(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2025年1月16日(16.01.2025)

WIPO | PCT

(10) 国際公開番号
WO 2025/013261 A1

(51) 国際特許分類:
G06F 21/60 (2013.01) *G06F 21/57* (2013.01)

(21) 国際出願番号： PCT/JP2023/025784

(22) 国際出願日： 2023年7月12日(12.07.2023)

(25) 国際出願の言語： 日本語

(26) 国際公開の言語： 日本語

(71) 出願人: 日本電信電話株式会社 (**NIPPON TELEGRAPH AND TELEPHONE CORPORATION**) [JP/JP]; 〒1008116 東京都千代田区大手町一丁目５番１号 Tokyo (JP).

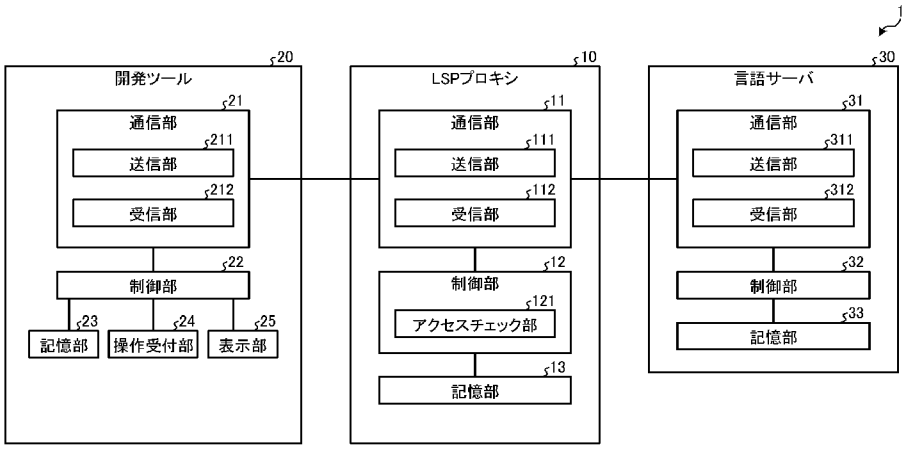
(72) 発明者: 斎藤 忍(**SAITO, Shinobu**); 〒1808585 東京都武蔵野市緑町３丁目９－１１ ＮＴＴ知的財産センタ内 Tokyo (JP).

(74) 代理人: 弁理士法人酒井国際特許事務所 (**SAKAI INTERNATIONAL PATENT OFFICE**); 〒1000013 東京都千代田区霞が関３丁目８番１号 虎ノ門ダイビルイースト Tokyo (JP).

(81) 指定国(表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU,

(54) **Title:** CONTROL DEVICE, CONTROL METHOD, AND CONTROL PROGRAM

(54) 発明の名称： 制御装置、制御方法および制御プログラム



10 LSP proxy
11, 21, 31 Communication unit
12, 22, 32 Control unit
13, 23, 33 Storage unit
20 Development tool
24 Operation reception unit
25 Display unit
30 Language server
111, 211, 311 Transmission unit
112, 212, 312 Reception unit
121 Access check unit

(57) **Abstract:** An LSP proxy (10) comprises: a transmission unit (111) that transmits, to a development tool (20) for editing code of a program, code that does not include confidential code; a reception unit (112) that receives, from the development tool (20), a command relating to editing of the code that does not include confidential code transmitted by the transmission unit (111); and an access check unit (121) that, on the basis of the command received by the reception unit (112), executes control relating to editing and checking relating to access to the code.

WO 2025/013261 A1

［続葉有］

LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) 指定国(表示のない限り、全ての種類の広域保護が可能):ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, RU, TJ, TM), ヨーロッパ (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

添付公開書類:

一 国際調査報告(条約第21条(3))

(57) 要約:ＬＳＰプロキシ（１０）は、プログラムのコードを編集するための開発ツール（２０）に、機密コードが含まれないコードを送信する送信部（１１１）と、送信部（１１１）によって送信された機密コードが含まれないコードの編集に関するコマンドを開発ツール（２０）から受信する受信部（１１２）と、受信部（１１２）によって受信されたコマンドに基づいて、コードへのアクセスに関するチェックおよび編集に関する制御を実行するアクセスチェック部（１２１）と、を有する。

明　細　書

発明の名称 ： 制御装置、制御方法および制御プログラム

技術分野

[0001]　本発明は、制御装置、制御方法および制御プログラムに関する。

背景技術

[0002]　ソフトウェア開発の現状として、フリーランスやクラウドソーシングの活用、リモートワークの推進により、企業の管理区画外（自宅やシェアオフィス等）でのプログラムのコードの編集、デバッグ等の開発の遂行が増加している。例えば、通常はコードを編集するための編集装置（プログラムの編集、デバッグ等の開発を行うための開発ツール等）によって提供されるプログラミング支援機能を、編集装置とは独立したサーバから提供可能にするための通信処理等を行うコードの編集処理の公開プロトコル（ＬＳＰ：Language Server Protocol）が近年公開されている。ＬＳＰの活用例として、ＬＳＰによって、コードジャンプといったコードの定義の参照元に移動する定義移動や、コードの入力の途中で文脈に沿うようにコードを提供してコードを補完するコード補完等のプログラミング支援機能を、編集装置から独立したサーバ（以降、言語サーバ）からの提供を可能とする例が存在する（例えば、非特許文献１参照）。

先行技術文献

非特許文献

[0003] 非特許文献1：Language Server Protocol, [online], [2023年6月19日検索], インターネット＜https://microsoft.github.io/language-server-protocol/＞

発明の概要

発明が解決しようとする課題

[0004]　しかし、従来技術では、機密コードの漏洩のリスクを抑制しつつ、プログラムのコードの編集を行わせることができない場合がある。例えば、従来技

術では、企業によって保有されていたり、顧客から委託されたりするプログラムに含まれる機密コードが、リモートでコードの編集等を行う開発者によって閲覧されうるセキュリティ上の懸念がある。また、開発者が外部の攻撃者ではない開発プロジェクトに所属するリモート開発者であっても、開発者によるオペレーションミス等により、外部のＷｅｂサイトや外部のリポジトリに、機密コードが公開されてしまうコードの漏洩のリスクが高くなっている。

課題を解決するための手段

[0005] 上述の課題を解決するため、本発明は、プログラムのコードを編集するための編集装置に、機密コードが含まれないコードを送信する送信部と、前記送信部によって送信された前記機密コードが含まれないコードの編集に関するコマンドを前記編集装置から受信する受信部と、前記受信部によって受信された前記コマンドに基づいて、前記コードへのアクセスに関するチェックおよび前記編集に関する制御を実行する制御部と、を有する。

発明の効果

[0006] 本発明によれば、機密コードの漏洩のリスクを抑制しつつ、プログラムのコードの編集を行わせることができる。

図面の簡単な説明

[0007] [図1]図１は、第一の実施形態に係る制御システムの概要の一例について説明するための図である。

[図2]図２は、機密コードの一例を示す図である。

[図3]図３は、第一の実施形態に係る制御システムの概要の一例について説明するための図である。

[図4]図４は、第一の実施形態に係る制御システムの構成の一例を示す図である。

[図5]図５は、第一の実施形態に係る制御システムによって実行される処理の流れの一例を示すフローチャートである。

[図6]図６は、第二の実施形態に係る制御システムの概要の一例について説明

するための図である。

[図7]図７は、第二の実施形態に係る制御システムの概要の一例について説明するための図である。

[図8]図８は、第二の実施形態に係る制御システムの構成の一例を示す図である。

[図9]図９は、第二の実施形態に係る制御システムによって実行される処理の流れの一例を示すフローチャートである。

[図10]図１０は、第一の実施形態に係る制御システムと比較するための図である。

[図11]図１１は、第三の実施形態に係る制御システムの概要の一例について説明するための図である。

[図12]図１２は、第三の実施形態に係る制御システムの概要の一例について説明するための図である。

[図13]図１３は、第三の実施形態に係る制御システムの構成の一例を示す図である。

[図14]図１４は、第三の実施形態に係る制御システムによって実行される処理の流れの一例を示すフローチャートである。

[図15]図１５は、第三の実施形態に係る制御システムの変形例に係る制御システムの概要の一例について説明するための図である。

[図16]図１６は、制御プログラムを実行するコンピュータの構成の一例を示す図である。

[図17]図１７は、参考技術に係る制御システムの概要の一例について説明するための図である。

発明を実施するための形態

[0008] 以下、図面を用いて本発明の実施形態について説明するが、本発明は、以下の実施形態に限定されない。また、図面の記載は、同一部分には同一の符号を付しており、重複する説明は省略され、同様の機能を有する部材および同様の処理の説明も省略される。

[0009] ［０．参考技術］

［概要の一例］

本実施形態について説明する前に、図１７を用いて、参考技術に係る制御システム１＃の概要の一例について説明する。図１７は、参考技術に係る制御システムの概要の一例について説明するための図である。制御システム１＃は、コードの編集処理を行う。図１７に示される例では、制御システム１＃は、プログラミング支援機能（コードジャンプといった定義移動やコード補完等）を、開発ツール２０と、開発ツール２０とは独立した言語サーバ３０と、ＬＳＰ１０＃とを有する。ＬＳＰ１０＃は、言語サーバ３０のコードに対する編集処理を提供するための通信処理を行う編集処理の公開プロトコルである。

[0010] ［プログラミング支援機能の利用の一例］

まず、図１７を用いて、制御システム１＃によるプログラミング支援機能の利用の一例について説明する。図１７に示される例では、制御システム１＃のＬＳＰ１０＃は、あらかじめ、参照用のソースコードを開発ツール２０および言語サーバ３０に送信する。ソースコードとは、例えば、プログラミング言語等の人間が理解、記述しやすい言語やデータ形式によって書き記されたコンピュータプログラムのことをいう。

[0011] 図１７に示される例では、制御システム１＃の開発ツール２０は、開発者からOKClass2というコードにコードジャンプする編集操作を受け付ける（図１７の（０）OKClass2にジャンプする操作）。次に、ＬＳＰ１０＃は、例えば、開発ツール２０から受信した編集処理に関するコマンド（OKClass2というコードにコードジャンプする編集操作の通知（命令））を言語サーバ３０に送信する（図１７の（１）コマンド送信（OKClass2へのジャンプ通知））。次に、言語サーバ３０は、例えば、コマンドの内容に含まれるジャンプ先であるOKClass2というコードを解析し、このコードの位置情報を取得する（図１７の（２）ジャンプ先の位置情報を取得）。

[0012] 次に、ＬＳＰ１０＃は、例えば、言語サーバ３０から受信したコマンドに

対する返信を開発ツール２０に対して行う（図１７の（３）コマンド返信（OKClass2の位置情報））。次に、開発ツール２０は、例えば、ＬＳＰ１０＃から受信したコマンドに対する返信におけるOKClass2というコードの位置情報に基づき、コードジャンプする（図１７の（４）位置情報に基づきジャンプ）。次に、開発ツール２０は、例えば、コードジャンプしたジャンプ先を開発者に向けて表示する（図１７の（５）ジャンプ先を表示）。

[0013] ［課題の一例］

上述のように制御システム１＃がＬＳＰ１０＃を利用した場合、参考技術には、機密コードの漏洩のリスクを抑制しつつ、プログラムのコードの編集を行わせることができない場合がある。例えば、参考技術では、企業によって保有されていたり、顧客から委託されたりするプログラムに含まれる機密コードが、リモートでコードの編集等を行う開発者によって閲覧されうるセキュリティ上の懸念がある。また、開発者が外部の攻撃者ではない開発プロジェクトに所属するリモート開発者であっても、開発者によるオペレーションミス等により、外部のＷｅｂサイトや外部のリポジトリに、機密コードが公開されてしまうコードの漏洩のリスクが高くなっている。

[0014] 本実施形態は、上述の課題を解決するためになされたものであり、機密コードの漏洩のリスクを抑制しつつ、プログラムのコードの編集を行わせることができる。

[0015] ［１．第一の実施形態］

［概要の一例］

以下、図１～図４を用いて、第一の実施形態に係る制御システム１の概要の一例について説明する。図１および図３は、制御システムの概要の一例について説明するための図である。図２は、機密コードの一例を示す図である。図１～図３に示される例では、制御システム１は、参考技術に係るＬＳＰ１０＃の代わりに、ＬＳＰプロキシ１０を有する。

[0016] 図１～図４に示される例では、制御システム１は、ＬＳＰプロキシ１０によって、開発ツール２０に、機密コードが含まれないコードを送信し、機密

コードが含まれないコードの編集に関するコマンドを開発ツール２０から受信し、コマンドに基づいて、コードへのアクセスに関するチェックおよび編集に関する制御を実行する。これにより、制御システム１は、機密コードの漏洩のリスクを抑制しつつ、プログラムのコードの編集を行わせることができる。

[0017] ここで、ＬＳＰプロキシ１０とは、例えば、開発ツール２０と言語サーバ３０との中継機構であり、参考技術のＬＳＰ１０＃に対応するものをいう。開発ツール２０とは、例えば、プログラムのコードを編集するための編集装置のことをいう。コマンドとは、例えば、コードの編集に関するコマンドのことをいう。編集に関する制御とは、例えば、編集に関する操作や表示の制御のことをいい、例えば、編集に関する操作を禁止する制御のことをいう。機密コードとは、例えば、開発者に対して、関数の中身等を非公開にすべきコードのことをいう。

[0018] また、図１～図４に示される例では、制御システム１は、開発ツール２０とは独立した言語サーバ３０に機密コードが含まれるコードを送信し、コマンドに基づいて、言語サーバ３０の機密コードが含まれるコードに編集の内容を反映させる。これにより、制御システム１は、開発ツール２０側には機密コードを含めず、開発ツール２０および言語サーバ３０の双方にソースコード等のコードを保持させることができる。また、制御システム１は、開発ツール２０側に、機密コード以外のコードの編集のみを実施させ、ＬＳＰコマンドの中継機構であるＬＳＰプロキシ１０を介して、言語サーバ３０のソースコード等のコードに編集内容を反映させることができる。

[0019] なお、図１～図４に示される例は一例にすぎず、本実施形態に係る制御システム１は、図１～図４に示される例に限定されない。本実施形態に係る制御システム１は、ＬＳＰプロキシ１０によって、機密コードが含まれないコードを送信し、機密コードが含まれないコードの編集に関するコマンドを開発ツール２０から受信し、コマンドに基づいて、コードへのアクセスに関するチェックおよび編集に関する制御を実行すればよい。

[0020] ［機密コードの一例］

次に、図２を用いて機密コードの一例について説明する。プログラムの開発では、ソースコードを解析し、コンピュータが直接実行可能な形式のプログラムに変換するコンパイルを行い、プログラムを実行することが要求される。機密コードも、開発過程において、別コードから呼び出される場合がある。

[0021] そこで、図１～図４に示される例では、制御システム１は、機密コードを内部隠ぺいコードと完全隠ぺいコードとの二つに分類する。図２に示されるように、内部隠ぺいコードとは、関数の中身が非公開である一方で、関数の呼び出しは許可されているコードのことをいう。すなわち、内部隠ぺいコードとは、開発者がプログラミングをする上で、呼び出せるコードであり、開発者に対して、関数の存在そのものは秘匿にしないコードのことをいう。完全隠ぺいコードとは、関数の中身が非公開であることに加えて、関数の呼び出しが許可されていないコードのことをいう。すなわち、完全隠ぺいコードとは、開発者がプログラミングをする上で、呼び出せないコードであり、開発者に対し、関数の存在そのものも秘匿にするコードのことをいう。図１～図４に示される例では、制御システム１は、どちらのコードであっても、開発過程において実装内容を秘匿することを目的とする。

[0022] ［機密コードの分離およびアクセスチェックの一例］

図１および図３を用いて制御システム１による機密コードの分離およびアクセスチェックの一例について説明する。図３は、制御システム１による機密コードの分離およびアクセスチェック処理の流れの一例を示すシーケンス図でもある。

[0023] 図１に示される例では、まず、開発ツール２０は、開発者の端末のＧＵＩ（Grafic User Interface）およびインターネット等のネットワークを介して、開発者からOKClassに変数を追加する編集操作を受け付ける（図１の（０）OKClassに変数を追加する操作）。次に、開発ツール２０は、例えば、ＬＳＰプロキシ１０に、OKClassのファイルを変更することを通知するコマンドを

送信する（図１の（１）コマンド送信（OKClassのファイル変更通知））。次に、ＬＳＰプロキシ１０のアクセスチェック部１２１は、例えば、コマンドに基づいて、コードへのアクセスに関するチェックおよび編集に関する制御を実行する。一例として、アクセスチェック部１２１は、編集されるコードであるOKClassが公開可能コードであるかどうかで、アクセスの条件が満たされるかどうかをチェックする。ここで、アクセスの条件とは、例えば、コードにアクセス可能な条件のことをいう。一例として、アクセスの条件とは、アクセス先のコードが機密コードでないコード（例えば、後述のアクセスすることができるAllow list（許可リスト＝公開可能コードのリスト）として設定されているOKClassおよびOKClass2等）であることをいう。

[0024] 次に、ＬＳＰプロキシ１０は、例えば、アクセスの条件が満たされる場合、言語サーバ３０に、OKClassのファイルを変更することを通知するコマンドを送信する（図１の（２）コマンド送信（OKClassのファイル変更通知））。一例として、ＬＳＰプロキシ１０は、編集されるコードであるOKClassが公開可能コードであれば、アクセスの条件が満たされると判断し、言語サーバ３０のメモリ（記憶部）上のOKClass2に変数を追加する等、言語サーバ３０にOKClassのファイルを変更することを依頼する。これにより、アクセスチェック部１２１は、アクセス先のコードが公開可能コード等の機密コードでない場合のみ、プログラムのコードの編集を行わせるため、機密コードの漏洩のリスクを抑制しつつ、プログラムのコードの編集を行わせることができる。

