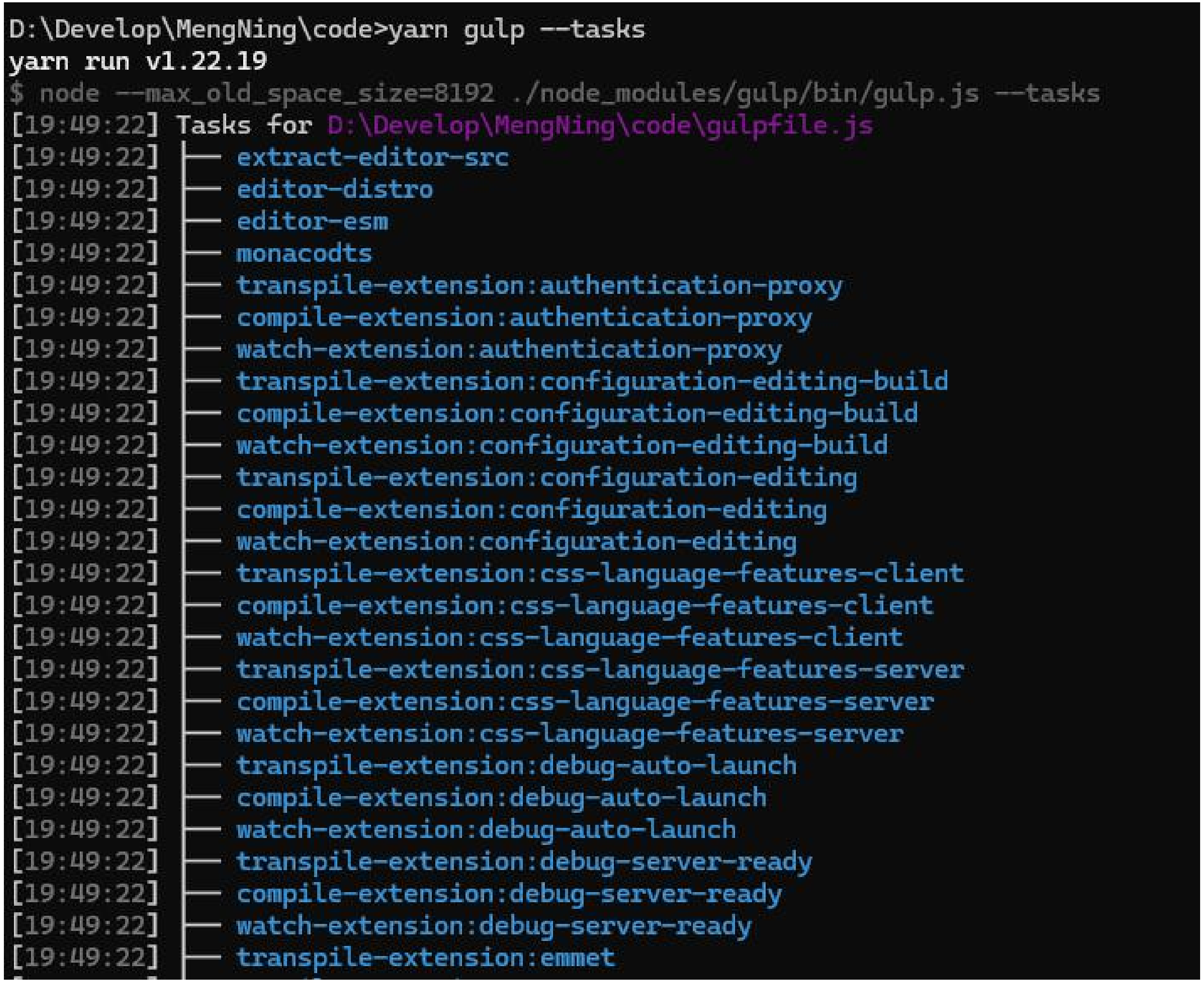
4 同步内置扩展，依据 product.json 中 builtInExtensions 从微软插件市场下载指定的扩展。

