

ctlm SFTP Extension - 아키텍처 문서

▣ 목차

1. 전체 플로우차트
2. 시퀀스 다이어그램
3. 클래스 다이어그램
4. 상태 다이어그램
5. 시스템 아키텍처
6. 데이터 플로우

전체 플로우차트

확장 프로그램의 전체 작동 흐름을 보여줍니다.

```
flowchart TD
    Start([VS Code 시작]) --> Activate[Extension 활성화  
extension.ts]

    Activate --> InitComponents[컴포넌트 초기화]
    InitComponents --> TreeProvider[SftpTreeProvider 생성]
    InitComponents --> Commands[명령어 등록]
    InitComponents --> Watchers[파일 감시자 등록]

    %% Tree Provider Flow
    TreeProvider --> LoadConfigs[설정 파일 로드  
.vscode/ctlm-sftp.json]
    LoadConfigs --> ShowServers[서버 목록 표시  
Activity Bar]

    ShowServers --> UserClickServer{사용자가  
서버 클릭?}
    UserClickServer -->|Yes| Connect[SftpClient.connect]
    Connect --> Connected[연결 성공]
    Connected --> ListRemote[원격 파일 목록 조회]
    ListRemote --> ShowFiles[트리 뷰에 파일 표시]

    %% File Save Flow
    Watchers --> OnSave{파일 저장  
감지}
    OnSave -->|uploadOnSave=true| CheckScheme{파일 스킴  
체크}
    CheckScheme -->|scheme !== 'file'| Ignore1[업로드 무시  
Git commit 등]
    CheckScheme -->|scheme === 'file'| FindConfig1[설정 찾기  
findConfigByMetadata]

    FindConfig1 --> HasMetadata{메타데이터}
```

```

존재?}
HasMetadata -->|Yes| CheckConflict[충돌 검사
원격 수정시간 비교]
HasMetadata -->|No| CalcRemotePath[원격 경로 계산]

CheckConflict --> IsConflict{충돌 발견?}
IsConflict -->|Yes| ShowConflictDialog[충돌 대화상자]
ShowConflictDialog --> UserChoice{사용자 선택}
UserChoice -->|덮어쓰기| Upload[uploadFile
skipConflictCheck=true]
UserChoice -->|비교| ShowDiff[Diff 뷰 열기]
UserChoice -->|취소| CancelUpload[업로드 취소]

IsConflict -->|No| Upload
CalcRemotePath --> Upload

Upload --> EnsureDir[원격 디렉토리 생성
ensureRemoteDir]
EnsureDir --> PutFile[ssh2-sftp-client.put]
PutFile --> SaveMetadata[메타데이터 저장
.sftp-metadata/]
SaveMetadata --> Complete1([업로드 완료])

%% Download Flow
ShowFiles --> UserClickFile{사용자가
파일 클릭?}
UserClickFile -->|Yes| OpenRemoteFile[openRemoteFile 명령]
OpenRemoteFile --> CheckMetadata{메타데이터
존재?}

CheckMetadata -->|Yes| GetOriginalPath[원본 원격 경로 사용
metadata.remotePath]
CheckMetadata -->|No| CalcDownloadPath[원격 경로 계산]

GetOriginalPath --> DownloadFile[downloadFile]
CalcDownloadPath --> DownloadFile

DownloadFile --> CreateLocalDir[로컬 디렉토리 생성]
CreateLocalDir --> GetFile[ssh2-sftp-client.get]
GetFile --> SaveMetadata2[메타데이터 저장
remotePath + modifyTime]
SaveMetadata2 --> OpenEditor[에디터에서 열기]
OpenEditor --> Complete2([다운로드 완료])

%% Manual Commands
Commands --> ManualUpload[수동 업로드
ctlimSftp.upload]
Commands --> ManualDownload[수동 다운로드
ctlimSftp.download]
Commands --> ManualSync[전체 동기화
ctlimSftp.sync]
Commands --> DeleteRemote[원격 파일 삭제
ctlimSftp.deleteRemoteFile]