[0025] 次に、開発ツール２０は、例えば、最初にＬＳＰプロキシ１０から送信された機密コードが含まれないソースコードを更新する（図１の（３－１）ソースコードの更新）。次に、開発ツール２０は、例えば、ＬＳＰプロキシ１０に、OKClassのファイルを保存することを通知するコマンドを送信する（図１の（３－２）コマンド送信（OKClassのファイル保存通知））。

[0026] 次に、アクセスチェック部１２１は、例えば、コマンドに基づいて、コードへのアクセスに関するチェックおよび編集に関する制御を実行する。一例として、アクセスチェック部１２１は、保存されるコードであるOKClassが公

開可能コードであるかどうかで、アクセスの条件が満たされるかどうかをチェックする。次に、ＬＳＰプロキシ１０は、例えば、アクセスの条件が満たされる場合、言語サーバ３０に、OKClassのファイルを保存することを通知するコマンドを送信する（図１の（３－３）コマンド送信（OKClassのファイル保存通知））。一例として、ＬＳＰプロキシ１０は、保存されるコードであるOKClassが公開可能コードであれば、アクセスの条件が満たされると判断し、言語サーバ３０に、OKClassのファイルを保存することを依頼する。これにより、ＬＳＰプロキシ１０は、アクセス先のコードが公開可能コード等の機密コードでない場合のみ、プログラムのコードの編集を行わせるため、機密コードの漏洩のリスクを抑制しつつ、プログラムのコードの編集を行わせることができる。

[0027] 次に、言語サーバ３０は、例えば、開発ツール２０側（ＬＳＰプロキシ１０）からのファイルを保存することを通知するコマンドに基づいて、開発ツール２０側のソースコードと同期して、言語サーバ３０のソースコードを更新する（図１の（４）ソースコードの更新（開発ツール側からのファイル保存通知により、ソースコードを更新する（開発ツール側のソースコードと同期する））。

[0028] 図３に示される例では、ＬＳＰプロキシ１０には、コードへのアクセスの条件が定義されたアクセス条件定義ファイルによって、アクセスすることができるAllow list（許可リスト＝公開可能コードのリスト）として、OKClassおよびOKClass2があらかじめ設定されている。また、図３に示される例では、ＬＳＰプロキシ１０には、アクセス条件定義ファイルによって、アクセスできない場合があるExecutable list（実行可能リスト＝内部隠ぺいコードのリスト）として、RefNGClassがあらかじめ設定されている。

[0029] 図３に示される例では、ＬＳＰプロキシ１０は、言語サーバ３０に参照される機密コードが含まれるソースコードから機密コードを分離する。これにより、ＬＳＰプロキシ１０は、機密コードが除かれたソースコードを作成し、機密コードが含まれないソースコードを開発ツール２０に送信する。次に

、開発ツール２０は、例えば、ＬＳＰプロキシ１０に、OKClassのファイルを変更することを通知するコマンドを送信する（図３の（１）コマンド送信（OKClassのファイル変更通知））。

[0030] 次に、ＬＳＰプロキシ１０は、例えば、コマンドに基づいて、コードへのアクセスに関するチェックおよび編集に関する制御を実行する。一例として、ＬＳＰプロキシ１０は、編集されるコードであるOKClassが公開可能コードであるかどうかで、アクセスの条件が満たされるかどうかをチェックする。次に、ＬＳＰプロキシ１０は、アクセスの条件が満たされる場合、言語サーバ３０にOKClassのファイルを変更することを通知するコマンドを送信する（図３の（２）コマンド送信（OKClassのファイル変更通知））。一例として、ＬＳＰプロキシ１０は、編集されるコードであるOKClassが公開可能コードであれば、アクセスの条件が満たされると判断し、言語サーバ３０のメモリ（記憶部）上のOKClass2に変数を追加する等、言語サーバ３０にOKClassのファイルの変更を依頼する。

[0031] これにより、ＬＳＰプロキシ１０は、アクセス先のコードが公開可能コード等の機密コードでない場合のみ、プログラムのコードの編集を行わせるため、機密コードの漏洩のリスクを抑制しつつ、プログラムのコードの編集を行わせ、言語サーバ３０のコードに反映させることができる。

[0032] 次に、開発ツール２０は、例えば、最初にＬＳＰプロキシ１０から送信された機密コードが含まれないソースコードを更新する（図３の（３－１）更新）。次に、開発ツール２０は、例えば、ＬＳＰプロキシ１０に、OKClassのファイルを保存することを通知するコマンドを送信する（図３の（３－２）コマンド送信（OKClassのファイル保存通知））。

[0033] 次に、ＬＳＰプロキシ１０は、例えば、コマンドに基づいて、コードへのアクセスに関するチェックおよび編集に関する制御を実行する。一例として、ＬＳＰプロキシ１０は、保存されるコードであるOKClassが公開可能コードであるかどうかでアクセスの条件が満たされるかどうかチェックする。次に、ＬＳＰプロキシ１０は、例えば、アクセスの条件が満たされる場合、言語

サーバ３０に、OKClassのファイルを保存することを通知するコマンドを送信する（図３の（３－３）コマンド送信（OKClassのファイル保存通知））。一例として、ＬＳＰプロキシ１０は、編集されるコードであるOKClassが公開可能コードであれば、アクセスの条件が満たされると判断し、言語サーバ３０にOKClassのファイルを変更することを依頼する。

[0034] これにより、ＬＳＰプロキシ１０は、アクセス先のコードが公開可能コード等の機密コードでない場合のみ、プログラムのコードの編集を行わせるため、機密コードの漏洩のリスクを抑制しつつ、プログラムのコードの編集を行わせ、言語サーバ３０のコードに反映させることができる。次に、言語サーバ３０は、例えば、開発ツール２０側（ＬＳＰプロキシ１０）からのファイルを保存することを通知するコマンドに基づいて、開発ツール２０側のソースコードと同期して言語サーバ３０のソースコードを更新する（図３の（４）ソースコードの更新）。

[0035] ［制御システムの構成の一例］

図４を用いて、第一の実施形態に係る制御システム１の構成の一例について説明する。図４は、第一の実施形態に係る制御システムの構成の一例を示す図である。図４に示される例では、制御システム１は、ＬＳＰプロキシ１０と、開発ツール２０と、言語サーバ３０とを有する。

[0036] ［ＬＳＰプロキシの構成の一例］

ＬＳＰプロキシ１０は、開発ツール２０に、機密コードが含まれないコードを送信し、機密コードが含まれないコードの編集に関するコマンドを開発ツール２０から受信し、コマンドに基づいて、コードへのアクセスに関するチェックおよび編集に関する制御を実行する制御装置である。図４に示される例では、制御装置は、編集処理の公開プロトコルであるＬＳＰプロキシ１０であるが、上述の機能を有するものであれば、特に限定されない。ただし、制御装置は、ＬＳＰプロキシ１０に適用されることにより、参画プロジェクト（ＰＪ：Project）によって採用されるプログラミング言語や開発環境によらず、ＰＪが変更されても、開発者自身の開発ツールを切り替えずに、プ

ログラミング支援機能を利用できる。図４に示される例では、ＬＳＰプロキシ１０は、通信部１１と、制御部１２と、記憶部１３とを有する。

[0037] ［通信部］

通信部１１は、開発ツール２０と言語サーバ３０との間で各種情報を送受信する。通信部１１は、例えば、ＮＩＣ（Network Interface Card）等によって実現される。図４に示される例では、通信部１１は、送信部１１１と、受信部１１２とを有する。

[0038] ［送信部］

送信部１１１は、開発ツール２０や言語サーバ３０に各種情報を送信する。例えば、送信部１１１は、プログラムのコードを編集するための開発ツール２０に、機密コードが含まれないコードを送信する。例えば、送信部１１１は、開発ツール２０とは独立した言語サーバ３０に機密コードが含まれるコードを送信してよい。一例として、送信部１１１は、機密コードが含まれないソースコードを開発ツール２０に送信し、機密コードが含まれるソースコードを言語サーバ３０に送信し、開発ツール２０および言語サーバ３０の双方にソースコードを保持させつつ、開発ツール２０には機密コードを保持させない。

[0039] ここで、開発者は、開発ツール２０に全てのソースコード等のコードが存在しなくても、コードが送信されていれば、開発ツール２０によって編集処理を実行することができる。このように、機密コードを物理的に開発ツール２０内に配置する必要がないため、送信部１１１は、機密コードが含まれないコードを開発ツール２０に送信することによって、機密コードを開発ツール２０から分離して機密コードの漏洩を抑制した上で、開発ツール２０による編集処理を可能とする。

[0040] ［受信部］

受信部１１２は、送信部１１１によって送信された機密コードが含まれないコードの編集に関するコマンドを開発ツール２０から受信する。例えば、受信部１１２は、インターネット等のネットワークを介して、開発ツール２

０の通信部２１から、コードの編集処理に関するコマンドを受信する。例えば、受信部１１２は、言語サーバ３０から、編集処理に関する返信を受信してよい。一例として、受信部１１２は、ネットワークを介して、言語サーバ３０の通信部３１からコマンドに対する返信を受信する。

[0041] ［制御部］

制御部１２は、ＬＳＰプロキシ１０全体を制御する。例えば、制御部１２は、各種処理手順が規定されたプログラムや、制御情報が格納された内部メモリを有する一以上のプロセッサによって構成され、プロセッサは、プログラムや内部メモリを用いて各処理を実行する。制御部１２は、例えば、ＣＰＵ（Central Processing Unit）、ＭＰＵ（Micro Processing Unit）、ＧＰＵ（Graphics Processing Unit）等の電子回路やＡＳＩＣ（Application Specific Integrated Circuit）、ＦＰＧＡ（Field Programmable Gate Array）等の集積回路によって実現される。図４に示される例では、制御部１２は、アクセスチェック部１２１を有する。

[0042] ［アクセスチェック部］

アクセスチェック部１２１は、受信部１１２によって受信されたコマンドに基づいて、コードへのアクセスに関するチェックおよび編集に関する制御を実行する。これにより、アクセスチェック部１２１は、アクセス先のコードが公開可能コード等の機密コードでない場合のみ、プログラムのコードの編集を行わせるため、機密コードの漏洩のリスクを抑制しつつ、プログラムのコードの編集を行わせることができる。

[0043] 例えば、アクセスチェック部１２１は、受信部１１２によって受信されたコマンドに基づいて、言語サーバ３０の機密コードが含まれるコードに編集の内容を反映させてよい。一例として、アクセスチェック部１２１は、編集されるコードであるOKClassが公開可能コードであれば、言語サーバ３０の記憶部３３上のOKClass2に変数を追加する等、言語サーバ３０にOKClassのファイルを変更することを依頼する。これにより、アクセスチェック部１２１は、機密コードの漏洩のリスクを抑制しつつ、プログラムのコードの編集を言

語サーバ３０のコードに反映させることができる。

[0044] ［記憶部］

記憶部１３は、各種情報を記憶する。例えば、記憶部１３は、機密コードが含まれるソースコードおよびＬＳＰプロキシ１０によって実行されるＯＳ（Operating System）や各種プログラム等を記憶する。記憶部１３は、例えば、ＨＤＤ（Hard Disk Drive）、ＳＳＤ（Solid State Drive）、光ディスク等の記憶装置やＲＡＭ（Random Access Memory）、フラッシュメモリ、ＮＶＳＲＡＭ（Non Volatile Static Random Access Memory）等の情報を書き換え可能な半導体メモリ等によって実現される。

[0045] ［開発ツールの構成の一例］

開発ツール２０は、プログラムのコードに対して編集処理を行うための編集装置である。開発ツール２０は、例えば、Vscode等によって実現される。図４に示される例では、編集装置は、プログラムの編集、デバッグ等の開発を行うための開発ツール２０であるが、上述の機能を有するものであれば、特に限定されない。図４に示される例では、開発ツール２０は、通信部２１と、制御部２２と、記憶部２３と、操作受付部２４と、表示部２５とを有する。

[0046] ［通信部］

通信部２１は、ＬＳＰプロキシ１０との間で各種情報を送受信する。通信部２１は、例えば、ＮＩＣ等によって実現される。図４に示される例では、通信部２１は、送信部２１１と、受信部２１２とを有する。

[0047] ［送信部］

送信部２１１は、ＬＳＰプロキシ１０に各種情報を送信する。例えば、送信部２１１は、コードに対する編集処理に関するコマンドをＬＳＰプロキシ１０に送信する。

[0048] ［受信部］

受信部２１２は、ＬＳＰプロキシ１０から各種情報を受信する。例えば、受信部２１２は、ＬＳＰプロキシ１０から機密コードが含まれないソースコ

ードを受信する。

[0049] ［制御部］

制御部２２は、開発ツール２０全体を制御する。例えば、制御部２２は、各種処理手順が規定されたプログラムや、制御情報が格納された内部メモリを有する一以上のプロセッサによって構成され、プロセッサは、プログラムや内部メモリを用いて各処理を実行する。制御部２２は、例えば、ＣＰＵ、ＭＰＵ、ＧＰＵ等の電子回路やＡＳＩＣ、ＦＰＧＡ等の集積回路によって実現される。

[0050] ［記憶部］

記憶部２３は、各種情報を記憶する。例えば、記憶部２３は、機密コードが含まれないソースコードおよび開発ツール２０によって実行されるＯＳや各種プログラム等を記憶する。記憶部２３は、例えば、ＨＤＤ、ＳＳＤ、光ディスク等の記憶装置やＲＡＭ、フラッシュメモリ、ＮＶＳＲＡＭ等の情報を書き換え可能な半導体メモリ等により実現される。

[0051] ［操作受付部］

操作受付部２４は、各種操作を受け付ける。例えば、操作受付部２４は、開発者の端末から、コードに対する編集処理に関するコマンドの操作を受け付ける。操作受付部２４は、例えば、インターネットを介して開発者の端末から編集処理に関する編集操作を受け付けるものや、ＧＵＩ等によって実現される。

[0052] ［表示部］

表示部２５は、各種情報を表示する。例えば、表示部２５は、コマンドに対する返信の内容を表示する。表示部２５は、例えば、端末のデスクトップ等によって実現される。

[0053] ［言語サーバの構成の一例］

言語サーバ３０は、編集装置とは独立したサーバである。図４に示される例では、言語サーバ３０は、開発ツール２０とは独立した編集処理を行うためのサーバである。図４に示される例では、言語サーバ３０は、通信部３１

と制御部３２と記憶部３３とを有する。

[0054] ［通信部］

通信部３１は、ＬＳＰプロキシ１０との間で各種情報を送受信する。通信部３１は、例えば、ＮＩＣ等によって実現される。図４に示される例では、通信部３１は、送信部３１１と、受信部３１２とを有する。

[0055] ［送信部］

送信部３１１は、ＬＳＰプロキシ１０に各種情報を送信する。例えば、送信部３１１は、コマンドに対する返信をＬＳＰプロキシ１０に送信する。

[0056] ［受信部］

受信部３１２は、ＬＳＰプロキシ１０から各種情報を受信する。一例として、受信部３１２は、ＬＳＰプロキシ１０から機密コードが含まれるソースコードを受信する。別の一例として、受信部３１２は、ＬＳＰプロキシ１０から許可リストに無いNGClass、RefNGClassおよびSecretFramework等の関数のスタックフレームが含まれるコマンドを受信する。

[0057] ［制御部］

制御部３２は、言語サーバ３０全体を制御する。例えば、制御部３２は、各種処理手順が規定されたプログラムや、制御情報が格納された内部メモリを有する一以上のプロセッサによって構成され、プロセッサは、プログラムや内部メモリを用いて各処理を実行する。制御部３２は、例えば、ＣＰＵ、ＭＰＵ、ＧＰＵ等の電子回路やＡＳＩＣ、ＦＰＧＡ等の集積回路によって実現される。

[0058] ［記憶部］

記憶部３３は、各種情報を記憶する。例えば、記憶部３３は、機密コードが含まれるソースコードおよび言語サーバ３０により実行されるＯＳや各種プログラム等を記憶する。記憶部３３は、例えば、ＨＤＤ、ＳＳＤ、光ディスク等の記憶装置やＲＡＭ、フラッシュメモリ、ＮＶＳＲＡＭ等の情報を書き換え可能な半導体メモリ等により実現される。

[0059] ［処理の流れの一例］

以下、図５を用いて、第一の実施形態に係る制御システム１のＬＳＰプロキシ１０によって実行される処理の流れの一例について説明する。図５は、第一の実施形態に係る制御システムによって実行される処理の流れの一例を示すフローチャートである。

[0060] ステップＳ１１において、送信部１１１は、プログラムのコードを編集するための開発ツール２０に、機密コードが含まれないコードを送信する。例えば、送信部１１１は、機密コードが含まれないソースコードを開発ツール２０に送信し、機密コードが含まれるソースコードを言語サーバ３０に送信する。

[0061] ステップＳ１２において、受信部１１２は、送信部１１１によって送信された機密コードが含まれないコードの編集に関するコマンドを開発ツール２０から受信する。例えば、受信部１１２は、インターネット等のネットワークを介して、開発ツール２０の通信部２１から、コードの編集処理に関するコマンドを受信する。

[0062] ステップＳ１３において、アクセスチェック部１２１は、受信部１１２によって受信されたコマンドに基づいて、コードへのアクセスに関するチェックおよび編集に関する制御を実行する。一例として、アクセスチェック部１２１は、編集されるコードであるOKClassが公開可能コードであれば、言語サーバ３０の記憶部３３上のOKClass2に変数を追加する等、言語サーバ３０にOKClassのファイルを変更することを依頼する。