源码构建

基于 gulp，共 303 项任务



```
gulpfile.cli.js
gulpfile.compile.js
gulpfile.editor.js
gulpfile.extensions.js
gulpfile.hygiene.js
gulpfile.js
gulpfile.reh.js
gulpfile.scan.js
gulpfile.vscode.js
gulpfile.vscode.linux.js
gulpfile.vscode.web.js
gulpfile.vscode.win32.js
hygiene.js
```



```
D:\Develop\MengNing\code>yarn gulp --tasks
yarn run v1.22.19
$ node --max_old_space_size=8192 ./node_modules/gulp/bin/gulp.js --tasks
[19:49:22] Tasks for D:\Develop\MengNing\code\gulpfile.js
[19:49:22] ├── extract-editor-src
[19:49:22] ├── editor-distro
[19:49:22] ├── editor-esm
[19:49:22] ├── monacodts
[19:49:22] ├── transpile-extension:authentication-proxy
[19:49:22] ├── compile-extension:authentication-proxy
[19:49:22] ├── watch-extension:authentication-proxy
[19:49:22] ├── transpile-extension:configuration-editing-build
[19:49:22] ├── compile-extension:configuration-editing-build
[19:49:22] ├── watch-extension:configuration-editing-build
[19:49:22] ├── transpile-extension:configuration-editing
[19:49:22] ├── compile-extension:configuration-editing
[19:49:22] ├── watch-extension:configuration-editing
[19:49:22] ├── transpile-extension:css-language-features-client
[19:49:22] ├── compile-extension:css-language-features-client
[19:49:22] ├── watch-extension:css-language-features-client
[19:49:22] ├── transpile-extension:css-language-features-server
[19:49:22] ├── compile-extension:css-language-features-server
[19:49:22] ├── watch-extension:css-language-features-server
[19:49:22] ├── transpile-extension:debug-auto-launch
[19:49:22] ├── compile-extension:debug-auto-launch
[19:49:22] ├── watch-extension:debug-auto-launch
[19:49:22] ├── transpile-extension:debug-server-ready
[19:49:22] ├── compile-extension:debug-server-ready
[19:49:22] ├── watch-extension:debug-server-ready
[19:49:22] ├── transpile-extension:emmet
```

源码构建

x64 arm64

典型最终桌面版任务：vscode-platform-arch[-min][-ci]

win32 linux darwin

典型最终服务器任务：vscode-reh[-web]-platform-arch[-min][-ci]