```

```

    ManualUpload --> FindConfig2[설정 선택]
    loadConfigWithSelection]
        FindConfig2 --> Upload

    ManualDownload --> FindConfig3[설정 선택]
        FindConfig3 --> DownloadFile

    ManualSync --> FindConfig4[설정 선택]
        FindConfig4 --> SyncFolder[syncFolder
모든 파일 업로드]

%% Multi-Server Support
subgraph MultiServer[다중 서버 지원]
    FindConfigByName[findConfigByName
metadata.configName으로 찾기]
    FindConfigByMetadata[findConfigByMetadata
localPath 인코딩으로 찾기]
    FindConfigForFile[findConfigForFile
경로 매칭으로 찾기]
end

%% Metadata System
subgraph MetadataSystem[메타데이터 시스템]
    MetaEncoding[파일명 인코딩
: → _c_
- → _u_
/ → __]
    MetaStorage[저장 위치
.vscode/.sftp-metadata/]
    MetaContent[저장 내용
remotePath
remoteModifyTime
localPath
configName]
end

%% Styling
classDef activeNode fill:#4CAF50,stroke:#2E7D32,stroke-width:2px,color:#fff
classDef errorNode fill:#F44336,stroke:#C62828,stroke-width:2px,color:#fff
classDef decisionNode fill:#FF9800,stroke:#E65100,stroke-width:2px,color:#fff
classDef processNode fill:#2196F3,stroke:#1565C0,stroke-width:2px,color:#fff

class Upload,DownloadFile,Connect activeNode
class IsConflict,UserChoice decisionNode
class CancelUpload,Ignore1 errorNode
class SaveMetadata,SaveMetadata2,FindConfig1,FindConfig2 processNode

```

시퀀스 다이어그램

파일 업로드 시 컴포넌트 간 상호작용을 시간 순서로 표현합니다.

```

sequenceDiagram
    actor User as 사용자
    participant VSCode as VS Code
    participant Ext as Extension
    participant SftpClient as SftpClient
    participant Metadata as Metadata System
    participant Server as SFTP Server

    User->>VSCode: 파일 저장 (Ctrl+S)
    VSCode->>Ext: onDidSaveTextDocument 이벤트

    Ext->>Ext: 파일 스킴 체크
    alt scheme !== 'file'
        Ext-->>VSCode: 업로드 무시 (Git commit 등)
    else scheme === 'file'
        Ext->>Metadata: findConfigByMetadata(localPath)
        Metadata->>Metadata: localPath 인코딩
    (: → _c_, _ → _u_)
        Metadata->>Metadata: 메타데이터 파일 검색

        alt 메타데이터 존재
            Metadata-->>Ext: SftpConfig + workspaceRoot
            Ext->>Server: stat(remotePath) - 수정시간 조회
            Server-->>Ext: remoteModifyTime

            Ext->>Metadata: 저장된 modifyTime과 비교

            alt 충돌 발견
                Ext->>User: 충돌 대화상자 표시
                User->>Ext: 선택 (덮어쓰기/비교/취소)

                alt 덮어쓰기
                    Ext->>SftpClient: uploadFile(skipConflictCheck=true)
                else 비교
                    Ext->>VSCode: Diff 뷰 열기
                else 취소
                    Ext-->>User: 업로드 중단
                end
            else 충돌 없음
                Ext->>SftpClient: uploadFile()
            end
        else 메타데이터 없음
            Ext->>Ext: 원격 경로 계산
            Ext->>SftpClient: uploadFile()
        end

        SftpClient->>SftpClient: ensureRemoteDir()
        SftpClient->>Server: mkdir -p (recursive)
        SftpClient->>Server: put(localPath, remotePath)
        Server-->>SftpClient: 업로드 완료

        SftpClient->>Server: stat(remotePath) - 최신 modifyTime
        Server-->>SftpClient: remoteModifyTime
    
```

```

SftpClient->>Metadata: saveFileMetadata()
(remotePath, modifyTime, configName)
    Metadata->>Metadata: JSON 파일 저장
(.vscode/.sftp-metadata/)

SftpClient-->>Ext: 업로드 성공
Ext->>VSCode: 상태 메시지 표시
VSCode-->>User: "파일이 업로드되었습니다"
end