[0063] ［２．第二の実施形態］

［概要の一例］

以下、図６および図７を用いて第二の実施形態に係る制御システム１Ｘの概要の一例について説明する。図６および図７は、第二の実施形態に係る制御システムの概要の一例について説明するための図である。図６および図７に示される例では、制御システム１Ｘは、第一の実施形態に係るＬＳＰプロキシ１０の代わりにＬＳＰプロキシ１０Ｘを有する。

[0064] 図６および図７に示される例では、制御システム１Ｘは、ＬＳＰプロキシ

１０Ｘによって、コマンドに、機密コードの情報漏洩につながる操作が含まれる場合、機密コードの情報漏洩につながる操作を禁止する。これにより、制御システム１Ｘは、編集処理に伴う開発ツール２０とのやりとりにおいて、機密コードの漏洩のリスクを抑制できる。

[0065] なお、図６および図７に示される例は一例にすぎず、本実施形態に係る制御システム１Ｘは、図６および図７に示される例に限定されない。本実施形態に係る制御システム１Ｘは、ＬＳＰプロキシ１０Ｘによって、コマンドに、機密コードの情報漏洩につながる操作が含まれる場合、機密コードの情報漏洩につながる操作を禁止すればよい。

[0066] ［機密コードの情報漏洩につながる操作の禁止の一例］

図６および図７を用いて、制御システム１Ｘによる機密コードの情報漏洩につながる操作の禁止の一例について説明する。図７は、制御システム１Ｘによる機密コードの情報漏洩につながる操作の禁止処理の流れの一例を示すシーケンス図でもある。

[0067] 図６に示される例では、まず、開発ツール２０は、開発者の端末のＧＵＩおよびインターネット等のネットワークを介して、開発者から内部隠ぺいコードであるRefNGClassへコードジャンプする編集操作を受け付ける（図６の（０）RefNGClassへジャンプする操作）。次に、開発ツール２０は、例えば、ＬＳＰプロキシ１０Ｘに、RefNGClassへコードジャンプすることを通知するコマンドを送信する（図６の（１）コマンド送信（RefNGClassへのジャンプ通知））。

[0068] 次に、ＬＳＰプロキシ１０Ｘのアクセスチェック部１２１Ｘは、例えば、開発ツール２０から受信したコマンドを解析し、コマンドに機密コードの情報漏洩につながる操作が含まれる場合、機密コードの情報漏洩につながる操作を禁止する。一例として、ＬＳＰプロキシ１０Ｘのアクセスチェック部１２１Ｘは、コマンドに内部隠ぺいコードであるRefNGClassへコードジャンプすることを通知する操作が含まれていると判断し、この操作を禁止する。ＬＳＰプロキシ１０Ｘは、言語サーバ３０に、RefNGClassへコードジャンプす

ることを通知しない。次に、ＬＳＰプロキシ１０Ｘは、例えば、開発ツール２０に、コードジャンプがＮＧ（No good）であることを通知する、コマンドに対する返信を行う（図６の（２）コマンド返信（ジャンプＮＧの通知））。開発ツール２０は、RefNGClassへコードジャンプすることを通知する操作が禁止されたことを表示する（図６の（３）RefNGClassへはジャンプ禁止の表示）。

[0069] 図７に示される例では、開発ツール２０は、ＬＳＰプロキシ１０ＸにRefNGClassへコードジャンプすることを通知するコマンドを送信する（図７の（１）コマンド送信（RefNGClassへのジャンプ通知））。例えば、ＬＳＰプロキシ１０Ｘのアクセスチェック部１２１Ｘは開発ツール２０から受信したコマンドを解析し、コマンドに機密コードの情報漏洩につながる操作が含まれる場合、機密コードの情報漏洩につながる操作を禁止する。

[0070] 図７に示される例では、ＬＳＰプロキシ１０Ｘのアクセスチェック部１２１Ｘは、コマンドに内部隠ぺいコードであるRefNGClassへコードジャンプすることを通知する操作が含まれていると判断する。また、ＬＳＰプロキシ１０Ｘのアクセスチェック部１２１Ｘは、例えば、RefNGClassについて、実行可能であるが、公開不可な内部隠ぺいコードなので、コードジャンプを禁止するように判断する。次に、ＬＳＰプロキシ１０Ｘは、例えば、開発ツール２０に、コードジャンプがＮＧであることを通知する、コマンドに対する返信を行う（図７の（２）コマンド返信（ジャンプＮＧの通知））。

[0071] ［制御システムの構成の一例］

図８を用いて、第二の実施形態に係る制御システム１Ｘの構成の一例について説明する。図８は、第二の実施形態に係る制御システムの構成の一例を示す図である。図８に示される例では、制御システム１Ｘは、第一の実施形態に係るＬＳＰプロキシ１０の代わりに、ＬＳＰプロキシ１０Ｘを有する。

[0072] ［ＬＳＰプロキシの構成の一例］

ＬＳＰプロキシ１０Ｘは、機密コードの情報漏洩につながる操作が含まれる場合、機密コードの情報漏洩につながる操作を禁止する。図８に示される

例では、ＬＳＰプロキシ１０Ｘは第一の実施形態における制御部１２の代わりに制御部１２Ｘを有する。この点以外、ＬＳＰプロキシ１０Ｘは第一の実施形態に係るＬＳＰプロキシ１０と同様である。

[0073] ［制御部］

制御部１２Ｘは、第一の実施形態におけるアクセスチェック部１２１の代わりに、アクセスチェック部１２１Ｘを有する。この点以外、制御部１２Ｘは、第一の実施形態における制御部１２と同様である。

[0074] ［アクセスチェック部］

アクセスチェック部１２１Ｘは、受信部１１２によって受信されたコマンドに、機密コードの情報漏洩につながる操作が含まれる場合、機密コードの情報漏洩につながる操作を禁止する。例えば、アクセスチェック部１２１Ｘは、機密コードの閲覧の操作を禁止する。これにより、アクセスチェック部１２１Ｘは、編集処理に伴う開発ツール２０とのやりとりにおいて、機密コードの漏洩のリスクを抑制することができる。

[0075] ここで、機密コードの情報漏洩につながる操作には、上述の例のような機密コードに直接的にアクセスする操作だけではなく、機密コードへの間接的なアクセスを可能にする操作も含まれうる。例えば、プログラムの実行時の情報を取得する機能（リフレクション）を有する関数（以降、リフレクション関数）へのアクセスを可能にする操作（例えば、リフレクション関数のインポート）によって、開発ツール２０の開発者は、プログラム実行中の関数名（クラスやメソッド）を取得できるため、リフレクション関数を利用すると機密コード閲覧が可能となる。

[0076] そこで、アクセスチェック部１２１Ｘは、例えば、受信部１１２によって受信されたコマンドに、機密コードの情報漏洩に間接的につながる操作が含まれる場合、機密コードの情報漏洩に間接的につながる操作を禁止してよい。一例として、アクセスチェック部１２１Ｘは、上述のリフレクション関数のような禁止関数へアクセスする操作を禁止する。これにより、アクセスチェック部１２１Ｘは、編集処理に伴う開発ツール２０とのやりとりにおいて

、機密コードの漏洩のリスクをより抑制することができる。

[0077] 例えば、アクセスチェック部１２１Ｘは、受信部１１２によって受信されたコマンドに、コードの関数が非公開の機密コードへのアクセスにつながる操作が含まれる場合、コードの関数が非公開の機密コードへのアクセスにつながる操作を禁止してよい。一例として、アクセスチェック部１２１Ｘは、コマンドに内部隠ぺいコードであるRefNGClassへコードジャンプすることを通知する操作が含まれていると判断した場合、この操作を禁止する。内部隠ぺいコード等、コードの関数が非公開の機密コードであっても、関数の呼び出しは許可されている機密コードは、操作によってはアクセスにつながるおそれがある。これに対し、アクセスチェック部１２１Ｘは、コードの関数が非公開の機密コードへのアクセスにつながる操作を禁止することによって、編集処理に伴う開発ツール２０とのやりとりにおいて、機密コードの漏洩のリスクをより抑制することができる。

[0078] ［処理の流れの一例］

以下、図９を用いて、第二の実施形態に係る制御システム１ＸのＬＳＰプロキシ１０Ｘによって実行される処理の流れの一例について説明する。図９は、第二の実施形態に係る制御システムによって実行される処理の流れの一例を示すフローチャートである。図９のステップＳ２１およびステップＳ２２は、図５のステップＳ１１およびステップＳ１２と同様である。

[0079] ステップＳ２３において、アクセスチェック部１２１Ｘは、受信部１１２によって受信されたコマンドに、機密コードの情報漏洩につながる操作が含まれるかどうかを判定する。アクセスチェック部１２１Ｘは、受信部１１２によって受信されたコマンドに、機密コードの情報漏洩につながる操作が含まれると判断した場合（ステップＳ２３のＹｅｓ）、ステップＳ２４において、機密コードの情報漏洩につながる操作を禁止する。例えば、アクセスチェック部１２１Ｘは、機密コードの閲覧の操作を禁止する。

[0080] アクセスチェック部１２１Ｘは、受信部１１２によって受信されたコマンドに、機密コードの情報漏洩につながる操作が含まれないと判断した場合（

ステップＳ２３のＮｏ）、ステップＳ２５において、編集に関する制御を実行する。一例として、アクセスチェック部１２１は、編集されるコードであるOKClassが公開可能コードであれば、言語サーバ３０の記憶部３３上のOKClass2に変数を追加する等、言語サーバ３０にOKClassのファイルを変更することを依頼する。

[0081] ［３．第三の実施形態］

［第一の実施形態との比較の一例］

第三の実施形態に係る制御システム１Ｙの概要の一例について説明する前に、図１０を用いて第三の実施形態に係る制御システム１Ｙと第一の実施形態に係る制御システム１とを比較する。図１０は、第一の実施形態に係る制御システムと比較するための図である。

[0082] 図１０に示される例では、第一の実施形態に係る制御システム１の開発ツール２０は、開発者の端末のＧＵＩおよびインターネット等のネットワークを介して、開発者から「com.xxx.」という入力操作を受け付ける（図１０の（０）com.xxx.と入力）。次に、開発ツール２０は、例えば、ＬＳＰプロキシ１０に「com.xxx.」に基づいてコードを補完する旨を通知するコマンドを送信する（図１０の（１）コマンド送信（コード補完の通知））。

[0083] 次に、ＬＳＰプロキシ１０のアクセスチェック部１２１は、例えば、コマンドに基づいて、コードへのアクセスに関するチェックおよび編集に関する制御を実行する。例えば、ＬＳＰプロキシ１０は、アクセスされるコードがOKClass等の公開可能コードであれば、アクセスの条件が満たされると判断し、言語サーバ３０に、「com.xxx.」に基づいてコードを補完することを通知するコマンドを送信する（図１０の（２）コマンド送信（コード補完の通知））。

[0084] 次に、言語サーバ３０は、例えば、ＬＳＰプロキシ１０から受信したコマンドに基づいてソースコードを解析し、完全隠ぺいコードのクラスも含まれる「com.xxx.」以下の補完候補のコードのクラス名の一覧をリストアップし、ＬＳＰプロキシ１０に返信する（図１０の（３）コマンド返信（補完候補

のクラス名一覧））。次に、ＬＳＰプロキシ１０は、例えば、言語サーバ３０から受信した、完全隠ぺいコードのクラスも含まれる「com.xxx.」以下の補完候補のコードのクラス名の一覧を開発ツール２０に返信する（図１０の（４）コマンド返信（補完候補のクラス名一覧））。次に、開発ツール２０は、例えば、ＬＳＰプロキシ１０から受信した完全隠ぺいコードのクラスも含まれる「com.xxx.」以下の補完候補のコードのクラス名の一覧に基づいて、補完候補を表示する（図１０の（５）補完候補の表示）。

[0085] ここで、コマンドの種類によっては、上述のように、コードの補完の要求（リクエスト）の返信内容の補完候補に、機密コードの関数の名前やパスが含まれる等、コマンドに対する返信内容に機密コードの情報も含まれてしまう場合も零ではない。また、補完候補に含まれる機密コードの名前やパスが分かれば、開発ツール２０側でも機密コードの解析（コンパイル等）の要求が可能となる。このため、たとえ開発ツール２０側に機密コードがなくても、補完候補のコードの名前やパスが分かれば、開発ツール２０側は、機密コードの内部情報（プロパティやメソッドの名前）を取得できるため、実装内容を把握できてしまう場合も零ではない。このように、第一の実施形態に係る制御システム１では、コマンドの返信内容から機密コードの実装内容が漏洩してしまうリスクも零ではない。

[0086] これに対し、第三の実施形態に係る制御システム１Ｙは、編集に関する返信を開発ツール２０に行い、編集に関する返信の内容に、機密コードの情報漏洩につながる内容が含まれる場合、当該内容を秘匿にするか、または削除する。これにより、制御システム１Ｙは、第一の実施形態に係る制御システム１と比較して、編集処理に伴う開発ツール２０とのやりとりにおいて、機密コードの漏洩のリスクをより抑制することができる。

[0087] ［概要の一例］

以下、図１１および図１２を用いて第三の実施形態に係る制御システム１Ｙの概要の一例について説明する。図１１および図１２は、第三の実施形態に係る制御システムの概要の一例について説明するための図である。図１１

および図１２に示される例では、制御システム１Ｙは、第一の実施形態に係るＬＳＰプロキシ１０の代わりにＬＳＰプロキシ１０Ｙを有する。ＬＳＰプロキシ１０Ｙは、返信内容秘匿部１２２をさらに有する。

[0088] なお、図１１および図１２に示される例は一例にすぎず、本実施形態に係る制御システム１Ｙは、図１１および図１２に示される例に限定されない。本実施形態に係る制御システム１Ｙは、ＬＳＰプロキシ１０Ｙによって、編集に関する返信を開発ツール２０に行い、編集に関する返信の内容に、機密コードの情報漏洩につながる内容が含まれる場合、当該内容を秘匿にするか、または削除すればよい。

[0089] ［機密コードの情報漏洩につながる内容の秘匿または削除の一例］

図１１および図１２を用いて、制御システム１Ｙによる機密コードの情報漏洩につながる内容の秘匿または削除の一例について説明する。図１２は、制御システム１Ｙによる機密コードの情報漏洩につながる内容の秘匿処理または削除処理の流れの一例を示すシーケンス図でもある。

[0090] 図１１に示される例では、中継機構であるＬＳＰプロキシ１０Ｙの返信内容秘匿部１２２はコマンドに対する返信内容を解析し、返信内容に機密コードの情報漏洩につながる内容が含まれる場合、返信内容を覆い隠すマスクにより返信内容を秘匿にするか、または削除する（以下、秘匿、または削除を「マスキング」ともいう）。例えば、返信内容秘匿部１２２は、返信内容にNGClass等の完全隠ぺいコードのクラス、関数の名前やパスが含まれる場合、NGClass等の完全隠ぺいコードのクラス、関数の名前やパスを補完候補から削除する（図１１の（４）返信内容のマスキング（機密コード（完全隠ぺいコード）のクラスは削除））。次に、ＬＳＰプロキシ１０Ｙは、例えば、完全隠ぺいコード等の機密コードがマスキングされた補完候補のコードのクラス名の一覧を通知する、コマンドに対する返信を行う（図１１の（５）コマンド返信（マスキングされた補完候補のクラス名一覧））。次に、開発ツール２０は、例えば、完全隠ぺいコード等の機密コードがマスキングされた補完候補のコードのクラス名の一覧を表示する（図１１の（６）補完候補の表示

）。

[0091] 図１１に示される例では、補完候補の中に、機密コードのクラスは表示されないため、開発者には、完全隠ぺいコード等の機密コードが存在することは分からない。このため、ＬＳＰプロキシ１０Ｙは、機密コードの露見の手がかりを開発者に見せることなく、コードの編集を行わせることができる。

[0092] 図１２に示される例では、ＬＳＰプロキシ１０Ｙの返信内容秘匿部１２２は、コマンドに対する返信内容を解析し、返信内容に、機密コードの情報漏洩につながる内容が含まれる場合、返信内容を覆い隠すマスクによって返信内容を秘匿にするか、または削除する。例えば、返信内容秘匿部１２２は、コマンドに対する返信内容に公開可能でも実行可能でもないNGClass等の完全隠ぺいコードのクラス、関数の名前やパスが含まれる場合、NGClass等の完全隠ぺいコードのクラス、関数の名前やパスを補完候補から削除する（図１２の（４）返信内容の秘匿（マスキング））。次に、ＬＳＰプロキシ１０Ｙは、例えば、完全隠ぺいコード等の機密コードがマスキングされた補完候補のコードのクラス名の一覧を通知する、コマンドに対する返信を行う（図１２の（５）コマンド返信（機密コード（完全隠ぺいコードのクラスは削除））。

[0093] ［制御システムの構成の一例］

図１３を用いて、第三の実施形態に係る制御システム１Ｙの構成の一例について説明する。図１３は、第三の実施形態に係る制御システムの構成の一例を示す図である。図１３に示される例では、制御システム１Ｙは、第一の実施形態に係るＬＳＰプロキシ１０の代わりに、ＬＳＰプロキシ１０Ｙを有する。

[0094] ［ＬＳＰプロキシの構成の一例］

ＬＳＰプロキシ１０Ｙは、編集に関する返信を開発ツール２０に行い、返信の内容に、機密コードの情報漏洩につながる内容が含まれる場合、当該内容を秘匿にするか、または削除する。図１３に示される例では、ＬＳＰプロキシ１０Ｙは、第一の実施形態における制御部１２の代わりに、制御部１２

Ｙを有する。この点以外、ＬＳＰプロキシ１０Ｙは、第一の実施形態に係るＬＳＰプロキシ１０と同様である。

[0095] ［制御部］

制御部１２Ｙは、返信内容秘匿部１２２をさらに有する。この点以外、制御部１２Ｙは、第一の実施形態における制御部１２と同様である。

[0096] ［返信内容秘匿部］

返信内容秘匿部１２２は、送信部１１１によって行われる編集に関する返信の内容に、機密コードの情報漏洩につながる内容が含まれる場合、当該内容を秘匿にするか、または削除する。これにより、返信内容秘匿部１２２は、編集処理に伴う開発ツール２０とのやりとりにおいて、機密コードの漏洩のリスクをより抑制することができる。