x64 arm64 armhf alpine

Windows 安装任务：vscode-win32-platform-system/user-setup

vscode-win32-platform-inno-updater

源码构建

compile = series

- clean
- build-web-node-paths
- compile-api-proposal-names
- compile-src

optimize 区分入口文件及依赖进行合并、嵌入

minify　最小化资源 js/css/svg

vscodeTaskCI = series

- packageTask
- patchWuin32DependenciesTask

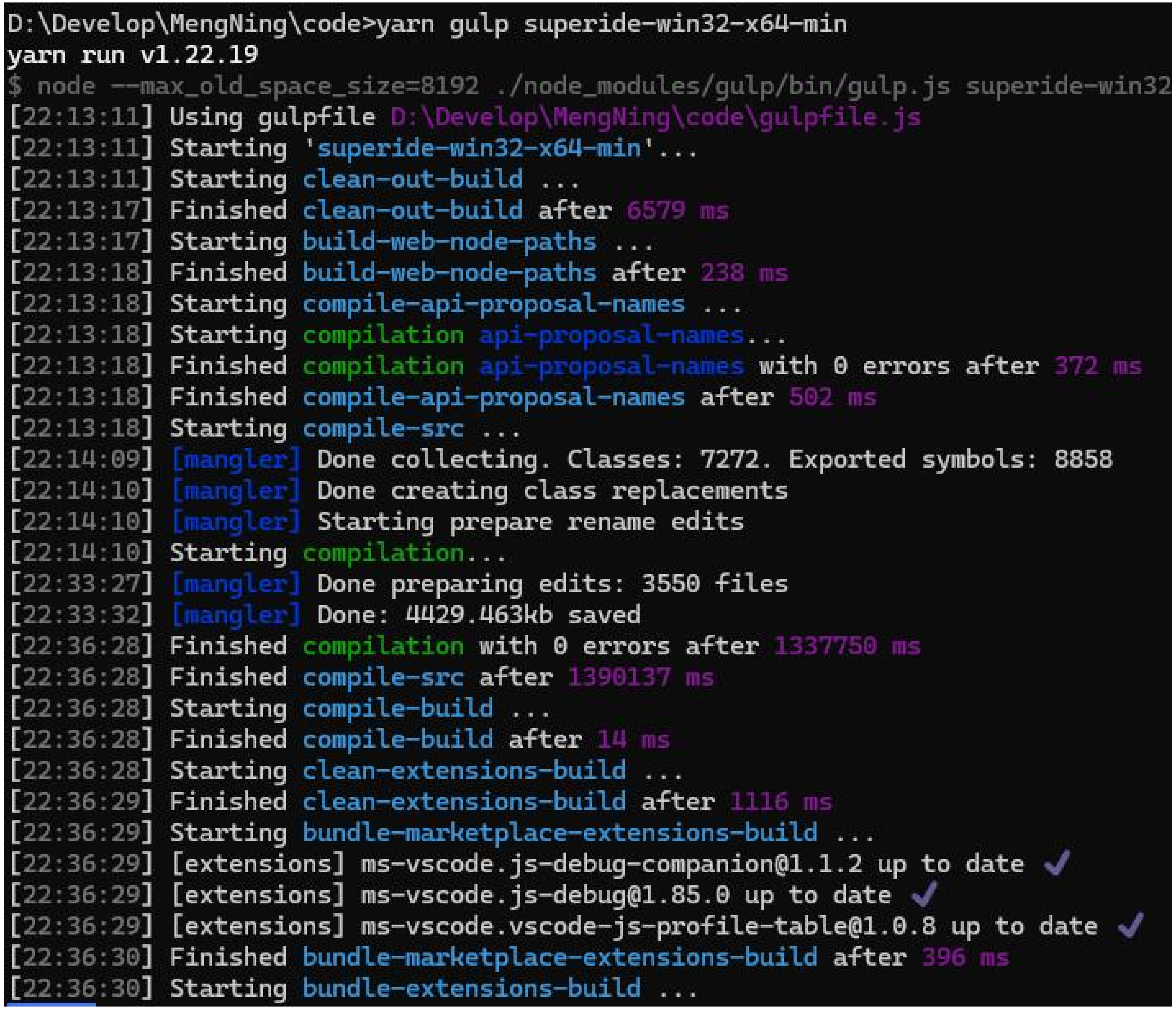
vscodeTask = series

- compile-build
- compile-extensions-build
- compile-extension-media-build
- optimize-vscode
- minify-vscode
- vscodeTaskCI