```

클래스 다이어그램

코드의 클래스 구조와 관계를 표현합니다.

```

classDiagram
class Extension {
    -SftpClient sftpClient
    -SftpTreeProvider treeProvider
    -SftpConfig currentConfig
    +activate()
    +registerCommands()
    +registerWatchers()
    +findConfigByName(configName)
    +findConfigByMetadata(filePath)
    +findConfigForFile(filePath)
    +loadConfigWithSelection()
}

class SftpClient {
    -SftpClient2 client
    -boolean connected
    -OutputChannel outputChannel
    +connect(config)
    +disconnect()
    +isConnected()
    +uploadFile(localPath, config, skipConflictCheck)
    +downloadFile(localPath, config, workspaceFolder)
    +listRemoteFiles(remotePath)
    +deleteRemoteFile(remotePath, isDirectory)
    +syncFolder(localFolder, config)
    +getRemoteFileStats(remotePath)
    +getWorkspaceMetadataDir()
    +getDownloadFolder(remotePath, workspaceFolder, config)
    -ensureRemoteDir(remotePath)
    -saveFileMetadata(localPath, remotePath, time, config)
    -getFileMetadata(localPath, remotePath, config)
    -getMetadataDir(config)
    -getMetadataPath(localPath, remotePath, config)
    -getAllFiles(dir, ignore)
}

```

```
}

class SftpTreeProvider {
    -Map<string, Connection> connections
    -EventEmitter<TreeItem> _onDidChangeTreeData
    +getChildren(element)
    +getTreeItem(element)
    +connectToServer(serverItem)
    +disconnectServer(serverName)
    +refresh()
    +getConnectedServer(name)
    -loadServerList()
    -listRemoteFiles(path, client, config)
}

class SftpTreeItem {
    +string label
    +TreeItemCollapsibleState collapsibleState
    +string itemType
    +string remotePath
    +boolean isDirectory
    +SftpConfig config
    +ServerListItem serverItem
    +ThemeIcon iconPath
    +Command command
    -getFileIcon(fileName)
}

class SftpConfig {
    +string name
    +string context
    +string host
    +number port
    +string username
    +string password
    +string privateKey
    +string passphrase
    +string remotePath
    +boolean uploadOnSave
    +boolean|string downloadOnOpen
    +string[] ignore
    +object watcher
    +object profiles
    +string defaultProfile
    +string workspaceRoot
}

class FileMetadata {
    +string remotePath
    +number remoteModifyTime
    +string localPath
    +number downloadTime
    +string configName
    +string workspaceRoot
}
```

```

}

class RemoteFile {
    +string name
    +string path
    +boolean isDirectory
    +number size
    +Date modifyTime
}

class ServerListItem {
    +string name
    +string host
    +number port
    +string username
    +string remotePath
    +string configPath
}

Extension --> SftpClient : uses
Extension --> SftpTreeProvider : manages
Extension --> SftpConfig : loads

SftpClient --> SftpConfig : requires
SftpClient --> FileMetadata : creates/reads
SftpClient --> RemoteFile : returns

SftpTreeProvider --> SftpClient : creates
SftpTreeProvider --> SftpTreeItem : creates
SftpTreeProvider --> SftpConfig : uses
SftpTreeProvider --> ServerListItem : uses

SftpTreeItem --> SftpConfig : contains
SftpTreeItem --> ServerListItem : contains
SftpTreeItem --> RemoteFile : represents

```

상태 다이어그램

서버 연결 및 파일 작업의 상태 전환을 표현합니다.

```

stateDiagram-v2
[*] --> Disconnected: Extension 시작

Disconnected --> Connecting: connectServer 명령
Connecting --> Connected: 연결 성공
Connecting --> Error: 연결 실패

Connected --> Listing: 파일 목록 조회
Listing --> Idle: 조회 완료