[0097] 例えば、返信内容秘匿部１２２は、送信部１１１によって行われる編集に関する返信の内容に、コードの関数が非公開であり、当該関数を呼び出すことができない機密コードの情報漏洩につながる内容が含まれる場合、当該内容を秘匿にするか、または削除してよい。一例として、返信内容秘匿部１２２は、コマンドに対する返信内容に公開可能でも実行可能でもないNGClass等の完全隠ぺいコードのクラス、関数の名前やパスが含まれる場合、NGClass等の完全隠ぺいコードのクラス、関数の名前やパスを補完候補から削除する。コードの関数が非公開であり、当該関数を呼び出すことができない機密コードの漏洩は特に漏洩のリスクが高いため、返信内容秘匿部１２２は、コードの関数が非公開であり、当該関数を呼び出すことができない機密コードの情報漏洩につながる内容を秘匿にするか、または削除することによって、機密コードの漏洩のリスクをさらに抑制できる。

[0098] ［処理の流れの一例］

図１４を用いて第三の実施形態に係る制御システム１ＹのＬＳＰプロキシ１０Ｙにより実行される処理の流れの一例について説明する。図１４は、第三の実施形態に係る制御システムによって実行される処理の流れの一例を示すフローチャートである。図１４のステップＳ３１～ステップＳ３３は、図

５のステップＳ１１～ステップＳ１３と同様である。

[0099] ステップＳ３４において、返信内容秘匿部１２２は、送信部１１１によって行われる編集に関する返信の内容に、機密コードの情報漏洩につながる内容が含まれるかどうかを判断する。返信内容秘匿部１２２は、返信の内容に機密コードの情報漏洩につながる内容が含まれると判断した場合（ステップＳ３４のＹｅｓ）、ステップＳ３５において、当該内容を秘匿にするか、または削除する。例えば、返信内容秘匿部１２２は、コマンドに対する返信内容に公開可能でも実行可能でもないNGClass等の完全隠ぺいコードのクラス、関数の名前やパスが含まれる場合、NGClass等の完全隠ぺいコードのクラス、関数の名前やパスを補完候補から削除する。

[0100] ステップＳ３５の後、または、返信内容秘匿部１２２によって、返信の内容に機密コードの情報漏洩につながる内容が含まれないと判断された場合（ステップＳ３４のＮｏ）、ステップＳ３６において、送信部１１１は、返信の内容を開発ツール２０に送信する。例えば、送信部１１１は、ステップＳ３５の後、NGClass等の完全隠ぺいコードのクラス、関数の名前やパスが補完候補から削除されたコマンドに対する返信を開発ツール２０に対して行う。

[0101] ［４．第三の実施形態の変形例］

第三の実施形態に係る制御システム１ＹのＬＳＰプロキシ１０Ｙは、実行環境との中継機構にも拡張可能である。実行環境とは、例えば、編集に関する処理をテストする装置であるテスト環境等、編集に関する処理を実行する装置のことをいう。制御システム１ＹのＬＳＰプロキシ１０Ｙの返信内容秘匿部１２２は、例えば、送信部１１１によって行われる実行環境による実行結果の内容に、機密コードの情報漏洩につながる内容が含まれる場合、当該内容を秘匿にするか、または削除する。

[0102] 以下、図１５を用いて、第三の実施形態の変形例に係る制御システム１Ｙの概要の一例について説明する。図１５は、第三の実施形態に係る制御システムの変形例に係る制御システムの概要の一例について説明するための図である。図１５に示される例では、言語サーバ３０は、実行環境４０に変更さ

れている。

[0103] 図１５に示される例では、まず、開発ツール２０は、ＬＳＰプロキシ１０Ｙにテスト実行のリクエストに関するコマンドを送信する（図１５の（１）コマンド送信（テスト実行のリクエスト））。次に、ＬＳＰプロキシ１０Ｙは、例えば、実行環境４０にテスト実行のリクエストに関するコマンドを送信する（図１５の（２）コマンド送信（テスト実行のリクエスト））。次に、実行環境４０は、例えば、ＬＳＰプロキシ１０Ｙから受信したテスト実行のリクエストに基づいて、テストを実行する。

[0104] 次に、実行環境４０は、例えば、テストが失敗し、エラーが発生した場合、テストが失敗した場合のテスト実行結果であるスタックトレース（失敗直前まで実行されていた関数やメソッド等の履歴）を通知する、コマンドに対する返信をＬＳＰプロキシ１０Ｙに行う。一例として、実行環境は、NGClass等の完全隠ぺいコードのクラスが含まれるテスト実行結果（スタックトレース）を通知する、コマンドに対する返信をＬＳＰプロキシ１０Ｙに対して行う（図１５の（３）コマンド返信（テスト実行結果））。

[0105] 次に、ＬＳＰプロキシ１０Ｙの返信内容秘匿部１２２は、例えば、送信部１１１によって行われる実行環境４０によるテスト実行結果の内容に、機密コードの情報漏洩につながる内容が含まれる場合、当該内容を秘匿にするか、または削除する。一例として、返信内容秘匿部１２２は、テスト実行結果の内容に、NGClass等の完全隠ぺいコードのクラスが含まれるので、NGClass等の完全隠ぺいコードのクラスを覆い隠すマスクによって、完全隠ぺいコードのクラスを秘匿にする（図１５の（４）返信内容の秘匿（マスキング））。

[0106] 次に、ＬＳＰプロキシ１０Ｙは、例えば、返信内容秘匿部１２２の秘匿処理により機密コードの情報漏洩につながる内容が秘匿にされたか、または削除されたテスト実行結果を通知する、コマンドに対する返信を開発ツール２０に対して行う。一例として、ＬＳＰプロキシ１０Ｙは、NGClass等の完全隠ぺいコードのクラスを覆い隠すマスクによって、完全隠ぺいコードのクラス

が秘匿にされたテスト実行結果を通知する、コマンドに対する返信を開発ツール２０に対して行う（図１５の（５）コマンドの送信（機密コード（完全隠ぺいコード）が秘匿されたテスト実行結果）。

[0107] 例えば、テストが失敗した場合のテスト実行結果のスタックトレースに機密コードが存在すれば、機密コードの関数やメソッド名が露見してしまう。これに対し、第三の実施形態の変形例に係る制御システム１ＹにおけるＬＳＰプロキシ１０Ｙの返信内容秘匿部１２２は、テスト実行結果のマスキングにも適用でき、実行環境４０とのやりとりからの情報漏洩の抑制にも貢献することができる。

[0108] また、返信内容秘匿部１２２は、上述の例のように、送信部１１１によって行われる実行環境４０による実行結果の内容に、機密コードの情報漏洩につながる内容が含まれる場合、当該内容を秘匿にすることによって、当該内容に機密コードが含まれることを開発者に示唆することができる。これにより、返信内容秘匿部１２２は、例えば、エラーの原因が開発者の編集処理等によるものではなく、秘匿にされた内容によるものであることを示唆することができる場合がある。このため、返信内容秘匿部１２２は、テスト環境等の実行環境４０による実行結果の内容を秘匿することによって、当該内容を削除する場合に比べて、開発者にエラーの原因を把握させやすくすることができる。

[0109] ［５．プログラム］

ＬＳＰプロキシ１０、ＬＳＰプロキシ１０ＸおよびＬＳＰプロキシ１０Ｙは、パッケージソフトウェアやオンラインソフトウェアとして制御プログラムをコンピュータにインストールさせることにより実装できる。例えば、コンピュータが制御プログラムを実行することにより、ＬＳＰプロキシ１０、ＬＳＰプロキシ１０ＸおよびＬＳＰプロキシ１０Ｙとして機能させることができる。

[0110] 図１６は、制御プログラムを実行するコンピュータの構成の一例を示す図である。図１６に示される例では、コンピュータ１０００は、メモリ１０１

０とＣＰＵ１０２０とハードディスクドライブインタフェース１０３０とディスクドライブインタフェース１０４０とシリアルポートインタフェース１０５０とビデオアダプタ１０６０とネットワークインタフェース１０７０とを有する。これらの各部は、バス１０８０によって接続される。

[0111] メモリ１０１０は、ＲＯＭ（Read Only Memory）１０１１およびＲＡＭ１０１２を含む。ＲＯＭ１０１１は、例えば、ＢＩＯＳ（Basic Input Output System）等のブートプログラムを記憶する。ハードディスクドライブインタフェース１０３０は、ハードディスクドライブ１０９０に接続される。ディスクドライブインタフェース１０４０は、ディスクドライブ１１００に接続される。例えば、磁気ディスクや光ディスク等の着脱可能な記憶媒体が、ディスクドライブ１１００に挿入される。シリアルポートインタフェース１０５０は、例えば、マウス１１１０、キーボード１１２０に接続される。ビデオアダプタ１０６０は、例えば、ディスプレイ１１３０に接続される。

[0112] ハードディスクドライブ１０９０は、例えば、ＯＳ１０９１、アプリケーションプログラム１０９２、プログラムモジュール１０９３およびプログラムデータ１０９４を記憶する。すなわち、ＬＳＰプロキシ１０によって実行される各処理を規定するプログラムは、コンピュータにより実行可能なコードが記述されたプログラムモジュール１０９３として実装される。プログラムモジュール１０９３は、例えば、ハードディスクドライブ１０９０に記憶される。例えば、ＬＳＰプロキシ１０における機能構成と同様の処理を実行するためのプログラムモジュール１０９３が、ハードディスクドライブ１０９０に記憶される。なお、ハードディスクドライブ１０９０は、ＳＳＤによって代替されてよい。

[0113] また、上述の実施形態の処理において用いられるデータは、プログラムデータ１０９４として、例えば、メモリ１０１０やハードディスクドライブ１０９０に記憶される。そして、ＣＰＵ１０２０が、メモリ１０１０やハードディスクドライブ１０９０に記憶されたプログラムモジュール１０９３やプログラムデータ１０９４を、必要に応じてＲＡＭ１０１２に読み出して実行

する。

[0114] なお、プログラムモジュール１０９３やプログラムデータ１０９４は、ハードディスクドライブ１０９０に記憶される場合に限定されず、例えば、着脱可能な記憶媒体に記憶され、ディスクドライブ１１００等を介してＣＰＵ１０２０によって読み出されてもよい。あるいは、プログラムモジュール１０９３およびプログラムデータ１０９４は、ネットワーク（ＬＡＮ（Local Area Network）、ＷＡＮ（Wide Area Network）等）を介して接続される他のコンピュータに記憶されてもよい。そして、プログラムモジュール１０９３およびプログラムデータ１０９４は、他のコンピュータから、ネットワークインタフェース１０７０を介してＣＰＵ１０２０によって読み出されてもよい。

符号の説明

[0115] １　制御システム

１０　ＬＳＰプロキシ（制御装置）

１１、２１、３１　通信部

１２、２２、３２　制御部

１３、２３、３３　記憶部

２０　開発ツール（編集装置）

２４　操作受付部

２５　表示部

３０　言語サーバ

１１１、２１１、３１１　送信部

１１２、２１２、３１２　受信部

１２１　アクセスチェック部

請求の範囲

[請求項1] プログラムのコードを編集するための編集装置に、機密コードが含まれないコードを送信する送信部と、

前記送信部によって送信された前記機密コードが含まれないコードの編集に関するコマンドを前記編集装置から受信する受信部と、

前記受信部によって受信された前記コマンドに基づいて、前記コードへのアクセスに関するチェックおよび前記編集に関する制御を実行する制御部と、

を有することを特徴とする制御装置。

[請求項2] 前記送信部は、前記編集装置とは独立したサーバに前記機密コードが含まれるコードを送信し、

前記制御部は、前記受信部によって受信された前記コマンドに基づいて、前記サーバの前記機密コードが含まれるコードに前記編集の内容を反映させることを特徴とする請求項１に記載の制御装置。

[請求項3] 前記制御部は、前記受信部によって受信された前記コマンドに、前記機密コードの情報漏洩につながる操作が含まれる場合、前記機密コードの情報漏洩につながる操作を禁止することを特徴とする請求項１または２に記載の制御装置。

[請求項4] 前記制御部は、前記受信部によって受信された前記コマンドに、前記機密コードへのアクセスに間接的につながる操作が含まれる場合、前記機密コードへのアクセスに間接的につながる操作を禁止することを特徴とする請求項３に記載の制御装置。

[請求項5] 前記送信部は、前記編集に関する返信を前記編集装置に行い、

前記制御部は、前記送信部によって行われる前記編集に関する返信の内容に、前記機密コードの情報漏洩につながる内容が含まれる場合、当該内容を秘匿にするか、または削除することを特徴とする請求項１に記載の制御装置。

[請求項6] 前記制御部は、前記送信部によって行われる前記編集に関する返信

の内容に、前記コードの関数が非公開であり、当該関数を呼び出すことができない前記機密コードの情報漏洩につながる内容が含まれる場合、当該内容を秘匿にするか、または削除することを特徴とする請求項５に記載の制御装置。

[請求項7] 制御装置によって実行される制御方法であって、

プログラムのコードを編集するための編集装置に、機密コードが含まれないコードを送信する送信工程と、

前記送信工程によって送信された前記機密コードが含まれないコードの編集に関するコマンドを前記編集装置から受信する受信工程と、

前記受信工程によって受信された前記コマンドに基づいて、前記コードへのアクセスに関するチェックおよび前記編集に関する制御を実行する制御工程と、

を含むことを特徴とする制御方法。

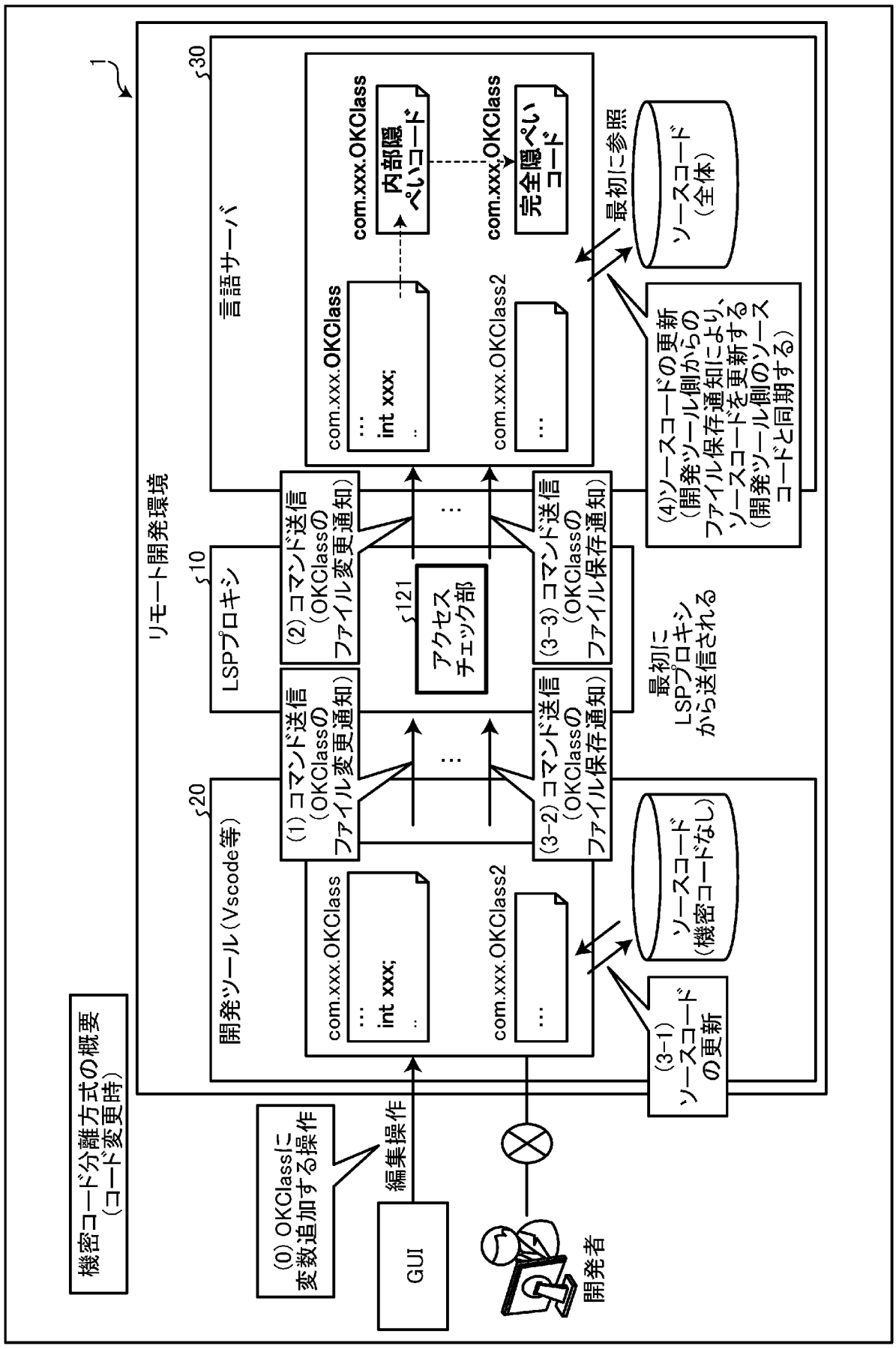
[請求項8] プログラムのコードを編集するための編集装置に、機密コードが含まれないコードを送信する送信ステップと、