源码构建

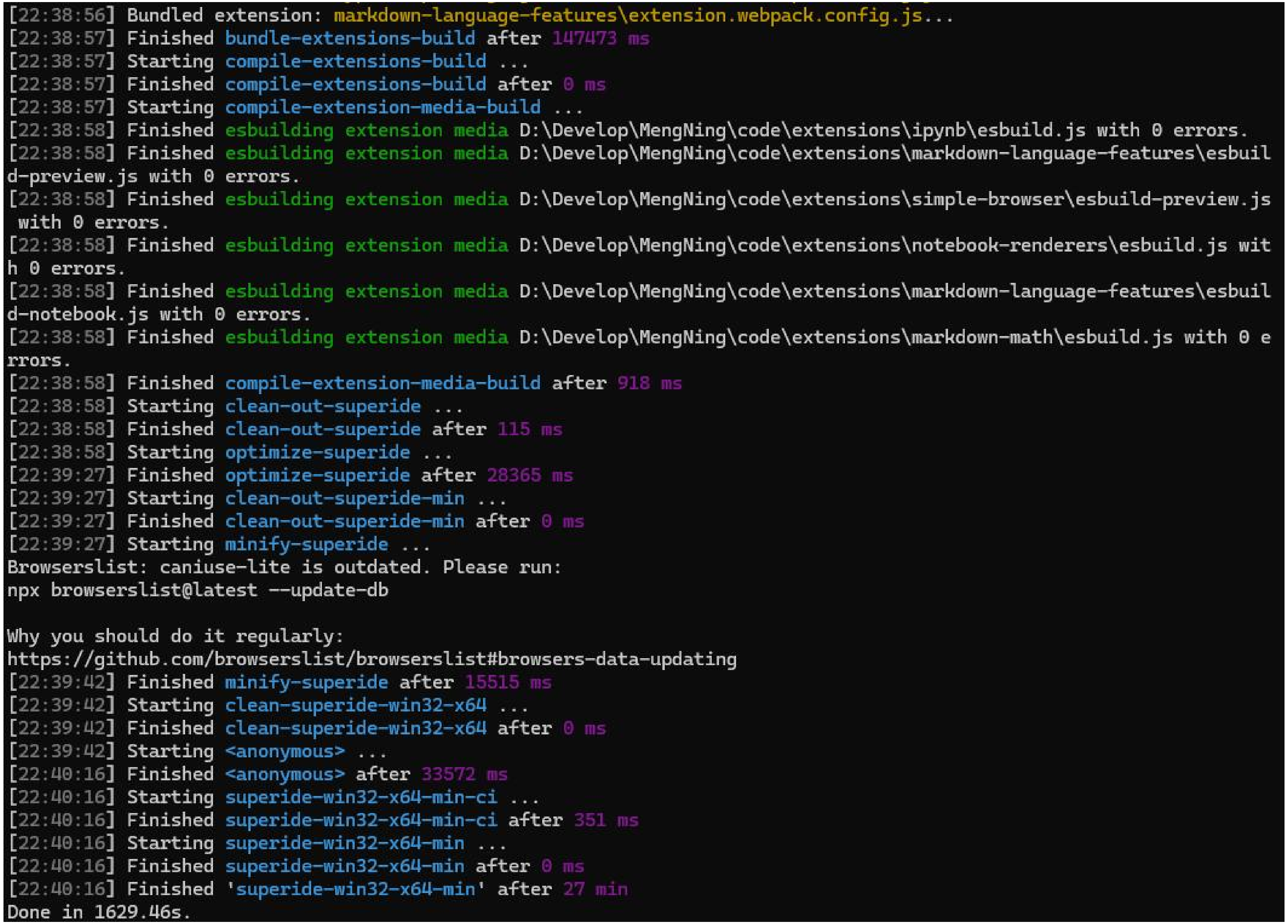
[mangler] 用于 TypeScript 代码的名称简化

可以在 makeCompileBuildTask 里设置 disableMangle: true 来禁用它，提升编译速度



```
D:\Develop\MengNing\code>yarn gulp superide-win32-x64-min
yarn run v1.22.19
$ node --max_old_space_size=8192 ./node_modules/gulp/bin/gulp.js superide-win32
[22:13:11] Using gulpfile D:\Develop\MengNing\code\gulpfile.js
[22:13:11] Starting 'superide-win32-x64-min'...
[22:13:11] Starting clean-out-build ...
[22:13:17] Finished clean-out-build after 6579 ms
[22:13:17] Starting build-web-node-paths ...
[22:13:18] Finished build-web-node-paths after 238 ms
[22:13:18] Starting compile-api-proposal-names ...
[22:13:18] Starting compilation api-proposal-names...
[22:13:18] Finished compilation api-proposal-names with 0 errors after 372 ms
[22:13:18] Finished compile-api-proposal-names after 502 ms
[22:13:18] Starting compile-src ...
[22:14:09] [mangler] Done collecting. Classes: 7272. Exported symbols: 8858
[22:14:10] [mangler] Done creating class replacements
[22:14:10] [mangler] Starting prepare rename edits
[22:14:10] Starting compilation...
[22:33:27] [mangler] Done preparing edits: 3550 files
[22:33:32] [mangler] Done: 4429.463kb saved
[22:36:28] Finished compilation with 0 errors after 1337750 ms
[22:36:28] Finished compile-src after 1390137 ms
[22:36:28] Starting compile-build ...
[22:36:28] Finished compile-build after 14 ms
[22:36:28] Starting clean-extensions-build ...
[22:36:29] Finished clean-extensions-build after 1116 ms
[22:36:29] Starting bundle-marketplace-extensions-build ...
[22:36:29] [extensions] ms-vscode.js-debug-companion@1.1.2 up to date ✓
[22:36:29] [extensions] ms-vscode.js-debug@1.85.0 up to date ✓
[22:36:29] [extensions] ms-vscode.vscode-js-profile-table@1.0.8 up to date ✓
[22:36:30] Finished bundle-marketplace-extensions-build after 396 ms
[22:36:30] Starting bundle-extensions-build ...
```

源码构建



```
[22:38:56] Bundled extension: markdown-language-features\extension.webpack.config.js...
[22:38:57] Finished bundle-extensions-build after 147473 ms
[22:38:57] Starting compile-extensions-build ...