```

Idle --> Uploading: uploadFile 호출
 Idle --> Downloading: downloadFile 호출
 Idle --> Deleting: deleteRemoteFile 호출
 Idle --> Syncing: syncFolder 호출

Uploading --> ConflictCheck: 메타데이터 있음
 Uploading --> Uploading2: 메타데이터 없음

ConflictCheck --> ConflictDetected: 수정시간 불일치
 ConflictCheck --> Uploading2: 충돌 없음

ConflictDetected --> UserPrompt: 대화상자 표시
 UserPrompt --> Uploading2: 덮어쓰기 선택
 UserPrompt --> DiffView: 비교 선택
 UserPrompt --> Idle: 취소 선택

DiffView --> Idle: 닫기

Uploading2 --> MetadataUpdate: 업로드 완료
 MetadataUpdate --> Idle: 메타데이터 저장

Downloading --> MetadataUpdate2: 다운로드 완료
 MetadataUpdate2 --> OpenFile: 메타데이터 저장
 OpenFile --> Idle: 에디터에서 열기

Deleting --> Idle: 삭제 완료

Syncing --> ProcessFiles: 파일 목록 수집
 ProcessFiles --> Uploading: 각 파일 업로드
 ProcessFiles --> Idle: 모든 파일 처리 완료

Connected --> Disconnecting: disconnectServer 명령
 Idle --> Disconnecting: disconnectServer 명령
 Error --> Disconnected: 재시도 대기

Disconnecting --> Disconnected: 연결 종료

Disconnected --> [*]: Extension 종료

```

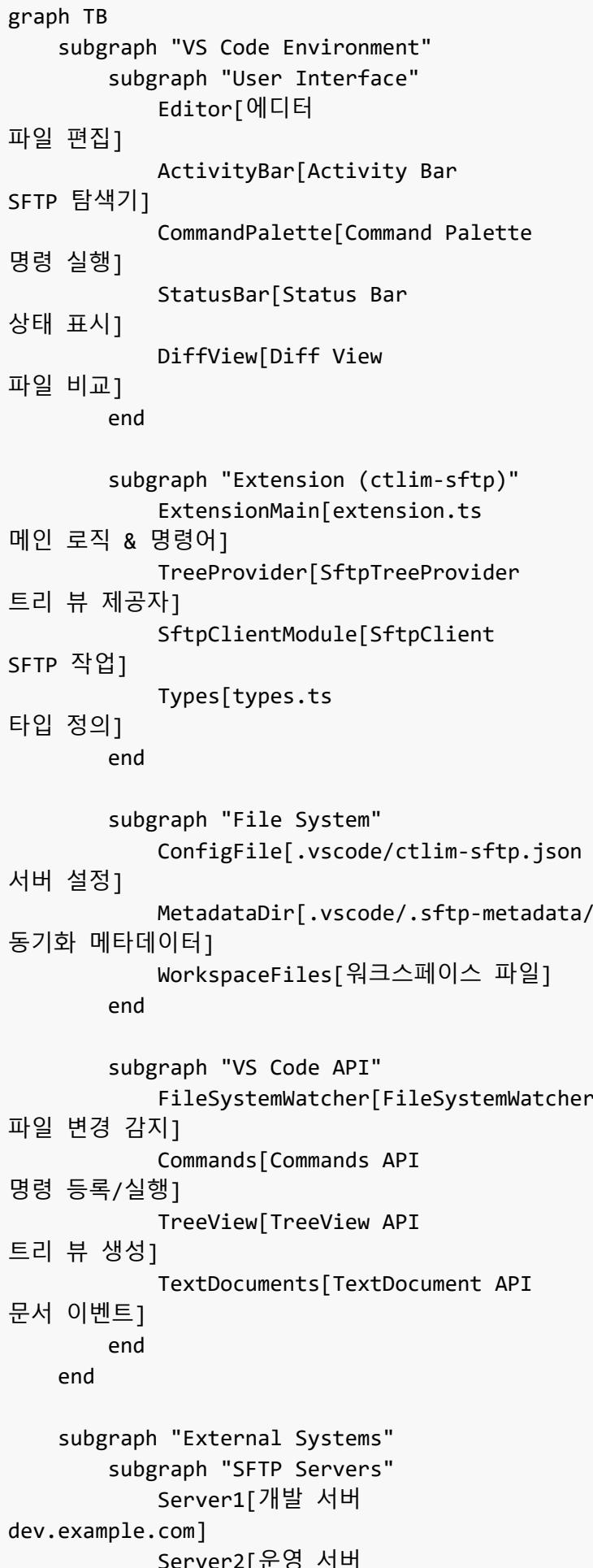
note right of ConflictCheck
  원격 파일의 수정시간과
  메타데이터의 수정시간 비교
end note
  