前記送信ステップによって送信された前記機密コードが含まれないコードの編集に関するコマンドを前記編集装置から受信する受信ステップと、

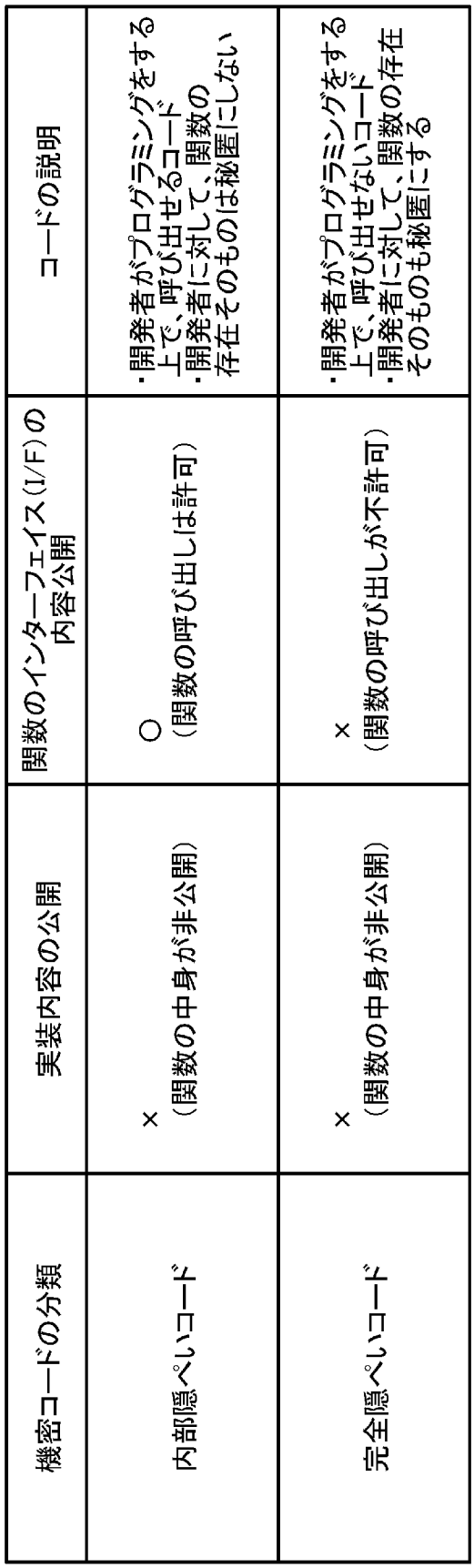
前記受信ステップによって受信された前記コマンドに基づいて、前記コードへのアクセスに関するチェックおよび前記編集に関する制御を実行する制御ステップと、

をコンピュータに実行させることを特徴とする制御プログラム。

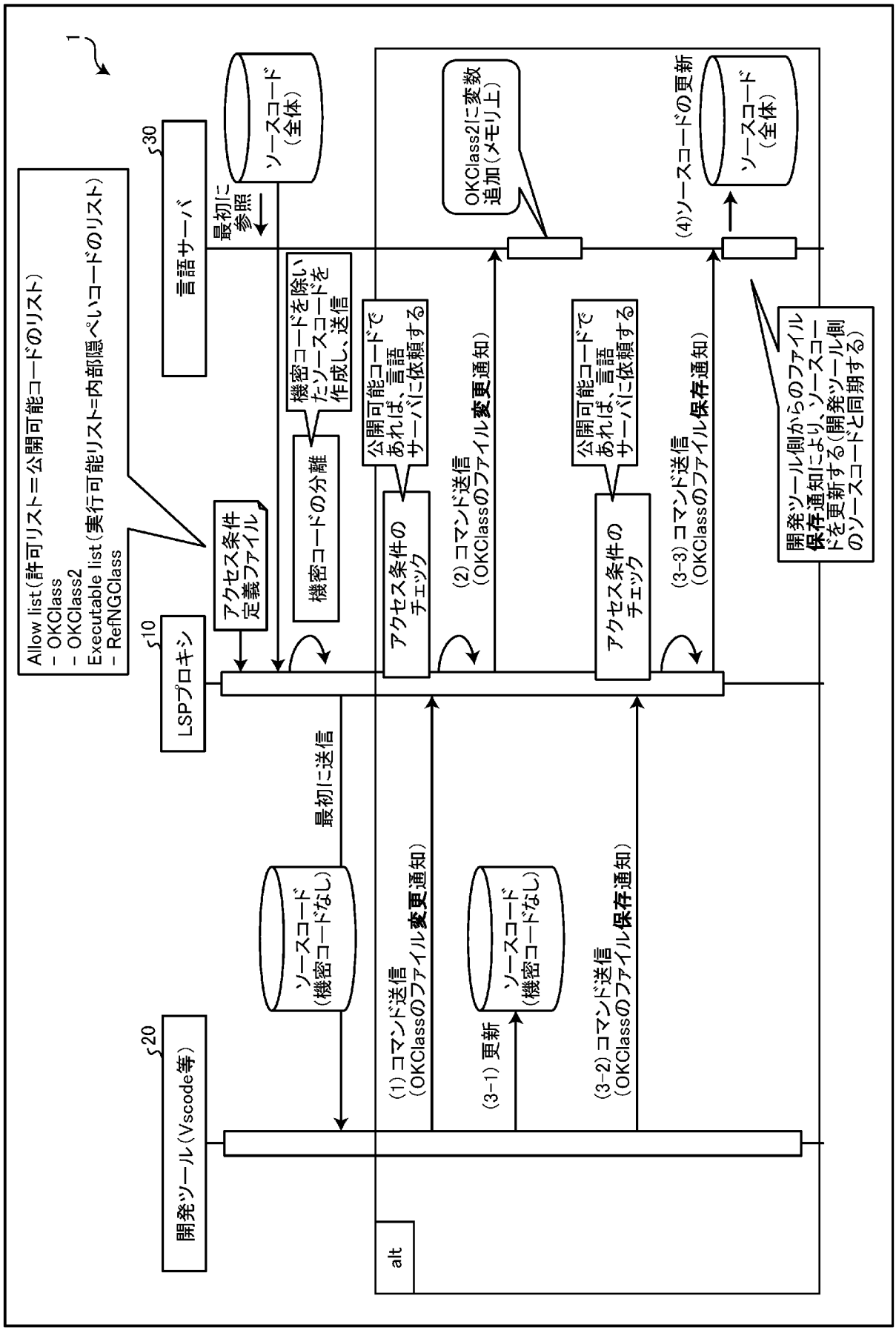
[図1]


機密コード分離方式の概要
（コード変更時）
1
リモート開発環境
開発ツール（Vscode等）
20
LSPプロキシ
10
言語サーバ
30
com.xxx.OKClass
int xxx;
com.xxx.OKClass2
com.xxx.OKClass
内部隠ぺいコード
com.xxx.OKClass
完全隠ぺいコード
(0) OKClassに
変数追加する操作
編集操作
GUI
開発者
(1) コマンド送信
（OKClassの
ファイル変更通知）
(2) コマンド送信
（OKClassの
ファイル変更通知）
121
アクセス
チェック部
(3-2) コマンド送信
（OKClassの
ファイル保存通知）
(3-3) コマンド送信
（OKClassの
ファイル保存通知）
最初に
LSPプロキシ
から送信される
(3-1)
ソースコード
の更新
ソースコード
（機密コードなし）
最初に参照
ソースコード
（全体）
(4)ソースコードの更新
（開発ツール側からの
ファイル保存通知により、
ソースコードを更新する
（開発ツール側のソース
コードと同期する）

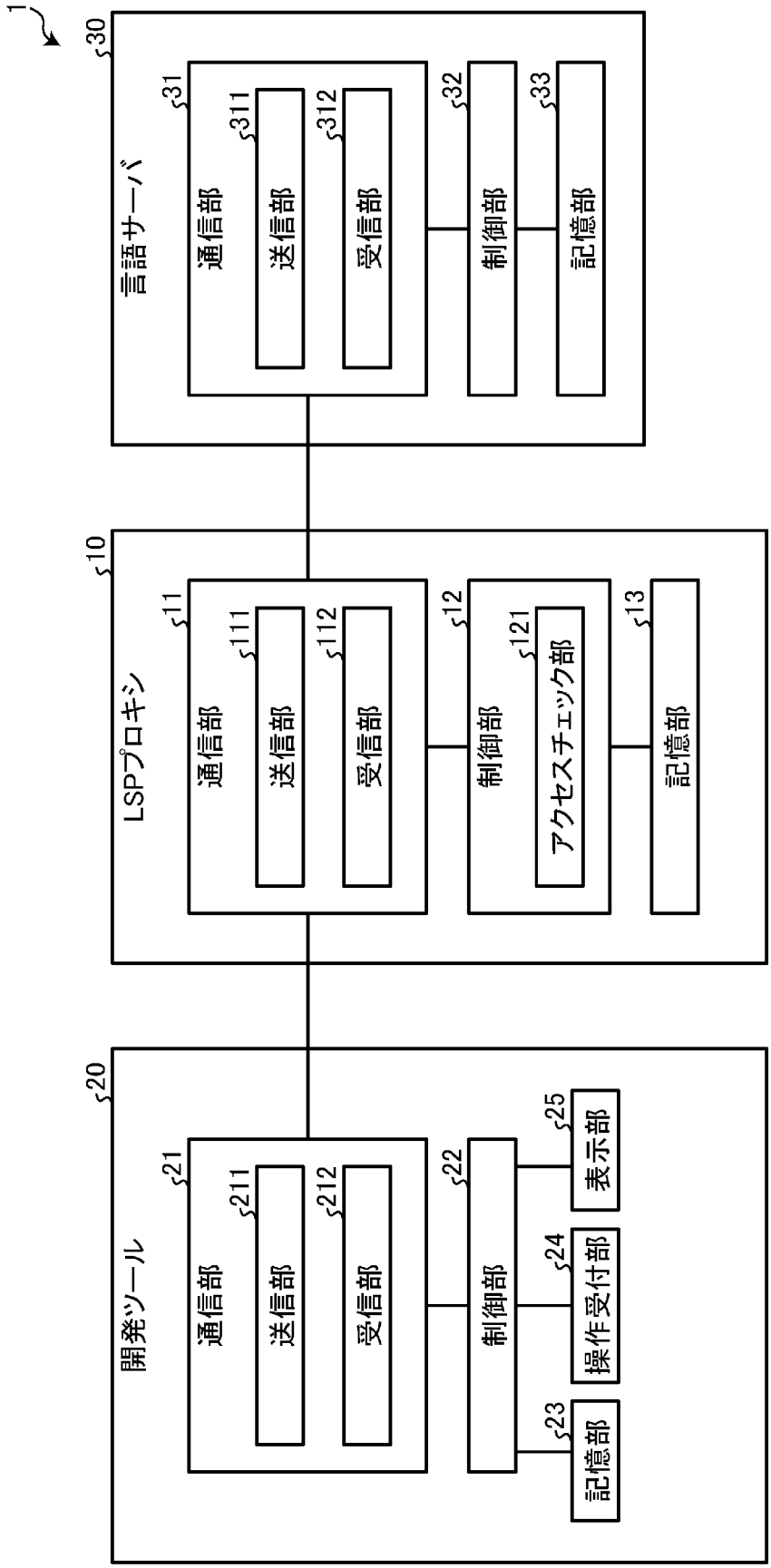
[図2]


機密コードの分類
実装内容の公開
関数のインターフェイス(I/F)の内容公開
コードの説明
内部隠ぺいコード
×
（関数の中身が非公開）
○
（関数の呼び出しは許可）
・開発者がプログラミングをする上で、呼び出せるコード
・開発者に対して、関数の存在そのものは秘匿にしない
完全隠ぺいコード
×
（関数の中身が非公開）
×
（関数の呼び出しが不許可）
・開発者がプログラミングをする上で、呼び出せないコード
・開発者に対して、関数の存在そのものも秘匿にする

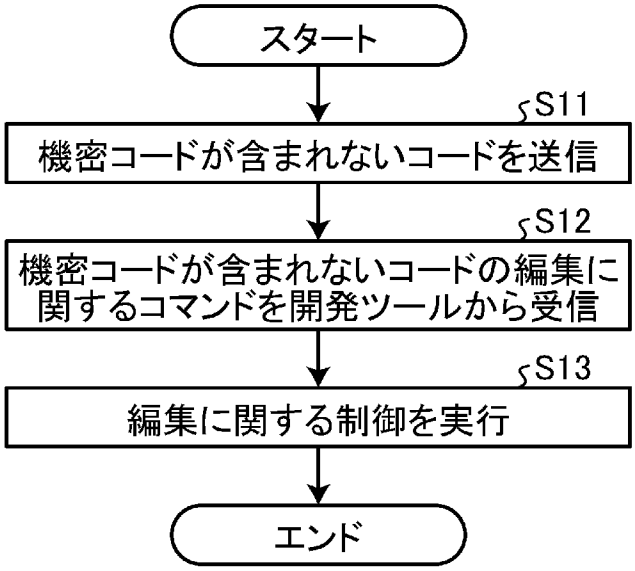
[図3]


1
Allow list（許可リスト＝公開可能コードのリスト）
- OKClass
- OKClass2
Executable list（実行可能リスト=内部隠ぺいコードのリスト）
- RefNGClass
10
LSPプロキシ
20
開発ツール（Vscode等）
30
言語サーバ
ソースコード（全体）
最初に参照
アクセス条件定義ファイル
機密コードの分離
機密コードを除いたソースコードを作成し、送信
最初に送信
ソースコード（機密コードなし）
公開可能コードであれば、言語サーバに依頼する
アクセス条件のチェック
(1) コマンド送信（OKClassのファイル変更通知）
(2) コマンド送信（OKClassのファイル変更通知）
OKClass2に変数追加（メモリ上）
(3-1) 更新
(3-2) コマンド送信（OKClassのファイル保存通知）
(3-3) コマンド送信（OKClassのファイル保存通知）
(4)ソースコードの更新
開発ツール側からのファイル保存通知により、ソースコードを更新する（開発ツール側のソースコードと同期する）
alt

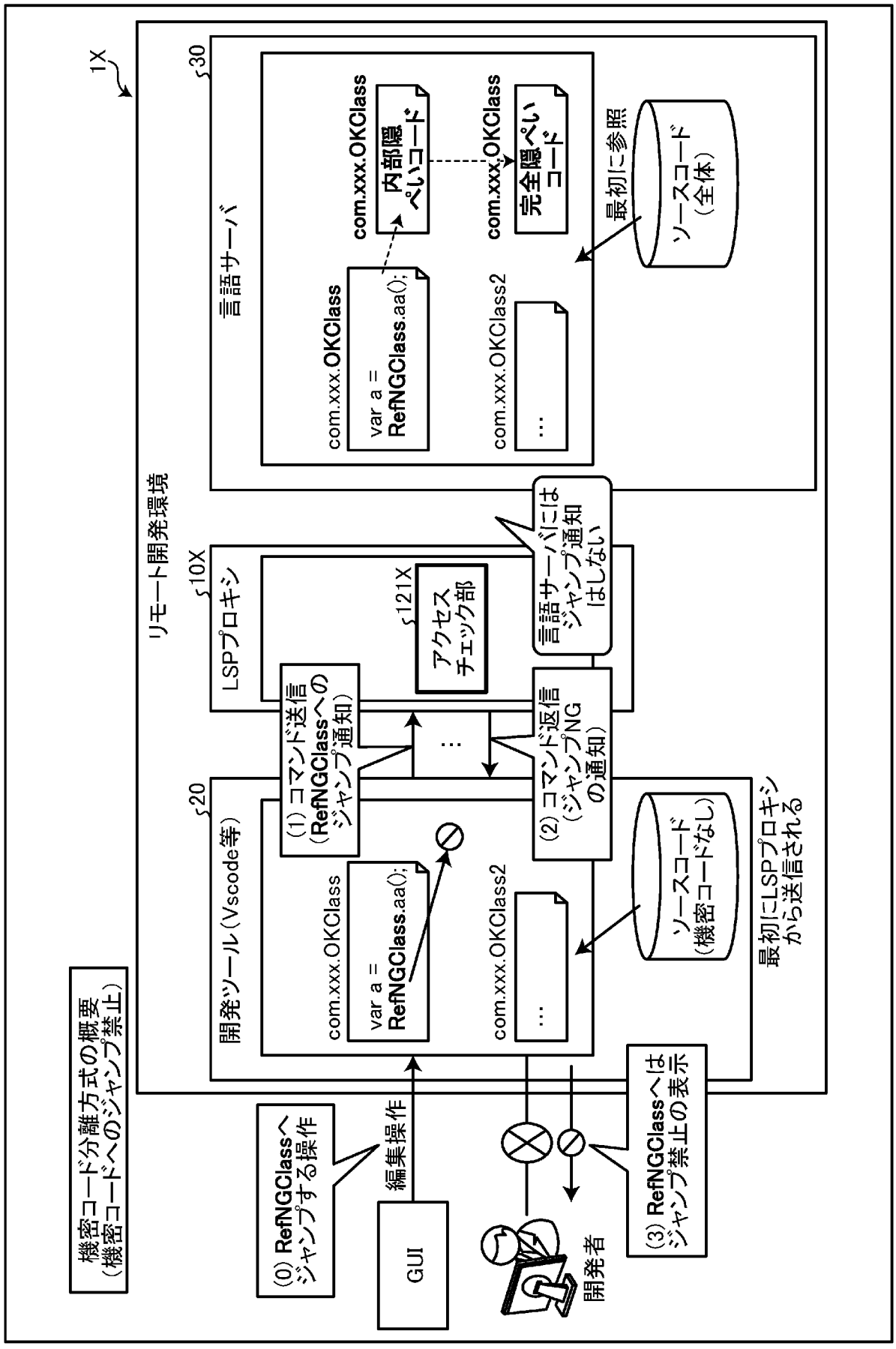
[図4]