[22:38:57] Finished compile-extensions-build after 0 ms
[22:38:57] Starting compile-extension-media-build ...
[22:38:58] Finished esbuilding extension media D:\Develop\MengNing\code\extensions\ipynb\esbuild.js with 0 errors.
[22:38:58] Finished esbuilding extension media D:\Develop\MengNing\code\extensions\markdown-language-features\esbuil
d-preview.js with 0 errors.
[22:38:58] Finished esbuilding extension media D:\Develop\MengNing\code\extensions\simple-browser\esbuild-preview.js
 with 0 errors.
[22:38:58] Finished esbuilding extension media D:\Develop\MengNing\code\extensions\notebook-renderers\esbuild.js wit
h 0 errors.
[22:38:58] Finished esbuilding extension media D:\Develop\MengNing\code\extensions\markdown-language-features\esbuil
d-notebook.js with 0 errors.
[22:38:58] Finished esbuilding extension media D:\Develop\MengNing\code\extensions\markdown-math\esbuild.js with 0 e
rrors.
[22:38:58] Finished compile-extension-media-build after 918 ms
[22:38:58] Starting clean-out-superide ...
[22:38:58] Finished clean-out-superide after 115 ms
[22:38:58] Starting optimize-superide ...
[22:39:27] Finished optimize-superide after 28365 ms
[22:39:27] Starting clean-out-superide-min ...
[22:39:27] Finished clean-out-superide-min after 0 ms
[22:39:27] Starting minify-superide ...
Browserslist: caniuse-lite is outdated. Please run:
npx browserslist@latest --update-db

Why you should do it regularly:
https://github.com/browserslist/browserslist#browsers-data-updating
[22:39:42] Finished minify-superide after 15515 ms
[22:39:42] Starting clean-superide-win32-x64 ...
[22:39:42] Finished clean-superide-win32-x64 after 0 ms
[22:39:42] Starting <anonymous> ...
[22:40:16] Finished <anonymous> after 33572 ms
[22:40:16] Starting superide-win32-x64-min-ci ...
[22:40:16] Finished superide-win32-x64-min-ci after 351 ms
[22:40:16] Starting superide-win32-x64-min ...
[22:40:16] Finished superide-win32-x64-min after 0 ms
[22:40:16] Finished 'superide-win32-x64-min' after 27 min
Done in 1629.46s.
```

Q & A

魏兆华

江苏 南京

扫一扫上面的二维码图案，加我为朋友。

扫码来知乎关注我

魏兆华