```

```

note right of MetadataUpdate
  remotePath, remoteModifyTime,
  localPath, configName 저장
end note
  
```

시스템 아키텍처

전체 시스템의 구성 요소와 관계를 표현합니다.



```

prod.example.com]
    ServerN[기타 서버
...
end

SSHLibrary[ssh2-sftp-client
npm package v10.0.3]
end

%% UI to Extension
Editor -->|저장 이벤트| ExtensionMain
ActivityBar -->|클릭 이벤트| TreeProvider
CommandPalette -->|명령 실행| ExtensionMain
ExtensionMain -->|메시지 표시| StatusBar
ExtensionMain -->|파일 비교| DiffView

%% Extension Internal
ExtensionMain -->|관리| TreeProvider
ExtensionMain -->|사용| SftpClientModule
TreeProvider -->|생성| SftpClientModule
ExtensionMain -->|참조| Types
TreeProvider -->|참조| Types
SftpClientModule -->|참조| Types

%% Extension to VS Code API
ExtensionMain -->|등록| Commands
ExtensionMain -->|등록| FileSystemWatcher
TreeProvider -->|등록| TreeView
ExtensionMain -->|구독| TextDocuments

%% Extension to File System
ExtensionMain -->|읽기/쓰기| ConfigFile
SftpClientModule -->|읽기/쓰기| MetadataDir
ExtensionMain -->|읽기| WorkspaceFiles
SftpClientModule -->|읽기/쓰기| WorkspaceFiles

%% Extension to External
SftpClientModule -->|의존| SSHLibrary
SSHLibrary -->|SSH/SFTP 프로토콜| Server1
SSHLibrary -->|SSH/SFTP 프로토콜| Server2
SSHLibrary -->|SSH/SFTP 프로토콜| ServerN

%% Styling
classDef uiStyle fill:#E3F2FD,stroke:#1976D2,stroke-width:2px
classDef extStyle fill:#FFF3E0,stroke:#F57C00,stroke-width:2px
classDef fsStyle fill:#F1F8E9,stroke:#689F38,stroke-width:2px
classDef serverStyle fill:#FCE4EC,stroke:#C2185B,stroke-width:2px
classDef apiStyle fill:#F3E5F5,stroke:#7B1FA2,stroke-width:2px

class Editor,ActivityBar,CommandPalette,StatusBar,DiffView uiStyle
class ExtensionMain,TreeProvider,SftpClientModule,Types extStyle
class ConfigFile,MetadataDir,WorkspaceFiles fsStyle
class Server1,Server2,ServerN,SSHLIBRARY serverStyle
class FileSystemWatcher,Commands,TreeView,TextDocuments apiStyle