1
30
言語サーバ
31
通信部
311
送信部
312
受信部
32
制御部
33
記憶部
10
LSPプロキシ
11
通信部
111
送信部
112
受信部
12
制御部
121
アクセスチェック部
13
記憶部
20
開発ツール
21
通信部
211
送信部
212
受信部
22
制御部
23
記憶部
24
操作受付部
25
表示部

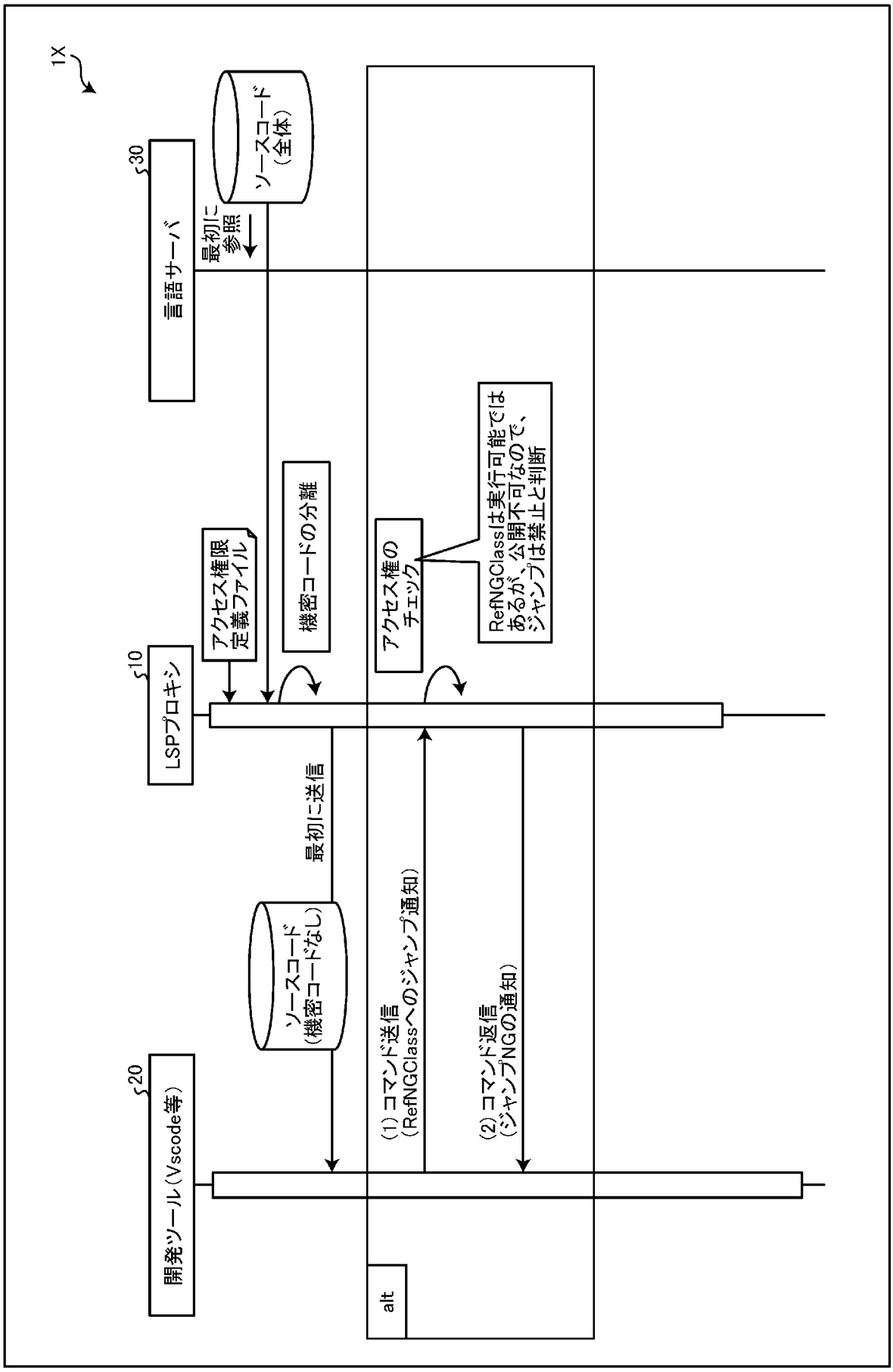
[図5]


スタート
S11
機密コードが含まれないコードを送信
S12
機密コードが含まれないコードの編集に関するコマンドを開発ツールから受信
S13
編集に関する制御を実行
エンド

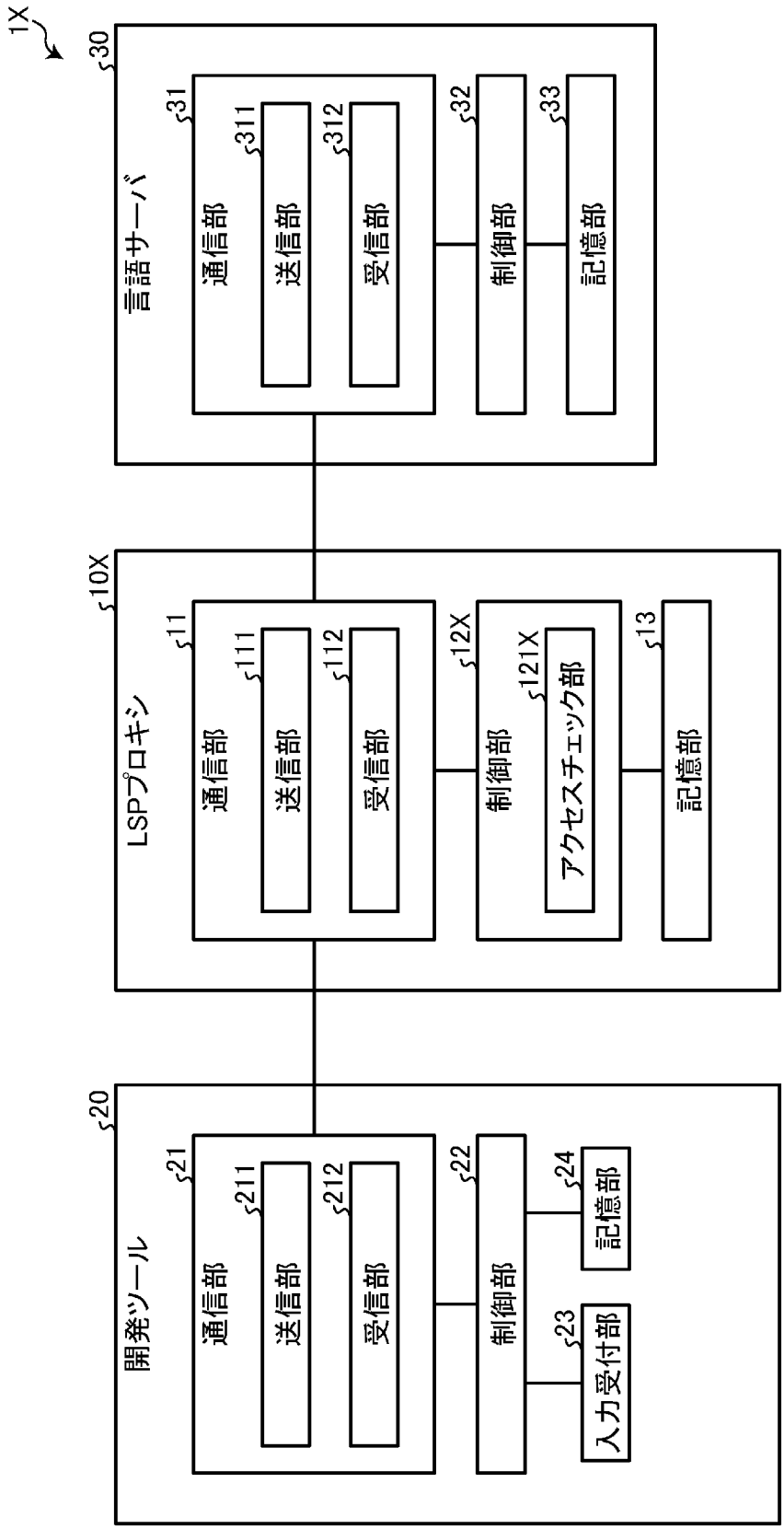
[図6]


1X
リモート開発環境
機密コード分離方式の概要
(機密コードへのジャンプ禁止)
30
言語サーバ
com.xxx.OKClass
var a =
RefNGClass.aa();
com.xxx.OKClass2
...
com.xxx.OKClass
内部隠ぺいコード
com.xxx.OKClass
完全隠ぺいコード
最初に参照
ソースコード
(全体)
10X
LSPプロキシ
121X
アクセスチェック部
言語サーバにはジャンプ通知はしない
20
開発ツール(Vscode等)
(1) コマンド送信
(RefNGClassへのジャンプ通知)
(2) コマンド返信
(ジャンプNGの通知)
com.xxx.OKClass
var a =
RefNGClass.aa();
com.xxx.OKClass2
...
ソースコード
(機密コードなし)
最初にLSPプロキシから送信される
(0) RefNGClassへジャンプする操作
編集操作
GUI
開発者
(3) RefNGClassへはジャンプ禁止の表示

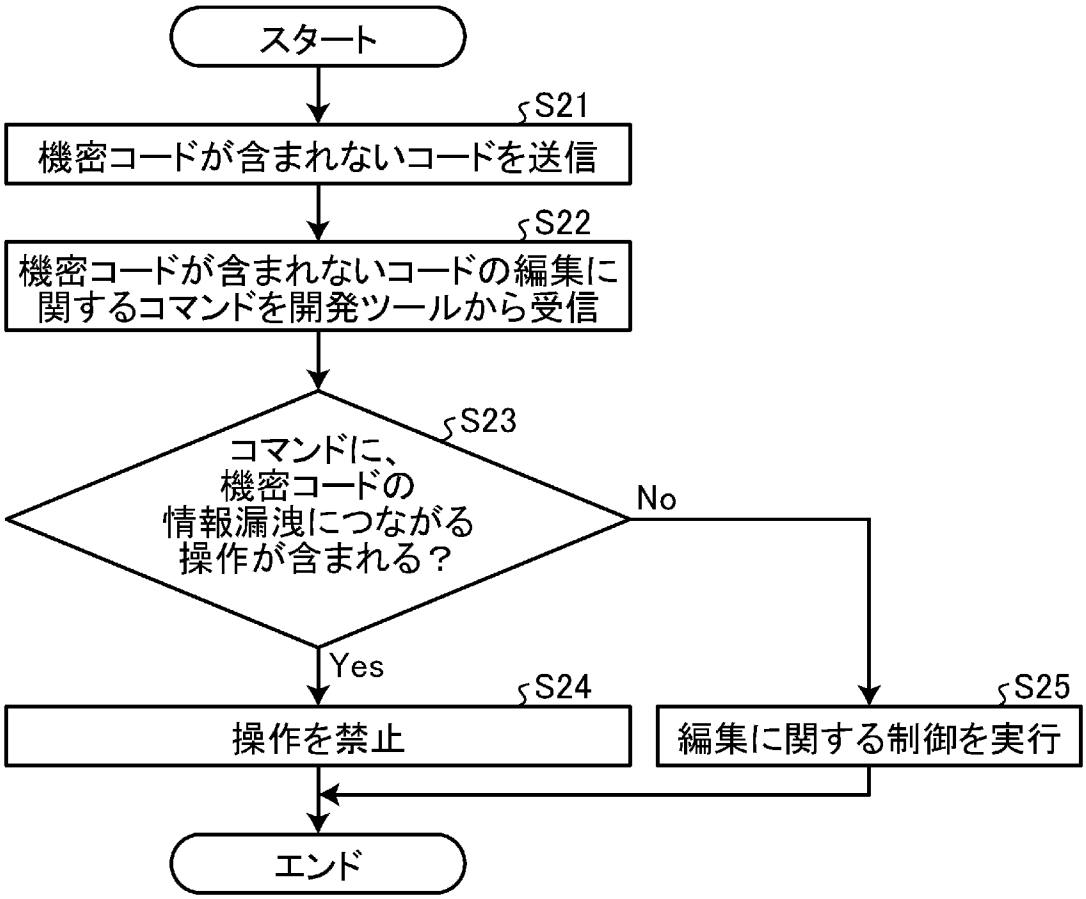
[図7]


1X
言語サーバ
30
ソースコード
（全体）
最初に
参照
LSPプロキシ
10
アクセス権限
定義ファイル
機密コードの分離
アクセス権の
チェック
RefNGClassは実行可能では
あるが、公開不可なので、
ジャンプは禁止と判断
最初に送信
ソースコード
（機密コードなし）
開発ツール（Vscode等）
20
(1) コマンド送信
(RefNGClassへのジャンプ通知)
(2) コマンド返信
(ジャンプNGの通知)
alt

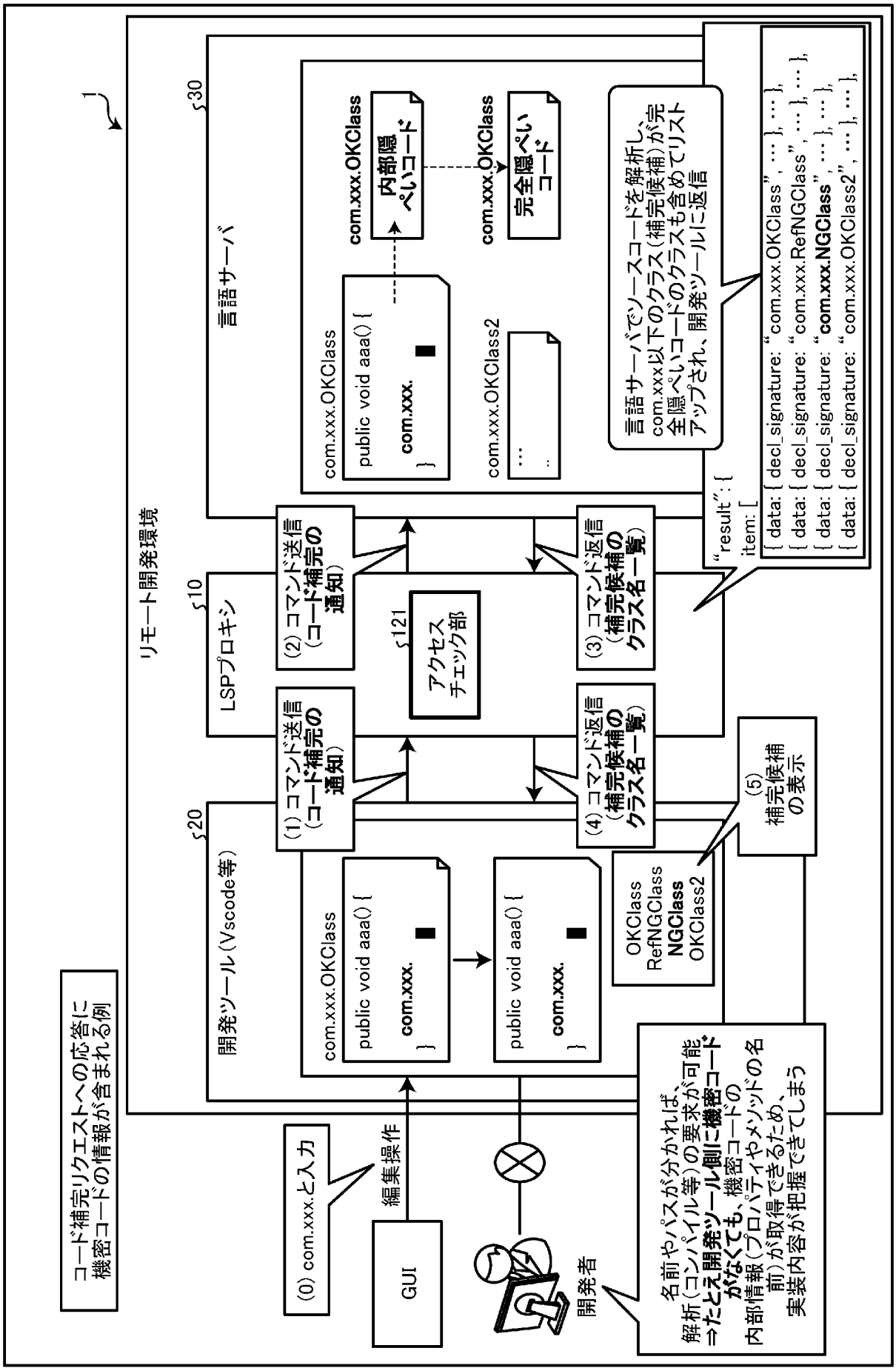
[図8]


1X
30
言語サーバ
31
通信部
311
送信部
312
受信部
32
制御部
33
記憶部
10X
LSPプロキシ
11
通信部
111
送信部
112
受信部
12X
制御部
121X
アクセスチェック部
13
記憶部
20
開発ツール
21
通信部
211
送信部
212
受信部
22
制御部
23
入力受付部
24
記憶部

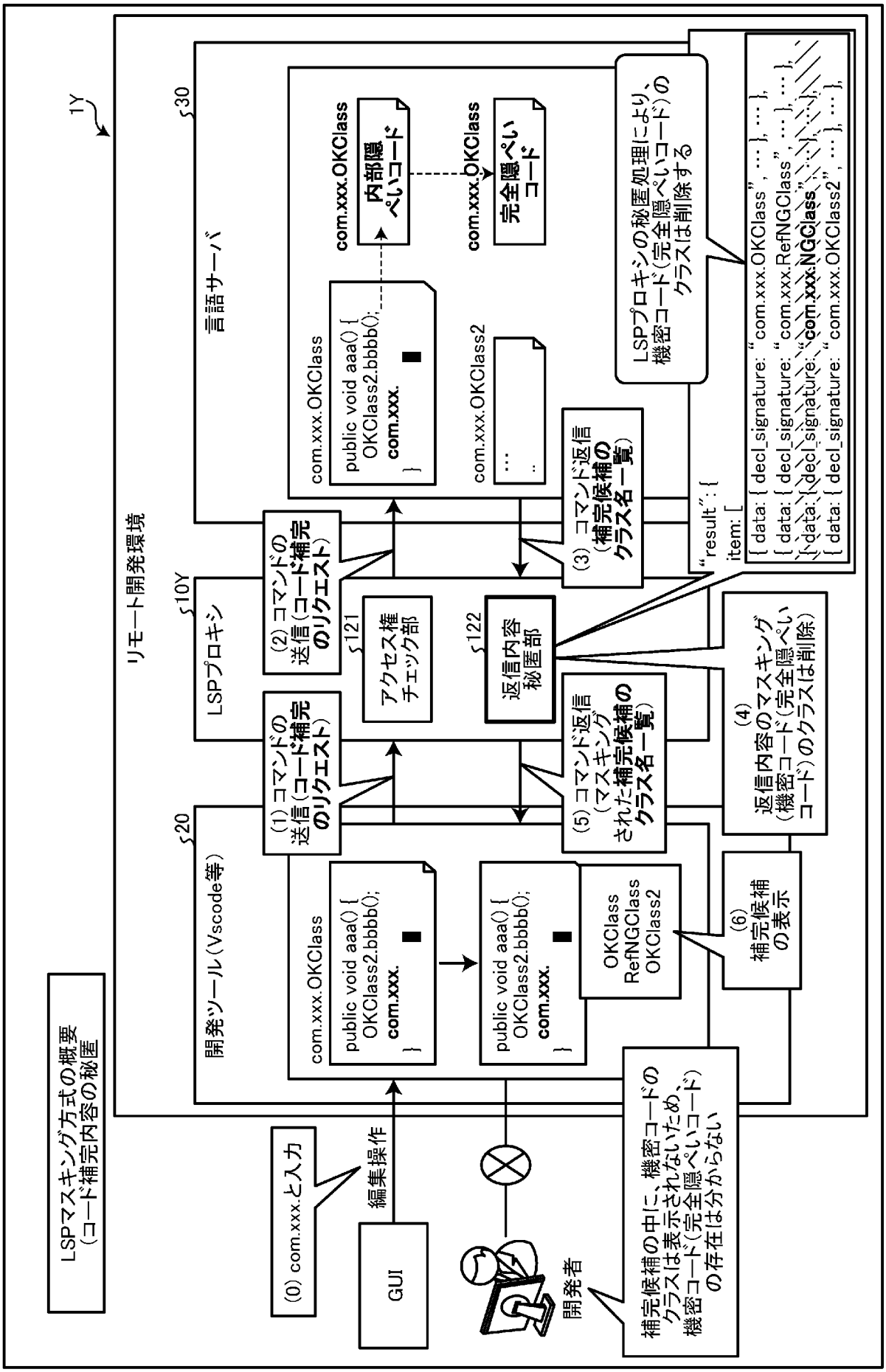
[図9]


スタート
S21
機密コードが含まれないコードを送信
S22
機密コードが含まれないコードの編集に関するコマンドを開発ツールから受信
S23
コマンドに、機密コードの情報漏洩につながる操作が含まれる？
No
Yes
S24
操作を禁止
S25
編集に関する制御を実行
エンド

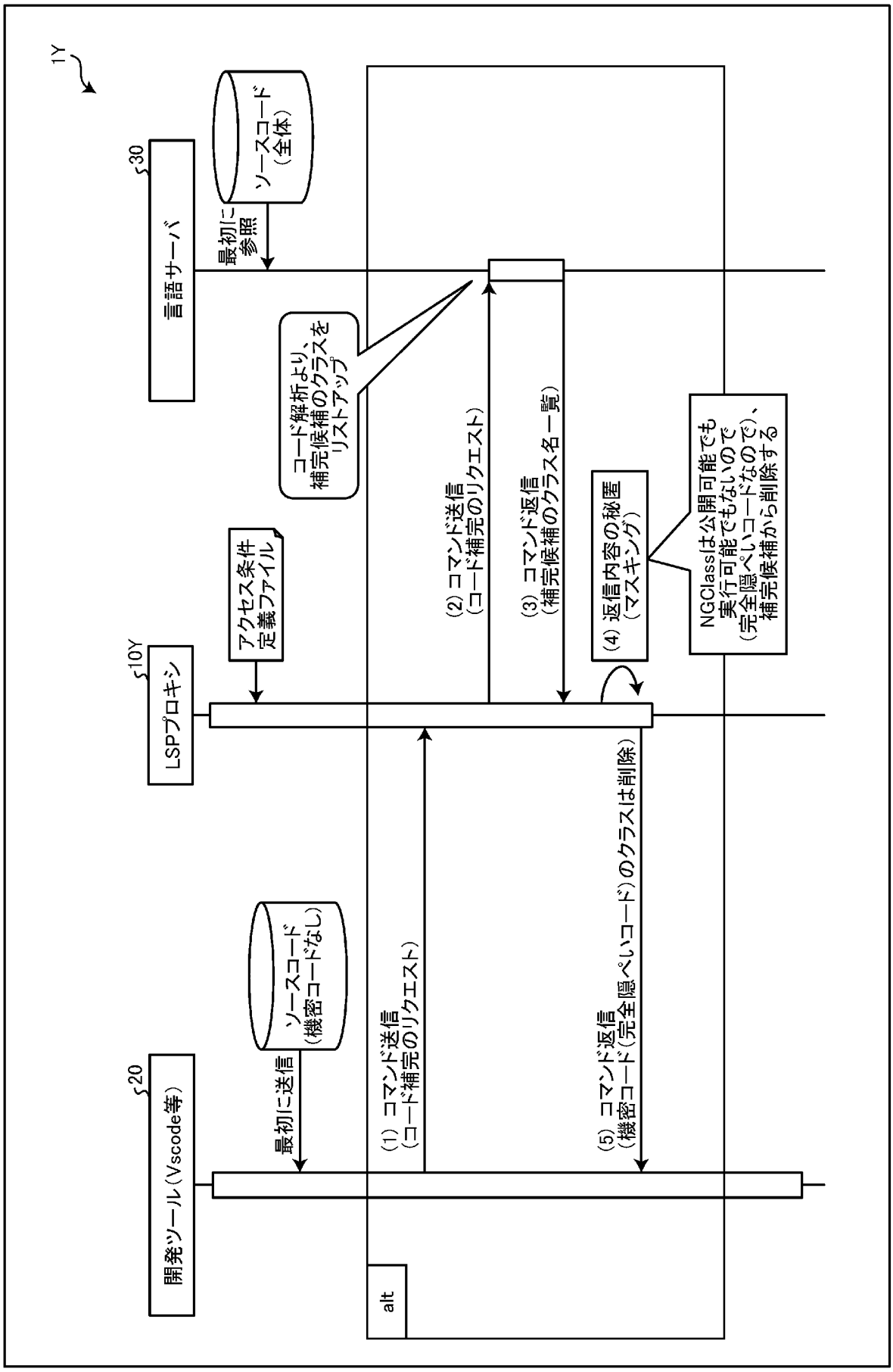
[図10]


コード補完リクエストへの応答に
機密コードの情報が含まれる例
1
リモート開発環境
開発ツール（Vscode等）
20
LSPプロキシ
10
言語サーバ
30
(0) com.xxx.と入力
GUI
編集操作
開発者
(1) コマンド送信
（コード補完の
通知）
(2) コマンド送信
（コード補完の
通知）
(3) コマンド返信
（補完候補の
クラス名一覧）
(4) コマンド返信
（補完候補の
クラス名一覧）
(5)
補完候補
の表示
アクセス
チェック部
121
com.xxx.OKClass
public void aaa() {
com.xxx.
}
com.xxx.OKClass
public void aaa() {
com.xxx.
}
com.xxx.OKClass
内部隠
べいコード
com.xxx.OKClass
完全隠べい
コード
com.xxx.OKClass2
...
言語サーバでソースコードを解析し、
com.xxx以下のクラス（補完候補）が完
全隠べいコードのクラスも含めてリスト
アップされ、開発ツールに返信
“result”: {
item: [
{ data: { decl_signature: “com.xxx.OKClass”, ... }, ... },
{ data: { decl_signature: “com.xxx.RefNGClass”, ... }, ... },
{ data: { decl_signature: “com.xxx.NGClass”, ... }, ... },
{ data: { decl_signature: “com.xxx.OKClass2”, ... }, ... },
OKClass
RefNGClass
NGClass
OKClass2
名前やパスが分かれば、
解析（コンパイル等）の要求が可能
⇒たとえ開発ツール側に機密コード
がなくても、機密コードの
内部情報（プロパティやメソッドの名
前）が取得できるため、
実装内容が把握できてしまう

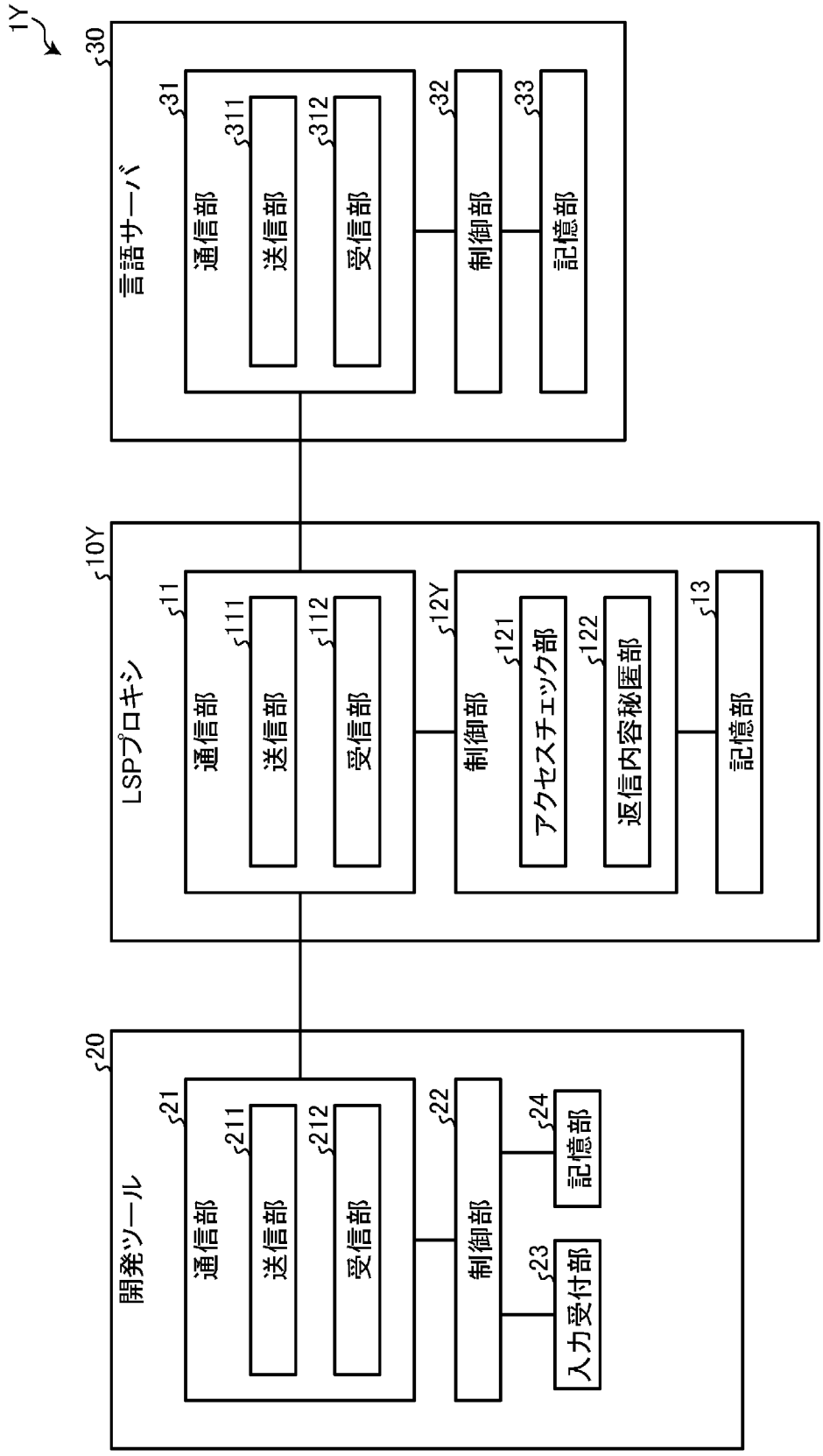
[図11]


LSPマスキング方式の概要
(コード補完内容の秘匿
1Y
リモート開発環境
言語サーバ
30
LSPプロキシ
10Y
開発ツール(Vscode等)
20
com.xxx.OKClass
public void aaa() {
OKClass2.bbbb();
com.xxx.
}
com.xxx.OKClass
内部隠ぺいコード
com.xxx.OKClass
完全隠ぺいコード
com.xxx.OKClass2
(1) コマンドの送信(コード補完のリクエスト)
(2) コマンドの送信(コード補完のリクエスト)
121
アクセス権チェック部
122
返信内容秘匿部
(3) コマンド返信(補完候補のクラス名一覧)
LSPプロキシの秘匿処理により、機密コード(完全隠ぺいコード)のクラスは削除する
"result": {
item: [
{ data: { decl_signature: "com.xxx.OKClass", ... }, ... },
{ data: { decl_signature: "com.xxx.RefNGClass", ... }, ... },
{ data: { decl_signature: "com.xxx.NGClass", ... }, ... },
{ data: { decl_signature: "com.xxx.OKClass2", ... }, ... },
(4)
返信内容のマスキング
(機密コード(完全隠ぺいコード)のクラスは削除)
(5) コマンド返信(マスキングされた補完候補のクラス名一覧)
OKClass
RefNGClass
OKClass2
(6)
補完候補の表示
(0) com.xxx.と入力
編集操作
GUI
開発者
補完候補の中に、機密コードのクラスは表示されないため、機密コード(完全隠ぺいコード)の存在は分からない

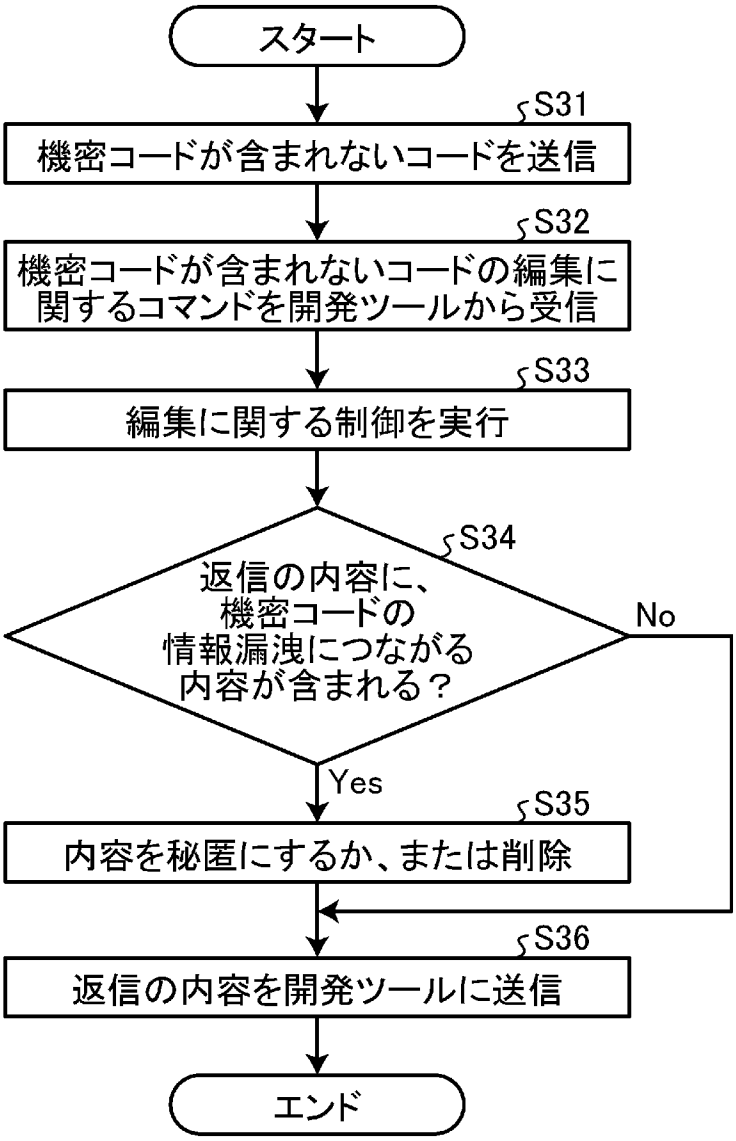
[図12]


1Y
言語サーバ
30
ソースコード
(全体)
最初に
参照
コード解析より、
補完候補のクラスを
リストアップ
(2) コマンド送信
(コード補完のリクエスト)
(3) コマンド返信
(補完候補のクラス名一覧)
(4) 返信内容の秘匿
(マスキング)
NGClassは公開可能でも
実行可能でもないので
(完全隠ぺいコードなので)、
補完候補から削除する
アクセス条件
定義ファイル
LSPプロキシ
10Y
ソースコード
(機密コードなし)
最初に送信
(1) コマンド送信
(コード補完のリクエスト)
(5) コマンド返信
(機密コード(完全隠ぺいコード)のクラスは削除)
開発ツール(Vscode等)
20
alt