```

데이터 플로우

메타데이터 시스템의 데이터 처리 흐름을 표현합니다.

```
graph TD
    subgraph Input [입력 데이터]
        LocalFile[로컬 파일  
D:\project\src\test.php]
        RemoteFile[원격 파일  
/var/www/html/test.php]
        ModifyTime[수정 시간  
1702876543210]
        ConfigName[서버 설정  
Development Server]
        end

        subgraph Encoding [경로 인코딩]
            LocalPath[localPath  
D:\project\src\test.php]
            SafePath[safeLocalPath  
D_c__u_MyProject_..._test.php]

            LocalPath -->|:_| SafePath
            LocalPath -->|_u_| SafePath
            LocalPath -->|/_| or \ SafePath
            end

            subgraph Metadata [메타데이터 저장]
                MetaFile[메타데이터 파일  
D_c__u_MyProject_..._test.php.json]
                MetaContent["{
                    remotePath: '/var/www/html/test.php',
                    remoteModifyTime: 1702876543210,
                    localPath: 'D:\\project\\\\src\\\\test.php',
                    downloadTime: 1702876600000,
                    configName: 'Development Server',
                    workspaceRoot: 'D:\\project'
                }"]
                end

                subgraph Storage [저장 위치]
                    MetaDir[.vscode/.sftp-metadata/  
workspaceRoot 별로 저장]
                    end

                    subgraph Lookup [Config 조회]
                        FindByMetadata[findConfigByMetadata
                            1. localPath 인코딩
                            2. 모든 config의 workspaceRoot 확인
                        ]
                        end
                    end
                end
            end
        end
    end
```

3. 메타데이터 파일 존재 체크]

 FindByName[findConfigByName

 metadata.configName 매칭]

 FindByPath[findConfigForFile

 경로 패턴 매칭]

 end

 subgraph Operations[작업 수행]

 Upload[업로드

 - 충돌 검사

 - 파일 전송

 - 메타데이터 생성]

 Download[다운로드

 - 원본 경로 복원

 - 파일 전송

 - 메타데이터 저장]

 ConflictCheck[충돌 감지]

 시간 비교]

 end

%% Data Flow

 LocalFile --> LocalPath

 LocalPath --> SafePath

 SafePath --> MetaFile

 RemoteFile --> MetaContent

 ModifyTime --> MetaContent

 LocalFile --> MetaContent

 ConfigName --> MetaContent

 MetaContent --> MetaFile

 MetaFile --> MetaDir

 MetaDir --> FindByMetadata

 MetaDir --> FindByName

 FindByMetadata -->|Config 반환| Upload

 FindByMetadata -->|Config 반환| Download

 FindByName -->|Config 반환| Upload

 FindByPath -->|Fallback| Upload

 MetaDir --> ConflictCheck

 ConflictCheck -->|시간 일치| Upload

 ConflictCheck -->|시간 불일치| Conflict[충돌 대화상자]

%% Styling

style Input fill:#E8F5E9,stroke:#4CAF50,stroke-width:2px

style Encoding fill:#FFF9C4,stroke:#FBC02D,stroke-width:2px

style Metadata fill:#E1F5FE,stroke:#0288D1,stroke-width:2px

style Storage fill:#F3E5F5,stroke:#7B1FA2,stroke-width:2px

style Lookup fill:#FFE0B2,stroke:#F57C00,stroke-width:2px

style Operations fill:#FFEBEE,stroke:#C62828,stroke-width:2px

📝 주요 개념 설명

메타데이터 인코딩

로컬 파일 경로를 안전한 파일명으로 변환:

- `:` → `_c_` (colon)
- `_` → `_u_` (underscore)
- `/` 또는 `\` → `__` (double underscore)

Config 조회 우선순위

1. **findConfigByMetadata**: 메타데이터 파일 존재 기반 (가장 정확)
2. **findConfigByName**: configName 필드 매칭
3. **findConfigForFile**: 경로 패턴 매칭 (fallback)

충돌 감지 메커니즘

1. 다운로드 시 `remoteModifyTime` 저장
2. 업로드 전 현재 원격 파일 시간 조회
3. 불일치 시 사용자 선택:
 - 덮어쓰기
 - 비교 (Diff 뷰)
 - 취소

다중 서버 지원

각 서버 config는 독립적인 `workspaceRoot`를 가지며, 메타데이터는 각 `workspaceRoot`의 `.vscode/.sftp-metadata/` 디렉토리에 저장됩니다.

🔧 기술 스택

- 언어: TypeScript (ES2020, strict mode)
- 플랫폼: VS Code Extension API v1.85.0+
- SFTP 라이브러리: ssh2-sftp-client v10.0.3
- 지원 프로토콜: SSH/SFTP with legacy algorithms

생성일: 2025-12-18