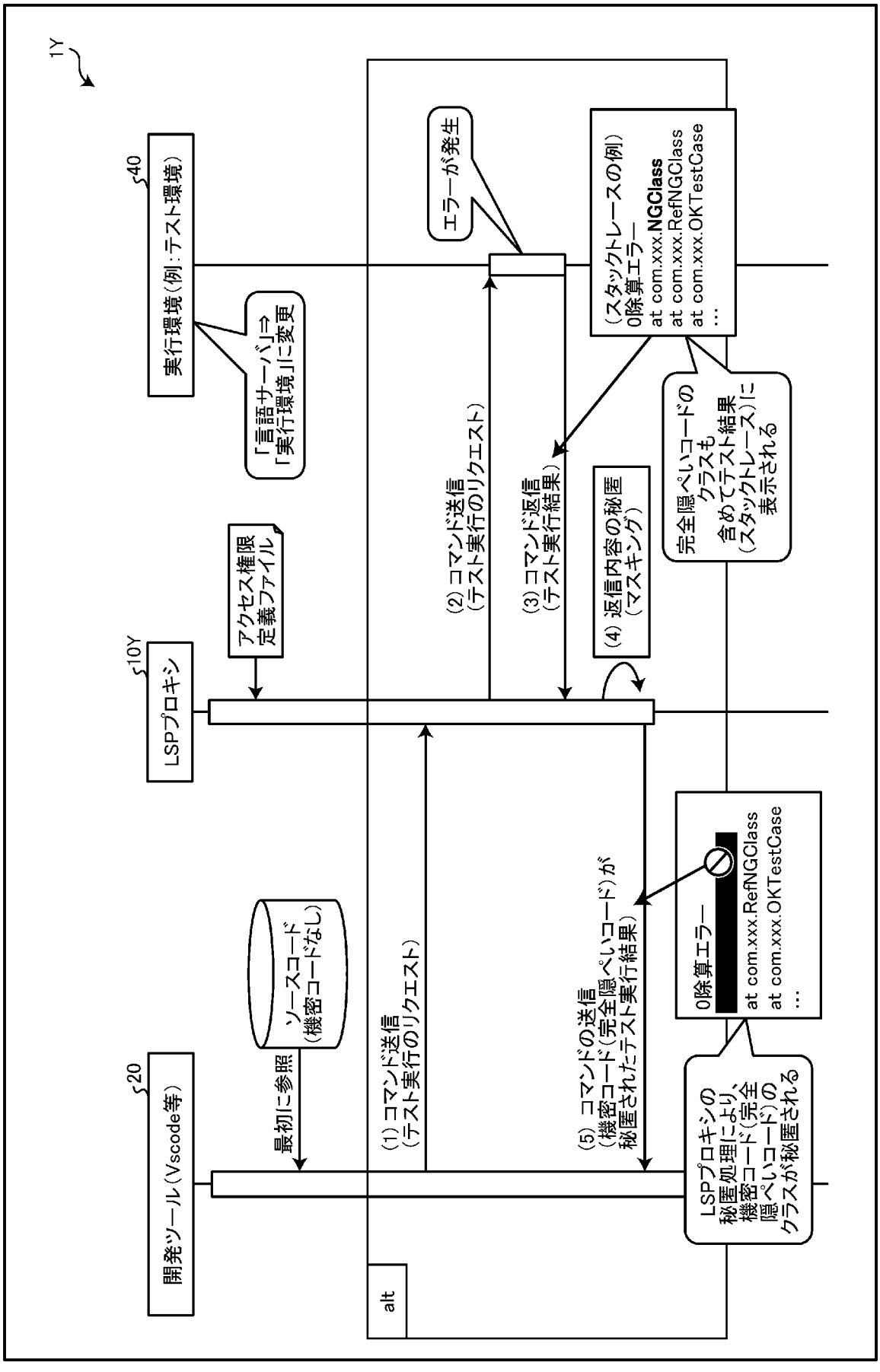
[図13]


1Y
30
言語サーバ
31
通信部
311
送信部
312
受信部
32
制御部
33
記憶部
10Y
LSPプロキシ
11
通信部
111
送信部
112
受信部
12Y
制御部
121
アクセスチェック部
122
返信内容秘匿部
13
記憶部
20
開発ツール
21
通信部
211
送信部
212
受信部
22
制御部
23
入力受付部
24
記憶部

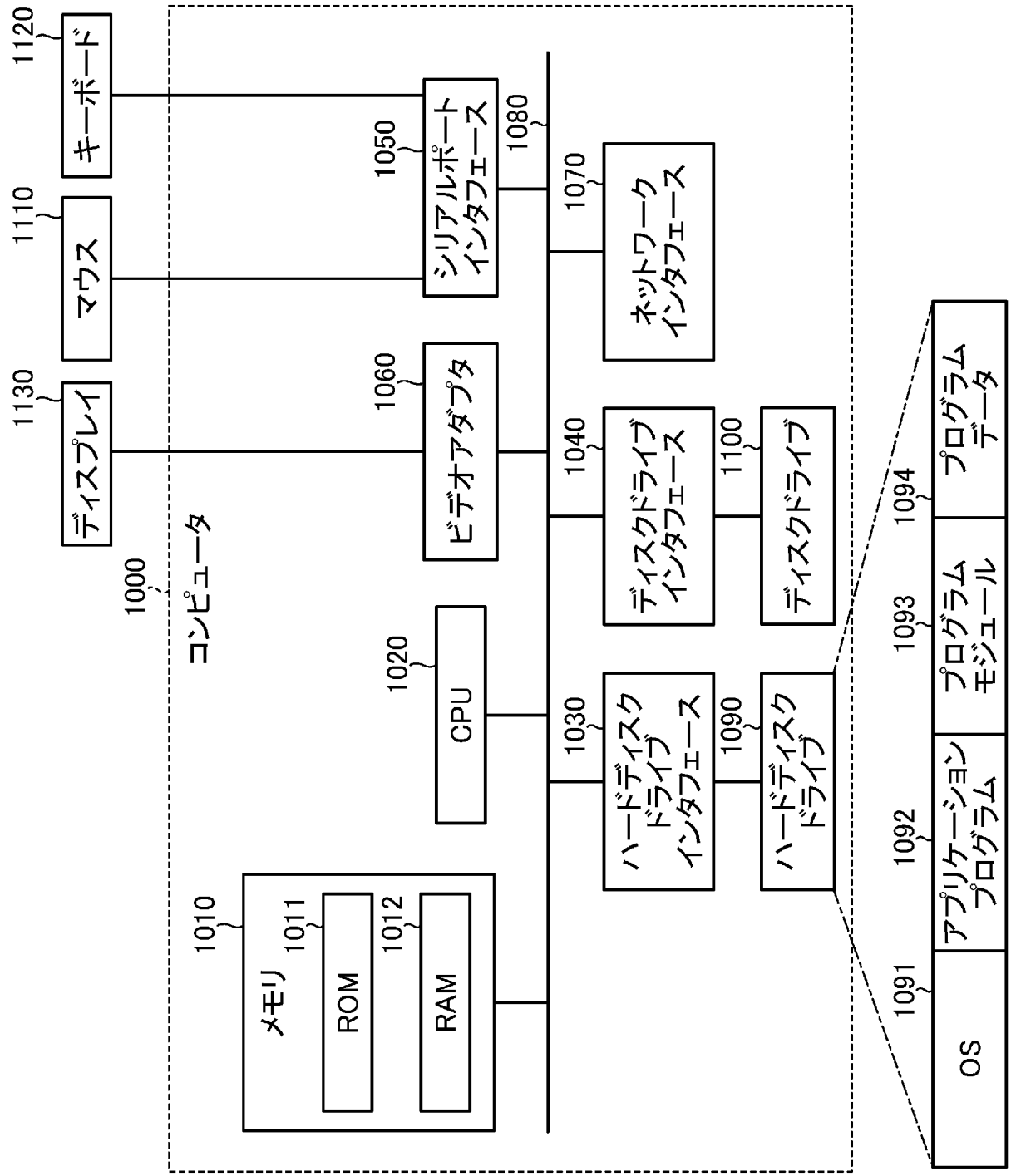
[図14]


スタート
S31
機密コードが含まれないコードを送信
S32
機密コードが含まれないコードの編集に関するコマンドを開発ツールから受信
S33
編集に関する制御を実行
S34
返信の内容に、機密コードの情報漏洩につながる内容が含まれる？
No
Yes
S35
内容を秘匿にするか、または削除
S36
返信の内容を開発ツールに送信
エンド

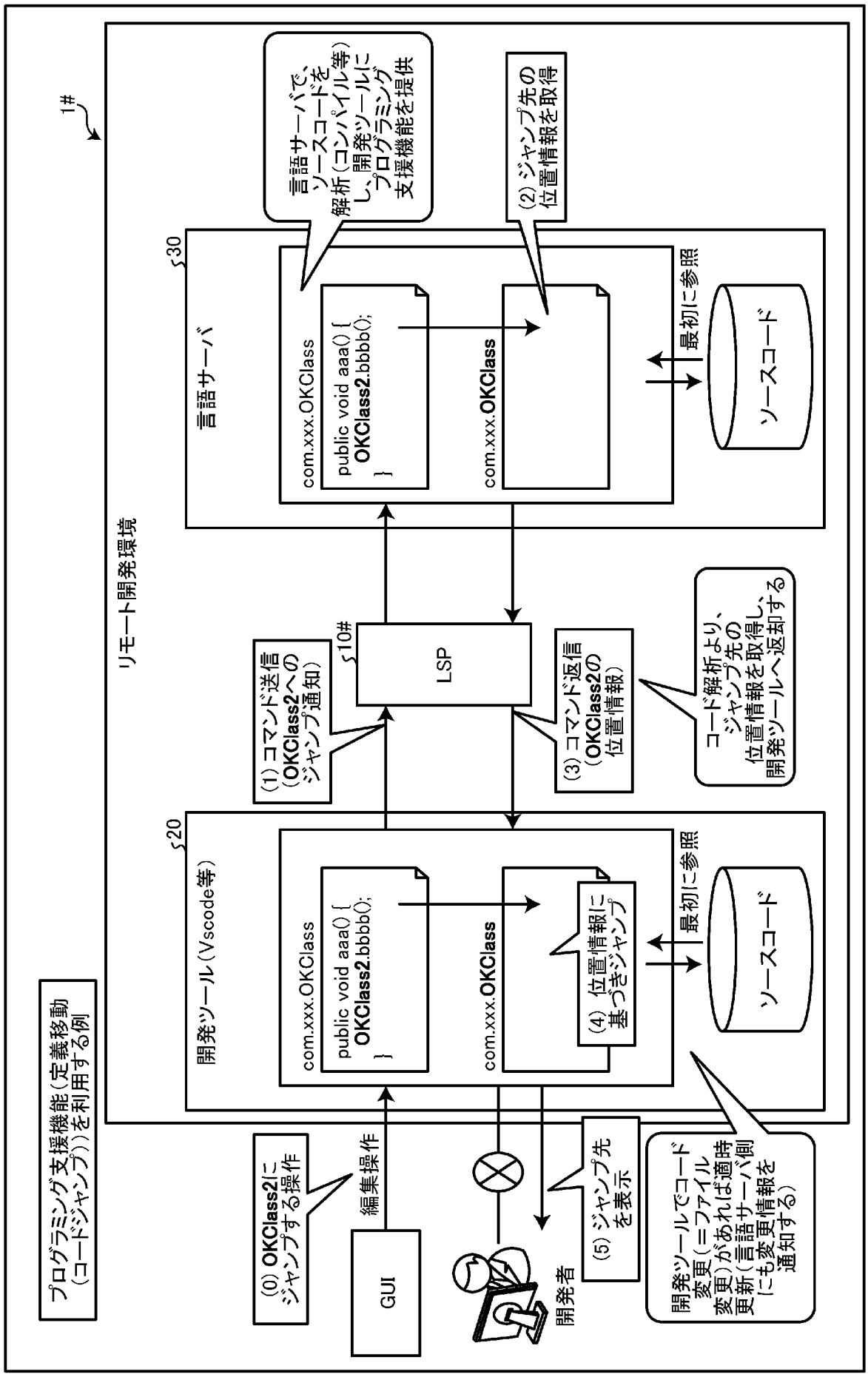
[図15]


1Y
開発ツール（Vscode等）
20
LSPプロキシ
10Y
実行環境（例：テスト環境）
40
alt
最初に参照
ソースコード
（機密コードなし）
アクセス権限
定義ファイル
「言語サーバ」⇒
「実行環境」に変更
(1) コマンド送信
（テスト実行のリクエスト）
(2) コマンド送信
（テスト実行のリクエスト）
エラーが発生
(3) コマンド返信
（テスト実行結果）
(4) 返信内容の秘匿
（マスキング）
(5) コマンドの送信
（機密コード（完全隠ぺいコード）が
秘匿されたテスト実行結果）
（スタックトレースの例）
0除算エラー
at com.xxx.NGClass
at com.xxx.RefNGClass
at com.xxx.OKTestCase
…
完全隠ぺいコードの
クラスも
含めてテスト結果（スタックトレース）に
表示される
0除算エラー
at com.xxx.RefNGClass
at com.xxx.OKTestCase
…
LSPプロキシの
秘匿処理により、
機密コード（完全
隠ぺいコード）の
クラスが秘匿される

[図16]


1000
コンピュータ
1010
メモリ
1011
ROM
1012
RAM
1020
CPU
1030
ハードディスクドライブインタフェース
1040
ディスクドライブインタフェース
1050
シリアルポートインタフェース
1060
ビデオアダプタ
1070
ネットワークインタフェース
1080
1090
ハードディスクドライブ
1100
ディスクドライブ
1110
マウス
1120
キーボード
1130
ディスプレイ
1091
OS
1092
アプリケーションプログラム
1093
プログラムモジュール
1094
プログラムデータ

[図17]


1#
リモート開発環境
プログラミング支援機能（定義移動（コードジャンプ））を利用する例
言語サーバ
30
言語サーバで、ソースコードを解析（コンパイル等）し、開発ツールにプログラミング支援機能を提供
(2) ジャンプ先の位置情報を取得
com.xxx.OKClass
public void aaa() {
OKClass2.bbbb();
}
com.xxx.OKClass
最初に参照
ソースコード
開発ツール（Vscode等）
20
(1) コマンド送信（OKClass2へのジャンプ通知）
10#
LSP
(3) コマンド返信（OKClass2の位置情報）
コード解析より、ジャンプ先の位置情報を取得し、開発ツールへ返却する
(0) OKClass2にジャンプする操作
編集操作
GUI
開発者
(4) 位置情報に基づきジャンプ
(5) ジャンプ先を表示
開発ツールでコード変更（＝ファイル変更）があれば適時更新（言語サーバ側にも変更情報を通知する）

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2023/025784

A. CLASSIFICATION OF SUBJECT MATTER

G06F 21/60(2013.01)i; ***G06F 21/57***(2013.01)i
FI: G06F21/60; G06F21/57

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F21/60; G06F21/57

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Published examined utility model applications of Japan 1922-1996
Published unexamined utility model applications of Japan 1971-2023
Registered utility model specifications of Japan 1996-2023
Published registered utility model applications of Japan 1994-2023

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 2009-129326 A (CANON KABUSHIKI KAISHA) 11 June 2009 (2009-06-11) abstract	1-8
A	WO 2008/064049 A2 (CISCO TECH INC.) 29 May 2008 (2008-05-29) abstract	1-8

☐ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
28 August 2023	**12 September 2023**
Name and mailing address of the ISA/JP **Japan Patent Office (ISA/JP)** **3-4-3 Kasumigaseki, Chiyoda-ku, Tokyo 100-8915** **Japan**	Authorized officer Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/JP2023/025784

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
JP 2009-129326 A	11 June 2009	(Family: none)	
WO 2008/064049 A2	29 May 2008	US 2008/0120415 A1	
		GB 2457840 A	
		CN 101187965 A	

国際調査報告

国際出願番号
PCT/JP2023/025784

A. 発明の属する分野の分類（国際特許分類（ＩＰＣ））

G06F 21/60(2013.01)i; G06F 21/57(2013.01)i
FI: G06F21/60; G06F21/57

B. 調査を行った分野

調査を行った最小限資料（国際特許分類（ＩＰＣ））

G06F21/60; G06F21/57

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	１９２２－１９９６年
日本国公開実用新案公報	１９７１－２０２３年
日本国実用新案登録公報	１９９６－２０２３年
日本国登録実用新案公報	１９９４－２０２３年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

C. 関連すると認められる文献

引用文献のカテゴリー*	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求項の番号
A	JP 2009-129326 A (キヤノン株式会社) 11.06.2009 (2009-06-11) [要約]	1-8
A	WO 2008/064049 A2 (CISCO TECHNOLOGY, INC.) 29.05.2008 (2008-05-29) abstract	1-8

☐ C欄の続きにも文献が列挙されている。

☑ パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー
“A” 特に関連のある文献ではなく、一般的技術水準を示すもの
“E” 国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
“L” 優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）
“O” 口頭による開示、使用、展示等に言及する文献
“P” 国際出願日前で、かつ優先権の主張の基礎となる出願の日の後に公表された文献
“T” 国際出願日又は優先日後に公表された文献であって出願と抵触するものではなく、発明の原理又は理論の理解のために引用するもの
“X” 特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
“Y” 特に関連のある文献であって、当該文献と他の１以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
“&” 同一パテントファミリー文献

国際調査を完了した日	国際調査報告の発送日
28.08.2023	12.09.2023

名称及びあて先
日本国特許庁(ISA/JP)
〒100-8915
日本国
東京都千代田区霞が関三丁目４番３号

権限のある職員（特許庁審査官）
平井　誠 5S 9071
電話番号 03-3581-1101 内線 3546

国際調査報告
パテントファミリーに関する情報

国際出願番号
PCT/JP2023/025784

引用文献	公表日	パテントファミリー文献	公表日
JP 2009-129326 A	11.06.2009	（ファミリーなし）	
WO 2008/064049 A2	29.05.2008	US 2008/0120415 A1	
		GB 2457840 A	
		CN 101187965 A	